



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



SmartMesh IP Easy Start Guide for the Embedded Manager

Table of Contents

1	Introduction	3
1.1	Revision History	4
2	Setup	5
3	Basic Steps	7
3.1	Step 1: Form a Wireless Mesh Network	7
3.2	Step 2: Connect Your PC to the Manager	7
3.3	Step 3: Bring Up Manager CLI	8
3.4	Ping a Mote	9
3.5	That's It!	9
3.6	Next Steps - Steps In a Design	10
3.7	Troubleshooting	11
4	References	14

1 Introduction

The purpose of this document is to provide a quick and easy out-of-the-box experience with the [SmartMesh IP Starter Kit](#).

You will use the components highlighted in gray:

- A SmartMesh starter kit including a manager and one or more motes
- A terminal application to communicate with the manager's command line interface (CLI)
- An FTDI driver for converting USB connections to virtual COM ports

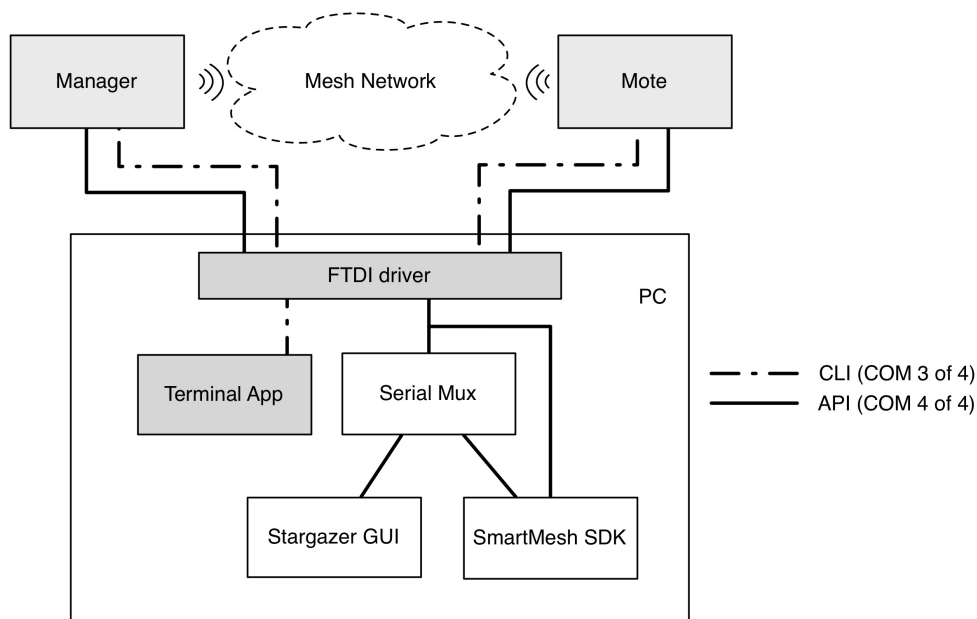


Figure 1: SmartMesh Demonstration Software Components

1.1 Revision History

Revision	Date	Description
1	03/18/2013	Initial Release
2	07/26/2013	Added DC9021A Kit
3	10/22/2013	Clarification of reset requirements, other minor corrections
4	04/04/2014	Numerous small clarifications
5	10/24/2014	Added DC9004A
6	04/22/2015	Minor corrections
7	12/03/2015	Added DC2274A; simplified content; renamed guide to focus on embedded manager

2 Setup

Figure 2 shows the Evaluation Kit including a SmartMesh IP Manager (with interface board), five Eterna motes, and an additional interface board. There are different manager/mote combinations available, as shown in Table 1.

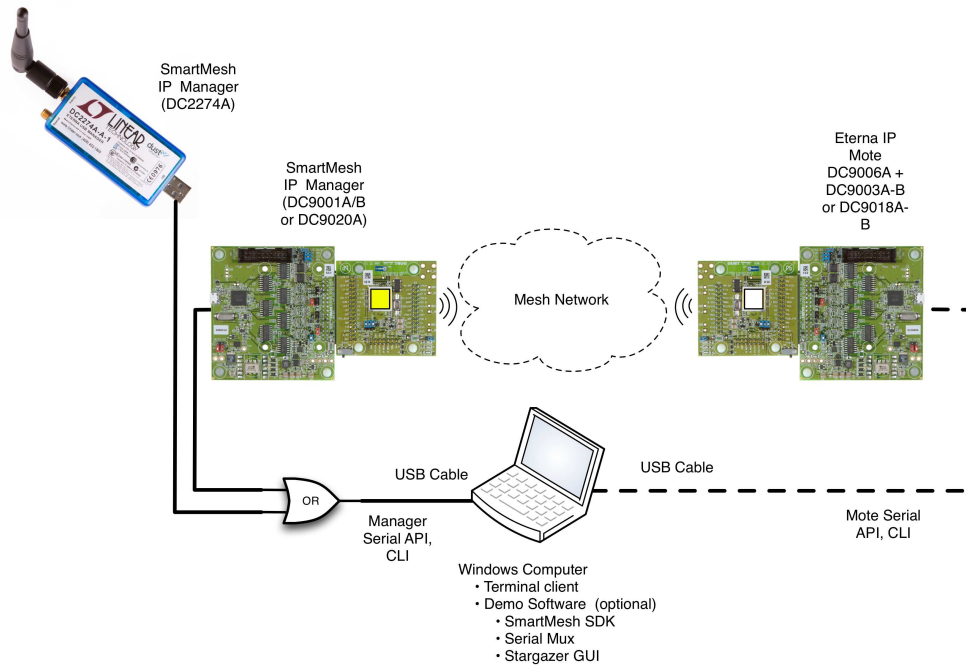


Figure 2: SmartMesh Evaluation Kit

Kit	Mote	Manager	Notes
DC9000A	DC9003A-B	DC9001A*	Eval/Dev Kit with 32 mote manager (discontinued)
DC9000B	DC9003A-B	DC9001B* or DC2274A-A	Eval/Dev kit with 100 mote manager
DC9021A	DC9018A-B	DC9020A* or DC2274A-A	RF Certified Eval/Dev kit with 100 mote manager

Table 1 - Kit configurations

*Orderable part number includes [DC9006](#). Manager boards (**yellow** sticker) may be marked with a different sub-component part number (e.g. DC9011A) than the value in Table 1.

- 1) The interface boards for both the manager and mote are identical (DC9006).
- 2) The DC2274A-A manager does not come with a DC9006 interface board. It connects directly via USB.

Standard kit motes ship with chip antennas. RF Certified devices are available for order individually with MMCX connectors:

- Mote - DC9018B-B (MMCX connector)
- Manager - DC9020B (MMCX connector)


The DC2274A-A manager has an external removable RPSMA antenna.

3 Basic Steps

There are three steps to demonstrate the capabilities of the kit:

1. Form a wireless mesh network - just power everything on, and a network will form
2. Get manager COM ports working on two USB ports on your PC
3. Bring up the manager command line interface - this will allow you to see the network on the manager

Once you've done these three steps you'll perform one exercise - you'll ping a mote to see its temperature and battery voltage.

 The CLI is intended for human interaction with a mote or manager, e.g. during development, or for interactive troubleshooting. The API is intended for machine-to-machine communications, e.g. a host program talking to the manager or a sensor processor talking to the mote.

3.1 Step 1: Form a Wireless Mesh Network

Even though much of the effort associated with this guide involves using the PC, it is important to point out that your PC has nothing to do with forming a wireless mesh network. To illustrate this, power on the manager using the slide switch (labeled POWER) in the lower right corner. The Blue LED (labeled INDICATOR_0) will light to indicate that the manager is on and is attempting to form a network by sending out advertisements.

Power on one mote, again using the slide switch. Watch the two green LEDs. One will blink slowly, indicating the mote is searching for the network. When that green LED changes from blinking to solid, that means the mote has sent a join request. When the second green LED turns on, the mote has joined.

Power on all the remaining motes, and within one or two minutes you will have a complete wireless mesh network running. The rest of the steps we go through will provide ways to interact with that mesh network.


3.2 Step 2: Connect Your PC to the Manager

You will now connect the PC to the manager through a serial (COM) port.

Connect the manager assembly that was supplied with your kit, (DC9001B, DC9020A, or DC2274A-A) into one of your PC USB ports. Note which USB port you are using. To make keeping track of which COM port is used to connect software clients, you should consider this your "manager port" for all future use with this PC.

Four virtual COM ports should automatically be added. These can be viewed using the Windows Device Manager (Control Panel -> System -> Hardware -> Device Manager -> Ports). Note the four COM port numbers - for example, they may be "COM7", "COM8", "COM9" and "COM10". You will be using the third (COM9) for CLI and the fourth (COM10) for the API connection. Other OSes will label these ports /dev/ttyUSBx (Linux) or /dev/tty.usbserial-xxxx (OS X).

Most modern OSes come with FTDI drivers pre-installed, but you may have to install them if this doesn't happen automatically. Refer to the [SmartMesh IP Tools Guide](#) for detailed instructions on how to install FTDI drivers.


 It has been observed in some installations under Windows 7 that the serial ports do not enumerate in order, and the CLI and API ports may not be the 3rd and 4th ports, respectively. If this occurs, you will need to test each port using APIExplorer (in the SmartMesh SDK) to find the API port, and use a terminal program to find the CLI port.

3.3 Step 3: Bring Up Manager CLI

The CLI allows you to type commands from a PC to interact with the manager. Interacting with the CLI requires no software installation on the PC if you are running Windows XP, as it comes with the Hyperterminal program (Start -> All Programs -> Accessories -> Communication). If you are running Windows 7, you will need to download a serial terminal client (such as [TeraTerm](#) or PuTTY). Note that the boards must be powered on (slide switch labeled Power) to access CLI.

To connect to the manager CLI open the terminal client program. Connect to the third COM port (e.g. COM 9 from step 2) that was added when you plugged in the manager. The COM port settings are:

- 9600 bits per second (sometimes listed as bps or baud)
- 8 data bits
- No parity
- 1 stop bit
- No flow control

 Save this "connection profile" (in Hyperterm or TeraTerm) so that you can use it in future

Hit enter a few times and you should see a > prompt character. Type 'help' to see a list of commands:

```
> help
help <command>
Commands:
  login
  logout
```

Next login as 'user' by typing the command:

```
> login user
```

Use the `sm` command (this is short for "show motes") to obtain the list of connected motes:

```
> sm
MAC MoteId State Nbrs Links Joins Age StateTime
00-17-0D-00-00-38-0E-4D 1 Oper 4 20 1 0 0-05:31:44
00-17-0D-00-00-38-0E-4E 2 Oper 5 16 1 25 0-05:30:40
00-17-0D-00-00-38-0E-CD 3 Oper 2 10 1 7 0-05:29:29
00-17-0D-00-00-38-0E-77 4 Oper 3 11 1 25 0-04:55:25
00-17-0D-00-00-38-0E-4C 5 Oper 2 10 1 41 0-04:55:18
00-17-0D-00-00-38-0E-78 6 Oper 2 10 1 18 0-04:53:37
Number of motes (max 17): Total 6, Live 6, Joining 0
```

3.4 Ping a Mote

You can interact with your motes via the manager CLI. Type the following command to send a "ping" to a mote. The mote will respond with battery voltage (in mV) and temperature (in °C) and the manager will add a round-trip time when printing the response. You ping a mote using its mote ID, not its MAC Address.

```
> ping 2
Sending ping request to mote 2

> Ping response from mote 2, time= 540 msec v=3542 t=27
```

Note that this is an internal mote command, not an ICMPv6 Echo request.

3.5 That's It!

You've successfully set up a network and interacted with the manager and a mote. There are many other things you can do with the manager CLI, and typing 'help' after logging in reveals the full CLI command set. See the [SmartMesh IP Manager CLI Guide](#) for details. Many of the [SmartMesh IP Application Notes](#) involve interacting with the manager CLI.

3.6 Next Steps - Steps In a Design

With the starter kit, hardware design and software design may be done separately.

For Software design:

- The application note "Data Publishing for SmartMesh IP" walks through the steps needed to use the mote APIs to join a network and send data. The mote needs to be placed into **slave** mode via its CLI in order to activate the API - evaluation kit motes ship in **master** demo mode. See the " Master vs. Slave" section of the [SmartMesh IP User's Guide](#) for details on changing the mode
- The [SmartMesh IP User's Guide](#) defines basic network terms and concepts, and discusses the use of APIs at a high level
- At a minimum, a mote application needs to:
 - Configure any parameters needed prior to join (such as *joindutycycle*)
 - Use the *join* API to cause a mote to begin searching for a network
 - Monitor the mote state to see when it is ready to accept data
 - Open a socket and bind a UDP port to it
 - Request services in order to publish data
 - The [SmartMesh IP Mote API Guide](#) covers other commands to configure the mote
 - The [SmartMesh IP Mote CLI Guide](#) covers using the human interface to observe mote activity
- At a minimum, a host application connected to the manager needs to:
 - Configure any parameters needed prior to join (such as *networkID*)
 - Subscribe to notifications to observe mote status and collect data
 - The [SmartMesh IP Manager API Guide](#) covers other commands to configure the manager, e.g. configure security (use of ACL), or collect detailed statistics from Health Report notifications
 - The [SmartMesh IP Manager CLI Guide](#) covers using the human interface to observe manager activity (including traces of mote state or data)
- Some of the Advanced software topics covered in the SmartMesh IP User's Guide, [SmartMesh IP Tools Guide](#), and SmartMesh IP Application Notes include:
 - Over-the-Air-Programming
 - Using the TestRadio API commands for top-level assembly testing
 - Use of the Serial Multiplexer to support concurrent manager clients
 - Using a low-power border router for internet connectivity

For Hardware design:

- Select a hardware platform - modularly certified or chip level?
- The hardware integration guides (see the [SmartMesh IP webpage](#) for links to specific guides) cover the important considerations for robust hardware development

3.7 Troubleshooting

1. For kits containing a DC9003-based manager, the board with the yellow sticker is the manager. Motes have a white sticker. The interface boards ([DC9006](#)) for both manager and mote are the same. A DC2274A-A has a built-in USB connector and does not require an interface board.

2. If you are consistent with which USB port you plug the mote and manager into, the COM port assignments should remain consistent. Mark the physical USB ports on your machine for manager and mote usage respectively and then do not change or mix their usage. Moving to a new USB port will create a new set of virtual COM ports.

3. Using the Windows **Device Manager**, make sure that the baud rate is set to 115,200 for the API COM ports. Double-click the port and select the "Port Settings" tab. As an example, see Figure 3:

- COM 7-10 are for the manager: COM10 is for API
- COM 11-14 are for the mote: COM14 is for API

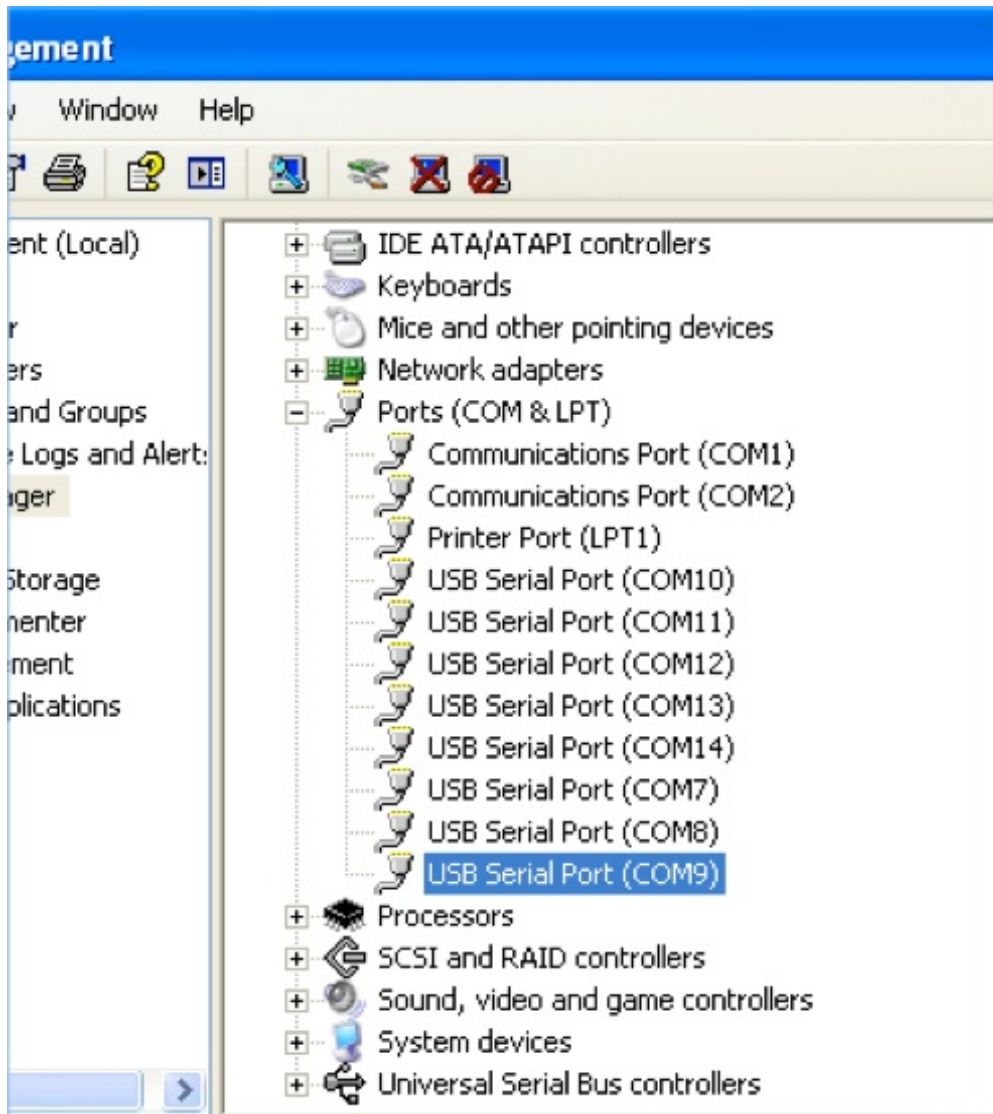


Figure 3: Device Manager (see step 2) Example

The baud rate for the manager and mote CLI will be set in your terminal program (e.g. Hyperterminal or TeraTerm).

4. By default, an evaluation kit mote joins automatically when it boots since it is in the **master** demo mode. If you set a mote to be in **slave** mode, the status LEDs don't turn on which you may mistake for a dead board/battery. In **slave** mode, the API is enabled and a *join* command must be given to have a mote join a network.

5. Status LED description:

Status 0	Status 1	Mote State
Blinking	Off	The mote is searching for the network
On	Off	The mote has found the network and is attempting to join

On	On	The mote has joined the network and is operational
----	----	--

7. Keep the **Device Manager** connected when you connect a mote to your PC to see the new COM port assignments for its CLI and API.

4 References

The following documents are available for the SmartMesh IP network:

Getting Started with a [Starter Kit](#)

- [SmartMesh IP Easy Start Guide](#) - walks you through basic installation and a few tests to make sure your network is working
- [SmartMesh IP Tools Guide](#) - the Installation section contains instructions for installing the serial drivers and example programs used in the Easy Start Guide and other tutorials.

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.

Interfaces for Interaction with a Device

- [SmartMesh IP Manager CLI Guide](#) - used for human interaction with a Manager (e.g. during development of a client, or for troubleshooting). This document covers connecting to the CLI and its command set.
- [SmartMesh IP Manager API Guide](#) - used for programmatic interaction with a manager. 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.

Software Development Tools

- [SmartMesh IP Tools Guide](#) - describes the various evaluation and development support tools included in the [SmartMesh SDK](#), including tools for exercising mote and manager APIs and visualizing the network.

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.

Documents Useful When Starting a New Design


- The Datasheet for the [LTC5800-IPM SoC](#), or one of the [modules](#) based on it.
- The Datasheet for 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.
- ESP software - used to program firmware images onto a mote or module.
- Fuse Table software - used to construct the fuse table as discussed in the [Board Specific Configuration Guide](#).

Other Useful Documents

- A glossary of wireless networking terms used in SmartMesh documentation can be found in the [SmartMesh IP User's Guide](#)
- A list of [Frequently Asked Questions](#)

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 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. 2012-2015 All Rights Reserved.