**Send Data from one Raspberry pi to another Raspberry Pi [Lab 3]**

**Objective:**

In this Experiment we shall learn to create our own server and client with raspberyy pi uisng python Socket Programming.

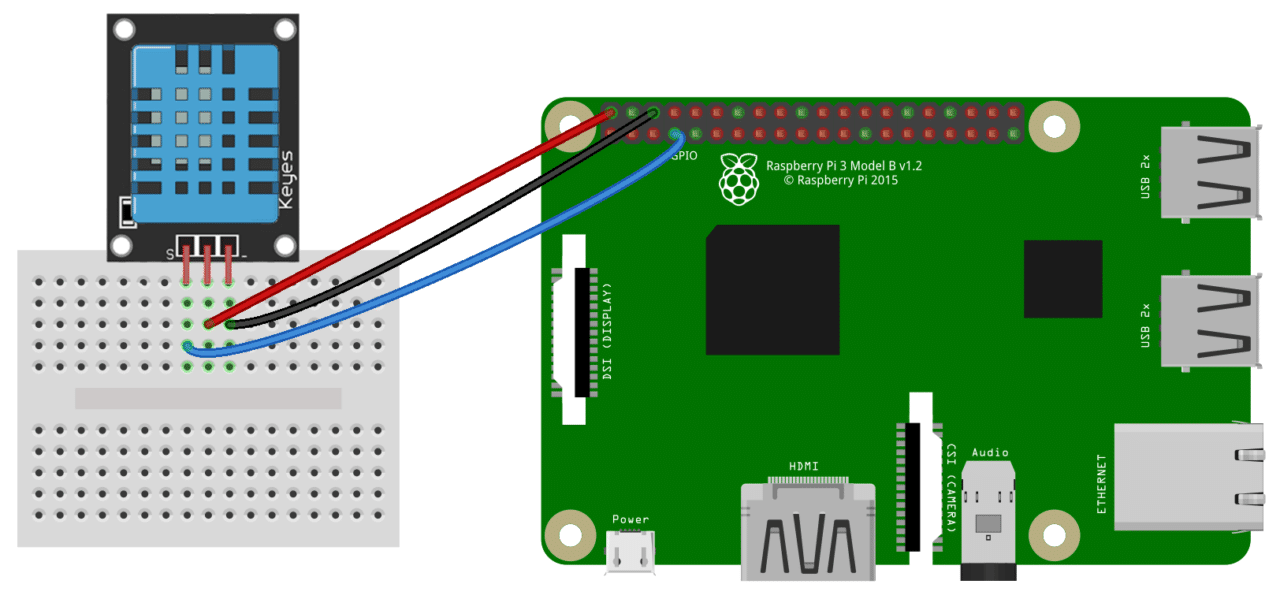
Our Target is to create a server. the server would be Running 24x7 on Raspberry pi and server is always listening for request from client When a client request server it shall send the Temperature and Humidity over Network using Socket Programming

**Tutorial:**

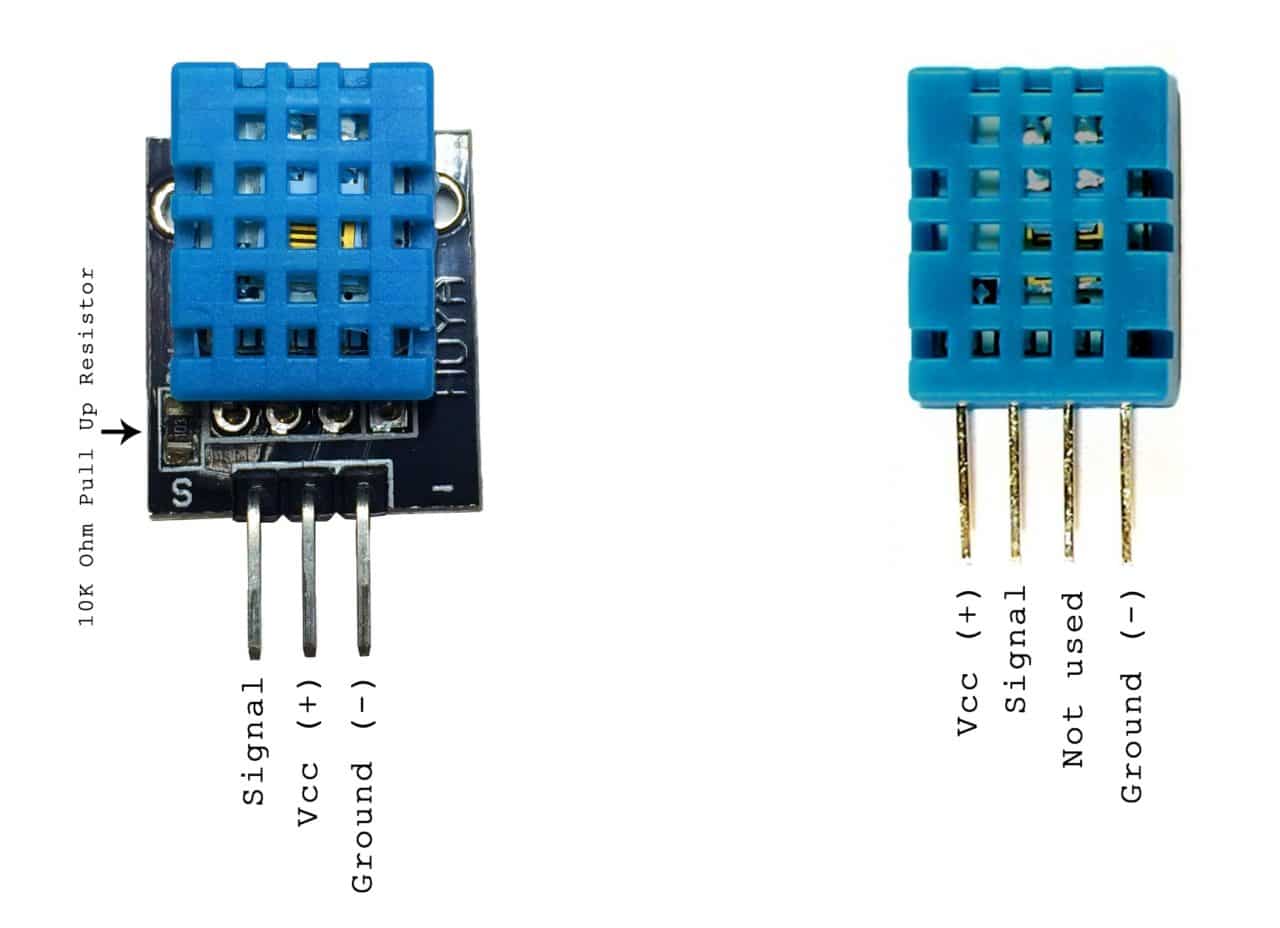
https://www.youtube.com/watch?v=QihjI84Z2tQ

****

### Connection Diagram



### DHT Pinout



# **Step 1: Create Server:**

Define the Library

import socket

import numpy as np

import encodings

Define the HOST and PORT

HOST = '127.0.0.1' # Standard loopback interface address (localhost)

PORT = 65432 # Port to listen on (non-privileged ports are > 1023)

For the Experiment we shall send Dummy Data and then once we know the concept we can send actual Sensor value over the socket programming.

### Define the function which will send dummy data when client sends request

### def random\_data():

### x1 = np.random.randint(0, 55, None) # Dummy temperature

### y1 = np.random.randint(0, 45, None) # Dummy humidigy

### my\_sensor = "{},{}".format(x1,y1)

### return my\_sensor

### Define the actual server when client sends string Data it shall send Data

def my\_server():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

print("Server Started waiting for client to connect ")

s.bind((HOST, PORT))

s.listen(5)

conn, addr = s.accept()

with conn:

print('Connected by', addr)

while True:

data = conn.recv(1024).decode('utf-8')

if str(data) == "Data":

print("Ok Sending data ")

my\_data = random\_data()

x\_encoded\_data = my\_data.encode('utf-8')

conn.sendall(x\_encoded\_data)

elif str(data) == "Quit":

print("shutting down server ")

break

if not data:

break

else:

pass

if \_\_name\_\_ == '\_\_main\_\_':

while 1:

my\_server()

### Explanation[¶](http://localhost:8888/notebooks/Desktop/Important%20Documents%20/IoT%20final/Combined/Lab%203/Lab%203.ipynb#Explanation)

when client sends Data as a string server will send sensor data over the network

# **Entire Code for Server**

import socket

import numpy as np

import encodings

HOST = '127.0.0.1' # Standard loopback interface address (localhost)

PORT = 65432 # Port to listen on (non-privileged ports are > 1023)

def random\_data():

x1 = np.random.randint(0, 55, None) # Dummy temperature

y1 = np.random.randint(0, 45, None) # Dummy humidigy

my\_sensor = "{},{}".format(x1,y1)

return my\_sensor # return data seperated by comma

def my\_server():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

print("Server Started waiting for client to connect ")

s.bind((HOST, PORT))

s.listen(5)

conn, addr = s.accept()

with conn:

print('Connected by', addr)

while True:

data = conn.recv(1024).decode('utf-8')

if str(data) == "Data":

print("Ok Sending data ")

my\_data = random\_data()

x\_encoded\_data = my\_data.encode('utf-8')

conn.sendall(x\_encoded\_data)

elif str(data) == "Quit":

print("shutting down server ")

break

if not data:

break

else:

pass

if \_\_name\_\_ == '\_\_main\_\_':

while 1:

my\_server()

# **Step 2: Write the Code for Client**

Define the Library

import socket

import numpy as np

import encodings

Define the HOST and PORT

HOST = '192.168.0.111' # The server's hostname or IP address

PORT = 65432 # The port used by the server

### we need to process the Data that server sends us lets us define a function for that[¶](http://localhost:8888/notebooks/Desktop/Important%20Documents%20/IoT%20final/Combined/Lab%203/Lab%203.ipynb#we-need-to-process-the-Data-that-server-sends-us-lets-us-define-a-function-for-that)

def process\_data\_from\_server(x):

x1, y1 = x.split(",")

return x1,y1

### Write the Client Code:

### def my\_client():

### threading.Timer(11, my\_client).start()

### with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

### s.connect((HOST, PORT))

### my = input("Enter command ")

### #my = "Data"

### my\_inp = my.encode('utf-8')

### s.sendall(my\_inp)

### data = s.recv(1024).decode('utf-8')

### x\_temperature,y\_humidity = process\_data\_from\_server(data)

### print("Temperature {}".format(x\_temperature))

### print("Humidity {}".format(y\_humidity))

### s.close()

### time.sleep(5)

if \_\_name\_\_ == "\_\_main\_\_":

while 1:

my\_client()

**Entire Client Code**

import socket

import threading

import time

HOST = '192.168.0.111' # The server's hostname or IP address

PORT = 65432 # The port used by the server

def process\_data\_from\_server(x):

x1, y1 = x.split(",")

return x1,y1

def my\_client():

threading.Timer(11, my\_client).start()

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.connect((HOST, PORT))

my = input("Enter command ")

#my = "Data"

my\_inp = my.encode('utf-8')

s.sendall(my\_inp)

data = s.recv(1024).decode('utf-8')

x\_temperature,y\_humidity = process\_data\_from\_server(data)

print("Temperature {}".format(x\_temperature))

print("Humidity {}".format(y\_humidity))

s.close()

time.sleep(5)

if \_\_name\_\_ == "\_\_main\_\_":

while 1:

my\_client()

**part II**

# **Great Job try the Code and lets send Actual Sensor Data over Network**[**¶**](http://localhost:8888/notebooks/Desktop/Important%20Documents%20/IoT%20final/Combined/Lab%203/Lab%203.ipynb#Great-Job-try-the-Code-and-lets-send-Actual-Sensor-Data-over-Network)

# **Server Code which will run on Raspberry pi**

import socket

import numpy as np

import encodings

import Adafruit\_DHT

HOST = '192.168.1.4' # Standard loopback interface address (localhost)

PORT = 65432 # Port to listen on (non-privileged ports are > 1023)

def random\_data():

pin = 4

sensor = Adafruit\_DHT.DHT22

humidity, temperature = Adafruit\_DHT.read\_retry(sensor, pin)

if humidity is not None and temperature is not None:

print('Temp={0:0.1f}\*C Humidity={1:0.1f}%'.format(temperature, humidity))

print("data was written on database T{} H{}".format(temperature,humidity))

data = '{},{}'.format(temperature,humidity)

return data

def my\_server():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

print("Server Started waiting for client to connect ")

s.bind((HOST, PORT))

s.listen(5)

conn, addr = s.accept()

with conn:

print('Connected by', addr)

while True:

data = conn.recv(1024).decode('utf-8')

if str(data) == "Data":

print("Ok Sending data ")

my\_data = random\_data()

x\_encoded\_data = my\_data.encode('utf-8')

conn.sendall(x\_encoded\_data)

elif str(data) == "Quit":

print("shutting down server ")

break

if not data:

break

else:

pass

if \_\_name\_\_ == '\_\_main\_\_':

while 1:

my\_server()

# Client Code will be same as Above.