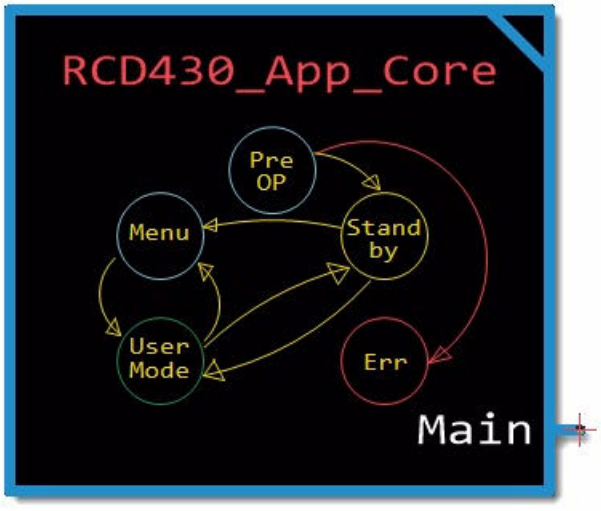
ENGINEERING
TOMORROW

User Manual

PLUS+1® GUIDE Software
RCD430 Application Library



Revision history

Table of revisions

Date	Changed	Rev
June 2024	First edition	0101

Contents

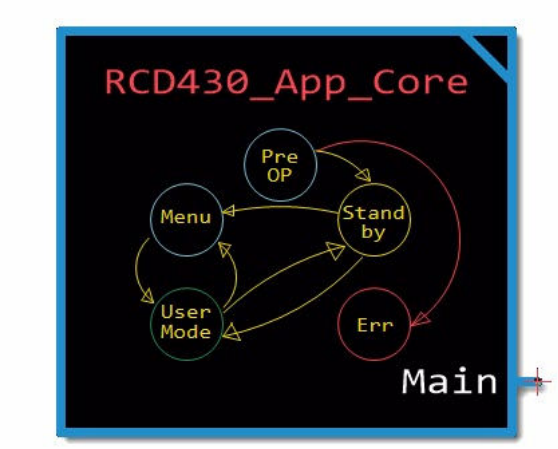
Introduction

The RCD430 Application Library contains function blocks used to customize the RCD430 Application to transmit and receive messages from the display and activate the buzzer on the transmitter.

The RCD430 Application supports only one of each of the function blocks in the RCD430 Application Library.

RCD430_App_Core Function Block

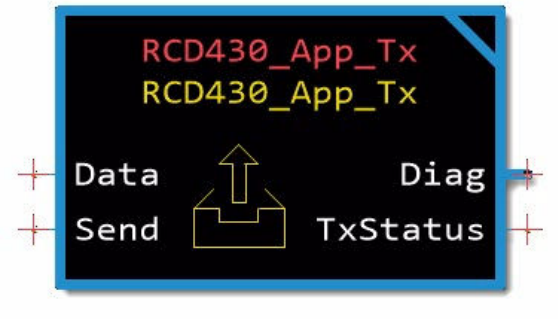
The RCD430_App_Core function block enables the RCD430 Display in the RCD430 Application and calls predefined screen definitions.



For more information, refer to *RCD430_App_Core Function Block* on page 18.

RCD430_App_TX Function Block

The RCD430_App_TX function block transmits CAN messages to the receiving machine application.

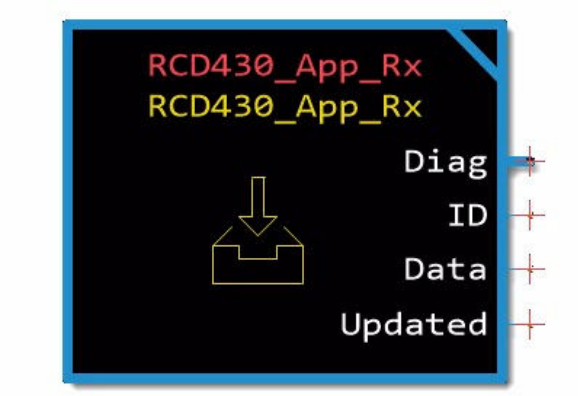


For more information, refer to *RCD430_App_TX Function Block* on page 19.

RCD430_App_Rx Function Block

The RCD430_App_Rx function block receives CAN messages from the machine application.

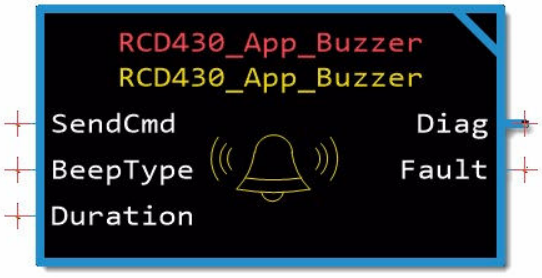
Introduction



For more information, refer to *RCD430_App_Rx Function Block* on page 19.

RCD430_App_Buzzer Function Block

The RCD430_App_Buzzer configures and activates the buzzer on the RCT Device.



For more information, refer to *RCD430_App_Buzzer Function Block* on page 20.

RCD430 Display Widgets

The RCD430 Application Library includes widgets used in the RCD430 Application.

The widgets are used in the predefined screens and are unavailable to use in custom screens in this version.

PLUS+1® GUIDE Requirement

The RCD430 Application Library requires PLUS+1® GUIDE version 2024.1 or later.

RCT Device and Machine Communication

The following describes how the RCT Device communicates with the machine receiver in an RCT System. It also describes the role of each component involved in communication.

RCT Device

The RCT Device contains customized console box hardware and a display with a microprocessor. The operator uses it to remotely control a machine.

The RCT Device must have the Ikuflex option enabled in the system configuration file.

1. Console Box Hardware

Introduction

Customized joystick, buttons, and auxiliary hardware used to control the machine remotely. It transmits the control signals directly to the receiver.

2. RCD430 Display

   Shows predefined and custom screens configured in the display application. It also contains a microprocessor that holds the display application.

   a. Display Application

      Contains the RCD430 Application with predefined and custom screen definitions. It also contains logic used to communicate with the machine application from the display. For examples, refer to *Configuration and Examples* on page 8.

Machine

The machine is any device or vehicle controlled by the RCT Device.

1. Danfoss Receiver

   Receives signals from the hardware control and CAN messages from the display application to be used in the machine application. One of the following Danfoss Receivers are required:

   - MPCAN
   - R13F
   - MP20A
   - MP20V
   - MP08V

   Other receivers can be used but they do not function with the RCD430 Application Library

   The Danfoss Receiver must have the Ikuflex option enabled in the EEPROM file.

2. Machine Controller

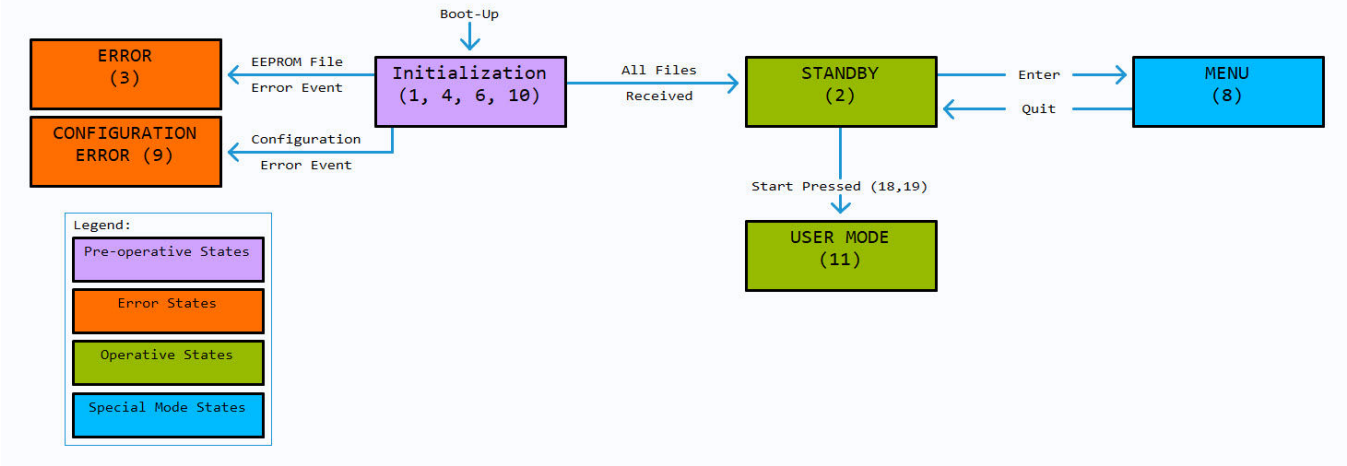
   The controller on the machine that contains the machine application. The Danfoss Receiver is connected to the machine controller allowing it to receive signals from the transmitter console box hardware and display application.

   a. Machine Application

      Receives signals from the transmitter console box hardware and uses them to drive the machine. It also communicates with the RCD430 Application. Contains logic used to drive the machine using the signals from the transmitter console box hardware

RCD430 Application State Diagram

The following diagram shows the progression of states of the RCD430_App_Core after booting up.



Introduction

- **Initialization**(States 1,4,6,10): After the RCD430 Display is powered on, the RCD430 Display initializes the core code and reads the configuration and EEPROM files.
- **Error** (State 3): The RCD430 Display encountered an issue with the EEPROM file.
- **Configuration Error** (State 9): The RCD430 Display encountered an issue with the configuration files.
- **Standby** (State 2): All files received and are valid. The display on the RCD430 Display displays the standby screen.
- **Menu** (State 8): The user pressed the scroll wheel to enter the configuration menu. To exit the menu, select **Quit**.
- **User Mode** (State 11): Custom screens are enabled.

For a full list of the states, refer to *RCD430_App_Core Function Block Outputs* on page 18.

Configuration and Examples

This section contains instructions for configuring the RCD430 Application and machine application.

Configuring the RCD430 Application

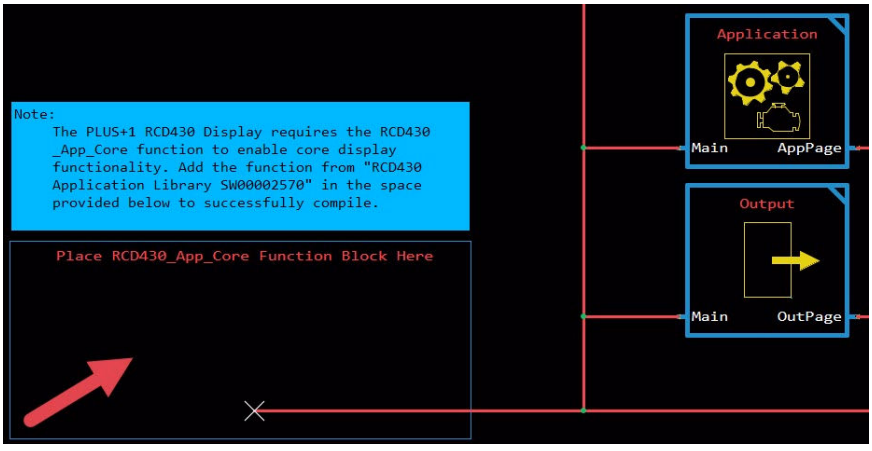
This section contains instructions for configuring the RCD430 Application.

Prepare the RCD430 Application

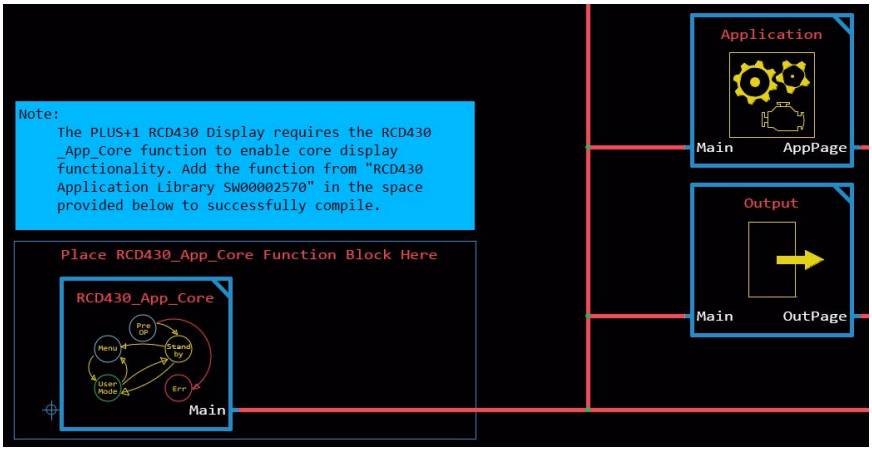
The RCD430 Application can be downloaded from the PLUS+1 Update Center. Follow these steps to download the RCD430 Application and prepare it for use with the RCD430 Display.

1. Download the RCD430 Application from PLUS+1 Update Center.

   The application is located in the **Files** > **Applications** section.
2. Open the application in PLUS+1 GUIDE.
3. Enter the **Application** page.
4. Place the RCD430_App_Core function block in the lower-left corner.



5. Route the **Main** output on the RCD430_App_Core function block to the **Main** bus in the application.



Configuring the Machine Application

This section contains instructions for configuring the machine application. The machine application receives signals from the RCT Device. Then, using the system configuration file, the machine application

Configuration and Examples

interprets the signals into input signals that can be use in the application. This section contains examples that show possible machine application configurations that use the signals from the RCT Device to control the machine.

Refer to the following for more information on configuring the machine application:

- *Add the System Configuration File* on page 9
- *RCT System Receiver Example* on page 9

Add the System Configuration File

The system configuration file is an XML file that defines input signals from the RCT allowing the application to use the signals to control a machine. Danfoss provides it with the transmitter hardware. Follow these steps to add the system configuration file to the application.

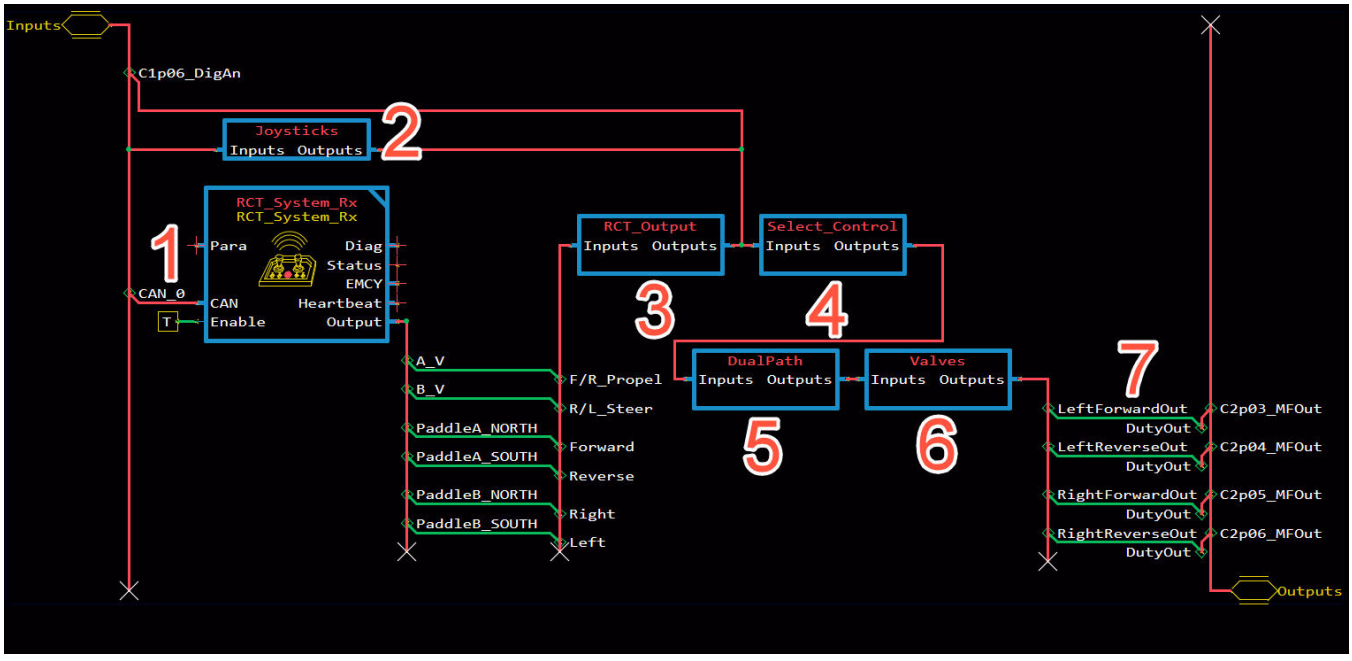
Before completing this task, locate the system configuration file that came with the RCT Device.

1. Move the system configuration file to the project folder.
2. Add the **RCT_System_Rx** function block to the application.
   PLUS+1 GUIDE adds multiple files including a generic system configuration file called **RCTConfiguration.xml**.
3. Delete the **RCTConfiguration.xml** file from the Project Manager.
4. In the **Project Manager**, right click on **PLC Units** then select **Add Existing PLC Unit**.
   The **Select PLC Project File** dialog box opens.
5. Select the system configuration file then click **Open**.
   PLUS+1 GUIDE loads the RCT configuration file.

RCT System Receiver Example

The following example shows an application that configures RCT and on-board joysticks to control the machine. The application receives signals from the RCT using the RCT_System_Rx function block, configures on-machine joysticks, switches between the two signals, and converts those signals to control the machine and show on the on-board display.

Before following this example, follow the steps in *Add the System Configuration File* on page 9.



Configuration and Examples

1. The RCT_System_Rx function block receives inputs from the remote transmitter defined and configured by the system configuration file received with the RCT. It outputs signals used to drive the machine. On the new bus, the signals are renamed to indicate their purpose in the application.

   The RCT_System_Rx function block is in the Remote Controls Library.
2. The **Joysticks** page configures the on-board joysticks. One joystick is configured to steer the machine and the other is configured to control propulsion. These joysticks are the primary method for controlling the machine.
3. The **RCT_Output** page converts the output values from the **RCT_System_Rx** function block into propel and steer signals.
4. The **Select_Control** page switches control signals between the joystick on the machine and the RCT. When the **DigIn** input on the **C1p06_DigAn** bus is True, the application uses the on-board joystick to drive the machine. This signal can be controlled by a switch or any other Boolean input.
5. The **DualPath** page configures the RCT or on-board joystick signals to control a dual-path machine.
6. The **Valves** page uses the dual path signals to control the four valves that drive the machine.
7. The **LeftForwardOut**, **LeftReverseOut**, **RightForwardOut**, and **RightReverseOut** are connected to the **DutyOut** signals in the output pins connected to the pumps.

Examples

This section contains examples that show the RCD430 Application and machine application configured to accomplish a goal.

- *Machine Fuel Level Examples* on page 11
- *Light Switch Example* on page 13
- *RCD430_App_Buzzer Function Block Example* on page 15

Configuration and Examples

Machine Fuel Level Examples

The following examples show the RCD430 Application configured to receive the fuel level from the machine application and the machine application transmitting the fuel level data to the RCD430 Application.

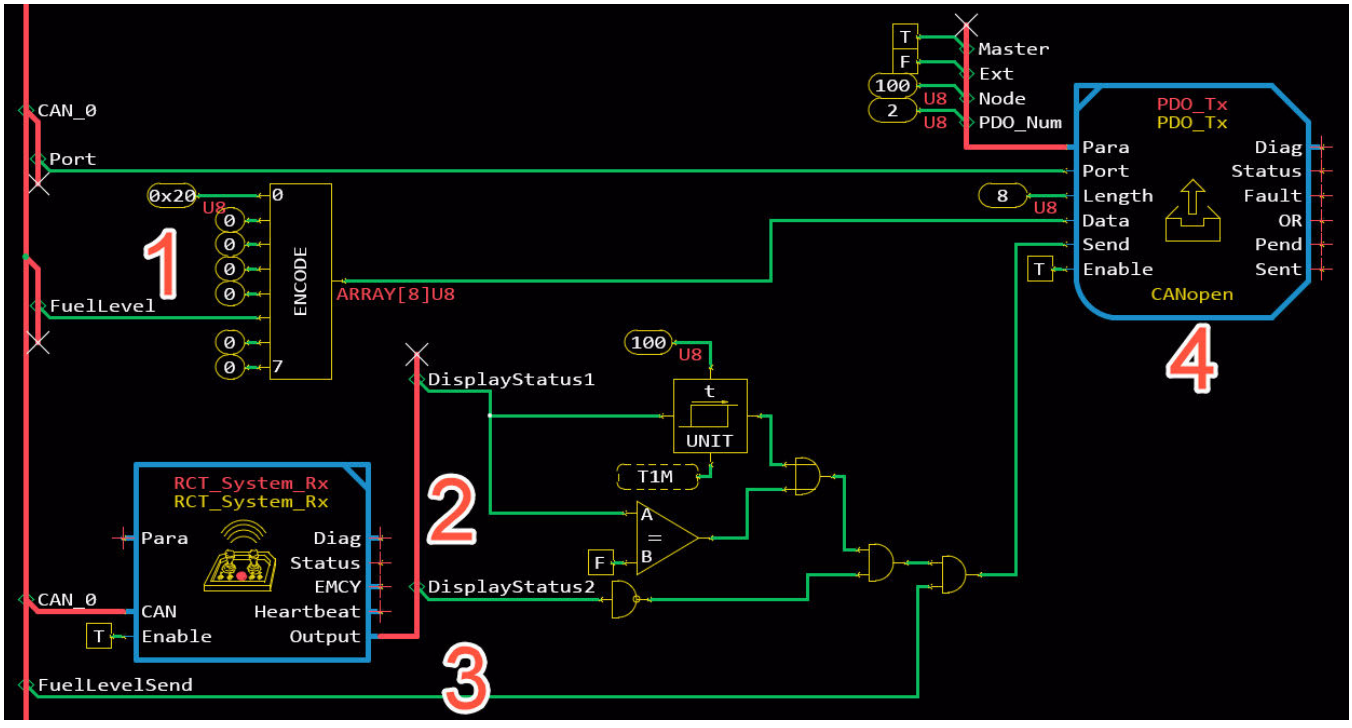
Requirements

- Install the CANopen FB Library.
- Configure the machine app to output the fuel level to the application.

- *Transmit the Fuel Level from the Machine Application* on page 11
- *Receive Fuel Level on the RCD430 Application* on page 12

Transmit the Fuel Level from the Machine Application

The following example shows the machine application configured to transmit the level of the fuel on the machine to the RCD430 Application on the RCD430 Display.



1. The **FuelLevel** signal is converted into an Array[8]U8 data type and combined with a message ID.
   - The **FuelLevel** signal supplies the fuel level value from the machine. This can be provided by a sensor in the fuel tank. Then, an **Encode 8** component converts the signal in to a **Array[8]U8** data type with the fuel level value at the 5th element.
   - The first element in the array sets the ID of the message to 0x20.This is connected to the **Data** input on the **Transmit CAN** component.

     0x20 is an example of an ID. You can use any value to identify the message. The receiver must validate that the ID matches the value set here.

   - The output of the **Encode 8** component is connected to the **Data** input on the PDO_Tx function block.
2. The RCT_System_Rx function block receives status information from the transmitter and the application uses that to queue messages to transmit.

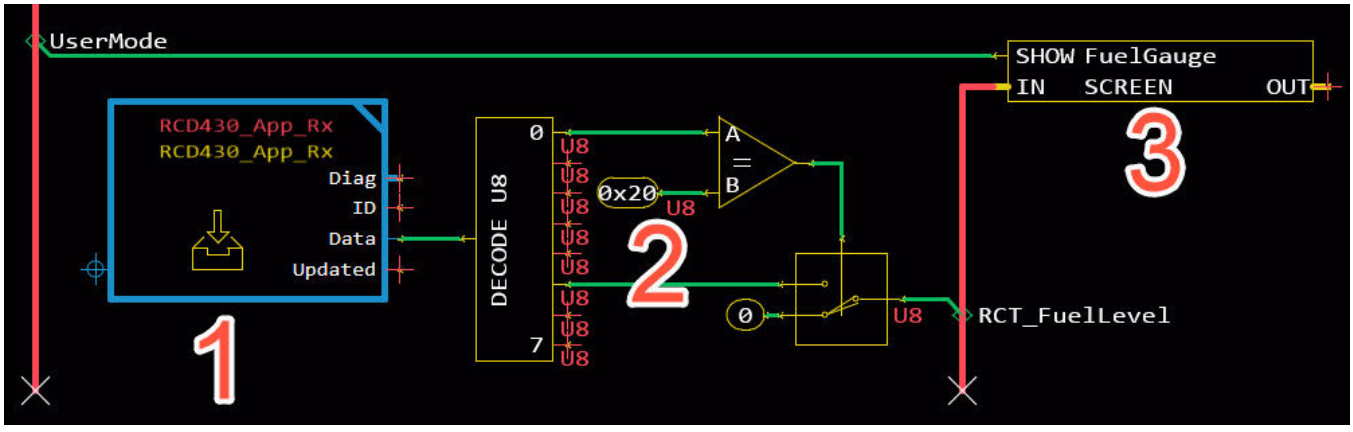
Configuration and Examples

- The **DisplayStatus1** signal from the **RCT_System_Rx** function block indicates that the RCD430 Display is busy processing a message. An **On Delay** component delays a True signal by 100 ms. This delays transmission of the message 100 ms when the RCD430 Display is busy processing a message.
- The **DisplayStatus2** signal from the **RCT_System_Rx** function block indicates that the buffer on the RCD430 Display is full. A **Not** component converts a True signal to a False signal. This prevents transmission of the message when the buffer is full.
- Both signals are combined using an **And2** component. The signal is True only when both signals into the **And2** component are True.

3. The **FuelLevelSend** signal transmits a True value once every 100ms per loop. This signal ensures that the fuel level value transmits at a regular interval without overloading the buffer on the RCD430 Display. The **FuelLevelSend** signal is combined with the **DisplayStatus** signal from item 2. When both are True, it commands the PDO_Tx function block to transmit the message.
4. The PDO_Tx function block (from the CANopen FB Library) transmits the fuel level value to the RCD430 Display.

Receive Fuel Level on the RCD430 Application

This example shows the RCD430_App_Rx function block configured to receive the fuel level signal from the machine application then display it on a screen.



1. The RCD430_App_Rx function block receives the fuel status message.
2. The fuel level value is extracted from the **Data** output and the message ID is validated.
   - The **Decode 8** component splits the **Data** signal into 8 U8 signals.
   - Element 0 in the array is the message ID. That signal is connected to an **Equal** component. If the ID from the received message matches **0x20**, the **Equal** component outputs a True signal that controls a **Switch 2 (Switch Boolean Controlled)** component.
   - Element 5 in the array is the fuel level value. That signal is connected to the **Switch 2 (Switch Boolean Controlled)** component. When the Message ID is **0x20**, the switch is enabled forwarding the fuel level signal.
3. A **Show Screen** component is configured to show the fuel level value on a fuel gauge.
   - The **RCT_FuelLevel** signal is connected to the input bus for the **Show Screen** component.
   - The **UserMode** signal enables the **Show Screen** component when the RCD430 Display is enabled to display user-defined screens.
   - The Show Screen component calls the **FuelGauge** screen definition. This screen is configured to display the fuel status on a gauge.

     For more information on creating a screen definition, refer to the Screen Editors section of the PLUS+1 GUIDE Software Help content.

Configuration and Examples

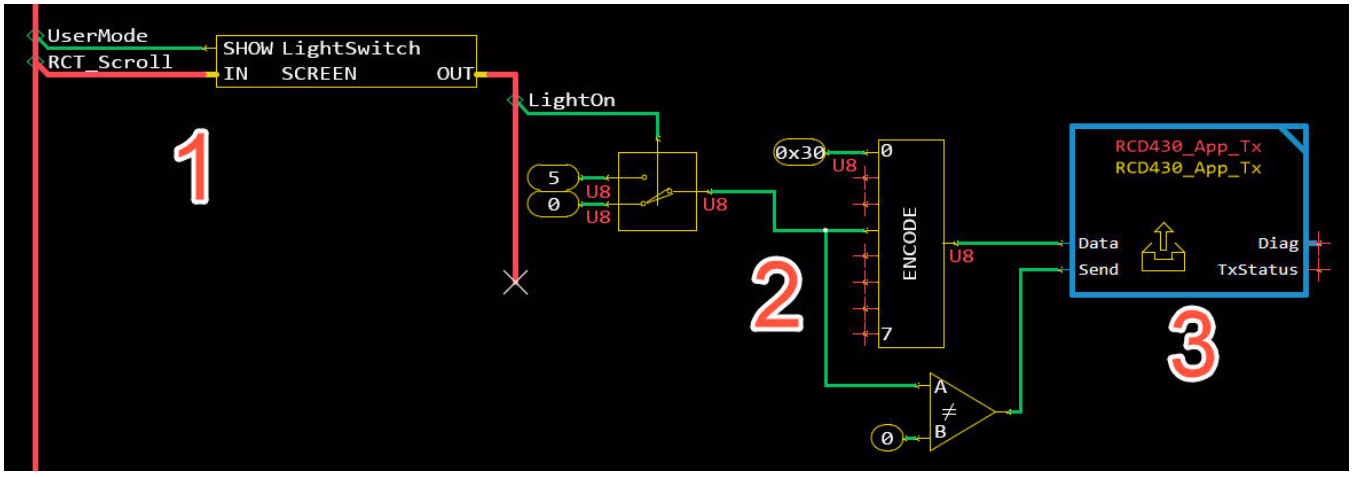
Light Switch Example

The examples in this section show how to configure the RCD430 Application to transmit a light-on command to the machine application. It also shows the machine application configured to receive the light-on command.

- *Transmit Light On Example* on page 13
- *Receive Light On Example* on page 14

Transmit Light On Example

This example shows the RCD430_App_TX configured to transmit a signal from a screen.



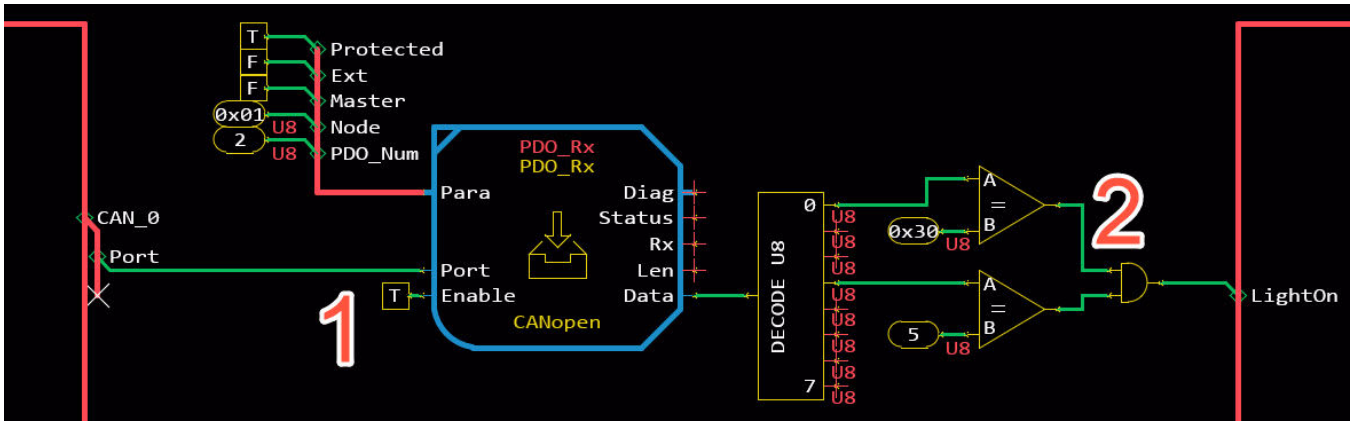
1. A **Show Screen** component calls a screen definition configured with a light switch.
   - The **UserMode** signal enables the **Show Screen** component when the RCD430 Display is in user mode.
   - The **RCT_Scroll** bus inputs signals from the scroll wheel on the side of the display.
   - The **Show Screen** component outputs a signal called **LightOn** that is controlled by the light switch on the screen.
2. The **LightOn** signal is encoded into an array then input into the RCD430_App_TX function block.
   - The **LightOn** signal controls a **Switch 2 (Switch Boolean Controlled)** component. When True, it forwards a value of 5.
   - An **Encode 8** places the value from the **LightOn** signal into element 3 in the array.
   - Element 0 of the array sets the ID of the message to **0x30**.
   - The **LightOn** signal is also connected to a **Not Equal** component with a 0 value. The output of the **Not Equal** component is connected to the **Send** input on the RCD430_App_TX function block.
3. The RCD430_App_TX function block is transmits the **LightOn** command to the machine application.
   - The **Data** input receives the **LightOn** message from the **Encode 8** component.
   - The Send input commands the function block to transmit the message when the **LightOn** value is not equal to 0.

As a result, when the light switch on the display is activated, the RCD430_App_TX function block transmits the message to the machine application to turn on the light.

Configuration and Examples

Receive Light On Example

The following example shows the PDO_Rx function block configured to receive the Light on command from the RCD430 Display.



1. The **PDO_Rx** function block receives the Light on signal from the transmitter example.
   - The parameters are configured to receive messages from the RCD430_App_TX function block.
   - The **Port** input is connected to the **Port** signal in the **CAN_0** bus.
   - **Enable** is set to True.
2. The Light On message is extracted from the **Data** output and the message ID is validated.

Configuration and Examples

- The **Data** output is connected to a **Decode 8** component.
- The 0 element in the array outputs the message ID. It is connected to an **Equal** that checks if it equals **0x30**.
- The 3 element in the array outputs the **LightOn** signal. An **Equal** component converts the light on signal from 5 to True.
- Both signals are combined with an **And2** component. When both are True, the **LightOn** signal is True.
- The **LightOn** signal is connected to the output bus and is used to control a light on the machine.

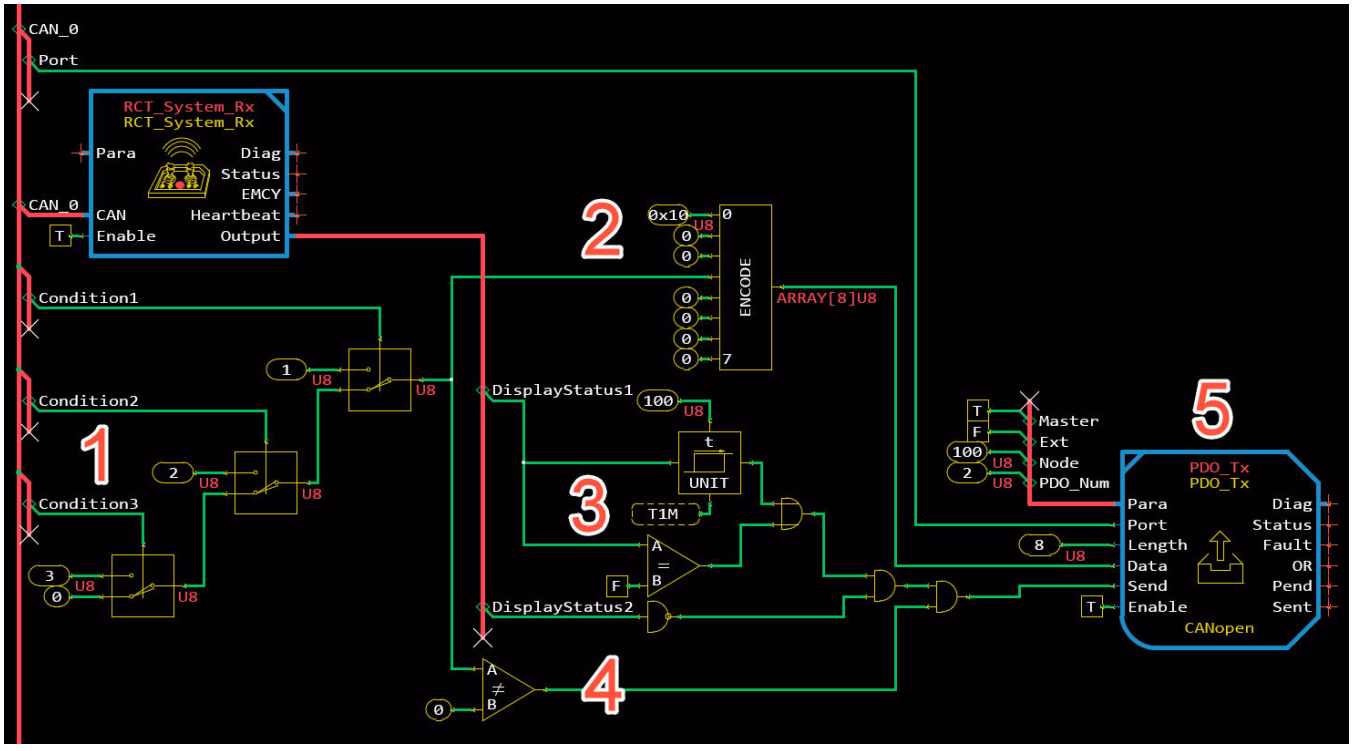
RCD430_App_Buzzer Function Block Example

This section contains examples using the RCD430_App_Buzzer function block to configure and activate the buzzer audible indicator on the RCT Device.

- *Transmit the Error Condition Number from the Machine* on page 15
- *Receive the Error Condition Number and Activate the Buzzer on the Display Application* on page 16

Transmit the Error Condition Number from the Machine

The following example shows the machine application configured to transmit an error condition number to the RCD430 Application. The RCD430 Application uses this error condition number to active the buzzer.



1. Boolean signals indicating the status of error conditions are prioritized and converted into numerical values. Condition 1 is the highest priority and condition 3 is the lowest priority. These conditions represent any error that the machine might encounter.

Configuration and Examples

- The **Condition3** signal connects to a **Switch2 (Switch Boolean Controlled)**. When True, it outputs a value of 3. When False, it outputs a value of 0.
- The **Condition2** signal connects to a **Switch2 (Switch Boolean Controlled)**. When True, it outputs a value of 2. When False, it outputs the value provided by the **Condition3** switch.
- The **Condition1** signal connects to a **Switch2 (Switch Boolean Controlled)**. When True, it outputs a value of 1. When False, it outputs the value provided by the **Condition2** switch.

2. The condition number is encoded into an Array[8]U8 signal with the message ID.
   - The condition number is set in element 3 of the array.
   - The message ID is set to **0x10** in element 0 of the array. This is used to identify the message when the display application receives it.
3. The RCT_System_Rx function block receives status information from the transmitter and the application uses that to queue messages to transmit.
   - The **DisplayStatus1** signal from the **RCT_System_Rx** function block indicates that the RCD430 Display is busy processing a message. An **On Delay** component delays a True signal by 100 ms. This delays transmission of the message 100 ms when the RCD430 Display is busy processing a message.
   - The **DisplayStatus2** signal from the **RCT_System_Rx** function block indicates that the buffer on the RCD430 Display is full. A **Not** component converts a True signal to a False signal. This prevents transmission of the message when the buffer is full.
   - Both signals are combined using an **And2** component. The signal is True only when both signals into the **And2** component are True.
4. When the condition number is also connected to a Not Equal component. When the condition number is not 0, it outputs a True signal. That signal is combined with the **DisplayStatus** signal using an **And2** component. When both are True, it commands the PDO_Tx function block to transmit.
5. The PDO_Tx function block is configured to transmit the condition number to the RCD430 Application.

Result: When there is an active condition and the transmitter buffer is not full or busy, the PDO_Tx function block transmits the condition number to the RCD430 Application.

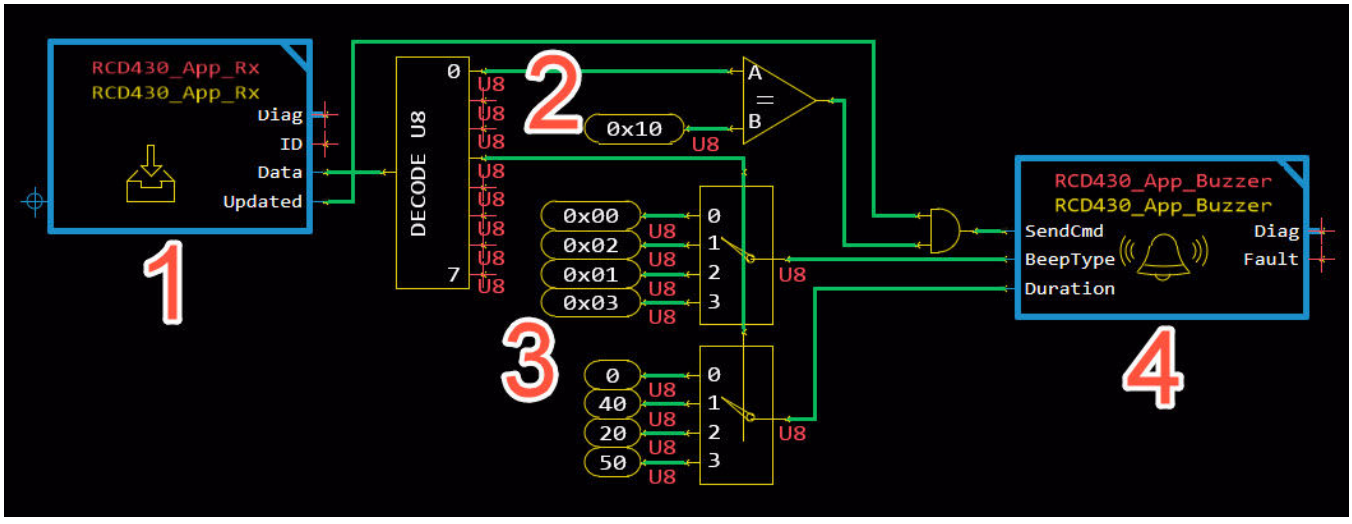
Receive the Error Condition Number and Activate the Buzzer on the Display Application

This example shows the RCD430_App_Buzzer function block configured to activate the buzzer based on error condition numbers transmitted from the machine application.

This example shows the RCD430_App_Buzzer function block switching between the following beep types based on the error condition numbers received:

- Condition 1: Slow repeating beeps for 4 seconds.
- Condition 2: Steady on beep for 2 seconds.
- Condition 3: Fast repeating beeps for 5 seconds.

Configuration and Examples



1. The RCD430_App_Rx function block receives the condition number (0-3) from the machine application. A **Decode 8** component extracts the elements from the **Data** output array.
2. An **Equal** component checks that the ID in element 0 in the array of the received message equals **0x10**. If the ID equals **0x10**, it forwards a True value. That signal is combined with the **Updated** output using an **And2** component. When both are True, it sends a True signal to the **SendCmd** input on the RCD430_App_Buzzer function block sending the buzzer command.
3. 2 **Switch 4** components are connected to element 3 on the Decode 8 component. The fist switch sets the **BeepType**. The second switch sets the **Duration**.
4. The RCD430_App_Buzzer function block receives the **BeepType** and **Duration** value. It activates the buzzer on the RCT Device when the ID matches **0x10** and **Updated** is True.

Inputs, Outputs, and Parameters

This section contains descriptions of inputs, outputs, and parameters in the function blocks. Refer to the following sections for more information.

- *RCD430_App_Core Function Block* on page 18
- *RCD430_App_Rx Function Block* on page 19
- *RCD430_App_TX Function Block* on page 19
- *RCD430_App_Buzzer Function Block* on page 20

RCD430_App_Core Function Block

The RCD430_App_Core function block contains predefined screen definitions and logic that enables the RCD430 Display in the RCD430 Application. This section describes the outputs in the RCD430_App_Core function block.

RCD430_App_Core Function Block Outputs on page 18

RCD430_App_Core Function Block Outputs

The following table describes outputs in the RCD430_App_Core function block.

Item	Type	Range	Description [Unit]
Main	BUS	——	Bus containing signals that report the state of the RCD430_App_Core.
RCD430_DisplayState	U16	1 to 19	Reports the current state of the RCD430_App_Core. 1: Initializing 2: Standby 3: Error 4: Receiving Files 5: Remote Configured 6: Remote Started 7: Latency 8: Menu 9: Configuration Error 10: API Committed 11: User Mode 12: Indetermination 13: Calibration 14: Limit Adjustment 15: Config Change Timeout 16: Loading File 17: Special Mode 18: Rx Enabled 19: User Mode Enabled
RCD430_UserMode	BOOL	T/F	Indicates that the current state of the RCD430_App_Core is User Mode and the user can display custom screens. T: User Mode is active and custom screens are enabled. F: Not in User Mode and custom screens are unable to be shown.

Inputs, Outputs, and Parameters

RCD430_App_Rx Function Block

The RCD430_App_Rx function block receives CAN messages from the machine application. This section contains descriptions of input and output signals for the function block.

- *RCD430_App_Rx Function Block Inputs* on page 19
- *RCD430_App_Rx Function Block Outputs* on page 19

RCD430_App_Rx Function Block Inputs

The following table describes input signals in the RCD430_App_Rx function block.

Item	Type	Range	Description [Unit]
Chkpt	BOOL	T/F	Enables Advanced Checkpoints with Namespace for each **Diag** signal. T: Include checkpoints when compiled. F: Do not include checkpoints when compiled.

RCD430_App_Rx Function Block Outputs

The following table describes output signals in the RCD430_App_Rx function block.

Item	Type	Range	Description [Unit]
Diag	BUS	——	Bus that provides diagnostic values for troubleshooting. In addition, it contains values for all inputs, parameters, and output signals.
ID	U16	0x201 to 0x27F 0x301 to 0x37F 0x401 to 0x47F	Reports the CANopen ID of the received CAN message. The CANopen ID is a combination of the PDO number and Node ID set on the transmitted message.
Data	(ARRAY[8]U8)	0 to 255	Outputs the data received in the CAN message.
Updated	BOOL	T/F	Indicates that new data is available in the current program loop. T: Received new data. F: Did not receive new data.

RCD430_App_TX Function Block

The RCD430_App_TX function block transmits CAN messages to the receiving machine application. This section contains descriptions of input and output signals for the function block.

- *RCD430_App_TX Function Block Inputs* on page 19
- *RCD430_App_TX Function Block Outputs* on page 20

RCD430_App_TX Function Block Inputs

The following table describes input signals in the RCD430_App_TX function block.

Item	Type	Range	Description [Unit]
Chkpt	BOOL	T/F	Enables Advanced Checkpoints with Namespace for each **Diag** signal. T: Include checkpoints when compiled. F: Do not include checkpoints when compiled.
Data	(ARRAY[8]U8)	0 to 255	Sets the data to be transmitted.
Send	BOOL	T/F	Transmits the message. T: Transmits the message. F: Does not transmit the message.

Inputs, Outputs, and Parameters

RCD430_App_TX Function Block Outputs

The following table describes output signals in the RCD430_App_TX function block.

Item	Type	Range	Description [Unit]
Diag	BUS	——	Bus that provides diagnostic values for troubleshooting. In addition, it contains values for all inputs, parameters, and output signals.
TxStatus	S8	0 to 3	Reports the status of the buffer. 0: Buffer is empty. 1: Buffer is processing a message. 2: Reserved 3: Buffer is full.

RCD430_App_Buzzer Function Block

The RCD430_App_Buzzer function block configures and activates the buzzer on the RCT Device. This section contains descriptions of input and output signals for the function block.

- *RCD430_App_Buzzer Function Block Inputs* on page 20
- *RCD430_App_Buzzer Function Block Outputs* on page 20

RCD430_App_Buzzer Function Block Inputs

The following table describes input signals in the RCD430_App_Buzzer function block.

Item	Type	Range	Description [Unit]
SendCmd	BOOL	T/F	A transition from False to True transmits the buzzer command. T: Transmits the buzzer command. F: No action.
BeepType	U8	0x00 to 0x03	Sets the functionality of the buzzer when activated. 0x00: Buzzer disabled. 0x01: Steady on. 0x02: Slow repeating beeps. 0x03: Fast repeating beeps.
Duration	U8	0 to 255	Sets the length in time of the buzzer action. [100 ms]
Chkpt	BOOL	T/F	Enables Advanced Checkpoints with Namespace for each **Diag** signal. T: Include checkpoints when compiled. F: Do not include checkpoints when compiled.

RCD430_App_Buzzer Function Block Outputs

The following table describes output signals in the RCD430_App_Buzzer function block.

Item	Type	Range	Description [Unit]
Diag	BUS	——	Bus that provides diagnostic values for troubleshooting. In addition, it contains values for all inputs, parameters, and output signals.
Fault	U16	——	Reports runtime errors encountered by the function block. It is bitwise code that can report multiple items. 0x0000: No fault. 0x8002: Input value too high.

ENGINEERING
TOMORROW

Products we offer:

- Cartridge valves
- DCV directional control valves
- Electric converters
- Electric machines
- Electric motors
- Gear motors
- Gear pumps
- Hydraulic integrated circuits (HICs)
- Hydrostatic motors
- Hydrostatic pumps
- Orbital motors
- PLUS+1® controllers
- PLUS+1® displays
- PLUS+1® joysticks and pedals
- PLUS+1® operator interfaces
- PLUS+1® sensors
- PLUS+1® software
- PLUS+1® software services, support and training
- Position controls and sensors
- PVG proportional valves
- Steering components and systems
- Telematics

Hydro-Gear
www.hydro-gear.com

Daikin-Sauer-Danfoss
www.daikin-sauer-danfoss.com

Danfoss Power Solutions is a global manufacturer and supplier of high-quality hydraulic and electric components. We specialize in providing state-of-the-art technology and solutions that excel in the harsh operating conditions of the mobile off-highway market as well as the marine sector. Building on our extensive applications expertise, we work closely with you to ensure exceptional performance for a broad range of applications. We help you and other customers around the world speed up system development, reduce costs and bring vehicles and vessels to market faster.

Danfoss Power Solutions – your strongest partner in mobile hydraulics and mobile electrification.

Go to www.danfoss.com for further product information.

We offer you expert worldwide support for ensuring the best possible solutions for outstanding performance. And with an extensive network of Global Service Partners, we also provide you with comprehensive global service for all of our components.

Local address:

Danfoss
Power Solutions (US) Company
2800 East 13th Street
Ames, IA 50010, USA
Phone: +1 515 239 6000

Danfoss
Power Solutions GmbH & Co. OHG
Krokamp 35
D-24539 Neumünster, Germany
Phone: +49 4321 871 0

Danfoss
Power Solutions ApS
Nordborgvej 81
DK-6430 Nordborg, Denmark
Phone: +45 7488 2222

Danfoss
Power Solutions Trading (Shanghai) Co., Ltd.
Building #22, No. 1000 Jin Hai Rd
Jin Qiao, Pudong New District
Shanghai, China 201206
Phone: +86 21 2080 6201