

RFM products are now  
Murata products.

**LPR2430 Series**

**2.4 GHz  
802.15.4  
Wireless Industrial  
Transceivers**



**Integration Guide**

**Revision History**

Revision	Date	Author	Change Description
1	10/29/2010	F. Perkins	Initial issue
2	01/29/2015	R. Willett	Reformatted to comply with new Murata V.I.

---

# Important Regulatory Information

---

## RFM Product FCC ID

**LPR2430:**

**HSW-Z2430**

**IC 4492A-Z2430**

**LPR2430A:**

**HSW-Z2430A**

**IC 4492A-Z2430A**

**LPR2430ER:**

**HSW-Z2430HP**

**IC 4492A-Z2430HP**

**LPR2430ERA:**

**HSW-Z2430HPA**

**IC 4492A-Z2430HPA**

**Note:** These units have been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their expense.

---

## FCC MPE Requirements

**Information to user/installer regarding FCC Maximum Permissible Exposure (MPE) limits.**

**Notice to users/installers using the following antennas with MURATA RF products:**

**Omnidirectional, Patch and Corner Reflector Antennas:** The field strength radiated by these antennas, when connected to MURATA RF products, may exceed FCC mandated RF exposure limits. FCC rules require professional installation of these antennas in such a way that the general public will not be closer than 20 cm from the radiating aperture of any of these antennas. End users of these systems must also be informed that RF exposure limits may be exceeded if personnel come closer than 20 cm to the apertures of any of these antennas.

---

*See Section 3.10 of this manual for regulatory notices and labeling requirements. Changes or modifications to LPR2430, LPR2430A, LPR2430ER or LPR2430ERA modules not expressly approved by MURATA may void the user's authority to operate the module.*

---

## Table of Contents

1.0	Introduction.....	5
1.1	Features .....	6
1.2	Applications .....	6
2.0	LPR2430 Series Radio Operation.....	7
2.1	IEEE 802.15.4 .....	7
2.2	CSMA Channel Access.....	7
2.3	RF Transmission Error Control .....	7
2.4	Network Topologies .....	7
2.5	Network Formation.....	8
2.5.1	Network Association and Registration.....	8
2.5.2	Network Addresses and Leases .....	8
2.5.3	Network Relays.....	9
2.5.4	Transaction IDs .....	10
2.5.5	Broadcast Packets.....	10
2.3	Security .....	10
2.4	Transparent and Protocol Serial Port Modes.....	10
2.4.1	Throughput and Flow Control .....	11
2.5	Sensor I/O .....	11
2.5.1	Automatic Sensor I/O Reporting .....	11
2.5.2	I/O Binding .....	12
2.6	Sleep Modes .....	13
3.0	LPR2430 Series Hardware.....	15
3.1	Absolute Maximum Ratings .....	16
3.2	Electrical Specifications .....	16
3.3	Module Pin Descriptions .....	16
3.4	RFIO stripline.....	18
3.5	Input Voltages .....	19
3.6	ESD and Transient Protection .....	19
3.7	Interfacing to 5 V Logic Systems .....	20
3.8	Power-On Reset Requirements .....	20
3.9	Mounting and Enclosures.....	20
3.10	Labeling and Notices.....	20
4.0	Serial Protocol .....	22
4.1	Protocol Message Formats .....	22
4.1.1	Message Types.....	23
4.1.2	Message Format Details .....	23
4.1.3	Protocol Escape Sequence .....	25
4.1.4	Protocol Mode Data Message Example .....	25
4.2	Configuration Registers.....	26
4.2.1	Bank 0 - Transceiver Setup .....	26
4.2.2	Bank 1 - System Settings .....	27
4.2.3	Bank 2 - Status Registers .....	30
4.2.4	Bank 3 - Serial Settings .....	31
4.2.5	Bank 4 - Host Protocol Settings .....	32
4.2.6	Bank 5 - I/O Peripheral Registers .....	33
4.2.7	Bank 6 - I/O Setup .....	34
4.2.8	Bank FF - Special Functions.....	38
4.2.9	Protocol Mode Configuration/Sensor Message Example .....	39
4.2.10	Protocol Mode Event Message Example .....	40

---

5.0	LPR2430 Series Developer's Kits .....	41
5.1	Items Supplied in a Kit .....	41
5.2	Additional Items Needed .....	41
5.3	Developer's Kit Assembly and Testing .....	42
5.4	Developer Board Features .....	45
5.5	LPRDemo 2.0 Utility Program.....	47
6.0	Troubleshooting.....	65
7.0	Appendices.....	66
7.1	Ordering Information .....	66
7.2	Technical Support .....	66
7.3	LPR2430 Series Mechanical Specifications .....	67
7.4	Developer Board Schematic, LPR2430A and LPR2430ERA Version.....	69
7.5	Developer Board Schematic, LPR2430 and LPR2430ER Version.....	72
8.0	Warranty .....	75

## 1.0 Introduction

MURATA's 2.4 GHz LPR2430 series transceivers provide a low cost solution for point-to-point, point-to-multipoint and peer-to-peer wireless networks. LPR2430 series modules provide the flexibility and versatility to serve applications ranging from cable replacements to sensor networks. Based on the IEEE 802.15.4 wireless standard, LPR2430 series modules are easy to integrate and provide robust wireless communications in applications where full mesh network operation is not required. The LPR2430 series modules include MURATA's CNL V2.0 Network Layer firmware, which features a flexible and simple-to-use Application Programming Interface that provides a rich set of module functions.

LPR2430 series modules include the LPR2430, which transmits up to 1 mW of RF power. The small footprint and low peak current consumption of the LPR2430 make it ideal for short range applications where small size and low power are required. The LPR2430 is designed for use with an external 2.4 GHz antenna. The LPR2430A adds a chip antenna to the basic LPR2430 circuitry, providing a self contained radio-antenna module.

The LPR2430ER transmits up to 63 mW of RF power, making it suitable for longer range applications that are less sensitive to size and power consumption constraints. The LPR2430ER is designed for use with an external 2.4 GHz antenna. The LPR2430ERA adds a chip antenna to the basic LPR2430ER circuitry, providing a self contained radio-antenna module. The RF power level on the LPR2430ER and LPR2430ERA can be readily adjusted to 10 mW for operation under ETSI regulations. All LPR2430 series modules are FCC, Canadian IC and European ETSI certified for unlicensed operation in the worldwide 2.4 GHz ISM band.

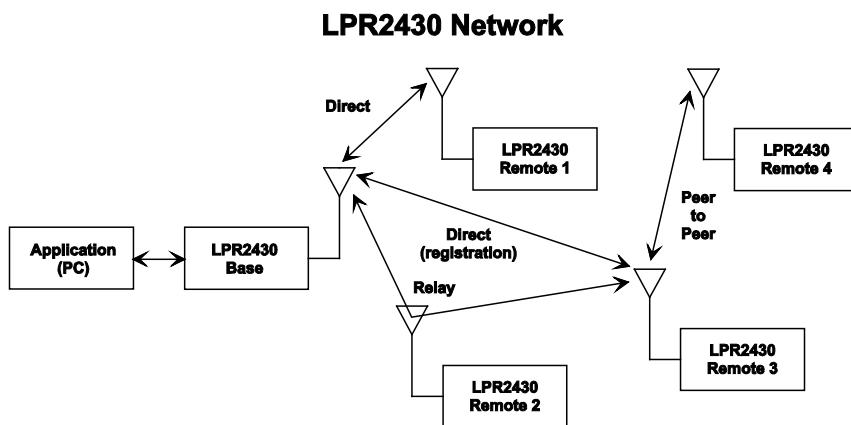


Figure 1.0.1

An example LPR2430 network is shown in Figure 1.0.1. An application running on a PC communicates with one or more LPR2430 remote nodes through an LPR2430 series base station. An LPR2430 network supports communications directly between a base and a single remote, referred to as a point-to-point network, or between a base and up to 63 remotes, referred to as point-to-multipoint. Peer-to-peer communications between remotes is also supported following the registration of each remote with the base station. MURATA's CNL Network Layer adds an important enhancement to point-to-point and peer-to-peer communications - relay forwarding. One-hop relay forwarding significantly mitigates transmission problems such as multipath fading, but with much less latency and complexity than full mesh network implementations. Thousands of LPR nodes with one-hop relay forwarding have been deployed, demonstrating the robustness it adds to network operation.

An LPR2430 series module is integrated with other components to create a complete node. These components include a host circuit board, a power supply (battery), serial and sensor I/O electronics and/or a host microcontroller, an antenna and a housing. Two common configurations are show in Figure 1.0.2.

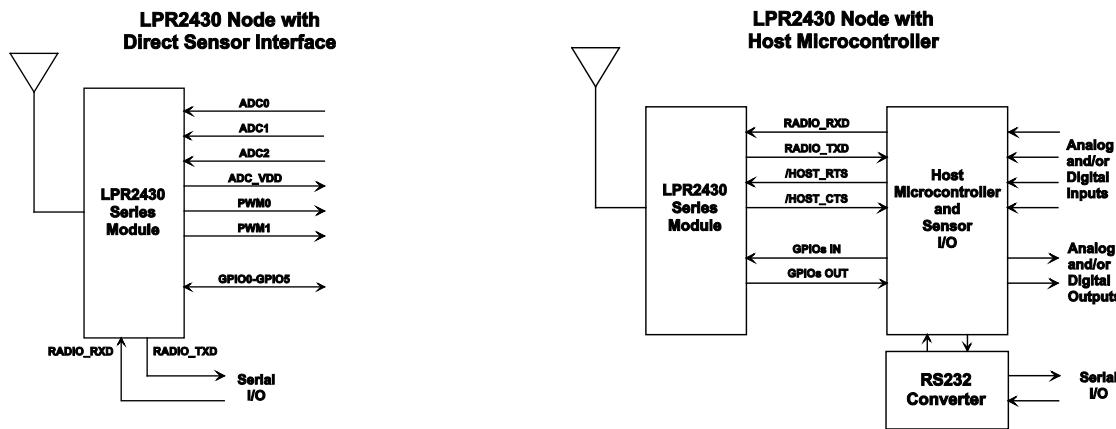


Figure 1.0.2

## 1.1 Features

LPR2430 series modules provide a unique set of features for wireless network applications:

- 2.4 GHz world-wide operation
- Compliant with IEEE 802.15.4 standard
- 128-bit AES data encryption
- Relay forwarding for robust, low latency network operation
- Low power consumption with sleep mode for long battery life operation
- Full -40 to +85 C industrial temperature range operation
- Analog and digital I/O plus serial data port
- Automatic and/or manual I/O data reporting
- Transparent or protocol formatted serial communications
- US FCC, Canadian IC and European ETSI certifications
- Analog and Digital I/O Binding

## 1.2 Applications

LPR2430 networks are well suited to applications where IEEE 802.15.4 compliance, strong encryption, industrial temperature range operation and long battery life are important. Many applications match these criteria, including:

- Encrypted Point-of-Sale Transactions
- Security and Access Control Systems
- HVAC and other Energy Management Control Systems

- 
- Industrial Process Control Monitoring
  - Food and Pharmaceutical Processing

## 2.0 LPR2430 Series Radio Operation

### 2.1 IEEE 802.15.4

The radio and media access control technology used by LPR2430 series modules is base on the IEEE 802.15.4 standard as defined for 2.4 GHz operation. Direct sequence spread spectrum (DSSS) using off-set quadrature phase shift keying (O-QPSK) modulation is used for RF transmissions. Data is sent over the air at 250 kb/s, carried on a spreading code running at 2 Mc/s. This provides 9 dB of processing gain against interference from other systems in the 2.4 GHz band.

### 2.2 CSMA Channel Access

As specified by the 802.15.4 standard, LPR2430 series modules use carrier sense multiple access (CSMA) to minimize packet collisions on the RF channel. CSMA is used by many data communication systems including Ethernet. In a CSMA radio network, a radio node that has data to transmit first listens to hear if any other node is transmitting. If the channel is clear, the node transmits its packet. If another node is currently transmitting, the first node waits a random period of time and listens again. This listening/waiting cycle continues until the first node detects a clear channel. When it does, it transmits its data.

If two radios detect a clear channel and transmit at the same time, the transmissions will collide and not be successfully received. As discussed in Section 2.3, any node that transmits a packet expects to receive an acknowledgement that the packet was received. The absence of an acknowledgement signals a node to retransmit. Because the transmitting nodes each wait random periods of time before transmitting again, the likelihood is low that the two nodes will both attempt to retransmit at precisely the same time.

### 2.3 RF Transmission Error Control

LPR2430 series modules use Automatic Retransmit reQuests (ARQ) for error control. Any node in a LPR2430 network that sends a packet expects to receive an acknowledgement packet (ACK) in response. If an ACK is not received, the packet is resent until an ACK is received, up to a limit set by the *ARQ\_AttemptLimit* parameter (see Section 4.2.1). The *ARQ\_AttemptLimit* parameter can be used twice. If the *ARQ\_AttemptLimit* is reached without a packet being sent directly to the destination node, the sender node will then attempt to send the packet through a relay node until an ACK is received or the *ARQ\_AttemptLimit* is reached again. Acknowledgements are not used with broadcast transmissions. In this case, the *BroadcastNumReps* parameter sets the number of times each broadcast transmission will be sent regardless of the success or failure of any transmission. Nodes successfully receiving more than one copy of the transmission will discard the duplicates.

### 2.4 Network Topologies

An LPR2430 series radio network is a point-to-multipoint star configuration consisting of a one base and one to 63 associated remotes. The base always uses the network address of 0x0000. Remotes are assigned network addresses in the range of 0x0001 to 0x003F by the base. A point-to-point system consists of a base and a single remote. Peer-to-peer routing between remotes in a point-to-multipoint network is also supported by LPR2430 series modules.

## 2.5 Network Formation

A network is identified by its PAN ID. Remotes may be configured to join either a specific PAN ID or to join any PAN they find. Multiple networks operating in the same area must be assigned different PAN IDs. When the base is first turned on, it will tune to one of the channels specified in its *ChannelList* (see Section 4.2.2) and detect if there is already a network with its PAN ID on this channel. If there is, the base will go to another channel in the list and keep trying until it finds an available channel or exhausts its list of channels. The first channel the base finds that is available will be used by the base. The base remains tuned to the channel and listens for requests to join its PAN.

Every time the base finds a channel that already has a PAN established with its PAN ID, the base generates an *Announce* packet indicating that a conflict was detected along with the channel on which the conflict occurred and the PAN ID of the conflict (see Section 4.1.2). If the base cycles through its entire channel list and finds conflicts on every channel, it will repeatedly cycle through its list until an available channel is found or its PAN ID and/or channel mask is reconfigured. It will continue to generate *Announce* packets as it cycles through the list. Note that the order the base cycles through its *ChannelList* is indeterminate.

When a remote is first turned on, it will tune to the first frequency in its channel mask (see Section 4.2.2), transmit a join request, and listen for a response for 250 ms. If there is more than one frequency enabled in its channel mask, it will loop through transmitting a join request on each frequency and listening for a response. Once this scanning phase is complete, the remote will attempt to associate with the first base it hears that also meets a minimum receive power level set by the *NwkFormThreshold* parameter (see Section 4.2.2). If no viable responses are received, the scan phase is repeated indefinitely.

### 2.5.1 Network Association and Registration

The normal 802.15.4 association process is used to initially admit remotes to the network. The CNL firmware initiates a registration handshake between the base and remote, and announce packets are generated on both sides to notify their respective hosts that a connection has been made. The base assigns each remote a 16-bit address with the lower 6-bits containing the relevant network address. The base keeps a persistent table of network addresses it has assigned to each remote so that it can assign the same address if a remote leaves the network and comes back.

If a 64<sup>th</sup> remote attempts to join the network, the base will deny the registration request. The remote will resume the scanning loop, but does not remember a base that denied it, and may request to register again if it encounters it.

### 2.5.2 Network Addresses and Leases

As specified in the 802.15.4 standard, each LPR2430 series radio has a 64-bit MAC address (IEEE address) permanently assigned at the factory, and a 16-bit network address that it receives when it joins a network. All network traffic except association is identified and delivered using network addresses. MAC addresses are only used to uniquely identify radios when they first register. It is the responsibility of the host to keep track of which MAC address corresponds to a given network address, if this information is needed. Each LPR2430 series remote receives a 16-bit network address from the base, of which only the lower 6 bits are used. The address assigned to a given remote is static - it will not change if the remote drops link and rejoins, or if any of the radios in the network are power cycled. The table of network addresses in the base can be cleared if desired by writing to the *AddressWipe* register. Previously registered nodes will need to be reset or power cycled, at which time they will be assigned new network ad-

---

resses. This is typically used in networks with large numbers of remotes that come and go from the network to allow new remotes to register in place of remotes that have left the area or have otherwise been serviced. A remote will also get a new network address if it joins a different base. Note that 0xFFFF is the broadcast address used to send a message to all nodes. See section 2.5.5.

All LPR2430 series remotes including sleeping remotes can be configured to send periodic heartbeat packets. These heartbeats serve the following purposes:

- To inform their base that they are still present in the network. The base will generate an *Announce* message to the host whenever a remote fails to report in.
- To control the state of the LINK output signal on a remote. The LINK output is held high when heartbeats are being acknowledged by the base.

When a remote sends a heartbeat packet, its base will respond with a heartbeat reply. The interval between heartbeats is derived from the *LeasePeriod* parameter (analogous to a DHCP lease time) as follows:

$$\begin{aligned}\text{Normal Heartbeat Interval} &= \text{one-half the lease period plus a random factor 0 to 511 ms} \\ \text{Fast Heartbeat Interval} &= \text{one-tenth the lease period plus a random factor 0 to 511 ms}\end{aligned}$$

If the *LeasePeriod* parameter is set to zero, heartbeats are disabled. In this case the base cannot inform its host that a remote has left the network, and remotes will only drop their LINK signal and look for a new base after being reset or power cycled.

Each heartbeat interval is offset by a random factor of 0 to 511 ms to prevent remotes from sending heartbeat packets in lockstep. If the base fails to respond to a heartbeat, the remote will continue to attempt to communicate with its base using relays. If this fails, the remotes period between heartbeats will accelerate from normal to fast until either the base responds or it fails to respond to three heartbeats in a row, at which point the remote will reenter scan mode and attempt to find another base. If the unlinked remote is in protocol mode, an announce packet is issued to its host that the connection has failed.

A base maintains a list of lease timers for its network addresses. Whenever a remote's lease timer expires, the base issues an *Announce* message to its host that the remote has left the network. As mentioned, the address assignment table can be manually cleared by writing to the *AddressWipe* register. If this operation is performed, none of the remote's networks address will be valid any longer. It is the responsibility of the user to reboot all of the remotes in the network in order for them to receive new network addresses. Otherwise erratic network operation can occur.

### 2.5.3 Network Relays

LPR2430 series modules include one-hop relay packet forwarding to enhance network performance. One-hop relay forwarding mitigates transmission problems such as multipath fading and temporary obstructions without introducing the latency and complexity inherent in a full mesh network. When a remote first joins a network, it is not aware of its neighbors and has not selected a relay node to assist with its communications. To collect this information, a remote scans for other remotes and evaluates their relay potential as a background task. A remote then reports its current relay node selection to the base for use in the base station's relay table. This allows the base to transmit to a remote using the appropriate relay node when necessary.

Each remote takes the following actions to update and report its relay selection:

- Periodically pings all other remotes and obtains the following scores:

- Path strength between its base and the candidate remote, as reported by the candidate.
- Path strength between itself and the candidate remote, as directly measured.
- Ranks the candidate remotes according to the two path strength scores and selects the top score for its relay.
- Reports the currently chosen relay to the base with a relay status packet.

#### 2.5.4 Transaction IDs

A Transaction ID is an identifier supplied by the host device when sending commands and protocol mode packets to distinguish replies from multiple messages that may be in process. The host application must supply a new value for each packet it sends, preferably using a sequence counter that increments for each new command. Transaction IDs for commands and replies are paired - the reply will return the same ID that was sent with the command. Packets such as *RxEvent* and *Announce* are not produced in response to a command, but are generated automatically. A separate transaction ID counter is maintained in LPR2430 series modules for events, which is initialized to 0x80 at startup and counts over a range of 0x80 to 0xFF. It is the responsibility of the host application to distinguish the transaction IDs for these packets from the ones that it supplies. This is easily accomplished by the host limiting its transaction IDs to the range 0x00 to 0x7F.

#### 2.5.5 Broadcast Packets

Network address 0xFFFF is used for broadcast packets. Since ARQ is not practical for broadcasts, a broadcast packet is sent multiple times. The number of times a broadcast packet is sent is controlled by the *BroadcastNumReps* parameter (default value of 4). To reduce the chance of a collision with other network packets, a random delay of 0 to *BroadcastMaxBackoff* is observed between broadcast transmissions (default value is 250 ms). See Section 4.2.2 for more broadcast parameter details. Note that sleeping remotes cannot receive broadcasts. While broadcast packets are intended primarily for use by the base station, they may be originated by a remote if necessary. The same distribution rules apply.

### 2.3 Security

LPR2430 series modules provide optional security from eavesdropping and unauthorized access by scrambling messages with a 128-bit AES key. Nodes that do not have the same security settings cannot communicate with each other. All nodes in a network must have the same security settings. Security keys must be commissioned on a node-by-node basis, either through the serial interface or through the RF interface. When using the RF interface, note that transmitting new keys over the air to a node that does not yet have security enabled makes the keys vulnerable to interception, so this should only be performed in a secure location. Note that a change to a security key does not take effect immediately but requires a restart to take effect.

RF packets that are encrypted include *RxData*, *RxEvent*, *GetRemoteRegister*, *GetRemoteRegisterReply*, *SetRemoteRegister*, and *SetRemoteRegisterReply*. Only the payload sections of these packets are encrypted, not the headers. *JoinRequest* and *JoinReply* RF packets are not encrypted.

### 2.4 Transparent and Protocol Serial Modes

LPR2430 series modules can work in one of two serial port data modes: transparent or protocol. Transparent mode requires no formatting and is simply the raw user data. Protocol mode formatting includes a start-of-packet framing character, length byte, addressing, command bytes, etc. Transparent mode opera-

---

tion is especially useful in point-to-point systems that act as simple cable replacements or in multipoint systems that use an application addressing scheme. In point-to-multipoint systems where the base needs to send data specifically to each remote, protocol formatting must be used. Protocol formatting is also required for configuration commands and responses, and sensor I/O commands and responses. Protocol mode can be used at the base radio while transparent mode is used at the remotes. Protocol formatting details are covered in Section 4.

### 2.4.1 Throughput and Flow Control

LPR2430 series modules can support continuous bidirectional streaming of serial data within the throughput limits of the of the RF channel. For applications that use streaming or operate in encrypted mode, it is recommended that the HOST\_CTS signal be observed to avoid overrunning the transmit buffer (flow control is disabled by default). It takes 5 ms to send a transmission independent of the payload data length, and 5 ms to return an ACK. This sets an upper limit of 100 acknowledged transmissions per second. Packets without encryption can carry up to 87 bytes, and packets with encryption can carry up to 66 bytes. The network capacity is  $100 \times 87 \times 10 = 87$  kb/s for non-encrypted data and  $100 \times 66 \times 10 = 66$  kb/s for encrypted data. These throughputs are maximums, and the actual throughputs will be less due to retries and channel contention. Where individual message streams can be longer than 66 bytes when using encryption or longer than 87 bytes when not using encryption, flow control is recommended to avoid transmit buffer overruns and lost data.

## 2.5 Sensor I/O

In addition to serial I/O, LPR2430 series modules include analog and digital I/O to support sensor network applications. Three ADC inputs, two PWM (DAC) outputs and six general purpose digital I/O ports are provided. LPR2430 series modules include commands to read each of the three ADC inputs individually. Commands are also provided to set each of the two PWM (DAC) outputs individually, plus commands to set the initial value of each PWM output at power up.

To support the six digital GPIO ports, LPR2430 series modules include commands to set digital I/O direction, read inputs, write outputs and set the outputs at power up and when sleeping. When configured as inputs, four of the GPIO lines can be configured as interrupts to the module. Any combination of these four lines can be enabled as interrupts simultaneously.

### 2.5.1 Automatic Sensor I/O Reporting

LPR2430 series remotes are capable of automatically sending reports of their current digital and analog I/O readings in response to a set of programmable trigger conditions. When a trigger occurs, the remote sends an *RxEVENT* message containing the contents of the I/O Register Bank (see Section 4.2.6) from GPIO0 up to and including the *EventFlags*, but not the PWM settings which are outputs only. When a base receives an *RxEVENT* message it automatically forwards it to its host. The normal ARQ process applies to the sending of an *RxEVENT* message, so while the report is sent multiple times it is possible that an *RxEVENT* may not be received.

If timer sleep mode is enabled, an I/O report trigger will wake the radio to send the report message. Sleep mode is not required, however, and automatic I/O reporting may be enabled for non-sleeping remotes. Automatic I/O reporting is enabled by setting one or more I/O triggers using the *IO\_ReportTrigger* register. The *IO\_ReportTrigger* register is a bitmask containing bits to enable each of the following trigger sources:

---

#### I/O Reporting Trigger Sources:

- bit 7*    ADC2 high/low thresholds
- bit 6*    ADC1 high/low thresholds
- bit 5*    ADC0 high/low thresholds
- bit 4*    Periodic report timer
- bit 3*    GPIO3 falling edge
- bit 2*    GPIO2 falling edge
- bit 1*    GPIO1 falling edge
- bit 0*    GPIO0 falling edge

Each of these triggers has some additional settings. The periodic report period is set by the *IO\_ReportInterval* register, the ADCs sample period set by the *ADC\_SampleInterval*, the individual high and low ADC trigger thresholds are set by *ThresholdLo* and *ThresholdHi* parameters, and GPIOs falling edge triggers are enabled on the *GPIO\_EdgeTrigger* register (see Section 4.2.7).

When a trigger fires, only a single I/O report is generated. If the condition persists, as in the case of a GPIO being held low or an ADC input continuing to exceed a threshold value, recurring reports are not generated. Each respective GPIO or ADC must return to its non-triggered condition before it can fire again. Each trigger source is independent. For instance, if a report is generated in response to an ADC trigger and the ADC remains above the threshold, a subsequent GPIO edge trigger can generate another report. Should additional events occur while an event report is being sent, flags for these events will be accumulated and sent as a follow on report.

#### 2.5.2 I/O Binding

I/O binding configures a module to map the states of GPIO0 and GPIO1, and the value of ADC0 contained in an I/O report sent to it by another module to its GPIO2, GPIO3 and PWM0 outputs. The mapping is shown in Table 2.5.2.1. This function can be enabled independently from the other I/O reporting functions that control when a module transmits its reports. An LPR2430 module can send its I/O report to only one other module address. I/O binding is typically used in a point-to-point application with a single base and remote, although other topologies are possible. A module that receives an I/O report from another module can have its I/O report sent back to that module, to a different module, or to the base. It is only necessary to configure I/O binding on a module that will map its I/O to another module's I/O report.

Received I/O	Mapped to Output I/O
GPIO0	GPIO2
GPIO1	GPIO3
ADC0	PWM0

Table 2.5.2.1

When I/O binding is enabled on a module, GPIO0 and GPIO1 are forced to be inputs, GPIO2 and GPIO3 are forced to be outputs, and the *GPIO\_Dir* and *GPIO\_Alt* settings are ignored for GPIO0..GPIO3.

I/O binding requires the sending module to have one or more automatic I/O reporting triggers enabled, as discussed in Section 2.5.1. The receiving module must be awake for reliable operation. The sending module can be awake or in one of the sleep modes discussed in Section 2.6. If a sleep mode is being used at the sending node, its *WakeDuration* timer must be set long enough to receive a response if that is a system requirement. The periodic report timer is used to send I/O mapping updates from the transmitting node that are independent of ADC threshold and GPIO edge trigger events. The minimum practical report timer period is 100 ms, and this setting will consume most of the modules' resources, making other

---

traffic such as configuration commands slow. A commonly used configuration is to enable the edge triggers on GPOI0 and GPIO1 and the threshold triggers on ADC0 on the transmitting node to map the onset of an alarm event, and using the report timer set at an interval of several seconds or more to clear the alarm event mapping. This provides fast mapping of an alarm event at the receiving node, with a delay of up to several seconds in mapping the clearing of the event, which is often not critical information. This configuration provides adequate network resources to responsively transmit configuration commands, routine data, etc., in addition to the I/O mapping.

## 2.6 Sleep Modes

To facilitate battery operation, LPR 2430 series modules includes two low power sleep modes, periodic sleep (wake on periodic report timer or I/O triggers), and deep sleep (wake on GPIO triggers only). When a sleep mode is enabled, an LPR2430 series module will stay in a low-power state and only wake up in response to its allowed I/O report triggers. The sleep mode is set by the *SleepMode* parameter (see Section 4.2.1).The following explains the rules that sleeping remotes follow:

- In periodic sleep, the module will wake up when the periodic report timer or any enabled I/O report trigger fires.
- In deep sleep, the module will only wake up when an enabled GPIO report trigger fires.
- When any of the ADC triggers are enabled in periodic sleep, the module will wake up every *ADC\_Sample/Interval* just long enough to sample the ADCs and compare the values against the thresholds and then go back to sleep. The module does not wake up to sample the ADCs in deep sleep.
- A remote with either sleep mode enabled will not enter sleep mode until it has associated and registered with a base and has at least one trigger enabled.
- After a remote has received an acknowledgement for its I/O report, a *WakeDuration* timer is started before the remote goes back to sleep. The purpose of this timer is to allow the base a window of time to send a message back to the remote. Note that the only notification the base application will have that a remote is awake is its I/O report packet. In order to send it data, the base application must ensure that the message is transmitted and received before the remote's *WakeDuration* timer elapses. If this function is not needed, the *WakeDuration* may be set to zero to disable it. A remote will resend an I/O report if an acknowledgement is not received from the base, until it reaches the *ARQ\_AttemptLimit*. The remote will then remain awake for the duration set by the *WakeDuration* timer, and go back to sleep.

To summarize, while a remote is awake, the following list of checks are used to determine if and when it will go back to sleep:

- If the remote receives an acknowledgement for a packet it has sent, it resets the *WakeResponseTime* timer to remain awake, or if *WakeResponseTime* is zero it goes back to sleep.
- So long as a GPIO for which edge triggered I/O reporting is enabled remains at a logic low, the remote will remain awake.
- The remote will remain awake while it still has any ARQ attempts left for a queued transmit packet of any type.
- The remote will remain awake while it has serial characters in its buffer left to transmit to its local host, plus whatever time is required for the last transmitted character to clear the TXD pin.

---

Sleep functions are controlled by the following registers (see Section 4.2.1):

*SleepMode* - Enables/disables sleep modes

*WakeDuration* - Amount of time that a remote will wait for a response after sending an I/O report

Behavior of the GPIO pins during sleep is configured by the *GPIO\_SleepMode*, *GPIO\_SleepDir*, and *GPIO\_SleepState* parameters. If a module's host application requires a way to monitor whether the module is awake or not, the HOST\_CTS signal (alternate function of GPIO4) may be used. HOST\_CTS will usually be low (active) when the module is awake and high (internal pullup) when the module is asleep. Alternatively, a user may configure any GPIO to function in a similar way using the *GPIO\_SleepMode*, *GPIO\_SleepDir* and *GPIO\_SleepState* parameters. For example, to configure GPIO3 to generate a logic low when the module is awake, GPIO3 is initialized as a logic low output, the GPIO3 *GPIOSleepDir* is set to an output, and the GPIO3 *GPIO\_SleepState* is set to a logic high.

### 3.0 LPR2430 Series Hardware

The block diagram of the LPR2430 and LPR2430A is shown in Figure 3.0.1, and the block diagram of the LPR2430ER and LPR2430ERA is shown in Figure 3.0.2.

**LPR2430 & LPR2430A Block Diagram**

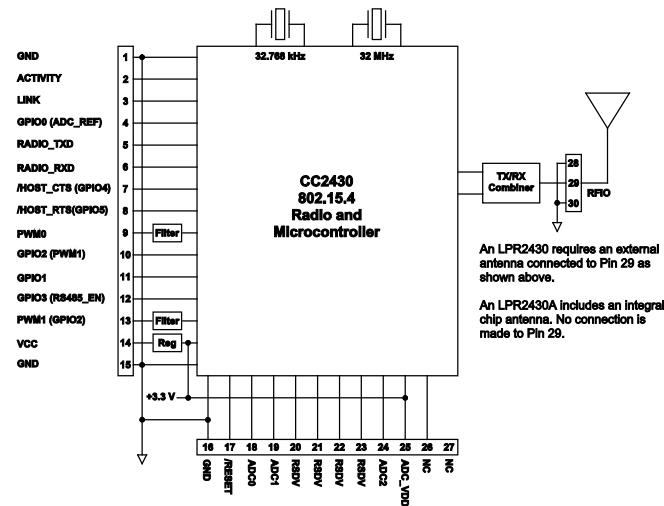


Figure 3.0.1

**LPR2430ER & LPR2430ERA Block Diagram**

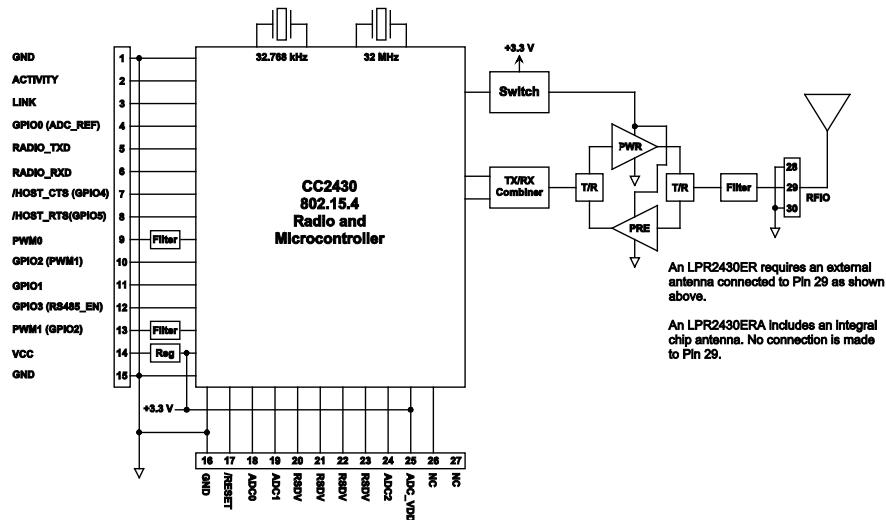


Figure 3.0.2

The major hardware component of the LPR2430 series modules is the CC2430 IEEE 802.15.4 compliant transceiver with integrated 8051 microcontroller. The LPR2430 series modules operate in the frequency band of 2405 to 2475 MHz. The LPR2430ER and LPR2430ERA include a low noise preamplifier in the receiver path and a power amplifier in the transmitter path, greatly increasing the operating range of the CC2430. Two crystals are provided to operate the CC2430, a 32 MHz crystal for normal operation and a 32.768 kHz crystal for precision sleep mode operation. The LPR2430 series modules provide a variety of application hardware interfaces including a UART interface, three 11-bit ADC inputs, two PWM (DAC) outputs, and six general purpose digital I/O ports.

### 3.1 Absolute Maximum Ratings

Rating	Sym	Value	Units
Input/Output Pins Except ADC Inputs		-0.5 to +3.63	V
Non-Operating Ambient Temperature Range		-40 to +85	°C

Table 3.1.1

### 3.2 Electrical Specifications

Characteristic	Sym	Minimum	Typical	Maximum	Units
Operating Frequency Range		2405		2475	MHz
Operating Frequency Tolerance		-300		300	kHz
Spread Spectrum Method			Direct Sequence		
Modulation Type			O-QPSK		
Number of RF Channels			15		
RF Data Transmission Rate			250		kb/s
Symbol Rate Tolerance				120	ppm
RF Channel Spacing			5		MHz
10E-5 BER Receiver Sensitivity, LPR2430/A			-92		dBm
10E-5 BER Receiver Sensitivity, LPR2430ER/ERA			-95		dBm
Upper Adjacent Channel Rejection, +5 MHz			41		dB
Lower Adjacent Channel Rejection, -5 MHz			30		dB
Upper Alternate Channel Rejection, +10 MHz			55		dB
Lower Alternate Channel Rejection, -10 MHz			53		dB
RF Transmit Power, LPR2430			0		dBm
RF Transmit Power, LPR2430A			0		dBm
RF Transmit Power, LPR2430ER			18		dBm
RF Transmit Power, LPR2430ERA			18		dBm
Transmit Power Adjustment, LPR2430/A				26	dB
Transmit Power Adjustment, LPR2430ER/ERA				20	dB
Optimum Antenna Impedance			50		Ω
ADC Input Range		0		3.3	V
ADC Input Resolution			11		bits
ADC Input Impedance		55			MΩ
PWM Output Resolution*		8		16	bits
Data Serial Port Baud Rates			1.2, 2.4, 4.8, 9.6 (default), 19.2, 28.8, 38.4, 57.6, 76.8, 115.2		kb/s

\*PWM0 has 8-bit resolution, PWM1 has 16-bit resolution. Built in PWM output filters suppress ripple to 7 bits. Additional filtering can be added externally.

Characteristic	Sym	Minimum	Typical	Maximum	Units
Digital I/O:					
Logic Low Input Level		-0.3		0.5	V
Logic High Input Level		2.8		3.6	V
Input Pull Up/Down Resistor		20			KΩ
Logic Low Output Level		0		0.4	V
Logic High Output Level		2.4		3.3	V
Power Supply Voltage Range	V <sub>cc</sub>	+3.3		+5.5	V <sub>dc</sub>
Power Supply Voltage Ripple				10	mV <sub>P-P</sub>
Receive Mode Current, LPR2430/A			27		mA
Receive Mode Current, LPR2430ER/ERA			33		mA
Transmit Mode Current, LPR2430/A			28		mA
Transmit Mode Current, LPR2430ER/ERA			130		mA
Sleep Mode Current				3	μA
Operating Temperature Range		-40		85	°C
Operating Relative Humidity Range, Non-condensing		10		90	%

Table 3.2.1

### 3.3 Module Pin Descriptions

Pin	Name	I/O	Description
1	GND	-	Power supply and signal ground. Connect to the host circuit board ground. All ground pins must be connected.
2	ACTIVITY	O	RF activity indicator. Output pulses high when a packet is sent or received.
3	LINK	O	Link indicator. Output is high when the radio has successfully joined a network.
4	GPIO0 (ADC_REF)	I/O	Configurable digital I/O port 0. When configured as an output, the power-on state is also configurable. This pin can also be configured as a reference voltage input for the ADCs, 0 to 3.3 V, 1.25 V typical.
5	RADIO_TXD	O	Serial data output (UART) from the radio to the host.
6	RADIO_RXD	I	Serial data input (UART) from the host to the radio.
7	GPIO4 (/HOST_CTS )	I/O	Configurable digital I/O port 4. When configured as an output, the power-on state is also configurable. Also configurable as UART flow control output. The LPR2430 sets this line low to indicate it is ready to accept data from the host on the RADIO_RXD input. When the LPR2430 sets this line high, the host must stop sending data. The default state is GPIO4.
8	GPIO5 (/HOST_RTS)	I/O	Configurable digital I/O port 5. When configured as an output, the power-on state is also configurable. Also configurable as UART flow control input. The host sets this line low to allow data to flow from the RADIO_TXD pin. When the host sets this line high, the LPR2430 will stop sending data to the host. The default state is GPIO5
9	PWM0	O	Pulse-width modulated output 0 with internal low-pass filter. Provides a DAC function, 0 to 3.3 V.
10	GPIO2 (PWM1)	I/O	Configurable digital I/O port 2. When configured as an output, the power-on state is also configurable. This pin is connected to the input of the low-pass filter driving Pin 13, and is also configurable as a PWM output. The default setting is for GPIO2.

11	GPIO1	I/O	Configurable digital I/O port 1. When configured as an output, the power-on state is also configurable.
12	GPIO3 (RS485_EN)	I/O	Configurable digital I/O port 3. When configured as an output, this high current port can sink up to 20 mA. The power-on output state is also configurable. Can also be configured as active low transmit enable for controlling an RS485 or other half-duplex bus driver. The default state is GPIO3.
13	PWM1 (GPIO2)	O	GPIO2 (Pin 10) drives this pin through a low-pass filter. Provides a DAC function when GPIO2 is configured as a PWM output. The default settings is for GPIO2.
14	VCC	I	Power supply input, +3.3 to +5.5 Vdc.
15	GND	-	Power supply and signal grounds. Connect to the host circuit board ground. All ground pins must be connected.
16	GND	-	Power supply and signal grounds. Connect to the host circuit board ground. All ground pins must be connected.
17	/RESET	I	Active low module hardware reset. Hold this input low when the power supply input is less than 3.3 Vdc. The module firmware boots up and will accept commands about 3 seconds after this input goes high.
18	ADC0	I	7-bit to 12-bit ADC input 0. ADC full scale reading can be referenced to the module's +3.3 V regulated supply, the ADC's internal +2.5 V reference, or ADC_REF (Pin 4).
19	ADC1	I	ADC input 1. Same configuration options as ADC0.
20	RSVD	-	Reserved pin. Leave unconnected.
21	RSVD	-	Reserved pin. Leave unconnected.
22	RSVD	-	Reserved pin. Leave unconnected.
23	RSVD	-	Reserved pin. Leave unconnected.
24	ADC2	I	ADC input 2. Same configuration options as ADC0.
25	ADC_VDD	O	Module's +3.3 V regulated supply, used for ratiometric ADC readings. Current drain should be less than 5 mA.
26	NC	-	No connection.
27	NC	-	No connection.
28	GND	-	RF ground. Connect to the host circuit board ground plane, and to shield when using a coaxial cable. All ground pins must be connected.
29	See Text	I/O	RFIO port on LPR2430 and LPR2430ER. No connection on LPR2430A and LPR2430ERA. For the LPR2430 and LPR2430ER, connect the antenna to this port with a 50 ohm stripline or semi-rigid coaxial cable.
30	GND	-	RF ground. Connect to the host circuit board ground plane, and to shield when using a coaxial cable. All ground pins must be connected.

Table 3.3.1

### 3.4 RFIO Stripline

To maintain the RF performance of the LPR2430 series modules, the PCB trace running from the RFIO pin of the module to the RF connector or antenna must have an impedance of 50 ohms. This is easily accomplished by controlling the width of the trace. Referring to Figure 3.4.1, the required width of the stripline depends on the dielectric constant and the thickness of the circuit board between the stripline

and the groundplane. For FR-4 type circuit board materials (dielectric constant of 4.7), the width of the stripline is equal to 1.75 times the thickness of the circuit board. Note that other circuit board traces should be spaced away from the stripline to prevent signal coupling, as shown in Table 3.4.1. The stripline trace should be kept short to minimize its insertion loss.

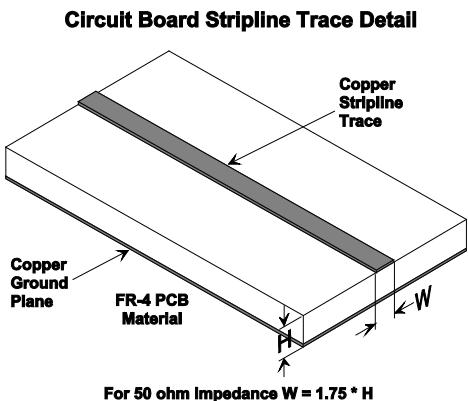


Figure 3.4.1

Trace Separation from 50 ohm Microstrip	Length of Trace Run Parallel to Microstrip
100 mil	125 mill
150 mil	200 mil
200 mil	290 mil
250 mil	450 mil
300 mil	650 mil

Table 3.4.1

### 3.5 Input Voltages

LPR2430 series radio modules can operate from an unregulated DC input (Pin 14) in the range of 3.3 V (trough) to 5.5 V (peak) with a maximum ripple of 5% over the temperature range of -40 to 85 °C. Applying AC, reverse DC, or a DC voltage outside the range given above can cause damage and/or create a fire and safety hazard. Further, care must be taken so logic inputs applied to the radio stay within the voltage range of 0 to 3.3 V. Signals applied to the analog inputs must be in the range of 0 to ADC\_REF (Pin 25). Applying a voltage to a logic or analog input outside of its operating range can damage the module.

### 3.6 ESD and Transient Protection

LPR2430 series circuit boards are electrostatic discharge (ESD) sensitive. ESD precautions must be observed when handling and installing these components. Installations must be protected from electrical transients on the power supply and I/O lines. This is especially important in outdoor installations, and/or where connections are made to sensors with long leads. Inadequate transient protection can result in damage and/or create a fire and safety hazard.

### 3.7 Interfacing to 5 V Logic System

All logic signals including the serial ports on LPR2430 series modules are 3.3 V signals. To interface to 5 V signals, the resistor divider network shown in Figure 3.8.1 below must be placed between the 5 V signal outputs and the LPR2430 signal inputs. The output voltage swing of the LPR2430 3.3 V signals is sufficient to drive 5 V logic inputs.

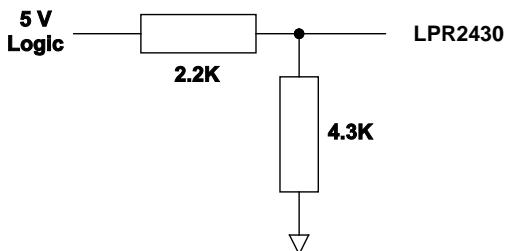


Figure 3.8.1

### 3.8 Power-On Reset Requirements

When applying power to an LPR2430 series module, the /RESET pin should be held low until the power supply voltage reaches 3.3 volts for 100 milliseconds. The module firmware boots up and will accept commands about 3 seconds after this input goes high.

### 3.9 Mounting and Enclosures

LPR2430 series modules are mounted by reflow soldering them to a host circuit board. Refer to Section 7.3 and the LPR2430, LPR2430A, LPR2430ER and LPR2430ERA Data Sheets for mounting details.

Enclosures for the LPR2430 series modules must be made of plastics or other materials with low RF attenuation to avoid compromising antenna performance where the antennas are internal to the enclosure. Metal enclosures are not suitable for use with internal antennas as they will block antenna radiation and reception. Outdoor enclosures must be water tight, such as a NEMA 4X enclosure.

### 3.10 Labeling and Notices

LPR2430, LPR2430A, LPR2430ER, and LPR2430ERA FCC Certifications - The LPR2430, LPR2430A, LPR2430ER, and LPR2430ERA hardware has been certified for operation under FCC Part 15 Rules, Section 15.247. *The antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.*

LPR2430, LPR2430A, LPR2430ER, LPR2430ERA FCC Certification FCC Notices and Labels - *This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.*

A clearly visible label is required on the outside of the user's (OEM) enclosure stating "Contains FCC ID: HSW-Z2430" for the LPR2430, or "Contains HSW-Z2430A" for the LPR2430A, or "Contains FCC ID: HSW-Z2430HP" for the LPR2430ER or "Contains FCC ID: HSW-Z2430HPA" for the LPR2430ERA.

---

**WARNING:** This device operates under Part 15 of the FCC rules. Any modification to this device, not expressly authorized by Murata N.A., Inc., may void the user's authority to operate this device.

Canadian Department of Communications Industry Notice - IC: 4492A-Z2430 for the LPR2430, IC: 4492A-Z2430A for the LPR2430A, IC: 4492A-Z2430HP for the LPR2430ER, IC: 4492A-Z2430HPA for the LPR2430ERA

This apparatus complies with Health Canada's Safety Code 6 / IC RSS 210.

#### **ICES-003**

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus as set out in the radio interference regulations of Industry Canada.

Le present appareil numerique n'emett pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de Classe B prescrites dans le reglement sur le brouillage radioelectrique edicte par Industrie Canada.

#### **ETSI EN 300 328**

LPR2430 series modules have passed ETSI EN 300 328 testing conducted by an independent test laboratory.

#### **USE IN THE EU**

The LPR2430 series modules are intended for use in the European Union under the following condition: The Effective Isotropic Radiated Power (EIRP), which is the module's RF transmit power plus any antenna gain, must not exceed 12.7 dBm. LPR2430-based products must have the RF transmit power of the module set to meet this requirement, and cannot allow end users of the products to change the module RF transmit power in such a way to exceed the EIRP limit.

## 4.0 Serial Protocol

### 4.1 Protocol Message Formats

LPR2430 series modules can work in one of two serial port data modes - transparent or protocol. Transparent mode requires no formatting and is simply the raw user data. Protocol mode formatting includes a start-of-packet framing character, length byte, addressing, command bytes, etc. In point-to-multipoint systems where the base needs to send data specifically to each remote, protocol formatting must be used. Protocol formatting is also required for configuration commands and responses, and sensor I/O commands and responses.

The configuration of an LPR2430 module is stored in a set of variable length registers. Most registers are read-write, with a few read-only or write-only. Changes made to the register settings are temporary until a *MemorySave* (to EEPROM) command is executed. Resetting or power-cycling the module will clear any changes that have not been saved to permanent memory using the *MemorySave* command. LPR2430 series modules can be configured to start in protocol mode at power-up, in which case the *EnterProtocolMode* command is not required.

All protocol messages have a common header format as shown in Figure 4.1.1:

0	1	2	3	4 ...
SOP	Length	TransID	PktType	variable number of arguments ...

Figure 4.1.1

The scale above is in bytes.

- The *Start-of-Packet* (SOP) character, 0xFB, is used to distinguish the beginning of a message and to assure synchronization in the event of a glitch on the serial port at startup.
- The *Length* byte is defined as the length of the remainder of the message following the length byte itself (or the length of the entire message - 2).
- The Packet Type (PktType) byte specifies the type of message. It is a bitfield-oriented specifier, decoded as follows:

Bits 7..5	Reserved for future use
Bit 4	Reply - this bit is set to indicate a packet is a reply
Bits 3..0	Type - these bits indicate the packet type

The lower 4 bits (3..0) specify a message type. Bit 4 is a modifier indicating that the message is a command (0) or a reply (1). A reply message has the original command type in bits 3..0.

- The transaction ID (*TransID*) is an identifier supplied by the host device to distinguish replies from multiple commands that may be in process. The host application must supply a new value for each packet it sends, preferably using a sequence counter that increments for each new command. Transaction IDs for commands and replies are paired - the reply will return the same *TransID* that was sent with the command. Packets such as *RxEVENT* and *Announce* are not produced in response to a command, but are generated automatically. A separate transaction ID counter is maintained in LPR2430 series modules for events, which is initialized to 0x80 at startup and counts over a range of 0x80 to 0xFF. It is the responsibility of the host to distinguish the transaction IDs for these packets from the ones that it supplies. This is usually accomplished by the host limiting its transaction IDs to the range 0x00 to 0x7F.

- Arguments vary in size and number depending on the type of packet and whether it is a packet sent from the user or a reply from the module. See Table 4.1.2.1 below.

Packets that are generated on the serial interface by the user are referred to as *host* packets. Packets that are generated by the module are referred to as *reply* packets. For many packet types, there is a reply packet that corresponds to a host packet. For example, when the host sends a *TxData* packet, the radio will reply to indicate the status of the transmission, whether it succeeded or failed. Some packet types are host-only or reply-only as shown in Table 4.1.2.1.

### 4.1.1 Message Types

Each message generally has two forms, a command from the host and a reply from the module. Depending on the direction, they have different arguments as shown in Table 4.1.2.1. Event messages from the module such as received data packets or status announcements make up a third category of messages. To assist in interpreting the command-reply data flow, the direction is indicated by the high nibble in the message type. For example, an *EnterProtocolMode* command from the host is a message type 0x00, and the *EnterProtocolModeReply* from the radio is a message type 0x10. Event messages, including *RxData*, *RxEVENT* and *Announce* packets are indicated by 0x20 in the high nibble of the type byte. If multiple arguments are to be provided, they are to be concatenated in the order shown. Little-Endian byte format is used for all multi-byte arguments, where the lowest order byte is the left-most byte of the argument and the highest order byte in the right-most byte of the argument.

### 4.1.2 Message Format Details

Com-mand	Reply	Event	Description	Direction	Arguments
0x00	-	-	EnterProtocolMode	from Host	LPR2400 (ASCII characters)
-	0x10	-	EnterProtocolModeReply	from Radio	none
0x01	-	-	ExitProtocolMode	from Host	none
-	0x11	-	ExitProtocolModeReply	from Radio	none
0x02	-	-	SoftwareReset	from Host	none
-	0x12	-	SoftwareResetReply	from Radio	none
0x03	-	-	GetRegister	from Host	Reg, Bank, Span
-	0x13	-	GetRegisterReply	from Radio	Reg, Bank, Span, Val
0x04	-	-	SetRegister	from Host	Reg, Bank, Span, Val
-	0x14	-	SetRegisterReply	from Radio	none
0x05	-	-	TxDATA	from Host	Addr, Data
-	0x15	-	TxDATAReply	from Radio	TxStatus, LQI
-	-	0x26	RxDATA	from Radio	Addr, LQI, Data
-	-	0x27	Announce	from Radio	AnnStatus, additional fields
-	-	0x28	RxEVENT	from Radio	Addr, LQI, Data
0x0A	-	-	GetRemoteRegister	from Host	Addr, Reg, Bank, Span
-	0x1A	-	GetRemoteRegisterReply	from Radio	TxStatus, Addr, LQI, Reg, Bank, Span, Val
0x0B	-	-	SetRemoteRegister	from Host	Addr, Reg, Bank, Span, Val
-	0x1B	-	SetRemoteRegisterReply	from Radio	TxStatus, Addr, LQI
-	-	0x2C	JoinRequest	from Radio	MacAddr, NwkAddr, DeviceMode, SleepMode
-	0x1C	-	JoinReply	from Host	PermitStatus

Table 4.1.2.1

---

Arguments:

Reg = Register location (*1 byte*)

Bank = Register bank: provides a logical separation from other data regions (*1 byte*)

Span = Number of bytes of register data to get or set; must align to a parameter boundary (*1 byte*)

Val = Value to read from or write to register (see *Table 4.1.2.1* for size and acceptable range)

Data = User data (*variable size*)

Addr = Network address of sender or recipient (*2 bytes*)

MacAddr = MAC address of sender or recipient (*8 bytes*)

TxStatus = Result of last TxData operation (*1 byte*)

0 = Acknowledgement received

1 = No acknowledgement received

2 = (Remote) Not linked

LQI = LQI measurement, 0x01 to 0xFE; values of 0x00 and 0xFF have special meanings: (*1 byte*)

0x00 = No LQI measured because no ACK was received

0xFF = No LQI measured because packet was relayed

PanID = Network identifier of network joined (*2 bytes*)

Channel = Channel number where a PanID conflict was detected (*1 byte*)

AnnStatus = Status announcement (*1 byte*); additional fields are also reported depending on the status code:

Status code

Additional fields

A0 = Radio has completed startup initialization

*none*

A1 = Base: PAN has been formed, ready for data

PanID

A2 = Base/router: a remote has joined me

MacAddr, Addr

A3 = Remote/router: joined a PAN, ready for data

PanID, Addr

A4 = Remote/router: exited PAN (out of range)

*none*

A5 = Base: node has left the network

Addr

A7 = Base: PanID conflict detected

PanID, Channel

Status codes for error conditions

Additional fields

E0 = Protocol error - invalid packet type

*none*

E1 = Protocol error - invalid argument

*none*

E2 = Protocol error - general error

*none*

E3 = Protocol error - parser timeout

*none*

E4 = Protocol error - register is read-only

*none*

E5 = Routing error - unknown destination address

*none*

E8 = Serial error - UART receive buffer overflow

*none*

E9 = Serial error - UART receive overrun

*none*

EA = Serial error - UART framing error

*none*

PermitStatus = Permission for new node to join: 0x00 = denied, 0x01 = permitted (*1 byte*)

### 4.1.3 Protocol Escape Sequence

The escape sequence is a string of bytes that is input in transparent mode to switch an LPR2430 series module to protocol mode. The escape sequence provides a means of switching between sending transparent data and configuring an LPR2430 series module. The *EnterProtocolMode* command is used as the escape sequence. The escape sequence must be preceded and followed by a six-character time gap, and must be entered with no interval between characters greater than *TransparentModeTimeout* interval in order to be accepted. Otherwise, the packet will be sent over-the-air as transparent data.

### 4.1.4 Protocol Mode Data Message Example

In this example, ASCII text *Hello* is sent from the base to a remote using a *TxDATA* command. The network address of the remote is 0x0010. The transaction ID used is 0x02. The protocol formatting for the host message is:

SOP	Length	TransID	PktType	Addr	Addr	"H"	"e"	"l"	"l"	"o"
0xFB	0x09	0x02	0x05	0x10	0x00	0x48	0x65	0x6C	0x6C	0x6F

There are 9 bytes following the length byte, so the length byte is set to 0x09. Note that the 0x0010 network address is entered in Little-Endian byte order 0x10 0x00.

When an ACK to this message is received from the remote, the base outputs a *TxDATAReply* message to its host:

SOP	Length	TransID	PktType	TxStat	LQI
0xFB	0x04	0x02	0x15	0x00	0x35

The 0x00 *TxStatus* byte value indicates the ACK reception from the remote. The *LQI* value of the received ACK is 0x35, indicating a received signal strength of approximately -70 dBm on an LPR2430 or LPR2430A or -82 dBm on an LPR2430ER or LPR2430ERA.

If the remote is in protocol mode, the message is output in the following format:

SOP	Length	TransID	PktType	Addr	Addr	LQI	"H"	"e"	"l"	"l"	"o"
0xFB	0x0A	0x02	0x26	0x00	0x00	0x35	0x48	0x65	0x6C	0x6C	0x6F

The message is output as an 0x26 *RxDATA* event. The address field contains the sender's address, 0x00 0x00, which is the base. The LQI value 0x35 is the signal strength of the received message. The data following the LQI value is the *Hello* text. If the remote is in transparent mode, only the *Hello* text is output.

## 4.2 Configuration Registers

A summary of the configuration registers supported by the LPR2430 series modules is presented in Sections 4.2.1 - 4.2.8 below. Note that registers are organized into banks according to similarity of function.

### 4.2.1 Bank 0 - Transceiver Setup

Bank	Reg	Name	Size in			
			R/W	bytes	Range	Default
00	00	DeviceMode	R/W/R	1	0..2	0 (remote) - must reset after write
00	01	InitialPanID	R/W	2		0xFFFF
00	03	SleepMode	R/W	1	0..2	0 (always on)
00	04	Reserved	R	2		
00	06	WakeDuration	R/W	2	full	0x0000 (disabled)
00	08	Reserved	R/W	2		
00	0A	TxPower	R/W	1	0..25	0x00 (high power)
00	0B	Reserved	R	1		
00	0C	Reserved	R	16		
00	1C	UserTag	R/W	16		LPR2430 (ASCII string)

Data is specific to each module. The R/W/R designation indicates a reset is required to activate a parameter change.

*DeviceMode* - this parameter selects the operating mode of the LPR2430 series module:

- 0 = Remote
- 1 = Reserved
- 2 = Base

The radio module must be reset before a device mode change takes effect. There can be only one base radio for a network. The other device mode setting is reserved for future use.

*InitialPanID* - this parameter is the ID of the PAN a base will use to set up a network or a remote is allowed to join. A value of 0xFFFF instructs a remote to operate in "promiscuous mode" and join any network it finds. A base will use the last two bytes of its MAC address if this parameter is set to 0xFFFF.

*SleepMode* - this parameter enables sleep mode, which may be used in conjunction with the automatic I/O reporting feature to wake up on specified triggers. Sleep mode is only available for remotes.

- 0 = Always awake
- 1 = Periodic sleep (wake on periodic timer or any I/O trigger)
- 2 = Deep sleep (wake on GPIO triggers only)

Current consumption is lowest in deep sleep mode, at the expense of having fewer wakeup choices. To prevent the possibility of putting a remote to sleep with no means of waking it back up again, a remote will not go to sleep unless at least one bit of the *IO\_ReportTrigger* register is set.

*WakeDuration* - in sleep mode, this parameter sets the length of time that a remote will wait for a response after sending an I/O report before going back to sleep, from a minimum of 1 ms to a maximum of 65 s. Parameter scaling is 1 ms/count. *WakeDuration* allows a host application at the base time to respond to a remote before it goes back to sleep. If this capability is not needed, setting this value to 0 will disable the function.

*TxPower* - this parameter sets the level of transmit output power attenuation. Only the values shown in Table 4.2.1.1 are valid:

TxPower Setting (hex/dec)	Relative Output Power	Nominal Output Power (0 dBm reference assumed for LPR2430/A)	Nominal Output Power (+18 dBm reference assumed for LPR2430ER/A)
0x00 / 0	0 dB	0 dBm	18 dBm
0x01 / 1	-1 dB	-1 dBm	17 dBm
0x03 / 3	-3 dB	-3 dBm	15 dBm
0x04 / 4	-4 dB	-4 dBm	14 dBm
0x05 / 5	-5 dB	-5 dBm	13 dBm
0x08 / 8	-7 dB	-7 dBm	11 dBm
0x0B / 11	-10 dB	-10 dBm	8 dBm
0x0F / 15	-15 dB	-15 dBm	3 dBm
0x13 / 19	-19 dB	-19 dBm	-1 dBm
0x19 / 25	-25 dB	-25 dBm	-7 dBm

Table 4.2.1.1

Power in milliwatts may be related to dBm as follows:

$$\text{Power in mW} = 10^{(\text{Power in dBm} / 10)}$$

For operation in regions of Europe where the regulatory limit is 12.7 dBm (18.6 mW), a *TxPower* setting of 0x05 must be selected on the LPR2430ER/A to ensure compliance.

## 4.2.2 Bank 1 - System Settings

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
01	00	SecurityMode	R/W/R	1	0..1	0 (security disabled) - must reset after write
01	01	NetworkKey	W/R	16	full	all 16 bytes 0x00 (NULL)
01	11	ChannelList	R/W	2	0..32767	0x7FFF (all channels 2405 to 2475 MHz)
01	13	ARQ_AccessLimit	R/W	1	0..16	4 attempts
01	14	NwkFormThreshold	R/W	1	full	0x1B (-80 dBm)
01	15	Reserved	R	1		
01	16	Reserved	R	1		
01	17	BroadcastNumReps	R/W	1	full	4
01	18	BcastMaxBackoff	R/W	1		0xFA (250 ms)
01	19	LeasePeriod	R/W	4	full	0x0000EA60 (60 s)
01	1D	RemoteLeaseEnable	R/W	1	0..1	1 (enabled)
01	1E	RelayScanRateNorm	R/W	2	full	0x7530 (30 s)
01	20	RelayScanRateFast	R/W	2	full	0x07D0 (2 s)
01	22	Reserved	R	1		
01	23	Reserved	R	1		

Data is specific to each module. The R/W/R designation indicates a reset is required to activate a parameter change.

**SecurityMode** - this parameter enables or disables the security function to encrypt over-the-air packets. A value of 1 enables encryption, a value of 0 disables encryption. The base and all remotes in a network must have the same *SecurityMode* setting.

**NetworkKey** - this parameter sets the 128-bit AES encryption key that is used to encrypt data, event reports, and GET/SET packets containing data values. All LPR2430 series radios in the network must be configured with the same key in order to interoperate. To protect the key, it is a write-only parameter for

the user. Refer to Section 2.3 on security for further information. A change to this parameter requires a restart to take effect.

*ChannelList* - this 15-bit bitmask parameter sets the list of channels that an LPR2430 series module is allowed to use. Refer to Table 4.2.1.1 for a list of channels. Note the most significant bit is not used and must be set to 0 for proper operation. For a base radio, it is recommended that only one channel be selected in the *ChannelList* mask. This provides a known channel for frequency planning purposes. If more than one channel is enabled, the base will pick one of them at random at startup, and not switch from it unless the radio is reset or power-cycled. For a remote, the channel mask specifies which channels it will look for a base on. For greatest flexibility, it is useful to set remotes for all channels, so that the entire network does not need to be reconfigured if it is necessary to assign the base to a new frequency. For faster link time, select as few channels as are required. If a radio loses link, is reset or is power cycled, it will rescan all of the channels in its *ChannelList* mask continually until its base is found. Bit 0 in the *ChannelList* mask corresponds to 2405 MHz. Bit 14 corresponds to 2475 MHz.

Channel (MHz)	ChannelList Bit Mask
2405	0000 0000 0000 0001
2410	0000 0000 000 00010
2415	0000 0000 000 00100
2420	0000 0000 0000 1000
2425	0000 0000 0001 0000
2430	0000 0000 0010 0000
2435	0000 0000 0100 0000
2440	0000 0000 1000 0000
2445	0000 0001 0000 0000
2450	0000 0010 0000 0000
2455	0000 0100 0000 0000
2460	0000 1000 0000 0000
2465	0001 0000 0000 0000
2470	0010 0000 0000 0000
2475	0100 0000 0000 0000

Table 4.2.2.1

*ARQ\_AttemptLimit* - This sets the maximum number of attempts that will be made to send a data packet on the RF link. This number applies to both the direct path and the relay path transmit attempts, so the total number of attempts is effectively twice this limit when a relay is present. A radio will exhaust all of its direct path attempts before switching to the relay path.

*NwkFormThreshold* - This parameter specifies the minimum link quality index (LQI) for a remote to consider acceptable in selecting a base. LQI is related to received signal strength as shown in the table below. The difference in values between the low-power LPR2430/LPR2430A modules and the high-power LPR2430ER/LPR2430ERA modules reflect the additional gain of the extra receiver amplifier. MURATA recommends leaving this value at its default setting. Changing it may result in unexpected network operational issues.

Received signal strength in dBm (LPR2430/A)	Received signal strength in dBm (LPR2430ER/ERA)	LQI (decimal)	LQI (hex)
special	special	255	FF
10	-2	254	FE
5	-7	242	F2
0	-12	229	E5
-5	-17	217	D9
-10	-22	204	CC
-15	-27	191	BF
-20	-32	179	B3
-25	-37	166	A6
-30	-42	154	9A
-35	-47	141	8D
-40	-52	128	80
-45	-57	116	74
-50	-62	103	67
-55	-67	90	5A
-60	-72	78	4E
-65	-77	65	41
-70	-82	53	35
-75	-87	40	28
-80	-92	27	1B
-81	-93	25	19
-85	-	15	0F
-90	-	2	02
-91	-	1	01
special	special	0	00

Table 4.2.1.2

All signal strength values above should be considered approximate ( $\pm 3$  dB). Values of 0x00 and 0xFF have special meanings.

0x00 = No LQI measured because no ACK was received.

0xFF = No LQI measured because packet was relayed.

*BroadcastNumReps* - this parameter specifies the number of times a broadcast packet will be repeated. Legal values are 0x01 to 0xFF. Setting to 0x00 will have the same effect as setting to 0x01.

*LeasePeriod* - when remote lease periods are enabled, remotes will periodically send a heartbeat to the base to renew their network registration. The base monitors these heartbeat packets to determine when a remote has left the network and to notify its host. The lease period parameter scaling is 1 ms/count. Remotes normally transmit heartbeats to the base at an interval of one-half the *LeasePeriod*. If a remote does not receive an answer from the base on a given heartbeat attempt, it will switch to a shorter heartbeat interval of one-tenth the *LeasePeriod*. Generally, the base will notify the host application that a remote has left the network with a latency of less than two-thirds of the *LeasePeriod*. Setting this value to 0 is equivalent to setting *RemoteLeaseEnable* to 0.

*RemoteLeaseEnable* - When set to 1, this parameter enables the remote lease function for a network, allowing it to monitor the connection to each remote in the network and to notify the host application with an announce message whenever a remote leaves the network. This value must be set identically for all radios in the network - base and remotes.

*RelayScanRateNorm* - this parameter specifies the normal period between relay scans. The parameter scaling is 1 ms/count. LPR2430 series modules include one-hop relay packet forwarding to enhance network performance. One-hop relay forwarding mitigates transmission problems such as multipath fading and temporary obstructions without introducing the latency and complexity inherent in a full mesh network. When a remote first joins a network, it is not aware of its neighbors and has not selected a relay node to assist with its communications. To collect this information, a remote scans for other remotes and evaluates their relay potential as a background task. A remote then reports its current relay node selection to the base for use in the base station's relay table. Since conditions often change over time, LPR2430 series modules scan periodically to keep their relay selections current.

*RelayScanRateFast* - When a remote is first turned on, it performs a fast scan of all possible network addresses to initially populate its table of potential relay partners. This parameter specifies the period between relay scans in fast mode. Parameter scaling is 1 ms/count.

#### 4.2.3 Bank 2 - Status Registers

<i>Bank</i>	<i>Req</i>	<i>Name</i>	<i>R/W</i>	<i>Size in bytes</i>	<i>Range</i>	<i>Default</i>
02	00	MacAddress	R	8	full	factory assigned
02	08	CurrNwkAddress	R	2	full	base = 0xFFFF remote = 0xFFFF
02	0A	CurrPanID	R	2	full	0xFFFF
02	0C	CurrChannel	R	1	0..16	0x0E (2475 MHz)
02	0D	LinkStatus	R	1	0..1	0x00 (not joined)
02	0E	HardwareVersion	R	1	full	0x00
02	0F	FirmwareVersion	R	1	full	0x20
02	10	FirmwareBuildNum	R	2	full	0x002F
02	12	Reserved	R	1		
02	13	LQI_Last	R	1	full	
02	14	RSSI_Last	R	1	full	
02	15	Reserved	R	2		
02	17	Reserved	R	2		
02	19	Reserved	R	1		
02	20	Reserved	R	32		
02	40	Reserved	R	64		

Bank 2 contains read-only status registers. Data is specific to each module.

*MacAddress* - this parameter returns the radio's unique 64-bit IEEE MAC address.

*CurrNwkAddress* - this parameter is the module's current network address. Two special values should be noted. The base always reports 0x0000. If a remote does not have a network address assigned it reports 0xFFFF.

*CurrPanID* - this parameter returns the PAN ID of the network the LPR2430 series module is currently connected to. A value of 0xFFFF means the radio is in promiscuous mode and scanning for a network but has not yet joined one.

*CurrChannel* - this parameter returns the current RF channel that the radio is operating on.

---

*LinkStatus* - this parameter returns the module's current connection status to the network. The following codes are defined:

- 0 = not currently part of a network
- 1 = successfully started or joined a network

*HardwareVersion* - this parameter returns an identifier indicating the LPR2430 series module type.

*FirmwareVersion* - this parameter returns the firmware version of the module in 2-digit BCD format: x.y.

*FirmwareBuildNum* - this parameter returns the firmware build number, in binary format.

*LQI\_Last* - this parameter returns the last LQI measurement made during the receipt of an RF packet with a valid CRC. Used for diagnostic purposes and troubleshooting.

*RSSI\_Last* - this parameter returns the last RSSI measurement made during receipt of an RF packet with a valid CRC. Used for diagnostic purposes and troubleshooting.

#### 4.2.4 Bank 3 - Serial Settings

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
03	00	SerialRate	R/W	1	full	0x06 (9.6 kb/s)
03	01	SerialParams	R/W	1	0..56	0x00 (8, n, 1)

Data is specific to each module.

*SerialRate* - this parameter selects the UART baud rate. The allowed values are:

Hex Value	Baud Rate
0x03	1200
0x04	2400
0x05	4800
0x06	9600
0x07	19200
0x08	38400
0x09	57600
0x0A	76800
0x0B	115200

Table 4.2.4.1

*SerialParams* - this parameter selects the UART parity and stop bit options:

Hex value	Parity	Stop bits
0x00	none	1
0x08	none	2
0x20	even	1
0x28	even	2
0x30	odd	1
0x38	odd	2
other values	reserved	reserved

Table 4.2.4.2

---

Note that 8-bit data with no parity is capable of carrying 7-bit data with parity for compatibility without loss of generality for legacy applications that may require it.

#### 4.2.5 Bank 4 - Host Protocol Settings

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
04	00	ProtocolMode	R/W	1	0..2	0x00 (host protocol for point-to-point)
04	01	ProtocolOptions	R/W	1	0..3	0x03 (Error and Announce packets enabled)
04	02	TxTimeout	R/W	1	full	0x32 (50 ms)
04	03	Reserved	R	1		
04	04	Reserved	R	1		
04	05	TransLinkAnnEn	R/W	1	0..1	0x00 (<LINK> and <DROP> text inserts disabled)
04	06	EscapeSequenceEn	R/W	1	0..1	0x01 (escape sequence enabled at any time)
04	07	TransDefaultAddr	R/W	2	full	0xFFFF for a base, 0x0000 for a remote

This bank contains options for the host serial protocol. Data is specific to each module.

*ProtocolMode* - this parameter selects transparent point-to-point (0x00), transparent point-to-multipoint (0x01), or protocol mode (0x02). The default is 0x00, transparent point-to-point mode, meaning the LPR2430 series module transmits the characters sent to it transparently, without requiring the host to understand or conform to the LPR2430's serial protocol. This setting is recommended for point-to-point applications such as wire replacements where another serial protocol may already exist. Transparent mode also works well in applications where the data being sent includes addressing information to allow host devices connected to LPR2430 nodes to determine if the data is for them. Setting the base for transparent point-to-multipoint in this environment will cause all of the remotes to receive the data and send it their local host. The local hosts will parse the data to determine if it is for them.

Setting this parameter to 0x02 enables the LPR2430 protocol mode, which is the most efficient mode for point-to-multipoint applications. In protocol mode, the module network addresses are used to send specific packets of data to specific remotes.

It is not necessary to use the same protocol mode for all LPR2430 modules in a network. For example, it is frequently useful to configure all the remotes for transparent mode and the base for protocol mode. Since by default, data from a remote is sent to the base, there is no need to explicitly address a transmission to the base. It is also possible to establish transparent communications between two remotes by changing the *TransDefaultAddr* parameter at each remote to the other remote's network address. Note that the host can switch a module from transparent mode to protocol mode and back by using the *Enter Protocol Mode* escape sequence. See Section 4.1.3 for more information.

*ProtocolOptions* - this parameter is a bitmask that selects various options for the protocol mode. The default is 0x03:

- bits 2..7 Reserved
- bit 1 Enable error packets
- bit 0 Enable announce packets

*TxTimeout* - when a module is in transparent mode, this parameter specifies the length of time of no activity on its RxD line that the module will wait before deciding the message is complete and sending it over the RF link. This is used to prevent any application packet from being unnecessarily split into multiple RF packets. The value is entered in milliseconds. Generally, this value should be set to at least twice the length of a serial character at the currently selected baud rate. If the user's host application is subject to occasional gaps in data that it sends this value should be set higher.

---

*TransLinkAnnEn* - this parameter enables a link announcement function for transparent mode. Whenever a link is acquired or dropped, the strings "<LINK>" or "<DROP>" are sent to the host.

*EscapeSequenceEn* - this parameter enables or disables the escape sequence (*EnterProtocolMode* command) which can be used to switch from transparent mode to protocol mode. Valid settings are 0 = one chance at startup, 1 = enabled at any time. Enabled at anytime is the default.

*TransDefaultAddr* - when an LPR2430 series module is configured for transparent mode, this parameter sets the address where the module will send data packets. For a base, the default will be 0xFFFF, for a remote the default will be 0x0000. To send data from one remote to another remote while in transparent mode, enter the network address of the desired remote into this parameter location.

#### 4.2.6 Bank 5 - I/O Peripheral Registers

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
05	00	GPIO0	R/W	1	0..1	
05	01	GPIO1	R/W	1	0..1	
05	02	GPIO2	R/W	1	0..1	
05	03	GPIO3	R/W	1	0..1	
05	04	GPIO4	R/W	1	0..1	
05	05	GPIO5	R/W	1	0..1	
05	06	ADC0	R	2	0..2047	
05	08	ADC1	R	2	0..2047	
05	0A	ADC2	R	2	0..2047	
05	0C	ADCBatt	R	2	0..2047	
05	0E	Temperature	R	1	0..255	
05	0F	EventFlags	R	2	0..255	
05	11	PWM0	R/W	2	full	0x00 (0 V output)
05	13	PWM1	R/W	2	full	0x00 (0 V output)

Data is specific to each module.

This bank controls the LPR2430's user-configurable I/O. Values are refreshed upon reading of any register in the bank, and any time any IO is affected, such as event reporting, ADC interval sampling, etc. To simplify block write operations, writes to the read-only *ADC0..2*, *ADCBatt*, *Temperature* and *EventFlags* registers are ignored and do not generate an error.

*GPIO0..5* - writing to these registers sets the output states of pins that are configured as outputs. Reading these registers returns the current levels detected on the corresponding pins.

*ADC0..2* - reading one of these registers returns the latest ADC measurement in 11-bit unsigned integer format. The ADC sampling interval is set by *ADC\_SampleInterval*. Manually reading any one of these registers also refreshes the values of all three ADCs.

*ADCBatt* - reading this register returns an ADC reading of the module's supply voltage ( $V_{Batt}$ ), in 11-bit unsigned integer format. The actual voltage may be derived from the ADC reading according to the following equation:

$$V_{Batt} = (3.750 * \text{ADC}_{\text{Batt}}) / 2047$$

*Temperature* - reading this register returns the value of the CC2430's internal temperature sensor, in degrees Celsius. Relative error over temperature is  $\pm 2^{\circ}\text{C}$ .

*EventFlags* - when using the automatic I/O reporting feature, this register indicates which I/O events were triggered since the last report message by setting the appropriate flag bits to 1. Once the report is sent, the flags are cleared to 0:

<i>bits 15..8</i>	<i>Reserved</i>
<i>bit 7</i>	ADC2 high/low thresholds
<i>bit 6</i>	ADC1 high/low thresholds
<i>bit 5</i>	ADC0 high/low thresholds
<i>bit 4</i>	Periodic report timer
<i>bit 3</i>	GPIO3 falling edge
<i>bit 2</i>	GPIO2 falling edge
<i>bit 1</i>	GPIO1 falling edge
<i>bit 0</i>	GPIO0 falling edge

These flags provide an indication of which triggers generated the current report. This is useful when multiple triggers fire before the report is sent. This reduces unnecessary RF traffic by permitting only one report to be sent. Sending one report also provides better battery life.

*PWM0..1* - these registers set the PWM outputs. Both PWM registers are 16-bit. However, only the upper 8 bits of PWM0 are significant. All of the bits of PWM1 are significant. The two PWMs also have different periods, 1.024 ms for PWM0 and 2.048 ms for PWM1. Thus each bit of PWM0 =  $1.024\text{ ms}/256 = 4\text{ }\mu\text{s}$  while each bit of PWM1 =  $2.048\text{ ms}/65,536 = 30\text{ ns}$ . Thus a PWM0 value of 0x80 will produce a 50% duty cycle square wave with a period of 1.024 ms, but it takes a PWM1 value of 0x8000 to produce a 50% duty cycle square wave with a period of 2.048 ms. A simple onboard low-pass filter is provided on the module for both of these outputs, but for best results, these outputs should be buffered and provided with additional filtering. Note that PWM1 shares a pin with GPIO2 and is only enabled when the appropriate bit is set in the GPIO\_Alt register.

#### 4.2.7 Bank 6 - I/O Setup

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
06	00	GPIO_Dir	R/W	1	0..63	0x00 (all inputs)
06	01	GPIO_Init	R/W	1	0..63	0x00 (all zeros)
06	02	GPIO_Alt	R/W	1	0..63	0x00 (no alternate functions)
06	03	GPIO_EdgeTrigger	R/W	1	full	0x01 (GPIO0 falling edge trigger enabled)
06	04	GPIO_SleepMode	R/W	1	0..1	0x00 (all GPIOs inputs/internal pull-ups when sleeping)
06	05	GPIO_SleepDir	R/W	1	0..63	0x00 (all inputs)
06	06	GPIO_SleepState	R/W	1	0..63	0x00 (all zeros)
06	07	PWM0_Init	R/W	2	10	0x0000 (0 V output)
06	09	PWM1_Init	R/W	2	10	0x0000 (0 V output)
06	0B	ADC_Mode	R/W	1	0..2	0x00 (V <sub>DD</sub> referenced full scale)
06	0C	ADC_SampleInterval	R/W	1	full	0x00 (automatic sampling disabled)
06	0E	ADC0 ThresholdLo	R/W	2	0..2047	0x0000 (0 V)
06	10	ADC0 ThresholdHi	R/W	2	0..2047	0x07FF (ADC full scale)
06	12	ADC1 ThresholdLo	R/W	2	0..2047	0x0000 (0 V)
06	14	ADC1 ThresholdHi	R/W	2	0..2047	0x07FF (ADC full scale)
06	16	ADC2 ThresholdLo	R/W	2	0..2047	0x0000 (0 V)
06	18	ADC2 ThresholdHi	R/W	2	0..2047	0x07FF (ADC full scale)
06	1A	IO_ReportInterval	R/W	4	full	0x00007530 (30 s)
06	1E	IO_ReportTrigger	R/W	1	full	0x01 (GPIO0)
06	1F	IO_ReportAddress	R/W	2	full	0x0000 (base station)
06	21	IO_ReportMode	R/W	1	0..1	0x01 (send report as an RxEvent packet)
06	22	IO_BindingEnable	R/W	1	0..1	0x00 (RF mapping of I/O disabled)

This bank contains setup options for user-configurable I/O. Data is specific to each module.

---

*GPIO\_DIR* - this parameter is a bitmask that sets whether each GPIO is an input (0 bit) or an output (1 bit). The default is all inputs, 0x00.

*GPIO\_Init* - this parameter is a bitmask that sets the initial value for GPIOs set as outputs.

*GPIO\_Alt* - this parameter is used to enable alternate GPIO pin functions, such as RS-485 driver enable.

Bit	Option
7	-
6	-
5	GPIO5 → /HOST_RTS
4	GPIO4 → /HOST_CTS
3	GPIO3 → RS-485 driver
2	GPIO2 → PWM1
1	Reserved
0	GPIO0 → ADC_REF

Table 4.2.7.1

GPIO5 → /HOST\_RTS: a 1 in this bit position causes GPIO5 to be used as a flow control input from the host to the module to start or stop serial data flow from the module's UART. Note that per convention, the RTS signal (request to send) is active low - a bit value of 0 enables data flow from the module.

GPIO4 → /HOST\_CTS: setting this bit position to 1 causes GPIO4 to be used as a flow control output to the host from the module to start or stop serial data flow from the host. Note that per convention, the CTS signal (clear to send) is active low; a bit value of 0 enables data flow from the host.

GPIO3 → RS-485 TX Enable: setting this bit position to 1 causes GPIO3 to be used as an active low transmit enable suitable for controlling an RS-485 or other half-duplex bus driver. When enabled, GPIO3 will go low at the beginning of the first start bit, and return to an idle high state at the end of the last stop bit.

GPIO2 → PWM\_1: setting this position to 1 configures GPIO2 as a second PWM output controlled by the PWM1 register.

GPIO0 → ADC\_REF: setting this bit position to 1 configures GPIO0 as a voltage reference input for the ADC. The appropriate ADC mode must also be selected in the *ADC\_Mode* register (discussed later in this section) for the input to be functional.

*GPIO\_EdgeTrigger* - this bit mask parameter enables any combination of GPIO0..3 to be used as interrupts to wake a sleeping module:

<i>bit 7..4</i>	Reserved
<i>bit 3</i>	0 - disabled; 1 - GPIO3 falling edge enabled
<i>bit 2</i>	0 - disabled; 1 - GPIO2 falling edge enabled
<i>bit 1</i>	0 - disabled; 1 - GPIO1 falling edge enabled
<i>bit 0</i>	0 - disabled; 1 - GPIO0 falling edge enabled

For each enabled GPIO, a falling edge wakes up the module and holding the GPIO low keeps the module awake. To have the interrupt generate an IO\_Report, the appropriate bit in the IO\_ReportTrigger must be set.

---

*GPIO\_SleepMode* - when this parameter is set to 0x01, it enables the use of the *GPIO\_SleepDir* and *GPIO\_SleepState* registers to configure a direction and state for each GPIO whenever a device is asleep. This function is disabled by default, in which case all GPIOs will be set as inputs with internal pull-ups whenever the module is asleep.

*GPIO\_SleepDir* - when *GPIO\_SleepMode* is enabled, the six LSBs of this byte are used to set the direction of the GPIOs during a device's sleep period. A 0 bit value sets a GPIO as an input when sleeping, a 1 bit value sets a GPIO as an output when sleeping. This enables the user to provide alternate configurations during sleep that help minimize current consumption, etc. Bits 0..5 correspond to GPIO0..GPIO5.

*GPIO\_SleepState* - when *GPIO\_SleepMode* is enabled, the six LSBs of this byte are used to set the state of any GPIOs configured as outputs by the *GPIO\_SleepDir* parameter during sleep. This enables the user to specify alternate configurations during sleep that help minimize current consumption, etc. Bits 0..5 correspond to GPIO0..GPIO5.

*PWM0\_Init* - This sets the initial value for PWM0 at startup.

*PWM1\_Init* - This sets the initial value for PWM1 at startup (assuming PWM1 is enabled).

*ADC\_Mode* - This register selects the reference voltage that will be used for the internal ADCs.

Hex Value	ADC Reference
0x00	Internal $V_{DD}$ (3.3 V)
0x01	Internal $V_{REF}$ (2.5 V)
0x02	External $V_{REF}$ (GPIO0)
other values	Reserved

Table 4.2.7.2

If the external  $V_{REF}$  is selected, it must also be enabled in the *GPIO\_Alt* register for proper operation.

*ADC\_SampleInterval* - this parameter sets the interval at which the ADC will be sampled and threshold levels will be checked for purposes of triggering an I/O event. Parameter scaling is 1 ms/count.

Setting the parameter to 0x00 disables automatic sampling. This parameter is different than the *I/O\_ReportInterval* in that no report is generated on this interval unless the ADC sample value causes an interrupt to be generated. As long as the ADC threshold triggers are not set or the ADC values are within the ADC threshold limits, the module will return to sleep after taking the samples.

*ADC0..2\_ThresholdLo/Hi* - these parameter values define thresholds to trigger an I/O report based on ADC measurements. If I/O reporting is enabled, a single *RxEVENT* report containing the contents of the I/O bank is generated whenever a threshold is crossed. Reporting is edge-triggered with respect to threshold boundaries, not level-triggered. If the measurement remains there, additional reports are not triggered until the value exceeds the threshold again. A second interrupt is not generated when the signal returns to the allowed range. The thresholds are met whenever one of the following inequalities is satisfied:

$$\begin{aligned} \text{ADC}_x &< \text{ADC}_x\_\text{ThresholdLo} \\ \text{ADC}_x &> \text{ADC}_x\_\text{ThresholdHi} \end{aligned}$$

---

*IO\_ReportTrigger* - when a selected trigger source is enabled, a trigger event will cause the remote to send an *RxEVENT* message to its base containing the entire current values of the I/O Register Bank from GPIO0 up to and including the *EventFlags*, but not the PWM settings which are output only.

<i>bit 7</i>	ADC2 high/low thresholds
<i>bit 6</i>	ADC1 high/low thresholds
<i>bit 5</i>	ADC0 high/low thresholds
<i>bit 4</i>	Periodic report timer
<i>bit 3</i>	GPIO3 falling edge
<i>bit 2</i>	GPIO2 falling edge
<i>bit 1</i>	GPIO1 falling edge
<i>bit 0</i>	GPIO0 falling edge

To avoid inadvertently making an LPR2430 series module non-responsive, a remote will not sleep unless at least one bit in the *IO\_ReportTrigger* register is set. Note that the GPIOs must have first been enabled in the *GPIO\_EdgeTrigger* parameter for them to wake a sleeping module and send a report.

*IO\_ReportInterval* - When the periodic timer trigger for I/O reporting is enabled, it sets the interval between reports. The value is entered in milliseconds, and the default is once every 30 seconds. The practical minimum is 100 ms, and the maximum interval is over 49 days. See Section 2.5.2 for details of using the periodic report timer with I/O binding.

*IO\_ReportAddress* - Sets the network address that I/O event packets will be sent to (default is the base, 0x0000). Normally, this should be left at the default. However, if the *IO\_BindingEnable* feature is enabled, this should be set to the address of the radio to which it is intended to be bound.

*IO\_ReportMode* - When I/O reporting is enabled, this function is used to control the type of report that is sent whenever a GPIO is triggered. This register is a bit field with the following individually selectable switches:

<i>bits 7..3</i>	Reserved
<i>bit 1</i>	LocalReportEnable
<i>bit 0</i>	ReportType

LocalReportEnable: a 1 in this bit position causes a remote to output a report on its local serial port as well as sending an RF message. The default is a bit value of 0 (off).

ReportType: Controls the type of message sent when an I/O trigger occurs. A 1 in this bit position causes an *RxEVENT* packet to be sent containing an I/O report. A 0 in this bit position causes a registration packet to be sent with no I/O report. The default is a bit value of 1.

Note that only GPIO triggers are supported by this function. Timer and ADC triggers always send an *RxEVENT* report, and never produce a local serial report or a registration packet.

*IO\_BindingEnable* - setting this parameter to 0x01 configures a module to map the states of GPIO0 and GPIO1, and the value of ADC0 in an I/O report *addressed to it from another module* to its GPIO2, GPIO3 and PWM0 outputs, as shown in Table 4.2.7.3. This function can be enabled independently from the other I/O reporting functions that control when a module transmits its reports. An LPR2430 module can send its I/O report to only one other module address. *IO\_BindingEnable* is typically used in a point-to-point application with a single base and remote, although other topologies are possible. A module that receives an I/O report from another module can have its I/O report addressed back to that module, to a different

module, or to the base. It is only necessary to set the *IO\_BindingEnable* on modules that will map another module's I/O report.

Received I/O	Mapped to Output I/O
GPIO0	GPIO2
GPIO1	GPIO3
ADC0	PWM0

Table 4.2.7.3

When I/O binding is enabled on a module, GPIO0 and GPIO1 are forced to be inputs, GPIO2 and GPIO3 are forced to be outputs, and the *GPIO\_Dir* and *GPIO\_Alt* settings are ignored for GPIO0..GPIO3.

I/O binding is useful in creating “wireless” analog voltage and switch connections. For example, assume that a remote with network address 0x0002 has a alarm (logic level) connected to GPIO1. To monitor this alarm at another remote with the network address of 0x0005:

1. Connect the alarm signal to GPIO1 on remote 0x0002
2. Set the *IO\_ReportTrigger* on remote 0x0002 to 0x02 (trigger on input GPIO1)
3. Set the *IO\_ReportMode* on remote 0x0002 to 0x01 (send I/O report as *RxEVENT* packet)
4. Set the *IO\_ReportAddress* on remote 0x0002 to 0x0005 (send I/O report to remote 0x0005)
5. Set *IO\_BindingEnable* on remote 0x0005 to 0x01 (enable I/O binding)
6. Monitor the alarm signal on remote 0x0005 output GPIO3

See Section 2.5.2 for addition details on using I/O binding.

#### 4.2.8 Bank FF - Special Functions

<u>Bank</u>	<u>Reg</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default</u>
FF	00	UcReset	W	1	0	write 0x00 to reset module
FF	05	AddressWipe	R/W/R	1	full	any write wipes address after reset
FF	FF	MemorySave	W	1	0..1	write 0x00 to load factory defaults write 0x01 to save to EEPROM

This bank contains three user-accessible functions, *UcReset*, *MemorySave* and *AddressWipe*. No attempt should be made to read or write to other parameters in this bank. The functions are specific to each module.

*UcReset* - writing a value of 0x00 to this location forces a software reset of the module's microcontroller causing the module to execute a soft reboot. Writing any other value returns an error. No reply packet, either local or over-the-air, is generated.

*AddressWipe* - on a base, writing any value to this register will clear its table of MAC address - network address associations for remotes. A reset of all radios in the network must be performed after this operation in order to assign new addresses to the remotes. Writing to this register has no effect on a remote.

*MemorySave* - writing a 0x00 to this location restores all registers to factory defaults. Writing a 0x01 to this location commits the current register settings to EEPROM. When programming registers, all changes are temporary until this command is executed. This includes restoring the factory defaults - they are only temporary until 0x01 is written to this location.

#### 4.2.9 Protocol Mode Configuration/Sensor Message Examples

In this example, the host configures the base through the serial port to transmit 10 dBm (10 mW) of RF power using the *SetRegister* command, 0x04. A transaction ID of 0x05 is used. The *TxPower* parameter is stored in bank 0x00, register 0x0A. A one-byte parameter value of 0x08 selects the 10 dBm (10 mW) power level. The protocol formatting for the command is:

SOP	Length	TransID	PktType	Reg	Bank	Span	Val
0xFB	0x06	0x05	0x04	0x0A	0x00	0x01	0x08

Note the order of the bytes in the command argument: register, bank, span, parameter value. When the base receives the command it updates the parameter setting and returns a *SetRegisterReply* message as follows:

SOP	Length	TransID	PktType
0xFB	0x02	0x05	0x14

In order for this new RF power setting to persist through a base power down, *MemorySave* must be invoked. This is done by setting the one-byte parameter in register 0xFF of bank 0xFF to 0x01 with another *SetRegister* command, with the transaction ID incremented by one:

SOP	Length	TransID	PktType	Reg	Bank	Span	Val
0xFB	0x06	0x06	0x04	0xFF	0xFF	0x01	0x01

The base will write the current parameter values to EEPROM and return a *SetRegisterReply* message:

SOP	Length	TransID	PktType
0xFB	0x02	0x06	0x14

The change will not take effect until the module is reset or power-cycled.

In this example, the base host requests an ADC1 reading from a remote using the *GetRemoteRegister* command, 0x0A. A transaction ID of 0x07 is used. Since this request is being sent over the air, the network address of the remote is used to address the request - in this example the remote network address is 0x0008. The current ADC1 measurement is read from register 0x08 in bank 0x05. The ADC reading spans two bytes. The protocol formatting for this command is:

SOP	Length	TransID	PktType	Addr	Addr	Reg	Bank	Span
0xFB	0x07	0x07	0x0A	0x08	0x00	0x08	0x05	0x02

Note the remote network address 0x0008 is entered in Little-Endian byte order, 0x08 0x00. The ADC reading is returned in a *GetRemoteRegisterReply* message:

SOP	Length	TransID	PktType	TxStatus	Addr	Addr	LQI	Reg	Bank	Span	Lo Val	Hi Val
0xFB	0x0A	0x07	0x1A	0x00	0x08	0x00	0x35	0x08	0x05	0x02	0xFF	0x02

Substantial information is returned in the message. The last two bytes of the message give the ADC reading in Little-Endian format, 0xFF 0x02. The ADC reading is thus 0x02FF. The LQI value is the byte following the address, 0x35, indicating a received signal strength of -70 dBm on an LPR2430 or -82 dBm on an LPR2430ER or LPR2430ERA. The *TxStatus* byte to the right of the *GetRemoteRegisterReply* Packet Type is 0x00, showing the packet was acknowledged on the RF channel.

#### 4.2.10 Protocol Mode Event Message Example

In this example, the *IO\_ReportInterval* is set to 10 seconds and the *periodic report timer* bit in the *IO\_ReportTrigger* parameter is set on. The network address of the remote is 0x0008. This causes event messages to be sent from this remote every 10 seconds. The *IO\_ReportInterval* and the *IO\_ReportTrigger* parameters are loaded using *SetRemoteRegister* commands. The command to set the *IO\_ReportInterval* to 10 seconds using transaction ID of 0x09 is:

SOP	Length	TransID	PktType	Addr	Addr	Reg	Bank	Span	Lo Val	Val	Val	Hi Val
0xFB	0x0B	0x09	0x0B	0x08	0x00	0x1A	0x06	0x04	0x10	0x27	0x00	0x00

The *IO\_ReportInterval* parameter starts in location 0x1A of bank 0x06. The report interval scaling is 1 ms/count, so a 10 second report interval is 10,000 units or 0x00002710 (Little-Endian format 10 27 00 00). The *IO\_ReportInterval* parameter is updated and *SetRemoteRegisterReply* is returned:

SOP	Length	TransID	PktType	TxStat	Addr	Addr	LQI
0xFB	0x06	0x09	0x1B	0x00	0x08	0x00	0x35

The command to set the *periodic report timer* bit in *IO\_ReportTrigger* using a transaction ID of 0x0A is:

SOP	Length	TransID	PktType	Addr	Addr	Reg	Bank	Span	Val
0xFB	0x08	0x0A	0x0B	0x08	0x00	0x1E	0x06	0x01	0x10

The *periodic report timer* bit in *IO\_ReportTrigger* is located in bit position four (00010000b) or 0x10. The *IO\_ReportTrigger* parameter is updated and *SetRemoteRegisterReply* is returned:

SOP	Length	TransID	PktType	TxStat	Addr	Addr	LQI
0xFB	0x06	0x0A	0x1B	0x00	0x08	0x00	0x35

The remote will start sending event messages on 10 second intervals as shown in the log records below:

```
FB 16 80 28 08 00 35 00 01 00 01 00 00 7A 04 36 05 21 01 09 07 1C 10 00
FB 16 81 28 08 00 35 00 01 00 01 00 00 7C 04 36 05 25 01 09 07 1C 10 00
FB 16 82 28 08 00 35 00 01 00 01 00 00 7B 04 35 05 20 01 08 07 1C 10 00
FB 16 83 28 08 00 35 00 01 00 01 00 00 7B 04 35 05 23 01 06 07 1C 10 00
```

SOP	Length	TransID	PktType	Addr	Addr	LQI	Data
0xFB	0x16	0x80	0x28	0x08	0x00	0x35	

GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	ADC0	ADC1	ADC2	BATT	Temp	Flags
0x00	0x01	0x00	0x01	0x00	0x00	0x7A	0x04	0x36	0x05	0x21	0x01

The *IO\_ReportTrigger* generates *RxEVENT* messages (*PktType* 0x28). The message payload consists of the first 17 bytes in Bank 5, including the state of GPIO0 through GPIO5, the input voltages measured by ADC0 through ADC2, the battery voltage, the temperature, and the state of the event flags. Note the ADC readings and the event flags are presented in Little-Endian order. The remote is assumed to be always on in this example. If the remote is placed in periodic sleep mode (*SleepMode* = 1), a suitable value of the *WakeDuration* parameter should be set to allow the base application to analyze the I/O report and send back a command to the remote as needed.

## 5.0 LPR2430 Series Developer's Kits



Figure 5.1.1

### 5.1 Items Supplied in a Kit:

- Four LPR2430, LPR2430A, LPR2430ER, or LPR2430ERA radios, two of which are installed on developer boards, labeled Base and Remote
- Two patch antennas, and two dipole antennas with MMCX to SMA-R adaptor cables, LPR2430 and LPR2430ER only (antennas are built into the LPR2430A and LPR2430ERA radios)
- Two 9 V wall-plug power suppliers, 120/240 VAC, plus two 9 V batteries
- Two RJ-11 cables with DB-9F adaptors, LPR2430 and LPR2430ER kits
- Two Cat 5 Ethernet cables with RJ-45/DB-9F adaptors, LPR2430A and LPR2430ERA kits
- Two A/B USB cables
- One LPR2430DK/LPR2430ADK/LPR2430ERDK/LPR2430ERADK documentation and software CD

### 5.2 Additional Items Needed:

- One PC with Microsoft Windows XP or Vista Operating System

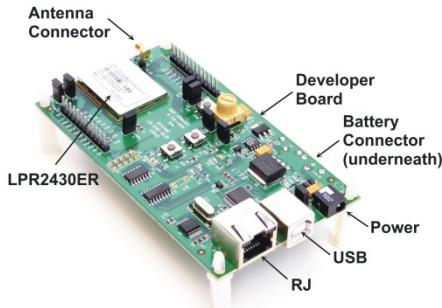


Figure 5.3.1

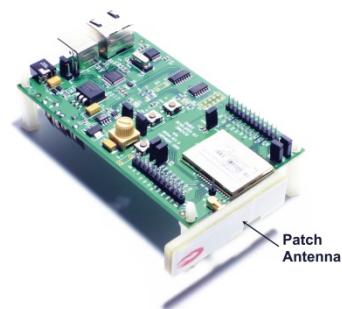


Figure 5.3.2

### 5.3 Developer Kit Setup and Testing:

1. Observe ESD precautions when handling the LPR2430 series developer boards. The developer boards can be powered with either the 9 V wall-plug power supplies or the 9 V batteries. If using the wall-plug power supplies, install an AC plug on each supply and connect the power supplies to the developer boards. See Figure 5.3.1 for the location of the power connector. Do not plug in the power supplies at this time. If using the batteries, do not install them yet.
2. For the LPR2430 and LPR2430ER kits, referring to Figures 5.3.1 and 5.3.2, install a patch antenna on each developer board antenna connector. The antenna “snaps” onto the connector with moderate pressure. Antennas are built into the radio modules for the LPR2430A and LPR2430ERA kits.
3. As shown in Figure 5.3.1, there are two serial connectors on the developer boards. The RJ-45 connector or RJ-11 connector provides an RS232 interface. The USB connector provides an optional interface. Labels on the bottom of the boards indicate which board is the Base and which is the Remote.
- 4a. If the PC has a serial port, use one of the Cat 5 cables with an RJ-45/DB-9F adaptor or one of the RJ-11 cables to make a connection to the Base. Then power up the Base by plugging in the power supply or installing the battery. If using a USB connection, follow step 4b instead of 4a.
- 4b. The USB interface is based on an FT232RL serial-to-USB converter IC manufactured by FTDI. The FT232RL driver files are located in the *i386* and *AMD64* folders on the kit CD, and the latest version of the drivers can be downloaded from the FTDI website, [www.ftdichip.com](http://www.ftdichip.com). The drivers create a virtual COM port on the PC. Power up the Base by plugging in its power supply or installing a battery. Then connect the Base to the PC with one of the USB A/B cables. The PC will find the new USB hardware and open a driver installation dialog window. Enter the letter of the PC drive holding the kit CD and click *Continue*. The installation dialog will run twice to complete the driver installations.

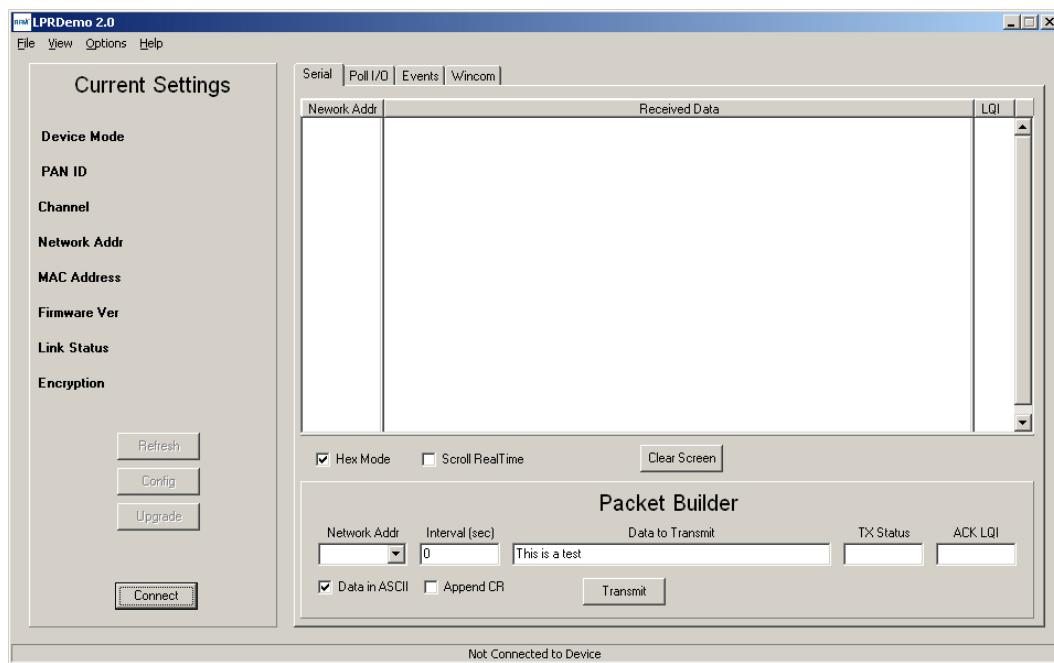


Figure 5.3.3

5. Copy LPRDemo2.exe from the *Programs* folder on the kit CD to a convenient folder on the PC. This program runs using ordinary Windows resources and does not require any framework installations, registry entries, etc., to run. Start LPRDemo2. The start-up window is shown in Figure 5.3.3.

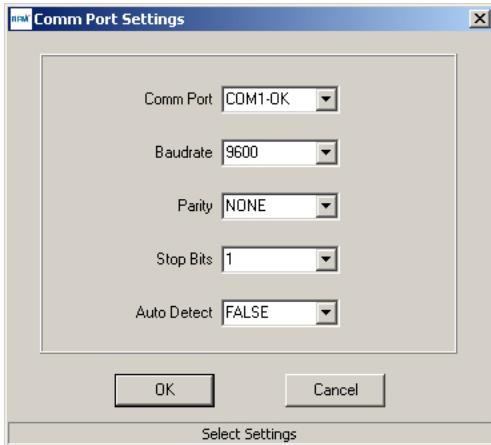


Figure 5.3.4

6. Click on the *Connect* button in the lower left of the main LPRDemo2 window to open the *Comm Port Settings* dialog window, as shown in Figure 5.3.4. Set the *Baudrate* to 9600 b/s. Set the *Comm Port* to match the serial port connected to the Base, either the RS232 serial port or the USB virtual serial port. Then click *OK* to activate the serial connection.

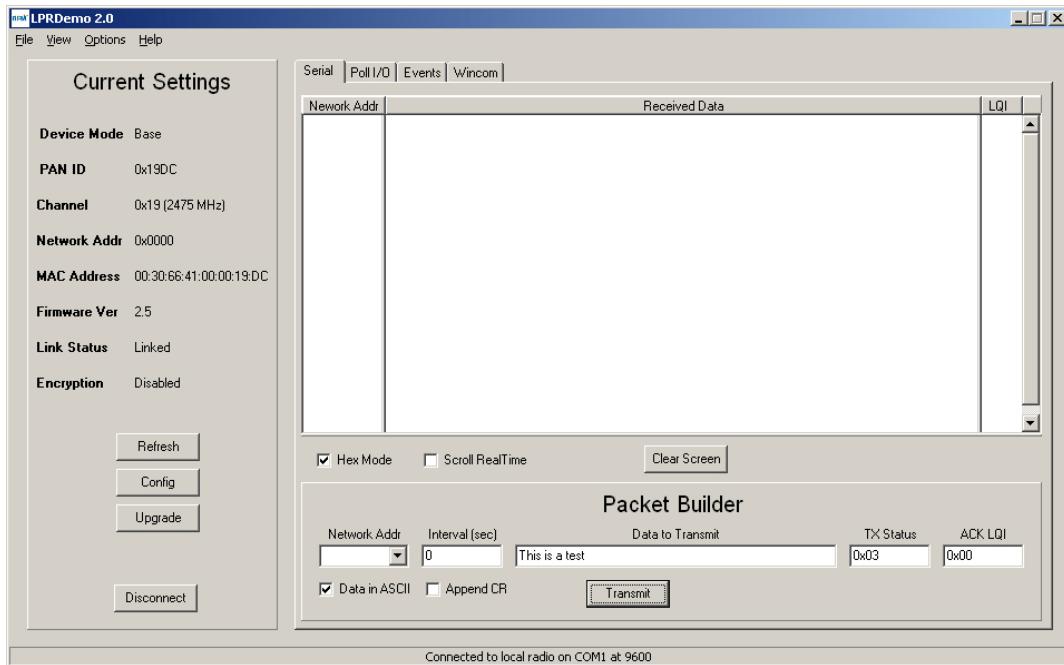


Figure 5.3.5

7. At this point the program will collect and display data from the Base under *Current Settings* on the left side of the main window, as shown in Figure 5.3.5.

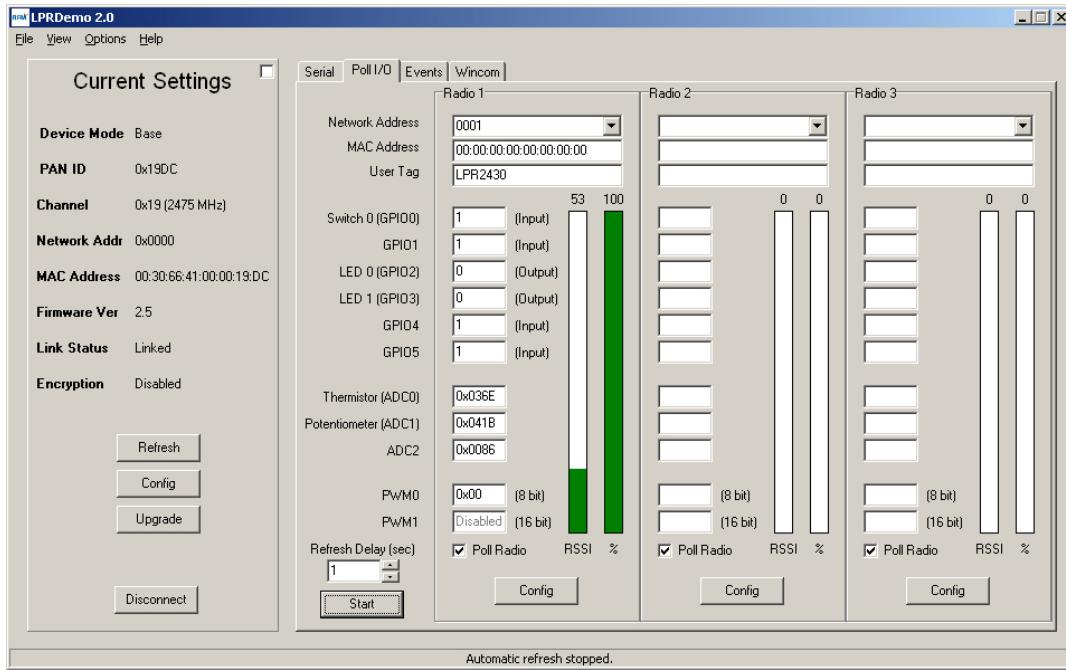


Figure 5.3.6

8. Next power up the Remote by plugging in its power supply or installing a battery. Click on the *Poll I/O* tab. Figure 5.3.6 shows the Poll I/O screen. Select the Remote's network address, 0001, in the *Network Address* drop-down box for *Radio 1*. If the address is not present, wait a few seconds to give the remote time to register with the base.

9. Click on the *Start* button. Data from the Remote will be displayed under *Radio 1*, including bar graphs of received signal strength (*RSSI*), and percent of successful requests-replies (%). Note that turning the pot on the Remote developer board will change the *Potentiometer (ADC1)* reading, and holding Switch 0 down will change the *Switch 0 (GPIO0)* state to 0. Setting *LED 0 (GPIO2)* to a 1 state will turn on LED 0. The developer kit is now ready for use.

If any difficulty is encountered in setting up your LPR2430 Series Developer's Kit, contact MURATA's module technical support group. The phone number is +1.678.684.2000. Phone support is available from 8:30 AM to 5:30 PM US Eastern Time Zone, Monday through Friday. The E-mail address for module technical support is [tech\\_sup@murata.com](mailto:tech_sup@murata.com).

## 5.4 Developer Board Features

A schematic of the LPR2430/LPR2430ER developer board is provided in the Appendix Section 7.4. The location of key components is shown in Figures 5.4.1 and 5.4.2.

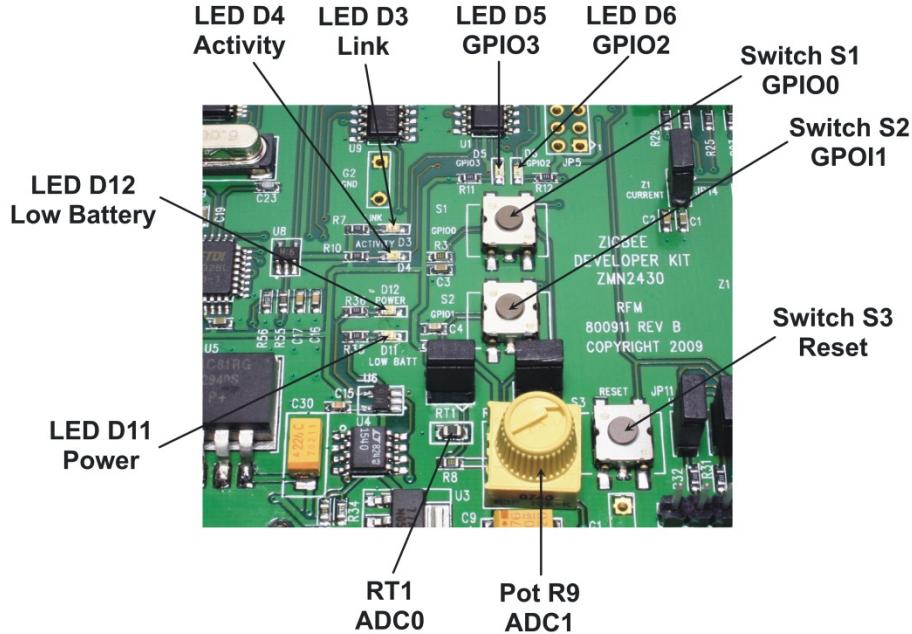


Figure 5.4.1

Momentary-contact buttons S1 and S2 set the logic input states to GPIO0 and GPIO1 respectively. Pressing a button presents a logic low to its GPIO input, otherwise the input is logic high. Button S3 provides a hardware reset to the radio. Potentiometer R9 provides an adjustable voltage to the input of ADC1. The voltage to ADC1 decreases as the R9 is turned clockwise. Thermistor RT1 is part of a voltage divider that provides a temperature dependent voltage to the input of ADC0. The voltage to ADC0 decreases with increasing temperature.

The Link LED D3 lights on a base station when it finds a clear channel to set up its network (PAN). On remotes, the Link LED illuminates when communication is established with the base. The Activity LED D4 blinks on a base or remote when a packet is transmitted or received. Power LED D11 illuminates when the developer board is receiving adequate voltage for normal operation. The Power LED will go off and the Low Battery LED D12 will illuminate as the input voltage to the developer board drifts toward its lower operating limit. Note that the silkscreen on some developer boards have the label for D11 and D12 switched. LED D5 is illuminated when GPIO3 is configured as a logic high output, and LED D6 is illuminated when GPIO2 is configured as a logic high output.

The main silkscreen label on the LPR2430A and LPR2430ERA developer board is “802.15.4 Developer Kit, LPR2430.” The label on the LPR2430 and LPR2430ER developer board is “ZigBee Developer Kit, ZMN2430”. The LPR2430 and LPR2430ER modules on the developer boards have a different form factor than the LPR2430 and LPR2430ER modules included in the kit but are otherwise identical.

Note that while the LPR2430A and LPR2430ERA developer boards support hardware flow control through the RJ-45 RS-232 connector, the LPR2430 and LPR2430ER developer boards only support

hardware flow control through JP-1 pins 9 (RTS) and 10 (CTS) since they use a 4-pin RJ-11 connector for the RS-232 interface. Hardware flow control is the alternate function for GPIO4 and GPIO5 and must be enabled through bits 4 and 5 of the *GPIO\_Alt* parameter.

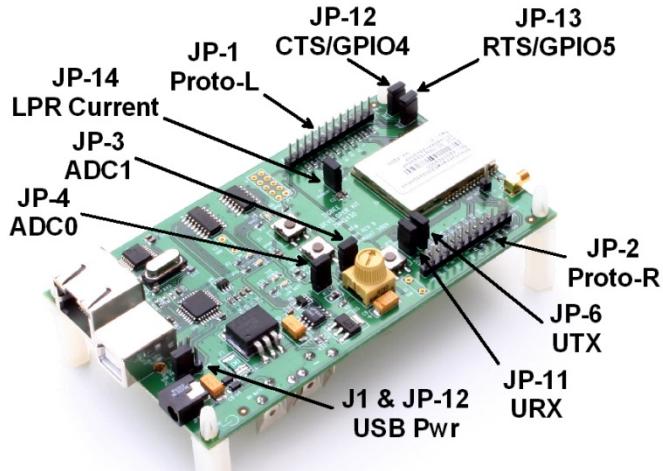


Figure 5.4.2

There are a number of jumper-configurable options on the developer board, and two jumper strips that provide access to the radio's I/O pins. Removing the jumper on JP-3 disconnects Pot R9 from ADC1, allowing an external signal to be applied to ADC1 on JP-2. Removing the jumper on JP-4 disconnects the thermistor voltage divider from ADC0, allowing an external signal to be applied to ADC0 on JP-2. Note that on some developer boards, ADC0 is labeled ADCX, and ADC1 is labeled ADCY.

Removing the jumper on JP-6 disconnects the radio's serial output (UART) from the developer board's RS232/USB circuitry, allowing a direct connection to this logic level signal on JP-2. Likewise, removing the jumper on JP-11 disconnects the radio's serial input (UART) from the developer board's RS232/USB circuitry, allowing a direct connection to this logic level signal on JP-2. GPIO4 and GPIO5 can be configured to provide CTS/RTS hardware flow control. If hardware flow control is not used, removing the jumpers on JP-12 and JP-13 disconnects them from the developer board's RS232/USB circuitry, allowing them to be used for other purposes by connecting to them on jumper strip JP-1.

The radio's current consumption can be monitored by removing the jumper on JP-14 and connecting a current meter to its pins. The developer board is normally powered through the power connector or with a 9 V battery, even when communicating through the USB port. The developer board can be optionally powered through the USB port by moving the jumper on JP-12 to the two pins toward the middle of the circuit board, and adding a jumper to J1.

## 5.5 LPRDemo 2.0 Utility Program

LPRDemo2.exe is a Windows-based utility that allows LPR2430 series modules to be configured and exercised. LPRDemo2 configures the modules using the commands detailed in Section 4 of this manual. It is important to note that LPRDemo2 uses protocol mode for all functions except sending and receiving data through the Wincom tab, where transparent mode is used. If the developer board is disconnected from the PC while LPRDemo2 is still running, the module on the developer board will still be in protocol mode. Closing LPRDemo2 before disconnecting the developer board will return the module to transparent mode.

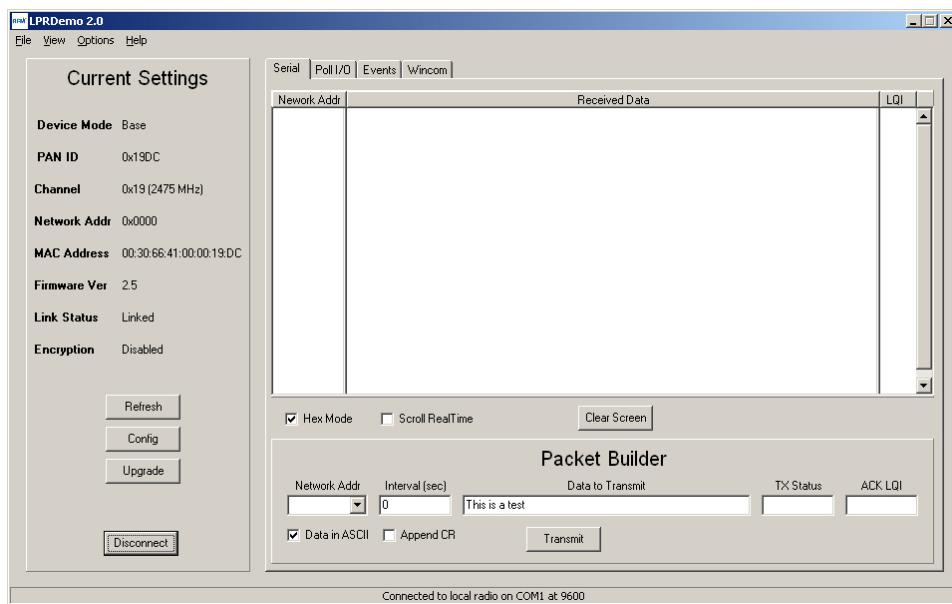


Figure 5.5.1

LPRDemo2 starts up displaying the *Serial* tab, as shown in Figure 5.5.1. Connect PCs to both the Base and the Remote for serial communication testing.

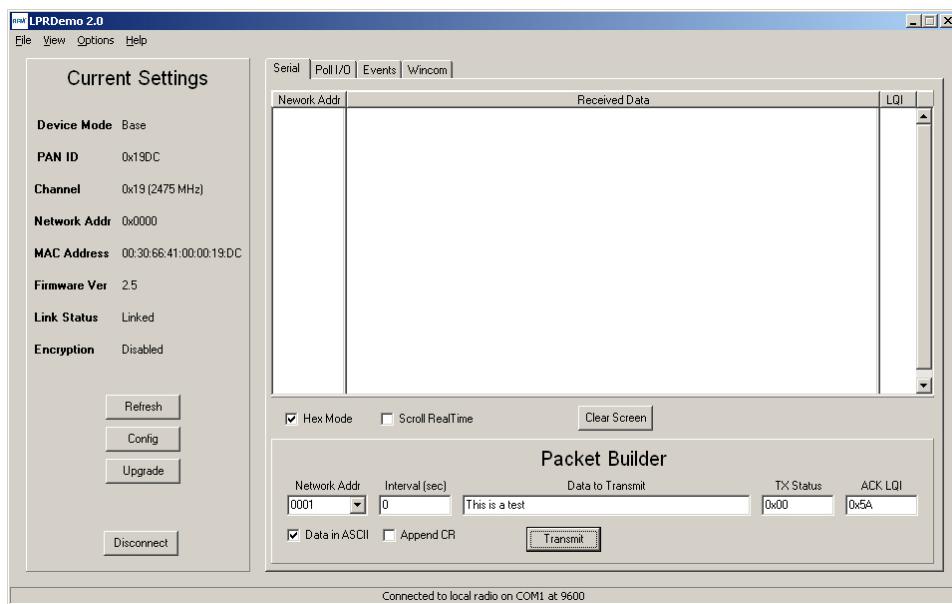


Figure 5.5.2

Pressing the *Transmit* button on this screen sends the message in the *Data to Transmit* text box to the selected *Network Address*. Data sent to the local radio is displayed in the *Received Data* text box.

Received data can be displayed as ASCII or in hexadecimal format (default) by checking the *Hex Mode* check box. When the *Transmit Interval* is set to zero, *Data to Transmit* is sent once when the *Transmit* button is clicked. When the *Transmit Interval* is set to a positive number, Pressing the *Transmit* button once will cause a transmission each transmit interval (seconds) until the button is pressed again.

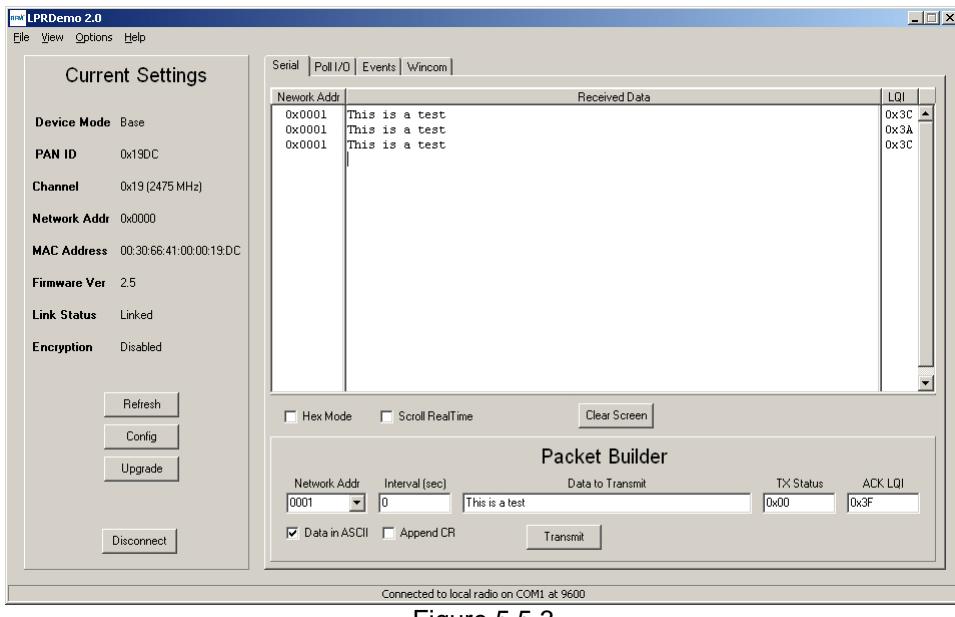


Figure 5.5.3

Figure 5.5.3 shows *Received Data* in ASCII mode. The *Network Address* column in the *Received Data* text box lists the node that sent the message.

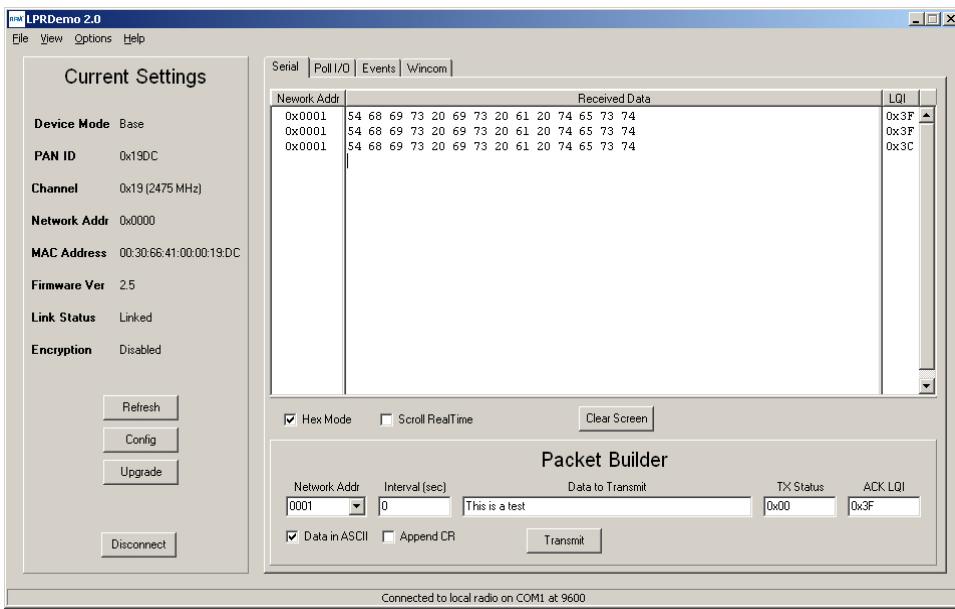


Figure 5.5.4

Figure 5.5.4 shows *Data Received* with *Hex Mode* checked.

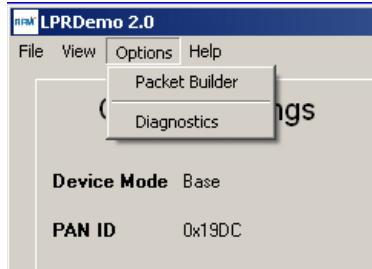


Figure 5.5.5

A *Packet Builder* dialog window can be called from the *Options* menu. The dialog window facilitates building protocol formatted messages, as discussed in Section 4.0. The four configurations of this dialog window are shown in Figures 5.5.6 through 5.5.9 below.

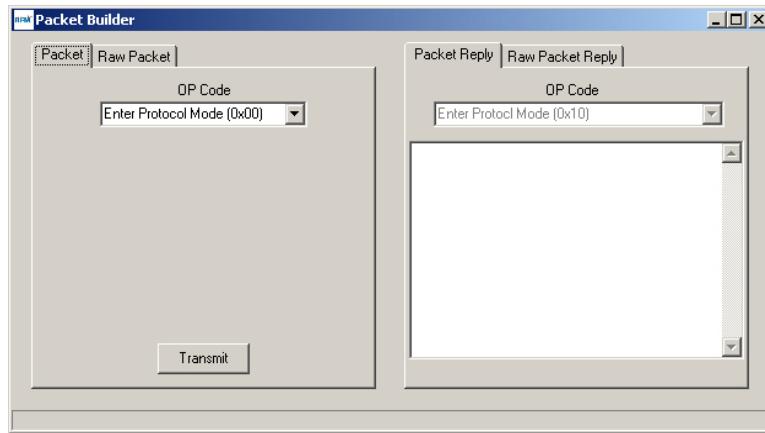


Figure 5.5.6

The *Packet* drop-down box on the left of Figure 5.5.6 allows the *OP Code* for any protocol command to be selected and the command transmitted.

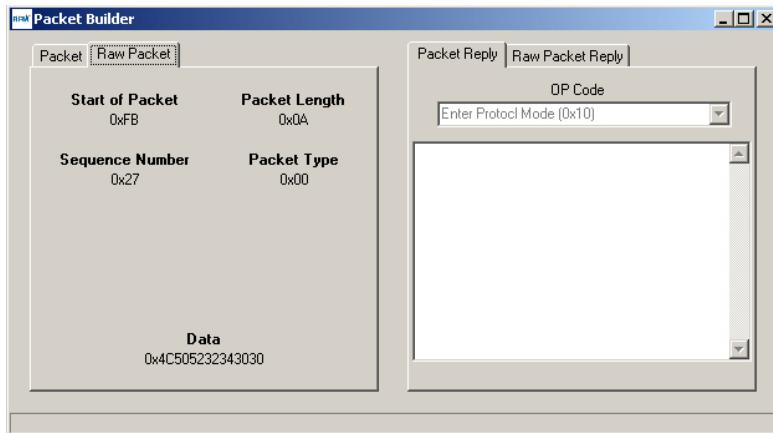


Figure 5.5.7

After selecting the *OP Code* as shown in Figure 5.5.6, switch to the *Raw Packet* tab to see the components of the protocol command.

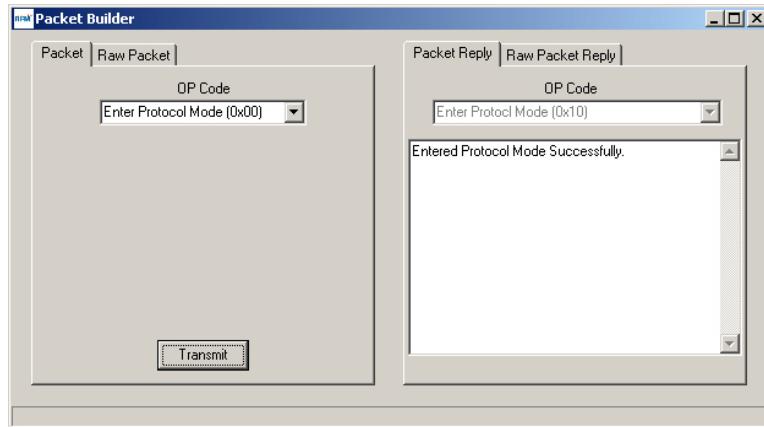


Figure 5.5.8

The *Packet Reply* tab displays the OP Code in the response to the transmitted command, and a description of the response message.

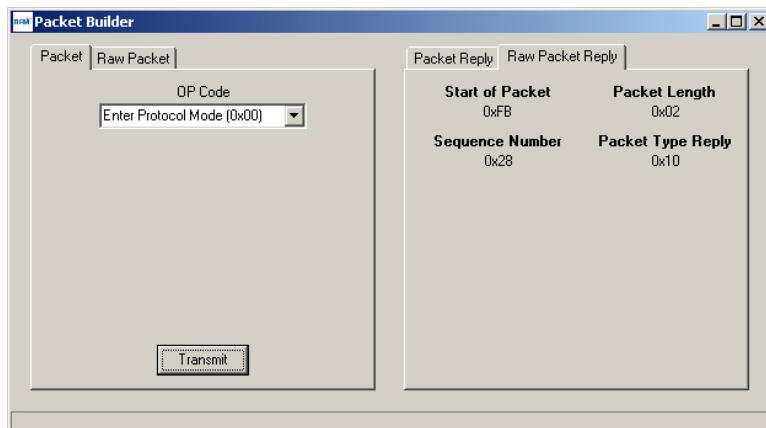


Figure 5.5.9

After receiving a *Packet Reply* as shown in Figure 5.5.8, switching to the *Raw Packet Reply* tab shows the key components of the protocol reply message. The information shown in these windows should be read from top left to right.

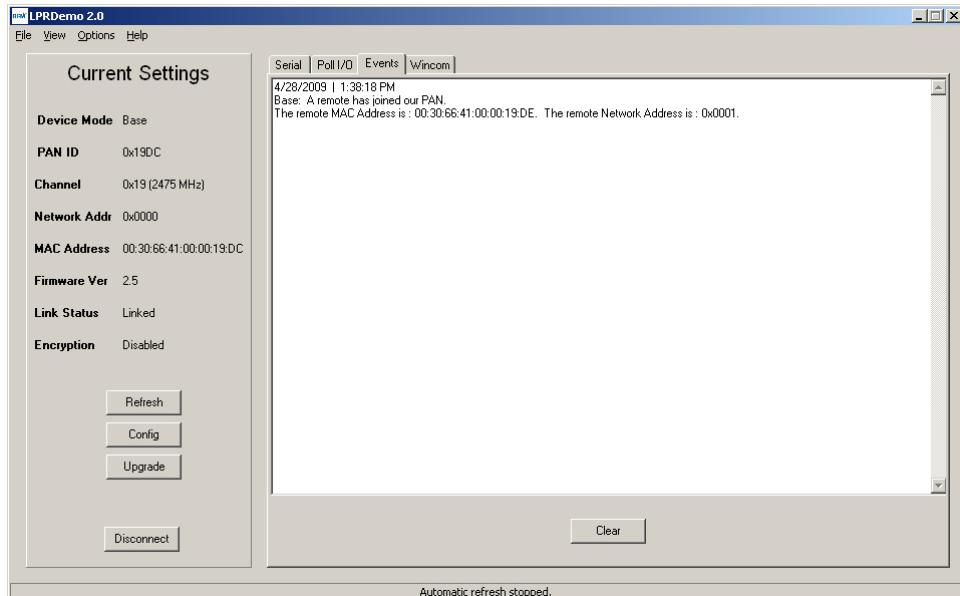


Figure 5.5.10

The *Events* Tab displays network events useful in understanding network operation. An event message related to a remote joining the PAN network is shown in Figure 5.5.10 above.

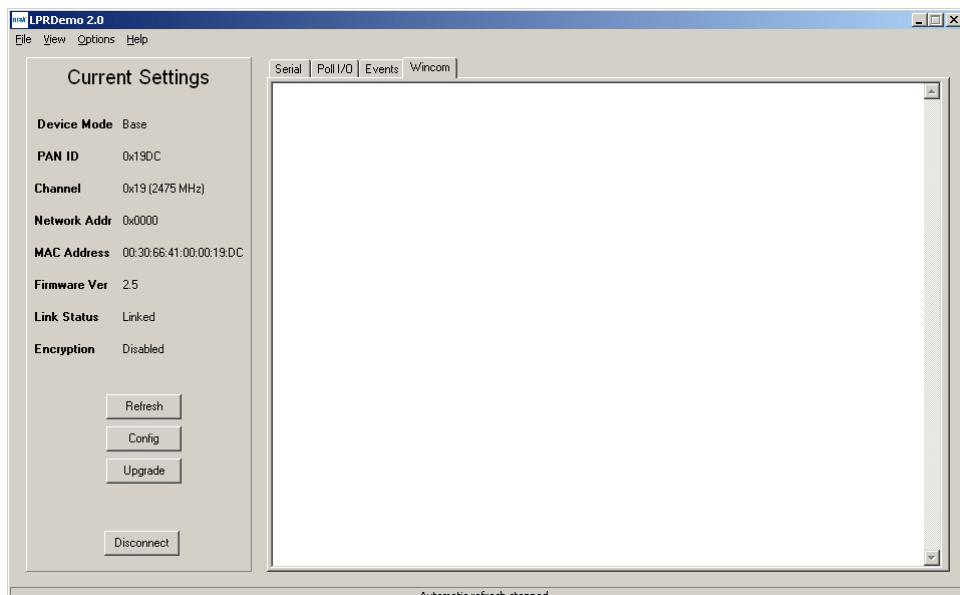


Figure 5.5.11

LPRDemo2 also has a built-in terminal program, *Wincom*, which is especially useful for transparent mode communications. Use the F8 key to switch the *Wincom* screen display from ASCII to hexadecimal. Note that the keyboard does not echo to the terminal program window.

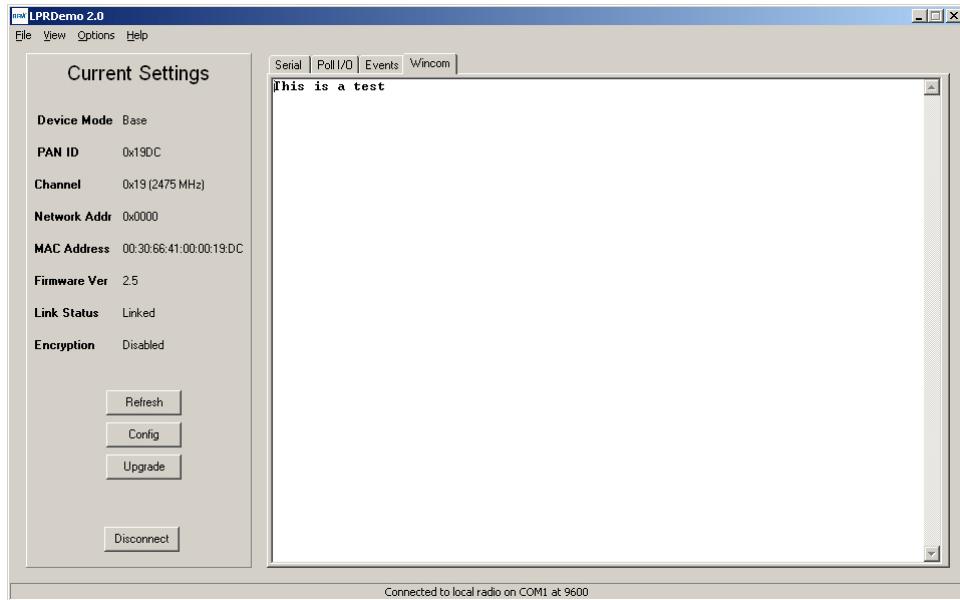


Figure 5.5.12

Figure 5.5.12 shows a text message received by *Wincom* in ASCII display mode.

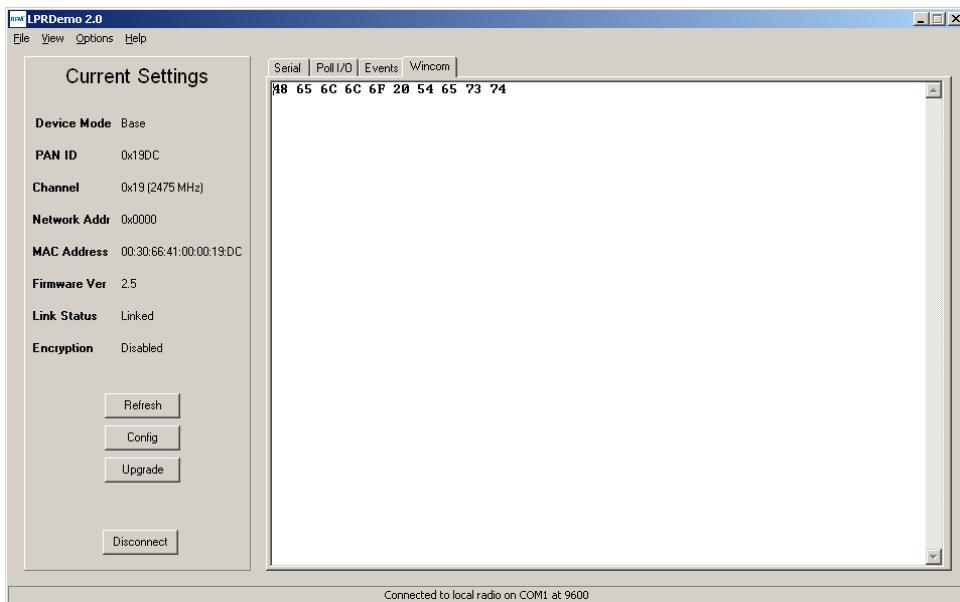


Figure 5.5.13

Figure 5.5.13 shows a text message received by *Wincom* in hexadecimal display mode.

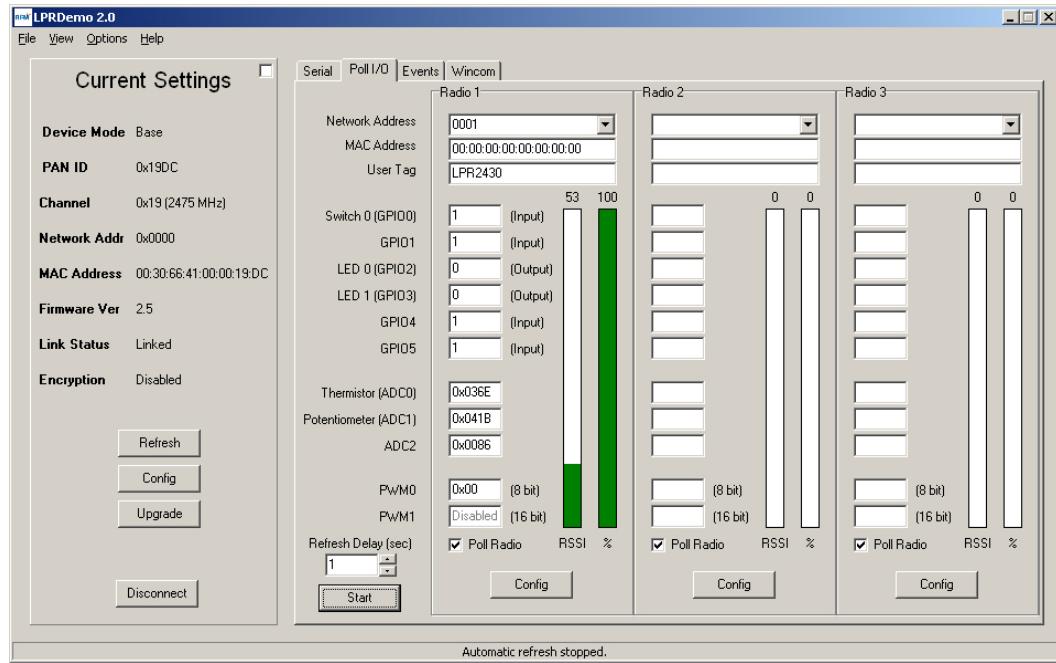


Figure 5.5.14

The *Poll I/O* tab provides an alternate display that shows data on four remotes instead of the Base and three remotes. Clicking on the check box just to the right of the *Current Settings* text in Figure 5.5.14 switches to the display in Figure 5.5.15. Clicking on the check box to the left of the *Serial* tab in Figure 5.5.15 switches back to the default display shown in Figure 5.5.14.

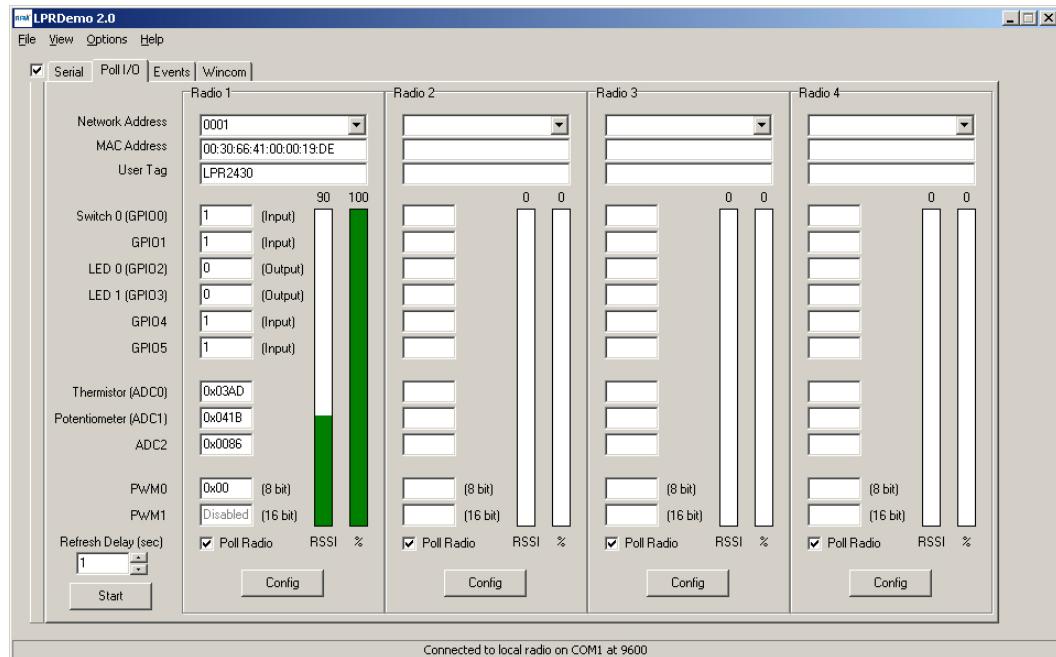


Figure 5.5.15

A multi-tab *Device Config* dialog window can be accessed for the Base and each Remote in the network by clicking the radio's *Config* button.

*Refresh*, *Apply* and *Cancel* buttons are located below the set of tabs on the *Device Config* dialog window. Clicking the *Refresh* button loads a copy the configuration parameters currently running in the radio into the tabs on the *Device Config* window. The *Apply* button writes all changes made on all the tabs into the radio's EEPROM. The *Cancel* button closes the *Device Config* window without writing any changes made on the tabs to the radio. However, once the *Apply* button has been clicked, *Cancel* will not undo those changes.

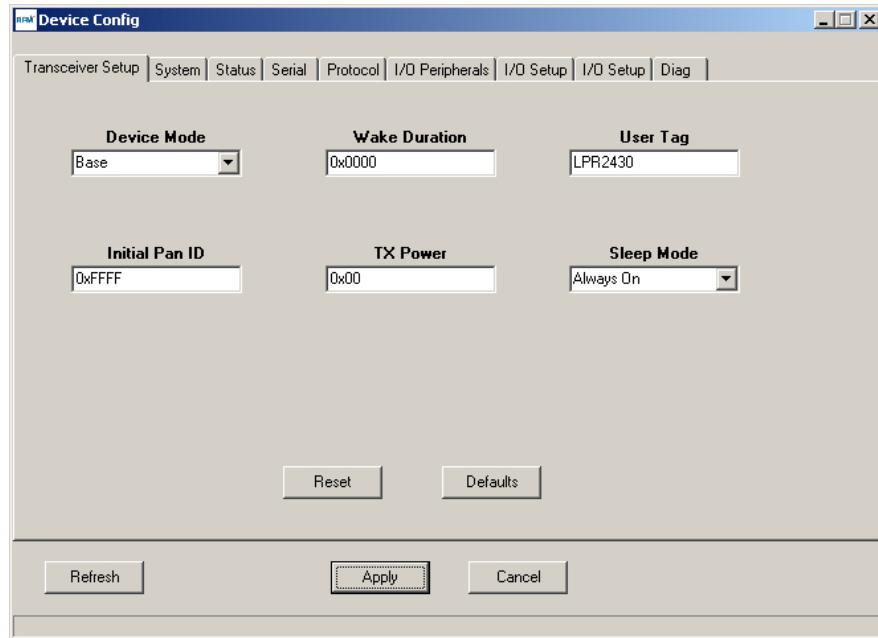


Figure 5.5.16

The radio's basic setup can be viewed and configured on the *Transceiver Setup* tab. The *Device Mode* drop-down box allows *Base* or *Remote* operating mode to be selected. The *Initial Pan ID* text box displays the current initial PAN ID and accepts changes. *Wake Duration* specifies how long a Remote stays awake after sending an I/O report in sleep mode. The *TX Power* text box displays the current transmitter power setting and accepts changes. The *User Tag* text box displays the current tag and accepts changes. The *Sleep Mode* drop-down box allows the sleep mode to be viewed and/or changed - *Always On*, *Timer Sleep* or *Deep Sleep*.

The radio must be reset in order for some changes, such as device mode, to take effect. Click on *Apply*, then on *Reset* to process these changes. Refer to Section 4 to determine which changes require resets. The *Defaults* button sets all configuration parameters back to their factory default values. Click on *Defaults*, *Apply* and then *Reset* to begin running on factory defaults.

The *System* tab allows basic system parameters to be viewed and set, as shown in Figure 5.5.17. The *Channel Set* box specifies the channels the system can use. The Base will establish its network on the first clear channel it finds and will remain there, even if the channel degrades, until it is reset. Specifying the channels a Base can use allows overlapping networks to be coordinated and known noisy channels to be avoided.

To enable encryption, set the 128-bit (16 byte) AES *Encryption Key* and check *Enable Encryption*. This must be done for all radios in the network. All radios must use the same *Encryption Key*. The key cannot be read back, so care must be used in setting the keys.

The Base uses the *Lease Period* to detect when a remote has left the network. *ARQ Attempt Limit* sets the number of transmission retries for an unacknowledged packet. *Broadcast Attempts* sets the number of times a broadcast packet transmission is repeated. The *Relay Scan Rate Normal* and *Relay Scan Rate Fast* parameters set the normal and fast scan intervals for a Remote to update its relay partner table.

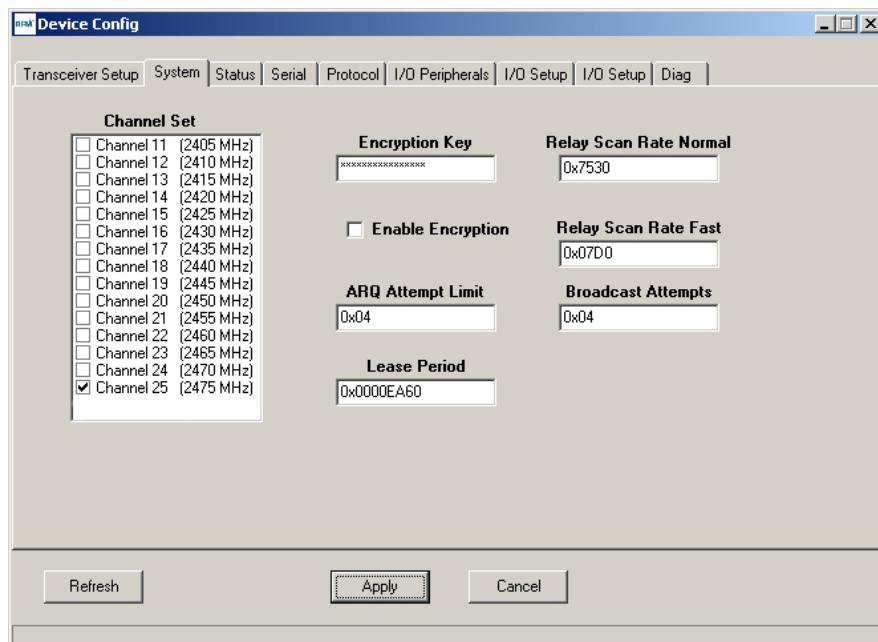


Figure 5.5.17

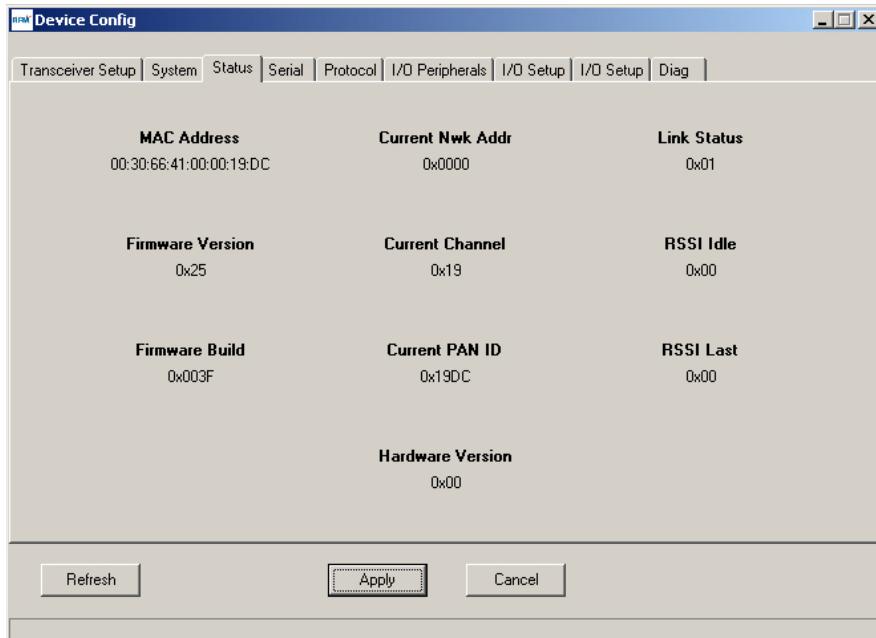


Figure 5.5.18

The *Status* tab provides read-only status information, as shown in Figure 5.5.18.

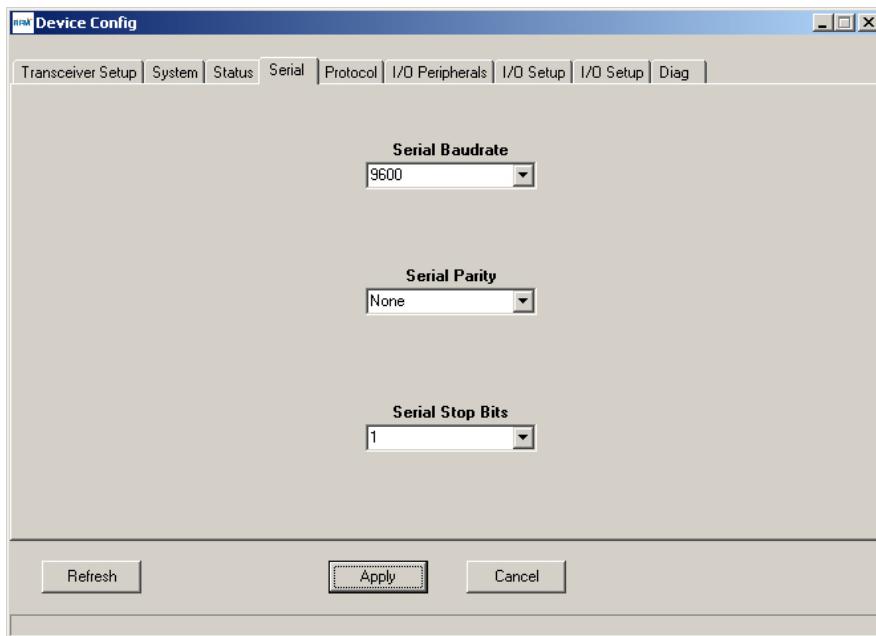


Figure 5.5.19

The *Serial* tab provides serial port configuration drop-down boxes that allow the baud rate, parity and number of stop bits to be set. Hardware flow control is set up on the right-most *I/O Setup* tab.

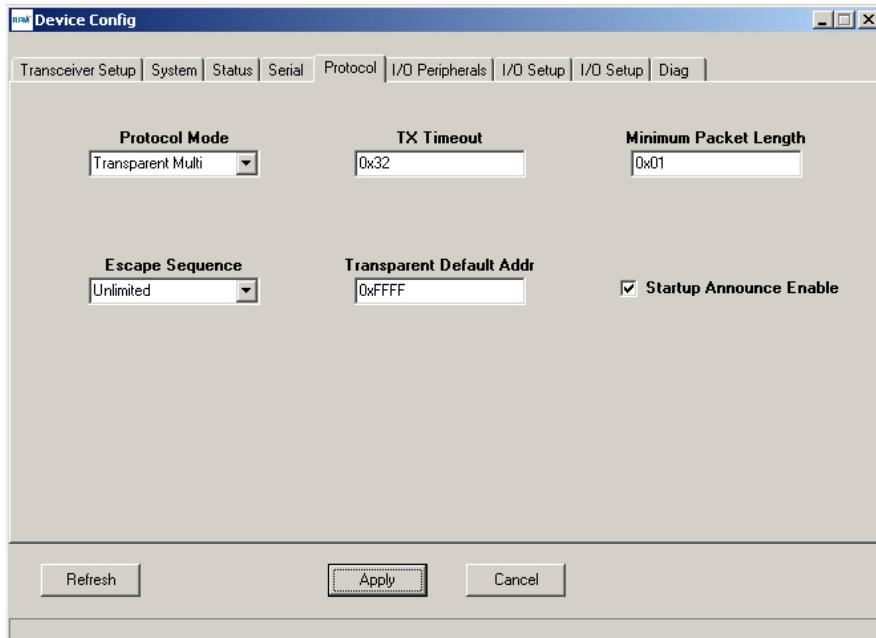


Figure 5.5.20

The *Protocol* tab is shown in Figure 5.5.20. The *Protocol Mode* drop-down box allows the selection of *Transparent Multi* (point-to-multipoint), *Transparent Pt-Pt* (point-to-point) and *Protocol* mode. The *Escape Sequence* drop-down box allows either *Once* (one chance at startup) or *Unlimited* (anytime) to be selected. The *TX Timeout* and *Minimum Packet Length* parameters control when transparent data is sent, as discussed in Section 4.2.5. *Transparent Default Addr* sets the destination address for transparent mode operation. The default for a base is the 0xFFFF network broadcast address. The default for a remote is 0x0000, the base network address. The *Startup Announce Enable* checkbox enables announce messages.

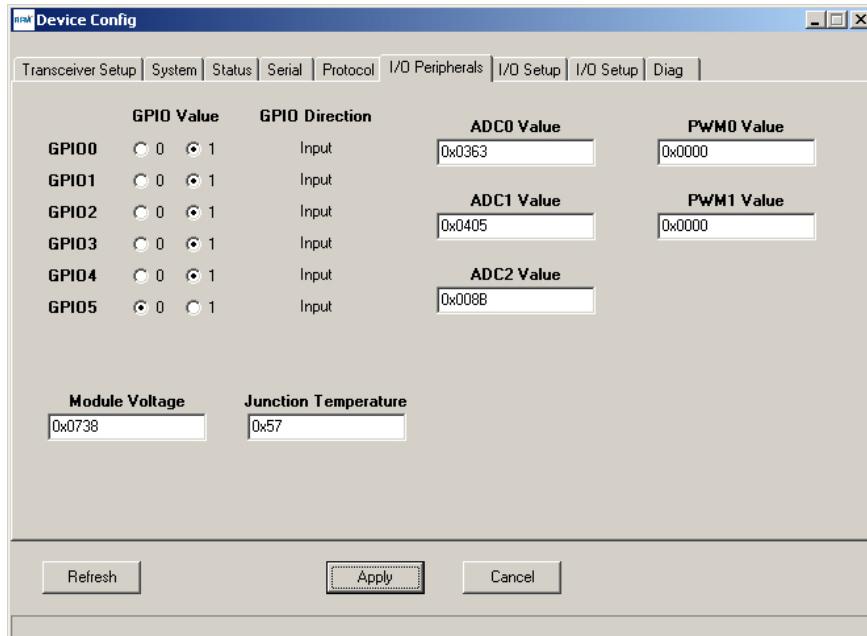


Figure 5.5.21

Figure 5.5.21 shows the *I/O Peripherals* tab contents. All GPIO ports are shown configured as inputs (the GPIO ports are configured as inputs or outputs on the left *I/O Setup* tab). Ports *GPIO0-4* have logic high inputs, *GPIO5* has a logic low. When a GPIO is defined as an output, its state can be set with its *GPIO Value* radio buttons. The 11-bit *Module Voltage* (module power supply voltage), *Junction Temperature* (module internal temperature) and *ADC0*, *ADC1* and *ADC2* readings are given in hexadecimal. The PWM output settings can be read and adjusted in the *PWM0 Value* (8-bit) and *PWM1 Value* (16-bit) text boxes.

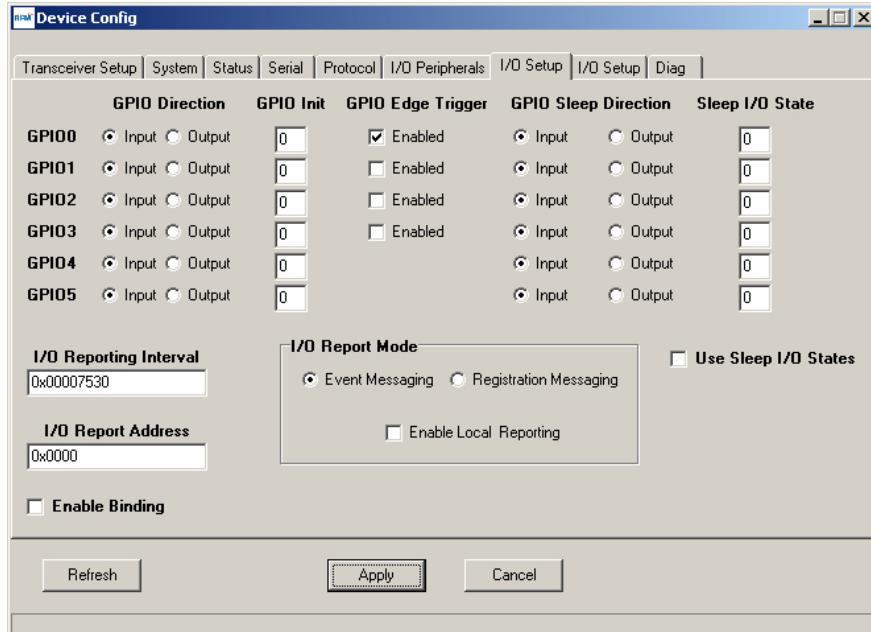


Figure 5.5.22

Figure 5.5.22 shows the first *I/O Setup* tab contents. This tab allows the direction of the GPIO ports to be set both for active and sleep modes, and in the case of GPIO outputs, the initial power up states can be set in the *GPIO Init* column and the sleep mode states can be set in the *Sleep I/O State* column (check *Use Sleep I/O States*). The *I/O Report Mode*, *I/O Reporting Interval* and *I/O Report Address* are also configured on this tab. The *Enable Binding* check box allows specific analog and digital input values from the *I/O Report* of the base or another remote (*I/O Report Address*) to be mapped as I/O outputs on this remote. See Table 4.2.7.3 for additional details.

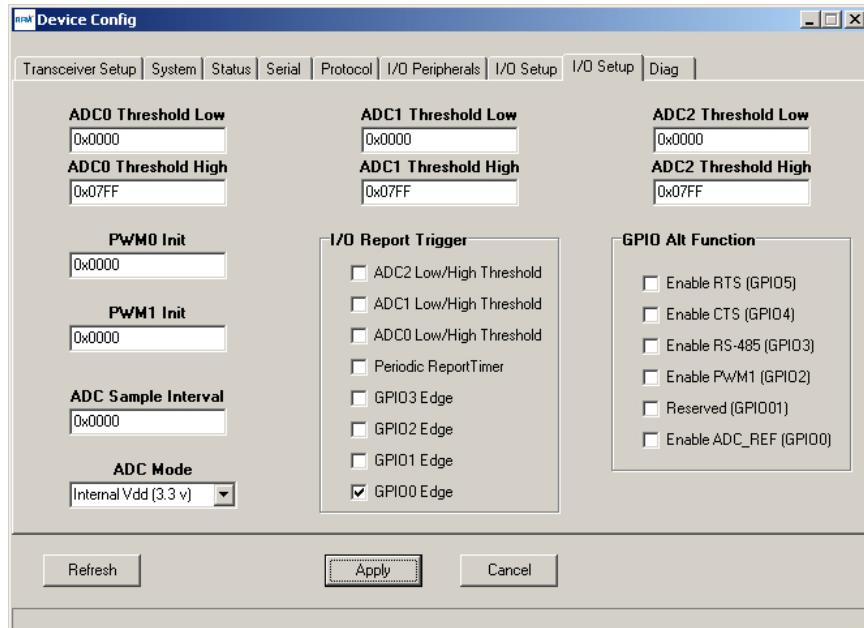


Figure 5.5.23

Figure 5.5.23 shows the second *I/O Setup* tab contents. The *ADC Sample Interval* and the *Threshold Low* and *High* limits for each ADC channel can be set on this tab, plus the event reporting triggers for each ADC channel. The *ADC Mode* drop-down box selects the ADC full scale reference. *PWM0 Init* and *PWM1 Init* set the start-up output values for each PWM channel.

When *GPIO00-3* are configured as inputs, *I/O Report Triggers* can be set for each of them with check boxes. *GPIO Alt Functions* can also be set under this tab.

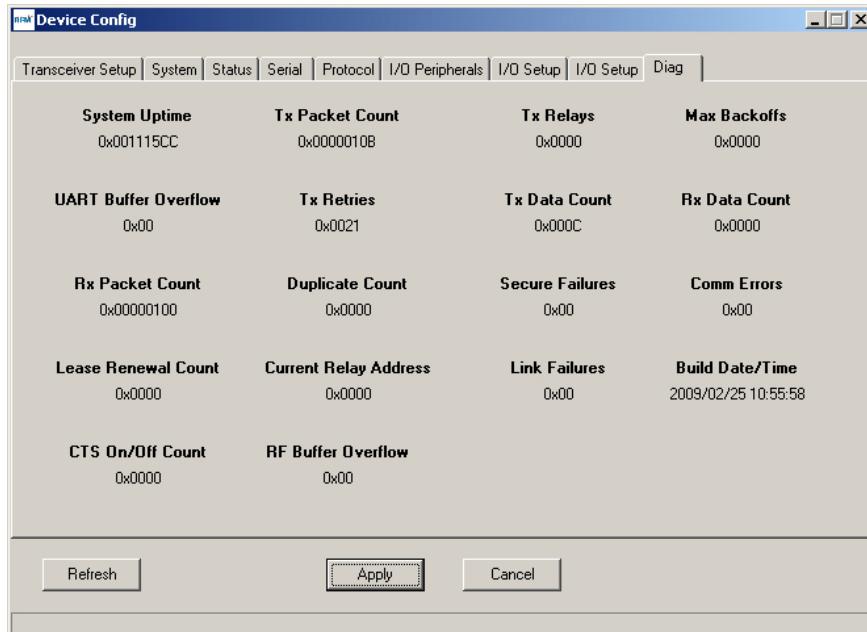


Figure 5.5.24

Figure 5.5.24 displays diagnostic information useful for network tuning and troubleshooting. The *Diag* tab is particularly helpful in coordinating serial data flow rates with RF packet delivery rates, and evaluating the robustness and stability of the RF link between this node and the rest of the network.

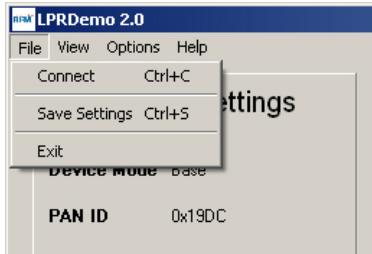


Figure 5.5.25

The *File* drop-down menu is shown in Figure 5.5.25. This menu provides an alternate access to the *Connect* dialog window, allows the configuration of LPRDemo2 to be saved (not the module settings), plus a conventional Windows program *Exit*.

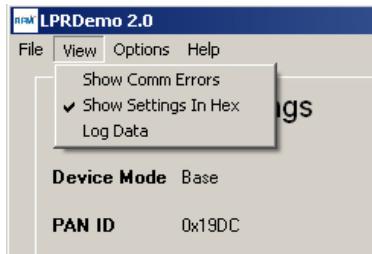


Figure 5.5.26

The *View* drop down menu is shown in Figure 5.5.26. Enabling *Show Comm Errors* causes a dialog window to appear if the serial connection experiences an overrun, underrun, or framing error. *Show Setting in Hex* performs the same function at the F8 key for the *WinCom* terminal program, allowing received data to be shown as either hexadecimal or ASCII data. *Log Data* creates a log file of the communication between LPRDemo2 and the radio module. The log file name is *logfile.dat*, and is created in the same directory as LPRDemo2. The file is ASCII text. A sample of a *logfile.dat* is shown below:

```

Sent Data : FB 08 A3 0B 01 00 02 05 01 00 (Set Remote Register)
Recv Data : FB 06 A3 1B 00 01 00 58 (Set Remote Register Reply)
Sent Data : FB 08 A4 0B 01 00 03 05 01 00 (Set Remote Register)
Recv Data : FB 06 A4 1B 00 01 00 58 (Set Remote Register Reply)
Sent Data : FB 07 A5 0A 01 00 00 00 2C (Get Remote Register)
Recv Data : FB 35 A5 1A 00 01 00 58 00 00 00 2C 00 FF FF 00 64 00 00 00 64 00 00 02 FF
FF FF FF FF FF FF FF FF FF FF FF FF 4C 50 52 32 34 33 30 00 00 00 00 00 00 00 00 00 00
00 00 00 (Get Remote Register Reply)

```

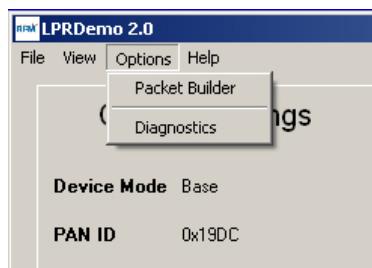


Figure 5.5.27

The *Options* drop-down menu includes access to *Packet Builder*, as discussed in the text associated with Figures 5.5.6 through 5.5.9 above. The *Diagnostics* menu item provides one of two dialog windows, depending on whether the radio is operating as a base or a remote.

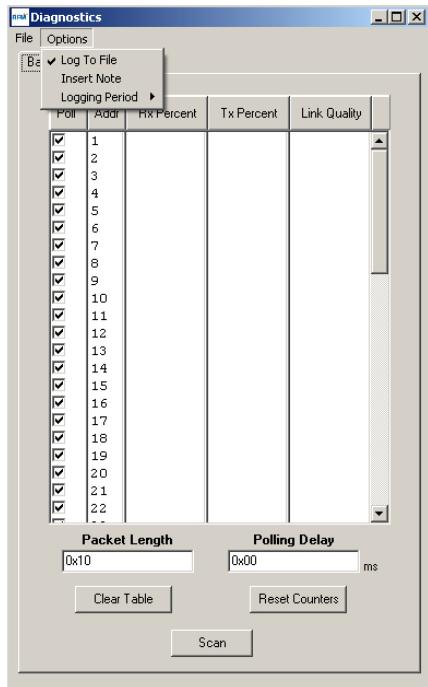


Figure 5.5.28

The diagnostic window for the Base is shown in Figure 5.5.28 above. Checking a network address in the *Poll* column adds it to the diagnostic data. A logging option is provided to capture the diagnostic data. The name of the file is *diagfile.dat*. It is created in the same directory as LPRDemo2.

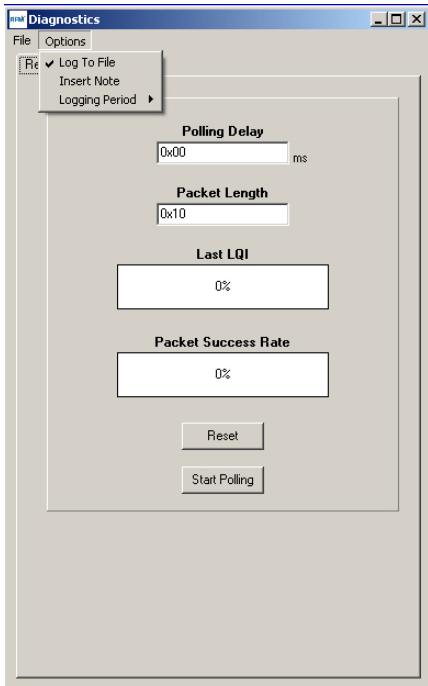


Figure 5.5.29

The diagnostic window for a Remote is shown in Figure 5.5.29 above. A logging option is also provided to capture diagnostic data. The name of the file is *diagfile.dat*. It is created in the same directory as LPR-Demo2.



Figure 5.5.30

The *Help* drop-down menu provides an *About* item that shows the version number, etc., of LPRDemo2.



Figure 5.5.31

Clicking the *Upgrade* button under the Base *Current Settings* data on the main LPRDemo2 window launches the *Upgrade Firmware* dialog base shown in Figure 5.5.31. Contact MURATA module technical support before attempting an update. See Section 7.2.

---

## 6.0 Troubleshooting

*LPR2430 series radio not responding* - make sure the /RESET line is not asserted (logic low). Logic high brings the radio out of reset mode.

*Cannot enter protocol mode* - make sure the host data rate is correct. LPR2430 series radios default to 9.6 kb/s. If using the *EnterProtocolMode* command, send the complete protocol format for this command.

*Range is extremely limited* - this is usually a sign of a poor antenna connection or the wrong antenna. Confirm that the 50 ohm stripline connecting the radio to the antenna or antenna connector complies with Section 3.4. Check that the antenna is firmly connected. If possible, remove any obstructions near the antenna.

---

## **7.0 Appendices**

### **7.1 Ordering Information**

LPR2430: 1 mW low current 802.15.4 module for solder reflow mounting

LPR2430A: 1 mW low current 802.15.4 module with integral antenna for solder reflow mounting

LPR2430ER: 63 mW high power 802.15.4 module for solder reflow mounting

LPR2430ERA: 63 mW high power 802.15.4 module with integral antenna for solder reflow mounting

### **7.2 Technical Support**

For LPR2430 series module technical support call MURATA at (678) 684-2000 between the hours of 8:30 AM and 5:30 PM Eastern Time

## 7.3 LPR2430 Series Mechanical Specifications

### LPR2430 Outline and Mounting Dimensions

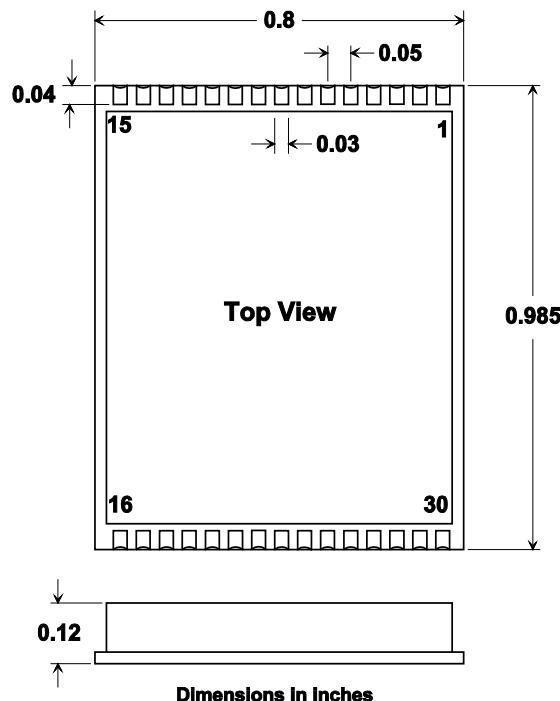


Figure 7.3.1

### LPR2430A Outline and Mounting Dimensions

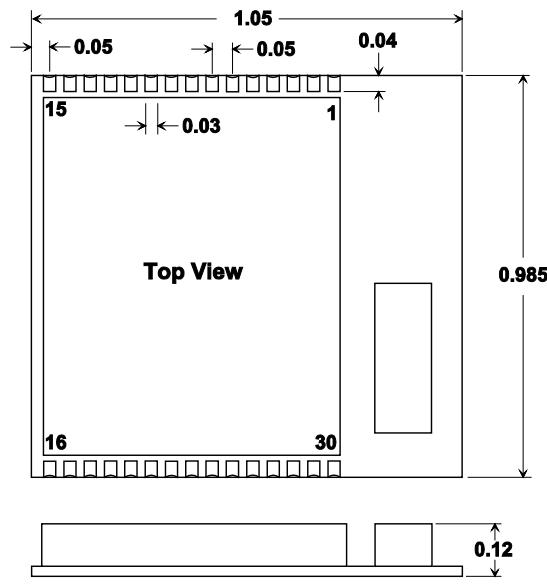


Figure 7.3.2

### LPR2430ER Outline and Mounting Dimensions

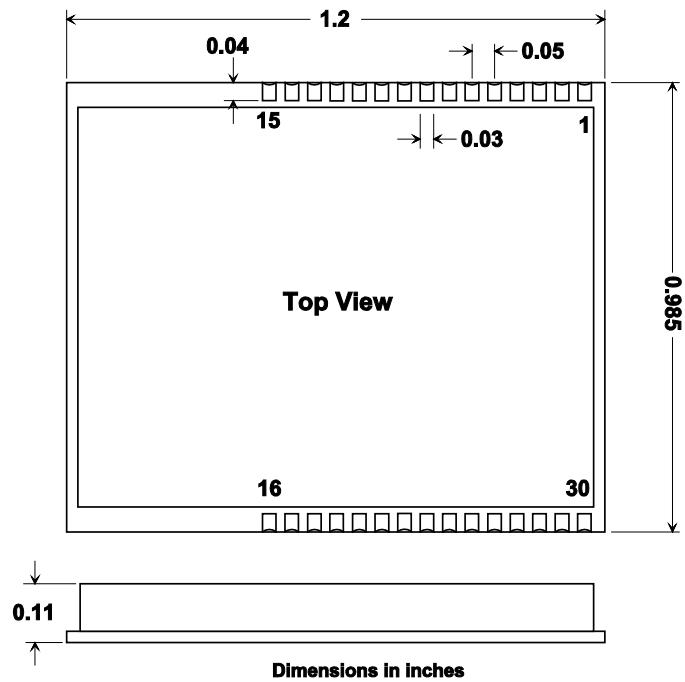


Figure 7.3.3

### LPR2430ERA Outline and Mounting Dimensions

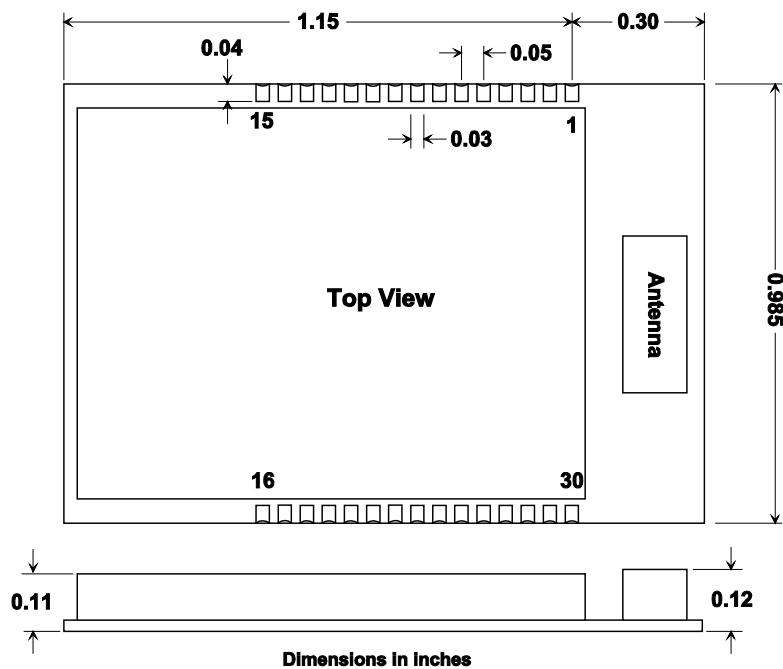
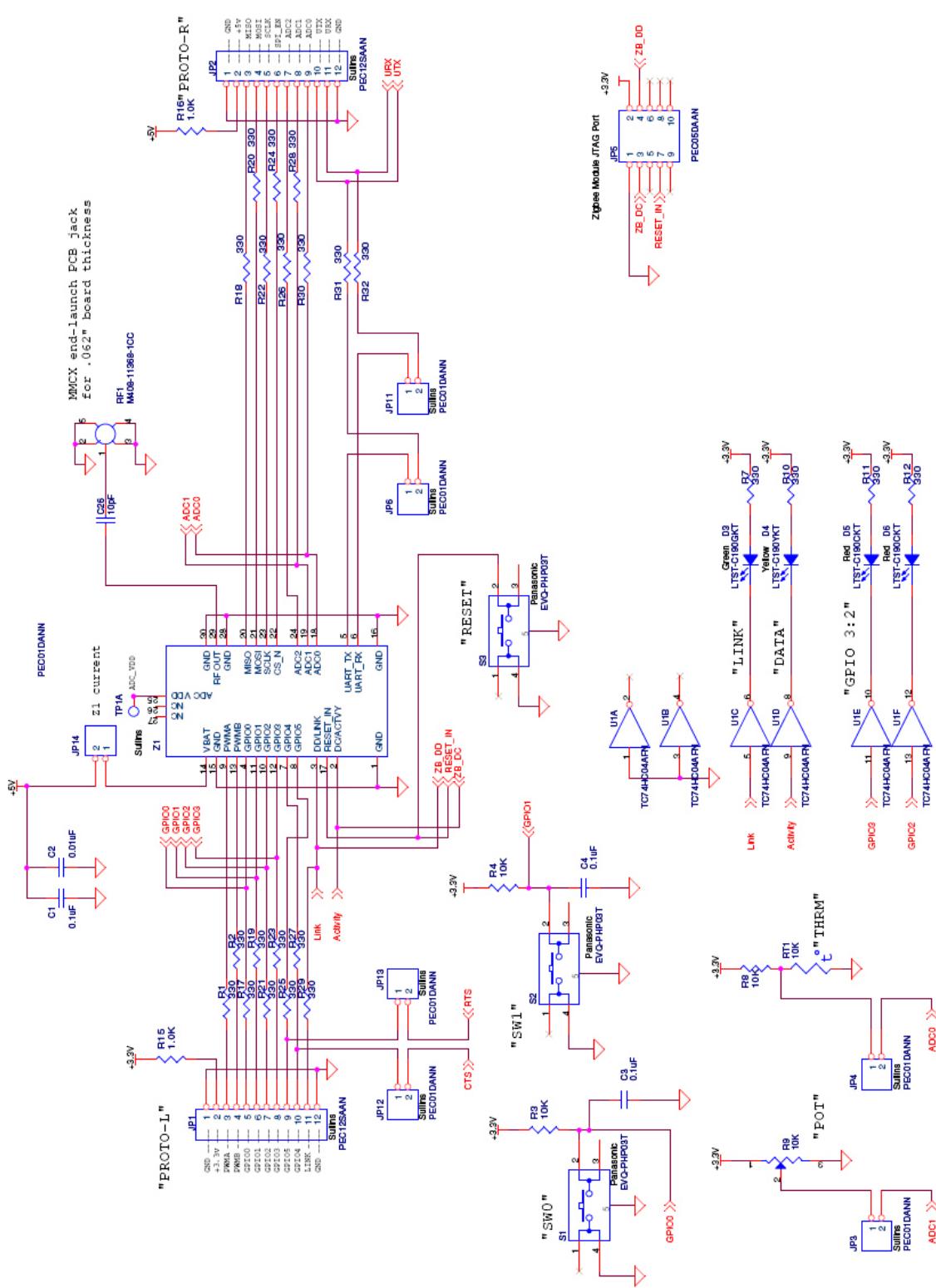
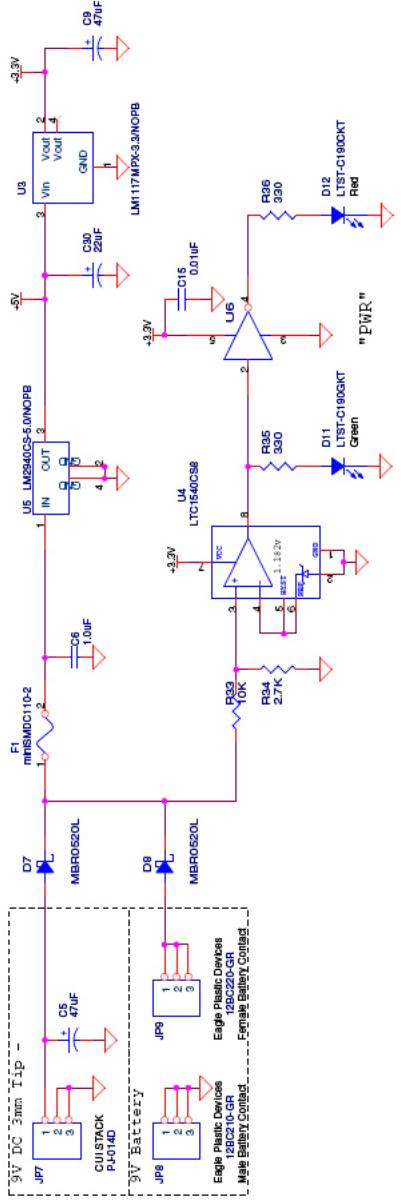
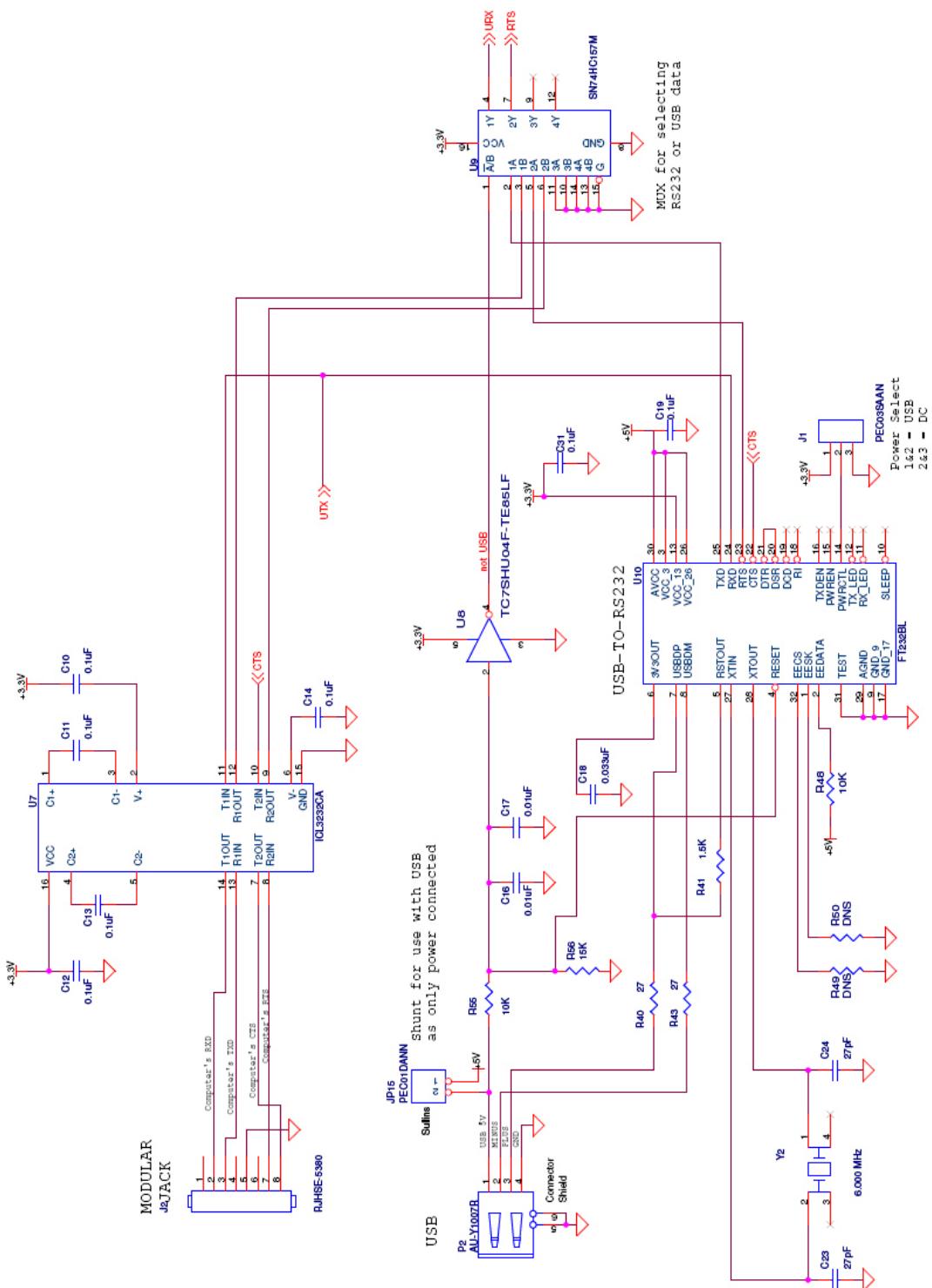


Figure 7.3.4

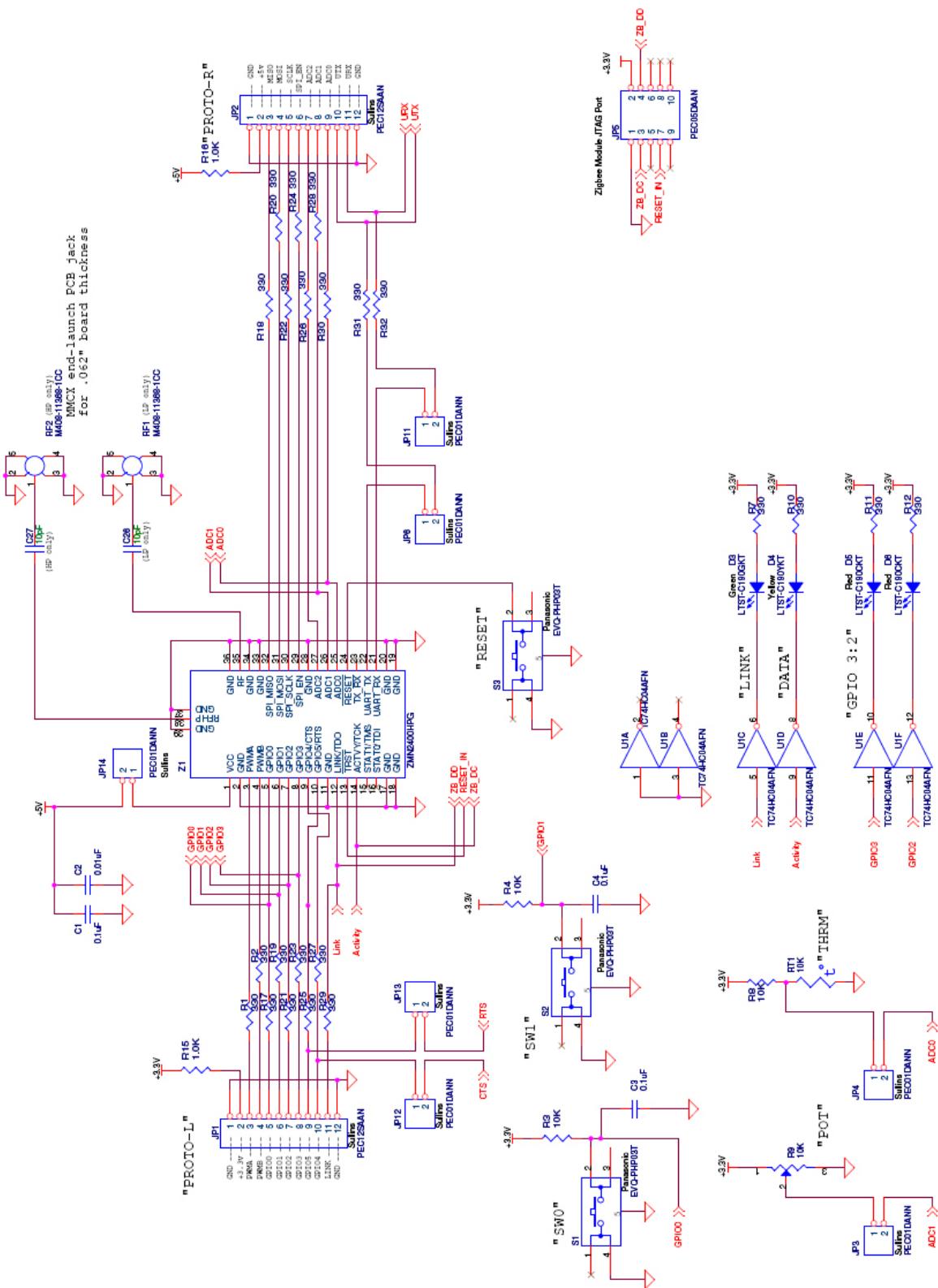
## 7.4 Developer Board Schematic, LPR2430A & LPR2430ERA Version

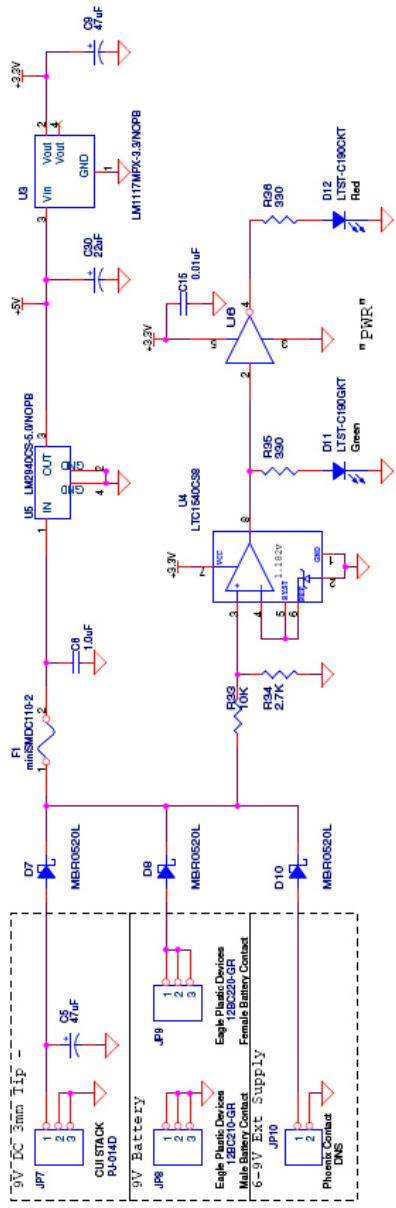


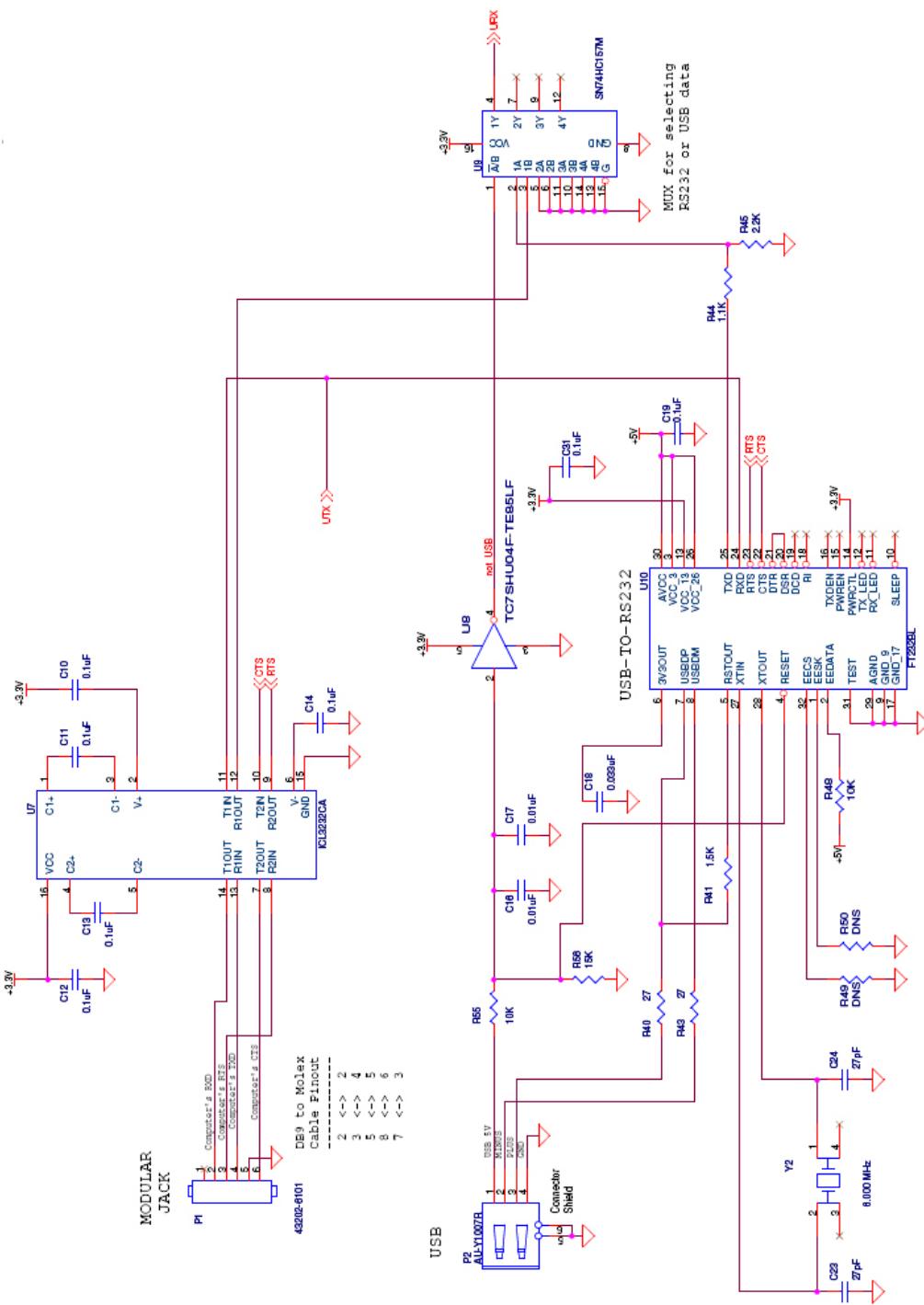




## 7.5 Developer Board Schematic, LPR2430 & LPR2430ER Version







---

## **8.0 Warranty**

Seller warrants solely to Buyer that the goods delivered hereunder shall be free from defects in materials and workmanship, when given normal, proper and intended usage, for twelve (12) months from the date of delivery to Buyer. Seller agrees to repair or replace at its option and without cost to Buyer all defective goods sold hereunder, provided that Buyer has given Seller written notice of such warranty claim within such warranty period. All goods returned to Seller for repair or replacement must be sent freight prepaid to Seller's plant, provided that Buyer first obtain from Seller a Return Goods Authorization before any such return. Seller shall have no obligation to make repairs or replacements which are required by normal wear and tear, or which result, in whole or in part, from catastrophe, fault or negligence of Buyer, or from improper or unauthorized use of the goods, or use of the goods in a manner for which they are not designed, or by causes external to the goods such as, but not limited to, power failure. No suit or action shall be brought against Seller more than twelve (12) months after the related cause of action has occurred. Buyer has not relied and shall not rely on any oral representation regarding the goods sold hereunder, and any oral representation shall not bind Seller and shall not be a part of any warranty.

**THE PROVISIONS OF THE FOREGOING WARRANTY ARE IN LIEU OF ANY OTHER WARRANTY, WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL (INCLUDING ANY WARRANTY OR MERCHANT ABILITY OR FITNESS FOR A PARTICULAR PURPOSE). SELLER'S LIABILITY ARISING OUT OF THE MANUFACTURE, SALE OR SUPPLYING OF THE GOODS OR THEIR USE OR DISPOSITION, WHETHER BASED UPON WARRANTY, CONTRACT, TORT OR OTHERWISE, SHALL NOT EXCEED THE ACTUAL PURCHASE PRICE PAID BY BUYER FOR THE GOODS. IN NO EVENT SHALL SELLER BE LIABLE TO BUYER OR ANY OTHER PERSON OR ENTITY FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS, LOSS OF DATA OR LOSS OF USE DAMAGES ARISING OUT OF THE MANUFACTURE, SALE OR SUPPLYING OF THE GOODS. THE FOREGOING WARRANTY EXTENDS TO BUYER ONLY AND SHALL NOT BE APPLICABLE TO ANY OTHER PERSON OR ENTITY INCLUDING, WITHOUT LIMITATION, CUSTOMERS OF BUYERS.**