IoT Based Home Automation Using App & AWS

Aman Kumar Singh, Abdullah Ahmed Arifi, Harikrishnan R, Bhuvi Datta and Shivali Amit Wagle

Dept. Electronics and Telecommunication Symbiosis Institute of Technology, Symbiosis International Deemed University Pune, India
E-mail : happy90hp@gmail.com, dr.rhareish@gmail.com, bhuvidatta@gmail.com,kulkarni_shivali@yahoo.co.in, a3kube@gmail.com

2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) | 978-1-6654-9529-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ACCAI53970.2022.9752635

***Abstract*- For time-saving and making our day-to-day life much easier, home automation was introduced. As the devices are connected to a common instructor, together they form the Internet of Things (IoT). Our topic 'IoT Based Home Automation Using App & AWS', is bringing Raspberry pi, AWS, and MQTT App all together under one umbrella. This automation will be managed using an App , and the data of it will be stored securely using AWS. Our aim in this project is to control the devices smartly, keep the sensitive data of the automation securely in such a way that only the user can access it. All this can be achieved by mixing Raspberry pi, AWS , and MQTT App. Integrating IoT home automation with AWS is a great scoop for data protection , and management, as the user can choose who to share the data with. The devices such as lighting, predicts and can turn on/off and also can be changed remotely. As for the fan, it predicts the change in the temperature and changes its speed(motor) accordingly, and storing temperature data securely in form of a Table in AWS DynamoDB. The door sensor used here sends a notification to the user if someone is breaking in or if the door is open using SNS (Simple Notification Service). Thus, saving up time and unnecessary worries by keeping a constant check on the details using App**

Keywords— Internet of Things, Raspberry Pi, MQTT Dashboard, Amazon Web Services, AWS IAM (Identity and Access Management), AWS DynamoDB, AWS SNS (Simple Notification Service)

I. INTRODUCTION

To make the user's lifestyle easy and more feasible, home automation was proposed back in 1966. Jim Sutherland invented ECHO IV, which was controlling temperature and other appliances. Jumping to 2021, as the new technology sweeps in, home automation systems have attracted many users in the market due to the advancement of communications technology. Home automation is the subgroup of everyday computing that includes smart technology whose main use is to provide comfort, security, smart lifestyle and energy reduction to the user. All this is possible by bringing in the Internet of Things (IoT). Over the past few years, IoT has taken over the advancing technology becoming the most important technology of the 21st century. Internet of Things is described as the interconnection of hardware with sensors and software, that runs by collecting the data and exchanging it with the devices over the communication network.

Home automation using IoT collects the data of the house using various sensors and exchange this data with the other devices which function accordingly with the user's preference.

This research paper presents an IoT-based Home Automation system that controls lighting, speed of the fan, notifies about the main door security, securing the sensitive data of the house. We can monitor these devices using our smartphone. The lighting of the house is controlled using App, whereas the speed of the fan changes depending upon the change in the temperature and humidity of the surrounding environment and send notification through AWS in mobile whether the door is open or closed. In many research papers, new forms of home automation with different ideas have been put forward. The potential of combining machine learning with IoT helps in achieving an intelligent control system [1]. Another way, the sensors and actuators are connected to the NodeMCU controller, which updates the data to the IoT server [2].

In most home automation systems, the lights turning on/off using voice command or clapping or simply instructing through the app is quite common, but when the user is outside or unable to give a signal, it can be problematic. As seen in many other home automation, changing the speed of the fan can be done using an App, but sometimes it does not fit according to the temperature at home. Also seen, many IoT-based security systems, put a burglary alarm, which sometimes for the slightest reason gets triggered again causing a nuisance to the user.

With the advancement in technology, personal data theft has become quite common. Home automation data is quite sensitive and very personal, having home automation in our hands also means exposing your personal data and space. These data can be used and seen by external users too. To overcome this we are using AWS, i.e., Amazon Web Services to store

our sensor's readings. AWS is a subset of Amazon providing cloud computing platforms, it protects the sensitive data of the user. There are no external viewers of data, keeping it limited to access, which also will be decided and tracked by the user with who to share it with. Users can access the data using AWS IAM, i.e., AWS Identity and Access Management. AWS IAM helps the user to manage the access of their AWS services and resources securely. It has three main components: user, groups, and permissions. Using this, we can create, manage other AWS users, groups, and deciding whether to grant permission or deny their access to AWS resources. To create Tables, we have used AWS DynamoDB, it is a fully supervised NoSQL database service , and it provides much faster, smoother, and predictable performance. With the help of DynamoDB, we can design the database tables that will store and get any sum of data at any degree of request traffic.

In this paper, we have tried to keep the automation simple but effective, we are introducing the lighting control by application and depending on the surrounding. If the surrounding is dimly lighted or dark the lights will turn on and vice versa. Also, if users want more intensity of light, using App they can control light by switching on/off. Alongside, we have used a sensor to depict the temperature and humidity of the surrounding, this data will be sent to AWS to be stored in Tables. Using these readings, the motor of the fan will change accordingly. Security is important both physically and digitally, thus for the main door, instead of a burglary alarm, that not only requires constant maintenance but can be sensitive too, a notification will be sent to the user whenever someone is breaking in or if the door is kept open for too long. To send notifications we are using Amazon SNS, i.e., Simple Notification Services. SNS provides the message exchange from the publisher to the subscriber. Publishers, i.e., consumers communicate with subscribers asynchronously by sending messages through logical access points and communication channels.

II. Literature Survey

The 21st century is all about living your life with smartness, let it be driving or teaching or learning, but getting to your very basic daily routine and living it with smartness. Home automation is all about interconnections of physical devices with their sensors and software.

Home automation also called smart homes to defines itself as a technology that helps in settling an environment that provides the user with comfort, convenience, energy efficiency, and security. With technology at every corner, people are highly dependent on it, so making a more positive and smart dependency on technology is what pushes the Smart homes in the market [3]. As the technology gets more advanced, the cost increases rapidly, and it becomes more complex, making it less feasible to reach more people, making this the biggest con, and challenge for home automation [4].

Internet of things or IoT is a new technology that represents the future of technology let it be in daily life or big fields, for it touches everyone [5]. Getting wireless helps in controlling the basic appliances automatically using the internet from wherever or whenever [6]. After reading many reports, we got a great insight into the evolutions done using IoT and how it took over home automation. The direct connection of user's smartphone and their appliances is itself the beginning of a new era that is defeating the long tradition of using appliances manually [7]. Home automation has its pros and cons, but with so many changes and new additions it is shifting more to pros, home automation is seen as vital for the elderly and disabled too, as the environment becomes more feasible, and hassle-free. To get over such challenges, solutions are coming in a hybrid form combining the benefits of different platforms to bring out a desirable result [8].

For many patients, elderly, and disabled, living a hassle-free life is just in one click where they enjoy their privacy, and practice their independence without relying on others [9]. The added innovations in home automation provide a comfortable and flexible usage to users regardless of their conditions and at a much lower rate, as it booms in the market [10]. IoT grows exponentially with the need of society by monitoring factors in the surrounding [11].

In the world of IoT, combining it with machine-to-machine (M2M) brings in more opportunities and creative ideas to implement in automation [12]. Combining Machine learning with home automation can achieve next-generation smartness and intelligence that can also help in conserving energy [1].

One of the hurdles faced by a smart home is how it takes up the data and gives the prediction according to the user's liking, making the user's behavioral prediction quite a valuable asset in home automation [13]. Deep learning, a subset of machine learning can be one of those solutions to provide a better analysis that matches perfectly with the user's behavior [14]. The smarter homes get, the more energy gets used up, this can be said is another con home automation face. To provide a lesser energy consumption many researchers and papers are working on this, as energy is an important design resource [15].

IoT-based home automation should be designed in such a way that it should lessen up its energy consumption, be hassle-free, and less complex [2]. With all these data out there given to the internet, just for the betterment of smart living, another main criterion is to make sure the user's personal data remains safe, and secured and not get used by other platforms. Data theft is becoming quite a common thing in today's era. As present-day home automation is all about the combination of computing and wireless sensors, there is a high expectation to provide much safer means and restricted access, and this has become a new hurdle in today's home automation [16].

Amazon has introduced several new services in the market, that can be useful in changing the data game for the future. Home security is a vital part of smart homes, i.e., security physically and digitally is being constructed by integrating AWS and IoT [17].

There are many cloud computing platforms, but with most of them undergoing some or the other up-gradation or other related problems, AWS in comparison to them is much more viable and stable and brings in all the desirable materials that are required [18].

III. Proposed Work

In this project, we are using RPi, and firstly integrating it with AWS. For lighting, LDR is being used to detect the abundance of light around, alongside we are connecting the lighting with the MQTT app. For temperature, we are using the sensor DHT11, which takes the temperature and humidity readings and takes this data to DynamoDB. The software used over here is Python, MQTT Dashboard, and AWS services. The coding for LED-LDR was done in python as well as MQTT Dashboard, the coding for DHT11, and door sensor was done using python and their data was sent to DynamoDB

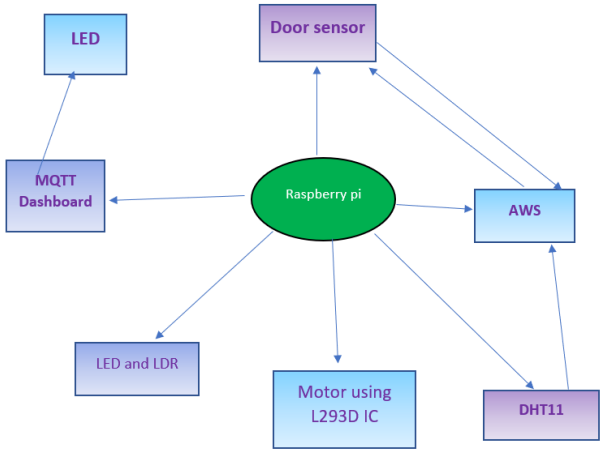


Fig. 1: Block diagram of the workflow

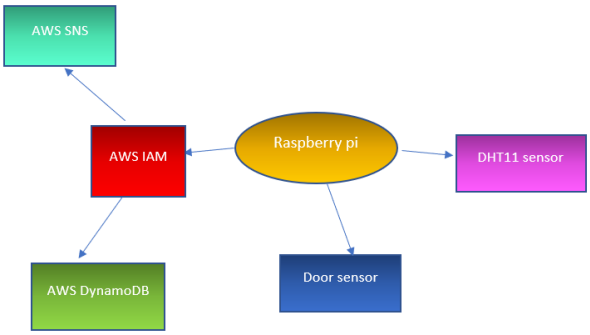


Fig. 2: Block diagram of AWS working

The direction of the workflow is as follows:
As the RPi is the central head of the project and it is physically connected to the sensors, alongside it is integrated with AWS CLI using the terminal of the RPi. The python programming is done in RPi itself using the python IDE, as the terminal is also used for AWS commands.

For lighting, LDR takes up the amount of light that it catches in the surrounding. According to our code using the if-else statement, if LDR catches any amount of high abundance light, LED is instructed to switch off. Vice versa, as the LDR catches a dim amount or no light, it instructs the LED to switch on. Alongside this, the MQTT Dashboard app is also connected to the LED, so that the user can switch it on/off according to their preference. In MQTT app, there is a button that the user has to click to operate it.

The sensor DHT11 is programmed using python. The temperature and humidity data are taken from the sensor at a regular interval and are then pushed to the DynamoDB table, i.e., created for storing these data. The AWS is the service provider over here that we linked, the services such as DynamoDB, that stores the door sensor, and DHT11 data in the tables created. This can only be accessed by users authorized by IAM service.

Model and sensors used over here are as follows,

A. HARDWARE REQUIREMENT

LED: Light Emitting diode emits lights whenever current flows through it.

LDR: Also known as photoresistor that decreases resistance with respect to light received on its sensitive surface.

Motor: LDR is also known as a photoresistor that decreases resistance with respect to light received on its sensitive surface.

*1) **Raspberry Pi:***

Raspberry pi 3 model B+ is a tiny computer that can be used to learn programming. It has a 1.4 GHz 64_bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, Ethernet, 4 USB 2.0 ports and 40-pin GPIO. It is a 16-pin IC. It is used to drive motors. It can run two DC motors in independent directions at the same time.

*2) **DHT11:***

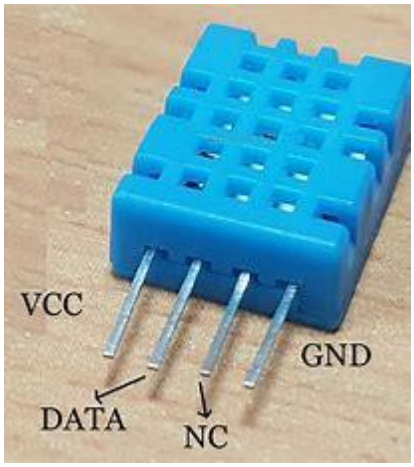


Fig. 3: DHT11 Sensor

It is a commonly used temperature and humidity sensor that is used to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

3) Door Sensor:

Fig. 4: Magnetic Door Sensor

It is a magnetic door sensor that is used for opening/closing the door. One part will be connected to board and other part is free/movable.

B. SOFTWARE IMPEMENTATION

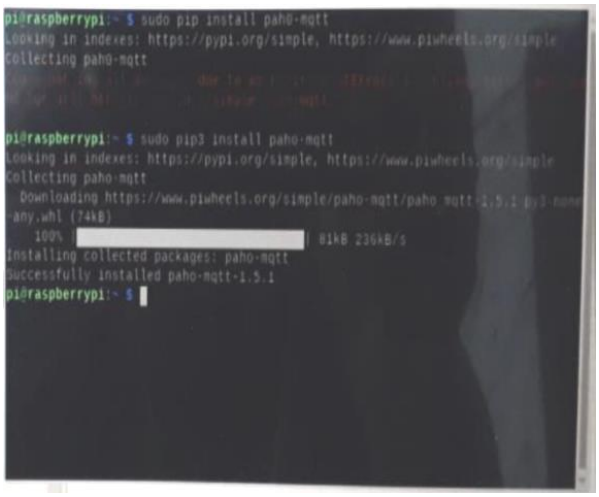


Fig. 5: Installing paho-mqtt libraries

We began our project by installing Raspbian OS and boosting up our raspberry pi, we faced some common errors that occurs at the beginning of the installation. After this, we successfully managed to run the Raspbian desktop. At first, we installed and updated python 3.7 files in Raspberry pi, and AWS libraries, followed by AWS command-line interface (CLI) installation in the terminal, where we used the commands,

pip3 install awscli --upgrade --user
aws –version

For controlling lighting using MQTT app, we installed paho-mqtt, as shown in Fig. 5,
Using commands,

sudo pip3 install paho-mqtt

Paho-MQTT provides a connection between the user's application and the device. The one publishing/sending the message is called broker and the one receiving the message is called subscriber.

We faced some issues in between but it was solved by installing urlparse 2 and urlparse 4,

Using commands
sudo pip install urlparse
pip install urlparse4
pip install urlparse6

Urlparse takes up an URL string, resolves the string and returns the object with every part of the address having its own properties. It provides a more uniform way to store resources.

The next task in hand is to create a new user in AWS IAM, where access keys and secret access would be displayed. This was done by going to the site and filling in the required details, afterward it will guide us to a welcome page where it shows that the account was created successfully, as shown in Fig. 6, with the user's name, Access Key ID and Secret Access Key.

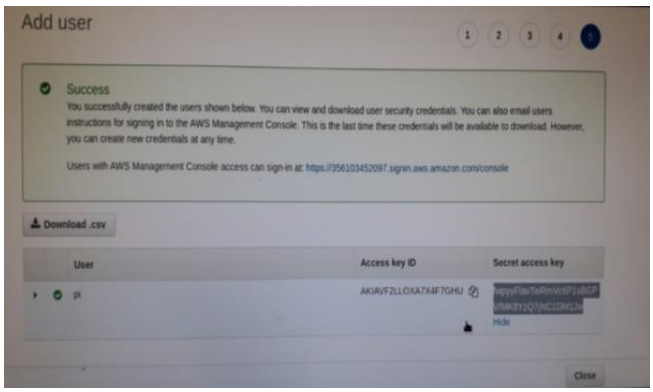


Fig. 6: User creation in AWS IAM

AWS DynamoDB is being used to create Tables for door sensor(door_sensor) and temperature (DHT11) change data, to access these tables from the terminal we used commands,
aws configure
aws DynamoDB list_tables

For door sensor we had to create a topic named doorSensor to send notifications using AWS SNS, as shown in Fig. 7, with the details of the topic,

To configure the message for SNS from the terminal we had to use the command,
sudo nano message.txt
cat message.txt
aws sns publish --topic-arn "arn:aws:sns:ap-south-1:356103452697:doorSensor"

As seen in Fig. 8.

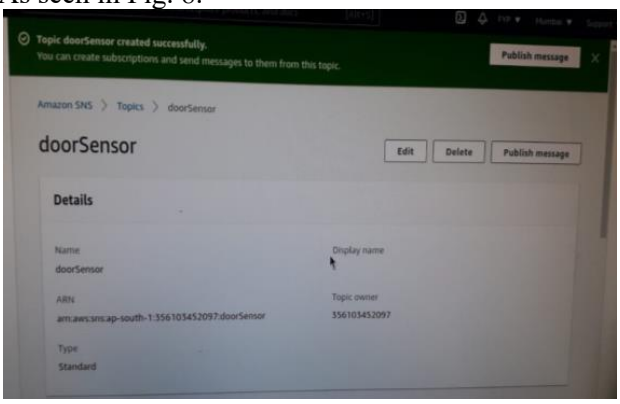


Fig. 7: Topic creation in AWS SNS

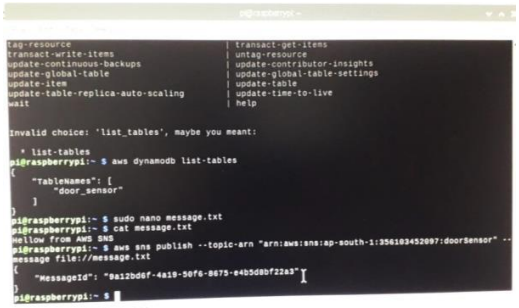


Fig. 8: Creating notification file for AWS SNS

In hardware connection, we have used jumper wires to connect the pins of Raspberry pi to the sensors and other devices at breadboard. As shown in Fig. 9 below, the breadboard connection holds LED-LDR, DHT11 sensor, door sensor, and Motor with a Battery.

The inputs and the outputs of the project are as follows,
1) The first sensor is the Lighting (LED – LDR)
The input, in this case, is the surrounding light from the environment, if the light is low or insufficient then the LDR will switch on the LED. And if the light from the surrounding is sufficient then the LED will turn off.
2) The second sensor is the Lighting using MQTT.
Here we have used the LED to represent A Light that can be controlled from the MQTT Dashboard application. The interface of the application is easy to use and simple. There are two buttons in the

application labeled "Light On" and "Light Off". Users can operate the LED using these.

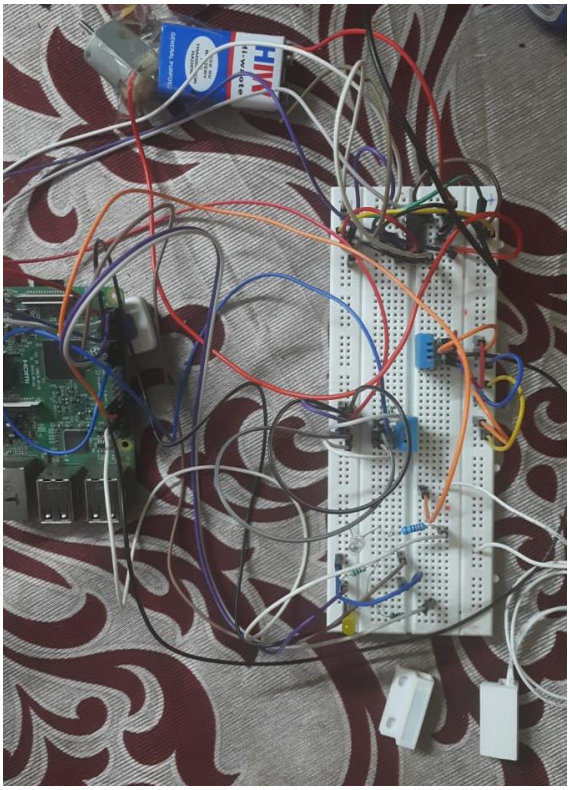


Fig. 9: All hardware components connected

3) The Temperature change sensor (DHT11-AWS)
Here the DHT11 sensor is used to keep track of the temperature changing in the surrounding. The DHT11 sensor is also integrated with AWS to keep track of the temperature and humidity changes in the AWS DynamoDB. The AWS and DHT11 are connected through AWS CLI on Raspberry pi here the temp data will also be stored in the DynamoDB table.

4) Motor-DHT11 sensor
The motor is used to represent the fan. The motor is driven by the data provided by the sensor and the speed of the motor is changed according to the temperature.

If the temperature is between 5 and 15 the motor will rotate at the speed of 20%. If the temperature is between 16 and 23 the motor will rotate at the speed of 50%. If the temperature is between 24 and 30 the motor will rotate at the speed of 70%. If the temperature is above 30 the motor will rotate at a speed of 95%.

IV. Result And Discussion

A. Regular Lighting and using the app
The lighting can be controlled by both the LDR and the App both, as shown in Fig. 10, the LED is switched on successfully using LDR.

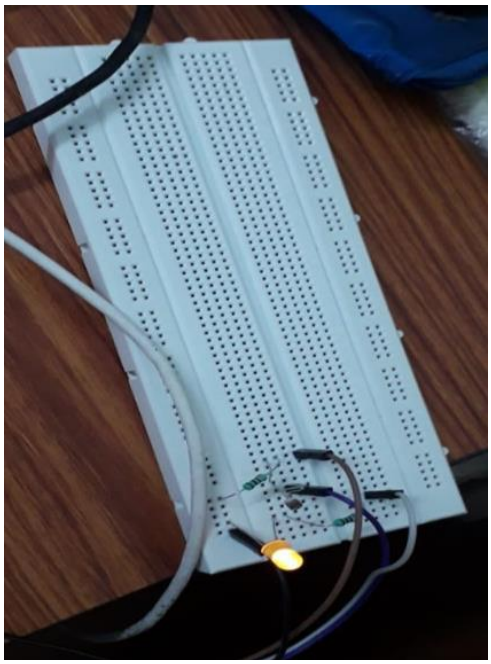


Fig. 10: Turning LED on/off using LDR

Our first milestone to cross was to create a switch button for LED blinking using MQTT Dashboard App as shown in Fig 11, using this the user switch on/off the lights from wherever they want.

Fig. 11: Interface of MQTT dashboard application in mobile

Fig 12 and Fig 13 shows the ideal characteristics accordingly. As the ‘Light On’ switch, is clicked, the LED is turned on and turned off when the switch ’Light Off’ is clicked.

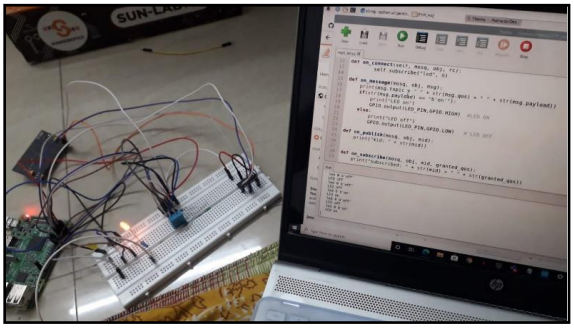


Fig. 12: Turning led on while pressing “Light On” switch in mobile

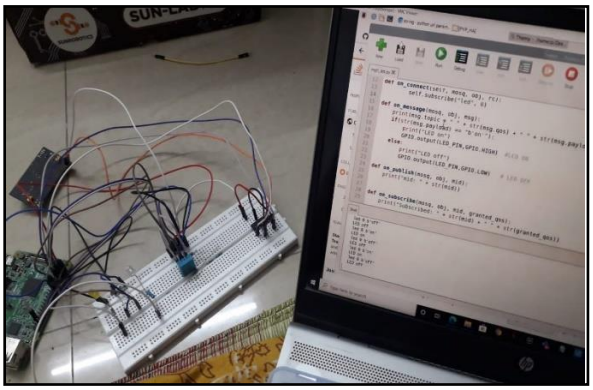
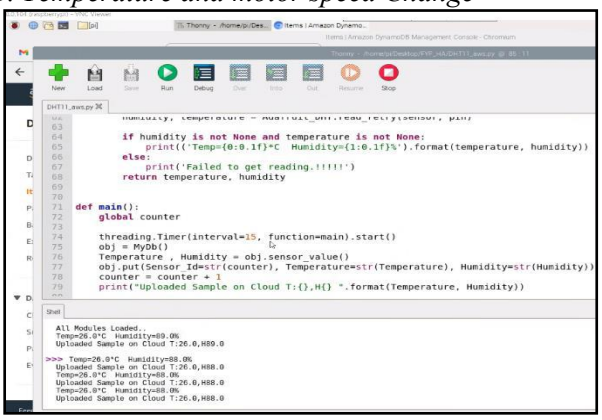


Fig. 13: Turning led on while pressing “Light On” switch in mobile

B. Temperature and motor speed Change



```
        if humidity is not None and temperature is not None:
            print(('Temp={0:0.1f}*C  Humidity={1:0.1f}%').format(temperature, humidity))
        else:
            print('Failed to get reading.!!!!!')
        return temperature, humidity

def main():
    global counter

    threading.Timer(interval=15, function=main).start()
    obj = MyDb()
    Temperature , Humidity = obj.sensor_value()
    obj.put(Sensor_Id=str(counter), Temperature=str(Temperature), Humidity=str(Humidity))
    counter = counter + 1
    print("Uploaded Sample on Cloud T:{},H{} ".format(Temperature, Humidity))
```

```
All Modules Loaded..
Temp=26.0*C  Humidity=89.0%
Uploaded Sample on Cloud T:26.0,H89.0

>>> Temp=26.0*C  Humidity=88.0%
Uploaded Sample on Cloud T:26.0,H88.0
Temp=26.0*C  Humidity=88.0%
Uploaded Sample on Cloud T:26.0,H88.0
Temp=26.0*C  Humidity=88.0%
Uploaded Sample on Cloud T:26.0,H88.0
```

Fig. 14: Displaying temp & humidity in terminal

At first, using the DHT11 sensor, the data is taken. The temperature and humidity are being displayed in the terminal as shown in Fig 14.

With the successful completion of the account created, the next step was to create Tables using DynamoDB. Two tables were created according to our use, i.e., for door sensor and Temperature data, as shown in Fig 15. The table for temperature change holds the value that is being passed down by the sensor, as shown in Fig 16.

According to the data, as the temperature and humidity change in the environment, therefore, the speed of the motor is changing along with it, as shown in Fig 17.

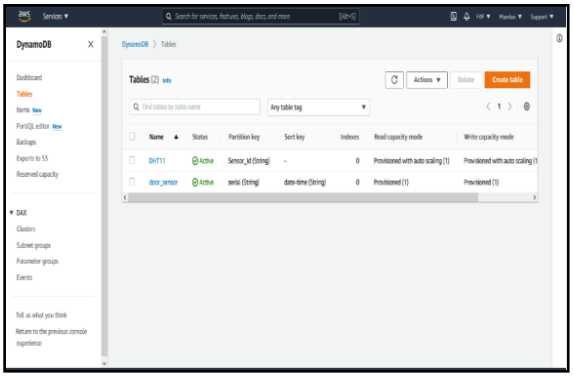
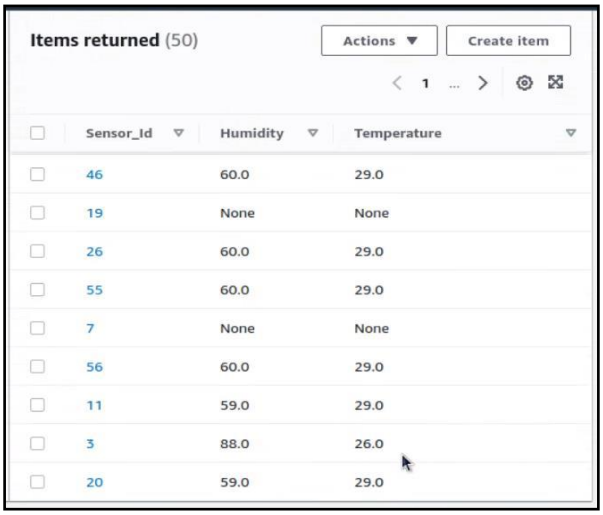


Fig. 15: Tables created in AWS DynamoDB



Items returned (50)

Actions | Create item

Sensor_Id	Humidity	Temperature
46	60.0	29.0
19	None	None
26	60.0	29.0
55	60.0	29.0
7	None	None
56	60.0	29.0
11	59.0	29.0
3	88.0	26.0
20	59.0	29.0

Fig. 16: Uploading temp & humidity in DynamoDB table

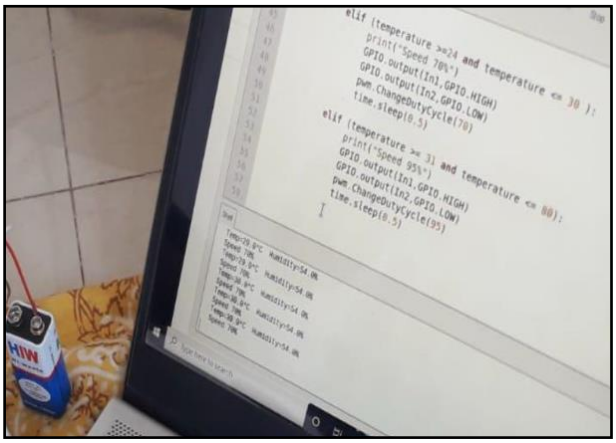


Fig. 17: Motor speed varying depending on temp value

C. Door Sensor

The door sensor is to detects the status of the door and notifies the user, whether it's "open" or "closed". AWS is used to store the data such as the status of the door in DynamoDB and to send a notification using SNS. The SNS is integrated with Raspberry pi through AWS CLI and using the ARN (Amazon Resource Number) and sends a notification i.e., SMS to the registered mobile number whenever there is a change in status of the door, fig 18 shows the table for door sensor.

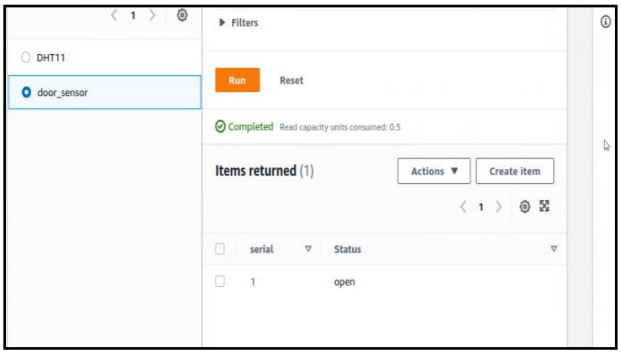


Fig. 18: Door sensor status in a table

The notification that alerts the user every time the door is open/closed, is displayed using SNS as shown in Fig 19, just like that the minute the door was open it stores the data, and when it closes it updates the status through notification

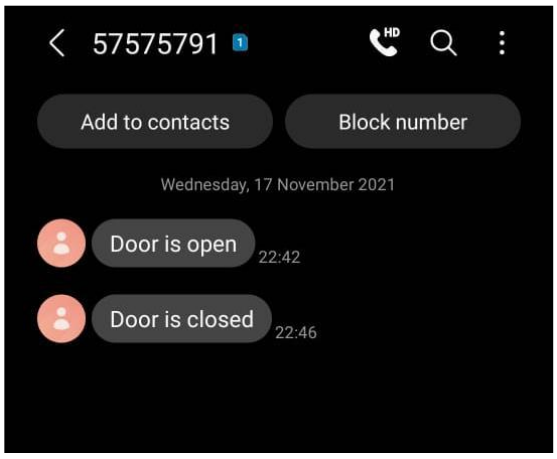


Fig. 19: Receiving notification for AWS SNS

V. Conclusion

This research paper shows the merit of combining Raspberry Pi, MQTT, and AWS app to achieve a better functioning home automation. Successfully controlling the lighting and the speed of the fan depending upon the surroundings and user's preferences. Storing and displaying data (Temperature and humidity), that is picked up by DHT11 sensor, in the Tables using AWS DynamoDB. AWS ensures the protection of the user's data that can be accessed by limited people, that also if the user allows it, using AWS IAM. To provide a worry-free state of mind to the user, the door sensor helps in keeping track and sending notifications to the user whenever someone is trying to break in or if the door is kept open, whenever a user is outside. The main aim of home automation is to provide a worry-free and smart way of living to the user, with all its functions and devices under one touch. We were able to work on this project using the Raspberry Pi 3B+ model and Raspbian OS, which has a faster CPU clock speed, i.e., 1.4GHz and increased Ethernet throughput. The working of all the devices seen above are swift and according to the presence and they run successfully.

In the coming age, with much-advanced technology, home automation will be bringing a massive change in day-to-day lifestyle. As the speed of the fan is adjusted according to the temperature surrounding us, an air purifier can be added along with it. We can use Machine Learning to predict the power consumption happening every day to keep track on it and suggest the calculated pricing. Home automation is a new era, that will be soon found in every household, at a much cheaper cost.

References

[1] Jain College of Engineering, Institute of Electrical and Electronics Engineers. Bangalore Section., and Institute of Electrical and Electronics Engineers, 2020 International Conference for Emerging Technology (INCET) : Belgaum, India. Jun 5-7, 2020. .

[2] N. S. M. Z. Waheb A. Jabbar, Tee Kok Kian, Roshahliza M. Ramli, Siti Nabila Zubir, "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System," IEEE Access, vol. XX, p. 16, 2017, doi: 10.1109/ACCESS.2017.Doi.

[3] M. Ali et al., "An IoT based Approach for Efficient Home Automation with ThingSpeak," 2020. [Online]. Available: www.ijacsa.thesai.org.

[4] C. Stolojescu-Crisan, B. P. Butunoi, and C. Crisan, "An IoT Based Smart Irrigation System," IEEE Consum. Electron. Mag., 2021, doi: 10.1109/MCE.2021.3084123.

[5] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," J. Comput. Commun., vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[6] G. J. Rao, A. Vinod, N. Priyanka, and C. S. Hari Kumar. K, "IOT Based Web Controlled Home Automation Using Raspberry PI," Int. J. Sci. Res. Sci. Eng. Technol., no. March, pp. 229–234, 2019, doi: 10.32628/ijsrset196246.

[7] T. Kim, H. Lee, and Y. Chung, "Advanced universal remote controller for home automation and security," IEEE Trans. Consum. Electron., vol. 56, no. 4, pp. 2537–2542, Nov. 2010, doi: 10.1109/TCE.2010.5681138.

[8] P. Franco, J. M. Martinez, Y. C. Kim, and M. A. Ahmed, “IoT Based Approach for Load Monitoring and Activity Recognition in Smart Homes,” IEEE Access, vol. 9, pp. 45325–45339, 2021, doi: 10.1109/ACCESS.2021.3067029.

[9] G. Song, F. Ding, W. Zhang, and A. Song, “A Wireless Power Outlet System for Smart Homes,” 2008.

[10] S. F. Islam, M. I. Hasan, M. Akter, and M. S. Uddin, “Implementation and Analysis of an IoT-Based Home Automation Framework,” J. Comput. Commun., vol. 09, no. 03, pp. 143–157, 2021, doi: 10.4236/jcc.2021.93011.

[11] O. Taiwo and A. E. Ezugwu, “Internet of Things-Based Intelligent Smart Home Control System,” Secur. Commun. Networks, vol. 2021, 2021, doi: 10.1155/2021/9928254.

[12] K. Venkatesh, S. Hemaswathi, B. Rajalingam, and R. Scholar, “IoT Based Home Automation Using Raspberry Pi,” Special Issue, 2018. [Online]. Available: https://www.researchgate.net/publication/327423472.

[13] T. Liang, B. Zeng, J. Liu, L. Ye, and C. Zou, “An unsupervised user behavior prediction algorithm based on machine learning and neural network for smart home,” IEEE Access, vol. 6, pp. 49237–49247, Sep. 2018, doi: 10.1109/ACCESS.2018.2868984.

[14] Y. Liu, D. Zhang, and H. B. Gooi, “Optimization strategy based on deep reinforcement learning for home energy management,” CSEE J. Power Energy Syst., vol. 6, no. 3, pp. 572–582, Sep. 2020, doi: 10.17775/CSEEJPES.2019.02890.

[15] A. S. Shah, H. Nasir, M. Fayaz, A. Lajis, I. Ullah, and A. Shah, “Dynamic user preference parameters selection and energy consumption optimization for smart homes using deep extreme learning machine and bat algorithm,” IEEE Access, vol. 8, pp. 204744–204762, 2020, doi: 10.1109/ACCESS.2020.3037081.

[16] A. C. Jose, R. Malekian, and N. Ye, “Improving Home Automation Security; Integrating Device Fingerprinting into Smart Home,” IEEE Access, vol. 4, pp. 5776–5787, 2016, doi: 10.1109/ACCESS.2016.2606478.

[17] M. Mehra, V. Sahai, and E. Dsouza, “Home Security System using IOT and AWS Cloud Services,” 2019.

[18] N. Imtiaz Jaya and M. F. Hossain, “A Prototype Air Flow Control System for Home Automation Using MQTT over Websocket in AWS IoT Core,” in Proceedings - 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2018, Feb. 2019, pp. 111–117, doi: 10.1109/CyberC.2018.00032.