Summer Internship on IOT

Under the Guidance of:

Bibhuti Bikash Bhattacharya

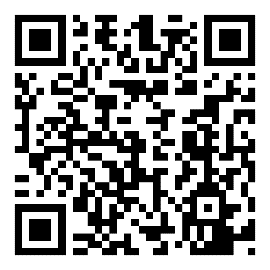
Vishal Rana Patgiri

Assam Electronics Development Corporation Ltd.,

Guwahati – 21

Submitted by: Prabhjit Kumar Dutta (4th semester, CSE),

Amity University, Kolkata.



For Project Files/Codes visit: https://github.com/PrabhjitDutta/Internship\_Project\_Files

OR SCAN:

Acknowledgement

This opportunity to undergo summer internship under AMTRON is a great chance for learning and professional development. Therefore, I consider myself very lucky as I was provided with this opportunity to be a part of it. I am also grateful to get a chance to meet so many wonderful people and professionals who led me through this internship period.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to Mr. Vishal Rana Patgiri and Mr. Bibhuti Bikash Bhattacharya for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

I perceive as this opportunity as a big milestone in my career development. I will strive to use these gained skills and knowledge in the best way possible, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

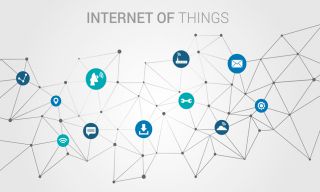
Sincerely,

Prabhjit Kumar Dutta.

Contents

|  |  |  |
| --- | --- | --- |
| SL NO | CONTENTS | PAGE NO. |
| #1 | Introduction to IOT. | 4 |
| #2 | Introduction to Raspberry Pi. | 5 |
| #3 | Temperature and Humidity data with DHT 11 Sensor. | 6 |
| #4 | Distance data with Ultrasonic Sensor. | 7 |
| #5 | Motion detection with Proximity Sensor. | 8 |
| #6 | Temperature & Humidity display using seven segment display. | 9 |
| #7 | Temperature & Humidity display using OLED panel. | 10 |
| #8 | Sensor Integrated multi-purpose alarm clock. | 11 |
| #9 | Server-Client Message Passing through sockets. | 13 |
| #10 | Server-Client File Transfer through sockets. | 14 |
| #11 | Temperature & Humidity data on local server. | 15 |
| #12 | Temperature and Humidity data display on Android Application | 17 |
| #13 | Final Project – 1 | 19 |
| #14 | Final Project -2 | 22 |
| #14 | Conclusion | 24 |

Internet of Things (IOT)

The **Internet of things** (**IOT**) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

Raspberry Pi

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core 1.4GHz CPU with 1GB RAM.

The Raspberry Pi board used in these Experiments is the Model 3B+ whose specification are as stated below:

* SOC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
* GPU: Broadcom Videocore-IV
* RAM: 1GB LPDDR2 SDRAM
* Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
* Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
* Storage: Micro-SD
* GPIO: 40-pin GPIO header, populated
* Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
* Dimensions: 82mm x 56mm x 19.5mm, 50g

Operating System used on the Device is Raspbian which is an operating system based on debian & the linux kernel and specifically designed with the requirements of the Raspberry pi in mind.

Project – #1

Reading Temperature and Humidity from DHT 11 Sensor

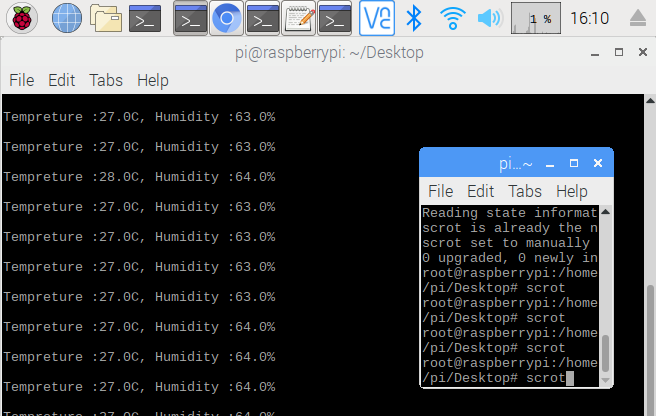
**Aim:** To get Temperature and Humidity data from the DHT 11 sensor.

**Materials Used:** Raspberry Pi Board, DHT 11 Sensor, Jumper Wires.

**Programming Language used:** Python.

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. The program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.



Project – #2

Reading Distance data with the Ultrasonic Sensor

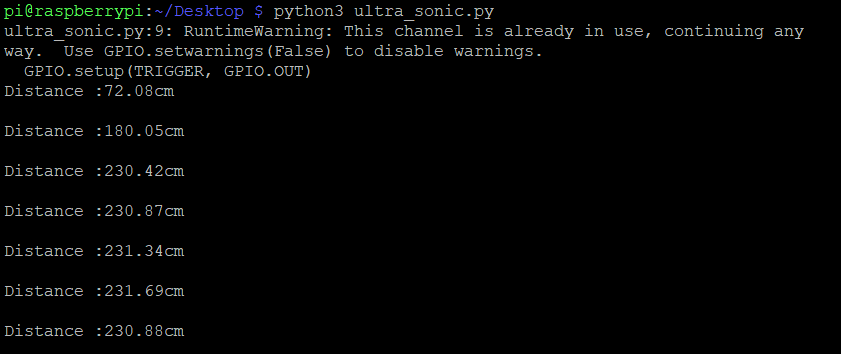
**Aim:** To get the distance data from the Ultrasonic Sensor.

**Materials Used:** Raspberry Pi Board, Ultrasonic Sensor, Jumper Wire.

**Programming Language Used:** Python

**Methodology:**

1. The Ultrasonic sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The Trigger pin and the Echo pin are attached to any of the general-purpose GPIO pins.
2. The Python program uses the **Rpi.GPIO** module to interact with the GPIO pins.
3. The Trigger pin is set as an output pin and the echo pin is set as an input pin.
4. A trigger signal is sent through the trigger pin and the program waits for the echo pin to receive some signal. Once it finishes receiving the signal, the time duration between the trigger pin sending the signal and the echo pin finishing receiving the signal is used to calculate the distance using a certain formula.



Project – #3

Motion Detection using the Proximity Sensor

**Aim:** To detect movement and turn the buzzer on in that case.

**Materials Used:** Raspberry Pi board, Proximity Sensor, Buzzer, Jumper Wires

**Programming Language Used:** Python

**Methodology:**

1. The Proximity sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin and the output pin is connected to any of the general-purpose pins. While the Buzzer’s +ve is connected to any of the general-purpose pins connection and the -ve is connected to a GND GPIO pin in the raspberry pi.
2. The Python program uses the **Rpi.GPIO** module to interact with the GPIO pins.
3. The output pin of the sensor is set as input and the +ve pin of the buzzer is set as output.
4. Whenever the sensor sends a output signal the program sets the buzzer as high thus, turning it on.



Project – #4

Temperature and Humidity Display on a 7-segement Display

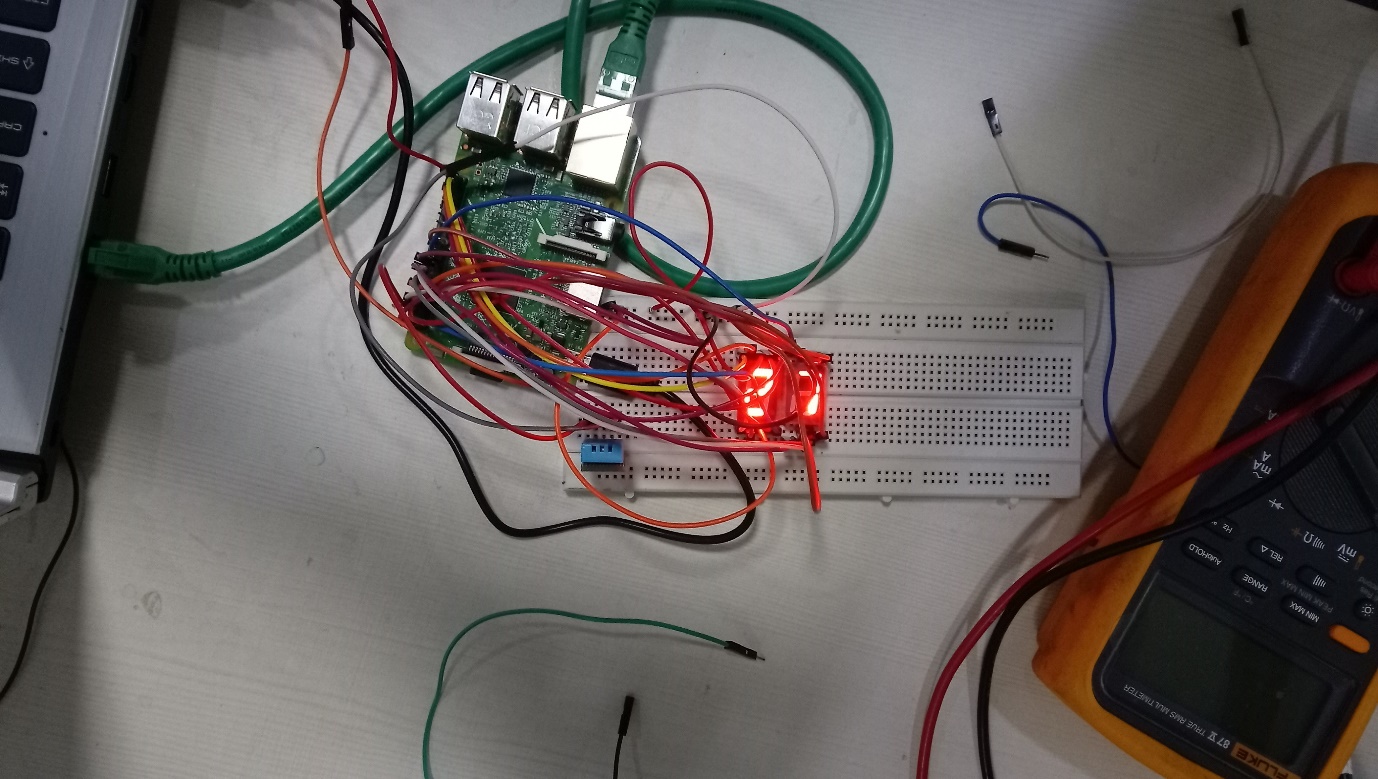
**Aim:** To display DHT 11 sensor data on a 7-segment display.

**Materials Used:** Raspberry Pi board, DHT 11 Sensor, 2 x 7-segment display, Jumper Wires.

**Programming Language Used:** Python

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin. The Common pins of the 7-seg display is connected to the 5v connection while all the other pins are connected to different general purpose GPIO pins.
2. The Python program uses the **Rpi.GPIO** module to interact with the GPIO pins.
3. All the 7-seg pins except the commons pins are set as output pins.
4. The program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.
5. 7-segment display has 8 different LEDs which can be combined to display a number. Thus, for each number the combination of LEDs need to be programmed. For e.g. for the Number 1 the LEDs (a, b) need to be turned on.
6. The temperature read is displayed on the two 7-seg display.



Project – #5

Displaying Temperature and Humidity data on and OLED panel.

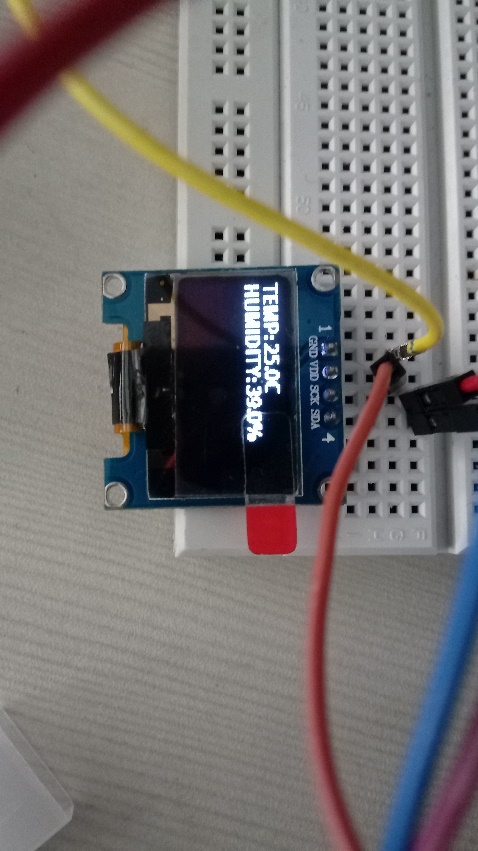
**Aim:** To display DHT 11 sensor data on a OLED Panel.

**Materials Used:** Raspberry Pi board, DHT 11 Sensor, OLED Panel, Jumper Wires.

**Programming Language Used:** Python

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. Connect the +ve pin of the OLED panel to the 5v connection and the GND to the GND of the pi board. Connect the SCL pin of the OLED panel to the GPIO3 pin and SDA to the GPIO2 pin.
3. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** module to read the input stream from the DHT 11 sensor. The program also uses the **Adafruit\_SSD1306** module to interact with the OLED panel.
4. The program uses the pillow python library to draw an image and then sets the image to be displayed on the OLED panel.



Project – #6

Sensor Integrated Multi-Purpose Alarm Clock

**Aim:** To Build and Program a Sensor Integrated Multi-Purpose Alarm Clock.

**Materials Used:** Raspberry Pi board, DHT 11 Sensor, OLED Panel, Jumper Wires, 2 x Push Down Buttons, Buzzer, Internet Connection.

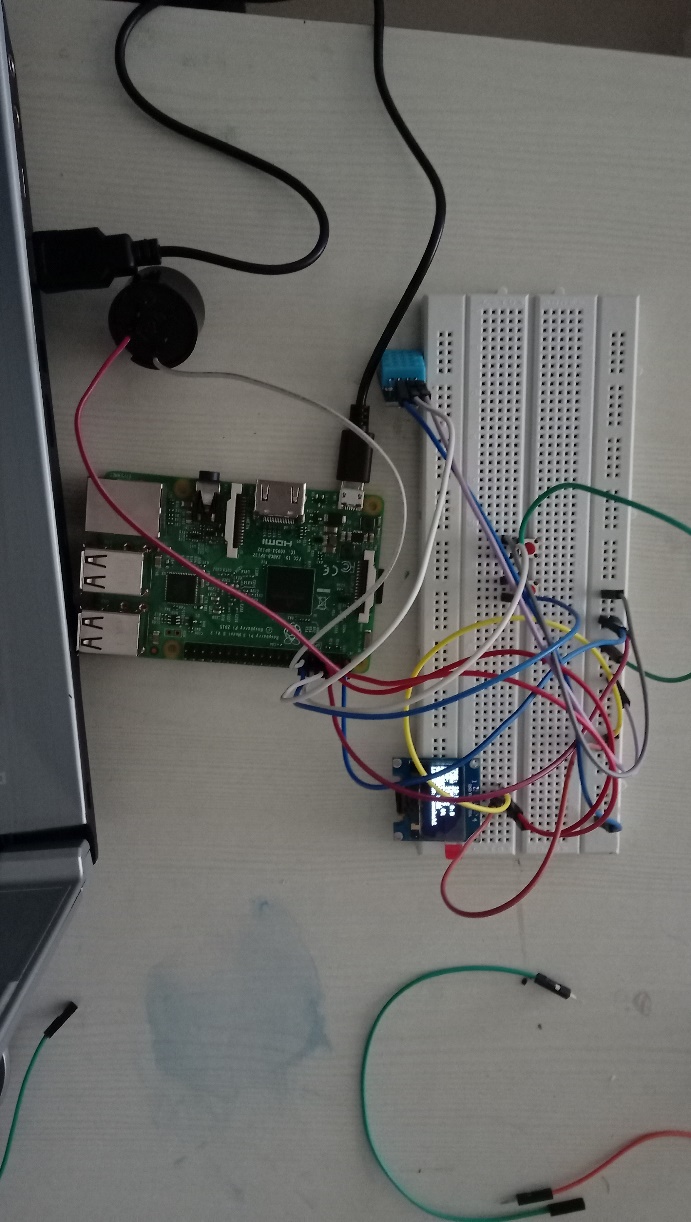
**Programming Language Used:** Python

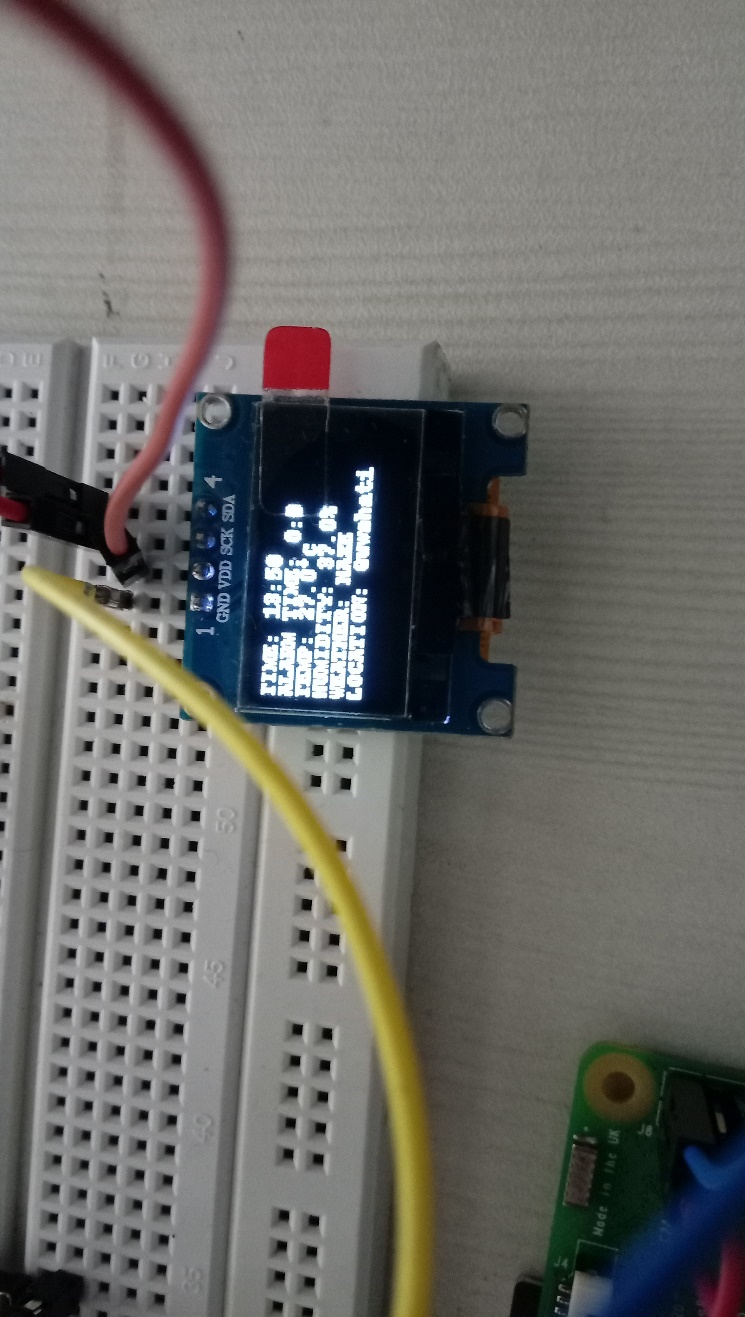
**Operating the Alarm Clock:** The Alarm Clock will display 6 different information – the time, the alarm time which has been set, the temperature, humidity, weather and the current location of the device.

There are two buttons connected to the device, the first button can be used for 3 things i.e. turning off the alarm if its on but while it is on pressing the button will result in a mode of the device where you will be able to set the hour of the alarm time, and pressing it again you will be able to set the minute of the alarm time.

The second button can be used when the first button is pressed and the mode of the device is set to set the alarm minute or hour, the second button can be pressed to change the alarm time.

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. Connect the +ve pin of the OLED panel to the 5v connection and the GND to the GND of the pi board. Connect the SCL pin of the OLED panel to the GPIO3 pin and SDA to the GPIO2 pin.
3. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** module to read the input stream from the DHT 11 sensor. The program also uses the **Adafruit\_SSD1306** module to interact with the OLED panel.
4. The Program uses the **Rpi.GPIO** module to interact with the GPIO pins.
5. One end of the push down buttons are connected to the 5v connected while the other end are connected to any of the general-purpose GPIO pins.
6. The general purpose GPIO pins connected to the buttons are set-up as input pins with their pull\_up\_down resistors set as down.
7. The buzzer is connected to one of the general-purpose GPIO pins and setup as output.
8. The program uses the ip address of the raspberry pi to find the location of the device with the help of the **geocoder** module and then uses the **OpenWeatherMap API** to find out the weather of that location this step is done every two hours.
9. The system time is found using the **datetime** module.
10. The buttons need to be programmed to work as desired.
11. If the system time and the alarm time is same the buzzer is set to HIGH until and unless the first button is pushed.
12. The system time, alarm time, the temperature, humidity fromm the DHT 11 sensor and the location and the weather found are all displayed on the OLED panel.



Project – #7

Client-Server Message passing between IOT devices

**Aim:** To Write and Program a Sensor Integrated Multi-Purpose Alarm Clock.

**Materials Used:** Raspberry Pi, Server Computer, Local Area Network.

**Theory:** The concept of IOT which is to have interconnected devices that works in sync/with each other while interacting with each other. This project implements a basic method of inter-process communication which is message passing implemented through Sockets.

**Programming Language Used:** Python

**Methodology:**

1. A Server Socket is created in the server side i.e. the computer and a client socket is created in the raspberry pi using the **socket** module. This socket is created on a certain single port. The Socket is set as STREAM i.e. is uses TCP.
2. The Server will start listening for connections, while the client will request for connection and server will accept it.
3. A Message is taken in the client side and is sent through the socket after being encoded.
4. The Message is read in the server side from the socket buffer and then decoded.

**Disclaimer:** The Program will on work for a single client-server connection. In order to implement multiple client, single server connections, the server side should be implemented with multithreading/multiprocessing or there should be connections made with multiple ports.

Project – #8

Client-Server File Transfer between IOT devices.

**Aim:** To Write and Program a Sensor Integrated Multi-Purpose Alarm Clock.

**Materials Used:** n x Raspberry Pi, Server Computer, Local Area Network.

**Operation Theory:** The concept of IOT which is to have interconnected devices that works in sync/with each other while interacting with each other. This project implements a basic method of inter-process communication which is message passing implemented through Sockets.

**Programming Language Used:** Python

**Methodology:**

1. A Server Socket is created in the server side i.e. the Server computer and a client socket is created in the raspberry pi using the **socket** module. This socket is created on a certain single port. The Socket is set as STREAM i.e. is uses TCP.
2. The Server will start listening for connections, while the client will request for connection and server will accept it.
3. Every time a new connection is accepted by the server, the server creates a new thread for that client.
4. First the program asks for the names of the files to be sent and their location
5. Then the size of the file is sent, then the name of the file.
6. Then the file is read in byte format using the same buffer as the buffer we will use to send the data of the file. The part of the file that is read is then send through the socket.
7. The data is then received by the server by reading the socket buffer. The socket buffer to send and receive the data is set as high as possible to make the file transfer efficient for larger files.
8. The data received by the server is then written in a file which is given the name which is received from the client.
9. Steps 5, 6, 7, 8 is repeated for each and every file.

**Disclaimer:** The Program will on work for a multiple client-server connection but since it is only using multithreading it will have limitations to the number of clients that can connect to the server at the same time. In order to increase the number of possible simultaneous connections, the server side should also be implemented with multiprocessing or there should be connections made with multiple ports.

Project – #9

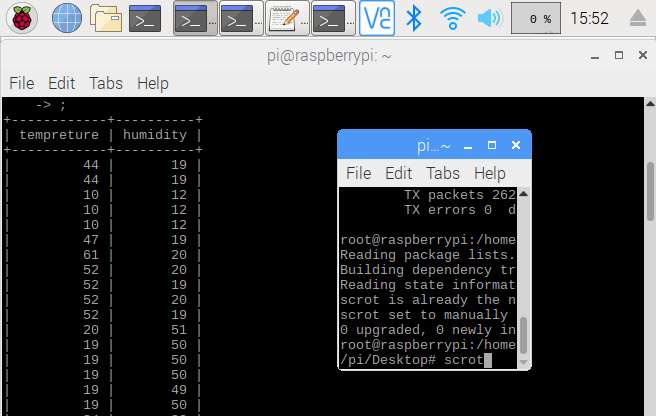
Displaying Temperature and Humidity data in a Local Server

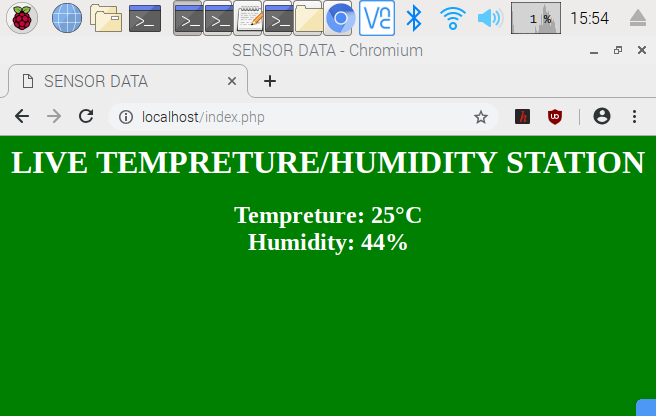
**Aim:** To Write and Program which stores the temperature and humidity data (received by the raspberry pi from the DHT 11 sensor) in a database. Then use that database to host a webpage in a local server that displays that data.

**Materials Used:** n x Raspberry Pi, DHT 11.

**Programming Language Used:** Python, PHP, HTML, CSS, SQL.

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.
3. The temperature and humidity is then updated in a previously created database in mysql using the **mysql.connector** module.
4. Then the PHP script will execute a sql query to get the latest data and display it on the local server which was created using Apache.



Project – #10

Displaying Temperature and Humidity data on Multiple Android Devices

**Aim:** To display Temperature and Humidity data on Android Application with client-server socket-based data transfer using Android Application as Client and Raspberry Pi as Server.

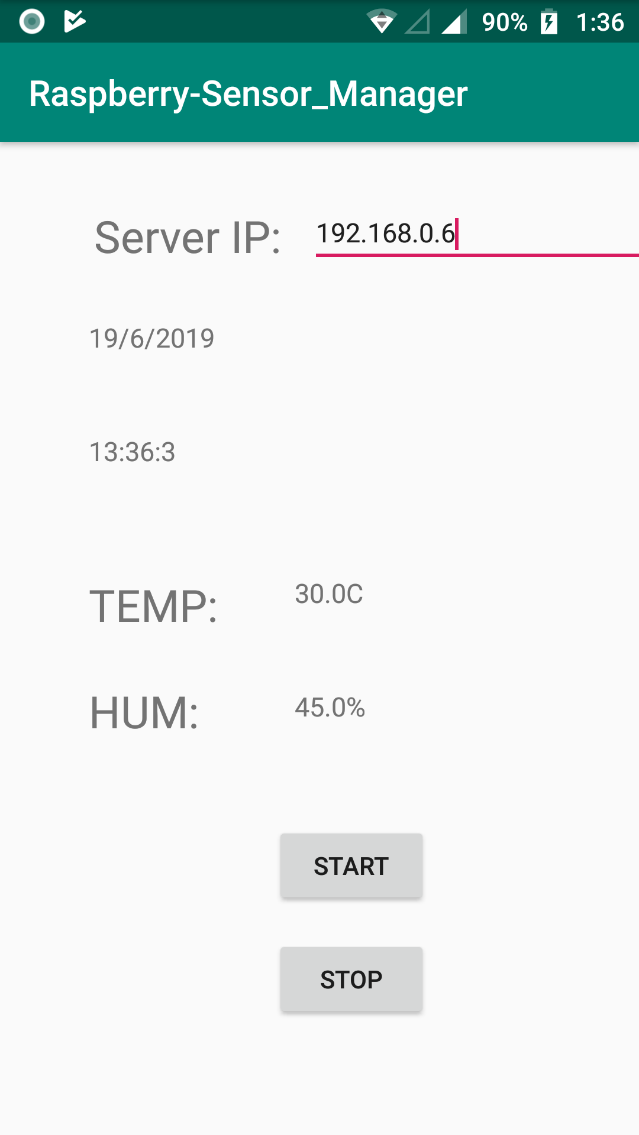
**Materials Used:** Raspberry Pi, DHT 11, n x Android Devices, Local Area Network.

**Programming Language Used:** Python, Java (Android).

**Methodology:**

1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.
3. A Server Socket is created in the server side i.e. the raspberry pi and a client socket is created in the android app using the **socket** module in python and **Socket** class in java. This socket is created on a certain single port. The Socket is set as STREAM i.e. is uses TCP.
4. The Server will start listening for connections, while the client will request for connection and server will accept it.
5. Every time a new connection is accepted by the server, the server creates a new thread for that client.
6. The raspberry pi (acting as the server) sends the temperature and humidity data to the android app (acting as the client).
7. The Android App displays the data.

**Disclaimer:** The Program will on work for a multiple client-server connection but since it is only using multithreading it will have limitations to the number of clients that can connect to the server at the same time. In order to increase the number of possible simultaneous connections, the server side should also be implemented with multiprocessing or there should be connections made with multiple ports.



Project – #11

Final Project - 1

**Aim:** To receive Temperature and Humidity data form multiple devices connected with DHT 11 sensor in a single server (On the Same Local Area Network) and store that data in a database in the Server Machine. Then use that database to show the latest data in a local server as well as in multiple Android devices connected on the same Local Area Network.

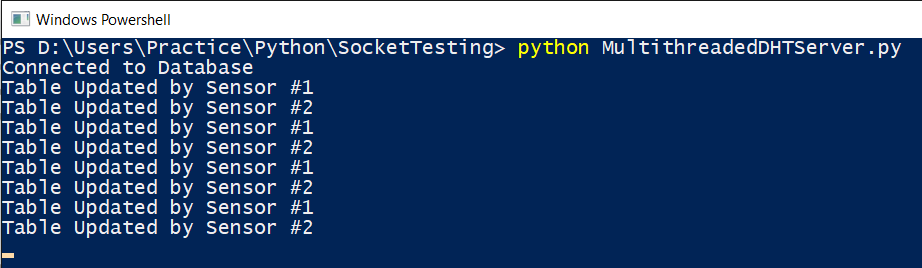
**Materials Used:** n x Raspberry Pi, n x DHT 11, n x Android Devices, Server Computer, Local Area Network.

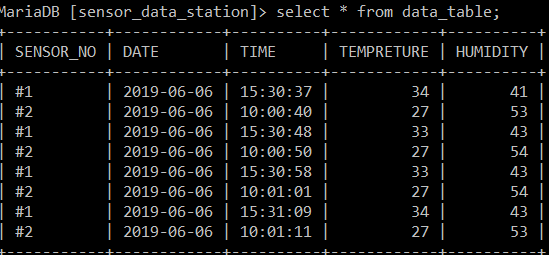
**Programming Language Used:** Python, Java (Android), HTML, CSS, PHP, SQL.

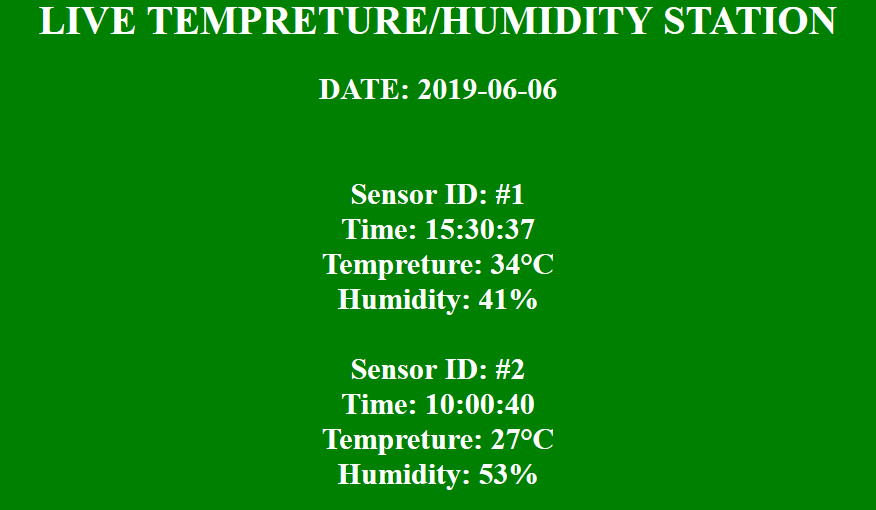
**Methodology:**

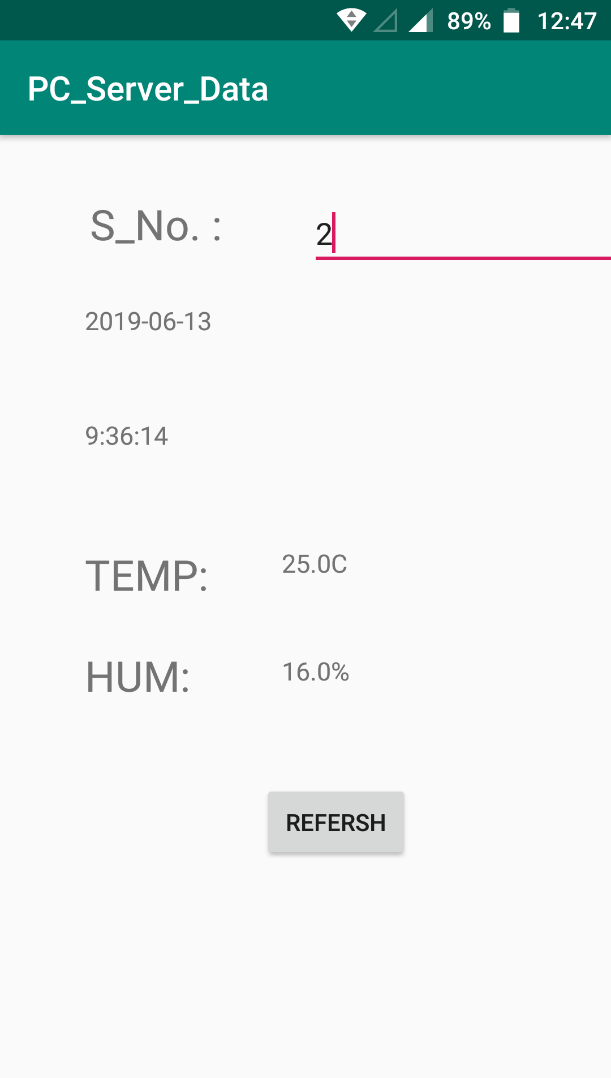
1. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
2. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.
3. A Server Socket is created in the server side i.e. the Server computer and a client socket is created in the raspberry pi using the **socket** module. This socket is created on a certain single port. The Socket is set as STREAM i.e. is uses TCP.
4. The Server will start listening for connections, while the client will request for connection and server will accept it.
5. Every time a new connection is accepted by the server, the server creates a new thread for that client.
6. The Raspberry pi sends the Temperature and Humidity data to the Server Machine through the sockets.
7. The temperature and humidity is then updated in a previously created database in the Server Machine in mysql using the **mysql.connector** module.
8. Then the PHP script will execute a sql query to get the latest data and display it on the local server which was created using Apache.
9. Another Server Socket is created in the Server machine on a different port (using socket module) and a client socket is created in the Android devices (using Socket class).
10. The Server socket sends the latest data to the android device and the Android Device displays it.

**Disclaimer:** The Program will on work for a multiple client-server connection but since it is only using multithreading it will have limitations to the number of clients that can connect to the server at the same time. In order to increase the number of possible simultaneous connections, the server side should also be implemented with multiprocessing or there should be connections made with multiple ports.









Project – 12

Final Project - 2

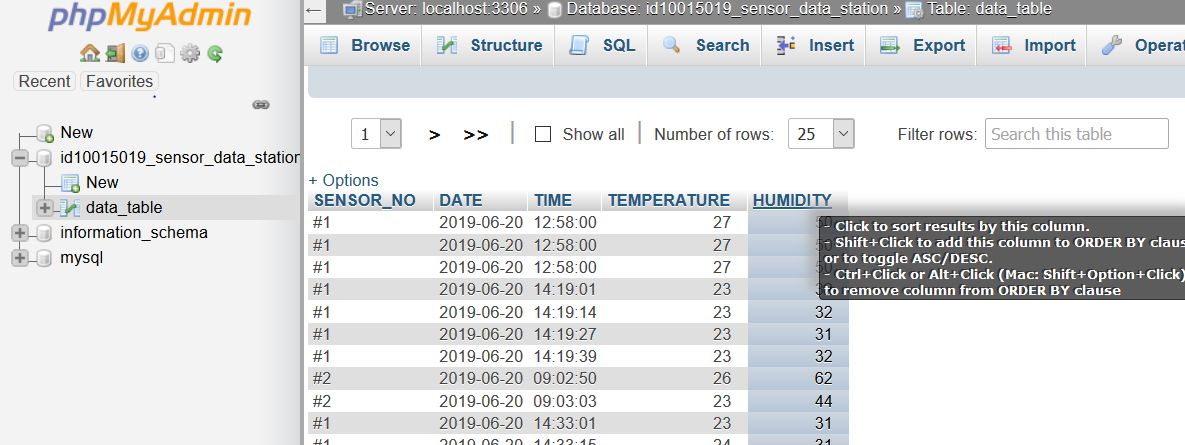
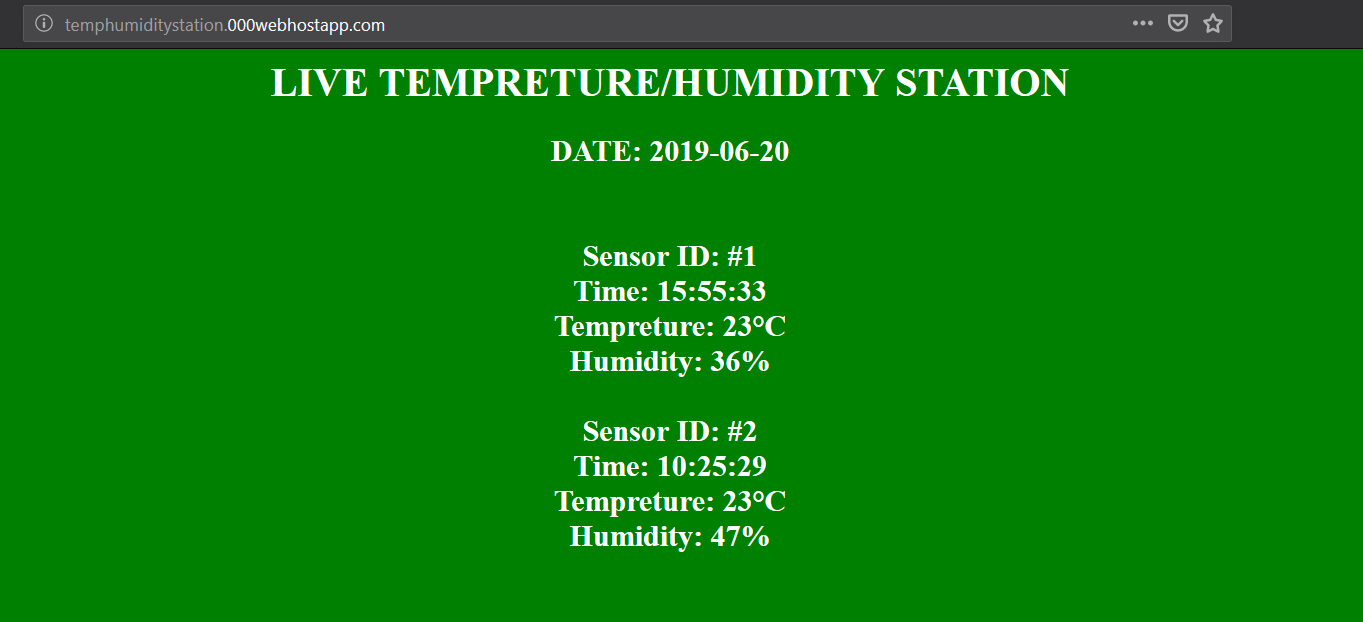
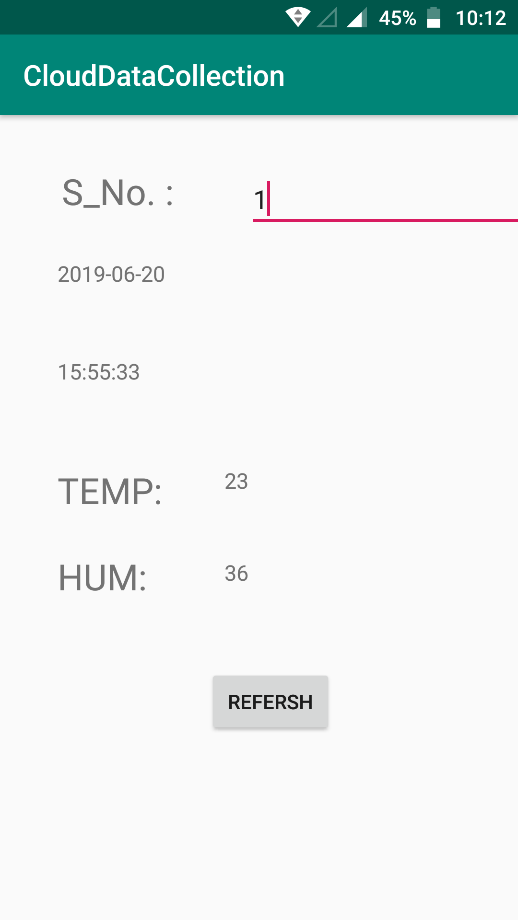
**Aim:** To store Temperature and Humidity data form multiple devices connected with DHT 11 sensor in a cloud database. Then use that database to show the latest data in a webpage as well as in multiple Android devices.

**Materials Used:** n x Raspberry Pi, n x DHT 11, n x Android Devices, Internet Connection.

**Programming Language Used:** Python, Java (Android), HTML, CSS, PHP, SQL.

**Methodology:**

1. Create an Account in the free webhosting service website https://www.000webhost.com
2. Create new website for your use.
3. Create a new database for your website. Create a new Table.
4. Go to the file manager of your website.
5. Create a new PHP script ‘get\_data.php’ that accepts data from get requests and inputs that data into the database you previously created.
6. Create a new PHP script ‘set\_data.php’ that accepts the sensor\_no from get requests and returns the data associated with the sensor from the database.
7. Create a new PHP script ‘index.php’ that displays the latest data from the database.
8. The DHT 11 sensor’s +ve pin is connected to the 5v GPIO pin in the Raspberry pi and the -ve pin is connected to the GND GPIO pin. The third pin of the sensor i.e. the data pin is attached to any of the general-purpose GPIO pin.
9. The Python program uses the **read\_retry()** function from the **Adafruit\_DHT** library to read the temperature and humidity from the DHT 11 sensor.
10. The program then makes a GET request to the ‘set\_data.php’ sending the data and updating the database.
11. The Webpage uses the ‘index.php’ file to show the latest data from the database.
12. The Android App makes a GET request to the ‘get\_data.php’ file while sending the sensor\_no required.
13. The Android App then displays the data.



Conclusion

My training in AMTRON, Assam has truly elevated my knowledge in the field of IOT and has made me confident in this field. Through this internship I have learned the various aspects of one of the most important topics of the future, which is IOT. I got a glimpse of how the real world can be merged with this technology. I am thankful to my guides and all the officials that have direct or indirect influence during the internship and have taught me new and important things which will be very helpful in my education and career in the future.