

# TLF100 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 TLF100 Pro light via an IO-Link Master from a Siemens PLC. The function covers parsing and display of the TLF100 Pro sensor Process Data Out.

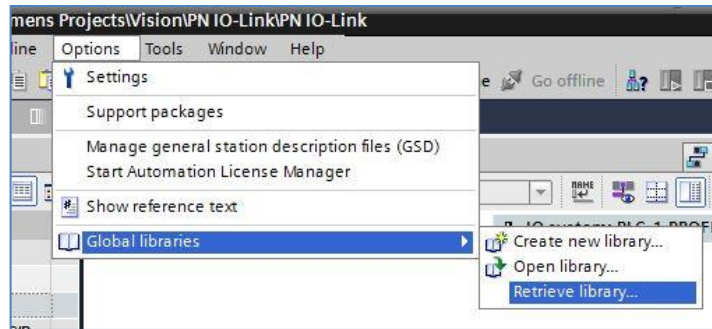
## **Components**

Banner TLF100 Pro.zal14

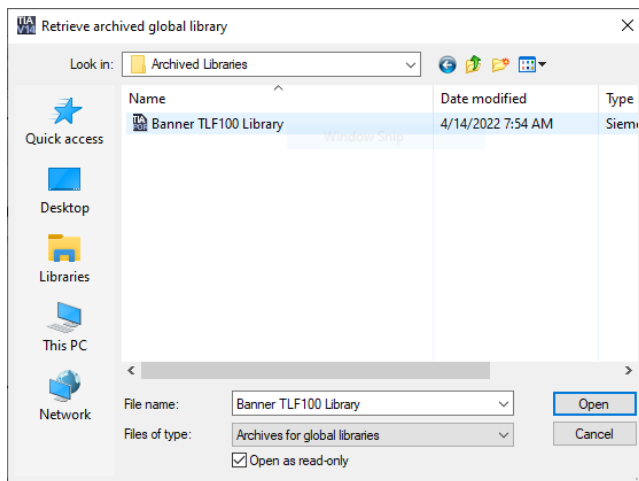
There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

## **Installation Instructions**

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



3. Select the Banner TLF100 Pro. 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 TLF100 Pro with a Banner DXMR90-4K**

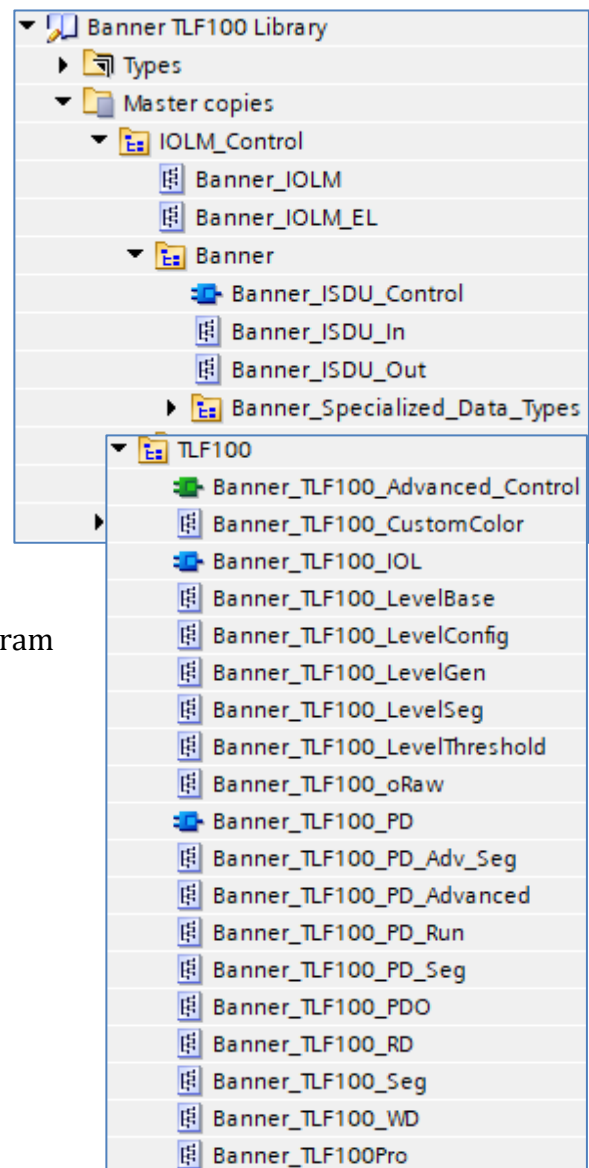
1. Go to Device and Networks to configure the DXMR90-4K. Add the DXMR90-4K if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR90-4K for IO-Link mode.

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
--------------------------------------	---	---	---------	--------	------------------------------------

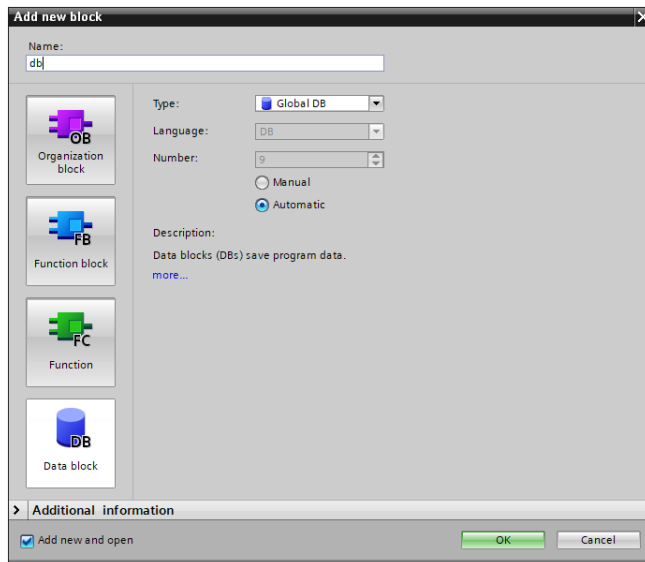
4. Drag the necessary tag from IOLM\_Control > Banner > Banner\_Specialized\_Data\_Types. The tag used in this example is "Banner\_32out". This tag represents the full raw process data along with port status information.
5. Drag the necessary files from the TLF100 Pro Folder.
  - a. Move Banner\_TLF100\_oRaw, Banner\_TLF100\_PD, Banner\_TLF100\_PD\_Adv\_Seg, Banner\_TLF100\_Advanced, Banner\_TLF100\_PD\_Run, Banner\_TLF100\_PD\_Seg, and Banner\_TLF100\_PDO to the PLC Data Types area under your PLC.
  - b. Move Banner\_TLF100\_PD and Banner\_TLF100\_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 “TLF100 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 “TLF100 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
▶ TLF100 IOLM1 01 PDO	"Banner_32Out"	%Q1.0
▶ TLF100 IOLM1 01 outRaw	"Banner_TLF100_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\_TLF100\_PDO” for the new tag.

Name	Data type
▼ Static	
■ ▼ TLF100 IOLM1 01 PD	"Banner_TLF100_PDO"
■ ▶ 0-SegMode	"Banner_TLF100_PD_Seg"
■ ▶ 1-RunMode	"Banner_TLF100_PD_Run"
■ 2-LevelMode	UInt
■ ▶ 3-Advanced	"Banner_TLF100_PD_Advanced"

9. Add the “Banner\_TLF100\_PD” function to an OB ladder. Link the “ProcessData” 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 “TLF100 IOLM1 01 outRaw” in this example. The “PDO” needs to be linked to the variable created in step 7. It was called “TLF100 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 TLF100, there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TLF100 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 TLF100 Process Data Function to the TLF100 Parameter Data Function Block (see Fig. 2). See Appendix A for more information about TLF100 Pro Process Data Out.

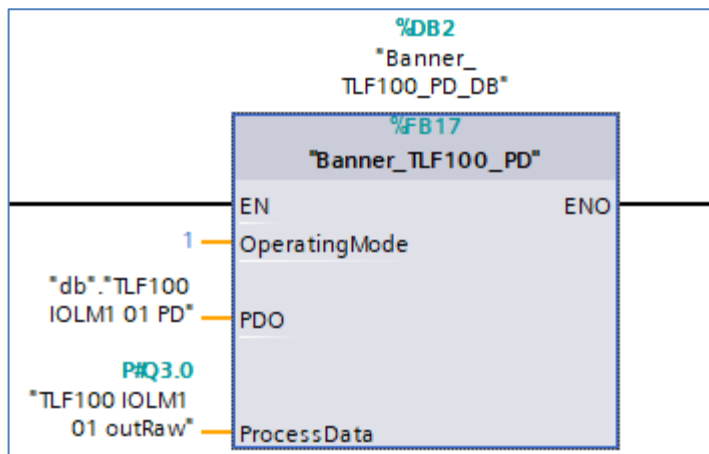


Figure 1: Hand type correct number for Operational Mode

**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” (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” (Advanced Segment; where the four segments can be controlled individually). The default is “1”.

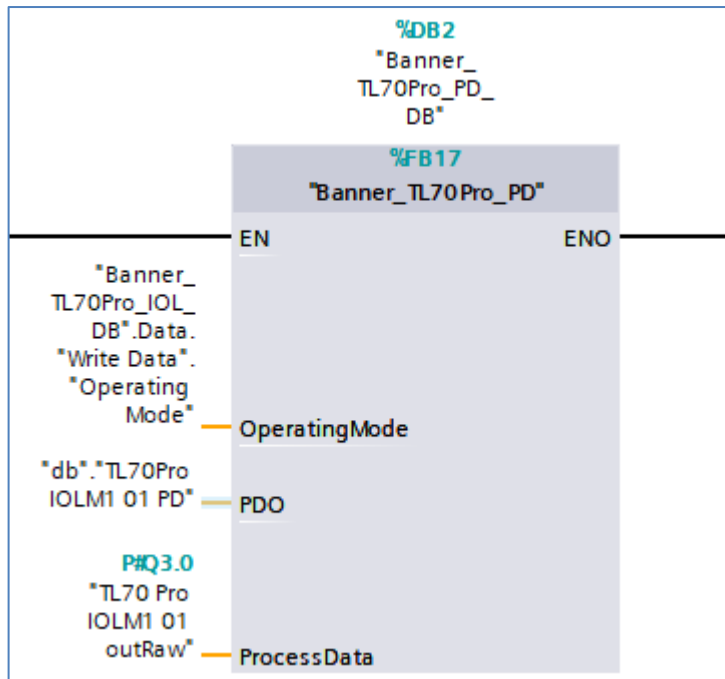
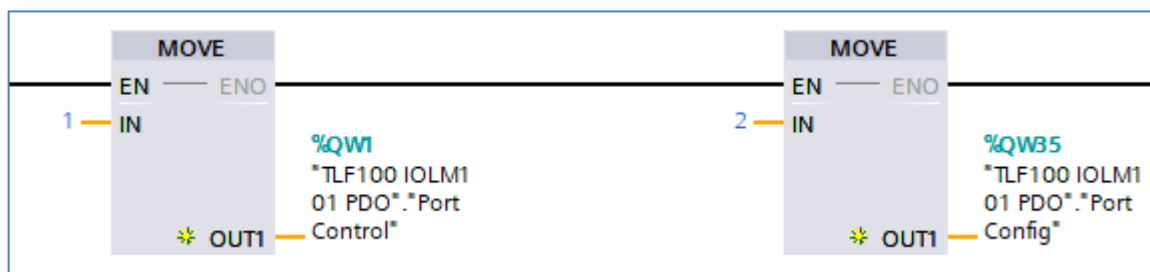


Figure 2: Linking Operational Mode variable to TLF100 Pro Parameter Data Function Block

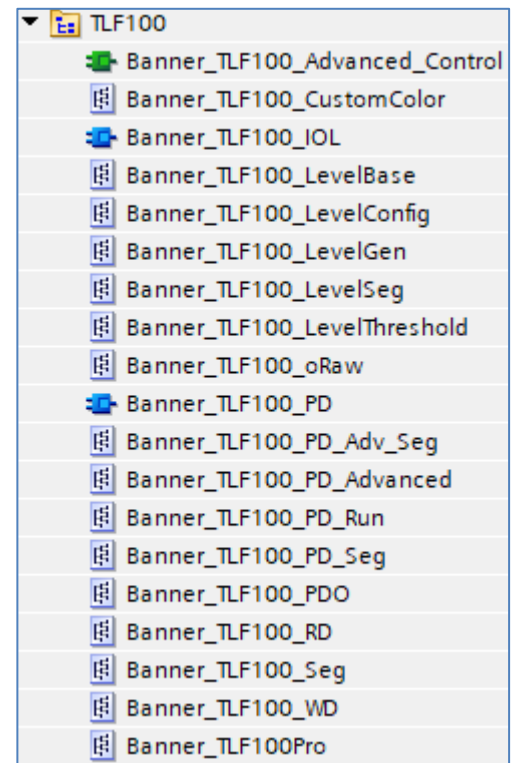
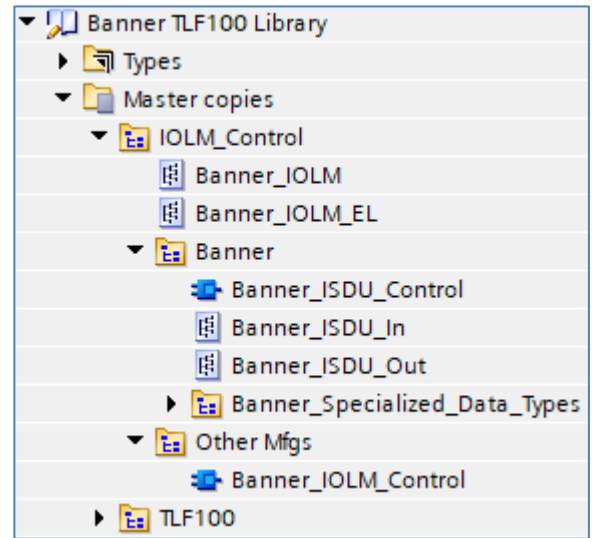
10. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 “TLF100 IOLM1 01 PDO”.



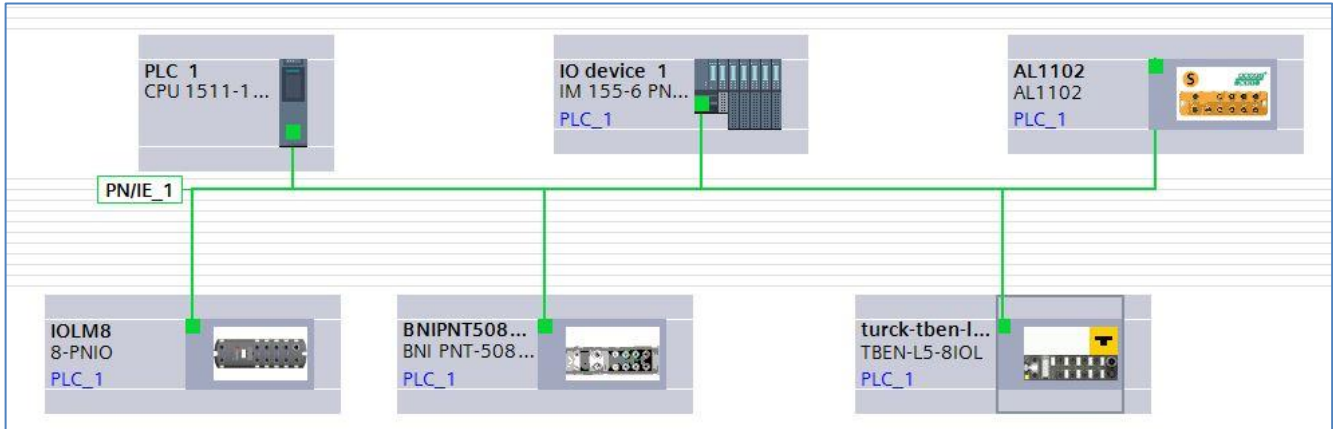
11. Process Data Setup is complete.  
12. Compile and download the configuration to the PLC, then go online. Open the “db” data block and click Monitor all. The TLF100 Pro can be controlled now.

### **Setup of TLF100 Pro with other IO-Link Masters**

1. The Banner TLF100 Pro library will now be in the Global Library List. Expand the Master copies section. The K50 Pro Audible folder contains elements for both Process Data and Parameter Data connections to a K50 Pro Audible device. As Process Data is the focus of this paper, we will concern ourselves with these eight items:  
Banner\_TLF100\_Advanced\_Control,  
Banner\_TLF100\_oRaw, Banner\_TLF100\_PD,  
Banner\_TLF100\_PD\_Adv\_Seg,  
Banner\_TLF100\_Advanced,  
Banner\_TLF100\_PD\_Run,  
Banner\_TLF100\_PD\_Seg, and  
Banner\_TLF100\_PDO.
2. Drag Banner\_TLF100\_Advanced\_Control and Banner\_TLF100\_PD to the Program Blocks area under your PLC.
3. Drag Banner\_TLF100\_oRaw, Banner\_TLF100\_PD, Banner\_TLF100\_PD\_Adv\_Seg, Banner\_TLF100\_Advanced, Banner\_TLF100\_PD\_Run, Banner\_TLF100\_PD\_Seg, and Banner\_TLF100\_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.

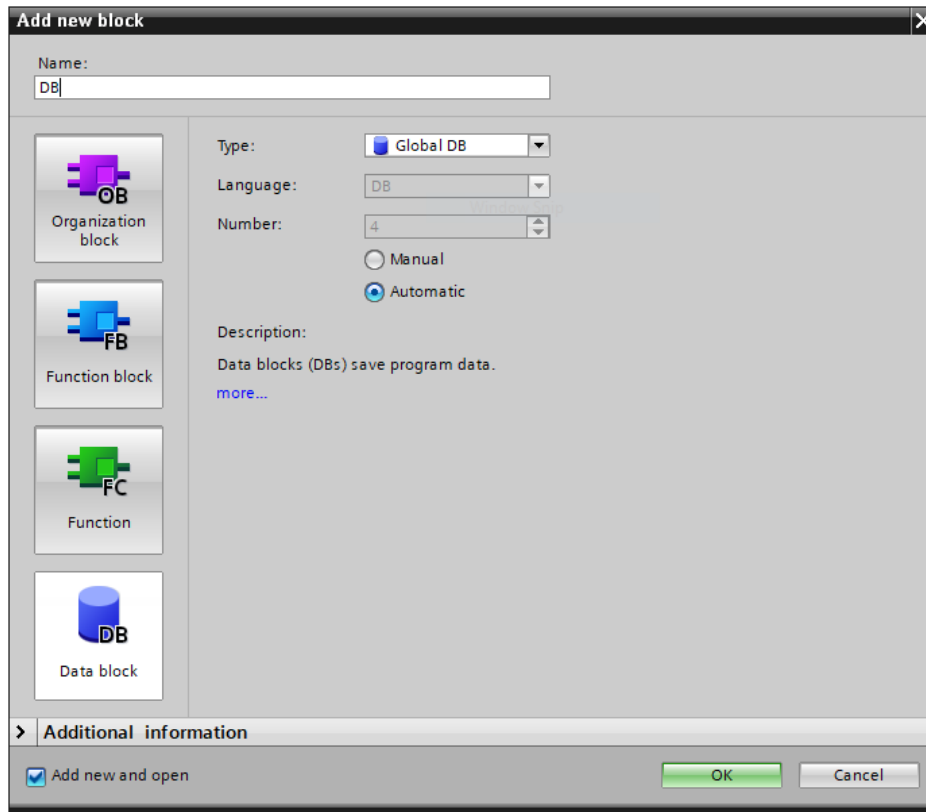


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a TLF100 Pro requires 22 bytes of space for the Process Data Out. This will likely require a 32 byte OUT type.
6. Record the “Q” addresses where this TLF100 Pro Process Data is to be stored, as these addresses will be required in the next step. In this example, 22 bytes of Process Data Out for port 2 on the IO-Link Master will be stored in Q1 through Q22.
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 “TLF100 IOLM1 01 PDO” was created using a Data Type of “Banner\_TLF100\_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
▶ TLF100 IOLM1 oRaw	"Banner_TLF100_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 TLF100 Pro. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type "Banner\_TLF100\_PDO" for the new tag.

Name	Data type
▼ Static	
■ ▶ TLF100 IOM1 01 PD	"Banner_TLF100_PDO"

10. Add the "Banner\_TLF100\_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 TLF100 Audible, there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TLF100 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 TLF100 Pro Process Data Function to the K50 Pro Audible Parameter Data Function Block (see Fig. 2). See Appendix A for more

information about TLF100 Pro Process Data Out.

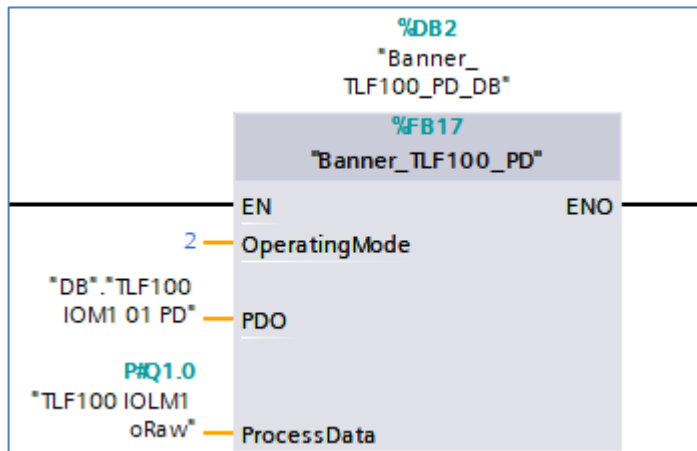


Figure 3: Hand type correct number for Operational Mode

**NOTE:** if you type in the incorrect number (i.e. it does not match the TLF100's current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

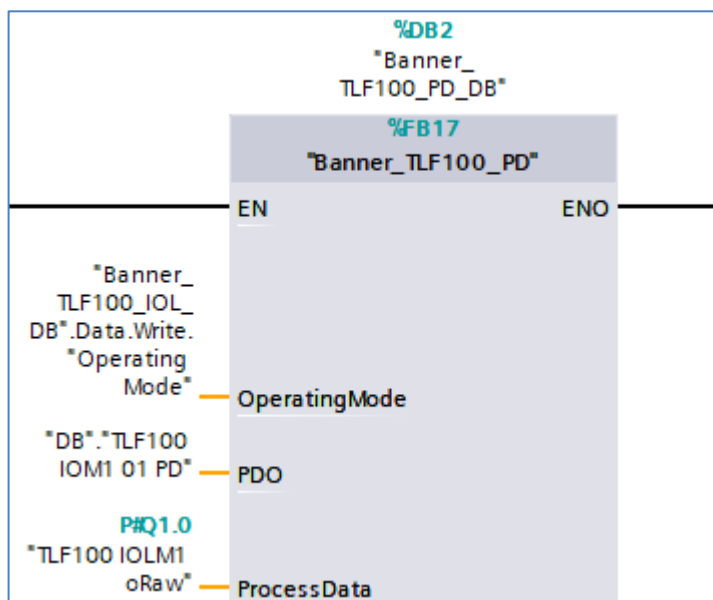


Figure 4: Linking Operational Mode variable to TLF100 Pro Parameter Data Function Block

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 TLF100 Pro Process Data.

13. Process Data is broken up into four different types. When Operational Modes controls which of four types is used to parse the raw byte data.

Name	Data type
▼ Static	
■ ▼ TLF100 IOM1 01 PD	"Banner_TLF100_PDO"
■ ▶ 0-SegMode	"Banner_TLF100_PD_Seg"
■ ▶ 1-RunMode	"Banner_TLF100_PD_Run"
■ ▶ 2-LevelMode	UInt
■ ▶ 3-Advanced	"Banner_TLF100_PD_Advanced"

0: Segment  
1: Run  
2: Level  
3: Advanced

- a. Segment has five pieces of data. Each Segment allows a value of 0 for OFF and 1 for ON. Audible allows a value of 0 for OFF and 1 for ON.

▼ 0-SegMode	"Banner_TLF100_P..."		
■ Segment1	USInt	0	1
■ Segment2	USInt	0	1
■ Segment3	USInt	0	0
■ Segment4	USInt	0	0
■ Audible	USInt	0	0

- b. Run Mode has eleven 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.

▼ 1-RunMode	"Banner_TLF100_P..."		
■ Animation Type	USInt	0	2
■ Color 1	USInt	0	0
■ Color 1 Intensity	USInt	0	0
■ Speed	USInt	0	0
■ Pulse Pattern	USInt	0	0
■ Color 2	USInt	0	0
■ Color 2 Intensity	USInt	0	0
■ % Width of Color1	USInt	0	0
■ Direction	USInt	0	0
■ Audible Type	USInt	0	0
■ Audible Volume	USInt	0	0

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

2-LevelMode	UInt	0	0
-------------	------	---	---

- d. Advanced allows complete control the TLF100. Each Segment opens full parameters like the Run type.

Name	Data type	Start value	Monitor value
■ ▼ 3-Advanced	*Banner_TLF100_P...		
■ ▼ Segment 1	*Banner_TLF100_P...		
■ Animation	USInt	0	1
■ Color1	USInt	0	9
■ Color1 Intensity	USInt	0	0
■ Color2	USInt	0	0
■ Color2 Intensity	USInt	0	0
■ Speed	USInt	0	0
■ Pulse Pattern	USInt	0	0
■ Scroll Bounce	USInt	0	0
■ Direction	USInt	0	0
■ % Width of Col...	USInt	0	0
■ ▶ Segment 2	*Banner_TLF100_P...		
■ ▶ Segment 3	*Banner_TLF100_P...		
■ ▶ Segment 4	*Banner_TLF100_P...		
■ Audible Type	USInt	0	0
■ Audible Volume	USInt	0	0

**Appendix A****TLF100 Pro Process Data**

The TLF100 Pro has 22 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: 176

data type: 176-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	0 = Off, 1 = On					Segment 1	The state of the segment. Related parameters defined in Segment Parameter Data
2	16	16-bit UInteger	0 = Off, 1 = On					Segment 2	The state of the segment. Related parameters defined in Segment Parameter Data
3	32	16-bit UInteger	0 = Off, 1 = On					Segment 3	The state of the segment. Related parameters defined in Segment Parameter Data
4	48	16-bit UInteger	0 = Off, 1 = On					Segment 4	The state of the segment. Related parameters defined in Segment Parameter Data
5	64	16-bit UInteger	0 = Off, 1 = On					Audible	The state of the audible. Related parameters defined in Basic Audible Parameter Data

The next mode, “1”, is “Run”.

#### ProcessDataOut "Process Data Out Run Mode" id=V\_Pd\_OutRunMode

bit length: 176

data type: 176-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 = Scroll, 5 = Bounce, 6 = Intensity Sweep, 7 = Two Color Sweep, 8 = Spectrum					Animation	The Animation type
2	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					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	13	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	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					Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
5	21	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
6	24	2-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom Flash Rate					Speed	The speed of the Animation
7	26	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
8	29	2-bit UInteger						Reserved	Reserved
9	31	Boolean	false = Up, true = Down					Direction	The direction of Animation
10	32	8-bit UInteger	1..100 = Percent Width of Color 1					Percent Width of Color 1	The size of scrolling Segment
11	40	8-bit UInteger	0 = Pulse, 1 = Wobble, 2 = Strobe, 4 = Whoop, 5 = Stacatto, 6 = Siren, 8 = Continuous 1, 9 = Continuous 2, 12 = Sync, 16 = Jingle, 17 = Melody 1, 18 = Melody 2, 19 = Melody 3, 20 = Custom					Audible Type	The audible tone to be played when active
12	48	8-bit UInteger	0 = Off, 1 = Low, 2 = Medium, 3 = High					Audible Volume	The audible volume

Mode 2 is “Level”.

#### ProcessDataOut "Process Data Out Level Mode" id=V\_Pd\_OutLevelMode

bit length: 176

data type: 176-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

Mode 3 is "Advanced". Segment 1 is the data shown in the image.

### ProcessDataOut "Process Data Out Advanced Segment Mode" id=V\_Pd\_OutAdvancedSegmentMoc

bit length: 176

data type: 176-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 = Ends Steady, 6 = Ends Flash, 7 = Scroll, 8 = Center Scroll, 9 = Bounce, 10 = Center Bounce, 11 = Intensity Sweep, 12 = Two Color Sweep, 13 = Spectrum, 14 = Level Steady, 15 = Level Flash					Segment 1 Animation	The Animation type
2	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
3	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
4	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
5	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
6	24	2-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom Flash Rate					Segment 1 Speed	The speed of the Animation
7	26	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Segment 1 Pulse Pattern	The pattern of Animation
8	29	2-bit UInteger	0 = Solid, 1 = Tail, 2 = Ripple					Segment 1 Scroll/Bounce Style	The style of scrolling Segment
9	31	Boolean	false = Up, true = Down					Segment 1 Direction	The direction of Animation
10	32	8-bit UInteger	1..100 = Percent Width of Color 1					Segment 1 Percent Width of Color 1	The size of scrolling Segment
11	40	8-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Two Color Shift, 5 = Ends Steady, 6 = Ends Flash, 7 = Scroll, 8 = Center Scroll, 9 = Bounce, 10 = Center Bounce, 11 = Intensity Sweep, 12 = Two Color Sweep, 13 = Spectrum, 14 = Level Steady, 15 = Level Flash					Segment 2 Animation	The Animation type