Introduction to Internet of Things
Prof. Sudip Misra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
Implementation of IoT with Raspberry Pi- I

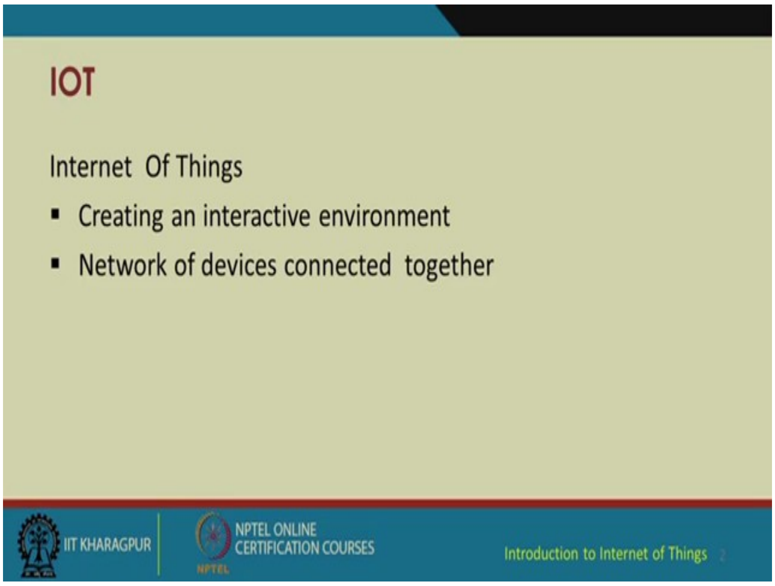
In the previous lectures on introduction to Raspberry pi, we have learned about how to configure Raspberry pi and how to use it along with 2 bear basic sensors on is killer sensor and the other one a camera sensor. So, 2 sensors were used and the basic interfacing was introduced to you in the previous 2 lectures. In this lecture you are going to learn about how to integrate Raspberry pi for for enabling IoT development.

So, here you are going to learn about you know these things in more detail. So, not only this lecture but the next 3 lecture I mean this one and the next 2 lectures you are going to learn about how to integrate the sensors, how to integrate the sensors different types of sensors not just one or two, but different types of sensors at the same time then after these sensors have collected this data then how to send the data how to disseminate the data through some kind of creation of socket to a remote server for that processing using in the UDP protocol which is a transport layer protocols. So, using UDP how to do particular this particular thing you are going to learn and there after you are also going to learn in this and the next 2 lectures about you know about how to visualize the data at the server.

So, the data is received then we have to visualize the data is received at the server at the server how to visualize the data this is what we are going to learn. So, I and Mr. Anandroop Mukharjee; your TA is going to taking through these few steps for achieving these things; that means, the data acquisition through these different types of sensors then sending through the network and there after you know visualizing the data at the server.

Hello, in this lecture this lecture will be covering 3 different parts of implementation of IoT with Raspberry pi. So, in part one will be discussing about using Raspberry pi to capture data from sensors and making a basic decision on the basis of capture data to actuate some device.

(Refer Slide Time: 02:55)



IOT

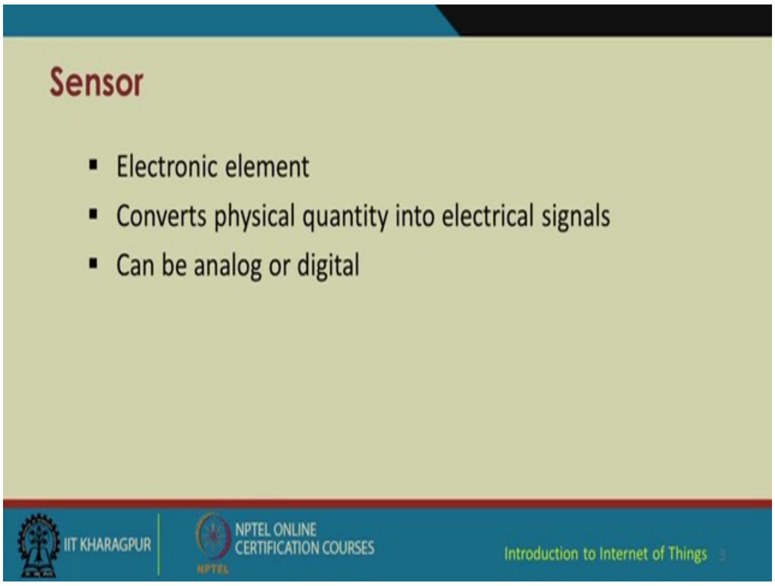
Internet Of Things

- Creating an interactive environment
- Network of devices connected together

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 2

So, first of all 2 recapitulate internet of things basically needs to create an interactive environment and additionally it has a network of devices which are connected together. So, bringing these 2 together for this particular topic will get into the hands on.

(Refer Slide Time: 03:16)



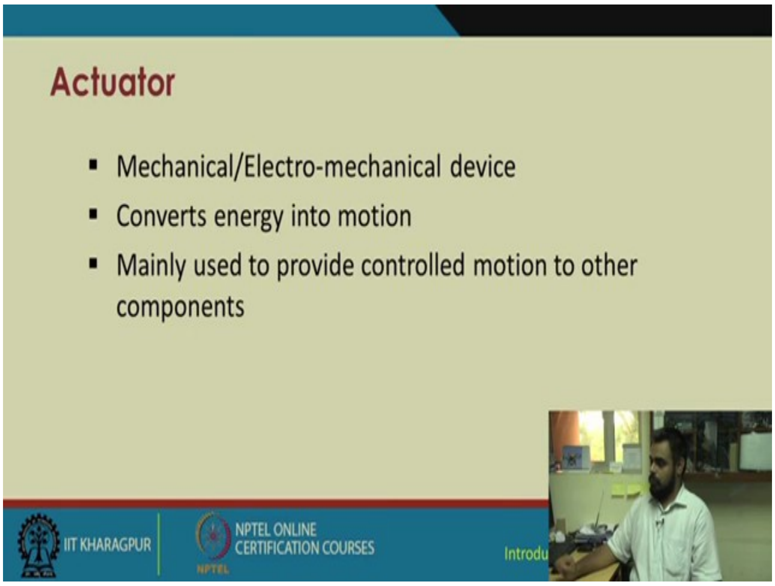
Sensor

- Electronic element
- Converts physical quantity into electrical signals
- Can be analog or digital

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 3

So, another thing regarding sensors as of already learnt by now and it has been discussed many times, sensors are electronic elements which convert physical quantity into electrical signals or any measurable quantity into electrical signals and sensors can be primarily analog or digital.

(Refer Slide Time: 03:37)

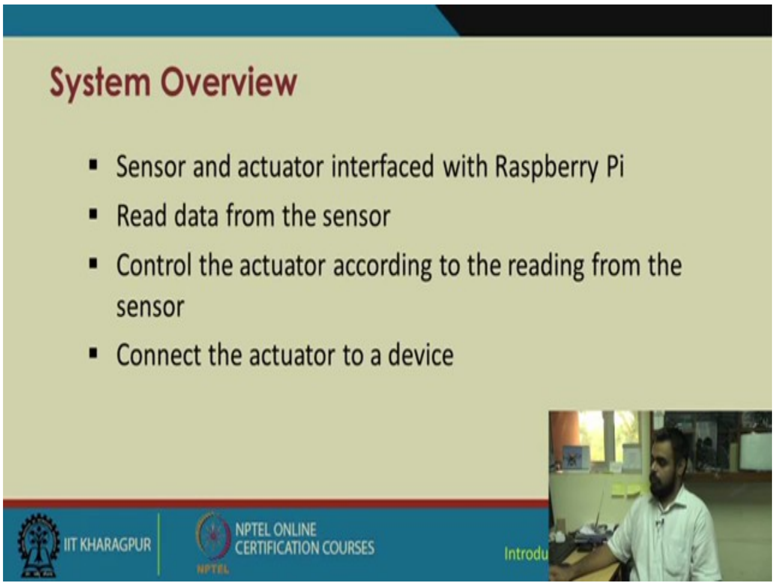


Actuator

- Mechanical/Electro-mechanical device
- Converts energy into motion
- Mainly used to provide controlled motion to other components

Similarly Actuators; Actuators are electro mechanical devices or they can be standalone mechanical devices also and generally they actuate or convert energy into motion. So, mainly they are used for providing control motion to other components in a big system.

(Refer Slide Time: 03:59)

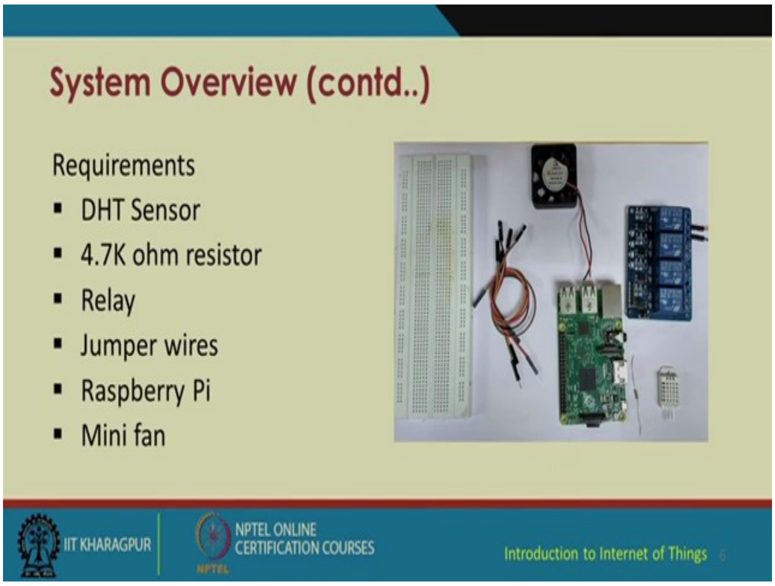


System Overview

- Sensor and actuator interfaced with Raspberry Pi
- Read data from the sensor
- Control the actuator according to the reading from the sensor
- Connect the actuator to a device

So, in this a system overview is is as follows sensor and actuator are interfaced with Raspberry pi, data is read from the sensor the actuator is controlled according to the readings from the sensor and the actuator basically which is connected to the sensor is being controlled from the readings and this control mechanism will be showing a brief decision making loop which can be replaced by additional much higher inversion like machine learning statistical learning or even deep learning based methods.

(Refer Slide Time: 04:42)



System Overview (contd..)

Requirements

- DHT Sensor
- 4.7K ohm resistor
- Relay
- Jumper wires
- Raspberry Pi
- Mini fan

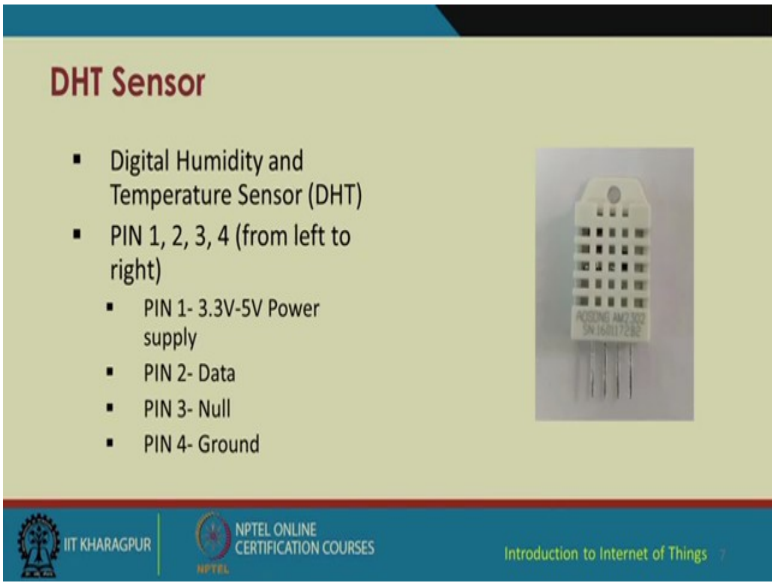
IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Introduction to Internet of Things 6

So, for this, the following are the system requirements we are again using a DHT sensor which as a digital humidity and temperature sensor we are using a 4.7 kilo ohm resistor relay some jumper wires Raspberry pi and mini fan which we have going to connect to the relay to show the effectiveness of our decision making approach.

(Refer Slide Time: 05:05)



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

AOSONG AM2302
SN 16011728

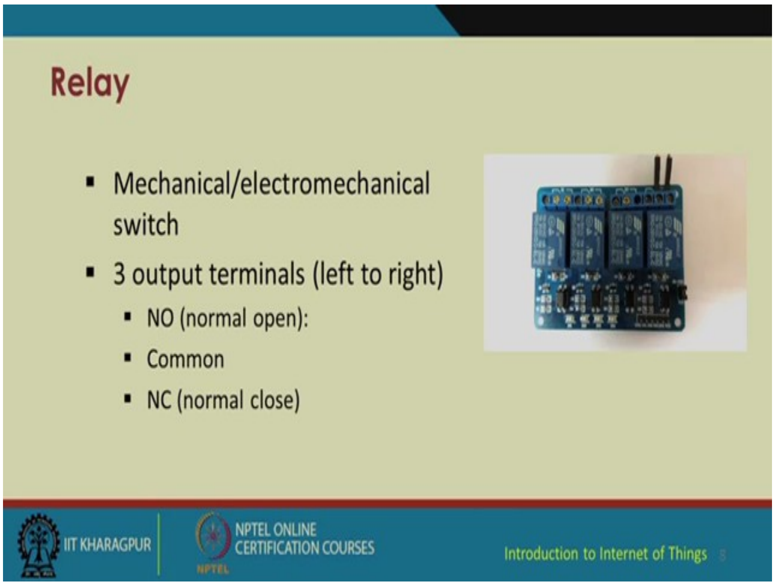
IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Introduction to Internet of Things 7

So, as you already know this digital humidity and temperature sensor it has got 4 pins. So, generally when you keep it like it is shown in the slide from left to right you number the pins 1 up to 4 and pin one is generally use for power supply rangers from 3.3 to 5 volt pin 2 supplies the data to the process a board to which this sensor is connected, pin 3 is generally kept open and pin 4 is connected to the ground.

(Refer Slide Time: 05:39)



Relay

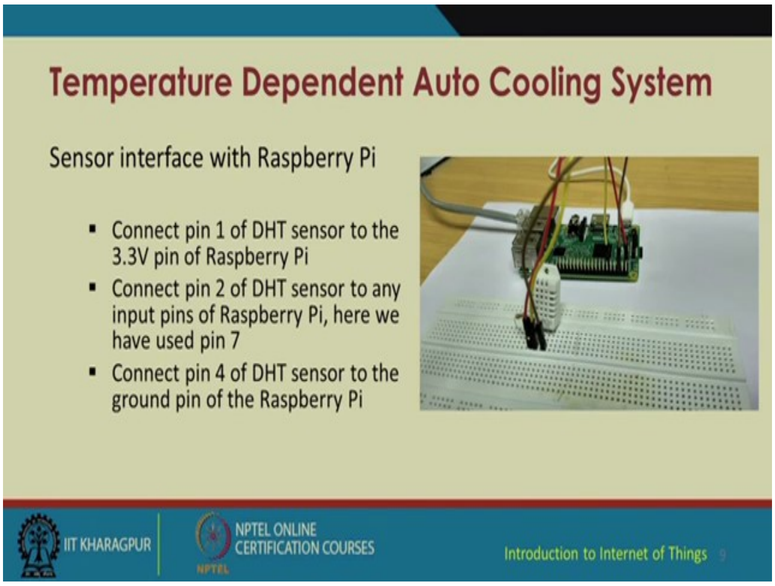
- Mechanical/electromechanical switch
- 3 output terminals (left to right)
  - NO (normal open):
  - Common
  - NC (normal close)

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

And this one is called the relay board it is an electromechanical switch. So, it has got 3 output terminals starting from left to right. So, in this particular board you can see there are actually four relays this is a composite board consisting of four different relays. So, you can connect four different devices to it this single entity is known as one relay and in electronic we call this a sugar cube relay generally you can find sugar cube relays ranging from operating voltage as of 6 volts up to 12 volts.

So, as you can see there are 3 terminals corresponding to each relay. So, from left to right the first terminal is designation NO or normally open the middle one is common and the third terminal is known as NC are normally close.

(Refer Slide Time: 06:37)



Temperature Dependent Auto Cooling System

Sensor interface with Raspberry Pi

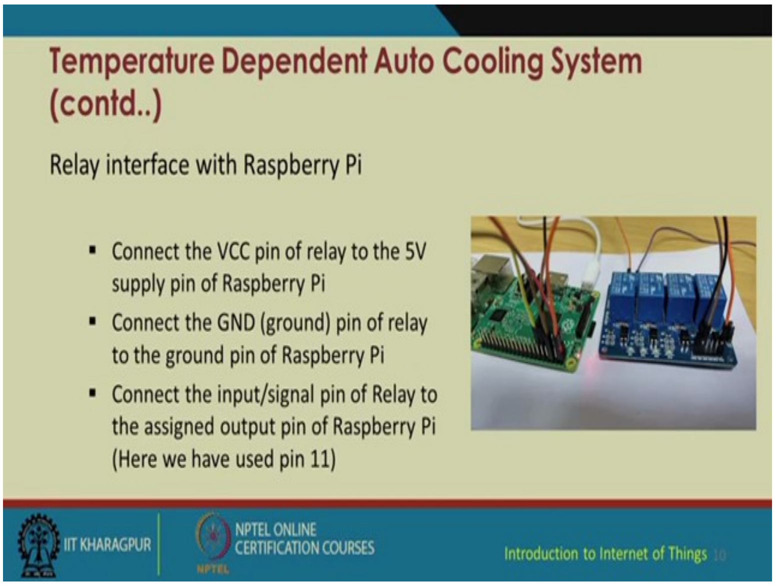
- Connect pin 1 of DHT sensor to the 3.3V pin of Raspberry Pi
- Connect pin 2 of DHT sensor to any input pins of Raspberry Pi, here we have used pin 7
- Connect pin 4 of DHT sensor to the ground pin of the Raspberry Pi

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 9

Now, what we are about to attempt is a temperature dependent auto cooling system. So, for this the primary sensor interface with the Raspberry pi is going to be using DHT. So, we have already discussed and showed you the demos of how a DHT is connected to Arduino board as well as the Raspberry pi. So, again just to recapitulate we connect pin one of the DHT sensor to the 3.3 volt power supply of Raspberry pi we connect pin 2 of this DHT sensor which supplies data to Raspberry pi to pin seven and finally, we connect pin four of the DHT sensors to the ground pin on the Raspberry pi.

However it is to be noted for this DHT sensor this follows the BCM configuration of Raspberry pi. So, if you recall Raspberry pi has 2 configurations we generally discussed with the board configuration these 2 configurations are board and BCM. So, board configuration we had discussed in the previous lecture. Now in this one will be mainly working using this DHT sensor in the BCM mode as specified by its manufacturer.

(Refer Slide Time: 07:57)



Temperature Dependent Auto Cooling System (contd..)

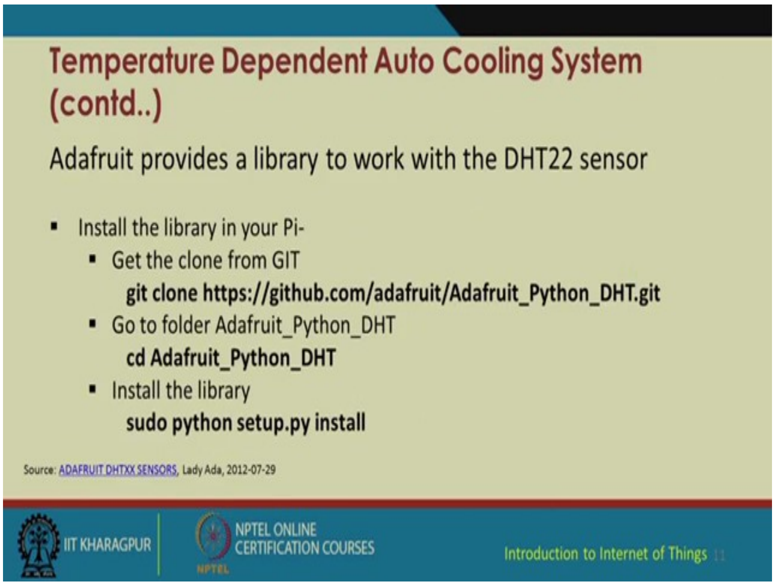
Relay interface with Raspberry Pi

- Connect the VCC pin of relay to the 5V supply pin of Raspberry Pi
- Connect the GND (ground) pin of relay to the ground pin of Raspberry Pi
- Connect the input/signal pin of Relay to the assigned output pin of Raspberry Pi (Here we have used pin 11)

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

Now additionally the relay interface with the Raspberry pi is as follows we connect the VCC pin of the relay to 5 volts supply of Raspberry pi you connect the ground to the ground of Raspberry pi and the input signal to the relay is assign to pin 11 and another point to remember is when we are using relay we are not using the BCM mode we are using the board mode. So, this pin number 11 is going to be according to board mode whereas, in the previous slide this pin 7 is according to BCM mode. So, these both these modes will be used simultaneously.

(Refer Slide Time: 08:42)



Temperature Dependent Auto Cooling System (contd..)

Adafruit provides a library to work with the DHT22 sensor

- Install the library in your Pi-
  - Get the clone from GIT

    git clone https://github.com/adafruit/Adafruit_Python_DHT.git
  - Go to folder Adafruit_Python_DHT

    cd Adafruit_Python_DHT
  - Install the library

    sudo python setup.py install

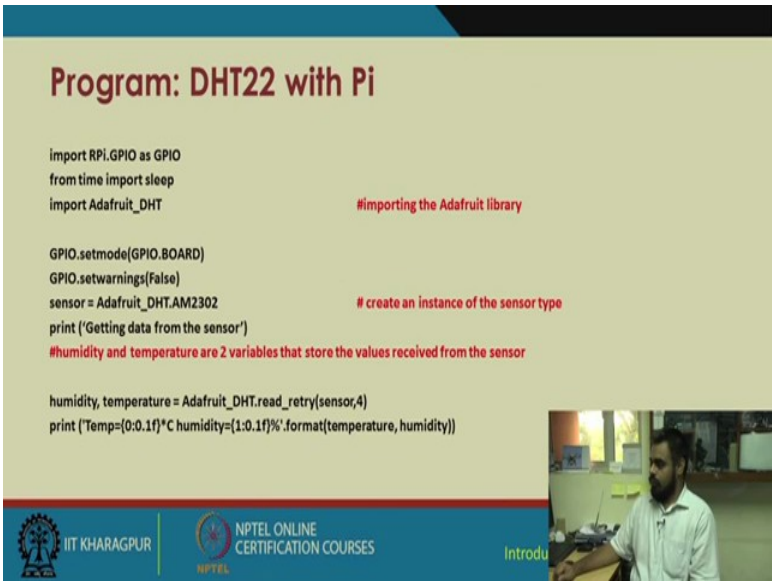
Source: ADAFRUIT DHTXX SENSORS, Lady Ada, 2012-07-29

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 11

Now, try to beginning the programming part like in Arduino we install and additional library for this DHT sensor. Similarly for Raspberry pi we need to install in additional library from Adafruit which basically supplied this sensor. So, we are using this DHT 22 sensor first of all in your Raspberry pi you need to install your Adafruit DHT sensor library using the following command and a point to remember is since you your using python base scripting we are going to implement the DHT python library you may be able to find come or cross DHT C library, DHT C++ libraries but we are more interested on in the python library.

So, the first line is this git clone https you follow this link and you press enter you will see here Raspberry pi it terminal its starts if it is connected to the internet it will start downloading this folder if go to the once the download is finished we go to the downloaded folder by putting in this command cd Adafruit_python_DHT because this will be the name of the folder which has been downloaded on your Raspberry pi system and remember this is not the installation the installation is yet to come. So, after you go into the directory you install the library by running this command sudo python setup.py install. So, once your installation is successful you can easily start creating codes for DHT using python and Raspberry pi.

(Refer Slide Time: 10:32)



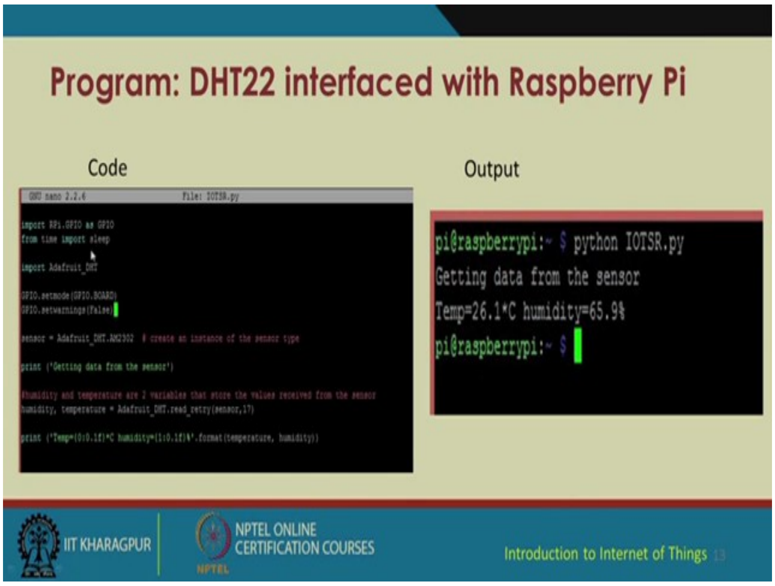
Program: DHT22 with Pi

```
import RPi.GPIO as GPIO
from time import sleep
import Adafruit_DHT                                  #importing the Adafruit library

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,4)
print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
```

So, we have a sample DHT interfacing program with Raspberry pi first line starts off with importing the GPIO pins. So, import RPi.GPIO as GPIO then we import the time library for calling in the sleep function which provides delays 2 hour program and then we import the Adafruit_DHT library which we previously installed now the GPIO mode is initially set as board and warning have been set to false for the Adafruit it automatically takes the board mode as BCM. So, we are not explicitly identifying anything. So, sensor equal to Adafruit _DHT.AM232; 2302. So, this line has to be explicitly mentioned according to the documentation provided by the manufacturer and then we just print a line to signify whether this sensor have been successfully initialized or not and we assigned humidity and temperature values from the Adafruit library function by using read _retry and in name this sensor which we initially called and which is connected to BCM pin 4.

In the next line we print the temperature and humidity values iteratively not a iteratively we print this once, but we format it according to our needs.

(Refer Slide Time: 12:11)



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))
```

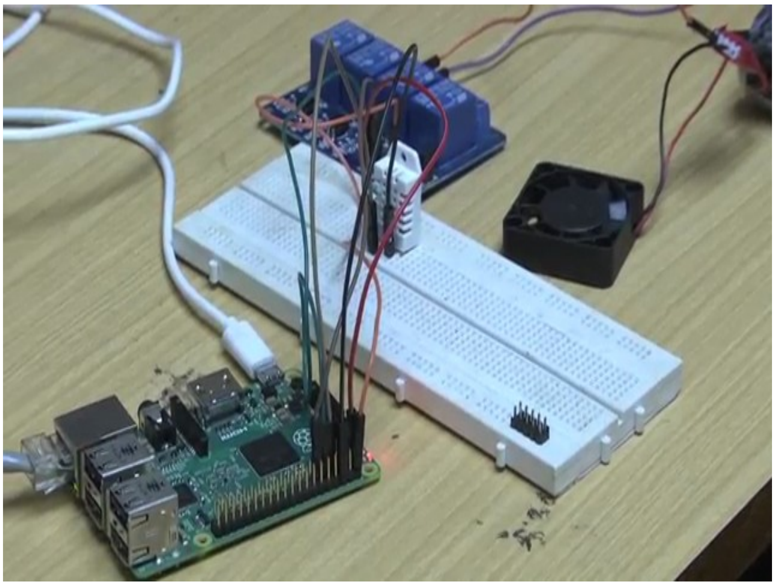
Output

```
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

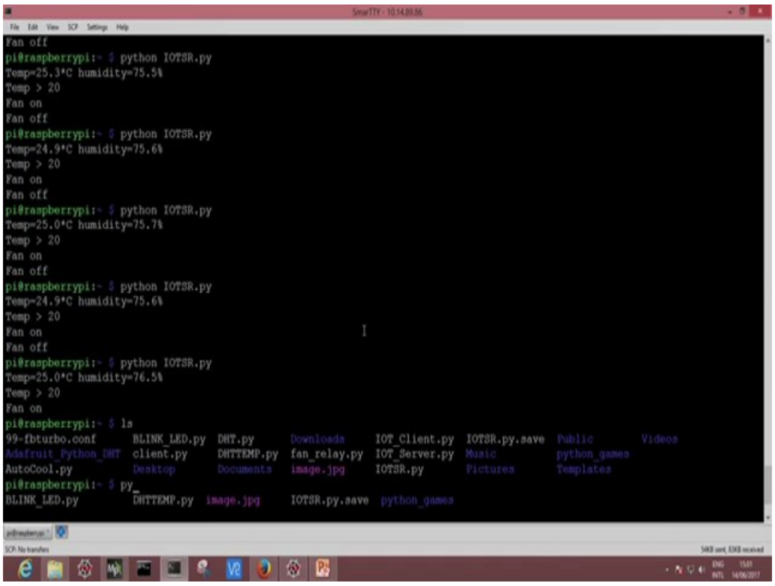
So, this is the code which we just showed on the right hand side you can see the output we created a python file on the Raspberry pi named IOTSR.py once it is executed it gathers data from this sensor and you can see your formatted print line is temp equal to 26.1 Celsius and humidity is 65.9 percent.

(Refer Slide Time: 12:52)



So, we can zoom into the hardware circuitry now as you can see this is your DHT sensor it is placed on the breadboard and according to the configuration mentioned previously in this slide we have connected the VCC ground and the data pin to the appropriate pins on the Raspberry pi board there are additional component attached will come to those later. But for now will be focusing on only these 3 wires the brown wire the red wire and the black wire.

(Refer Slide Time: 13:35)

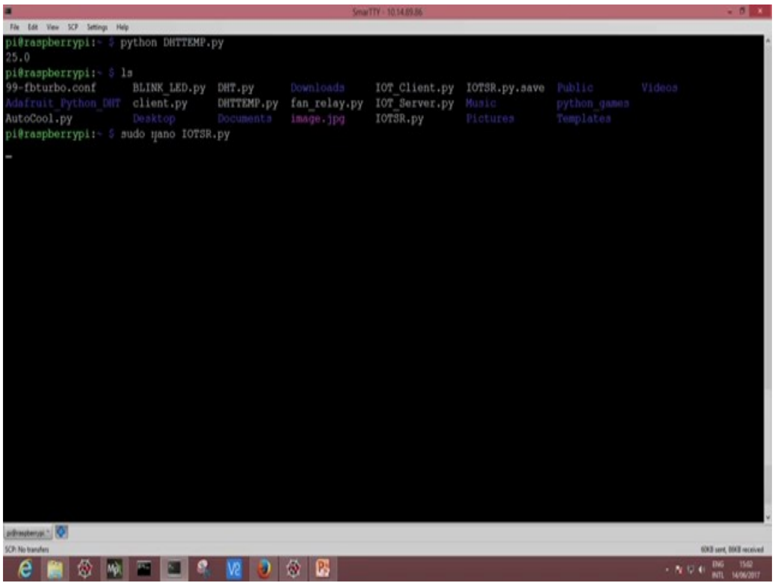


```
Fan off
pi@raspberrypi:~ $ python IOTSR.py
Temp=25.3*C humidity=75.5%
Temp > 20
Fan on
Fan off
pi@raspberrypi:~ $ python IOTSR.py
Temp=24.9*C humidity=75.6%
Temp > 20
Fan on
Fan off
pi@raspberrypi:~ $ python IOTSR.py
Temp=25.0*C humidity=75.7%
Temp > 20
Fan on
Fan off
pi@raspberrypi:~ $ python IOTSR.py
Temp=24.9*C humidity=75.6%
Temp > 20
Fan on
Fan off
pi@raspberrypi:~ $ python IOTSR.py
Temp=25.0*C humidity=76.5%
Temp > 20
Fan on
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:~ $ py_
BLINK_LED.py         DHTTEMP.py  image.jpg     IOTSR.py.save  python_games
```

So, coming back to the programming part I have remotely logged into the Raspberry pi system just increase the font a little bit. So, using this terminal I have actually logged into the

Raspberry pi. As you can see over here pi at the rate Raspberry pi I go into the directory I see that I have testing file named DHTTEMP.py lets run this first.

(Refer Slide Time: 14:41)

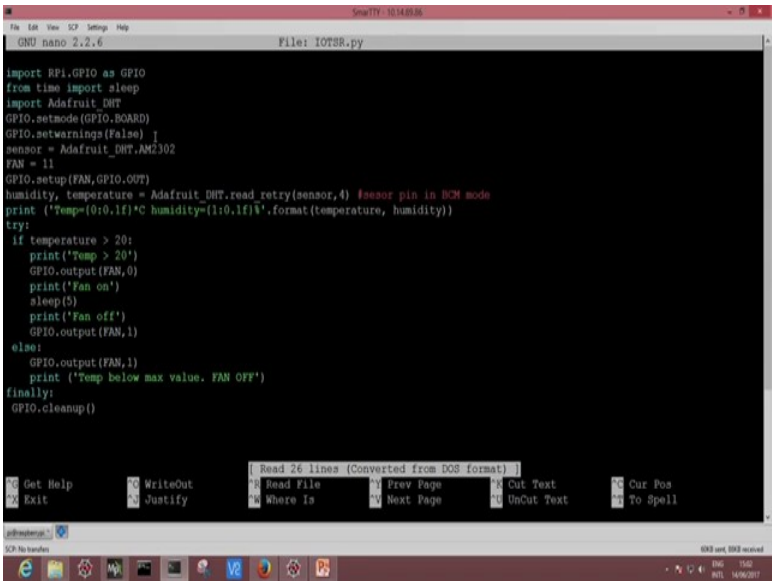


```
pi@raspberrypi:~ $ python DHTTEMP.py
25.0
pi@raspberrypi:~ $ ls
99-fbturbo.conf      BLINK_LED.py  DHT.py       Downloads     IOT_Client.py  IOTSR.py.save  Public        Videos
Adafruit_Python_DHT  client.py     DHTTEMP.py   fan_relay.py  IOT_Server.py  Music          python_games
AutoCool.py          Desktop       Documents    image.jpg     IOTSR.py       Pictures       Templates
pi@raspberrypi:~ $ sudo nano IOTSR.py
_
```

So, we give the simple command python DHTTEMP.py as soon as it is executed is a simple code to fetch only the temperature readings from the DHT sensor, so, the temperature reading as been read as 25.0 degree Celsius.

Now, will look at the IOTSR file, so, we have our IOTSR file over here lets open the editor and check whether the contents are the same.

(Refer Slide Time: 15:33)

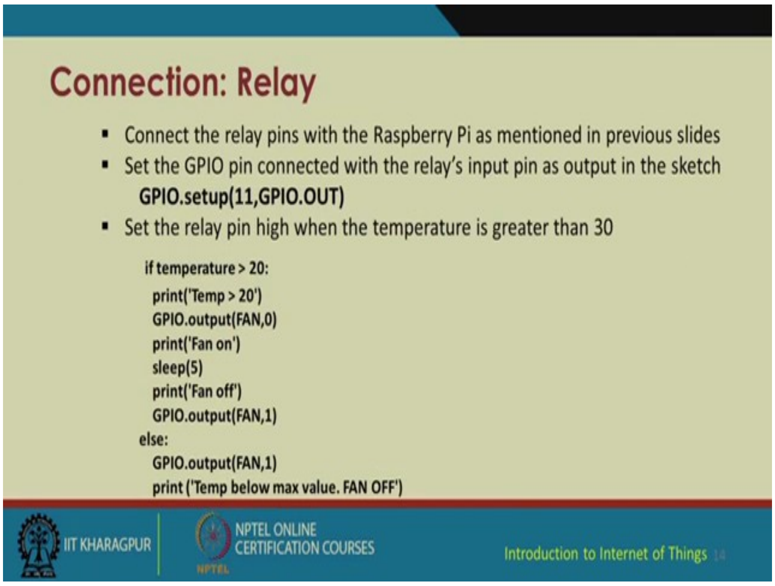


```
  GNU nano 2.2.6                        File: IOTSR.py

import RPi.GPIO as GPIO
from time import sleep
import Adafruit_DHT
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
sensor = Adafruit_DHT.AM2302
FAN = 11
GPIO.setup(FAN,GPIO.OUT)
humidity, temperature = Adafruit_DHT.read_retry(sensor,4) #sesor pin in BCM mode
print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
try:
 if temperature > 20:
    print('Temp > 20')
    GPIO.output(FAN,0)
    print('Fan on')
    sleep(5)
    print('Fan off')
    GPIO.output(FAN,1)
 else:
    GPIO.output(FAN,1)
    print ('Temp below max value. FAN OFF')
finally:
 GPIO.cleanup()

                         [ Read 26 lines (Converted from DOS format) ]
^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, we have imported the GPIO library we have imported the time module we have imported the Adafruit library we installed we have set the board mode. So, as I have told you before the Adafruit.read_retry this thing is pin is on by defaults setting BCM mode. So, this pin number four is according to BCM mode whereas, I have connected a relay I will come to that in the consecutive slides have connected a relay and it is connected to pin eleven according to the board mode.

Before going furthers, we have not checked our Raspberry pi is working fine the DHT sensor is working fine.

(Refer Slide Time: 16:41)



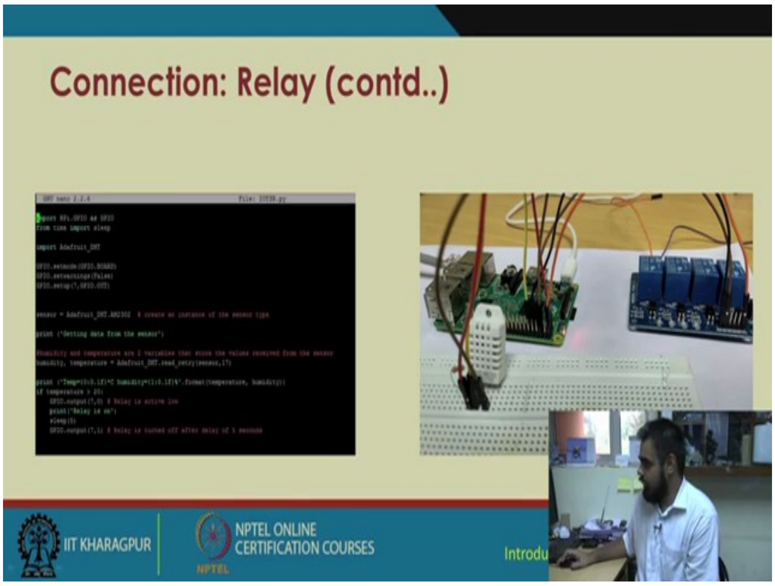
Connection: Relay

- Connect the relay pins with the Raspberry Pi as mentioned in previous slides
- Set the GPIO pin connected with the relay's input pin as output in the sketch
  GPIO.setup(11,GPIO.OUT)
- Set the relay pin high when the temperature is greater than 30

```
 if temperature > 20:
  print('Temp > 20')
  GPIO.output(FAN,0)
  print('Fan on')
  sleep(5)
  print('Fan off')
  GPIO.output(FAN,1)
else:
  GPIO.output(FAN,1)
  print ('Temp below max value. FAN OFF')
```

We are now connecting the relay part. So, we connect the relay input pin at pin eleven of the Raspberry pi using board mode and we put in the basic decision making loop that if temperature is greater than twenty you print temperature greater than 20 and it will turn on the relay. So, it will a fan is connected to the relay. So, I have just return fan comma 0 that is the fan will be turned on and then it will sleep for 5 minutes and then fan will be turned off and output pin will be again set to 1 otherwise if temperature is lesser then 20 the fan going to be switched on.

(Refer Slide Time: 17:32)

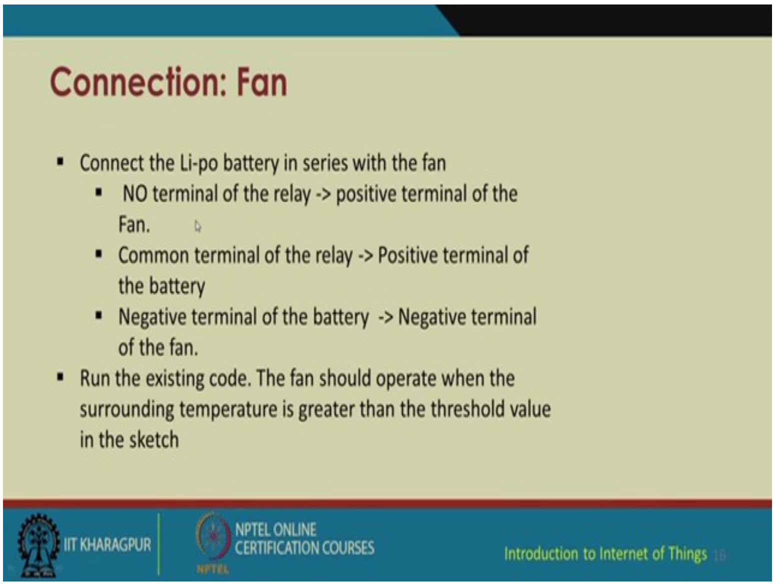


Connection: Relay (contd..)

```
GNU nano 2.2.6                         File: [illegible]

import RPi.GPIO as GPIO
from time import sleep

import Adafruit_DHT

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(7,GPIO.OUT)


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))
if temperature > 20:
    GPIO.output(7,0) # Relay is active low
    print('Relay is on')
    sleep(5)
    GPIO.output(7,1) # Relay is turned off after delay of 5 seconds
```

So, on the left hand side this is the code actually open this code few movements back on the Raspberry pi console and on the right hand side you have this relay board you have this DHT sensor connected to the Raspberry pi.

(Refer Slide Time: 17:52)



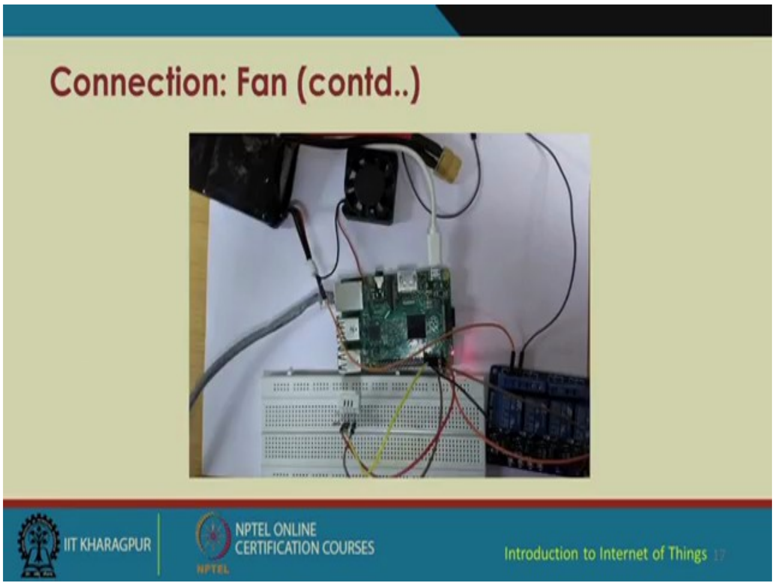
Connection: Fan

- Connect the Li-po battery in series with the fan
  - NO terminal of the relay -> positive terminal of the Fan.
  - Common terminal of the relay -> Positive terminal of the battery
  - Negative terminal of the battery -> Negative terminal of the fan.
- Run the existing code. The fan should operate when the surrounding temperature is greater than the threshold value in the sketch

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

Now this connection of the Li-po with the fan sorry, the connection of the relay with the fan is as follows we are connecting we are using lithium polymer battery we can use any other battery of sufficient rating since the fan we are using runs on 12 volt we are using a 12 volt Li-po battery to operate the fan the normally open terminal of the relay or the positive terminal is connected to the positive terminal of the fan in the common terminal of the relay is connected to positive terminal of battery negative terminal of battery is connected to negative terminal of the fan and when these connections have been made and the connection have been rechecked we run the IOTSR.py file. So, the connection will looks something like this.

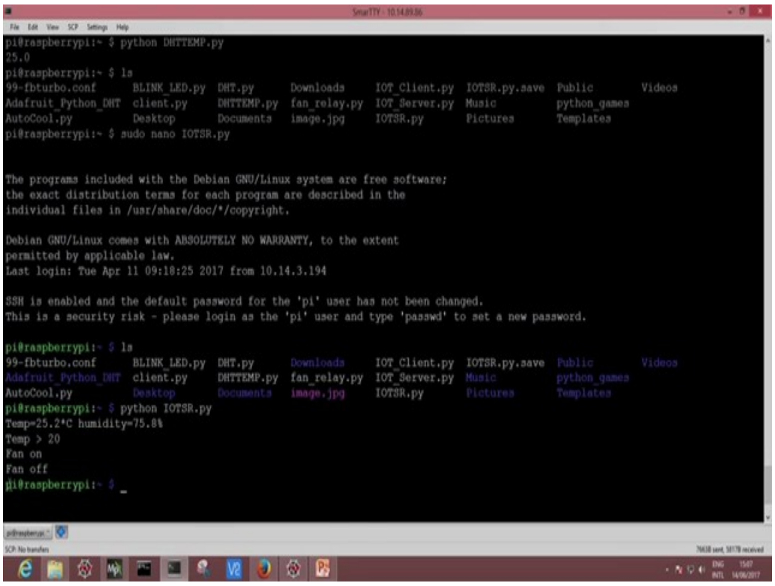
(Refer Slide Time: 18:46)



Connection: Fan (contd..)

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

(Refer Slide Time: 18:58)



```
pi@raspberrypi:~ $ python DHTTEMP.py
25.0
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 IOTSR.py


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 09:18:25 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:~ $ python IOTSR.py
Temp=25.2*C humidity=75.8%
Temp > 20
Fan on
Fan off
pi@raspberrypi:~ $ _
```

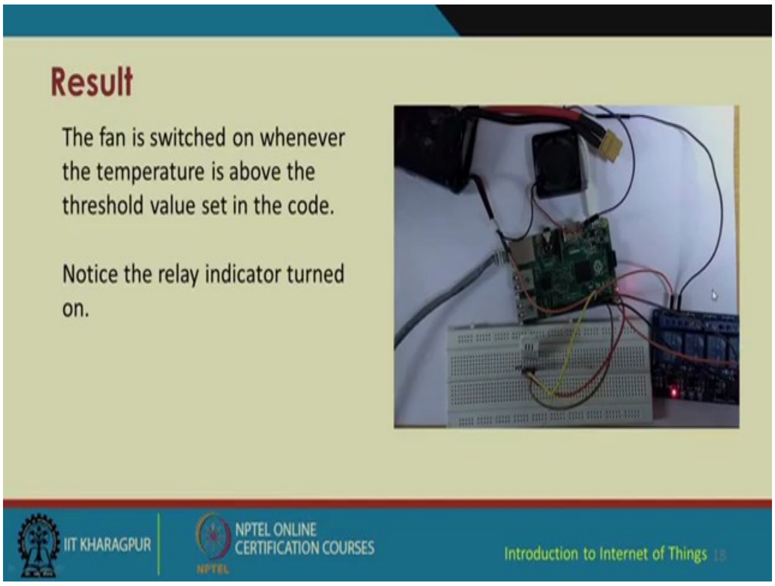
Now let us again log into Raspberry pi. So, once we execute this file you will see the output but prior to that I like to show you this is my Raspberry pi this was my DHT sensor giving me humidity and temperature readings this is my relay board this is the four channel relay board because you can connect four devices simultaneously. So, this one unit is known as a relay this single unit is called a sugar cube relay as you can see it has got 3 terminals normally open common and normally close and we have this small fan over here which is connected to a Li-po battery 3 cell Li-po. So, again back to the terminal will execute the code now.

So, as soon as the code is executed it gets the reading of temperature is 25.2 degree Celsius and humidity is almost 76 percent now the decision making loop has detected that the temperature is greater than twenty. So, it turns on the fan and after that after 5 seconds the fan is turned off. So, this can be modified the modifying the loop and instead of using a normal if else loop you can go for if else loop or a normal rule based decision making we can; obviously, go for various machine running best approaches only condition been that you need a lot of data historical data to predict the next action.

So, if you again focus on the circuit I like to show when I run this program again if you pay attention to the fan and it is connected to this relay board as soon as the program is executed and if the relay is turned on there will be a light blinking against the corresponding relay to which the fan is connected to this relay.

So, you see it has detected the temperature is greater than 25 the relay has been turned on, you see the fan is now also turned on after 5 second it will turn off. So, basic application is you can use this for automated cooling systems may be for your PC or may be for your various other systems or may be in industrial systems. Also suppose if the temperature of a Furness gets higher or if a temperature of a particular room or work place gets higher your fan automatically turns on and the network connected device remotely intimates you that your temperature is going high so, the fan has been turned on.

(Refer Slide Time: 22:22)



Result

The fan is switched on whenever the temperature is above the threshold value set in the code.

Notice the relay indicator turned on.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things 18

So, the outputs have been explode. I hope this gave you a bit of learning experience well cover other things and details in our next slide.

Thank you.