Application Note

RF Communication Using Two MDS GP Application Boards

AN019202-1207

Abstract

This Application Note describes the communication between two Zilog® General-Purpose Modular Development System Application Board (MDS GP), each including Z8F642x microcontroller (MCU), via an RF link. The Z8F642x MCU's UART ports connect to the HyperTerminal communications application running on a PC to view the exchanged data. The MaxStream XStream RF Module, which is easily plugged into the underside of the MDS GP Application Board, is used for communicating via the RF link.

The source code associated with this Application Note can be found in the `AN0192-SC01.zip` file, which is available at www.zilog.com.

MDS GP Application Board Overview

Zilog's MDS GP Application Board is a platform for designing innovative applications using the Z8 Encore! XP® and eZ80Acclaim!® MDS-compatible microcontroller modules. The MDS GP Application Board features a character LCD display, a keypad matrix, 512 KB of fast SRAM, two relays with a terminal block, six high-drive outputs with terminal blocks, and an RS-485 interface/screw terminal block. Battery operation is supported to add portability and mobility to your design.

The MDS GP Application Board acts as the perfect foundation for innovative development ideas, with compatible connectors for the optional Trimble Lassen™ SQ GPS Module, the Maxstream Wireless Data Module (900 MHz or 2.4 GHz), and the Dinsmore 1490 Digital Compass, along with a large breadboard area.

Z8 Encore! XP Flash Microcontrollers Overview

Zilog's Z8 Encore! XP products are based on the new eZ8 CPU and introduce Flash memory to Zilog's extensive line of 8-bit MCU. Flash memory in-circuit programming capability allows for faster development time and program changes in the field. The high-performance register-to-register based architecture of the eZ8 core maintains backward compatibility with Zilog's popular Z8® MCU.

Z8 Encore! XP MCUs combine a 20 MHz core with Flash memory, linear-register SRAM, and an extensive array of on-chip peripherals. These peripherals make the Z8 Encore! XP MCU suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Discussion

Communication between MCU devices can occur in several ways. This section focuses on the following methods:

- Communication via cables (wired communication)
- Communication via electromagnetic radiation, which includes
  - Visible (light) waves
  - Infrared waves
  - Radio waves

Wired Communication

Wired communication use cables to exchange information. Some examples of wired communication standards are RS-232, RS-485, and I^2C. The advantages of using wired communications are:

- It is not necessary for the communicating devices to be within line-of-sight of each other.
- The cables used for communication also serve as a power line for any additional microcontrollers.
- The cost for short distance communication is within acceptable limits.

There are inherent limitations imposed by wired communication, however. These limitations are:

- Distance between devices can be a factor when communicating via wired communications; longer distances require more cable length, which increases cost.
- Longer distances can often require focus on cable layouts, which can be time-consuming and impact cost.
- The impedance of the cable greatly affect the communication process.

Communicating Using Light or Infrared Waves

The infrared medium or the fiber optic technology that uses light waves is used to transmit information. Some of the advantages of using these methods are:

- Additional devices can communicate without requiring a new layout of the communication link between existing devices.
- The power supply lines of the communicating devices are totally isolated from each other and do not affect the performance of similar devices within the vicinity.
- Communication speed is superior to wired communication.

These two wireless technologies also have their limitations.

- For infrared communication, the communicating distance is limited because the communicating devices must remain within line of sight.
- For fiber optic communication, the cost is high.

Communication Using Radio Waves

With radio frequency (RF) communication, information exchange between devices occurs via radio waves that are propagated through the air. The following list contains some of the advantages of using the radio frequency (RF) communication medium.

- The cost is lower compared to the fiber optic medium.
- Designing a new layout of the communicating devices is not critical.
- Over longer distances, RF communication is more advantageous than wired and Infrared communication media.
- Between RF communication devices, the power lines are isolated from the signal lines.

The main limitations of RF technology are:

- The frequencies available for RF communication are limited.
- Restricted power must be used to prevent electromagnetic interference.
- Electrostatic discharge in the atmosphere affects radio wave transmissions.

Weighing the advantages and disadvantages of the different communication media presented in this section, it is apparent that the most flexible medium to use between devices communicating over long distances is the RF medium.

The MaxStream XStream RF Module

For purposes of this demonstration, an RF application running on the MDS GP Module uses the MaxStream RF data module, part number X09-019WNC. This MaxStream module can be directly mounted on the RF header provided on the underside of the MDS GP Application Board. This RF module operates in the 900 MHz range with a transmit power of only 140 mW and a 5 V power supply. According to the product specifications for the RF module, the transmit distance can reach up to 450 meters in indoor and urban settings using a 3-inch stranded wire antenna soldered onto the antenna port of the XStream RF module. When connected to a high-gain antenna, the distance can reach up to 32 km within line of sight. The XStream RF module operates at a default baud rate of 19200 bps via the RS-232 serial data interface.

The XStream RF module can be used for advanced networking and security with protocols such as true peer-to-peer (with no master required), point-to-point, point-to-multipoint, and multidrop protocols.

The dimensions of the XStream RF module are 2.825 inches in length x 1.600 inches in width x 0.665 inches in height, which is ideal for small-scale applications requiring a long transmission range.

Developing the MDS GP-Based RF Application

To demonstrate the RF application, two of Zilog's MDS GP Application Boards are used with the Z8F642x Development Board plugged into them and the XStream RF module affixed to the underside of each board. Figure 1 displays the MDS GP Application Board. This diagram displays where the Z8F642x board and the RF module are located.

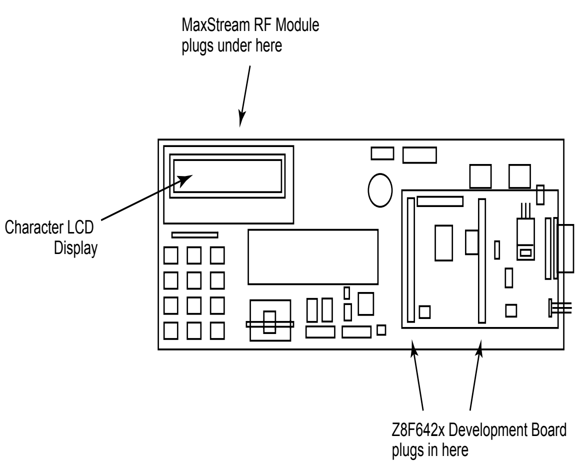

Figure 1. Zilog's MDS GP Application Board

The MDS GP Application Board serves as an accessory kit or an expansion board for the Z8F642x development system. The Xstream RF module features a ready female header to communicate with the UART0 port pins of the Z8F642x MCU. The basic antenna that comes with the XStream RF module is a 3-inch stranded wire antenna that allows communication over a distance of up to 450 meters within line of sight according to the product specifications of the XStream RF module.

A discussion of the hardware architecture and the software implementation used for the RF application follows in the next sections.

Hardware Architecture

Figure 2 displays a block diagram that describes the hardware connections between the XStream RF module and the Z8F642x Development Board. These connections are already present on the MDS GP Application Board.

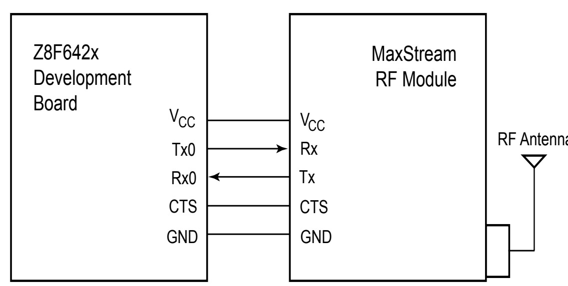

Figure 2. Connection between the Z8F642x Board and the RF Module

The network configuration for the Z8F642x MCU is peer-to-peer (no Master required). For the purpose of the demonstration, two MDS GP boards are used, each featuring a Z8F642x MCU and an RF module. Each Z8F642x MCU must be preprogrammed with a unique, one-byte address (ranging from `00h` to `FFh`) to avoid data bus contention.

The serial port (UART0) of the Z8F642x MCU is used to connect to the RF module. A user-defined protocol is used to establish an RF link between the processors.

The user-defined protocol programmed to the Z8F642x MCU is comprised of the following facets (see Figure 3):

- Data synchronization to indicate that RF data is going to be transmitted.
- An RF header to indicate that valid RF data is being transmitted.
- The destination address (to allow the receiver to either accept or reject the data).
- The source address to indicate the sender's address.
- The message/frame length in bytes.
- The message/frame data (ASCII, HEX, or otherwise).
- A terminating byte to indicate an end of message.

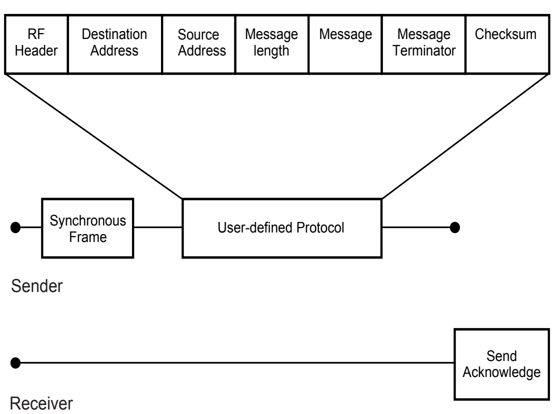

Figure 3. User-Defined Protocol for RF Communication

The 16-bit checksum is a reference for the receiving device to acknowledge that a valid message has been received.

The receiver sends an acknowledgement only if the correct checksum is obtained when compared with the reference checksum transmitted. If the receiver does not acknowledge the message transmitted within the waiting period, the message is sent three more times before the transmission is aborted. The system informs you that the message was not sent successfully.

Software Implementation

The software is designed to facilitate RF communication between any of the Z8 Encore! XP microcontrollers. The network strategy implemented is a true peer-to-peer communication where a master device is not required.

The reception of the message is interrupt-driven. An LED indicator at the receiving MCU's Port A Pin 0 (PA0) illuminates whenever a new message is received. To view the received message that currently resides in the inbox buffer, the **r** key is entered at the HyperTerminal prompt. The received message is then displayed in the HyperTerminal console.

This application is designed to transmit and store messages that are stored in a message box located in the upper Flash memory area of the Z8 Encore! XP MCU. The message box is implemented as a circular buffer to store a maximum of 5 messages. Storing the messages in the upper Flash memory area prevents loss of message in the event of any inadvertent power interruptions.

To remove the messages from the Flash memory area, the `Delete message (d)` option is selected from the main menu displayed in the HyperTerminal window.

Table 1 lists the details contained within the RF user protocol message frame.

Table 1. RF User Protocol Message Frame

RF Data	Description	Length of Data
‘$’	RF start header	1 byte
Source address (e.g 00h)	Sender address	1 byte
Destination address (e.g 01h)	Receiver address	1 byte
Message length (e.g. 2Bh)	RF Frame length	1 byte
Message data (e.g. ASCII string)	Sender data (message)	Maximum of 255 bytes (default)
0x00	Message terminator	1 byte
‘,’	RF frame terminating string	1 byte
Checksum (e.g 3Ah)	Odd byte frame checksum	1 byte

Table 2 lists details contained within the RF user protocol acknowledge frame.

Table 2. RF User Protocol Acknowledge Frame

RF Data	Description	Length of Data
‘$’	RF start header	1 byte
Destination address (e.g. 01h)	Reply address (sender address)	1 byte
Source address (e.g. 00h)	Receiver address	1 byte
0x02	Acknowledge frame length	1 byte
0x7F	Acknowledge data	1 byte
0x00	Message terminator	1 byte
‘,’	RF frame terminating string	1 byte
Checksum (e.g. 5Ah)	Odd byte checksum	1 byte
Checksum (e.g. 1Ch)	Even byte checksum	1 byte

Note: *The RF protocol described above is specific to the application described in this Application Note. It is not a standard protocol.*

The main function of the RF module is to transmit/receive a message via the RF link. The main function of the Z8 Encore! XP MCU is to encode and decode the message(s). The software is implemented in the form of the three functions and one Interrupt Service Routine (ISR) that are described below.

init_uart()

The `init_uart()` function performs the following tasks:

- Calculates baud rate values for the UART baud rate generator based on the frequency and baud rate specified
- Configures the GPIO UART ports
- Enables the control register for the UART ports (Tx and Rx) based on the selected mode

Syntax

```
void init_uart
(
long int frequency,
long int baud rate,
char mode,
char uart);
```

send_rf()

The `send_rf()` function performs the following tasks:

- Encodes the RF user-defined protocol
- Computes the RF message length
- Calculates the checksum
- Sends a synchronizing signal
- Sends the RF message frame
- Waits for a reply from the receiver
- Returns a 1 if acknowledgement is received or else returns a 0 after three unsuccessful attempts.

Syntax

```
char send_rf
(
char *source,
char destination);
```

Figure 10 on page 13 displays the flow of the `send_rf()` routine.

send_acknowledge()

The `send_acknowledge()` function performs the following tasks:

- Encodes the RF user-defined protocol
- Includes the RF acknowledge byte
- Computes the checksum
- Sends the acknowledge frame to the sender

Syntax

```
void send_acknowledge
(
char destination);
```

Figure 11 on page 14 displays the routine that processes the data and sends the acknowledgement to the sender.

isr_UART0_RxD()

The `isr_UART0_RxD()` ISR performs the following tasks:

- If a valid RF header is received, then the function enables the RF receive routine
- Checks if the destination address is the same as the default address
- Validates the destination address and stores the message in the receive buffer
- Calculates the checksum of the frame and compares the frame checksum to the calculated checksum
- If the checksum is the same, this function activates the *message received* LED connected at PA0
- Sends an acknowledgement to the sender

Syntax

```
#pragma interrupt
void isr_UART0_RxD(void);
```

Figure 12 on page 15 displays the ISR that receives data in the RF receive data buffer.

Demonstration

This section contains the details necessary for demonstrating RF communication between MDS GP Application Boards with the Z8F642x Development Board plugged into it. The MDS RF Demo setup uses two MDS GP Application Boards with Z8F642x Development Boards affixed, two PCs running the HyperTerminal application, and the MaxStream XStream RF module. The HyperTerminal application is used for display and as an user interface for MDS RF communication. The RF modules are located on the MDS board on the reverse side of the LCD display. The demo setup diagram is displayed in Figure 4.

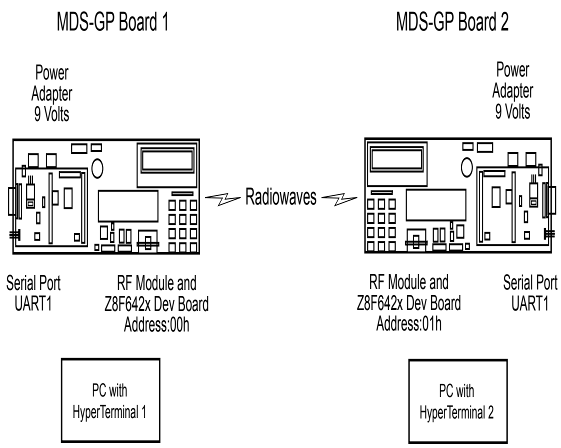

Figure 4. Demo Setup for RF Communication between Two MDS GP Boards

Equipment Used

The following equipment are used in the setup:

- 2 MDS GP Application Boards (ZGENPRP0100MDS) with Z8F642x Development Boards
- 2 Smart cables to connect the PC to the development boards
- Zilog ZDS II–Z8 Encore! (IDE)
- 2 DC adapters (5 V, 9 V, or 12 V) or (4) pairs AA batteries
- 2 MaxStream XStream RF Modules (Maxstream X09-019WNC) attached to the underside of the MDS GP board
- 2 PCs with the HyperTerminal application installed

HyperTerminal must be modified to reflect the following settings:

- Communication port = COM1 or any DB9 serial port
- Baud rate = 19200 bps
- Data = 8 bits, no parity, and one stop bit
- No flow control

Setting Up the MDS RF Demo

Follow the steps below to set up the MDS RF Demo:

1. Connect the Demo setup as displayed in Figure 4.
2. Set the HyperTerminal applications running on PC 1 and PC 2 according to the settings provided at the end of the previous section. HyperTerminal is used to view a message or enter a new message.
3. Up on power-up, the application software automatically detects the current configuration of the J2 pin on the MDS board. Jumper J2 can be used to enable the RF and RS232-2 ports. If there are no shorting jumpers installed on J2, the application software enables both or either of the RF and RS232-2 ports depending on the preset J2 configuration on the MDS GP Application Board.
4. Connect the power adapters to each MDS board. For mobility, you can also install four (4) AA batteries to power the MDS Board.
5. Download the `AN0192-SC01.zip` file from www.zilog.com. Extract the file contents to a folder. This folder will be referred to as: `\<extracted folder>`

6. Launch ZDS II and open the `rf_main.pro` file located in the `\<extracted folder>`.
7. Open the `rf_api.h` file and locate the following hash define:

   `#define MY_ADDRESS`
8. Change the default address `0` to a unique 1-byte HEX address or a 1-byte decimal address.
9. Recompile and build the project, then download the program to the Z8F642x board[1].
10. For each additional Z8F642x board, repeat the above steps while entering a unique address in Step 8 for each microcontroller.

When the MDS RF Demo is set up and the application code is downloaded successfully, the MDS RF Demo is ready to be executed.

Notes: *For schematic diagrams and further details about the MDS GP Application Board, refer to Z8 Encore! Flash Microcontroller Development Kit User Manual (UM0146).*

For schematic diagrams and further details about the Z8F642x Development Board, refer to MDS General-Purpose Board Development Kit User Manual (UM0169).

Executing the MDS RF Demo

Follow the steps below to execute the MDS RF Demo.

1. Up on power-up of each MDS GP board, a menu is displayed on the HyperTerminal console that displays a list of available commands. Figure 5 displays a screen shot of the main menu displayed at the sending device's HyperTerminal window.

Note: *If the menu is not displayed, double-check the connections and HyperTerminal settings.*

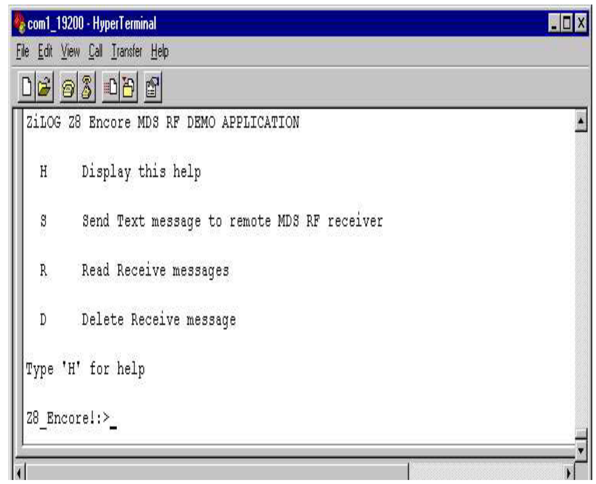

Figure 5. Main Menu of the MDS RF Application

2. Type **s** at the HyperTerminal prompt to send a text message.
3. Enter the destination address, which is the address of the other MDS GP board. When a valid destination address is entered, you will see a prompt on the HyperTerminal console to enter a message to be sent to the receiver.
4. Type a text message and press the **Enter** key. Another prompt asking whether to send the message or not is displayed.
5. Type **N** at the prompt to continue typing the message. Type **Y** to send the typed message to the specified destination address. Figure 6 on page 9 displays a screen shot of the display when text is sent from the sending device.

[1]For details about how to download the code to Flash memory on the Z8 Encore! MCU, refer to *Zilog Developer Studio II–Z8 Encore! User Manual (UM0130)*.

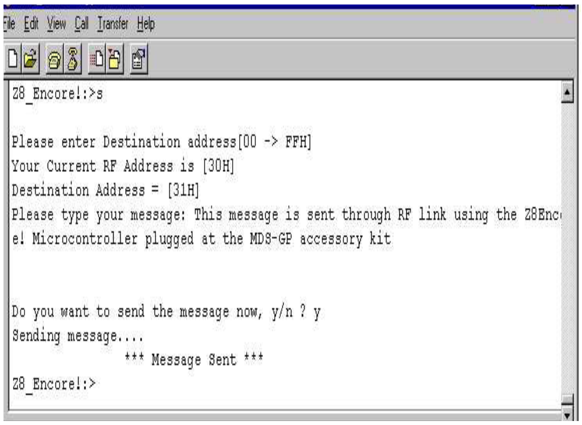

Figure 6. Sending a Text Message

6. At the sending device's HyperTerminal console, a **Message Sent** notification is displayed when the receiving device acknowledges a valid message. Figure 7 displays a screen shot of these entries and messages on both the sending and receiving device HyperTerminal consoles.

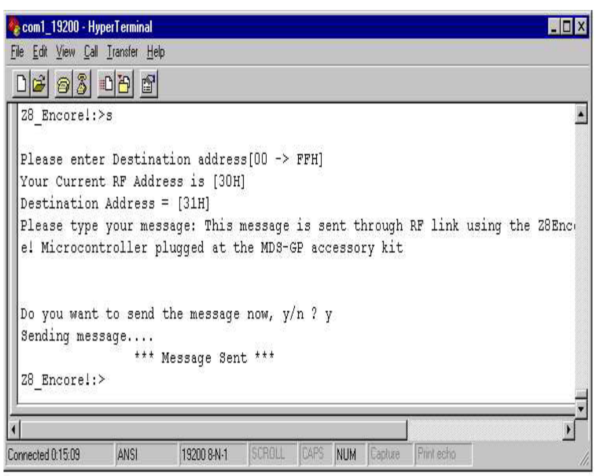

Figure 7. Message Sent Notification at Sending Device

When an invalid destination address is entered, the processor, up on verifying that the destination address is not the default address of the receiving device, displays the message in Figure 8.

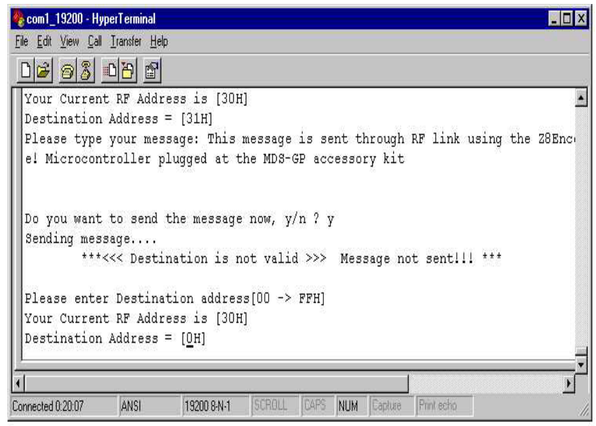

Figure 8. Message Not Sent Notification at Sending Device

7. Press **Esc** on the keyboard to return to the main menu at any point in the transmission of messages.

Notes: *If an acknowledgement message is not received from the receiving device, the HyperTerminal console operating on the sending device displays the message* **Message Not Sent** *and prompts for a valid address at its HyperTerminal prompt.*

The receiving device sends an acknowledgement to the sending device before the message-received LED (green) is activated on the Z8F642x board of the receiving device.

After a message is successfully sent, the sending device's HyperTerminal console displays the main menu. If a message is transmitted but the data received is corrupted, the corrupt LED (red) illuminates on the Z8F642x board of the receiving device to indicate that the message received was corrupted.

8. Type **r** at the receiving device's HyperTerminal prompt to read the received messages. You are prompted for the number of the message. When a valid message number is selected, the received message is displayed. See Figure 9 on page 10.

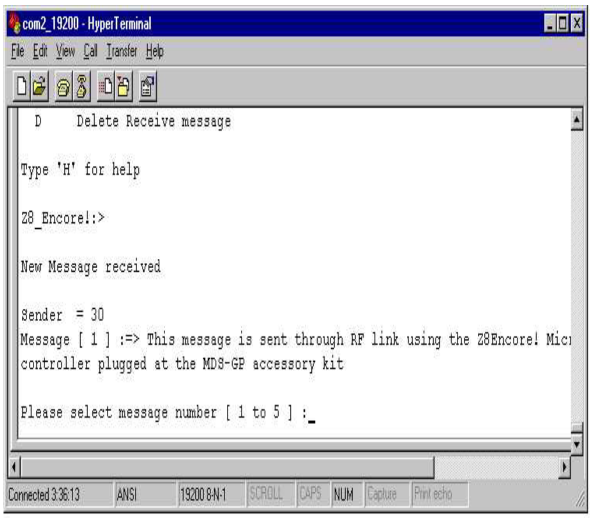

Figure 9. Message Received Notification at Receiving Device

Notes: *The prompt displays the maximum number of text messages. This message number corresponds to the internal message counter pointer stored within Flash memory. The message counter recycles whenever the maximum message number is exceeded. The previous message stored is overwritten with the new message that is now pointed to by the message counter.*

Selecting any non-integer or out-of-range message number causes the following message to be displayed on the console:

```
***Sorry message number requested is not valid!***
```

9. Enter **d** at the prompt to delete the stored text message. A notification that the message is successfully deleted is received.

Results

The MDS RF Demo was executed successfully and tested for a range of working distances. In Table 3, RF-A refers to MDS GP Application Board 1, and RF-B refers to MDS GP Application Board 2. RF-A was placed at a fixed location. RF-B was powered by four AA batteries to render it mobile. RF-A was configured to continuously transmit messages to RF-B at 2-second intervals. RF-B was moved to locations that were 10 meters away from the previous location, starting from RF-A. The LED indicators on RF-B were checked every 10 meters.

When the new message indicator (green LED) illuminates, it indicates that RF-B has received the message and has sent an acknowledge message to RF-A.

70 m is the maximum distance (with obstructions) at which this setup was tested. However, this distance does not represent the maximum distance that can be covered with this setup. See Table 3.

Table 3. Working Distance between R-A and RF-B

RF-B Distance from R-A	Status
10 meters	New message indicator GREEN LED illuminates.
20 meters	New message indicator GREEN LED illuminates.
30 meters	New message indicator GREEN LED illuminates.
40 meters	New message indicator GREEN LED illuminates.

Table 3. Working Distance between R-A and RF-B (Continued)

RF-B Distance from R-A	Status
50 meters	New message indicator GREEN LED illuminates.
60 meters	New message indicator GREEN LED illuminates.
70 meters	New message indicator GREEN LED illuminates.

The antenna used for this setup was the integrated 3-inch wire antenna featured on the MaxStream RF module. This antenna was sufficient for the purposes of this MDS RF application. For applications requiring longer distances, the recommended dipole or high-gain antenna can be used, which can extend the distance up to 32 km line-of-sight, depending on the RF module's product specifications.

Precautions and Limitations

Following are the limitations and considerations when setting up and executing the MDS RF Demo:

- The Maxstream Xstream RF Module with its 3-inch stranded wire antenna can transmit data up to 450 m line-of-sight without any obstruction between the two RF modules. When using any other RF module, it is to be noted that successful data transmission via the RF link is dependent on the power transmitted by the RF modules, and the obstructions present between the modules. Refer to the product specifications of each of the RF modules you employ to determine recommended distance.
- The Z8 Encore! XP boards are physically labeled with their RF addresses to avoid communication conflict. It is strongly recommended that you follow this labeling by pasting a unique RF address on the MDS GP boards that they use for the demonstration.
- The default maximum data that the application can store is 1 KB. However, you can change the values to fit your application's capacity, given due consideration to the available Flash memory area. To change the message buffer memory size, make changes in the `RF_API.h` file contained in the `AN0192-SC01.zip` file.
- If you do not choose to use the RF module discussed in this Application Note, Zilog recommends that you select an RF module based on the following criteria:
  - Use RF modules that do not use the same operating frequency as existing modules within the vicinity.
  - Assign a unique address to each device to distinguish them from existing RF communicating devices.

Summary

This Application Note describes the implementation of RF communication between two MDS GP Application Boards for up to a distance of 70 meters (with obstacles) using the Z8 Encore! XP development boards and the RF modules plugged onto them.

The MDS GP Application Board is a feature-rich board that serves as a peripheral platform for the Z8 Encore! XP development boards. These development boards plug directly onto the MDS GP Application Board.

The built-in features available on the MDS GP boards allow a great deal of flexibility. The MDS GP board is designed to accommodate a MaxStream RF module plugged into its underside, allowing for communication with other MCUs via an RF link, which is free of all of the complications of direct-wired connections. The battery option adds the final touch, providing complete mobility to the communicating devices.

References

The documents associated with the MDS GP Application Board, Z8 Encore! XP products, ZDS II, and the RF Module used for the demonstration can be found in the references listed below.

- MDS General Purpose Board Product Brief (PB0146)
- MDS General Purpose Board Quick Start Guide (QS0045)
- MDS General Purpose Board Development Kit User Manual (UM0169)
- Z8 Encore!® Flash Microcontroller Development Kit User Manual (UM0146)
- Z8 Encore! XP® 64K Series Flash Microcontroller Development Kit User Manual (UM0151)
- Z8 Encore! XP® F0822 Series Flash Series Development Kit User Manual (UM0150)
- Z8 Encore! XP® 64K Series Flash Microcontrollers Product Specification (PS0199)
- eZ8 CPU Core User Manual (UM0128)
- Zilog Developer Studio II–Z8 Encore!® User Manual (UM0130)
- MaxStream XStream RF Module document available on http://www.maxstream.net

Appendix A—Flowcharts

This appendix contains flowcharts that characterize the association between the MDS-GP Application Board and the RF application described in this document.

The routine for sending data via an RF link is displayed in Figure 10.

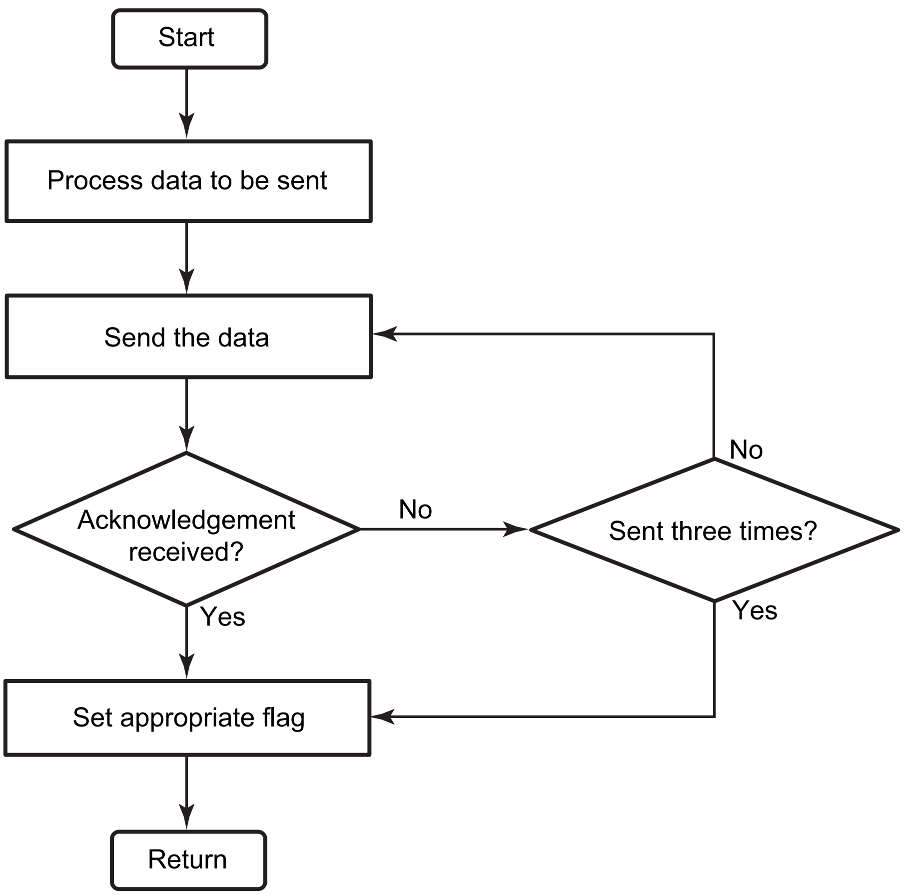

Figure 10. Flow of the RF Send (`send_rf()`) Data Routine

The routine for processing data is displayed in Figure 11.

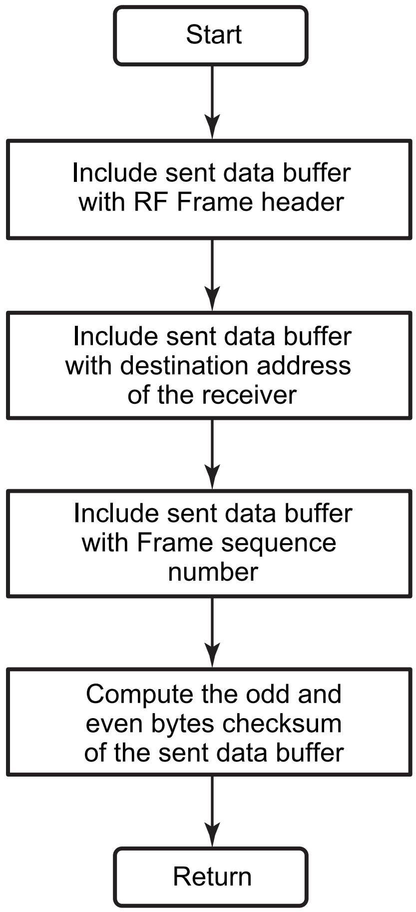

Figure 11. Flow of the Process Data (`send_acknowledge()`) Routine

Figure 12 displays the flow of the interrupt-driven routine that receives data from the RF receive buffer.

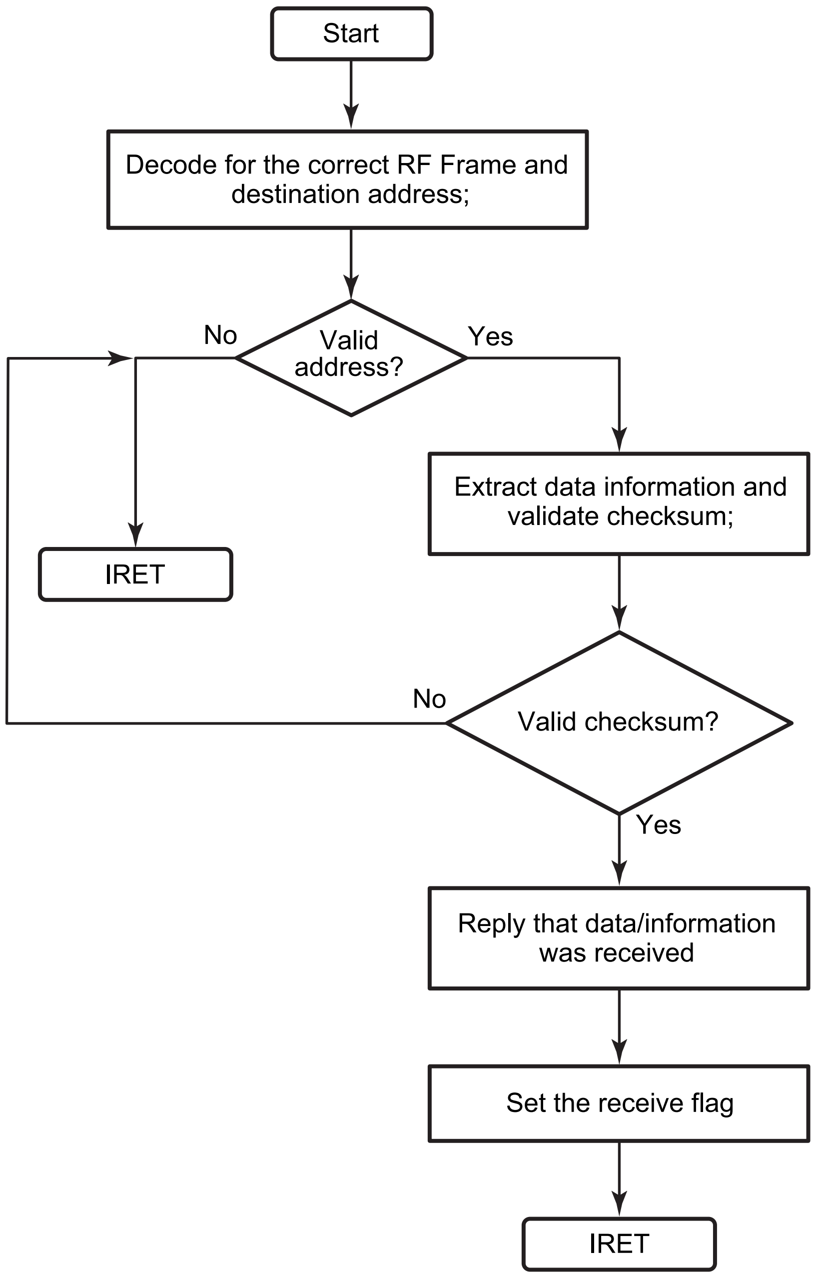

Figure 12. Flow of the RF Receive Buffer (`isr_UART0_RxD()`) Routine

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer