DM1810 Firmware V2.5 Application Programming

by Frank Perkins

1.0 DM1810 Radio Modules

DM1810 radio modules can be used to add wireless connectivity to a wide range of OEM products. The DM1810-916 series modules are certified for operation under FCC Part 15 and Canadian IC radio regulations, and the DM1810-434 series modules are certified for operation under ETSI SRD radio regulations. DM1810 radio modules are small and operate on very low DC power, making them easy to design into new OEM products and to retrofit into existing OEM designs. The application programming interface (API) for the DM1810 radio modules is also straightforward and capable. This application note discusses the DM1810 firmware V2.5 API, and includes an example application program.

1.1 DM1810 Network Overview

A DM1810 radio network consists of a base station and up to 1023 routers plus field nodes, with a maximum of 15 routers allowed. DM1810 networks are master-slave networks. All radio transmissions either originate from the base station or are sent to the base station. DM1810 application programming, in turn, focuses on messages to and from the base station. Example messages include commands addressed to routers and field nodes, and responses to these commands which verify receipt and/or return data.

A DM1810 network is formed by *binding* field nodes and routers to a base station. Binding can be done during the manufacture of a product that utilizes DM1810 radio modules, or it can be done when the product is placed into service. During binding, the base station sends configuration data to the node being bound, including an *alias ID* (AID) network address.

DM1810 miniMESH™ V2.5 protocol provides distributed network management, greatly simplifying application programming requirements. AID address assignment, network message routing, etc., are handled automatically by the protocol. This allows the application programmer to concentrate directly on the application itself. The only network detail the application programmer will usually be concerned with is the AID address of each node. Beginning with address 001H, AIDs are assigned sequentially as each node is bound (the base station AID is always 000H). The AID of any node can be checked by turning its power off and back on. On power up, a field node or router transmits its AID in a *Power Up* notification message.

There are four hardware *I/O* interfaces on each DM1810 field node; a digital (binary) input, a digital output, an analog-to-digital converter (ADC) input, and a UART. Each hardware interface is supported by a set of configuration options which can be selected by the application program.

2.0 DM1810 API Message Sets

The DM1810 API is composed of six message sets, including one set of *local messages*, and five sets of *network messages* (network messages contain an AID address). The function of each set is outlined below:

Local base station commands and responses - these are two-byte messages going directly between the base station and its host (application processor). Local messages are not sent or received over the radio network. Local messages include two commands and eight response messages.

Read application I/O commands and responses - these application commands and responses allow I/O states, values and serial (UART) messages to be retrieved from a field node.

Write application I/O commands and responses - these application commands and responses allow the digital output to be set, a serial (UART) message to be sent across the network and output at a field node, and the digital input pulse counter to be reset.

Event and notification messages - these messages are sent from a node when certain conditions occur. This allow the host application to receive information on an event without explicitly issuing a command.

Read system configuration commands and responses - details of the network configuration of a node, and details of the I/O configuration of a node can be retrieved with this set of network commands and responses.

Write system configuration commands and responses - certain parts of the network configuration of a node, and all or part of the I/O configuration of a node can be modified with this set of network commands and responses. The rich set of I/O configuration options allows a DM1810 radio module to be adapted to a wide range of applications. For example, there are 12 possible configurations for the digital input, allowing it to be configured to count metering pulses, or to send an event message when a contact opens, etc.

2.1 DM1810 API Message Framing

DM1810 API message structures are simple, consistent and intuitive. API messages are easy to compose and interpret on any type of application host, from a low-cost microcontroller to a PC. All API messages begin with a DLE (10H) byte. Local messages consist of a DLE byte and one command or response byte. All other API messages are network messages, and contain a minimum of six, and up to 70 *core* bytes. Network messages are sent and received using *bi-sync* framing. The procedure to send a core network message with bi-sync framing is as follows:

1. Compute a checksum byte by adding up all the bytes in the core message, disregarding overflow. Add this checksum byte to the end of the core message.

2. Send the bi-sync start of frame bytes, DLE + STX (10H + 02H).

3. Send each byte of the core message and then the checksum byte. If a DLE (10H) byte occurs in the core message or the checksum, send the byte followed by a second DLE “escape” byte.

4. Send the bi-sync end of frame bytes, DLE + ETX (10H + 03H).

To extract a message from bi-sync framing, reverse the above procedure:

1. Discard the bi-sync start of frame DLE + STX bytes.

2. Collect each byte of the core message, plus the checksum byte. If two DLE bytes occur in a row, discard the second DLE “escape” byte.

3. Discard the bi-sync end of frame DLE + ETX bytes.

4. Compute a checksum byte on the received core message and compare its value to the received checksum byte.

3.0 DM1810 Local Message Structures

Figure 3.0.1 below details the local commands and responses. All command and response bytes is this table and the following tables are in *hexadecimal* format. Any command sent to the base station will receive *either a local response or a network response*. Application programs should *wait for a response to one command before sending the next command*. It is only necessary for an application program to include a five second response time out to cover contingencies such as a disconnected base station.

Byte Position	1	2
Base Reset Command	10	04
Base AutoSend Command	10	16
Command Accepted	10	30
Base Reset Complete	10	04
Command No Op	10	31
Response Timeout	10	32
Network Timeout	10	33
Response Error	10	34
Serial (UART) Error	10	35
Invalid Length	10	36
Invalid DLE Sequence	10	37

Figure 3.0.1

For example, if a network command is sent to a field node and the base station hears it being routed by one or more repeaters but no response is returned, the base station will send a local DLE + '2' (10H + 32H) *Response Timeout* message.

Base Reset Command - this command resets the base station software. It does not alter the base station configuration. The *Base Reset Command* receives a sequence of two responses, a *Command Accepted* followed by a *Base Reset Complete*, as discussed below.

Base AutoSend Command - this command causes the base station to begin automatically sending diagnostic "ping" packets to field node addresses 1, 2 and 3.

Base Reset Complete - this response has the same format as the *Base Reset Command*, but is sent from the base station to the host.

Command Accepted - this is the local response generated for a local command.

Command No Op - this local response is generated for a null network command, such as DLE + STX + DLE + ETX.

Response Timeout - if a network command is sent to a field node and the base station hears it being routed by one or more repeaters, but no response is returned within a computed timeout period, the base station returns this local response to the host.

Network Timeout - if a network command is sent to a field node and the base station does not hear it being routed and no response is returned within a computed timeout period, the base station returns this local response to the host.

Response Error - if a network response is received with a bad checksum, the wrong length, etc., the base station returns this local response to the host.

Serial (UART) Error - this response is returned to the host when the base station receives a message with a framing error, etc.

Invalid Length - this response is returned to the host when the base station receives a network command that has the wrong number of bytes.

Invalid DLE Sequence - this response is returned to the host when the base station receives a message with an invalid DLE sequence.

3.1 DM1810 Network Message Structures

The basic structure of all network messages is shown in the table below. The interpretation of the first six bytes of a network message is uniform.

Byte Position	1	2	3	4	5	6	7 through 70
Network Command	t0	00	00	0a	aa	uc	payload
Network Response	t0	00	00	0a	aa	sc	payload

Figure 3.1.1

The upper four bits *(t)* of byte 1 indicate the network *message type*. Here is the coding for the *t* bits:

Message Type	Bit Pattern (t)
Router/Field Node System Configuration Command	0111
Base Station System Configuration Command	1111
System Configuration Response	1111
Application I/O Command	0110
Application I/O Response	1110
Event/Notification Message	1111

Figure 3.1.2

Event and notification messages are *implied* responses to an earlier configuration command, and have the same type coding as a configuration response.

The lower four bits of byte 1, all bits in bytes 2 and 3, and the upper six bits of byte 4 are used by the miniMESH protocol for message routing. These bits should be set to zero when composing a network command. The routing bit locations in a network response will not be all zeros. In application programs, however, it rarely necessary to interpret the routing bit values in response messages.

The lower two bits of byte 4 and all eight bits in byte 5 *(aaa)* form the 10-bit AID address of a network command, or the AID address of a response or event message. The AID address must be set when composing a network command message, and considered when interpreting a network response message.

Byte 6 is referred to as the *argument* byte. The upper four bits *(u)* indicate the *end user type* in a command message, and the *status* in a response message. Here is the coding for the *u* bits:

Application Command End User Type	Bit Pattern (u)
Digital Input	0001
ADC, 8-bit Resolution	0010
Digital Output	0011
UART Buffer	0100
ADC, 10-bit Resolution	1010
Event Flags	1110
Power Management (Dynamic)	1111

Figure 3.1.3

System Command End User Type	Bit Pattern (u)
Overall Network Configuration	0001
Overall I/O Configuration	0010
Bind Configuration	0011
Base Station Bind List	0100
Digital Input Configuration	0101
ADC Configuration	0110
Digital Output Configuration	0111
UART Configuration	1000
Timer Configuration	1001
System List	1010
User A Message String	1011
User B Message String	1100
Node Configuration	1101
Firmware Version	1110
32-bit Hardware ID	1111
Response Status	**Bit Pattern (s)**
Valid Response	0000
Invalid Response	0100

Figure 3.1.4

The lower 4 bits *(c)* in byte 6 indicate the *command/response type*. Here is the coding for the *c* bits:

Command/Response Type	Bit Pattern (c)
Read System Configuration Command/Response	1101
Write System Configuration Command/Response	0101
Read Application I/O Command/Response	1001
Write Application I/O Command Response	0001
Deliver System/Application Event Message	1010

Figure 3.1.5

The type bits *(t)* and the argument bits *(uc or sc)* indicate the function of a network command and its response. For example, if the *type* bit coding is 0111b and the *argument* bits are 0110 0101b, the network message is a *Write ADC Configuration* system command. Some type and argument combinations are not allowed. For example, over-

writing the Firmware Version with a network command is not allowed. The disallowed network commands are usually self-evident. Depending on the nature of a network message, it may include no payload, or a payload of 1 to 64 bytes, starting in byte position 7. An example of each valid network command, response and event message is given below, including payload structure information. Unless otherwise noted, any configuration parameters shown in a payload are the *factory default values*. In most examples, an AID address of 001H is used. A few network messages directly address the base station. In these cases, the base station AID of 000H is used. The use of an AID address of 000H in a network command creates a broadcast packet, which is received and acted on by all router and field nodes. Broadcast packets are useful in setting network wide configurations.

Application and configuration I/O commands and responses use payload byte 7 to hold a set of I/O mode control and status flags. The upper two bits indicate the I/O type, and set the interpretation of the lower six bits. Additional I/O data may also be carried in bytes 8 and 9. Here is the interpretation of payload byte 7 for each I/O type:

Byte 7 Basic I/O Structure

Bit Position	7	6	5	4	3	2	1	0
Flag	t	t	f	f	f	f	f	f

Figure 3.1.6

ADC Input

Bit	Flag	Function
7	0	01b specifies an ADC Input
6	1	
5	m	enable threshold event messaging
4	e	enable read RSSI
3	s	current digital input state (level)
2	h	high threshold event flag
1	l	low threshold event flag
0	r	enable 10-bit resolution

Figure 3.1.7

The value in byte 8 is used for the high threshold, and the value in byte 9 is used for the low threshold. These threshold values are aligned with the 8 most significant bits of the ADC measurement.

For the ADC input, payload byte 7 can be written and read by both application and system commands. In miniMESH V2.5, however, the *Read ADC Input* application command sets the ADC resolution at the time the measurement is made.

Digital Input

Bit	Flag	Function
7	0	00b specifies a Digital Input
6	0	
5	m	enable input event messaging
4	r	enable reset counter at threshold
3	c	enable pulse/edge count
2	e	edge count enable, high-to-low if b is 0
1	b	low-to-high edge count if b is 1
0	s	current digital input state (level)

Figure 3.1.8

The digital input mode control is especially flexible. If input event messaging is enabled, an event message can be sent on any state transition, or on a high-to-low transition, or on a low-to-high transition, or on a pulse count reaching the threshold value held in bytes 8, 9 and 10, or on an edge (transition) count reaching the threshold value held in bytes 8, 9 and 10. Bits 1 *(b)*, 2 *(e)* and 3 *(c)* control transition and counting modes as follows:

c	e	b	Function
0	0	0	any transition is an event
0	1	0	high-to-low transition is an event
0	1	1	low-to-high transition is an event
1	0	0	count all transitions
1	1	0	count high-to-low transitions
1	1	1	count low-to-high transitions

Figure 3.1.9

In miniMESH V2.5, byte 7 in the digital input payload can be read by both application and system commands, but can only be changed (written) by a *Write Digital Input Configuration* system command.

Digital Output

Bit	Flag	Function
7	1	10b specifies a Digital Output
6	0	
5	0	reserved (set to 0)
4	s	current digital input state (level)
3	d	power up default state
2	q	2.2 ms tick if q is 1, 143 ms tick if q is 0
1	0	reserved (set to 0)
0	s	current digital output state (level)

Figure 3.1.10

Byte 7 in the digital output payload can be written and read by both application and system commands. A *Write Digital Output Configuration* command sets the power-up digital output configuration. The digital output is set to the state stored in bit 3 at power up. This state can be changed with a *Write Digital Output* command as discussed in the Application Messages section below. The values held in bit 2 *(q)* and byte 8 control the duration of a state change. If the value in byte 8 is 00H, a change persists until the next *Write Digital Output* command is received. If the value in byte 8 is in the range 2 to 255, the state change persists until the value in byte 8 is decremented to zero, at which time the output state switches. The decrement rate is controlled by flag bit 2 *(q)*. If the flag is clear, the decrement rate is every 143 ms. If the flag is set, the decrement rate is every 2.2 ms. Note that byte 7 can be configured so that one pulse is automatically output following the 2.4 s power up initialization interval. See the *Write Digital Output Configuration* command in section 5 for details.

UART

Bit	Flag	Function
7	1	11b specifies a UART I/O port
6	0	
5	m	enable UART event messaging
4	d	enable 7-bit data
3	0	reserved (set to 0)
2	p	enable parity/stop bit options
1	b	baud rate: 00b for 9.6 kb/s, 01b for 4.8 kb/s, 10b for 2.4 kb/s, 11b for 1.2 kb/s
0	b	

Figure 3.1.11

Byte 7 in the UART payload can be read by both application and system commands, but can only be changed by a *Write UART Configuration* system command. If input event messaging is enabled, a string received by a field node UART will start the event message sequence. Note that the received string is not sent in the event message. The string must be retrieved with a *Read UART Buffer* application command. In miniMESH V2.5, bit 4 in the UART payload enables 7-bit data (8-bit is the default). Bit 2 in the UART payload and bits 6 and 7 in byte 7 of the node auxiliary configuration payload select parity and stop bit options. See Figure 5.0.46 for UART configuration details.

Power Management Parameters

Power management parameters are carried in payload bytes 7, 8 and 9. Bytes 7 and 8 contain a mix of flags and counter bits. Byte 9 is the lower 8 bits of a 13-bit sleep time counter, with the high order bits are carried in byte 8. Power management parameters are stored in two places, nonvolatile memory, referred to a *static* parameters, and RAM memory, referred to as *dynamic* parameters. When a router or field node is turned on or reset, the power management parameters are copied from nonvolatile memory to RAM memory, providing default power management functionality. The parameters in RAM memory can be updated at any time to modify power management functionality. Pa-

rameters can also be written to nonvolatile memory without changing the values in RAM until the node is reset or the power is cycled off and on.

Power Management Byte 7

Bit	Flag	Function
7	e	enable power management
6	t	enable quick sleep
5	l	enable listen-before-sleep (cycling)
4	n	5-bit listen-before-sleep timer. If bit 6 is set, "tick" interval is taken from the *Node Auxiliary Configuration* clock setting. If bit 6 clear, the "tick" if fixed at 15 s
3	n	
2	n	
1	n	
0	n	

Figure 3.1.12

If bit 7 *(e)* is clear, power management is disabled and the node is continuously active. If bit 7 *(e)* is set, power management is enabled. In miniMESH V2.5 a new flag, *enable quick sleep*, has been added in bit position 6 *(t)*. When this bit is clear, miniMESH V2.5 power management is directly compatible with miniMESH V2.2 and V2.3.

If bit 6 *(t)* is clear, then if bit 5 *(l)* is clear, the listen-before-sleep interval is fixed at approximately 15 seconds. If bit 5 *(l)* is set, the listen-before-sleep interval is adjustable from 1 to 31 steps of 15 seconds. Also, if bit 5 *(l)* is set, the node will continuously cycle between an active period set by the listen-before-sleep interval and the sleep interval. When bit 5 *(l)* is clear, the node will stay active following a sleep interval until a new power management command is received.

If bit 6 *(t)* is set, then if bit 5 *(l)* is clear, the node goes to sleep immediately following a power management command. If bit 5 *(l)* is set, the listen-before-sleep interval is adjustable from 1 to 31 "ticks". The "tick" clock, in turn, is set by the *Write Node Auxiliary Configuration* command from 3 to 7.43 seconds per tick, with 5 seconds being the default value (see Figure 5.0.47). Also, if bit 5 *(l)* is set, the node will continuously cycle between an active period set by the listen-before-sleep interval and the sleep interval. When bit 5 *(l)* is clear, the node will stay active following sleep until a new power management command is received.

Power Management Byte 8

If bit 7 *(s)* of byte 8 is clear, power management is directed at a field node(s). If bit 7 *(s)* is set, power management is directed at a router(s). If bit 6 *(r)* is clear, a node will not transmit an event message when it wakes up. If bit 6 *(r)* is set, a node will transmit an event message when it wakes up. If bit 5 *(a)* is clear, inputs are not tested to check for events during a sleep interval. If bit 5 *(a)* is set, the node will periodically wake up and measure the analog input and digital input. If the analog input is above the high threshold or below the low threshold, the node will stay awake and begin sending event mes-

sages related to the analog input. Likewise, the node will stay awake and begin sending event messages related to an event condition on the digital input.

Bit	Flag	Function
7	s	router/field node select
6	r	enable report on wake up
5	a	enable event checking during sleep
4	n	upper 5 bits of 13-bit sleep timer, 15 seconds per count
3	n	
2	n	
1	n	
0	n	

Figure 3.1.13

The bind input can be used to wake a sleeping node before the end of a sleep interval. Bits 4, 3, 2, 1 and 0 are the upper 5 bits of the 13-bit sleep interval timer. Byte 9 holds the lower 8 bits. The sleep timer covers approximately 15 seconds to 34 hours. Each count represents about 15 seconds. The minimum interval is 0 0000 0000 0001b, and the maximum interval is 1 1111 1111 1111b. A low current R-C timer is used in the DM1810 for sleep interval timing, so sleep interval settings are only approximate.

Power Management Byte 9

Byte 9 holds the lower eight bits of the 13-bit sleep interval timer.

4.0 Application I/O Commands, Responses and Event/Notification Messages

Read ADC Input with 10-bit Resolution

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	A9			
Network Response	E0	00	00	00	01	09	41	01	D9

Figure 4.0.1

As shown in Figure 3.1.7, the bit pattern 41H in byte 7 of the response indicates this is a 10-bit unsigned ADC reading. The value is right justified in payload bytes 8 and 9. The value shown is slightly below a half-scale reading of 0200H.

Read ADC Input with 8-bit Resolution

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	29			
Network Response	E0	00	00	00	01	09	40	76	

Figure 4.0.2

The flag bit pattern 40H in byte 7 of the response indicates this is a 8-bit unsigned ADC reading. The value is right justified in payload byte 8. The value shown is slightly below a half-scale reading of 80H.

Read Digital Input

Byte Position	1	2	3	4	5	6	7	8	9	10
Network Command	60	00	00	00	01	19				
Network Response	E0	00	00	00	01	09	01	00	00	00

Figure 4.0.3

As shown in Figures 3.1.8 and 3.1.9, the flag bit pattern 01H in byte 7 of the response indicates this is a digital input with a current state of 1. Bytes 8, 9 and 10 hold the accumulated pulse counts if the digital input is configured for pulse counting.

Read Digital Output

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	39			
Network Response	E0	00	00	00	01	09	81	80	

Figure 4.0.4

As shown in Figure 3.1.10, the flag bit pattern 81H in byte 7 of the response indicates this is a digital output with a current value of 1. If a pulse is being output when a read command is received, byte 8 in the response will hold the remaining pulse duration value (80H in the above example). Otherwise byte 8 will read 00H.

Read UART Buffer

Byte Position	1	2	3	4	5	6	7 through 70
Network Command	60	00	00	00	01	49	
Network Response	E0	00	00	00	01	09	payload

Figure 4.0.5

The response to this command returns a string received in the UART buffer in payload bytes 7 through 70. If there is nothing in the UART buffer, there will be no payload (null string). Reading the UART buffer does *not* clear it. The UART cannot receive another string until the buffer is cleared. The buffer can be cleared by a *Write UART Buffer* command with no payload.

Read Dynamic Power Management Parameters

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	F9			
Network Response	E0	00	00	00	01	09	00	00	00

Figure 4.0.6

The response to this command returns the power management parameters currently being used in bytes 7, 8 and 9. See Figures 3.1.12 and 3.1.13 for interpretation details of these bytes. This command also acknowledges a wake up event message.

Write (Reset) Analog Input

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	21			
Network Response	E0	00	00	00	01	01			

Figure 4.0.7

The purpose of this command is to reset the threshold event flag. This command will also cancel any pending analog input event message transmission retries.

Write (Reset) Digital Input

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	11			
Network Response	E0	00	00	00	01	01			

Figure 4.0.8

The purpose of this command is to reset the digital input pulse counter to 000000H, and/or reset any digital input event flags. This command will also cancel any pending digital input event message transmission retries.

Write (Set) Digital Output

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	31	81	00	
Network Response	E0	00	00	00	01	01			

Figure 4.0.9

If byte 8 is set to 00H, the *Write Digital Output* command will set the digital output to the value of bit 0 in byte 7. If the value in byte 8 is set to a value between 2 and 255 (avoid a value of 1), this command will output a pulse of the value of bit 0 in byte 7. The pulse duration depends on the value in byte 8 and the value of bit 2 in byte 7, as given in the equations below. Once the pulse duration has expired, the digital output will change to opposite state. The pulse duration counter is a low priority protocol task, so the pulse duration may be somewhat longer than nominal when there is heavy radio traffic. The nominal duration is given as:

$T_{seconds} = 0.143\ (N + 1)$ when bit 2 in byte 7 is 0
$T_{milliseconds} = 2.2*(N + 1)$ when bit 2 in byte 7 is 1
where N is the value in byte 8, in the range of 2 to 255

Write (Send) UART Buffer

Byte Position	1	2	3	4	5	6	7 through 70
Command	60	00	00	00	01	41	payload
Response	E0	00	00	00	01	01	

Figure 4.0.10

This command writes a string contained in payload bytes 7 through 70 to the UART buffer. This, in turn, cases the UART to output the string. Sending this command to the UART buffer with no payload clears the buffer of a received message.

Write Dynamic Power Management Parameters

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	F1	00	00	00
Network Response	E0	00	00	00	01	01			

Figure 4.0.11

This command immediately updates the power management parameters being used by a node. The parameters are carried in bytes 7, 8 and 9. See Figures 3.1.12 and 3.1.13 for the details. This command also acknowledges a wake up event message.

Event Message

Byte Position	1	2	3	4	5	6	7	8	9
Network Event Msg	FF	00	00	00	01	8A			

Figure 4.0.12

Specific conditions can be set to trigger the transmission of event messages. An event message is automatically sent from a field node when one or more trigger conditions occur. The condition(s) creating the event message is flagged in the upper nibble of byte 6, as shown in Figure 4.0.13. An event message is sent until all conditions are acted upon, or up to eight times. In miniMESH V2.5, the interval between event message retransmissions increases in the following progression: 1.5, 3, 6, 12, 24, 36 and 36 seconds, to which a randomly chosen delay of 0, 143, 286 or 429 ms is added on each retry. A field node processing an event will not enter a sleep cycle until the event is acted upon or it has sent the event message eight times *unless the power management quick sleep bit is set*. In this case, the first retry interval is increased to 4 seconds and the node will enter a sleep cycle unconditionally at the end of the listen-before-sleep period. In V2.5, reading the ADC will reset an ADC threshold trigger.

Bit	Flag	Function
7	w	wake up event flag
6	u	UART event flag
5	a	ADC event flag
4	d	digital input event flag

3	1	event message type nibble
2	0	
1	1	
0	0	

Figure 4.0.13

A set bit 5 *(a)* indicates that the ADC reading is either at or below the low threshold setting, or is at or above the high threshold setting. ADC event messaging is enabled by setting a flag in the ADC configuration. The actual ADC value is retrieved by sending a *Read ADC Input* command, and the analog input event flag is reset with a *Write (Reset) Analog Input* command.

A set bit 4 *(d)* indicates that the digital input value or change in value has triggered an event, or that the digital pulse count is above the threshold setting held in the digital input configuration. Digital input event messaging is enabled by setting a flag in the digital input configuration. The actual digital input value and/or pulse count is retrieved by sending a *Read Digital Input* command, and the digital input event flag is reset with a *Write (Reset) Digital Input* command.

A set bit 6 *(u)* indicates that a string has been received through the UART input. UART event messaging is enabled by setting a flag in the UART configuration. The actual string is retrieved by sending a *Read UART Buffer* command, which also resets bit 6 and ends the sending of the event message due to a UART event.

A set bit 7 *(w)* indicates that a field node or router is awake following a sleep cycle. A *Read (or Write) Dynamic Power Management Parameters* command resets bit 7 and ends the sending of the event message due to a wake up event.

More than one event flag can be set in an event message. Acting on any set event flag will stop the retransmission of the event message. Note, however, that individual event flags remain set until acted on.

Read Status of Event Flags

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	60	00	00	00	01	E9			
Network Response	E0	00	00	00	01	09	71		

Figure 4.0.14

The response to this command indicates the status of the addressed node's event flags, as shown in Figure 4.0.15 below:

Bit	Flag	Function
7	a	event messaging active
6	c	transmission retry count (00H on first transmission)
5	c	
4	c	
3	w	wake up event flag
2	u	UART event flag
1	a	ADC event flag
0	d	digital input event flag

Figure 4.0.15

During the time an event message is being transmitted, bit 7 *(a)* is set. After all events causing the event message have been acted on or the event message has been sent eight times (initial transmission plus seven retries), this bit will be reset. Bits 6, 5 and 4 indicate the last recorded retry count. A set bit 3 *(w)* indicates the last wake up event has not been acted on. A set bit 2 *(u)* indicates the last UART event has not been acted on. A set bit 1 *(a)* indicates the last ADC event has not been acted on. A set bit 0 *(d)* indicates the last digital input event has not been acted on.

Power On Notification Message

Byte Position	1	2	3	4	5	6	7	8	9
Network Event Msg	FF	00	00	00	01	2B	25		

Figure 4.0.16

If enabled (default), this message is sent once when a node is powered up, and can be used to confirm a router or field node's AID. Byte 7 holds the code version number in the upper nibble and the revision number in the lower nibble, in this case, version 2.5.

5.0 System Configuration Command and Response Messages

System configuration commands addressed to field nodes or routers use a message type *(t)* bit pattern of 0111b, and a non-zero AID address. System configuration commands addressed to the base station use a message type *(t)* bit pattern of 1111b, and a 000H AID address. All system configuration responses return a message type *(t)* bit pattern of 1111b.

Read ADC Input Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	6D			
Network Response	F0	00	00	00	01	0D	40	FF	00

Figure 5.0.1

As shown in Figure 3.1.7, the flag bit pattern 40H in byte 7 of the response indicates ADC configuration data. Bytes 8 and 9 hold the 8 most significant bits of the upper and lower event thresholds for the ADC.

Read Router/Field Node RSSI Values

Byte Position	1	2	3	4	5	6	7 through 22
Network Command	70	00	00	00	01	29	
Network Response	F0	00	00	00	01	09	RSSI payload

Figure 5.0.2

DM1810 RSSI measurements are supported by miniMESH V2.5. To read the RSSI values from a router or field node, disable periodic ADC readings with the *Write Timer Configuration* command by setting the timer interval to 00H. Next, use the *Write ADC Input Configuration* command to set the RSSI enable bit, as shown in Figure 3.1.7. Next send a *Reset Node* command. The *Read Router or Field Node RSSI Values* command can then be used. The response message will hold the RSSI value for the base station in byte 7 and the RSSI values for routers 1 through 15 in bytes 8 through 22. An RSSI value of 00H is returned for a node that cannot be directly heard. To return to normal ADC operation, use the *Write ADC Input Configuration* command to reset the RSSI enable bit. If periodic ADC readings are desired, set the interval with the *Write Timer Configuration* command. Then send a *Reset Node* command.

Read Base Station RSSI Values

Byte Position	1	2	3	4	5	6	7 through 22
Network Command	F0	00	00	00	00	29	
Network Response	F0	00	00	00	01	09	RSSI payload

Figure 5.0.3

DM1810 base station RSSI measurements are also supported by miniMESH V2.5. The base station does not require any configuration changes to make RSSI measurements. Reading the base station RSSI values is a two step process. First, use the *Read Router or Field Node RSSI* command to read the RSSI values from any router or field node (active or unused address OK). Then send a *Read Base Station RSSI Values* command to return the base station RSSI values. The response message will hold the 8-bit RSSI values heard by the base station, with byte 7 always 00H and the RSSI values for routers 1 through 15 in bytes 8 through 22. An RSSI value of 00H is also returned for a router that cannot be heard directly by the base station.

Read Digital Input Configuration

Byte Position	1	2	3	4	5	6	7	8	9	10	11
Network Command	70	00	00	00	01	5D					
Network Response	F0	00	00	00	01	0D	28	00	00	00	08

Figure 5.0.4

As shown in Figure 3.1.8, the flag bit pattern 01H in byte 7 of the response indicates this is digital input configuration data. Bytes 8, 9 and 10 hold the pulse count event message threshold. The value in byte 11 is the de-bounce filter value. The default value for byte 11 is 08H. The valid range for byte 11 is 08H to FFH. The de-bounce period is 2.2*N ms, where N is the value in byte 11.

Read Digital Output Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	7D			
Network Response	F0	00	00	00	01	0D	80	00	

Figure 5.0.5

As shown in Figure 3.1.10, the bit pattern 80H in byte 7 of the response indicates this is digital output configuration data. Byte 8 holds the power up pulse output duration value.

Read UART Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	8D			
Network Response	F0	00	00	00	01	0D	C0	00	

Figure 5.0.6

As shown in Figure 3.1.11, the bit pattern C0H in byte 7 above indicates this is UART configuration data (byte 8 is not used in V2.5). Also see Figure 5.0.46.

Read Static (Stored) Power Management Parameters

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	2D			
Network Response	F0	00	00	00	01	0D	00	00	00

Figure 5.0.7

The response to this command returns the power management parameters stored in a router or field node. The parameters are held in bytes 7, 8 and 9. These parameters control the power management cycle following a software reset or power up cycle. See Figures 3.1.12 and 3.1.13 for interpretation details of these bytes.

Read Router/Field Node Bind Configuration

Byte Position	1	2	3	4	5	6	7 through 10
Command	70	00	00	00	01	3D	
Response	F0	00	00	00	01	0D	payload

Figure 5.0.8

The response to this command returns the 4 bytes in the bind configuration control block. Here is the interpretation of each payload byte:

Byte	Interpretation
7	network ID bits and upper alias ID bits
8	lower alias ID bits
9	system ID
10	system key

Figure 5.0.9

Read Base Station Bind Configuration

Byte Position	1	2	3	4	5	6	7 through 10
Command	F0	00	00	00	00	3D	
Response	F0	00	00	00	00	0D	payload

Figure 5.0.10

The response to this command returns the 4 bytes in the base station bind configuration control block. See Figure 5.0.9 for the interpretation of each payload byte. Note that the base station alias ID bits are set to 0.

Read Base Station Bind List

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	4D			
Network Response	F0	00	00	00	00	0D	20	08	

Figure 5.0.11

The response to this command returns the base station's bind list in payload bytes 7 and 8. The upper four bits in byte 7 is the number of routers bound to the base station. The lower two bits in byte 7 and the 8 bits in byte 8 give the total number of nodes bound to the base station, both field nodes and routers.

Read User Message A

Byte Position	1	2	3	4	5	6	7 through 30
Command	70	00	00	00	01	BD	
Response	F0	00	00	00	01	0D	payload

Figure 5.0.12

The response to this command returns user-defined message A in payload bytes 7 through 30. The message is 24 bytes in length, and can be used for a node description, installation or maintenance record, or any other use chosen by the user.

Read User Message B

Byte Position	1	2	3	4	5	6	7 through 30
Command	70	00	00	00	01	CD	
Response	F0	00	00	00	01	0D	payload

Figure 5.0.13

The response to this command returns user-defined message B in payload bytes 7 through 30. The message is 24 bytes in length, and can be used for a node description, installation or maintenance record, or any other use chosen by the user.

Read Node I/O Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	DD			
Network Response	F0	00	00	00	01	0D	8F		

Figure 5.0.14

The response to this command returns the addressed node's configuration in payload byte 7. In this example, the device is a field node with all hardware I/O enabled. See Figure 5.0.43 for the interpretation of byte 7.

Read Firmware Version

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	ED			
Network Response	F0	00	00	00	01	0D	25		

Figure 5.0.15

The response to this command returns the addressed node's firmware version in payload byte 7. In this example, the firmware version is 2.5.

Read Hardware ID

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	FD			
Network Response	F0	00	00	00	01	0D	AA	BB	CC

Figure 5.0.16

The response to this command returns the addressed node's 24-bit hardware ID in payload bytes 7, 8 and 9. The hardware ID in this example is AABBCCH. The hardware ID is used in binding transactions.

Read Timer Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	9D			
Network Response	F0	00	00	00	01	0D	00	00	

Figure 5.0.17

The response to this command returns the interval between automatic ADC readings for event messages in byte 7 (byte 8 is reserved and is set to 00H). Setting byte 7 to 00H disables automatic ADC readings. Each count in byte 7 represents 15 seconds, with a value of FFH representing slightly over an hour.

Read Router/Field Node Mode and Model Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	1D			
Network Response	F0	00	00	00	01	0D	01		

Figure 5.0.18

The response to this command returns the mode, either DM1810 native or DM1800 compatible, and the model (operating band), either DM1810-434 or DM1810-916, for a router or field node as shown below:

Bit	Flag	Function
7	m	0 is DM1810 native, 1 is DM1800 compatible
6	0	reserved (set to 0)
5	0	
4	0	
3	0	
2	f	01H is DM1810-434, 02H is DM1810-916
1	f	
0	f	

Figure 5.0.19

Read Base Station Mode and Model Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	1D			
Network Response	F0	00	00	00	00	0D	01		

Figure 5.0.20

The response to this command returns the base station mode, either DM1810 native or DM1800 compatible, and the model (operating band), either DM1810-434 or DM1810-916, as shown in Figure 5.0.19.

Read Event Routing Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	AD			
Network Response	F0	00	00	00	01	0D	0F		

Figure 5.0.21

The response to this command returns the number of routers assumed by a DM1810 field node when sending an event message. The default number of routers is set at bind time according to the base station operating mode. Event routing at a field node defaults to 15 for DM1810 native mode operation or to 7 for DM1800 compatible operation. If the actual number of routers used in a system is fixed and is less than the maximum number of routers allowed by the operating mode, the actual number of routers can be specified by a *Write Event Routing Configuration* command for more efficient event message routing. If the base station mode is changed using the *Write Base Station Mode Configuration* command, the event routing configuration of the field nodes must changed match the new mode.

Read Router/Field Node Link Map

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	FE			
Network Response	F0	00	00	00	01	0E	00	00	

Figure 5.0.22

The response to this command returns the routing nodes (base station plus routers) that can be directly received by a router or field node. Bytes 7 and 8 are used in the native DM1810 mode, and byte 7 only is used in the DM1800 compatible mode, where the number of routers are limited to 7. In the native DM1810 mode, bit 0 of byte 8 references the base station, bit 1 of byte 8 references router 1, bit 2 of byte 8 references router 2, etc., with bit 0 of byte 7 referencing router 8, bit 1 of byte 7 referencing router 9, etc. In the DM1800 compatible mode, bit 0 of byte 7 references the base station, bit 1 of byte 7 references router 1, bit 2 of byte 7 references router 2, etc. A *Reset Link Map* command should be issued just before this command to assure fresh data.

Read Base Station Link Map

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	FE			
Network Response	F0	00	00	00	00	0E	00	00	

Figure 5.0.23

The response to this command returns the routers that can be directly received by the base station. Bytes 7 and 8 are used in the native DM1810 mode, and byte 7 only is used in the DM1800 compatible mode, where the number of routers are limited to 7. In the native DM1810 mode, bit 0 of byte 8 is set to 0. Bit 1 of byte 8 references router 1,

bit 2 of byte 8 references router 2, etc., with bit 0 of byte 7 referencing router 8, bit 1 of byte 7 referencing router 9, etc. In the DM1800 compatible mode, bit 0 of byte 7 is set to 0. Bit 1 of byte 7 references router 1, bit 2 of byte 7 references router 2, etc. To use this command, first send a *Reset Base Station Link Map* command to assure fresh data. Then send a command to any field node to "seed" the base station link map, such as a *Read Digital Input* command. Then send a *Read Base Station Link Map* command to retrieve the fresh link map.

Reset Router/Field Node Link Map

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	7E			
Network Response	F0	00	00	00	01	0E			

Figure 5.0.24

This command clears the link map bits in the addressed router or field node.

Reset Base Station Link Map

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	7E			
Network Response	F0	00	00	00	00	0E			

Figure 5.0.25

This command clears the base station link map bits.

Reset Node

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	F8			
Network Response	F0	00	00	00	01	08			

Figure 5.0.26

This command resets the addressed node. *Issuing this command after a configuration write command will cause the new configuration data to be used by the node. Otherwise the old configuration data will continue to be used until the power to the node is turned off and back on.*

Write ADC Input Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	65	40	FF	00
Network Response	F0	00	00	00	01	05			

Figure 5.0.27

As shown in Figure 3.1.7, the upper two bits in byte 7 of this command are set to 01b to indicate this is ADC configuration data. In miniMESH V2.5, the lower four bits of byte 7 should be written as zeros. The value in byte 8 is used for the high threshold, and the value in byte 9 is used for the low threshold. These threshold values are aligned with the 8 most significant bits of the ADC measurement.

Write Digital Input Configuration

Byte Position	1	2	3	4	5	6	7	8	9	10	11
Network Command	70	00	00	00	01	55	28	00	00	00	08
Network Response	F0	00	00	00	01	0D	05				

Figure 5.0.28

The upper two bits in byte 7 of this command are set to 00b to indicate this is digital input configuration data. As shown in Figures 3.1.8 and 3.1.9, the lower six bits are set to specify if event messaging is enabled, if counting is enabled, and to select event and counting modes. Note that if event messaging is enabled and bit 3 *(c)* in byte 7 is clear, event messaging is controlled by input transitions. If bit 3 *(c)* is set, event messaging is controlled by the 24-bit count threshold in bytes 8, 9 and 10. The value in byte 11 is the de-bounce filter value. The default value for byte 11 is 08H. The valid range for byte 11 is 08H to FFH. The de-bounce period is 2.2*N ms, where N is the value in byte 11.

Write Digital Output Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	75	80	00	
Network Response	F0	00	00	00	01	05			

Figure 5.0.29

As shown in Figure 3.1.10, the upper two bits in byte 7 of this command are set to 10b to indicate this is digital output configuration data. The lower six bits are set to specify the power up default output state, and to select either fast pulse duration counting (2.2 ms/tick) or slow pulse duration counting (143 ms/tick). This command can be used to automatically generate a single pulse following the 2.4 s power on initialization period. During the power on initialization, the digital output is set to zero. To output a high pulse, set bits 3 and 0 to one, set the pulse duration bit 2 as needed, and load byte 8 with the pulse duration value. To extend the low output for a period following the start up interval, set bits 3 and 0 to zero, set the pulse duration bit 2 as needed, and load byte 8 with the pulse duration value. Once the low output duration has expired, the digital output will change to a high state. The pulse duration counter is a low priority protocol task, so the pulse duration may be somewhat longer than nominal when there is heavy radio traffic. The nominal duration is given as:

$T_{seconds}$ = 0.143 (N + 1) when bit 2 is 0
$T_{milliseconds}$ = 2.2*(N + 1) when bit 2 is 1
where N is the value in byte 8, in the range of 2 to 255

Write UART Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	85	C0	00	
Network Response	F0	00	00	00	01	05			

Figure 5.0.30

As shown in Figure 3.1.11, the upper two bits in byte 7 of this command are set to 11b to indicate this is UART configuration data. Bit 5 controls receive event messaging. Bit 4 selects 7-bit data. Bit 3 is reserved and should be set to 0. See Figure 5.0.46 for the interpretation of bit 2. Bits 1 and 0 set the baud rate. Set byte 8 to 00H.

Write Static (Stored) Power Management Parameters

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	25	00	00	00
Network Response	E0	00	00	00	01	05			

Figure 5.0.31

This command updates the power management parameters stored in a node. The parameters are carried in bytes 7, 8 and 9. These parameters control the power management cycle following a software reset or power up cycle. See Figures 3.1.12 and 3.1.13 for the details of these parameters. Note that power management parameters can be changed during node operation using a *Write Dynamic Power Management Parameters command.*

Write User Message A

Byte Position	1	2	3	4	5	6	7 through 30
Command	70	00	00	00	01	B5	payload
Response	F0	00	00	00	01	05	

Figure 5.0.32

This command writes user-defined message A to the addressed node. The message is carried in payload bytes 7 through 30. The message is 24 bytes in length, and can be used for a node description, installation or maintenance record, or any other use chosen by the user.

Write User Message B

Byte Position	1	2	3	4	5	6	7 through 30
Command	70	00	00	00	01	C5	payload
Response	F0	00	00	00	01	05	

Figure 5.0.33

This command writes user-defined message B to the addressed node. The message is carried in payload bytes 7 through 30. The message is 24 bytes in length, and can be used for a node description, installation or maintenance record, or any other use chosen by the user.

Write Timer Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	95	00	00	
Network Response	F0	00	00	00	01	05			

Figure 5.0.34

This command sets the interval between automatic ADC readings for event messages in byte 7 (byte 8 is reserved and should be set to 00H). Setting byte 7 to 00H disables automatic ADC readings. Each count in byte 7 represents 15 seconds, with a value of FFH representing slightly over an hour.

Write Router/Field Node Mode Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	15	00		
Network Response	F0	00	00	00	01	05			

Figure 5.0.35

This command sets the mode, either DM1810 native (bit clear) or DM1800 compatible (bit set), in bit 7 of byte 7 of a router or field node. The rest of the bits in byte 7 are read only and/or reserved. This command must be followed by a *Reset Node* command. See Figure 5.0.19 for additional information.

Write Base Station Mode Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	15	00		
Network Response	10	30							

Figure 5.0.36

This command sets the mode, either DM1810 native (bit clear) or DM1800 compatible (bit set), in bit 7 of byte 7 of a base station. The base station is addressed at 000H. The rest of the bits in byte 7 are read only and/or reserved. This command must be followed by a *Base Reset* command. See Figure 5.0.19 for additional information.

Write Event Routing Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	A5	0F		
Network Response	F0	00	00	00	01	05			

Figure 5.0.37

This command sets the number of routers assumed by a DM1810 field node when sending an event message. The default number of routers is set at bind time according to the base station operating mode at 15 for DM1810 native operation or 7 for DM1800 compatible operation. If the actual number of routers used in a system is fixed and is less than the maximum number of routers allowed by the operating mode, this command can be used to specify the actual number of routers being used for more efficient event message routing. If the base station mode is changed using the *Write Base Station Mode Configuration*, the event routing configuration must changed at the field nodes to match the new mode.

Write Router/Field Node Bind Configuration

Byte Position	1	2	3	4	5	6	7 through 10
Command	70	00	00	00	01	35	00 00 00 00
Response	F0	00	00	00	01	05	

Figure 5.0.38

This command can be used to modify the bind configuration of a router or field node after it has been bound to a network. Its primary use is to set the bind configuration of a new node to match the bind configuration of an existing node being removed from service. Here is the interpretation of each payload byte:

Byte	Interpretation
7	network ID bits and upper alias ID bits
8	lower alias ID bits
9	system ID
10	system key

Figure 5.0.39

This command must be *used very carefully* to avoid assigning the same AID to two active nodes, or assigning an AID that is out of the range of the bind list in the base station.

Write Base Station Bind Configuration

Byte Position	1	2	3	4	5	6	7 through 10
Command	F0	00	00	00	00	35	00 00 00 00
Response	10	30					

Figure 5.0.40

This command can be used to clear or modify the base station bind configuration. When the base station bind configuration and bind list are both cleared, a new network can be built around the base station by (re)binding field nodes and routers to it. This command must be *used very carefully* to avoid unintentionally disabling a network.

Write Base Station Bind List

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	F0	00	00	00	00	45	00	00	
Network Response	10	30							

Figure 5.0.41

This command can be used to clear or modify the base station bind list. When the base station bind list and bind configuration are both cleared, a new network can be built around the base station by (re)binding field nodes and routers to it. This command must be *used very carefully* to avoid unintentionally disabling a network.

Write Node I/O Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	D5	8F		
Network Response	F0	00	00	00	01	05			

Figure 5.0.42

This command enables/disables I/O functions and continuous LED operation on a field node. Here is the interpretation of byte 7 for this command:

Bit	Flag	Function
7	d	device type (read only bits): 00 is base, 01 is router, 10 is field node
6	d	
5	c	continuous LED operation enable
4	0	reserved (set to 0)
3	u	UART enable
2	o	digital output enable
1	a	analog input enable
0	i	digital input enable

Figure 5.0.43

Read Node Auxiliary Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	4D			
Network Response	F0	00	00	00	01	0D	00	00	

Figure 5.0.44

This command reads the auxiliary configuration of a field node or router. The bit functions in byte 7 are shown in Figure 5.0.45. Bits 2 and 4 in UART configuration byte 7 (see Figure 3.1.11) and bits 6 and 7 in auxiliary configuration byte 7 are use to configure the UART in a field node, as shown in Figure 5.0.46 below. Setting bit 3 disables the power on notification transmissions in a field node or router.

Bit	Flag	Function
7	p	parity/stop bit select (field node)
6	p	
5	0	reserved (set to 0)
4	0	reserved (set to 0)
3	o	disable power on notification
2	0	reserved (set to 0)
1	0	reserved (set to 0)
0	0	reserved (set to 0)

Figure 5.0.45

UART	Figure 3.1.11		Figure 5.0.45	
Configuration	Bit 2	Bit 4	Bit 6	Bit 7
8-N-1	0	0	0	0
8-E-1	1	0	0	0
8-O-1	1	0	1	0
8-N-2	1	0	1	1
7-E-1	1	1	0	0
7-O-1	1	1	1	0
7-N-2	1	1	1	1
7-E-2	0	1	0	0
7-O-2	0	1	1	0

Figure 5.0.46

Byte 8 of the node auxiliary configuration payload is shown in Figure 5.0.45. Bits 0 - 4 implement a 5-bit clock “tick” that drives the listen-before-sleep timer used with sleep mode power management commands (see Figure 3.1.12). The clock “tick” interval can be set from 3 to 7.43 seconds.

Bit	Flag	Function
7	0	reserved (set to 0)
6	0	reserved (set to 0)
5	0	reserved (set to 0)
4	n	listen-before-sleep clock tick, t = 3 + 0.143*n seconds default n value is 1101b (5 seconds)
3	n	
2	n	
1	n	
0	n	

Figure 5.0.47

Write Node Auxiliary Configuration

Byte Position	1	2	3	4	5	6	7	8	9
Network Command	70	00	00	00	01	45	00	00	
Network Response	F0	00	00	00	01	05			

Figure 5.0.48

This command sets the auxiliary configuration of a field node or router. The bit functions in byte 7 are shown in Figure 5.0.45. Bit 2 and 4 of the UART configuration payload (see Figure 3.1.11) and bits 6 and 7 in the auxiliary configuration payload are use to configure the UART in a field node, as shown in Figure 5.0.46. The listen-before-sleep clock tick control in byte 8 is shown in Figure 5.0.47.

Field Node UART Reset Vector

Once a message is received in the UART buffer of a field node, it will not accept another message until it is reset by a *Write UART Buffer* command with no payload sent from the base station (Figure 4.0.10), or it is reset with a special 9-byte reset vector *applied to the field node UART input.* The field node does not test for the reset vector unless it is holding a message in the UART buffer.

Byte Position	1	2	3	4	5	6	7	8	9
Reset Vector	00	6E	29	6D	36	4A	7C	4E	6B

Figure 5.0.49

6.0 Temperature Sensor Example Application

The example application shown in Figure 6.0.1 measures temperature using a thermistor circuit connected to a DM1810 field node. The application is written in Microsoft Visual Basic 6.0 (SP6). The sensor used is a Vishay BCcomponents 2322 640 66103 NTC thermistor, which has a nominal resistance of 10K at 77 °F (25 °C). This thermistor is readily available from Digi-Key and other electronic component distributors. The data sheet for the thermistor is located at www.vishay.com-docs-29049-23226403.pdf. The thermistor is connected between the 3 V regulated output on the DM1810 and the ADC input, and a 10K, 1% resistor is connected from the ADC input to ground. The thermistor and the 1% resistor form a voltage divider to drive the ADC input, where the voltage increases with increasing temperature, as shown in Figure 6.0.2. The photograph in Figure 6.0.3 shows the thermistor and resistor installed on an IM1800 interface module. This thermisor-resistor circuit is built into the IM1800-1 module.

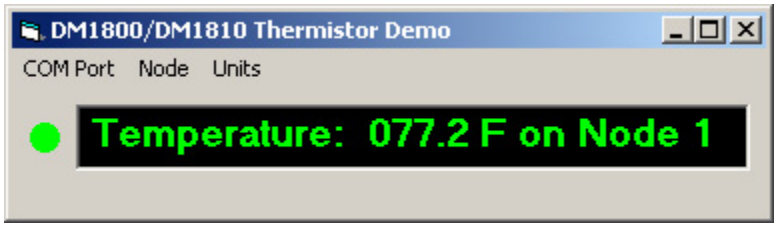


Figure 6.0.1: DM1800/DM1810 Themistor Demo

The example program is contained in a single Visual Basic form. The heart of the program is *Timer1_Timer*, which is launched each second by a timer control. This routine first checks the serial port input buffer. If the input buffer is holding one or more bytes, the *GetRX* subroutine is called to process them. *TXCmd* is then called to output a *Read ADC Input with 10-bit Resolution* command, which propagates through the miniMESH network to the addressed field node. The field nodes makes the ADC measurement and sends a response back to be read on the next *Timer1_Timer* cycle. Note that the first time *TXCmd* is called, it sends a *Digital Output High* command. This selects the thermistor-resistor circuit built into the IM1800-1 if this interface board is being used. Each time *TXCmd* sends a command it sets a response pending flag, *RPFLag*, and does not send out another command until this flag is reset by *GetRX* when the response is received, or after four second gross time out.

Each time *GetRX* is called, it collects the newly received bytes in *ComData$*. *GetRX* then adds *ComData$* to the end of the *FIFO$* string. The *FIFO$* string is scanned byte-by-byte, removing bi-sync framing and escape characters from each complete network message found in the *FIFO$* string. Local command strings are ignored, and any partial messages at the end of the FIFO$ string are saved for future processing.

Each time a complete network message is extracted, *GetRX* resets *RPFLag* and calls *ShowT*. This subroutine filters out all network messages except the *Read ADC Input with 10-bit Resolution* response messages. The ADC measurements are then converted to temperature readings and displayed. The *GetTemp* subroutine does the temperature

conversion using the *TmpArry* table values and interpolation. The *TmpArry* table values were derived from a resistance versus temperate table in the BCcomponents data sheet. These resistance values were, in turn, used to compute the *TmpArry* table of output voltage ratios, in terms of ADC counts, for the thermistor-resistor voltage divider.

Note that AID addresses 001H through 00AH can be selected from the Node menu. *TXCmd* selects and addresses command messages, which are then formatted and sent by *SndTX*. A checksum is added to the end of the command, and then DLE + STX and DLE + ETX framing is added. Note that if AID address 007H is selected, a DLE escape character is added after the checksum byte.

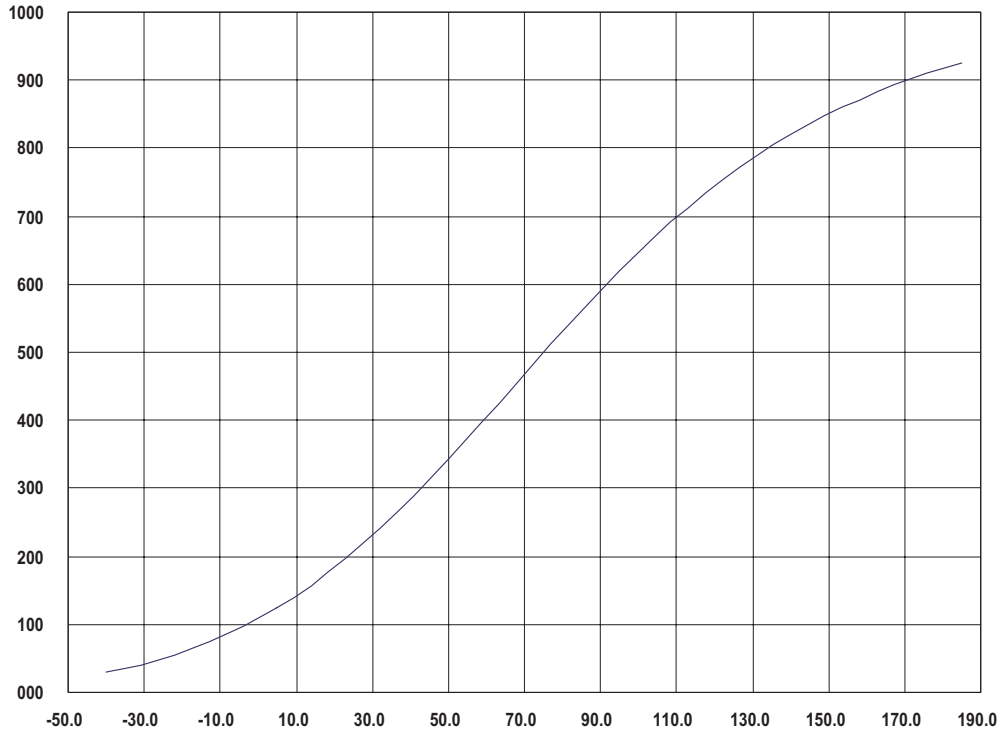


Figure 6.0.2: ADC Count$_{10}$ versus Temperature in °F

The temperature sensor example has been kept simple to avoid obscuring the basic DM1810 network command-response program methodology. An application install package, the project files and a spreadsheet tabulation of the temperature sensor voltage divider can be downloaded from Cirronet's web site at www.cirronet.com.

After the application is installed, launch it and select the COM port assigned to the IM1800 or IM1800-1 USB port. *COM 5* is shown in the example in Figure 6.0.4. Then select the field node AID that has the thermistor-resistor voltage divider installed (*Node 1* is shown in Figure 6.0.1). The *Units* menu selects Fahrenheit or Celsius for the temperature display.

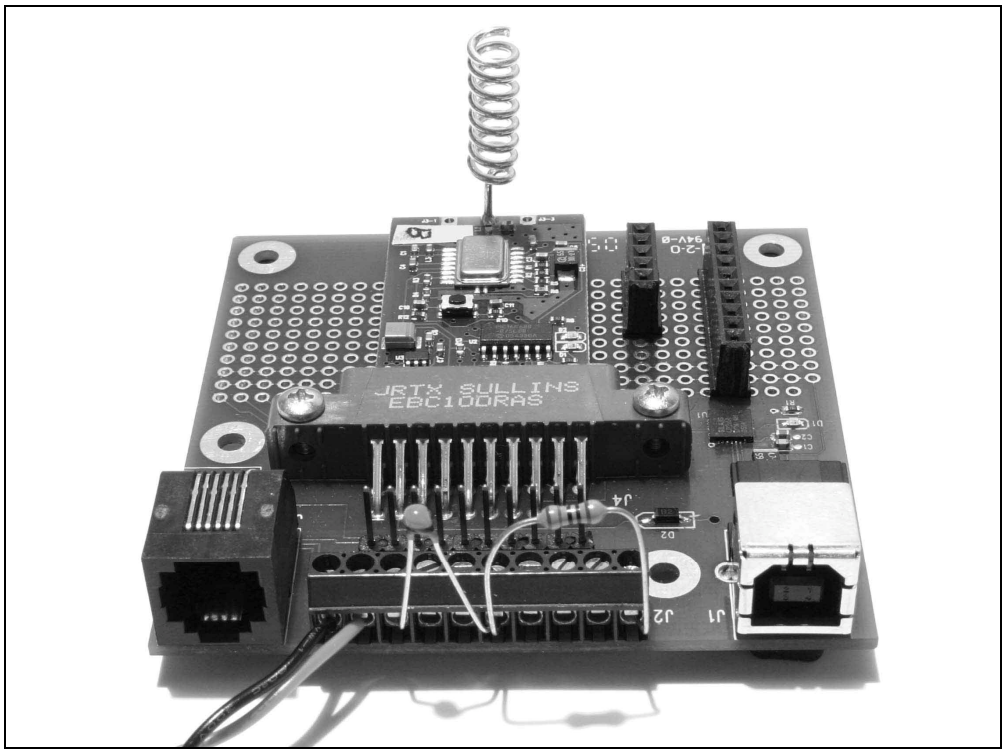


Figure 6.0.3: IM1800 Thermistor-Resistor Installation

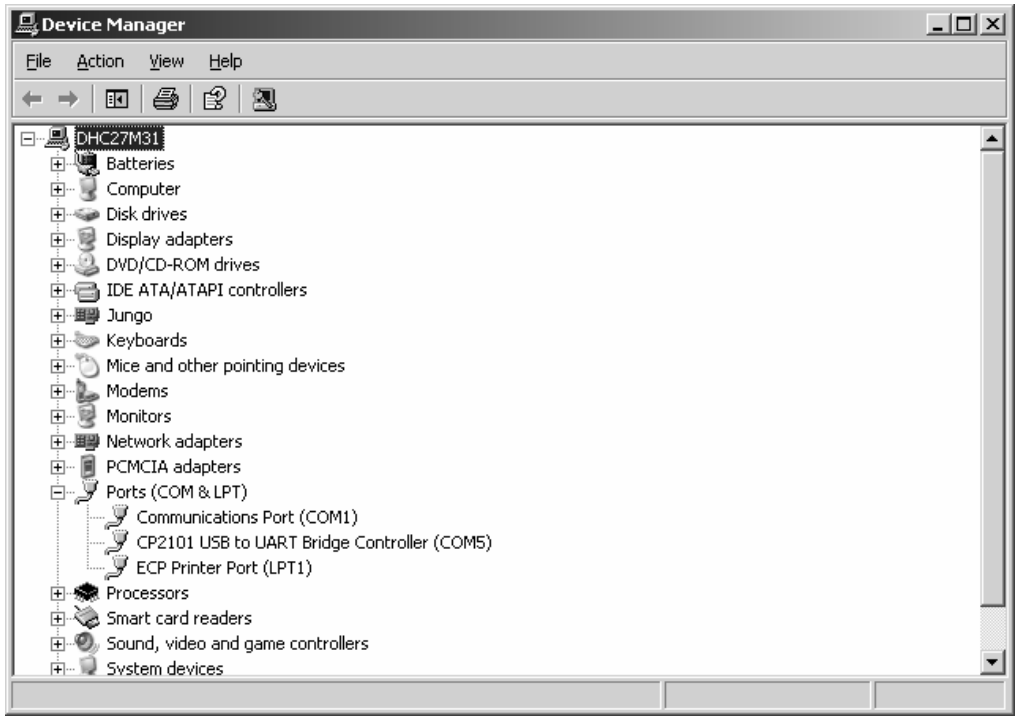


Figure 6.0.4: COM Port Verification

6.1 Temperature Sensor Application Source Code

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Begin VB.Form Form1
   Caption         =   "DM1800/DM1810 Thermistor Demo"
   ClientHeight    =   945
   ClientLeft      =   165
   ClientTop       =   735
   ClientWidth     =   5655
   ForeColor       =   &H00000000&
   LinkTopic       =   "Form1"
   ScaleHeight     =   945
   ScaleWidth      =   5655
   StartUpPosition =   3  'Windows Default
   Begin MSCommLib.MSComm MSComm1
      Left            =   4920
      Top             =   120
      _ExtentX        =   1005
      _ExtentY        =   1005
      _Version        =   393216
      DTREnable       =   -1  'True
   End
   Begin VB.Timer Timer1
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   4440
      Top             =   120
   End
   Begin VB.Shape LED
      BackColor       =   &H00000000&
      BackStyle       =   1  'Opaque
      BorderColor     =   &H80000004&
      FillColor       =   &H00008000&
      FillStyle       =   0  'Solid
      Height          =   255
      Left            =   120
      Shape           =   3  'Circle
      Top             =   240
      Width           =   255
   End
   Begin VB.Label Label1
      BackColor       =   &H00000000&
      BorderStyle     =   1  'Fixed Single
      Caption         =   " Select COM Port"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   13.5
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FF00&
      Height          =   495
      Left            =   480
      TabIndex        =   0
      Top             =   120
      Width           =   5055
   End
```

```
Begin VB.Menu mnuSerial
   Caption         =   "COM Port"
   Begin VB.Menu mnuCOM1
      Caption         =   "COM 1"
   End
   Begin VB.Menu mnuCOM2
      Caption         =   "COM 2"
   End
   Begin VB.Menu mnuCOM3
      Caption         =   "COM 3"
   End
   Begin VB.Menu mnuCOM4
      Caption         =   "COM 4"
   End
   Begin VB.Menu mnuCOM5
      Caption         =   "COM 5"
   End
   Begin VB.Menu mnuCOM6
      Caption         =   "COM 6"
   End
   Begin VB.Menu mnuCOM7
      Caption         =   "COM 7"
   End
   Begin VB.Menu mnuCOM8
      Caption         =   "COM 8"
   End
End
Begin VB.Menu mnuNode
   Caption         =   "Node"
   Begin VB.Menu mnuNode1
      Caption         =   "Node1"
      Checked         =   -1  'True
   End
   Begin VB.Menu mnuNode2
      Caption         =   "Node 2"
   End
   Begin VB.Menu mnuNode3
      Caption         =   "Node 3"
   End
   Begin VB.Menu mnuNode4
      Caption         =   "Node 4"
   End
   Begin VB.Menu mnuNode5
      Caption         =   "Node 5"
   End
   Begin VB.Menu mnuNode6
      Caption         =   "Node 6"
   End
   Begin VB.Menu mnuNode7
      Caption         =   "Node 7"
   End
   Begin VB.Menu mnuNode8
      Caption         =   "Node 8"
   End
   Begin VB.Menu mnuNode9
      Caption         =   "Node 9"
   End
   Begin VB.Menu mnuNode10
      Caption         =   "Node 10"
   End
End
Begin VB.Menu mnuUnits
   Caption         =   "Units"
```

```
      Begin VB.Menu mnuF
         Caption         =   "Deg F"
         Checked         =   -1  'True
      End
      Begin VB.Menu mnuC
         Caption         =   "Deg C"
      End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

' DM1800R01.FRM, 2007.08.25 @ 17:50 CDT
' DM1800/DM1810 tutorial software
' Thermistor temperature sensor demo
' NO representation is made that this
' software is suitable for any purpose
' Compiled in Microsoft Visual Basic 6.0

' declare form variables:
  Dim DLE$                                          ' bi-sync DLE
  Dim STX$                                          ' bi-sync STX
  Dim ETX$                                          ' bi-sync ETX
  Dim RdADC$                                        ' ADC read command
  Dim DOut$                                         ' digital output high command
  Dim FIFO$                                         ' character FIFO
  Dim RPkt$                                         ' response message
  Dim AID As Byte                                   ' AID address (low byte)
  Dim ADC As Single                                 ' ADC count
  Dim TempArray(32, 2) As Single                    ' temperature conversion array
  Dim TempFF As Variant                             ' formatted temperature, F
  Dim TempCF As Variant                             ' formatted temperature, C
  Dim P As Integer                                  ' COM port
  Dim T As Integer                                  ' temperature F or C select
  Dim S As Integer                                  ' TX state variable
  Dim RPFlag As Integer                             ' response pending flag
  Dim TOut As Integer                               ' response pending time out counter

Private Sub Form_Load()

' initialize form variables:
  DLE$ = Chr$(&H10)                                 ' DLE = &H10
  STX$ = Chr$(&H2)                                  ' STX = &H02
  ETX$ = Chr$(&H3)                                  ' ETX = &H03
  RdADC$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&HA9)  ' read 10-bit ADC command
  DOut$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&H31) _
  + Chr$(&H81) + Chr$(&H0)                          ' digital out high command (IM1800-1
support)
  FIFO$ = ""                                        ' clear FIFO string
  RPkt$ = ""                                        ' clear response string
  AID = 1                                           ' default node address
  ADC = 0                                           ' clear ADC
  Call LoadArray                                    ' load temperature conversion array
  TempFF = 0                                        ' clear TempFF
  TempCF = 0                                        ' clear TempCF
  P = 1                                             ' default COM port number
  T = 0                                             ' display default deg F
  S = 0                                             ' initial state
  RPFlag = 0                                        ' clear response pending flag
  TOut = 0                                          ' clear time out counter
```

```
  Form1.Left = (Screen.Width - Form1.Width) / 1.2          ' put form at
  Form1.Top = (Screen.Height - Form1.Height) / 5           ' upper right

  LED.FillColor = &HC000&                                  ' green LED OFF

  Timer1.Interval = 1000                                   ' set timer interval to 1 second
  Timer1.Enabled = False                                   ' start timer when COM port selected

End Sub

Private Sub Timer1_Timer()
  LED.FillColor = &HC000&                                  ' green LED OFF
  Label1.Caption = " Temperature:  "                       ' default caption
  If MSComm1.InBufferCount > 0 Then                        ' if bytes in COM input buffer
    Call GetRX                                             ' call to process RX bytes
  End If
  Call TXCmd                                               ' call to send TX command
End Sub

Public Sub TXCmd()
  If (RPFlag = 0) Then                                     ' if no response pending
    RPFlag = 1                                             ' set response pending flag
    TOut = 4                                               ' initialize 4 s response time out
    If (S = 0) Then                                        ' if TX state 0
      Mid(DOut$, 5, 1) = Chr$(AID)                         ' update command address
      Call SndTX(DOut$)                                    ' send digital out high command
      S = S + 1                                            ' increment state
    ElseIf (S = 1) Then                                    ' else if TX state = 1
      Mid(RdADC$, 5, 1) = Chr$(AID)                        ' update command address
      Call SndTX(RdADC$)                                   ' send read ADC command
    End If
  Else                                                     ' else
    TOut = TOut - 1                                        ' decrement response pending time out
    If (TOut <= 0) Then                                    ' if time out reaches 0
      RPFlag = 0                                           ' reset response pending flag
      Label1.Caption = " No response"                      ' display message
    End If
  End If
End Sub

Public Sub SndTX(Cmd$)
  Dim N As Integer                                         ' command length
  Dim M As Integer                                         ' sum
  Dim Y As Integer                                         ' loop counter
  Dim CK As Byte                                           ' checksum
  Dim C$                                                   ' character buffer
  Dim GTPkt$                                               ' bi-sync framed command
  N = Len(Cmd$)                                            ' get length of command
  M = 0                                                    ' clear sum
  For Y = 1 To N                                           ' loop to add Cmd$ bytes
    M = M + Asc(Mid$(Cmd$, Y, 1))                          ' add value of each byte to sum
  Next Y
  CK = CByte(M Mod 256)                                    ' checksum is lower byte of sum
  GTPkt$ = DLE$ + STX$                                     ' start GTPkt$ with DLE$ + STX$
  For Y = 1 To N                                           ' loop to build GTPkt$ from Cmd$
    C$ = Mid$(Cmd$, Y, 1)                                  ' load C$ with next Cmd$ character
    If (C$ = DLE$) Then                                    ' if character is DLE$
      GTPkt$ = GTPkt$ + C$ + DLE$                          ' add character plus DLE$ escape
    Else                                                   ' else
      GTPkt$ = GTPkt$ + C$                                 ' just add character
    End If
  Next Y
  If (Chr$(CK) = DLE$) Then                                ' if checksum is DLE$
    GTPkt$ = GTPkt$ + Chr$(CK) + DLE$                      ' add checksum plus DLE$ escape
```

```
  Else                                                  ' else
    GTPkt$ = GTPkt$ + Chr$(CK)                          ' just add checksum
  End If
  GTPkt$ = GTPkt$ + DLE$ + ETX$                         ' complete GTPkt$ with DLE$ + ETX$
  MSComm1.Output = GTPkt$                               ' send GTPkt$ command to read ADC
End Sub

Public Sub GetRX()
  Dim J As Integer                                      ' RPkt$ string position
  Dim K As Integer                                      ' current message length
  Dim Q As Integer                                      ' FIFO$ length
  Dim R As Integer                                      ' DLE$ flag
  Dim U As Integer                                      ' build message flag
  Dim Y As Integer                                      ' FOR loop counter
  Dim ComData$                                          ' COM input string
  Dim C$                                                ' character buffer
  RPkt$ = ""                                            ' clear RPkt$
  J = 1                                                 ' point J to first character position
  K = 0                                                 ' clear current message length
  R = 0                                                 ' clear DLE$ flag
  U = 0                                                 ' clear build message flag
  On Error Resume Next                                  ' set up COM error handler

  ComData$ = MSComm1.Input                              ' load ComData$ from COM input buffer
  FIFO$ = FIFO$ + ComData$                              ' add ComData$ to end of FIFO$ string
  Q = Len(FIFO$)                                        ' load Q with number of characters in
FIFO$
  If (Q > 256) Then                                     ' if FIFO$ is overflowing (garbage)
    FIFO$ = ""                                          ' clear FIFO$
    Exit Sub                                            ' and exit
  End If
  If (Q < 2) Then                                       ' if FIFO$ less than 2 characters
    Exit Sub                                            ' exit
  End If
  For Y = Q To 1 Step -1                                ' else loop to scan all FIFO$ charac-
ters
    C$ = Mid$(FIFO$, J, 1)                              ' load C$ with next character
    If (R = 0) And (C$ = DLE$) Then                     ' if character is a DLE$
      R = 1                                             ' set DLE$ flag
    ElseIf (R = 1) And (C$ = STX$) Then                 ' else if DLE$ + STX$ sequence
      R = 0                                             ' clear DLE$ flag
      U = 1                                             ' set build message flag
      RPkt$ = ""                                        ' clear RPkt$
    ElseIf (R = 0) And (U = 1) Then                     ' else if build flag set and DLE$
flag clear
      RPkt$ = RPkt$ + C$                                ' add character to message
    ElseIf (R = 1) And (U = 1) And (C$ = DLE$) Then     ' else if DLE$ + DLE$ sequence
      R = 0                                             ' reset DLE$ flag
      RPkt$ = RPkt$ + C$                                ' add only one DLE$ to RPkt$
    ElseIf (R = 1) And (U = 1) And (C$ = ETX$) Then     ' else if DLE$ + ETX$ sequence
      R = 0                                             ' clear DLE$ flag
      U = 0                                             ' clear build message flag
      K = J                                             ' load K with current message length
      If RPkt$ <> "" Then                               ' if RPkt$ is not null
        RPFlag = 0                                      ' reset response pending flag
        Call ShowT                                      ' filter for and display temperature
        RPkt$ = ""                                      ' clear RPkt$
      End If
    ElseIf (R = 1) Then                                 ' else if DLE$ sequence not used here
      R = 0                                             ' clear DLE$ flag and move on
    End If
    J = J + 1                                           ' increment character position
  Next Y                                                ' do FOR loop Q times
```

```
  If (K < Q) Then                                   ' if K is less Q after looping
    FIFO$ = Right$(FIFO$, (Q - K))                  ' load FIFO$ with remaining charac-
ters
  Else                                              ' else
    FIFO$ = ""                                      ' clear FIFO$
  End If
End Sub

Public Sub ShowT()
  Dim N As Integer                                  ' command length
  Dim M As Integer                                  ' sum
  Dim Y As Integer                                  ' loop counter
  Dim HB As Integer                                 ' ADC high byte
  Dim LB As Integer                                 ' ADC low byte
  Dim VL As Integer                                 ' total ADC value
  Dim CK As Byte                                    ' checksum
  Dim NN$                                           ' node number
' filter for valid 10-bit ADC response
  N = Len(RPkt$)                                    ' get response length
  M = 0                                             ' clear sum
  For Y = 1 To (N - 1)                              ' for all bytes but received checksum
    M = M + Asc(Mid$(RPkt$, Y, 1))                  ' add bytes to sum
  Next Y
  CK = CByte(M Mod 256)                             ' checksum is lower byte
  If (CK <> Asc(Right$(RPkt$, 1))) Then             ' if bad checksum
    Exit Sub                                        ' exit
  ElseIf (Asc(Mid$(RPkt$, 1, 1)) And &HF0) <> &HE0 Then  ' else if not application response
    Exit Sub                                        ' exit
  ElseIf (Len(RPkt$) <> 10) Then                    ' else if not 10 bytes in length
    Exit Sub                                        ' exit
  ElseIf (Asc(Mid$(RPkt$, 6, 1)) <> &H9) Then       ' else if not valid response
    Exit Sub                                        ' exit
  ElseIf (Asc(Mid$(RPkt$, 7, 1)) And &HC1) <> &H41 Then  ' else if not 10-bit ADC reading
    Exit Sub                                        ' exit
  End If
' calculate and show new temperature
  NN$ = Str(Asc(Mid$(RPkt$, 5, 1)))                 ' get node AID number
  HB = Asc(Mid$(RPkt$, 8, 1))                       ' get high ADC byte
  LB = Asc(Mid$(RPkt$, 9, 1))                       ' get low ADC byte
  VL = 256 * HB + LB                                ' calculate total ADC value
  ADC = CSng(VL)                                    ' covert ADC value to single
  LED.FillColor = &HFF00&                           ' green LED ON
  Call GetTemp                                      ' convert to temperature
  If (T = 0) Then                                   ' if temperature flag clear
    Label1.Caption = " Temperature:  " + TempFF _
    + " F on Node" + NN$                            ' show F temperature
  Else                                              ' else
    Label1.Caption = " Temperature:  " + TempCF _
    + " C on Node" + NN$                            ' show C temperature
  End If
End Sub

Public Sub GetTemp()
  Dim CH As Single                                  ' high ADC interpolation count
  Dim CL As Single                                  ' low ADC interpolation count
  Dim TH As Single                                  ' high temperature interpolation
value
  Dim TL As Single                                  ' low temperature interpolation value
  Dim TempF As Single                               ' raw temperature, F
  Dim TempC As Single                               ' raw temperature, C
  Dim X As Integer                                  ' loop counter
' interpolate from table
```

```
  For X = 2 To 25
    If (ADC <= TempArray(X, 1)) Then
      CH = TempArray(X, 1)
      CL = TempArray((X - 1), 1)
      TH = TempArray(X, 2)
      TL = TempArray((X - 1), 2)
      TempF = TL + (((ADC - CL) * (TH - TL)) / (CH - CL))
      TempFF = Format(TempF, "000.0")
      TempC = (5 / 9) * (TempF - 32)
      TempCF = Format(TempC, "000.0")
      Exit For
    End If
  Next X
  End Sub

Public Sub LoadArray()
  TempArray(1, 1) = 30
  TempArray(1, 2) = -40
  TempArray(2, 1) = 41
  TempArray(2, 2) = -31
  TempArray(3, 1) = 55
  TempArray(3, 2) = -22
  TempArray(4, 1) = 74
  TempArray(4, 2) = -13
  TempArray(5, 1) = 96
  TempArray(5, 2) = -4
  TempArray(6, 1) = 124
  TempArray(6, 2) = 5
  TempArray(7, 1) = 157
  TempArray(7, 2) = 14
  TempArray(8, 1) = 196
  TempArray(8, 2) = 23
  TempArray(9, 1) = 241
  TempArray(9, 2) = 32
  TempArray(10, 1) = 290
  TempArray(10, 2) = 41
  TempArray(11, 1) = 343
  TempArray(11, 2) = 50
  TempArray(12, 1) = 398
  TempArray(12, 2) = 59
  TempArray(13, 1) = 455
  TempArray(13, 2) = 68
  TempArray(14, 1) = 512
  TempArray(14, 2) = 77
  TempArray(15, 1) = 567
  TempArray(15, 2) = 86
  TempArray(16, 1) = 619
  TempArray(16, 2) = 95
  TempArray(17, 1) = 668
  TempArray(17, 2) = 104
  TempArray(18, 1) = 712
  TempArray(18, 2) = 113
  TempArray(19, 1) = 753
  TempArray(19, 2) = 122
  TempArray(20, 1) = 788
  TempArray(20, 2) = 131
  TempArray(21, 1) = 820
  TempArray(21, 2) = 140
  TempArray(22, 1) = 847
  TempArray(22, 2) = 149
  TempArray(23, 1) = 871
  TempArray(23, 2) = 158
  TempArray(24, 1) = 892
  TempArray(24, 2) = 167
```

```
  TempArray(25, 1) = 910
  TempArray(25, 2) = 176
  TempArray(26, 1) = 925
  TempArray(26, 2) = 185
End Sub

Private Sub mnuCOM1_Click()
  P = 1                                          ' COM port 1
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM1.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM2_Click()
  P = 2                                          ' COM port 2
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM2.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM3_Click()
  P = 3                                          ' COM port 3
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM3.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM4_Click()
  P = 4                                          ' COM port 4
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM4.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM5_Click()
  P = 5                                          ' COM port 5
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM5.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM6_Click()
  P = 6                                          ' COM port 6
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM6.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM7_Click()
  P = 7                                          ' COM port 7
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM7.Checked = True                         ' set this checkmark
End Sub

Private Sub mnuCOM8_Click()
  P = 8                                          ' COM port 8
  Call SetCOM                                    ' open COM port
  Call ResetCOM                                  ' clear all checkmarks
  mnuCOM8.Checked = True                         ' set this checkmark
End Sub

Public Sub SetCOM()
  On Error GoTo Skip                              ' invalid COM port recovery
```

```
  If (MSComm1.PortOpen = True) Then                    ' if COM port open
    MSComm1.PortOpen = False                           ' close it
    Timer1.Enabled = False                             ' stop timer
  End If
  MSComm1.CommPort = P                                 ' initialize COM port P
  MSComm1.Settings = "9600,N,8,1"                      ' at 9600 b/s
  MSComm1.RThreshold = 0                               ' poll only, no interrupts
  MSComm1.InputLen = 0                                 ' read all bytes
  MSComm1.PortOpen = True                              ' open COM port
  Label1.Caption = " Initializing  "                   ' update display
  Timer1.Enabled = True                                ' start timer
  Exit Sub
Skip:
  Label1.Caption = " Invalid COM  "                    ' update display
End Sub

Public Sub ResetCOM()
    mnuCOM1.Checked = False                            ' clear checkmark
    mnuCOM2.Checked = False                            ' clear checkmark
    mnuCOM3.Checked = False                            ' clear checkmark
    mnuCOM4.Checked = False                            ' clear checkmark
    mnuCOM5.Checked = False                            ' clear checkmark
    mnuCOM6.Checked = False                            ' clear checkmark
    mnuCOM7.Checked = False                            ' clear checkmark
    mnuCOM8.Checked = False                            ' clear checkmark
End Sub

Private Sub mnuNode1_Click()
  AID = 1                                              ' node 1
  S = 0                                                ' reset TX state
  Call ResetNode                                       ' clear all checkmarks
  mnuNode1.Checked = True                              ' set this checkmark
End Sub

Private Sub mnuNode2_Click()
  AID = 2                                              ' node 2
  S = 0                                                ' reset TX state
  Call ResetNode                                       ' clear all checkmarks
  mnuNode2.Checked = True                              ' set this checkmark
End Sub

Private Sub mnuNode3_Click()
  AID = 3                                              ' node 3
  S = 0                                                ' reset TX state
  Call ResetNode                                       ' clear all checkmarks
  mnuNode3.Checked = True                              ' set this checkmark
End Sub

Private Sub mnuNode4_Click()
  AID = 4                                              ' node 4
  S = 0                                                ' reset TX state
  Call ResetNode                                       ' clear all checkmarks
  mnuNode4.Checked = True                              ' set this checkmark
End Sub

Private Sub mnuNode5_Click()
  AID = 5                                              ' node 5
  S = 0                                                ' reset TX state
  Call ResetNode                                       ' clear all checkmarks
  mnuNode5.Checked = True                              ' set this checkmark
End Sub
```

```
Private Sub mnuNode6_Click()
  AID = 6                                          ' node 6
  S = 0                                            ' reset TX state
  Call ResetNode                                   ' clear all checkmarks
  mnuNode6.Checked = True                          ' set this checkmark
End Sub

Private Sub mnuNode7_Click()
  AID = 7                                          ' node 7
  S = 0                                            ' reset TX state
  Call ResetNode                                   ' clear all checkmarks
  mnuNode7.Checked = True                          ' set this checkmark
End Sub

Private Sub mnuNode8_Click()
  AID = 8                                          ' node 8
  S = 0                                            ' reset TX state
  Call ResetNode                                   ' clear all checkmarks
  mnuNode8.Checked = True                          ' set this checkmark
End Sub

Private Sub mnuNode9_Click()
  AID = 9                                          ' node 9
  S = 0                                            ' reset TX state
  Call ResetNode                                   ' clear all checkmarks
  mnuNode9.Checked = True                          ' set this checkmark
End Sub

Private Sub mnuNode10_Click()
  AID = 10                                         ' node 10
  S = 0                                            ' reset TX state
  Call ResetNode                                   ' clear all checkmarks
  mnuNode10.Checked = True                         ' set this checkmark
End Sub

Public Sub ResetNode()
    mnuNode1.Checked = False                       ' clear checkmark
    mnuNode2.Checked = False                       ' clear checkmark
    mnuNode3.Checked = False                       ' clear checkmark
    mnuNode4.Checked = False                       ' clear checkmark
    mnuNode5.Checked = False                       ' clear checkmark
    mnuNode6.Checked = False                       ' clear checkmark
    mnuNode7.Checked = False                       ' clear checkmark
    mnuNode8.Checked = False                       ' clear checkmark
    mnuNode9.Checked = False                       ' clear checkmark
    mnuNode10.Checked = False                      ' clear checkmark
End Sub

Private Sub mnuC_Click()
    mnuC.Checked = True                            ' check deg C
    mnuF.Checked = False                           ' uncheck deg F
    T = 1                                          ' show deg C
End Sub

Private Sub mnuF_Click()
    mnuC.Checked = False                           ' uncheck deg C
    mnuF.Checked = True                            ' check deg F
    T = 0                                          ' show deg F
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
  If (MSComm1.PortOpen = True) Then                    ' if COM port open
    MSComm1.PortOpen = False                           ' close it
  End If
  End                                                  ' done
End Sub
```

API Message Index

Message	Page	Message	Page
Base Reset Command	4	Read Base Station Bind List	19
Base AutoSend Command	4	Read User Message A	19
Base Reset Complete	4	Read User Message B	20
Command Accepted	4	Read Node I/O Configuration	20
Command No Op	4	Read Firmware Version	20
Response Timeout	4	Read Hardware ID	20
Network Timeout	4	Read Timer Configuration	21
Response Error	4	Read Router/Field Node Mode & Model Configuration	21
Serial (UART) Error	4	Read Base Station Mode & Model Configuration	21
Invalid Length	4	Read Event Routing Configuration	22
Invalid DLE Sequence	4	Read Router/Field Node Link Map	22
Read ADC Input with 10-bit Resolution	11	Read Base Station Link Map	22
Read ADC Input with 8-bit Resolution	11	Reset Router/Field Node Link Map	23
Read Digital Input	12	Reset Base Station Link Map	23
Read Digital Output	12	Reset Node	23
Read UART Buffer	12	Write ADC Input Configuration	23
Read Dynamic Power Management Parameters	12	Write Digital Input Configuration	24
Write (Reset) Analog Input	13	Write Digital Output Configuration	24
Write (Reset) Digital Input	13	Write UART Configuration	25
Write (Set) Digital Output	13	Write Static Power Management Parameters	25
Write (Send) UART Buffer	14	Write User Message A	25
Write Dynamic Power Management Parameters	14	Write User Message B	25
Event Message	14	Write Timer Configuration	26
Read Status of Event Flags	15	Write Router/Field Node Mode Configuration	26
Power On Notification Message	16	Write Base Station Mode Configuration	26
Read ADC Input Configuration	16	Write Event Routing Configuration	27
Read Router/Field Node RSSI Values	17	Write Router/Field Node Bind Configuration	27
Read Base Station RSSI Values	17	Write Base Station Bind Configuration	27
Read Digital Input Configuration	17	Write Base Station Bind List	28
Read Digital Output Configuration	18	Write Node I/O Configuration	28
Read UART Configuration	18	Read Node Auxiliary Configuration	28
Read Static Power Management Parameters	18	Write Node Auxiliary Configuration	30
Read Router/Field Node Bind Configuration	18	Field Node UART Reset Vector	30
Read Base Station Bind Configuration	19		