ASHESI UNIVERSITY

ROS-E: A SMART BUILDING MONITORING SYSTEM

APPLIED PROJECT

B.Sc. Computer Science

YAW DARFOUR BOTWE

2020

ASHESI UNIVERSITY

ROS-E: A SMART BUILDING MONITORING SYSTEM

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

YAW DARFOUR BOTWE

2020

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

Acknowledgements

To my supervisor, Dr. Nathan Amanquah and all the other wonderful lecturers that have instilled knowledge in me to make me come this far. Thank You.

Abstract

Through technology, buildings are no longer just containers for life. With the large amount of data that can be accessed about everything from occupancy, to ambient room conditions and energy use, buildings are becoming smarter. Smart buildings offer new solutions to the problems of smart living. Unfortunately, most buildings in Africa fail to tap into the data that is provided by the ambient conditions inside them as well as the occupants residing in them. This project provides a smart building monitoring system that keeps track of the ambient conditions inside a building by the use of an Arduino powered module connected to sensors that transfers data via Bluetooth and makes further analyses on this data to improve comfortability and usability of occupants.

Table of Contents

Chapter 1: Introduction

1.1 Background

Throughout time, the way in which buildings were constructed has changed drastically. Early buildings were made out of more simple materials such as clay, stones, wood and tree branches but through industrialization and with the introduction of new and more durable materials like iron, cement and concrete, buildings became taller and stronger. The development of buildings did not stop there but grew with the introduction of technology. Buildings now offer their occupants the option of controlling heating, humidity, ventilation, air conditioning, lightning and security systems. The next step for buildings is to make them more data driven and autonomous.

The current state of buildings fails to take account of the data provided by its occupants and surroundings. This data could be monitored, accumulated and analyzed to make informed decisions for occupants and resident managers. The data collected would help facilitate communication between humans and buildings to promote comfort in buildings. Most attempts to make buildings more autonomous and data driven are expensive and difficult to install and deploy. Also, these solutions cannot be easily implemented on existing buildings and require them to be renovated and rewired.

1.2 Aim

This project aims to tackle the issue of lack of data driven and autonomous buildings and introduce the next generation of buildings. This project would focus on making an inexpensive intelligent building monitoring system using Internet of Things technology to accumulate and analyse the data of the surroundings of the building as well as its occupants.

1.3 Problem Statement

A large amount of data is produced by the ambient conditions inside a building as well as by the occupants residing in the building and the environmental conditions surrounding it. Buildings of today fail to tap into this pool of data and make valuable insights to promote comfortability, usability, and security of occupants and resident managers. Although modern buildings may have sensors built into them, they often solve only a single purpose such as lightning control or conference room booking. The resident managers or users of the building are still required to have some physical interaction with the system to actually function i.e. they are not autonomous. The use of an intelligent building system to monitor, accumulate and analyze data from the conditions inside and outside the building as well as its occupants would be an effective solution. However, these systems are expensive to install and deploy and cannot be easily integrated into existing buildings.

1.4 Proposed Solution

The project proposes the implementation of an intelligent building monitoring system that can be easily integrated into existing buildings to make these buildings smart. This system would integrate all the core systems of the building such as fire alarms, power, lightning, heating, ventilation and air conditioning to be linked with each other. This would give these systems the ability to talk to one another. The system would consist of a number of sensors to detect and collect data ambient conditions in the building like light intensity, motion, temperature, humidity and air quality. This data would be then stored into a database and analyzed to make informed decisions. The information analyzed by the system would be done in real time and the system would make automated adjustments to the conditions in the building where the need arises. The system would also have a web platform to enable occupants to remotely turn on and off appliances and provide graphical status on the data gathered and analyzed by the system. The underlying constraint of the system is to use minimal amount of wiring in the implementation.

1.5 Project Objectives

- Provide a relatively cheap intelligent building management system.
- Provide a platform that allows building occupants to control appliances as well as view information generated by the system from data collected.
- Provide a system that can easily be integrated into existing buildings without renovation and minimal wiring is used.
- A programmable system that can measure the ambient conditions of the building and analyze them to make informed decisions.

1.6 Benefits of Proposed Solution

One major benefit of the proposed system is that it makes occupants more productive. The improvement in air quality, comfort, security, air conditioning and space availability would lead to the satisfaction of occupants in the building which would be translated to productivity especially in workplaces. Also, the system’s ability to detect whether a room is occupied or not can provide building managers with key insights on how to better use their resources by revealing which rooms are overused or underused. The health and safety of the building occupants with this intelligent system is also improved through the air quality monitoring feature existent in the system as well as the smoke and hazardous gas detection feature can help prevent fires. Another important benefit of this system is the efficient use of energy leading to the reduction in electricity costs. The system would use sensor data to predict the need for necessary adjustments to systems such as cooling and heating systems. The system would also turn off lights when a room is not occupied to conserve energy. This would go a long way to reduce the electricity billing of the smart building occupant. Lastly, the system would be relatively cheap to install and deploy into existing buildings making it easy on the pockets of building occupants and resident managers and also increases the value of the building by making it smart.

1.7 Overview of Chapters

This paper consists of six chapters. Chapter 1 introduces the reader to the project proposed and gives some background to the problem being tackled. It also lists some benefits of the proposed project as well as the objectives of the project. Chapter 2 defines the scope of the project and presents the requirements that are needed to be done to bring the project to life. Chapter 3 talks about the system architecture both hardware and software. It gives insight into how the design of the system would be and the different components required to implement the solution. Chapter 4 focuses on the implementation of the project. The technologies used both hardware, software and programming languages and how they are integrated with each other are discussed in this chapter. Chapter 5 talks about testing the system and the results gained from testing. The last chapter summarizes the project and provides the user with some insight on what has been achieved. It also includes limitations, recommendations and future work of the proposed system.

Plan for requirement analysis

Requirement analysis for this project would consist of requirements derived from literature review and related works as well as the requirements achieved from stakeholders of the system, in this case building occupants and resident managers. This would be used to develop both functional and non-functional requirements for the system.

1.8 Related Work

The increase in the interest in the field of Internet of Things has provided some research in building monitoring and automation as well as the wireless sensor networks used in making these systems. This section talks about the work done by other researchers ranging from designing cloud-based systems for smart building energy management to implementing IoT based ambient monitoring systems.

Vcelak et. al [1], focused on smart monitoring for both the building structure as well as the indoor environment of the building. Three main factors that the system focused on was moisture monitoring of wooden constructions, smart glue laminated timber (GLT) with integrated Fiber Bragg Grating (FBG) monitoring and monitoring of indoor air quality using IoT enabled sensor platforms. The last factor was used in reviewing this paper. The team developed an all-in-one IoT enabled sensor platform that can be used to monitor indoor air quality and communicate its results to IoT networks using modules like LoRa, Sigfox or IQRF. It was discovered that the CO_2 levels measured in a high school in Czech Republic where the system was installed reported dangerous limits of 3000ppm. This shows the importance of a system like this that warns occupants of the quality of air in their indoor environment.

Saad et. al [2], explored the development of an indoor air quality monitoring system with the use of wireless sensor networks and a web interface. The system was made up of three parts, namely sensor modules, a base station and web-based interface. The ambient conditions that were measured were temperature, humidity, dust and the level of gas concentration. The system showed that the testbed which was a research lab had its indoor air quality within the acceptable limit and standards for the health and safety of occupants in the building. As with the system being built for this project, the design and architecture of this system has a web-based interface attached to the sensor modules. The paper however fails to address the way in which data would be sent from the sensor modules to the base station.

Lee et. al [3], tackled a similar problem to that of Saad et. al but this project was more focused on the implementation of an intelligent HVAC system. The system is made up of a sensor box which consists of dust, temperature, humidity and gas sensors that are connected to an array of relays used to control heating, ventilation and air conditioning devices. The system uses Bluetooth low energy signals from the user's smartphone to detect the presence of a user in the room and make informed decisions based on that. The paper however does not establish a relationship between multiple sensor boxes where these devices talk to each other.

Dutta and Roy [4], propose a prototype of a smart building with IoT-Fog-Cloud based architecture. This project focuses on all rooms of a building and has separate functionalities for each room. The system composes of different number of sensors for different rooms that are connected to a fog and then cloud. This project is closely related to Ros-E because it uses a similar array of sensors that Ros-E uses however the system fails to achieve the goal of smart devices being able to talk to each other.

These projects are informative in most aspects concerning the implementation of the Ros-E system from matters including selection of sensors, development of sensor modules and design of system architecture.

Chapter 2: Requirements Specification

This chapter comprises of a description of the project as well as the project scope. The various functionalities of the system would also be defined both functional and non-functional user requirements. The target market and users that the system is made for would also be highlighted. This intelligent automated building system would be called Ros-E.

2.1 User Identification and Use Cases

This automated building system is meant to be used by building occupants and resident managers who are in charge of maintaining the buildings. The following are some use cases where the system would be used:

1. A resident manager who is always under constant pressure from his supervisor to provide relevant information about the state of the building. The manager opens the Ros-E web application to get easy to understand information about the building in a graphical format.
2. The building occupant who constantly has to stress over constantly changing the temperature and humidity of his room would no longer face this problem. The occupant who has the Ros-E system installed in his building would have real-time changes to the ambient conditions in the room as well as conscious monitoring of the quality of air in their surroundings.
3. A building occupant who forgets to turn off the lights and other appliances in his building can correct this mistake by logging unto the Ros-E web application to turn off the appliances remotely.

4. Building senior managers can quickly deploy new monitoring systems without any significant wiring or installation delays.

2.2 Project Description and Scope

The Ros-E automated building system would have four main components that makes the system come together;

- An IoT smart object made up of a microcontroller and numerous sensors that generate a large amount of data from the ambient conditions in rooms.
- A PHP server side and MySQL database to store all the data generated from the sensors to be later analyzed and also feed the web application with information.
- A web application that allows the user to interface with the system and provides a graphical information.

2.3 Procedure for Requirement Gathering

Requirements for this project were derived from the requirement documents of similar projects in home automation and smart building systems. Also, a good number of requirements were retrieved from building occupants and resident managers that were interviewed. These interviews were conducted to obtain insights on how the system would perform and what stakeholders would expect from the system. The requirements of the home automation system developed in [5] are applicable to the Ros-E system because the interest of these home automation systems is to promote autonomous living, energy efficiency, savings and an overall better quality of life for its occupants. Requirements received from the input of the stakeholders of the system i.e. building occupants and resident managers provide a realistic framework with which success can be measured.

2.4 User Classes and Characteristics

The system would have two main users being the building occupants and resident managers. These users would have different level of privileges and information provided to them by the system as well as variety of ways to interact with the system.

- Building occupants – These are the people who reside in the building and are mostly directly affected by the system. They would use the system for adjusting temperatures, humidity, and quality of air. They would have an individual view of the system that only includes the information generated by Ros-E. They would require a log in to the system with a password.
- Resident managers – These are the people in charge of overseeing the general status of the entire building and are in charge of the day to day activities that help maintain the building. They would use the system to aid in the performance of their work by viewing which rooms are underutilized and also viewing which rooms are available to be used. They would have a more wholistic administrative view of the system that includes information from all the systems installed in the building. They also would require a log in to the system with a password.

2.5 System Requirements

System requirements gathered from the requirement documents of similar projects as well as interviews with stakeholders helped to construct the functional requirements necessary for the system to perform for smooth usage of the system.

2.5.1 FUNCTIONAL REQUIREMENT

The functionalities of the system are in two main categories. Main functionalities which include the basic operations that the system would take and extra functionalities which are additional features that aid in the improvement of the normal functioning of the system. From gathering of requirements through interviews and related works the following is a list of requirements for the system.

I. Monitor rooms ambient conditions.

a. Description and Priority

This feature allows the user to get readings on the various ambient conditions in the room i.e. temperature, light, humidity and the quality of air. This would be done through the use of multiple sensors attached to the microcontroller of the system. This is of high priority to the system because the data gathered from this would be further analysed and used to generate informed decisions.

b. Stimulus/Response Sequences

The user should be able to access the Ros-E system via the web application provided and should see the current readings of humidity, temperature and air quality of the room in which the system is installed in. The user should then be able to expand this view to see previous readings of the ambient condition selected.

c. Functional Requirements

REQ-1: The system should be able to read the temperature of the room.

REQ-2: The system should be able to read the humidity of the room.

REQ-3: The system should be able to determine smoke/hazardous gases in different parts of the building as well as the oxygen level in the room.

II. Detection of the actual occupancy in the room

a. Description and Priority

This functionality should allow users to know whether or not a room is occupied. This is of high priority to users of the system especially resident managers in determining which rooms are overutilized or underutilized.

b. Stimulus/Response Sequences

The user should be able to access the Ros-E system via the web application provided to see the rooms that are currently occupied or free. Resident managers could get general overview the way in which rooms are used in their buildings.

c. Functional Requirements

REQ-1: The system should be able to detect motion in the room

III. Use of a web application with graphical information of data gathered.

a. Description and Priority

The system should have an available platform in which users can view information generated by the system and interact with the system. A web platform that cuts across all device constraints would be able to allow users to easily interact with the Ros-E system. This is important because it allows the users to get a response from the system to view the information produced as well as interact with the system.

b. Stimulus/Response Sequences

The user should get an aesthetically pleasing display when the user logs unto the Ros-E web platform which would show the analysed results from the data gathered by the system.

c. Functional Requirements

REQ-1: The user should be able to view current information about the system on a web portal.

IV. Data should be well stored in a database for further analysis

a. Description and Priority

The system should have well-structured database to store all the data that it has generated and later use this data to populate the web platform as well as be used for further analyses that can be used to make real time informed decisions on the functioning of appliances connected to systems.

b. Stimulus/Response Sequences

A fixed table should exist for sensors to store its data and the web platform would request from to populate its pages.

c. Functional Requirements

REQ-1: The system should be able to store data recorded into a database for further analysis.

V. The system should be installed with minimal wiring.

a. Description and Priority

The system would be installed into buildings with a small amount of wiring of about 30cm from the system to the endpoint which could be an actuator such as a fan, air condition or humidifier. This is important because it makes Ros-E easy to install into buildings and prevents the system from interfering with the normal wiring of the building.

b. Stimulus/Response Sequences

The system should be able to be installed with ease into any already existing system in the building.

c. Functional Requirements

REQ-1: The system should be able to easily be installed into buildings with minimal wiring.

2.5.2 EXTRA FUNCTIONALITIES

VI. Data analytics

The system should be able to use data mining techniques to make informed decisions on the suitable temperature and humidity preferred by building occupants.

2.5.3 NON-FUNCTIONAL REQUIREMENTS

1. Usability:

The different levels of technical knowledge that might exist inside a building would require the system to be very simple to use. The interface of the system being the web application would have a vital role to play in the usability of the system. Features like colours, layout and even fonts chosen can influence the usability of this web portal. The navigation of the system should also be simple. It would be advisable to use more graphical ways to display the information generated by the system as opposed to using raw figures.

2. Performance:

The system should operate with very low latency of about 12s in performing its various functionalities from sensing the ambient conditions to storing the data into the database to requesting the data to populate the web portal. These instructions should be executed seamlessly with no lags.

3. Interoperability:

The system should be able to make good use of the information it generates. It should also be able to work well with the various components such as other Ros-E systems as well as existing air conditioning systems, lightning systems and humidifiers that make up the system to exchange, access and cooperatively use data to achieve the system’s goal.

4. Security:

The system has access to private information of building occupants like whether a room is occupied, and it is important that it is kept safe. Unauthorized and accidental usage of sensor data retrieved from the Ros-E system should be prevented, and access should be allowed to only legitimate users.

5. Scalability:

The system should be able to cater for increasing number of users as well as an increasing number of nodes that need to be connected in various rooms. The system should be able to work well when installed in a very large building with many rooms and existing systems like fire alarms, ventilation and security systems.

6. Reliability & Availability:

The system should be always available to be used when the user wants to access the system and should also have very low downtime and maintenance time. The setup time for the system should take about a week if the building has about 10 rooms and approximately 1-2 months if there are a large number of rooms. The configuration time should take 3-7 days to finish.

Chapter 3: Architecture and Design

3.1 System Architecture

The Ros-E system is based on the underlying principles of Internet of Things (IoT) technology hence the architecture and design of the system should follow suite. IoT systems generally use two standard architectures being the oneM2M IoT architecture and the IoT World Forum architecture. The oneM2M creates a common services layer which can readily be embedded in field devices to allow communication to application devices. It consists of three layers being the applications layer, services layer and network layer [6]. Figure 3.1 shows the reference model.

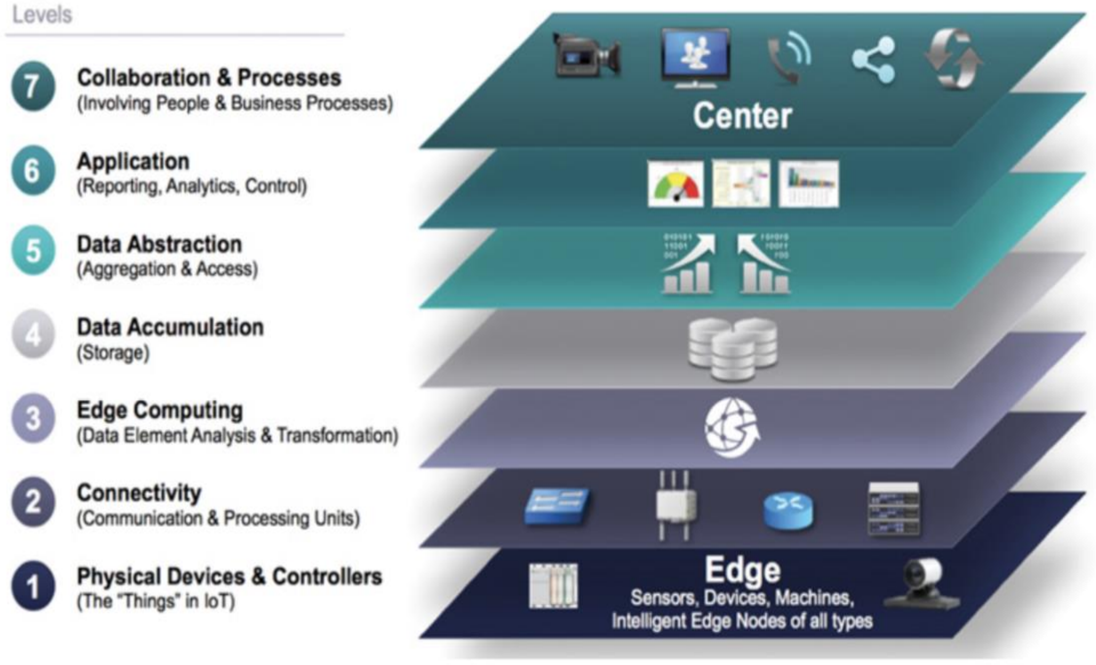


Fig 3.1: IoT Reference model

Image referenced from IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things.

The first layer also known as the physical layer consists of the various endpoint devices and sensors of the system. They are referred to as the “Things” in IoT. They are responsible for generating data. The second layer being the connectivity layer involves how the devices in the first layer are connected to the network and each other. It consists of how data is transmitted throughout the network. Layer 3 also known as the edge computing layer focuses on reducing the data that into useful information that is ready to store and be processed.

The following layers involves storing the data retrieved from the previous layers and then reconciles multiple data formats and ensures that they have consistent semantics. The data is then interpreted into software applications to be viewed by users in a reasonable format and the final step is to ensure collaboration of the analyzed data retrieved with other business processes and this makes IoT useful.

For this project, a modified version of the IoTWF architectural model would be used. This version would combine certain layers that are related and eliminate layers that are not needed. Figure 3.2 shows the layered architecture that would be used for implementing the Ros-E system. This layered architecture would include all the layers of the IoTWF reference model but combines similar layers together.

Layered Architecture

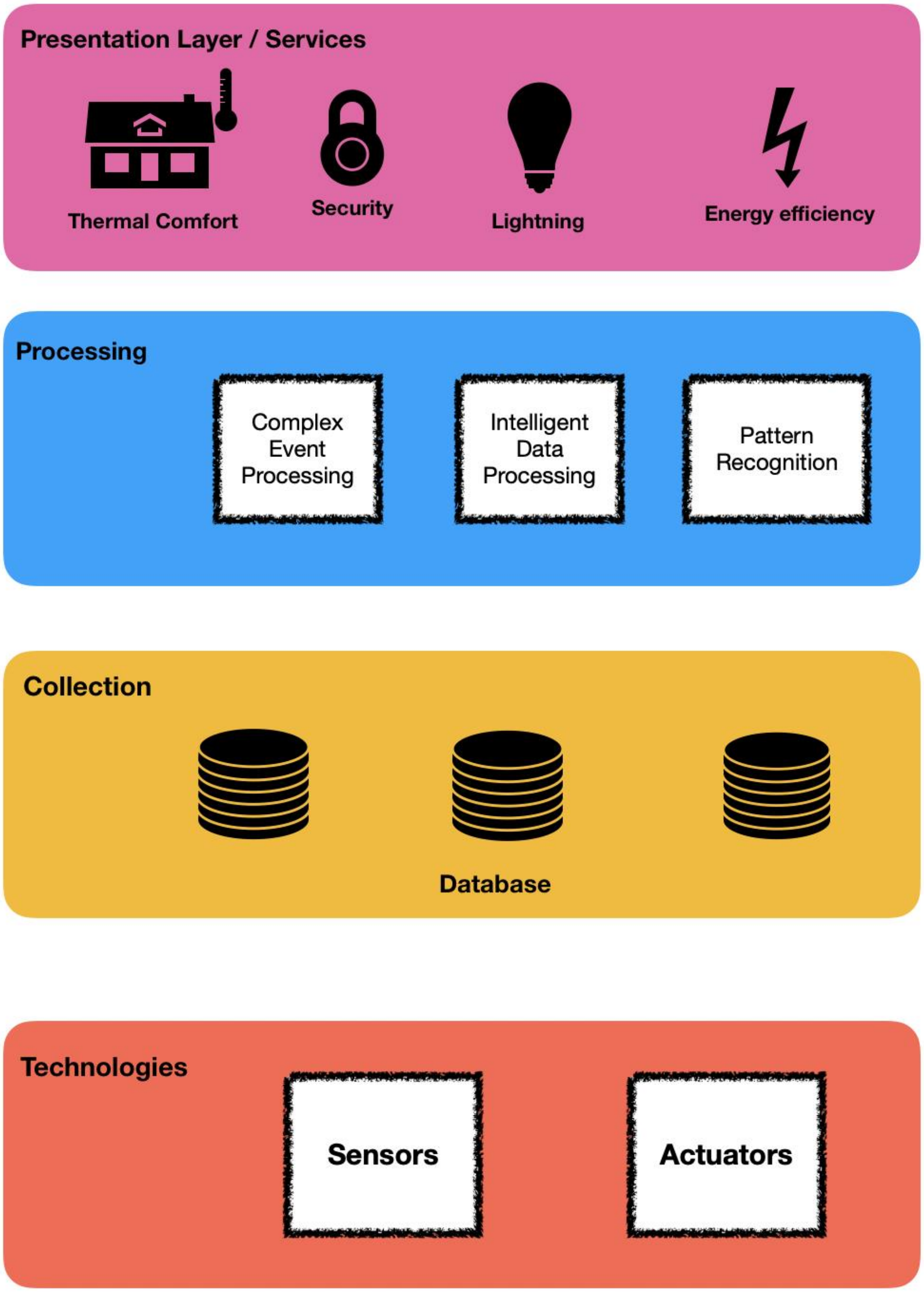


Fig 3.2: Layered Architecture for Ros-E system

Presentation Layer / Services: This layer includes the services that the Ros-E system provides to its users. This includes thermal comfort, security, improved lightning and energy efficiency.

Processing Layer: This layer is responsible for all the activities involved in analyzing the data collected and stored in the database to useful information that can be used to make informed decisions and provide the services mentioned in the layer above.

Collection Layer: This layer involves collecting all the data generated by the technologies layer in a format that is suitable for further processing and analysis.

Technologies Layer: This layer consists of the various endpoint devices that are used to generate data. These devices include motion detection devices, temperature sensing devices, humidity sensing devices and light sensing devices.

The operation of the system can also provide us with another system architecture. The Ros-E system consists of several sensors to measure the ambient conditions. The data generated by the sensors are then preprocessed by edge nodes to provide the data in a suitable format before it reaches the database to be stored and further analyzed. This useful information obtained from analysis is then fed to the web application for the users to make use of. A sequence diagram can be generated from the operation described. Fig 3.3 shows a sequence diagram of the operation of the system.

START END

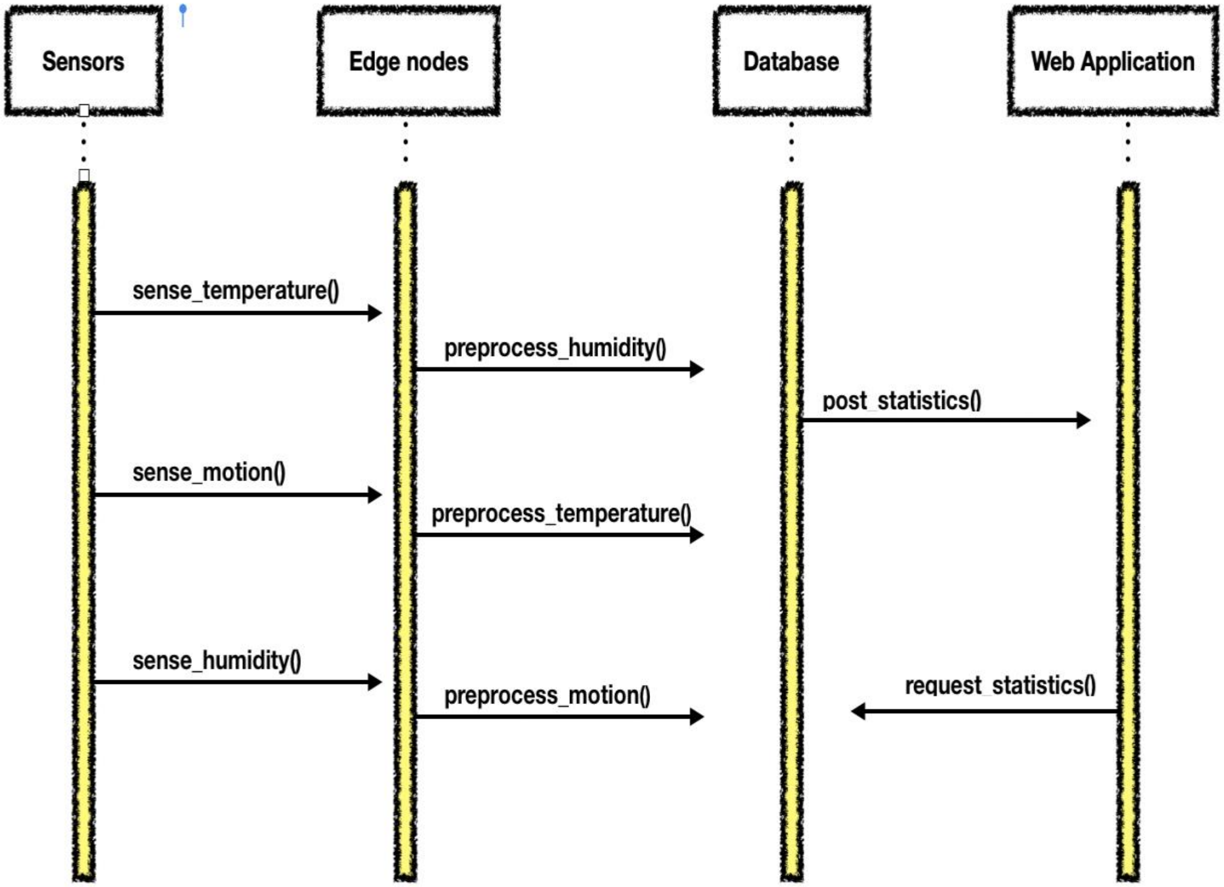


Fig. 3.3: Sequence diagram of Ros-E system

3.2 System Design and Specifications

The Ros-E system involves the use of multiple sensors which would require a microprocessor or a microcontroller to handle the operation of these sensors. The Arduino Uno microcontroller was chosen for the Ros-E system for several factors such as cost, processing power and ease of use. The other alternatives were the Raspberry Pi 3 B+ small single-board computer (SoC), the

BeagleBone Black another SoC and the PCDuino. These alternative devices are generally more powerful than the Arduino Uno and have higher specifications than what is required than the system. The Arduino microcontroller is also better when it comes to working with sensors than the other alternatives. For example, the Arduino Uno also comes for a cheap price of about $20 and can be made cheaper if only the ATmega328P chip that comes with the board is used. This was a major factor in using the Arduino Uno for the Ros-E as making the system cheap and easy to install is a major requirement feature.

The Ros-E system makes use of three different sensors namely the DHT22 sensor, the MQ-135 sensor, PIR sensor and the LDR sensor. The DHT22 sensor measures temperature and humidity. It measures temperatures between -20°C to 80°C which is suitable for measuring room temperatures. It also has a humidity range of 0 – 100%C. The MQ-135 sensor measures the gas concentration of the surrounding air. It was chosen because it has a wide detecting scope as well as fast response and high sensitivity. It also has a stable and long life which means the system would not have to go through maintenance often. Another advantage is that it can be used as a digital or analog sensor. The only disadvantage of this sensor is that it takes about 20 seconds to preheat when it is switched on. The PIR sensor detects motion in an environment. It has a wide range on input voltage varying from 4.0V to 12.0V. It also consumes a low amount of power of 65mA. It covers an angle of 120° and distance of about of 7 meters. The LDR sensor measures the intensity of light. It is small and cheap and can be easily used with a breadboard or a perf board.

The connectivity chosen for Ros-E is the Bluetooth which is a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves. Bluetooth has a range of about 10 meters and a bandwidth of 2.4 GHz. One reason Bluetooth was selected for the Ros-E system is because it requires no line of sight hence can connect through any obstacle. Bluetooth is also cheap and has a lower power consumption than some other connection types.

3.3 Design and Implementation Constraints

A constraint of the system is that it requires internet connection in order to allow remote communication between user and the Ros-E system as well as uploading data read by the sensors to a database for analysis.

Chapter 4: Implementation

This chapter talks about the tools and technologies used in making the Ros-E system as well as the procedures used in developing the system. The chapter focuses on 3 major areas being the hardware component made up of the Arduino Uno and sensors, the web application to help users view data from the system and the database to store the data acquired from the sensor and to link the hardware component to the web application.

4.1 Overview

The Ros-E system is a mixture of hardware and software components communicating with each other through a database by sending and reading sensor data to and from it. The part of the system that does the sensing of data is the hardware component. It consists of an Arduino Uno connected to an LDR (Light Dependent Resistor), a PIR sensor for motion, a DHT22 temperature sensor, a MQ-135 sensor for gas and a HC-05 Bluetooth module. This component is responsible for sensing the ambient conditions in the room and sending this data to the database. The Ros-E module sends the data to the closest module available wirelessly over Bluetooth in a multi hop fashion until it reaches the master module which then sends the data to the database. The data is then viewed by the user through a web application by sending API requests to the database. This data is provided in a form that is easily digestible by the user.

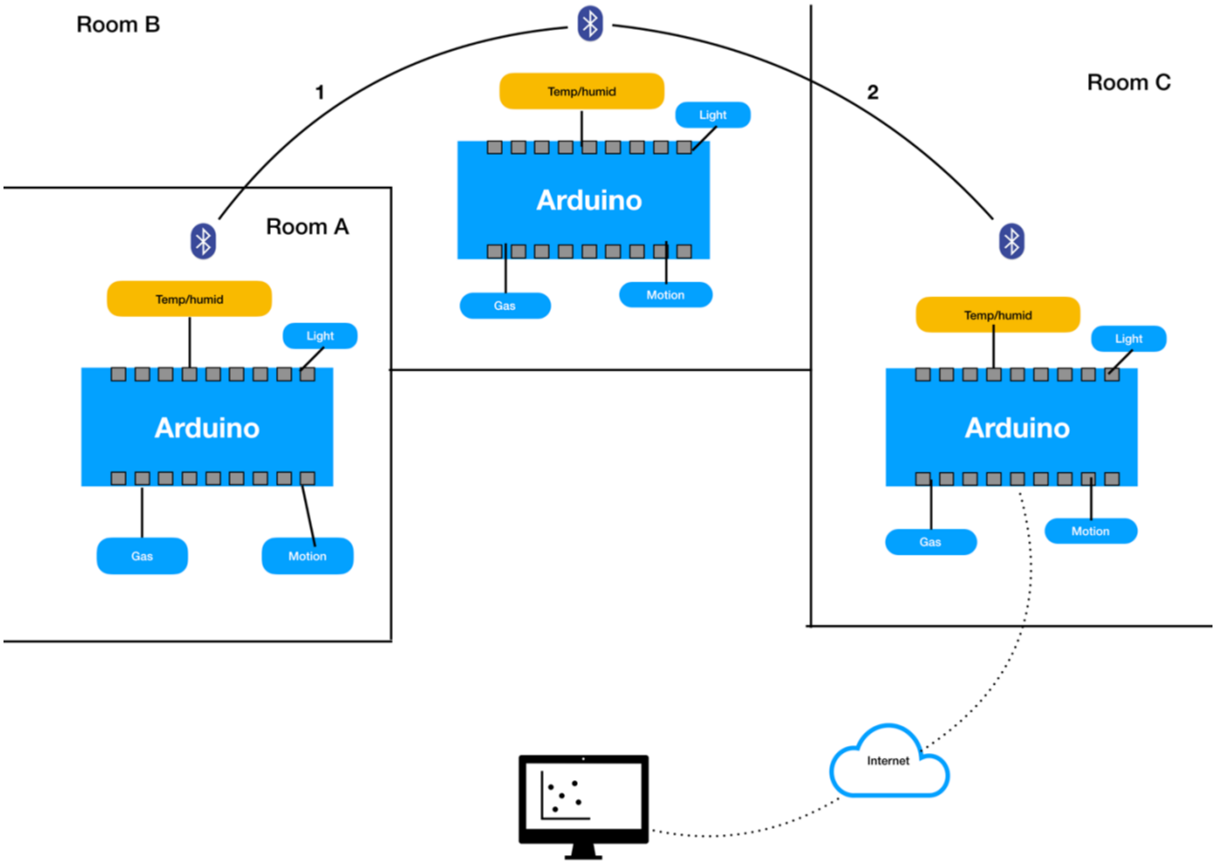


Fig. 4.1: A diagram showing an overview of the Ros-E system

4.2 Tools & Technologies

There are a lot of tools that come together to make the Ros-E system both hardware and software.

Hardware

a. Arduino Uno: This is the microcontroller that collects sensor data from the sensors that are connected to it and either sends this data to another device or to the database through ethernet connection. The Arduino Uno is a microcontroller board based on the

ATmega329P and has 14 digital I/O pins and 6 analog pins. With a flash memory of 32 KB and an SRAM of 2KB, the Arduino Uno is suitable enough to handle the work required by a Ros-E system module.

b. DHT 22 Temperature Sensor: DHT22 utilizes exclusive digital signal collecting technique and humidity sensing technology and can output calibrated digital signal. Small size and low consumption and long transmission distance (20 meters) enables the sensor to be suited in all kinds of harsh applications.

c. LDR sensor: This is a component that has a variable resistance that changes with the intensity of light that falls on it. This could be used to detect whether there is sunlight in a room or whether a room is dark. This could be used to signal some actuator to turn off or on.

d. PIR sensor: This is a component that measures the infrared light radiating from objects in its field. This sensor is often used to detect motion within the range of 25cm to 10m.

e. MQ-135 Gas sensor: This is an air quality sensor that can detect a wide range of gases such as NH3, benzene, alcohol, CO2 and other harmful gases. It has a sensing range of 0.04 mg/L to 4 mg/L.

f. HC-05 Bluetooth module: This is a serial Bluetooth module for the Arduino Uno. It follows the IEEE 802.15.1 protocol and works with serial communication and is TTL compatible. It has a range up to 100 meters.

g. Breadboard: This is a solderless component that allows for temporary prototyping and testing of electronic circuit designs. Components are connected by inserting terminals into the holes of the breadboard and making connections through wires that are positioned appropriately.

h. Ethernet Shield: This shield allows the Arduino Uno to connect to the internet via Ethernet connection. This can be done by using the Ethernet Library to write sketches to the Arduino Uno which connects to the internet via RJ45 Ethernet jack.

Software

a. Arduino IDE: This is a cross-platform application that is used to write, and upload programs often referred to as sketches to Arduino compatible boards. Functions are normally written in C and C++.
b. Vue.js: This is a library for building interactive web interfaces. This library is focused on the view layer only but when used with other supporting libraries and frameworks, Vue.js can be used to make sophisticated single-page web applications.
c. PHP: This is a server-side scripting language that is embedded in HTML. It can be easily integrated with a number of popular databases like MySQL and PostgreSQL. It is open-source and free as well as cross-platform.
d. Laravel: This is a free PHP-based web framework that is intended for the development of web applications following the Model View Controller (MVC) software architecture.

4.3 Implementation Process

The implementation process for this system takes place in three major ways namely; Database design, Hardware development and software development.

4.3.1 DATABASE DESIGN

At the heart of the Ros-E system architecture lies a MySQL database that holds all the information retrieved by the sensors. This database is also connected to the web application and provides it with the data retrieved from the sensors through API requests. The web application also can store information like devices and user profile data into the database. Refer to table for the database design:

Table Name	Columns
Users	user id, User first name, user last name, user email, user password.
Articles	title, subscript, content
Devices	id, reference number, user id
Data	id, timestamp, temp, humid, pir, ldr, gas, gas status, deviceID
Password_resets	email, token, created_at

Table 1.1 Table for database table names and columns

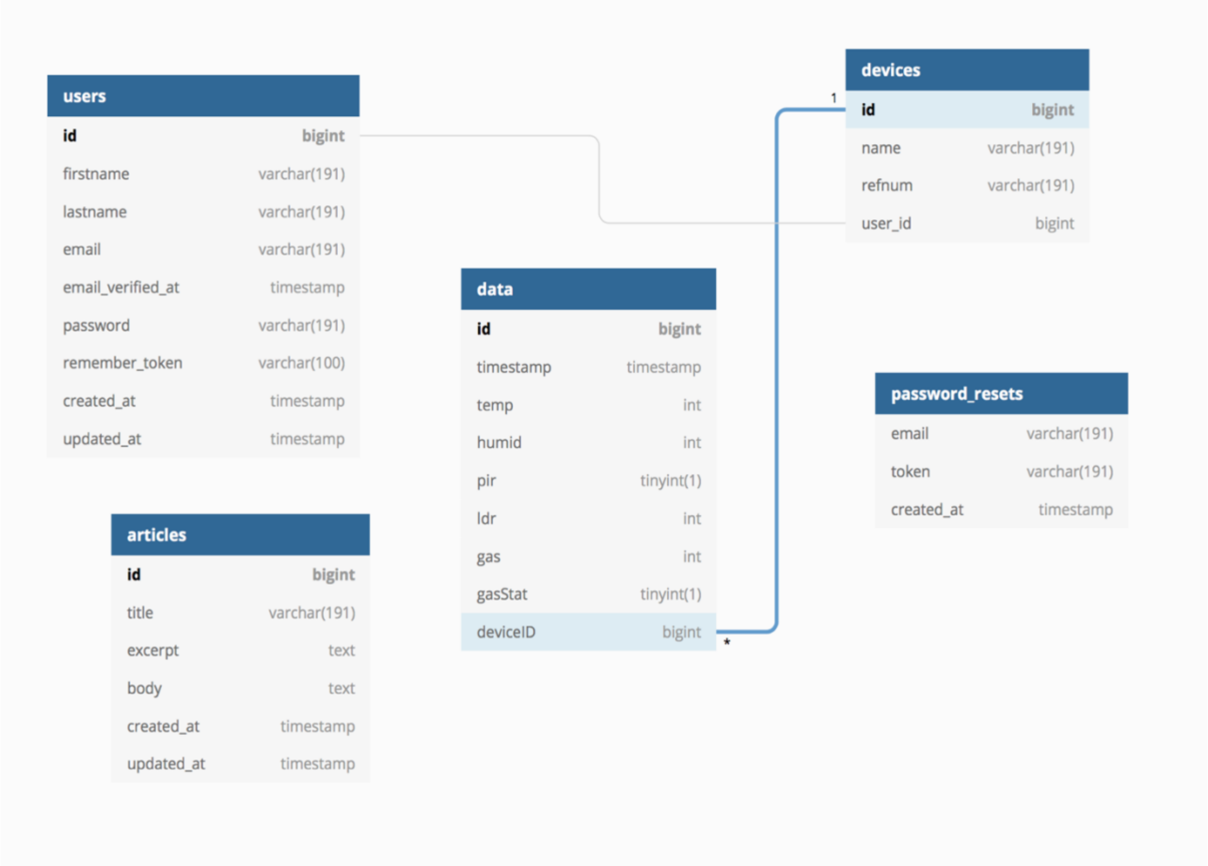


Fig. 4.2: A relational diagram of the database

4.3.2 Physical Connection and Hardware Development

Temperature and Humidity sensor connection: The DHT22 temperature and humidity sensor has 3 pins used to connect with the Arduino.

- A + Pin for connecting to 5V power supply from the Arduino Uno.
- A – Pin for connecting to GND of the Arduino Uno.
- An out pin for connecting to pin 5 of the Arduino Uno.

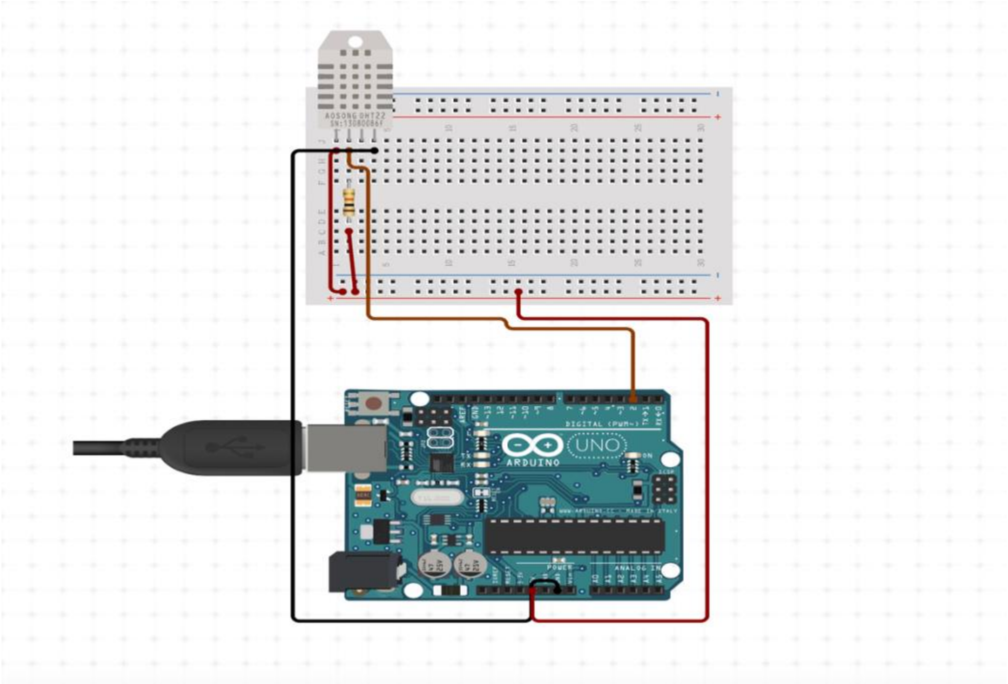


Fig. 4.3: Circuit diagram for Temperature and humidity sensor connection

PIR infrared motion sensor: The PIR motion sensor has 3 pins used to connect with the Arduino Uno.

- A VCC Pin for connecting to 5V power supply from the Arduino Uno.
- An OUT Pin for connecting to pin 2 of the Arduino Uno.
- A GND Pin for connecting to the GND supply of the Arduino Uno.

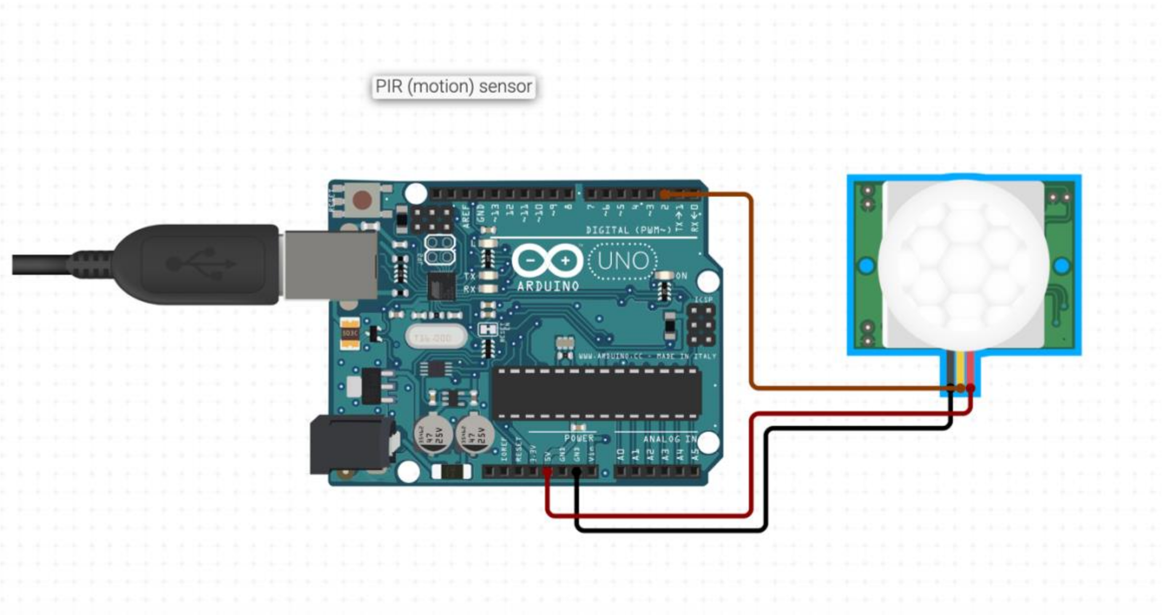


Fig. 4.4: Circuit diagram for Motion sensor connection

MQ-135 Gas Sensor: Although the MQ-135 gas sensor has 4 pins, only 3 pins are used in connecting to the Arduino Uno.

- VCC pin for connecting to 5V power supply from the Arduino Uno.
- A0 pin for connecting to the analog pin A0 of the Arduino Uno.
- GND pin for connecting to the GND supply of the Arduino Uno.

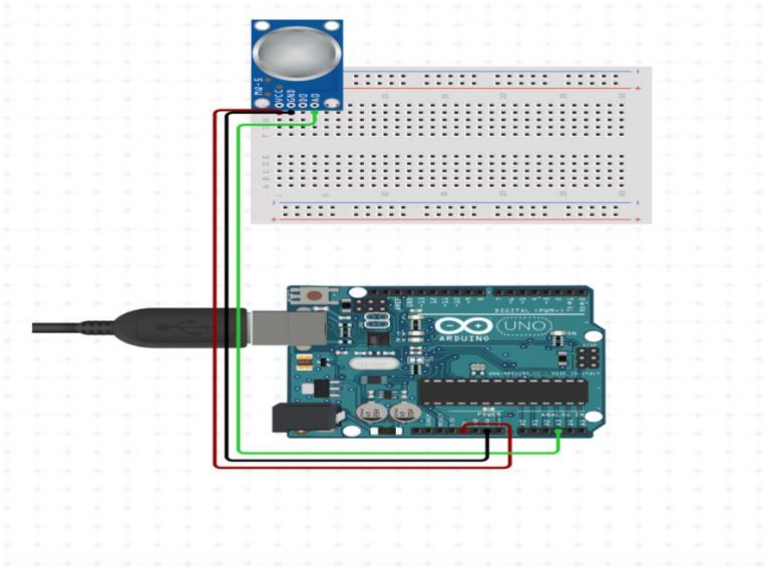


Fig. 4.5: Circuit diagram for MQ-135 connection

<u>LDR Sensor</u>: The LDR sensor is a photoresistor cell that has 2 pins that are not assigned. The LDR sensor can measure a range of values from 0 – 1023. Values above 400 often show sign that there is light in the room and values below 400 show that the room is dark. One end of the LDR is connected to the 5V power supply from the Arduino and the other end is connected to a 10k resistor as well as an analog pin connection to pin A2 of the Arduino Uno. The other end of the resistor is connected to GND of the Arduino to close the connection.

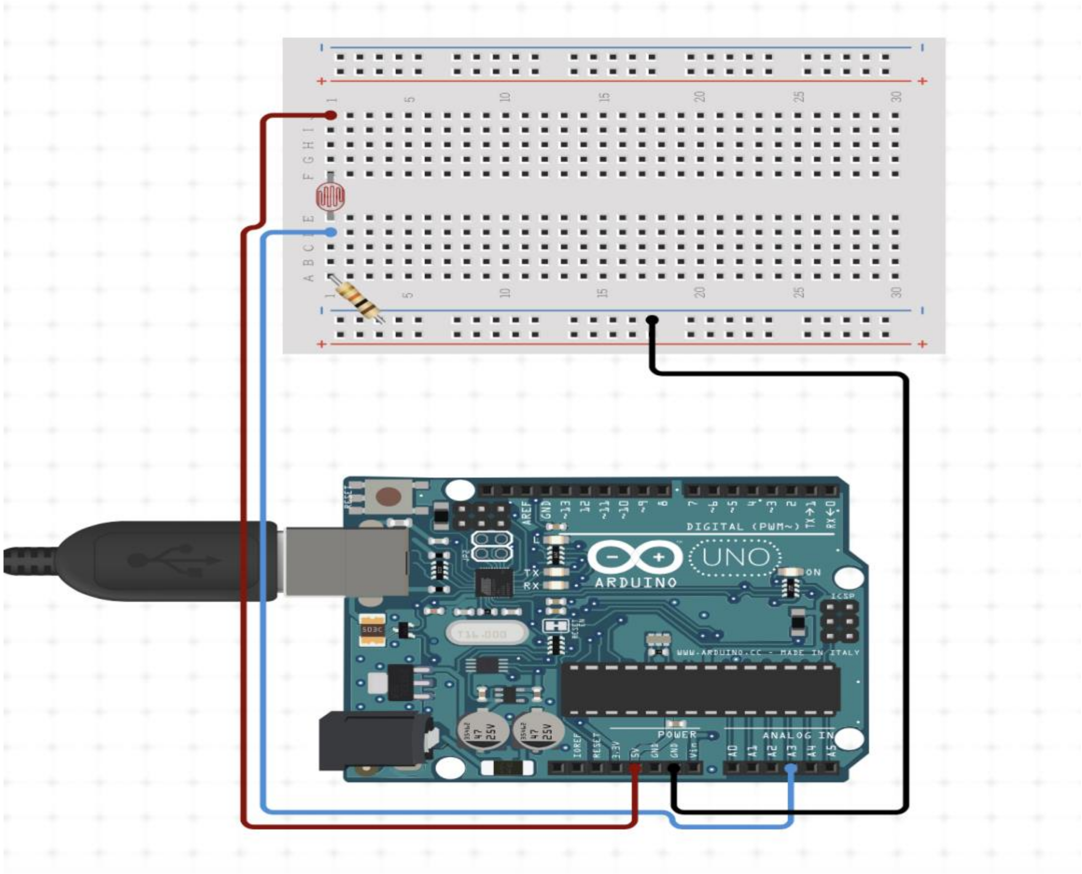


Fig. 4.6: Circuit diagram for LDR connection

<u>HC-05 Bluetooth Module</u>: Although the HC-05 has 6 pins, only 4 pins are connected to the Arduino Uno.

- VCC pin is connected to 5V power supply from the Arduino Uno.
- GND pin is connected to the GND supply of the Arduino Uno.
- TXD pin is connected to pin 0 (RX) pin of the Arduino Uno.
- RXD pin is connected to pin 1 (TX) pin of the Arduino Uno.

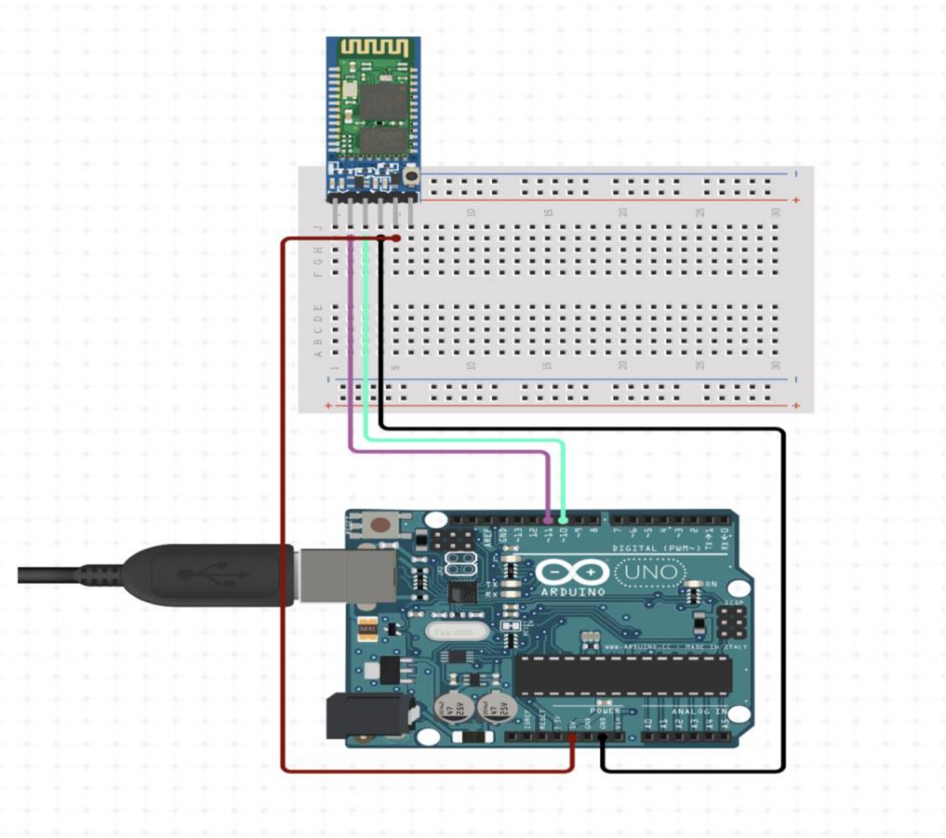


Fig. 4.7: Circuit diagram for Bluetooth connection

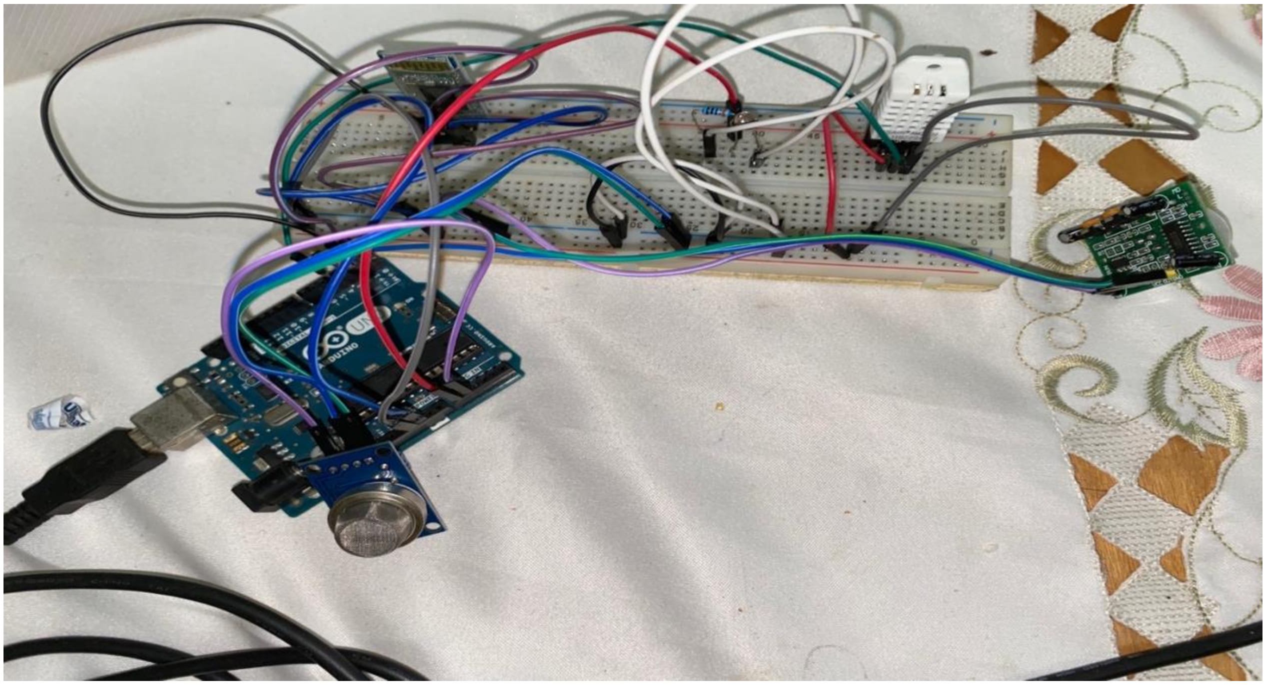


Fig. 4.8: An image of a Ros-E device module

4.3.3 SOFTWARE DEVELOPMENT

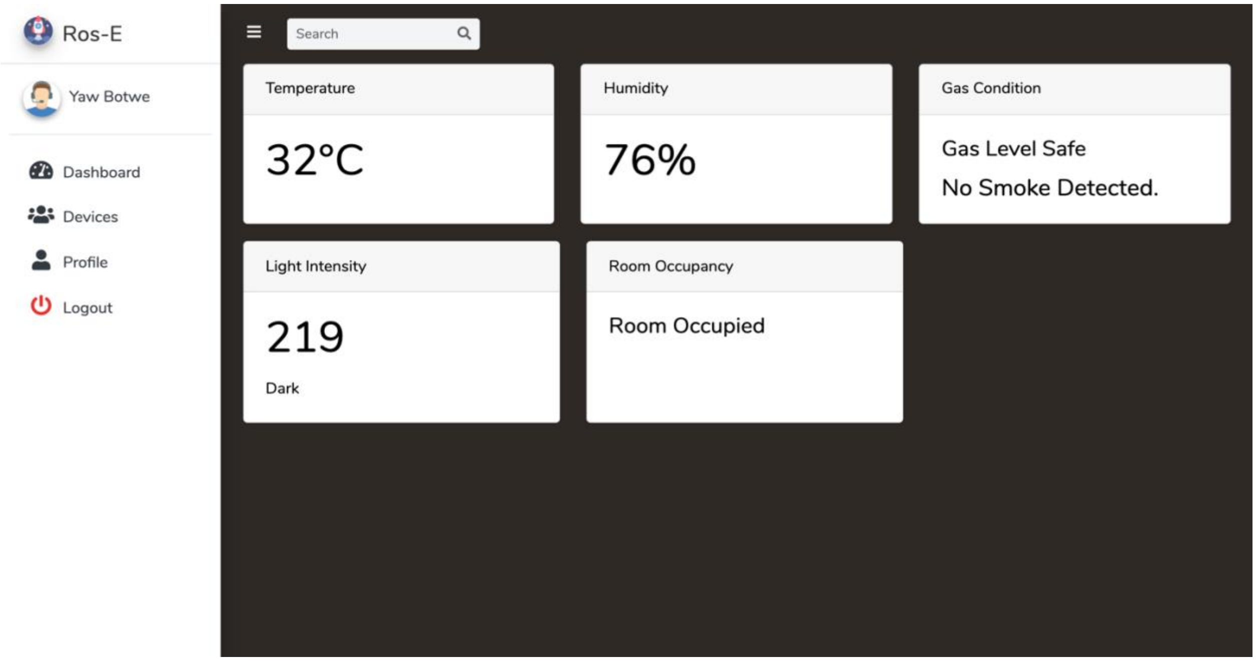
Web Application Description

The software part of the project constitutes of the web application that shows users data that was retrieved from the hardware components of the Ros-E system. The web application was built using the PHP framework, Laravel and the Javascript framework Vue.js. Laravel follows the Model View Controller software architecture which makes it more useful than PHP and makes it suitable for the development of web applications. Laravel was selected for this project because of its performance, features and scalability. Vue.js when used with Laravel makes few calls when requesting for data and makes changes to the user interface without actually reloading pages. This provides users a smooth and lag-free experience when using the Ros-E web platform.

Web Application Features

There are several features that come together to make up the web application. The features that will make the Ros-E system work efficiently are described below;

- Homepage: The homepage is basically a landing page that every user sees when they request for the Ros-E website. It has an about link that takes users to an about page and tells users what the Ros-E system is about and what inspired its creation. The homepage also has a link for news that has news articles published by the Ros-E team. The last aspect of the homepage is the mission link that leads to a page that talks about the mission that the Ros-E system wants to achieve.
- User Authentication pages: This consists of two pages namely the login page and the registration page. The registration page has a detailed form for users to fill personal data in. The login page has a form for the user's email and password. The user's information should correspond with that of the database. User authentication is vital because it helps users to only access data that is important to them as well as protect user data from being accessed by others.
- Dashboard view: This is the main view that is seen when the user logs in. The view contains charts for both temperature and humidity as well as a list of the devices that have been integrated into the Ros-E system.
- Device view: This view has a list of all devices that have been integrated into the Ros-E system. It has functionality for adding devices into a user's network of systems as well as deleting devices from it.
- Update Profile view: This view simply provides a prefilled form for the user to make any updates to his/her profile information stored in the Ros-E database.



Temperature

32°C

Humidity

76%

Gas Condition

Gas Level Safe

No Smoke Detected.

Light Intensity

219

Dark

Room Occupancy

Room Occupied

Fig 4.9: Screenshot of web application

Chapter 5: Testing and Results

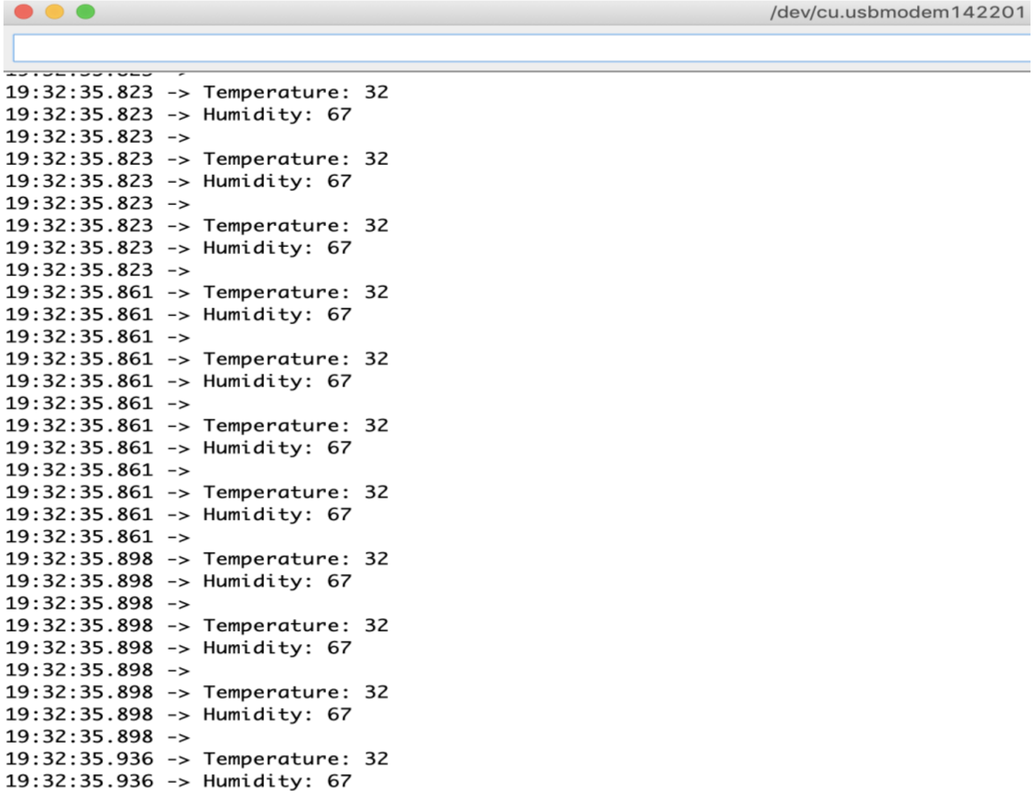
This chapter talks about how the functionalities of the Ros-E system are tested and the insights that came out from testing. Testing is necessary for checking if customer satisfaction has been met and to determine the quality of the product.

5.1 Component Testing

For component testing, each sensor and module that makes up the Ros-E system was individually tested to ensure that the overall functionality of the system would be efficient.

Temperature & Humidity Sensor Testing

The DHT11 sensor was tested by uploading a sketch to the Arduino to see the rate at which temperatures can be recorded as well as the accuracy in the recorded values.



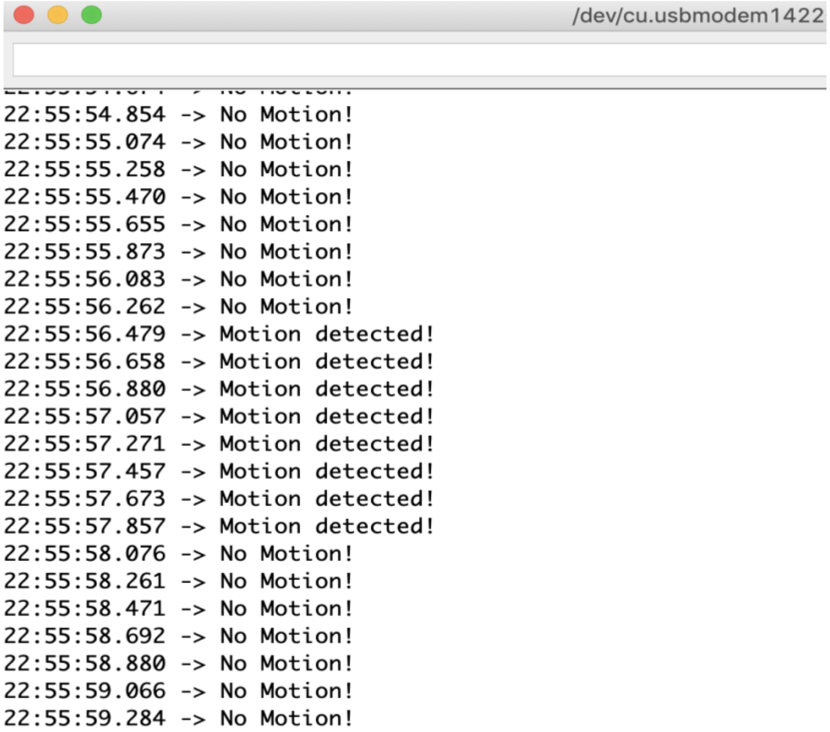
```
19:32:35.823 -> Temperature: 32
19:32:35.823 -> Humidity: 67
19:32:35.823 ->
19:32:35.823 -> Temperature: 32
19:32:35.823 -> Humidity: 67
19:32:35.823 ->
19:32:35.823 -> Temperature: 32
19:32:35.823 -> Humidity: 67
19:32:35.823 ->
19:32:35.861 -> Temperature: 32
19:32:35.861 -> Humidity: 67
19:32:35.861 ->
19:32:35.861 -> Temperature: 32
19:32:35.861 -> Humidity: 67
19:32:35.861 ->
19:32:35.861 -> Temperature: 32
19:32:35.861 -> Humidity: 67
19:32:35.861 ->
19:32:35.861 -> Temperature: 32
19:32:35.861 -> Humidity: 67
19:32:35.861 ->
19:32:35.898 -> Temperature: 32
19:32:35.898 -> Humidity: 67
19:32:35.898 ->
19:32:35.898 -> Temperature: 32
19:32:35.898 -> Humidity: 67
19:32:35.898 ->
19:32:35.898 -> Temperature: 32
19:32:35.898 -> Humidity: 67
19:32:35.898 ->
19:32:35.936 -> Temperature: 32
19:32:35.936 -> Humidity: 67
```

Fig. 5.1: Serial monitor output for temperature and humidity sensor test

The results showed an accuracy of ± 1°C from the true temperature of the room and humidity had an accuracy of 3%.

PIR Motion Sensor Testing

The PIR motion sensor was tested by uploading a sketch to the Arduino to see the frequency at which motion is detected accurately.



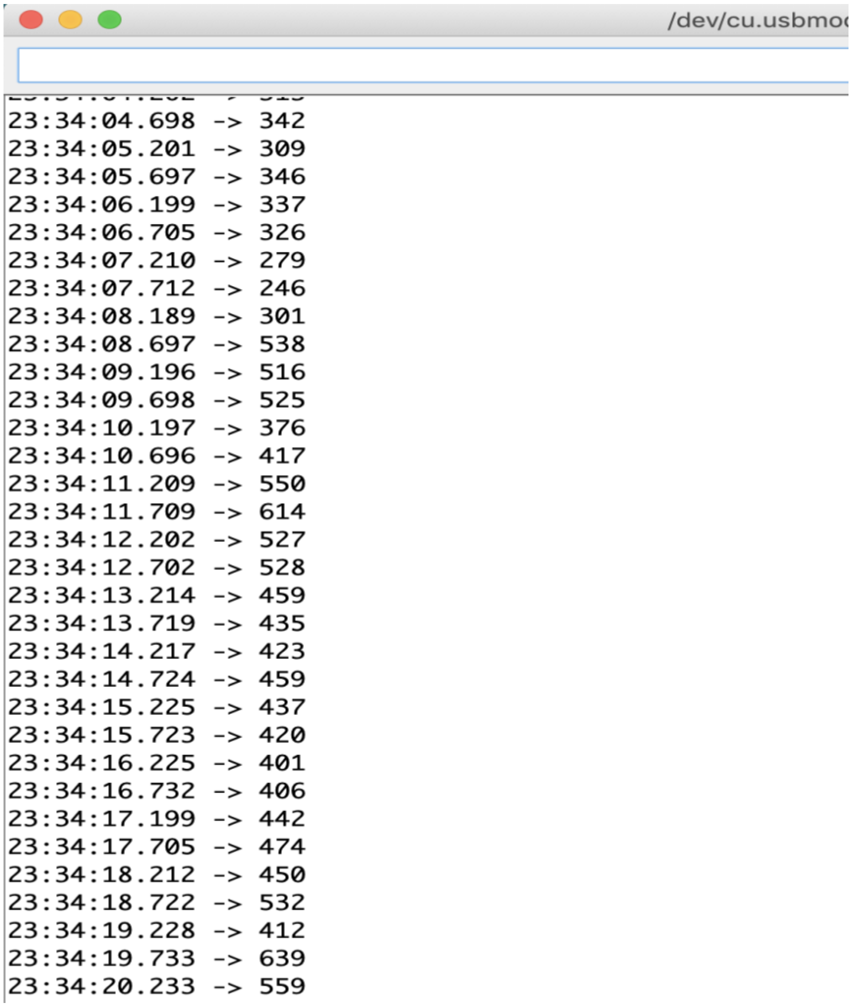
```
22:55:54.854 -> No Motion!
22:55:55.074 -> No Motion!
22:55:55.258 -> No Motion!
22:55:55.470 -> No Motion!
22:55:55.655 -> No Motion!
22:55:55.873 -> No Motion!
22:55:56.083 -> No Motion!
22:55:56.262 -> No Motion!
22:55:56.479 -> Motion detected!
22:55:56.658 -> Motion detected!
22:55:56.880 -> Motion detected!
22:55:57.057 -> Motion detected!
22:55:57.271 -> Motion detected!
22:55:57.457 -> Motion detected!
22:55:57.673 -> Motion detected!
22:55:57.857 -> Motion detected!
22:55:58.076 -> No Motion!
22:55:58.261 -> No Motion!
22:55:58.471 -> No Motion!
22:55:58.692 -> No Motion!
22:55:58.880 -> No Motion!
22:55:59.066 -> No Motion!
22:55:59.284 -> No Motion!
```

Fig 5.2: Serial monitor output for motion sensor test

From the results, the motion sensor observed to have a delay of 2 seconds after making a reading. This delay is suitable for the Ros-E system as it measures its sensor values within a 30-minute interval.

LDR sensor testing

The LDR sensor was tested by uploading a sketch to the Arduino to see how responsive the sensor works.



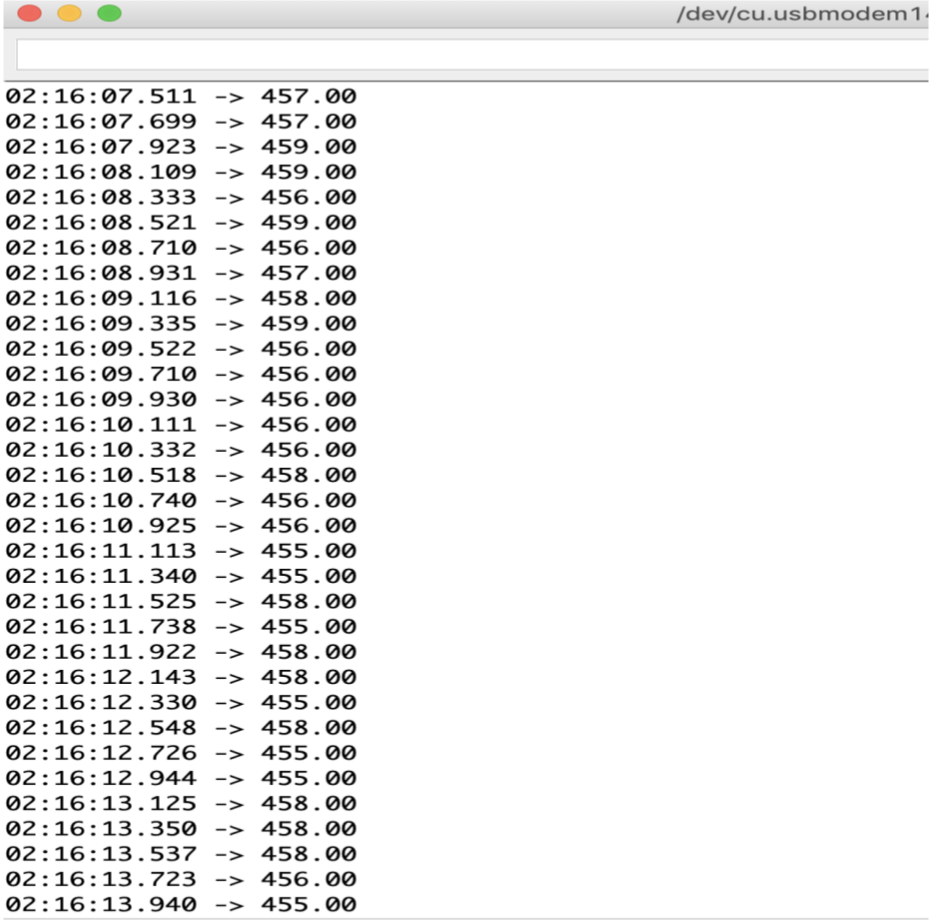
```
23:34:04.698 -> 342
23:34:05.201 -> 309
23:34:05.697 -> 346
23:34:06.199 -> 337
23:34:06.705 -> 326
23:34:07.210 -> 279
23:34:07.712 -> 246
23:34:08.189 -> 301
23:34:08.697 -> 538
23:34:09.196 -> 516
23:34:09.698 -> 525
23:34:10.197 -> 376
23:34:10.696 -> 417
23:34:11.209 -> 550
23:34:11.709 -> 614
23:34:12.202 -> 527
23:34:12.702 -> 528
23:34:13.214 -> 459
23:34:13.719 -> 435
23:34:14.217 -> 423
23:34:14.724 -> 459
23:34:15.225 -> 437
23:34:15.723 -> 420
23:34:16.225 -> 401
23:34:16.732 -> 406
23:34:17.199 -> 442
23:34:17.705 -> 474
23:34:18.212 -> 450
23:34:18.722 -> 532
23:34:19.228 -> 412
23:34:19.733 -> 639
23:34:20.233 -> 559
```

Fig 5.3: Serial monitor output for LDR sensor test

The LDR was also tested by shining varying light intensities on its surface and recording the time it took for changes in values to occur. From the results the LDR sensor had a response time of about 90ms which is suitable for the Ros-E system.

Gas sensor testing

The MQ-135 sensor was tested by uploading a sketch to the Arduino to see how responsive the sensor works.



```
02:16:07.511 -> 457.00
02:16:07.699 -> 457.00
02:16:07.923 -> 459.00
02:16:08.109 -> 459.00
02:16:08.333 -> 456.00
02:16:08.521 -> 459.00
02:16:08.710 -> 456.00
02:16:08.931 -> 457.00
02:16:09.116 -> 458.00
02:16:09.335 -> 459.00
02:16:09.522 -> 456.00
02:16:09.710 -> 456.00
02:16:09.930 -> 456.00
02:16:10.111 -> 456.00
02:16:10.332 -> 456.00
02:16:10.518 -> 458.00
02:16:10.740 -> 456.00
02:16:10.925 -> 456.00
02:16:11.113 -> 455.00
02:16:11.340 -> 455.00
02:16:11.525 -> 458.00
02:16:11.738 -> 455.00
02:16:11.922 -> 458.00
02:16:12.143 -> 458.00
02:16:12.330 -> 455.00
02:16:12.548 -> 458.00
02:16:12.726 -> 455.00
02:16:12.944 -> 455.00
02:16:13.125 -> 458.00
02:16:13.350 -> 458.00
02:16:13.537 -> 458.00
02:16:13.723 -> 456.00
02:16:13.940 -> 455.00
```

Fig. 5.4: Serial monitor output for gas sensor test

From the results of the test, the gas sensor had a response time that was less than or equal to 10s which is suitable for the Ros-E system.

5.2 User Testing

The main part of the Ros-E system that users interact with is the web application. User testing would then need to be done on the navigation, layout and design of the web application by users of the system. The first test was to see how the web application would look on different browsers and at different aspect ratios. This test went well as the web application uses bootstrap which allows for very interactive web applications.

The next test was to give users the chance to interact with the system without given any prior instructions or directions on how to use the system. This is to find out how easy it is to use the web application and to see if the layout of the web application is desirable. Users were able to successfully find their way around the web application with little to no issues when interacting with the system.

Chapter 6: Limitations, Future Work & Conclusion

6.1 Conclusion

The Ros-E system is a stepping block for the development of smart building infrastructure and in the long run the development of smart cities. This project sought to develop an efficient system for monitoring the ambient conditions existent ins most buildings for occupants to view. The implementation of this project involved interaction between both hardware components being the Arduino Uno and an array of sensors and software components being the MySQL database and the Laravel web application. The system is capable of monitoring temperature, humidity, motion, gas and light intensity and reports this data to the user over a web application whilst storing this information for further analysis.

This project proves the possibility of creating a real-time smart building monitoring system that aids in the safety, usability and comfort of building occupants.

6.2 Limitations

One major limitation to the Ros-E system is that it currently has ethernet as its only mode of connection to the internet. This means that necessary cabling in rooms must be done for the Ros-E system to function properly. In instances where there are no ethernet connection in the room or there is a problem with the ethernet cable, the Ros-E system would be completely offline and data from sensors would not be logged into the database.

Another limitation of the Ros-E system is that the current system design is not as small as desired with the use of the Arduino Uno microcontroller. The system is also not robust as it uses jumper wires to connect with an array of sensors on a breadboard. These cables may easily be removed or tampered with if not placed in the system's casing.

6.3 Future Work

The Ros-E system currently monitors the ambient conditions in the rooms of a building and sends that data to be viewed on a web application. This is just the beginning of a long journey in making buildings smarter. One factor that can be added to the Ros-E system is the use of the ATmega328P chip as well as printed circuit boards in the manufacturing of the sensor modules. This would provide a smaller, more robust and aesthetically pleasing system that would be used to monitor the ambient conditions of a room.

Another additional functionality that can be incorporated into future Ros-E systems is the ability to interact with a wide range of electronic appliances over a wireless connection. For example, the Ros-E system can detect the change in the gas concentration in a building and sound the fire alarm all over a wireless connection.

The ability of the Ros-E system to analyze data that it stores in the database and make informed and automated decisions would be beneficial to users of the system. Big data analytics would be able to efficiently measure the use of electricity by appliances, identify faults, provides root-cause analysis, and prioritizes opportunities for improvement based on cost, comfort and maintenance impact. For example, the Ros-E system of the future can learn the optimal temperature that occupants prefer in each room and adjust the HVAC system in the room to suit the occupant's preference without any human intervention.

References

[1] J. Vcelak, A. Vodicka, M. Maska, and J. Mrna. 2017. Smart building monitoring from structure to indoor environment. In *2017 Smart City Symposium Prague (SCSP)*, IEEE, Prague, Czech Republic, 1–5. DOI:https://doi.org/10.1109/SCSP.2017.7973859

[2] S. M. Saad, A. M. Saad, A. M. Kamarudin, A. Zakaria, and A. Y. Shakaff. 2013. Indoor air quality monitoring system using wireless sensor network (WSN) with web interface. In *2013 International Conference on Electrical, Electronics and System Engineering (ICEESE)*, IEEE, Kuala Lumpur, Malaysia, 60–64. DOI:https://doi.org/10.1109/ICEESE.2013.6895043

[3] T. Lee, S. Jeon, D. Kang, L. W. Park, and S. Park. 2017. Design and implementation of intelligent HVAC system based on IoT and Bigdata platform. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, 398–399. DOI:https://doi.org/10.1109/ICCE.2017.7889369

[4] J. Dutta and S. Roy. 2017. IoT-fog-cloud based architecture for smart city: Prototype of a smart building. In *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, 237–242. DOI:https://doi.org/10.1109/CONFLUENCE.2017.7943156

[5] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar. 2017. A smart home energy management system using IoT and big data analytics approach. *IEEE Transactions on Consumer Electronics* 63, 4 (November 2017), 426–434. DOI:https://doi.org/10.1109/TCE.2017.015014

[6] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. 351.