

LAB #5

Script Language, basic

Leonardo Fusser (1946995)

Objectives:

Cover the basics of script language

Procedure:

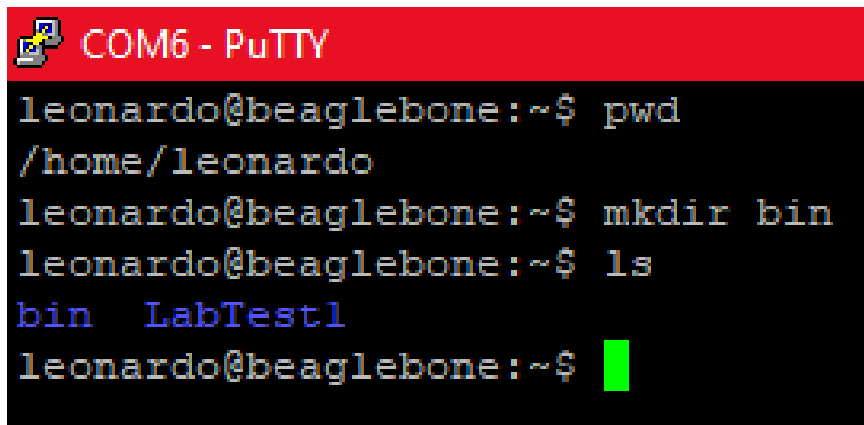
In each steps of your lab make screen shots and add to your report

Part 1

1. Login to BeagleBone as a user (not root)

Logged in as "leonardo".

2. Create a bin directory in your home directory



```
leonardo@beaglebone:~$ pwd
/home/leonardo
leonardo@beaglebone:~$ mkdir bin
leonardo@beaglebone:~$ ls
bin  LabTest1
leonardo@beaglebone:~$
```

Created bin folder in home directory

3. View your PATH variable.



```
leonardo@beaglebone:~$ pwd
/home/leonardo
leonardo@beaglebone:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games
leonardo@beaglebone:~$
```

PATH variable without newly created bin folder added to PATH

4. If not already done add the bin directory you just created to your PATH
5. Pause here and understand why you did this!

```
leonardo@beaglebone:~$ nano ~/.bashrc
GNU nano 2.2.6 File: /home/leonardo/.bashrc

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH="$HOME/bin:$PATH"

[ Wrote 112 lines ]
```

Adding newly created bin folder to PATH (using permanent method)

```
leonardo@beaglebone:~$ source ~/.bashrc
leonardo@beaglebone:~$ echo $PATH
/home/leonardo/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
/usr/games:/usr/local/games
leonardo@beaglebone:~$
```

PATH variable with newly created bin folder added to PATH

6. Create a new directory in your home to work in for today's lab. This is where you should be for the rest of the lab today!

COM6 - PuTTY

```
leonardo@beaglebone:~$ mkdir Lab5
leonardo@beaglebone:~$ ls
bin  Lab5  LabTest1
leonardo@beaglebone:~$
```

Lab5 folder created in home directory to create and test scripts before copying them to bin folder

Part 2

1. Create a script file that prints hello world to the screen.



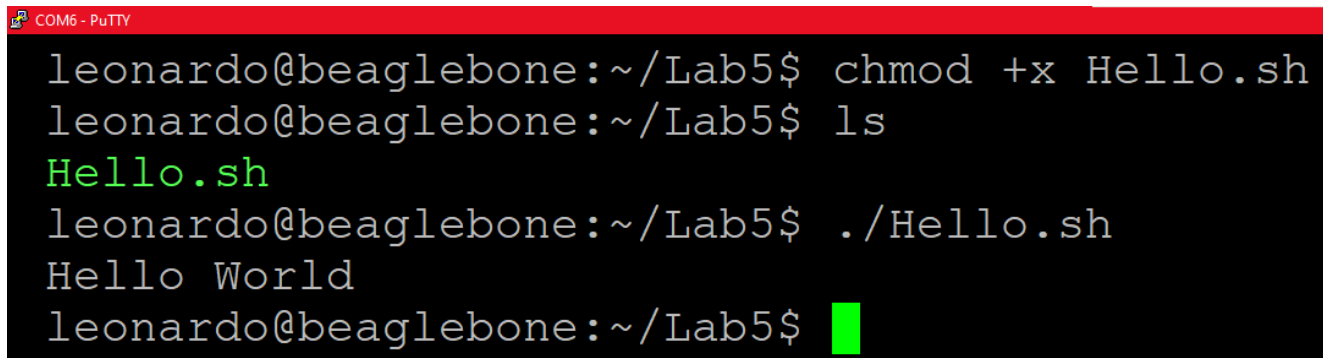
```
GNU nano 2.2.6 File: Hello_script.sh

#!/bin/sh
echo "Hello World"

[ Wrote 4 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Simple "Hello World" script created using nano editor

2. How do you Run it?



```
leonardo@beaglebone:~/Lab5$ chmod +x Hello.sh
leonardo@beaglebone:~/Lab5$ ls
Hello.sh
leonardo@beaglebone:~/Lab5$ ./Hello.sh
Hello World
leonardo@beaglebone:~/Lab5$
```

Making script created above as an executable (shown in green) and testing script from Lab5 folder

3. Copy it in your bin folder. Now how do you run your command now? Change directory and run the command.

```
leonardo@beaglebone:~/Lab5$ pwd
/home/leonardo/Lab5
leonardo@beaglebone:~/Lab5$ ls
Hello.sh
leonardo@beaglebone:~/Lab5$ cp Hello.sh /home/leonardo/bin
leonardo@beaglebone:~/Lab5$ cd /home/leonardo/bin
leonardo@beaglebone:~/bin$ pwd
/home/leonardo/bin
leonardo@beaglebone:~/bin$ ls
Hello.sh
leonardo@beaglebone:~/bin$ ./Hello.sh
Hello World
leonardo@beaglebone:~/bin$
```

*Copied script (shown in green) from Lab5 folder to bin folder created before and testing script
(Not shown: script was also tested from home directory by doing: Hello.sh)*

4. Create a new script file. In it, print to the screen your HOME PATH USER \$0 variables

```
leonardo@beaglebone:~/Lab5$ nano Print.sh
GNU nano 2.2.6 File: Print.sh

#!/bin/sh

echo $HOME
echo $PATH
echo $USER
echo $0

[ Read 6 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
```

Script created to print 4 linux environmental variables using nano editor

5. Test your script and then place it in your bin directory

```
leonardo@beaglebone:~/Lab5$ chmod +x Print.sh
leonardo@beaglebone:~/Lab5$ ls
Hello.sh  Print.sh
leonardo@beaglebone:~/Lab5$ ./Print.sh
/home/leonardo
/home/leonardo/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
/usr/games:/usr/local/games
leonardo
./Print.sh
leonardo@beaglebone:~/Lab5$
```

Making script executable and testing script from Lab5 folder created before

```
leonardo@beaglebone:~/Lab5$ pwd
/home/leonardo/Lab5
leonardo@beaglebone:~/Lab5$ ls
Hello.sh  Print.sh
leonardo@beaglebone:~/Lab5$ cp Print.sh /home/leonardo/bin
leonardo@beaglebone:~/Lab5$ cd /home/leonardo/bin
leonardo@beaglebone:~/bin$ pwd
/home/leonardo/bin
leonardo@beaglebone:~/bin$ ls
Hello.sh  Print.sh
leonardo@beaglebone:~/bin$ ./Print.sh
/home/leonardo
/home/leonardo/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
/usr/games:/usr/local/games
leonardo
./Print.sh
leonardo@beaglebone:~/bin$
```

*Copied script (shown in green) from Lab5 folder to bin folder created before and testing script
(Not shown: script was also tested from home directory by doing: Print.sh)*

Part 3

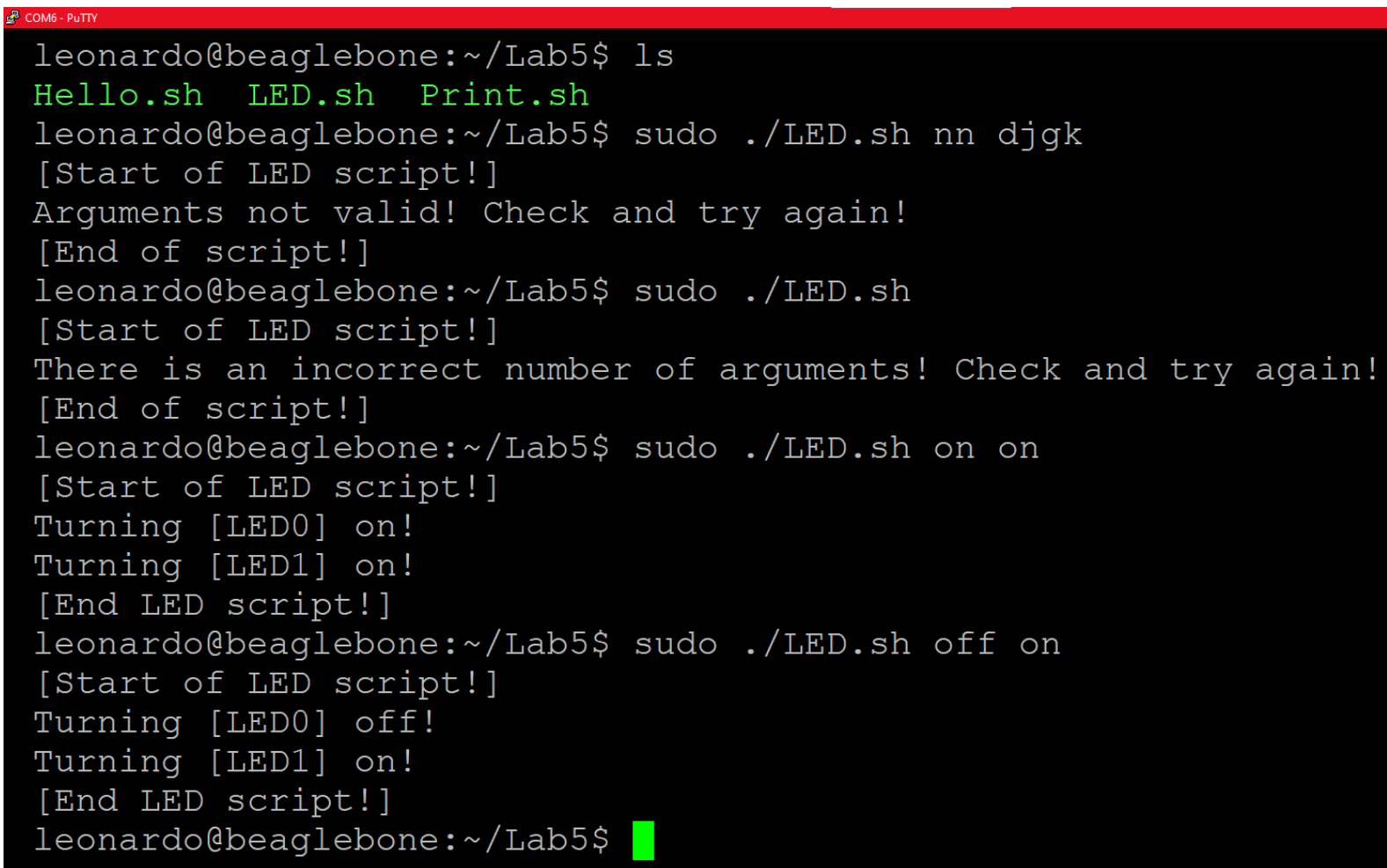
1. Now create a script file that will automatically turn on and off usr0 **OR** usr1 LED depending on the arguments provided by the user. Also, you must provide a proper “usage” message.

Example:

```
root@beaglebone:/home/project# ./bashUSR 0 on
Starting the LED Bash Script
Turning ON usr0 LED

root@beaglebone:/home/project# ./bashUSR 1 off
Starting the LED Bash Script
Turning OFF usr1 LED

root@beaglebone:/home/project# ./bashUSR 9 off
Starting the LED Bash Script
First argument is wrong
Usage is bashUSR arg1 arg2
where arg1 is one of 0,1,2 or 3
and arg2 is one of on or off.
e.g. bashUSR 1 on
```

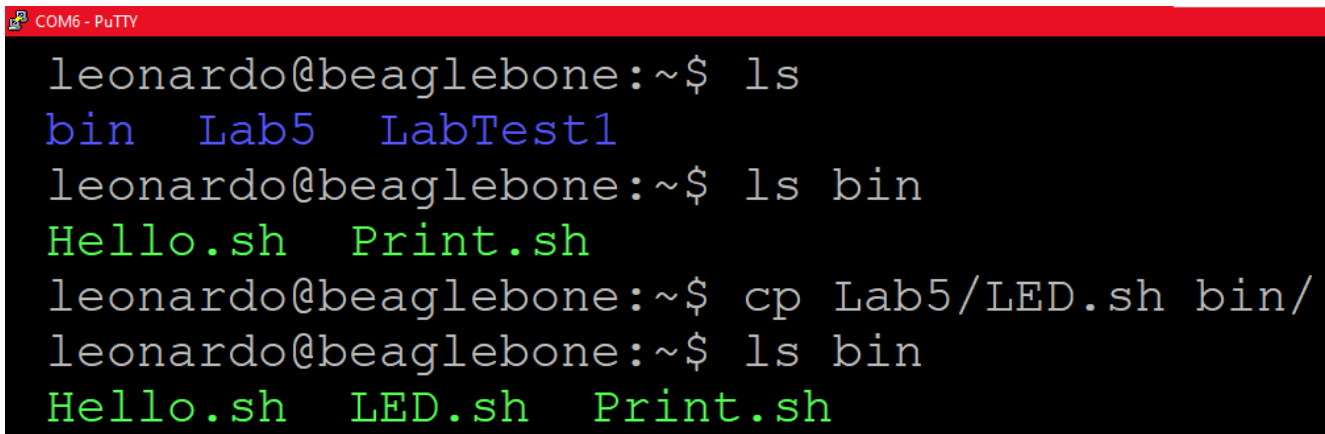


```
COM6 - PuTTY
leonardo@beaglebone:~/Lab5$ ls
Hello.sh LED.sh Print.sh
leonardo@beaglebone:~/Lab5$ sudo ./LED.sh nn djgk
[Start of LED script!]
Arguments not valid! Check and try again!
[End of script!]
leonardo@beaglebone:~/Lab5$ sudo ./LED.sh
[Start of LED script!]
There is an incorrect number of arguments! Check and try again!
[End of script!]
leonardo@beaglebone:~/Lab5$ sudo ./LED.sh on on
[Start of LED script!]
Turning [LED0] on!
Turning [LED1] on!
[End LED script!]
leonardo@beaglebone:~/Lab5$ sudo ./LED.sh off on
[Start of LED script!]
Turning [LED0] off!
Turning [LED1] on!
[End LED script!]
leonardo@beaglebone:~/Lab5$
```

Script tested from Lab5 folder using different possible arguments inputted by the user

(Not shown: script was also tested from home directory by doing: sudo bin/LED.sh \$1 \$2)

2. Test your script and then place it in your bin directory.



A terminal window titled 'COM6 - PuTTY' showing a user named 'leonardo' at a 'beaglebone' prompt. The user runs 'ls' in their home directory, showing 'bin', 'Lab5', and 'LabTest1'. Then, they run 'ls bin', showing 'Hello.sh' and 'Print.sh'. Next, they run 'cp Lab5/LED.sh bin/'. Finally, they run 'ls bin' again, showing 'Hello.sh', 'LED.sh', and 'Print.sh'. The script names are highlighted in green in the original image.

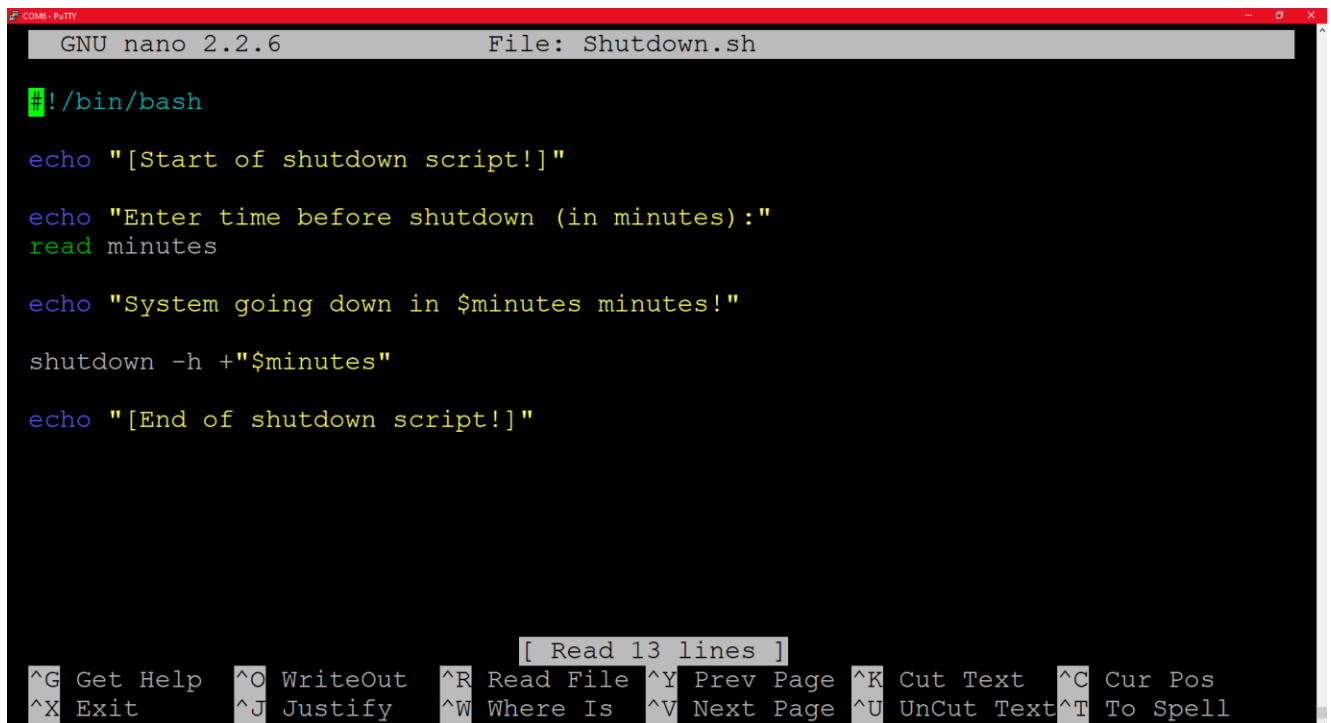
```
leonardo@beaglebone:~$ ls
bin  Lab5  LabTest1
leonardo@beaglebone:~$ ls bin
Hello.sh  Print.sh
leonardo@beaglebone:~$ cp Lab5/LED.sh bin/
leonardo@beaglebone:~$ ls bin
Hello.sh  LED.sh  Print.sh
```

Script (shown in green) copied from Lab5 folder to bin folder created before (while in home directory)

*Script was created to turn on/off one or two LEDs based on user arguments inputted using nano. Script was also made executable and permissions were set for full control for all users/groups. See LED.sh script attached to this submission.

Part 4

3. Create a new script file to shutdown -h time where time is inputted from command line.



The screenshot shows a terminal window with the nano 2.2.6 text editor open. The file being edited is named 'Shutdown.sh'. The script content is as follows:

```
#!/bin/bash

echo "[Start of shutdown script!]"

echo "Enter time before shutdown (in minutes):"
read minutes

echo "System going down in $minutes minutes!"

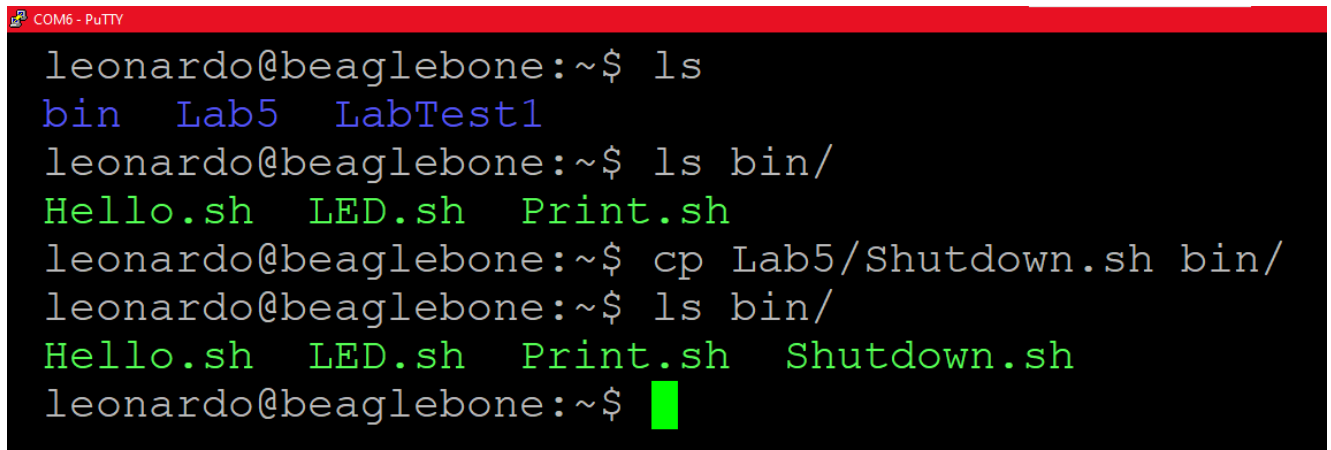
shutdown -h +"$minutes"

echo "[End of shutdown script!]"
```

The bottom status bar of the nano editor shows the following keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, and ^T To Spell. A message '[Read 13 lines]' is also visible.

Simple shutdown script created using nano

4. Test your script and then place it in your bin directory.



The screenshot shows a terminal window with the following commands and output:

```
leonardo@beaglebone:~$ ls
bin  Lab5  LabTest1
leonardo@beaglebone:~$ ls bin/
Hello.sh  LED.sh  Print.sh
leonardo@beaglebone:~$ cp Lab5/Shutdown.sh bin/
leonardo@beaglebone:~$ ls bin/
Hello.sh  LED.sh  Print.sh  Shutdown.sh
leonardo@beaglebone:~$
```

Script (shown in green) tested from Lab5 folder and home folder then copied to bin folder created before

5. Restart your BeagleBone and test all your script files.

Demo

All scripts created for this lab are attached to this submission for your convenience.