K50 Pro Process Data Function

11/17/2022

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 K50 Audible or Banner K50 Pro Touch light via an IO-Link Master from a Siemens PLC. The function covers parsing and display of the K50 Indicator sensor Process Data Out.

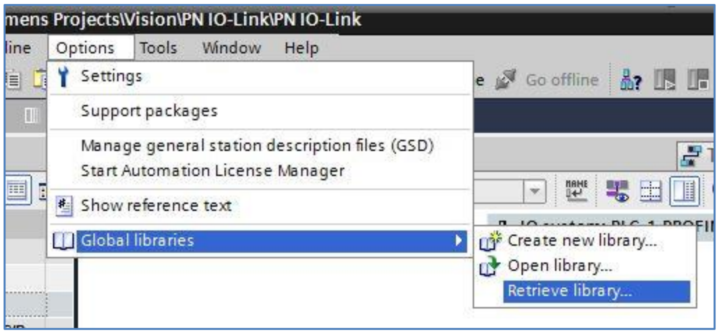
Components

Banner K50 Pro Touch.zal14

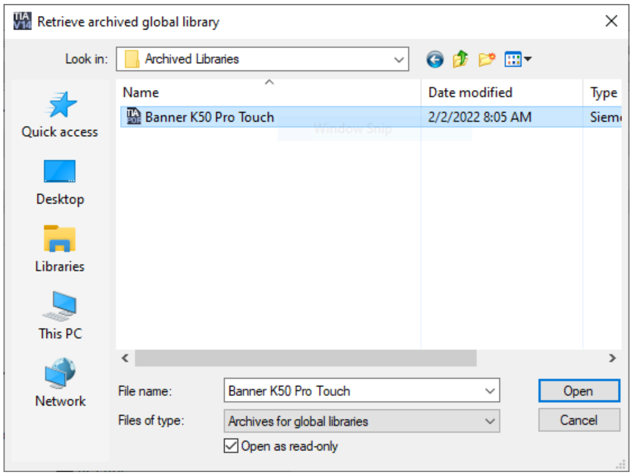
There are two methods for the process data. The first is used when creating a connection to Banner’s IO-Link masters. The second set of instructions are for systems using other manufacturer’s IO-Link masters.

Installation Instructions

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



3. Select the Banner K50 Pro. Click Open.



Look in: Archived Libraries

Name	Date modified	Type
Banner K50 Pro Touch	2/2/2022 8:05 AM	Siem

File name: Banner K50 Pro Touch

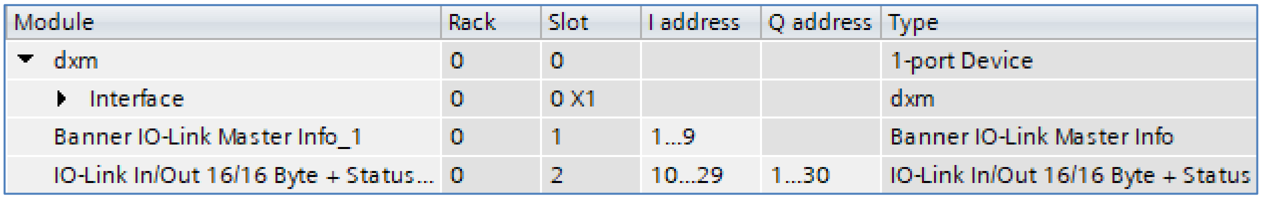
Files of type: Archives for global libraries

☑ Open as read-only

4. The library is now accessible in the Libraries tab.
5. Go to page 3 for Banner IO-Link Masters, to page 8 for K50 Indicator for all other IO-Link Masters, and page13 for K50 Pro Audible for all other IO-Link Masters.

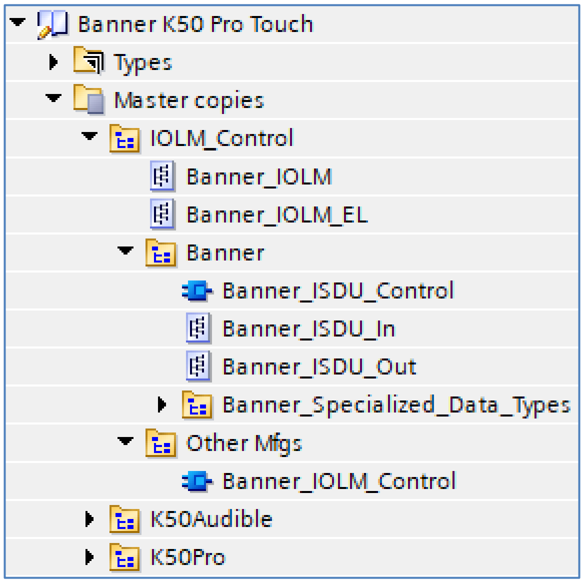
Setup of K50 Pro with a Banner DXMR90-4K

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.
3. Open the IO-Link Generic Devices and select the proper module. The 16/16 byte is required for K50 Pro Touch. Make note of the I address for the Slot 2 which represents Port 1. Slot 2 starts are 1 for outputs. The other number needed is I3. The data for the port start at that point (I3). The previous two bytes Port Control.



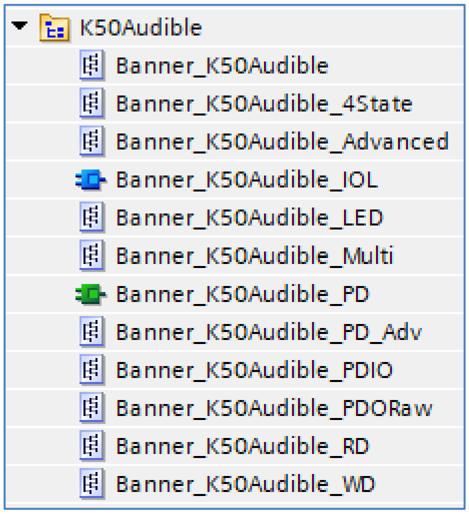
Module	Rack	Slot	I address	Q address	Type
▾ dxm	0	0			1-port Device
▸ Interface	0	0 X1			dxm
Banner IO-Link Master Info_1	0	1	1...9		Banner IO-Link Master Info
IO-Link In/Out 16/16 Byte + Status...	0	2	10...29	1...30	IO-Link In/Out 16/16 Byte + Status

4. Drag the necessary tag from IOLM_Control > Banner > Banner_Specialized_Data_Types. The tag used in this example is "Banner_16out" and "Banner_16in". This tag represents the full raw process data along with port status information.
5. Go to step 5 if using a K50 Pro Audible and step 6 if a K50 Pro Touch is being used.

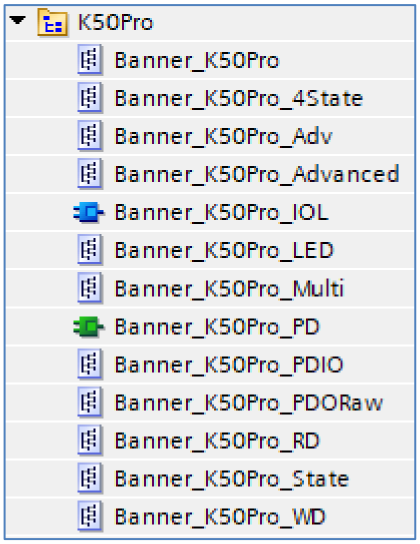


- Banner K50 Pro Touch
  - Types
  - Master copies
    - IOLM_Control
      - Banner_IOLM
      - Banner_IOLM_EL
      - Banner
        - Banner_ISDU_Control
        - Banner_ISDU_In
        - Banner_ISDU_Out
        - Banner_Specialized_Data_Types
      - Other Mfgs
        - Banner_IOLM_Control
    - K50Audible
    - K50Pro

6. Drag the necessary files from the K50Audible Folder.
   a. Move Banner_K50Audible, Banner_K50Audible_4State, Banner_K50Audible_Advanced, Banner_K50Audible_LED, Banner_K50Audible_Multi, Banner_K50Audible_PD_Adv, Banner_K50Audible_PDIO, and Banner_K30Pro_PDORaw to the PLC Data Types area.
   b. Move Banner_K30Pro_PD to the Program Blocks area.



K50Audible
- Banner_K50Audible
- Banner_K50Audible_4State
- Banner_K50Audible_Advanced
- Banner_K50Audible_IOL
- Banner_K50Audible_LED
- Banner_K50Audible_Multi
- Banner_K50Audible_PD
- Banner_K50Audible_PD_Adv
- Banner_K50Audible_PDIO
- Banner_K50Audible_PDORaw
- Banner_K50Audible_RD
- Banner_K50Audible_WD

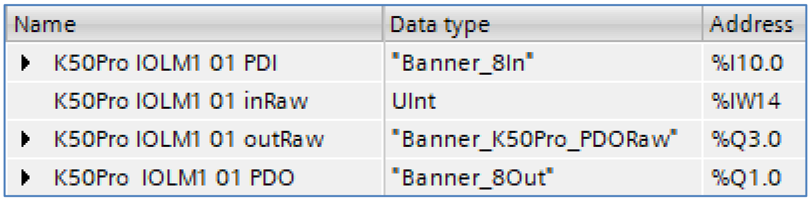


K50Pro
- Banner_K50Pro
- Banner_K50Pro_4State
- Banner_K50Pro_Adv
- Banner_K50Pro_Advanced
- Banner_K50Pro_IOL
- Banner_K50Pro_LED
- Banner_K50Pro_Multi
- Banner_K50Pro_PD
- Banner_K50Pro_PDIO
- Banner_K50Pro_PDORaw
- Banner_K50Pro_RD
- Banner_K50Pro_State
- Banner_K50Pro_WD

7. Drag the necessary files from the K50Pro Folder.
   a. Move Banner_K30Pro, Banner_K30Pro_4State, Banner_K30Pro_Adv, Banner_K30Pro_Advanced, Banner_K30Pro_LED, Banner_K30Pro_Multi, Banner_K30Pro_PDIO, and Banner_K30Pro_PDOraw to the PLC Data Types area.
   b. Move Banner_K30Pro_PD to the Program Blocks area.

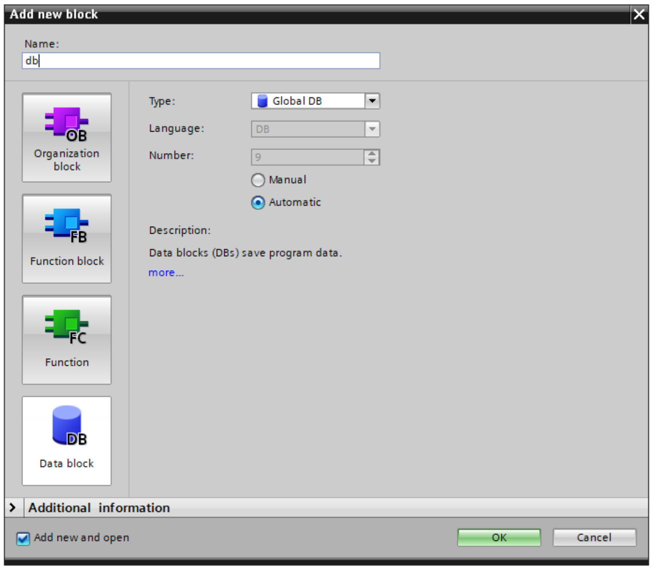
8. The rest of the steps will focus on K50 Pro Touch model. The K50 Pro Audible follows similar steps. The only difference for Process data is that Audible options are added for the K50 Pro Audible models.

9. Go to PLC Tags. Create four tags. One set of tags is for the full data structure while the second set creates tags to represent the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, then the tag “K50Pro IOLM1 01 PDO” was created using a Data Type of “Banner_16out”. 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 tag that represents the raw data is “K50Pro IOLM1 01 outRaw” and uses the “Q” address found in step 2 (%Q3). Tags “K50Pro IOLM1 02 PDI” (%I10) and “K50Pro IOLM1 01 inRaw” (%IW14) are created for the inputs also. This is the tag that will be used in the Function block.



Name	Data type	Address
K50Pro IOLM1 01 PDI	"Banner_8In"	%I10.0
K50Pro IOLM1 01 inRaw	UInt	%IW14
K50Pro IOLM1 01 outRaw	"Banner_K50Pro_PDORaw"	%Q3.0
K50Pro IOLM1 01 PDO	"Banner_8Out"	%Q1.0

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



Add new block

Name: db

Type: Global DB

Language: DB

Number: 9

○ Manual

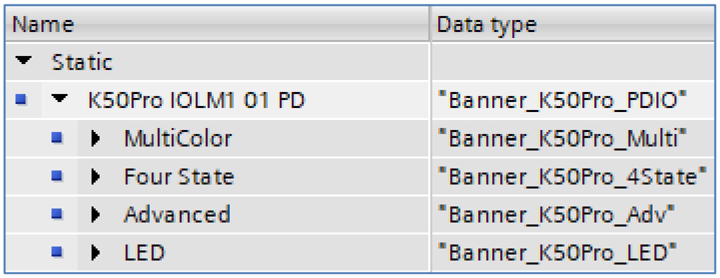
● Automatic

Description:

Data blocks (DBs) save program data.

more...

11. In the new data block, create a new tag to represent the parsed Process Data Output for our K30 Pro Touch. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_K50PRO_PDIO” for the new tag.



Name	Data type
▼ Static	
■ ▼ K50Pro IOLM1 01 PD	"Banner_K50Pro_PDIO"
■ ▶ MultiColor	"Banner_K50Pro_Multi"
■ ▶ Four State	"Banner_K50Pro_4State"
■ ▶ Advanced	"Banner_K50Pro_Adv"
■ ▶ LED	"Banner_K50Pro_LED"

8. Add the “Banner_K50Pro_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 called “K50Pro IOLM1 01 outRaw” in this example. Link the ”PDI” to the raw process data variable from step 5. Use the variable called “K50Pro IOLM1 01 inRaw” from step 5. The “K50 Pro PD” needs to be linked to the variable created in step 7. It was called “K50Pro IOLM1 01 PD” for this example.

   The last variable, “Operational Mode”, allow the function to correctly interpret the Process Data Out. In the case of the K30 Pro Touch , there are four user-selected modes for the Process Data Out. This function needs to know what choice has been made in the TL50 Pro Select 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 K30 Pro Touch Process Data Function to the K30 Pro Touch Parameter Data Function Block (see Fig. 2). See Appendix A for more information about K30 Pro Touch Process Data Out.

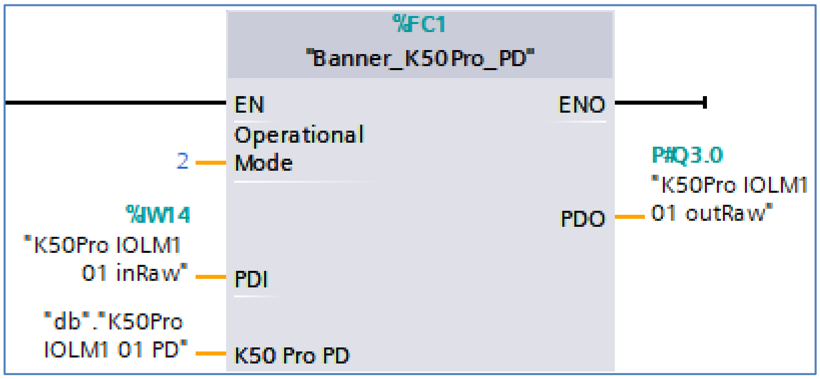


Figure 1: Hand type correct number for Operational Mode

> ***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” (MultiColor Mode), “1” (Four State Mode), “2” (Advanced Mode), and “3” (LED Mode); where the entire tower light behaves as a level indicator). The default is “2”.

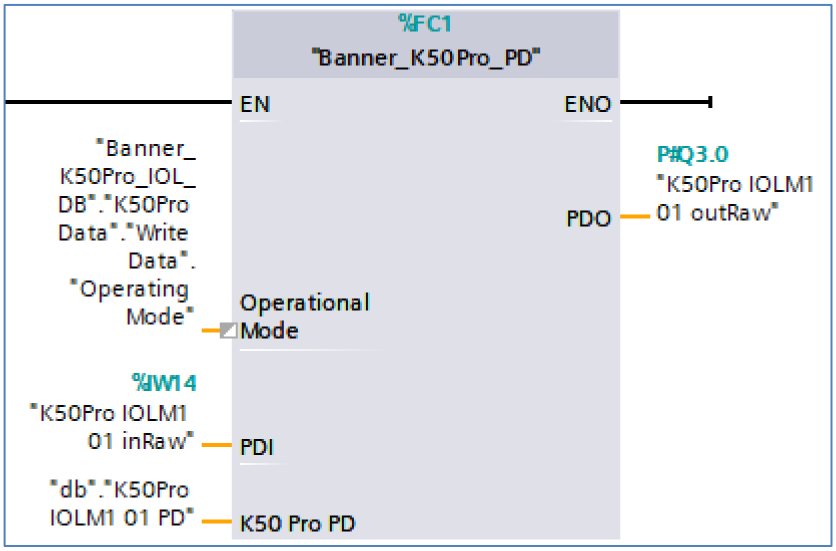
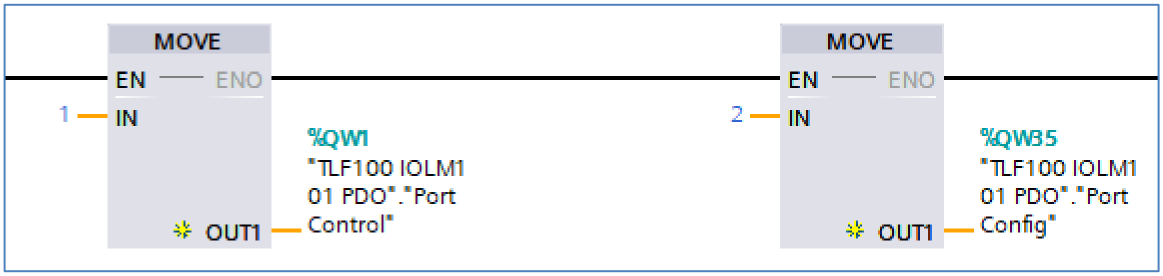


Figure 2: Linking Operational Mode variable to K30 Pro Touch Parameter Data Function Block

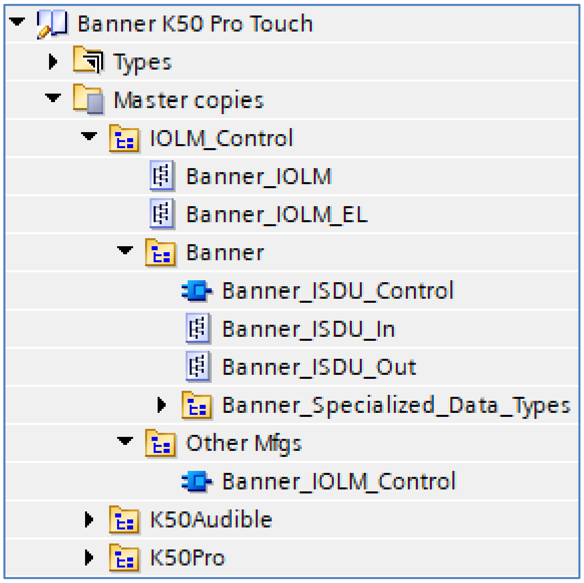
9. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 “TLF100 IOLM1 01 PDO”.


MOVE
EN
ENO
1
IN
%QW1
"TLF100 IOLM1 01 PDO"."Port Control"
OUT1
MOVE
EN
ENO
2
IN
%QW35
"TLF100 IOLM1 01 PDO"."Port Config"
OUT1

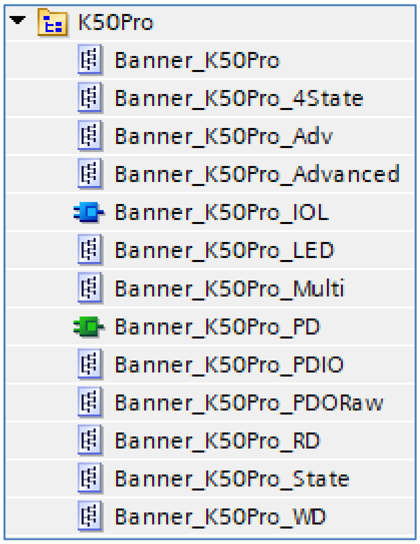
10. Process Data Setup is complete.
11. Compile and download the configuration to the PLC, then go online. Open the “db” data block and click Monitor all. The K50 Pro Touch can be controlled now.

Setup of K50 Pro Touch with other IO-Link Masters

1. The Banner K50 Pro Touch library will now be in the Global Library List. Expand the Master copies section. The K50 Pro Touch folder contains elements for both Process Data and Parameter Data connections to a K50 Pro Touch device. As Process Data is the focus of this paper, we will concern ourselves with these seven items: Banner_K50_Advanced, Banner_K50Pro_4State, Banner_K50Pro_Adv, Banner_K50Pro_LED, Banner_K50Pro_Multi, Banner_K50Pro_PD, Banner_K50Pro_PDIO, and Banner_K50Pro_PDORaw.
2. Drag Banner_K50Pro_PD to the Program Blocks area under your PLC.
3. Drag Banner_K50_Advanced, Banner_K50Pro_4State, Banner_K50Pro_Adv, Banner_K50Pro_LED, Banner_K50Pro_Multi, Banner_K50Pro_PDIO, and Banner_K50Pro_PDORaw to the PLC Data Types area under your PLC.

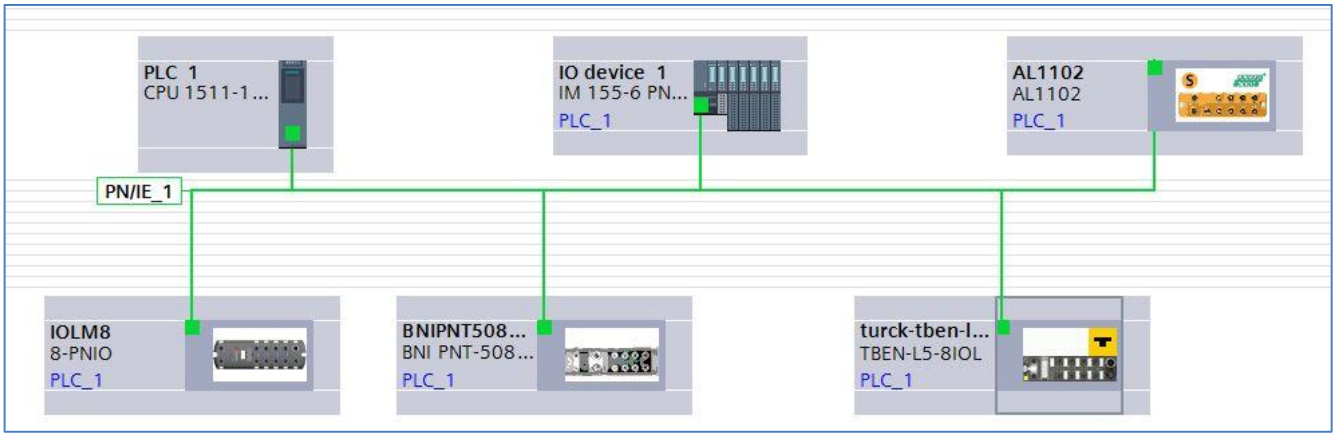


- Banner K50 Pro Touch
  - Types
  - Master copies
    - IOLM_Control
      - Banner_IOLM
      - Banner_IOLM_EL
      - Banner
        - Banner_ISDU_Control
        - Banner_ISDU_In
        - Banner_ISDU_Out
        - Banner_Specialized_Data_Types
      - Other Mfgs
        - Banner_IOLM_Control
    - K50Audible
    - K50Pro

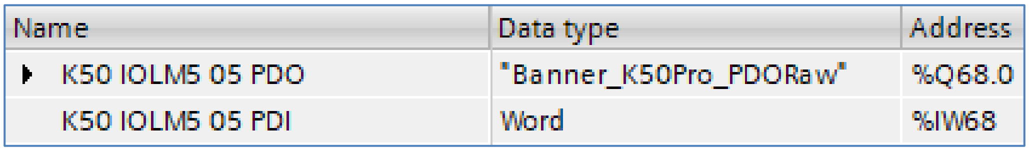


- K50Pro
  - Banner_K50Pro
  - Banner_K50Pro_4State
  - Banner_K50Pro_Adv
  - Banner_K50Pro_Advanced
  - Banner_K50Pro_IOL
  - Banner_K50Pro_LED
  - Banner_K50Pro_Multi
  - Banner_K50Pro_PD
  - Banner_K50Pro_PDIO
  - Banner_K50Pro_PDORaw
  - Banner_K50Pro_RD
  - Banner_K50Pro_State
  - Banner_K50Pro_WD

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.

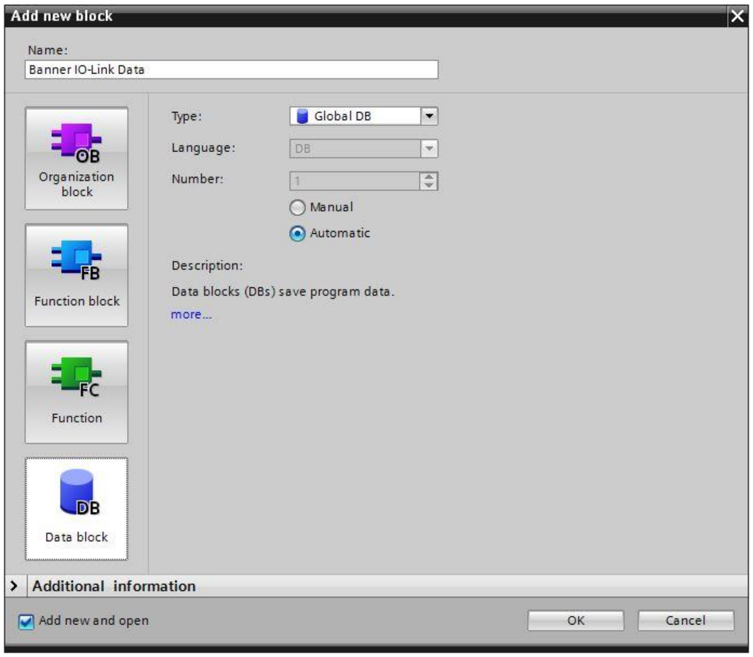


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a K50 Pro Touch requires 9 bytes of space for the Process Data Out and 1 byte for the Process Data In. This will likely require a 16 byte IN/OUT type.
6. Record the "I" and "Q" addresses where this K50 Pro Touch Process Data is to be stored, as these addresses will be required in the next step. In this example, 9 bytes of Process Data Out for port 2 on the IO-Link Master will be stored in Q68 through Q77 and the 1 word of Process Data In will be stored in I68 through I69.
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 "K50 Pro Touch IOLM5 05 PDO" was created using a Data Type of "Banner_K50Pro_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 IOLM1 or IOLM2, 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. Similarly, we need to make a tag for the Process Data In. Here the name "K50 Pro Touch IOLM5 05 PDI" was chosen, with a Data Type of Word. The "I" address from step 9 is used.



Name	Data type	Address
▸ K50 IOLM5 05 PDO	"Banner_K50Pro_PDORaw"	%Q68.0
K50 IOLM5 05 PDI	Word	%IW68

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".



Name: Banner IO-Link Data

Type: Global DB

Language: DB

Number: 1

○ Manual

● Automatic

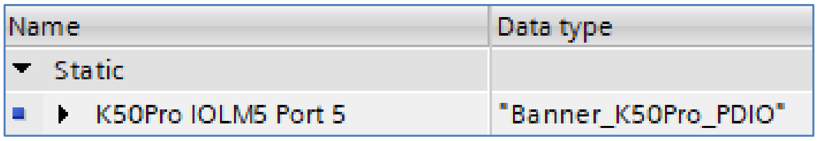
Description:

Data blocks (DBs) save program data.

more...

☑ Add new and open

9. In the new data block, create a new tag to represent the parsed Process Data In for our K50 Pro Touch. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type "Banner_K50Pro_Touch_PDIO" for the new tag.



Name	Data type
▼ Static	
■ ▶ K50Pro IOLM5 Port 5	"Banner_K50Pro_PDIO"

10. Add the "Banner_K50Pro_PD" function to an OB ladder. Link the "Raw PDI" to the raw Process Data In variable from step 10. Link the "Raw PDO" to the raw Process Data Out variable from step 10. Link "K50 Pro PD" 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 K50 Pro Touch, there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the K50 Pro Touch 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 K50 Pro Touch Process Data Function to the K50 Pro Touch Parameter Data Function Block (see Fig. 2). See Appendix A for more information about K50 Pro Touch Process Data Out.

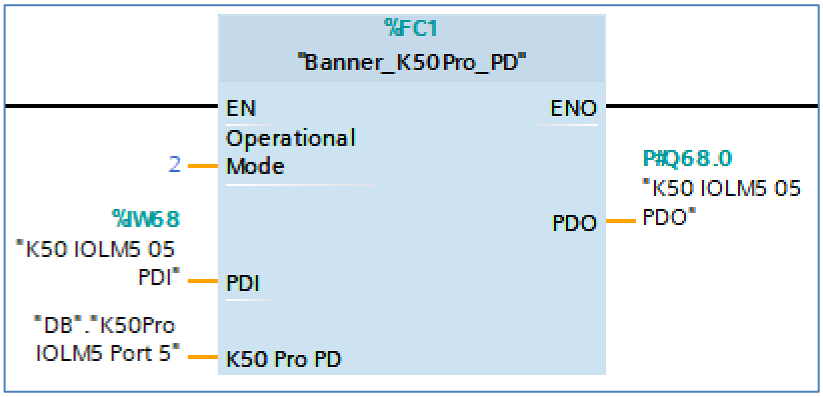


Figure 3: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the K50's current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

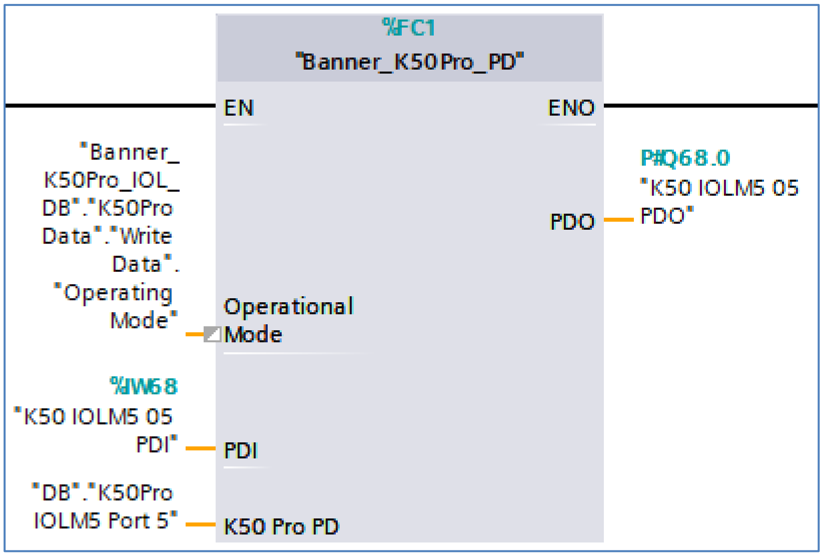
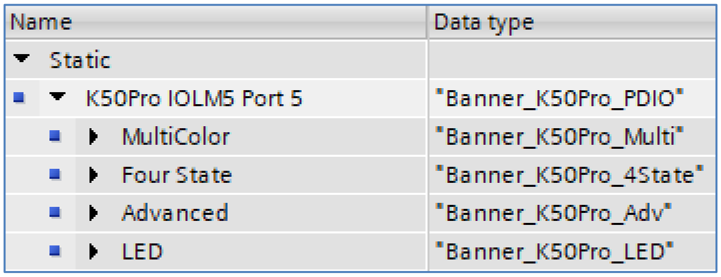


Figure 4: Linking Operational Mode variable to K50 Pro Touch Parameter Data Function Block

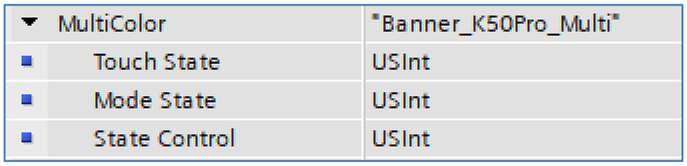
11. Process Data setup is complete.
12. Compile and download the configuration to the PLC, then go online. Open the “Banner IO-Link Data” data block and click Monitor all. You should see parsed K50 Pro Touch Process Data In.
13. Process Data is broken up into four different types. When Operational Modes controls which of four types is used to parse the raw byte data.



Name	Data type
Static	
K50Pro IOLM5 Port 5	"Banner_K50Pro_PDIO"
MultiColor	"Banner_K50Pro_Multi"
Four State	"Banner_K50Pro_4State"
Advanced	"Banner_K50Pro_Adv"
LED	"Banner_K50Pro_LED"

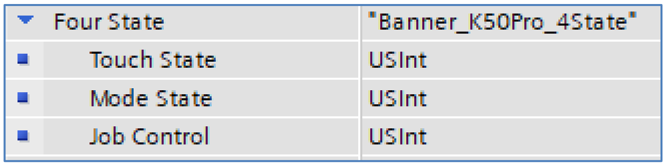
0: MultiColor
1: Four State
2: Advanced
3: LED

a. MultiColor has three pieces of data. State Control is an output that controls the light. Touch State and Mode State are inputs giving the state of the light.



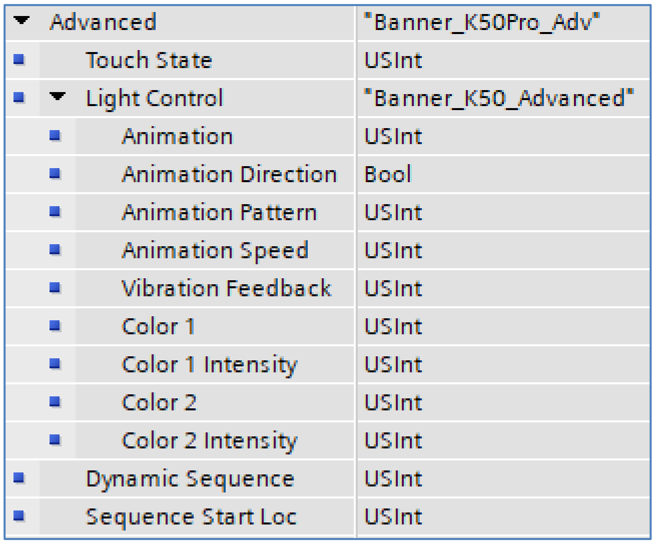
MultiColor	"Banner_K50Pro_Multi"
Touch State	USInt
Mode State	USInt
State Control	USInt

b. Four State has three pieces of data. Job Control tells the light if a pick is being requested. Depending on the Touch State the light will be automatically set to one of four colors. Touch State and Mode State are inputs giving the state of the light.



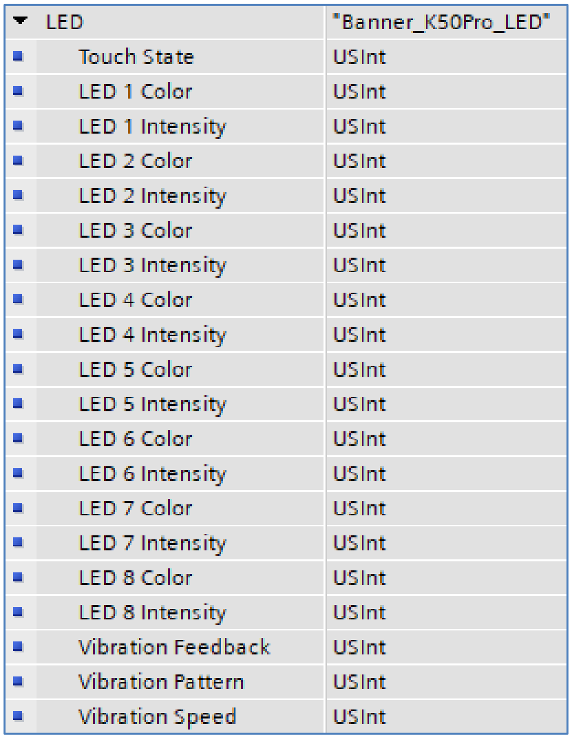
Four State	"Banner_K50Pro_4State"
Touch State	USInt
Mode State	USInt
Job Control	USInt

c. Advanced allows for the complete control of the light. In MultiColor and Four State the light has a few preconfigured modes that can be activated depending on the State Control and Job Control values. Advanced mode the process data fully control the light. Nothing is preconfigured. Turning on the light requires a non-zero value to be entered into animation. Depending on the value entered multiple modes can be activated. Color 1 and Color 2 control the color the light will emit. Color 2 is only used in a few modes activated by Animation.



Advanced	"Banner_K50Pro_Adv"
Touch State	USInt
Light Control	"Banner_K50_Advanced"
Animation	USInt
Animation Direction	Bool
Animation Pattern	USInt
Animation Speed	USInt
Vibration Feedback	USInt
Color 1	USInt
Color 1 Intensity	USInt
Color 2	USInt
Color 2 Intensity	USInt
Dynamic Sequence	USInt
Sequence Start Loc	USInt

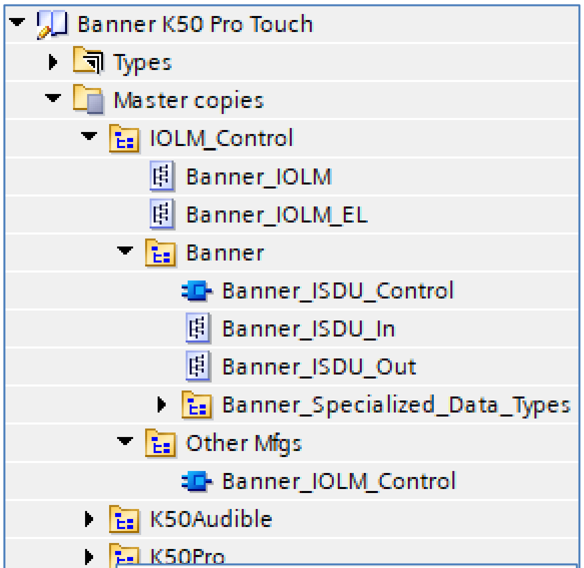
d. LED controls each LED individually. There process data breaks out each of the eight LEDs individually. When a non-zero value is entered the LED will light up with the color corresponding to that value.

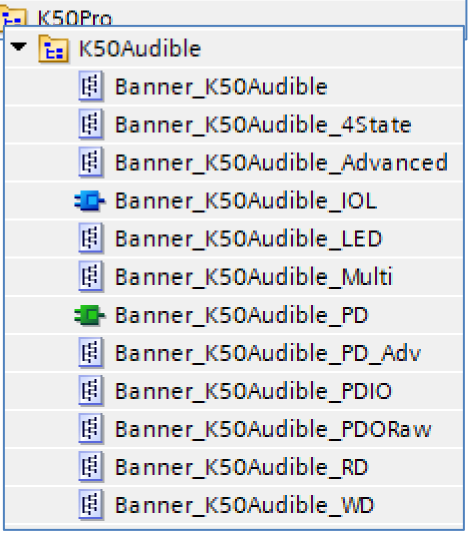


LED	"Banner_K50Pro_LED"
Touch State	USInt
LED 1 Color	USInt
LED 1 Intensity	USInt
LED 2 Color	USInt
LED 2 Intensity	USInt
LED 3 Color	USInt
LED 3 Intensity	USInt
LED 4 Color	USInt
LED 4 Intensity	USInt
LED 5 Color	USInt
LED 5 Intensity	USInt
LED 6 Color	USInt
LED 6 Intensity	USInt
LED 7 Color	USInt
LED 7 Intensity	USInt
LED 8 Color	USInt
LED 8 Intensity	USInt
Vibration Feedback	USInt
Vibration Pattern	USInt
Vibration Speed	USInt

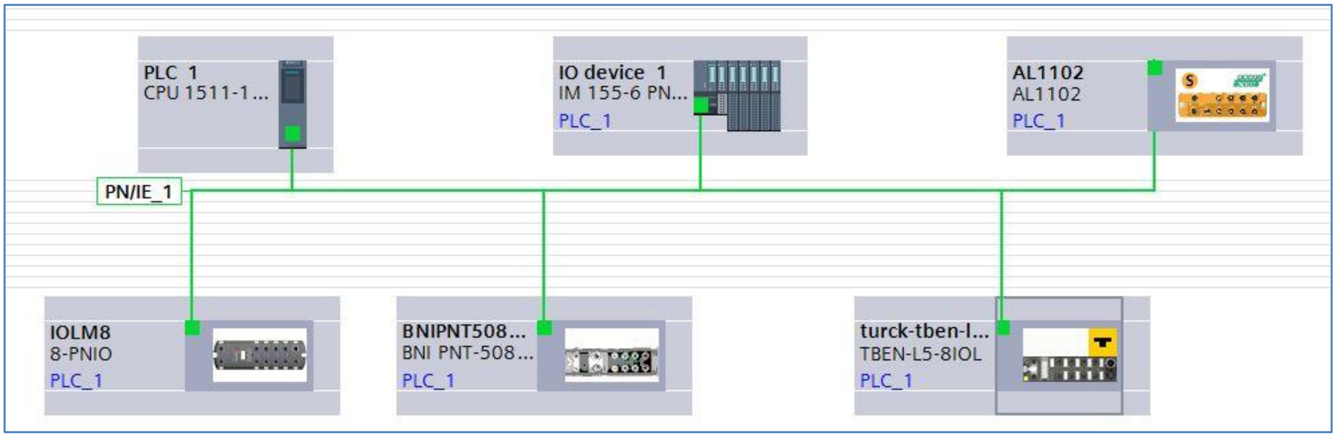
Setup of K50 Pro Audible with other IO-Link Masters

1. The Banner K50 Pro Audible 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_K50Audible_4State, Banner_K50Audible_Advanced, Banner_K50Audible_LED, Banner_K50Audible_Multi, Banner_K50Audible_PD, Banner_K50Audible_PD_Adv, Banner_K50Audible_PDIO, and Banner_K50Pro_PDORaw.
2. Drag Banner_K50Audible_PD to the Program Blocks area under your PLC.
3. Drag Banner_K50Audible_4State, Banner_K50Audible_Advanced, Banner_K50Audible_LED, Banner_K50Audible_Multi, Banner_K50Audible_PD_Adv, Banner_K50Audible_PDIO, and Banner_K50Pro_PDORaw to the PLC Data Types area under your PLC.

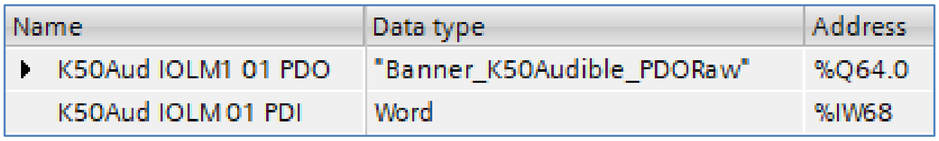




4. Go to Devices and networks to configure the system as necessary. Below is an example of what a configuration might look like. This example shows 5 different IO-Link Masters connected to the same PLC.

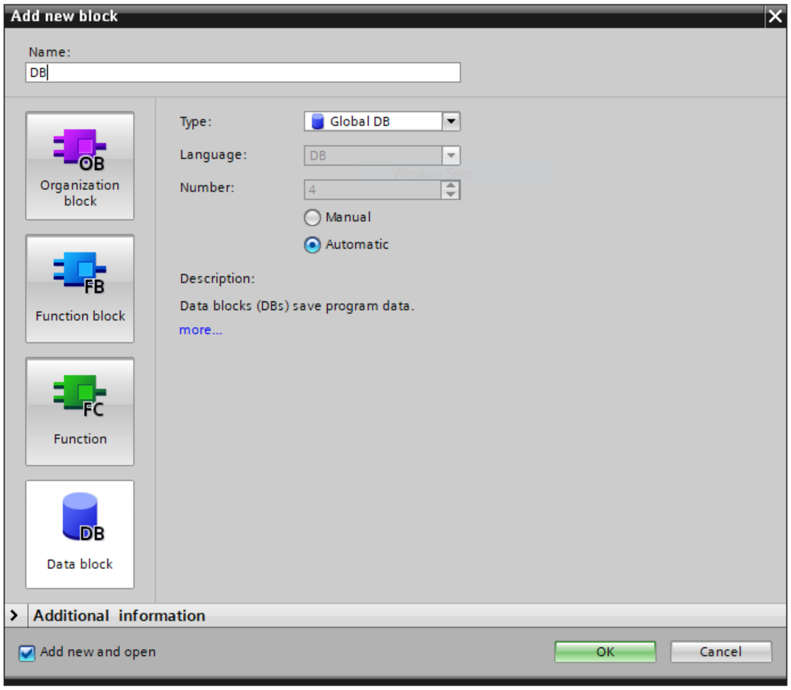


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a K50 Pro Audible requires 9 bytes of space for the Process Data Out and 1 byte for the Process Data In. This will likely require a 16 byte IN/OUT type.
6. Record the "I" and "Q" addresses where this K50 Pro Audible Process Data is to be stored, as these addresses will be required in the next step. In this example, 9 bytes of Process Data Out for port 2 on the IO-Link Master will be stored in Q64 through Q72 and the 1 word of Process Data In will be stored in I68 through I69.
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 "K50Aud IOLM1 01 PDO" was created using a Data Type of "Banner_K50Audible_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. Similarly, we need to make a tag for the Process Data In. Here the name "K50Aud IOLM1 01 PDI" was chosen, with a Data Type of Word. The "I" address from step 9 is used.



Name	Data type	Address
▸ K50Aud IOLM1 01 PDO	"Banner_K50Audible_PDORaw"	%Q64.0
K50Aud IOLM 01 PDI	Word	%IW68

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".



Name: DB

Type: Global DB

Language: DB

Number: 4

○ Manual

● Automatic

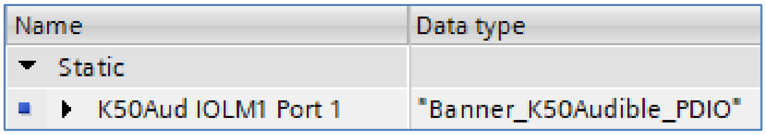
Description:

Data blocks (DBs) save program data.

more...

☑ Add new and open

9. In the new data block, create a new tag to represent the parsed Process Data In for our K50 Pro Audible. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type "Banner_K50Audible_PDIO" for the new tag.



Name	Data type
▼ Static	
▶ K50Aud IOLM1 Port 1	"Banner_K50Audible_PDIO"

10. Add the "Banner_K50Audible_PD" function to an OB ladder. Link the "Raw PDI" to the raw Process Data In variable from step 10. Link the "Raw PDO" to the raw Process Data Out variable from step 10. Link "K50 Pro PD" 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 K50 Pro Audible, there are five user-selected modes for the Process Data Out. This function needs to know what choice has been made in the K50 Pro Audible 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 K50 Pro Audible Process Data Function to the K50 Pro Audible Parameter Data Function Block (see Fig. 2). See Appendix A for more information about K50 Pro Audible Process Data Out.

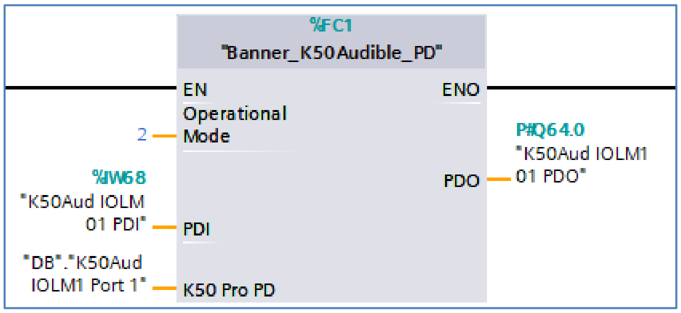


Figure 5: Hand type correct number for Operational Mode

> ***NOTE:*** if you type in the incorrect number (i.e. it does not match the K50's current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

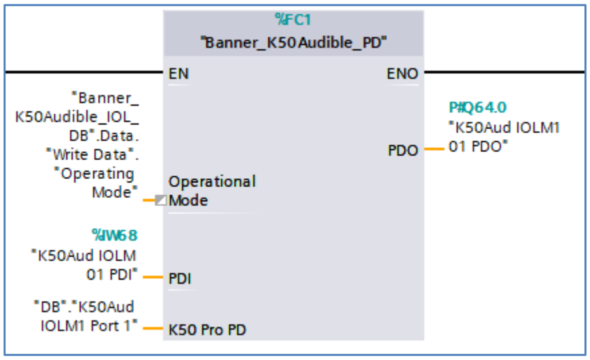
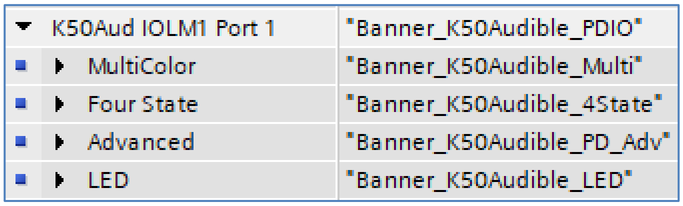


Figure 6: Linking Operational Mode variable to K50 Pro Audible Parameter Data Function Block

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 K50 Pro Audible Process Data In.

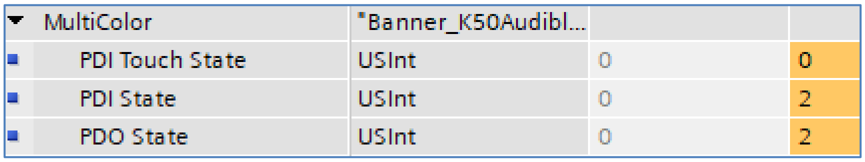
13. Process Data is broken up into four different types. When Operational Modes controls which of four types is used to parse the raw byte data.



K50Aud IOLM1 Port 1	"Banner_K50Audible_PDIO"
MultiColor	"Banner_K50Audible_Multi"
Four State	"Banner_K50Audible_4State"
Advanced	"Banner_K50Audible_PD_Adv"
LED	"Banner_K50Audible_LED"

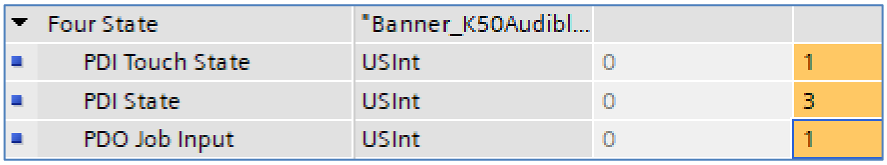
0: MultiColor
1: Four State
2: Advanced
3: LED

a. MultiColor has three pieces of data. State Control is an output that controls the light. Audible State and Mode State are inputs giving the state of the light.



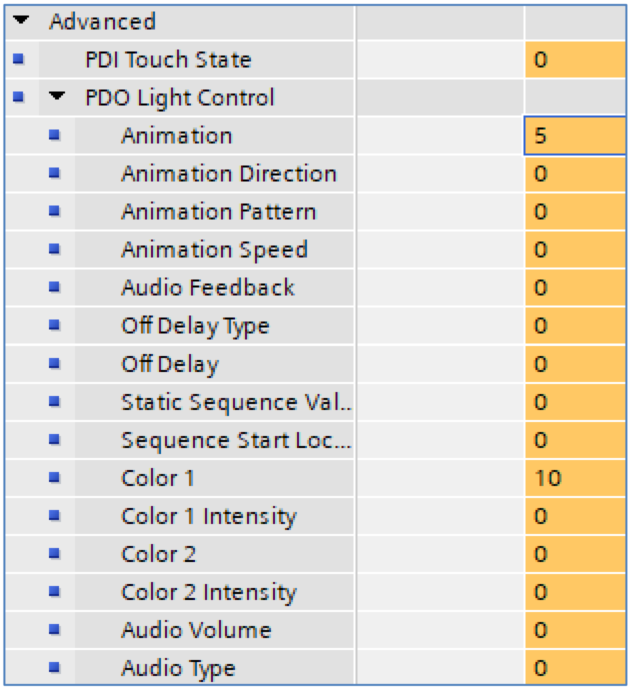
MultiColor	"Banner_K50Audibl...		
PDI Touch State	USInt	0	0
PDI State	USInt	0	2
PDO State	USInt	0	2

b. Four State has three pieces of data. Job Control tells the light if a pick is being requested. Depending on the Audible State the light will be automatically set to one of four colors. Audible State and Mode State are inputs giving the state of the light.



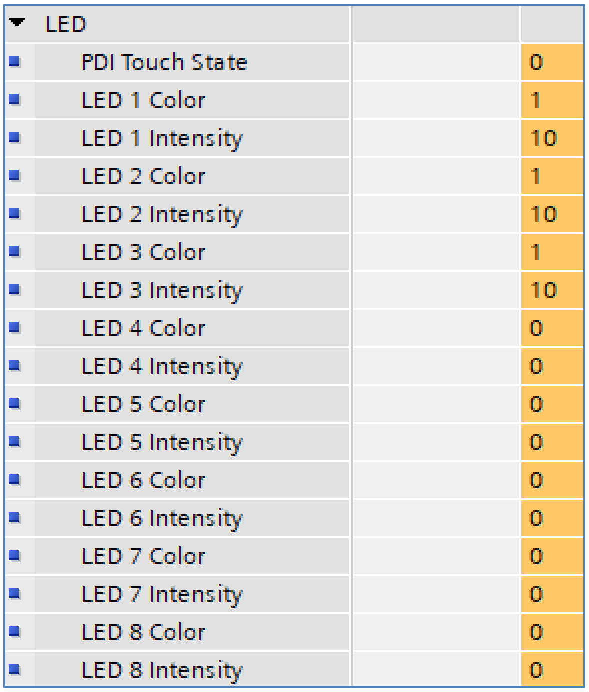
Four State	"Banner_K50Audibl...		
PDI Touch State	USInt	0	1
PDI State	USInt	0	3
PDO Job Input	USInt	0	1

c. Advanced allows for the complete control of the light. In MultiColor and Four State the light has a few preconfigured modes that can be activated depending on the State Control and Job Control values. Advanced mode the process data fully control the light. Nothing is preconfigured. Turning on the light requires a non-zero value to be entered into animation. Depending on the value entered multiple modes can be activated. Color 1 and Color 2 control the color the light will emit. Color 2 is only used in a few modes activated by Animation.



Advanced		
PDI Touch State		0
PDO Light Control		
Animation		5
Animation Direction		0
Animation Pattern		0
Animation Speed		0
Audio Feedback		0
Off Delay Type		0
Off Delay		0
Static Sequence Val...		0
Sequence Start Loc...		0
Color 1		10
Color 1 Intensity		0
Color 2		0
Color 2 Intensity		0
Audio Volume		0
Audio Type		0

d. LED controls each LED individually. There process data breaks out each of the eight LEDs individually. When a non-zero value is entered the LED will light up with the color corresponding to that value.

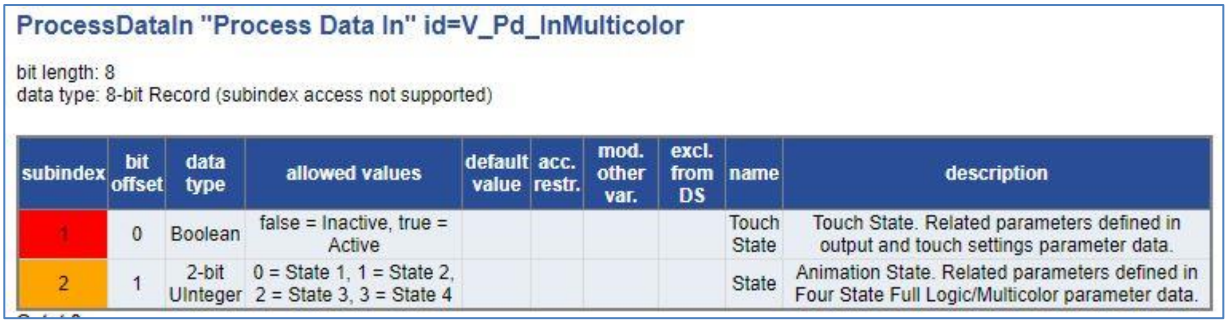


LED		
PDI Touch State		0
LED 1 Color		1
LED 1 Intensity		10
LED 2 Color		1
LED 2 Intensity		10
LED 3 Color		1
LED 3 Intensity		10
LED 4 Color		0
LED 4 Intensity		0
LED 5 Color		0
LED 5 Intensity		0
LED 6 Color		0
LED 6 Intensity		0
LED 7 Color		0
LED 7 Intensity		0
LED 8 Color		0
LED 8 Intensity		0

Appendix A K50 Pro Touch Process Data

The K50 Pro Touch has 2 bytes of Process Data In and 9 bytes of Process Data Out. There are five 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.

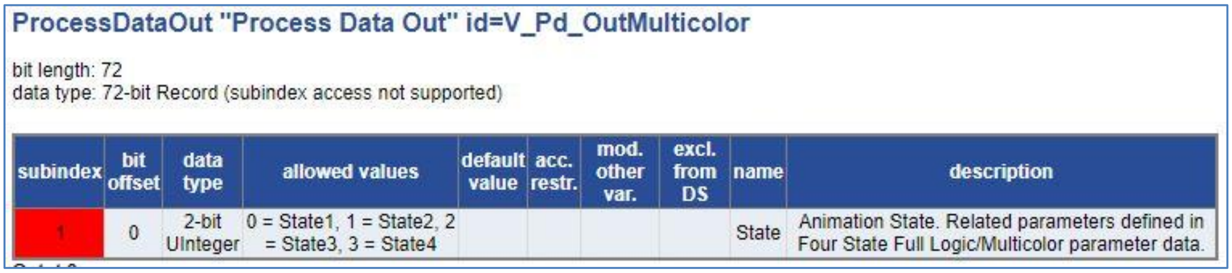
The first is mode 0, “Mulitcolor”.



ProcessDataIn "Process Data In" id=V_Pd_InMulticolor

bit length: 8
data type: 8-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					Touch State	Touch State. Related parameters defined in output and touch settings parameter data.
2	1	2-bit UInteger	0 = State 1, 1 = State 2, 2 = State 3, 3 = State 4					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.

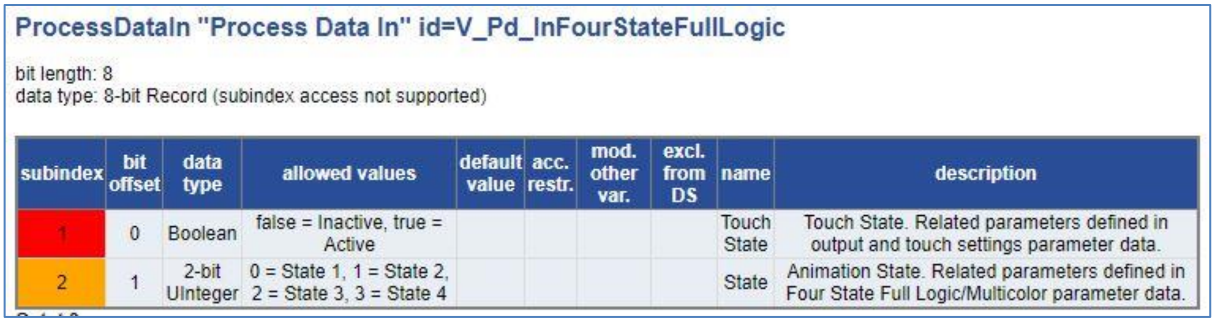


ProcessDataOut "Process Data Out" id=V_Pd_OutMulticolor

bit length: 72
data type: 72-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	2-bit UInteger	0 = State1, 1 = State2, 2 = State3, 3 = State4					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.

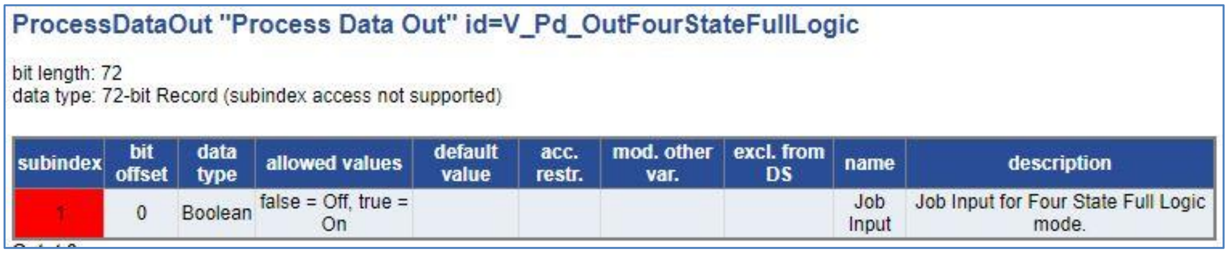
The next mode, “1”, is “Four State Full Logic”.



ProcessDataIn "Process Data In" id=V_Pd_InFourStateFullLogic

bit length: 8
data type: 8-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					Touch State	Touch State. Related parameters defined in output and touch settings parameter data.
2	1	2-bit UInteger	0 = State 1, 1 = State 2, 2 = State 3, 3 = State 4					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.

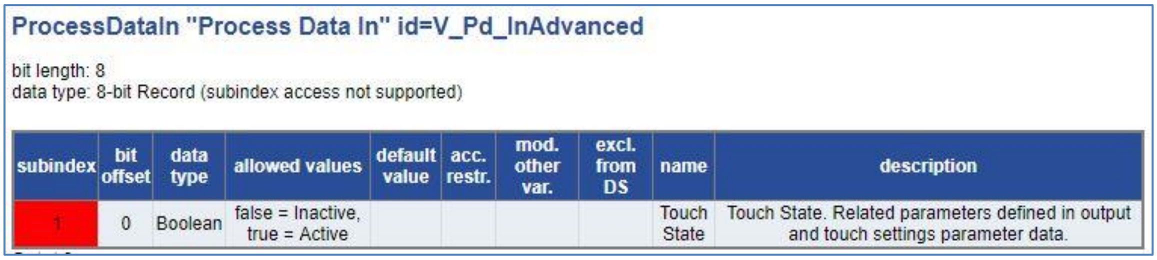


ProcessDataOut "Process Data Out" id=V_Pd_OutFourStateFullLogic

bit length: 72
data type: 72-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 = Off, true = On					Job Input	Job Input for Four State Full Logic mode.

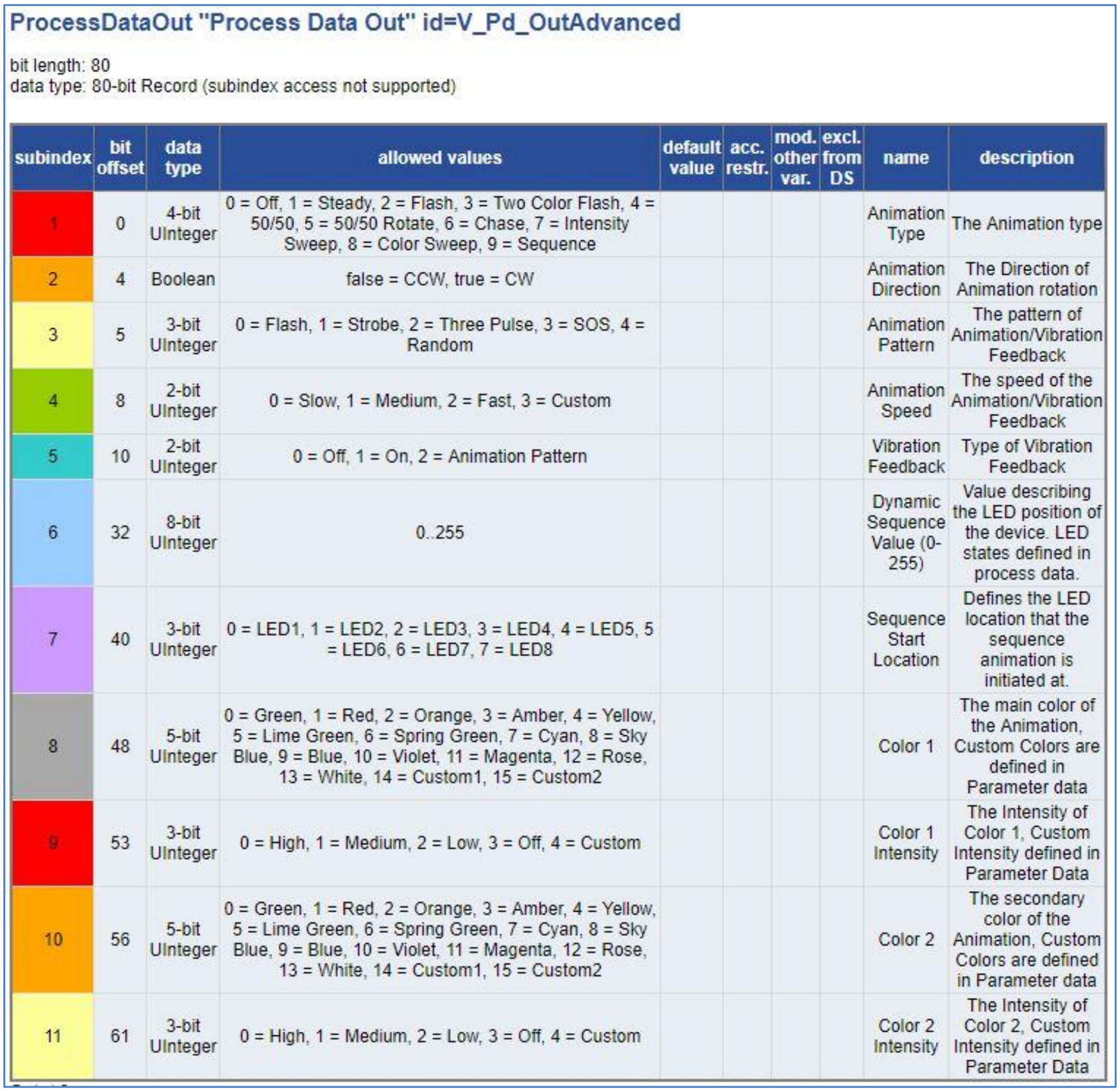
Mode 2 is “Advanced”.



ProcessDataIn "Process Data In" id=V_Pd_InAdvanced

bit length: 8
data type: 8-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					Touch State	Touch State. Related parameters defined in output and touch settings parameter data.

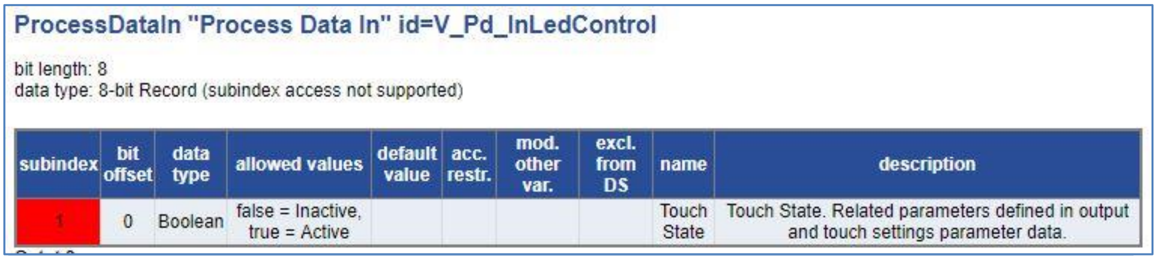


ProcessDataOut "Process Data Out" id=V_Pd_OutAdvanced

bit length: 80
data type: 80-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	4-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = 50/50, 5 = 50/50 Rotate, 6 = Chase, 7 = Intensity Sweep, 8 = Color Sweep, 9 = Sequence					Animation Type	The Animation type
2	4	Boolean	false = CCW, true = CW					Animation Direction	The Direction of Animation rotation
3	5	3-bit UInteger	0 = Flash, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Animation Pattern	The pattern of Animation/Vibration Feedback
4	8	2-bit UInteger	0 = Slow, 1 = Medium, 2 = Fast, 3 = Custom					Animation Speed	The speed of the Animation/Vibration Feedback
5	10	2-bit UInteger	0 = Off, 1 = On, 2 = Animation Pattern					Vibration Feedback	Type of Vibration Feedback
6	32	8-bit UInteger	0..255					Dynamic Sequence Value (0-255)	Value describing the LED position of the device. LED states defined in process data.
7	40	3-bit UInteger	0 = LED1, 1 = LED2, 2 = LED3, 3 = LED4, 4 = LED5, 5 = LED6, 6 = LED7, 7 = LED8					Sequence Start Location	Defines the LED location that the sequence animation is initiated at.
8	48	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, 14 = Custom1, 15 = Custom2					Color 1	The main color of the Animation, Custom Colors are defined in Parameter data
9	53	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
10	56	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, 14 = Custom1, 15 = Custom2					Color 2	The secondary color of the Animation, Custom Colors are defined in Parameter data
11	61	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

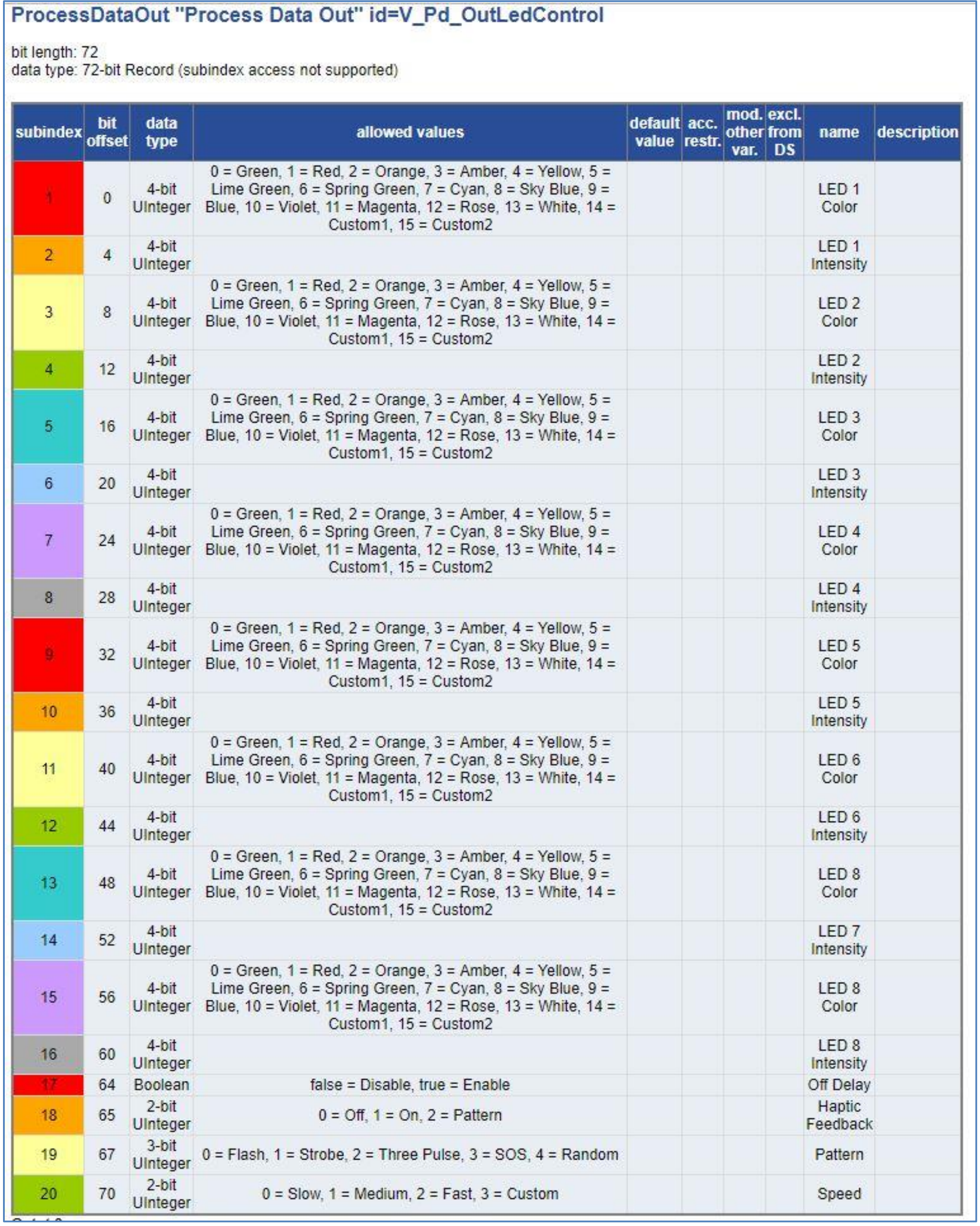
Mode 3 is “LED Control”.



ProcessDataIn "Process Data In" id=V_Pd_InLedControl

bit length: 8
data type: 8-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					Touch State	Touch State. Related parameters defined in output and touch settings parameter data.



ProcessDataOut "Process Data Out" id=V_Pd_OutLedControl

bit length: 72
data type: 72-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	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 = Custom1, 15 = Custom2					LED 1 Color	
2	4	4-bit UInteger						LED 1 Intensity	
3	8	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 = Custom1, 15 = Custom2					LED 2 Color	
4	12	4-bit UInteger						LED 2 Intensity	
5	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 = Custom1, 15 = Custom2					LED 3 Color	
6	20	4-bit UInteger						LED 3 Intensity	
7	24	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 = Custom1, 15 = Custom2					LED 4 Color	
8	28	4-bit UInteger						LED 4 Intensity	
9	32	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 = Custom1, 15 = Custom2					LED 5 Color	
10	36	4-bit UInteger						LED 5 Intensity	
11	40	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 = Custom1, 15 = Custom2					LED 6 Color	
12	44	4-bit UInteger						LED 6 Intensity	
13	48	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 = Custom1, 15 = Custom2					LED 8 Color	
14	52	4-bit UInteger						LED 7 Intensity	
15	56	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 = Custom1, 15 = Custom2					LED 8 Color	
16	60	4-bit UInteger						LED 8 Intensity	
17	64	Boolean	false = Disable, true = Enable					Off Delay	
18	65	2-bit UInteger	0 = Off, 1 = On, 2 = Pattern					Haptic Feedback	
19	67	3-bit UInteger	0 = Flash, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pattern	
20	70	2-bit UInteger	0 = Slow, 1 = Medium, 2 = Fast, 3 = Custom					Speed	