Write acyclic data to IO-Link devices via RCCA

Contents

Applies to the project "1200_RCCA-D_IOL_Parameterisation_via_Index". The manual provided by the manufacturer serves as a reference for indices and instructions for the "Werma ClearSIGN compact" demonstration hardware (download: https://www.werma.com/de/s_c1510i2688/ClearSIGN_BM_24VDC_MC/65610002.html)

FEIG ELECTRONIC GmbH, Industriestrasse 1a, D-35781 Weilburg
Tel.: +49 6471 3109-0, Fax: -99, info@feig.de, www.feig.de

Creating the hardware configuration

An RCCA-D variant is used in the example project. Apart from the first IO-Link port, no other features are required here and are therefore not parameterised.

First, a suitable CPU is integrated into the project. An S7-1212FC is used in the example. However, an F-CPU is not absolutely necessary for the demonstration, as safe programme processing is not demonstrated. However, it is important to define the setting under "Start-up".

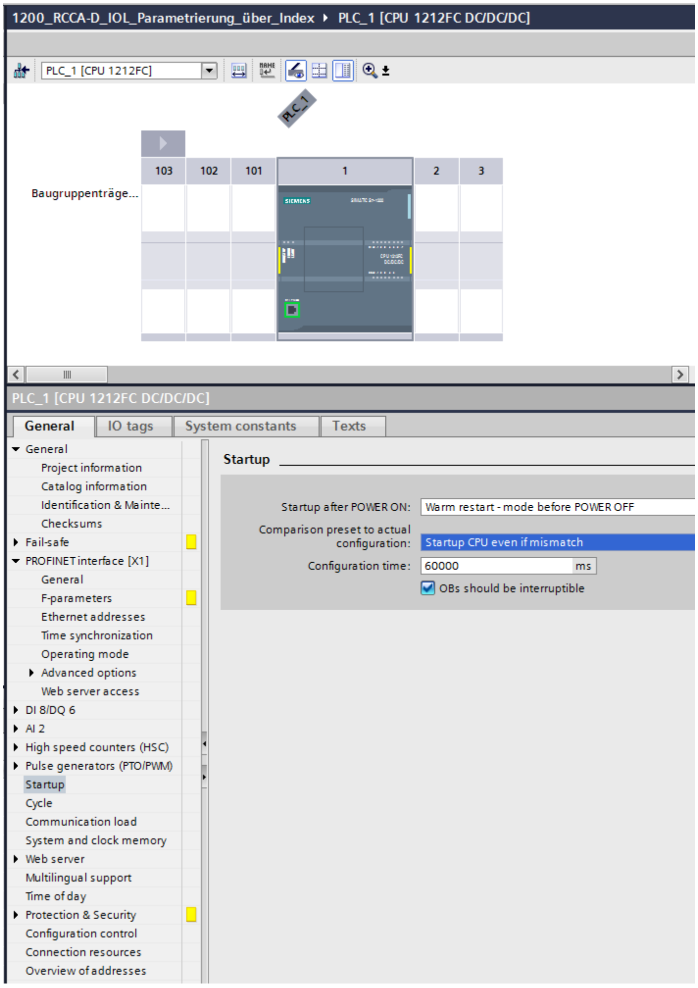


Figure 1: Hardware configuration S7-1212F

The setting "Start-up of CPU even with differences" also allows a demonstration without all components being connected to the RCCA/PLC.

After the RCCA module has been integrated into the hardware configuration - here TST-RCCA-D with the GSDML 20210906 - all unnecessary IO-Link modules are removed for demonstration purposes and the CRC is set for the default configuration. In order to be able to use the full output power of the IO-Link port, the "IQ behaviour" of the corresponding port in the IO-Link master module is set to "Digital output".
The RCCA and PLC are then assigned to a common network.

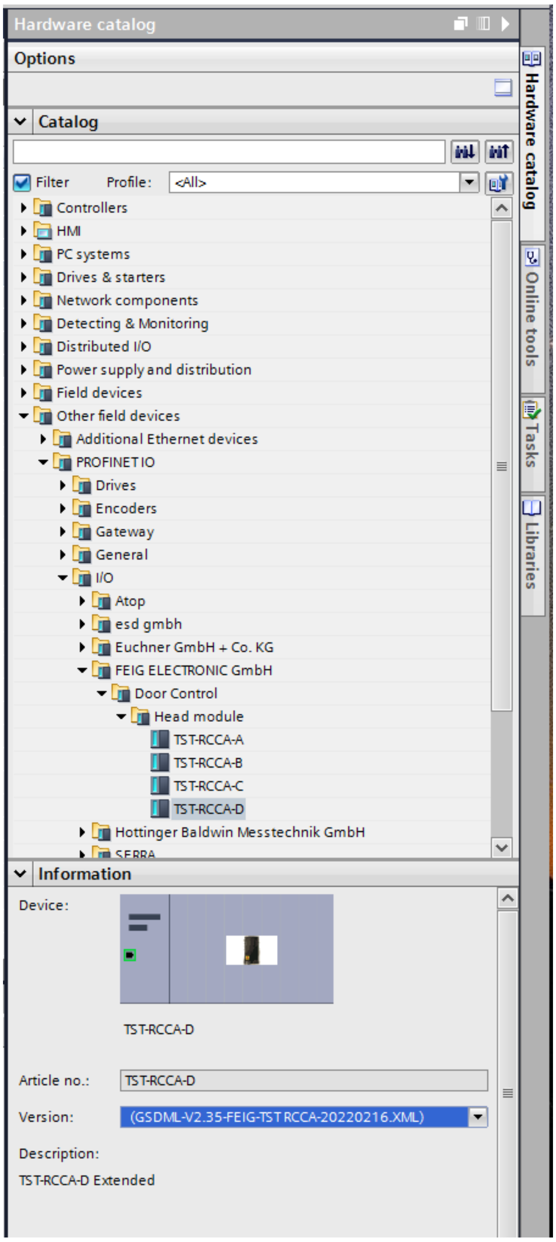


Figure 2: Hardware catalogue

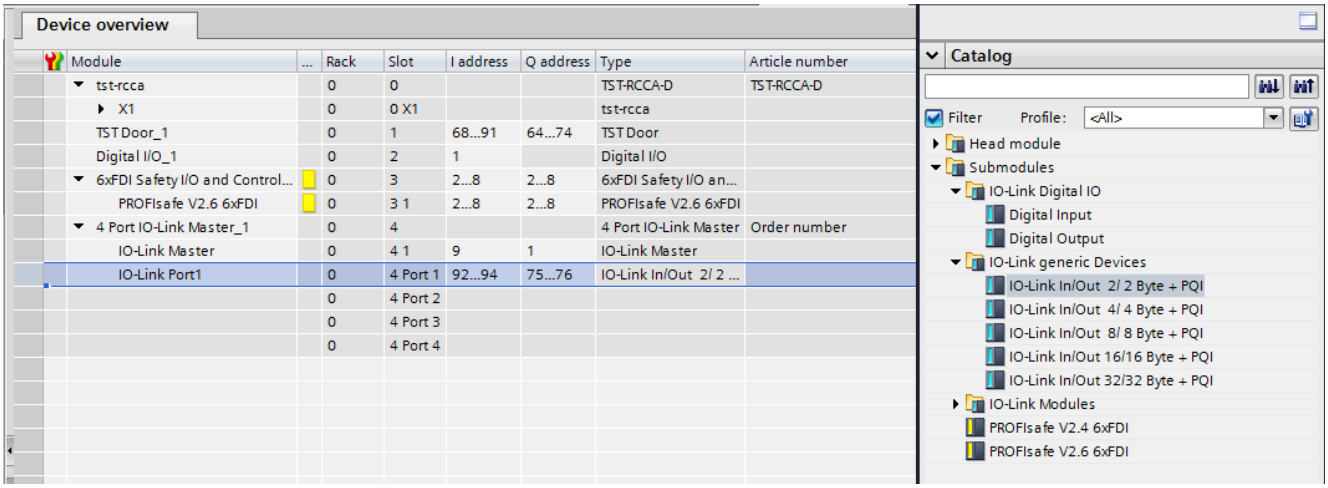


Figure 3: IO-Link port configuration

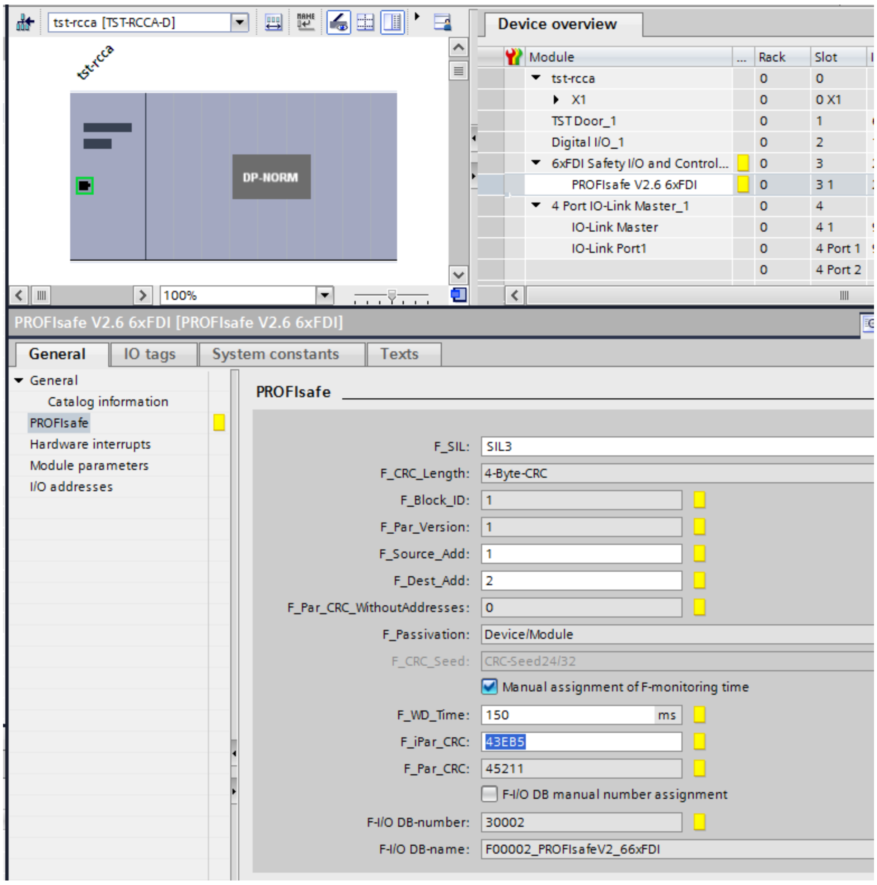


Figure 4: F-CRC default configuration

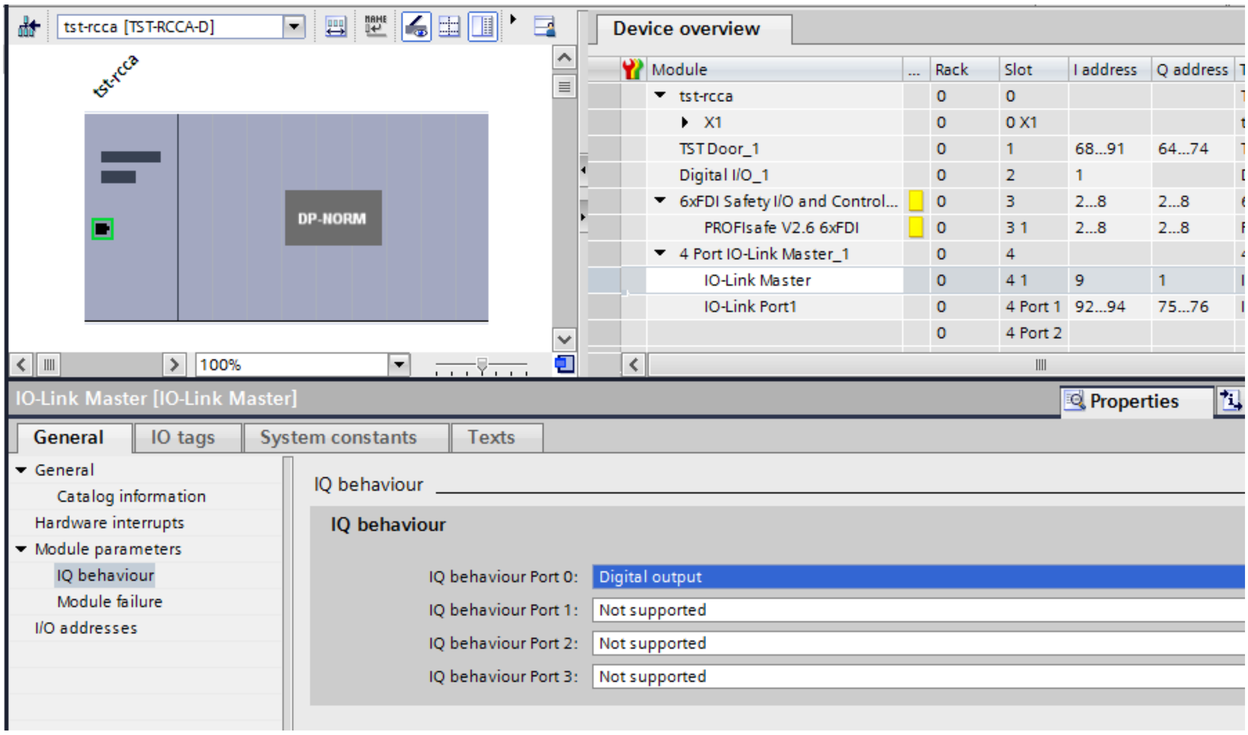


Figure 5: IQ behaviour in master module

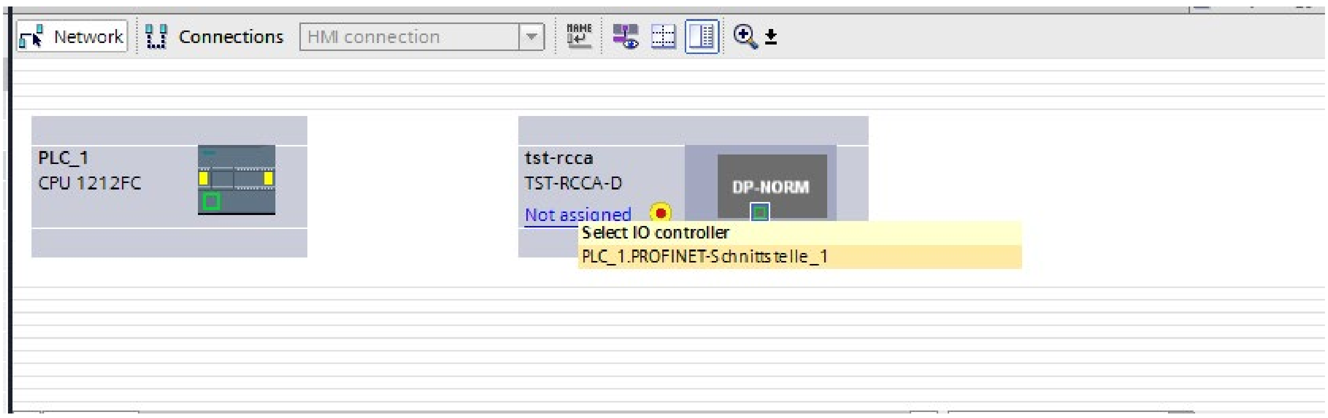


Figure 6: Establishing a network connection

Finally, the hardware configuration is translated and transferred to the target device.

FEIG

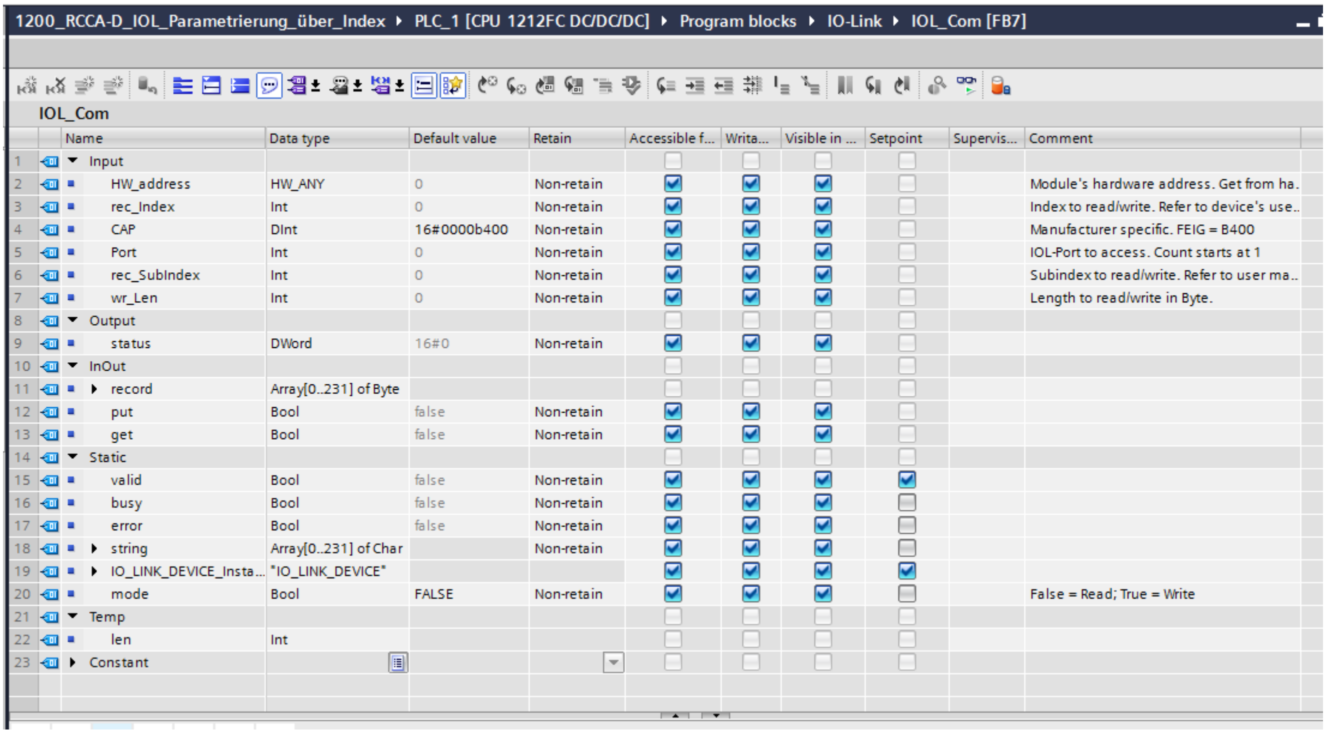
Creating the test program

IO-Link library

To be able to use IO-Link technology with Siemens components, at least TIA Portal V15.0 is required. All the modules required for this can be found in the library "IO_LINK_Library V5.2" or V6.0 for TIA V15 or V16. (https://support.industry.siemens.com/cs/document/82981502/bibliothek-f%C3%BCr-io-link-(iolink)?dti=0&lc=en-WW)
The "IO_LINK_DEVICE V3.3" module is transferred from this library to the project.

The remaining program

To simplify the use of the FB, a further FB "IOL_Com" is created which is structured as follows:



1200_RCCA-D_IOL_Parametrierung_über_Index ▸ PLC_1 [CPU 1212FC DC/DC/DC] ▸ Program blocks ▸ IO-Link ▸ IOL_Com [FB7]

IOL_Com

	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Input				☐	☐	☐	☐		
2	HW_address	HW_ANY	0	Non-retain	☑	☑	☑	☐		Module's hardware address. Get from ha...
3	rec_Index	Int	0	Non-retain	☑	☑	☑	☐		Index to read/write. Refer to device's use...
4	CAP	DInt	16#0000b400	Non-retain	☑	☑	☑	☐		Manufacturer specific. FEIG = B400
5	Port	Int	0	Non-retain	☑	☑	☑	☐		IOL-Port to access. Count starts at 1
6	rec_SubIndex	Int	0	Non-retain	☑	☑	☑	☐		Subindex to read/write. Refer to user ma...
7	wr_Len	Int	0	Non-retain	☑	☑	☑	☐		Length to read/write in Byte.
8	▼ Output				☐	☐	☐	☐		
9	status	DWord	16#0	Non-retain	☑	☑	☑	☐		
10	▼ InOut				☐	☐	☐	☐		
11	▸ record	Array[0..231] of Byte			☐	☐	☐	☐		
12	put	Bool	false	Non-retain	☑	☑	☑	☐		
13	get	Bool	false	Non-retain	☑	☑	☑	☐		
14	▼ Static				☐	☐	☐	☐		
15	valid	Bool	false	Non-retain	☑	☑	☑	☑		
16	busy	Bool	false	Non-retain	☑	☑	☑	☐		
17	error	Bool	false	Non-retain	☑	☑	☑	☐		
18	▸ string	Array[0..231] of Char		Non-retain	☑	☑	☑	☐		
19	▸ IO_LINK_DEVICE_Insta...	"IO_LINK_DEVICE"			☑	☑	☑	☑		
20	mode	Bool	FALSE	Non-retain	☑	☑	☑	☐		False = Read; True = Write
21	▼ Temp				☐	☐	☐	☐		
22	len	Int			☐	☐	☐	☐		
23	▸ Constant				☐	☐	☐	☐		

Figure 7: IOL_Com IO range

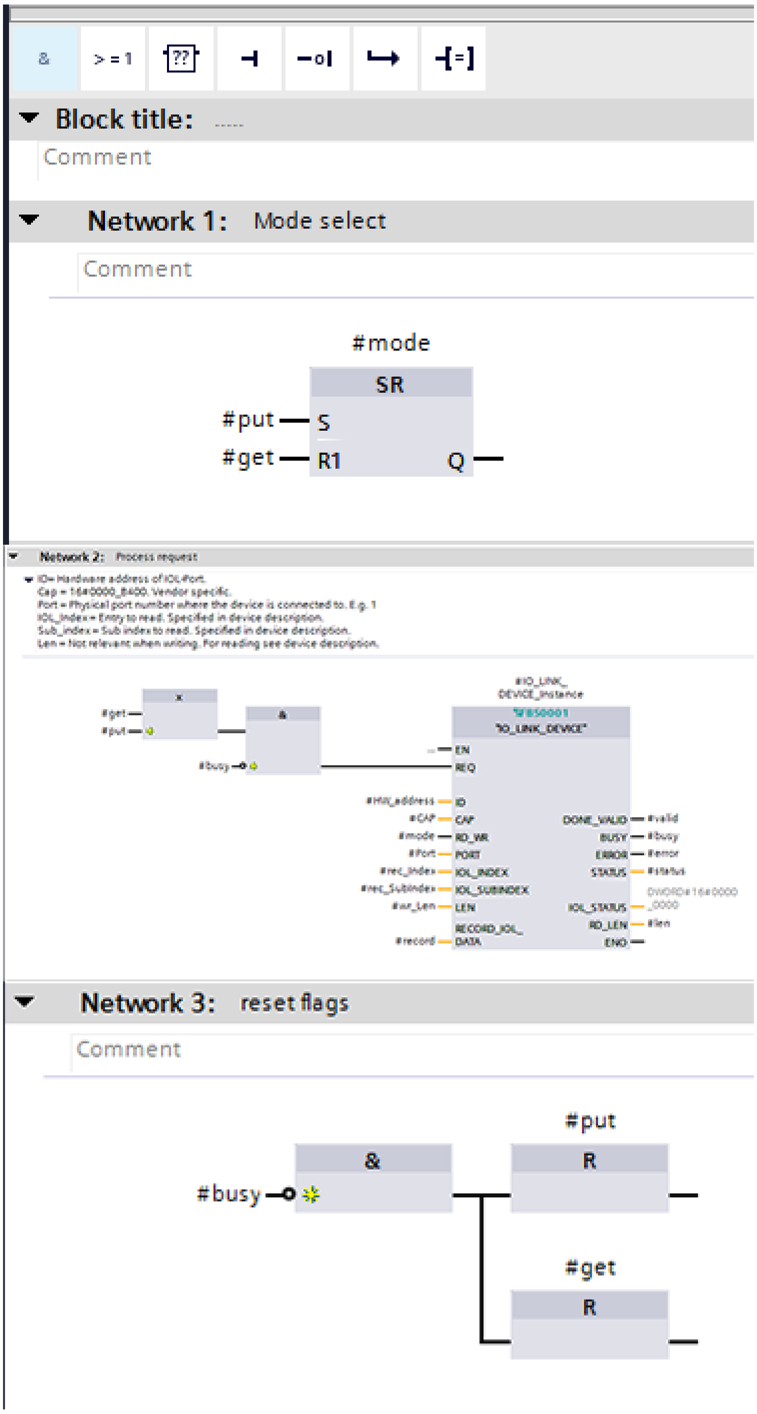


Figure 8: IOL_Com program sequence

To store the user data, the target for #record is now required, which is created as a DB without optimised memory access and contains a byte array with 232 entries.

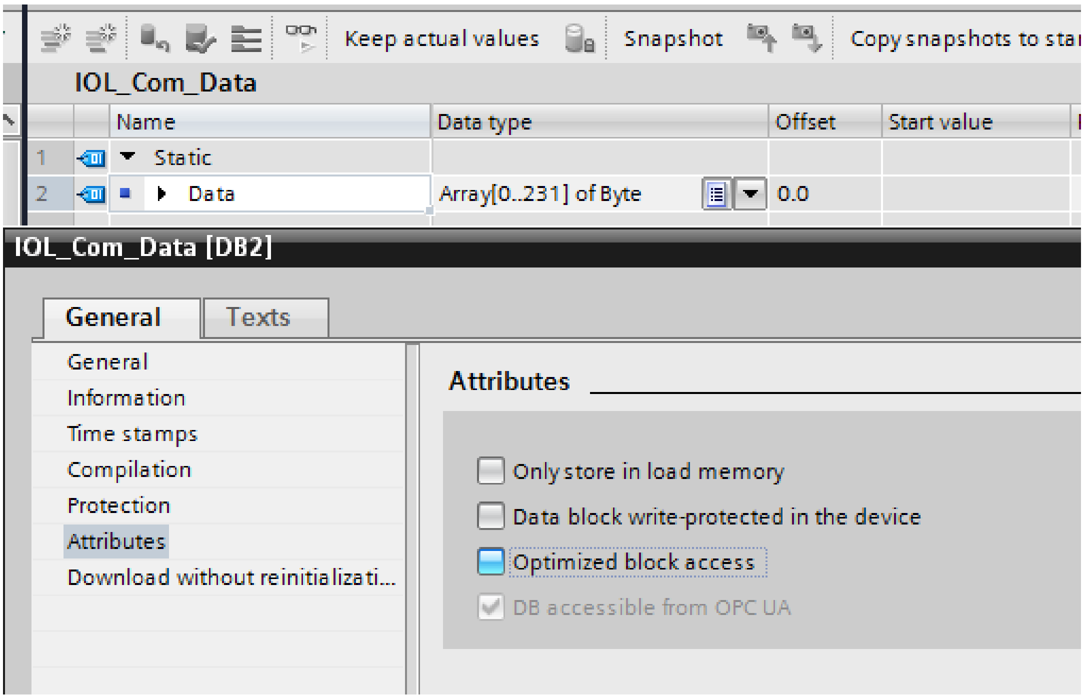


Figure 9: IOL data storage without optimisation

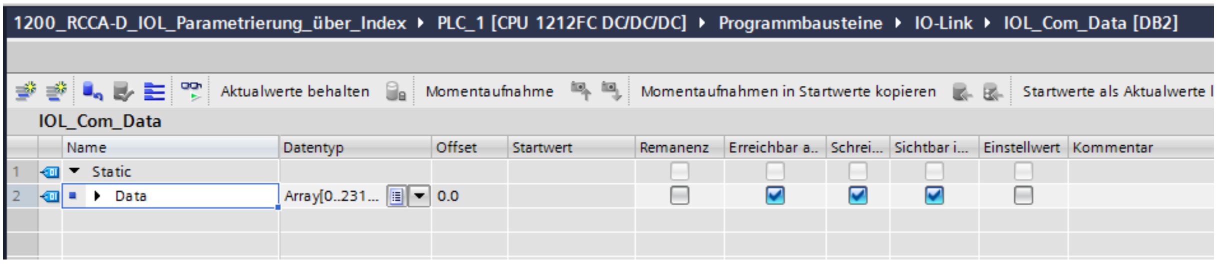


Figure 10: IOL data storage data view

Once this configuration has been finished, the program is completed by calling up the FB created in the main program.

For demonstration purposes, a simple variable table is created and assigned in the main programme.

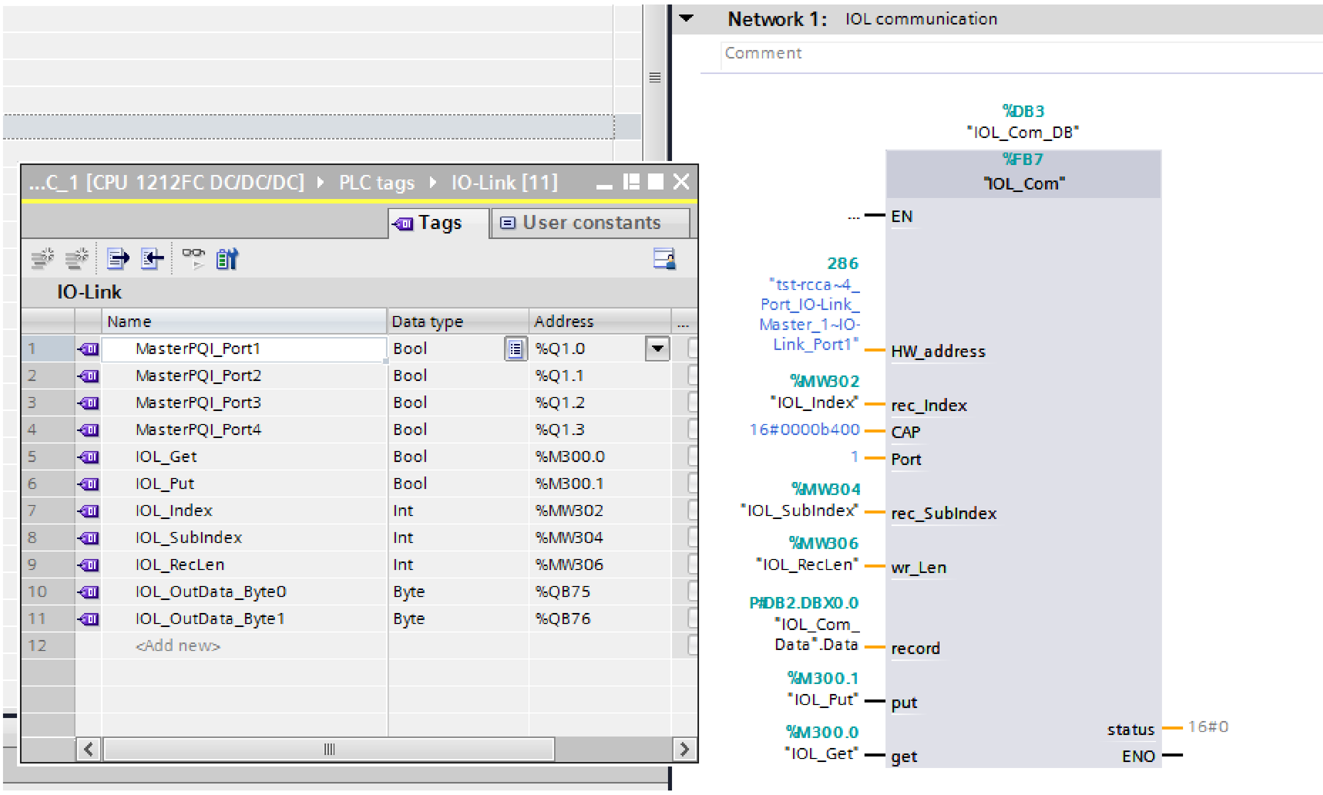


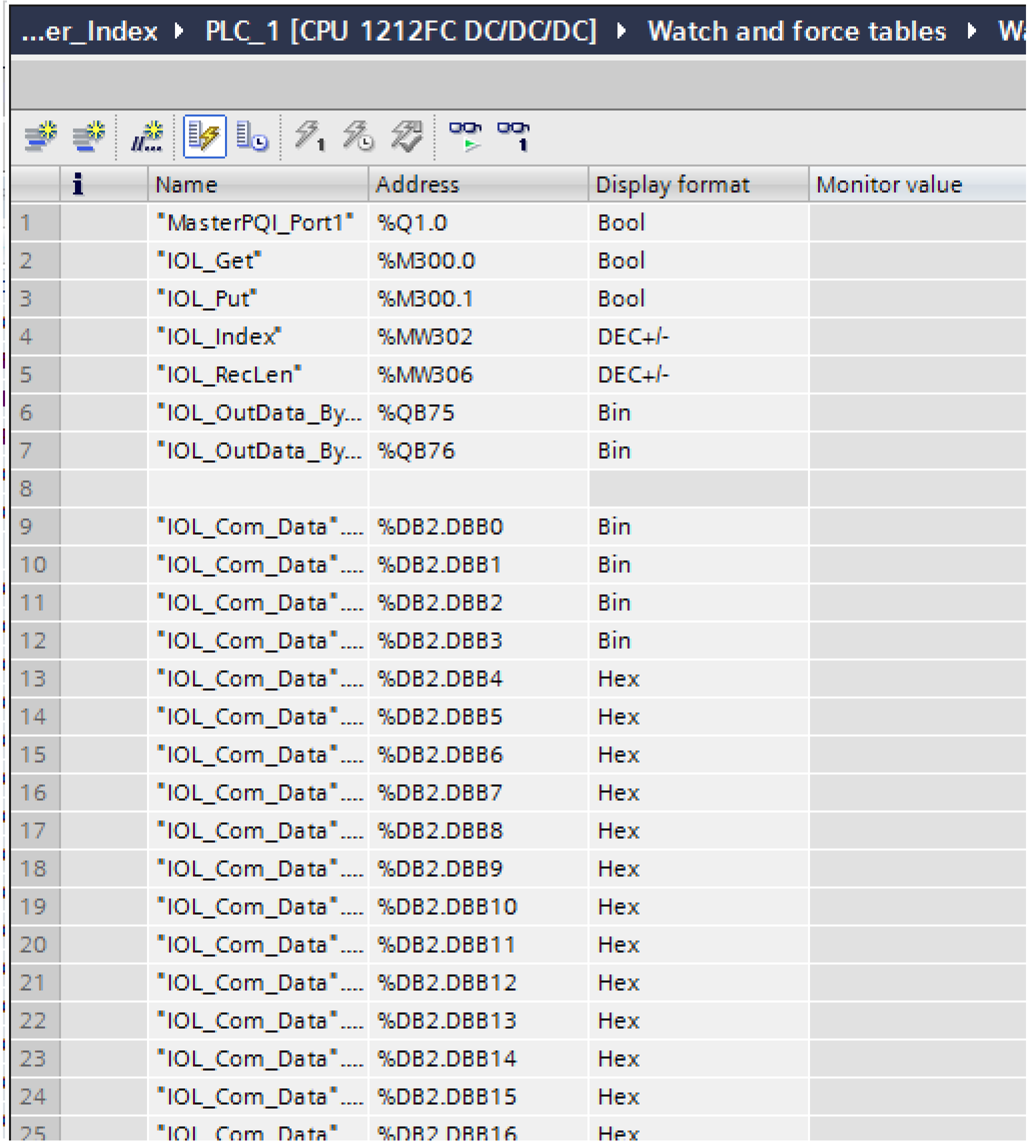
Figure 11: Interaction variables and call in OB1

The program is now translated and loaded into the target device. The program execution can be started.

Use the example program

Write the index

A monitoring and force table is now created to control the program sequence. All relevant variables are recorded there.



...er_Index ▸ PLC_1 [CPU 1212FC DC/DC/DC] ▸ Watch and force tables ▸ W

	i	Name	Address	Display format	Monitor value
1		"MasterPQI_Port1"	%Q1.0	Bool	
2		"IOL_Get"	%M300.0	Bool	
3		"IOL_Put"	%M300.1	Bool	
4		"IOL_Index"	%MW302	DEC+/-	
5		"IOL_RecLen"	%MW306	DEC+/-	
6		"IOL_OutData_By...	%QB75	Bin	
7		"IOL_OutData_By...	%QB76	Bin	
8					
9		"IOL_Com_Data"....	%DB2.DBB0	Bin	
10		"IOL_Com_Data"....	%DB2.DBB1	Bin	
11		"IOL_Com_Data"....	%DB2.DBB2	Bin	
12		"IOL_Com_Data"....	%DB2.DBB3	Bin	
13		"IOL_Com_Data"....	%DB2.DBB4	Hex	
14		"IOL_Com_Data"....	%DB2.DBB5	Hex	
15		"IOL_Com_Data"....	%DB2.DBB6	Hex	
16		"IOL_Com_Data"....	%DB2.DBB7	Hex	
17		"IOL_Com_Data"....	%DB2.DBB8	Hex	
18		"IOL_Com_Data"....	%DB2.DBB9	Hex	
19		"IOL_Com_Data"....	%DB2.DBB10	Hex	
20		"IOL_Com_Data"....	%DB2.DBB11	Hex	
21		"IOL_Com_Data"....	%DB2.DBB12	Hex	
22		"IOL_Com_Data"....	%DB2.DBB13	Hex	
23		"IOL_Com_Data"....	%DB2.DBB14	Hex	
24		"IOL_Com_Data"....	%DB2.DBB15	Hex	
25		"IOL_Com_Data"....	%DB2.DBB16	Hex	

Figure 12: Watch and force table

The process data of the IO-Link device can be written directly here. In the example, these are QB75 and QB76. In the standard setting of the ClearSIGN (lamp), this means

Process Data		Single Segment	RGB	Level dimmed	Level blinking
Byte 0	Bit 0	Segment 1 Red	Segment 1	A	A
	Bit 1	Segment 1 Green	Segment 2	n	n
	Bit 2	Segment 1 Blue	Segment 3	a	a
	Bit 3	Segment 2 Red	Segment 4	l	l
	Bit 4	Segment 2 Green		o	o
	Bit 5	Segment 2 Blue		g	g
	Bit 6	Segment 3 Red		Value (0..100%)	Value (0..100%)
	Bit 7	Segment 3 Green			
Byte 1	Bit 0	Segment 3 Blue			
	Bit 1	Segment 4 Red			
	Bit 2	Segment 4 Green			
	Bit 3	Segment 4 Blue			
	Bit 4				
	Bit 5				
	Bit 6				
	Bit 7	Akustik (optional)	Akustik (optional)	Akustik (optional)	Akustik (optional)

Figure 13: Excerpt of Werna ClearSIGN

(Unfortunately there is a bug in the implementation of the lamp. The sequence of the process data is reversed. To control the lamp, byte 0 from the documentation corresponds to IOL_OutData_Byte1 and vice versa.)
To change the parameterisation of the device, the previously created variables and the documentation provided by the device manufacturer are important. In the example, the operating mode of the segments is to be changed from the standard single segment mode to RGB mode.

4.3 Konfiguration der Clear*SIGN* über Indexparametrierung

Beschreibung der Parameter

Index	Parameter	Zugang	Byte Länge	Wert	
02	System Command	wo	1	130	Reset Factory Settings
16	Vendor Name	ro	48	WERMA Signaltechnik GmbH + Co. KG	
17	Vendor Text	ro	48	www.werma.com	
18	Product Name	ro	32	ClearSIGN compact	
19	Product ID	ro	16	656.100. …	
20	Product Text	ro	64	ClearSIGN compact / ClearSIGN compact Contin. tone	
21	Serial Number	ro	16	Not used	
22	Hardware Revision	ro	16	AB	
23	Firmware Version	ro	16	1.21	
24	Application Text	rw	32		
64	Operating Mode	rw	1	0	Single Segment Mode
				1	RGB Mode
				2	Level Meter Mode dimmed
				3	Level Meter Mode blinking
65	Appearance Single	rw	12	0	Continuous
				1	Blinking
				2	Flashing
				3	EVS
66	Intensity Single	rw	12	0..100	
69	Segment Color (wird bei Betriebsart RGB und Level Meter verwendet)	rw	5	0	Dark
				1	Red
				2	Green
				3	Yellow
				4	Blue
				5	Purple
				6	Cyan
				7	White
70	Appearance RGB (wird nur bei Betriebsart RGB verwendet)	rw	5	0	Continuous
				1	Blinking
				2	Flashing
				3	EVS
71	Intensity RGB	rw	5	0..100	
74	OperatingHours	ro	4		

The table shows that the index with number 64 must be written with a length of one byte. The variables created are assigned as follows:

IOL_Put = TRUE
IOL_Index = 64d

IOL_RecLen = 1
IOL_Com_Data.Data[0] = 1 (RGB mode)

After setting the variables once (button in the ribbon menu), the operating mode was changed.
To activate the segments, the process data in the range Q76.0 to Q75.3 is no longer relevant, but only Q76.0 to Q76.3. (note byte rotation)

Another example now changes the colour of the individual segments:

IOL_Put = TRUE
IOL_Index = 69d
IOL_RecLen = 5
IOL_Com_Data.Data[0] = 2d (Green)
IOL_Com_Data.Data[1] = 3d (Yellow)
IOL_Com_Data.Data[2] = 4d (Blue)
IOL_Com_Data.Data[3] = 5d (Purple)
IOL_Com_Data.Data[4] = 0 (Only relevant with acoustic element)

It is essential to ensure that the variable IOL_RecLen matches the specified length (here 5). Otherwise, the write operation will fail.
After setting the variables once, the colour change is immediately visible.

Read index

Reading the entries works in the same way as writing them.
Index 20 "Product text" is selected here as an example.
The variables are written as follows:

IOL_Get = TRUE
IOL_Index = 20d
After writing to the variables once (button in the ribbon menu), the first 64 bytes are written to the DB IOL_Com_Data. To simplify matters, the display can be set to "characters" here.
It is not necessary to write to the IOL_RecLen variable when reading entries.