

# TL70 Process Data Function

---

December 18<sup>th</sup>, 2025

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

## **Components**

Banner TL70 Library v16.zal16

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 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 TL70 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.

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

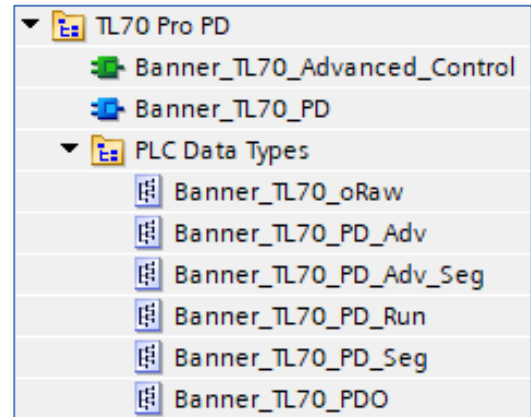
3. Open the IO-Link Generic Devices and select the proper module. The 32/32 byte is required for Pro. Make note of the Q address for the Slot 2 which represents Port 1. Slot 2 starts are 1 for outputs. The other number needed is Q3. The data for the port start at that point (I3). The previous two bytes 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 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 PD Folder.

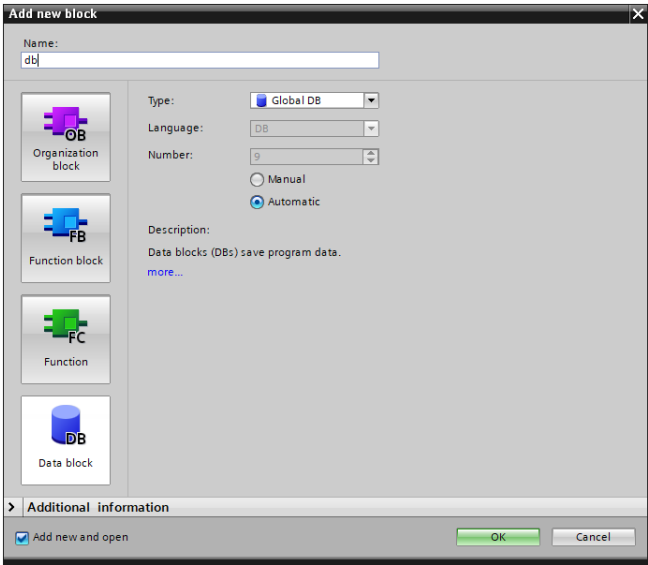
- a. Move Banner\_TL70\_Advanced\_Control and Banner\_TL70\_PD to the Program Blocks area.
- b. In the PLC Data Types folder move the Banner\_TL70\_oRaw, Banner\_TL70\_PD\_Adv, Banner\_TL70\_PD\_Adv\_Seg, Banner\_TL70\_PD\_Run, Banner\_TL70\_PD\_Seg, and Banner\_TL70\_PDO to the PLC Data Types folder in TIA Portal.



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 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 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 IOLM1 01 PDO	"Banner_32Out"	%Q1.0
▶ TL70 IOLM1 01 oRaw	"Banner_TL70_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.

TL70 IOLM1 01 PDIO	"Banner_TL70_PDO"
0-SegMode	"Banner_TL70_PD_Seg"
1-RunMode	"Banner_TL70_PD_Run"
2-Advanced	"Banner_TL70_PD_Adv"
3-Level	UInt
Audible	USInt

9. Add the “Banner\_TL70\_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 IOLM1 01 oRaw” in this example. The “TL70PDO” needs to be linked to the variable created in step 7. It was called “TL70 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, there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL70 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 to the Operating Mode parameter from the Banner IOL Parameter Function (see Fig. 2). See Appendix A for more information about TL70 Process Data Out. The default is “2”.

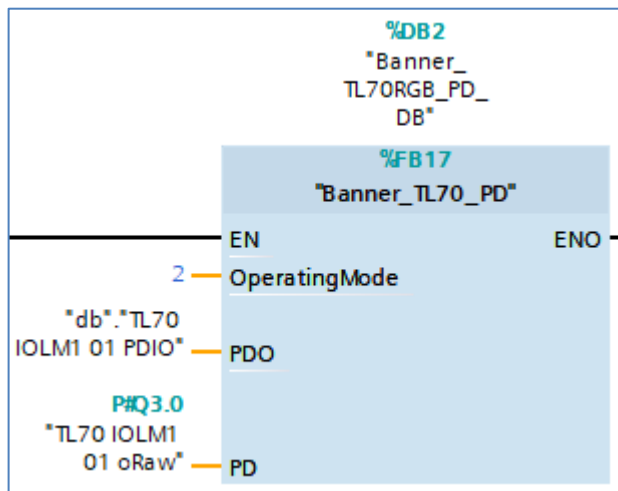


Figure 1: Hand type correct number for Operational Mode

**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 “4”.

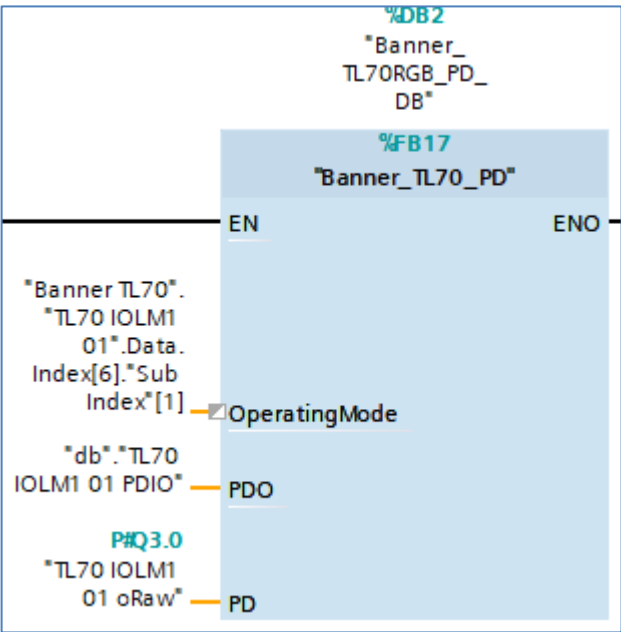
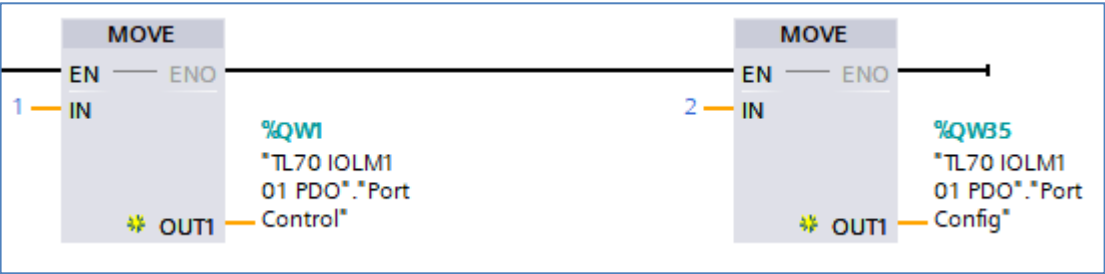


Figure 2: Linking Operational Mode variable to TL70 Pro Parameter Data Function Block

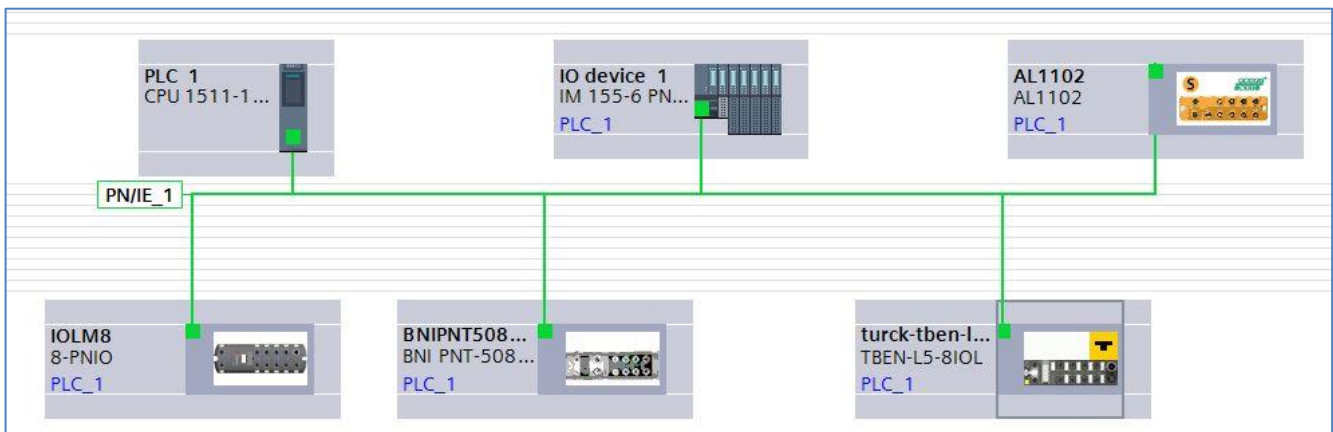
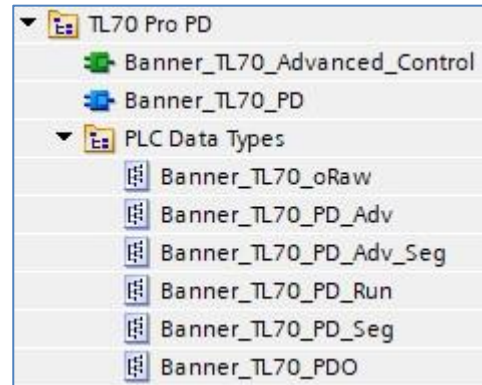
10. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 “TL70 IOLM1 01 PDO”.



- 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 TL70 can be controlled now.

### Setup of TL70 with other IO-Link Masters

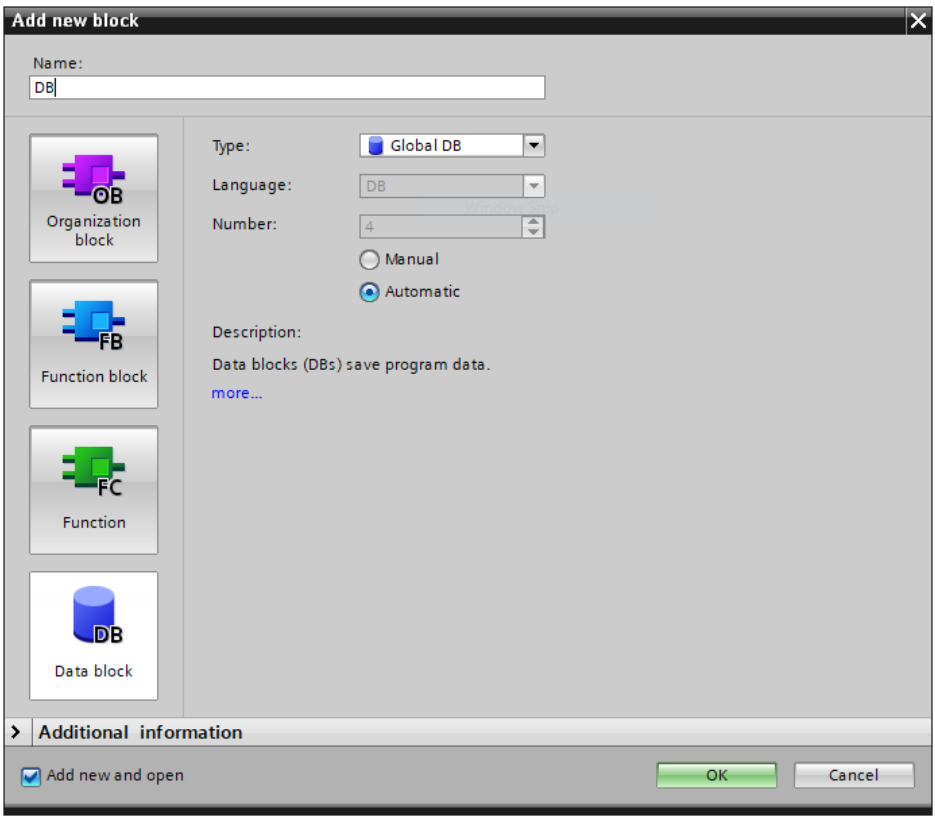
1. The Banner TL70 library will now be in the Global Library List. Expand the Master copies section.
2. Drag Banner\_TL70\_Advanced\_Control and Banner\_TL70\_PD to the Program Blocks area under your PLC.
3. Drag Banner\_TL70\_oRaw, Banner\_TL70\_PD\_Adv, Banner\_TL70\_PD\_Adv\_Seg, Banner\_TL70\_PD\_Run, Banner\_TL70\_PD\_Seg, and Banner\_TL70\_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 17 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, 17 bytes of Process Data Out for port 1 on the IO-Link Master will be stored starting at Q1.
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\_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 6 is tied to this new tag.

▶ TL70 IOLM1 01 oRaw	"Banner_TL70_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. The tag name again calls out the type of light, the IO-Link Master, and the port number. Use the data type “Banner\_TL70\_PDO” for the new tag.

TL70 IOLM1 01 PDIO	"Banner_TL70_PDO"
0-SegMode	"Banner_TL70_PD_Seg"
1-RunMode	"Banner_TL70_PD_Run"
2-Advanced	"Banner_TL70_PD_Adv"
3-Level	UInt
Audible	USInt



10. Add the “Banner\_TL70\_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 9.

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.

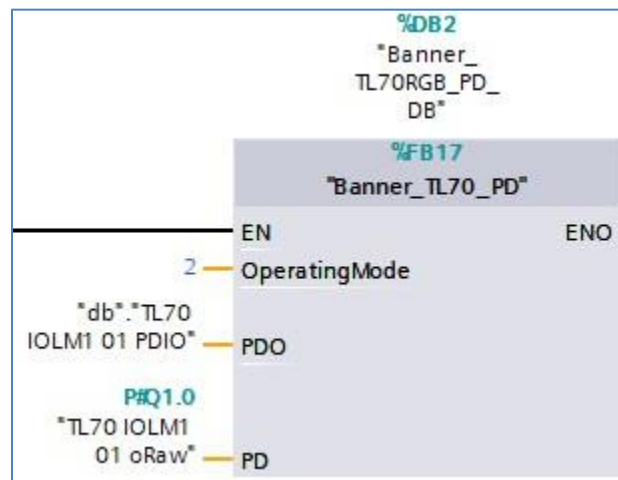


Figure 3: Hand type correct number for Operational Mode

**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.

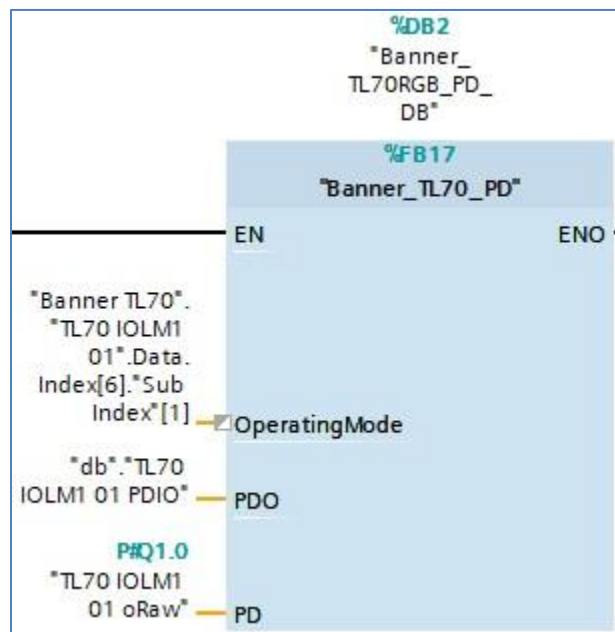


Figure 4: Linking Operational Mode variable to Pro Parameter Data Function Block

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 Process Data.
13. Tower Light can now be controlled.

**Appendix A****TL70 Pro Process Data**

The TL70 has 17 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=PO\_PDOut\_Segment**

bit length: 136

data type: 136-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	88	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	96	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	104	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	112	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	120	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	128	8-bit UInteger	0 = Off, 1 = On					Audible State	State of Audible segment

The next mode, “1”, is “Run”.

ProcessDataOut "Process Data Out Run Mode" id=PO_PDOut_Run									
bit length: 136 data type: 136-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	72	8-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Scroll, 5 = Bounce					Animation	The Animation type
2	80	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 = White					Color 1	Color value
3	88	8-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 1 Intensity	Color Intensity
4	96	8-bit UInteger	0 = Medium, 1 = Slow, 2 = Fast, 3 = Custom					Speed	Animation Speed
5	104	8-bit UInteger	0 = Flash, 1 = Strobe, 2 = 3 Pulse, 3 = SOS, 4 = Random					Pattern	Flash Pattern
6	112	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 = White					Color 2	Color value
7	120	8-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 2 Intensity	Color Intensity
8	128	8-bit UInteger	0 = Up, 1 = Down					Animation Direction	Scroll/Bounce Animation Direction
9	64	8-bit UInteger	0 = Off, 1 = On					Audible State	State of Audible segment

Mode 2 is “Advanced”.

ProcessDataOut "Process Data Out Advanced Mode" id=PO_PDOut_Advanced									
bit length: 136 data type: 136-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	128	3-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Half/Half, 5 = Half/Half Rotate, 6 = Chase, 7 = Intensity Sweep					Segment 1 Animation	The Animation type
2	131	2-bit UInteger	0 = Medium, 1 = Slow, 2 = Fast, 3 = Custom					Segment 1 Speed	Animation Speed
3	133	3-bit UInteger	0 = Flash, 1 = Strobe, 2 = 3 Pulse, 3 = SOS, 4 = Random					Segment 1 Pattern	Flash Pattern
4	120	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 = White					Segment 1 Color 1	Color value
5	125	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Segment 1 Color 1 Intensity	Color Intensity
6	112	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 = White					Segment 1 Color 2	Color value
7	117	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Segment 1 Color 2 Intensity	Color Intensity

Mode 3 is “Level”.

ProcessDataOut "Process Data Out Level Mode" id=PO_PDOut_Level									
bit length: 136									
data type: 136-bit Record									
subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	112	16-bit UInteger						Process Level	Current level value
2	128	8-bit UInteger	0 = Off, 1 = On					Audible State	State of Audible segment