SmartMesh IP Easy Start Guide

Table of Contents

1 Introduction

The purpose of this document is to get the user a quick and easy positive out-of-the-box experience with the SmartMesh IP Starter Kit and Software Development Kit (SDK). More details are discussed in the SmartMesh IP Tools Guide. A number of software components are available for demonstrating interaction with your network. In this document, we will use the components highlighted in gray - FTDI drivers for converting USB connections to virtual COM ports; a terminal application; and the SmartMesh SDK, which allows the user to interact graphically with mote and manager APIs.

Figure 1: SmartMesh Software Components
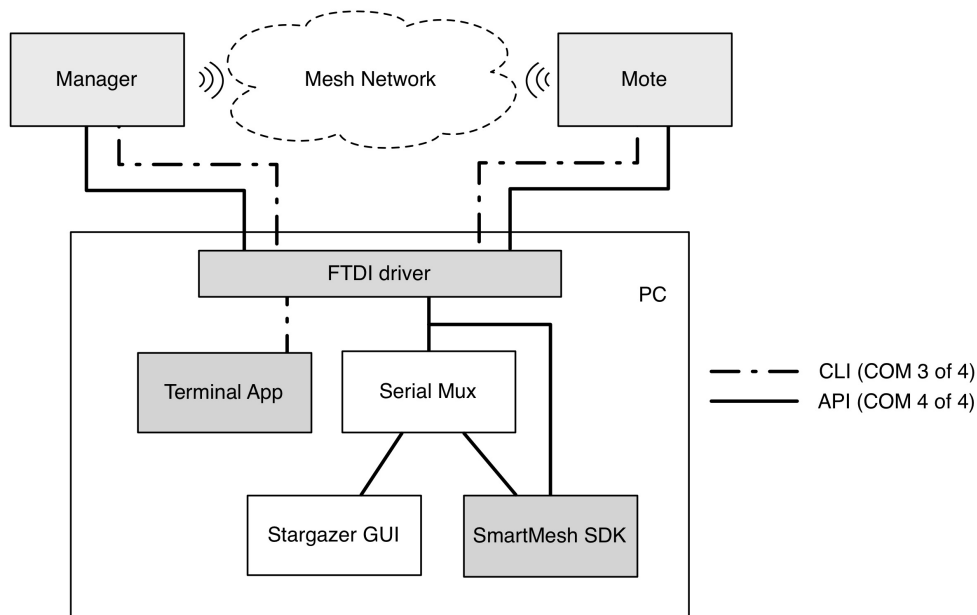
1.1 Revision History

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

2 Setup

Figure 2 shows the SDK 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.

KIt	Mote	Manager	Notes
DC9000A	DC9003A-B	DC9001A	Eval/Dev Kit with 32 mote manager
DC9000B	DC9003A-B	DC9001B	Eval/Dev kit with 100 mote manager
DC9021A	DC9018A-B	DC9020A	RF Certified Eval/Dev kit with 100 mote manager

> ⓘ All kits 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 SmartMesh IP Manager looks almost identical to the Mote out of the box. The **yellow** sticker identifies the Manager. The interface boards for both the Manager and Mote are identical (DC9006).
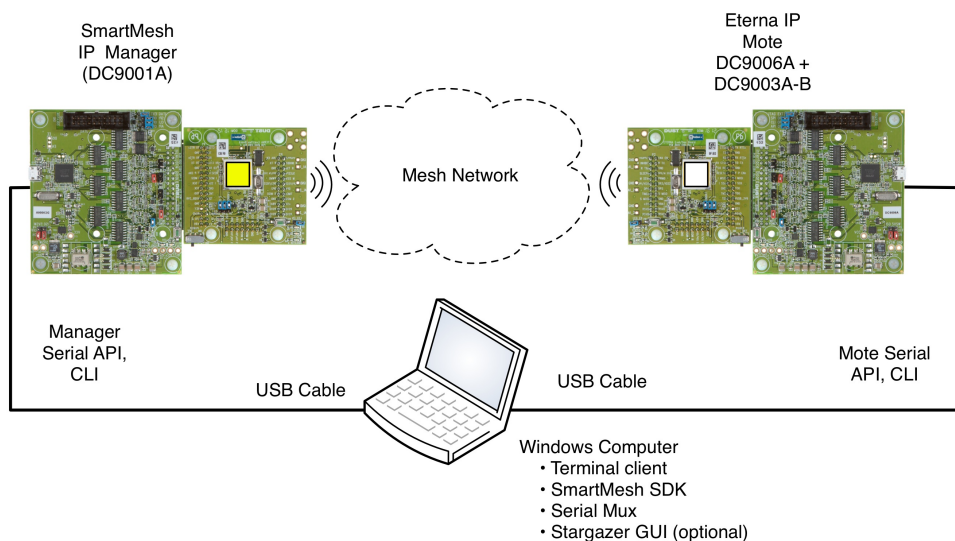

Figure 2: SmartMeshSDK Setup

3 Basic Steps

There are five steps to get the SDK running:

1. Form a wireless mesh network - just power everything on, and a network will form
2. Get Manager and Mote COM ports working on two USB ports on your PC
3. Connect to Manager CLI and Mote CLI
4. Install SDK Software
5. Connect to Manager API and Mote API

Once you've done these 5 steps you'll perform one exercise - you'll use the Mote API to make a Mote join the Manager.

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 in the lower right corner. Note the Blue LED. That indicates 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 simply give us ways to interact with that mesh network.

3.2 Step 2: Establish PC Connection to Manager and Mote

You will now connect the PC to both the Manager and Mote through their respective serial ports. If not already connected, connect the Manager to the DC9006 interface board using the side connector, if not already done. Next, plug the USB cable into one of your PC USB ports. Note which USB port you are using. You should consider this your Manager port for all future use with this PC.

Four virtual COM ports should automatically be added. You may have to install FTDI drivers if this doesn't happen automatically. Refer to the SmartMesh IP Tools Guide for detailed instructions.

Note the number of these four COM ports, they will be called something like "COM11", "COM12", "COM13" and "COM14". You will be using the third of the four for CLI and the fourth (highest chronological number) for the API connection. In this case, COM13 and COM14, correspondingly.

⚠ 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.

Repeat these steps for the Mote with a different PC USB port. Note which USB port you are using. Consider this as your Mote port for all future use with this PC. Another four COM ports will be added automatically, choose again the third of these new ports.

⚠ If you forget which USB ports you used for the Manager and Mote and get them reversed later, the COM ports to the two devices will also be flipped. If you plug either device into a third USB port, the drivers will be installed yet again and a third set of three COM ports will be assigned. You should consider canceling the driver install and switching ports until you match your original configuration.

3.3 Step 3: Connect to Manager CLI and Mote CLI

The CLI for a device allows you to type commands from a PC to interact with the Manager or Mote. Interacting with the CLI requires no software installation on the PC if you are running Windows XP, as it comes with the Hyperterminal client. If you are running Windows 7, you will need to download a terminal client (such as PuTTY).

3.3.1 Manager CLI

To connect to the Manager CLI open the terminal client. Connect to the third of the four COM ports that were added when you plugged in the Manager. The COM port settings are 9600-8-N-1. Hit enter a few times and you should see a > prompt character. If so, type 'help' to see a list of commands:

```
> help
help <command>
Commands:
mlog
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
```

⊘   Save this *"connection profile"* (in Hyperterm or PuTTY) so that you can use it in future.

There are many other things you can do with the Manager CLI. See the SmartMesh IP Manager CLI Guide for details. Many of the Application Notes involve interacting with the Manager CLI.

3.3.2 Mote CLI

Connecting to the Mote CLI is very similar to connecting to the Manager CLI above.

Connect the Mote to the second DC9006 interface board using the side connector. Since there are two DC9006 interface boards included in the kit, you shouldn't have to disconnect the Manager before connecting to the Mote.

Plug the USB cable into the PC USB port you marked before for Mote use if it is not already connected.

Open another terminal window and connect it to the third of the four COM ports that were added when you plugged in the Mote. The settings for this session are the same as for the Manager, 9600-8-N-1.

Hit enter a few times to get a ">" prompt, then type the `minfo` command to get info from the Mote:

```
> minfo
 ipmote ver 1.0.4 #0
 state: Oper
 mac: 00:17:0d:00:00:38:0e:cd
 moteid: 3
 netid: 1229
 blSwVer: 13
 UTC time: 1025717109:224250
 reset st: 100
```

Now we will "turn on" this Mote's API port. Motes in the kit were shipped to you in **master** mode. In **master** mode, the Motes join on their own, and the API is turned off. We will switch this Mote to **slave** mode. This mode turns on the API and the Mote will only join when told to do so by the sensor application. In our case today, the PC plays the role of the sensor application. You can get the current mode of the Mote with the command:

```
> get mode
```

To change the Mote to **slave** mode, type the following command:

```
> set mode slave
```

This stores the mode for use after the next boot. After setting the mode you must reset the mote via the reset button or by power cycling for the command to take effect.

Verify that you have set the Mote to **slave** by executing the `get mode` command again.

3.4 Step 4: Install SDK Software

While the CLI is designed for direct human interactions through text input, the API is a device interface for interacting with other processors. The SDK contains applications that perform various useful functions exercising the APIs through a GUI interface. The SDK is based on Python, but you do not need to install Python to use the pre-compiled applications.

Download the latest rev of SmartMeshSDK zip file.

Unzip the file and a folder by the same name will be created with 4 sub-folders: `api`, `doc`, `src`, and `win`. Each of these has sub-folders. The important one is `win`, where executable versions of the utilities are stored. The SmartMeshSDK folder can be moved to any convenient location on your computer. NO other installation is required.

# 3.5 Step 5: Join a Mote to the Manager with the API

Ensure that you are plugged into a Manager and a Mote. You have a terminal window connected to the CLI of the Manager and the Mote. We will now connect simultaneously to the API of each of these devices with the application APIExplorer.exe.

⚠ You can stay connected to the CLI terminal window while you connect to the API in a different application. This is possible because the CLI uses the third COM port for each device and the API uses the fourth COM port.

We now connect to the Mote API and make it join the Manager.

In the SmartMeshSDK directory, double click on the `/win/APIExplorer.exe` application. This opens the APIExplorer window. Tell the application you want to connect to a SmartMesh IP Mote by selecting **SmarMesh IP** as the network type and **mote** as device type:
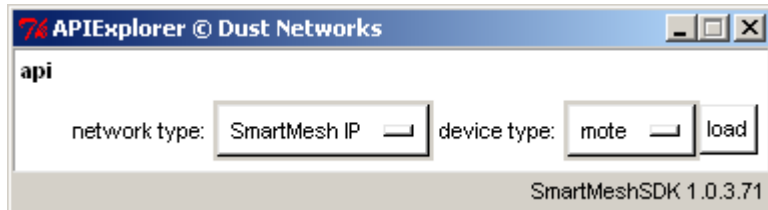
Click the **load** button – this loads the API for the IP Mote.

In the **connection** frame, enter the **port name** (e.g. COM25) of the API port - this is the 4th COM port added for the mote in step 2.
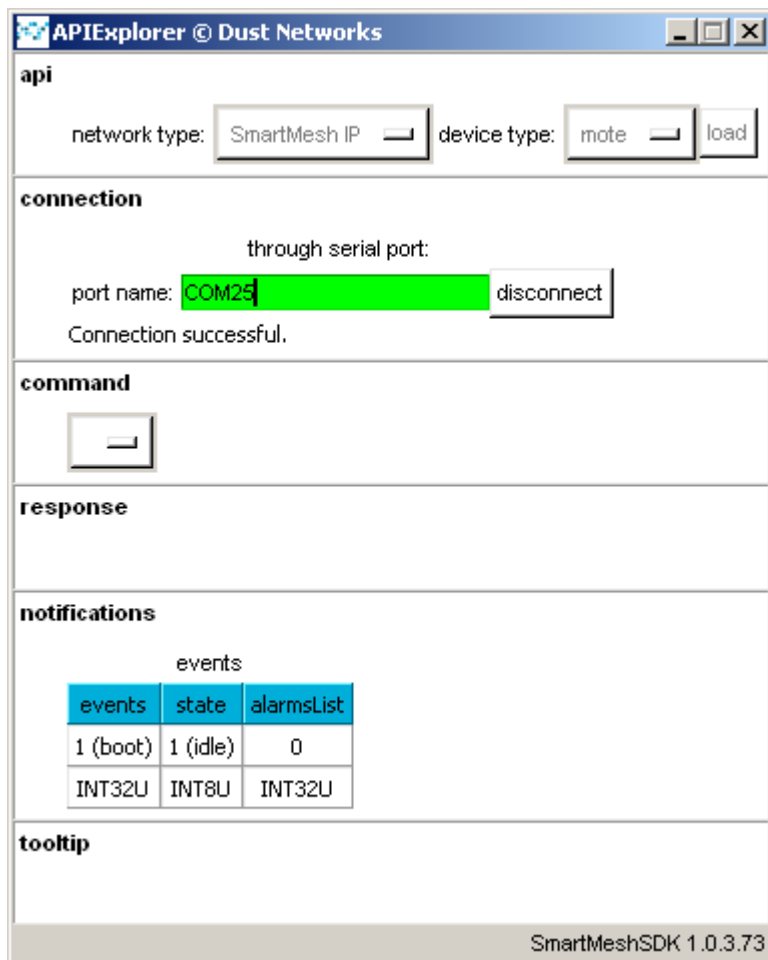
Click **connect**. The port field will turn green indicating the connection is successful.

> ⚠ This connection to the Mote won't work if the Mote is in **master** mode. APIExplorer will also fail to open the port if another application is trying to use the port.

To join the Mote to the Manager, go through these four steps**:**

1. Issue a *getParameter.networkId* command to verify that your SmartMesh IP Mote is configured with the correct Network ID (1229 by default)
2. Issue a *getParameter.status* command to verify that your SmartMesh IP Mote is in the Idle state
3. Issue a *join* command to tell the Mote to search and join the network
4. Repeat the *getParameter.status* command a few times over the next 30 seconds or so and you should see the Mote states proceed through to **Operational**

The Mote is now joined to the Manager.

Connecting to the Manager API is similar to how we connected to the Mote API above, although we select Manager for the device type. Launch a new instance of the APIExplorer, and follow the steps above until the connect field turns green to indicate a successful connection. Make sure to use the fourth (highest) COM port number that pertains to the Manager.

```
network type: SmartMesh IP
device type: manager
port name: your SmartMesh IP Manager's API COM port (the fourth added)
```

We will now *ping* one of the Motes in the network using a Manager API command.

1. Select subscribe and enter hex ffffffff in the **filter** box, and 00000000 in the **unackFilter** box. After clicking **send**, you have subscribed to all available notifications which you will see appear in the notifications frame.
2. The drop-down menu in the command frame lists all the commands defined in the SmartMesh IP Manager API Guide.
3. Issue a *getMoteConfig* command with a macAddress of 0000000000000000 and the next field set to true to obtain the macAddress of the first mote in the network.
4. Select the *pingMote* command and enter the MAC address of the Mote from the *getMoteConfig* command.
5. The Mote will respond with the round trip delay time, temperature and voltage.

4 Additional Tools

Once you have demonstrated basic network formation and the ability to interact with a Mote and Manager, you are ready to install the additional tools discussed in the SmartMesh IP Tools Guide. See the SmartMesh IP Tools Guide for installation instructions. These additional tools provide additional visualization options (Stargazer), support testing of APIs, allow for multiple simultaneous connections to the manager (Serial Mux), provide IPv6 routing (Low-power Border Router), and to generally aid in application development.

5 Steps in a Design

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

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 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 *joindutycyle*)
    - Use the *join* API to cause a mote to being searching for a network
    - Monitor the mote state to see when it is ready to accept data
    - Request services in order to publish data
    - The SmartMesh IP Mote API guide covers other commands to configure the mote
    - The SmartMesh IP Mote CLU 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).
- 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, and using a low-power border router for internet connectivity, among others

For Hardware design:

- Select a hardware platform - modularly certified or chip level?
- The hardware integration application notes and integration guides cover the important considerations for robust hardware development

6 Troubleshooting

1. When connected to a DC9006A board and a computer, the Mote and Manager may appear to be operating (as seen by the LEDs) in spite of the power switch being off. The 4 COM ports for each device will appear but you will not be able to communicate with them reliably. Make sure that the power switch on all boards is set to on to ensure proper operation.

2. 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.

3. If you are consistent with which USB port you plug the Mote and Manager into, the COM port assignments would should remain consistent. Please mark the physical USB ports on your machine for Manager and Mote usage respectively and then do not change or mix their usage.

4. Make sure that the baud rate is set to 9,600 for CLI and 115,200 for API. As an example , see Figure 2:

- COM 7-10 are for the Manager: COM10 is for API at baud rate of 115,200 and COM9 is for CLI at baud rate of 9,600
- COM 11-14 are for the Mote: COM14 is for API at baud rate of 115,200 and COM13 is for CLI at baud rate of 9,600
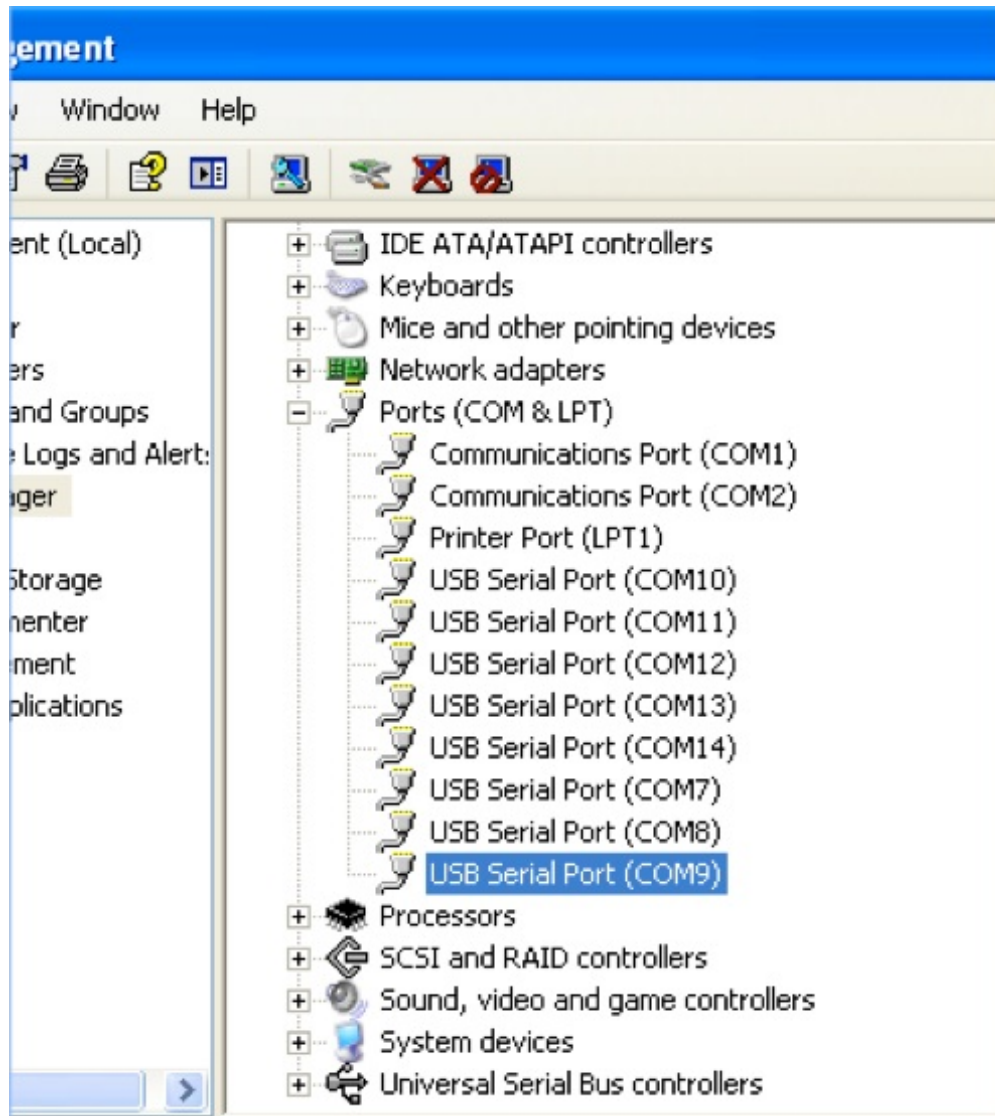
Figure 2: Device Manager Example

4. To enable and use the API/interface of the Mote, it must be programmed set to be in **slave** mode, using the CLI interface using the `set mode` command followed by a reset command. Once set in **slave** mode, you have to manually execute the *join* command on the Mote API interface, using APIExplorer in order to join the network. This setting is non-volatile.

5. If you program a Mote to be in **slave** mode, the status LEDs don't turn on which you may mistake for a dead board/battery. The blue indicator LED remains operational (if actuated via the LEDPing application or via the Stargazer GUI).

6. 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. By default, a Mote joins automatically when it boots if it is in **master** mode (default mode as shipped from the factory). In **slave** mode, a *join* command must be given.

8. The Manager *ping* command is a great way to get feedback from a Mote connected to a Manager, such as returning temperature, voltage and roundtrip time. This is a very useful command to show for demo purposes and can be easily executed by from either the CLI or the API interface of the Manager.

9. Keep the device Manager connected when you install the Mote so that another set of COM ports can be assigned.

6.1 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 the installing the serial drivers and example programs used in the Easy Start Guide and other tutorials.

User 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 applicaition, 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 program 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 Integration 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.