Part Number D301709X012
October 2021

IEC 62591 Wireless Interface Instruction Manual (for ControlWave® Micro)

Remote Automation Solutions

EMERSON.

Device Safety Considerations

- ### Reading these Instructions
  Before operating the device, read these instructions carefully and understand their safety implications. In some situations, improperly using this device may result in damage or injury. Keep this manual in a convenient location for future reference. Note that these instructions may not cover all details or variations in equipment or cover every possible situation regarding installation, operation, or maintenance. Should problems arise that are not covered sufficiently in the text, immediately contact Customer Support for further information.

- ### Protecting Operating Processes
  A failure of this device – for whatever reason -- may leave an operating process without appropriate protection and could result in possible damage to property or injury to persons. To protect against this, you should review the need for additional backup equipment or provide alternate means of protection (such as alarm devices, output limiting, fail-safe valves, relief valves, emergency shutoffs, emergency switches, etc.). Contact Remote Automation Solutions for additional information.

- ### Returning Equipment
  If you need to return any equipment to Remote Automation Solutions, it is your responsibility to ensure that the equipment has been cleaned to safe levels, as defined and/or determined by applicable federal, state and/or local law regulations or codes. You also agree to indemnify Remote Automation Solutions and hold Remote Automation Solutions harmless from any liability or damage which Remote Automation Solutions may incur or suffer due to your failure to ensure device cleanliness.

- ### Grounding Equipment
  Ground metal enclosures and exposed metal parts of electrical instruments in accordance with OSHA rules and regulations as specified in *Design Safety Standards for Electrical Systems*, 29 CFR, Part 1910, Subpart S, dated: April 16, 1981 (OSHA rulings are in agreement with the National Electrical Code). You must also ground mechanical or pneumatic instruments that include electrically operated devices such as lights, switches, relays, alarms, or chart drives.
  Important: Complying with the codes and regulations of authorities having jurisdiction is essential to ensuring personnel safety. The guidelines and recommendations in this manual are intended to meet or exceed applicable codes and regulations. If differences occur between this manual and the codes and regulations of authorities having jurisdiction, those codes and regulations must take precedence.

- ### Protecting from Electrostatic Discharge (ESD)
  This device contains sensitive electronic components which be damaged by exposure to an ESD voltage. Depending on the magnitude and duration of the ESD, it can result in erratic operation or complete failure of the equipment. Ensure that you correctly care for and handle ESD-sensitive components.

System Training

A well-trained workforce is critical to the success of your operation. Knowing how to correctly install, configure, program, calibrate, and trouble-shoot your Emerson equipment provides your engineers and technicians with the skills and confidence to optimize your investment. Remote Automation Solutions offers a variety of ways for your personnel to acquire essential system expertise. Our full-time professional instructors can conduct classroom training at several of our corporate offices, at your site, or even at your regional Emerson office. You can also receive the same quality training via our live, interactive Emerson Virtual Classroom and save on travel costs. For our complete schedule and further information, contact the Remote Automation Solutions Training Department at 800-338-8158 or email us at *education@emerson.com*.

Ethernet Connectivity

This automation device is intended to be used in an Ethernet network which **does not** have public access. The inclusion of this device in a publicly accessible Ethernet-based network is **not recommended**.

System Training

A well-trained workforce is critical to the success of your operation. Knowing how to correctly install, configure, program, calibrate, and trouble-shoot your Emerson equipment provides your engineers and technicians with the skills and confidence to optimize your investment. Remote Automation Solutions offers a variety of ways for your personnel to acquire essential system expertise. Our full-time professional instructors can conduct classroom training at several of our corporate offices, at your site, or even at your regional Emerson office. You can also receive the same quality training via our live, interactive Emerson Virtual Classroom and save on travel costs. For our complete schedule and further information, contact the Remote Automation Solutions Training Department at 800-338-8158 or email us at *education@emerson.com*.

Contents

Chapter 4 – Troubleshooting 4-1

Chapter 1 – General Information

In This Chapter

1.1 Overview

This manual covers both the hardware – the IEC 62591 Wireless Interface module for the ControlWave Micro device and the Emerson™ Wireless 781S Smart Antenna ("781S") – and the software you need to configure and commission the hardware components.

Note: The IEC 62591 Wireless Interface uses open source software. Refer to *Open Source Software Listing* (included in the same .zip file as this manual) for a complete listing of all components. Source code is available upon request by contacting Remote Automation Solutions Technical Support.

This chapter details the structure of this manual and provides an overview of the IEC 62591 Wireless Interface and its components.

The International Electrotechnical Commission's 62591 standard (commonly called *Wireless*HART®) is a global IEC-approved standard that specifies an interoperable self-organizing mesh technology in which field devices form wireless networks that dynamically mitigate obstacles in the process environment. This architecture creates a cost-effective automation alternative that does not require wiring and other supporting infrastructure.

Remote Automation Solutions' IEC 62591 implementation consists of an IEC 62591 Wireless Interface module installed in a ControlWave Micro RTU. The module is wired to a field-installed 781S. The wiring powers the 781S and transmits data between the 781S and a number of field-installed *Wireless*HART devices. The ControlWave implementation supports up to 100 devices. (See *Figure 1-1,* which shows a ControlWave Micro, a 781S, and several *Wireless*HART devices.)

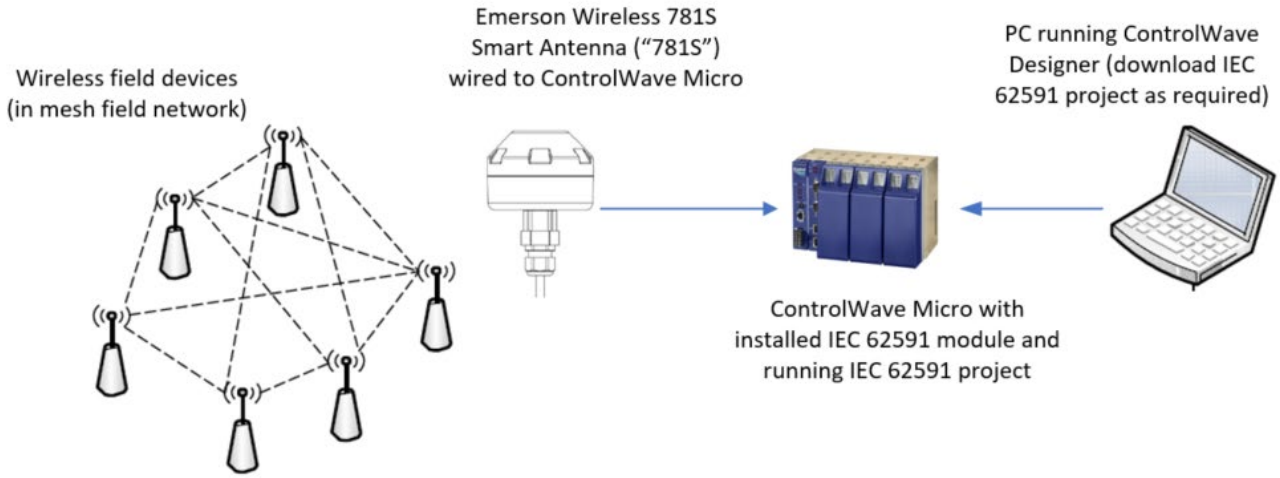


Figure 1-1. IEC 62591 Field Installation

1.2 Scope of Manual

This manual contains the following chapters:

Chapter 1 General Information	Provides an overview of the hardware for the IEC 62591 Wireless Interface.
Chapter 2 Installation	Provides information on installing the IEC 62591 Wireless Interface modules, installing the 781S, and wiring the 781S to the module.
Chapter 3 Configuring and Commissioning	Provides information on using ControlWave Designer to configure and commission the Wireless Interface.
Chapter 4 Troubleshooting	Provides general troubleshooting tips.

1.3 Hardware

The IEC 62591 Wireless Interface has two basic components: the IEC 62591 Wireless Interface module ("module") and the 781S.

1.3.1 IEC 62591 Wireless Interface Module

The IEC 62591 module has the standard appearance of a ControlWave Micro module (see *Figure 1-2*).

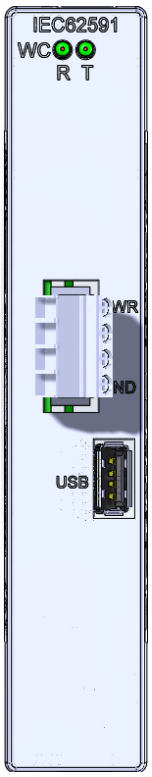


Figure 1-2. IEC 62591 Wireless Interface Module

You can place the module in any open I/O slot on the ControlWave Micro **except the last slot of any housing** (due to a mechanical restriction). Each ControlWave Micro can support only **one** IEC 62591 Wireless Interface module.

You **cannot** use the IEC 62591 with the ControlWave Micro Distributed I/O System.

Note: For information on installing modules in the ControlWave Micro, refer to the *ControlWave Micro Instruction Manual* (part D301392X012).

USB Port The module's USB port supports firmware upgrades and provides debug information for product support. For further information, refer to *Chapter 3, Configuration and Commissioning*.

1.3.2 Emerson Wireless 781 Smart Antenna

The second component in the Wireless Interface is the Wireless 781S Smart Antenna (see *Figure 1-3*). You install the 781S away from the controller in the optimal location for best network performance. A 4-wire connection between the module and the 781S provides the 12-30 Vdc power the 781S requires and transmits communication signals sent to the 781S from the various *Wireless*HART field devices.

Figure 1-3. Emerson Wireless 781S Smart Antenna

For instructions on installing the 781S, refer to *Chapter 2, Installation*.

1.3.3 *Wireless*HART Field Devices

The two components of Remote Automation Solutions' IEC 62591 Wireless Interface provide you with the ability to manage data from a network of *Wireless*HART field devices. Remote Automation Solutions supports transmitters that conform to the *Wireless*HART protocol. The physical configuration of the IEC 62591 Wireless Interface is based on the ControlWave Micro and the total number of field devices. A ControlWave Micro implementation supports up to 100 devices.

Note: The IEC 62591 Wireless Interface Module is designed to return the process and dynamic variables (PV, SV, TV, QV, slot 0, 1, 2, 3) from any device which meets the IEC 62591 specification (HCF_SPEC-285, Revisions 2.0). For a current list of the transmitters Remote Automation Solutions has tested with the IEC 62591 Interface, refer to the product data sheet *ControlWave Micro IEC 62591* (D301714X012) (available at *www.EmersonProcess.com/RemoteAutomation*). If you have a *Wireless*HART device which does not appear in the product data sheet. consult with the manufacturer of the device to determine whether the process variable values you want to collect are available through the PV, SV, TV, QV and slot 0, 1, 2, and 3. If the device meets the discrete control specification, it should work with the IEC 62591 Wireless Interface; alternatively, it may be treated like an analog wireless device. Always test any *Wireless*HART devices not listed to see whether they work with the IEC 62591 Wireless Interface **before** you install them in the field.

1.4 Configuration/Commissioning Software

Once you have installed the modules and wired them to the 781S, use ControlWave Designer, the AMS Device Configurator, or a hand-held HART configurator (such as the 375 or 475 Field Communicator or the AMS Trex™ Device Communicator) to configure and then commission ("activate") the entire network.

To directly configure newer transmitters (such as the Rosemount 928 transmitter), use a hand-held field communicator, the AMS Device Manager, the AMS Wireless Configurator, or any *Wireless*HART communicator. For remote configuration, use the AMS Device Manager, the AMS Wireless Configurator, or the Smart Wireless Gateway.

Refer to *Chapter 3, Configuring and Commissioning*, for specific instructions.

1.5 Additional Technical Information

Refer to the following technical documentation (available at *www.Emerson.com*) for additional technical and most-current information:

Table 1-1. Additional Technical Information

Name	Part Number
ControlWave Micro IEC 62591 Interface	D301714X012
Field Tools Quick Start Guide	D301703X412

Chapter 2 – Installation

In This Chapter

This chapter describes installing the IEC 62591 module in a ControlWave Micro, installing the Emerson Wireless 781S Smart Antenna ("781S"), and connecting the 781S to the IEC 62591 Wireless Interface module.

Note: This chapter covers the physical installation process. To configure and commission the IEC 62591 Wireless Interface, refer to *Chapter 3, Configuring and Commissioning*.

2.1 Installing the IEC 62591 Module

You install the IEC 62591 Wireless Interface module in the ControlWave Micro as you would any other module. However, you can install only one IEC 62591 module.

Notes:

- Only ControlWave Micro's with 05.50 (or newer) system firmware support the IEC 62591 module. Version 5.70 (or newer) system firmware is required for discrete control.
- Version 05.80 (or newer) system firmware is required with IEC Module Firmware V1.20 or later to support new Burst and Event parameters.
- You can install the IEC 62591 module in any I/O slot (slot 3 or higher **except** for the last slot in any chassis/housing) in the ControlWave Micro. You **cannot** use the IEC 62591 with the ControlWave Micro Distributed I/O System.

1. Disconnect power from the ControlWave Micro.
2. Choose an open I/O slot (except for the last slot of any housing).
3. Locate the built-in guides on the top and bottom of the housing.

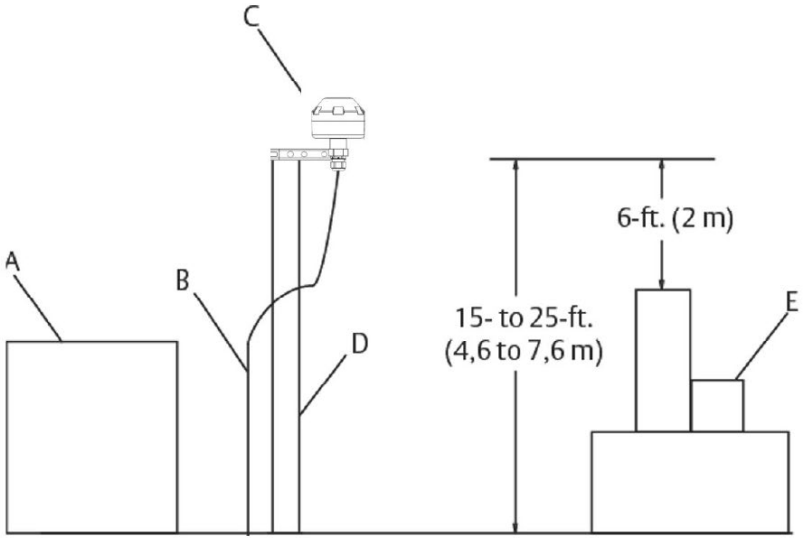
4. Gently slide the IEC 62591 module into the housing until it seats into the connectors on the interior back of the housing.
5. Restore power to the unit.

2.2 Installing the Wireless 781S Smart Antenna

This section covers where and how to install the 781S.

2.2.1 Optimizing the Location

Mount the 781S in a location that provides convenient access to the host system network (wireless I/O devices) and the network of wireless field devices. Find a location where the 781S has optimal wireless performance. Ideally, this is 4.6 to 7.6 m (15-25 ft) above the ground or 2 m (6 ft) above obstructions or major infrastructures. See *Figure 2-1*.



A Control room
B RS-485 cable
C Emerson Wireless 781S Smart Antenna
D Mast or pipe
E Infrastructure

Figure 2-1. Mounting the 781S

2.2.2 Positioning the 781S

Position the 781S vertically approximately 3 ft. (1 m) from any large structure, building, or conductive surfaces to allow clear communication with other devices. If you are installing multiple antennas, ensure that each antenna has at least 3 feet of horizontal separation from any other. See *Figure 2-1*.

2.2.3 Mounting the 781S

You typically mount the 781S on a pipe or mast using the clamps provided in the kit (see *Figure 2-2*).

1. Insert the U-bolt around a 2-in. pipe or mast, through the saddle, through the L-shaped bracket, and through the washer plate.
2. Use a ½-in. socket-head wrench to fasten the nuts to the U-bolt.
3. Secure the antenna to the L-shaped bracket with a 5/16-in. threaded bolt.
4. Use a 5/16 in. wrench to tighten the nuts to the housing.

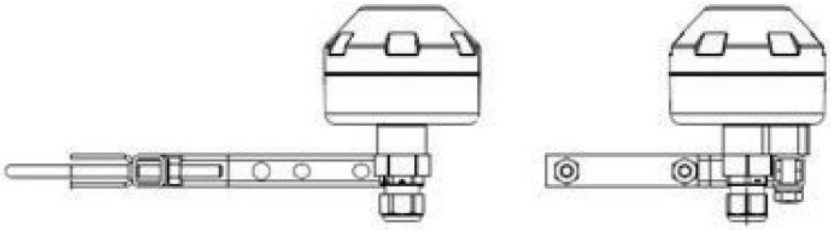


Figure 2-2. 781S Mounting

2.3 Wiring the Module and 781S

Note: Although its housing is permanently sealed, the 781S is prewired and only needs to be connected to the module. Ensure that wiring between the IEC 62591 module and the 781S meets all appropriate local requirements (use of conduit, etc.).

This section assumes you have already installed the IEC 62591 module in the ControlWave Micro **and** installed the 781S in its permanent field location.

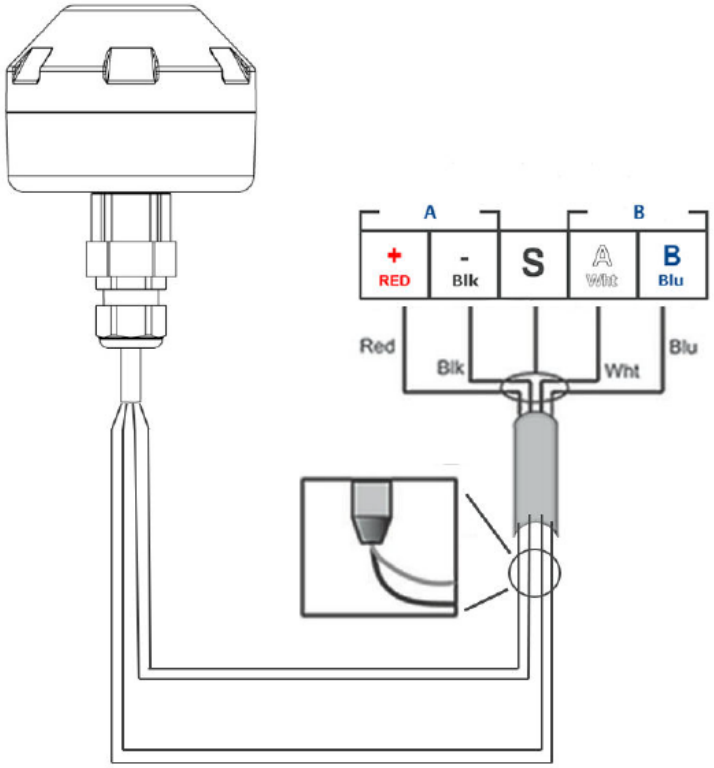
Communications between the IEC 62591 module and the 781S occur through an RS-485 connection. Remote Automation Solutions recommends that you use shielded, twisted-pair cable for I/O signal wiring. The twisted-pair minimizes signal errors caused by electro-magnetic interference (EMI), Radio Frequency Interference (RFI), and transients. The removable terminal blocks on the module accept wire sizes up to 14 AWG.

2.3.1 Wiring the 781S

1. Power down the IEC 62591 module (if it is currently powered).
2. Connect the positive power lead to the "+" power terminal and the negative power lead to the "–" power terminal.
3. Connect the data + lead to the "A (+)" terminal and the data – lead to the "B (–)" terminal (see *Figure 2-3*).

4. Connect the grounding wire to the modules and seal any unused conduit connectors.

If you are connecting multiple antennas, repeat this process for terminal connection 2.



A. Power output
B. RS-485 comm

Figure 2-3. 781S Power and Data Wiring

2.3.2 Wiring the IEC 62591 Module to the 781S

Note: Although its housing is permanently sealed, the 781S is prewired and only needs to be connected to the module. Examine the two leads coming out of the 781S and identify which are the communication (RS-485) leads (white and blue) and the power leads (red and black). Connect the 781S to the module as shown in *Figure 2-4.*

Figure 2-4 provides a wiring diagram for the IEC 62591 module. You must connect an RS-485 cable between the RS-485 port on the module and the 781S. This requires a Belden 2-twisted pair cable. *Table 2-1* shows connections for the twisted pair cable.

You must **also** connect termination jumper wires as shown in *Figure 2-5.*

Table 2-1. RS-485 Cable Connections

From 781S	IEC 62591 Module
A (+) (White/RS-485)	A
B (–) (Blue/RS-485)	B
POWER – (Black)	GND
POWER + (Red)	PWR

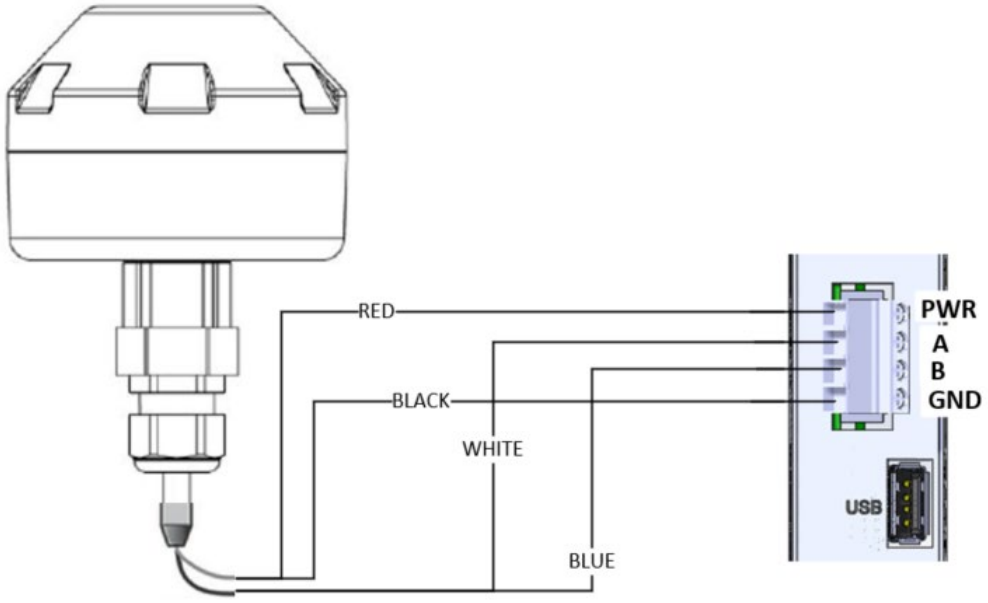


Figure 2-4. Power and Data Wiring from 781S to IEC 62591 Module

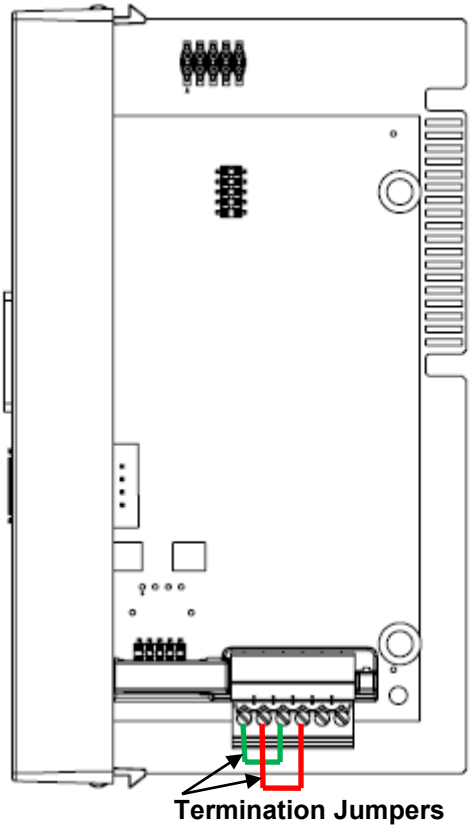


Figure 2-5. IEC 62591 Module – Connecting Termination Jumpers

2.4 Configuring Wireless Devices for the Network

See the product data sheet (available on *www.Emerson.com/RemoteAutomation*) for a list of devices Emerson has tested with the IEC 62591 Wireless Interface.

You must configure the following for each device in the wireless network:

- A long tag name (which is unique in the wireless network)
- A Network ID (must be the same for every device in the wireless network)
- A Join Key (must be same for every device in the wireless network)

⚠ Caution **Configure all devices belonging to a site to use the same Network ID and join key. To avoid network errors, configure all devices in adjacent networks to use a different Network ID and join key.**

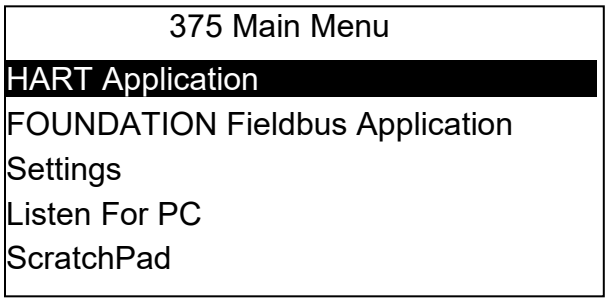
You configure these values using the 375 or 475 Field Communicator or the AMS Device Configurator. If you have AMS Device Configurator software, see its online help for details on how to do this.

Basic instructions for setting the long tag, Network ID and Join Key using the 375 Field Communicator are included below; see the *375*

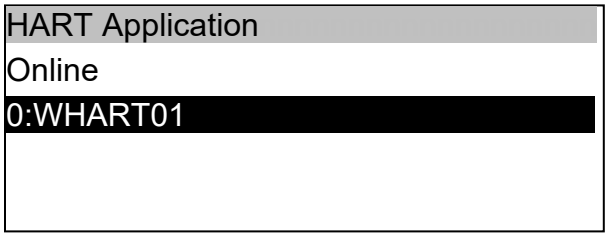
Field Communicator User's Manual (part 00375-0047-0001) if you need more information.

1. Connect the Field Communicator clips to the communication connectors on the wireless device, and turn the communicator on.

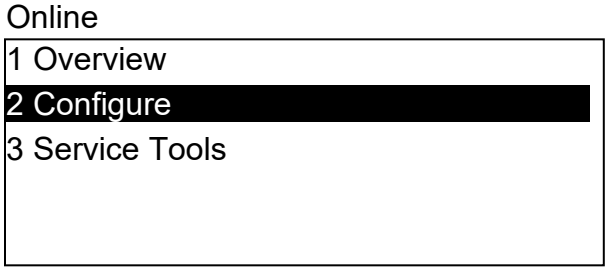
   From the Main Menu, double-tap **HART Application**.



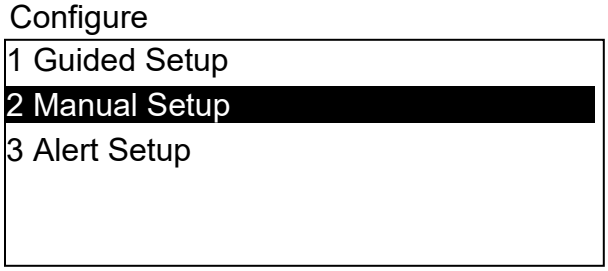
2. When the HART application detects the device you want to configure, double-tap on it.



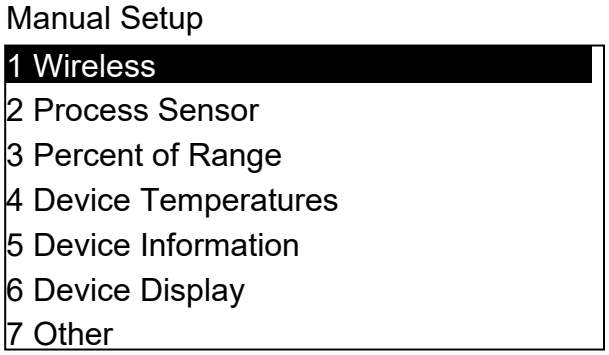
3. From the Online Menu, double-tap **Configure**.



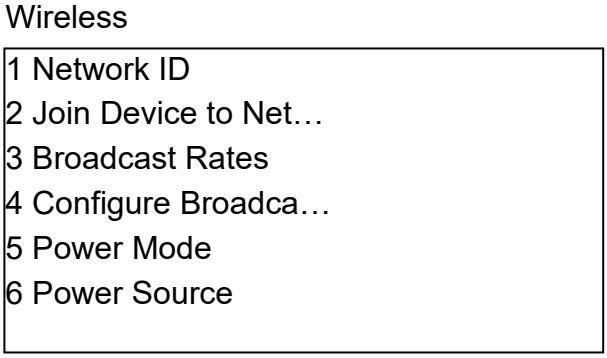
4. From the Configure Menu, double-tap **Manual Setup**.



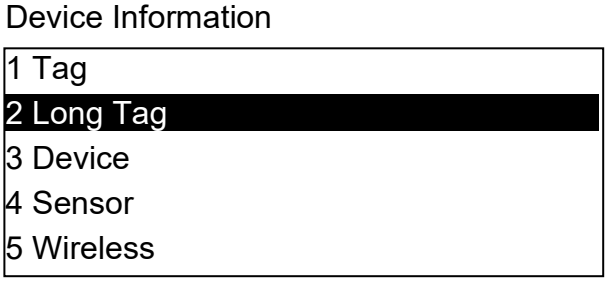
5. From the Manual Setup Menu, double-tap **Wireless**.



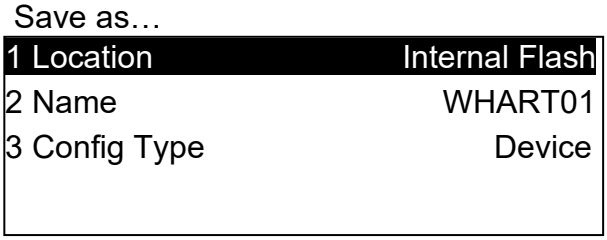
6. From the Wireless Menu, double-tap either:
   a. **Network ID** to set the Network ID using the on-screen keypad.
   b. **Join Device to Net…** to set the Join Key using the on-screen keypad.



7. From the Manual Setup Menu (shown in Step 7), double-tap **Device Information** to call up the Device Information menu, then choose **Long Tag** to set the long tag using the on-screen keypad.



8. When you're finished with your configuration, tap **SAVE**, and choose to save the changes to the Internal Flash memory of the device and tap **SAVE** again.



2.5 Preparing for Configuration and Commissioning

Once you have completed the wiring between the 781S and the ControlWave Micro, re-attach the plastic bezel covers and apply power to the ControlWave Micro.

Proceed to *Chapter 3*.

Chapter 3 – Configuration and Commissioning

In This Chapter

In addition to wiring the Emerson Wireless 781S Smart Antenna ("781S") to the IEC 62591 module and applying power to the module, you need to configure your ControlWave project to access the wireless network so it can discover and commission each *Wireless*HART device in the entire network. You also need to individually configure each *Wireless*HART device that will belong in the network with a unique long tag name and a common Network ID and common Join Key.

Note: Refer to the *AMS Device Manager Installation Guide* (part AW703M01V131EN) for instructions on using the AMS Device Configurator to configure the *Wireless*HART devices with the long tag name, Network ID, and Join Key.

Keep in mind that for **each** device configuration and commissioning is a two-step process:

- Configure each device using the AMS Device Configurator and a HART modem (or you can use a hand-held configuration device such as the Emerson 375 or 475 Field Communicator. During this step you individually add network information (Network ID, Join Key, and long tag name) to the field-based wireless device.

- Use ControlWave Designer to program the ControlWave Micro so it can detect the network and activate (or "commission") each device.

3.1 Overview

As indicated previously, a wireless interface network consists of wireless devices, a 781S, and an IEC 62591 module installed in a ControlWave Micro. The IEC 62591 module can communicate with a maximum of 100 wireless devices; the actual number allowed for a given application varies depending upon the burst rate. An 8-second burst rate allows the full number of 100 devices, a faster burst rate (such as 1-second) reduces the number of devices allowed.

The configuration and commissioning tasks described in this chapter use a PC running ControlWave Designer.

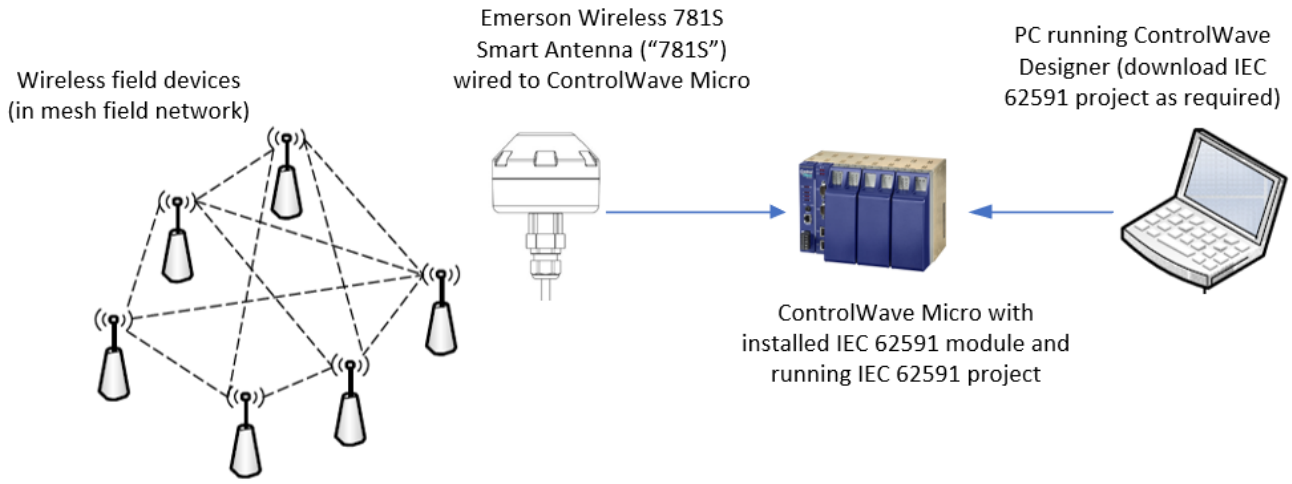


Figure 3-1. Wireless Interface

3.1.1 Configuring Devices and Planning the Network

Before you can use a *Wireless*HART device, you must first configure it. For this task you may use the AMS Device Configurator and a HART modem (or you can use a field communicator (such as Emerson's 375 or 475 Field Communicator. Ideally, you commission individual devices at a workbench in a protected environment, although you can field-commission a device you might add to the network. During the configuration, you identify the Network ID to which the device eventually belongs and provide the network-specific Join Key (see *Network ID and Join Key*).

During configuration, you must also give the wireless device a unique 32-character tag name based on its use or location (such as *PMP1TEMP*, *PMP2PRES*, or *WEL02LVL*).

Notes:

- Tag names cannot exceed 32 characters, and the tag name must be unique for this wireless network.
- You must configure the tag name as a **long tag name**. All references to tag names in this manual refer to the long tag name.
- Use upper-case (capital) letters for tag names; this corresponds to how the program internally stores tag names.

The individual devices should fit into a general organizational plan for your fields. By identifying logical groups and pre-assigning devices to those groups, you can eliminate guesswork during commissioning, efficiently define networks, and more quickly begin to acquire data.

Note: An important restriction in planning networks is to know that **a** network can have only **one** Network ID, **one** Join Key, **one** 781S, and **one** controller (a ControlWave Micro supporting up to 100 devices).

3.1.2 Network ID and Join Key

A 5-character Network ID defines one logical grouping of *Wireless*HART devices, all of which send their information to one 781S. (You define a device's Network ID when you first configure the device.) Valid values for the Network ID range from 1 to 36863.

Note: A Network ID or Join Key **cannot** be all zeros (such as 00000).

The Join Key is the password that allows a device to access its defined network. During configuration, you also provide the device with its network-specific Join Key. When you configure your ControlWave project using ControlWave Designer, you will need to know the Network ID and Join Key for this network.

3.1.3 Rosemount THUM™ Adapter

Note: Each THUM adapter supports only one **wired** HART device.

Rosemount's THUM Adapter provides wireless connectivity to a wired HART device. If you have already commissioned a wired HART device into your network and want to connect it to a THUM adapter, you must first decommission the device, attach the THUM adapter, and then re-commission the device. For further information about THUM adapters, refer to:

- *Emerson™ Wireless 775 THUM™ Adapter Reference Manual*, 00809-0100-4075
- *Emerson™ Wireless 775 THUM™ Adapter Quick Installation Guide*, 00825-0100-4075

The Quick Installation Guide was packed in the box with the THUM; the Reference Manual is available through the Emerson website (*www.Emerson.com/Rosemount*).

3.1.4 Active List and Commission List

When the 781S detects a wireless device that has the correct Network ID and Join Key, the IEC 62591 program running in the ControlWave Micro stores information about that device in a structure called the Active List. The Active List represents the entire network of wireless devices from which the 781S could potentially collect process variable data.

The program compares tag names for devices in the Active List to another structure called the Commission List. If a device in the Active List has a matching tag name in the Commission List, and its associated decommission flag is FALSE, the program stores configuration and process variable data for the device in the Commission List. The device is then considered to be "commissioned" as an active device in the network.

3.2 IEC 62591 Sample Program(s)

The ControlWave Designer software includes sample IEC 62591 programs that you can refer to or modify when creating your own IEC 62591-capable application. Alternatively, you can copy all the POUs, structures, and variables from one of the sample IEC 62591 programs into an existing ControlWave Designer project.

The sample programs include the IEC 62591 function block, as well as structure and type definitions needed for the commission list, active list, and statistics list.

Note: For details on specific parameters in the IEC 62591 function block, refer to the ACCOL3 online help in ControlWave Designer.

3.2.1 Before You Begin

Some of the structure data types used in the IEC 62591 sample programs have changed over time to add functionality.

- The ControlWave Micro firmware version **5.80** includes modifications for burst and event functionality for newly released transmitters. To use features available in the 5.80 firmware or later, earlier ControlWave projects must replace the *COMMISSION_ARRAY* **or** the *COMMISSION_ARRAY_DISCRETE* and the *COMMISSION_POINT* **or** the *COMMISSION_POINT_DISCRETE* structure data types with the *COMMISSION_ARRAY_V1_20* and *COMMISSION_POINT_V1_20* structure data types, and then re-compile the projects. These burst and event data type definitions are available in the ACCOL3 online help for the IEC62591 function block in 5.80 or in the IEC62591_V1_20_Example.zwt sample project.
- ControlWave Micro firmware version **5.70** includes modifications for discrete control functionality. To use features available in the 5.70 firmware, earlier ControlWave projects must replace the *COMMISSION_ARRAY* and the *COMMISSION_POINT* structure data types with the *COMMISSION_ARRAY_DISCRETE* and *COMMISSION_POINT_DISCRETE* structure data types, and then re-compile the projects. These discrete data type definitions are available in the ACCOL3 online help for the IEC62591 function block in 5.70 or in the *IEC62591_DIO_Example.zwt* sample project.
- You do not need to modify your application if you are not upgrading the existing firmware (version **5.60** or earlier).

In addition to familiarity with ControlWave Designer programming, you will need to know the following information prior to modifying one of the sample programs.

- The Network ID for the wireless network
- The Join Key for the wireless network
- The unique long tag names assigned to each wireless device
- The slot number in the ControlWave Micro that holds the IEC 62591 module
- The maximum number of wireless devices in your wireless network. We recommend you specify a maximum number that includes room for growth of the network, in case you add a few more devices later.

3.2.2 Accessing the IEC 62591 Sample Program(s)

1. Start ControlWave Designer.
2. Click **File >Open Project/Unzip Project**
3. If not already there, navigate to the OpenBSI/Projects area and open one of the IEC62591 sample projects.

Table 3-1. IEC 62591 Sample Programs

Sample Program Name	Description
IEC62591_V1_20_Example.zwt	IEC62591_V1_20_Example.zwt; Additional support for Burst Msg0 and Burst Msg1 for newer transmitters with CW Micro Firmware 5.80 or newer and IEC module Firmware 1.20 or newer.
IEC62591_DIO_Example.zwt	Addition of support for discrete control (use with CW Micro firmware 5.70 or newer).
IEC62591_Example.zwt	Original version of IEC 62591 sample project (use with CW Micro firmware 5.60 or earlier).

Notes:

- The structures and names shown in the following sections are representative of the original IEC 62591_Example.zwt project; subsequent sample projects may have variations in names but the purpose of the basic structures and variables (commission list, active list, join key, etc.) remains the same.
- See the program comments in the sample projects for the most current information.

4. When you open the project, you can see the project tree. *Figure 3-2* shows the different worksheets in the project tree.

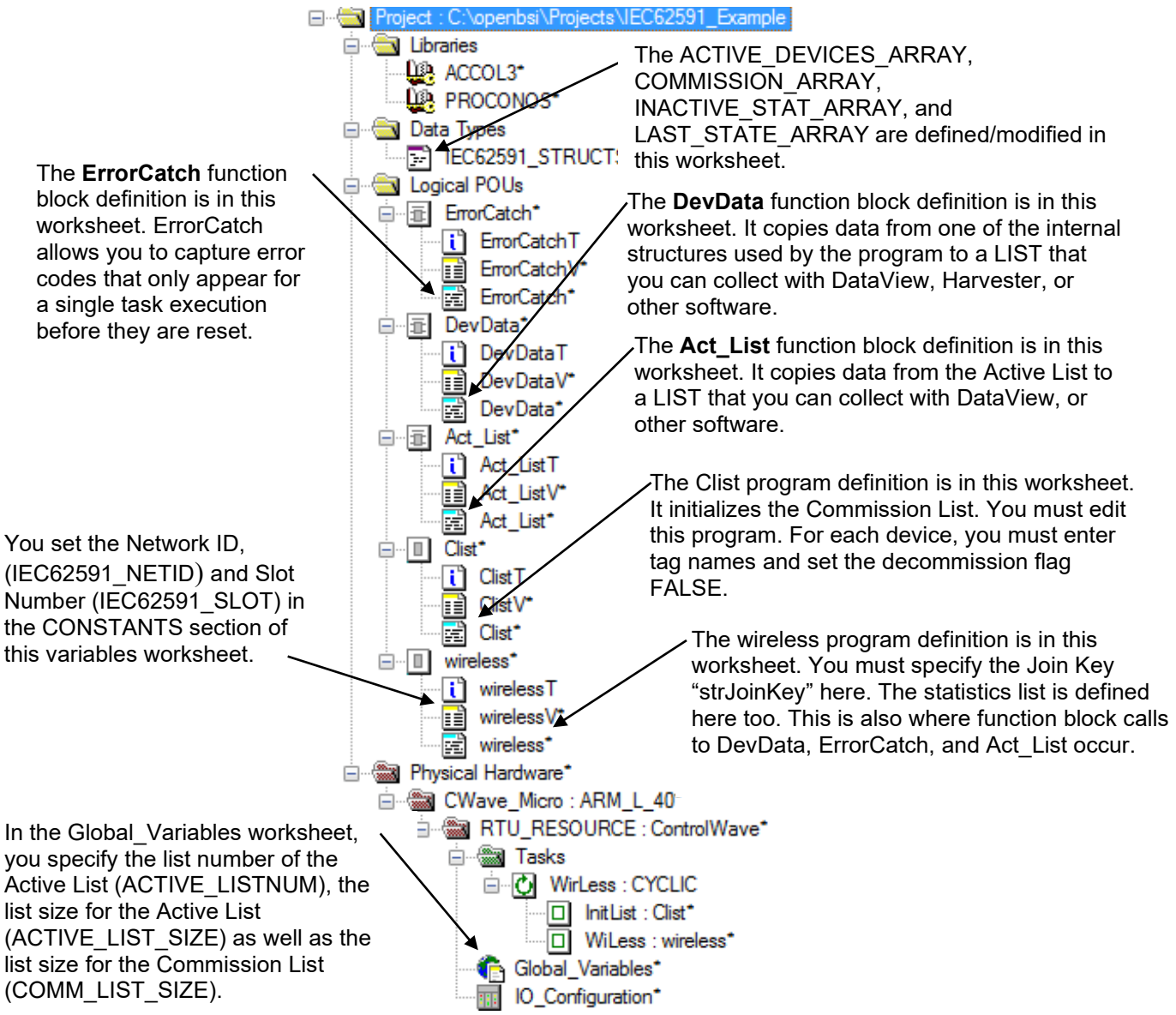


Figure 3-2. IEC 62591 Example – Project Tree

The next several sections outline how to modify the various worksheets in the project tree to meet the needs of your wireless network.

3.2.3 Modifying the IEC62591_STRUCTS Datatypes Worksheet

The IEC62591_STRUCTS datatypes worksheet defines the data types used by the program. Double-click the IEC62591_STRUCTS worksheet to open it.

⚠ Caution

Do not modify the datatypes themselves or your IEC 62591 program will not function correctly. You should define **only** the **sizes** of specific arrays (which may vary based on the number of devices in the wireless network.

3.2.3.1 ACTIVE_DEVICES_ARRAY_V1_20 datatype

The ACTIVE_DEVICES_ARRAY_V1_20 datatype defines an array of active points that make up the Active List structure for transmitters with adapter type and identification in the live list for firmware version 05.80.00 and later. The Active List maintains identification information on each wireless device discovered by the 781S.

Specify the array size to be equal to the maximum number of wireless devices you expect to exist in the network. This number should match the value of the ACTIVE_LIST_SIZE variable in the Global_Variables sheet.

```
TYPE
        ACTIVE_DEVICES_ARRAY_V1_20 : ARRAY [1..?] OF ACTIVE_POINT_V1_20;
END_TYPE
```

3.2.3.2 ACTIVE_DEVICES_ARRAY datatype

The ACTIVE_DEVICES_ARRAY datatype defines an array of active points that make up the Active List structure. The Active List maintains identification information on each wireless device discovered by the 781S.

Specify the array size to be equal to the maximum number of wireless devices you expect to exist in the network. This number should match the value of the ACTIVE_LIST_SIZE variable in the Global_Variables sheet.

```
TYPE
        ACTIVE_DEVICES_ARRAY : ARRAY [1..?] OF ACTIVE_POINT;
END_TYPE
```

3.2.3.3 COMMISSION_ARRAY_V1_20/COMMISSION_ARRAY_ DISCRETE/COMMISSION_ARRAY datatypes

These datatypes define an array of commission points that make up the Commission List structure. Which datatype you use depends on your application:

- *COMMISSION_ARRAY_V1_20* and *COMMISSION_POINT_ V1_20* datatypes are used in the IEC62591_V1_20_Example.zwt project. These structures provide for new transmitter controls (used in 5.80 or newer ControlWave Micro firmware).
- *COMMISSION_ARRAY_DISCRETE* and *COMMISSION_POINT_DISCRETE* datatypes are used in the IEC62591_DIO_Example.zwt project. These structures are required for discrete I/O control (used in 5.70 or newer ControlWave Micro firmware).
- The *COMMISSION_ARRAY* and *COMMISSION_POINT* datatypes are used in the original IEC62591_Example.zwt project (used in 5.60 or earlier ControlWave Micro firmware).

The Commission List holds a long tag name and a decommission flag for each device from which you intend to collect data in your network. For any wireless device in the Commission List that has a matching tag name to a device in the Active List (meaning it was detected by the

781S) *and* its decommission flag is FALSE, the Commission List also stores configuration information and actual process variable data.

Specify the commission array size to be equal to the maximum number of commissioned devices you expect to exist in the network. This number should match the value of the COM_LIST_SIZE variable in the Global_Variables sheet.

```
TYPE
        COMMISSION_ARRAY_V1_20:ARRAY [1..?] OF
        COMMISSION_POINT_V1_20;
END_TYPE

Or

TYPE
        COMMISSION_ARRAY_DISCRETE:ARRAY [1..?] OF
        COMMISSION_POINT_DISCRETE;
END_TYPE

Or

TYPE
        COMMISSION_ARRAY : ARRAY [1..?] OF COMMISSION_POINT;
END_TYPE
```

where **?** is the maximum expected number of devices (including future expansion) for your network.

3.2.3.4 INACTIVE_STAT_ARRAY and LAST_STATE_ARRAY datatypes

The INACTIVE_STAT_ARRAY and LAST_STATE_ARRAY are a pair of optional arrays that hold statistics about transmitter communication failures. Both these arrays (if used) should be sized to be equal to the maximum number of commissioned devices you expect to exist in the network. This number should match the value of the COM_LIST_SIZE variable in the Global_Variables sheet.

```
TYPE
        INACTIVE_STAT_ARRAY : ARRAY [1..?] OF INT;
END_TYPE

TYPE
        LAST_STATE_ARRAY : ARRAY [1..?] OF BOOL;
END_TYPE
```

3.2.4 ErrorCatch Function Block (*Optional*)

The ErrorCatch function block is called from within the Wireless worksheet. It maintains counts for different types of configuration errors and status conditions. There is no need to modify it. The ErrorCatch function block stores the error counts in the list specified by the ERROR_CATCH_LISTNUM variable in the Global_Variables worksheet.

3.2.5 Modifying the DevData Function Block (*Optional*)

The various structures in the IEC 62591 program are maintained internally; they cannot be collected by external utilities such as DataView. To support this sort of data collection, the IEC 62591 program does include a special function block (DevData) that copies

specific device data from the Commission array to a LIST structure that can be collected by an external program such as DataView.

As written in the original IEC62591_Example.zwt project, the DevData function block copies the following ten items from the COMMISSION_ARRAY.

Item	Description
TagName	The unique long tag name assigned to the device (up to 32 characters)
bActive	A flag which indicates communication with the device
PV	Primary process variable from the device (1st variable)
SV	Secondary process variable from the device (2nd variable)
TV	Tertiary process variable from the device (3rd variable)
QV	Quaternary process variable from the device (4th variable)
Slot0Var, Slot1Var, Slot2Var, Slot3Var	Wireless devices include a series of variables called “slots” which hold process data for the device such as temperature, pressure, scaling factors, altitude, flow, and so on. You can use the Slot*x*Var items to specify up to four of these slots for collection from the device. Consult the documentation for the particular wireless device for information on which slots hold which variables.

If you decide that you do not want all these items in the LIST structure, or you want to choose one or more different items from the list, you can modify the DevData function block definition.

To do this, double-click the DevData worksheet to open it.

For example, if you decided that you didn’t want the Slot3Var, and instead you wanted the serial number of the device you would do the following:

Change the **ianyElement10** entry in the LIST structure to reference SerialNum instead of slot3Var. (To see the exact names of the different items you can choose for entries in the LIST structure, click on the IEC62591_STRUCTS worksheet and review the COMMISSION_POINT_DISCRETE or COMMISSION_POINT structure definitions.)

```
LIST010_1(iiListNumber:=      inputListNum,
          ianyElement1:=      TagName,
          ianyElement2:=       bActive,
          ianyElement3:=      PV,
          ianyElement4:=      SV,
          ianyElement5:=        TV,
          ianyElement6:=        QV,
          ianyElement7:=        slot0Var,
          ianyElement8:=        slot1Var,
          ianyElement9:=        slot2Var,
          ianyElement10:=       SerialNum);
          listStatus:=LIST010_1.odiStatus;
```

Now change the DEMUX function call for the tenth item to reference the serial number parameter of the COMMISSION_POINT structure instead of the iorSlot3 parameter:

```
DEMUX_10(ianyInput := commissionList[inputDevNum].odiSerialNumber,
```

```
                    iiSelect  := 10,
                    iiOutlist := inputListNum);
demuxStat := DEMUX_10.odiStatus;
```

Now, instead of the slot3var value, the device serial number is copied.

Another possible modification you could make, if you wanted to copy more than ten items, would be to use a larger size LIST definition, say LIST30 or LIST100. Remember that if you do this, you must define the destination list to be the same type.

You may also choose to replace these LIST function blocks with the lists defined using ControlWave Designer's Variable Extension Wizard and its resulting _LIST.INI file.

3.2.6 Modifying the ACT_LIST Function Block (*Optional*)

The Active List is a list maintained by the IEC62591 function block of all wireless devices detected by the 781S. The purpose of the ACT_LIST function block is to copy the contents of the Active List to another LIST that can be collected by external software such as DataView.

Note: In the IEC62591_V1_20_Example.zwt and the IEC62591_DIO_Example.zwt projects, the ACT_LIST function block is called ActiveList.

The only thing you might modify for this is the LIST function block that is part of the ACT_LIST function block. By default, it uses a LIST020 function block which can have up to 20 elements – this allows a maximum of ten devices (defined by two elements for the tag name and device ID) in the active list. You can change this if you need to.

To do this, double-click the Act_List worksheet to open it.

```
LIST020_1(      iiListNumber  :=     inputListNum,
                ianyElement1  :=     Tag_1,
                ianyElement2  :=     DevId_1,
                ianyElement3  :=     Tag_2,
                ianyElement4  :=     DevId_2,
                ianyElement5  :=     Tag_3,
                ianyElement6  :=     DevId_3,
                ianyElement7  :=     Tag_4,
                ianyElement8  :=     DevId_4,
                ianyElement9  :=     Tag_5,
                ianyElement10 :=     DevId_5,
                ianyElement11 :=     Tag_6,
                ianyElement12 :=     DevId_6,
                ianyElement13 :=     Tag_7,
                ianyElement14 :=     DevId_7,
                ianyElement15 :=     Tag_8,
                ianyElement16 :=     DevId_8,
                ianyElement17 :=     Tag_9,
                ianyElement18 :=     DevId_9,
                ianyElement19 :=     Tag_10,
                ianyElement20 :=     DevId_10);
listStatus := LIST020_1.odiStatus;
```

Suppose, for example, that you needed to allow up to 50 devices in the active list?

ControlWave Designer supports LIST010, LIST020, LIST030, LIST050, and LIST100 function blocks. The easiest way to accommodate 50 devices is to replace the LIST020 function block with a single LIST100 function block. In the code, below, we don't show devices 11 through 45 for space reasons:

```
LIST100_1(          iiListNumber  :=    inputListNum,
                    ianyElement1  :=    Tag_1,
                    ianyElement2  :=    DevId_1,
                    ianyElement3  :=    Tag_2,
                    ianyElement4  :=    DevId_2,
                    ianyElement5  :=    Tag_3,
                    ianyElement6  :=    DevId_3,
                    ianyElement7  :=    Tag_4,
                    ianyElement8  :=    DevId_4,
                    ianyElement9  :=    Tag_5,
                    ianyElement10 :=    DevId_5,
                    ianyElement11 :=    Tag_6,
                    ianyElement12 :=    DevId_6,
                    ianyElement13 :=    Tag_7,
                    ianyElement14 :=    DevId_7,
                    ianyElement15 :=    Tag_8,
                    ianyElement16 :=    DevId_8,
                    ianyElement17 :=    Tag_9,
                    ianyElement18 :=    DevId_9,
                    ianyElement19 :=    Tag_10,
                    ianyElement20 :=    DevId_10
                    :
                    :
                    ianyElement91 :=    Tag_46,
                    ianyElement92 :=    DevId_46,
                    ianyElement93 :=    Tag_47,
                    ianyElement94 :=    DevId_47,
                    ianyElement95 :=    Tag_48,
                    ianyElement96 :=    DevId_48,
                    ianyElement97 :=    Tag_49,
                    ianyElement98 :=    DevId_49,
                    ianyElement99 :=    Tag_50,
                    ianyElement100 :=   DevId_50);
listStatus := LIST100_1.odiStatus;
```

To support the maximum number of devices (100), you chain two LIST100 function blocks together. To do this, use the same iiListNumber parameter for each; that connects the two to allow for a 200-element list.

3.2.7 Configuring the Commission List

When the 781S detects a wireless device that can be included in the network, it adds information about that device to the Active List. Process data from the device is collected **only** if it has a matching tag name in the Commission List **and** if its decommission flag is FALSE.

Notes:

- The Commission List structure in the IEC62591 V1 20_Example.zwt project uses the *COMMISSION_ARRAY_V1_20* datatype, the CommishList POU worksheet, and the CommissionList array (use with ControlWave Micro firmware 5.80 or newer).
- The Commission List structure in the IEC62591_DIO_Example.zwt project uses the *COMMISSION_ARRAY_DISCRETE* datatype, the

the CommishList POU worksheet, and the CommissionListDiscrete array (use with ControlWave Micro firmware 5.70 or newer).

- The Commission List structure in the IEC62591_Example.zwt project uses the *COMMISSION_ARRAY* datatype, the Clist POU worksheet, and the CommissionList array (use with ControlWave Micro firmware 5.60 or earlier)

You must create an entry in the Commission List that includes the long tag name for the device and decommission flag (set to FALSE) for **each** device you want the IEC 62591 module to access.

To do this, double-click the CommisList (or Clist) worksheet to open it, then add or modify Commission List definitions to include a valid long tag name for each iostrTagName and specify FALSE for the ibDecommission flag for each device you want to commission. Specify TRUE for the ibDecommission flag only if you don't want to commission the device (for example, it is not yet ready to be added to your system).

In the code below, devices DEV_0001, DEV_0002, DEV_0003 and DEV_0004 all have their ibDecommission flag set to FALSE, so they all can be commissioned and have process variable data collected over the network. DEV_0005 has a decommission flag set to TRUE, so it cannot be commissioned and its process data is not collected unless that decommission flag is changed to FALSE.

```
IF (bInitCommList = TRUE) THEN
        bInitCommList := FALSE;

commissionList[1].iostrTagName   := 'DEV_0001';       (* Tagname in wireless transmitter.  *)
commissionList[1].ibDecommission := FALSE; (* Commission the device when found. *)

commissionList[2].iostrTagName   := 'DEV_0002';
commissionList[2].ibDecommission := FALSE;

commissionList[3].iostrTagName   := 'DEV_0003';
commissionList[3].ibDecommission := FALSE;
c
commissionList[4].iostrTagName   := 'DEV_0004';
commissionList[4].ibDecommission := FALSE;

commissionList[5].iostrTagName   := 'DEV_0005';
commissionList[5].ibDecommission := TRUE;  (* Device is commissioned only after
                                            * this is changed to FALSE. *)

END_IF;
```

There are other parameters in the Commission List that populate automatically when the program sees a match between the long tag names in the Commission List, and the long tag names in the Active List – the iostrTag Name and ibDecommission flag are the only portions you create or modify.

If, for some reason, you want to temporarily prevent data collection from a device in your wireless network, you can just set that device's decommission flag to TRUE.

3.2.8 Specifying the Join Key

All wireless devices in your wireless network share the same Join Key.

You must also specify the same Join Key in your ControlWave project. You specify the Join Key in the Wireless worksheet.

Double-click on the Wireless worksheet and specify the Join Key using the strJoinKey variable.

```
strJoinKey := '12345678-00000000-00000000-00000000';
```

3.2.9 Specifying the Network ID and Slot Number

The wireless network of devices detected by the 781S has a Network ID.

You must specify the Network ID in the IEC62591_NETID variable in the WirelessV worksheet of your ControlWave project.

In the same worksheet you use the IEC62591_SLOT variable to specify the slot number of the ControlWave Micro controller that holds the IEC 62591 module. The IEC 62591 module can reside in any open I/O slot (slot 3 or higher) except for the last slot in the chassis.

Double-click on the WirelessV worksheet and specify the Network ID and slot number.

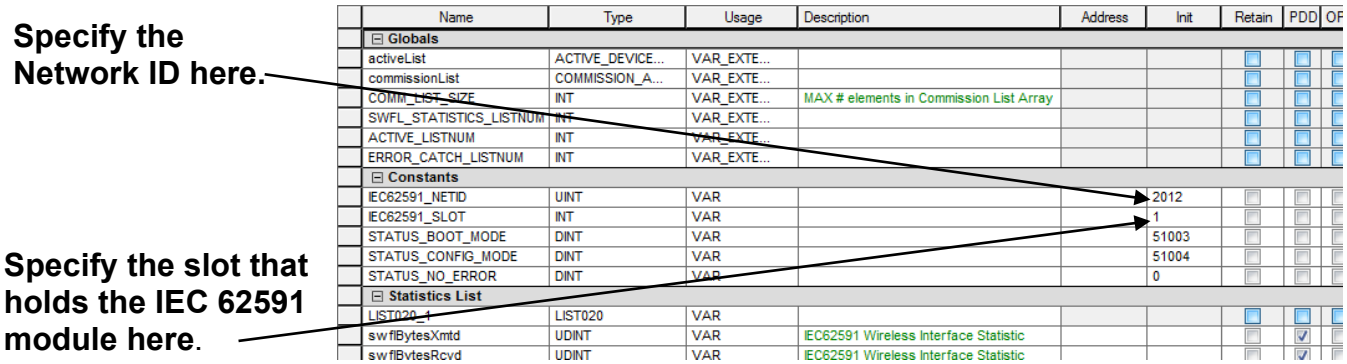


Figure 3-3. Editing the WirelessV Worksheet

3.2.10 Configuring the Statistics List (Optional)

The optional Statistics List maintains information on the health of IEC 62591 communications for debugging purposes.

- In the IEC62591_V1_20_Example.zwt and the IEC62591_DIO_Example.zwt projects, the Statistics list is defined in the StatsList worksheet.
- In the IEC62591_Example.zwt project, the Statistics list is defined in the Wireless worksheet.

```
LIST020_1(iiListNumber:= SWFL_STATISTICS_LISTNUM,
ianyElement1:=  swflBytesXmtd,          (* IEC62591 Interface # bytes transmitted    *)
ianyElement2:=  swflBytesRcvd,          (* IEC62591 Interface # bytes received       *)
ianyElement3:=  swflBytesDiscarded,     (* IEC62591 Interface # bytes discarded      *)
ianyElement4:=  swflMsgsXmtd,           (* IEC62591 Interface # messages transmitted *)
```

```
ianyElement5:=  swflMsgsRcvd,              (* IEC62591 Interface # messages received    *)
ianyElement6:=  swflNacksXmtd,             (* IEC62591 Interface # NAKs transmitted     *)
ianyElement7:=  swflNacksRcvd,             (* IEC62591 Interface # NAKs received        *)
ianyElement8:=  swflMsgRetriesRcvd,        (* IEC62591 Interface # of message retries   *)
ianyElement9:=  swflSessionInitRcvd,       (* IEC62591 Interface # session Initializations received *)
ianyElement10:= swflSessionRestartXmtd,    (* IEC62591 Interface # session restarts transmitted     *)
ianyElement11:= swflSetTimeMsgsXmtd,       (* IEC62591 Interface # Set Time messages transmitted    *)
ianyElement12:= swflSetTimeMsgsRcvd,       (* IEC62591 Interface # Set Time messages received       *)
ianyElement13:= swflResetApmMsgsXmtd,      (* IEC62591 Interface # reset APM messages transmitted   *)
ianyElement14:= swflResetApmMsgsRcvd,      (* IEC62591 Interface # reset APM transmitted            *)
ianyElement15:= swflTunnelMsgsXmtd,        (* IEC62591 Interface # Tunnel messages transmitted      *)
ianyElement16:= swflTunnelMsgsRcvd,        (* IEC62591 Interface # Tunnel messages received         *)
ianyElement17:= swflOtherHARTMsgsXmtd,     (* IEC62591 Interface # other HART messages transmitted
*)
ianyElement18:= swflOtherHARTMsgsRcvd,(* IEC62591 Interface # other HART messages received     *)
ianyElement19:= swflRadioMsgsXmtd,         (* IEC62591 Interface # Radio messages transmitted       *)
ianyElement20:= swflRadioMsgsRcvd);        (* IEC62591 Interface # Radio messages received          *)

        diStatisticsListStatus := LIST020_1.odiStatus;
```

The number of the Statistics List is set in the Global_Variables worksheet and using the SWFL_STATISTICS_LISTNUM variable.

3.2.11 Generating Alarms Based on IEC62591 Function Block Status (*Optional*)

The IEC 62591 function block includes an **odiStatus** parameter which reports error and status codes for the IEC 62591 wireless interface. The ControlWave Designer online help provides descriptions of what the error and status codes mean. You may choose to implement control logic to generate an alarm for certain odiStatus values.

For example, beginning with 1.01 IEC 62591 module firmware, odiStatus codes in the range -51120 to -51129 indicate the IEC 62591 Wireless Interface Module had to re-start and cannot provide live data updates until the re-start process is complete. You could add the code below to your ControlWave project to implement an alarm for that range of values:

```
(* Execute IEC62591 Function Block Instance.*)
IEC62591_1(…);
(* Save the FB processing status. *)
IEC62591Status   := IEC62591_1.odiStatus;
(* Set/Reset the alarm condition.  *)
bXMTRFrozen := (IEC62591Status <= DINT#-51120) AND
(IEC62591Status >= DINT#-51129);
(* Execute the ALARM FB when variable is TRUE/ON *)
ALARM_LOGICAL_ON_1
    ( :
iaAlarmVar:=bXMTRFrozen,
  :
 );
```

3.2.12 Additional Programming Notes

The IEC 62591 example program includes two required program type POUs. **InitList** initializes the Commission List and **WiLess** uses the Wireless worksheet definitions and includes the actual IEC62591 function block. Both these program POUs reside within a CYCLIC task called “WirLess.”

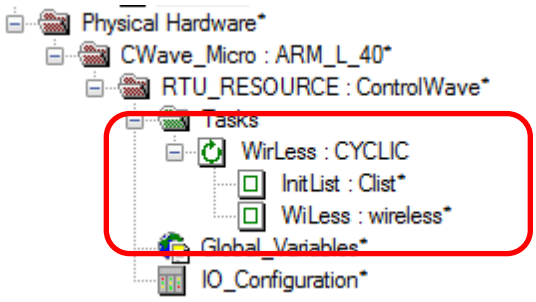


Figure 3-4. CYCLIC Task Running Wireless Program

Note: The **WirLess** task in the example program executes once per second to ensure processing of data from the wireless network. **Do not** change the task **Interval** setting from 1000 ms (1 second).

Once programming is complete, compile and build the ControlWave project and download it into the ControlWave Micro.

3.3 Commissioning Wireless Devices

Once you download the completed project with the configured IEC 62591 program to the ControlWave Micro, the project executes and commissioning begins automatically.

Note: The 781S detects **only** those wireless devices that you have already individually configured with a unique long tag name, a common Network ID, and common Join Key. You perform this configuration separately using the AMS Device Configurator and a HART modem or an Emerson 375/475 Field Communicator. For information on configuring these devices using the AMS Device Configurator, see the AMS Device Configurator documentation.

The 781S detects any configured wireless devices and reports them to the IEC 62591 module.

The executing program adds these devices to the Active List and checks for each device to see whether a matching device tag exists in the Commission List. Each device that has a matching tag along with an ibDecommission flag set to FALSE automatically becomes an active commissioned device in the network.

3.3.1 Active Advertising

In addition to the normal mode for detecting wireless devices, the system supports **active advertising.** In active advertising, the IEC 62591 module sends messages to the wireless network to keep radios active for a longer period of time to facilitate quicker detection of new (or replaced) wireless devices. Because leaving radios on consumes power, active advertising is only used under certain conditions:

- When the IEC 62591 module is first powered on, or is restarted after being powered off, it automatically remains in active advertising mode for a period of time to detect wireless devices.
- Whenever a new device is added to the network, active advertising is activated for a period of time in case additional devices are also added.
- Whenever a device leaves the network (becomes unreachable) active advertising is activated for a period of time to allow communications to be re-established.
- Beginning with ControlWave firmware version 5.70, the application programmer can force active advertising using the **ioaiMode** parameter in the IEC62591 function block. See the ACCOL3 online help in ControlWave Designer for details.

3.4 Decommissioning or Replacing Wireless Devices

If you want to temporarily remove a wireless device from the network, modify the ControlWave project to edit the Commission List and set the ibDecommission flag for that device to TRUE, then compile and download the revised project.

If you want to permanently remove a wireless device from the network, modify the ControlWave project to remove the device from the Commission List, then compile and download the revised project.

If you only want to replace a device, for example, because it failed and you want to put an identical device in the same location, there is no need to decommission the device; simply use the AMS Device Configurator or a hand-held configuration device to specify the same long tag name, join key, and network ID in the replacement device, then install the new device normally, and it will join the network in place of the failed device.

3.5 *Wireless*HART Data Access and Statistics

The IEC 62591 Wireless Interface collects both *Wireless*HART data and statistics.

3.5.1 *Wireless*HART Data Access

The IEC62591 function block is pre-configured to return the Universal and Common HART parameters including:

- Long Tag
- User Defined Message
- User Defined Descriptor
- Extended Device Type
- Device ID
- Manufacturer ID
- Device Serial Number

- Adapter Type – THUM's Expanded Device Type
- Adapter ID – THUM's Device ID
- PV, SV, TV and QV Variable Units
- Slot 0, 1, 2 and 3 Variable Units
- PV, SV, TV and QV Variable Value
- Slot 0, 1, 2 and 3 Variable Value
- Primary Variable Loop Current
- Device Status
- Battery Life (you may want to generate an alarm when this falls below a certain threshold; see *Section 3.2.11* for information on generating alarms.)
- PV Loop current
- Burst Rate

The following parameters require the discrete control version of the application and associated structures:

- Number of discrete channels
- Set class, Live Class, Set Point, and Live Value for each of up to four (4) discrete channels
- Failsafe mode (hold last value or use failsafe value)
- Failsafe value for PV, SV, TV, and QV
- PV validity flags – such as NAN (Not a Number)

The following parameters require ControlWave firmware 5.80 or newer to support the application and associated structures:

- Displays the Min Update Time
- Max Update Time
- Trigger Mode
- Trigger Level
- Device Variable Classification
- Unit Code
- HART Command Execution Status.

Notes:

- The Commission List structure holds these items. You use the DEV_DATA function block to access the parameters you need. See *Section 3.2.5* for more information.
- The application you create must handle the PV validity flags. For example, the IEC 62591 module reports the flag but does not set the PVs to NAN. It is up to the application to choose whether to force a NAN value, use the failsafe value, or use the last reported good value.
- Usage of the primary value (PV), secondary value (SV), tertiary value (TV) and quaternary value (QV) vary depending on the type

of wireless device. For some devices, the primary value might be battery voltage, whereas for a different device it might be the loop current.

3.5.2 *Wireless*HART Communication Statistics

Detailed communication statistics are accumulated for the wireless network and optionally stored in the Statistics List (see *Section 3.2.10*). Transmit and receive data is accumulated for byte, message, session, tunnel, radio and other HART messages.

3.6 Upgrading Module Firmware

If a new version of firmware is released for the IEC 62591 module, either to support new features or correct problems, you can install it in the field through the module's USB port.

⚠ Caution **Do not** use the USB port unless the area is known to be non-hazardous.

To do this, you must know the name of the variable associated with the **ioaiMode** parameter of the IEC62591 function block.

1. Create a folder named **upgrade** in the root of a USB drive.
2. Copy the firmware file to the upgrade folder.
3. Insert the USB drive into the USB port of the IEC 62591 Wireless Interface Module.
4. Using DataView or through ControlWave Designer in online operation, change the value of the **ioaiMode** parameter to **2** to start the firmware download.
5. Monitor the **odiStatus** parameter on the IEC62591 function block. While the download progresses, status code **51005** is reported. When the download completes the success code of **51006** appears momentarily; then **0** appears which means the firmware upgrade is complete. The upgrade takes approximately five minutes.

Chapter 4 – Troubleshooting

In This Chapter

This chapter includes some general troubleshooting guidelines, as well as tips on certain common problems and how to overcome them.

4.1 Troubleshooting Guidelines

Before you begin to trouble-shoot the interface, you should observe the following guidelines.

- **Don't overlook the obvious.** With all the activity involved in setting up a wireless network, it is easy to accidentally unplug an antenna or disconnect power from a device. Check those things *first*. (For a list of common problems, see the *Troubleshooting Checklist* at the end of this chapter.)
- **If something worked previously, but now has stopped working, did you change something?** If, for example, you change the application running in the ControlWave Micro and re-download it, and now it stopped working, it's possible that the change you made to the application might have caused a problem.
- **Adopt a systematic approach.** Don't try to solve the problem by changing several different things at once. Change one thing, see if it causes an improvement, then make notes about what you did, *then* you can try to make other changes. If you haphazardly begin swapping hardware modules, re-routing cables, and changing software parameters, you may end up in worse shape than when you started, or you may end up masking symptoms of an underlying problem.
- **Try to isolate the problem.** If, for example, you can communicate with *some* wireless devices but not others, then concentrate on what's different with the non-functional wireless devices, or their configuration parameters. If you can't communicate with any wireless devices, your might not have correctly configured network parameters in the application, or there may be a problem at the 781S.

- **Use the hardware and software diagnostic tools provided with the product.** The ControlWave Micro process automation controller includes status LEDs on both the CPU and PSSM modules; see the ControlWave Micro instruction manual for explanations of what these status LEDs mean. The IEC 62591 wireless application running in the ControlWave Micro includes error codes which you can check; often these will identify configuration problems for you.
- **Collect and save as much relevant information as you can.** If possible, make notes concerning what steps you took leading up to the initial occurrence of the problem. Save printouts, screen captures, error codes, and so on so you can refer to them if you need to call for technical assistance.

The remainder of this section includes:

- Troubleshooting Techniques
- Troubleshooting Checklist

4.2 Common Troubleshooting Techniques

4.2.1 Identify which System Components are Working

The wireless interface has several different pieces of hardware and software. A failure in any one of them can cause problems, so you should consider all the different pieces to try to identify the source of your problem.

For hardware you have:

- ControlWave Micro controller with IEC 62591 module installed in a slot (other than slot 3)
- Emerson Wireless 781S Smart Antenna ("781S")
- PC or laptop connected the ControlWave Micro
- Cable between the IEC 62591 module and the 781S
- One or more wireless devices in the wireless network
- Hand-held Field Communicator (optional)

For software you have:

- The IEC 62591 application (ControlWave project) running in the ControlWave Micro
- ControlWave Designer and AMS Device Configurator software running in the PC or laptop
- IEC 62591 protocol software running in the 781S and in all the wireless devices

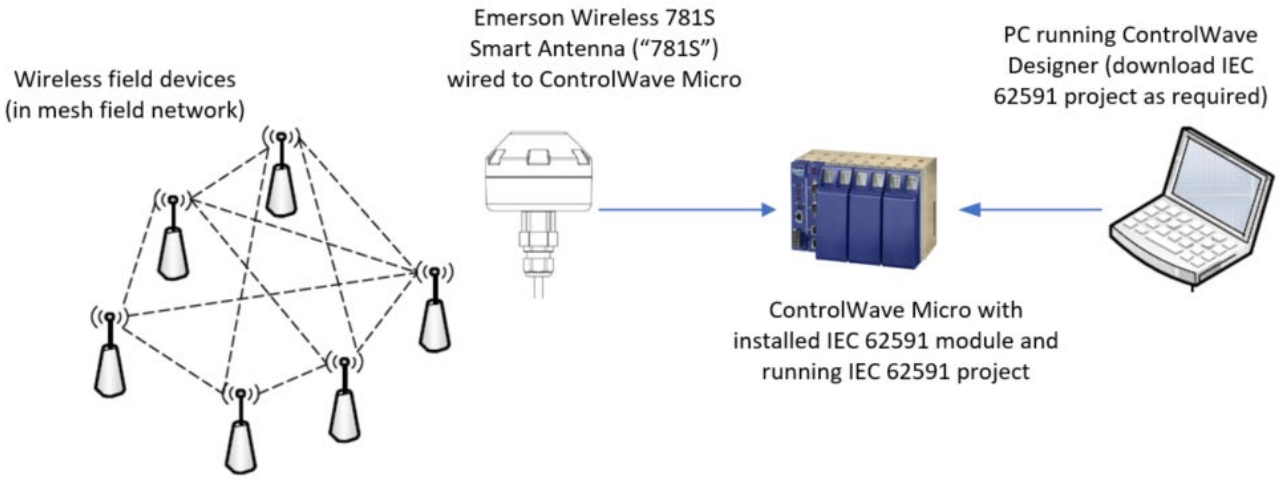


Figure 4-1. Wireless Interface

4.2.2 Basic Items to Check for Hardware

- Ensure power is connected.
- Check that all modules are properly seated in slots.
- Ensure cable connections are good between the 781S and controller, and between the PC/laptop and the controller.
- Check status LEDs on the controller. For information on what the LEDs mean, see the *ControlWave Micro Instruction Manual* (part D301392X012).
- Check for indications on the 781S. See its accompanying documentation for details.

4.2.3 Checking Error/Status Codes in the IEC 62591 Application

This procedure assumes that:

- You have ControlWave Designer installed on the PC/laptop.
- You are running the standard IEC 62591 application (ControlWave project) in the ControlWave Micro and have an identical copy of the project on the PC/laptop.
- You have a successful communication connection between the PC/laptop and the ControlWave Micro.

1. Start ControlWave Designer (if it is not already running).
2. Click **File >Open Project/Unzip Project**
3. If not already there, navigate to the OpenBSI/Projects area and open the file IEC62591_Example.mwt. When you open the project, you can see the project tree. (See *Figure 4-2*.)

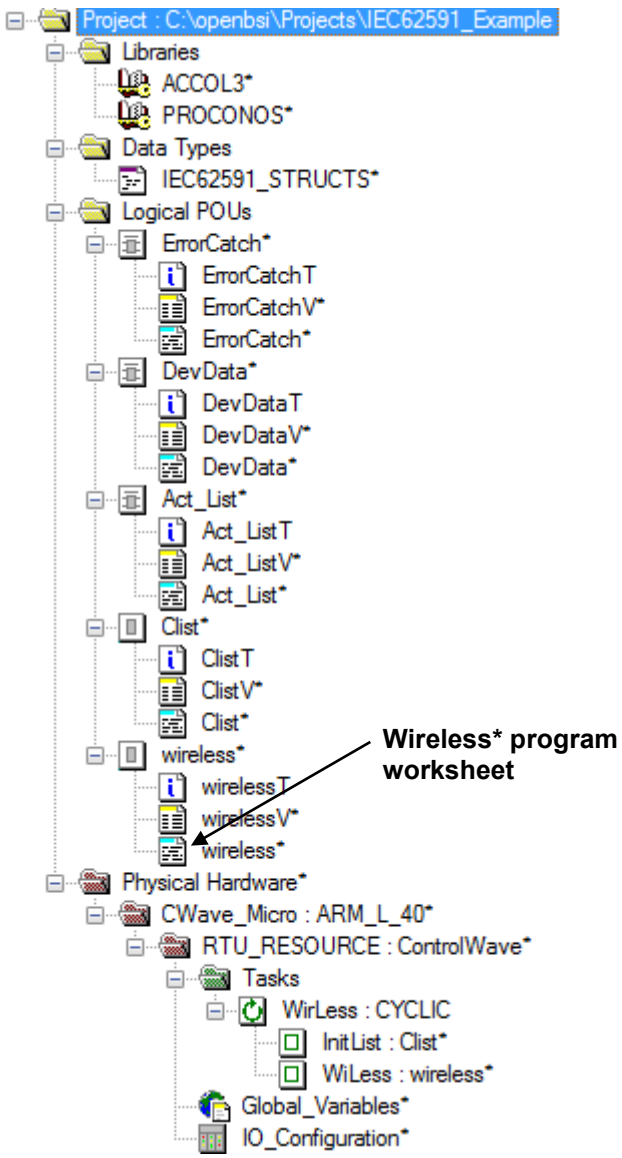


Figure 4-2. IEC 62591 Example – Project Tree

4. Click **Online > Debug**.
5. Double-click on the **wireless*** program worksheet.
6. Scroll down through the worksheet until you see the IEC 62591 function block. (In Debug Mode, a column of live values sits to the left of the program statements; these are the actual parameter values in the program as it executes in the ControlWave Micro.)

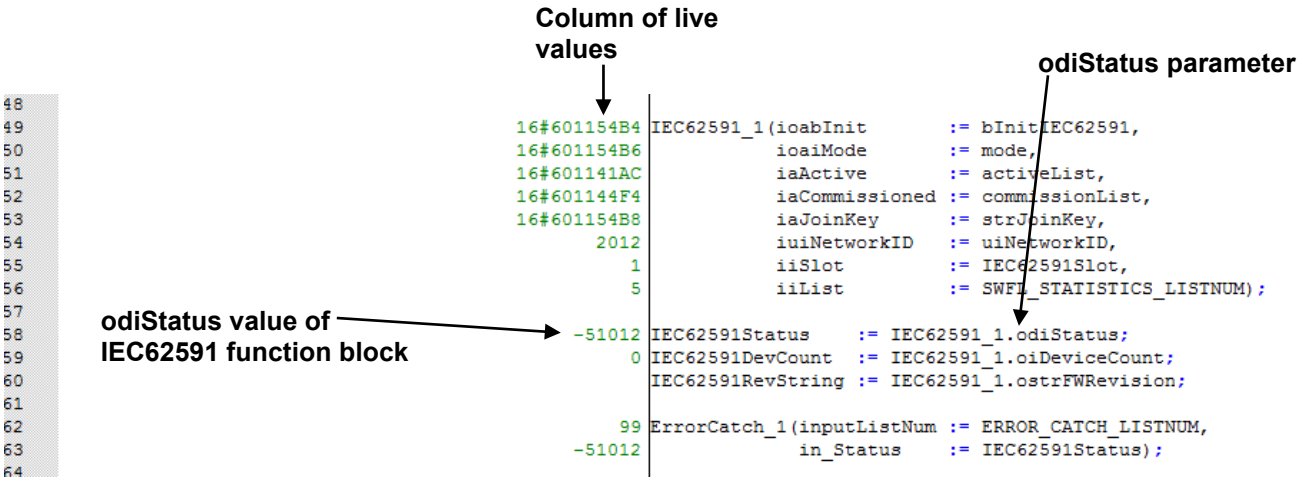


Figure 4-3. Wireless Program Worksheet

7. Find the odiStatus parameter for the IEC62591 function block (see *Figure 4-3*). In this example, the odiStatus parameter has the variable name IEC62591Status; if your programmer used a different name, then the name will be different but the parameter is the same.
8. Look for the value of the odiStatus parameter value in the column of live values. In *Figure 4-3* that value is **–51012;** the negative value indicates an error.
9. To find out what the **–51012** error means, go to the project tree and right-click on the ACCOL3 library icon, and choose **Help on 'ACCOL3' library.**

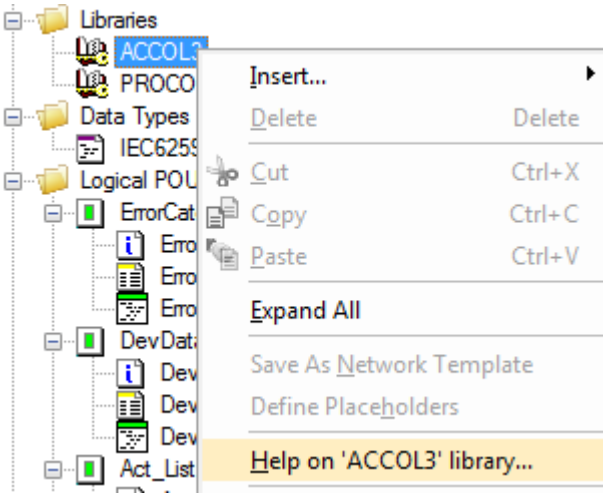


Figure 4-4. Calling Up the ACCOL3 Help Files

10. Go to the ACCOL3 Function Blocks help and scroll down (or expand the tree item) until you get to the IEC62591 help topic, then click on it.

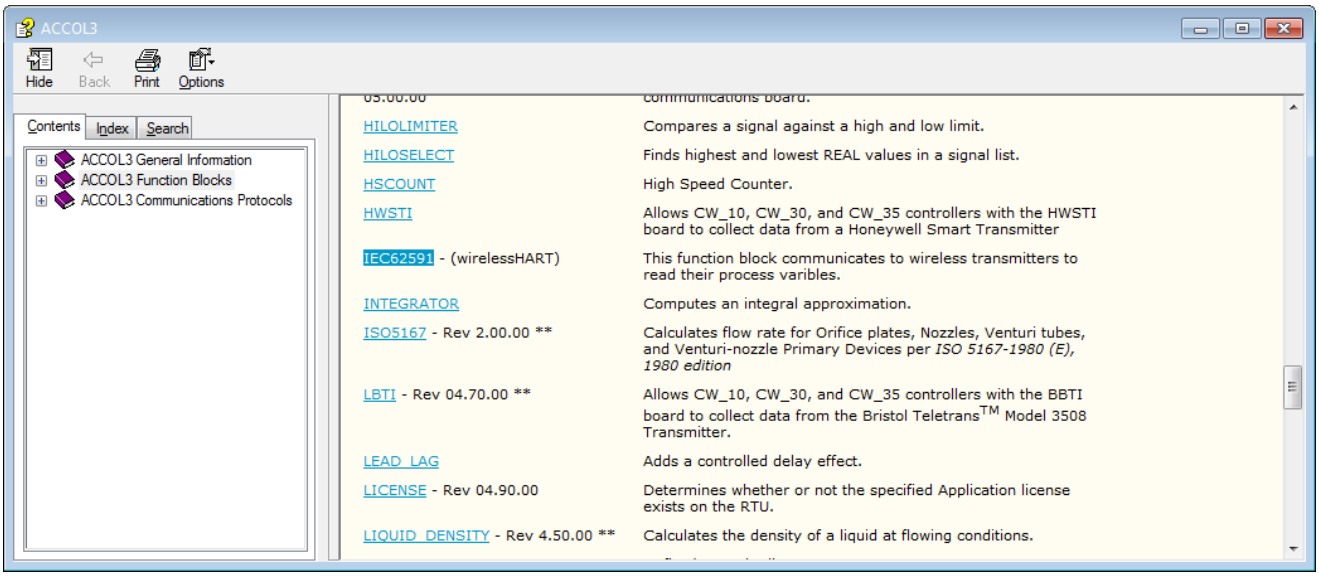


Figure 4-5. Selecting the IEC62591 Help Topic

11. Scroll down in the IEC62591 help topic until you find the **odisStatus** description and click on the **status code** link to bring up a list of status codes. (You may want to print out this list for reference as you're troubleshooting.)

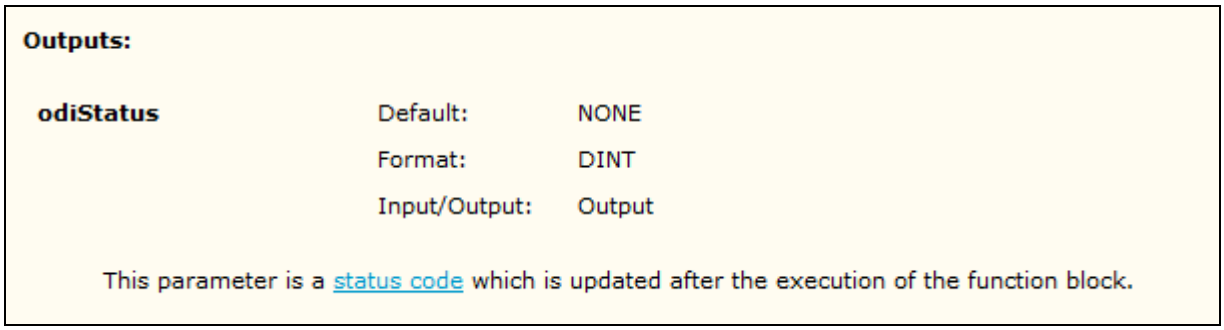


Figure 4-6. Status Code Link

12. Find the code you're looking for (here, **–51012**) and you'll see what the error means. For "-51012" the error says:

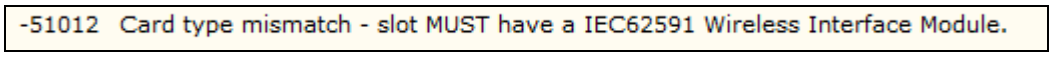


Figure 4-7. Example of Error Message

That means the IEC 62591 function block **does not** detect an IEC 62591 Wireless Interface module in the specified slot. This happens if you specify an incorrect slot number in the application or if the module is missing entirely.

Alternatively, if you know the complete error number, you can use the Search tab to type in the error code, then click **List Topics**, and double-click on the results to see the error.

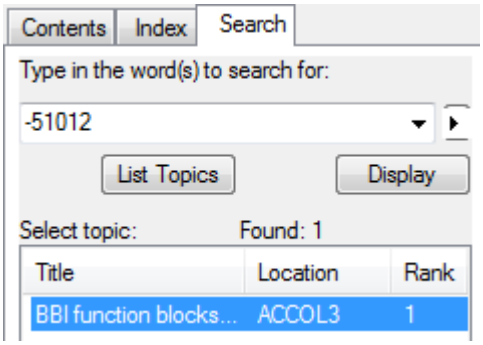


Figure 4-8. Search Tab

Note: You can follow this same basic procedure to look up any odiStatus parameter value. In addition, if you specified a list for storing error totals in the application, you can view statistics on what types of errors are occurring. To do this, you should double-click on the ErrorCatch worksheet. Alternatively, you could use DataView to view the ErrorCatch list.

4.3 Troubleshooting Checklist

- ☐ Does the IEC 62591 Wireless Interface support your wireless device(s)? Check the product data sheet (available on *www.Emerson.com*) to verify that your devices have been tested with the interface.
- ☐ Did you assign a unique Long Tag Name to each wireless device and specify the exact same long tag names in the IEC 62591 application running in the ControlWave Micro? If not, use the AMS Device Configurator software or a 375/475 Field Communicator to set the long tag name in the device, and use ControlWave Designer to set the long tag names within the IEC 62591 application running in the ControlWave Micro.
- ☐ Did you assign a Network ID which must be the same in **each** wireless device in this network, and must also match the Network ID defined in the IEC 62591 application running in the ControlWave Micro? If not, use the AMS Device Configurator software or a 375/475 Field Communicator to set the Network ID in the devices, and use ControlWave Designer to set the Network ID within the IEC 62591 application running in the ControlWave Micro.
- ☐ Did you assign a Join Key which must be the same in each and every wireless device in this network, and must also match the Join Key defined in the IEC 62591 application running in the ControlWave Micro? If not, use the AMS Device Configurator software or a 375/475 Field Communicator to set the Join Key in the devices, and use ControlWave Designer to set the Join Key within the IEC 62591 application running in the ControlWave Micro.

- ☐ Is the IEC62591 module in the ControlWave Micro in the proper slot as specified in the IEC 62591 application running in the ControlWave Micro? If not, place the module in the proper slot or change the slot number defined in the IEC 62591 application to match the slot in which the module resides.
- ☐ Are you using the proper datatypes for the Commission List and Commission Point?
- ☐ Is the decommission flag for each device set to FALSE in the IEC 62591 application running in the ControlWave Micro?
- ☐ Does each device in the network have power? If not, connect power.
- ☐ Are cables connected properly between the 781S and the ControlWave Micro?
- ☐ Are any status LEDs lit on the ControlWave Micro PSSM module? If so, consult the *ControlWave Micro Instruction Manual* (D301392X012) for more information.
- ☐ Did you check for error/status codes in the IEC 62591 application? If not, follow the instructions in *Section 4.2.3*.

4.4 Best Practices

While the IEC62591 function block makes device data available to the application and SCADA it does not take independent action unless the application is designed to do so. The best recommended practice to ensure that the end device is operating optimally is to either configure the SCADA system to monitor the status of the field devices or develop application solutions using the RTU's alarm, event, or Report By Exception features. Information needed to make these decisions is available in following elements of the commission list entry and the IEC62591 function block status:

- **odiStatus:** IEC6259 function block status output parameter
- **obActive**: Device is actively communicating
- **ousDeviceStatus**: Device status bits
- **ouiBatteryDays**: Status of the device's lithium battery
- **ousDevCommishStatus:** Device commissioning status
- **iousiFailSafeMode**: Device failsafe mode
- **obyteNaNFlag:** Whether slot or process variable(s) are Not-A-Number
- **obyte'xx'Status:** Four process variable status bits
- **obyteLive'n'Status:** Four live value status bits

4.5 Data Updates

If you notice that data updates for process variables are slow, this can occur if the IEC 62591 device cannot process a data request fast enough. This can generate a delayed response message, causing the RTU to re-request the data.

To see if you are receiving delayed response messages, look for a 9 (Delayed Response Received) status on the **ousDevCommishStatus** parameter of the Commission List entry.

[This page is intentionally left blank.]

Index

For customer service and technical support,
visit *www.Emerson.com/SupportNet*.

Global Headquarters,
North America, and Latin America:
Emerson Automation Solutions
Remote Automation Solutions
6005 Rogerdale Road
Houston, TX 77072 U.S.A.
T +1 281 879 2699 | F +1 281 988 4445
www.Emerson.com/RemoteAutomation

Europe:
Emerson Automation Solutions
Remote Automation Solutions
Unit 1, Waterfront Business Park
Dudley Road, Brierley Hill
Dudley DY5 1LX UK
T +44 1384 487200 | F +44 1384 487258

Middle East/Africa:
Emerson Automation Solutions
Remote Automation Solutions
Emerson FZE
P.O. Box 17033
Jebel Ali Free Zone – South 2
Dubai U.A.E.
T +971 4 8118100 | F +971 4 8865465

Asia-Pacific:
Emerson Automation Solutions
Remote Automation Solutions
1 Pandan Crescent
Singapore 128461
T +65 6777 8211 | F +65 6777 0947