VManager AP Bridge User's Guide
# Table of Contents

1 About This Guide .................................................................................................................. 4
  1.1 Related Documents ........................................................................................................... 4
    1.1.1 Getting Started with a Starter Kit .............................................................................. 4
    1.1.2 User’s Guide ............................................................................................................... 4
    1.1.3 Interfaces for Interaction with a Device ...................................................................... 4
    1.1.4 Access Point Motes .................................................................................................... 5
    1.1.5 Software Development Tools .................................................................................... 5
    1.1.6 Application Notes ...................................................................................................... 5
    1.1.7 Documents Useful When Starting a New Design ....................................................... 5
    1.1.8 Software .................................................................................................................... 6
    1.1.9 Other Useful Documents ............................................................................................ 6
  1.2 Conventions Used ............................................................................................................ 7
  1.3 Revision History ............................................................................................................... 8
2 Introduction ......................................................................................................................... 9
3 Installation ............................................................................................................................ 10
  3.1 Raspberry Pi Software Installation .................................................................................. 10
  3.2 IP Address Configuration ............................................................................................... 11
  3.3 Adding and Configuring AP Motes ................................................................................ 11
  3.4 Adding or Removing an AP Mote .................................................................................... 12
4 Configuring the AP Bridge .................................................................................................... 13
  4.1 VManager Association ..................................................................................................... 13
  4.2 AP Mote Configuration .................................................................................................... 13
5 CLI Commands ..................................................................................................................... 15
  5.1 clear .................................................................................................................................. 16
  5.2 exit/logout/quit ................................................................................................................ 17
  5.3 get ..................................................................................................................................... 18
  5.4 help .................................................................................................................................. 20
  5.5 info .................................................................................................................................. 21
    5.5.1 VManager info ........................................................................................................ 21
    5.5.2 AP Mote info ........................................................................................................... 22
    5.5.3 AP Bridge Info ........................................................................................................ 22
    5.5.4 GPS Info ................................................................................................................ 22
  5.6 set ..................................................................................................................................... 23
  5.7 version ............................................................................................................................. 25
  5.8 CLI Debug Commands .................................................................................................... 26
    5.8.1 logger ......................................................................................................................... 26
    5.8.2 reset .......................................................................................................................... 27
    5.8.3 stats .......................................................................................................................... 28
    5.8.4 stop ........................................................................................................................... 30
6  AP Mote Commands  
6.1  info  
6.2  get  
6.3  set  
6.4  minfo  
6.5  mget  
6.6  mset  
6.7  reset  
6.8  Diagnostic and debug commands  
6.8.1  mfs  
6.8.2  radiotest  
6.8.3  trace  
6.8.4  mtrace  
6.8.5  stats  
6.8.6  mlog  
6.8.7  mxtal
1 About This Guide

1.1 Related Documents

The following documents are available for the SmartMesh IP network:

1.1.1 Getting Started with a Starter Kit

- **SmartMesh VMManager Easy Start Guide** - walks you through basic VMManager installation and a few tests to make sure your network is working.
- **SmartMesh IP Embedded Manager Easy Start Guide** - walks you through basic embedded manager installation and a few tests to make sure your network is working.
- **SmartMesh IP Embedded Manager Tools Guide** - the installation section contains instructions for installing the serial drivers and example programs used in the Easy Start Guide and other tutorials.

1.1.2 User's Guide

- **SmartMesh IP User's Guide** - describes network concepts, and discusses how to drive mote and manager APIs to perform specific tasks, e.g. to send data or collect statistics. This document provides context for the API guides. It also contains a glossary of SmartMesh terms.

1.1.3 Interfaces for Interaction with a Device

There are two interfaces for interaction with a Manager - an Application Programming Interface (API) for programmatic interaction, and a Command Line Interface (CLI) for human interaction.

- **SmartMesh IP Embedded Manager CLI Guide** - used for human interaction with an embedded manager (e.g. during development of a client, or for troubleshooting). This document covers connecting to the CLI and its command set.
- **SmartMesh IP Embedded Manager API Guide** - used for programmatic interaction with an embedded manager. This document covers connecting to the API and its command set.
- **SmartMesh IP VMManager CLI Guide** - used for human interaction with a VMManager (e.g. during development of a client, or for troubleshooting). This document covers connecting to the CLI and its command set.
- **SmartMesh IP VMManager API Guide** - used for programmatic interaction with a VMManager. This document covers connecting to the API and its command set.
- **SmartMesh IP Mote CLI Guide** - used for human interaction with a mote (e.g. during development of a sensor application, or for troubleshooting). This document covers connecting to the CLI and its command set.
• **SmartMesh IP Mote API Guide** - used for programmatic interaction with a mote. This document covers connecting to the API and its command set.

### 1.1.4 Access Point Motes

• **SmartMesh IP User's Guide** - describes reprogramming DC2274 for use as an Access Point Mote.
• **VManger AP Bridge User’s Guide** - user’s guide for the Access Point Bridge reference software

### 1.1.5 Software Development Tools

• **Dustcloud.org** - contains documentation and links to various open source software tools for iexercising mote and manager APIs and visualizing the network.

### 1.1.6 Application Notes

• **SmartMesh IP Application Notes** - Cover a wide range of topics specific to SmartMesh IP networks and topics that apply to SmartMesh networks in general.

### 1.1.7 Documents Useful When Starting a New Design

• The Datasheet for the mote being used, e.g. the LTC5800-IPM SoC, or one of the modules based on it.
• The Datasheet for the embedded manager being used, e.g. the LTC5800-IPR SoC, or one of the embedded managers based on it.
• A **Hardware Integration Guide** for the mote/manager SoC or module - this discusses best practices for integrating the SoC or module into your design.
• A **Hardware Integration Guide** for the embedded manager - this discusses best practices for integrating the embedded manager into your design.
• A **Board Specific Integration Guide** - For SoC motes and Managers. Discusses how to set default IO configuration and crystal calibration information via a “fuse table”.
• **Hardware Integration Application Notes** - contains an SoC design checklist, antenna selection guide, etc.
• The **ESP Programmer Guide** - a guide to the DC9010 Programmer Board and ESP software used to load firmware on a device.
1.1.8 Software

- ESP software - used to program firmware images onto a mote or module. Described in the [ESP Programmer Guide](#).
- Fuse Table software - used to construct the fuse table as discussed in the [Board Specific Configuration Guide](#).

1.1.9 Other Useful Documents

- A list of [Frequently Asked Questions](#).
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, menu commands, and device states and modes.

**Italic type** is used to introduce a new term, and to refer to APIs and their parameters.

![Tips](image) Tips provide useful information about the product.

![Informational](image) Informational text provides additional information for background and context.

![Notes](image) Notes provide more detailed information about concepts.

![Warning](image) 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.**
# 1.3 Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12/17/2015</td>
<td>Initial Release</td>
</tr>
<tr>
<td>2</td>
<td>08/19/2016</td>
<td>Phase I Production</td>
</tr>
<tr>
<td>3</td>
<td>11/07/2016</td>
<td>Clarified stunnel configuration; Described disconnection and reconnection; Numerous small improvements</td>
</tr>
<tr>
<td>4</td>
<td>03/15/2017</td>
<td>Minor updates</td>
</tr>
</tbody>
</table>
2 Introduction

When the Access Points are required to be physically separated from where the VManager is installed, a gateway is required. The gateway could be as simple or as complex as is required for the application, but will typically be a device that acts as a bridge between the AP Motes and a TCP/IP connection to the VManager. That TCP/IP connection could be a standard hard-wired RJ-45 connection, WiFi, Cellular, etc.

For convenience and easy prototyping of a system, an AP Bridge reference gateway has been built for the Raspberry Pi 2 and a fully functional prebuilt image is available for download and installation. The source code and project is also available for anyone who wishes to port to a different Linux based platform, such as a Beagle Bone. Refer to the AP Bridge Integrator’s Guide on dustcloud.org for more information.
3 Installation

For convenience and easy prototyping of a system, an AP Bridge reference gateway has been built for the Raspberry Pi 2, and the full install image is available through your Linear contact.

3.1 Raspberry Pi Software Installation

The reference AP Bridge is available as a complete system image. The image is available for download through your MyLinear account. Contact your local sales representative to gain access through your myLinear document locker.

- raspbian-jessie-lite-dust-XXXX.zip

This file contains the following:

- Raspbian Jessie Lite from Raspberry Pi official website
- Most recent AP Bridge Software release
- AP Bridge runtime dependencies

Download the .zip file and unzip it to retrieve the raspbian-jessie-lite.img file.

⚠️ It is highly recommended to use a high quality 16 GByte (or larger) Class 10 SD card.

Prepare the SD card - Using a Windows system:

- Download the Win32DiskImager utility from the Sourceforge Project page
- Extract the executable from the Win32DiskImager.zip file and run the Win32DiskImager utility. You may need to run the utility as administrator by right-clicking the file and select Run as administrator.
- Select the image file extracted earlier and write it to the SD card. Be sure to double check that the volume matches that of the SD card, as the utility will erase whatever volume is selected, including your primary system drive.
- After the write finishes, exit Win32DiskImager and eject the SD card.

Complete the installation - On the Raspberry Pi (rPi)

- While powered down, insert your SD card into the rPi
- Connect an HDMI monitor, and a USB keyboard
- Connect an Ethernet cable
- Connect the mini USB power cable to boot
- Once booted, log in (username = dust and password = dust)
- The raspi-config utility will appear on the first boot up. Choose option 1 - Expand File System (use the tab key to highlight the Select button to proceed)
3.2 IP Address Configuration

By default, the rPi uses DHCP to obtain an IP address. If you need to know this address to connect via SSH, use the `ifconfig` command to get the Ethernet (eth0) IP address (inet addr):

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:3d:74:72
          inet addr:10.70.7.138  Bcast:10.70.7.255  Mask:255.255.255.0
          inet6 addr: fe80::254:68ff:fe0b:97a2/64 Scope:link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:100
          RX packets:6799 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2751 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:618011 (597.3 KiB)  TX bytes:362569 (349.1 KiB)
          Interrupt:12  Base address:0x300
```

Refer to the online guides, such as [How to give your Raspberry Pi a Static IP Address](#), for the Jessie release of Raspbian for instructions on how to configure a static IP address if required.

Now that the IP address is known, it will be possible use SSH to connect to the rPi.

```
$ ssh dust@192.168.1.10 -p 22
```

3.3 Adding and Configuring AP Motes

The system is now ready for the final installation step, namely adding AP Motes.

- Make sure that the rPi is powered ON
- Connect one or more AP Motes to the USB port

Within a few seconds, the devices should be detected and configured by the software.

To verify that the AP Mote installation was successful, run the following command. An entry should appear for each AP Mote installed.

```
$ apcctl status
apc-30080e           RUNNING    pid 3406, uptime 0:03:30
```

If an AP mote is plugged in for the first time while the Raspberry Pi either powered OFF or is in the process of booting, the configuration files may not be created automatically. If this happens, remove the AP Motes and re-connect them once the system is completely booted.
3.4 Adding or Removing an AP Mote

Additional AP Motes can be added at any time when the system is powered on. The AP Bridge software will automatically recognize an AP Mote and create the required configuration. If an AP Mote is removed from the system, it is recommended to uninstall the matching configuration. While the AP Bridge software remains configured for an unconnected device, the system will attempt to reconnect to the missing device.

To remove a device, unplug it from the rPi USB connector, and follow the steps below to remove the device’s configuration.

```bash
$ apcctl status
apc-30080e                       RUNNING    pid 3406, uptime 0:03:30
$ update-apc-config delete --name apc-30080e
```

In the event that all AP Motes on a system need to be swapped, the following commands can be used as a faster way of doing it.

```bash
$ update-apc-config delete --name all
$ update-apc-config auto
```
4 Configuring the AP Bridge

The complete system that has been built by following all of the steps in the Installation section of this manual is referred to as an "AP Bridge".

The following configurations steps are required to connect the AP Bridge to a VManager so that a network can be formed, or so the AP Bridge can participate in an existing network.

4.1 VManager Association

When the AP Bridge is running separately from the VManager host, the AP Bridge must be configured to connect securely to the desired VManager. The connection between the AP Bridge and the VManager is secured by encapsulating the TCP communication with a TLS session using **stunnel**. The configuration is performed by running the `update-apc-config stunnel` command.

⚠️ This step is only necessary when the AP Bridge software is running on a separate host from the VManager. Do not perform this step on the VManager VM.

- Login to the AP Bridge host (e.g. a Raspberry Pi) with the 'dust' user, either locally or remotely with SSH.
- Execute the following commands

```
$ update-apc-config stunnel --host 10.70.48.46
/opt/dust-apc/etc/stunnel/apc_stunnel.conf created
installed stunnel conf file
Stopping SSL tunnels: [stopped: /etc/stunnel/apc_stunnel.conf] stunnel.
   stopped stunnel
   started stunnel

$ update-apc-config stunnel
Stunnel configuration
   client: yes
   accept: 9100
   connect: 10.70.48.46:9101
```

4.2 AP Mote Configuration

The AP Mote(s) may need to have their configuration changed if the default configuration is not suitable. The only user-settable parameters on the AP Mote are TX power, join key and clock source. These parameters can be changed using the APC CLI Console.
For the VManager 1.0.1 release, the only supported clock source is the default value, *Auto*.

Each AP Mote connected to the system can be configured individually. There is an instance of the APC software running for each AP Mote connected to the system. The APC CLI Console is used to pass the configuration parameters to the AP Mote. To configure a specific AP Mote:

- **Identify the name of the APC process connected to the desired AP Mote by issuing the `apcctl status` command.**
  The last three bytes of the AP Mote MAC address are part of the APC instance name.

  ```
  $ apcctl status
  apc-30080e                       RUNNING    pid 3967, uptime 0:00:11
  apc-6036e3                       RUNNING    pid 3851, uptime 0:37:47
  ```

- **Open the AP Bridge CLI.** If there is only one AP Bridge running, no parameters are required, otherwise use the `-n` switch with the name of the APC to be configured.

  ```
  $ apc-console -n apc-30080e
  Welcome to the APC CLI Console on Linux Version 1.0.1.10
  $>
  ```

- **Configure the AP Mote.** For example, use the `set` command to specify the join key matching the VManager’s ACL.
- **Exit the CLI**
5 CLI Commands

The APC Console is written in Python and provides access to the APC components. This component is meant as a reference and a final customer AP Gateway is free to implement part or all of this functionality.

To get into the APC CLI, use the following:

```
apc-console [-n <apc-xxxxxx>]
   If only one APC instance is running on the system, no parameters are required
```

The apc service name can be retrieved with the command "apcctl status"

```
$ apcctl status
apc-30080e       RUNNING   pid 3967, uptime 0:00:11
apc-6036e3        RUNNING   pid 3851, uptime 0:37:47

$ apc-console -n apc-30080e
Welcome to the APC CLI Console on Linux
Version 1.0.1.10
$>
```
5.1 clear

Description

This command clears the apc console screen

Syntax

```
clear
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> clear
```
5.2 exit/logout/quit

Description
This command exits the console.

Syntax

```
<exit|logout|quit>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> logout
```
5.3 get

Description

This command displays information about the APC or AP(s)

Syntax

get <ap|apc> [args]

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap</td>
<td>Display information about the AP, where [args] is one of</td>
</tr>
<tr>
<td></td>
<td>• apInfo</td>
</tr>
<tr>
<td></td>
<td>• apStatus</td>
</tr>
<tr>
<td></td>
<td>• clkSrc</td>
</tr>
<tr>
<td></td>
<td>• macAddr</td>
</tr>
<tr>
<td></td>
<td>• netid</td>
</tr>
<tr>
<td></td>
<td>• time</td>
</tr>
<tr>
<td></td>
<td>• txpwr</td>
</tr>
<tr>
<td>apc</td>
<td>Display information about the APC, where [args] is one of</td>
</tr>
<tr>
<td></td>
<td>• clientId</td>
</tr>
<tr>
<td></td>
<td>• gpsState</td>
</tr>
<tr>
<td></td>
<td>• managerHost</td>
</tr>
<tr>
<td></td>
<td>• managerPort</td>
</tr>
</tbody>
</table>

Example
$> \text{get ap macAddr}
00-17-0D-00-00-12-34-56

$> \text{get ap apstatus}
DN_API_ST_OPERATIONAL

$> \text{get apc gpsstate}
GPS_ST_NO_DEVICE

$> \text{get ap clksrc}
AUTO

Notes:

- \textit{txpwr} - the get command will return the value selected by the radio driver plus the configured antenna gain value. Refer to the AP Mote Commands section for the procedure for setting the antGain parameter.
- \textit{clkSrc} - the get command will return the configured value for the AP Mote, AUTO or GPS.
5.4 help

Description

This command list all commands apc console supports.

Syntax

help

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command list all available commands.</td>
</tr>
<tr>
<td>command</td>
<td>This command shows detailed help for the command.</td>
</tr>
</tbody>
</table>

Example

$> help
Documented commands (type help <topic>):
========================================
clear  get  info  logout  reset  stats  subscribe  unsubscribe
exit   help  logger  quit   set   stop   trace   version
5.5 info

Description

This command displays information about the VManager, the AP Mote (AP), the AP Bridge software (APC), and the status of GPS time source.

Syntax

```plaintext
info
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> info
Manager Host                                     localhost
Manager Port                                          9100
Manager State                       APCCLIENT_STATE_ONLINE
AP Mac Address                     00-17-0D-00-00-30-08-0E
AP State                             DN_API_ST_OPERATIONAL
AP Version                                        1.4.0.72
AP Clock Source              DN_API_AP_CLK_SOURCE_INTERNAL
APC Client Id                                   apc-30080e
APC Version            1.0.1.3 (built 2016/03/18 13:34:59)
GPS State                                 GPS_ST_NO_DEVICE
Satellites Used                                          0
Satellites Visible                                       0
```

5.5.1 VManager info

- Manager Host - The host name of the VManager. It is configurable in the apc configuration file. With stunnel used, it will be localhost or 127.0.0.1
- Manager Port - The TCP port number that the VManager is monitoring for a connection request
- Manager State - Indicates whether the APC is connected to the VManager, possible values are APCCLIENT_STATE_ONLINE or APCCLIENT_STATE_OFFLINE
5.5.2 AP Mote info

- AP MAC Address - MAC address of the AP
- AP State - One of DN_API_ST_IDLE, DN_API_ST_SEARCHING, DN_API_ST_CONNECTED, or DN_API_ST_OPERATIONAL
- AP Version - The AP software version, it has the same value of "info" from AP CLI.
- AP Clock Source - One of DN_API_AP_CLK_SOURCE_INTERNAL, DN_API_AP_CLK_SOURCE_NETWORK or DN_API_AP_CLK_SOURCE_PPS

5.5.3 AP Bridge Info

- APC Client Id - APC client name.
- APC Version - The APC version automatically generated by Dust build system.

5.5.4 GPS Info

- GPS State - One of GPS_ST_NO_DEVICE, GPS_ST_NO_SYNC, GPS_ST_SYNC or GPS_ST_CONN_LOST
- Satellites Used - The number of satellites locked by the GPS device
- Satellites Visible - The number of satellites tracked by the GPS device
5.6 set

Description

This command is used to set the AP Mote parameters specific to a given deployment. Additionally, the log level for the specified logger can be set.

Syntax

```
set <ap|loglevel> <arg> <value>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap</td>
<td>Set the AP parameter specified by &lt;arg&gt; to &lt;value&gt;. Valid &lt;arg&gt; options are:</td>
</tr>
<tr>
<td></td>
<td>- clkSrc [ AUTO</td>
</tr>
<tr>
<td></td>
<td>- jkey (in hex)</td>
</tr>
<tr>
<td></td>
<td>- txpwr</td>
</tr>
<tr>
<td>loglevel</td>
<td>Set the logger specified by &lt;arg&gt; to the log level &lt;value&gt;. Valid loggers are listed by the logger command. The following log levels are available and are specified by name in the &lt;value&gt; field:</td>
</tr>
<tr>
<td></td>
<td>- FATAL (1)</td>
</tr>
<tr>
<td></td>
<td>- ERR (2)</td>
</tr>
<tr>
<td></td>
<td>- WARN (3)</td>
</tr>
<tr>
<td></td>
<td>- INFO (4)</td>
</tr>
<tr>
<td></td>
<td>- DEBUG (5)</td>
</tr>
<tr>
<td></td>
<td>- TRACE (6)</td>
</tr>
</tbody>
</table>
$> \textit{set ap clkSrc auto}

$> \textit{set ap jkey 445553544E4554574F524B53524F434B}

$> \textit{set loglevel apc INFO}

\textit{Done}

Notes:

- \textit{clksrc} - By default, the AP Bridge configuration sets the clock source to "Auto". If a GPS time source is used, the AP Bridge configuration must be set to "GPS".

- \textit{jkey} - The AP Mote join key must match the join key specified in the VManager’s ACL. This setting is persisted in the AP Mote, however not persisted in the AP Bridge software. This is to allow an \textit{exchangeJoinKey} to be performed by the VManager.

- \textit{txpwr} - The \textit{txpwr} configuration must be set on each AP Mote separately if the default value is not used. The \textit{txpwr} command may be issued at any time and takes effect immediately. If the provided \textit{txpwr} does not match an appropriate value for the hardware, the AP Mote will select the nearest appropriate value. The nearest appropriate value varies depending on the hardware and calibration.
5.7 version

Description

This command displays the AP Bridge Console version

Syntax

```
version
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> version
Version 1.0.1.9
```
5.8 CLI Debug Commands

The following commands are for intended for the purpose of debugging a system and should not be used under normal circumstances.

5.8.1 logger

Description

This command lists available loggers or the details about a specific logger.

Syntax

```plaintext
logger [logger_name]
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logger_name</td>
<td>The name of the logger to display</td>
</tr>
</tbody>
</table>

Example

```
$>  logger
Loggers: apc, apc.client, apc.io, apc.main, apm, apm.io, apm.io.raw, apm.io.serial, rpc.server, rpc.worker, rpc.worker.internalmid

$>  logger apc
Logger: apc, rc: 0, logLevel: L_INFO
```
5.8.2 reset

Description

This command resets the AP

Syntax

```
reset ap
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> reset ap
```
5.8.3 stats

Description

This command displays or clears the AP or manager stats

Syntax

```
stats <ap|manager|clear>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter (one of the following)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap</td>
<td>Display APC &lt;-&gt; AP stats</td>
</tr>
<tr>
<td>manager</td>
<td>Display APC &lt;-&gt; manager stats</td>
</tr>
<tr>
<td>clear</td>
<td>Clear all stats</td>
</tr>
</tbody>
</table>

Example
$> stats
AP/ APC Statistics

$> stats clear

AP/ APC Statistics

Packets Rcvd includes response and notifications. Usually Packets Rcvd will be bigger than Response Rcvd.
5.8.4 stop

Description

This command unsubscribes to all traces. This is a shortcut and does exactly the same thing as "unsubscribe all"

Syntax

```
stop
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command has no parameters</td>
</tr>
</tbody>
</table>

Example

```
$> stop
```
5.8.5 subscribe/unsubscribe

Description

This command subscribe or unsubscribe to the specified notification type or all types.

Syntax

```
subscribe|unsubscribe [notifType|all]
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifType or all</td>
<td>Subscribe or unsubscribe to the specified notification type or all types</td>
</tr>
</tbody>
</table>

Example

```
$> subscribe
logListener:
apc, apc.client, apc.io, apc.main, apm, apm.io, apm.io.raw, apm.io.serial, rpc.server, rpc.worker

$> subscribe apc
Subscribed to apc

$> unsubscribe all
Done
```
5.8.6 trace

Description

This command turns the trace for a specified notification type. The log level must be set to debug or trace to see the enabled trace.

Syntax

```
trace <notifType|all> <on|off>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifType or all</td>
<td>Trace the specified notification type</td>
</tr>
<tr>
<td>on or off</td>
<td>Turn the specified trace on or off</td>
</tr>
</tbody>
</table>

Example

```
$>  trace apm.io on
Trace enabled for apm.io

$>  logger apm.io
Logger: apm.io, rc: 0, level=logLevel: L_DEBUG

$> set loglevel apm.io trace
Done

$> 2015-12-16 21:10:00.765 apm.io: L_DEBUG Ping interval expired
2015-12-16 21:10:00.769 apm.io: L_DEBUG Sending ping
2015-12-16 21:10:00.773 apm.io: L_DEBUG Sending packet, cmdId: 0x2
2015-12-16 21:10:00.794 apm.io: L_DEBUG INP cmd: 0x2 RSP:2
2015-12-16 21:10:00.797 apm.io: L_DEBUG INP ACK cmd: 0x2 rc: 0x0
2015-12-16 21:10:00.807 apm.io: L_DEBUG AP RX GetParam: paramId=1
2015-12-16 21:10:01.794 apm.io: L_DEBUG Ping interval expired
2015-12-16 21:10:01.795 apm.io: L_DEBUG Sending ping
2015-12-16 21:10:01.799 apm.io: L_DEBUG Sending packet, cmdId: 0x2
2015-12-16 21:10:01.808 apm.io: L_DEBUG INP cmd: 0x2 RSP:0
2015-12-16 21:10:01.809 apm.io: L_DEBUG INP ACK cmd: 0x2 rc: 0x0
2015-12-16 21:10:01.819 apm.io: L_DEBUG AP RX GetParam: paramId=1
2015-12-16 21:10:01.898 apm.io: L_DEBUG INP cmd: 0x27 REQ:0
2015-12-16 21:10:01.900 apm.io: L_TRACE received AP data
2015-12-16 21:10:01.909 apm.io: L_DEBUG OUT ACK cmd:27
2015-12-16 21:10:02.771 apm.io: L_DEBUG INP cmd: 0x27 REQ:2
2015-12-16 21:10:02.776 apm.io: L_TRACE received AP data
...
```
6 AP Mote Commands

As with a regular Mote, the AP Mote provides a command line interface. This interface should only be used for diagnostic purposes and should not be required for normal use. The only items that must be configured when setting up an AP Mote for use with a manager are typically the join key and the clock source, both of which have been added to the AP Bridge software CLI for convenience. Note that any setting found in the AP Bridge software will take precedence over any setting change made directly on the AP Mote.

Similarly to the IP Mote, the AP Mote CLI can be accessed from any serial terminal program from the AP Bridge host device. If connecting to a DC2274-A 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

On the VManager VM or the Raspberry Pi-based reference design for the AP Bridge, miniterm.py can be used to connect to the AP Mote CLI port.
6.1 info

Description
Displays information about the application layer.

Syntax

```
info
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

Example

```
> info

AP IP
ver:01.04.00.74
Radio Test: off
```
6.2 get

Description

This command allows to get the values of application level configurable parameters.

Syntax

```
get <parameter>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sri</td>
<td>Serial Retry Interval (in msec)</td>
</tr>
</tbody>
</table>

Example

```
> get sri
20 msec
```
6.3 set

Description

This command allows to set application level configurable parameters.

Syntax

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sri</td>
<td>Serial Retry Interval (in msec). Takes effect after reset</td>
</tr>
</tbody>
</table>

Example

```
> set sri 500
```
6.4 minfo

Description

Displays various information about the AP Mote

Syntax

```plaintext
minfo
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

Example

```
> minfo
Net stack 1.4.0.10
state: Idle
mac: 00:17:0d:00:00:3f:fe:36
moteid: 0
netid: 258
blSwVer: 13
ldrSwVer: 1.0.6.3
board id/rev:0x2/0x3
UTC time: 1025665200.015 sec
reset st: 400
battery: 3553 mV
temp: 27 C
```
6.5 mget

Description

This command is used to retrieve the values of configurable parameters in the network stack.

Syntax

```
mget <param>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clkSrc</td>
<td>Clock source (2=GPS PPS, 3=Auto). The default value is ‘Auto’, and values 0 and 1 are reserved.</td>
</tr>
<tr>
<td>txpwr</td>
<td>Radio transmit power. The <code>mget txpwr</code> command returns the persistent configuration value, which matches the value from the last <code>mset txpwr</code> command. The persistent value may be different from the value that is used by the radio.</td>
</tr>
<tr>
<td>antGain</td>
<td>Antenna gain (signed 8-bit value) - used to properly calculate radiated power. Defaults to +2 dBi</td>
</tr>
<tr>
<td>compMode</td>
<td>Constrains AP duty cycle to power-appropriate limits imposed by EN 300 328. 0=off (default), 1 = on.</td>
</tr>
</tbody>
</table>

Example

```
> mget clkSrc
clkSrc = 0
```
6.6 mset

Description

This command allows to set configurable parameters in the network stack.

Syntax

mset <param> <value>

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| clkSrc    | Clock source (2=GPS PPS, 3=Auto). Default is 'Auto', and 0 and 1 are reserved
  |
|           | * The reference AP Bridge software settings take precedence and will overwrite any changes made on the AP Mote |
| txpwr     | Radio transmit power. The value takes effect on next transmission. Refer to product datasheets for supported RF output power values. If the provided value does not match an appropriate value for the hardware, the radio driver will select the nearest appropriate value. The nearest appropriate value varies depending on the hardware and calibration. The mset command stores the provided value persistently. The mget txpwr command will return the persistent value. The getParameter<txpwr> API command will return the value selected by the radio driver. |
| antGain   | Antenna gain (INT8S) - needed to properly calculate radiated power. Defaults to +2 dBi |
| compMode  | Constrains AP duty cycle to power-appropriate limits imposed by EN 300 328. 0=off (default), 1 = on. |
| jkey      | Join key (hex) |

Example

> mset clkSrc 3
## 6.7 reset

**Description**

Resets the AP

**Syntax**

```
reset
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

**Example**

```
> reset
>
> AP IP 1.4.0-74 (0x100)
```
6.8 Diagnostic and debug commands

6.8.1 mfs

Description

File system commands. These are intended for debugging.

- The zeroize command will render the device inoperable. It must be re-programmed via SPI or JTAG in order to be usable.

Syntax

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>One of:</td>
</tr>
<tr>
<td></td>
<td>show - show a list of files (-f) or partitions (-p)</td>
</tr>
<tr>
<td></td>
<td>fcs - calculate CRC for a filename (-f &lt;filename&gt;) or partition (-p &lt;parId&gt; &lt;offset&gt; &lt;length&gt;)</td>
</tr>
<tr>
<td></td>
<td>del - delete file (-f &lt;filename&gt;)</td>
</tr>
<tr>
<td></td>
<td>zeroize &lt;password&gt; - zeroize device keys per FIPS-140 requirements. The password is 57005 (0xDEAD).</td>
</tr>
</tbody>
</table>

Example

```
> mfs show -p
ID  Size    Address  Page
 1  32768  0x000b7800 2048 exec
 2  258048 0x00041000 2048 exec
 4  227328 0x00080000 2048
 6   2048  0x00bf8000 2048

> mfs show -f
 1mote.cfg 12 shadow
 1ini.cfg 0 shadow
 2main.cfg 8 shadow
```
Partitions:

- ID - the partition ID
- Size - Partition size in bytes
- Address - Starting address of partition
- Page - Page size in bytes
- Pages marked as exec can contain an executable image

Files:

- 1st column - Filename. Files starting with 1 are created by the network stack. Files starting with a 2 are created by the AP Mote application.
- 2nd column - Size in bytes. 0 indicates an empty file
- 3rd column - Indicates whether the file is shadowed (there is a backup copy) or temporary
6.8.2 radiotest

radiotest on/off

Description

Enables or disables radiotest mode. Radiotest functionality can be used to exercise the radio for certification and testing purposes. This command takes effect after reboot and the selected mode persists until changed, i.e. if ON, it will remain on even after reset or power cycle until the mode is set to OFF and the AP Mote is rebooted.

Syntax

```
radiotest <mode>
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>on = put AP Mote into radiotest mode after reboot</td>
</tr>
<tr>
<td></td>
<td>off = put AP Mote into normal mode after reboot</td>
</tr>
</tbody>
</table>

Example

```
AP IP 1.4.0-74 (0x100)
>
> radiotest on
>
AP IP 1.4.0-74 (0x200)
Radio Test on
```
**radiotest tx**

**Description**

The `radiotest tx` command allows the user to initiate a radio transmission test. This command may only be issued in `radiotest` mode. These types of transmission tests are supported:

- **pk** - Packet Transmission
- **cm** - Continuous Modulation
- **cw** - Continuous Wave (unmodulated signal)
- **pkcca** - Packet transmission with clear channel assessment (CCA) enabled

In a packet transmission test, AP Mote generates a `repeatCnt` number of packet sequences. Each sequence consists of up to 10 packets with configurable sizes and delays. Each packet consists of a payload of up to 125 bytes, and a 2-byte 802.15.4 CRC at the end. Byte 0 contains sender’s stationID. Bytes 1 and 2 contain the packet number (in big-endian format) that increments with every packet transmitted. Bytes 3..N contain a counter (from 0..N-3) that increments with every byte inside payload. Transmissions occur on the set of channels defined by `chanMask`, selected in pseudo-random order.

In a continuous modulation test, AP Mote generates continuous pseudo-random modulated signal, centered at the specified single channel. The test is stopped by resetting the AP Mote.

In a continuous wave test, the AP Mote generates an unmodulated tone, centered at the specified single channel. The test tone is stopped by resetting the AP Mote.

In a packet transmission with CCA test, the AP Mote is configured identically to that in the packet transmission test, however it does a clear channel assessment before each transmission and aborts that packet if the channel is busy.

⚠️ Channel numbering is 0-15, corresponding to IEEE 2.4 GHz channels 11-26.

**Syntax**

```
radiotest tx <testType> <chanMask> <power> [<stationId> <repeatCnt> {<pkLen><delay>...}]
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>testType</code></td>
<td>Type of tx to initiate: ‘pk’ = packets, ‘cm’ = continuous modulation, ‘cw’ = continuous wave, “pkcca” = packets with CCA.</td>
</tr>
<tr>
<td><code>chanMask</code></td>
<td>Hexadecimal bitmask of channels (0–15) for the test. Bit 0 corresponds to channel 0. For continuous wave and continuous modulation tests, only one channel should be enabled.</td>
</tr>
<tr>
<td><code>power</code></td>
<td>Transmit power, in dB.</td>
</tr>
<tr>
<td>stationId</td>
<td>Unique (0-255) station id of the sender. Must match station id value of the receiver.</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>repeatCnt</td>
<td>Number of times to repeat the packet sequence (0=do not stop). Applies only to packet transmission tests.</td>
</tr>
<tr>
<td>pkLen</td>
<td>Length of packet (2-125 bytes)</td>
</tr>
<tr>
<td>delay</td>
<td>Delay after transmission (in microseconds)</td>
</tr>
</tbody>
</table>

**Example**

Initiate packet test on channels 0,1 (chMap=0x03), with output tx power of 0 dBm, station id = 26
Repeat the sequence 5 times: 50-byte packet, 20ms delay, 30-byte packet, 20msec delay:

```
radiotest tx pk 0x3 0 26 5 50 20000 30 20000
```

Start transmission with continuous modulation on channel 0 with output tx power of 8 dB:

```
radiotest tx cm 0x1 8
```

Start transmission with continuous wave on channel 1 with output tx power of 8 dB:

```
radiotest tx cw 0x2 8
```
**radiotest rx**

**Description**

The `radiotest rx` command puts the radio into receive mode where statistics on packet reception are collected. The nonzero station id specified must match station id of the sender, which is necessary to isolate traffic of multiple tests running in the same radio space. Statistics may be viewed with the `radiotest stat` command.

⚠️ Channel numbering is 0-15, corresponding to IEEE 2.4 GHz channels 11-26.

**Syntax**

```
radiotest rx <chanMask> <time> <stationId>
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chanMask</td>
<td>Hexadecimal bitmask of channels (0–15) for the test. Bit 0 corresponds to channel 0. Only a single channel may be specified for this command.</td>
</tr>
<tr>
<td>time</td>
<td>Duration of receive test, in seconds. 0=do not stop</td>
</tr>
<tr>
<td>stationId</td>
<td>Unique (1-255) id of the receiver. Must match sender’s station id. Station id 0 may be used to accept packets from any sender.</td>
</tr>
</tbody>
</table>

**Example**

Put device into receive mode for 60 seconds on channel 2, use station id 26:

```
radiotest rx 0x4 60 26
```
radiotest stat

Description

This command displays packet reception statistics collected during the previously run `radiotest rx` command. This command may only be used when the device is in radiotest mode.

Syntax

```plaintext
radiotest stat
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

Example

```plaintext
>radiotest stat
Radio Test Statistics
  OkCnt   : 0
  FailCnt : 0
```
### 6.8.3 trace

**Description**

Turn application layer traces on or off. If called with no arguments, returns current state of all traces. If called with the argument ‘save’ it stores current settings to non-volatile memory.

**Syntax**

```
trace [save | {<module>|all on|off}
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>save</td>
<td>Saves all current traces</td>
</tr>
<tr>
<td>ser</td>
<td>Serial module traces</td>
</tr>
<tr>
<td>loc</td>
<td>Local (NET layer) module traces</td>
</tr>
<tr>
<td>clk</td>
<td>Clock-source module traces</td>
</tr>
</tbody>
</table>

**Example**

```
> trace ser on
> 7578562 : serial_api TX:
   000 : 0f 09 0a 00 00 00 01 01 00 00 00 00
7579121 : serial_api TX:
   000 : 0f 09 0a 00 00 00 01 01 00 00 00 00
7579681 : serial_api TX:
   000 : 0f 09 0a 00 00 00 01 01 00 00 00 00
```
6.8.4 mtrace

Description

Turn various traces on or off. This change is persistent if called with the 'save' parameter. If called with no arguments, returns current state of all mtraces.

Syntax

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>save</td>
<td>Save current mtrace flags to flash</td>
</tr>
<tr>
<td>mac</td>
<td>MAC layer TXs and RXs</td>
</tr>
<tr>
<td>io</td>
<td>NET layer TXs and RXs</td>
</tr>
</tbody>
</table>

Example

```
> mtrace mac on
7497319 : MAC R: a=57423 t=7 ch=13 s=5 rc=0 rs=-23 ad=14 q=0,0
7497457 : MAC T: a=57442 t=7 ch=1 d=3 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
```
### 6.8.5 stats

**Description**

Displays UART statistics when called with 'get' argument, clears it when called with 'clear' argument.

**Syntax**

```plaintext
stats { get | clear }
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

**Example**

```plaintext
> stats get
UART stats:
  Tx total:  15615
  Rx total:  0
  Tx RtsCts tout: 0
  Rx CtsRx tout: 0
  Rx EopRts tout: 0
  Rx IB tout:  0
  Rx DMA miss: 0
  errors:  0
  no_first_flag: 0
  no_last_flag: 0
  overflow: 0
  pkg_len:  0
  status:
    crc:  0
    fram: 0
    par:  0
    ofl:  0
    ple:  0
  Notif fail: 0
  Q cnt:  1
  PO: 0
  PFE: 0
```
6.8.6 mlog

Description

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

Syntax

mlog

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

Example

> mlog
last task: 0
next task: 0
timer: 0x0
IPSR: 0x0
ISR: no
Low-level log: 'empty'
6.8.7 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. See the Board Specific Configuration Guide for fuse table details.

Syntax

mxtal [trim|meas] [<i>|<h>]

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trim</td>
<td>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 device; the function output should be used to determine 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.</td>
</tr>
<tr>
<td>meas</td>
<td>Outputs the ppm error of the 20MHz reference with value loaded from the fuse table. This function requires the device be connected to the programming board. It could take up to 30 sec for command to execute.</td>
</tr>
<tr>
<td>i</td>
<td>h</td>
</tr>
</tbody>
</table>

Example

> mxtal meas
Fuse Table pullVal used for measurement=95

> mxtal trim i
The optimal pullVal for this board is 90, which yields 0/16 PPM error
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 LT 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 Linear Technology 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 Linear Technology.

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.

Linear Technology 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.

Linear Technology 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. Linear Technology customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify and hold Linear Technology 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 Linear Technology was negligent regarding the design or manufacture of its products.

Linear Technology 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.
Linear Technology does not warrant or represent that any license, either express or implied, is granted under any Linear Technology patent right, copyright, mask work right, or other Linear Technology intellectual property right relating to any combination, machine, or process in which Linear Technology products or services are used. Information published by Linear Technology regarding third-party products or services does not constitute a license from Linear Technology 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 Linear Technology under the patents or other intellectual property of Linear Technology.

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

© Linear Technology Corp. 2012-2016 All Rights Reserved.