DUST NETWORKS™

SmartMesh-XD™ M1310-1

900 MHz Wireless Serial Mote

Product Description

The SmartMesh-XD™ M1310-1 embedded wireless mote uses Time Synchronized Mesh Protocol (TSMP) to enable low-power wireless sensors and actuators with highly reliable wireless mesh networking. The M1310-1 is tailored for use in battery- and line-powered wireless devices for applications that demand proven performance, scalability, and reliability.

The M1310-1 uses a 900 MHz radio to achieve more than 200-meter communication distance outdoors, while consuming down to 9 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 M1310-1 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 M1310-1 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 M1310-1 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 M1310-1 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

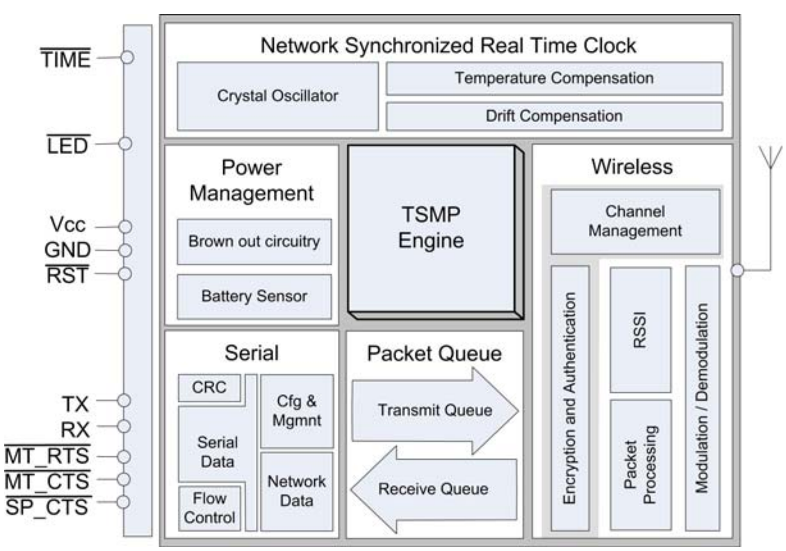
- Ultra-low power components for long battery life
- Network-wide coordination for efficient power usage
- Down to 9 µA typical power consumption

Efficient Radio

- 0.4 mW (–3.5 dBm) RF output power
- –95 dBm typical receiver sensitivity
- Outdoor range >200 m typical

Predictable Integration

- High-level Data Link Control (HDLC) serial interface with bidirectional flow control
- FCC and IC modular intentional radiation certification
- Industrial temperature range –40 °C to +85 °C
- Supports socket or solder assembly
- Rugged design for Class I Division I environments



Contents

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 M1310-1 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 up to 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)	2.75		3.3	V	Including noise and load regulation
Voltage on analog input pins	0		1.5	V	
Voltage supply noise			10	mVp-p	50 Hz–2 GHz
Peak current			75 8 6	mA mA mA	Updating flash contents Tx, 12 ms maximum Rx, searching for network, 60 minutes, maximum
Average current		9		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.

Table 3 Current Consumption

Parameter	Min	Typ	Max	Units	Comments
Transmit		2.5		mA	
Receive		1.7		mA	
Sleep		5		µA	

3.0 Electrical Specifications

.

Table 4 Device Load

Parameter	Min	Typ	Max	Units	Comments
Total capacitance		9.6	12	µF	V_{cc} to GND

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

Table 5 Digital I/O

Digital signal	Min	Typ	Max	Units	Comments
V_{IH} (logical high input)	2.0		3.6	V	
V_{IL} (logical low input)	GND – 0.3	GND	GND + 0.8	V	
V_{OH} (logical high output)	2.4			V	
V_{OL} (logical low output)			0.4	V	
Digital current*					
Output source (single pin)		3.7		mA	V_{OH} = 2.3 V, 25 °C
Output sink (single pin)		2.0		mA	V_{OL} = 0.4 V, 25 °C
Input leakage current			(TBD)	nA	

* This current level guarantees that the output voltage meets V_{OL} of 0.4 V and V_{OH} of 2.4 V.

3.1 Application Circuit

The following schematic shows how the M1310-1 mote is used in a circuit.

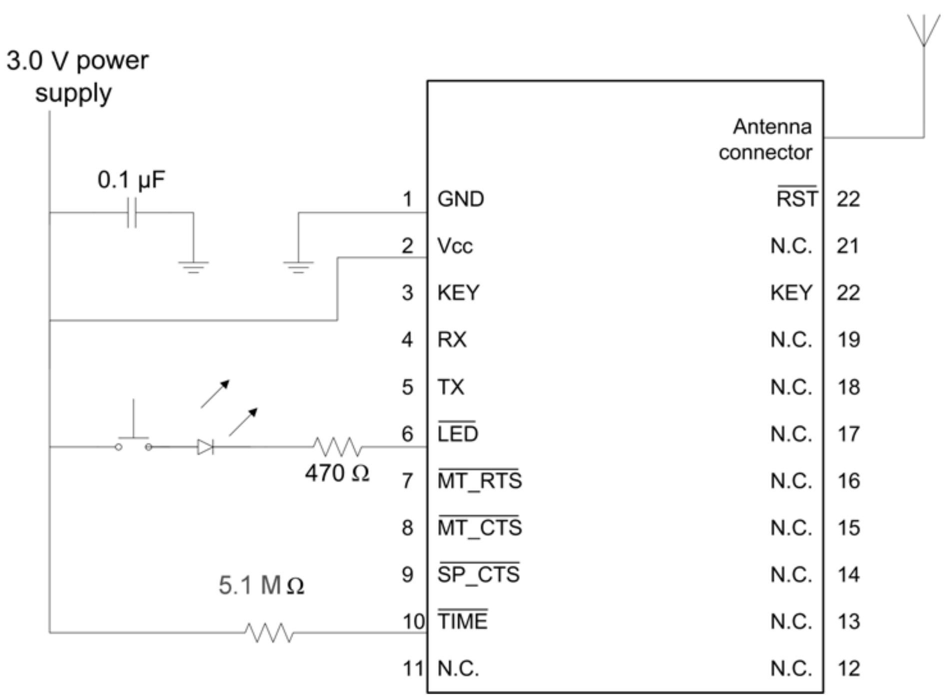


Figure 1 M1310-1 Mote in Application Circuit

4.0 Radio

4.1 Detailed Radio Specifications

Table 6 Radio Specifications

Parameter	Min	Typ	Max	Units	Comments
Operating frequency	902		928	MHz	
Number of channels		15			
Channel separation		1.5		MHz	
Channel bandwidth		250		kHz	
Modulation					FSK
Raw data rate		92		kbps	
Receiver sensitivity At 25 °C		–95		dBm	At 10^{-3} BER, Vcc = 3 V
Output power (conducted) At 25 °C		–3.5		dBm	Vcc = 3 V
Range* Indoor Outdoor		80 200		m 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 7. For a list of FCC-approved antennae see 9.1.2.

Table 7 Antenna Specifications

Parameter	Value
Frequency range	902-928 MHz
Impedance	50 Ω
Gain	+6 dBi maximum
Pattern	Omni-directional
Maximum VSWR	3:1
Connector	MMCX*

* The M1310-1 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 M1310-1 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 M1310-1.

The M1310-1 provides a bidirectional flow-controlled serial interface (serial protocol is specified in 7.3.1).

Figure 2 M1310-1 Package with Pin Labels

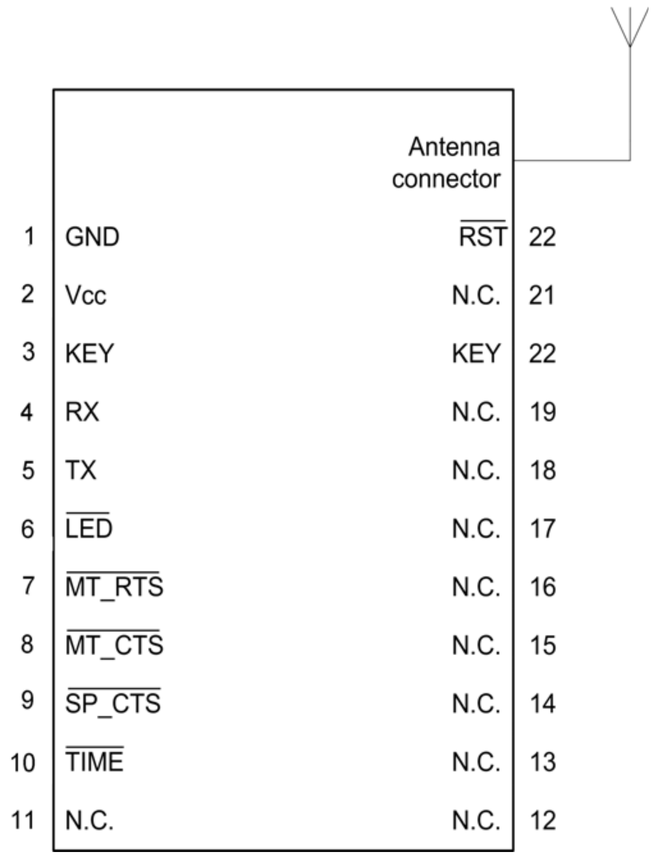


Table 8 M1310-1 Pin Functions

Pin Number	Name	Mote I/O Direction	Internal Pull Up/Down
1	GND	-	None
2	VCC	-	None
3	KEY (no pin)	-	None
4	RX	In	None
5	TX	Out	None
6	$\overline{\text{LED}}$	Out	None
7	$\overline{\text{MT\_RTS}}$	Out	None
8	$\overline{\text{MT\_CTS}}$	Out	None
9	$\overline{\text{SP\_CTS}}$	In	None
10	$\overline{\text{TIME}}$	In	None
11	*No Connection*	-	None
12	*No Connection*	-	None
13	*No Connection*	-	None
14	*No Connection*	-	None
15	*No Connection*	-	None
16	*No Connection*	-	None
17	*No Connection*	-	None
18	*No Connection*	-	None
19	*No Connection*	-	None
20	KEY (no pin)	-	None
21	*No Connection*	-	None
22	$\overline{\text{RST}}$	In	100 kΩ pull up

The $\overline{\text{RST}}$ input pin is internally pulled up, and is optional. When driven active low, the mote is hardware reset until the signal is deasserted. Refer to section 6.1 for timing requirements on the $\overline{\text{RST}}$ pin. Note that the mote may also be reset using the mote serial command (see section 7.3.3.8).

The $\overline{\text{TIME}}$ input pin is optional, and must either be driven or pulled up with a 5.1 MΩ resistor. Unless noted otherwise, all signals are active low.

6.0 Mote Boot Up

6.1 Power-on Sequence

TBD

6.2 Inrush Current

During power on, the mote can be modeled as a lumped impedance, as shown in Figure 3. With a source impedance (R_{src}) of 1 Ohm, the inrush current on the mote appears as shown in Figure 4.

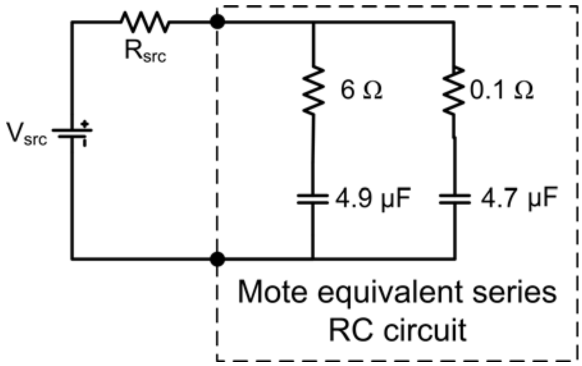


Figure 3 M1310 Equivalent Series RC Circuit

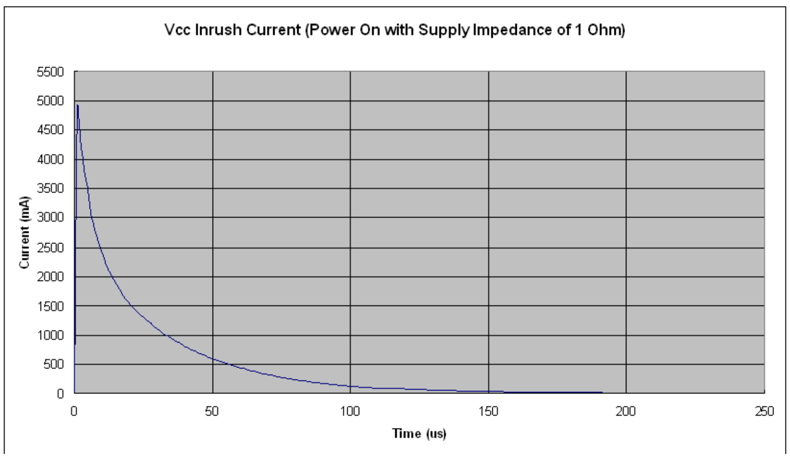


Figure 4 Vcc Inrush Current

6.3 Serial Interface Boot Up

Upon M1310-1 power up, the $\overline{\text{MT\_CTS}}$ line is high (inactive). The M1310-1 serial interface boots within boot_delay (see Table 12) of the mote powering up, at which time the M1310-1 will transmit an HDLC Mote Information packet, as described below in section 7.3.3.7. Note that full handshake (see 7.3.1.2) is in effect and is required to receive this packet.

7.0 Interfaces

7.1 Timestamps

The M1310-1 has the ability to deliver network-wide synchronized timestamps. The M1310-1 sends a time packet (as described in Table 40) through its serial interface when one of the following occurs:

- Mote receives an HDLC Get Parameter request for time/state (see Table 39)
- Mote $\overline{\text{TIME}}$ signal is activated

The $\overline{\text{TIME}}$ pin is optional and has the advantage of being more accurate. The value of the timestamp is taken within approximately a millisecond of receiving a $\overline{\text{TIME}}$ signal activation. If the HDLC request is used, because of packet processing, the value of the timestamp may be captured several milliseconds after receipt of the packet. The real time delivered to the sensor processor is relative to the real time clock on the Manager which serves as the network real time clock (NRTC). The time stamp skew across the network is guaranteed to be within ±250 ms of the NRTC.

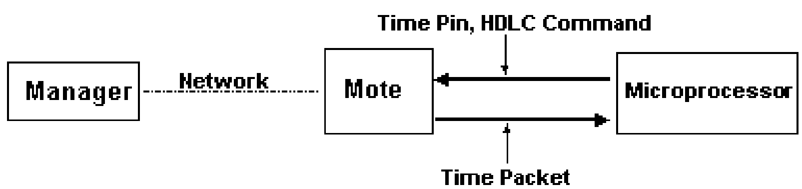


Figure 5 Real Time

When the time pin is activated for at least min_strobe_length (see Table 12), the mote responds by sending the time packet within 100 ms delay.

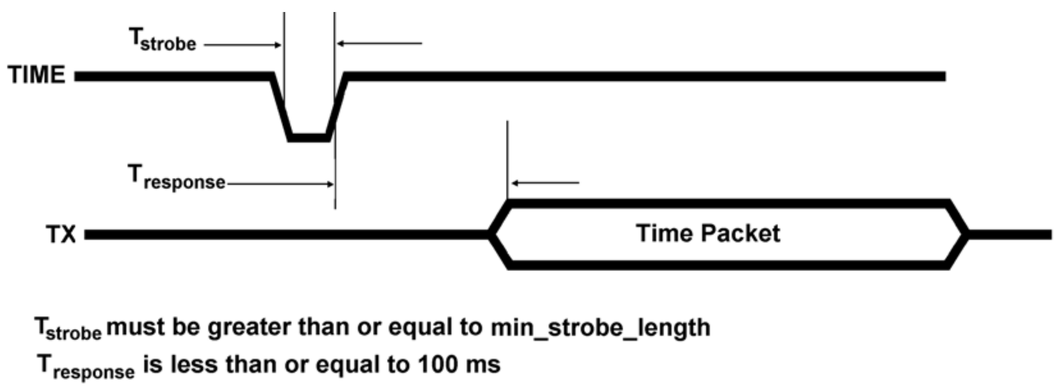


Figure 6 Operation of Time Pin

7.2 Status LED Signal

The M1310-1 provides an output that can be used to drive a status LED. This signal indicates network connectivity information which is most useful during mote installation. Alternatively, the mote's network status may be polled via serial using the Get Parameter request (see 7.3.3.6) with the mote state parameter (see 7.3.4.3). See Figure 1 for an example application circuit.

Table 9 Status LED Signal

LED Signal Behavior	Mote State
High	Off, or in sleep mode
Single blink (750 ms low, 3 s high)	On, and searching for potential network
Double blink (750 ms low, 750 ms high, 750 ms low, 3 s high)	On, and attempting to join the network
Triple blink (750 ms low, 750 ms high, 750 ms low, 750 ms high, 750 ms low, 3 s high)	On, and attempting to establish redundant links
Low	On, fully configured into network with redundant parents

7.3 Serial Interface

The M1310-1 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 handshake pins ($\overline{\text{MT\_RTS}}$, $\overline{\text{MT\_CTS}}$, $\overline{\text{SP\_CTS}}$) used for bidirectional flow control. Through this port, the M1310-1 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.

7.3.1 Serial Handshake

The serial handshake provides for flow control of packets transmitted via the M1310-1 serial interface. Packet delineation and error control are handled separately. The handshake supports the following:

- Full-duplex communication
- Bidirectional byte-level flow control

7.3.1.1 Serial Port

The five-pin serial port is comprised of the data pins (TX, RX) as well as the handshake pins ($\overline{\text{MT\_RTS}}$, $\overline{\text{MT\_CTS}}$, $\overline{\text{SP\_CTS}}$) used for bidirectional flow control. This port supports 9600 bps operation in full-duplex mode. The handshake signals are active low.

Table 10 Serial Parameters

Parameter	Value
Bit rate	9600
Stop bit	1
Data bits	8
Parity	None

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

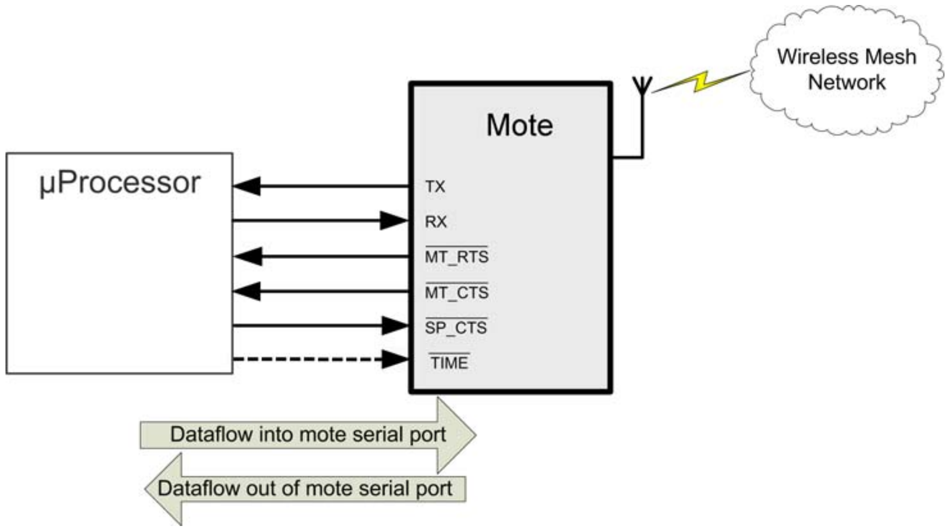


Figure 7 Diagram of Pins Used in Handshaking Protocol

Table 11 Pin Usage

Pin	Usage
RX, TX	Used for serial data flow into and out of the mote.
MT_RTS	This signal goes active low when the mote is ready to send a serial packet. The signal stays low until the SP_CTS signal from the microcontroller goes active low (indicating readiness to receive a packet) or the ack_delay timeout (see Table 12) expires.
SP_CTS	SP_CTS should transition from high to active low in response to the MT_RTS signal from the mote.This indicates that the microcontroller is ready to receive serial packets. Following this, the microcontroller should strobe SP_CTS after receiving each byte. After all packets are received, the microcontroller should de-assert the SP_CTS signal.
MT_CTS	MT_CTS indicates the state of the network connection and availability of data buffers to receive packets destined for the network. Once the mote has established wireless network connection, it will use the MT_CTS pin to signify availability to accept serial packets for wireless transmission. At certain critical times during communication, the mote may bring MT_CTS high. MT_CTS will remain high if the mote does not have enough buffer space to accept another packet. It will also remain high if the mote is not part of the network. OEM designs must check that the MT_CTS pin is low before initiating each serial packet for wireless transmission. Note that the mote may receive diagnostic serial packets at any time regardless of the CTS state. Upon receipt of the first byte of the HDLC packet, the mote strobes MT_CTS in acknowledgement of each subsequent byte. After the last byte of the packet is received, MT_CTS switches back to signaling the availability of the network connection and data buffers. The microcontroller should wait a minimum of interpacket_delay (see Table 12) before initiating another packet transmission. The mote can accept diagnostics (packets that are not sent through the network) at any time, and the status of the MT_CTS pin may be ignored when initiating these packets. (MT_CTS acknowledges each byte, as specified in 7.3.1.2.1).
TIME	The TIME pin is optional and can be used for triggering a timestamp packet. For details, refer to 7.1.

7.3.1.2 Serial Interface Timing Requirements

7.3.1.2.1 CTS Byte-level Handshake

The following diagram shows generic CTS byte-level flow control timing. The following details are applicable to both MT_CTS and SP_CTS.

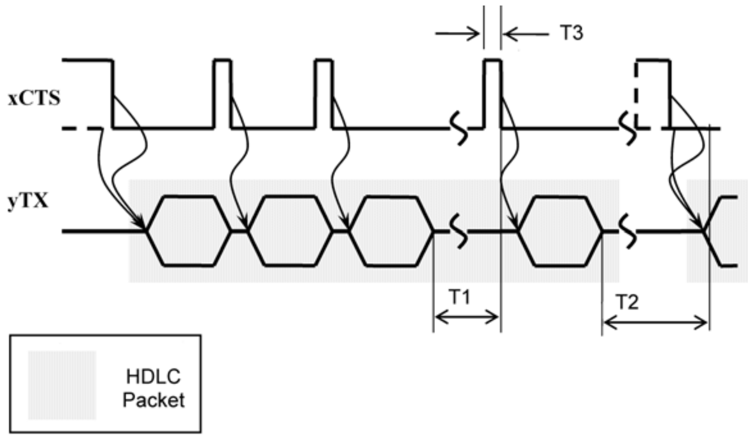


Figure 8 CTS Byte-level Flow Control Timing

Timeouts T1, T2, and T3 are defined as follows (refer to Table 12 for values):

- **T1:interbyte_timeout—**Maximum time between the transmit module sending a byte and the receiving module acknowledging the byte using CTS (requests the next byte).
- **T2: interpacket_delay—**For communications into the mote, the minimum time after the mote receives the last byte of a packet before it can start receiving the next packet. For communications out of the mote, the minimum time between the mote receiving acknowledgement of the last byte reception (or timeout) and the mote driving RTS to request to send another packet.
- **T3: min_strobe_length—**The minimum length of time that CTS must be held active to be recognized by the receiver.

In idle mode or upon expiration of the interbyte_delay timeout, the transmit side treats CTS as level triggered (MT_CTS is disregarded in case of diagnostic serial packets). After transfer of the first byte of a packet, the meaning of CTS signal is changed to a byte acknowledgement strobe, active on a falling edge. In other words, CTS becomes a request signal for the next byte of a packet. This acknowledgement strobe will occur for all packets (both diagnostic and network packets). Whenever timeouts T1 or T2 occur, the packet is discarded and both sides switch to idle mode and start hunting for the next HDLC packet, assuming CTS active low. If a packet is transferred completely, the interbyte_delay after the last byte naturally takes care of switching to idle mode.

7.3.1.2.2 Data Flow Out of the Mote Serial Port

Figure 9 illustrates the process the mote uses to transmit serial data:

1. The mote ensures the interpacket_delay time has passed since the last transmission.
2. The mote drives MT_RTS to active, waits for a falling edge on SP_CTS. Timeout is defined as ack_delay (see Table 12), and is long enough to handle the worst case response.
3. If the mote times out before the SP_CTS becomes active, the mote restores MT_RTS to inactive and drops the packet.
4. If SP_CTS is active, then the mote transmits the first byte and follows the CTS byte-level handshaking rules for subsequent bytes.
5. MT_RTS is restored to inactive after the ack_delay timeout has expired.

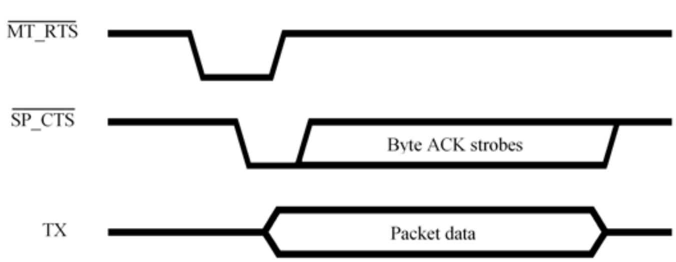


Figure 9 Packet Transmission from Mote

7.3.1.2.3 Data Flow into the Mote Serial Port

Figure 10 illustrates the process the mote uses to receive serial data:

The mote may receive serial packets for local commands (not intended for wireless transmission) at any time regardless of the MT_CTS status.

The mote signals its readiness to receive serial packets for wireless transmission (serial payload command 0x80) by driving MT_CTS active low. The mote will drive MT_CTS low within interpacket_delay time (see Table 12) after the transmission of the last packet.

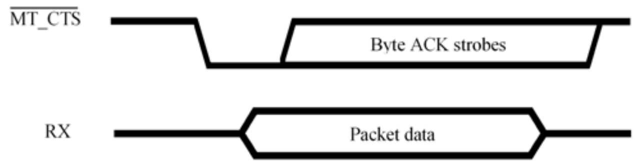


Figure 10 Packet Transmission to Mote

7.3.1.2.4 Timing Values

Table 12 Timing Values

Variable	Meaning	Min	Max	Unit
interbyte_delay	The time between consecutive data bytes cannot exceed this time.	0	7	ms
interpacket_delay	The sender of an HDLC packet must wait at least this amount of time before sending another packet.	N/A	20	ms
ack_delay	The max time delay between the MT_RTS and the receivers acknowledge, SP_CTS.	0	500	ms
time_ack_timeout	The mote responds to all TIME pin activation requests within this time.	N/A	100	ms
diag_ack_timeout	The mote responds to all requests within this time.	N/A	125	ms
min_strobe_length	The length of the strobe signal.	500		ns
boot_delay	The time between mote power up and serial interface availability.		250	ms

7.3.2 Mote Command Data Types

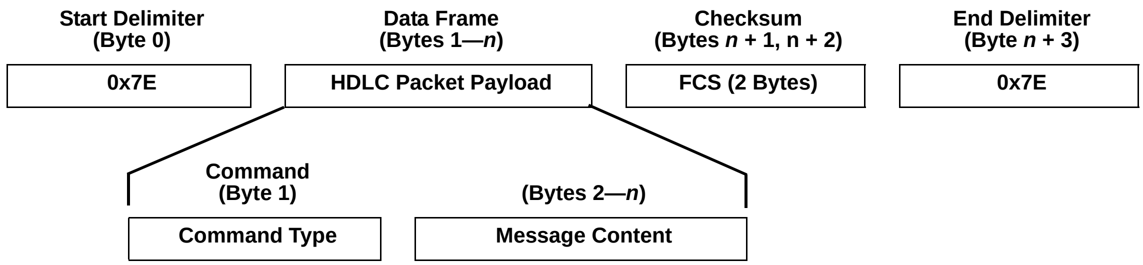
Table 13 defines the command data types used in the commands. All values are unsigned.

Table 13 Command Data Types

Data Type	Length
unsigned long	4 bytes
unsigned short	2 bytes
unsigned char	1 byte

7.3.3 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.

FCS is calculated based on 16-bit FCS computation method (RFC 1662). The mote checks the FCS and drops packets that have FCS 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 7.3.5 provides an example of HDLC packet construction and HDLC packet decoding.

Table 14 provides a summary of mote commands, which are described in detail in the following sections. For error handling, all other packet types should be ignored.

Table 14 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

7.3.3.1 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 (excluding byte-stuffing bytes). There is no response by the mote upon reception of this command.

Table 15 Command 0x80 Serial Payload to Mote

Msg Byte	Description	Data Type	Request (Sent to Mote)
1	Cmd Type	unsigned char	0x80
2		(Transparent to mote)	First byte of data
...2+n		(Transparent to mote)	Up to n–1 additional bytes of data

7.3.3.2 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 (excluding byte-stuffing bytes).

Table 16 Command 0x81 Unacknowledged Serial Payload from Mote

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x81
2		(Transparent to mote)	First byte of data
...2+n		(Transparent to mote)	Up to n–1 additional bytes of data

7.3.3.3 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 (excluding byte-stuffing bytes). The microcontroller receives exactly one copy of the message that was sent through the network.

Table 17 Command 0x82 Acknowledged Serial Payload Downstream

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x82
2		(Transparent to mote)	First byte of data
...2+n		(Transparent to mote)	Up to n–1 additional bytes of data

7.3.3.4 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 Parameter request with time as the parameter (described later) or when the $\overline{\text{TIME}}$ pin is strobed high to low for minimum of min_strobe_length, as defined in Table 12. Usage of the $\overline{\text{TIME}}$ pin is described in section 7.1.

Table 18 Command 0x84 Time/State Packet

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x84
2-5	The sequential number of the frame	unsigned longunsigned long	Cycle
6-9	The offset from start of frame in microseconds	unsigned long	Offset
10-11	Frame length in timeslots	unsigned short	Frame Length
12-15	UTC time seconds	unsigned long	Real Time part1
16-19	UTC time microseconds	unsigned long	Real Time part2
20-23	Time from the last mote reset in milliseconds	unsigned long	Mote uptime
24	Mote state (see Table 33)	unsigned char	Mote state
25	Mote diagnostics status (see Table 34)	unsigned char	Mote diagnostics status

7.3.3.5 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 12). The command structure for individual Parameter Types and can be found in section 7.3.4. The length of payload (*n*) depends on the Parameter Type and is specified in the Parameter Data Packet section of this document.

Table 19 Command 0x87 Set Parameter Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x87
2		unsigned char	Parameter Type
3	Data		First byte of data
...3+n	Data		Up to n–1 additional bytes of data

.

Table 20 Command 0x88 Set Parameter Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x88
2		unsigned char	Parameter Type
3	Error code	unsigned char	Error code (see Table 28)
4	Data Length	unsigned char	0x00

7.3.3.6 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 (see Table 12). The command structure for individual parameter types can be found in section 7.3.4. 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 28.

Table 21 Command 0x89 Get Parameter Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x89
2		unsigned char	Parameter Type
3	Data		First byte of data
...3+n	Data		Up to n–1 additional bytes of data

Table 22 Command 0x8A Get Parameter Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8A
2		unsigned char	Parameter Type
3	Error code	unsigned char	Error code (see Table 28)
4	Data length	unsigned char	n
5	Data		First byte of data
...5+n	Data		Up to n–1 additional bytes of data

7.3.3.7 Command 0x8C Mote Information

The mote sends this packet on bootup, supplying information about mote properties. For details on bootup, see 6.3.

Table 23 Command 0x8C – M1310-1 Information

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8C
2-4	HW model	Array of 3 unsigned char	HW model
5-6	HW revision	Array of 2 unsigned char	HW revision
7-10	SW revision	Array of 4 unsigned char	SW revision
11-18	MAC address	Array of 8 unsigned char	MAC addr
19	Networking type	unsigned char	TBD
20-21	Network ID	unsigned short	Network ID
22-29	Datasheet ID	Array of 8 unsigned char	Datasheet ID
30-31	Mote ID	unsigned short	Mote ID
32			Reserved
33	Mote diagnostics status (see Table 34)	unsigned char	Mote diagnostics status

7.3.3.8 Command 0x8D Reset Mote

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

Table 24 Command 0x8D Reset Mote

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8D

7.3.4 Mote Get / Set Command Parameters

This section specifies the parameters that may be used with the Set and Get Commands. Table 25 provides an overview of the these parameters.

Table 25 Set and Get Command Parameters

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

All requests have the following structure:

Table 26 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 27 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.

7.3.4.1 Error Codes

Table 28 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

7.3.4.2 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 29 Parameter Type 0x01 Network ID Set Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x87
2	Parameter Type	unsigned char	0x01
3-4	Network ID	unsigned short	Network ID

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

Table 30 Parameter Type 0x01 Network ID Set Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x88
2	Parameter Type	unsigned char	0x01
3	Error code	unsigned char	Error code (see Table 28)
4	Data Length	unsigned char	0x00

7.3.4.3 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 33).

Table 31 Parameter Type 0x02 Mote State Get Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x89
2	Parameter Type	unsigned char	0x02

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

Table 32 Parameter Type 0x02 Mote State Get Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8A
2	Parameter Type	unsigned char	0x02
3	Error code	unsigned char	Error code (see Table 28)
4	Data Length	unsigned char	0x02
5		unsigned char	Mote State
6	Mote diagnostics status (see Table 34)	unsigned char	Mote diagnostics status

Table 33 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 a join request, waiting to be activated.
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. The mote is ready to transmit data to the network.
10	ONLINE2	The mote has joined a network, has been fully configured, and has multiple parents. The mote is ready to transmit data to the network.

Table 34 Diagnostics Status

Bit	Name	Details
7	---	Reserved.
6	---	Reserved.
5	---	Reserved.
4	---	Reserved.
3	---	Reserved.
2	---	Reserved.
1	CCF	Configuration change flag (see section 7.3.4.3.1).
0	NV_ERR	Non-volatile memory error.

7.3.4.3.1 Configuration Change Flag (CCF)

The Configuration Change Flag (CCF) bit is set high when the network ID is changed. Note that when the network ID is changed over the air (using the XML-API), the entire network synchronously changes over to the new network ID. There is no delay between when the XML-API command is received and when motes change over to the new network ID. The CCF bit is set high when the new network ID becomes active. The CCF bit is cleared when the mote receives a Mote Information Get request (Command 0x07) or the mote is reset.

7.3.4.4 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.

Table 35 Parameter Type 0x03 Frame Length Get Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x89
2	Parameter Type	unsigned char	0x03
3		unsigned char	Frame ID

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

Table 36 Parameter Type 0x03 Frame Length Get Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8A
2	Parameter Type	unsigned char	0x03
3	Error code	unsigned char	Error code (see Table 28)
4	Data Length	unsigned char	0x05
5		unsigned char	Frame ID
6-9	Frame Length (ms)	unsigned long	Frame Length

7.3.4.5 Parameter Type 0x04 Join Key

The join key is needed to allow a mote on the network. The join key is specific for the network and used for data encryption. This parameter is only valid for a 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 37 Parameter Type 0x04 Join Key Set Request

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

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

Table 38 Parameter Type 0x04 Join Key Set Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x88
2	Parameter Type	unsigned char	0x04
3	Error code	unsigned char	Error code (see Table 28)
4	Data Length	unsigned char	0x00

7.3.4.6 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 39 Parameter Type 0x05 Time/State Get Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x89
2	Parameter Type	unsigned char	0x05

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

Table 40 Parameter Type 0x05 Time/State Get Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x8A
2	Parameter Type	unsigned char	0x05
3	Error code	unsigned char	Error code (see Table 28)

Table 40 Parameter Type 0x05 Time/State Get Response

Msg Byte	Description	Data Type	Value
4	Data Length	unsigned char	0x18
5-8		unsigned long	Cycle
9-12		unsigned long	Offset (μsec)
13-14	Frame Length (slots)	unsigned short	Frame Length
15-18	UTC Time sec	unsigned long	UTC Time sec
19-22	UTC Time μsec	unsigned long	UTC Time μsec
23-26	Mote uptime msec	unsigned long	Mote uptime msec
27	Mote state	unsigned char	Mote state
28	Mote diagnostics status (see Table 34)	unsigned char	Mote diagnostics status

7.3.4.7 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 41 Parameter Type 0x07 Mote Information Get Request

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	0x89
2	Parameter Type	unsigned char	0x07

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

Table 42 Parameter Type 0x07 Mote Information Get Response

Msg Byte	Description	Data Type	Value
1	Cmd Type	unsigned char	140 (0x8A)
2	Parameter Type	unsigned char	0x07
3	Error Code	unsigned char	Error Code
4	Data length	unsigned char	0x20
5-7	HW model	Array of 3 unsigned char	HW model
8-9	HW revision	Array of 2 unsigned char	HW revision
10-13	SW revision	Array of 4 unsigned char	SW revision
14-21	MAC address	Array of 8 unsigned char	MAC addr
22	Networking type	unsigned char	TBD
23-24	Network ID	unsigned short	Network ID
25-32	Datasheet ID	Array of 8 unsigned char	Datasheet ID
33-34	Mote ID	unsigned short	Mote ID
35			Reserved
36	Mote diagnostics status (see Table 34)	unsigned char	Mote diagnostics status

7.3.5 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 => 7D

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

Step 2 Calculate FCS:

- Calculate the FCS using FCS-16 algorithm (RFC 1662) on the hexadecimal sequence '87 01 00 7D'. The FCS (including 1's complement) is 74 2F.
- Append FCS to payload, FCS is sent least significant byte first (RFC 1662):

HDLC Packet Payload	FCS
87 01 00 7D	2F 74

Step 3 Perform byte stuffing.

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

7D => 7D 5D
7E => 7D 5E
Note that the additional control bytes do not count against the 8D byte payload limit.

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

Step 4 Add start and stop delimiters:

Enclose the above in start/stop flags (RFC 1662).

Start Delimiter	HDLC Packet Payload (stuffed)	FCS (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)	FCS (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	7D C4	7E

Step 1 (HDLC layer) strip off delimiters:

HDLC Packet Payload (stuffed)	FCS (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	7D C4

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	FCS
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	7D C4

Step 3 Confirm FCS.

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 FCS matches the FCS sent with the packet. Because the packet encodes FCS least significant byte first, in this example the calculated FCS should match “C4 7D”.

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 7.3.3.6., 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 (shown for 900 MHz mote):

HW model =	00091	(00 00 5B)
HW Rev =	001	(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)

8.0 Packaging Description

8.1 Mechanical Drawings

TBD

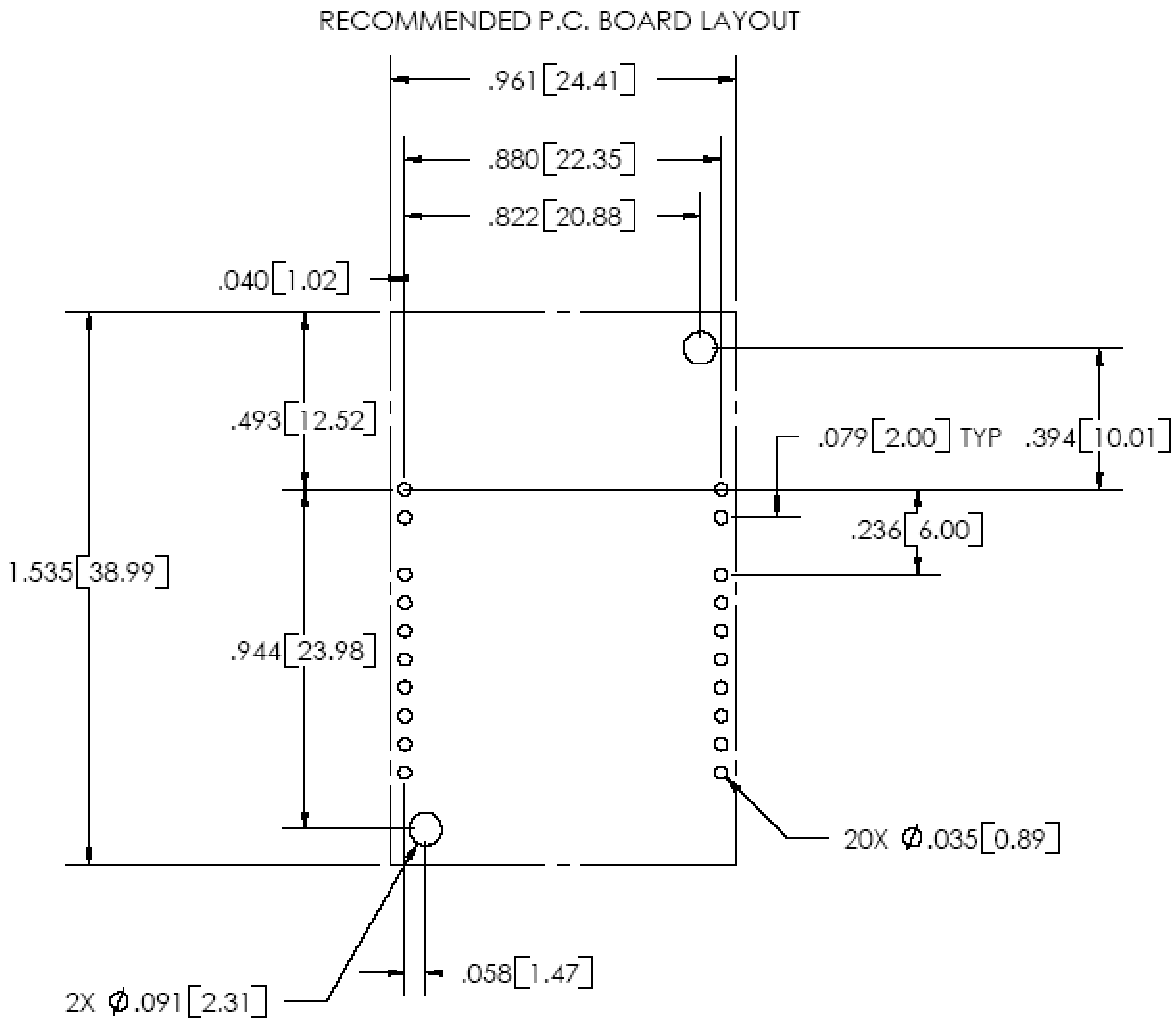


Figure 11 M1310-1 Mote Footprint—Mechanical drawing

8.2 Soldering Information

The M1310-1 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.

9.0 Regulatory and Standards Compliance

9.1 FCC Compliance

9.1.1 FCC Testing

The M1310-1 mote will comply with Part 15.247 modular (Intention Radiator) of the FCC rules and regulations. In order to fulfill FCC certification requirements, products incorporating the M1310-1 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 as described in 9.1.3 below.
2. The antenna must be electrically identical to the FCC-approved antenna specifications for the M1310-1 as described in 9.1.2 or the gain may be lower than specified in Table 43.
3. The device integrating the M1310-1 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 M1310-1 mote, per FCC Rules and Regulations, Title 47, Part 15, Subpart B. See FCC rules for specifics on requirements for declaration of conformity.

9.1.2 FCC-approved Antennae

The following are FCC-approved antenna specifications for the M1310-1.

Table 43 FCC-approved Antenna Specifications for the M1310-1

Gain	Pattern	Polarization	Frequency	Connector
+6 dBi maximum	Omni-directional	Vertical	902-928 MHz	MMCX

9.1.3 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-XXXX

or

Contains FCC ID: SJC-XXXX

9.2 IC Compliance

9.2.1 IC Testing

The M1310-1 should be certified for modular Industry Canada (IC) RSS-210 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 equivalent to FCC Part 15 Sub. B and Industry Canada accepts FCC test reports or CISPR 22 test reports for compliance with ICES-003.

9.2.2 IC-approved Antennae

The following are IC-approved antenna specifications for the M1310-1.

Table 44 IC-approved Antenna Specifications for the M1310-1

Gain	Pattern	Polarization	Frequency	Connector
+6 dBi maximum	Omni-directional	Vertical	902-928 MHz	MMCX

9.3 Industrial Environment Operation

The M1310-1 is designed to meet the specifications of a harsh industrial environments which includes:

- **Shock and Vibration**—The M1310-1 complies with high vibration pipeline testing, as specified in IEC 60770-1.
- **Hazardous Locations**—The M1310-1 design is consistent with operation in UL Class 1 Division 1 and UL Class 1 Division 2 Hazardous Locations.
- **Temperature Extremes**—The M1310-1 is designed for industrial storage and operational temperature range of –40 °C to +85 °C.

10.0 Ordering Information

Product List:

M1310-1: SmartMesh-XD / 900 MHz Serial Mote

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: 020-0018 rev 1 M1310-1 Mote Datasheet
Last Revised: November 3, 2006

Document Status	Product Status	Definition
Advanced Information	Planned or under development	This datasheet contains the design specifications for product development. Dust Networks reserves the right to change specifications in any manner without notice.
Preliminary	Engineering samples and pre-production prototypes	This datasheet contains preliminary data; supplementary data will be published at a later time. Dust Networks reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product is not fully qualified at this point.
No Identification Noted	Full production	This datasheet contains the final specifications. Dust Networks reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not in production	This datasheet contains specifications for a product that has been discontinued by Dust Networks. The datasheet is printed for reference information only.