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 get a microcontroller to do all the controls and calculations. However microcontroller needs to involved 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, build in HDMI, also much more powerful processor. Both product can run Linux as the on board operation system, which means, you can running almost any language you want. This application 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 setup the Beaglebone Black if you have one in your hand. Beaglebone Black communicate with computers using USB cable. It need drivers to be installed on the computer. The reason using driver is because the driver setup an IP address for Beaglebone Black even though it only connected through USB. In that way you can SSH to Beaglebone just like connect to a Linux server using SSH.

1 Connect Beaglebone to PC. An extra drive called “BeagleBone Getting Started” appears in computer. Run the drivers under “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.

3 **Update on board Operation System.** (This step can be skipped) Download the latest software from [http://beagleboard.org/latest-images](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](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.
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:

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_BBBIO
```

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


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

```python
import Adafruit_BBIO.PWM as PWM
```

Start/Stop

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

Set duty cycle or frequency

```python
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.

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

Read raw value from the pin

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

Read voltage. ADC range from 0 to 1.8V

```python
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/