

Communication Protocols

User Manual

Version © 2010-2022 Exor International S.p.A.

Disclaimer

© 2010-2022 Exor International S.p.A.

Subject to change without notice

The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided 'as is' without warranty of any kind.

- The copyright of this manual is owned by Exor International S.p.A.
- Unauthorized reproduction of this manual is strictly prohibited.
- Windows is a registered trademark of Microsoft Corporation in the U.S. and other countries.
- Ethernet is a registered trademark of FUJIFILM Business Innovation Co., Ltd. and Xerox Corporation.
- QR Code is a registered trademark of DENSO WAVE INCORPORATED.
- Other company and product names are trademarks or registered trademarks of their respective companies.

Third-party brands and names are the property of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logo, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred.

This products/software contains software licensed under the GNU General Public License, Version 2.0 (GPL V2.0), software licensed under the GNU LESSER General Public License, Version 2.1 (LGPL V2.1), and/or open source software other than the software licensed under the GPL V2.0 and/or LGPL V2.1. The software open source included is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

For more details or for a copy of sources where required by license, as well as the respective copyright notice, please ref to device settings menu or https://www.exorint.com or contact us at support.it@exorint.com



Contents

A-B DF1	3
A-B DH-485	15
A-B ENET	
ABB Mint Controller HCP	40
BACnet	47
Beckhoff ADS	89
CAN Direct v2.0x	104
CANopen HMI	114
CANopen SDO	121
Client System Variables	127
CODESYS V2 ETH	129
CODESYS V2 SER	142
CODESYS V3 ETH	151
Control Techniques Modbus TCP	162
CT Modbus CMP ETH	166
Delta Modbus RTU	172
Direct Serial	182
Direct Socket	190
DMX512 Digital Multiplex	201
Eaton Suconet-K	206
Environment Variables	211
Ethernet/IP CIP	212
Fatek FACON ETH	237
Fatek FACON SER	244
GE Intelligent Platforms SNP	250
GE Intelligent Platforms SRTP	261
GE SRTP	271
Hitachi SER	282
Hitachi ETH	287
IDEC Maintenance	292

J1939	303
Jetter Ext ETH	315
Keyence KV	. 323
Koyo DL	. 333
Koyo DL ETH	. 339
KNX TP/IP	. 344
Lenze CANopen	. 364
Modbus RTU	368
Modbus RTU Server	. 383
Modbus TCP	398
Modbus TCP Server	415
Mitsubishi FX ETH	. 426
Mitsubishi FX SER	440
Mitsubishi iQ/Q/L ETH	449
Mitsubishi iQ/Q/L SER	462
NMEA 0183	. 471
NMEA 2000	. 495
Omron FINS ETH	. 509
Omron FINS SER	. 522
OPC UA Client	. 531
Panasonic FP	545
Ping	. 552
ProConOS ETH	557
Profibus DP	565
Profibus DP S7	569
Rexroth IndraControl	614
ROBOX BCC/31	620
ROC Plus	. 629
SAIA S-BUS	635
SAIA S-BUS ETH	. 644

Simatic S7 PPI	651
Siemens S7 Optimized	.658
Simatic S7 ETH	. 673
Simatic S7 MPI	.712
System Variables	.747
Uni-Telway	749
Variables	757

A-B DF1

The A-B DF1 communication driver has been designed to connect HMI devices to a Allen-Bradley controllers through serial communication.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

A-B DF1		×
	Comm	ОК
Node ID	0	Cancel
Checksum type	BCC ~	
PLC Models		
PLC3 PLC5/10/12/15/25/30 PLC5/40/40L PLC5/60/60L SLC500 Fixed I/O SLC500 Modular I/O	~	

Element	Description
Node ID	Serial node associated to the PLC.
Checksum type	It can be BCC or CRC , depending on PLC settings.
PLC Models	PLC models available: • PLC3 • PLC5/10/12/15/25 • PLC5/40/40L • PLC5/60/60L • SLC500 Fixed I/O

Element	Description	
	 SLC500 Mod Micrologix 10 Micrologix 15 Ultra5000 	ular I/O 00 00
Comm	If clicked displays the	e communication parameters setup dialog.
	Comm Parameter Dia	alog
	Port Baudrate Parity Data bits Stop bits Mode	Com1 19200 even 8 1 RS-232
	Element	Parameter
	Port	Serial port selection.
		 COM1: device PLC port. COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board. COM3: optional Plug-In module plugged on Slot 3/4 for panels with 4 serial part on board.
Baudrate, Parity, Data Bits, Stop bits		Serial line parameters.
	Mode	Serial port mode. Available modes:
		• RS-232.
		 RS-485 (2 wires). RS-422 (4 wires)

Tag Editor Settings

In Tag Editor select the protocol A-B DF1.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

File Type	Element	Subindex		
Disc In 🔹	0	0		
File Num	Data Type		Arraysize	
7	short		0	
Conversion	Sub Elem	ent		
-	+/-		-	
	+/-			

Element	Description			
Memory Type	Memory Type	Description		
Type	Disc Out	Discrete output value. O resource on PLC.		
	Disc In	Discrete input value. I resource on PLC.		
	Status	Status value. S resource on PLC.		
	Bit	Bit value. B resource on PLC.		
	Timer	Timer value. T resource on PLC.		
	Counter	Counter value. C resource on PLC.		
	Control	Control value. R resource on PLC.		
	Integer	Integer value. N resource on PLC.		
	Float	Float value. F resource on PLC.		
Element	Represents the line of the resource while monitoring PLC values.			
Subindex	Represents the column of the resource while monitoring PLC values.			
File Num	Instance of resource of the PLC).		
Data Type	Available data types:			
	 boolean byte short int unsignedByte unsignedShort unsignedInt float double string binary See "Programming concepts" section in the main manual. Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]). 			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 			

Element	Description				
	Note: number of bytes corresponds to number of string characters if Encoding property set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.				
Sub Element	Allows to point t	int to specific part of a resource:			
	 0 (entire PRE ACC LEN POS 	resource)			
Conversion	sion Conversion to be applied to the tag.				
	Conversion				
	inv,swap2 Depending on d	Allowed BCD AB->BA ABCD->CDAB ABCD=>CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK Atta type selected, the list Allowed shows one or more conversion types.			
	Value	Description			
	Inv bits	inv: Invert all the bits of the tag.			
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)			
	Negate	neg : Set the opposite of tag value. <i>Example:</i>			
		25.36 → -25.36			
	AD -> BA	swapnibbles : Swap nibbles in a byte. <i>Example:</i> $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)			
	ABCD -> CDAB	swap2: Swap bytes in a word.			

Element	De

Description		
Value	Description	
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)	
ABCDEFGH	swap4: Swap bytes in a double word.	
-> GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)	
ABCNOP -	swap8: Swap bytes in a long word.	
> OPMDAB	Example: $142.366 \rightarrow .893553517.588905$ (in decimal format) 0.1000000110 00011100101110110100100101101000011100101	
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)	
	Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
Select conversion	on and click +. The selected item will be added to list Configured .	
If more conversi	ions are configured, they will be applied in order (from top to bottom (

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

Tag Import

Exporting Tags from PLC

The A-B DF1 tag import filter accepts symbol files with extension ".csv" created by the Rockwell RSLogix 500.

To create the file select Tool > Database > ASCII Export

RSLogix 500 Pro - UNTITLED		
File Edit View Search Comms	Tools Window Help	
D 🖻 🖬 🎒 🐰 🖻 💼	Options	2
OFFLINE 🛃 No Forces	Delete Unused Memory	
No Edits 🛃 Forces Disable	Database 🔹 🕨 ASCII Export	
Driver: AB_ETHIP-1	Security ASCII Import	
UNTITLED	Visual Basic	
⊡ - în Project	FactoryTalk Diagnostics Delete Unused Addr.	_
⊡	Add-In Manager Edit Using Excel	
🕺 1 Controller Properties	Edit Device Codes	
	Adjust Rung Offset Convert Rung Attachment	
⊕ লেলায়েলেলা Configuratio আলি Multipoint Monitor	n Rebuild DB	
🚊 📄 Program Files		
SYS0-		

From **CSV** tab select the data to be exported and give a name to the output csv file.

Documentation Database ASCII Exp	ort 🛛 🔀						
RSLogix 500 A.I. AB APS CSV							
┌─ Data to be exported:───── ┌─ D	estination file names and extensions :						
Addr/Symbol Desc. 🔽 👖	ile1 .CSV						
Instruction Comments 🔲 🗍	ile2 .CSV						
Page Title / Rung Desc. 🔲 🗍							
Symbol Groups 🗖	ile3 .CSV						
Program File Names 🗖 🗍	INTITLED						
AI/AB Address and Instruction description formatting :							
Characters per line in target database	✓ 20						
Treat Source Description as 5 lines (I	runcating each line if necessary)						
C Treat Source Description as 1 line (tr	incating from the end if necessary)						
ОК	Cancel Help						

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Т	ags	×								
ŀ	┣	_	Z	đ	ß	>]	₽	A 9B	B>	ŧ.	1
Di	ata			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

Multiple tag importers are available for this protocol. Please select the importer type and continue Version Type RSLogix500 v1.0 Linear Tag Editor exported xml 1.1 General
Version Type RSLogix500 v1.0 Linear Tag Editor exported xml 1.1 General
RSLogix500 v1.0 Linear Tag Editor exported xml 1.1 General
Tag Editor exported xml 1.1 General
OK Cancel

Importer	Description													
RSLogix500 v1.0	Requires an .csv file.													
Linear	All variables will be displayed at the same level.													
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.													
	Tags X													
	🕂 — 🎽 🕲 🔎 🔰 🚺 🕼 🖬 🔂 1													
	Data Tag URI													

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 — 🎽 🕲 🖉 🕻 📾 🕅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1			Y Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			Y Dictionary	
Holding Registers 3 unsignedShort			and a second sec	febra.
MRTU1 unsignedShort			Array	Taise
-MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
MRTU4 unsignedShort			Comment	
- MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
B ≱	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name -	Searches tags in the dictionary basing on filter combo- box item selected.

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

Communication Protocols addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.

		Current Cards A	vailable Filter All IO
PowerSuppl # Part # 0 Bul.1766 1762-IQ16 2 1762-IQ16 3 1762-0W16 4 1762-IF4 5 1762-0F4 7	Read 10 Config.	Part # 1762-IA8 1762-IF20F2 1762-IF4 1762-IQ8 1762-IQ80W6 1762-IQ16 1762-OA8 1762-OB8 1762-OB8 1762-OB16 1762-OW16 1762-IF4 1762-IF4	Description 8-Input 79/132 VAC Analog 2 Chan. Input, 2 Chan. Output Analog 4 Chan. Input, 2 Chan. Output 8-Input 10/30 VDC 8-Input 10/30 VDC 6-Output (RLY) 16-Input 10/30 VDC 8-Output 120/240 VAC 8-Output 120/240 VAC 8-Output (TRANS-SRC) 10/50 VDC 16-Output (TRANS-SRC) 10/50 VDC 8-Output (TRANS-SRC) 10/50 VDC 8-Output (RLY) 240 VAC 4-Channel Thermocouple Input Module 4-Channel RTD/Resistance Input Module
Adv Config	Help Hide All Cards	1762-0×6i	6-Ch High Current Isolated Relay Outputs Other Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

This example shows how to configure the Communication Protocols Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).

1:0.0 0 <th>MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - WORD 18</th> <th>ffset</th> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>11</th> <th>10</th> <th>9</th> <th>8</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> <th></th> <th>WORD 0</th>	MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - WORD 18	ffset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		WORD 0
1:0.1 0 <td>Micrologix 1400 Series A Micrologix 1400 Series A i - 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input</td> <td>:0.0</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A 🔫</td> <td></td>	Micrologix 1400 Series A Micrologix 1400 Series A i - 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input	:0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A 🔫	
1:0.2 0 <td>Micrologix 1400 Series A Micrologix 1400 Series A i - 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input</td> <td>:0.1</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	Micrologix 1400 Series A Micrologix 1400 Series A i - 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input	:0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
I:0.3 0 <td>MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A = 16-Input 10/30 VDC = Analog 4 Chan. Input = 4-Channel Analog I/V Output Module = WORD 18</td> <td>:0.2</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A = 16-Input 10/30 VDC = Analog 4 Chan. Input = 4-Channel Analog I/V Output Module = WORD 18	:0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.4 0 <td>MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A : 16-Input 10/30 VDC : 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input</td> <td>::0.3</td> <td>0</td> <td>۵</td> <td>0</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A : 16-Input 10/30 VDC : 16-Input 10/30 VDC - Analog 4 Chan. Input - Analog 4 Chan. Input	::0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	۵	0	0	Bul.1766 MicroLogix 1400 Series A	
1:1.05 0 <td>MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - halog 4 Chan. Input - Analog 4 Chan. Input</td> <td>:0.4</td> <td>0</td> <td>0</td> <td>٥</td> <td>0</td> <td>0</td> <td>0</td> <td>۵</td> <td>0</td> <td>0</td> <td>٥</td> <td>0</td> <td>0</td> <td>0</td> <td>٥</td> <td>0</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	MicroLogix 1400 Series A MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - halog 4 Chan. Input - Analog 4 Chan. Input	:0.4	0	0	٥	0	0	0	۵	0	0	٥	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
I:0.6 0 <td>MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</td> <td>:0.5</td> <td>0</td> <td>Ď</td> <td>0</td> <td>٥</td> <td>0</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	MicroLogix 1400 Series A MicroLogix 1400 Series A - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:0.5	0	Ď	0	0	0	0	0	0	0	0	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
I:0.7 0 <td>MicroLogix 1400 Series A - 16-Input 10/30 VDC - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</td> <td>:0.6</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	MicroLogix 1400 Series A - 16-Input 10/30 VDC - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
I:1.0 0 <td><pre>- 16-Input 10/30 VDC - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</pre></td> <td>:0.7</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>٥</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>٥</td> <td>0</td> <td>Bul.1766 MicroLogix 1400 Series A</td> <td></td>	<pre>- 16-Input 10/30 VDC - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</pre>	:0.7	0	0	0	0	0	0	0	0	0	٥	0	0	0	0	٥	0	Bul.1766 MicroLogix 1400 Series A	
I:2.0 0 <td><pre>i - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</pre> WORD 18</td> <td>:1.0</td> <td>0</td> <td>1762-IQ16 - 16-Input 10/30 VDC</td> <td></td>	<pre>i - 16-Input 10/30 VDC - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module</pre> WORD 18	:1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IQ16 - 16-Input 10/30 VDC	
I:4.0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.1 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.1 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.2 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.3 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.4 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	٥	٥	0	1762-IQ16 - 16-Input 10/30 VDC	
I:4.1 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.2 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.3 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.4 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:4.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
I:4.2 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.3 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input I:4.4 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:4.1	0	0	Ű	0	0	0	٥	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
I:4.3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan, Input I:4.4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan, Input	- Analog 4 Chan, Input - Analog 4 Chan, Input - Analog 4 Chan, Input - Analog 4 Chan, Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:4.2	0	0	C	0	0	0	0	0	0	0	0	0	0	o	0	0	1762-IF4 - Analog 4 Chan. Input	
I:4.4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- Analog 4 Chan. Input - Analog 4 Chan. Input - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:4.3	0	0	0	0	٥	0	0	0	0	0	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan, Input	
	- Analog 4 Chan. Input - Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	::4.4	0	0	0	0	0	0	٥	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
I:4.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- Analog 4 Chan. Input - 4-Channel Analog I/V Output Module - 4-Channel Analog I/V Output Module	:4.5	0	0	0	O	0	0	٥	0	0	٥	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan. Input	
I:4.6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-IF4 - Analog 4 Chan. Input	- 4-Channel Analog I/V Output Module WORD 18 - 4-Channel Analog I/V Output Module -	:4.6	0	0	0	0	0	0	0	0	Ó	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
I:5.0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-0F4 - 4-Channel Analog I/V Output Module	- 4-Channel Analog I/V Output Module	:5.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-OF4 - 4-Channel Analog I/V Output Module	WORD 1
I:5.1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1762-0F4 - 4-Channel Analog I/V Output Module 🔫 🚽 🚽		:5.1	0	0	0	0	0	0	0	0	0	٥	0	0	0	0	0	0	1762-OF4 - 4-Channel Analog I/V Output Module -	
<u>ا ا</u>			0/0																Radix Binary	
Lin/0 Radik Bray	Radix Binary	Symbol		_	_	_	_		_	_	_		_	_	_	_	_	_	Columns: 16 💌	

The following figure shows the Communication Protocols Tag configuration.

A-B DF1		×
A-B DF1		
File Type Disc Out V	Element Subindex	
File Num	Data Type Arraysize boolean 0	
Conversion	Sub Element	
1	+/- 0 ~	
	OK Cancel Apply	Help

The Communication Protocols Tag configured in the example above points on the element shown in the following figure.

-		-				10		-			÷	*		•	÷	-	
OLLSEC	12	14	13	12	-	10	9	8	-	0		4	ų	4	1	0	
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	ů	0	2
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
0:1.3	Pale	m	ant	-0	0	0	0	0	0	0	0	0	0	0	1	1	
0:1.4	en	-	- m		0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	~
•																•	E
0	:1/19			_	_		_	_	_			Rad	ix Bi	nary			
Symbol:														olut	mne	16	Y
Desc:															-	_	
	1	Pro	nortic			11				13	Fore		1		Hel		1

Examples

I:0/19 (I1:0.1/3 in word terms) – 20th Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

A-B DH-485

The A-B DH-485 communication driver has been designed to connect HMI devices to a Allen-Bradley controllers through serial communication.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

A-B DH-485		×
PLC Network	Comm	ОК
Alias		Cancel
Panel ID	2	
Slave ID	1	
MaxID	2	
PLC Models		
SLC500 Fixed I/O		
SLC500 Modular I/O Micrologix 1000		
Micrologix 1500		

Element	Description
Panel ID	Serial node associated to the HMI.
Slave ID	Serial node associated to the PLC.
MaxID	Represent the maximum ID available in the serial network.
PLC	PLC models available:
Models	SLC500 Fixed I/O



Element	Description			
	A-B DH-485		×	
	PLC Network	Comm	ОК	
	Alias		Cancel	
	Panel ID	2		
	Slave ID	-1		
	MaxID	A-B DH-485		
	PLC Models	Alias	1	OK Cancel
	Micrologix 1000 Micrologix 1500	Slave ID	1	
	Slaves	PLC Models		
	Slave Id Mod	el SLC500 Modular I/O Micrologix 1000 Micrologix 1500		

Tag Editor Settings

In Tag Editor select the protocol **A-B DH-485**.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

File Type	Element	Subindex		
Disc In 🔹	0	0 👻		
File Num	Data Type		Arraysize	
7	short	•	0	
Conversion	Sub Elem	nent		
	+/-		•	

Element	Description		
Memory Type	Memory Type	Description	
.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Disc Out	Discrete output value. O resource on PLC.	
	Disc In	Discrete input value. I resource on PLC.	
	Status	Status value. S resource on PLC.	
	Bit	Bit value. B resource on PLC.	
	Timer	Timer value. T resource on PLC.	
	Counter	Counter value. C resource on PLC.	
	Control	Control value. R resource on PLC.	
	Integer	Integer value. N resource on PLC.	
	Float	Float value. F resource on PLC.	
	String	String value. STR resource on PLC.	
Element	Represents the line of the resource while monitoring PLC values.		
Subindex	Represents the column of the r	esource while monitoring PLC values.	
File Num	Instance of resource of the PLC.		



Element	Description		
Data Type	Available data types:		
	• boolean		
	• byte		
	 snort int 		
	 unsignedByte 		
	unsignedShort		
	unsignedInt		
	• float		
	 double string binary 		
	See "Programming concepts" section in the main manual.		
	Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]).		
Arraysize	In case of array tag, this property represents the number of array elements.		
	 In case of string tag, this property represents the maximum number of bytes available in the string tag. 		
	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor.		
	If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.		

Element	Description		
Sub Element	Allows to point t	o specific part of a resource:	
	 0 (entire PRE ACC LEN POS 	resource)	
Conversion	Conversion to be applied to the tag.		
	inv,swap2	Allowed Configured BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK	
	Depending on data type selected, the list Allowed shows one or more conversion types Value Description		
	Inv bits	inv : Invert all the bits of the tag.	
	Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	neg: Set the opposite of tag value.	
	<i>Example:</i> 25.36 → -25.36		
	AB -> BA	swapnibbles: Swap nibbles in a byte.	
	Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
	ABCD ->	swap2: Swap bytes in a word.	
	CDABExample: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.	

Element	Description	
	Value	Description
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -	swap8: Swap bytes in a long word.
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ $0001\ 1100101110110100001011010000111001010101$
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversi	on and click +. The selected item will be added to list Configured .
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom of list
	Use the arrow b	outtons to order the configured conversions.

Tag Import

Exporting Tags from PLC

The A-B DH-485 tag import filter accepts symbol files with extension ".csv" created by the Rockwell RSLogix 500.

To create the file select Tool > Database > ASCII Export

RSLogix 500 Pro - UNTITLED		
File Edit View Search Comms	Tools Window Help	
D 😅 🖬 🎒 🕺 🖻 💼	Options	🔽 🖌 🖓 🔽 🛛
OFFLINE 🛓 No Forces	Delete Unused Memory	∃E∃∕E <> <0> <00> abl abs
No Edits 🛃 Forces Disable	Database 🔹 🕨	ASCII Export
Driver: AB_ETHIP-1	Security 🕨 🕨	ASCII Import
UNTITLED	Visual Basic Compare FactoryTalk Diagnostics Add-In Manager	Native Import Delete DB Delete Unused Addr. Edit Using Excel Edit Device Codes Adjust Rung Offset
Multipoint Monitor	n	Convert Rung Attachment Rebuild DB
Program Files		

From **CSV** tab select the data to be exported and give a name to the output csv file.

Docu	mentation Database ASCII	Export	
RSI	Logix 500 A.I. AB APS CS	5V	
	Data to be exported :	Destination file names and ext	ensions :
A	ddr/Symbol Desc. 🔽	file1	.CSV
lr	nstruction Comments	file2	.CSV
P	age Title / Rung Desc. 🛛 🗖		
S	ymbol Groups 📃	file3	.CSV
P	rogram File Names 🛛 🗖	UNTITLED	
	AI/AB Address and Instruction de	scription formatting :	
	Characters per line in target data	abase 💌 20	
0	Treat Source Description as 5 li	nes (truncating each line if neces:	sary)
0	Treat Source Description as 1 li	ne (truncating from the end if nece	essary)
		Cancel	Help
			пер

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	z	đ	ß	>]	₽	A 9B	B>	ŧ.	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio		×								
i Multiple tag impo	rters are available for this protocol. Please select the	importer type and continue.								
Version	Туре									
RSLogix500 v1.0	Linear									
Tag Editor exported xml 1.1 General										
		OK Cancel								

Importer	Description												
RSLogix500 v1.0	Requires an .csv file.												
Linear	All variables will be displayed at the same level.												
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.												
	Tags x												
	+ - 🎽 🕲 🔊 🚺 🕼 📾 🖄												
	Data												

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
+ - 👗 🕲 🖉 >] [> 🗞 🖬 前	R Search Trilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			Y Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			✓ Dictionary	
Holding Registers 3 UnsignedShort			Array	false
MRTU1 unsignedShort			Array size	0
MPTU2 unsignedShort			Arravinday Subinday	400003
MDT114 unsignedShort			Anayindex.Subindex	100003
MDTLIE unsignedShort			Comment	
MRT05 unsignedshort			Data type	unsignedShort

Toolbar item	Description
Ka	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
δā.	Update Tag(s). Click on this icon to update the tags in the project, due a
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

Communication Protocols addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.

		Current Cards A	vailable Filter All IO 🗨
		Part #	Description
	Read IU Config.	1762-IA8	8-Input 79/132 VAC
		1762-IF20F2	Analog 2 Chan. Input, 2 Chan. Output
<u>P</u> owerSupply	J	1762-IF4	Analog 4 Chan. Input
-		1762-IQ8	8-Input 10/30 VDC
# Part #	Description	1762-IQ80W6	8-Input 10/30 VDC 6-Output (RLY)
Bul 1766	MicroLogix 1400 Series A	1762-IQ16	16-Input 10/30 VDC
1762-1016	16-Input 10/30 VDC	1762-0A8	8-Output 120/240 VAC
1762-1016	16-Input 10/30 VDC	1762-OB8	8-Output (TRANS-SRC) 10/50 VDC
1762-0W16	16-Output (BLY) 240 VAC	1762-0B16	16-Output (TRANS-SRC) 10/50 VDC
1762-IF4	Analog 4 Chan Input	1762-0W8	8-Output Relay
1762-0F4	4-Channel Analog I/V Output Module	1762-0W16	16-Output (RLY) 240 VAC
	, ended and ended	1762-IT4	4-Channel Thermocouple Input Module
8		1762-IR4	4-Channel RTD/Resistance Input Module
		1762-0F4	4-Channel Analog I/V Dutput Module
		1762-0×6	6-Ch High Current Isolated Relay Outputs
			Other Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

1

This example shows how to configure the Communication Protocols Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		WORD 0
I:0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A 🔫	
1:0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	۵	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.4	0	0	0	0	0	0	۵	0	0	٥	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.7	0	D	0	0	0	٥	0	0	0	٥	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
1:1.0	0	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IQ16 - 16-Input 10/30 VDC	
1:2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	٥	٥	0	1762-IQ16 - 16-Input 10/30 VDC	
I:4.0	0	0	0	0	0	0	0	0	0	٥	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.2	0	0	C	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.3	0	0	0	0	٥	0	0	0	0	0	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan, Input	
1:4.4	0	0	0	0	0	0	٥	0	0	٥	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.5	0	0	0	C	0	0	۵	0	0	٥	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:5.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-OF4 - 4-Channel Analog I/V Output Module	WORD 18
1:5.1	0	0	0	0	0	0	٥	0	0	٥	0	0	0	0	0	0	1762-OF4 - 4-Channel Analog I/V Output Module 🔫	
4																	•=	
	0/0	_	_	_	_	_	_	_	_	_	_	_	_	_	_		Badie Binary -	
Symbol			_		_		_		_				_	_	_		Columns: 16 -	
Desc							_						_	_	_			

The following figure shows the Communication Protocols Tag configuration.

A-B DH-485					×
A-B DH-485					
File Type Disc Out ~	Element	Subindex			
File Num	Data Type boolean	~	Arraysize		
Conversion	Sub Eler	nent			
1	+/-		\sim		
	C	K (Cancel	Apply	Help

The Communication Protocols Tag configured in the example above points on the element shown in the following figure.

Data F	ile O	0 (bin)		01	ITPL	J s	ubi	nde	x	-	~			-		X
Offset	15	14	13	12	11	10	9	8	7	6	5	4	0	2	1	0	
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	^
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
0:1.3	al	am	ant	-0	0	0	0	0	0	0	0	0	0	0	1	1	
0:1.4	ere	enne	em	0	0	0	0	0	0	0	0	0	0	1	0	0	Ē
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	~
•																,	E
0	1/19		_	_	_		_		_	_		Rad	ix Bi	nary	1		•
Symbol:														olu	mne	16	
Desc:												_	_	_	-	_	
00 -		Pro	perti	es		U	sag	8			Forc	es	1		Hel	þ	1

Examples

I:0/19 (I1:0.1/3 in word terms) – 20^{th} Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

A-B ENET

The A-B ENET communication protocol is normally used on the Allen-Bradley controllers via Ethernet communication.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

A-B ENET		×
PLC Network		ОК
Alias	PLC1	Cancel
IP address	192 . 168 . 10 . 5	
Port	44818	
Timeout	1000	
PLC Models PLC5 via NET-ENI PLC5/10-25 SLC500/Micrologix 1000/1200, Micrologix 1100/1400	/1500 via NET-ENI	

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP Address	Ethernet IP address of the controller.
Port	Port number used by the Ethernet interface.

Element	Description		
Timeout	Time delay in millisecond controller.	ls between two re	tries in case of missing response from the
PLC Network	Enable access to multiple option.	e networked cont	rollers. For every controller (slave) set the proper
	A-B ENET		×
	PLC Network		ок
	Alias		Cancel
	IP address	0 0 0	ionaries Enable Offline Algorithm available
	Port 44818		
	Timeout		
	PLC Models		
	PLC5/10-25 SLC500/Micrologix 1000/1200/1500 via NET Micrologix 1100/1400	-ENI	
	Slaves	Add Delete Modify	
	Slave Id Model		×
	192, 168, 10, 1 PLC5 via NET-ENI		ок
		Alias	PLC1 Cancel
		IP address	192 . 168 . 10 . 1
		Port	44818
		Timeout	1000
		PLC Models	
		PLC5/10-25 SLC500/Micrologix 1000/1 Micrologix 1100/1 200	200/1500 via NET-ENI
		MICrologix 1100/1400	

Controller configuration

The PLC has to be correctly configured to match the IP address configured in the Protocol Editor. Normally the PLC configuration can be left as default.

Channel Configuration		X
General Chan. 1 - Sys	tem Chan. 0 - System Chan. 0 - U	ser
Driver Ethernet	V	
Broadcast	Address: 0.0.0.0	DHRIO Link ID 0
Hardware	Address: 00:00:BC:1D:D1:FC	
IP	Address: 192.168.0.140	Pass Thru Routing 0
Subn	et Mask: 255.255.255.0	
Gateway	Address: 192.168.0.199	
- Protocol Control		
Bootp Enable	м	sg Connection Timeout (x 1mS): 15000
		Msg Reply Timeout (x 1mS): 3000
		Inactivity Timeout (x Min): 30
Contact:		
Location:		
	ОК С	ancel Apply Help

Configuring 1761-NET-ENI

Here is the procedure to configure the 1761-NET-ENI module using the Allen Bradley's ENI/ENIW Utility. The procedure requires a 1761-CBL-PM02 communication cable.

- 1. Connect the 8 pin din to the port 2 on the NET-ENI device and the 9 pin female D-shell to the computer COM port.
- 2. Connect the SLC 5/0x controller and go online.
- 3. In the Utility Settings tab, set COM Port and Baud Rate.

ENLIP Addr Message Routing Email Reset (Utility Set	ings Web Conlig Web Data Desc	i Heb
COM Port Parameter Upload Behavior	Parameter Download Behavior—	Load From Save To
	C All	File Load File Save
Baud Rate Image: Baud Rate Image: Control of Con		Defaults ENI ROM
Configuration Security Mask	Modilied	Iext Text
000.000.000.000		Device Values

4. In the ENI IP Addr tab, select the correct ENI Series from the list and set ENI IP Address, Subnet Mask and Baud Rate, if needed.

(ENI IP Addr) Message Routin	g Email Reset	Utility Setting: Web Config Web Data Desc	. <u>H</u> e	lp
ENI Series D	232 Baud Bate	Auto 🔽 CompactLogix Routing 🗖	Load From	-Save To-
Obtain via BootP		002.059.127.092	File Load	File <u>S</u> ave
Always	ENLIF Address	255 255 252 000	<u><u>E</u>NI</u>	ENI R <u>A</u> M
Fallback	Gateway		<u>D</u> elaults	ENI R <u>O</u> M
Ethernet Speed/Dupley	Security Mask 1	000.000.000	<u>T</u> ext	Text
Auto Negotiate	Security Mask 2	000.000.000		
			Device Value	s

5. Save the configuration to the NET-ENI device.



Two separate memory areas are reserved for saving the configuration : **ENI/RAM** (for temporary configurations) and **ENI/ROM** (for permanent configurations).

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

Communication Protocols addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.

		Current Cards A	vailable Filter All IO 🗨
PowerSuppl # Part # 9 Bul.1766 1 1762-IQ16 2 1762-IQ16 3 1762-0W16 4 1762-IF4 5 1762-0F4 5	Read 10 Config.	Part # 1762-IA8 1762-IF20F2 1762-IF4 1762-IQ8 1762-IQ80W6 1762-Q88 1762-Q88 1762-Q88 1762-Q816 1762-W16 1762-IF4 1762-IF4	Description 8-Input 79/132 VAC Analog 2 Chan. Input, 2 Chan. Output Analog 4 Chan. Input 8-Input 10/30 VDC 8-Input 10/30 VDC 8-Output 10/30 VDC 8-Output 120/240 VAC 8-Output (TRANS-SRC) 10/50 VDC 16-Output (TRANS-SRC) 10/50 VDC 8-Output Relay 16-Output Relay 16-Output (RLY) 240 VAC 4-Channel Thermocouple Input Module 4-Channel BTD/Resistance Input Module
		1762-0F4	4-Channel Analog I/V Dutput Module
Adu Canka		1762-0×61	6-Ch High Current Isolated Relay Outputs Other Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

This example shows how to configure the Communication Protocols Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).

LADEL	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		WORD 0
1:0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A 🔫	
:0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	۵	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.4	0	0	0	0	0	0	۵	0	0	٥	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.5	0	0	0	0	0	0	0	0	0	٥	0	0	0	٥	0	0	Bul.1766 MicroLogix 1400 Series A	
6.0:1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bul.1766 MicroLogix 1400 Series A	
1:0.7	0	0	0	0	0	٥	0	0	0	۵	0	0	0	٥	٥	0	Bul.1766 MicroLogix 1400 Series A	
1:1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IQ16 - 16-Input 10/30 VDC	
1:2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	٥	0	1762-IQ16 - 16-Input 10/30 VDC	
1:4.0	0	0	0	0	0	0	0	0	0	٥	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.1	0	0	0	0	0	0	٥	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.3	0	0	0	0	٥	0	0	0	0	0	0	0	0	٥	0	0	1762-IF4 - Analog 4 Chan, Input	
I:4.4	0	0	0	0	0	0	٥	0	0	٥	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.5	0	0	0	0	0	0	٥	0	0	٥	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:4.6	0	D	0	0	0	0	0	0	Ó	0	0	0	0	0	0	0	1762-IF4 - Analog 4 Chan. Input	
1:5.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1762-OF4 - 4-Channel Analog I/V Output Module	WORD 18
1:5.1	0	0	0	0	0	0	0	0	0	٥	0	0	0	0	0	0	1752 OF4 4 Channel Inclose T/U Output Medule	

The following figure shows the Communication Protocols Tag configuration.

File Type	Element	Subindex	
Disc Out	1	3 🔻	
File Num	Data Type	Arraysize	
0	boolean	• 0	
Conversion	Sub E	Element	
	+/- 0	•	

The Communication Protocols Tag configured in the example above points on the element shown in the following figure.

Data F	ile O	0 (I	bin))	OL	ITPL	JS	ubi	nde	x	-	~			-		×
Offset	15	14	13	12	11	10	9	8	7	6	5	4	(3)	2	1	0	
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	^
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
0:1.3		-	ant	-0	0	0	0	0	0	0	0	0	0	0	1	1	
0:1.4	ere	enne	ein		0	0	0	0	0	0	0	0	0	1	0	0	Γ
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	~
•																•	E
0	1/19		_	_	_		_	_	_	_		Radi	K Bi	nary	,		*
Symbol:														olu	nne	16	
Desc:	_				_				_		_	_		_	_	-	
00 -		Pro	perti	es		U	sage	е			Forc	es	1		He	b	1

Examples

I:0/19 (I1:0.1/3 in word terms) -20^{th} Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1

Parameter	Setting
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

File Type	Element	Subindex		
Node Override IP 👻	0	0 *		
File Num	Data Type		Arraysize	
7	unsignedBy	te 🛛 🔻	4	
Conversion	Sub Elen	nent		
	+/-		-	

Tag Import

Exporting Tags from PLC

The A-B Ethernet tag import filter accepts symbol files with extension ".csv" created by the Rockwell RSLogix 500.

To create the file select Tool > Database > ASCII Export

RSLogix 500 Pro - UNTITLED					
File Edit View Search Comms	Tools Window Help				
D 😅 🖬 🎒 🕺 🖻 💼	Options	🔽 🖌 🖓 💽 🗹 🖻			
OFFLINE 🛓 No Forces	Delete Unused Memory	∃E ∃⁄E <> <0> <00> abl abs			
No Edits 🛃 Forces Disable	Database 🔹 🕨	ASCII Export			
Driver: AB_ETHIP-1	Security 🕨 🕨	ASCII Import			
UNTITLED	Visual Basic Compare FactoryTalk Diagnostics	Native Import Delete DB Delete Unused Addr.			
E Controller	Add-In Manager	Edit Using Excel Edit Device Codes			
		Adjust Rung Offset Convert Rung Attachment			
	n	Rebuild DB			
⊡ — 🧰 Program Files — 📓 SYS 0 -					

From **CSV** tab select the data to be exported and give a name to the output csv file.

Docu	Documentation Database ASCII Export						
RSI	RSLogix 500 A.I. AB APS CSV						
	Data to be exported :	Destination file names and ext	ensions :				
A	ddr/Symbol Desc. 🔽	file1	.CSV				
lr	nstruction Comments	file2	.CSV				
P	age Title / Rung Desc. 🛛 🗖						
S	ymbol Groups 📃	file3	.CSV				
P	rogram File Names 🛛 🗖	UNTITLED					
	AI/AB Address and Instruction de	scription formatting :					
	Characters per line in target data	abase 💌 20					
0	Treat Source Description as 5 lines (truncating each line if necessary)						
0	C Treat Source Description as 1 line (truncating from the end if necessary)						
			пер				

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	Z	đ	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag impo	orters are available for this protocol. Please select the importer typ	e and continue.
Version	Туре	
RSLogix500 v1.0	Linear	
Tag Editor exported xm	l 1.1 General	
Watabad distingants flat		
watched dictionary file:		
	OK	Cancel
	UK UK	Cancer

Importer	Description						
RSLogix500 v1.0	Requires an .csv file.						
Linear	All variables will be displayed at the same level.						
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.						
	Tags 🗙						
	+ - 🎽 🕲 🖉 🚺 🚺 🕼 🖄						
	Data						

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

	Tags ;	×	Protocols																-
+	-	Ж (0 0	>]	Þ	₿ ₿	۵ð	翻	R	₽- Search	7	Filter by: Data	•	Ite	ms used:6/1	10000 Protocol:	Show all	Show all tags	•
Data					Тур	e			Con	nment				^	Property			Value	
L . •	1odbus 1	TCP:pr	ot1		Con	tainer									✓ Driver				
- N	Iodel: M	lodicon	Modbus(L-based)	-									Mo	odel		Modicon Modbus(1-based)	
	Holdi	ing Reg	gisters 1		unsi	igned	Short								Pro	otocol		Modbus TCP:prot1	
	Holdi	ing Reg	gisters 2		unsi	igned5	Short								✓ Diction	arv			
	Holdi	ing Re	gisters 3		unsi	igned	short								Δn	rav		false	
	MRI	U1			unsi	igned	short												
	MRT	U2			unsi	igned S	Short								An	ray size		U	
	MRTI	U3			unsi	igned 9	Short								An	rayindex.Subind	lex	400003	
	MRTI	U4			unsi	igned 9	Short								Co	omment			
	- MRTI	U5			unsi	igned	Short								Da	ata type		unsignedShort	

Toolbar item	Description
Re	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s). Click on this icon to update the tags in the project, due a
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid	The device did received a response with invalid	Check if the data programmed in the project are



Error	Cause	Action
response	format or contents from the controller.	consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

ABB Mint Controller HCP

This communication protocol allows the HMI devices to connect to the ABB motion and servo drive devices using the HCP and HCP2 communication protocols.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

		x
PLC Network	Com	т ОК
Node ID	1	Cancel
Protocol Type	HCP2	•
Timeout (ms)	200	A
Retry count	2	A.
PLC Models		
ABB Mint Controller		

Element	Description
Node ID	Node ID assigned to the controller device.
Protocol Type	Two protocols are available: HCP HCP2
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
Retry count	Number of times a certain message will be sent to the controller before reporting the communication error status.
PLC Models	PLC model you are going to connect to.



Element	Description	
PLC Network	The protocol allows the c connections, check "PLC	onnection of multiple controllers to one HMI device. To set-up multiple network" checkbox and enter the node ID per each slave you need to access.
	Image: PLC Network Node D -1 Protocol Type HCP2 Timeout (ms) 200 Retry count 2 PLC Models Image: PLC Models Slaves Slave Id Model	Image: Comm OK Cancel Image: Cance
Comm	If clicked displays the cor	nmunication parameters setup dialog.
-	Comm Parameter Dialog	
	Port Baudrate Parity Data bits Stop bits Mode	OK com1 9600 even 8 1 RS-485

Element	Description					
	Element	Description				
	Port	Serial port selection.				
		COM1= device PLC port.				
		COM2= computer/printer port.				
	Baudrate, Parity, Data Bits, Stop bits	Serial line parameters.				
	Mode	Serial port mode. Available modes:				
		• RS-232.				
		• RS-485 (2 wires).				
		• RS-422 (4 wires).				

Data types

The ABB Mint Controller HCP driver provides the support for two Memory Types which are referring to the same physical memory area in the Mint controller:

- **Comms**: should only be used with floating point values. The Mint program on the ABB controller should use COMMS to access this data.
- CommsInteger: allows a variety of integer-based data types to be selected.

If the Mind controller program uses	then
COMMS keyword for a tag setup to use the Commsinteger memory type	only the bottom 23 bits will be accurate (due to floating point precision of the COMMS keyword).
COMMSINTEGER keyword for a tag setup to use the Commsinteger memory type	the value is precise for the full 32 bits.

ABB Mint Controller HCP				
Memory Type	Offset	SubIndex		
CommsInteger 🔹	1	0 -		
Data Type	Arraysize	Conversion		
int 👻	0		+/-	
boolean byte short				
int unsignedByte unsignedShort unsignedInt string				
				11-1-

See "Programming concepts" section in the main manual.

Tag Conversion

Conversion to be applied to the tag.

Conversion

inv,swap2 Allowed Configured BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	Conversion			
Cancel	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ - ~ 	Configured Inv bits ABCD->CDAB

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)
Negate	neg : Set the opposite of tag value.
	Example:

Value	Description
	25.36 → - 25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
ABCNOP ->	swap8: Swap bytes in a long word.
OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
	Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	X	Đ	ß	>]	Þ	A 9B	B>	ŧ3	1
Data	1		^		-	Та	g URI			_

Locate the .xml file exported from Tag Editor and click Open.

Organize 👻 Nev	/ folder		BEE	- 🗔 🌘
	▲ N	ame	Date modified	Туре
Desktop	_	Exported Tags.xml	03/06/2016 16:42	XML File
Videos				
User Computer Network				

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags x Protocols				•		
🕂 — 🎽 🔞 💿 >] [> 🗞 🗗 🥻 R 🔎 Search 🛛 🏹 Filter by: Data 🔹 Items used:6/10000 Protocol: Show all 💽 Show all tags 😳 🗍						
Data Type	Comment	^	Property	Value		
Modbus TCP:prot1			✓ Driver			
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)		
Holding Registers 1 unsignedShort Protocol		Protocol	Modbus TCP:prot1			
Holding Registers 2 unsignedShort		✓ Dictionary				
Holding Registers 3 UnsignedShort			Array	false		
MRTU1 unsignedShort			Anay	Idise		
MRTU2 unsignedShort			Array size	0		
MRTU3 unsignedShort			Arrayindex.Subindex	400003		
MRTU4 unsignedShort			Comment			
MRTU5 unsignedShort			Data type	unsignedShort		

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
Ki	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.

Toolbar item	Description	
	Example of both checked a	and unchecked result:
	Data Type Con CoDESS V3 Entry II2 Container Container Model: CoESY V3 Extra V101 Extra V101 11 Diff Extra V101 Extra V101 12 Diff Extra V101 Extra V101 13 Diff Extra V101 Extra V101 Motthgal Diff	Data Data Data Model CODES% 3 Application Applicatio
P → Search Y Filter by: Tag name	Searches tags in the dictio box item selected.	nary basing on filter combo-

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Line Error	An error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits).	Check if the communication parameter settings of the controller is compatible with the device communication setup.
Invalid response	The device did received a response with invalid format or contents from the controller.	Ensure the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

BACnet

The BACnet communication driver has been designed to connect HMI devices to BACnet networks and supports IP and MS/TP communication.

The HMI device operates as a BACnet device.

Implementation details

This implementation of the BACnet communication protocol allows integrating HMIs in a BACnet network and exchange data between HMI and other devices connected to the BACnet network. HMIs provide client capability for displaying properties of BACnet objects in real time using BACnet/IP or MS/TP network types.

BACnet communication protocol can be:

- Configured as BACnet IP: communication with BACnet devices is established over Ethernet using HMI Ethernet port;
- Configured as BACnet MS/SP: communication with BACnet devices is established over serial line, using HMI serial port;

Communication protocol configuration allows defining HMI BACnet ID and object name used to identify HMI in BACnet network.

BACnet object properties are reachable from HMI using explicit Tag configuration. A single Tag represents a single property for a BACnet object.

Using the property Present_Value (85) in Tag configuration, the Tag will be connected to the current value of a specific object (for example in the case of analog values, it will be the measured value).

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

net				
	Comm			
Panel Device ID	262000	Analog Value Count	0	
Object Name	DEV262000	Binary Value Count	0	
Description	НМІ	Multi State Value Count	0	
Media	MS/TP 🕔	, Notification Class Count	0	
Timeout (ms)	5000	IP UDP Port	47808	
Panel Node	1	Local IP		
COV Lifetime (s)	60			
COV Confirmed	<u>.</u>			
Max Master	127			
Max Info Frames	1]		
max MS/TP APDU	480			
max IP APDU	1476			
Time Sync Interval (s)	0			
Time Sync UTC				
PLC Models				
default				
			OK	Can

Element	Description
Panel Device ID	Identifies the HMI device in the network.
Object Name	BACnet Object Name for the HMI device.
Description	HMI device description, for documentation purposes.
Media	Type of communication of the protocol.
	MS/TP: Master-Slave/Token-Passing communication (RS-485).

Element	Description						
	IP: based on standard UDP/IP communication.						
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the BACnet device.						
Panel Node *	MS/TP address. Physical device address on the link; it is not passed through routers.						
COV Lifetime (s)	Desired lifetime of the subscription in seconds before the it shall be automatically cancelled. A value of zero indicates an indefinite lifetime, without automatic cancellation.						
Max Master *	Highest allowable address for master nodes. Must be less than or equal to 127.						
Max Info Frames *	Maximum number of information frames the node may send before it must pass the token. Max Info Frames may have different values on different nodes and may be used to allocate more or less of the available link bandwidth to particular nodes.						
Max MS/TP APDU *	Maximum length of APDU (Application Layer Protocol Data Unit), which means the actual packet length on BACnet network. This value cannot exceed 480 (default value).						
Max IP APDU **	Maximum length of APDU (Application Layer Protocol Data Unit), which means the actual packet length on BACnet network. This value cannot exceed 1476 (default value).						
Time Sync Interval (s)	Represent the interval between every time synchronization, in seconds. If left to 0, time synchronization is disabled.						
Time Sync UTC	Option to synchronize time in UTC format. If disabled, local time format used.						
PLC Models	Reserved for future use.						
Comm *	If clicked displays the communication parameters setup dialog.						
	Comm Parameter Dialog						
	Port com1 Baudrate 9600 Parity none Data bits 8 Stop bits 1 Mode RS-485						

Element	Description						
	Element	Description					
	Port	Communication port.					
	Baudrate, Parity, Data bits, Stop bits	Communication parameters.					
	Mode	Communication mode. Available modes:					
		• RS-232					
		• RS-485					
		• RS-422					
Analog Value Count ***	Number of Analog Value objects to be instanced in BACnet Server. Min: 0 Max: 200						
Binary Value Count ***	Number of Binary Value objects to be instanced in BACnet Server. Min: 0 Max: 200						
Multi State Value Count ***	Number of Multi State Value objects to be instanced in BACnet Server. Min: 0 Max: 200						
Notification Class Count ***	Number of Notifications Class objects to be instanced in BACnet Server. Min: 0 Max: 200						
IP UDP Port **	Port number for IP communication.						
Local IP **	IP Address of the network adapter to use for protocol. Not required if the device has only one Ethernet adapter.						



Note *: Available only if media is set to **MS/TP**.



Note **: Available only if media is set to IP.



Note ***: Check Using BACnet Server chapter.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **BACnet** from the **Driver** list: the tag definition dialog is displayed.

Object Type	Device ID	Data Type	
Analog Value	508	float 👻	
Arraysize Cor	iversion	Object Instance	
Object Property	Array Index	Write Priority	
85	-1	0	
Cov			

.

Elemen t	Description						
Object	Type of BACnet object to be referenced. Available object types:						
Гуре	DeviceAnalog Input						
	Analog Output						
	Analog Value						
	Binary Input						
	Binary Output						
	Binary Value						
	Multi-state Input						
	Multi-state Output						
	Multi-state Value						
	Integer Value						
	Positive Integer Value						
	Large Analog Value						
Device	ID of the device containing the object.						
ID							
Data	Data type for display presentation. Available data types:						
Туре	• boolean						

Elemen t	Description							
	 int unsignedInt float double string binary boolean[] 							
	I hese data types are data types as defined in the software. The equivalence with BACnet data types is shown in the table:							
	BACnet data type Software data type Notes							
	BOOLEAN Boolean -							
	INTEGER	Int	-					
		-						
	REAL	-						
	BIT_STRING	boolean-x	x = size					
	CHARACTER_STRING string-x x = size							
	OCTET_STRING	binary-x	x = size					
	DATE	int or unsignedInt	-					
	TIME	int or unsignedInt	-					
	BACnetObjectIdentifier	int or unsignedInt	Use conversions instance and objType for proper display					
Arraysi ze	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to 							
Conver	If Encoding property is set to requires 2 bytes.	UCS-2BE, UCS-	2LE, UTF-16BE or UTF-16LE one character					
sion								

Elemen t	Description								
	Conversion								
	inv,swap2	Allowed Configured							
		BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK							
	Depending on data type selected, the list Allowed shows one or more conversion								
	value								
	Inv bits	inv : Invert all the bits of the tag.							
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)							
	Negate	neg: Set the opposite of tag value.							
		<i>Example:</i> 25.36 → -25.36							
	AB -> BA	swapnibbles: Swap nibbles in a byte.							
		<i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)							
	ABCD ->	swap2: Swap bytes in a word.							
	CDAB	<i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)							
	ABCDEFGH -	swap4: Swap bytes in a double word.							
	> GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)							
	ABCNOP ->	swap8: Swap bytes in a long word.							
	OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 1000000110							

Elemen t	Description									
	Value	De	scription							
		$\begin{array}{c c} 0001110010111011011001000101101000011100101$								
	BCD	bc (9)	\mathbf{cd} : Separate byte in two nibbles, read them as decimal (from 0 to							
		Exa 23 000 01	Ex <i>ample:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)							
	Select conve	ersion an	n and click +. The selected item will be added to list Configured .							
	If more conve Configured)	ersions a).	ns are configured, they will be applied in order (from top to bottom of list							
	Use the arroy	w button	s to order the	configu	ire	ed conversio	ons.			
Object Instanc e	BACnet ID o	f the obje	ect to be refer	enced.						
Object Propert y	Numeric value of the property to be referenced (example: the value 85 means <i>present-value</i> for most standard objects). The table below specifies all the BACnet Object Properties.									
	Property	Val ue	Property	Val ue		Propert y	Val ue	Property	Val ue	
	accepted- modes	175	effective- period	32		max- info- frames	63	reason-for- halt	100	
	acked- transitions	0	elapsed- active- time	33		max- master	64	recipient- list	102	
	ack- required	1	error-limit	34		max- pres- value	65	records- since- notification	140	
	action	2	event-	35		max-	167	record-	141	

	Ele	en	n
1	t		

Property	Val ue	Property	Val ue	Propert y	Val ue	Property	Val ue
		enable		segment s- accepte d		count	
action- text	3	event- state	36	member- of	159	reliability	103
active-text	4	event- time- stamps	130	minimu m-off- time	66	relinquish- default	104
active-vt- sessions	5	event- type	37	minimu m-on- time	67	required	105
active- cov- subscripti ons	152	event- paramet ers	83	minimu m-output	68	resolution	106
adjust- value	176	exceptio n- schedule	38	minimu m-value	136	scale	187
alarm- value	6	fault- values	39	minimu m-value- timesta mp	150	scale- factor	188
alarm- values	7	feedbac k-value	40	min- pres- value	69	schedule- default	174
all	8	file- access- method	41	mode	160	segmentati on- supported	107
all-writes- successfu I	9	file-size	42	model- name	70	setpoint	108
apdu- segment- timeout	10	file-type	43	modifica tion-date	71	setpoint- reference	109

Ε	le	en	ne
t			

en Description

Property	Val ue	Property	Val ue	Propert y	Val ue	 Property	Val ue
apdu- timeout	11	firmware- revision	44	notificati on-class	17	slave- address- binding	171
applicatio n- software- version	12	high-limit	45	notificati on- threshol d	137	setting	162
archive	13	inactive- text	46	notify- type	72	silenced	163
attempte d- samples	124	in- process	47	number- of- APDU- retries	73	start-time	142
auto- slave- discovery	169	input- reference	181	number- of-states	74	state-text	110
average- value	125	instance- of	48	object- identifier	75	status- flags	111
backup- failure- timeout	153	integral- constant	49	object- list	76	stop-time	143
bias	14	integral- constant- units	50	object- name	77	stop-when- full	144
buffer- size	126	last- notify- record	173	object- property- referenc e	78	system- status	112
change- of-state- count	15	last- restore- time	157	object- type	79	time-delay	113
change- of-state- time	16	life- safety- alarm- values	166	operatio n- expecte d	161	time-of- active- time-reset	114

E	e	ne	
t			

en Description

Property	Val ue	Property	Val ue	Propert y	Val ue	Proper	ty Val ue
client-cov- increment	127	limit- enable	52	optional	80	time-of state- count-r	- 115 eset
configurat ion-files	154	limit- monitorin g-interval	182	out-of- service	81	time- synchro ation- recipier	oniz 116
controlle d- variable- reference	19	list-of- group- members	53	output- units	82	total- record- count	145
controlle d- variable- units	20	list-of- object- property- referenc es	54	polarity	84	trackinų value	g- 164
controlle d- variable- value	21	list-of- session- keys	55	prescale	185	units	117
count	177	local- date	56	present- value	85	update interval	- 118
count- before- change	178	local- time	57	priority	86	update time	- 189
count- change- time	179	location	58	pulse- rate	186	utc-offs	et 119
cov- increment	22	log-buffer	131	priority- array	87	valid- sample	146 s
cov- period	180	log- device- object- property	132	priority- for- writing	88	value- before- change	190
cov- resubscri	128	log- enable	133	process- identifier	89	value-s	et 191

E	Ele	er	n	e
t				

en Description

Property	Val ue	Property	Val ue	Propert y	Val ue	Property	Val ue
ption- interval							
database- revision	155	log- interval	134	profile- name	168	value- change- time	192
date-list	23	logging- object	183	progra m- change	90	variance- value	151
daylight- savings- status	24	logging- record	184	progra m- location	91	vendor- identifier	120
deadband	25	low-limit	59	progra m-state	92	vendor- name	121
derivativ e- constant	26	maintena nce- required	158	proportio nal- constant	93	vt-classes- supported	122
derivativ e- constant- units	27	manipula ted- variable- reference	60	proportio nal- constan t-units	94	weekly- schedule	123
descriptio n	28	manual- slave- address- binding	170	protocol- object- types- supporte d	96	window- interval	147
descriptio n-of-halt	29	maximu m-output	61	protocol- revision	139	window- samples	148
device- address- binding	30	maximu m-value	135	protocol- services- supporte d	97	zone- members	165

Elemen t	Description	Description												
	Property	Val ue	Property	Val ue		Propert y	Val ue		Property	Val ue				
	device- type	31	maximu m-value- timestam p	149		protocol- version	98							
	direct- reading	156	max- apdu- length- accepted	62		read- only	99							
Array Index	Index for sut • -1 me • 0 to n	Index for subscribing elements in BACnet arrays.												
	Priority Arr To read a pri to refer to the The following	ay exan iority arr e priority g figure	nple ay object it is r r item to be rea shows how to	necess ad. read th	ary ie ⁻	/ to set Obje 16th item of	ect Pro p a priorit	pe	rty = 87 and / array.	Array I	ndex has			

Elemen t	Description
	BACnet
	BACnet
	Object Type Device ID Data Type
	Binary Value
	Arraysize Conversion Object Instance
	0 +/- 3000108
	Object Property Array Index Write Priority
	87 16 0
	□ cov
	OK Cancel Apply Help
Write Priority	Write requests priority level. The value is in the range 1-16. 0 is interpreted as 16.
COV	Enable the Change Of Value notification.

Clear/Set Priority

The system offers actions for a more flexible handling of Write Priority.

Action	Description
BACnetClearPriority	Clears the priority array at the position associated to the BACnet tag passed as parameter.
	This action has immediate effect on the BACnet device.
BACnetClearAllPriorities	Clears all positions in the priority array.
	This action has immediate effect on the BACnet device.
BACnetSetPriority	Overrides the Write Priority value configured in the BACnet tag definition.
	This action has two parameters:
	TagName: name of the BACnet tag.
	 TagPriority: new value of Write Priority for the BACnet tag passed as parameter.
	This action only overrides the value of Write Priority in the BACnet tag definition and does not perform any communication with the BACnet device. Any write command that will be performed to the Present Value property of the BACnet device identified by the tag, will be performed using the new Write Priority value.
	The priority value will be valid until:
	 A new call to the BACnetSetPriority action changes it. The HMI device is restarted. The value of WritePriority defined in the project is valid in this case.

Tag Import

BACnet object information can be imported from BACnet EDE (Engineering Data Exchange) files. The EDE file must have the .csv extension.

The importer uses the characters "," and ";" as delimiters. They are considered as reserved characters and you cannot use them in file name.

Use the hierarchical importer to have a ordered list of BACnet objects and properties.

Tags will be created using the string specified in the column object-name of the EDE file. The importer will add the device ID as a prefix to avoid duplication of tag names.



Note: The importer will ask to locate the State-Texts, Unit-Texts and Object-Types files. Click Cancel to ignore.

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	Z	đ	ß	>]	₽	A 9B	B>	ŧ.	1
Data			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importer	s are available for this protocol. Please select the importer type and continue.
Version	Туре
BACnet EDE v1.0	Linear
BACnet EDE v1.0	Hierarchical
Tag Editor exported xml 1.1	. General
Natched dictionary file:	
Keep synchronized	
	OK Cancel

Importer	Description			
BACnet EDE v1.0	Requires a .csv file.			
Linear	All variables will be displayed at the same level.			
BACnet EDE v1.0	Requires a .csv file.			
Hierarchical	All variables will be displayed according to BACnet EDE Hierarchical view.			
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.			
	Tags x			
	+ - ¾ ⓓ ℗ >] ▷ \$B ▷] ⊡ ↓ Data			

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 - 👗 🕲 🖉 🕽 🗞 🖬 🅅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			Y Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			Y Dictionary	
Holding Registers 3 unsignedShort			Dictoriary	61
MRTU1 unsignedShort			Array	taise
-MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
- MRTU4 unsignedShort			Comment	
MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description			
B	Import Tag(s).			
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project			
酸	Update Tag(s).			
	Click on this icon to update the tags in the project, due a new dictionary import.			
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:			
P - Search	Searches tags in the dictionary basing on filter combo-			

For tags referring to BACnet objects of type Calendar or Schedule the tag refresh rate is set to "Manual".

The following BACnet object properties are required for operation of the widgets.

Object	Tags to import			
Calendar	Date_List			
Schedule	Weekly_Schedule			
	Exception_Schedule			
	Default_Value			
	Effective_Period			

DEVICE Object Properties

A BACnet network scanner can detect properties when exploring the network and obtaining data from HMI device.

This are the supported DEVICE object properties:

Property	Description
Object_Identifier	BACnetObjectIdentifier
Object_Name	CharacterString
Object_Type	BACnetObjectType
System_Status	BACnetDeviceStatus
Vendor_Name	CharacterString
Vendor_Identifier	Unsigned16
Model_Name	CharacterString
Firmware_Revision	CharacterString
Application_Software_Version	CharacterString
Protocol_Version	Unsigned
Protocol_Revision	Unsigned
Protocol_Services_Supported	BACnetServicesSupported
Protocol_Object_Types_Supported	BACnetObjectTypesSupported
Object_List	BACnetARRAY[N]of BACnetObjectIdentifier
Max_APDU_Length_Accepted	Unsigned
Segmentation_Supported	BACnetSegmentation
APDU_Timeout	Unsigned
Number_Of_APDU_Retries	Unsigned
Device_Address_Binding	List of BACnetAddressBinding
Database_Revision	Unsigned

BACnet Alarm Events

The special "protAlarm:BACN" trigger mode, available from the Alarms Editor, give the possibility to receive alarm events from the BACnet native alarms module.

Trigger			Х		
Triggers	Properties				
limitAlarm	protAlarm	protAlarm			
deviationAlarm	deviceID	4016			
valueAlarm protAlarm:BACN[prot1]	notificationClassID	3000015			
	activeMonday	true			
	activeTuesday	true			
	activeWednesday	true			
	activeThursday	truo	Υ.		
)K Cano	el		

Property	Description			
deviceID	Identifies the BACnet device in the network.			
notificationClassID	Notification Class ID to subscribe for the alarm events retrieving			
processID	Not used			
activeMonday activeTuesday activeWednesday activeThrusday activeFriday activeSaturday activeSunday	 Define in which days keep active the alarm events subscription False Subscription not active True Subscription active 			
startHour startMinute startSecond endHour endMinute endSecond	Define the time window where the alarm events subscription will be active			

The alarm widgets will report the alarm information that are provided from the BACnet device.

Select	Name	State	Value	Time	Description
	SISMI3NCE/Programming.4016.SUMMER-SP-SUPPLY:toOffNormal	Triggered Not Acked	90	13/02/2017 04:09:42	SUMMER ALARM
	SISMI3NCE/Programming.4016.WINTER-SP-SUPPLY:toOffNormal	Triggered Not Acked	5	13/02/2017 04:10:06	WINTER ALARM
•					•
ſ	Check/Uncheck All Eilter · Hide Not Tringered		•	Ack	eset
	Filler : Hide Not Inggered		•		



When the special "protAlarm:BACN" trigger mode is used, the widget of the active alarms show the timestamp provided from the BACnet device while the widget of the historical alarms show the timestamp of when the alarm events are received from the HMI device. Generally, both timestamps are the same but if you need to show the timestamp from the BACnet device even inside the widget of the historical alarms you can add a new column configured to use the "allSourceTimestamp" value from the alarm history widget.

field1.value						
Source:	🔿 Tag	O Alias	○ System	• Widget	O Recipe	
₽- Search						
Name						
▷ _AlarmsM	1gr					
▷ _EventMe	gr					
MultiLan	lgMgr					
AlrmHstry	/					
▲ Alrm	Hstry.evn	twgt.evntl	bfrwgt			
-a	alAlarmID					
	alBGColor					
	alCustomField1					
	alCustomF	ield2				
	alDateTime					
	alDescripti	on				
a	lEvently	pe				
a	GCOIO					
	Name					
	SourceTi	mestamo				
	IState	mestamp				
	alValue					
-	vDate					
-e	vTime					



BACnet alarm is a special alarm that require a double space to be stored inside the events buffer. This means, for example, if the events buffer is configured to contain 1.000 events only the last 500 BACnet events will be stored.

BACnet Trend Buffer

To use a BACnet trend object as a trend buffer:

- 1. Open the Trends Editor
- 2. Click the "Add PLC Trend" button (This button is enabled only when at least one BACnet protocol is configured)
- 3. Configure the below parameters to identify the BACnet trend object to use.

• 💽 — 🐵 🗈 🛛	> >] [Total memory Space		0%	
Trend1	Active BAC	N:prot2 ∨	>]	
Property				Value
 Location 				
Device ID				0
Object ID				0
Bit Index				-1
✓ Data				
Data type				float
✓ Time				
Time Specification				local

Property	Description			
Device ID	Identifies the BACnet device in the network.			
Object ID	BACnet ID of the trend object to be referenced.			
Bit Index	When the data type is boolean, it is the index to select the bit to use inside the BACnet bit_string. It is not used with the other data types.			
Data type	Specify the type of data of the BACnet trend object. The supported data types are: boolean int unsignedInt float			
Time Specification	Time format used inside the selected BACnet trend object local global (UTC) 			

The trend buffer thus configured can then be used inside any trend widgets.

BACnet Calendar Widget

Use Calendar widget to display content of a BACnet Calendar object.

Property	Description					
Date_List	Connect to the "Date_List" tag of a BACnet calendar object in ReadOnly or Read/Write.					
	Note: it can be connected to an alias which indexes a list of BACnet calendar Date_List(s), in order to use one calendar widget for more than one calendar object.					

Operation of Calendar Widget

The widget shows data for one month.

	MON	TUE	WED	THU	FRI	SAT	SUN
52	26	27	28	29	30	31	1
1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22
4	23	24	25	26	27	28	29
5	30	31	1	2	3	4	5
<	01/2	017	>	Nev	v Cle	ar All	efresh

Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

Swing gesture can be used on the widget to select the date.

New

Press the button "New" to enter a new calendar item. The button is active only if the tag associated to the calendar has been configured as Read/Write.

Calendar item	Description
Single	Click on a day to select a single day into the calendar
	Select a day
	MON TUE WED THJ FRU SAT SUN 52 28 28 28 30 31 1
	1 2 3 4 5 6 7 8 2 9 10 11 12 13 14 15
	3 16 17 18 19 20 21 22 4 23 24 25 26 27 28 29
	30 31 2 3 4
	< 01/2017 >> Cancel Prev DK
Range	Click on the first day and on the last day to select a range of days into the calendar.
	Single click on a day to change previous selected last day of the range.
	 Double click on a day to change previous selected first selected day of the range.
	Select a date range
	MON TUE WED THU FRI SAT SUN
	1 2 3 4 5 6 7 8
	2 9 10 11 12 13 14 15
	3 10 17 18 19 20 21 22 4 23 24 25 26 27 28 29
	7 30 31 3 4 3
	Prev 01/2017 >> Cancel Prev OK

Calendar item	Dese	Description						
MWD	Sele	ct a [Day c	or a W	/eek	for e	ach y	/ear
	Sel	lect a M	WD					
		MON	TUE	WED	тни	FR1	SAT	SUN
	1-7	1						
	8-14							
	15-21							
	22-28							
	29-31							
	last 7 d.	i i i						

Clear All

Press the button "Clear All" to clear the content of the calendar object. The button is active only if the tag associated to the calendar has been configured as Read/Write. The button is configured to react to an onMouseHold event, to reduce risk of data loss.

Refresh

Press the "Refresh" button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the calendar.

BACnet Schedule Widget

Use Schedule widget to display content of BACnet Schedule object.

Property	Description						
Туре	Select the type of BACnet object controlled by the schedule.						
	Options are:						
	• Binary						
	• Real						
	Multistate						
Weekly_Schedule	Attach to the Weekly_Schedule tag of the schedule object. The tag can be Read Only or Read/Write.						
Exception_Schedule	Optionally attach to the Exception_Schedule tag of the schedule object. The tag can be Read Only or Read/Write. Only attach this property if exceptions are used.						
Default_Value	Optionally attach to the Default_Value tag of the schedule object. The tag can be Read Only or Read/Write. Only attach this property if default values are used.						
Cal. 0 (Date_List)	Optionally attach to the Date_List tag of the schedule widget in Read Only mode. Use this options to show the "calendar reference" exceptions.						
	Note: An exception can be a single date, a date range, a mwd or a calendar reference. In this last case, exception_list does not contain						

Property	Description							
	 the date information, but only time-value-priority and a reference to the calendar. The date_list needed to show the scheduling into the widget is stored into the relative BACNCalendar, and this is why we need this datalink. If there is no need to show calendar exceptions in the schedule, this property can be left void. Note: If it is not attached to a calendar, it is not possible to insert calendar exception. See BACNSchedKeypad for details. 							
Cal. 0 (Object_Name)	Optionally attach to the property of the calendar. This name is used to identify the calendar in the BACNSchedKeypad used to insert calendar exceptions. If Object_Name is not attached, the calendar is identified with its instance number. This property is used only if a Cal. 0 (Date_List) is attached to a calendar.							
Cal. 1 (Date_List)	Option for a second calendar.							
Cal. 1 (Object_Name)	Option for a second calendar.							
Value-color-text Map	Defines the association value – Color/Text shown in the schedule. Use this option to define all possible values available in the BACNSched keypad. Value-Color Dialog ? × + -							
	Tag value Mapped color Text							
	1 1 #00aatt Saving							
	3 3 #55ff7f Normal							
	Ok Cancel							

Operation of Schedule Widget

The widget shows data for one week.
Default	t Value	e: Norn	nal		ew Cl	ear All	Refresh
	MON	TUE	WED	THU	FRI	SAT	SUN
00:00							
04:00		E, 04:00 Normal					
08:00						E, 08:00 Confort	
12:00		E, 12:00 Confort					
16:00							
20:00		E, 20:00 Saving				E, 20:00 Saving	
<	16/01	/2017	- 22/01	/2017	>		

Use the < and > buttons to select the week to be displayed. The date of first day and last day of the week is shown.

Swing gesture can be used on the widget to select the date.

New

Press the button "New" to enter a new schedule item. The button is active only if the tag associated to Weekly Schedule or Exception Schedule has been configured as Read/Write.

Schedule item	Description						
Weekly	Then select the desired value and the time when it should be set. Press OK to confirm the new item.						
Exception Single	Click on a day to select a single day into the calendar. On the next dialog select the time window, the desired value and its priority.						
Exception Range	Click on the first day and on the last day to select a range of days into the calendar.						

Schedule item	Description
	 Single click on a day to change previous selected last day of the range. Double click on a day to change previous selected first selected day of the range. On the next dialog select the time window, the desired value and its priority.
Exception MWD	Select a Day or a Week for each year or each month. On the next dialog select the time window, the desired value and its priority.
Exception Cal Ref	This option is available only if scheduler is linked to a calendar (configured as Read/Write) Select the time window, the desired value and its priority. Value will set on all days defined from the calendar. If there are more calendars associated with Scheduler widget, select the calendar to use.

Clear All

Press the button "Clear All" to clear the content of the schedule object. The button is active only if the tag associated to the calendar has been configured as Read/Write. The button is configured to react to onMouseClick and onMouseHold events. The onMouseHold event will clear all data in the schedule. The onMouseClick event will recall a dialog box for selection of data to clear. It is needed to choice to clear weekly data or exception data.

Mor	nday	+			
1	Clear Weekly	1	1	Clear All Exc.	1

Refresh

Press the "Refresh" button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the schedule.

BACnet Effective Period Widget

Use the Effective Period widget to feed information to the Effective_Period tag of a Schedule object, if this is requested.

Property	Description
BACnet Effective_ Period	Attach to the Effective_Period tag of the Schedule object

01/10/2017 - 01/13/2017 Refresh

Operation of Effective Period Widget

The widget shows starting date and end date for the period.

Click on the area showing the dates to activate the data entry procedure showing the keypad BACNDateRange.

Ah	ways	ys All month		nth	1	All year	
	MON	TUE	WED	THU	FRI	SAT	SUN
52	26	27	28	29	30	31	1
1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22
4	23	24	25	26	27	28	29
5	30	31	1	2	3	4	5

The keypad shows data for one month.

Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

You may use the swing gesture on the widget to select the date.

Select the period clicking of first day and last day of the period. The Effective_Period is show with a different color.

The keypad offers three predefined options:

Option	Description
Always	The schedule will be always active.
	//**** - **/**/**** Refresh
All Month	The selected period will be extended to all months.
	**/03/2017 - **/12/2017 Refresh
All Year	The selected period will be extended to all years.
	01/03/**** - 01/12/**** Refresh

Refresh

Press the "Refresh" button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the widget.

BACnet Keypads

Keypad	Description
BACNCal	Keypad for BACnet Calendar.
BACNDateRange	Keypad for BACnet Effective_Period.
BACNDefVal	Keypad for default value (embedded in the BACnet Schedule).
BACNSched	Keypad for BACnet Schedule.
	This keypad is context sensitive. It will show different options depending on the type of schedule.

BACnet widgets require dedicated keypads for data entry.

The system is configured to recall the appropriate keypad for each BACnet widget.

Using BACnet Server

BACnet protocol is capable to act as BACnet Server, by exposing BACnet objects.

To properly setup BACnet Server, it is needed to execute the following steps:

1. Configure objects to expose from **Protocol Editor Settings.**

	Co	omm]		
Panel Device ID	262000		Analog Value Count	12	*
Object Name	DEV262000		Binary Value Count	11	×
Description	НМІ		Multi State Value Count	18	* *
Media	IP	-	Notification Class Count	5	(A)
Timeout (ms)	5000	ا	IP UDP Port	47808	A
Panel Node	1		Local IP		
COV Lifetime (s)	60	_			
COV Confirmed					
Max Master	127				
Max Info Frames	1				
max MS/TP APDU	480	- A-			
max IP APDU	1476				
Time Sync Interval (s)	0				
Time Sync UTC					
PLC Models					
AT WE WITH T					



Note: Objects configured in above image can be discovered by BACnet clients:

File	Functions	Options	Help							
0 🕽	K									
Devices					Subscription	is. Periodic P	ollina. Even	nts/Alarms		
🛃 Dev	vices			~					-	
÷	Udp:47808				Device	ObjectId	Name	Value	Time	Stat
	· 💂 DEV26200	0 [262000]		~						
Address	Space									
B	DEV262000			~						
8	ANALOG VALU	JE 0								
×	ANALOG_VAL	UE:1								
<u></u>	ANALOG_VAL	UE:2								
<u></u>	ANALOG_VAL	UE:3								
- <u>×</u>	ANALOG_VAL	UE:4								
8	ANALOG_VAL	UE:5								
- <u>×</u>	ANALOG_VAL	UE:6								
<u>×</u>	ANALOG_VAL	UE:7								
~	ANALOG_VAL	UE:8								
<u>×</u>	ANALOG_VAL	UE:9								
2	ANALOG_VAL	UE:10								
<u> </u>	ANALOG_VAL	UE:11								
	BINARY_VALU	JE:0								
	BINARY_VALU	JE:1								
	BINARY_VALU	JE:Z								
	BINARY_VALU	JE:3								
	DINART_VALU	JE:4 IE:5								
		JE.O								
	BINARY VALU	JE.0 IE·7								
	BINARY VALU	JE.7 JE-8								
	BINARY VALL	JE-9								
- ă	BINARY VALU	JE-10								
	NOTIFICATION	V CLASS:0								
	NOTIFICATION	V CLASS:1								
	NOTIFICATION	V CLASS:2								
	NOTIFICATION	V_CLASS:3								
	NOTIFICATION	V_CLASS:4								
	MULTI_STATE	E_VALUE:0								
	MULTI_STATE	E_VALUE:1								
	MULTI_STATE	E_VALUE:2								
	MULTI_STATE	E_VALUE:3								
	MULTI_STATE	E_VALUE:4								
	MULTI_STATE	E_VALUE:5								
	MULTI_STATE	E_VALUE:6		~						

2. Create Tags that points to local BACnet objects, setting Device ID as the Device ID configured in Protocol Editor Settings:

Object Type	Device ID	Data Type	
Analog Value	262000	float 👻	
Arraysize Co	iversion	Object Instance	
0	<u>+</u> /	0	
Dbject Property	Array Index	Write Priority	
85	-1	0	*
cov			

Device objects description

Property Name	Code	Default value	Permanent	Note	Data Type
APDU timeout	11	Parameter	Yes		UnsignedInt
Application software version	12		Read-only		String
Database version	155		Read-only		UnsignedInt
Daylight saving status	24		Read-only		Boolean
Read-only	28	Parameter	Yes		String
Device address binding	30		Read-only		String
Firmware revision	44		Read-only		String
Local date	56		Read-only		UnsignedInt
Local time	57		Read-only		UnsignedInt

Property Name	Code	Default value	Permanent	Note	Data Type
Location	58	Parameter	Yes		String
Max APDU length accepted	62		Read-only		UnsignedInt
Max info frames	63	Parameter	Yes	Only if MSTP	String
Max master	64	Parameter	Yes	Only if MSTP	String
Model name	70		Read-only		String
Number of APDU retries	73	Parameter	Yes		UnsignedInt
Object identifier	75	Parameter	Yes		UnsignedInt + Conversion
Object list	76		Read-only		UnsignedInt + Conversion
Object name	77	Parameter	Yes		String
Object type	79		Read-only		UnsignedInt
Protocol object types supported	96		Read-only		Boolean(51)
Protocol revision	139		Read-only		UnsignedInt
Protocol services supported	97		Read-only		Boolean(40)
Protocol version	98		Read-only		UnsignedInt
Segmentation supported	107		Read-only		UnsignedInt
System status	112		Read-only		UnsignedInt
UTC offset	119		Read-only		Int
Vendor identifier	120		Read-only		UnsignedInt
Vendor name	121		Read-only		String

Analog Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolen(3)
COV increment	22	0	Yes		Float
Deadband	25	0	Yes		Float
Description	28	"ANALOG	Yes		String



Property Name	Code	Default value	Permanent	Note	Data Type
		VALUE n"			
Event enable	35	0	Yes		Boolean(3)
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
High limit	45	0	Yes		Float
Limit enable	52	0	Yes		Boolean(2)
Low limit	59	0	Yes		Float
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	2:n	Read-only		UnsignedInt + Conversion
Object name	77	"ANALOG VALUE n"	Yes		String
Object type	79	2	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Present value	85	0			Float
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		Float
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt
Units	117	98	Yes		Units

Binary Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolean(3)
Active text	4		Yes		String
Alarm value	6	0	Yes		Boolean
Description	28	"BINARY VALUE n"	Yes		String

Property Name	Code	Default value	Permanent	Note	Data Type
Event enable	35	0	Yes		Boolean(3)
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
Inactive text	46		Yes		String
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	5:n	Read-only		UnsignedInt + Conversion
Object name	77	"BINARY VALUE n"	Yes		String
Object type	79	5	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Polarity	84	0	Yes		UnsignedInt
Present value	85	0			Boolean
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		Boolean
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt

Multi State Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolean(3)
Alarm values	7		Yes	Defines number of array elements	UnsignedInt
				Array of alarm values (0:n)	UnsignedInt(n)
Description	28	"MULTI STATE VALUE n"	Yes		String
Event enable	35	0	Yes		Boolean(3)

EXOR

Property Name	Code	Default value	Permanent	Note	Data Type
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
Fault values	39		Yes	Defines number of array elements	UnsignedInt
				Array of fault values (0:n)	UnsignedInt(n)
Number of states	74	1	Yes		UnsignedInt
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	19:n	Read-only		UnsignedInt + Conversion
Object name	77	"MULTI STATE VALUE n"	Yes		String
Object type	79	19	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Present value	85	0			UnsignedInt
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		UnsignedInt
State text	110		Yes		UnsignedInt
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt

Notification Class objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Ack required	1	0	Yes		Boolean(3)
Description	38	"NOTIFICATION CLASS n"	Yes		String
Notification class	17	4194303	Yes		UnsignedInt
Object identifier	75	15:n	Read-only		UnsignedInt +

Property Name	Code	Default value	Permanent	Note	Data Type
					Conversion
Object name	77	"NOTIFICATION CLASS n"	Yes		String
Object type	79	15	Read-only		UnsignedInt
Priority	86	255,255,255	Yes		UnsignedInt(3)
Recipient list	102		Yes		UnsignedInt(n)

Example of usage

Once BACnet Server Tags are configured, they can be used in combination with Data Transfer feature.

Example: Modbus TCP/RTU Tags can be transferred to BACnet Tags (with same data type). In this way, all BACnet clients can reach BACnet Server and see actual value of Modbus Tags, using BACnet Tags as interface.



JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

The following commands are supported:



Command	Description
scan (minID, maxID, <timeout>)</timeout>	Executes a scan for devices in the given range.
scan_status	Get the scanning result.
devices	Get the list of devices.
objectCount (deviceID, objectType)	Get the object count of given object types in given device.
objectNames (start, count)	Get the part of object names asked by previous objectCount.
properties (deviceID, objectType, objectInstance)	Get the properties of given device/object.

scan

Scan the bus to find all present devices having ID in the range minID – maxID.

To scan the whole network use 0 and 999999 ad minID and maxID.

The optional timeout can be indicated in milliseconds. Default value is 2000 ms.

The function starts the scan operation; the function scan_status can be used to know the status of the operation. The result of the operation is "**scanning**".

scan_status

Get the status of last started scan operation. It returns **"scanning**" or **"finished**". Scan operation finishes when the timeout time is expired

devices

Get the list of devices found by latest scan operation. The result is a JSON string containing of each device:

- device name
- model name
- vendor name
- vendor ID

Example:

```
{"minID":0,"maxID":999999,"devices":[262000,1101],"deviceNames":
["DEV262000","S01101"],"modelNames":["HMI model","EY-AS525F001"],"vendorNames":
["Company Name","SAUTER"],"vendorIDs":[262,80]}
```

objects

Get the list of all objects from the devices having the given ID. The list is returned as a JSON string containing for each object

- type
- instance number

type can be:

OBJECT_ANALOG_INPUT = 0,

OBJECT_ANALOG_OUTPUT = 1,

OBJECT_ANALOG_VALUE = 2,

OBJECT_BINARY_INPUT = 3,

OBJECT_BINARY_OUTPUT = 4,

OBJECT_BINARY_VALUE = 5,

 $OBJECT_CALENDAR = 6,$

OBJECT_COMMAND = 7,

OBJECT_DEVICE = 8,

OBJECT_EVENT_ENROLLMENT = 9,

OBJECT_FILE = 10,

 $OBJECT_GROUP = 11,$

OBJECT_LOOP = 12,

OBJECT_MULTI_STATE_INPUT = 13,

OBJECT_MULTI_STATE_OUTPUT = 14,

OBJECT_NOTIFICATION_CLASS = 15,

OBJECT_PROGRAM = 16,

OBJECT_SCHEDULE = 17,

OBJECT_AVERAGING = 18,

OBJECT_MULTI_STATE_VALUE = 19,

OBJECT_TRENDLOG = 20,

OBJECT_LIFE_SAFETY_POINT = 21,

OBJECT_LIFE_SAFETY_ZONE = 22,

OBJECT_ACCUMULATOR = 23,

OBJECT_PULSE_CONVERTER = 24,

OBJECT_EVENT_LOG = 25,

OBJECT_GLOBAL_GROUP = 26,

OBJECT_TREND_LOG_MULTIPLE = 27,

OBJECT_LOAD_CONTROL = 28,

OBJECT_STRUCTURED_VIEW = 29,

OBJECT_ACCESS_DOOR = 30,

OBJECT_TIMER = 31,

OBJECT_ACCESS_CREDENTIAL = 32,

OBJECT_ACCESS_POINT = 33,

OBJECT_ACCESS_RIGHTS = 34,

OBJECT_ACCESS_USER = 35,

OBJECT_ACCESS_ZONE = 36,

OBJECT_CREDENTIAL_DATA_INPUT = 37,

OBJECT_NETWORK_SECURITY = 38,

OBJECT_BITSTRING_VALUE = 39,

OBJECT_CHARACTERSTRING_VALUE = 40,

OBJECT_DATE_PATTERN_VALUE = 41,

OBJECT_DATE_VALUE = 42,

OBJECT_DATETIME_PATTERN_VALUE = 43,

OBJECT_DATETIME_VALUE = 44,

OBJECT_INTEGER_VALUE = 45,

OBJECT_LARGE_ANALOG_VALUE = 46,

OBJECT_OCTETSTRING_VALUE = 47,

OBJECT_POSITIVE_INTEGER_VALUE = 48,

OBJECT_TIME_PATTERN_VALUE = 49,

OBJECT_TIME_VALUE = 50,

OBJECT_NOTIFICATION_FORWARDER = 51,

OBJECT_ALERT_ENROLLMENT = 52,

OBJECT_CHANNEL = 53,

OBJECT_LIGHTING_OUTPUT = 54,

OBJECT_BINARY_LIGHTING_OUTPUT = 55,

OBJECT_NETWORK_PORT = 56,

Other types are manufacturer specific.

objectCount

Returns the number of objects of a defined type in the device having the indicated ID. If specified type is -1 the command will return the number of all objects.

```
Example:
objectCount 1101 -1
77
objectCount 1101 0
1
objectCount 1101 1
1
objectCount 1101 3
2
```

objectCount 1101 29 16

objectNames

Returns a part of the objects listed by a previous **objectCount** command, from start index. The list contains only counted objects according to filter previously used

The list is returned as a JSON string containing for each object

- type
- instance number
- name

Example:

```
{"deviceID":1101,"objects":[{"type":29,"instance":0,"name":"0x7400000"},
{"type":29,"instance":16,"name":"0x7400010"},
{"type":29,"instance":18,"name":"0x7400012"},
{"type":29,"instance":19,"name":"0x7400013"},
{"type":29,"instance":20,"name":"0x7400014"},
{"type":29,"instance":21,"name":"0x7400015"},
{"type":29,"instance":22,"name":"0x7400016"},
{"type":29,"instance":23,"name":"0x7400017"},
{"type":29,"instance":24,"name":"0x7400018"},
{"type":29,"instance":25,"name":"0x7400019"},
{"type":29,"instance":26,"name":"0x740001a"},
{"type":29,"instance":27,"name":"0x740001b"},
{"type":29,"instance":28,"name":"0x740001c"},
{"type":29,"instance":29,"name":"0x740001d"},
{"type":29,"instance":30,"name":"0x740001e"},
{"type":29,"instance":31,"name":"0x740001f"}]}
```

properties

Returns the list of properties available for object with given type and instance number in device having the given ID.

The list is returned as a JSON string containing for each object

- deviceID
- object type
- object instance
- list of available properties

Example:

```
{"deviceID":1101,"objectType":2,"objectInstance":1,
"properties":
[22,28,36,65,69,75,77,79,81,85,87,103,104,111,117,168,8309,8314,8332,8333]}
```

Example of usage:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var params = String(fromId) + " " + String(toId) + " " + String
(timeout); // fromID and toID are min and max IDs
var json_str = tagMgr.invokeProtocolCommand(protID , "scan", params, state); //json_
str containts JSON string with scanned devices.
```

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Error	Cause
Cannot bind to the device_id	Cannot establish communication with the Device ID provided for this tag.
Cannot read the property data type	The type of the property to write cannot be determined.
write conversion error	A conversion associated to this tag has failed.
Cannot write ICOM type BACnet type	A datatype selected for this tag is not compatible with the BACnet property to set.
Timeout on COV subscription	A request for COV subscription for this tag has timed out.
Timeout on waiting COV update	A COV notification has not been received for this tag whithin timeout.
Can't get COV for this property	The selected property for COV notification is unsupported.
datagramItem conversion error	A conversion associated to a tag that is part of a datagram has failed.
Timeout waiting on response	No response for a request of read or write property within timeout.
datagram element, no data available	No data available for a tag that is part of datagram.
datagram element, Unsupported BACnet data type	Read datagram element is of unsupported BACnet type.
datagram element, can't convert BACnet type to	A Data Type selected for a tag which is part of a datagram is not compatible with the BACnet property to read.

Codes supported by this communication driver:

Error	Cause
No data in response	No data available for a tag.
Datagram element 'element_ URI' error: 'error_class': error_code	The reading of indicated datagram element 'element_URI' was reported as error. The error descriptions error_class and error_code are included in the message.
datagram object does not match	The object of the received datagram item does not match the asked object.
datagram property does not match	The property of the received datagram item does not match the asked property.
BACnet abort: reason_of abort	BACnet abort message was received. The reason of abort is given.
BACnet reject: reason_of_ rejection	BACnet reject message was received. The reason of rejection is given.
BACnet error: error_class: error_code	BACnet error message was received. The error description is given as combination of error_class and error_code .
parameter 'parameter_name' out of range	The protocol parameter parameter_name value is out of range.

Beckhoff ADS

Beckhoff ADS protocol driver is used for communication with Beckhoff controllers through Ethernet connection. This implementation of Beckhoff ADS protocol driver is based on the information published by Beckhoff.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol "Beckhoff ADS" from the list of available protocols.

Alias Car Panel AMS Net ID 0.0.0.0.0 Panel AMS Port 32976 Target IP Address 0 . 0 . 0 . 0 Target TCP Port 48898	ncel
Panel AMS Net ID0.0.0.0.0Panel AMS Port32976Target IP Address0 . 0 . 0 . 0Target TCP Port48898	
Panel AMS Port 32976 Target IP Address 0 . 0 . 0 . 0 Target TCP Port 48898	
Target IP Address 0 0 0 0 Target TCP Port 48898	
Target TCP Port 48898	
Target AMS Net ID 0.0.0.0.0	
Target AMS Port 801	
Timeout 3000	
PLC Models	
BC/BX	

Element	Description
Alias	Name to be used to identify nodes in the plc network configuration. The name will be added as a prefix to each tag name imported for each network node.
Panel AMS Net ID	Specifies the AMS net ID of the panel; the first 4 bytes must match the panel IP address assigned to the HMI device. If panel has IP address 192.168.10.100 then AMS Net ID could be 192.168.10.100.1.1
Panel	Specifies the panel AMS port number to be used on panel.

Element	Description								
AMS Port	Using TwinCAT2, default Panel AMS Port is 32976.								
	Using TwinCAT3, default Panel AMS Port is 32844.								
Target IP Address	Specifies the IP address of the target controller.								
Target AMS Net ID	Specifies the Target AMS net ID of the target controller.								
Target	Specifies the port number dedicated to the communication on target device.								
ANIS Port	Using TwinCAT2, default Target AMS Port is 801.								
	Using TwinCAT3, default Target AMS Port is 851.								
Timeout	The number of milliseconds between retries when communication fails.								
PLC models	Select the model which corresponds to the device to be connected. Model selection is very important to be set properly.								
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter the Target Controller settings for every node.								

TwinCAT2 Route Settings

Beckhoff controllers require some specific settings to allow connection from HMI devices.

In TwinCAT2 System Manager you need to configure Static Route.

First of all the system must be reset in Configuration Mode using the toolbar button as showed in the following figure.

File Edit Actions View Options	; Help	•
] 🗅 🚅 📽 🖬 🎒 💁 🗼		Q 🚺
SYSTEM - Configuration	Current Routes Static Routes Project Routes Netle Management	
Route Settings	Route AmsNetId Address Type	Commer
PLC - Configuration		- 🥄
I/O - Configuration		- 2
	and the second s	1

Then confirm to Restart TwinCAT2 System in Config Mode as in the figure below.

Restart Twinc	AT System in 1	Config Mode
		-
ОК	Cancel	
	Restart TwinC	Restart TwinCAT System in OC

Once restarted, as in the next figure, follow these steps to add a new Route:

- 1. Open Route Settings.
- 2. Select Static Routes tab.
- 3. Click on [Add] button.

i <mark>le Edit Actions</mark> View Options	Help
	14 16 16 🙀 8 🔍 📾 🗸 🌌 👧 🌺 🔧 🔨
SYSTEM - Configuration	Current Routes Static Routes Project Routes NetId Manager
Route Settings	Route AmsNetId Address
PLC - Configuration	
I/O Devices	
	Add
1220	
	and the second second

Into Add Route Dialog user must set:

1. Route Name: a name useful to indentify the Route i.e. "HMI", AmsNetId: The Panel AMS Net ID as configured into Beckhoff ADS protocol, Transport Type: TCP/IP.

Address Info: Type in the Panel IP Address with "IP Address" option selected.

- 2. Target Route: Static.
- 3. Click on [Add Route] button. Note: no warning or message will be shown.
- 4. Click on [Close] button.

Add Route Dialog				×
Enter Host Name / IP:]	Refresh Status	Broadcast Search
Host Name	Connected Address	AMS NetId	TwinCAT OS Ve	ersion Comment
< 1				>
Route Name (Target):	JMobile HMI		Route Name (Remote):	TS-VM-XP2
AmsNetId:	192.168.10.100.1.1	9	Target Route	Remote Route
Transport Type:	TCP/IP			⊙ None
Address Info:	192.168.10.100	1 1	Temporary	C Temporary
🔾 🔿 Host Name 🛛 💿 IP	Address			
Connection Timeout (s):	5	3	Add Route	4 Close

Then the route will appear under Static Routes list.

TwinCAT3 Route Settings

Beckhoff controllers require some specific settings to allow connection from HMI devices. In TwinCAT3 XAE you need to configure a Static Route.

First of all TwinCAT3 system must be reset in Configuration Mode using the toolbar button as showed in the following figure.



Then confirm to Restart TwinCAT3 System in Config Mode.

TwinCAT XAE
Restart TwinCAT System in Config Mode
OK Cancel

Once restarted, as in the next figure, follow these steps to add a new Route:

- 1. Open Routes.
- 2. Select Static Routes tab.
- 3. Click on [Add] button.

File	Edit	View	Project	Build	Debug	TwinCAT	PLC	Tools • 🖳	Window	Help	- Twin	nCAT3	32 👻 🖉 🚆
🦉 Server Explorer 🕉 Toolbox	TwinC/	AT Proje ent Route Route Add		ic Route Ams Rem	es Proje	ct Routes	NetId ress	Manage	Type	P	Comment		Solution Explorer Solution 'TwinCAT Project1' (1 project) TwinCAT Project1 TwinCAT Project1 Solution 'TwinCAT Project1' (1 project) TwinCAT Project1 Solution 'TwinCAT Project1' (1 project) TwinCAT Project1' (1 project) Solution 'TwinCAT Project1' (1 project) Solution 'TwinC

Into Add Route Dialog user must set:

- Route Name: a name useful to indentify the Route i.e. "HMI", AmsNetId: The Panel AMS Net ID as configured into Beckhoff ADS protocol, Transport Type: TCP/IP. Address Info: Type in the Panel IP Address with "IP Address" option selected.
- 2. Target Route: Static.
- 3. Click on [Add Route] button. Note: no warning or message will be shown.
- 4. Click on [Close] button.

Add Route Dialog				×
Enter Host Name / IP:			Refresh Status	Broadcast Search
Host Name	Connected Address	AMS NetId	TwinCAT OS Ve	ersion Comment
<	111			
Route Name (Target):	JMobile HMI		Route Name (Remote):	TS-VM-XP2
AmsNetId:	192.168.10.100.1.1	4	Target Route	Remote Route
Transport Type:	TCP/IP 🔽			None Challer
Address Info:	192.168.10.100		Temporary	O Temporary
🔾 🔿 Host Name 🛛 💿 IP	Address			
Connection Timeout (s):	5	3	Add Route	4 Close

Then the route will appear under Static Routes list.

Tag Import

Exporting Tags from PLC

The data in the Beckhoff system is based on tags.

The organization of the internal memory of the controller is not fixed but it is configured by the user at development time. Each data item can be identified by a string called "tag".

The TwinCAT development environment generates the list of tags created for each controller in the configuration of the application.

The project in the panel must refer to the tag names assigned in the TwinCAT PLC Control programming software at development time. The Designer Tag Editor supports direct import of the tag file generated by the Beckhoff software.

<u>File Edit Project Insert Extras Online Window H</u> elp	_ 8 ×
<u>``</u> ₽ <u>₽</u> ₽ <u>₽</u> <u></u>	
Image: Pous 0001 PROGRAM MAIN Image: Pous 0002 VAR Image: Pous 0003 open_the_door: BOOL; 0004 check_list:DWORD; 0005 double_check:INT; 0006 Temperature:REAL; 0007 0008 0008 END_VAR	
	>
0001 0002 IF open_the_door THEN	^
0003 check_list:=5; 0004 double_check:= double_check + 1; 0005END_IF;	
0006	~
	>
Size of used data: 49 of 1048576 bytes (0.00%) Size of used retain data: 0 of 32768 bytes (0.00%)	^
U Error(s), U Warning(s).	~
	>
Target: Local (192.168.1.239.1.1), Run Ti	me: 1 TwinCAT Confi

To export tags defined for the selected controller, click on Project > Build as shown.

File	Edit	Project	Insert	Extras	Online	Window	Help	- 8 ×
1		Build				Ctrl+F8		
		Rebui	ild all					
PO 🔁)Us	Load	i ali dowoloa	d informa	ation			
	MAIN		00111100				— r: BOOL;	
		Objec	:t 				DRD;	
		Proje	ct datab	ase				
		Optio	ns					
		Trans	late into:	other lar	nguages		▶ _	
		Docur	ment					>
		Expor	rt					~
		Impor	rt				HEN	
		Merge	e				- double, check + 1:	
		Comp	are					
		Cloba	ct Info I Search					~
		Globa	il Replaci	 e				>
		View	Instance					
		Show	Call Tre	е			'MAIN'	<u> </u>
		Show	Cross R	eference			Standard'	-
	■ t <mark>:</mark>	Check	k					>
		Add A	Action					
Compile	es the c	Hser	Group Pa	acciulorde			Larget: Local (192.168.1.239.1.1), Run Time: 1	TwinCAT Confi

The TwinCAT PLC Control software will create a file with extension TPY.

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	z	ß	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Ta	g URI			

The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag import	ters are available for this protocol. Please select the importer type and	d continue.
Version	Туре	
TwinCAT v1.0	Linear	
Tag Editor exported xml 1	1.1 General	
Vatched dictionary file:		
Keep synchronized		
	ОК	Cancel

Importer	Description
TwinCAT v1.0	Requires a .tpy file.
Linear	All variables will be displayed at the same level.
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	Tags x
	+ — 🎽 🕲 🖉 🚺 🚺 🖉
	Data Tag URI



Note: the Beckhoff driver supports direct access to the PLC tags using the handles; this means that if no tags are added to the PLC and the PLC program is just re-compiled, you do not need to re-import tags as the access to them does not depend from the offset, but only from name.

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols			•
+ - 🎽 🕲 🖉 🦆 🍪	R 🔎 - Search 🍸 Filter by: Data	▼ Items used:6/10000 Protoco	ol: Show all 💿 💮 🗇
Data Type	Comment	Property	Value
Modbus TCP:prot1		✓ Driver	
Model: Modicon Modbus(1-based)		Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort		Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort		Y Dictionary	
Holding Registers 3 unsignedShort		and a second sec	feler.
MRTU1 unsignedShort		Array	Taise
-MRTU2 unsignedShort		Array size	0
MRTU3 unsignedShort		Arrayindex.Subir	ndex 400003
MRTU4 unsignedShort		Comment	
MRTU5 unsignedShort		Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
(3)	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Teps X + - & @ @ > X > D & b & b & b & b & b & b & b & b & b &
P → Search Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Using TwinCAT v1.0 Import Filter

When importing tags, the user can decide to import all the tags from the **.tpy** file or apply a filter importing only a subset of them.

The figure below shows how to specify the filter. The filter consist in a string (no wildcards are supported). The import filter will import only the tags having the specified string in the description.

If the description is applied to an "instance declaration" of a Function Block, all the tags within the block will be imported.

If the string is contained only as comment of some variables inside the Function Block, only that variables will be imported.

Beckhoff ADS importer - Filter selection		?	×
○ Import all		Ok	(
Import tags containing one item (CSV list) in name or comment	HMI	Can	cel
Apply filter also to expansion of structures			
Expand arrays with elements less than	100		
Do not expand arrays of struct (placeholder)			
	Beckhoff ADS importer - Filter selection Import al Import tags containing one item (CSV list) in name or comment Apply filter also to expansion of structures Expand arrays with elements less than Do not expand arrays of struct (placeholder)	Beckhoff ADS importer - Filter selection Import all Import tags containing one item (CSV list) in name or comment Apply filter also to expansion of structures Expand arrays with elements less than 100 Do not expand arrays of struct (placeholder)	Beckhoff ADS importer - Filter selection ? O Import all Or Import tags containing one item (CSV list) in name or comment HMI Apply filter also to expansion of structures Import tags Expand arrays with elements less than 100 Do not expand arrays of struct (placeholder) Import tags

As an example for the use of the import filter, please see the following case.

```
FUNCTION BLOCK FB Motor
VAR INPUT
    bStartMotor: BOOL;
     bReset: BOOL;
END VAR
VAR OUTPUT
     bMotorOn: BOOL;
     bAlarm: BOOL; (* HMI Thermal alarm *)
END VAR
VAR
     sData: STRING;
    bResetStatistics: BOOL; (* HMI Reset statistics *)
END VAR
VAR PERSISTENT
     stStat: ST_MotorStats; (* HMI Motor statistics *)
END VAR
Function block instances declaration:
VAR
     fbMotor1: FB Motor;
     fbMotor2: FB Motor; (* HMI only show Motor 2!! *)
END VAR
```

The following tags will be imported:

- MAIN/fbMotor2/bAlarm
- MAIN/fbMotor2/bResetStatistics
- MAIN/fbMotor2/ST_MotorStats

Override Data Types

The protocol provides special data types which allow you to change the protocol configuration at runtime.

If added in the project, these variables are initialized with the value specified in the project at programming time.



The table below shows which data type to use for any protocol parameter to override at runtime.

Override Data Type	Proto	col Parameter	Description
IP Override	Target IP Address	0.0.0.0	Overrides the PLC IP address. It is an unsignedByte array of 4 elements, one per each byte of IP address.
AMS NetID Override	Target AMS Net ID	0.0.0.0.0	Overrides the PLC AMS NetID. It is an unsignedByte array of 6 elements, one per each byte of AMS NetID.
Panel AMS NetID Override	Panel AMS Net ID	0.0.0.0.0	Overrides the PLC AMS NetID. It is an unsignedByte array of 6 elements, one per each byte of AMS NetID.
Port Override	Target TCP Port	48898	Overrides the PLC TCP port. It is an unsignedShort .
AMS Port Override	Target AMS Port	801	Overrides the PLC AMS port. It is an unsignedShort .
Panel AMS Port Override	Panel AMS Port	32976	Overrides the Panel AMS port. It is an unsignedShort .

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 2 0 0 >] [> 🗞 🛙	제 전철 R	P- Search	Trili	ter by: Data	~	Items used:8/10000 Protocol: Show	ı all
Data A Beckhoff ADS:prot1 Slave id: Alias: Node1	Type Container Container	Tag name		Tag UR			Prope
Model: Insur - Statement - Roll: Statement - Rol	atari Baa anagnadiri anagnadiri anagnadiri anagnadiri	Number Compare Name Compare Name Compare Name Compare Name Name Name Name Name Name Name Name Name Name Name Name Name Name	Import Tag	g(s) - Network select one or mo	e nodes io ass	sociate this dictionary and continue.	×
Water_Level Slave: i: Alias: Node2 Model:	unsignedSho Container	rt Node1/Water_Level	Slave id	Model	Alias Node1 Node2		
			For each se	elected node impo	ort only symbol	s matching that node	Cancel

- Note: Aliasing tag names is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.
 - The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

CAN Direct v2.0x

CAN Direct communication driver allows to communicate with CAN devices over CAN ports of HMI.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

CANport	0	-	Cancel
CANbaud	250	•	
PLC Models			
default			

Element	Description
CANport	Indicates the CAN port used.
	Allowed values are 0, 1, 2, 3 according to hardware platform.
CANbaud	Indicates the baudrate.
	Allowed values are 100, 125, 250, 500, 800, 1000.
PLC Models	Fixed to default.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select CAN Direct from the protocol list: tag definition dialog is displayed.

type	datatype	Arraysize		
FRAME ~	unsignedByte v	0		
Conversion	CAN frame ID		Frame Length /	Mux Val
I	+/- 0x500		85	
Mux Info / Start in frame	Bit size of field	f	rame MODE	~
Time / Flags		<u>()))</u>		
☐ false				

There are two Tags type:

- FRAME
- VAR

FRAME definition

Element	Description			
Туре	This table refers to FRAME type			
Datatype	For FRAME type, there are two data type available:			
	Data Type	Memory Space	Limits	
	unsignedByte frame Mode	8-bit data	0 255	
	unsignedInt[3] idx 0 frame Mode idx 1 frame Length idx 2 frame Time	Array of 32 bit integers		

Element	Description	
Conversion	No conversion allowed for FRAME type	
CAN frame ID	11 or 29 bit CAN frame identifier	
Extended Frame	checked: Frame is 29 bit type	
	not checked: Frame is 11 bit type	
Frame Length	11024: Length of data, expressed in bytes. Default is 8	
Mux Info	Multiplexer information. If the frame is multiplexed it contains the following info:	
	Bit 0 to 15 position of multiplexer in frame (MSbit if multiplexer is big-endian)	
	Bit 16 to 21 size if multiplexer in bit (1 to 63)	
	Bit 22 set if multiplexer is big-endian	
	If frame is not multiplexed Mux Info contains 0	
Bit Size of field	(Grayed out) not available for frames	
Frame MODE	Indicates direction and transmission mode:	
	RX: received frame	
	TXall: frame is sent when all variables in frames are written	
	TXany : frame is sent when any variable in frame is written	
	TXperiod : frame is sent every TIME milliseconds	
Validity Time or Period	For RX frames: indicates the validity time of variables (0=forever), expressed in milliseconds	
	For TX frames: indicates the sending period, expressed in milliseconds	

Writing to elements of the frame tag it is possible to control the behavior

idx 0 frame Mode it is possible to redefine the mode from RX to TX---- etc.

idx 1 frame Length it is possible to adapt frame length on the fly before sending the frame

idx 2 frame Time it is possible to redefine the sending time period for TXperiod mode or the timeout for Rx mode

in case of Mode = TXperiod and Time = 0 every write acceszs to the frame TAG will trigger a transmission of the frame

VAR definition

Element	Description						
Туре	This table refers to VAR type						
Datatype	For VAR type, available data types are:						
Element	Description						
--------------	--	---	----------------------	--	--	--	--
	Data Type	Memory Space	Limits				
	boolean	1-bit data	0 1				
	byte	8-bit data	-128 127				
	short	16-bit data	-32768 32767				
	int	32-bit data	-2.1e9 2.1e9				
	int64	64-bit data	-9.2e18 9.2e18				
	unsignedByte	8-bit data	0 255				
	unsignedShort	16-bit data	0 65535				
	unsignedInt	32-bit data	04.2e9				
	uint64	64-bit data	01.8e19				
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38				
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308				
	string	Array of elements containing character by selected encoding					
Conversion	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB ABCDEFGHIJKLMNOP->OPI Ca BCD Swap Swap2 Swap4 Swa	ancel OK	e according to				
	DataType						
CAN frame ID	11 or 29 bit CAN frame identifie	er. It must match the predefined FRAME	Tag.				
Mux Val	If the frame is multiplexed each frame can contain different information according to the value of multiplexed field. This filed define for which value of multiplexer this VAR must be updated. In this case bit 0 of flags must be set. Otherwise tha VAR will be updated every time the frame is received						
Start bit in	Indicates the starting position int bits (0-63) of the data						

Element	Description
frame	
Var size in bits	Indicates the number of bits of data
Frame Mode	(grayed out) not available for VAR
Flags	Bit 0 indicates that the VAR is multiplexed
	Bit1 indicates that VAR value is big-endian

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected

HMIStudio	×
Multiple tag importer	s are available for this protocol. Please select the importer type and continue.
Version	Туре
CANDirect v1.0	Linear
CANDirect DBC++ v1.0	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	OK Cancel



Туре	Description					
CANDirect v1.0 Linear	Requires a .sym file generated by WE ICCS SDK PLUS specifying inversion of direction and type 1, or by other Symbol editors like PCAN Symbol Editor					
Vector DBC+ v1.0 Linear	Requires a .dbc file generated by Vector CANdb++ Editor					
	All the frames will be generated with type = Rx, so frames created for transmission must be reedit after importation					
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.					
	Tags x					
	+ - ≱ ⓓ ℗ >] [> \$B ₪] ∰] J Data					

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖄	() 🗞 😡 🖏	R 🔎 - Search YFilter by: Data	▼ Ite	<i>tems used:6/10000</i> Protoco	ol: Show all 🛛 😧 Show all tags
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
- MRTU1	unsignedShort			Array	false
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subir	ndex 400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Tags* x Tags* x + - & @ @ > >] () \$ a = 2] () # P Data Trice - CODESTS VSTEMay N2 Dontaner - Model CODESTS 3 Dontaner - Model CoDESTS Dontane
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Example of usage

Define TX Frame

In this example, Tag1 is declared as FRAME tag.

It sends a message to the ID 600 (258HEX) every 100 miliseconds.

type	datatype	Arraysize	
FRAME ~	unsignedInt [] 🛛 🗸	3	
Conversion	CAN frame ID	Frame Le	ength / Mux Val
1	+/- 0x600	\$	*
Nux Info / Start in frame	Bit size of field	frame MODE	~
Time / Flags			



Define RX Frame

In data definition on CODESYS V2:

- 1. In sendpack object, define the frame ID (yellow)
- 2. In sendpack object, define the frame length
- 3. In "Data" array write the value that must be written

Declare Tag2 as FRAME type, with frame MODE set to RX. CAN frame ID must be the same as ID set in data definition on CODESYS side (yellow).



Note: On protocol side, CAN frame ID is expressed in decimal format.

Declare Tag3, Tag4 and Tag5 as VAR type. CAN frame ID must be the same as ID set in data definition on CODESYS side (yellow).

For each VAR tag, set the correct Start bit in frame property.

Refer to following image, which summerize the above example.



JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

This interface provides the access to token queue and sending function. The following commands are supported:

Command	Description
put	Put the token to send contained in string parameter.
get	Get the received token.
get_token_length	Get the length of received token.
tokens_available	Get number of tokens received.
token_ack	Acknowledge reading token.

Using the command interface the following JS code should receive data:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var avail = tagMgr.invokeProtocolCommand(protID, "tokens_available", "");
while (parseInt(avail) > 0)
{
    var str = tagMgr.invokeProtocolCommand(protID, "get", ""); // get the next
token
    var status = tagMgr.invokeProtocolCommand(protID, "token_ack",""); //
acknowledge current token
    avail = tagMgr.invokeProtocolCommand(protID, "tokens_available",""); // get
number of available tokens in queue
}
```

CANopen HMI

The CANopen HMI communication driver has been designed to connect HMI products to a CANopen network. A new device communication profile has been developed for the HMI. This profile takes advantage from the advanced user interface features of the products, while retaining the simple networking concept supported by the CANopen network.

The basic idea is create a client/server communication structure where the HMI is the client and the CANopen controller is the server.

Connection to CANopen network requires the optional CANopen communication module. Verify the suitable version for your HMI model.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Please ensure that the latest driver is used in the application.

CANopen HMI Profile

In this communication model the HMI initiates the communication sessions, acting as a source of messages.

The basic messages are PDO messages with the standard size of 8 bytes.

The COB-ID of the messages is defined in a way that makes clear, from the well-known CANopen rules, what is the target of the PDO message.

The format of the PDO message has been defined according to a custom application layer protocol. This application layer protocol defines a device-independent communication profile optimized for HMI applications.

When the CANopen master controller receives the PDO message, it will interpret its contents and produce a PDO message with the response addressed to the HMI device.

The definition of this client/server relationship is independent of the CANopen Master in the sense that it can easily be supported in any particular CANopen master system. The resulting solution is easily portable to any CANopen master.

The software IDE offers a user interface that adapts itself to show the typical addressing model of CANopen master controller where the panel is going to be connected.

Adapting to different masters is possible using a profile customization file that may contain data definitions for different controller types.

Profile Details

This chapter provides the specification of the HMI profile and describes the subset of the request/response formats used by this implementation of the protocol.

The communication driver in the HMI generates PDO messages initiating communication request sessions as soon as the HMI runtime requires data from the protocol.

The panel is using the first transmit PDO identified by the COB-ID 0x180 combined with the Node Number assigned to the panel.

The communication profile uses only one transmit PDO and one receive PDO; the limited number of bytes available in standard PDO message maybe limiting, in some cases, the driver capabilities especially in terms of performance.

Request Format: HMI to Controller (Transmit PDO)

The PDO message transmitted by the HMI is formatted according to the following table.



Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Offset Low	Offset High	Data 0	Data 1	Data 2	Data 3	Data Length and Job Number	Operation Type and Controller ID

The request frame includes the following elements:

Offset Low	Low byte of the offset (16 bits address) for the requested block of data
Offset High	High byte of the offset (16 bits address) for the requested block of data
Data 0 Data 3	Data for Write Operations; not used in Read Operations
Data Length and Job Number	Contains: number of requested bytes
Operation Type and Controller ID	 Contains: type of operation requested the Controller ID that identifies the target of the message;

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data	Data	Job	Job	Job	Job	Job	Job
Length [1]	Length [0]	Number [5]	Number [4]	Number [3]	Number [2]	Number [1]	Number [0]

The "Data Length" parameter is coded in 2 bits and takes values between 1 and 4 according to the following rules:

00	1 bytes
01	2 bytes
10	3 bytes
11	4 bytes

Note that the elementary size of each data item depends on the Controller memory organization.

The "Job Number" occupies 6 bits and can have values between 0 and 63; the "Job number" parameter is placed as last element in the PDO to ensure data consistency; the PLC program running the controller should constantly monitor the value of the "Job Number" parameter and consider the received message as valid only when detecting a change in the value of the "Job Number" field. "Job Number" is automatically increased at each new communication session (new request frame).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Operation	Controller						
Type	ID [6]	ID [5]	ID [4]	ID [3]	ID [2]	ID [1]	ID [0]

The "Operation Type" uses one bit with the following definition:

0	Read	data is transferred from controller
1	Write	data is transferred to controller

The "Controller ID" uses 6 bits; it represents the Node Number in the CANopen network of the master controller addressed by the current request.

This parameter is required in case the CAN network has more than one master controller; the CANopen standard defines in fact the COB-ID of the messages in a way that all the partners of the bus known the originator. In case more than one master device is present in the same network, the "Controller ID" field will specify the target of each individual request message. Only the master controller that recognizes in this field its own Node ID will consider the message and process the PDO contents.

Response Format: Controller to Panel (Receive PDO)

The PDO message returned by the controller must be formatted as defined in the following table.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Status Flag / Error Code	Dummy – Always 0	Data 0	Data 1	Data 2	Data 3	Data Length and Job Number	Operation Type and Controller ID

The request frame consists of the following elements:

Status Flag / Error Code	Contains the information related to the execution of the operation type of the request; the next table shows the coding information
Data 0 Data 3	Contain the data information returned to the panel in response to a Read request
Data Length and Job Number	It is the copy of the corresponding field of the request frame
Operation Type and Controller ID	It is the copy of the corresponding field of the request frame

	Status Flag / Error Code	
Operation Type in the Request Frame	No Errors	Error
Read	0x01	0x81
Write	0x02	0x82

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "CANopen HMI" from the list of available protocols.

The driver configuration dialog is shown in figure.

2		1:Page1 Protocols x	Tags	Î
ſ	+	· — ^ V @		
ľ		PLC		
	Þ	CANopen HMI:prot1	-	CfgVer=1 (
		A-B DF1	~	1
		A-B DH-485		
		A-B ENET		
		ABB Mint Controller HCP		
		Automation Direct Do-more		
		BACnet		
		Beckhoff ADS		
		CAN Direct		
		CANopen HMI		
		CANopen SDO		
		Client System Variables		
		CODESYS V2 ETH		
		CODESYS V2 SER		

Element	Description
Panel ID	CANopen node ID assigned to the HMI
Controller ID	CANopen Node ID assigned to the CAN controller device
Baud Rate (kbps)	Speed of the CANopen network
Timeout (s)	Maximum allowed time the driver will wait for a response from the PLC before reporting a communication error
Enable Update Rate	Use this option to enable a wait time between two communication requests
Update Rate (ms)	Minimum interval time between two requests; it can be useful when the bus load needs to be properly controller and limited

Element	Description
PLC Models	The list allows selecting the controller model you are going to connect to. The selection will influence the data range offset per each data type according to the specific controller memory resources
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter the node ID per each slave you need to access.

Connecting the HMI to CODESYS V2 Controllers

This chapter describes all the steps you have to follow in order to establish a successful connection between the HMI and CODESYS CANopen master controller.

The PLC support program has been developed with CODESYS programming software version 2.

PLC Library Call

The server function running in the PLC program has been designed in the form of Library called "HMI_Canh", written using the "ST" programming language. Proper working example is available on demand.

The Function Block parameters are the following:

MasterID	CANopen Master Node number;	
MinBound	Lower limit of the PLC memory addressable (visible) by the HMI	
MaxBound	pper limit of the PLC memory addressable (visible) by the HMI	
HHIr	Offset in the PLC memory where the PDO message received from the panel is mapped	
HMIt	Offset in the PLC memory where the PDO message to be sent to the panel is mapped	
MemPt	Offset in the PLC memory where the data is received	
Status	Status	

The PLC Function block support the use of more than one panel simply repeating the call of the same function for all the additional units specifying before each call the proper calling parameters.



CODESYS V2 4PDO

In some cases it is useful to choose the model "CODESYS 4 PDO" where 4 PDO objects are used for transmission and 4 for reception. This solution may provide higher communication speed between the two devices.

To operate with 4 PDO the correct model should be set in HMI project and the PDOs for receive and transmit slots.



Note: CANopen Master PLC Configuration must be configured properly. In case of "CODESYS 4 PDO".

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Line Error	Returned when an error on the communication parameter setup is detected (baud rate); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
CAN port not found	Make sure option module is correctly plugged
CAN port in use	Make sure option module is not already in use
General error	Error cannot be identified; should never be reported; contact technical support

The codes supported for this communication driver are:

CANopen SDO

CANopen SDO communication driver allows to communicate with CANopen Master devices over CAN ports of HMI.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Nopen SDO		X
Panel ID	1	OK Cancel
CANport	0	•
CANbaud	125	•
🦳 Enable Handst	aking	
Timeout (s)	2	* *
🔲 use uniopds40	3-04	
PLC Models		

Element	Description	
PanellD	Indicates the ID of the panel as a slave in the CANopen network.	
	Allowed values are 1 to 127	
CANport	Indicates the CAN port used.	
	Allowed values are 0, 1, 2, 3 according to hardware platform.	

Element	Description
CANbaud	Indicates the baudrate.
	Allowed values are 100, 125, 250, 500, 800, 1000.
Enable Handshaking	Allow handshaking with the CANopen Master during single or multiple write to PLC operations.
	If selected the NewData Ready flag (obj 0x9000 sub1) will remain at 1 until Master will read the indicated object, until a timeout.
	Default is FALSE.
Timeout (s)	Timeout for waiting for Master reading new data when Handshaking is enabled (in seconds).
	Default is 2 seconds.
Use	Select the compatibility with uniopDS403-04.EDS file.
uniopDS403- 04	If selected the object 0x9000 will be compatible with older versions.
	Default is FALSE (use uniopDS403-04B).
PLC Models	Fixed to default.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select CANopen SDO from the Driver list: tag definition dialog is displayed.

type	Offset	subindex		
INT32 Data Base	• 1	0 *		
datatype	Arraysize	Conversion		
int 👻	0	1	+/-	

Element	Description				
Туре	Selects the internal data base in which the tag will be defined Possible values are:				
	 INT232 Data Base (any 	combination of binary and decimal value	es)		
	FLOAT Data Base (sing	le IEEE754 floating point values)			
Datatype	For INT32 Data Base type, the	e available types are:			
	Data Type	Memory Space	Limits		
	boolean	1-bit data	0 1		
	byte	8-bit data	-128 127		
	short	16-bit data	-32768 32767		
	int	32-bit data	-2.1e9 2.1e9		
	unsignedByte	8-bit data	0 255		
	unsignedShort	16-bit data	0 65535		
	unsignedInt	32-bit data	04.2e9		
	string	Array of elements containing character code defined by selected encoding			
	For FLOAT Data Base type the possible value is:				
	Data Type	Memory Space	Limits		
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38		

Element	Description				
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 				
	Note: number of byt UTF-8 or Latin1 in T If Encoding property 2 bytes.	of bytes corresponds to number of string characters if Encoding property is set to 11 in Tag Editor. operty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires			
Conversion	Conversion to be applied to the tag.				
	Conversion				
	inv,swap2 Depending on data	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK type selected, the list Allowed shows one or more conversion types.			
	Value	Description			
	Inv bits	inv : Invert all the bits of the tag.			
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)			
	Negate	neg: Set the opposite of tag value.			
		<i>Example:</i> 25.36 → -25.36			
	AB -> BA	swapnibbles: Swap nibbles in a byte.			
		<i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)			
	ABCD -> CDAB	swap2: Swap bytes in a word.			
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)			
	ABCDEFGH ->	swap4: Swap bytes in a double word.			



Element	Description	
	Value	Description
	GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion a	and click +. The selected item will be added to list Configured .
	If more conversions Configured).	s are configured, they will be applied in order (from top to bottom of list
	Use the arrow butto	ons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



It is possible to import a Tag Editor exported xml

Туре	Description
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	+ - ¾ ⓓ ☑ >] ▷ ▷] Å] ↓ Data

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					-
+ - 👗 🕲 🖉 🔰 🕨	\$ _B ₪] ∰] R	P- Search Tilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 💮 🗖
Data Typ	pe O	omment	^	Property	Value
Modbus TCP:prot1	ntainer			Y Driver	
Model: Modicon Modbus(1-based)	ricanici			Model	Modicon Modbus(1-based)
Holding Registers 1 uns	signedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 uns	signedShort			✓ Dictionary	·
Holding Registers 3 uns	signedShort			Array	false
MRTUI UNS	signedShort			Array size	0
MRTU2 UNS	signedShort			Arraviaday Subiaday	400003
MRTU3 UNS	signedShort			ArrayIndex.Subindex	400003
MRT04 UNS	signedation			Comment	
MRT05 Uns	signedshort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
都	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Team Team Team X + - 2 S >1 D Sa B + - 2 S >1 D Sa B B A
	Data Type Accobers V3 Stringers 2 Container Accobers V3 Stringers 2
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Client System Variables

Client System Variables communication driver allows to create Tags that point to system information.

Refer to <u>Client System Variables > Protocol</u> chapter of User's Manual.

Protocol Editor Settings

Client System Variables communication driver allows to create Tags that point to system information.

Refer to <u>Client System Variables > Protocol</u> chapter of User's Manual.

CODESYS V2 ETH

CODESYS V2 ETH communication driver for supports communication through Ethernet connection with controllers based on the CODESYS V2.3 version.

Protocol Editor settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

CODESYS V2 ETH		×
PLC Network		OK
Alias		Cancel
IP address	0.0.0.0	
Port	1200	
BlockSize	128	
Timeout (ms)	1000	
Protocol type	Level2 ~	
Source Address	0	
Destination Address	0	
PLC Models	l	
Intel Motorola		

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Port	Port number used by the CODESYS V2 Ethernet driver. The default value is set to 1200 , which is also the default setting of CODESYS-based controllers.
Block Size	Maximum block size supported by your controller (limit is 1024 KB).
Timeout (ms)	Time delay in milliseconds between two retries of the same message when communication fails.

Element	Description
Protocol type	Protocol variant to be used. Please make sure you check which protocol variant is supported by the CODESYS runtime you want to connect.
Source Address, Destination Address	Available only when TCP/IP Level 2 Route is selected in Protocol Type . The Destination is the node of the PLC and allows the protocol to read variables in a sub- network. The address is used to read variables when multiple PLCs are connected in a sub-network (serial network) but only one have the Ethernet interface.
PLC Models	Two PLC models are available.
	IntelMotorola
PLC Network	The address for all controllers in multiple connections. PLC network check box must be selected to enable multiple connections.

CODESYS V2 Ethernet driver supports connection to multiple controllers starting from version V1.60.

Note: CODESYS V2 Ethernet driver is recommended when creating projects for the internal controller iPLC CODESYS. To use the CODESYS V2 Ethernet driver with iPLC, configure the IP address of the PLC as localhost (127.0.0.1).

iPLC CODESYS supports communication with CODESYS V2 Ethernet driver with symbol based support starting from V1.55 and above.

Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

+-~~ *		Modbus TCP:prot1		-6		
Name /		Group	Driver	Addre	ss	Comment
Node's Tot Individual		Add We	Bue TCP pet 1	1 11 Durnin	mailthat	
Node's Cate booltonia		A012 Wee	disa TCP pest1	1 12 During	maddiment	
Phinder/1/Phi 18/87228. Janu		A012 Wee	disa TCP pett	1 Dillummer	mathat	
Note 1 OBOGETBROW		Add We	disa TCP peak!	1 245 Dumm	prestiment	
No. doi:10.0101_0317_houde	-ofia	,1885 Wee	dison TCP protil	1.1 During	method	
Node 1/R, DATA, Nodes	fia .	A012 Wee	disa TCP prot1	1 2 During	Indiana	
THAN DON TO ANNA THE OWNER		Add We	dison TCP prot/1	1 3 Durnip	Horfdhan	
Node1/Water_level		Addentical	adan Ethianali	1 10 0 unsig	redShort	
1	Net	and a				
	netw	OIK				
		Slave Id	Model		Alias	
		(4) 新田 (1)	Mailtin ma	(These	Node1	
		电振荡器	Themeses - man	But .	Node2	
8 督 8						
tagname	me	1				
Water_level				0	ol.	
W/srttzte					UK	
R_3//78_builtetie						
OVER_BARE_Healthing	C	-			· ····	1000
CRECKLETERNCAR	1944/42	245	0	12/19	igned/3hof	Tallor
IN WITTER ave	1996(42)	0	0	13/19	igned that	Tallon
Take, Peolindia	1996(42)	12	0	12/19	igreal Traff	Tallor
Edd. Northropics	1855(72)	11	9	14479	terrent frank	Faller



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

CODESYS software settings

When creating the project in CODESYS, select **Download symbol file**.

Target Platform Memory Layout Gen	eral Network functionality Visualization	
1/0-Configuration		
Configurable		
	No <u>a</u> ddress check	
	Download symbol file	
Byte addressing mode	PLC Browser	Load bootproject automatically
	✓ <u>I</u> race	
🔽 🖸 nline Change	Cycle independent forcing	🗖 Retain for <u>c</u> ing 🛛 S <u>a</u> ve
Update unused I/Os		

Note: CODESYS V2 Ethernet communication driver supports the automatic symbol file (SDB) upload from the PLC; any change in the tag offset due to new compilation of the PLC program does not require a symbol file re-import. Tag file has to be re-imported only in case of tag rename or definition of new tags.

When the option **Download symbol file** is not available or cleared, the protocol can work only if the **ProjectId** tag is imported. If the tag offset changes because of a new compilation of the PLC program, the symbol file must be re-imported.

	\square	1:Page	1	Proto	cols	7	Tags	×							
	+	-	X	Ð	ß	>]	Þ	A 9B	B	翻	R	₽- Search	7 F	ilter by: Data	~
I	Dat	а	^		1	Туре		Tag n	ame	Grou	ips	Tag URI		Encoding	
	4	CODES Model:	YS V2 E	TH:pr	ot1 (Contai	ner								
I		Pro	jectId		L.	unsign	edInt	Proje	ctId			0?ProjectId?PROJ_ID?1	106673?unsigned	dInt	

Data types

I

The import module supports variables of standard data types and user defined data types.

Supported data types	 BOOL WORD DWORD INT UINT UDINT DINT STRING* REAL TIME DATE & TIME and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main
	manual. Note *: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.
Unsupported data types	 LWORD LINT LREAL

Tag conversion

Conversion to be applied to the tag.



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description						
Inv bits	inv: Invert all the bits of the tag.						
	Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)						
Negate	neg: Set the opposite of tag value.						
	<i>Example:</i> 25.36 → -25.36						
AB -> BA	swapnibbles: Swap nibbles in a byte.						
	Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)						
ABCD -> CDAB	swap2: Swap bytes in a word.						
	Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)						
ABCDEFGH ->	swap4 : Swap bytes in a double word.						
GHEFCDAB	<i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)						
ABCNOP ->	swap8: Swap bytes in a long word.						
OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 1000000110						
	$ \begin{array}{c} 0001110010111011011001000101101000011100101$						
	1 10000011100 1010101000010100010110110110						
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)						
	Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)						

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

DDESYS V2 ETH				
Address Space Node Override IP	Tag Name		Offset	
SubIndex Data Type 0 v BYTE[]	-	Arraysize 4		
Conversion	+/-			
		ок	Annulla Ap	plica ?

Tag Import

Exporting Tags from PLC

When configuring PLC using the manufacturer's configuration software, enable Symbol file (.sym extension) creation under the CODESYS programming software:

- 1. In the **Project** menu, click **Options**.
- 2. Click Symbol configuration.
- 3. Select Dump symbol entries.
- 4. Click OK.



Note: Click then **Configure symbol file...** and select **Export variables of object**. We recommend to clear the check box and re-select to be sure about the proper settings.



In some cases, duplication of symbols for variables associated to integrated I/O modules in the ".sym" file may be experienced. To remove the duplication selected the "PLC Configuration" voice from the objects list and uncheck the option "Export variables of object".

III	
Export variables of object	
Export data entries	
Export structure components	
Export array entries	
₩rite access	

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	-	z	ß	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importer	s are available for this protocol. Please select the importer type and continue.
Version	Туре
CODESYS2 sym v1.1	Linear
CODESYS2 sym v1.1	Hierarchical
CODESYS2 sym_xml v1.0	Hierarchical
Tag Editor exported xml 1.1	General
Natched dictionary file:	
Keep synchronized	
	OK Cancel

Importer	Description
CODESYS2 sym v1.1	Requires a .sym file.
Linear	All variables will be displayed at the same level.
CODESYS2 sym v1.1	Requires a .sym file.
Hierarchical	All variables will be displayed according to CODESYS V2 Hierarchical view.

Importer	Description					
CODESYS2 sym_xml v1.0 Hierarchical	Requires a .sym_xml file.					
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.					

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

+	- 🎽 🕲 🖄 🚺	> 🗞 🖬 🚮	R 🔎 - Search Trilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🗸 🗹 Show all tags 🔅 🗍
Data	^	Туре	Comment	^	Property	Value
Modbus TCP:prot1 Container						
	Nodel: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
	- Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
	Holding Registers 2	unsignedShort			M Dictionary	
	Holding Registers 3	unsignedShort			+ Dicuonary	
	-MRTU1	unsignedShort			Array	false
	- MRTU2	unsignedShort			Array size	0
	MRTU3	unsignedShort			Arrayindex.Subindex	400003
	- MRTU4	unsignedShort			Comment	
	- MRTU5	unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Teas Teas <th< th=""></th<>
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Exporting tag arrays

In CODESYS V2 program tag arrays are split into individual elements and one tag for each element is created. In the following example one array with 10 elements.





Note: If **Export array entries** is selected, a tag for each element will be created and exported into the .sym file. The entire tag list will be automatically imported into the Tag editor.

By clearing **Export array entries** only one tag for each one array can be created.





Note: When **Export array entries** has been cleared, only one tag is created and exported into the .sym file. The array is not automatically imported in the Tag editor and tags need to be manually configured in Tag editor.

	ST CoDeSin ETUrnet1				
	Cobesys Emploit		Add	I Commond	D OM Antina
Vame PLC_PRG/Y_POS_P1	Group	CoDeSvs ETH prot 1	Address	Comment	B/W false
	CoDeSys ETH CoDeSys ETH Address Space	Tag Name	Offset		
	Local/Global Var SubIndex Data Typ short	PLC_PRG/Y_POS_P1	144 -	\array_collapsed SYM	40 50 60
tagname /	Conversion	+/-	2 ;ProjecId=120366 3 ;Checksum=0 4 ;Filesze=01 5 .rVersion:REAL:4: 6 PLC_PRG.Y_POS_P1 7	140:4:b:16#02000044 ARRAY [010] OF INT:4	:144:22:b:16#0000040
PLC_PRG/Y_POS_P1[0] PLC_PRG/Y_POS_P1[10] PLC_PRG/Y_POS_P1[1] PLC_PRG/Y_POS_P1[2] PLC_PRG/Y_POS_P1[3]	short	OK		148 4 150	
PLC_PRG/Y_POS_P1[4]	short	false	0	4 152	
PLC PRG/Y POS P1151	ebort	falee	0	4 154	

All tag elements can be referenced in the editor using TagIndex in the Attach to Tag dialog.

Source:	Units
🖲 Tag (🖱 System 🔘 Widget 🔘 Recipe
Tag:	PLC_PRG/Y_POS_P1
	PLC_PRG/Y_POS_P1

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause and action
Symbols file not present	Check Symbol file and download again the PLC program.
"tag" not present in Symbols files	Check if the Tag is present into the PLC project.
Time out on Acknoledge	Controller didn't send acknowledge.
Time out on last Acknoledge	Controller didn't sent last ack.
Time out on data reciving	Controller does not reply with data.
Connection timeout	Device not connected.

CODESYS V2 SER

The CODESYS V2 SER communication driver has been designed for serial communication with controllers based on CODESYS V2.3.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Accordingly, always ensure that the latest driver is used in the application.

Limitations

Max block size is 1024 byte.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "CODESYS Serial" from the list of available protocols.

	×
Comm	ОК
	Cancel
128	
1000	
2	
	Comm

Element	Description					
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node					
Block Size	Enter the max block size supported by your controller (limit is 1024)					
Timeout	The number of milliseconds between retries when communication fails					
Num of repeats	This parameter defines the number of times a certain message will be sent to the controller before reporting the communication error status. A value of 1 for the parameter "No of repeats" means that the panel will eventually report the communication error status if the response to the first request packet is not correct.					
PLC Model	Defines the byte order that will be used by the communication driver when sending					
Element	Description					
--	---	--	--	--	--	--
	communication frames to the PLC					
	Comm Parameter Dialog					
	OK Port com1 Parity none Baudrate 19200 Stop bits 1					
Port	Serial port selection.					
	 COM1 is the PLC port. COM2 is PC/Printer port on panels with 2 serial ports or refers to the optional plug-in module plugged in Slot 1/2 for panels with 1 serial port on-board. COM3 refers to the optional plug-in module plugged in Slot 3/4 for panels with 1 serial port on-board. 					
Baudrate, Parity, Data bits, Stop bits	Communication parameters for the serial line.					

CODESYS Software Settings

When creating the project in CODESYS, the option Download Symbol File (in Target Settings/General) must be checked.

Target Settings			x
Configuration: EXOR/SITEK CoDeSy	s for JMobileWCE (WCE/ARM UN31)	•	
Target Platform Memory Layout Gene	ral Network functionality Visualization	1	
- 1/O-Configuration			
Configurable			
	No <u>a</u> ddress check		
	✓ Download <u>s</u> ymbol file		
Byte addressing mode	PLC Browser	Load bootproject automatically	
☑ _Online Change	Cycle independent forcing	🗖 Retain for <u>c</u> ing 🗖 S <u>a</u> ve	
Update unused I/Os			
		Default OK Cancel	

Note: CODESYS Serial communication driver supports the automatic symbol file (SDB) upload from the PLC; any change in the tag offset due to new compilation of the PLC program does not require a symbol file reimport. Tag file has to be re-imported only in case of tag rename or definition of new tags.

When the option Download symbol file is not available or not checked, the protocol can work only if the ProjectId tag is imported. Any change in the tag offset due to new compilation of the PLC program requires that symbol file is imported again.

1:Page1 Protocols Tags* ×		
+ - 🖌 🗇 🖉 >] D 🖕	23 53 R P- Search	Titems used: 273/20000 Protocol: Show all
Deta CODESYS V2 SER:prot1. Model: amount	Type Tag name	Tag URI
Popictid Popictid	unignedint Projectid	07Projectid PROJ_10780625Punsignedint

Standard Data Types

The following data types in the CODESYS programming tool are considered standard data types by the import module:

BOOL WORD DWORD INT UINT UDINT DINT STRING REAL TIME DATE & TIME and 1-dimensional ARRAY of the types above.

The 64-bit data types LWORD, LINT and LREAL are not supported.

String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35)) or default size (str: STRING) which is 80 characters.

Tag Conversion

Conversion to be applied to the tag.

Conversion

Inv,swap2

Allowed

Configured

Inv bits

ABCD->CDAB

ABCDEFGH->GHEFCDAB
Inv bits

Cancel
OK

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)
Negate	neg : Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)

Value	Description
ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 1000000110$ 0001110010111011010001011010000111001010
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Tag Array

Tag Arrays are split into individual elements and one Tag for each element is created. The figure below shows an example of one Array with 10 elements





Note: When "Export array entries" is set, a tag for each element is created and exported into the SYM file. The entire tag list is automatically imported into Tag Editor.

The amount of tags can be reduced and only one Tag for each one array can be created by removing the checkbox "Export array entries", see figure below.





Note: When "Export array entries" is not set, only one tag is created and exported into the SYM file. The Array will not be automatically imported in Tag Editor and Tags need to be manually configured in Tag Editor

1:Page1 protocols*	Tags* ×				
+-~~*	CoDeSys ETH:prot1	- 🖓 🗖			
Name	Group	Driver	Address	Comment	R/W Active
PLC_PRG/Y_POS_P1		CoDeSys ETH:prot1	D PLC_PRG/Y_POS_P1 4 14		R/W false Va
	CoDeSys ETH CoDeSys ETH Address Space Local/Global Var • SubIndex Data Typ o • SubIndex Conversion	Tag Name PLC_PRG/Y_POS_P1 e Arraysize v 10 +// OK	Offset 144 CANADA CANADA CONTRACTOR CO	Aarray_collapsed SVM 20,30,	:4;144;22;b:16#00000040 148
PLC_PRG/Y_POS_P1[4]	short	false	0	4	152
PLC PRG/Y POS P1151	short	false	0	4	154

All Tag elements can be referenced in the editor using "TagIndex" in the "Attach to Tag" dialog

ld1.value		
Tag	Scale XForms	
Source:		
Tag	🔘 System 🔘 Widget 🔘 Recipe	
Tag:	PLC_PRG/Y_POS_1	٩
	CODESYS V2 SER:prot1	
	PLC_PRG/Y_POS_1	
Rea	d Only 💿 Read/Write 💿 Write Only	TagIndex: 0
		OK Cancel

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias".



Note: An Aliasing tag name is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name. The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

When configuring PLC using the manufacturer's configuration software, make sure to enable Symbol file creation (file with .SYM extension). It can be done under the CODESYS programming software, by selecting "Project\Option\Symbol configuration" and mark the check box "Dump symbol entries" as shown in the picture below.

Options			22
Category: Load & Save User Information Editor Desktop Colors Directories Log Build Passwords Source download Symbol configuration Database-connection Macros	Dump symbol entries Dump XML symbol table Configure symbol file	Set object attributes ModbusTCPMaster.pro PDJs PDL_PRG (PRG) B- Global Variables P- Global Variables Global Variables P- Glob	OK SX
		Export variables of object Export data entries Export structure components Export array entries Write access	



Note: Click then on the "Configure symbol file..." button and make sure the "Export variables of object" check box is marked as shown in the following picture. We recommend to un-check the check box and mark it again to be sure about the proper settings.

Select the driver in the Studio tag editor and click on the "Import tag" button to start the importer.

<u>ي</u>	ProjectView 4 ×	1:Page1 Protocols Tags x	
Obje	+ − 4 ∧ ∨	🕇 — 🐰 🕲 🔊 🎝 🌓 🎭 📾 📸 R 🔎 Search	Filter by: Data
ctView	 Improject 1 Improject properties Improject Project properties 	Data Type Tag name Tag URI Dictionary CODESYS V2 SER:prot1 Model: Intel Container	
~	 Gunified Mormal 	HMIStudio	×
Tag Cro	 1: Page 1 Dialog Templates 	D Multiple tag importers are available for this protocol. Please select the importer typ	e and continue.
oss F	Y 📄 Configuration	Version Type	
Refer	Protocols	CODESYS2 sym v1.1 Linear	
ence	Indexed Tag Set	CODESYS2 sym v1.0 Hierarchical	
	Trends	CODESYS2 sym_xml v1.0 Hierarchical	
	Alarms	Tag Editor exported xml 1.1 General	
	📌 Events Buffer 🐨 Scheduler		
	🖗 MultiLanguage 🗊 Screen Saver		
	😓 Database Links		
	Data transfers A Interfaces	Watched dictionary file:	
	> 🔒 Security		
	Recipe	Keep synchronized	
	 Dictionaries Environmentaries Keypads 	OK	Cancel

Once the importer has been selected, locate the symbol file and click Open.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in the following figure.

togname	memorytype	arrayindex.subin	index	datatype	апау	arraysize
str	MW0	8	0	string-16	true	16
ARRAY_WORD[1]	MW0	0	0	unsignedShort	false	0
ARRAY_WORD[2]	MW0	1	0	unsignedShort	false	0
ARRAY_WORD[3]	MW0	2	0	unsignedShort	false	0
ARRAY_WORD[4]	MW0	3	0	unsignedShort	false	0
MDW2	MD0	2	0	unsignedInt	false	0
MDW3	MD0	3	0	unsignedInt	false	0

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
Symbol file not present	Check Symbol file and download again the PLC program
"tag" not present in Symbol file	Check if the Tag is present in the PLC project
Time out on Acknowledge	Controller didn't send acknowledge
Time out on last Acknoledge	Controller didn't send last acknowledge
Time out on data receiving	Controlled does not reply with data
Connection timeout	Device not connected

CODESYS V3 ETH

The CODESYS V3 ETH communication driver supports communication thought Ethernet connection with controllers based on the CODESYS V3 PLC software by the company 3S.



Note: To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Make sure the latest driver is used in the application.



Note: Changes in the controller protocol or hardware may have occurred since this documentation was created. This may interfere with the functionality of this driver. Therefore, always test and verify the functionality of the application.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

CODESYS V3 ETH		×
PLC Network		ОК
Alias		Cancel
Protocol	TCP ~	
IP address	0.0.0.0	
Timeout (ms)	1000	
Variable list count	255	
PLC Models		
CODESYS 3		
Schneider		

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
Protocol	Select between TCP and UDP protocol type.
IP address	Ethernet IP address of the controller
Variable	Variable List is the best method to achieve higher performance in the CODESYS V3

Element	Description							
list count	communication protocol, as it allows requesting multiple data items in a single protocol session.							
	Since some implementations of CODESYS V3 at runtime have a limited number of Variable Lists that can be allocated, this parameter allows you to set the maximum number of Variable Lists the communication driver tries to create in the PLC.							
PLC Model	Byte order that will be used by the communication driver when sending communication frames to the PLC.							
Timeout	Number of milliseconds between retries when communication fails.							
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.							
	CODESYS V3 ETH CPLC Network Alias Protocol Protocol Top P address 0 Tomeout (ms) 1000 Variable list count 255 PLC Models Schneider Slaves Add Delete Model							



Note: Refer to the controller documentation to verify required values for the parameters **Full node address** or **Variable list count**.

Tag Import

Exporting Tags from PLC

When creating the project using CODESYS V3, properly configure the symbol file to contain the required variables.

 To add the Symbol configuration in CODESYS V3 project, right click on the Application item from the project tree, then into the context menu select Add Object > Symbol configuration. The symbol configuration item will be added to the project tree.



2. Double click on Symbol configuration item, then click on "Build" button.

🥂 💾 Syn	nbol Configuration	I X						
🛛 View 🚽 🛗 Build 🛛 🛱 Settings 👻								
Execute "Build" command to be able to select variables (you need an error-free build). Details Changed symbol configuration will be transferred with the next download or online change								
Symbols	Access Rights	Maximal	Attribute	Туре	Members	Comment		

3. Symbol configuration item contains a list of all the variables available into the CODESYS V3 project, single variables or groups of variables can be selected by checking the corresponding item in the list.

Symbol Configuration 🗙						
📉 View 👻 🎬 Build 🛛 🛱 Settings 👻						
Changed symbol configuration will be transf	erred with the next (download or o	online change			
Symbols	Access Rights	Maximal	Attribute	Туре	Members	Comment
Gonstants						
CompilerVersion		*		VERSION		Does the target support an FPU
🔤 🔌 RuntimeVersion		*		VERSION		Does the target support an FPU
GVL						
🔤 🛷 MyVARGlobal1	*	*		INT		
🛛 🕼 MyVARGlobal2	*	*		INT		
IoConfig_Globals						
🔲 🔌 nIoConfigTaskMapCount		N		DINT		
🔤 🔌 pIoConfigTaskMap		*		POINTER TO IoConfigTaskMap		
🖶 🔲 📄 PLC_PRG						
🐨 📝 myVAR1	*	*		INT		
myVAR2		*		BOOL		
wyVAR3	*	*		REAL		
🖶 🔄 {} BPLog						
🗄 🔄 🚼 IecVarAccessLibrary						

4. After the symbols have been configured, download the project or use the **Generate code** function (Build > Generate code) to create an .xml file containing all the variables read to be imported in the Tag Editor.



Note: GVL global variables are listed in Symbols Configuration only if they are used in PLC program. To always list global variables right click on GVL and select "Properties". From "Build" tab check "Link Always" option.

Common	Link To File	Access control	Network properties	Build
Exc	ude from bu	iild		
Exte	ernal implem	entation		
(La	te link in the	runtime system)	
Ena	ble system o	call		
🔽 Lini	c Always			
Compi	ler defines:			

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio	×						
Multiple tag importers are available for this protocol. Please select the importer type and continue.							
Version	Туре						
CODESYS3 xml v1.0	Linear						
CODESYS3 xml v1.0	Hierarchical						
Tag Editor exported xml 1.	1 General						
Watched dictionary file:							
Keep synchronized							
	OK Cancel						

Importer	Description			
CODESYS3 xml v1.0	Requires an .xml file.			
Linear	All variables will be displayed at the same level.			
CODESYS3 xml v1.0	Requires an .xml file.			
Hierarchical	All variables will be displayed according to CODESYS V3 Hierarchical view.			
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.			
	Tags ×			
	+ - 🎽 🕲 🖄 🕩 🎝 🕨			
	Data Tag URI			

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					-
+ - 👗 🕲 🖄 🔰 [> 🗞 🖬 🚮	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🔅 🗌
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	Container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	·
Holding Registers 3	unsignedShort			Array	false
MRIUI	unsignedShort			Array size	0
- MRTU2	unsignedShort		_	Airdy Size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
Holding Registers 1 Holding Registers 2 Holding Registers 3 MRTU1 MRTU2 MRTU2 MRTU3 MRTU4 MRTU5	unsignedShort unsignedShort unsignedShort unsignedShort unsignedShort unsignedShort unsignedShort unsignedShort		-	Protocol Victionary Array Array size Arrayindex.Subindex Comment Data type	Modbus TCP:prot1 false 0 400003 unsignedShort

Toolbar item	Description				
BA	Import Tag(s).				
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project				
€ ä	Update Tag(s).				
	Click on this icon to update the tags in the project, due a new dictionary import.				
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:				
P → Search Y Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo- box item selected.				

Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

エークインを回		Modbus T	CP:prot1	→]@_		
Name /	1	Group	Driver		Address	Comment
Node 5 Tot, Nodrotia			ADD Wordhow TO	Panel 11	1 Dumagnes/Direct	
Node's/Calls_hodrolls			Add Wordson TC	Panki 11	2 Example (Short	
Node1/PL_W/X7ER_level			Add Wordson TC	Panti 10	DurageadDeat	
Notes CROSPERCIP			Add Wordson TC	Pank) 1.34	1.Curvigreal@hoft	
Node5.42427_BX7_hodeolo			ADD Modeue TC	Pgnt1 11	Duragenet@hoff	
The der 1/FL_CATA_holdholde			ADD Modeum TC	Pgrit1 12	Compared Short	
TANK A TANK TELE			ABD Wordhum TC	Pgnk1 13	Durnigreed/Droet	
Node1/Water_level			AND COLORING ST	11 1	0 0 unsignedShort	
	0	Select Netwo	ork node id	lodel	Alias	
		A.X.A.A.		Selferer I mailling	Node 1	
		444.8		And and a state of the state	Node2	
8 dd R						
tagname	ne	-				
Water_level					Colo	
Wenter					UK	Cancel
R_GMTH_Healthe						
DATE, BHT, Hulbers	C		-		www.entres	1000
CONTRACTOR AND	1855(72)		245	0	ant singer staff Stream	Tallor
IN, WITCH, and	1016(72)		0		un signal Short	Tallon
Takis, Pendrolas	1844(12)		12		unningented Transf	Tallor
the second	the second se					



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

Data Types

The import module supports variables of standard data types and user defined data types.

Supported	
Supporteu data typos	
uala lypes	• INT
	• SINT
	• UINT
	• UDINT
	• DINT
	STRING*
	• REAL
	• LREAL
	• BYTE
	• ULINT
	• LINT
	and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.
	Note *: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.
Unsupported	LWORD
data types	• LINT

Tag conversion

Conversion to be applied to the tag.

Conversion			
inv,swap2	Allowed		Configured
	BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ - ~ ~	Inv bits ABCD->CDAB
		(Cancel OK

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
Negate	neg : Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
ABCNOP ->	swap8: Swap bytes in a long word.
OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.10000000110 0001110010111011001000101101000011100101
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
	Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

CODESYS V3 ETH			×
CODESYS V3 ETH			
memTvne	Data type	Arraysize	
Node override IP V	BYTE[]	✓ 4	
Conversion	Tag name		
	+/-		
	ОК	Cancel Apply H	elp

Application Status

The protocol provides the special data type Application Status which allows you to check or change the applications status.



The tags pointing to Application Status must contains into field "**Tag name**" the name of the PLC application (frequently the default name is "Application")

If the HMI device is connected to a network with more than one controller node, each node has its own Application Status variable.

CODESYS V3 ETH	×
CODESYS V3 ETH	
memType Application Status Conversion	Data type Arraysize BYTE 0 Tag name +/-
Application Status	OK Cancel Apply Help Description
0	RUNNING
1	STOPPED
2	HALTED ON BreakPoint
	It is not possible to write 2 as new status
251	Reboot CODESYS device
252	Shutdown CODESYS
253	Reset ORIGIN

Reset COLD

Reset WARM

Communication Status

254

255

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Control Techniques Modbus TCP

Control Techniques Unidrive M Series are using Modbus TCP protocol where the device id should be always set to 0 or 255. This communication protocol is known as Control Techniques Modbus TCP. The HMI protocol identifies Control Techniques Modbus TCP devices using their IP addresses

You should take note of these addresses as you assign them because you will need them later in the set-up phase of the user interface application. The HMI protocol can be set to access to a different menu range

Different physical media, gateways, routers and hubs can be used in the communication network. Also, other devices can independently make simultaneous use of the network. However, it is important to ensure that the traffic generated by these devices does not degrade the communication speed (round-trip time) to an unacceptable level.

The implementation of the protocol operates as a Modbus TCP client only.

The HMI Control Techniques Modbus TCP protocol uses the standard port number 502 as the destination port.

The HMI Control Techniques Modbus TCP protocol supports the standard commonly referred as "Ethernet II".

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "Control Techniques Modbus TCP" from the list of available protocols.

The driver configuration dialog is shown in figure.

Control Techniques Modbu	is TCP		×
PLC Network		OK	
Alias		Cancel	
IP address	0.0.0		
Port	502		
Modbus Mode	Standard ~		
Timeout (ms)	2000		
PLC Models			
default			

Element	Description			
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node			
IP address	Ethernet IP address of the controller			
Port	Port number used by the Modbus TCP driver; the default value can be changed when the communication goes through routers or Internet gateways where the default port number is already in use			
Modbus Mode	This parameter define the communication protocol used and needs to be set in according with the setting made on the drive (parameter S.15.013). Modified mode is provided to allow register numbers up to 255 to be addressed. If any menus with numbers above 63 should contain more than 99 parameters, then these parameters cannot be accessed via Modbus.			
	Standard	(menu number * 100) + parameter number - 1		
	where menu number \leq 162 and parameter number \leq 99			
	Modified (menu number * 256) + parameter number – 1			
		where menu number \leq 63 and parameter number \leq 255		
Timeout (ms)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from the server device. Value is expressed in milliseconds.			

Element	Description					
PLC	Selection of device models that may affect operation of the protocol.					
Models	Currently only one model is available					
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter IP Address for all controllers.					
	Control Techniques	Modbus TCP		×		
	✓ PLC Network OK					
	Alias	ntrol Techniques Mo	odbus TCP		×	
	IP address			ОК		
	Port	Alias		Cancel		
	Modbus M	IP address	0.0.0.1			
	Timeout (r	Port	502			
	PLC Model	Modbus Mode	Standard ~			
		Timeout (ms)	2000			
		PLC Models default				
	Slaves					
	Slave Id					

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The error codes supported by this communication driver are:

Error	Notes
No response	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Incorrect node address in response	The panel did receive from the controller a response with invalid node address

Error	Notes
The received message too short	The panel did receive from the controller a response with invalid format
Incorrect writing data acknowledge	Controller did not accept write request; ensure the data programmed in the project are consistent with the controller resources

CT Modbus CMP ETH

The CT Modbus CMP ETH communication protocol is known also as "Modbus over CTNet"

CMP stands for CTNet Message Protocol; it is a messaging system designed to implement distributed control applications. The protocol permits exchange of parameters between Control Techniques drives and HMI devices, SCADA systems or other computer applications.

CMP is normally encapsulated in an existing network protocol. CMP has been successfully encapsulated also into the Modbus network. The communication protocol support implements the Modbus encapsulation of CMP.

Unidrive SP drives support the CTNet network using optional communication units called "SM Applications" modules.

Please note that changes in the communication protocol specifications or PLC hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in PLC hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Concept of Operation

The network topology supported by the HMI communication protocol is shown in the figure below.

The HMI panel will communicate with a set of drives over the network; drives are addressed using their Modbus ID node number.

Each drive can host up to three SM application boards; they may be used for CTNet communication.

The addressing model is based on a three level space; from the HMI point of view, each drive is identified with a unique ID composed of a maximum of three numbers; the ID can be calculated looking to the network topology.



Protocol Editor Settings

Add (+) a driver in the Protocol Editor and select the protocol called "CT Modbus CMP ETH" from the list of available protocols.

The CT Modbus CMP ETH driver supports three different protocol types:

- none
- CTNet
- Ethernet

The protocol type can be selected from the "Routing Type" combo box in the dialog.

	Commun					
Alise		Routing Type	None 🗸	Destination Subsode	0	Ĩ
Modbus ID Slave Nr	0	Routing CTNede ID	0			
Siot Nr	0	Routing IP Address	0 0 0 0			
Transport Layer	Ethernet 🗸	Routing Port	-502			
IP Address	0.0.0	Routing Sleve D	0			
Port	502	Destination Port	0			
PLC Models						

Some of the parameters of the dialog are common to all the protocol types, some others are specific.

The parameters common to all protocol types are the following:

Element	Description			
Modbus ID Slave Nr	Valid slave node addresses are 1 through 247. In the request from the protocol master this value indicates the target slave node.			
Slot Nr	This value gives the option slot to which the message is directed. The port/slot mapping is as follows:			
	Port	Message addressed to		
	1	Option in slot 1		
	2	Option in slot 2		
	3	Option in slot 3		
IP address	Ethernet IP address of the controller			
Port	Allows to change the default port number used by the Modbus TCP driver; it could be useful whenever the communication goes through Routers or Internet gateways where the default port number is already in use. Default value for this parameter is 502.			
Routing type	The FC64 encapsulate routing between nodes and CMP destination s Modbus TCP server to message through ano	ed protocol includes extra destination fields to be used for message s on different networks. The combination of CMP destination port subnode address or subnode addressing scheme fields, allows a o decide whether to process a received message or retransmit the ther port onto a different communications network.		
	User can select one of	f the following routing methods:		
	 None: means the message is direct on the subnetwork of the subnetwork o	hat the communication will be established directly to the drive. The ected to a drive or to an option in the drive, and there is no routing ork to be performed.		
	 CTNet: users c subnetwork. 	an enter CTNet node number which represents the drive in the		
	Ethernet: SM-E Ethernet.	Ethernet modules will provide the capability to reroute messages on		
Routing CTNode ID	When Ethernet routing Routing port, Routing	g method has been selected, you have to enter Routing IP Address, Slave ID, Destination Port and Destination Subnode of the drive		
Routing IP Address	you want to connect. For more information on routing, please check the drive user's manual or CT Modbus specification.			
Routing Port				
Routing Slave ID				
Destination Port				
Destination Subnode				

Element	Description
PLC Network	The protocol allows the connection of multiple drives to one operator panel. To set-up multiple connections, check "PLC Network" checkbox and enter parameters for each drive you want to connect.
PLC Model	Selection of device models that may affect operation of the protocol.
	Currently only one model is available

	CT Modbus CMP ETH						×
CT Modbus CMP ETH							
PLC Network	Alas		Routing Type	None 🗸 🗸	Destination Subnode	0	
Alas	Nodbus ID Slave Nr	D	Routing CTNode ID	0			
	Silot Nr	1	Routing IP Address	0 . 0 . 0 . 0			
MODUS ID SAVE	Transport Layer	Ethemet ~	Routing Port	542			
Slot Nr	IP Address	0.0.0.0	Routing Stave ID	0			
Transport Layer	Port	502	Dectination Port	0			
IP Address	PLC Models	1-5		12			
Port	UNIDRIVE SP	1					
PLC Nodels							
Constant of the			-	OK Cancel			
		42					
			CK .	Cancel			

Configuring the Drives

This protocol only supports Ethernet connection.

The Unidrive SP does not have a built-in Ethernet interface. So SM-Ethernet modules are required. The Modbus ID must be set in each drive (parameter 00.37 or 11.23)

The "Reduce SP serial interface priority" parameter in the SM-Ethernet module should be set to "True" (parameter 15.37, 16.37 or 17.37 depending on which slot the SM-Ethernet module is found).

Addressing the Drives

The HMI will address the drives in different ways, depending on their position in the network.

In case the drive to be addressed is attached to the Modbus network and is the master of a CTNet network, it is sufficient to specify its Modbus address.

In case the drive is a CTNet slave, it will require an address depending on its logical position in the network. The 3-digit identifier is composed of the following elements:

- the first number is the Modbus Node ID of the drive master of the CTNet network
- the second number is the slot where the SM application card is plugged-in
- the third number is the CTNet node number of the drive.

When the drive master of the CTNet network has only one SM application unit, the slot information specified into the HMI project is not relevant. In fact, the communication protocol supports an automatic recognition of the slot number; this makes possible to move the SM application board to another slot, maintaining the same configuration at HMI project level.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



Locate the .xml file exported from Tag Editor and click Open.

		· ····································	1		
Irganize 🔻 🛛 New	folder			1955 1955	• 🔳 🤇
_	*	Name		Date modified	Туре
Desktop	8	Exported Tags.xml		03/06/2016 16:42	XML File
🁌 Music	Ш				
Music Pictures Videos Homegroup User Computer Network	m				
Music Pictures Videos Homegroup User Computer Network Control Panel	•				

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags × Protocols					•
+ - 🗸 🕲 🖉 🔰	> & B 副 韵	R 🔎 - Search Tilter by: Data	▼ Ite	ems used:6/10000 Protocol: Show all	🗸 🗹 Show all tags
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			Y Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			 Dictionary 	·
Holding Registers 3	unsignedshort			Array	false
MRTUI	unsignedShort			Array cize	0
- MRTU2	unsignedShort		_	Airdy size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s) . Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: Image: A problem in the p
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the chapter "system variables" about available types and their use.

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General Error	Error cannot be identified; should never be reported; contact technical support

The codes supported for this communication driver are:

Delta Modbus RTU

Delta Modbus RTU communication driver has been designed to connect HMI devices to Delta PLC through Serial connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the $\ensuremath{\text{PLC}}$ list.

The protocol configuration dialog is displayed.

PLC Network		Comm	ОК]
Alias	-	 	Cancel]
Node ID	1	 		
Timeout(ms)	1000			
delay	0			
Num of repeats	2			
Transmission Mode	RTU	•		
PLC Models				
DELTA DVP PLC				

Element	Description			
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.			
Node ID	Serial node associated to the PLC.			
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.			
delay	Time delay in milliseconds between the end of the last received frame and the starting of a new request. If set to 0, the new request will be issued as soon as the internal system is able to reschedule it.			
Num of repeats	Number of times a certain message will be sent to the controller before reporting the communication error status.			
	When set to 1 the panel will report the communication error if the response to the first request packet is not correct.			
Transmission Mode	 RTU: use RTU mode ASCII: use ASCII mode Note: When PLC network is active, all nodes will be configured with the same Transmission Mode. 			
PLC Models	PLC model available:			
	DELTA DVP-PLC			

Element	Description			
PLC Network	IP address for all control enable multiple connecti	lers in multiple connectors.	ctions. PLC Netwo	ork must be selected to
	Delta Modbus RTU	22		
	PLC Network	Comm		
	Alias	Cancel		
	Node ID	Delta Modbus RTU		
	Timeout(ms)	(A)		OK
	delay 0	Alias		
	Num of repeats	Timeout(ms)	1	
	PI C Models	delay		
	DELTA DVP-PLC	Num of repeats	2	
		PLC Models	1	
	Slaves	Add		
	Slave Id Model	Alias		
			ļ	
Comm	If clicked displays the co	mmunication parame	ters setup dialog	
	Comm Parameter Dialog		x	
			ОК	
	Port	com1	•	
	Baudrate	9600	•	
	Parity	none	•	
	Data bits	8	•	
	Stop bits	1	•	
	Mode	RS-485	•	

Element	Description		
	Element	Parameter	
	Port	Serial port selection.	
		• COM1 : On-board port	
		 COM2: Optional Plug-in module plugged on slot#1 or slot#2 	
		 COM3: Optional Plug-in module plugged on slot#3 or slot#4 	
	Baudrate, Parity, Data Bits, Stop bits	Serial line parameters.	
	Mode	Serial port mode. Available modes:	
		• RS-232.	
		• RS-485 (2 wires).	
		• RS-422 (4 wires).	

Tag Editor Settings

In Tag Editor select **Delta Modbus RTU** protocol.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

Delta Modbus RTU				
Memory Type	Offset	subindex		
Input ·	• 0	0 *		
Data Type	Arraysize	Conversion		
boolean	•		+/-	
				炭 (
		OK Cancel	Apply	Help

Element	Description					
Memory Type	Memory Type	Description				
1900	Input	X resources. Corresponding to internal point.	digital Input			
	Output	Y resources. Corresponding to internal digital Output point.				
	Auxiliary Relay	M resources. Corresponding to PLC int	rnal memory.			
	Step Relay	S resources.				
	Timer Contact	T resources.				
	Counter Contact	C resources.				
	Timer Value	TV resources.				
	Counter Value	CV resources.				
	Counter 32bit Value	CV32 resources.				
	Data Register	D resources.				
	Node Override ID	see Special Data Types for mode deta	tails			
Offset	Starting address for the Tag. T	he possible range depend on PLC mode	l selected.			
Subindex	This allows resource offset sele	ection depending on the selected data ty	pe.			
Data Type	Data Type	Memory Space	Limits			
	boolean	1-bit data	01			
	byte	8-bit data	-128 127			
	short	16-bit data	-32768 32767			
	int	32-bit data	-2.1e9 2.1e9			
	int64	64-bit data	-9.2e18 9.2e18			
	unsignedByte	8-bit data	0 255			
	unsignedShort	16-bit data	0 65535			
	unsignedInt	32-bit data	04.2e9			
	uint64	64-bit data	0 1.8e19			
	float	IEEE single-precision 32-bit floating	1.17e-38			

Element	Description	Description					
	Data Type		Memory Space	Limits			
			point type	3.4e38			
	double		IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308			
	string Array of elements containing of selected encoding			haracter code defined by			
	binary		Arbitrary binary data				
	Note: brack	to define array ets like "byte[]"	s. select one of Data Type format followe , "short[]"…	ed by square			
Arraysize	 In case of the second second	f array tag, this f string tag, this in the string tag	property represents the number of array property represents the maximum num g.	elements. ber of bytes			
	Note: number o set to UTF-8 or If Encoding prop character requir	f bytes correspo Latin1 in Tag Eo perty is set to Uo res 2 bytes.	onds to number of string characters if En ditor. CS-2BE, UCS-2LE, UTF-16BE or UTF-1	coding property is I6LE one			
Conversion	Conversion to b	e applied to the	e tag.				
	Conversion						
	inv,swap2	Allowed Configured BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK					
	Depending on data type selected, the list Allowed shows one or more conversion type Value Description						
	Inv bits inv: Invert all the bits of the tag.						
		Example: 1001 \rightarrow 0110 9 \rightarrow 6 (in decin	(in binary format) mal format)				
	Negate	neg: Set the o	pposite of tag value.				

Element	Des

Value	Description
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH	swap4: Swap bytes in a double word.
-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
ABCNOP -	swap8: Swap bytes in a long word.
> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011010000101101000011100101
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
	Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.
Node Override ID

The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.

I

Note: Node Override ID value assigned at runtime is retained through power cycles.

Delta Modbus RTU				
Memory Type	Offset	subindex		
Node Override ID 🔹	0	0 *		
Data Type	Arraysize	Conversion		
unsignedByte 🔹 👻	0		+/-	

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	-	z	ß	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Ta	g URI			

HMIStudio		>
Multiple tag im	porters are available for this protocol. Please select the in	nporter type and continue.
Version	Туре	
DELTA PLC v1.0	Linear	
ag Editor exported x	ml 1.1 General	
/atched dictionary file	::	
/atched dictionary file	2:	
/atched dictionary file	2:	

The following dialog shows which importer type can be selected.

Туре	Description				
DELTA PLC v1.0	Requires a . csv file.				
Linear	All variables will be displayed at the same level.				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				
	Tags ×				
	+ — 🎽 🕲 刘 [< 🕲 🔊 🖌 - +				
	Data				

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				•
🛉 — 👗 🕲 🖉 🔰 🕻 📾 📓	R 🔎 - Search YFilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			Dictionary	
Holding Registers 3 unsignedShort			Dictionary	felse.
MRTU1 unsignedShort			Array	Taise
-MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
MRTU4 unsignedShort			Comment	
MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description			
	Import Tag(s).			
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project			
₩.	Update Tag(s).			
	Click on this icon to update the tags in the project, due a new dictionary import.			
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:			
P - Search	Searches tags in the dictionary basing on filter combo- box item selected.			

Direct Serial

Direct Serial communication driver is a generic protocol that allows low level access to serial functions.

Using this protocol the application itself can realize some serial based protocol (RS-232/485/422) without requirement for a development of a dedicated protocol.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Direct Serial		×
	Comm	ОК
Rx Token Prefix		Cancel
Rx Token Suffix		
Token Gap	0	
Tx Token Prefix		
Tx Token Suffix		
Hexadecimal Tokens		
Token Queue Size	100	
ConnectAtPowerUp		
PLC Models		
default		

Element	Description
Rx Token Prefix	Indicates the prefix for read token, as string specified by hexadecimal characters.
Rx Token Suffix	Indicates the suffix for read token, as string specified by hexadecimal characters.
Token Gap	Indicates the period between tokens, in milliseconds.
Tx Token Prefix	Indicates the prefix for sent token, as string specified by hexadecimal characters.
Tx Token Suffix	Indicates the suffix for sent token, as string specified by hexadecimal characters.
Hexadecimal	checked = tokens are in hexadecimal
IOKENS	not checked = tokens are not in hexadecimal

Token Queue Size Indicates the number of tokens in the queue, as an integer value from 1 to 10000 (default: 100)



These parameters are determining the behavior of the driver during RX and TX operations, as defined in next paragraphs. In addition the standard communication parameters are available.



All protocols parameters can be overwritten at runtime using the appropriate memory types, so the complete setup can be achieved during runtime using Tags. Settings using memory types are saved to permanent storage using standard procedures. The "Serial Done" memory type is used in order that all set parameters are transferred to usage at once. If any of the serial parameter is changed the serial driver is re-programmed.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Direct Serial** from the protocol list: tag definition dialog is displayed.

Memory Type		nada		Data Turca	
Token To Send	-	node		string	
Arraysize	Conversion	ı			
0		+	-/-		

Element	Description					
Memory Type	Name	Datatype	Description			
1900	Token To Send	string	Write only. Writing given string to cor	g on this memory type sends the nmunication.		
	Token Received	string	Read only. Readir the front token fro	ng from this memory type gets m the receiving queue.		
	Length of Token Received	unsignedInt	Read only. Return token from the rec	ns the length in bytes of the front ceiving queue.		
	Tokens Available	unsignedInt	Read only. Gives receiving queue.	the number of tokens in the		
	Token Acknowledge	boolean	Write only. Writing to this memory type removes the front token from the receiving queue.			
	Serial Baudrate	unsignedInt	Overrides serial baudrate parameter.			
	Serial Bits	unsignedByte	Overrides serial bits parameter.			
	Serial Stop Bits	unsignedByte	Overrides serial s	top bit parameter.		
	Serial Parity	unsignedByte	Overrides serial parity parameter.			
	Serial Mode	unsignedByte	Overrides serial m	node parameter.		
	Rx Token Prefix	string	Overrides protocol parameters. Check " <i>Protoco</i> <i>Editor Settings</i> " from details.			
	Rx Token Suffix	string				
	Token Gap	unsignedInt				
	Tx Token Prefix	string				
	Tx Token Suffix	string				
	Hexadecimal Tokens	boolean	-			
	Token Queue Size	unsignedInt				
	Serial Done	boolean	Writing to this mer values written in th parameters, and t	mory type transfers all new ne other tags to protocol o permanent storage.		
Data Type	Data Type	Memory Space		Limits		
	boolean	1-bit data		01		
	unsignedByte	8-bit data		0 255		

Element	Description						
	Data Type		Memory Space	Limits			
	unsignedInt		32-bit data	0 4.2e9			
	string		Array of elements containing character code defined by selected encoding				
	Note: to defir []", "short[]"	Note: to define arrays. select one of Data Type format followed by square brackets like "by []", "short[]"					
Arraysize	In case of array t	ag, this	property represents the number of array	elements.			
	 In case of string string tag. 	 In case of string tag, this property represents the maximum number of bytes available in the string tag. 					
	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.						
Conversion	Conversion to be applied to the tag.						
	Conversion						
	inv,swap2 Allowed Configured						
		A >CDAB FGH->GHEFCDAB S Cancel OK					
	Depending on data type	e selecte	ed, the list Allowed shows one or more o	conversion types.			
	Value	Descri	ption				
	Inv bits	inv: Inv	ert all the bits of the tag.				
		Example 1001 - 0 9 - 0 (10)	le: ▶ 0110 (in binary format) in decimal format)				
	Negate	neg: Se	et the opposite of tag value.	_			
		Examp 25.36 –	le: → -25.36				
	AB -> BA	swapn	i bbles : Swap nibbles in a byte.				

Element	Description		
	Value	Description	
		<i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	
	ABCD -> CDAB	swap2: Swap bytes in a word.	
		<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)	
	ABCDEFGH ->	swap4: Swap bytes in a double word.	
	GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)	
	ABCNOP -> OPMDAB	swap8: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011010001011101000011100000	
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)	
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	Select conversion and	click +. The selected item will be added to list Configured .	
	If more conversions ar	e configured, they will be applied in order (from top to bottom of list Configured).	

Use the arrow buttons to order the configured conversions.

Implementation Details

Receiving algorithm

The protocol applies a separate thread that receives the characters from specified serial port.

When tokens (substrings) are identified they are put into the receiving queue (as strings).



Both ASCII and binary mode are available. When binary data can be present into receiving stream the **Hexadecimal Tokens** parameter can be set. In this case tokens are stored in queue using hex string coding (each byte is stored using two chars representing the hex value 0 to F). When defining the tags used to read tokens the appropriate string length should be computed considering the binary mode.

The **Token Queue Size** parameter specifies the maximum number of tokens saved into the queue. When the queue becomes full the oldest token is discarded.

The token identification is as follows:

- if the parameters specify a rx-prefix all characters before detecting the prefix are ignored
- if protocol specifies a rx-suffix it is used to detect the token end
- if rx-suffix is specified the parameter 'gap' specifies the timeout after which the token receiving is restarted
- if rx-suffix is not specified the parameter gap specifies the timeout that terminates the token (anything received up to this interval). If within this time the rx-prefix is detected again the token is ended and stored and reception of a new token is started

In summary we can have four combinations:

- a. No rx-prefix and rx-suffix: the incoming stream is divided in tokens according to gap detection
- b. Rx-prefix specified but no suffix: all the received chars before prefix are ignored. All the chars after prefix are stored in a token till the gap detection
- c. Rx-prefix and Rx-suffix specified: all the chars between prefix and suffix are stored in a token. All the chars received before prefix or after suffix till the gap detection or till a new prefix are ignored
- d. Rx-suffix specified but not RX-prefix: all the chars received till suffix are stored in a token. All the chars received after suffix till the gap detection are ignored

The rx-prefix and rx-suffix parameters are specified as hex strings, so any characters can be specified (like DLE STX CR LF etc...). i.e. to define the string "STR" as prefix the string "535452" must be used.

Before putting string to the receiving queue the prefix and suffix are removed (only 'payload' saved).

Transmission algorithm

The strings to be transmitted are prepared adding the "Tx-prefix" in front and the "Tx-suffix" in the end, if defined. Then the whole string is transmitted immediately.

Interface to user project

Reading a tag defined as **Token Received** gets the front string from the queue. If there are no new tokens an empty string is returned.

Reading a tag defined as Length of Token Received gets the length in bytes of the token.

Reading a tag defined as Tokens Available gets the number of tokens currently stored in the queue.

Writing to a tag defined as **Token Acknowledge** removes the token from queue and makes available the next token if present.

Writing to a tag defined as Token To Send means immediate sending, without any queue used.

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

This interface provides the access to token queue and sending function. The following commands are supported:

Command	Description
put	Put the token to send contained in string parameter.
get	Get the received token.
get_token_length	Get the length of received token.
tokens_available	Get number of tokens received.
token_ack	Acknowledge reading token.

Using the command interface the following JS code should receive data:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var avail = tagMgr.invokeProtocolCommand(protID, "tokens_available", "");
while (parseInt(avail) > 0)
{
        var str = tagMgr.invokeProtocolCommand(protID, "get", ""); // get the next
token
        var status = tagMgr.invokeProtocolCommand(protID, "token_ack",""); //
acknowledge current token
        avail = tagMgr.invokeProtocolCommand(protID, "tokens_available",""); // get
number of available tokens in queue
}
```

VCS access

The protocol supports the remote (virtual com port) access in exclusive mode.

When VCS is enabled the serial line usage is suspended and serial line becomes available for remote user. At the end the protocol is restarted. The content of the token queue is lost.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



Locate the Tag Editor Exported symbol file and click **Open**.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags × Protocols			-
+ - 👗 🕲 🗈 🔰 🗅 🕏	B D Filter by: Dat	a v Items used:6/10000 Protocol: Sho	w all 🛛 🗹 Show all tags 🖉 🗖
Data Type	Comment	Property	Value
Modbus TCP:prot1	iner	✓ Driver	
Model: Modicon Modbus(1-based)		Model	Modicon Modbus(1-based)
Holding Registers 1 Unsign	hedShort	Protocol	Modbus TCP:prot1
Holding Registers 2 unsign	nedShort	✓ Dictionary	
- MRTI I unsign	aedShort	Array	false
- MRTU2 unsign	hedshort	Array size	0
MRTU3 unsign	nedShort	Arrayindex.Subindex	400003
MRTU4 unsign	nedShort	Comment	
- MRTU5 unsign	nedShort	Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops X + - X X + X X + X X + X X + X X + X X + X X + X X
P - Search Y Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo- box item selected.

Direct Socket

Direct Socket protocol is a generic protocol that allows low level access to socket functions.

Using this protocol the application itself can realize some IP based protocol without requirement for a development of a dedicated protocol.

Direct Socket protocol can be used as a standard (tag interface) protocol but also there is the appropriate implementation of DoCommand interface to enable using protocol from JavaScript.

The protocol can be used only with client socket type.

The protocol supports just one client socket. In case that application requires many sockets there could be many protocols installed, as the protocol supports multi-instance.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the PLC list.

The protocol configuration dialog is displayed.

Socket type	UDP	•	Cancel
Remote IP address	127.0.0.1		
Remote port	0		
Local IP address	0.0.0.0		
localPort	0		
Broadcast type	Global	•	
Rx Token Prefix			
Rx Token Suffix			
Token Gap	0		
Tx Token Prefix			
Tx Token Suffix			
Hexadecimal Toke	ns		
Token Queu <mark>e</mark> Size	100		
PLC Models			
default			

Protocol parameters define a way how the connection is set and how the tokens are exchanged. The parameters are generally defined by the project. Many parameters can be accessed also as variables, allowing the runtime changes.

Element	Description
Socket type	Type of socket used for communication. Possible choices are UDP or TCP.
Remote IP Address	String. Indicates the IP address of remote device.
Remote Port	Integer. Indicates the port used by remote device.
Local IP Address	String. Indicates the IP address of local device. Mandatory for UDP usage.

Element	Description
Local Port	Integer. Indicates the port used by local device. Mandatory for UDP usage.
Broadcast Type	Type of broadcast used. Possible choices are Global or Local.

The following parameters are determining the behavior of the driver during RX and TX operations, as defined *Implementation Details* chapter.

Element	Description	
Rx Token Prefix	Indicates the prefix for read token, as string specified by hexadecimal characters.	
Rx Token Suffix	Indicates the suffix for read token, as string specified by hexadecimal characters.	
Token Gap	Indicates the period between tokens, in milliseconds.	
Tx Token Prefix	Indicates the prefix for sent token, as string specified by hexadecimal characters.	
Tx Token Suffix	Suffix Indicates the suffix for sent token, as string specified by hexadecimal characters.	
Hexadecimal	checked = tokens are in hexadecimal	
Tokens	not checked = tokens are not in hexadecimal	
Talan Olar Olar		

Token Queue Size | Indicates the number of tokens in the queue, as an integer value from 1 to 10000 (default: 100)

All protocols parameters can be overwritten at runtime using the appropriate memory types, so the complete setup can be achieved during runtime using Tags. Settings using memory types are saved to permanent storage using standard procedures. The "Done" memory type is used in order that all set parameters are transferred to usage at once. If any parameter is changed the driver is re-programmed.

Tag Editor Settings

ñ

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Direct Socket** from the protocol list: tag definition dialog is displayed.

Memory Type	Data Type	Arraysize	
Token To Send	 string	 0	
Conversion	 		

Element	Description			
Memory Type	Name	Datatype	Description	
	Token To Send	string	Write only. Writing on this memory type sends the given string to communication.	
	Token Received	string	Read only. Reading from this memory type gets the front token from the receiving queue.	
	Length of Token Received	unsignedInt	Read only. Returns the length in bytes of the front token from the receiving queue.	
	Tokens Available	unsignedInt	Read only. Gives the number of tokens in the receiving queue.	
	Token Acknowledge	boolean	Write only. Writing to this memory type removes the front token from the receiving queue.	
	Connect	boolean	Write only. Writing 1 to this variable enables the connection.	
	Connection Status	boolean	Read only. Gives the status of the connection In TCP mode it reflects effective connection with the peer. In UDP mode it is TRUE as soon as Connect is TRUE	
	Socket type	string	Overrides protocol parameters. Check	
	Remote IP Address	string	"Protocol Editor Settings" from details.	
	Remote Port	unsignedShort		
	Local IP Address	strgin		
	Local Port	unsignedShort		
	Broadcast Type	string		
	Rx Token Prefix	string		
	Rx Token Suffix	string		
	Token Gap	unsignedInt		
	Tx Token Prefix	string		
	Tx Token Suffix	string		
	Hexadecimal Tokens	boolean		
	Token Queue Size	unsignedInt		
	Done	boolean	Writing to a tag of this memory type transfers all new values written in the other tags to protocol parameters, and to permanent storage.	

Element	Description				
Data Type	Data Type	Memory Space	Limits		
	boolean	1-bit data	01		
	unsignedByte	8-bit data	0 255		
	unsignedShort	16-bit data	0 65535		
	unsignedInt	32-bit data	04.2e9		
	string	Array of elements containing character encoding	ray of elements containing character code defined by selected acoding		
	Note: to define arrays. select one of Data Type format followed by square brackets li []", "short[]"				
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 				
	Note: number of bytes corresponds to number of string characters if Encoding property is set to U or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character require bytes.				
Conversion	Conversion to be applied	to the tag.			
	Conversion				
	inv,swap2	Allowed GCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB inv bits Cancel OK			

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Element	Description		
	Value	Description	
	Inv bits	inv: Invert all the bits of the tag.	
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)	
	Negate	neg: Set the opposite of tag value.	
		<i>Example:</i> 25.36 → -25.36	
	AB -> BA	swapnibbles: Swap nibbles in a byte.	
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)	
	ABCD -> CDAB	swap2: Swap bytes in a word.	
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)	
	ABCDEFGH ->	swap4: Swap bytes in a double word.	
	GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)	
	ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010	
	BCD	bcd: Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	

Select conversion and click +. The selected item will be added to list **Configured**.

Element	Description
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Implementation Details

Principle of operation

Protocol is parameterized by number of protocols parameters. The parameters define which socket type is used and the host address.

The data access is based on 'tokens'. Token is data string that can be surrounded by prefix and suffix.

The protocol receiving process reads data from the specified IP/port and identifies tokens. Identified tokens are put to the queue from where they can be read by application. In the sending direction the application writes the token to protocol.

Protocol adds the defined tx_prefix/tx_suffix and sends data to the defined host.

Token extraction

The token extraction is slightly different for UDP and TCP sockets.

UDP protocols starts searching for tokens at the start of the received datagram. The search ends at the datagram end. If no rx_prefix is specified the token starts at datagram start. If no rx_suffix is specified the token ends on the datagram end. By specifying neither prefix nor suffix the whole datagram is delivered as a token. When both prefix and suffix are specified there can be many tokens extracted from a single datagram.

TCP protocol starts searching for tokens immediately after the previous rx_prefix. The search ends either when suffix is found or if the time gap without data is detected. If neither prefix nor suffix is specified the tokens will be all received data separated by time gaps.

The tokens can be plain ASCII strings, or hexadecimal strings. This is defined by the parameter 'hex_tokens'.

The prefix/suffix strings must always be in hexadecimal format.

Common behavior

Both ASCII and binary mode are available. When binary data can be present into receiving stream the **Hexadecimal Tokens** parameter can be set. In this case tokens are stored in queue using hex string coding (each byte is stored using two chars representing the hex value 0 to F). When defining the tags used to read tokens the appropriate string length should be computed considering the binary mode.

The **Token Queue Size** parameter specifies the maximum number of tokens saved into the queue. When the queue becomes full the oldest token is discarded.

The token identification is as follows:

- if the parameters specify a rx-prefix all characters before detecting the prefix are ignored
- if protocol specifies a rx-suffix it is used to detect the token end
- if rx-suffix is specified the parameter 'gap' specifies the timeout after which the token receiving is restarted

• if rx-suffix is not specified the parameter gap specifies the timeout that terminates the token (anything received up to this interval). If within this time the rx-prefix is detected again the token is ended and stored and reception of a new token is started

In summary we can have four combinations:

- a. No rx-prefix and rx-suffix: the incoming stream is divided in tokens according to gap detection
- b. Rx-prefix specified but no suffix: all the received chars before prefix are ignored. All the chars after prefix are stored in a token till the gap detection
- c. Rx-prefix and Rx-suffix specified: all the chars between prefix and suffix are stored in a token. All the chars received before prefix or after suffix till the gap detection or till a new prefix are ignored
- d. Rx-suffix specified but not RX-prefix: all the chars received till suffix are stored in a token. All the chars received after suffix till the gap detection are ignored

The rx-prefix and rx-suffix parameters are specified as hex strings, so any characters can be specified (like DLE STX CR LF etc...). i.e. to define the string "STR" as prefix the string "535452" must be used

Before putting string to the receiving queue the prefix and suffix are removed (only 'payload' saved).

Interface to user project

Reading a tag defined as **Token Received** gets the front string from the queue. If there are no new tokens an empty string is returned.

Reading a tag defined as Length of Token Received gets the length in bytes of the token.

Reading a tag defined as Tokens Available gets the number of tokens currently stored in the queue.

Writing to a tag defined as **Token Acknowledge** removes the token from queue and makes available the next token if present.

Writing to a tag defined as Token To Send means immediate sending, without any queue used.

Data traffic control

The TCP sockets can be controlled by variables "Connect" and "Connection Status". If the bool variable "Connect" is set the protocol will permanently try to make the connection to the specified host. If the TCP connection breaks it will be reestablished automatically. If the variable "Connect" is false the protocol will wait. The state of connection can be read by variable Connection Status".

For UDP there is no connection control. The socket is always connected and sends/receives data.

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

This interface provides the access to token queue and sending function. The following commands are supported:



Command	Description		
set_ip_port <ipaddress> <port></port></ipaddress>	Specify the remote IP/port couple to use for connection.		
	If protocol is already connected it is disconnected from current peer and re- connected to new one.		
	Example of usage in JavaScript:		
	<pre>var tagMgr = project.getWidget("_TagMgr"); var protID = "prot2"; // to be set according to protocol numbering tagMgr.invokeProtocolCommand(ProtID,"set_ip_ port","127.0.0.1 502");</pre>		
connect <on off></on off>	Enables/disables the connection.		
get_stat	Status of connection <connected disconnected>.</connected disconnected>		
put <string></string>	Put the token to send contained in string parameter.		
get	Get the received token.		
get_token_length	Get the length of received token.		
tokens_available	Get number of tokens received.		
token_ack	Acknowledge reading token.		

Using the command interface the following JS code should receive data:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var avail = tagMgr.invokeProtocolCommand(protID, "tokens_available", "");
while (parseInt(avail) > 0)
{
    var str = tagMgr.invokeProtocolCommand(protID, "get", ""); // get the next
token
    var status = tagMgr.invokeProtocolCommand(protID, "token_ack",""); //
acknowledge current token
    avail = tagMgr.invokeProtocolCommand(protID, "tokens_available",""); // get
number of available tokens in queue
}
```

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	Z	đ	ß	>]	₽	A 9B	B>	ŧ3	1
Data	1		^		-	Та	g URI			

Locate the Tag Editor Exported symbol file and click **Open**.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags × Protocols					~
+ - 🗸 🕲 🖉 🔰	> \$ ⊮ ₪ 🕅	R P- Search TFilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 🔅 🗖
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
- MRTU1	unsignedShort			Array	false
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort		_	Data type	unsignedShort

Toolbar item	Description				
	Import Tag(s).				
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project				
督	Update Tag(s).				
	new dictionary import.				
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:				
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.				

DMX512 Digital Multiplex

This document describes and specifies the implementation of DMX512 Digital Multiplex communication driver.

Purpose of implementation is to allow driving up to 512 channels connected to a RS485 serial line, or to merge additional channels, or to overwrite existing channels to an existing DMX controller.

Possible topologies

Normal mode



In normal mode only Tx signal of the serial line is connected.

Merge mode



In merge mode the existing serial line must be opened and the origin line must be connected to Rx input.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the PLC list.

The protocol configuration dialog is displayed.

Port	com1	•	Cancel
Number of Channels	512	×	
Frame Delay (ms)	100		
Merge Mode			
PLC Models			

Element	Description
Port	COM port to be used. Serial line parameters are fixed.
Number of Channels	1 - 512. Defines the number of channels transmitted in the multiplex flame.
Frame Delay (ms)	10 - 1000. Defines inter-frame delay to adapt to specifications of slaves. Delay is applied at the end of frame so the real frame rate is determined by formula: (approx) <i>Time (microsec)</i> = $120 + 20 + 40 \times (nr \text{ of channels}) + Frame Delay * 1000$
Merge Mode	Selects the Merge Mode in which the unit receives a frame from an external controller and substitutes the values of some of the channels or add other channels in the end of the frame
PLC Models	Only "default" is available.

Tag Editor Settings

In Tag Editor select DMX512 Digital Multiplex protocol.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

DMX512 Digital Multiplex	
memtype	index
channel -	1 0 ~
datatype	
short 👻	

Each channel can be assigned to a Tag.

Element	Description					
memtype	Memory Type Description					
	channel	Only available memory type.				
index	Refer to channel number to po	Refer to channel number to point to.				
datatype	Data Type Memory Space Limits					
	short	16-bit data -32768 32767				

Channel behavior

Only available DataType is short (signed 16-bit data) so a Tag can assume values from -32768 to 32767. Anyway the protocol uses only values from 0 to 255.

Other values are used in Merge Mode: when the channel overwrites an existing channel the negative values are used to disable overwriting.

Value	Normal Mode	Merge Mode
0 to 255	0 to 255	0 to 255
> 255	255	255
< 0	0	original value of channel in the incoming frame

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



It is possible to import a Tag Editor exported xml

Туре	Description
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	Tags ×
	+ - 🗸 🕲 🔊 🚺 🚺 🖬 🖬

Once the importer has been selected, locate the symbol file and click **Open**.

Data

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tag URI

Tags × Protocols					
+ - 👗 🕲 🖉 👌 🕨	\$B ₪} ∰	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🔅 🗖
Data Ty	ype	Comment	^	Property	Value
Modbus TCP:prot1	ontainer			✓ Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1 un	nsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 un	nsignedShort			✓ Dictionary	
Holding Registers 3 un	nsignedShort			and a second sec	felas.
MRTU1 un	nsignedShort			Array	Taise
-MRTU2 un	nsignedShort			Array size	0
MRTU3 un	nsignedShort			Arrayindex.Subindex	400003
MRTU4 un	nsignedShort			Comment	
-MRTU5 un	nsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
蘭	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Tags* Tags* <th< th=""></th<>
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Eaton Suconet-K

The Eaton Suconet-K communication driver has been designed to connect HMI devices to a Suconet-K network with a Möeller PLC.

Protocol Editor Settings

			ОК
Port	com1	•	Cancel
slave ID	2	<u>*</u>	
Input Buffer Size	6	*	
Output Buffer Size	7	(A.)	
timeout	3000	<u>*</u>	
PLC Models			
PS4 PS341 PS306-316 PS416			

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
Port	Serial port selection.
	COM1: device PLC port.
	 COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board.
	 COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on- board.
slave ID	node of the slave device.
Input Buffer Size	Size of Input Buffer. Input data length must be exactly the same as in PLC configuration.
Output Buffer Size	Size of Output Buffer. Output data length must be exactly the same as in PLC configuration.
timeout	Time delay in milliseconds between two retries in case of missing response from the server device.
PLC	Two PLC models are available:
Models	• PS4
	• PS341
	• PS306-316
	• PS416

Tag Editor Settings

In Tag Editor select the protocol Eaton Suconet-K.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

Eaton Suconet_K			
Memory Type	Offset	SubIndex	
Internal Relay	• 0	0 -	
Data Type	Arraysize	Conversion	
boolean 👻	0		+/-

Element	Description		
Memory Type	Memory Type Description		
туре	Internal relay	Internal memory of PLC. It can be addressed using Offset and Data Type.	
Offset	Starting address for the Tag. The possible range depend on PLC model selected.		
SubIndex	This allows resource offset selection depending on the selected data type.		
Data Type	Available data types: • boolean • byte • short • int • unsignedByte • unsignedShort • unsignedInt • float • float • double • string • binary See "Programming concepts" so Note: To define array brackets (byte[], sho	section in the main manual. ys, select one of Data Type format followed by square rt[]).	
	See "Programming concepts" s Note: To define array brackets (byte[], sho	section in the main manual. ys, select one of Data Type format followed by square rt[]…).	

Element	Description		
Arraysize	 In case of In case of available Note: number of set to UTF-8 or If Encoding proping character required 	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. f Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 	
Conversion	Conversion to be applied to the tag.		
	inv,swap2	Allowed Configured BCD Inv bits	
		AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK	
	Depending on data type selected, the list Allowed shows one or more conversion Value Description		
	Inv bits	inv: Invert all the bits of the tag. <i>Example:</i> 1001 - 0110 (in binary format)	
		$9 \rightarrow 6$ (in decimal format)	
	Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte.		
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)	
	ABCD ->	swap2: Swap bytes in a word.	
	CDAB	<i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)	
	ABCDEFGH	swap4: Swap bytes in a double word.	

Element	Description	
	Value	Description
	-> GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ $0001\ -\rightarrow$ $1\ 10000011100$ 1010101000010110110110110110010110110000
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion If more convers Configured).	on and click +. The selected item will be added to list Configured . ions are configured, they will be applied in order (from top to bottom of list

Use the arrow buttons to order the configured conversions.

Environment Variables

This protocol gives the possibility to copy the environment variables of the hosting Operative System inside tags. All variables will be read only, namely, is not possible to modify them.



Environment Variables communication driver is not counted as physical protocol. Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In the **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the Environment Variables protocol from the PLC list.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select Environment Variables from the protocol list: tag definition dialog is displayed.

Environment Variables				×
Environment Variables				
Data Type string ~	Arraysize	Conversion	+/-	
Name PATH				
	ОК	Cancel	Apply	Help

Element	Description
Name	Name of the environment variable that you want to read.
Data Type	System variables are of type string, but if a different type is chosen, e.g. int, casting to the chosen type will be made.
Arraysize	This property represents the maximum number of bytes available in the string or in the array Tag.

Ethernet/IP CIP

The protocol has been implemented according to the published Ethernet/IP specifications (available from www.odva.org).

The Ethernet/IP CIP driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. Although the Ethernet/IP CIP driver is fast, we suggest to use short Tag names. Tags are read from and written to the device by specifying their symbolic name in the communications request, therefore the longer the tag name is, the larger the request will be.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Ethernet/IP CIP		×
PLC Network		ОК
Alias		Cancel
IP Address	0.0.0.0	
Slot	0	
Timeout	1000	
PLC Models		
Logix 5000		
Micro800 Omron NJ/NX Serie		
Omron CJ Serie		

Field	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP Address	Ethernet IP address of the controller.
Slot	CPU slot number for Logix 5000 models (typically 0). Refer to the controller documentation for further details.

Field	Description		
PLC Models	PLC model used to import tags file.		
PLC Models PLC Network	Enable access of option.	to multiple networked controllers. For every controller (slave) set the proper	

Controller Model Logix 5000

The Ethernet/IP CIP driver allows to connect Allen-Bradley ControlLogix and CompactLogix Ethernet controllers.

Communication with ControlLogix® 5500 controllers can be accomplished through an Ethernet/IP communication module for Ethernet such as the 1756-EN2T or 1756-ENET.

Ethernet communication with CompactLogix[™] 5300 controllers requires a processor with a built-in Ethernet/IP port such as the 1769-L32E.

All trademarks are the property of their respective owners.

The internal memory organization of the Logix CPUs is not fixed but configured by the user at development time. Each data item can be identified by a string called "Tag". The RSLogix 5000 software can then export to the application the list of Tags created for each controller.

The project loaded on the HMI device must refer to Tag names assigned in RSLogix 5000 software at development time. The Tag Editor supports direct import of the Tag file generated by RSLogix 5000 software in .CSV format.

The implementation of the Ethernet/IP driver also supports access to structured data types which can be imported from .L5X files.

The driver supports access to both Controller and Program Tags.

Export CSV and L5X files using RSLogix5000

To export the .CSV Tag file:

- 1. From the Controller Organizer pane, right-click on Controller Tags.
- 2. Select Export Tags: the Export dialog is displayed.

	👻 🌉 🌺 🕼 🕼 🖉 🔍 🔍 Select a Language 👻 🧶
Forces FOK Edits A FUN	Path (rome) ▼ H hat ++ N Favorities Add-On Add-On A Safety A James A Safety
Controller Organizer	
Concoller au Mew Tag Ctrl+W Power-Up Har MainTask MainTask MainTask MainProgr Unscheduled Deport Tags Motion Group: Print Add-On Instructions Data Types Mule-Defined Strings Add-On-Defined Predefined Module-Defined Trends U/O Configuration Backplane, CompactLogix System 1769-132E Hyperget 1769-132E Hyperget 1769-132E Hyperget 1769-132E Hyperget	File name: My_project-Controller Tags Export Save at Inf Same at Inf Save at Inf Save at Inf Save at Inf Save at Inf
∰ CompactBus Local	Tags: 3 Controller Logic Comment: All Programs Phases Controller Programs Programs Pagement Phases - Sappment Phases - Add On Instructions

- 3. Choose **All** from the **Tags** list to export all Tags.
- 4. Select the Save as type option to .CSV.
- 5. Click **Export**: all the Tags are exported to an **.CSV** file.

To export the .L5X data type file:

- 1. Choose File > Save As.
- 2. Select the Save as type option to .L5X.
- 3. Click **Save**: all the Tags are exported to an **.L5X** file.
| lew Ctri
Ipen Ctri
Jose | •O
Path: <none></none> | ,# 4, 72 () □ | 1 📴 ℚ. Q. Select a
▼ 🔡 | Languaga 🔻 🥪 | |
|--|---------------------------|--|---|----------------------------------|-----------------|
| ave Ctrl
ave As | SHENE | + + + + + () + (0)+
Add-On ⋌ Safety ⋌ A | -(∟)- ►
larms 🔏 Bt 🔏 Timer/C | | |
| le <u>w</u> Component
mport Component | ,
Be Sav | ve As | | | |
| ompact | | Save in: | | - 0 0 🕫 💷 | |
| age Setup
ienerate Report
rint
rint Options | | LE | - | | |
| My_project.ACD
Libreria_Rockwell_Cimplicity_L63_EXOR.ACD
Libreria_Rockwell_Cimplicity_L63.ACD | | | and Tamas | | |
| st Weit Add-On-Defined Predefined Module-Defined Trends UO Configuration To Configuration To 1769-132E My_project To 1769-132E My_project Configuration Configura | | * | | | |
| 다 유 Ethernet | | File name:
Save as typ | My_project
e: RSLogix 5000 Project
RSLogix 5000 Project | File (* ACD)
File (* ACD) | Save Cancel |
| | | Lincode | ASLOGA 5000 kmp of / | Export File (*1.5K)
e (*1.5K) | Нер |

Import Files in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers are	e available for this protocol. Please select the importer type and continue.
Version	Туре
Allen-Bradley RSLogix5000 v 1. 1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Select Allen-Bradley RSLogix5000 v15 option.

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags 🗙 Protocols					▼
+ - 👗 🕲 👂 🔰	[> € B ₪ 🕅	R 🔎 - Search Tilter by: Data	▼ Ite	ems used:6/10000 Protocol: Show a	all 💽 Show all tags 🔅 🗔
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	. Container			Y Driver	,
Model: Modicon Modbus(1-bas	ed) Container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			Y Dictionary	
Holding Registers 3	unsignedShort			Dictionary	(1)
- MRTU1	unsignedShort			Array	faise
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.

Toolbar item	Description
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Note: When importing the array data types, the importer is expanding them creating individual Tags per each array element; this is valid for all the data types, except for arrays of boolean. In this case they are imported as "boolean-32" and the single array element can be addressed using "Tag Index" parameter from "Attach to…" dialog.

Module-Defined and User-Defined data types

RSLogix 5000 allows you to define Tags with several data types.

Data type group	Description
Predefined	Standard data types such as BOOL, DINT, SINT, INT and other less common data types such as PID, COUNTER, TIMER.
Module-Defined	Data type associated with I/O optional modules usually referenced by aliases.
User-Defined	Custom data type defined by user

In order to import Predefined (with the exception of standard data types which are always imported) and Module-Defined data type you need to edit the ETIPSpecialDataTypes.xml file located under *languages\shared\studio\tagimport* or *studio\tagimport* depending on installed version.

In RSLogix5000 software:

Ð.

1. From the Controller Organizer pane, select Controller Tags.

RSLogix 5000			
🖉 File Edit View Search Logic Comn	nunications Tools Window Help		_ 8 ×
🖀 🖷 🖶 🏶 🎍 🏙 につい	- 🏕 🕰 📴 🛛 💇 🔍	Select a Language	9
Offline 🛛 🗸 🗏 RUN	Path: <none></none>	▼ 品	
No Forces	-9		
No Edits	▲ ⊢ ⊢ ⊨ ⊨ ↔ ↔ ↔ ↔	F	
Redundancy 5.0	Favorites Add-On A Safety A Alarms A Bit	Timer/Co	
Controller Organizer 🗸 🗣 🗙	Scorer 101 51 PCS Show Al Taga	🔹 😿 Enter Description	r Filer
Controller G1 PCS		In a Tau	
- Ontroller Tags	Name I DEC DEC	Data Type	Style ^ 2
- Controller Foult Andler		TIMER	
- 🗁 Power-Up Handler		TIMEN	Daviaal
🖨 🔄 Tasks	HTS_ALM_TWH_UE	8000	Decimal
⊕- □ → MainTask ☐		POOL	Decimal
HTS_BK_CUNF_CTL_DE		ROOL	Decimal
Recipe_Handling		8001	Decimal
B-Control	HYS BKT CONV HME TO SBY OF	8001	Decimal
- Unscheduled Programs / Phases	E-HVS BKT DIST 1 2 CALC SP	DINT	Decimal
H 🗀 Motion Groups	HYS BKT DIST 1 3 CALC SP	DINT	Decimal
🛓 📇 Add-On Instructions	HYS BKT HME DI	BOOL	Decimal
🛓 🕼 Jog_Axis	HYS BKT HME DIST 1 2 OFST SP	DINT	Decimal
🖨 🔄 Data Types	HYS_BKT_HME_DIST_1_3 OFST_SP	DINT	Decimal
🕀 🕞 User-Defined	±-HYS_BKT_LNG_SP	DINT	Decimal
🗄 🖳 Strings	E-HYS_BKT_MAM_ACC_SP	DINT	Decimal
Add-On-Defined E HYS_BKT_MAM_HGH_ACC_SP		DINT	Decimal
Predefined T-HYS_BKT_MAM_HGH_JRK_SP		DINT	Decimal
Transfer		DINT	Decimal
In 1/0 Configuration		DINT	Decimal
	Monitor Tags Edit Tags	<	• a
Enter a tag name			-

2. Filter tags to display only **Module-Defined** Tags.

Define Tag Filter	-
Filter On:	ОК
<all> ▼</all>	Cancel
Include Data Types:	
⊕-User-Defined	Clear Filter
☐ ⊕ Strings ☐ ⊕ Add-On-Defined	
	1
	J
	Help

Only tags (alias) with data type belonging to optional I/O Modules will be displayed.

Scope: 🛐 G1_PCS → Show: AB:1734_12SL0T:1:0, AB:1734_12SL0T:0:0, → 🔽 Enter Description Filter						
Name	Data Type 🔚 🛆	Style 🔺				
HYS_Point_I0_Rack_20:1	AB:1734_3SLOT:1:0					
HYS_Point_I0_Rack_20:0	AB:1734_3SLOT:0:0					
HYS_Point_I0_Rack_1:I	AB:1734_13SLOT:1:0					
HYS_Point_I0_Rack_1:0	AB:1734_13SLOT:0:0					
HYS_Point_I0_Rack_1:2:C	AB:1734_D18:C:0	E				
HYS_Point_I0_Rack_1:3:C	AB:1734_D18:C:0					
HYS_Point_IO_Rack_1:4:C	AB:1734_D18:C:0					
HYS_Point_IO_Rack_1:5:C	AB:1734_D18:C:0					
HYS_Point_I0_Rack_1:6:C	AB:1734_D18:C:0					
HYS_Point_IO_Rack_1:7:C	AB:1734_D18:C:0					
HYS_Point_I0_Rack_1:8:C	AB:1734_D18:C:0					
HYS_Point_I0_Rack_20:1:C	AB:1734_D18:C:0					
HYS_Point_IO_Rack_1:9:C	AB:1734_D08_NoDiag:C:0					
HYS_Point_I0_Rack_1:10:C	AB:1734_D08_NoDiag:C:0					
HYS_Point_I0_Rack_1:11:C	AB:1734_D08_NoDiag:C:0					
HYS_Point_I0_Rack_1:12:C	AB:1734_D08_NoDiag:C:0					
HYS_Point_I0_Rack_20:2:C	AB:1734_D08_NoDiag:C:0					
HYS_Point_I0_Rack_1:1:C	AB:1734_VHSC:C:0					
HYS_Point_I0_Rack_1:1:1	AB:1734_VHSC:1:0					
Monitor Tags A Edit Tags /	· · · · · · · · · · · · · · · · · · ·	► La				

In this example alias HYS_Point_IO_Rack_20:I refers to data type AB:1734_3SLOT:I:0. Expand this tag to see how this data type is structured:

S	cope: Show:	🗸 🔽 Enter Description F	Mes
	Name	Data Type 🖂 🗅	Style
	HYS_Point_I0_Rack_20:1	AB:1734_3SLOT:1:0	
	HYS_Point_I0_Rack_20:I.SlotStatusBits0_31	DINT	Binary
	HYS_Point_I0_Rack_20:I.SlotStatusBits32_63	DINT	Binary
	HYS_Point_I0_Rack_20:I.Data	SINT[3]	Binary
	5 871, Page 2, Page 201	and the second second	

To make sure that HYS_Point_IO_Rack_20:I, and all his sub-tags, will be imported into the project, open the ETIPSpecialDataTypes.xml file in any text editor and check if the AB:1734_3SLOT:I:0 data type is included. If so you can proceed with the following data type. If not, you need to add it manually.

The structure is as in this example:

```
<DataType Name="aaa">
</Members>
</Member Name="bbb" DataType="ccc" Dimension="ddd" Radix="eee"/>
</Members>
</DataType>
```

where:

- aaa = Alias/Tag data type
- bbb = Sub-tag Name (it's sub-tag name part after dot)
- ccc = Sub-tag data type
- ddd = Array dimension (0 if it is not an array)
- eee = Style

In the example above:

블 ETIPSpecialDataTypes.xml 🗵

```
238
239 CataType Name="AB:1734_3SLOT:I:0">
240 Ammbers>
241 Ammber Name="SlotStatusBit0_31" DataType="DINT" Dimension="0" Radix="Binary"/>
242 Ammber Name="SlotStatusBit32_63" DataType="DINT" Dimension="0" Radix="Binary"/>
243 Ammber Name="Data" DataType="SINT" Dimension="3" Radix="Binary"/>
244 -</Members>
245 -</DataType>
```

- 3. Repeat step 2 for all Module-Defined data types.
- 4. Repeat the procedure from step 2, filtering Tags to display only **Predefined** Tags.

Controller Model Omron Sysmac

Data in NJ and CJ controllers can be accessed via CIP protocol.

Each data item can be identified by a string called "Tag". Use appropriate programming tools for controller to export the list of Tags.

NJ series controller are programmed using Sysmac Studio:

- NJ301-xxxx
- NJ501-xxxx

CJ series controller are programmed using CX-One:

- CJ2M CPU-3x
- CJ2H CPU 6x-EIP
- Any CPU with a CJ1W-EIP21 attached.

The project loaded on the HMI device must refer to the Tag names assigned in the programming software at development time. The Tag Editor supports direct import of the Tag file generated by Sysmac Studio software in .NJF format or generated by CX-One in the .CJF format.

All Tags to be accessed by the HMI device must be declared as Global Variables.

Export NJF files using Sysmac Studio

To export the **.NJF** Tag file:

- 1. In Sysmac Studio declare Tags as Global Variables.
- 2. Set the Network Publish attribute to Publish Only.

Edit View Insert Project	Controller Simulation To	ools Help		_	_				
					-				
	1 - A 10 - 1	* v = × • *		10 0					
o Casar 👻 🖣					_	_		7	_
	Programming		_	_	_	_			
	Global Variables X		_	_	-	-	_		
onfigurations and Setup	Name	Data Type	I Initial Value	AT .	I Retain	I Constan	nt I Network Publish	I Comment I	
▶ 篇 HherCAT	N1_Input_Bit_00	8001.		ECAT://mo-+			Publish Only		
▼ IS CPU/Expansion Racks	N3_REM_SF_RST	BOOL		ECAT://no_			Publish Only		
F III CPU Rack	HML_SAFETY_RST	BOOL			0	8	Publish Only	SAFETY RESET FROM HME	
▼ Controller Seturn	PLC_BYTE	BYTE			1	1	Do not publish		
C Operation Settings	HMLBYTE	8Y7E					Publish Only		
- # Built-in EtherNet/IF	PLC_DINT	DINT					Do not publish		
V 🗟 Motion Control Setup	HMI_DINT	DINT			13	13	Publish Only		
Ans Settings	PLC_OWORD	OWORD					Do not publish		
Axes Group Setting	HML DWORD	OWORD	- 53			B	Publish Only		
er Cam Data Settings	PLC_INT	DNT			B	8	Do not publish		
Task Settings	HMUINT	INT					Publish Only		
L III Data Trace Settings	PLC_LINT	UNT					Do not publish		
Programming	HMILIDNT	LINT			E	n	Publish Only		
▼ @ POUs	PLC LREAL	LREAL					Do not publish		
▼ [f] Programs	HMI LREAL	LREAL			E	1	Publish Only	2	
.▼ Main	PLC LWORD	LWORD			17	E	Output		_
Z: Safety	HMI LWORD	LWORD			B		Publish Only		
CommsTest	PLC REAL	REAL			E		Publish Only		
Eurotion Blocks	HMI REAL	REAL			1	E	Publish Only	1	_
▼ III Date	PLC SINT	SINT			m	0	Publish Only		
- 50 Data Types	HML SINT	SINT				E III	Publish Only		-
Clicked Verlation	PLC LIDENT	LIDINT		-	8	10	Publish Only		-
► Ell Tasks	HMI LIDINT	UDINT			m	E E	Publish Only		-
	PLC UINT	UINT			B		Publish Only		
	HMEUINT	UINT			8		Publish Only	-	_
	PLC URINT	URINT			0	10	Publish Only		-
	HMEDUNT	UEINT			0	0	Publish Only		
	PLC USINT	USINT			0		Publish Only		-
	UKR LICENT	UCDUT	_		0		Bublich Calu		_
	CINO. INCOM	1.04.000			44-490.000			1	
	Output								

2. From the **Tools** menu, choose **Export Global Variables > CX-Designer**.

File Edit View Insert Project Controller	Simula 1	Tools Help	
	ن کې له پ	Troubleshooting	A & C & C & C
Demo Case • 4 Master_PLC • I Clobel Varia	ming bles Sat	Export Global Variables Comments for Variables and Data Types Import ST Program Option	Network Configurator. CX-Designer
	Sysmac	Studio Global variables are copied to t Paste them to the symbol table	the clipboard. e of the CX-Designer.

- 3. Click **OK** to confirm.
- 4. Cut and paste the content of the clipboard in any text editor.

File Edit View Insert Project	Controller Simulation To	ols Help		_	_	_					
	1 8 4 5 5 5	# A 🛛 R 🔺 🦄 /	u 🖉 🌾 🦷	0 %	P						
Demo Case 🚽 🖡								1			
	Programming				_	_					
Master_PLC 🔻	Global Variables 🗙										
▼ Configurations and Setup	Name	I Data Type	Initial Value	I AT	Retain	I Constar	t I Network Publish	Comment	T		
► ﷺ EtherCAT	N1 Input Bit 00	BOOL		ECAT://no	E	E	Publish Only		2		
▼ ISt CPU/Expansion Racks	N3 REM SF RST	BOOL		ECAT://np	П	E	Publish Only				
► *** CPU Rack	HML SAFETY RST	BOOL				0	Publish Only	SAFETY RESET FROM HMI	8		
L = 1/O Map	PLC_BYTE	BYTE			E	E	Do not publish				
Operation Settings	HML_BYTE (
Built-in EtherNet/IF	PLC_DINT	Export.NJF - Blocco note									
▼ 🖶 Motion Control Setup	HMI_DINT	File Modifica Formato	Visualizza	?							
🗆 🗆 🗄 Auis Settings	PLC_DWORD										
Axes Group Setting	HMI_DWORD	NAME DA	ATATTPE		JURESS	nodet	ENI TAGLIN	R RW PUU	ut Rit 00		
- & Cam Data Settings	PLC_INT	N3 REM SE	RST BO		AT://	node#	[1,3]/REM	SE RST	TRUE RW		
Event Settings	HMLINT	HMI_SAFETY	RST BO	DL		SAFE	TY RESET FR	ROM HMI TRUE	RW		
El Data Trace Settings	PLC_LINT	HMI_BYTE	BY	TE			TRUE	RW			
▼ Programming	HMI LINT	HMI_DINT	DI	T			TRUE	RW			
V 🖪 POUs	PLC_LREAL	HMI_DWORD	DW	ORD		TOUT	TRUE	RW			
■ ▼ 🕷 Programs	HMI LREAL		I TI	UT.		TRUE	TRUE	Diel			
🗮 🔻 🖂 Main	PLC_LWORD		LEI	FAL			TRUE	RW			
💷 🖾 Safety	HMI LWORD	HMI_LREAL	LRI	EAL			TRUE	RW			
CommsTest	PLC_REAL	PLC_LWORD	LW	DRD			TRUE	RW			
2° Function Blocks	HMI_REAL	HMI_LWORD	LW	DRD			TRUE	RW			
V III Data	PLC_SINT	PLC_REAL	REA	AL			TRUE	RW			
📃 💷 🖯 🖞 Data Types	HMLSINT		KE/				TRUE	RW			
📕 🖉 🖂 Global Variables	PLC_UDINT	HMT STNT	STI	NT			TRUE	RW			
► En Tasks	HMI_UDINT	PLC_UDINT	UD	INT			TRUE	RW			
	PLC UINT	HMI_UDINT	UD:	INT			TRUE	RW			
	LINE LITER	PLC_UINT	UII	T			TRUE	RW			

4. Save the file as .NJF.



Export CJF file using CX-One

To export the **.CJF** Tag file:

- 1. In CX-One open the Symbols file in the project.
- 2. In the Edit Symbol dialog set the Net. Variables attribute to Publication.



3. Copy and paste all the Tags in any text editor.

					D 57
Untitled - Notepad	10 10 10 10 10 10 10 10 10 10 10 10 10 1		and a first start		
File Edit Format View Help					
SBT_ISland1_LPSD0_OUT_Buff	WORD[24]	Seperate input o	of array data for loop	0	
SBT_Island1_MaxDev1_Time	DINT ISla	nd 1 Device 1 Maximu	um Transmission Time	0	
SBT_Island1_MaxDev2_Time	DINT ISla	nd 1 Device 2 Maximu	Im Transmission Time	0	
SBT_IS[and1_MaxDev3_Time	DINT IS a	nd 1 Device 3 Maximu	Im Transmission Time	0	
SBT_Island1_MaxDev4_Time	DINT IS	Saug Ar	the statement of the st		X
SBT_Island1_MaxDev5_Time	DINT IS	Jave As	a management road		
SBI_ISTANDI_RD_TT BOOL	ISTAND I RE				
SBI_ISLANDI_II FUNCTION BLOCK		G C export		▼ Search CJ export	Q
SBT_ISTandI_WADDAck	Application		100 C		
SBT_ISTANUI_WAPPACK WORD	Application	Organize Vew fold	er	855 🗸	
SBT_ISland1_wDjacode_wOPD	EB Diag Cod				~
SBT_Island1_wDiagCodeDev1	WORD Dia		 Name 	 Date modified 	Type
SBT Island1 wDiagCodeDev2	WORD Dia	a facolites			10262
SBT_Island1_wDiagCodeDev3	WORD Dia	Station .		7 24 21	
SBT_Island1_wDiagCodeDev4	WORD Dia		No	tems match your search.	
SBT_Island1_wDiagCodeDev5	WORD Dia	(re Computer	E		
SBT_Island1_wDiagCodeLPSD0	WORD Dia	 Description 	1		
SBT_Island1_xAcceptDiffLogic	BOOL ACC				
SBT_Island1_xAckDev1 BOOL	Acknowledge	Cropbes			
SBT_Island1_xAckDev2 BOOL	Acknowledge	St. Recard Places			
SBI_ISTANDI_XACKDEV3 BOOL	ACKNOWledge				
SBT_ISTANDI_XACKDEV4 BOOL	Acknowledge				
SET_ISTATUL_XACKDEVS BOOL	Acknowledge	Libraries			
SBT_ISland1_XACKEPSD0 BOOL	Operator Ad	10.0			
SBT_Island1_xActivate BOOL	Activate FR	C.S. Documents			
SBT_Tsland1_xActive BOOL	FB Active	A Music			
SBT_ISland1_XCOK BOOL	Communicati				
SBT_ISland1_xDevError BOOL	Operator Ad	See Pictures	I		1 C
SBT_Island1_xDiffLogicDetected	BOOL Dif				
SBT_IS]and1_xError BOOL	Error	File name: CJF			-
SBT_IS and1_xOpAckReq BOOL	Power Up Re	-			
SBT_ISIand1_xPUR BOOL	Device Errd	Save as type: All Fil	es (^.^)		•
SBI_ISTANDL_XRUNLOGIC BOOL	SBT LOGIC R				
SBI_ISLAND_I FUNCTION BLOCK	COT NO COFO				
strilonamo strike	SBI VZ SATE	Hide Folders	Encoding: ANSI	 Save Can 	cel
	Teland 1 Transmissi				
udtSBT_TSland1_STRUCT_D5000	Tsland 1 Safety Data	1	WU SOL 97		
adessi_istandi Sikoci biooo	is raise i sarecy baca				

4. Save the file as **.CJF**.

1

Note: Using Notepad as text editor, make sure to save the text file with **.CJF** extension by selecting "Save as type" as "All Files" although the file will be named *.cjf.txt and it will not be visible from importer.

Export User Defined structures

To export the .CJS Tag file:

1. In CX-One open the Data Types file in the project.

	10 10 16 4 3 55			
2 (2) 2 2 1 4 0 3 3 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				TOPISIAN
	IX Name	Data Type	Array Size	Comment
- NewProject +	2 😒 adtSB7_V2			
E-SBT_Test[CJ2M] Offline	DEV_1_IN_Buff	WORD	[4]	
Data Types	- DEV_2_IN_Buff	WORD	[4]	
	DEV_3_IN_Buff	WORD	[4]	
	DEV 4 IN Suff	WORD	[4]	
Settings	DEV_5_IN_Buff	WORD	[4]	
- A Memory	- DEV 1_OUT_Buff	WORD	[4]	
E Se Programs	DEV_2_OUT_Buff	WORD	[4]	
B- SBT_V2_TEST_PRG (00)	DEV_3_OUT_Buff	WORD	[4]	
E IF Function Blocks	DEV_4_OUT_Buff	WORD	[6]	
	DEV_S_OUT_Buff	WORD	[4]	
- SET V2 ISLAND	🚝 iProjRead	INT		
SBT V2 OPERATE	🚟 iBlockNum	INT		
SBT wbOUT	* xBlockRead	BOOL		
- I SRT whIST	 xDiffLogicDetected 	BODL		
	* xLPSDO_RUN	8001		
	* xActivate	BOOL		
	* XCOK	BOOL		
	* xTransmit5rid	BOOL	[6]	

2. Copy and paste all the Tags in any text editor.

Untitled - Notepad			
File Edit Format View Help			
SBT_IS]and1_LPSD0_OUT_Buff SBT_IS]and1_MaxDev1_Time SBT_IS]and1_MaxDev2_Time SBT_IS]and1_MaxDev3_Time	WORD[24] DINT IS]a DINT IS]a DINT IS]a	Seperate input of array data for loc nd 1 Device 1 Maximum Transmission Time nd 1 Device 2 Maximum Transmission Time nd 1 Device 3 Maximum Transmission Time	pp 0 0 0 0
SBT_ISTANDI_MAXDEV4_TIME SBT_ISTANDI_MAXDEV5_TIME SBT_ISTANDI_RD_TT BOOL SBT_ISTANDI_TT FUNCTION BLOCK	DINT IST DINT IST ISTAND 1 Re	Save As	
SBT_Island1_wAddDiagCode SBT_Island1_wAppAck WORD SBT_Island1_wAppDiag WORD SBT_Island1_wDiagCode WORD	WORD FB Application Application FB Diag Cod	Organize ▼ New folder	8≡ • @
SBT_ISland1_wDiagCodeDev1 SBT_ISland1_wDiagCodeDev2	WORD Dia WORD Dia	A Name	✓ Date modified Type
<pre>[SBT_IS1 and1_wbiagCodeDev3 SBT_IS1 and1_wbiagCodeDev4 SBT_IS1 and1_wbiagCodeDev5 SBT_IS1 and1_wbiagCodeLPSD0 SBT_IS1 and1_xAcceptDiffLogic SBT_IS1 and1_xAckDev1 BOOL SBT_IS1 and1_xAckDev2 BOOL</pre>	WORD Dia WORD Dia WORD Dia WORD Dia BOOL Acc Acknowledge Acknowledge	Computer Discrimation Discrimation	Vo items match your search.
SBT_IS1and1_XAckDev3 BOOL SBT_IS1and1_XAckDev4 BOOL SBT_IS1and1_XAckDev5 BOOL SBT_IS1and1_XAckLPSD0 BOOL SBT_IS1and1_XAckLPSD0 BOOL SBT_IS1and1_XAckTPSD0 BOOL	Acknowledge Acknowledge Acknowledge Acknowledge Operator Ac Activate Fe	Californias Californias Californias	
SBT_IS andl_xActive BOOL SBT_IS andl_xCOK BOOL SBT_IS andl_xDevError BOOL	FB Active Communicati Operator Ac	Fatures v (m
SBT_ISTand1_xError BOOL SBT_ISTand1_xOpAckReq BOOL	Error Power Up Re	File name: CJS	7
SBT_ISland1_XPUR BOOL SBT_ISland1_XRunLogic BOOL SBT_ISLAND_1 FUNCTION BLOCK	Device Erro SBT Logic R	Save as type: All Files (*.*)	
SBT_V2_SFPRG FUNCTION BLOCK strFilename STRING[40] TT_TMR_PV UINT	SBT V2 Safe Island 1 Transmissi	Hide Folders Encoding: ANSI	Save Cancel
udtSBT_Island1 STRUCT D5000	Island 1 Safety Data	uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu	

3. Save the file as **.CJS**.

Note: Using Notepad as text editor, make sure to save the text file with **.CJS** extension by selecting "Save as type" as "All Files" although the file will be named *.cjs.txt and it will not be visible from importer.

Import Files in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	X	Đ	ß	>]	₽	A 9B	B>	ŧ.	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers are	available for this protocol. Please select the importer type and continue.
Version	Туре
Allen-Bradley RSLogix5000 v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Select Omron Sysmac to import a .NJF Tags file or Omron CX-One to import a .CJF Tags file.

Once the importer has been selected, locate the Tags file and click **Open**. The system will ask for User Defined structures **.CJS** file. If not required, skip the dialog by clicking on Cancel button.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags × Protocols				
· + - → ◎ ② >] [> \$= 助 韵	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🗸 🗹 Show all tags 🔅 🗖
Data Type	Comment	^	Property	Value
Modbus TCP:prot1			Y Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			Y Dictionary	
Holding Registers 3 unsignedShort				61
MRTU1 unsignedShort			Array	taise
MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
MRTU4 unsignedShort			Comment	
MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
鼓	Update Tag(s).
	Click on this icon to update the tags in the project, due a

Toolbar item	Description
	new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
	Tug2 DR/T Tug2 DR/T Tug3 DR/T Tug3 DR/T
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.



Note: When importing the array data types, the importer is expanding them creating individual Tags per each array element; this is valid for all the data types, except for arrays of boolean. In this case they are imported as "boolean-32" and the single array element can be addressed using "Tag Index" parameter from "Attach to…" dialog.

Controller Model Micro800

The Ethernet/IP CIP driver provides an easy and reliable way to connect to Allen-Bradley Micro800 controllers.

The scope of variables into a Micro800 controller can be local to a program or global:

Scope	Description
Local Variables	Program-scoped Tags. Tags are assigned to a specific program in the project and available only to that program.
	These Tags are not supported within this driver.
Global Variables	Controller-scoped Tags. Tags belong to the controller in the project and are available to any program in the project.
	These Tags are supported within this driver.

Export ISAXML file using Connected Component Workbench

To export .ISAXML global variables including I/O tags:

- 1. Select Device tab.
- 2. Expand Export item.
- 3. Select Export Device.



- 4. Click on Export Exchange File tab.
- 5. Click Export button.

Import Export	× □ ×
📑 Import Exchange File 🔄 Export Exchange File	
Set Password	
Password	
Password	
Confirm Password	
Element Exported Controller.Micro850	ort Close

6. Choose a location where to save the export file and click **Save**.

Save As							×
🚱 🗢 💻 Deskto	op 🕨		•	44	Search Desktop		٩
Organize 👻 Nev	v folde	ir				88 -	0
		ibraries Homegroup Admin Computer Network					
File name:	Contr	oller.Micro850					•
Hide Folders	JEVEN	ар шо (<i>т</i> .с.			Save	Cance	

7. When the export is completed successfully the output information is displayed:





Note: CCW export file is a 7-zip compressed archive. Use a suitable zip utility to extract archive content into a local folder.

Import Files in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	X	đ	ß	>]	₽	A 9B	B>	樹	1
Data			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers are	available for this protocol. Please select the importer type and continue.
Version	Туре
Allen-Bradley RSLogix5000 v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Select Allen-Bradely CCW v15 option.

Directory structure extracted from 7z file is something like: "..\<folder_name>\Micro8xx\Micro8xx\"

Inside this last folder, select the Micro8xx.isaxml file as shown below:

Irganize 🔻 📄 Open				-	6
Favorites	Name	Date modified	Туре	Size	
E Desktop	Demo_logic.annex	18/06/2015 09:58	ANNEX File	8 KB	
🐞 Downloads	Demo logic.isaxml	18/06/2015 09:58	ISAXML File	6 KB	
Recorded TV	Micro850.isaxml	18/06/2015 09:58	ISAXML File	21 KB	
ᡖ Google Drive	Mod_Message.annex	18/06/2015 09:58	ANNEX File	16 KB	
📃 Recent Places	Mod_Message.isaxml	18/06/2015 09:58	ISAXML File	8 KB	
	oee_calcs.annex	18/06/2015 09:58	ANNEX File	1 KB	
🚽 Libraries	oee_calcs.isaxml	18/06/2015 09:58	ISAXML File	7 KB	
	PID_Feedback.annex	18/06/2015 09:58	ANNEX File	1 KB	
💐 Homegroup	PID_Feedback.isaxml	18/06/2015 09:58	ISAXML File	3 KB	
	PID_OutputRegulator.annex	18/06/2015 09:58	ANNEX File	1 KB	
Computer	PID_OutputRegulator.isaxml	18/06/2015 09:58	ISAXML File	3 KB	
Local Disk (C:)	PID_PWM_LD.annex	18/06/2015 09:58	ANNEX File	7 KB	
Controller.Micro850	PID_PWM_LD.isaxml	18/06/2015 09:58	ISAXML File	5 KB	
Micro850	PID_PWM_SCALE.annex	18/06/2015 09:58	ANNEX File	1 KB	
Micro850	PID_PWM_SCALE.isaxml	18/06/2015 09:58	ISAXML File	4 KB	
	PIDWaterLvl.annex	18/06/2015 09:58	ANNEX File	9 KB	

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 🗸 🕲 🖉 👌	S 🚯 🖬	R 🔎 - Search Trilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 💮 🗖
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	Container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
Holding Registers 3	unsignedShort			Array	falce
- MRIU1	unsignedShort			Andy	
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
	- 1-1			and the	an angi na aran na na

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
₫ 3	Update Tag(s).



Toolbar item	Description
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

トークイ ショ	a 📖 💙	Modbus TCP:pro	vt1	- 5		
Name /		Group	Driver	Addre	ess	Comment
Node 1/Dat, Nodentia	10	Add	Medlue TCP pret1	1 11 Dune	presidine i	1
Node's/Oata_hodrotta		AG11	Wodeus TCP prot1	1 12 Duma	prest(Short)	
Node 1/19, MATER, Jeve	4	Add	Wodeue TCP pret1	1 Dithureg	redPort	
Notes CHOCHERCHE		Add	Wollow TCP profil	1 245 Dune	medilion	
Neder 1.42421, BX7, Inciden	din.	Add	Wedlaue TCP prot 1	1.1 During	redither	
Node 1/R, CATA, Instituti		Add	Wodeue TCP pret1	1 2 During	redither	
PROVINE TO THE R. THE P.		Add	Wodeus TCP pret1	1 3 Dureig	Hereither	
Node1/Water_level		-1010	Collective ETHINAT	1 10 0 unsig	gredShort	
	0	Node id as defined	in import file de id		2	
		Slave Id	Model		Alias	
		Slave Id	Model	alitar	Alias Node1	
		Slave Id	Model	ellue ellue	Alias Node1 Node2	
Ra R		Slave Id	Model	iltur iltur	Alias Node1 Node2	
tagname V	me	Slave Id	Model	iltur iltur	Alias Node1 Node2	
tagname Water_level	me	Slave Id	Model	effice effice	Alias Node1 Node2 Ok) Cancel
tagname Water_level	me	Slave Id	Model	illus illus	Alias Node1 Node2 Ok	Cancel
tagname Water_level	me	Slave Id	Model	inini inini	Alias Node1 Node2 Ok	Cancel
tagname Water_level	me	Slave Id	Model	illing illing	Alias Node1 Node2 Ok	Cancel
tagname Water_level	me	Slave Id	Model	ilhus ilhus (Alias Node1 Node2 Ok	Cancel
tagname Water_level	me	Slave Id	Model		Alias Node1 Node2 Ok) Cancel



Note: Aliasing tag names is only available for imported tags. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached on the import. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

thernet/IP CIP				
Memory Type	ArrayIndex	SubIndex		
Data Type	Arraysize	Conversion		
unsignedByte [] 🛛 👻	4	-	+/-	
Folder Name	Structure N	ame	Tag Name	

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio Multiple tag importers are available for this protocol. Please select the importer type and conversion Version Type Allen-Bradley RSLogix5000 v1.1 Hierarchical Allen-Bradley RSLogix5000 v15 Linear Allen-Bradley CCW v15 Linear Omron Sysmac v15 Linear Omron CX-One v15 Linear	
Multiple tag importers are available for this protocol. Please select the importer type and control Version Type Allen-Bradley RSLogix5000 v1.1 Hierarchical Allen-Bradley RSLogix5000 v15 Linear Allen-Bradley CCW v15 Linear Omron Sysmac v15 Linear Omron CX-One v15 Linear	
VersionTypeAllen-Bradley RSLogix5000 v1.1HierarchicalAllen-Bradley RSLogix5000 v15LinearAllen-Bradley CCW v15LinearOmron Sysmac v15LinearOmron CX-One v15Linear	ntinue.
Allen-Bradley RSLogix 5000 v1.1HierarchicalAllen-Bradley RSLogix 5000 v15LinearAllen-Bradley CCW v15LinearOmron Sysmac v15LinearOmron CX-One v15Linear	
Allen-Bradley RSLogix5000 v15 Linear Allen-Bradley CCW v15 Linear Omron Sysmac v15 Linear Omron CX-One v15 Linear	
Allen-Bradley CCW v15 Linear Omron Sysmac v15 Linear Omron CX-One v15 Linear	
Omron Sysmac v15 Linear Omron CX-One v15 Linear	
Omron CX-One v15 Linear	
Tag Editor exported xml 1.1 General	
Watched dictionary file:	
Keep synchronized	
OK Car	cel

Importer	Description
Allen-Bradley L5X v1.1	Requires a .L5X file.
Hierarchical	Check Controller Model Logix 5000 for more details.
	All variables will be displayed according to RSLogix5000 Hierarchical view.
Allen-Bradley	Requires a .CSV and .L5X (optional) files.
RSLogix5000 v15 Linear	Check Controller Model Logix 5000 for more details.
	All variables will be displayed at the same level.
Allen-Bradley CCW v15	Requires a .ISAXML file.
Linear	Check Controller Model Micro800 for more details.
	All variables will be displayed at the same level.
Omron Sysmac v15	Requires a . NJF file.
Linear	Check Controller Model Omron Sysmac for more details.
	All variables will be displayed at the same level.
Omron CX-One v15	Requires a .CJFand .CJS (optional) files.
Linear	Check Controller Model Omron Sysmac for more details.



Importer	Description				
	All variables will be displayed at the same level.				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖄 🔰 🕻	> \$ ∎ ₪ 🚮	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data	Type	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			× Dictionary	
- Holding Registers 3	unsignedShort			• Dictionary	
- MRTU1	unsignedShort			Array	talse
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
	in the second se			10 C	

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.

Toolbar item	Description
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
	Tops N Tops N + - & & >
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Fatek FACON ETH

The Fatek FACON ETH communication driver has been designed to connect HMI devices to a Fatek FACON PLC through Ethernet connection.

Protocol Editor Settings

Fatek FACON ETH		×
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	500	
station	1	
Timeout	2000	
PLC Models FB Series		

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
IP Address	Ethernet IP address of the PLC.
Port	Port number used to communicate with PLC.
station	station number according to PLC configuration.

Element	Description				
Timeout	Time delay in milliseco	onds between two retries in ca	ase of missing response	from the PLC.	
PLC Models	PLC model available:				
	FB Series				
PLC Networ k	IP address for all contr enable multiple conne	rollers in multiple connections ctions.	. PLC Network must b	e selected to	
	Fatek FACON ETH			×	
	PLC Network		c	К	
	Alias		Car	ncel	
	IP address	0.0.	0.0		
	Port	500			
	station	Fatek FACON ETH			×
	Timeout				OK
	PLC Models	Alias		Č	Cancel
		IP address	0.0.	0.1	
		Port	500	▲ ▼	
	Slaves	station	1	A	
	Slave Id	Timeout	2000		
		PLC Models			
		rb series			

Tag Editor Settings

In Tag Editor select the protocol Fatek FACON ETH.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

Memory TypeOffsetSubIndexInput Discrete00Data TypeArraysizeConversionboolean01				н	Fatek FACON ETH
Input Discrete O Data Type Arraysize Conversion boolean O		SubIndex	Offset		Memory Type
Data Type Arraysize Conversion boolean ▼ 0 +/-		0 👻	0	•	Input Discrete
boolean v 0 +/-		Conversion	Arraysize		Data Type
	+/-	1	0	•	boolean

Element	Description			
Memory Type	Memory Type	Description		
iype	Input Discrete	X resources. Corresponding to External Digital Input Point.		
	Output Relay	Y resources. Corresponding to External Digital Output Point.		
	Internal Relay	M resources. Corresponding to PLC internal memory.		
	Step Relay	S resources.		
	Timer Discrete T resources.			
	Counter Discrete	C resources.		
	Timer Register	Current Time Value Register.		
	Counter Register	Current Counter Value Register.		
	Data Register - HR	R resources.		
	Data Register - DR	D resources.		
	Run	Boolean value. Corresponding to PLC status.		
	Node Override IP	See Special Data Types for specifications.		
Offset	Starting address for the Tag. The possible range depend on PLC model selected.			
SubIndex	This allows resource offset sel	ection depending on the selected data type.		
Data Type	Available data types:			
	• boolean			
	• byte			
	• snort			
	 unsignedByte 			
	 unsignedShort 			
	 unsignedInt 			
	• float			
	• double			
	string binary			
	See "Programming concepts"	section in the main manual		
	Note: To define array	vs. select one of Data Type format followed by square		
	brackets (byte[], sho	rt[]).		

Arraysize In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. Conversion Conversion to be applied to the tag. Conversion Conversion to be applied to the tag. Value Depending on data type selected, the list Allowed shows one or more conversion types. Value Description Inv bits Inv: Invert all the bits of the tag. Value Description Inv bits Inv: Invert all the bits of the tag. Example: 1001 → 0110 (in binary format) 9 → 6 (in decimal format) 9 → 6 (in decimal format) 9 → 6 (in decimal format) S588 → 20.36 AB > BA swapnibbles: Swap nibbles in a byte. Example: 1504 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format) 3962B → 52378 (in decimal format) 3962B → 52378 (in decimal format) 3962B → 52378 (in decimal format) 3962B → 52378 (in decimal format) 3962B → 52378 (in decimal format)	Element	Description				
Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. Conversion Conversion Conversion Conversion Inv.swap2 BCO BCD ABCD-SCDAB BCDC-SCDAB BCDC-SCDAB Depending on data type selected, the list Allowed shows one or more conversion types. Value Description Inv bits inv: Invert all the bits of the tag. Example: 1001 → 0110 (in binary format) 9 → 6 (in decimal format) 9 → 6 (in decimal format) 9 → 6 (in decimal format) 588 → 20813 (in decimal format) ABCD-SC 36 AB -> BA Swapplibles: Swap tibbles in a byte. Example: 15D4 → 514D (in hexadecimal format) 588 → 20813 (in decimal format) 3628 → 22378 (in decimal format) 3628 → 52378 (in decimal format) 3628 → 52378 (in decimal format) 3628 → 52378 (in decimal format) 3628 → 52378 (in decimal format)	Arraysize	 In case o In case o available 	of array tag, this property represents the number of array elements. If string tag, this property represents the maximum number of bytes in the string tag.			
Conversion Conversion to be applied to the tag. Conversion Inv.swap2 Inv.swap2 Isoured Inv.bits Inv.bits Inv.bits Inv.bits Depending on data type selected, the list Allowed shows one or more conversion types. Value Description Inv bits inv: Invert all the bits of the tag. Example: 1001 -> 0110 (in binary format) 9 -> 6 (in decimal format) 9 -> 6 (in decimal format) Negate neg: Set the opposite of tag value. Example: 25.36 AB -> BA swapnibbles: Swap nibbles in a byte. Example: 1504 -> 514D (in hexadecimal format) 5588 -> 20813 (in decimal format) 5588 -> 20813 (in decimal format) ABCD -> Swap2: Swap bytes in a word. Example: 94054 (in hexadecimal format) 39628 -> 52378 (in decimal format) 39628 -> 52378 (in decimal format)		Note: number of set to UTF-8 or If Encoding prop character requir	f bytes corresponds to number of string characters if Encoding property is Latin1 in Tag Editor. perty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one res 2 bytes.			
Conversion Imv,swap2 BCD Allowed Configured BCD ABCD>-CDAB ABCD>-CDAB ABCD=YCDAB ABCD>-CDAB ABCD>-CDAB ABCD=YCDAB Inv bits Inv bits Depending on data type selected, the list Allowed shows one or more conversion types. Value Description Inv bits inv: Invert all the bits of the tag. Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format) 9 \rightarrow 6 (in decimal format) 9 \rightarrow 6 (in decimal format) 9 \rightarrow 6 (in decimal format) Negate neg: Set the opposite of tag value. Example: 25.36 AB \rightarrow BA swapnibbles: Swap nibbles in a byte. Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format) 5588 \rightarrow 20813 (in decimal format) 5588 \rightarrow 20813 (in decimal format) 39628 \rightarrow 52378 (in decimal format) 39628 \rightarrow 52378 (in decimal format) 39628 \rightarrow 52378 (in decimal format) 39628 \rightarrow 52378 (in decimal format) 39628 \rightarrow 52378 (in decimal format)	Conversion	Conversion to b	e applied to the tag.			
AllowedConfiguredBCDBCDTry bitsAB:>BAAB:CD->CDABAB:CD->CDABAB:CD->CDABAB:CD->CDABCancel OKDepending on data type selected, the list Allowed shows one or more conversion types.ValueDescriptionInv bitsinv: Invert all the bits of the tag.Example: 1001 0110 (in binary format) 9 6 (in decimal format)Negateneg: Set the opposite of tag value.Example: 25.36 25.36AB -> BAswapnibbles: Swap nibbles in a byte.Example: 1504 514D (in hexadecimal format)AB:CD -> CDABswap2: Swap bytes in a word.AB:CD -> CDABswap4: Swap bytes in a double word.		Conversion				
$\begin{array}{ c c c c c } & & & & & & & & & & & & & & & & & & &$		inv,swap2	Allowed Configured			
Depending on data type selected, the list Allowed shows one or more conversion types.ValueDescriptionInv bitsinv: Invert all the bits of the tag. $Example:$ $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D$ (in hexadecimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ $9ACC \rightarrow CC9A$ (in hexadecimal format)ABCDEFGHswap4: Swap bytes in a double word.			BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK			
ValueDescriptionInv bitsinv: Invert all the bits of the tag. $Example:$ $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D$ (in hexadecimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ $9ACC \rightarrow CC9A$ (in hexadecimal format)ABCDEFGHswap4: Swap bytes in a double word.		Dependina on d	on data type selected, the list Allowed shows one or more conversion types.			
Inv bitsinv: Invert all the bits of the tag. $Example:$ $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D$ (in hexadecimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ $9ACC \rightarrow CC9A$ (in hexadecimal format)ABCDEFGHswap4: Swap types in a double word.		Value Description				
International displayed and displayed ano		Inv bits	inv : Invert all the bits of the tag			
Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)ABCDEFGHswap4: Swap bytes in a double word.			<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)			
Example: $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte.Example: $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)ABCDEFGHswap4: Swap bytes in a double word.		Negate	neg: Set the opposite of tag value.			
AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)ABCDEFGHswap4: Swap bytes in a double word.			<i>Example:</i> 25.36 → -25.36			
Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. 		AB -> BA	swapnibbles: Swap nibbles in a byte.			
ABCD -> CDABswap2: Swap bytes in a word. $Example:$ 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)ABCDEFGHswap4: Swap bytes in a double word.			<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)			
CDABExample: $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)ABCDEFGHswap4: Swap bytes in a double word.		ABCD ->	swap2: Swap bytes in a word.			
ABCDEFGH swap4: Swap bytes in a double word.		CDAB	<i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)			
		ABCDEFGH	swap4 : Swap bytes in a double word.			

Element	Description	
	Value	Description
	-> GHEFCDAB	<i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010000101101000011100101
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion	on and click +. The selected item will be added to list Configured .
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom of list
	Use the arrow b	outtons to order the configured conversions.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the PLC at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the PLC IP specified in the project at programming time.

Node Override IP	Modbus operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one PLC node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

atek FACON ETH				
Memory Type	Offset	SubIndex		
Node Override IP 👻	0	0 *		
Data Type	Arraysize	Conversion		
unsignedByte [] 🛛 👻	4		+/-	

Fatek FACON SER

The Fatek FACON SER communication driver has been designed to connect HMI devices to a Fatek FACON PLC through Serial connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Fatek FACON SER		×
PLC Network	Comm	ОК
Alias		Cancel
Node ID	1	
PLC Models FB Series		

Element	Description
Node ID	Serial node associated to the PLC.
PLC Models	PLC model available: • FB Series
PLC Netwo rk	IP address for all controllers in multiple connections. PLC Network must be selected to enable multiple connections.

Element	Description		
	Fatek FACON SER	×	
	PLC Network	Comm OK	
	Alias	Cancel	
	Node ID	-1	
	PLC Models		
		Fatek FACON SER	×
			ОК
	Slaves	Alias	Cancel
	Slave Id	Node ID 1	
		PLC Models	
		T D Jenes	
Comm	If clicked displays the com	munication parameters setup dialog.	
	Comm Parameter Dialog	×	
		ОК	
	Port	com1 👻	
	Baudrate	9600 👻	
	Parity	none 💌	
	Data bits	8	
	Stop bits	1	
	Mode	RS-485	

Element	Description		
	Element	Parameter	
	Port	Serial port selection.	
COM1: device PLC port.		COM1: device PLC port.	
COM2: computer/printer port on panels with 2 optional Plug-In module plugged on Slot 1/2 free serial port on-board.		 COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board. 	
COM3: optional Plug panels with 1 serial p		 COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board. 	
	Baudrate, Parity, Data Bits, Stop bitsSerial line parameters.		
	Mode	Serial port mode. Available modes:	
		• RS-232.	
		• RS-485 (2 wires).	
		• RS-422 (4 wires).	

Tag Editor Settings

In Tag Editor select the protocol Fatek FACON SER.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

Fatek FACON SER				
Memory Type	Offset	SubIndex		
Input Discrete	• 0	0 🐨		
Data Type	Arraysize	Conversion		
boolean	•		+/-	
				822
		OK Canaal	å oply	Hele

Element	Description		
Memory	Memory Type	Description	
туре	Input Discrete	X resources. Corresponding to External Digital Input Point.	
	Output Relay	Y resources. Corresponding to External Digital Output Point.	
	Internal Relay	M resources. Corresponding to PLC internal memory.	
	Step Relay	S resources.	
	Timer Discrete	T resources.	
	Counter Discrete	C resources.	
	Timer Register	Current Time Value Register.	
	Counter Register	Current Counter Value Register.	
	Data Register - HR	R resources.	
	Data Register - DR	D resources.	
	Run	Boolean value. Corresponding to PLC status.	
Offset	Starting address for the Tag. The possible range depend on PLC model selected.		
SubIndex	This allows resource offset selection depending on the selected data type.		
Data Type	Available data types:		
	• boolean		
	• byte		
	• short		
	unsignedByte		
	 unsignedShort 		
	 unsignedInt 		
	• float		
	double		
	string binary		
	See "Programming concepts" (section in the main manual	
		re colort and of Data Type format followed by agrees	
	brackets (byte[], short[]).		

Element	Description			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 			
Conversion	Conversion to be applied to the tag.			
	Conversion			
	inv,swap2	Allowed Configured Inv bits ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK		
	Depending on data type selected, the list Allowed shows one or more conversion types.			
	Value	Description		
	Inv bits	inv : Invert all the bits of the tag.		
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	neg: Set the opposite of tag value.		
		<i>Example:</i> 25.36 → -25.36		
	AB -> BA	swapnibbles : Swap nibbles in a byte. <i>Example:</i> $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)		
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.		

Element	Description		
	Value	Description	
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)	
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010	
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	Select conversion If more conversion Configured). Use the arrow b	on and click +. The selected item will be added to list Configured . ions are configured, they will be applied in order (from top to bottom of list uttons to order the configured conversions.	
		-	

GE Intelligent Platforms SNP

The GE Intelligent Platforms SNP driver can be used to connect the HMI device to the GE controllers through serial connection using the native and proprietary SNP communication protocol.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the $\ensuremath{\text{PLC}}$ list.

The protocol configuration dialog is displayed.

GE Intelligent Platforms SNP		×
PLC Network	Comm	ОК
Alias		Cancel
SNP ID	0]
Protocol type	SNP ~	
PLC Models		
90-20		<u>^</u>
90-30 311		
90-30 331		-
90-70 731/732		
90-70 771/772	_ / / / / / /	
90-70 780/781/782/789/914/91	15/924/925/928/935	<u> </u>
90-30 340/341 90-30 Micro 14 points		
90-30 313/321/323		
90-30 351/352/360/363/364		
90-70 788		
90-30 Micro 28 points		
90-30 350/374		
VersaMax Nano (NDD101, NDR)		
VersaMax Micro 14 points		
VersaMax Micro 20 points		
VersaMax CPU001		
VersaMax (CPU005, CPUE05)		
PACSystem RX3i		
VersaMax Micro 64 points		
Fanuc CNC		
Eleme nt	Description	
-------------------	---	
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.	
PLC Model s	PLC models available.	

Eleme nt	Description	n								
Proto col type	Allows to se	ect between SNP and SNP-X protocol.								
Com m	If clicked displays the communication parameters setup dialog.									
	Comm Para	ameter Dialog								
	Port Baud Parity Data Stop Mode	Com1 rate 19200 odd odd bits 8 bits 1 RS-485								
	Element	Parameter								
	Port	Serial port selection.								
		 COM1: On-board port COM2: Optional Plug-in module plugged on slot#1 or slot#2 COM3: Optional Plug-in module plugged on slot#3 or slot#4 								
	Baudrat e, Parity, Data Bits, Stop bits	Serial line parameters.								

Eleme nt	Description	n				
	Element	Parameter				
	Mode					
		 RS-232. RS-485 (2 wir RS-422 (4 wir 	res). res).			
	PLC Network	Multiple controllers c connections, select F	an be connected to one PLC networkand click	HMI device. To set-up mul Add to configure each slave	tiple	
		GE Intelligent Platford	ms SNP	Comm	OK Cancel	
		Protocol type	GE Intelligent Platforms	SNP		×
		PLC Models VersaMax CPU0 VersaMax (CPU0 PACSystem RX3 VersaMax Micro Fanuc CNC Slaves Slave Id	Alias SNP ID Protocol type PLC Models 90-30 311 90-30 331 90-70 731/732 90-70 771/772 90-70 780/781/782/ 90-30 340/341	0 SNP 789/914/915/924/925/928/935	~	OK Cancel

Tag Editor Settings

In Tag Editor select the protocol GE Intelligent Platforms SNP.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

GE Intelligent Platforms	s SNP						
Memory Type		Offset		SubIn	dex		
Register	•	1	×	0			
Data Type		Arraysize		Conver	sion		
boolean	•	0				+/-	

Element	Description							
Memory Type	Memory Type	Description						
,	Register	R resource on PLC.						
	Discrete Input	I resource on PLC.						
	Discrete Output	Q resource on PLC.						
	Discrete Global	G resource on PLC.						
	Internal Coil	M resource on PLC.						
	Temporary Coil	T resource on PLC.						
	System Status	S resource on PLC.						
	Analog Input	AI resource on PLC.						
	Analog Output	AQ resource on PLC.						
	Clear I/O Fault	IOF resource on PLC.						
	Clear PLC Fault	PLF resource on PLC.						
Offset	Offset address where tag is located. Offset range depends on specific memory type and PLC model selected.							



Element	Description
Data Type	Available data types:
	• boolean
	• byte
	• short
	• int
	unsignedByte unsignedShort
	unsignedInt
	• float
	• double
	• string
	• binary
	See "Programming concepts" section in the main manual.
	Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]).
Arraysize	In case of array tag, this property represents the number of array elements.
	 In case of string tag, this property represents the maximum number of bytes available in the string tag.
	Note: number of bytes corresponds to number of string characters if Encoding property is
	set to UTF-8 or Latin1 in Tag Editor.
	character requires 2 bytes.
Conversion	Conversion to be applied to the tag.
	Conversion
	inv,swap2 Allowed Configured
	BCD AB->BA + ABCD->CDAB
	ABCD->CDAB ABCDEFGH->GHEFCDAB
	Inv bits
	Cancel
	Depending on data type selected, the list Allowed shows one or more conversion types.

lement	Description							
	Value	Description						
	Inv bits	inv: Invert all the bits of the tag.						
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)						
	Negate	neg: Set the opposite of tag value.						
		<i>Example:</i> 25.36 → -25.36						
	AB -> BA	swapnibbles: Swap nibbles in a byte.						
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)						
	ABCD ->	swap2: Swap bytes in a word.						
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)						
	ABCDEFGH	swap4: Swap bytes in a double word.						
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)						
	ABCNOP -	swap8: Swap bytes in a long word.						
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010000101101000011100101						
		\rightarrow 1 10000011100 1010101000010100010110110110						
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)						
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)						

I.

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Exporting Tags from PLC

The GE Intelligent Platforms SRTP Ethernet driver support the Tag Import facility.

Variables can be exported by the controller programming software Proficy Machine Edition,

selecting "Variables" tab, then right mouse click and from context menu select the Export option as shown in following figure.



In the following dialog select then the file name and the file location on the computer.

*	File	<u>E</u> dit	<u>S</u> ear	rch	Proj	ject	Targ	et	Vari	able:	5 <u>T</u>	ools	<u>W</u> ir	ndow	He	lp				
. 6	1 🖻		6	\checkmark	Ē	₽	$\langle 0 \rangle$		Ж	Þ	Ċ	Ω	2	\times	72		<u></u>	8 1	≽	8
1		H	►	\$		П	Ċ	±.	1	ŀ,	-11-	, //	-0-	$\langle \rangle$	-(1)-	-()-	-(5)-	-(R)-	₽ 	
Naviç	gator																		ņ	×
																				ľ
⊡∛)≣ Va ∃…∳	ariable Tarç GEP GEP	List: jet1 V_pr AI00	Sort _AI0 03	ed b 001	y Ado	dress,	, Filt	er =	No S	öyste	m Va	riabl	es						
	Eж	port																?	×	
	9	Speich	iern	<u>(</u>	🛛 Eig	jene	Datei	en					•	- 1		<u>*</u>	;; -			
	D	ateina	ime:	Ν	1ach	ine_E	ditior	∟Va	ariabl	es					[Sp	eich	ern]	
	D	ateity	D:	C	Comn	na Se	parat	ed V	/alue	s(*.c	sv)			-		Ab	brecł	nen		
•	AQ0001[008]										◄									
2	Opt	ions	1	Utili	ties		👌 Ma	nage	er j	E 10	Proje	ect	Ø	Vari	able	s	? I	nfoV	iew	Ţ

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag import	ers are available for this protocol. Please select the impo	rter type and continue.
Version	Туре	
Proficy Machine v1.0	Linear	
Tag Editor exported xml 1	.1 General	
Vatched dictionary file:		
Keep synchronized		

Importer	Description								
Proficy Machine v1.0	Requires an .csv file.								
Linear	All variables will be displayed at the same level.								
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.								
	Tags x								
	+ - 👗 🔞 🔊 >] 💽 🗞 🖬 🕅 1								
	Data Tag URI								

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags 🗙 Protocols						•	
+ - 👗 🕲 👂] [> 🗞 🖬 🚮	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ems used:6/10	000 Protocol: Show a	l 💽 Show all tags 🔅 🗌	
Data	Туре	Comment	^	Property		Value	
Modbus TCP:prot1	Prot1 Container						
Model: Modicon Modbus(1-ba	ased) container			Model Modicon Modbus(1-based)			
Holding Registers 1	unsignedShort			Protocol Modbus TCP:prot1			
Holding Registers 2	unsignedShort			✓ Dictionary			
- MRTU1	unsignedShort			Arra	у	false	
- MRTU2	unsignedShort			Arra	ay size	0	
MRTU3	unsignedShort			Arra	yindex.Subindex	400003	
- MRTU4	unsignedShort			Com	ment		
- MRTU5	unsignedShort			Data	a type	unsignedShort	
					/		

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
都	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tager X + -
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

GE Intelligent Platforms SRTP

The GE Intelligent Platforms SRTP driver can be used to connect the HMI device to the GE controllers through Ethernet connection using the native and proprietary SRTP communication protocol.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "GE Intelligent Platforms SRTP" from the list of available protocols.

GE Intelligent Platforms SR	ТР	—
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	18245	
Timeout	2000	
PLC Models		
90-30 311 90-30 331 90-70 731/732 90-70 771/772 90-70 780/781/782/70 90-30 340/341	89/914/915/924/925/928/935	•

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
IP Address	The IP address of the Ethernet interface of the controller
Port	Communication Port number for the Ethernet interface
Timeout	The time the protocol waits the answer from the controller before issuing a new retry.

Element	Description	Description											
PLC Models	List of compat configuring th	List of compatible controller models. Make sure to select the right model in this list when configuring the protocol.											
PLC Network	The protocol s To enable this node.	supports connec s, check the "PLC	tion to multiple C Network" ch	e controllers. eck box and prov	vide the config	uration per each							
	GE Intelligent Platforms S	SRTP	×]									
	V PLC Network		GE Intelligent Platforms	SRTP	— ×								
	Alias		Alias		OK Cancel								
	IP address Port	0.0.0.0	IP address	0.0.0.1									
	Timeout	2000	Port	18245									
	PLC Models	l	Timeout	2000									
	B0-30 311 90-30 331 90-70 731/732 90-70 771/772 90-70 780/781/782, 90-30 340/341	/789/914/915/924/925/928/935	PLC Models 90-30 331 90-70 731/732 90-70 771/772 90-70 780/781/782	2/789/914/915/924/925/928/935	Â								
	Slaves	Add	90-30 340/341		~								
	Slave Id	Model Alias											

Data Types

The import module supports variables of standard data types as per the following list.

- BOOL
- BYTE (8-bits unsigned integers)
- DINT (32-bits signed integers)
- DWORD (32-bit bit strings, displayed as unsigned integers)
- INT (16-bit signed integers)
- REAL (32-bit floating point data)
- STRING (character string)
- UINT (16-bit unsigned integers)
- WORD (16-bit bit strings, displayed as unsigned integers)



Note: User defined structure and predefined structures are not supported. 64-bit data are also not supported

Tag Conversion

Conversion to be applied to the tag.

Conversion			
inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ - ~	Configured Inv bits ABCD->CDAB
		[Cancel OK

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
Negate	neg : Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)

Value	Description
ABCNOP -> OPMDAB	swap8: Swap bytes in a long word. Example: 142.366 \rightarrow -893553517.588905 (in decimal format) 0 1000000110 000111001011101101000101101000011100001110000
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Special Data Types

The GE Intelligent Platforms SRTP driver provides one special data type called "Node Override IP".

The Node Override IP allows changing at runtime the IP address of the target controller you want to connect. This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

If the IP Override is set to 0.0.0.0, all the communication with the node is stopped, no request frames are generated anymore.

If the IP Override has a value different from 0.0.0.0, it is interpreted as node IP override and the target IP address is replaced at runtime with the new value.

In case the panel has been configured to access to a network of controllers, each node has its own Override variable.



Note: the IP Override values assigned at runtime are retained through power cycles.

GE Intelligent Platforms SRT	ſP			×
GE Intelligent Platforms SRT	P			
Memory Type	Offset	SubIndex		
Node Override IP 👻	0	0 -		
Data Tura	America	Conversion		
	Arraysize	Conversion		
unsignedByte 🛛 👻	8		+/-	
	0	K Cancel	Apply Help	

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

/ `	1:Page1	Υ	Protocols	· · ·	Tags	×														
+	- (2	60	>]	[>	A '∋B	BX	齖	R	P- Sear	ch		7 F	Filter by	y: Data		~	Items us	ed:11/10	000
Data	E Intellio	gent	^ Platforms	SRTP:p	orot1	Type Contai	ner		Tag n	ame		Tag	URI	[Dictionary				Pro	pe
	⊿ Slave Alias: Mode	id: 1 Nod: 1:90	192.168.0 e1 -30 311	.1		Contai	ner		1				_							
		1003	1			short			Node	1/AI0031		1?AI	?31?s <mark>ł</mark> or	rt						
	- 4	1003	2			short			Node	1/AI0032		1?AI	?32?sl or	rt						
	- 4	1003	3			short			Node	1/AI0033		1?AI	(?33?sl o	rt						
	- 4	1003	4			short			Node	1/AI0034		1?AI	(?34?sł or	rt						
	-0	1_In	ιp			boolea	n		Node	1/C1_Imp		1?I?	1?bool a	an						
	- c	2_In	ιp			boolea	n		Node	1/C2_Imp		1?1?	3?boolea	an						
	- c	3_In	ιp			boolea	n													
	-I	0000	4			boolea	n		In	nport Tag(s) - Network		- 1						×	1
L	- N	1001	02			boolea	n						_							
	- N	1001	03			boolea	n			-										
L	- N	1001	04			boolea	n			Please se	lect one or m	ore no	odes to a	issociat	e this dictio	onary and	contini	ue.		
	Slave Alias: Mode	id: 1 Nod: 1:90	192.168.0 e2 -30 311	.2		Contai	ner		Slave	id ^	Model	1	Alias							
L .									192.1	168.0.1	90-30 31	1	Node1							
									192.1	168.0.2	90-30 31	1	Node2							
4									Fo	r each sele	cted node imp	port or	nly symbo	ols mat	ching that	node Of	<	Car	ncel	

6

Note: Aliasing tag names is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name. The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

Exporting Tags from PLC

The GE Intelligent Platforms SRTP Ethernet driver support the Tag Import facility.

Variables can be exported by the controller programming software Proficy Machine Edition,

selecting "Variables" tab, then right mouse click and from context menu select the Export option as shown in following figure.

📌 Eile Edit Search	<u>Project Target Variables Tools Window H</u> elp
i 🔯 🖨 🖬 🍈 🗸	🖡 🖡 🔍 🗄 X 🖿 🛍 으로 X 🛛 🗍 🗖 🔩 🗖 😭
if 🕘 H 🕨 🛼	■ ■ • • ! ! ► ++ # • ↔ ⊕ ⊕ ⊕ ⊕ ⊕
Navigator	₽ ×
➡ Wariable List: Sort GEP A_AO1 GEP A_G1 GEP A_G1 GEP A_S1 GEP A_S1 GEP A_S1 GEP A_S1 GEP A_S1 GEP A_S1 GEP A_10003 GEP A10004 GEP A10005	New Variable > Paste Ctrl+V Import Import Autogenerated Tags Export Sort By Report Ctrl+T Filter By > Filter Configuration >
	Delete Unused Variables Refresh Variable References
📑 📝 Options 📝 Utili	ies 🛃 Manager 🕼 Project 🕄 Variables 🧣 InfoView

In the following dialog select then the file name and the file location on the computer.

🖋	File	<u>E</u> dit	<u>S</u> ea	rch	Proj	ject	Targ	et	Va <u>r</u> i	able:	s <u>T</u>	ools	<u>W</u> ir	ndow	He	lp				
. 6	1 🖻	: 日	6	\checkmark	Ē	₽		1	Ж	Ē	Ċ	Ω	2	\times	72			₽,t	≽	8
1		H.	►	₽		П	Ċ	±.		k	-11-	₩	-0-	()	-(1)-	-(1)-	-(S)-	-(R)-	0- 	
Naviç	gator																		ņ	×
E3	0≣ Va ≐… ∳	ariable Taro GEP GEP	List: get1 V_pr AI00	Sort _AIO 03	ed b 001	y Ado	dress,	, Filte	er =	No S	ōyste	m Va	riabl	es						1
	Eж	port																?	×	
	9	Speich	iern	<u>(</u>) Eig	jene	Datei	en					•	- 1		ř 🗄	!: •			
	D	ateina	ame:	N	1ach	ine_E	Editior	_Va	riabl	es					[Sp	eich	ern]	
	D	ateity	D:	0	Comn	na Se	eparat	ed V	alue	s(*.c	sv)			•		Ab	brecł	nen		
4		GED	AQ0 AQ0 AQ0	001[0 001[0 001[0)07])07])08]															•
2	Opt	ions	2	Utili	ties		👌 Ma	nage	er _	- 8	Proje	ect	Ø	Vari	able	es [? I	nfoV	iew	Ţ

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag import	ers are available for this protocol. Please select the impo	rter type and continue.
Version	Туре	
Proficy Machine v1.0	Linear	
Tag Editor exported xml 1	.1 General	
Vatched dictionary file:		
Keep synchronized		

Importer	Description		
Proficy Machine v1.0 Linear	Requires an .csv file.		
	All variables will be displayed at the same level.		
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.		
	Tags x		
	+ - 👗 🔞 🔊 >] 💽 🗞 🖬 🕅 1		
	Data Tag URI		

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags 🗙 Protocols						•
+ - 👗 🕲 👂] [> 🗞 🖬 🚮	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ems used:6/10	000 Protocol: Show a	l 💽 Show all tags 🔅 🗌
Data	Туре	Comment	^	Property		Value
Modbus TCP:prot1	Container			✓ Driver		
Model: Modicon Modbus(1-ba	ased) container			Mod	el	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Prot	ocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionar	ry	
- MRTU1	unsignedShort			Arra	у	false
- MRTU2	unsignedShort			Arra	ay size	0
MRTU3	unsignedShort			Arra	yindex.Subindex	400003
- MRTU4	unsignedShort			Com	ment	
- MRTU5	unsignedShort			Data	a type	unsignedShort
					/	

Toolbar item	Description		
	Import Tag(s).		
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project		
a a a a a a a a a a a a a a	Update Tag(s).		
	Click on this icon to update the tags in the project, due a new dictionary import.		
R	Check this box to import all sub-elements of a tag.		
	Example of both checked and unchecked result:		
	Tops: X + -		
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.		

Communication Status

The communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General Error	Error cannot be identified; should never be reported; contact technical support

The status codes supported for this communication driver are:

GE SRTP

The GE SRTP communication driver has been designed to connect HMI devices to GE PLCs.

The driver allows symbolic communication with GE PLC model PacSystemRx3i.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

ie SRTP		
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	18245	
Timeout	2000	
PLC Models		
90-30 311 90-30 331 90-70 731/732 90-70 771/772 90-70 780/781/7 90-30 340/341	782/789/914/915/924/925/928/935	•

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Port	Port number used by the driver. The default value is 18245 .
Timeout	Time delay in milliseconds between two retries in case of missing response from the server device.
PLC Models	SAIA PLC models available:

Element	Description
	• 90-30 311
	• 90-30 331
	 90-70 731/732
	 90-70 771/772
	 90-70 780/781/782/789/914/915/924/925/928/935
	• 90-30 340/341
	• 90-30 313
	 90-30 351/352/360/363/364
	• 90-70 788
	 90-30 350/374
	VersaMax CPU001
	VersaMax CPU002
	 VersaMax (CPU005, CPUE05)
	PACSystem RX3i
PLC Network	Multiple controllers can be connected to one HMI device. To set-up multiple connections, select PLC network and click Add to configure each node

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **GE SRTP** from the **Driver** list: tag definition dialog is displayed.

Memory Type	Offset Su	bIndex	
Register 👻	1		
Symbol Name	Data Type	Arraysize	
	unsignedShort	•	
Conversion			
	+/-		

Element	Description		
Memory Type	Memory Type	Description	
1960	Register	unsigned 16 bit data register (default)	
	Discrete Input	1 bit data input (default)	
	Discrete Output	1 bit data output (default)	
	Discrete Global	1 bit data global (default)	
	Internal Coil	1 bit data coil (default)	
	Temporary Coil	1 bit data coil (default)	
	System Status	1 bit data status	
	System Status A	1 bit data status	
	System Status B	1 bit data status	
	System Status C	1 bit data status	
	Analog Input	unsigned 16 bit data input (default)	
	Analog Output	unsigned 16 bit data output (default)	
	SYMBOL	1 bit data symbol (default)	
	Node Override IP	unsigned 8 bit array (see Special Data Types for mode details)	
Offset	This parameter is the address on the physical memory of the controller. The range for any memory type depends on the PLC model.		
SubIndex	This allows resource offs	set selection within the register.	
Data Type	Available data types:		
	• boolean		
	• byte		
	• snort • int		
	 unsignedByte 		
	 unsignedShort 		
	 unsignedInt 		
	float		
	double string		
	See "Programming cond	epts" section in the main manual	
	Note: To defin	e arrays, select one of Data Type format followed by square brackets.	

Element	Description			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 			
	Note: number of bytes or Latin1 in Tag Editor. If Encoding property is bytes.	corresponds to number of string characters if Encoding property is set to UTF-8 set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2		
Conversio	onversio Conversion to be applied to the tag. Conversion			
n				
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCD=>GHEFCDAB Inv bits Cancel OK Cancel OK		
	Value	Description		
	Inv bits	inv : Invert all the bits of the tag.		
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	neg : Set the opposite of tag value. <i>Example:</i>		
		25.36 → -25.36		
	AB -> BA	swapnibbles : Swap nibbles in a byte. <i>Example:</i> $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)		
	ABCD -> CDAB	swap2: Swap bytes in a word.		
		Example: $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)		
	ABCDEFGH ->	swap4: Swap bytes in a double word.		

Element	Description	
	Value	Description
	GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011001000101101000011100100101
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion and	click +. The selected item will be added to list Configured .
	If more conversions are	e configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.

Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

1

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

Memory Type	Offset	SubIndex		
Node Override IP 👻	0	0 *		
Symbol Name	Data Type		Arraysize	
	unsignedBy	te [🛛 🔻	4	
Conversion				
	+/-			
	+/-			

Tag Import

For GE PLC model PacSystemRx3i it is possible to create symbolic variables.

To create a new variable, right-click on the **Variables View** and select **New Variable**. To edit an existing variable, rightclick on it and then select **Properties**.



In both cases, the variable's Properties Inspector dialog will appear as shown below.

Variable [Target1]	
Name	MySymbolicW0RD
Description	
Publish	External
Array Dimension 1	0
Data Source	GE FANUC PLC
Ref Address	
Data Type	WORD
Current Value	0
Initial Value	0
Default Display Format	Decimal
Retentive	True

Important: In order for a symbolic variable to be visible to this driver, **Publish** must be set to **External**. The access must be set to **Read/Write**.

To export these variables from **PACSystem** programming software, right click on **Variable list** (or on selected variables) and click **Export**.

Variable List: Sorted by Name, Filte	r — No Sustan Variablas	. 1	
MyMappedWord1	New Variable		1
MyMappedWord2	Paste	(CDI)+V	
GEP MyMappedWordArray_1D	Import		
How MyMappedWordArray_2D	Export		
MySymbolicBOOL	Sort By		
- MySymbolicBool01	Report	Ctrl+T	
MySymbolicBool02	Eiltor Pu		
MySymbolicBool03	Filter Dy Filter Configuration		
MySymbolicBool04	Filter Configuration	·	
MySymbolicBool05	Delete Unused Variab	les	
MySymbolicBool06	Refresh Variable Refe	erences	
MySymbolicBool07			
MySymbolicBool08			
MySymbolicBool09			
MySymbolicBool10			
MySymbolicBool100			
MySymbolicBool101			
MySymbolicBool102			
MySymbolicBool103			
MySymbolicBool104			
			+

In the **Save as Type** drop-down list, select **Comma Separated Variable** (*.csv) as the export file type. The dialogs should appear as shown below.

Export			? ×
Save in: 🗀	ExportData	- 🗢 🖻 🖻	• •
			60550 AR 1
File name:	MyExportedVariables	F	Save
Save as type:	Comma Separated Values(".csv)		Cancel

Select the driver in the Studio tag editor and click on the "Import tag" button to start the importer.

+ - 2 0	🔊 🔰 [> 🖧 🖻	R P- Search	YFilter by: Data
Data	Type Tag name	Tag URI Dictionary	
GE SRTP:prot1 Model: 90-30 311	Container		
	HMIStudio		×
	Multiple tag importers	are available for this protocol. Please select	the importer type and continue.
	Version	Туре	
	Proficy Machine v1.0	Linear	
	Tag Editor exported xml 1.1	General	
	Watched dictionary file:		
	Keep synchronized		
			OK Cancel

Select Linear and locate the .csv file, then confirm.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in the following figure.

🖹 🛃 🔲 Recursive	: 🔎 Search 🍸 Filter by:	Data	I	•
Data	Туре	•	Property	Value
GE SRTP:prot1	Container		Model	90-30 331
Model: 90-30 331	Container	=	Protocol	GE SRTP:prot1
- AI0031	short		,	
- AI0032	short			
- AI0033	short		8	
- AI0034	short		0.00	
C1_Imp	boolean			
C2_Imp	boolean			
C3_Imp	boolean			
- I00004	boolean			
- M00102	boolean			
M00102	hooloop	*		
Ready		D	ictionary Name:	[GE SRTP prot1] ME_Tag_Export Protocol Name: GE SRTP



In case of **Online Changes** performed on PLC side, the tag database must be updated manually to correctly **Read** from PLC.

Write operations do not need a database update.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Hitachi SER

HMI devices can be connected to a Hitachi EH/EHV PLC as the network master using this communication driver.

This driver has been designed for serial connection to the programming port of the PLC.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Hitachi SER" from the list of available protocols.

The driver configuration dialog box is shown in figure.

Hitachi SER		×
PLC Network	Comm OK	
Alias	Cancel	
PLC Models		_
EH150 CPU104		A
EH150 CPU208		
EH150 CPU316	-	
EH150 CPU516		
EH150 CPU548		
EHV CPU16.32.64		T

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
PLC Models	Select from the list the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.
PLC Network	The protocol allows the connection of multiple controllers to one HMI. To set-up multiple connections, check "PLC network" checkbox and create the list of controllers pressing the "Add" button. You must specify the node ID for each device you want to connect.

Element	Description		
	Hitachi SER PLC Network Alias PLC Models FISO CPU208 EHISO CPU208 EHISO CPU316 EHISO CPU516 EHISO CPU56 EHISO CPU56 Slaves Slave Id Mo PLC Mod Hitachi SER Alias Slave ID PLC Mod EHISO EHIS	Comm OK Cancel	
Comms.	Opens the serial Comm Parameter Dialog Port Baudrate Parity Data bits Stop bits Mode	port configuration paramete	rs as shown in figure.
Port	Serial port select	tion	
		Series 400	Series 500
	com1	PLC Port	Serial Port
	com2	PC/Printer Port	Option Module
Baud rate, Parity, Data bits, Stop bits	Communication	parameters for serial commu	inication
Mode	Serial port mode • RS-232, • RS-485 (2 • RS-422 (4	; available options: 2 wires) 4 wires)	

Tag Name Aliasing in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 🔏 🖻 🖻 >] [> 🗞 🖻	R [₽- Search	Y Filt	ter by: Data	~	Items used:8/10000 Protocol: Show all	G
Data Aliachi SER:prot1	Type Container Container	Tag name		Tag UR	I		Prope
Model: Install - Anterna - The December - T	diari Radi orogradiri orogradiri orogradiri orogradiri orogradiri	Paulit Composition Raulite (************************************	Import Ta	g(s) - Network select one or mo	re nodes to ass	sociate this dictionary and continue.	×
Water_Level _Slave id: Alias: Node2 Model:	unsignedShor Container	t Node If Water_Level	Slave id	Model	Alias Node1 Node2		
			For each se	elected node imp	ort only symbol	s matching that node	ancel

Note: Tag name aliasing is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

The Hitachi SER communication driver supports importing tags from the PLC programming software. The tag import filter accepts symbol files with extension ".txt" created by the Actwin-H programming tool.

In the Actwin-H Software, click on the menu "Operations" then "Export symbol list" and then select the project which should be exported as shown in figure.





In the Tag Editor select the driver and click on the "Import tag" button to start the importer

Once the importer has been selected, locate the symbol file and click Open.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in figure.

	1:Page1 Protocol	s Tags 🗙				
+	- 260	>] [> 🖏	B [3] [3] B	CP- Search	Y Filter by: Data 🗸 🗸	Items
Data	I	Туре 1	Tag name	Tag URI	Dictionary	
4 ¦	litachi SER:prot1 Model:	Container				
	str	string-16 s	str	0??4?0?string-16		
	ARRAY_WORD[1]	short-8 A	ARRAY_WORD[1]	0?ARRAY_WORD[1]?4?1?short-8		
	ARRAY_WORD[2]	short-8 A	ARRAY_WORD[2]	0?ARRAY_WORD[1]?4?2?short-8		
	ARRAY_WORD[3]	short-8 A	ARRAY_WORD[3]	0?ARRAY_WORD[1]?4?3?short-8		
	ARRAY_WORD[4]	short-8 A	ARRAY_WORD[4]	0?ARRAY_WORD[1]?4?4?short-8		
	- MDW2	unsignedShort N	MDW2	0?ARRAY_WORD[1]?4?5?unsignedSh	ort	
	MDW3	unsignedShort N	MDW3	0?ARRAY_WORD[1]?4?6?unsignedSh	ort	

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured for communication

Error	Notes
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
Hitachi ETH

This communication driver has been designed to support communication to Hitachi controllers with Ethernet connection. Hitachi controllers must either have an on-board Ethernet port (EHV CPU) or be equipped with an appropriate Ethernet interface (EH-ETH, ET-ETH2 or OB- ETH).

The communication driver supports both TCP/IP and UDP/IP communication protocols.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Hitachi ETH" from the list of available protocols.

The driver configuration dialog is shown in figure.

Hitachi ETH		×
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	3004	
enable waiting		
time between request (ms)	300 •	
Connection Type	UDP ~	
PLC Models		
EH150 CPU316	^]
EH 150 CPU516 EH 150 CPU548 EH V CPU16.32.64 EH V CPU128 EH-W 10.23	~	

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
IP address	Ethernet IP address of the controller
Port	Port number used for the communication. Default value 3004 and it corresponds to the default setting of Hitachi controllers.
Enable waiting	Introduces a wait time between two communication requests
Time between request	Wait time between two requests if enable waiting option has been activated

Element	Description		
Connection	UDP: use communication based on UDP/IP protocol		
type	TCP: use communication based on TCP/IP protocol		
PLC Models	Select from the list the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.		
PLC Network	To set-up multiple connections, check "PLC network" checkbox and create the list of controllers pressing the "Add" button. The IP address for each device you want to connect must be specified.		
	Alias 0 . 0 . 0 . 0		
	Port 3004		
	enable waiting Hitachi ETH ×		
	OK OK Connection Type Alias PLC Models IP address IP address 0 · 0 · 0 · 1 EH150 CPU516 Port EHV CPU5.32.64 Port EH-V CPU5.23 Connection Type Slaves PLC Models Slave Id EH150 CPU536 EH150 CPU536		

Controller Configuration

The PLC must to be properly configured to support either UPD/IP or TCP/IP communication using port numbers 3004, 3005, 3006 or 3007.

Tag Name Aliasing in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 🕺 🖻 🔊 🚺 🖓	R P- Search	Tilter by: Data V Items used:8/10000 Pro	tocol: Show all
ata T 4 Hitachi ETH:prot1 C Slave id:	Tag name Container	Tag URI	Prop
Ailas: Node1 Model:	tari taliki daane	(Committee) (Committee)	
This Proceedings of This Proceedings of This Proceedings of This Proceedings of the This Pro	regression tests (%), 250,840 regression fields (%), 250,840 regressioner fields (%), 250,840 regressioner fields (%), 250,840 regression fields (%), 250,840	Import Tag(s) - Network Please select one or more nodes o associate this dictionary and o	ontinue.
Slave: (to Get Alae: (to Get Model:	nsignedshort Node Ijvivater_Level	Slave id Model Alkas Node1 Node2	
		For each selected node import only symbols matching that node	



Note: Tag name aliasing is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

The Hitachi ETH communication driver supports importing tags from the PLC programming software. The tag import filter accepts symbol files with extension ".txt" created by the Actwin-H programming tool.

In the Actwin-H Software, click on the menu "Operations" then "Export symbol list" and then select the project which should be exported as shown in figure.

File Edit View Insert O	perations Communication Tools	Help
12 🖻 🖪 🖪 🞒 [Force I/O	D D D D I + 0 E
	Force output	
	Release all forced	
	Move symbol addresses	Program main
Program main	Exchange addresses	Main
Final Symbols	Exchange addresses	
Hardware confi	Export symbol list	Project U_65535_2
and a settings	Edit data memory	Lees de actue de wat

In the tag editor select the driver and click on the "Import tag" button to start the importer

si.	ProjectView 4 ×	1:Page1 Tags x	
Obj	+ - ∉ ∧ ∨	+ - 🐰 🕲 👂 👌 🚯 👸 R 🔎 Search 🛛 🗍 Filter by: Data	~
ObjectView Tag Cross Reference		+ - Container Filter by: Data Data Type Tag name Tag URI Comment Hitachi ETH:prot1 Container Container Model: HMIStudio X Image: Multiple tag importers are available for this protocol. Please select the importer type and continue. Version Type ActWin v1.0 Linear Tag Editor exported xml 1.1 General	~
	 Scheduler Scheduler MultiLanguage Screen Saver Database Links Data transfers An Interfaces Security Recipe Dictionaries Keypads 	Watched dictionary file:	

Once the importer has been selected, locate the symbol file and click Open.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in figure.

1:	Page1 Protoco	ols Tags 🗙				
+	- 2 6 6)) [> 💱	B [B] [J] B	P- Search	▼ Filter by: Data ✓ Iten	ns
Data		Type T	Tag name	Tag URI	Dictionary]
⊿ Hi Ma	tachi ETH:prot1 odel:	Container				
	- str	string-16 s	str	0??4?0?string-16		
	- ARRAY_WORD[1]	short-8 A	ARRAY_WORD[1]	0?ARRAY_WORD[1]?4?1?short-8		
	- ARRAY_WORD[2]	short-8 A	ARRAY_WORD[2]	0?ARRAY_WORD[1]?4?2?short-8		
	ARRAY_WORD[3]	short-8 A	ARRAY_WORD[3]	0?ARRAY_WORD[1]?4?3?short-8		1
	ARRAY_WORD[4]	short-8 A	ARRAY_WORD[4]	0?ARRAY_WORD[1]?4?4?short-8		
	MDW2	unsignedShort M	MDW2	0?ARRAY_WORD[1]?4?5?unsigned5	Short	1
	- MDW3	unsignedShort N	MDW3	0?ARRAY_WORD[1]?4?6?unsigned5	Short	
						1

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured for communication
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller

Error	Notes
	resources
General Error	Error cannot be identified; should never be reported; contact technical support

IDEC Maintenance

IDEC Maintenance communication driver has been designed to connect HMI devices to IDEC PLC through Serial or Ethernet connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the $\ensuremath{\text{PLC}}$ list.

The protocol configuration dialog is displayed.

PLC Network	Comm	ОК
Alias		Cancel
Node ID	1	
Media	Serial 🔹	
IP address	0.0.0.0	
Port	2101	
Timeout(ms)	500	
PLC Models		
FC6A - MicroSmar FC6A - MicroSmar FC5A - MicroSmar FC4A - MicroSmar FT1A - SmartAvis	t Plus t t Pentra t	

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
Node ID	Serial node associated to PLC.
Media	Allows the selection of transport Media.

Element	Description	
	 select Serial to connect via serial line select Ethernet to connect via TCP/IP 	
IP address	IP address of PLC (only available if Ethernet media is selected)	
Port	Port number of PLC	
Timeout (ms)	Time delay in milliseconds between retries in case of missing response	
PLC Models	PLC model available:	
	FC6A - MicroSmart Plus	
	FC6A - MicroSmart	
	FC5A - MicroSmart Pentra	
	FC4A - MicroSmart	
	FT1A - SmartAxis	
PLC Networ k	Enable configuration of multiple connections.	

Element	Description	
	IDEC Maintenance	23
	PLC Network	Comm) OK
	Alias	Cancel
	Node ID	-1
	Media	Ethernet
	IP address	IDEC Maintenance
	Port	ОК
	Timeout(ms)	Alias Cancel
	PLC Models	Node ID 1
	FC6A - MicroSmart FC5A - MicroSmart Pent	tra IP address 0.0.1
	FT1A - SmartAxis	Port 2101
	Slaves	PLC Models
	Slave Id	M FC6A - MicroSmart FC5A - MicroSmart Pentra FC4A - MicroSmart FT1A - SmartAxis
Comm	If clicked displays the comm media is selected)	nunication parameters setup dialog (only available if Serial

Element	Description			
	Comm Parameter Dialog			
			ОК	
	Port	com1	-	
	Baudrate	9600	•	
	Parity	even	_	
	Data bits	7	·	
	Stop bits	1	•	
	Mode	RS-485	-	
	Element		Parameter	
	Port		Serial port selection.	
			COM1: On-board port	
			 COM2: Optional Plug-in module plugged on slot#1 or slot#2 	
			 COM3: Optional Plug-in module plugged on slot#3 or slot#4 	
	Baudrate, Parity, Data Stop bits	Bits,	Serial line parameters.	
	Mode		Serial port mode. Available modes:	
			• RS-232.	
			• RS-485 (2 wires).	
			• RS-422 (4 wires).	

Tag Editor Settings

In Tag Editor select IDEC Maintenance protocol.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

IDEC Maintenance				
Memory Type	Offset	SubIndex		
I -Input 🔹	0	0 👻		
Data Type	Arraysize	Conversion		
unsignedByte 🗸	0	1	+/-	

EX05

Element	Description					
Memory Type	Memory Type	Description				
1,960	I - Input	I resources. Corresponding to internal	digital Input point.			
	Q - Output	Q resources. Corresponding to internal digital Output point.				
	M - Internal Relay	M resources. Corresponding to PLC int	ernal memory.			
	R - Shift Register	S resources. Corresponding to PLC sh	ift registers.			
	T - Timer	T resources. Corresponding to PLC tim	iers.			
	TC - Timer Current Value	TC resources. Corresponding to PLC ti values.	mer current			
	TP - Timer Preset Value	TP resources. Corresponding to PLC ti values.	mer preset			
	C - Counter	C resources. Corresponding to PLC counters.				
	CC - Counter Current Value	CC resources. Corresponding to PLC counter current values.				
	CP - Counter Preset Value	CP resources. Corresponding to PLC counter preset values.				
	D - Data register	D resources. Corresponding to PLC da	ta registers.			
Offset	Starting address for the Tag. The possible range depend on PLC model selected.					
Subindex	This allows resource offset sele	ection depending on the selected data ty	pe.			
Data Type	Data Type	Memory Space	Limits			
	boolean	1-bit data	0 1			
	byte	8-bit data	-128 127			
	short	16-bit data	-32768 32767			
	int	32-bit data	-2.1e9 2.1e9			
	int64	64-bit data	-9.2e18 9.2e18			
	unsignedByte	8-bit data	0 255			
	unsignedShort	16-bit data	0 65535			
	unsignedInt	32-bit data	0 4.2e9			

Element	Description				
	Data Type	Memory Space	Limits		
	uint64	64-bit data	0 1.8e19		
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38		
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308		
	string	Array of elements containing character selected encoding	code defined by		
	binary	Arbitrary binary data			
	Note: to define array brackets like "byte[]"	s. select one of Data Type format followe , "short[]"…	ed by square		
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one 				
Conversion	Conversion to be applied to the	e tag.			
	Conversion				
	inv,swap2 Allowed BCD AB->B ABCD- ABCD- ABCDE Inv bit Depending on data type selected	d Configured Inv bits ABCD->CDAB FGH->GHEFCDAB Cancel OK ed, the list Allowed shows one or more of	conversion types.		

Element	Description			
	Value	Description		
	Inv bits	inv : Invert all the bits of the tag. <i>Example:</i> $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	neg : Set the opposite of tag value. <i>Example:</i> $25.36 \rightarrow -25.36$		
	AB -> BA	swapnibbles: Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)		
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.		
	GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)		
	ABCNOP - > OPMDAB	swap8: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011010001011101000011100101		
	BCD	(in binary format) bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)		

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	z	ß	ß	>]	₽	A 9B	B>	63	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio X					
Multiple tag importers are available for this protocol. Please select the importer type and continue.					
Version	Туре				
WindLDR v1.0	Linear				
Tag Editor exported xm	l 1.1 General				
Watched dictionary file:					
Keep synchronized	Keep synchronized				
			OK	Cancel	

Туре	Description				
WindLDR v1.0	Requires a .csv file.				
Linear	All variables will be displayed at the same level.				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				
	Tags ×				
	+ — 🎽 🕲 刘 🚺 🌡 🖬 🕅				
	Data Tag URI				

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols			
+ - 👗 🕲 🖉 >] [> 🔩 🖻	R Search Tritter by:	Data	Show all 💽 Show all tags 🔅 🗌
Data Type	Comment	Property	Value
Modbus TCP:prot1		Y Driver	
Model: Modicon Modbus(1-based)		Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort		Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort	Holding Registers 2 unsignedShort		
Holding Registers 3 unsignedShort		Amount	false
MRTU1 unsignedShort		Array	Idise
MRTU2 unsignedShort		Array size	0
MRTU3 unsignedShort		Arrayindex.Subind	lex 400003
MRTU4 unsignedShort		Comment	
MRTU5 unsignedShort		Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
6 3	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Tage X
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

J1939

Use this communication driver to connect HMI devices to CAN networks including devices communicating with SAE J1939.

Please note that changes in the communication protocol specifications or J1939 hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in J1939 hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Protocol Editor Settings

Select Add [+] in Protocol Editor and select J1939.

The driver configuration dialog is shown in figure.

J1939 ×		
		ОК
CAN channel	can0 ~	Cancel
ISO ECU instance	0	
ISO Function Instance	0	
Claiming Address	81	
Baud Rate (kbps)	250 ~	
Timeout (ms)	2000	
Disable Filter on Dest		
UDS Response Timeout	500	
UDS Access Timeout	0	
UDS Tester Present Timeout	10000	
PLC Models		
J1939 Device		

Element	Description
CAN	Configure the CAN Channel.
Channel	CAN interface is available only with a proper option module.
	UN31 platforms allow only one module, select Can0.
	UN30 platforms allow up to two modules, select Can0 or Can1.
ISO ECU Instance	Identifier of the equipment in the J1939 network (in case several HMI are coexisting in the network)
ISO Function Instance	Identifier of the function in the network (in case more than one device is providing the same functionality)
Claiming Address	Default value of the address of the equipment used as starting value for the Address Claim algorithm
Baud Rate (kbps)	Baud rate of the CAN bus (typical is 250)
Timeout (ms)	Timeout for the validity of received values. After the time indicated since last reception any value is declared "old" and its quality changed to "bad". The value 0 disables the timeout check

Tag Editor Settings

In Tag Editor select the protocol "J1939" from the list of defined protocols and add a tag using [+] button. Tag settings can be defined using the following dialog:

latatype	Arraysize	Conversion		
boolean 👻	0		+/-	
Parameter Group Number	Index		Selector type	
	1		NONE	•
SO Ecu-Function instance	ISO Function	n	Vehicle System / Instand	ce
0	0		0	

Element	Description			
Data Type	Data Type	Memory Space	Limits	
	boolean	1 bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
	unsignedByte	8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	0 4.2e9	
	float	IEEE single-precision	1.17e-38 3.40e38	
		32-bit floating point type		
	string	Array of elements containin selected encoding	g character code defined by	
	Note: to o brackets	define arrays, select one of D like "byte[]", "short[]"…	ata Type format followed by squa	re
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 			
	Note: number of by property is set to U If Encoding propert character requires	tes corresponds to number o TF-8 or Latin1 in Tag Editor. y is set to UCS-2BE, UCS-2L 2 bytes.	f string characters if Encoding .E, UTF-16BE or UTF-16LE one	
Conversion	Conversion to be a	pplied to the tag.		
	Conversion			
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	Configured Inv bits ABCD->CDAB	
	Depending on data types.	type selected, the list Allow	ed shows one or more conversion	

Element

Description	
Value	Description
Inv bits	inv : Invert all the bits of the tag.
	<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)
Negate	neg: Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD ->	swap2: Swap bytes in a word.
CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFG	swap4: Swap bytes in a double word.
H -> GHEFCDA B	Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
ABCNOP	swap8: Swap bytes in a long word.
-> OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 1000000110 0001110010111011001000101101000011100101
	→ 1 10000011100 10101000010100010110110110110010110110000
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
	Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Element	Description	
	Select conversion and click +. The selected item will be added to list Configured .	
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).	
	Use the arrow b	outtons to order the configured conversions.
Parameter Group Number	Parameter Group Number value	
Index	Index value	
Selector Type	When adding tags it can be necessary to duplicate them to read data coming from several devices generating same physical quantity. In this case the Address of the tag must be edited. The Tag Editor dialog is shown in figure:	
	In case of duplication of the tag, the selection of incoming data can be done using one of following methods:	
	NONE	Selector Type not selected
	INSTANCE	uses a defined bitfield value in data of PGN to distinguish between the possible sources. The value of received bitfield is compared with parameter "Vehicle System / Instance" for matching
	DEVICE	uses the source address to find out the device sending the PGN based on Address Claim algorithm. The devices are selected based on parameter "ISO function"
	ADDRESS	uses directly the source address as it is to select the source. The received source address is compared with parameter "ISO Ecu – Function Instance"
ISO Ecu- Function Instance	Instance of ISO Ecu-Function checked with Selector Type "DEVICE"	
ISO Function	ISO Function pa	arameter
Vehicle System/Instance	Vehicle System / Instance parameter used with Selector Type "INSTANCE"	

J1939 PGN Definition File

J1939 can connect hundreds of different devices offering access to thousands of different physical values. The standard defines several hundred PGNs for various applications. However, many devices use manufacturer-specific PGN definitions.

In order to manage this complex application scenario, the J1939 driver loads the PGN definition table at startup from a configuration file. The file with the PGN definition table is "J1939_pgnTable.csv" located in the folder "*target\protocols*"; it is loaded automatically from disk when downloading the project.

The file containing the PGN defined by the standard protocol specification is placed in the proper folder when the driver is installed. It can be edited adding or removing PGN definitions. The user must respect the following rules:

- the file contains most of the PGN defined by the standard. Custom PGN and SPN can be added assigning free indexes.
- description of a PGN is composed by a PGN declaration line followed by a list of Field description lines

PGN declaration line

PGN: Name, PGN number, DefaultPriority, DefaultRate, InstanceIndex, Direction [, PGN request rate]

Name	Name of the PGN
PGN number	Number code of PGN
DefaultPriority	Transmission priority (output PGN)
DefaultRate	Transmission rate (output PGN)
Instance Index	Index of instance (output PGN)
Direction	INPUT/OUTPUT
PGN request rate	Optional parameter. Time in milliseconds. If PGN not received in the meanwhile, it is requested

Example of PGN declaration:

PGN: Torque/Speed Control 1, 0, 3, 100, 0, INPUT

// Torque/Speed Control 1 id PGN nr.0, its default priority is 3 and default transmission rate is 100 ms. Instance Index is 0 and direction is INPUT

Field declaration line

```
FieldIndex, FieldName, FieldPosition, FieldBitSize, SPN Conversion, AccessType,
FieldDataType
```

FieldIndex	Index of field in the PGN
FieldName	Name of the field
FieldPosition	N (1 to 8) byte position
	N.M (1.1 to 8.8) bit position
	N-M (N from 1 to 7, M from 2 to 8) byte range
FieldBitSize	1-64
	number of bits of the field
SPN	SPN conversion is indicated by "SPN"index es. SPN79
Conversion	SPN0 indicates a raw copy of data
AccessType	Defines usage of field in combination with PGN direction.

If PGN direction is declared as OUTPUT, the fields can be only used for write operations.

If PGN direction is declared as INPUT the fields can always be read. In case they are written the behavior is described below.

PGN Direction	Access Type	Behavior
OUTPUT	WRITE	the PGN is sent immediately with current value of the fields
	READ_ ONLY	the PGN is sent as soon as all the fields are written with a fresh value
	REPLY	
INPUT	READ_ ONLY	Error
	REPLY	the PGN is sent only if it was received almost once, with update value of the written field
	WRITE	the PGN is sent immediately with current value of the fields

FieldDataType

boolean-nn
byte
unsignedByte
short
unsignedShort
int
unsignedInt
float
double
string-nn

Boolean

Example of Field declaration:

1, Engine Override Control Mode, 1.1, 2, SPN0, READ_ONLY, unsignedByte

SPN declaration line

SPN: index, constK, constL, type [,bigEndian]

index	index of SPN
constK	SPN conversion parameters
constL	the conversion applied when reading is:
	var(type) = raw value * constK + constL
	the conversion applied when writing is:
	raw value = (var(type) - constL) / constK
type	bits
	char
	uchar
	short
	ushort
	int
	uint
	float
	double
	longlong
	ulonglong
	float80
bigEndian	Optional parameter. Defines if endianity conversion is needed on raw data before applying the SPN conversion.
	0 default endianity, do not change
	1 apply endianity transformation

Example of SPN declaration:

SPN:, 79, 0.03125, -273, short, 1

Send DM1 alarms

Diagnostic Trouble Codes (DTC) are part of SAE J1939 standard which is widely used as a communication and diagnostic protocol for the physical CAN layer in automotive applications. The SAE J1939 standard provides 12 Diagnostic Messages (DMs).

The HMI device can send the DM1 message through the JavaScript using the *invokeProtocolCommand()* interface.

invokeProtocolCommand(protID, command, params)

Example of usage:

var params = "B1 B1 SPN FMI CM"; // to be filled with real data

tagMgr.invokeProtocolCommand(protID, "SetAlarm", params);

Parameter	Descripti	on					
protID	Protocol id	Protocol id, available from the protocol editor (e.g. "prot2")					
command	Command	l to exec	ute, it can be:				
	• "Se	etAlarm"					
	• "Re	esetAlarr	n"				
params	It is a strin	g that co	ntains five numbers separated by a spaces (e.g.: "4 4 520803 31 53")				
	Byte	Bits	Description				
	B1	8-7 6-5 4-3 2-1	Malfunction Indicator Lamp Status Red Stop Lamp Status Amber Warning Lamp Status Protect Lamp Status				
	B2	8-7 6-5 4-3 2-1	Flash Malfunction Indicator Lamp Flash Red Stop Lamp Flash Amber Warning Lamp Flash Protect Lamp				
	SPN	-	Suspect Parameter Number (as per J1939 specifications)				
	FMI	-	Failure mode identifier (as per J1939 specifications)				
	СМ	-	Conversion Method (as per J1939 specifications)				

Tag Import

The J1939 driver can import tag information from any CSV file, following same rules of PGN definition file and maintain several dictionaries for different scenarios.

The user can also import the whole "J1939_pgnTable.csv" and use only one large dictionary.

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	X	Đ	ß	>]	₽	A 9B	B>	ŧ3	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected

HMIStudio	HMIStudio X						
Multiple tag importe	ers are available for this protocol. Please select the importer type and continue.						
Version	Туре						
J1939 v1.0	Linear						
J1939 DBC++ v1.0	Linear						
Tag Editor exported xml 1.	1 General						
Watched dictionary file:							
Keep synchronized							
	OK Cancel						

Туре	Description				
J1939 v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.				
J1939 DBC+ v1.0 Linear	Requires a .dbc file generated by Vector CANdb++ Editor All the frames will be generated with type = Rx, so frames created for transmission must be reedit after importation				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				

The tags resulting from the import process may be used as they are if there is only one source for such value in the network. When several sources are supplying the same value the associated tags must be duplicated and named using one of the addressing methods shown in the Tag Editor chapter.

Communication Diagnostic

The error types supported for this communication driver are:

Error Class	Error					
Configuration Errors	invalid CAN channel					
	cannot read MACID					
	Unable to access the PGN Table					
	Unable to get the PGN file path					
	SPN conversion not supported					
	Sending PGN with dynamic field length not supported					
	Preparing PGN field for sending failed					
	Writing a read-only tag					
	The output PGN can't be read					
	invalid offset in PGN					
	Not byte boundary on dynamic field					
	Something wrong with the PGN data block size					
	Too many bits to use					
	Not byte boundary on dynamic field					
	SPN conversion not supported					
Runtime Errors	Communication Failure > Can't send the APL PGN message					
	Not Connected > The PGN for the command reply has not been received yet					
	Not Connected > PGN block not registered					
	Not Connected > the value never received					
	Timeout Error > timeout on the value refresh					

Error Class	Error				
Tag Definition Errors	there must be 7 tag specification fields				
	PGN field missing				
	SPN definition not found in the table				
	index field missing				
	ecuFunctionInstance field missing				
	function field missing				
	classOrInstance field missing				
	icomType field missing				
	Can't access protocol common parameters				
	Can't access protocol node parameters				
	Can't access model				
	Can't access memory type				
	strError.c_str()				
	not allowed icom type				
	invalid natural data type for this memory type				
	invalid field 'selector type'				
	PGN definition not found in the table				
	The field not found in this PGN				

Jetter Ext ETH

The Jetter Ext ETH driver has been developed to communicate with Jetter devices using the PCOM7 protocol.

The HMI protocol identifies Jetter devices using their IP addresses. You should take note of these addresses as you assign them because you will need them later in the set-up phase of the user interface application.

Different physical media, gateways, routers and hubs can be used in the communication network. Also, other devices can independently make simultaneous use of the network. However, it is important to ensure that the traffic generated by these devices does not degrade the communication speed (round-trip time) to an unacceptable level. Too slow communication between the device and the Jetter device may result in low display update rate.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "Jetter Ext ETH" from the list of available protocols.

The driver configuration dialog is shown in the following figure.

PLC Network		ОК
Alias		Cancel
IP Address	0.0.0.0	
Local Port	50001	
PLC Models		
JetControl 3xx p	com /	

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the PLC.
Local Port	Allows to specify the source Port used from the HMI to communicate with PLC.

Element	Description								
PLC Models	An unique PLC model is available: JetControl 3xx pcom7.								
PLC Network	The protocol a multiple conne	llows the connection of multiple controllers to one HMI device. To set-up ections, check "PLC network" checkbox and enter IP Address for all PLCs.							
	Jetter Ext ETH PLC Network Alias IP Address Local Port PLC Models	Alias IP Address PLC Models JetControl 3xx pc	om7	0	. 0	. 0 .	2	OK Cancel	
	Slaves Slave Id 0.0.0.1	Model JetControl 3xx pc	Add Alias	Delete	Modify			<u> </u>	

Tag Editor Settings

Into Tag editor select the protocol "Jetter Ext ETH" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:



letter Ext ETH				
Memory Type Input	Offset ▼ 0	Subindex		
Type boolean	Arraysize	Conversion	+/-	

Element	Description						
Memory Type	Area of PLC where tag is located.						
Offset	Offset address where t	ag is located.					
SubInde x	This allows resource offset selection within the register.						
Туре	Data Type Memory Space Limits						
	boolean	1 bit data	01				
	byte 8-bit data -128 127						
	short 16-bit data -32768 32767						
	int	32-bit data	-2.1e9 2.1e9				
	unsignedByte	8-bit data	0 255				

Element	Description							
	Data Type	Memory Space	Limits					
	unsignedShort	16-bit data	0 65535					
	unsignedInt	32-bit data	0 4.2e9					
	float	IEEE single-precision	1.17e-38 3.40e38					
		32-bit floating point type						
	string	Refer to "String data type chapt	er"					
	Note: to defin like "byte[]", "	ne arrays, select one of Data Type format followed by square brackets "short[]"…						
Arraysiz e	 In case of array t In case of string taging the string tag. Note: number of bytes of to UTF-8 or Latin1 in Taging property is strequires 2 bytes. 	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. te: number of bytes corresponds to number of string characters if Encoding property is set JTF-8 or Latin1 in Tag Editor. Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character urises 2 bytes 						
Conversi	Conversion to be applie	d to the tag.						
on	Conversion							
	inv,swap2	Allowed Co BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits ~	nfigured v bits BCD->CDAB					
	Depending on data type selected, the Allowed list shows one or more conversions, listed below.							

Element	t Description		
	Value	Description	
	Inv bits	Invert all the bits of the tag.	
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)	
	Negate	Set the opposite of the tag value.	
		<i>Example:</i> 25.36 → -25.36	
	AB -> BA	Swap nibbles of a byte.	
		Example: $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)	
	ABCD -> CDAB	Swap bytes of a word.	
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)	
	ABCDEFGH -> GHEFCDAB	Swap bytes of a double word.	
		Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)	
	ABCNOP ->	Swap bytes of a long word.	
	OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.10000000110	
		$ \rightarrow 110000011100 \\ 1010100001010001011011011001010101$	
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)	
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	New Format	Jetter "string" data format	

Element	Description			
	ect the conversion and click on plus button. The selected item will be added on nfigured list.			
	If more conversions are configured, they will be applied in order (from top to bottom of Configured list).			
	Use the arrow buttons to order the configured conversions.			

Special data types

The Jetter Ext ETH driver provides one special data type called "Node Override IP".

The Node override IP allows changing at runtime the IP address of the controller. This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

If the Node Override IP is set to 0.0.0.0, all the communication with the slave is stopped, no request frames are generated anymore.

If the Node Override IP has a value different from 0.0.0.0, it is interpreted as node IP override and the controller IP address is replaced runtime with the new value.

In case the device has been configured to access to a network of controllers, each node has its own Node Override IP variable.



Note: the Node Override IP values assigned at runtime are retained through power cycles

etwork Jetter Ext Er	н			
Memory Type	Offset	Subindex		
Node Override IP	0	0 -		
Туре	Arraysize	Conversion		
unsignedByte []	8		+/-	

String data type

The Jetter devices allow to define within the programming software two different type of string variables: "Regstring" is the old format while "string" is the new format, both these formats are supported by the Jetter Ext ETH driver.

When "Regstring" format is used the corresponding Tag must be configured simply selecting string as data type as shown in the following figure, no further steps are required.

Memory Type	Offset	Subindex		
Register	• 0	0 🔻		
Туре	Arraysize	Conversion		
string	▼ 8		+/-	

When "string" format is used once selected the string data type in the Tag definition dialog it is necessary, as shown in the following figure, to add a New Format conversion.

Jetter Ext ETH			23	
Jetter Ext ETH				
Memory Type Register 🗸	Offset	Subindex		
Туре	Arraysize	Conversion		
string -	8	newForm	Allowed AB->BA	Configured New Format
			ABCD->CDAB New Format	-
				0
				Cancel OK
		OK Cancel	Apply Help	

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
No response	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Incorrect node address in response	The device did receive from the controller a response with invalid node address
The received message too short	The device did receive from the controller a response with invalid format
Incorrect writing data acknowledge	Controller did not accept write request; ensure the data programmed in the project are consistent with the controller resources
Keyence KV

Keyence KV communication driver has been designed to connect HMI devices to KEYENCE PLCs through Serial or Ethernet connection.

Please note that changes in the communication protocol specifications or PLC hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in PLC hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Setting-up the PLC for Communication

Keyence KV PLC's do not require any particular setup-up for communication at the programming port.

Protocol Editor Settings

Add (+) a driver in the Protocol Editor and select the protocol called "Keyence KV" from the list of available protocols.

The driver configuration dialog is shown in figure.

PLC Network	Comm	, ок
Alias		Cancel
Media	Ethernet 👻	
IP address	0.0.0.0	
Port	8501	
Timeout (ms)	600	
Number of retries	2	
PLC Models		
KV10/16		<u> </u>
KV24 KV40/80		=

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
Media	Allows the selection of transport Media.

Element	Description
	 select Serial to connect via serial line select Ethernet to connect via TCP/IP
IP address	IP Address of the controller. Only available for Ethernet Media.
Port	Port number used by PLC. The default value is 8501. Only available for Ethernet Media.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from PLC.
Number of retries	Number of times a communication session is repeated before declaring reporting communication error.



Element	Description						
PLC Models	each data type according to the specific PLC memory resources. Available models: • KV10/16 • KV24 • KV40/80 • KV300 • KV700 • KV700 • KV1000 • KV3000/5000/5500 • KV7300/7500 • KV8000						
Comm	Opens the serial port configuration dialog box. Only available Comm Parameter Dialog Comm Parameter Dialog Image: Common c	ailable for Serial Media.					
	Element	Parameter					
	Port	 Serial port selection. COM1: On-board port COM2: Optional Plug-in module plugged on slot#1 or slot#2 COM3: Optional Plug-in module plugged on slot#3 or slot#4 					
	Baudrate, Parity, Data Bits, Stop bits	Serial line parameters.					

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select Keyence KV from the Driver list: tag definition dialog is displayed.

Keyence KV				×
Keyence KV				
memorytype	Offset	subindex		
Int. Utility Relay	~ 10	• •		
datatype	Arraysize	Conversion		
unsignedShort \sim	0		+/-	
	ОК	Cancel	Apply	Help



Element	Description							
Memory Type	Resource where tag is located	on PLC.						
	Available resources are:							
	 Int. Utility Relay 							
	Data Memory							
	Timer Contact							
	Timer Current							
	Timer Preset							
	Counter Contact							
	Counter Current Counter Preset							
	Counter Preset Digital Trimmer							
	Digital Trimmer Control Memory							
	Temporary Data Memor	v						
	Control Relay	,						
	Link Relay							
	 Int. Aux. Relay 							
	Latch Relay							
	Virtual Relay							
	Ext. Data Memory							
	Curr. File Register							
	Dial File Register							
	Virtual Memory							
	Index Register Ink Register							
Offset	Offset address where tag is loc	ated						
		action within the register						
Subilidex								
datatype	Data Type	Memory Space	Limits					
	boolean	1-bit data	01					
	byte	8-bit data	-128 127					
	short	16-bit data	-32768 32767					
	int	32-bit data	-2.1e9 2.1e9					
	int64	64-bit data	-9.2e18 9.2e18					
	unsignedByte	8-bit data	0 255					

Element	Description						
	Data Type	Memory Space	Limits				
	unsignedShort	16-bit data	0 65535				
	unsignedInt	32-bit data	04.2e9				
	uint64	64-bit data	0 1.8e19				
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38				
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308				
	string	Array of elements containing character code defined by select encoding					
	binary	Arbitrary binary data					
	Note: to define array "byte[]", "short[]"	ays. select one of Data Type format followed by square brackets like					
Arraysize	 In case of array tag, this In case of string tag, this string tag. 	property represents the number of array property represents the maximum numl	elements. ber of bytes available in the				
	Note: number of bytes correspo UTF-8 or Latin1 in Tag Editor. If Encoding property is set to U 2 bytes.	ponds to number of string characters if Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires					
Conversion	Conversion to be applied to the	e tag.					
	Conversion						
	inv,swap2 Allowed BCD AB->B ABCD- ABCDE Inv bit	d Configured IA ->CDAB EFGH->GHEFCDAB Is Cancel OK					
	Depending on data type select	ed, the list Allowed shows one or more of	conversion types.				



Element	Description	
	Value	Description
	Inv bits	inv : Invert all the bits of the tag.
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
	Negate	neg: Set the opposite of tag value.
		<i>Example:</i> 25.36 → -25.36
	AB -> BA	swapnibbles: Swap nibbles in a byte.
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
	ABCD -> CDAB	swap2: Swap bytes in a word.
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH		swap4: Swap bytes in a double word.
	GHEFCDAB	Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP ->	swap8: Swap bytes in a long word.
		Example: 142 366 → -893553517 588905 (in decimal format)
		0 1000000110
		→
		1 10000011100 1010101000010100010110110110
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

Element	Description
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	-	ž	ß	ß	>]	Þ	A 9B	B>	63	1
Data	1		^		-	Ta	g URI			

Locate the .xml file exported from Tag Editor and click Open.

Organize 🔻 🛛 New	folder		833 -	- 🗖 🖉
	* N	ame	Date modified	Туре
Desktop		Exported Tags.xml	03/06/2016 16:42	XML File
Pictures				
Homegroup User Computer Network				
Homegroup User Computer Network Control Panel	+ +	m		

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags × Protocols				
+ - 👗 🕲 🖉 >] [> 😘 🗟 🖄	R 🔎 - Search 🌱 Filter by: Data 🗣	Ite	ms used:6/10000 Protocol: Show all	र 🗹 Show all tags 🔅 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			 Dictionary 	· · · · · · · · · · · · · · · · · · ·
MDT111 unsigned Short			Array	false
- MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
MRTU4 unsignedShort			Comment	
- MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description		
⊡ }	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project		
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.		
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: Image: A probability of the particular of t		
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.		

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Description
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Timeout receiving response characters	Returned when a request is not replied within the specified timeout period between chars in frame, should never be reported; contact technical support
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources

Koyo DL

The Koyo DL driver has been developed for the communication with Koyo DL series controllers trough serial connection.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "Koyo DL" from the list of available protocols.

The driver configuration dialog is shown in the following figure:

Koyo DL	×
PLC Network OK	
Alias	
Node ID 1	
PLC Models	
DL440 Net Port	
DL440 Prg.Port	
DL105, DL230	
DL240 Pra.Port	
DL340 Y	

Element	Description
Node ID	Controller Node ID
PLC Models	The driver supports communication with different DL controllers. Please check directly in the programming IDE software for a complete list of supported controllers.
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and configure all controllers.

Element	Description
	Koyo DL X
	Comm OK
	Alias
	PLC Models
	DL440 Prg.Port DL105, DL230 DL240 Net.Port DL240 Prg.Port DL340 Slaves Add Delete Modify
	Slave Id Koyo DL X
	Alias OK Cancel
	Node ID PLC Models DL440 Prot DL440 Prot, Port DL 105, DL230 DL240 Net.Port
	DL240 Prg.Port DL340
Comm	Gives access to the serial port configuration parameters as shown in the figure below.
	ОК
	Port com1
	Baudrate 9600 -
	Parity even 🔹
	Data bits
	Stop bits
	Mode RS-485
Port	Serial port selection

Element	Description
Baud rate, Parity, Data bits, Stop bits	Communication parameters for serial communication
Mode	Serial port mode; available options: RS-232, RS-485 (2 wires)
	• RS-422 (4 wires)

Tag Editor Settings

Into Tag editor select the protocol "Koyo DL" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Memory Type	Offset	SubIndex		
Data Type	Arraysize	Conversion		
short -	0] [+/-	

Element	Description
Memory Type	Memory resource where tag is located.
Offset	Offset address where tag is located.
SubIndex	This allows resource offset selection within the register.

Element	Description			
Data Type	Data Type	Memory Space	Limits	
	boolean	1 bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
	unsignedByte	8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	04.2e9	
	float	IEEE single-precision	1.17e-38 3.40e38	
		32-bit floating point type		
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308	
	string	Array of elements containing cha selected encoding.	racter code defined by	
	binary	Arbitrary binary data		
	Note: to brackets	define arrays, select one of Data Type format followed by square s like "byte[]", "short[]"…		
Arraysize	In case of aIn case of st	rray tag, this property represents th tring tag, this property represents th	ne number of array elements. The maximum number of bytes	
	available in	the string tag.		
	set to UTF-8 or Lat If Encoding proper character requires	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. f Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.		
Conversion	Conversion to be a	pplied to the tag.		

Element	Description		
	Conversion		
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK	
	Depending on d	ata type selected, the list Allowed shows one or more conversion types.	
	Value	Description	
	Inv bits	inv : Invert all the bits of the tag.	
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)	
	Negate	neg: Set the opposite of tag value.	
		<i>Example:</i> 25.36 → -25.36	
	AB -> BA	swapnibbles : Swap nibbles in a byte.	
		$15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)	
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)	
	ABCDEFGH	swap4: Swap bytes in a double word.	
	-> GHEFCDAB	Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)	
	ABCNOP -	swap8: Swap bytes in a long word.	
	> OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011001000101101	

Element	Description		
	Value	Description	
		0001 \rightarrow 1 10000011100 1010100001010001011011011001011011000010011 1101 (in binary format)	
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	Select conversion	on and click +. The selected item will be added to list Configured .	
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom of list	
	Use the arrow b	uttons to order the configured conversions.	

Koyo DL ETH

The Koyo DL ETH driver has been developed for the connection of Koyo DL series controllers trough Ethernet.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "Koyo DL ETH" from the list of available protocols.

The driver configuration dialog is shown in the following figure:

ίογο DL ETH		×
PLC Network		ОК
Alias		Cancel
Address Type	IP v	
IP Address	0.0.0.0	
Device ID	1	
Device Name		
UDP Port	28784	
🗹 use new EBC Mode		
Timeout (ms)	100	
Retry count	3	
PLC Models		
ECOM		
EBC		

Element	Description
Address Type	Allow to select which address type to use
IP Address	When Address Type is "IP", define the controller IP Address
Device ID	When Address Type is "ID", define the controller Device ID

Element	Description		
Device Name	When Address Type is "Name", define the controller name		
UDP Port	UDP port of controller		
use new EBC Mode	If PLC Model is "EBC" allow to use the new EBC Mode		
Timeout	Defines the time inserted by the protocol between two retries of the		
(ms)	same message in case of missing response from the server device.		
	Value is expressed in milliseconds.		
Retry count	Defines the number of times a certain message will be sent to the controller before reporting the communication error status.		
	A value of 1 for this parameter means the HMI will eventually report the communication error status if the response to the first request packet is not correct.		
PLC Models	The driver supports communication with different DL controllers. Please check directly in the programming IDE software for a complete list of supported controllers.		
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and configure all controllers.		

Tag Editor Settings

Into Tag editor select the protocol "Koyo DL ETH" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Memory Type	Offset	SubIndex		
TA - Timer Acc wor 💌	0	0 👻		
Data Type	Arraysize	Conversion		
boolean 👻	0	1	+/-	

Element	Description			
Memory Type	Memory resource where tag is located.			
Offset	Offset address where tag is located.			
SubIndex	This allows resource offset selection within the register.			
Data Type	Data Type Memory Space Limits		Limits	
	boolean	1 bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
unsignedByte 8-b		8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	04.2e9	
	float	IEEE single-precision	1.17e-38 3.40e38	

Data TypeMemory SpaceLimits32-bit floating point typedoubleIEEE double-precision 64-bit2.2e-308 1.79e308				
32-bit floating point type double IEEE double-precision 64-bit 2.2e-308 1.79e308				
double IEEE double-precision 64-bit 2.2e-308 1.79e308				
floating point type				
string Array of elements containing character code defined by selected encoding.				
binary Arbitrary binary data				
NOTE: to define arrays, select one of Data Type format followed by squar brackets like "byte[]", "short[]"	to define arrays, select one of Data Type format followed by square ts like "byte[]", "short[]"…			
 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 			
Note: number of bytes corresponds to number of string characters if Encoding properset to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.	f bytes corresponds to number of string characters if Encoding property is Latin1 in Tag Editor. perty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one res 2 bytes.			
version Conversion to be applied to the tag.	version to be applied to the tag.			
Conversion				
inv,swap2 Allowed Configured Inv bits AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
Depending on data type selected, the list Allowed shows one or more conversion ty	ypes.			
Value Description				
Inv bits inv: Invert all the bits of the tag.				
Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)				
$9 \rightarrow 6$ (in decimal format)				

Element	Description		
	Value	Description	
		<i>Example:</i> 25.36 → -25.36	
	AB -> BA	swapnibbles: Swap nibbles in a byte.	
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)	
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)	
	ABCDEFGH	swap4: Swap bytes in a double word.	
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)	
	ABCNOP -	swap8: Swap bytes in a long word.	
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ $0001\ 1100\ 1011\ 101\ 100\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 101\ 100\ 10\ 1$	
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	Select conversi	on and click +. The selected item will be added to list Configured .	
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom of list	

Use the arrow buttons to order the configured conversions.

KNX TP/IP

KNX is the association that promotes the KNX communication standard, designed for applications in home and building automation systems.

The KNX standard, approved as European Standard EN 50090, EN 13321-1, is based on the communication stack of EIB with some extensions. EIB is the acronym for European Installation Bus.

Additional information and further details can be found in the KNX web site www.knx.org.

The network communication media supported by the HMI panels are:

- TP-1: twisted pair, type 1, which corresponds to a bus line operating at 9600 bit/s.
- IP: network connection via TCP/IP over Ethernet network.



Note: Connection to KNX systems in TP Mode requires the optional KNX communication module PLCM02. Verify the suitable version of communication module for your HMI model.

The EIB is an event-driven decentralized automation system.

The information to be transmitted over the bus is organized in "telegrams" sent by a source to one or more destination devices.

The bus line of EIB systems carries both data and power for the devices. The data is modulated over the DC voltage of the power supply.

HMI panels are not powered from the network and they still need the usual power supply.

The planning, design and commissioning of KNX installations are normally done using the ETS configuration software. This software tool is supplied by the KNX organization. ETS is a registered trademark of KNX.

This document contains the information required to use ETS in combination with the HMI panels.

All KNX compliant devices come with a device descriptor delivered as a file to be imported in the configuration tool.

The model adopted by HMIs corresponds to a KNX device with no objects. For what concerns the ETS, the only function supported by the HMI panels is the device physical address assignment.

Protocol Editor Settings

Add (+) a driver in the Protocol Editor and select the protocol called "KNX TP/IP" from the list of available protocols.

The protocol parameters can be selected from the dedicated dialog box:

KNX TP/IP			×
PLC Network	Comm		ОК
Alias			Cancel
Media	ТР	~	
IndividualAddressH	15		
IndividualAddressM	15		
IndividualAddressL	255		
Search Interval (s)	300		
Router IP address	0.0.	0.0	
Interface IP address	0.0.	0.0	
PLC Models			
Three-stage addressing			
Two stage data cosing			



The Individual Phisical Address can be assigned on the HMI screen at the first download of the project configured for the KNX protocol. This is the Phisical Address that identifies the panel in the KNX network. The default address value is: 15.15.255.



Note: As any other KNX device, also the HMI device must have unique Individual Address in the KNX network and it must correspond to the real point in the network where the HMI device is connected.

Element	Description	
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.	
Media	 Allows the selection of the transport Media. select TP to connect to the KNX network using the optional KNX communication module PLCM02 select IP to connect to the KNX network via TCP/IP 	
IndividualAddressH	Physical Address High Part (Area)	
IndividualAddressM	Physical Address Medium Part (Line)	
IndividualAddressL	Physical Address Low Part (Device)	
PLC Models	 Allows to choose if KNX telegrams have two or three stage addressing. This selection have to be made basing on KNX device used. Two-stage addressing = KNX telegrams are composed by GoupAddressH / GroupAddressL Three-stage addressing = KNX telegrams are composed by 	

Element	Description
	GoupAddressH / GroupAddressM / GroupAddressL
Search Interval (s)	Available only when Media property is set to IP . The KNX driver will re-evaluate the network with period "Search Interval" (default: 300 seconds). On searching the network, the KNX driver will discover the tunneling endpoints that are available at that time. Endpoints will therefore be registered as possible sources / destinations for group address operations. Depending on endpoints settings or endpoints temporary unavailability the available sources / destinations for group address operations may vary. Thus the capability for the KNX driver to re-evaluate periodically its knowledge about the network.
Router IP address	Available only when Media property is set to IP . This option allows to define the KNX router IP address. If this property is left "0.0.0.0", a multicast request is sent (with timing specified in Search Interval property) via TCP/IP to find a valid KNX TCP interface.



Element	Description			
PLC Network	This option allows to define a network of devices, by specifying Alias, Router IP address and PLC Model for each node.			
	KNX TP/IP	×		
	PLC Network	Comm OK		
	Alias	Cancel		
	Media	TP v		
	IndividualAddressH	15 KNX TP/IP ×		
	IndividualAddressM	15 OK		
	IndividualAddressL	255 Alias Cancel		
	Bouter IP address	300 Router IP address 0 0 1		
	Interface IP address	PLC Models Three-stage addressing Two-stage addressing		
PLC Models				
	Two-stage addressing			
	Add Delete Modify			
	Slave Id Model	Alias		
Comm	If clicked displays t	he communication parameters setup dialog.		
	Comm Parameter Dialog			
	Poling Time 5.	Poling Time 5		
	Insmission Rate	ion Rate 500 V		
	Element	Description		
	Polling Time	Defines how often the tags with Polling attribute		
		enabled are requested to the network (seconds).		
	Transmission	Defines the interval of time between two		
	Rate	consecutive write operations performed by the		

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select KNX TP/IP from the Driver list: tag definition dialog is displayed.

Memory Type	GroupAddressH	GroupAddressM	
BIT	0		
GroupAddressL	Data Type	Arraysize	
0	boolean	•	
Conversion			
	+/-		
Delline			

Element	Description			
Memory Type	KNX resource where tag is located.			
	Memory Type	KNX Data Type	KNX Datapoint Type	
	BIT	Bit	1.0xx	
	1BIT	1 Bit Controlled	2.0xx	
	3BIT	3 Bits Controlled	3.007	
	CS	Character Set	4.00x	
	ou	Octet, Unsigned	5.00x 17.001 18.001	
	OS	Octet, Signed	6.001 6.010	
	20U	2 Octets, Unsigned	7.0xx	
	2OS	2 Octets, Signed	8.0xx	
	20F	2 Octets, Float	9.0xx	
	ТІМ	Time	10.001	
	DAT	Date	11.001	
	STR	String	16.000 16.001	
	4OU	4 Octets, Unsigned	12.001	
	40S	4 Octets, Signed	13.0xx	
	40F	4 Octets, Float	14.0xx	
	ACC	Access	15.000	
	U1	Uncertain (1 byte)	Uncertain	
	U2	Uncertain (2 Bytes)	Uncertain	
	U3	Uncertain (3 Bytes)	Uncertain	
	U4	Uncertain (4 Bytes)	Uncertain	
	Programming Mode	Check "Special Data Types" ch	apter for details	
	Individual Address			

For some KNX Datapoint Types it may be needed to apply the "Scaling" functionality, available from Tag editor. The next figure shows an example of scaling conversion for Percent values of dimmer actuators (Datapoint Type 5.001 DPT_Scaling). Applying this

ŧ,

Element	Description			
	Scaling conversion, the "0/0/1 % Value" tag manage values in range 0÷100 instead of standard range 0÷255 of Unsigned Octet. 1:Page1 protocols* 1:Page1 protocols* * *			
GroupAddres sH	High Group Address of KNX resource. Range: 0 - 31			
GroupAddres sM	Middle Group Address of KNX resource. Range: 0 - 2047 Available only if PLC Model property is set to Three-stage addressing . Check "Protocol Editor Settings" chapter for details.			
GroupAddres sL	Low Group Address of KNX rerource. Range: 0 - 255			
Data Type	Data Type	Memory Space	Limits	
	boolean	1-bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
	unsignedByte	8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt 32-bit data 0 4.2e9		0 4.2e9	
	floatIEEE single-precision 32-bit floating point type1.17e-38 3.4e38			
	string	Array of elements containing character code defined by selected encoding		
	Note: to define arrays. select one of Data Type format followed by square brackets like "byte[]", "short[]"			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 			
	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF- 8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.			

Element	Description				
Conversion	Conversion to be appli	ipplied to the Tag.			
	Conversion				
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK			
	Depending on data type selected, the Allowed list shows one or more conversions, listed b				
	Value	Description			
	Inv bits	Invert all the bits of the tag.			
		Example: 1001 → 0110 (in binary format) 9 → 6 (in decimal format)			
	Negate	Set the opposite of the tag value.			
		<i>Example:</i> 25.36 → -25.36			
	AB -> BA	Swap nibbles of a byte.			
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)			
	ABCD -> CDAB	Swap bytes of a word.			
		<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)			
	ABCDEFGH ->	Swap bytes of a double word.			
	GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)			
	ABCNOP ->	Swap bytes of a long word.			
	OPMDAB	Example: 142.366 \rightarrow -893553517.588905 (in decimal format)			

Element	Description		
	Value	Description	
		0 1000000110 0001110010111011010001011010000111001010	
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)	
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	KNX_DATE	Check "Special Data Types" chapter for details	
	KNX_TIME		
	KNX_DayOfWeek		
	Select the conversion a If more conversions are list). Use the arrow buttons	and click on plus button. The selected item will be added on Configured list. e configured, they will be applied in order (from top to bottom of Configured to order the configured conversions.	
Polling	If checked, this option a of polling requests is de Check "Protocol Editor	allows to force continuous read requests from the HMI to the Tag. The timing efined from "Polling Time" option available in "Comm" window. Settings" chapter for details.	

Special Data Types

Programming Mode

Programming Mode is a special device operating mode that allows changing some system parameters. It is common to most KNX TP devices.

Programming Mode for Individual Address programming via ETS can be set directly in the HMI device.

The first time a HMI project made for the KNX TP communication driver is downloaded to an HMI panel, the unit is assigned the specified Physical Address.

Programming Mode for the HMI panel can be enabled by placing on the screen a widget assigned to the Programming Mode internal variable.

At present there are no database files that can be imported in ETS, so the HMI device can't be programmed using ETS software. The Programming Mode is available only for future functions.

Memory Type	GroupAddressH	GroupAddressM	
Programming Mode 👻	0	0	
GroupAddressL	Data Type	Arraysize	
0	unsignedShort 👻	0	
Conversion			
	+/-		
Delline			

The "Programming Mode" value can be 0 or 1.

Individual Address

The Individual Address can be displayed placing on the HMI screen an object for "Individual Address" data type.

Memory Type	GroupAddressH	GroupAddressM	
Individual Address 👻	0	* *	*
GroupAddressL	Data Type	Arraysize	
0	unsignedShort	•	
Conversion			
	+/-		

The Individual Address can be alternatively assigned directly on HMI screen with a write operation to the internal variable.

Please note that, as any other KNX device, also the HMI device must have unique Individual Address in a KNX network.

In the following figure an example of how the individual address in hex format has to be interpreted.

0xABFF 10/11/255



Note: The max value for Individual address is 15.15.255

Date

The Date data type requires a special data conversion.

(NX TP/IP			\times
KNX TP/IP			
Memory Type	GroupAddressH	GroupAddressM	
GroupAddressL	Data Type ▲ unsignedInt ✓	Arraysize	
Conversion KNX_DATE	Allowed	Configured	
Polling	BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ KNX_DATE	
	< >	Cancel OK	

The correct visualization of the date information from this tag can be achieved using the widget dedicated to handle "DateTime" data source.



Time

The Time data type requires a special data conversion.

KNX TP/IP		>
KNX TP/IP		
Memory Type	GroupAddressH	GroupAddressM
GroupAddressL 0 Conversion	Data Type	Arraysize 0
KNX_TIME	Allowed	Configured
Polling	BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	
	< >>	Cancel OK

The correct visualization of the time information from this tag can be achieved using the widget dedicated to handle "Time" data source.

1:Page1* × protocols Tags Events Bu	ffer 🔻	Pro	perties	
	h 🔂 ₽ 🕒 ▼ 😕 ▼ 🚿	6	ST ST	
			DateTime	
	• 12 • »		Value	-1
	<u>^</u>		DataLink	Tag1:_TagMgr
			Number Forma	hh:mm:ss
00.20.50			Time Spec	local
09:38:50			OnDataUpdate	
		÷	Text	
	=			

Note: In the "DateTime" widget it is important to set properly the "Time Spec" property in order to avoid the influence on the visualization of the HMI clock timezone and DST settings; Select Number format properly.

Note: Write operation from HMI to KNX network will be executed only with "No Day" information.

Day of Week

1

Π

The Day of Week data type is part of Time telegram and requires a special data conversion.

NX TP/IP		
(NX TP/IP		
Memory Type	GroupAddressH	GroupAddressM
GroupAddressL	Data Type	Arraysize
KNX_DOW	Allowed	Configured KNX DavOfWeek
Polling	AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+
	< >	Cancel OK



Note: This object is in read-only mode

Dimming function

3 Bits Controlled data type has to be used to operate a dimming function.

This is a 4 bit data where the 1st bit is used to determine if increment or decrement the value and the remaining 3 bits determines the percentage of dimming applied.

The Tag will represent a fixed percentage value (from 0% to 100%) of increasing or decreasing of a particular device value.

NX TP/IP			:
Memory Type 3BIT ~	GroupAddressH	GroupAddressM	
GroupAddressL	Data Type	Arraysize	
Conversion	+/-		
Polling			
	ОК	Cancel Apply	Help

The table below reports the action performed for each value assumed by the Tag.

For example, to increase the dimmed value of 25% it is necessary to write into the Tag that manages the dimming the binary value "1011", which in decimal code, corresponds to "11".

Direction	Data	Action
0	001	Down 100%
0	010	Down 50%
0	011	Down 25%
0	100	Down 12%
0	101	Down 6%
0	110	Down 3%
0	111	Down 1%
1	001	Up 100%
1	010	Up 50%
1	011	Up 25%

Direction	Data	Action
1	100	Up 12%
1	101	Up 6%
1	110	Up 3%
1	111	Up 1%

As mentioned before to increase the dimmed value by 25% it is necessary to write 11 in the corresponding Tag. To do this a Write Tag action programmed as shown in the next figure must be created.

Multi Language SetLanguage SetLanguage Tag Data Transfer ToggleBit SetBit SetBit ResetBit WriteTag StepTag ActivateGroup DeactivateGroup DeactivateGroup BACnetClearAllPriority BACnetClearAllPriority BACnetSetPriority BACnetSetPriority DeactivateGroup	Action Properties WriteTag TagName Project:_TagMgr;Tag1;Tag + TagValue 11 TagName Source tag
--	---

Tag Import

Exporting Tags from PLC

The KNX TP/IP driver supports the Tag import facility. The import filter accepts symbol files with extension ".esf" created by the ETS programming tools.

The ETS configuration software can export the database information related to group addresses.

To export database information select "Extract data" from the File menu of ETS software.


Select the option "Export to OPC Server" to export data in ".esf" format.

Clicking on "Export..." creates the ".esf" file to be imported in the Tag Editor.

Export Foreign Format	×
Export to CSV/XML Export the content of the active list view (right browser pane) to a CSV or XML file (e.g. for further use in a spreadsheet program).	Export
Export to OPC Server Exports project data for use by the KNX OPC server.	Export
	Cancel

Exporting Tags from PLC using ETS5

From ETS5 programming software click on *Extras > Export OPC*

ETS Edit Workplace Commission	oning Diagnostic	s Extras Window			~ 0
👩 Close Project 🛛 🎸 Undo 🛝	Redo 😝 Repo	ort Export OPC	alogs 🔤 Diagnos	tics	
New project - Group A × New pro	oject - Buildings	Replace Product			Properties >
Group Addresses 🔻		3	6	∧ □ ×	
🕂 Add Middle Groups 🛛 🔹 🗙 Delete	🛨 Download 🔹	🕕 Info 🔹 <u> </u> Reset	Search	Q	Settings Comments Information
Group Addresses 🔹	Main Group	Name	Description	Pass Thr	Name
Dynamic Folders	8日 0	Heating		No	First Floor
▲ 器 0 First Floor					Address
🖁 0/0 Heating					0 -
				Description	
	¢ Dataik			>	 Pass through Line Coupler Security Automatic Find and Replace Workspaces Todo Items Pending Operations Undo History
<no +="" 11="" interface="" li<="" new="" selecte="" td=""><td>ne</td><td>0 First Floor</td><td></td><td></td><td>Last used workspace</td></no>	ne	0 First Floor			Last used workspace

In next step select location and file name for .esf file.

EX05

🚖 Favorites 📰 Desktop		ocuments library	A	rrange by: Folder 🔻
\rm Downloads 🗐 Recent Places	Nai	me 🖍	Date modified 24/10/2017 10:58	Type File folder
🔒 Libraries	5	My Music	04/04/2011 14:27	File folder
Documents	4	My Music	14/07/2009 06:53	File folder
Music		My Pictures	14/07/2009 06:53	File folder
Videos	2	My Videos My Videos	04/04/2011 14:27	File folder
🍓 Homegroup			III	
File name:	New project	.esf		
Save as type: E	IB session f	iles (.esf) (*.esf)		
Hide Folders			Save	Cancel

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	z	ß	ß	>]	₽	A 9B	B>	63	1
Data	1		~		-	Ta	g URI			

The following dialog shows which importer type can be selected.

lersion	Туре	
TS v 1.0	Linear	
ag Editor expor	ted xml General	

Importer	Description						
ETS v1.0	Requires a .esf file.						
Linear	All variables will be displayed at the same level.						
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.						
	Tags x						
	+ — 🎽 🕲 刘 [< 🕲 🔊 🖌 – +						
	Data						

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖉 🔰	SB 🖬 👬	R 🔎 - Search YFilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 🔅 🗖
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	- Int -			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
Holding Registers 3	unsignedShort			Array	falco
- MRTU1	unsignedShort			Alldy	Idisc
MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort		_	Data type	unsignedShort



Toolbar item	Description
B	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tage X
P - Search	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Response error	The tag requested by the panel may be not available in the system or communication session completed with errors
General Error	Error cannot be identified; should never be reported; contact technical support
Internal software error	Unrecognized error

The status codes supported for this communication driver are:

Lenze CANopen

The Lenze CANopen communication driver has been designed to connect HMI products to Lenze controllers using the CANopen network. A new device communication profile has been developed to takes advantage from the advanced user interface features of the software, while retaining the simple networking concept supported by the CANopen network.

Connection to CANopen networks requires the optional CANopen communication module.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Please ensure that the latest driver is used in the application.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Lenze CANopen" from the list of available protocols.



The protocol configuration dialog is shown in figure.

Lenze CANopen			×
PLC Network			
Alias		SendSync	
CAN port	0		
Node ID	1		
Baud Rate (Kbps)	500 ~		
Scan Time (ms)	10		
Timeout (ms)	500		
PLC Models			
default			
		OK Cancel	

Element	Description					
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node					
Node ID	CANopen Node ID assigned to the slave device					
Baud Rate (kbps)	Speed of the CANopen network					
Scan Time (ms)	Scan time is dependent upon your specific process or application requirements and the capabilities of your controller.					
Timeout (ms)	Maximum allowed time the driver will wait for a response from the device before reporting a communication error					
SendSync	The Sync-Producer provides the synchronization-signal for the Sync-Consumer. When the Sync-Consumers receive the signal they start carrying out their synchronous tasks.					
PLC Models	This version supports only one device model.					
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter the node ID per each slave you need to access.					

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 🕺 🖻 🖻 >] [> 🗞 🖻	R P- Sean	🍸 Filter by: Data 🗸 🗸	Items used:8/10000 Protocol: Show all
ata Lenze CANopen:prot1 Slave id: Alias: Node1	Type Tag nam Container Container	Tag URI	Prop
Model:	altanti Hanika D	(Channel 1993) and	
- Presses	Real House of	an La stringenaum (18 La strikknas)	
 Ref., Proceedings (Second Proceeding) Ref., Proceedings (Second Procedure) Ref., Proceedings (Second Procedure) Ref., Proceedings (Second Procedure) Ref., Proceedings (Second Procedure) Ref., Proced	aragentilet fasik fi aragentilet fasik fi aragentileget fasik fi aragentileget fasik fi aragentilet fasik fi	Import Tag(s) - Network	${\times}$ clate this dictionary and continue.
Water_Level	unsignedShort NodeI/W	Level Slave id Model Alias	
Slave id: Alias: Node2 Model:	Container	Node1	
		Node2	
		For each selected node import only symbols i	matching that node



Note: Aliasing tag names is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

The Lenze CANopen driver supports the Tag import facility. The import filter accepts symbol files with extension ".eds" provided by Lenze, the device manufacturer.

In Tag Editor select the communication driver and click on the "Import tag" button to start the importer.

ProjectView 📮 🗙	1:Page1 Protocols* Tags x
+ - 4 ^ ~	+ - 🐰 🖄 😥 🔰 [> 🖧 🖻 👸 R 🔎 Search 🛛 🍸 Filter by:
V Project1	Data Type Tag name
Project properties	Lenze CANopen:prot1
Pages	Model: default
V Normal	HMIStudio X
Dialog	
Templates	Multiple tag importers are available for this protocol. Please select the importer type and continue.
✓ Configuration	
Protocols	Version Type
✓ E Tags	Lenze EDS v1.0 Linear
Indexed Tag Set	
Trends	Tag Editor exported xml 1.1 General
Reports	
🔔 Alarms	
📌 Events Buffer	
Scheduler	
두 MultiLanguage	
Screen Saver	
😽 Database Links	
Data transfers	
> CP Interfaces	Watched dictionary file:
> 👸 Security	
Recipe	
Dicuonaries	
reypaus	OK Cancel

Once the importer has been selected, locate the symbol file and click Open.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the "add tags" button as shown in figure.

	1:Page1 Protocol	s Tags 🗙				
+	- 200	>] [> (ы <u>Ка</u> ка	R P- Search	YFilter by: Data	✓ Items
Data		Туре	Tag name	Tag URI	Dictionary	
l ₄ ¦	enze CANopen:prot1 Iodel: default	Container				
	str	string-16	str	0??4?0?string-16		
	ARRAY_WORD[1]	short-8	ARRAY_WORD[1]	0?ARRAY_WORD[1]?4?1?short-8		
	ARRAY_WORD[2]	short-8	ARRAY_WORD[2]	0?ARRAY_WORD[1]?4?2?short-8		
	- ARRAY_WORD[3]	short-8	ARRAY_WORD[3]	0?ARRAY_WORD[1]?4?3?short-8		
	- ARRAY_WORD[4]	short-8	ARRAY_WORD[4]	0?ARRAY_WORD[1]?4?4?short-8		
	-MDW2	unsignedShort	MDW2	0?ARRAY_WORD[1]?4?5?unsigned	Short	
	MDW3	unsignedShort	MDW3	0?ARRAY_WORD[1]?4?6?unsigned	Short	

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
No response	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid access to var	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Wrong answer frame from server	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
Var is too long	The panel did receive from the controller a response, but its length exceeded the max length admitted ensure the data programmed in the project are consistent with the controller resources.

Modbus RTU

The operator panels can be connected to a Modbus network as the network master using this communication driver.

Implementation details

The Modbus RTU implementation supports only a subset of the Modbus standard RTU function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register	Presets a value in a Register
16	Preset Multiple Registers	Presets value in multiple Registers



Note: Communication speed with controllers is supported up to 115200 baud.



Note: Floating point data format is IEEE standard compliant.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the PLC list.

The protocol configuration dialog is displayed.

Cancel

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
Node ID	Modbus node of the slave device.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
Delay (ms)	Time delay in milliseconds between the end of the last received frame and the starting of a new request. If set to 0, the new request will be issued as soon as the internal system is able to reschedule it.
Num of repeats	Number of times a certain message will be sent to the controller before reporting the communication error status.
	When set to 1 the panel will report the communication error if the response to the first request packet is not correct.

Element	Description			
Max read block	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.			
Max read bit block	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.			
Write Holding Register	Modbus function for write operations to Holding Registers. Select between the function 06 (preset single register) and function 16 (preset multiple registers).			
	If function 06 is selected, the protocol will always use function 06 for writing to the controller, even when writing to multiple consecutive registers.			
	If function 16 is selected, the protocol will always use function 16 to write to the controller, even for a single register write request and the Max read block size parameter of the query is set to 2 . The use of function 16 may result in higher communication performance.			
Write Coils	Modbus function for write operations to Output Coils. Select between the function 05 (write single coil) and function 15 (write multiple coils).			
	If Modbus function 05 is selected, the protocol will always use function 05 for writing to the controller, even when writing to multiple consecutive coils.			
	If Modbus function 15 is selected, the protocol will always use function 15 to write to the controller, even for a single coil write request. The use of function 15 may result in higher communication performance.			
Transmission	RTU: use RTU mode			
Mode	ASCII: use ASCII mode			
	Note: When PLC network is active, all nodes will be configured with the same Transmission Mode.			
PLC Models	Allows to select between different PLC models:			
	 Modicon Modbus (1-based): Modbus implementation where all resources starts with offset 1. 			
	 Generic Modbus (0-based): Modbus implementation where all resources starts with offset 0. 			
	 Enron Modbus (1-based): Extends Modicon Mobdus implementation with 32 bit registers memory area. 			
	 Enron Modbus (0-base): Extends Generic Modbus implementation with 32 bit registers memory area. 			
	Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.			
Comm	If clicked displays the communication parameters setup dialog.			

Element	Description			
	Comm Parameter Dialog	(100 m)	x	
	Port Baudrate Parity Data Bits Stop bits Mode	com1 19200 Even 8 1 RS-485	ОК • • • •	
	Element			Parameter
	Port			Serial port selection.
				 COM1: On-board port COM2: Optional Plug- in module plugged on slot#1 or slot#2 COM3: Optional Plug- in module plugged on
				slot#3 or slot#4
	Baudrate, Parity, I	Data Bits, Stop I	oits	Serial line parameters.
	Mode			Serial port mode. Available modes:
				 RS-232. RS-485 (2 wires). RS-422 (4 wires).
PLC Network	Multiple controllers c connections, select f	an be connected PLC networkand	to one HMI dev click Add to co	ice. To set-up multiple nfigure each slave

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Modbus RTU** from the protocol list: tag definition dialog is displayed.

Memory Type	Offset	SubIndex		
		Conversion		
boolean			+/-	

Element	Description				
Memory	Modbus resource where tag is located.				
турс	Memory Type	Description			
	Coil Status	Coils			
	Input Status	Discrete Input	Discrete Input		
	Input Registers	Input Registers	Input Registers		
	Holding Registers	Holding Registers	Holding Registers		
	32 bit Registers	32 bit registers mem	ory area.		
		Available only for Enron Modbus PLC Models			
	Node Override ID				
	Modicon Mode	_			
	Serial Baudrate		(see Snocial Data Typos	for mode	
	Serial Parity	details)	details)		
	Serial Stop Bits	_			
	Serial Mode				
	Serial Done				
Offset	Offset address where tag is lo	ocated.			
	Offset addresses are six digits composed by one digit data type prefix + five digits resource address.				
	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range	
	Coil Status	0 – 65535			
	Input Status	100000 – 165535			
	Input Registers	300000 - 365535	1 – 65536	0 – 65535	
	Holding Registers	400000 - 465535			
	32 bit Registers	0 – 65535			
SubIndex	This allows resource offset selection within the register.				

Element	Description				
Data Type	Data Type	Memory Space	Limits		
	boolean	1-bit data	01		
	byte	8-bit data	-128 127		
	short	16-bit data	-32768 32767		
	int	32-bit data	-2.1e9 2.1e9		
	int64	64-bit data	-9.2e18 9.2e18		
	unsignedByte	8-bit data	0 255		
	unsignedShort	16-bit data	0 65535		
	unsignedInt	32-bit data	04.2e9		
	uint64	64-bit data	01.8e19		
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38		
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308		
	string	Array of elements containing character code defined by selected encoding			
	binary Arbitrary binary data				
	Note: to define arrays. select one of Data Type format followed by square brackets like "byte[]", "short[]"				
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 				
	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.				
Conversion	Conversion to be applied to the	e tag.			

Conversion inv,swap2 Allowed BCD ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB Inv bits inv: Invert all the bits of the tag. Example: 1001 -> 0110 (in binary format) 9 -> 6 (in decimal format) SS8 -> 20813 (in decimal format) SS82 -> 52378 (in decimal format) S928 -> 52378 (in decimal format) S9242 -> 523553517 588905 (in decimal format) S9441236 -> 1425062286 (in decimal format)	Element	Description			
Inv.swap2 Alowed Inv bits BCD BCD Inv bits ABCD->CDAB ABCD->CDAB ABCD->CDAB Inv bits Inv bits Inv: Invert all the bits of the tag. Example: 1001 → 0110 (in binary format) 9 → 6 (in decimal format) Negate neg: Set the opposite of tag value. Example: 25.36 → -25.36 AB -> BA swapnibbles: Swap nibbles in a byte. Example: 1504 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format) 5588 → 20813 (in decimal format) ABCD -> Swap2: Swap bytes in a word. Example: 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format) 39628 → 52378 (in decimal format) 39628 → 52378 (in decimal format) 39628 → 52378 (in decimal format) 39628 → 52378 (in decimal format) 3955411236 → 1426062386 (in decimal format) 327CFF54 → 54FFFC32 (in hexadecimal format) 35541236 → 1426062386 (in decimal format) 35541236 → 1426062386 (in decimal format) 35541236 → 1426062386 (in decimal format) 3566 → -93553517.588905 (in decimal format) 01000000110 00011100101110110101010101010101010101		Conversion			
Depending on data type selected, the list Allowed shows one or more conversion types.ValueDescriptionInv bitsinv: Invert all the bits of the tag. Example: 1001 \rightarrow 0110 (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. Example: 25.36 \rightarrow -25.36AB -> BAswapnibbles: Swap nibbles in a byte. Example: 1504 \rightarrow 514D (in hexadecimal format)ABCD -> CDABswap2: Swap bytes in a word. Example: 39628 \rightarrow 52378 (in decimal format)ABCD => GHEFCDABswap4: Swap bytes in a double word. Example: 327C FF54 \rightarrow 54FFFC32 (in hexadecimal format) 35541236 \rightarrow 1426062386 (in decimal format)ABCNOP - >OPMDABswap8: Swap bytes in a long word. Example: 142.366 \rightarrow -893553517.588905 (in decimal format) 00011100101110110100011100100111001011010		inv,swap2	Allowed Configured BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK		
ValueDescriptionInv bitsinv: Invert all the bits of the tag. $Example:$ $1001 \rightarrow 0110 (in binary format)9 \rightarrow 6 (in decimal format)Negateneg: Set the opposite of tag value.Example:25.36 \rightarrow -25.36AB -> BAswapnibbles: Swap nibbles in a byte.Example:15D4 \rightarrow 514D (in hexadecimal format)ABCD ->CDABswap2: Swap bytes in a word.Example:9ACC \rightarrow CC9A (in hexadecimal format)ABCD ->CDABswap4: Swap bytes in a double word.Example:39628 \rightarrow 52378 (in decimal format)ABCDEFGHSF544 \rightarrow 54FFFC32 (in hexadecimal format)ABCDEFGHSF5441236 \rightarrow 1426062386 (in decimal format)ABCNOP -PMDABswap3: Swap bytes in a long word.Example:142.366 \rightarrow -893553517.588905 (in decimal format)0100000011000011100101110110101010101010101010101$		Depending on d	ata type selected, the list Allowed shows one or more conversion types.		
Inv bitsinv: Invert all the bits of the tag. $Example:$ 1001 \rightarrow 0110 (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. $Example:$ 25.36 \rightarrow -25.36AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ 15D4 \rightarrow 514D (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format) $5588 \rightarrow 20813$ (in decimal format) $5588 \rightarrow 20813$ (in decimal format) $ABCD ->$ CDAB $example:$ $9ACC \rightarrow CC9A (in hexadecimal format)39628 \rightarrow 52378 (in decimal format)39628 \rightarrow 52378 (in decimal format)ABCDEFGHSEFFECDABexample:32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format)855441236 \rightarrow 1426062386 (in decimal format)855441236 \rightarrow 1426062386 (in decimal format)ABCNOP -PMDABexample:142.366 \rightarrow -893553517.588905 (in decimal format)0 1000000110000111001011101101000101110100011100100$		Value Description			
Example: 1001 \rightarrow 0110 (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. Example: 25.36 \rightarrow -25.36AB \rightarrow BAswapnibbles: Swap nibbles in a byte. Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD \rightarrow CDABswap2: Swap bytes in a word. Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)ABCDEFGH \rightarrow GHEFCDABswap4: Swap bytes in a double word. Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)ABCNOP - \rightarrow OPMDABswap8: Swap bytes in a long word. Example: 142.366 \rightarrow -893553517.588905 (in decimal format) 0 1000000111001011101101000111000011100101		Inv bits	inv : Invert all the bits of the tag.		
Negateneg: Set the opposite of tag value. $Example:$ $25.36 \rightarrow -25.36$ AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D (in hexadecimal format)5588 \rightarrow 20813 (in decimal format)ABCD ->CDABswap2: Swap bytes in a word.Example:9ACC \rightarrow CC9A (in hexadecimal format)ABCDEFGH39628 \rightarrow 52378 (in decimal format)ABCDEFGHSS441236 \rightarrow 1426062386 (in decimal format)ABCNOPS55441236 \rightarrow 1426062386 (in decimal format)ABCNOPSOPMDABswap8: Swap bytes in a long word.Example:142.366 \rightarrow -893553517.588905 (in decimal format)010000001110 00011100101101010101010101$			Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
Example: 25.36 \rightarrow -25.36AB -> BAswapnibbles: Swap nibbles in a byte. Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. Example: 		Negate	neg: Set the opposite of tag value.		
AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. $Example:$ 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)ABCDEFGH -> GHEFCDABswap4: Swap bytes in a double word. $Example:$ 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)ABCNOP - > OPMDABswap8: Swap bytes in a long word.Example: 142.366 \rightarrow -893553517.588905 (in decimal format) 01000000110 0001110010111011010001110000111001010101			<i>Example:</i> 25.36 → -25.36		
Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)ABCD -> CDABswap2: Swap bytes in a word. 		AB -> BA	swapnibbles: Swap nibbles in a byte.		
ABCD -> CDABswap2: Swap bytes in a word. Example: $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)ABCDEFGH -> GHEFCDABswap4: Swap bytes in a double word. $Example:$ $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)ABCNOP - > OPMDABswap8: Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0 1000000110$ $0011100101110110010010110000111001010101$			<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
CDABExample: $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)ABCDEFGH -> GHEFCDABswap4: Swap bytes in a double word. $Example:$ $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) 		ABCD ->	swap2: Swap bytes in a word.		
ABCDEFGH -> GHEFCDABswap4: Swap bytes in a double word. $\stackrel{P}{BHEFCDAB}$ $Example:$ $32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format)855441236 \rightarrow 1426062386 (in decimal format)ABCNOP ->OPMDABswap8: Swap bytes in a long word.\stackrel{P}{I42.366} \rightarrow -893553517.588905 (in decimal format)0 10000001100001110010111011000010110000111001010101$		CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
GHEFCDABExample: $32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format)855441236 \rightarrow 1426062386 (in decimal format)ABCNOP -OPMDABswap8: Swap bytes in a long word.Example:142.366 \rightarrow -893553517.588905 (in decimal format)0 1000000110 \\ 0001110010111011010001011101000011100101$		ABCDEFGH	swap4: Swap bytes in a double word.		
ABCNOP - swap8: Swap bytes in a long word. > Example: OPMDAB 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010101010001011010000111001010101		GHEFCDAB <i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)			
OPMDAB Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011001000101101		ABCNOP -	swap8: Swap bytes in a long word.		
		OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011001000101101		

Element	Description				
	Value	Description			
		001 → 1 10000011100 1010101000010100010110110110010110110000			
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)			
	Select conversion and click +. The selected item will be added to list Configured . If more conversions are configured, they will be applied in order (from top to bottom of list Configured)				
	Use the arrow b	uttons to order the configured conversions.			

Node Override ID

The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.



Note: Node Override ID value assigned at runtime is retained through power cycles.

Memory Type	Offset	SubIndex	
Node Override ID 🔻	0	0 *	
Data Type	Arraysize	Conversion	
unsignedByte 👻	0		+/

Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

Memory Type	Offset	SubIndex		
Data Type	Arraysize	Conversion		
boolean	•		+/-	

Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

Parameter	Description	on				
Serial Baudrate	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.					
Serial Parity	unsigned	unsigned 8 bit value for parity overriding. Possible values are described in the following list.				
	Value Description					
	0 none parity 1 even parity					
	2	odd parity				
Serial Stop Bits	unsigned	3 bit value for stop bits overriding. Possible values are 1, 2.				
Serial Mode	unsigned	3 bit value for serial mode overriding. Possible values are described in the following list.				



Parameter	Description	Description				
	Value	Description				
	0	RS-232 mode				
	1	RS-485 mode				
	2	RS-422 mode				
Serial Done	Set to 1 to together o	Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1				

Modbus RTU				×
Modbus RTU				
Memory Type Serial Baudrate ✓ Input Status ✓ Coil Status ✓ Holding Registers Input Registers Input Registers Node Override ID Modicon Mode Serial Baudrate Serial Parity Serial Stop Bits Serial Mode Serial Done	Offset 0 Arraysize 0	SubIndex 0 ~ Conversion	+/-	
	c	K Cancel	Apply	Help

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	Z	đ	ß	>]	₽	A 9B	B>	63	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers are	available for this protocol. Please select the importer type and continue.
Version	Туре
Modbus Generic csv v1.0	Linear
DELTA PLC csv v1.0	Linear
Schneider SoMachine Basic csv v	.0 Linear
Schneider Unity v1.0	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	OK Cancel

Туре	Description				
Modbus Generic csv v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.				
DELTA PLC csv v1.0	Requires a .csv file. All variables will be displayed at the same level.				
Schneider Unity v1.0 Linear	Requires a .uny file. The file containing symbols must be exported in .txt format and later renamed as .uny . The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 — 🎽 🕲 🖉 🐛 — 🕂	R 🔎 - Search YFilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗖
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			✓ Dictionary	
Holding Registers 3 UnsignedShort			Array	false
MRTI01 unsignedShort			Array size	0
MRTU3 unsignedShort			Arravindex Subindex	400003
- MRTU4 unsignedShort			Comment	
- MRTU5 unsignedShort			Data type	upsignedShort
I I I summer a state of the second			oute type	unagricuonore

Toolbar item	Description			
B	Import Tag(s).			
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project			
₿ ₿	Update Tag(s).			
	Click on this icon to update the tags in the project, due a new dictionary import.			
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:			
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.			

Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
NodelD	Node the tag belongs to
TagName	Tag description
MemoryType	 OUTP INP IREG HREG
Address	Offset compatible with Modbus notation
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

Tag file example

Example of .csv line:

2, Holding Register 1, HREG, 400001, unsignedShort,



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
No response	No reply within the specified timeout.	Check if the controller is connected and properly configured to get network access.
Incorrect node address in response	The device received a response with an invalid node address from the controller .	-
The received message too short	The device received a response with an invalid format from the controller .	-
Incorrect writing data acknowledge	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

Modbus RTU Server

Modbus RTU Server communication driver allows connecting the HMI device as a slave in a Modbus RTU network. Standard Modbus messages are used for information exchange.

This approach allows connecting HMI devices to SCADA systems through the universally supported Modbus RTU communication protocol.

Principle of operation

This communication driver implements a Modbus RTU slave unit in the HMI device. A subset of the complete range of Modbus function codes is supported. The available function codes allow data transfer between the master and the slave.

The following diagram shows the system architecture.



The HMI device is actually simulating the communication interface of a PLC: Coils and Registers are respectively boolean and 16 bit integers.

The device always access data in its internal memory. Data can be transferred to and from the Modbus Master only on initiative of the Master itself.

Implementation details

This Modbus RTU slave implementation supports only a subset of the standard Modbus function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area.
03	Read Holding Registers	Read multiple device Registers.

Code	Function	Description
05	Force Single Coil	Forces a single device Coil to either ON or OFF.
06	Preset Single Register	Presets a value in a device Register.
08	Loopback Diagnostic Test	Only sub function 00 (Return Query Data) is supported.
15	Force Multiple Coils	Forces multiple device Coils to either ON or OFF.
16	Preset Multiple Registers	Presets value in multiple device Registers.
17	Report Slave ID	Returns diagnostic information of the controller present at the slave address.
23	Read Write Multiple Registers	Read & presets values in multiple device Registers

Exception Codes

Code	Description
01	Illegal Function. the function code received in the query is not supported
02	Illegal Data Address . Data Address received in the query exceeds the predefined data range (see Tag Definition for detailed ranges of all types).
03	Illegal Data Value . A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the $\ensuremath{\text{PLC}}$ list.

The protocol configuration dialog is displayed.

Modbus ID	1 Cancel
🔄 Enron 32bit regi	sters
32bit reg Start	0
32bit reg Size	0
PLC Models	
Modicon Modbus(1 Generic Modbus(0	-based) -based)

Eleme nt	Description				
Modbu s ID	Modbus node ID. Every Modbus server device in the network must have its own Modbus ID.				
Enron 32bit registe rs	If selected, allows to define the first register address and the number of registers for 32 bit registers memory area. Note: 32 bit registers are available only for Enron Modbus PLC Models.				
	32 bit registries memory area definition.				
32bit reg Start 32bit reg Size	Start value represents the first register address. Size value represents the number of registries. Image: Note: A request to one of the registries inside this area gives a 4 byte answer.				
PLC Models	 Allows to select between different PLC models: Modicon Modbus (1-based): Modbus implementation where all resources starts with offset 1. Generic Modbus (0-based): Modbus implementation where all resources starts with offset 0. Enron Modbus (1-based): Extends Modicon Mobdus implementation with 32 bit registers memory area. Enron Modbus (0-base): Extends Generic Modbus implementation with 32 bit registers memory area. 				

Eleme nt	Description				
	Note: The addr for the Holding	ress rai Regist	nge used in the Modbus frames is always between 0 and 65535 ters and between 0 and 65535 for Coils.		
Com m	If clicked, displays the communication parameters setup dialog. You have to set parameters according to the values programmed in Modbus Master.				
	Comm Parameter Dialog				
		2	OK		
	Baudrate 9600)			
	Parity	1			
	Data bits 8 Stop bits 1				
	Mode RS-4	185	▼		
	Element		Description		
	Uart		Serial port selection.		
			COM1: On-board port		
			 COM2: Optional Plug-in module plugged on slot#1 or slot#2 		
	COM3: Optional Plug-in module plugged on slot#3 or slot#4				
	Baudrate, Parity, Data Stop bits	bits,	Serial line parameters.		
	Mode		Serial port mode. Available options:		
			• RS-232		
			• RS-485 (2 wires)		
			• RS-422 (4 wires)		

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Modbus RTU Server** from the protocol list: tag definition dialog is displayed.

Modbus RTU Server				
Memory Type	Offset	SubIndex		
Coil status 🔹	1	0 *		
Data Type	Arraysize	Conversion		
boolean 👻	0		+/-	

Element	Description					
Memory	Modbus resource where tag is located.					
туре	Memory Type	Modbus Resource	Modbus Resource			
	Coil Status	Coils				
	Input Status	Discrete Input				
	Input Registers	Input Registers				
	Holding Registers	Holding Registers				
	32 bit Registers	32 bit registers mem	nory area.			
		Available only for Enron Modbus PLC Models				
	Node Override ID					
	Modicon Mode					
	Serial Baudrate		/ - · · - · -			
	Serial Parity	_ protocol parameter (details)	protocol parameter (see Special Data Types for mode details)			
	Serial Stop Bits					
	Serial Mode					
	Serial Done					
Offset	Offset address where tag is lo	ocated.				
	Offset addresses are six digit address.	s composed by one dig	it data type prefix + five di	gits resource		
	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range		
	Coil Status	0 – 65535				
	Input Status	100000 – 165535				
	Input Registers	300000 - 365535	1 – 65536	0 – 65535		
	Holding Registers	400000 – 465535				
	32 bit Registers	0 – 65535				
SubIndex	This allows resource offset selection within the register.					

Element	Description				
Data type	Data Type		Memory Space		Limits
	boolean		1-bit data	1-bit data	
	byte		8-bit data		-128 127
	short		16-bit data		-32768 32767
	int		32-bit data		-2.1e9 2.1e9
	int64		64-bit data		-9.2e18 9.2e18
	unsignedByte		8-bit data		0 255
	unsignedShort		16-bit data		0 65535
	unsignedInt		32-bit data		0 4.2e9
	uint64		64-bit data		0 1.8e19
	float		IEEE single-precision 32-bit floating point type		1.17e-38 3.4e38
	double		IEEE double-precision 64-bit floating point type		2.2e-308 1.79e308
	string		Array of elements containing character code defined by selected encoding		
	binary		Arbitrary binary data		
	Note: to define brackets like "I	e array byte[]"	ys. select one of Data Type format followed by square ", "short[]"…		
Arraysize	When configuring array or string tags, this option define the amount of array elements or characters of the string.				
Conversion	Conversion to be applied to the tag.				
	Conversion	nversion			
	inv,swap2	Allowe	d	Configured	
		AB->B ABCD- ABCDE Inv bit	A ->CDAB EFGH->GHEFCDAB IS	ABCD->CDAB	

Element	Description				
	Depending on data type selected, the list Allowed shows one or more conversion types.				
	Value	Description			
	Inv bits	inv : Invert all the bits of the tag.			
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)			
	Negate	neg: Set the opposite of tag value.			
	<i>Example:</i> 25.36 → -25.36				
AB -> BAswapnibbles: Swap nibbles in a byte. $Example:$ $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)		swapnibbles: Swap nibbles in a byte.			
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)			
	ABCD ->	swap2: Swap bytes in a word.			
	CDAB	Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)			
	ABCDEFGH	swap4: Swap bytes in a double word.			
	-> GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)			
	l				

Element	Description			
	Value	Description		
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010		
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)		
Select conversion and click +. T If more conversions are configu Configured). Use the arrow buttons to order t		on and click +. The selected item will be added to list Configured . Ions are configured, they will be applied in order (from top to bottom of list		
		uttons to order the configured conversions.		

Node Override ID

The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the slave is stopped. In case of write operation, the device will not respond to request frames.
1 to 255	It is interpreted as the value of the new node ID and is replaced for runtime operation.



Note: Node Override ID value assigned at runtime is retained through power cycles.

1odbus RTU Server			
Memory Type	Offset	SubIndex	
Node Override ID 👻	0	0 -	
Data Type	Arraysize	Conversion	
unsignedByte 🗸	0	+/-	

Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

10dbus RTU Server				
Memory Type	Offset	SubIndex		
Modicon Mode 🔹	0	0 👻		
Data Type	Arraysize	Conversion		
boolean 👻	0		+/-	

Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

Parameter	Description			
Serial Baudrate	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.			
Serial Parity	unsigned 8 bit value for parity overriding. Possible values are described in the following list.			
	Value	Description		
	0	none parity		
	1	even parity		
	2	odd parity		
Serial Stop Bits	unsigned 8 bit value for stop bits overriding. Possible values are 1, 2.			
Serial Mode	unsigned 8 bit value for serial mode overriding. Possible values are described in the following list.			

Parameter	Description				
	Value	Description			
	0 RS-232 mode				
	1	RS-485 mode			
	2 RS-422 mode				
Serial Done	Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1				

Modbus RTU Server			×
Modbus RTU Server			
Memory Type Serial Baudrate ✓ Coil Status ✓ Input Status ✓ Input Registers ✓ Holding Registers ✓ Node Override ID ✓ Modicon Mode ✓ Serial Baudrate ✓ Serial Parity ✓ Serial Stop Bits ✓ Serial Done ✓	Offset 0 Arraysize 0	SubIndex 0 v Conversion	+/-
	0	K Cancel	Apply Help

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.
HMIStudio		×
Multiple tag importers	are available for this protocol. Please select the importer t	ype and continue.
Version	Туре	
Modbus Generic csv v1.0	Linear	
Tag Editor exported xml 1.1	General	
	ОК	Cancel

Туре	Description		
Modbus Generic csv v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.		
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.		
	Hags × + - ≫ © © >] ▷ ♣B ▷] È] ↓ Data ^ Tag URI		

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols						
+ - 🎽 🛍 🔎	>] [> 🗞 🖂 🗟	R 🔎 - Search 🍸 Filter b	oy: Data 🔻 Ita	ems usea	6/10000 Protocol: Show a	all 🛛 🗹 Show all tags 🖉 🗖
Data	Type	Comment	^	Proper	ty	Value
Modbus TCP:prot1	Container			✓ Dri	iver	
Model: Modicon Modbus(1-	based)				Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort				Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			V Di	tionary	
 Holding Registers 3 	unsignedShort			+ 010	Luonary	
- MRTU1	unsignedShort				Array	false
- MRTU2	unsignedShort				Array size	0
- MRTU3	unsignedShort				Arrayindex.Subindex	400003
- MRTU4	unsignedShort				Comment	
- MRTU5	unsignedShort				Data type	unsignedShort
						-

Toolbar item	Description			
	Import Tag(s).			
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project			
₿ ₩	Update Tag(s).			
	Click on this icon to update the tags in the project, due a new dictionary import.			
R	Check this box to import all sub-elements of a tag.			
	Image X Type Image X Type			
P - Search Y Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo- box item selected.			

Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat,..., [Comment]



Field	Description
NodelD	Node the tag belongs to
TagName	Tag description
MemoryType	 OUTP INP IREG HREG
Address	Offset compatible with Modbus notation

n

Field	Description
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

Tag file example

Example of .csv line:

2, Holding Register 1, HREG, 400001, unsignedShort,



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Communication status

Current communication status can be displayed using system variables. This communication protocol acts as server and doesn't return any specific Protocol Error Message.

See "System Variables" section in the main manual.

Modbus TCP

Various Modbus TCP-capable devices can be connected to HMI devices. To set-up your Modbus TCP device, please refer to the documentation you have received with the device.

The implementation of the protocol operates as a Modbus TCP client only.

Implementation details

This Modbus TCP implementation supports only a subset of the Modbus TCP standard function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the HMI device Coil area.
02	Read Input Status	Reads the ON/OFF status of the discrete inputs (1x reference) in the slave.
03	Read Holding Registers	Reads multiple registers.
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave.
05	Force Single Coil	Forces a single coil to either ON or OFF.
06	Preset Single Register	Writes a value to one register.
15	Write Multiple Coils	Writes each coil in a sequence of coils to either ON or OFF.
16	Preset Multiple Registers	Writes values to a block of registers in sequence.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the PLC list.

The protocol configuration dialog is displayed.

Modbus TCP		×
		OK
		UK
Alias		Cancel
IP address	0.0.00	
Port	502	
use UDP/IP		
Encapsulated RTU		
Timeout (ms)	2000	
Server Busy Timeout (ms)	0	
Busy Retry Time (ms)	20	
Modbus ID	1	
Max read block	250	
Max read bit block	2000	
Write Holding Register	16 ~	
Write Coils	15 ~	
PLC Models		
Modicon Modbus(1-based)		
Enron Modbus(1-based) with 3	32bit registers	
Enron Modbus(0-based) with 3 EPSON Robot	32bit registers	

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Address of the controller.
Port	Port number used by the Modbus TCP driver. The default value is 502 and can be changed when the communication goes through routers or Internet gateways where the default port number is already in use.
use UDP/IP	If selected, the protocol will use connectionless UDP datagrams.
Encapsulate	If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP

Element	Description
d RTU	protocol, independently from TCP or UDP usage.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
Modbus ID	Usually used when communicating over Ethernet-to-serial gateways and then interpreted as the Slave ID. This value is simply copied into the Unit Identifier field of the Modbus TCP communication frame. This must correspond to server configuration. In most cases, server answers to Modbus ID 1, so this parameter can be left 1.
Max read block	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.
Max read bit block	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.
Write Holding Register	Modbus function for write operations to Holding Registers. Select between the function 06 (preset single register) and function 16 (preset multiple registers).
	If 06 is selected, the protocol will always use function 06 for writing to the controller, even when writing to multiple consecutive registers.
	If 16 is selected, the protocol will always use function 16 to write to the controller, even for a single register write request and the Max read block size parameter of the query is set to 2 . The use of function 16 may result in higher communication performance.
	If Auto is selected, the protocol will use both function 06 or function 16 depending on number of registries to be written.
Write Coils	Modbus function for write operations to Output Coils. Select between the function 05 (write single coil) and function 15 (write multiple coils).
	If Modbus function 05 is selected, the protocol will always use function 05 for writing to the controller, even when writing to multiple consecutive coils.
	If Modbus function 15 is selected, the protocol will always use function 15 to write to the controller, even for a single coil write request. The use of function 15 may result in higher communication performance.

Element	Description					
PLC Models	Allows to select between different PLC models:					
	 Modicon Modbus (1-based): Modbus implementation where all resources starts with offset 1. 					
	 Generic Modbus (0-based): Modbus implementation where all resources starts with offset 0. 					
	 Enron Modbus (1-based): Extends Modicon Mobdus implementation with 32 bit registers memory area. 					
	 Enron Modbus (0-base): Extends Generic Modbus implementation with 32 bit registers memory area. 					
	Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.					
PLC Network	IP address for all controllers in multiple connections. PLC Network must be selected to enable multiple connections.					

Element	Description			
	Modbus TCP	Modbus TCP		×
	PLC Network			ОК
	Alias	Alias		Cancel
	IP address	IP address	0.0.0.1	
	Port	Port	502	
	use UDP/IP	use UDP/IP		
	Encapsulated RT	Encapsulated RTU		
	Timeout (ms)	Timeout (ms)	2000	
	Server Busy Timeou	Modbus ID	1	
	Busy Retry Time (ms	Max read block	250	
	Modbus ID	Max read bit block	2000	
	Max read block	Write Holding Register	16 ~	
	Max read bit block	Write Coils	15 ~	
	Write Holding Regist	PLC Models Modicon Modbus(1-based Generic Modbus(0-based)	i))	I
	Write Coils	Enron Modbus(1-based) v Enron Modbus(0-based) v EPSON Robot	with 32bit registers with 32bit registers	
	PLC Models Modicon Modbus(1-			
	Generic Modbus(0-b Enron Modbus(1-ba Enron Modbus(0-ba EPSON Robot	based) sed) with 32bit registers sed) with 32bit registers		
	Slaves	Add	Delete Modify	
	Slave Id	Model Alias		

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Modbus TCP** from the **Driver** list: tag definition dialog is displayed.

Modbus TCP				
Memory Type	Offset	SubIndex		
Input bits	▼ 100001	0 *		
Data Type	Arraysize	Conversion		
boolean	•] [+/-	

Element	Description			
Memory	Modbus resource where tag	is located.		
туре	Memory Type	Modbus Resource		
	Coil Status	Coils		
	Input Status	Discrete Input		
	Input Registers	Input Registers		
	Holding registers	Holding Registers		
	32 bit Registers	32 bit registers memory area.		
		Available only for Enron Modbus PLC Models		
	Node Override IP			
	Node Override Port	nuetoool novemeter (aas Cresiel Dete Turnes fer mede deteile)		
	Node Override ID	protocol parameter (see Special Data Types for mode details)		
	Modicon Mode			
Offset	Offset address where tag is	located.		
	Offset addresses are six dig	its composed by one digit data type prefix + five digits resource address.		

Element	Description				
	Memory Type	Studio Offset range	Modicon Offset range	9	Generic Modbus Offset range
	Coil Status	0 – 65535			
	Input Status	100000 – 165535			
	Input Registers	300000 - 365535	1 – 65536		0 – 65535
	Holding Registers	400000 - 465535			
	32 bit Registers	0 – 65535			
SubIndex	This allows resource offset sele	ection within the register.			
Data Type	Data Type	Memory Space		Limits	
	boolean	1-bit data		0 1	
	byte	8-bit data		-128 127	
	short	16-bit data		-32768 32	.767
	int	32-bit data		-2.1e9 2.1	e9
	int64	64-bit data		-9.2e18 9	2e18
	unsignedByte	8-bit data		0 255	
	unsignedShort	16-bit data		0 65535	
	unsignedInt	32-bit data		0 4.2e9	
	uint64	64-bit data		0 1.8e19	
	float	IEEE single-precision 32 point type	-bit floating	1.17e-38 3	3.4e38
	double	IEEE double-precision 64 point type	4-bit floating	2.2e-308	1.79e308
	string	Array of elements contain encoding	ning character	code defined	by selected
	binary	Arbitrary binary data			
	Note: to define array []", "short[]"	s. select one of Data Type	format followe	ed by square b	orackets like "byte
Arraysize	In case of array tag, thisIn case of string tag, this	property represents the n property represents the n	umber of array	velements. ber of bytes av	vailable in the



Element	Description			
	string tag.			
	Note: number of bytes or Latin1 in Tag Editor. If Encoding property is bytes.	orresponds to number of string characters if Encoding property is set to UTF-8 set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2		
Conversion	Conversion to be appli	ed to the tag.		
	Conversion			
	Allowed Configured BCD AB->BA ABCD->CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB ABCD=>CDAB Cancel OK			
	Depending on data type selected, the list Allowed shows one or more conversion types.			
	Value	Description		
Inv bits inv : Invert all the bits of the tag.		inv : Invert all the bits of the tag.		
	Example: 1001 \rightarrow 0110 (in binary format) $9 \rightarrow 6$ (in decimal format)Negateneg: Set the opposite of tag value. Example: $25.36 \rightarrow -25.36$			
	AB -> BA	swapnibbles: Swap nibbles in a byte.		
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
	ABCD -> CDAB	swap2: Swap bytes in a word.		
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
	ABCDEFGH -> GHEFCDAB	swap4 : Swap bytes in a double word. <i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format)		

Element	Description		
	Value	Description	
		$855441236 \rightarrow 1426062386$ (in decimal format)	
	ABCNOP -> OPMDAB	swap8: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011010000101101000011100101	
	BCD	 bcd: Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble) 	
	Select conversion and	I click +. The selected item will be added to list Configured .	

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

Memory Type	Offset	SubIndex		
Node Override IP 👻	0	0 -		
Data Type	Arraysize	Conversion		
unsignedByte [] 🛛 🔻	4		+/-	

Node Override Port

The protocol provides the special data type Node Override Port which allows you to change the network Port of the target controller at runtime.

This memory type is unsigned short.

Node Override Port is initialized with the value of the controller Port specified in the project at programming time.

Node Override Port	Modbus operation
0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0	It is interpreted as the value of the new port and is replaced for runtime operation.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override Port variable.



Note: Node Override Port values assigned at runtime are retained through power cycles.

Memory Type	Offset	SubIndex		
	0 Interview	Conversion		
unsignedShort 👻	0		+/-	

Node Override ID

The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.



Note: Node Override ID value assigned at runtime is retained through power cycles.

lodbus TCP				
Memory Type	Offset	SubIndex		
Node Override ID 🔹	0	0 *		
Data Type	Arraysize	Conversion		
unsignedByte 👻	0		+/-	

Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

Memory Type	Offset	SubIndex		
Data Type	Arravsize	Conversion		
boolean ·	•		+/-	

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	Z	ß	ß	>]	Þ	A 9B	B>	š 3	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers are available for	this protocol. Please select the importer type and continue.
Version	Туре
Modbus Generic csv v1.0	Linear
DELTA PLC csv v1.0	Linear
Automation Direct Productivity Series csv v1.0	Linear
Schneider SoMachine Basic csv v1.0	Linear
CODESYS2 sym v1.0	Linear
Kollmorgen txt v1.0	Linear
Schneider Unity v1.0	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Туре	Description	
Modbus Generic csv v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.	
DELTA PLC csv v1.0	Requires a .csv file. All variables will be displayed at the same level.	
CODESYS2 sym v1.0 Linear	Requires a .sym file. All variables will be displayed at the same level. After selecting the .sym file, the following dialog will appear for PLC model selection.	

Туре	Description		
	Modbus TCP importer - Filter selection Image: Concelement of the selection Available PLC Models OK ABB Cancel MAGO Image: Concelement of the selection		
Kollmorgen txt v1.0 Linear	Requires a .txt file. All variables will be displayed at the same level.		
Schneider Unity v1.0 Linear	Requires a .uny file. The file containing symbols must be exported in .txt format and later renamed as .uny . The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.		
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.		

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖉 🔰	SB 🖬 👬	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🗹 Show all tags 🔅 🗖
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	Container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			Y Dictionary	
Holding Registers 3	unsignedShort			Array	false
- MRTU1	unsignedShort			Array	Taise
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort



Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops X • A set Size Pring vol 2 • A set Size Pring vol 2 • Constance • Constance <td< th=""></td<>
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Modbus Generic csv file structure

n.

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat,..., [Comment]

Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description	
NodelD	Node the tag belongs to	
TagName	Tag description	
MemoryType	 OUTP INP IREG HREG 	
Address	Offset compatible with Modbus notation	

Field	Description
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

Tag file example

Example of .csv line:

2, Holding Register 1, HREG, 400001, unsignedShort,



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
No response	No reply within the specified timeout.	Check if the controller is connected and properly configured to get network access.
Incorrect node address in responseThe device received a response with an invalid node address from the controller .		-
The received message too shortThe device received a response with an invalid format from the controller .		-
Incorrect writing data acknowledge	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

Modbus TCP Server

Modbus TCP Server communication driver allows connecting the HMI device as a server in a Modbus TCP network. It is possible for Modbus TCP clients to connect then to multiple HMI panels acting as servers. Standard Modbus TCP messages are used for information exchange.

This approach allows connecting HMI devices to SCADA systems through the universally supported Modbus TCP communication protocol.

Principle of operation

This communication driver implements a Modbus TCP Server unit in HMI device. A subset of the complete range of Modbus function codes is supported. The available function codes allow data transfer between clients on the TCP network and the server. The HMI device acts as a server in the network. It can exchange data with up to 32 clients. This means that up to 32 clients can be connected to the HMI device at the same time. If all the 32 available connections are in use, any further attempt to connect by a client will be refused by the system.

The following diagram shows the system architecture.



The device simulates the communication interface of a PLC: Coils and Registers data types are respectively boolean and 16 bit integers.

The device always access data in its internal memory. Data can be transferred to and from the Modbus Client only on the initiative of the client itself.

Implementation details

This Modbus TCP Server implementation supports only a subset of the Modbus standard function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area.
02	Read Input Status	Reads multiple bits in the device Coil area.
03	Read Holding Registers	Read multiple device Registers.

Code	Function	Description
04	Read Input Registers	Read multiple device Registers.
05	Force Single Coil	Forces a single device Coil to either ON or OFF.
06	Preset Single Register	Presets a value in a device Register.
15	Force Multiple Coils	Forces multiple device Coils to either ON or OFF.
16	Preset Multiple Registers	Presets value in multiple device Registers.
23	Read Write Multiple Registers	Read & presets values in multiple device Registers



Note: For both PLC models the Read Coil Status and Read Input Status function codes both access the same Coil memory area in the HMI device memory. The Read Holding Registers and Read Input Registers function codes both access the same Register area in the HMI device memory.

Exception Codes

Code	Description
01	Illegal Function. the function code received in the query is not supported
02	Illegal Data Address . Data Address received in the query exceeds the predefined data range (see Tag Editor Settings for detailed ranges of all types).
03	Illegal Data Value . A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Modbus TCP Server		×
PLC Network		ОК
Modbus ID	1	Cancel
Port	502	
Connection Lifetime	30	
use UDP/IP		
Encapsulated RTU		
Enron 32bit registers		
32bit reg Start	0	
32bit reg Size	0	
PLC Models Modicon Modbus (1-based) Generic Modbus (0-based) Enron Modbus (1-based) with Enron Modbus (0-based) with	32bit registers 32bit registers	

Element	Description
Modbus ID	Modbus node ID of the HMI device. Every Modbus server device in the network must have its own Modbus ID.
Port	Port number used by the Modbus TCP protocol. Default value is 502 . Set the value accordingly to the port number used by your Modbus TCP Network.
use UDP/IP	If selected, the protocol will use connectionless UDP datagrams.
Encapsulated RTU	If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP protocol, independently from TCP or UDP usage.
Enron 32bit registers	If selected, allows to define the first register address and the number of registers for 32 bit registers memory area. Note: 32 bit registers are available only for Enron Modbus PLC Models.
	32 bit registries memory area definition.
32bit reg Start	Start value represents the first register address.
32bit reg Size	Size value represents the number of registries.

Element	Description
	Note: A request to one of the registries inside this area gives a 4 byte answer.
PLC Models	Allows to select between different PLC models:
	 Modicon Modbus (1-based): Modbus implementation where all resources starts with offset 1.
	 Generic Modbus (0-based): Modbus implementation where all resources starts with offset 0.
	 Enron Modbus (1-based): Extends Modicon Mobdus implementation with 32 bit registers memory area.
	 Enron Modbus (0-base): Extends Generic Modbus implementation with 32 bit registers memory area.
	Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Modbus TCP Server** from the protocol list: tag definition dialog is displayed.

Modbus TCP Server			
Memory Type	Offset	SubIndex	
Coil status	• 1	▲ 0 ₩	
Data Type	Arraysize	Conversion	
boolean 👻	0		+/-

Element	Description				
Memory Type	Modbus resource where tag	ag is located.			
1960	Memory Type	Modbus Resource			
	Coil Status	Coils			
	Input Status	Discrete Input			
	Input Registers	Input Registers			
	Holding Registers	Holding Registers			
	32 bit Registers	32 bit registers memory a	area.		
		Available only for Enron	Modbus PLC Mo	dels.	
	Modicon Mode	protocol parameter (see	Special Data Typ	bes for m	ode details)
Offset	Offset address where tag is	located.			
	Offset addresses are six dig address.	gits composed by one digit data type prefix + five digits resource			
	Memory Type	Studio Offset range	Modicon Offset range		Generic Modbus Offset range
	Coil Status	0 – 65535			
	Input Status	100000 – 165535			
	Input Registers	300000 - 365535	1 – 65536		0 – 65535
	Holding Registers	400000 - 465535			
	32 bit Registers	0 – 65535			
SubIndex	This allows resource offset	selection within the regist	er.		
Data type	Data Type	Memory Space		Limits	
	boolean	1-bit data		01	
	byte	8-bit data		-128	127
	short	16-bit data		-32768	32767
	int	32-bit data		-2.1e9	2.1e9
	int64	64-bit data		-9.2e18	39.2e18

Element	Description		
	Data Type	Memory Space	Limits
	unsignedByte	8-bit data	0 255
	unsignedShort	16-bit data	0 65535
	unsignedInt	32-bit data	0 4.2e9
	uint64	64-bit data	01.8e19
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308
	string	Array of elements containing character selected encoding	code defined by
	binary	Arbitrary binary data	
	Note: to define arra brackets like "byte	ays. select one of Data Type format followe]", "short[]"…	ed by square
Arraysize	 In case of array tag, th In case of string tag, th in the string tag. 	is property represents the number of array is property represents the maximum num	/ elements. ber of bytes available
	Note: number of bytes corres to UTF-8 or Latin1 in Tag Edi If Encoding property is set to requires 2 bytes.	ponds to number of string characters if En tor. UCS-2BE, UCS-2LE, UTF-16BE or UTF- ²	coding property is set 16LE one character
Conversio n	Conversion to be applied to the Conversion	ne tag.	
	inv,swap2 Allov BCD AB-: ABC ABC Inv	ABA D->CDAB DEFGH->GHEFCDAB Dits Cancel OK	

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Element	Description	
	Value	Description
	Inv bits	inv: Invert all the bits of the tag.
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)
	Negate	neg: Set the opposite of tag value.
		<i>Example:</i> 25.36 → -25.36
	AB -> BA	swapnibbles: Swap nibbles in a byte.
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
	ABCD ->	swap2: Swap bytes in a word.
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH	swap4: Swap bytes in a double word.
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011001000101101000011100101
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

Modbus TCP Server				
Memory Type	Offset	SubIndex		
Modicon Mode	• 0	▲ ▼ 0 ▼		
Data Type	Arraysize	Conversion		
boolean 👻	0		+/-	

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

Version	Туре
Modbus Generic csv v1.0	Linear
CODESYS2 sym v1.0	Linear
Tag Editor exported xml	General

Importer	Description
Modbus Generic csv v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.
CODESYS2 sym v1.0 Linear	Requires a .sym file. All variables will be displayed at the same level. After selecting the .sym file, the following dialog will appear for PLC model selection. Modbus TCP Server importer - Filter s Available PLC Models Cancel
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					
+ - 🗸 🕲 🖉 🔰	> \$ ⊮ ₪ 🕅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗍
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	Container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
MOTI 11	unsignedShort			Array	false
MRT01	unsignedShort			Array size	0
MRTU3	unsignedShort			Arravindex. Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
₫3	Update Tag(s). Click on this icon to update the tags in the project, due a
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: Image: A transformer of the second seco
P - Search	Searches tags in the dictionary basing on filter combo- box item selected.

Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
NodelD	Node the tag belongs to
TagName	Tag description
MemoryType	 OUTP INP IREG HREG
Address	Offset compatible with Modbus notation
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

Tag file example

Example of .csv line:

2, Holding Register 1, HREG, 400001, unsignedShort,



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Communication status

The HMI device is a server station in the Modbus TCP network. The current implementation of the protocol doesn't report any communication error code apart from standard communication error codes related to the proper driver loading.

See "System Variables" section in the main manual.

Mitsubishi FX ETH

Mitsubishi FX ETH implements the MELSEC-F (or MC) communication protocol that can be used with FX CPUs as described in the Mitsubishi document "FX3U-ENET USER'S MANUAL", chapter 8 "Communication using MC protocol".



Note: Mitsubishi FX3U controller must be equipped with the appropriate Ethernet module: FX3U-ENET

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the protocol called "Mitsubishi FX ETH" from the list of available protocols.

PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	5551	
PLC Models		
FX2N FX2N FX3G FX3U		

Element	Description
IP address	Ethernet IP address of the controller
Port	Specifies the port number (decimal) used in the communication with the PLC.

Element	Description
PLC Model	Defines the PLC model connected
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter IP Address for all controllers.

Controller Settings with GX Developer

The Mitsubishi FX system must be properly configured for Ethernet communication using the Mitsubishi FX Configurator. Click on "Operational settings" as shown at point (1) in the following figure:

Image: Second section Module settings Image: Second section Image: Second section Image: Second section Image: Second	e View Help				
Ethernet Module settings Module 0 1 Operational settings 2 Initial settings 3 Open settings Router relay parameter E-mail settings Router relay parameter E-mail settings Set if it is needed(No setting / Already set) Default Set if it is needed(No setting / It is needed(No setting / Already set) Check Online Diagnostics Write Read Verify) 🖻 🖬 🎒				
Module 0 Operational settings 2 Initial settings 3 Open settings Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read Verify	- Ethernet Moo	dule settings			
Module 0 Operational settings 2 Initial settings 3 Open settings Router relay parameter E-mail settings Set if it is needed(No setting / Already set Check Online Transfer setup PLC remote operation Diagnostics Write Read Verify					
Image: Comparison of the settings Image: Comparison of the setting of the settings Image: Comparison of the setting of the settings Image: Comparison of the setting of the setting of the settings Image: Comparison of the setting of the		Module 0	•		
2 Initial settings 3 Open settings Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read	0	Operational settings			
Image: Set of the set of	2	Initial settings			
Router relay parameter E-mail setting: Vecessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read Verify	3	Open settings			
E-mail settings Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read		Router relay parameter			
Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online		E-mail settings			
Write Read Verify	Online	PLC remote operation	Diagnostics	1	
Write Read Verify				1	
	Write	Read	Verify		
				1	
	dv				NUM

Into Operational Settings dialog, verify the "Communication data code" is set to "Binary code",

Then type-in the Controller IP Address and confirm with [End] button.

File View Help	
Communication data code Binary code C ASCII code ASCII code C ASCII cod C ASC	unications nunication
IP address	Send frame setting
Input format DEC.	Ethernet(V2.0)
IP address 192 168 1 254	C IEEE802.3
End Cancel	KeepAlive Ping
Ready	NUM

Click now on "Initial settings" as shown at point (2) of Figure below:

File View Help			
D 🗃 🖬 🎒			
	le settings		
	io comingo		
	Module O	•	
1	Operational settings		
2	Initial settings		
3	Open settings		
	Router relay parameter		
	E-mail settings		
Necessary setting(No setting	/ Alreadyset)	Default	
Set if it is peeded(No setting	((())) () () () () () () () (
Set in it is needed(No setting	Already set	Check	
Online			
Transfer setup	PLC remote operation	Diagnostics	
Write	Read	Verify	
Ready			NUM

For proper communication between HMI and controller it is required to change "Destination existence confirmation starting interval" from the default value of 1200 to 10ms.

In case of communication error, this avoid controller keeps alive the connection for a too long time before to allow a new connection from the HMI.

Click now on "Open settings" as shown at point (3) of Figure below

Image: Second section Ethernet Module settings Image: Second section Image: Second section Image: Second section Image: Second section Image: Second section Image: Second section Image: Second section Image: Second second section Image: Second second section Image: Second secon	
Ethernet Module settings Module 0 Operational settings 2 Initial settings 3 Open settings Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read	
Module 0 1 Operational settings 2 Initial settings 3 Open settings B-mail settings B-mail settings Set if it is needed(No setting / Already set) Online Transfer setup PLC remote operation Diagnostics Write Read	
Image: Section of Sections Im	
2 Initial settings 3 Open settings Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Vvite Read	
3 Open settings Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read Verify	
Router relay parameter E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online Transfer setup PLC remote operation Diagnostics Write Read Verify	
E-mail settings Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Online Transfer setup PLC remote operation Diagnostics Write Read	
Necessary setting(No setting / Already set) Default Set if it is needed(No setting / Already set) Check Online	
Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set) Online	
Transfer setup PLC remote operation Diagnostics Write Read Verify	
Write Read Verify	

The next figure shows the "Ethernet open settings" configuration.

The detailed explanation of the meaning of each setting is available in Chapter 5.5 of the Mitsubishi "FX3U-ENET USER'S MANUAL".

"Host station Port No." defined here is the same must be used into Protocol Editor Settings chapter.
Fil	ile View Help																
Ĺ.																	
		Protocol		Open system		Open system		Fixed buffer	Fixed buffer communication procedure		Pairin; open	g	Existence confirmation	n	Host station Port No. (DEC.)	Transmission target device IP address	Transmission target device Port No. (DEC.)
	1	TCP	•	Unp <i>as</i> sive ·	-	Send 🔻	Procedure exist(MC)	•	Disable	•	No confirm	▼ 5	5551				
	2		•		•	-		•		•		•					
	3		•	· · · · · · · · · · · · · · · · · · ·	-	•		•		•		•					
	4		•	· · · · · · · · · · · · · · · · · · ·	-	•		•		•		-					
	0 6		-	 	-	-		• •		-		-					
	7		• •		÷			Ŧ		• •		Ŧ					
	8		-		Ŧ	-		•		•		-					
							End		Car	nce	1						
					_			_		_				• [



Note: the usage of more than one panel communicating with the same controller requires to define proper settings in the "Open settings" configuration dialog: one connection per each panel must be configured with proper properties

Controller Settings with GX Works2

The Mitsubishi FX system must be properly configured for Ethernet communication inside GX Works2 programming suite. FX Parameter dialog can be recalled with double-click on PLC Parameter:



Then select "Ethernet Port" tab where is possible to configure IP Address.

Verify the "Communication data code" is set to "Binary code" as shown below:

FX Parameter		
Memory Capacity PLC Name PLC S	ystem(1) PLC System(2) Positioning	Ethernet Port
Channel CH1 💌		
IP Address Setting		Open Setting
	Input Format DEC	
IP Address	192 168 1 250	Time Setting
Subpet Mask Pattern	255 255 255 0	Log Record Setting
Defects Deuter ID Address		
Derault Router IP Address	192 168 1 254	Optional Settings (Default / Changed)
Communication Data Code	7	
Binary Code		
C ASCII Code		
Disable direct connection t	o MELSOFT	
Do not respond to search	for CPU on network	
Print Window Print Window	Preview Default	Check End Cancel

Then click on "Open Settings" button to recall the "Ethernet Port Open Setting" dialog.

FX Parameter								×		
Memory Capacity PLC Name PLC System(1) PLC System(2) Positioning Ethernet Port										
Channel CH1										
Ethernet Port Open Setting										
IP Add										
		Protocol	Open System	n	Host Station Port No.	Destinati IP Addre	ion Destination ess Port No.			
Subnel	1	ТСР	MC Protocol	-	102	5				
	2	TCP •	MELSOFT Connection	on 🔻						
Defaul	3	TCP TCP	MELSOFT Connection	on 🔻				nged)		
	4	I CP		UII 🔹				1		
Communi	Inpul	t decimal value for tl	ne Host Station Port N	Vo., Des	tination IP Ad	dress and Dest	tination Port No			
Comman										
🖲 Bina			End		Cancel					
	Code									
🗌 Disable	Disable direct connection to MELSOFT									
🗌 Do not	respor	nd to search for CPL	on network							
Print Window	F	Print Window Preview	u	De	fault	Check	End	Cancel		

"Host station Port No." defined here is the same must be used into Protocol Editor Settings chapter.



Note: For FX3GE Controller, the Open System must be set as "Data Monitor" and Port set to 1025.

1

Note: the usage of more than one panel communicating with the same controller requires to define proper settings in the "Open settings" configuration dialog: one connection per each panel must be configured with proper properties.

Tag Editor Settings

Into Tag editor select the protocol "Mitsubishi FX ETH" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Mitsubishi FX ETH				
Resources	Offset	Subindex		
Input	• 0	▲ ▼ 0 ▼		
Туре	Arraysize	Conversion		
boolean 👻	0		+/-	

Element	Description								
Resources	Area of PLC where tag is located								
Offset	Offset address where tag is located.								
SubIndex	This allows resource offset selection within the register.								
Туре	Data Type	Memory Space	Limits						
	boolean	1 bit data	01						
	byte	8-bit data	-128 127						
	short	16-bit data	-32768 32767						
	int	32-bit data	-2.1e9 2.1e9						
	unsignedByte	8-bit data	0 255						
	unsignedShort	16-bit data	0 65535						

Element	Description							
	Data Type		Memory Space	Limits				
	unsignedInt		32-bit data	0 4.2e9				
	float		IEEE single-precision	1.17e-38 3.40e38				
			32-bit floating point type					
	string		Refer to "String data type of	hapter"				
	Note: brack	to defin ets like	e arrays, select one of Data Type format followed by square "byte[]", "short[]"…					
Arraysize	 In case o In case o available Note: number of 	f array t f string t in the s bytes c	array tag, this property represents the number of array elements. string tag, this property represents the maximum number of bytes in the string tag. bytes corresponds to number of string characters if Encoding property is					
	set to UTF-8 or I If Encoding prop character requir	or Latin1 in Tag Editor. roperty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one juires 2 bytes.						
Conversion	Conversion to b	e applie	d to the tag.					
	Conversion							
	inv,swap2		Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	Configured Inv bits ABCD->CDAB Cancel OK				
	Depending on d	ata type	e selected, the list Allowed s	hows one or more conversion types.				
	Value	Descri	ption					
	Inv bits	inv: Inv	vert all the bits of the tag.					
		<i>Examp</i> 1001 – 9 → 6 (ole: → 0110 (in binary format) (in decimal format)					
	Negate	neg: S	et the opposite of tag value.					
		Examp	ole:					

Element	Description							
	Value	Description						
		25.36 → - 25.36						
	AB -> BA	swapnibbles: Swap nibbles in a byte.						
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)						
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)						
	ABCDEFGH	swap4: Swap bytes in a double word.						
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)						
	ABCNOP -	swap8: Swap bytes in a long word.						
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011010001011010000111001010						
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)						
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)						
	Select conversi	on and click +. The selected item will be added to list Configured .						
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom						

Use the arrow buttons to order the configured conversions.

of list

Tag Import

Exporting Tags from PLC

The Mitsubishi FX Ethernet tag import accepts symbol files with extension "csv" created by the Mitsubishi GX Works2 (Not from GX Developer).

The ".csv" file can be exported from the Project tree, as shown in the following figure.

- 1. Right-click on the Global variable list that need to be exported,
- 2. Select "Write to CSV File..."



Into following dialog select the file name and location:

Write to CSV Fi	le	?×
Save in:	😰 Desktop 💌 🖛 🛍 🕂 🏢 -	
My Recent Documents Desktop My Documents My Computer	My Documents My Computer My Network Places	
My Network Places	File name: Global1.csv Save as type: CSV(tab delimited)(*.csv)	Save Cancel

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	z	đ	ß	>]	₽	A 9B	B>	63	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	>					
Multiple tag importer	rs are available for this protocol. Please select the importer type and continue					
Version Type						
GX Works2 v1.0	Linear					
Tag Editor exported xml 1.3	1 General					
Natched dictionary file:						
Natched dictionary file:						
Watched dictionary file:						

Importer	Description							
GX Works2 v1.0	Requires a .csv file.							
Linear	All variables will be displayed at the same level.							
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.							
	Tags x							
	🕂 — 🎽 🕲 🔊 🚺 🚺 🕹							
	Data Tag URI							

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				•
H = ≱ @ ◎ >] [> \$= B 部	R 🔎 - Search YFilter by: Data -	Iter	ms used:6/10000 Protocol: Show all	🗸 🗹 Show all tags 🖉 🗖
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			 Dictionary 	
Holding Registers 3 Unsignedshort			Array	false
MRT01 UnsignedShort			, and y	
MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
- MRTU4 unsignedShort			Comment	
- MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
(4)	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops X • • • • • • • • • • • • •
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

The codes supported for this communication driver are:

Mitsubishi FX SER

The HMI operator panels can be connected to Mitsubishi FX PLC as the network master using this communication driver.

The protocol has been designed to connect to the programming port of the PLC.

Please note that changes in the communication protocol specifications or PLC hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in PLC hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the protocol called "Mitsubishi FX SER" from the list of available protocols.

	Comm
PLC Models	Cance
FX FX0/FX0S	
FXON	
FX 1N	
-X15 =Y2N	

Element	Description
PLC Models	The list allows selecting the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.
Comm	Gives access to the serial port configuration parameters as shown in the figure below.

Element	Description	I	
	Comm Paran	neter Dialog	×
	Port Baudra Parity Data bi Stop bi Mode	te <u>9600</u> even ts <u>7</u> ts <u>1</u> RS-422	
Port	Serial port s	election:	
	Port	Series 400	Series 500/600
	com1	PLC Port	Onboard Serial Port
	com2	PC/Printer Port	Optional Module on slot #1 or #2
	com3	Not available	Optional Module on slot #3 or #4
Baud rate, Parity, Data bits, Stop bits	Communication parameters for serial communication		
Mode	Serial port m	node; available options	
	RS-232,		
	RS-485 (2 w	rires)	
	RS-422 (4 w	rires)	

Tag Editor Settings

Into Tag editor select the protocol "Mitsubishi FX SER" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Mitsubishi FX SER			
Resources	Offset	Subindex	
Input	• 0	• 0 •	
Туре	Arraysize	Conversion	
boolean 🗸	• 0	1	+/-

Element	Description					
Resources	Area of PLC where tag	Area of PLC where tag is located				
Offset	Offset address where	tag is located.				
SubIndex	This allows resource o	ffset selection within the registe	er.			
Туре	Data Type	Memory Space	Limits			
	boolean	1 bit data	01			
	byte	8-bit data	-128 127			
	short	16-bit data	-32768 32767			
	int	32-bit data	-2.1e9 2.1e9			
	unsignedByte	8-bit data	0 255			
	unsignedShort	16-bit data	0 65535			
	unsignedInt	32-bit data	0 4.2e9			
	float	IEEE single-precision	1.17e-38 3.40e38			
		32-bit floating point type				
	string	pter"				
Note: to define arrays, select one of Data Type format followed by brackets like "byte[]", "short[]"						
Arraysize	In case of array	tag, this property represents th	e number of array elements.			

Element	Description					
	Note: number o set to UTF-8 or If Encoding pro character requi	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.				
Conversion	Conversion to be applied to the tag.					
	Conversion					
	inv,swap2	Allowed Configured				
		BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
	Depending on data type selected, the list Allowed shows one or more conve					
	Value Description					
	Inv bits	inv : Invert all the bits of the tag.				
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)				
	Negate	neg : Set the opposite of tag value.				
		<i>Example:</i> 25.36 → -25.36				
	AB -> BA	swapnibbles: Swap nibbles in a byte.				
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)					
	ABCD ->	swap2: Swap bytes in a word.				
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)				
	ABCDEFGH	swap4: Swap bytes in a double word.				
	GHEFCDAB Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)					

Element	Description	
	Value	Description
	ABCNOP - > OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion of the select conversion of the select converse of	on and click +. The selected item will be added to list Configured . ions are configured, they will be applied in order (from top to bottom of list
	Use the arrow b	uttons to order the configured conversions.

Tag Import

Exporting Tags from PLC

The Mitsubishi FX Serial tag import accepts symbol files with extension "csv" created by the Mitsubishi GX Works2 (Not from GX Developer).

The ".csv" file can be exported from the Project tree, as shown in the following figure.

- 1. Right-click on the Global variable list that need to be exported,
- 2. Select "Write to CSV File..."



Into following dialog select the file name and location:



Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	Z	ß	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Ta	g URI			

The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag impo	rters are available for this protocol. Please select the importer type and contir	ue.
Version	Туре	
GX Works2 v1.0	Linear	
Tag Editor exported xml	1.1 General	
Watched dictionary file:		
Keep synchronized		
	OK Cancel	

Importer	Description				
GX Works2 v1.0 Linear	Requires a .csv file.				
	All variables will be displayed at the same level.				
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.				
	Tags x				
	+ - 🎽 🔞 🖄 >] [> 🛟 🖬 🕅 1				
	Data Tag URI				

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 - 👗 🕲 🖉 🕽 🗞 🖬 🕅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	र 🗹 Show all tags 🔅 🗖
Data Type	Comment	^	Property	Value
Modbus TCP:prot1			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			✓ Dictionary	
Holding Registers 3 unsignedShort			Array	false
- MRTU1 unsignedShort			Anay	-
MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
- MRTU4 unsignedShort			Comment	
MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
Ea Ea	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name -	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the

Error	Notes
	controller is connected and properly configured to get network access
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

Mitsubishi iQ/Q/L ETH

The Mitsubishi iQ/Q/L ETH driver supports communication with Mitsubishi controllers with integrated Ethernet port and with external Ethernet card (QJ71E71-100).

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Mitsubishi iQ/Q/L ETH" from the list of available protocols.

The driver configuration dialog is shown as in the following figure:

Mitsubishi iQ/Q/L ETH								×
PLC Network								ОК
Alias				 				Cancel
IP address	0)	. 0	0	•	0		
Port	5002	2						
PLC Models								
iQ-FX5U iQ-R Q00J/Q00/Q01 Q02/Q02H/Q05H/Q12H/Q25H							^	
QnU Q170M-PLC CPU							~	

Element	Description
IP address	Ethernet IP address of the controller
Port	Specifies the port number (decimal) used in the communication with the PLC.

rs. . If a ange values ned.
et-up ing the

Controller Settings

GX Works2

The Mitsubishi Q system must be properly configured for Ethernet communication using the Mitsubishi GX Developer software version 7 or higher, from GX Works2 software.

The Figure below shows an example of network configuration for Ethernet communication.

Ethernet operations		×
Communication data code Binary code ASCII code	☐ Initial timing	Communications) Communication
IP address Input format DEC.	•	Send frame setting Ethernet(V2.0)
IP address 192	168 10 132	C IEEE802.3
Enable Write at RUN time	e	
	End Cancel	

Please note that the communication protocol supports only Binary code communication.

The PLC system must be configured to accept incoming data from the external device.

In the GX Developer Software open "Parameters", "Network Param" and select Ethernet/ CC IE/ MELSECNET". Add the number of connections of the operator panels you want to configure in the network.

When using the Mitsubishi CPU with external Ethernet card (QJ71E71-100) the connections have to be configured according to the following figure as "Unpassive":



When the "Existence confirmation" setting has been set to Confirm, the TCP connection will be closed when it is not used (connection lost); by default the TCP port remains open and it is not possible to reconnect.



Note: The GX Developer software allows entering the conventional representation settings (decimal or hexadecimal) for the port number; in the above figure it is in hexadecimal.

In the next figures there are 2 examples about how to set "Initial settings" for 5 and 15 seconds timeout.

<u>F</u> ile <u>V</u> iew <u>H</u> elp					
Timer setting					
Module will operate with default values if setting is left to	olank				
	Setting value	Default value	In units		
TCP ULP timer	5	60	X500ms		
TCP zero window timer	2	20	X500ms		
TCP resend timer	2	20	X500ms		
TCP end timer	3	40	X500ms		
IP assembly timer	1	10	X500ms		
Response monitoring timer	5	60	X500ms		
Destination existence confirmation starting interval	8	1200	X500ms		
Destination existence confirmation interval timer	2	20	X500ms		
Destination existence confirmation resend	1	3	Times		
	-				
DNS setting					
Input format DEC.					
IP address of DNS server 1					
IP address of DNS server 7					
IP address of DNS server 3					
IP address of DNS server 4					
End	Cancel				
Ready					
neudy				NUM	
neady				NUM	///
Eile <u>V</u> iew <u>H</u> elp				NUM	
Eile <u>V</u> iew <u>H</u> elp				NUM	
<u>File ⊻iew H</u> elp				NUM	
Eile ⊻iew Help				NUM 	
Eile View Help Image: Imag	vlank			NUM 	
Eile Yiew Help Image: Imag	olank	Defention	Le unite	NUM	
Eile Yiew Help Image: Imag	olank Setting value	Default value	In units	NUM	
Eile View Help Image:	lank Setting value	Default value 60	In units X500ms	NUM	
Eile View Help Image: Constraint of the setting Timer setting Module will operate with default values if setting is left to TCP ULP timer TCP zero window timer	lank Setting value 8 4	Default value 60 20	In units X500ms X500ms	NUM	
Eile Yiew Help Image: Imag	lank Setting value 8 4 4	Default value 60 20 20	In units X500ms X500ms X500ms	NUM	
Eile Yiew Help Image: Imag	lank Setting value 8 4 4 6	Default value 60 20 20 40	In units X500ms X500ms X500ms X500ms	· NUM	
Eile Yiew Help Image: Imag	Setting value	Default value 60 20 20 40 10	In units X500ms X500ms X500ms X500ms	, , NUM)	
Eile View Help Image: Second	Setting value Setting value	Default value 60 20 20 40 10 60	In units X500ms X500ms X500ms X500ms X500ms X500ms	, , NUM)	
Eile Yiew Help Image: Second seco	Setting value Setting value 8 4 4 6 2 8 2 8 20	Default value 60 20 20 40 10 60 1200	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms	· NUM]	
Eile Yiew Help Image: Imag	Setting value Setting value 8 4 4 6 2 8 20 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	Default value 60 20 20 40 10 60 1200 20	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms		
Eile Yiew Help Image: Imag	blank Setting value 8 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
Eile Yiew Help Image: Imag	blank Setting value 8 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
Eile View Help Image: Second Seco	Vank Setting value 8 4 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	· NUM	
Eile View Help Image: Imag	Vank Setting value 8 4 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	, ,NUM]	
Eile View Help Image: Section of the section o	Setting value Setting value 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	, ,NUM]	
Eile Yiew Help Image: Imag	Vank Setting value 8 4 4 6 2 8 20 4 3	Default value 60 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	· NUM	
Eile Yiew Help Image: Imag	Vank Setting value 8 4 4 6 2 8 20 4 3	Default value 60 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	· NUM	
Eile Yiew Help Image: Imag	Vank Setting value 8 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	· NUM	
Eile Yiew Help Image: Imag	Vank Setting value	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times	· NUM	
Eile Yiew Help Image: Section of the section o	Vank Setting value	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
Eile View Help Image: Setting Module will operate with default values if setting is left to TCP ULP timer TCP zero window timer TCP end timer IP assembly timer Response monitoring timer Destination existence confirmation starting interval Destination existence confirmation interval timer Destination existence confirmation resend DNS setting Input format DEC. IP address of DNS server 1 IP address of DNS server 2 IP address of DNS server 3 IP address of DNS server 4	Vank Setting value 8 4 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
Eile Yiew Help Image: Section of the section o	Vank Setting value 8 4 4 4 6 2 8 20 4 3	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
Eile Yiew Help Image: Section of the section o	Vank Setting value 8 4 4 4 4 6 2 8 20 4 3 20 4 3 Cancel	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
File Yiew Help Image: Setting Image: Setting Module will operate with default values if setting is left to TCP ULP timer TCP zero window timer TCP resend timer TCP end timer IP assembly timer Destination existence confirmation starting interval Destination existence confirmation interval timer Destination existence confirmation resend DNS setting Input format DEC. IP address of DNS server 1 IP address of DNS server 2 IP address of DNS server 3 IP address of DNS server 4	Vank Setting value 8 4 4 4 4 6 2 8 20 4 3 20 4 3 Cancel	Default value 60 20 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		
File Yiew Help Image: Setting Image: Setting Module will operate with default values if setting is left to TCP ULP timer TCP zero window timer TCP resend timer TCP end timer IP assembly timer Response monitoring timer Destination existence confirmation starting interval Destination existence confirmation interval timer Destination existence confirmation resend DNS setting Input format DEC. IP address of DNS server 1 IP address of DNS server 2 IP address of DNS server 3 IP address of DNS server 4	Setting value Setting value Setting value Setting value Setting value Cancel	Default value 60 20 40 10 60 1200 20 3	In units X500ms X500ms X500ms X500ms X500ms X500ms X500ms Times		

When using Mitsubishi CPU with integrated Ethernet port the "Open System" settings should be changed to "MC connection"

Built	t-in	Ethernet p	ort	open settings						×
									Port No. input forma	at HEX. 💌
		Protocol		Open system		TCP connection		Host station port No.	Transmission target device IP address	Transmission target device port No.
	1	TCP	•	MC Protocol	-		•	0500		
	2	TCP	•	MC Protocol	-		•	0501		
	3	TCP	•	MC Protocol	•		٠	0502		
	4	TCP	•	MELSOFT connection	•		•			
	5	TCP	•	MELSOFT connection	•		٠			
	6	TCP	•	MELSOFT connection	•		٠			
	7	TCP	•	MELSOFT connection	•		Ŧ			
	8	TCP	•	MELSOFT connection	•		٩			
	9	TCP	•	MELSOFT connection	•		Ŧ			
	0	тср	•	MELSOFT connection	•		٩			
1	1	TCP	•	MELSOFT connection	•		٠			
	2	тср	•	MELSOFT connection	٠		٩			
1	3	TCP	•	MELSOFT connection	•		٠			
	4	тср	•	MELSOFT connection	•		٩			
1	5	TCP	•	MELSOFT connection	•		٠			
	6	ТСР	•	MELSOFT connection	•		٠			
						End	Ca	ncel		



Note: The number format for Host Station Port No. is hexadecimal, not decimal.

GX Works3

The Mitsubishi Q system must be properly configured for Ethernet communication using GX Works3 software.

The communication driver is based on SLMP function.

SLMP (Seamless Message Protocol) is a protocol for accessing SLMP-compatible devices from an external device (such as HMI) using TCP or UDP through Ethernet.

From GX Works3 software, Ethernet port parameters must be set from **Module parameter > Ethernet Port > Basic** Settings > Own Node Settings.

Project Edit Find/Replace Convert Vie	w Online Debu	ug Diagnostics Tool	Window Help				
i 🗅 😬 💾 🧉 🔹 🔹 🔹	Banal	······································	「「「「「「「「「「」」」」」	📮 📫 🛃		7	🛫 i 🎟 💷 🥝 🥥 Max.:
128 2 III M I II- III	3 10 10 18	1 10° -					
Navigation P	× 💀 ProgPou [PRG] [LD] (Read Only	音 Global [Global Lab	oel Setting]	B Module Parameter	Ethernet P	Port ×
े ट . भ							
Project		ibal [Global Label Setting]					
🖬 🕵 Program	d M	odule Parameter Ethernet	Port				
🚰 FB/FUN		etting Item List		5	Setting Item		
Giobal Label Giobal Label Giobal Giobal HHI_Giobal Giobal Gither Constraints Gither Constraints		Put the Setting Item to Search Basic Settings Com Node Settings Com Node Settings Com Node Settings Com Node Settings	nga Gorfigunston		Item IP Address IP Address Subnet Mask Default Gateway Communication Data (External Device Config External Device Config External Device Config	Code wation guration	192, 168 . 0 . 250 255, 255, 255 . 0 192, 168 . 0 . 1 Binary <detailed setting=""></detailed>
 485 Serial Port High Speed I/O Input Response Time Analog Input Analog Output Expansion Board 		tem List Find Result			Check	Res	tore the Default Settings
Memory Card Parameter							

SLMP Connection Module must be added in **Module parameter > Ethernet Port > Basic Settings > External Device Configuration > Detailed Settings > Ethernet Configuration (Built-in Ethernet Port)**. **Port No.** parameter must be the same as per **Port** parameter from Protocol Editor Settings (see images below).

									Module List
									Ethernet Selection Find Module My Favorites
-			1		Eivert Buffer	PLC	2	Sensor/Device	122 段 (122 年) 🖈 🖄 🗙
	No.	Model Name	Communication Method	Protocol	Send/Receiv e Setting	IP Address	Port No.	MAC Address	Ethernet Device (General)
100		Host Station				192, 168, 0, 250			SLMP Connection Module -
MEL	1	MELSOFT Connection Module	MELSOFT Connectic	TCP		192, 168, 0, 250			UDP Connection Module -
nEL	2	MELSOFT Connection Module	MELSOFT Connectic	TCP		192, 168, 0, 250			Active Connection Module -
-	3	MELSOFT Connection Module	MELSOFT Connectic	TCP		192, 168, 0, 250			Unpassive Connection Module -
	4	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.0.250			Fulpassive Connection Module -
-	5	SLMP Connection Module	SLMP	TCP		192, 158, 0, 250	5002		and the second sec
		Connection Connection No.1 No.2	Connection Conne No.3 No	action Co	nnection No.5			,	
c itabi	on 3 Count	Connection Connection No.2	Connection Conne No:3 No MEL	ection Co	nnection No.5			>	

Mitsubishi iQ/Q/L ETH		×
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	5002	
PLC Models		
iQ-FX5U	^	
iQ-R Q00J/Q00/Q01 Q02/Q02H/Q06H/Q12H/Q25H		
QnU Q170M-PLC CPU	~	

Note: To actually get communication with HMI it is necessary to initialize the PLC after the above settings have been applied.

To initialize the PLC it possibile to use the Run/Stop/Reset switch or by simply rebooting the PLC.

Tag Editor Settings

1

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Mitsubishi iQ/Q/L ETH** from the protocol list: tag definition dialog is displayed.

Bassing	offert	A AND AND		
Internal Relay		Subindex		
Туре	Arraysize	Conversion		
boolean	•		+/-	

Element	Description				
Resources	PLC resources. Available reso	urces are:			
	 Internal Relay Error Relay Input (hex) Output (hex) Latch Relay Link Relay Data Register Link Register Link Register Timer (Current) Counter (Current) Timer (Switch) Timer (Coil) Counter (Coil) Special Relay Special Relay Special Register File Register Input (oct) Output (oct) 				
Offset	Offset address where tag is located.				
SubIndex	Allows resource offset selection.				
Туре	Data Type	Memory Space	Limits		
	boolean	1-bit data	01		
	byte	8-bit data	-128 127		
	short	16-bit data	-32768 32767		
	int	32-bit data	-2.1e9 2.1e9		
	int64	64-bit data	-9.2e18 9.2e18		
	unsignedByte	8-bit data	0 255		
	unsignedShort	16-bit data	0 65535		
	unsignedInt	32-bit data	04.2e9		

Element	Description				
	Data Type	Memory Space	Limits		
	uint64	64-bit data	0 1.8e19		
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38		
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308		
	string	Array of elements containing character code defined by selected encoding			
	binary	Arbitrary binary data			
	Note: to define array brackets like "byte[]"	s. select one of Data Type format followe , "short[]"…	ed by square		
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 				
Conversion	Conversion to be applied to the	e tag.			
	Conversion				
	inv,swap2 Allowed BCD AB->B ABCD- ABCD- ABCDE Inv bit	d A >CDAB FGH->GHEFCDAB s Cancel OK Cancel OK	conversion types.		

Element	Description			
	Value	Description		
	Inv bits	inv: Invert all the bits of the tag.		
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)		
	Negate	neg: Set the opposite of tag value.		
		<i>Example:</i> 25.36 → -25.36		
	AB -> BA	swapnibbles: Swap nibbles in a byte.		
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
ABCD -> CDAB		swap2: Swap bytes in a word.		
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.		
-> GHEFCDAB		Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)		
	ABCNOP -	swap8: Swap bytes in a long word.		
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010		
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)		
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)		

L.

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	-	Z	ß	ß	>]	₽	A 9B	B>	53	1
Data			~		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×			
Multiple tag importers are available for this protocol. Please select the importer type and continue.				
Version	Туре			
GX Works2/GX Works3 (OCTAL IO) v1.0	Linear			
GX Works2/GX Works3 (HEX IO) v1.0	Linear			
Tag Editor exported xml 1.1	General			
Watched dictionary file:				
Keep synchronized	OK Cancel			

Importer	Description		
GX Works2/GX Works3 v1.0 Linear	Requires a .csv file. All variables will be displayed at the same level.		
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.		

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

+ - 👗 🕲 🗅 🔸] [> 🗞 🖬 🕅	R 🔎 Search 🍸 Fil	ter by: Data 🔻 Iter	ms used:6/10000 Protocol: Show all	🗹 Show all tags 🎲 📋
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	. Container			✓ Driver	
Model: Modicon Modbus(1-ba	ised)			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			× Dictionary	
 Holding Registers 3 	unsignedShort			 Dictoriary 	
- MRTU1	unsignedShort			Array	false
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
MRTUS	unsignedshort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

There we Protocole

Toolbar item	Description
	Tager X + -
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes		
NAK	Returned in case the controller replies with a not acknowledge		
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access		
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources		
General Error	Error cannot be identified; should never be reported; contact technical support		

Mitsubishi iQ/Q/L SER

The Mitsubishi iQ/Q/L SER driver supports communication with Mitsubishi controllers with integrated serial port.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Mitsubishi iQ/Q/L SER" from the list of available protocols.

The driver configuration dialog is shown as in the following figure:

Alias	-		Cancel
timeout	l.		
uncour	500	<u>.</u>	
PLC Models			
LC Models 2001/200/201 202/202H/206H/	/Q12H/Q25H		

Element	Description
Alias	Name identifying PLC. The name will be added as a prefix to each tag name.
timeout	Time delay in milliseconds between two retries in case of missing response from the device.

Element	Description			
PLC	The driver supports communication with different Mitsu	ubishi iQ, Q and L controllers.		
Model	Note: PLC Model selection has only effect on range values of variables. If a particular model is not present in the list, try selecting a similar one. If range values of variables are the same, the communication will be correctly established.			
Comm	If clicked displays the communication parameters setu Comm Parameter Dialog Port Port Baudrate 9600 Parity even Data bits	p dialog.		
	Data bits 8 Stop bits 1 Mode RS-232			
	Element	Parameter Serial part selection		
		 COM1: On-board port COM2: Optional Plug-in module plugged on slot#1 or slot#2 COM3: Optional Plug-in module plugged on slot#3 or slot#4 		
	Baudrate, Parity, Data Bits, Stop bits	Serial line parameters.		
	Mode	 Serial port mode. Available modes: RS-232. RS-485 (2 wires). RS-422 (4 wires). 		

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Mitsubishi iQ/Q/L SER** from the protocol list: tag definition dialog is displayed.

Mitsubishi iQ/Q/L SER				
Resources	Offset	Subindex		
Internal Relay	• 0	0 -		
Туре	Arraysize	Conversion		
boolean	0		+/-	

Element	Description			
Resources	PLC resources. Available resources are:			
	 Internal Relay Error Relay Input (hex) Output (hex) Latch Relay Link Relay Data Register Link Register Timer (Current) Counter (Current) Timer (Switch) Timer (Coil) Counter (Coil) Special Relay Special Relay Special Register File Register Input (oct) Output (oct) 			
Offset	Offset address where tag is located.			
SubIndex	Allows resource offset selection.			
Туре	Data Type	Memory Space	Limits	
	boolean	1-bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
	int64	64-bit data	-9.2e18 9.2e18	
	unsignedByte	8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	04.2e9	

Element	Description				
	Data Type	Memory Space	Limits		
	uint64	64-bit data	0 1.8e19		
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38		
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308		
	string	Array of elements containing character code defined to selected encoding			
	binary	Arbitrary binary data			
	Note: to define array brackets like "byte[]"	ine arrays. select one of Data Type format followed by square e "byte[]", "short[]"…			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one 				
	character requires 2 bytes.				
Conversion	Conversion to be applied to the tag. Conversion				
	inv,swap2 Allower BCD AB->B ABCD- ABCD- ABCDE Inv bit Depending on data type selected	d A >CDAB EFGH->GHEFCDAB s Cancel OK Cancel OK	conversion types.		
Element	Description				
---------	-----------------	---	--		
	Value	Description			
	Inv bits	inv : Invert all the bits of the tag. <i>Example:</i> $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)			
	Negate	neg : Set the opposite of tag value. <i>Example:</i> $25.36 \rightarrow -25.36$			
	AB -> BA	swapnibbles: Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)			
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)			
	ABCDEFGH	swap4: Swap bytes in a double word.			
	GHEFCDAB	Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)			
	ABCNOP -	swap8: Swap bytes in a long word.			
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011001000101101000011100101			
		1 10000011100 1010101000010100010110110110			
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)			
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)			

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	z	ß	ß	>]	₽	A 9B	B>	63	1
Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	×
Multiple tag importers	s are available for this protocol. Please select the importer type and continue.
Version	Туре
Mitsubishi (OCTAL IO) v1.1	Linear
Mitsubishi (HEX IO) v1.0	Linear
Tag Editor exported xml 1.1	General
	OK Cancel

Importer	Description	
Mitsubishi v1.1 Linear	Requires a .csv file generated by GX Works2/GX Works3 software. All variables will be displayed at the same level.	
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.	
	Data Tag URI	

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols			
+ - 👗 🕲 🖉 >] [> 🔩 🖻	R Search Tritter by:	Data	Show all 💽 Show all tags 🔅 🗌
Data Type	Comment	Property	Value
Modbus TCP:prot1		Y Driver	
Model: Modicon Modbus(1-based)		Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort		Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort		Y Dictionary	
Holding Registers 3 unsignedShort		Amount	false
MRTU1 unsignedShort		Array	Idise
MRTU2 unsignedShort		Array size	0
MRTU3 unsignedShort		Arrayindex.Subind	lex 400003
MRTU4 unsignedShort		Comment	
MRTU5 unsignedShort		Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
(A)	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Tags: Tags: <th< th=""></th<>
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

NMEA 0183

The NMEA 0183 driver has been developed to communicate with NMEA 0183 compatible devices trough the operator panel serial ports.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "NMEA 0183" from the list of available protocols.

The driver configuration dialog is shown in the following figure.

NMEA 0183		×
	Comm OK	
Timeout (sec)	5 Cancel	
Cho Echo		
PLC Models		_
NMEA0183 sentences		
Virtual variables		

Element	Description
Timeout (sec)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from the server device. It is expressed in seconds.
Echo	If selected the NMEA messages received on the RX channel of serial port are sent out from the TX channel. This allows to continue the NMEA network downstream of the operator panel whether required.
PLC Models	Two PLC models are available: NMEA 0183 Sentences: when selected the Tags will point univocally to the specified NMEA sentence. Virtual variables: when selected the Tag will show the value coming from any NMEA sentence of the specified type, for example any NMEA sentence of Latitude type.

Tag Editor Settings

Into Tag editor select the protocol "NMEA 0183" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

NMEA 0183		
NMEA 0183		
Variable	Data Type	Arraysize
ALM_01_TotalNumber(-	float	▼ 0
Conversion		
	+/-	
	ОК	Cancel Apply Help

Element	Description		
Variable	The NMEA Sentence or Virtual variable		
Data Type	Data Type	Memory Space	Limits
	boolean	1 bit data	01
	byte	8-bit data	-128 127
	short	16-bit data	-32768 32767
	int	32-bit data	-2.1e9 2.1e9
	unsignedByte	8-bit data	0 255
	unsignedShort	16-bit data	0 65535
	unsignedInt	32-bit data	0 4.2e9
	float	IEEE single-precision	1.17e-38 3.40e38
		32-bit floating point type	
	string	String data	
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 		
	Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor.		

Element	Description		
	If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.		
Conversion	Conversion to be applied to the tag.		
	Value	Description	
	Degrees	Shows Degrees data only from coordinates sentence	
	Minutes	Shows Minutes data only from coordinates sentence	
	Seconds	Shows Seconds data only from coordinates sentence	

List of supported NMEA 0183 commands

The NMEA 0183 commands supported from the communication protocol are the following:

- AAM_01_StatusArrivalCircle
- AAM_02_StatusPerpendicular
- AAM_03_ArrivalCircleRadius
- AAM_04_UnitsOfRadius
- AAM_05_WaypointID
- ACK_01_LocalAlarmNumber
- ALM_01_TotalNumberOfMessages
- ALM_02_MessageNumber
- ALM_03_SatelliteNumber
- ALM_04_WeekNumber
- ALM_05_SVhealth
- ALM_06_Eccentricity
- ALM_07_AlmanacReferenceTime
- ALM_08_InclinacionAngle
- ALM_09_RateOfRightAscension
- ALM_10_RootOfSemimajorAxis
- ALM_11_ArgumentOfPerigee
- ALM_12_LongitudeOfAscesionNode
- ALM_13_MeanAnomaly
- ALM_14_ClockParameter0
- ALM_15_ClockParameter1
- ALR_01_TimeOfAlarmConditionChange

ALR_02_LocalAlarmNumber

- ALR_03_AlarmCondition
- ALR_04_AlarmAcknowledgeState
- ALR_05_AlarmDescriptionState
- APB_01_StatusSNR
- APB_02_StatusLock
- APB_03_MagnitudeOfXTE
- APB_04_DirectionToStear
- APB_05_UnitsXTE
- APB_06_StatusArrivalCircle
- APB_07_StatusPerpendicular
- APB_08_BearingOriginToDestination
- APB_09_MagneticOrTrue
- APB_10_DestinatonWaypointID
- APB_11_Bearing
- APB_12_BearingMagneticOrTrue
- APB_13_HeadingToSteer
- APB_14_HeadingMagneticOrTrue
- APB_15_ModeIndicator
- BEC_01_ObservationUTC
- BEC_02_WaypointLatitude
- BEC_03_WaypointLatitudeInd
- BEC_04_WaypointLongitude
- BEC_05_WaypointLongitudeInd
- BEC_06_BearingTrue
- BEC_07_BearingTrueInd
- BEC_08_BearingMagnetic
- BEC_09_BearingMagneticInd)
- BEC_10_Distance
- BEC_11_DistanceUnits
- BEC_12_WaypointID
- BOD_01_BearingTrue
- BOD_02_BearingTrueInd
- BOD_03_BearingMagnetic



BOD_04_BearingMagneticInd

- BOD_05_DestinationWaypointID
- BOD_06_OriginWaypointID
- BWC_01_ObservationUTC
- BWC_02_WaypointLatitude
- BWC_03_WaypointLatitudeInd
- BWC_04_WaypointLongitude
- BWC_05_WaypointLongitudeInd)
- BWC_06_BearingTrue
- BWC_07_BearingTrueInd
- BWC_08_BearingMagnetic
- BWC_09_BearingMagneticInd
- BWC_10_Distance
- BWC_11_DistanceUnits
- BWC_12_WaypointID
- BWC_13_ModeIndicator
- BWR_01_ObservationUTC
- BWR_02_WaypointLatitude
- BWR_03_WaypointLatitudeInd
- BWR_04_WaypointLongitude
- BWR_05_WaypointLongitudeInd
- BWR_06_BearingTrue
- BWR_07_BearingTrueInd
- BWR_08_BearingMagnetic
- BWR_09_BearingMagneticInd
- BWR_10_Distance
- BWR_11_DistanceInd
- BWR_12_WaypointID
- BWR_13_ModeIndicator
- BWW_01_BearingTrue
- BWW_02_BearingTrueInd
- BWW_03_BearingMagnetic
- BWW_04_BearingMagneticInd
- BWW_05_ToWaypointID

BWW_06_FromWaypointID

- DBT_01_WaterDepthFeet
- DBT_02_WaterDepthFeetInd
- DBT_03_WaterDepthMeters
- DBT_04_WaterDepthMetersInd
- DBT_05_WaterDepthFathoms
- DBT_06_WaterDepthFathomsInd
- DCN_01_DeccaChainIdentifier
- DCN_02_RedZoneIdentifier
- DCN_03_RedLineOfPosition
- DCN_04_StatusRedMasterLine
- DCN_05_GreenZoneIdentifier
- DCN_06_GreenLineOfPosition
- DCN_07_StatusGreenMasterLine
- DCN_08_PurpleZoneIdentifier
- DCN_09_PurpleLineOfPosition
- DCN_10_StatusPurpleMasterLine
- DCN_11_RedLineNavigationUse, A=Valid
- DCN_12_GreenLineNavigationUse, A=Valid
- DCN_13_PurpleLineNavigationUse, A=Valid
- DCN_14_PositionUncertainty
- DCN_15_PositionUncertaintyInd
- DCN_16_FixDataBasis
- DPT_01_WaterDepth
- DPT_02_OffsetFromTransducer
- DPT_03_MaximumRangeScale
- DSC_01_FormatSpecifier
- DSC_02_Address
- DSC_03_Cattegory
- DSC_04_NatureOfDistress
- DSC_05_TypeOfCommunication
- DSC_06_PositionOrChannel
- DSC_07_TimeOrTelNo
- DSC_08_ShipMMSI



DSC_09_NatureOfDistress DSC_10_Acknowledgment DSC_11_ExpansionIndicator DSE_01_TotalNumberOfMessages DSE_02_MessageNumber DSE_03_Query_ReplyFlag DSE_04_Vessel_MMSI DSE_05_DataSet1Code DSE_06_Dataset1Data DSE_07_Dataset2Code DSE_08_Dataset2Data DSE_09_Dataset3Code DSE_10_Dataset3Data DSE_11_Dataset4Code DSE 12 Dataset4Data DSE_13_Dataset5Code DSE_14_Dataset5Data DSE_15_Dataset6Code DSE_16_Dataset6Data DSE 17 Dataset7Code DSE_18_Dataset7Data DSE_19_Dataset8Code DSE_20_Dataset8Data DSE_21_Dataset9Code DSE_22_Dataset9Data DSE_23_Dataset10Code DSE_24_Dataset10Data DSI_01_TotalNumberOfMessages DSI_02_MessageNumber DSI_03_Vessel_MMSI DSI_04_VesselCourse DSI_05_VesselType DSI_06_GeographicArea DSI 07 Commandset1Code

DSI_08_Commandset1Data

- DSI_09_Commandset2Code
- DSI_10_Commandset2Data
- DSI_11_Commandset3Code
- DSI_12_Commandset3Data
- DSI_13_ExpansionIndicator
- DSR_01_TotalNumberOfMessages
- DSR_02_MessageNumber
- DSR_03_Vessel_MMSI
- DSR_04_Dataset1Code
- DSR_05_Dataset1Data
- DSR_06_Dataset2Code
- DSR_07_Dataset2Data
- DSR_08_Dataset3Code
- DSR_09_Dataset3Data
- DSR_10_ExpansionIndicator
- DTM_01_LocalDatumCode
- DTM_02_LocalDatumSubdivisioncode
- DTM_03_LatOffset
- DTM_04_LatOffsetInd
- DTM_05_LonOffset
- DTM_06_LonOffsetInd
- DTM_07_AltitudeOffset
- DTM_08_ReferenceDatumCode
- FSI_01_TransmitingFrequency
- FSI_02_ReceivingFrequency
- FSI_03_ModeOfOperation
- FSI_04_PowerLevel
- GBS_01_UTC
- GBS_02_ExpectedLatitudeError
- $GBS_03_ExpectedLongitudeError$
- GBS_04_ExpectedAltitudeError
- GBS_05_FailedSatelliteID
- GBS_06_ProbabilityOfMissedDetection



GBS_07_EstimateOfBiasMeters GBS_08_StandardDeviationOfBiasEstimate GGA_01_UTC GGA_02_Latitude GGA_03_LatitudeInd GGA_04_Longitude GGA_05_LongitudeInd GGA_06_QualityIndicator GGA_07_NumberOfSatellitesInUse GGA_08_HorizontalDilutionOfPrecision GGA_09_Altitude GGA_10_AltitudeInd GGA_11_GeoidalSeparation GGA_12_GeoidalSeparationInd GGA_13_AgeOfDifferentialData GGA_14_DifferentialReferenceID GLC_01_GRI GLC_02_MasterTOA GLC_03_SignalStatus1 GLC_04_TD1 GLC_05_SignalStatus2 GLC_06_TD2 GLC_07_SignalStatus3 GLC_08_TD3 GLC_09_SignalStatus4 GLC_10_TD4 GLC_11_SignalStatus5 GLC_12_TD5 GLC_13_SignalStatus6 GLL_01_Latitude GLL_02_LatitudeInd GLL_03_Longitude GLL_04_LongitudeInd GLL 05 UTC

- GLL_06_Status
- GLL_07_ModeIndicator
- GNS_01_UTC
- GNS_02_Latitude
- GNS_03_LatitudeInd
- GNS_04_Longitude
- GNS_05_LongitudeInd
- GNS_06_ModeIndicator
- GNS_07_NumberOfSatellitesInUse
- GNS_08_HDOP
- GNS_09_AntennaAltitude
- GNS_10_GeoidalSeparation
- GNS_11_AgeOfDifferentialData
- GNS_12_DifferentialStationID
- GRS_01_UTC
- GRS_02_Mode
- GRS_03_RangeResidual
- GRS_04_RangeResidual
- GRS_05_RangeResidual
- GRS_06_RangeResidual
- GRS_07_RangeResidual
- GRS_08_RangeResidual
- GRS_09_RangeResidual
- GRS_10_RangeResidual
- GRS_11_RangeResidual
- GRS_12_RangeResidual
- GRS_13_RangeResidual
- GRS_14_RangeResidual
- GSA_01_Mode
- GSA_02_Mode
- GSA_03_ID
- GSA_04_ID
- GSA_05_ID
- GSA_06_ID



- GSA_07_ID
- GSA_08_ID
- GSA_09_ID
- GSA_10_ID
- GSA_11_ID
- GSA_12_ID
- GSA_13_ID
- GSA_14_ID
- GSA_15_PDOP
- GSA_16_HDOP
- GSA_17_VDOP
- GST_01_UTC
- GST_02_RMSvalueOfStandardDeviation
- GST_03_StandardDeviationOfSemiMajorAxis
- GST_04_StandardDeviationOfSemiMinorAxis
- GST_05_OrientationOfSemiMajorAxis
- GST_06_StandardDeviationOfLatitude
- GST_07_StandardDeviationOfLongitude
- GST_08_StandardDeviationOfAltitude
- GSV_01_NumberOfMessages
- GSV_02_MessageNumber
- GSV_03_NumberOfSatellitesInView
- GSV_04_SET1_SatelliteID
- GSV_05_SET1_Elevation
- GSV_06_SET1_Azimuth
- GSV_07_SET1_SNR
- GSV_08_SET2_SatelliteID
- GSV_09_SET2_Elevation
- GSV_10_SET2_Azimuth
- GSV_11_SET2_SNR
- GSV_12_SET3_SatelliteID
- GSV_13_SET3_Elevation
- GSV_14_SET3_Azimuth
- GSV_15_SET3_SNR

GSV_16_SET4_SatelliteID

GSV_17_SET4_Elevation

GSV_18_SET4_Azimuth

GSV_19_SET4_SNR

HDG_01_MagneticHeading

HDG_02_MagneticDeviation

HDG_03_MagneticDeviationInd

HDG_04_MagneticVariation

HDG_05_MagneticVariation

HDM_01_MagneticHeading

HDM_02_MagneticHeadingInd

HDT_01_Heading

HDT_02_HeadingInd

HMR_01_HeadingSensor1ID

HMR_02_HeadingSensor2ID

HMR_03_DifferenceLimit

HMR_04_HeadingSensorDifference

HMR_05_WarningFlag

HMR_06_HeadingReadingSensor1

HMR_07_StatusSensor1

HMR_08_TypeSensor1

HMR_09_DeviationSensor1

HMR_10_DeviationSensor1Ind)

HMR_11_HeadingReadingSensor

HMR_12_StatusSensor2

HMR_13_TypeSensor2

HMR_14_DeviationSensor2

HMR_15_DeviationSensor2Ind)

HMR_16_Variation

HMR_17_VariationInd)

HMS_01_HeadingSensor1ID

HMS_02_HeadingSensor2ID

HMS_03_MaximumDifference

HSC_01_CommandedHeading



- HSC_02_CommandedHeadingInd
- HSC_03_CommandedHeadingMagnetic
- HSC_04_CommandedHeadingMagneticInd
- HTC_01_Override
- HTC_02_CommandedRudderAngle
- HTC_03_CommandedRudderDirection
- HTC_04_SelectedSteeringMmode
- HTC_05_TurnMode
- HTC_06_CommandedRudderLimit
- HTC_07_CommandedOffHeadingLimit
- HTC_08_CommandedRadiusOfTurn
- HTC_09_CommandedRateOfTurn
- HTC_10_CommandedHeadingToSteer
- HTC_11_CommandedOffTrackLimit
- HTC_12_CommandedTrack
- HTC_13_HeadingReferenceInUse
- HTD_01_Override
- HTD_02_CommandedRudderAngle
- HTD_03_CommandedRudderDirection
- HTD_04_SelectedSteeringMode
- HTD_05_TurnMode
- HTD_06_CommandedRudderLimit
- HTD_07_CommandedOffHeadingLimit
- HTD_08_CommandedRadiusOfTurn
- HTD_09_CommandedRateOfTurn
- HTD_10_CommandedHeadingToSteer
- HTD_11_CommandedOffTrackLimit
- HTD_12_CommandedTrack
- HTD_13_HeadingReferenceInUse
- HTD_14_RudderStatus
- HTD_15_OffHeadingStatus
- HTD_16_OffTrackstatus
- HTD_17_VesselHeading
- LCD_01_GRI

- LCD_02_MasterSNR
- LCD_03_MasterECD
- LCD_04_Secondary1_SNR
- LCD_05_Secondary1_ECD
- LCD_06_Secondary2_SNR
- LCD_07_Secondary2_ECD
- LCD_08_Secondary3_SNR
- LCD_09_Secondary3_ECD
- LCD_10_Secondary4_SNR
- LCD_11_Secondary4_ECD
- LCD_12_Secondary5_SNR
- LCD_13_Secondary5_ECD
- MDA_01_BarometricPressureInchesOfMercury
- MDA_02_BarometricPressureInchesOfMercuryInd
- MDA_03_Barometric pressureBars
- MDA_04_Barometric pressureBarsInd
- MDA_05_AirTemperature
- MDA_06_AirTemperatureInd
- MDA_07_WaterTemperature
- MDA_08_WaterTemperatureInd
- MDA_09_RelativeHumidity
- MDA_10_AbsoluteHumidity
- MDA_11_DewPoint
- MDA_12_DewPointInd
- MDA_13_WindDirectionTrue
- MDA_14_WindDirectionTrueInd
- MDA_15_WindDirectionMagnetic
- MDA_16_WindDirectionMagneticInd
- MDA_17_WindSpeedKnots
- MDA_18_WindSpeedKnotsInd
- MDA_19_WindSpeedMs
- MDA_20_WindSpeedMsInd
- MLA_01_TotalNumberOfMessages
- MLA_02_MessageNumber



- MLA_03_SatelliteID
- MLA_04_CalendarDay
- MLA_05_GeneralizedHealth
- MLA_06_Eccentricity
- MLA_07_DOT
- MLA_08_ArgumentOfPerigee
- MLA_09_SystemTimeScaleCorrectionMSB
- MLA_10_CorrectionOfAverageValueDraconitic
- MLA_11_TimeOfAscensionNode
- MLA_12_GreenwichLongitude
- MLA_13_CorrectionToAverageValueInclination
- MLA_14_SystemTimeScaleCorrectionLSB
- MLA_15_CourseValueOfTimeScaleShift
- MSK_01_BeaconFrequency
- MSK_02_Auto_Manual_Frequency
- MSK_03_BeaconBitRate
- MSK_04_Auto_Manual_BitRate
- MSK_05_IntervalForSending
- MSK_06_ChannelNumber
- MSS_01_SignalStrength
- MSS_02_SNR
- MSS_03_BeaconFrequency
- MSS_04_BeaconBitRate
- MSS_05_ChannelNumber
- MTW_01_Temperature
- MTW_02_TemperatureInd
- MWD_01_WindDirection
- MWD_02_WindDirectionInd
- MWD_03_WindDirectionMagnetic
- MWD_04_WindDirectionMagneticInd
- MWD_05_WindSpeedKnots
- MWD_06_WindSpeedKnotsInd
- MWD_07_WindSpeedMs
- MWD_08_WindSpeedMsInd

MWV_01_WindAngle

MWV_02_Reference

MWV_03_WindSpeed

MWV_04_WindSpeedInd

MWV_05_Status

NMEA_Altitude

NMEA_Course

NMEA_Latitude

NMEA_LatitudeInd

NMEA_Longitude

NMEA_LongitudeInd

NMEA_SpeedKnots

NMEA_UTC

OSD_01_Heading

OSD_02_HeadingStatus

OSD_03_VesselCourse

OSD_04_CourseReference

OSD_05_VesselSpeed

OSD_06_SpeedReference

OSD_07_VesselSet

OSD_08_VesselDrift

OSD_09_SpeedUnits

RMA_01_Status

RMA_02_Latitude

RMA_03_LatitudeInd

RMA_04_Longitude

RMA_05_LongitudeInd

RMA_06_TimeDifferenceA

RMA_07_TimeDifferenceB

RMA_08_SpeedOverGroundKnots

RMA_09_CourseOverGround

RMA_10_MagneticVariation

RMA_11_MagneticVariationInd

RMA_12_ModeIndicator



RMB_01_DataStatus

RMB_02_CrossTrackError

- RMB_03_DirectionToSteer
- RMB_04_OriginWaypointID
- RMB_05_DestinationwaypointID
- RMB_06_DestinationwaypointLat
- RMB_07_DestinationwaypointLatInd
- RMB_08_DestinationWaypointLongitude
- RMB_09_DestinationWaypointLongitudeInd
- RMB_10_RangeToDestination
- RMB_11_BearingToDestination
- RMB_12_DestinationClosingVelocity
- RMB_13_ArrivalStatus
- RMB_14_ModeIndicator
- RMC_01_UTC
- RMC_02_Status
- RMC_03_Latitude
- RMC_04_LatitudeInd
- RMC_05_Longitude
- RMC_06_LongitudeInd
- RMC_07_SpeedOverGround
- RMC_08_CourseOverGround
- RMC_09_Date
- RMC_10_MagneticVariation
- RMC_11_MagneticVariationInd
- RMC_12_ModeIndicator
- ROT_01_RateOfTurn
- ROT_02_Status
- RPM_01_SourceShaftEngine
- RPM_02_EngineOfShaftNumber
- RPM_03_Speed
- RPM_04_PropellerPitch
- RPM_05_Status
- RSA_01_StarboardRudderSensor

RSA_02_StatusRudderSensor)

- RSA_03_PortRudderSensor
- RSA_04_StatusPortRudderSensor)
- RSD_01_Origin1Range
- RSD_02_Origin1Bearing
- RSD_03_VariableRangeMarker1
- RSD_04_BearingLine1
- RSD_05_Origin2Range
- RSD_06_Origin2Bearing
- RSD_07_VRM2
- RSD_08_EBL2
- RSD_09_CursorRange
- RSD_10_CursorBearing
- RSD_11_RangeScale
- RSD_12_RangeScaleUnits
- RSD_13_DisplayRotation
- RTE_01_TotalNumberOfMessages
- RTE_02_MessageNumber
- RTE_03_MessageMode
- RTE_04_RouteIdentifier
- RTE_05_WaypointIdentifier1
- RTE_06_WaypointIdentifier2
- RTE_07_WaypointIdentifier3
- RTE_08_WaypointIdentifier4
- RTE_09_WaypointIdentifier5
- RTE_10_WaypointIdentifier6
- RTE_11_WaypointIdentifier7
- RTE_12_WaypointIdentifier8
- RTE_13_WaypointIdentifier9
- RTE_14_WaypointIdentifier10
- SFI_01_TotalNumberOfMessages
- SFI_02_MessageNumber
- SFI_03_1stFrequency
- SFI_04_1stMode



SFI_05_2ndFrequency SFI_06_2ndMode SFI_07_3rdFrequency SFI_08_3rdMode SFI_09_4thFrequency SFI_10_4thMode SFI_11_5thFrequency SFI_12_5thMode SFI_13_6thFrequency SFI_14_6thMode STN_01_TalkerID TLB_01_TargetNumber TLB_02_LabelAssigned TLB_03_TargetNumber1 TLB 04 LabelAssigned1 TLB_05_TargetNumber2 TLB_06_LabelAssigned2 TLB_07_TargetNumber3 TLB_08_LabelAssigned3 TLB 09 TargetNumber4 TLB 10 LabelAssigned4 TLB_11_TargetNumber5 TLB_12_Labelassigned5 TLB_13_TargetNumber6 TLB_14_LabelAssigned6 TLB_15_TargetNumber7 TLB_16_LabelAassigned7 TLB_17_TargetNumber8 TLB_18_LabelAssigned8 TLB_19_TargetNumberReported TLB_20_TargetLabelAssigned TLL_01_TargetNumber TLL_02_TargetLatitude TLL 03 TargetLatitudeInd

TLL_04_TargetLongitude

TLL_05_TargetLongitudeInd

TLL_06_TargetName

TLL_07_UTC

TLL_08_TargetStatus

TLL_09_ReferenceTarget

TTM_01_TargetNumber

TTM_02_TargetDistance

TTM_03_Bearing

TTM_04_BearingInd

TTM_05_TargetSpeed

TTM_06_TargetCourse

TTM_07_TargetCourseInd

TTM_08_DistanceOfClosestPoint

TTM_09_TimeToCPA

TTM_10_SpeedAndDistanceUnits

TTM_11_TargetName

TTM_12_TargetStatus

TTM_13_ReferenceTarget

TTM_14_UTC

TTM_15_TypeOfAcquisition

TXT_01_TotalNumberOfMessages

TXT_02_MessageNumber

TXT_03_TextIdentifier

TXT_04_TextMessage

VBW_01_LongitudinalWaterSpeed

VBW_02_TransverseWaterSpeed

VBW_03_StatusWaterSpeed

VBW_04_LongitudinalGroundSpeed

VBW_05_TransverseGroundSpeed

VBW_06_StatusGroundSpeed

VBW_07_SternTransverseWaterSpeed

VBW_08_StatusSternWaterSpeed

VBW_09_SternTransverseGroundSpeed



VBW_10_StatusSternGroundSpeed

VDR_01_Direction

VDR_02_DirectionInd

VDR_03_DirectionMagnetic

- VDR_04_DirectionMagneticInd
- VDR_05_CurrentSpeed
- VDR_06_CurrentspeedInd

VHW_01_Heading

- VHW_02_HeadingInd
- VHW_03_HeadingMagnetic
- VHW_04_HeadingMagneticInd
- VHW_05_SpeedKnots
- VHW_06_SpeedKnotsInd
- VHW_07_SpeedKmh
- VHW_08_SpeedKmhInd
- VLW_01_TotalCumulativeDistance
- VLW_02_TotalCumulativeDistanceInd
- VLW_03_DistanceSinceReset
- VLW_04_DistanceSinceResetInd
- VPW_01_SpeedKnots
- VPW_02_SpeedKnotsInd)
- VPW_03_SpeedMs
- VPW_04_SpeedMsInd
- VTG_01_CourseOverGround
- VTG_02_CourseOverGroundInd
- VTG_03_CourseOverGroundMagnetic
- VTG_04_CourseOverGroundMagneticInd
- VTG_05_SpeedOverGroundKnots
- VTG_06_SpeedOverGroundKnotsInd
- VTG_07_SpeedOverGroundKmh
- VTG_08_SpeedOverGroundKmhInd
- VTG_09_ModeIndicator
- VWR_01_MeasuredWindAngle
- VWR_02_VesselHeading

VWR_03_MeasuredWindSpeed

- VWR_04_MeasuredWindSpeedInd
- VWR_05_WindSpeedMeters
- VWR_06_WindSpeedMetersInd
- VWR_07_WindSpeedKmh
- VWR_08_WindSpeedKmhInd
- VWT_01_CalculatedWindAngle
- VWT_02_VesselHeading
- VWT_03_CalculatedWindSpeed
- VWT_04_CalculatedWindSpeedInd
- VWT_05_WindSpeedMeters
- VWT_06_WindSpeedMetersInd
- VWT_07_WindSpeedKmh
- VWT_08_WindSpeedKmhInd
- WCV_01_VelocityComponent
- WCV_02_VelocityComponentInd
- WCV_03_WaypointIdentifier
- WCV_04_ModeIndicator
- WNC_01_DistanceMiles
- WNC_02_DistanceMilesInd
- WNC_03_DistanceKm
- WNC_04_DisttanceKmInd
- WNC_05_WaypointIdentifierFrom
- WNC_06_WaypointIdentifierTo
- WPL_01_WaypointLatitude
- WPL_02_WaypointLatitudeInd
- WPL_03_WaypointLongitude
- WPL_04_WaypointLongitudeInd
- WPL_05_WaypointIdentifier
- XDR_01_Transducer1Type
- XDR_02_Measurmnt1Data
- XDR_03_UnitsOfMeasure1
- XDR_04_Transducer1
- XDR_05_Transducer2Type



XDR_06_Measurment2Data

XDR_07_UnitsOfMeasure2

XDR_08_Transducer2

XDR_09_Transducer3Type

XDR_10_Measurment3Data

XDR_11_UnitsOfMeasure3

XDR_12_Transducer3

XDR_13_Transducer4Type

XDR_14_Measurment4Data

XDR_15_UnitsOfMeasure4

XDR_16_Transducer4

XDR_17_Transducer5Type

XDR_18_Measurment5Data

XDR_19_UnitsOfMeasure5

XDR_20_Transducer5

XDR_21_Transducer6Type

XDR_22_Measurment6Data

XDR_23_UnitsOfMeasure6

XDR_24_Transducer6

XDR_25_Transducer7Type

XDR_26_Measurment7Data

XDR_27_UnitsOfMeasure7

XDR_28_Transducer7

XDR_29_Transducer8Type

XDR_30_Measurment8Data

XDR_31_UnitsOfMeasure8

XDR_32_Transducer8

XTE_01_Status1

XTE_02_Status2

XTE_03_MagnitudeOfCrossTrackError

XTE_04_DirectionToSteer

XTE_05_Units

XTE_06_ModeIndicator

XTR_01_MagnitudeOfCrossTrackError

XTR_02_DirectionToSteer

XTR_03_Units

ZDA_01_UTC

ZDA_02_Day

ZDA_03_Month

ZDA_04_Year

ZDA_05_LocalZoneHours

ZDA_06_LocalZoneMinutes

ZDL_01_TimeToPoint

ZDL_02_DistanceToPoint

ZDL_03_TypeOfPoint

ZFO_01_UTC

ZFO_02_ElapsedTime

ZFO_03_OriginWaypointID

ZTG_01_UTC

ZTG_02_TimeToGo

ZTG_03_DestinationWaypointID

NMEA 2000

NMEA 2000 (IEC 61162-3) describes a low cost, moderate capacity, bi-directional multi-transmitter/multi-receiver instrument network to interconnect marine electronic devices.

This implementation of the protocol allows connection of multiple NMEA 2000-capable devices. To setup your NMEA 2000 device and to prepare the information needed for the communication driver, please refer to the documentation you have received with the device.

This communication driver identifies directly NMEA 2000 messages on the bus using the PGN number. You should keep track of the PGN generated by the equipment you want to communicate because you will need them later in the set-up phase of the user interface application.

NMEA 2000 is based on CAN bus; CAN interface is required for operation.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol "NMEA 2000" from the list of available protocols.

The driver configuration dialog is shown in figure.

MEA 2000		;	×
		ОК	
CAN channel	can0 ~	Cancel	
ISO ECU instance	0		
ISO Function Instance	0		
System Instance	0		
Claiming Address	81		
Baud Rate (kbps)	250 ~		
Timeout (ms)	2000		
PLC Models			
NMEA 2000 Device			

Element	Description
CAN channel	Select CAN port for use with this protocol.
ISO ECU	According to NMEA 2000 standard this number (0 to 7) specifies which of a group of

Element	Description	
instance	ECUs associated with a given function is referenced.	
ISO Function instance	According to NMEA 2000 standard this number (0 to 31) specifies the occurrence of a function on a particular device system of a network.	
System Instance	According to NMEA 2000 standard this number is a 4-bit field with a valid range from 0 to 15 that indicates the occurrence of devices in additional network segments, redundant or parallel networks, or sub networks.	
Claiming Address	Specifies the starting value for Address Claiming algorithm.	
Baud Rate (kbps)	Even if NMEA 2000 define 250kpbs as standard, it is possible to modify the CAN bus baud rate.	
Timeout (ms)	NMEA 2000 reading timeout expressed in milliseconds.	
	Note: if the sensor sends data every 2.5 seconds (2500), this parameter must be set to a value greater than 2500 (example: 3000).	
PLC Models	Available models are:	
	NMEA 2000 Device	

The parameters ISO ECU instance, ISO Function Instance and System Instance are also programmable on the field, according to NMEA2000 Standard V2.1, using PGN 126208. The values eventually modified on the field are saved locally by the driver and reloaded at the next start-up.

Tag Editor Settings

In Tag Editor select the protocol "NMEA 2000" from the list of protocols and add a new tag using [+] button.

NMEA 2000		×
NMEA 2000		
datatype A boolean V	rraysize Conversion	+/-
Parameter Group Number	Proprietary Code	Proprietary ID
×	0	
Index	Selector type	ISO Ecu-Function instance
1	NONE \checkmark	
ISO Function	NMEA Class / Instance	
0	0	
	OK Cano	el Apply Help

Element	Description		
Data Type	Data Type	Memory Space	Limits
	boolean	1 bit data	01
	byte	8-bit data	-128 127
	short	16-bit data	-32768 32767
	int	32-bit data	-2.1e9 2.1e9
	unsignedByte	8-bit data	0 255
	unsignedShort	16-bit data	0 65535
	unsignedInt	32-bit data	04.2e9
	float	IEEE single-precision	1.17e-38 3.40e38
		32-bit floating point type	
	string	Array of elements containing	character code defined by
	0 Note: to brackets	define arrays, select one of Da s like "byte[]", "short[]"…	ta Type format followed by square
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property set to UTF-8 or Latin1 in Tag Editor. 		
	If Encoding proper character requires	ty is set to UCS-2BE, UCS-2LE 2 bytes.	E, UTF-16BE or UTF-16LE one
Conversion	Conversion to be a	applied to the tag.	
	Conversion		
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	Configured Inv bits ABCD->CDAB
			Cancel OK
	Depending on data	a type selected, the list Allowe	d shows one or more conversion types.

Element	Description	
	Value	Description
	Inv bits	inv: Invert all the bits of the tag.
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)
	Negate	neg: Set the opposite of tag value.
		<i>Example:</i> 25.36 → -25.36
	AB -> BA	swapnibbles: Swap nibbles in a byte.
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
	ABCD ->	swap2: Swap bytes in a word.
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH	swap4: Swap bytes in a double word.
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -	swap8: Swap bytes in a long word.
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011010000101101000011100101
		1 10000011100 1010101000010100010110110110
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Element	Description	
	Select conversi	on and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).	
	Use the arrow b	outtons to order the configured conversions.
Parameter Group Number	NMEA2000 Parameter Group Number code. Refer to standard NMEA tables.	
Proprietary Code	Proprietary Code value. Refer to standard NMEA tables.	
Proprietary ID	Proprietary ID value. Refer to standard NMEA tables.	
Index	Indicates the index of the field in PGN data to point to.	
Selector Type	When adding tags it can be necessary to duplicate them to read data coming from several devices generating same physical quantity or to select several instances of same PGN. In this case the Selector of the tag must be edited.	
	In case of duplication of the tag, the selection of incoming data can be done using one of following methods:	
	NONE Selector Type not selected	
	INSTANCE	Uses a defined bitfield value in data of PGN to distinguish between the possible sources. The value of received bitfield is compared with NMEA Class / Instance parameter for matching.
	DEVICE	Uses a combination of parameters to select one of the possible devices sending the PGN. Each device is identified using its ECU instance number its function its Function Instance number and its NNEA class number.
	DEV&INST	It is a combination of both DEVICE and INSTANCE selectors. It will allow to distinguish between several PGN containing different instance numbers and sent by several devices. Uses a combination of parameters to select one of the possible devices sending the PGN.
ISO Ecu- Function Instance	ISO Ecu-Functi It must contain	on instance parameter used with Selector Type DEVICE or DEV&INST . (Function instance * 8 + ECU instance) of the selected device.

Element	Description
ISO Function	ISO Function parameter used with Selector Type DEVICE or DEV&INST . It must contain the function of the selected device.
NMEA Class / Instance	NMEA Class / Instance parameter.
	With selector INSTANCE It must contain the instance of the PGN.
	With selector DEVICE it must contain the NMEA class of the device.
	With selector DEV&INST it must contain (instance * 256 + NMEA class).

Definition of List of PGN Available in the System

The system can operate on a limited subset of PGNs defined by an additional file named "pgnTable.csv".



Note: the file pgnTable.csv is mandatory and used by the protocol and it will be automatically downloaded to the runtime. The file must be located in C:\.....\target\protocols\.

The file can be created or edited using any text editor and it contains a list of standard or proprietary PGNs that the system can manage and the specification of each field in the PGN.

Each element is composed by a line starting with "PGN:" the pgnTable is composed by two parts,

List of elements and list of fields

The meaning of each parameter is described below.

Parameter	Description
PGN Name	name assigned to the PGN by NMEA 2000 standard or by the manufacturer
PGN number	the PGN number (decimal) you can find in standard
ISO/NMEA type	indicates if PGN is following NMEA 2000 standard or other ISO standards such as J1939
SINGLE/MULTIPLE	indicated if the PGN is SINGLE frame or MULTIPLE frame type
DefaultPriority	indicates the default priority in case of transmission (not used for receive only PGN)
DefaultRate	indicates the default transmission rate in case of transmission (not used for receive only PGN)
ProprietaryCode	in case of proprietary PGN indicates the proprietary code identifying the manufacturer
Proprietary ID Field	in case of proprietary PGN it indicates the field used to select the meaning of the PGN
Proprietary ID Value	in case proprietary PGN it indicates the value of the ID identifying the PGN

Parameter	Description
InstanceIndex	Specify which of the fields in the PGN is containing the NMEA 2000 instance. Value = 0 means no instance
INPUT/OUTPUT	Indicates the direction of the PGN. For PGNs defined for reading and reply use INPUT. For PGN defined for generation use OUTPUT

Example:

PGN:, Wind Data, 130306, NMEA, SINGLE, 2, 100, 0, 0, 0, 0, INPUT

The meaning of each field parameter, including reserved bit fields, is described below.

Field	Description
FieldName	standard name assigned to the field by NMEA 2000 standard or by the manufacturer
FieldBitSize	dimension of data in bits
Field DataFormat	numerical conversion following NMEA 2000 standard (DFxx) This data is always available in PGN description or must be specified by manufacturer in case of proprietary PGNs
READ_ ONLY/REPLY	Indicates if the field can be written by the application. In such case the PGN (if received almost once) will be retransmitted embedding the new data

Examples:

PGN:, Wind Data, 130306, NMEA, SINGLE, 2, 100, 0, 0, 0, 0, INPUT

1, SequenceID, 8, DF53, READ_ONLY

2, Wind Speed, 16, DF35, READ_ONLY

3, Wind Direction, 16, DF02, READ_ONLY

4, Wind Reference, 3, DF52, READ_ONLY

5, reserved, 21, DF52, READ_ONLY

PGN:, Rudder, 127245, NMEA, SINGLE, 2, 100, 0, 0, 0, 1, INPUT

1, Rudder Instance, 8, DF52, READ_ONLY

2, Direction Order, 3, DF52, REPLY

3, reserved,5, DF52, READ_ONLY

4, Angle Order, 16, DF04, REPLY

5, Position, 16, DF04, READ_ONLY

6, reserved, 16, DF52, READ_ONLY

PGN:, Switch Bank Control, 127502, NMEA, SINGLE, 3, 1000 , 0, 0, 0, 1, OUTPUT

1, Switch Bank Instance,8, DF53, READ_ONLY

2, Switch 01,2, DF52, READ_ONLY
3, Switch 02,2, DF52, READ_ONLY

4, Switch 03,2, DF52,READ_ONLY

5, Switch 04,2, DF52, READ_ONLY

6, Switch 05,2, DF52, READ_ONLY

7, Switch 06, 2, DF52, READ_ONLY

.....

29,Switch 28,2, DF52,READ_ONLY

Definition of Configuration Information (PGN 126998)

By default the driver answer to a request of PGN 126998 with a set of 3 empty strings. It is possible to customize the answer introducing the 3 strings (up to 70 chars each) containing:

- Installation description field1
- Installation description field2
- Manufacturer Information

The three strings must be prepares into a text file named "conflnfo.txt" (case sensitive) that can be manually deployed into unit in folder <runtime>/protocols.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	-	X	ß	ß	>]	Þ	A 9B	B>	63	1
Data	1		^		-	Ta	g URI			

The following dialog shows which importer type can be selected.

HMIStudio	>
Multiple tag importers	s are available for this protocol. Please select the importer type and continue.
Version	Туре
NMEA 2000 v1.0	Linear
Tag Editor exported xml 1.1	General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Importer	Description								
NMEA 2000 v1.1	Requires a .csv file.								
Linear	All variables will be displayed at the same level.								
	A NMEA2000 PGN Table is included into Studio distribution. It can be found in <installation_folder>/target/protocols/pgnTable.csv</installation_folder>								
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.								
	Tags x								
	+ - 🎽 🔞 🔊 🔰 🚺 🕹 🖬 副								
	Data Tag URI								

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 — 🎽 🕲 🖉 🕻 📾 🕅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗌
Data Type	Comment	^	Property	Value
Modbus TCP:prot1			Y Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			Y Dictionary	
Holding Registers 3 unsignedShort			and a second sec	febra.
MRTU1 unsignedShort			Array	Taise
-MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arrayindex.Subindex	400003
MRTU4 unsignedShort			Comment	
- MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
B ≱	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
₩.	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Data Data Data Type Con Data Type Con Data Con D
	A Application Container A Application Container A Application Container A Application P Application A Application Application Application Application Application
	- 10 br - 00 br - 11 br - 12 b
	- 19 RF - 19 RT - 19 R
	- 121 PAT - 121 DAT - 161 DAT - 161 DAT - 181 DAT - 199 DAT - 100 DAT - 100 DAT
	- series 1 0/7 - berings 0/7 - series 2/7 - series 0/7 - series 0/7 - series 0/7 -
	та ₂ 2 олт та ₂ 2 олт та ₂ 3 олт та ₂ 2 олт та ₂ 3 олт та ₂ 3 олт
P → Search Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Instance Definition

If the selected PGN "supports" instances, you must modify the "instance" and "selector type" field in tag configuration.

For this Example we have defined 2 instances of "Engine Speed" for **Port Engine (Instance 0)** /**Starboard Engine (Instance 1)** as follow:

1) Open Tags tab

2) Open the editor for the already created Engine Speed Tag, then configure the tag propriety like is shown in the figure below

+ - 🎽 🕲 🖄 🔰	[> ♣B 📴 📅 R 🔎 Search	YFilter by: Data V It	tems used: 1/1	10000 Protocol: Show all	V Show all tags 🔅 🗍
Data NMEA 2000:prot1	Type Tag name Groups	Tag URI	Encodi F	Property ✓ Driver	Value
Model: NMEA 2000 Device	Container			Model	NMEA 2000 Device
(127488-2, Engine Speed)	float (127488-2, Engine Speed)	127488?0?0?2?INSTANCE?0?0?0?float		Protocol	NMEA 2000:prot1
	NIMEA 2000		·	✓ Tag	
	NIVIEA 2000			Data Type	float
	NMEA 2000			Tag name	(127488-2, Engine Speed)
	10.24 2000			PLC tag name	
				Groups	
	datatype Arraysize Conve	ersion		Tag URI	127488?0?0?2?INSTANCE?0?0?0?float
				Comment	
	float V 0	+/-		Rate	500
				R/W	R/W 1
	Parameter Group Number Proprietary Code	Proprietary ID		Active	false
		,,.		Simulator	Variables
	127488 0	÷ 0 ÷		Scaling	None
				Min value	-3.40282e+38
	Index Selector type	ISO Ecu-Function instance		Max value	3.40282e+38
	2 INSTANCE ISO Function NMEA Class / Instant 0 0	ce 2 Inst 0 : Port Engine	T	ag URI lemory address of controller.	
<	OK	Cancel Apply He			
Ready					

Selector Type: INSTANCE (this tag works with instances) **NMEA Class / Instance:** Indicates the instance of this tag (for this example, 0 -> Port Engine)

3) Press OK then rename the tag to remind which engine it refers:

	1:Pa	ge1	r	Prot	ocols	<u> </u>	Tags*	×									
+	_	8	S	Ð	ß	>]	₽	₿ 9B	B	樹	R	₽- Search		🛛 🍸 Fi	lter by: Data 🗸 🗸	Items	used: 1/10000
Dat	а								Туре		Tag	g name		Groups	Tag URI		Encoding
4	NME/ Mode	200 200	0:pr IEA :	ot1 2000	Devio	e			Conta	iner	1						
	(1274	88-2	2, Eng	jine Sp	beed_	PORT))	float		(12	27488-2, Engine Speed_PO	RT))	127488?0?0?2?INSTANCE?0?0?	0?float	
															Instance 0	> Po	rt Engine

4) Now you have to define the 2nd Engine Speed Instance for STARBOARD Engine (Instance 1). Copy and paste the Port tag as shown in the figure below:



	1:Page	e 1	Protocol	s	Tags*	×									
+	-	×	C C	>]	Þ	A 9B	B	樹	R	₽- Search	¶	Filter by: Data	~	Items	used: 1/10000
Data	1		^				Туре		Tag	g name	Groups	Tag URI			Encoding
- L	MEA 2 Model: I	2000:p NMEA	orot1 2000 Dev	ice			Conta	iner							
			а. е Та -	° eed_	PORT)		float		(12	7488-2, Engine Speed_PORT)	127488?0?0?2?INSTANC	E?0?0?	0?float	
11	2 3	Copy	Tag												
	٩.	Paste	Tag												
14	-t	Renam	ne tag												
	1														
	righ	ht cl	lick												
	1:Page	1	Protocol	s ·	Tags*	×									
+	1:Page	1	Protocol	s >]	Tags*	× §B	B>	樹	R	P - Search	F	Filter by: Data	~	Items	used: 1/10000
+ Data	1:Page	:1	Protocol	s >]	Tags*	× \$B	ि <mark>></mark> Type	K	R Tag	P- Search	Groups	Filter by: Data	~	Items	<i>used: 1/10000</i> Encoding
+ Data	1:Page	2000:p	Protocol	s >]	Tags*	× 9B	Type Conta	iner	R Tag	P- Search	Groups	Filter by: Data Tag URI	~	Items	<i>used: 1/10000</i> Encoding
+ Data	1:Page	1 3000:p	Protocol	s)) ce	Tags*	A 9B	Dype Contai float	iner	R Tag	P - Search name 17488-2, Engine Speed_PORT	Groups	Filter by: Data Tag URI 12748870?0?2?INSTANC	× E?0?0?	<i>Items</i>	<i>used: 1/10000</i> Encoding
Data	1:Page	2000:p	Protocol	s)] ce	Tags*	× \$B	Type Contai float	iner	R Tag (12	P - Search name 7488-2, Engine Speed_PORT	Groups	Filter by: Data Tag URI 127488?0?0?2?INSTANC	× E?0707	<i>Items</i> 0?float	<i>used: 1/10000</i> Encoding
Data	1:Page	2000:p NMEA Copy Past	Protocol	s)] 	Tags*	× \$B	Type Conta float	iner	R Tag (12	P - Search name 17488-2, Engine Speed_PORT	Groups	Filter by: Data Tag URI 12748870?0?2?INSTANC	¥	<i>Items</i> 0 0?float	<i>used: 1/10000</i> Encoding
+ Data	1:Page	2000:p NMEA Copy Past Rena	Protocol	ce	Tags*	× \$B	Type Conta float	iner	R Tag (12	P - Search g name 17488-2, Engine Speed_PORT	Groups	Filter by: Data Tag URI 127488?0?0?2?INSTANC	¥	Items	<i>used: 1/10000</i> Encoding

5) Open the editor for the copied Engine Speed Tag, then configure the tag propriety like is shown in the figure below:

Data	Туре	Tag name	Groups	Tag URI		Encoding	Property	Value
NMEA 2000:prot1	Container						✓ Driver	
Model: NMEA 2000 Device	Annt	(107400 0 5	Carad DODT)	107400000			Model	NMEA 2000 Device
(12/488-2, Engine Speed_PORT)	noat	(12/488-2, Engine	e Speed_PORT)	12/468/0/0	J?Z?LINSTAINCE?U?U?U?TIOAT		Protocol	NMEA 2000:prot1
	NMEA	2000				×	Y Tag	
							Data Type	float
	NMEA	2000					Tag name	(127488-2, Engine Speed_PORT)
							PLC tag name	
							Groups	
	data	type	Arraysize Co	nversion			Tag URI	0?0?2?INSTANCE?0?0?0?toat
							Comment	
	noa	t v	U		+/-		Rate	500
							R/W	R/W 📕
	Para	meter Group Number	Proprietary Code		Proprietary ID		Active	true
							Simulator	Variables
	127	488	•	-	0	-	Scaling	None
							Min value	-3.40282e+38
	Index	c	Selector type		ISO Ecu-Function instan	ce	Max value	3.40282e+38
	2		INSTANCE	\sim	0	-		
	ISO F	unction	NMEA Class / Ins	tance	`			
	0		-	•) 2		Tag URI	
<	_		24		al tradu	11-1-	Pienory address of control	c.
Ready			OK	Cano	Apply	нер		

NMEA Class / Instance: Indicates the instance of this tag (for this example, 1 -> Starboard Engine)

6) Press OK then rename the tag to remind which engine it refers:

1:Page1 Protocols Tags* x					
+ - 🎽 🕲 💋 🔰 🏷 🗞	©3 63 R	P- Search	lter by: Dat	ta ∨ Items used:2/1	0000
Data	Туре	Tag name	Groups	Tag URI	En
MMEA 2000:prot1 Model: NMEA 2000 Device	Container				
(127488-2, Engine Speed_PORT)	float	(127488 2, Engine Speed_PORT)		127488?0?0?2?INSTANCE?0?0?0?flo	at
(127488-2, Engine Speed_STARBOARD)	float 🧹	(127488-2, Engine Speed_STARBOARD		127488?0?0?2?INSTANCE?0?0?1?flo	at
				Instance 1 : Starboard	l Eng

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
the value never received	PGN of the Tag was never received
timeout on the value refresh	Tag data is too old (see timeout value in driver configuration)
Not byte boundary on dynamic field	error in PGn definition
Something wrong with the PGN	
Too many bits to use	

Omron FINS ETH

This driver supports the FINS protocol via Ethernet connection. For a list of models that support the FINS Communications Service, refer to the manufacturer's website.

Protocol Editor Settings

Omron FINS ETH		×
PLC Network		ОК
Alias		Cancel
IP address	192 . 168 . 2 . 18	
port	9600	
panel network	0	
panel node	0	
panel unit	0	
network	0	
node	18	
unit	0	
Timeout	1000	
PLC Models		
CJx/CS1x/CP1x		

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
IP address	The Ethernet IP address of the controller connected to the operator panel
Port	Defines the port number used in the communication with the PLC. The UDP Port number must match the value specified in the PLC configuration; the default value is 9600. Most

Element	Description					
	applications will use the default value.					
Network	Parameters that define the FINS address of the device.					
Node Unit	There is a conversion rule to determine the IP address of a device starting from the FINS address in the Omron network.					
	When using the FINS communication service, it is necessary to specify the node addressing according to the FINS addressing scheme. Even in this case, data must be sent and received on the Ethernet network using IP addresses. Therefore, IP addresses are converted from FINS addresses.					
	There are three ways to convert the FINS addresses into the corresponding IP address; they are:					
	Automatic generation (default)					
	IP address table					
	 Combined method (uses Automatic and IP address table) 					
	The Omron documentation contains all the details related to determine the IP address of the controller depending on the FINS address assigned to it. The next chapter shows an example of controller configuration based on IP address table.					
Panel Network	The Panel Network/Node/Unit parameters assigned to HMI should be compatible with the ones assigned in the Omron network to the PLC:					
Panel	 Network Number must match the one specified for the PLC 					
Node	 Node Number should match the last number of the IP address of the HMI; in the figure above the panel has been configured with IP address 192.168.2.15. 					
Unit	 Unit represent the possible different network cards over the same node; for the HMI should be always set to zero since there is always only one communication unit. 					

The protocol supports the connections to multiple controllers.

To enable this, check the "PLC Network" check box and provide the configuration per each node.

Omron FINS ETH			×
PLC Network			OK
Alias			Cancel
IP address	0	mron FINS ETH	×
port	9600		OK
panel network	0	Alias	Cancel
panel node	0	IP address	192 . 168 . 2 . 18
panel unit	0	port	9600
network	0	network	
node	0	node	18
unit	0	unit	
Timeout	1000	Timeout	1000
PLC Models		PLC Models	
CLOQUES DOVER DO		CJx/CS1x/CP1x	
Slaves	L	Add Delete Mo	odify
Slave Id	Model	Alias	

Controller Settings

PLC must be properly configured to handle the communication with HMI.

Below an example of configuration based on a real scenario.



Configuration windows in this chapter are depending on PLC model. Following lines must be used as guidelines for any specific configuration.

Example Setup

HMI IP address = 192.168.2.16

PLC IP address = 192.168.2.18

In Ethernet configuration Tab:

1. Make sure that last number of IP address is the same of FINS Node No.

IP Address	g 0 Inverter Positioning		
IP Address 192 . 168 . 2 . 18 Sub-net Mask 255 . 255 . 255 . 0		Del	
FINS Node No. Node 18	Broadcast	C AI 0 (4.28SD)	
TCP/IP keep-alive	[
FINS/TCP Setting FINS/UDP Setting	DNS Setting	Clock Auto Adjustment	

In FINS/UDP Setting

- 2. Set Conversion to "IP address table"
- 3. Set FINS/UDP Options to "Destination IP is changed dynamically"
- 4. Insert HMI IP address

IP Address 19 IP Address 19	FINS/UDP Port C Default (9600) C User defined	FINS/UDP © Destina © Destina	Option tion IP is changed dynamically tion IP is Not changed dynamically
Sub-net Mask 25 FINS Node No. Node 18 TCP/IP keep-alive	Conversion C Auto (dynamic) C Auto (Static) C Combined IP address table	2 IP Address Table	
0 min	-		OK
FINS/TCP Setting	FINS/UDP Setting	DNS Setting	Clock Auto Adjustment



IP Address Table can contain more than one address. In these cases make sure that index of IP addresses is consecutive: 001 192.168.002.016 002 192.168.002.017 003 192.168.002.033



Add PC IP address in IP Address Table described above to allow communication between PLC and online Simulation.

In protocol editor

- 5. Set the IP address of PLC
- 6. Insert last number of HMI IP address in panel node parameter
- 7. Insert last number of PLC IP address in node parameter

Omron FINS ETH		×
PLC Network		ОК
Alias		Cancel
IP address	192 . 168 . 2 . 18	5
port	9600	
panel network	0	▲ ▼
panel node	16 6	•
panel unit	0	•
network	0	•
node	18 7	▲ ▼
unit	0	▲ ▼
Timeout	1000	•
PLC Models		
CJx/CS1x/CP1x		

Tag Import

Exporting Tags from PLC

The Omron FINS Ethernet driver can import tag information from CX-Programmer PLC programming software. The tag import filter accepts symbol files with extension ".cxr" created by the Omron programming tool.

The ".cxr" files can be exported from the symbol table utility.

See in figure how to access the Symbol Table (if configured) from the Omron programming software.

💬 File Edit View Insert PLC Program Simulation	Tools Window	/ Help				
]] D й 🖬 🎝 🗇 🖪 🐰 🖿 🖻 🖻 🤰	୍ର 🗛 🛱	: 🏗 🕼 🛛 🖁	R 🛛 🔺 🚴	🍇 👼 📲 II	A B B	8
□ \$\$ Q Q \$ 	🗟 ныни	99999	ーやゆす	₿ҡ⊑┕╳║	🔜 🕸 🛗	
] 🗖 🗖 🖓 🖓 🖓 🖀 🕷 1.	10 16	⊧ 🐂 🐯 🗍 🕄	3 🕫 💀 4	S S ► ■ ■	▶발발♥	\geq
	Name		Data Type	Address / Value	Rack Location	Usaç
🖃 🦣 NewProject	MyData_(01	WORD	D0		Wo
E ····································	- MyData_(02	WORD	D2		Wo
👷 Symbols	- MyData_(03	WORD	D3		Wo
Softions						
Memory						
Programs						
E SewProgram1 (00)						
Section Edit						
🛱 Sectic 😓 Insert Symbol						
The CPU010 Validate Symbols						
<u>se</u> - S <u>m</u> all Icons						
B-B- List						
Details						
X Cut						
Copy						
🔂 Paste						
Delete						
N Project /						
Reusable <u>Fi</u> le	• <u>A</u> dd	to Project				
NewPLC1/NewProgram1/	3 <u>S</u> ave	e As				
Hida						
<u>E</u> loat In Main Window						
Properties						
Compile Find Report Transfer						
Save as a reusable file			NewPLC1(Net:0,N	ode:0) - Offline		

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio	>
Multiple tag importer	s are available for this protocol. Please select the importer type and continue
Version	Туре
CX-Programmer v1.1	Linear
Tag Editor exported xml 1.1	. General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

Importer	Description					
CX-Programmer v1.1	Requires a .cxr file.					
Linear	All variables will be displayed at the same level.					
Tag Editor exported xmlSelect this importer to read a generic XML file exported from Tag Editor appropriate button.						
	Tags x					
	🛨 — 🎽 🕲 🖉 🔰 🚺 🕼 👘					
	Data Tag URI					

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖉 🔰 🕨	🗞 🖏 👬 R 🔎	▼ Search Trilter by: Data ▼	Iter	ns used:6/10000 Protocol: Show all	🕑 Show all tags 🖉 🗖
Data T	ype Commer	nt	^	Property	Value
Modbus TCP:prot1	ontainer			✓ Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1 u	insignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 u	insignedShort			Y Dictionary	
Holding Registers 3 u	insignedShort			Array	false
- MRTU1 u	InsignedShort			Array	Taise
MRTU2 u	InsignedShort			Array size	0
- MRTU3 u	InsignedShort			Arrayindex.Subindex	400003
MRTU4 u	InsignedShort			Comment	
MRTU5 u	InsignedShort			Data type	unsignedShort

Toolbar item	Description
B	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
₩ Ka	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops X + -
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Override variables

The protocol provides the special data types to override the following protocol settings:

PLC Network		ОК
Alias		Cancel
IP address	0.0.	• . • 1
port	9600	2
panel network	0	3
panel node	0	
panel unit	0	€ 5
network	0	6
node	0	1
unit	0	8
Timeout	1000	
PLC Models		
CDX/CD1X/CP1X		

Tags can be created by manually add them from Tag Editor

Omron FINS ETH				X
Omron FINS ETH				
Memory Type Verride Port I/O area Auxilary area Holding area Timer completion flags Timer PVs DM area Counter completion flags Counter CVs EM area Work area Index registers Data registers Data registers Data registers Data registers Data registers Data registers Override IP Override Port Override Node Override Vode Override Vode Override PanelNet Override PanelNode Override PanelUnit	Offset 0 Type unsignedSho	Subindex 0 • Arraysize rt • 0		
	ОК	Cancel	Apply	Help

Tag Name	Description	
Override IP	Permits to override "IP address" property (1) in runtime.	
	Data type: array unsigned bytes.	
	Notes:	
	- when address is set as "0.0.0.0" communication with the controller is stopped, no request frames are generated anymore.	
	- when address is different than "0.0.0.0" it is interpreted as a real IP address to override and target PLC IP address is replaced in runtime with the new value.	
Override	Permits to override "port" property (2) in runtime.	
Port	Data type: unsignedShort.	
Override	Permits to override "network" property (6) in runtime.	
Network	Data type: unsignedByte.	
Override	Permits to override "node" property (7) in runtime.	
Node	Data type: unsignedByte.	

Tag Name	Description
Override	Permits to override "unit" property (8) in runtime.
Unit	Data type: unsignedByte.
Override	Permits to override "panel network" property (3) in runtime.
PanelNet	Data type: unsignedByte.
Override	Permits to override "panel network" property (4) in runtime.
PanelNode	Data type: unsignedByte.
Override PanelUnit	Permits to override "panel unit" property (5) in runtime.
	Data type: unsignedByte.



Note: Override Tags are initialized with the value of properties specified in Protocol Editor. Override values assigned at runtime are retained through power cycles.

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 🔏 🖻 🔊 🎝 🌔 א	Tilter by: Data V Items used:8/10000 Pr	otocol: Show all
Data Type Tag name Omron FINS ETH:prot1 Container Slave id: Alias: Node1 Container	Tag URI	Prop
Model: Instal - Strategies allocation Parallel Franke Disease - Polity Petrolectual/Presciedingth antegrateliste Heater Proj. (Petro	Import Tag(s) - Network	X
Water Level unsignedShort Node1Water Le	Please select one or more nodes to associate this dictionary and	continue.
Slave id: Alias: Node2 Container Model:	Slave id Model Alias Node 1	
	Node2	
	For each selected node import only symbols matching that node	



Note: aliasing tag names is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you



modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge; can be returned also in case the network/node/unit parameters contained in the PLC response are not matching with panel configuration
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources. The same error can be returned also in case the PLC could not complete the processing of the panel request and sent back to the panel and invalid/not completed response.
Cnt error	Returned when a specific control character in the protocol frame received does not match with the corresponding one in the request; verify the proper settings of the controller network configuration
General Error	Error cannot be identified; should never be reported; contact technical support

Omron FINS SER

This driver supports the FINS protocol via serial connetcion. For a list of models that support the FINS Communications Service, refer to the manufacturer's website.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the $\ensuremath{\text{PLC}}$ list.

The protocol configuration dialog is displayed.

Cano	C Models
	Jx/CS1x/CP1x

Element	Description
PLC	PLC models available:
Models	• CJx/CSx/CP1x
Comm	If clicked displays the communication parameters setup dialog.

Description					
Comm Parameter Dia	alog				
	ОК				
Uart	com1 💌				
Baudrate	9600 👻				
Parity	even 💌				
Data bits	7				
Stop bits	2				
Mode	RS-485				
Element	Parameter				
Element	Parameter Serial port selection.				
Element Port	Parameter Serial port selection. • COM1: device PLC port.				
Element Port	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 7 serial port on-board.				
Element Port	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 7 serial port on-board. • COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board.				
Element Port Baudrate, Parity, Data Bits, Stop bits	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 3 serial port on-board. • COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board. Serial line parameters.				
Element Port Baudrate, Parity, Data Bits, Stop bits Mode	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 4 serial port on-board. • COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board. Serial line parameters. Serial port mode. Available modes:				
Element Port Baudrate, Parity, Data Bits, Stop bits Mode	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board. • COM3: optional Plug-In module plugged on Slot 3/4 for panel with 1 serial port on-board. Serial line parameters. Serial port mode. Available modes: • RS-232.				
Element Port Baudrate, Parity, Data Bits, Stop bits Mode	Parameter Serial port selection. • COM1: device PLC port. • COM2: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board. • COM3: optional Plug-In module plugged on Slot 3/4 for panel with 1 serial port on-board. Serial line parameters. Serial port mode. Available modes: • RS-232. • RS-485 (2 wires).				

Tag Editor Settings

In Tag Editor select the protocol **Omron FINS SER**.

Add a tag using [+] button. Tag setting can be defined using the following dialog:

Omron FINS SER				
Memory Type	Offset	Subindex		
I/O area	• 0	0 👻		
Data Block	Туре		Arraysize	
0	boolean	•	0	
Conversion				
	+/-			

Element	Description		
Memory Type	Memory Type	Description	
	I/O area	Corresponds to CIO resource on PLC	
	Auxiliary area	Corresponds to A resource on PLC	
	Holding area	Corresponds to H resource on PLC	
	Timer completion flags	Corresponds to T resource on PLC	
	Timer PVs	Corresponds to TPV resource on PLC	
	DM area	Corresponds to D resource on PLC	
	Counter completion area	Corresponds to C resource on PLC	
	Counter CVs	Corresponds to CVS resource on PLC	
	EM area	Corresponds to E resource on PLC	
	Work area	Corresponds to W resource on PLC	
	Index registers	Corresponds to IR resource on PLC	
	Data registers	Corresponds to DR resource on PLC	
Offset	Starting address for the Tag. The	ne possible range depend on memory type selected.	

Element	Description			
Subindex	This parameter allow to select a single part of the resource if the selected data type is shorter than the resource data type			
Data block	Instance of resource of the PLC.			
Data Type	Available data types:			
	• boolean			
	• byte			
	• short			
	• int			
	unsignedByte			
	unsignedShortunsignedInt			
	• float			
	• double			
	• string			
	• binary			
	See "Programming concepts" section in the main manual.			
	Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]).			

Element	Description			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding properties to UTE-8 or Latin 1 in Tag Editor. 			
	If Encoding pro character requi	perty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one res 2 bytes.		
Conversion	Conversion to b	be applied to the tag.		
	inv,swap2 Depending on c	Allowed BCD AB->BA ABCD->CDAB ABCD=>CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK data type selected, the list Allowed shows one or more conversion types.		
	Value	Description		
	Inv bits	inv : Invert all the bits of the tag. <i>Example:</i> $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	neg : Set the opposite of tag value. <i>Example:</i> $25.36 \rightarrow -25.36$		
	AB -> BA	swapnibbles : Swap nibbles in a byte. <i>Example:</i> $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)		
	ABCD -> CDAB	swap2 : Swap bytes in a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.		

Element	Description	
	Value	Description
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -	swap8: Swap bytes in a long word.
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010001011010000111001010
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversi	on and click +. The selected item will be added to list Configured .
	If more convers Configured).	ions are configured, they will be applied in order (from top to bottom of list
	Use the arrow b	outtons to order the configured conversions.

Tag Import

Exporting Tags from PLC

The Omron FINS SER driver can import tag information from CX-Programmer PLC programming software. The tag import filter accepts symbol files with extension ".cxr" created by the Omron programming tool.

The ".cxr" files can be exported from the symbol table utility.

See in figure how to access the Symbol Table (if configured) from the Omron programming software.

💾 File Edit View Insert PLC Program Simulation T	ools Window Help			
] D 🚅 🖬 🙀 🥔 🖪 🐧 🖇 🖻 🖻 🙁	🗠 🎮 🕽 🕵 🖓 🤋	N? 🛛 🕭 🚴	🍇 👼 🛄 II	
호 🛒 📑 🔡 배 📰 온 오 온 🖉	<u> </u> } + F + AF 4 P 4AP →	- 0 必 告 3	∄ ┯∈ ┗ ᅛᆃ ║	🔜 l 🅸 🛗 🛙 🖆
西岡岡 🖓 🖓 🖓 🖆 😭 🕺 🗇 🗐 🌉 1.9	끓 \$\$ 🛊 🐂 懸 📗	a 🕫 🖶 🚓	S ► ■ ■ ■	에 발 할 <mark>어</mark> 가 받
	Name	Data Type	Address / Value	Rack Location Us
🖃 🂑 NewProject	- MyData 01	WORD	D0	
🖻 🛄 NewPLC1[CJ1M] Offline	- MyData 02	WORD	D2	V
🚽 📆 Symbols	- MyData_03	WORD	D3	V
IO Table and Unit Setup				
Settings				
Programs				
Programs				
Symb				
Edit				
🛱 Secti 🐜 Tocert Sumbol				
Function Bloc				
TCPU010 😺 Validate Symbols				
Large Icons				
ਿੱਖ S <u>m</u> all Icons				
B-B- List				
Details				
aδ Cu <u>t</u>				
Ba Copy				
🖪 <u>P</u> aste				
Delete				
\Project/				
Reusable <u>Fi</u> le	Add to Project			
NewPLC1/NewProgram1/	3 <u>S</u> ave As			
Lide				
Eloat In Main Window				
Properties				
Compile / Find Report & Transfer /				
Save as a reusable file		NewPLC1(Net:0.N/	ode:0) - Offline	

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio ×			
Multiple tag importer	s are available for this protocol. Please select the importer type ar	nd continue.	
Version	Туре		
CX-Programmer v1.1	Linear		
Tag Editor exported xml 1.1 General			
Watched dictionary files			
watched dictionary file:			
Keep synchronized			
	OK	Cancel	

Importer	Description
CX-Programmer v1.1	Requires a .cxr file.
Linear	All variables will be displayed at the same level.
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	Tags X
	🕂 — 🎽 🕲 🖉 🔰 🚺 🕼 🖄 🖬
	Data Tag URI

Once the importer has been selected, locate the symbol file and click **Open**.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.

Tags 🗙 Protocols					•
+ - 👗 🛍 🙆 >) [> 🛟 🖬 🚮	R 🔎 - Search 🍸 Filter by: Data	▼ Items u	sed:6/10000 Protocol: Show a	I 🛛 🗹 Show all tags 🖉 🗖
Data	Type	Comment	Pro	perty	Value
Modbus TCP:prot1	. Container		~	Driver	·
Model: Modicon Modbus(1-ba	ased) container			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort		~	Dictionary	
MOTULE	unsignedShort			Array	false
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
	1 1 1 1				

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops: X + - & @
P → Search Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

OPC UA Client

The OPC UA Client communication driver has been designed to connect HMI devices to OPC UA servers.

This implementation of the protocol operates as a client only.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "OPC UA Client" from the list of available protocols.

The driver configuration dialog is shown as in the following figure:

OPC UA Client		×
PLC Network		ОК
Alias		Cancel
Host	0.0.0.0]
Port	4840	
Timeout (ms)	1000	
Security Policy	None ~]
Security Mode	None ~]
Username]
Password]
Server Certificate		
Client Certificate		
Client Private Key		
Hostname validation		
App URI validation		
Time validation		
PLC Models		
Default		

Element	Description	
PLC Network	Enable access to multiple networked controllers. For every controller set proper options.	
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.	
IP Address	IP address of the server.	
Port	Port number where the server is listening.	
Timeout (ms)	Time delay in milliseconds between two retries in case of no response from the server device.	
Security Mode	Type of authentication:	
	None: Certificates are not used	
	Sign: Certificates only used for authentication with server.	
	 SignAndEncrypt: Certificates used for authentication with server and data encryption. 	
Security Policy	Encryption level to use (used only when Security Mode is active).	
	• Basic256	
	Basic256Sha256	
Username Password	Authentication with user name and password	
Server Certificate	Certificate for OPC UA Server.	
	Server certificate can be downloaded using tag importer. See "Remote OPC UA Server certificate" on page 542	
Client Certificate	Certificate used by OPC UA client. If blank, a certificate is automatically generated.	
Client Private Key	Key used by OPC UA client. If blank, a key is automatically generated.	
PLC Models	No options available.	

Notes:

- Before choosing security options, be aware that not all security modes might be supported by the OPC UA server. Make sure to use security mode that is supported.
- When working within a private network you do not need to provide devices' certificates because you trust used devices. On a public network, instead, the certificate will give you a guarantee of the identity of devices.

External Certificate

ASCII version of the certificate (usually a file with .pem extension) is required.

Edit the certificate files and then copy and paste the full text of your certificate to the certificate fields.

Step 1: Remove header and footer lines

----BEGIN CERTIFICATE-----

MIIDNjCCAh4CCQCJtJqjqDDUqjANBqkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJJ VDEPMA0GA1UEBwwGVmVyb25hMRQwEgYDVQQKDAtDb21wYW55TmFtZTERMA8GA1UE CwwIUiZEIFRlYW0xFDASBgNVBAMMC0hNSURldmljZUlEMB4XDTE4MDMyNjA5MTAz OFoXDTI4MDMyMzA5MTAzOFowXTELMAkGA1UEBhMCSVQxDzANBqNVBAcMB1Z1cm9u YTEUMBIGA1UECgwLQ29tcGFueU5hbWUxETAPBgNVBAsMCFImRCBUZWFtMRQwEgYD VQQDDAtITUlEZXZpY2VJRDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB ALONtzGwlrGv6cXH8i7sNWbwmx9Xo4tp20khnt/VJnDLoYHv7ZvV1vQYHom3/HiC IaWV/uUvYnXaNB1xHnPsQPV0bEEg26Np01ne8jXEHY6bcMVK3XBV3eno3adOwHA5 vio0MmF6fPQVWTfyVb4/MrcfqUke1qWk3sF1FxEtxX1RLOwNK1+G7Wbnb30j4oPL Ev60VN3DwisDzvivpW7Nv4RPjNK9XJ2DVI+/+KDCNNLlP8GpD0xB1iIpj1S8BwqZ om1+SUs10IM1cfv/AfArZj9QaIo3c2uPwkLncqQxfDvmlC1fCfsRVxm5N3bmimwC 2F6hbkZksLp7ovCx/haKhfkCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEALVjkNEa/ 40JnMZIVkSZZWGylHHGZ8rphcUPH4olbq7MkaHk7mKacYKqI/qorrIPhmKf7Y2x5 UcTN4Uff6NT0xjrMUg2Q6Lp+a/fBqOUvEebrtmd8NYbhjTs4iVYg3R/NBlgrfx9N 6Ipp060Jo0hYXjwDZU0HADnSXVABeBxzAESvLVK7mxgXypdB1D+kgcC6hL9Xv4u5 melNI24LNkRiBT35Exlo2YTu4I9YHFelc5iILvC6DpUYHeSlIEKiNmccL2DDGEBZ TscRZykvWRilXpm2WMzjbf9HE0XNRM8DTCkOscxcrYZrcTVpm0a0WH50D2531LnF XsH5sLPyOxtKFw==

----END CERTIFICATE-----

Step 2: Remove all Newline characters

MIIDNjCCAh4CCQCJtJgjqDDUqjANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJJVDEPMA0GA1.....

Step 3: Copy and paste the single text line of the certificate to the protocol dialog

Script to generate a Certificate

If you want to use your own certificate, note that the certificate must include the "Subject Alternative Name (SAN)" parameters as required by the OPC UA standard.

Here is an example of how to generate certificate files using a public OpenSSL-Win32 library (Reference: https://www.openssl.org/)

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Client
rem Generate an RSA key
   %OpenSSL% genrsa -out client-key.pem 2048
rem Creating Certificate Signing Requests
    %OpenSSL% req -new -key client-key.pem -out client.csr -subj "/ST=NY/C=US/L=New
York/O=CompanyName/OU=R&D Team/CN=OPCUAClient@%NodeName%"
rem Creating Certificate (.pem)
    echo subjectAltName=URI:urn:%NodeName%:CompanyName:OPCUAClient > san.txt
    echo
keyUsage=digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyCertSign
>> san.txt
    echo extendedKeyUsage=critical,serverAuth,clientAuth >> san.txt
    echo authorityKeyIdentifier=keyid,issuer >> san.txt
   echo basicConstraints=CA:TRUE >> san.txt
```

```
%OpenSSL% x509 -req -days 3650 -in client.csr -signkey client-key.pem -out
client.crt -extfile san.txt
rem Convert Certificate (.der)
%OpenSSL% x509 -in client.crt -outform der -out client.der
rem Not necessary files
    del san.txt
```

pause

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. Select OPC UA Client from the protocol list.
- 2. To add a tag, click +: tag definition dialog is displayed.

OPC UA Client)	×
OPC UA Client		
Memory Type	Data type Arraysize	
Tag	✓ int ✓ 0	
Conversion	Tag name	
	+/-	
	OK Cancel Apply Help	



Element	Description		
Data Type	Available data types:		
	• boolean		
	• byte		
	• short		
	• int		
	unsignedByte		
	unsignedShort		
	unsignedInt		
	• float		
	double		
	• time		
	• unito4		
	• string		
	• binary		
	See "Programming concente" section in the main manual		
	brackets.		
Arraysize	In case of array tag, this property represents the number of array elements.		
-	 In case of string tag, this property represents the maximum number of bytes 		
	available in the string tag.		
	Note: number of bytes corresponds to number of string characters if Encoding property is		
	set to UTF-8 or Latin1 in Tag Editor.		
	character requires 2 bytes.		
Conversion	Conversion to be applied to the tag.		
	Conversion		
	inv,swap2 Allowed Configured		
	BCD Inv bits AB->BA + ABCD->CDAB		
	ABCD->CDAB		
	Inv bits		
	Cancel		
	Depending on data type selected, the list Allowed shows one or more conversion types.		

Element	Description	
	Value	Description
	Inv bits	inv : Invert all the bits of the tag.
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
	Negate	neg: Set the opposite of tag value.
		<i>Example:</i> 25.36 → -25.36
	AB -> BA	swapnibbles: Swap nibbles in a byte.
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
	ABCD ->	swap2: Swap bytes in a word.
	CDAB	Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH	swap4: Swap bytes in a double word.
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP -	swap8: Swap bytes in a long word.
	> OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011001000101101000011100101
		→ 1 10000011100 1010101000010100010110110110
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

_



Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.
Tag name	Name of tag to be used in communication.



Note: Tag properties result from import process. In most cases manual creation of new tags is not necessary.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

OPC UA Client			×
OPC UA Client			
Memory Type	Data type	Arraysize	
Node Override IP ~	string	✓ 8	
Conversion	Tag name		
	ОК	Cancel Apply He	elp

Node Override Port

The protocol provides the special data type Node Override Port which allows you to change the network Port of the target controller at runtime.

This memory type is unsigned short.

Node Override Port is initialized with the value of the controller Port specified in the project at programming time.

Node Override Port	Modbus operation
0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0	It is interpreted as the value of the new port and is replaced for runtime operation.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override Port variable.



Note: Node Override Port values assigned at runtime are retained through power cycles.
OPC UA Client		×
OPC UA Client		
Memory Type	Data type	Arraysize
Node Override Port	✓ unsignedShort	✓ 0
Conversion	Tag name	
	+/-	
	ОК	Cancel Apply Help

Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

+ - ^ V & 風	a 📖 🔀	Modbus TCP:prot1		1.	
Name /	1	Group	Driver	Address	Comment
Node 5 Gal Josénsia		ADD We	disa TCP profit	1 11 Duragned/Short	
Node's Calia, hodrotta		AGED WHO	disa TCP prot 1	1 12 Duragent/Short	
Node 1/19, MATER, Java	4	Added West	disa TCP pet	1.0.0.unegred@hot	
Number's CARCESTREEOW		Add We	disa TCP profit	1.285@unwgred@ind	
Nede 5/04/7_887_hedre	itia.	Added Mile	disa TCP proti	1.1.Durreigned/Dhoft	
Note1/R_SATA_hodeos		AGED Wes	disa TCP prot 1	1.2.0 unegreef@holt	
TANK BER MAN TELE		ADD We	disa TCP prik!	1.3 Duraigned Droft	
Node1/Water_level		1000	Tertine Ethilipse	1 10 0 unsigned Short	
1	Netw	ork			
	-				
		Slave Id	Model	Alias	
		(1) 新田市(1)	Number - must	Node1	
		电振荡器	Transformer - result	Node2	
Ka K					
tagname	me				
Water_level				(
W/sritze				Ok	Cancel
R_3//7X_builtetia					
GNT, BHT, Hullinia	<u></u>	-		way and a	
CRECKLETERNC MAN	1855(72)	245	0	unsigned line	et Salva
HI, WATTER, and	1844(42)	0	0	unsigned the	et Galue
Property Personnelling	1855/721	1.2		Last manager age of The age	and Tables
Comm. Commission					



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

Importing tags

Tags for OPC UA Client protocol must be imported from OPC UA servers.

Path: ProjectView> Config > double-click Tags

- 1. Select OPC UA Client in the list of available protocols.
- 2. Click Import Tags.
- 3. Select Hierarchical importer.
- 4. Enter address of the server.
- 5. Choose Security and Authentication mode.
- 6. Click **Browse** to connect and retrieve tag dictionary from the OPC UA server.
- 7. The OPC UA Server will provide its own certificate. You have to accept the certificate to continue and retrieve data.
- 8. When the discovery process has been completed, click **OK** to create the dictionary with the tags.

OPCUA Client importer		×
Symbol discovery, click 'Browse' to pull symbols. Do you	want to continue?	
opc.tcp://192.168.44.165:48010	Manual browse	Full Browse
Security Settings		
Security Policy None		•
Security Mode None		-
Client Certificate Client's own certificate		
Private Key Client's private key		
Authentication Settings		
C Username Both security policy and mode should Both security policy and mode should Both security policy and mode should be s	d be none. d be none.	
Symbols found: 0 / Remaining nodes to process: 0 ServerName ServerName Server Server Tags Tag1 Tag2 Tag3 Mag3 Protocols Protocols Types		
	OK	Cancel

Element	Description
Remote URI	Address of OPC UA Server in the form: <i>opc:tcp:<ipaddress>:<port></port></ipaddress></i>
	Example:
	• opc.tcp://192.168.44.165:4840
Security Mode	Type of authentication:
	None: No authentication with server and no data encryption.
	 Sign: Certificates only used for authentication with server.
	 SignAndEncrypt: Certificates used for authentication with server and data encryption.
Security Policy	Encryption level to use (used only when Security Mode is active).

Element	Description	
	 Basic128Rsa15 Basic256 Basic256Sha256 	
Username Password	Authentication with user name and password	
Client Certificate	Certificate used by OPC UA client. If blank, a certificate is automatically generated. The certificate is used by the importer only if requested by the server	
Client Private Key	Key used by OPC UA client. If blank, a key is automatically generated.	



To be allowed to retrieve data from the OPC UA Server you must provide the required security parameters. Dialog will be filled automatically with the parameters provided by protocol editor settings (you can simply accept the proposed values)

Remote OPC UA Server certificate

Certif	icate details		×
Do yo Cert	u want to trust <u>t</u> tificate details	his server certificate permanently?	
	CommonN	OPCUAServer@HMI-Server	^
	Organization	CompanyName	
	Organizatio	R&D Team	
	Locality	Peronal Sec	
	State	dially setting	
	Country		
	DomainCo		
	lssuer	OPCUAServer@HMI-Server	
	Valid from	2018-06-01T10:36:06.000Z	
	Valid to	2028-05-29T10:36:06.000Z	
	Serial Num	8785CE5BDC6B570D	
	Thereasterning	L-2616660	–
Acce	pt permanently	Accept temporarily Copy to clipboard Cance	el

When OPC UA Server provides its own certificate, you have the option to:

Accept temporarily

Certificate is accepted for current working session only.

Accept permanently

Certificate is accepted and copied to computer. Any future import request for the same OPC UA Server will be accepted automatically without asking confirmation.



The certificate file will be copied inside the folder:

%AppData%\Roaming\...\studio\OPCUA\pki\trusted\certs

Copy to clipboard

ASCII format of the certificate is copied to the clipboard to allow you to verify its authenticity, save and insert it into protocol configuration (if required).



To verify a certificate, use a text editor to paste it from the clipboard to a text file with the extension .crt. You can then double-click the .crt file to allow Windows to view the properties of certificate.

0A	Certi	ficate						\times
Ge	eneral	Details	Cer	tification	Path			
	8	Certi	ficat	te Inforn	nation			-
	Thi: inst Aut	s CA Roo tall this c horities	t ce erti stor	rtificate ficate in 'e.	is not tru the Trus	isted. To ei ted Root Co	nable trust, ertification	
		Issued	to:	HMI-Serv	ver			
		Issued	by:	HMI-Ser	ver			
		Valid fr	om	24/05/20	018 to 2 3	1/05/2028		
					Install Ce	ertificate	Issuer Statement	
							OK	

Cancel

Cancel the import operation

Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Description
Connecting < Error description>	Error during connection
Connection while reading: <error description=""></error>	Error encountered when connecting for read operation
Bad status while reading: <error description=""></error>	Error in read operation

Error	Description
Connection while writing: <error description=""></error>	Error encountered when connecting for write operation
Bad status writing: <error description=""></error>	Error in write operation
OPC UA client for given node ID not found	Wrong node ID information

<Error description> can be one of the following:

Error	Notes
BadTimeout	Timeout error. No answer from server.
BadSecurityChecksFailed	Error during exchange of certificates. Typically occurs when the server does not accept the client certificate as trusted.
BadCertificatexxxInvalid	Error in client or server certificate.
BadNodeUnknown	The tag (node) does not exist.
BadAttributeNotFound	Attempt to access an invalid attribute.
BadNotWritable	Attempt to write to a read-only attribute.

Panasonic FP

The operator panels can be connected to a Panasonic FP PLC as the network master using this communication driver.

This driver has been designed for connection to the programming port of the PLC.

Please note that changes in the communication protocol specifications or PLC hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in PLC hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Panasonic FP" from the list of available protocols.

The driver configuration dialog is shown in the following figure.

Panasonic FP	×
PLC Network	Comm OK
Node ID 1	Cancel
PLC Models	
FP1 C16 FP1 C24	
FP0(2.7K)/FP1C40/FPM(2. FP0(5K)/FP1C56	7К)
FP1C72/FPM(5K) FP3	.

Element	Description
Node ID	Node number of the slave device
PLC Models	The list allows selecting the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.
Comm	Recalls the serial port configuration parameters as shown in the figure below.

Element	Description
	Comm Parameter Dialog
	OK
	Port com1
	Baudrate 9600 👻
	Parity odd
	Data bits
	Stop bits
	Mode RS-485
Port	Serial port selection.
	COM1 is the PLC port.
	COM2 is PC/Printer port on panels with two serial ports or refers to the optional
	Plug-In module plugged on Slot 1 (or 2) for panels with one serial port on-board.
	COM3 refers to the optional Plug-In module plugged on Slot 3 (or 4) for panels with one serial port on-board.
Baud rate	Communication parameters for serial communication
Parity	
Data bits	
Stop bits	



Element	Description
Mode	Serial port mode. Available options:
	RS-232,
	RS-485 (2 wires)
	RS-422 (4 wires)
PLC Network	The protocol supports connection to multiple controllers. To enable this, check the "PLC Network" check box and provide the configuration per each node.
	V PLC Network Comm Node ID -1
	PLC Models Cancel PP1 C16 PP1 C24 PP0(SX)/FP1 C56 PP1 C16 PP1 C72/FPM(SK) PP1 C24 PP3 PP1 C24 Slaves Add

Tag Import

Exporting Tags from PLC

The Panasonic FP driver supports the Tag Import facility. The symbol file can be exported by the controller programming software FPWIN.

In FPWIN menu, click on "Project > Export > Variables as CSV file", then you can choose if you want to export only the Global variables or All project variables.

:	Proje	ct Object	Edit	Tools	Online	Мо	nitor	Debug	Extras	Wir	ndow He	lp							
E		New				►			-	χĘ	a 🔁 🖉	🕫 🖓	11	帰	†	⊒	-E (2 ê
1	2	Open			Ctrl+C)		-	ąх	}	act on error	· (0-2)	🥜 Gl	obal v	ariabl	es	Drg	Pro	gramm
		Open recent	project:	s		. *					Class		Iden	tifier					-
		Save			Ctrl+S	· .	s∖ad	lmin\My	Docur	0	VAR		Merk	cer0					
		Dave as				•				1 2	VAR_E VAR_E	XTERNAL	aktu anzu	elle_⊢ izeige	IM_Se nde_l	site HM_S	eite		
		Import								3 4	VAR VAR E	XTERNAL	Merk	œr1 orrado	ositia	n			
		Export				•	05)	011 objec	+c	-	VAD		alte_ Moto	Pos	ichtur				
	면영	Compile all		Ct	rl+Shift+A			Objects	selected	l in nav	vigator		sys_	iPotiIr	nputV	1			
	9	Compile incre	ementall	y a	trl+Shift+:	I		Variable	s as CSV	file	-	+	1	Glot	pal va	riable	s		
		Used memor	y					Program	i code ar	nd PLC	configurati	ion	1	All p	projec	t vari	ables		
		Printer setup)				ation	(530-535,	981, 1			vlotorra	dp.osiți D	20	- 0111	ADE	Ď	-	T
	4	Print preview	/		Ctrl+C	2	16, 84 Ins (1	165-1169	59, 861 , 1181,)	\langle / \rangle		• •	5—					
	3	Print			Ctrl+F)	(142	1-1425, 1	437, 14		\vee								
		Open cross-r	referenc	e list							Λ		• •		· ·		· ·	• •	· ·
	H	Change secu	irity leve '	el			ıs)				$\langle \rangle$								
	0	Change pass	words	•											· ·		· ·	• •	· ·
	_	Exit			Alt+F4	ł . –	tries)	,						•					

If you choose to export only the Global variables, FPWIN will show the window of the following picture that allow to customize the elements of the exported csv file.

CSV export global variable list			×
Project Class Identifier IEC address FP address Type Initial Comment	Assign >> << Remove	Feld 1(Class)Feld 2(Identifier)Feld 3(IEC address)Feld 4(FP address)Feld 5(Type)Feld 6(Initial)Feld 7(Comment)	
		Customize column title	
Export column title as first line		Feld 1	
Export each array element also as a s	single variable	Course and and firm with a	
Only for arrays with fewer than		Save export configuration	
		Load export configuration	
elements		Reset to default configuration	
Only export variables with explicit add	dress		
(Export	Cancel Help	-

Then, in the "Export to file" window, choose the "CSV file (Unicode)" format.

Export to file							? 🛛
Save in:	🞯 Desktop		*	6	1 🖻	•	
My Recent Documents Desktop My Documents	My Documents My Computer My Network Pla Backup_FP0_W Configurazione Shortcut to Put	aces KNX Jlic chSupp					
My Computer							
S	File name:	Untitled.csv			~		Save
My Network	Save as type:	CSV files (Unicode) (*.csv)			~		Cancel

Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the Import Tags button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag importer	s are available for this protocol. Please select the importer type and continu	Je.
Version	Туре	
FPWIN v1.0	Linear	
Tag Editor exported xml 1.1	General	
Watched dictionary file:		
Keep synchronized		
	OK Cancel	

Importer	Description
FPWIN v1.0	Requires a . csv file.
Linear	All variables will be displayed at the same level.
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	Tags x
	🛨 — 🎽 🔞 🔊 🔰 🚺 🎍 🖬 🕅 1
	Data Tag URI

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags 🗙 Protocols						
+ - 🗸 🕲 🖉 👌	> 🗞 🔯 🚮	R 🔎 - Search 🍸 Filter by: Data	•	Iten	ms used:6/10000 Protocol: Show all	Show all tags 🔅 🗖
Data	Туре	Comment	1	^	Property	Value
Modbus TCP:prot1	Container				✓ Driver	
Model: Modicon Modbus(1-based)	Container .				Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort				Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort				 Dictionary 	
- MPTL1	unsignedShort				Array	false
- MRTU2	unsignedShort				Array size	0
MRTU3	unsignedShort				Arravindex.Subindex	400003
- MRTU4	unsignedShort				Comment	
- MRTU5	unsignedShort				Data type	unsignedShort

Toolbar item	Description
B	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
(4)	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops X • • • • • • • • • • • • •
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

The codes supported for this communication driver are:

Ping

Ping communication driver allows to send ping commands to a specific IP address.

The purpose of this communication driver are:

- test a connection between the HMI and another device in the same network
- check internet connectivity by executing ping commands to a public IP address (example 8.8.8.8)

In case it is needed to send ping commands to many IP addresses at the same time, it is possible to create many instances of Ping protocol:

Protocols ×		
+ - ^ V 🔊		
PLC	Configuration	
Ping:prot1	CfgVer=1 model=1 ipAddress=8.8.8.8 protocol=ICMP port=80 timeout=5000	1
Ping:prot2	CfgVer=1 model=1 ipAddress=192.168.2.2 protocol=ICMP port=80 timeout=5000	1
Ping:prot3	CfgVer=1 model=1 ipAddress=192.168.2.3 protocol=ICMP port=80 timeout=5000	1



Ping communication driver is not counted as physical protocol. Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the PLC list.

The protocol configuration dialog is displayed.

IP address	8.8.8.8	Cancel
Protocol	ICMP	•
Port	80	
Timeout (ms)	5000	
PLC Models		
default		

Element	Description
IP address	Destination IP address to which ping commands are sent.
Protocol	Network protocol used to send ping commands (default is ICMP).
Port	Network port used for sending ping commands (fixed to 53 for ICMP Protocol).
Timeout (ms)	Polling time between each ping command sent.
PLC Models	Fixed to default.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select **Ping** from the protocol list: tag definition dialog is displayed.

lemory Type	Data Type	Arraysize	
Node Override IP 🛛 👻	string	8	1
-			
Conversion	(G24)		
	+/-		

Element	Description					
Memory Type	Name	Des	Description			
, j þe	Node Override IP	lf de whic	If defined, this Tag allows to change the destination IP address to which ping commands are sent, at runtime.			
	Status	Rep	resents the result of last pi	ng command:		
		 0 = last ping command failed 				
		1 = last ping command got response				
	Last ping time	Represents the result of last ping time, expressed in milliseconds.				
Data Type	Data Type		Memory Space	Limits		
.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	boolean		1-bit data	01		
	unsignedByte[]		8-bit data	0 255		
	unsignedShort		16-bit data	0 65535		
	unsignedInt		32-bit data	0 4.2e9		
	string		Express the number of characters used to specify the destination IP address <i>Example: string[15]> xxx.xxx.xxx</i>			
Arraysiz e	This property represents the maximum number of bytes available in the string or in the array Tag.					
	Note: number of bytes co UTF-8 or Latin1 in Tag E	orresp ditor.	oonds to number of string o	hars if Encoding property is set to		

If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one char requires 2 bytes.

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	_	z	ß	ß	>]	₽	A 9B	B>	63	1
Data	1		^		-	Та	g URI			

The system will require a generic XML file exported from Tag Editor by appropriate button.

	Tags	×								
+	-	×	ß	ß	>]	Þ	A 9B	B>	š 3	1
Data	1		^			Та	g URI			

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols				
🕂 — 🎽 🕲 🖉 🔰 🚯	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 💮 🗖
Data Type	Comment	^	Property	Value
Modbus TCP:prot1 Container			✓ Driver	
Model: Modicon Modbus(1-based)			Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedShort			✓ Dictionary	
MDTU1			Array	false
- MRTU2 unsignedShort			Array size	0
MRTU3 unsignedShort			Arravindex.Subindex	400003
MRTU4 unsignedShort			Comment	
MRTU5 unsignedShort			Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description				
	Tags* x Tags* x				
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.				

ProConOS ETH

The ProConOS ETH driver has been developed for the connection to ProConOS compatible controllers via Ethernet.

Yaskawa MPiec controllers that can communiucate using ProConOSdriver are:

- MP2300Siec
- MP2310iec

For such models it is possible to export variables to be imported in Tag Editor (see Tag Import chapter).

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "ProCoNos ETH" from the list of available protocols.

ProConOS ETH			×
PLC Network		ОК	
Alias		Cancel	
IP Address	0.0.0.0		
port	20547		
Motorola Byte Order			
Timeout	3000		
PLC Models			
Generic ProConOS Con	troller		

Element	Description
Alias	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
IP Address	Controller IP address
Port	Controller port number for Ethernet interface
Motorola Byte Order	This option is used to identify if the PLC you're working with is a Big Endian type (default, option checked), or Little Endian (option unchecked).
Timeout	The time the protocol waits the answer from the controller before issuing a new retry.

Element	Description
PLC Models	List of compatible controller models. Make sure to select the right model in this list when configuring the protocol.
PLC Notwork	The protocol supports connection to multiple controllers.
Network	To enable this, check the "PLC Network" check box and provide the configuration
	per each node.

ProConOS ETH		23		
PLC Network		ProConOS ETH		×
Alias				ОК
IP Address	0.0.0.0	Alias		Cancel
port	20547	IP Address	192 . 168 . 0 . 10	
🥅 Motorola Byte Or	'der	port	20547	
Timeout	3000	Motorola Byte Order	r	
PLC Models	[Timeout	3000	
Generic ProConOS	Controller	PLC Models Generic ProConOS Cor	ntroller	
Slaves	Add			
Slave Id	Model Alias			

Data Types

The import module supports variables of standard data types as per the following list.

- BOOL
- SINT (8-bits signed integers)
- INT (16-bit signed integers)
- DINT (32-bits signed integers)
- USINT (8-bits unsigned integers)
- BYTE (8-bits unsigned integers)
- UINT (16-bit unsigned integers)
- WORD (16-bit bit strings, displayed as unsigned integers)
- UDINT (32-bits unsigned integers)
- DWORD (32-bit bit strings, displayed as unsigned integers)
- REAL (32-bit floating point data)
- LREAL (64-bit floating point data)
- TIME
- STRING (character string)

Tag Conversion

Conversion to be applied to the tag.

Conversion			
inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ - ~	Configured Inv bits ABCD->CDAB
		(Cancel OK

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
Negate	neg : Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
ABCNOP ->	swap8: Swap bytes in a long word.
OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011001000101101

Value	Description
	\rightarrow 1 10000011100 101010000101000101101101100101101
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Special Data Types

The ProCoNos Ethernet driver provides one special data type called "Node Override IP".

The Node Override IP allows changing at runtime the IP address of the target controller you want to connect. This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

If the IP Override is set to 0.0.0.0, all the communication with the node is stopped, no request frames are generated anymore.

If the IP Override has a value different from 0.0.0.0, it is interpreted as node IP override and the target IP address is replaced at runtime with the new value.

In case the panel has been contabld to access to a network of controllers, each node has its own Override variable.



Note: the IP Override values assigned at runtime are retained through power cycles.

ProConOS ETH		
ProConOS ETH		
Memory Type	Tag Name	Parent Struct
offset	subindex Data Type 0 unsignedByte	
Arraysize	Conversion +/]
	ОК	Cancel Apply Help

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

1:Page1 Protocols Tags* 🗙				
+ - 2 0 0 >] [> 🗞	в 🖬 🚮	R 🔎 Search	T Filter by: Data	✓ Items used: 11/10000
Data	Type Tag	name	Groups Tag URI	E
ProConOS ETH:prot1	Container			
Slave id: 192.168.0.1 Alias: Node 1 Model: Generic ProConOS Controller @GV/PLC_TSYS_TICK_CNT @GV/PLC_TASK_AVAILABLE @GV/PLC_TASK_DEFINED @GV/PLC_TICKS_PER_SEC @GV/PLCDEBUG_BPSET @GV/PLCDEBUG_FORCE @GV/PLCDEBUG_POWERFLOW @GV/PLCMDE_HAIT	Container int Node short Node short Node short Node boolean Node boolean Node	e1/@GV/PLC_SYS_TICK_CNT e1/@GV/PLC_TASK_AVAILABLE e1/@GV/PLC_TASK_DEFINED e1/@GV/PLC_TICKS_PER_SEC e1/@GV/PLCDEBUG_BPSET e1/@GV/PLCDEBUG_FORCE e1/@GV/PLCDEBUG_POWERFLOW	12@GV/PLC_SYS_TICK 12@GV/PLC_TASK_AVA 12@GV/PLC_TASK_DEF 12@GV/PLC_TICKS_PEF 12@GV/PLCDEBUG_BOS 12@GV/PLCDEBUG_POV 12@GV/PLCDEBUG_POV 12@GV/PLCDEBUG_POV	_CNT??DINT?0?int ILABLE??INT?0?short INED??INT?0?short R_SEC??INT?0?short ET??BOOL?0?boolean RCE??BOOL?0?boolean WERFLOW??BOOL?0?boolean WERFLOW??BOOL?0?boolean
@GV/PLCMODE_HALI	boolean Node	1/@GV/PLCMODE_NALT	12@CV/PLCMODE_HALI	2BOOL202boolean
- @GV/PLCMODE_RUN	boolean Node	/@GV/PLCMODE_RUN	1?@CV/PLCMODE_RUN	??BOOL?0?boolean
@GV/PLCMODE_STOP	boolean Node	e1/@GV/PLCMODE_STOP	1?@CV/PLCMODE_STO	P??BOOL?0?boolean
Slave id: 192.168.0.2 Alias: Node2 Model: Generic ProConOS Controller	Container	Import Tag(s) - Network	e nodes to associate this dictionar	y and continue.
		Slave id Model	Alias	
		192.168.0.1 Generic ProConOS	Controlle Node1	
		192.168.0.2 Generic ProConOS	Controller Node2	
Dandy				
кеаду		For each selected node impor	t only symbols matching that node	e
			E	OK Cancel



Note: Aliasing tag names is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name. The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

Tag Import

The ProCoNos Ethernet driver support the Tag Import facility.

The symbol file can be exported by the controller programming software.

To import the tags from IEC project:

- 1. Select the Tags tab from ProjectView
- 2. Click the "Import tag" button
- 3. In the Tag Import window click the "OK" button to select the .csv file
- 4. Point to the "sr.csv" file from the IEC project

The Path is "ProjectFolder > C > Configuration > R > Resource"



5. After "sr.csv" file import, select the "DT" Directory for Data Types.

If the IEC project contains custom data types you have to select the "DT" folder from IEC Project to correctly import all the Tags.

Organize 🔻 🛛 New	folde	r			. 0
🔆 Favorites	^	Name	^	Date modified	Type
Nesktop		🝌 C		28/01/2015 15:01	File fol
\rm Downloads		🕖 DT 🚽		28/01/2015 15:01	File fol
🔠 Recent Places		📕 HW		13/11/2014 11:14	File fol
	=	🗼 LIB		28/01/2015 15:01	File fol
 Libraries Documents Music Pictures Videos 		🍌 POE		28/01/2015 15:01	File fo
🖳 Computer	÷	٠ [m		

6. Now all the variables are available as Dictionary in project. Select the desired variables and add to the tag list as shown in the figure below.

ProjectView 📮 🗙	1:Page1 Protocols Tags* x	
+ - 4 ∧ ∨	🕂 — 🐰 🖄 🔊 🔰 🚺 🏟 🛤 🔎 Search) Y f
Project2 Project properties	Data Type Tag name Gro	oups
 Project properties Pages 	ProConOS ETH:prot1 Model: Generic ProConOS Controller Container	
Unified	@GV/DATA_AX[1]/Commands/E unsignedShort @GV/DATA_AX[1]/Commands/E	
Y 🗁 Normal	@GV/DATA_AX[1]/Commands/F double @GV/DATA_AX[1]/Commands/F	
🍿 1:Page1	@GV/DATA_AX[1]/Commands/F double @GV/DATA_AX[1]/Commands/F	
C Dialog	@GV/DATA_AX[1]/Commands/S double @GV/DATA_AX[1]/Commands/S	
Townlates	@GV/DATA_AX[1]/Commands/S boolean @GV/DATA_AX[1]/Commands/S	
remplates	@GV/DATA_AX[1]/Commands/S unsignedInt @GV/DATA_AX[1]/Commands/S	1
 Configuration 		
Protocols		
Y 🖺 Tags		
📑 Indexed Tag Set		
Trends		
🚔 Reports		
🔔 Alarms	I	

Communication Status

The communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The status codes supported for this communication driver are:

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General Error	Error cannot be identified; should never be reported; contact technical support

Profibus DP

The Profibus DP communication driver has been designed to connect HMI products to a Profibus DP network as slave nodes. With the Profibus DP driver, the HMI simply exchanges Input and Output data with the Master. It is up to the Master to make sense of this data.

Connection to Profibus DP network requires the optional Profibus DP communication module. Verify the suitable version for your HMI model.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Please ensure that the latest driver is used in the application.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "Profibus DP" from the list of available protocols.

The driver configuration dialog is shown in figure.

Profibus DP		×
		ОК
Panel Node ID	1	Cancel
PLC Models default		
extended		

Element	Description
Panel Node ID	The Profibus node ID assigned to the HMI

Configuring the HMI as a Slave Node

The Profibus DP master must be configured to communicate with the slaves devices present in the network. To configure the Master System you will generally need a software package available from the manufacturer of the Master System. Before the master configuration software can recognize the the HMI device as slave, it must be included in the catalog of devices. For this purpose it is available a device description file in the standard GSD format. The device description file is EX9649AX.GSD. It must be installed following the instructions of the network configuration software you are using.

One of the fundamental steps of the configuration of a slave station in a Profibus DP system is the mapping of the slave's I/O buffers in the memory of the master.

The HMI panels support Input / Output buffer sizes of 8, 16 or 32 bytes and they expect that both the Input and the Output areas are configured to the same size, i.e. both 8 bytes, either 16 bytes or both 32 bytes. The HMI panels will automatically detect the buffer size used by the master.

The feature generally referred to as Response Monitoring should always be disabled in the master for the HMI panel slaves.

Tag Editor Settings

Profibus DP				×
Profibus DP				
Memory type	Offset	SubIndex		
Input buffer Input buffer	 ✓ 0 	• • •		
Output buffer I/O buffer length Internal buffer Node override	Arraysize 0	Conversion	+/-	
	OK	Cancel	Apply	Help

- Studio allows you to access the HMI panel "Output Buffer", the area containing data sent from the PLC, as well as the HMI panel "Input Buffer", the area containing data to be sent to the PLC. The data in the Output Buffer is read only, while the data in the Input Buffer is read write. The Address Offset range (in bytes) for these 2 types is from 0 - 31. It should be borne in mind, however, that that Input / Output buffer range configured in the PLC for the panel can be either in the range 0 - 7, 0 - 15 or 0 - 31.
- 2. In addition to the Input Buffer and the Output Buffer Designer also allows you to access the "Internal Work Buffer" data type. This buffer is purely an internal buffer in the panel. The panel sets aside 256 bytes for this buffer. The data in this buffer is neither read from nor written to the PLC. It is purely a work area.

Tag Conversion

Conversion to be applied to the tag.

Conversion		
inv,swap2	Allowed	 Configured
	BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ ABCD->CDAB
		 Cancel OK

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	inv: Invert all the bits of the tag.
	Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
Negate	neg : Set the opposite of tag value.
	<i>Example:</i> 25.36 → -25.36
AB -> BA	swapnibbles: Swap nibbles in a byte.
	<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
ABCD -> CDAB	swap2: Swap bytes in a word.
	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
ABCDEFGH ->	swap4: Swap bytes in a double word.
GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
ABCNOP ->	swap8: Swap bytes in a long word.
OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.10000000110 0001110010111011001000101101000011100101
BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
	Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Special Data Types

The Profibus DP communication driver provides one special data type called "Node Override".

The Node Override ID allows changing at runtime the value of Panel Node ID. This memory type is an unsigned byte.

The Node Override ID is initialized to the value defined as Panel Node ID in the project at programming time.

The communication with the master is described in the table.

Node Override ID value	Behavior
0	The communication with the master is stopped
1 to 255	If Node Override ID has a value different from 0, it is interpreted as the new node ID for the slave device.



Note: the Node Override values assigned at runtime are retained through power cycles

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General error	Error cannot be identified; should never be reported; contact technical support

Profibus DP S7

The Profibus DP S7 communication driver has been designed to connect the HMI products to a Profibus DP network as slave nodes. This communication driver has been specially created to offer optimal data exchange features for Profibus DP networks where the bus master is a Siemens Simatic S7 PLC.

This Technical Note gives the technical details for a successful connection.

A Profibus DP network can contain multiple nodes. A node in a Profibus DP network can be either a Master or a Slave. The Masters in the network have a group of Slaves assigned to them. A Master is able to exchange data with the Slaves that are under its control.

The HMI panel is always a Slave device in a Profibus DP network and it is only able to exchange data with a single Master PLC. The HMI has a complex communication profile, as it needs to access data in the Master PLC memory. This communication profile is not something normally available for Profibus DP Slave devices. To enable the HMI to communicate under this profile, a set of special function blocks must be added to the PLC program in the Master PLC. These special function blocks are required by the PLC to process the requests from the HMI. These special function blocks use a Data Block, called the Comm DB, within the Master PLC to store configuration information. This approach has the advantage that it offers to the HMI slave device full access to the data in the PLC, as if the HMI device was directly connected to the programming port of the PLC.



Note: Connection to Profibus DP network requires the optional Profibus communication module. Verify the suitable version for your HMI model.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Please ensure that the latest communication driver is used in the application.

Protocol Editor Settings

Add [+] a driver in the Protocol Editor and select the protocol called "Profibus DP S7" from the list of available protocols.

ous DP S7			ОК	
Alias			Cancel]
Panel Node ID	1	<u> </u>		
PLC Models		- 142	-	
S7-3xx				
\$7-313/314				=
S7-315				-
5/-31/				
57-310				+

The driver configuration dialog is shown in figure.

Element	Description
Panel Node ID	The Profibus node ID assigned to the HMI.
PLC Models	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.

Configuring HMI as a slave station with STEP7

The Master PLC must be configured to communicate with Profibus DP slaves. You can do this with the STEP 7 programming software. This package configures the Profibus DP network attached to the Master PLC (or to the CP communication processor) so that it exchanges data with the specified Slaves. With this package you can select different types of Slaves such as HMI, distributed I/O, drives, etc.



Note: Step7 versions 5.5 SP1 has been used to create the examples included in this Tech Note. Using another version of the Step7 software package may require changes to the procedures described in this document.

Adding the DDB file to your system

A Profibus DP Slave type file (GSD) is available for the Profibus DP configuration. The filename is EX9649AX.GSD; this file contains the description of the HMI devices as Profibus DP Slaves.

To include the file in the system, follow the procedure:

- 1. Select your station
- 2. Double click on "Hardware" to open "HW Config" editor
- 3. From menu Options select "Install GSD File..." and follow the wizard



This will enable STEP 7 to recognize the HMI panels as an element of the class 'Additional Field Devices'.

Network configuration

The basic steps of the Profibus DP configuration are described below.

Create a new Step7 project or open an existing project.

Configure the system (Hardware Configuration) using components from the Hardware Catalog

🕅 Station Edit Insert PLC View Options Window Help	_ 7	×
D 🍃 🖫 🖳 🎆 🎒 🖪 🕄 🏜 🏜 👔 🗖 😫 💦		
<u>^</u>	·	x
III	Eind:	ni
	Profile: Standard	•
	🕒 💼 CPU 314 IFM	<u>~</u>
	🕀 🧰 CPU 314C-2 DP	
	⊕	
	⊕	
	6ES7 315-2AF00-0AB0	
	EES7 315-24F07-0480	
	⊕ 6ES7 315-24E03-04B0 ⊕	
~	6ES7 315-2AF82-QAB0	~
	6ES7 315-2AF02-0AB0 3	٤d
	Work memory 64 KB; 0.3 ms/1000 instructions; MPI + DP connections; for multi-tier configuration up to 32 modules	
Press F1 to get Help.	Chg	1

Create and configure the Profibus DP network.

r system		X
s Group Assignment		
DP master system		
mybus		_
1		
SINEC L2 Network (1)		
Properties		
	Cancel	Help
	r system s Group Assignment DP master system mybus 1 v SINEC L2 Network (1) Properties	r system s Group Assignment DP master system mybus 1 SINEC L2 Network (1) Properties Cancel

Properties - PROF	ibus 🛛 🔀
General Network	Settings
Name: S7 subnet ID:	SINEC L2 Network (1) 4711 • 0002
Project path:	P315P2-2\SIMATIC S7-300 Station 1\CPU 314C-2 DP(1)\DP
Storage location of the project:	C:\Program Files\Siemens\Step7\S7Proj\P315P2_2
Author:	
Date created: Last modified:	05/20/1997 11:18:42 AM 03/14/2013 11:49:50 AM
Comment:	
OK	CancelHelp

Properties - PROFIBUS			X
General Network Settings			1
Highest PROFIBUS Address:	126 💌	Change	Options
Transmission Rate:	500 Kbps 1.5 Mbps 3 Mbps 6 Mbps 12 Mbps		
Profile:	DP Standard User-Defined		Bus Parameters
ОК			Cancel Help

Station Edit Insert PLC View Options Window Help	
D 😂 🗣 🖳 🥌 🖶 🖻 🏜 🏜 📳 🗖 😤 💦	
🕮 SIMATIC S7-300 Station 1 (Configuration) P315P2-2	
	Profile: Standard
2 I CPU 314C-2 I X2 DP 22 D/24/D016 23 A/5/A02 24 Count	CPU 314C-2 PtP CPU 315 CPU 315 CPU 315-2 DP 6ES7 315-2AF00-0AB0 6ES7 315-2AF01-0AB0
SINEC L2 Network (1): mybus (1)	
	⊕- 6657 315-24610-0480 ⊕- 6657 315-24H14-0480 ⊕- 6657 315-24H14-0480 ⊕- 70H 315-2 PN/DP
	6657 315-24F02-0A80 Work memory 64 KB; 0.3 ms/1000 instructions; MPI + DP connections; for multi-tier configuration up to 32 modules
Press F1 to get Help.	Chg

The Operator Panels will then be available for selection in the Hardware Catalog as shown in the figure below. Note that the DDB Files must have been updated as described in chapter 1.1.



Select the Device from Hardware catalog and Drag & Drop it to the Bus line, once added assign the Address properly

Properties - PROFIBUS interface UniOP MMI	
General Parameters	
Address: 3	
Transmission rate: 12 Mbps	
Subnet:	
SINEC L2 Network (1) 12 Mbps	New
	Properties
	Delete
OK Ca	ancel Help

Once the HMI devices have been included in the Profibus DP network configuration, you will have to open the slave configuration and enter the required parameters. 2 or 4 blocks must be configured in the DP image area for the device depending on the size of the buffer (16 or 32 bytes) which has been selected in the previous step. The HMI panels can work with a DP image size of 16 bytes or 32 bytes. Using 32 bytes will offer improved communication performance at the expense of an increased memory usage in the process image area

Station Edit Insert PLC	View Options Window Help					
0 🗃 🔓 🖩 🖏 🎒	🖻 🔁 🏜 🏦 📳 🗖 🖣	R.				
🔍 SIMATIC S7-300 Stati	ion 1 (Configuration) P315F	2-2			L ·	
				Eind:	uniop	nt ni
				Profile:	Standard	•
2 CPU 31	4C-2 C	iork (1): mybus	<u>(1)</u>		ROFIBUS DP	~
2.2 DF 2.2 DI24/DC	076 1 📷 (3) UniOl	PN		· · · ·	- Switching Devices	=
2.3 🚺 A/5/AO2		all		+	- <u></u> 1/0	
2.4 Count		9		-	- MMI - Turion MMI	
25 Position				2 Universal module 2 32 byte DIN/DOUT 16 byte DIN/DOUT		
(3) UniOP MMI	3				🦾 📳 8 byte DIN/DUUT	
Slot DPID	Order Number / Designation	Address	Q Address	÷	Compatible PROFIBUS DP Slaves	
1 55	32 byte DIN/DOUT	07	07	ļ	CiR-Object	
2 55	> 32 byte DIN/DOUT	815	815	E	Closed-Loop Controller	v
3 55	> 32 byte DIN/DOLIT	1623	1623		L ontidured Stations	T T
4 55	-> 32 byte DIN/DUUT	2431	2437			-
7				J		
Insertion possible						Chg //

Configure the blocks in the DP image area. If buffer size of 16 bytes is selected, unused blocks are automatically set to 'Empty slot'.
roperties	- DP slave						
Address / ID							
I/O type:		Out- input		-		Di	ect Entry
_ Output -							
	Address:	Length:	Unit:		Consistent over:		
Start:	8	8 📫	Byte	-	Unit	-	
End:	15						
Process	image:	OB1 PI		Ψ.			
Input							
	Address:	Length:	Unit:		Consistent over:	_	
Start:	8	8 🛨	Byte	Ψ.	Unit	7	
End:	15						
Process	image:	OB1 PI		Ψ.			
Manufactu	rer-snecific o	lata:					
(Maximum)	14 hutes hex	adecimal sena	arated by co	mma or bla	ank snacel		
0.12							
ОК	1				C	ancel	Help
	_						

roperties - DP s	lave	
Address / ID		
I/O type:	Empty slot	Direct Entry
Manufacturer-spe	cific data:	
(Maximum 14 byte	s hexadecimal, separated by comma or blank spa	ce)
UK		Uancel Help

🕅 Station Edit Insert PLC View Options Window Help			- 8 ×
D 😅 💱 📓 🙀 🎒 💼 💼 🛍 🏙 🖺 🗊 💳 🞇 📢			
SINEC L2 Network (1) mybus (1) SINEC L2 Network (1) mybus (1) SINEC L2 Network (1) mybus (1) (3) UniOP N (4) UniOP N DP-NORM DP-NORM	Eind: Profile	uniop Standard PROFIBUS DP Additional Field Devices C Switching Devices UN MMI UniOP MMI UniOP MMI C UniOP MMI C Device Module C Star Double	■ = = = = = = = = = = = = = = = = = = =
SIMATIC S7-300 Station 1 Slot Designation D UR CINECC 2 Natural (1) and (1)			
Press F1 to get Help.		DP V0 slaves	► <u></u>

The configuration procedure must be repeated for all the HMI devices to be included as slaves in the Profibus DP network. Finally the network configuration will have to be transferred to the master PLC.

Using the function blocks in the master PLC

To make possible for the HMI device to access all the data in the Master PLC, some support from the PLC program is required. It is accomplished by adding to the user PLC application some special program modules. Samples of these program modules required to support Profibus DP communication are available.

The core functionality is provided by one special function block must be added to the user's program. The complete support includes also 2 Data Blocks.

The Function Block and the other blocks are available in the form of ready-to-run sample projects. Function and Data Blocks may be extracted from the sample projects for integration into the user's project.

Apart from adding the special blocks to the PLC program you also need to cyclically call FB1. You can do this by adding a call to FB1 in OB1.

It is important that FB1 is called cyclically; you should not call it only one time, as the function block only processes the requests from the slave devices when it is called.

The HMI devices will not be able to communicate with the Master PLC if it is in STOP mode as the special function block will not be called.



Note: if you have multiple HMI devices connected to the Master PLC you do NOT need to call FB1 once for each panel. One call to the FB1 for every cycle of the PLC program is sufficient to process all the HMI slave devices in the Profibus DP network attached to the Master PLC.

Sample PLC programs

Sample PLC programs are available on our website in Software section.

Click on Profibus DP S7 example projects to start the download, as shown in picture below.

S7DP Profibus DP S7 example projects Siemens Step7 example projects for Profibus DP S7

Creating the comm data block

The Comm DB (Communication Data Block) is used to provide the program modules supporting Profibus DP communication with information on:

- the number of HMI devices configured as Profibus DP slaves and
- the addresses for the Input and Output data of the slave devices in the Master PLCs memory.

The Comm DB has 2 distinct parts; the first part contains information about the configuration of the Profibus DP network of the PLC while the second part contains information about the various HMI devices that are connected to this port. Basically this information is a duplication of the data that you enter in the Profibus Master with Step7.

The Profibus DP Port part is placed in the first 14 bytes of the Comm DB and has the following format:

DBB0	Number of Panels
DBB1	Frame Length
DBB2	Data Type for Input Buffer
DBB3	Data Type for Output Buffer
DBW4	DB Number (Input Buffer)
DBW6	Input Area Base (Input Buffer)
DBW8	DB Number (Output Buffer)
DBW10	Output Area Base (Output Buffer)
DBB12	Sequence Type
DBB13	Reserved for Internal Use

Number of Panels	total number of HMI panels that have to communicate with the Master PLC.					
Frame Length	size of the Profibus buffers used to communicate with the Master. Two buffer sizes are supported: 16 bytes and 32 bytes. Enter the appropriate number in this location. Input and Output buffers always have the same size					
Data Type	type of PLC data where the Profibus DP input buffer for the panels is located.					
Buffer	Value Data Type					
	0	DB				
	4	1				
	The Input buffer contains the information received by the Master from the slave.					
Data Type for Output Buffer	type of PLC data where the Profibus	DP output buffer for the panels is located.				

	Value	Data Type					
	0	DB					
	5	Q					
	The Output buffer contains the inforr	nation written by the Master to be sent to a Slave.					
DB Number (Input Buffer)	if the location specified for the Input I	if the location specified for the Input Buffer is a DB, enter here the DB number					
Input Area Base (Input Buffer)	offset in the Input Buffer where the data for the panels starts.						
DB Number (Output Buffer)	if the location specified for the Outpu	f the location specified for the Output Buffer is a DB, enter here the DB number.					
Output Area Base (Output Buffer)	offset in the Output Buffer where the data for the panels starts.						
Sequence Type	specifies how you want to handle the 1. If you set this item to 0 then the fur panels before returning. If you set thi request from only a single panel befor next panel on the subsequent call. T requests from the HMI panels will be program will be longer. If the increas problems for your application you ca	e case of having Number of Panels set to greater than nction block will process the requests from all the HMI is item to 1 then the function block will process the pre returning, it will then process the request for the his means that if Sequence Type is set to 0 the processed faster but the execution time of the PLC ed execution time of the PLC program causes in set Sequence Type to 1.					
Reserved For Internal Use	is actually used to keep track of whic Type is set to 1	h panel was processed last. This is used if Sequence					



Note: in this chapter the terms 'Input' and 'Output' are referred to the Master PLC and not to the slaves. The information entered in this section must be the same entered in the Profibus DP network configuration.

Following on from the header data comes the HMI panel data. The number of HMI panels connected to this port is specified by 'Number of Panels'. Each HMI panel is assigned 8 bytes in the Comm DB.



Note: each panel included in the Profibus DP network must have its descriptor in the Comm Data Block. All descriptors have to be placed in consecutive memory locations.

The format of the data block for the individual slave devices has the following format:

DBW14	Input Area Offset
DBW16	Output Area Offset
DBB18	Reserved
DBB19	Reserved
DBB20	Error Code for Last Request
DBB21	Last Job Number

Input Area Offset	this number is added to the Input Area Base (in the header) to obtain the address where this input data for this panel starts					
Output Area Offset	this number is added to the Output Area Base (in the header) to obtain the address where this output data for this panel starts					
Error Code for Last	Error Code for the last communication request for this panel. The error codes have the following meaning:					
Request	Error Code	Meaning				
	0 No Request Received					
	1 Request Processed OK					
	2 Request Rejected					
	You do not need to set this element. The function blocks will actually write to this element to give you an indication of the error status of the communication. This field is therefore just for information.					
Last Job Number	Job Number for the last comm makes a request it includes th incremented for every new rea therefore just for information.	nunication request for this panel. Every time the HMI panel e Job Number in the request to the PLC. This Job Number is quest. You do not need to set this element. This field is				

Example

As an example, imagine we have 2 HMI devices attached to a Master PLC that uses I/O addressing and 16 bytes Frame Length. The Input address for the first panel is set to IB16 and the Output address to QB16. The Input address for the second panel is set to IB32 and the Output address to QB32. The Comm DB would take the following form:

🕒 File Edit	Insert PLC Debug View Op	tions Window Help		_ 8 ×
🗋 🖻 🔓	📙 🎒 👗 🖻 🛍 🗠 🕫	- 071 🏜 🔽 ºs	ଙ୍କ !≪ ≫! 🗖] ⊡ № ?
Address	Name	Туре	Initial value	Comment
0.0		STRUCT		
+0.0	NR_P	BYTE	B#16#2	Number of panels
+1.0	FR_LEN	BYTE	B#16#10	Length of frame (16 or 32 bytes)
+2.0	IN_AM	BYTE	B#16#4	Input area : I (O=DB 4=Input)
+3.0	OU_AM	BYTE	B#16#5	Output area : Q (O=DB 5=Output)
+4.0	IN_DBN	WORD	W#16#0	Only with DB input
+6.0	IN_BASOF	WORD	W#16#0	Input area Base Offset
+8.0	OU_DBN	WORD	W#16#0	Only with DB output
+10.0	OU_BASOF	WORD	W#16#0	Output area Base Offset
+12.0	SEQ_TYPE	BYTE	B#16#0	Sequence type : Scan (0=SCAN 1=ONE SHOT)
+13.0	res13	BYTE	B#16#0	Reserved
+14.0	IN1_OFSADJ	WORD	W#16#10	Offset input area panel l
+16.0	OU1_OFSADJ	WORD	W#16#10	Offset output area panel l
+18.0	res18	WORD	W#16#0	Reserved
+20.0	ERRCOD1	BYTE	B#16#0	Error code for Panel l
+21.0	LASTJOBNUM1	BYTE	B#16#0	Last job number for Panel 2
+22.0	IN2_OFSADJ	WORD	W#16#20	Offset input area panel 2
+24.0	OU2_OFSADJ	WORD	W#16#20	Offset output area panel 2
+26.0	res26	WORD	W#16#0	Reserved
+28.0	ERRCOD2	BYTE	B#16#0	Error code for Panel 2
+29.0	LASTJOBNUM2	BYTE	B#16#0	Last job number for Panel 2
=30.0		END_STRUCT		
<				

When you download a data block to the PLC or when you modify any values in the data block, you will have to make sure that the modified values are also the current values into the PLC memory. To do this you should change the viewing mode of the data block in the Step7 software from "Declaration View" to "Data View" as shown in the next figure.

When you are in Data View mode, the values in the column 'Actual Value' must match the values on the column 'Initial Value'. If there are some differences you have to correct the wrong value on the 'Actual Value' column and download again the Data Block to the PLC. The 'Actual Value' column displays at any time the actual PLC data values.

🗊 File Edit	Insert PLC Debug	View 0	ptions Wind	low Help			_ @ ×
🗋 🚔 🔓	- 🔚 🎒 X 🖻 🕻	Overv	views	Ctrl+K	!«»	>! 🗖	■ №
Address	Name	🗸 Detail	ls		++-1 -	walue l	Commont
		PLC R	legister			varue	Commente
+0.0	NP P	• LAD		Ctrl+1	6#2		Number of namels
+1.0	FR LEN	STL		Ctrl+2	6#10		Length of frame (16 or 32 bytes)
+2.0	IN AM	- FBD		Ctrl+3			Input area : I (0=DB 4=Input)
+3.0	OU AM	Data	View	Ctrl+4	.6#5		Output area : Q (O=DB 5=Output)
+4.0	IN_DBN	• Declar	ration View	Ctrl+5	.6#0		Only with DB input
+6.0	IN_BASOF	Displa	ay with		▶ .6#0		Input area Base Offset
+8.0	OU_DBN		In	Ctrl+Num+	.6#0		Only with DB output
+10.0	OU_BASOF	Zoom	Out	Ctrl+Num-	.6#0		Output area Base Offset
+12.0	SEQ_TYPE	Zoom	Factor		.6#0		Sequence type : Scan (0=SCAN 1=ONE SHOT)
+13.0	res13		ər		.6#0		Reserved
+14.0	IN1_OFSADJ	Break	ooint Bar		.6#10		Offset input area panel l
+16.0	OU1_OFSADJ	🗸 Statu:	s Bar		.6#10		Offset output area panel l
+18.0	res18	Colum	an cuidhe		.6#0		Reserved
+20.0	ERRCOD1				_ + .6#0		Error code for Panel 1
+21.0	LASTJOBNUM1	Displa	ay Columns	F11	.6#0		Last job number for Panel 2
+22.0	IN2_OFSADJ	Updat	te	F5	.6#20		Offset input area panel 2
+24.0	OU2_OFSADJ		WORD		w#1 6#20		Offset output area panel 2
+26.0	res26		WORD		W#16#0		Reserved
+28.0	ERRCOD2		BYTE		B#16#0		Error code for Panel 2
+29.0	LASTJOBNUM2		BYTE		B#16#0		Last job number for Panel 2
=30.0			END_STRU	ICT			
<							

🕞 File Edit	t Insert PLC Debu	g View Options Wind	dow Help		_ & ×
🗋 🚔 🔓	- 🖬 😂 👗 🖻	💼 တ က 🕅 🏜	1 🔁 📲 🕷	!« »! 🗖 🖪	▶?
Address	Name	Туре	Initial value	Actual value	Comment
0.0	NR_P	BYTE	B#16#2	B#16#2	Number of panels
1.0	FR_LEN	BYTE	B#16#10	B#16#10	Length of frame (16 or 32 bytes)
2.0	IN_AM	BYTE	B#16#4	B#16#4	Input area : I (0=DB 4=Input)
3.0	OU_AM	BYTE	B#16#5	B#16#5	Output area : Q (O=DB 5=Output)
4.0	IN_DBN	WORD	W#16#0	W#16#0	Only with DB input
6.0	IN_BASOF	WORD	W#16#0	W#16#0	Input area Base Offset
8.0	OU_DBN	WORD	W#16#0	W#16#0	Only with DB output
10.0	OU_BASOF	WORD	W#16#0	W#16#0	Output area Base Offset
12.0	SEQ_TYPE	BYTE	B#16#0	B#16#0	Sequence type : Scan (0=SCAN 1=ONE SHO
13.0	res13	BYTE	B#16#0	B#16#0	Reserved
14.0	IN1_OFSADJ	WORD	W#16#10	W#16#10	Offset input area panel l
16.0	OU1_OFSADJ	WORD	W#16#10	W#16#10	Offset output area panel l
18.0	res18	WORD	W#16#0	W#16#0	Reserved
20.0	ERRCOD1	BYTE	B#16#0	B#16#0	Error code for Panel 1
21.0	LASTJOBNUM1	BYTE	B#16#0	B#16#0	Last job number for Panel 2
22.0	IN2_OFSADJ	WORD	W#16#20	W#16#20	Offset input area panel 2
24.0	OU2_OFSADJ	WORD	W#16#20	W#16#20	Offset output area panel 2
26.0	res26	WORD	W#16#0	W#16#0	Reserved
28.0	ERRCOD2	BYTE	B#16#0	B#16#0	Error code for Panel 2
29.0	LASTJOBNUM2	BYTE	B#16#0	B#16#0	Last job number for Panel 2
<					>

Direct Import of TIA Portal project

It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

Data Blocks must be set as Not optimized:

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

General Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Date block units protected in the douise
Attributes	Optimized block access

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Export using TIA Portal v13, v14 or newer

Exporting Program blocks

Π.

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

General	
Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
Attributes	Cptimized block access



Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

4. Right-click on the Data Block and choose Generate source from blocks:



5. Save the file as DBxxx.db, where xxx=number of DB.

\$7-1200	Name		Data type	Offset	Start	value	Retain	Accessible f
🚰 Add new device 🥤								
h Devices & networks	Save As							—X
🛅 PLC_1 [CPU 1212C DC/DC/DC] 🧾	COO⊽ L « Use	r 🕨 My Documents	Automation	•	4+	Search Au	tomation	Q
🔢 Device configuration								
😡 Online & diagnostics	Organize 🔻 Nev	v folder					:== •	• 0
🕶 🚘 Program blocks	Recent Places	A Nama	*		Date	modified	Tuno	
📑 Add new block	incent Places	Name			Date	moainea	туре	
🚘 Main [OB1]	-		N	n items match s	/our se	arch.		
🥃 DB2 [DB2]	Cibraries							
🥃 DB4 [DB4]	Documents							
🕨 🚂 Technology objects	a) Music							
External source files	📔 Pictures							
🕶 🚂 PLC tags	🛃 Videos	E						
🍇 Show all tags								
📑 Add new tag table	🝓 Homegroup							
🍯 Default tag table [29]								
🍓 Tag table_1 [0]	Computer							
🔻 [PLC data types	1 computer							
📑 Add new data type	• •••	+ +						- F
d UDT1		DB2 .//						
Watch and force tables	File name:	DB2.db						
🕨 🙀 Online backups	Save as type:	DB files(*.db)						-
🕨 🔄 Traces								
🕨 🏢 Device proxy data	A Hide Folders					Save	Ca	incel
📴 Program info	- Hide Folders							
🔄 Text lists								

Exporting PLC tags

An Excel file refers to PLC tags.

- 1. Double-click **Show all tags**: the tag table is displayed.
- 2. Click the **Export** button and browse for path file.
- 3. Define file name.



4. Click **Save** to confirm.

	JTRACOT)	
	•	(
Date modified	Туре	
No items match your search.		
m		
	No items match your search.	▲ Date modified Type No items match your search.

5. Click **OK** to export.

xport to Excel X
Path of export file:
C:\Users\User\Documents\Automation\PLCTags.xlsx
Elements to be exported:
🗹 Tags
Constants
6
OK Cancel

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

🍯 Default tag table [31 🖳 Tag table_1 [0]		
PLC data types		
E UDT2	Open	
Watch and force tables	🔏 Cut	Ctrl+X
🕨 📴 Online backups	E Copy	Ctrl+C
🕨 🔄 Traces	E Paste	Ctrl+V
🕨 🚂 Device proxy data	Copy as text	
Program info	🗙 Delete	Del
Text lists	Rename	F2
Local modules	Compile	•
 Common usua Documentation settings 	💋 Go online	Ctrl+K
Cocumentation settings	🔊 Go offline	Ctrl+M
M Online access	📑 Generate source from blocks	
🦉 Card Reader/USB memory	Cross-reference information	Shift+F11
	Cross-references	F11
	🔢 Call structure	
	Assignment list	
	昌 Print	Ctrl+P
	Print preview	
	國 Properties	Alt+Enter

In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .UDT file that contains all the PLC data types defined.

Add new data type Image: UDT1 Image: UDT2 Image: UDT3 Image: UDT3 Image: UDT4 Image: Watch and force table Image: Watch and force table
Image: Second
Image: Control of the sector of the secto
Image: Weatch and force table X Cut Ctrl+X Image: Weatch and force table Paste Ctrl+V
UDT4 III Copy Ctrl+C
Watch and force table Paste Ctrl+V
🕨 🙀 Online backups 🛛 🗙 Delete 🛛 🛛 Del
F2 Rename F2
Device proxy data Compile
🔤 Program info 🛛 💋 Go online Ctrl+K
🔄 Text lists 🛛 📝 Go offline Ctrl+M
In Local modules
Generate Source from Discuss
Documentation settings Cross-reference information Shift+F11
Cross-references F11
Online access
📴 Card Reader/USB memory 🔠 Assignment list
📑 Print Ctrl+P
🔒 Print preview
Properties Alt+Enter

In the next step, give a name to the .UDT file and choose the path to where to save the file.

EX05

Save As			X
Us 🖉 🖓 🗸	er MyDocuments Automation	👻 🍫 Search Auton	ration 🔎
Organize 🔻 Ne	w folder		:= • 🔞
📃 Recent Places	^ Name	Date modified	Туре
 ➢ Libraries ➢ Documents ➢ Music ➢ Pictures ➢ Videos ↔ Homegroup 	E	No items match your search.	
💻 Computer			
• • •	▼ <	III	•
File name:	myUDTfile		-
Save as type:	UDT files(*.udt)		•
) Hide Folders		Save	Cancel

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check Tag Import chapter for more details.

Export using TIA Portal v10, v11, v12

Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

General	
Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
Attributes	Optimized block access

1

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Pr	oject Edit View Insert Online	Optic	ns Tools Window Help
	🗳 🎦 🔒 Save project ا 🐰 🗓	i ×	(🔊 ± (색 ± 👔 🗟 🛯 👔 🖳 🖉 🗛 🍠 Go online 🧬 Go offline 👬
	Project tree [Tia_project_Lute_IVTV/ PLC_1 [CPU 1211C DC/DC/DC] > PLC tags
	Devices		
	B O O 🖻	ų	20 H C

4. Build the project to make sure TIA Portal calculates the tags offset.

Project tree		Tia_p	roject_Luca (V1	1] → PLC_1 [C	PU 1211C I	DADADC] 🕨	Program t
Devices							
	B	🥩 :	🖗 💺 🐺	R. R. 🚍 🔢			
		DB	2				
晶 Devices & networks	^	C	Name	Dat	a type	Offset	Start value
▼ 1 PLC_1 [CPU 1211C DC/DC/		1 🕣	👻 Static				
Device configuration		2 📶	tag1	🛒 Insert	row		
😯 Online & diagnostics		3 📶	tag2	Add ro	W		
🗢 🔜 Program blocks		4 📶	tag3	💥 Cut		c	trl+X)
Add new block		5 📶	 Static_1 	🚺 📋 Сору	N	C	trl+C
		6 🕣	 Static 	1[0] 💼 Paste	4	C	trl+V
6 = DB2 [DB2]		7 🕣	 Static 	1[1] 🗙 Delete			Del
B4 [DB4]		8 📶	Static_2	Renam	ne		F2
🕨 🔜 Techen							
🕨 🕞 External 🚺 Untitled - N	lotepad						
▼ → PLCtag: File Edit Forma	at View	Help		Cross-	reference ink	o ma eldi o n — Sih ini	
🗞 Show Sta	tic	_					
Addr tag	1 E 2 (Bool Thar	0.0 fa	ise False ' False	True		
📲 Defai 🛛 🛛 tag	3 E	3yte	2.0 16	⊭0 False	True		
Tagt Sta	tic_1	-	Array [0	L] of Bool	4.0	F	alse Tr
▼ I PLC data Sta	tic_1	[1]	Bool 0.1	L false	False	True	

- 5. Double-click on a DB name.
- 6. Expand the view of program block selected.
- 7. Select all rows.
- 8. Copy and paste into any text editor.
- 9. Save the file as DBxxx.tia, where xxx=number of DB.



Note: Make sure you use the **Save As** function or the file will be named DB2.tia.txt and will not be visible from the importer.

10. Repeat from step 5 for all program blocks.



Note: Make sure that only the following columns are shown in DB editor before copying all data in the txt file



Exporting PLC tags

An Excel file refers to PLC tags.

1. Double-click **Show all tags**: the tag table is displayed.



- 2. Click the **Export** button and browse for path file.
- 3. Define file name.
- 4. Click Save to confirm.



5. Click OK to export.

Path of export file:			
C:\Documents and S	ettings\admin\Des	top\PLC tags.xlsx	
Elements to be expo	orted:		
🛃 Tags			
Constants			

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

Devices			
		Open	
		Edit type	
🗖 Data blo	Ж	Cut	Ctrl+X
Data_blo		Сору	Ctrl+C
Data_blo	Ē	Paste	Ctrl+V
Data_blo		Copy as text	
📒 Data_blo	×	Delete	Del
🕨 🔚 System b		Rename	F2
👻 🙀 Technology (Compile	
📑 Add new	1	Go online	Ctrl∓K
🕨 🔚 External sou	3	Go offline	Ctrl+M
▶ 🛃 PLC tags ▼ 📴 PLC data tvp	-	Generate source from blocks	
Add new		Cross-reference information	Shift+F11
₿ A	×	Cross-references	F11
₿ B	Ē	Call structure	
t≣ C		Assignment list	
🖪 D		Print	Ctrl+P
🚯 E	4	Print preview	
<u>₿</u> F	Q	Properties	Alt+Enter
围 My UDT			
🕨 🕨 🔜 Watch and fo	rce	tables	
Traces			
🐏 Program info			
🖂 PLC alarms			
🛅 Text lists			
International Internation International Internation	20		

In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .SCL file that contains all the PLC data types defined.

			00	11	
🕶 🛅 Project_Test_User	r_da	ata_Type	^	Name	
📑 📑 Add new devip	e.		1	_ <add< th=""><th>new></th></add<>	new>
📥 Devices & net		Open			
🔹 🛨 🛅 PLC_1 [CPU 1:	5 ³	Terminate connection to type	2		
📑 Device cor		Edit type			
😵 Online & d	Ж	Cut	Ctrl+X	-	
🕨 🛃 Program bl	ň	Сору	Ctrl+C		
🕨 🕨 🙀 Technology	ĥ	Paste	Ctrl+V		
🕨 🕨 📷 External sc	×	Delete	Del	-	
🕨 🍃 PLC tags		Rename	F2		
🔻 💽 PLC data ty		Rename	12	-	
📑 Add nev		Compile	•		
🚯 UDT1	ø	Go online	Ctrl+K		
E UDT2	P	Go offline	Ctrl+M		
	Þ	Generate source from blocks			
民 UDT4		Cross-reference information	Shift+F11		
🕨 🔜 Watch and	×	Cross-references	F11		
🍱 Program in		Call structure			1
🛓 Text lists		Assignment list		ral 🚺	Cross-references
🕨 🕨 🛅 Local mod		Print	Ctrl+P		

In the next step, give a name to the .SCL file and choose the path to where to save the file.

Save As							? 🔀
Save in:	😂 exported			~	0 1	• 🛄 💙	
My Recent Documents							
Desktop							
My Documents							
My Computer							
S	File name:	my_SCL_wit	h_all_UDT			~	Save
My Network	Save as type:	scl files (*.sc	Ŋ.			~	Cancel

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check Tag Import chapter for more details.

Export using STEP7

The Simatic S7 ETH Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.

File Edit Insert PLC View Options Window Help Image: State in the stat	SIMATIC Manager - S7_ProtocolTest	
Image: Structure Image: Structure <td< th=""><th>File Edit Insert PLC View Options Window Help</th><th></th></td<>	File Edit Insert PLC View Options Window Help	
S7_ProtocolTest Sources SIMATIC 300 Station Sources CPU315:2DP(1) Sources Sources Blocks	🗋 🗅 😅 🔡 🛲 🕹 🛍 🛍 🕼 😰 🐾 🕒 📴 🧱 🗰 🔂 🔍 < No Filter >	🔄 🏹 🞇 😂 🖷 🗖 🚺 📢
Sources	S7_ProtocolTest C:\Program Files\Siemens\Step7\s7proj\S7_Pro-1	
	S7_ProtocolTest SIMATIC 300 Station CPU315-20P(1) Sources Blocks	

- 1. From the Symbol Table menu in the Symbol Editor choose Export.
- 2. Assign a name and save the symbol table as ASCII file.

Symbol Table Edit Insert View Options Window Help							
🚰 🖬 🖓 🐰 🖻 💼 🗠 🖂 🛛 All Symbols 📃 🔽 🕅 💦							
	Status	Symbol 🛆		Address	Data type	Comment	
1		A234567890123456789012	234	MVV 65524	WORD		
2		BasicDataType_UDT		UDT 1	UDT 1		
3		bit_M_2_0		M 2.0	BOOL		
4		bit_M_32770_0		M 32770.0	BOOL		
5		bit_M_65522_0		M OFFOR	10001		
6		bit_M_7_5	Export			? 🔀	
7		byte_MB_3					
8		byte_MB_32771	Save in: 🖾 ProtTes	st		💌 🗮 🖿 📖 •	
9		byte_MB_65523	DrotTest Symbols	01.555			
10		char_MB_7	DvokTook Symbols	02.555			
11		ComplexDataTypes	Procrest_symbols_	juz.ast			
12		Cycle Execution					
13		date_MVV_24641					
14		dint_MD_32773					
15		dint_MD_5					
16		dint_MD_65525					
17		dword_MD_0					
18		dword_MD_32768	File name:	at Combala 01	1	C	
19		dword_MD_65520	rile name.	st_symbols_uz	.asc	Save	
20		int_MVV_32774	Save as tupe: ACCII	Format (* ACC)		_ Cancel	
21		int_MVV_6	Jave as type. ASCI	romat (.ASC)			
22		int_MVV_65528		1000 00020	INT		
23		real_MD_32777		MD 32777	REAL		

Exporting Sources

These files are created exporting source code.

- 1. Open any program block in the editor, "OB1" in this example.
- 2. From the File menu choose Generate Source: the following dialog is displayed:

LAD/STL/FBD - [OB1 "Cycle Exe	ecution" S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\\0B1]
🖬 File Edit Insert PLC Debug Vi	iew Options Window Help
	Contents Of: 'Environment\Interface'
	Interface Name
New network	E-E TEMP
FB blocks	
SFB blocks	New 🔀
SFC blocks	Entru noint View
Multiple instances	Project Component view Conline Coffine
	Name: Storage path:
	S7_ProtocolTest
	E ProtTest UDT etc
	Comment
	⊡ s7 Program(1)
	Blocks
	Object name: Sources
	Object type:
	STL Source
	OK Cancel Help
<u>±</u>	+D
	T "dint MD 5"

1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.

Generate source Sources						
Note: Automatic generation of single source Menu 'Options' > 'Customize' in the 'So	s per block: burces' tab					
Path: S7_ProtocolTest\SIMATIC 300 Station\0	Path: S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\					
Blocks Not Selected:	Blocks Selected:					
> 	DB1 UDT Struct Test OB1 Cycle Execution UDT1 BasicDataType_UDT UDT2 ComplexDataTypes					
Name/Family:						
 Include reference blocks Sort according to program structure 	Addresses C Absolute C Symbolic					
ОК	Cancel Help					

- 2. Click **All >** to generate source for all blocks.
- 3. Select the following options:
- Include reference blocks
- Sort according to program structure
- Symbolic address
- 4. Click OK to confirm: the "Sources" object is generated in the Step7 project as in the example.

SIMATIC Manager - S7_Protocol	Test	
File Edit Insert PLC View Optic	ns Window Help	
🗋 🗅 😅 🔡 🛲 🛛 🔏 🛍 💼	📩 😰 🗣 🖕 🐩 🇱 🗰 主 🛛 < No Filter >	💽 🏹 🞇 😂 🖷 🗖 😢
S7_ProtocolTest C:\Program	Files\Siemens\Step7\s7proj\S7_Pro~1	
• ● ● S7_ProtocolTest • ● ● SIMATIC 300 Station • ● ● CPU315-2DP(1) • ● ● S7 Program(1) • ● ● S7 Program(1)	ProtTest_All ProtTest_UDT_etc	

5. Right click on the object and select **Export Sources**.

📔 🔍 < No Filter >	✓ ¹ / ₂ ¹ / ₂	s 🛞 🖷		▶?
proj\\$7_Pro~1				
:t_UDT_etc 🛅 Sources				
	Open Object	Ctrl+Alt+O	_	
	Cut	Ctrl+X		
	Сору	Ctrl+C		
	Paste	Ctrl+V		
	Delete	Del		
	Insert New Object		•	
	PLC		×	
	Compile	Ctrl+B		
	Export Source			
	Print		۲	
	Rename	F2		
	Object Properties	Alt+Return		
	Special Object Properties		×	

The generated .awl file can be imported in the Tag Editor.



Note: The .awl file contains additional information not included in the .asc file exported from the symbol table.

Make sure that reference to all data blocks is inserted in the symbol table. The tags from a data block are imported only if the symbol table contains a line with the data block name and related comment.

S7 Pro	57 Program(2) (Symbols) CPU314C-2PNDP_MPI_187K\SIMATIC S7-300 Station 1\CPU 314C-2 PN/DP					
	Status	Symbol 🛆	Address	Data type	Comment	
1		CPU_FLT	OB 84	OB 84	CPU Fault	
2		I/O_FLT2	OB 83	OB 83	I/O Point Fault 2	
3		OBNL_FLT	OB 85	08 85	OB Not Loaded Fault	
4		Prova Data Block	DB 123	DB 123		
5		Prova MB0	MB 0	BYTE		
6		VAT_1	VAT 1			
7						

Each entry enables the import filter to import the tags related to the specified data block.

Tag Editor Settings

Into Tag editor select the protocol "Profibus DP S7" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Memory Type	Offset	SubIndex		
Internal Memory 🛛 👻	0	0 +		
Data block	Data Type		Arraysize	
1	boolean	•	0	
Conversion				
	+/-			

Element	Description				
Memory	Area of PLC where tag i	s located			
гуре	Data Type		Simatic Type		
	Internal Memory		М		
	Data Block		DB		
	Input		I (E)		
	Output	O (A)			
	Timer value	Т			
	Counter value		С		
Offset	Offset address where tag is located				
SubInde x	In case of Boolean data type, this is the offset of single bit				
Data Block	If Memory Type is "Data Block", this will identify the DB number				
Data Type	Data Type	Memory Space		Limits	
1900	boolean	1 bit data		01	
	byte	8-bit data		-128 127	
	short	16-bit data		-32768 32767	
	int	32-bit data		-2.1e9 2.1e9	
	unsignedByte	8-bit data		0 255	
	unsignedShort	16-bit data		0 65535	
	unsignedInt	32-bit data		0 4.2e9	
	float	IEEE single-precisio	n	1.17e-38 3.40e38	
		32-bit floating point t	уре		
	string	Refer to "String data	type chanr	nel"	
	Note: to defin like "byte[]", "s	e arrays, select one o short[]"…	f Data Type	e format followed by square b	rackets
Arraysiz e	 In case of array tag In case of string tag in the string tag. 	ag, this property repre ag, this property repre	esents the n esents the r	umber of array elements. naximum number of bytes av	vailable

Element	Description					
	Note: number of byt to UTF-8 or Latin1 ir If Encoding property requires 2 bytes.	Ite: number of bytes corresponds to number of string characters if Encoding property is set UTF-8 or Latin1 in Tag Editor. Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character quires 2 bytes.				
Conversi	rsi Conversion to be applied to the tag.					
on	Conversion					
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
	Depending on data type selected, the Allowed list shows one or more conversions, listed below.					
	Value	Description				
	Inv bits	Invert all the bits of the tag. <i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)				
	Negate	Set the opposite of the tag value.				
		Example: 25.36 \rightarrow -25.36				
	AB -> BA	Swap nibbles of a byte.				
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)				
	ABCD -> CDAB	Swap bytes of a word.				
		<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)				
	ABCDEFGH -> GHEFCDAB	Swap bytes of a double word. <i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format)				

ement	Description	
	Value	Description
		$855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP -> OPMDAB	Swap bytes of a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011001000101101000011100000
		101010100001010001011011011011001011011
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	S5timer(BCD)	Used to support S5timer. Check Simatic S5timer special data type for more details.
	S5timer(BIN)	Legacy transformation for S5timer in binary format.
	Select the convers Configured list.	ion and click on plus button. The selected item will be added on
	If more conversion Configured list).	s are configured, they will be applied in order (from top to bottom of

Use the arrow buttons to order the configured conversions.

String data type

In ProfibusDP S7 PLC it's possible to define two different types of tags to manage string variables.

- as Array [1..xx] of Chars,
- as String[xx].

Step7 string declaration is showed in the following figure:

Addres	s	Name	Туре	Initial value	Comment	S7 String
	0.0		STRUCT			or ounig
4	+0.0	Stringl	STRING[254]	'sample'		
+23	56.0	String2	ARRAY[110] 🔤			
+	*1.0		CHAR		and the second se	String as array of char
=26	56.0		END_STRUCT			

TIA Portal string declaration is showed in the following figure:

	D	🌮 ata_	е ъ	₽ 12 E+ E+ E+ E	3 🔢 🚏	S7 S	String			-
-	-	Na	ame	2	Data type	Utfset	Start value	Retain	Accessible	Visible in
1			St	atic						
2				String1	String		'sample'			
3		•	€	String2	Array [1 10] of Char				\sim	
						No. of Concession, Name				
							String as a	irray of cha	ar	

1

Note: Usage of String[xx] data type is allowed but a specific Conversion must be applied to the tag. Anyway using tag importer to import tag dictionary from TIA Portal or Step7 string tags are automatically configured and no changes/conversion are needed.

To manually add an "Array [1..xx] of Chars" data type tag, press the [+] button in the Tag Editor,

then select "string" as Data Type of the Tag and type the string length in the "Arraysize" field:

Profibus DP S7	×
Profibus DP S7	
Memory Type Offset SubIndex Internal Memory 114 0	
Data block Data Type Arraysize 1 Conversion	
OK Cancel Apply	Help

Profibus DP S7				×
Profibus DP S7				
Memory Type C	offset	SubIndex		
Data block	Data Type string	~	Arraysize	
Conversion	+/-			1
	O	К	Cancel Ap	ply Help

then click on [+/-] button to open the Conversion dialog.

Profibus DP S7	×
Profibus DP S7	
Memory Type Offset Internal Memory V 114	SubIndex
Data block Data Type 1	Arraysize
Conversion +/-	-
ОК	Cancel Apply Help

Into conversion dialog:

- select the "S7 String" conversion type
- click on [+] button to add the conversion.

Profibus DP S7			×
Profibus DP S7			
Memory Type Internal Memory V	Offset	SubIndex	
Data block	Data Type	Arraysize	
Conversion	Allowed AB->BA ABCD->CDAB S7 String	Configured S7 String	
		Cancel OK	Help

The conversion will be listed into the Configured window on the right.

Confirm with OK button.

Simatic S5timer data type

Simatic drivers support a special data type, called S5Timer.

The tag must be configured with a specific data type and a conversion must be applied to the Tag to correctly read/write a Simatic S5Timer Variable.

Open the Tag Editor and add a Tag pressing the Plus button.



Select "unsignedInt" as Data Type of the Tag.

Profibus DP S7					>
Profibus DP S7					
Memory Type	Offset	SubIndex]		
Data Block	Data Type	a	Arraysize		
1	unsigned	int 🗸	0		
Conversion	bolean byte short int +/- unsignedB	lyte			
	unsignedS unsignedIr float double string	hort nt	-		
	boolean [] byte [] short []		Cancel	Apply	Help
	unsignedB unsignedS unsignedI float [] double [] wstring	iyte [] ihort [] nt []			

Click on +/- button to open the Conversion dialog.

Profibus DP S7					×
Profibus DP S7					
Memory Type	Dffset	SubIndex			
Data block	Data Type unsignedInt	~	Arraysize		
Conversion	+/				
	Ok	((Cancel	Apply	Help

In the Conversion dialog select the S5timer(BCD) conversion type [A] then click on Plus button [B] to add the conversion, the configured conversion will be listed into the Configured window on the right. Then confirm with OK.

Profibus DP S7		×
Profibus DP S7		
Memory Type Internal Memory V	Offset SubIndex	
Data block	Data Type Arraysize Image: mail of the second sec	
Conversion A S5timerBCD	Allowed Configured S5timer(BCD) S5timer(BIN))
		Help
	Cancel	ОК

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

	Tags	×								
+	_	z	đ	ß	>]	₽	A 9B	B>	63	1
Data			^		-	Та	g URI			

The following dialog shows which importer type can be selected.

Version	Туре	
TIA Portal Project v 12 or newer	Linear	
TIA Portal v13, v14 or newer	Linear	
TIA Portal v10, v11, v12	Linear	
Step7	Linear	
Tag Editor exported xml	General	


Importer	Description		
TIA Portal Project v12 or newer Linear	Allows to import the whole TIA Portal project file using .apxx file (where "xx" is the TIA Portal version, example: for TIA Portal 13, file name is "project.ap13").		
	All variables will be displayed at the same level.		
TIA Portal v13, v14 or newer Linear	Allows to import: Program blocks using .db file PLC tags using .xlsx file PLC data types using .udt file 		
	Check Export using TIA Portal v13, v14 or newer for more details.		
	All variables will be displayed at the same level.		
TIA Portal v10, v11, v12 Linear	 Allows to import: Program blocks using .tia file PLC tags using .xlsx file PLC data types using .scl file Check Export using TIA Portal v10, v11, v12 for more details. All variables will be displayed at the same level. 		
Step7 Linear	 Allows to import: Symbols table .asc file Sources using .awl file Check Export using STEP7 for more details. All variables will be displayed at the same level. 		
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.		

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					-
+ - 👗 🕲 🖉 👌 🖏	▶ 🛃 R 🔎 Search	Tilter by: Data 🔹 Ite	ms used:6/100	00 Protocol: Show all	🕑 Show all tags 🔅 🗌
Data Type	Comment	^	Property		Value
Modbus TCP:prot1			✓ Driver		
Model: Modicon Modbus(1-based)	-		Mode	1	Modicon Modbus(1-based)
Holding Registers 1 unsigned	IShort		Proto	col	Modbus TCP:prot1
Holding Registers 2 unsigned	1Short		V Dictionary	1	
Holding Registers 3 Unsigner	ISnort		Array	,	false
MRT01 Unsigned	IShort		Array	size	0
MRTU3 unsigned	IShort		Array	index.Subindex	400003
- MRTU4 unsigned	IShort		Com	nent	100005
MRTU5 unsigner	dShort		Data	type	unsignedShort
I I i i i i i i i i i i i i i i i i i i			Dutu	100	an angine aan tan c

Toolbar item	Description	
Re	Import Tag(s).	
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project	
	Update Tag(s).	
	Click on this icon to update the tags in the project, due a new dictionary import.	
R	Check this box to import all sub-elements of a tag.	
	Example of both checked and unchecked result:	
	T A T	
	Model: COESYS 3 Lonanee Model: COESYS 3 Lonanee Application Containee Application Containee Application Containee Application Containee Application Containee Application Containee	
	- Caloritation - Alternation - Caloritation - Caloritatio - Caloritation - Caloritation - Calori	
	- [1] Per - [1]	
	[4] [6] [6] [7] <th[7]< th=""> <th[7]< th=""> <th[7]< th=""></th[7]<></th[7]<></th[7]<>	
	L D01 D17 L D07	
	- Tag1 DH/T - Tag1 DH/T - Tag2 DH/T - Tag2 DH/T - Tag3 DH/T - Tag3 DH/T	
P - Search Y Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo- box item selected.	

Communication status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid	The panel did receive from the controller a response, but its format or its contents or its

Error	Notes
response	length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General error	Error cannot be identified; should never be reported; contact technical support

Rexroth IndraControl

The Rextoth IndraControl communication driver has been designed to connect HMI devices to Bosch Rexroth PLC trough ethernet connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

		OK
Alias		Cunce
IP address	0.0.0.0	
port	6042	
Timeout (ms)	1000	
PLC Models		
CODESYS 3		

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Port	Port number used by the driver. The default value is 6042 .
Timeout	Time delay in milliseconds between two retries in case of missing response from the server device.
PLC Models	PLC models available:



Element	Description
	CODESYS 3
PLC Network	Multiple controllers can be connected to one HMI device. To set-up multiple connections, select PLC network and click Add to configure each node

Data Types

The import module supports variables of standard data types and user defined data types.

Supported data types	 BOOL INT SINT UINT UDINT DINT STRING* REAL LREAL BYTE ULINT LINT
	and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.
	Note *: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.
Unsupported data types	 LWORD LINT

Tag Conversion

Conversion to be applied to the tag.

Conversion		
inv,swap2	Allowed	Configured
	BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	+ ABCD->CDAB
		Cancel OK

Value Description Inv bits inv: Invert all the bits of the tag. Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format) Negate neg: Set the opposite of tag value. Example: $25.36 \rightarrow -25.36$ AB -> BA swapnibbles: Swap nibbles in a byte. Example: $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format) ABCD -> CDAB swap2: Swap bytes in a word. Example: $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format) ABCDEFGH -> swap4: Swap bytes in a double word. **GHEFCDAB** Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format) ABC...NOP -> swap8: Swap bytes in a long word. OPM...DAB Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0 1000000110 1 10000011100 (in binary format) BCD **bcd**: Separate byte in two nibbles, read them as decimal (from 0 to 9) Example: $23 \rightarrow 17$ (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Select conversion and click +. The selected item will be added to list Configured.

If more conversions are configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Tag import

When creating the project using IndraWorks programming software, properly configure the symbol file to contain the required variables.

Symbol configuration item contains a list of all the variables available into the IndraWorks project, single variables or groups of variables can be selected and moved to **Selected variables** column.



 After the symbols have been configured, download the project or use the Generate code function (Build > Generate code) to create an .xml file containing all the variables read to be imported in the Tag Editor. The .xml file is created in "C:\ProgramData\IW-Projects\0\Project\IndraLogic" by default.

- 2. Select the driver in the Tag Editor.
- 3. Click the **Import Tags** button to start the import process.

HMIStudio		×
Multiple tag importer	rs are available for this protocol. Please select the importer type a	nd continue.
Version	Туре	
Rexroth IndraWorks v1.0	Linear	
Rexroth IndraWorks v1.0	Hierarchical	
Tag Editor exported xml 1.1	1 General	
Watched dictionary file:		
Keep synchronized	ОК	 Cancel

Select the importer by choosing from the list above.

Linear All variables will be displayed at the same level.

Da	ta	Туре
4	Rexroth IndraControl:prot1 Model: CODESYS 3	Container
	- Application/PLCPROG/mMyInt1	short
	Application/PLCPROG/mMyInt2	short
	Application/PLCPROG/xMyBool1	boolean

Hierarchical All variables will be displayed according to CODESYS V3 Hierarchical view

Data	Туре
 Rexroth IndraControl:prot Model: CODESYS 3 	Container
 Application 	Container
PLCPROG	Container
- mMyInt1	INT
- mMyInt2	INT
xMyBool1	BOOL

General Select this importer to read a general XML file exported from the Tag editor

- 4. Locate the .xml file and click **OK**: the tags included in the created document are listed in the tag dictionary.
- 5. To add the tags to the project click the **Add tags** button.

	_ Scardi			
Data	Туре		Property	Value
Rexroth IndraControl:prot:	1 Container		Model	CODESYS 3
Model: CODESYS 3			Protocol	Rexroth IndraControl:p
 Application PLCPROG mMyInt1 mMyInt2 xMyBool1 	Container INT INT BOOL			
Ready		Dictionary Name:	[Rexroth Indra	Control prot1] IndraLogic

See "My first project" section in the main manual.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this of	communication driver:
-----------------------------	-----------------------

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

ROBOX BCC/31

ROBOX BCC/31 communication driver has been designed to connect HMI devices to ROBOX BCC/31 PLC through Ethernet connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

PLC Network		OK
Alias		Cancer
IP address	0.0.0.0	
Timeout (ms)	2000	
PLC Models		
RODOX BBC/31		

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Address of PLC.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
PLC Models	Allows to select between different PLC models:

Element	Description
	Robox BBC/31
PLC Network	IP address for all PLCs in multiple connections. PLC Network must be selected to enable multiple connections.
	ROBOX BCC/31
	PLC Network OK Cancel
	Alias
	IP addres ROBOX BCC/31
	Timeout (
	PLC Mode Alias Cancel
	IP address 0 . 0 . 1
	Timeout (ms) 2000
	Slaves PLC Models
	Slave Ic

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select ROBOX BCC/31 from the Driver list: tag definition dialog is displayed.

ROBOX BCC/31				
Memory Type	Offset	SubIndex		
Logic Input Bit 🛛 👻	1	0 *		
Axis Index	Data Type		Arraysize	
1	boolean		0	
Conversion				
[+/-			



Element	Description		
Memory Type	Resource where tag is located on PLC.		
	Available resources are:		
	Logic Input Bit		
	Logic Input Word		
	Logic Output Bit		
	Logic Output Word		
	Phis Input Bit		
	Phys Input Word		
	 Phys Output Bit 		
	 Phys Output Word 		
	Non Volatile I32		
	Non Volatile Double		
	Non Volatile string		
	Volatile I32		
	Volatile Double		
	Volatile stilling Parameter 132		
	Parameter Double		
	Axis Parameter I32		
	Axis Parameter Double		
	Alarm Mask		
	Alarm Code		
	Alarm string		
Offset	Offset address where tag is loc	ated.	
	Offset addresses are six digits	composed by one digit data type prefix +	five digits resource address.
SubIndex	This allows resource offset selection within the selected memory type.		
Axis Index	Allows to select Axis index. Available only for Axis memory types.		
Data Type	Data Type	Memory Space	Limits
	boolean	1-bit data	01
	byte	8-bit data	-128 127
	short	16-bit data	-32768 32767
	int	32-bit data	-2.1e9 2.1e9

Element	Description			
	Data Type	Memory Space	Limits	
	int64	64-bit data	-9.2e18 9.2e18	
unsignedByte		8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	0 4.2e9	
	uint64	64-bit data	0 1.8e19	
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38	
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308	
string Array of elements containing character code defi encoding		code defined by selected		
	binary	Arbitrary binary data		
	Note: to define array "byte[]", "short[]"	s. select one of Data Type format followe	ed by square brackets like	
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 			
Conversion	Conversion to be applied to the	e tag.		
	Conversion			
	inv,swap2 Allower BCD AB->B ABCD- ABCD- ABCDE Inv bit Depending on data type selected	A >CDAB SFGH->GHEFCDAB S Cancel OK Cancel OK	conversion types.	



Element		
	Value	Description
	Inv bits	inv : Invert all the bits of the tag.
		Example: $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
	Negate	neg: Set the opposite of tag value.
		<i>Example:</i> 25.36 → -25.36
	AB -> BA	swapnibbles: Swap nibbles in a byte.
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)
	ABCD -> CDAB	swap2: Swap bytes in a word.
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH ->	swap4: Swap bytes in a double word.
	GHEFCDAB	Example: $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP ->	swap8: Swap bytes in a long word.
		Example: 142 366 → -893553517 588905 (in decimal format)
		0 10000000110
		→
		1 10000011100 1010101000010100010110110110
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

Element	Description
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



It is possible to import a Tag Editor exported xml

Туре	Description
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.
	Tags 🗙
	🛨 — 🎽 🕲 🔎 🔰 🚺 🕼 🖬 🕅
	Data Tag URI

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🗈 🔰 🕩	\$ ₿ ₪} ∰	R 🔎 - Search 🌱 Filter by: Data -	Ite	ems used:6/10000 Protocol: Show all	🗹 Show all tags 🔅 🗖
Data Ty	ype	Comment	^	Property	Value
Modbus TCP:prot1	ontainer			Y Driver	
Model: Modicon Modbus(1-based)				Model	Modicon Modbus(1-based)
Holding Registers 1 ur	nsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2 ur	nsignedShort			✓ Dictionary	
- MRTI 1	nsignedShort			Array	false
- MRTU2 ur	nsignedShort			Array size	0
MRTU3 ur	nsignedShort			Arravindex.Subindex	400003
-MRTU4 ur	nsignedShort		-	Comment	
-MRTU5 ur	nsignedShort			Data type	unsignedShort
				Data t/pc	an ang reacher c



Toolbar item	Description
BA	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
K 3	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:
	Tops: Image: Image: </th
P - Search Y Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo- box item selected.

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

The following commands are supported:

Command	Description
dir (node,path)	Get directory of node starting from path.
readFile (node,deviceFilePath,localFilePath)	Get file from node.
writeFile (node,deviceFilePath,localFilePath)	Write file to node.
deleteFile	Delete file into node.

Example of usage:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var params = "0 /F@/file.ext /mnt/usbmemory/file.ext";
tagMgr.invokeProtocolCommand(protID, "writeFile", params, state);
```

ROC Plus

The HMI device can be connected to a ROC Plus network as the network master using this communication driver. Communication with the ROC800 controllers is over an Ethernet or serial link. Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated.

Accordingly, always ensure that the latest driver is used in the application.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called "ROC Plus" from the list of available protocols. The driver configuration dialog is shown in figure.

ROC Plus		×
PLC Network	Comm	ОК
Alias		Cancel
Media	Serial V	
TCP/IP Address	0.0.0.0	
TCP/IP Port	4000	
Panel Address	240	
Panel Group	2	
Controller Address	×	
Controller Group	2	
Operator ID		
Password	×	
Access Level	-1	
Timeout (ms)		
PLC Models		
ROC8xx		
ROC300		
PIODOSS		

Element	Description
Media	Specify if the HMI is connected to the controller via serial communication link or Ethernet (TCP/IP)
TCP/IP Address	Ethernet IP address of the controller
TCP/IP Port	Port number used by the ROC plus driver; the default value can be changed when the

Element	Description
	communication goes through routers or Internet gateways where the default port number is already in use
Panel Address	Indicates the address of the HMI, this must be a unique number. 0 represents "broadcast within group" and 240 is the "direct connect address."
Panel Group	Indicates the group code for the station address. This is user-configurable and usually set to 2.
Controller Address	Indicates the address of the controller, this must be unique
Controller Group	Indicates the group code for the station address. This is user-configurable and usually set to 2.
Operator ID	Sets operator identification code for the communications port through which communications are occurring. The operator identification is logged with an event, indicating the operator responsible for creating the event.
Password	A numerical value that is used as a password for the Operator Identifier
Access Level	A value that is used to limit access to parameters.
Timeout (ms)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from the server device. Value is expressed in milliseconds.
PLC Models	The driver supports the communication with a number of different Emerson controllers. Please check directly in the programming IDE software for a complete list of supported controllers.
PLC Network	The protocol allows the connection of multiple controllers to one HMI device. To set-up multiple connections, check "PLC network" checkbox and create your network using the command "Add" per each slave device you need to include in the network.
Comm	Click on this button to configure the serial port on the panel to be used

Element	Description			
	Comm Parameter Dialog			
	OK			
	Port com1 -			
	Baudrate 2400 -			
	Parity even -			
	Data bits 8			
	Stop bits			
	Mode RS-485			
Port	On UN20:			
	com1 is the HMI port labeled "PLC",			
	com2 is the HMI port labeled "PC/Printer"			
	On UN31 or UN30:			
	 com1 is the integrated serial port, 			
	com2 is an add-on module plugged in Slot#1 or #2			
	coms is an add-on module plugged in Slot#3 of #4			
Baudrate, Parity, Data bits, Stop bits	Communication parameters for the serial line.			
Mode	Serial port mode; options are:			
	• RS-232,			
	• RS-485 (2 wires)			
	• RS-422 (4 wires)			

Tag Import

The ROC Plus driver, support the generic import of tags when provided in XML. Import procedure is described.

- make a copy of saved configuration file ".800" and rename as ".MDB"
- open the ".MDB" using Microsoft Access
- export the table "tConfigData" to a XML file choosing XML format

🔜 47 + (≌ + - File Home C	Microsoft Access reate External Data Datab	ase Tools	UtiliteUtools Fields Tab	le				×
Views Clipboard 3	Image: Sort & Filter Image: Sort & Filter	Σ F	ind Rate of the second	Switch Windows /indow	A Text Formatting *		122	
ables	⊗ «							
tAlarmHeader	<u> </u>							
tAlarms	tConfigData							
tConfig	ConfigData	- Confi	igHeade 🔹 Point	Tyj - Poin	tNuml - Para	imet 🚽	Valı -	Γ
tConfigData		1	1	0	1	0	0.0	
tConfigHeade	<u>O</u> pen		1	0	1	1	0, 0, 0	
tEventHeader	<u>D</u> esign View		1	0	1	2	0, 0, 0	
#Events	I <u>m</u> port	+	1	0	1	3	0, 0, 0	
2 LEVENIS	Export	> (X)	Excel	0	1	4	0, 0, 0	
tFileInformati	Collect and Update Data via E-ma	il 🔟	SharePoint List	0	1	5	0,0,0	
TFileInfoToRC	Rena <u>m</u> e	W	Word RTF File	0	1	7	0,0,0	
tFST	Hide in this Group		PDF or XPS	0	1	8	0.0.0	
tHistoryData	Delete	A	Access	0	1	9	0, 0, 0	
tHistoryDataH	Cut		Text File	0	1	10	0, 0, 0	
tHistoryHead	Copy	3	XML File	0	1	11	0, 0, 0	
tHistoryPointl	Paste	100	ODB <u>C</u> Database	0	1	12	0, 0, 0	
tHistorySegm	Lare		HTML Document	0	1	13	0, 0, 0	
tHistoryTypes	Linked lable Manager	dB	dBASE File	0	1	14	0, 0, 0	
tMaterCalibra 🥷	Convert to Local Table	W	Word Merge	0	1	15	0, 0, 0	
tMeterCalibrationE		5	No F	Iter Searc	h			
stack ast Visuu	and the first firs				Marrie I.	ante (PP)	an an	1.4

• In the tag editor select the driver and click on the "Import tag" button to start the importer.

HMIStudio	×
Multiple tag importer	s are available for this protocol. Please select the importer type and continue.
Version	Туре
ROCLink v1.0	Linear
Tag Editor exported xml 1.1	. General
Watched dictionary file:	
Keep synchronized	
	OK Cancel

• Locate the ".XML" file and confirm. The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in figure.

	1:Page1 Protocol	s Tags 🗙				
+	- 2 6 0	>] [> \$I	B [3] [3] R	P- Search	T Filter by: Data	√ Items
Data	3	Туре Т	Tag name	Tag URI	Dictionary	
4	ROC Plus:prot1 Model:	Container				
	str	string-16 s	tr	0??4?0?string-16		
	ARRAY_WORD[1]	short-8 A	ARRAY_WORD[1]	0?ARRAY_WORD[1]?4?1?short-8		
	- ARRAY_WORD[2]	short-8 A	ARRAY_WORD[2]	0?ARRAY_WORD[1]?4?2?short-8		
L	ARRAY_WORD[3]	short-8 A	ARRAY_WORD[3]	0?ARRAY_WORD[1]?4?3?short-8		
	- ARRAY WORD[4]	short-8 A	ARRAY WORD[4]	0?ARRAY_WORD[1]?4?4?short-8		
L .	- MDW2	unsignedShort M	1DW2	0?ARRAY_WORD[1]?4?5?unsignedS	Short	
	MDW3	unsignedShort M	IDW3	0?ARRAY_WORD[1]?4?6?unsignedS	Short	
L .		-				

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
No response	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Not expected response TLP	The panel did receive from the controller a response with invalid Type Logical Parameter
Can't find the TLP location	The panel can't get the physical location of the type or the logical number in the ROC800
Not expected number of items	Controller did not accept write request; ensure the data programmed in the project are consistent with the controller resources
Wrong datagram data length	The panel did receive from controller a response frame contains wrong data length
PLC is in the firmware update mode	Firmware Update Mode – Extremely limited functionality is available.
Not expected response length	The panel did receive from the controller a response with invalid message length
Can't read port security mode	Security Access Mode for the port the request was not received
Can't read compatibility mode	Logical Compatibility Mode was not received

Error	Notes
Can't get IO point types	The ROC Plus database is broken into individual parameters. Each database parameter is uniquely associated by parameter number and point The panel did not receive the requested Point Type
Can't send the request	The panel cannot sent any request to the controller
Not expected response group/unit	The panel did receive from the controller a response with invalid Group/Unit
Not expected opcode in the response	The panel did receive from the controller a response contains an unexpected operation code action to perform.
Invalid format received	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
Message checksum error	The panel did receive from the controller a response contains an invalid checksum

SAIA S-BUS

The SAIA S-BUS communication driver has been designed to connect HMI devices to SAIA PLCs through serial connection.



HMIs from UN65 and UN70 platforms do not support PARITY mode on PLC configuration due hardware incompatibility.

DATA mode is supported in all HMI platforms.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

SAIA S-BUS		×
PLC Network	Comm	ОК
Alias		Cancel
Node ID	1	
Timeout (ms)	200	
Retry count	2	
data/parity protocol		
PLC Models		
PCD1 PCD2 PCD3		

Element	Description
Node ID	SAIA PLC node on the serial network.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
Retry count	Defines the number of times a certain message will be sent to the controller

Element	Description	
	before reporting the	communication error status.
data/parity protocol	SAIA protocol mode unchecked checked: da 	e: (default): parity mode ta mode
PLC Models	SAIA PLC models a • PCD1 • PCD2 • PCD3	vailable:
Comm	If clicked displays the Comm Parameter D Port Baudrate Mode	he communication parameters setup dialog.
	Element	Parameter
	Port	 Serial port selection. COM1: device PLC port. COM2: computer/printer port on panels with 2 partial parts or antiapal Plug is modulo plugged.
		 COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board. COM3: optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board.
	Baudrate	Serial baudrate. Available speeds:

Element	Description	
	Element	Parameter
		• 9600.
		• 19200.
		• 38400.
		• 57600 .
	Mode	Serial port mode. Available modes:
		• RS-232.
		• RS-485 (2 wires).
		• RS-422 (4 wires).
PLC Network	Multiple controllers connections, select	can be connected to one HMI device. To set-up multiple PLC network and click Add to configure each node

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select SAIA S-BUS from the Driver list: tag definition dialog is displayed.

Memory Type	Offset	SubIndex		
R#-Register 🔹	0	0 🔻	1	
Data Block	Data Type		Arraysize	
1	boolean	×	0	
Conversion				
	+/-			

Element	Description					
Memory Type	Memory Type	Description				
.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	R # -Register	unsigned 32 bit data register (default)				
	C # -Counter	unsigned 32 bit data counter (default)				
	T # -Timer	unsigned 32 bit data timer (d	unsigned 32 bit data timer (default)			
	F # -Flag	1 bit data flag				
	l # -Input	1 bit data input	1 bit data input			
	O # -Output	1 bit data output				
	Data Block	unsigned 32 bit data block (default)				
	Real Time Clock	unsigned 8 bit real time clock (default) (see Special Data Types for mode details)				
	Node Override	protocol parameter (see Spe	ecial Data Types for mode det	ails)		
Offset	Memory Type	Offset PCD1	Offset PCD2	Offset PCD3		
	R # -Register	0 – 4095	0 – 4095	0 – 16383		
	C # -Counter	0 – 1599	0 – 1599	0 – 1599		
	T # -Timer	0 – 1599	0 – 1599	0 – 1599		
	F # -Flag	0-8191	0-8191	0-8191		
	l # -Input	0-512	0-8192	0-5120		
	O # -Output	0-512	0-8192	0-5120		
	Data Block	0 – 3333	0 – 3333	0 – 16383		
	Real Time Clock	1-8	1-8	1-8		
	Node Override	0	0	0		
SubIndex	This allows resource	offset selection within the regi	ster.			
Data Type	Available data types:					
	 boolean 					
	• byte					
	• short					
	 unsignedByte 	à				
	 unsignedSho 	rt				
	unsignedInt					

Element	Description	
	 float string See "Programming con Note: To defi 	ncepts" section in the main manual. The arrays, select one of Data Type format followed by square brackets.
Arraysize	 In case of array for a read of a read of a read of string tag. Note: number of bytes of or Latin1 in Tag Editor. If Encoding property is bytes. 	tag, this property represents the number of array elements. tag, this property represents the maximum number of bytes available in the corresponds to number of string characters if Encoding property is set to UTF-8 set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2
Conversio n	Conversion to be applie Conversion inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCD=FGH->GHEFCDAB Inv bits Cancel OK
	Depending on data type	e selected, the list Allowed shows one or more conversion types.
	Value	Description
	Inv bits	inv: Invert all the bits of the tag. <i>Example:</i> $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)
	Negate	neg : Set the opposite of tag value. <i>Example:</i> $25.36 \rightarrow -25.36$
	AB -> BA	swapnibbles: Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format)

Element	Description	
	Value	Description
		$5588 \rightarrow 20813$ (in decimal format)
	ABCD -> CDAB	swap2: Swap bytes in a word.
		<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH ->	swap4: Swap bytes in a double word.
	GHEFCDAB	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)
	ABCNOP ->	swap8: Swap bytes in a long word.
	OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.10000000110 0001110010111011001000101101000011100101
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion and	click +. The selected item will be added to list Configured .
	If more conversions are	e configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons	to order the configured conversions.

Real Time Clock

The protocol provides the special data type Real Time Clock which allows you to change the date and time on PLC. This memory type is an unsigned byte.

Offset	Description
1	Number of week
2	Day of week

Offset	Description
3	Year
4	Month
5	Day
6	Hours
7	Minutes
8	Seconds

Memory Type	Offset Su	bIndex	
Real Time Clock 🔹	1	×	
Data Block	Data Type	Arraysize	
1	unsignedByte	•	
Conversion			
	+/-		

It is also possible to use the Date/Time control widget to directly write in Real Time Clock variable.

- 1) Define a Real Time Clock, as per above picture
- 2) Drag and drop the Date/Time control widget



3) From Property Pane, click on the + button beside **Value** property. Then locate the Real Time Clock variable from Tag source, and select Read/Write option.

	11	operfies			. 4
- L - B R - H -	e.] 🐺 🖁			
	12	DateTime : d	dateTime1		
]= =		Value	-1		1
*		Datal ink	Surtom Time: FurBrea	Mar	
		Mumber Form	System mileSystem		
		number Form	MM/DD/YY - hh:mm	355	
		Time Spec	local		
		OnDataUpdat	te .		ŀ
	Œ	Text			
03/30/10 - 10.01.17 -					
dataTimal value					
, date i me Lvaide					<u></u>
surce: 🛛 🔘 Tag 🔘 Alias 🖑 System 👘 Widget 💮 Recipe					
			2 at 1	TTI channel beau	
Ve Search	_	Protocol: ND0	UV. 41	NOW ALLARS.	
• Search	-05	Protocol: Sho	wai •	Show all tags	
ata Type Tag name	7053	Protocol: Sho	Property	Value	
ata Type Tag name SAUAS BUSyprot1 Container		Protocol: Sho	Property	Value	
Search Type Tag name SALAS = UStroot Model POD1 Container Model POD1 Container		Protocol: Isno	Property	Value PCD1	
A Search Fine Dys Data ata Type Tag name SATA 5-BUSsprot1 Container Real Time Clock		Protocol: Sno	Property	Value PCD1 SAIA S-8US:prot1	
Start Type Tag name SANA SEUSIDOTI Container Model: PCD1 Container Real Time Clock unsignedByte Real Time Clock		Protocol: Sho	Property A Driver Model Protocol A Tag	Value PCD1 SAIA S-BUS:prot1	
Search Type Tag name SALA - BUSTOR1 Model: PCD1 Container Real Time Clock - unsignedByte: Real Time Clock		Protocol: <u>Sno</u>	Property 4 Driver Model Protocol 4 Tag Active	Value PCD1 SAIA S-BUS:prot1 false	
Saint Type Tag name SAN 5-BUSIOFOIL Container Model: PCD1 Container Real Time Clock: unsignedByte: Real Time Clock		Protocol: <u>bno</u>	Property	Value PCD1 SAIA S-8US:prot1 false	
Start Type Tag name StAAS BUSSTOTI Container Model: PCD1 Container Real Time Clock unsignedbyte Real Time Clock		Protocol: 509	Property Driver Model Protocol Tag Active Comment Deta Type	Value PCD1 SAIA S-8US:prot1 false unsignedByte	
Search Type Tag name SANA 5 EUSTOT1 Container Model: PCD1 Container Real Time Clock. unsignedByte: Real Time Clock		Protocol: [509	Property Protecty Protocol Active Comment Data Type Default value	Value PCD1 SAIA S-BUSiprot1 false unsignet@yte	
Search Type Tag name SAAS BUSSTOTI Gentainer Model: PCD1 Gentainer Real Time Clock: unsignedByte: Real Time Clock		Protocol: Sho	Property Property Protocol Tag Active Comment Data Type Default valu Encoding	Value PCD1 SAIA S-8US:pro11 false unsignetByte 8	
Search Type Tag name SALA 5-8250rot1 Model: PCD1 Container Real Time Clock, unsignedByte Real Time Clock		Protects: Sho	Property A Driver Model Protocol A Tag A Ctive Comment Default value Encoding Groups	Value PCD1 SATA S-8USipro11 fabe unsignedByte	
Search Type Tag name SAM S-BUSIONTI Container Model: PCD1 Container Real Time Clock: unsignedByte: Real Time Clock		Protects: Sho	Property Protect Protect Protect A Tag Active Comment Data Type Default valu Encoding Groups Max value	Value PCD L SATA S-BUS:prot1 false unsignedByte e	
Sarch Type Tag name SALAS #USIONTI Container Model: PCD1 Container Real Time Clock unsignedByte Real Time Clock		Protects Sho	Property Protecol Protocol 4 Tag Active Comment Default value Encoding Groups Max value Min value	Value PCD1 SATA S-8US:pro11 fabe unsignedByte	
Search Type Tag name SALA 5-8USTOTL Model: PCD1 Container Real Time Clock. unsignedByte Real Time Clock		Protects Sho	Property A Driver Model Protocol A Tag Active Comment Default value Encoding Groups Max value Min value Pic Las nem	Value PCD1 SATA S-8USipro11 fabe unsignedByte e	
Search Type Tag name SAAS SEUSTOT1 Gontainer Model: PCD1 Gontainer Real Time Clock unsignedByte Real Time Clock		Produces Sho	Property Protect Protocol Trye Comment Deta Type Default value Encoding Groups Max value Min value PLC tag nem R/W	value PCD1 SATA S-8US:pro11 false unsignedByte R R/W	
Search Type Tag name SAAS BUSSTOTI Gontainer Real Time Clock unsignedByte Real Time Clock		Produces Sho	Property Protecol Protocol 4 Tag Active Comment Default value Encoding Groups Max value PLC tag nem R,W Rate	Value PCD1 SATA S-BUS;prot1 fabe unsignedByte R R,/W So0	
Saran Type Tag name SAA 5 8USS011 Container Model: 9201 Real Time Clock unsgriedbyte Real Time Clock		Produces sho	Property Protect Model Protocol A Tag Active Comment Data Type Default valu Encoding Groups PLC tag near R/W Rate Scaling	value PCD1 SATA S-BUS:pro11 false unsignedByte R/W S00 None	

Node Override

The protocol provides the special data type Node Override which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override is initialized with the value of the node ID specified in the project at programming time.

Node Override	Description
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.



Note: Node Override ID value assigned at runtime is retained through power cycles.

Memory Type	Offset	SubIndex		
Node Override 🔹 👻	0	0 -		
Data Block	Data Type		Arraysize	
1	unsignedBy	te 🔻	0	
Conversion				
	+/-			

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured for communication
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources

SAIA S-BUS ETH

The SAIA S-BUS ETH communication driver has been designed to connect HMI devices to SAIA PLCs through ethernet connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

SAIA S-BUS ETH		×
PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Port	5050	
Slave ID	0	
Timeout (ms)	1000	
PLC Models		_
PCD3		

Element	Description
IP address	Ethernet IP address of the controller.
Port	Port number used by the driver. The default value is 5050 .
Slave ID	ID if the controller.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
PLC Models	SAIA PLC models available:



Element	Description
	• PCD3
PLC Network	Multiple controllers can be connected to one HMI device. To set-up multiple connections, select PLC network and click Add to configure each node

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select SAIA S-BUS ETH from the Driver list: tag definition dialog is displayed.

SAIA S-BUS ETH				
Memory Type	Offset	SubIndex		
R# -Register 🔹	0	0 -		
Data Block	Data Type		Arraysize	
0	unsignedInt	t 🔻	0	
Conversion				
	+/-			

Element	Description					
Memory	Memory Type	Description				
Type	R # -Register	unsigned 32 bit data register (default)				
	C # -Counter	unsigned 32 bit data counter (default)				
	T # -Timer	unsigned 32 bit data timer (default) 1 bit data flag				
	F # -Flag					
	l # -Input	1 bit data input				
	O # -Output	1 bit data output				
	Data Block	unsigned 32 bit data block (default)				
	Real Time Clock	unsigned 8 bit real time clock (default) (see Special Data Types for mode details)				
Offset	Memory Type	Offset				
	R # -Register	0 – 16383				
	C # -Counter	0 – 1599				
	T # -Timer	0 – 1599				
	F # -Flag	0-8191				
	l # -Input	0 – 5120				
	O # -Output	0 – 5120				
	Data Block	0 – 16383				
	Real Time Clock	1-8				
SubIndex	This allows resource offset selection within the register.					
Data Type	Available data types:					
	 boolean byte short int 					
	 unsignedByte unsignedSho 	e rt				
	 unsignedSho unsignedInt 	rτ				
	 float 					
	 string 					
Element	Description					
-----------	--	--				
	See "Programming cor	ncepts" section in the main manual.				
	Note: To defi	ine arrays, select one of Data Type format followed by square brackets.				
Arraysize	In case of array	tag, this property represents the number of array elements.				
	 In case of string string tag. 	tag, this property represents the maximum number of bytes available in the				
	Note: number of bytes or Latin1 in Tag Editor. If Encoding property is bytes.	corresponds to number of string characters if Encoding property is set to UTF-8 set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2				
Conversio	Conversion to be applie	ed to the tag.				
n	Conversion					
	inv,swap2	Allowed Configured				
		BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
	Depending on data type	e selected, the list Allowed shows one or more conversion types.				
	Value	Description				
	Inv bits	inv: Invert all the bits of the tag.				
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)				
	Negate	neg : Set the opposite of tag value.				
		<i>Example:</i> 25.36 → -25.36				
	AB -> BA	swapnibbles: Swap nibbles in a byte.				
		Example: 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)				
	ABCD -> CDAB	swap2: Swap bytes in a word.				

Element	Description	
	Value	Description
		Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)
	ABCDEFGH -> GHEFCDAB	swap4 : Swap bytes in a double word. <i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)
	ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) $0\ 10000000110$ 0001110010111011010000101101000011100101
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion and	click +. The selected item will be added to list Configured .
	If more conversions are	e configured, they will be applied in order (from top to bottom of list Configured).

Use the arrow buttons to order the configured conversions.

Real Time Clock

I

The protocol provides the special data type Real Time Clock which allows you to change the date and time on PLC. This memory type is an unsigned byte.

Offset	Description
1	Number of week
2	Day of week
3	Year
4	Month



Offset	Description
5	Day
6	Hours
7	Minutes
8	Seconds

Memory Type	Offset SubInd	ex	
Real Time Clock 🔹	1 0	w	
Data Block	Data Type	Arraysize	
0	unsignedByte	0	
Conversion			
	+/-		

It is also possible to use the Date/Time control widget to directly write in Real Time Clock variable.

1) Define a Real Time Clock, as per above picture

2) Drag and drop the Date/Time control widget

_		
	Widget Gallery	+ × 🕸
	Controls	▼ Widg
A	05/30/16 - 09:57:25	tGallery
	05/30/16 - 09:57:25	
	Combo Box -	
E		

3) From Property Pane, click on the + button beside **Value** property. Then locate the Real Time Clock variable from Tag source, and select Read/Write option.

		diversites .			
- L B R - H -	6		2		
	10	DateTime : d	lateTime1		
]= =	. 8	Value	-1		T+
	*	DataLink	System Time: SysProp	Mar	
		Number Form	MM (DD /W - bbrmm		
		Time Ener	o Prey Doy 11 - Incline.	55	
		nine spec	IOCAI		
801 855 855	2.14	OnDataOpdat	/E		+
steTimeTvelue					<u> </u>
auroe: Gerag C Anas C System C Widget C Reape					
P+ Search Y Filter by: Data	٠	Protocol: Sho	wal 💌	Show all tags	1
P+ Search Type Tag name	•	Protocol: Sho	w al	Show all tags	1
P - Search Type Tag name SNA S - BUSjord1 Container	*	Protocol: Sho	w al Property A Driver	Show all tags	1
Search Type Tag name SAIA 5 8UStoro11 Container Baal Time Click understand and and and and and and and and and		Protocol: Sho	Property A Driver Model	Value PCD1	1
Search Type Tag name SAIA 5 UKS0711 Container Real Time Clock unsigned byte. Real Time Clock	•	Protocel: She	w al Property Property Driver Model Protocol	Value PCD1 SAIA S-8US:prot1	1
Search Type Tag name SAUS 58050701 Container Real Time Clock unsigned byte: Real Time Clock		Protocol: Sho	w al Property Driver Model Protocol Tag	Value PCD1 SAIA S-BUS:prot1	1
De Search Type Tag name SAIA S EUSigroff Container Model: PCD1 Container Real Time Clack unsignedByte Real Time Clack		Protocol: Sho	V al V Property A Driver Model Protocol A Tag Active	Show all tags Value PCD1 SAIA S-8US:prot1 false	Ē
D• Search Type Tag name Atts Type Tag name SA15 \$US50711 Container Real Time Clock UnsignedByte Real Time Clock	•	Protocol: Show	W al Property Driver Model Protocol Tag Active Comment	Show all tags Value PCD1 SAIA S-8US:prot1 false	Ē
D- Search Type Tag name SAIAS 58,552701 Container Medel: PCD1 Container Real Time Clack unsignedByte: Real Time Clack	•	Protocol: Shor	Property	Show all tags Value PCD1 SATA S-BUSiprot1 false unsignedByte	
Search Type Tag name SAAS-SUSSORTI Container Madel: PCII Real Time Clock unsignedByte Real Time Clock	•	Protocol: Shor	W al Property Drobcol Tag Active Comment Data Type Default value Exercise	Show all tags Value PCD1 SATA S-BUS:prot1 false unsignedByte	
Search Type Tag name SAIA 5 USSOT1 Container Real Time Clock UnsignedByte Real Time Clock		Protocol: Show	Property Property Driver Product Protocal Protocal Active Comment Deta Type Default value Encoding Grupe	Show all tags Value PCD1 SAIA S-BUS:prot1 false unsignedByte	
D- Search Type Tag name SAIA 5 80,50701 Container Medie: PCDI Container Real Time Clack: unsignedByte: Real Time Clack	•	Protocol: Show	w al Property Driver Model Protocol Tag Active Comment Default value Encoding Groups Parvaiue	Show all tags Value PCD1 SAIA S-8USapro11 false unsignedByte	
Search Type Tag name SAA 5-8USD011 Container Real Time Cloid: unsignedByte Real Time Cloid:	•	Protocol: Show	Property Property Model Protocol Tag Active Comment Default volume Encoding Groups Max volume Min value	▼ Show all tage Value PCD1 SAIA S-8US:prot1 false unsignedByte	
D• Search Type Tag name stats Type Tag name SAIA 54UStort1 Container Real Time Clock UnsignedByte Real Time Clock	•	Protocol: Show	Property Driver Property Driver Producil Protocal Active Comment Delia Type Default value Encoding Groups Plax value Plax value	♥ Show all tage Value PCD1 SAIA 5-8USpro11 false unsignedByte	
D- Search Type Tag name SAIA 5 80,50701 Container Medie: PCDI Container Real Time Clack: unsignedByte: Real Time Clack	•	Protocol: Sho	W al Property Departy Departy Tag Protocol Tag Active Comment Data Type Default value Encoding Groups Plax value PluC tag neme R/W	V Show all tags Value PCD1 SAIA 5-8US:prot1 false unaignedByte R/W	
Search Type Tag name SAIA 5 9US0011 Container Real Time Clock unsignedByte Real Time Clock	•	Protocol: Sho	w al Property Default value Encoding Groups Phate R/W Rate	▼ Show all tage Value PCD1 SAIA 5-BUSsprot1 false unsignedByte	

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Simatic S7 PPI

HMI devices can be connected to the Siemens Simatic S7-200 family of PLCs. The communication is performed via the PLC programming ports using the PPI and the PPI+ protocols.

This document describes the PPI+ protocol and includes the information needed for a successful connection.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Simatic S7 PPI		×
PLC Network	Comm	ОК
Alias		Cancel
Panel ID	1	
Slave ID	2	
MaxID	1	
PPI +		
Timeout (ms)	1000	
PLC Models		
S7-212	A	•
\$7-214 \$7-215/216 \$7-221		
\$7-222 \$7-224/226	~	,

Element	Description
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.
Panel ID	Node number of the operator panel.
Slave ID	Node number of the connected PLC.
Max ID	Available only if PPI+ protocol is in use. Contains the highest node number in PPI+ network.

Element	Description
PPI+	Checked to use PPI+ protocol instead of PPI protocol.
Timeout (ms)	Time delay in milliseconds between two retries of the same message when no answer is received from the controller.

ement	Description						
LC odels	Several Siemens contr software for a complete	ollers are supported. e list of supported cor	Please check directly in the programming IDE trollers.				
omm	If clicked displays the c	If clicked displays the communication parameters setup dialog.					
	Comm Parameter Dialo	g					
			ок				
	Port	com1	•				
	Baudrate	9600	•				
	Parity	even	•				
	Data bits	8					
	Stop bits						
	Mode	DS 485	•				
	Mode	13-405					
	Mode	13-103					
	Element	13-403	Parameter				
	Element Port	13-103	Parameter Serial port selection.				
	Element Port		Parameter Serial port selection. On UN20:				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port.				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30:				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2				
	Element Port		Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2 • COM3: optional module plugged on Slot 3/4				
	Element Port Baudrate, Parity, Da	ta Bits, Stop bits	Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2 • COM3: optional module plugged on Slot 3/4				
	Element Port Baudrate, Parity, Da Mode	ta Bits, Stop bits	Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2 • COM3: optional module plugged on Slot 3/4 Serial line parameters. Serial port mode. Available modes:				
	Element Port Baudrate, Parity, Da Mode	ta Bits, Stop bits	Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2 • COM3: optional module plugged on Slot 3/4 Serial line parameters. Serial port mode. Available modes: • RS-232.				
	Element Port Baudrate, Parity, Da Mode	ta Bits, Stop bits	Parameter Serial port selection. On UN20: • COM1: device PLC port. • COM2: PC/printer port On UN31 or UN30: • COM1: integrated serial port • COM2: optional module plugged on Slot 1/2 • COM3: optional module plugged on Slot 3/4 Serial line parameters. Serial port mode. Available modes: • RS-232. • RS-485 (2 wires).				

Tag Editor Settings

Memory Type	Offset	SubIndex		
Variable memory	• 0	0 -		
Data Type	Arraysize	Conversion		
short -	•] [+/-	

In the Tag Editor select Simatic S7 PPI from the list of defined protocols and click + to add a tag.

Element	Description			
Memory Type	Area of PLC where tag is located.			
Offset	Offset address where tag is located.			
SubIndex	In case of Boolean data type, this is the offset of single bit.			
Data Type Available data types: • boolean • byte • short • int • unsignedByte • unsignedShort				
	 unsignedInt float string See "Programming concepts" section in the main manual. 			
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character 			

Element	Description			
	requires 2 bytes.			
Conversion	Conversion to be Conversion	e applied to the tag.		
	inv,swap2	Allowed Configured		
	Depending on da	BCD AB->BA ABCD->CDAB ABCD=FGH->GHEFCDAB Inv bits Cancel OK ata type selected, the list Allowed shows one or more conversion types		
	Value	Description		
	Inv bits	inv : Invert all the bits of the tag.		
		Example: 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)		
	Negate	neg : Set the opposite of tag value.		
		<i>Example:</i> 25.36 → -25.36		
	AB -> BA	swapnibbles: Swap nibbles in a byte.		
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
	ABCD ->	swap2: Swap bytes in a word.		
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
	ABCDEFGH -	swap4: Swap bytes in a double word.		
	GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)		
	ABCNOP -> OPMDAB	swap8 : Swap bytes in a long word. Example:		

Element	Description		
	Value	Description	
		$\begin{array}{l} 142.366 \rightarrow -893553517.588905 \mbox{ (in decimal format)} \\ 0 \ 10000000110 \\ 0001110010111011010000101101000011100101$	
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$	
		0001 = 1 (first nibble) 0111 = 7 (second nibble)	
	Select conversio	n and click +. The selected item will be added to list Configured .	
	If more conversion Configured).	ons are configured, they will be applied in order (from top to bottom of list	
	Use the arrow bu	Ittons to order the configured conversions.	

PPI+ Connectivity

HMI devices can be connected to more than one CPU S7-200, more than one operator panel can also be connected to the same PLC.

Operator panels will not interfere with PPI+ communication between the PLC's.



PPI+ protocol allows you to use more complex configurations than the standard PPI protocol.

Each PLC can execute read and write operations to and from other PLCs. At the same time more than one panel can be connected on the PPI network and can access all the variables from all the PLCs.

PLC programming software can be used and online programming can be performed without interfering with the panel-PLC communication .

Communication Status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Siemens S7 Optimized

Siemens S7 Optimized communication driver has been designed to communicate with Siemens PLCs through Ethernet connection.

PLC must either have an on-board Ethernet port or be equipped with an appropriate Ethernet interface (either built-in or with a module).

This communication driver allows communication with PLCs which have been programmed using optimized Data Blocks.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

PLC Network		OK
Alias		Cancer
IP address	0.0.0.0	
Slot	2	
Timeout	1000	
PLC Models		
S7-1200/1500		

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP	Ethernet IP address of PLC.



S7-1200 and S7-1500 PLC configuration

S7-1200 (starting from firmware version 4.0) and S7-1500 PLC Series from Siemens have a built-in firewall; by default the maximum protection level is enabled. To establish communication with these PLC models it is necessary to enable S7 communication with 3rd party devices; this setting is available in TIA Portal programming software.

- 1. Open the PLC project in TIA Portal.
- 2. Select the PLC from the project tree and open PLC Properties.



3. In General > Protection choose a permission between the top three (make sure that the tick is present on HMI column).

General	IO tags	System co	nstants	Texts					
General		^	Deat	reation					
PROFINET in	terface [X1]		PIOL	ection					
General			Pr	otection					
Ethernet	addresses								
Time syn	chronization		S	elect the access level for the Pl	LC.				
Operatin	g mode								
Advance	d options			Access lev	el	Access		Access permi.	
Web serv	eraccess				HMI	Read	Write	Password	
Hardware	e identifier			Full access (no protection)	on) 🗸	~	~	1.	e
DI 8/DQ 6		105		Read access	1	1			-
Al 2			8	HMI access	×	- 74			
High speed	counters (HSC)			No access (complete pro	otection)				
Pulse gener	ators (PTO/PWM	0							
Startup									
Cycle									
Communication load		F	Full access (no protection):						
System and	clock memory		N	to password is required.	uons will have access to all fun	ictions.			
 Web server 				•					
User interfa	ce languages								
Time of day									
Protection		1000							
Configuratio	n control	~							

Note: If "No access" is selected, the communication with the panel will not be established.

4. Scroll down the page and check "Permit access with PUT/GET communication from remote partner".

PL	C_1 [CPU 1511-1	PN]	×
	. . .		
	General	Jtags Texts	
	General		•
	PROFINET inte		
	Startup		
	Cycle	Connection mechanisms	
	Communicati		
	System and clo	Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC,)	
	System diagn		
	Web server		
	Display		
	User interface I		
	Time of day		
	Protection		
	System powe		
	Connection res		
	Overview of a		=
			4
			*
		OK Cancel	3

Note: If variables are defined in "Program blocks", DB must configured as "Optimized".

To check or change DB optimization, open DB Properties:

1

 Program blocks Add new block Add news DB4 [DB Open DB4 [DB Open Snapshot of the monitor values Snapshot of the monitor values Apply snapshot values as start values + Cut Cut+X Copy Cut+X Copy as text Delete Paste Cut+K Copy as text Download to device Conpile Download to device Conss-reference information Shift+F11 Cross-references form blocus Call structure Assignment list Switch programming language Know-how protection	😼 Unline & dia	igno	STICS	5 📲	рірро
Add new block Main [OB1] DB2 [DB2] DB4 [DB Default till Datal view Datal view Portal vi	🗢 🚽 🕞 Program blo	cks		6 📲	pluto
Main [0B1] DB4 [DB4 DB4 [DB4 Open Fachnology Generate source from blocks Snapshot of the monitor values Apply snapshot values as start values • Cut Cut Ctrl+C Tag table Opy Ctrl+C Tag table Opy as text Potal view Go online Cutl+K Copy as text Open Copy as text <p< th=""><td>📑 Add new</td><td>bloc</td><td>x 🛛</td><td>7</td><td>«Add new»</td></p<>	📑 Add new	bloc	x 🛛	7	«Add new»
DB2 [DB2] DB4 [DB DB4 [DB DB4 [DB External sou External sou PLC tags Show all Add new Match and filling Program infi Text lists Documentation Compile Documentation Compile Documentation Conss-references Conss-references F11 Coss-references Assignment list Switch programming language Name Print Ctrl+P Print Print Print Print Print Print Print preview	💶 Main [OB	1]			
● DB4 [DB4 Open ● Technology ● Generate source from blocks ● PLC tags Snapshot of the monitor values ● PLC tags Apply snapshot values as start values • ● Add new ● Copy ● Tag table Open ● PLC data typ Copy as text ● Add new ● Paste ● PLC data typ Copy as text ● Add new ● Delate ● PLC data typ Copy as text ● Add new ● Delate ● PLC data typ Copy as text ● Program infe Delete ● Text lists ● Delatils view ● Documentation Shift+F11 ● Details view Switch programming language ● Print Ctrl+P ● Print Ctrl+P ● Print preview Alt Enter	📕 DB2 (DB2	1			
 Generate source from blocks Snapshot of the monitor values Apply snapshot values as start values Cut Ctrl+X Copy Ctrl+X Copy as text Copy as text Copy as text Compile Details view Call structure Switch programming language Know-how protection Print Ctrl+P Print preview Protal view 	🥃 DB4 (DB4		Open		
Snapshot of the monitor values PLC tags Add new Add new Add new Copy Tag table PLC data typ Add new Add new Tag table PLC data typ Add new Match and fi Program infi Program infi Text lists Common data Common data Common data Conss-references Call structure Call structure Switch programming language Name Portal view	🕨 🕨 🙀 Technology	-	Generate source	from block:	s
 PLC tags Apply snapshot values as start values Apply snapshot values as start values Apply snapshot values as start values Cut Ctrl+X Copy Ctrl+X Copy as text Conpile Copy as text Conpile Consor-reference information Shift+F11 Corsor-references Call structure Assignment list Switch programming language Know-how protection Ctrl+P Print preview Portal view	🕨 🕨 📷 External sou		Snapshot of the n	nonitor val	ues
Show all Apply snapshot values as start values Add new Default till Tag table Tag table PLC data typ Add new Mutch and fill Mutch and fill Mutch and fill Text lists Decal modu Mutch and fill Text lists Documentation Documentation Details view Name Portal view Apply snapshot values as start values Cut Cut Ctrl+X Copy Ctrl+C Copy as text Cop	🕶 浸 PLC tags				
Add new Default till Tag table Tag table Paste Copy as text Add new Mugdata	🍇 Show all		Apply snapshot ve	alues as st	art values 🕨
Default tr Tag table Paste Copy as text Add new Made new Match and f Match and f Match and f Match and f Program info Text lists Compile Text lists Common data Common data Common data Common data Compulation Compile Documentation Conss-reference information Shift+F11 Cross-references Call structure Assignment list Switch programming language Know-how protection Print. Print preview	📑 Add new	Ж	Cut		Ctrl+×
Image table Image table </th <td>💐 Default t</td> <td></td> <td>Сору</td> <td></td> <td>Ctrl+C</td>	💐 Default t		Сору		Ctrl+C
 PLC data type Add new my_data mu_data mu_data	🍓 Tag table	i i	Paste		Ctrl+V
Add new Image: my_data Image: my_d	🕶 💽 PLC data typ		Copy as text		
Image: my_data Image	📑 Add new	×	Delete		Del
Image: my_data Image	🦉 my_data		Rename		F2
Watch and f Program info Text lists Text lists Text lists Common data Common data Common data Conss-reference information Shift+F11 Coss-references Call structure Call structure Assignment list Switch programming language Know-how protection Name Portal view	📑 🔣 my_data		0		`
Program info Text lists Image: Common data Image: Common data <td>🕨 🕽 🔛 Watch and f</td> <td>1</td> <td>Complie Doumload to doui</td> <td>~~</td> <td></td>	🕨 🕽 🔛 Watch and f	1	Complie Doumload to doui	~~	
Text lists Image: Common data Image: Common d	🕮 Program inf		Co online	ce	Ctrl . K
 Local modu Common data Common data Courserteference information Shift+F11 Cross-references F11 Call structure Assignment list Switch programming language Know-how protection Name Print Ctrl+P Print preview 	🛅 Text lists	2	Go offline		Ctrl+M
 Common data Cross-reference information Shift+F11 Documentation Cross-references Call structure Call structure Assignment list Switch programming language Know-how protection Name Print Print Ctrl+P Print preview 	🕨 🧾 Local modu	- 10e.			
 Documentation Cross-references Call structure Call structure Assignment list Switch programming language Know-how protection Name Print Print preview Properties 	🔹 🕨 🧊 Common data		Cross-reference ir	nformation	Shift+F11
Call structure Contine access Online access Assignment list Switch programming language Know-how protection Name Print Print Ctrl+P Print preview Print preview Alt Enter	🔹 🕨 🧾 Documentation	×	Cross-references		F11
Online access Assignment list Switch programming language Know-how protection Name Print Otrl+P Print preview Portal view	🕩 [🐻 Languages & r		Call structure		
✓ Details view Switch programming language Switch programming language Know-how protection Name Print Print preview Print preview Portal view Properties	Gnline access		Assignment list		
Name Know-how protection Name Print Print preview Print preview Portal view Properties	 Details view 		Switch programm	iing langua	ige 🕨 🕨
Name Print Ctrl+P Print preview Print preview Portal view Offenties			Know-how protec	tion	
Print preview Portal view Properties Alt Enter	Name		Print		Ctrl+P
Portal view Properties Alt Enter	c	A	Print preview		
	Portal view	Q	Properties		AlthEnter

In General > Attributes check "Optimized block access":

EXOR

082 [D82]	<
General	
General Information Time stamps Compilation Protection Attributes	Attributes Only store in load memory Data block write-protected in the device Optimized block access
	OK Cancel

If check box "Optimized block access" is not available (grayed-out) it could be because DB is an "instance DB" linked to an "optimized access FB".

After compiling the project, tag offsets will be shown close to variable name.

These settings can be applied to TIA Portal programming software, S7-1200 PLC family starting from PLC firmware version 4.0 and S7-1500 PLC family.

Direct Import of TIA Portal project

It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

Data Blocks must be set as Optimized:

- 1. Configure the Data Block as **Optimized**.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and select Optimized block access.



DB2 [DB2]	\mathbf{x}
General	
General Information Time stamps Compilation Protection Attributes	Attributes Only store in load memory Data block write-protected in the device Optimized block access
	OK Cancel



Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Tag Editor Settings

In the Tag Editor select "Simatic S7 ETH" from the list of defined protocols and click + to add a tag.

Siemens S7 Optimized		
Access Type	Data Type Arraysia	ze
PLC variable	✓ unsignedByte ✓ 0	
Conversion	Variable name	
	<u>+/-</u>	

Element	Description			
Memory Type	Y Area of PLC where tag is located.			
1900	Туре	Description		
	PLC variable		Variables in from TIA P	mported ortal project.
	Node Override IP		Check "Special data type" chapter	
Data Type	Data Type	Memory Space		Limits
3 1	boolean	1-bit data		0 1
	byte	8-bit data		-128 127
	short	16-bit data		-32768 32767
	int 32-bit data		-2.1e9 2.1e9	
	unsignedByte	8-bit data		0 255
	unsignedShort 16-bit data			0 65535
	unsignedInt	32-bit data		0 4.2e9
	float	IEEE single-precision 32-bit floating point type		1.17e-38 3.4e38
	double	IEEE double-precision 64-bit floating point type		2.2e-308 1.79e308
	string Array of elements containing characteristic selected encoding		ng character	code defined by

Element	Description			
Arraysiz e	 In case of arra In case of striin in the string ta Note: number of byte to UTF-8 or Latin1 in If Encoding property requires 2 bytes.	ray tag, this property represents the number of array elements. ring tag, this property represents the maximum number of bytes available tag. tes corresponds to number of string characters if Encoding property is set in Tag Editor. y is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character		
Conversi on	Conversion to be ap Conversion inv,swap2	Allowed Configured		
	Depending on data t below.	AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK type selected, the Allowed list shows one or more conversions, listed		
	Value	Description		
	Inv bits	Invert all the bits of the tag. <i>Example:</i> $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)		
	Negate	Set the opposite of the tag value. <i>Example:</i> $25.36 \rightarrow -25.36$		
	Swap nibbles of a byte. <i>Example:</i> $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)			
	ABCD -> CDAB	Swap bytes of a word. <i>Example:</i> $9ACC \rightarrow CC9A$ (in hexadecimal format) $39628 \rightarrow 52378$ (in decimal format)		

Element	Description	
	Value	Description
	ABCDEFGH ->	Swap bytes of a double word.
	Example: 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)	
	ABCNOP -> OPMDAB	Swap bytes of a long word. Example:
		142.366 → -893553517.588905 (in decimal format) 0 1000000110
		$\begin{array}{c} 0001110010111011001000101101000011100101$
		101010100001010001011011011011001011011
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$
		0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select the conversion Configured list.	on and click on plus button. The selected item will be added on
	If more conversions Configured list).	are configured, they will be applied in order (from top to bottom of
	Use the arrow buttons to order the configured conversions.	

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

siemens 57 Opunized			
Access Type	Data Type	Arraysize	
Node Override IP	unsignedByte []	• 8	
Conversion	Variable name		
	+/-		

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.

HMIStudio		×
Multiple tag importers are	available for this protocol. Please select the importer type and	continue.
Version	Туре	
TIA Portal Project v12 or newer	Linear	
Tag Editor exported xml 1.1	General	
Mark hard at a transmission of the		
vatched dictionary file:		
_		•••
Keep synchronized		
	ОК Са	ancel

Importer	Description	
TIA Portal Project v12 or newer Linear	Allows to import the whole TIA Portal project file using .apxx file (where "xx" is the TIA Portal version, example: for TIA Portal 13 , file name is "project.ap13"). All variables will be displayed at the same level.	
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.	
	Data Tag URI	

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

			•
R 🔎 - Search Trilter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	I 💽 Show all tags 🔅 🗌
Comment	^	Property	Value
		✓ Driver	
		Model	Modicon Modbus(1-based)
		Protocol	Modbus TCP:prot1
		✓ Dictionary	
		Array	false
		Allay	Taise
		Array size	0
		Arrayindex.Subindex	400003
		Comment	
		Data type	unsignedShort
	R P- Search Tilter by: Data	R P- Search There by: Data - Iter Comment	R P- Search Filter by: Data Items used:6/10000 Protocol: Show all Comment Property Driver Model Protocol Protocol V V Dictionary Array size Array size Array size Comment Data type

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
督	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
	Data Type Con Accodests V3 Emlagoni2 Container Container Container Container Container Model Contai
P → Search Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid	The device did received a response with invalid	Ensure the data programmed in the project are

Codes supported by this communication driver:

Error	Cause	Action
response	format or contents from the controller .	consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Simatic S7 ETH

Simatic S7 ETH communication driver has been designed to communicate with Simatic controllers through Ethernet connection.

The Simatic controller must either have an on-board Ethernet port or be equipped with an appropriate Ethernet interface (either built-in or with a module).

Communication is based on the PG/OP (ISO on TCP) communication functions.

This documents describes the driver settings to be applied in programming IDE software and in S7 PLC programming software.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In Config node double-click Protocols.
- 2. To add a driver, click +: a new line is added.
- 3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

PLC Network		ОК
Alias		Cancel
IP address	0.0.0.0	
Slot	2	
PLC Models		
S7-3xx		<u>^</u>
\$7-313/314		
S7-317		
\$7-318		-

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Slot	Number of the slot where the CPU is mounted. 2 for S7-300, may take a higher value for S7-

Element	Description
	400 systems.
PLC Models	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.

S7-1200 and S7-1500 PLC configuration

S7-1200 (starting from firmware version 4.0) and S7-1500 PLC Series from Siemens have a built-in firewall; by default the maximum protection level is enabled. To establish communication with these PLC models it is necessary to enable S7 communication with 3rd party devices; this setting is available in TIA Portal programming software.

- 1. Open the PLC project in TIA Portal.
- 2. Select the PLC from the project tree and open PLC Properties.



3. In General > Protection choose a permission between the top three (make sure that the tick is present on HMI column).

General	IO tags	System co	onstants	Texts						
General		^	Dra	tection						
PROFINET in	terface [X1]		PIC	itection						
General			P	rotection						
Ethernet	addresses									
Time syn	chronization			Select the acce	ss level for the PLC.					
Operatin	g mode									
Advance	d options				Access level		Access		Access perm	ai
Web serv	eraccess					HMI	Read	Write	Password	
Hardware	e identifier			Full acc	ess (no protection)	1	~	×		w.
DI 8/DQ 6		10		Read ag	cess	1	1			-
AI 2			E.	HMI acc	ess	~				
High speed	counters (HSC)			No acce	ess (complete protection)		-			
Pulse gener	ators (PTO/PWM	0	-	0						
Startup										
Cycle										
Communica	tion load			Full access (no	protection):	H.P.				
System and	clock memory			No password is	and HMI applications will na required	ave access to all fun	ctions.			
Web server										
User interfa	ce languages	1								
Time of day										
Protection			-							
Configuratio	in control	~								

Note: If "No access" is selected, the communication with the panel will not be established.

4. Scroll down the page and check "Permit access with PUT/GET communication from remote partner".

PL	C_1 [CPU 1511-1	PN]	×
	. . .		
	General	Jtags lexts	
	General		•
	PROFINET inte		
	Startup		
	Cycle	Connection mechanisms	
	Communicati		
	System and clo	Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC,)	
	System diagn		
	Web server		
	Display		
	User interface I		
	Time of day		
	Protection		
	System powe		
	Connection res		
	Overview of a		=
	e		*
		OK Cancel	3

Note: If variables are defined in "Program blocks", DB must configured as "Not optimized".

To check or change DB optimization, open DB Properties:

i

😼 Uniine & dia	gnostics 🔰 🐿	рірро
🗢 🚽 Program blo	cks 6 👊	pluto
📑 Add new l	olock 7	«Add new»
💁 Main (OB1]	
🥃 DB2 [DB2		
🥃 DB4 (DB4	Open	
🕨 🙀 Technology	🖶 Generate source from blocks	;
🕨 🖬 External sou		les
🕶 🌄 PLC tags		
🍇 Show all	Apply snapshot values as sta	irt values 🕨
📑 Add new	🗶 Cut	Ctrl+X
💐 Default ta	💼 Сору	Ctrl+C
🍇 Tag table	💼 Paste	Ctrl+V
👻 💽 PLC data typ	Copy as text	
📑 Add new	▼ Delete	Del
📑 my_data	Rename	F2
📑 my_data		
🕨 🛄 Watch and fi	Compile Doumland to douise	•
📴 Program info	Download to device	Perel - K
🛅 Text lists	So offline	Ctrl+N Ctrl+M
🕨 🚺 Local modu	Se onine	CLN+W
🕨 🏹 Common data	Cross-reference information	Shift+F11
🕨 🛅 Documentatior	Cross-references	F11
🕨 🚺 Languages & ri	Call structure	
🕨 🖬 Online access	Assignment list	
 Details view 	Switch programming langua	ge 🕨 🕨
	Know-how protection	
Name	💻 Print	Ctrl+P
¢	Print preview	
Portal view	🔨 Properties	AltxEnter
i oltai vien		NF

In General > Attributes uncheck "Optimized block access":

General	
General General Information Time stamps Compilation Protection Attributes	Attributes

If check box "Optimized block access" is not available (grayed-out) it could be because DB is an "instance DB" linked to an "optimized access FB".

After compiling the project, tag offsets will be shown close to variable name.

These settings can be applied to TIA Portal programming software, S7-1200 PLC family starting from PLC firmware version 4.0 and S7-1500 PLC family.

Logo! PLC configuration

To configure communication with Logo! PLC:

- 1. Open the Logo!Soft Comfort project.
- 2. Select **Tools > Ethernet Connections**: the Configure Ethernet Connections dialog is displayed.

Configure Ethernet	Connections	
Module Address		
IP Address	192.168. 2.210	
Subnet Mask:	255.255.255. 0	
Default gateway	192.168. 20. 10	
Ethernnet connection	ons onnections tion1(Server)	_
	OK Cancel	Help

- 3. Right-click on Ethernet Connections and add a server connection.
- 4. Double-click on the newly created connection: the connection properties dialog is displayed.

Connection1(Server)
Local Properties (Server)
 ✓ Connect with an Operator Panel (OP) ✓ Accept all connection requests
Only this connection:
Remote Properties (Client)
Keep Alive
Enable the Keep Alive function for this connection
Keep Alive Interval: 🔜 이는 📗 Seconds
OK Cancel Help

- 5. Select the Connect with an operator panel (OP) (0BA7 model only, do noth check for Logo! 0BA8 model)
- 6. Select Accept all connection requests options.
- 7. In the Remote Properties (Client) section, set TSAP to 02.00.

Direct Import of TIA Portal project

It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

Data Blocks must be set as Not optimized:

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

General	Attributes
Time stemps	
Compilation	Only store in load memory
Protection	
Attributes	Data block write-protected in the device

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Export using TIA Portal v13, v14 or newer

Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as Not optimized.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

Information Time stamps	
Time stamps	
•	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
	Optimized block access



Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

4. Right-click on the Data Block and choose Generate source from blocks:
| PLC_1 [CPU 1212C DC/DC/DC] 3 Image: pluto Image: program blocks Image: pluto Image: pluto Open < | 📥 Devices & networks | | | 2 | - | • | pippo | |
|--|----------------------------|-----|-----------------|--------|-------|--------|--------------|---|
| Image: Device configuration Image: Online & diagnostics Image: Device configuration Image: Device proxy date Image: Device proxy date <td>▼ 1 PLC_1 [CPU 1212C DC/I</td> <td>DQ</td> <td>DC]</td> <td>3</td> <td>-</td> <td>•</td> <td>pluto</td> <td></td> | ▼ 1 PLC_1 [CPU 1212C DC/I | DQ | DC] | 3 | - | • | pluto | |
| ♥ Online & diagnostics ♥ Program blocks ● Main [OB1] ● DB2 [DB2] ● DB4 [DB4] ● PLC tags ○ Default tag table_1 [0] ● PLC data types ● Add new datat ● UDT1 ● Coldata types ● Add new datat ● UDT1 ● Watch and force ts ● Online backups ● Tactsiss ● Program info ● Coldat modules ● Common data ● Documentation settif ● Common data ● Common data ● Common data ● Conline access | 时 Device configuratio | n | | | | | | |
| Program blocks Add new block Main [OB1] DB2 [DB2] Open Open Open Ctrl+X Copy Ctrl+C Copy as text Paste Ctrl+V Copy as text Add new tag tag Add new tag tag Add new tag tag Add new tag tag Add new data Compile PLC tags Add new data Compile PLC tags Add new data Online backups Snapshot of the monitor values Program info Traces Program info Traces Program info Traces Construction Shift+F11 Construction Construction Construction Construction Switch programming language Know-how protection Print Ctrl+P Print preview Program info Traces Program info Traces Program info Traces Program info Constructure Program info </td <td>限 Online & diagnostic</td> <td>s</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> | 限 Online & diagnostic | s | | | | | | |
| Add new block Main [OB1] DB2 [DB2] Open Open Ctrl+X Copy of the monitor values Add new tagta Copy as text Copy as text Copy as text Compile Delete Delete Delete Delete Compile Common data Conss-reference information Shift+F11 Coss-references F11 Coss-references Shift+F11 Coss-references F11 Carl Reader/USB memori Print Ctrl+P Print preview Properties Alt+Enter | 🔻 🔙 Program blocks | | | | | | | |
| Main [OB1] DB2 [DB2] DB4 [DB4] Technology object Technology object Technology object Technology object Technology object Cut Cut+X Copy Cut+C Paste Cut+V Show all tags Add new tag tag Tag table_1 [0] Test lists Traces Online backups Snapshot of the monitor values Apply snapshot values as start values > Coross-reference information Shift+F11 Coross-references TratText lists Common data Common data Common data Conline access Card Reader/USB memore | 💣 Add new block | | | | | | | |
| ● DB2 [DB2] Open ● DB4 [DB4] Open ● Technology object Cut Ctrl+X ● External source file Copy Ctrl+C ● PLC tags Paste Ctrl+V ● Default tag table Open Paste Ctrl+V ● PLC data types ● Paste Del Rename F2 ● PLC data types ● Oompile ● Paste Del ● Del C data types ● Add new data ● Oompile ● Paste Del ● Online backups ● Generate source from blocks ● Generate source from blocks ● Generate source from blocks ● Crl+M ● Common data ● Card Reader/USB memor ● Print Ctrl+P ● Print preview ● Print preview ● Print preview ● Print preview | 💶 Main [OB1] | | | | | | | |
| DB4 [DB4]OpenImage: Component of the sector | 🥃 DB2 [DB2] 🗧 | | | | | | | 1 |
| Technology object Cut Cut | 🥃 DB4 [DB4] | | Open | | | | | |
| External source file Copy Ctrl+C Paste Ctrl+V Show all tags Add new tag tas Default tag table_1 [0] Tag table_1 [0] Tag table_1 [0] Add new datat UDT1 Add new datat UDT1 Add new datat UDT1 Compile Compile Conjile Compile Compile Go online Ctrl+K Go online Ctrl+K Go offline Ctrl+M Snapshot of the monitor values Apply snapshot values as start values Cross-references Call structure Assignment list Switch programming language Know-how protection Print_meview Properties | 🕨 🕨 🙀 Technology object | Ж | Cut | | | | Ctrl+X | |
| PLC tags Show all tags Add new tag ta Add new tag ta Default tag table Tag table_1 [0] Tag table_1 [0] PLC data types Add new data tig UDT1 Watch and force ta Online backups Traces Online backups Text lists Text lists Text lists Text lists Common data Common data Common data Common data Contine access Card Reader/USB memori Print Ctrl+P Print preview Properties | 🕨 🔚 External source file | È | Сору | | | | Ctrl+C | |
| Show all tags Add new tag ta Add new tag ta Tag table_1 [0] Image: Tag table_1 [0] Ima | 🔻 🚂 PLC tags 🔤 | Ì | Paste | | | | Ctrl+V | |
| Add new tag ta Default tag table_1 [0] Image: Tag table_1 [0] | 🝇 Show all tags | | Copy as text | | | | | |
| Default tag table Image: Tag table_1 [0] Image: PLC data types Image: PLC data types< | 📑 Add new tag ta | × | Delete | | | | Del | |
| Image: Tag table_1 [0] Image: PLC data types Image: PLC data ty | 🍯 Default tag tab | | Rename | | | | F2 | |
| Image: PLC data types Compile Image: PLC data types Image: Add new data time: UDT1 Download to device Image: PLC data types Image: Qubic time: Qubic time: Program info Image: Qubic time: Qubic | 👆 🔩 Tag table_1 [0] | | | | | | | |
| Add new data t UDT1 Image: Second | 🔻 [🕞 PLC data types | | Compile | | | | • | |
| Image: Solution of the sector of the sector of the sector of the sector of the monitor values Image: Solution of the sector o | 📑 Add new data t | _ | Download to de | vice | | |)
Omlar | |
| Watch and force ts Online backups Traces Device proxy data Program info Text lists Common data Common data Common data Common data Conline access Card Reader/USB memor Card Reader/USB memor Print Print Print Print preview Properties | 🖪 UDT1 🕴 | 2 | Go online | | | | Ctri+K | |
| Image: Second sec | 🕨 🦳 Watch and force ta | | Goomne | | | | CLN+IW | |
| Image: Second Secon | 🕨 📴 Online backups 🛛 | R- | Snapshot of the | mor | hitor | valu | les | |
| | 🕨 🔄 Traces | | Apply snapshot | valu | es a | is sta | art values 🕨 | |
| Image: Program info Cross-reference information Shift+F11 Image: Text lists Cross-references Image: Common data Call structure Image: Common data Assignment list Image: Common data Switch programming language Image: Card Reader/USB memore Print Image: Card Reader/USB memore Print preview Image: Program info Alt+Enter | 🕨 🗽 Device proxy data | ₽ | Generate sourc | e froi | m bl | locks | ; | |
| Text lists Coal modules Common data Common data Common data Canguages & resourc Online access Card Reader/USB memor Print Print preview Properties | 📴 Program info | _ | Cross-reference | info | ma | tion | Shift, E11 | |
| Coal modules Common data Common data Common data Call structure Assignment list Switch programming language Know-how protection Print Print preview Properties | 🔄 Text lists | v | Cross-reference | < | inia | LIOIT | 5111L+111 | |
| Common data Common data Common data Assignment list Switch programming language Know-how protection Print Ctrl+P Print preview Properties | 🕨 🦬 Local modules | Ê. | Call structure | 5 | | | | |
| Image: Second Sector Image: Second Sector Image: Second Sector Image: Second Sector Image: Second Second Sector Image: Second S | 🕨 🙀 Common data | | Assignment list | | | | | |
| Image: Switch programming language Image: Switch programming language | 🕩 🛅 Documentation settir - | === | | | | | | |
| Card Reader/USB memor
Card Reader/USB memor
Print preview
Properties
Alt+Enter | 🕨 词 Languages & resourc | | Switch program | iminę | g lar | ngua | ge 🕨 | |
| Card Reader/USB memor
Print Ctrl+P
Print preview
Properties Alt+Enter | 📷 Online access | | Know-how prote | ectio | n | | | |
| Print preview | 🄄 Card Reader/USB memor | | Print | | | | Ctrl+P | |
| Rroperties Alt+Enter | | R | Print preview | | | | | |
| | | Q | Properties | | | | Alt+Enter | |

5. Save the file as DBxxx.db, where xxx=number of DB.

S7-1200	Name			Data type	Offset	Start	value	Retain	Accessible f
💣 Add new device 🦷				1					
h Devices & networks	Save As								— ×
🛅 PLC_1 [CPU 1212C DC/DC/DC] 👘	🔘 🖉 🖉 🖉 🖉	er ► My	/Documents	Automation	•	44	Search Au	tomation	2
🛐 Device configuration			-				L		
😨 Online & diagnostics	Organize 🔻 Nev	v folder							• 0
🔻 🚘 Program blocks	Recent Places		Name	<u>^</u>		Date	modified	Tune	
📑 Add new block	inconcernation in the		INGILIE			Date	mounieu	Type	
💁 Main [OB1]	🔭 Lilenenia e			N	Jo items match v	our se	arch.		
🥃 DB2 [DB2]	Cibraries				,				
🥃 DB4 [DB4]	Documents								
🕨 🙀 Technology objects	🚽 Music								
External source files	📔 Pictures								
🔻 🚂 PLC tags	📑 Videos	=							
🍇 Show all tags									
📑 Add new tag table	🜏 Homegroup								
💥 Default tag table [29]									
🍓 Tag table_1 [0]	Computer								
🔻 [PLC data types	I compater								
📑 Add new data type	• ••••								- F
围 UDT1	F 1								
Watch and force tables	File name:	DB2.0D							
🕨 📴 Online backups	Save as type:	DB files((*.db)						
🕨 🔄 Traces									
🕨 强 Device proxy data	A Hide Folders						Save	C;	ancel
📴 Program info	- inde l'olders								
🛅 Text lists									

Exporting PLC tags

An Excel file refers to PLC tags.

- 1. Double-click **Show all tags**: the tag table is displayed.
- 2. Click the **Export** button and browse for path file.
- 3. Define file name.



4. Click **Save** to confirm.

	•		1.4.11		
Organize 🔻 New	folder			• ==	6
Favorites Fosktop Desktop Recent Places Libraries Documents Music	Name	^ No item	Date modified s match your search.	Туре	
E Pictures Videos Homegroup File name: Save as type: 2	← ← PLCTags.xlsx Klsx files (*.xlsx)	in			

5. Click **OK** to export.

Export to Excel X
Path of export file:
C:\Users\User\Documents\Automation\PLCTags.xlsx
Elements to be exported:
🗹 Tags
Constants
6 OK Cancel

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

🍯 Default tag table [31]	
🍇 Tag table_1 [0]		
🔻 [PLC data types		
📑 Add new data type		
년 UDT1	Open	
UDT2		
🔹 🕨 🔜 Watch and force tables	X Cut	Ctrl+X
🕨 📴 Online backups	E Copy	Ctrl+C
🕨 🔄 Traces	Paste	Ctrl+V
🕨 🔚 Device proxy data	Copy as text	
🖳 Program info	🗙 Delete	Del
🔄 Text lists	Rename	F2
Local modules	Compile	
🕨 📑 Common data	🧭 Go online	Ctrl+K
Documentation settings	🖉 Go offline	Ctrl+M
Languages & resources		
🔚 Online access	Generate source from blocks	
🤠 Card Reader/USB memory 🦳	Cross-reference information	Shift+F11
	🔀 Cross-references	F11
	🔢 Call structure	
	🧾 Assignment list	
	昌 Print	Ctrl+P
	📤 Print preview	
	🔍 Properties	Alt+Enter

In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .UDT file that contains all the PLC data types defined.

🍇 Tag table_1 [0]		
🔻 🌆 PLC data types		
📑 Add new data typ	e	
度 UDT1 度 UDT2	Open	
UDT3	🗶 Cut	Ctrl+X
UDT4	🛅 Сору	Ctrl+C
Watch and force table	💼 Paste	Ctrl+V
🕨 🙀 Online backups	🗙 Delete	Del
🕨 🔄 Traces	Rename	F2
🕨 📴 Device proxy data	Compile	•
📴 Program info	🝠 Go online	Ctrl+K
🖹 Text lists	🔊 Go offline	Ctrl+M
🕨 🫅 Local modules	Generate source from block	5
🕨 🏹 Common data		
Documentation setting:	Cross-reference information	Shift+F11
🕨 🐻 Languages & resources	Cross-references	F11
🚮 Online access	Call structure	
📑 Card Reader/USB memory	Assignment list	
	昌 Print	Ctrl+P
	📤 Print preview	
	🔯 Properties	Alt+Enter

In the next step, give a name to the .UDT file and choose the path to where to save the file.

Save As			×
U » 🌡 🤊 🔾	er 🕨 My Documents 🕨 Automation 🛛 👻 🍫	Search Automation	Q
Organize 🔻 N	w folder	:== ▼	0
📃 Recent Place:	A Name Da	ate modified Type	
 □ Libraries □ Documents □ Music □ Pictures □ Videos ○ Homegroup ○ Computer 	No items match your	search.	
• • • •	✓ < III		÷.
File name:	myUDTfile		•
Save as type:	UDT files(*.udt)		•
🔿 Hide Folders		Save Cancel	

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check **Tag Import** chapter for more details.

Export using TIA Portal v10, v11, v12

Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:

😼 Online & alagi	nostics	5	📶 bibbo
🗢 🔫 🌄 Program block	s	6	🕣 pluto
📑 Add new bl	ock	7	<add new=""></add>
💶 Main (OB1)			
😈 DB2 [DB2]			
🥃 DB4 (DB4	Open		
🕨 🕨 🙀 Technology	🕈 Generate sou	arce from b	locks
🕨 🔚 External sou		he monito	rvalues
💌 🌄 PLC tags 🚽			
🍇 Show all	Apply snapsh	ot values a	is start values 🕨
📑 Add new	🕻 Cut		Ctrl+X
💐 Default ta 🎚	📄 Сору		Ctrl+C
🎚 Tag table 🗐	Paste		Ctrl+V
🕶 💽 PLC data typ	Copy as text		
📑 Add new			
🥵 my_data	K Delete		Del
🥵 my_dataj	кепате		F2
🕨 🔙 Watch and fi	Compile		•
📴 Program info	Download to	device	•
🔄 Text lists 👂	🍠 Go online		Ctrl+K
🕨 🕽 🖬 Local modu	न Go offline		Ctrl+M
🕨 🙀 Common data	Cross-referen	ce informa	tion Shift+F11
🕨 🛅 Documentation	≷ Cross-referen	ces	F11
🕞 🔚 Languages & ri	👔 Call structure		
Conline access	📗 Assignment l	ist	
 Details view 	Switch progra	amming lar	nguage 🕨 🕨
	Know-how pr	otection	
Name	Print		Ctrl+P
e []	🕺 Print preview		
	- B Properties		Alts Enter
🔹 Portal view 📃 🕒	s riopenies		ALCINEI

3. In the General tab select Attributes and unselect Optimized block access.

General		
General Information Time stamps Compilation Protection Attributes	Attributes Only store in load memory Data block write-protected in the device Optimized block access	

1

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Pr	oject Edit View Insert Online	Optio	ons Tools Window Help
	🛉 🎦 🔒 Save project ا 🐰 🧾	1 i >	(🕥 ± (주 ± 🙀 🗟 🛯 👔 🖳 🖉 🗛 🌽 Go online 🧬 Go offline 👬 📘
	Project tree		Tia_project_Live_Ivery PLC_1 [CPU 1211C DC/DC/DC] > PLC tags
	Devices		
	🖻 O O	B	2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
			DL G .

- Tia_project_Luca [V11] → PLC_1 [CPU 1211C DC/DC/DC] → Program b Devices 🖻 O O 🛫 🛫 🎭 🤛 🗰 🗛 📾 🔛 🖤 DB2 📥 Devices & networks Data type Offset Start value ~ Name 🕶 🛅 PLC_1 [CPU 1211C DC/DC/... 🕣 👻 Static 🚔 Insert row III Device configuration 2 - 1 tag1 🚔 Add row 😼 Online & diagnostics 3 tag2 ¥ Cut Ctrl+X 🕶 🔙 Program blocks 4 📶 = tag3 📑 Add new block 5 Static_1 💼 Copy Ctrl+C 💼 Paste k Ctrl+V Static_1[0] 💶 Main [OB1] 6 🕣 . a) 🥃 DB2 [DB2] 7 -Static_1[1] 🗙 Delete Del 📕 DB4 [DB4] 8 Static 2 Rename F2 🕨 🙀 Tech 📕 Untitled - Notepad 🕨 词 External File Edit Format View Help 🛨 🚂 PLC tag: 👆 Shov Static 0.0 false False 1.0 ' False 2.0 16#0 False Array [0.1] of Bool Bool 0.0 false Bool 0.1 false tag1 tag2 Bool True 💣 Add True True 4.0 False Char 🝯 Defai tag3 Byt Static_1 Static_1[0] Static_1[1] Byte False Tr 🔩 Tag t True 🕶 🛅 PLC date False True
- 4. Build the project to make sure TIA Portal calculates the tags offset.

- 5. Double-click on a DB name.
- 6. Expand the view of program block selected.
- 7. Select all rows.
- 8. Copy and paste into any text editor.
- 9. Save the file as DBxxx.tia, where xxx=number of DB.



Note: Make sure you use the **Save As** function or the file will be named DB2.tia.txt and will not be visible from the importer.

10. Repeat from step 5 for all program blocks.



Note: Make sure that only the following columns are shown in DB editor before copying all data in the txt file

		🗸 Find an	d replace
Accessible from HMI	Comment		
		Show/Hide 🕨 🕨	📝 Name
	CURRENT PAGE DISPLAYED ON C	Show all columns	📝 Data type
	PANEL PUSHBUTTON PB01	Optimize width	🗹 Offset
	PANEL PUSHBUTTON PB02	Optimize width of all columns	📃 Default value
	PANEL PUSHBUTTON PB03		🛃 Start value
	PANEL PUSHBUTTON PB04	Find in :	sı 🥅 Snapshot
	FEEDBACK FOR FIELD COLOR	Eind in l	i 📃 Monitor value
			🛛 🗹 Retain
		Use will	🏽 🗹 Accessible from HMI
		Use reg	u 📃 Visible in HMI
			Setpoint Setpoint
		Whole whole	🛿 🗹 Comment
		From c	u More

Exporting PLC tags

An Excel file refers to PLC tags.

1. Double-click **Show all tags**: the tag table is displayed.



- 2. Click the Export button and browse for path file.
- 3. Define file name.
- 4. Click Save to confirm.



5. Click OK to export.

Path of export fi	e:				
C:\Documents	and Settings\a	admin\Desktop	VPLC tags xis	×	
Elements to be	exported:				
🖌 Tags					
Constants					

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .SCL file that contains all the PLC data types defined.



In the next step, give a name to the .SCL file and choose the path to where to save the file.

Save As							? 🛛
Save in:	C exported			~	00	🕫 🛄 •	
My Recent Documents							
My Documents							
My Computer							
My Network	File name: Save as type:	my_SCL scl files (with_all_UD1 *.scl)	1		~	Save Cancel

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check Tag Import chapter for more details.

Export using STEP7

The Simatic S7 ETH Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.

SIMATIC Manager - S7_Protocol	lTest		
File Edit Insert PLC View Optic	ons Window Help		
🗋 🖆 🚼 🛲 🕺 🛍 💼		🗈 📔 < No Filter >	🔄 🏹 🞇 😂 🖷 🗖 🚺 🙌
S7_ProtocolTest C:\Program	m Files\Siemens\Step7\s7proj	j\\$7_Pro~1	
S7_ProtocolT est SIMATIC 300 Station CPU315-2DP(1) S7 Program(1) Sources Blocks	Sources 🔂 Blocks	Symbols	

- 1. From the Symbol Table menu in the Symbol Editor choose Export.
- 2. Assign a name and save the symbol table as ASCII file.

Symbol Table Edit Insert View Options Window Help							
🖻 🖬	6	👗 🖻 💼 Ю СЧ	All Symbols	-	🏹 💦		
	Status	Symbol 🛆	Address	Data type	Comment		
1		A234567890123456789012	MVV 65524	WORD			
2		BasicDataType_UDT	UDT 1	UDT 1			
3		bit_M_2_0		M 2.0	BOOL		
4		bit_M_32770_0		M 32770.0	BOOL		
5		bit_M_65522_0		M OFFOR	DOO!		
6		bit_M_7_5	Export			? 🔀	
7		byte_MB_3					
8		byte_MB_32771	Save in: 🗁 Prot le:	st		💌 🗮 🖿 💷 •	
9		byte_MB_65523	BrotTest Symbols	01.acc			
10		char_MB_7	Process_pyndors_01.asc				
11		ComplexDataTypes	Procrest_symbols	_02.asc			
12		Cycle Execution					
13		date_MVV_24641					
14		dint_MD_32773					
15		dint_MD_5					
16		dint_MD_65525					
17		dword_MD_0					
18		dword_MD_32768	File name:	at Cambrill 01	1	C	
19		dword_MD_65520	File fidille.	st_bymbols_02	2.850	Save	
20		int_MW_32774	Save as tupe: ASCII	Format (* ASC)		- Cancel	
21		int_MVV_6	Jave as type. ASCI	ronnac(.ASC)			1
22		int_MVV_65528		1999 00020	INT	l	"
23		real_MD_32777		MD 32777	REAL		
		1		1	1	1	

Exporting Sources

These files are created exporting source code.

- 1. Open any program block in the editor, "OB1" in this example.
- 2. From the File menu choose Generate Source: the following dialog is displayed:

LAD/STL/FBD - [OB1 "Cycle Exe	ecution" S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\\OB1]
File Edit Insert PLC Debug Vi	iew Options Window Help
D≓₽₽₽₽₩₽₽₽	· · · · · · · · · · · · · · · · · · ·
X New network FB blocks FC blocks SFB blocks SFB blocks SFB blocks Multiple instances Multiple instances Libraries	Contents Of: 'Environment\Interface' Name TEMP TEMP Entry point: View: Project Component view Name: Storage path: S7_ProtocolTest C:\Program Files\Siemens\Step7\s7pr Browse Entry point: S1: S1ProtocolTest Commer S1ProtocolTest S1: S1ProtocolTest S1: S1ProtocolTest S1: S1ProtocolTest S2: S1ProtocolTest S1: S1ProtocolTest S2: S1ProtocolTest S2: S1ProtocolTest S2: S1ProtocolTest S3: S1ProtocolTest S3: S1ProtocolTest S3: S1ProtocolTest S3: S1ProtocolTest S3: S1ProtocolTest S1: S1ProtocolTest S1: S1ProtocolTest S1: S1ProtocolTest S1: S1ProtocolTest S2: S1ProtocolTest S2: S1ProtocolTest S1:
	Object name: Sources Object type: STL Source
<u> </u>	DK Cancel Help

1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.

Generate source Sources					
Note: Automatic generation of single sources Menu 'Options' > 'Customize' in the 'So	per block: urces' tab				
Path: S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\					
Blocks Not Selected:	Blocks Selected:				
> [All > [<	DB1 UDT Struct Test OB1 Cycle Execution UDT1 BasicDataType_UDT UDT2 ComplexDataTypes				
Name/Family:					
 Include reference blocks Sort according to program structure 	Addresses C Absolute © Symbolic				
ОК	Cancel Help				

- 2. Click **All >** to generate source for all blocks.
- 3. Select the following options:
- Include reference blocks
- Sort according to program structure
- Symbolic address
- 4. Click **OK** to confirm: the "Sources" object is generated in the Step7 project as in the example.

SIMATIC Manager - S7_ProtocolTest	
File Edit Insert PLC View Options Window Help	
🗋 🖸 🚅 🚼 🐖 👗 🛍 🕄 🕍 😰 🏪 🏝 🏣 🔛 🚺 🔍 (No Filter >	💽 🏹 🐮 😂 🖷 🗖 😢
\$7_ProtocolTest C:\Program Files\Siemens\Step7\s7proj\\$7_Pro~1	
ST_ProtocolTest Image: SIMATIC 300 Station Image: STPogram(1) Image: STProgram(1) Image: Sources Image: Stress Stress Image: Stress Stress Stress Image: Stress Stress Image: Stress Stress	

5. Right click on the object and select **Export Sources**.

📔 🔤 🕹 🕹 🕹	- V 8	: 🗐 🖷		
proj\\$7_Pro~1				
t_UDT_etc 🗎 Sources				
	Open Object	Ctrl+Alt+O		
	Cut	Ctrl+X		
	Сору	Ctrl+C		
	Paste	Ctrl+V		
	Delete	Del		
	Insert New Object		×	
_	PLC		۲	
	Compile	Ctrl+B		
	Export Source			
	Print		۲	
	Rename	F2		
	Object Properties	Alt+Return		
	Special Object Properties		×	

The generated .awl file can be imported in the Tag Editor.



Make sure that reference to all data blocks is inserted in the symbol table. The tags from a data block are imported only if the symbol table contains a line with the data block name and related comment.

\$7	Program(2	?) (Symbols) CPU3	14C-2PNDP	_MPI_187K\SIMATIC	S7-300 Station 1\CPU 314C-2 PN/DP
	Status	Symbol 🛆	Address	Data type	Comment
1		CPU_FLT	OB 84	OB 84	CPU Fault
2		I/O_FLT2	OB 83	OB 83	I/O Point Fault 2
3		OBNL_FLT	OB 85	OB 85	OB Not Loaded Fault
4		Prova Data Block	DB 123	DB 123	
5		Prova MB0	MB 0	BYTE	
6		VAT_1	VAT 1		
7					

Each entry enables the import filter to import the tags related to the specified data block.

Tag Editor Settings

In the Tag Editor select "Simatic S7 ETH" from the list of defined protocols and click + to add a tag.

Memory Type	Offset	SubIndex		
Internal Memory 👻	0	0 *		
Data Block	Data Type		Arraysize	
1	unsignedBy	vte 🔹 🔻	0	
Conversion				
	+/-			

Element	Description					
Memory	Area of PLC where tag is located.					
туре	Data Type	Simatic Type				
	Internal Memory	М				
	Data Block	DB				
	Input	I (E)				
	Output	O (A)				
	Timer value	Т				
	Counter value	С				
Offset	Offset address where tag is located.					
SubInde x	Resource offset within the register.					
Data Block	Data block number for Data Block Memory Type.					
Data Type	Available data types: • boolean • byte • short • int • unsignedByte • unsignedShort • unsignedInt • float • string See "Programming concepts" section in the new Note: To define arrays, select one of	nain manual. of Data Type format followed by square brackets.				

Element	Description				
Arraysiz e	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 				
Conversi on	Conversion to be ap Conversion inv,swap2 Depending on data	Allowed BCD AB->BA ABCD->CDAB ABCD=FGH->GHEFCDAB Inv bits Cancel OK type selected, the Allowed list shows one or more conversions, listed			
	below.				
	below. Value	Description			
	below. Value Inv bits	DescriptionInvert all the bits of the tag. $Example:$ $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)			
	below. Value Inv bits Negate	DescriptionInvert all the bits of the tag. $Example:$ $1001 \rightarrow 0110$ (in binary format) $9 \rightarrow 6$ (in decimal format)Set the opposite of the tag value. $Example:$ $25.36 \rightarrow -25.36$			
	below. Value Inv bits Negate AB -> BA	DescriptionInvert all the bits of the tag. $Example:$ 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)Set the opposite of the tag value. $Example:$ 25.36 \rightarrow -25.36Swap nibbles of a byte. $Example:$ 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)			

ement	Description	Description			
	Value	Description			
	ABCDEFGH -> GHEFCDAB	Swap bytes of a double word. <i>Example:</i>			
		$32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format) $855441236 \rightarrow 1426062386$ (in decimal format)			
	ABCNOP -> OPMDAB	Swap bytes of a long word. Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.1000000110 0001110010111011001000101101000011100101			
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)			
	S5timer(BCD)	Used to support S5timer. Check Simatic S5timer special data type for more details.			
	S5timer(BIN)	Legacy transformation for S5timer in binary format.			
Select the conv Configured list		ion and click on plus button. The selected item will be added on			
	If more conversion Configured list).	s are configured, they will be applied in order (from top to bottom of			

Use the arrow buttons to order the configured conversions.

Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

1:Pager protocols	Tag	5* X					
- ~ V & b		Modbus TCP:pr	ot1	- 5. 🗖			
Name /		Group	Driver	Addr	ess	Comment	
Node 5 Gat, holinoite		801	3 Modeus TCP prot 1	T TTDures	predShort	1	
Node's Calis, hodrotte		301	E Wodeue TCP gent1	1 12/Dume	pretiliner		
Node 1/19, WX752, Javed		301	E Wodeue TCP prot 1	1 @@uneg	redPort		
No. 4 CHOCHERCON		.101	E Wodeue TCP prot 1	1.245-0-une	pred that		
Nederli 40421, Billi T, Inndrokie		301	\$ Modeus TCP prot 1	1.1 During	netBet		
Node1/FLOATA_hodrolla		301	E Wodeue TCP gent1	1.20umig	netBet		
No. 100 10 10 11 11 11		Ad1	E Wodeue TCP pret1	1 3 During	net(Prot)		
Node1/Water_level		100	Collection CTH Lond	1 10 0 unsi	gr <mark>edShort</mark>		
	0	Node id as defined Select Network no	d in import file ode id				
		Slave Id	Model		Alias		
		-R-R-R-11	Honor in	- Theorem	Node1		
		444.8.3	Thusan one	all the second sec	Node2		
	- 1 mill						
tagname m	e						
tagname m Water_level	e			(Ok	Cancel	
Water_level	e			[Ok	Cancel	
Vater_level	e			[Ok	Cancel	
taoname m Water_level	e			(Ok	Cancel	
taoname <u>m</u> Water_level		24		[Ok	Cancel	
taoname M m Water_level	e 	24		(Ok egres/Drof egres/Drof	Cancel	



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

String data type

In Simatic S7 PLC two different types of tags manage string variables:

- as Array [1..xx] of characters,
- as String[xx].

Step7 string declaration is shown in this example:

I	Address	Name	Туре	Initial value	Comment	S7 String
I	0.0		STRUCT			Or Samg
I	+0.0	Stringl	STRING[254]	'sample'		
I	+256.0	String2	ARRAY[110] 🔤			
I	*1.0		CHAR			String as array of char
I	=266.0		END_STRUCT			
Ľ						

TIA Portal string declaration is shown in this example:

Ŵ	1	ŝ	• • • • • • •	- 📙 🕏	S7 S	tring			
	Da	ta_	_block_1		5				
		Na	ime	Data type	Utfset	Start value	Retain	Accessible	Visible in
1		-	Static						
2		•	String1	String		'sample'			
3		•	String2	Array [1 10] of Char					
					No. of Concession, Name				
						String as a	rray of cha	ur de la companya de	

1

Note: When using String[xx] data type specific a conversion must be applied to the tag. If the tag dictionary is imported from TIA Portal or Step7 using the import tool, however, conversion of the string tags is performed automatically and no further action is required.

To add a string as an array of characters:

1. Press the + in the Tag Editor.

Simatic S7 ETH	
Simatic S7 ETH	
Memory Type Data Block	Offset SubIndex ▼ 114 0 ▼
Data Block	Data Type Arraysize string IData Type 10
Conversion	+/-
	OK Cancel Apply Help

- 2. Select string as Data Type.
- 3. Enter string length in Arraysize.
- 4. Click OK to confirm.

To add a string data type:

1. Press the + in the Tag Editor.

Simatic S7 ETH		
Simatic S7 ETH		
Memory Type Data Block	Offset SubIndex ▼ 114 ♥ 0 ▼	
Data Block	Data Type Arraysize String 10	
Conversion	+/-	
	OK Cancel Apply Help]

- 2. Select string as Data Type.
- 3. Enter string length in Arraysize.
- 4. Click +/- to open the Conversion dialog.

Simatic S7 ETH			— ×-
Simatic S7 ETH			
Memory Type	Offset	SubIndex	
Data Block	▼ 114	0 -	
Data Block	Data Type	Arraysize	
1	string	▼ 10	
Conversion		_	
	+/-		
	ОК	Cancel	Apply Help

5. In the conversion dialog select the **S7 String** conversion type.

imatic S7 ETH				X
Simatic S7 ETH				
Memory Type Data Block	Offset ▼ 114 ▼	SubIndex		
Data Block	Data Type		Arraysize	
Conversion	B			
A —	Allowed AB->BA ABCD->CDAB S7 String	E	S7 String	
			Cancel OK	Help

- 6. Click + to add the conversion: the conversion will be listed into the **Configured** list on the right.
- 7. Click OK to confirm.

Simatic S5Timer data type

Simatic drivers support a special data type, the S5Timer data type.

The tag must be configured with a specific data type and a conversion must be applied to the tag to correctly read/write a Simatic S5Timer Variable.

1. In the Tag Editor click + to add a tag.



2. Select unsignedInt as Data Type.

imatic S7 ETH		;
Simatic S7 ETH		
Memory Type	Offset	SubIndex
Internal Memory ~	50	0 ~
Data Block	Data Type	Arraysize
1		✓
Conversion	byte short	
	+/- unsignedByte	
	unsignedInt float	
	double string boolean D	
	byte [] short []	Cancel Apply Help
	int [] unsignedByte unsignedShort unsignedInt [] float [] double []	0

3. Click +/- to open the Conversion dialog.

Simatic S7 ETH		×
Simatic S7 ETH		
Memory Type Internal Memory V	Offset SubIndex	
Data Block	Data Type Arr	aysize
Conversion	+/	
	OK Cance	el Apply Help

- 4. In the conversion dialog select the **S5timer(BCD)** conversion type.
- 5. Click + to add the conversion: the conversion will be listed into the **Configured** list on the right.

matic S7 ETH			>
imatic S7 ETH			
Memory Type	Offset Subin	idex	
Internal Memory V	50 • 0	~	
Data block	Data Type	Arraysize	
1	≜ ▼ unsignedInt	√ 0	
Conversion A	B		
S5timerBCD	Allowed S5timer(BCD) S5timer(BIN)	Configured S5timer(BCD)	
		~	
	<	> >	Help

6. Click OK to confirm.

Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation				
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.				
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.				

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

Simatic S7 ETH				
Memory Type	Offset Su	ubIndex		
Data Block	Data Type		Arraysize	
Conversion	+/-			

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

2		Tags	×								
ſ	÷	-	z	ß	ß	>]	₽	A 9B	B>	6 3	1
	Data	1		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

Version	Туре
TIA Portal Project v12 or newer	Linear
TIA Portal v13, v14 or newer	Linear
TIA Portal v10, v11, v12	Linear
Step7	Linear
Tag Editor exported xml	General



Importer	Description					
TIA Portal Project v12 or newer Linear	Allows to import the whole TIA Portal project file using .apxx file (where "xx" is the TIA Portal version, example: for TIA Portal 13, file name is "project.ap13").					
	All variables will be displayed at the same level.					
TIA Portal v13, v14 or newer Linear	Allows to import: Program blocks using .db file PLC tags using .xlsx file PLC data types using .udt file 					
	Check Export using TIA Portal v13, v14 or newer for more details.					
	All variables will be displayed at the same level.					
TIA Portal v10, v11, v12 Linear	 Allows to import: Program blocks using .tia file PLC tags using .xlsx file PLC data types using .scl file Check Export using TIA Portal v10, v11, v12 for more details. All variables will be displayed at the same level. 					
Step7 Linear	 Allows to import: Symbols table .asc file Sources using .awl file Check Export using STEP7 for more details. All variables will be displayed at the same level. 					
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.					

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

	Tags 🗙	Protoc	ols													-
+	- 🎽	G	0 >]	Þ	₿B	BŊ	樹	R	₽- Search) Y i	ilter by: Data	•	Ite	em	s used:6/10000 Protocol: Show all	🕑 Show all tags 🔅 🗌
Data		^		Тур	e			Com	ment				^		Property	Value
A M	Modbus TCP:prot1 Container												✓ Driver			
- M	odel: Mod	licon Modb	us(1-based	i) con	conner									IF	Model	Modicon Modbus(1-based)
	Holding	Registers	1	unsi	gnedS	hort									Protocol	Modbus TCP:prot1
	Holding	Registers	2	unsi	gnedS	hort									 Dictionary 	·
	Holding	Registers	3	unsi	gneas	nort								IF	Array	false
	MOTUO			unsi	gneus	hort								lŀ	Array size	0
	MRTUZ			unsi	gneas	nort								Iŀ	Array size	400000
	MICTUS			unsi	gneas	nort								lŀ	ArrayIndex.Subindex	400003
	- MRTU4			unsi	gnedS	nort									Comment	
	- MRTU5			unsi	gnedS	hort									Data type	unsignedShort

Toolbar item	Description
BA	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s) . Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.

Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid	The device did received a response with invalid	Ensure the data programmed in the project are

Error	Cause	Action
response	format or contents from the controller .	consistent with the controller resources.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Simatic S7 MPI

HMI products support direct Siemens MPI communication without any additional module.

The driver supports the standard communication speed 187Kbit/s.

here is a minimum requirement also for the version of operating system running in the HMI (this is normally referenced as BSP version). See in user manual how to read the BSP version with the System Settings menu. The minimum requirements are shown in the following table.

Platform	BSP Version
UN30/31	v1.38 or newer
UN65/UN71	v1.0.300 or newer
UN60/UN70	v1.0.413 or newer
UN73	v1.0.142 or newer

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the "Simatic S7 MPI" protocol from the list of available protocols.

The protocol type can be selected from the dedicated combo box in the dialog.

10000000000000000000000000000000000000				
PLC Network		Comm	ОК	
Alias			Cancel	
Timeout (ms)	1000			
Panel MPI address	1			
Highest MPI address	15			
PLC MPI address	2			
PLC Models				
S7-3xx				
S7-313/314				=
\$7-315				1.000
57-318				
57-319				-

Element	Description								
Alias	Name to be used to identify nodes in the plc network configuration. The name will be added as a prefix to each tag name imported for each network node.								
Timeout (ms)	Defines the time inserted by the protocol between two retries of the								
	same message in case of missing response from controller.								
	Value is expressed in milliseconds.								
Panel MPI Address	MPI node number assigned to the device.								
Highest MPI Address	The highest node number in the MPI network where the device is operating and communicating.								
PLC MPI Address	The MPI address of the controller to which the device needs to communicate.								
PLC Models	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.								
Comm	Click on this button to configure the serial port on the device to be used as MPI port (see example in the following figure)								
	Comm Parameter Dialog								
	Communication parameters for Simatic S7 MPI are fixed at: • Baud rate=187500 • Parity=Even • Data=bits8 • Stop=bit1 On UN20: • com1 is the HMI port labeled "PLC",								

Element	Description
	On UN31 or UN30:
	 com1 is the integrated serial port,
	 com2 is an add-on module plugged in Slot#1 or #2
	 com3 is an add-on module plugged in Slot#3 or #4
	The connection between device and PLC can be made:
	 Creating a custom cable following the scheme provided with document CA255 "eTOP400/500 serie PLC Port to MPI Port"
	 Using a standard MPI cable with ADP-0001 "MPI wiring adapter"
PLC Network	The protocol supports connection to multiple controllers. To enable this option, check the "PLC Network" check box and enter the configuration per each controller node.

Direct Import of TIA Portal project

It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

Data Blocks must be set as Not optimized:

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose **Properties**:

😼 Online & dia	Online & diagnostics						
🔻 🛃 Program blo	Program blocks						
📑 Add new	bloc	sk 🛛 👘					
💶 Main (OB	1]						
🥃 DB2 (DB2		Open					
🗧 DB4 [DB4	V	Out	OtrlaX				
🕨 🛃 Technology (*	Conv	Ctrl+C				
🕨 🔙 External sou		Paste	Ctrl+V				
🔻 🔙 PLC tags		0					
la Show all		Copy as text					
🃑 Add new	×	Delete	Del				
🍇 Default ti		Rename	F2				
table 🔁 🔁		Compile	•				
▼ Lt PLC data typ		Download to device	•				
Add new	ø	Go online	Ctrl+K				
B UDT1	12	Go offline	Ctrl+M				
Watch and for the set of the s	R.	Snapshot of the monitor valu	es				
Image: Contine Dack	_	Apply snapshot values as sta	rt values 🕨				
 Inaces Device preva 	•	Generate source from blocks					
Program info							
Toyt lists	- 5	Cross-reference information	Shift+F11				
Text lists	r an	Cross-references	FTT				
Common dete		Assignment list					
Common data	==	Assignment list					
Documentation		Switch programming langua	ge 🕨				
Online access		Know-how protection					
🧓 Card Reader/USB m		Print	Ctrl+P				
_	A	Print preview					
	Q	Properties	Alt+Enter				

3. In the General tab select Attributes and unselect Optimized block access.

General Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
Autouces	Optimized block access

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Export using TIA Portal v13, v14 or newer

Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as Not optimized.
- 2. Right-click on the Data Block and choose Properties:



3. In the General tab select Attributes and unselect Optimized block access.

General	Π
Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
Attributes	Optimized block access
	OK Cancel



Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

4. Right-click on the Data Block and choose Generate source from blocks:
| 📥 Devices & networks | 2 🕣 = pippo |
|-------------------------------|---|
| - Time PLC_1 [CPU 1212C DQ/DQ | /DC] 3 🕣 💶 pluto |
| 时 Device configuration | |
| 😵 Online & diagnostics | |
| 🔻 🔙 Program blocks | |
| 📑 Add new block | |
| 💁 Main [OB1] | |
| 🥃 DB2 [DB2] | |
| 🥃 DB4 [DB4] | Open |
| 🔹 🕨 🙀 Technology object 🐰 | Cut Ctrl+X |
| 🔹 🕨 🔚 External source file 🧾 | Copy Ctrl+C |
| 🔻 🛃 PLC tags 🛛 💼 | Paste Ctrl+V |
| 🝇 Show all tags | Copy as text |
| Add new tag ta 🗙 | Delete Del |
| 🛎 Default tag tab 个 | Rename F2 |
| 👆 Tag table_1 [0] | On wells |
| 🔻 [🔃 PLC data types | Complie P |
| 📑 Add new data t | Coloning Child |
| | Go offline OtduM |
| Watch and force ta | |
| 🕨 🖡 Online backups 🛛 🕷 | , Snapshot of the monitor values |
| 🕨 🔄 Traces | Apply snapshot values as start values 🕨 |
| 🔹 🕨 🏢 Device proxy data 📑 | Generate source from blocks |
| 📴 Program info | Cross-reference information Shift+E11 |
| 🛅 Text lists 🗙 | Cross-references E11 |
| 🔹 🕨 🫅 Local modules 👔 | Call structure |
| 🕨 🙀 Common data 🛛 📲 | Assignment list |
| 🕩 🛅 Documentation settir 🚟 | |
| 🕨 词 Languages & resourc | switch programming language |
| 🔚 Online access | Know-how protection |
| 🄄 Card Reader/USB memor | Print Ctrl+P |
| | Print preview |
| | |

5. Save the file as DBxxx.db, where xxx=number of DB.

\$7-1200	Name		Data type	Offset	Start value	Retain	Accessible f
🗳 Add new device 👔		•					
h Devices & networks	Save As						×
🛅 PLC_1 [CPU 1212C DC/DC/DC] 👘	OO V Use	er 🕨 My Documents	Automation	•	🗛 🛛 Search Au	tomation	Q
🛐 Device configuration				I	-)(
🗓 Online & diagnostics	Organize 🔻 Nev	w folder					- 0
🔻 ⋥ Program blocks	Recent Places	Name	<u>^</u>		Date modified	Tune	
💣 Add new block	in the contractor	INGILIE			Datemounieu	Type	
💶 Main [OB1]	😂 1 Sharaha a		No it	tems match vo	ur search.		
📒 DB2 [DB2]	Cibraries						
🥃 DB4 [DB4]	Documents						
🕨 🙀 Technology objects	a) Music						
External source files	📔 Pictures						
🔻 📜 PLC tags	🛃 Videos	=					
🍇 Show all tags							
💣 Add new tag table	🍓 Homegroup						
💐 Default tag table [29]							
🍓 Tag table_1 [0]	Computer						
🔻 [eLC data types	1 computer						
💣 Add new data type	A	▼					- F
😰 UDT1	5 1	DP2 db					
Watch and force tables	File name:	DB2.0D					
🕨 📴 Online backups	Save as type:	DB files(*.db)					
🕨 🔄 Traces							
🕨 🚟 Device proxy data	A Hide Folders				Save	Ca	incel
📴 Program info	- Inde Folders						
🛅 Text lists 🛛 🕹							

Exporting PLC tags

An Excel file refers to PLC tags.

- 1. Double-click **Show all tags**: the tag table is displayed.
- 2. Click the **Export** button and browse for path file.
- 3. Define file name.



4. Click **Save** to confirm.

EX02

Save As						-		×
Us 🖉 🖉	er ▶ N	/ly Document	s 🕨 Automation	-	47	Search Auto	mation	Q
Organize 🔻 Ne	w folde	r					853 👻	0
 ☆ Favorites ■ Desktop ● Downloads ● Recent Places ○ Libraries ● Documents ● Music ● Pictures ● Videos 	* H	Name		No items match y	Date our se	e modified	Туре	
🤞 Homegroup	•	•		m				Þ
File name: Save as type:	PLCT Xlsx fil	igs.xlsx es (*.xlsx)						•
lide Folders						4 Save	Cance	

5. Click **OK** to export.

Export to Excel X
Path of export file:
C:\Users\User\Documents\Automation\PLCTags.xlsx
Elements to be exported:
🗹 Tags
Constants
6 OK Cancel
تر

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

💥 Default tag table [31]	
Tay Lable_1 [0]		
Add new data types		
	Open	
Watch and force tables	🗶 Cut	Ctrl+X
Online backups	💼 Сору	Ctrl+C
🕨 🔀 Traces	💼 Paste	Ctrl+V
🕨 🗽 Device proxy data	Copy as text	
📴 Program info	🗙 Delete	Del
Text lists	Rename	F2
Limit Local modules	Compile	•
🕨 📑 Common data	of Go online	Ctrl+K
Documentation settings	🐼 Go offline	Ctrl+M
Longuages & resources	EN Concrete course from blocks	
Donline access	Generate source from blocks	_
👼 Card Reader/USB memory	Cross-reference information	Shift+F11
	K Cross-references	F11
	🛅 Call structure	
	🛄 Assignment list	
	昌 Print	Ctrl+P
	鹶 Print preview	
	🧟 Properties	Alt+Enter

In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .UDT file that contains all the PLC data types defined.

🍓 Tag table_1 [0]		
🔻 [🗃 PLC data types		
📑 Add new data type	2	
😫 UDT1	Open	
UDT2		
UDT3	X Cut	Ctrl+X
UDT4	Copy	Ctrl+C
🕨 🕨 🔜 Watch and force tabl	E Paste	Ctrl+V
🕨 🙀 Online backups	🗙 Delete	Del
🕨 🔄 Traces	Rename	F2
🕨 🖳 Device proxy data	Compile	•
📴 Program info	🝠 Go online	Ctrl+K
🛅 Text lists	🔊 Go offline	Ctrl+M
🕨 🛅 Local modules	Generate source from blocks	
🕨 🙀 Common data		
Documentation settings	Cross-reference information	Shift+F11
🕨 🐻 Languages & resources	Cross-references	F11
🗟 Online access	E Call structure	
👼 Card Reader/USB memory	Assignment list	
	昌 Print	Ctrl+P
	🐴 Print preview	
	Roperties	Alt+Enter

In the next step, give a name to the .UDT file and choose the path to where to save the file.

Save As					×
🔾 🔾 🗸 Us	er My Documents Automation	- - - + - + + + + + + + + + +	Search Auton	nation	Q
Organize 🔻 Ne	w folder			•	0
🖳 Recent Places	^ Name	Dat	e modified	Туре	
 ➢ Libraries ➢ Documents ➢ Music ➢ Pictures ☑ Videos ở Homegroup 	E	No items match your s	earch.		
💻 Computer					
• • •		III			۰.
File name:	myUDTfile				-
Save as type:	UDT files(*.udt)				-
) Hide Folders			Save	Cance	ш .н

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check Tag Import chapter for more details.

Export using TIA Portal v10, v11, v12

Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

- 1. Configure the Data Block as **Not optimized**.
- 2. Right-click on the Data Block and choose Properties:

😼 Uniine & diag	ырро		
🗢 🔙 Program bloc	oluto		
📑 Add new b	Add new>		
💶 Main (OB1]		
🥃 DB2 (DB2)			
🥃 DB4 (DB4		Open	
🕨 🙀 Technology	•	Generate source from blocks	
🕨 🔚 External sou		Snapshot of the monitor value	es
🔻 浸 PLC tags 🛛 -		· · · · · · · · · · · · · · · · · · ·	
🍇 Show all		Apply snapshot values as stai	t values 🕨
📑 Add new	Ж	Cut	Ctrl+X
💐 Default ta		Сору	Ctrl+C
🍓 Tag table		Paste	Ctrl+V
🔻 💽 PLC data typ		Copy as text	
📑 Add new	×	Delete	Del
🥵 my_data		Rename	F2
🦉 my_data		0il-	
🕨 🥅 Watch and fi		Complie Doumland to douico	
🕮 Program info	đ	Co online	Ctrl.K
🛅 Text lists	2	Go offline	Otel-M
🕨 🖢 Local modu	<u>1</u> 12.		CLITHM
🕨 📑 Common data	-	Cross-reference information	Shift+F11
Documentation	×	Cross-references	F11
🕨 🚺 Languages & n		Call structure	
🖬 Online access		Assignment list	
 Details view 		Switch programming languag	je 🕨
		Know-how protection	
Name		Print	Ctrl+P
e	R	Print preview	
	0	Properties	AltNEnter
Portal view	_		

3. In the General tab select Attributes and unselect Optimized block access.

General	Attributes
Information	
Compilation	Only store in load memory
Protection	Dete block wite protected in the douise
Attributes	Optimized block access

1

Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Pr	oject Edit View Insert Online	Optic	ns Tools Window Help
	🞐 🎦 🔒 Save project ا 🐰 🗎	i ×	(🕥 ± (색 ± 🙀 📴 🛯 🏛 🖳 🕼 🖉 Go online 🖉 Go offline 🛔
	Project tree		Tia_project_Live_JV11// PLC_1 [CPU 1211C DC/DC/DC] > PLC tags
	Devices		
	B O O E	a	

4. Build the project to make sure TIA Portal calculates the tags offset.



- 5. Double-click on a DB name.
- 6. Expand the view of program block selected.
- 7. Select all rows.
- 8. Copy and paste into any text editor.
- 9. Save the file as DBxxx.tia, where xxx=number of DB.



Note: Make sure you use the **Save As** function or the file will be named DB2.tia.txt and will not be visible from the importer.

10. Repeat from step 5 for all program blocks.



Note: Make sure that only the following columns are shown in DB editor before copying all data in the txt file

		✓ Find and	1 replace
Accessible from HMI	Comment		1
		Show/Hide 🕨 🕨	📝 Name
	CURRENT PAGE DISPLAYED ON C	Show all columns	📝 Data type
	PANEL PUSHBUTTON PB01	Optimize width	🗹 Offset
	PANEL PUSHBUTTON PB02	Optimize width of all columns	📃 Default value
	PANEL PUSHBUTTON PB03		🗹 Start value
	PANEL PUSHBUTTON PB04	Find in s	u 🥅 Snapshot
	FEEDBACK FOR FIELD COLOR	Find in h	📄 Monitor value
	TEEDBACKTOR TIEED COLOR		🗹 Retain
		Use wild	🗹 🗹 Accessible from HMI
		🗖 Use reau	🔲 Visible in HMI
			Setpoint Setpoint
		🔘 Whole d	Comment
		Strom or	
		• From cu	More

Exporting PLC tags

An Excel file refers to PLC tags.

1. Double-click **Show all tags**: the tag table is displayed.



- 2. Click the **Export** button and browse for path file.
- 3. Define file name.
- 4. Click Save to confirm.

Name		Tag table	Data type	Address	Retain
ciao		Tag table_1	USInt	%MB0	
pr Save As		* . 11 *	101.1	or he he	? 🛛
pr Save in	n 🞯 Desktop		~	G 🕸 🔛 🛄 -	
p p p p p p m y Recent Documents Desktop My Documents My Documents	My Documer My Compute My Network	nts ar Places			
9	File name:	PLC tags.xisx			Save
	ciao pr pr Save As pr pr Save in pr My Recent Documents Desktop My Documents My Computer My Network	ciao Pr Save As Pr Save in: Pr My Recent Documents Desktop My Documents My Documents My Computer File name: Save as type:	ciao Tag table_1 Pr Save As Pr Save in: Desktop Pr My Recent Documents Public File name: Public File name: Xtas files (* day)	ciao Tag table_1 USint Pr Save As Pr Save in: Desktop My Recent Desktop Desktop Desktop Desktop My Network Places Public File name: PIC togs.x5x My Network Save as type: Xiax files (".xiax)	ciao Tag table_1 USint %MB0 Pr Save As Pr Save in: Desktop V O I I I I I I I I I I I I I I I I I I

5. Click **OK** to export.

Path of export file:		
C:\Documents and Settings\admin	Desktop\PLC tags.xlsx	
Elements to be exported:		
🗹 Tags		
Constants		
	-	
	5 ок	Cancel

Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .SCL file that contains all the PLC data types defined.



In the next step, give a name to the .SCL file and choose the path to where to save the file.

Save As							? 🔀
Save in:	🔁 exported			~	00	>	
My Recent Documents							
Desktop							
My Documents							
My Computer							
Mu Network	File name:	my_SCL_with	n_all_UDT			•	Save
My Computer	File name: Save as type:	my_SCL_with scl files (*.scl	n_all_UDT)			✓✓	Save Cancel

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

Check Tag Import chapter for more details.

Export using STEP7

The Simatic S7 MPI Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.

SIMATIC Manager - S7_ProtocolTest						
File Edit Insert PLC View Options Window Help						
🗅 😅 🎇 🥽 👗 🛍 💼 🤷 🗣 🏪 🏗 🏥 🔁 < No Filter> 🗾 🍞 🎇 😂 着 🖃 🛄 📢						
S7_ProtocolTest C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1						
S7_ProtocolTest SIMATIC 300 Station CPU315-ZDP(1) Sources Blocks Blocks						

- 1. From the Symbol Table menu in the Symbol Editor choose Export.
- 2. Assign a name and save the symbol table as ASCII file.

Symbol Table Edit Insert View Options Window Help						
🖻 🖻	5	👗 🖻 💼 🛛 🖂	All Symbols	•	🏹 💦	
	Status	Symbol 🛆		Address	Data type	Comment
1		A234567890123456789012	MVV 65524	WORD		
2		BasicDataType_UDT		UDT 1	UDT 1	
3		bit_M_2_0		M 2.0	BOOL	
4		bit_M_32770_0		M 32770.0	BOOL	
5		bit_M_65522_0		N. 05500.0	loogi	
6		bit_M_7_5	Export			? 🔀
7		byte_MB_3				
8		byte_MB_32771	Save in: 📁 ProtTest 🛛 🔽 🖛 🔝 🕶			
9		byte_MB_65523	DrotTest Symbols	01.acc		
10		char_MB_7	ProtTest_Symbols_ DestTest_Symbols_	01.asc		
11		ComplexDataTypes	E Procresc_Symbols_	U2.asc		
12		Cycle Execution				
13		date_MVV_24641				
14		dint_MD_32773				
15		dint_MD_5				
16		dint_MD_65525				
17		dword_MD_0				
18		dword_MD_32768	File name:	an Cambrille Of	1	Caura
19		dword_MD_65520	File fiame.	st_Symbols_02	.asc	Save
20		int_MVV_32774	Save as tupe: ASCII	Format (* ASC)		- Cancel
21		int_MVV_6	Save as ope. ASCI	romat (.ASC)		
22		int_MVV_65528		19199 00020	INT	
23		real_MD_32777		MD 32777	REAL	

Exporting Sources

These files are created exporting source code.

- 1. Open any program block in the editor, "OB1" in this example.
- 2. From the File menu choose Generate Source: the following dialog is displayed:

LAD/STL/FBD - [OB1 "Cycle Exe	ecution" \$7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\\0B1]
🕞 File Edit Insert PLC Debug Vie	ew Options Window Help
	·····································
X	Contents Of: 'Environment\Interface'
HKX Mau polyuosla	Interface Name Interface Imp TEMP
B blocks	
- 🔁 FC blocks	New
5FC blocks	Entry point: View:
Multiple instances	Project Component view C Online © Offine
	Name: Storage path:
	S7_ProtocolTest C:\Program Files\Siemens\Step7\s7pr Browse È 📖
	Generation S7_ProtocolTest
	0B1 : SIMATIC 300 Station
	Commer Er S7 Program(1)
	Sources Blocks
	Commer
	Object name: Causas
	Object time:
	STL Source
The second secon	
	+D T "dint MD 5"

1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.

Generate source Sources	
Note: Automatic generation of single sources Menu 'Options' > 'Customize' in the 'Sou	per block: irces' tab
Path: S7_ProtocolTest\SIMATIC 300 Station\CF	PU315-2DP(1)\
Blocks Not Selected:	Blocks Selected:
> All > <	DB1 UDT Struct Test OB1 Cycle Execution UDT1 BasicDataType_UDT UDT2 ComplexDataTypes
Name/Family:	
	Addresses
✓ Include reference blocks	C Absolute
Sort according to program structure	Symbolic
ОК	Cancel Help

- 2. Click **All >** to generate source for all blocks.
- 3. Select the following options:
- Include reference blocks
- Sort according to program structure
- Symbolic address
- 4. Click **OK** to confirm: the "Sources" object is generated in the Step7 project as in the example.

SIMATIC Manager - S7_ProtocolTest							
File Edit Insert PLC View Options Window Help							
🗋 🗅 😂 🔡 🐖 👗 🛍 🛍 😰 🐾 🗣 📴 🏥 🏛 🔁 < No Filter > 💽 🍹 🞇 🕮 🖷 🚍 🗂							
S7_ProtocolTest C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1							

5. Right click on the object and select **Export Sources**.

主 < No Filter >	- V -	🖁 🗐 🖷	
proj\\$7_Pro~1			
t_UDT_etc 🗎 Sources			
	Open Object	Ctrl+Alt+O	
	Cut	Ctrl+X	
	Сору	Ctrl+C	
	Paste	Ctrl+V	
-	Delete	Del	
	Insert New Object PLC		•
-	Compile	Ctrl+B	
	Export Source		
_	Print		•
	Rename	F2	
	Object Properties	Alt+Return	
	Special Object Properties		•

The generated .awl file can be imported in the Tag Editor.

Note: The .awl file contains additional information not included in the .asc file exported from the symbol table.

Make sure that reference to all data blocks is inserted in the symbol table. The tags from a data block are imported only if the symbol table contains a line with the data block name and related comment.

S7 Pro	\$7 Program(2) (Symbols) CPU314C-2PNDP_MPI_187K\\$IMATIC \$7-300 Station 1\CPU 314C-2 PN/DP 📃 🔲 🔀					
	Status	Symbol 🗠	Address	Data type	Comment	
1		CPU_FLT	OB 84	OB 84	CPU Fault	
2		I/O_FLT2	OB 83	OB 83	I/O Point Fault 2	
3		OBNL_FLT	OB 85	08 85	OB Not Loaded Fault	
4		Prova Data Block	DB 123	DB 123		
5	100	Prova MB0	MB 0	BYTE		
6		VAT_1	VAT 1			
7						

Each entry enables the import filter to import the tags related to the specified data block.

Tag Editor Settings

1

Into Tag editor select the protocol "Simatic S7 MPI" from the list of defined protocols and add a tag using [+] button.

Memory Type	Offset	SubIndex		
Internal Memory 👻	0	0 👻		
Data Block	Data Type		Arraysize	
1	boolean	•	0	
Conversion				
1	+/-			

Tag settings can be defined using the following dialog:

Element	Description				
Memory	Area of PLC where tag is	located.			
туре	Data Type		Simatic ⁻	Туре	
	Internal Memory		М		
	Data Block		DB		
	Input		I (E)		
	Output		O (A)		
	Timer value		Т		
	Counter value		С		
Offset	Offset address where tag is located.				
SubInde x	In case of Boolean data type, this is the offset of single bit.				
Data Block	If Memory Type is "Data Block", this will identify the DB number.				
Data Type	Data Type	Memory Space		Limits	
Туре	boolean	1 bit data		01	
	byte	8-bit data		-128 127	
	short	16-bit data		-32768 32767	
	int	32-bit data		-2.1e9 2.1e9	_
	unsignedByte	8-bit data		0 255	
	unsignedShort	16-bit data		0 65535	
	unsignedInt	32-bit data		0 4.2e9	
	float	IEEE single-precisio	n	1.17e-38 3.40e38	
		32-bit floating point t	уре		
	string	Refer to "String data	type chan	nel"	
	Note: to define like "byte[]", "sl	arrays, select one of hort[]"…	Data Type	format followed by square b	rackets
Arraysiz e	 In case of array ta In case of string ta in the string tag. 	g, this property repres ig, this property repres	ents the nusents the m	umber of array elements. naximum number of bytes av	ailable

Element	Description					
	Note: number of byt to UTF-8 or Latin1 ir If Encoding property requires 2 bytes.	ytes corresponds to number of string characters if Encoding property is set I in Tag Editor. rty is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character				
Conversi	Conversion to be ap	pplied to the tag.				
on	Conversion					
	inv,swap2	Allowed Configured				
		BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
	Depending on data type selected, the Allowed list shows one or more conversions, listed below.					
	Value	Description				
	Inv bits	Invert all the bits of the tag.				
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)				
	Negate	Set the opposite of the tag value.				
		<i>Example:</i> 25.36 → -25.36				
	AB -> BA	Swap nibbles of a byte.				
		Example: $15D4 \rightarrow 514D$ (in hexadecimal format) $5588 \rightarrow 20813$ (in decimal format)				
	ABCD -> CDAB	Swap bytes of a word.				
		<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)				
	ABCDEFGH -> GHEFCDAB	Swap bytes of a double word. <i>Example:</i> $32FCFF54 \rightarrow 54FFFC32$ (in hexadecimal format)				

lement	Description				
	Value	Description			
		$855441236 \rightarrow 1426062386$ (in decimal format)			
	ABCNOP ->	Swap bytes of a long word.			
	OPMDAB	Example: $142.366 \rightarrow -893553517.588905$ (in decimal format) 0.10000000110 0001110010111011001000101101000011100101			
	BCD	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)			
		Example: $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)			
	S5timer(BCD)	Used to support S5timer. Check Simatic S5timer special data type for more details.			
	S5timer(BIN)	Legacy transformation for S5timer in binary format.			
	Select the conversi Configured list.	on and click on plus button. The selected item will be added on			
	If more conversions Configured list).	s are configured, they will be applied in order (from top to bottom of			

Use the arrow buttons to order the configured conversions.

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names have to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

+ - 2 🖻 🖻 >] [> 🗞	■ 3 🚮 R	₽• Search	🛛 🍸 Filter b	by: Data	~	Items used:8/100	00 Protocol: Show all
A Simatic S7 MPI:prot1	Type Container	Tag name		Tag URI			Prop
Slave id: Alias: Node 1 Model:	Container	_			-		
- America	allasti	Teacher Casses		1 Conner	SPECIAL CO.		
- In Manager	These -	There is a second		1210/0210	warnet i an et Maa	÷	
 Mod., Physical Conditional Co	anagradia	HALF PLI PRIMA	Import Tag(s)	/ - Network			×
Ref. 2002/06-0648 Total Look Conc. C	anapadita anapadita	<pre>internet internet.com inte</pre>	Please sele	ect one or more	e nodes to ass	sociate this dictiona	ry and continue.
Water_Level	unsignedSho	rt Node 17Water_Level	Slave id I	Model	Alias		
Alias: Node2 Model:	Container		1276.8.2	inal	Node1		
			107968.0	bradi	Node2		
			For each select	ted node impo	t only symbol	s matching that noc	le

Note: Aliasing tag names are only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If Alias string is modified after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

String data type

ij.

In Simatic S7 PLC it's possible to define two different types of tags to manage string variables.

- as Array [1..xx] of Chars.
- as String[xx].

Step7 string declaration is showed in the following figure:

ľ	Address	Name	Туре	Initial value	Comment	S7 String	
	0.0		STRUCT			orounig	
I	+0.0	Stringl	STRING[254]	'sample'			
I	+256.0	String2	ARRAY[110] 🛛 🗧				
I	*1.0		CHAR			String as array of char	
	=266.0		END_STRUCT				

TIA Portal string declaration is showed in the following figure:

101	≝ 🔹 💀 課 🐘 & B 🖻 🔢 🍄 Data_block 1				S7 String				
1	-	Na	- — ime Static	Data type	Uffset	Start value	Retain	Accessible	Visible in
2 3	- -		String1 String2 	String Array [1 10] of Char		'sample'			
						String as a	rray of cha	ar	



Note: Usage of String[xx] data type is allowed but a specific Conversion must be applied to the tag. Anyway using tag importer to import tag dictionary from TIA Portal or Step7 string tags are automatically configured and no changes/conversion are needed.

To manually add an "Array [1..xx] of Chars" data type tag, press the [+] button in the Tag Editor, then select "string" as Data Type of the Tag and type the string length in the "Arraysize" field:

Simatic S7 MPI		×
Simatic S7 MPI		
Memory Type Internal Memory V	Offset SubIndex	
Data block	Data Type Arraysize string 10	
Conversion	+/-	
	OK Cancel Apply	Help

and confirm with OK button.

To manually add a "String[xx]" data type tag, press the [+] button in the Tag Editor, then select "string" as Data Type of the Tag and type the string length in the "Arraysize" field,

Simatic S7 MPI	×
Simatic S7 MPI	
Memory Type Offset Internal Memory ~ 114	SubIndex
Data block Data T	iype Arraysize
Conversion	
	OK Cancel Apply Help

then click on [+/-] button to open the Conversion dialog.

Simatic S7 MPI					×
Simatic S7 MPI					
Memory Type Internal Memory	Offset	SubIndex			
Data block	Data Type	~	Arraysize		
Conversion	+/-				
	(ж	Cancel	Apply	Help

Into conversion dialog:

- select the "S7 String" conversion type
- click on [+] button to add the conversion.

imatic S7 MPI			×
Simatic S7 MPI			
Memory Type	Offset	SubIndex	
Data block	Data Type	Arraysize	
Conversion	В		
A	Allowed AB->BA ABCD->CDAB S7 String	S7 String	
		Cancel OK	Help

The conversion will be listed into the Configured window on the right.

Confirm with OK button.

Simatic S5timer data type

Simatic drivers support a special data type, called S5Timer.

The tag must be configured with a specific data type and a conversion must be applied to the Tag to correctly read/write a Simatic S5Timer Variable.

Open the Tag Editor and add a Tag pressing the Plus button.



Select "unsignedInt" as Data Type of the Tag.

Simatic S7 MPI		×
Simatic S7 MPI		
Memory Type Offs Internal Memory V 50	set SubInde>	<
Data Block	Data Type	Arraysize
1	unsignedInt 🗸 🗸	0
Conversion +/-	boolean byte short int unsignedByte unsignedShort unsignedInt float double string boolean [byte] short [int] unsignedByte] unsignedShort [Cancel Apply Help
	unsignedShort [] unsignedInt [] float [] double [] wstring	



Simatic S7 MPI		×
Simatic S7 MPI		
Memory Type	Offset SubIndex	
Data Block	Data Type Arraysiz	e
Conversion		
	OK Cancel	Apply Help

In the Conversion dialog select the S5timer(BCD) conversion type [A] then click on Plus button [B] to add the conversion, the configured conversion will be listed into the Configured window on the right. Then confirm with OK.

matic S7 MPI	N	23
Simatic S7 MPI		
Memory Type	Offset SubIndex	
Data Block	Data Type Arraysize	
	B	
	Allowed Configur S5timer (BCD) S5timer (BIN)	red (BCD)
		Help
	Cancel	ОК

Tag Import

Select the driver in Tag Editor and click on the Import Tags button to start the importer.

	Tags	×								
+	—	X	đ	ß	>]	₽	A 9B	B>	ŧ i ł	1
Data	9		^		-	Та	g URI			

The following dialog shows which importer type can be selected.

Version	Туре		
TIA Portal Project v12 or newer	Linear		
TIA Portal v13, v14 or newer	Linear		
TIA Portal v10, v11, v12	Linear		
Step7	Linear		
Tag Editor exported xml	General		

Importer	Description
TIA Portal Project v12 or newer Linear	Allows to import the whole TIA Portal project file using .apxx file (where "xx" is the TIA Portal version, example: for TIA Portal 13, file name is "project.ap13"). All variables will be displayed at the same level.
TIA Portal v13, v14 or newer Linear	Allows to import: Program blocks using .db file PLC tags using .xlsx file PLC data types using .udt file Check Export using TIA Portal v13, v14 or newer for more details. All variables will be displayed at the same level.
TIA Portal v10, v11, v12 Linear	 Allows to import: Program blocks using .tia file PLC tags using .xlsx file PLC data types using .scl file Check Export using TIA Portal v10, v11, v12 for more details. All variables will be displayed at the same level.

Importer	Description	
Step7 Linear	 Allows to import: Symbols table .asc file Sources using .awl file Check Export using STEP7 for more details. All variables will be displayed at the same level. 	
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.	

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols					•
+ - 👗 🕲 🖉 🔰	() 🗞 🕅 🕅	R 🔎 - Search 🍸 Filter by: Data	▼ Ite	ms used:6/10000 Protocol: Show all	Show all tags 🔅 🗖
Data	Туре	Comment	^	Property	Value
Modbus TCP:prot1	Container			✓ Driver	
Model: Modicon Modbus(1-based)	Correlation			Model	Modicon Modbus(1-based)
Holding Registers 1	unsignedShort			Protocol	Modbus TCP:prot1
Holding Registers 2	unsignedShort			✓ Dictionary	
- MRTU1	unsignedShort			Array	false
- MRTU2	unsignedShort			Array size	0
MRTU3	unsignedShort			Arrayindex.Subindex	400003
- MRTU4	unsignedShort			Comment	
- MRTU5	unsignedShort			Data type	unsignedShort
- Holding Registers 3 - MRTU1 - MRTU2 - MRTU3 - MRTU4 - MRTU5	unsignedShort unsignedShort unsignedShort unsignedShort unsignedShort		-	Dictionary Array Array size Arrayindex.Subindex Comment Data type	false 0 400003 unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
樹	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag.
	Example of both checked and unchecked result:

Toolbar item	Description
	Tage* x Tage* x
P- Search	Searches tags in the dictionary basing on filter combo- box item selected.

Communication status

The communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Invalid response	The device did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General Error	Error cannot be identified; should never be reported; contact technical support

System Variables

System Variables communication driver allows to create Tags that point to system information.

Refer to <u>System Variables > Protocol</u> chapter of User's Manual.

Protocol Editor Settings

System Variables communication driver allows to create Tags that point to system information.

Refer to <u>System Variables > Protocol</u> chapter of User's Manual.

Uni-Telway

Uni-Telway is a field bus used to communicate between devices of the same type according to a protocol defined by Schneider Electric.

The physical access is based on a Serial Link transmission (half-duplex type). The electrical interface allows multi-point mode connection.

Numerous proprietary or third-party devices can be used on this bus, which has become one of the industry standards.

The operator panels can be connected to a Uni-Telway controller using this communication driver.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called "Uni-Telway" from the list of available protocols.

The driver configuration dialog is shown in figure.

Uni-Telway		×
PLC Network	Comm	OK
Alias		Cancel
panelNode	1	
Multiple Connections		
Retry count	3	
Network	0	
Station	254	
Gate	5	
Module	254	
Channel	65	
PLC Models		
TSX17-20	A]
TSX47		
TSX07 Nano		
TSX37 Micro		
TSX57 Premium	~	

Element	Description
panelNode	Node of the panel into the Uni-Telway network.
Multiple Connections	Not used. Available for future implementation.
Retry count	This parameter defines the number of times a certain message will be sent to the

Element	Description		
	controller before reporting the communication error status.		
	A value of 1 for the parameter "Retry count" means that the panel will eventually report the communication error status if the response to the first request packet is not correct.		
Network, Station, Gate, Module, Channel	Controller's parameters as defined into controller's programming tool.		
PLC Models	The driver supports communication with different controllers. Please check directly in the programming IDE software for a complete list of supported controllers.		



Element	Description			
PLC Network	The protocol allows the con set-up multiple connections controllers.	nection of multiple controllers to one operator panel. To s, check "PLC network" checkbox and configure all		
	Uni-Telway	×		
	PLC Network	n		
	Alias	Cancel		
	panelNode 1			
	Multiple Connections			
	Retry count 3			
	Network 0	×		
	Station 25	OK Carrel		
	Gate 5			
	Module 25			
	Channel 65	254		
	PLC Models	5		
	TSX47 Module TSX47-1 TSX07 Nano Channa	254		
	TSX37 Micro TSX57 Premium			
	Slaves PLC Mo	dels		
	Slave Id Model TSX17 TSX47 TSX47	-20		
	TSX07 TSX37 TSX37	Nano Micro Perenium		
Comm	Gives access to the serial p figure:	port configuration parameters as shown in the following		
	Comm Parameter Dialog	×		
		ОК		
	Port	com1 🔹		
	Baudrate	9600 💌		
	Parity	odd 🗸		
	Data bits	8		
	Stop bits	1		
	Mode	RS-232		

t Description	Description				
Element	Descript	Description Serial port selection:			
Port	Serial po				
		Series 400	Series 500/600		
	com1	PLC Port	Onboard Serial Port		
	com2	PC/Printer Port	Optional Module on slot #1 or #2		
	com3	Not available	Optional Module on slot #3 or #4		
Baud rate, Parity, Data bits, Stop bits	Commun communi	Communication parameters for serial communication			
Mode	Serial po • R • R • R	Serial port mode; available options: • RS-232, • RS-485 (2 wires) • RS-422 (4 wires)			

Tag Editor Settings

Into Tag editor select the protocol "Uni-Telway" from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:

Memory Type	Offset	Subindex		
Input Bit	▼ 0			
Station	Module		Data Type	
			boolean	•
Arraysize Co	nversion			
0		+/-		

Element	Description		
Memory Type	Memory resource where tag is located.		
Offset	Offset address where tag is located.		
SubIndex	This allows resource offset selection within the register.		
Station	Station number.		
	Property available only for Memory Type "Common Word".		
Module	Module number.		
	Property available for Memory Type:		
	Input Bit		
	Output Bit		
	Input Word		
	Output Word		

Element	Description			
Data Type	Data Type	Memory Space	Limits	
	boolean	1 bit data	01	
	byte	8-bit data	-128 127	
	short	16-bit data	-32768 32767	
	int	32-bit data	-2.1e9 2.1e9	
	unsignedByte	8-bit data	0 255	
	unsignedShort	16-bit data	0 65535	
	unsignedInt	32-bit data	0 4.2e9	
	float	IEEE single-precision	1.17e-38 3.40e38	
		32-bit floating point type		
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes. 			
Conversion	Conversion to be applied to the tag.			
	Conversion			
	inv,swap2 Depending on data typ	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits	Cancel OK	
	Depending on data typ	e selected, the list Allowed sho	ows one or more conversion types.	
Element	Description			
---------	----------------	--	--	
	Value	Description		
	Inv bits	inv: Invert all the bits of the tag.		
		<i>Example:</i> 1001 \rightarrow 0110 (in binary format) 9 \rightarrow 6 (in decimal format)		
	Negate	neg: Set the opposite of tag value.		
		<i>Example:</i> 25.36 → -25.36		
	AB -> BA	swapnibbles: Swap nibbles in a byte.		
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)		
	ABCD ->	swap2: Swap bytes in a word.		
	CDAB	<i>Example:</i> 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)		
	ABCDEFGH	swap4: Swap bytes in a double word.		
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)		
	ABCNOP -	swap8: Swap bytes in a long word.		
	> OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 1000000110 0001110010111011010000101101000011100101		
		→ 1 10000011100 1010101000010100010110110110010110110000		
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9)		
		Example: $23 \rightarrow 17$ (in decimal format) $0001 \ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)		

Element	Description
	Select conversion and click +. The selected item will be added to list Configured .
	If more conversions are configured, they will be applied in order (from top to bottom of list Configured).
	Use the arrow buttons to order the configured conversions.

Variables

Variables communication driver allows to define Tags which points to HMI internal memory.

Variables Tags are not retentive: when the project starts, the starting value of any Variables Tag is 0 (or "" in case of string Tag).



Variables communication driver is not counted as physical protocol.

Refer to Table of functions and limits from main manual in "Number of physical protocols" line.

Protocol Editor Settings

Adding a protocol

To configure the protocol:

- 1. In the **Config** node double-click **Protocols**.
- 2. To add a driver, click +: a new line is added.
- 3. Select the Variables protocol from the PLC list.

Tag Editor Settings

Path: ProjectView> Config > double-click Tags

- 1. To add a tag, click +: a new line is added.
- 2. Select Variables from the protocol list: tag definition dialog is displayed.

Data type	Arraysize	Conversion	
boolean	•		+/-

Element	Description					
Data Type	Data Type	Memory Space	Limits			
	boolean	1-bit data	0 1			
	byte	8-bit data	-128 127			
	short	16-bit data	-32768 32767			
	int	32-bit data	-2.1e9 2.1e9			
	int64	64-bit data	-9.2e18 9.2e18			
	unsignedByte	8-bit data	0 255			
	unsignedShort	16-bit data	0 65535			
	unsignedInt	32-bit data	0 4.2e9			
	uint64	64-bit data	0 1.8e19			
	float	IEEE single-precision 32-bit floating point type	1.17e-38 3.4e38			
	double	IEEE double-precision 64-bit floating point type	2.2e-308 1.79e308			
	string	Array of elements containing character code defined by selected encoding				
	binary	Arbitrary binary data				
	Note: to define arrays. select one of Data Type format followed by square brackets like "byte[]", "short[]"					
Arraysize	 In case of array tag, this property represents the number of array elements. In case of string tag, this property represents the maximum number of bytes available in the string tag. 					
	Note: number of bytes corresponds to number of string characters if Encodir set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE character requires 2 bytes.					
Conversion	Conversion to be applied to the tag.					

Element	Description					
	Conversion					
	inv,swap2	Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Cancel OK				
	Depending on data type selected, the list Allowed shows one or more conversion types					
	Value	Description				
	Inv bits	inv : Invert all the bits of the tag.				
		Example: $001 \rightarrow 0110$ (in binary format) $0 \rightarrow 6$ (in decimal format)				
	Negate	neg: Set the opposite of tag value.				
		<i>Example:</i> 25.36 → -25.36				
	AB -> BA swapnibbles: Swap nibbles in a byte.					
		<i>Example:</i> 15D4 \rightarrow 514D (in hexadecimal format) 5588 \rightarrow 20813 (in decimal format)				
	ABCD -> swap2: Swap bytes in a word.					
	CDAB	Example: 9ACC \rightarrow CC9A (in hexadecimal format) 39628 \rightarrow 52378 (in decimal format)				
	ABCDEFGH	swap4: Swap bytes in a double word.				
	-> GHEFCDAB	<i>Example:</i> 32FCFF54 \rightarrow 54FFFC32 (in hexadecimal format) 855441236 \rightarrow 1426062386 (in decimal format)				
	ABCNOP -	swap8: Swap bytes in a long word.				
	> OPMDAB	Example: 142.366 → -893553517.588905 (in decimal format) 0 1000000110 0001110010111011001000101101000011100101				

Element	Description			
	Value	Description		
		0001 \rightarrow 1 10000011100 1010100001010001011011011001011011000010011 1101 (in binary format)		
	BCD	bcd : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> $23 \rightarrow 17$ (in decimal format) $0001\ 0111 = 23$ 0001 = 1 (first nibble) 0111 = 7 (second nibble)		
	Select conversion	on and click +. The selected item will be added to list Configured .		
	If more conversions are configured, they will be applied in order (from top to bottom o Configured).			
	Use the arrow buttons to order the configured conversions.			

Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The system will require a generic XML file exported from Tag Editor by appropriate button.

	Tags	×								
+	-	×	ß	ß	>]	Þ	A 9B	B>	ŧ3	1
Data			^			Та	g URI			

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

Tags × Protocols			•
🛨 — 👗 🕲 🖄 🔰 [> 🗞 (3) 🙀 R 🔎 Search 🍸 Filter by: Data	a ▼ <i>Items used:6/10000</i> Protoco	l: Show all 🛛 🗹 Show all tags 🔅 🗌
Data Type	Comment	^ Property	Value
Modbus TCP:prot1 Container		✓ Driver	
Model: Modicon Modbus(1-based)		Model	Modicon Modbus(1-based)
Holding Registers 1 unsignedSt	nort	Protocol	Modbus TCP:prot1
Holding Registers 2 unsignedSh	nort	× Dictionary	
Holding Registers 3 unsignedSh	nort	- Dicubriary	feler.
MRTU1 unsignedSh	nort	Array	Taise
- MRTU2 unsignedSh	nort	Array size	0
MRTU3 unsignedSh	nort	Arrayindex.Subin	dex 400003
MRTU4 unsignedSh	nort	Comment	
- MRTU5 unsignedSh	nort	Data type	unsignedShort

Toolbar item	Description
	Import Tag(s).
	Select tags to be imported and click on this icon to add tags from tag dictionary to the project
₩ ki ki ki ki ki ki ki ki ki ki	Update Tag(s).
	Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:
P → Search Y Filter by: Tag name →	Searches tags in the dictionary basing on filter combo- box item selected.



Exor International S.p.A. Via Monte Fiorino, 9 37057 San Giovanni Lupatoto (Verona) Italy

> info@exorint.com **phone**: +39 045 8750404 **fax**: +39 045 8779023

Communication Protocols User Manual

2022-05-18

Copyright © 2010-2022