

# LTF Process Data Function

---

11/21/2022

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 In from a Banner LTF sensor via an IO-Link Master to a Siemens PLC. The function covers parsing and display of the LTF sensor Process Data In.

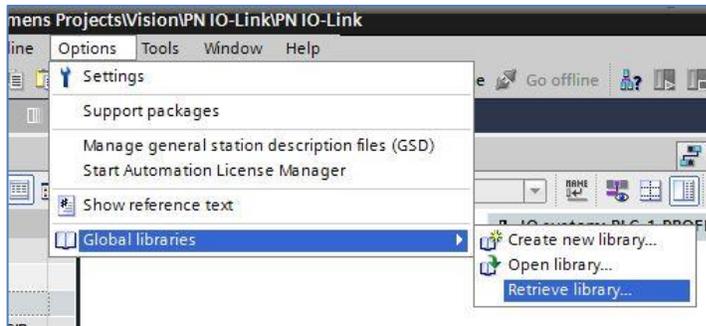
## **Components**

Banner LTF Library.zal14

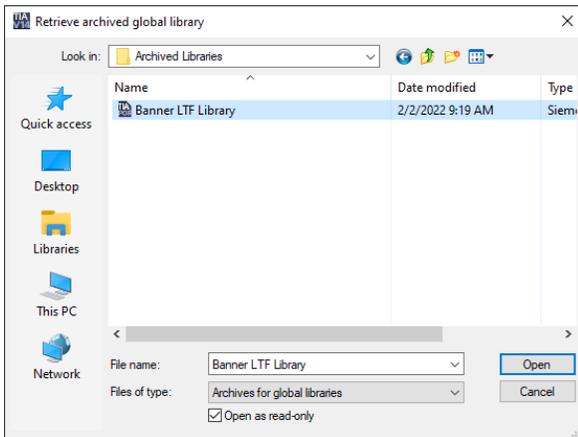
There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

## **Installation Instructions**

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



3. Select the Banner LTF Library. Click Open.



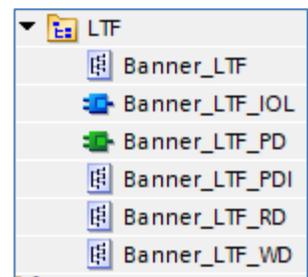
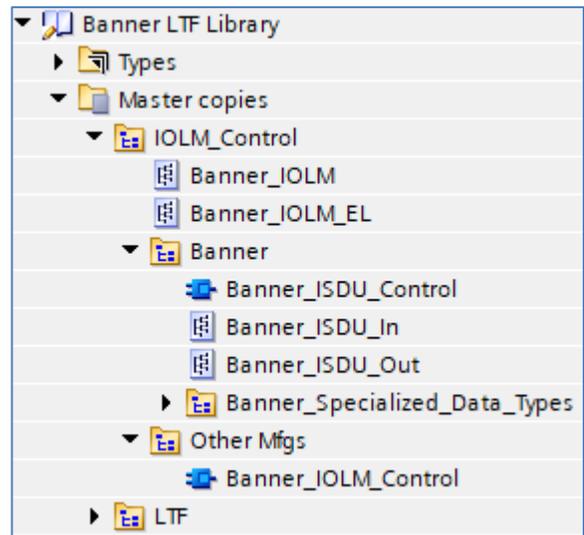
4. The library is now accessible in the Libraries tab.
5. Go to page 3 for Banner IO-Link Masters and to page 6 for all other IO-Link Masters.

**Setup of LTF 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 4/4 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.

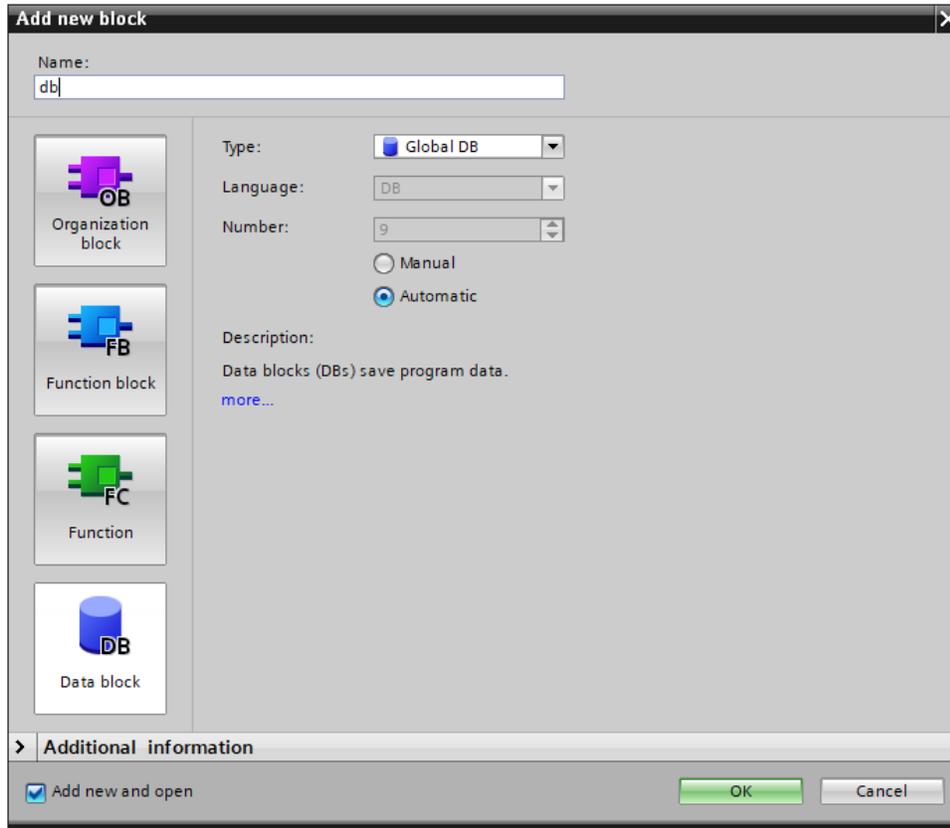
Module	Rack	Slot	I address	Q address	Type
▼ dxm	0	0			1-port Device
▶ Interface	0	0 X1			dxm
Banner IO-Link Master Info_1	0	1	1...9		Banner IO-Link Master Info
IO-Link In/Out 4/ 4 Byte + Status_1	0	2	10...17	1...18	IO-Link In/Out 4/ 4 Byte + Status

4. Drag the Banner\_LTF\_PDI to the PLC Data Types area under your PLC. Banner\_LTF\_PDI is found in the LTF folder in the library. Drag the Banner\_LTF\_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\_4in". 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 "LTF IOLM1 01 PDI" was created using a Data Type of "Banner\_4In". 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 "LTF IOLM1 01 inRaw". This is the tag that will be used in the Function block.



Name	Data type	Address
▶ LTF IOLM1 01 PDI	"Banner_4In"	%I10.0
LTF IOLM1 01 inRaw	DWord	%ID14

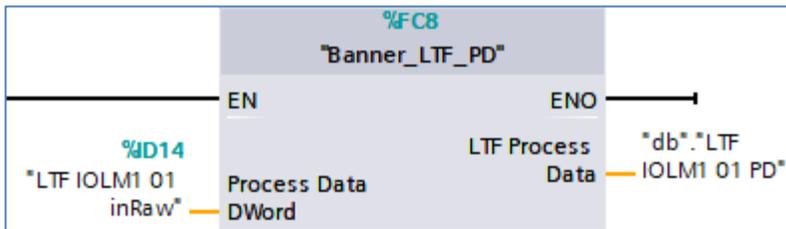
- Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named "db".



- In the new data block, create a new tag to represent the parsed Process Data In for our LE. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type "Banner\_LTF\_PDI" for the new tag.

Name	Data type
▼ Static	
■ ▼ LTF IOLM1 01 PD	"Banner_LTF_PDI"
■ Channel 1 Output State	Bool
■ Channel 2 Output State	Bool
■ Stability	Bool
■ Measurement Value	Real

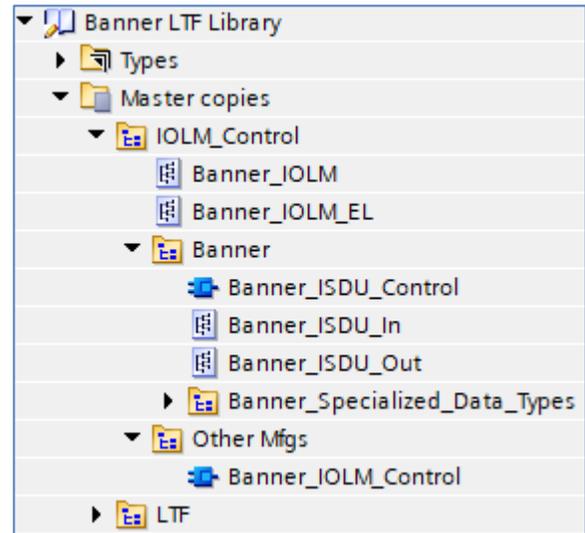
9. Add the "Banner\_LTF\_PD" function to an OB ladder. Link the "Process Data Word" to the raw Process Data variable from step 4. Link the "LTF Process Data" to the parsed Process Data variable from step 6.



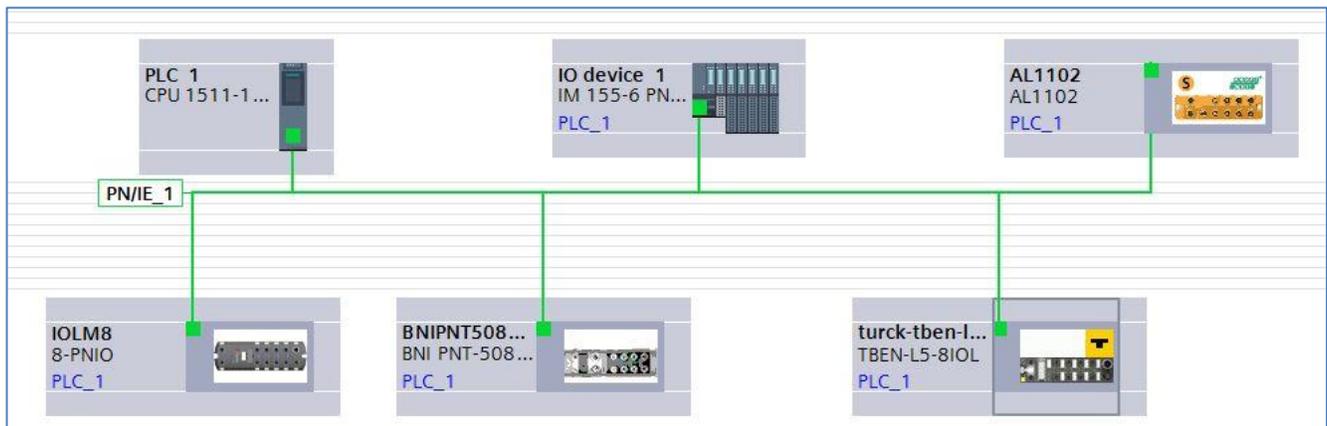
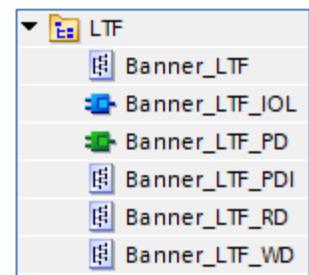
10. Process Data setup is complete.
11. Compile and download the configuration to the PLC, then go online. Open the "db" data block and click Monitor all. You should see parsed LTF Process Data In, like that shown below.

### Setup of LTF with other IO-Link Masters

1. The Banner LTF Library will now be in the Global Library List. Expand the Master copies section. The LTF folder contains elements for both Process Data and Parameter Data connections to an LTF sensor. As Process Data is the focus of this paper, we will concern ourselves with these two items: Banner\_LTF\_PD and Banner\_LTF\_PDI.
2. Drag Banner\_LTF\_PD to the Program Blocks area under your PLC.
3. Drag the Banner\_LTF\_PDI to the PLC Data Types area under your PLC.

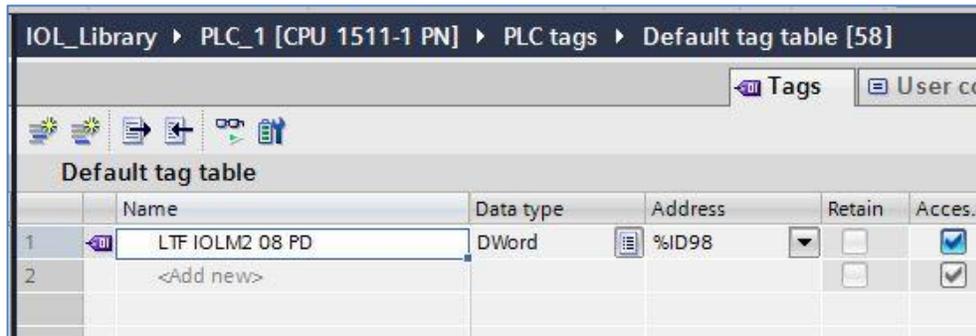


4. Go to Devices and networks to configure the system as necessary. Below is an example of what a configuration might look like. This example shows 5 different IO-Link Masters connected to the same PLC.

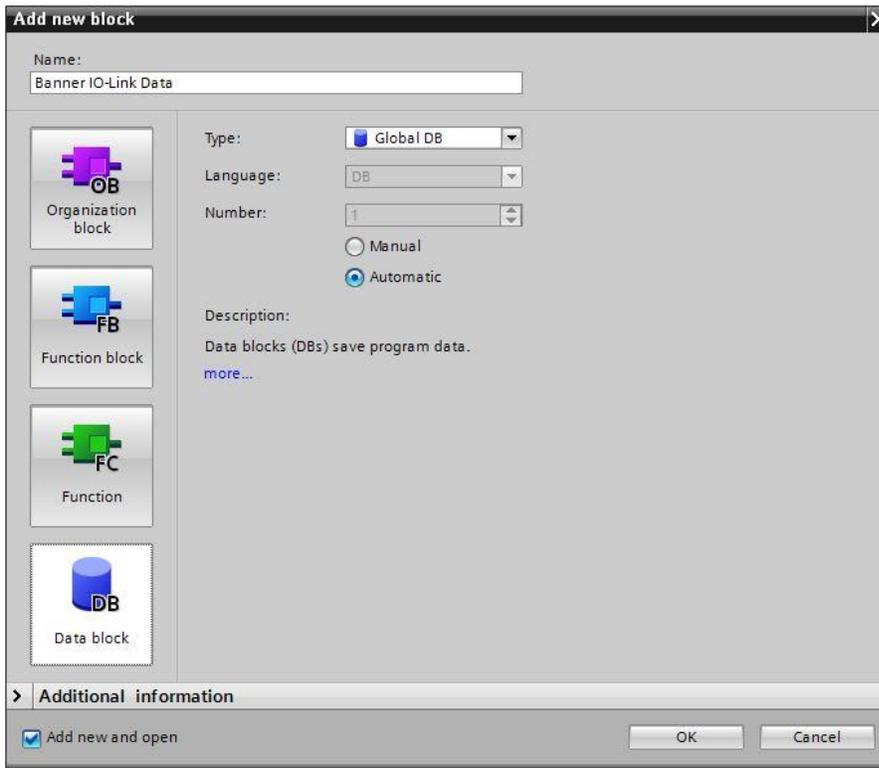


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that an LTF requires 4 bytes of space for the Process Data.
6. Record the "I" address where this LTF Process Data is to be stored, as the address will be required in the next step. In this example, 4 bytes of Process Data In for port 8 on the IO-Link Master will be stored in I98 through I101.

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 “LTF IOLM2 08 PD” was created using a Data Type of “DWord”. 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 IOLM3, 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.



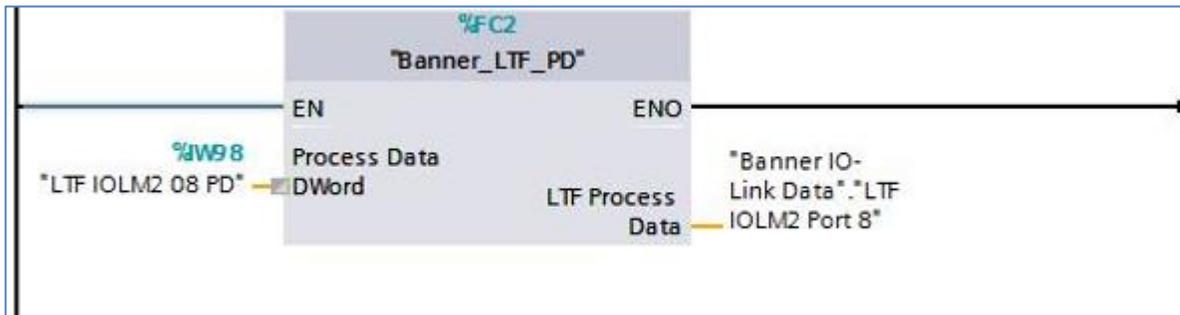
8. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “Banner IO-Link Data”.



- In the new data block, create a new tag to represent the parsed Process Data In for our LTF. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type "Banner\_LTF\_PDI" for the new tag.

Banner IO-Link Data				
	Name	Data type	Start value	Snapshot
1	Static			
2	LTF IOLM2 Port 8	*Banner_LTF_PDI*		

- Add the "Banner\_LTF\_PD" function to an OB ladder. Link the "Process Data DWord" to the raw Process Data variable from step 10. Link the "LTF Process Data" to the parsed Process Data variable from step 12.



- Process Data setup is complete.
- 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 LTF Process Data In, like that shown below.

Banner IO-Link Data				
	Name	Data type	Start value	Monitor value
1	Static			
2	LTF IOLM2 Port 8	*Banner_LTF_PDI*		
3	Channel 1 Output State	Bool	false	TRUE
4	Channel 2 Output State	Bool	false	FALSE
5	Stability	Bool	false	TRUE
6	Measurement Value	Real	0.0	542.7

The LTF has 4 bytes of Process Data In, as shown below.

**ProcessDataIn "Process Data" id=PD\_ProcessDataIn**  
 bit length: 32  
 data type: 32-bit Record (subindex access not supported)

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	Boolean	false = Inactive, true = Active					Channel 1 Output State	Channel 1 Output State
2	1	Boolean	false = Inactive, true = Active					Channel 2 Output State	Channel 2 Output State
3	2	Boolean	false = No target or Marginal, true = Stable					Stability	Stability state
4	3	29-bit UInteger						Measurement Value	The current measured distance in tenths of a millimeter.

This Process Data is mapped to a specific group of EtherNet/IP registers. The 32-bits of Process Data actually encode four separate pieces of information. Bit 0 is the state of BDC1 (Binary Data Channel 1, also known simply as Output Channel 1). Bit 1 is BDC2 (Channel 2). Bit 2 is the stability indicator. The remaining 29 bits are used to communicate the LTF measurement value (in 0.1mm increments).

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