

How to use all the GPIO on Beaglebone Black in Python

GPIO include I/O, PWM, ADC, I2C, SPI, and UART

Meng Cao

8 November 2013

Summary

Beaglebone Black original using BoneScript, which is similar to Java. Usually we use C for hardware programming. Python is a programming language which does much better on programming and debugging than C. So, this application note will teach about how to write programs using python. Also, the program would be able to control all the hardware through GPIO, PWM, ADC.

Introduction

When doing complex project, people usually think about getting a microcontroller to do all the controls and calculations. However, a microcontroller needs to be involved in PCB design, also need protection circuit designed. Raspberry Pi is an option which has the microcontroller on the board with other basic devices such as USB, Ethernet, and video output. Texas Instrument has a similar product called Beaglebone. The newest version of Beaglebone is Beaglebone Black. Compare to Raspberry Pi, the Beaglebone Black has more GPIO pins, built in HDMI, also much more powerful processor. Both products can run Linux as the on-board operation system, which means, you can run almost any language you want. This application is not going to talk about using Python as the program language.

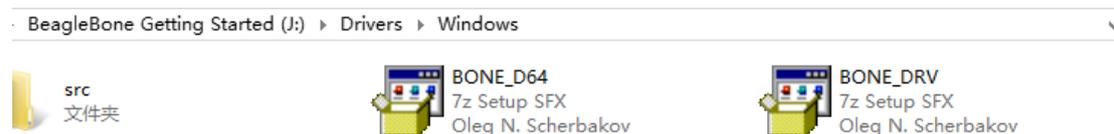
Things we need

1. Beaglebone Black
2. Python and libraries
3. Using libraries for GPIO, PWM, ADC
4. Setup SPI drivers on Beaglebone Black
5. Using library for SPI

Setup Beaglebone Black

The first step is to setup the Beaglebone Black if you have one in your hand. Beaglebone Black communicates with computers using a USB cable. It needs drivers to be installed on the computer. The reason for using a driver is because the driver sets up an IP address for Beaglebone Black even though it is only connected through USB. In that way, you can SSH to Beaglebone just like connecting to a Linux server using SSH.

1 **Connect Beaglebone to PC.** An extra drive called "BeagleBone Getting Started" appears in the computer. Run the drivers under the "Drivers" folder. "BONE_D64" for x64 windows or "BONE_DRV" for x32 windows.



2 **Browse Beaglebone Black.** Type “192.168.7.2” into address of your browser. (Avoid to use Internet Explorer, may not working properly). You should see the following page.

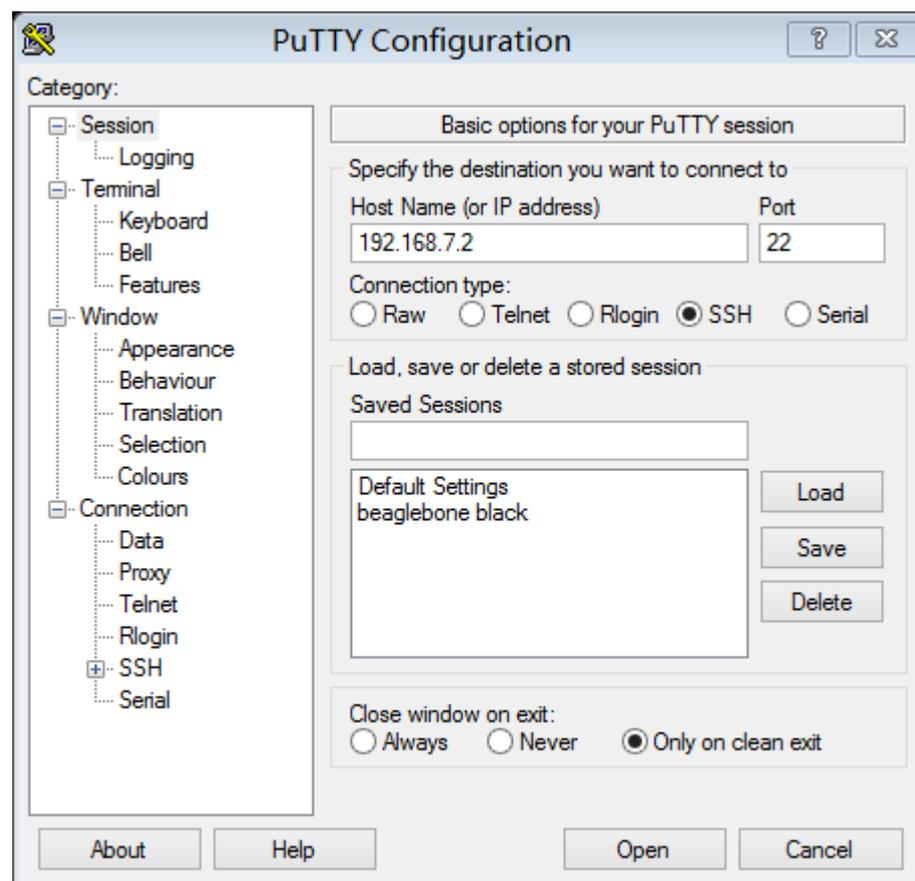
The screenshot shows the BeagleBoard.org website interface. At the top left is the BeagleBoard logo. To the right are social media icons for Facebook, Twitter, LinkedIn, and Google+. Below the header is a navigation menu for 'BeagleBone 101' with categories: Software (Update image, Cloud9 IDE, GateOne SSH), Hardware (Headers, Capes), Bone Script Functions (getPlatform(), pinMode(), getPinMode(), digitalWrite(), digitalWrite(), analogWrite(), analogRead(), attachInterrupt(), detachInterrupt(), readTextFile(), writeTextFile()), JavaScript (console(), setTimeout(), clearTimeout(), setInterval(), clearInterval(), typeof operator), and Libraries (require()). A green banner at the top right states 'Your board is connected! BeagleBone Black rev 0A5C S/N 2113BBBK2408 at 192.168.7.2'. The main heading is 'BeagleBone: open-hardware expandable computer' followed by 'Artist-tested, engineer approved' and a sub-heading 'The left-hand navigation bar will help you explore your board and learn how to program it.' Below this is a grid of 12 images showing various BeagleBone Black projects and hardware. At the bottom, it says 'Latest ARM open source focused on easy hardware experimentation' and lists 'Ships ready to use' including 'Angstrom Distribution with C++, Perl, Python, ...'.

3 **Update on board Operation System.** (This step can be skipped) Download the latest software from <http://beagleboard.org/latest-images>. Unzip it and write the image to SD card. Insert SD card into Beaglebone Black, hold USER/BOOT button when start Beaglebone Black. Wait until all 4 LEDs be lit solid. It can take up to 45 mins. For more detail, go to <http://beagleboard.org/Getting%20Started>.

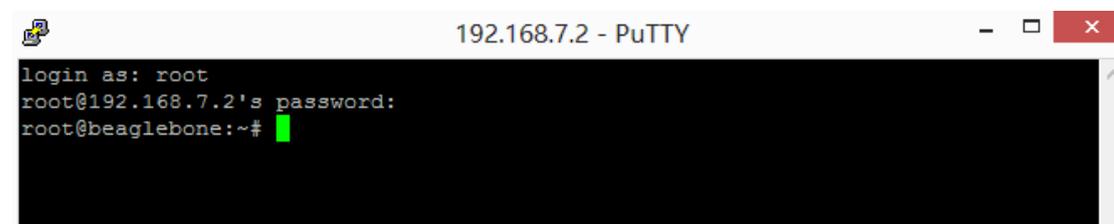
4 **SSH to Beaglebone Black.** Two ways to SSH to Beaglebone Black. One is using “GateOne SSH” on the webpage of step 2.

This screenshot is similar to the previous one but highlights the 'GateOne SSH' option in the 'Software' section of the navigation menu. The green status bar and main content area are the same as in the previous screenshot.

Other options is using third party SSH software such as PuTTY.



Use the username “root”, leave the password blank.



You already connected successfully to your Beaglebone Black. The system running on Beaglebone Black is a distribution of Linux called **Angstrom**. You can also install **Ubuntu** or **Debian** on board.

Python Library

Linux usually has build-in python. If we want to control all the GPIO using python, we need to find a GPIO library for python. There are several GPIO libraries on the internet for Python. PyBBIO and Adafruit_BBIO are most commonly used. In this application note, will talk about how to install Adafruit_BBIO. Internet are required for installation.

Adafruit_BBIO:

<http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/>

PyBBIO: <https://github.com/alexanderhiam/PyBBIO/wiki>

1 Install and Update Python. When you get SSH to Beaglebone Black, run the following command.

```
opkg update && opkg install python-pip python-setuptools python-smbus  
pip install Adafruit_BBIO
```

2 Test Installation of library. Run the following command.

```
python -c "import Adafruit_BBIO.GPIO as GPIO; print GPIO"
```

You should see something similar to the following

```
<module 'Adafruit_BBIO.GPIO' from '/usr/local/lib/python2.7/dist-packages/Adafruit_BBIO/GPIO.so'>
```

Up to here, the we already have the library required for GPIO.

Using GPIO, ADC, PWM in Python

Download Beaglebone Black System Reference Manual as a reference of the pin mode. https://github.com/CircuitCo/BeagleBone-Black/blob/master/BBB_SRM.pdf

1 GPIO. For using GPIO, you need to include following line in the front of your python program.

```
import Adafruit_BBIO.GPIO as GPIO
```

Output

```
GPIO.setup("Pin_Name", GPIO.OUT) #Pin_Name such as P8_29, or GPIO2_1  
GPIO.output("Pin_Name",GPIO.HIGH) #GPIO.HIGH for high or GPIO.LOW for low
```

Input

```
GPIO.setup("Pin_Name", GPIO.IN) #Pin_Name such as P8_29, or GPIO2_1  
Input = GPIO.input("Pin_Name") #Input = True when High, False when Low
```

Edge detect

```
GPIO.add_event_detect("Pin_Name", GPIO.RISING)  
# GPIO.RISING for rising edge, GPIO.FALLING) for Falling edge  
Detect = GPIO.event_detected("Pin_Name")  
#Detect = True when detect the rising or falling edge
```

More

```
| help(GPIO)      # for other command in GPIO library
```

2 PWM. For using PWM, you need to include following line in the front of your python program.

```
| import Adafruit_BBIO.PWM as PWM
```

Start/Stop

```
| PWM.start("Pin_Name", Duty_Cycle)      #Duty_Cycle from 0 to 100  
| PWM.stop("Pin_Name")
```

Set duty cycle or frequency

```
| PWM.set_duty_cycle("Pin_Name", Duty_Cycle)      #Duty_Cycle from 0 to 100  
| PWM.set_frequency("Pin_Name", Frequency)      #Frequency
```

3 ADC. Beaglebone Black has 7 ADC pin build in. Check System Reference Manual for Pin numbers. For using ADC, you need to include following line in the front of your python program.

```
| import Adafruit_BBIO.ADC as ADC  
| ADC.setup()
```

Read raw value from the pin

```
| Value = ADC.read_raw("Pin_Name")
```

Read voltage. ADC range from 0 to 1.8V

```
| Value = ADC.read("Pin_Name")      # the read function return a 0 to 1 value  
| Voltage = Value * 1.8
```

4 Adafruit_BBIO also has some other libraries for I2C, UART and SPI.

<http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/>