

R95C 8UI Process Data Function

January 12th, 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 Out to a Banner R95C 8UI light via an IO-Link Master from Siemens PLC. The function covers parsing and display of the R95C 8UI sensor Process Data Out.

Components

Banner R90C R95C R130C Hubs v16.zal16

There are two methods for 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 manufacturers' 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.



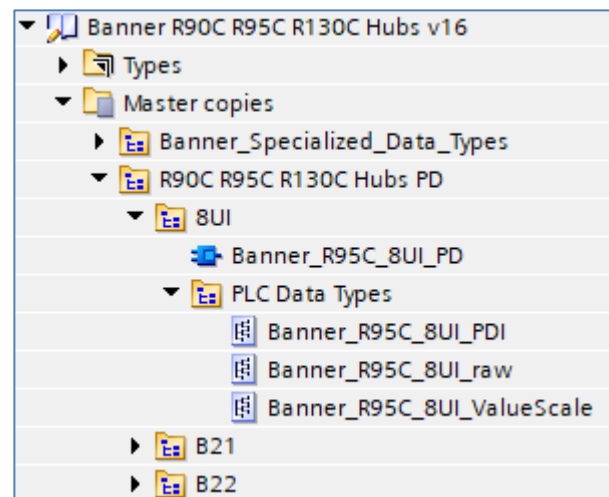
3. Switch the “Files of type” to Compressed libraries. Go to the location of the compressed library.
4. Press the Open button and the library will be uncompressed and opened.
5. The library is now accessible in the Libraries tab in v16 or greater.

Setup of R95C 8UI 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 32/32 byte is required for SD50 Select. Make note of the Q address for Slot 2 which represents Port 1 (%Q2).

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	68...76		Banner IO-Link Master Info
IO-Link In/Out 32/32 Byte + Status_1	0	2	2...37	2...47	IO-Link In/Out 32/32 Byte + Status

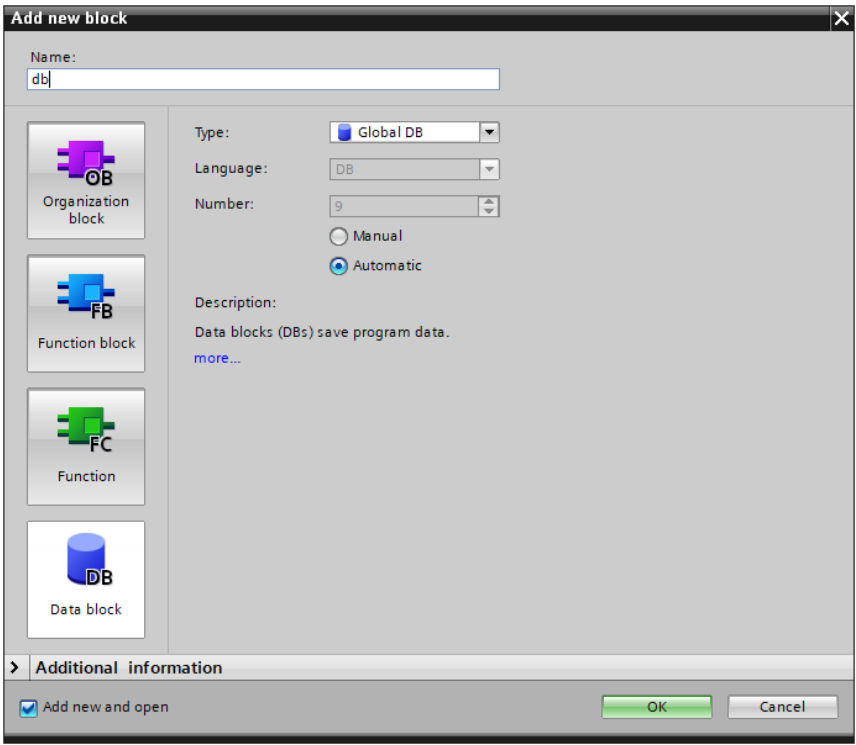
4. Drag the necessary tag from Banner_Specialized_Data_Types. The tag used in this example is "Banner_32In". This tag represents the full raw process data along with port information.
5. Drag the necessary files from the 8UI Folder under the R90C R95C R130C Hubs PD folder.
 - a. Move Banner_R95C_8UI_PDI, Banner_R95C_8UI_raw, and Banner_R95C_8UI_ValueScale to the PLC Data Types area.
 - b. Move Banner_R95C_8UI_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 "R95C 8UI IOLM1 01 PDI" was created using a Data Type of "Banner_32In". 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 3 (%I10) is tied to this new tag. The second is "R95C 8UI IOLM1 01 inRaw" and uses the "I" address found in step 3 plus 4 (%I14). This is the tag that will be used in the Function block.

Name	Data type	Address
▶ R95C 8UI IOLM1 01 PDI	"Banner_32In"	%I10.0
▶ R95C 8UI IOLM1 01 inRaw	"Banner_R95C_8UI_raw"	%I14.0

- 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 R95C 8UI. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_R95C_8UI_PDI” for the new tag.

▼ R95C 8B22 IOLM1 01 PD	"Banner_R95C_8UI_PDI"
■ ▶ Measurement	Array[1..8] of UDInt
■ ▶ Measurement Scale	Array[1..8] of "Banner_R95C_8UI_ValueScale"

9. Add the “Banner_R95C_8UI_PD” function to an OB ladder. Link the “PDI” to the tag “R95C 8UI IOLM1 01 inRaw” created from step 6. Link “R95C 8UI PD” to the tag “db.R95C 8UI IOLM1 01 PD” created from step 8.

The last variable, “PDI Config”, allows the function to correctly interpret the Process Data In. In the case of the R95C 8UI, there are two user-selected modes for the Process Data Out. This function needs to know what choice has been made in the R95C 8UI for this PDI Config 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 R95C 8UI Process Data Function to the R95C 8UI Parameter Data Function Block (see Fig. 2). See Appendix A for more information about R95C 8UI Process Data Out.

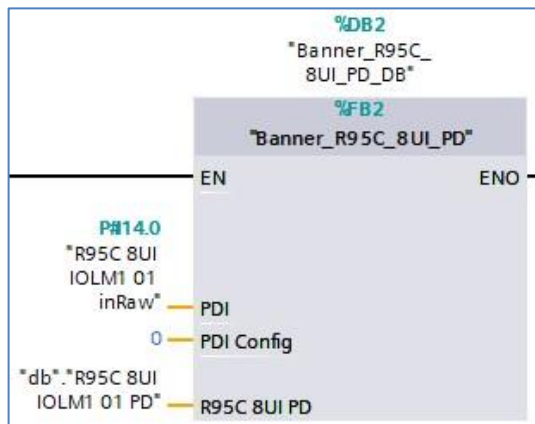


Figure 1: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the display module, R95C 8UI, current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

Operational Mode: the options here are “0” (Analog Value Mode) and “1” (Digital Measuring Sensor Mode). The default is “0”.

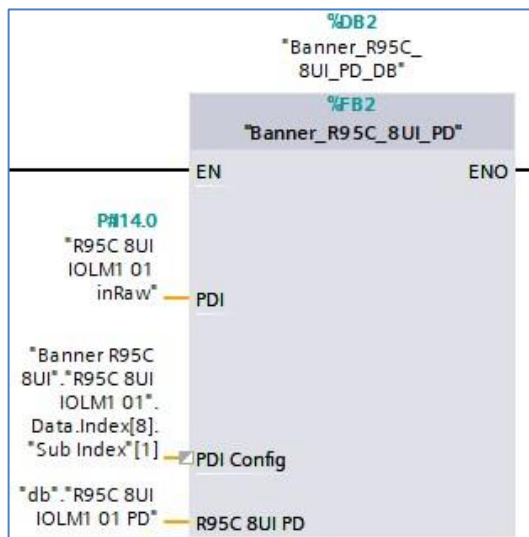


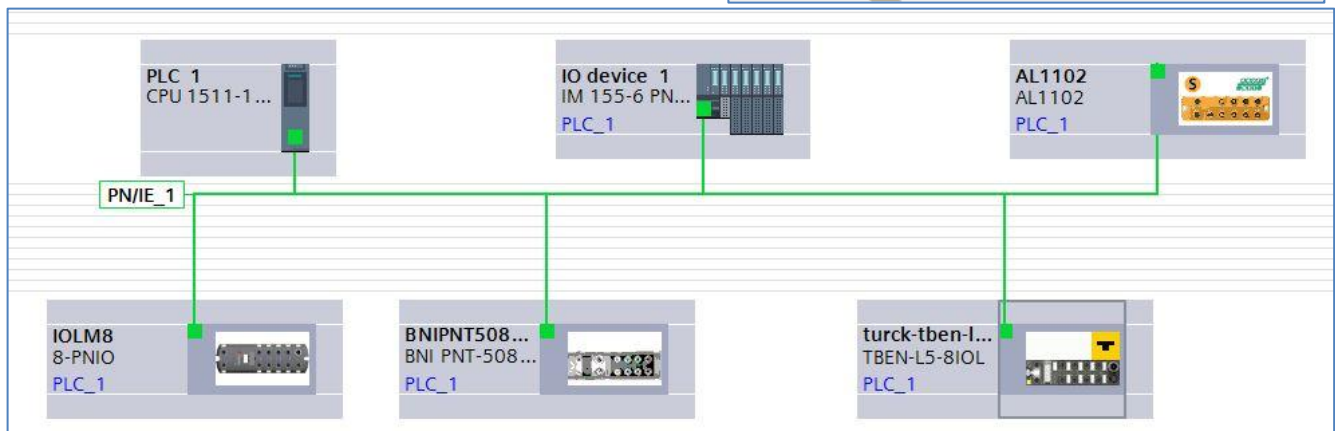
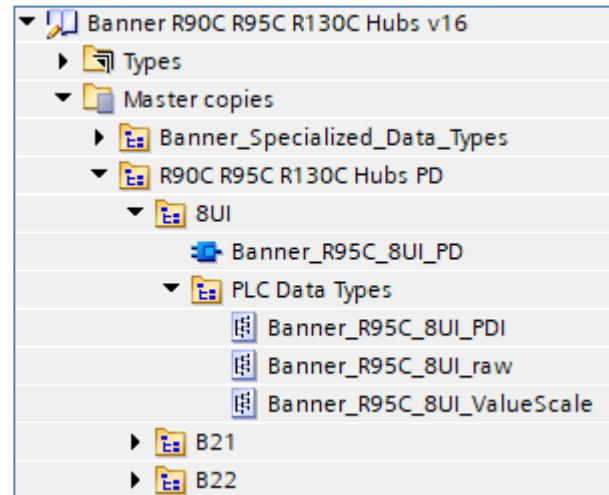
Figure 2: Linking PDI Config variable to R95C Parameter Data Function Block

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 can now control the R95C 8UI via the data located here.

▼ R95C 8UI IOLM1 01 PD	*Banner_R95C_8UI_PDI*		
■ ▼ Measurement	Array[1..8] of UDInt		
■ Measurement[1]	UDInt	0	6045
■ Measurement[2]	UDInt	0	37
■ Measurement[3]	UDInt	0	43
■ Measurement[4]	UDInt	0	15
■ Measurement[5]	UDInt	0	31
■ Measurement[6]	UDInt	0	41
■ Measurement[7]	UDInt	0	37
■ Measurement[8]	UDInt	0	35
■ ► Measurement Scale	Array[1..8] of *Banner_...		

Setup of R95C 8UI with other IO-Link Masters

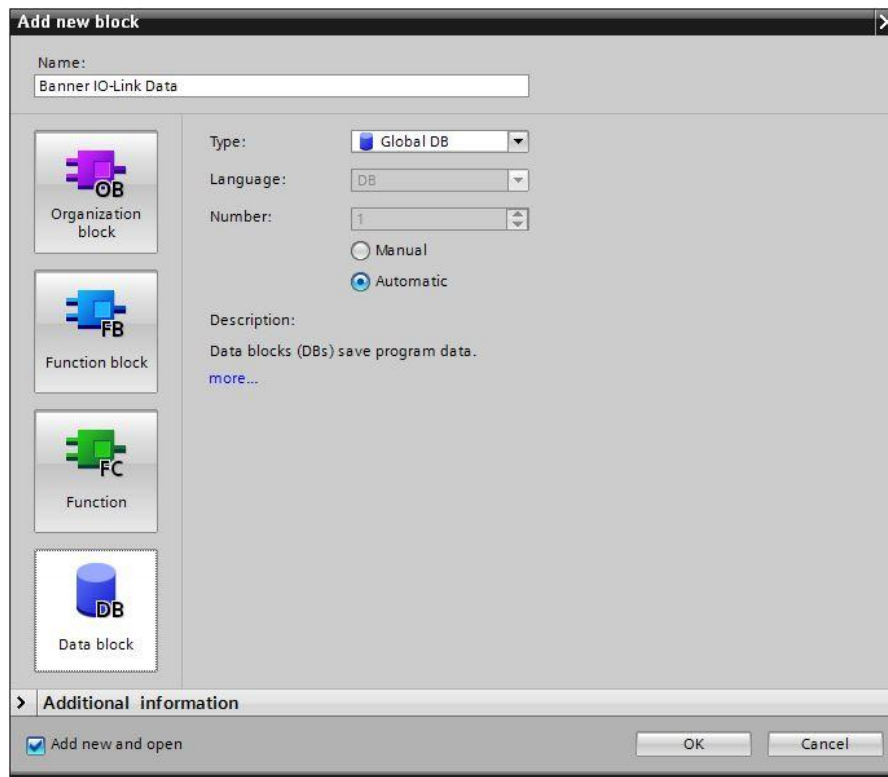
1. The Banner R95C Hubs Library will now be in the Global Library List. Expand the Master copies section.
2. Drag Banner_R95C_8UI_PD to the Program Blocks area under your PLC.
3. Drag Banner_R95C_8UI_PDI, Banner_R95C_8UI_raw, and Banner_R95C_8UI_ValueScale to the PLC Data Types area under your PLC.
4. 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 R95C 8UI requires 32 bytes of space for the Process Data.
6. Record the "Q" address where this R95C 8UI Process Data is to be stored, as the address will be required in the next step. In this example, 32 bytes of Process Data In for port 1 on the IO-Link Master will be stored starting at Q68.
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 "R95C 8UI IOLM1 01 inRaw" was created using a Data Type of "Banner_R95C_8UI_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 IOLM2 or IOLM3, for instance, and other specific sensors may be connected to different port numbers. Reference IO-Link Documentation for addressing. This example uses %Q2.

▶ R95C 8UI IOLM1 01 inRaw	"Banner_R95C_8UI_raw"	%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".



- In the new data block, create a new tag to represent the parsed Process Data In for our R95C 8UI. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_R95C_8UI_PDI” for the new tag.

▼ R95C 8B22 IOLM1 01 PD	*Banner_R95C_8UI_PDI*
■ ▶ Measurement	Array[1..8] of UDInt
■ ▶ Measurement Scale	Array[1..8] of *Banner_R95C_8UI_ValueScale*

Add the “Banner_R95C_8UI_PD” function to an OB ladder. Link the “PDI” to the raw Process Data variable from step 7. Link “R95C 8UI PD” to the parsed Process Data variable from step 9.

The last variable, “PDI Config”, allow the function to correctly interpret the Process Data In. In the case of the R95C 8UI, there are four user-selected modes for the Process Data In. This function needs to know what choice has been made in the R95C 8UI for this “PDI Config” variable.

There are two ways to achieve this goal. We can simply type in the correct number for “PDI Config” (see Fig. 3), or we can link this R95C 8UI Process Data Function to the R95C 8UI Parameter Data Function Block (see Fig. 4). See Appendix A for more information about R95C 8UI Process Data.

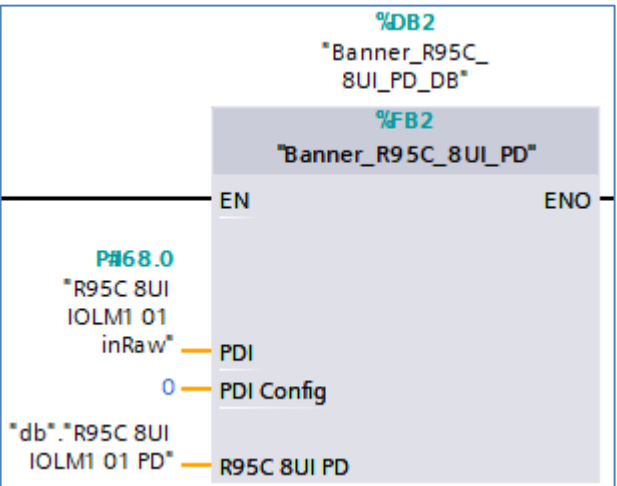


Figure 3: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the display module, SD50, current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

Operational Mode: the options here are “0” (Analog Value Mode) and “1” (Digital Measuring Sensor Mode). The default is “0”.

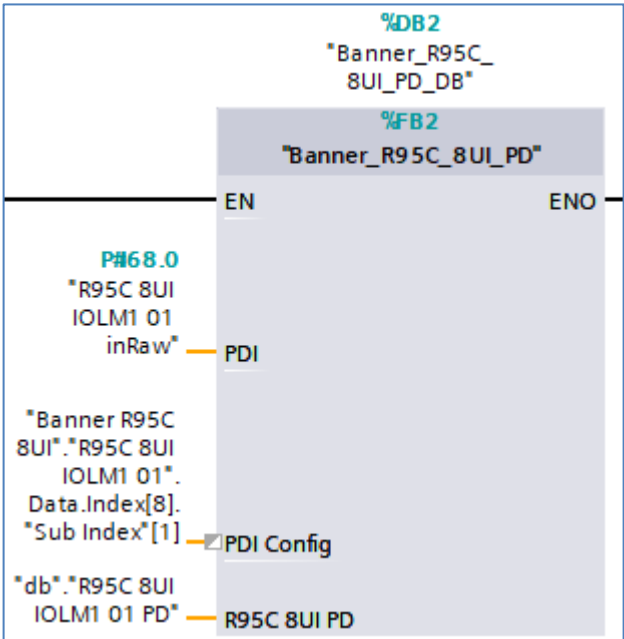


Figure 4: Linking Operational Mode variable to R95C 8UI Parameter Data Function Block

- 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 R95C 8UI Process Data In, like that shown below.

▼ R95C 8UI IOLM1 01 PD	*Banner_R95C_8UI_PDI*		
■ ▼ Measurement	Array[1..8] of UDInt		
■ Measurement[1]	UDInt	0	6045
■ Measurement[2]	UDInt	0	37
■ Measurement[3]	UDInt	0	43
■ Measurement[4]	UDInt	0	15
■ Measurement[5]	UDInt	0	31
■ Measurement[6]	UDInt	0	41
■ Measurement[7]	UDInt	0	37
■ Measurement[8]	UDInt	0	35
■ ► Measurement Scale	Array[1..8] of *Banner_...		

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

Appendix A**R95C 8UI Process Data Out**

The SD50 has 32 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 256-bits of Process Data encode many separate pieces of information, as shown below.

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

First is Run mode (mode 0).

ProcessDataOut "Process Data Out Run Mode" id=V_Pd_OutRunMode									
bit length: 256 data type: 256-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 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Half/Half Steady, 5 = Half/Half Flash, 6 = Intensity Sweep, 7 = Two Color Sweep					Animation	The Animation type
2	4	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	8	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	11	2-bit UInteger	0 = Slow, 1 = Standard, 2 = Fast, 3 = Custom					Speed	The speed of the Animation
5	13	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					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					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 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	24	29-octet String UTF-8						Display Text	Run Mode Display Text

Here is the information for Measure mode (mode 1).

ProcessDataOut "Process Data Out Message Mode" id=V_Pd_OutMessageMode									
bit length: 256 data type: 256-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	224	4-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Half/Half Steady, 5 = Half/Half Flash, 6 = Intensity Sweep, 7 = Two Color Sweep					Animation	The Animation type
2	228	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	232	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	235	2-bit UInteger	0 = Slow, 1 = Standard, 2 = Fast, 3 = Custom					Speed	The speed of the Animation
5	237	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	240	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	244	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	248	4-bit UInteger	1..13 = Message Selection (1-13)					Message Selection 1	Message Mode Message Selection 1 (1-13)
9	252	4-bit UInteger	1..13 = Message Selection (1-13)					Message Selection 2	Message Mode Message Selection 2 (1-13)

Here is Measure mode (mode 2).

ProcessDataOut "Process Data Out Measure Mode" id=V_Pd_OutMeasureMode									
bit length: 256 data type: 256-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	240	16-bit UInteger						Measure Mode Value	Value describing the level of the device, range determined in Measure Mode Parameter Data

Here is Timer mode (mode 3).

ProcessDataOut "Process Data Out Timer Mode" id=V_Pd_OutTimerMode									
bit length: 256 data type: 256-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	254	Boolean						Run Timer	Run the timer in timer mode
2	255	Boolean						Reset Timer	Reset the timer in timer mode

Here is Counter mode (mode 4).

ProcessDataOut "Process Data Out Counter Mode" id=V_Pd_OutCounterMode									
bit length: 256 data type: 256-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	253	Boolean						Increment Count	Increment the counter value when in counter mode
2	254	Boolean						Decrement Count	Decrement the counter value when in counter mode
3	255	Boolean						Reset Count	Reset the count in counter mode