

QCM50-K1 Process Data Function

January 16th, 2026

This document covers the installation and use of a function for Siemen's TIA Portal software package. This function handles cyclic IO-Link Process Data In from a Banner QCM50-K1 sensor via an IO-Link Master to a Siemens PLC. The function covers parsing and display of the QCM50-K1 sensor Process Data In.

Components

Banner QCM50 Library v16.zal16

There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

1. Open a project.
2. Go to the Open Global Library option in the Libraries tab in TIA Portal v16 or greater.



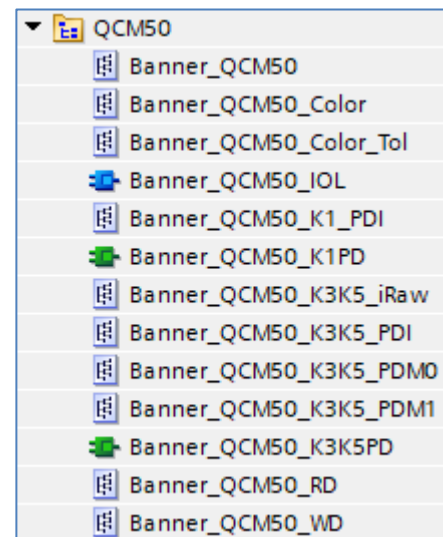
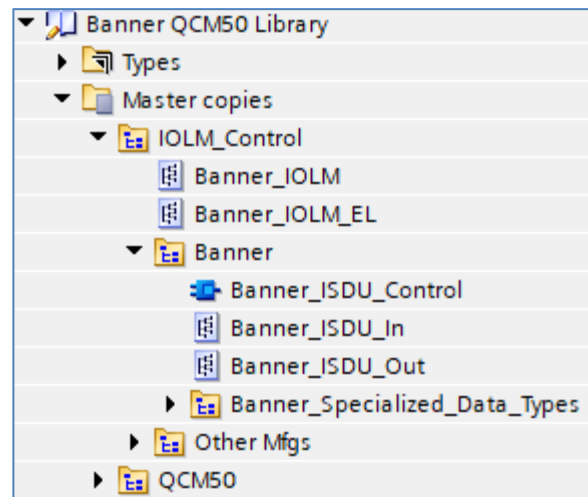
3. Switch the “Files of type” to Compressed libraries. Go to the location of the compressed library.
4. Press the Open button and the library will be uncompressed and opened.
5. The library is now accessible in the Libraries tab in v16 or greater.

Setup of QCM50 K1 with a Banner DXMR

1. Go to Device and Networks to configure the DXMR. Add the DXMR if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR for IO-Link mode.
3. Open the IO-Link Generic Devices and select the proper module. The 2/2 byte option has been selected for port 1. Make note of the I address for the Slot 2 which represents Port 1. Slot 2 starts are 10. The other number needed is I14. The data for the port start at that point (I14). The previous four bytes represents Port Status, Process Data In Size, and Process Data Out Size.

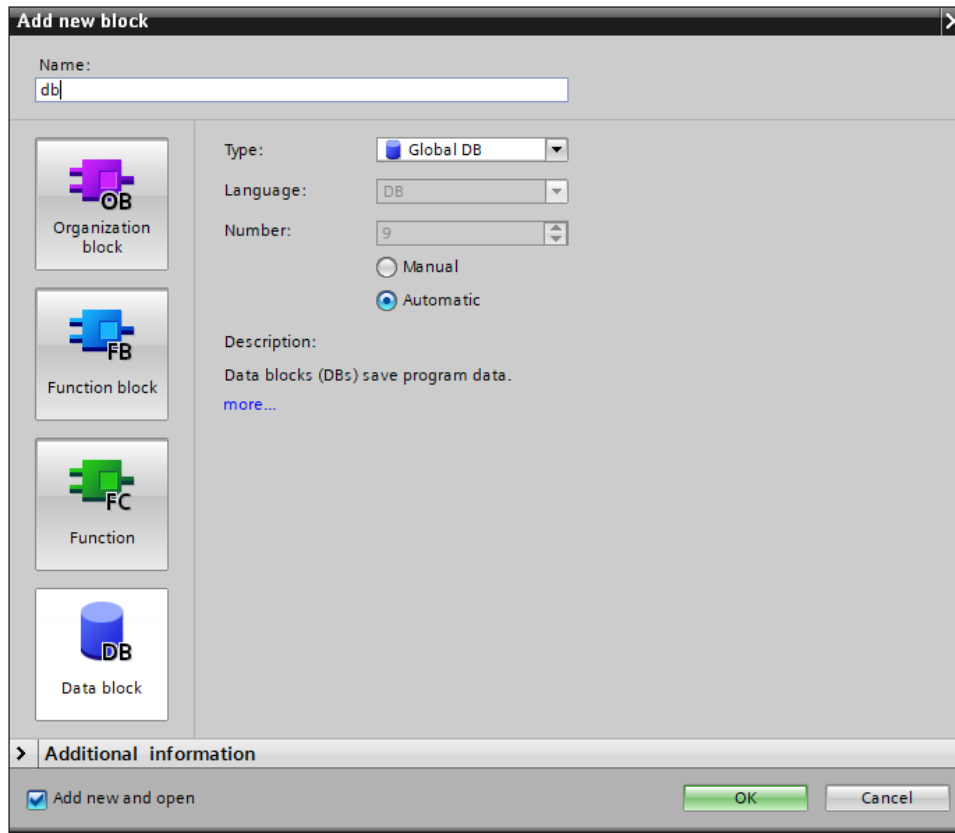
Module	Rack	Slot	I address	Q address	Type
▼ dxm	0	0			1-port Device
▶ Interface	0	0 X1			dxm
Banner IO-Link Master Info_1	0	1	1...9		Banner IO-Link Master Info
IO-Link In/Out 2/ 2 Byte + Status_1	0	2	10...15	1...16	IO-Link In/Out 2/ 2 Byte + Status

4. Drag the Banner_QCM50 K1_PDI to the PLC Data Types area under your PLC. Banner_QCM50 K1_PDI is found in the QCM50 K1 folder in the library. Drag the Banner_QCM50 K1_PD to the Program Blocks area.
5. Drag the necessary tag from IOLM_Control > Banner > Banner_Specialized_Data_Types. The tag used in this example is "Banner_8in". This tag represents the full raw process data along with port status information.
6. Go to PLC Tags. Create two tags. One tag is for the full data structure while the second creates a tag to represent the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, then the tag "QCM50 K1 IOLM1 01 PDI" was created using a Data Type of "Banner_8In". This naming convention calls out the type of sensor in question as well as the specific IO-Link Master and port number where the sensor is connected. A different IO-Link Master might be named IOLM2 or IOLM3, for instance, and other specific sensors may be connected to different port numbers. The "I" address found in step 2 is tied to this new tag. The second is "QCM50 K1 IOLM1 01 inRaw". This is the tag that will be used in the Function block.



Name	Data type	Address
▶ QCM50K1 IOLM1 01 PDI	"Banner_2In"	%I10.0
QCM50K1 IOLM1 01 inRaw	UInt	%IW14

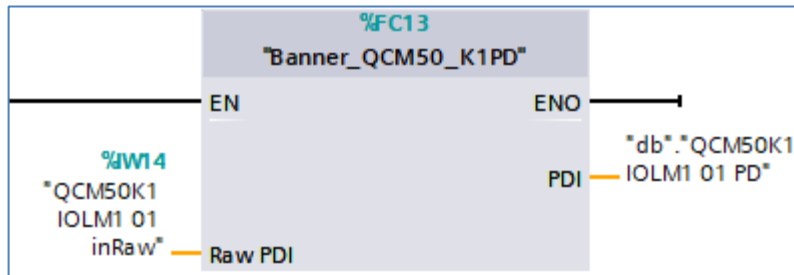
7. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.



8. In the new data block, create a new tag to represent the parsed Process Data In for our QCM50 K1. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_QCM50 K1_PDI” for the new tag.

Name	Data type
▼ Static	
■ ▼ QCM50K1 IOLM1 01 PD	"Banner_QCM50_K1_PDI"
■ Signal Quality	USInt
■ Switching Quality	Bool
■ Q1	Bool

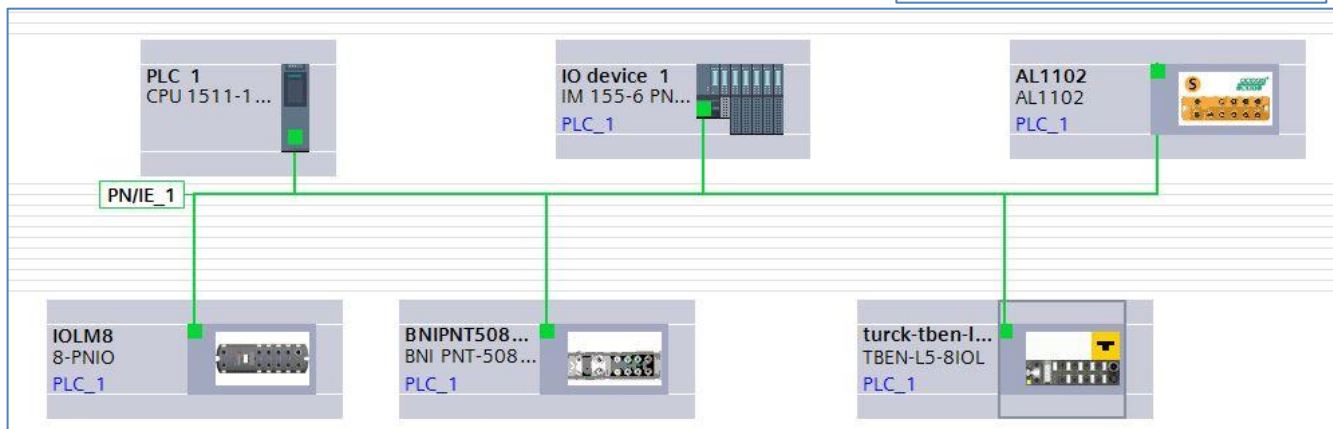
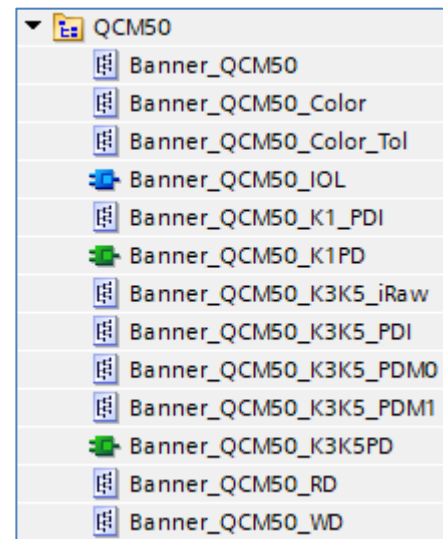
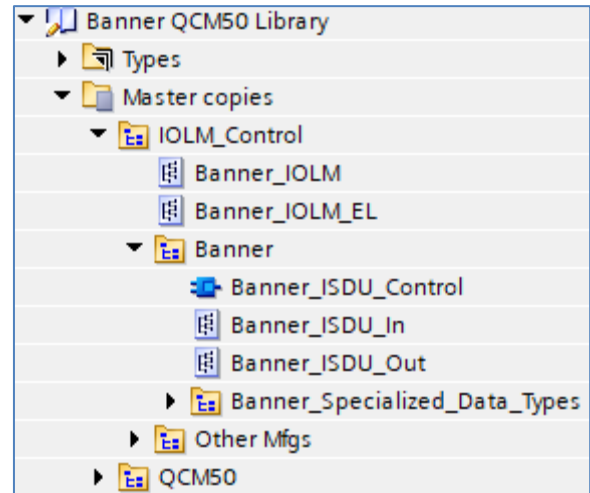
9. Add the “Banner_QCM50 K1_PD” function to an OB ladder. Link the “Process Data Word” to the raw Process Data variable from step 4. Link the “QCM50 K1 Process Data” to the parsed Process Data variable from step 6.



10. Process Data setup is complete.
11. Compile and download the configuration to the PLC, then go online. Open the “db” data block and click Monitor all. You should see parsed QCM50 K1 Process Data In, like that shown below.


Setup of QCM50 K1 with other IO-Link Masters

1. The Banner QCM50 Library will now be in the Global Library List. Expand the Master copies section. The QCM50 folder contains elements for both Process Data (for K1 and K3/5) and Parameter Data (for all) connections to a QCM50 sensor. As Process Data for a QCM50-K1 is the focus of this paper, we will concern ourselves with these two items: Banner_QCM50_K1_PDI and Banner_QCM50_K1PD.
2. Drag Banner_QCM50_K1PD to the Program Blocks area under your PLC.
3. Drag the Banner_QCM50_K1_PDI to the PLC Data Types area under your PLC.
4. Go to Devices and networks to configure the system as necessary. Below is an example of what a configuration might look like. This example shows 5 different IO-Link Masters connected to the same PLC.

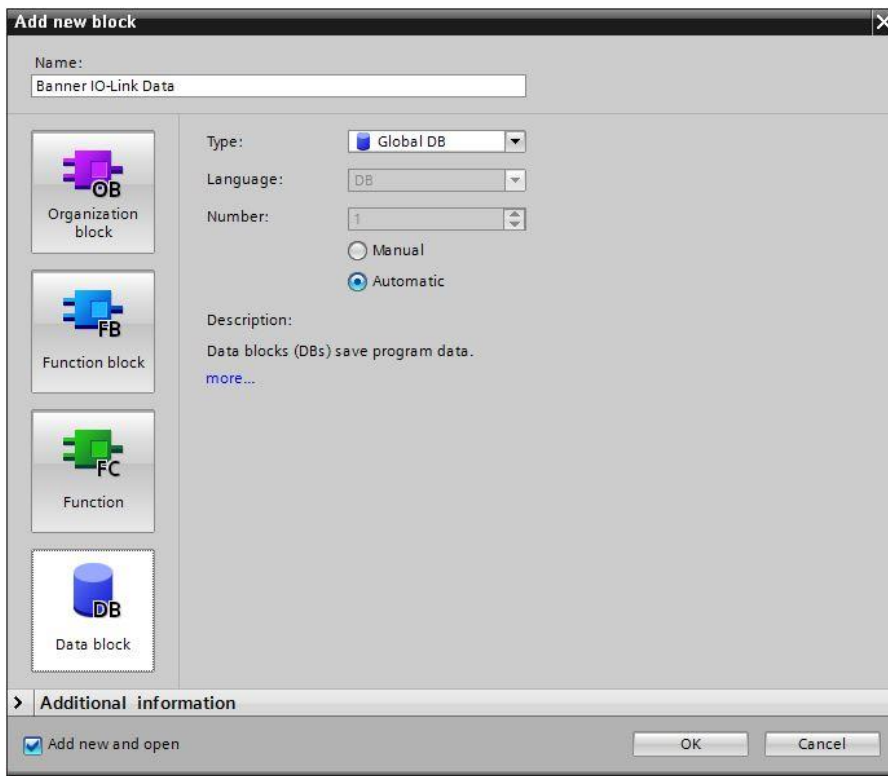


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a QCM50 requires 2 bytes of space for the Process Data.

6. Record the “I” address where this QCM50 Process Data is to be stored, as the address will be required in the next step. In this example, 2 bytes of Process Data In for port 1 on the IO-Link Master will be stored in I68 and I69.
7. Go to PLC Tags. Add a new tag table, then create a new tag to represent the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, then the tag “QCM50-K1 IOLM5 01 PD” was created using a Data Type of “Banner_QCM50_K1_PDI”. This naming convention calls out the type of sensor in question as well as the specific IO-Link Master and port number where the sensor is connected. A different IO-Link Master might be named IOLM1 or IOLM2, for instance, and other specific sensors may be connected to different port numbers. The “I” address found in step 9 is tied to this new tag.

Default tag table			
	Name	Data type	Address
1	 QCM50-K1 IOLM5 01 PD	"Banner_QCM50_K1_PDI"	%I68.0

8. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “Banner IO-Link Data”.



Add new block

Name: Banner IO-Link Data

Type: Global DB

Language: DB

Number: 1

☐ Manual

☒ Automatic

Description: Data blocks (DBs) save program data. [more...](#)

Additional information

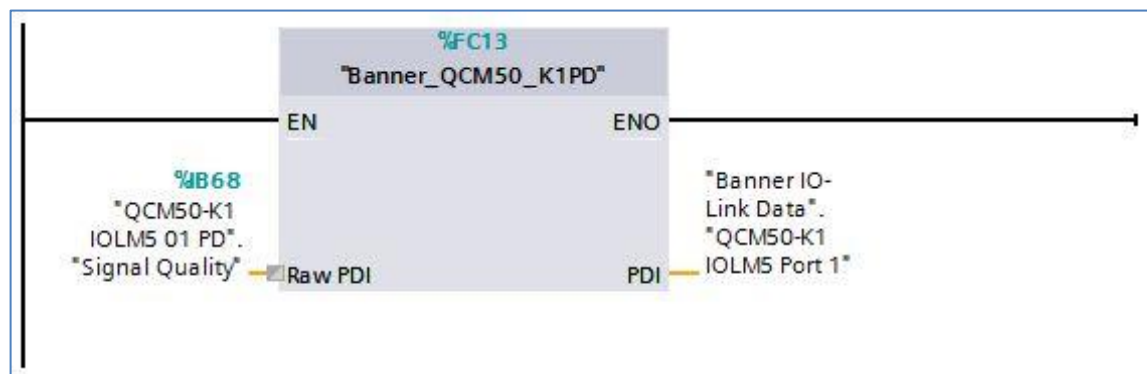
☒ Add new and open

OK Cancel

9. In the new data block, create a new tag to represent the parsed Process Data In for our QCM50. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_QCM50_K1_PDI” for the new tag.

Banner IO-Link Data		
	Name	Data type
1	Static	
2	QCM50-K1 IOLM5 Port 1	*Banner_QCM50_K1_PDI*

10. Add the “Banner_QCM50_K1PD” function to an OB ladder. Link the “Raw PDI” to the raw Process Data variable from step 10. Link the “PDI” to the parsed Process Data variable from step 12.



11. Process Data setup is complete.
12. Compile and download the configuration to the PLC, then go online. Open the “Banner IO-Link Data” data block and click Monitor all. You should see parsed QCM50-K1 Process Data In, like that shown below.

3	QCM50-K1 IOLM5 Port 1	*Banner_QCM50_K1_PDI*		
4	Signal Quality	USInt	0	0
5	Switching Quality	Bool	false	FALSE
6	Q1	Bool	false	TRUE

Appendix A**QCM50-K1 Process Data**

The QCM50-K1 has 2 bytes of Process Data In, as shown below.

ProcessDataIn "Switching channel" id=PDIN_SwitchingOutput									
bit length: 16 data type: 16-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	8	8-bit UInteger	0 = no signal, 1 = overflow, 2..100		ro			Signal quality	
2	1	Boolean			ro			Switching quality	
3	0	Boolean			ro			Q1	
Octet 0									
bit offset	15	14	13	12	11	10	9	8	
subindex									
element bit	7	6	5	4	3	2	1	0	
Octet 1									
bit offset	7	6	5	4	3	2	1	0	
subindex	//////	//////	//////	//////	//////	//////	2	3	

This Process Data is mapped to a specific group of PROFINET addresses.

This function intelligently parses this Process Data into its component pieces.