Department of Electrical and Computer Engineering

Gina Cody School of Engineering and Computer Science

Final Report (Intelligent Fog Broker System)

ELEC 6841 - Internet of Things

COURSE INSTRUCTOR:

Rodolfo Coutino

Submitted by:

Adit Shah
Id- 40172848
Email- adit.shah@mail.concordia.ca

Navaldeep Sandhu
Id- 40170635
Email- navaldeep.sandhu@mail.concordia.ca

Ishwinder Singh
Id- 40197552
Email- ishwinder.singh@mail.concordia.ca

Table of Contents

Figures

Objective

Now a days there are more and more use of IOT devices around us. The IOT industry have increased many folds in last few years. We are surrounded by appliances which can be controlled through our phones, tablet, and other devices easily. But the thing which is relatively new is using AI with IOT to take decision automatically.

In a typical household there are 2-4 people, and each have their own personal preferences on the taste of music to the temperature and lightning condition. So, to better cater to the individual needs we are developing this prototype.

The core objective of the project is to create smart home based upon personalization of user.

Introduction

IOT has become an important part of everyone’s life in this technological era. It can be used to create the smart home devices. For example, by using the mobile we can control lights and heater thermostats remotely using application. The smart home devices are now readily available to everyone, and it makes the life easier.

In our project we are more focused on personalization of this IOT devices working. We are developing Intelligent fog broker system which is used to automatically adjust the temperature of the house as a proof of concept. When the user enters in their home or their room, it will automatically adjust the temperature or lightning as per their requirement.

The working of Intelligent FOG broker system is based on the Publish-Subscribe communication and client server model. The implementation of the Publisher – subscriber model is done using MQTT which handles real time data from sensors and the client server model is used for sending high data like image to server for further processing.

What is FOG computing?

It is to bring compute near to the embedded device. In this the some of the preprocessing of the data is done on the embedded device itself before sending it to the cloud or server for the further processing of the data.

The main function of the POC is to make a personalized IOT system which will take the decision according to user preference.

Requirements

1. The device should be able to identify who the person is.
2. The device should be able to capture sensor data.
3. The device should be able to take the decision according to the identified user and capture the data to improve itself.
4. The device should be able to process the raw data before passing onto cloud for further processing of the data.
5. The process could be replicable for multiple devices.
6. User should be able to communicate with the device.

Analysis

1. The device should be able to identify who the person is.
   For this we will need the camera and a face recognition algorithm to make it work and after the face is recognized the dashboard should be updated.
2. The device should be able to capture sensor data.
   There will be multiple sensors which will be taking information from the surrounding for example the Light Intensity, Room Temperature, Pressure etc.
3. The device should be able to take the decision according to the identified user.
   Once the user is identified it should start predicting the set temperature according to the user. It shall also be able to take temperature from the user. Like if he is feeling cold then there should be provision where he can enter the desired temperature. And this information is captured to further train and personalize the prediction for better accuracy.
4. The device should be able to process the raw data before passing onto cloud for further processing of the data.
   The important concept is here we are showing that some of the preprocessing can be done at the device end before sending it to the local server.
5. The process could be replicable for multiple devices
   The code should run on multiple devices and can be reused so that new sensors can be added.
7. User should be able to communicate with the device.
   The device should have a basic UI so that he can pass values and get output.

Scenario based analysis

- **Scenario 1(Preregistered user):**

  When a person comes into the room, he will go the user interface. Then by clicking the capture button the device will identify the person's name using face recognition and set the temperature using the previous information on which AI model is trained on.

- **Scenario 2 (New user):**

  When a new user comes, he will first register himself to the device. Then there will be pre-set temperature already set on the device. If he is willing to change, he can change the temperature. When he changes the temperature, the change is noted into the data base under his profile. As he goes on using it will keep on noting down the changes.

- **Scenario 3(Old user not satisfied with the temperature):**

  The user comes capture himself and detects the person depending upon on that it sets the temperature and updates the temperature automatically every hour and if he is not satisfied, he can update the temperature and that is stored in the database. New model is trained once it has specific number of entries.

Design

To design the whole system, we have divided the project into four sub parts.

1) Sensing
2) User interface and Dashboard
3) Intelligence
4) Communication

Sensing

Sensing is to take inputs from user and sensors. Then the data to be saved at some database for the future use.

For our project there are 3 input sensing devices

1) BMP180 for pressure and temperature
2) TSL2561 Light sensor
3) Camera

User Interface and Dashboard

It is the user interface where the user can see and add user and also set temperature and can monitor various parameters.

Intelligence

Mainly in the project there are two type of intelligence algorithm which is required to function.

1) Face Recognition
2) Regression algorithm

Communication

There are two types of communication depending upon the use

1) MQTT for the sensors data to the broker

2) Local Server for sending the photos to add new user using socket programming.

Design Decisions

1) The basic requirement of our project is to create a client-server model and to increase the better interaction between the client and the server, we will use the broker which will act as an intermediate between the client and the server. So, we use the Publisher-Subscriber communication model which matches with our basic requirement. We want a broker which will follow the protocol of IOT. So, for that we are using adafruit server which is using the MQTT protocol.
2) The face detection will be done on the Raspberry Pi as this will reduce number of actual image frame to be sent to the server and thereby reducing the load on the network.
3) We will be using the socket programming based on the client – server model for sending the image pickle when needed to and add a user and capture new photo.
4) Using the local server in place of cloud server for showing the privacy concern about the photos of user.

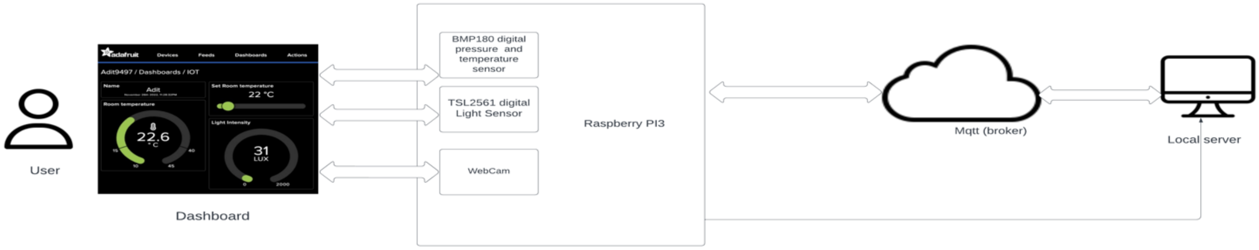


Figure 1 Architectural design

Implementation

Intelligent Fog Broker System uses Raspberry PI3 is mainly used as the centralized board. All the sensors including BMP180 digital pressure sensor, TSL2561 digital light sensor, BMP180 digital temperature sensor are inserted inside the Raspberry Pi model. The Raspberry Pi model is then connected to the broker which is a mosquito server, and it uses the protocols of MQTT to connect to the local host. We have also added dashboard to display all the data being fetched by the sensors from the servers. The subscriber can also change the parameters such as

Temperature, Pressure, or the intensity of light from the dashboard and then the data will be saved on the server.

When a user enters the room, the camera sensor will recognize its face and capture the picture of the user and then send it to the local server. The server will run facial recognition it will send back the information to the broker. Then the device which is subscribing to the name feed it will change and use that profile.

Hardware Implementation

In order to implement the intelligent fog computing system, we have used the following components whose detail is listed below:

- **BMP180 Pressure sensor**: It is a digital pressure as well as Temperature sensor which is used to measure the absolute pressure of the air by using I2C communication protocol. Its working depends upon the weight of air and based upon its weight; the pressure can be measured. It can be formed by using piezo-resistive technology and it offers high accuracy and long-term stability.

  Technical details of the sensor are:

  - Its PCB size is 2.3*2.3 cm.
  - Pressure can be measured between 300-1100hPa.
  - The temperature can be measured -40 to +85^{o}C.
  - The input voltage ranges from 3 to 5v and the voltage will be DC.
  - When the output is connected to I2C interface, it can act as a temperature sensor.

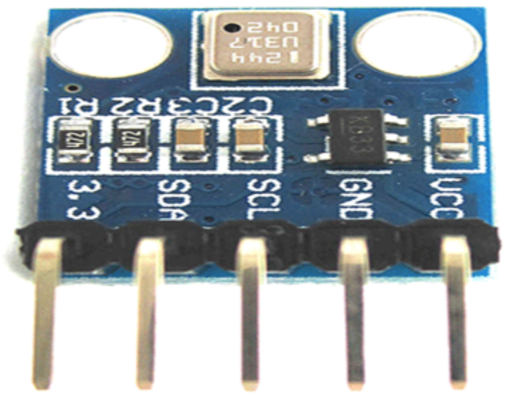


Figure 2 BMP180 Pressure sensor

- **TSL2561 Light sensor**: It is basically a type of photosensor which is used to measure the intensity of light. This sensor consists of infrared and full spectrum diodes which helps in measuring infrared, full spectrum and the light which is visible by human eyes separately.

  Technical details of the sensor are:

- The temperature should range from -30 to 80^{o}C.
- Its voltage should range from 2.7 to 3.6V.
- The interface should be I2C.
- The detected light ranges from 0.1 - 40,000 Lux. (2)

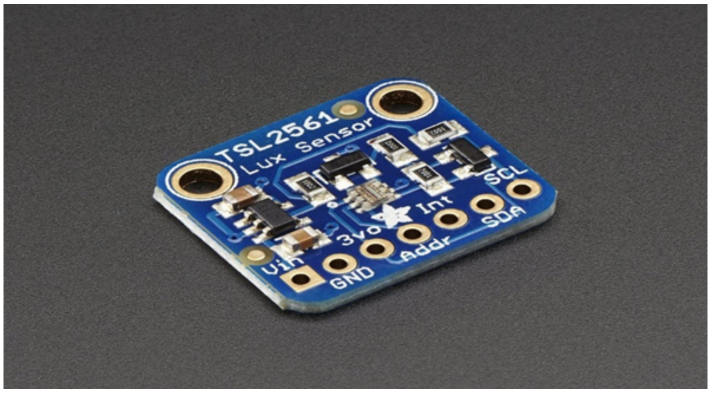


Figure 3 TSL2561 Light sensor

- **Raspberry PI3:** It is a third-generation model of a single board computers which is used for creating the hardware projects such as home automation. It usually runs on the Linux operating system, and it has certain components which then helps in making the IOT devices.

  The technical details of Raspberry PI3 are:

  - 1.2GHZ quad-core 64bit Arm Cortex-A53 CPU.
  - 1GB SDRAM.
  - HDMI Port
  - Four USB 2.0 Ports
  - CSI-2 camera connector. (3)

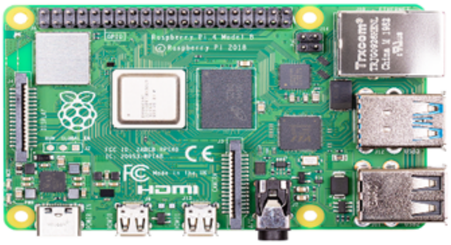


Figure 4 Raspberry PI3

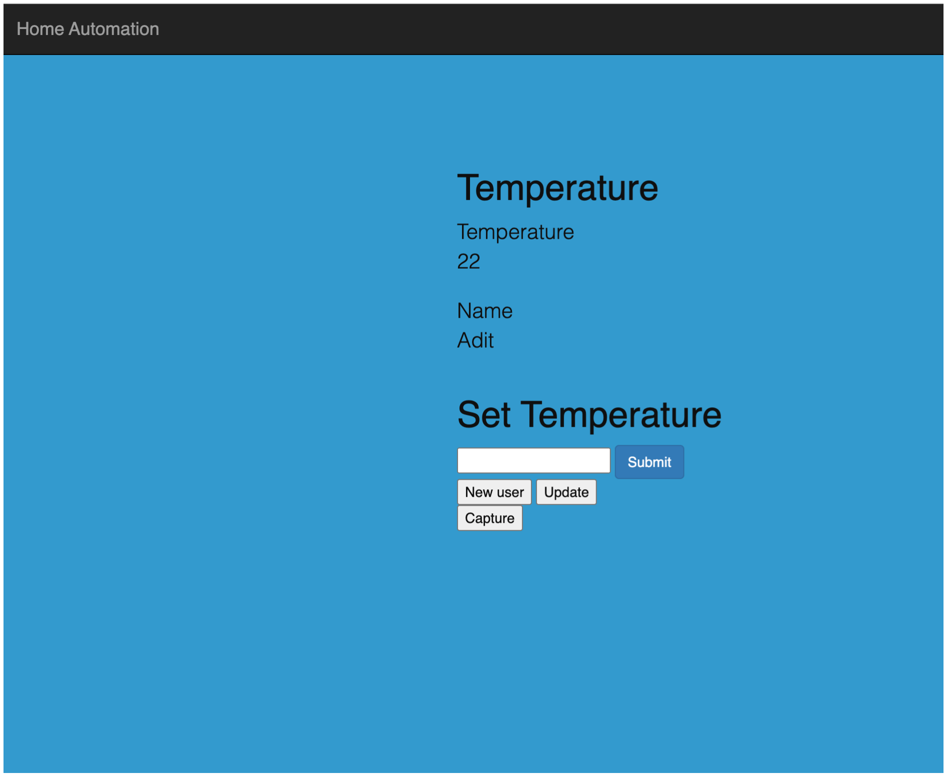
- **Webcam**: It is a video camera which is usually inserted in the gadgets such as laptops. It is designed for recording and to capture the picture. There are two types of Webcams namely basic webcams and another one is the IP Network Webcam which is used for the networking of cameras through internet.

Figure 5 Webcam

Software Implementation

The intelligent fog computing system uses the following software for its implementation as an IOT device:

- **Dashboard:** It is basically a collection of webpages which is used to display the data from the local server to the screen. It can be built by using the programming languages such as HTML and CSS. The backend is made using the python flask which will route the webpages according to the need.



Temperature

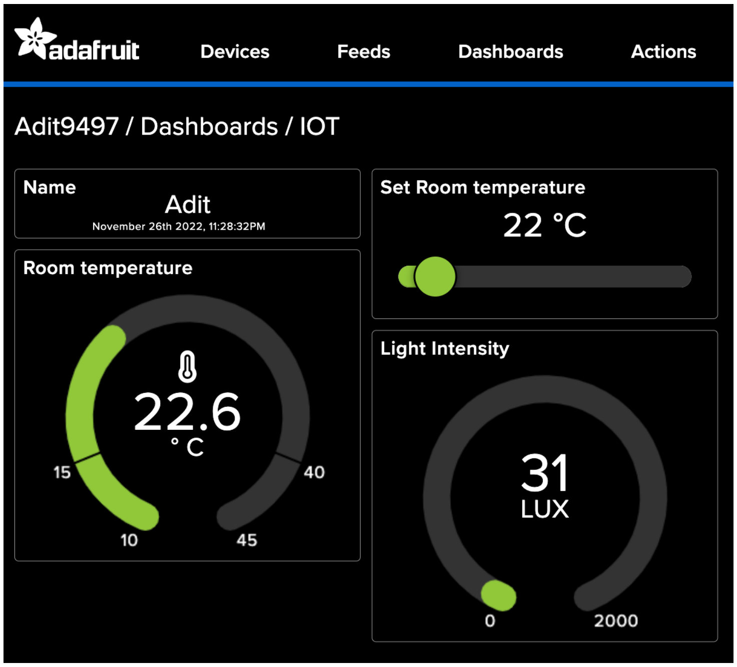
Temperature
22

Name
Adit

Set Temperature

Figure 6 Sample GUI

- **Adafruit server**: The adafruit server can be operated with the help of MQTT protocols. It is basically an open-source message broker means it can be operated on any of the operating system. As it is open source, it can be used for all the devices starting from low power devices to the devices which contains full power



Adit9497 / Dashboards / IOT

Name Adit
November 26th 2022, 11:28:32PM

Set Room temperature 22 °C

Room temperature 22.6 ° C

Light Intensity 31 LUX

Figure 7 Adafruit server

.

- **Local Server:** It is a server which is running on a local machine it has opened port for the socket. When the image is captured, the image is sent to server from the client. Then the local server will run the image recognition as soon as it detects the image it will share the result to the broker cloud and which is then available to the device.

Protocols used: The intelligent fog computing system uses following protocols for the communication.

- **MQTT protocols**: It is a network protocol which is basically designed for the transfer of message from one device to another device by using the publish-subscribe communication model. It is used in the devices which has the lower bandwidth, and it usually runs on the transport protocols. We have used MQTT protocol for communication between Raspberry Pi, MQTT broker and local server.
- **Client-server protocol**: It is basically used for the communication between the subscriber and the server. The standard protocol used for the communication between the client and the server is the Transmission control protocol i.e., TCP/IP. This protocol has been used for Raspberry Pi and local server communication.

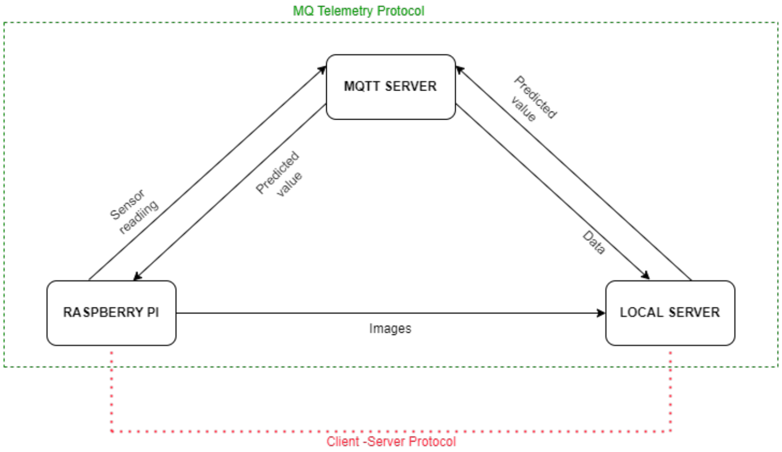


Figure 8 Implementation

Lessons Learned

- Initially we were looking for various cloud MQTT server like HiveMQ, AWS IoT Core. We decided to go ahead with AWS IoT Core but later we switched Adafruit and implemented our whole project on that. We got to learn and work on both AWS IoT core as well as Adafruit.
- In this project, we have used sensors like BMP180, and TSL2561, and for these sensors to work according to our requirement few libraries were required. As one of the libraries for the sensor BMP180 that is BMP085 was no longer available, so we had to take the code for the sensor and use the code for working this sensor. While resolving this issue, we gained understanding about this workaround as well as new libraries
- As we have integrated AI with IoT so to solve the complexity, we used multi-threading which helped in the smooth execution of multiple functions concurrently. Multi-threading techniques enabled efficient communication between different functions or threads.

Conclusion

- Smart homes enhance the quality of life and increases the comfort. Smart home has appliances, heating, lightning devices which can communicate with each other and act accordingly. This is possible through integration of IoT with intelligence embedded sensors.
- The purpose of our project to build a personalised system which could set the parameters according to the user. It can monitor and control the device specific to the user.

Future scope

- As for now, our model has been trained by few user inputs only, but we would like to extend our implementation by accepting multiple user inputs and introducing multiple sensors. We intend to measure the heartbeat and number of steps by taking data from gadget like smartphone or smart watch.
- We would also like to incorporate voice-controlled assistant and controlling the smart home devices. With this each sensor or device can be connected and managed according to users want.
- More data can be collected, and model training can be done using MLops optimisation with the evolutionary algorithm can be used to change how the model will predict and according to need.
- Even reinforcement learning can also be tested to make the system and the local server can be shifted to cloud with use of better data protection.

References

- https://www.adafruit.com/BMP180/BarometricPressure/Temperature sensor/1603#description.
- https://learn.adafruit.com/tsl2561.
- https://seeedstudio.com/Raspberry_Pi_3_Model_B/
- https://www.logitech.com/en-ca/products/webcams/streamcam.960-001286.html