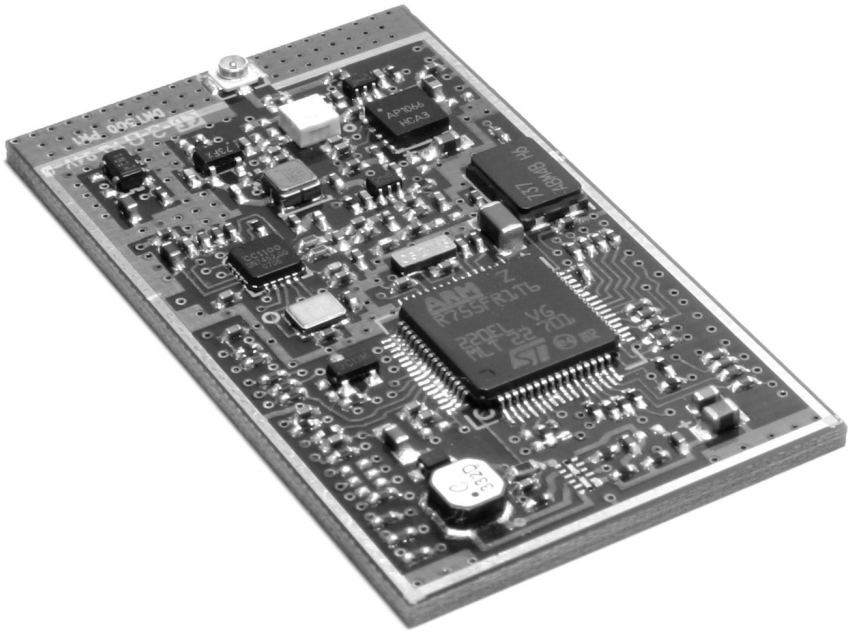
DNT500 Series

900 MHz Spread Spectrum Wireless Industrial Transceivers



Integration Guide

3079 Premiere Pkwy Ste 140
Norcross, Georgia 30097
www.rfm.com
+1 (678) 684-2000

Important Regulatory Information

RFM Product FCC ID: HSW-DNT500P
IC 4492A-DNT500P

Note: This unit 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 when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their expense.

The DNT500 has been designed to operate with the RWA092R 2 dBi reverse-pin (polarity) SMA sleeved dipole antenna (U.FL female to reverse-pin SMA female adaptor or equivalent required).

See section 3.10 of this manual for regulatory notices and labeling requirements. Changes or modifications not expressly approved by RFM may void the user's authority to operate the module.

TABLE OF CONTENTS

1. INTRODUCTION

The DNT500 series transceivers provide highly reliable wireless connectivity for either point-to-point or point-to-multipoint 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 900 MHz ISM band allows license-free use in the US, Canada, Australia and New Zealand. The DNT500 supports all standard serial data rates for host communications from 1.2 to 460.8 kb/s. On-board data buffering and an error-correcting air protocol provide smooth data flow and simplify the task of integration with existing applications. Key DNT500 features include:

- Multipath fading impervious frequency hopping technology with up to 50 frequency channels (902 to 928 MHz).
- Supports point-to-point or multipoint applications.
- Meets FCC rules 15.247 for license-free operation.
- 20 mile plus range with omnidirectional antennas.
- Transparent ARQ protocol with buffering ensures data integrity.
- Selectable 0, 10, 19, 24, 27 or 28 dBm transmit power with a firmware interlock of 19 dBm maximum for 500 kb/s operation.
- Built-in data scrambling reduces possibility of eavesdropping.
- Nonvolatile memory stores configuration when powered off.
- Dynamic TDMA slot assignment that maximizes throughput.
- Simple serial interface handles both data and control at up to 460.8 kb/s.

1.1 Why Spread Spectrum?

A radio transmission 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 through 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 is a problem 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. This means that from a probabilistic viewpoint, a conventional radio system faces a 1% - 2% chance of signal impairment at any given time due to multipath.

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

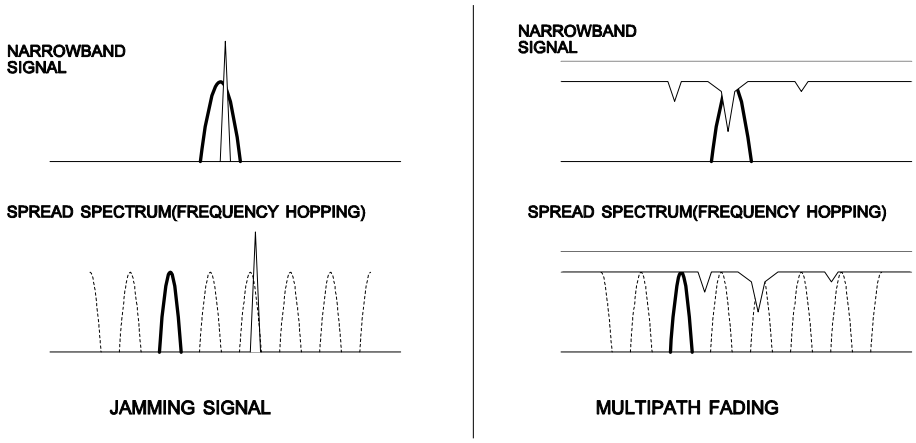


Figure1
Narrowband vs. spread spectrum in the presence of interference

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 narrowband 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 spectrum system.

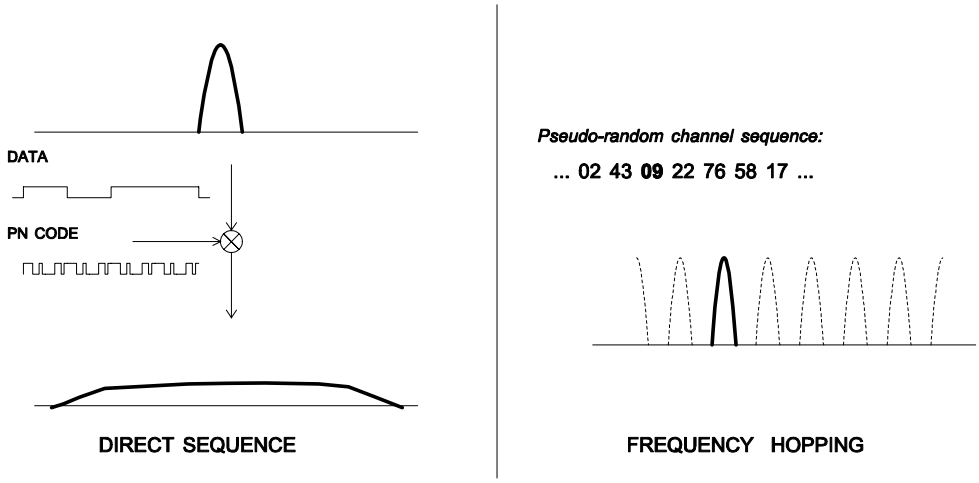


Figure 2
Forms of spread spectrum

One disadvantage of direct sequence systems is that due to spectrum constraints and the design difficulties of broadband receivers, they generally employ only a minimal amount of spreading (typically 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 probing the entire band if necessary to find a channel free of interference. Essentially, this means that a FHSS system will degrade gracefully as the channel 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 tend to offer few non-overlapping channels, whereas multiple hoppers may interleave with less interference. Frequency hopping does carry some disadvantage in that as the transmitter cycles through the hopping pattern it is nearly certain to visit a few blocked channels where no data can be sent. If these channels are the same from trip to trip, they can be memorized and avoided. Unfortunately, this is generally not the case, as it may take several seconds to completely cover the hop sequence during which time the multipath delay profile may have changed substantially. To ensure seamless operation throughout these outages, a hopping radio must be capable of buffering its data until a clear channel can be found. A second consideration of frequency hopping systems is that they require an initial acquisition period during which the receiver must lock on to 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.

DNT500 series modules achieve regulatory certification under FHSS rules at air data rates of 38.4, 115.2 and 200 kb/s. At 500 kb/s, the DNT500 series modules achieve regulatory certification under “digital modulation” or DTS rules. At 500 kb/s DNT500 series modules still employ frequency hopping to mitigate the effects of interference and multipath fading, but hop on fewer, more widely spaced frequencies than at lower data rates.

2. DNT500 RADIO OPERATION

2.1 Network Synchronization and Registration

As discussed above, frequency hopping radios such as the DNT500 periodically change the frequency at which they transmit. In order for the other radios in the network to receive the transmission, they must be listening to the frequency over which the current transmission is being sent. To do this, all the radios in the network must be synchronized and must be set to the same hopping pattern.

In point-to-point or point-to-multipoint networks, one radio module is designated as the base station. All other radios are designated remotes. One of the responsibilities of the base station is to transmit a synchronization signal to the remotes to allow them to synchronize with the base station. Since the remotes know the hopping pattern, once they are synchronized with the base station, they know which frequency to hop to and when. Every time the base station hops to a different frequency, it immediately transmits a synchronizing signal.

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

Once a remote has synchronized with the base station, it will request registration information to allow it to join the network. Registration can be handled automatically by the base station, or it can be controlled by allowing the base station host application to authenticate the remote for registration. When a remote is registered, it receives several network parameters from the base station, including *HopDuration*, *Nwk_ID*, *FrequencyBan*d and *Nwk_Key* (see Section 5.2 for parameter details). Note that if a registration parameter is changed at the base station, it will update the parameter in the remotes over the air.

Among other things, registration allows the tracking of remotes entering and leaving a network, up to a limit of 255 remotes. The base station builds a table of serial numbers of registered remotes using their three-byte serial numbers (MAC addresses). To detect if a remote has gone offline or out of range, the registration is “leased” must be “renewed” once every 250 hops. Any transmission from a remote running on a leased registration will renew its lease with the base station.

2.2 Transparent and Protocol Serial Port Modes

DNT500 radios can work in two serial port data modes: transparent and packet protocol. Transparent formatting is simply the raw user data. Packet protocol formatting uses a 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 systems where the base station needs to send data specifically to each remote, protocol formatting must be used. Protocol formatting is also required for configuration commands and responses, and sensor I/O commands and responses. Protocol formatting details are covered in Section 5.

The DNT500 provides two ways to switch between transparent and protocol modes. If CFG input Pin 18 on the DNT500 is switched from logic high to low, protocol mode is invoked. Or if the ASCII escape sequence "DNT500" is sent (without quotation marks) to the primary serial input following at least a 20 ms pause in data flow, the DNT500 will switch to the protocol mode. When input Pin 18 is switched from logic low to high, or an *ExitProtocolMode* command is sent to the primary serial input, the DNT500 will switch to transparent operation. *Note that if the escape sequence is used to switch to protocol mode, the sequence will be transmitted before protocol mode is invoked.*

When operating in transparent mode, two configuration parameters control when a DNT500 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 both the *MinPacketLength* and the *TxTimeout* parameters is zero, so that any bytes that arrive in the DNT500 transmit buffer will be sent on the next hop. As discussed in Section 2.7.3, it is useful to set these parameters to non-zero values in point-to-multipoint systems where some or all the remotes are in transparent mode.

2.3 RF Data Communications

At the beginning of each hop, the base station transmits a synchronizing signal. After the synchronizing signal has been 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 to non-zero. The maximum amount of user data that the base station can transmit per hop is limited by the *BaseSlotSize* parameter, which has a maximum value of 232 bytes. If there is no user data or reception acknowledgements (ACKs) to be sent on a hop, the base station will only transmit the synchronization signal.

The operation for remotes is similar to the base station, but without the synchronizing signal. The *RemoteSlotSize* parameter sets the maximum number of user bytes a remote can transmit on one hop, up to a limit of 243 bytes per hop. The *RemoteSlotSize* must be coordinated with the *HopDuratio*n and *BaseSlotSize* parameters and the number of registered remotes, as discussed in Section 2.5.3. The *MinPacketLength* and *TxTimeout* parameters operate in a remote in the same manner as in the base station.

2.4 RF Transmission Error Control

The DNT500 supports two error control modes: redundant transmissions and automatic transmission repeats (ARQ). 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 packet. Packet IDs are included in each transmission to allow recipients to identify if the packet is new or has been received before.

In the redundant transmission mode, packets are repeated a fixed number of times without any acknowledgement (ACK) from the recipient. This error control method is useful in latency-critical applications such as voice, video and real-time telemetry, where only a

few transmission repeats can be made before the current data is replaced with new data. It is wasteful of bandwidth to send ACKs in these types of applications. Redundant transmissions are also used where messages are broadcast to multiple recipients and it is not practical to receive ACKs from each one.

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 *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.

2.5 Network Configurations

The DNT500 supports two network configurations: point-to-point and point-to-multipoint. In a point-to-point network, one radio is set up as the base station 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 station acting as the central communications point and all other radios in the network set up as remotes. In this configuration, all communications take place between the base station and any one of the remotes. Remotes cannot communicate directly with each other. It should be noted that point-to-point operation is a subset of the point-to-multipoint operation, so there is no need to specify one or the other.

2.5.1 Point-to-Point Network Operation

Most point-to-point networks act as serial cable replacements and both the base station and the remote use transparent mode. Unless the *MinPacketLength* and *TxTimeout* parameters have been set to non-zero, the base station will send the data in its transmit buffer on each hop, up to the limit set by the *BaseSlotSize* parameter, less ACK bytes. If the base station is buffering more data than can be sent on one hop, the remaining data will be sent on subsequent hops. The base station 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 allows 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 station will retransmit the same data on the next hop.

In point-to-point operation, by default the remote will send the data in its transmit buffer on each hop, up to the limit set by its *RemoteSlotSize* parameter. If desired, the *MinPacketLength* and *TxTimeout* parameters can be set to non-zero values, which configures the remote to wait until the specified amount of data is available or the specified delay had expired before transmitting. 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 station if the base station is in transparent mode. When a transmission is received by the base station, 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.

2.5.2 Point-to-Multipoint Network Operation

In a point-to-multipoint network, the base station 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 the address of the destination (remote) and other overhead bytes to the user data. If the addressed remote is using transparent formatting, the destination address and the other overhead bytes are removed. If the remote is using protocol formatting, the destination 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 station is always the destination. In transparent operation, a remote will add addressing, a packet sequence number and error checking bytes as in a point-to-point network. When the base station 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 will often have the *MinPacketLength* and *TxTimeout* parameters set to non-zero values to reduce the chance of transmission collisions. This is covered in more detail in section 2.6.3.

2.6 Full-Duplex Serial Data Communications

From an host application's perspective, DNT500 serial communications appear full duplex. Both the base station host application and each remote host application can send and receive serial data at the same time. At the radio level, the base station and remotes do not actually transmit at the same time. If they did, the transmissions would collide. As discussed earlier, the base station transmits a synchronization signal at the beginning of each hop followed by its user data. After the base station transmission, the remotes can transmit. Each base station and remote 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 station can receive data from a remote before it completes the transmission of a message to the remote and visa versa.

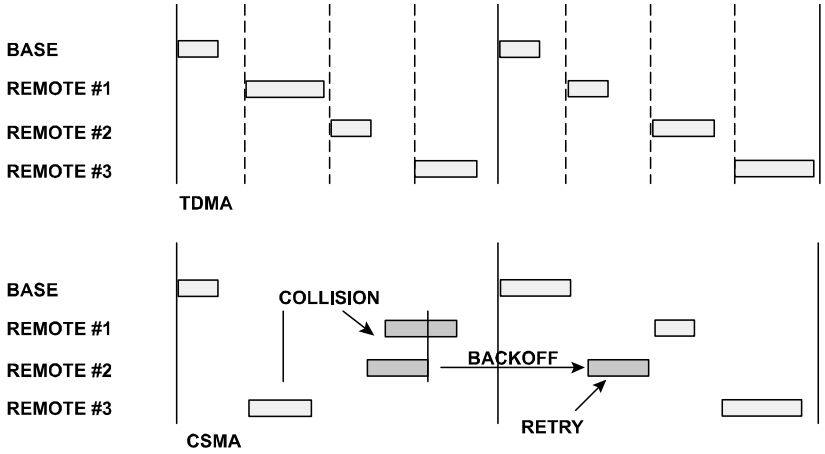
2.7 Channel Access

The DNT500 provides two methods of channel access: CSMA or TDMA. Each method supports several options as shown in the table below. The channel access setting is distributed to all remotes in the base station status packet, so changing it at the base station sets the entire network. 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. The DNT500 includes a CSMA polling mode for coordinated remotes and a CSMA contention mode for uncoordinated and/or reporting remotes. Time Division Multiple Access (TDMA) provides a scheduled time slot for each remote to transmit on each hop. The default DNT500 access mode is CSMA polling.

Access Mode	Description	Max # of Remotes	Remote Slot Size
0	CSMA polling	1024	manual
1	CSMA contention	1024	manual
2 (default)	TDMA dynamic slots	up to 15	automatic
3	TDMA fixed slots	up to 15	automatic
4	TDMA with PTT	1024	automatic

2.7.1 CSMA Modes

When using CSMA, each remote with data to send listens to see if the channel is clear and then transmits. If the channel is not clear, a remote will wait a random period of time and listen again. CSMA works best when a large or variable number of remotes transmit infrequent bursts of data. There is no absolute to the number of remote radios that can be supported in this mode. For a DNT500 network, a maximum of 255 remotes can be supported if base station join-leave tracking is required, or a maximum of 1024 remotes is suggested if base station join-leave tracking is not required. The illustration below compares TDMA to CSMA operation.



There are two important parameters related to CSMA operation. The *CSMA_MaxBackoff* parameter defines the maximum time that a remote will wait after a collision before attempting to send the packet again (back-off interval). The *CSMA_Persistence* parameter sets the probability that a remote will transmit immediately rather than first waiting for a pre-transmit delay interval. Persistence is a one-byte parameter with a range of 0x00 to 0xFF:

0xFF = 100% probability
0x00 = 0% probability

CSMA polling (Mode 0) is used for point-to-point systems and point-to-multipoint systems where only one remote at a time can receive data to transmit (ModBus, etc.). Since only one remote will attempt to transmits at a time, the *CSMA_Persistence* parameter is fixed at 0xFF for minimum latency. This mode provides maximum throughput since there is no contention between remotes and the entire portion of the hop frame following the base station transmission is available for a remote to transmit. The user can set

CSMA_MaxBackoff, *BaseSlotSize* and *RemoteSlotSize* parameters when using this mode. Note that a *CSMA_Persistence* parameter setting of 0xFF would lead to collisions if more than one remote tried to transmit. Applications where more than one remote can receive serial data to transmit at the same time, or where periodic reporting and/or event reporting are enabled should not use this mode.

CSMA Contention (Mode 1) provides classical CSMA channel access, and gives the user control over both the *CSMA_MaxBackoff* and *CSMA_Persistence* parameters. This mode is well-suited for large numbers of uncoordinated remotes, and/or where periodic/event reporting is used. In addition to *CSMA_MaxBackoff* and *CSMA_Persistence*, the user can set the *BaseSlotSize* and *RemoteSlotSize* parameters when using this mode. The following guidelines are suggested for setting *CSMA_Persistence*:

- For lightly loaded CSMA contention networks, increase *CSMA_Persistence* to 0x80 or higher to reduce latency.
- For heavily loaded CSMA contention networks, reduce *CSMA_Persistence* to 0x20 or lower for better throughput.

CSMA modes can optionally track remotes entering and leaving the network for up to 255 remotes. The base station is operated in protocol mode and is configured to generate a CONNECT message for its host when a remote registers, and a DISCONNECT message when the remote's registration lease expires.

The base station in a CSMA network can generate CONNECT messages for more than 255 remotes. This allows the host application to track remotes entering and leaving a large CSMA network by creating a table of MAC addresses and periodically sending a ping to each remote in the table. Failure to answer the ping indicates the remote is no longer active in the network.

The CSMA modes work well in many applications, but CSMA does have 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.

2.7.2 TDMA Modes

The TDMA modes provide guaranteed bandwidth to some or all of the remotes in the network. Remotes that register with the base station receive several special parameters, including ranging information and a specific channel access slot assignment. TDMA registrations are always leased and must be renewed every 250 hops. The DNT500 provides three different modes of TDMA access, 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 remote that registers with the base receives an equal time slice. As new remotes join, the size of the TDMA slots shrink accordingly. The number of slots, individual slot start times, and

the *RemoteSlotSize* are computed automatically by the DNT500 network in this mode. The user should note that the bandwidth to each remote will change immediately as remotes join and leave the network.

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 DNT500 network in this mode. The user must set the number of slots.

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 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).

The slot start time and the *RemoteSlotSize* are computed automatically by the DNT500 network in this mode. The user must specify the number of slots. The last slot is reserved for the PTT remotes. The user must configure PTT remotes individually to select Mode 4 operation. The network makes no guarantee that PTT remote transmissions will not collide in the shared slot. The user's application must ensure that no more than one PTT remote is using the slot at a time.

2.7.3 Network Configuration Planning

Some planning is necessary for a DNT500 network to coordinate the *RF_DataRate, HopDuration*, *BaseSlotSize*, *RemoteSlotSize*, *MinPacketLength, TxTimeout* and *TDMA_MaxNumSlots* parameters to achieve a practical configuration. This is true even for modes that automatically compute some of these parameters. Each parameter has a limited range of usable values, as shown in the table below:

Parameter	Useable Range	Value
RF_DataRate	0..3	500, 200, 115.2 and 38.4 kb/s
HopDuration	40..4095	2..204.75 ms (0.05 ms/count)
TDMA_MaxNumSlots	1..15	max number of TDMA slots (MNS) for remotes
BaseSlotSize	6..232	max number of user data bytes transmitted per hop
RemoteSlotSize	3..243	max number of user data bytes transmitted per hop
MinPacketLength	0..255	0..255 bytes
TxTimeOut	0..255	0..255 ms (1 ms/count)

The highest RF data rate, 500 kb/s, provides the highest throughput and the most flexibility with respect to the other parameters. The maximum RF power that can be used at 500 kb/s is 19 dBm. The three lower data rates can run up to 28 dBm of RF power, and the receiver becomes progressively more sensitive as the data rate is lowered. So for greatest operating range, one of the three lower RF data rates should be used.

The maximum DNT500 *HopDuration* setting is about 200 ms regardless of the RF data rate chosen. For a given data rate, FHSS operation tends to become more robust as hop duration is reduced. However, running with a shorter hop duration may require setting the

BaseSlotSize and *RemoteSlotSize* parameters well below their maximum values at the lower RF data rates. The equation below calculates the minimum hop duration needed at a given RF data rate for a specific number of remote slots and *BaseSlotSize* and *RemoteSlotSize* parameter settings. Support for optimizing a DNT500 configuration for a specific application is also available from RFM's Technical Support Group. See Section 10.3. for technical support contact information.

The minimum required hop duration for a DNT500 configuration is:

$$T_{HD} = T_{BRO} + N_{RS}*T_{RO} + T_{RFB}*(B_{BSS} + N_{RS}*B_{RSS})$$

Where:

T_{HD}	is the minimum required hop duration in milliseconds
T_{BRO}	is the base and registration request overhead time for each hop (RF data rate dependent)
N_{RS}	is the number of remote slots
T_{RO}	is the remote overhead time for each hop (RF data rate dependent)
T_{RFB}	is the transmission time for one user byte (RF data rate dependent)
B_{BSS}	is the *BaseSlotSize* parameter in bytes
B_{RSS}	is the *RemoteSlotSize* parameter in bytes

The constants in the equation for each RF data rate are given in the following table:

RF Data Rate kb/s	**T_{BRO} ms**	**T_{RO} ms**	**T_{RFB} ms**
38.40	11.620	4.817	0.2080
115.2	4.953	2.039	0.0694
200	3.540	1.450	0.0400
500	2.388	0.970	0.0160

For Example 1, consider a point-to-point CSMA Mode 0 system operating at 38.4 kb/s with the *BaseSlotSize* parameter set to 133 bytes and the *RemoteSlotSize* parameter set to 128 bytes. The minimum hop duration needed to support one-hop transmissions of full slot size messages in both directions for this configuration is:

= 11.620 + 1*4.817 + 0.2080*(133 + 1*128)
= 16.437 + 0.2080*261
= 70.725 ms

The closest programmable hop duration is 70.750 ms.

It should be noted that the base station operating system will commandeer 5 bytes from the *BaseSlotSize* allocation in Mode 0 and up to 13 bytes in Mode 1 to send reception acknowledgements (ACKs) back to the remotes. The *BaseSlotSize* should be sized accordingly. In the above example, the *BaseSlotSize* parameter is set five bytes larger than the *RemoteSlotSize* parameter to accommodate the ACK bytes.

When running a point-to-multipoint network with uncoordinated remotes using CSMA Mode 1, it is useful to set N_{RS} to a value of 3 or higher in the equation. Although CSMA

does not create reserved time slots for remotes, extending the hop duration this way allows several uncoordinated transmissions of user data and/or periodic/event reports to arrive in the same slot with a relatively few collisions.

The performance of a CSMA Mode 1 system can often be helped by setting the *MinPacketLength* and *TxTimeout* parameters on any remotes running transparent mode to non-zero values, especially if host messages only contain a few bytes each and transmission latency is not critical. For starting point values, set the *MinPacketLength* equal to the *RemoteSlotSize* and *TxTimeout* to at least three times the hop duration. This will help avoid excessive transmission collisions due to having many packets transmitted, each carrying only a small amount of user data on top of the relatively large packet overhead structure.

For Example 2, consider a TDMA Mode 2 or 3 system operating at 500 kb/s. Up to 10 registered remotes need to be accommodated. A *BaseSlotSize* of 138 bytes is needed, and each remote needs enough slot time to support a *RemoteSlotSize* of 64 bytes. The minimum hop duration needed to support this configuration is:

= 2.388 + 10*0.970 + 0.0160*(138 + 10*64)
= 12.088 + 0.0160*778
= 24.536 ms

The closest programmable hop duration is 24.550 ms.

In all TDMA modes, the base station operating system will commandeer one byte from the *BaseSlotSize* allocation for each registered remote to send ACKs to the remotes. The *BaseSlotSize* and *MinPacketLength* should be sized accordingly.

2.7.4 Serial Port Operation

DNT500 networks are often used for wireless communication of serial data. The DNT500 supports serial baud rates from 1.2 to 460.8 kb/s. Listed in the table 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
38.4	3.84	0.2604
115.2	11.52	0.0868
230.4	23.04	0.0434
460.8	46.08	0.0217

To support continuous full-duplex serial port data flow, an RF data rate much 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 3, consider a CSMA Mode 0 transparent data system operating at 500 kb/s with the *BaseSlotSize* parameter set to 133 bytes (128 bytes net after the five byte allocation for sending ACKs) and the *RemoteSlotSize* parameters set to 128 bytes. The minimum hop duration needed to efficiently support this configuration is:

= 2.388 + 1*0.970 + 0.0160*(133 +1*128)
= 3.358 + 0.0160*261
= 7.534 ms

Setting the hop duration to 7.55 ms, the average full-duplex serial port byte rate that can be supported under error free conditions is:

128 Bytes /7.55 ms = 16.942 kB/s, or 169.42 kb/s for 8N1

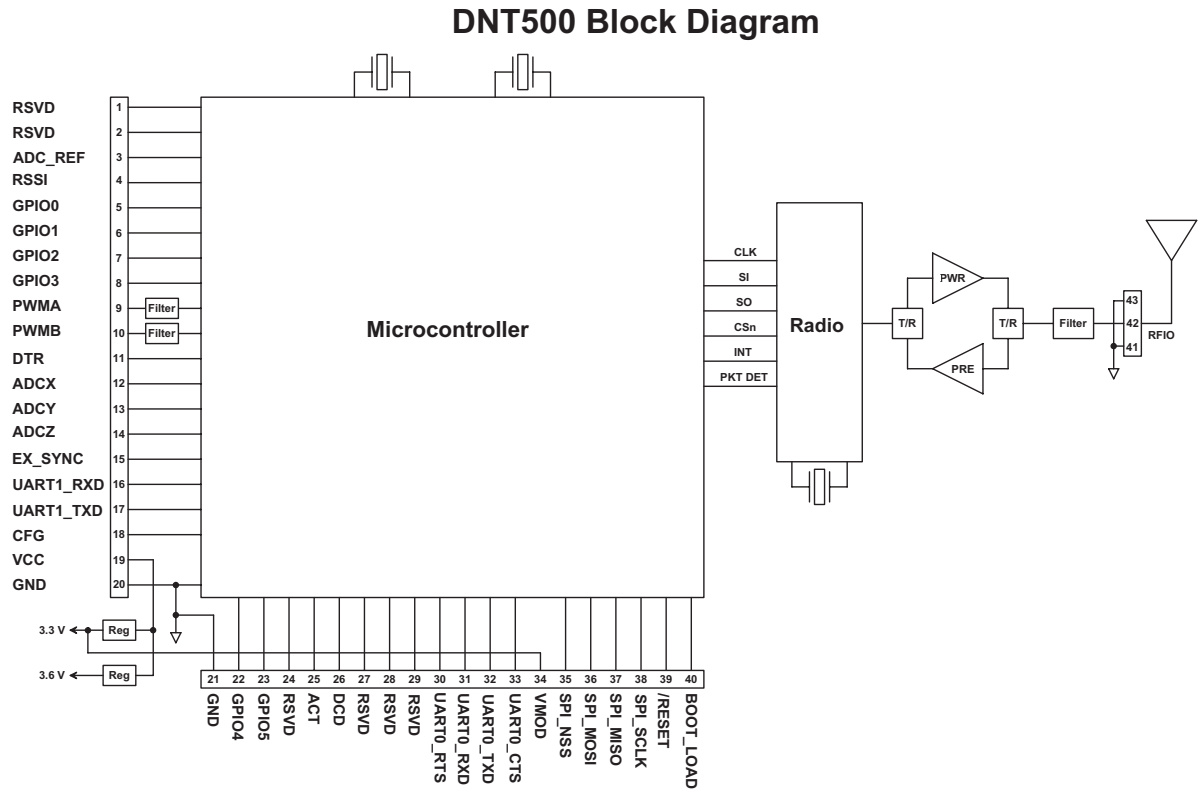
Continuous full-duplex serial port data streams at a baud rate of 115.2 k/bs can be supported by this configuration, provided only occasional RF transmission errors occur. Plan on an average serial port data flow of 75% of the calculated error-free capacity for general-purpose applications, and 50% of the calculated error-free capacity for RF challenging applications such as vehicle telemetry and heavy industrial process environments.

Most applications do not require continuous serial port data flow. The DNT500 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 Example 3 is not exceeded. It is strongly recommended that the DNT500 host use hardware flow control. The host must send no more than 32 bytes additional bytes to the DNT500 when the DNT500 de-asserts the hosts CTS line. In turn, the DNT500 will send no more than one byte following the host de-asserting its RTS line. Three-wire serial port operation is allowed by connecting the DNT500 CTS output to its RTS input. However, three-wire operation should be limited to applications that send small bursts of data occasionally at an average serial port data flow less than 50% of the calculated error-free capacity. Data loss is possible under adverse RF channel conditions when using three-wire serial operation.

2.7.5 Sleep Mode

To save power in applications where a remote transmits infrequently, the DNT500 supports a Sleep Mode. Sleep Mode is entered by switching DTR Pin 11 on the DNT500 from logic low to high. While in Sleep Mode, the DNT500 consumes less than 0.5 mA. This mode allows a DNT500 to be powered off while its host device remains powered. After leaving Sleep Mode (Pin 11 low to high), the radio must re-synchronize with the base station and re-register.

3. DNT500 HARDWARE



The major components of the DNT500 include a 900 MHz FHSS transceiver and a 32-bit microcontroller. The DNT500 operates in the frequency band of 902 to 928 MHz. There are 32 selectable hopping patterns including patterns compatible the frequency allocations in the US, Canadian, Australian and New Zealand. The DNT500 has six selectable RF output power levels: 0, 10, 19, 24, 27 and 28 dBm. Also, there are four selectable RF transmission rates: 38.4, 115.2, 200 and 500 kb/s. The power level is firmware interlocked to a maximum of 19 dBm at 500 kb/s to assure regulatory compliance.

The DNT500 includes a low-noise preamplifier protected by two SAW filters, providing an excellent blend of receiver sensitivity and out-of-band interference rejection.

The DNT500 provides a variety of hardware interfaces. There are two UART serial ports, one for data and a second for diagnostics. The data port supports baud rates from 1.2 to 460.8 kb/s and the diagnostic port supports baud rates from 38.4 to 460.8 kb/s. Other hardware interfaces include an SPI interface, three 10-bit ADC inputs, two 8-bit resolution PWM (DAC) 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. A 3.6 Vdc signal can be switched on the RF output port for diversity antenna control. The radio is available in two mounting configurations. The DNT500 is designed for solder reflow mounting. The DNT500P is designed for plug-in connector mounting.

3.1 Specifications

The DNT500 specifications are listed in the table below:

Characteristic	Sym	Notes	Minimum	Typical	Maximum	Units
Operating Frequency Range			902.75		927.25	MHz
FCC 15.247 FHSS		1	38.4, 115.2 and 200 kb/s, up to 28 dBm			
FCC 15.247 Digital Modulation (DSS)		1	500 kb/s, up to 19 dBm			
Number of Hopping Patterns				32		
Hop Dwell Time			5		200	ms
Number of RF Channels					50	
RF Data Transmission Rates		1	38.4, 115.2, 200 and 500			kb/s
Receiver Sensitivity						
10^{-5} BER @ 38.4 kb/s				-108		dBm
10^{-5} BER @ 500 kb/s				-94		dBm
Transmitter RF Output Power Levels		1	0, 10, 19, 24, 27, 28			dBm
Optimum Antenna Impedance				50		Ω
RF Connection			U.FL Connector or PCB Pad			
Network Topologies			Point-to-Point, Point-to-Multipoint, Mesh			
Access Schemes			TDMA and CSMA			
Number of Network Nodes						
TDMA					15	
CSMA					1024	
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
Primary Serial Port Baud Rates			1.2, 2.4, 4.8, 9.6, 19.2, 38.4, 115.2, 230.4, 460.8			kb/s
Diagnostic Serial Port Baud Rates			38.4, 115.2, 230.4, 460.8			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_{CC}		+3.3		+5.5	Vdc
Power Supply Voltage Ripple					10	mV_{P-P}
Receive Mode Current				50		mA
Transmit Mode Current					900	mA
DTR High Sleep Current					0.5	mA
Operating Temperature Range			-40		85	°C
Operating Relative Humidity Range			10		90	%

1. RF output power is interlocked in firmware to a maximum of 19 dBm at the 500 kb/s RF data rate to assure compliance with regulatory requirements.

3.2 Module Interface

Electrical connections to the DNT500 are made through the I/O pads and through the I/O pins on the DNT500P. The hardware I/O functions are detailed in the table below:

Pad	Name	Description
1	RSVD	Reserved pad. Leave unconnected.
2	RSVD	Reserved pad. Leave unconnected.
3	ADC_REF	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	RSSI	Analog voltage proportional to received signal strength, range 0 to 3.3 V.
5	GPIO0	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.
6	GPIO1	Configurable digital I/O port 1. Same configuration options as GPIO0.
7	GPIO2	Configurable digital I/O port 2. Same configuration options as GPIO0.
8	GPIO3	Configurable digital I/O port 3. Same configuration options as GPIO0.
9	PWM0	Pulse-width modulated output 0 with internal low-pass filter. Provides an 8-bit DAC resolution.
10	PWM1	Pulse-width modulated output 1 with internal low-pass filter. Provides an 8-bit DAC resolution.
11	SLEEP	Default functionality is active high module sleep input. When switched low after sleep, the module executes a power-on reset.
12	ADC0	10-bit ADC input 0. Full scale reading is referenced to the ADC_REF input.
13	ADC1	10-bit ADC input 1. Full scale reading is referenced to the ADC_REF input.
14	ADC2	10-bit ADC input 2. Full scale reading is referenced to the ADC_REF input.
15	EX_SYNC	Optional rising-edge triggered input for synchronizing co-located base stations. See *External-SyncEn* on Page 25 for additional details.
16	DIAG_TX	Diagnostic output (for factory use).
17	DIAG_RX	Diagnostic input (for factory use).
18	/CFG	Protocol selection input. Leave unconnected when using software commands to select trans-parent/protocol mode (default is transparent mode). Logic low selects protocol mode, logic high selects transparent mode.
19	VCC	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	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.
23	GPIO5	Configurable digital I/O port 5. Same configuration options as GPIO4.
24	RSVD	Reserved pad. Leave unconnected.
25	ACT	Data activity output, logic high when data is being transmitted or received.
26	/DCD	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 station when at least one remote is connected to it.

Pad	Name	Description
27	RSVD	Reserved pad. Leave unconnected.
28	RSVD	Reserved pad. Leave unconnected.
29	RSVD	Reserved pad. Leave unconnected.
30	/UART0_RTS	Default functionality is UART flow control input for the module's host. A logic low allows data flow from the module to the host; a logic high blocks data flow from the module to the host.
31	UART0_RXD	UART received data output to host from module.
32	UART0_TXD	UART host data input to be transmitted by module.
33	/UART0_CTS	UART flow control output from the module to the host. A logic low should allow data flow from the host; a logic high should block data flow from the host.
34	VMOD	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	SPI_NSS	Active low SPI enable (slave). This input must be held low for the duration of an SPI transmission.
36	SPI_MOSI	SPI port data output.
37	SPI_MISO	SPI port data input.
38	SPI_SCLK	SPI port clock signal.
39	/RESET	Active low module hardware reset.
40	BOOT_LOAD	Logic high at power up enables module boot loader. Hold low for normal operation.
41	GND	RF ground (DNT500 only). Connect to the host circuit board ground plane.
42	RFIO	Alternate RF port to the U.FL connector (DNT500 only). The antenna can be connected to this port with a 50 Ω stripline or coaxial cable. Leave unconnected when using the U.FL connector.
43	GND	RF ground (DNT500 only). Connect to the host circuit board ground plane.

3.3 Input Voltages

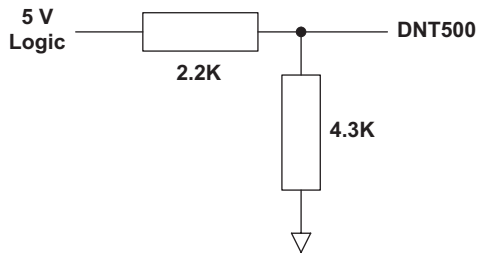
DNT500 radio modules can operated from an unregulated DC input (Pad/Pin 19) in the range of 3.3 (trough) to 5.5 V (peak) with a maximum ripple of 5% over the temperature range of -40 to 85 C. *Applying AC, reverse DC, or a DC voltage outside the range given above can cause damage and/or create a fire and safety hazard. Further, care must be taken so logic inputs applied to the radio stay within the voltage range of 0 to 3.3 V. Signals applied to the analog inputs must be in the range of 0 to ADC_REF(Pad/Pin 3). Applying a voltage to a logic or analog input outside of its operating range can damage the DNT500 module.*

3.4 ESD and Transient Protection

The DNT500 and DNT500P 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.*

3.5 Interfacing to 5 V Logic Systems

All logic signals including the serial ports on the DNT500 are 3.3 V signals. To interface to 5 V signals, the resistor divider network shown below must be placed between the 5 V signal outputs and the DNT500 signal inputs. The output voltage swing of the DNT500 3.3 V signals is sufficient to drive 5 V logic inputs.



3.6 Power-On Reset Requirements

The DNT500 has an internal reset circuit that generates and maintains the DNT500 in a reset state until the power supply voltage reaches 3.3 volts for 100 milliseconds. This reset circuit protects the radio and non-volatile memory from brown-out voltage conditions. If devices that communicate with the DNT500 have shorter reset periods, an allowance must be made to allow the DNT500 to come out of reset. Commands and data sent before the DNT500 is out of reset will be ignored.

3.7 Analog RSSI Output

Pin 4 on the DNT500 provides a 0.3 to 3.0 V output proportional to received signal strength in dB, as follows:

$$V_{RSSI} = 0.03*S_{RF} + 3.6$$

Where:

V_{RSSI} is the RSSI output in volts, over the range of 0.3 to 3.0 V
S_{RF} is the RF signal strength in dBm, over the range of -110 to -20 dBm

The analog RSSI output on a DNT500 remote represents the signal strength of the last base station transmission received. The RSSI output on a base station represents the signal strength of the last remote transmission heard.

3.8 Mounting and Enclosures

DNT500 radio modules are mounted by reflow soldering them to a host circuit board. DNT500P modules are mounted by plugging their pins into a set of mating connectors on the host circuit board. Refer to the DNT500 data sheet for a suitable solder reflow profile and details of the connectors for the DNT500P.

DNT500 radio module enclosures must be made of plastics or other materials with low RF attenuation to avoid compromising antenna performance. Metal enclosures are not suitable as they will block antenna radiation and reception. Outdoor enclosures must be water tight, such as a NEMA 4X enclosure.

3.9 Connecting Antennas

A U.FL miniature coaxial connector is the primary RF connection point on the DNT500, and the only RF connection point on the DNT500P. On the DNT500, it is also possible to connect an antenna using a stripline from pad 42 instead of the U.FL connector. It is important that this connection be implemented as a 50 ohm stripline trace. A short coaxial jumper cable should be used to connect to between the U.FL connector on the DNT500P and the host circuit board U.FL connector. The host PCB U.FL connector should connect to the antenna or antenna connector with a 50 ohm stripline trace. The design details of the stripline are covered in the DNT500 Data Sheet.

3.10 Labeling and Notices

DNT500 FCC Certification - The DNT500 hardware has been certified for operation under FCC Part 15 Rules, Section 15.247. *The antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.*

DNT500 FCC Notices and Labels - *This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.*

A clearly visible label is required on the outside of the user's (OEM) enclosure stating that this product contains a DNT500 transceiver assembly, FCC ID: HSW-DNT500P. WARNING: This device operates under Part 15 of the FCC rules. Any modification to this device, not expressly authorized by RFM, Inc., may void the user's authority to operate this device.

Canadian Department of Communications Industry Notice - IC: 4492A-DNT500P

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

ICES-003

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus as set out in the radio interference regulations of Industry Canada.
Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de Classe B prescrites dans le reglement sur le brouillage radioelectrique edicte par Industrie Canada.

4. PROTOCOL MESSAGES

4.1 Protocol Message Formats

The DNT500 is configured and controlled through a series of protocol mode messages. All protocol mode messages have a common header format:

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

The scale above is in bytes. General comments:

- 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). If the LRC is enabled, it is included in the length.
- 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 the table provided below for reference.
- The LRC is an optional checksum byte that verifies the integrity of the message received. It is the two's complement of the sum of all the bytes in the message. If the sum is larger than 1 byte, only the LSB is used. For example, 0xFB 0x05 0x04 0x01 0x0A 0x00 0x01 → LRC = 0xF0.

  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; please refer to the table for specifics.

4.1.1 Serial 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 the table

below. Event messages from the radio such as receive 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 Bit 4 in the message type. For example, an *EnterProtocolMode* command from the host is message type 0x00, and the *EnterProtocolModeReply* from the radio is message type 0x10. Event messages, including RxData, RxEvent and ANNOUNCE, packets are indicated by setting Bit 5 high in 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 byte of the argument and the highest order byte in the right byte of the argument.

Message Type Code					
Command	**Reply**	**Event**	**Description**	**Direction**	**Arguments**
0x00	-	-	EnterProtocolMode	*from Host*	"DNT500"
-	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
-	-	0x26	RxData	*from Radio*	Addr, RSSI, Data
-	-	0x27	Announce	*from Radio*	AnnStatus, *add'l fields*
-	-	0x28	RxEvent	*from Radio*	Addr, RSSI, Reg, Bank, Span,Val
0x0A	-	-	GetRemoteRegister	*from Host*	Addr, Reg, Bank, Span
-	0x1A	-	GetRemoteRegisterReply	*from Radio*	TxStatus, Addr, RSSI, Reg, Bank, Span, Val*
0x0B	-	-	SetRemoteRegister	*from Host*	Addr, Reg, Bank, Span, Val
-	0x1B	-	SetRemoteRegisterReply	*from Radio*	TxStatus, Addr, RSSI
-	-	0x2F	Instrumentation	*from Radio*	DiagInfo

*If *TxStatus* is non-zero in a *GetRemoteRegisterReply*, the Reg, Bank, Span and Val bytes will not be present in the message

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 register *(see table for size and acceptable range).*

Data = User data *(variable size, 0 to 128 bytes).*
Addr = MAC address of the sender for a reply or event, or recipient for a command *(3 bytes).*
TxStatus = Result of last TxData operation *(1 byte).*
0 = Acknowledgement received.
1 = No acknowledgement received.
2 = (Remote) Not linked.
3 = Recipient holding for flow control.
RSSI = RSSI, range 0x01 to 0xFE. Values 0x00 & 0xFF have special meanings *(1 byte).*
0x00 = No RSSI measured because no ACK was received.
0xFF = No RSSI measured because packet was relayed.
NwkID = Network identifier of network joined *(1 byte).*
BaseMacAddr = MAC address of base that the remote joined *(3 bytes).*
BootSelect = Code indicating whether to do a normal reset or a reset to the bootloader *(1 byte).*
(0 = normal reset, 1 = reset to bootloader)
AnnStatus = Status announcement *(1 byte).*
Additional fields are also reported depending on the status code:

Status code	*Add'l fields*
A0 = Radio has completed startup initialization.	none
A1 = Base: Network formed, ready for data.	NwkID
A2 = Base: A remote has joined me.	MacAddr (0xFF if none)
A3 = Remote: Joined a network, ready for data.	NwkID, BaseMacAddr, Range
A4 = Remote: Exited network (base is out of range).	NwkID
A5 = Remote: Base has restarted.	none
A7 = Base: Remote has left the network.	Addr

Status codes for error conditions	*Add'l fields*
E0 = Protocol error -- invalid message type.	none
E1 = Protocol error -- invalid argument.	none
E2 = Protocol error -- general error.	none
E3 = Protocol error -- parser timeout.	none
E4 = Protocol error -- register is read-only.	none
E8 = UART receive buffer overflow.	none
E9 = UART receive overrun.	none
EA = UART framing error.	none

JoinAddr = MAC address of radio joining *(3 bytes).*
Range = Range measurement of radio joining. *(1 byte).*

4.1.2 Escape sequence

The escape sequence is a sequence of bytes that the user can input in transparent mode to switch the radio to configuration mode. In the DNT500, we define the *EnterProtocol-Mode* command as the ASCII escape sequence “DNT500” (quotation marks are not part of the sequence). A radio that is already in protocol mode will respond to this command the same way as a radio that is in transparent mode. For the escape sequence to be recognized, byte flow must pause at least 20 ms before the escape sequence is sent.

4.1.3 CFG select pin

A falling edge on the CFG pin is the equivalent of entering the escape sequence to invoke the protocol mode. A rising edge on the CFG pin is the equivalent to sending the exit protocol command.

4.1.4 Flow control

There are two flow control signals between the radio and the host, RTS and CTS. See Section 2.7.4 for flow control details.

4.1.5 Protocol mode data message example

For Example 4, ASCII text “Hello World” is sent from the base station 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 station outputs a *TxDataReply* message to its host:

0xFB 0x06 0x15 **0x00** 0x02 0x01 0x00 **0x80**

The 0x00 TxStatus byte value indicates the ACK reception from the remote. The RSSI value is 0x80.

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

0xFB 0x10 **0x26** 0x02 0x01 0x00 **0x8A** 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64

The message is output as an 0x26 event. Note that the RSSI value 0x8A is inserted between the remote’s MAC address and the “Hello World” user data.

4.2 Configuration Registers

The configuration registers supported by the DNT500 are described below. Registers are sorted into banks according to similarity of function.

4.2.1 Bank 0 - Transceiver Setup

Bank	Loc'n	Name	R/W	Size in bytes	Range	Default, Options
00	00	DeviceMode	*R/W*[1]	1	0..2	0 = Remote, 1= Base, 2 = PTT Remote
00	01	RF_DataRate	*R/W*	1	0..4	0 = 500, 1 = 200, 2 = 115.2, 3 = 38.4 kb/s, 0xFF = auto
00	02	HopDuration	*R/W*	2	4..4000	10 ms (0x00C8)
00	04	InitialNwkID	*R/W*	1	0..255	0xFF = broadcast
00	05	SecurityKey	*R/W*	16	$0..2^{128}$	0 = security disabled
00	15	SleepMode	*R/W*	1	0..1	0 = off, 1 = timer, 2 = interrupt
00	16	SleepInterval	*R/W*	2	$0..2^{16}$	0 = off
00	18	TxPower	*R/W*	1	0..5	1 = 10 dBm; 0 = 1, 2 = 19, 3 = 24, 4 = 27, 5 = 28 dBm
00	19	ExternalSyncEn	*R/W*	1	0..1	0 = disabled, 1 = input, 2 = output
00	1A	DiversityMode	*R/W*	1	0..1	0 = 0 V
00	1B	JoinPermit	R/W	1	0..1	0 = no join, 1 = remotes (default), 2 =any
00	1C	UserTag	*R/W*	16		"DNT500"

Note: These settings are individual to each module.

DeviceMode

Selects the operating mode for the radio: remote, base, or PTT remote (listen mostly remote). Note that the setting takes effect immediately.

RF_DataRate

This sets the over-the-air RF data rate. Radios with different RF rates cannot intercommunicate. The following codes are defined:

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

A setting of "auto" will cause a remote to scan all 4 possible RF rates for a network to join. A base set to "auto" will run at the maximum rate of 500 kb/s. A change to this setting on the base will trigger an epoch change to reboot the network.

HopDuration

This sets the duration of the hop frame. The duration is set as a 12-bit value, 0.05 ms/count.

InitialNwkID

Selects the initial network ID that the radio will start (if a base) or join (if a remote). 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 of 0x00.) The network ID also sets the base frequency at which the hopping pattern starts, as illustrated by the following equation:

FrequencyIndex[n] = HoppingPattern[n + 2*NetworkID mod 32]

This allows the user to coordinate frequency spacing of co-located networks to maintain a constant separation as they hop.

SecurityKey

This sets the 128-bit AES encryption key that will be used. The intent is for this to act like a password that all radios in the network are configured with. To protect the key, this is a write-only parameter for the user, unless the manufacturing write enable is set, in which case it is also readable. Refer to the section on Encryption for further information.

SleepMode

Sets the sleep mode.

SleepInterval

Sets the sleep interval as the number of superframes that a remote will sleep between wake intervals. A superframe interval is 64 hops.

TxPower

Sets the transmit power level:

0 = 0 dBm or 1 mW
1 = 10 dBm or 10 mW (default)
2 = 19 dBm or 79.4 mW
3 = 24 dBm or 250 mW
4 = 27 dBm or 500 mW
5 = 28 dBm or 1000 mW (1 W)

When the data rate is set to 500 kb/s, the firmware interlocks the transmit power level to 19 dBm or less to comply with FCC regulations.

ExternalSyncEn

Enables the external sync input. This option allows a base radio's hopframes to be triggered by an external synchronization signal. Valid settings are 0 = disabled, 1 = sync pin is input, 2 = sync pin is an output. This last mode allows a base radio to source the sync signal for another radio.

DiversityMode

The DNT500 supports the following diversity antenna switching options:

0 = 0 V on the diversity pin (default)
1 = 3.3 V on the diversity pin
2 = 0 - 3.3 V toggle on every hop frame

JoinPremit

Valid parameter on a base only. Controls whether remote nodes are permitted to join the base. Parameter values are 0 = no joining permitted, 1 = remotes only may join, 2 = remotes or routers may join (future function).

UserTag

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

4.2.2 Bank 1 - System Settings

Bank	Loc'n	Name	R/W	Size in bytes	Range	Default; Options
01	00	FrequencyBand	*R/W*	1	0..1	0x00 = North America; 0x01 = Australia, 0xFF = auto
01	01	AccessMode	*R/W*	1	0..4	2 = TDMA Dynamic Slots
01	02	BaseSlotSize	*R/W*	1	6..232	20 bytes
01	03	LeasePeriod	*R/W*	1	0..250	5 s (0 to disable)
01	04	ARQ_Mode	*R/W*	1	0..1	0 = ARQ, 1 = redundant TX
01	05	ARQ_AttemptLimit	*R/W*	1	0..16	5 attempts
01	06	TDMA_MaxSlots	*R/W*	1	0..15	8 slots
01	07	CSMA_Persistence	*R/W*	1	0..255	0xFF
01	08	CSMA_MaxBackoff	*R/W*	1	0..255	5 ms
01	09	MaxPropDelay	*R/W*	1	0..255	50 µs (~4 miles)
01	0A	EpochMode	*R/W*	1	0..1	0 = use previous; 1 = increment 2 = random

Note: The base station propagates these setting to all remotes.

FrequencyBand

This sets the range of frequencies over which the radio will operate. Two settings are defined: North America (902-928 MHz) and Australia-New Zealand (915-928 MHz).

AccessMode

This sets the channel access mode that remotes will use to communicate with the base:

Access Mode	Description	Max # of Remotes	Remote Slot Size
0	CSMA polling	1024	manual
1	CSMA contention	1024	manual
2 (default)	TDMA dynamic slots	up to 15	automatic
3	TDMA fixed slots	up to 15	automatic
4	TDMA with PTT	1024	automatic

BaseSlotSize

This sets the size of the base slot. This number is specified in terms of the number of application payload bytes that can be supported. This value excludes the message length byte, the BASE_STATUS packet, the checksum and CRC bytes, and a presumed REG_REPLY packet. The BaseSlotSize indicates the number of bytes that are available for MAC and network ACKs, plus user data packets. This number does not include the

overhead required for headers of these packets, which must be factored into the slot size budget. To ensure that there is never a restriction on when registration or renewal may take place, space for a registration reply packet is always assumed. If this packet is not needed on a given hop, this space is unused.

LeasePeriod

This sets the duration for network address leases that remotes may receive from the base. If a period of zero is specified, then lease and network address functions are disabled.

ARQ_Mode

This sets the ARQ mode for delivery of application messages. In full ARQ mode, an ACK is expected from the receiving radio for each message addressed and sent to it. If no ACK is received, up to *ARQ_AttemptLimit*, efforts to send the data will be made, after which the message is discarded. In redundant transmit mode, each message is sent exactly *ARQ_AttemptLimit* times. No ACKS are sent or expected. This is primarily useful in cases where the number of attempts is 2 or 3, the data flow is time-critical, and the overhead of sending ACKs is not desired.

ARQ_AttemptLimit

This sets the maximum number of attempts that will be made to send a data packet on the RF link. Setting this parameter to the maximum value of 15 is a flag value indicating that there should be no limit to the number of attempts to send each packet (infinite number of attempts). This mode is intended for point-to-point networks in serial data cable replacement applications where absolutely no packets can be lost.

TDMA_MaxNumSlots

In TDMA access modes, this sets the number of slots that are allowed. In fixed slot mode, this allocates the number of slots directly. In dynamic slot mode, this sets the maximum number of slots that may be allocated according to the number of remotes that are registered.

CSMA_Persistence

In CSMA mode, this sets the 'persistence' parameter, or probability that a remote will transmit when it senses an open channel. Please refer to the section on Channel Access for more information.

CSMA_MaxBackoff

In CSMA mode, this sets the maximum length of time that a remote will back off for after a failed transmit attempt. Please refer to the section on Channel Access for more information.

MaxPropDelay

This is the maximum propagation delay that remotes and base will use in their slot timing calculations, in units of microseconds. This is used to pad the amount of time dedicated

to the signup slot. Increasing this value will subtract slightly from the overall slot time available to remotes for sending data.

EpochMode

This is a base-only parameter that governs how the base will select an epoch number at startup when it creates a network. In mode 0, the base will read the epoch number from NVRAM. In mode 1, it reads the value from NVRAM and increments it (and stores the result). In mode 2, it will generate a random epoch number at every startup.

4.2.3 Bank 2 - Status Registers (read only)

Bank	*Loc'n*	*Name*	*R/W*	*Size in bytes*	*Range*	*Default*
02	00	MacAddress	*R*	3	$0..2^{24}$	fixed value
02	03	CurrNwkAddress	*R*	1	0..255	as set
02	04	CurrNwkID	*R*	1	0..255	as set
02	05	CurrRF_DataRate	*R*	1	0..4	as set
02	06	CurrFreqBand	*R*	1	0..1	as set
02	07	LinkStatus	*R*	1	0..1	current status
02	08	RemoteSlotSize	*R*	1	0..255	current size
02	09	TDMA_NumSlots	*R*	1	0..15	as set
02	0A	TDMA_SlotStart	*R*	1	0..255	current setting
02	0B	TDMA_CurrSlot	*R*	1	0..15	current slot
02	0C	HardwareVersion	*R*	1	0..255	0x00 = DNT500 rev A
02	0D	FirmwareVersion	*R*	1	0..255	current firmware load
02	0E	FirmwareBuildNum	*R*	2	$0..2^{16}$	current firmware load
02	0F	Epoch	*R*	1	0..255	as set
02	10	SuperframeCount	*R*	1	0..255	current value
02	11	RSSI_Idle	*R*	1	0..255	as set
02	12	RSSI_Last	*R*	1	0..255	as set
02	14	CurrAttemptLimit	*R*	1	0..255	as set
02	15	CurrRangeDelay	*R*	1	0..255	as set

MacAddress

Returns the radio's unique 24-bit MAC address.

CurrNwkAddress

Returns the radio's current network address, if any. Some special values should be noted. The base always reports 0x00. If a remote does not have a network address, either because lease mode is not enabled or it has not yet received one, it reports 0xFF. This returns the network ID of the network that the radio is currently assigned to or connected to. A value of 0xFF means the radio is in promiscuous mode and scanning for a network but has not yet joined one.

CurrNwkID

This returns the network ID of the network that the radio is currently assigned to or connected to. A value of 0xFF means the radio is in promiscuous mode and scanning for a network but has not yet joined one.

CurrRF_DataRate

This returns the RF data rate of the network that the radio is currently assigned to or connected to. A value of 0xFF means the radio is scanning for a network but has not yet joined one.

CurrFreqBand

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

LinkStatus

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

0 = initializing
1 = unlinked, scanning for network
2 = linked

RemoteSlotSize

This returns the current remote slot size, defined as the maximum possible user data payload size in bytes. In TDMA modes where the slot size is automatically computed, this value is read-only. In manual TDMA mode and CSMA mode, this value must be set by the user. This value excludes the message length byte, REMT_STATUS packet, and checksum and CRC bytes. It indicates the number of bytes available for REMT_DATA and/or REMT_DATA_EXTs. It does not include the overhead bytes required for these packets, which must be figured into the slot size budget.

TDMA_NumSlots

In TDMA access modes, this returns the number of slots currently allocated.

TDMA_CurrSlot

This returns the current TDMA slot number. in modes where the slot position is automatically computed. In modes where this number is not applicable, the value is read as 0xFF.

TDMA_SlotStart

This returns the current TDMA slot position. In TDMA modes where the slot position is automatically computed, this value is read-only.

HardwareVersion

This returns an identifier indicating the type of radio. A value of 0x00 is defined for the DNT500 Rev A hardware.

FirmwareVersion

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

FirmwareBuildNum

This returns the firmware build number, in binary format.

Epoch

Returns the current epoch number.

SuperframeCount

Returns the current superframe count. The superframe counter increments every 64 hops.

RSSI_Idle

Returns the last measurement of RSSI made during a time when the RF channel was idle. May be used to detect interferers. Read-only.

RSSI_Last

Returns the last measurement of RSSI made during receipt of an RF packet with valid CRC. Used for network commissioning and diagnostic purposes.

CurrAttemptLimit

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

CurrRangeDelay

This returns the current propagation delay for this remote as measured from the base.

4.2.4 Bank 3 - Serial

Bank	*Loc'n*	*Name*	*R/W*	*Size in bytes*	*Range*	*Default*
03	00	SerialRate	*R/W*	2	$0..2^{16}$	115.2 kb/s (0x0004)
03	01	SerialParams	*R/W*	1	0..7	8N1
03	03	SerialControls	*R/W*	1	0..7	0X07

SerialRate

This sets the serial rate divisor according to the following formula:

Serial rate in b/s = 460800/SerialRate

Serial rate division setting for commonly used baud rates are:

Setting	*Serial rate*
0x0000	460.8 kb/s
0x0001	460.8 kb/s
0x0002	230.4 kb/s
0x0004	115.2 kb/s (default)
0x0006	76.8 kb/s
0x0008	57.6 kb/s
0x000C	38.4 kb/s
0x0010	28.8 kb/s

0x0018	19.2 kb/s
0x0030	9.6 kb/s
0x0060	4.8 kb/s
0x00C0	2.4 kb/s
0x0180	1.2 kb/s

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

SerialParams

This sets the serial mode options for parity and stop bits:

Setting	*Mode*
0x00	No parity, 8 data bits, 1 stop bit
0x01	No parity, 8 data bits, 2 stop bits
0x02-03	Reserved
0x04	Even parity, 8 data bits, 1 stop bit
0x05	Even parity, 8 data bits, 2 stop bits
0x06	Odd parity, 8 data bits, 1 stop bit
0x07	Odd parity, 8 data bits, 2 stop bits

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

SerialControls

This register affects the way the radio responds to the various serial control lines. Enabling or disabling response to some serial control signals can facilitate communicating with devices that support only a reduced serial interface. The register is defined as a bitmask, with the following options:

bits 7..3 Reserved

bit 2 Base DCD mode.

1 = The base will only assert DCD when at least one remote is registered (default).

0 = The base always asserts DCD, regardless of whether any remotes are attached.

bit 1 RTS enable.

1 = Radio will respond to changes on the RTS control line (default).

0 = Radio ignores the RTS pin and assumes flow control is always asserted.

bit 0 SLEEP enable.

1 = Radio will respond to changes on the SLEEP (DTR) control line (default)

0 = Radio ignores the SLEEP (DTR) pin and is always in the awake state.

4.2.5 Bank 4 - Host Protocol Settings

Bank	*Loc'n*	*Name*	*R/W*	*Size in bytes*	*Range*	*Default; Options*
04	00	ProtocolMode	*R/W*	1	0..1	0 = transparent; 1 = protocol
04	01	ProtocolOptions	*R/W*	1		
04	02	TxTimeout	*R/W*	1	0..255	5 ms
04	03	MinPacketLength	*R/W*	1	0..255	1 byte
04	04	StartupAnnEn	*R/W*	1	0..1	0 = disabled; 1 = enabled

04	05	TransLinkAnnEn	*R/W*	1	0..1	0 = disabled; 1 = LINK announce
04	06	EscapeSequenceEn	*R/W*	1	0..2	0 = enabled; 1 = startup, 2 = anytime
04	07	TransPtToPtMode	*R/W*	1	0..1	0 = multipoint, 1 = point-to-point

ProtocolMode

Selects the host protocol mode. The default is 0, which is transparent mode, meaning the radio conveys whatever characters that are sent to it transparently, without requiring the host to understand or conform to the DNT500's built-in protocol. This setting is recommended for point-to-point applications for legacy applications such as wire replacements where another serial protocol may already exist. Setting this parameter to 1 enables the DNT500 host protocol, which is recommended for point-to-multipoint applications and is preferred for new designs. It is not necessary to define the same protocol mode for all radios in a network. For example, it is frequently useful to configure all the remotes for transparent mode and the base for protocol mode. Note that it is possible for the host to switch the radio from transparent mode to protocol mode and back if desired by transmitting a special "escape sequence" code. See the section on the serial host protocol for more information.

ProtocolOptions

This is a bitmask that selects various options for the protocol mode. Default is 0x01.

bit 7	Enable output of Instrumentation packets.
bits 2..6	reserved
bit 1	Enable LRC checksum byte in serial protocol.
bit 0	Enable output of Announce packets.

TxTimeout

This sets the transmit timeout used for determining message boundaries in transparent data mode. Units are in milliseconds. The default is that a message boundary is determined whenever there is more than a 5 ms gap detected between consecutive characters.

MinPacketLength

This sets the minimum message length used for determining packet boundaries in transparent data mode. The default is one byte.

TransLinkAnnEn

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

EscapeSequenceEn

Enables or disables the escape sequence which can be used to switch from transparent mode to protocol mode. Enabled by default. Valid settings are 0 = disabled, 1 = one chance at startup, 2 = enabled at any time.

TransPtToPtMode

This controls the behavior for addressing packets in transparent mode. When this setting is zero (default), in transparent mode the base will direct packets to the broadcast address. This is useful for point-to-multipoint where the base is sending data to multiple remotes,

for instance in applications where a wireless link is replacing an RS-485 serial bus. When this setting is one, in transparent mode the base will direct packets to the last remote that registered with it. This is useful for point-to-point networks where there are only two endpoints, for instance in applications where a simple serial cable is being replaced.

4.2.6 Bank 5 - I/O Peripheral Registers

Bank	Loc'n	Name	R/W	Size in bytes	Range in bits	Default
05	00	GPIO0	*R/W*	1	1	0
05	01	GPIO1	*R/W*	1	1	0
05	02	GPIO2	*R/W*	1	1	0
05	03	GPIO3	*R/W*	1	1	0
05	04	GPIO2	*R/W*	1	1	0
05	05	GPIO3	*R/W*	1	1	0
05	06	ADC0	*R*	2	10	N/A
05	07	ADC1	*R*	2	10	N/A
05	08	ADC2	*R*	2	10	N/A
05	09	PWM0	*R/W*	2	9	0
05	0A	PWM1	*R/W*	2	9	0

GPIO0..5

Writing to these registers sets the corresponding driver for pins that are enabled outputs. Writing to pins that are enabled as inputs enables or disables the internal pull-up. Reading these registers returns the current level detected on the corresponding pin.

ADC0..2

Read-only, returns the current 10-bit ADC reading for the selected register. See the discussion of the *ADC_SampleIntvl* parameter below.

PWM0..1

Sets the PWM (DAC) outputs. The DC voltage derived from the integrated low-pass filters on the PWM output provides and effective DAC resolution of 8 bits. The range of this parameter is 0x0000 to 0x0100.

4.2.7 Bank 6 - I/O setup

Bank	Loc'n	Name	R/W	Size in bytes	Range in bits	Default; Options
06	00	GPIO_Dir	*R/W*	1	4	0 (all inputs)
06	01	GPIO_Init	*R/W*	1	4	0 (all zeros)
06	02	GPIO_Alt	*R/W*	1	4	0x08 = use GPIO3 for RS485 enable
06	03	GPIO_MessageMode	*R/W*	1	8	GPIO messages disabled
06	04	GPIO_SleepMode	*R/W*	1	1	0 = off; 1 = use sleep I/O states
06	05	GPIO_SleepDDR	*R/W*	1	6	0 (all inputs)
06	06	GPIO_SleepState	*R/W*	1	6	0 (all zeros)
06	07	PWMA_Init	*R/W*	2	10	0x0000
06	09	PWMB_Init	*R/W*	2	10	0x0000
06	0B	ADC_SampleIntvl	*R/W*	2	16	0x0001 (10 ms)
06	0D	ADC0 ThresholdLo	*R/W*	2	10	0x0000

06	0F	ADC0_ThresholdHi	*R/W*	2	10	0x03FF
06	11	ADC1_ThresholdLo	*R/W*	2	10	0x0000
06	13	ADC1_ThresholdHi	*R/W*	2	10	0x03FF
06	15	ADC2_ThresholdLo	*R/W*	2	10	0x0000
06	17	ADC2_ThresholdHi	*R/W*	2	10	0x03FF
06	19	IO_ReportEnable	*R/W*	1	0..1	0 = off
06	1A	IO_ReportInterval	*R/W*	4	32	0x00007530 (every 30,000 hops)
06	1E	IO_ReportAddress	*R/W*	3	24	0x000000

GPIO_DIR

This is a bitmask that sets whether the GPIOs are inputs (0) or outputs (1). The default is all inputs.

GPIO_Init

This is a bitmask that sets the initial value for any GPIOs which are enabled as outputs. For GPIOs enabled as inputs, this sets the initial pull-up setting.

PWM0_Init

This sets the initial value for PWM0 at startup.

PWM1_Init

This sets the initial value for PWM1 at startup.

GPIO_Alt

Provides and alternate function for GPIO3 as an RS-485 driver enable.

GPIO_MessageMode

This register enables a message to be sent to the base station whenever one of the GPIOs is triggered. If the radio is asleep, it will be awakened while the particular GPIO is asserted.

Bit	Option
7-6	Message type for GPIO_3
5-4	Message type for GPIO_2
3-2	Message type for GPIO_1
1-0	Message type for GPIO_0

Message options:

0b00: Disabled	Events are ignored and the radio is not awakened from sleep
0b01: Button Message	An event message is sent reporting the state of the corresponding GPIO input. The state will be 0b0 since I/Os are triggered on logic low.

0b10: Module I/O Message	An event message is sent reporting the states of the entire I/O register bank
0b11: Registration Message	A registration packet is sent to the base station.

GPIO_SleepMode

Enables setting of GPIOs to the designated direction and state whenever a device is asleep.

GPIO_SleepDDR

When GPIO_SleepMode is enabled, the three LSBs of this byte are used to set the direction of the GPIOs during a device's sleep period. This enables the user to provide alternate configurations during sleep that will help minimize current consumption. Bits 0..2 correspond to GPIO0..GPIO2.

GPIO_SleepState

When GPIO_SleepMode is enabled, the three LSBs of this byte are used to set the output state of the GPIOs during a device's sleep period. This enables the user to provide alternate configurations during sleep that will help minimize current consumption. Bits 0..2 correspond to GPIO0..GPIO2.

ADC_SampleIntvl

The *ADC_SampleIntvl* sets the interval between the beginning of one ADC read cycle and the next ADC read cycle. The three ADC inputs are read on each ADC read cycle. An *ADC_SampleIntvl* count equals 10 ms.

ADC0..2_ThresholdLo/Hi

These values define thresholds to trigger an I/O report based on ADC measurements. If I/O reporting is enabled, single EVENT report containing the contents of the I/O bank is generated when a threshold is crossed. Reporting is "edge-triggered" with respect to threshold boundaries, not "level-triggered"; i.e., if the measurement remains there, additional reports are not triggered until the value crosses the threshold again. The thresholds are met whenever one of the following inequalities are satisfied:

ADC*x* < ADC*x*_ThresholdLo
ADC*x* > ADC*x*_ThresholdHi

IO_ReportEnable

When enabled, this causes a remote to periodically send an EVENT message to its base containing the contents of the I/O bank.

IO_ReportInterval

When periodic I/O reporting is enabled, this sets the interval between reports. The default is once every 30000 hops (every 5 minutes, at the default 10 ms hop duration).

IO_ReportAddress

Address to send I/O reports. Usually the base station address.

4.2.8 Bank FF - Special function

This bank contains two user functions, UcReset and MemorySave.

Bank	*Loc'n*	*Name*	*R/W*	*bytes*	*Range*	*Description*
FF	00	UcReset	*W*	1	0..90	00 = reset, 1 = clear status/address and reset, 0x5A = reset with factory defaults
FF	FF	MemorySave	*W*	1	0..1	0 = load factory defaults, 1 = save settings to EEPROM

UcReset

Writing a value of 0x00 to this location forces a software reset of the microcontroller. Writing a value of 0x01 resets the Link Status and erases any assigned Network Address before resetting. Writing a value of 0x5A forces a factory default before resetting. Writing any other value returns an error.

MemorySave

Writing a zero to this location clears all registers back to factory defaults. Writing a one to this location commits the current register settings to EEPROM. When programming registers, all changes are considered temporary until this command is executed.

4.2.9 Protocol Mode Configuration/Sensor Message Examples

For Example 5, the host configures the base station to transmit 24 dBm of RF power using the *SetRegister* command, 0x04. The TxPower parameter is stored in bank 0x00, register 0x18. A one-byte parameter value of 0x03 selects the 24 dBm power level. The protocol formatting for the command is:

0xFB 0x05 0x04 **0x18 0x00 0x01 0x03**

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

0xFB 0x02 **0x17 0x14**

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

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

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

0xFB 0x01 **0x14**

For Example 6, the base station host requests an ADC1 reading from a remote using the *GetRemoteRegister* command, 0x0A. The MAC address of the remote is 0x000102. The current ADC1 measurement is read from register 0x07 in bank 0x05. The ADC reading spans two bytes. The protocol formatting for this command is:

0xFB 0x07 0x0A **0x02 0x01 0x00** 0x07 0x05 0x02

Note the remote MAC address 0x000102 is entered in Little-Endian byte order, 0x02 0x01 0x00. The ADC reading is returned in a *GetRemoteRegisterReply* message:

0xFB 0x0B 0x1A **0x00** 0x02 0x01 0x00 **0x80** 0x07 0x05 0x02 **0xFF 0x02**

Substantial information is returned in the message. The last two byes of the message give the ADC reading in Little-Endian format, 0xFF 0x02. The ADC reading is thus 0x02FF. The RSSI value is the byte following the address, 0x80. The *TxStatus* byte to the right of the *GetRemoteRegisterReply* Packet Type is 0x00, showing the packet was acknowledged on the RF channel.

4.2.10 Protocol Mode Event Message Examples

For Example 4, input GPIO2 (only) is configured to initiate an event message on remote 0x000102. This is done by configuring one-byte parameter *GPIO_MessageMode* with a *SetRemoteRegister* command. *GPIO_MessageMode* is register 0x03 of bank 0x06. Bits 4-5 control GPIO2 event messaging. *Button Message* mode is chosen, which sends the state of GPIO2, located in register 0x02 of bank 0x05, when a high-to-low transition occurs on GPIO2 (*Button Message* always reports a low state). The required *GPIO_MessageMode* bit pattern is 00**01**0000b or 0x10. The protocol formatting for this command is:

0xFB 0x08 0x0B 0x02 0x01 0x00 0x03 0x06 0x01 **0x10**

The *GPIO_MessageMode* parameter is updated and *SetRemoteRegisterReply* is returned:

0xFB 0x06 0x1B **0x00** 0x02 0x01 0x00 **0x8F**

The RSSI value is the byte following the address, 0x8F. The *TxStatus* byte to the right of the *GetRemoteRegisterReply* Packet Type is 0x00, showing the packet was acknowledged on the RF channel. In order for this new *GPIO_MessageMode* setting to persist through a base station power down, *MemorySave* must be invoked, as discussed in Example 5.

When a high-to-low “button push” transition occurs on GPIO2, remote 0x001002 will send the following *RxEvent* message to the base station host:

0xFB 0x09 **0x28** 0x02 0x01 0x00 **0x8E** 0x02 0x05 0x01 0x00

The message is output as a *PktType* 0x28 *RxEvent*. Note that the RSSI value 0x8E is inserted between the remote’s MAC address and the register address byte. Register 0x02 in bank 0x05 uniquely identifies GPIO2 as the source of the event message.

5. DNT500 DEVELOPER'S KIT

Figure 5.1 shows the main contents of a DNT500DK Developer's kit:

Figure 5.1

5.1 DNT500DK Kit Contents

The kit contains the following items:

- Two DNT500P Radios
- Two DNT500 Interface Boards
- Two 9 V Wall Plug Power Suppliers, 120/240 VAC
- Two U.FL RF Jumper Cables
- Two RJ-45 to DB-9F Cable Assemblies
- Two A/B USB Cables
- One RJ-11 to DB-9F Cable Assembly
- Two 900 MHz Dipole Antennas
- One DNT500 Documentation and Software CD

5.2 Additional Items Needed

To operate the kit, the following additional items are needed:

- Two PCs with Microsoft Windows® XP or Vista Operating System

To fully test the kit's functionality, the PCs should be equipped with high-speed serial ports capable of operation at 460.8 kb/s.

5.3 Developer's Kit Default Operating Configuration

The default operating configuration of the DNT500DK developer's kit is TDMA Mode 2, point-to-point, with transparent serial data at 115.2 kb/s, 8N1. One DNT500P is preconfigured as a base station and the other as a remote. The defaults can be overridden to test other operating configurations using the DNT500 Wizard utility discussed in Section 5.5. The default RF power setting is 10 dBm, which is suitable for side-by-side operation. The RF power level should be set higher as needed for longer range operation. Note that setting the RF power to a high level when doing side-by-side testing will overload the DNT500P receiver and cause erratic operation.

5.4 Development Kit Hardware Assembly

Observe ESD precautions when handling the kit circuit boards. Referring to Figure 5.4.1, confirm each DNT500P is correctly plugged into an interface board, with the radio oriented so that its U.FL connector is next to the U.FL connector on the interface board. Check each radio's alignment in the socket on the interface board. No pins should be hanging out over the ends of the connector. Next, install the U.FL jumper cables between the U.FL connectors on the radios and the interface boards. Then install the dipole antennas. As shown in Figure 5.4.2, confirm there is a jumper on pins J14. The interface boards can now be powered by the 9 V wall plug transformer power supplies (the interface boards can also be run for a short time from the 9 V batteries for range testing, etc.).

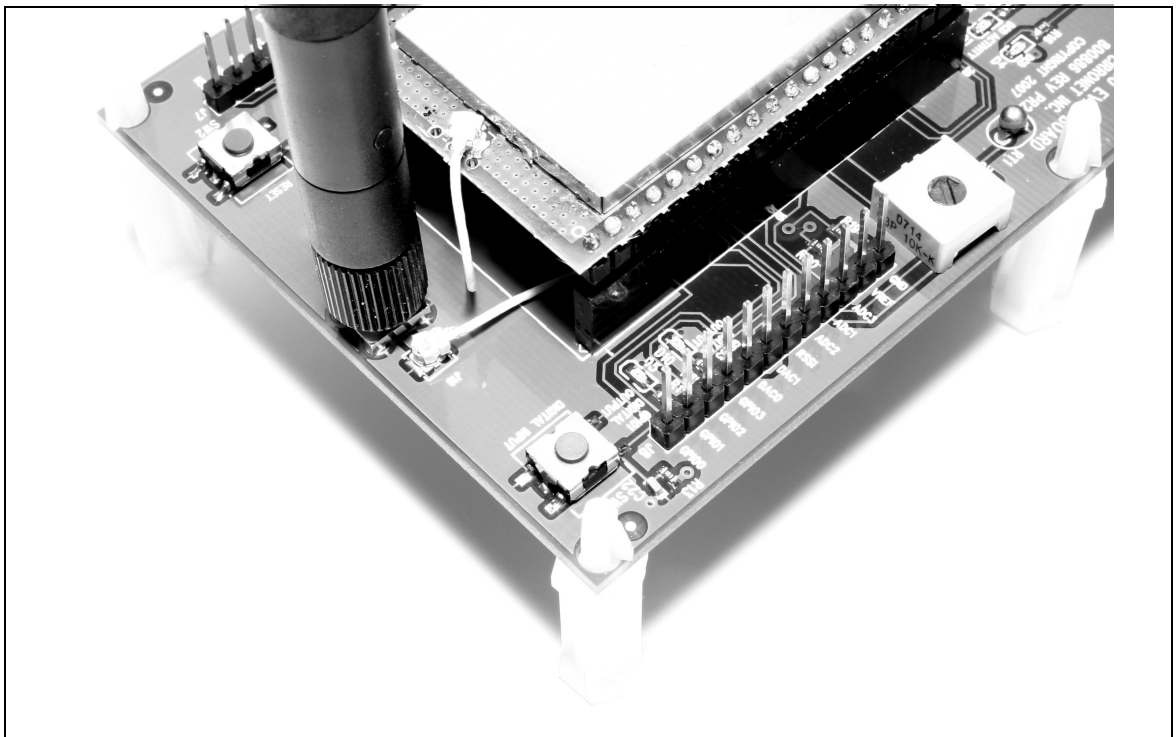


Figure 5.4.1