

K50 Pro Process Data Function

February 16th, 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 K50 Audible Touch, K50 Display, K50 Compact Audible, or Banner K50 Pro Touch lights via an IO-Link Master from a Siemens PLC. The function covers parsing and display of the K50 Pro sensor Process Data Out.

Components

Banner K50 Library 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.



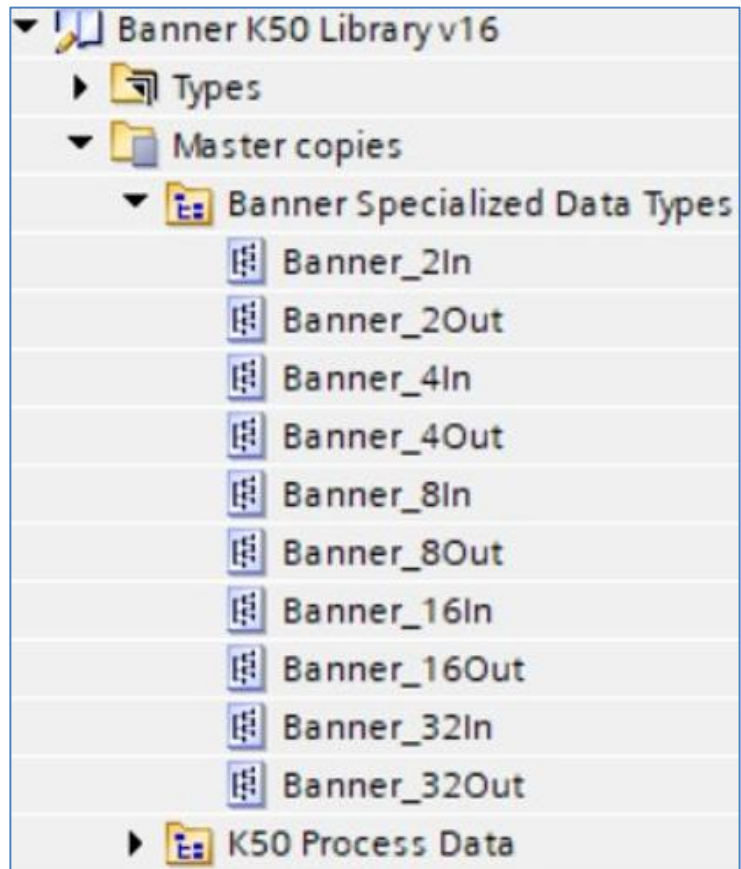
3. Switch the “Files of type” to Compressed libraries. Go to the location of the compressed library.
4. Press the Open button and the library will be uncompressed and opened.
5. The library is now accessible in the Libraries tab in v16 or greater.

Setup of K50 Pro with a Banner DXMR

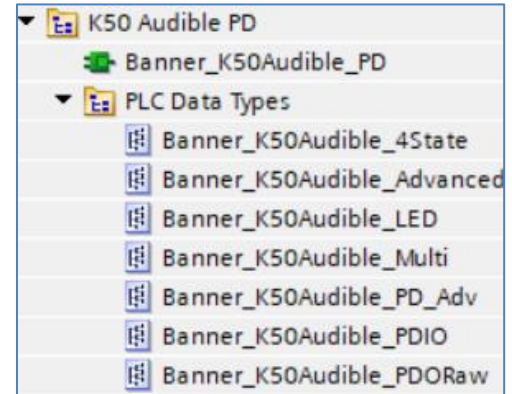
1. Go to Device and Networks to configure the DXMR. Add the DXMR if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR for IO-Link mode.
3. Open the IO-Link Generic Devices and select the proper module. The **16/16 byte** is required for **K50 Pro Touch** and **K50 Pro Audible**. The **K50 Compact Audible** uses **2/2 byte** and **K50 Display** uses **32/32 byte**. Make note of the input %I10 address for Slot 2 which represents Port 1. Slot 2 starts are %Q1 for outputs.

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

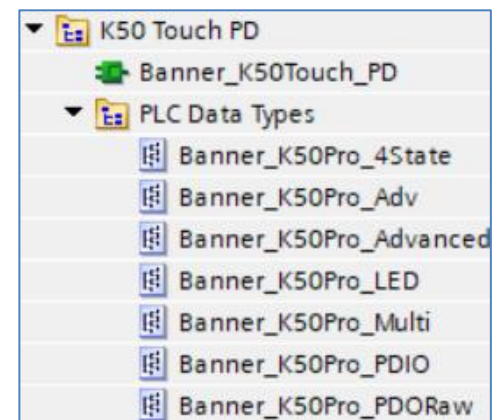
4. Drag the necessary tag from “Banner_Specialized_Data_Types”. The tag used in this example is “Banner_16Out” and “Banner_16In”. This tag represents the full raw process data along with port status information.
5. Open K50 Process Data folder.
6. Go to step 7 if using a K50 Pro Audible, step 8 if a K50 Pro Touch, and step 9 if a K50 Pro Display, and Step 10 if a K50 Pro Compact Audible is being used.



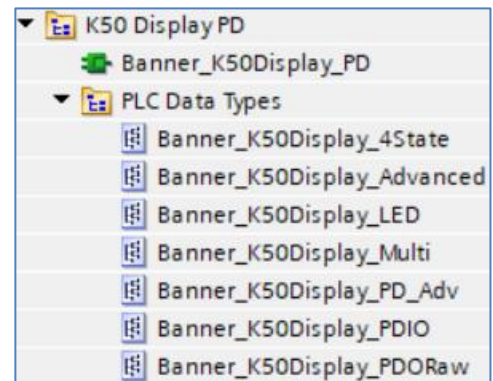
7. Drag the necessary files from the “K50 Audible PD” folder.
 - a. Move “Banner_K50Audible_4State”, “Banner_K50Audible_Advanced”, “Banner_K50Audible_LED”, “Banner_K50Audible_Multi”, “Banner_K50Audible_PD_Adv”, “Banner_K50Audible_PDIO”, and “Banner_K50Audible_PDORaw” to the PLC Data Types area.
 - b. Move “Banner_K50Audible_PD” to the Program Blocks area.



8. Drag the necessary files from the “K50 Touch PD” folder.
 - a. Move “Banner_K50Pro_4State”, “Banner_K50Pro_Adv”, “Banner_K50Pro_Advanced”, “Banner_K50Pro_LED”, “Banner_K50Pro_Multi”, “Banner_K50Pro_PDIO”, and “Banner_K50Pro_PDORaw” to the PLC Data Types area.
 - b. Move Banner_K50Touch_PD to the Program Blocks area.



9. Drag the necessary files from the “K50 Display PD” folder.
 - a. Move “Banner_K50Display_4State”, “Banner_K50Display_Advanced”, “Banner_K50Display_LED”, “Banner_K50Display_Multi”, “Banner_K50Display_PD_Adv”, “Banner_K50Display_PDIO”, and “Banner_K50Display_PDORaw” to the PLC Data Types Area.
 - b. Move “Banner_K50Display_PD” to the Program Blocks area.



10. Drag the necessary files from the “K50 Compact Audible PD” folder.
 - a. Move “Banner_K50_ComAud_PDO” to the PLC Data Types Area
 - b. Move “Banner K50 Compact Audible” to the Program Blocks Area.



11. 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 “K50 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 to which 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 3 (%Q1) is tied to this new tag. The tag that represents the raw data is “K50 IOLM1 01 outRaw” and uses the “Q” address found in step 3 (%Q3). Tags “K50 IOLM1 02 PDI” (%I10) and “K50 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
▶ K50 IOLM1 01 PDI	*Banner_16In*	%I10.0
K50 IOLM1 01 inRaw	UInt	%IW14
▶ K50 IOLM1 01 PDO	*Banner_16Out*	%Q1.0
▶ K50 IOLM1 01 outRaw	*Banner_K50Pro_PDORaw*	%Q3.0

K50 Pro Touch Example

▶ K50D IOLM1 02 inRaw	*Banner_32In*	%I32.0
K50D IOLM1 02 PDI	UInt	%IW36
▶ K50D IOLM1 02 outRaw	*Banner_32Out*	%Q42.0
▶ K50D IOLM1 02 PDO	*Banner_K50Display_PDORaw*	%Q44.0

K50 Display Example

▶ K50A IOLM1 03 inRaw	*Banner_16In*	%I68.0
K50A IOLM1 03 PDI	UInt	%IW72
▶ K50A IOLM1 03 outRaw	*Banner_16Out*	%Q88.0
▶ K50A IOLM1 03 PDO	*Banner_K50Audible_PDORaw*	%Q90.0

K50 Audible Example

▶ K50CA IOLM1 04 inRaw	*Banner_2In*	%I88.0
▶ K50CA IOLM1 04 outRaw	*Banner_2Out*	%Q118.0
K50CA IOLM1 04 PDO	Byte	%QB120

K50 Compact Audible Example

12. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.
13. The rest of the steps will focus on the K50 Pro Touch model. The K50 Pro Audible and Display follow similar steps.

14. In the new data block, create a new tag to represent the parsed Process Data Output for our K50 Pro. 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	
■ ▼ K50 IOLM1 01 PD	"Banner_K50Pro_PDIO"
■ ▶ MultiColor	"Banner_K50Pro_Multi"
■ ▶ Four State	"Banner_K50Pro_4State"
■ ▶ Advanced	"Banner_K50Pro_Adv"
■ ▶ LED	"Banner_K50Pro_LED"

15. 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 “K50 IOLM1 01 outRaw” in this example. Link the “PDI” to the raw process data variable from step 5. Use the variable called “K50 IOLM1 01 inRaw” from step 5. The “K50 Pro PD” needs to be linked to the variable created in step 7. It was called “K50 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 K50 Pro, there are four user-selected modes for the Process Data Out. This function needs to know what choice has been made in the K50 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 K50 Pro 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.

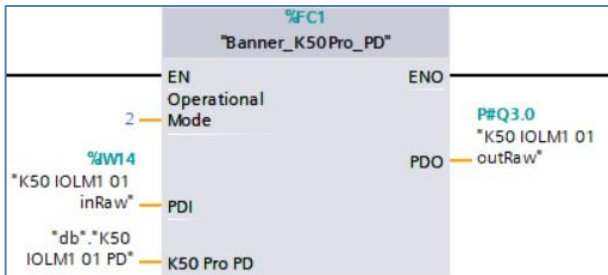


Figure 1: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the light’s current Operational Mode) 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”.

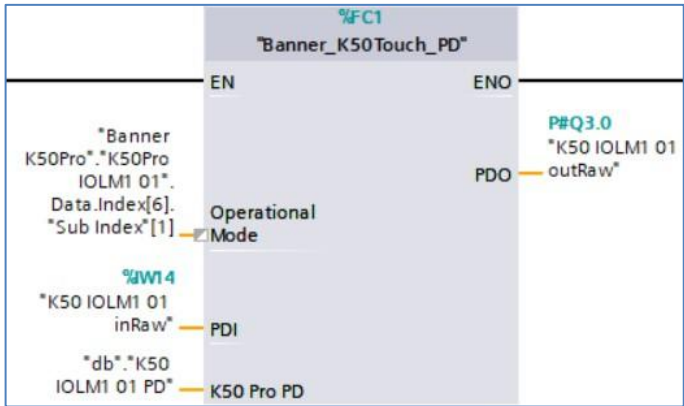


Figure 2: Linking Operational Mode variable to K50 Pro Touch Parameter Data Function Block

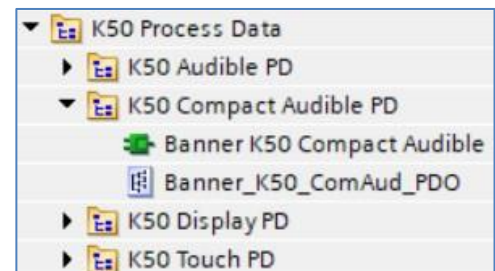
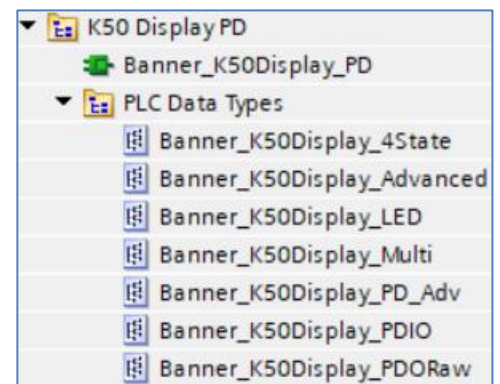
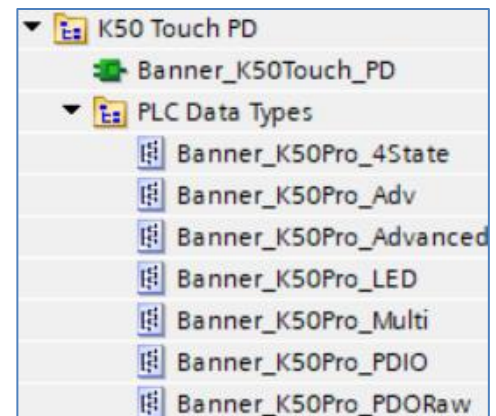
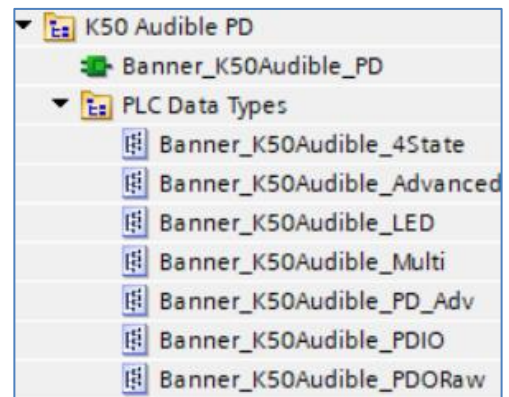
16. The final step is to configure the IO-Link output control. This is done by sending a 1 to Port Control and a 2 to Port Config. Both parameters are part of the tag created in step 6 "K50 IOLM1 01 PDO".



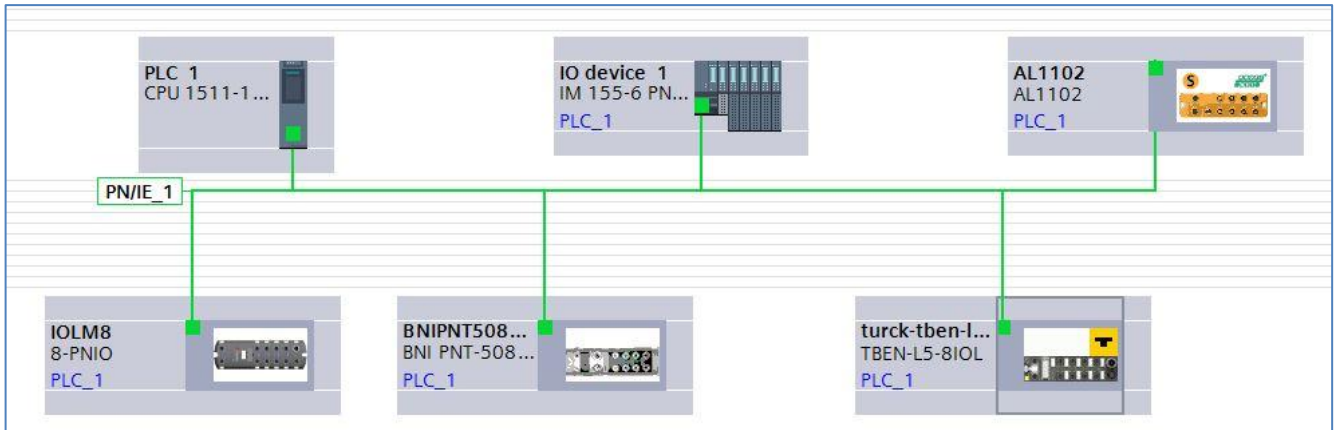
17. Process Data Setup is complete.
18. Compile and download the configuration to the PLC, then go online. Open the "db" data block and click Monitor all. The K50 Pro can be controlled now.

Setup of K50 Pro with other IO-Link Masters

1. The Banner K50 library will now be in the Global Library List. Expand the Master copies section. Open the K50 Process Data folder.
2. Go to step 3 if using a K50 Pro Audible, step 4 if a K50 Pro Touch, and step 5 if a K50 Pro Display, and Step 6 if a K50 Pro Compact Audible is being used.
3. Drag the necessary files from the “K50 Audible PD” folder.
 - a. Move “Banner_K50Audible_4State”, “Banner_K50Audible_Advanced”, “Banner_K50Audible_LED”, “Banner_K50Audible_Multi”, “Banner_K50Audible_PD_Adv”, “Banner_K50Audible_PDIO”, and “Banner_K50Audible_PDORaw” to the PLC Data Types area.
 - b. Move “Banner_K50Audible_PD” to the Program Blocks area.
4. Drag the necessary files from the “K50 Touch PD” folder.
 - c. Move “Banner_K50Pro_4State”, “Banner_K50Pro_Adv”, “Banner_K50Pro_Advanced”, “Banner_K50Pro_LED”, “Banner_K50Pro_Multi”, “Banner_K50Pro_PDIO”, and “Banner_K50Pro_PDORaw” to the PLC Data Types area.
 - d. Move “Banner_K50Pro_PD” to the Program Blocks area.
5. Drag the necessary files from the “K50 Display PD” folder.
 - e. Move “Banner_K50Display_4State”, “Banner_K50Display_Advanced”, “Banner_K50Display_LED”, “Banner_K50Display_Multi”, “Banner_K50Display_PD_Adv”, “Banner_K50Display_PDIO”, and “Banner_K50Display_PDORaw” to the PLC Data Types Area.
 - f. Move “Banner_K50Display_PD” to the Program Blocks area.
6. Drag the necessary files from the “K50 Compact Audible PD” folder.
 - g. Move “Banner_K50_ComAud_PDO” to the PLC Data Types Area
 - h. Move “Banner K50 Compact Audible” to the Program Blocks Area.



- 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.



- Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. **16in/16out byte data** is required for **K50 Pro Touch** and **K50 Pro Audible**. The **K50 Compact Audible** uses **2in/2out byte** and **K50 Display** uses **32in/32out byte**.
- Record the “I” and “Q” addresses where this K50 Process Data is to be stored, as these addresses will be required in the next step.

10. Go to PLC Tags. Create 2 tags for K50 Pro Touch, 2 tags for K50 Pro Display, 2 tags for K50 Pro Audible, and 1 tag for K50 Pro Compact Audible. In this example, Tag table_1 was created, then the tag “K50 IOLM1 01 PDO” was created using a Data Type of “Banner_K50Pro_PDORaw”. This naming convention calls out the type of device in question as well as the specific IO-Link Master and port number to which 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 3 (%Q14) is tied to this new tag. Tags “K50 IOLM1 01 PDI” (%IW16). This is the tag that will be used in the Function block.

▶ K50 IOLM1 01 PDO	"Banner_K50Pro_PDORaw"	%Q14.0
K50 IOLM1 01 PDI	UInt	%IW16

K50 Pro Touch Example

K50D IOLM1 02 PDI	UInt	%IW36
▶ K50D IOLM1 02 PDO	"Banner_K50Display_PDORaw"	%Q44.0

K50 Display Example

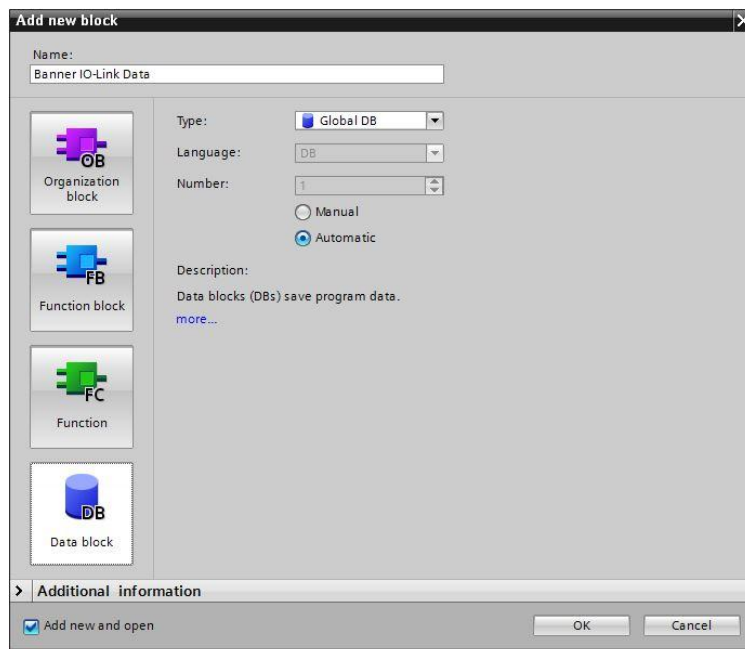
K50A IOLM1 03 PDI	UInt	%IW72
▶ K50A IOLM1 03 PDO	"Banner_K50Audible_PDORaw"	%Q90.0

K50 Audible Example

K50CA IOLM1 04 PDO	Byte	%QB120
--------------------	------	--------

K50 Compact Audible Example

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



12. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.
13. The rest of the steps will focus on the K50 Pro Touch model. The K50 Pro Audible and Display follow similar steps.
14. In the new data block, create a new tag to represent the parsed Process Data Output for our K50 Pro. 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	
■ ▼ K50 IOLM1 01 PD	"Banner_K50Pro_PDIO"
■ ▶ MultiColor	"Banner_K50Pro_Multi"
■ ▶ Four State	"Banner_K50Pro_4State"
■ ▶ Advanced	"Banner_K50Pro_Adv"
■ ▶ LED	"Banner_K50Pro_LED"

15. 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 “K50 IOLM1 01 outRaw” in this example. Link the “PDI” to the raw process data variable from step 5. Use the variable called “K50 IOLM1 01 inRaw” from step 10. The “K50 Pro PD” needs to be linked to the variable created in step 14. It was called “K50 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 K50 Pro, there are four user-selected modes for the Process Data Out. This function needs to know what choice has been made in the K50 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 K50 Pro 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.

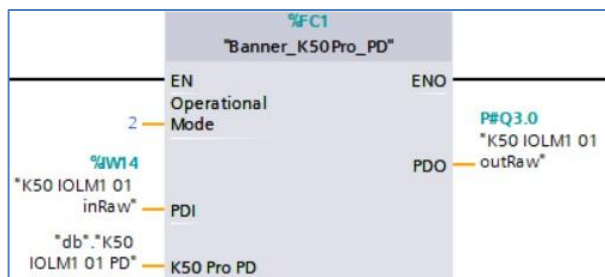


Figure 3: Hand type correct number for Operational Mode

NOTE: if you type in the incorrect number (i.e. it does not match the light’s current Operational Mode) 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”.

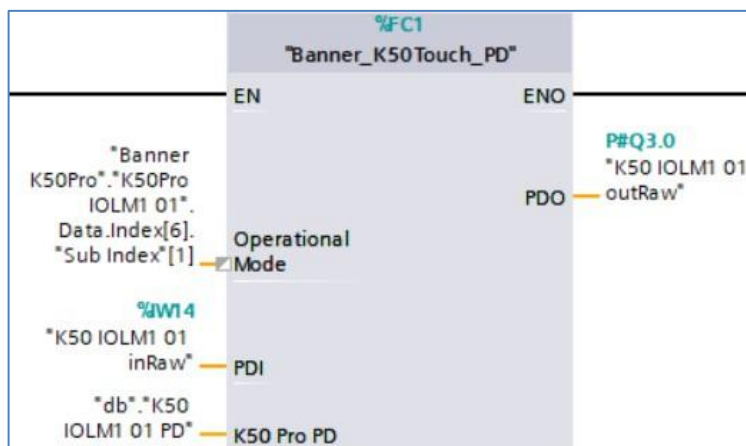


Figure 4: Linking Operational Mode variable to K50 Pro Touch Parameter Data Function Block

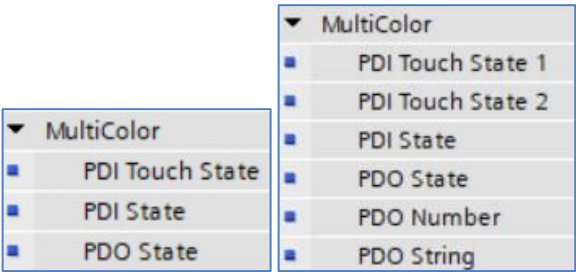
16. Process Data Setup is complete.

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.

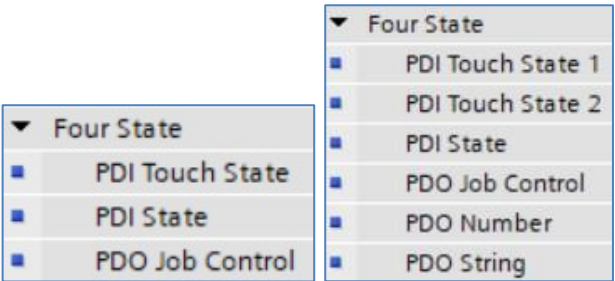
Using K50 Pro Process Data

IO-Link Devices K50 Audible, K50Display and K50 Touch all have similar Process Data. The K50 Compact Audible is different than the other three. Below each of the process data modes will be described for the devices. There are slight differences for the three units that are similar, but the overall operation follows the same pattern.

- 1. K50 Audible, Display, and Touch
 - a. Multicolor Mode (0)
 - i. PDI Touch State: gives the status of the touch button. K50 Display has two tags for touch state.
 - ii. PDI State: Current state that the device is in.
 - iii. PDO State: Set by a user to tell the device what state should be displayed.
 - iv. PDO Number: Used only by K50 Display units. Device must be in number mode. Value entered into this location is displayed on the device.
 - v. PDO String: Used only by K50 Display units. The device must be in string mode. String entered into this location is displayed on the device.

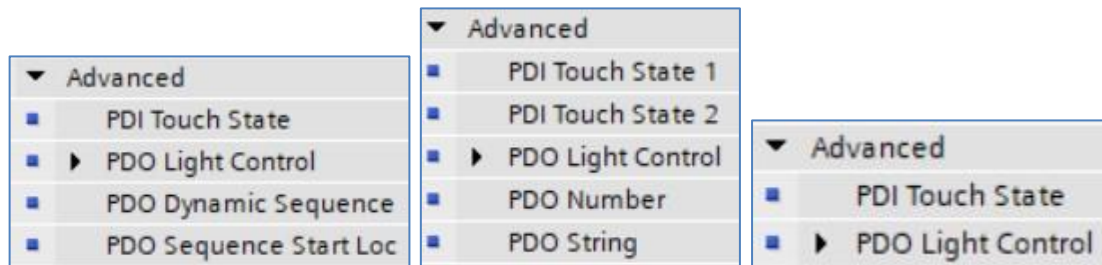


- b. Four State Full Logic Mode (1)
 - i. PDI Touch State: gives the status of the touch button. K50 Display has two tags for touch state.
 - ii. PDO State: Current state that the device is in.
 - iii. PDO Job Control: Used to control which of the 4 modes the unit is in along with the touch state.
 - iv. PDO Number: Used only by K50 Display units. Device must be in number mode. Value entered into this location is displayed on the device.
 - v. PDO String: Used only by K50 Display units. The device must be in string mode. String entered into this location is displayed on the device.



c. Advanced Mode (2, Default)

- i. PDI Touch State: gives the status of the touch button. K50 Display has two tags for touch state.
- ii. PDO Light Control: tags that control the light portion of the device.
- iii. PDO Number: Used only by K50 Display units. Device must be in number mode. Value entered into this location is displayed on the device.
- iv. PDO String: Used only by K50 Display units. The device must be in string mode. String entered into this location is displayed on the device.



d. LED Mode (3)

- i. PDI Touch State: gives the status of the touch button. K50 Display has two tags for touch state.
- ii. LED (1 to 8) Color: Sets the color for the LED. Each LED can have a different color.
- iii. LED Intensity (1 to): Sets the intensity of the LED.
- iv. PDO Number: Used only by K50 Display units. Device must be in number mode. Value entered into this location is displayed on the device.
- v. PDO String: Used only by K50 Display units. The device must be in string mode. String entered into this location is displayed on the device.
- vi. Vibration Tags: Used by certain K50 Touch models that have vibration feedback. See IODD file for more information.
- vii. Audio Tags: Used by K50 Audible models. Controls the audible portion of the K50. See IODD file for more information.

2. K50 Compact Audible (0, 2)

a. Multicolor Mode (0): Value set there places the K50 Compact Audible into that mode.

a. Advanced Mode (2)

- i. Audible Feedback: Turns the audible on or off.
- ii. Audible Volume: Controls the volume of the audible.
- iii. Audible Type: Controls the style of the audible.

Appendix A**K50 Pro Touch Process Data**

The K50 Pro Touch has 2 bytes of Process Data In and 10 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, "Multicolor".

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	

Appendix B**K50 Pro Audible Process Data**

The K50 Pro Audible has 2 bytes of Process Data In and 10 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, “Multicolor”.

ProcessDataIn "Process Data In" id=V_Pd_InMulticolor

bit length: 16

data type: 16-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					Output State	Output State. Related parameters defined in output and touch settings parameter data.
2	8	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: 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	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.

Mode 1 is “4State”.

ProcessDataIn "Process Data In" id=V_Pd_InFourStateFullLogic

bit length: 16

data type: 16-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					Output State	Output State. Related parameters defined in output and touch settings parameter data.
2	8	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: 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	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: 16

data type: 16-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					Output State	Output 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	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, 10 = Wave, 11 = Double Wave					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/Audio Feedback
4	8	2-bit UInteger	0 = Slow, 1 = Medium, 2 = Fast, 3 = Custom					Animation Speed	The speed of the Animation/Audio Feedback
5	10	2-bit UInteger	0 = Off, 1 = On, 2 = Animation Pattern, 3 = Advanced Audible					Audio Feedback	Type of Audio Feedback
6	12	Boolean	false = Leading Edge, true = Trailing Edge					Off Delay Type	A leading edge delay is triggered on the rising edge of a touch input. A trailing edge delay is triggered on a the falling edge of a touch input.
7	16	16-bit UInteger	0..65535					Off Delay (ms)	Length of time before the device returns to 'touch inactive' state after button is released.
8	32	8-bit UInteger	0..255					Static Sequence Value (0-255)	Value describing the LED position of the device. LED state defined in parameter data.
9	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.
10	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
11	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
12	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
13	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
14	64	2-bit UInteger	1 = Low, 3 = High					Audio Volume	The volume of the piezzo buzzer
15	66	6-bit UInteger	0 = Pulse, 1 = Wobble, 2 = Strobe, 3 = Whoop, 4 = Staccato, 5 = Siren, 6 = Continuous 1, 7 = Continuous 2, 9 = Jingle, 10 = Melody 1, 11 = Melody 2, 12 = Melody 3, 13 = Custom					Audio Type	The type of audio sound being played

Mode 3 is "LED".

ProcessDataIn "Process Data In" id=V_Pd_InLedControl

bit length: 16

data type: 16-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					Output State	Output State. Related parameters defined in output and touch settings parameter data.

ProcessDataOut "Process Data Out" id=V_Pd_OutLedControl

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 = 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	Defines the color of the designated LED. LED 1 is oriented at the 12 O'clock position
2	4	4-bit UInteger	0..10					LED 1 Intensity (0-10)	Defines the intensity of the designated LED
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	Defines the color of the designated LED
4	12	4-bit UInteger	0..10					LED 2 Intensity (0-10)	Defines the intensity of the designated LED
17	64	2-bit UInteger	0 = Off, 1 = On, 2 = Animation Pattern, 3 = Advanced Audible					Audio Feedback	Type of Audio Feedback
18	66	2-bit UInteger	1 = Low, 3 = High					Audio Volume	The volume of the piezzo buzzer
19	72	6-bit UInteger	0 = Pulse, 1 = Wobble, 2 = Strobe, 3 = Whoop, 4 = Stacatto, 5 = Siren, 6 = Continuous 1, 7 = Continuous 2, 9 = Jingle, 10 = Melody 1, 11 = Melody 2, 12 = Melody 3, 13 = Custom					Audio Type	The type of audio sound being played

Appendix C

K50 Pro Display Process Data

The K50 Pro Display has 2 bytes of Process Data In and 19 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, "Multicolor".

ProcessDataIn "Process Data In" id=V_Pd_InMulticolor

bit length: 16

data type: 16-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					Output State Area 1	Output State. Related parameters defined in output and touch settings parameter data.
2	1	Boolean	false = Inactive, true = Active					Output State Area 2	Output State. Related parameters defined in output and touch settings parameter data.
3	8	3-bit UInteger	0 = State 1, 1 = State 2, 2 = State 3, 3 = State 4, 4 = State 5					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.

ProcessDataOut "Process Data Out" id=V_Pd_OutMulticolor

bit length: 152

data type: 152-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	3-bit UInteger	0 = State1, 1 = State2, 2 = State3, 3 = State4, 4 = State5					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.
2	72	16-bit UInteger						Number	Number to display
3	88	8-octet String US_ASCII						String (ASCII)	String to display

Mode 1 is “4State”.

ProcessDataIn "Process Data In" id=V_Pd_InFourStateFullLogic

bit length: 16

data type: 16-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					Output State Area 1	Output State. Related parameters defined in output and touch settings parameter data.
2	1	Boolean	false = Inactive, true = Active					Output State Area 2	Output State. Related parameters defined in output and touch settings parameter data.
3	8	3-bit UInteger	0 = State 1, 1 = State 2, 2 = State 3, 3 = State 4, 4 = State 5					State	Animation State. Related parameters defined in Four State Full Logic/Multicolor parameter data.

ProcessDataOut "Process Data Out" id=V_Pd_OutFourStateFullLogic

bit length: 152

data type: 152-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.
2	72	16-bit UInteger						Number	Number to display
3	88	8-octet String US_ASCII						String (ASCII)	String to display

Mode 2 is “Advanced”.

ProcessDataIn "Process Data In" id=V_Pd_InAdvanced

bit length: 16

data type: 16-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					Output State Area 1	Output State. Related parameters defined in output and touch settings parameter data.
2	1	Boolean	false = Inactive, true = Active					Output State Area 2	Output State. Related parameters defined in output and touch settings parameter data.

ProcessDataOut "Process Data Out" id=V_Pd_OutAdvanced

bit length: 152

data type: 152-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, 10 = Wave, 11 = Double Wave, 12 = Steady Area 1, 13 = Steady Area 2, 14 = Alternate Area 1/Area 2					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
4	8	2-bit UInteger	0 = Slow, 1 = Medium, 2 = Fast, 3 = Custom					Animation Speed	The speed of the Animation
7	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.
8	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.
9	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
10	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
11	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
12	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
13	72	16-bit UInteger						Number	Number to display
14	88	8-octet String US_ASCII						String (ASCII)	String to display

Mode 3 is “LED”.

ProcessDataIn "Process Data In" id=V_Pd_InLedControl

bit length: 16

data type: 16-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					Output State Area 1	Output State. Related parameters defined in output and touch settings parameter data.
2	1	Boolean	false = Inactive, true = Active					Output State Area 2	Output State. Related parameters defined in output and touch settings parameter data.

ProcessDataOut "Process Data Out" id=V_Pd_OutLedControl

bit length: 152

data type: 152-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	Defines the color of the designated LED. LED 1 is oriented at the 12 O'clock position
2	4	4-bit UInteger	0..10					LED 1 Intensity (0-10)	Defines the intensity of the designated LED
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	Defines the color of the designated LED
4	12	4-bit UInteger	0..10					LED 2 Intensity (0-10)	Defines the intensity of the designated LED
17	72	16-bit UInteger						Number	Number to display
18	88	8-octet String US_ASCII						String (ASCII)	String to display

Appendix D**K50 Compact Audible Process Data**

The K50 Compact Audible has 1 byte of Process Data Out. There are two 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, "Multicolor".

ProcessDataOut "Process Data Out" id=V_Pd_OutMulticolor

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	2-bit UInteger	0 = State1, 1 = State2, 2 = State3, 3 = State4					State	Animation State. Related parameters defined in Multicolor parameter data.

Mode 2 is "Advanced".

ProcessDataOut "Process Data Out" id=V_Pd_OutAdvanced

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	2-bit UInteger	0 = Off, 3 = On					Audible Feedback	Type of Audible Feedback
2	2	2-bit UInteger	0 = Off, 1 = Low, 2 = Medium, 3 = High					Audible Volume	The volume of the piezzo buzzer
3	4	4-bit UInteger	0 = Pulse, 1 = Wobble, 2 = Strobe, 3 = Whoop, 4 = Stacatto, 5 = Siren, 6 = Continuous 1, 7 = Continuous 2, 9 = Jingle, 10 = Melody 1, 11 = Melody 2, 12 = Melody 3, 13 = Custom					Audible Type	The type of audible sound being played