The purpose of this lab is to introduce SSH (Secure Shell) protocol, VNC server and Analog input in BBB.

BEAGLEBONE BLACK PCB layout:

- Processor: AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
Problem 1. Connect and control BBB board in PuTTY using SSH protocol.

- Connect Power Adapter to Board and start PC.
- Once the PC start, go to my computer, and then install drivers using following path: BeagleBone Getting Started/drivers/windows/bone_drv.
- Go to your browser and enter this IP address 192.168.7.2 then you will see that BBB Board is connected.
- Type 192.168.7.2:3000 and open Cloud9 IDE editor in which we write and run code.
- Now open PuTTY/SSH CLIENT, write IP address of board and connect your board to PuTTY via SSH.
- After open it, we have to run following code to see how much space available in on chip eMMC flash memory which is around 4 GB.
  1. df -h
  2. du -h

Expected Output:

- The command df -h show total memory usage and available while du -h show the memory of particular directory.

Problem 2: Create VNC server and access your board from another computer

Procedure:

- First open PuTTY and login as debian instead of root and enter password temppwd.
- Write ifconfig and note down Ethernet address.
- Now write vncserver in PuTTY and open vncserver
- Then enter IP address of board then enter password 123456 so you can see Beaglebone board’s desktop.
- Now open CHROMIUM browser and open www.daiict.ac.in website.
- Now you can access your board’s desktop from other PC by opening vncserver and entering Ethernet address.
Problem 3: Reading Analog Voltage from Potentiometer

Note: Don’t give more than 1.8V to analog pin because it can damage board.

Software: Use the Cloud9 IDE to write the following Python code to view Analog input values. Your code should print the following,

- If input voltage is less than equal to 500 mV: Print “Normal”
- If input voltage is greater than 500 mV and less than equal to 1000 mV: Print “Above normal”
- If input voltage is greater than 1000 mV and less than equal to 1800 mV: Print “Danger”

Reference code:

```python
import Adafruit_BBIO.ADC as ADC
import time

sensor_pin='P9_40'
ADC.setup()

while True:
    reading=ADC.read(sensor_pin)
    millivols=reading*1800
    print('mv=%d  %d(millivols))
    time.sleep(1)
```
Connections: The potentiometer is connected using pins 32 (1.8 V supply), 34 (GND) and 40 (Analog input 1)

Output:
It will give output voltage ranging from 50 mV to 1750 mV.

Problem4. LED Blinking using GPIO of Beaglebone Black
Code:
```python
import Adafruit_BBIO.GPIO as GPIO
import time
GPIO.setup("P8_10", GPIO.OUT)
while True:
    GPIO.output("P8_10", GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output("P8_10", GPIO.LOW)
    time.sleep(0.5)
```