

NT Series Transceiver Command Data Interface Reference Guide RG-00101



Introduction

The NT Series transceiver has a serial Command Data Interface (CDI) that offers the option to configure and control the transceiver through software instead of through hardware. This interface consists of a standard Universal Asynchronous Receiver Transmitter (UART) with a serial command set. This allows the module to be configured dynamically instead of hardwired, which enables applications such as frequency hopping or listen before talk.

Connecting the Command Data Interface

The CDI uses five lines on the module (Figure 1). These are described in Figure 2.

The CDI is activated by pulling the CMD_DATA_TYPE line low. It uses Command Data In (CMD_DATA_IN) and Command Data Out (CMD_DATA_OUT) as a serial port. The data format follows standard UART conventions of 8 data bits, 1 start bit, 1 stop bit and no parity. The UART voltage levels are mark (1) = V_{CC} and space (0) = GND. The Command Data Baud (CMD_DATA_BAUD) line sets the baud rate of the serial interface to either 9,600bps (low) or 57,600bps (high).

The serial interface is available for use when the transceiver is not in Power Down or Standby mode. The Ready output ($\overline{\text{READY}}$) can be used for simple hardware flow control similar to Clear To Send (CTS). This output is logic low when the transceiver is able to receive command data. If $\overline{\text{READY}}$ is logic high, the transceiver cannot receive commands.

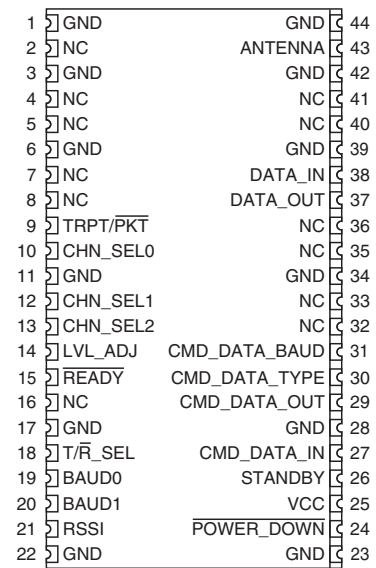


Figure 1: NT Series Transceiver Pinout (Top View)

NT Series Transceiver Pin Descriptions		
Pin #	Name	Description
15	$\overline{\text{READY}}$	Ready. This line is low when the transceiver is ready to communicate and high when its busy. This line can be used for hardware handshaking on the command port.
27	CMD_DATA_IN	Command Data In. This line is the input for command data to set up the module or packet data to be transmitted over the air.
29	CMD_DATA_OUT	Command Data Out. This line outputs the command data responses when setting up the module or packet data that has been received.
30	CMD_DATA_TYPE	Command Data Type. This line sets the serial data as either command data or packet data. Pull low for command data; pull high for packet data.
31	CMD_DATA_BAUD	Command Data Baud. This line sets the baud rate of the command and packet data. Pull low for 9,600bps; pull high for 57,600bps. This line is checked once on power-up.

Figure 2: NT Series Transceiver Pin Descriptions

The Command Data Interface Command Set

The serial message consists of four bytes: Command (read/write/program), Parameter, Value, and Terminator.

Input message:

Command + Parameter + Value + Terminator (0xFF)

A response is returned after a command is received. If the command is correct, the response message is an ACKnowledgement (Command = 0x06) with the Parameter that was received and either the Value that was received for a write Command or the Value that was queried for a read Command. If the Command is incorrect, the response message is a Negative ACKnowledgement (Command = 0x15) with the Parameter that was received and an error code in the Value byte. Every message must have a Terminator byte (0xFF) that identifies the end of the message.

Reply Message:

ACK: Command (0x06) + Parameter Received + Value Received + Terminator (0xFF)

NACK: Command (0x15) + Parameter Received + Error Code + Terminator (0xFF)

The Command byte distinguishes between three different actions.

Read – 0x01: When Command is 0x01, the module reads the current parameter settings. This is useful to poll the module for its current operating status. The Value byte is not used on a read input message, so is set to 0x00. This byte will contain the appropriate value on the output message.

Write – 0x02: When Command is 0x02, the module writes the parameter into temporary memory. Once power is removed from the module, these settings are lost and the default settings are restored on power up. This allows the module to be controlled through software while allowing a return to the factory default state on a power cycle. This is also useful in frequency hopping algorithms where the channel can be changed quickly without having to wait the required time for writing to non-volatile memory.

Program – 0x03: When Command is 0x03, the module programs the parameter into non-volatile memory. These settings are used after the module has been turned off instead of the factory default settings. They can be used as a one-time setup through software or as custom default settings when power is cycled to the module.

The command set uses binary data and not ASCII data.

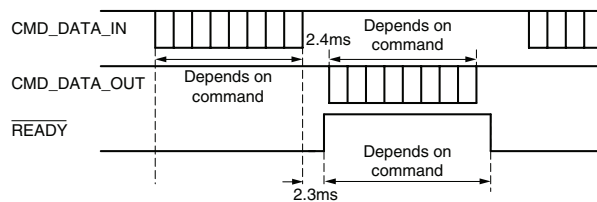


Figure 3: NT Series Transceiver Command Data Interface Timing

The Command Data Interface Commands

The commands available through the serial interface are equivalent to the hardware interface pins. This provides the option of using software and two lines to configure the module instead of hardware and up to seven lines. Figure 4 shows the commands for the Command Data Interface. See Appendix B for more details of the command set.

NT Series Transceiver Command Data Interface Command Set				
Command Data Structure				
Input Message	Command + Parameter + Value + Terminator (0xFF)			
ACK Reply Message	Cmd (0x06) + Rcvd Parameter + Value + Terminator (0xFF)			
NoACK Reply Message	Cmd (0x15) + Rcvd Parameter + Error Code + Terminator (0xFF)			
Function	Cmd	Parameter	Value	Description
Read / Write / Program				
Channel	01 – 03h	10h	00 – 64h	Channel (0 – 100)
Baud Band	01 – 03h	11h	01 – 04h	Baud Band (1 – 4)
TX/RX Mode	01 – 03h	12h	00 – 01h	Transmit (1) / Receive (0)
Level Adjust Value	01 – 03h	13h	00 – 39h	TX output power (0 – 57)
Pin Override Mask	01 – 03h	14h	00 – 0Fh	Software override of hardware settings
Read Only				
Device Name	01h	01h	00h	Returns the 2 character name (NT)
Firmware Version	01h	02h	00h	Returns the 2 byte firmware version
Serial Number	01h	03h	00h	Returns the 4 byte serial number
Get RSSI Value	01h	1Fh	00h	Returns the RSSI value
Write Only				
Return to Default	03h	80h	00h	Return to factory default
Error Codes				
ERR_CMD	15h		F1h	Invalid Command
ERR_CHAN	15h		F2h	Channel value is out of range
ERR_BAUD	15h		F3h	Baud band value is out of range
ERR_RXTX	15h		F4h	RX / TX value is out of range
ERR_LVL_ADJ	15h		F5h	Level Adjust value is out of range
ERR_MASK	15h		F6h	Command is not enabled by the mask

Figure 4: NT Series Transceiver Command Data Interface Command Set

The pin override mask on the following page (Figure 5) enables or disables software control of the configuration settings. It uses the four LSB bits of the Value byte to control the four settings: Channel Selection (bit 0), Baud Band (bit 1), TX/RX Mode (bit 2), and Level Adjust (bit 3). A 0 activates hardware control through the pins; a 1 activates software control. If software control is activated then the states of the hardware lines are ignored. Likewise, software control of a feature is disabled when hardware control is active.

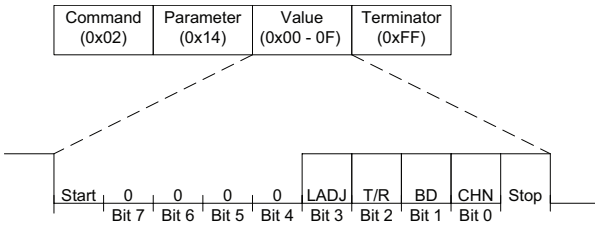


Figure 5: NT Series Transceiver Command Data Interface Timing

The RF channel is set by entering the number of the desired channel. The range of this value depends on the frequency version. The valid range for the 900MHz version is 0 to 100 as shown in Figure 6. The valid range for the 868MHz version is 0 to 67, as shown in Figure 9. This command is the equivalent of the CHN_SEL0, CHN_SEL1 and CHN_SEL2 lines but offers many more channels.

NT Series Transceiver 900MHz Serial Channels							
Channel	Frequency	Channel	Frequency	Channel	Frequency	Channel	Frequency
0	902.62	26	909.12	51	915.37	76	921.62
1	902.87	27	909.37	52	915.62	77	921.87
2	903.12	28	909.62	53	915.87	78	922.12
3	903.37	29	909.87	54	916.12	79	922.37
4	903.62	30	910.12	55	916.37	80	922.62
5	903.87	31	910.37	56	916.62	81	922.87
6	904.12	32	910.62	57	916.87	82	923.12
7	904.37	33	910.87	58	917.12	83	923.37
8	904.62	34	911.12	59	917.37	84	923.62
9	904.87	35	911.37	60	917.62	85	923.87
10	905.12	36	911.62	61	917.87	86	924.12
11	905.37	37	911.87	62	918.12	87	924.37
12	905.62	38	912.12	63	918.37	88	924.62
13	905.87	39	912.37	64	918.62	89	924.87
14	906.12	40	912.62	65	918.87	90	925.12
15	906.37	41	912.87	66	919.12	91	925.37
16	906.62	42	913.12	67	919.37	92	925.62
17	906.87	43	913.37	68	919.62	93	925.87
18	907.12	44	913.62	69	919.87	94	926.12
19	907.37	45	913.87	70	920.12	95	926.37
20	907.62	46	914.12	71	920.37	96	926.62
21	907.87	47	914.37	72	920.62	97	926.87
22	908.12	48	914.62	73	920.87	98	927.12
23	908.37	49	914.87	74	921.12	99	927.37
24	908.62	50	915.12	75	921.37	100	927.62
25	908.87						

Dark Gray = Hardware Selectable Channels

Figure 6: NT Series Transceiver 900MHz Serial Channels

The European regulations are quite different from the North American regulations. The 900MHz rules in North America only specify output power and harmonic levels. The 868MHz rules in Europe subdivide the 863MHz to 870MHz band into sub-bands that are intended for specific applications. The output power varies by sub-band and there is a requirement to either listen before transmitting or comply with a duty cycle limit.

Listen Before Talk (LBT) is simply checking the RSSI level of the channel before transmitting. If the RSSI level is high, the channel is occupied and the system waits until it is clear. The system may also use Adaptive Frequency Agility (AFA) which has the system change channels to avoid ones that are occupied. These features can be implemented in a microcontroller outside the module.

If LBT is not implemented, then the system must comply with duty cycle limits. This refers to the amount of time the transmitter is on per hour. This requires that the transmit time and the total time be tracked. The limits vary with the sub-band.

Figure 9 shows the max output power and duty cycle limits by channel. Figure 7 shows the 868MHz channels that are available through the hardware pin selection and Figure 8 shows the 900MHz channels.

NT Series Transceiver 868MHz Parallel Channels				
CS2	CS1	CS0	CHANEL	FREQUENCY
0	0	0	0	863.15
0	0	1	10	864.15
0	1	0	20	865.15
0	1	1	30	866.15
1	0	0	52	868.35
1	0	1	58	868.95
1	1	0	64	869.55
1	1	1	67	869.85

Figure 7: NT Series Transceiver 868MHz Parallel Channels

NT Series Transceiver 900MHz Parallel Channels				
CS2	CS1	CS0	CHANEL	FREQUENCY
0	0	0	3	903.37
0	0	1	15	906.37
0	1	0	21	907.87
0	1	1	27	909.37
1	0	0	39	912.37
1	0	1	51	915.37
1	1	0	69	919.87
1	1	1	75	921.37

Figure 8: NT Series Transceiver 900MHz Parallel Channels

NT Series Transceiver 868MHz Serial Channels

Channel	Frequency	D.C.	Max Power	Channel	Frequency	D.C.	Max Power
0	863.15	0.10%	25mW	34	866.55	0.10%	25mW
1	863.25	0.10%	25mW	35	866.65	0.10%	25mW
2	863.35	0.10%	25mW	36	866.75	0.10%	25mW
3	863.45	0.10%	25mW	37	866.85	0.10%	25mW
4	863.55	0.10%	25mW	38	866.95	0.10%	25mW
5	863.65	0.10%	25mW	39	867.05	0.10%	25mW
6	863.75	0.10%	25mW	40	867.15	0.10%	25mW
7	863.85	0.10%	25mW	41	867.25	0.10%	25mW
8	863.95	0.10%	25mW	42	867.35	0.10%	25mW
9	864.05	0.10%	25mW	43	867.45	0.10%	25mW
10	864.15	0.10%	25mW	44	867.55	0.10%	25mW
11	864.25	0.10%	25mW	45	867.65	0.10%	25mW
12	864.35	0.10%	25mW	46	867.75	0.10%	25mW
13	864.45	0.10%	25mW	47	867.85	0.10%	25mW
14	864.55	0.10%	25mW	48	867.95	0.10%	25mW
15	864.65	0.10%	25mW	49	868.05	1%	25mW
16	864.75	0.10%	25mW	50	868.15	1%	25mW
17	864.85	0.10%	25mW	51	868.25	1%	25mW
18	864.95	0.10%	25mW	52	868.35	1%	25mW
19	865.05	0.10%	25mW	53	868.45	1%	25mW
20	865.15	0.10%	25mW	54	868.55	1%	25mW
21	865.25	0.10%	25mW	55	868.65	0.10%	25mW
22	865.35	0.10%	25mW	56	868.75	0.10%	25mW
23	865.45	0.10%	25mW	57	868.85	0.10%	25mW
24	865.55	0.10%	25mW	58	868.95	0.10%	25mW
25	865.65	0.10%	25mW	59	869.05	0.10%	25mW
26	865.75	0.10%	25mW	60	869.15	0.10%	25mW
27	865.85	0.10%	25mW	61	869.25	0.10%	25mW
28	865.95	0.10%	25mW	62	869.35	0.10%	25mW
29	866.05	0.10%	25mW	63	869.45	10%	500mW
30	866.15	0.10%	25mW	64	869.55	10%	500mW
31	866.25	0.10%	25mW	65	869.75	None / 1%	5mW / 25mW
32	866.35	0.10%	25mW	66	869.85	None / 1%	5mW / 25mW
33	866.45	0.10%	25mW	67	869.95	None / 1%	5mW / 25mW

Figure 9: NT Series Transceiver 868MHz Serial Channels

The Baud Band command sets the Baud Band for the over-the-air data. The range of this value is 1 to 4 as shown in Figure 10. This command is the equivalent of the BAUD0 and BAUD1 lines.

NT Series Transceiver Baud Selection					
Baud Band	BAUD1	BAUD0	Baud rate (kbps)	IF Bandwidth (kHz)	Typical Receiver Sensitivity (dBm)
1	0	0	1 to 19.2	100	-113
2	0	1	19.2 to 80	150	-110
3	1	0	80 to 160	200	-106
4	1	1	160 to 300	300	-102

Figure 10: NT Series Transceiver Baud Selection

The transceiver has an optional internal packet generator that can be used instead of a custom protocol. Please see the Packet Generator Reference Guide for more information on Packet Mode. In Packet Mode, this value sets the over-the-air baud rate as shown in Figure 11. This does not affect the baud rate of the Command Data Interface. Using a lower baud rate results in better range.

NT Series Transceiver Packet Mode Baud Selection				
Baud Band	BAUD1	BAUD0	Baud rate (kbps)	
1	0	0	19.2	
2	0	1	57.6	
3	1	0	153.6	
4	1	1	300	

Figure 11: NT Series Transceiver Packet Mode Baud Selection

The TX/RX Mode command puts the transceiver into Receive Mode or Transmit Mode. This value is 0 for receive or 1 for transmit. This command is the equivalent of the T/\bar{R} _SEL line. Note that this command is ignored when the transceiver is in Packet Mode.

The Level Adjust value sets the transmitter output power. The range of this value is 0 for the lowest power to 57 for maximum power. This command is the equivalent of the LVL_ADJ line. Figure 12 shows the typical output power for each level.

NT Series Transceiver Level Adjust Value vs. Typical Output Power							
Power Level	P ₀ (dBm)	Power Level	P ₀ (dBm)	Power Level	P ₀ (dBm)	Power Level	P ₀ (dBm)
57	12.22	42	5.38	27	-1.46	12	-8.36
56	12.12	41	4.83	26	-1.84	11	-8.83
55	12.14	40	4.33	25	-2.39	10	-9.39
54	11.86	39	4.05	24	-2.83	9	-9.13
53	11.85	38	3.49	23	-3.27	8	-9.68
52	9.58	37	3.11	22	-3.79	7	-10.23
51	9.78	36	2.77	21	-4.30	6	-10.86
50	8.94	35	2.12	20	-4.85	5	-11.50
49	8.33	34	1.65	19	-5.47	4	-12.23
48	8.02	33	1.16	18	-5.78	3	-13.04
47	7.42	32	0.81	17	-6.12	2	-13.98
46	6.99	31	0.38	16	-6.72	1	-14.59
45	6.72	30	-0.18	15	-7.09	0	-15.78
44	6.33	29	-0.66	14	-7.52		
43	5.80	28	-0.93	13	-7.91		

Figure 12: NT Series Transceiver Level Adjust Value vs. Typical Output Power

The Get RSSI command returns a value that is proportional to the received signal strength. This command is the equivalent of the RSSI line, but offers slightly more resolution. This value is a signed twos complement byte that can be converted into a power level with the following formula:

$$\text{Signal Strength (dBm)} = \text{RSSI Value} - 256$$

The Return to Default command returns the module to the factory settings. Figure 13 shows some examples of the Command Data Interface.

NT Series Transceiver Command Data Interface Examples		
Example Messages Input to the Module		
Read Baud Band	0x01 + 0x11 + 0x00 + 0xFF	Returns the current baud band
Write Baud Band	0x02 + 0x11 + 0x02 + 0xFF	Temporary (RAM) Baud = Baud1
Program Baud Band	0x03 + 0x11 + 0x04 + 0xFF	New (EEPROM) Baud = Baud3
Example Messages Output from the Module		
ACKnowledge Baud Band	0x06 + 0x11 + 0x04 + 0xFF	ACKnowledge Baud = Baud3
No-ACKnowledge Baud Band	0x15 + 0x11 + 0xF3 + 0xFF	Value is out of range

Figure 13: NT Series Transceiver Command Data Interface Examples

Typical Applications

Figure 14 and Figure 15 show two circuits using the CDI.

Figure 14 shows the transceiver set up for transparent data operation. The TRPT/PKT line is pulled high, placing the module into Transparent Mode. The microcontroller monitors the READY line for flow control and to monitor the status of the module. It controls POWER_DOWN and STANDBY for power conservation, though these could be hardwired if power conservation is not necessary for the application. CMD_DATA_IN and CMD_DATA_OUT are connected to the TX and RX lines of the UART in the microcontroller. CMD_DATA_TYPE is hardwired low for command data. CMD_DATA_BAUD is hardwired to the low data rate (9,600bps). DATA_IN and DATA_OUT are connected to GPIOs on a microcontroller that generates and decodes the over-the-air data.

The channel, power level, transmit/receive state and baud band are controlled by the microcontroller through the CDI rather than by the hardware lines. The microcontroller also reads the RSSI level through the CDI.

Figure 15 shows the transceiver set up for packet operation. The TRPT/PKT line is pulled low, placing the module into Packet Mode. CMD_DATA_IN and CMD_DATA_OUT are connected to the TX and RX lines of the UART in the microcontroller. The microcontroller controls the CMD_DATA_TYPE line to toggle between Command Data for changing the module's settings and Packet Data for transmission or reception. It also controls POWER_DOWN and STANDBY for power conservation. The microcontroller monitors the READY line for flow control and to monitor the status of the module. CMD_DATA_BAUD is hardwired to the high data rate (57,600bps).

The channel, power level and baud band are controlled by the microcontroller through the CDI rather than by the hardware lines. The microcontroller also reads the RSSI level through the CDI.

Please see the Wireless UART Reference Guide for more information on Packet Mode.

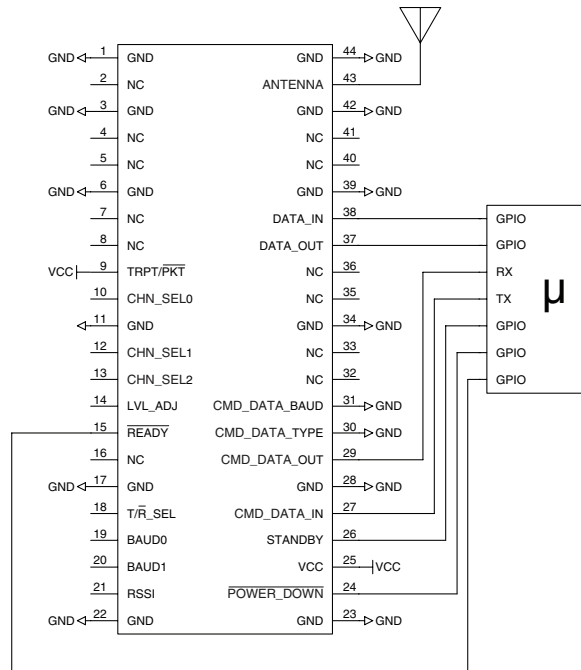


Figure 14: NT Series Transceiver Typical Application Circuits

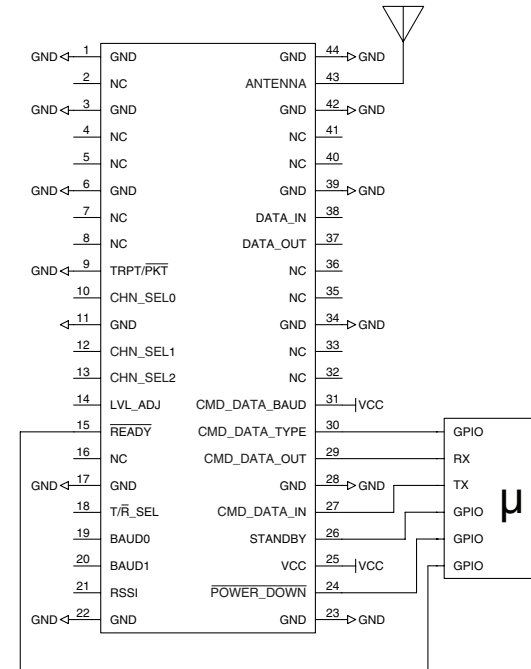


Figure 15: NT Series Transceiver Typical Application Circuits (2)

Appendix A – Sample Firmware

The following sample firmware allows an application to use a frequency agility algorithm to have the receiver listen on an optimal frequency. In this sample, there are three channels used: channel 27 (909.37MHz), channel 51 (915.36MHz) and channel 75 (921.37MHz).

The application periodically polls the receiver's RSSI on each channel, selecting the one with the lowest RSSI. This becomes the receiver's base listening channel. When the application needs to transmit data, it sends the same data message on each of the three channels. When done transmitting, the application puts the module back into receive mode on its base channel.

```
//*****  
/*  
-----  
Overview:   This function reads RSSI from the NT module on each of three  
            channels. The channel with the lowest value for RSSI  
            is selected as the active listening channel.  
  
            The order for serial communication is as follows:  
            1. Set the NT to software control for channel and RX/TX select.  
            2. Set the NT to receive mode.  
            3. Set the NT to each channel frequency while reading RSSI.  
            4. Set the NT to the 1st channel frequency.  
            5. Read RSSI from the NT module.  
            6. If RSSI is lower than the previous RSSI, use this channel.  
            7. Set the NT to the 2nd channel frequency.  
            8. Read RSSI from the NT module.  
            9. If RSSI is lower than the previous RSSI, use this channel.  
            10. Set the NT to the 3rd channel frequency.  
            11. Read RSSI from the NT module.  
            12. If RSSI is lower than the previous RSSI, use this channel.  
            13. Set the NT channel to the quietest frequency selected.  
  
Input:      none  
  
Output:     none  
-----  
*/  
//*****  
void Select_RX_Channel(void)  
{  
    static unsigned int8 i, rx_ch, temp;  
    static unsigned int8 nt_rply[4] = {0,0,0,0};  
    static unsigned int8 channel[3] = {27,51,75};  
    static signed int8 prev_rssi;  
    static signed int8 rssi_val;  
  
    // Initialize the previous RSSI value to 0dBm  
    prev_rssi = 0;  
    // Initialize the RX channel to the 1st channel in the sequence  
    rx_ch = channel[0];  
  
    // Set the NT module for software control of Channel and RX/TX Select.  
    printf("%c%c%c%c", 0x02, 0x14, 0x05, 0xFF);  
    // Flush the NT ACK response by waiting for the terminating character  
    while (getc() != 0xFF);  
  
    // Set the NT module to receive mode  
    printf("%c%c%c%c", 0x02, 0x12, 0x00, 0xFF);  
    // Flush the NT ACK response by waiting for the terminating character
```

```

while (getc() != 0xFF);

// Get RSSI for each channel & assign the receiver to the quietest frequency
for (i=0; i<3; i++)
{
    // Set the NT module to the next channel in sequence
    printf("%c%c%c%c", 0x02, 0x10, channel[i], 0xFF);
    // Flush the NT ACK response by waiting for the terminating character
    while (getc() != 0xFF);

    // Read RSSI from the NT module
    printf("%c%c%c%c", 0x01, 0x1F, 0x00, 0xFF);

    // Get the RSSI value from the NT
    nt_rply[0] = getc();
    nt_rply[1] = getc();
    rssi_val = getc();
    nt_rply[3] = getc();

    // Use the channel with the lowest RSSI value for receive mode
    if (rssi_val < prev_rssi)
    {
        // Assign the quieter channel frequency
        rx_ch = channel[i];
        // Update the previous RSSI value
        prev_rssi = rssi_val;
    }
}

// Set the NT module to the selected channel frequency
printf("%c%c%c%c", 0x02, 0x10, rx_ch, 0xFF);
// Flush the NT ACK response by waiting for the terminating character
while (getc() != 0xFF);
}
/*****

//*****
/*
-----
Overview:   This function allows over-the-air (OTA) application data from pin
            DATA_IN to be transmitted on each of three channels.
            The receiver listens on one of these three channels.
            The idea is to send the same message on each channel so the
            receiver has the opportunity to listen on the quietest channel.

            The order for serial communication is as follows:
            1. Set the NT to software control for channel and RX/TX select.
            2. Set the NT to transmit mode.
            3. Set the NT to the 1st channel frequency
            4. Send the OTA data packet into DATA_IN pin of the NT module.
            5. Set the NT to the 2nd channel frequency
            6. Send the OTA data packet into DATA_IN pin of the NT module.
            7. Set the NT to the 3rd channel frequency
            8. Send the OTA data packet into DATA_IN pin of the NT module.

            NOTE:

            Input:   none

            Output:  none
-----
*/
/*****
void Transmit_Data(void)
{

```

```

static unsigned int8 i;
static unsigned int8 channel[3] = {27,51,75};

// Set the NT module for software control of Channel and RX/TX Select.
printf("%c%c%c%c", 0x02, 0x14, 0x05, 0xFF);

// Flush the NT ACK response by waiting for the terminating character
while (getc() != 0xFF);

// Set the NT module to transmit mode
printf("%c%c%c%c", 0x02, 0x12, 0x01, 0xFF);
// Flush the NT ACK response by waiting for the terminating character
while (getc() != 0xFF);

// Transmit a data packet on each of the 3 channel frequencies
for (i=0; i<3; i++)
{
    // Set the NT module to each channel frequency used in the sequence
    printf("%c%c%c%c", 0x02, 0x10, channel[i], 0xFF);
    // Flush the NT ACK response by waiting for the terminating character
    while (getc() != 0xFF);

    // *** Application sends data packet to NT's DATA_IN pin here ***

}
}
/*****

```

Appendix B – Example CDI Value Bytes

Values to set the channel frequency range from 0 to 100 (0 to 64 hexadecimal) as shown in Figure 16.

NT Series Transceiver Command Data Interface Channel Value Examples		
Channel Number	Value	Description
0	00h	Center frequency is 902.62MHz
3	03h	Center frequency is 903.37MHz (parallel channel 0)
15	0Fh	Center frequency is 906.37MHz (parallel channel 1)
21	15h	Center frequency is 907.87MHz (parallel channel 2)
27	1Bh	Center frequency is 909.37MHz (parallel channel 3)
39	27h	Center frequency is 912.37MHz (parallel channel 4)
51	33h	Center frequency is 915.37MHz (parallel channel 5)
69	45h	Center frequency is 919.87MHz (parallel channel 6)
75	4Bh	Center frequency is 921.37MHz (parallel channel 7)
100	64h	Center frequency is 927.62MHz

Figure 16: NT Series Transceiver Command Data Interface Channel Value Examples

Values to set the Baud Band range from 0 to 3 as shown in Figure 17.

NT Series Transceiver Command Data Interface Baud Band Value Examples			
Baud Band Number	Value	Over-the-Air Baud Rates	
		Transparent Mode	Packet Mode
1	01h	1kbps to 19.2kbps	19.2kbps
2	02h	19.2kbps to 80kbps	57.6kbps
3	03h	80kbps to 160kbps	153.6kbps
4	04h	160kbps to 300kbps	300kbps

Figure 17: NT Series Transceiver Command Data Interface Baud Band Value Examples

NT Series Transceiver Command Data Interface Transmit / Receive Value Examples		
Operation	Value	Description
Receive Mode	00h	Turns on the receiver
Transmit Mode	01h	Turns on the transmitter

Figure 18: NT Series Transceiver Command Data Interface Transmit / Receive Value Examples

NT Series Transceiver Command Data Interface Level Adjust Value Examples		
Operation	Value	Description
Min. TX output power	00h	Typical output power should be -15.78dBm
Transmit Mode	01h	Turns on the transmitter

Figure 19: NT Series Transceiver Command Data Interface Level Adjust Value Examples

The pin override mask tells the module to use either the hardware pins or the CDI commands to configure the channel, baud band, TX/RX mode and output power level. The values range from 0 to 15 (0 to F hexadecimal) and depend on which items are to be controlled with the CDI and which are to be controlled with the hardware pins. Figure 20 shows example values.

NT Series Transceiver Command Data Interface Pin Override Mask Value Examples		
Operation	Value	Description
No bits set	00h	All settings are controlled with the hardware pins
Bit 0 set	01h	Channel is controlled with the CDI and the hardware pin is ignored
Bit 1 set	02h	Baud rate is controlled with the CDI and the hardware pin is ignored
Bit 2 set	04h	TX/RX mode is controlled with the CDI and the hardware pin is ignored
Bit 3 set	08h	Level Adjust is controlled with the CDI and the hardware pin is ignored
Bits 0 and 2 set	05h	Channel and TX/RX mode are controlled with the CDI, others with the hardware pins
All bits set	0Fh	All settings are controlled with the CDI

Figure 20: NT Series Transceiver Command Data Interface Pin Override Mask Value Examples