SNAP® CONNECT E10

Embedded SNAP Connect Appliance

User Guide

Wireless Technology to Control and Monitor Anything from Anywhere™

October, 2010
S/W Version 1.0
Doc Revision 1.0

Publications Part Number: 600034-01A
Revision: 1.0
Publication Date: October, 2010

Printed in the U.S.A.

Warranty
The manufacturer has made every effort to record accurate information and instructions in this document. However, no warranty is given with regard to this material. This denial of warranty includes, but is not limited to, the implied warranties of merchantability and fitness for a particular purpose. The manufacturer's liability for any errors in documentation is limited to the correction of known errors and to the provision of any further explanation (per customer request) necessary for clarification of product functionality.

500 Discovery Drive
Huntsville, Alabama 35806
877-982-7888

License governing any code samples presented in this document

Redistribution of code and use in source and binary forms, with or without modification, are permitted provided that it retains the copyright notice, operates only on SNAP® networks, and the paragraphs below in the documentation and/or other materials are provided with the distribution:

Neither the name of Synapse nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided “AS IS,” without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SYNAPSE AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SYNAPSE OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SYNAPSE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

FCC Information

These devices comply with part 15 of the FCC rules. Operation is subject to the following two conditions:

(1) These devices may not cause harmful interference, and

(2) These devices must accept any interference received, including interference that may cause harmful operation.

Please refer to the full FCC and IC statements found at the end of this user guide.

Disclaimers

Information contained in this document is provided in connection with Synapse products and services and is intended solely to assist its customers. Synapse reserves the right to make changes at any time and without notice. Synapse assumes no liability whatsoever for the contents of this Document or the redistribution as permitted by the foregoing Limited License. The terms and conditions governing the sale or use of Synapse products is expressly contained in the Synapse's Terms and Condition for the sale of those respective products.

Synapse retains the right to make changes to any product specification at any time without notice or liability to prior users, contributors, or recipients of redistributed versions of this document. Errata should be checked on any product referenced.

Synapse and the Synapse logo are registered trademarks of Synapse. All other trademarks are the property of their owners.

For further information on any Synapse product or service, contact us at:

Synapse Wireless, Inc.
500 Discovery Drive
Huntsville, Alabama 35806

256-852-7888
877-982-7888
256-852-7862 (fax)

www.synapse-wireless.com

Table of Contents

Chapter 1
Introduction

SNAP CONNECT E10 OVERVIEW

The SNAP® Connect E10 (shown in Figure 1-1) is a rugged and powerful embedded connectivity appliance built to interface directly with SNAP mesh networks.

Figure 1-1: The SNAP Connect E10

The E10 can fill several roles:

- It can provide connectivity between SNAP Wireless Nodes and TCP/IP networks such as the Internet
- It can participate as a SNAP Node itself (with SNAPpy scripting, RPCs)
- It can be the application platform for *your* control and monitoring application (Linux)

Easy-To-Use Technology

For many applications, the E10 requires little or no setup. The E10 ships pre-configured to:

- Use DHCP to learn its IP address.
- Automatically participate in any SNAP Network that is on the matching radio type, radio rate, radio channel, network ID, and encryption settings.
- Automatically accept incoming TCP/IP connections from other devices running SNAP Connect.

User configuration is only required if you want to change from these default settings, or if you want to connect to additional networks over TCP/IP.

Key Features

Key features of the E10 include:

- 32-bit RISC architecture
- 400 MHZ CPU
- 256 MB flash, expandable through external USB drives
- 64 MB RAM
- 10/100 Mb Ethernet port
- USB 2.0 (host) port
- micro-USB serial port
- Small form factor
- Rugged metal case
- Internal Synapse SNAP Engine with external antenna

Shipment Includes

Your E10 shipment includes the following:

- SNAP® Connect E10 unit
- Power adapter
- Micro USB cable
- Mounting ears

ABOUT THIS MANUAL

This manual provides information specifically regarding the SNAP Connect E10 and covers topics such as:

- Initial setup
- Port usage (e.g., using the micro-USB serial port, Ethernet port, and USB host port)
- Common applications
- Recovery processes that are specific to this product

The author assumes that:

- You have read and understood the *SNAP Primer* (or are otherwise familiar with the SNAP product line).
- You have installed the Portal software.
- You are familiar with the basics of discovering nodes, uploading SNAPpy scripts to them, and controlling and monitoring them from Portal.

Other Important Documentation

The E10 is typically running Synapse's SNAP Connect software internally. Be aware that SNAP Connect has its own dedicated user manual. You will likely need to refer to that manual as well because generic information on SNAP Connect is not repeated here.

All of our documentation is available for free download on our dedicated support forum. Please visit http://forums.synapse-wireless.com for more information.

Some of the related documents on the forum are listed below:

- *SNAP Primer* **(**600037-01A or later**)**
- *SNAP Connect Users Guide* (600022-01B or later)
- *SNAP Reference Manual* (600-0007F or later)
- *Portal Reference Manual* (60024-01B or later)
- *SNAP Hardware Technical Manual* (600-101.01D or later)
- *SNAP Sniffer Users Guide* (600026-01A or later)
- *SNAP Firmware Release Notes*
- *Portal Release Notes*

If you bought standalone modules (instead of buying a kit), you may also want to download the latest version of the EK2500 and EK2100 user guides.

When the Manuals Are Not Enough

The dedicated support forum offers more than just user documentation. In this forum, you can also view questions and answers posted by other users (as well as post your own questions for discussion). The forum also has examples and application notes available for you to download. You may also download the latest SNAP, Portal, E10, and SNAP Connect software from the forum.

Chapter 2
Network Application Examples

The following pages offer some high-level examples of how the E10 can be used. Of course, these are just a few examples. Many other network configurations and applications are possible.

EXAMPLE 1: COMBINING DISPARATE NETWORKS

By acting as a bridge between the wireless and Ethernet networks, E10s allow SNAP networks located in separate geographical locations to act as a single large network.

The SNAP subnets to be joined might be located in different parts of a building (as depicted in Figure 2-1), located across town (Figure 2-2 on page 12), or located across the country (Figure 2-3 on page 13).

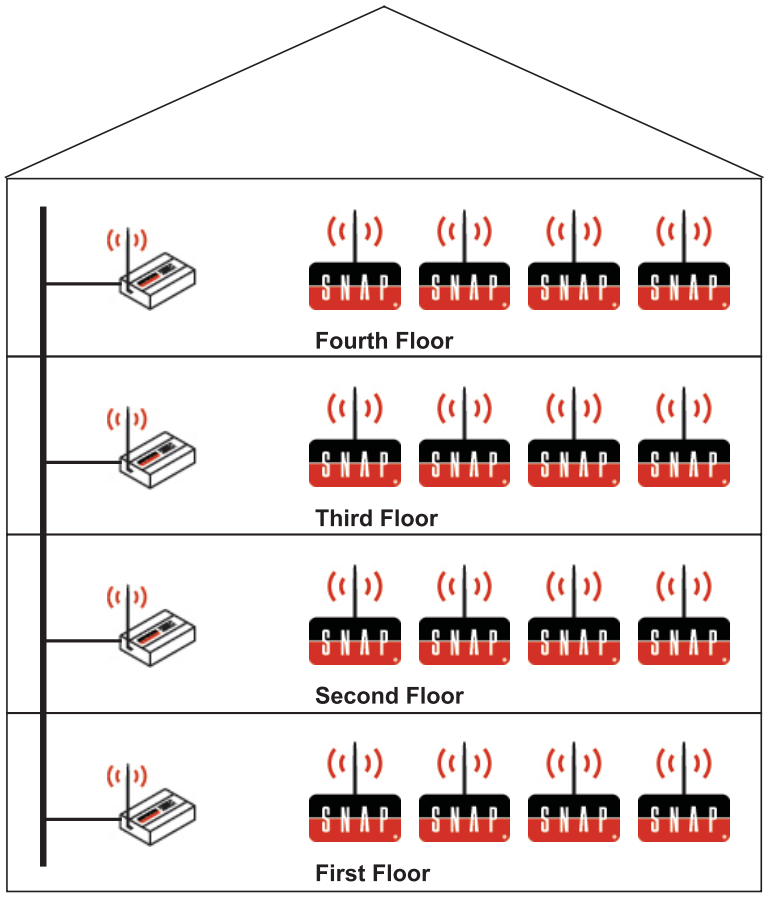


Figure 2-1: Joining Networks in a Building

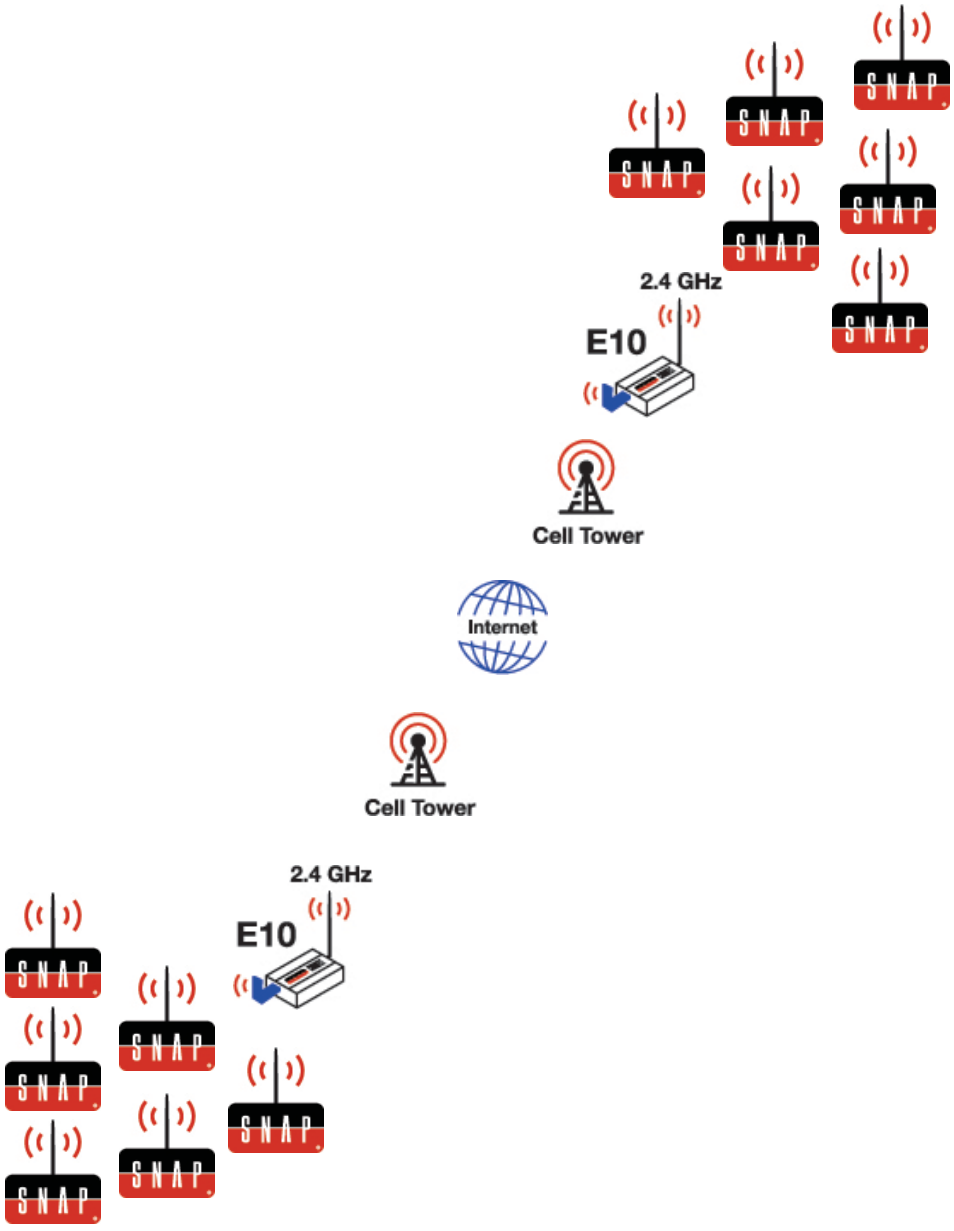


Figure 2-2: Joining Networks Across Town

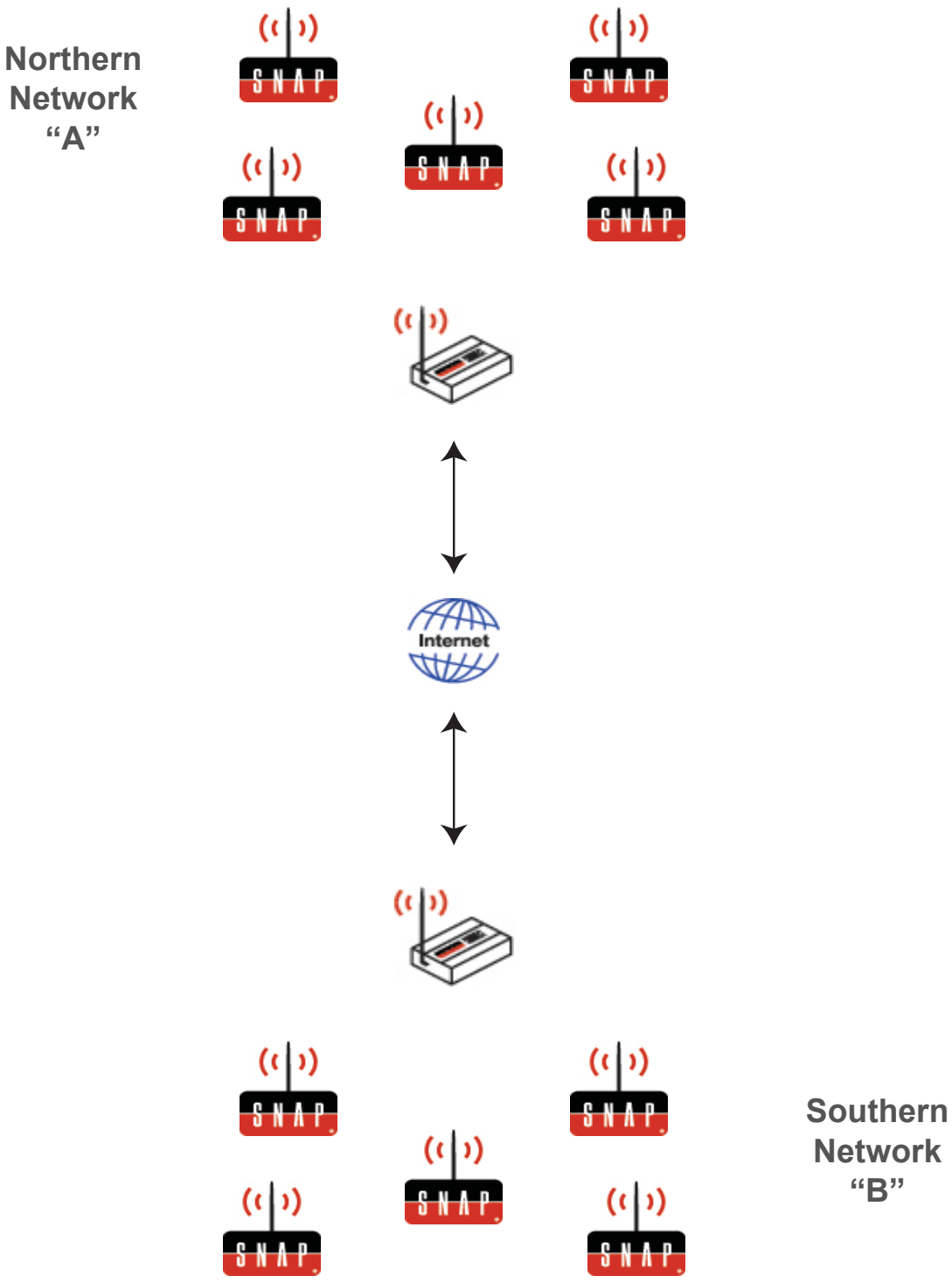


Figure 2-3: Joining Networks Across the Country

When communicating over TCP/IP networks, E10s are not restricted to talking only to other E10s.

Internally, E10s are running Synapse's SNAP Connect software, and can talk over TCP/IP to other computers that are also running the SNAP Connect software. See Figure 2-4.

For more on this topic, refer to the *SNAP Connect Users Guide*.

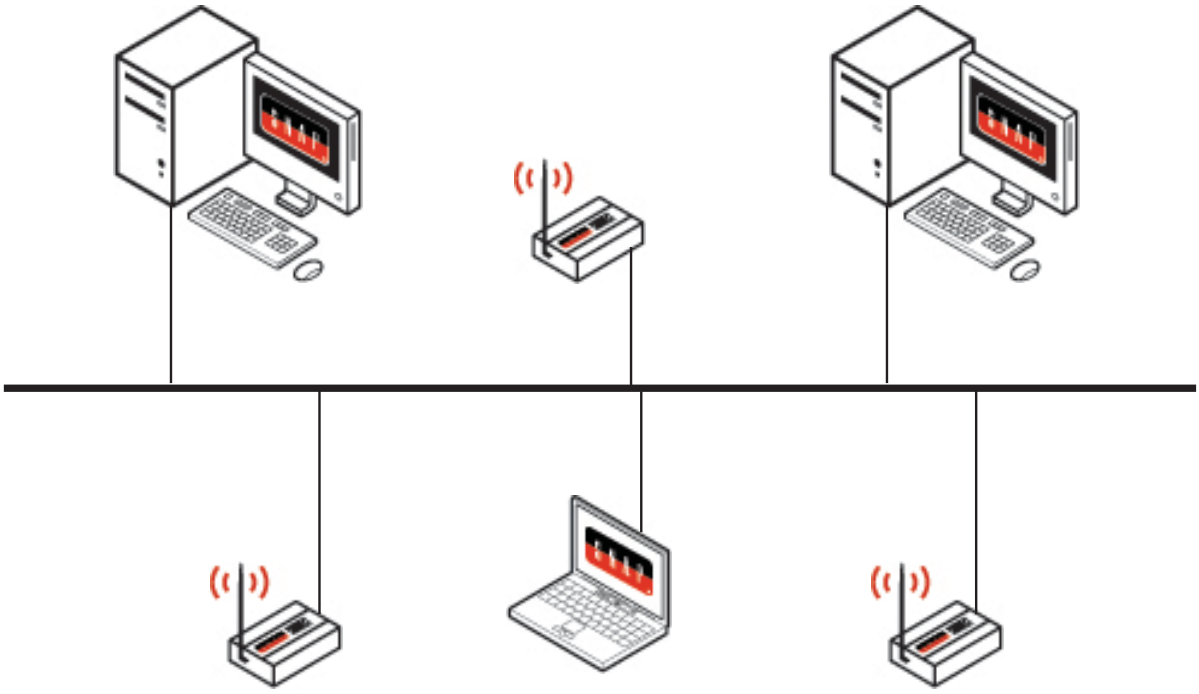


Figure 2-4: Communicating with other Devices Running SNAP Software

It is also possible to join different types of wireless networks by using E10s (as illustrated in Figure 2-5).

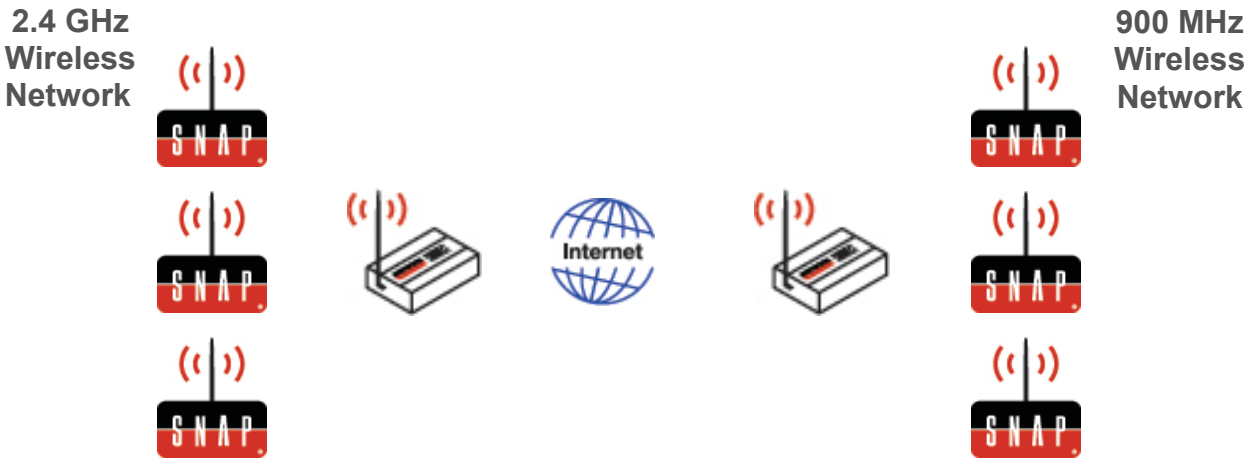


Figure 2-5: Joining Different Wireless Network Types

EXAMPLE 2: LOGGING DATA

Using its built-in USB host port, an E10 can provide a logging service to a field of SNAP nodes in which the nodes do not have their own removable USB drives. The logged data could then be analyzed locally (or taken offsite for more detailed analysis).

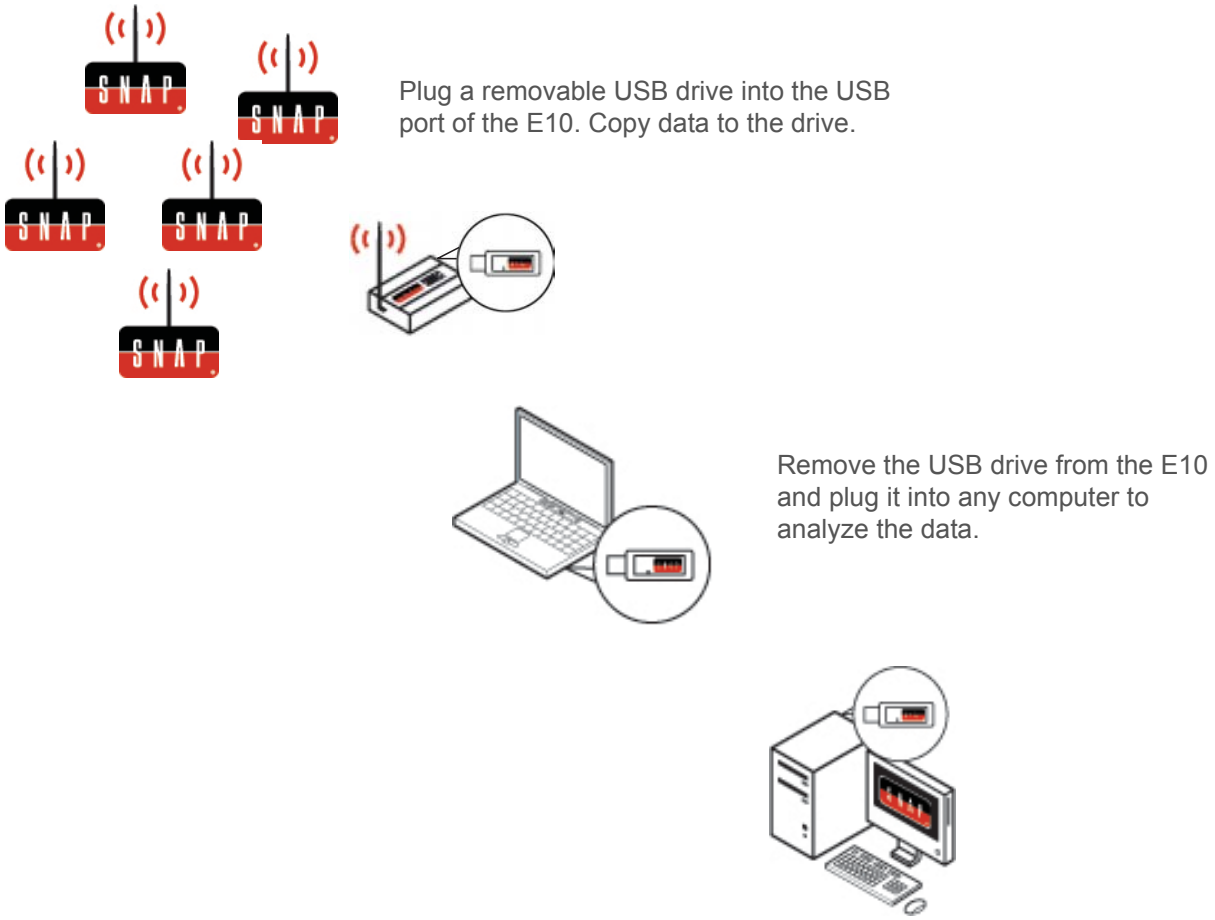


Figure 2-6: Providing Logging Services to Other Nodes

EXAMPLE 3: OFFLOADING HOST

E10s can be inserted as an extra “tier” into larger networks, offloading the master computer. Python script running on the E10s can be polling for data, performing data filtering (or other calculations), checking for alarm conditions, etc.

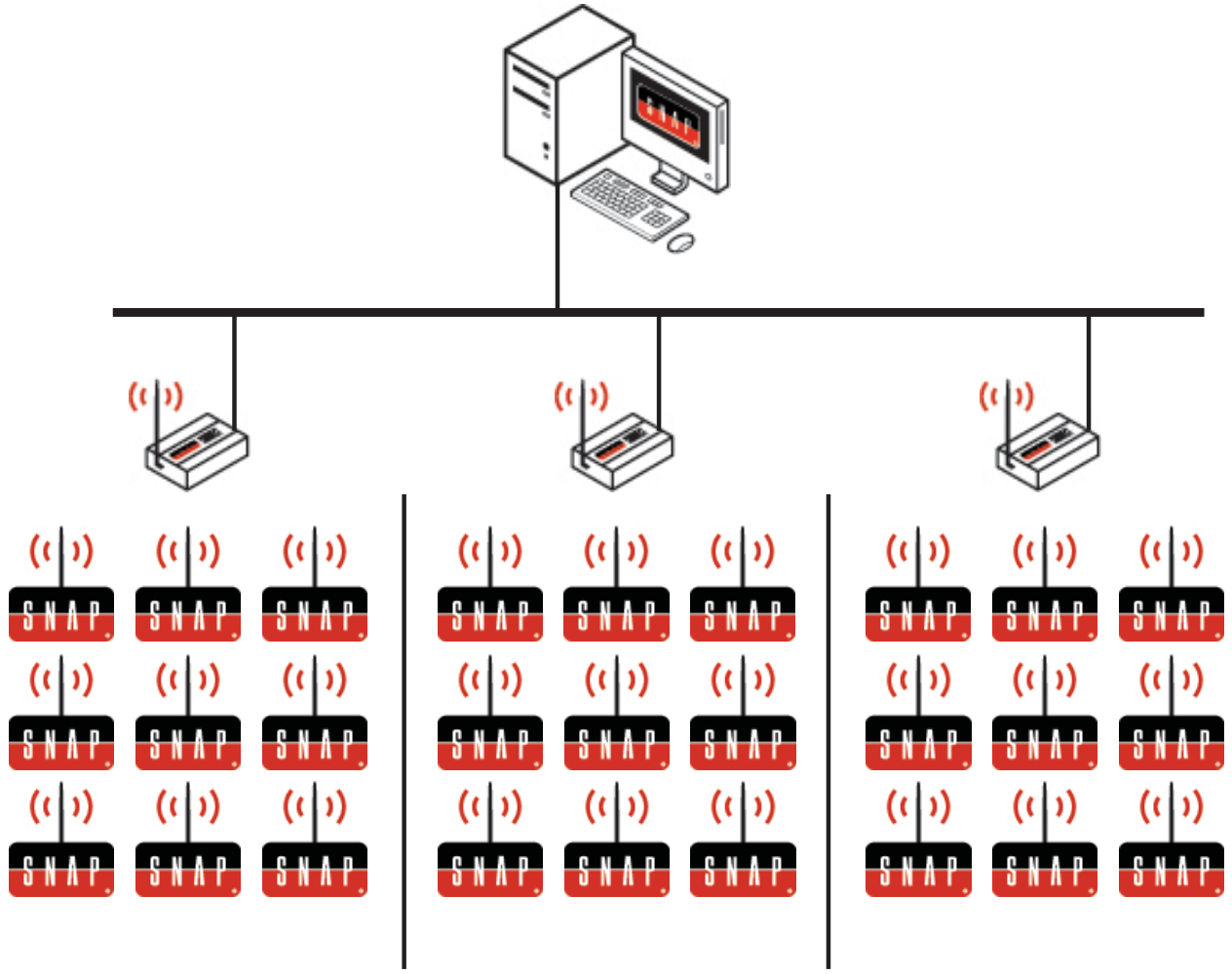


Figure 2-7: Offloading Host Computer by Providing Performance Checks

EXAMPLE 4: SNAP LIGHTING FROM ANYWHERE IN THE WORLD

When combined with Synapse's SNAP Cloud product, E10s support tunneling through firewalls, allowing full connectivity to wireless sensor networks from any web browser.

If your E10 is part of a SNAP Lighting Kit, there will be a Kit-ID and password on the bottom sticker of the E10. This is your login information for the SNAP Lighting website (at www.SNAPlighting.com).

Once logged in, you can control and monitor your lighting system from anywhere in the world with access to the Internet.

This simple-to-use interface is designed to let you remotely perform a number of tasks, such as:

- Turn on/off any of the four LED lights on each lighting board (included in kit).
- Dim any of the four LED lights on each board.
- Mix the red, green, blue and white light intensities.
- Read the on-board temperature sensor.
- Read the self-calibrating light sensor.

Note: *If your E10 is part of a SNAP Lighting Kit, please refer to the* ***SNAPLighting.com Kit User Guide*** *for more in-depth information on your product.*

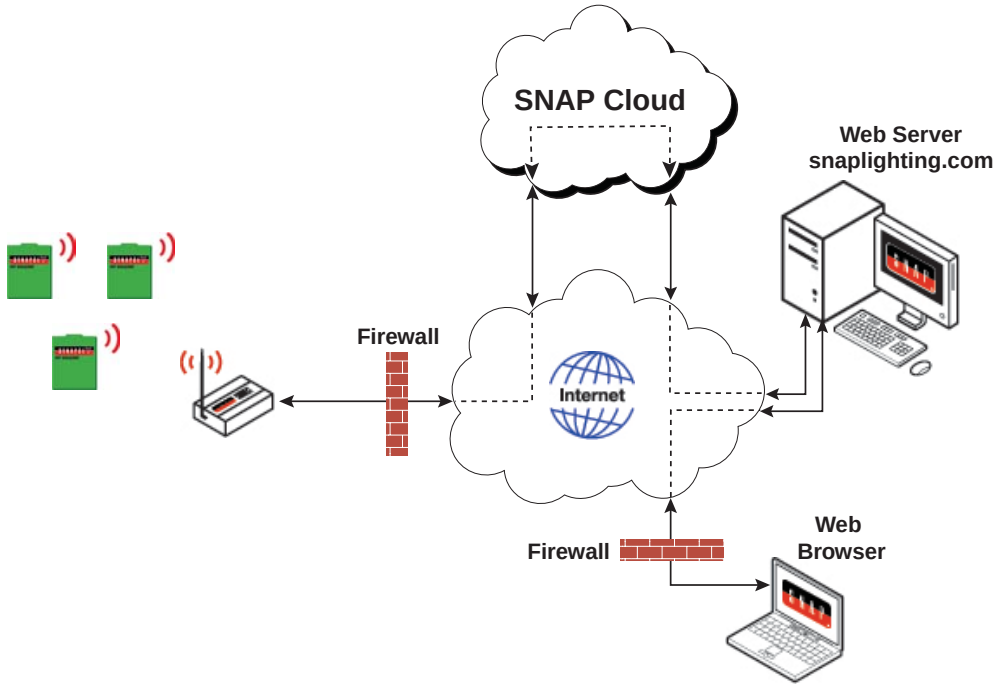


Figure 2-8: Controlling/Monitoring the System from Anywhere

Chapter 3
SNAP Connect E10 Overview

This chapter helps you become more familiar with the E10 by providing a more detailed product description as well as a visual tour of the product.

PRODUCT DESCRIPTION

There are three main things to know about the SNAP Connect E10. Depending on what you are trying to do with your E10, you may be more focused on one of these aspects over the others.

Table 3-1: SNAP Connect E10 Main Functionality

	Description
The E10 is a small Linux computer.	As a Linux computer, the E10 is capable of running custom applications that you create. This manual covers how to gain access to the Linux command prompt, but *does not* cover how to create such applications. If you need further assistance in this area, please contact the Synapse Custom Solutions Group. ***Note:*** *Advanced Linux users will be interested to know that the E10 is based on the AT91SAM9G20 processor, is running a variant of ARM-Linux, and comes with a version of Python pre-installed.*
The E10 is a SNAP Node with serial, Ethernet, and USB interfaces.	To act as a SNAP Node, the E10 runs Synapse's SNAP Connect software internally. This gives the E10 the ability to perform *mesh routing over the Internet*, letting SNAP Nodes in one networked location access a SNAP Network in another location. Also, you can extend the SNAP Connect software (using Python) to add more functions. These can then be invoked via RPC, much like the functions in SNAPpy scripts. For more information on this feature, refer to the *SNAP Connect User Guide*.
The E10 is equipped with an internal SNAP Engine, used to gain wireless access to other SNAP nodes.	This topic is covered in more detail later in this manual (see Accessing via SNAP Wireless on page 29). For now, the important thing to understand is that there are actually two computers contained inside the E10 housing. They communicate internally using Synapse's Packet Serial protocol over a dedicated serial link.

A VISUAL TOUR

This section helps you become more familiar with the SNAP E10's physical layout.

Top View

The top view of the E10 is shown in Figure 3-1. Notice there is an antenna connector on the right-hand side (as pointed out in the figure). Throughout this document, we will refer to the right-hand side as the "wireless side" of the unit, and the left-hand side as the "wired side" of the unit.

Figure 3-1: E10 Top View

Note: *The optional mounting ears are not shown here, but the E10 can be wall-mounted.*

Wireless Side View

The wireless side of the E10 is shown and described in Figure 3-2.

Note: *The behaviors of both the* **A** *LED and the* **MODE** *push-button are actually user-programmable. The default behaviors (as shipped from the factory) are described in Figure 3-2. For information on how to modify these defaults, refer to the chapter Common Procedures on page 30.*

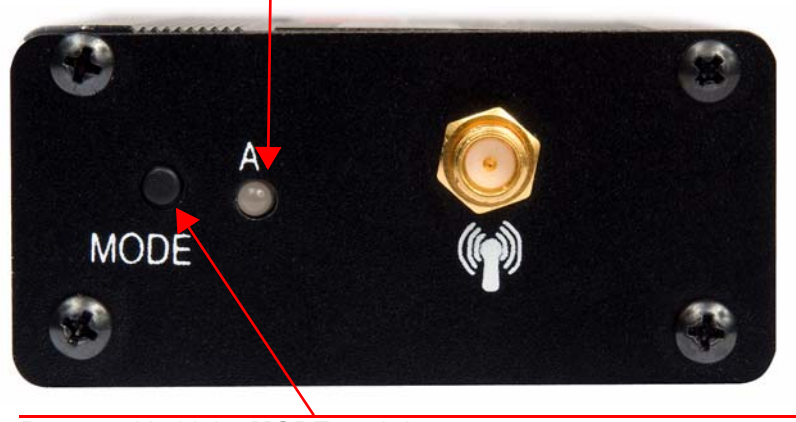
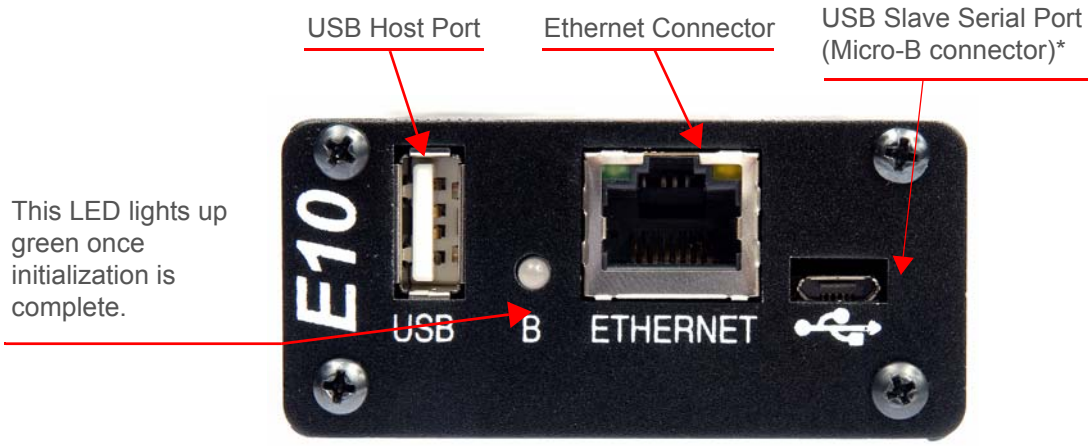


Figure 3-2: E10 Wireless Side View

Wired Side View

The wired side of the E10 is shown and described in Figure 3-3.



**The USB Slave Serial Port is the port from which the unit gets it power. See the section Power and Connectivity on page 24 for more information.*

Figure 3-3: E10 Wired Side View

Bottom View

The bottom side of the E10 is shown and described in Figure 3-4. Your label may look slightly different, but it will still show the MAC addresses assigned to the E10 Ethernet interface and the SNAP (RF) interface.

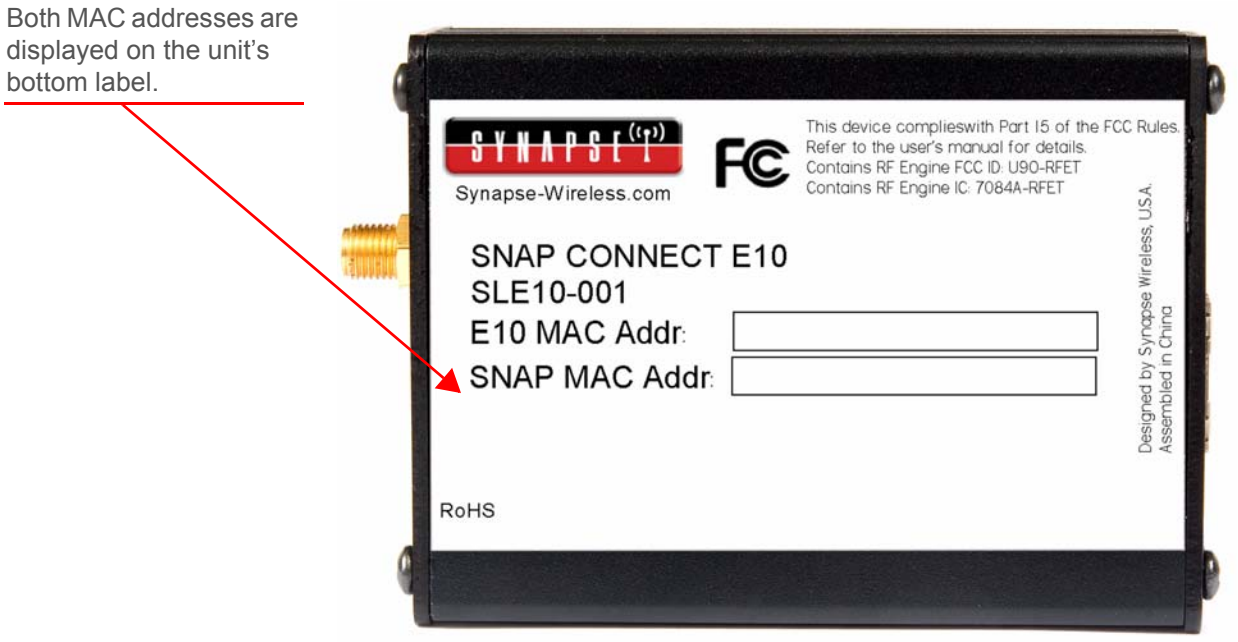


Figure 3-4: E10 Bottom View

Note: *If your E10 has a kit-ID and a password on the bottom label, it is actually part of a SNAP Lighting Kit (www.SNAPlighting.com). Please refer to the* ***SNAPLighting.com Kit User Guide*** *for information on your product.*

Chapter 4
Power and Connectivity

This chapter covers the E10's powering options and then, beginning on page 25, it explores the many available connectivity options.

POWERING THE E10

The E10 gets its power from the micro-USB serial port in the lower, right-hand corner of the wired side of the unit. Refer back to Figure 3-3 on page 22 to see the location of this connector.

To connect the unit to its power source, first plug the micro-USB cable (included in shipment) into the micro-USB serial port. Then, connect the other end of the cable to either of these two power source options:

- Option A: Any available powered host
- Option B: A wall-mount USB power adapter

Choosing Option A

If you choose to connect to an available host USB port for power, your computer may prompt you for the location of driver files. The driver files are necessary for accessing the E10 serially. (Most users prefer to have this access.)

Note: *If you only intend to use the computer as a power source (i.e., you do not need to access the E10 serially), simply click the* **Cancel** *button during the driver installation. The E10 will still be able to draw power through the USB connector.*

Installing the Drivers

The FTDI drivers you need in order to access the E10 serially are available from FTDI's website at http://www.ftdichip.com.

You will need the **VCP** drivers for the **FT2232D** chip. Websites often change, but at the time of this document's publication, the latest drivers could be found at:

http://www.ftdichip.com/Drivers/CDM/CDM20602.zip

Choosing Option B

If you choose Option B, you can use the power supply that comes with the E10. You may also be able to use any available USB power supply that provides at least 500 mA, but the E10 is only guaranteed to work with the supplied adapter.

ESTABLISHING CONNECTIVITY TO THE E10

The E10's Linux-based design provides a wide range of supported connectivity options. The remainder of this chapter explores the different ways you can connect to your unit.

See the following sections for more information:

- Accessing via Serial Port (below)
- Accessing via Ethernet Interface on page 26
- Accessing via SNAP Wireless on page 29

Accessing via Serial Port

The micro-USB cable that provides power to the E10 also provides USB-serial access to the "ttys0" UART of the E10.

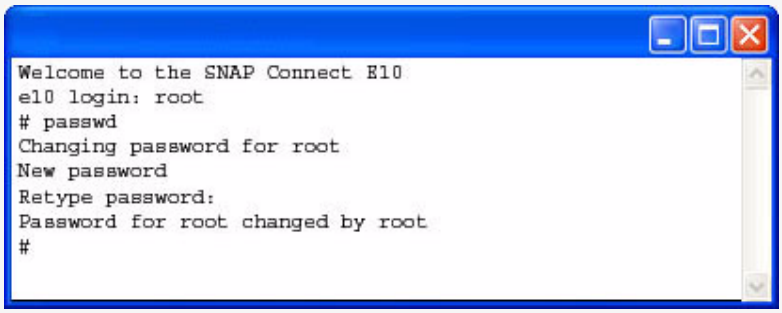
By default, this port is configured to be a Linux serial console. The UART settings are as follows:

- 115200 bps
- 8 data bits
- no parity bit,
- 1 stop bit

E10s ship from Synapse with a default username of **`root`** (and no password).

Note: *You should change your E10 password as soon as possible, using the* **`passwd`** *command.*

Here is a sample terminal session:



Accessing via Ethernet Interface

The E10 has a built-in Ethernet interface located on the wired side of the unit (refer back to Figure 3-3 on page 22 for location). When a live network cable is plugged into this interface, a green LED (built into the connector) lights up.

If the E10 detects any Ethernet traffic (even if it is addressed to a different device), a yellow LED (on the same connector) begins to blink.

If you are not getting the expected indicator lights, troubleshoot your network cabling first before trying to configure the E10.

Accessing the E10 via SSH

Determining IP Address

Once the E10 is successfully connected to the Ethernet, you may attempt to log in remotely. To accomplish this, you need to know the E10's IP address. This can be determined by first logging into the E10 using the micro-USB serial port, and entering the **`ifconfig eth0`** command.

For example:

```
# ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:1C:2C:FF:B5:EB
      inet addr:192.168.1.82  Bcast:192.168.1.255
Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:507 errors:35 dropped:0 overruns:0 frame:0
      TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:55342 (54.0 KiB)  TX bytes:4644 (4.5 KiB)
      Interrupt:21 Base address:0x4000
#
```

In the example above, the E10 has been assigned an IP address of **`192.168.1.82`** by the local DHCP server. Your E10's actual IP address will likely be different.

Assigning an IP Address Manually

If your site does not utilize a DHCP server, you must tell your E10 what IP address to use.

Edit the file named **`inittab`** in the E10's **`/etc`** directory and add a line like the following, putting it up near the top of the **`inittab`** file (and after the other, similar looking lines). For example:

```
null::sysinit:/sbin/ifconfig eth0 192.168.1.82 netmask 255.255.255.0 up
```

Of course, you must replace the IP address shown above with your desired IP address. If your network is not a Class C network, update the netmask as well.

Note: *Be sure to choose an IP address not already in use by some other device on your network!*

Reboot your E10 to allow the IP address change to take effect.

Enabling Remote Access

The OS running on the E10 only allows remote access with user accounts that have a password already set.

If you are still logging into your E10 as user **`root`** (with no password), then remote SSH logins will not be allowed. This is a security feature.

Either create a separate user account for remote access or, at a minimum, use the standard Linux **`passwd`** command to assign a password to the **`root`** account.

```
# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
#
```

Logging in via SSH

Once the E10 has been set up to allow remote logins, you will need an SSH client on some other PC. One example of a Windows Telnet/SSH program that works with the E10 is PuTTY, which can be found on the following website:

http://www.chiark.greenend.org.uk/~sgtatham/putty

When launched, PuTTY prompts you for the details of the system you plan to connect to (see Figure 4-1).

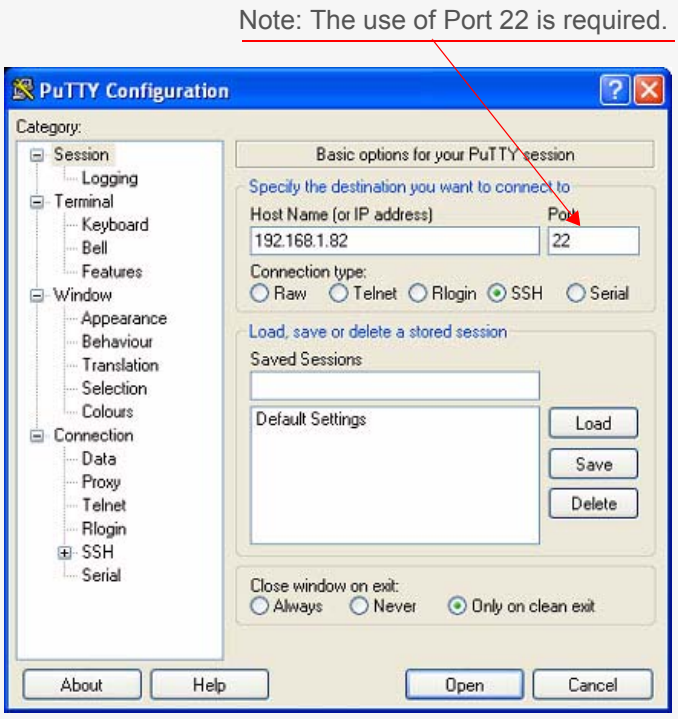


Figure 4-1: PuTTY Configuration Screen

Depending on your local network's security settings, you may get a warning dialog like the one shown in Figure 4-2:

Figure 4-2: Warning Message

When prompted by the E10, enter the username and password as usual.

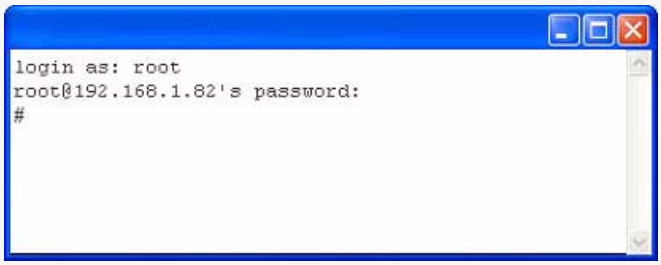


Figure 4-3: Entering Username and Password

Accessing TFTP servers from the E10

The E10 comes with a TFTP client that can be used to transfer files to/from the E10.

However, to use this client software you will have to have access to a TFTP server somewhere.

One example of a free Windows TFTP server that can be used with the E10 can be found on the following website:

http://tftpd32.jounin.net/

Note: *This free TFTP server can also act as a DHCP server (plus other features). Be sure to only enable the functionality you actually want.*

Once you download, install, and run this TFTP server (or something equivalent), you will be able to use your desktop PC as a source of files for the E10.

Use a command of **`tftp -help`** to see the command line options on the E10.

Accessing via SNAP Wireless

The E10 contains a full-blown, internal Synapse RF100 SNAP Engine.

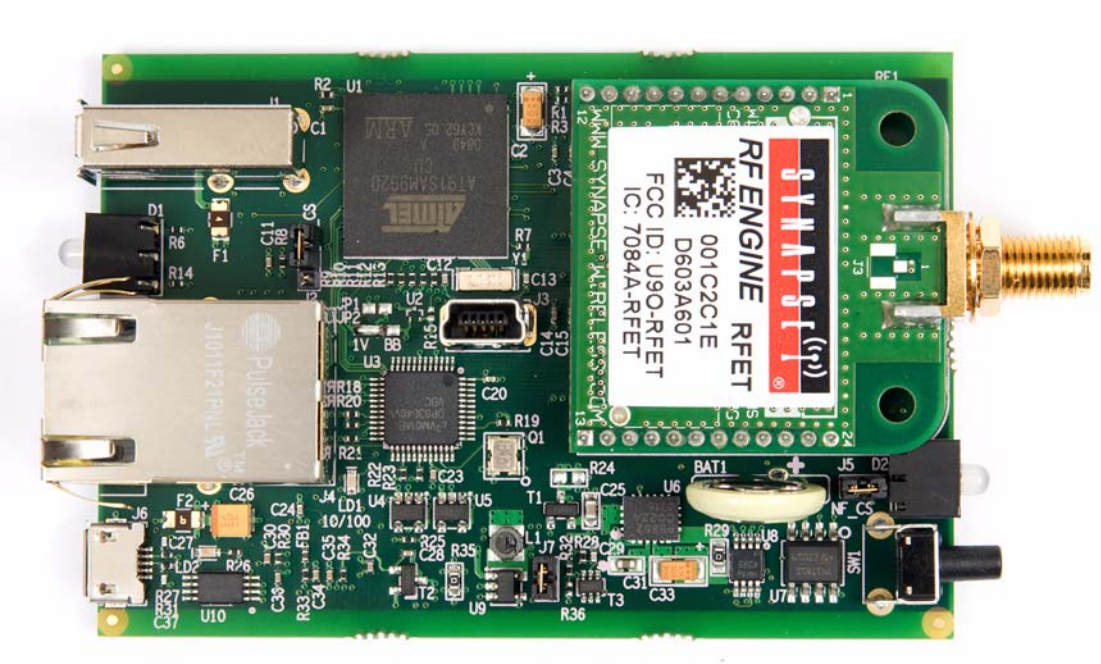


Figure 4-4: E10 Internal View

This SNAP Engine can be accessed over the air (for instance, via Portal) just like any other node in your SNAP network.

This SNAP engine runs standard SNAP firmware and can therefore be uploaded with a SNAPpy script.

The SNAP Connect software running on the E10's Linux processor uses this internal SNAP Engine as a bridge to reach the rest of your network (similar to how Portal uses its own bridge node).

You are not required to put a SNAPpy script into this SNAP Engine, but be aware that you can.

Note: *If you do put a script into the SNAP Engine, be sure to maintain the Packet Serial connection on the second UART (UART1 AKA DS_UART1) so that the Linux processor continues to have access to the SNAP Network.*

Chapter 5
Common Procedures

This section contains short example procedures for some common E10 functions.

Note: *For more information on these topics, please refer to the standard Linux documentation.*

Editing E10 Files

The E10 comes with the popular “vi” and “nano” editors pre-installed.

Viewing Other Available Commands

1. Look in directories **`bin`**, **`sbin`**, **`usr/bin`**, and **`usr/sbin.`**
2. The E10 uses the popular **`busybox`** multi-call binary.

To see what commands are built-in to BusyBox, try **`busybox -help`**.

Look on the next page for example output (too long to fit on this page).

```
# busybox --help
BusyBox v1.13.4 (2010-06-29 15:20:13 UTC) multi-call binary
Copyright (C) 1998-2008 Erik Andersen, Rob Landley, Denys Vlasenko
and others. Licensed under GPLv2.
See source distribution for full notice.

Usage: busybox [function] [arguments]...
   or: function [arguments]...
        BusyBox is a multi-call binary that combines many common Unix
        utilities into a single executable.  Most people will create a
        link to busybox for each function they wish to use and BusyBox
        will act like whatever it was invoked as!

Currently defined functions:
        [, [[, addgroup, adduser, ar, arping, ash, awk, basename,
        bunzip2, bzcat, cat, catv, chattr, chgrp, chmod, chown,
        chroot, chrt, chvt, cksum, clear, cmp, cp, cpio, crond,
        crontab, cut, date, dc, dd, deallocvt, delgroup, deluser,
        devmem, df, dhcprelay, diff, dirname, dmesg, dnsd, dos2unix,
        du, dumpkmap, dumpleases, echo, egrep, eject, env, ether-wake,
        expr, false, fdflush, fdformat, fgrep, find, fold, free,
        freeramdisk, fsck, fuser, getopt, getty, grep, gunzip, gzip,
        halt, hdparm, head, hexdump, hostid, hostname, hwclock,
        id, ifconfig, ifdown, ifup, inetd, init, insmod, install,
        ip, ipaddr, ipcrm, ipcs, iplink, iproute, iprule, iptunnel,
        kill, killall, killall5, klogd, last, length, less, linux32,
        linux64, linuxrc, ln, loadfont, loadkmap, logger, login,
        logname, losetup, ls, lsattr, lsmod, lzmacat, makedevs,
        md5sum, mdev, mesg, microcom, mkdir, mkfifo, mknod, mkswap,
        mktemp, modprobe, more, mount, mountpoint, mt, mv, nameif,
        netstat, nice, nohup, nslookup, od, openvt, passwd, patch,
        pidof, ping, pipe_progress, pivot_root, poweroff, printenv,
        printf, ps, pwd, rdate, readlink, readprofile, realpath,
        reboot, renice, reset, resize, rm, rmdir, rmmod, route,
        run-parts, runlevel, sed, seq, setarch, setconsole, setkeycodes,
        setlogcons, setsid, sh, sha1sum, sleep, sort, start-stop-daemon,
        strings, stty, su, sulogin, swapoff, swapon, switch_root,
        sync, sysctl, syslogd, tail, tar, tee, telnet, test, tftp,
        time, top, touch, tr, traceroute, true, tty, udhcpc, udhcpd,
        umount, uname, uniq, unix2dos, unlzma, unzip, uptime, usleep,
        uudecode, uuencode, vconfig, vi, vlock, watch, watchdog,
        wc, wget, which, who, whoami, xargs, yes, zcat
#
```

Controlling the A LED

The **A** LED on the wireless end of the E10 is controlled by the internal SNAP Engine using GPIO pins GPIO_0 and GPIO_1. This LED is actually a tri-color LED, controlled as follows:

- GPIO_0 controls the *green* of the LED (0 is off, 1 is green).
- GPIO_1 controls the *red* of the LED (0 is off, 1 is red).

If you turn on both GPIO_0 and GPIO_1, the LED will be *amber* (red + green)

As shipped, the internal SNAP Engine is loaded with a script that lights the LED up green. It also provides the following functions that can be invoked (via RPC) over the air:

- setLedAOff()
- setLedAGreen()
- setLedARed()
- setLedAYellow()

Controlling the B LED

The **B** LED on the wired end of the E10 is controlled by the Linux processor.

Try the following commands from the Linux command line:

```
echo "1" > /sys/class/leds/greenled/brightness
echo "0" > /sys/class/leds/greenled/brightness
echo "1" > /sys/class/leds/redled/brightness
echo "0" > /sys/class/leds/redled/brightness
```

Like the **A** LED, turning on both *red* and *green* for LED **B** will create an *amber* color.

Assuming you have not changed the **`UserMain.py`** file that comes preloaded on the E10, you can control the **B** LED over the air too:

- setLedBOff()
- setLedBGreen()
- setLedBRed()
- setLedBYellow()

To see how this works, look in the file **`UserMain.py`**.

Reading the MODE Button

The **MODE** button on the wireless end of the E10 is connected to the Linux processor.

One way it can be monitored is via **`:/dev/event0`**.

Here is example hexdump output:

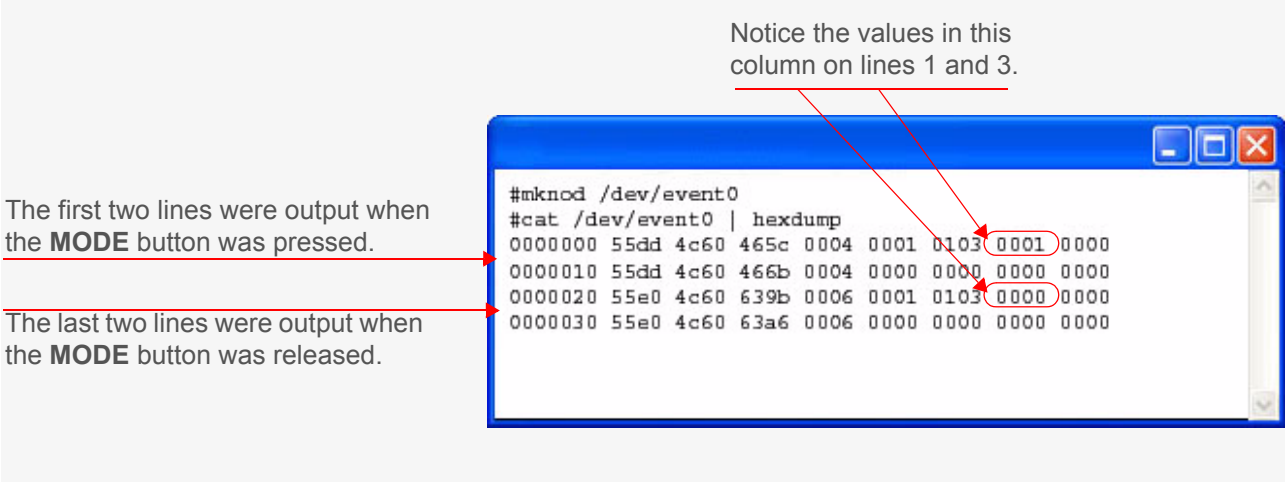


Figure 5-1: Mode Button Output

A simpler way to read the **MODE** button is to use the **`gpio9260`** program that comes with the E10:

`/usr/bin/gpio9260 ?PB10`

This program will return **`0`** if the button is pressed, and will return a non-zero value if the button is not pressed.

Assuming you have not changed the default **`UserMain.py`** file, the button can also be read via SNAP RPC by calling function **`readButton()`**.

Using an External USB FLASH Drive

The exact commands needed to "mount" different USB FLASH drives vary between drives, but here are some examples:

Example 1:

```
>mdev -s
>mount /dev/sda /mnt
```

Example 2:

```
>mdev -s
>mount -t vfat /dev/sda1 /mnt
```

After executing the correct command, you should be able to find your USB FLASH drive files under the **/mnt** directory.

Repairing DOS and Windows Text Files

You may sometimes need to clean up text files that came from a DOS or Windows system. These two systems use a *line ending* convention that differs from Linux systems.

If you transfer a file over to your E10 (for example, some Python source code) and it seems to have a lot of funny characters (**^M**) at the end of some or all lines, try entering this command:

```
>dos2unix <file to be repaired>
```

Creating Custom Python Software

This procedure shows you how to create your own custom Python software. The E10 comes with Python 2.6.5 already installed. You can use Python in interactive mode:

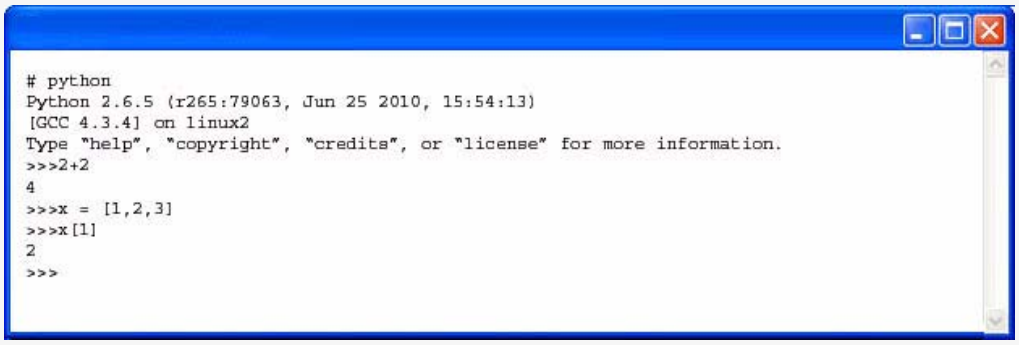


Figure 5-2: Python in Interactive Mode

You can also write and execute Python programs:

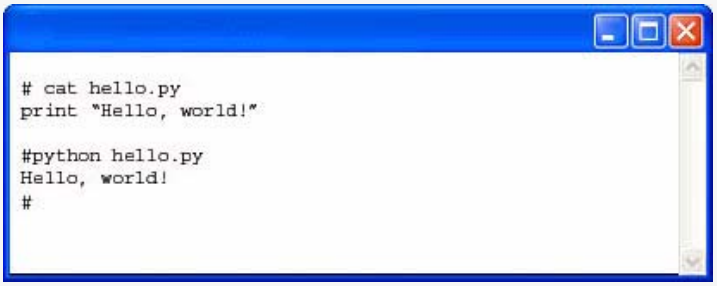


Figure 5-3: Simple Python Program

Note: *Advanced users — Most of the standard Python modules are available for use in your code, with the following exceptions:*

_bsddb	**_tkinter**	**bsddb185**	**dbm**	**dl**
Gdbm	**imageop**	**sunaudiodev**	**nis**	

Making Custom Software Run Automatically at Bootup

Choose one of the following methods to cause your custom software to automatically run at system startup:

Method 1: Put your own custom Python code in the **UserMain.py** file.

The E10 comes pre-configured to automatically launch Python file **/root/UserMain.py** during system startup, so the easiest thing to do is just modify this source file to do what you want.

You can see *how* this Python program gets launched by looking in the **/etc/init.d/S999snap** file.

Method 2: Add a command like the following to your **/etc/inittab** file:

```
null::sysinit:<path to program> <arguments to program>
```

For example:

```
null::sysinit:/user/bin/python /root/Main.py
```

Shutting Down UserMain.py Manually

If you need to halt **UserMain.py** (for example, you are going to make more changes to it), use the following command:

```
/etc/init.d/S999snap stop
```

Starting UserMain.py Manually

To manually restart **UserMain.py** (without rebooting), you can use the following command:

```
/etc/init.d/S999snap start
```

Restarting UserMain.py Manually

To stop and then restart **UserMain.py**, you could use the following commands:

```
/etc/init.d/S999snap stop
```

```
/etc/init.d/S999snap start
```

However, an easier way would be to do the following single command:

```
/etc/init.d/S999snap restart
```

Restoring the Original Behavior of Your E10

If you have made customizations to your **UserMain.py** and want the original functionality back, be aware that file **SynapseMain.py** (located in the same directory) is a backup copy of **UserMain.py**. If you overwrite **UserMain.py** with **SynapseMain.py**, the original functionality will be restored (and all your customizations will be lost).

Note: *If you have modified or deleted the* **SynapseMain.py** *file, this procedure does not apply.*

Temporarily Restoring the Original Behavior of Your E10

The above method procedure permanently erases your changes. If you just want to run the **SynapseMain.py** code briefly, reboot the E10 and then *hold down the MODE button while the unit is booting up. Python file* **SynapseMain.py** *will be launched instead of* **UserMain.py**.

Note: *If you have modified or deleted the* **SynapseMain.py** *file, this procedure does not apply.*

Chapter 6
Upgrade Procedures

The E10 ships from the factory pre-loaded with the latest versions of both the operating system (Linux based) and Synapse SNAP Connect software (i.e., the latest versions at the time that E10 was built).

From time to time, Synapse releases updated versions of both the underlying operating system and the SNAP Connect software.

Note: *To perform these upgrades you must be logged into your E10 with root-level privileges.*

FILE LEVEL UPGRADES

Use the instructions in this section to upgrade to a new version of SNAP Connect (or as a template for loading/upgrading your own custom application software).

Transferring the Latest SNAP Connect Software to your E10

This process can be done via tftp or using an external USB drive.

Examples of recent SNAP Connect **`tar`** files include:

```
connect-e10-20100827.tar
tftp -g -r <filename> <tftp server IP address>
```

or

```
cp /mnt/<filename>
```

As on most *nix systems, **`tar`** (archive) files can be extracted by a command such as:

```
tar -xvf <filename>
```

OS LEVEL UPGRADES

The E10 ships from the factory pre-loaded with the latest version of the operating system software at the time that E10 was built.

From time to time, Synapse releases updated versions of the E10 operating system.

This section describes how to upgrade your E10 to a newer version of the operating system.

Note: *This upgrade procedure may permanently erase any and all user content (programs, files, or data) saved on your E10. If necessary, back these files up via tftp or using an external USB drive.*

OS Upgrade Steps

Step 1: Transfer the latest E10 software image to your device.

This can be done via tftp or using an external USB drive.

Examples of recent E10 image filenames include:

```
connect-e10-rootfs.arm-20100614.jffs2
connect-e10-rootfs.arm-20100630.jffs2
tftp -g -r <filename> <tftp server IP address>
```

or

```
cp /mnt/<filename>
```

Step 2: Erase the existing image from the E10.

Use the command:

```
flash_eraseall -j /dev/mtd1
```

Step 3: Program the newly transferred image into the E10.

The command is:

```
nandwrite -a /dev/mtd1 <filename>
```

Step 4: Reboot the E10 into the U-Boot Manager.

The command is:

```
reboot now
```

Watch the console output for the message **`Hit any key to stop autoboot:`**. When you see this message, hit a key (like the space bar) to divert the E10 into its built-in U-Boot firmware manager.

Note: *If you take too long to hit a key, the system simply boots up into full-up Linux. If this happens, just log in as normal and enter the* **`reboot now`** *command again.*

Instead of a normal Linux prompt you should see a **`U-Boot`** prompt that looks like this:

```
U-Boot>
```

Step 5: Tell the E10 to use the newly programmed-in firmware image.

Do a **printenv** command. You should see output like the following:

```
U-Boot> printenv
bootargs=console=ttyS0,115200 root=/dev/mtdblock1
mtdparts=atmel_nand:16M(kernel)ro,120M(rootfs),-(other) rw
rootfstype=jffs2
bootdelay=3
baudrate=115200
autostart=yes
ethact=macb0
bootcmd=nboot 0x21000000 0 a0000
ethaddr=00:1c:2c:ff:b5:eb
stdin=serial
stdout=serial
stderr=serial

Environment size: 281/131067 bytes
U-Boot>
```

Enter the following command and press the **<Enter>** key:

```
set bootcmd nboot 0x21000000 0 a0000
```

Step 6: Correct the E10's MAC address if needed.

Look at the output of the **printenv** command and see what the E10's MAC address is currently set to.

In the output of the **printenv** command, this field is labeled **ethaddr**.

If it matches the label on the bottom of the E10, no action is needed.

If instead it shows something like **00:1c:2c:ff:ff:ff**, then the correct Ethernet MAC address still needs to be set.

Note: *Type carefully during this step! The E10 only allows the MAC address to be changed ONCE. If you make a mistake and set your Ethernet address incorrectly, refer to the chapter* Recovery Procedure *on page 40 for help.*

If your E10 does in fact need its MAC address set, use a command of the form:

```
set ethaddr 00:1c:2c:xx:yy:zz
```

Where **xx:yy:zz** should be replaced with the correct three hexadecimal bytes from the label on the bottom of your E10.

Step 7: Save your changes.

Use the command **saveenv.**

Step 8: Exit U-Boot and startup Linux.

Use the U-Boot command **reset.**

Chapter 7
Recovery Procedure

If you "brick" your E10 (i.e., if you make a programming or configuration mistake such that you are no longer able to log into your E10 at all), then you can use the recovery procedure described in this section to restore your E10 to its original factory configuration.

Note: *This recovery procedure permanently erases any and all user content (programs, files, or data) that may have been on your E10 prior to performing the recovery procedure.*

Basically, this procedure uses a special factory programming connector located inside the E10 to completely erase and reprogram the Linux processor, restoring the original operating software.

HARDWARE/SOFTWARE REQUIREMENTS

This section covers the hardware and software tools needed in order to perform the recovery procedure.

Required Hardware

- Screwdriver with the correct tip (at the time of this document's publication, E10 units were shipping with "Torx" style screws)
- USB cable (with *Type A* connector on one end and *Mini A* connector on the other end)

Note: *This is different from the Micro-A cable that comes with the E10 (you will need that cable, also).*

For more information about USB connectors, refer to the website:

http://en.wikipedia.org/wiki/USB_connector#Connector_types

Optional Hardware

- Needle-nose pliers to remove/reinstall a jumper (unless you have very small fingers)
- Magnifying glass

Required Software

All of the files and software needed for the recovery process can be found in the file **`E10_Recovery.zip`**, available on the *Synapse User Forum* at:

http://forums.synapse-wireless.com.

Required software program. The E10 is based on an ATMEL chipset and an ATMEL tool is used to do the initial programming steps. This program runs under Windows and is named **SAM-BA CDC**. This tool is included in the **E10_Recovery.zip** file.

Required Files. The following files are also necessary for this process and can be found in the **E10_Recovery.zip** file as well.

- **bootstrap.bin**
- **u-boot-20000.bin**
- **uImage-a0000.bin**
- **uImage-400000.bin**

RECOVERY STEPS

Step 1: Remove power from the unit.

Disconnect the micro-USB cable. The same micro-USB cable that provides serial access to the E10 also provides its power. If instead of a USB serial cable you are using a USB *power-only* cable, unplug that cable instead.

Step 2: Open your E10.

In order to perform this recovery procedure, you must open up your E10. To do so, remove all screws from both ends of the unit. The main circuit board can then slide out of the wireless end of the unit (you can go the other way if you first remove the external antenna).

Step 3: Remove Jumper J5.

To access the reprogramming mode, you must remove **Jumper J5**. The jumper (which is labeled on the silk-screen) is located next to the **MODE A** LED. Figure 7-1 points out the jumper's location.

Note: *You may find it necessary to remove the SNAP Engine first for easier access to the jumper.*

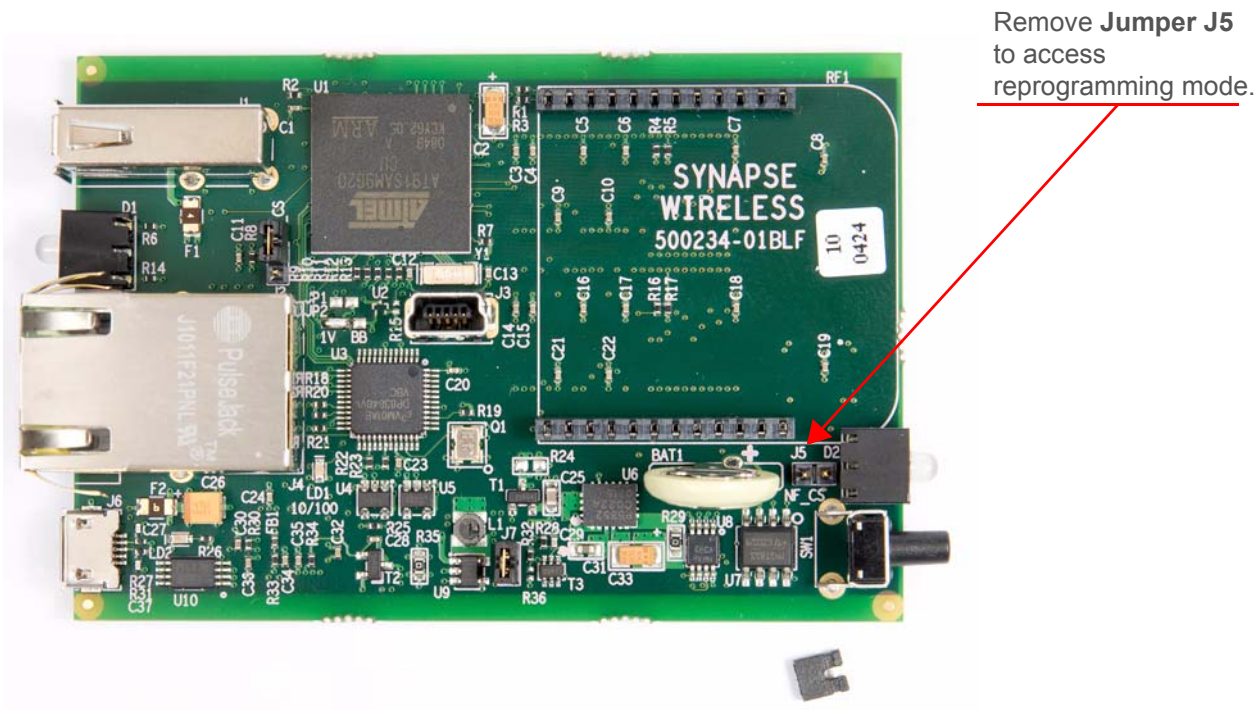


Figure 7-1: Location of Jumper J5

Step 4: Power up the E10.

With the jumper removed, re-apply power (i.e., reconnect the micro-USB cable) and wait three seconds.

Step 5: Reinstall Jumper J5.

While leaving the E10 powered up, carefully put **Jumper J5** back on.

Step 6: Connect the mini-USB cable.

Connect your Windows-based PC to the connector *in the center of the E10 circuit board* using the mini-USB cable (i.e., *not* the micro-USB cable). Refer to Figure 7-2 to see what the E10 looks like once both cables are connected.

Note: *REMINDER - This is a different sized connector than the one used on the outside of the E10.*

Figure 7-2: E10 with Both Cables Attached

Once this connection is made, your PC may prompt you for the location of a driver file. Direct the driver installer to the **`drv`** directory of the files you extracted from the **`E10_Recovery.zip`** file.

Step 7: Launch the SAM-BA CDC program.

The **SAM-BA CDC** program is required for recovering your E10.

Note: *As mentioned previously on page 40, this program is in the* **`E10_Recovery.zip`** *file which you should have already downloaded from the Synapse User Forum.*

Once you launch the **SAM-BA CDC** program, a dialog like the one shown in Figure 7-3 may appear. If so, just click the **RUN** button.

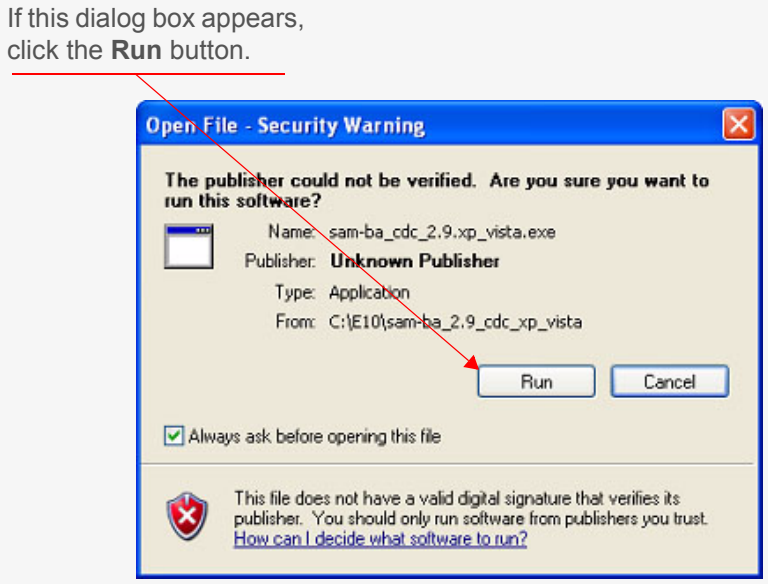


Figure 7-3: Running the SAM-BA CDC Program

Step 8: Make the connection.

After selecting **Run**, a dialog box like the one shown in Figure 7-4 should appear. From this dialog, it may be necessary to select the correct board type. (The value **at91sam9g20-ek** corresponds to the ATMEL CPU chip used inside the E10).

With the correct board type selected, hit the **CONNECT** button.

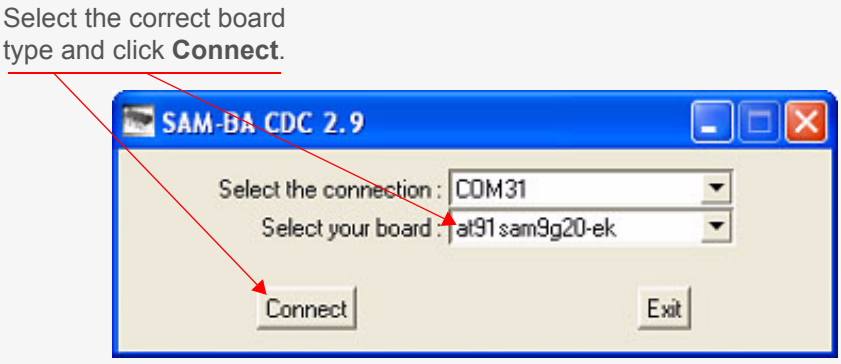


Figure 7-4: Making the Connection

Step 9: Enable the NandFLASH. A few seconds after clicking **Connect**, the main GUI displays. The lower half of the GUI is made up of a series of tabbed dialogs. Click the **NandFlash** tab (if it is not already activated), choose **Enable NandFlash** from the **Scripts** drop-down menu, and click **Execute**. These steps are illustrated in Figure 7-5.

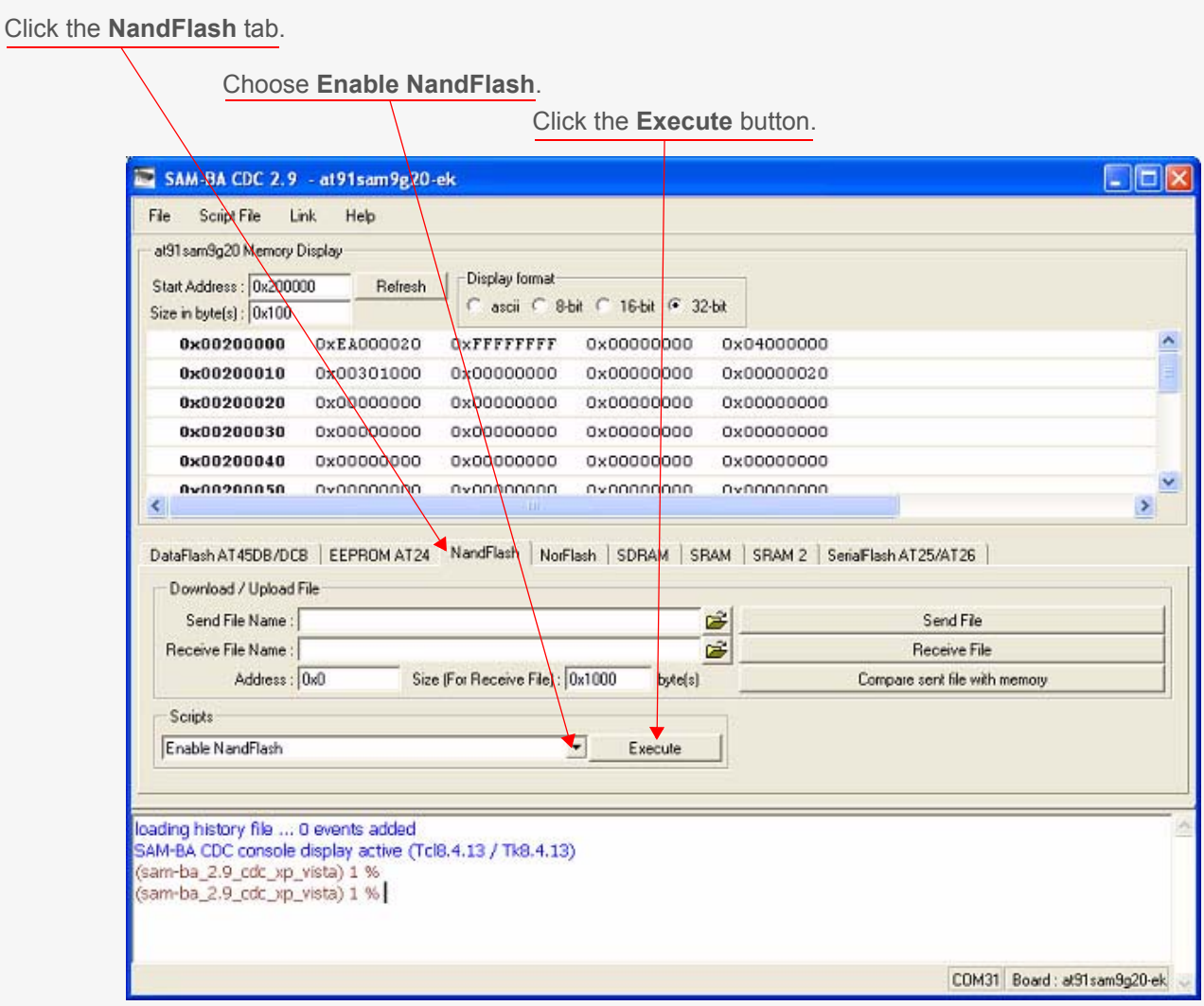


Figure 7-5: Enabling Nandflash

Step 10: Erase the E10.

Now the GUI should look similar to Figure 7-6. In the **Scripts** pull-down, choose **EraseAll** and click the **Execute** button.

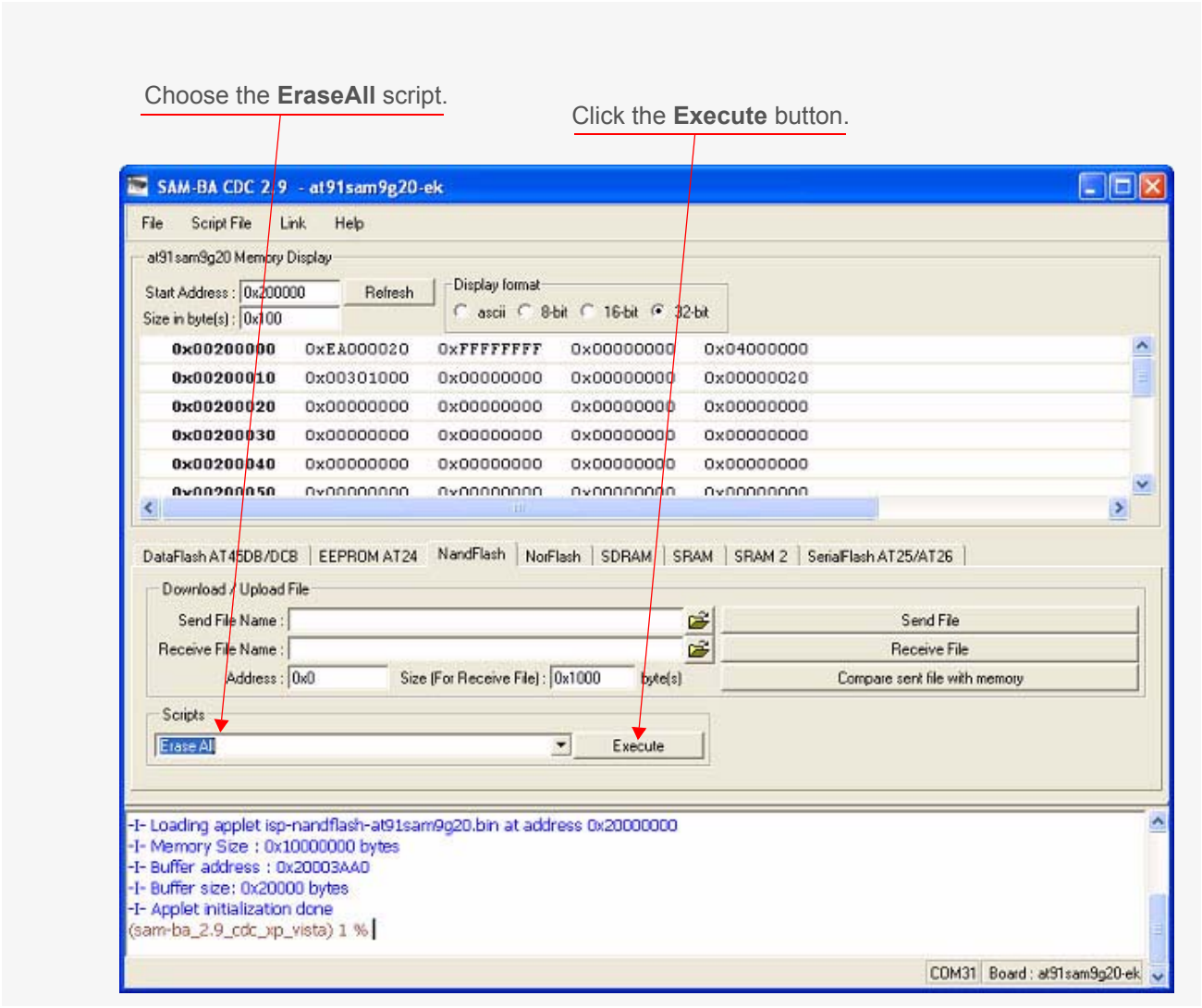


Figure 7-6: Choosing the EraseAll Script

Step 11: Send the Boot File to the E10.

The GUI should now look similar to Figure 7-7. In the **Scripts** pull-down, choose **Send Boot File** and click the **Execute** button.

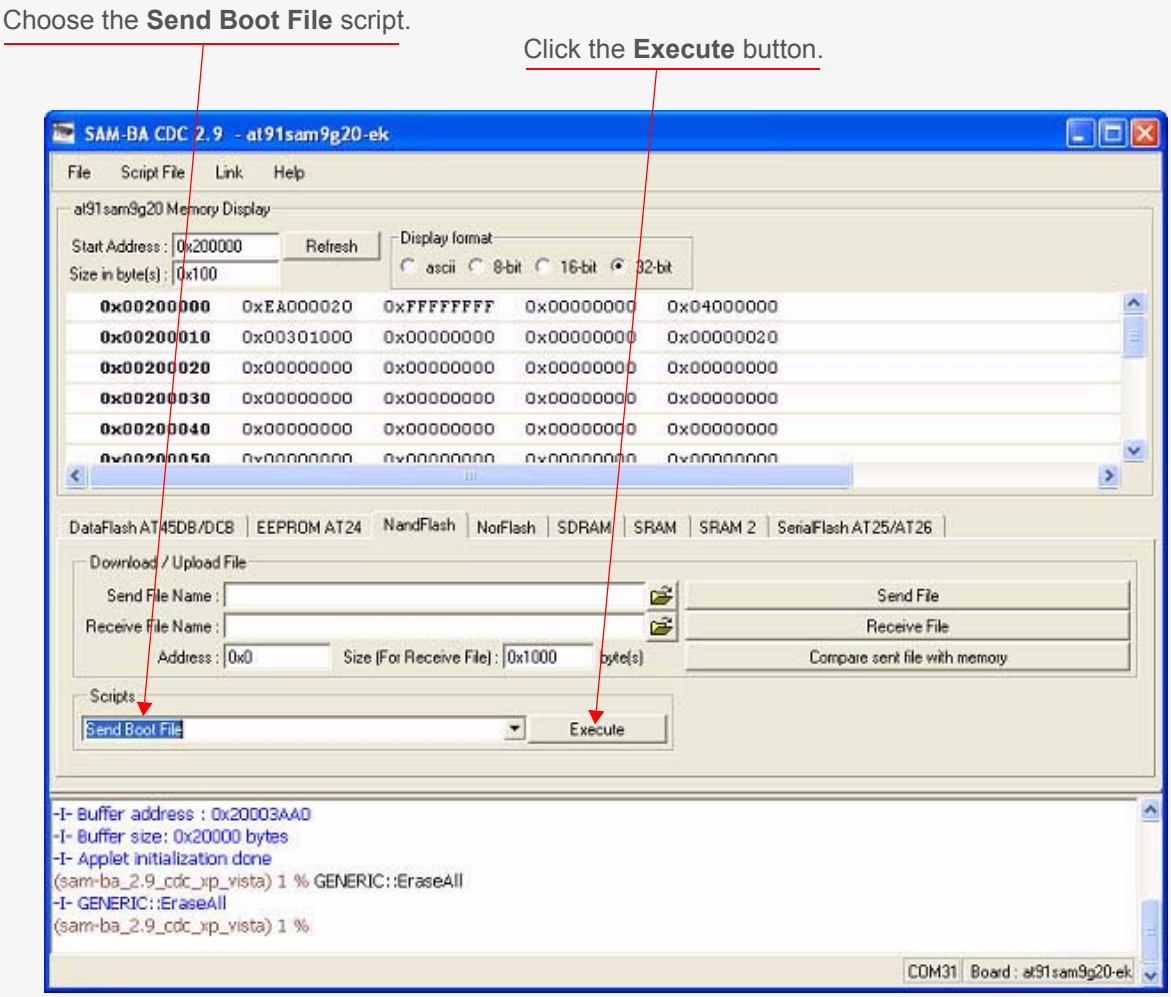


Figure 7-7: Sending the Boot File

You will be prompted for the file to send. Browse to your **E10_Recovery** directory and select the **bootstrap.bin** file. See Figure 7-8.

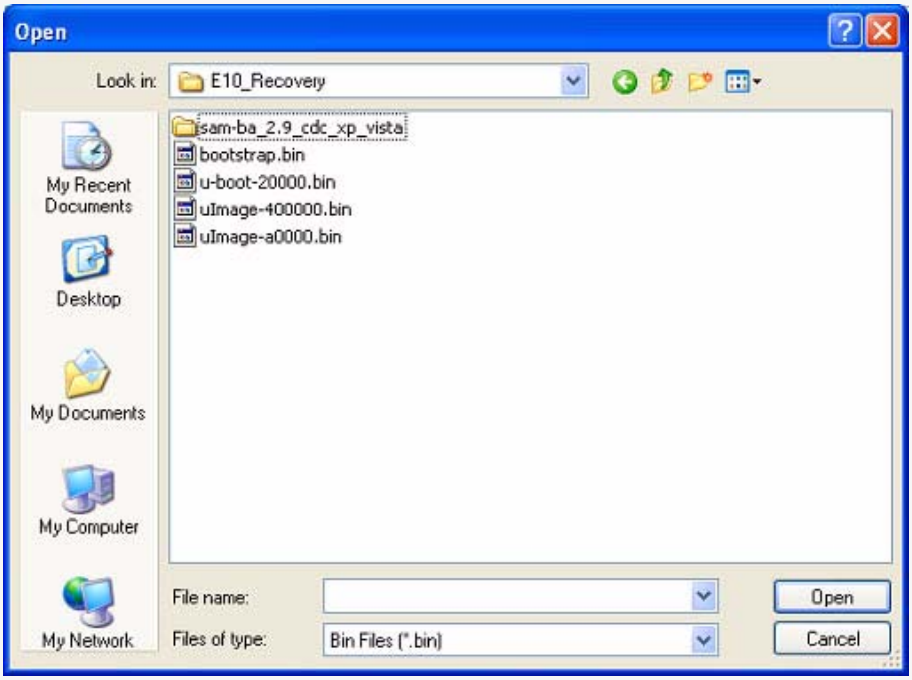


Figure 7-8: Browsing to the File

The **SAM-BA CDC** program transfers the file, and the GUI shown in Figure 7-9 on page 49 is displayed.

Step 12: Send the remaining files.

The remaining three files are transferred in a different way so you will not be using the **Scripts** pull-down menu for this step.

In the **Download/Upload File** section of the GUI there is a **Send File** button with a *choose file* icon (i.e., a folder icon) to its left. Click this icon (pointed out in Figure 7-9) to open a dialog box which allows you to browse to the **u-boot-20000.bin** file.

You then have to fill in the **Address:** field with a value of **0x20000** (notice that there are four zeros). Click the **Send File** button. The file is transferred to your E10.

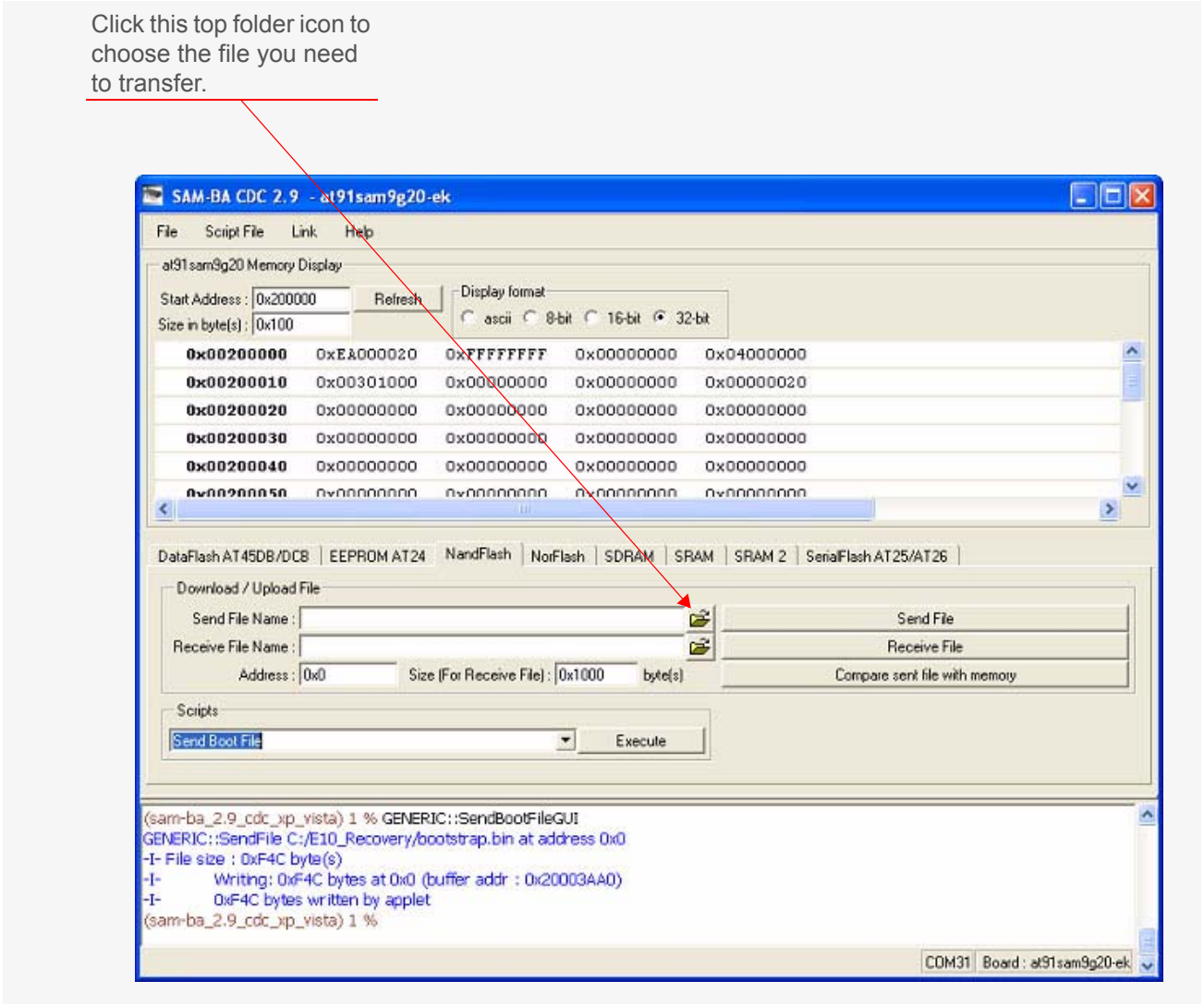


Figure 7-9: Sending Other Files

Once the file is transferred, the GUI should look like Figure 7-10.

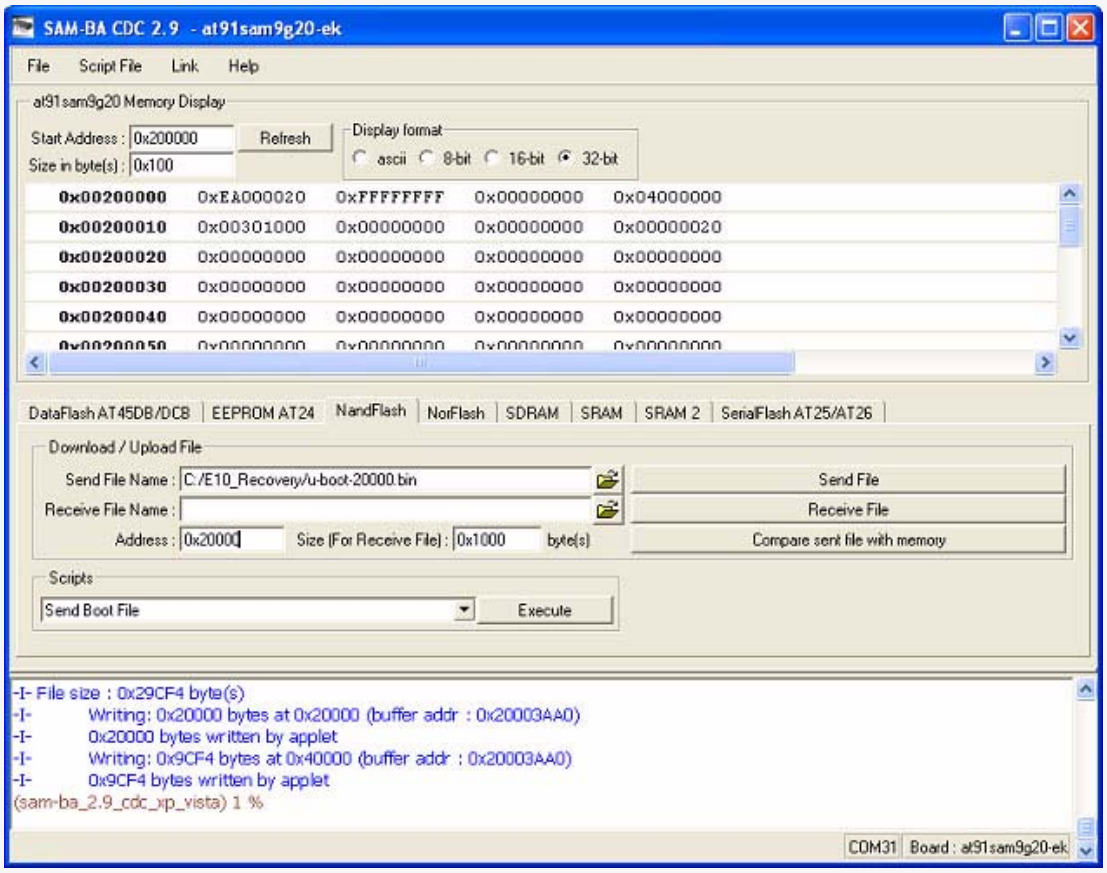


Figure 7-10: First File Transfer Complete

Using the same process, choose file **uImage-a0000.bin**, set the address to **0xA0000**, then send that file. The screen shown in Figure 7-11 appears.

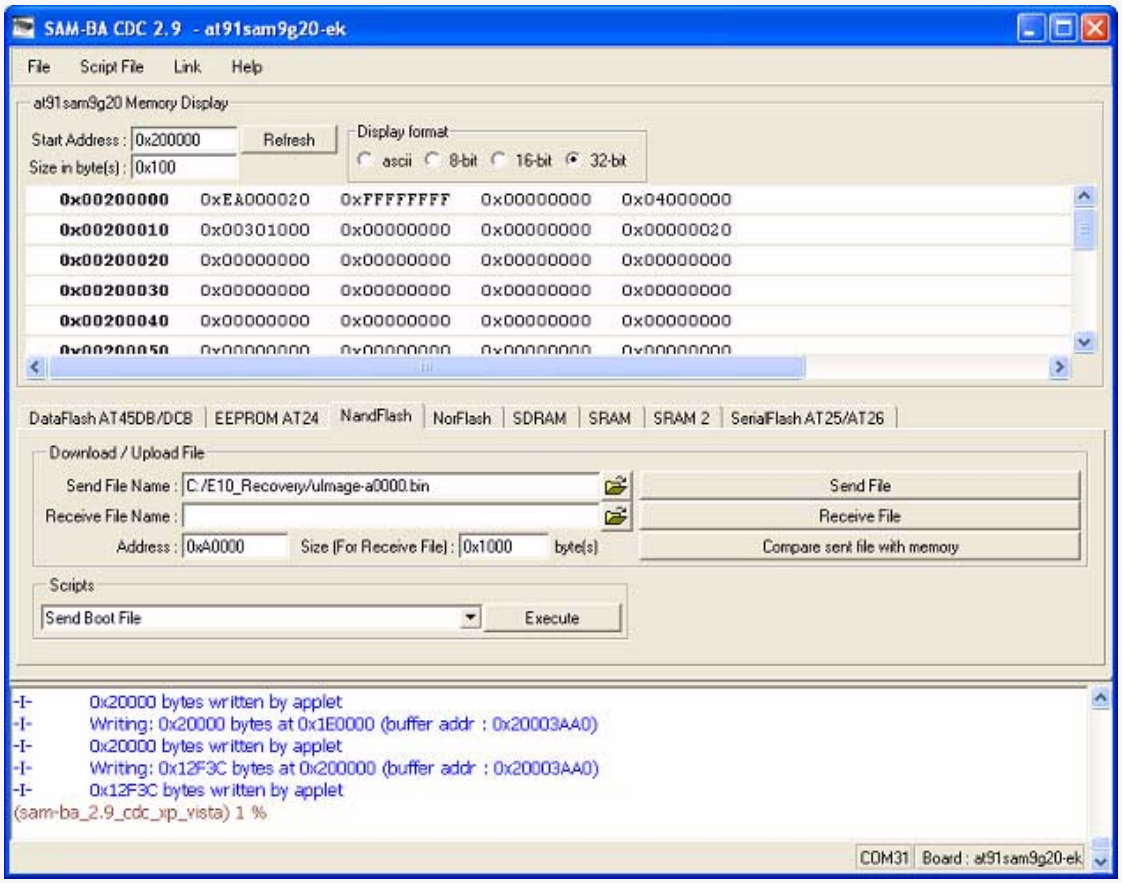


Figure 7-11: Second File Transfer Complete

Finally repeat the same procedure to choose file **uImage-400000.bin**, specify address **0x400000**, and send the last file. The screen shown in Figure 7-12 appears.

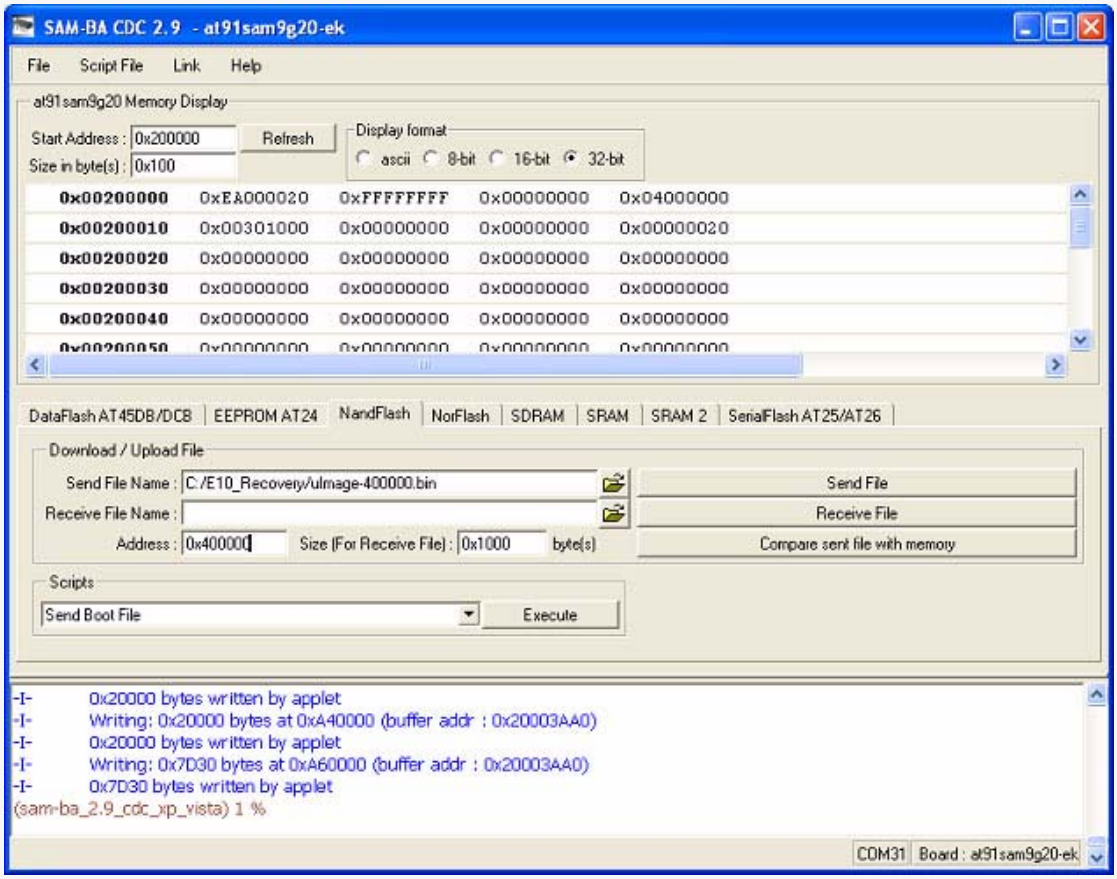


Figure 7-12: Last File Transfer Complete

Step 13: Reboot the E10.

Manually reboot your E10 by removing power, waiting a few seconds, then reapplying power.

After rebooting the E10, it automatically boots up into a Linux system with a root username of **`root`** and an empty password (just press **<Enter>** for the password field).

Now that you have restored *A* version of Linux and regained access to your E10, you will probably want to load the latest version of the Synapse software. For more information, refer to the chapter Upgrade Procedures on page 37.

Step 14: Reassemble your E10.

When reassembling your E10, notice that the circuit board slides into the *second* slot from the bottom (not the lowest slot). Reinsert the board and reattach both end plates. Your unit is now restored to the original operating software.

FURTHER ASSISTANCE

As mentioned before, we offer a wide range of documentation (user manuals, application notes, etc.) on our dedicated support forum. This forum also allows you to view questions and answers posted by other users (as well as post your own questions for discussion). You may also download the latest SNAP, Portal, E10, and SNAP Connect software.

Please visit http://forums.synapse-wireless.com to join.

Contact Information.

For any further information on Synapse products and services, contact us at:

Synapse Wireless, Inc.
500 Discovery Drive
Huntsville, Alabama 35806

256-852-7888
877-982-7888
256-852-7862 (fax)

www.synapse-wireless.com

Appendix A
Regulatory Information and Certifications

RF EXPOSURE STATEMENT

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance of 20cm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

FCC CERTIFICATIONS AND REGULATORY INFORMATION (USA ONLY)

FCC PART 15 CLASS A

These devices comply with part 15 of the FCC rules. Operation is subject to the following two conditions: (1) These devices may not cause harmful interference, and (2) These devices must accept any interference received, including interference that may cause harmful operation.

RADIO FREQUENCY INTERFERENCE (RFI) (FCC 15.105)

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

LABELING REQUIREMENTS (FCC 15.19)

This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

If the FCC ID for the module inside this product enclosure is not visible when installed inside another device, then the outside of the device into which this product is installed must also display a label referring to the enclosed module FCC ID.

MODIFICATIONS (FCC 15.21)

Changes or modifications to this equipment not expressly approved by Synapse Wireless, Inc. may void the user's authority to operate this equipment.

DECLARATION OF CONFORMITY

(In accordance with FCC 96-208 and 95-19)

Manufacturer's Name: Synapse Wireless, Inc.

Headquarters: 500 Discovery Drive
Huntsville, Al 35806

Synapse Wireless, Inc. declares that the product:

Product Name: SNAP Connect E10

Model Number: SLE10-001

to which this declaration relates, meet the requirements specified by the Federal Communications Commission as detailed in the following specifications:

- Part 15, Subpart B, for Class A equipment
- FCC 96-208 as it applies to Class A personal computers and peripherals

The products listed above have been tested at an External Test Laboratory certified per FCC rules and has been found to meet the FCC, Part 15, Emission Limits. Documentation is on file and available from Synapse Wireless, Inc.

INDUSTRY CANADA (IC) CERTIFICATIONS

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class A prescrites dans le Règlement sur le brouillage radioelectrique edicte par le ministère des Communications du Canada.