CHAPTER 1

Getting Started

Like many makers, I had started out making electronics using the Arduino and BASIC stamp microcontroller boards. Since then, I have transitioned onto using many other boards such as Freescale 8/16bit, MSP430 chipset boards, Digilent Chipkit, and Raspberry Pi. Although these boards are great, I always found myself wanting more—there was always something lacking in their features or they were not cost effective for my projects.

I heard about the Beagleboard and BeagleBone a while back, but it didn't grab my attention at the time, with no video output and a heavy price tag of $100–$150. When I read a press release from CircuitCo about a new BeagleBone Black, I thought it was too good to be true: the specification was incredible, with an Arm Cortex A8 1GHz CPU with 65 possible GPIO pins, HDMI video/audio output, and 512MB of DDR3 memory. Surely this would come with higher price tag than that of a small mobile computer. Well, I was wrong. CircuitCo announced that the price would be around $45—an incredible value.

Since receiving my BeagleBone Black, I have been hooked. All my projects use this microcontroller, and the more I learn about this board, the more features I unravel and the more it shows its true full potential as the ultimate microcontroller board. This chapter unravels the unknown mysteries of the BeagleBone Black. It gets you familiar with using the hardware, software, and setting up your first project using the BoneScript programming language.

I know that having so much flexibility and versatility in a single device can sometimes make things seem hard when you are just getting started. You will soon realize that there isn't a single right way of doing things: there can be many different ways of achieving the same outcome. Hopefully, this chapter will get you heading in the right direction.

Powering Up Your BeagleBone Black

You have many ways to connect to your BeagleBone Black development board—more specifically, you have many ways to access it to start programming using BoneScript. The preferred software tool for programming the BeagleBone Black is the Cloud9 IDE, but we'll cover every option for programming in this chapter.

Connecting Using USB

The BeagleBone Black comes with software/drivers already embedded in its operating system (OS) and includes documentation that will help you get connected to your computer. A great new hardware feature for the BeagleBone Black is that it comes with 2GB eMMC Flash memory

with the Angstrom OS preinstalled, so there is no need to download and flash a micro-SD card.

Go ahead and plug the mini universal serial bus (USB) cable supplied with your BeagleBone Black (BBB) board into your BeagleBone Black and the other end into your computer, as shown in Figure 1-1.

This will power the BeagleBone Black board and also provide a development interface in which you can program. The BeagleBone Black will boot Angstrom OS from the onboard 2GB embedded multimedia controller (eMMC), and the new revision C board will boot Debian OS. Once the BeagleBone Black is plugged in, the power LED (light-emitting diode) will be lit up and the adjacent bank of four LEDs will be flashing, as shown in Figure 1-2. The board is now alive!

The four LEDs are configured as follows:

- USR0 is configured at bootup to blink in a heartbeat pattern.
- USR1 is configured at bootup to light during micro-SD card access.
- USR2 is configured at bootup to light during CPU activity.
- USR3 is configured at bootup to light during eMMC access.

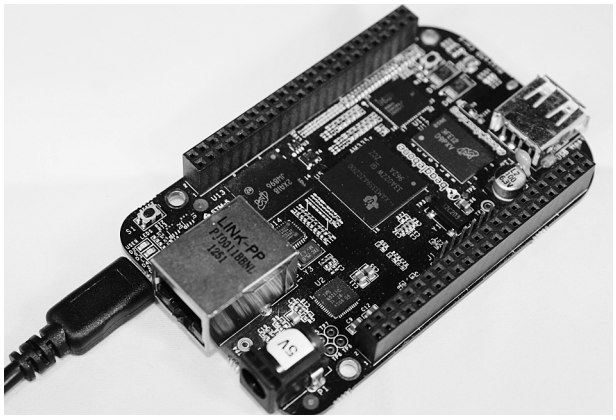

Figure 1-1 Connecting a USB cable to BBB

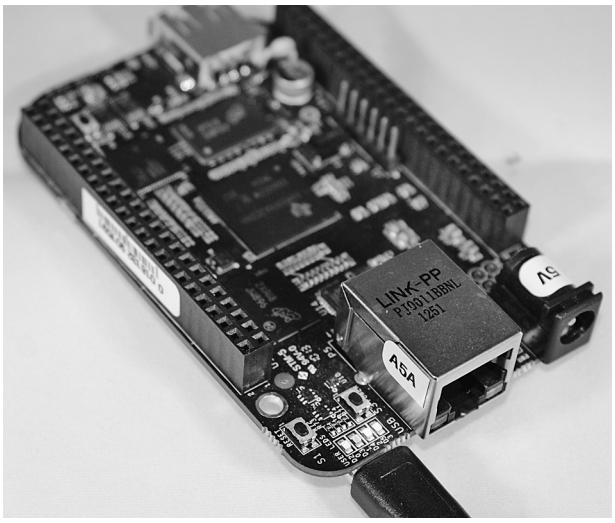

Figure 1-2 BeagleBone Black's bank of four LED indicators

NOTE **Holding down the boot switch when powering your BeagleBone Black will tell the hardware to boot from the micro-SD card instead of the onboard eMMC.**

Installing Drivers

Drivers need to be installed on your computer system before you can do anything with the board, so the following instructions show you how to install drivers on the Windows, Mac OS X, and Linux operating systems. Once you have powered up your BeagleBone Black through USB, it will operate as a flash drive, giving you all the necessary files you need to get started, including drivers and documentation, as shown in Figure 1-3.

Open the file directory on the flash drive and double-click start.html to open this document in your default web browser. The web page you see before you is a step-by-step quick start guide on installing all the relevant software. It provides web links for all stages, making it easy to follow. Go to step 2, shown in Figure 1-4, and select the operating system you are currently using; if prompted, click Run and follow the instructions to install the drivers.

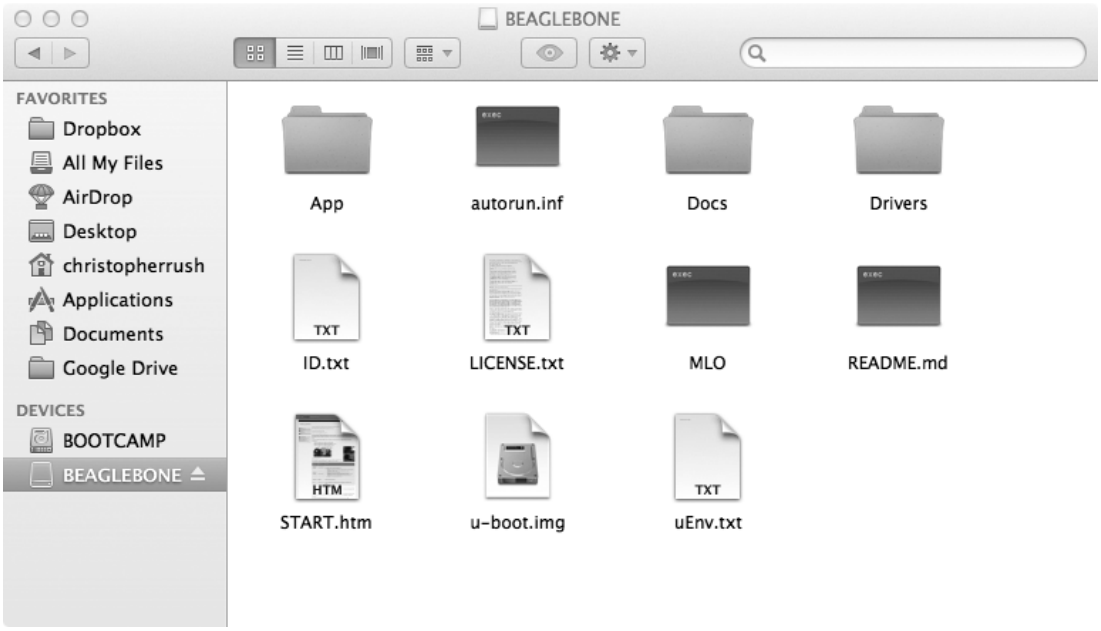

Figure 1-3 The file directory for BBB

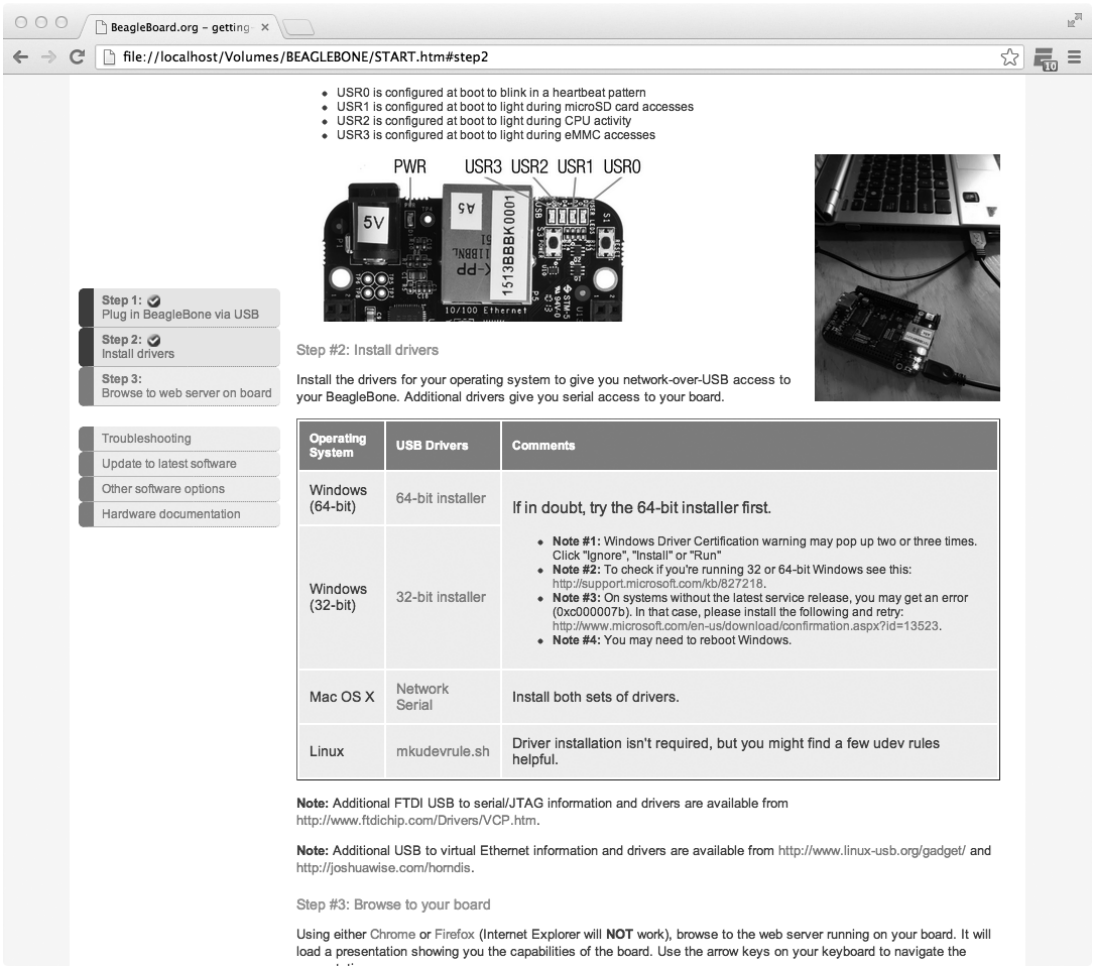

Figure 1-4 Selecting operating system drivers

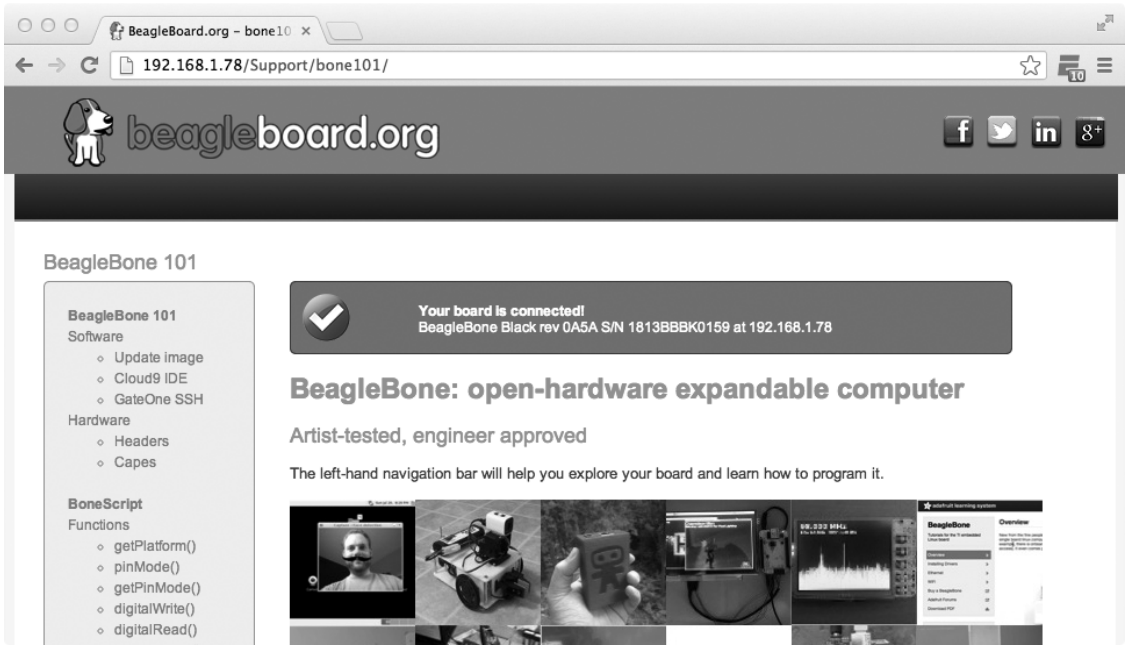

Figure 1-5 The Bone101 web page

NOTE **You must install the "FTDI USB to serial/JTAG" drivers if you are connecting through USB.**

Once this step is complete, open your web browser (either Firefox or Chrome, not Internet Explorer) and type in the URL http://192.168.7.2/. This will confirm the connection to your BeagleBone Black and load up the Bone101 web page on the device, as shown in Figure 1-5.

Congratulations, you are now one step closer to becoming an Evil Genius.

Connecting to Your BeagleBone Black Through Ethernet

Connecting your BeagleBone Black to your Ethernet network is very convenient when you are connecting using different computers in different locations. This method does require a 5V PSU with a 2.1mm DC jack for powering up the BeagleBone Black. So go ahead and plug in the Ethernet cable connected to your router/modem and then plug in the DC power jack, as shown in Figure 1-6.

Next, you need to find out what the BeagleBone Black's Internet Protocol (IP) address is so you can access it in your web

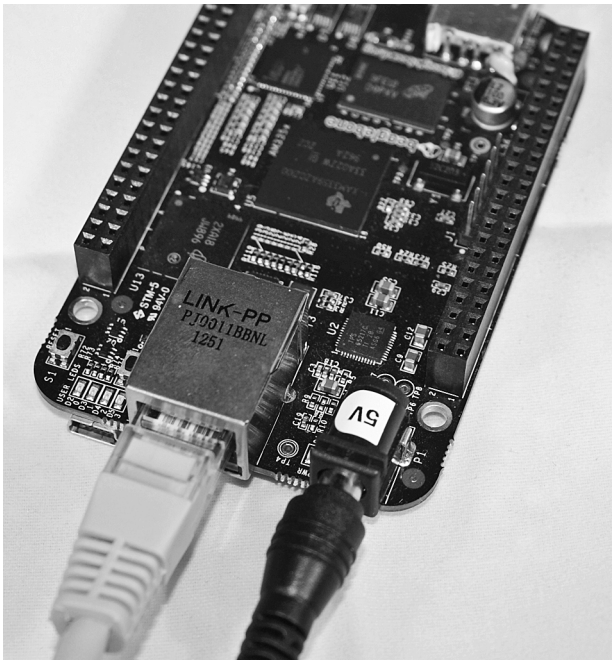

Figure 1-6 Ethernet and DC jack connected to BBB

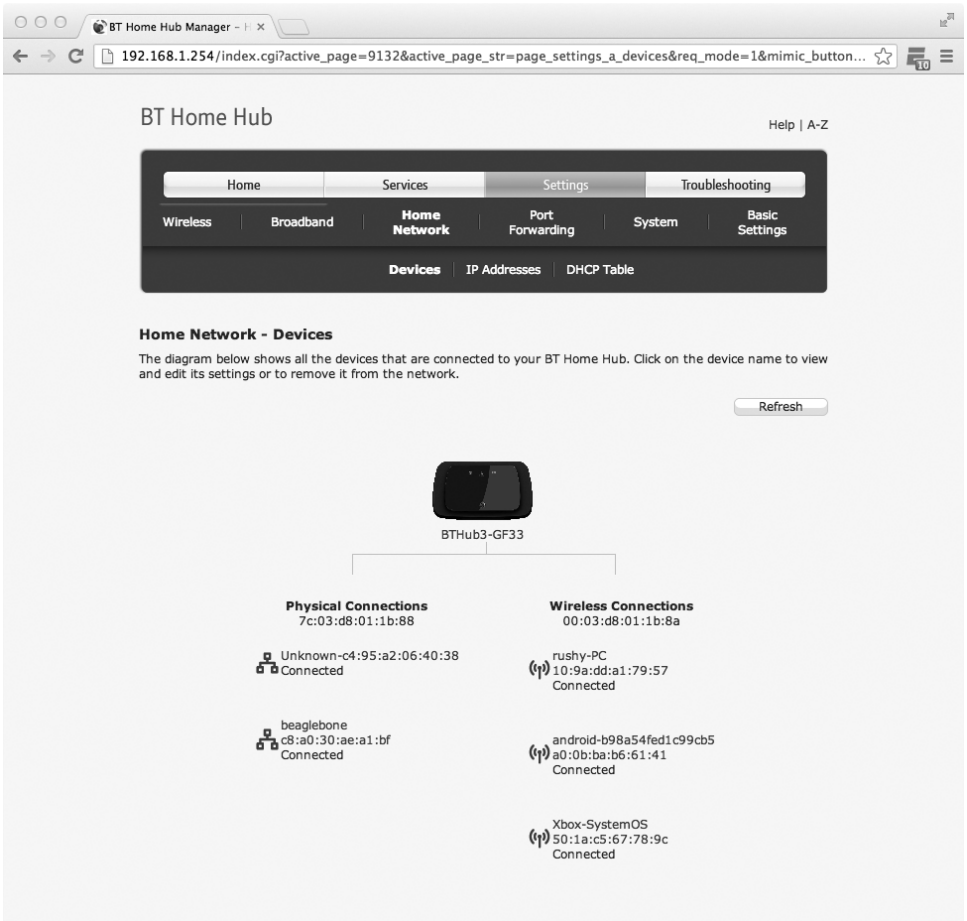

Figure 1-7 BBB IP address

browser. This address is automatically assigned to the BeagleBone Black using the Dynamic Host Control Protocol (DHCP) on your router/modem. The easiest way to find this is to open up your router/modem settings and locate the device under the wired Ethernet settings, like in Figure 1-7.

Once you have located the BeagleBone Black IP address, open your web browser (either Firefox or Chrome, not Internet Explorer). In the URL bar, type the address http://x.x.x.x/, where "x.x.x.x" is your IP address. Once you have done this, press ENTER. You should now be connected to your BeagleBone Black board and see the Bone101 web page, as shown in Figure 1-5.

Congratulations, you are now connected to your BeagleBone Black through an Ethernet connection. The Bone101 page provides a lot of information about the board itself, including some great examples using the BoneScript JavaScript library and some great add-on boards called "capes." Feel free to browse around.

Connecting via SSH Through USB and Ethernet

Secure Shell (SSH) is a cryptographic network protocol for secure data communications between devices, and is more commonly known for remote command execution using a client/server model. Accessing the BeagleBone Black through SSH can be accomplished either by connecting the BeagleBone Black to your computer using the USB cable provided or

by connecting the BeagleBone Black to your Ethernet network.

SSH clients come standard on most operating systems (except Microsoft Windows). This is not a problem because a load of software is available that can allow you to connect to your BeagleBone Black through SSH, and this section shows you how to do that.

SSH Through GateOne SSH

Due to the fantastic features on the BeagleBone Black, GateOne SSH comes preinstalled. GateOne SSH is a terminal emulator that you can use through your web browser without installing any additional software on your computer. This is by far the easiest method of connecting via SSH, and the most convenient for connecting using different operating systems, even on mobile devices.

The first step is to open your web browser and navigate to the Bone101 web page (http://192.168.1.78/Support/bone101/). In the navigation menu on the left side, you'll see GateOne SSH under the heading Software. Click the link, which directs you to the GateOne SSH web page shown in Figure 1-8.

Once the web page has loaded, you should see a window that looks similar to a terminal window on your BeagleBone Black. At first you will be prompted to enter connection details

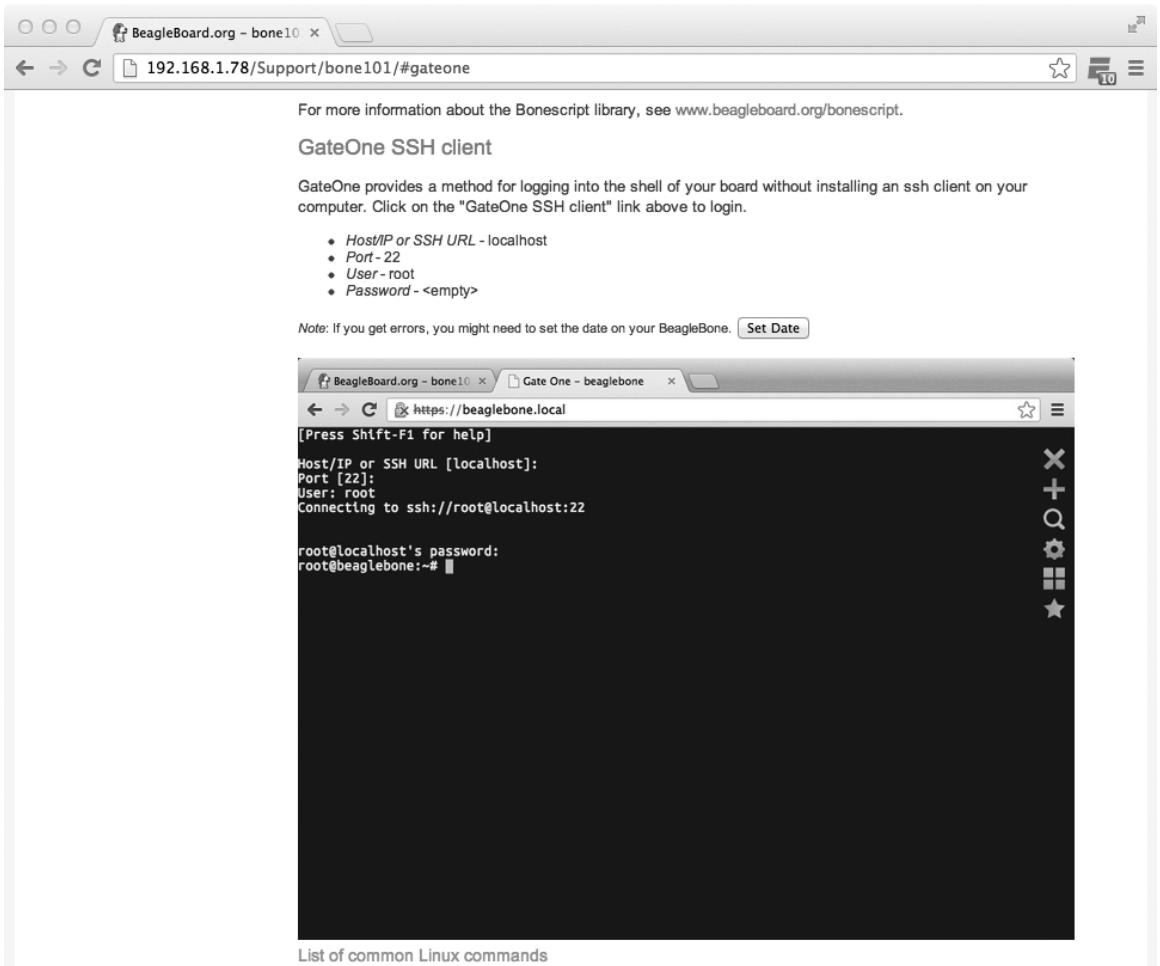

Figure 1-8 The GateOne SSH page

such as IP address, port number, username, and password. If you are connected to the BeagleBone Black through either USB or the local Ethernet network, leave the first line blank when asked for an host/IP address and also when you're prompted to enter a port number, because this instance will be using the default settings to connect. Enter your username and password, and you will be connected to your BeagleBone Black through SSH. You will see "root@beaglebone:~#" once connected.

NOTE **The default username is root, and no password is set.**

Connecting via SSH in Windows

To get SSH running in Windows, you need to download an SSH client program. The most popular program to use is PuTTY, and it can be downloaded from www.putty.org. Once it is downloaded and installed, you can go ahead and run the putty.exe program file. You will enter the configuration window upon starting PuTTY (see Figure 1-9). This is where you add your BeagleBone Black settings, which will enable you to connect over SSH.

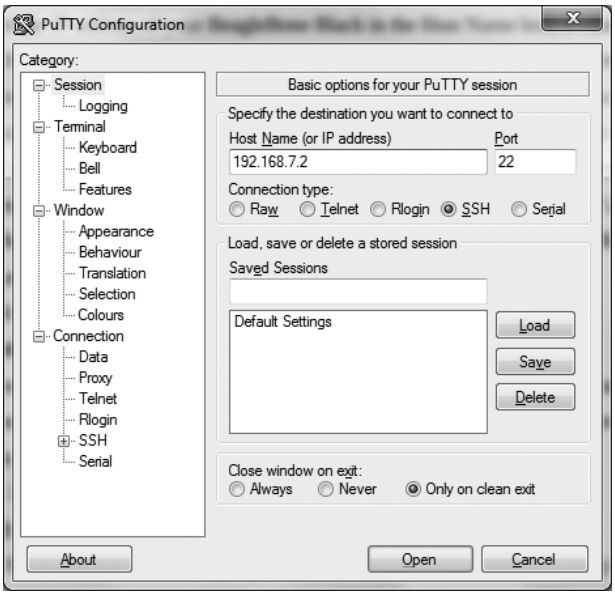

Figure 1-9 The PuTTY Configuration menu

Enter the IP address of your BeagleBone Black in the Host Name box; if you are connected via USB, the IP address will be 192.168.7.2. Alternatively, if you are connected through the local Ethernet network, your BeagleBone Black will have a local IP address that you can obtain through your router/modem. Next, click Open, and the client will try to establish a connection. A security box will appear; you will be required to enter the default username and password details of the BeagleBone Black. Log in with "root" and no password, after which you can just press ENTER. You are now connected to your BeagleBone Black, as shown in Figure 1-10, and can view the root file system and issue commands.

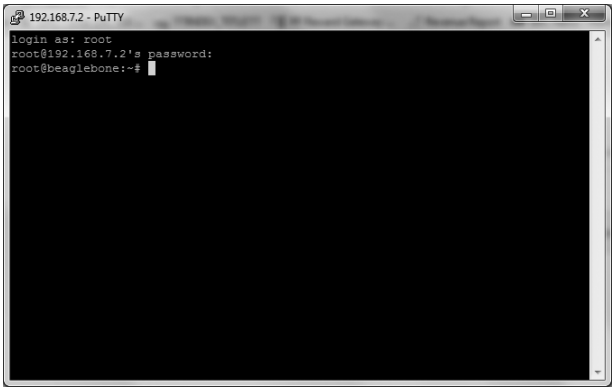

Figure 1-10 The PuTTY window after you have logged in

Connecting via SSH in Mac OS/Linux

If you are a Mac or Linux user, you're in luck: an SSH client comes with the standard installation and can be accessed using the Terminal window (see Figure 1-11). All you have to do is type the following command:

```
ssh 192.168.7.2 –l root
```

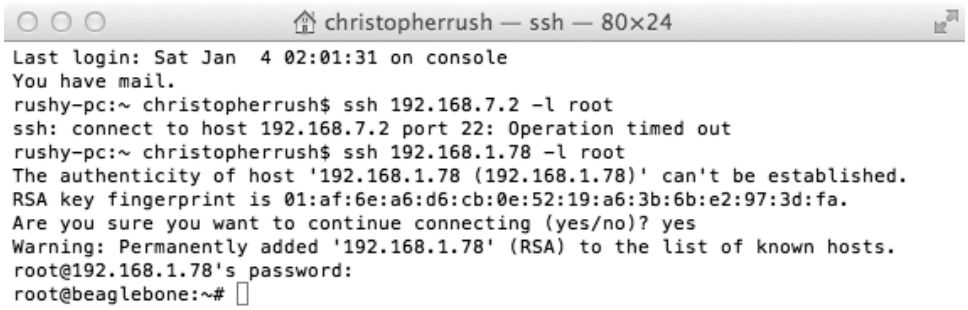

Figure 1-11 The SSH login procedure for Mac OS X

The first time you log in, you will receive an authentication warning message. Just type yes to continue with the SSH session and press RETURN (or ENTER) when prompted for a password, given that no default password is set on the BeagleBone Black device.

Installing Software and Running Updates

The BeagleBone Black comes preinstalled with the Angstrom operating system, but sometimes you may want a clean install or you may want to run an update of the current files. I always recommend that you have the latest and most up-to-date operating system and software, so I'll show you how to do both.

Installing the New OS Image

Before we get started installing a new image, a couple of items are required:

- 4GB micro-SD card
- Micro-SD card reader

The BeagleBone Black has 2GB of eMMC storage (4GB on RevC) that can be initialized by a program booted off a micro-SD card.

Download the latest distribution from http://beagleboard.org/latest-images, where you will find the most up-to-date image of the OS. Depending on your Internet connection speed, the download may take up to 30 minutes.

Writing the Image in Windows

Once the distribution is downloaded, you will notice that the file has an .img.xz extension, which is a compressed sector-by-sector image of the SD card. To decompress this file, you need to download and install 7-Zip, which is available for all operating systems from www.7-zip.org/download.html (see Figure 1-12).

NOTE **Make sure you download BeagleBone Black (eMMC flasher) if you want to replace the current image on the eMMC.**

Once 7-Zip is downloaded and installed, decompress the Angstrom image file by right-clicking the image file and selecting 7-Zip | Decompress Image. A decompression in progress is shown in Figure 1-13.

Once the image file is decompressed, you can now download and install an image writer program that will write the image file to the micro-SD card. For Windows computers, you can download Win32 Disk Imager, which is free from SourceForge (http://sourceforge.net/projects/win32diskimager). Launch the Win32 Disk Imager software by double-clicking the Win32DiskImager file in the folder to which you extracted it and then click the file icon next to the Image File field (see Figure 1-14). Navigate

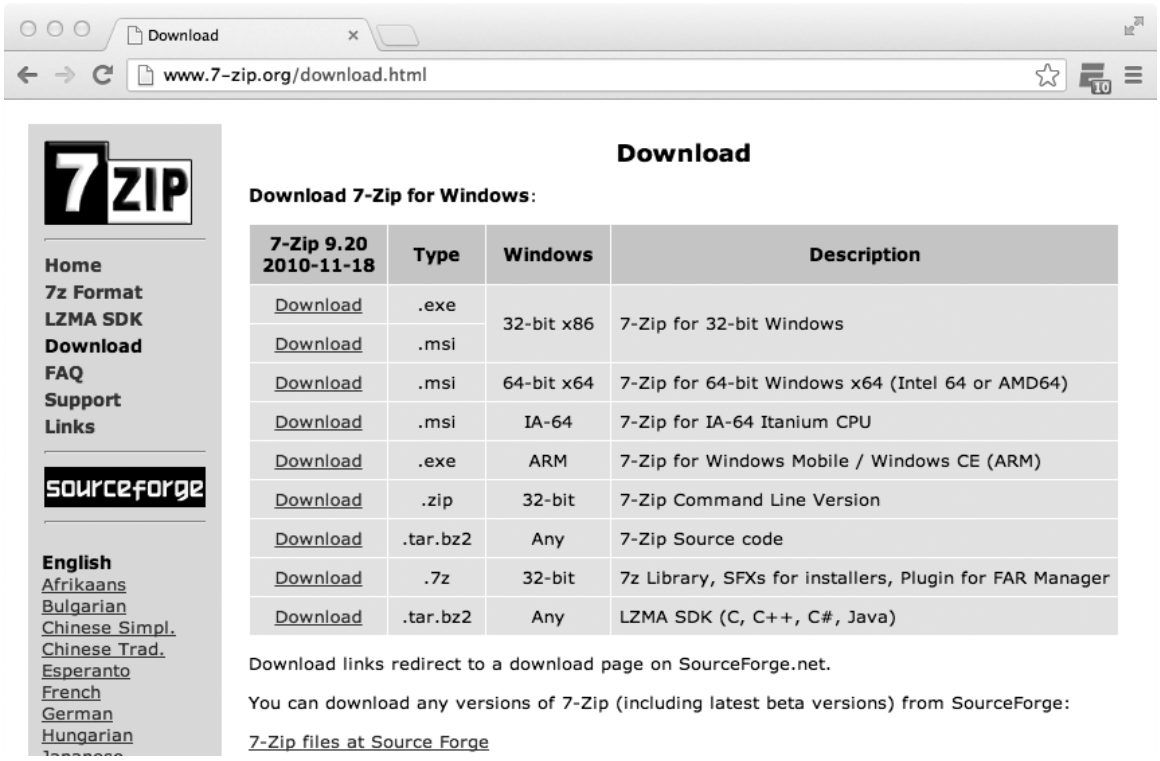

Figure 1-12 The 7-Zip download page

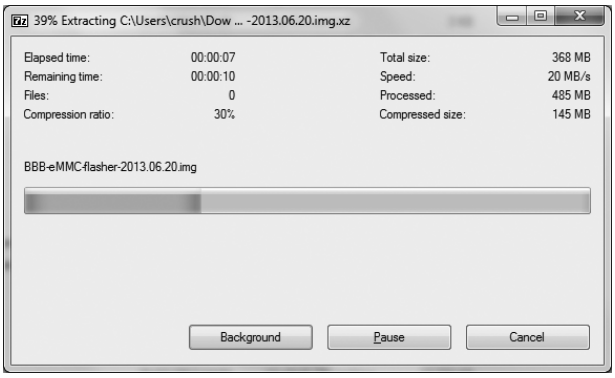

Figure 1-13 7-Zip decompressing the image

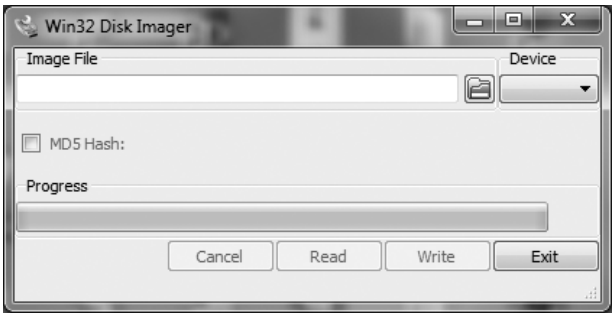

Figure 1-14 Win32 Disk Imager

to where you decompressed the Angstrom image and select the image file (see Figure 1-15).

Once this is done, click the Write button and wait for the image file to complete writing to the microSD card.

CAUTION **Make sure you select the correct drive letter; otherwise, you risk losing data on other devices!**

Writing the Image in Mac OS X

To decompress the image file in Mac OS X, the best program you can use is The Unarchiver, which you can find in the App Store or from http://wakaba.c3.cx/s/apps/unarchiver.html. Once you have downloaded and installed the application, navigate to where the Angstrom image file was downloaded and double-click the

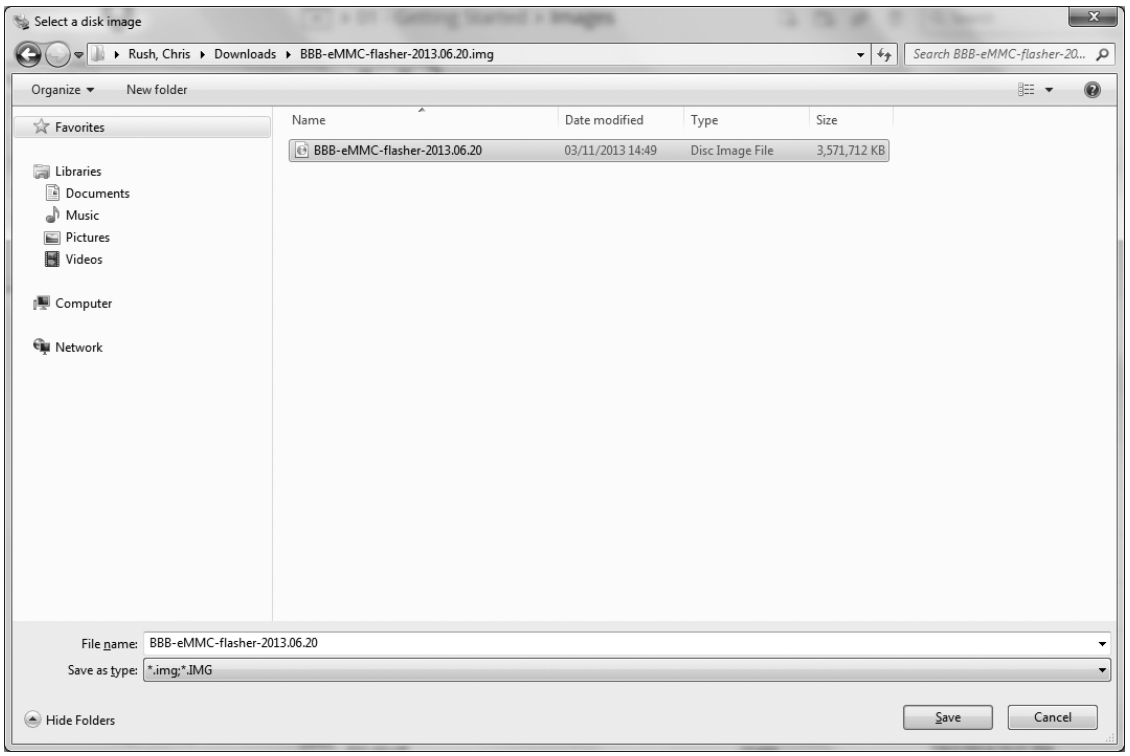

Figure 1-15 Navigate to Angstrom image

image to begin decompressing. It will take a few minutes to extract the file.

Now that you have the image file, you can write the image to the micro-SD card. On the Mac OS X operating system, you have a couple of ways you can do this. For safe measure, we will cover both ways.

Using a GUI The first thing you'll want to do is download a program called PiFiller, which you can find at http://ivanx.com/raspberrypi. Even though this program was specifically designed for use with the Raspberry Pi, you can apply the program concept to any SD card using any image.

CAUTION **Remove your SD card before launching the PiFiller program.**

Once this program is downloaded, double-click the application and follow the onscreen instructions for selecting the Angstrom image file and inserting the micro-SD card. Click Continue once you have selected the correct SD card in your computer, as shown in Figure 1-16, and the program will start to write the Angstrom image to your SD card. This process may take between 15 and 20 minutes.

Figure 1-16 Selecting the correct SD card in Mac OS X

Using the Command Line A great alternative is to flash the micro-SD card entirely using the command-line interface. You can achieve this simply by opening up the Terminal window in Mac OS X and typing the following command:

```
Df –h
```

Executing this command gives you a list of storage devices connected to your Mac. Identify your micro-SD card from the list and take note of the file system name.

The following command will unmount the micro-SD card (/dev/disk4s1) from your Mac so you can then write the image file safely:

```
Sudo diskutil unmount /dev/disk4s1
```

Now you will want to input the following command, which will write the image to the SD card. When typing this command, make sure you choose the correct location for your image file (in this case, it's /Downloads/BBB/*filename*.img).

```
Sudo ddb s=1m if=~/Downloads/BBB/eMMC-
flasher-2013-05.08.img of=/dev/disk4s1
```

It should take between 15 and 30 minutes to write the image to the micro-SD card, depending on the speed of your computer.

CAUTION **Make sure you input the correct disk name because you risk losing data on your master hard drive!**

Flashing the BeagleBone Black's eMMC

Now you should have a fully flashed micro-SD card with the latest version of Angstrom downloaded from the BeagleBone Black web page. Insert the micro-SD card into the BeagleBone Black (powered down) and then power on the board, either via the USB cable or a 5V DC adaptor, while holding down the USER/BOOT button at the same time. Holding down the USER/BOOT button will force the hardware of the BeagleBone Black to boot from the micro-SD card, and the process of flashing the onboard 2GB eMMC will begin. You'll need to wait for up to 45 minutes for this process to complete. Once the flashing has completed, your BeagleBone Black board will light up all four of the LEDs. Power down your board and remove the micro-SD card to complete the process. Your BeagleBone Black is now up to date with the latest Angstrom image and can be booted back up as normal from the 2GB eMMC memory.

Running Updates on the BeagleBone Black

The easiest way to get the latest software updates (and by far the quickest) is to connect your BeagleBone Black to the Internet and run the update commands in the Terminal window. Software updates and upgrades come in packages using a program called opkg, which is specifically designed for embedded Linux devices to manage the process of installing or updating the packages on your BeagleBone Black. The opkg package manager keeps a list of current software versions locally on your BeagleBone Black, so before you run an upgrade, you must update the list. Type the following command in the Terminal:

```
opkg update
```

You will see the BeagleBone Black download the latest listings from the Internet. Now that you have the most up-to-date list of software packages, as shown in Figure 1-17, you can go ahead and upgrade them to the latest version using the following command:

```
opkg upgrade
```

Depending on how many different software packages need upgrading, this process may take a while.

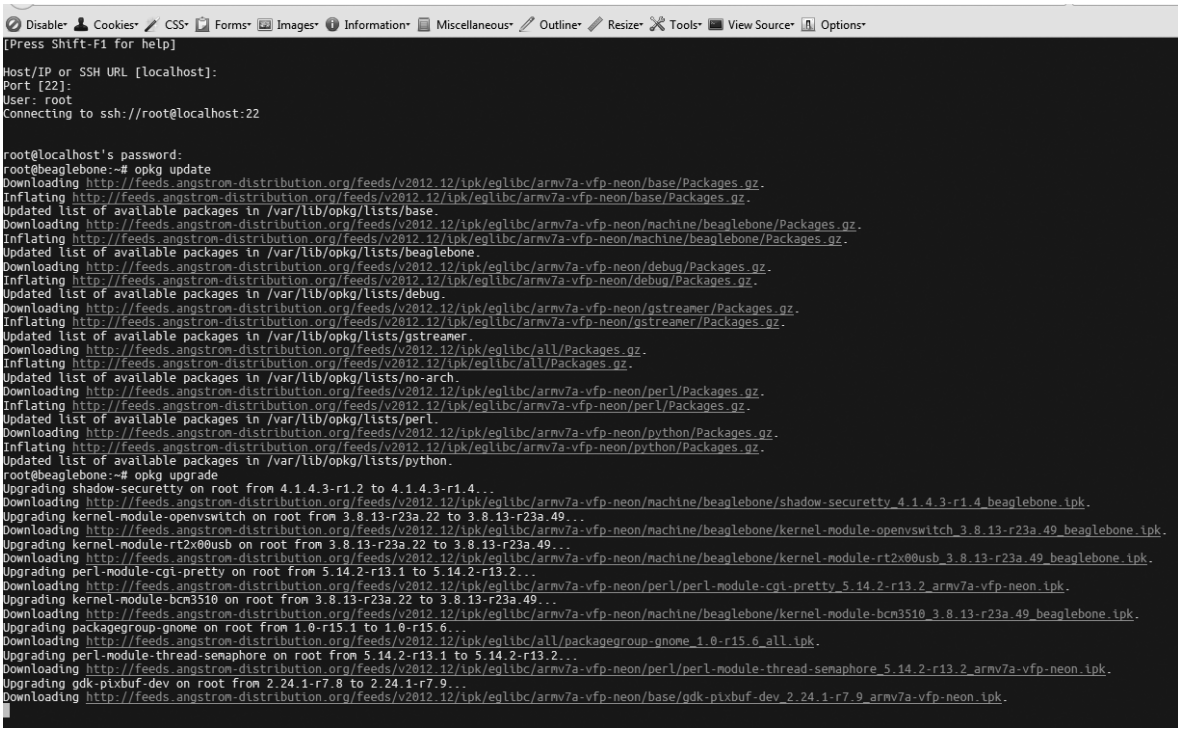

Figure 1-17 Upgrading packages

BeagleBone Black Pinout

Most micro-development platforms provide a range of different inputs and outputs using headers called GPIO (General Purpose Input/Output) pins. These pins allow you to control things electronically using both hardware and software, and each pin can serve a specific function—either analog or digital. Most microcontrollers come with a pinout diagram with labeled pins stating their function.

The BeagleBone Black comes with two 46-pin dual-row expansion headers, labeled P9 and P8, also known as Expansion A and Expansion B, respectively (see Figure 1-18). Each pin provides 3.3V, unless otherwise stated.

Digital GPIO Pins

The BeagleBone Black comes with 65 GPIO pin headers, which is a huge amount of control at your fingertips. These headers are labeled GPIO_xx, and you can control these pins by switching an output to either "on" or "off" (1 or 0, respectively). You can also detect a digital input signal to sense when a digital input device such as a switch has been activated (on, or 1) or deactivated (off, or 0). These pins are perfect for controlling a vast array of LEDs.

CAUTION **Unlike other microcontroller boards, the BeagleBone Black's pins operate using 3.3V. Using anything above this level can permanently damage the board.**

Analog Pins

Seven analog pins are available on the BeagleBone Black, labeled AINx. These pins are designed to detect analog signals coming into the BeagleBone Black from such devices as temperature sensors. The BeagleBone Black has a built-in 12-bit ADC function that allows you to convert the in-coming analog signal to a more readable digital value.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_40
GPIO_48	15	16	GPIO_51
GPIO_4	17	18	GPIO_5
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_125	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

Figure 1-18 BeagleBone Black GPIO pinout

CAUTION **Make sure you don't input more than 1.8V to the analog input pins; otherwise, you risk damaging the board.**

I2C Pins

The BeagleBone Black contains two I2C pins, labeled I2CX_SCL and I2CX_SDA. The first I2C bus is used for reading EEPROMS on the BeagleBone cape add-on boards and can't be used for other digital I/O operations. However, you can still use it to add other I2C devices. The second I2C bus is available for you to use to configure your devices. The I2C bus allows you to add multiple devices to the BeagleBone Black using the I2C addressing system.

SPI Pins

Two SPI (serial peripheral interface bus) ports are available for use with SPI-compatible devices. SPI is a synchronous data link between devices and operates using a full duplex mode, which enables a faster data transfer. Generally, one device operates as a master and the other a slave in order to synchronize. This also allows you to add multiple slave devices, usually in a daisy chain configuration.

UART Pins

The BeagleBone Black comes with a dedicated header for getting to the UART0 pins and for connecting a debug cable. In addition to the debug header, there are five serial ports in the headers, but only one of them has a single direction.

Project 1
Blinking an Internal LED

Having successfully set up your BeagleBone Black, you are now eagerly anticipating your first project. This project will get you familiar with using the BeagleBone—nothing exciting is

going to happen, nor any evildoing! However, it's a start—and we all have to start somewhere. This project requires no additional electrical hardware, thus allowing us to primarily focus on the programming side of things. This also ensures that you have everything set up correctly on your BeagleBone Black board.

So, without further ado, we are going to write a program that will blink the onboard LEDs on the BeagleBone Black. If you have had some previous programming experience, this is going to be our "Hello World" program. We will write this program from scratch so you can get a feel for the structure of using BoneScript in Cloud9 IDE. We will go through the code line by line to give you an explanation of what each function does.

The code for blinking the internal LED is as follows:

```
var b = require('bonescript');
var led = "USR3";

b.pinMode(led, b.OUTPUT);

var state = b.LOW;

b.digitalWrite(led, state);

setInterval(toggle, 1000);

function toggle() {
    if(state == b.LOW) state = b.HIGH;
    else state = b.LOW;
    b.digitalWrite(led, state);
}
```

When writing a program using BoneScript, we need to point our program to the BoneScript library so we can access the GPIO headers and other functions on the BeagleBone Black. Therefore, the first line of our code creates a variable (b) that we can reference in our code to access the BoneScript listed in the parentheses:

```
var b = require('bonescript');
```

The next logical piece of code is to create a new variable so we can refer to the onboard LED labeled USR3. For this instance, we call the variable led, and the string to access the LED is `"USR3"`:

```
var led = "USR3";
```

The GPIO digital pins on the BeagleBone Black can be set as either an input or an output pin, so in our code we need to tell the BeagleBone Black that we want our onboard LED to be an output. To do this, we use a function called pinMode and then select in parentheses the pin we wish to use; in this case, we are using the variable led, and then we set the pin as an output using variable `b.OUTPUT`:

```
b.pinMode(led, b.OUTPUT);
```

In this program, we create a loop that gathers the state of the LED (either HIGH or LOW) and then alternates between these values to switch the LED on and off. To do this, we need to set another variable, called `state`, and assign a value to it; in this case, we initially set the LED to "off" (or `b.LOW`):

```
var state = b.LOW;
```

Now that we have set the variable for the LED, we need to send the command to the digital pin on the BeagleBone Black to set the LED to `LOW` (off). To do this, we use the function `digitalWrite` and select the GPIO pin led and choose whether it is `HIGH` or `LOW` (that is, we set its "state"):

```
b.digitalWrite(led, state);
```

After the state of the LED has been set, we need to toggle the LED to flash on and off. To do this, we set a time interval of 1,000 milliseconds using the function `setInterval`. After 1,000 milliseconds, we call the function `toggle`:

```
setInterval(toggle, 1000);
```

We now have created a function called `toggle`, and every 1,000 milliseconds, this function will be called. Now we need to alternate the state of the LED between `HIGH` and `LOW`.

The easiest way to achieve this is to use an `if else` statement to create a different action for different decisions. The statement is either `TRUE` or `FALSE`, as shown here:

```
if (condition) TRUE
  {
  code to be executed if condition is
true
  }
else FALSE
  {
  code to be executed if condition is
not true
  }
```

In our function, we check to see if the state is equal to `LOW`, and if it is (`TRUE`) we then set the state to `HIGH` (that is, turn the LED on). If the state is not equal to `LOW`, we then trigger the `else` statement (`FALSE`) and we set the state to `LOW` (that is, turn the LED off). Once we have set the state (or alternated between `ON` and `OFF`, or `HIGH` and `LOW`), we then issue the `digitalWrite` function to turn the LED on or off:

```
function toggle() {
    if(state == b.LOW) state = b.HIGH;
    else state = b.LOW;
    b.digitalWrite(led, state);
}
```

Go ahead and run the program by clicking Run in the Cloud9 menu bar, shown in Figure 1-19.

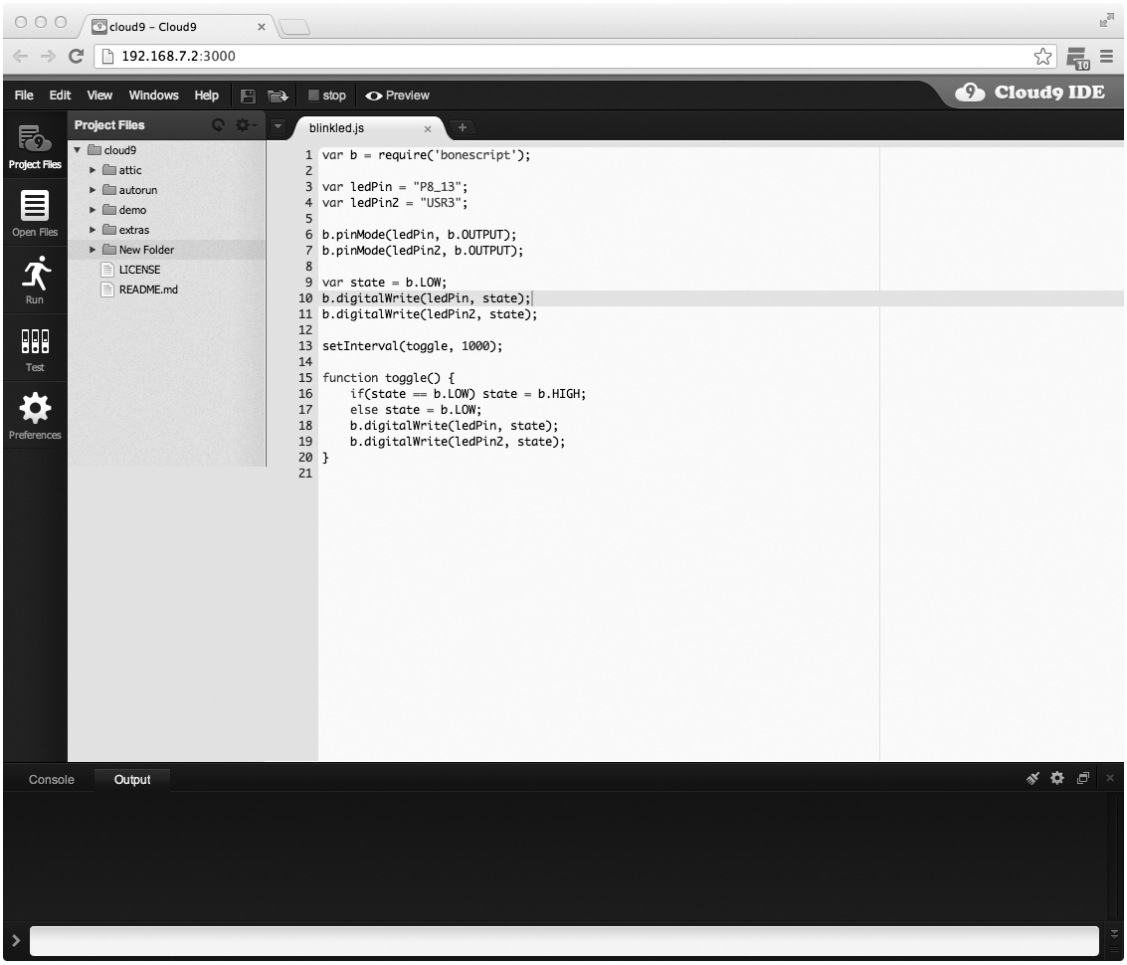

Figure 1-19 The Cloud9 Run program

You should now see the USR3 LED flash on and off every 1,000 milliseconds. If you want to change the interval timing between flashes, you can change the timing in the following line:

```
setInterval(toggle, 1000);
```

Summary

You now have your BeagleBone Black set up and have created your first project. It's not the most exciting Evil project we will conjure, but you now know that everything works and you have learned the basics of how the BeagleBone Black operates. You will move on to more advanced projects as you progress through this book. Along the way, you will gain insight into various ways you can create your own Evil projects.