Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Linx
TECHNOLOGIES

**TRM-915-R250
RF Transceiver Module**
Data Guide

Wireless made simple®

Warning: Linx radio frequency ("RF") products may be used to control machinery or devices remotely, including machinery or devices that can cause death, bodily injuries, and/or property damage if improperly or inadvertently triggered, particularly in industrial settings or other applications implicating life-safety concerns. No Linx Technologies product is intended for use in any application without redundancies where the safety of life or property is at risk.

The customers and users of devices and machinery controlled with RF products must understand and must use all appropriate safety procedures in connection with the devices, including without limitation, using appropriate safety procedures to prevent inadvertent triggering by the user of the device and using appropriate security codes to prevent triggering of the remote controlled machine or device by users of other remote controllers.

Do not use this or any Linx product to trigger an action directly from the data line or RSSI lines without a protocol or encoder/decoder to validate the data. Without validation, any signal from another unrelated transmitter in the environment received by the module could inadvertently trigger the action.

All RF products are susceptible to RF interference that can prevent communication. RF products without frequency agility or hopping implemented are more subject to interference. This module does have a frequency hopping protocol built in, but the developer should still be aware of the risk of interference.

Do not use any Linx product over the limits in this data guide. Excessive voltage or extended operation at the maximum voltage could cause product failure. Exceeding the reflow temperature profile could cause product failure which is not immediately evident.

Do not make any physical or electrical modifications to any Linx product. This will void the warranty and regulatory and UL certifications and may cause product failure which is not immediately evident.

Table of Contents

A large-print version of this document is available at www.linxtechnologies.com.

TRM-915-R250
RF Transceiver Module

Data Guide

Description

The 250 Series RF transceiver module is designed for reliable bi-directional transfer of serial data over distances of up to 4 miles (6.4km) line of sight. Operating in the 902 to 928MHz frequency band, the module is capable of generating +23.5dBm into a 50-ohm load and achieves an outstanding typical sensitivity of –105dBm. This high output power gives the module exceptional range and also helps overcome noisy environments at shorter ranges.
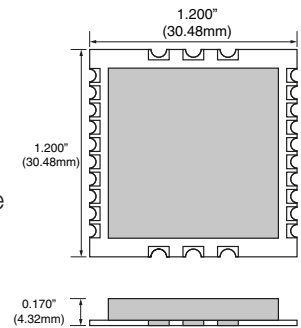
1.200" (30.48mm)
1.200" (30.48mm)
0.170" (4.32mm)
Figure 1: Package Dimensions

The module implements a Frequency Hopping Spread Spectrum (FHSS) protocol along with networking and assured delivery features. It has a Universal Asynchronous Receiver Transmitter (UART) serial interface that can be directly connected to microcontrollers, RS-232 converters or USB adaptors. The module automatically handles all radio functions resulting in a UART-to-antenna wireless link. All configuration settings and data are accessed through the UART interface.

Features

- True UART to antenna solution
- Frequency Hopping (FHSS)
- 153.6kbps max RF data rate
- Includes robust protocol (CSMA, assured delivery, addressing)
- Low Power Standby, Sleep and Deep Sleep modes
- Adjustable output power
- 32-bit unique address
- 5 volt tolerant I/O

Applications

- Direct RS-232/422/485 Wire replacement
- Asset tracking
- Automated meter reading
- Industrial/home automation
- Wireless sensors
- Remote data logging
- Fleet management
- Traffic and display signs
- Mass-transit communications
- Oil and gas sensing
- Long-range data links

Revised 3/18/2015

Ordering Information

Ordering Information		
Product Part No.	Description	Radiotronix Part No.
TRM-915-R250	Embedded Wireless Module, 250mW (900MHz)	Wi.232FHSS-250-R
TRM-915-R250-CFT	Embedded Wireless Module, 250mW (900MHz), Mexico	Wi.232-FHSS-250-CFTC-R
EVM-915-250-FCx	Pinned, Pre-Certified Module, 250mW (900MHz)	Wi.232FHSS-250-FCC-xx-R
EVM-915-250-CFx	Pinned, Pre-Certified Module, 250mW (900MHz), Mexico	Wi.232FHSS-250-FCC-CFTC-xx-R

x = 'R' for right angle connector, 'S' for straight connector
Transceivers are supplied in trays of 50 pieces

Figure 2: Ordering Information

Absolute Maximum Ratings

Absolute Maximum Ratings				
Supply Voltage V_{cc}	0	to	4.2	VDC
Any Input or Output Pin	0	to	5.0	VDC
Max Supply Voltage Rise Time (GND to 2.7V)			1	ms
Max RF Input		12		dBm
Operating Temperature	−40	to	+85	℃
Storage Temperature	−40	to	+85	℃

Exceeding any of the limits of this section may lead to permanent damage to the device. Furthermore, extended operation at these maximum ratings may reduce the life of this device.

Figure 3: Absolute Maximum Ratings

Warning: This product incorporates numerous static-sensitive components. Always wear an ESD wrist strap and observe proper ESD handling procedures when working with this device. Failure to observe this precaution may result in module damage or failure.

Electrical Specifications

250 Series Transceiver Specifications						
Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Power Supply						
Operating Voltage	V_{CC}	2.7	3.3	3.6	VDC	
TX Supply Current	I_{CCTX}					1,2
@+8dBm			54		mA	
@+13dBm			71		mA	
@+18dBm			109		mA	
@+23.5dBm			190		mA	
RX Supply Current	I_{CCRX}		25		mA	1
Standby Current	I_{STD}		1.5		mA	1
Sleep Current	I_{SLP}		1.5		mA	1
Deep Sleep Current	I_{DSLP}		3		µA	1
RF Section						
Operating Frequency Band	F_C	902.2		927.8	MHz	
Center Frequency Accuracy			2	4	PPM	3
Number of Channels			32		Chan.	
Channel Spacing			750		kHz	
Hop Sequences			6			4
Max Data Rate		115.2			kbps	
Antenna Port						
RF Impedance	R_{IN}		50		Ω	
Environmental						
Operating Temp. Range		-40		+85	℃	
Receiver Section						
Receiver Sensitivity						5
9.6kbps			−105		dBm	
38.4kbps			−102		dBm	
153.6kbps			−100		dBm	
Input IP3			−24		dBm	6
Adjacent Channel Rejection			60		dBc	7
IF Bandwidth			200		kHz	
Transmitter Section						
Max Output Power	P_O		23.5	24	dBm	2
Harmonic Emissions	P_H		−50		dBc	2
Frequency Deviation	F_{DEV}		50		kHz	

250 Series Transceiver Specifications

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Interface Section						
Input Logic Low	V_{IL}	0		0.8	VDC	
Input Logic High	V_{IH}	2.0		V_{CC}	VDC	
Output Logic Low	V_{OL}					
I_{OL} = 8.5mA				0.6	VDC	
I_{OL} = 10µA				0.1	VDC	
I_{OL} = 25mA			1.0		VDC	
Output Logic High	V_{OH}					
I_{OH} = -3mA		V_{CC}–0.7			VDC	
I_{OH} = -10µA		V_{CC}–0.1				
I_{OH} = -10mA			V_{CC}-0.8			
Flash Specifications (Non-Volatile Registers)						
Flash Write Duration			16		ms	
Flash Write Cycles		20k	100k		cycles	

1. V_{CC} = 3.3V
2. Into a 50-ohm load
3. At 25ºC
4. 26 channels each
5. At 10^{-3} BER
6. P_{in} = -20dBm, 2 CW interferers, F_{RF} = 915MHz, $F_1 = F_{RF}$ + 3MHz, $F_2 = F_{RF}$ + 6MHz, max gain, high-sensitivity
7. Desired signal 3dB above input sensitivity level, CW interferer power level increased until BER = 10^{-2}, +/–1MHz
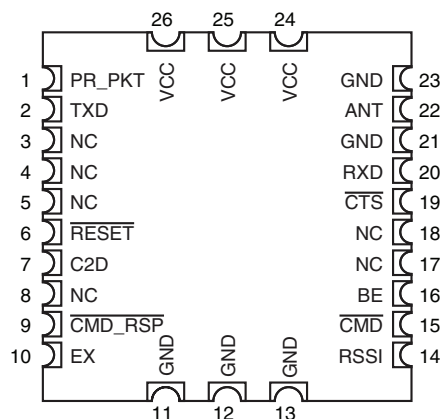
Figure 4: Electrical Specifications

Pin Assignments

Figure 5: 250 Series Transceiver Pin Assignments (Top View)

Pin Descriptions

Pin Number	Name	I/O	Description
1	PR_PKT	O	Processing Packet Indicator. This line goes high when the module is processing a valid or potentially valid packet.
2	TXD	O	UART Transmit Data Output. This is the output line for the configuration command responses as well as the data received over the air.
3, 4, 5, 8, 17, 18	NC	—	No Electrical Connection. Do not connect any traces to these lines.
6	$\overline{\text{RESET}}$	I/O	Reset line. This line is normally an input that acts as an active low hardware reset line. It does occasionally act as an output, so please see the Reset section for details.
7	C2D	—	Reserved
9	$\overline{\text{CMD\_RSP}}$	O	Command Response. This line is low when the data on the TXD line is a response to a command and not data received over the air.
10	EX	O	Exception Output. A mask can be set to take this line high when an exception occurs. The line is lowered when the exception register is read (regEXCEPTION)
11, 12, 13, 23	GND	—	Ground
14	RSSI	O	This line outputs an analog voltage that is proportional to the strength of the incoming signal.
15	$\overline{\text{CMD}}$	I	Command Input. This line sets the serial data as either command data to configure the module or packet data to be sent over the air. Pull low for command data; pull high for packet data.
16	BE	O	Buffer Empty. This line goes high when the UART input buffer is empty, indicating that all data has been transmitted.
19	$\overline{\text{CTS}}$	O	UART Clear To Send, active low. This line indicates to the host microcontroller when the module is ready to accept data. When CTS is high, the module is busy. When CTS is low, the module is ready for data.
20	RXD	I	UART Receive Data Input. This is the input line for the configuration commands as well as data to be sent over the air.
22	ANT	—	50-ohm RF Antenna Port
24, 25, 26	VCC	—	Supply Voltage

Figure 6: 250 Series Transceiver Pin Descriptions

Theory of Operation

The 250 Series transceiver is a low-cost, high-performance synthesized FSK transceiver. Its wideband operation gives it outstanding range while still meeting regulatory requirements. Figure 7 shows a block diagram for the module.
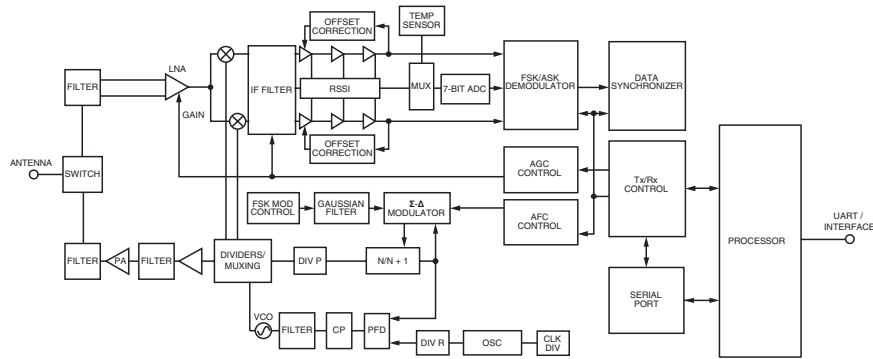

Figure 7: 250 Series Transceiver Block Diagram

The 250 Series transceiver is designed for operation in the 902 to 928MHz frequency band. The RF synthesizer contains a VCO and a low-noise fractional-N PLL. The receive and transmit synthesizers are integrated, enabling them to be automatically configured to achieve optimum phase noise, modulation quality and settling time.

The transmitter output power is programmable from +8dBm to +23.5dBm with automatic PA ramping to meet transient spurious specifications. The ramping and frequency deviation are optimized to deliver the highest performance over a wide range of data rates.

The receiver incorporates highly efficient low-noise amplifiers that provide up to –105dBm sensitivity.

An onboard controller performs the radio control and management functions. A processor performs the higher level protocol functions and controls the serial and hardware interfaces.

Module Description

The 250 Series RF transceiver module has a Universal Asynchronous Receiver Transmitter (UART) serial interface and is designed to create a complete UART-to-antenna wireless solution capable of direct wire replacement in most embedded RS-232/422/485 applications.

Note: Although the module is capable of supporting the serial data communications required by RS-232, RS-422, and RS-485 networks, it is not compatible with the electrical interfaces for these types of networks. The module has CMOS inputs and outputs and requires an appropriate converter for the particular type of network being used.

The module is designed to interface directly to a host UART. Three lines are used to transfer data between the module and the host UART: TXD, RXD, and $\overline{\text{CTS}}$. TXD is the data output from the module. RXD is the data input to the module. The $\overline{\text{CTS}}$ output indicates if the module is ready to accept data. The UART interface is capable of operating in full duplex at baud rates from 2.4 to 115.2kbps.

The module has a built-in protocol that automatically transmits the data input on the UART. All encoding, transmitting, receiving and decoding functions are handled by the internal processor, so no overhead is required by an external processor. The networking modes in the protocol allow for point-to-point and broadcast transmissions as well as allowing for the creation of subnets and more complicated network topologies.

The module can be put into a Sleep mode through serial commands. In Sleep mode, the RF section is completely shut down and the protocol processor is in an idle state. Once the module has been placed in the sleep mode, it can be awakened by sending a power-up sequence through the serial port.

If the current draw in sleep mode is too high for a particular application, power to the module can be switched through an external FET to turn off the module when it is not needed. If this technique is used, the volatile registers are reset to the values in their non-volatile mirrors, so any changes from the default will have to be reloaded.

Every module has a 32-bit GUID address that can be used by the host application to uniquely identify each module. This address can be read through the serial interface.

Module Operation

The module employs a Frequency Hopping Spread Spectrum (FHSS) algorithm. It has 32 channels spaced on 750kHz boundaries with a guard band on either side. These channels are pseudo-randomly arranged into six unique hopping tables comprised of 26 channels. The order of these tables is chosen so that cross-correlation is minimized, allowing multiple networks to operate in proximity with minimal interference.

When the module is not actively transmitting or receiving packets, it is in a scan state. It cycles through the channels in the hop sequence looking for a synchronizing packet. If it detects a preamble, it pauses to wait for the start code and packet header. If the packet is addressed to it, the module processes the packet and outputs the payload on the UART. If the packet is not addressed to the module or the start code and header fail their checks, the module resumes scanning for another packet.

When data is input on the RXD line for transmission, the module fills a buffer. Once the UART has buffered enough data to send (either regUARTMTU bytes input or regTXTO has expired between bytes on the RXD line), it transmits the data. The protocol engine makes a best-effort attempt to keep the data in at least regUARTMTU-sized packets, but splits the data based on the remaining dwell time before hopping. New data is not sent within the last 5% of the hop sequence, but data which is already in the process of being sent is processed normally.

The module prefixes the data with a packet header and postfixes the data with a 16-bit CRC. The 16-bit CRC error checking can be disabled to allow the host application to do its own error checking.

Initially, the transmission of the packet begins on a random hop index within the current hop sequence, and follows the hop sequence thereafter until synchronization is lost. The module uses a Carrier-Sense-Multiple-Access (CSMA) protocol to determine if another module is already transmitting on the selected channel. If the channel is occupied then the module waits for it to clear before transmitting its data.

Once the module gains access to the channel, if it is not already synchronized, it assigns itself master status, and sends a synchronizing preamble. Following a hop, the module that sends the first transmission assigns itself master status, sends a synchronizing preamble, and communications resume.

If acknowledgements are enabled for assured delivery, then once the packet is sent the module looks for an ACK from the other side. If the ACK is not received, a retry is performed and the transmission is sent again. If the number of transmission retries exceeds the value in the regMAXTXRETRY register, an exception (EX_NORFACK) is raised.

Once the packet is sent, the transmitter deactivates but remains tuned to the current channel until its hop time expires. If another packet is queued for transmission, the module transmits this packet once the CSMA mechanism allows access to the channel. Once the hop timer expires, the module hops to the next channel.

Synchronization is lost whenever there is no more data to transfer and the module has detected two consecutive hop indices without data present. The module then returns to scan mode.

If another unit is transmitting when the module is ready to transmit a packet, the module receives that data before attempting to transmit its data. If the UART receive buffer gets full, the $\overline{CTS}$ line goes high to prevent the host UART from over-running the receive buffer.

The CSMA mechanism introduces a variable delay to the transmission if it detects that the channel is occupied. This delay is the sum of a random period and a weighted period that is dependent on the number of times that the module has tried and failed to access the channel. For applications that guarantee that only one module is transmitting at any given time, the CSMA mechanism can be turned off to avoid this delay.

Low-Power States

The module supports three power saving modes: Standby, Sleep and Deep Sleep. Standby and Sleep are included primarily for legacy compatibility with DTS and EUR Series modules. The hardware required to support these two low-power modes fully is not present in the 25 Series modules. As a result, the current consumption in these two modes is considerably higher than their DTS / EUR counterparts. It is recommended that applications utilize the Deep Sleep mode for power savings.

In the Sleep and Deep Sleep modes, the transceiver is powered down and does not synchronize with other modules. Sleep mode draws more current than Deep Sleep mode. In Deep Sleep mode the module draws the least current. To wake the module up from this mode the $\overline{RESET}$ line must be held low for at least 20µs and then taken high. The module does not monitor the receive channel in either mode. Therefore, a sleeping module cannot be woken through the RF interface.

If regACKONWAKE is enabled, the module transmits a 0x06 character on the TXD line once awakened from a low-power mode or power-off state. This indicates that the module is ready to resume operations.

Figure 8 indicates the line states while in a low-power mode.

250 Series Transceiver Low-Power Line States		
Line Name	Pin Number	Pin State
PR_PKT	1	Driven low
TXD	2	Input with weak pull-up
$\overline{RESET}$	6	Input with weak pull-up
C2D	7	Input with weak pull-up
$\overline{CMD\_RSP}$	9	Input with weak pull-up
EX	10	Driven low
$\overline{RSSI}$	14	Driven low
$\overline{CMD}$	15	Input with weak pull-up
BE	16	Input with weak pull-up
$\overline{CTS}$	19	In Standby, Sleep: Driven Low, In Deep Sleep: Driven High
RXD	20	Input with weak pull-up

Figure 8: 250 Series Transceiver Low-Power Line States

Standby

Standby is selected by writing a 0x02 to *regOPMODE*. In this mode, the internal oscillator of the module's protocol controller is lowered to its slowest setting. The transmitter and receiver hardware is in power-down, but the radio's oscillator is enabled and running. The module wakes from standby in less than 6ms. A low pulse on the RXD line wakes the module. This pulse should be at least 1 bit-time in duration, so sending any byte to the UART wakes it with the low start bit. Because the module's oscillator is not capable at running at ultra-low speeds, use of this mode is not recommended for new applications. The RAM contents are preserved during standby. If the RAM fails an integrity check, the module issues itself a software reset to force re-initialization.

Sleep

Sleep is selected by writing a 0x01 to *regOPMODE*. The internal oscillator of the module's protocol controller is lowered to its slowest setting, and all radio services are stopped (receiver, transmitter, oscillator, etc.). The module wakes from sleep in less than 6ms. A low pulse on the RXD line wakes the module. This pulse should be at least 1 bit-time in duration, so sending any byte to the UART wakes it with the low start bit. Because the module's oscillator is not capable at running at ultra-low speeds, use of this mode is not recommended for new applications. The RAM contents are preserved during sleep. If the RAM fails an integrity check, the module issues itself a software reset to force re-initialization.

Deep Sleep

Deep sleep is selected by writing a 0x03 to *regOPMODE*. When the module is put into deep sleep, the $\overline{CTS}$ line is brought high to indicate that the module is not ready to accept UART data. The radio is placed in its lowest power mode and all services are stopped. The protocol controller's oscillator is also stopped and all non-essential functions are turned off. While powered, this mode consumes the least amount of current. The module wakes from deep sleep in less than 6ms. A low pulse of at least 20µs on the $\overline{RESET}$ line starts the waking process, but the module doesn't begin executing wake instructions until the $\overline{RESET}$ line is returned high. As with the other low-power modes, the RAM contents are preserved. If the RAM fails an integrity check, the module issues itself a software reset to force re-initialization. Note that, if the volatile data rate register is changed during the host application initialization (*regUARTDATARATE*), the re-initialization returns the module to the value in the non-volatile counterpart (*regNVUSERDATARATE*).

Reset to Factory Default

It may be necessary to reset the non-volatile registers to their factory defaults. To reset the module, hold the $\overline{\text{CMD}}$ line low and cycle power to hardware-reset the module. The $\overline{\text{CMD}}$ line must remain low for a minimum of 600ms after resetting the module. Once the $\overline{\text{CMD}}$ line is released, the module's non-volatile registers are reset to factory defaults.

Compatibility Mode

The 250 Series modules support a mode that allows them to communicate with the smaller, lower power 25 Series modules. The 250 Series operates at a much narrower receive bandwidth (200kHz) than the 25 Series (600kHz). To allow interoperability, the 250 and 25 Series transceivers support a compatibility mode that allows the modules to communicate effectively with each other.

Compatibility mode reduces the maximum RF data rate to 76.8kbps. All UART baud rates are supported, although the RF data rates associated with baud rates 31,250; 38,400; 57,600 and 115,200 are reduced.

Automatic Gain Control and Manual Gain Control

The gain setting of the receiver's low noise amplifier (LNA) is adjustable. By default, the 250 Series is factory-configured to use its internal automatic gain control (AGC) circuit to manage receiver sensitivity. Reducing the gain increases the linearity of the receiver, but reduces maximum sensitivity; increasing the gain does the opposite. Generally speaking, higher linearity (increased third order input intercept point, IIP3) gives improved performance in high-interference environments; high gain yields better performance in low-interference environments.

The module contains an AGC circuit that manages these settings automatically, and it should be used whenever possible. However, when attempting to make analog RSSI measurements, fixing the LNA gain produces more meaningful results. Digital RSSI readings are internally compensated and may be taken with AGC enabled.

Exception Engine

The modules are equipped with an internal exception engine. If errors occur during module operation, an exception is raised. Exception codes are stored in the *regEXCEPTION* register and are cleared once they are read. If an exception code is already present in *regEXCEPTION* when an error occurs, the new exception code overwrites the old value.

Exception Codes

Exception codes are organized by type for ease of masking. Figure 9 lists the exception codes and their meanings. All other values are reserved.

250 Series Transceiver Exception Codes		
Exception Code	Exception Name	Description
0x08	EX_BUFOVFL	Internal UART buffers overflowed.
0x09	EX_RFOVFL	Internal RF packet buffer overflowed.
0x13	EX_WRITEREGFAILED	Attempted write to register failed.
0x20	EX_NORFACK	Acknowledgement packet not received after maximum number of retries.
0x40	EX_BADCRC	Bad CRC detected on incoming packet.
0x42	EX_BADHEADER	Bad CRC detected in packet header.
0x43	EX_BADSEQID	Sequence ID was incorrect in ACK packet.
0x44	EX_BADFRAMETYPE	Unsupported frame type specified.

Figure 9: 250 Series Transceiver Exception Codes

Exception Masking

The EX line can be asserted to indicate to the host that an error has occurred. The exception mask provides a simple method of choosing which errors cause the line to toggle. If the result of ANDing the exception code with the exception mask is non-zero, the EX line is asserted. The regEXCEPTION register must be read to reset the line. Figure 10 lists some example exception masks.

250 Series Transceiver Example Exception Masks	
Exception Mask	Exception Name
0x08	Allows only EX_BUFOVFL and EX_RFOVFL to trigger the EX line
0x10	Allows only EX_WRITEREGFAILED to trigger the EX line
0x20	Allows only EX_NORFACK to trigger the EX line
0x40	Allows only EX_BADCRC, EX_BADHEADER, EX_BADSEQID and EX_BADFRAMETYPE exceptions to trigger the EX line
0x60	Allows EX_BADCRC, EX_BADHEADER, EX_BADSEQID, EX_BADFRAMETYPE and EX_NORFACK exceptions to trigger the EX line
0xFF	Allows all exceptions to trigger the EX line

Figure 10: 250 Series Transceiver Example Exception Masks

The exception mask has no effect on the exceptions stored in the exception register. It only controls which exceptions affect the EX line.

Networking Modes

The module has a very flexible addressing and networking scheme selected with the regNVNETWORKMODE and regNETWORKMODE registers. It can be changed during operation. The transmitting module addresses packets according to the network mode configuration. The receiving module processes all addressing types regardless of the network mode configuration. If the received message matches the addressing criteria, it is output on the UART. Otherwise it is discarded.

There are three networking modes: GUID, User and Extended User. Each mode offers different communications schemes, but all use source and destination addressing. The source address is for the transmitting unit, the destination address is the intended receiver. Each mode uses different registers for the source and destination addresses.

The module supports an automatic addressing mode that reads the Source Address from a received packet and uses it to fill the Destination Address register. This makes sure that a response is sent to the device that transmitted the original message. This also allows the host microcontroller to read out the address of the sending unit.

The automatic addressing is enabled for the different networking modes with register regAUTADD and regNVAUTADD.

GUID Networking Mode

GUID networking mode is the simplest mode and supports point-to-point and broadcast communications. Each module is programmed at the factory with a unique 4-byte ID number that cannot be changed. These bytes are found in the non-volatile read only MYGUID registers (regMYGUID[0-3]). GUID networking mode uses these IDs as addresses. The transmitting unit's GUID is used as the source address and the intended receiver's GUID is written into the destination address register (regDESTGUID[0-3]). All modules within range hear the transmission, but only the module with the ID that matches the destination address outputs the data on its UART. All others ignore the transmission.

A broadcast message is created when the destination address is 0xFFFFFFFF. In this case, all modules within range output the data. It is not recommended to send broadcast messages when acknowledgements are enabled. Figure 11 lists some examples of how GUID networking works.

250 Series Transceiver GUID Network Mode Examples				
Sender			**Receiver**	
Network Mode	MyGUID	Destination GUID	MyGUID	Response
0x04 (GUID)	0x00001000	0xFFFFFFFF	0x00002000	Data output by both modules. No RF ACK sent by either module.
			0x00003000	
0x14 (GUID + ACK)	0x00001000	0xFFFFFFFF	0x00002000	Data output by both modules. No ACK sent by either module. This configuration causes transmission problems.
			0x00003000	
0x14 (GUID + ACK)	0x00001000	0x00003000	0x00002000	Not processed – discarded.
			0x00003000	Data output. RF ACK sent to 0x00001000
0x04 (GUID)	0x00001000	0x00002000	0x00002000	Data output. No RF ACK sent.
			0x00003000	Not processed – discarded.

Figure 11: 250 Series Transceiver GUID Network Mode Examples

User Networking Mode

User Networking Mode is a more complicated scheme than GUID mode. It uses the customer ID bytes (regCUSTID[0-1]) and two of the user destination bytes (regUSERDESTID[0-1]) as a destination address. The customer ID bytes are programmed at the factory and cannot be changed. The module's local address is contained in two of the user source ID registers (regUSERSRCID[0-1]). Each module also has a user ID mask (regUSERIDMASK[0-1]) that provides an additional logical layer of addressing and can be used to create sub-networks. The receiving module masks its local address and the received destination address by calculating the logical AND with the user ID mask. If the results are equal, then the payload is output on the UART. The customer ID bytes are not masked, but must match the local value.

If acknowledgements are enabled, only the module with a user source ID that matches the transmitted user destination ID responds. The mask is not used for this determination.

If the result of the user ID Mask AND the received user destination address equals the same value as the user ID mask, then the payload data is output on the UART. This acts as a broadcast message to the network.

Setting the mask to 0xFFFF removes the mask and only the source and destination addresses are used for networking. When using user network mode to send packets to multiple users and the mask is not equal to 0xFFFF, acknowledgements must be disabled. Failure to do so could cause extreme delays in transmission and loss of data.

As an example, if the mask is 0xFFF0 and the destination address transmitted by the sender is 1234, then all modules with a source ID of 123x respond. This gives a subnet of 16 modules (where x = 0 to F) and acts as a broadcast message to the sub-net. Acknowledgements should be disabled.

Figure 14 shows this example and Figure 12 and Figure 13 show some more examples of user networking mode.

250 Series Transceiver User Network Mode Examples					
Destination ID from Received Packet	Receiver Source ID	Receiver User ID Mask	Result of Dest AND Mask	Result of Source AND Mask	Action
3000	2000	E000	2000	2000	The results are equal, so the payload is output on the UART.
	3000			2000	The results are equal, so the payload is output on the UART. The destination ID and the source ID match, so an ACK is transmitted if enabled.
	4000			4000	The results do not match, so the packet is discarded.
3000	2000	F000	3000	2000	The results do not match, so the packet is discarded.
	3000			3000	The results are equal, so the payload is output on the UART. The destination ID and the source ID match, so an ACK is transmitted if enabled.
	4000			4000	The results do not match, so the packet is discarded.
E000	2000	E000	E000	2000	The destination ID matches the user ID mask, so the data is output on the UART.
	3000			2000	
	4000			4000	

Figure 12: 250 Series Transceiver User Network Mode Examples

250 Series Transceiver User Network Mode Examples					
Sender			Receiver		
Network Mode	User SRCID	User DESTID	User SRCID	User IDMASK	Response
0x06 (User)	0x1000	0xFFFF	0x2000	0xFFFF	Data output by both modules. No ACK sent by either module.
			0x3000	0xFFFF	
0x16 (User + ACK)	0x1000	0xFFFF	0x2000	0xFFFF	Data output by both modules. No ACK sent by either module. This configuration causes transmission problems.
			0x3000	0xFFFF	
0x16 (User + ACK)	0x1000	0x3000	0x2000	0xE000	Data output. No ACK sent.
			0x3000	0xE000	Data output. ACK sent to 0x1000.
0x6 (User)	0x1000	0x3000	0x2000	0xF000	Not processed – discarded.
			0x3000	0xF000	Data output. No ACK sent.

Figure 13: 250 Series Transceiver User Network Mode Examples

250 Series Transceiver User Network Mode Examples

Destination ID from Received Packet	Receiver Source ID	Receiver User ID Mask	Result of Dest AND Mask	Result of Source AND Mask	Action
1234	Any module with 123x	FFF0	1230	1230	The results are equal, so the payload is output on the UART. **Do not enable acknowledgements**

Figure 14: 250 Series Transceiver User Network Mode Examples

Extended User Addressing Mode

Extended User Networking Mode is the same as User Networking Mode but uses longer addresses. The two customer ID bytes are still used (regCUSTID[0-1]) but all four bytes are used for the user destination address (regUSERDESTID[0-3]), user source ID (regUSERSRCID[0-3]) and user ID mask (regUSERIDMASK[0-3]). This provides more addressing capabilities at the expense of more overhead in the packet. Otherwise all functionality is the same.

250 Series Transceiver Extended User Network Mode Examples

Sender			Receiver		
Network Mode	User SRCID	User DESTID	User SRCID	User IDMASK	Response
0x07	0x10000000	0xFFFFFFFF	0x20000001	0XFFFFFFFF	Data output by both modules. No ACK sent by either module.
			0x20000002	0xFFFFFFFF	
0x17	0x10000000	0xFFFFFFFF	0x20000001	0xFFFFFFFF	Data output by both modules. No ACK sent by either module. This configuration will cause transmission problems.
			0x20000002	0xFFFFFFFF	
0x17	0x10000000	0x30000001	0x20000001	0xE0000000	Data output. No ACK sent.
			0x30000001	0xE0000000	Data output. ACK sent to 0x1000.
0x07	0x10000000	0x30000002	0x20000001	0xF0000000	Not processed – discarded.
			0x30000001	0xF0000000	Data output. No ACK sent.

Figure 15: 250 Series Transceiver Extended User Network Mode Examples

Assured Delivery (Acknowledgement)

While not an addressing mode on its own, assured delivery can be enabled for each of the addressing modes. When a module transmits with assured delivery enabled, it obligates the receiving module to return an acknowledgement packet. The transmitting module waits for this acknowledgement for a preset amount of time based on the data rate. If an acknowledgement is not received, it retransmits the packet. If the receiver receives more than one of the same packet, it discards the packet contents but sends an acknowledgment. This way, duplicate data is not output by the module. It is extremely important that assured delivery be used only when the unmasked user/extended user Destination ID or Destination GUID points to a specific module. Failure to specifically address a valid module could cause the module to appear slow or unresponsive due to repeated retransmissions. This also serves to congest the network, impeding valid communications.

If the received destination address matches the local address, the receiving module immediately sends an RF ACK packet. This packet lets the sending module know that the message has been received. An RF ACK packet is sent immediately following reception; CSMA delay is not applied to RF ACK packets. When the sending module receives the RF ACK packet, it marks the current block of data as completed. If this is the last message in the queue, the sending module asserts the BE line to indicate the state of the incoming buffer.

Troubleshooting Hint: If modules are unable to communicate with each other, check the following:

- Check to make sure that both modules are set to the same data rate. Modules programmed with different data rates will not communicate or share an RF channel with one another.

- Ensure that the network mode and addressing is configured to properly access the module of interest. Also, ensure that a specific module is addressed when using acknowledgment. Failure to do so causes large delays and loss of data.

Extended Preamble

As the receivers scan the hop sequence channels they look for the preamble from a transmitter. When they detect this preamble, they stop scanning and wait for a packet header. From the packet header they are able to lock on to the transmitter and synchronize the timing.

It is an advantage in some applications to keep the receivers asleep for long periods of time and wake them up only periodically to look for data. In this scenario it is quite possible for the receivers to miss the preamble from the transmitter and go to sleep without finding the transmission.

To address this issue, the modules can send an extended preamble. The extended preamble is much longer than the normal preamble and gives modules a larger window to detect and lock on to the transmitter.

There are two types of packets sent by the transmitter: synchronizing packets and data packets. The receivers use the synchronizing packets to lock on to the transmitters and follow them through the next hop. This packet is the first packet sent after every hop. The packets sent after the synchronizing packet and before the next hop are data packets. Receivers can catch and process data packets, but do not lock on and follow the transmitter unless they process the synchronizing packet.

It is important to note that an extended preamble packet may not be a synchronizing packet. If it is a data packet the receiver processes the packet and begins scanning again. It is important for the external processor or application to keep modules awake long enough to catch the synchronizing packet on the next hop. This ensures that the modules lock on to the transmitting module and receive all of the data.

The dwell time on each channel is approximately 395ms plus the times per baud rate shown in Figure 39. This can be shorter if the host processor determines that the necessary information has been received. Additionally the PR_PKT line can be used to determine that the module is processing a packet and keep it awake.

This page is intentionally blank.

Voltage Supply Rise Time

The power supply rise time is extremely important. It must rise from ground to 2.7V in less than 1ms. If this specification cannot be met, an external reset supervisor circuit must be used to hold the module in reset until the power supply stabilizes. Failure to ensure adequate power supply rise time can result in loss of important module configuration information.

Using the Buffer Empty (BE) Line

The BE line indicates the state of the module's UART buffer. When the module receives data in the RXD line and the $\overline{CMD}$ line is high, the BE line is lowered until all data in the buffer has been processed by the protocol engine. If acknowledgement is not enabled, the BE line is raised as soon as the protocol engine processes the outgoing packets. If acknowledgement is enabled, the buffer is not updated until either the data transmissions are acknowledged by the remote end or delivery fails after the maximum number of retries. When the BE line returns high, the EX line may be sampled, or the regEXCEPTION register polled to determine if an error occurred during transmission.

Using the Exception (EX) Line

The EX line indicates whether or not a module exception has occurred. The line is normally low, but it is raised if an exception occurs that passes masking. When the regEXCEPTION register is read, the exception is cleared and the EX line returns low. If more than one exception occurs before the regEXCEPTION register is read, the old exception is overwritten by the new one. Please see the Exception Engine section for more details.

Using the Processing Incoming Packet (PR_PKT) Line

The PR_PKT line indicates whether the protocol engine has determined there to be valid or potentially valid data incoming. The line is normally low (not processing). When awake and not transmitting, the protocol engine is constantly searching for incoming data. When scoring indicates that a potential packet is inbound, this line is raised until either the scoring falls below a given threshold or the complete packet is received. It is possible that the packet scoring will fall below the threshold during reception, causing the line to be lowered. Such an instance can occur when the module hops to a channel late in the transmitter's extended preamble. Since there aren't a large number of valid bits to score, the line may fall during the packet start sequence. Once this sequence arrives, the PR_PKT signal rises and latches for the duration of the packet reception.

Receive Signal Strength Indication (RSSI)

The RSSI line outputs an analog voltage that is proportional to the signal strength present on the channel at the time. In normal operation, the module is hopping rapidly from channel to channel. In this case, the RSSI value varies greatly and does not provide much useful information. However, it can be used to keep a module awake by sampling the RSSI line to determine if the module is processing a packet before putting it to sleep.

The 250 Series module has an internal digital RSSI indication of the immediate ambient environment and of the last good packet received. Additionally, the PR_PKT line can be used to indicate the state of the protocol engine.

RSSI level is dependent on the power of the signal received at the antenna port and the mode the LNA is in. regLNAMODE controls the mode of the internal LNA. Figure 16 shows typical traces of RSSI voltage versus signal strength.
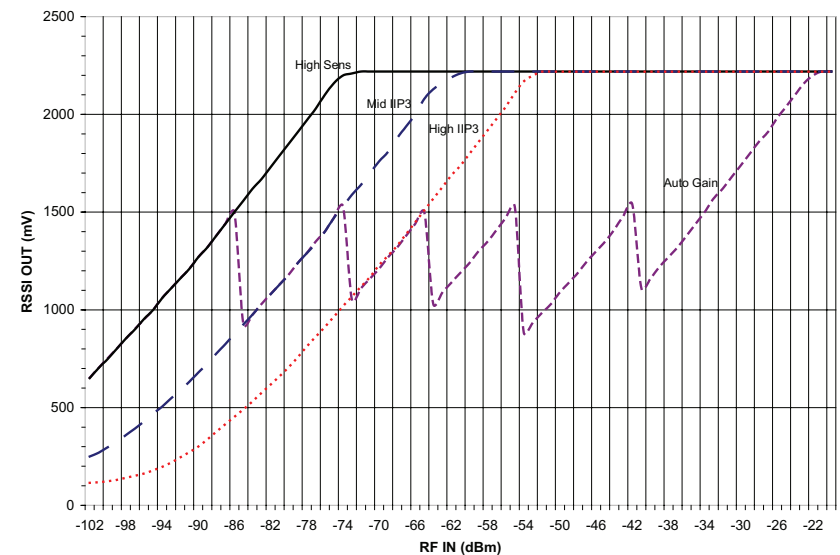

Figure 16: 250 Series Transceiver P_{IN} vs RSSI Voltage

Using the $\overline{\text{RESET}}$ Line

The $\overline{\text{RESET}}$ line has different functions depending on the state the module is in. It is an open-drain input/output line with an integrated weak pull-up, so it is normally high. Because it periodically operates as an output, external control should only pull this line low, not high.

Hardware Reset (Input)

During normal operation, the $\overline{\text{RESET}}$ line functions as an active-low hardware reset input. Taking this line low for at least 15µs forces the module's controller into hardware reset. While the line is low, execution of module operations are suspended and all module lines revert to open-drain inputs with weak pull-ups. This behavior can be exploited during power-up if the V_{CC} ramp time exceeds 1ms. By suspending execution, the dangers associated with slow V_{CC} ramp are eliminated.

Wake from Deep Sleep (Input)

When the module is in deep sleep, all execution is suspended in the controller and the radio is in its lowest power mode. The $\overline{\text{RESET}}$ line must be lowered for at least 15µs to wake the module. When the $\overline{\text{RESET}}$ line is raised, execution begins in the controller. The module maintains its state engine while asleep. Because of this, it can detect whether the hardware reset was intended to cause a hard reset or wake the module. The controller's RAM is preserved during deep sleep. The RAM is checked prior to entering deep sleep, and examined upon waking. If the RAM contents are corrupted upon wake, the module issues itself a software reset to reinitialize the module.

Hardware Reset Indicator (Output)

When the module starts from power-off, or is reset by the internal V_{CC} monitor circuitry, the $\overline{\text{RESET}}$ line is driven low to indicate the reset state. During power-on reset, assuming the V_{CC} ramp time is valid, $\overline{\text{RESET}}$ is driven low from the time that V_{CC} reaches approximately 1V until V_{CC} reaches $V_{RST} + T_{PORDelay}$. $T_{PORDelay}$ is the power-on reset delay imposed by the controller's hardware.

The other event that drives the $\overline{\text{RESET}}$ line low is a low-voltage or brown-out condition. In this case, the V_{CC} monitor holds the module in reset, thus driving the $\overline{\text{RESET}}$ line low. It remains low until the power drops below the operating threshold for that circuit (becoming indeterminate), or until the module's power supply returns to V_{RST}. Figure 17 illustrates the operation of $\overline{\text{RESET}}$ as an output.
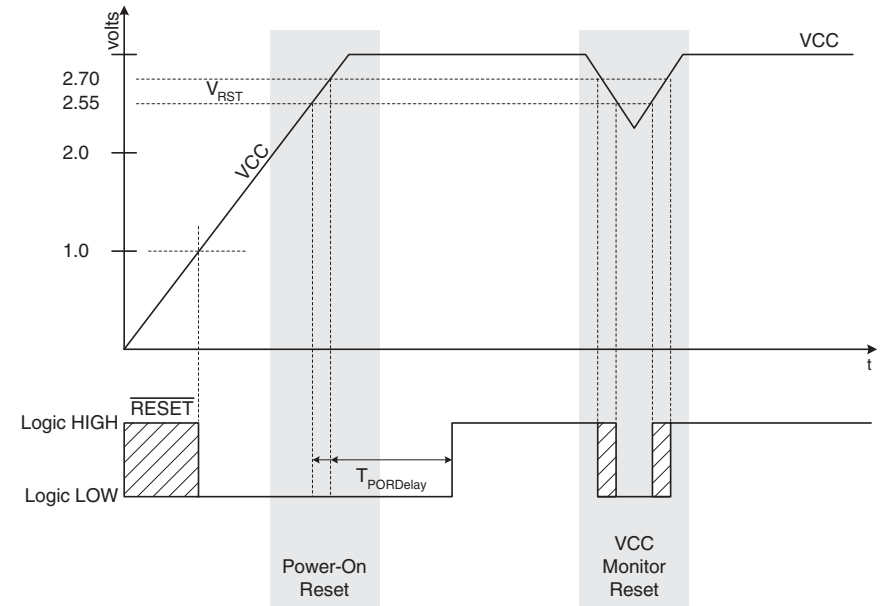
Figure 17: 250 Series Transceiver Reset Timing Diagram

250 Series Transceiver Reset Circuit Specifications

Parameter	Min.	Typ.	Max.	Units	Notes
$\overline{\text{RESET}}$ Output Low Voltage			0.6	V	V_{CC} = 2.7 – 3.6V
$\overline{\text{RESET}}$ Input Pull-up Current		25	40	µA	$\overline{\text{RESET}}$ = 0.0V
V_{CC} Monitor Threshold (V_{RST})	2.40	2.55	2.70	V	
Minimum $\overline{\text{RESET}}$ Low Time to Generate a Hardware Reset	15			µs	
Power-on Reset Delay ($T_{PORDelay}$)		<300		µs	V_{CC} Ramp Time is Valid
Allowed/Valid V_{CC} Ramp Time			1	ms	

Figure 18: 250 Series Transceiver Reset Circuit Specifications

Warning: If the $\overline{\text{RESET}}$ line experiences noise, it can cause multiple triggers (wake from sleep, hardware reset, hardware reset, etc.) and cause the volatile registers to be reloaded with their non-volatile values. If the circuit introduces noise onto this line, a bypass capacitor or RC filter should be placed on the line as close to the module as is practical.

Using the Command Response (CMD_RSP) Line

The $\overline{\text{CMD\_RSP}}$ line is normally high, but the module lowers this line when responding to a UART command. This indicates to an external processor that the data on the TXD line is a response to a command and not data received over-the-air.

The module outputs received RF data immediately following the command response. The $\overline{\text{CMD\_RSP}}$ line does rise before resuming RF data, but some processors cannot react quickly enough to this signal and may not able to separate the command responses from RF data.
The regCMDHALT register controls the behavior of the TXD line when the $\overline{\text{CMD}}$ line is low and the external processor is configuring the module.
If this register is set to 0x01 and the $\overline{\text{CMD}}$ line is low, the module stops outputting the RF data and internally buffers it. Once the $\overline{\text{CMD}}$ line is raised, the buffered RF data is output on the TXD line. This allows the external processor to have separate configuration times and data times instead of potentially having to handle both at once.

The CMD Line

The $\overline{\text{CMD}}$ line is used to inform the module where incoming UART data should be routed. When the line is high or left floating, all incoming UART data is treated as payload data and is routed to the transmitter to be sent over the air. If the $\overline{\text{CMD}}$ line is low, the incoming UART data is routed to the command parser for processing. Since the module's controller looks at UART data one byte at a time, the $\overline{\text{CMD}}$ line must be held low for the entire duration of the command plus a 20μs margin for processing. Leaving the line low for additional time (for example, until the ACK byte is received by the application) does not adversely affect the module. If RF packets are received while the $\overline{\text{CMD}}$ line is active, they are still processed and output on the module's UART. Figure 19 shows this timing.
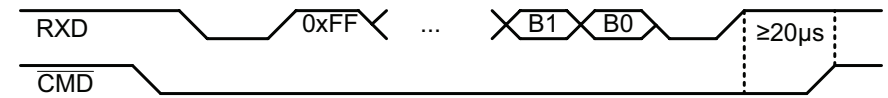
Figure 19: 250 Series Transceiver $\overline{\text{CMD}}$ Line Timing

The $\overline{\text{CMD}}$ line is also used during the module startup process to determine whether or not to reload the non-volatile registers with factory defaults. The module startup process is executed when the module is powered on from an off state or is issued a software or hardware reset. When the module goes through this startup process, it checks the state of the $\overline{\text{CMD}}$ line. If it is low, the module clears the non-volatile registers and re-populates them with factory default values. It is important to ensure that $\overline{\text{CMD}}$ is held high or left floating during power-up under normal conditions.

Possible reset sources that could cause the module to reboot are power supply brown-out, power supply instability and noise present on the $\overline{\text{RESET}}$ line, noise/voltage spikes on digital I/O lines, issuing a reset command through the command interface, and toggling the $\overline{\text{RESET}}$ line when not in deep sleep.

The UART Interface

The module uses a standard UART interface for both data to be sent over the air and for configuring the module. The $\overline{CMD}$ line is used to tell the module if the data on the UART is for configuration or transmission. The lines follow the standard UART naming convention, so RXD is the data input into the module and TXD is the data output from the module. The UART interface expects 1 start bit, 8 data bits (LSB first), and 1 stop bit per byte with no parity (8-N-1).

The module has a 256 byte buffer for incoming data. The module can be programmed to automatically transmit when the buffer reaches a limit or based on the time between bytes on the UART. This allows the designer to optimize the module for fixed length and variable length data. The module supports streaming data as well. To optimize the module for streaming data, regUARTMTU should be set to 128, and regTXTO should be set to a value greater than 1 UART byte time at the current UART data rate (10 bit times rounded up) or 2, whichever is greater.

If the buffer gets nearly full (about 224 bytes), the module pulls the $\overline{CTS}$ line high, indicating that the host should not send any more data. Data sent by the host while the buffer is full is lost, so the the $\overline{CTS}$ line provides a warning and should be monitored. When there is data in the UART receive buffer, the BE line is low; when this buffer is empty, BE is high.

Configuration Command Formatting

The 250 Series module contains several volatile and non-volatile registers that control its configuration and operation. The volatile registers all have non-volatile mirror registers that are used to determine the default configuration when power is applied to the module. During normal operation, the volatile registers are used to control the module.

Placing the module in the command mode allows these registers to be programmed. Byte values in excess of 127 (0x80 or greater) must be changed into a two-byte escape sequence of the format:

0xFE, [value - 128]

For example, the value 0x83 becomes 0xFE, 0x03. The function in Figure 20 prepends a 0xFF header and size specifier to a command sequence and creates escape sequences as needed. It is assumed that *src is populated with either the register number to read (one byte, pass 1 into

src_len) or the register number and value to write (two bytes, pass 2 into src_len). It is also assumed that the *dest buffer has enough space for the two header characters plus the encoded command and the null terminator.

```c
int EscapeString(char *src, char src_len, char *dest)
{
  // The following function copies and encodes the first
  // src_len characters from *src into *dest. This
  // encoding is necessary for module command formats.
  // The resulting string is null terminated. The size
  // of this string is the function return value.
  // -------------------------------------------------
  char src_idx, dest_idx;
  // Save space for the command header and size bytes
  // -------------------------------------------------
  dest_idx = 2;
  // Loop through source string and copy/encode
  // -------------------------------------------
  for (src_idx = 0; src_idx < src_len; src_idx++)
  {
    if (src[src_idx] > 127)
    {
      dest[dest_idx++] = 0xFE;
    }/*if*/
    dest[dest_idx++] = (src[src_idx] & 0x7F);
  }/*for*/
  // Add null terminator
  // -------------------
  dest[dest_idx] = 0;
  // Add command header
  // ------------------
  dest[0] = 0xFF;
  dest[1] = dest_idx - 2;
  // Return escape string size
  // -------------------------
  return dest_idx;
}
```

Figure 20: Command Conversion Code

Module Configuration

The 250 Series module contains several registers that control its configuration and operation. The module's default settings allow it to operate out of the box without any changes; however the registers allow the link to be customized to better suit the application if necessary. The register settings are stored in two types of memory inside the module. Volatile memory is quick to access, but it is lost when power is removed from the module. Non-volatile memory takes longer to access, but is retained when power is removed.

All of the configuration settings have registers in both types of memory. The settings are read from non-volatile registers on power up and saved in volatile registers. The values in the volatile registers are used during normal operation since it is faster to read and write the volatile memory locations. There are commands to read and write both locations.

Figure 21 shows the volatile read-only registers. Figure 22 shows the volatile read and write registers. Figure 23 shows the non-volatile read-only registers. Figure 24 shows the non-volatile read and write registers.

250 Series Volatile Read-Only Configuration Registers

Name	Address	Description
regEXCEPTION	0x79	Stores latest exception code
regLGPRSSI	0x7B	Last Good Packet RSSI value
regIMMEDRSSI	0x7C	Current RSSI value

Figure 21: 250 Series Volatile Read Only Configuration Registers

250 Series Volatile Read / Write Configuration Registers

Name	Address	Description
regCRCERRCOUNT	0x40	CRC error count value
regHOPTABLE	0x4B	Hop table
regPWRMODE	0x4D	Power amplifier setting
regUARTDATARATE	0x4E	UART data rate
regNETWORKMODE	0x4F	Sets the networking mode
regTXTO	0x50	UART to transmit timeout
regMAXTXRETRY	0x52	Maximum times to retry packet transmission
regUSECRC	0x53	Enable / Disable CRC checking
regUARTMTU	0x54	Minimum transmission unit
regCSMAMODE	0x56	Enable / Disable CSMA

250 Series Volatile Read / Write Configuration Registers Continued

Name	Address	Description
regOPMODE	0x58	Sets operating mode
regACKONWAKE	0x59	Enable / Disable ACK sent to UART upon wake
regUSERDESTID[3]	0x5A	Destination Address for Extended User Networking Mode
regUSERDESTID[2]	0x5B	Destination Address for Extended User Networking Mode
regUSERDESTID[1]	0x5C	Destination Address for User and Extended User Networking Mode
regUSERDESTID[0]	0x5D	Destination Address for User and Extended User Networking Mode
regUSERSRCID[3]	0x5E	Source Address for Extended User Networking Mode
regUSERSRCID[2]	0x5F	Source Address for Extended User Networking Mode
regUSERSRCID[1]	0x60	Source Address for User and Extended User Networking Mode
regUSERSRCID[0]	0x61	Source Address for User and Extended User Networking Mode
regUSERIDMASK[3]	0x62	Address Mask for Extended User Networking Mode
regUSERIDMASK[2]	0x63	Address Mask for Extended User Networking Mode
regUSERIDMASK[1]	0x64	Address Mask for User and Extended User Networking Mode
regUSERIDMASK[0]	0x65	Address Mask for User and Extended User Networking Mode
regDESTGUID[3]	0x68	GUID Networking Mode Destination Address
regDESTGUID[2]	0x69	GUID Networking Mode Destination Address
regDESTGUID[1]	0x6A	GUID Networking Mode Destination Address
regDESTGUID[0]	0x6B	GUID Networking Mode Destination Address
regEXCEPTIONMASK	0x6C	Exception and Mask used to activate the EX line
regCMDHALT	0x6E	Half RF traffic when the $\overline{CMD}$ line is low
regLNAMODE	0x6F	Receiver LNA gain / linearity setting
regCOMPATMODE	0x70	Compatibility mode for 25 and 250 intercommunication
regAUTADD	0x71	Sets automatic addressing

Figure 22: 250 Series Volatile Read / Write Configuration Registers

Warning: Modules that are not configured in the same way will not be able to communicate reliably, causing poor performance or outright failure of the wireless link. All modules in a network must have compatible configurations to ensure interoperability.

250 Series Non-Volatile Read-Only Registers

Name	Address	Description
regMyGUID[3]	0x34	Factory programmed GUID used in GUID Networking Mode
regMYGUID[2]	0x35	Factory programmed GUID used in GUID Networking Mode
regMYGUID[1]	0x36	Factory programmed GUID used in GUID Networking Mode
regMYGUID[0]	0x37	Factory programmed GUID used in GUID Networking Mode
regCUSTID[1]	0x39	Factory programmed customer ID, default 0xFF
regCUSTID[0]	0x3A	Factory programmed customer ID, default 0xFF
regRELEASENUM	0x78	Holds release number indicating h/w and f/w

Figure 23: 250 Series Non-volatile Read-Only Configuration Registers

250 Series Non-Volatile Read / Write Registers

Name	Address	Description	Factory Default
regNVHOPTABLE	0x00	Hop table	0
regNVPWRMODE	0x02	Power amplifier setting	3 (High Power)
regNVUARTDATARATE	0x03	UART data rate	0 (2400)
regNVNETWORKMODE	0x04	Sets the networking mode	4 (MAC/GUID)
regNVTXTO	0x05	UART to transmitter timeout	16 (15–16ms)
regNVMAXTXRETRY	0x07	Maximum times to retry packet transmission	26
regNVUSECRC	0x08	Enable/Disable CRC checking	1 (Enable)
regNVUARTMTU	0x09	Minimum transmission unit	64 (64 bytes)
regNVSHOWVERSION	0x0A	Enable/disable startup message	1 (Enabled)
regNVCSMAMODE	0x0B	Enable/Disable CSMA	1 (Enable)
regNVOPMODE	0x0D	Sets operating mode	0 (Awake)
regNVACKONWAKE	0x0E	Enable/Disable ACK sent to UART upon wake from	1 (Enable)
regNVUSERDESTID[3]	0x0F	Destination Address for Extended User Networking Mode	0xFF
regNVUSERDESTID[2]	0x10	Destination Address for Extended User Networking Mode	0xFF
regNVUSERDESTID[1]	0x11	Destination Address for User and Extended User Networking Mode	0xFF

250 Series Non-Volatile Read / Write Registers Continued

Name	Address	Description	Factory Default
regNVUSERDESTID[0]	0x12	Destination Address for User and Extended User Networking Mode	0xFF
regNVUSERSRCID[3]	0x13	Source Address for Extended User Networking Mode	0xFF
regNVUSERSRCID[2]	0x14	Source Address for Extended User Networking Mode	0xFF
regNVUSERSRCID[1]	0x15	Source Address for User and Extended User Networking Mode	0xFF
regNVUSERSRCID[0]	0x16	Source Address for User and Extended User Networking Mode	0xFF
regNVUSERIDMASK[3]	0x17	Address Mask for Extended User Networking Mode	0xFF
regNVUSERIDMASK[2]	0x18	Address Mask for Extended User Networking Mode	0xFF
regNVUSERIDMASK[1]	0x19	Address Mask for User and Extended User Networking Mode	0xFF
regNVUSERIDMASK[0]	0x1A	Address Mask for User and Extended User Networking Mode	0xFF
regNVDESTGUID[3]	0x1D	GUID Networking Mode Destination Address	0xFF
regNVDESTGUID[2]	0x1E	GUID Networking Mode Destination Address	0xFF
regNVDESTGUID[1]	0x1F	GUID Networking Mode Destination Address	0xFF
regNVDESTGUID[0]	0x20	GUID Networking Mode Destination Address	0xFF
regNVEXCEPTIONMASK	0x21	Used to mask exception for the EX line	0xFF (All)
regNVCMDHALT	0x23	Halt RF traffic when the $\overline{CMD}$ line is low	0 (Disabled)
regNVLNAMODE	0x24	Receiver LNA gain / linearity setting	0 (Auto)
regNVCOMPATMODE	0x25	Compatibility mode for 25 and 250 intercommunication	0 (Disabled)
regNVAUTADD	0x26	Sets automatic addressing	0 (Disabled)

Figure 24: 250 Series Non-volatile Read / Write Configuration Registers

Writing to Registers

Writing to a volatile register is nearly instantaneous. Writing to a non-volatile register typically takes 16ms. Because the packet size can vary based on the need for encoding, there are two possible packet structures. The first structure writes a value that is less than 128 (0x80) and the second writes a value that is higher. The higher value must be split into two values. Figure 25 shows the byte sequences for writing a register in each case.

250 Series Write to Configuration Register Command				
Command for a Value less than 128 (0x80)				
Header	Size	Address	Value	
0xFF	0x02	REG	V1	
Command for a Value greater than 128 (0x80)				
Header	Size	Address	Value 1	Value 2
0xFF	0x03	REG	0xFE	V2

Figure 25: 250 Series Write to Configuration Register Command

The module responds with an ACK (0x06). If it is not received, the command should be resent. The module responds with a NACK (0x15) if a write is attempted to a read-only or invalid register.

> **Warning:** The module must remain powered for the duration of the register write or important configuration information could be lost.

Reading from Registers

A register read command is constructed by placing an escape character (0xFE) before the register number. The module responds by sending an ACK (0x06) followed by the register number and register value. The register value is sent unmodified, so if the register value is 0x83, 0x83 is returned. If the register number is invalid, the module responds with a NACK (0x15). The command and response are shown in Figure 26.

250 Series Read From Configuration Register			
Command			
Header	Size	Escape	Address
0xFF	0x02	0xFE	REG
Response			
ACK	Address	Value	
0x06	REG	V1	

Figure 26: 250 Series Read from Configuration Register Command and Response

Configuration Registers

The following sections give details on each configuration register. Green addresses in the tables are volatile locations and blue are non-volatile.

CRC Error Count - Address = 0x40

The value in the regCRCERRCOUNT register is incremented each time a packet is received that fails CRC check. Writing 0x00 to this register initializes the count. Figure 27 shows the command and response.

250 Series CRC Error Count							
Read Command				**Read Response**			
Header	Size	Escape	Address	ACK	Address	Value	
0xFF	0x02	0xFE	0x40	0x06	0x40	V1	
Write Command							
Header	Size	Address	Value				
0xFF	0x02	0x40	V1				

Figure 27: 250 Series CRC Error Count Command and Response

Channel Hop Table - Address = 0x4B; NV Address = 0x00

The module supports 6 different hop sequences with minimal correlation. The sequence is set by the value in the regHOPTABLE register. Changing the hop sequence changes the physical band utilization, much the same way that a channel does in a static transmitter. Valid values are 0-5. Figure 28 shows the command and response.

250 Series Channel Hop Table							
Read Command				**Read Response**			
Header	Size	Escape	Address	ACK	Address	Value	
0xFF	0x02	0xFE	0x4B 0x00	0x06	0x4B 0x00	V1	
Write Command							
Header	Size	Address	Value				
0xFF	0x02	0x4B 0x00	V1				

Figure 28: 250 Series Channel Hop Table Command and Response

Figure 29 shows the RF channels used by the 250 Series and the hop sequences referenced by channel number. The default hop sequence is 0.

250 Series RF Channels and Hop Sequences

Channel Number	Frequency (MHz)	Hop Sequence by Channel Number					
		0	1	2	3	4	5
0	902.971	16	15	3	28	22	9
1	903.723	1	30	20	11	7	24
2	904.475	2	29	21	10	8	23
3	905.226	5	26	24	7	11	20
4	905.978	10	21	29	2	16	15
5	906.730	21	10	8	23	27	4
6	907.482	11	20	30	1	17	14
7	908.234	23	8	10	21	29	2
8	908.986	14	17	1	30	20	11
9	909.737	29	2	16	15	3	28
10	910.489	27	4	14	17	1	30
11	911.241	22	9	9	22	28	3
12	911.993	12	19	31	0	18	13
13	912.745	24	7	11	20	30	1
14	913.496	17	14	4	27	23	8
15	914.248	3	28	22	9	9	22
16	915.000	7	24	26	5	13	18
17	915.752	15	16	2	29	21	10
18	916.504	31	0	18	13	5	26
19	917.255	30	1	17	14	4	27
20	918.007	28	3	15	16	2	29
21	918.759	25	6	12	19	31	0
22	919.511	19	12	6	25	25	6
23	920.263	6	25	25	6	12	19
24	921.015	13	18	0	31	19	12
25	921.766	26	5	13	18	0	31
26	922.518						
27	923.270						
28	924.022						
29	924.774						
30	925.525						
31	926.277						

Figure 29: 250 Series RF Channels and Hop Sequences

Power Mode - Address = 0x4D; NV Address = 0x02

The value in the regPWRMODE register sets the module's output power.

Figure 30 shows the command and response and Figure 31 available power settings and typical power outputs for the module. The default setting is 0x03.

250 Series Power Mode

Read Command

Header	Size	Escape	Address
0xFF	0x02	0xFE	0x4D 0x02

Read Response

ACK	Address	Value
0x06	0x4D 0x02	PWR

Write Command

Header	Size	Address	Value
0xFF	0x02	0x4D 0x02	PWR

Figure 30: 250 Series Power Mode Command and Response

250 Series Power Mode Register Settings

PWR	Power Setting	Typical Output Power (dBm)
0x00	Low	+8
0x01	Mid – Low	+13
0x02	Mid – High	+18
0x03	High	+23.5

Figure 31: 250 Series Power Mode Settings

UART Data Rate - Address = 0x4E; NV Address = 0x03

The value in regUARTDATARATE sets the data rate of the UART interface. Changing the non-volatile register changes the data rate on the following power-up or reset. Changing the volatile register changes the data rate immediately following the command acknowledgement. Figure 32 shows the command and response and Figure 33 shows the valid settings.

250 Series UART Data Rate

Read Command					Read Response		
Header	Size	Escape	Address		ACK	Address	Value
0xFF	0x02	0xFE	0x4E 0x03		0x06	0x4E 0x03	V1

Write Command				
Header	Size	Address	Value	
0xFF	0x02	0x4E 0x03	V1	

Figure 32: 250 Series UART Data Rate Command and Response

250 Series UART Data Rate Register Settings

V1	Baud Rate
0x00	2,400
0x01	9,600
0x02	19,200
0x03	38,400
0x04	57,600
0x05	115,200
0x06	10,400*
0x07	31,250*

* These data rates are not supported by PC serial ports. Selection of these rates may cause the module to fail to respond to a PC, requiring a reset to factory defaults.

Figure 33: 250 Series UART Data Rate Settings

If the UART rate is different than the host processor UART rate then the module will not communicate correctly. If mismatched, every rate can be tested until the correct one is found or the module can be reset to factory defaults.

Network Mode - Address = 0x4F; NV Address = 0x04

The module supports three networking modes: GUID, User, and Extended User. For each of these modes, assured delivery (acknowledgement) and extended preamble can be either enabled or disabled.

Figure 34 shows the command and response and Figure 35 shows the valid settings.

250 Series Network Mode

Read Command					Read Response		
Header	Size	Escape	Address		ACK	Address	Value
0xFF	0x02	0xFE	0x4F 0x04		0x06	0x4F 0x04	V1

Write Command			
Header	Size	Address	Value
0xFF	0x02	0x4F 0x04	V1

Figure 34: 250 Series Network Mode Command and Response

250 Series Network Mode Register Settings

Network Mode	Meaning
0x04	GUID Networking Mode
0x06	User Networking Mode
0x07	Extended User Networking Mode
0x0C	GUID Networking Mode with Extended Preamble
0x0E	User Networking Mode with Extended Preamble
0x0F	Extended User Network Mode with Extended Preamble
0x14	GUID Networking Mode with Acknowledgement
0x16	User Networking Mode with Acknowledgement
0x17	Extended User Networking Mode with Acknowledgement
0x1C	GUID Networking Mode with Acknowledgement & Extended Preamble
0x1E	User Networking Mode with Acknowledgement & Extended Preamble
0x1F	Extended User Networking Mode with Acknowledgement & Extended Preamble

Figure 35: 250 Series Network Mode Register Settings

Transmit Wait Timeout - Address = 0x50; NV Address = 0x05

When a byte is received from the UART, the module starts a timer that counts down every millisecond. The timer is restarted when each byte is received. The value for the regTXTO register is the number of milliseconds to wait before transmitting the data in the UART receive buffer. The default setting for this register is 0x10 (~16ms delay).

If the timer reaches zero before the next byte is received from the UART, the module begins transmitting the data in the buffer. This timeout value should be greater than one byte time at the current UART data rate with a minimum of 0x02. It should not be set to a value of 0x01 or any value less than one byte time as unpredictable results could occur.

If the timeout value is set to 0x00, the transmit wait timeout is deactivated. In this case, the transceiver waits until a number of bytes equal to the Minimum Transmission Unit (MTU) have been received by the UART. All of the bytes are sent once the MTU has been reached. Figure 36 shows examples of the commands. Figure 37 shows the minimum timeout values based on baud rate.

250 Series Transmit Wait Timeout

Read Command

Header	Size	Escape	Address
0xFF	0x02	0xFE	0x50 0x05

Read Response

ACK	Address	Value
0x06	0x50 0x05	V1

Write Command

Header	Size	Address	Value
0xFF	0x02	0x50 0x05	V1

Figure 36: 250 Series Transmit Wait Timeout Command and Response

250 Series Minimum TXTO Values

Baud Rate	Minimum TXTO
2,400	6ms
9,600	3ms
19,200	2ms
38,400	2ms
57,600	2ms
115,200	2ms

Figure 37: 250 Series Transmit Wait Timeout Minimum Values

Maximum Transmit Retries - Address = 0x52; NV Address = 0x07

regMAXTXRETRY sets the number of transmission retries if an acknowledgement is not received. If an acknowledgement is not received after the last retry, EX_NORFACK is raised. Figure 38 shows examples of the command.

250 Series Maximum Transmit Retries

Read Command

Header	Size	Escape	Address
0xFF	0x02	0xFE	0x52 0x07

Read Response

ACK	Address	Value
0x06	0x52 0x07	V1

Write Command

Header	Size	Address	Value
0xFF	0x02	0x52 0x07	V1

Figure 38: 250 Series Maximum Transmit Retries Command and Response

The time between retries depends on the current baud rate. Figure 39 shows the time between retries based on baud rate. The retry number times the timeout times gives the potential latency before a new message can be sent.

250 Series Acknowledgement Timeout Times

Baud Rate	Timeout Time
2400	170ms
9600	75ms
19200	45ms
38400	30ms
57600	30ms
115200	30ms

Figure 39: 250 Series Acknowledgement Timeout Times

CRC Control - Address = 0x53; NV Address = 0x08

The 250 Series protocol includes a Cyclic Redundancy Check on the received packets to make sure that there are no errors. Any packets with errors are discarded and not output on the UART. This feature can be disabled if it is desired to perform error checking outside the module. Set the regUSECRC register to 0x01 to enable CRC checking, or 0x00 to disable it. The default CRC mode setting is enabled. Figure 40 shows examples of the commands and Figure 41 shows the available values.

250 Series CRC Control							
Read Command				Read Response			
Header	Size	Escape	Address	ACK	Address	Value	
0xFF	0x02	0xFE	0x53 0x08	0x06	0x53 0x08	V1	
Write Command							
Header	Size	Address	Value				
0xFF	0x02	0x53 0x08	V1				

Figure 40: 250 Series CRC Control Command and Response

250 Series CRC Control Register Settings	
V1	Mode
0x00	CRC Disabled
0x01	CRC Enabled

Figure 41: 250 Series CRC Control Register Settings

UART Minimum Transmission Unit - Addr = 0x54; NV Addr = 0x09

This register determines the UART buffer level that triggers the transmission of a packet. The minimum value is decimal 1 and the maximum value is 192. The default value for this register is 64, which provides a good mix of throughput and latency. At the maximum data rate, a value of 128 optimizes throughput. This register does not guarantee a particular transmission unit size; rather, it specifies the minimum desired size. If there is not enough time left in a hop, for instance, the protocol engine sends as many characters as it can to fill the current hop, and sends the remaining characters in the next hop. Figure 42 shows examples of the commands.

250 Series UART MTU							
Read Command				Read Response			
Header	Size	Escape	Address	ACK	Address	Value	
0xFF	0x02	0xFE	0x54 0x09	0x06	0x54 0x09	V1	
Write Command							
Header	Size	Address	Value				
0xFF	0x02	0x54 0x09	V1				

Figure 42: 250 Series UART MTU Command and Response

Show Version - Address = 0x0A

Setting this register to 0x00 suppresses the start-up message, including firmware version, which is sent to the UART when the module is reset. A value of 0x01 causes the message to be output after reset. By default, the module start-up message is output. Figure 43 shows examples of the commands and Figure 44 shows the available values.

250 Series Show Version

Read Command					Read Response		
Header	Size	Escape	Address		ACK	Address	Value
0xFF	0x02	0xFE	0x0A		0x06	0x0A	V1

Write Command			
Header	Size	Address	Value
0xFF	0x02	0x0A	V1

Figure 43: 250 Series Show Version Command and Response

250 Series Show Version Register Settings

V1	Meaning
0x00	Startup message is NOT output on reset or power-up.
0x01	Startup message is output on reset or power-up. This is a blocking call, and any incoming UART data is lost during the transmission of this message through the TXD line. All UART commands must be sent after this message has completed.
0x02	Startup message is displayed upon reset or power-up. This is a non-blocking call. Any incoming UART data is buffered, and incoming UART commands are processed. If a change of baud rate is commanded while the startup message is being output, the current byte finishes at the current baud rate, and subsequent bytes are transmitted at the new baud rate.

Figure 44: 250 Series Show Version Register Settings

CSMA Enable - Address = 0x56; NV Address = 0x0B

Carrier-Sense Multiple Access (CSMA) is a transmission protocol that listens to the channel before transmitting a message. If another module is already transmitting when a message is queued, the module waits before sending its payload. This helps to eliminate RF message corruption at the expense of additional latency. Setting the regCSMAMODE register to 0x01 enables CSMA and 0x00 disables CSMA. By default, CSMA is enabled. Figure 45 shows examples of the commands and Figure 46 shows the available values.

250 Series CSMA Enable

Read Command					Read Response		
Header	Size	Escape	Address		ACK	Address	Value
0xFF	0x02	0xFE	0x56 0x0B		0x06	0x56 0x0B	V1

Write Command			
Header	Size	Address	Value
0xFF	0x02	0x56 0x0B	V1

Figure 45: 250 Series CSMA Enable Command and Response

250 Series CSMA Enable Register Settings

V1	Mode
0x00	Disable CSMA
0x01	Enable CSMA

Figure 46: 250 Series CSMA Enable Register Settings