Keywords: wireless automatic meter reading, AMR, water meter, transceiver, transmitter, receiver, ISM, RF, PCB, layout, schematic, hardware, firmware, microcontroller, utility company, smart grid, protocol, antenna, link, range

REFERENCE DESIGN 5404 INCLUDES: √Tested Circuit √Schematic √BOM √Board Available √Description √Test Data √Software √Layout

LFRD003: Water Meter Automatic Meter Reading (AMR) Reference Design

By: Martin Stoehr, Principal Member of the Technical Staff, Applications
May 11, 2012

Abstract: This reference design provides a complete demonstration platform for using industrial/scientific/medical radio frequency (ISM-RF) products in an automatic meter reading (AMR) application. This document includes the hardware, firmware, and system structure requirements for implementing an AMR design that demonstrates operation with a physical water meter.

More Information
- Wireless Home
- Application Notes and Tutorials
- EV Kit Software
- Technical Support

General Description

Click here for an overview of the wireless components used in a typical radio transceiver.

The MAX7032 transceiver reference design (RD) is a self-contained evaluation platform for exercising the device as a wireless automatic meter reading (AMR) and water meter demo system. With the use of the Maxim USB-to-JTAG board (MAXQJTAG-USB), the MAXQ610 on both the handheld interface (HHI) unit and the various meter (MTR) radio modules can be programmed by the end user.

The meter board enables basic human interaction through a single momentary switch input and an LED for visual feedback. The MTR is designed to be compact, providing a self-contained transceiver board with a radio, microcontroller, and multiple "ports" for connecting various meter inputs to the system. Two separate designs are provided: one with an MMCX antenna connection, and the other with a small-footprint antenna-mounting option. Input to the MTR system can be configured with up to six ports, and the primary input interfaces with a "pulse" or dry contact (reed) output meter. This board can be operated from any 3V power source (1.7V to 3.6V for the MAXQ610, 2.1V to 3.6V for the MAX7032).

The HHI board has seven menu keys for user input, a reset switch, a 102 pixel x 64 pixel LCD display with multicolor LED backlighting for menu interactions, plus a receive signal indicator (RSI) LED. The shape of the HHI fits within a Series 55 BOX enclosure, and has a high-density connector for interfacing with an RF module.

Both systems are preprogrammed with operational firmware to demonstrate a simple wireless AMR meter (slave)/reader (master) system. Gerber files are available for simple cut-and-paste designs of the radio sections or the full implementation.

Features

- Proven printed circuit board (PCB) layout
- Proven component parts list
- Preprogrammed transceiver (TRX) pair for quick demonstration capabilities
- Free MAXQ® microcontroller programming tools available for flexible operation

The Handheld Interface (HHI)

The HHI is a two-part system that comprises a human interface board, which contains a LCD display, a key encoder, and a microcontroller. This HHI is mated to an RF module to form a complete handheld communication system. The Phase I design interfaces to a MAX7032 transceiver module through a 10-pin, high-density connector. The Phase III design and the RF modules described later interface through a 60-pin connector.

The HHI board provides a basic method of input with seven keys and one reset switch. This interface provides for flexible output through an Electronic Assembly DOGS102-6, 102 pixel x 64 pixel, transflective LCD display with a duo-color backlight. The board also contains an independent "signal" LED.

This system interfaces with the RF module through a board-to-board, mezzanine connector using a firmware-defined SPI. The MCU is programmed through an edge connector interface to USB. The power can be supplied through either the edge connector or a wired battery connection. The RF module has an MMCX connector, to which a 6" cable and a reduced-height Linx antenna may be attached, providing a small form factor assembly. The entire system has been designed to fit within a box enclosure, conveniently sized for handheld use (**Figure 1**).

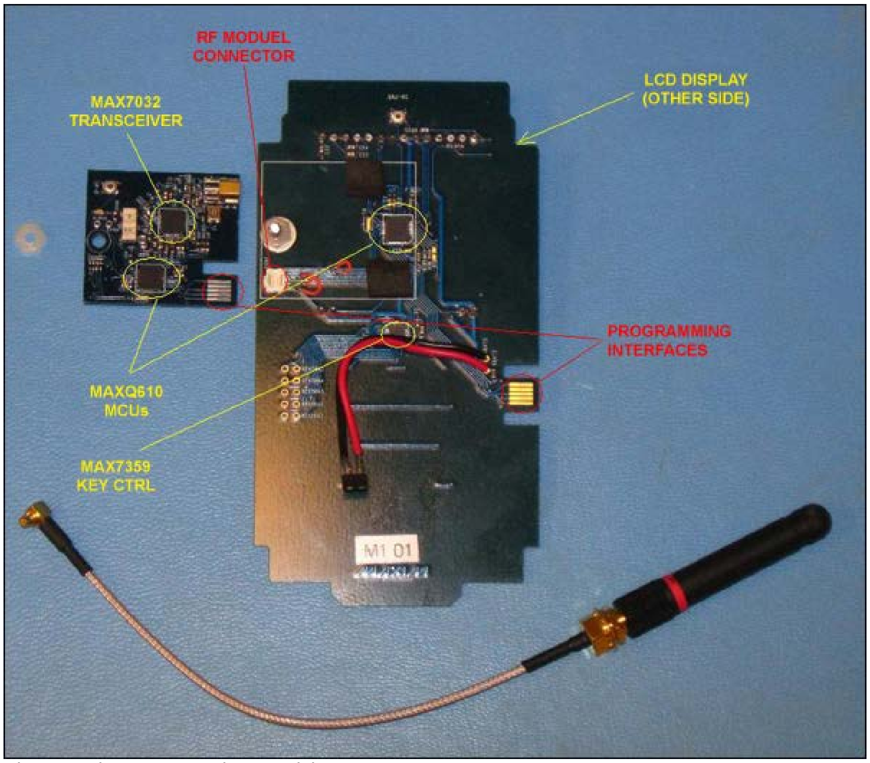


Figure 1. Phase I HHI and RF module.

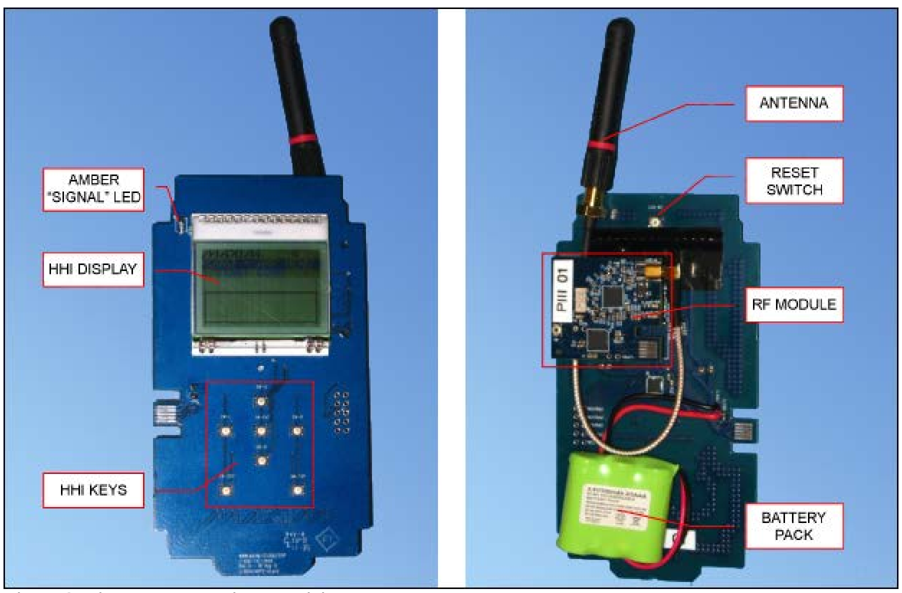


Figure 2. Phase III HHI and RF module.

Similar to the Phase I hardware, the Phase III HHI is a two-board system. The only changes in the HHI board between Phase I and Phase III is the use of a 60-pin hermaphroditic connector from Samtec in place of the 10-pin Hirose connector. This connector utilizes the same SPI communication lines between the microcontroller on the HHI and the microcontroller on the RF module. A few additional power lines are available in the larger connector but a majority of the additional signal pins are not used. This 60-pin connector is compatible with the Newport reference platform.

The RF Modules

The RF meter (MTR) modules used in this demonstration kit take one of three forms: they are either a Phase I or Phase III design, and for the

Phase I designs, they are either the SPI or non-SPI versions (**Figure 3**).

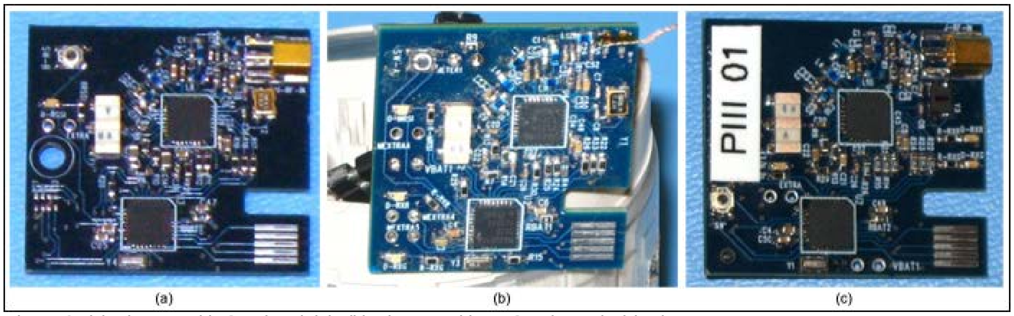


Figure 3. (a) Phase I with SPI (module), (b) Phase I without SPI (meter), (c) Phase III.

Each of these modules contain common components such as the MAX7032 transceiver, the MAXQ610 microcontroller (MCU), various support circuits, a switch, an amber "signal" LED, and a MCU programming-edge connector. The Phase I module with the SPI has a 10-pin Hirose connector on the bottom for communicating with the Phase I HHI. This particular board also has an edge-launch MMCX connector for the antenna. The Phase I module without the SPI connection has two additional LEDs: one red and one green, which are used to provide visual feedback for a simple ECHO or "ping" function. The module also provides a simple footprint for a wire antenna connection, and it has a 2-pin header connection for the physical meter interface. The Phase III board has combined the 60-pin SPI connector and the extra LEDs onto one board, as well as including the MMCX antenna connection.

All three modules are programmed with the same firmware, but with subtle changes to the hard-coded settings. For example, each module is programmed with a different communication address or ID. In the Water Meter Demo Kit, the module that is interfacing with the HHI has been programmed with the address 03 00 00 01 and has been flagged for SPI interfacing. The other "meter" modules have addresses 03 00 00 02, 03 00 00 03, and 03 00 00 04, and they have been hard-coded to operate as a MTR module, which does not use the SPI.

I/Os and Switches

Power to the MTR boards is typically supplied by a 3.6V lithium battery, but can be powered through the JTAG interface or the high-density connector as well. The HHI board is typically powered by a pair of AA batteries and also has the JTAG power provision.

HHI Switch Function Table

Switch	Position	Function	Connection
SW-RST	Momentary	"Reset"	Pin 11, P1.2 of µC
SW-U	Momentary	"^" or "Up"	Pins 12 & 23 of MAX7359
SW-L	Momentary	"<" or "Left"	Pins 13 & 24 of MAX7359
SW-ENT	Momentary	"Enter"	Pins 12 & 24 of MAX7359
SW-R	Momentary	">" or "Right"	Pins 11 & 24 of MAX7359
SW-D	Momentary	"v" or "Down"	Pins 1 & 12 of MAX7359
SW-ESC	Momentary	"Escape"	Pins 1 & 13 of MAX7359
SW-TOP	Momentary	"Top" of menu or Transmit	Pins 1 & 11 of MAX7359

MTR Switch Function Table

Switch	Position	Function	µC Connection
SW or SW'	Momentary	User TX	Pin 11, P1.2

HHI and MTR I/O Edge Connectors

Signal	Description	µC Connection
JTAG-1	TCK — clock	Pin 24, P2.4
JTAG-2	GND — ground	
JTAG-3	TDO — data out	Pin 27, P2.7
JTAG-4	VBAT — external supply	
JTAG-5	TMS — master select	Pin 26, P2.6

JTAG-6	nRST — reset	Pin 28, Reset
JTAG-7	N/A	
JTAG-8	N/A	
JTAG-9	TDI — data in	Pin 25, P2.5
JTAG-10	GND — ground	

MTR LED Indicator Table

LED	Function	µC Connection
D-MRSI, D-RSSI, or D-RSSI'	TX/receive signal indicator/heartbeat	Pin 7, P0.6
D-RXR or D-RXR1		Pin 12, P1.3
D-RXG	(Phase I)	Pin 20, P1.6
D-RXG or D-RXG1	(Phase III)	Pin 8, P0.7

HHI LED Indicator Table

LED	Function	µC Connection
D-RSSI or DLED	Comm Indicator	Pin 7, P0.6
LEDBL1	Red Backlight	Pin 21, P1.7
LEDBL2	Green Backlight	Pin 20, P1.6

Data Frame Structure

The basic structure of the data frame is ASK-modulated, Manchester-encoded, and 4.8kbps (0.2083ms/bit). It also has 144 bits per frame (18 bytes or nine 2-byte words), 30ms per frame, a pause of 70ms between frames, and three frame transmissions per burst. For information on Manchester encoding, refer to application note 3435, "Manchester Data Encoding for Radio Communications."

This structure is directly compatible with other reference design communication formats and utilities are provided in the HHI system to work directly with those other reference designs. Appendix I further describes each section of the remote keyless entry (RKE) frame structure.

Frame structure

Preamble				ID				Function		Data		Sync		Bat	Sig	Chk Sum	
FF	FF	FF	FD	03	00	00	00	00	01	00	00	43	21	11	22	01	68

The structure of this frame is arbitrary, but provides an example of the information that can be contained in any frame related to the many industrial, scientific, and medical (ISM) RF applications.

ID Structure

ID			
03	00	00	01

This design has been preprogrammed to use a 0x03 00 00 01 identification code in the HHI radio module, with the last byte being adjusted between the various MTR systems (typically 0x02, 0x03, and 0x04). This allows the HHI to communicate with each of the MTR boards separately by addressing them one at a time.

Function Structure

Function	
FF	00

This reference design uses the Function field to communicate "group" and "individual" functions to the various MTRs within reception of the HHI system. Again, this structure can be modified to suit the purposes of the user. In this application, the first byte of the Function field is used for high-level commands such as ATTN and CLOSE. The second byte of the Function field is used for various hand-shaking or command operations. **Table 1** describes the various functions and their hexadecimal values as seen on the display and within the firmware:

Table 1. Functions and Hexadecimal Values Within the LFRD003 Firmware

Function	Frame Value$_h$	Source

	Func[0]	Func[1]	
ATTN	FF	XX	HHI module
REQ	00	8P	HHI module
DAT	00	1P	MTR
SET	00	1P	HHI module
CLR	00	2P	HHI module
ACK	00	4P	HHI module
CLOSE	AA	XX	HHI module
ECHO	00	00	Node
Red (A)	00	01	RKE TX
Green (B)	00	02	RKE TX
Blue (C)	00	04	RKE TX
Amber (D)	00	08	RKE TX

P = Addressed port number, X = don't care

These values are an arbitrary definition for the structure of the Func field, which can always be modified by the user.

Data Structure	
Data	
00	00

The Data section of this frame is provided for transmitting information such as a temperature, pressure, or volume measurement. In this design, the Data field tallies pulse outputs from a water meter. Again, the use of this Data value is arbitrary and can be modified by the user.

Communication Protocol

The basic structure defines the interaction of the HHI system and one RF module connected to a physical water meter system.

To conserve power, the MTR systems are configured to operate with a predefined OFF/ON or sleep/listen (S/L) duty cycle. In this demo system, after power is applied to the MTR board, it acknowledges startup with four flashes of the LED, then enters into a 3.0s sleep/0.5s listen cycle. During the ON cycle, the radio is configured for RX mode, and the system will listen for any in-frequency ASK broadcasts. If a valid ATTN frame is received, the MTR system will suspend its S/L cycle and enter into a full wake state.

All communication is initiated with a broadcast from the HHI system. Since the MTRs are presumably in a 3.0s/0.5s S/L cycle, the HHI begins with a 5s transmission of Attention (ATTN) frames to awaken all MTRs within range. The ATTN frame is defined as Don't Care values for all of the fields, except for the checksum and the first byte of the Function Structure, which is set to 0xFF00.

After issuing the ATTN signal, the HHI module will communicate with all of the MTRs listed in the system. In this case only one MTR is predefined in the system, with one port set up for reading (MTR address 0x03 00 00 03). Port 0 on the MTR is configured to act as the pulse-tally port. The communication occurs as a batch process (only the one MTRs listed is contacted).

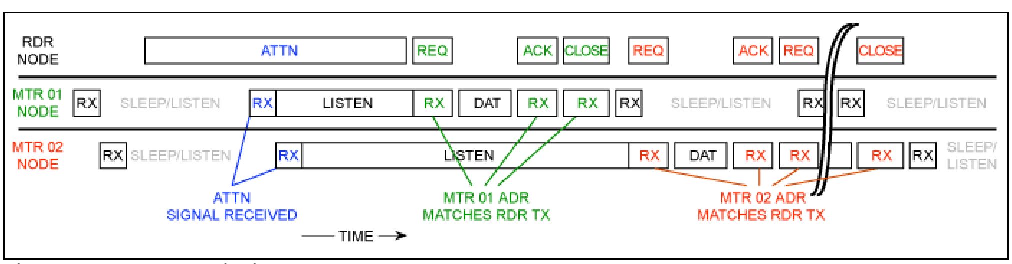


Figure 4. A MTR communication sequence.

A full communication sequence (**Figure 4**) takes the form of the HHI radio module sending a Request (REQ) frame burst followed by a short pause, and then switching to RX mode. The MTR with the matching ID will decode the REQ frame, prepare the data, and switch to TX mode. The MTR transmits a Data (DAT) frame to the HHI, then switches back to RX mode. The HHI radio module receives the DAT frame, and then switches to TX mode and sends an Acknowledge (ACK) frame to the MTR. At this point, the HHI system can either transmit another REQ frame (starting another communication sequence, possibly addressing a different port on the MTR) or it can close the communication session with that MTR. After all of the measurement ports of interest have been read from that MTR, the session with that ID is terminated by sending a Close (CLOSE) frame to the MTR. After a MTR has received the CLOSE frame, it will revert to the S/L cycle until it receives a new ATTN frame.

Handling Lost Frames and Errors

After the HHI radio module receives the requested information from the targeted MTR, it returns an ACK frame. Since this exchange consists of three functional commands, both ends of the communication can be confirmed. The HHI will recognize if the REQ frame was not received properly when the HHI does not receive a DAT frame response; if the HHI did not receive the DAT frame, the MTR will not see an ACK frame response. This process allows for retransmission of the REQ frame by the HHI (if no DAT frame is received) and similarly, the DAT frame can be retransmitted by the MTR (if no ACK frame is received).

- It should be assumed that the HHI radio module has the potential for a stronger TX signal than the MTR.
- If a MTR receives an ATTN signal but no other communication (within 30s), it reverts to its sleep/listen cycle.
- If a HHI does not receive the DAT frame (after 1s), it will repeat a REQ frame (two more times). If the DAT frame is still not received, the MTR should be skipped. In this case, the MTR will be flagged as NON-COMM.
- If a MTR sends a DAT frame but does not receive an ACK frame (within 1s), it will repeat the DAT frame (two more times). In this case, assume the session was closed and revert to its sleep/listen cycle.

This process will assure a minimum number of reattempts to communicate. If the three-burst frame is not sufficient and the multiple attempts also fail, both units will revert to their default state (sleep/listen for the MTR, and NON-COMM for the HHI). This lost frame process is not currently implemented.

ECHO Function

With each of the MTR modules, the microcontroller is programmed to send an ECHO frame when the on-board switch is closed. This process will elicit a response from the matching radio module at the other end—the "match" is defined by the destination ID stored in each module. For example, the internal source ID for the water meter module (WMM) is 03 00 00 03 and its destination ID is 03 00 00 01 (the HHI module). So when the switch is depressed on the WMM, the system will initiate an ECHO frame transmission to the HHI module. The HHI module will in turn respond with an Acknowledgement (ACK) frame.

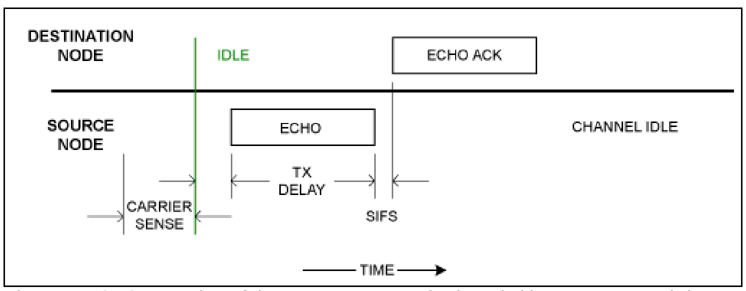


Figure 5. ECHO operation of the MAXBee protocol—the primitive AMR protocol does not perform carrier sensing.

When the WMM receives the ACK frame, it will indicate a valid frame by lighting the green LED for about 2s. If the WMM does not receive a valid ACK frame within about 0.5s of transmitting the ECHO frame, the module will light the red LED instead. Each of the three "meter" modules have 03 00 00 01 stored as their destination ID, so each of those modules will act as an ECHO source, using the HHI as the destination.

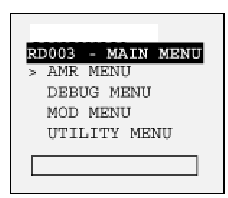
Because of this ECHO structure, this routine can be used as a simplified range test. Starting with the two matched radios near each other, a user can slowly separate them, pressing the button as the radios are moved apart. As the meter module is moved, every press of the button confirms the two-way radio link. Thus the basic operating range can be determined by separating the radios until the green LED no longer comes on.

Handheld Interface Menu System

The HHI operates with a basic user interface structure using a simple menu system and a few navigation keys. To browse through the menu system, there are seven keys: four directional arrows to scroll up (^) and down (v), right (>) and left (<), the ENTER key, the ESC key, and a TOP key. The menu interface indicates the selectable line with a > character (cursor) found in the left-most column of the display.

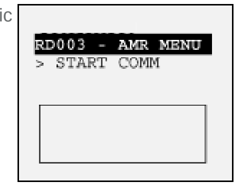
Main Menu

From the Main Menu, the user can navigate to the four primary functions of the HHI unit: the AMR functions, the Debug functions (formerly the RKE menu in the LFRD002 design), the Mod functions, and the Utility functions.



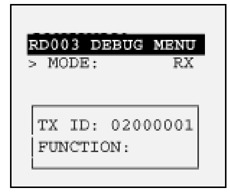
AMR Menu

The primary menu of interest for the LFRD003 water meter AMR reference design is the AMR Menu. From here, a basic water meter reading can be initiated and the read back values will be displayed.



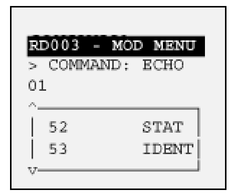
Debug Menu

This menu provides a number of user-oriented functions and is directly compatible with other reference designs. The mode can be used to switch the HHI module between transmit (TX) and receive (RX) modes of the MAX7032 transceiver. This mode will recognize any incoming "Primitive AMR" frames from other systems. If an incoming frame is recognized, this menu will automatically appear (even from a sleep state) and will display both the destination ID contained in the frame itself (TX ID), and the function value found in the ASK frame. This menu can be helpful in debugging transmitters used in the LFRD002 and LFRD003 reference designs.



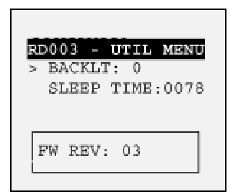
Module Menu

This interface allows for non-RF communication between the microcontroller on the HHI and the attached RF module MCU via the SPI connection. This is the primary debug window used for testing the SPI commands. The scrolling buttons (up and down keys) allow the user to select a few limited commands and execute them. Currently the list shows these commands, with the ones in brackets not performing any function: LED, ECHO, STAT, IDENT, REV. See Appendix V for additional information on the SPI interface.



Utility Menu

The utility menu available in the HHI system allows for basic setting and interaction of the user with internal microcontroller values. This menu shows the default backlight color (0 = none, 1 = red, 2 = green [default], and 3 = amber), the sleep time setting (adjustable in 30s increments with 120s as the default), and the firmware revision of the HHI unit itself.



Firmware Structure

MTR and HHI RF Modules

Functional operation of the MTR and HHI radio modules are almost identical. The MAXQ610 provides a number of inputs to the MAX7032, such as SCLK, DIO, and CS for the SPI interface and DATA, ENABLE, TR, and RSSI for various other RF controls. The purpose of the MTR unit is twofold: first is for the microcontroller to act as a manager and collect data from the meter system; second is to control the radio and communicate the data upon request. One user input is available to force the MTR unit to transmit a basic ECHO frame. The MAXQ610 and the MAX7032 are configured to be in a "stop mode" unless one of three interrupts occur: a wakeup command from the timer, a switch press from the user, or an incoming edge from the meter port. The last two items will cause an external interrupt to be triggered, whereas the wakeup timer generates an internal interrupt.

The interrupt is serviced by the microcontroller, which decodes the source of the interrupt and then takes the appropriate action. A switch press event will cause the microcontroller to go directly into transmit mode and send an ECHO frame. A positive edge on the meter port is accumulated in a counter and stored in memory until the information is requested. A wakeup command is the most complex of the three processes and involves timers, branching decisions, and possibly both modes (RX and TX) for the MAX7032. See Appendix II for the MTR firmware code.

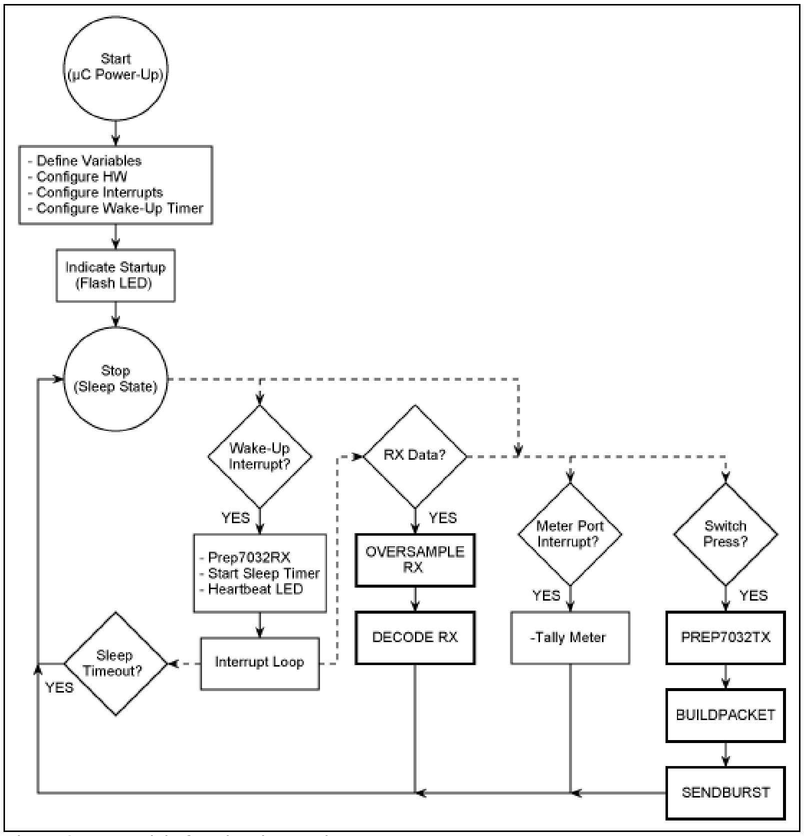


Figure 6. RF module functional operation.

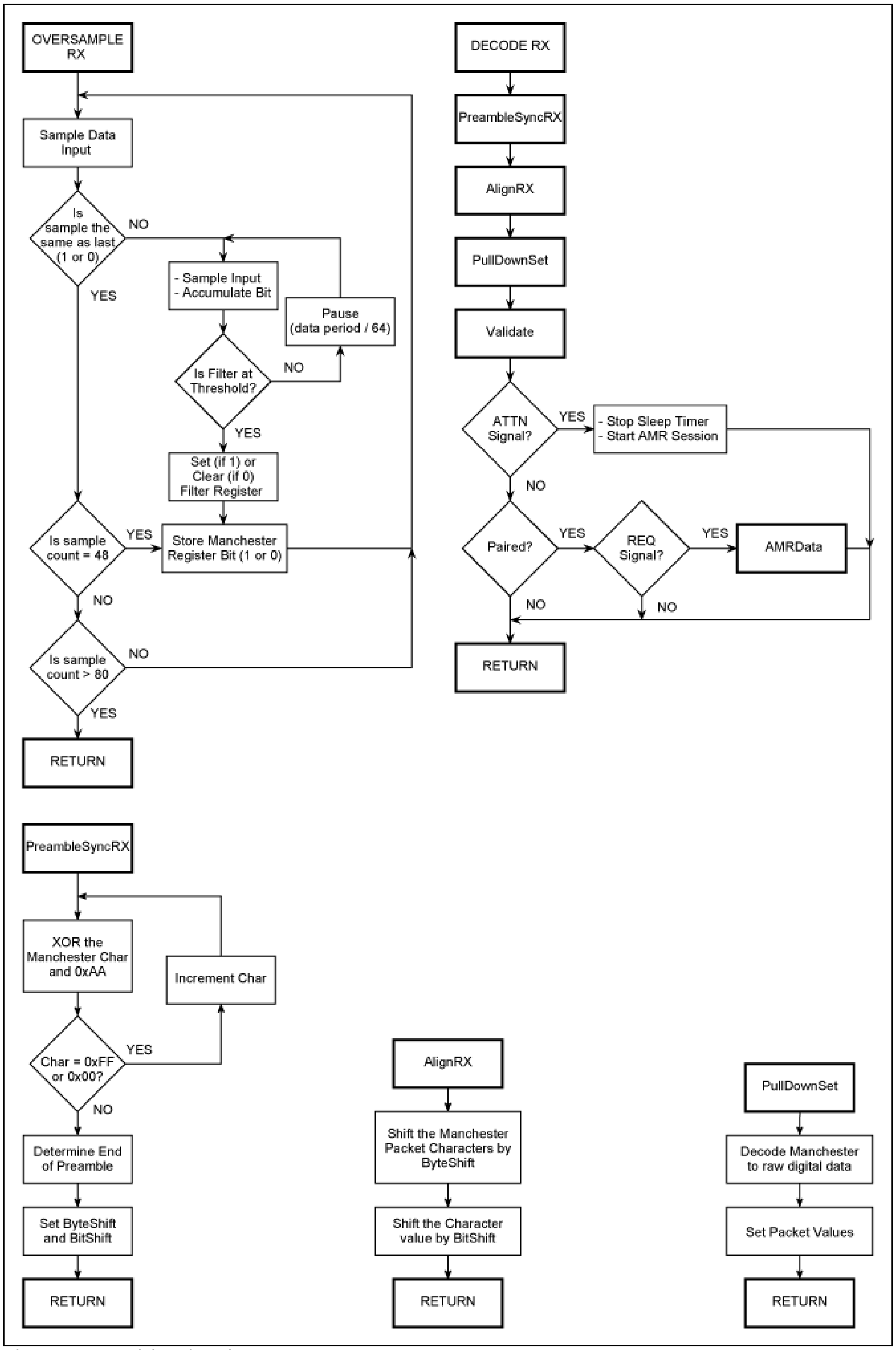


Figure 7. RF module subroutines.

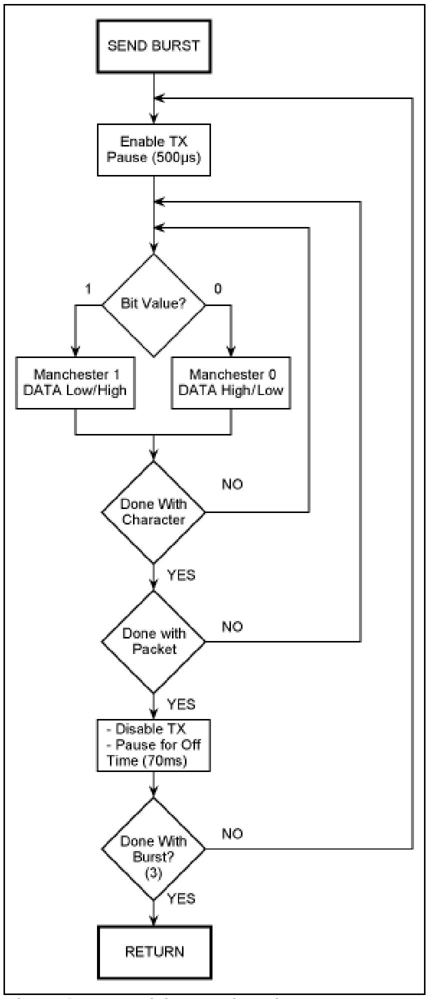


Figure 8. RF module Tx subroutine.

HHI Unit

The HHI serves as an interface between the user and an RF module that is plugged into the back of the unit. This interface uses SPI to transfer instructions, addresses, and data between the HHI and that radio module. In addition to the operational aspects of wired communication with the attached radio module, the MAXQ610 microcontroller has a great deal of user interface functionality. The HHI unit acts as the master in the SPI communication process, initiating the command and request sessions based on user input or an interrupt from the radio module.

Human interfacing is provided through the Electronic Assembly DOGS102-6 pixel display, along with the backlighting, an RSI LED, and the menu keys. The display operates with an SPI interface and thus shares the SCLK and DIO lines with the attached RF module as well as the MAX7359. The key controller operates with an I²C interface and shares the clock and data lines from the SPI devices. It also provides an interrupt signal to the microcontroller.

The interrupts are serviced by first determining the source of the interrupt, then operating with that source appropriately. A key press event will cause the microcontroller to "pop" the MAX7359 stack, and a switch press will wait for about 4s and reset the HHI system. An incoming transmission will be decoded by the attached RF module needed and may be acted upon immediately (such as returning an ACK frame), then an interrupt will be sent to the HHI requesting action. If the attached radio module is configured in RX mode for either an AMR session or RKE debug use, the HHI will display the incoming frame data for interpretation by the user. See Appendix III for the HHI firmware code.

Microcontroller Programming SW

The firmware in this reference design was developed within the IAR Embedded Workbench® software. A full version of this software (4k KickStart Edition) can be obtained at www.iar.com/. The IAR EW works in concert with the MAXQUSB-JTAG interface and the programming adapter, to flash the MAXQ610 on both the TX and RX boards.

Installation

Please refer to the IAR EW documentation for installation and guidance. The firmware in this projected was developed with MAXQ plug-in IAR EWMAXQ2.20I.

Operation

Be sure to have the USB port properly configured within the IAR EW: Project ⟶ Options... General Options — Debugger — JTAG. The COM Port should be set to match the "USB Serial Port (COM XX)" as indicated in the Windows® Device Manager.

For best performance during programming and debug of the MAXQ610, we suggest adjusting the Advanced Settings for the COM port (Device Manager ⟶ USB Serial Port (COM XX) Properties ⟶ Port Settings ⟶ Advanced...). Recommended values for receive and transmit buffer sizes are 512 bytes and a recommended latency timer of 4ms should provide optimal operation.

The USBJTAG translator (MAXQJTAG-USB) board uses a FTDI UART and a MAXQ2000 to convert the PC's serial port into a JTAG port. This board's firmware should be checked for the latest version. To determine the firmware revision, connect the USBJTAG board to a USB port; open the Microcontroller Tool Kit (MTK2) software; select Dumb Terminal in the Select Device window during startup (click OK); choose Options ⟶ Configure Serial Port and select the appropriate COM port and choose the 115200 Speed (click OK); choose Target ⟶ Open COMXX...; hit Enter a few times; type q and hit Enter. The terminal should read back the firmware revision as "83."

A JTAG interface adapter (JIA) must be used to program these systems because of the different supply levels. The USB port provides a 5V power supply that must be regulated down to the 3.3V level needed on the LFRD003 boards. One of these JIA boards is included in the LFRD003 kit.

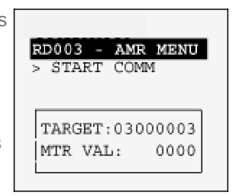
The edge connector should be oriented with JTAG pin 1 on the top of the MTR boards. The system can be programmed with batteries in place or without batteries installed (3.3V regulated USB power is used). Within the C code, there are some lines that can be uncommented to help with debugging firmware changes.

Operational Setup and Use

AMR Functionality

Water Meter Reading

Currently the HHI unit and the attached RF module are hard-coded to interact with only one of the remote MTR modules (target ID: 03 00 00 03). To communicate with that water meter module, select AMR MENU from the "RD003 - MAIN MENU" list. The HHI will now show "RD003 - AMR MENU" in its default state. Since there is only one selection on this menu (START COMM), pressing ENTER will initiate communication with the WMM. The HHI will send the REQ (0x22) command to the connected RF module over the wired SPI, which will in turn send out the wireless REQ frame to the destination meter. When the WMM receives this REQ command, it will respond with a DAT frame transmission, which is addressed to the origination source (in this case 03 00 00 01). Contained in the DAT frame is the "tally" of the WMM input, displayed in the AMR Menu as the MTR VAL, in hexadecimal form.



Suggested Demonstration Routines

To make the most efficient use of this water meter AMR reference design, try to follow a format similar to this staged routine.

Preparation:

1. Assemble the water meter module as directed in Appendix IV.
2. Keep a can of compressed air with the physical water meter assembly.
3. Connect a charged battery pack to the HHI unit in the enclosure.
4. Permit the HHI to "boot up." Then go to the AMR Menu and press the Enter key twice (the first press will not result in a TX).
5. Assemble the two additional meter modules by connecting battery packs to each of them.
6. Press the button on each of those meter modules while they are in close proximity to the HHI. Each of them should flash the amber LED indicating transmission, then again when receiving an ECHO ACK frame from the HHI. Finally they should temporarily light the green LED to indicate a valid RF frame exchange.
7. If either of the meter modules show the red LED, then check the range, antenna connections, etc. for indications of damage or interference.

1-Way Communication Range Demonstration:

1. Configure the HHI in a static location so that the antenna is reasonably exposed to the environment, preferably oriented such that the antenna is pointing "up."
2. With the LFRD001 key fob transmitter, install the CR2032 battery (+ side of the battery case in contact with the battery holder).
3. While holding the key fob transmitter, walk a few paces from the HHI and press any of the four buttons.
4. The HHI will indicate an RKE frame was received by displaying the DEBUG MENU page. On this page, the TX ID will be shown along with the FUNCTION (0000, 0001, 0002, or 0004, corresponding to one of the four buttons pressed).
5. The HHI backlight should change to correspond to the button pressed on the key fob (none, red, green, or amber).
6. Repeat Step 3 as desired.

2-Way Communication Range Demonstration:

1. Configure the HHI in a static location so that the antenna is reasonably exposed to the environment, preferably oriented so that the antenna is pointing "up."
2. Hold one of the additional meter modules such that it is easy to press the button, as well as see the red/green indicator LEDs.
3. While holding the additional meter module, walk a few paces from the HHI and press the button. The module will exchange an ECHO frame with the HHI and the green LED will illuminate for about two seconds.
4. Continue to walk away from the HHI while periodically pressing the button on the additional meter module until the red LED is illuminated. This should indicate an approximate range/orientation/environmental condition that is at the limit of two-way communication distance.
5. While the units are in range of each other, the HHI will indicate an ECHO frame was received by displaying the MOD MENU page, the RX CMD will be listed as 50 (ECHO), and VALID should read 02.
6. Repeat Step 3 as desired.

Water Meter Reading Demonstration:

1. Using the HHI, navigate to the AMR MENU page.
2. While standing in reasonable proximity to the WMM, press the ENTER key. The amber "signal" LED will light on the HHI (as well as the amber TX LED on the RF module). Then the AMR Menu page will show the target ID of the WMM and the latest tally reading from the meter.
3. Using the compressed air can, blow air through the water meter (in either direction). The RF module connected to the WMM will flash the amber LED each time the magnet actuates the relay.
4. Repeat Step 2 and observe that the MTR VAL has increased (will be about twice the number of flashes seen while blowing air through the meter. This is shown as a hexadecimal code: 0000 to FFFF).
5. Press ENTER again (as in Step 2) without actuating the water meter to demonstrate that the value does not change based on just the RF reading process.
6. Repeat combinations of Steps 2 and 3 to demonstrate the changing values of the water meter measurement.
7. To reset the meter value, remove and replace the battery connection on the WMM (this can also be used to confirm communication between the units).

After a predetermined length of time, the HHI will go into sleep mode (the display goes blank and any backlight is turned off). The HHI can be woken from this mode either by receiving an incoming transmission (one-way or two-way frames) or by pressing any key on the HHI.

Range

The predicted range in a flat unobstructed outdoor area is based on the following assumptions:

f_0 = 433.92MHz
P_{PA} = +10dBm
G_T = -18dBi (small loop antenna typical -18dBi)
h_{TX} = 1m
h_{RX} = 1m
G_R = 4.14dBi (ideal 1/4λ antenna = 5.14dBi)
L_{ConR1} = -0.57dB
Path loss varies as R^{-4} because of ground bounce interference
RX sensitivity set at -114dBm

The calculated estimate of "open field" range is approximately 370m (see application note 5142, "Radio Link-Budget Calculations for ISM-RF Products" for more information).

Indoor range testing resulted in a consistent useable distance of 30m using a Linx reduced-height antenna connected to the HHI; 35m was achieved with a 1/4λ antenna on the MTR with the MTR unit placed ~2m above the floor (cube wall); 30m was also reached in a lab environment with the MTR positioned 1m above the floor (bench top).

Battery Usage Analysis

Microcontroller

[1.8V nominal core voltage, 1.0V RAM (min) data retention/power-on-reset voltage]
The MAXQ610 microcontroller burns a maximum 12µA (with power-fail off) during "stop" mode.
The MAXQ610 microcontroller (with 12MHz SysClk) burns a maximum of 5.1mA during normal operation.

TRX

[2.1V to 3.6V operation]
The MAX7032 transceiver burns a maximum 8.8µA (3V, 85°C) during sleep mode.
The MAX7032 transceiver during TX operation burns a typical 12.4mA at 434MHz (max 20.4mA) with "always on"; when running at 50% duty cycle at 434MHz, it burns a typical 8.4mA (ASK) and a maximum of 13.6mA.
The MAX7032 transceiver typically burns 6.4mA at 434MHz, ASK (3V, 85°C) with a maximum of 8.3mA during RX operation.

Key Controller

[1.62V to 3.6V operation]
The MAX7359 key switch controller burns a maximum 5µA during sleep mode.
The MAX7359 key switch controller burns typical 25µA, maximum 60µA; with one key pressed the controller burns a typical 45µA during normal operation.

MTR Modules

LED: configured with 75Ω limiting resistors, yellow (V_F = 2.2), nominal 10.7mA on current.

The average MTR system sleep current was measured to be 233µA. If left in sleep mode consistently, using a 750mAh, CR2 battery, the MTR unit would last about 134 days. For the sleep/listen duty cycle of 14.3% (3.0s sleep/0.5s listen), with the microcontroller and MAX7032 cycling, the measured current averaged a maximum of 11.6mA. Using the CR2 battery, the typical MTR life (without any TX communication) would be approximately 403hrs [11.6mA × 0.143 + 0.233mA × 0.857 = 1.859mA] or about 16.8 days.

The average maximum current during communications (TX/RX with the HHI unit) was approximately 15.62mA. If a 10s communication session was held once a day, the overall life of the CR2 battery would be reduced by 0.73mA/day—a negligible amount of time.

This system has not yet been optimized for low-current "stop" mode.

HHI Unit

In addition to the same radio configuration as above, the HHI unit also has a MAXQ610, the MAX7359 used for key inputs, and the DOGS serial interface display with backlighting. A standby mode is implemented but the current consumption of the HHI unit has not been measured under varying conditions.

This system has not yet been optimized for mid-low current "standby" mode or low-current "stop" mode.

Hardware Details

Transceiver Specifications

Supply current (I_{DD}) at f_{RF} = 433MHz, TX 50% duty cycle	8.4mA (typ)
Deep-sleep supply current (I_{DD}) at 3V	8.8µA (max)
Output power (P_{OUT}) into 50Ω	+10.0dBm (typ)
Sensitivity (average power level)	-113dBm (typ)

Component List

The following table provides a list of components used to populate both the MTR boards (Phase III) and the HHI unit (Phase III). Maxim recommends high-quality, wire-wound inductors for components used on the RF boards.

RF Module Board

Designation	Qty	Description
C48-49	2	CAP, 0.01µF, 10%
C21-22	2	CAP, 0.047µF, 10%
C5-6, C50	3	CAP, 0.1µF, 10%
C4	1	CAP, 1.0µF, 10%
C2	1	CAP, 1.8pF, 5%
C1, C7-10	5	CAP, 100pF, 5%
C29	1	CAP, 10pF, 5%
C20	1	CAP, 1500pF, 5%
C3, C26, C31, C33-34, C45	6	CAP, 220pF, 10%
C27	1	CAP, 470pF, 10%
C28, C30	1	CAP, 6.8pF, 5%
C51	1	CAP, 680pF, 10%
Y1	1	CERAMIC-SMD, 12.000MHz
F1	1	FLT\MURATA\SFTLA10M7FA00-B0, 10.7MHz
L4	1	IND-MOLDED, 10nH, 5%

L2	1	IND-MOLDED, 20nH, 5%
L9-11	3	IND-MOLDED, 22nH, 5%
L1	1	IND-MOLDED, 68nH, 5%
D-RSSI'	1	LED-1, Amber
D-RXG	1	LED-1, Green
D-RXR	1	LED-1, Red
U4	1	MAX7032
U2	1	MAXQ610A-0000+
R1, RBAT2	2	RES, 0Ω
R14-15, R22-26	7	RES, 100Ω
R9, R13	2	RES, 100kΩ
R15	1	RES, 10kΩ
R7	1	RES, 1MΩ
R-RXG, R-RXR, RRSSI'	3	RES, 75Ω
J-RF-IN'	1	Emerson MMCX Jack
SW'	1	SW-SPST-NO-B, SPST NO
Y3	1	XTAL-SMD, 17.63416Mhz

HHI Board

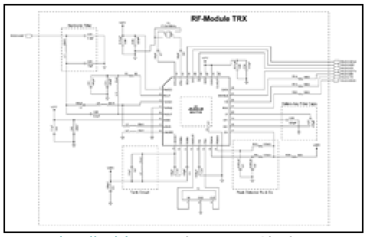
Designation	Qty	Description
	1	BOX, Series 55 enclosure
C24-25, C53-56	6	CAP, 0.01µF, 10%
C23	1	CAP, 1.0µF, 10%
Y4	1	CERAMIC-SMD, 12.000MHz
U6	1	DISPLAY/DOGS102-6
DLED	1	LED-1
U5	1	MAX7359ETG+
U3	1	MAXQ610A-0000+
R5, RBAT2	2	RES, 0Ω
R1, R17-20	5	RES, 100Ω
R2	1	RES, 4.7kΩ
RLED	1	RES, 75Ω
CON1	1	SAMTEC-LSEM-130-04.0-L-DV-A-N-K-TR
SW-D, SW-ENT, SW-ESC, SW-L, SW-R, SW-RST, SW-TOP, SW-U	8	SW-SPST-NO-B, SPST NO

Schematics

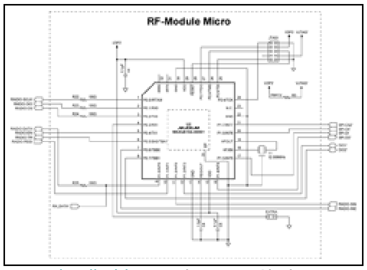
(Revision B1: detailed 11"x 17" copy available here)

RF Module Blocks

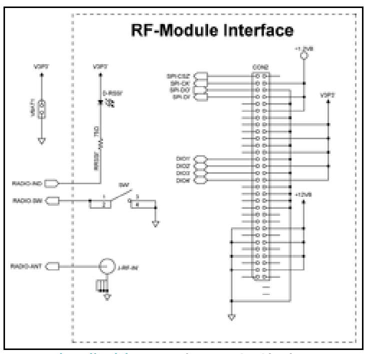
(Revision A1: detailed 11"x 17" copy available here)



More detailed image (PDF, 443kB)

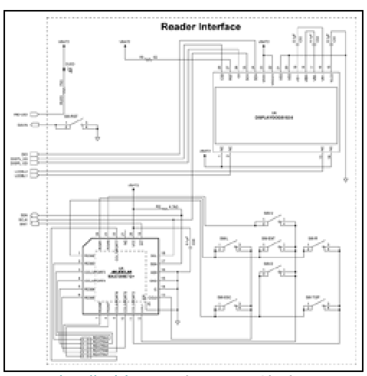


More detailed image (PDF, 412kB)

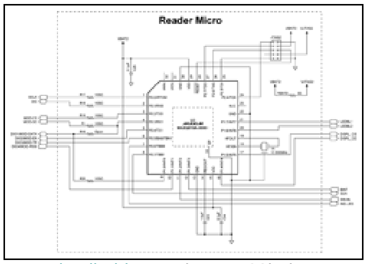


More detailed image (PDF, 376kB)

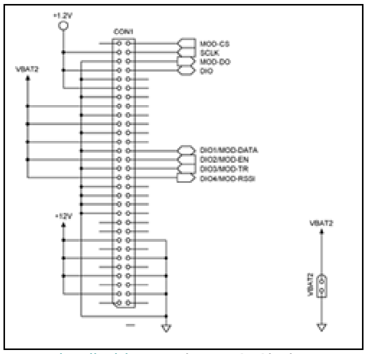
HHI Blocks



More detailed image (PDF, 440kB)



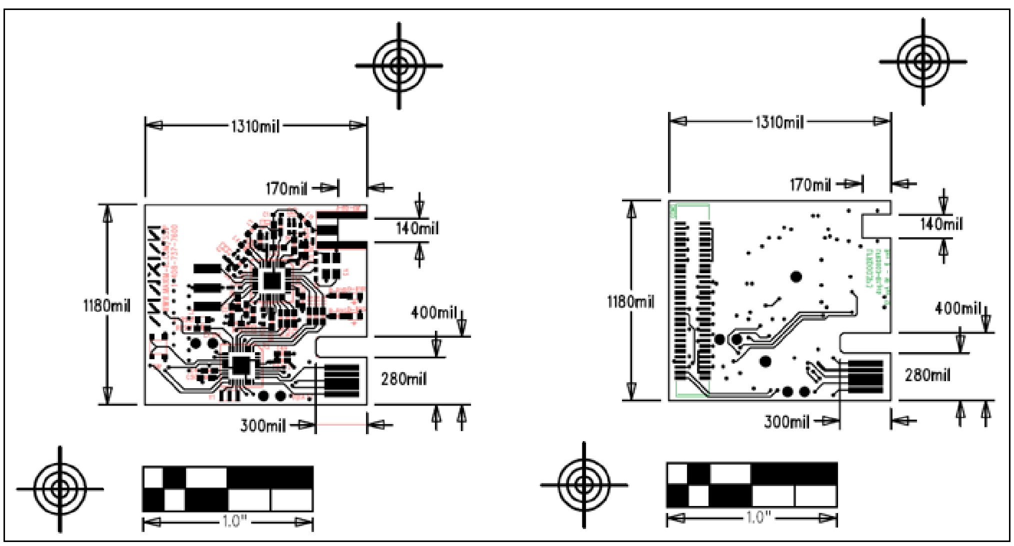
More detailed image (PDF, 408kB)



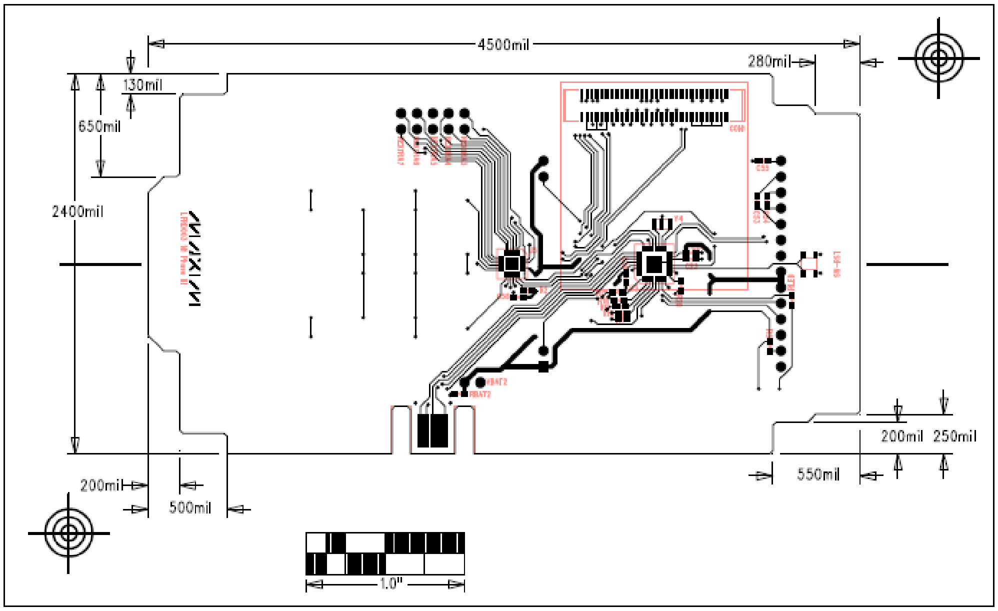
More detailed image (PDF, 272kB)

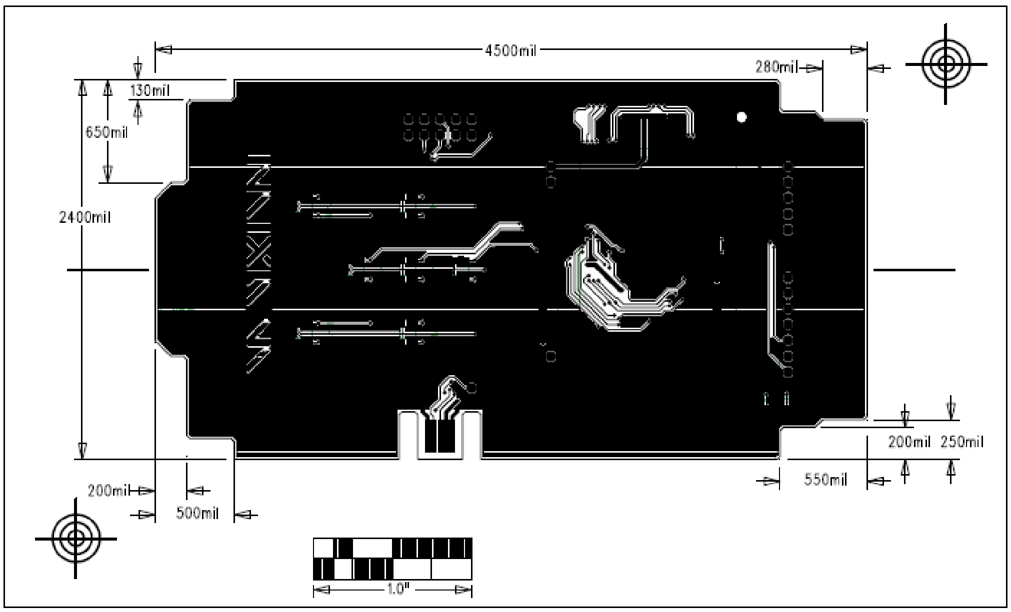
Layout

RF Module



HHI





References

1. MAXQ610 data sheet
2. MAXQ Family User's Guide
3. MAXQ610 User's Guide
4. MAXQ610 Revision A3 Errata
5. MAXQ610 evaluation kit
6. MAXQ USB-to-JTAG evaluation kit
7. Application note 4465, "Using the Serial Port on the MAXQ610 Microcontroller"
8. Application note 4314, "Getting Started with the MAXQ610 Evaluation Kit (EV Kit) and the IAR Embedded Workbench"
9. ISM-RF Reference Design 003 RF Module Schematic and Layout
10. ISM-RF Reference Design 003 HHI Unit Schematic and Layout
11. IAR Embedded Workbench IDE User Guide
12. MAXQ IAR Assembler Reference Guide, EWMAXQ_AssemblerReference.pdf
13. MAXQ IAR C Compiler Reference Guide, EWMAXQ_CompilerReference.pdf
14. Electronic Assembly, DOGS displays

Related Application Notes

Application note 2815, "Calculating the Sensitivity of an ASK Receiver"
Application note 3401 "Matching Maxim's 300MHz to 450MHz Transmitters to Small Loop Antennas"
Application note 3435 "Manchester Data Encoding for Radio Communications"
Application note 3671 "Data Slicing Techniques for UHF ASK Receivers"
Application note 4302 "Small Antennas for 300MHz to 450MHz Transmitters"
Application note 4314, "Getting Started with the MAXQ610 Evaluation Kit (EV Kit) and the IAR Embedded Workbench"
Application note 4465, "Using the Serial Port on the MAXQ610 Microcontroller"
Application note 4636 "Avoid PC-Layout "Gotchas" in ISM-RF Products"
Application note 5142, "Radio Link-Budget Calculations for ISM-RF Products"
Application note 5391, "LFRD002 Wireless Automatic Meter Reading Reference Design"

Appendix I: RKE Frame Structure

The basic structure of the data frame is ASK modulated, Manchester encoded, 4.8kbps (0.2083ms/bit), and has 144 bits per frame (18 bytes or 9, 2-byte words), 30ms per frame, a pause of 70ms between frames, and 3 frame transmissions per burst. For information on Manchester encoding, refer to application note 3435, "Manchester Data Encoding for Radio Communications."

Frame Structure

Preamble				ID				Function		Data		Sync		Bat	Sig	Chk Sum	
FF	FF	FF	FD	03	00	00	01	00	01	00	00	43	21	11	22	01	68

The structure of this frame is arbitrary but has been established to provide an example of the information that can be contained in any frame related to the many ISM-RF applications.

Preamble

Preamble			
FF	FF	FF	FD

The default preamble for this reference design is 4 bytes of high data with a pair of stop bits at the end. Since this is transmitted with Manchester encoding, the waveform appears as a 4.8kHz square wave lasting for 6.67ms. Having a preamble longer than roughly 1ms should provide ample time for the Rx system to wake up, given a strong received signal strength. The MAX1473 has a typical wake-up time of 250µs to start receiving valid data. Extra time can be padded onto this initial wake-up time to allow the Rx system to properly settle the baseband slicing circuit, which in turn provides optimum sensitivity. Providing a preamble of over 6ms gives ample opportunity for a relatively weak signal to wake up the receiver.

Assuming the MAXQ610 microcontroller used in the Rx is sitting in a stop mode, the system will consume a certain number of received bits to start up the microcontroller before it can begin decoding the data stream. The nanopower ring oscillator in the MAXQ610 typically runs at 8kHz (the wakeup timer interval can be $1/f_{NANO}$ to $65535/f_{NANO}$). If the Rx system is configured to use the RXSIG or the RXDATA line as an interrupt generator, the µC would have a warm-up time of $8192 \times t_{HFXIN}$. With t_{HFXIN} = 83.3ns (f_{XCLK} = 12MHz clock), the warm-up time works out to approximately 0.6827µs. This is well within the time available in one preamble transmission, and at a baud rate of 4.8kbps, the µC should be in a warmed-up state within 3.28 bits. The FD pattern at the end of the preamble is used to indicate the following start of the data frame and is the key to synchronizing the received bit stream.

ID Structure

ID			
01	23	45	67

The ID section is configured with 4 bytes of identification. This permits 2^{32} unique identifiers or over 4.29 billion codes. If one byte is used for class identification (256 different car models for example), 2^{24} unique identifiers or over 16 million codes remain. The structure can be modified to suit the purpose of the user. This RD has been preprogrammed to use an identification code as noted above, with the last byte being adjusted between RD systems. This allows for simultaneous operation of multiple, independent RKE systems.

Function Structure

Function	
00	01

This simple reference design has only four input switches on the transmitter, so 2 bytes of "function" is overkill. Again, this structure can be modified to suit the purposes of the user. In this application, when button A is pressed, the function value would be 00 01. When button B is pressed, the function value would be 00 02, and so forth. The function value is used to convey information (with individual buttons having their own bit), so it is possible to press multiple buttons simultaneously. In that instance, if buttons B and D were pressed together, the function value would be represented as 00 0A (this multibutton operation has not been implemented in the design). This is an arbitrary definition for the structure of the function value, and it can always be modified by the user.

Data Structure	
Data	
00	00

The data section of this frame is provided for transmitting information, such as a temperature or pressure measurement, a speed indicator, etc. In this design, the data section could work in concert with the function section to convey information whenever a button on the transmitter has been pressed. Again, the use of this data value is arbitrary and can be modified by the user. This operation is not currently implemented in the design.

Sync Structure	
Sync	
43	21

The synchronization block is set up to enable encryption coding. Users can work with this section to provide rolling code sync, a public key, etc. This operation is not currently implemented in the design.

Battery Gauge Indicator
Bat
11

This single byte allows the transmitter to send an indication of battery strength to the receiver. This section of the frame could have value when indicating a need to change the Tx battery. This operation is not currently implemented in the design.

Signal Strength Indicator
Sig
22

A possible use for a transceiver configuration, the received signal strength of the return channel could be shared between the nodes. This operation is not currently implemented in the design.

Checksum	
Chk Sum	
01	67

In this reference design, the checksum is used as a go/no-go gate for valid data. The frame values (except the preamble) are summed up during transmission, one byte at a time, and the full sum is tacked on to the end of the frame as the checksum. This value is compared to a received data stream, and a decision to use or discard the frame is made. The format of this checksum operation is arbitrary, but as long as the Tx/Rx and the encode/decode methods are equivalent, the checksum process will operate as intended.

Appendix II: MTR Firmware

main.h (14 Sep 11, Rev 0.8):

```
/*****************************************************************************
 * Copyright (C) 1999-2012 Maxim Integrated Products, All Rights Reserved.
 *
 * See main.c for additional information.
 *
 *****************************************************************************/
/* Global Constants */

/* ------------------------------------------------------------------------ */

/* Main Subroutines */
  void MasterInt();

/* M1 Subroutines */
  void ExeCmd(unsigned long int ReadVal);
  void SlvRqst();
```

```
/* RX Subroutines */
  void DecodeRX();

/* MAXBee Subroutines */
  void TXEcho();
  void Echo();
  char EchoCheck();

/* Meter Subroutines */
  void MeterIn(unsigned char MTRPort);

  /* TX Subroutines */
  void BuildPacket();
  void BuildMBusFrame();

/* Common Subroutines */
  void Lights8(long int WaitOn, long int WaitOff);
  void Pause(long int Count);
  void WriteFlash(unsigned int Address, unsigned int Data);
  unsigned int ReadFlash(unsigned int Address);
  void EraseFlash(unsigned int Address);
  void GoToSleep();

/* Interrupt Service Routines */
  void ExtISR();
  void SleepISR();
  void WakeUpISR();

/* Common Subroutines */
  void GenCRC(unsigned char const Msg[], unsigned char ByteCnt);
  char CheckCRC(unsigned char const Msg[], unsigned char ByteCnt);

/* AMR Subroutines */
  void AMRData(unsigned char MTRPort);
```

See the latest LFRD003-MTR*.zip file for all the firmware code.

```
main.c
isr.c
global.h
global.c
radioCTRL.h
radioCTRL.c
radioSPI.h
radioSPI.c
slaveSPI.h
slaveSPI.c
TRX.h
TRX.c

debug_maxq61x.s66
iomacro.h
iomaxq.h
iomaxq61x.h
iomaxq610.h

LFRD003.dep
LFRD003P3.dep
LFRD003P3.ewd
LFRD003P3.ewp
LFRD003P3.eww
```

Appendix III: HHI Firmware

main.h (06 Sep 11, Rev 0.3):

```
/*******************************************************************************
 * Copyright (C) 1999-2012 Maxim Integrated Products, All Rights Reserved.
```

```
 *
 * See main.c for additional information
 *
 ******************************************************************************/

/* Main Subroutines */
  void MasterInt();
  void InitRD003M1();
  void InitRD003M1PIII();

/* MAXBee Protocol Subroutines */
  void RFModIntr();
  void RFModSend();

/* Common Subroutines */
  void Lights7();
  extern void Pause(long int Count);
  void WriteFlash(unsigned int Address, unsigned int Data);
  unsigned int ReadFlash(unsigned int Address);
  void EraseFlash(unsigned int Address);
  void GoToSleep();
  void WakeUp();

/* Interrupt Service Routines */
  void ExtISR();
  void SleepISR();
  void WakeUpISR();
```

See the latest LFRD003-HHI*.zip file for all firmware code.

```
main.c
isr.c
global.h
global.c
dispmenu.h
dispmenu.c
keyctrlI2C.h
keyctrlI2C.c
masterSPI.h
masterSPI.c

debug_maxq61x.s66
iomacro.h
iomaxq.h
iomaxq61x.h
iomaxq610.h

LFRD003M1P3.dep
LFRD003M1P3.ewd
LFRD003M1P3.ewp
LFRD003M1P3.eww
```

Appendix IV: The Physical Water Meter

The Metherm/Zenner water meter is a simple impeller driven unit that tracks the flow of fluid in m^3 units (1000 liters). This meter is equipped to accept a remote indicator module, which consists of a simplistic magnetic relay and a two-conductor cable that provide a momentary closed circuit for each revolution of the 1-liter indicator dial. This dial has internal clockwork that contains a magnet on one of the horizontal gears in the gear train. When a magnetic relay is held in close proximity to this dial, the rotating magnet will actuate the relay, resulting in one closure of the relay for every rotation of the 1-liter indicator.

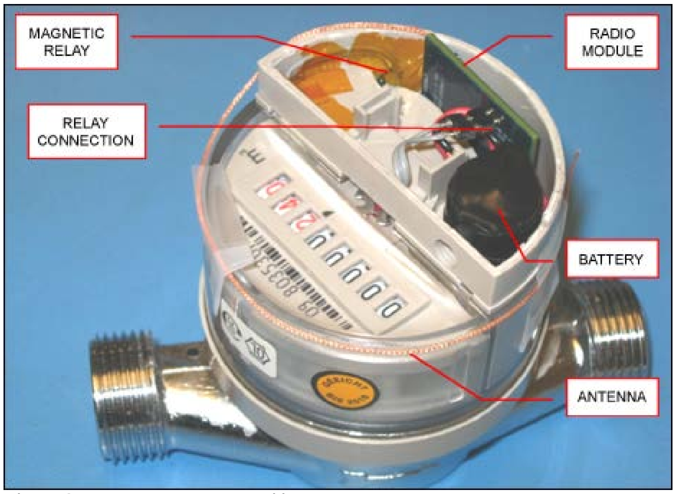


Figure 9. Zenner water meter with WMM.

In this design, one of the Zenner module enclosures was disassembled and a Maxim RF module was inserted into the system. This module provides the capability of tallying the relay closures using the on-board microcontroller (MAXQ610). That result is stored in volatile memory and can be reported back to a base unit when an RF request is received. Nonvolatile (NV) memory is available, but not yet implemented in the reference design.

The radio assembly consists of the remote module enclosure, a magnetic relay, the MAX7032 RF module, and a CR2 (750mAh, lithium) battery. The original relay supplied with the remote module was damaged during disassembly of the module so a different relay was used. This relay has a pair of leads that are wired to a 100mil header pin assembly, which in turn can be connected to the radio module board. The battery has been configured with a pair of wire leads, which also are connected to a 100mil header-pin assembly. Once both of these are "plugged in" to the RF module board, the system will be operational. The RF module board has a soldered antenna that consists of ~17cm of copper braid. The primitive AMR address for this module is hard-coded into the MAXQ610 MCU firmware as 03 00 00 03.

To enhance the magnetic coupling between the 1-liter dial and the relay, a supplemental magnet was placed on top of the existing magnet in the water meter clockwork. It should be noted that the orientation of the supplemental rare-earth magnet can have an effect on this coupling—one side of the cubic magnet was marked with a black permanent marker; this face should be facing "outward" from the gear axis for best performance.

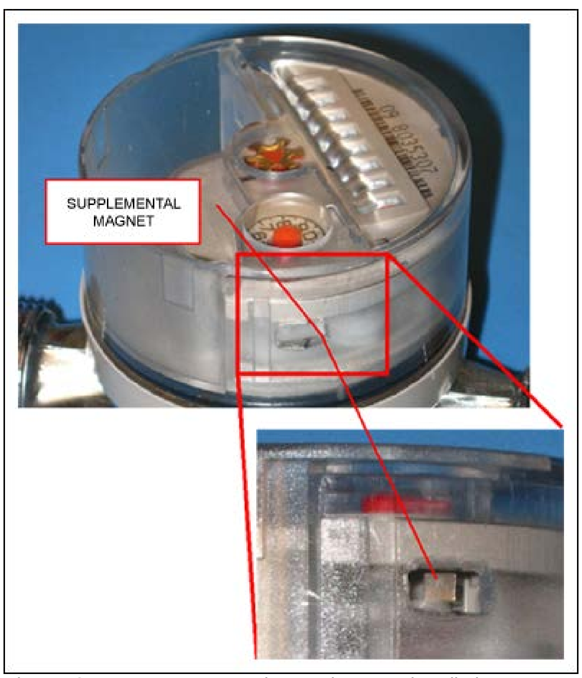


Figure 10. Zenner WMM, supplemental magnet installed.

The water meter has been packaged without the radio module installed and the battery has been shipped with electrical tape over the leads. To assemble the system for operation, locate the appropriate radio module from the LFRD003 demo kit box (marked with a "PI 03" sticker). **Use caution when handling the RF module to avoid any sheer forces on the antenna connection.** There is a clear plastic "lid" taped to the top of the enclosure, and it should be removed or flapped open. Remove the electrical tape from the battery leads and connect the battery header to the wired header on the RF module such that the red wires are mated (the battery has a dark blue wire that should match the black wire of the RF module). Connect the relay to the on-board header of the RF module (no polarity necessary). Gently place the module into the enclosure, taking caution to not disturb the relay and to not stress the antenna connection at the board. After seating the module into the enclosure, wrap the copper braid antenna around the top rim of the plastic water meter (**Figure 11**).

Figure 11. WMM connections.

The Measurement Process

As water (or in the demonstration case, compressed air) flows through the meter, the impeller will turn the clockwork gears of the meter. As noted earlier, this will slowly rotate a magnet past a relay, causing that relay to close and open with every liter of indicated water flow. The RF module senses this closure and responds by tallying the event. In essence, every liter of water that flows through the water meter can be counted. The water meter module tends to tally two counts for every closure (and thus every "liter of water").

Appendix V: Newport Application Programming Interface

Hardware Interface

In the Phase I design, a Hirose DF12 Series, 10-pin mezzanine connector was used as a simple board-to-board interface between the RF module and the HHI. Later, the Newport reference platform defined the Samtec LSEM Series, 60-pin hermaphroditic connector for use with the home area network (HAN) interface as well as the neighborhood area network (NAN) interface. This connector was adopted for the Phase III RF module design. In both instances, the connector provides interfacing between the master unit (either the HHI or Newport) and the RF module through a wired SPI.

- **3-Wire SPI**
  The standard interface to the RF module is through a 3-wire SPI. The HHI acts as the master of the SPI interface and the RF module is the slave. The HHI drives the CSB and SCLK lines while using the SDIO line as a two-way data interface. CSB is seen as an interrupt line on the RF Module.
- **Interrupt**
  The RF module requests an interaction with the HHI if there has been an autonomous broadcast (or other incoming communication) by asserting the interrupt line. Once the HHI is ready to service the interrupt, communication with the RF module occurs in the same manner as a standard SPI command.
- **Timing**
  Currently untested. Minor adjustments should be feasible given the built-in delays in the firmware. Characterization and adjustments are not currently planned.

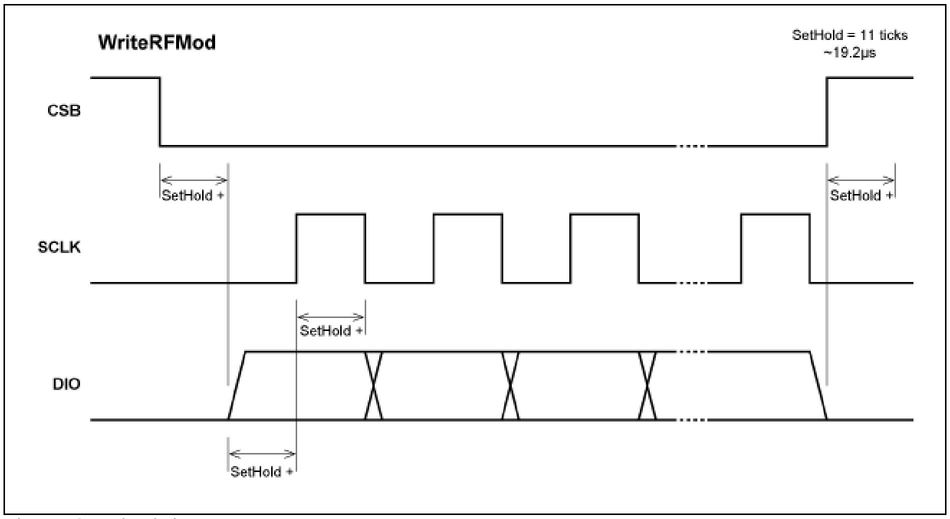


Figure 12. Write timing.

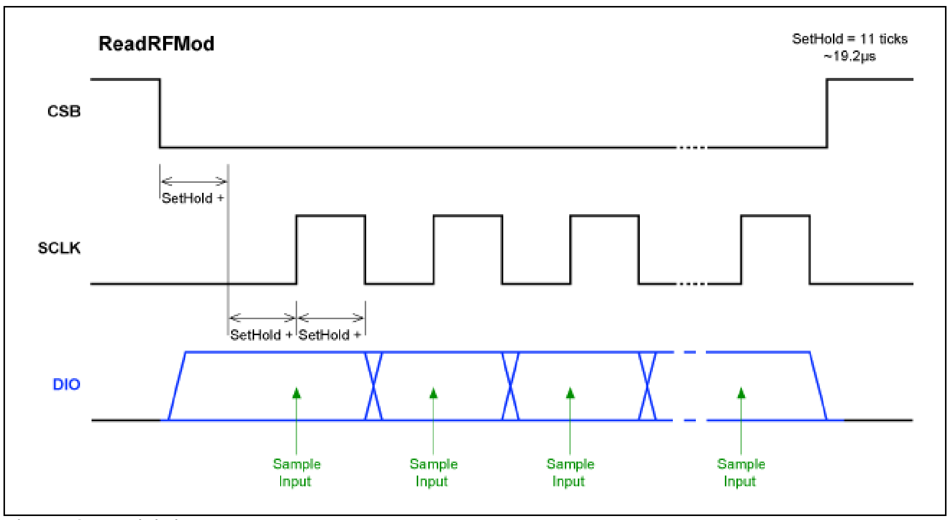


Figure 13. Read timing.

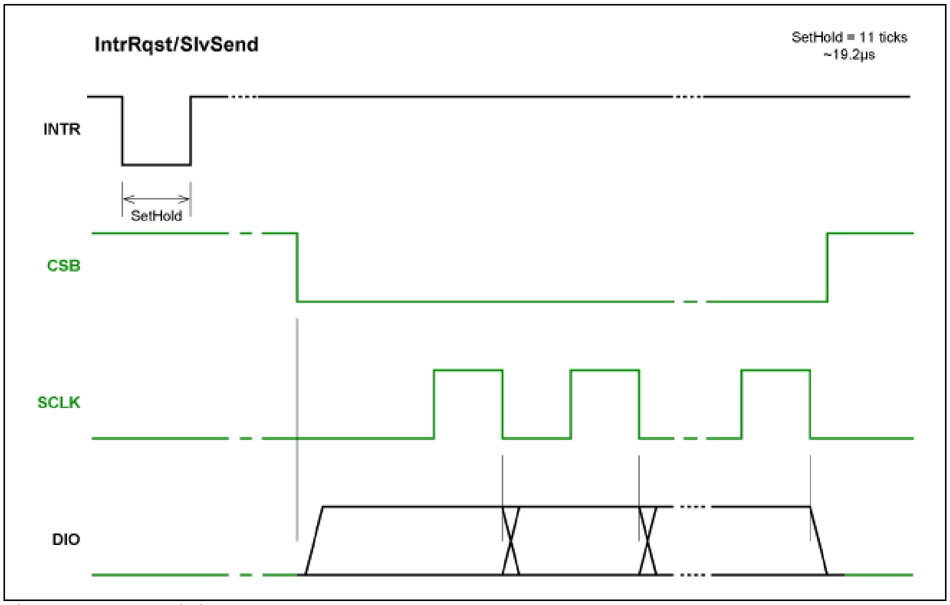


Figure 14. Interrupt timing.

Firmware Interface

Three-Byte Format: CMD/ADR/DAT

The basic structure of an outgoing command (HHI to RF module) takes the form of a 3-byte string. The first or most significant byte (MSB) is the command (CMD), the second byte is an address (ADR), and the final byte (least significant or LSB) is the associate data (DAT).

Module Command List

SPI Commands from HHI to RF Module

Value	CMD	Description	ADR	DAT
Basic µC Interface Commands				
0x00	NO OP	This command will perform no operation (used for debug purposes)	—	—
0x01	WRITE		Register Index	Data
0x02	READ		Register Index	—
0x03	SLEEP			
0x04	LED	Flash the RSSI LED on the RF module	—	# of flashes
0x06	S2	Used to signal the bidirectional S2 mode (installation mode) **[M-Bus command]**		
Basic Radio Interface Commands				
0x10	PASSTHRU			
0x11	WR TRX			
0x12	RD TRX			
0x13	TRX SLEEP			
0x14	RX MODE			
0x15	TX MODE			
0x16	TX FUNC		Func[0]	Func[1]
MAXBee Data Exchange Commands				
0x20	ATTN	Broadcast command for all nodes to change their RX status (session control command)		

0x21	OPEN	Reserved (session control command)	
0x22	REQ	Command for the node to send a reading from its sensor rather than waiting for autonomous data to be sent	
0x23	DATA	Return message from REQ; typically contains measurement/reading data from the sensor of the node	
0x24	SET	Used to set the sensor data to a defined value	
0x25	CLR	Used to clear the sensor data to a known reset state	
0x26	ACK	Acknowledge used for handshaking, confirmation of receipt, or other expected responses	
0x27	CLOSE	End a session with a particular node (session control command)	
0x28	AUTON	Autonomous messages originating from a node to the master; these data exchanges are usually event driven and occur without request from the master	Func[1]
MAXBee Forwarding List Commands			
0x30	ADD	Command to add an address to the forwarding list	
0x31	REMOVE	Command to remove an address from the forwarding list	
0x32	RENAME	Commands the node to take on a new address	
0x33	SWAP	Used to exchange addresses in the forwarding list table (for priority assessment of the communications link)	
0x34	MASK	Establish a forwarding list mask (subnet mask)	
M-Bus Commands			
0x44	SEND/NO REPLY	Transmit only submode S1	
0x46	S1	Used to signal the unidirectional S1 mode (installation mode)	
MAXBee Utility Commands			
0x50	ECHO	Directly respond to this message with the first data byte representing the RSSI of the received frame	
0x51	NET Network-wide broadcast (forward and act)		
0x52	STAT	Status information of the node (battery level, temperature, etc.)	
0x53	IDENT	Command for the node to flag its identity (light an LED)	
0x54	ALARM	A special event and AUTON data exchange	
0x55	VER	Notes the MAXBee protocol version in use by the node	
0x56	TIME	Time stamp/synchronization comparisons between nodes and the master	

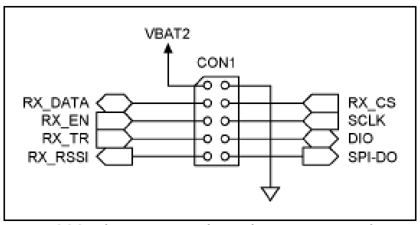
Module Request List

SPI Requests from RF Module to HHI

Value	CMD	Description	ADR	DAT
Basic µC Interface Commands				
0x00	NO OP			
0x01	PRESS			
0x02	ERR			
0x06	S2			
Basic Radio Interface Commands				
0x10	PASSTHRU			
MAXBee Data Exchange Commands				
		Return message from REQ, typically contains measurement/reading data from the sensor of		

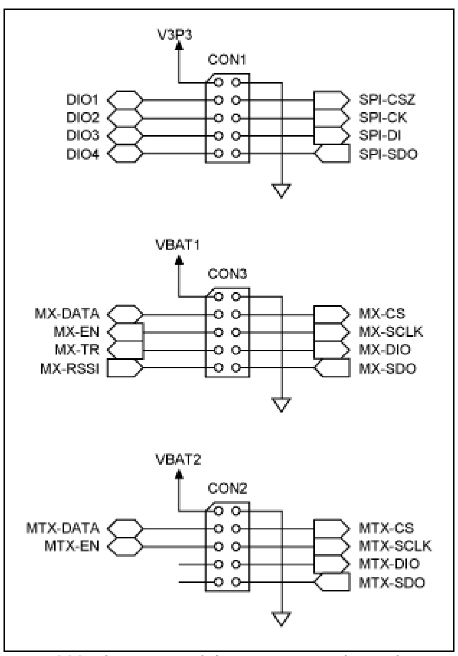
0x23	DATA	the node
0x28	AUTON	Autonomous messages originating from a node to the master; these data exchanges are usually event driven and occur without request from the master
M-Bus Commands		
0x44	SEND/NO REPLY	
0x46	S1	
MAXBee Utility Commands		
0x50	ECHO	Directly respond to this message with the first data byte representing the RSSI of the received frame
0x52	STAT	Status information of the node (battery level, temperature, etc.)
0x53	IDENT	Command for the node to flag its identity (light an LED)
0x54	ALARM	A special event and AUTON data exchange
0x55	VER	Notes the MAXBee protocol version in use by the node
0x56	TIME	Time stamp/synchronization comparisons between nodes and master

Appendix VI: Phase I/II/HHI Hardware Interface

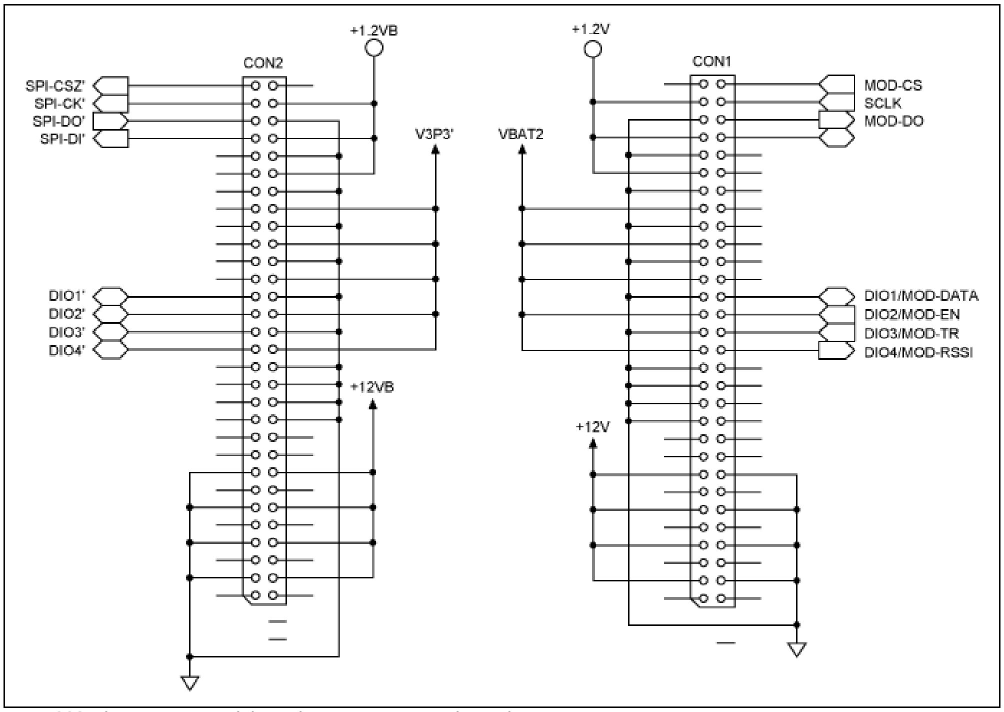


LFRD003 Phase II HHI board connector schematic.

Note: pin numbers are incorrect for the Hirose "header" footprint.



LFRD003 Phase II module connector schematics.



LFRD003 Phase III RF module and HHI connector schematics.

HHI Unit On-Board Communication

Microcontroller (MAXQ610)		LCD Display (DOGS102-6)	Key Controller (MAX7359)	RF Module Interface
SCLK	P0.0	SCK (pin 25)	SCL (pin 18)	SCLK (pin 6)
DIO	P0.1	SDA (pin 24)		DIO (pin 8)
RX_CS	P0.2			RX_CS (pin 4)
SDA	P0.7		SDA (pin 17)	
RX_DATA	P1.0			DIO1 (pin 3)
BINT	P1.3		!INT (pin 19)	
DISPL_CD	P1.4	OD (pin 26)		
DISPL_CS	P1.5	CS0 (pin 28)		

HHI to Phase I RF Module (PI Mod)

HHI			PI Mod			
Microcontroller (MAXQ610)		**M1 Connector**	**PI Mod Connector**	**Microcontroller (MAXQ610)**		**Transceiver (MAX7032)**
VBAT2		Power (pin 1)	V3P3 (pin 1)			
RX_DATA	P1.0	RX_DATA (pin 3)	DIO1 (pin 3)	DIO1	P1.3	RX_DATA' (pin 24)
RX_EN	P0.4	RX_EN (pin 5)	DIO2 (pin 5)	DIO2	P0.7	RX_EN (pin 23)
RX_TR	P0.5	RX_TR (pin 7)	DIO3 (pin 7)			RX_TR (pin 22)
RX_RSSI	P1.1	RX_RSSI (pin 9)	DIO4 (pin 9)			RX_RSSI (pin 21)
		SPI-DO (pin 10)	SPI-DO (pin 10)	SPI-DO	P1.4	DIO'
DIO	P0.1	DIO (pin 8)	SPI-DI (pin 8)	SPI-DI	P1.5	DIO' (pin 29)
SCLK	P0.0	SCLK (pin 6)	SPI-CK (pin 6)	SPI-CK	P1.6	SCLK' (pin 30)
RX_CS	P0.2	RX_CS (pin 4)	SPI-CSZ (pin 4)	SPI-CSZ	P1.7	RX_CS' (pin 28)
GND		GND (pin 2)	GND (pin 2)			

Phase I RF Module On-Board Communication			
Microcontroller (MAXQ610)		Transceiver (MAX7032)	PI Mod Connector
SCLK	P0.0	SCLK (pin 30)	SPI-CK (pin 6)
DIO	P01	DIO (pin 29)	SPI-DI (pin 8)
RX_CS	P0.2	!CS (pin 28)	SPI-CSZ (pin 4)
RX_EN	P0.4	ENABLE (pin 23)	DIO2 (pin 5)
RX_TR	P0.5	T/!R (pin 22)	DIO3 (pin 7)
DIO2	P0.7	RX_EN (pin 23)	DIO2 (pin 5)
RX_DATA	P1.0	RX_DATA (pin 24)	DIO1 (pin 3)
RX_RSSI	P1.1	RSSI (pin 21)	DIO4 (pin 9)
DIO1	P1.3	RX_DATA' (pin 24)	DIO1 (pin 3)
SPI-DO	P1.4	DIO'	SPI-DO (pin 10)
SPI-DI	P1.5	DIO' (pin 29)	SPI-DI (pin8)
SPI-CK	P1.6	SCLK' (pin 30)	SPI-CK (pin 6)
SPI-CSZ	P1.7	RX_CS' (pin 28)	SPI-CSZ (pin 4)

Related Parts		
MAX7032	Low-Cost, Crystal-Based, Programmable, ASK/FSK Transceiver with Fractional-N PLL	Free Samples
MAX7359	2-Wire Interfaced Low-EMI Key Switch Controller/GPO	Free Samples
MAXQ610	16-Bit Microcontroller with Infrared Module	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 5404: http://www.maximintegrated.com/an5404
REFERENCE DESIGN 5404, AN5404, AN 5404, APP5404, Appnote5404, Appnote 5404