

TL50 Pro Process Data Function

October 22nd, 2024

This document covers the installation and use of a function for Siemens's TIA Portal software package. This function handles cyclic IO-Link Process Data Out to a Banner TL50 Pro light via an IO-Link Master from a Siemens PLC. The function covers parsing and display of the TL50 Pro sensor Process Data Out.

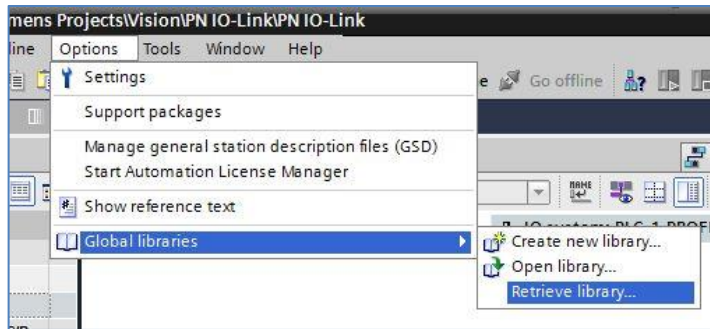
Components

Banner TL50 Library.zal14

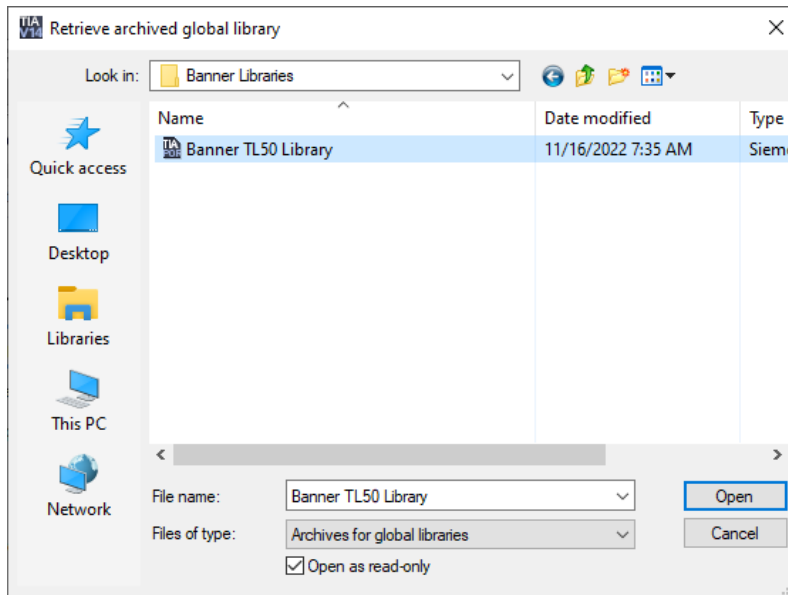
There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

1. Open a project.
2. Go to Options > Global Libraries > Retrieve Library.



3. Select the Banner TL50 Library. Click Open.



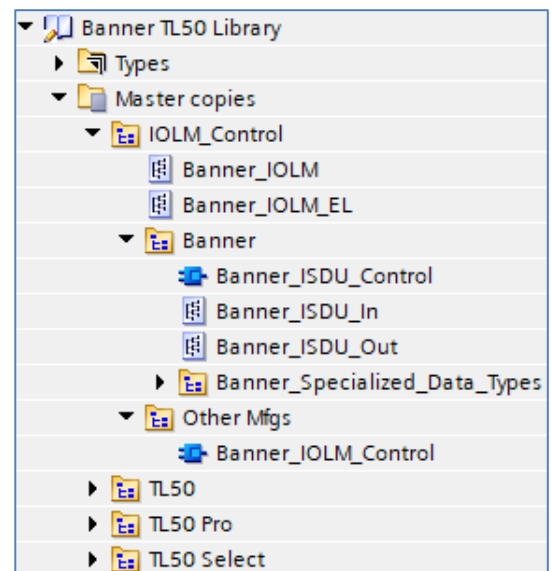
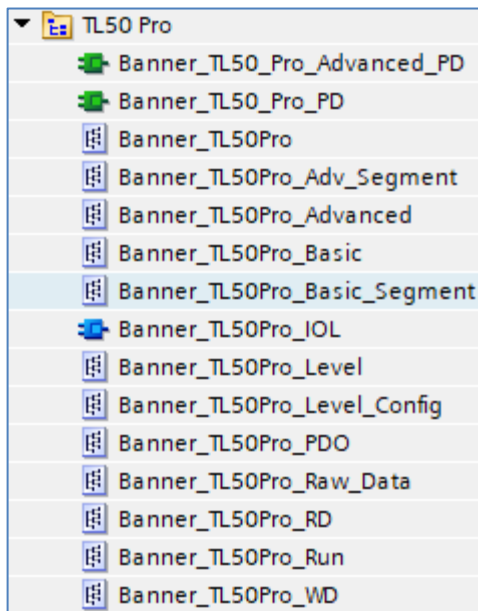
4. The library is now accessible in the Libraries tab.
5. Go to page 3 for Banner IO-Link Masters and to page 7 for all other IO-Link Masters.

Setup of TL50 Pro with a Banner DXMR90-4K

1. Go to Device and Networks to configure the DXMR90-4K. Add the DXMR90-4K if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR90-4K for IO-Link mode.
3. Open the IO-Link Generic Devices and select the proper module. The 32/32 byte is required for TL50 Select. Make note of the Q address for the Slot 2 which represents Port 1. Slot 2 starts are 1 for outputs. The other number needed is Q3. The data for the port start at that point (I3). The previous two bytes Port Control.

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 32/32 Byte + Status_1	0	2	10...45	1...46	IO-Link In/Out 32/32 Byte + Status

4. Drag the necessary tag from IOLM_Control > Banner > Banner_Specialized_Data_Types. The tag used in this example is “Banner_32out”. This tag represents the full raw process data along with port status information.
5. Drag the necessary files from the TL50 Pro Folder.
 - a. Move Banner_TL50Pro_Adv_Segment, Banner_TL50Pro_Advanced, Banner_TL50Pro_Basic, Banner_TL50Pro_Level, Banner_TL50Pro_PDO, Banner_TL50Pro_Raw_Data, and Banner_TL50Pro_Run to the PLC Data Types area.

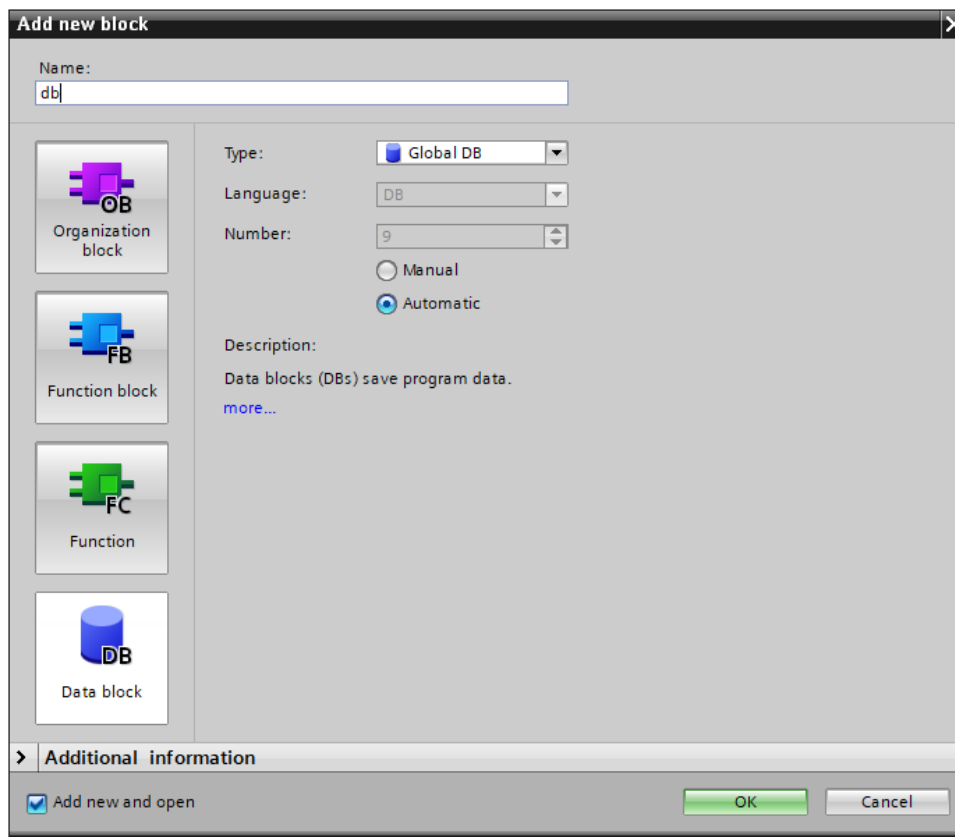


- b. Move Banner_TL50Pro_PD and Banner_TL50Pro_Advanced_PD to the Program Blocks area.

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 “TL510Pro IOLM1 02 PDO” was created using a Data Type of “Banner_32out”. This naming convention calls out the type of device 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 (%I68) is tied to this new tag. The second is “TL50Pro IOLM1 02 outRaw” and uses the “I” address found in step 2 (%I72). This is the tag that will be used in the Function block.

Name	Data type	Address
▶ TL50Pro IOLM1 01 PDO	"Banner_32Out"	%Q1.0
▶ TL50Pro IOLM1 01 outRaw	"Banner_TL50Pro_Raw_Data"	%Q3.0

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 TL50 Pro. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_TL50Pro_PDO” for the new tag.

Name	Data type
▼ Static	
■ ▼ TL50Pro IOLM1 01 PD	"Banner_TL50Pro_PDO"
■ ▶ Advanced Mode	"Banner_TL50Pro_Advanced"
■ ▶ Basic Mode	"Banner_TL50Pro_Basic"
■ ▶ Level Mode	"Banner_TL50Pro_Level"
■ ▶ Run Mode	"Banner_TL50Pro_Run"
■ Index	Int

9. Add the “Banner_TL50_Pro_PD” function to an OB ladder. Link the “Process Data Out” to the raw Process Data variable from step 5. Link “TL50 Pro PDO” to the parsed Process Data variable from step 7.

The last two variables, “Operational Mode” and “Segments”, allow the function to correctly interpret the Process Data Out. In the case of the TL50 Pro, there are four user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL50 Pro for this Operational Mode variable.

There are two ways to achieve this goal. We can simply type in the correct number for Operational Mode (see Fig. 1), or we can link this TL50 Pro Process Data Function to the TL50 Pro Parameter Data Function Block (see Fig. 2). See Appendix A for more information about TL50 Pro Process Data Out.

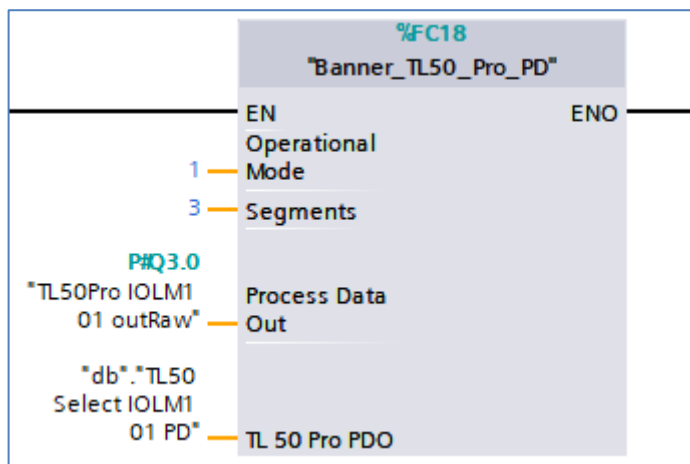


Figure 1: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the tower light’s current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

Operational Mode: the options here are “0” (Basic Mode; on/off/flash/animation state for up to 10 segments plus audible), “1” (Advanced Mode; full RGB control off all 10 segments), “2” (Run Mode; a situation where the entire tower light acts as one device), and “3” (Level Mode; where the entire tower light behaves as a level indicator). The default is “1”.

Segments: this is the number of segments in the tower light (excluding audible segment, if present).

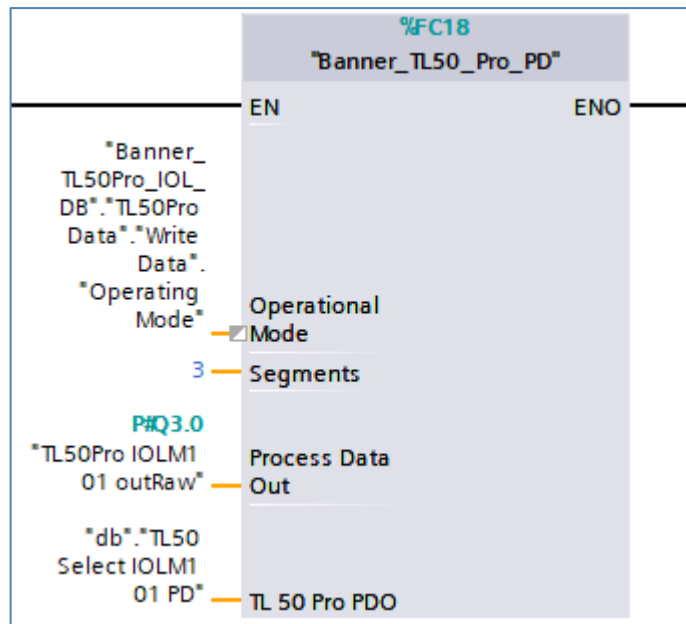
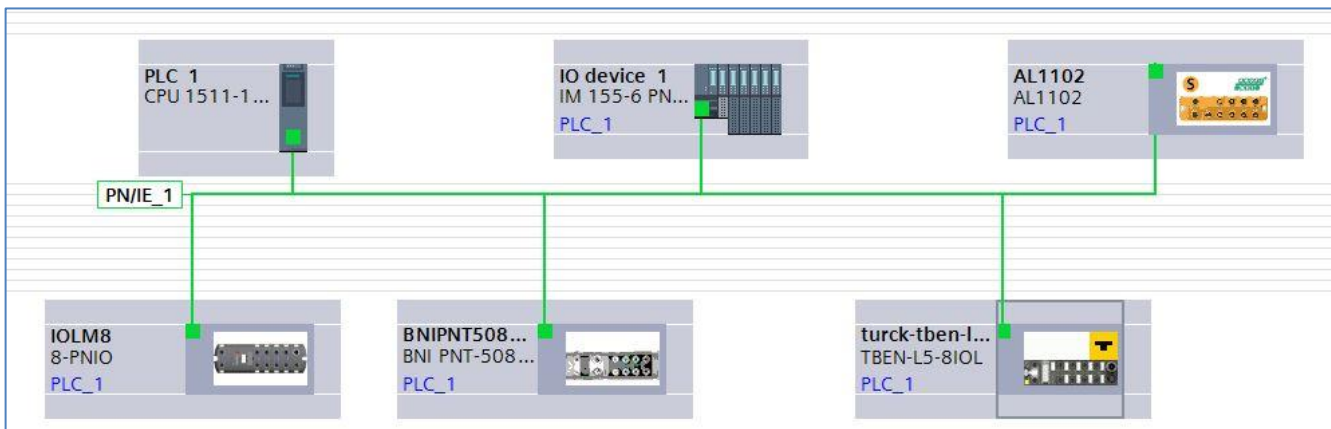
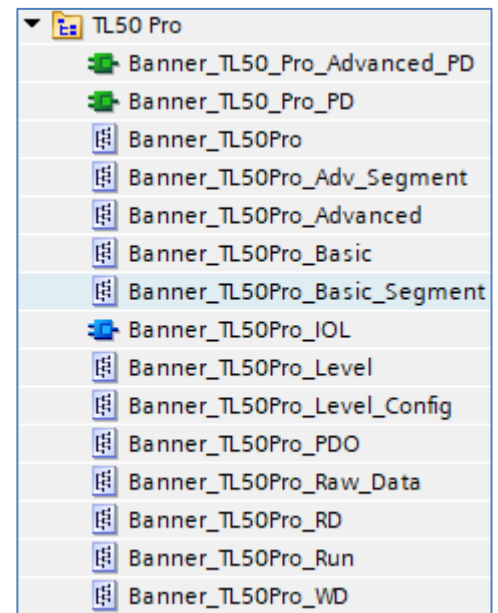
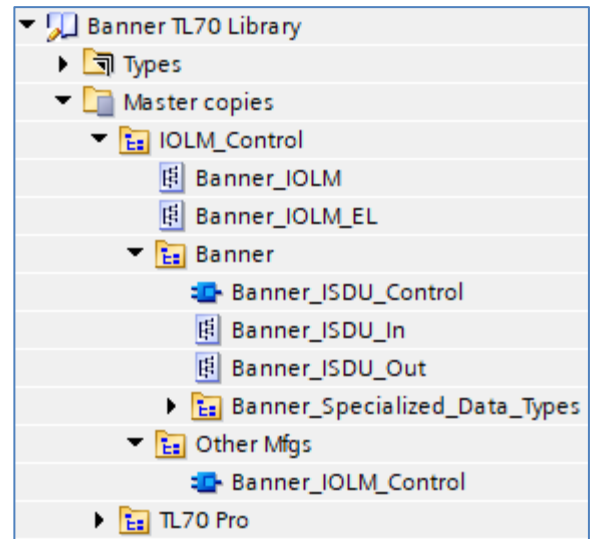


Figure 2: Linking Operational Mode variable to TL50 Pro Parameter Data Function Block

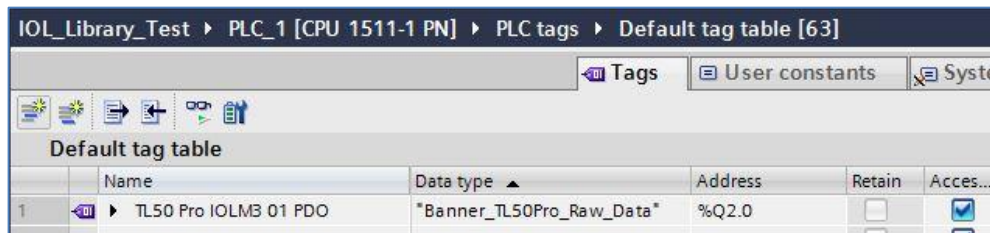
10. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 “TL50Pro IOLM1 01 PDO”.
11. Process Data setup is complete.
12. Compile and download the configuration to the PLC, then go online. Open the “db” data block and click Monitor all. You can now control the TL50 Pro via the data located here.

Setup of TL50 Pro with other IO-Link Masters

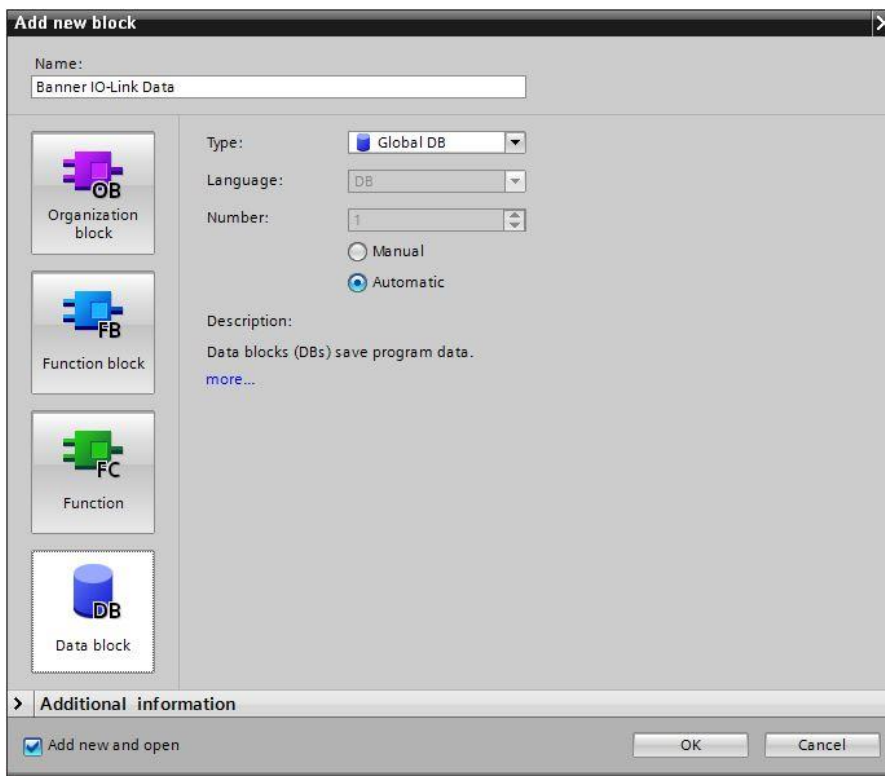
1. The Banner TL50 Library will now be in the Global Library List. Expand the Master copies section. The TL50 Pro folder contains elements for both Process Data and Parameter Data connections to a TL50 Pro device. As Process Data is the focus of this paper, we will concern ourselves with these items:
Banner_TL50_Pro_Advanced_PD,
Banner_TL50_Pro_PD,
Banner_TL50Pro_Adv_Segment,
Banner_TL50Pro_Advanced,
Banner_TL50Pro_Basic, Banner_TL50Pro_Level,
Banner_TL50Pro_PDO,
Banner_TL50Pro_Raw_Data, and
Banner_TL50Pro_Run.
2. Drag Banner_TL50_Pro_Advanced_PD and Banner_TL50_Pro_PD to the Program Blocks area under your PLC.
3. Drag the other seven items listed above 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.



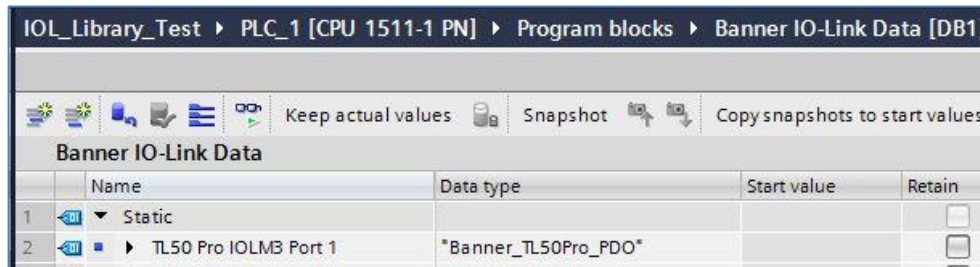
5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a TL50 Pro requires 32 bytes of space for the Process Data.
6. Record the “Q” address where this TL50 Pro Process Data is to be stored, as the address will be required in the next step. In this example, 32 bytes of Process Data Out for port 1 on the IO-Link Master will be stored in Q2 through Q33.
7. Go to PLC Tags. Add a new tag table, then create a new tag to represent the raw Process Data Out to be sent to the IO-Link Master. In this example, Tag table_1 was created, then the tag “TL50 Pro IOLM3 01 PDO” was created using a Data Type of “Banner_TL50Pro_Raw_Data”. 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 “Q” address found in step 9 is tied to this new tag.



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”.



9. In the new data block, create a new tag to represent the parsed Process Data In for our TL50 Pro. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_TL50Pro_PDO” for the new tag.



Add the “Banner_TL50_Pro_PD” function to an OB ladder. Link the “Process Data Out” to the raw Process Data variable from step 10. Link “TL50 Pro PDO” to the parsed Process Data variable from step 12.

The last two variables, “Operational Mode” and “Segments”, allow the function to correctly interpret the Process Data Out. In the case of the TL50 Pro, there are four user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL50 Pro for this Operational Mode variable.

There are two ways to achieve this goal. We can simply type in the correct number for Operational Mode (see Fig. 1), or we can link this TL50 Pro Process Data Function to the TL50 Pro Parameter Data Function Block (see Fig. 2). See Appendix A for more information about TL50 Pro Process Data Out.

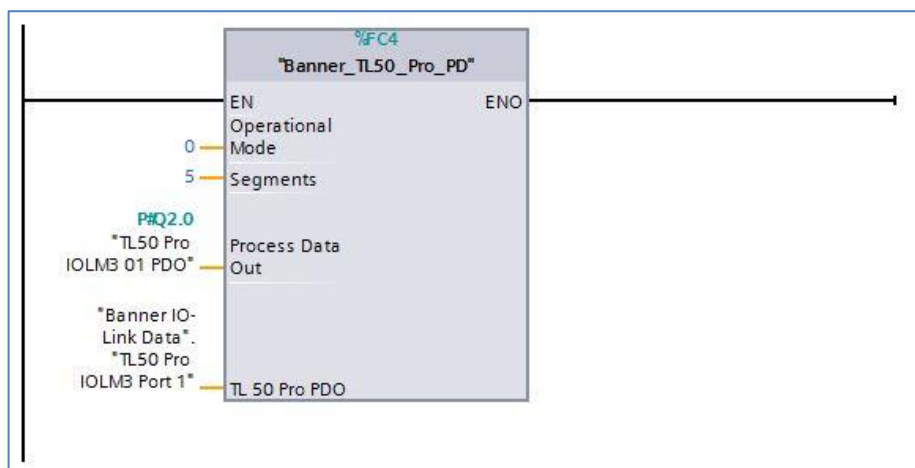


Figure 3: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the tower light’s current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

Operational Mode: the options here are “0” (Basic Mode; on/off/flash/animation state for up to 10 segments plus audible), “1” (Advanced Mode; full RGB control off all 10 segments), “2” (Run Mode; a situation where the entire tower light acts as one device), and “3” (Level Mode; where the entire tower light behaves as a level indicator). The default is “1”.

Segments: this is the number of segments in the tower light (excluding audible segment, if present).

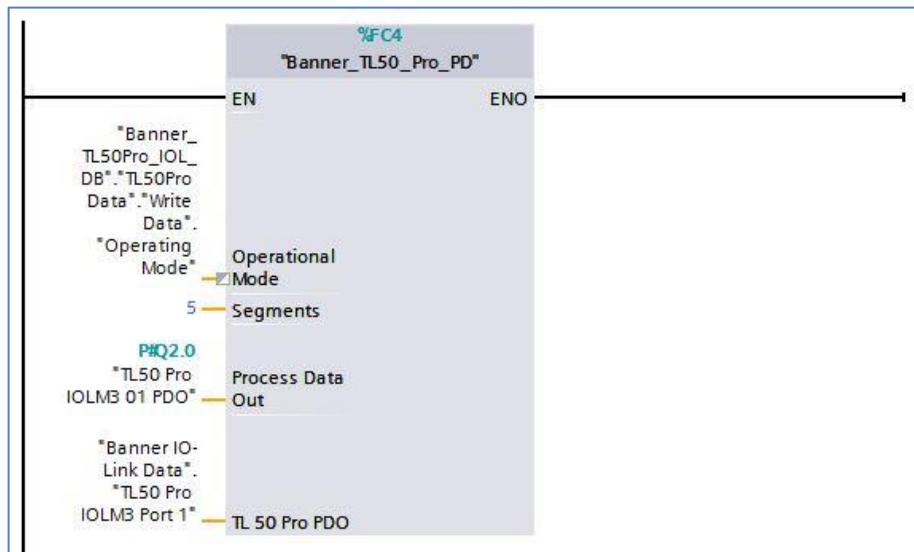


Figure 4: Linking Operational Mode variable to TL50 Pro Parameter Data Function Block

10. Process Data setup is complete.

11. 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 TL50 Pro Process Data Out, like that shown below.

Banner IO-Link Data				
	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ TL50 Pro IOLM3 Port 1	"Banner_TL50Pro_PDO"		
3	▶ Advanced Mode	"Banner_TL50Pro_Advanced"		
4	▼ Basic Mode	"Banner_TL50Pro_Basic"		
5	■ Audible State	USInt	0	0
6	■ Segment 1	USInt	0	1
7	■ Segment 2	USInt	0	2
8	■ Segment 3	USInt	0	0
9	■ Segment 4	USInt	0	0
10	■ Segment 5	USInt	0	0
11	■ Segment 6	USInt	0	0
12	■ Segment 7	USInt	0	0
13	■ Segment 8	USInt	0	0
14	■ Segment 9	USInt	0	0
15	■ Segment 10	USInt	0	0
16	▶ Level Mode	"Banner_TL50Pro_Level"		
17	▶ Run Mode	"Banner_TL50Pro_Run"		

Figure 5: Basic Mode Process Data Out (Operational Mode = 0)

Banner IO-Link Data				
	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ TL50 Pro IOLM3 Port 1	"Banner_TL50Pro_PDO"		
3	▼ Advanced Mode	"Banner_TL50Pro_Advanced"		
4	▼ Segment	Array[1..10] of "Banner_TL50Pro_...		
5	▼ Segment[1]	"Banner_TL50Pro_Adv_Segment"		
6	■ Color 1	USInt	0	0
7	■ Color 1 Intensity	USInt	0	0
8	■ Animation Type	USInt	0	0
9	■ Speed	USInt	0	0
10	■ Pulse Pattern	USInt	0	0
11	■ Color 2	USInt	0	0
12	■ Color 2 Intensity	USInt	0	0
13	■ Rotational Directi...	USInt	0	0
14	▶ Segment[2]	"Banner_TL50Pro_Adv_Segment"		
15	▶ Segment[3]	"Banner_TL50Pro_Adv_Segment"		
16	▶ Segment[4]	"Banner_TL50Pro_Adv_Segment"		
17	▶ Segment[5]	"Banner_TL50Pro_Adv_Segment"		
18	▶ Segment[6]	"Banner_TL50Pro_Adv_Segment"		
19	▶ Segment[7]	"Banner_TL50Pro_Adv_Segment"		
20	▶ Segment[8]	"Banner_TL50Pro_Adv_Segment"		
21	▶ Segment[9]	"Banner_TL50Pro_Adv_Segment"		
22	▶ Segment[10]	"Banner_TL50Pro_Adv_Segment"		
23	■ Audible State	USInt	0	0

Figure 6: Advanced Mode Process Data Out (Operational Mode = 1)

Banner IO-Link Data				
	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ TL50 Pro IOLM3 Port 1	"Banner_TL50Pro_PDO"		
3	▶ Advanced Mode	"Banner_TL50Pro_Advanced"		
4	▶ Basic Mode	"Banner_TL50Pro_Basic"		
5	▶ Level Mode	"Banner_TL50Pro_Level"		
6	▼ Run Mode	"Banner_TL50Pro_Run"		
7	■ Animation	USInt	0	0
8	■ Color 1	USInt	0	0
9	■ Color 1 Intensity	USInt	0	0
10	■ Speed	USInt	0	0
11	■ Pulse Pattern	USInt	0	0
12	■ Color 2	USInt	0	0
13	■ Color 2 Intensity	USInt	0	0
14	■ Segment Shift	USInt	0	0
15	■ Rotational Direction	USInt	0	0
16	■ Audible State	USInt	0	0

Figure 7: Run Mode Process Data Out (Operational Mode = 2)

Banner IO-Link Data				
	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ TL50 Pro IOLM3 Port 1	"Banner_TL50Pro_PDO"		
3	▶ Advanced Mode	"Banner_TL50Pro_Advanced"		
4	▶ Basic Mode	"Banner_TL50Pro_Basic"		
5	▼ Level Mode	"Banner_TL50Pro_Level"		
6	■ Audible State	Byte	16#0	16#00
7	■ Level Mode Value	UInt	0	0

Figure 8: Level Mode Process Data Out (Operational Mode = 3)

Appendix A**TL50 Pro Process Data Out**

The TL50 Pro has 30 bytes of Process Data Out, mapped into 4 different modes, as shown below.

This Process Data is mapped to a specific group of PROFINET addresses. The 248-bits of Process Data actually encode many separate pieces of information, as shown below.

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

First is the Basic mode (mode 0). This controls the basic on/off/flash/animation state of each segment and the off/on/pulse/sos state of the audible (if present).

ProcessDataOut "Process Data Out Basic" id=V_Pd_OutBasic									
bit length: 248 data type: 248-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	200	2-bit UInteger	0 = Off, 1 = On, 2 = Pulsed, 3 = SOS Pulse					Audible State	The state of the audible segment
2	216	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 1	The state of the segment. Related parameters defined in Basic Segment Parameter Data
3	218	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 2	The state of the segment. Related parameters defined in Basic Segment Parameter Data
4	220	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 3	The state of the segment. Related parameters defined in Basic Segment Parameter Data
5	222	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 4	The state of the segment. Related parameters defined in Basic Segment Parameter Data
6	224	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 5	The state of the segment. Related parameters defined in Basic Segment Parameter Data
7	226	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 6	The state of the segment. Related parameters defined in Basic Segment Parameter Data
8	228	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 7	The state of the segment. Related parameters defined in Basic Segment Parameter Data
9	230	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 8	The state of the segment. Related parameters defined in Basic Segment Parameter Data
10	232	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 9	The state of the segment. Related parameters defined in Basic Segment Parameter Data
11	234	2-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 10	The state of the segment. Related parameters defined in Basic Segment Parameter Data

Here is the information for 1 segment (out of 10) for Advanced mode (mode 1). This mode grants the user full control over every segment of the TL50 Pro.

ProcessDataOut "Process Data Out Advanced" id=V_Pd_OutAdvanced									
bit length: 248 data type: 248-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Segment 1 Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
2	4	3-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Segment 1 Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
3	8	3-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = 50/50, 5 = 50/50 Rotate, 6 = Chase, 7 = Intensity Sweep					Segment 1 Animation Type	The Animation type
4	11	2-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom					Segment 1 Speed	The speed of the Animation
5	13	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Segment 1 Pulse Pattern	The pattern of Animation
6	16	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Segment 1 Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
7	20	3-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Segment 1 Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	23	Boolean	false = Counter Clockwise, true = Clockwise					Segment 1 Rotational Direction	The Direction of Animation rotation

Here is Run mode (mode 2).

ProcessDataOut "Process Data Out Run Mode" id=V_Pd_OutRunMode									
bit length: 248 data type: 248-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	168	4-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = 50/50, 5 = 50/50 Rotate, 6 = Chase, 7 = Intensity Sweep, 8 = Scroll, 9 = Bounce, 10 = Color Spectrum, 11 = Demo					Animation	The Animation type
2	176	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	184	3-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	192	2-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow					Speed	The speed of the Animation
5	200	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	208	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
7	216	3-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	224	Boolean	false = No Shift, true = Shift Enabled					Segment Shift	When enabled, rotational animations will not line up, creating a different visual effect
9	232	Boolean	false = Counter Clockwise, true = Clockwise					Rotational Direction	The Direction of Animation rotation
10	240	2-bit UInteger	0 = Off, 1 = On, 2 = Pulsed, 3 = SOS Pulse					Audible State	The state of the audible segment

Here is Level mode (mode 3).

ProcessDataOut "Process Data Out Level Mode" id=V_Pd_OutLevelMode									
bit length: 248 data type: 248-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	216	2-bit UInteger	0 = Off, 1 = On, 2 = Pulsed, 3 = SOS Pulse					Audible State	The state of the audible segment
2	232	16-bit UInteger						Level Mode Value	Value describing the level of the device, range determined in Level Mode Parameter Data