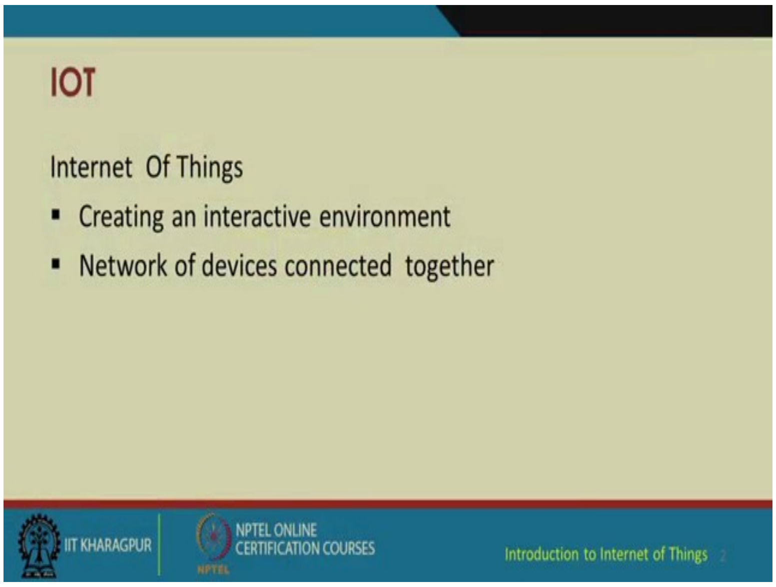
Introduction to Internet of Things

Prof. Sudip Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Implementation of IoT with Raspberry Pi – II

Hi, in this lecture of the second part of implementation of IoT with raspberry pi we are going to discuss the same integration we discussed previously that is integration of a DHT sensor with raspberry pi. Additionally we are going to include some networking components to it that is we are going to implement a python scripting for client making the raspberry pi behave as a client whereas, on the remote desktop which will act as a server will use another python script which will act as a UDP server. So, in this way we are going to create a datagram based connection between the raspberry pi client and the desktop based server.

(Refer Slide Time: 01:09)



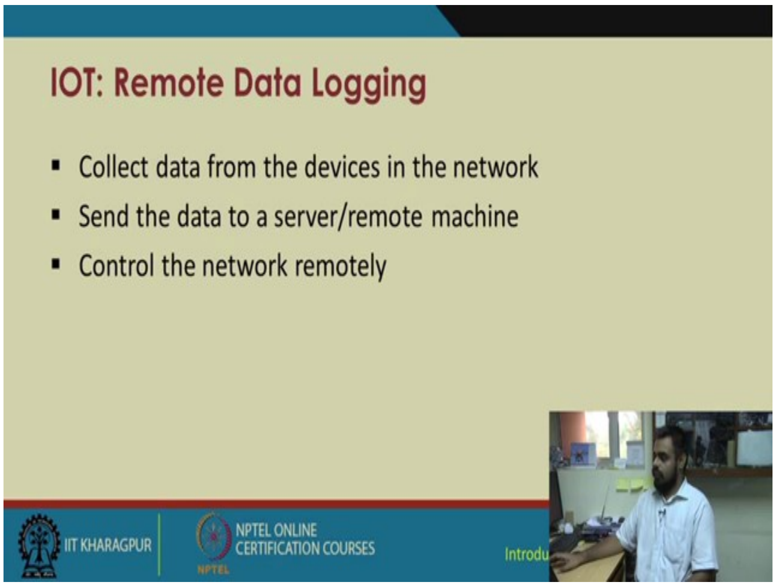
IOT

Internet Of Things
- Creating an interactive environment
- Network of devices connected together

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

So, again basic aim of this internet of things is to create an interactive environment of network devices which are all connected together additionally your devices more the number of devices.

(Refer Slide Time: 01:20)



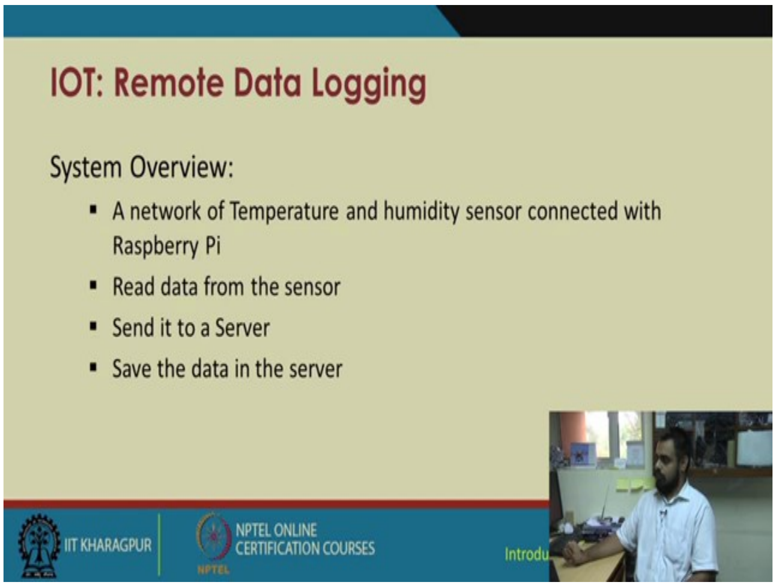
IOT: Remote Data Logging

- Collect data from the devices in the network
- Send the data to a server/remote machine
- Control the network remotely

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Introdu

The more intelligent and interactive your whole system should be it should not be that you are increasing your device number of devices whereas; your system performance is going down. So, by maintaining your system performance the device integration has to be made. So, I am sure the various strategies have been covered in the previous theoretical lectures in this one we are going to give a basic demonstration of whatever you have learnt in your previous lectures a very rudimentary demonstration.

(Refer Slide Time: 02:29)



IOT: Remote Data Logging

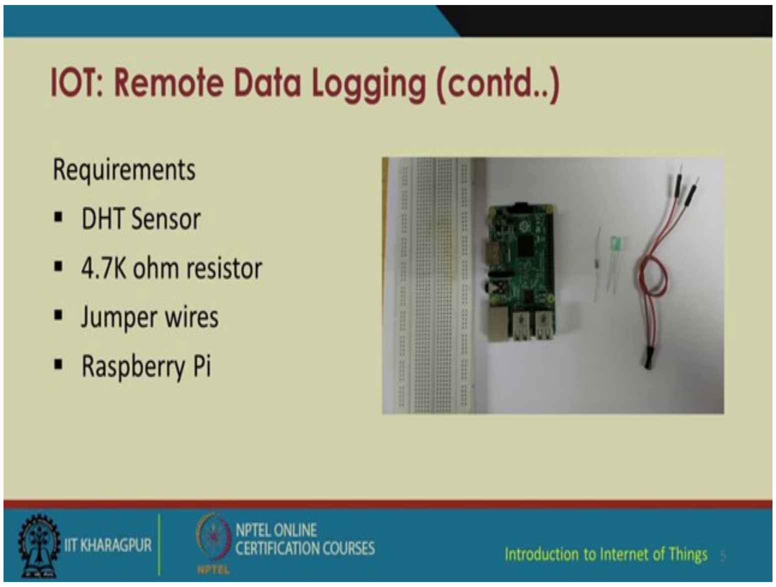
System Overview:

- A network of Temperature and humidity sensor connected with Raspberry Pi
- Read data from the sensor
- Send it to a Server
- Save the data in the server

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Introdu

So, in this second lecture we will be focus more on remote data logging. So, it will be collecting data from the devices in the network, it will send the data to a server or a remote machine and that remote machine will the remote machine will control the client or the data sending device remotely. So, the system overview is as I have told you previously and network of temperature and humidity sensors will be connected to raspberry pi. Here for the sake of demonstration we are only connecting one temperature and humidity sensor with single raspberry pi data from the sensors are read it is sent to the server over the network and the data is saved in the server.

(Refer Slide Time: 02:51)



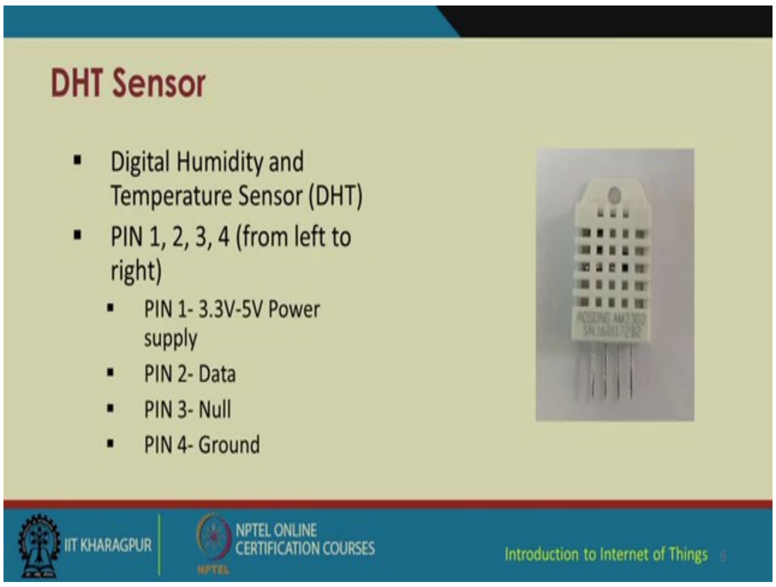
IOT: Remote Data Logging (contd..)

Requirements

- DHT Sensor
- 4.7K ohm resistor
- Jumper wires
- Raspberry Pi

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 5

(Refer Slide Time: 03:20)



DHT Sensor

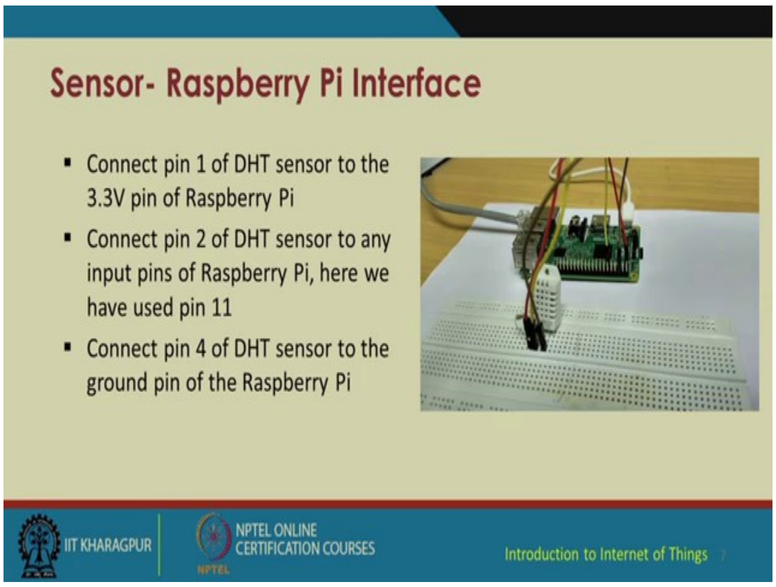
- Digital Humidity and Temperature Sensor (DHT)
- PIN 1, 2, 3, 4 (from left to right)
  - PIN 1- 3.3V-5V Power supply
  - PIN 2- Data
  - PIN 3- Null
  - PIN 4- Ground

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 6

So, same as before we have the same requirements a raspberry pi a DHT sensor a 4.0 kilo ohm resistor it is to be noted that for raspberry pi this 4.7 kilo ohm resistor must be obtain between the BCC and the data pin that is between pin one and 2 additionally we will be requiring jumper wires to make our connections again for recapitulation well go about discussing the DHT.

So, from left to right if you keep the grid side up; you have pins numbered 1 to 4 first being the Vcc pin 4th being the ground and the second pin is used for transmitting data.

(Refer Slide Time: 03:40)



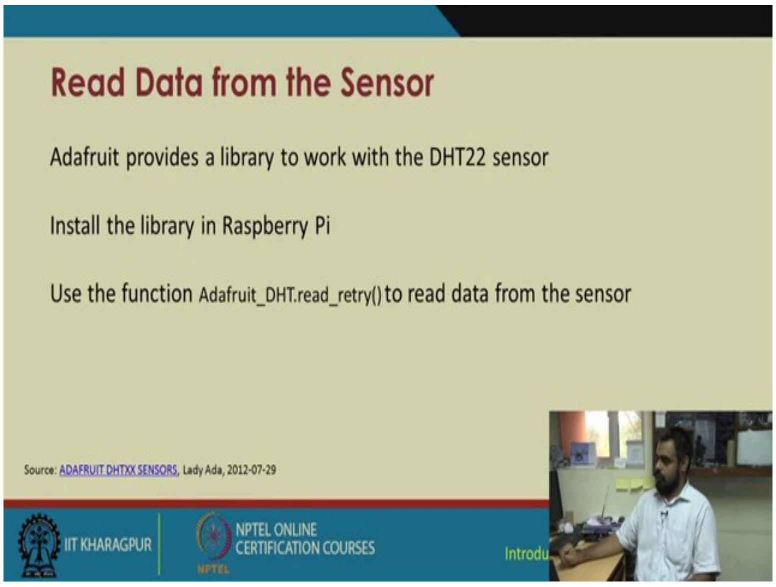
Sensor- Raspberry Pi Interface

- Connect pin 1 of DHT sensor to the 3.3V pin of Raspberry Pi
- Connect pin 2 of DHT sensor to any input pins of Raspberry Pi, here we have used pin 11
- Connect pin 4 of DHT sensor to the ground pin of the Raspberry Pi

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

So, the connection to the raspberry pi will be as follows from the previous lecture we have just debuted a little bit you can also use the same configuration as done in the previous lectures, but remember the board configuration for this DHT sensor must be in BCM mode not in board mode. So, we are connecting the DHT sensor to 3.3 volt pin of raspberry pi we are connecting the data pin or pin 2 of the DHT sensor to pin 11 of raspberry pi this pin 11 is in BCM mode and we are connecting pin 4 of DHT sensor to the ground pin of raspberry pi pin.

(Refer Slide Time: 04:26)



Read Data from the Sensor

Adafruit provides a library to work with the DHT22 sensor

Install the library in Raspberry Pi

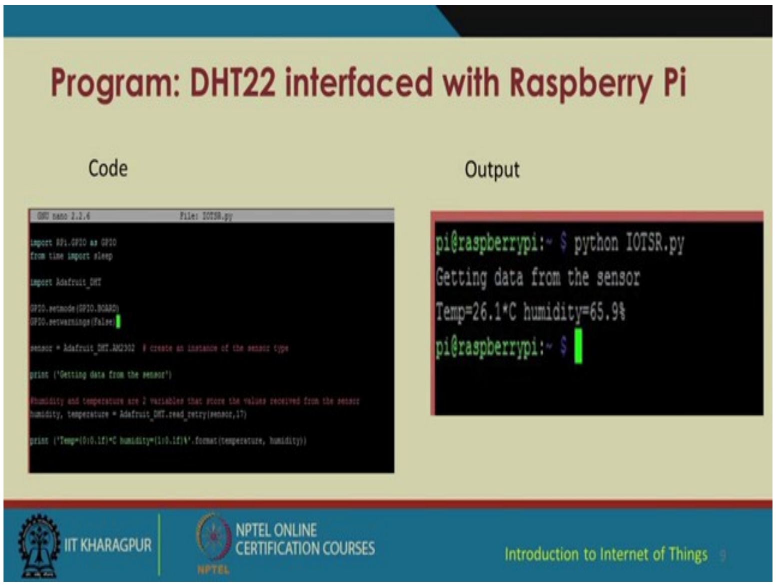
Use the function Adafruit_DHT.read_retry() to read data from the sensor

Source: ADAFRUIT DHTXX SENSORS, Lady Ada, 2012-07-29

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Introdu

Now, as in the previous lectures we have already covered this Adafruit which supplies this DHT 22 sensor they also provide a library to integrate this sensor to raspberry pi via python frame work. So, this has already been installed during our previous lecture. Now we are going to call the Adafruit_DHT.read_retry() function to read data from the sensor.

(Refer Slide Time: 05:01)



Program: DHT22 interfaced with Raspberry Pi

Code

```
GNU nano 2.2.6                    File: IOTSR.py

import RPi.GPIO as GPIO
from time import sleep

import Adafruit_DHT

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

sensor = Adafruit_DHT.AM2302  # create an instance of the sensor type

print ('Getting data from the sensor')

#humidity and temperature are 2 variables that store the values received from the sensor
humidity, temperature = Adafruit_DHT.read_retry(sensor,17)

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

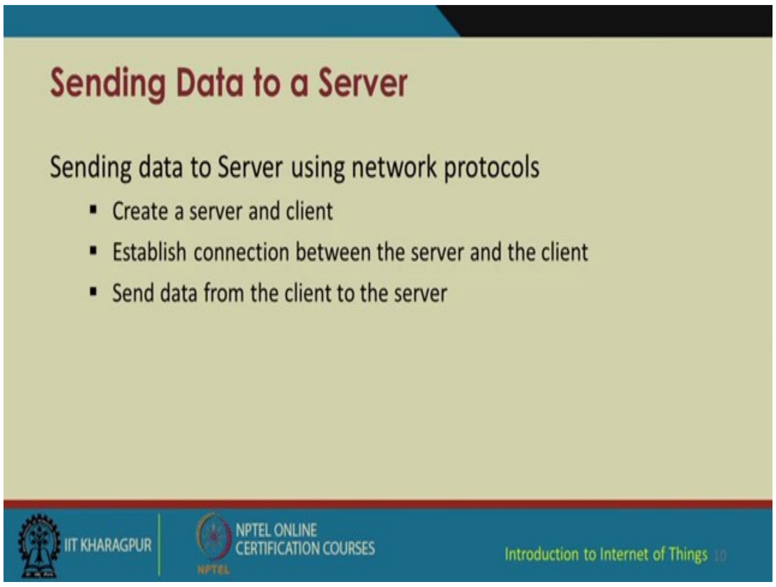
Output

```
pi@raspberrypi:~ $ python IOTSR.py
Getting data from the sensor
Temp=26.1*C humidity=65.9%
pi@raspberrypi:~ $
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Introduction to Internet of Things

So, as you can see the codes and the outputs or more or less similar to the previous one.

(Refer Slide Time: 05:16)



Sending Data to a Server

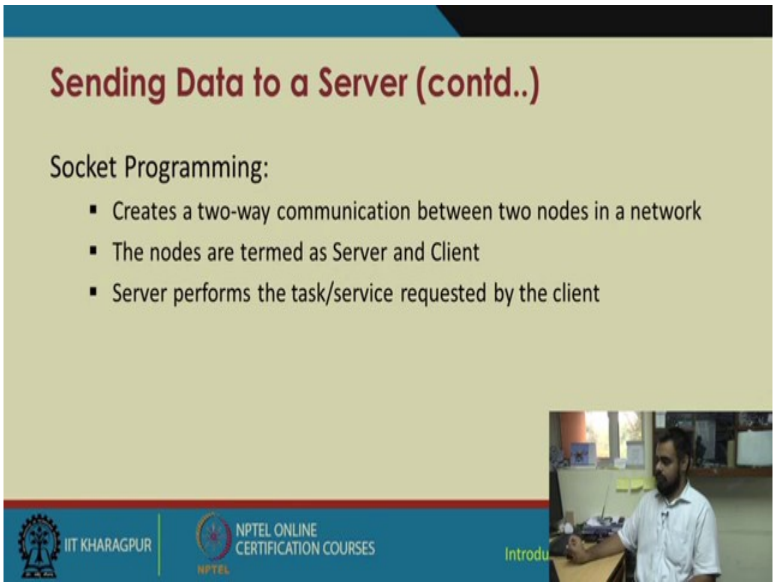
Sending data to Server using network protocols

- Create a server and client
- Establish connection between the server and the client
- Send data from the client to the server

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 10

We are going to read the temperature only the additionally you can attach your humidity readings also, but for the sake of testing you can just go with the temperature readings. Now for sending the data over the network we will be using a client server client server model and to establish the connection you are going to take the help of a UDP based socket will make a UDP based socket which will send data from client to the server now for this socket programming.

(Refer Slide Time: 05:51)



Sending Data to a Server (contd..)

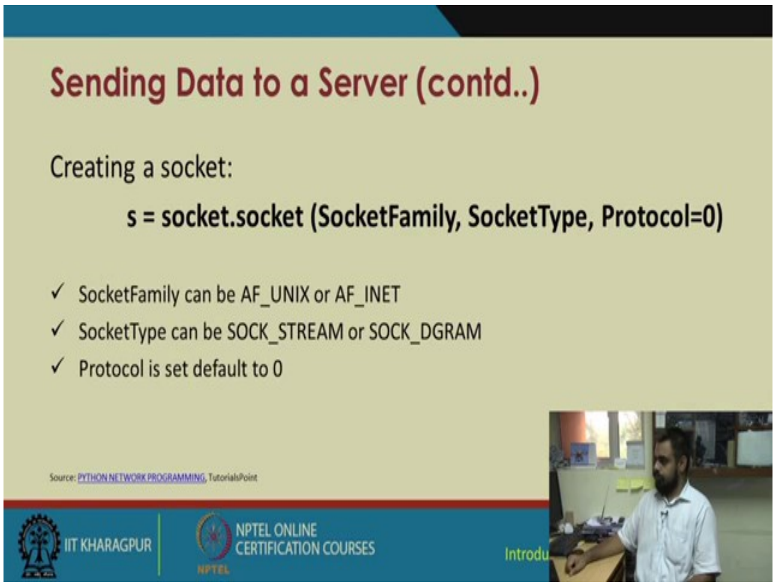
Socket Programming:

- Creates a two-way communication between two nodes in a network
- The nodes are termed as Server and Client
- Server performs the task/service requested by the client

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introdu

This generally sockets create a 2 way communication between 2 nodes in a network, 1 node is termed as the server whereas, the other as is termed as the client the server performs the task or the service as requested by the client.

(Refer Slide Time: 06:11)



Sending Data to a Server (contd..)

Creating a socket:

s = socket.socket (SocketFamily, SocketType, Protocol=0)

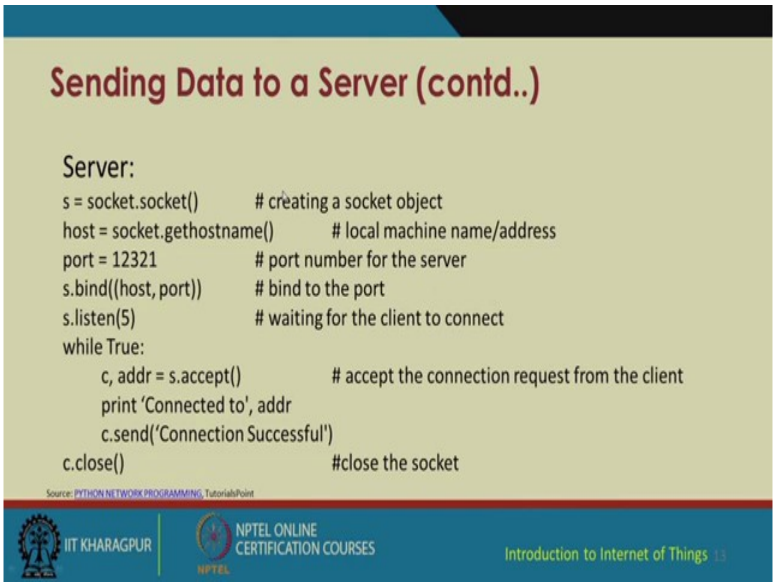
- ✓ SocketFamily can be AF_UNIX or AF_INET
- ✓ SocketType can be SOCK_STREAM or SOCK_DGRAM
- ✓ Protocol is set default to 0

Source: PYTHON NETWORK PROGRAMMING, TutorialsPoint

So, how do you proceed by creating a socket on raspberry pi this was also covered in the previous lectures on introduction to raspberry pi. So, it is quite simple to assign the socket.socket function to a variable and you manipulate that variable.

So, this socket.socket function it has three arguments first is SocketFamily second is SocketType and third is Protocol which is by default zero socket family can be either AF_UNIX or AF_INET. So, UNIX as you recall it is mainly for on the UNIX base systems whereas, for most of the general purpose programming we are going with AF_INET or internet protocols socket type can be either Socket_Stream or Socket_data gram.

(Refer Slide Time: 07:10)



Sending Data to a Server (contd..)

Server:

```
s = socket.socket()          # creating a socket object
host = socket.gethostname()          # local machine name/address
port = 12321                 # port number for the server
s.bind((host, port))         # bind to the port
s.listen(5)                  # waiting for the client to connect
while True:
    c, addr = s.accept()             # accept the connection request from the client
    print 'Connected to', addr
    c.send('Connection Successful')
c.close()                            #close the socket
```

Source: PYTHON NETWORK PROGRAMMING, TutorialsPoint

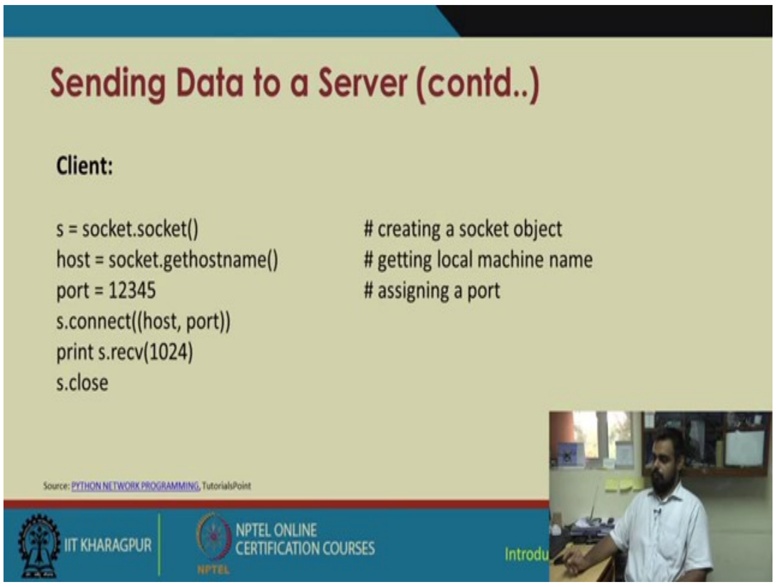
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 13

Now while sending a data from a client to a server we need 2 scripts one for the server and one for the client and 2 things will be required first is the IP address of the server and the port of the server to which your client will bind.

So, your client can have a variety of IP addresses, but it is important that your server retain a singular IP address see for this infrastructure for this exact project we are demonstrating you only require one way communication that is from the client to the server. So, your clients do not need to have fixed IP addresses may be they can have a variety of IP addresses about which the server has no prior knowledge, but it is very important that each and every client should be aware of the servers IP address as well as the port to which the client is going to connect. So, we start off with calling the functions of socket.socket we put the host as socket.gethostname we assign a particular port and we bind to the port by calling the host name is the first argument and the port number as the second argument and we start our listening function which waits for a client to connect to it.

Now, while true it signifies while a client has accepted the connection and connected to your server on that particular port it will generate one address and this address can be stored and later on used for later on used for time stamping your data to which on your server in various data logs. So, imagine your your server has been connected to 20 different IP addresses or clients. So, these 20 clients are sending variety of data to you, but you need to know the exact location of each of these clients. So, that later on you can segregate the data and connect to it.

(Refer Slide Time: 09:53)



Sending Data to a Server (contd..)

Client:

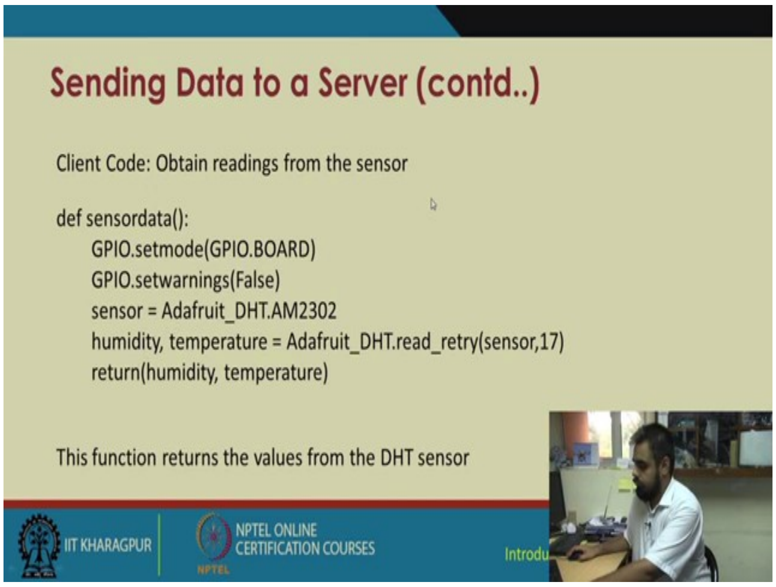
```
s = socket.socket()              # creating a socket object
host = socket.gethostname()      # getting local machine name
port = 12345                     # assigning a port
s.connect((host, port))
print s.recv(1024)
s.close
```

Source: PYTHON NETWORK PROGRAMMING, TutorialsPoint

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, once your connection has been accepted it sends the server sends connection successful and eventually when the socket is terminated by the user you call the functions c.close() for the client side again you call the functions socket.socket then you get the host name assign the port. Point to remember is this port number is similar to the servers port number because your client will be connecting to servers that particular port number and this host or the host name we are talking about this has to be the servers IP address.

(Refer Slide Time: 10:38)



Sending Data to a Server (contd..)

Client Code: Obtain readings from the sensor

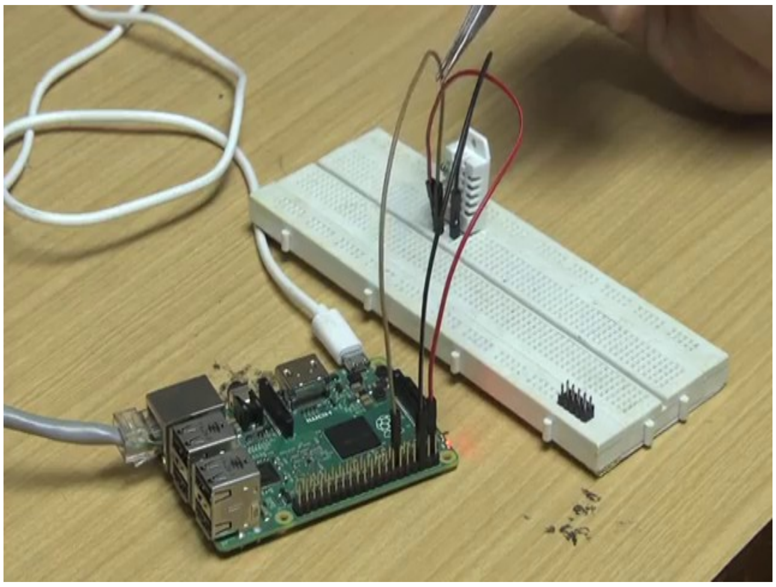
```
def sensordata():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setwarnings(False)
    sensor = Adafruit_DHT.AM2302
    humidity, temperature = Adafruit_DHT.read_retry(sensor,17)
    return(humidity, temperature)
```

This function returns the values from the DHT sensor

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, s.connect you connect to that particular IP address of that server on that particular port and you print that your data is being received s.receive and eventually you close the connection.

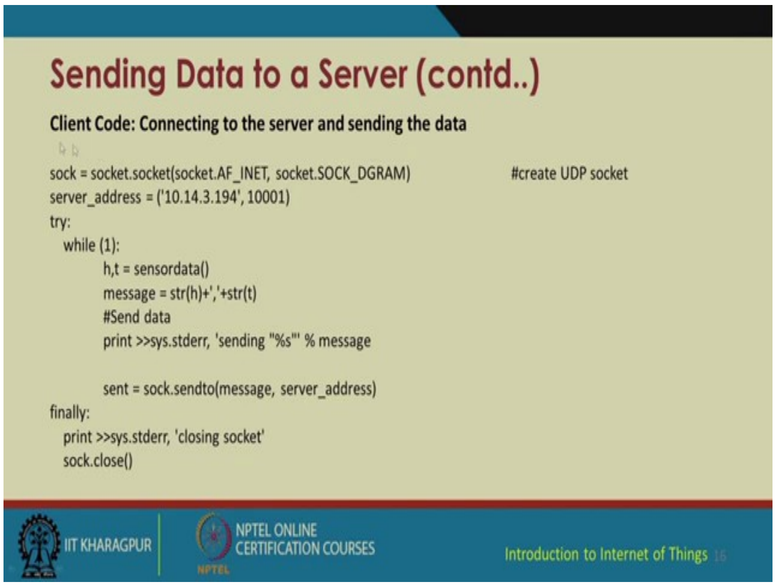
(Refer Slide Time: 10:57)



So, prior to this let to show the basic circuit connection, it is again back to the basics we have this DHT sensor the Vcc; the ground and the data pin are connected to their corresponding pins as defined in the previous slides this raspberry pi is connected to the network and we have already put in the client program on the raspberry pi we have already downloaded and install the Adafruit python library on this. So, we are ready to go now again back to the presentation. So, we are actually aiming to capture data from this sensor create a socket transmit a data to our remotely located server.

In this case my desktop will be the server and raspberry pi will be the client both of these are physically separate, but connected via a singular network to which various other devices are also connected, but since we are dealing with IP addresses and particular ports. So, we can expect zero interference from other devices. So, the client code for obtaining readings from the sensor can be given as we define a function called sensor data you can give it any other name. So, we set the board set the GPIO to board mode we set the warning to false sensor is Adafruit unless called DHT.AM2302 then we graph the humidity and temperature functions from the sensor and we return it to the calling function.

(Refer Slide Time: 12:39)



Sending Data to a Server (contd..)

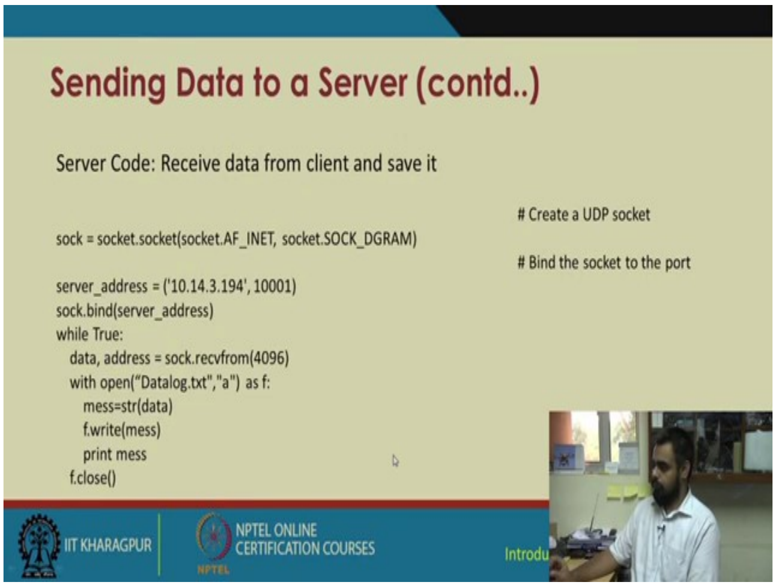
Client Code: Connecting to the server and sending the data

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)          #create UDP socket
server_address = ('10.14.3.194', 10001)
try:
  while (1):
        h,t = sensordata()
        message = str(h)+','+str(t)
        #Send data
        print >>sys.stderr, 'sending "%s"' % message

        sent = sock.sendto(message, server_address)
finally:
  print >>sys.stderr, 'closing socket'
  sock.close()
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 16

So, in this client code for creating the socket as you can see the AF_INET has been called and a socket datagram is being used for this particular program my pc has this following IP and I opened up port number 10001 if you can randomly choose ports to create a socket. So, I open port 10001 to accept connections from various clients. So, my system client will try on sending data over the network to the server. So, h, t; that means, these will be assigned the humidity and temperature data from the sensor data function which we defined in the previous slide and we formatted in the terms of a string and we transmitted over the socket to that particular server address and when this transmission is complete from input receipt from the user and the user terminates the code the sock the close function will execute.

(Refer Slide Time: 13:55)



Sending Data to a Server (contd..)

Server Code: Receive data from client and save it

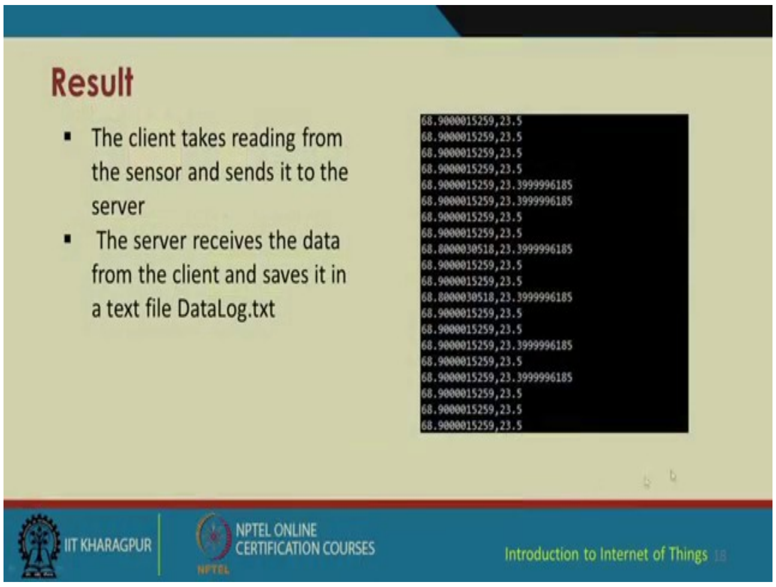
```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)        # Create a UDP socket

server_address = ('10.14.3.194', 10001)                        # Bind the socket to the port
sock.bind(server_address)
while True:
  data, address = sock.recvfrom(4096)
  with open("Datalog.txt","a") as f:
    mess=str(data)
    f.write(mess)
    print mess
  f.close()
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Introdu

Similarly, the modification for a server side is you again create an AF_INET and sock underscore data gram socket the address will be the servers address and the port will be the port which will be use for connecting various clients to it and sock.bind it starts the server port. Now, while it is true it will keep on infinitely looping the data and address will be received from the socket and chunks of 4096 bytes you open a any text file called data log text or any other name you want to chose it may even not be a text file it may be a CSV file also.

So, you use that file if you remember we covered file reading and writing in programming introduction to programming in python. So, using that similar concept you open a text file and whatever data you are receiving over the network you start writing to the text file as well as if you want you can keep one printing the data also.

(Refer Slide Time: 15:15)



Result

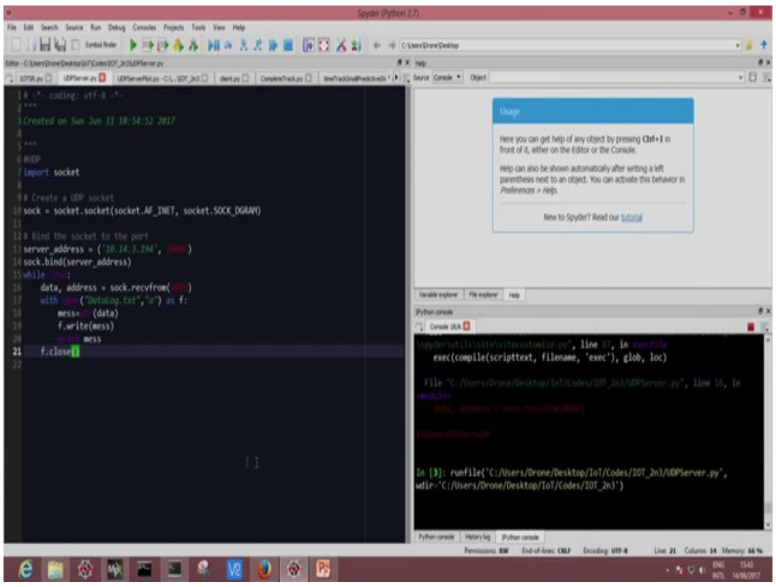
- The client takes reading from the sensor and sends it to the server
- The server receives the data from the client and saves it in a text file DataLog.txt

```
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.8000030518,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.8000030518,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Introduction to Internet of Things

So, if you see we will get an output something like this your temperature reading first is humidity and the next one is temperature. So, separated by a comma yours client will keep on streaming this data over and over again new values not the old redundant values every time every data pole it will generate a new value . So, let us look into the sever code first.

(Refer Slide Time: 15:44)

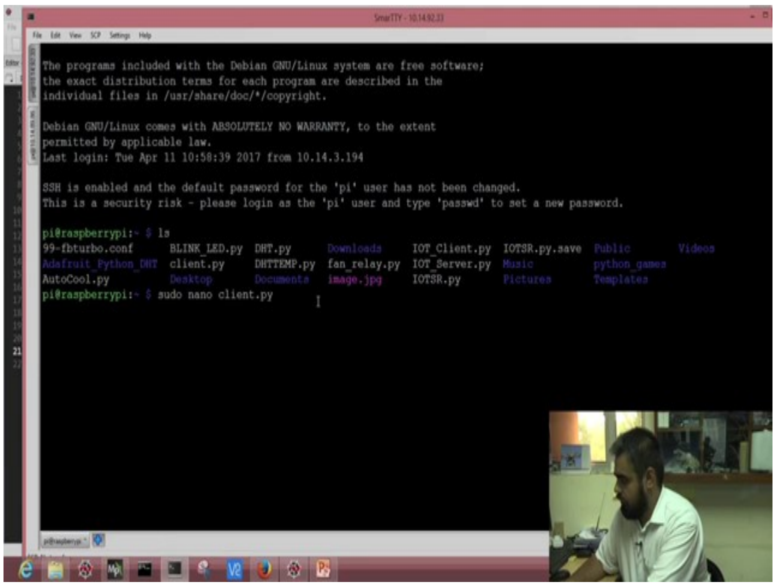


```python
# -*- coding: utf-8 -*-
"""
Created on Sun Jun 11 18:54:52 2017

"""
#UDP
import socket

# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the port
server_address = ('10.14.3.194', [illegible])
sock.bind(server_address)
while True:
    data, address = sock.recvfrom([illegible])
    with open("DataLog.txt","a") as f:
        mess=str(data)
        f.write(mess)
        print mess
    f.close()
```

```
\spyder\utils\site\sitecustomize.py", line 87, in execfile
    exec(compile(scripttext, filename, 'exec'), glob, loc)

  File "C:/Users/Drone/Desktop/IoT/Codes/IOT_2n3/UDPServer.py", line 16, in <module>
    data, address = sock.recvfrom(4096)

KeyboardInterrupt



In [3]: runfile('C:/Users/Drone/Desktop/IoT/Codes/IOT_2n3/UDPServer.py',
wdir='C:/Users/Drone/Desktop/IoT/Codes/IOT_2n3')
```

So, over here I have my UDP server. So, I have put in my IP and put in my port right data log txt for there was this file data log dot txt. I will delete it right this I have opened in append mode. So, that your data keeps on appending on to the same file again and again without overwriting the previous data. So, I start my server now as you can see the server has started it will wait for connections from the client and until then it will do nothing.

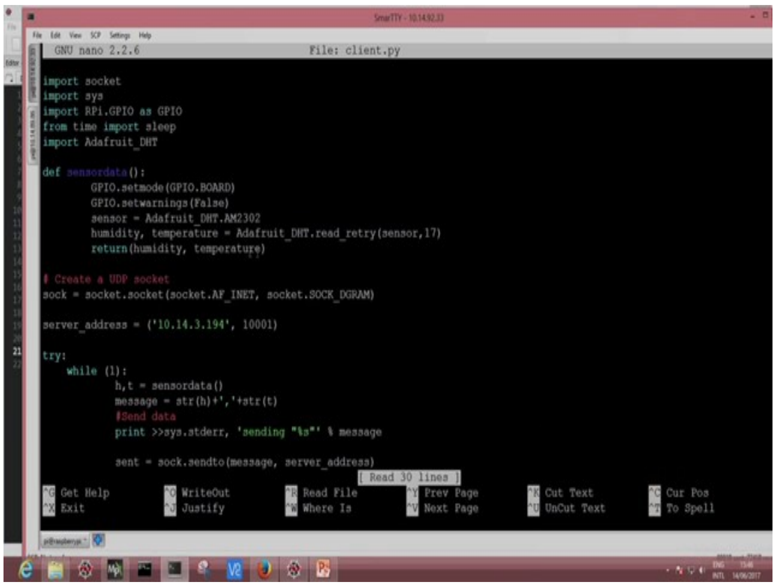
(Refer Slide Time: 16:33)



```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 11 10:58:39 2017 from 10.14.3.194

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $ ls
99-fbturbo.conf      BLINK_LED.py  DHT.py      Downloads     IOT_Client.py  IOTSR.py.save  Public        Videos
Adafruit_Python_DHT  client.py     DHTTEMP.py  fan_relay.py  IOT_Server.py  Music          python_games
AutoCool.py          Desktop       Documents   image.jpg     IOTSR.py       Pictures       Templates
pi@raspberrypi:~ $ sudo nano client.py
```

Now to access the client or the raspberry pi we remotely to this raspberry pi way system. So, we run this particular program client.py try to doing this you may just check whether our sensor is working fine or not let us try to read some temperature just like the previous lecture I will try running this DHT temp file, ok I need to change the port this system is not responding ok I will login to this raspberry pi again yes. So, this is my correct raspberry pi actually we have lots of raspberry pi is connected to over the network. So, we check; what exactly is the file content.

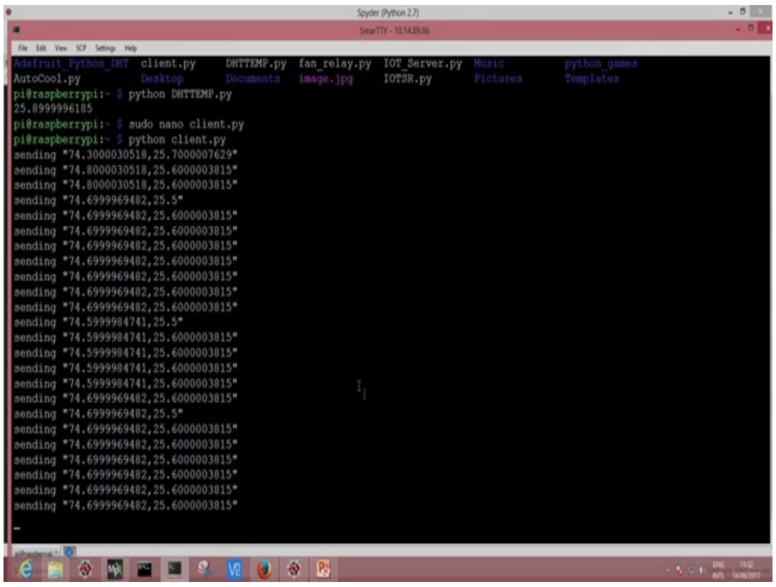
(Refer Slide Time: 18:45)



```
GNU nano 2.2.6                    File: client.py

import socket
import sys
import RPi.GPIO as GPIO
from time import sleep
import Adafruit_DHT

def sensordata():
        GPIO.setmode(GPIO.BOARD)
        GPIO.setwarnings(False)
        sensor = Adafruit_DHT.AM2302
        humidity, temperature = Adafruit_DHT.read_retry(sensor,17)
        return(humidity, temperature)

# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_address = ('10.14.3.194', 10001)

try:
    while (1):
            h,t = sensordata()
            message = str(h)+','+str(t)
            #Send data
            print >>sys.stderr, 'sending "%s"' % message

            sent = sock.sendto(message, server_address)
                                  [ Read 30 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
```

So, this was covered in the presentation we opened our script file in a editor. So, you are first going to check whether our temperature sensor working fine or not.

(Refer Slide Time: 19:04)



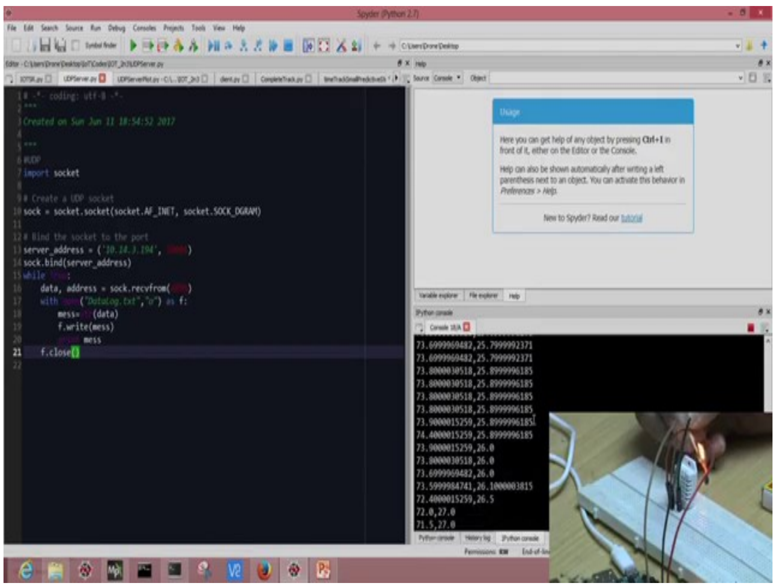
```
Adafruit_Python_DHT  client.py   DHTTEMP.py  fan_relay.py  IOT_Server.py  Music     python_games
AutoCool.py          Desktop     Documents   image.jpg     IOTSR.py       Pictures  Templates
pi@raspberrypi:~ $ python DHTTEMP.py
25.8999996185
pi@raspberrypi:~ $ sudo nano client.py
pi@raspberrypi:~ $ python client.py
sending "74.3000030518,25.7000007629"
sending "74.8000030518,25.6000003815"
sending "74.8000030518,25.6000003815"
sending "74.6999969482,25.5"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.5999984741,25.5"
sending "74.5999984741,25.6000003815"
sending "74.5999984741,25.6000003815"
sending "74.5999984741,25.6000003815"
sending "74.5999984741,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.5"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
sending "74.6999969482,25.6000003815"
```

We will try running this code DHTTEMP.py I have shown in the previous lecture. So, once this is executed if the sensor has been correctly connected if the pins are correct it is suppose to return temperature value on the terminal. So, yes it appears that sensor is properly connected returns a value of 25.89 degree Celsius now we can proceed with the more complicated part we will have a look at the client program.

This is a same program which was covered in the presentation we define a function called sensor data we open up a socket aimed at this IP address which is my server or desktop for this particular port 10001 you can change the port according to your needs and you iteratively keep on pushing data to this server . So, before beginning we will check yes the server is still on it is still waiting for incoming messages from any client which will connect to it now a last made client to send data to this server. So, you see it is sending 2 values this 74.3 and 25.7. So, first one is for the humidity value and the second one is from the temperature readings.

So, this will iteratively keep on going again and again until the user terminates this program over at the server side you see the same thing is being received that humidity readings and the corresponding temperature readings. So, for a demonstration I light up a match in front of the DHT sensor and you can check the corresponding readings change over at the server.

(Refer Slide Time: 22:02)



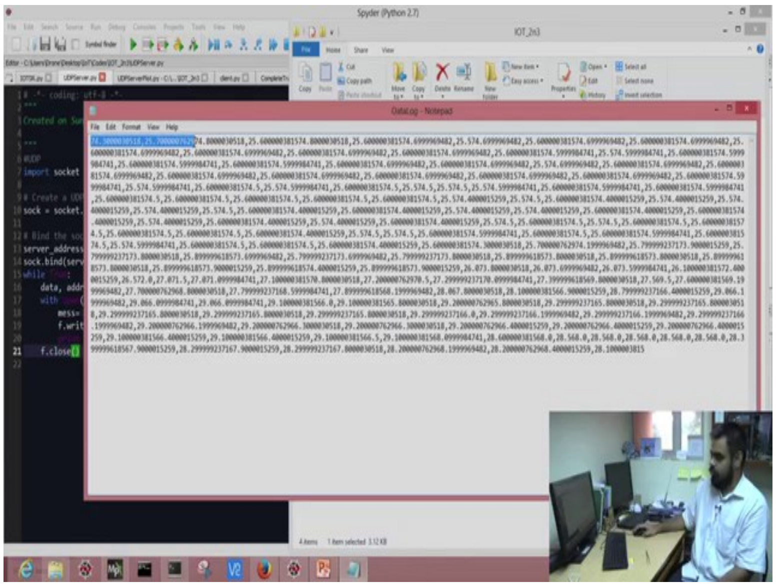
```
# -*- coding: utf-8 -*-
"""
Created on Sun Jun 11 18:54:52 2017

"""
#UDP
import socket

# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the port
server_address = ('10.14.3.194', [illegible])
sock.bind(server_address)
while True:
    data, address = sock.recvfrom([illegible])
    with open("DataLog.txt","a") as f:
        mess= str(data)
        f.write(mess)
        print mess
    f.close()
```

```
73.6999969482,25.7999992371
73.6999969482,25.7999992371
73.8000030518,25.8999996185
73.8000030518,25.8999996185
73.8000030518,25.8999996185
73.8000030518,25.8999996185
73.9000015259,25.8999996185
74.4000015259,25.8999996185
73.9000015259,26.0
73.8000030518,26.0
73.6999969482,26.0
73.5999984741,26.1000003815
72.4000015259,26.5
72.0,27.0
71.5,27.0
```

So, see the temperature is slowly increasing. So, it has reach all most 27 you can see the same variations over at the client side. Now I will terminate my client code. So, once this socket is close once I terminate this will send a message closing socket and closes down the socket you see the server code has stopped coming in right. So, whenever I start this code again my server will again start receiving this code. So, this we demonstrated using one single raspberry pi as a client and one single sensor you can; obviously, include multiple raspberry pis and multiple sensors all sending data to this particular server or a single IP at the port and another thing we will check when this data is incoming into the server initially I deleted this DataLog.txt file.

Now, you see it has automatically created another DataLog.txt and this function once I open it you see all the data has been stored.

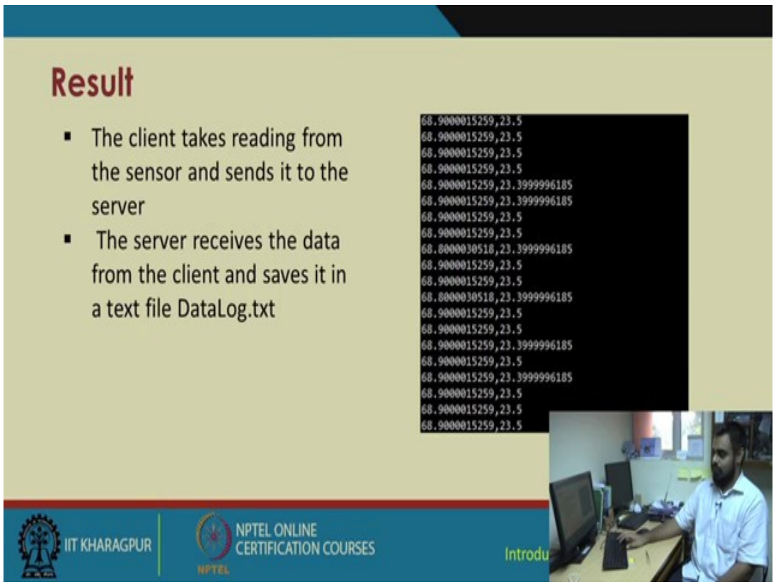
(Refer Slide Time: 23:52)



```
74.5000030518,25.700000762974.8000030518,25.600000381574.8000030518,25.600000381574.6999969482,25.574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.
600000381574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.600000381574.5999984741,25.574.5999984741,25.600000381574.5999
984741,25.600000381574.5999984741,25.600000381574.5999984741,25.600000381574.6999969482,25.600000381574.6999969482,25.574.6999969482,25.600000381574.6999969482,25.600003
81574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.600000381574.6999969482,25.600000381574.59
99984741,25.574.5999984741,25.600000381574.5,25.574.5999984741,25.600000381574.5,25.574.5,25.574.5,25.574.5999984741,25.600000381574.5999984741,25.600000381574.5999984741
,25.600000381574.5,25.600000381574.5,25.600000381574.5,25.600000381574.5,25.600000381574.5,25.574.4000015259,25.574.5,25.600000381574.4000015259,25.574.4000015259,25.574.
4000015259,25.574.4000015259,25.574.5,25.600000381574.4000015259,25.600000381574.4000015259,25.574.4000015259,25.574.4000015259,25.600000381574.4000015259,25.600000381574
.4000015259,25.574.4000015259,25.600000381574.4000015259,25.574.4000015259,25.600000381574.4000015259,25.574.5,25.600000381574.5,25.574.5,25.600000381574.5,25.6000003815
74.5,25.600000381574.5,25.600000381574.5,25.600000381574.4000015259,25.574.5,25.574.5,25.600000381574.5999984741,25.600000381574.5,25.600000381574.5999984741,25.60000038157
74.5,25.574.5999984741,25.600000381574.5,25.600000381574.5,25.600000381574.4000015259,25.600000381574.3000030518,25.700000762974.1999969482,25.799999237173.9000015259,25.
799999237173.8000030518,25.899999618573.6999969482,25.799999237173.6999969482,25.799999237173.8000030518,25.899999618573.8000030518,25.899999618573.8000030518,25.8999961
8573.8000030518,25.899999618573.9000015259,25.899999618574.4000015259,25.899999618573.9000015259,26.073.8000030518,26.073.6999969482,26.073.5999984741,26.100000381572.400
0015259,26.572.0,27.071.5,27.071.0999984741,27.100000381570.8000030518,27.200000762970.5,27.299999237170.0999984741,27.399999618569.8000030518,27.569.5,27.600000381569.19
99969482,27.700000762968.8000030518,27.799999237168.5999984741,27.899999618568.1999969482,28.067.8000030518,28.100000381566.9000015259,28.799999237166.4000015259,29.066.1
999969482,29.066.0999984741,29.066.0999984741,29.100000381566.0,29.100000381565.8000030518,29.200000762965.8000030518,29.299999237165.8000030518,29.299999237165.80000305
18,29.299999237165.8000030518,29.299999237165.8000030518,29.299999237165.8000030518,29.299999237166.0,29.299999237166.1999969482,29.299999237166.1999969482,29.299999237166
.1999969482,29.200000762966.1999969482,29.200000762966.3000030518,29.200000762966.3000030518,29.200000762966.4000015259,29.200000762966.4000015259,29.200000762966.4000015
259,29.100000381566.4000015259,29.100000381566.4000015259,29.100000381566.5,29.100000381568.0999984741,28.600000381568.0,28.568.0,28.568.0,28.568.0,28.568.0,28.568.0,28.3
99999618567.9000015259,28.299999237167.9000015259,28.299999237167.8000030518,28.200000762968.1999969482,28.200000762968.4000015259,28.100003815
```

So, this can be used as the server log or it can be extorted and use with various algorithms for higher processing.

(Refer Slide Time: 24:08)



Result

- The client takes reading from the sensor and sends it to the server
- The server receives the data from the client and saves it in a text file DataLog.txt

```
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.8000030518,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.8000030518,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.3999996185
68.9000015259,23.5
68.9000015259,23.5
68.9000015259,23.5
```

So, I hope you gained some experience of integrating various sensors over the network and creating a client server connection and I hope this will give you a little bit touch of this IoT.

The course for which you have been attending these lectures and in the next lecture we will be focusing more on what to do with the data and what exactly can be integrated can be done you can may be plot the data you can store the data you can refine the data you are storing and other such tasks.

Thank you.