

TL70 Pro Process Data Function

11/22/2022

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 TL70 Pro light via an IO-Link Master from a Siemens PLC. The function covers parsing and display of the TL70 Pro sensor Process Data Out.

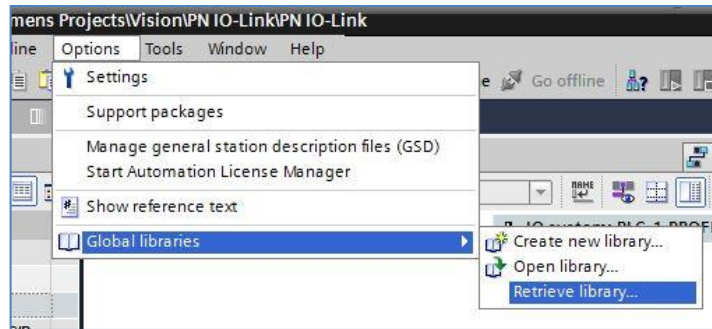
Components

Banner TL70 Library.zal14

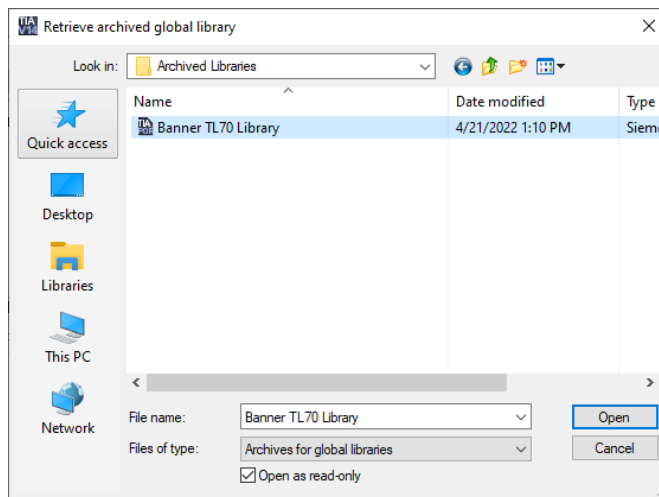
There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

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



3. Select the Banner TL70 Library. Click Open.



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

Setup of TL70 Pro with a Banner DXMR90-4K

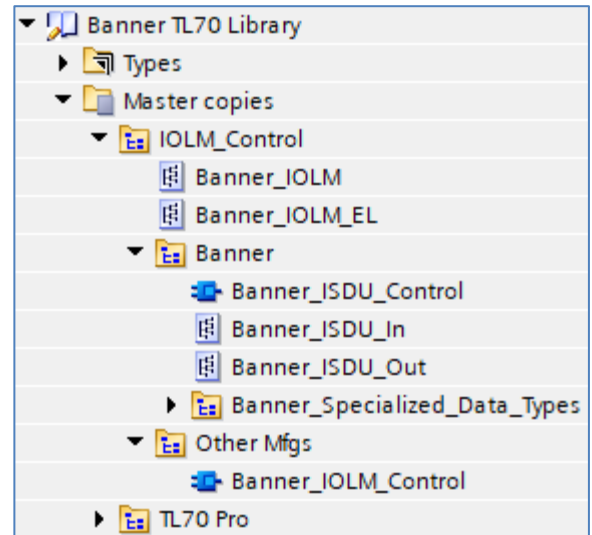
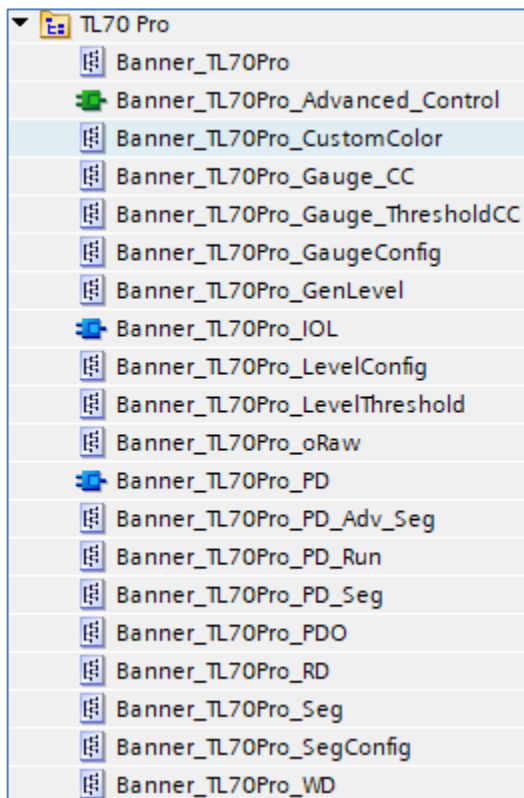
1. Go to Device and Networks to configure the DXMR90-4K. Add the DXMR90-4K if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR90-4K for IO-Link mode.

Banner IO-Link Master Info_1	0	1	1...9	Banner IO-Link Master Info
------------------------------	---	---	-------	----------------------------

3. Open the IO-Link Generic Devices and select the proper module. The 32/32 byte is required for Pro. 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.

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 TL70 Pro Folder.
 - a. Move Banner_TL70Pro_PD_Adv_Seg, Banner_TL70Pro_PD_Run,



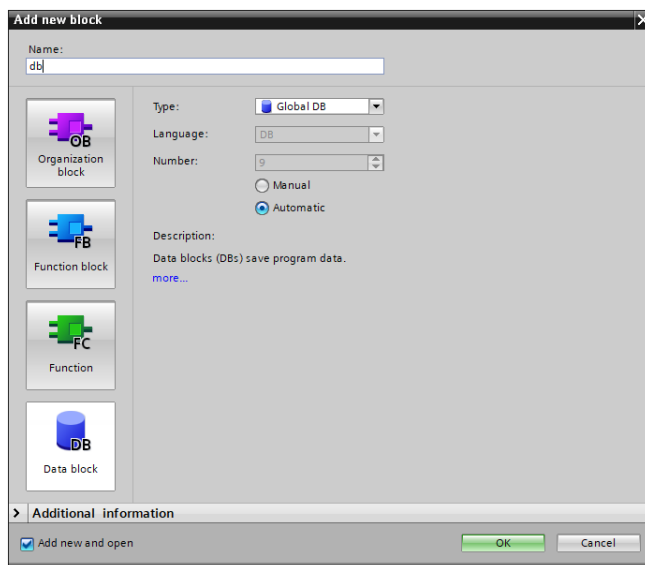
Banner_TL70Pro_PDO, and Banner_TL70Pro_oRaw to the PLC Data Types area.

- b. Move Banner_TL70Pro_PD and Banner_TL70Pro_Advanced_Control 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 “TL70 Pro IOLM1 01 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 “Q” address found in step 2 (%Q1) is tied to this new tag. The second is “TL70 Pro IOLM1 01 outRaw” and uses the “Q” address found in step 2 (%Q3). This is the tag that will be used in the Function block.

Name	Data type	Address
▶ TL70 Pro IOLM1 01 PDO	*Banner_32Out*	%Q1.0
▶ TL70 Pro IOLM1 01 outRaw	*Banner_TL70Pro_oRaw*	%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 Output for our Pro. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_TL70_Pro_PDO” for the new tag.

Name	Data type
▼ Static	
■ ▼ TL70Pro IOLM1 01 PD	*Banner_TL70Pro_PDO*
■ ▶ 0-SegMode	*Banner_TL70Pro_PD_Seg*
■ ▶ 1-RunMode	*Banner_TL70Pro_PD_Run*
■ 2-LevelMode	UInt
■ 3-GaugeMode	UInt
■ ▶ 4-Advanced	Array[1..6] of *Banner_TL70Pro_PD_Adv_Seg*
■ Audible	USInt

8. Add the “Banner_TL70Pro_PD” function to an OB ladder. Link the “PDO” to the raw process data variable from step 5. The tag name again calls out the type of device, IO-Link Master, and the port number. Use the variable was called “TL70 Pro IOLM1 01 outRaw” in this example. The “TL70PDO” needs to be linked to the variable created in step 7. It was called “TL70 Pro IOLM1 01 PD” for this example.

The last variable, “Operational Mode”, allows the function to correctly interpret the Process Data Out. In the case of the TL70 Pro , there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL70 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 TL70 Pro Process Data Function to the TL70 Pro Parameter Data Function Block (see Fig. 2). See Appendix A for more information about TL70 Pro Select Process Data Out.

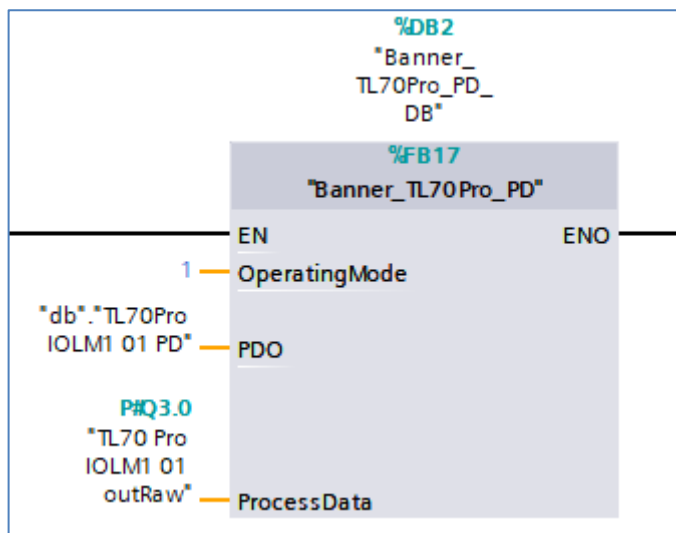


Figure 1: Hand type correct number for Operational Mode

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" (Segment Mode; on/off/flash/animation state for up to 10 segments plus audible), "1" (Run Mode; a situation where the entire tower light acts as one device), "2" (Level Mode); where the entire tower light behaves as a level indicator), "3" (Gauge Mode); where the entire tower light changes states based on the Gauge settings, and "4" (Advanced Mode; full RGB control off all 6 segments), The default is "1".

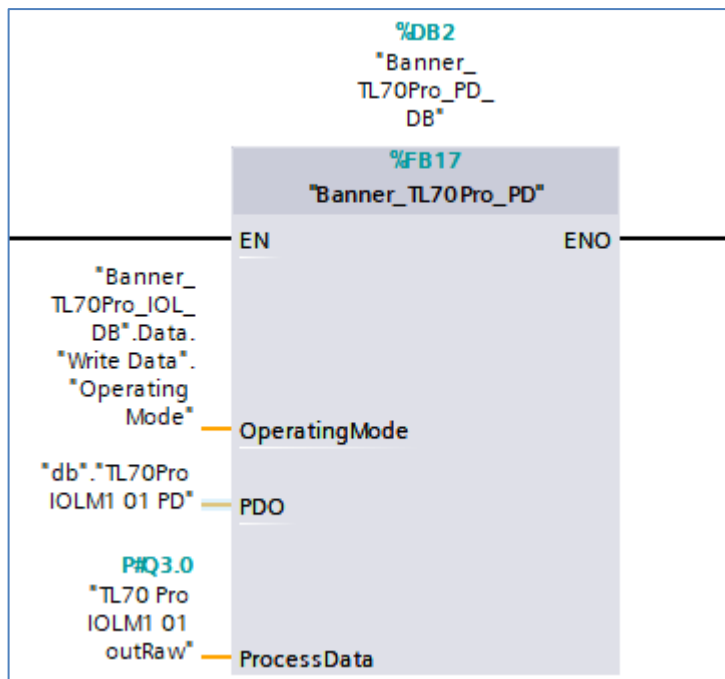
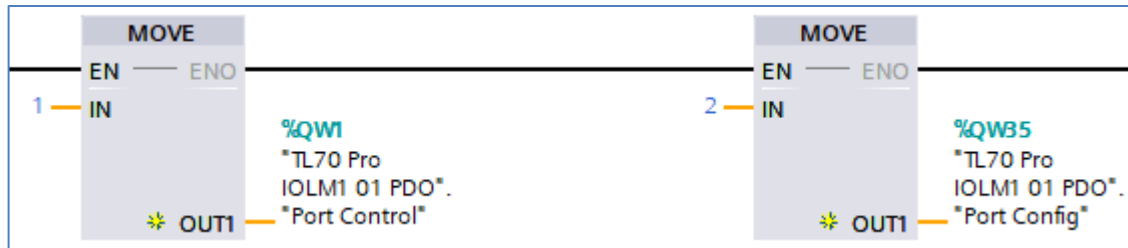


Figure 2: Linking Operational Mode variable to TL70 Pro Parameter Data Function Block

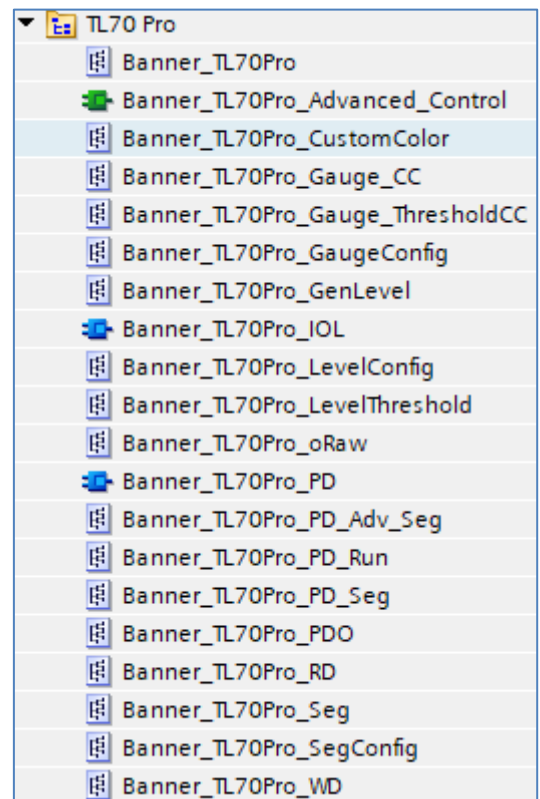
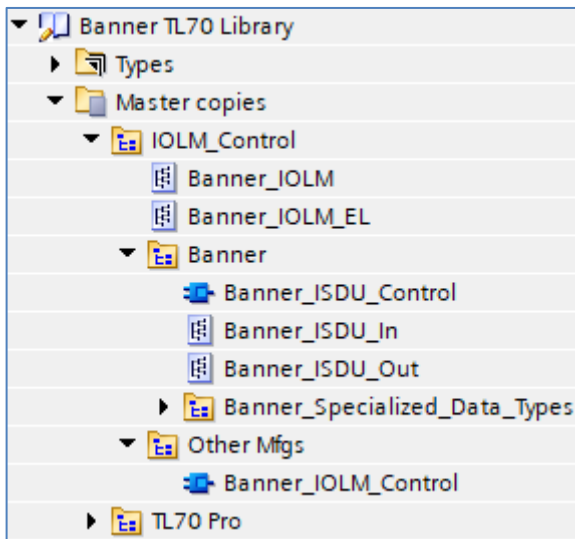
9. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 "TL70 IOLM1 01 PDO".

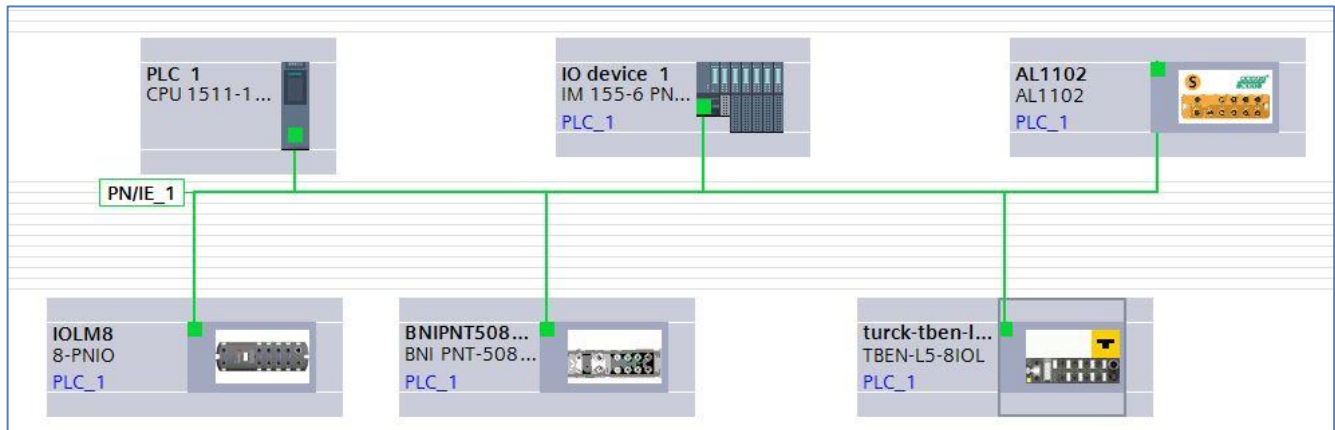


10. Process Data Setup is complete.
11. Compile and download the configuration to the PLC, then go online. Open the "db" data block and click Monitor all. The TL70 Pro can be controlled now.

Setup of TL70 Pro with other IO-Link Masters

1. The Banner TL70 Pro library will now be in the Global Library List. Expand the Master copies section. The TL70 Pro folder contains elements for both Process Data and Parameter Data connections to a TL70 Pro device. As Process Data is the focus of this paper, we will concern ourselves with these seven items: Banner_TL70_Pro_Advanced_Control, Banner_TL70_Pro_oRaw, Banner_TL70Pro_PD, Banner_TL70Pro_PD_Adv_Seg, Banner_TL70_Pro_Run, Banner_TL70Pro_PD_Seg, and Banner_TL70Pro_PDO.
2. Drag Banner_TL70_Pro_Advanced_Control and Banner_TL70_Pro_PD to the Program Blocks area under your PLC.
3. Drag Banner_TL70_Pro_oRaw, Banner_TL70_Pro_PD_Adv_Seg, Banner_TL70_Pro_PD_Run, Banner_TL70_Pro_PD_Seg, and Banner_TL70_Pro_PDO 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.

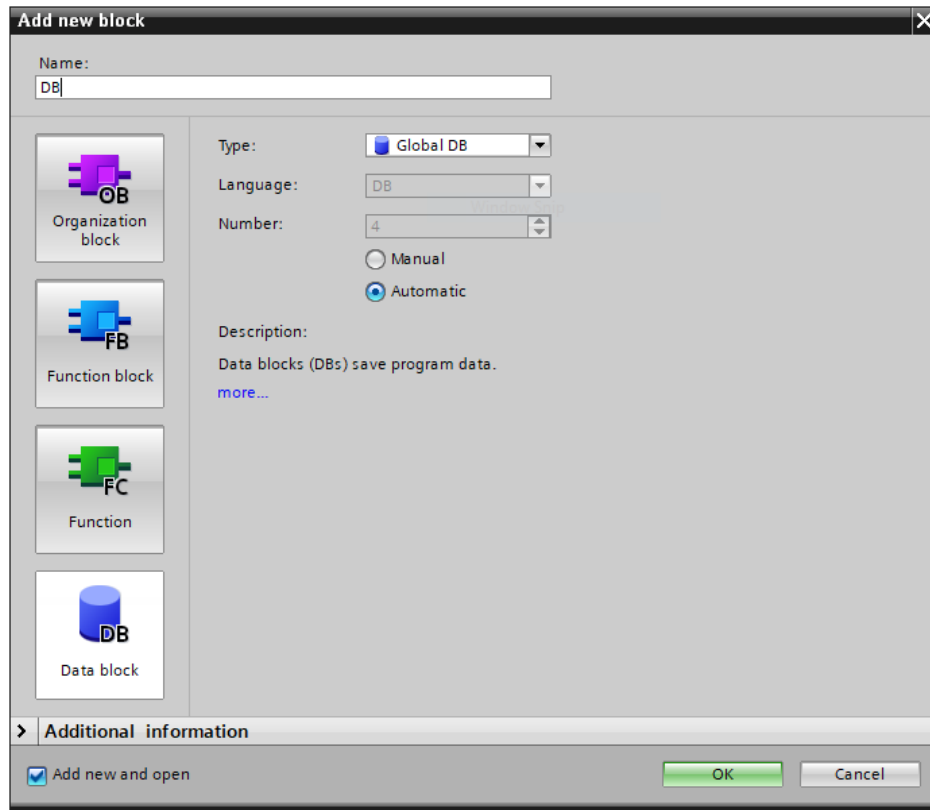




5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a TL70 Pro requires 20 bytes of space for the Process Data Out. This will likely require a 32 byte OUT type.
6. Record the “Q” addresses where this TL70 Pro Process Data is to be stored, as these addresses will be required in the next step. In this example, 20 bytes of Process Data Out for port 1 on the IO-Link Master will be stored in Q1 through Q20.
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 from the IO-Link Master. In this example, Tag table_1 was created, then the tag “TL70 IOLM1 01 PDO” was created using a Data Type of “Banner_TL70_Pro_PDORaw”. 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 “Q” address found in step 9 is tied to this new tag.

Name	Data type	Address
▶ TL70 IOLM1 01 PDO	"Banner_TL70Pro_oRaw"	%Q1.0

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



9. In the new data block, create a new tag to represent the parsed Process Data Out for our TL70 Pro. The tag name again calls out the type of light, the IO-Link Master, and the port number. Use the data type “Banner_TL70_Pro_PDO” for the new tag.

Name	Data type
▼ Static	
▶ TL70 IOLM1 01 PD	"Banner_TL70Pro_PDO"

10. Add the “Banner_TL70_Pro_Pro_PD” function to an OB ladder. Link the “ProcessData” to the raw Process Data Out variable from step 10. Link “PDO” to the parsed Process Data variable from step 12.

The last variable, “Operational Mode”, allows the function to correctly interpret the Process Data Out. In the case of the , there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL70 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 TL70 Pro Process Data Function to the TL70 Pro Audible Parameter Data Function Block (see Fig. 2). See Appendix A for more information about TL70 Pro Process Data Out.

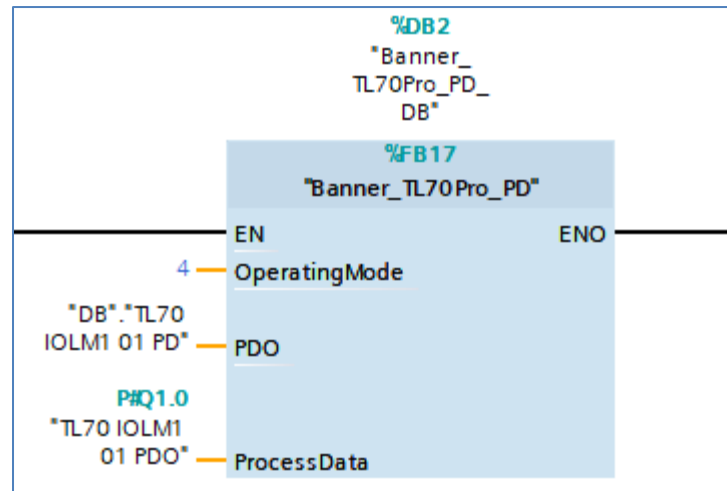


Figure 3: Hand type correct number for Operational Mode

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

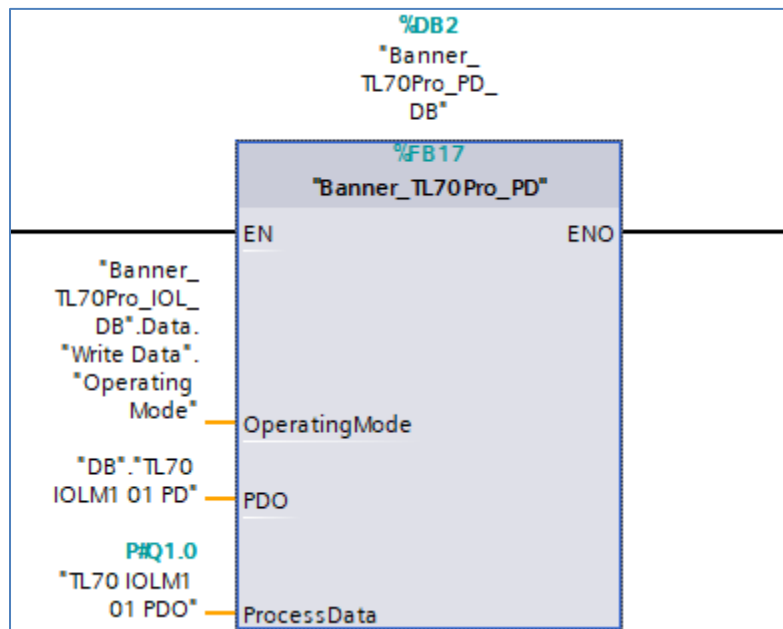


Figure 4: Linking Operational Mode variable to Pro Parameter Data Function Block

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 TL70 Pro Process Data.
13. Process Data is broken up into four different types. When Operational Modes controls which of five types is used to parse the raw byte data. All modes will use the Audible parameter if the tower light has an audible unit.

▼ TL70 IOLM1 01 PD
■ ▶ 0-SegMode
■ ▶ 1-RunMode
■ 2-LevelMode
■ 3-GaugeMode
■ ▶ 4-Advanced
■ Audible

- a. Segment has five pieces of data. Each Segment allows a value of 0 for OFF, 1 for ON, 2 for Flash, and 3 for Animation. Audible allows a value of 0 for OFF and 1 for ON.

▼ TL70 IOLM1 01 PD	*Banner_TL70Pro_P...			
■ ▼ 0-SegMode	*Banner_TL70Pro_P...			Process Data Segment Mode
■ Segment1	USInt	0	1	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ Segment2	USInt	0	2	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ Segment3	USInt	0	0	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ Segment4	USInt	0	0	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ Segment5	USInt	0	0	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ Segment6	USInt	0	0	0 = Off, 1 = On, 2 = Flash, 3 = Animation
■ ▶ 1-RunMode	*Banner_TL70Pro_P...			Process Data Run Mode
■ 2-LevelMode	UInt	0	0	Value describing the level of the device, re
■ 3-GaugeMode	UInt	0	0	Value describing the position of the main
■ ▶ 4-Advanced	Array[1..6] of *Ban...			Process Data Advanced Mode
■ Audible	USInt	0	0	0 = Off, 1 = On

- b. Run Mode has ten pieces of data. Animation Type must have a non-zero value for the light to turn on. Depending on the Animation Type, it is necessary to set the color 1 and color 2

parameters. Not all types require color 2. Audible Type and Volume control the output of the audible. The other parameters only affect certain Animation Types.

▼ TL70 IOLM1 01 PD	"Banner_TL70Pro_P...			
■ ▶ 0-SegMode	"Banner_TL70Pro_P...			Process Data Segment Mode
■ ▼ 1-RunMode	"Banner_TL70Pro_P...			Process Data Run Mode
■ Animation Type	USInt	0	5	0 = Off, 1 = Steady, 2 = Flash, 3 = 2 Color Flash,
■ Color 1	USInt	0	10	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 =
■ Color 1 Intensity	USInt	0	0	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Cust
■ Speed	USInt	0	0	0 = Medium, 1 = Fast, 2 = Slow, 3 = Off, 4 = Cust
■ Pulse Pattern	USInt	0	0	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SO
■ Color 2	USInt	0	5	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 =
■ Color 2 Intensity	USInt	0	0	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Cust
■ Scroll Bounce	USInt	0	0	0 = Solid, 1 = Tail, 2 = Ripple.
■ % Width of Color1	USInt	0	0	1 to 100. % Width of color 1.
■ Direction	USInt	0	0	0 = Forward, 1 = Backward.
■ 2-LevelMode	UInt	0	0	Value describing the level of the device, range
■ 3-GaugeMode	UInt	0	0	Value describing the position of the main anim
■ ▶ 4-Advanced	Array[1..6] of "Ban...			Process Data Advanced Mode
■ Audible	USInt	0	0	0 = Off, 1 = On

- c. Level takes a numerical value. This controls how much of the light lights up.

2-LevelMode	UInt	0	60	Value describing the level of the device,
-------------	------	---	----	---

- d. Gauge takes a numerical value. This controls how much of the light lights up.

3-GaugeMode	UInt	0	75
-------------	------	---	----

- e. Advanced allows complete control the . Each Segment opens full parameters like the Run type.

▼ TL70 IOLM1 01 PD	*Banner_TL70Pro_P...			
■ ▶ 0-SegMode	*Banner_TL70Pro_P...			Process Data Segment Mode
■ ▶ 1-RunMode	*Banner_TL70Pro_P...			Process Data Run Mode
■ 2-LevelMode	UInt	0	0	Value describing the level of the device, (
■ 3-GaugeMode	UInt	0	0	Value describing the position of the main
■ ▼ 4-Advanced	Array[1..6] of *Ban...			Process Data Advanced Mode
■ ▼ 4-Advanced[1]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ Animation	UInt	0	3	0 = Off, 1 = Steady, 2 = Flash, 3 = 2 Color (
■ Speed	UInt	0	0	0 = Medium, 1 = Fast, 2 = Slow, 3 = Off, 4 =
■ Pulse Pattern	UInt	0	0	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3
■ Color1	UInt	0	9	0 = Green, 1 = Red, 2 = Orange, 3 = Ambe
■ Color1 Intensity	UInt	0	0	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 =
■ Color2	UInt	0	12	0 = Green, 1 = Red, 2 = Orange, 3 = Ambe
■ Color2 Intensity	UInt	0	0	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 =
■ ▶ 4-Advanced[2]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ ▶ 4-Advanced[3]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ ▶ 4-Advanced[4]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ ▶ 4-Advanced[5]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ ▶ 4-Advanced[6]	*Banner_TL70Pro_P...			Process Data Advanced Mode
■ Audible	UInt	0	0	0 = Off, 1 = On

Appendix A TL70 Pro Process Data

The TL70 Pro has 20 bytes of Process Data Out. There are four modes for displaying this data, as shown below. This Process Data is mapped to a specific group of PROFINET addresses. This function intelligently parses this Process Data into its component pieces.

The first is mode 0, "Segment".

ProcessDataOut "Process Data Out Segment Mode" id=V_Pd_OutSegment

bit length: 160

data type: 160-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 1	The state of the segment. Related parameters defined in Segment Parameter Data
2	8	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 2	The state of the segment. Related parameters defined in Segment Parameter Data
3	16	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 3	The state of the segment. Related parameters defined in Segment Parameter Data
4	24	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 4	The state of the segment. Related parameters defined in Segment Parameter Data
5	32	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 5	The state of the segment. Related parameters defined in Segment Parameter Data
6	40	8-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 6	The state of the segment. Related parameters defined in Segment Parameter Data
7	48	4-bit UInteger	0 = Off, 1 = On					Audible	The state of Audible function

The next mode, "1", is "Run".

ProcessDataOut "Process Data Out Run Mode" id=V_Pd_OutRunMode

bit length: 160

data type: 160-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	8-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Two Color Shift, 5 = Scroll, 6 = Center Scroll, 7 = Bounce, 8 = Center Bounce, 9 = Intensity Sweep, 10 = Two Color Sweep, 11 = Spectrum, 12 = Single End Steady, 13 = Single End Flash					Animation	The Animation type
2	8	8-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 = Daylight White (5000K), 14 = Custom 1, 15 = Custom 2					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	16	8-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	24	8-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom Flash Rate					Speed	The speed of the Animation
5	32	8-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	40	8-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 = Daylight White (5000K), 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	48	8-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	56	8-bit UInteger	0 = Solid, 1 = Tail, 2 = Ripple					Scroll/Bounce Style	The style of scrolling Segment
9	64	8-bit UInteger	1..100 = Percent Width of Color 1					Percent Width of Color 1	The size of scrolling Segment
10	72	8-bit UInteger	0 = Up, 1 = Down					Direction	The direction of Animation
11	80	4-bit UInteger	0 = Off, 1 = On					Audible	The state of Audible function

Mode 2 is "Level".

ProcessDataOut "Process Data Out Level Mode" id=V_Pd_OutLevelMode

bit length: 160

data type: 160-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	16-bit UInteger						Level Mode Value	Value describing the level of the device, range determined in Level Mode Parameter Data
2	16	4-bit UInteger	0 = Off, 1 = On					Audible	The state of Audible function

Mode 3 is "Gauge".

ProcessDataOut "Process Data Out Gauge Mode" id=V_PD_OutGaugeMode

bit length: 160

data type: 160-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	16-bit UInteger						Gauge Mode Value	Value describing the position of the main animation
2	16	4-bit UInteger	0 = Off, 1 = On					Audible	The state of Audible function

Mode 4 is "Advanced". Segment 1 is the data shown in the image.

ProcessDataOut "Process Data Out Advanced" id=V_Pd_OutAdvanced

bit length: 160

data type: 160-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	3-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Intensity Sweep, 5 = Two Color Sweep, 6 = Spectrum					Segment 1 Animation Type	The Animation type
2	3	2-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom Flash Rate					Segment 1 Speed	The speed of the Animation
3	5	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Segment 1 Pulse Pattern	The pattern of Animation
4	8	5-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 = Daylight White (5000K), 14 = Custom 1, 15 = Custom 2					Segment 1 Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
5	13	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
6	16	5-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 = Daylight White (5000K), 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	21	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