The purpose of this lab is to give the introduction of socket programming and to show how to setup a communication link between two Beagle Bone Blacks over the Ethernet interface.

**Beagle Bone Black board:**

![Beagle Bone Black board diagram]

**Socket-based Communication**

Sockets provide an interface for programming networks at the transport layer. Network communication using Sockets is very much similar to performing file I/O. In fact, socket handle is treated like file handle.

The streams used in file I/O operation are also applicable to socket-based I/O. Socket-based communication is independent of a programming language used for implementing it. That means, a socket program written in Java language can communicate to a program written in C or C++ socket program. A server runs on a specific computer and has a socket that is bound to a specific port. The server listens to the socket for a client to make a connection request. If
everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket so that it can continue to listen to the original socket for connection requests while serving the connected client.

**Problem statement:**
To implement socket programming in Python and in the process setup a communication link between two Beagle bone Blacks over the Ethernet interface.

Refer to the following link on socket programming:
https://docs.python.org/2/howto/sockets.html

Note: Questions will be asked after the lab regarding this.

The following are samples of basic code at the server and client side:

**At the server:**
```python
#!/usr/bin/python
import socket
s = socket.socket()          # Import socket module
s.bind((host, port))         # Create a socket object
# Get local machine name
s.listen(5)                  # Reserve a port for your service.
s.bind((host, port))         # Bind to the port
s.listen(5)                  # Now wait for client connection.
while True:
    (c, addr) = s.accept()   # Establish connection with client.
    print 'Got connection from', addr
    c.send('Thank you for connecting')
    c.close()                # Close the connection
```
At the client:

```python
#!/usr/bin/python
import socket
s = socket.socket() # Import socket module
# Create a socket object
host = socket.gethostname() # Get local machine name
port = 12346 # Reserve a port for your service.
s.connect((host, port))
print s.recv(1024)
s.close() # Close the socket when done
```

Run the client code after server code on separate terminals on your Beaglebone Black.

Note down the outputs observed and the reason in your own words.
Note down the various functions used and their significance.

Try connecting to another BBB over the network using its (BBB’s) IP address by setting `host = “ip address of BBB you want to connect to”` in the client script.

Now modify the program so that analog value measured by the server (BBB running the server script) is displayed on the terminal of the client connected to the server. (According to the Problem 3: Reading Analog Voltage from Potentiometer from lab_3)

Note down the modified client and server program in your respective books.
(Use connection of the previous practical which involved using a potentiometer)