Smart Green House

Gorgonzola

13th June 2022

1 Team members

Younsuk Choi
Yigitcan Aydin
Syed Rafsan Ishtiaque
Muhammad Rohail Usman

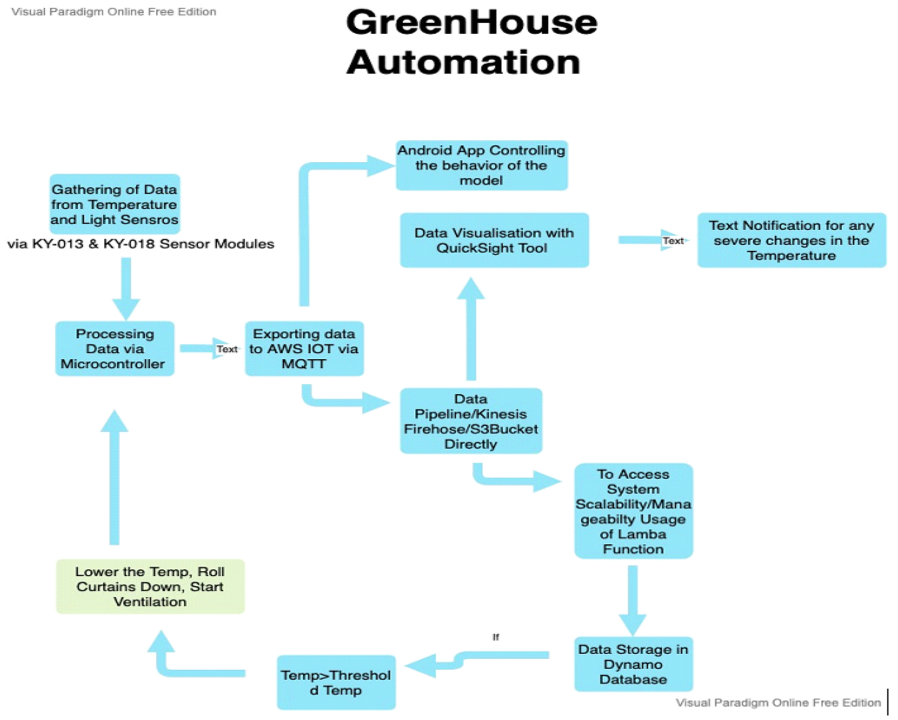
2 Introduction

Like many other smart systems, the management in agriculture can also be enhanced by implementing the distributed systems with IoT and WSN(Wireless Sensor Network) techniques. Particularly, WSN, a network of decentralized sensor nodes which forward collected data to a central location via a wireless connection, enables the users to monitor the environmental conditions of the system and to act upon it remotely.

Our smart agriculture model employs a distributed system in which micro processors and sensors are communicating through WSN techniques. To be more specific, our smart agriculture specifies the environmental constraints of a greenhouse where we can manage our crop by handling data from temperature sensors and light sensors. The central communication units, the Raspberry Pi model in our case, will function as a server where sensors and Arduino Uno work as clients. This will facilitate a bi-directional link between the end-users and the physical space. Then the users can control the system autonomously or dependently by commanding the appropriate execution of the actuators.

3 Concept description

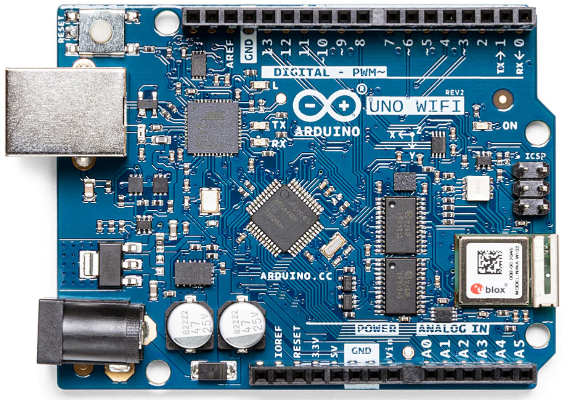
What is the main application for your prototype?

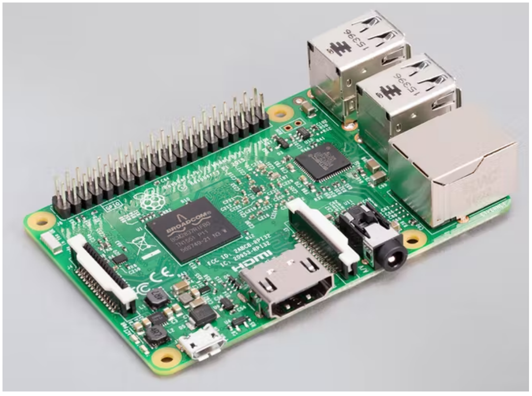
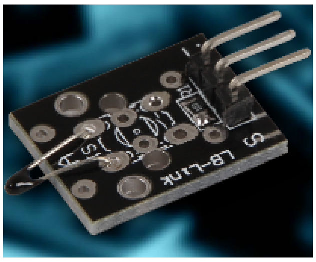
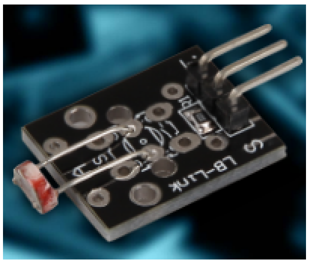


- Smart Agriculture
- Smart Green House

Initiating with the accumulation of the data via temperature and light sensor with the use of Arduino Uno, the data will be exported to AWS IOT via the RapberryPi's MQTT. To increase the interactivity with the system, an interface (android app) will be made to control the behavior of the model. In order to visualize the change in the data. The representation will be demonstrated with the QuickSight tool following on with the text push notification on the mobile device. For ease of the process either a data pipeline is intended to be executed or directly an S3 Bucket can be executed. To ensure the possibility of scalability, the usage of the Lambda function directs the data to be stored in the Dynamo Database.

Which devices, sensors, actuators, apps, etc. are using for your application?



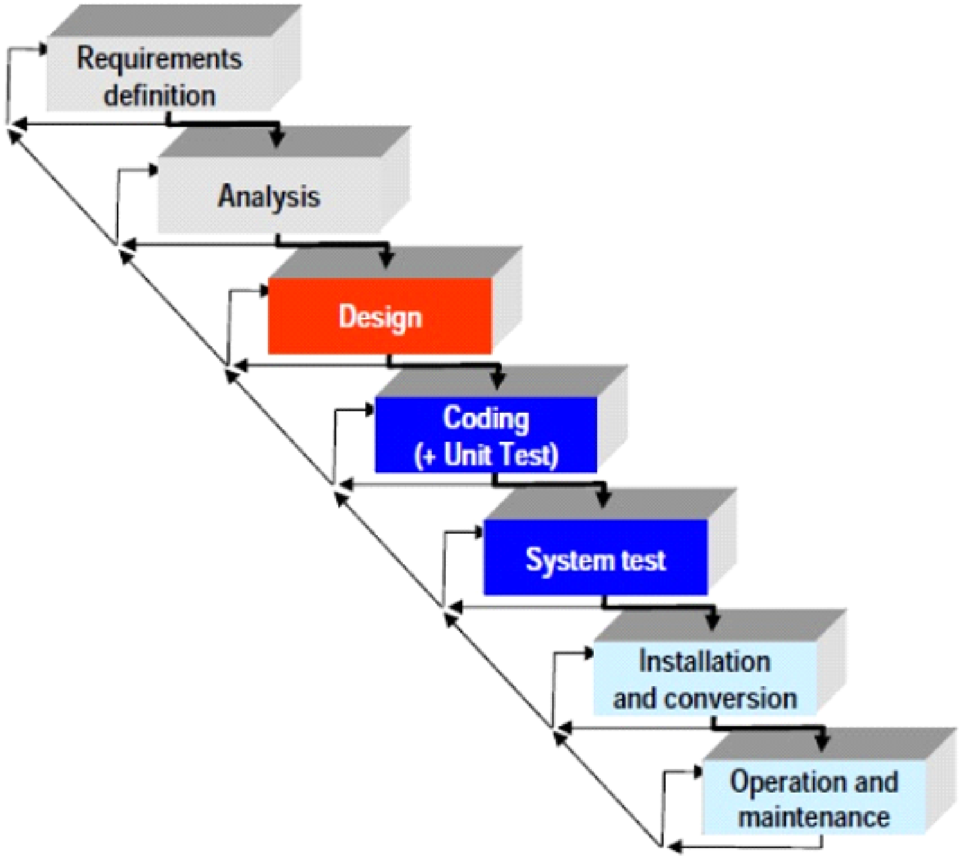




Arduino Uno Wifi Rev 2 [4]
Raspberry Pi 3 [5]
KY-013 Temperature sensor (NTC) [6]
KY-018 Photo-resistor [6]
KY-053 Analog-Digital Converter [6]
RB 35 Motor [7]
LED

4 Project/ Team management

Project methods used in the project:

For the project, we followed an updated version of the waterfall model. At the beginning of the individual lecture, we had our goal fixed. Then after the explanation of the topic from the professor's slide, we started that segment by dividing our tasks for that day.



We started by describing the requirements first. After that, we came up with analyzing those requirements. After defining the requirement we started with the design part. In this design part, we went through the available components and the modules we need for the part of the project. The next step was working with the code. For our project, we used the Arduino environment for coding the implementation of the components.

We also used online references and got our coding ideas from other literature. After we are done with our coding, we tested our system. For our project, the installation was simplified. As the project only requires some sensors and actuators, it was tested several times. The installation was done and verified. The last step we had was to put the whole system in operation. In this stage, we added the maintenance part too. For that, we carefully noted the previous versions of the project and used them as references to modify or maintain the upcoming one.

Managing tasks:

Our group had arranged several meetings throughout each week. We had an on-campus meeting during our laboratory time. Apart from them, we worked on the university campus and also collaborated via online medium. We have our GitHub repository for the project. Members contributed through GitHub. We uploaded our progress there and rechecked for further developments. During the on-campus session, if we faced any challenges, we had discussions with the professor. Meanwhile, the regular correspondence among the team members enabled us to perform the project in due time and in an organized way.

Different tasks/roles of the team members in the project:
For our project, from the very beginning, we were careful about the distribution of the individuals' tasks regarding different aspects of the project. Suppose when we started the different in-project segments, i.e. working on the sensors, one of us started doing the online research regarding the sensor, another started looking into the code of setting up that sensor, and another worked on synthesizing the sensor or establishing the sensor with a client such as Arduino, and the other member started with bridging it with the main system.

In that way, whenever we had a new sensor that we had to integrate into our system, we worked as a team and implemented the extension. This goes the same for the other actuator, client, broker, etc., parts.

Team members' part:

As in the last section we have described thoroughly how we managed our tasks, from that you can have an idea about our task management. We divided the micro-segments among the group members. When we had a micro part of the project, we all tried to contribute to finishing the part. Though the whole project was distributed into small micro-segment, we had responsible group members for individual segments to avoid any backlog or to make the process faster.

Choi: Technologies part in the document, Subsidiary role in MQTT connection (commenting on the codes)
Yigitcan Aydin: Implementation of the MQTT Protocol code, use case diagram
Rafsan: Actuator/ Motor part, Project management part in the document, some commenting in the codes
Rohail: Implementation of the AWS (using TCP/IP), MongoDB part

5 Technologies

Describe the technological approaches you will use to implement your project.

- **Sensor technologies**

  Our Smart Greenhouse needs to be able to respond to any environmental change in the system. To do this, we first need sensors to collect physical data and actuators to make a physical change in the system. Sensors gather physical data and convert them into electrical signals. This change in electrical signals at the output of the sensors will be converted to digital data and detected by a microcontroller to be processed. In our cases, for the sensors, we use the KY-013 Temperature sensor (NTC) and the KY-018 Photoresistor from Sensorkit X40, for actuators, we use RB 35 motor, and for the microcontroller the Arduino Uno Wifi Rev 2 module.

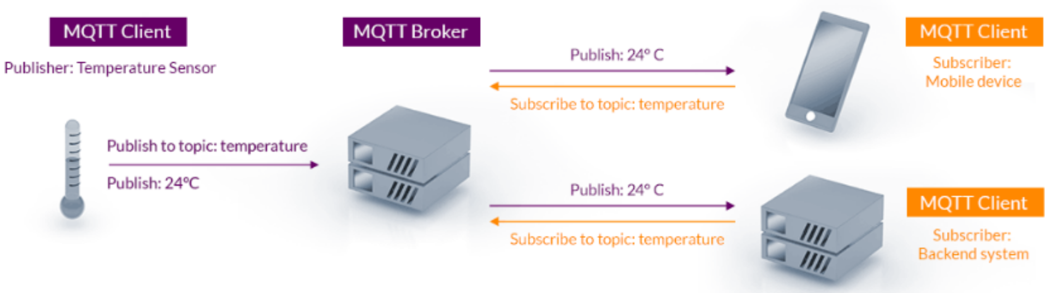
  A KY-013 Temperature sensor has NTC (Negative Temperature Coefficient resistor) thermistor. An NTC thermistor has a resistance that is inversely proportional to the temperature, which indicates that it will have high resistance when the temperature is low. Therefore, KY-013 will be able to read temperatures ranging from -55 C to 125 C, depending on the change in resistance. The resistance can be calculated from the measured voltage from a voltage divider. Also, a KY-053 Analog-Digital Converter has to be used to convert the analog data at the output of the temperature sensor to the digital data that the Arduino can process. [8]

A KY-018 Photoresistor works as a light sensor and has an LDR (Light-Dependent Resistor) resistor in the module. An LDR resistor indicates the presence or absence of light or the light's intensity by exhibiting photoconductive characteristics of metals. In the presence of light, the resistance is relatively low, and this resistance can be obtained from the measured voltage of the voltage divider in the module. [9]

An RB 35 Motor has the following specification. The operating voltage is 12V/DC. The speed under load is 174 rpm whereas the idling speed at 12V is 200 rpm. The maximum gear load is 6kg/cm whereas the steady gear load is 2kg/cm. The transmission ratio is 1:30. This motor will be used as an actuator and it will move the mechanical curtain which can block the light from the greenhouse. [7]

The data accumulated from these sensors will be processed and dealt with by the Arduino Uno Wifi Rev 2 module.

- **Communication protocols**
  - **MQTT**



[10]

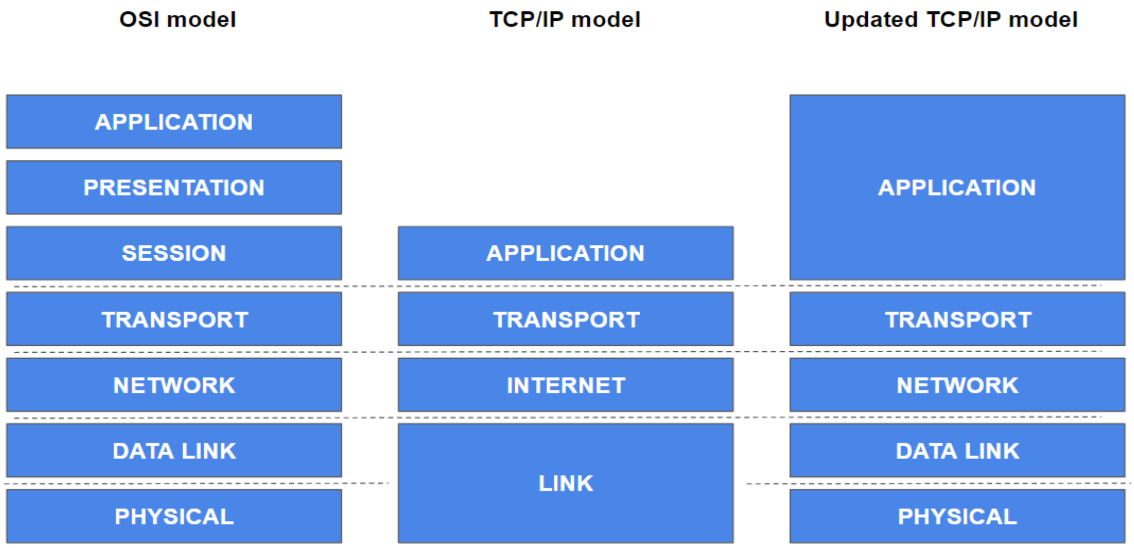
MQTT Protocol stands for Message Queuing Telemetry Transport and it can simply be seen as a messaging protocol. It realizes communication between multiple devices over the internet, especially when the devices are constrained or with low bandwidth. Due to this characteristic, the MQTT protocol often serves as an important tool when building an IoT system as well as Wireless Sensor Network.

MQTT has two different classes called MQTT broker and MQTT client. MQTT broker can be seen as a control hub since it receives all the messages and filters them and publishes them to all subscribed clients. Clients are devices in the system, which can be a user's smartphone, server, or some sensors that read external data. In terms of methodology, MQTT allows bi-directional communication by using publish/subscribe method. This indicates that a device can publish a message on a topic to the MQTT broker, making the device a publisher, or a device, a subscriber, can receive a message by subscribing to a topic on which another device is publishing. In this case, the message can be either a command or data. A topic is what an MQTT broker uses to decide which client receives which message. Topics are represented with strings separated by slashes that indicate the topic level. Topics are case-sensitive. MQTT protocol has some advantages over other messaging protocols. MQTT is lightweight and efficient and it can provide connections between a great number of devices. Also, it guarantees reliable delivery by having 3 levels of quality of service(QoS). QoS 0, QoS 1, and QoS 2 have distinct characteristics that the message will arrive at most once, at least once, and exactly once, respectively. [11]

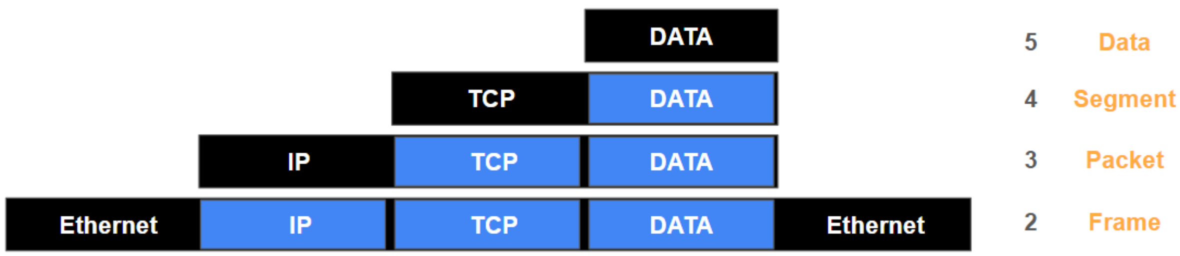
First, Arduino Uno Wifi Rev 2 module will act as an MQTT client that publishes sensor readings onto a topic. Then, the Raspberry Pi module will act as an MQTT broker and manage the message to be published. In this case, a user's smartphone will act as an MQTT client that receives the message by subscribing to the topic onto which Arduino publishes the message. The users can both receive the message and give commands to the actuators connected to the Arduino while publishing to another topic.

- **TCP/IP**

TCP/IP is a model designed to standardize computer networking. Compared to the OSI model, TCP/IP is a practical model used in computer networking. The TCP/IP model has four layers: application, transport, internet, and link layer, and it is numbered from the bottom up. This model is updated to have 5 layers: application, transport, network, data link, and physical layer.



There are distinct protocols and devices used at each layer. At the application layer, we have application protocols such as HTTP, FTP, and SMTP. The two most used transport protocols are TCP and UDP, including port numbers in this layer. At the network layer, we have Internet Protocol (IP) and routers operate at this layer. The data link layer includes Ethernet and switches operate at this layer. At the physical layer, we have cables and a Network Interface Card (NIC)



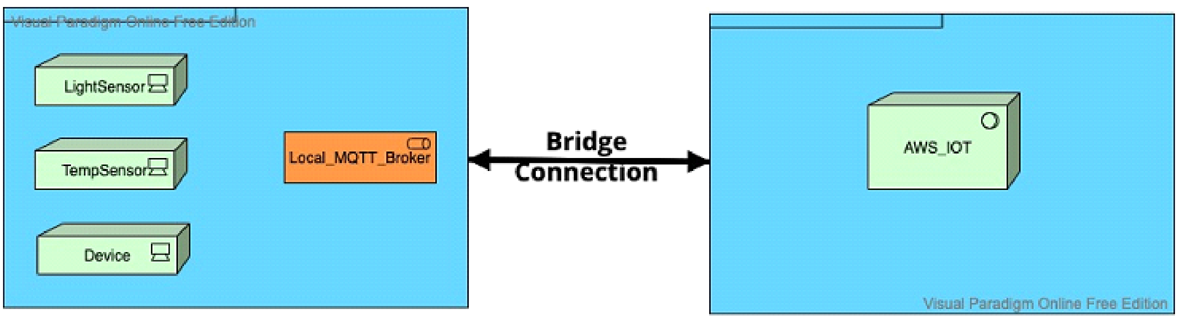
When we send or receive data, we have two procedures called encapsulation and decapsulation. When a device sends data, each layer will add its own information. (Encapsulation) Then, the data will be transmitted through the physical layer to the receiving device. Finally, the receiving device will decapsulate the data over each layer.

When encapsulation procedure, from layer 5 to 4, transport information in TCP header is added. A TCP header typically includes the source and the destination port number. From layer 4 to 3, an IP header will be added where a typical IP header contains the source and destination IP address. From layer 3 to 2, Ethernet header and trailer will be added. Information such as destination and the source MAC address will be in the header whereas error-checking information will be in the trailer. Finally, when it hits the physical layer, it will be physically transmitted. At each stage, the data has different names, as you can see in the figure. [12]

- **Mongo Database on Amazon Web Services using TCP/IP protocol**

In contrast to the major part of our project, we also came up with the idea to connect our Arduino Uno Rev 2 to the AWS-IOT cloud so that we can implement the storing of data in DynamoDB using Mac_Id, random number, and string.
Later on, at almost the verge of implementation, we discovered that a direct connection of Arduino Mqtt broker to AWS-IoT was certainly not possible. This was because AWS-IOT uses TLS1.2 for security and since the FS-files tools do not support the Arduino Uno Rev-2 we switched on to another solution. Either we could have switched to the use of Arduino Board ESP32, or created an MQTT broker on the server and used it as a bridge to send the JSON payloads.

Then, it was decided that the only solution to achieve a connection between Arduino rev-2 to AWS was to create a bridge. It initiated on with the creation of AWS certificates. Then creation of AWS-thing plus policy and attaching it to the certificate was done. Moreover, mosquito broker on the server was manageable to be implemented to some extent but we are now encountering errors while creating connections. Up till now, we were not able to fix it.



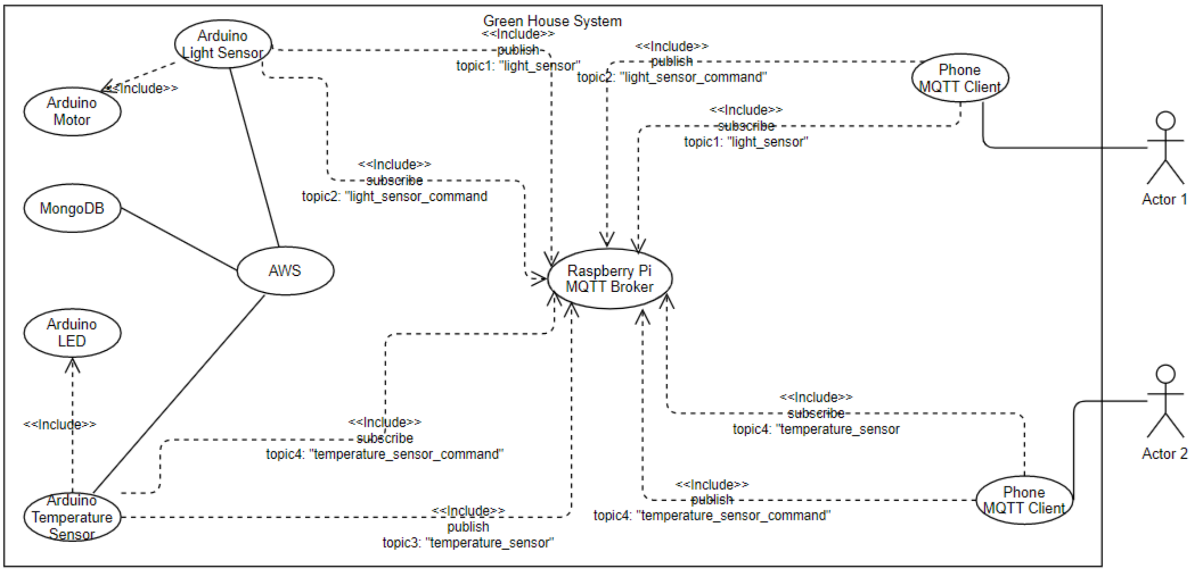
6 Implementation

The Static structure of the environment: A brief explanation of the environment in which our system is present


[13]

We limited our static structure of the environment to a greenhouse that is situated outdoors, possibly in a user's garden, and susceptible to external environmental changes such as rain, the amount of sunlight each day, outdoor temperature, etc. We built our greenhouse with a set of sensors and actuators with which users of the system can monitor the environment and manage the necessary course of actions to control the system. In other words, both temperature sensors and light sensors are situated inside the greenhouse to detect weather changes within the environment. Also, for example, motors will be actuated by a user so that they bring temperature drop within the system by turning on a fan in the system or by opening up mechanical curtains of the warehouse.

7 Use Case



In order to use the Green House application, first of all the user should have a MQTT Client (one of the client applications on the internet) has to be installed, assuming that the MQTT broker on raspberry pi and arduino are already installed and running. After the hardware installation via giving the network credentials and the topic names, the user is ready to monitor the green house system via his/her phone. Along with it, via the AWS (Amazon Web Services) all the sensor data and the user inputs are being stored in the database which is in our case MongoDB.

In short, in the Green House System, users are able to manipulate the lighting and the air conditioning system either using the MQTT protocol via the raspberry pi or TCP/IP via the AWS and MongoDB. This feature gives to the user the freedom in terms of the devices in usage.

8 Sources/References

1. "ARDUINO UNO WiFi REV2"
   https://store.arduino.cc/products/arduino-uno-wifi-rev2

2. "Raspberry Pi 3 Model B"
   https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

3. https://sensorkit.joy-it.net/en/

4. "RB 35 1:30 GEARBOX MOTOR"
   https://www.botnroll.com/en/dc-motor/2799-rb-35-1-30-gearbox-motor.html

5. "Arduino KY-013 Temperature sensor module"
   http://arduino-sensor.org/sensor-kits/arduino-ky-013-temperature-sensor-module/

6. “Light Dependent Resistor LDR: Photoresistor”
https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php

7. “MQTT Publish / Subscribe Architecture”
https://mqtt.org/

8. “MQTT Essentials_The Ultimate Guide to MQTT for Beginners and Experts”
https://www.hivemq.com/mqtt-essentials/?utm_source=adwords&utm_campaign=&utm_term=mqtt%20topics&utm_medium=ppc&hsa_tgt=kwd-1634535684706&hsa_cam=17295502954&hsa_src=g&hsa_net=adwords&hsa_kw=mqtt%20topics&hsa_ad=594094363642&hsa_grp=138486496202&hsa_ver=3&hsa_acc=3585854406&hsa_mt=e&gclid=Cj0KCQjw2MWVBhCQARIsAIjbwoNk9fdiVSJPpfuMVpzs86dfmNhAA-tqABG-yJLLMQ7sSXXYtjqxJtYaAvKCEALw_wcB

9. “What is Transmission Control Protocol TCP/IP?”
https://www.fortinet.com/resources/cyberglossary/tcp-ip

10. “How to choose the best greenhouse”
https://www.gardenersworld.com/plants/choosing-a-greenhouse/#:~:text=There%20are%20three%20main%20greenhouse,which%20is%20ideal%20for%20growing.

11. “Why Do Plants Grow Better In A Greenhouse?” Hartley Magazine, 2019,
https://hartley-botanic.com/magazine/plants-grow-better-greenhouse/