# SMARTMESH® M1030-2

## 900 MHz Wireless Analog / Digital / Serial Mote

## Product Description

The SmartMesh® M1030-2 embedded wireless mote uses Time Synchronized Mesh Protocol (TSMP) to enable low-power wireless sensors and actuators with highly reliable wireless mesh networking. The M1030-2 is tailored for use in battery- and line-powered wireless devices for applications that demand proven performance, scalability, and reliability.

The M1030-2 uses a 900 MHz radio to achieve more than 200-meter communication distance outdoors, while consuming down to 40 µA in a typical network deployment. The combination of extremely high reliability and low power consumption enables applications that require very low installation cost and low-maintenance, long-term deployments.

The standard serial interface of the M1030-2 gives it flexibility to be used in a wide variety of different applications, from industrial process control to security, to lighting. When integrated into a product, the M1030-2 acts like a network interface card (NIC)—it takes a data packet and makes sure that it successfully traverses the network. By isolating the wireless mesh networking protocols from the user, the M1030-2 simplifies the development process and reduces development risk.

## Key Features

### Reliable Networking

- Uses a Time Synchronized Mesh Protocol (TSMP) for high reliability (>99.9% typical network reliability)
- Frequency hopping for interference rejection
- Mesh networking for built-in redundancy
- Every M1030-2 acts as both an endpoint and a router, increasing network reliability: “mesh-to-the-edge”
- Automatic self-organizing mesh is built in

### Low Power Consumption

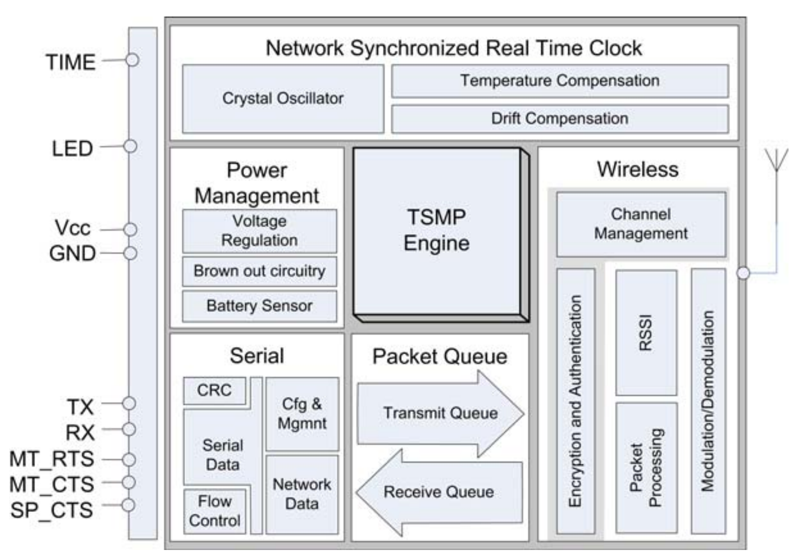
- Ultra-low power components for long battery life
- Network-wide coordination for efficient power usage
- Down to 40 µA typical power consumption

### Efficient Radio

- 2.5 mW (+4 dBm) RF output power
- –89 dBm receiver sensitivity
- Outdoor range >200 m typical

### Predictable Integration

- Standard HDLC serial interface with flow control in the receive direction
- FCC modular certification
- Industrial temperature range –40 to +85° C
- Supports socket or solder assembly
- Rugged design for class I div I environments




# Contents


**1.0 Absolute Maximum Ratings ........ 3**
**2.0 Normal Operating Conditions ........ 3**
**3.0 Electrical Specifications ........ 4**
3.1 Application Circuit ........ 4
**4.0 Radio ........ 5**
4.1 Detailed Radio Specifications ........ 5
4.2 Antenna Specifications ........ 5
**5.0 Pinout ........ 6**
**6.0 Interfaces ........ 6**
6.1 Status LED ........ 6
6.2 Digital I/O ........ 6
6.3 Analog Inputs ........ 7
6.4 Voltage Reference ........ 7
6.5 Serial Interface ........ 7
6.5.1 Serial Flow Control ........ 7
6.5.1.1 Serial Port ........ 7
6.5.1.2 Serial Interface Boot Up ........ 7
6.5.1.3 Timing Values ........ 8
6.5.2 Mote Command Data Types ........ 8
6.5.3 Mote Commands ........ 8
6.5.3.1 Command 0x80 Serial Payload Sent to Mote Serial ........ 9
6.5.3.2 Command 0x81 Unacknowledged Serial Payload Received from Mote Serial ........ 9
6.5.3.3 Command 0x82 Acknowledged Serial Payload Received from Mote Serial ........ 10
6.5.3.4 Command 0x84 Time/State Packet ........ 10
6.5.3.5 Commands 0x87 and 0x88 Set Parameter Request/Response ........ 10
6.5.3.6 Commands 0x89 and 0x8A Get Parameter Request/Response ........ 11
6.5.3.7 Command 0x8C Mote Information ........ 11
6.5.3.8 Command 0x8D Reset Mote ........ 11
6.5.4 Mote Get/Set Command Parameters ........ 12
6.5.4.1 Error Codes ........ 12
6.5.4.2 Parameter Type 0x01 Network ID ........ 12
6.5.4.3 Parameter Type 0x02 Mote State ........ 13
6.5.4.4 Parameter Type 0x03 Frame Length ........ 14
6.5.4.5 Parameter Type 0x04 Join Key ........ 14
6.5.4.6 Parameter Type 0x05 Time/State ........ 14
6.5.4.7 Parameter Type 0x07 Mote information ........ 15
6.5.5 HDLC Packet Processing Examples ........ 16
**7.0 Packaging Description ........ 18**
7.1 Mechanical Drawings ........ 18
7.2 Soldering Information ........ 19
**8.0 Regulatory and Standards Compliance ........ 19**
8.1 FCC Compliance ........ 19
8.1.1 FCC Testing ........ 19
8.1.2 FCC-approved Antennae ........ 20
8.1.3 OEM Labeling Requirements ........ 20
8.2 IC Compliance ........ 20
8.3 Industrial Environment Operation ........ 20
**9.0 Ordering Information ........ 20**

# 1.0 Absolute Maximum Ratings

The absolute maximum ratings shown below should under no circumstances be violated. Permanent damage to the device may be caused by exceeding one or more of these parameters.

**Table 1 Absolute Maximum Ratings**

| Parameter | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|
| Supply voltage (Vcc to GND) | -0.3 | | 3.6 | V | |
| Voltage on digital I/O pin | –0.3 | | $V_{CC}$ +0.3 up to 3.6 | V | |
| Input RF level | | | 10 | dBm | Input power at antenna connector |
| Storage temperature range | –45 | | +85 | °C | |
| Lead temperature | | | +230 | °C | For 10 seconds |
| VSWR of antenna | | | 3:1 | | |
| * All voltages are referenced to GND | | | | | |

The M1030-2 can withstand an electrostatic discharge of up to 2 kV Human Body Model (HBM) or 200 V Machine Model (MM) applied to any header pin, except the antenna connector. The antenna input can withstand a discharge of 50 V.

# 2.0 Normal Operating Conditions

**Table 2 Normal Operating Conditions**

| Parameter | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|
| Operational supply voltage range (between Vcc and GND) | 2.7 | | 3.3 | V | Including noise and load regulation |
| Voltage on analog input pins | 0 | | 1.5 | V | |
| Voltage supply noise | | | 10<br>20<br>10 | mVp-p | 50 Hz–10 kHz<br>10 kHz–200 kHz<br>200 kHz–15 MHz |
| Peak current | | | 38<br>18 | mA<br>mA | TX, 14 ms maximum<br>Searching for network, 60 minutes |
| Average current | | 40 | | µA | Assuming 40-byte packets, 1 per minute, data only mote |
| Storage and operating temperatures | –40 | | 85 | °C | |
| Maximum allowed temperature ramp | | | 8 | °C/min | –40° C to 85° C |

Unless otherwise noted, Table 3 assumes Vcc is 3.0 V and temperature is 25° C.

**Table 3 Current Consumption**

| Parameter | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|
| Transmit | | 28 | 32 | mA | |
| Receive | | 13 | 14 | mA | |
| Sleep | | 4 | 8 | µA | |

# 3.0 Electrical Specifications

Unless otherwise noted, Vcc is 3.0 V and temperature is –40 to +85° C.

**Table 4 Digital I/O**

| Digital signal | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|
| $V_{IH}$ (logical high input) | Vcc x 80% | Vcc | Vcc + 0.3 | V | |
| $V_{IL}$ (logical low input) | GND –0.3 | GND | GND + 0.6 | V | |
| $V_{OH}$ (logical high output) | 0.7 x Vcc | Vcc | Vcc | V | |
| $V_{OL}$ (logical low output) | GND | GND | 0.25 x Vcc | V | |
| Digital current* | | | | | |
| Output source (single pin) | | | 0.9 | mA | 25°C |
| Output sink (single pin) | | | 0.9 | mA | 25°C |
| Input leakage current | | | 50 | nA | |

* This current level guarantees that the output voltage meets $V_{OL}$ of 0.25 x Vcc and $V_{OH}$ of 0.7 x Vcc.

## 3.1 Application Circuit

The following schematic shows how the M1030-2 mote is used in a circuit.

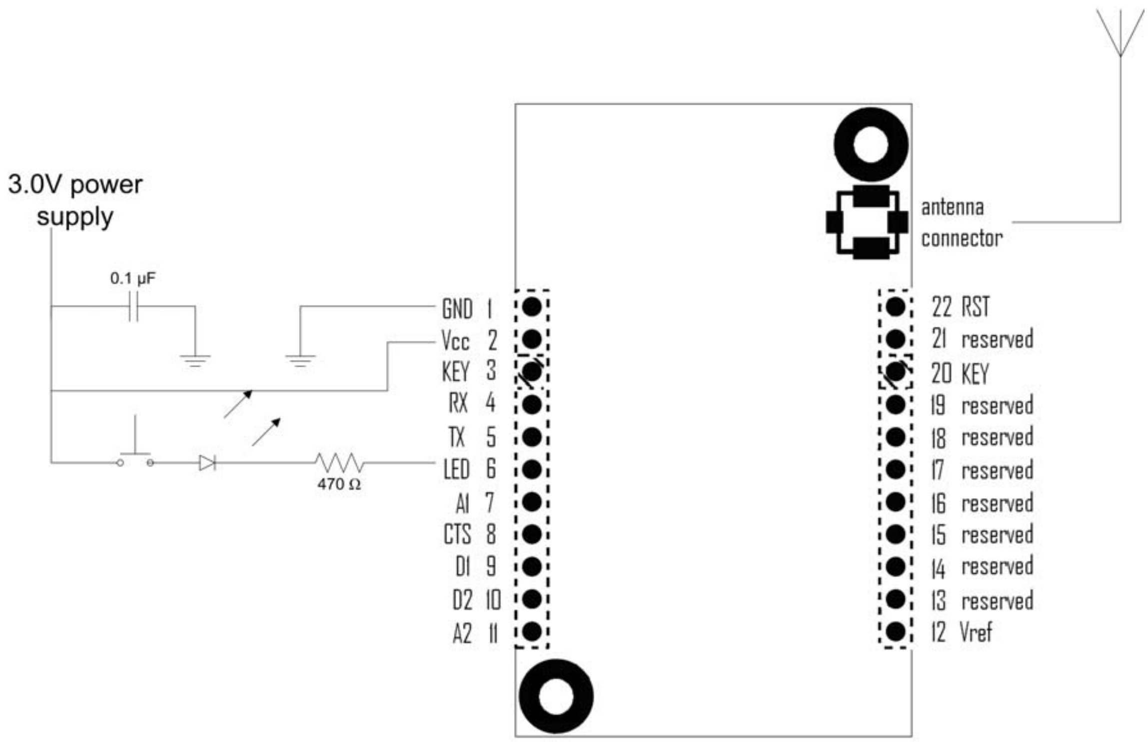



**Figure 1 M1030-2 Mote in Application Circuit**

# 4.0 Radio

## 4.1 Detailed Radio Specifications

**Table 5 Radio Specifications**

| Parameter | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|
| Operating frequency | 902 | | 928 | MHz | |
| Number of channels | | 50 | | | |
| Channel separation | | 470 | | kHz | |
| Channel bandwidth | | 170 | | kHz | At ±20 dBc |
| Modulation | | | | | Binary FSK (NRZ) |
| Raw data rate | | 76.8 | | kbps | |
| Receiver sensitivity | –88 | –89 | | dBm | At $10^{-3}$ BER, Vcc = 3 V, 25° C |
| Output power | +2.5 | +4 | | dBm | Vcc = 3 V, 25° C |
| Range* | | | | | |
| Indoor | | 80 | | m | |
| Outdoor | | 200 | | m | 25° C, 50% RH, 1 meter above ground, +2 dBi omni-directional antenna |

* Actual RF range performance is subject to a number of installation-specific variables including, but not restricted to ambient temperature, relative humidity, presence of active interference sources, line-of-sight obstacles, near-presence of objects (for example, trees, walls, signage, and so on) that may induce multipath fading. As a result, actual performance varies for each instance.

## 4.2 Antenna Specifications

A MMCX-compatible male connector is provided on board for the antenna connection. The antenna must meet specifications in Table 6. For a list of FCC-approved antennae see 8.1.2.

**Table 6 Antenna Specifications**

| Parameter | Value |
|---|---|
| Frequency range | 902-928 MHz |
| Impedance | 50 Ω |
| Gain | +2 dBi maximum |
| Pattern | Omni-directional |
| Maximum VSWR | 3:1 |
| Connector | MMCX* |

* The M1030-2 can accommodate the following RF mating connectors:
- MMCX straight connector such as Johnson 135-3402-001, or equivalent
- MMCX right angle connector such as Tyco 1408149-1, or equivalent

When the mote is placed inside an enclosure, the antenna should be mounted such that the radiating portion of the antenna protrudes from the enclosure, and connected using a MMCX connector on a coaxial cable. For optimum performance, allow the antenna to be positioned vertically when installed.

# 5.0 Pinout

The M1030-2 has two 11-pin Samtec MTMM-111-04-S-S-175-3 (or equivalent) connectors on the bottom side for handling all of the I/O. The third pin in each of the connectors is not populated, and serves as a key for alignment. The connectors are mounted on opposite edges of the long axis of the M1030-2.

The M1030-2 serial interface (serial protocol is specified in 6.5.1) provides flow control in the receive direction only.

**Figure 2 M1030-2 Package with Pin Labels**

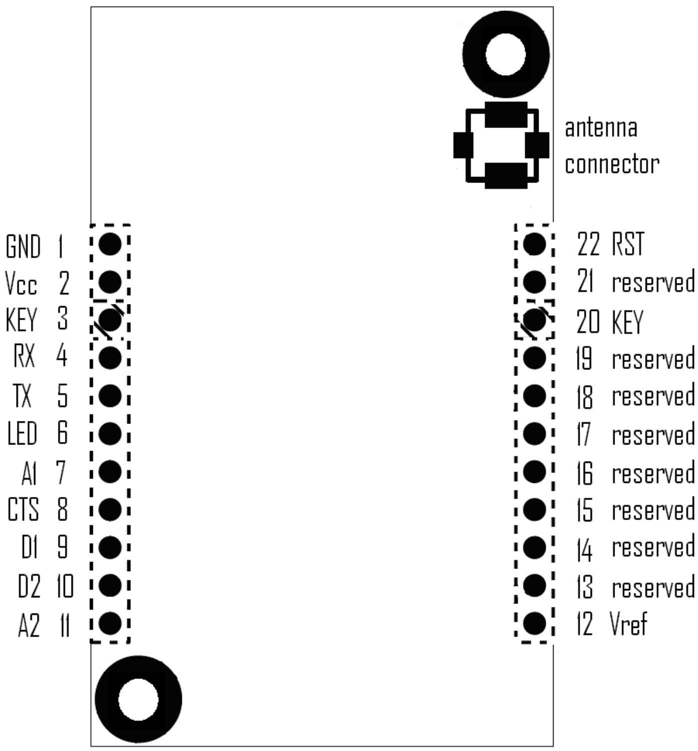



**Table 7 M1030-2 Pin Functions**

| Pin Number | Name | Mote I/O Direction |
|---|---|---|
| 1 | GND | In |
| 2 | VCC | In |
| 3 | KEY (no pin) | - |
| 4 | RX | In |
| 5 | TX | Out |
| 6 | LED | Out |
| 7 | A1 | In |
| 8 | CTS | Out |
| 9 | D1 | Out |
| 10 | D2 | In |
| 11 | A2 | In |
| 12 | VRef | Out |
| 13 | *Reserved* | - |
| 14 | *Reserved* | - |
| 15 | *Reserved* | - |
| 16 | *Reserved* | - |
| 17 | *Reserved* | - |
| 18 | *Reserved* | - |
| 19 | *Reserved* | - |
| 20 | KEY (no pin) | - |
| 21 | *Reserved* | - |
| 22 | RST | In |

# 6.0 Interfaces

## 6.1 Status LED

The M1030-2 provides an output signal driving a status LED. This LED displays network connectivity information and is used during mote installation. Alternatively, the mote's network status may be polled via serial using the Get Parameter command (see 6.5.3.6) with the mote state parameter (see 6.5.4.3).

**Table 8 Status LED**

| LED Appearance | Mote State |
|---|---|
| Off | Off, or in sleep mode |
| Slow single blink (100 ms on, 900 ms off) | On, and searching for potential network |
| Single blink (100 ms on, 400 ms off) | On, and attempting to join network |
| Double blink (100 ms on, 100 ms off, 100 ms on, 700 ms off) | On, connected to network, attempting to establish redundant links |
| Solid on | On, fully configured into network with redundant parents |

## 6.2 Digital I/O

The M1030-2 supports one digital input (D2) and one digital output (D1).

## 6.3 Analog Inputs

The M1030-2 supports two analog inputs, A1 and A2.

## 6.4 Voltage Reference

The M1030-2 provides a voltage reference output to allow for ratiometric sensors.

## 6.5 Serial Interface

The M1030-2 offers a well-defined serial interface that is optimized for low-powered embedded applications. This serial interface offers a serial port comprised of the data pins (TX, RX) as well as the flow control pin, CTS. Through this port, the M1030-2 provides a means of transmitting and receiving serial data through the wireless network, as well as a command interface which provides synchronized time stamping, local configuration and diagnostics.

The following sections detail the Serial Interface Protocol, the Mote Command Interface, and the timestamping capability of the M1030-2 serial interface.

### 6.5.1 Serial Flow Control

The Serial Interface Protocol provides for flow control of packets flowing into the M1030-2 serial interface. Packet delineation and error control are handled separately.

#### 6.5.1.1 Serial Port

The three-pin serial port is comprised of the data pins (TX, RX) as well as the CTS flow control pin used to prevent the microprocessor from overflowing the mote. This port supports 4800 bps operation. The CTS signal is active low.

**Table 9 Serial Parameters**

| Parameter | Value |
|---|---|
| **Baud rate** | 4800 |
| **Start bit** | 1 |
| **Data bits** | 8 |
| **Parity** | None |

The following diagram illustrates the pins used in the handshaking protocol:

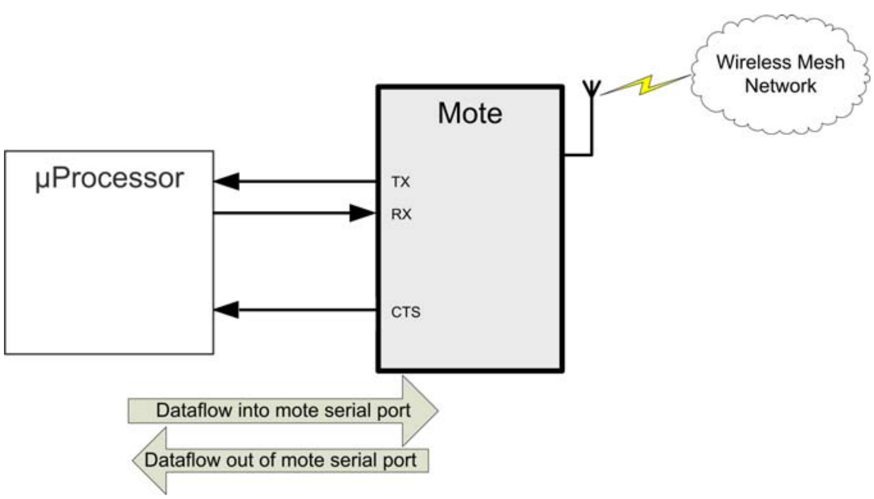



**Figure 3 Diagram of Pins Used in Handshaking Protocol**

#### 6.5.1.2 Serial Interface Boot Up

Upon M1030-2 power up, the CTS line is high (inactive). The M1030-2 serial interface boots within boot_delay (see Tables 10) of the mote powering up, at which time the M1030-2 transmits an HDLC Mote Information packet, as described in section 6.5.3.7.

Once the M1030-2 has established wireless network connection, it uses the CTS pin to signify availability to accept serial packets for wireless transmission. At certain critical times during communication, the M1030-2 may bring CTS high. CTS remains high if the M1030-2 does not have enough buffer space to accept another packet. It also remains high if the mote is not part of the network. Sensor processors must check that the CTS pin is low before initiating each serial packet for wireless transmission. Note that the M1030-2 may receive diagnostic serial packets at any time regardless of the CTS state.

#### 6.5.1.3 Timing Values

**Table 10 Timing Values**

| Variable | Meaning | Min | Max | Unit |
|---|---|---|---|---|
| **diag_ack_timeout** | The mote responds to all requests within this time. | N/A | 100 | ms |
| **boot_delay** | The time between mote power up and serial interface availability. | | 250 | ms |

### 6.5.2 Mote Command Data Types

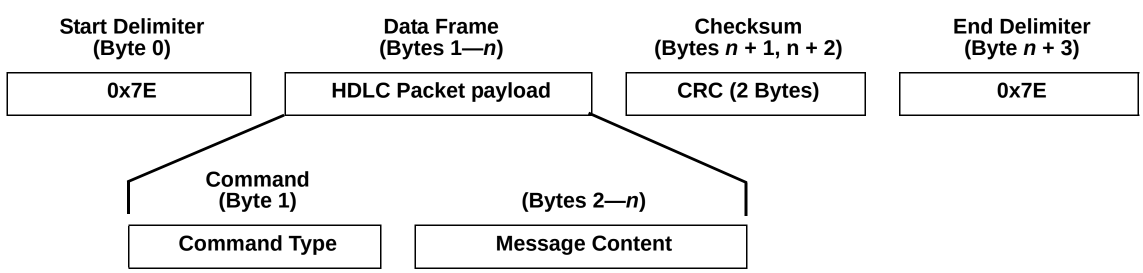
Table 11 defines the command data types used in the commands.

**Table 11 Command Data Types**

| Data Type | Description |
|---|---|
| ULong | Unsigned long: 4 bytes |
| UShort | Unsigned short: 2 bytes |
| UChar | Unsigned character: 1 byte character |

### 6.5.3 Mote Commands

The mote command interface provides a way to send and receive network packets, access local configuration and diagnostics, and receive time stamps. All packets between the microcontroller and the mote are encapsulated in the HDLC format (RFC 1662) and have the following structure.




The command type indicates which API message is contained in the message content. The message content for each command type is described within the following sections.

CRC is calculated based on 16-bit FCS computation method (RFC 1622). The mote checks the CRC and drops packets that have CRC errors. There is no mechanism for the mote to tell the microcontroller that a packet has been discarded, so the applications layer must implement reliable delivery, if desired. All numerical fields in a packet are in big endian order (MSB first), unless otherwise noted. Section 6.5.5 provides an example of HDLC packet construction and HDLC packet decoding.

Table 12 provides a summary of mote commands, which are described in detail in the following sections.

**Table 12 Mote Command Summary**

| Command Type (HEX) | Direction | Description |
|---|---|---|
| 0x80 | Microcontroller to Mote | Packet destined for the network |
| 0x81 | Mote to Microcontroller | Unacknowledged packet received from the network and destined for microcontroller |
| 0x82 | Mote to Microcontroller | Acknowledged packet received from the network and destined for microcontroller |
| 0x83 | -- | Reserved |
| 0x84 | Mote to Microcontroller | Time and mote state information |
| 0x85 | -- | Reserved |
| 0x86 | -- | Reserved |
| 0x87 | Microcontroller to Mote | “Set Parameter” request |
| 0x88 | Mote to Microcontroller | “Set Parameter” response |
| 0x89 | Microcontroller to Mote | “Get Parameter” request |
| 0x8A | Mote to Microcontroller | “Get Parameter” response |
| 0x8C | Mote to Microcontroller | Mote information |
| 0x8D | Microcontroller to Mote | Reset mote |

#### 6.5.3.1 Command 0x80 Serial Payload Sent to Mote Serial

Serial Data Packets going into the mote serial port use the command type 0x80. Upon reception of the packet, the mote forwards it to the network. The format of the serial packet payload is transparent to the mote. The maximum length of the payload is 80 bytes. There is no response by the mote upon reception of this command.

**Table 13 Command 0x80 Serial Payload to Mote**

| Msg Byte | Description | Data Type | Request (Sent to Mote) |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x80 |
| **2** | | (Transparent to mote) | n bytes of data |
| **2+1** | | (Transparent to mote) | * |
| **2+…** | | (Transparent to mote) | * |

#### 6.5.3.2 Command 0x81 Unacknowledged Serial Payload Received from Mote Serial

Unacknowledged serial data packets going out of the mote serial port use command type 0x81. The network uses this command to send data out through the mote serial interface. Upon receiving this packet from the network, the mote forwards it to the microcontroller without sending acknowledgement to Manager. The format of the serial packet payload is transparent to the mote. The maximum length of the payload is 80 bytes.

**Table 14 Command 0x81 Unacknowledged Serial Payload from Mote**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x81 |
| **2** | | (Transparent to mote) | n bytes of data |
| **2+1** | | (Transparent to mote) | * |
| **2+…** | | (Transparent to mote) | * |

#### 6.5.3.3 Command 0x82 Acknowledged Serial Payload Received from Mote Serial

Acknowledged serial data packets going out of the mote use command type 0x82. The network uses this command to send data out through the mote serial interface. Upon receiving this packet from the network, the mote forwards it to the microcontroller and sends an acknowledgement back to Manager. The format of the serial packet payload is transparent to the mote. The maximum length of the payload is 80 bytes. The microcontroller receives exactly one copy of the message that was sent through the network.

**Table 15 Command 0x82 Acknowledged Serial Payload Downstream**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x82 |
| **2** | | (Transparent to mote) | n bytes of data |
| **2+1** | | (Transparent to mote) | * |
| **2+…** | | (Transparent to mote) | * |

#### 6.5.3.4 Command 0x84 Time/State Packet

Time data packets use the command type 0x84. The time packet includes the network time and the current real time relative to the Manager. The mote sends this response when it receives a “get request” with “time” parameter (described later).

**Table 16 Command 0x84 Time/State Packet**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x84 |
| **2-5** | The sequential number of the frame | ULong | Cycle |
| **6-9** | The offset from start of frame in microseconds | ULong | Offset |
| **10-11** | Frame length in timeslots | UShort | Frame Length |
| **12-15** | UTC time seconds | ULong | Real Time part1 |
| **16-19** | UTC time microseconds | ULong | Real Time part2 |
| **20-23** | Time from the last mote reset in milliseconds | ULong | Mote uptime |
| **24** | Mote state (see Table 31) | UChar | Mote state |

#### 6.5.3.5 Commands 0x87 and 0x88 Set Parameter Request/Response

The Set Parameter command allows the setting of a number of configuration parameters in the mote. When the Set Parameter Request command is sent, the response to the request is sent within the diag_ack_timeout (see Table 10). The command structure for individual Parameter Types and can be found in section 6.5.4. The length of payload '*n*' is dependant on the Parameter type and is specified in the Parameter Data Packet section of this document.

**Table 17 Command 0x87 Set Parameter Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x87 |
| **2** | | UChar | Parameter Type |
| **3** | Data | (Transparent to mote) | n bytes of data |
| **3+1** | Data | (Transparent to mote) | * |
| **3+…** | Data | (Transparent to mote) | * |

.

**Table 18 Command 0x88 Set Parameter Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x88 |
| **2** | | UChar | Parameter Type |
| **3** | | UChar | Error code |
| **3+1** | | UChar | Data Length (0x00) |

#### 6.5.3.6 Commands 0x89 and 0x8A Get Parameter Request/Response

The Get Parameter command allows a number of configuration parameters in the mote to be read by serial. When a Get Parameter Request command is sent, the response to the request is sent within the diag_ack_timeout of 100 ms. The command structure for individual parameter types can be found in section 6.5.4. The length of payload '*n*' depends on the parameter type and is specified in that section. If the error code is not equal to 0, then no data is returned in the response. Error codes are described in Table 26.

**Table 19 Command 0x89 Get Parameter Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x89 |
| **2** | | UChar | Parameter Type |
| **3** | Data | (Transparent to mote) | n bytes of data |
| **3+1** | Data | (Transparent to mote) | * |
| **3+...** | Data | (Transparent to mote) | * |

**Table 20 Command 0x8A Get Parameter Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8A |
| **2** | | UChar | Parameter Type |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length |
| **5** | Data | (Transparent to mote) | n bytes of data<br>(If Error Code != 0) |
| **5+1** | Data | (Transparent to mote) | * |
| **5+...** | Data | (Transparent to mote) | * |
| **5+n** | Data | (Transparent to mote) | * |

#### 6.5.3.7 Command 0x8C Mote Information

The mote sends this packet on power-up, supplying information about mote properties.

**Table 21 Command 0x8C – M1030-2 Information**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8C |
| **2-4** | HW model | Array of 3 UChar | HW model |
| **5-6** | HW revision | Array of 2 UChar | HW revision |
| **7-10** | SW revision | Array of 4 UChar | SW revision |
| **11-18** | MAC address | Array of 8 UChar | MAC addr |
| **19** | Networking type | UChar | 1 = 900 MHz network |
| **20-21** | Network ID | UShort | Network ID |
| **22-29** | Datasheet ID | Array of 8 UChar | Datasheet ID |
| **30-31** | Mote ID | UShort | Mote ID |
| **32** | | | Reserved |

#### 6.5.3.8 Command 0x8D Reset Mote

Upon receiving this command, the mote notifies its neighbors about an upcoming reset, then proceeds to reset itself. The delay to the actual reset depends on the network configuration.

**Table 22 Command 0x8D Reset Mote**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8D |

### 6.5.4 Mote Get/Set Command Parameters

This section specifies the parameters that may be used with the Set and Get Commands. Table 23 provides an overview of the these parameters.

**Table 23 Set and Get Command Parameters**

| Parameter | Set Parameter | Get Parameter | Description |
|---|---|---|---|
| Parameter type 0x01 | X | | Sets the mote's network ID |
| Parameter type 0x02 | | X | Retrieves the mote's current network connection state |
| Parameter type 0x03 | | X | Retrieves the network frame length |
| Parameter type 0x04 | X | | Sets the network join key on the mote |
| Parameter type 0x05 | | X | Retrieves the network time and mote state information |
| Parameter type 0x06 | | -- | Reserved |
| Parameter type 0x07 | | X | Retrieves the mote's properties |

All requests have the following structure:

**Table 24 Request Structure for Parameter Data Packets**

| Command Type | Parameter Type | Data (Optional) |
|---|---|---|
| **1 byte** | 1 byte | Up to 33 bytes |

All replies have the following structure:

**Table 25 Reply Structure for Parameter Data Packets**

| Command Type | Parameter Type | Error Code | Data Length | Data (Optional) |
|---|---|---|---|---|
| **1 byte** | 1 byte | 1 byte | 1 byte | Up to 31 bytes |

Command Types, Parameter types, and error codes are discussed in the following sections. Data length is the number of bytes of following data, set to 0 in case of non-zero error code.

#### 6.5.4.1 Error Codes

**Table 26 Error Codes**

| Number | Error | Description |
|---|---|---|
| **0** | DIAG_NO_ERR | No Command-Specific Errors |
| **1** | DIAG_EXE_ERR | Mote unable to execute command |
| **2** | DIAG_PARAM_ERR | Illegal parameter in the request |

#### 6.5.4.2 Parameter Type 0x01 Network ID

The network ID is the identification number used to distinguish different wireless networks. In order to join a specific network, the mote must have the same network ID as the network Manager. This parameter is only valid for the Set Parameter command. Upon receiving this request, the mote stores the new network ID in its persistent storage area, but continues to use the existing network ID. The mote must be reset in order to begin using the new network ID.

**Table 27 Parameter Type 0x01 Network ID Set Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x87 |
| **2** | Parameter Type | UChar | 0x01 |
| **3-4** | Network ID | UShort | Network ID |

The following packet is sent in response to a request to set the network ID.

**Table 28 Parameter Type 0x01 Network ID Set Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x88 |
| **2** | Parameter Type | UChar | 0x01 |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length (0x00) |

#### 6.5.4.3 Parameter Type 0x02 Mote State

This parameter is only valid for the Get Parameter command and is used to retrieve the mote’s current network connection state (see Table 31).

**Table 29 Parameter Type 0x02 Mote State Get Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x89 |
| **2** | Parameter Type | UChar | 0x02 |

:The following packet is sent in response to a request to retrieve the mote’s current network connection state.

**Table 30 Parameter Type 0x02 Mote State Get Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8A |
| **2** | Parameter Type | UChar | 0x02 |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length (0x01) |
| **5** | | UChar | Mote State |

**Table 31 Mote States**

| State # | Description | Details |
|---|---|---|
| **1** | ACTIVE | The mote has joined the network and is waiting to be configured |
| **2** | JOINING | The mote has sent JOIN request, waiting for ACTIVATE |
| **3** | ACT SEARCH | The mote is actively searching for neighbors |
| **4–5** | PASS SEARCH | The mote is passively searching for neighbors |
| **6** | SYNCHRONIZED | The mote is synchronized to a network, listening in active search |
| **7–8** | RESETTING | The mote is going through the reset process |
| **9** | ONLINE1 | The mote has joined a network and has been fully configured, but has only one parent |
| **10** | ONLINE2 | The mote has joined a network, has been fully configured, and has multiple parents |

#### 6.5.4.4 Parameter Type 0x03 Frame Length

This parameter is only valid for the Get Parameter command and is used to retrieve the frame length of the specified frame ID.

**Table 32 Parameter Type 0x03 Frame Length Get Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x89 |
| **2** | Parameter Type | UChar | 0x03 |
| **3** | | UChar | Frame ID |

The following packet is sent in response to a request to retrieve the frame length.

**Table 33 Parameter Type 0x03 Frame Length Get Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8A |
| **2** | Parameter Type | UChar | 0x03 |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length (0x05) |
| **5** | | UChar | Frame ID |
| **6-9** | Frame Length (ms) | ULong | Frame Length |

#### 6.5.4.5 Parameter Type 0x04 Join Key

The join key is needed to allow an mote on the network. The join key is specific for the network and used for data encryption. This parameter is only valid for the Set Parameter command. Upon receiving this request, the mote stores the new join key in its persistent storage. The mote must be reset in order to begin using the new join key.

**Table 34 Parameter Type 0x04 Join Key Set Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x87 |
| **2** | Parameter Type | UChar | 0x04 |
| **3-18** | New Join Key | Array of 16 UChar | New Join Key |

The following packet is sent in response to a request to set the join key.

**Table 35 Parameter Type 0x04Join Key Set Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x88 |
| **2** | Parameter Type | UChar | 0x04 |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length (0x00) |

#### 6.5.4.6 Parameter Type 0x05 Time/State

This parameter is only valid for the Get Parameter command and is used to request the network time and mote state information. The response to this command returns the same information as Command 0x84 (Time/State Packet), with the only difference being that this command can be solicited using a software Get command, rather than a hardware pin.

**Table 36 Parameter Type 0x05 Time/State Get Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x89 |
| **2** | Parameter Type | UChar | 0x05 |

The following packet is sent in response to a request for the network time and mote state information.

**Table 37 Parameter Type 0x05 Time/State Get Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x8A |
| **2** | Parameter Type | UChar | 0x05 |
| **3** | | UChar | Error code |
| **4** | | UChar | Data Length (0x17) |
| **5-8** | | ULong | Cycle |
| **9-12** | | ULong | Offset (µsec) |
| **13-14** | Frame Length (slots) | UShort | Frame Length |
| **15-18** | UTC Time sec | ULong | UTC Time sec |
| **19-22** | UTC Time µsec | ULong | UTC Time µsec |
| **23-26** | Mote uptime msec | UShort | Mote uptime msec |
| **27** | | UChar | Mote state |

### 6.5.4.7 Parameter Type 0x07 Mote information

This parameter is only valid for the Get Parameter command. It is a diagnostics request that retrieves information about the mote’s properties.

**Table 38 Parameter Type 0x07 Mote Information Get Request**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 0x89 |
| **2** | Parameter Type | UChar | 0x07 |

The following packet is sent in response to a request for information about mote properties.

**Table 39 Parameter Type 0x07 Mote Information Get Response**

| Msg Byte | Description | Data Type | Value |
|---|---|---|---|
| **1** | Cmd Type | UChar | 140 (0x8A) |
| **2** | Parameter Type | UChar | 0x07 |
| **3** | Error Code | UChar | Error Code |
| **4** | Data length | UChar | Data length (0x1F) |
| **5-7** | HW model | Array of 3 UChar | HW model |
| **8-9** | HW revision | Array of 2 UChar | HW revision |
| **10-13** | SW revision | Array of 4 UChar | SW revision |
| **14-21** | MAC address | Array of 8 UChar | MAC addr |
| **22** | Networking type | UChar | 1 = 900 MHz network |
| **23-24** | Network ID | UShort | Network ID |
| **25-32** | Datasheet ID | Array of 8 UChar | Datasheet ID |
| **33-34** | Mote ID | UShort | Mote ID |
| **35** | | | Reserved |
| **36-37** | CRC | UShort | CRC |
| **38** | | UChar | 126 (0x7E) |

### 6.5.5 HDLC Packet Processing Examples

**Example 1: Constructing an HDLC packet to send to the mote**

This example demonstrates how you would construct an HDLC packet to set the network ID value to 125. (All values are in hexadecimal.)

**Step 1** Define HDLC packet payload:

Command type => 87
Parameter => 01
Network ID => 07

| HDLC Packet Payload | |
|---|---|
| **Command Type** | **Message Content** |
| 87 | 01 00 7D |

**Step 2** Calculate CRC:

**a.** Calculate the CRC using CRC-16 algorithm (RFC 1622) on the hexadecimal sequence '87 01 00 7D'. The CRC (including 1's complement) is 74 2F.

**b.** Append CRC to payload, CRC is sent least significant byte first (RFC 1622):

| HDLC Packet Payload | CRC |
|---|---|
| 87 01 00 7D | 2F 74 |

**Step 3** Perform byte stuffing.

To perform byte stuffing, check the HDLC Packet Payload and CRC for instances of "7D" or "7E" and replace as follows:

7D => 7D 5D
7E => 7D 5E

| HDLC Packet Payload (stuffed) | CRC (stuffed) |
|---|---|
| 87 01 00 **7D 5D** | 2F 74 |

**Step 4** Add start and stop delimiters:

Enclose the above in start/stop flags (RFC 1622).

| Start Delimiter | HDLC Packet Payload (stuffed) | CRC (stuffed) | Stop Delimiter |
|---|---|---|---|
| 7E | 87 01 00 7D 5D | 2F 74 | 7E |

Or simply, the hexadecimal sequence:

7E 87 01 00 7D 5D 2F 74 7E

**Example 2: Decoding an HDLC packet received from the mote**

To understand how to decode an HDLC packet sent from the mote, let's assume that the mote received a "get mote information" command, and replied with the following HDLC Packet. (All values are in hexadecimal.)

| Start Byte | HDLC Packet Payload (stuffed) | CRC (stuffed) | Stop Byte |
|---|---|---|---|
| 7E | 8A 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7D 5E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 | 3F 85 | 7E |

**Step 1** (HDLC layer) strip off delimiters:

| HDLC Packet Payload (stuffed) | CRC (stuffed) |
|---|---|
| 8A 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7D 5E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 | 3F 85 |

**Step 2** Remove byte stuffing.

To remove byte stuffing, check for instances of “7D 5D” or “7D 5E” and replace as follows:

7D 5D => 7D
7D 5E => 7E

| HDLC Packet Payload (stuffed) | CRC (stuffed) |
|---|---|
| 8A 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 | 3F 85 |

**Step 3** Confirm CRC.

Calculate the checksum for the HDLC payload.

| HDLC Packet Payload |
|---|
| 8A 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 |

Confirm that the CRC matches the CRC sent with the packet. Because the packet encodes CRC least significant byte first, in this example the calculated CRC should match “85 3F”.

**Step 4** (Application layer) parse HDLC payload content.

The resulting packet payload is as follows:

| HDLC Packet Payload |
|---|
| 8A 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 |

| Command Type | Message Content |
|---|---|
| 8A | 07 00 1F 00 00 5B 00 01 01 06 00 3C 00 00 00 00 00 00 7E C3 01 00 08 30 30 30 5F 45 56 30 31 00 13 00 |

As described in section 6.5.3.6., an 0x8A command with parameter type 0x07 has the following message content structure:

| Param | Error Code | Length | Hw Model | Hw Rev | Sw Rev | MAC | Mote Type | Net ID | Datasheet ID | Mote ID | Rsvd |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07 | 00 | 1F | 00 00 5B | 00 01 | 01 06 00 3C | 00 00 00 00 00 00 7E C3 | 01 | 00 08 | 30 30 30 5F 45 56 30 31 | 00 13 | 00 |

Therefore, this is a Mote Information response with no errors (and a payload length of 31 bytes). The Mote information is as follows:

| | | |
|---|---|---|
| **HW model =** | “00091” | (00 00 5B) |
| **HW Rev =** | “0001” | (00 01) |
| **SW rev =** | “1.6.60” | (01 06 00 3C) |
| **MAC Address =** | 00 00 00 00 00 00 7E C3 | |
| **Mote type =** | 01 = 900 MHz | (01) |
| **Network ID =** | 8 | (00 08) |
| **Datasheet ID =** | “000_EV01” | (30 30 30 5F 45 56 30 31) |
| **Mote ID =** | “19” | (00 13) |

# 7.0 Packaging Description

## 7.1 Mechanical Drawings

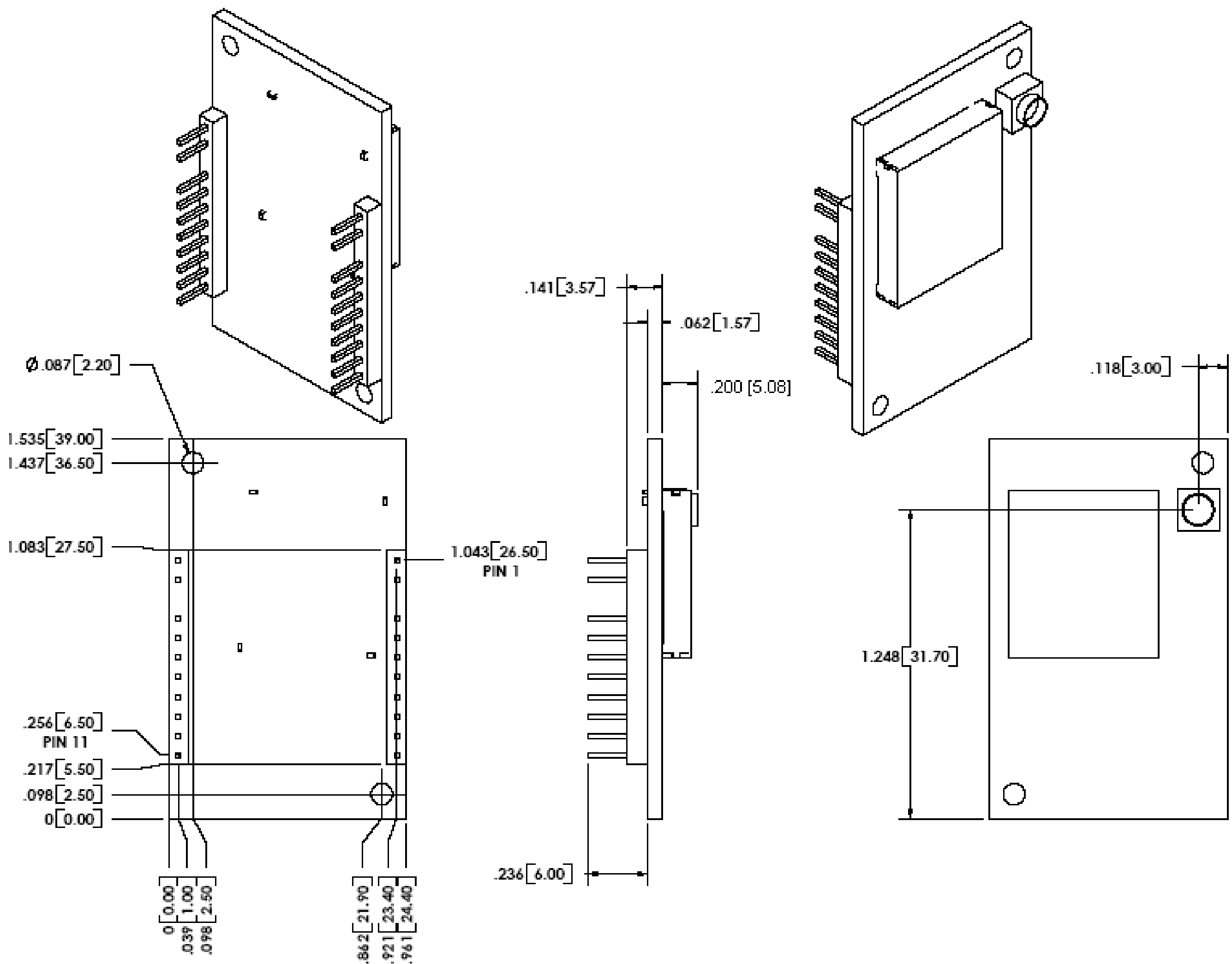



**Figure 4 M1030-2 Mote—Mechanical Drawing**

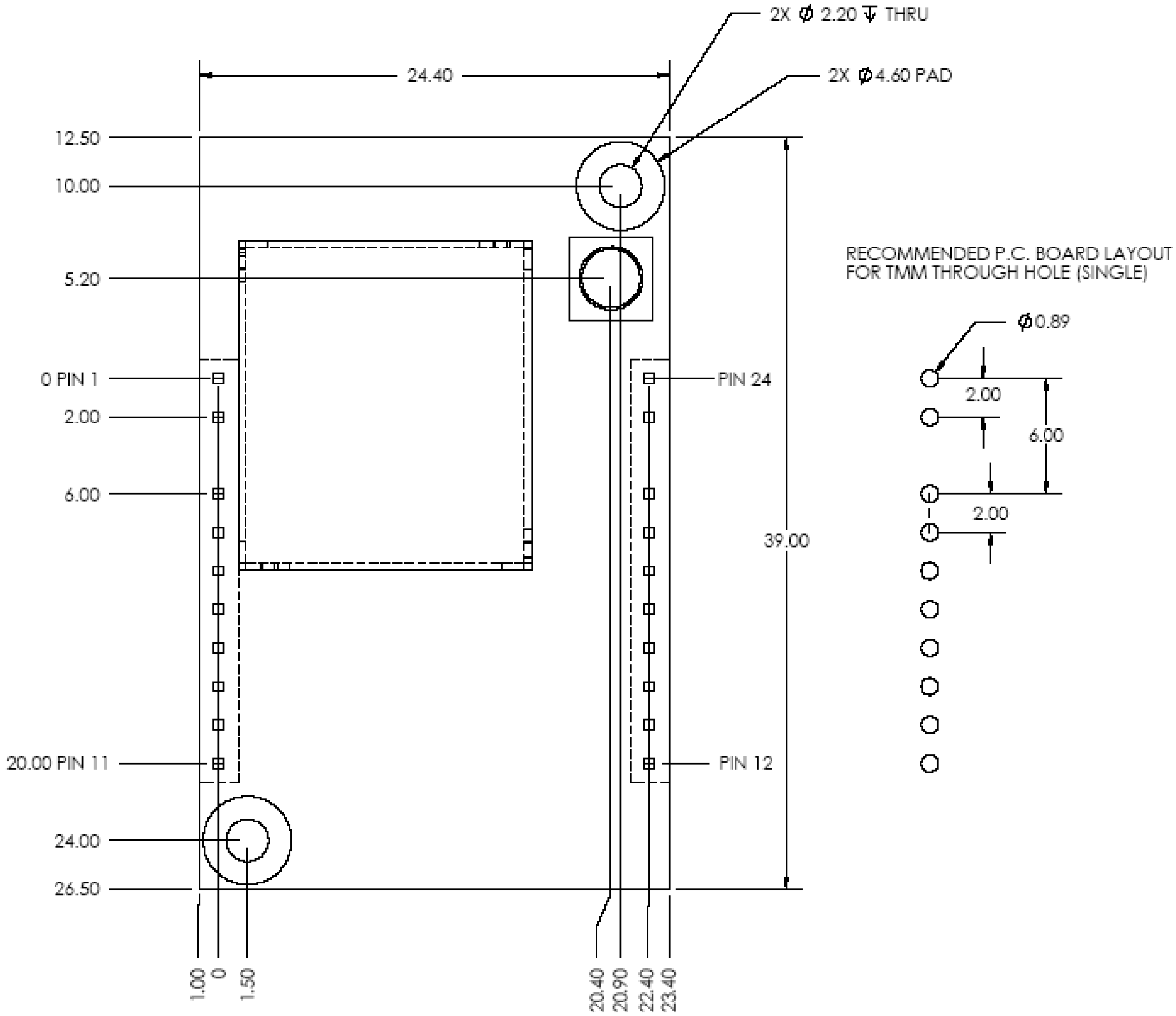



**Figure 5 M1030-2 Mote Footprint—Mechanical drawing**

## 7.2 Soldering Information

The M1030-2 can be hand soldered with a soldering iron at 230° C. The soldering iron should be in contact with the pin for 10 seconds or less.

# 8.0 Regulatory and Standards Compliance

## 8.1 FCC Compliance

### 8.1.1 FCC Testing

The M1030-2 mote complies with Part 15.247 modular (Intention Radiator) of the FCC rules and regulations. In order to fulfill FCC certification requirements, products incorporating the M1030-2 mote must comply with the following:

1. An external label must be provided on the outside of the final product enclosure specifying the FCC identifier (SJC-M1030), as described in 8.1.3 below.
2. The antenna must be electrically identical to the FCC-approved antenna specifications for the M1030-2 as described in 8.1.2 or the gain may be lower than specified in Table 3.
3. The device integrating the M1030-2 mote may not cause harmful interference, and must accept any interference received, including interference that may cause undesired operation.
4. An unintentional radiator scan must be performed on the device integrating the M1030-2 mote, per FCC Rules and Regulations, Title 47, Part 15, Subpart B. See FCC rules for specifics on requirements for declaration of conformity.

### 8.1.2 FCC-approved Antennae

The following are FCC-approved antenna specifications for the M1030-2:

**Table 3 FCC-approved Antenna Specifications for the M1030-2**

| Gain | Pattern | Type | Frequency | Connector |
|---|---|---|---|---|
| **+2 dBi** | Omni-directional | 1/4 $\lambda$ | 902-928 MHz | MMCX |
| **+2 dBi** | Omni-directional | 1/2 $\lambda$ | 902-928 MHz | MMCX |

### 8.1.3 OEM Labeling Requirements

The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. The outside of the final product enclosure must have a label with the following (or similar) text specifying the FCC identifier. The FCC ID and certification code must be in Latin letters and Arabic numbers and visible without magnification.

Contains transmitter module FCC ID: SJC- M1030

or

Contains FCC ID: SJC-M1030.

## 8.2 IC Compliance

The M1030-2 shall be certified for modular Industry Canada (IC) approval. The OEM is responsible for its product to comply with IC ICES-003 and FCC Part 15, Sub. B – Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test reports or CISPR 22 test reports for compliance with ICES-003.

## 8.3 Industrial Environment Operation

The M1030-2 is designed to meet the specifications of a harsh industrial environments which includes:

- **Shock and Vibration**—The M1030-2 complies with high vibration pipeline testing, as specified in IEC 60770-1.
- **Hazardous Locations**—The M1030-2 design is consistent with operation in UL Class 1, Division 2 Hazardous Locations.
- **Temperature Extremes**—The M1030-2 is designed for industrial storage and operational temperature range of –40°C to 85°C.

# 9.0 Ordering Information

**Product List:**

**M1030-2:** SmartMesh-XT / 900 MHz Analog/Digital/Serial Mote

**KT1029:** SmartMesh-XT/ 900 MHz Evaluation Kit

**Contact Information:**

Dust Networks
30695 Huntwood Ave.
Hayward, CA 94544

**Toll-Free Phone:** 1 (866) 289-3878

**Website:** www.dustnetworks.com

**Email:** sales@dustnetworks.com

Document Number: 020-0013 rev A M1030-2 Datasheet

Last Revised: June 21, 2006