



Excellence in Compliance Testing

---

## **Certification Exhibit**

**FCC ID: HSW-DNT2400P  
IC: 4492A-DNT2400P**

**FCC Rule Part: 15.247  
IC Radio Standards Specification: RSS-247**

**ACS Project Number: 16-0346**

**Manufacturer: Murata Electronics North America  
Models: DNT2400PC, DNT2400PP**

## **Manual**

*1 of 2*

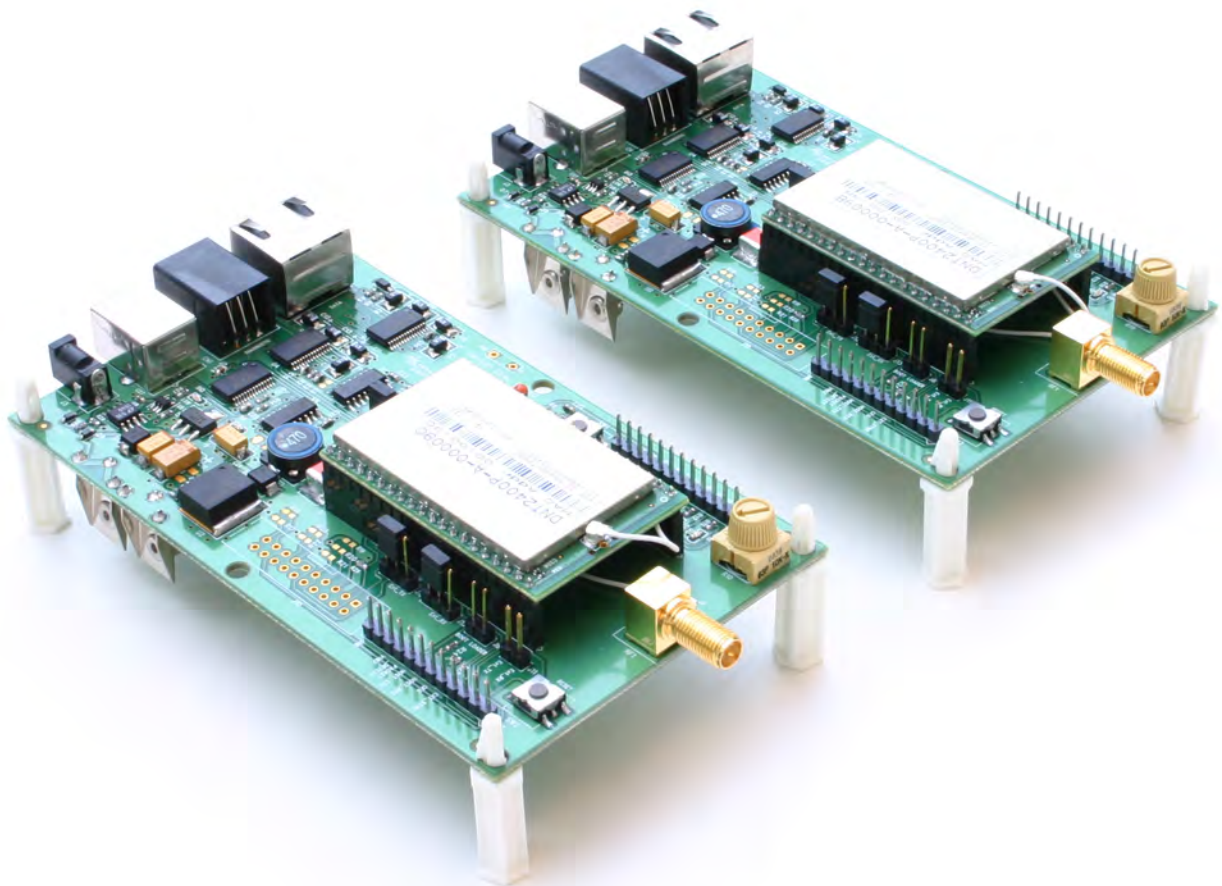
## DNT2400

2.4 GHz  
Spread Spectrum  
Wireless Transceivers

Integration Guide

## Revision History

Revision	Date	Author	Change Description
1.0	08/18/09	F. Perkins	Initial issue
2.0	11-11-14	R. Willett	Reformatted document to conform to Murata V.I.
2.1	06-03-15	R. Willett	Added EN 300.328 v 1.8.1 compliance parameters to section 3.10.
3.0	10-25-16	R. Willett	Added information for Industry Canada on pages



# Important Regulatory Information

**Murata Product FCC ID: HSW-DNT2400  
IC: 4492A-DNT2400P**

---

**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:

1. Reorient or relocate the receiving antenna.
  2. Increase the separation between the equipment and receiver.
  3. Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
  4. 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.

## **FCC Antenna Gain Restriction and MPE Statement:**

### RF Exposure Information:

For mobile operating conditions (>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 mobile operating conditions, the DNT2400 has been designed to operate with dipole antennas of up to 9 dBi of gain, or patch antennas of up to 6 dBi gain. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

*Cet équipement est conforme aux limites d'exposition aux radiations définies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé à une distance minimale de 20 cm entre le radiateur et votre corps. Cet émetteur ne doit pas être situé ou opérant en conjonction avec une autre antenne ou émetteur.*

### **Notices:**

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

---

FCC NOTICE: This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

*Innovation, Science, and Economic Development (ISED) Canada Notice: This device complies with ISED Canada's license-exempt RSSs. Operation is subject to the following two conditions: (1) This device may not cause harmful interference; and (2) This device must accept any interference, including interference that may cause undesired operation.*

*Le présent appareil est conforme aux CNR ISED Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:*

*(1) l'appareil ne doit pas produire de brouillage, et*

*(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.*

#### **ISED RSS-247 Detachable Antenna Gain Restriction:**

*This radio transmitter (DNT2400P), has been approved by ISED Canada to operate with the antenna types listed below with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, or having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.*

*Le présent émetteur radio (DNT2400P) a été approuvé par ISED Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.*

*Antennas not included in this list or having a gain greater than 9dBi dipole or 6 dBi patch are strictly prohibited for use with this device. The required antenna impedance is 50 ohms:*

Omnidirectional Dipole Antenna, 9 dBi  
Patch Antenna, 6 dBi

See Section 3.9 of this manual for regulatory notices and labeling requirements. Changes or modifications to a DNT2400P not expressly approved by MURATA may void the user's authority to operate the module.

For Portable operating conditions (<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.

For portable operating conditions, the DNT2400 has been designed to operate with dipole or patch antennas of up to 3.47 dBi of gain. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device. This device does not comply with ISED exposure limits for portable applications. Any use for portable applications in Canada is strictly prohibited.

Ce dispositif ne respecte pas les limites d'exposition ISED pour les applications portables. Toute utilisation pour les applications portables au Canada est strictement interdite.

See Section 3.9 of this manual for additional regulatory notices and labeling requirements.

## Table of Contents

1.0	Introduction .....	7
1.1	Why Spread Spectrum? .....	7
1.2	Frequency Hopping versus Direct Sequence .....	8
2.0	DNT2400 Radio Operation .....	9
2.1	Network Synchronization and Registration .....	9
2.2	Authentication .....	10
2.3	Serial Port Modes .....	10
2.4	SPI Port Modes .....	11
2.5	RF Data Communications .....	11
2.6	RF Transmission Error Control .....	12
2.7	Transmitter Power Management .....	12
2.8	Network Configurations .....	12
2.8.1	Point-to-Point Network Operation .....	12
2.8.2	Point-to-Multipoint Network Operation .....	13
2.8.3	Multipoint Peer-to-Peer Network Operation .....	13
2.8.4	Tree Routing System Operation .....	14
2.9	Full-Duplex Serial Data Communications .....	14
2.10	Channel Access .....	14
2.10.1	Polling Mode .....	15
2.10.2	CSMA Mode .....	15
2.10.3	TDMA Modes .....	16
2.11	Transmission Configuration Planning .....	17
2.11.1	TDMA Throughput .....	17
2.11.2	Polling Throughput .....	18
2.11.3	CSMA Throughput .....	18
2.11.4	Latency .....	19
2.11.5	Configuration Validation .....	20
2.12	Tree-Routing Systems .....	22
2.12.1	Example Tree-Routing System .....	22
2.12.2	Tree-Routing System Networks .....	24
2.12.3	Tree-Routing System Addressing .....	24
2.12.4	Tree-Routing System Implementation Options .....	25
2.13	Serial Port Operation .....	27
2.14	SPI Port Operation .....	28
2.15	Sleep Modes .....	33
2.16	Encryption .....	33
2.17	Synchronizing Co-located Bases .....	33
3.0	DNT2400 Hardware .....	35
3.1	Specifications .....	36
3.2	Module Interface .....	37
3.3	DNT2400 Antenna Connector .....	38
3.4	Input Voltages .....	39
3.5	ESD and Transient Protection .....	39
3.6	Interfacing to 5 V Logic Systems .....	39
3.7	Power-On Reset Requirements .....	40
3.8	Mounting and Enclosures .....	40
3.9	Labeling and Notices .....	40

4.0	Protocol Messages .....	43
4.1	Protocol Message Formats.....	43
4.1.1	Message Types .....	43
4.1.2	Message Format Details .....	44
4.1.3	/CFG Select Pin.....	46
4.1.4	Flow Control .....	46
4.1.5	Protocol Mode Data Message Example.....	46
4.1.6	Protocol Mode Tree-Routing MAC Address Discovery Example.....	46
4.2	Configuration Registers .....	47
4.2.1	Bank 0 - Transceiver Setup.....	47
4.2.2	Bank 1 - System Settings .....	50
4.2.3	Bank 2 - Status Registers .....	51
4.2.4	Bank 3 - Serial and SPI Settings .....	55
4.2.5	Bank 4 - Host Protocol Settings .....	58
4.2.6	Bank 5 - I/O Peripheral Registers .....	59
4.2.7	Bank 6 - I/O Setup.....	60
4.2.8	Bank 7 - Authentication List .....	61
4.2.9	Bank 8 - Tree-Routing Active Router ID Table .....	62
4.2.10	Bank 9 - Registered MAC Addresses .....	62
4.2.11	Bank FF - Special Functions .....	63
4.2.12	Protocol Mode Configuration Example.....	64
4.2.13	Protocol Mode Sensor Message Example .....	64
4.2.14	Protocol Mode Event Message Example .....	64
5.0	DNT2400DK Developer's Kit .....	66
5.1	DNT2400DK Kit Contents.....	66
5.2	Additional Items Needed .....	66
5.3	Developer's Kit Operational Notes .....	66
5.4	Developer's Kit Default Operating Configuration .....	67
5.5	Developer's Kit Hardware Assembly .....	67
5.6	DNT Demo Utility Program .....	69
5.6.1	Initial Kit Operation .....	70
5.6.2	Serial Communication and Radio Configuration .....	73
5.7	DNT Wizard Utility Program .....	80
5.8	DNT2400 Interface Board Features .....	83
6.0	Demonstration Procedure .....	91
7.0	Troubleshooting .....	92
7.1	Diagnostic Port Commands.....	92
8.0	Appendices .....	95
8.1	Ordering Information.....	95
8.2	Technical Support.....	95
8.3	DNT2400 Mechanical Specifications.....	96
8.4	DNT2400 Development Board Schematic .....	98
9.0	Warranty .....	100

## 1.0 Introduction

The DNT2400 series transceivers provide highly reliable wireless connectivity for point-to-point, point-to-multipoint, peer-to-peer or tree-routing applications. Frequency hopping spread spectrum (FHSS) technology ensures maximum resistance to multipath fading and robustness in the presence of interfering signals, while operation in the 2.4 GHz ISM band allows license-free use world wide. The DNT2400 supports all standard serial data rates for host communications from 1.2 to 460.8 kb/s plus SPI data rates from 6.35 to 80.64 kb/s. On-board data buffering and an error correcting radio protocol provide smooth data flow and simplify the task of integration with existing applications. Key DNT2400 features include:

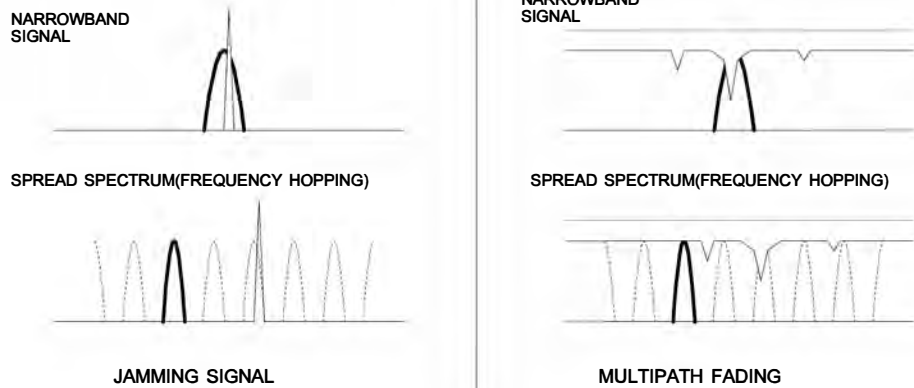
- Multipath fading resistant frequency hopping technology with up to 37 frequency channels per subband
- Support for point-to-point, point-to-multipoint, peer-to-peer and tree-routing networks
- FCC 15.247, IC and ETSI certified for license-free operation
- 10 mile plus range with omni-directional antennas (antenna height dependent)
- Transparent ARQ protocol with data buffering ensures data integrity
- Analog and digital I/O simplifies wireless sensing
- Dynamic TDMA slot assignment that maximizes throughput and CSMA modes that maximizes network size
- AES encryption provides protection from eavesdropping
- Nonvolatile memory stores DNT2400 configuration when powered off
- Selectable 1, 10 and 63 mW transmit power levels
- Simple interface handles both data and control at up to 460.8 kb/s on the serial port or 80.64 kb/s on the SPI port
- Auto-reporting I/O mode simplifies application development

### 1.1 Why Spread Spectrum?

A radio channel can be very hostile, corrupted by noise, path loss and interfering transmissions from other radios. Even in an interference-free environment, radio performance faces serious degradation from a phenomenon known as multipath fading. Multipath fading results when two or more reflected rays of the transmitted signal arrive at the receiving antenna with opposing phases, thereby partially or completely canceling the signal. This problem is particularly prevalent in indoor installations. In the frequency domain, a multipath fade can be described as a frequency-selective notch that shifts in location and intensity over time as reflections change due to motion of the radio or objects within its range. At any given time, multipath fades will typically occupy 1% - 2% of the band. From a probabilistic viewpoint, a conventional radio system faces a 1% - 2% chance of signal impairment at any given time due to multipath fading.

Spread spectrum reduces the vulnerability of a radio system to both multipath fading and jammers by distributing the transmitted signal over a larger frequency band than would otherwise be necessary to send the information. This allows a signal to be reconstructed even though part of it may be lost or corrupted in transmission.

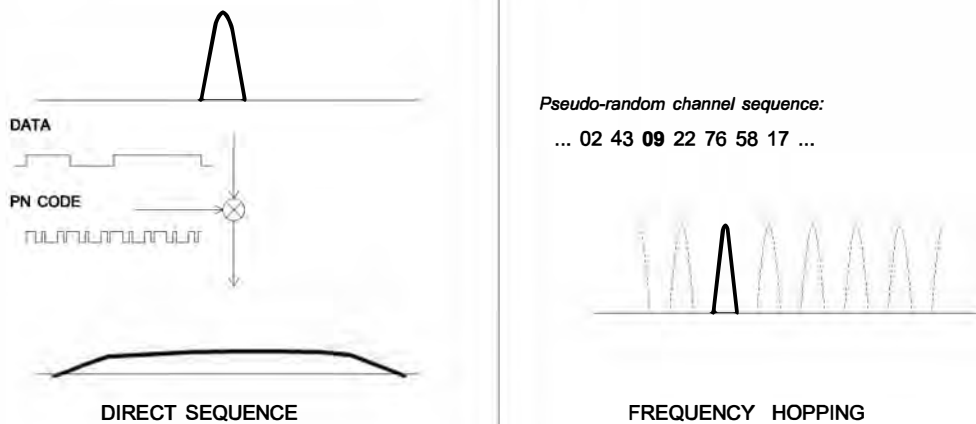




Narrow-band versus spread-spectrum transmission  
Figure 1.1.1

## 1.2 Frequency Hopping versus Direct Sequence

The two primary approaches to spread spectrum are direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS), either of which can generally be adapted to a given application. Direct sequence spread spectrum is produced by multiplying the transmitted data stream by a much faster, noise-like repeating pattern. The ratio by which this modulating pattern exceeds the bit rate of the base-band data is called the processing gain, and is equal to the amount of rejection the system affords against narrow-band interference from multipath and jammers. Transmitting the data signal as usual, but varying the carrier frequency rapidly according to a pseudo-random pattern over a broad range of channels produces a frequency hopping spread spectrum system.



Forms of spread spectrum - direct sequence and frequency hopping  
Figure 1.1.2

One disadvantage of direct sequence systems is that due to design issues related to broadband transmitters and receivers, they generally employ only a minimal amount of spreading, often no more than the minimum required by the regulating agencies. For this reason, the ability of DSSS systems to overcome fading and in-band jammers is relatively weak. By contrast, FHSS systems are capable of hopping throughout the entire band, statistically reducing the chances that a transmission will be affected by fading or interference. This means that a FHSS system will degrade gracefully as the band gets noisier, while a DSSS system may exhibit uneven coverage or work well until a certain point and then give out completely.

Because it offers greater immunity to interfering signals, FHSS is often the preferred choice for co-located systems. Since direct sequence signals are very wide, they can offer only a few non-overlapping channels, whereas multiple hoppers can interleave, minimizing interference. Frequency hopping systems do carry some disadvantages, in that they require an initial acquisition period during which the receiver must lock onto the moving carrier of the transmitter before any data can be sent, which typically takes several seconds. In summary, frequency hopping systems generally feature greater coverage and channel utilization than comparable direct sequence systems. Of course, other implementation factors such as size, cost, power consumption and ease of implementation must also be considered before a final radio design choice can be made.

## 2.0 DNT2400 Radio Operation

### 2.1 Network Synchronization and Registration

As discussed above, frequency hopping spread spectrum radios such as the DNT2400 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 DNT2400 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 37 frequencies and the remotes and routers are scanning up to 37 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*, *FrequencyBand* and *Nwk\_Key* (see Section 4.2 for parameter details). Note that if a registration parameter is changed at the base, it will update the 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. DNT2400 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 DNT2400 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.

## 2.2 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.

## 2.3 Serial Port Modes

DNT2400 radios can work in two data modes on the primary serial (UART) port: 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 simple 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, and sensor I/O 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 4.

The DNT2400 provides two ways to switch between transparent and protocol modes. To enter protocol mode, the /CFG input Pin 18 can be switched from logic high to low, or the *EnterProtocolMode* command can be sent. When input Pin 18 is switched from logic low to high, or an *ExitProtocolMode* command is sent to the primary serial input, the DNT2400 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.

When operating in transparent mode, two configuration parameters control when a DNT2400 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 DNT2400 transmit buffer will be sent on the next hop. As discussed in Section 2.8.2, 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.

## 2.4 SPI Port Modes

DNT2400 radios can be configured to transmit and receive data through the serial peripheral interface (SPI) port instead of the primary serial (UART) port. A DNT2400 can operate as either an SPI Master or Slave. Messages routed through the SPI port must be protocol formatted, as described in Section 4. When a DNT2400 is acting as an SPI Slave, all messages are routed through the SPI port. When a DNT2400 is acting as an SPI Master, only data messages are routed through the SPI port; radio configuration commands and responses are routed through the primary serial (UART) port. A radio configured as an SPI Master can use a command stored in the *SPI\_MasterCmdStr* parameter to interrogate a Slave peripheral for data to transmit to the base. This is especially useful where periodic I/O reporting is enabled on the remote. Alternately, the base can send an interrogation command to the radio to fetch peripheral data. SPI operation is configured with the *SPI\_Mode* parameter.

## 2.5 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 in its transmit buffer, unless in transparent mode the *MinPacketLength* and/or *TxTimeout* parameters have been set above their default values. The maximum amount of user data bytes that the base can transmit per hop is limited by the *BaseSlotSize* parameter, as discussed in Section 2.8.1. 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 *HopDuration* 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.

## 2.6 RF Transmission Error Control

The DNT2400 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.

## 2.7 Transmitter Power Management

The DNT2400 includes provisions for setting the base transmit power level and the remote maximum transmit power level with the *TxPower* parameter. DNT2400 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. Refer to Section 4.2.1 for details.

## 2.8 Network Configurations

The DNT2400 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, as discussed in Section 2.8.3. Tree-routing networks can retransmit messages through one or more routers, greatly expanding the area that can be covered by a single DNT2400 system, as discussed in Section 2.8.4.

### 2.8.1 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.

### 2.8.2 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 DNT2400 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.

### 2.8.3 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 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 *TxDatReply* packet is reported to the host.

## 2.8.4 Tree-Routing Operation

A DNT2400 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 DNT2400 tree-routing system can cover a much larger area than other DNT2400 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. See Section 2.12 below.

## 2.9 Full-Duplex Serial Data Communications

From a host application's perspective, DNT2400 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 visa versa.

## 2.10 Channel Access

The DNT2400 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 DNT2400 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 2.10.1



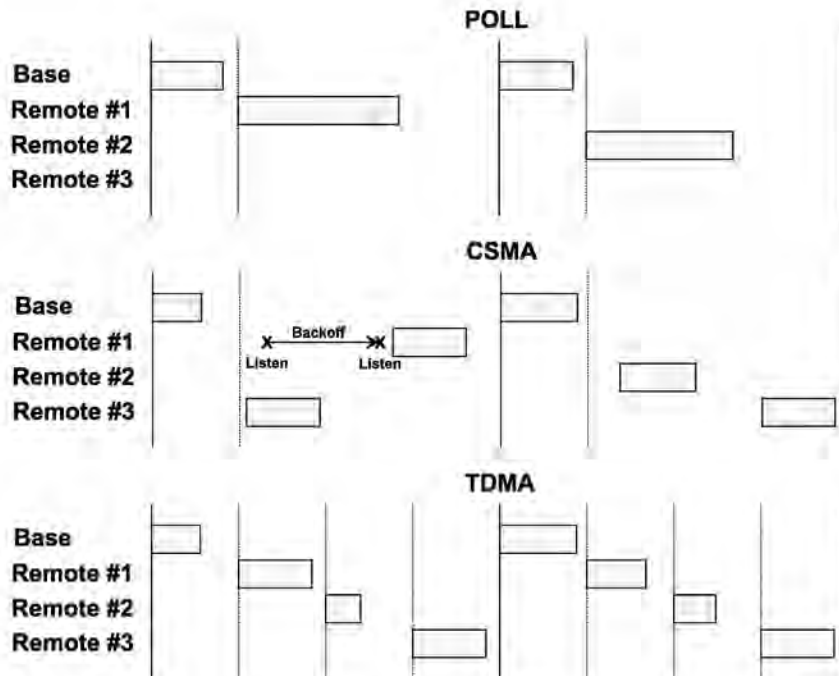


Figure 2.10.1

### 2.10.1 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.

### 2.10.2 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\_RemotSlotSize* 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.

### 2.10.3 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 DNT2400 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 DNT2400 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 DNT2400 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 them 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 DNT2400 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.

## 2.11 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 DNT2400 has several configuration parameters that allow latency and throughput to be optimally balanced to the needs of an application.

### 2.11.1 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:

$$\text{Number of bytes per hop/Hop Duration}$$

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

$$\text{BaseSlotSize/HopDuration}$$

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:

$$\text{RemoteSlotSize}/\text{HopDuration}$$

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.

### 2.11.2 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\_RemotSlotSize* 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 DNT2400 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 DNT2400. 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 DNT2400 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.

### 2.11.3 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 DNT2400 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 DNT2400 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 of 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.

#### 2.11.4 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 DNT2400 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.

### 2.11.5 Configuration Validation

Although slot durations are automatically calculated by the DNT2400, 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 DNT2400 hop duration is limited to 200 ms. A DNT2400 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\_RemotSlotSize* 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\_RemotSlotSize* 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 2.11.5.1

The *DNT Throughput Calculator* utility program is shown in Figure 2.11.5.1. 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 DNT2400 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 DNT2400'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 DNT2400 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 DNT2400 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.



## 2.12 Tree-Routing Systems

As discussed in Section 2.8.4, DNT2400 tree-routing systems can cover much larger areas than other DNT2400 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 DNT2400 network configurations, however, tree-routing systems require somewhat more initial planning and commissioning steps, as discussed in this Section.

### 2.12.1 Example Tree-Routing System

An example tree routing system is shown in Figure 2.12.1.1. 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 DNT2400, 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 DNT2400 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.

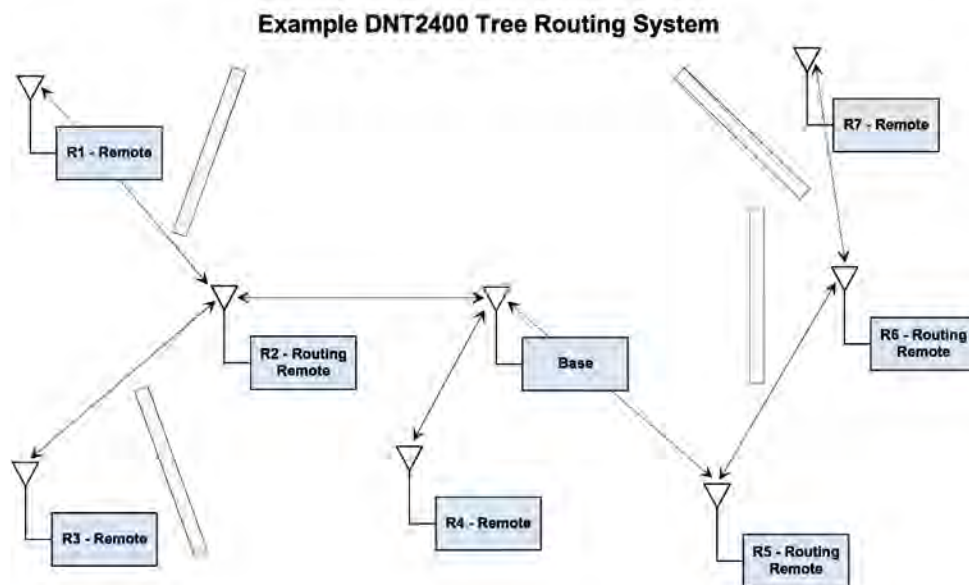


Figure 2.12.1.1

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 DNT2400 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 DNT2400s, 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 2.12.1.2 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 the tree routing related configuration commands plus details of the addressing used in a tree routing system.

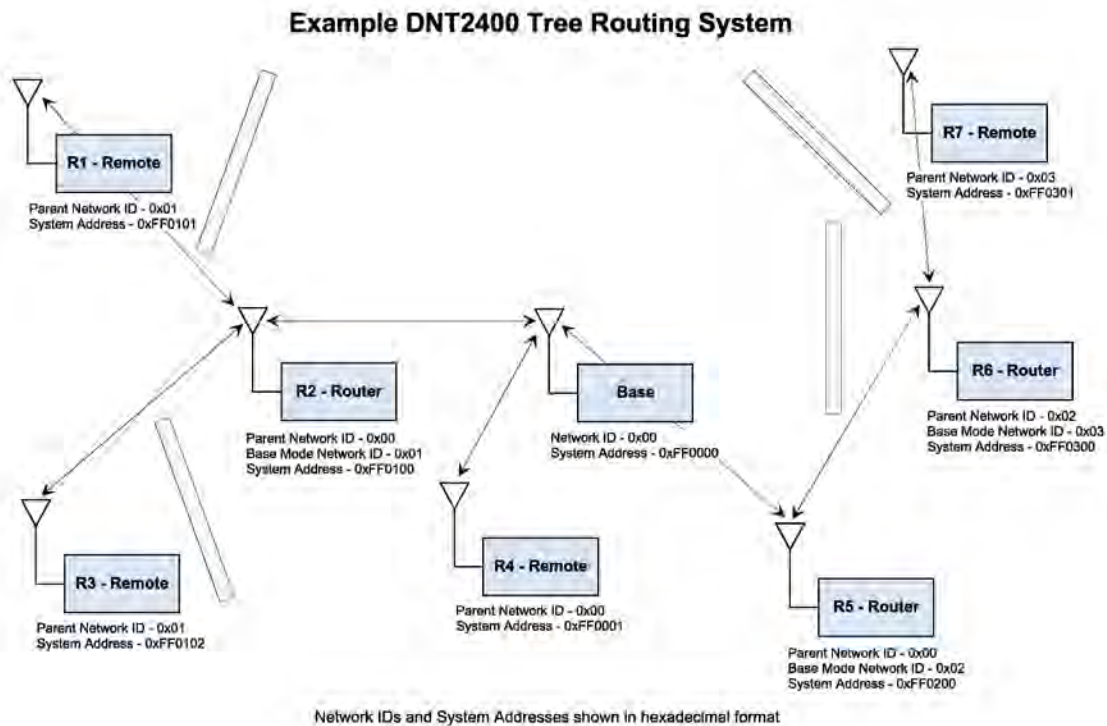


Figure 2.12.1.2



## 2.12.2 Tree-Routing System Networks

A DNT2400 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)<sup>1</sup>. A child radio can be either a remote or router, within the system limitation of 63 routers total.

Within a DNT2400 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).

## 2.12.3 Tree-Routing System Addressing

Except for tree-routing systems, DNT2400 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 DNT2400 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 DNT2400 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 DNT2400 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.

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.



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 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.

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 DNT2400 tree-routing system is shown in Figure 2.12.1.2. 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 contains 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.

#### 2.12.4 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 2.12.4.1 summarizes radio parameter settings for each assignment method.

<b>Dynamic Router Parent Assignment and Dynamic Remote System Address Assignment</b>			
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)
<b>Manual Router Parent Address Assignment with Dynamic Remote System Address Assignment</b>			
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)
<b>Manual Router Parent and Remote System Address Assignment</b>			
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 2.12.4.1

Tree-routing networks can support peer-to-peer communications. However, the value of the *P2PReplyTimeout* parameter (see Section 4.2.2) 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 2.12.1.2, the minimum useable value for peer-to-peer communications between Remote 1 and Remote 7 is determined as follows:

<b>Route Segment</b>	<b>Required Hops</b>
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
<b>Total</b>	<b>13</b>

Table 2.12.4.2

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.



## 2.13 Serial Port Operation

DNT2400 networks are often used for wireless communication of serial data. The DNT2400 supports serial baud rates from 1.2 to 460.8 kb/s. Listed in Table 2.13.1 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 2.13.1

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 2.13.1, the hop duration from the *DNT Throughput Calculator* program for this configuration is 4.85 ms:



Figure 2.13.1

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

$$64 \text{ Bytes}/4.85 \text{ ms} = 13.2 \text{ kB/s, or } 132 \text{ kb/s for } 8N1$$

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.

The DNT2400 transmit and receive buffers hold at least 1024 bytes and will accept brief bursts of data at high baud rates, provided the average serial port data flow such as shown in the example above is not exceeded. It is strongly recommended that the DNT2400 host use hardware flow control in applications where the transmit buffer can become full. The host must send no more than 32 additional bytes to the DNT2400 when the DNT2400 de-asserts the host's CTS line. In turn, the DNT2400 will send no more than one byte following the host de-asserting its RTS line. Three-wire serial port operation is allowed through parameter configuration, as discussed in Section 4.2.4. However, data loss is possible under adverse RF channel conditions when using three-wire serial operation due to buffer overruns.

## 2.14 SPI Port Operation

The DNT2400 SPI port data rate is configurable in 127 steps from 6.35 to 80.64 kb/s using the *SPI\_Divisor* parameter. The SPI data rate is the clocking rate the DNT2400 uses in Master mode. The SPI data rate is also used in Slave mode to time SPI Select (/SS) sampling, etc. Where possible, devices connected to the DNT2400 SPI port should be configured to match the 80.64 kb/s data rate. In any event, the SPI data rate used by the DNT2400 and the connected device should be matched within a few percent for efficient data transfers. SPI port operation is full duplex in the sense that a single clock signal simultaneously shifts data into and out of the SPI port. In order for the Master (host) to receive data from a DNT2400 SPI Slave, it must clock bytes into the DNT2400.

As show in Figure 2.14.1, DNT2400 SPI slave mode operation is supported by two control signals from the DNT2400. The /HOST\_CTS line provides the same flow control function for SPI Slave mode as it does for the serial (UART) port. The Master (host) can clock messages into the DNT2400 SPI Slave whenever /HOST\_CTS is set to a logic low state. The Master can complete clocking one protocol formatted message into the DNT2400 if /HOST\_CTS switches high part way through the message as shown in Figure 2.14.2, but must then stop sending transmit message bytes until the DNT2400 resets /HOST\_CTS to a logic low state. If the DNT2400 slave is holding a received message(s) when the SPI master clocks in a message to transmit, received message bits will be simultaneously clocked out on the MISO line while the message to be transmitted is clocked in on the MOSI line. The end of a received message will often not be aligned with the end of a message to be transmitted, so additional clocking may be required to complete the transfer of a received message(s). It is acceptable to clock 0x00 null bytes in on the MOSI line to retrieve received message bytes when /HOST\_CTS is high, but not transmit message bytes.

### DNT2400 SPI Slave Mode Signaling

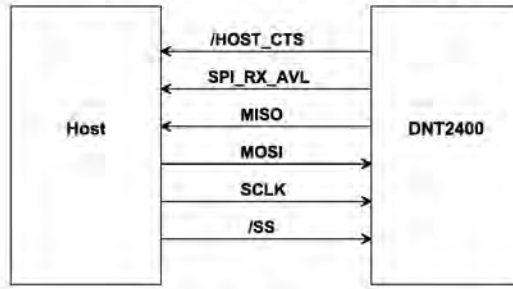


Figure 2.14.1

### DNT2400 SPI Slave Mode TX Message Load

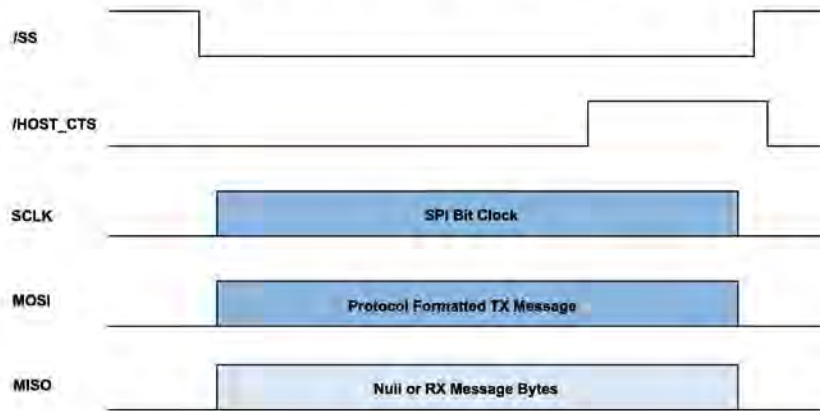


Figure 2.14.2

### DNT2400 SPI Slave Mode RX Message Retrieval

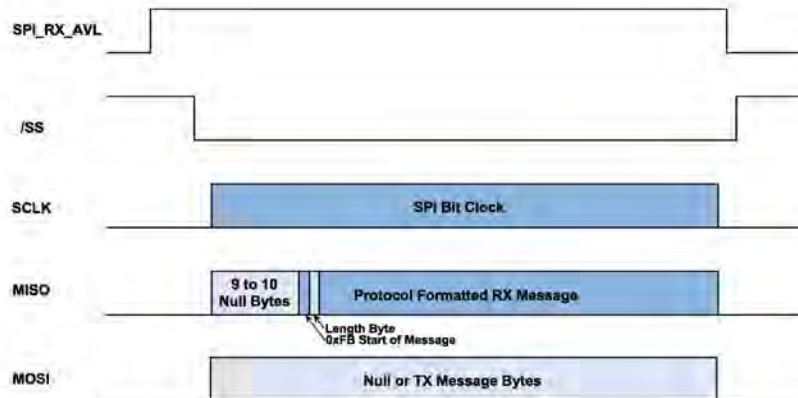


Figure 2.14.3



GPIO port 4 can be alternately configured to provide a received message flag, referred to as SPI\_RX\_AVL. Operation of this line is shown in Figure 2.14.3. Once the SPI\_RX\_AVL flag goes high, RX message bytes can be clocked out on the MISO line by clocking 0x00 null bytes and/or transmit message bytes in on the MOSI line. Note that nine to ten 0x00 null bytes will precede each received message, which is always protocol formatted. Following the null bytes, the 0xFB start of message symbol will then be clocked out, followed by the message length byte. The length byte can be used to confirm that all the bytes in each received message have been clocked out. If a DNT2400 slave is holding more than one received message, the SPI\_RX\_AVL flag will remain high until all messages have been completely clocked out.

In SPI slave mode, de-asserting and then asserting the /SS line resets the DNT2400 SPI port on a byte boundary. The /SS line can be toggled this way between every byte to assure bit streams into and out of the SPI port remain byte framed. Less frequent /SS line toggling is also acceptable in most applications. It is recommended that /SS be toggled at the start and end of each transmit message, and after every 256 null bytes when clocking to output a received message. The /SS line should also be toggled at the end of each received message.

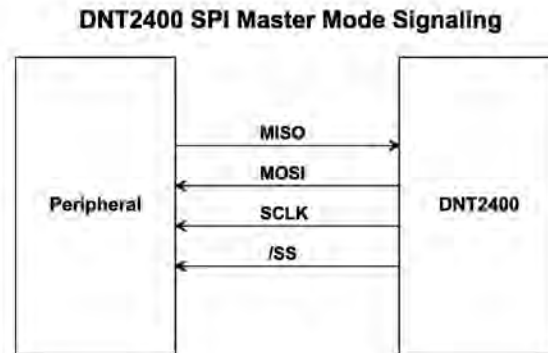


Figure 2.14.4

Figure 2.14.4 shows the signals a DNT2400 uses in SPI Master mode. When periodic I/O reporting is enabled on a DNT2400 remote configured as an SPI Master, the remote will clock out a stored command string, *SPI\_MasterCmdStr*, to collect data from a Slave peripheral each time the I/O report timer fires. The collected data is then transmitted as a *TXData* message. Alternatively, a host connected to the base can transmit an SPI command as a *TXData* message to the remote. The remote will clock the command into its Slave peripheral and transmit back the Slave's response, as show in Figure 2.14.5. In either case, the command string and response string are limited to 32 bytes. Only data messages are routed through the DNT2400's SPI port in Master mode. Command packets and command replies are routed through the primary serial port. See Section 4.2.4 for additional SPI parameter details.



Figure 2.14.5

## 2.15 Sleep Modes

To save power in applications where a remote transmits infrequently, the DNT2400 supports hardware and firmware sleep modes. Hardware sleep mode is entered by switching SLEEP/DTR Pin 11 on the DNT2400 from logic low to high. While in hardware sleep mode, the DNT2400 consumes less than 50  $\mu\text{A}$  at room temperature. This mode allows a DNT2400 to be powered off while its host device remains powered. After leaving hardware sleep mode, the radio must re-synchronize with the base and re-register. Note that sleep mode is not available on tree-routing routers.

In addition to the sleep mode controlled by Pin 11, in CSMA mode the DNT2400 remotes support an additional sleep mode to support battery-powered applications. When this mode is enabled, the DNT2400 is in a low-power state and only wakes up in response to the I/O report triggers. The following list explains the rules that sleeping remotes follow:

- The DNT2400 will wake up when any of the enabled I/O report trigger conditions fire. When any of the ADC triggers are enabled, the radio will also wake up every *ADC\_SampleIntvl* long enough to sample the ADCs, and then go back to sleep.
- When a sleeping radio wakes up, it must acquire and synchronize to its base before it can send or receive any data. To prevent excessive battery use, if the remote is unable to acquire before the *WakeLinkTimeout* elapses (because it is no longer in range, for example), it will cancel any pending event trigger(s) and go back to sleep.
- If a remote is linking for the first time or if its last attempt to acquire and synchronize was unsuccessful, it will scan and record the entire broadcast system parameter list before it goes back to sleep. Otherwise, in order to conserve battery life, a sleeping remote will update any values that it may hear while it is awake, but is not required to listen to the entire list.
- If a remote is linking for the first time or if its last attempt to acquire and synchronize was unsuccessful, it will send a registration request to the base, allowing it to announce its presence to the host. Otherwise, in order to conserve battery life, a sleeping remote will not register each time it reacquires link with its base on successive wakeups.



- After a remote has received an acknowledgement for its I/O report, a *WakeResponseTime* timer is started before the remote goes back to sleep. This allows the base host time to send a message to the remote. Note that the only notification that the base host application has that a remote is awake is its report packet. In order to send it data, the base host must ensure that the message is transmitted and received before the remote's *WakeResponseTime* window elapses. If this function is not needed, the *WakeResponseTime* can be set to zero to disable it.

The lease renewal mechanism is not supported for sleeping remotes. In order to successfully use sleeping remotes, the user must ensure that the system is configured for CSMA mode and that leases are disabled. If these settings are not used, there is no guarantee that the remotes will be able to communicate reliably. Because leases are not supported, there is no built-in mechanism for the base to detect or announce to its host if a remote leaves the network.

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

- If the remote is linking for the first time or was unsuccessful linking on its last attempt, it will remain awake to record the beacon system parameter list.
- At wakeup, the *WakeLinkTimeout* timer is started. If the remote is unable to acquire link before this elapses, it goes back to sleep.
- If the remote receives an acknowledgement for a data packet it has sent (typically an Event packet, but in theory it could be any other type of message), it starts or resets the *WakeResponseTime* timer to remain awake.
- So long as a GPIO for which I/O reporting is enabled for a level trigger remains in its triggered state, the remote will remain awake.
- The remote will remain awake while it still has any ARQ attempts left for a queued transmit packet of any type.
- The remote will remain awake while it has serial characters in its buffer left to transmit to its local host.

Sleep functions are controlled by the following registers (see Section 4.2):

- *SleepMode* - enables/disables sleep mode.
- *WakeResponseTime* - sets the amount of time that a remote will wait for a response after sending an I/O report.
- *WakeLinkTimeout* - sets the maximum time that a remote will spend trying to acquire it base before giving up.

Sleep is also affected by the following registers associated with I/O reporting: *IO\_ReportTrigger*, *IO\_ReportInterval*, *ADC\_SampleIntvl*, and *GPIO\_EdgeTrigger*. The following table indicates how the status and control pins function on sleeping remotes:

Pin	Awake	Sleep
/HOST_RTS	Normal operation	high impedance
/HOST_CTS	Normal operation	0 V
/DCD	Normal operation	0 V
ACT	3 V	0 V
DIVERSITY	Normal operation	0 V
RADIO_TXD	Normal operation	Hi-Z
RADIO_RXD	Normal operation	0 V

Table 2.15.1

Note that the ACT pin may be used by a local host to detect when a sleeping remote is awake. The behavior of the GPIOs during sleep is governed by the *GPIO\_SleepMode*, *GPIO\_SleepDir*, and *GPIO\_SleepState* configuration registers. Refer to the register definitions in Section 4.2.

## 2.16 Encryption

The DNT2400 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.

## 2.17 Synchronizing Co-located Bases

The EX\_SYNC input (Pin 15) on the DNT2400 allows co-located bases to synchronize their transmissions so they all transmit at the same time. This prevents the situation where one base is transmitting while another nearby base is trying to hear a distant remote. Even though the base radios may be on different frequencies, because of their close proximity, the transmitting base can reduce the receiving base's ability to hear distant remotes. The EX\_SYNC input has the following characteristics:

1. Base radios trigger on the rising edge of the pulse applied to their EX\_SYNC input.
2. All co-located bases must use the same hop duration and the period of the pulse train applied to the EX\_SYNC input must be within  $\pm 10 \mu\text{s}$  of this hop duration.
3. The co-located bases must use *different* network IDs but the *same* frequency subband, which assures they are using different hopping patterns.
4. A "narrow" pulse of 50 to 800  $\mu\text{s}$  triggers base beacon synchronization. A train of narrow pulses as described above will synchronize a group of co-located base stations after a period of time.
5. An optional "wide" pulse of 1 to 2 ms triggers a hopping pattern reset. **Note: a wide pulse can only be used in ETSI applications. It is not allowed under FCC and Canadian IC regulations.** The benefit provided by the wide pulse is to keep co-located networks from ever being on the same frequency at the same time. When the wide pulse is not used, it is possible that co-located networks may occasionally try to transmit on the same frequency at the same time. This can slightly reduce the network throughput. Where a wide pulse can be used, it should be

sent to reset all base patterns to their first frequency following the reset or power-up of any of the co-located bases. Thereafter a cycle of  $N - 1$  narrow pulses and then 1 wide pulse should be sent, where  $N$  is the number of frequencies in the subband being used. For example, if the frequency subband contains 15 frequencies, a repeating cycle containing 1 wide pulse followed by 14 narrow pulses should be used.



### 3.0 DNT2400 Hardware

**DNT2400 Block Diagram**

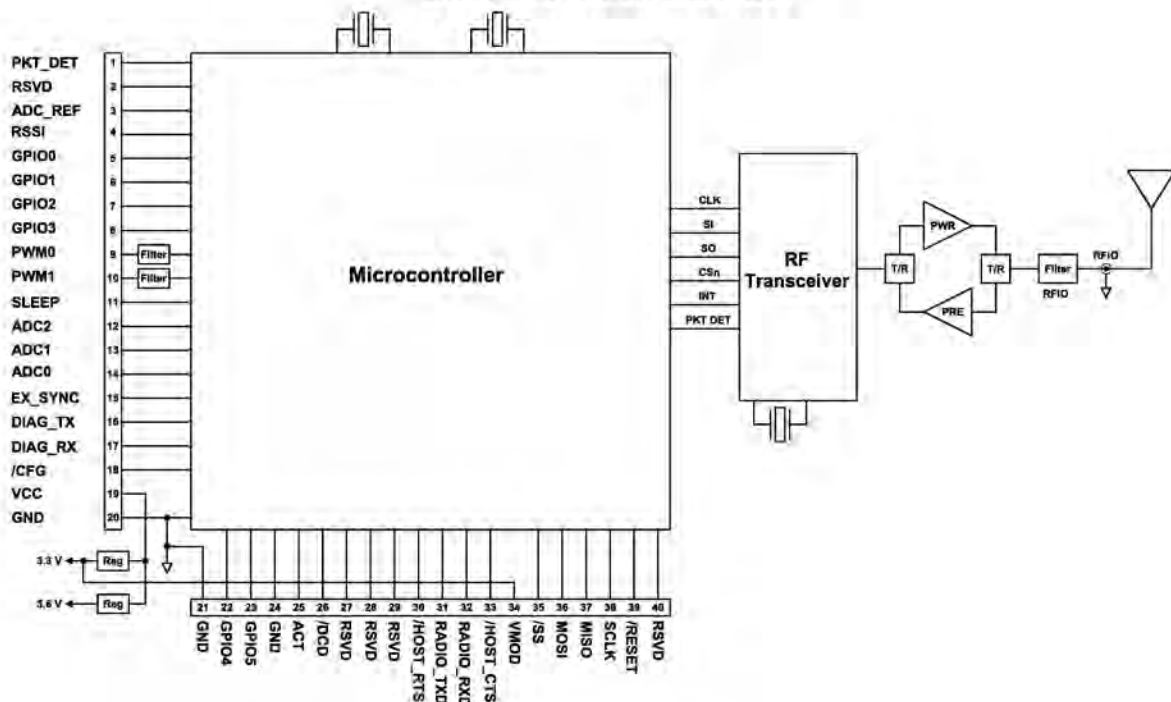


Figure 3.0.1

The major components of the DNT2400 include a 2.4 GHz FHSS transceiver and a low current 32-bit microcontroller. The DNT2400 operates in the 2.4 GHz ISM band. The module includes nine frequency subbands and 37 total frequency channels to support the various 2.4 GHz frequency allocations used throughout the world. The DNT2400 has three selectable RF output power levels: 1, 10, and 63 mW. Also, there are four selectable RF transmission rates: 38.4, 115.2, 200 and 500 kb/s. The DNT2400 includes a low-noise receiver preamplifier and a high efficiency transmitter amplifier, providing excellent range in outdoor applications.

The DNT2400 provides a variety of hardware interfaces. There are two serial ports plus one SPI port. Either the primary serial port or the SPI port can be selected for data communications. The second serial port is dedicated to diagnostics. The primary and diagnostic serial ports support standard baud rates up to 460.8 kb/s. The SPI port supports data rates up to 80.64 kb/s. Also included are three 10-bit ADC inputs, two 8-bit PWM outputs, and six general-purpose digital I/O ports. Four of the digital I/O ports support an optional interrupt-from-sleep mode when configured as inputs. The radio is available in two mounting configurations. The DNT2400C is designed for solder reflow mounting. The DNT2400P is designed for plug-in connector mounting.

### 3.1 DNT2400 Specifications

Characteristic	Sym	Minimum	Typical	Maximum	Units
Operating Frequency Range		2409.3		2467.1	MHz
Hop Dwell Time		5		200	ms
Number of Frequency Hopping Subbands				9	
Number of RF Channels in a Subband		15		37	
Modulation		FSK/MSK			
RF Data Transmission Rates		38.4, 115.2, 200 and 500			kb/s
Receiver Sensitivity:					
10 <sup>-5</sup> BER @ 38.4 kb/s			-104		dBm
10 <sup>-5</sup> BER @ 200 kb/s			-96		dBm
10 <sup>-5</sup> BER @ 500 kb/s			-90		dBm
Transmitter RF Output Power Levels		1, 10, 63			mW
Optimum Antenna Impedance			50		Ω
RF Connection		U.FL Connector			
Network Topologies		Point-to-Point, Point-to-Multipoint Peer-to-Peer, Tree Routing			
Access Schemes		TDMA and CSMA			
Number of Network Nodes:					
TDMA				16	
CSMA				unlimited	
ADC Input Range		0		3.3	V
ADC Input Resolution			10		bits
Signal Source Impedance for ADC Reading				10	KΩ
PWM (DAC) Output Range		0		3.3	V
PWM (DAC) Output Resolution				8	bits
PWM Output Period			20		μs
Primary and Diagnostic 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			kb/s
Serial Peripheral Interface Data Rate		6.35		80.64	kb/s
Digital I/O:					
Logic Low Input Level		-0.5		0.8	V
Logic High Input Level		2		3.3	V
Logic Input Internal Pull-up Resistor		50		200	KΩ
Logic Input Internal Pull-down Resistor		50		180	KΩ
Power Supply Voltage Range	V <sub>DC</sub>	+3.3		+5.5	Vdc
Power Supply Voltage Ripple				10	mV <sub>P-P</sub>
Peak Transmit Mode Current, 63 mW Output				300	mA
Average Operating Receive Current:					
Base			105		mA
Remote, No Data Transmission			35		mA
Remote, 9.6 kb/s Continuous Data Stream			40		mA
Remote, 115.2 kb/s Continuous Data Stream			53		mA
Sleep Current			50	225 <sup>1</sup>	μA
Operating Temperature Range		-40		85	°C
Operating Relative Humidity Range (non condensing)		10		90	%

1. Maximum sleep current occurs at +85 °C

Table 3.1.1



## 3.2 Module Interface

Electrical connections to the DNT2400C are made through the I/O pads and through the I/O pins on the DNT2400P. The hardware I/O functions are detailed in the table below:

Pad	Name	I/O	Description
1	PKT_DET	O	Packet detect output. Signal switches logic high at the end of the start-of-packet symbol and switches logic low at the end of the end-of-packet symbol on both received and transmitted packets. PKT_DET provides a timing reference for use in network timing evaluations, etc.
2	RSVD	-	Reserved pad. Leave unconnected.
3	ADC_REF	I	ADC supply and external full scale reference voltage input. Voltage range is 2.4 to 3.3 Vdc. Connect pad 34 to this input to reference the ADC full scale reading to the module's 3.3 V regulated supply.
4	RSVD	-	Reserved pad. Leave unconnected.
5	GPIO0	I/O	Configurable digital I/O port 0. When configured as an input, an internal pull-up resistor can be selected and interrupt from sleep can be invoked. When configured as an output, the power-on state is also configurable. Sleep mode direction and state are also configurable.
6	GPIO1	I/O	Configurable digital I/O port 1. Same configuration options as GPIO0.
7	GPIO2	I/O	Configurable digital I/O port 2. Same configuration options as GPIO0.
8	GPIO3	I/O	Configurable digital I/O port 3. Same configuration options as GPIO0, with RS485 driver enable an alternate function.
9	PWM0	O	8-bit pulse-width modulated output 0 with internal low-pass filter. Filter is 1 <sup>st</sup> order, 159 Hz 3 dB BW.
10	PWM1	O	8-bit pulse-width modulated output 1 with internal low-pass filter. Filter is 1 <sup>st</sup> order, 159 Hz 3 dB BW.
11	SLEEP/DTR	I	Default functionality is active high module sleep input (active low DTR). When switched low after sleep, the module executes a power-on reset. Usually connected to host DTR.
12	ADC2	I	10-bit ADC input 0. Full scale reading is referenced to the ADC_REF input.
13	ADC1	I	10-bit ADC input 1. Full scale reading is referenced to the ADC_REF input.
14	ADC0	I	10-bit ADC input 2. Full scale reading is referenced to the ADC_REF input.
15	EX_SYNC	I	Rising-edge triggered input for synchronizing co-located bases. Synchronization pulse interval must equal the hop dwell time $\pm 10 \mu\text{s}$ . Pulse width must be in the range of 50 to 800 $\mu\text{s}$ .
16	DIAG_TX	O	Diagnostic UART transmitter output.
17	DIAG_RX	I	Diagnostic UART receiver input.
18	/CFG	I	Protocol selection input. Leave unconnected when using software commands to select transparent/protocol mode (default is transparent mode). Logic low selects protocol mode, logic high selects transparent mode.
19	VCC	I	Power supply input, +3.3 to +5.5 Vdc.
20	GND	-	Power supply and signal ground. Connect to the host circuit board ground.
21	GND	-	Power supply and signal ground. Connect to the host circuit board ground.
22	GPIO4	I/O	Configurable digital I/O port 4. When configured as an input, an internal pull-up resistor can be selected. When configured as an output, the power-on state is configurable. The SPI RX data available flag, SPI_RX_AVL, is a configurable alternate function.
23	GPIO5	I/O	Configurable digital I/O port 5. Same configuration options as GPIO4, with antenna diversity control an alternate function.
24	GND	-	Logic ground.

25	ACT	O	Data activity output, logic high when data is being transmitted or received.
26	/DCD	O	Default functionality is data carrier detect output, which provides a logic low on a remote when the module is locked to FHSS hopping pattern and logic low on a base when at least one remote is connected to it.
27	RSVD	-	Reserved pad. Leave unconnected.
28	RSVD	-	Reserved pad. Leave unconnected.
29	RSVD	-	Reserved pad. Leave unconnected.
30	/HOST_RTS	I	UART flow control input. The host sets this line low to allow data to flow from the DNT2400 on the RADIO_TXD pin. When the host sets this line high, the DNT2400 will stop sending data.
31	RADIO_TXD	O	UART transmitter output. The DNT2400 sends serial data to the host on this pin. The sleep mode state of this pin is configurable.
32	RADIO_RXD	I	UART receiver input. The DNT2400 receives serial data from the host on this pin.
33	/HOST_CTS	O	UART or SPI flow control output. The DNT2400 sets this line low to indicate it is ready to accept data from the host on the RADIO_RXD or MOSI input. When the DNT2400 sets this line high, the host must stop sending data. The sleep mode state of this pin and /DCD is jointly configurable.
34	VMOD	O	Module's +3.3 V regulated supply output. Connect to pad 3 to support 3.3 V full scale and/or ratiometric ADC readings, etc. Current drain on this output should be no greater than 5 mA.
35	/SS	-	SPI active low slave select. This pin is an output when the DNT2400 operating as a master, and an input when it is operating as a slave.
36	MOSI	-	SPI master out, slave in function. This pin is an output when the DNT2400 is operating as a master, and is an input when the DNT2400 is operating as a slave.
37	MISO	-	SPI master in, slave out function. This pin is an input when the DNT2400 is operating as a master, and is an output when the DNT2400 is operating as a slave.
38	SCLK	-	SPI clock signal. This pin is an output when operating as a master, and an input when operating as a slave.
39	/RESET	I	Active low module hardware reset.
40	RSVD	-	Reserved pad. Leave unconnected.

Table 3.2.1

### 3.3 DNT2400 Antenna Connector

A U.FL miniature coaxial connector is provided on both DNT2400 configurations for connection to the RFIO port. A short U.FL coaxial cable can be used to connect the RFIO port directly to an antenna. In this case the antenna should be mounted firmly to avoid stressing the U.FL coaxial cable due to antenna mounting flexure. Alternately, a U.FL coaxial jumper cable can be used to connect the DNT2400 module to a U.FL connector on the host circuit board. The connection between the host circuit board U.FL

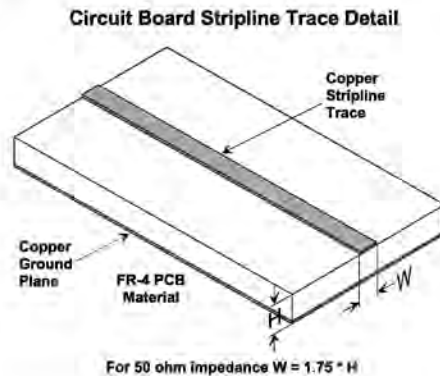


Figure 3.3.1



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

Table 3.3.2

connector and the antenna or antenna connector on the host circuit board should be implemented as a 50 ohm stripline. Referring to Figure 3.3.1, the width of this stripline depends on the thickness of the circuit board between the stripline and the groundplane. For FR-4 type circuit board materials (dielectric constant of 4.7), the width of the stripline is equal to 1.75 times the thickness of the circuit board. Note that other circuit board traces should be spaced away from the stripline to prevent signal coupling, as shown in Table 3.3.2. The stripline trace should be kept short to minimize its insertion loss.

### 3.4 Input Voltages

DNT2400 radio modules can operated from an unregulated DC input (Pad 19) in the range of 3.3 to 5.5 V with a maximum ripple of 5% over the temperature range of -40 to 85 °C. *Applying AC, reverse DC, or a DC voltage outside the range given above can cause damage and/or create a fire and safety hazard. Further, care must be taken so logic inputs applied to the radio stay within the voltage range of 0 to 3.3 V. Signals applied to the analog inputs must be in the range of 0 to ADC\_REF (Pad/Pin 3). Applying a voltage to a logic or analog input outside of its operating range can damage the DNT2400 module.*

### 3.5 ESD and Transient Protection

The DNT2400C and DNT2400P circuit boards are electrostatic discharge (ESD) sensitive. ESD precautions must be observed when handling and installing these components. Installations must be protected from electrical transients on the power supply and I/O lines. This is especially important in outdoor installations, and/or where connections are made to sensors with long leads. *Inadequate transient protection can result in damage and/or create a fire and safety hazard.*

### 3.6 Interfacing to 5 V Logic Systems

All logic signals including the serial ports on the DNT2400 are 3.3 V signals. To interface to 5 V signals, the resistor divider network shown in Figure 3.5.1 below must be placed between the 5 V signal outputs and the DNT2400 signal inputs. The output voltage swing of the DNT2400 3.3 V signals is sufficient to drive 5 V logic inputs.

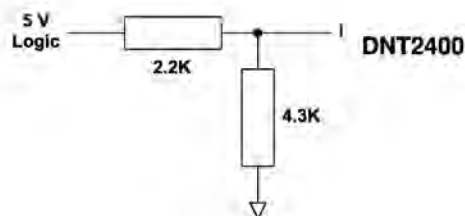


Figure 3.6.1



### 3.7 Power-On Reset Requirements

When applying power to the DNT2400, the /RESET Pin 39 and the RADIO\_TXD Pin 31 must be initially held low. The /RESET pin must be held low until the power supply voltage reaches 3.3 volts for 100 ms, and then set high. The RADIO\_TXD must be held low an additional 10 ms after the /RESET pin goes high. RADIO\_TXD is weakly pulled down with a 100K ohm resistor to meet the power-on reset requirement, unless this line is driven high by an external signal.

### 3.8 Mounting and Enclosures

DNT2400C radio modules are mounted by reflow soldering them to a host circuit board. DNT2400P modules are mounted by plugging their pins into a set of mating connectors on the host circuit board. Refer to Section 8.3 and/or the DNT2400 Data Sheet for DNT2400P connector details.

DNT2400 enclosures must be made of plastics or other materials with low RF attenuation to avoid compromising antenna performance where antennas are internal to the enclosure. Metal enclosures are not suitable for use with internal antennas as they will block antenna radiation and reception. Outdoor enclosures must be water tight, such as a NEMA 4X enclosure.

### 3.9 Labeling and Notices

#### Labeling:

*A clearly visible label is required on the outside of the user's (OEM) enclosure stating the following:*

*"Contains FCC ID: HSW-DNT2400P"*

*"Contains IC: 4492A-DNT2400P"*

#### Notices:

The DNT2400PC and DNT2400PP is qualified for use as a Portable device (body worn) if - and only if - the following conditions are met. Variance from these requirements may result in harmful operation to the user and will violate FCC RF Exposure rules.

1. Antenna Gain - the antenna must be of a type already approved and have a maximum gain of less than 3.47 dBi.
2. The DNT2400 transmit duty cycle may not exceed 15.2% for the device to be used in a Portable application.

This transmit duty cycle must be calculated as follows:

$$Tduty = ((Max\_Bytes + Overhead\_Bytes)*8)/(OTA\_Data\_Rate)/(Tdwel)$$

Where:

*Tduty* - is the resultant transmit duty cycle (in fractions of a unit)

*Tdwell* - is the frequency hopping dwell time (in seconds)

*OTA Data Rate* - is the over the air data rate in bits per second (note - this is not the user interface rate)

*Overhead\_Bytes* - is the number of bytes sent along with the data packet by the DNT2400, equal to 5 bytes

*Max\_Bytes* - is the maximum amount of bytes that can be sent to the DNT2400 from the Host in any *Tdwe/1* number of seconds.

If in doubt about any values entered in this calculation or in the execution of the calculation, please contact Murata Applications for assistance.

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

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

This apparatus complies with Health Canada's Safety Code 6 /ISED RSS 247.

ISED RSS-247 Notice - Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause un- desired operation of the device.

Cet appareil est conforme au Code de sécurité 6 de Santé Canada / ISED RSS 247.

ISED RSS-247 Avis - Le fonctionnement est soumis aux deux conditions suivantes: (1) ce dispositif ne peut pas provoquer d'interférences et (2) cet appareil doit accepter toute interférence, y compris les interférences qui peuvent causer un fonctionnement indésirable de l'appareil.

### **ICES-003**

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

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de Classe B prescrites dans le règlement sur le brouillage radioélectrique édicté par ISED Canada.

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. 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 necessary for successful communication.

This radio transmitter, IC: 4492A-DNT2400P, has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

En vertu des règlements d'Industrie Canada, cet émetteur radio ne peut fonctionner à l'aide d'une antenne d'un type et maximum (ou moins) Gain approuvé pour l'émetteur par Industrie Canada. Pour réduire les interférences radio potentielles pour les autres utilisateurs, le type d'antenne et son gain doivent être choisis de manière que la puissance isotropiquement rayonnée équivalente (e.i.r.p.) ne dépasse pas ce qui est nécessaire pour une communication réussie.

Cet émetteur radio, IC: 4492A-DNT2400P, a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous avec le gain maximal admissible et de l'impédance d'antenne requise pour chaque type d'antenne indiqué. types d'antennes non inclus dans cette liste, ayant un gain supérieur au gain maximum indiqué pour ce type, sont strictement interdits pour une utilisation avec cet appareil.

RFM OMNI249 Omnidirectional Dipole Antenna, 9 dBi, 50 ohms

RFM PA2400 Patch Antenna, 6 dBi, 50 ohms

This device does not comply with ISED exposure limits for portable applications. Any use for portable applications in Canada is strictly prohibited.

Ce dispositif ne respecte pas les limites d'exposition ISED pour les applications portables. Toute utilisation pour les applications portables au Canada est strictement interdite.

Canadian ICES-003 - This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus as set out in the radio interference regulations of Industry Canada.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de Classe B prescrites dans le règlement sur le brouillage radioélectrique édicté par Industrie Canada.



## 4.0 Protocol Messages

### 4.1 Protocol Message Formats

The DNT2400 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 4.1.1

The scale above is in bytes.

The *Start-of-Packet* (SOP) character, 0xFB, is used to distinguish the beginning of a message and to assure synchronization in the event of a glitch on the serial port at startup.

The *Length* byte is defined as the length of the remainder of the message following the length byte itself (or the length of the entire message - 2).

The *Packet Type* (PktType) byte specifies the type of message. It is a bitfield-oriented specifier, decoded as follows:

- Bits 7-6* 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 4.1.2.1 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 4.1.2.1 for specifics.

#### 4.1.1 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 4.1.2.1. 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*, *RxEvt* and *Announce* packets are indicated by 0x20 in the high nibble of the type byte. If multiple arguments are to be provided, they are to be concatenated in the order shown. Little-Endian byte format

is used for all multi-byte arguments, where the lowest order byte is the left-most byte of the argument and the highest order byte in the right-most byte of the argument.

#### 4.1.2 Message Format Details

Com-mand	Reply	Event	Description	Direction	Arguments
0x00	-	-	EnterProtocolMode	from Host	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	-	DiscoverReply	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 4.1.2.1

#### Arguments:

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 4.1.2.1 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:

<u>Status code</u>	<u>Additional fields</u>
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, AvgTxAttempts, 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)

<u>Status codes for error conditions</u>	<u>Additional fields</u>
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)

#### 4.1.3 /CFG Select Pin

A falling edge on the /CFG pin is the equivalent of the *EnterProtocolMode* command. A rising edge on the /CFG pin is the equivalent to sending the *ExitProtocolMode* command. The input to the /CFG pin is debounced to make it compatible with a mechanical switch or jumper.

#### 4.1.4 Flow Control

There are two flow control signals between the radio and the host, /HOST\_RTS and /HOST\_CTS. See Section 2.11 for flow control details.

#### 4.1.5 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 an 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.

#### 4.1.6 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 2.12.1.2, this is the system address of Remote R1.

## 4.2 Configuration Registers

The configuration registers supported by the DNT2400 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.

### 4.2.1 Bank 0 - Transceiver Setup

<u>Bank</u>	<u>Loc'n</u>	<u>Name</u>	<u>R/W</u>	<u>Size in bytes</u>	<u>Range</u>	<u>Default, Options</u>
00	00	DeviceMode	R/W	1	0..3	<b>0 = Remote</b> , 1 = Base, 2 = PTT Remote, 3 = Router
00	01	RF_DataRate	R/W	1	0..4	<b>0 = 500</b> , 1 = 200, 2 = 115.2, 3 = 38.4 kb/s, 0xFF = auto
00	02	HopDuration	R/W	2	4..4000	<b>10 ms (0x00C8)</b>
00	04	InitialParent NwkID	R/W	1	0..255	<b>0xFF = join any network</b>
00	05	SecurityKey	R/W	16	0..2 <sup>128</sup>	<b>all null bytes (0x00) = security disabled</b>
00	15	SleepMode	R/W	1	0..1	<b>0 = off</b> , 1 = timer, 2 = interrupt
00	16	WakeResponseTime	R/W	1	0..255	<b>1 = 50 ms</b> , 0 to disable
00	17	WakeLinkTimeout	R/W	1	0..255	<b>5 s</b>
00	18	TxPower	R/W	1	0..2	<b>0 = 1 mW</b> ; 1 = 10 mW, 2 = 63 mW
00	19	ExtSyncEnable	R/W	1	0..1	<b>0 = off</b> ; 1 = enable
00	1A	DiversityMode	R/W	1	0..2	<b>0 = 0 V</b> , 1 = 3.3 V, 2 = toggle
00	1B	Reserved		1		
00	1C	UserTag	R/W	16		<b>"DNT2400"</b>
00	2C	RegDenialDelay	R/W	2		<b>10 s</b>
00	2E	RmtTransDestAddr	R/W	3		<b>0x000000 (Base)</b>
0x00	0x34	TreeRoutingEn	R/W	1	0..1	<b>0x00 = off</b> , 1 = enable
0x00	0x35	BaseModeNetID	R/W	1	1..63	<b>0xFF</b> (must change on routers)
0x00	0x36	StaticNetAddr	R/W	1	1..126, 255	<b>0xFF = dynamic assignment</b>
0x00	0x37	HeartbeatIntrvl	R/W	2	1..65535	<b>0x14 (20 seconds)</b>
0x00	0x39	TreeRoutingSysID	R/W	1	0..255	<b>0x00</b>
0x00	0x3A	EnableRtAcks	R/W	1	0..1	<b>0x00</b> (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. DNT2400'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 (See Section 4.2.9) 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.

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

*WakeResponseTime* - this parameter sets the length of time that a remote in sleep mode will wait for a response after sending an I/O report before going back to sleep, from a minimum of 10 ms to a maximum of 2.5 seconds in 10 ms units. This time interval is set to allow the base host application to respond to a remote with a packet before the remote returns to sleep. If this parameter is set to 0, the remote will stay awake indefinitely after sending an I/O report. This allows the application as much time as needs for any initial configuration after which it can cause the remote to re-enter sleep by setting this to the operating value. The default value for this parameter is 50 ms.

*WakeLinkTimeout* - this parameter sets the maximum length of time that a remote in sleep mode will spend trying to acquire a link to its base before going back to sleep, from a minimum of 100 ms to 25.5 s



in 100 ms units. If this value is set to 0, the remote will stay awake and continue trying to link to its base indefinitely.

*TxPower* - this parameter sets the transmit power level used by the base (fixed), and the maximum transmit power level used by a remote. A remote may automatically adjust its power below the *TxPower* settings based on the signal strength it receives from the base plus the *TxPower* setting being used by the base. The default 1 mW *TxPower* setting is useful for desktop testing. *TxPower* is usually set to a higher level to increase operating range in actual applications. Setting bit 4 (0001xxxx) disables the remote automatic transmit power function.

0x00 = 0 dBm or 1 mW (default)  
0x01 = 10 dBm or 10 mW  
0x02 = 18 dBm or 63 mW

*ExtSyncEnable* - this parameter enables or disables the hardware EX\_SYNC function on Pin 15. When enabled, a pulse train can be input on Pin 15 to synchronize the transmission of co-located bases. Disabled by default.

*DiversityMode* - the following diversity antenna switching options are supported as an output on GPIO5:

0 = 0 V output (default)  
1 = 3.3 V output  
2 = hop-by-hop toggle between 0 and 3.3 V

*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, 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 *BaseModeNetIDs*. 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