DN-900 Series

900 MHz Serial Modems

User Guide

Technical support +1.678.684.2000
E-mail: tech_sup@rfm.com

Important Regulatory Information

RFM Product FCC ID: HSW-DNT900 and HSW-DNT900P
IC: 4492A-DNT900 and 4492A-DNT900P

Note: This equipment has 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 in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Warning: Changes or modifications to this device not expressly approved by RFM could void the user's authority to operate the equipment.

RF Exposure Information, DN-900G, DN-900I and DN-900U:

For mobile operating conditions (greater than 20 cm to the body) - This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

For portable operating conditions (less than 20 cm to the body) - This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment may operate in direct contact with the body of the user under normal operating conditions. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

FCC Antenna Gain Restrictions, DN-900GX, DN-900IX and DN-900UX:

The DN-900GX, DN-900IX and DN-900UX have been designed to operate with any dipole antenna of up to 5.1 dBi of gain, or any Yagi of up to 6.1 dBi gain.

The antenna(s) used with these transmitters 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.

IC RSS-210 Detachable Antenna Gain Restriction, DN-900GX, DN-900IX and DN-900UX:

These devices have been designed to operate with the antennas listed below, and having a maximum gain of 6.1 dBi. Antennas not included in this list or having a gain greater than 6.1 dB are strictly prohibited for use with this device. The required antenna impedance is 50 ohms:

RFM RWA092R Omnidirectional Dipole Antenna, 2 dBi
RFM OMNI095 Omnidirectional Dipole Antenna, 5 dBi
RFM YAGI099 Directional Antenna, 6.1 dBi
RFM DNT900 Patch Antenna, 2 dBi

To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that permitted for successful communication.

Table of Contents

1.0 DN-900 Series 900 MHz Serial Modems

DN-900 series 900 MHz serial modems provide ready-to-use solutions for robust wireless data communications in the 900 MHz ISM band. There are currently six products in the DN-900 series - the DN-900G and DN-900GX, the DN-900I and DN-900IX, and the DN-900U and DN-900UX. The DN-900G and DN-900GX provide an RS-232C serial interface. The DN-900I and DN-900IX offer a two-wire, multi-drop RS-485 interface plus a selectable RS-232C interface for configuration programming. The DN-900U and DN900UX provide a USB interface. The modems without the "X" suffix incorporate internal antennas; modems with the "X" suffix are fitted with reverse TNC (RTNC) connectors for use with external antennas. The DN-900 series modems are based on RFM's DNT900 frequency hopping spread spectrum (FHSS) transceiver, and can communicate with other DNT900-based RFM products as well as customer developed products. DN-900 series modems consist of a radio module in a NEMA 4X/IP66 waterproof case connected to a serial interface adapter by a power and signal cable that allows the radio module to be located remotely at a point of good RF propagation. DN-900 series modems are well-suited for serial data networks carrying moderate traffic that need robust communications in locations with non-ideal RF propagation and/or where RF interference or noise are present.

Figure 1.0.1

Key Features of the DN-900 Series 900 MHz Serial Modems include:

- 900 MHz RS-232C, RS-485/RS-232C and USB Serial Modems
- Optional 128-Bit AES Encryption
- Point-to-point, Point-to-multipoint, Peer-to-peer and Tree-routing Network Capabilities
- Frequency Hopping Spread Spectrum Transceiver
- Transceiver Housed in a NEMA 4X/IP66 Waterproof Case
- RF Data Rate Configurable from 38.4 to 500 kbps
- Transmitter Power up to 1.58 W EIRP using internal antenna
- FCC and Canadian IC Certified for Unlicensed Operation

2.0 DN-900 Systems

DN-900 wireless serial modems can be configured to operate in one of three modes - *base, remote* or *router*. A base controls a DN-900 system, and interfaces to an application host such as a PC. A remote simply functions to transmit or receive data. A router alternates between functioning as a remote on one hop and a network base on the next hop. When acting as a remote, the router stores messages it receives from its *parent,* and then repeats the messages to its *child* radios when acting as a network base. Likewise, a router will store messages received from its child radios when acting as a base, and repeat them to its parent when acting as a remote. Any message addressed directly to a router is processed by the router rather than being repeated.

2.1 Point-to-Point Systems

A DN-900 system contains at least one network. The simplest DN-900 topology is a point-to-point system, as shown in Figure 2.1.1. This system consists of a base and one remote forming a single network. Point-to-point systems are often used to replace wired serial connections.

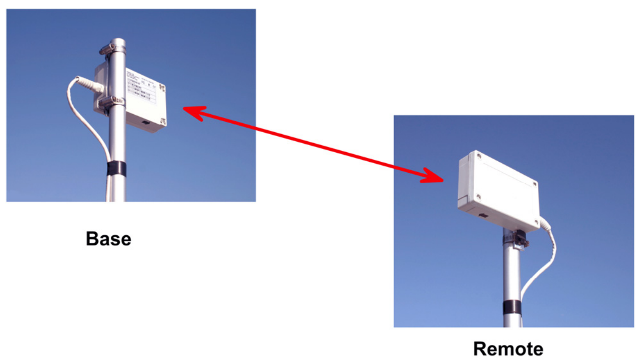

Figure 2.1.1

2.2 Point-to-Multipoint Systems

Figure 2.2.1 shows the topology of a point-to-multipoint (star) system, which consists of a base and more than one remote in a single network. Point-to-multipoint systems are typically used for data, sensor and alarm systems. While most traffic in a point-to-multipoint system is between the base and the remotes, DN-900 technology also allows for *peer-to-peer* communication from one remote to another.

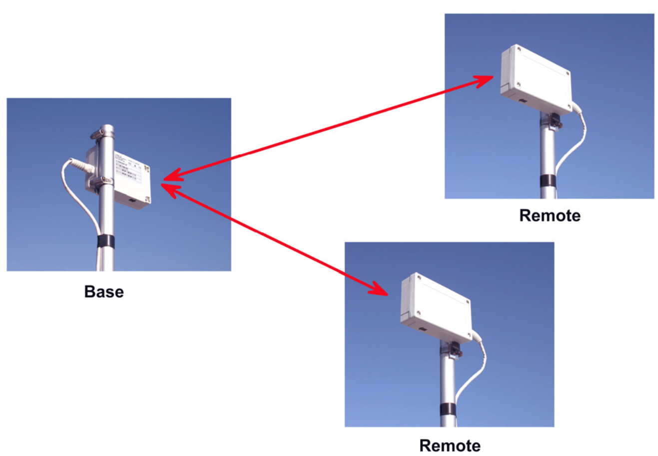

Figure 2.2.1

2.3 Tree-routing Systems

Figure 2.3.1 shows the topology of a tree-routing system, which consists of a base, one or more routers, one or more remotes, and *two or more* networks. Networks in a tree-routing system form around the base and around each router. The base and the routers are referred to as the *parents* of the networks they form. The rest of the radios in each network are referred to as *child* radios. Note that a router is a child of the base or another router while being the parent of its own network. Each network parent transmits beacons to allow child radios to synchronize with its hopping pattern and join its network. Different frequency hopping patterns are used by the parent radios in a system, minimizing interference between networks.

Tree-routing systems are used to cover a larger area than is possible with a point-to-point or point-to-multipoint system. The trade-off in tree routing systems is longer delivery times due to receiving and re-transmitting a message several times. Tree-routing systems are especially useful in applications such as agriculture where data is only collected and transmitted periodically.

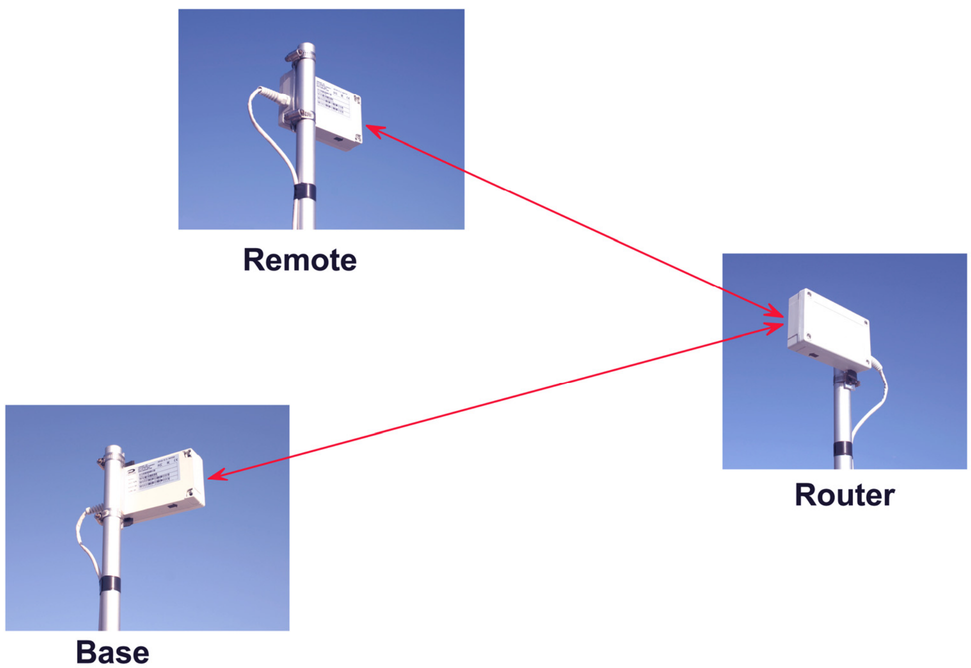

Figure 2.3.1

2.4 DN-900 Series Product Descriptions

The DN-900 series 900 MHz modems provide high performance, ready-to-use solutions for robust wireless data communications in the 900 MHz ISM band. There are currently six products in the DN-900 series - the DN-900G and DN-900GX, the DN-900I and DN-900IX, and the DN-900U and DN900UX. The DN-900G and DN-900GX provide an RS-232C serial interface with optional flow control. The DN-900I and DN900IX offer a two-wire, multi-drop RS-485 interface, plus a selectable three-wire RS-232C interface for configuration programming. The DN-900U and DN-900UX provide a USB interface. DN-900 series modems are based on RFM's DNT900 frequency hopping spread spectrum (FHSS) transceiver, and can communicate with other DNT900-based RFM products, as well as customer developed products.

Figure 2.4.1

As shown in Figure 2.4.1, DN-900G, DN-900I and DN-900U modems consist of a radio module in a NEMA 4X/IP66 waterproof case with an internal 2 dBi directional antenna, a signal cable that carries power and data, and an interface adaptor that includes a power supply connector and the appropriate serial data connector. The DN-900GX, DN-900IX and DN900UX radio modules are fitted with a reverse TNC (RTNC) antenna connector in lieu of the internal antenna (see Ordering Guide in Section 7.0). DN-900 serial modems with the RTNC connector are compatible with RFM's complete line of 900 MHz antennas, allowing extended operating range where allowed by local regulations.

DN-900 series modems are supplied with a 110/220 VAC wall-plug power supply that includes an international plug set. Optionally, DN-900 series modems can be powered from a user-supplied DC source as described in the System Specifications table (Section 3.2). The signal cable can specified as 4, 50, 100 or 300 feet in length (see Ordering Guide in Section 7.0), which allows the radio module to be located remotely indoors or outdoors at a point of good RF propagation. DN-900 modems can transmit data from 38.4 to 500 kbps, and the transmitter output power can be set from 1.58 mW to 1.58 W EIRP when using the internal antenna.

DN-900 series modems are well-suited for serial data networks carrying moderate traffic that need robust communications in locations with non-ideal RF propagation and/or where RF interference or noise are present. DN-900 serial modems can operate in point-to-point, point-to-multipoint, peer-to-peer and tree routing DNT900-based wireless networks.

DN-900 data modems are shipped configured to transmit transparent data, which requires no protocol formatting. All that is required to set up a transparent point-to-point serial data link to configure one modem as a base unit using a simple PC-based utility. DN-900 series modems can also operate in protocol mode, which is supported by a rich set of configuration parameters that allow a wide range of network layouts and configurations to be optimized.

2.5 DN-900 Modem Configuration

RFM provides a configuration utility program for the DN-900 series modems called DNWizard.exe. This program can be copied onto a PC (Windows XP or later operating system) and run. The start-up window is shown in Figure 2.5.1.

For interfacing DN-900G, DN900GX, DN-900I and DN900IX modems, the PC must be equipped with an RS-232C serial port or serial port adapter. Note that a DN-900I and DN-900IX must be in RS-232C mode to use this utility. The USB interface in the DN-900U and DN-900UX modems is based on an FT232RL serial/USB converter IC manufactured by FTDI. The latest drivers for this converter can be obtained from the FTDI website, www.ftdichip.com. The drivers create a virtual COM port on the PC. To install the drivers, power up a DN-900U or DN-900UX and connect it to the PC with a USB cable. The PC will find the new USB hardware and open a driver installation dialog box. Enter the path to the drivers and click *Continue*. The installation dialog will run *twice* to complete the FT232R driver installation.

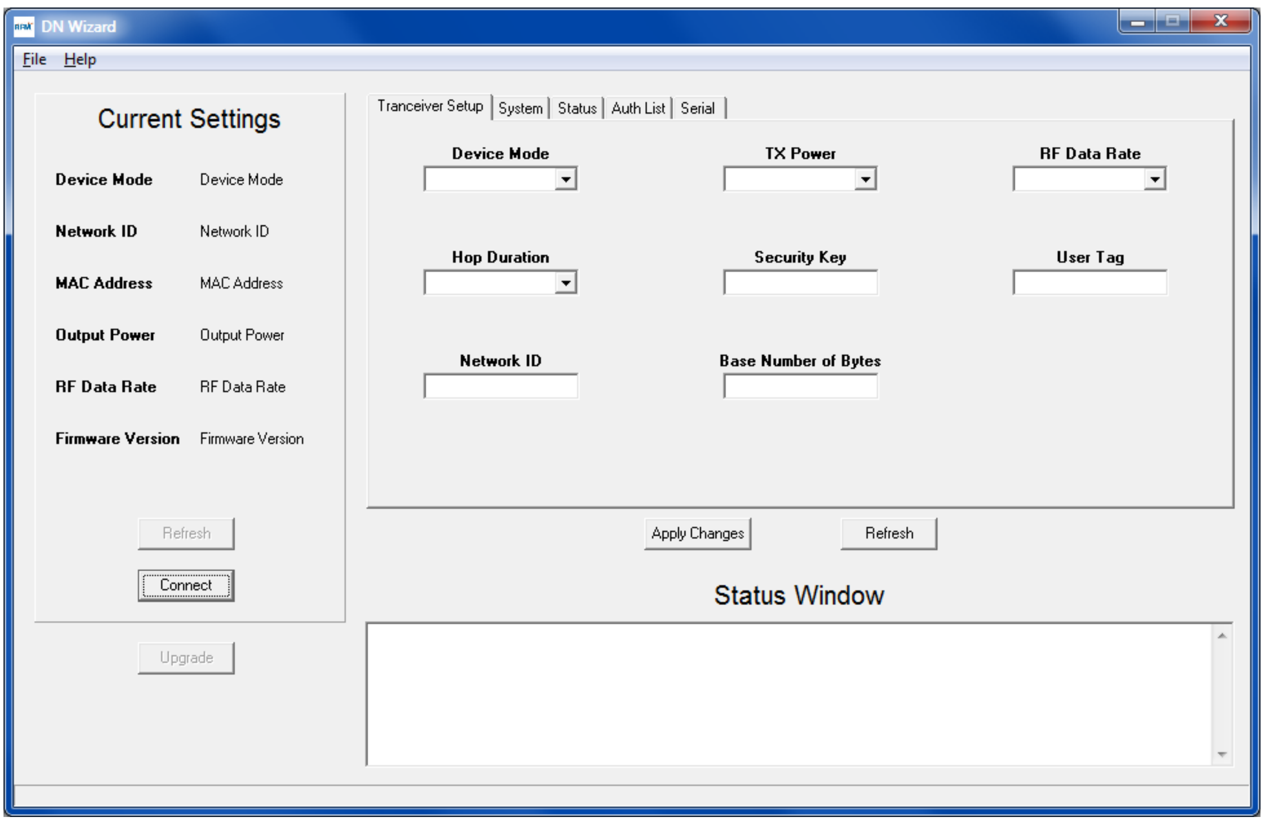

Figure 2.5.1

All DN-900 series modems are shipped preconfigured as *Remotes*. In each DN-900 system, one modem must be configured as a *Base*. This is easily done using DN Wizard. Connect the modem to be configured as a base to the PC, power the modem up and start DN Wizard.

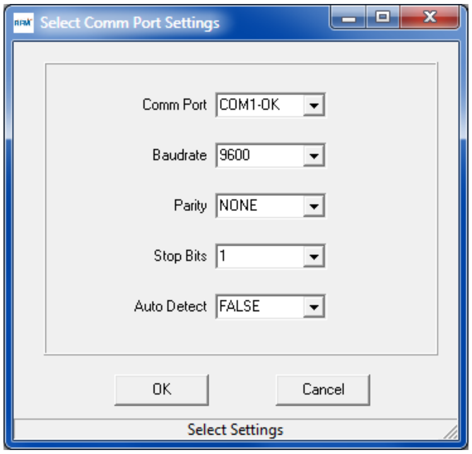

Figure 2.5.2

Click on the *Connect* button in the *DN Wizard* start-up window to open the dialog box shown in Figure 2.5.2. PCs with built-in serial ports will usually run *DN Wizard* on COM1. A serial port adapter and the virtual com port created by the FT232RL drivers will usually be on COM3 or higher. Click on the *Comm Port* drop-down control to select the serial port number associated with the connection to the DN-900. Leave the rest of the settings with their default values and click on *OK*.

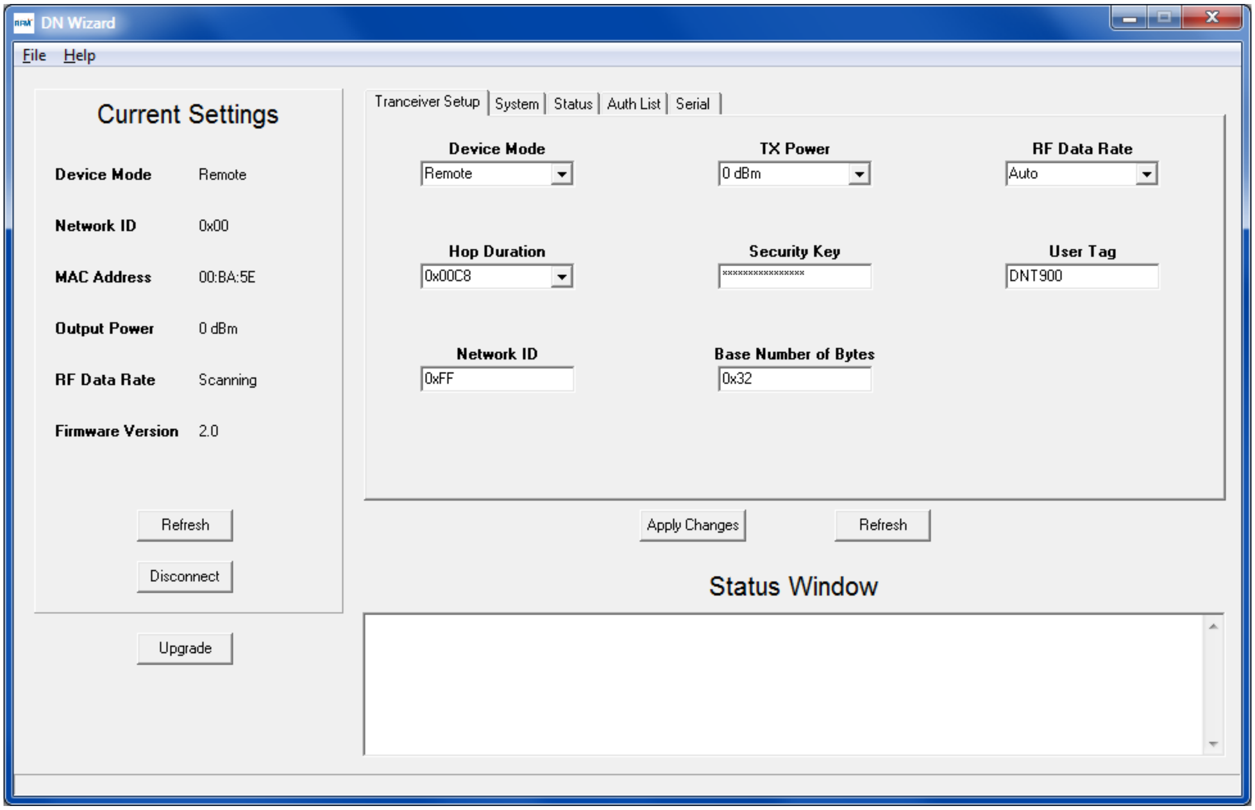

Figure 2.5.3

Once a serial connection is established, the *Current Settings* parameters will be displayed on the left side of the DN Wizard window. The current values of the eight parameters associated with the *Transceiver Setup* tab will also be displayed in their drop-down controls, as shown in Figure 2.5.3.

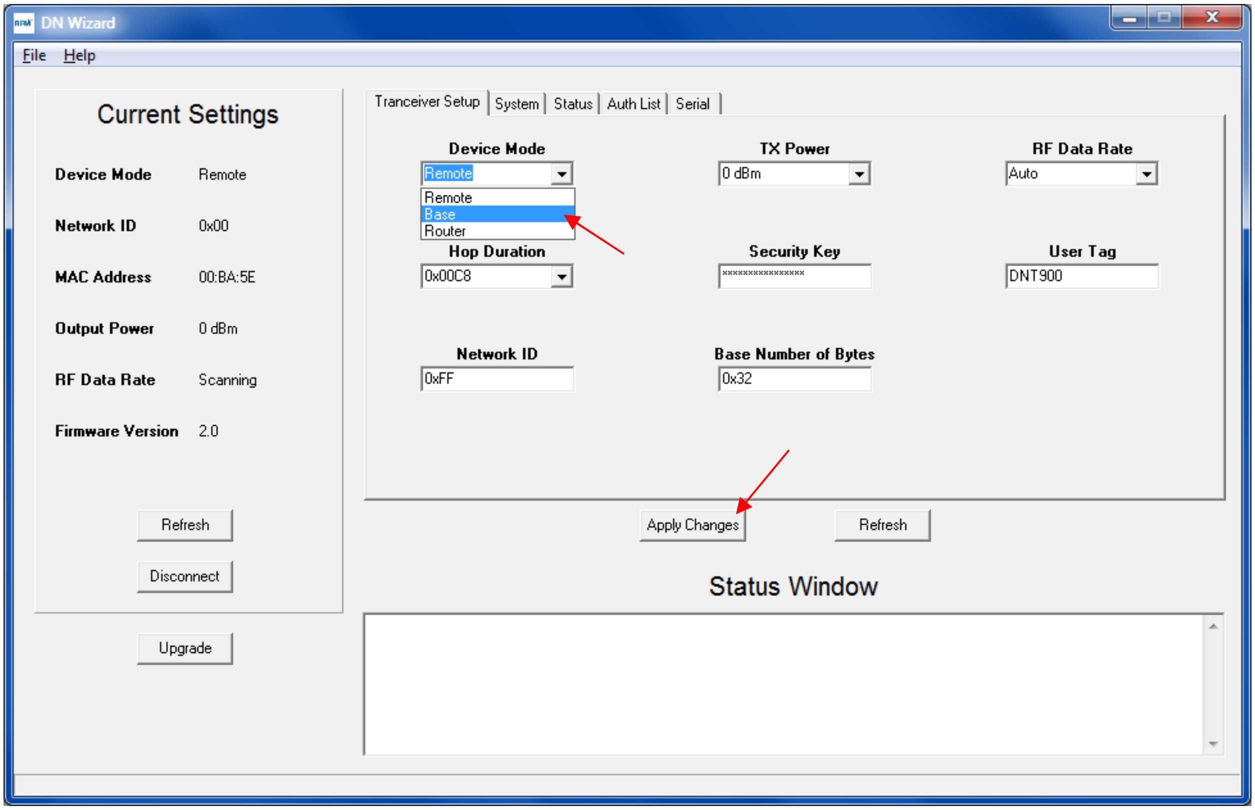

Figure 2.5.4

Select *Base* from the *Device Mode* drop down control and click on *Apply Changes*.

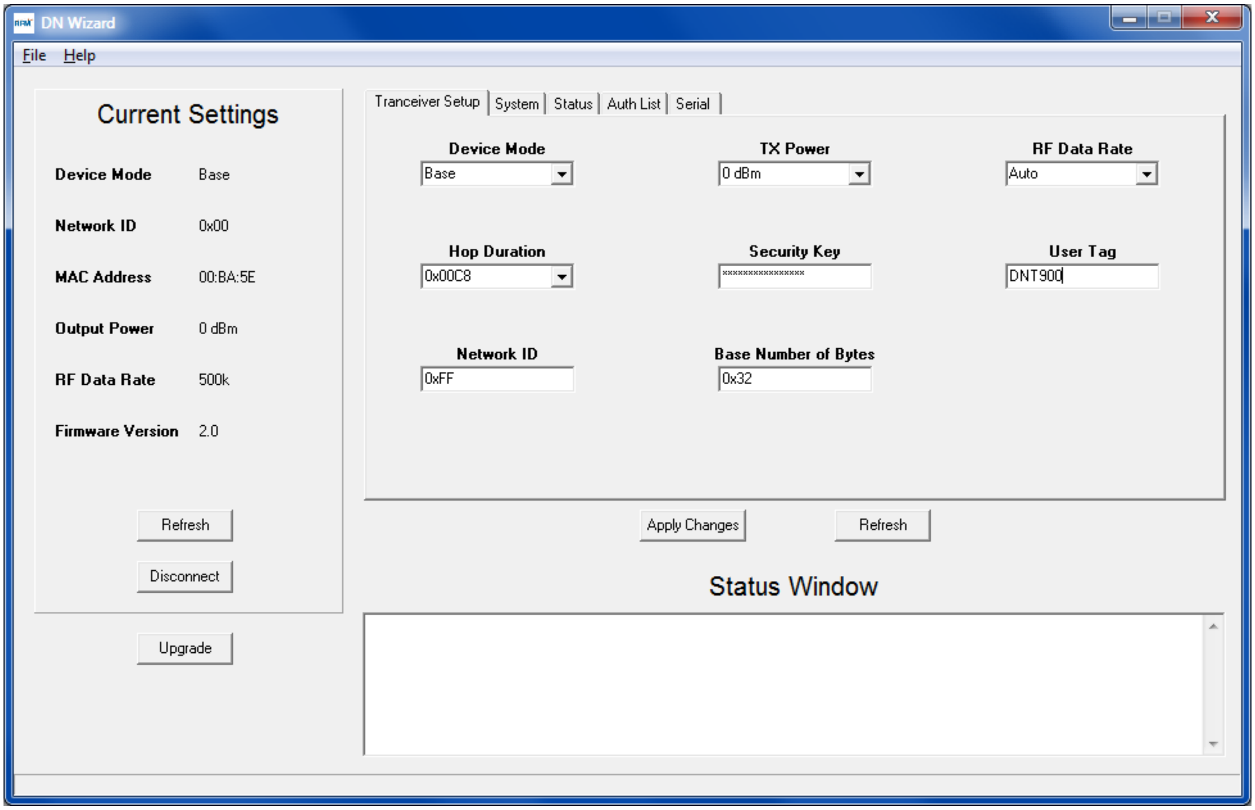

Figure 2.5.5

Click the *Refresh* button on the left side of the *DN Wizard* window to confirm the DN-900 is now in *Base* mode, as shown in Figure 2.5.5 above.

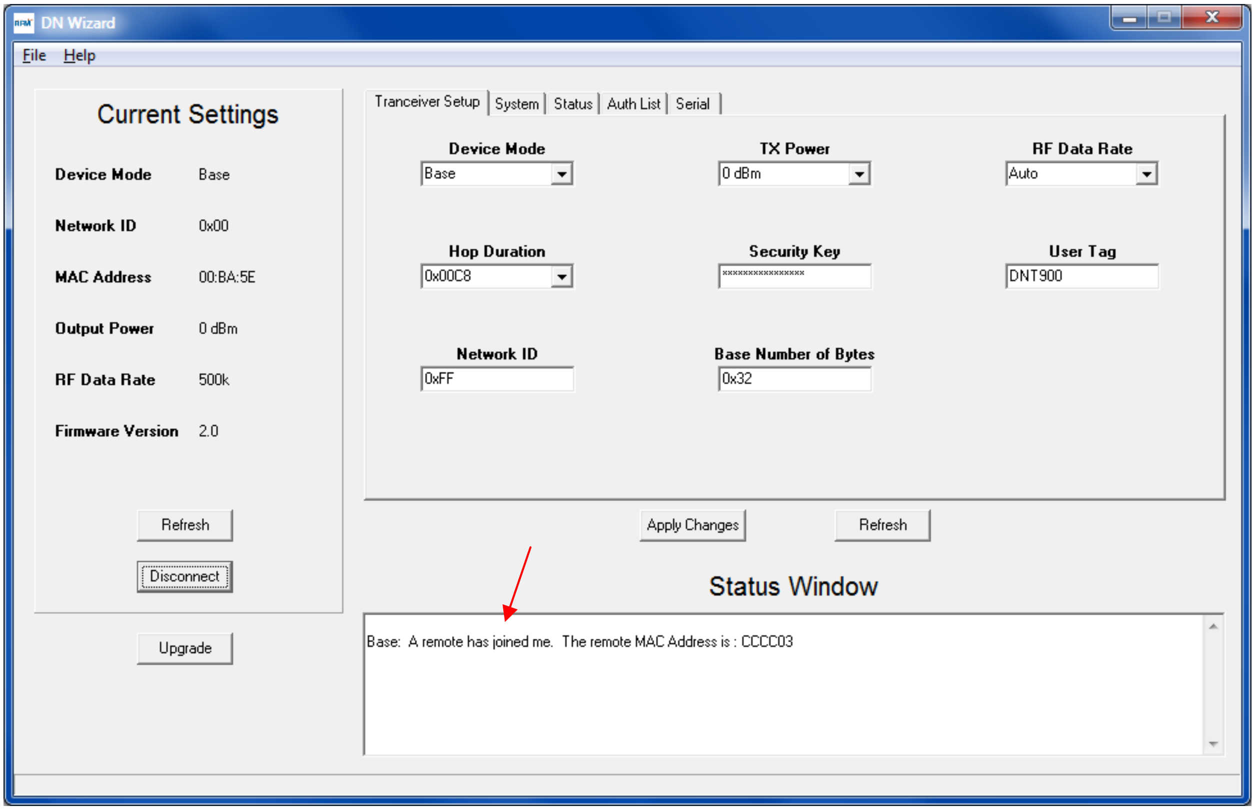

Figure 2.5.6

To conduct an initial test, power down the *Base* and exit *DN Wizard*. Then power up the *Base* and one *Remote*. Start *DN Wizard* and establish a serial connection with the *Base.* In a few moments a connection message should be displayed in the *Status Window*, as shown in Figure 2.5.6.

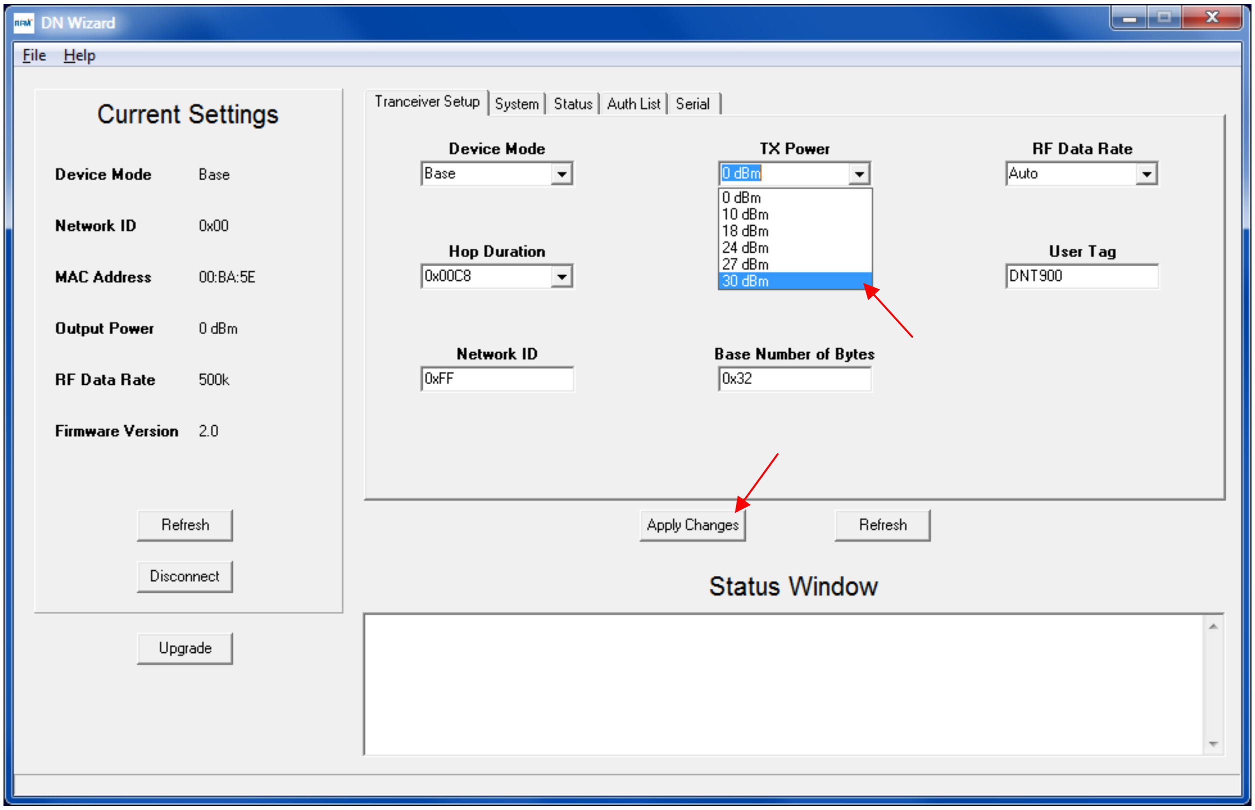

Figure 2.5.7

DN-900 transmitter power is set above the 0 dBm test level for most installations. The drop-down control under the *TX Power* label allows the selection of a number of power levels. Selecting the 30 dBm power level provides maximum range for mast-mounted outdoor installations.

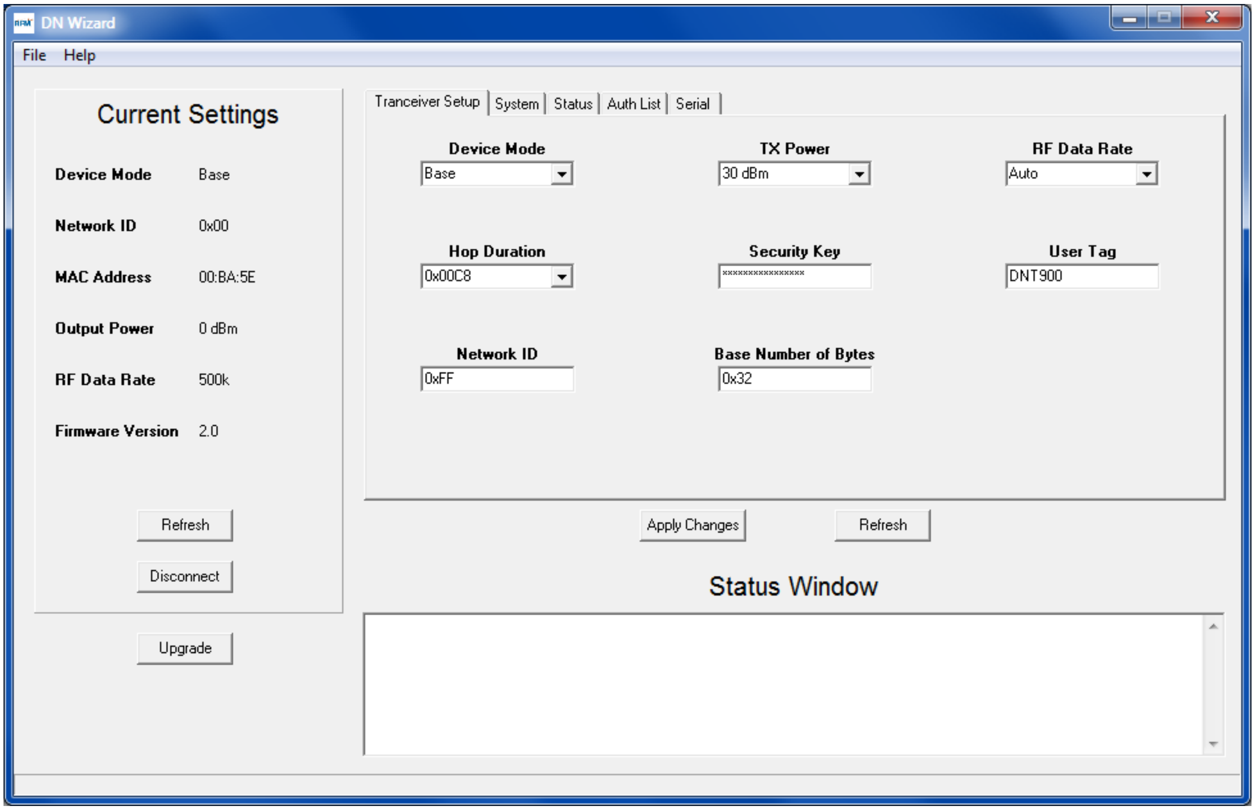

Figure 2.5.8

Figure 2.5.8 shows the updated *TX Power* level. The rest of the parameters on the *Transceiver Setup* tab and the other configuration tabs are discussed in Section 6 of this document.

Figure 2.5.9

Figure 2.5.9 shows the parameters on the *System* tab. Note that the *Auth List* parameter tab supports *Auth Mode* when it is set to *By Base Table*.

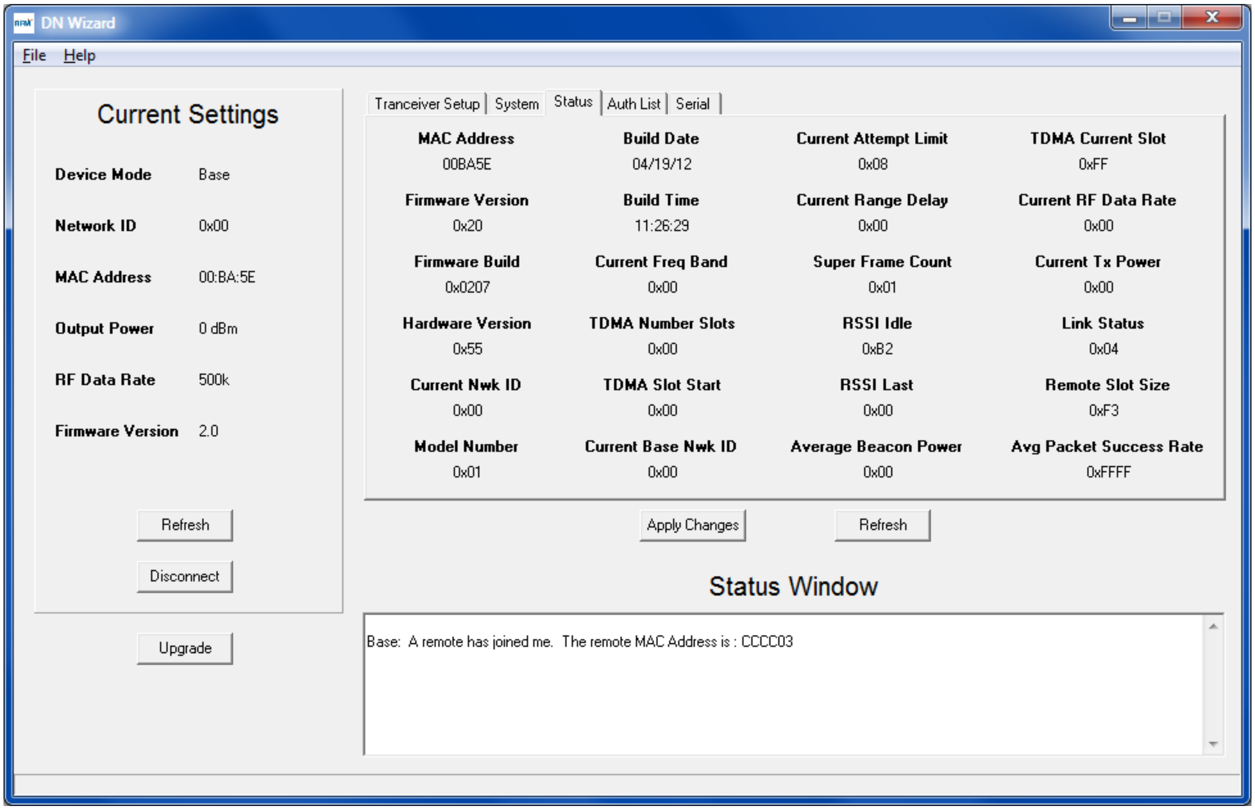

Figure 2.5.10

The *Status* tab displays read-only parameters useful in evaluating system performance. See Section 6 of this document for parameter details.

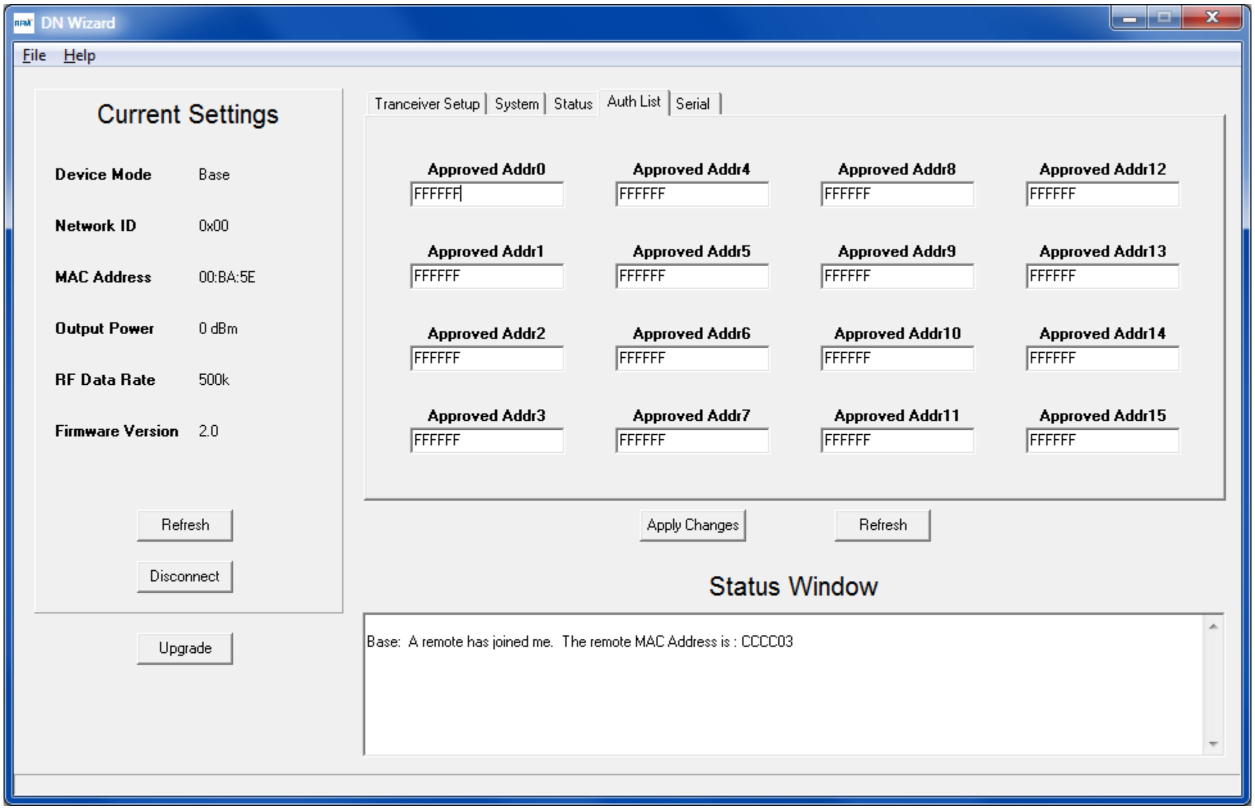

Figure 2.5.11

Figure 2.5.11 shows the MAC addresses of the Remotes authorized to join the Base when *Auth Mode* is set to *By Base Table*. A MAC address 0xFFFFFF indicates an unused approved address entry.

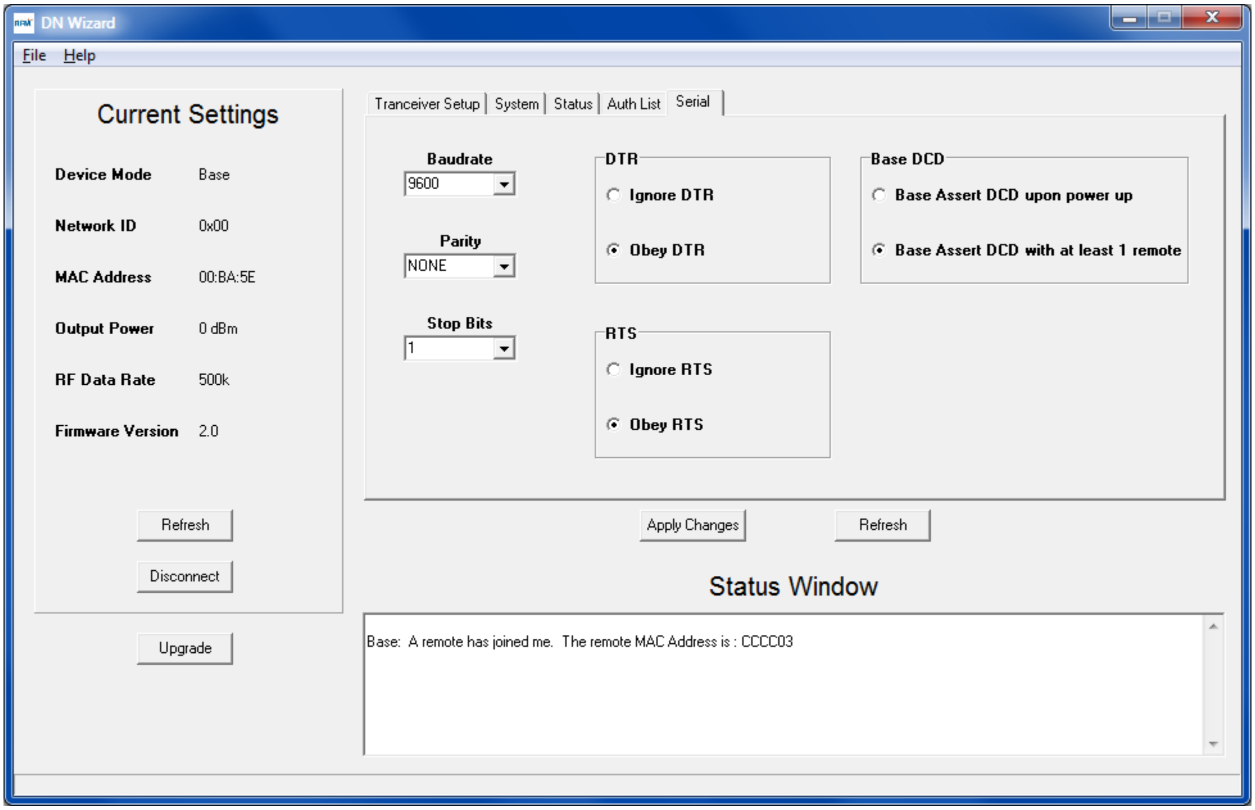

Figure 2.5.12

The *Serial* parameter tab allows the serial port parameters to be modified. DN Wizard tracks most changes automatically, preserving the serial connection.

3.0 Specifications

3.1 Absolute Maximum Ratings

Rating	Value	Units
Power Supply Input Voltage Range	-0.5 to +24	V
Non-Operating Ambient Temperature Range	-40 to +85	^oC

3.2 System Specifications

Characteristic	Notes	Minimum	Typical	Maximum	Units
Operating Frequency Range		902.75		927.25	MHz
Hop Dwell Time		5		200	ms
Number of RF Channels			50		
Modulation			FSK		
RF Data Transmission Rate			38.4, 115.2, 200 and 500		kbps
Standard Antenna			Internal 2 dBi directional antenna		
Receiver Sensitivity through 2 dBi Antenna:					
10^{-5} BER @ 38.4 kbps			-110		dBm
10^{-5} BER @ 200 kbps			-100		dBm
10^{-5} BER @ 500 kbps			-96		dBm
EIRP Output Power Levels through 2 dBi Antenna:					
38.4 to 200 kbps			1.58, 15.8, 100, 395, 790, 1580		mW
500 kbps			1.58, 15.8, 135		mW
Optional External Antenna Connector (“X” suffix)			RTNC		
External Antenna Impedance			50 ohms, VSWR < 3:1		
Direct Receiver Sensitivity through RTNC:					
10^{-5} BER @ 38.4 kbps			-108		
10^{-5} BER @ 200 kbps			-98		
10^{-5} BER @ 500 kbps			-94		
Direct Output Power Levels through RTNC:					
38.4 to 200 kbps			1, 10, 63, 250, 500, 1000		mW
500 kbps			1, 10, 85		mW
Network Topologies			Point-to-Point, Point-to-Multipoint, Peer-to-Peer, Tree Routing		
Access Scheme			TDMA/CSMA		
Number of Remote Nodes, TDMA Mode		1		16	
Number of Remote Nodes, CSMA Mode			Limited only by traffic density		
DN-900G/DN-900GX RS-232C Configuration			9-pin connector, flow control optional		
DN-900I/DN-900IX RS-232C Configuration			3-wire, no hardware flow control		
DN-900I/DN-900IX RS-485 Configuration			2-wire, multi-drop capable		
DN-900U/DN-900UX USB Configuration			USB Type B Connector		
Serial Port Baud Rates			1.2, 2.4, 4.8, 9.6, 19.2, 28.8, 38.4, 57.6, 76.8, 115.2, 230.4, 460.8		kbps
Signal Cable Length (see Ordering Guide in Section 7)			4, 50, 100, 300		ft
Power Supply Voltage Range, 4 to 50 ft of cable		+6		+24	Vdc
Power Supply Voltage Range, 50 to 300 ft cable		+12		+24	Vdc
Peak Transmit Mode Power				6.8	W

Characteristic	Notes	Minimum	Typical	Maximum	Units
Average Receive Mode Power:					
Base, Continuous Data Stream			760		mW
Remote, linked, No Data			317		mW
Remote, Continuous Data Stream, 9.6 kbps			345		mW
Remote, Continuous Data Stream, 115.2 kbps			428		mW
Interface Adapter Nominal Dimensions		3.85 x 1.70 x 0.85 inches (80 x 43 x 22 mm)			
Radio Module Case Rating		NEMA 4X/IP66 outdoor enclosure			
Radio Case Nominal Dimensions		5.15 x 5.00 x 1.40 inches (131 x 127 x 36 mm)			
Radio Case Mounting		Flange with Pre-drilled Holes; Mount Flange to Mast with Bolts or Hose Clamps			
Operating Temperature Range		-40		85	°C
Operating Relative Humidity Range, Non-condensing		10		90	%

4.0 Installation

4.1 Mounting and Location

The case drawing for the DN-900 series radios is shown below. The case should be mounted with its flange vertical. Arrows on the top and bottom of the DN-900G, DN-900I and DN-900U cases point in the direction of maximum antenna gain. A reverse TNC (RTNC) connector is located on the flange end of the DN-900GX, DN-900IX and DN-900UX cases for connecting external antennas. For best performance, the case should be mounted close to the external antenna to minimize antenna cable loss. A 24 inch RTNC to N connector adaptor cable, CBLRF24NR, is available for connecting external antennas with N connectors. Standard mounting is on a 1-inch aluminum mast. The case can be attached with either bolts or hose clamps. A DN-900 case or external antenna should be mounted at least six feet above the floor (ground) to achieve good communication in a typical office or retail location. To achieve maximum range and reliability, especially in industrial locations, a DN-900 case or external antenna should be mounted as high as practical. The radio is housed in a NEMA 4X/IP66 waterproof case, allowing it to be located indoors or outdoors. The signal cable should be dressed down the mounting mast with electrical tape.

DN-900 Series Transceiver Case

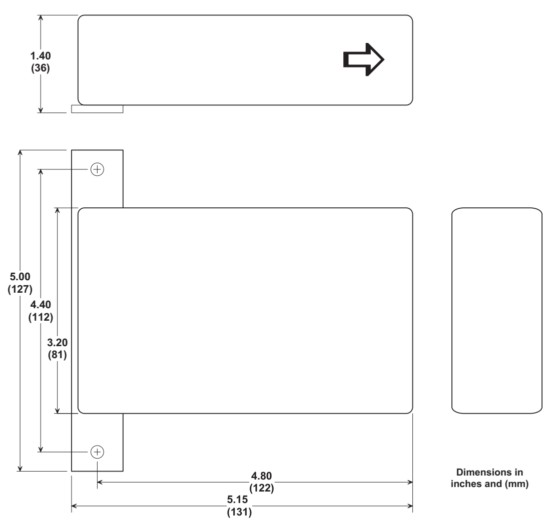

Figure 4.1.1

4.2 DN-900 Series Adapter Connector/Terminal Block Configurations

DN-900G, DN-900GX, DN-900U and DN-900UX Power Connection and LED Description

Pin	Name	I/O	Description
T	+PWR	I	The center coaxial conductor (tip) is the positive DC power input.
R	GND	-	The outer coaxial conductor (ring) is the DC power ground.
G	PWR	O	Green LED indicates the unit is powered.
A	LINK	O	Amber LED indicates the unit is linked.

Figure 4.2.1

DN-900G and DN-900GX Serial Port DB9 Connector Description

Pin	Name	I/O	Description
1	DCD	O	This pin is an output indicating the modem is linked to the radio network.
2	RADIO_TXD	O	This pin is the DN-900G serial data output.
3	RADIO_RXD	I	This pin is the DN-900G serial data input.
4	DTR	I	This pin is the data terminal ready input from the DN-900G host.
5	GND	-	This pin is signal ground.
6	-	-	No connection.
7	HOST_RTS	I	This pin is the request to send input from the DN-900G host.
8	HOST_CTS	O	This pin is the clear to send output from the DN-900G.
9	-	-	No connection.

Figure 4.2.2

DN-900I and DN-900IX Serial Port Terminal Block Description

Pin	Name	I/O	Description
1	+PWR	I	This terminal is the positive power supply input, +6 to +24 V.
2	GND	-	This terminal is a power supply and signal ground.
3	485 EN	I	When this terminal is unconnected, serial operation is RS-232C. When this terminal is grounded to terminal 2 or 4, serial operation is RS-485.
4	GND	-	This terminal is a power supply and signal ground.
5	TX/A	O	This terminal transmits data to the host (RS-232C TxD or RS-485 A signal).
6	RX/B	I	This terminal receives data from the host (RS-232C RxD or RS-485 B signal).

Figure 4.2.3

5.0 DN-900 Radio Advanced Information

5.1 Network Synchronization and Registration

Frequency hopping spread spectrum radios such as the DNT900 used in the DN-900 series wireless modems periodically change their transmit frequency. In order for the other radios in the network to receive the transmission, they must be listening to the frequency on which the current transmission is being sent. To do this, all the radios in the network must be synchronized to the same hopping pattern.

In all DN-900 networks, one radio is designated as the base. All other radios are designated as remotes or routers. The base transmits a beacon each time it hops to a different frequency, which allows the other radios in its network to synchronize with it. Since all radios in the network know the hopping pattern, once they are synchronized with the base, they know which frequency to hop to and when.

When a remote or router is powered on, it rapidly scans the frequency band for the synchronizing signal. Since the base is transmitting on up to 50 frequencies and the remotes and routers are scanning up to 50 frequencies, it can take several seconds to synchronize with the base.

Once a radio has synchronized with the base, it will request registration information to allow it to join the network. Registration is handled automatically by the base. When a radio is registered, it receives several network parameters from the base, including *HopDuration*, *InitialNwkID*, *FrequencyBan*d and *Nwk_Key*. Note that if a registration parameter is changed at the base, it will update this parameter in the remotes or routers over the air.

When leasing is enabled, registration also allows the base to track radios entering and leaving a network, up to a limit of 126. The base builds a table of serial numbers of registered radios using their three-byte MAC addresses. To detect if a radio has gone offline or out of range, a registration is leased and must be renewed within the configured lease interval. DN-900 radios automatically send lease renewal request to the base. There is nothing a remote host needs to do to keep the lease renewed. Note that more than 126 radios can join a network, but base-managed leasing cannot be used. In this case, the base can be configured to send join announcements to a host application for an unlimited number of radios. The application can then verify the continued presence of each radio in the network through periodic polling.

The DN-900 also supports a *RemoteLeave* command that allows a host application to release a radio from the network. This is useful to remove any rogue radios that may have joined then network when authentication is not being used. It is also useful to remove remotes from the network once they have been serviced by the application. The *RemoteLeave* command includes the amount of time the radio must leave the network, which can be set from 2 seconds to more than 36 hours. In addition, a radio can be told to leave and not rejoin until it has been power-cycled or reset. The base can use *RemoteLeave* to keep track of remotes that have not yet been serviced, allowing networks of more than 126 remotes to be indirectly tracked by the base.

5.2 Authentication

In many applications it is desirable to control which radios can join a network. This provides security from rogue radios joining the network and simplifies network segregation for co-located networks. Registration is controlled by the *AuthMode* parameter in the base. The *AuthMode* parameter can be set to one of four values, 0..3. The default value is 0, which allows any remote or router to register with a base.

When the *AuthMode* parameter is set to 1, a radio's address must be listed in Parameter Bank 7 before it will be allowed to register with the base. This is referred to as base authentication. Bank 7 must be pre-loaded with the addresses of the authorized remotes before using base authentication. If a radio whose MAC address is not in Bank 7 attempts to join the network the base will deny the registration request. A maximum of 16 remotes can be entered into Bank 7. To support larger networks, mode 2 must be used.

When the *AuthMode* parameter is set to 2, the address of a radio attempting to register with the base is sent to the host for authentication in a *JoinRequest* message. The host application determines if the radio should be allowed to register and returns a *JoinReply* message to the base containing a *PermitStatus* parameter that allows or blocks the radio from registering. The host application has 30 seconds in which to respond, after which time the base denies registration to the radio. Up to 16 join requests can be pending at any one time. If more than 16 radios are asking to join, the first 16 will be serviced and additional radios will be serviced after the earlier requests are handled. The *RegDenialDelay* parameter controls how often a radio will request registration after it has been denied. If it is anticipated that more than 16 radios will request registration before the application can service the first 16, this parameter should be set to the time it will take the application to service four requests, as this will speed the authentication process by freeing the base from issuing multiple denials to the same remotes.

When the *AuthMode* parameter is set to 3, authentication is locked to the addresses of the radios currently registered with the base. Mode 3 is typically used in conjunction with Mode 0 during a commissioning process. *AuthMode* is set to 0, radios are turned on and allowed to register with the base, and *AuthMode* is then switched to 3 to lock the network membership.

5.3 Serial Port Modes

DN-900 radios can work in two serial port data modes: *transparent* and *packet protocol*. Transparent mode formatting is simply the raw user data. Protocol mode formatting includes a start-of-packet framing character, length byte, addressing, command bytes, etc. Transparent mode operation is especially useful in point-to-point systems that act as cable replacements. In point-to-multipoint, peer-to-peer and tree-routing systems where the base needs to send data specifically to each radio, protocol formatting must be used unless the data being sent includes addressing information that the devices connected to the remotes/routers can use to determine the intended destination of the broadcast data. Protocol formatting is also required for configuration commands and responses. Protocol mode can be used at the base while transparent mode is used at the remotes. The one caution about protocol mode: the length of a protocol mode message cannot exceed the *BaseSlotSize* or *RemoteSlotSize* or the packet will be discarded. Protocol formatting details are covered in Section 6.

To enter protocol mode, the *EnterProtocolMode* command can be sent. When an *ExitProtocolMode* command is sent, the DN-900 will switch to transparent operation. Note that it is possible that part of the *EnterProtocolMode* command will be sent over the air as transparent data during a mode change.

When operating in transparent mode, two configuration parameters control when a DN-900 radio will send the data in its transmit buffer. The *MinPacketLength* parameter sets the minimum number of bytes that must be present in the transmit buffer to trigger a transmission. The *TxTimeout* parameter sets the maximum time data in the transmit buffer will be held before transmitting it, even if the number of data bytes is less than *MinPacketLength*. The default value for *MinPacketLength* parameter is one and the *TxTimeout* parameter is zero, so that any bytes that arrive in the DN-900 transmit buffer will be sent on the next hop. It is useful to set these parameters to values greater than their defaults in point-to-multipoint systems where some or all the remotes are in transparent mode.

5.4 RF Data Communications

At the beginning of each hop the base transmits a beacon, which always includes a synchronizing signal. After synchronization is sent, the base will transmit any user data it has, unless in transparent mode the *MinPacketLength* and/or *TxTimeout* parameters have been set above their default values. The maximum amount of user data the base can transmit per hop is limited by the *BaseSlotSize* parameter. If there is no user data or reception acknowledgements (ACKs) to be sent on a hop, the base will only transmit the synchronization signal in the beacon.

The operation for remotes and routers is similar to the base, but without a synchronizing signal. The *RemoteSlotSize* parameter indicates the maximum number of user data bytes a remote or router can transmit on one hop and is a read-only value. The *RemoteSlotSize* is determined by the *HopDuratio*n and *BaseSlotSize* parameters and the number of registered radios. The *MinPacketLength* and *TxTimeout* parameters operate in a remote in the same manner as in the base.

5.5 RF Transmission Error Control

The DN-900 supports two error control modes: automatic transmission repeats (ARQ), and redundant transmissions for broadcast packets. In both modes, the radio will detect and discard any duplicates of messages it receives so that the host application will only receive one copy of a given message. In the redundant transmission mode, broadcast packets are repeated a fixed number of times based on the value of the *ARQ_AttemptLimit* parameter. In ARQ mode, a packet is sent and an acknowledgement is expected on the next hop. If an acknowledgement is not received, the packet is transmitted again on the next available hop until either an ACK is received or the maximum number of attempts is exhausted. If the ARQ_*AttemptLimit* parameter is set to its maximum value, a packet transmission will be retried without limit until the packet is acknowledged. This is useful in some point-to-point cable replacement applications where it is important that data truly be 100% error-free, even if the destination remote goes out of range temporarily.

5.6 Transmitter Power Management

The DN-900 includes provisions for setting the base transmitter power level and the remote maximum transmit power level with the *TxPower* parameter. DN-900 networks covering a small area can be adjusted to run at lower transmitter power levels, reducing potential interference to other nearby systems. Radios that are located close to their base can be adjusted to run at lower maximum power, further reducing potential interference. Base units transmit at the fixed power level set by the *TxPower* parameter. Remotes and routers automatically adjust their transmitter power to deliver packets to the base at an adequate but not excessive signal level, while not transmitting more power than set by their *TxPower* parameter. Remotes and routers make transmitter power adjustments using the strength of the signals received from the base and the base transmitter power setting, which is periodically transmitted by the base. The automatic transmit power adjustment is enabled by default, but can be disabled if so desired.

5.7 Network Configurations

The DN-900 supports four network configurations: point-to-point, point-to-multipoint, peer-to-peer and tree-routing. In a point-to-point network, one radio is set up as the base and the other radio is set up as a remote. In a point-to-multipoint network, a star topology is used with the radio set up as a base acting as the central communications point and all other radios in the network set up as remotes. In this configuration, each communication takes place between the base and one of the remotes. Peer-to-peer communications between remotes using the base as a relay is also supported. Tree-routing networks can retransmit messages through one or more routers, greatly expanding the area that can be covered by a single DN-900 system.

5.7.1 Point-to-Point Network Operation

Most point-to-point networks act as serial cable replacements and both the base and the remote use transparent mode. Unless the *MinPacketLength* and *TxTimeout* parameters have been set above their default values, the base will send the data in its transmit buffer on each hop, up to a limit controlled by the *BaseSlotSize* parameter. In transparent mode, if the base is buffering more data than can be sent on one hop, the remaining data will be sent on subsequent hops. The base adds the address of the remote, a packet sequence number and error checking bytes to the data when it is transmitted. These additional bytes are not output at the remote in transparent mode. The sequence number is used in acknowledging successful transmissions and in retransmitting corrupted transmissions. A two-byte CRC and a one-byte checksum allow a received transmission to be checked for errors. When a transmission is received by the remote, it will be acknowledged if it checks error free. If no acknowledgment is received, the base will retransmit the same data on the next hop. Note that acknowledgements from remotes are suppressed on broadcast packets from the base.

In point-to-point operation, by default a remote will send the data in its transmit buffer on each hop, up to the limit controlled by its *RemoteSlotSize* parameter. If desired, the *MinPacketLength* and *TxTimeout* parameters can be set above their default values, which configures the remote to wait until the specified amount of data is available or the specified delay has expired before transmitting. In transparent mode, if the remote is buffering more data than can be sent on one hop, it will send the remaining data in subsequent hops. The remote adds its own address, a packet sequence number and error checking bytes to the data when it is transmitted. These additional bytes are not output at the base if the base is in transparent mode. When a transmission is received by the base, it will be acknowledged if it checks error free. If no acknowledgment is received, the remote will retransmit the same data on the next hop.

5.7.2 Point-to-Multipoint Network Operation

In a point-to-multipoint network, the base is usually configured for protocol formatting, unless the applications running on each remote can determine the data's destination from the data itself. Protocol formatting adds addressing and other overhead bytes to the user data. If the addressed remote is using transparent formatting, the source (originator) address and the other overhead bytes are removed. If the remote is using protocol formatting, the source address and the other overhead bytes are output with the user data.

A remote can operate in a point-to-multipoint network using either transparent or protocol formatting, as the base is the destination by default. In transparent operation, a remote DN-900 automatically adds addressing, a packet sequence number and error checking bytes as in a point-to-point network. When the base receives the transmission, it will format the data to its host according to its formatting configuration. A remote running in transparent mode in a point-to-multipoint network can have the *MinPacketLength* and *TxTimeout* parameters set to their default values to reduce latency, or above their default values to reduce the volume of small packet transmissions.

5.7.3 Multipoint Peer-to-Peer Network Operation

After a remote has joined a point-to-multipoint network, it can communicate with another remote through peer-to-peer messaging, where the base acts as an automatic message relay. In protocol mode, if a remote specifies a destination address other than the base address, peer-to-peer messaging is enabled. In transparent mode, the *RmtTransDestAddr* parameter sets the destination address. Changing *RmtTransDestAddr* from the default base address to the address of another remote enables peer-to-peer messaging. The broadcast address (0xFFFFFF) can also be used as a peer-to-peer destination address. In this case, the message will be unicast from the remote to the base (using ARQ) and then broadcast by the base (no ARQ). For peer-to-peer broadcasts, no acknowledgement is sent and no *TxDataReply* packet is reported to the host.

5.7.4 Tree-Routing Operation

A DN-900 tree-routing system consists of a base, remotes and up to 63 routers. A router is basically a remote that has been configured with two operating modes - a base mode for its “child” radios and a remote mode for its “parent” router or the system base. This allows a router to do tree-routing in addition to normal remote functions. Each router can forward messages to/from a total of 126 child radios. A DN-900 tree-routing system can cover a much larger area than other DN-900 networks, with the trade-off that tree-routing increases message transmission latency. Tree-routing systems are well suited to many industrial, commercial and agricultural data acquisition applications. Tree-routing operation is supported by CSMA (mode 1) channel access.

5.8 Full-Duplex Serial Data Communications

From a host application’s perspective, DN-900 serial communications appear full duplex. Both the base host application and each remote/router host application can send and receive serial data at the same time. At the radio level, the radios do not actually transmit at the same time. If they did, the transmissions would collide. As discussed earlier, the base transmits a synchronization beacon at the beginning of each hop, followed by its user data. After the base transmission, the remotes/routers can transmit. Each transmission may contain all or part of a complete message from its host application. From an application’s perspective, the radios are communicating in full duplex since the base can receive data from a remote/router before it completes the transmission of a message to the remote/router and conversely.

5.9 Channel Access

The DN-900 provides three methods of channel access: Polling, CSMA and TDMA, as shown in the table and figure below. The channel access setting is distributed to all radios by the base, so changing it at the base sets the entire network or system. Polling refers to an application sending a command from the base to one or more remote devices and receiving a response from only one remote device at a time. Polling is suitable for both large and small networks where periodic or event reporting by remotes is not required. Carrier Sense Multiple Access (CSMA) is very effective at handling packets with varying amounts of data and/or packets sent at random times from a large number of remotes. Time Division

Multiple Access (TDMA) provides a scheduled time slot for each remote to transmit on each hop. The default DN-900 access mode is TDMA dynamic mode.

Access Mode	Description	Max Number of Remotes
0	Polling	unlimited
1	CSMA	unlimited
2	TDMA dynamic slots	up to 16
3	TDMA fixed slots	up to 16
4	TDMA with PTT	up to 16

Table 5.9.1

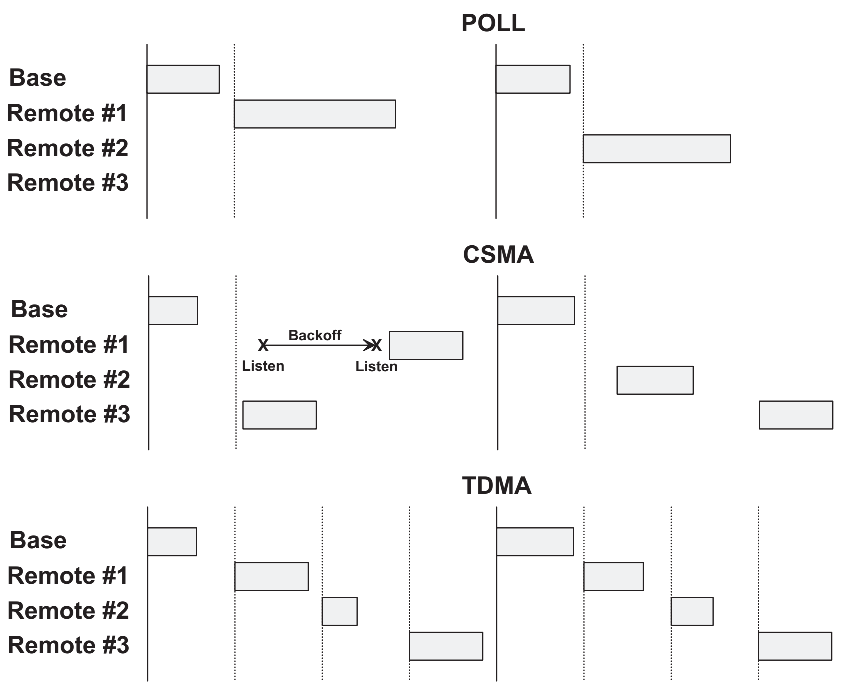

Figure 2.10.1

5.9.1 Polling Mode

Polling channel access is used for point-to-point and point-to-multipoint systems where only one remote will attempt to transmit data at a time, usually in response to a command from the base.

Polling (mode 0) is a special case of CSMA mode 1. The user can set the *BaseSlotSize* and *CSMA_RemtSlotSize* parameters when using this mode. Since only one remote will attempt to transmit at a time, to minimize latency, the *CSMA_Predelay* and *CSMA_Backoff* parameters are not used. Lease renewals are also not used, again to minimize latency. Thus, when the base is operated in protocol mode with Announce messages enabled, only join messages are generated. This mode provides high throughput as there is no contention between remotes and the entire portion of the hop frame following the base transmission is available for a remote to transmit. Applications where more than one remote may attempt to transmit at a time, where event and/or periodic I/O reporting are enabled, and/or tree-routing operation is required should not use this mode.

5.9.2 CSMA Mode

When using CSMA channel access, each remote/router with data to send listens to see if the channel is clear and then transmits. If the channel is not clear, a radio will wait a random period of time and listen again. CSMA works best when a large or variable number of radios transmit infrequent bursts of data. There is no absolute upper limit on the number of radios that can be supported in this mode - it depends on message density. A maximum of 126 radios can be supported if base-managed join/leave tracking is required, or an unlimited number of remotes if base join-leave tracking is not required or will be handled by the host application.

There are two important parameters related to CSMA operation. The *CSMA_Backoff* parameter defines the initial time that a radio will wait when it determines the channel is busy before again checking to see if the channel is clear (back-off interval). If, after finding the channel busy and backing off, the radio finds the channel busy a second time, the amount of time the radio will wait before checking the channel will increase. It will continue to increase each subsequent time the channel is busy until the channel is finally found idle. This is the classic CSMA technique that handles the situation where a number of radios hold data to send at the same time. The *CSMA_Predelay* parameter controls the maximum time that a radio will wait before first listening to see if the channel is clear for a transmission. This parameter is used to make sure that all the remotes do not transmit immediately after the base finishes transmitting.

CSMA (mode 1) provides classical CSMA channel access, and gives the user control over both the *CSMA_Predelay* and *CSMA_Backoff* parameters. This mode is well suited for large numbers of uncoordinated radios, where periodic/event reporting is used, or tree-routing operation is required. In addition to *CSMA_ Predelay* and *CSMA_ MaxBackoff*, the user can set the *BaseSlotSize* and CSMA_ *RemtSlotSize* parameters when using this mode. The following guidelines are suggested for setting *CSMA_Predelay*:

- For lightly loaded CSMA contention networks, decrease *CSMA_Predelay* to 0x03 or less to reduce latency.
- For heavily loaded CSMA contention networks, increase *CSMA_Predelay* to 0x05 or more for better throughput.

As an option, CSMA mode allows the base to directly track remotes entering and leaving the network, for up to 126 remotes. The base is operated in protocol mode and is configured to send Announce messages to its host when a remote joins, and when the remote’s registration lease expires.

While a base in a CSMA network can track a maximum of 126 remotes entering and leaving the network, it can generate join Announce messages for an unlimited number of remotes. This allows the host application to track remotes entering and leaving a CSMA network with more than 126 remotes by creating its own table of MAC addresses and periodically sending a *GetRemoteRegister* command to each remote in the table. Failure to answer a *GetRemoteRegister* command indicates the remote is no longer active in the network.

CSMA modes work well in many applications, but CSMA has some limitations, as summarized below:

- Bandwidth is not guaranteed to any remote.
- Marginal RF links to some remotes can create a relatively high chance of collisions in heavily loaded networks.

5.9.3 TDMA Modes

TDMA modes provide guaranteed bandwidth to some or all of the radios in a network. Radios that register with the base receive several special parameters, including ranging information and a specific channel access time slot assignment. TDMA registrations are always leased and must be renewed every 250 hops. The DN-900 provides three TDMA access modes, as discussed below.

TDMA Dynamic Slots (mode 2) is used for general-purpose TDMA applications where scaling the capacity per slot to the number of active remotes is automatic. Each radio that registers with the base receives an equal time slice. As new remotes join, the size of the TDMA remote slots shrinks accordingly. The number of slots, individual slot start times, and the *RemoteSlotSize* are computed automatically by the DN-900 network in this mode. The user should note that the bandwidth to each remote will change immediately as remotes join or leave the network. When running in protocol mode on a remote, care must be taken not to generate messages too long to be sent in a single hop due to automatic *RemoteSlotSize* reduction.

TDMA Fixed Slots (mode 3) is used for applications that have fixed data throughput requirements, such as isochronous voice or streaming telemetry. The slot start time and the *RemoteSlotSize* are computed automatically by the DN-900 network in this mode. The user must set the number of slots using the *MaxSlots* parameter. The base radio will allocate remote slot sizes as if *MaxSlots* number of radios are linked with the base, even when fewer remotes/routers are actually linked. In this mode, the remote slot sizes are constant.

TDMA with PTT (mode 4) supports remotes with a "push-to-talk" feature, also referred to as "listen-mostly" remotes. This mode uses fixed slot allocations. Remotes can be registered for all but the last slot. The last slot is reserved for the group of remotes that are usually listening, but occasionally need to transmit. In essence, the last slot is a shared channel for this group of remotes. When one of these remotes has data to send it keys its transmitter much like a walkie-talkie, hence the name push-to-talk (PTT). There is no limit to the number of remotes that can listen to the last slot.

The slot start time and the *RemoteSlotSize* are computed automatically by the DN-900 network in mode 4. The user must specify the number of slots using the *MaxSlots* parameter. The last slot is reserved for the PTT remotes. The user must configure PTT remotes individually to select mode 4 operation. The user's application must ensure that only one PTT remote at a time is using the slot. Mode 4 does not support tree-routing operation.

5.10 Transmission Configuration Planning

Because frequency hopping radios change frequency periodically, a single message may be sent in one or more RF transmissions. The length of time the radio stays on a frequency, the hop duration, impacts both latency and throughput. The longer the radio stays on a single frequency, the higher the throughput since the radio is transmitting for a higher percentage of the time, but latency is also higher since radios may have to wait longer to transmit. So latency and throughput trade off against one another. The DN-900 has several configuration parameters that allow latency and throughput to be optimally balanced to the needs of an application.

5.10.1 TDMA Throughput

For TDMA channel access without routers, throughput and latency are controlled by the RF data rate, the serial port baud rate, the *BaseSlotSize*, the *HopDuration*, and the number of remotes. A wide range of throughput and latency combinations can be obtained by adjusting these parameters. The throughput of a radio in a TDMA network is simply:

Number of bytes per hop/Hop Duration

For the base, the number of byes per hop is controlled by the *BaseSlotSize* parameter so the throughput of the base radio is:

BaseSlotSize/HopDuration

Note that if fewer bytes than the *BaseSlotSize* limit are sent to the base radio by its host during the hop duration time in transparent mode, the observed throughput of the base radio will be reduced. If the base is in protocol mode, it will wait until a protocol formatted message is completely received from its host before transmitting it. If the message is not completely received by the time the base transmits, the base will wait until the next hop to transmit the message. The throughput for each remote is:

RemoteSlotSize/HopDuration

In a TDMA mode, the *RemoteSlotSize* is set automatically based on the number of remotes and the *BaseSlotSize*. Note that the base radio always reserves *BaseSlotSize* amount of time in each hop whether or not the base has user data to send.

To help select appropriate parameter values, RFM provides the *DNT Throughput Calculator* utility program (DNTCalc.exe). This program is on the development kit CD. Enabling encryption (security) adds additional bytes to the data to be sent but the *Calculator* has a mode to take this into account.

5.10.2 Polling Throughput

In polling mode, the application sends data from the base to a specific remote, which causes the remote and/or its host to send data back to the application. The network operates like a point-to-point network in this case. In polling, the *HopDuration* should be set just long enough to accommodate a base transmission up to the limit allowed by the *BaseSlotSize* parameter, plus one remote transmission up to the limit allowed by the *CSMA_RemtSlotSize* parameter. These slot sizes and the hop duration set the polling throughput as in TDMA channel access.

The throughput in polling mode is also determined by the amount of time it takes for the remote host device to respond to the poll. For example, consider the situation where a remote host device communicates with the DN-900 at 38.4 kb/s, receives a 16-byte poll command, and takes 1 ms to generate a 32-byte response which it then sends to the DN-900. Sixteen bytes over a UART port is 160 bits using 8,N,1 serial parameters. Sending 160 bits at 38.4 kb/s takes 4.2 ms. Add 1 ms for the host device to process the command and begin sending the 32-byte response. The 32-byte response takes 8.4 ms to send at 38.4 kb/s, for a total turnaround time of 13.6 ms. This amount of time could be added to the base and remote slot times to allow the entire transaction to take place in a single hop. However, except at the 38.4 kb/s over-the-air data rate, this is likely to be much longer than the base and remote slot times. Thus, in practice, lengthening the hop duration to complete the transaction in a single hop doesn’t really affect the throughput. Nevertheless, it is important to note that the throughput for the remote in the example above is substantially less than the remote slot size in bytes divided by the hop duration.

It is not a DN-900 requirement that the complete application message be sent in a single hop, nor that the remote response is returned in a single hop, when in transparent mode. If either transmission occurs over more than one hop, then depending on the length of the data, the RF data rate and the serial port data rate at the receiving end, there may be a gap in the serial data. Some protocols, such as Modbus RTU, use gaps in data to determine packet boundaries.

5.10.3 CSMA Throughput

In CSMA mode, remote radios do not have a fixed throughput, which is why applications requiring guaranteed throughput should use polling or a TDMA mode. The reason that the throughput of a CSMA remote is not fixed is because its ability to transmit at any given time depends on whether another radio is already transmitting. The throughput of a remote is further affected by how many other remotes are waiting for the channel to become clear so they can transmit. This is not a problem when remotes, even a large numbers of remotes, only send data infrequently. The DN-900 includes several configuration parameters that can be used to optimize the performance of a CSMA network.

It is often desirable to limit the amount of data a CSMA remote can send in one transmission. This prevents one remote from hogging network throughput. To accommodate this, the DN-900 provides a CSMA_*RemtSlotSize* parameter that is user configurable. When a remote has transmitted CSMA_ *RemtSlotSize* bytes on a given hop, it will stop transmitting until the next hop. Note that this remote will have to contend for the channel on the next hop, so it is not guaranteed that it will be the first remote to transmit on the next hop or that it will be able to transmit on the next hop at all. To allow multiple remotes a chance to transmit on the same hop, the *HopDuration* parameter must be set long enough to support the *BaseSlotSize*, plus the number of remotes to transmit per hop multiplied by the CSMA_ *RemtSlotSize*, plus the number of remotes to transmit per hop multiplied by the *CSMA_Backoff*. Because of the way CSMA channel access works, this does not guarantee that the desired number of remotes to transmit on a hop will always be able to transmit on a single hop. This is due to the fact that when a remote with data to send finds the channel busy a second time, it waits for a longer period to time before testing the channel again. This time will continue to increase until the remote finds the channel clear. In practice this is unlikely to present a problem, as CSMA networks are used with devices that infrequently have data to send.

The *DNT Throughput Calculator* can be used to determine the *HopDuration,* but it will be necessary to increase the number of slots to a value greater than the number of remotes to transmit on a single hop to account for the backoffs. It is indeterminate how many backoffs may occur during a single hop, which is why the number of remotes that transmit on a given hop cannot be guaranteed. Note that the *CSMA_ Backoff* parameter sets the length of time a remote will wait to recheck the channel when it has detected that the channel is busy. The second time a remote detects that the channel is busy, it will increase the amount of time it waits until it checks again. Every subsequent time it detects a busy channel it will increase the amount of time it will wait in a geometric fashion. This continues until it detects an idle channel. So while a short *CSMA_Backoff* can decrease the time between when one remote transmits and the next remote transmits, it can actually lead to a longer time between remote transmissions than a longer backoff. This can occur when the remote checks the channel multiple times during the transmitting remote's transmission causing the back-off time to be increased.

5.10.4 Latency

The worst case latency for TDMA access (without routers), excluding retries, occurs when the radio receives data just after its turn to transmit. In this instance, it will have to wait the length of time set by the *HopDuration* to begin transmitting the data. If the radio is receiving data over its serial port at a rate higher than its throughput, this will only occur at the beginning of a transmission that spans several hops.

In a polling application, latency is affected by how long the remote and/or its host takes to respond, and when in the hop data is ready to be transmitted. Since a remote can begin transmitting at practically any time during the hop after the base has transmitted, the latency can be less than *HopDuration*. However, the remote transmission may extend over two hops if it starts late in the first hop.

Latency for any given remote in a CSMA network is particularly difficult to characterize. If many remotes have data to send, the latency for the last remote to send will be the length of time it takes all the other remotes to send. The CSMA scheme used in the DN-900 is designed to allow each remote an equal opportunity to transmit, so the concern is not that one remote is locked out, but just how long it will take a number of remotes with data to send to each gain access to the channel and send their data. The more data that needs to be sent, the more time will be consumed checking the channel and backing off when the channel is busy. Again, this is why CSMA networks are best used when there are a large number of nodes that send data infrequently.

The other factor impacting latency is retries. This impact is not unique to frequency hopping radios but is common among all wireless technologies. A radio only transmits data once per hop. It needs to wait until the next hop to see if the transmission was received at the destination. If not, the radio will transmit the data again and wait for the acknowledgement. This can happen up to *ARQAttemptLimit* number of times which is equal to *ARQAttemptLimit* times *HopDuration* amount of time.

5.10.5 Configuration Validation

Although slot durations are automatically calculated by the DN-900, the RF data rate, hop duration, etc., must be coordinated by the user to assure a valid operating configuration based on the following criteria:

1. Regardless of the RF data rate, the maximum DN-900 hop duration is limited to 200 ms. A DN-900 network must be configured accordingly.

2. In protocol mode, the *BaseSlotSize* and *RemoteSlotSize* parameters must be large enough to hold all the data bytes in the largest protocol formatted message being used. Protocol formatted messages must be sent in a single transmission. Any protocol formatted messages too large for the slot size setting will be discarded

3. In TDMA mode 2, the *RemoteSlotSize* will be reduced automatically when a new remote joins the network. This can cause a network to suddenly malfunction if the hop duration is not set to provide an adequately large remote slot allocation when fully loaded with remotes

4. When operating in polling mode 0, the *CSMA_RemtSlotSize* and *HopDuration* parameters are usually set to accommodate the number of data bytes in a maximum size transmission. This configuration provides low latency for polled messages.

5. When operating in CSMA mode 1 with multiple remotes, the *CSMA_RemtSlotSize* and *HopDuration* parameters are usually set to accommodate three times the number of data bytes in one maximum size transmission, to allow time for more than one remote to attempt to transmit during a single hop.

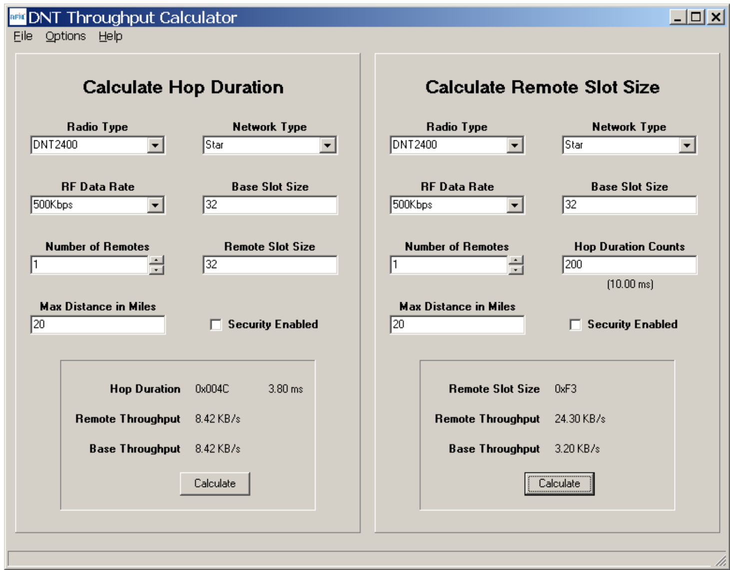

Figure 5.10.5.1

The *DNT Throughput Calculator* utility program is shown in Figure 5.10.5.1. Decimal data is entered by default. Hexadecimal data can also be entered using a 0x prefix, as shown in the *Hop Duration Counts* text box. When using the *DNT Throughput Calculator*, parameter coordination depends on the operating mode of a DN-900 network, as outlined below:

Polling (mode 0) - the user can set and must coordinate the RF data rate, hop duration, base slot size and remote slot size. First, set the *BaseSlotSize* to accommodate the maximum number of data bytes in a base transmission. Next, set the *CSMA_RemtSlotSize* to accommodate the maximum number data bytes in a remote transmission. Use these slot sizes, the RF data rate and the maximum operating range (20 miles is the default) as inputs to the *Calculator* program to determine minimum valid *HopDuration*.

CSMA contention (mode 1) - the same procedure as for polling is used, except that the *CSMA_RemtSlotSize* typically should be set at three times the maximum number of data bytes for point-to-multipoint networks. The default values for CSMA pre-delay and back-off are assumed.

TDMA dynamic (mode 2) - this is the DN-900's default operating mode and the default settings are optimized for point-to-point transparent operation. For other configurations the user must coordinate the RF data rate, hop duration, base slot size and maximum number of remotes. Although the remote slot size and remote slot time allocation are automatically set in mode 2, the user must predetermine these values to assure a valid operating configuration. First, set the *BaseSlotSize* to accommodate the maximum number of data bytes in a base transmission. Next, determine the *RemoteSlotSize* required to accommodate the maximum number of data bytes in a remote transmission. Use these slot sizes, the maximum number of remotes that will be used in the network, the RF data rate and the maximum operating range as inputs to the *Calculator* to determine the minimum valid *HopDuration* time. Note that when there are fewer remotes on the network than the maximum specified, the remotes will automatically be configured with a bigger *RemoteSlotSize* parameter.

TDMA fixed (mode 3) - First, set the *BaseSlotSize* to accommodate the maximum number of data bytes in a transmission. Next, determine the *RemoteSlotSize* required to accommodate the maximum number of data bytes in a remote transmission. Then set the number of remote slots. Use the slot sizes, the number of remotes, the RF data rate and maximum operating range as inputs to the *Calculator* to determine the minimum valid hop duration.

TDMA PTT (mode 4) - use the same procedure as for TDMA fixed mode 3.

The DN-900 base firmware can detect a significant number of invalid configurations and override the *HopDuration* parameter to establish a valid configuration. To take advantage of this feature, configure a DN-900 network in the following order:

1. In all system radios, set the *RF_DataRate* parameter and save it. Then reset all radios to establish the new RF data rate.
2. Set the *BaseSlotSize* and *TDMA_MaxSlots* or *CSMA_RemtSlotSize* as needed. Use the default maximum operating range unless links of more than 20 miles are planned.
3. Set the *HopDuration* parameter and then read it back. If the *HopDuration* parameter readout is different than the value set, the firmware detected an invalid configuration and is overriding it.

5.11 Tree-Routing Systems

As discussed in Section 5.7.4, DN-900 tree-routing systems can cover much larger areas than other DN-900 networks, with the trade-off that tree-routing increases message transmission latency. Tree-routing systems are well suited to many industrial, commercial and agricultural applications. Compared to other DN-900 network configurations, however, tree-routing systems require somewhat more initial planning and commissioning steps, as discussed in this Section.

5.11.1 Example Tree-Routing System

An example tree routing system is shown in Figure 5.11.1.1. In this example, seven sensor locations need to be monitored over a several acre outdoor site. All of the sensor data must be sent back to a central location, the base radio, for collection and analysis. Due to obstructions, remotes R1, R3, R6, and R7 are prevented from communicating directly with the base radio. R1, R3, and R6 have direct communications with either R2 or R5, both of which have direct communications with the base radio. R7 has direct communications with R6 and can use R6 to route messages to and from the base through R5.

Using the tree routing function of the DN-900, all nodes will be able to send and receive data to and from the centrally located base. R2, R5, and R6 which are configured to relay data to and from other nodes in addition to sending their own sensor data are called routing remotes, or routers. Note that there is no hardware or firmware difference between DN-900 base, remote and router nodes - they are simply configured for the particular mode of operation.

As the system is powered up, R2, R4, and R5 will join by registering with the base radio. R2 and R5, once they have registered with the base radio, will start sending out beacons so that R1, R3, and R6 can join the tree-routing system through them. In the case of R6, it will wait until it has joined the system through R5 before sending out the beacons that will let R7 join the system through it.

Example DNT900 Tree Routing System

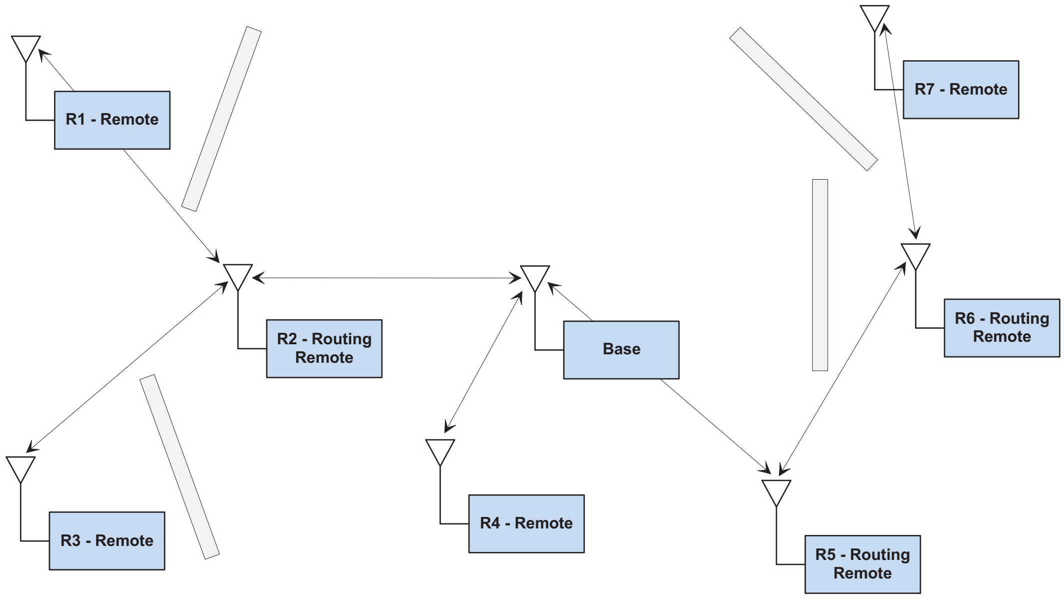

Figure 5.11.1.1

Data sent from the base radio in the central location will be routed down the "tree" to the intended node of the network. Data from the nodes will be routed up the tree to the base or to another node in the system. Note that it is possible to send data from one node to another node rather than sending it to the base.

A tree-routing system can operate in a polling mode where an application sends data requests to each of the nodes as needed, or it can operate where devices attached to the node radios send data whenever they have it to send. Additionally, the auto-report function of the DN-900 radio can be used to send data through the tree on a timer or interrupt basis.

To set up the example system, all of the DN-900s, including the base, must be configured with the same tree routing ID, and have tree routing option enabled. In addition, R2, R5, and R6 must be assigned individual *BaseModeNetID* parameters, and then configured for router operation. The network IDs and network addresses will be automatically assigned as the system forms. Figure 5.11.1.2 shows one way that the network IDs and system addresses could be assigned.

Note that the routing nodes, R2, R5, and R6 have two network-related IDs - a *ParentNetworkID* and a *BaseModeNetID*. The *ParentNetworkID* is used by a router to join the tree-routing system, and the *BaseModeNetID* is the ID the router uses to let other nodes join the system through it.

While the tree-routing system can form automatically, it is also possible to do additional node configuration to control how the system forms. The following sections provide details of all tree routing related configuration commands plus details of the addressing used in a tree routing system.

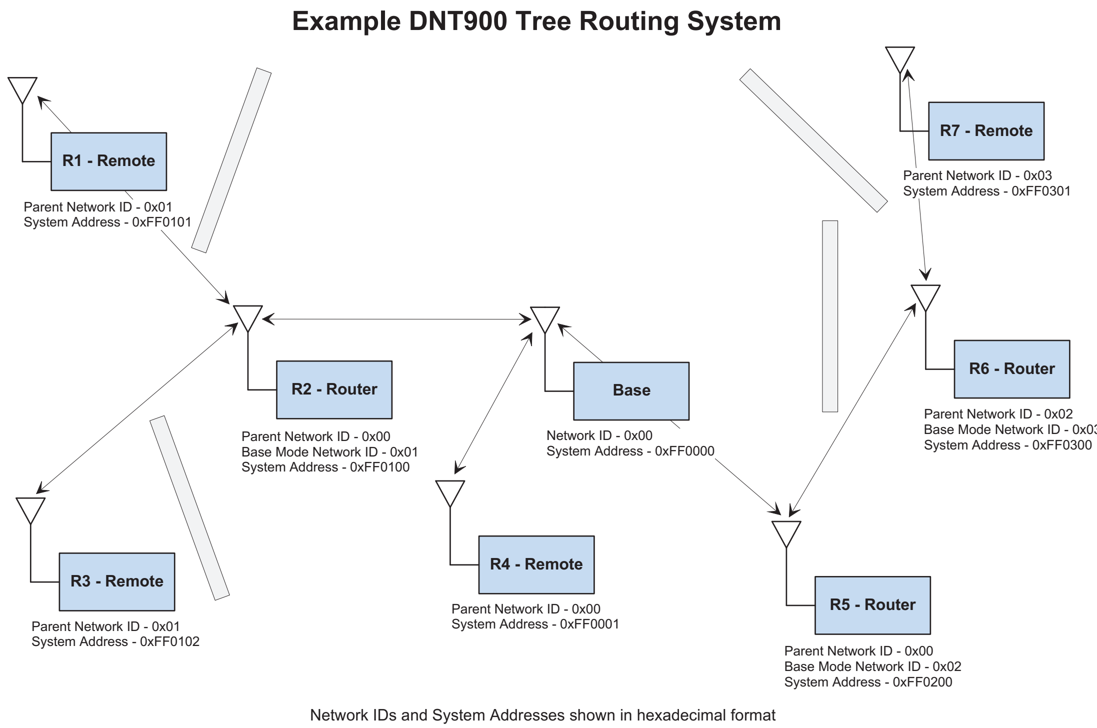

Figure 5.11.1.2

5.11.2 Tree-Routing System Networks

A DN-900 tree-routing system consists of one base and up to 63 routers, where the base and each router can forward messages to/from a total of 126 child radios (leases enabled)[1]. A child radio can be either a remote or router, within the system limitation of 63 routers total.

Within a DN-900 tree-routing system, a network refers to a group of radios communicating with a router or base acting as a parent, and all of the child radios that are linked to this parent. Hop-by-hop, a router alternates from being a child of its parent network on one hopping pattern to being a parent of its own network on a different hopping pattern.

The base maintains a routing table that describes the organization of all routers in its system. This table is used by the base and the routers to determine which direction, up to the base, or down to its children, to route a packet. The base updates the routing table using information in the heartbeat packets it receives from the routers in its system. The base periodically broadcasts this routing table to inform all system radios of the current system configuration. All heartbeat packets received by the base are also output to its host (PC). The default channel access for tree-routing systems is CSMA (mode 1).

1. Tree-routing systems can run without leases enabled to remove the 126 child limit on the base and routers in some circumstances. However, this takes special system planning. Contact RFM technical support for details.

5.11.3 Tree-Routing System Addressing

Except for tree-routing systems, DN-900 remotes are addressed from the base using their three-byte hardware *MAC addresses*. In turn, remotes address the base using the address 0x000000, rather than the base radio's hardware MAC address. In tree-routing systems, however, radios are addressed using *system addresses* rather than their hardware MAC addresses. Much of the planning and commissioning activities for a DN-900 system involve configuring system addresses.

Like MAC addresses, tree-routing system addresses contain three bytes. However, the most significant byte of a system address is currently unused and can be assigned any value (typically 0xFF). The middle byte of a system address is the network ID or *NwkID* of a base or router. The *NwkID* is always 0x00 for the base, and will have a value in the range of 0x01 to 0x3F for a router. The least significant byte of the system address is called the network address or *NwkAddr*. The *NwkAddr* is always 0x00 for the base and all the routers in a system. For a remote, the *NwkAddr* will have a value in the range of 0x01 to 0x7E, so the system address of a remote will contain the *NwkID* of its parent base or router, plus its own *NwkAddr*.

Several parameters are involved in the formation of a DN-900 tree-routing system. All radios that will become part of a tree-routing system must set the *TreeRouteEn* parameter to 0x01. Further, all radios must be loaded with the same tree-routing system ID parameter, referred to as the *TreeRouteSysID*. This parameter allows systems to physically overlap without ambiguity as to which system a radio should join.

When a DN-900 radio is configured as the base, it automatically assumes a *NwkID* of 0x00 and *NwkAddr* of 0x00. When a radio is configured as a router, it automatically assumes a *NwkAddr* of 0x00. The router's *NwkID* byte is held in its *BaseModeNetID* parameter, and will have value in the range of 0x01 to 0x3F. The *BaseModeNetID* parameter must be manually set in all routers. As discussed below, when a router's addressing is totally manually configured, the remote-mode *NwkAddr* (network address) is loaded from a router's *StaticNetAddress* parameter, otherwise the default value 0xFF of the parameter should be preserved to allow dynamic assignment by the router's parent.

The *InitialParentNwkID* parameter controls which parent a child router or remote can join. Setting this value to 0x00 forces a router or a remote to join only with the base. Setting this parameter to the *NwkID* of a parent router forces a child router or a remote to join only this parent's network. Setting this parameter to 0xFF in a router or remote allows them to join with any parent.

The *StaticNetAddr* parameter controls the assignment of the *NwkAddr* byte in a remote's system address. A remote's *NwkAddr* can be manually assigned by setting the *StaticNetAddr* to a value between 0x01 and 0x7E. Setting the *StaticNetAddr* parameter to 0xFF allows the remote's parent to dynamically assign a *NwkAddr*.

As discussed above, an example DN-900 tree-routing system is shown in Figure 5.12.1.2. The example system includes remote R4 which is directly linked to the base, routers R2 and R5 which are directly linked to the base, remotes R1 and R3 which are linked to router R2, remote R7 which is linked to router R6, which in turn is linked to router R5. The parent network ID, the system address and for routes the base-mode network ID are shown in hexadecimal format.

Note that when dynamic address assignment is used, the system addresses of some of the radios in the system will not be immediately apparent. A radio's system address can be obtained by broadcasting a *Discover* command from the base which contains the radio's hardware MAC address. The radio will send a *DiscoverReply* with its system address. After joining one of the system networks, all routers and remotes periodically transmit heartbeat status messages that contain their MAC address, network address, network ID and other information. Note that the address of any radio can be constructed as follows:

0xFF + NwkID + NwkAddr.

5.11.4 Tree-Routing System Implementation Options

There are three ways to assign parent network IDs and system addresses to the routers and remotes in a tree-routing system - dynamic router parent and remote system address assignment, manual router parent assignment with dynamic remote address assignment, and manual router parent and remote address assignment.

Dynamic router parent and remote address assignment is the preferred method for most systems that contain just a few routers. This assignment method provides several advantages. The router parent and remote system addresses do not have to be pre-assigned, reducing initial system planning details. In case of a parent failure, child devices will automatically attempt to join another parent. Once the system becomes organized, heartbeat status messages and/or Discover commands can be used to log the system addresses against the MAC addresses of each router and remote in the system.

Manual router parent assignment with dynamic remote address assignment is the preferred method for most systems with a large number of routers. Manual router parent assignment avoids the possibility of the system creating a long chain of parent router-child router links which would introduce unnecessary message latency. However, manual router assignment precludes a child router from attempting to link with another parent in case its parent router fails. The parent address of each router is known before the system becomes organized, and heartbeat status messages and/or *Discover* commands can be used to log the system addresses against the MAC addresses of each remote in the system.

Manual router parent and remote address assignment allows all radios addressing to be preplanned and preset before a system is installed. Manual system addressing precludes child radios from attempting to re-link to the system by joining another parent if the assigned parent fails, but simplifies application code development by removing the need to dynamically update a database that matches system addresses to MAC addresses for each router and remote. The task of manually assigning system addresses to all routers and remotes in a tree-routing system can be somewhat tedious. Contact RFM's module technical support group for the latest support tools for manual address assignment. Table 5.11.4.1 summarizes radio parameter settings for each assignment method.

Dynamic Router Parent Assignment and Dynamic Remote System Address Assignment			
Radio	InitialParentNwkID	StaticNetAddress	BaseModeNetID
Router	0xFF (join any parent)	0xFF (assigned by parent)	0x01 - 0x3F (router NwkID)
Remote	0xFF (join any parent)	0xFF(assigned by parent)	0xFF (not used by remote)
Manual Router Parent Address Assignment with Dynamic Remote System Address Assignment			
Radio	InitialParentNwkID	StaticNetAddress	BaseModeNetID
Router	0x00 - 0x3F (specifies parent)	0xFF (assigned by parent)	0x01 - 0x3F (router NwkID)
Remote	0xFF (join any parent)	0xFF(assigned by parent)	0xFF (not used by remote)
Manual Router Parent and Remote System Address Assignment			
Radio	InitialParentNwkID	StaticNetAddress	BaseModeNetID
Router	0x00 - 0x3F (specifies parent)	0x01 - 0x7E (sent to parent)	0x01 - 0x3F (router NwkID)
Remote	0x00 - 0x3F (specifies parent)	0x01 - 0x07 (sent to parent)	0xFF (not used by remote)

Table 5.11.4.1

Tree-routing networks can support peer-to-peer communications. However, the value of the *P2PReply-Timeout* parameter is interpreted differently in a tree-routing system compared to a point-to-multipoint network. In a point-to-multipoint network, the *P2PReplyTimeout* parameter is in units of hops. In tree-routing systems, this parameter is in units of hop pairs, due to the system routers alternating between remote mode and base mode hop-by-hop. Referring to Figure 5.11.1.2, the minimum useable value for peer-to-peer communications between Remote 1 and Remote 7 is determined as follows:

Route Segment	Required Hops
Remote R1 to Router R2	1
Router R2 to Base	1
Base Turn Around	1
Base to Router R5	1
Router R5 to Router R6	1
Router R6 to Remote R7	1
Remote R7 Reply Turn Around	1
Remote R7 to Router R6	1
Router R6 to Router R5	1
Router R5 to Base	1
Base Turn Around	1
Base to Router R2	1
Router R2 to Remote R1	1
Total	13

Table 5.11.4.2

The minimum number of hops required is 13, so the minimum *P2PReplyTimeout* parameter value would be 7 hop pairs. This minimum value provides no tolerance for transmission retries. Selecting a value 50% to 100% larger is recommended, in the range of 11 to 14 hop pairs.

5.12 Serial Port Operation

DN-900 networks are often used for wireless communication of serial data. The DN-900 supports serial baud rates from 1.2 to 460.8 kb/s. Listed in Table 5.12.1 below are the supported data rates and their related byte data rates and byte transmission times for an 8N1 serial port configuration:

Baud Rate kb/s	Byte Data Rate kB/s	Byte Transmission Time ms
1.2	0.12	8.3333
2.4	0.24	4.1667
4.8	0.48	2.0833
9.6	0.96	1.0417
19.2	1.92	0.5208
28.8	2.88	0.3472
38.4	3.84	0.2604
57.6	5.76	0.1736
76.8	7.68	0.1302
115.2	11.52	0.0868
230.4	23.04	0.0434
460.8	46.08	0.0217

Table 5.12.1

To support continuous full-duplex serial port data flow, an RF data rate higher than the serial port baud rate is required for FHSS. Radios transmissions are half duplex, and there are overheads related to hopping frequencies, assembling packets from the serial port data stream, transmitting them, sending ACK's to confirm error-free reception, and occasional transmission retries when errors occur.

For example, consider a TDMA mode 2 system with one remote operating up to 20 miles away at 500 kb/s, with the *BaseSlotSize* parameter set to 64 bytes and the *RemoteSlotSize* parameter set to 64 bytes. As shown in Figure 5.12.1, the hop duration from the *DNT Throughput Calculator* program for this configuration is 4.85 ms:

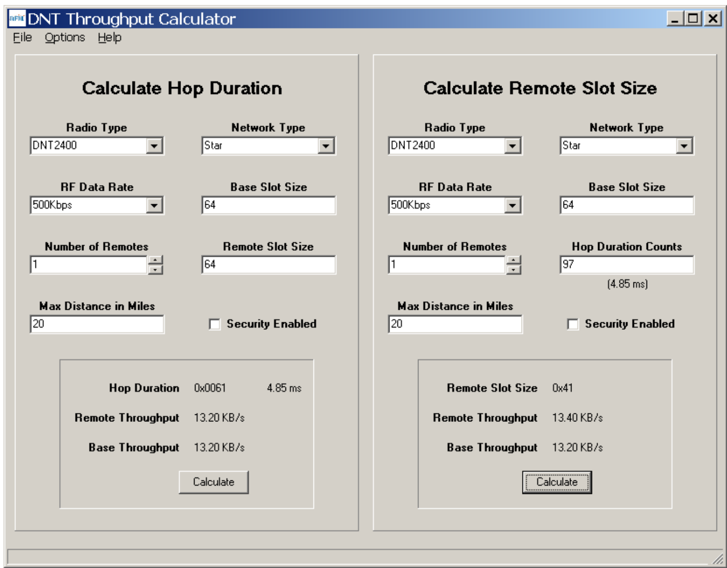

Figure 5.12.1

The average full-duplex serial port byte rate that can be supported under error free conditions is:

64 Bytes/4.85 ms = 13.2 kB/s, or 132 kb/s for 8N1

Continuous full-duplex serial port data streams at a baud rate of 115.2 kb/s can be supported by this configuration, provided only occasional RF transmission errors occur. Plan on an average serial port data flow of 90% of the calculated error-free capacity for general-purpose applications.

5.13 Encryption

The DN-900 supports 128-bit AES encryption of data and configuration packets. Encryption is enabled by setting the *EncryptionKey* register to a value other than a string of NULL (0x00) bytes. A remote without encryption enabled cannot link to an encrypted base, and an encrypted remote will not attempt to link to an unencrypted base. A remote's encryption key must match that of the base before it can link. The *EncryptionKey* register can be set over the air so it can be changed periodically if desired. Once an encryption key has been entered, it can be changed but it cannot be read back.

6.0 Protocol Messages

6.1 Protocol Message Formats

The DN-900 is configured and controlled through a series of protocol mode messages. All protocol mode messages have a common header format:

0	1	2	3 …
SOP	Length	PktType	variable number of arguments …

Figure 6.1.1

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 7-6 Reserved for future use
Bit 5 Event - set to indicate this message is an event
Bit 4 Reply - set to indicates this message is a reply
Bits 3-0 Type - indicates the message type/command

As indicated, the lower 4 bits (3-0) specify a message type. Bit 4 is a modifier indicating that the message is a command or a reply. A reply message has the original command type in bits 3:0, with bit 4 set to one.

Arguments vary in size and number depending on the type of message and whether it is a message sent from the host or is a reply from the radio; see Table 6.1.2.1 below. Messages that are generated on the serial interface by the user are referred to as host messages. Messages that are generated by the radio are referred to as reply messages. For many message types, there is a reply message that corresponds to a host message. For example, when the host sends a *TxData* message, the radio will reply to indicate the status of the transmission, whether it succeeded or failed. Some message types are host-only or reply-only; refer to Table 6.1.2.1 for specifics.

6.1.1 Message Types

Each message generally has two forms, a command from the host and a reply from the radio. Depending on the direction, they have different arguments as shown in Table 6.1.2.1. Event messages from the radio 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.

6.1.2 Message Format Details

Com-mand	Reply	Event	Description	Direction	Arguments
0x00	-	-	EnterProtocolMode	*from Host*	DNTCFG (ASCII characters)
-	0x10	-	EnterProtocolModeReply	*from Radio*	*none*
0x01	-	-	ExitProtocolMode	*from Host*	*none*
-	0x11	-	ExitProtocolModeReply	*from Radio*	*none*
0x02	-	-	SoftwareReset	*from Host*	BootSelect
-	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, Addr, RSSI
0x06	-	-	Discover	*from Host*	MacAddr
-	0x16	-	DiscoverRely	*from Radio*	Status, MacAddr, Addr
-	-	0x26	RxData	*from Radio*	Addr, RSSI, Data
-	-	0x27	Announce	*from Radio*	AnnStatus, additional fields
-	-	0x28	RxEvent	*from Radio*	Addr, RSSI, Reg, Bank, Span,Val
0x0A	-	-	GetRemoteRegister	*from Host*	Addr, Reg, Bank, Span
-	0x1A	-	GetRemoteRegisterReply	*from Radio*	If command successful: TxStatus, Addr, RSSI, Reg, Bank, Span, Val If command failed: TxStatus, Addr
0x0B	-	-	SetRemoteRegister	*from Host*	Addr, Reg, Bank, Span, Val
-	0x1B	-	SetRemoteRegisterReply	*from Radio*	TxStatus, Addr, RSSI
-	-	0x2C	JoinRequest	*from Radio*	MacAddr, Addr, DeviceMode, SleepMode
0x0C	-	-	JoinReply	*from Host*	MacAddr, PermitStatus
0x0D	-	-	RemoteLeave	*from Host*	MacAddr, BackOffTime

Table 6.1.2.1

<u>*Arguments:*</u>

Reg = Register location *(1 byte)*
Bank = Register bank, which provides logical isolation 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/write to/from a register *(see table 6.1.2.1 for size and acceptable range).*
Data = User data *(must fit within the slot size allocation)*
MacAddr = MAC address of sender, for a reply or an event, or the recipient for a command *(3 bytes)*
Addr = Same as MAC address *(3 bytes).* When specifying a destination address in a tree-routing system, a system address is used according to the following format (little-Endian byte order):

for routers - 0x00 0xNN 0xFF, where NN is the NwkID of the router.
for remotes - 0xMM 0xNN 0xFF, where NN is the NwkID of the parent router and MM is the network address of the remote.

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

0x00 = Acknowledgement received
0x01 = No acknowledgement received
0x02 = Not linked (remote)
0x03 = No ACK due to recipient holding for flow control

RSSI = 2’s complement value in dBm, with a range of -128 (0x80) to +125 (0x7D) dBm *(1 byte);* large positive RSSI values will not occur under ordinary circumstances. RSSI values 126 (0x7E) and 127 (0x7F) have special meaning:

0x7F = No RSSI measured because no ACK was received
0x7E = No RSSI because packet was routed

NwkID = Network identifier of network joined *(1 byte)*
BaseMacAddr = MAC address of base that the remote joined *(3 bytes).*
AnnStatus = Status announcement *(1 byte).* Additional fields are also reported depending on the status code:

Status code	*Additional fields*
0xA0 = Radio has completed startup initialization	none
0xA2 = Base: a child has joined the network	MacAddr, Reserved, Range
0xA3 = Remote: joined a network, ready for data	NwkID, BaseMacAddr, Range
0xA4 = Remote: exited network (base is out of range)	NwkID
0xA5 = Remote: the base has rebooted	none
0xA7 = Base: remote has left the network.	MacAddr
0xA8 = Base: heartbeat received from router or remote	MacAddr, NwkAddr, NwkID, ParentNwkID, BeaconRSSI, AvgTxAtt-mpts, ParentRSSI, Range
0xA9 = Base: router heartbeat timeout	NwkID

New status code field definitions:

ParentNwkID = Network ID of parent *(1 byte)*
BeaconRSSI = Average power of received beacons *(1 byte)*
ParentRSSI = Average power of received heartbeat as reported by parent *(1 byte)*
AvgTxAttempts = Average number of upstream transmit attempts per packet times 4 *(1 byte)*

Status codes for error conditions	*Additional fields*
E0 = Protocol error - invalid message 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
E8 = UART receive buffer overflow	none
E9 = UART receive overrun	none
EA = UART framing error	none
EE = Hardware error	none

Range = Range measurement of joining radio *(1 byte)*. Each count equals 0.29 miles.
BootSelect = Code indicating whether to do a normal reset or a reset to the bootloader *(1 byte)*
(0 = normal reset, 1 = reset to bootloader)
PermitStatus = Permission for new node to join, 0x00 = denied, 0x01 = permitted *(1 byte)*
BackoffTime = Time that a node will avoid trying to join a network, in seconds *(2 bytes)*
(0xFFFF = back off until reset or power cycled)

6.1.3 Protocol Mode Data Message Example

In this example, ASCII text *Hello World* is sent from the base to a remote using a *TxData* command. The MAC address of the remote is 0x000102. The protocol formatting for the host message is:

0xFB **0x0F** 0x05 **0x02 0x01 0x00** 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64

There are 15 bytes following the length byte, so the length byte is set to 0x0F. Note that the 0x000102 MAC address is entered in Little-Endian byte order 0x02 0x01 0x00.

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

0xFB 0x06 0x15 **0x00** 0x02 0x01 0x00 **0xC4**

The 0x00 *TxStatus* byte value indicates the ACK reception from the remote. The *RSSI* value of the received ACK is 0xC4 (-60 dBm).

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

0xFB 0x10 **0x26** 0x00 0x00 0x00 **0xC4** 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64

The message is output as a 0x26 event. The address field contains the base (originator) address. Note that the RSSI value 0xC4 is inserted between the base MAC address and the *Hello World* user data.

6.1.4 Protocol Mode Tree-Routing MAC Address Discovery Example

In this example, a *Discover* command is broadcast in a tree-routing system to obtain the system address of a remote with the MAC address 0x000102.

The protocol formatting for the host message is:

0xFB 0x04 0x06 0x02 0x01 0x00

There are 4 bytes following the length byte, so the length byte is set to 0x04. Note that the 0x000102 MAC address is entered in Little-Endian byte order 0x02 0x01 0x00.
The target remote will issue a *DiscoverReply* command as follows:

0xFB 0x08 0x16 **0x00** 0x02 0x01 0x00 **0x01 0x01 0xFF**

The 0x00 *Status* byte value indicates a response from the remote. The tree-routing system address of the target remote is received in Little-Endian byte order, so the address is 0xFF0101. Referring to Figure 5.11.1.2, this is the system address of Remote R1.

6.2 Configuration Registers

The configuration registers supported by the DN-900 are described below. Registers are sorted into banks according to similar functions. Default register values are in **bold**. All changes to parameters are temporary and will not persist through power cycling until 0x01 has been written to the *MemorySave* parameter in Bank FF. Writing 0x00 to the *MemorySave* parameter will reload the factory default values which also require writing 0x01 to the *MemorySave* location. Also note that some parameter changes do not take effect until they are saved and the module is reset or power cycled. Modules can be reset by writing 0x00 to the *UcReset* parameter in Bank FF. Certain DNT900 radio registers do not apply to DN-900 modem operation. These registers are grayed out in the tables below.

6.2.1 Bank 0 - Transceiver Setup

Bank	Loc'n	Name	R/W	Size in bytes	Range	Default, Options
0x00	0x00	DeviceMode	R/W	1	0..3	**0 = Remote**, 1 = Base, 2 = PTT Remote, 3 = Router
0x00	0x01	RF_DataRate	R/W	1	0..4	**0 = 500**, 1 = 200, 2 = 115.2, 3 = 38.4 kb/s, 0XFF = auto
0x00	0x02	HopDuration	R/W	2	80..4000	**10 ms (0x00C8)**
0x00	0x04	InitialParentNwkID	R/W	1	0..63, 255	**0xFF** = join any network
0x00	0x05	SecurityKey	R/W	16	$0..2^{128}-1$	**all null bytes (0x00) = security disabled**
0x00	0x15	SleepMode	R/W	1	0..2	**0 = off**, 1 = on, 2 = on after reset
0x00	0x16	WakeResponseTime	R/W	1	0..255	**0x05 = 50 ms**, 0 to disable
0x00	0x17	WakeLinkTimeout	R/W	1	0..255	**5 s**
0x00	0x18	TxPower	R/W	1	0..5	**0 = 1 mW**; 1 = 10 mW, 2 = 63 mW 3 = 250 mW, 4 = 500 mW, 5 = 1000 mW
0x00	0x19	ExtSyncEnable	R/W	1	0..1	**0 = off**; 1 = enable
0x00	0x1A	DiversityMode	R/W	1	0..2	**0 = 0 V**, 1 = 3.3 V, 2 = toggle
0x00	0x1B	Reserved		1		
0x00	0x1C	UserTag	R/W	16		**"DNT900"**
0x00	0x2C	RegDenialDelay	R/W	2		**10 s**
0x00	0x2E	RmtTransDestAddr	R/W	3		**0x000000 (Base)**
0x00	0x34	TreeRoutingEn	R/W	1	0..1	**0x00 = off**, 1 = enable
0x00	0x35	BaseModeNetID	R/W	1	1..63	**0xFF** (must change on routers)
0x00	0x36	StaticNetAddr	R/W	1	1..126, 255	**0xFF = dynamic assignment**
0x00	0x37	HeartbeatIntrvl	R/W	2	1..65535	**0x14 (20 seconds)**
0x00	0x39	TreeRoutingSysID	R/W	1	0..255	**0x00**
0x00	0x3A	enableRtAcks	R/W	1	0..1	**0x00 (suppress remote peer-to-peer ACKs)**

Note: These settings are individual to each module.

DeviceMode - selects the operating mode for the radio: remote, base, PTT remote (listen mostly remote) or tree-routing router. Note that changing this setting does not take effect immediately. It must be followed by a *MemorySave* command (See Section 4.2.9) and then a hardware reset.

RF_DataRate - this sets the over-the-air data rate. DNT900's with different RF data rates cannot inter-communicate. The following codes are defined:

0x00 = 500 kb/s (default)
0x01 = 200 kb/s
0x02 = 115.2 kb/s
0x03 = 38.4 kb/s
0xff = auto

The *auto* setting will cause a remote to try all four over-the-air rates when scanning for a network to join. Setting the *RF_DataRate* to a fixed value on the remotes will allow a network to link much faster than using the *auto* setting. However, if the base *RF_DataRate* is changed when the remotes are set to a fixed rate, the network will not link. Note that changing this setting does not take effect immediately. It must be followed by a *MemorySave* command and then a hardware reset.

HopDuration - this sets the duration of the hop frame. The duration is set as a 12-bit value, 0.05 ms/count. Changing the hop duration must be followed by a *MemorySave* command to allow the change to persist through a reset or power cycle. A *HopDuration* change takes effect immediately. Remotes will re-link following a *HopDuration* parameter change.

InitialParentNwkID - this parameter selects the network ID that a base will use to start a network, or a remote will be allowed to join. A value of 0xFF instructs a remote to operate in 'promiscuous mode' and join any network it finds (if set for a base, this will select the default network ID of 0x00.)

In a tree-routing system, this parameter controls which parent a child router or remote can join. Setting this value to 0x00 forces a router or a remote to join with only the base. Setting this parameter to the *NwkID* of a parent router forces a child router or a remote to join only this parent's network. Setting this parameter to 0xFF in a router or remote allows them to join with any parent. In a tree-routing system, the *InitialParentNwkID* parameter also determines the hopping pattern for the parent and children of each network. When a network ID equals or exceeds the number of unique hopping patterns, the hopping pattern selection will "wrap around" to the network ID modulo of the number of hopping patterns. To prevent interference between networks in a tree-routing system, networks with the same hopping pattern must be physically separated by enough distance that they cannot receive transmissions from each other.

It is often convenient to set the *InitialParentNwkID* parameter value on remotes to 0xFF to allow them to connect to any parent device. However, setting the *InitialParentNwkID* parameter value to 0xFF on a system with a large number of routers can be problematical. This could result in very long routing paths, slowing communications, and possibly causing reply delays to exceed the *P2PReplyTimeout* parameter setting. For this reason, it is often best to design a tree-routing system using a large number of routers so that each router has an explicit *InitialParentNwkID*.

SecurityKey - this sets the 128-bit AES encryption key to be used. To protect the key, this is a write-only parameter for the user (always reads back as 0x2A). Refer to the Section 2.16 for further information.

TxPower - Sets the transmit power level:

0x00 = 0 dBm or 1 mW (default)
0x01 = 10 dBm or 10 mW
0x02 = 18 dBm or 63 mW
0x03 = 24 dBm or 250 mW
0x04 = 27 dBm or 500 mW
0x05 = 30 dBm or 1000 mW (1 W)

Setting bit 4 to a 1 overrides automatic transmit power control and locks the transmit power at the set level, except when the data rate is set to 500 kb/s. At this data rate the firmware always limits the transmit power to a maximum of 18 dBm (63 mW) to comply with FCC regulations.

UserTag - this is a user definable field intended for use as a location description or other identifying tag such as a "friendly name".

RegDenialDelay - when a remote has been removed from a network through a *RemoteLeave*, the *RegDenialDelay* parameter sets the length of time the remote will wait before considering that network a candidate for joining again. Units are in seconds; the default is 10 s.

RmtTransDestAddr - this parameter sets the destination address that a remote will send packets to when configured to use transparent mode. The default destination is the base (0x000000). If this field is set to another remote's MAC address or to the broadcast address (0xFFFFFF), a peer-to-peer packet will be sent. Note that peer-to-peer packets have higher latency than direct packets between base and remote. This setting has no effect on the base.

TreeRoutingEn - this parameter enables tree-routing system operation. Tree-routing operation is disabled by default.

BaseModeNetID - this parameter holds the base-mode network ID used by a router. This parameter *must be set to a unique value* while the node is in remote mode. The node can then be configured as a router. The base-mode network ID of 0 is reserved for the system base. To keep the base routing table transmissions short, low *BaseModeNetID* values should be used when possible. The base routing table transmissions leave off all unused entries corresponding to higher *BaseModeNetID*s. For instance, if the largest *BaseModeNetID* being used is 63, the payload in the routing table packet will be 64 bytes long. In contrast, if the largest *BaseModeNetID* being used is five, then the payload in the routing table packet will be only 6 bytes long.

StaticNetAddr - this parameter holds the lower byte of the system address of a remote. Assigning a value other than 0xFF provides a fixed (static) address. When this parameter is set to 0xFF, the router assigns the lower byte of the system address to a remote dynamically. When using fixed network addressing, any device that can connect to a parent that has any children with fixed network addresses must also have a unique fixed network address, *including child routers*.

HeartbeatIntrvl - this parameter sets the interval that heartbeat status messages are sent from each system radio. Status messages include the node's parent's *BaseModeNetID*, its own routing address, and miscellaneous performance data. The default value for this parameter is 20 seconds. If the heartbeat interval is set to 0x0000, remote heartbeats are disabled. Since router heartbeats are needed to maintain the system routing table, setting the *HeartbeatIntrvl* value on a router to 0x0000 will cause heartbeats to be sent at the default 20 second rate. Setting the *HeartbeatIntrvl* parameter to 0xFFFF will suppress heartbeats except during registration or when an error is detected in the routing table.

The base maintains a 2-bit counter for each router in the system. The counter is decremented at the base's heartbeat interval. If the base does not receive a heartbeat packet from a router for two to three heartbeat intervals, that router and all of its child routers (and their child routers, etc.) are deleted from the routing table. It is possible to set the base's heartbeat interval to a value greater than the heartbeat interval for all the other devices in the network, to prevent premature router timeouts due to network congestion causing heartbeats from routers to be delayed or lost.

The base treats router heartbeat packets differently than remote heartbeat packets. Heartbeat packets from remotes are not ACKed, while ACKs are sent to a router originating a heartbeat packets to indicate reception. This prevents additional heartbeat transmissions by a router until the next heartbeat interval. If a router does not receive a reply to a heartbeat packet within the configured *P2PReplyTimeout* interval, it will persistently re-send the heartbeat packet until its is ACK'ed.

TreeRoutingSysID - this parameter holds the system ID for a tree-routing system.

enableRtAcks - this parameter controls remote ACK replies for peer-to-peer data packets. The default configuration for this parameter is 0, which suppresses remote peer-to-peer ACKs. Setting this parameter to 1 enables peer-to-peer ACKs. This parameter applies to both point-to-multipoint and tree-routing peer-to-peer communications.

6.2.2 Bank 1 - System Settings

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***	***Range***	***Default; Options***
0x01	0x00	FrequencyBand	*R/W*	1	0, 1, FF	**0x00 = North America, South America**; 0x01 = Australia, 0xFF = auto
0x01	0x01	AccessMode	*R/W*	1	0..4	**2 = TDMA Dynamic Slots**
0x01	0x02	BaseSlotSize	*R/W*	1	6..233	**50 bytes**
0x01	0x03	LeasePeriod	*R/W*	1	0..250	**5 s** (0 to disable)
0x01	0x04	ARQ_Mode	*R/W*	1	0..3	**1 = redundant broadcast with pass *ARQAttemptLimit* to remotes enabled**
0x01	0x05	ARQ_AttemptLimit	*R/W*	1	0..63	**8 attempts**
0x01	0x06	MaxSlots	*R/W*	1	0..15	**4 slots**
0x01	0x07	CSMA_Predelay	*R/W*	1	0..255	**0x03**
0x01	0x08	CSMA_Backoff	*R/W*	1	0..255	**0x0A**
0x01	0x09	MaxPropDelay	*R/W*	1	0..255	**0x45** (20 mi, 32.18 km)
0x01	0x0A	LinkDropThreshold	*R/W*	1	0..255	**0x0C**
0x01	0x0B	CSMA_RemtSlotSize	*R/W*	1	1..255	**64**
0x01	0x0C	CSMA_BusyThreshold	*R/W*	1	1..255	**20**
0x01	0x0D	RangingInterval	*R/W*	1	0..255	**0** (disable)
0x01	0x0E	AuthMode	R/W	1	0..3	**0** = authentication disabled
0x01	0x0F	P2PReplyTimeout	R/W	1	0..255	**16** hops

Bank 1 holds configuration parameters to be input to the base only. The base passes these parameters to the remotes as needed. The exception is the *ARQ_AttemptLimit* parameter. If *ARQ_Mode* bit 1 is set to 1 at the base, the *ARQ_AttemptLimit* parameter can be set in the remotes and used.

FrequencyBand - this parameter sets the range of frequencies over which the radio will operate. Two settings are defined: Band 0 - North and South America (902 to 928 MHz), and Band 1 - Australia (915 to 928 MHz). Band 0 includes 52 channels at RF data rates below 500 kbps and 24 channels at the 500 kbps RF data rate. Band 1 includes 24 channels at RF data rates below 500 kbps and 11 channels at the 500 kbps RF data rate.

AccessMode - this sets the channel access mode that remotes will use to communicate with the base:

Access Mode	**Description**	**Max # of Remotes**	**Remote Slot Size**
0	Polling	unlimited	manual
1	CSMA	unlimited	manual
2 (default)	TDMA dynamic slots	up to 16	automatic
3	TDMA fixed slots	up to 16	automatic
4	TDMA with PTT	up to 16	automatic

BaseSlotSize - This parameter set the maximum number of user data bytes that the base can send on a single hop. This value must be set by the user for all access modes (default value is 50 bytes).

LeasePeriod - this sets the duration in seconds for leases that remotes receive from the base. If a period of zero is specified, then lease functions are disabled. The minimum valid lease period is two seconds. Remotes will attempt to renew their leases at an interval equal to half the lease period. For example, if the lease period is set to four seconds, remotes will renew their leases every two seconds.

ARQ_Mode - this sets the ARQ mode for delivery of application messages. In ARQ mode, an ACK is expected from the receiving radio for each message addressed and sent to it. If no ACK is received, up to *ARQ_AttemptLimit,* attempts to send the data will be made, after which the message is discarded. In redundant broadcast mode, each broadcast message is sent exactly *ARQ_AttemptLimit* times. No ACKS are sent or expected. The following bit options control this function:

bits 7..2	Not used
bit 1	If set to 0, the base can pass a new *ARQ_AttemptLimit* to the remotes If set to 1, the remotes use their own *ARQ_AttemptLimit in Bank 1*

bit 0	If set to 1, the base will send broadcast packets *ARQ_AttemptLimit* times instead of once. If set to 0, broadcast packets are sent once

ARQ_AttemptLimit - this sets the maximum number of attempts that will be made to send a data packet on the RF link. Setting this parameter to the maximum value of 63 is a flag value indicating that there should be no limit to the number of attempts to send each packet (infinite number of attempts). This mode is intended for point-to-point networks in serial data cable replacement applications where absolutely no packets can be lost. Note - if this mode is used in a multipoint network, one remote that has lost link will shut down the entire network if the base is trying to send it data.

MaxNumSlots - in TDMA access modes, this sets the number of slots that are allowed. In fixed slot mode, this allocates the number of slots directly. In dynamic slot mode, this sets the maximum number of slots that may be generated regardless of the number of remotes that attempt to link with the base. Any remotes requesting registration after this limit is reached will be denied registration by the base.

CSMA_Predelay - in CSMA mode 1, this parameter sets the maximum delay between when the base transmission has finished and when a remote checks for a clear channel. The value of each parameter count depends on the data rate as shown below. Refer to Section 2.10.2 for more information.

Data Rate	Parameter
38.4 kbps	1000 µs/count
115.2 kbps	800 µs/count
200 kbps	600 µs/count
500 kbps	400 µs/count

CSMA_Backoff - in a CSMA mode 1, this parameter sets the maximum length of time that a remote will back off after it detects a busy channel. The value of each parameter count depends on the data rate as shown in the *CSMA_Predelay* table shown above. Refer to Section 2.10.2 for more information.

MaxPropDelay - this is the maximum propagation delay that the base and the remotes will use in their slot timing calculations, in units of 3.1 µs. This is used to increase the amount of time dedicated to the registration slot. Increasing this value will subtract slightly from the overall slot time available to remotes for sending data. Note that the free-space round trip propagation delay for one mile is 10.72 µs. Each increment of *MaxPropDelay* thus corresponds to a maximum radius from the remote to the base of 0.29 mi (0.46 km). The default setting provides enough time to handle remotes up to 20 miles away. It is recommended to use the default setting unless a path greater than 20 miles is planned at start up. Once linked to the base, remotes will periodically update their timing based on ranging information from the base, except in Polling Mode. The frequency with which this value is updated is set by the *RangingInterval* parameter discussed below. The current range information is available in the *CurrPropDelay* parameter.

LinkDropThreshold - this is the number of consecutive beacons missed by a remote that causes it to restart a link acquisition search. Contact RFM technical support before changing this parameter.

CSMA_RemtSlotSize - this sets the maximum size for a remote data transmission in polling or CSMA channel access modes. Setting this parameter to a large value allows a remote to send more data in a single hop, but can result in fewer remotes having time to send on a given hop. The default is 64 bytes.

CSMA_BusyThreshold - this sets the RSSI energy detection threshold that remotes use to determine whether the channel is occupied. The factory default should be sufficient for most applications and it is recommended that this value not be changed.

RangingInterval - this sets the interval in seconds/count at which remotes will reassess their range to the base. Polling (mode 0) disables the ranging interval mechanism, so remotes receive ranging information only once each time they join a network. The *RangingInterval* timer does not advance while a remote is sleeping.

AuthMode - this parameter is valid on the base only. It controls how remotes are permitted to join the network. Permitted values are:

0 = Any remote may join
1 = Authentication by base radio permission table
2 = Authentication by request to host application
3 = Lock authentication to permit only currently registered remotes

P2PReplyTimeout - This parameter sets the reply timeout for peer-to-peer packets sent from one node to another. Because each leg of the journey from one node to another and back may take multiple transmit attempts, the length of time to confirm receipt and issue a *TxDataReply* is subject to more variation than a transmission directly between a base and a remote. The *P2PReplyTimeout* parameter specifies the maximum number of hops or hop pairs that a remote will wait for a reply from its recipient. If a reply returns sooner than the timeout, the remote will send a *TxDataReply* indicating success to its host as soon as it is received, and cancels the timeout. If a reply does not come back before the timeout expires, the remote will send a *TxDataReply* to its host indicating failure. If a reply should come back after the timeout expires the remote will ignore it, as a *TxDataReply* has already been sent. The units of this parameter are in hops for non tree-routing operation and in hop pairs for tree-routing operation. The default is eight hops/eight hop pairs.

There is some coupling between the *HeartbeatIntrvl* parameter setting and the *P2PReplyTimeout* parameter setting. If the heartbeat interval in seconds is less than the *P2PReplyTimeout* in hop pairs, it is possible that a router will not repeat an un-ACKed heartbeat packet quickly enough to prevent the base from timing that router out (heartbeats are repeated only when an ACK is not received within the *P2PReplyTimeout* interval.) Thus, setting the *P2PReplyTimeout* to a very large value relative to the heartbeat interval could cause problems.

6.2.3 Bank 2 - Status Registers

Bank	Loc'n	Name	R/W	Size in bytes	Range	Default
0x02	0x00	MacAddress	R	3	0..0x0FFFFFF	fixed value
0x02	0x03	CurrNwkAddr	R	1	0..255	as set
0x02	0x04	CurrNwkID	R	1	0..255	as set
0x02	0x05	CurrRF_DataRate	R	1	0..3	as set
0x02	0x06	CurrFreqBand	R	1	0..1	as set
0x02	0x07	LinkStatus	R	1	0..4	current status
0x02	0x08	RemoteSlotSize	R	1	0..243	as set
0x02	0x09	TDMA_NumSlots	R	1	0..16	as set
0x02	0x0A	Reserved	R	1	0..255	reserved
0x02	0x0B	TDMA_CurrSlot	R	1	0..16	current slot
0x02	0x0C	HardwareVersion	R	1	0..255	0x00 = DNT900 rev A
0x02	0x0D	FirmwareVersion	R	1	0..255	current firmware load
0x02	0x0E	FirmwareBuildNum	R	2	0..2^{16}	current firmware load
0x02	0x10	Reserved	R	1	0..255	reserved
0x02	0x11	SuperframeCount	R	1	0..255	current value
0x02	0x12	RSSI_Idle	R	1	0..255	as set
0x02	0x13	RSSI_Last	R	1	0..255	as set
0x02	0x14	CurrTxPower	R	1	0..255	as set
0x02	0x15	CurrAttemptLimit	R	1	0..255	as set
0x02	0x16	CurrRangeDelay	R	1	0..255	as set
0x02	0x17	FirmwareBuildDate	R	8	ASCII	as set
0x02	0x1F	FirmwareBuildTime	R	8	ASCII	as set
0x02	0x27	ModelNumber	R	1	0..255	0x01

0x02	0x28	CurrBaseModeNetID	*R*	1	0..63, 255	0xFF
0x02	0x29	AveRXPwrOvHopSeq	*R*	1	0..255	as received
0x02	0x2A	ParentACKQual	*R*	1	0..255	4*number of attempts to get ACK

MacAddress - returns the radio's unique 24-bit MAC address.

CurrNwkAddr - this returns the address of the radio in its parent's network.

CurrNwkID - this returns the ID of the network the radio is currently assigned to or connected to. A value of 0xFF means the radio is scanning for a network but has not yet joined one.

CurrRF_DataRate - this returns the RF data rate of the network that the radio is currently assigned to or connected to. If the radio is scanning for a network, this is the current data rate it is using in the scan.

CurrFreqBand - this returns the frequency band of the network that the radio is currently assigned to or connected to. A value of 0xFF means the radio is scanning for a network but has not yet joined one.

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

LinkStatus	**Remote Status**	**Base Status**
0	initializing	initializing
1	unlinked, scanning for a network	not used
2	linked, acquiring network parameters	not used
3	linked, registering with the base	not used
4	linked and registered	ready for data transfer

RemoteSlotSize - returns the current remote slot size, defined as the maximum number of message bytes a remote can send on a single hop. When using protocol mode, the entire packet, including overhead bytes must be less than or equal to this value or the packet will be discarded. In the three TDMA modes the remote slot size is automatically computed, and this value is read-only. In polling and CSMA modes, the remote slot size must be set by the user. The parameter to set this is *CSMA_RemtSlotSize* in Bank 1.

TDMA_NumSlots - in TDMA access modes, this returns the number of slots currently allocated.

TDMA_CurrSlot - returns the current TDMA slot number assigned to the remote in modes where the slot position is automatically computed. In modes where this number is not applicable, it is read as 0xFF.

HardwareVersion - returns an identifier indicating the type of radio. A value of 0x43 is defined for the DNT900 Rev C hardware.

FirmwareVersion - returns the firmware version of the radio in 2-digit BCD format.

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

SuperframeCount - returns the current superframe count. This count increments every 64 hops.

RSSI_Idle - returns the last measurement of RSSI made during a time when the RF channel was idle. This parameter can be used to detect interferers.

RSSI_Last - returns the last measurement of RSSI made during the receipt of an RF packet with a valid CRC. This parameter can be used for network commissioning and diagnostic purposes.

CurrTxPower - returns the current transmitter power setting of a remote, allowing the automatic power setting to be tracked. This parameter is the nominal output power setting in dBm, and is a 2's complement value. Note that the *CurrTxPower* parameter value returned from a base or repeater is not valid.

CurrAttemptLimit - this returns the value of *ARQ_AttemptLimit* currently in use (depending on the selected ARQ_Mode, it may not always match the local EEPROM value).

CurrRangeDelay - returns the current propagation delay for this remote as measured from the base (applies to remote nodes only).

FirmwareBuildDate - date of firmware build in MM/DD/YY format.

FirmwareBuildTime - time of firmware build in HH:MM:SS format.

ModelNumber - DNT model number parameter, 0x01 = DNT900, 0x02 = DNT2400.

CurrBaseModeNetID - returns the current base-mode network ID.

AveRXPwrOvHopSeq - returns the average beacon power received over the last tree-routing hop sequence.

ParentACKQual - returns the number of transmission sent before and ACK is received, multiplied by 4.

6.2.4 Bank 3 - Serial Settings

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***	***Range***	***Default***
0x03	0x00	SerialRate	*R/W*	2	1..384	**0x0030** (9.6 kb/s)
0x03	0x02	SerialParams	*R/W*	1	0..7	**0x00** (8N1)
0x03	0x03	SerialControls	*R/W*	1	0..7	**0X07**
0x03	0x04	SPI_Mode	*R/W*	1	0..2	**0x00** (SPI disabled)
0x03	0x05	SPI_Divisor	*R/W*	1	$1..2^7$	**0x0A** (80.64 kb/s)
0x03	0x06	SPI_Options	*R/W*	1	0..3	**0x00** (standard SPI configuration)
0x03	0x07	SPI_MasterCmdLen	*R/W*	1	$0..2^5$	**0x00**
0x03	0x08	SPI_MasterCmdStr	R/W	32	ASCII	**all 0x00 bytes**

SerialRate - sets the serial rate divisor according to the following formula:

Serial rate in b/s = 460800/SerialRate

Serial rate division settings for commonly used baud rates are:

Setting	***Serial rate***
0x0001	460.8 kb/s
0x0002	230.4 kb/s
0x0004	115.2 kb/s
0x0006	76.8 kb/s
0x0008	57.6 kb/s
0x000C	38.4 kb/s
0x0010	28.8 kb/s
0x0018	19.2 kb/s
0x0030	9.6 kb/s (default)
0x0060	4.8 kb/s
0x00C0	2.4 kb/s
0x0180	1.2 kb/s

SerialParams - sets the serial mode options for parity and stop bits:

Setting	***Mode***
0x00	No parity, 8 data bits, 1 stop bit (default)
0x01	No parity, 8 data bits, 2 stop bits
0x02	Reserved
0x03	Reserved
0x04	Even parity, 8 data bits, 1 stop bit
0x05	Even parity, 8 data bits, 2 stop bits
0x06	Odd parity, 8 data bits, 1 stop bit
0x07	Odd parity, 8 data bits, 2 stop bits

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.

6.2.5 Bank 4 - Host Protocol Settings

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***	***Range***	***Default; Options***
0x04	0x00	ProtocolMode	*R/W*	1	0..1	**0 = transparent**; 1 = protocol
0x04	0x01	ProtocolOptions	*R/W*	1	0..255	**0x05**
0x04	0x02	TxTimeout	*R/W*	1	0..255	**0x00 (no timeout)**
0x04	0x03	MinPacketLength	*R/W*	1	1..255	**1 byte**
0x04	0x04	AnnounceOptions	*R/W*	1	0..7	**0x07 all enabled**
0x04	0x05	TransLinkAnnEn	*R/W*	1	0..1	**0 = disabled**; 1 = <LINK> announce
0x04	0x06	ProtocolSequenceEn	*R/W*	1	0..2	0 = disabled; 1 = startup, **2 = anytime**
0x04	0x07	TransPtToPtMode	*R/W*	1	0..1	**0 = multipoint**, 1 = point-to-point
0x04	0x08	MaxPktsPerHop	*R/W*	1	0..3	**0x03**

ProtocolMode - this parameter selects the host protocol mode. The default is 0, which is transparent mode, meaning the radio conveys whatever characters that are sent to it transparently, without requiring the host to understand or conform to the DN-900's built-in protocol. This setting is recommended for point-to-point applications for legacy applications such as wire replacements where another serial protocol may already exist. Setting this parameter to 1 enables the DN-900 host protocol, which is recommended for point-to-multipoint applications and is preferred for new designs. It is not necessary to define the same protocol mode for all radios in a network. For example, it is frequently useful to configure all the remotes for transparent mode and the base for protocol mode. Note that it is possible for the host to switch the radio from transparent mode to protocol mode and back if desired by transmitting an *EnterProtocolMode* command.

ProtocolOptions - this is a bitmask that selects various options for the protocol mode. The default is 0x05.

bits 7..3	Reserved
bit 2	Enable output of TxReply packets
bit 1	Reserved
bit 0	Enable output of Announce packets

AnnounceOptions - this is a bitmask that enables/disables different types of Announce packets:

bit 7..3	Reserved
bit 2	Enable bit for Announce types E0-EA (error notification)
bit 1	Enable bit for Announce types A1-A7 (<LINK> notifications)
bit 0	Enable bit for Announce types A0 (initialization)

TxTimeout - this parameter is the transmit timeout used for determining message boundaries in transparent data mode. Units are in milliseconds. A message boundary is determined whenever a gap between consecutive characters is equal to or greater than the *TxTimeout* value, or the number of bytes reaches the *MinPacketLength*. Either condition will trigger a transmission. The default *TxTimeout* value is 0 ms.

MinPacketLength - sets the minimum message length used for determining packet boundaries in transparent data mode. The default is one byte. A transmission is triggered when either the number of bytes reaches *MinPacketLength* or a gap is detected between consecutive characters greater than *TxTimout*.

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

ProtocolSequenceEn - enables or disables the *EnterProtocolMode* ASCII command string to switch from transparent mode to protocol mode. Valid settings are 0 = disabled, 1 = one time at startup, 2 = enabled at any time. The default is enabled at any time. Note that if this parameter is set to 0 and saved to memory, protocol mode can no longer be invoked through the radio's main serial port or SPI connection. In this case, protocol mode must be invoked using the module's /CFG hardware input or by an over-the-air command addressed to the radio.

TransPtToPtMode- controls the behavior for addressing packets in transparent mode. When this setting is zero (default), in transparent mode the base will direct packets to the broadcast address (0xFFFFFF). This is useful for point-to-multipoint where the base is sending data to multiple remotes, for instance in applications where a wireless link is replacing an RS-485 serial bus. When this setting is one, in transparent mode the base will direct packets to the last remote that registered with it. This is useful for point-to-point networks where there are only two endpoints, for instance in applications where a simple serial cable is being replaced.

MaxPktsPerHop - this parameter sets a limit on the maximum number of packets a radio can send on each frequency hop. The default value is 3, the range is 1 to 3.

6.2.6 Bank 7 - Authentication List

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***
0x07	0x00	ApprovedAddr0	*R/W*	3
0x07	0x03	ApprovedAddr1	*R/W*	3
0x07	0x06	ApprovedAddr2	*R/W*	3
0x07	0x09	ApprovedAddr3	*R/W*	3
0x07	0x0C	ApprovedAddr4	*R/W*	3
0x07	0x0F	ApprovedAddr5	*R/W*	3
0x07	0x12	ApprovedAddr6	*R/W*	3
0x07	0x15	ApprovedAddr7	*R/W*	3
0x07	0x18	ApprovedAddr8	*R/W*	3
0x07	0x1B	ApprovedAddr9	*R/W*	3
0x07	0x1E	ApprovedAddr10	*R/W*	3
0x07	0x21	ApprovedAddr11	*R/W*	3
0x07	0x24	ApprovedAddr12	*R/W*	3
0x07	0x27	ApprovedAddr13	*R/W*	3
0x07	0x2A	ApprovedAddr14	*R/W*	3
0x07	0x2D	ApprovedAddr15	*R/W*	3

ApprovedAdd0..15 - The three-byte parameters in Bank 7 are the MAC addresses of the remotes authorized to join the network. The addresses are entered in little-endian format such that a radio with MAC address 012345 would be entered 0x452301.

6.2.7 Bank 8 - Tree-Routing Active Router ID Table

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***
0x08	0x00	Base NetworkID (0x00)	*R*	1
0x08	0x01	ParentNetworkID1	*R*	1
	to			
0x08	0x3F	ParentNetworkID63	R	1

ParentNetID0..63 - This set of parameters contains the tree-routing active router ID table maintained by a base for its system. It describes the organization of all active routers in the system. This table is used by the base and the routers to determine which direction to send a packet. The base updates the information in the routing table from the heartbeat packets it receives from the routers in the system, and broadcasts the routing table periodically to inform all devices in the system of the current system configuration.

6.2.8 Bank 9 - Registered MAC Addresses

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***
0x09	0x00	RegMACAddr0	*R*	15
	to			
0x09	0x19	RegMACAddr25	R	15

RegMACAddr0..25 - This bank holds the MAC addresses of all radios registered to a base or router. Up to 126 MAC addresses can be registered. Each bank parameter can hold up to five MAC addresses, with each MAC address containing three bytes are in little-Endian order. Three-byte segments in a parameter not holding a MAC address with hold a null address: 0x00 0x00 0x00. Note that unlike parameters in other banks, the bank offset used in get commands is *by parameter rather than by byte*. Only one Bank 9 parameter can be retrieved at a time with a get command. In a remote, this bank will contain only null addresses.

6.2.9 Bank FF - Special Functions

This bank contains three user functions, *UcReset*, *SleepModeOverride* and *MemorySave*:

Bank	***Loc'n***	***Name***	***R/W***	***Size in bytes***	***Range***	***Description***
0xFF	0x00	UcReset	*W*	1	0..90	0x00 = reset, 0x5A = reset with factory defaults
0xFF	0x0C	SleepModeOverride	R/W	1	0..2	**0 = inactive**, 1 = stay awake, 2 = cancel stay awake
0xFF	0x1C	RoutingTableUpd	R/W	1	0..255	**0x14 (20 seconds)**
0xFF	0x20	DiagSerialRate	R/W	2	0..384	**0x000C (38.4 kb.s)**
0xFF	0xFF	MemorySave	*W*	1	0..2	0x00 = load factory defaults, 0x01 = save settings to EEPROM, 0x02 = save settings and reset

UcReset - writing a value of 0x00 to this location forces a software reset of the microcontroller. This will enable those changes which require a reset. If this is written to before 0x01 is written to the *MemorySave* parameter, the last parameter values saved before the reset will be in effect. A reply packet, either local or over-the-air, may not be received when writing a value to this register. Writing 0x5A to this location will reset the radio and load the factory default settings. This is equivalent to writing 0x00 to *UcReset* followed by writing 0x00 to *MemorySave*. Note that 0x01 must be written to *MemorySave* to save the retrieved factory defaults.

RoutingTableUpd - this parameter is the interval in seconds for the base station to broadcast the tree-routing table to its system. The default interval is 20 seconds.

0x0030	9.6 kb/s
0x0060	4.8 kb/s
0x00C0	2.4 kb/s
0x0180	1.2 kb/s

Note that if a value of 0x0000 is specified, the maximum data rate of 460.8 kb/s will be selected.

MemorySave - writing 0x00 to this location clears all registers back to factory defaults. Writing a 0x01 to this location commits the current register settings to EEPROM. Writing 0x02 to this location saves the current setting to EEPROM and forces a software reset. When programming registers, all changes are considered temporary until a 0x01 or 0x02 command is executed.

6.2.10 Protocol Mode Configuration Example

In this example, the host configures the base to transmit 10 dBm (10 mW) of RF power using the *SetRegister* command, 0x04. The *TxPower* parameter is stored in bank 0x00, register 0x18. A one-byte parameter value of 0x01 selects the 10 dBm (10 mW) power level. The protocol formatting for the command is:

0xFB 0x05 0x04 **0x18 0x00 0x01 0x01**

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 return a *SetRegisterReply* message as follows:

0xFB 0x01 **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 a one-byte parameter in register 0xFF of bank 0xFF to 0x01 with another *SetRegister* command:

0xFB 0x05 0x04 **0xFF 0xFF 0x01 0x01**

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

0xFB 0x01 **0x14**

7.0 DN-900 Part Number Ordering Guide

Serial Interface	Antenna/Connector	Cable Length, feet			
		4	50	100	300
RS-232C, DB9 Connector	Internal 2 dBi Antenna	DN-900G-4	DN-900G	DN-900G-100	DN-900G-300
RS-232C, DB9 Connector	RTNC Connector	DN-900GX-4	DN-900GX	DN-900GX-100	DN-900GX-300
RS-485/RS-232C, 6-Pin Terminal Block	Internal 2 dBi Antenna	DN-900I-4	DN-900I	DN-900I-100	DN-900I-300
RS-485/RS-232C, 6-Pin Terminal Block	RTNC Connector	DN-900IX-4	DN-900IX	DN-900IX-100	DN-900IX-300
USB, Type B Connector	Internal 2 dBi Antenna	DN-900U-4	DN-900U	DN-900U-100	DN-900U-300
USB, Type B Connector	RTNC Connector	DN-900UX-4	DN-900UX	DN-900UX-100	DN-900UX-300

RTNC to N connector adaptor cable for connecting to external antennas with N connectors: CBLRF24NR

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.

Part # M-9000-5002, Rev A