# SmartMesh IP Mote CLI Guide

Advance Information

This document contains advance information of a product in development. All specifications are subject to change without notice. Consult LTC factory before using.

# Table of Contents

**Trademarks**

Eterna, Mote-on-Chip, and SmartMesh IP, are trademarks of Dust Networks, Inc. The Dust Networks logo, Dust, Dust Networks, and SmartMesh are registered trademarks of Dust Networks, Inc. LT, LTC, LTM and  are registered trademarks of Linear Technology Corp. All third-party brand and product names are the trademarks of their respective owners and are used solely for informational purposes.

**Copyright**

This documentation is protected by United States and international copyright and other intellectual and industrial property laws. It is solely owned by Dust Networks, Inc. and its licensors and is distributed under a restrictive license. This product, or any portion thereof, may not be used, copied, modified, reverse assembled, reverse compiled, reverse engineered, distributed, or redistributed in any form by any means without the prior written authorization of Dust Networks, Inc.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g) (2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015 (b)(6/95) and DFAR 227.7202-3(a), and any and all similar and successor legislation and regulation.

**Disclaimer**

This documentation is provided “as is” without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

This documentation might include technical inaccuracies or other errors. Corrections and improvements might be incorporated in new versions of the documentation.

Dust Networks does not assume any liability arising out of the application or use of any products or services and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

Dust Networks products are not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Dust Networks customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify and hold Dust Networks and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Dust Networks was negligent regarding the design or manufacture of its products.

Dust Networks reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products or services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to Dust Network's terms and conditions of sale supplied at the time of order acknowledgment or sale.

Dust Networks does not warrant or represent that any license, either express or implied, is granted under any Dust Networks patent right, copyright, mask work right, or other Dust Networks intellectual property right relating to any combination, machine, or process in which Dust Networks products or services are used. Information published by Dust Networks regarding third-party products or services does not constitute a license from Dust Networks to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Dust Networks under the patents or other intellectual property of Dust Networks.

Dust Networks, Inc is a wholly owned subsidiary of Linear Technology Corporation.

© Dust Networks, Inc. 2012 All Rights Reserved.

# 1 About This Guide

This guide describes the commands used to communicate with the SmartMesh IP mote through a command line interface (CLI).

## 1.1 Related Documents

The following documents are available for the SmartMesh-enabled network:

- SmartMesh IP Quick Start Guide
- SmartMesh IP Network User Guide
- SmartMesh IP Manager User Guide
- SmartMesh IP Manager CLI Guide
- SmartMesh IP Manager API Guide
- SmartMesh IP Mote User Guide
- SmartMesh IP Mote API Guide
- SmartMesh IP Mote CLI Guide

## 1.2 Conventions Used

The following conventions are used in this document:

`Computer type` indicates information that you enter, such as specifying a URL.

**Bold type** indicates buttons, fields, and menu commands.

*Italic type* is used to introduce a new term

Tips provide useful information about the product.

Informational text provides additional information for background and context

Notes provide more detailed information about concepts.

Warning! Warnings advise you about actions that may cause loss of data, physical harm to the hardware or your person.

```
code blocks display examples of code or API
```

The CLI commands are described using the following notations and terminology:

| | |
|---|---|
| \| | Indicates alternatives for a field. For example,<br><*moteId*> \| #<*MAC*> indicates that you can specify a mote by its mote ID or MAC address. |
| < > | Indicates a required field. |
| { } | Indicates a group of fields. |
| [ ] | Indicates an optional field. |
| MAC address | When specifying a MAC address, do not use spaces. You may omit leading zeros and hyphens. In cases where the command syntax allows either the MAC address or mote ID to be specified, the MAC address must be preceded by the # symbol.<br><br>The following examples are all valid:<br><br>22CA<br><br>00000000000022CA<br><br>00-00-00-00-00-00-22-CA |

# 1.3 Revision History

| Revision | Date | Description |
|---|---|---|
| 1 | 07/17/2012 | Initial release |

# 2 Introduction

This guide describes the commands that you can send to a SmartMesh IP mote on its command line interface (CLI).

## 2.1 CLI Access

There are two dedicated serial ports on the SmartMesh IP mote: one is for API communication with an external application, and the other is dedicated to this command line interface.

You can access the CLI interface from any serial terminal program (such as HyperTerminal):

- If connecting to an evaluation board integrated with an FTDI serial-to-usb interface, the CLI will be found on the **3rd COM** port mapped onto your system.

The default serial port settings are as follows:

- 
  - Bits per second: 9600
  - Data bits: 8
  - Parity: None
  - Stop bits: 1
  - Flow control: None

# 3 Commands

## 3.1 get

**Description**
Get application parameters.

**Syntax**

```
get <parameter>
```

**Parameters**

| Parameter | Description |
|---|---|
| mode | Returns the current mode (master / slave) |

**Example**

```
> get mode
master
```

## 3.2 help

**Description**
Show help. Entering this command without parameters displays the list of all available commands. Help on a specific command may be obtained by entering that command as an argument.

**Syntax**

```
help [command]
```

**Parameters**

| Parameter | Description |
|---|---|
| command | Any of the CLI commands |

**Example**

```
help
```

# 3.3 info

**Description**

Displays information about the application layer.

**Syntax**

```
info
```

**Parameters**

| Parameter | Description |
|---|---|

**Example**

```
IP Mote: 1.1.0.36
Join state: Searching
Bandwidth Allocated: 0
Serial mode: Mode 4
Serial Baud Rate: 115200
```

# 3.4 loc

**Description**

Send a local command to the net layer

**Syntax**

```
loc <payload>
```

**Parameters**

| Parameter | Description |
|---|---|
| payload | Binary string up to 90 bytes in length |

**Example**

```
> loc 0102030405
```

# 3.5 mfs

**Description**

File system commands. These are intended for debugging.

**Syntax**

```
mfs <cmd> {-f|-p} [<param>...]
```

**Parameters**

| Parameter | Description |
|---|---|
| cmd | One of:<br><br>show - show a list of files (-f) or partitions (-p)<br><br>fcs - calculate CRC for a filename (-f <filename>) or partition (-p <parId> <offset> <length>)<br><br>dump - dump part of a file (-f <fileName> <offset> <length>) or partition (-p <parId> <offset> <length>)<br><br>del - delete file (-f <filename>) |

**Example**

```
> mfs show -p
ID   Size    Address  Page
  1  32768 0x000b7800 2048 exec
  2 258048 0x00041000 2048 exec
  4 227328 0x00080000 2048
  6   2048 0x000bf800 2048
```

# 3.6 mget/mset

**Description**

Used to get parameters that are available to user for mote configuration.

**Syntax**

```
mget <parameter>
```

**Parameters**

| Parameter | Description |
|---|---|
| netid | Network ID |
| rtmode | 0: routing enabled (default) or 1: routing disabled (can be used to force a mote to be a leaf mote) |
| joind | Duty cycle used during join process ( 0 - 255 ) 255 = 100% |
| txpower | Transmit power. 8=PA on (default), 0=PA off |
| autojoin | The netlayer will automatically try to join or not. 1=on - only valid in slave mode (See set), 0=off (default) |
| macaddr | MAC address (EUI-64), e.g.: 01-23-45-67-89-AB-CD-EF |
| otaplout | Restrict over the air programming. 1=no OTAP allowed |
| advkey | Advertisement key |
| maxStCur | Maximum current available (will be used by the manager to know how many links it can assign to this mote) |
| joincntr | Join counter used in the mote join request |

**Example**

```
> mget netid
```

# 3.7 mgeti/mseti

**Description**

Get internal configuration parameters. These are intended for internal mote development, evaluation, and advanced use under FAE supervision

**Syntax**

```
mgeti <param>
```

**Parameters**

| Parameter | Description |
|---|---|
| pftimer | Path fail timer (in seconds) |
| traceflgs | Traces enabled (see mtrace (IPMT_CLI)) |

#### Example

```
> mgeti pftimer
pftimer=60
```

## 3.8 minfo

#### Description

This command will return information about the mote, namely the code version, current join state, MAC address, Mote ID, Net ID, bootloader version, loader version, UTC time, and reset status.

#### Syntax

```
minfo
```

#### Parameters

| Parameter | Description |
|---|---|

#### Example

```
> minfo
Net stack  v1.1.0.0
state:     Oper
mac:       00:17:0d:00:00:38:09:8f
moteid:    7
netid:     63
blSwVer:   9
ldrSwVer:  1.0.3.11
UTC time:  1026005872:214750
reset st:  100
```

# 3.9 mlog

**Description**

This command retrieves the internal mote log which may contain debug information based on the last reset.

**Syntax**

```
mlog
```

**Parameters**

| Parameter | Description |
|---|---|

**Example**

```
> mlog
Low-level log: '<empty>'
```

# 3.10 mset

**Description**

Used to set parameters that are available to user for mote configuration.

**Syntax**

```
mset <param> <value>
```

**Parameters**

| Parameter | Description |
|---|---|
| netid | Network ID |
| jkey | Join key |
| rtmode | 0: routing enabled (default) or 1: routing disabled (can be used to force a mote to be a leaf mote) |
| joindc | Duty cycle used during join process ( 0 - 255 ) 255 = 100% |
| txpwr | Transmit power. 8=PA on (default), 0=PA off |
| autojoin | The netlayer will automatically try to join or not. 1=on - only valid in slave mode (See set), 0=off (default) |
| macaddr | MAC address (EUI-64), e.g.: 01-23-45-67-89-AB-CD-EF |
| otaplout | Restrict over the air programming. 1=no OTAP allowed |
| advkey | Advertisement key |
| maxStCur | Maximum current available (will be used by the manager to know how many links it can assign to this mote) |
| joincntr | Join counter used in the mote join request |

**Example**

```
mset netid 1234
```

# 3.11 mseti

**Description**

Set internal configuration parameters. These are intended for internal mote development, evaluation, and advanced use under FAE supervision

**Syntax**

```
mseti <param> <value>
```

**Parameters**

| Parameter | Description |
|---|---|
| pftimer | Path fail timer (in seconds) |

**Example**

```
mseti pftimer 60
```

# 3.12 mshow

**Description**

Show information about mote resources. Intended for debugging.

**Syntax**

```
mshow <object>
```

**Parameters**

| Parameter | Description |
|---|---|
| links | display assigned links |
| nbrs | display existing neighbors |
| pkstat | display statistics about packets |

| | |
|---|---|
| stacks | display information about task stacks |
| tasktime | display task time |

**Example**

```
> mshow links
4:84:0#65535 d:rf
5:68:0#65535 n:rlf
5:69:0#65535 n:rlf
5:70:0#65535 n:rlf
```

# 3.13 mtrace

**Description**

Turn MAC layer traces on or off

**Syntax**

```
mtrace save | {<parameter> on | off}
```

**Parameters**

| Parameter | Description |
|---|---|
| save | Save current trace flags to flash |
| mac | MAC layer TXs and RXs |
| mac_tof | Time of flight (mtrace mac must be on to see the mac_tof) |
| io | Description of the commands in the packet |
| otap | Progression/status of the over the air programming |
| all | all trace elements |

### Example

```
> mtrace mac on
7497319 : MAC R: a=57423 t=7 ch=13 s=1 rc=0 rs=-23 ad=14 q=0,0
7497457 : MAC T: a=57442 t=7 ch=1 d=1 rc=0 ad=0 po=180 pe=460 q=0,0
7498385 : MAC T: a=57570 t=2 ch=0 d=2 rc=0 ad=-20 po=182 pe=460 q=0,0
7500575 : MAC T: a=57872 t=7 ch=3 d=1 rc=0 ad=0 po=180 pe=460 q=0,0
>
> mtrace mac off
```

## 3.14 mxtal

### Description

This command is used to determine the optimal trim value to center the 20MHz crystal oscillator frequency given a particular PCB layout and crystal combination. It is used to measure the 20 MHz crystal, after which the user must enter trim values into the device's fuse table for access by software.

### Syntax

```
mxtal [trim|meas]
```

### Parameters

| Parameter | Description |
|---|---|
| trim | Trims the adjustable load capacitance for the 20MHz crystal to match the frequency reference on the programming board. Outputs the post-trim ppm error and the optimal value of the load-capacitance setting. The trimmed value of the load capacitance is not stored on the mote; the function output should be used to determine the the proper value of the load-capacitance setting for the BSP fuse table parameter. This function requires the mote be connected to the programming board. It could take up to 30 sec for command to execute. |
| meas | Outputs the ppm error of the 20MHz reference with value loaded from the fuse table . This function requires the mote be connected to the programming board. It could take up to 30 sec for command to execute. |

### Example

```
mxtal trim

mxtal meas
```

# 3.15 radiotest

**Description**

Invokes radiotest mode, which is used for certification and development.

**Syntax**

```
radiotest {on | off | rx <ch> <time> | tx {cw | cm | reg | hop} <ch | mask> <power> <numPackets>
<len>  | stat}
```

**Parameters**

| Parameter | Description |
|---|---|
| on/off | Turning radiotest on and resetting the mote overrides master mode and prevents joining. Turning if off and resetting the mote resumes normal behavior |
| rx | Receives packets on channel <ch> for time (seconds) <time> |
| tx | Transmits with the following available options:<br>cm - continuous modulation;<br>cw - continuous wave;<br>reg - packets, on single channel <ch> at power level <power>, number of packets <numPackets>, each packet <len> byte long.<br>hop - packets, using channel hopping on <mask> channels, number of packets <numPackets>, each packet <len> byte long. |
| stat | Show the statistics for an rx test |

**Example**

```
> radiotest stat
Radio Test Statistics
  OkCnt   : 0
  FailCnt : 0
```

## 3.15.1 radiotest tx examples

Continuous unmodulated transmission on channel 13 at power 8 dBm.

```
radiotest tx cw 13 8
```

Continuous modulated transmission on channel 1 at power 0 dBm.

```
radiotest tx cm 1 0
```

Regular packet transmission on channel 5, at 8 dBm, 100 80-byte packets.

```
radiotest tx reg 5 8 100 80
```

Hopping transmission with mask for channels 4 & 7, at 8 dBm, 1000 90-byte packets.

```
radiotest tx hop 0090 8 1000 90
```

# 3.16 reset

**Description**
Reset the mote.

**Syntax**

```
reset
```

**Parameters**

| Parameter | Description |
|---|---|

### Example

```
> reset
Riker app, ver 1.1.0.2
```

## 3.17 restore

### Description

This command will clear all application settings and parameters. This does not affect the net layer, application only.

### Syntax

```
restore
```

### Parameters

| Parameter | Description |
|---|---|

### Example

```
restore
```

## 3.18 set

### Description

Set application parameters. Only one parameter from the list below may be set at a time.

### Syntax

```
set <parameter> <value>
```

**Parameters**

| Parameter | Description |
|---|---|
| mode | Master: the application will terminate local commands. Slave: the local commands will be forwarded to the serial port. |

**Example**

```
> set mode slave
```

# 3.19 trace

**Description**

Turn application layer traces on or off

**Syntax**

```
trace [<module>|all] [on|off]
```

**Parameters**

| Parameter | Description |
|---|---|
| module | One of:<br>loc - local (net layer) commands<br>oap - application commands<br>ser - serial commands<br>all - all commands |

**Example**

```
> trace loc on
```