ISWE30011-IoT Programming
Group project Smart home system

Dilni De Silva

103616345

Joshua Mota

103020360

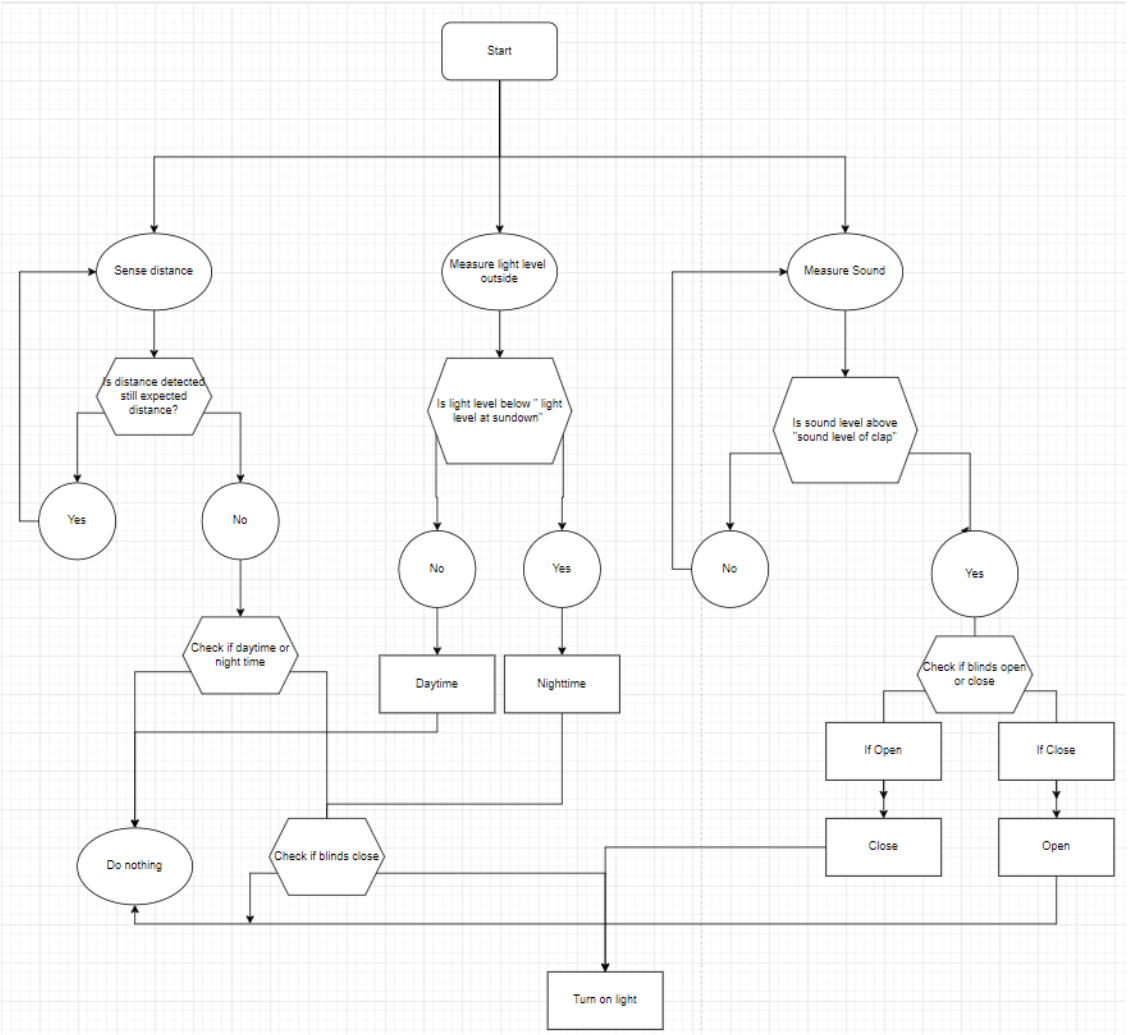
Table contents

Introduction

Advancements in IoT smart devices, in the 21st century, have led to many improvements in the facilitation of a higher standard of living. This is especially evident in the creation of the smart home system. [1] Which are networks of home appliances and devices which can be controlled or monitored through the homeowner's smartphone or any other API connected to the smart home system. Therefore, enabling users to monitor or control security, temperature, and lighting. Moreover, for the IoT group assignment, our group has proposed an automated lighting smart system where it allows homeowners to control natural and artificial lighting. This is done by utilizing two sensors that communicate with each other using an MQTT communication protocol, which is first published to the database and webserver, and then received by subscribing to the related topic. The database and web server are used to store and save the data collected from both sensors. This ultimately allows both sensors to access the data using MQTT communication protocols from the webserver.

Conceptual design

Conceptual diagram

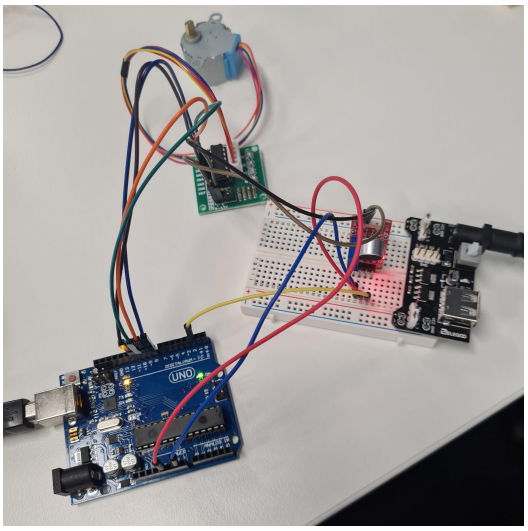


Conceptual Design description

This system has two components. One of the sound-activated blinds components monitors if a sound was produced at a certain volume that causes the system to check if the blinds are closed or not. If they are open they get closed, then if closed they get opened. Then the state of the blinds gets published using a communication protocol to a database in a webserver. This is where the seconds component of the motion-activated light system of system subscribes to this web server and accesses data if the blinds are closed or not. This second component first checks if there is a present object/person in the room using an ultrasonic sensor module. Additionally checks the time of day using a light-dependent resistor. Then using a subscribe MQTT protocol, it gathers the data from the database from the webserver checking if the blinds are closed. Hence turning the lights on / off depending on the received data. These have been demonstrated above in the flow chart as multiple decision blocks and loops.

Implementation

Sound activated blinds system



The system below shows the various components used to sense sound and wind the blinds up and down. Connected to the breadboard are the power module and the sound sensor. The power module is used to power the motor seen at the top of the photo, due to the Arduino not being able to provide enough amps to power it without fear of damage.

Motor

The motor is responsible for turning the blinds. It has a gear ratio of 64:1 and is powered by a ULN2003 Driver Board, which supplies 5V and 240mA to the motor. This motor is activated when it receives a signal from the sound sensor and turns a full revolution.

```
void setup() {
  Serial.begin(9600);
  // set the maximum speed, acceleration factor,
  // initial speed and the target position
  myStepper.setMaxSpeed(1000.0);
  myStepper.setAcceleration(50.0);
  myStepper.setSpeed(200);
//  myStepper.moveTo(2038);
  pinMode(soundSensor,INPUT);
}
```

As seen in this code above, the motor starts up and accelerates towards a max speed, before slowing down to a stop, simulating a real blind. When the blinds are activated, it sends a signal to the other edge server, indicating whether or not it is up or down. This is done because it is a condition needed if the user wants to automatically turn on their lights.

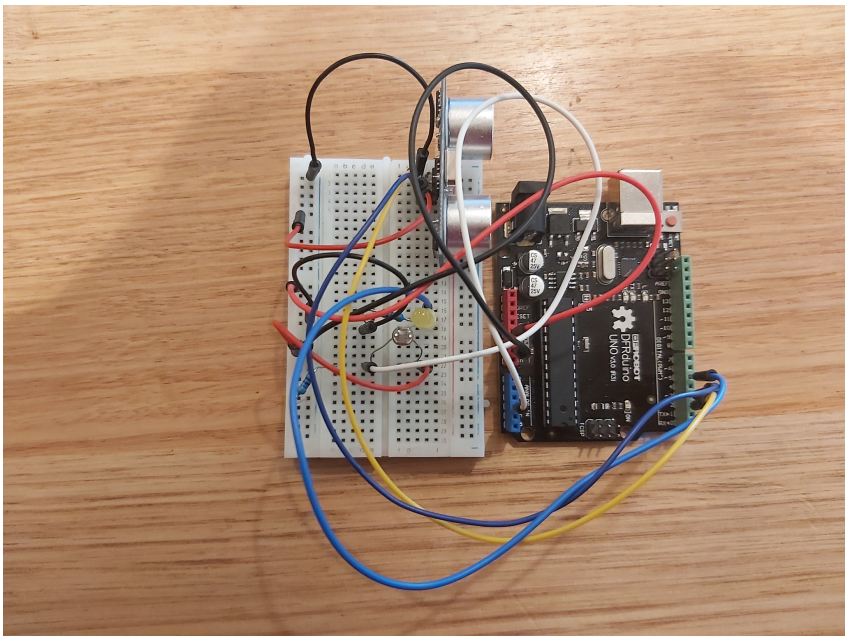
Sound sensor

The sound detection sensor module is responsible for receiving sound input and sending it to the motor. When the sensor senses a sound above a certain threshold, the sensor is

briefly set to high and activates the motor causing it to spin. Ideally, this sensor would be placed in a central location, allowing the user to clap from anywhere within a room, which closes the blinds.

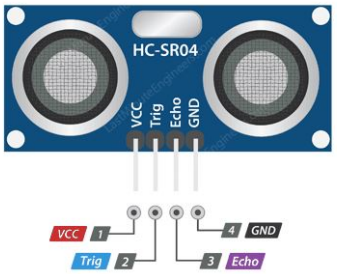
Motion-activated light system

The circuit below is the Motion-activated sensor light system, which has two sensors that give two conditions to be filled which turn on the lights. Such as one checking for the presence of a person who entered the room by comparing the distance of the room to the distance read by the sensor. Secondly, the circuit also checks the time of the day or light level of the outside and if it is nighttime or dark outside as that is when it's most appropriate to turn on the lights. In addition to checking if these sensor values fulfil the correct condition it also communicates through the edge server using an MQTT communication protocol to check if the sound-activated blinds have been closed. Then and only then will the lights turn on when a person enters the room. However, there are exceptions to this using a website to override these automated conditions stated in the code, giving a manual way for users to control the lights as well.



Sensors:

Distance sensor:



One of the sensors that have been implemented for this project is a distance sensor whose purpose is to detect if someone is in the room by checking if the distance of the room is less than the expected distance. This detection is what is tested against one of the conditions that if fulfilled turn on the lights.

The sensor functions by [2] emitting ultrasound waves at a speed of 343m/s or 0.034cm/us which are produced by the echo pin and thus are triggered by the Trig pin when an object is in front. When the sound wave reaches the object it bounces off the object and reflects back. Thus the time in seconds taken for those sound waves to be reflected back and be again detected by the sensor is what is used to detect the distance between an object and the sensor. However, the sensor itself only detects the time in seconds not the distance, hence further calculation has to occur to generate that distance reading. Therefore the distance formula is used to determine the distance between the sensor and the object, divided by 2 as both the sound emitted and reflected times are detected and the reflected sound waves time is what is needed. This formula was implemented in the code of this component seen here, to give the distance between the sensor and the person who has entered the room.

```
/////////////// Distance calculation///////////////
// Setting Time to the echo pin
long Time = pulseIn(echopin,HIGH);
int Distance = Time * 0.034/ 2;
```

Light dependant resistor sensor

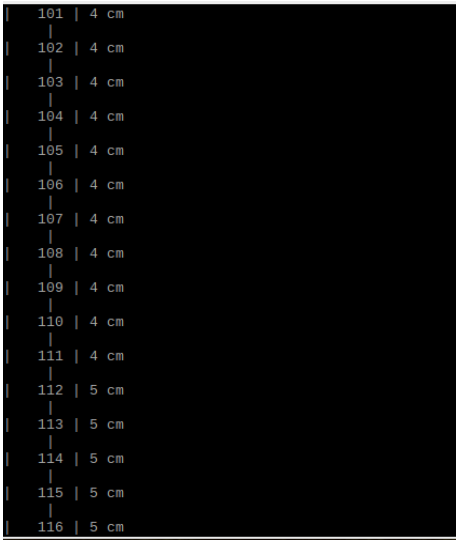
Another condition for the motion-activated light system to fulfil is to check whether it is day or night, as it would be more appropriate to turn on the lights when it is night. Hence an LDR is used to detect this.

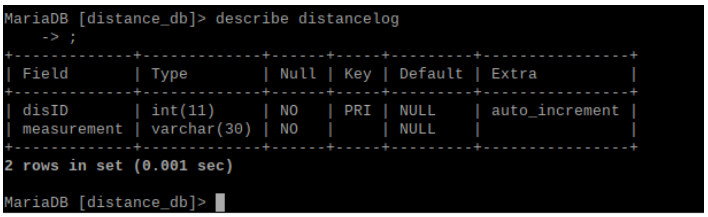
```
if(Distance < expected_distance && LDRvalue < 100)
{
    digitalWrite(Ledpin,HIGH);

   // Serial.print(" on ");

    Serial.print(Distance);
    Serial.println(" cm ");
    Serial.print(LDRvalue);

}
```

This sensor works by providing an analog voltage which when a high-level light hits the LDR it gives low resistance hence a high voltage, then a low light level gives low voltage and high resistance. Therefore it is implemented in the code if the condition of the analog voltage is less than 100 meaning that if it is night time the lights will turn on.

Edge server

The edge server used in this component is a raspberry pi which connects to the Arduino board code enabling multiple functionalities. Such as hosting a webserver to host the website that enables users to manually turn on and off the lights. Enabling MQTT communication between the sound automated blinds and the motion-activated light system will be discussed further in another section. In addition to this, the edge server also contains a database of all the distances recorded by the distance sensor. The purpose of this database is to have a record of the presence of someone in the room. Hence in a way acting as a small security system that monitors when someone entered the room or not.



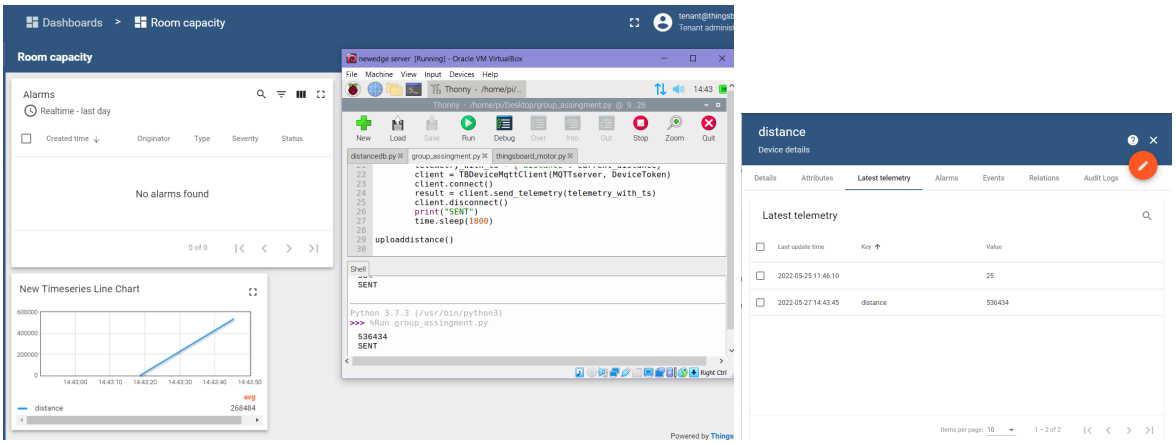


Communication protocols

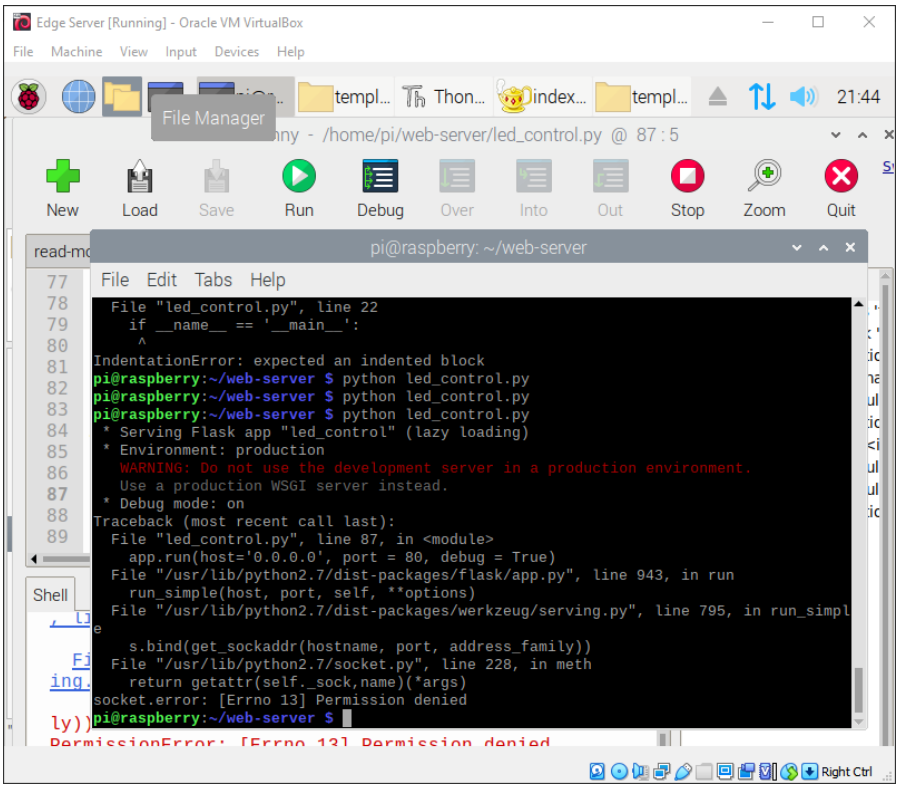
The communication protocol for the motion automated light system subscribes to the database of the sound automated blinds system. Collecting the data that if the blinds are closed or not is used in a condition in the motion automated light system to test if it is okay to turn on the lights if the blinds are closed, as appropriately lights are turned on when the blinds are closed.

API and website

The following images depict the API of things board and the website output of this portion of the project. The API shows the transfer of data using an MQTT protocol from the edge server to the things board depicted in the dashboard here. Show the most recent value transferred. In addition, the website here was intended to give the user the option of having a manual way of turning on the lights if they wanted to have the lights on either during the day or if the blinds are open.Html was used to construct the website and python was used to constant the back end and flask of the website, using a web-server to host the website.

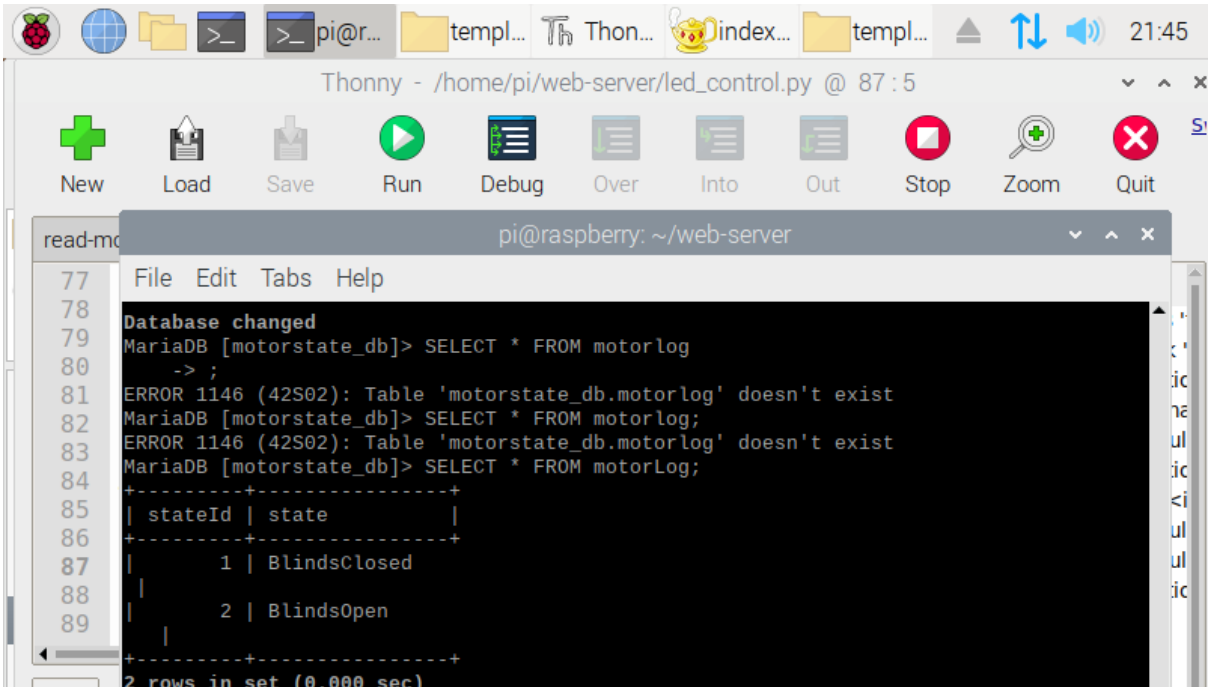


Python code sending data to things board and things board receiving the data



Cloud computing

Cloud computing was used here in many ways, such as the databases to store the data which can be accessed using an MQTT communication protocol and is sent to things board through an MQTT communication protocol where data can be displayed as an API.



Displayed above is the database for this IOT project. The only data displayed in this database is whether the blinds are open or closed. For the other edge server to use this information, it would have to take the last entry through an sql statement and read it back to the arduino.

User Manual

Sound sensor manual

Upon installing the blinds controlled by a motor, they are open by default. To close the blinds, all you need to do is clap or make a loud enough noise which activates the sound sensor. The sound sensor will then start the blinds and either open or close them depending on their previous setting. The blinds can also be controlled through buttons found on the website's front page. If the button pressed matches the blind's current setting, the buttons will do nothing and keep their current state, while if the opposite state is pressed, the blinds will open or close to match the button pressed.

Distance sensor user manual

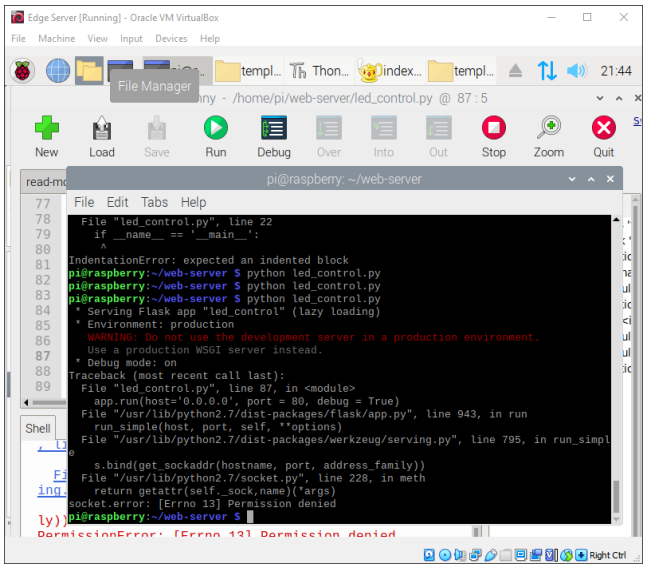
The distance sensor combines with the light sensor to provide automatic lighting in rooms or places where the distance sensor was installed. It works as follows: The user will enter the room where this system has been implemented. The system will first detect the presence of the user, secondly using an LDR will monitor whether or not it is day or night like that as generally, that's when normally user will turn on lights. Lastly, the blinds will also have to be closed by the user clapping their hands giving which causes the blinds to close this is generally how the system works.
Therefore the following conditions must adhere to

- Must have a low light level or be night time
- have closed blinds
- A person must be in the room

However, there is currently no way to change these settings, the lights can once again be turned on and off manually through buttons found on the web site's front page which will be provided to the user.

Limitations

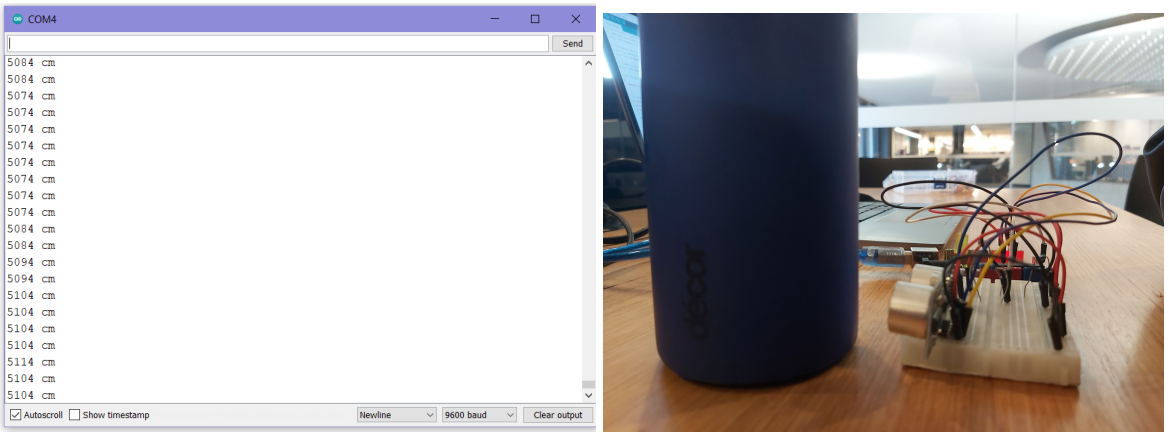
Unfortunately, the website does not work due to a server error seen below.
Hours of troubleshooting were dedicated to somehow get it to work, however, in the end the website would not establish a connection with the edge server.



Voice commands would've been a more ideal and better solution to manually control the blinds or lights, which could've also be allowed for other more diverse operations. While the current solutions work well, having voice command capability would allow the architecture to evolve and possess greater functionality.

A motion sensor was first bought, thinking it would allow the lights to turn on if it detected movement. However, a problem arose, being that if the user stopped moving, the lights would turn off. We then came to the solution that a distance sensor could instead be used, however, this sensor doesn't cover a wide range, and can only be used when the user is directly below or in front of the sensor. In an ideal solution, enough distance sensors would be installed to cover the whole room, so that any irregularities in measurements of the room would be sensed as a user being inside the room, and therefore the lights would turn on, however, the solution only accommodates for only 1sqm rooms using the 1 sensor if installed in the ceiling.

During the transport of the distance sensor, there was damage caused to the distance sensor leading to firstly large random readings being printed to the serial monitor as seen below. Previously this sensor did work and give an output when certain conditions were fulfilled. However, after the damage was done during transportation the sensor stopped working. This led to the circuit not operating as the readings of the distance sensor did not give a reading less than 100 cm causing the LED not to turn on as the condition of the code is not fulfilled.



Images show that even when an object is very close to the system it gives a faulty value.

Resources

1. Hayes, A., STAPLETON, C. and KVILHAUG, S., 2022. *Smart Home*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/s/smart-home.asp> [Accessed 19 May 2022].

2. meenakshy, m., 2022. *ULTRASONIC SENSOR HC-SR04 WITH ARDUINO*. [online] Elementzonline.com. Available at: <https://www.elementzonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino#:~:text=Trig%20(Trigger)%20pin%20is%20used,transmitted%20signal%20to%20be%20detected.> [Accessed 23 May 2022].

3. Sanjeev, A., 2018. *A simple project using an Arduino that automatically turn lights on when an LDR sensor detects darkness.*. [online] Maker pro. Available at: <https://maker.pro/arduino/tutorial/how-to-use-an-ldr-sensor-with-arduino#:~:text=The%20LDR%20is%20a%20special,DIY%20Arduino%20LDR%20sensor%20project.> [Accessed 19 May 2022].

4. Last Minute Engineers. 2022. *In-Depth: Control 28BYJ-48 Stepper Motor with ULN2003 Driver & Arduino*. [online] Available at: <https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/> [Accessed 27 May 2022].

5. Grizhnevich, A., 2022. *IoT for Smart Cities: Use Cases and Implementation Strategies*. [online] Scnsoft.com. Available at: <https://www.scnsoft.com/blog/iot-for-smart-city-use-cases-approaches-outcomes> [Accessed 25 April 2022].

6. Digitalocean.com. 2022. *How To Store and Retrieve Data in MariaDB Using Python on Ubuntu 18.04 | DigitalOcean*. [online] Available at: <https://www.digitalocean.com/community/tutorials/how-to-store-and-retrieve-data-in-mariadb-using-python-on-ubuntu-18-04#step-4-retrieving-data> [Accessed 27 May 2022].

Task break down

Stage/task	Member
Creating Flow chart draft	Josh
Conceptual description - Explain the ideal system in a single paragraph	Dilni
Task break down	Both
Introduction of project report	Dilni
Implementation of automated blinds section - Building of circuit - Coding Arduino code - Connection to single database using individual raspberry pi - Connection of database to single cloud server - Connection of cloud server to single web-server (flask) - Write up of implementation	Josh
Implementation of automated light on/off section - Building of circuit - Coding Arduino code - Connection to single database using individual raspberry pi - Connection of database to single cloud server - Connection of cloud server to single web-server (flask) - Write up of implementation	Dilni
User manual automated blinds section	Josh
User manual automated light on/off section	Dilni
Limitation	Both