Angeles University Foundation

**College of Engineering and Architecture**

GSM BASED HOME AUTOMATION USING ARDUINO

**by:**

[Daniel Patrick Punzalan](mailto:punzalan.danielpatrick@auf.edu.ph)

[Jesus Luigi Sison](mailto:sison.jesusluigi@auf.edu.ph)

June, 2022

ABSTRACT

Home automation makes it possible for electrical appliances (such as lighting systems, televisions, and refrigerators) to be easily accessed and effectively controlled. The key reasons behind home automation are making peoples' lives easy, driven more by technology and effectively managed. One of the rapidly evolving facets of today's technology is the means through which electronic devices can be controlled. These days, there are a number of applications that have been created with the help of a variety of technologies to make our lives simpler.

This posed a big challenge/risk of electrical disaster (such as fire outbreak) to people outside their vicinity that mistakenly left their appliances ON/OFF, as the case may be, and could also lead to energy wastage for appliances left connected to the mains for a period longer than required. Hence, there is a need for an effective means of remotely controlling home appliances beyond one's vicinity. However, this paper proposed a GSM based home automation system as a means to address the aforementioned issues.

The main objective of this project is to design a system that can remotely turn on and off home appliances using SMS and with a built-in phone application. System development involves implementing the different hardware modules (power supply, GSM, microcontroller, relays, and loads) RTC Module making up the system as well as programming the system controller (Arduino AtMega) and using App Inventor.

The result of the system after testing showed that the system is very effective in the remote control of electrical appliances, only that an average delay of switching the appliance depends on how strong the GSM signal is observed before the switching takes place whereas the scheduling timer properly executing the time input and it depends on the time scheduling input to switch the loads.

***Keywords:*** *Home Automation, GSM, SMS, Arduino*

ACKNOWLEDGEMENT

We would like to acknowledge and give our warmest thanks to our thesis advisor, Dr. Ruel Reyes who made this work possible. His continuous guidance and advice carried us through all the stages of constructing and establishing this project.

We would also like to thank the AUF CEA faculty for letting our defense be an unforgettable moment as Computer Engineering students, and for your brilliant comments and suggestions, thanks to you.

We would also like to give special thanks to our family as a whole for their continuous support and understanding when undertaking the project. Their love and prayers was what sustained us this far.

Finally, we would like to thank God, for letting us get through all the difficulties. We have experienced your guidance day by day. We will keep entrusting you for our future endeavors.

# LIST OF ABBREVIATIONS AND ACRONYMS

* **AT –** Attention
* **EEPROM –** Electrically Erasable Programmable Read Only Memory
* **GSM** – Global System for Mobile Communications
* **I2C –** Inter-Integrated Circuit
* **IC –** Integrated Circuit
* **IDE –** Integrated Development Environment
* **I/O –** Input/Output
* **LCD –** Liquid Crystal Display
* **RTC –** Real Time Clock
* **RX –** Receive
* **SIM –** Subscriber Identity Module
* **SMS** – Short Messaging Service, text messaging service
* **SRAM –** Static Random Access Memory
* **TX –** Transmit
* **UART –** Universal Asynchronous Receiver/Transmitter
* **USB –** Universal Serial Bus

TABLE OF CONTENTS

**ABSTRACT 1**

**ACKNOWLEDGEMENT 2**

[**LIST OF ABBREVIATIONS AND ACRONYMS**](#_cp6952kzt3yh) **3**

**CHAPTER I: INTRODUCTION 8**

[1.1 BACKGROUND OF THE STUDY](#_cvmjra8cajgs) **8**

[1.2 STATEMENT OF THE PROBLEM](#_ccpsm7yfjotr) **9**

[1.3 SIGNIFICANCE OF THE STUDY](#_6zjyqk1449pn) **10**

[1.4 SCOPE AND LIMITATIONS](#_qouy8smok8t4) **10**

[1.5 DEFINITION OF TERMS](#_5mx20dn2xjce) **11**

**CHAPTER II: REVIEW OF RELATED LITERATURE AND STUDIES 12**

[2.1 HOME AUTOMATION TECHNOLOGY](#_7o0pcgeinn8t) **12**

[2.2 GSM TECHNOLOGY](#_m9h6a2hvdbpg) **13**

[2.3 GSM BASED HOME AUTOMATION](#_qks67s4qzzwf) **14**

[2.4 APP INVENTOR INTEGRATION](#_ody5edgsca9a) **15**

[2.5 ARDUINO INTEGRATION](#_uyhfji8nindl) **15**

[2.6 PROPOSED SYSTEM](#_bj1t4sfr24r7) **16**

**CHAPTER III: RESEARCH METHODOLOGY 18**

[3.1 HARDWARE DEVELOPMENT](#_90n719ix0hsi) **18**

[3.1.1 Arduino Mega 2560](#_jdtargj5uraf) 18

[3.1.1.1 Arduino Mega Components](#_yp4n7gp0fnhy) 19

[3.1.1.1.1 Two Wire Interface (TWI)](#_loedhtwc5lvv) 20

[3.1.1.1.2: In-Circuit Serial Programming ( ICSP)](#_eyf3z775j51j) 21

[3.1.1.1.3 ATMEGA2560 & ATMEGA16U2](#_zaxr96c7bbbb) 21

[3.1.1.1.4 Pulse width modulation Pins( PWM)](#_191l4pqk1c2b) 21

[3.1.1.1.5 Power Pins](#_mz85jpckgm9s) 21

[3.1.1.1.6 Analog Pins and Digital Pins](#_s9akwjl8gf8e) 22

[3.1.2 SIM900A](#_r5ky3efhwq29) 22

[3.1.3 Switch Box](#_sg8a93uh5j2j) 24

[3.1.4 Relay](#_pr8ycm41w2ne) 24

[3.1.5 RTC Module (Real-Time Clock)](#_g0eouix3wijj) 25

[3.1.6 Push Button Switch](#_9unw9ac6mb8j) 26

[3.1.7 Liquid Crystal Display (LCD)](#_s5dvu3v18o35) 26

[3.1.8 Buzzer](#_k45yfpixnqy7) 27

[3.2 FIRMWARE DEVELOPMENT](#_4zx8fsv3rx1) **28**

[3.2.1 Arduino IDE](#_xm7fs0dpaf4g) 28

[3.2.1.1 Features of the Arduino IDE](#_67cv94j65kmz) 29

[3.2.2. App Inventor](#_hx3h0ivjlln1) 30

[3.3 PROGRAM FLOWCHART](#_njod9svzj52m) **31**

[3.4 TESTING OF THE PROJECT](#_arbzqweubgq3) **32**

[3.5 SCHEMATIC DIAGRAM](#_ngbvfd395y48) **33**

**CHAPTER IV: PRESENTATION AND ANALYSIS OF DATA 35**

[4.1 AUTOMATIC SWITCHING OF APPLIANCE](#_y1bgclo4c6m7) **35**

[4.2 REMOTELY CONTROLLING SWITCH BOX USING MOBILE PHONE](#_8p05pdo5xv2n) **37**

[4.3 INTEGRATION OF EXISTING WEB APPLICATION](#_qimdiehx5svc) **39**

[4.4 OTHER FEATURES OF THE SYSTEM](#_g067njczrzfc) **41**

[4.5 USER EVALUATION](#_eqb0233v09oz) **42**

[4.5.1 User Evaluation Results](#_8bp1ne57mclv) 43

**CHAPTER IV: SUMMARY, CONCLUSION, AND RECOMMENDATION 48**

[5.1 SUMMARY](#_s228606dxu4w) **48**

[5.2 CONCLUSION](#_g4pjx4s7tk73) **48**

[5.3 RECOMMENDATIONS](#_cipsxlt4nboa) **49**

**REFERENCES 50**

[**APPENDIX**](#_oqha1mu0iel2) **53**

LIST OF FIGURES

[Figure 1: Arduino Mega 2560](#_6ijwt0lzny5q) [1](#_6ijwt0lzny5q)9

[Figure 2: Components of the Arduino Mega 2560](#_qf0w1hqochcp) 20

[Figure 3: TWI Bus Configuration](#_e6qdy7ruegu8) 20

[Figure 4: ICSP Header Pin Configuration](#_g2g84brv9qt) [2](#_g2g84brv9qt)1

[Figure 5: SIM900A GSM module](#_biwnggahw4zs) [2](#_biwnggahw4zs)3

[Figure 6: Switch Box](#_g37p78o3yfpr) [24](#_g37p78o3yfpr)

[Figure 7: HL-54S Relay Module](#_dct1ioukri07) [25](#_dct1ioukri07)

[Figure 8: DS3231 RTC Module and Pinout](#_t3r6d5a6arwn) [25](#_t3r6d5a6arwn)

[Figure 9: Push Button Switch](#_c1ng61jxnq7v) [26](#_c1ng61jxnq7v)

[Figure 10: LCD 16x04 module](#_gw84vd5ubjds) [2](#_gw84vd5ubjds)7

[Figure 11: Piezo Buzzer](#_9gdlj1f0ysw7) [27](#_9gdlj1f0ysw7)

[Figure 12: Arduino IDE Interface](#_kemz09dazttp) [28](#_kemz09dazttp)

[Figure 13: Arduino IDE Serial Monitor](#_wxa2r1olay52) [29](#_wxa2r1olay52)

[Figure 14: App Inventor Interface](#_lkwxda5auiuc) [30](#_lkwxda5auiuc)

[Figure 15: System flowchart](#_2ytz671b8oar) [32](#_2ytz671b8oar)

[Figure 16: Feedback Message](#_8p5ghq7xsnyb) [34](#_8p5ghq7xsnyb)

[Figure 17: Schematic Diagram of GSM Based Home Automation System](#_hzd4ztroidjj) [35](#_hzd4ztroidjj)

[Figure 18: Remote Control](#_w11oezv8wnfp) [3](#_w11oezv8wnfp)7

[Figure 19: Block Diagram of the GSM Control Appliances.](#_q739d17jp6r) [3](#_q739d17jp6r)8

[Figure 20: Schematic Diagram of the GSM control appliance](#_d1cr6qh5j21y) [3](#_d1cr6qh5j21y)8

[Figure 21: Actual Switchbox](#_u4aq99kbyyff) [3](#_u4aq99kbyyff)9

[Figure 22: The Design of Web application](#_70opfp54twjh) 40

[Figure 23: The Blocks of the Web Application](#_6u8ua391kuei) 41

[Figure 24: Scheduling timer](#_5m8rkk7mk5w8) [4](#_5m8rkk7mk5w8)2

Figure 25: Results about Computer Literacy 43

Figure 26: Results about Awareness of the System 44

Figure 27: Results about Power Consumption 44

Figure 28: Results about People’s Lifestyle 45

Figure 29: Results about SMS Awareness 46

Figure 30: Results about Phone Application 46

Figure 31: Results about System Aspects 46

LIST OF TABLES

[Table 1: Arduino Mega 2560 Specifications](#_2u6rrwqmyddr) [18](#_2u6rrwqmyddr)

[Table 2: Basic AT commands](#_fxra6uo0234r) [23](#_fxra6uo0234r)

LIST OF APPENDICES

Appendix A: User Evaluation Questionnaire Sample 53

Appendix B: Arduino IDE Source Code 57

CHAPTER I

INTRODUCTION

# 1.1 BACKGROUND OF THE STUDY

Home automation has been extensively developed and make researches for years. Home automation is the concept of controlling all home electronic appliances into an automatic system. It is one of the standout technologies with the most noticeable improvements in innovation and in every household has access for home automation nowadays and has become a convenient way of simplifying controls to home appliances. With the continuous advancement of technology, things are becoming simpler and easier. In the scope of industrialization, automation is a step beyond mechanization, where it provides human operators with machinery to assist them with the muscular requirements of work (Henry, 2013). Most people have mobile phones as a necessity thus, the world indeed has transformed into a digital village. At any given moment, any individual can be contacted with the mobile phone. But given the advancements, the mobile phone applications cannot just be restricted to sending SMS or starting conversations. New innovations and ideas can be generated from it that can further enhance its capability.

The idea of home automation began in the 1900s with the introduction of electric power distribution that eventually led to creation of labor-saving appliances such as water heaters, washing machines, refrigerator, etc. (Rohith Sai, 2019). By 1975, the home automation technology was based on controlling signals through electric power transmission courtesy of the development of X10 products which included console, lamp and appliance modules that led to commercialization. Then, Arduino was introduced in 2005 to help students and professionals create working projects that can relate within the digital environment and led to many contributions and remains a great tool for creating projects today.

There are already existing products and components that aid automation of home appliances. Through this project we have tried to show how to control appliances in a house with added features that promote energy efficiency to reduce overload of appliances. The project GSM Based Home Automation using Arduino is a switch box device that can control appliances switching by sending an SMS through the GSM module with Arduino board.

# 1.2 STATEMENT OF THE PROBLEM

The aim of this project is to develop a home automation system based on the GSM technology to automatically switch home appliances. Specifically, the project pursues to answer the following:

1. How to Design a system that will automatically switch (on & off) an appliance.
2. How to Design a switch box that can be controlled remotely using a mobile phone.
3. How to Integrate an existing web application for home appliance control.
4. How to Design other features for the home automation system.

# 1.3 SIGNIFICANCE OF THE STUDY

The comfort of being able to control home devices without being physically present have been utilized for research given that it promotes efficiency and productivity. Hence, there is an ascending need to do this study. The proposed system is an extended approach of home automation. The results of this research will benefit the following:

* **Households/ home owners**. This will aid households who are looking for convenience, manage power consumption and tend to forget to ON or OFF devices at their homes.
* **Future researchers**. This research will serve as a reference for future studies that will tackle in creating and developing the similar technology used that can either enhance, replace or integrate with the proposed system.

# 1.4 SCOPE AND LIMITATIONS

The project is intended to control the switching of home appliances through sending an SMS from the mobile device. The system is limited in the use of GSM and a SIM card is needed for registration to control the home remotely using a mobile phone, and if the GSM or SIM is misplaced, the functions of the system stops. It has a feature of a scheduling timer that will schedule the switching on/off the appliances and temperature recorder. It can notify the change of the appliance switching through SMS. It also has a remote control phone application. No security standards were put in place to the system against hackers.

To develop the whole project, the researchers will utilize the Sim900A GSM module connected to an Arduino Mega microcontroller that has a software program, Arduino IDE. The mobile phone is used to send commands and as a recipient to receive the responses and alerts from the microcontroller unit, whereas the Arduino Board is the unit responsible for controlling the different components and acts as the brain of the system. The GSM module is responsible for communication between the microcontroller unit and the mobile phone. While the programming of the code is based on the Arduino aided with libraries of the given hardware modules. The phone application is built using App Inventor. The switch box is designed as an enclosure to control the appliances from the relay that is connected to the Arduino board.

# 1.5 DEFINITION OF TERMS

1. **Home Automation-** process of remotely controlling and managing various household devices.
2. **Home Appliances–** are electrical or mechanical devices that perform certain domestic tasks, such as cooking or cleaning.
3. **Arduino –** type of microcontroller for educational and professional purposes.
4. **Microcontroller instruction sets—** is a collection of commands that the device is configured to perform in order to execute its intended purpose.
5. **X10 Products—-**industry standard for electronic devices used for home automation

CHAPTER II

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter contains the related researches such as Home Automation system, Arduino, MIT App Inventor and GSM technology and application for home automation will be looked into as they relate to the control systems for the proposed project.

# 2.1 HOME AUTOMATION TECHNOLOGY

Janani et al. (2021) stated the home automation system as the conversion of all the electrical and electronic devices into an automatically operational system along with user-defined conditions reducing human efforts, interruption, and improves productivity.

Monowar et al. (2019) and their study about developing an radio frequency-based wireless home automation system prototype discussed some possible impacts that the technology has brought in the society in recent years. The authors have pointed out that the system can regulate security, heating, lights, cooking, washing appliances, power management, video, and audio systems as well as several products such as monitoring systems, and home publishing.

Singh et al. (2016) stated in their review Paper on Smart GSM Based Home Automation System that there are many home automation technologies available in the market that are popular technologies are X10, Z-wave, Zigbee, GSM technology etc.

In their project developing a low-cost Arduino-based home automation system spearheaded by Guden and Akdas (2016). The authors used an Xbee Shield, a module that allows Arduino to initiate wireless communication through Zigbee, which is a wireless protocol and one of the popular available home automation technologies, is used as the main controller for developing their system.

Sowah et al. (2020) and their study about developing Wireless Home Automation System using an open source software. The authors utilized OpenHab2, a Java-based home automation software. It incorporates the use of Arduino Mega and Raspberry Pi for the hardware and software that achieves the automation functions.

# 2.2 GSM TECHNOLOGY

The author Dubey (2012) and his study of developing an ARM-based GSM mobile for home security and automation highlighted the Global System for Mobiles or GSM network as a widely available technology capable of receiving and sending messaging services through using mobile devices, especially smartphones.

Karkera et al. (2018) in their study entitled “GPS-GSM based Vehicle Tracking System” explained the components of the GSM module which consists of a modem that accepts SIM card, and operates over a subscription to a mobile operators, the modem looks like just a mobile phone and this modem is used to send the location with the help of the Global Positioning System (GPS) via text.

The SIM card is one feature of the GSM service. It is compact and detachable in appearance and is widely used for the subscription of information and contacts. This allows the retrieval of information after switching handset on. In a research of developing a key exchange protocol for SIM card and service providers by Kerem and Yazgan (2021) the authors stated that this module enables users to link each other to different network operations. For this project to be achieved, a SIM card is required to establish identification and connection between the user and the supported home appliances.

# 2.3 GSM BASED HOME AUTOMATION

In the study of Jivani (2014) entitled GSM Based Home Automation System Using App-Inventor for Android Mobile Phone, the author detailed that the Arduino board is used to control the appliances by using GSM technology. They proposed a system that is executed when a mobile phone generates SMS messages based on the user commands and sends it to the GSM modem attached to the Arduino and controls the home appliances.

Malik and Bodwade (2017) explores the literature of different home automation systems. They provided solutions proposing the use of the mobile phone and GSM technology for communication purposes. SMS based home automation, General Packet Radio Service based home automation and Dual Tone Multi Frequency based home automation, options considered for GSM communication.

Isa and Sklavos (2017) propose a home automation system supported through the use of a GSM module. They presented some features including the notification and alert messaging system for the end users and the authorities in the home security offices.

# 2.4 APP INVENTOR INTEGRATION

The input of the project will be based on user commands through SMS technology and with the help of designing a remote control mobile application from the software known as App Inventor.

Research steered by Mohd Annuar et. al (2019) in his study of developing an application that manages home appliances using a mobile application integration from a mobile device. The authors used App Inventor for the programming. The objective of their system is to facilitate home owners with simplicity. In addition, they develop a model that comprises devices aiding the control of electrical appliances and its security measurements.

Abdul Raheem (2017) proposed Bluetooth based home automation wherein the Arduino was used to control the motor, bulb, fan and heater as the sample home devices by an integration of App Inventor mobile application with Bluetooth technology. The system requires Bluetooth-enabled devices and can select and add another device.

# 2.5 ARDUINO INTEGRATION

Arduino is an accessible microcontroller for digital learning intended for the professionals, programmers, hobbyists and anyone in creating digital developments such as projects that require sensors. The Arduino comprises both hardware board to make physical pin connections and software environment to program code designed for a desired function of a system.

Gunawan et al. (2018) conducted research evaluating the hardware and software performance of smart home systems using Internet of Things wherein the Arduino IDE was used to interface the required sensors and appliances required for their prototype.

In a study conducted by Hussain et al. (2019) about facilitating communications for disaster operations, Arduino can be utilized as the controller to process the transmit and receive process of the data fetch from the smartphone.

The fusion of modern technology, design and energy was discussed from the study of Tirian (2017) entitled “Smart Home Automation with Arduino” wherein the author used a variant of the board, the Arduino Mega in controlling various smart devices given the light sensor, temperature and humidity sensor, and the voltage stabilizer creating a smart home environment with the help of the function of the given embedded system.

# 2.6 PROPOSED SYSTEM

The proposed conceptual framework can be said to be a GSM based home Automation system using Arduino. The GSM innovation gives wide reach so the researchers can work for their specific framework. This innovation gives financially perceptive answers for controlling the home appliances distantly.

Figure 1 shows the block diagram of the system wherein each of the required hardware components and home devices is identified and the flow of the proposed home automation system.

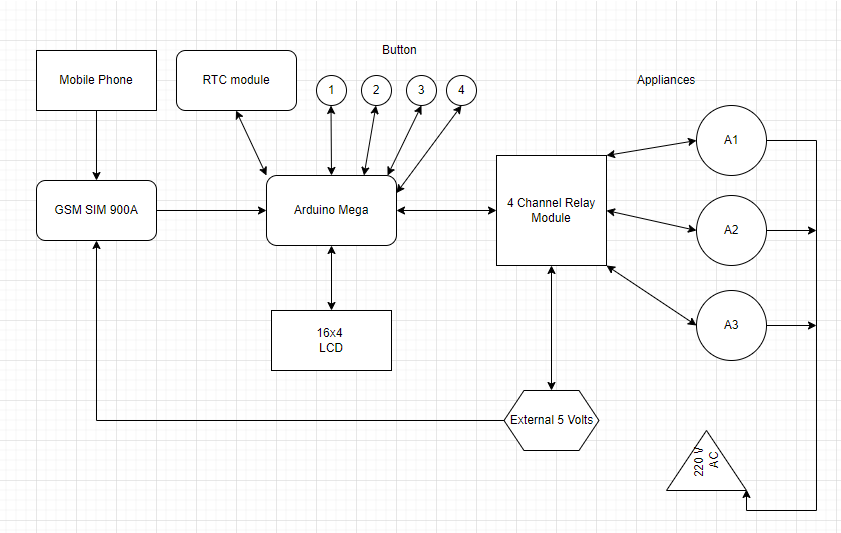


Figure 1: The System Conceptual Framework

The mobile phone will send a SMS through the GSM module while the GSM module will receive a text message from the mobile device. Then, it will go to the Arduino Mega microcontroller. The 4-relay channel module will serve as the voltage control of appliances ON/OFF from a message coming to the GSM module. The LCD module is used as an interface or display of the output of the appliance status and the RTC will set the time fixed for automatically turning ON/OFF the appliances at home. The researchers use buttons to set up a time scheduling in a specific appliance. The switchbox can manually turn on and turn off the appliances.

CHAPTER III

RESEARCH METHODOLOGY

This chapter explains various methodologies that were utilized for the project. The methodologies will include the discussions of hardware and firmware development, project testing, and analysis.

# 3.1 HARDWARE DEVELOPMENT

This section explains the contents of the hardware requirements to be used for the project that includes the integration of modules such as GSM Module, Relay, Liquid Crystal Display, and RTC, and other components to an Arduino Mega board.

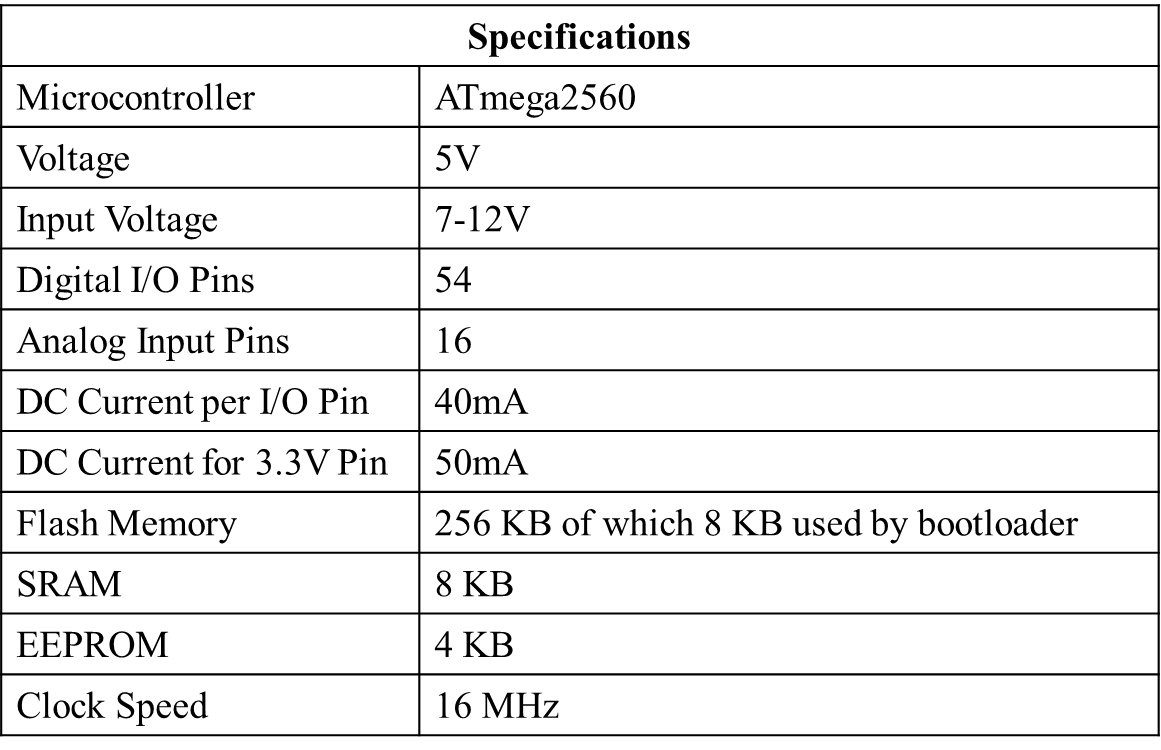
## 3.1.1 Arduino Mega 2560

The system uses the Arduino Mega 2560 board shown in Figure 1 which encompasses an ATmega2560 microcontroller that features 54 digital I/O pins, 15 PWM outputs, 16 analog inputs, 4 UARTs, 16 MHz crystal oscillator, USB port, power jack, and reset button. It includes all components required to support the microcontroller. With the use of a USB cable, you can connect it to a computer or power it with an AC-to-DC adapter or battery to run. Table 1 shows the specifications of the board.



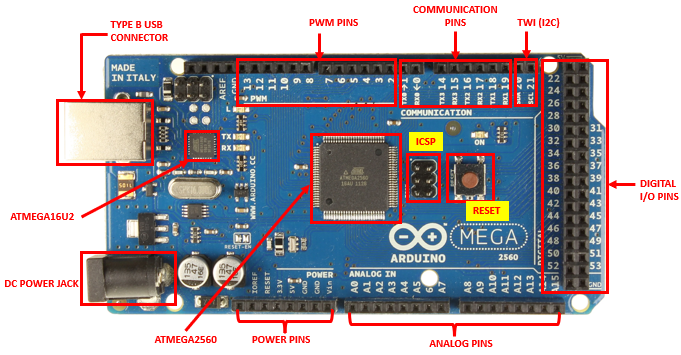
### Figure 1: Arduino Mega 2560

### Table 1: Arduino Mega 2560 Specifications



### **3.1.1.1 Arduino Mega Components**

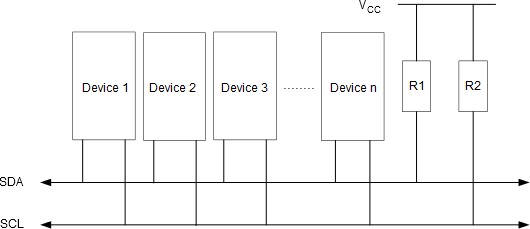
This part explains the functions of the components/ features of the Arduino Mega 2560 board which mainly includes pins such as for communications, power, analog and digital variables. The graphic shown in Figure 2 depicts the components of the Arduino Mega board.



### Figure 2: Components of the Arduino Mega 2560

#### **3.1.1.1.1 Two Wire Interface (TWI)**

It is also known as the I2C bus. It is a bus interface communication protocol that features two bidirectional dedicated lines, serial data and serial clock. Serial Data (SDA) is used for data transferring, located in pin 20 of the board while Serial Clock (SCL) is used for clock signal, located in pin 21. The illustration of the protocol shown in Figure 3.



#### Figure 3 : TWI Bus Configuration

#### **3.1.1.1.2: In-Circuit Serial Programming ( ICSP)**

These pins allow the user to reprogram the firmware on the Arduino board. There are two ICSP connections on the Arduino Mega 2560, one for "ATMEGA2560" and "ATMEGA16U2." Figure 4 shows the ICSP header pin configuration present in the Arduino board.



#### Figure 4: ICSP Header Pin Configuration

#### **3.1.1.1.3 ATMEGA2560 & ATMEGA16U2**

The microcontroller used in the Arduino Mega board is ATmega2560. The ATmega16u2 microchip is located in the USB interface and it is responsible for USB and serial communications. Both components were developed by Atmel.

#### **3.1.1.1.4 Pulse width modulation Pins( PWM)**

The pulse width modulation or PWM pins allow the user to produce analog outputs via digital methods (e.g. brightness of an LED).

#### **3.1.1.1.5 Power Pins**

There are 10 pins set up to power devices connected to the Arduino mega, including three pins for ground, one pin for 5 volts, one pin for 3.3 volts, and two pins for analog and digital reference voltage. The Arduino Mega 2560 can be powered by an AC adaptor (outside diameter 5.5mm, inner diameter 2.1mm).

#### **3.1.1.1.6 Analog Pins and Digital Pins**

Analog pins are utilized for reading analog sensors or components. This means that it can interface 0 to 1024-resolution analog devices. In terms of voltage, 5 volts will be 1024. And, digital pins are utilized for reading digital sensors or components. The board includes 54 digital pins, with 0 (RX0) and 1 (TX0) used for data reception and transmission. Digital implies 0 and 1.

**3.1.1.1.7 Communication Pins**

The communication pins of the Arduino Mega 2560 consist of 4 hardware UARTs (transmitting and receiving) pins allowing communication with serial devices.

## 3.1.2 SIM900A

Figure 5 shows the SIM900A Modem is utilized as the component for the GSM-based communication for the proposed home automation system. It is built with Dual Band GSM and works on frequencies 900/ 1800 MHz allowing functionality to transmit voice, SMS, data, and fax information at fast speeds while consuming little power. SIM900A can search these two bands automatically.



## Figure 5: SIM900A module components

The frequency bands and baud rate can be set and configured respectively through the AT commands. It is used to control the modems where AT stands for Attention. These sets of commands have four classifications: Test, Read, Set, and Execution. Some of the notable AT commands are listed in the Table 2

### Table 2: Basic AT commands

## 3.1.3 Switch Box

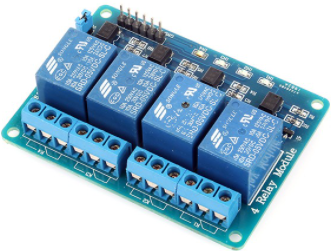
The switch box is the casing for the manual control for any electrical appliances. It can be used to turn ON/OFF appliances through pressing the manual switch. The figure for the sample component is shown in Figure 6.

### 

### Figure 6 : Switch Box

## 3.1.4 Relay

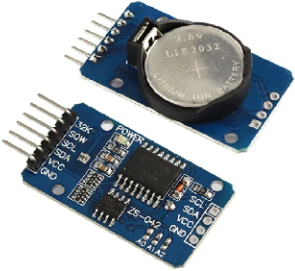
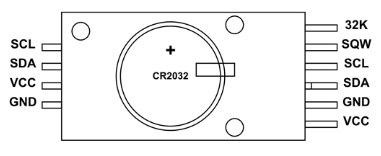
The relay is a switch that is operated electrically by an electromagnet. It is used to control high voltage [electronic devices](https://microcontrollerslab.com/electronics-projects/) such as [motors](https://microcontrollerslab.com/speed-control-dc-motor-using-microcontroller-part-2/) and as well as low voltage [electronic devices](https://microcontrollerslab.com/electronics-components-and-why-they-do/). It will be used in the system to control high voltage devices such as home appliances by simply connecting it using the Arduino Mega board. The model to be used is the HL-54S, which is a four-channel relay module as shown in Figure 7.



### Figure 7: HL-54S Relay Module

## 3.1.5 RTC Module (Real-Time Clock)

The RTC module as shown in Figure 8 is an integrated circuit that stores the current time and date. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. It records seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year.



### Figure 8: DS3231 RTC Module and Pinout

## 3.1.6 Push Button Switch

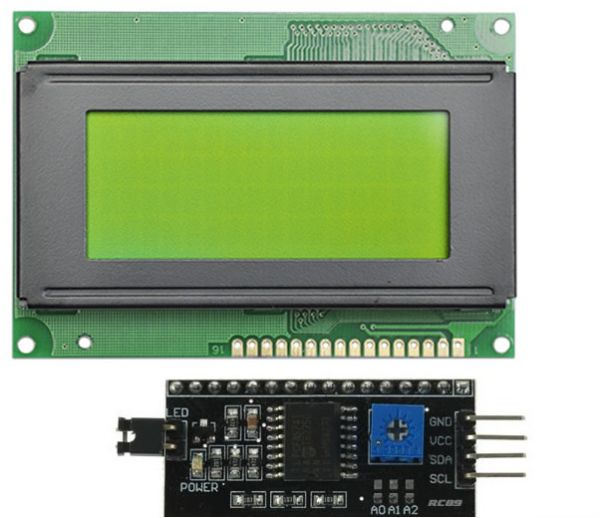
It is utilized in control functions such as dashboards and interfaces in automation systems. The connection to their contact terminals is established by means of screw terminals. A sample of a push button switch is presented in Figure 9.



### Figure 9: Push Button Switch

## 3.1.7 Liquid Crystal Display (LCD)

An LCD (Liquid Crystal Display) screen is an electronic display module. This component will present the status message of the proposed system. The model to be used for this hardware is the 16x04. It can display 16 characters per line with 4 available lines. It is capable of displaying 224 different characters and symbols. The image of the module is presented in figure 10.



### Figure 10: LCD 16x04 module

## 3.1.8 Buzzer

The buzzer module is used to make alarm sound when initialization executes and for button control of the timer. For this project, the researchers used the piezo buzzer. The picture of the component is presented in Figure 11.



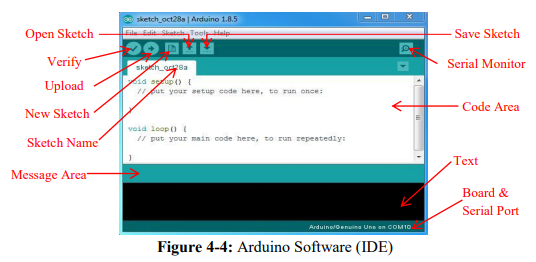
### Figure 11: Piezo Buzzer

# 3.2 FIRMWARE DEVELOPMENT

This section contains the firmware details to be used for the control of the project. It includes mainly the integration of Arduino IDE and App Inventor. Below are a brief description and details of the firmware components.

## 3.2.1 Arduino IDE

The Arduino Integrated Development Environment shown in Figure 12, is a cross-platform application that is written in functions from C and C++. With the IDE, the user can write computer code or the sketch for the system and can do additional programming or make changes depending on the purpose. It allows ease of programming the firmware at higher-level programming and conversion to hex for loading to the microcontroller. For the full source code for the project, refer to Appendix B.

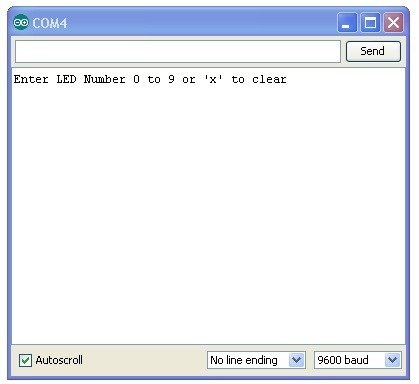
Figure 12 : Arduino IDE Interface

### **3.2.1.1 Features of the Arduino IDE**

Arduino's IDE contains a variety of handy features for developing and debugging sketches and controlling Arduino from a computer. It includes the following: Serial Monitor, Verify/Compile, and Libraries–in which will be discussed below.

**3.2.1.1.1 Serial Monitor**

The Serial Monitor is a pop-up window that functions as a separate terminal for transmitting and receiving serial data at the specified baud rate. It helps in debugging sketches and seeing data supplied by a functioning sketch. The image shown in Figure 13 shows a sample program showing the message placed in the serial window.



#### Figure 13: Arduino IDE Serial Monitor

**3.2.1.1.2 Verify/Compile**

This process generates a written sketch based on the rules specified in the board, board settings, and build options. Displays status and potential faults in a console-like view at the window's bottom. The Arduino environment uses the C/C++ compiler to accomplish the compilation process.

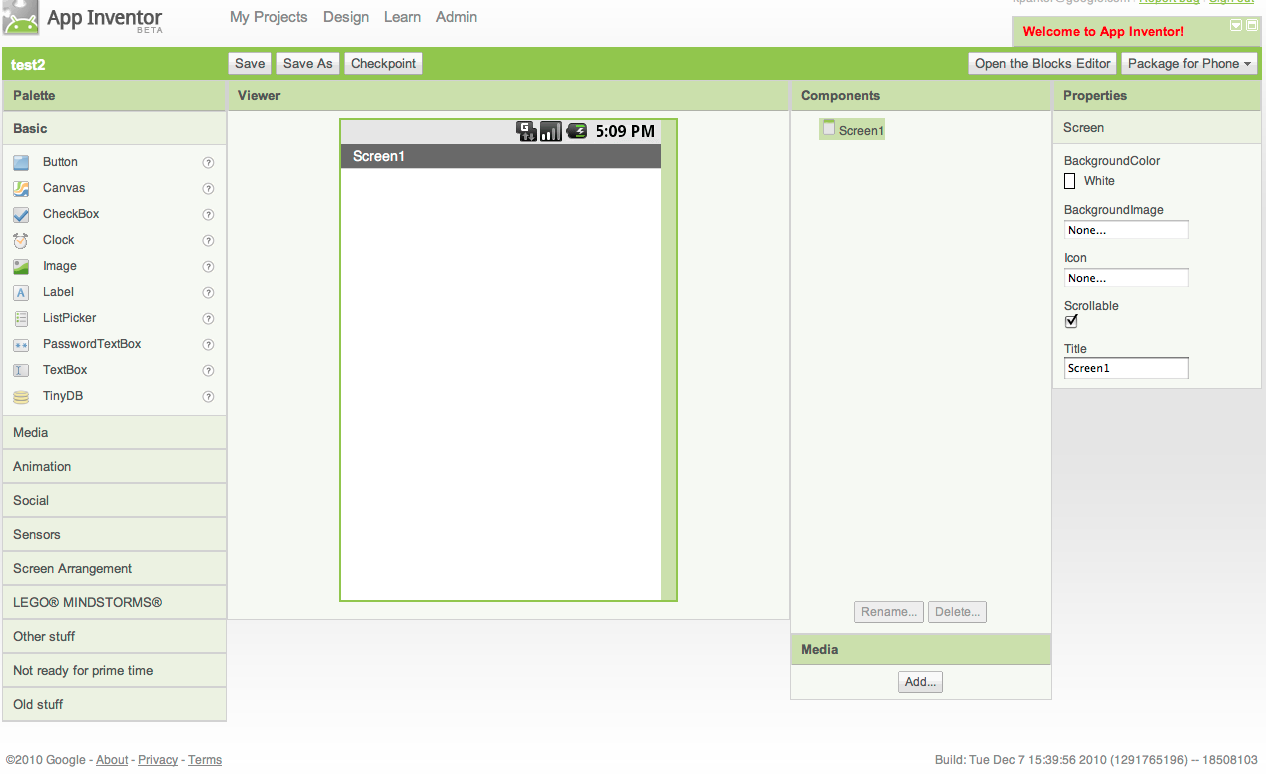
**3.2.1.1.3 Libraries**

Using libraries, the Arduino programming environment may be expanded like most programming languages. The open source libraries of Arduino facilitate the modules’ implementation and are intended to ease retrieval of data. A single function in a library provides access to retrieve data. The IDE comes with a number of libraries with some of its purpose designated below:

* **For communication**: I2C and SoftwareSerial
* **For connectivity**: GSM, Ethernet, Wifi
* **For display:** LiquidCrystal

## 3.2.2. App Inventor

App Inventor is an online web application IDE platform designed to teach various computational concepts through building Android mobile applications without any knowledge of programming. Figure 14 shows the console of the App inventor.

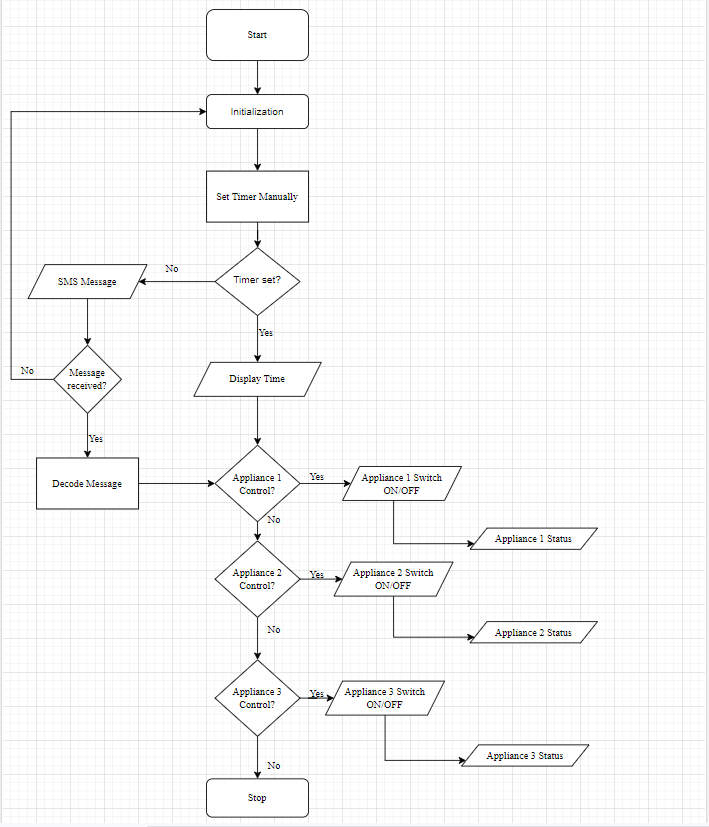


### Figure 14: App Inventor Interface

# 3.3 PROGRAM FLOWCHART

The contents of the program flowchart will be used for the firmware development of the proposed system. It features the graphical representation of the execution to understand the relation of each process.

The workflow starts with the initialization. It includes setting the required baud rate for communication and preparation of the device to use. The timer is automatically set on the time the home automation system operates with the help of the RTC module. The exact time is displayed on the LCD where we can see the countdown of the user input . If the user prefers the other method, the identified input of the system is the SMS message initiated from the mobile device and the home appliances as output. The decoded message will be used as the gateway to controlling the given home appliance. After the automatic switching, a text back message will be received by the user as the notification of the status of the appliance. The system flowchart is presented in Figure 15.



## Figure 15: System flowchart

# 3.4 TESTING OF THE PROJECT

The researchers are attempting to assess if the project is running as planned. The purpose of project testing is to ensure the project's function and quality.

* **Planning.** Discussing arrangements for how the suggested system would really function.
* **Requirement Analysis**. An examination of the necessary components that are required for the project.
* **System Design.** This includes configuration of the projects’ hardware, firmware, and communication.
* **Programming**. The writing of the code for Arduino and the phone application. The Arduino offers tools to facilitate system implementation. It has a serial connection with an Arduino microcontroller. Printing to the serial terminal in the IDE enables the tracking of hardware and firmware function events.
* **Unit Testing.** It involves testing libraries of each required hardware module of the system.
* **Integration Testing.** This test validates libraries of various modules are able to communicate with one another.
* **System Testing**. The real-time operation of the actual project. This checks how the whole application works, how it works with other parts. It checks that the project meets both functional and non-functional requirements.
* **Acceptance Testing.** Testing and evaluating the system in the user environment.
* **Project Maintenance.** Maintaining the quality of the components. Ongoing monitoring of the status of the components such as the SIM card storage capacity.

# 3.5 SCHEMATIC DIAGRAM

The schematic representation of modules, sensors, and microcontrollers enables the researchers to freely create the project. In this project, an Arduino Mega board is used for controlling the whole process. The SIM900A is used for wireless communication in controlling the home appliances. The button is set to a timer to manually turn on/off the appliances in a specific time. Arduino receives the message through GSM, then sends the signal to the relay driver. The researchers defined the commands to control the appliance for which the Arduino decodes the message from the GSM module.

In making the schematic of the project, the researchers utilized the Proteus software. It is an open-source circuit design and simulation prototype tool. It is a platform to create the wiring diagrams for the project and the supported libraries providing the visualization of hardware components.

For the schematic diagram of the system, refer to Figure 17 and 20.

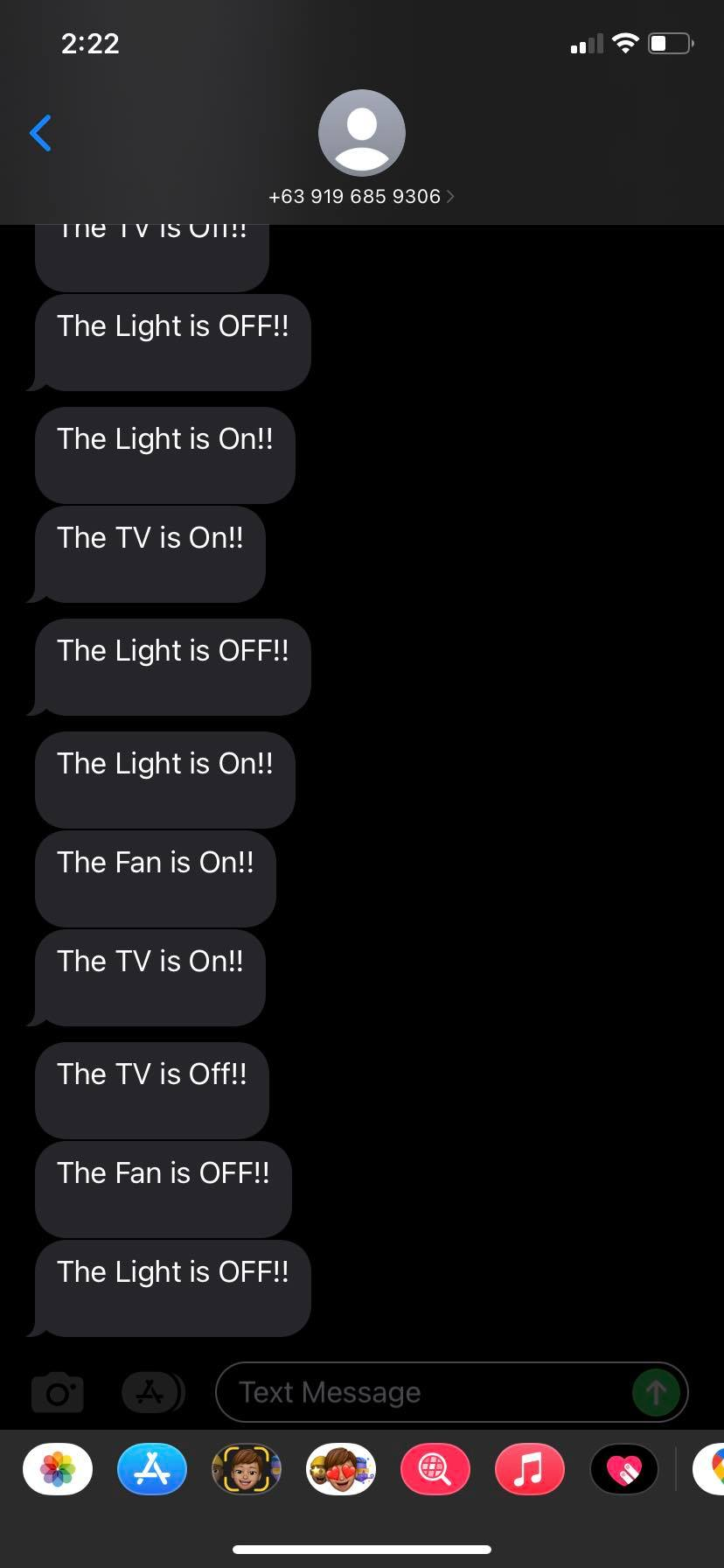
CHAPTER IV

PRESENTATION AND ANALYSIS OF DATA

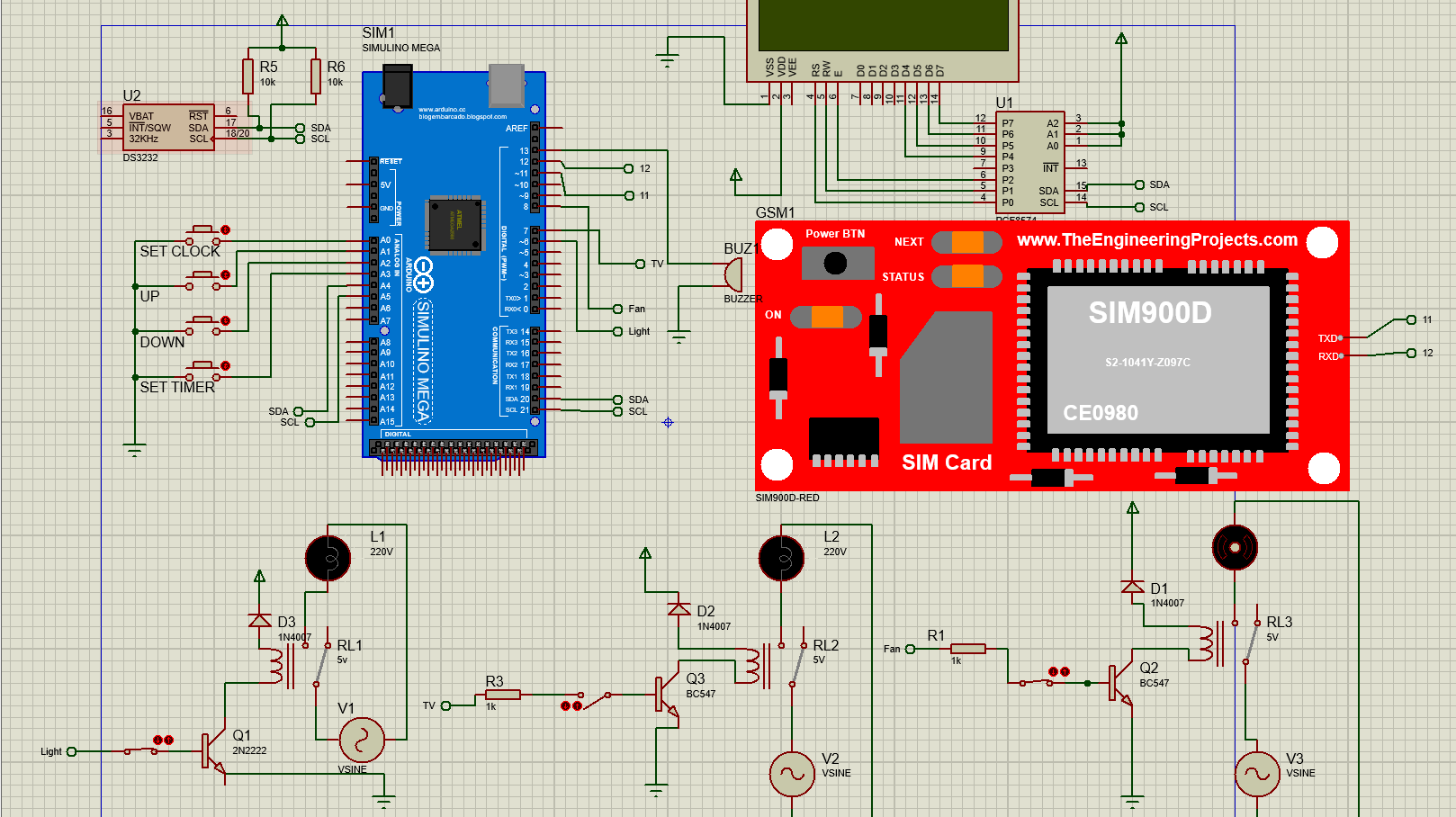
This chapter presents the data gathered during the conduct of the experiments. It includes discussion of the details such as the switching of the appliances, the controlling of switchbox using mobile phone, integrating an existing mobile application, and other suitable features of the system.

# 4.1 AUTOMATIC SWITCHING OF APPLIANCE

The researchers are able to identify the switching on and off appliances by sending an SMS that will automatically Switch the appliance. The delay of a message to transmit depends on the signal of the mobile phone or the GSM module. Afterwards, the user will receive a notification message containing the appliance status. Figure 16 presents the notification SMS of the appliance status.



### Figure 16: Feedback Message



### Figure 17: The schematic of GSM Based Home Automation System

Figure 17 presents the schematic in which the appliances are connected to the 220V power source. Pins 6, 7 and 8 from Arduino Mega are connected to the four relay channel module. The GSM device is connected to pins 11 and 12. The button is where the user sets up the timer schedule that is connected to the analog pins A0 to A3 while the RTC module is connected to the Digital Pins 20 and 21 to enable the 24 hours time clock.

# 4.2 REMOTELY CONTROLLING SWITCH BOX USING MOBILE PHONE

The researchers perform the controlling of the switchbox using a mobile phone by sending a SMS to the switchbox. The process of receiving to the switchbox starts when the GSM module executes the SMS from the mobile phone for the user to control the appliances. The SMS must contain with ‘ON#(the ‘#’ appliance number want to control)’ while the ‘OFF#(the’#’ Appliance number wants to control)’ to control the appliances the switching delay depends on the signal strength. The graphic below, Figure 18 shows the remote control phone application while the Figures 19, 20 and 21 presenting the block diagram of the GSM control appliances, the schematic diagram and the actual system switch box respectively.

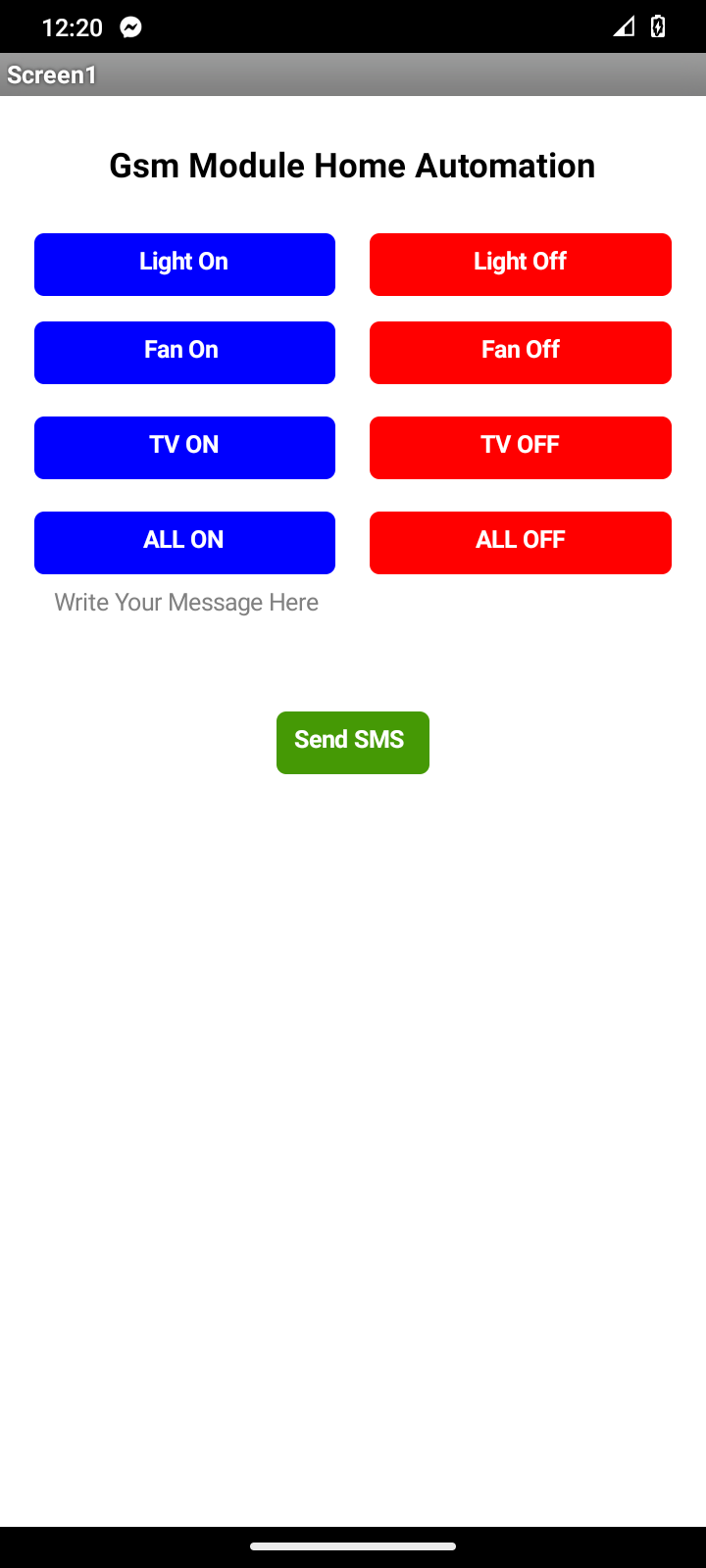
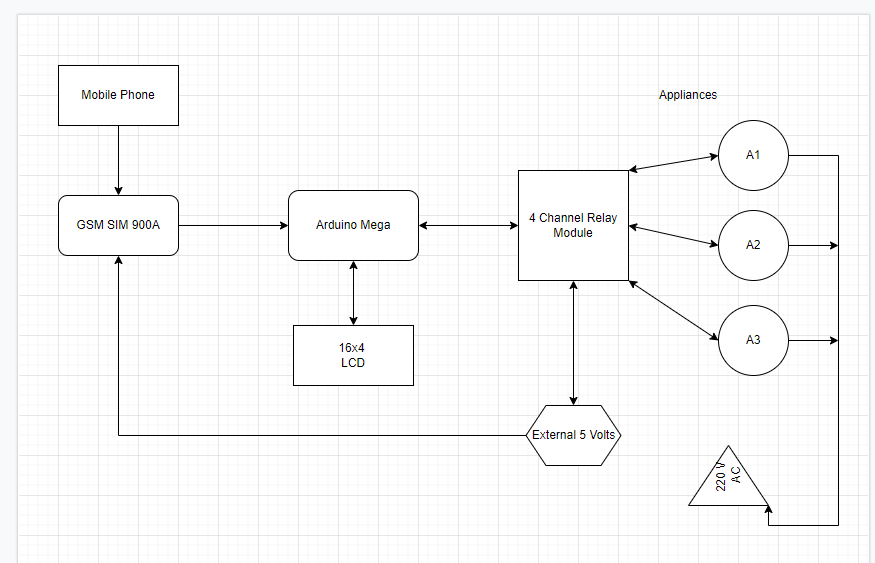
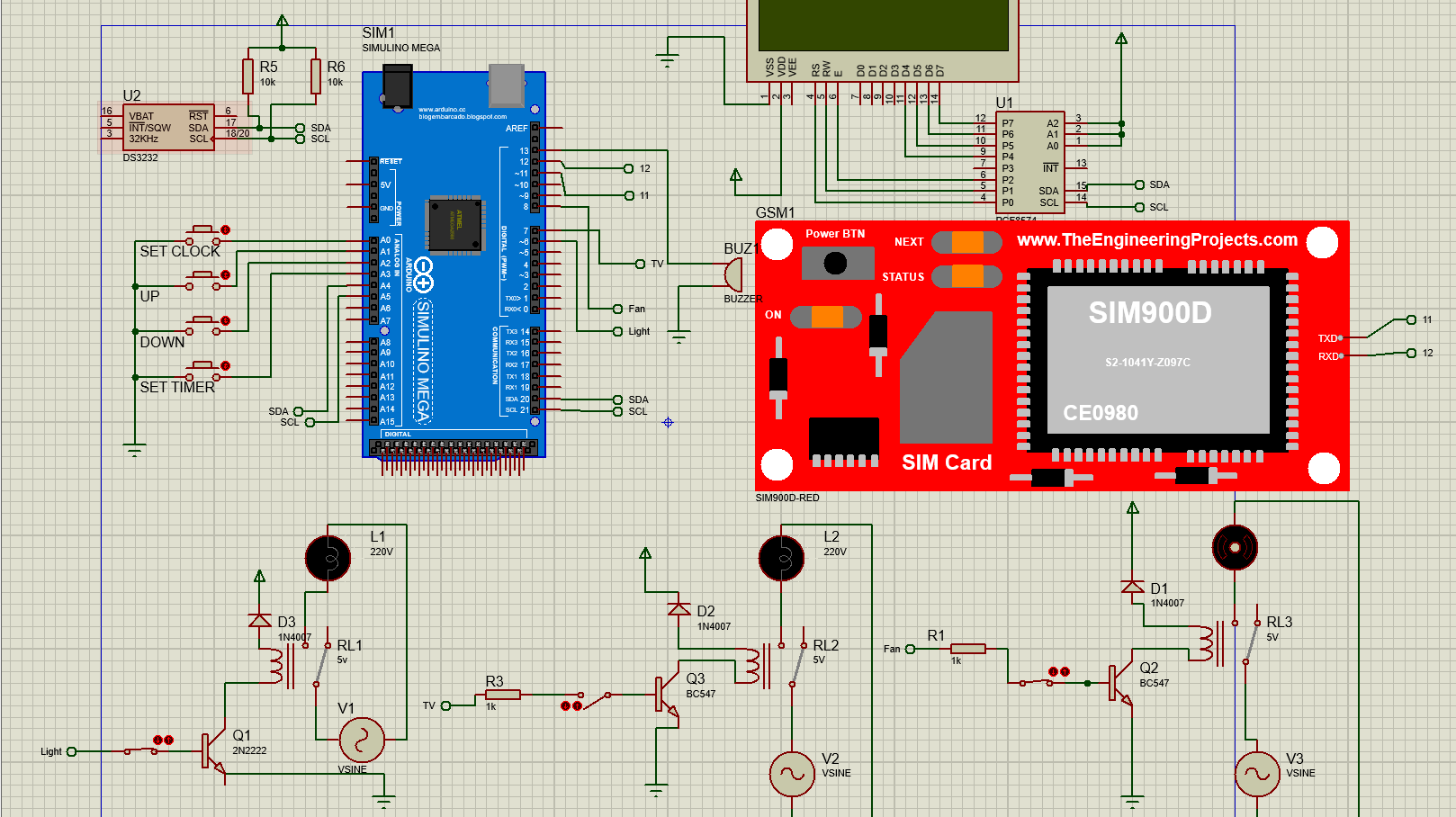


Figure 18: Remote Control



### Figure 19: Block Diagram of the GSM Control Appliances.

 Figure 20: Schematic Diagram of the GSM control appliance



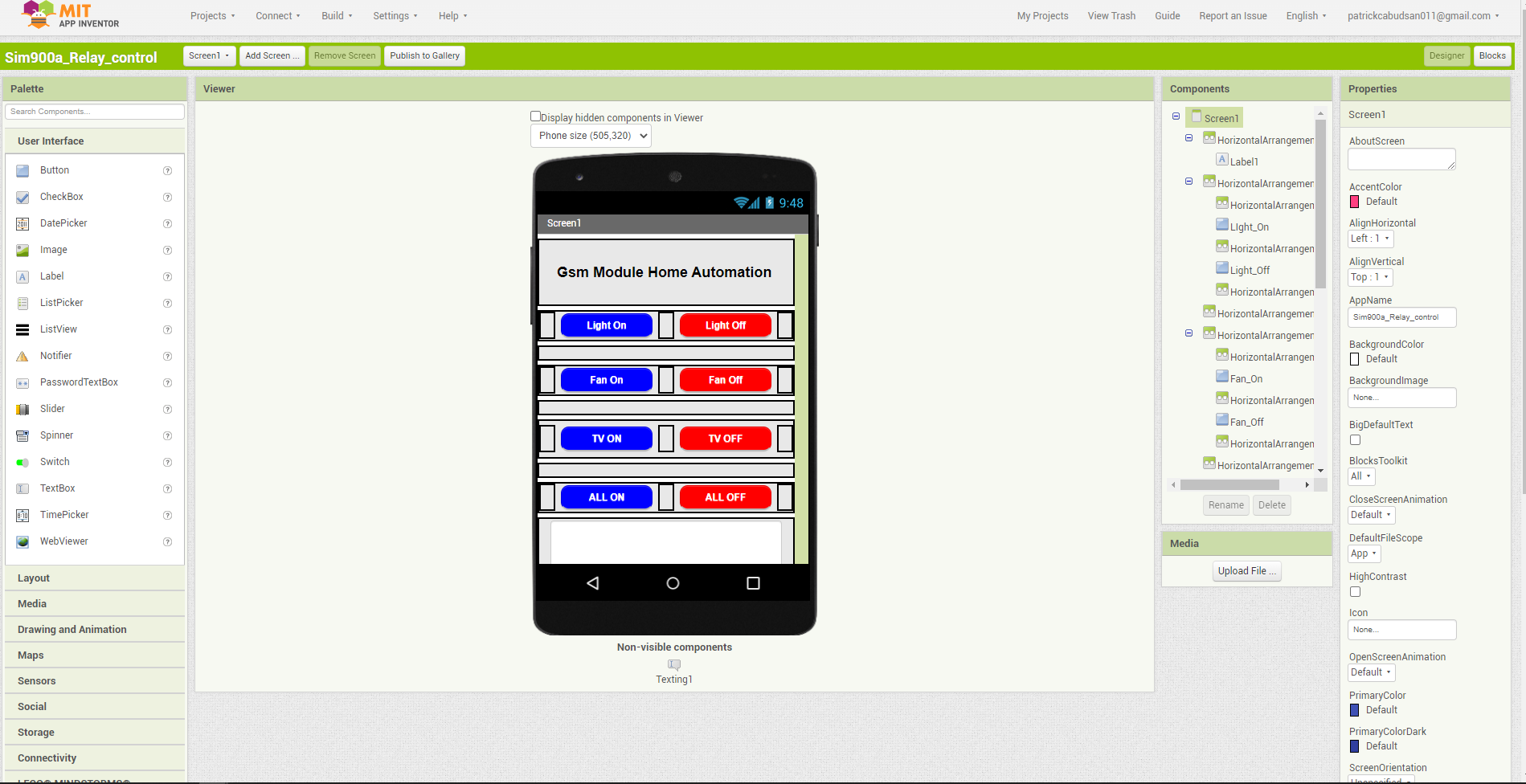
### Figure 21: Actual Switchbox

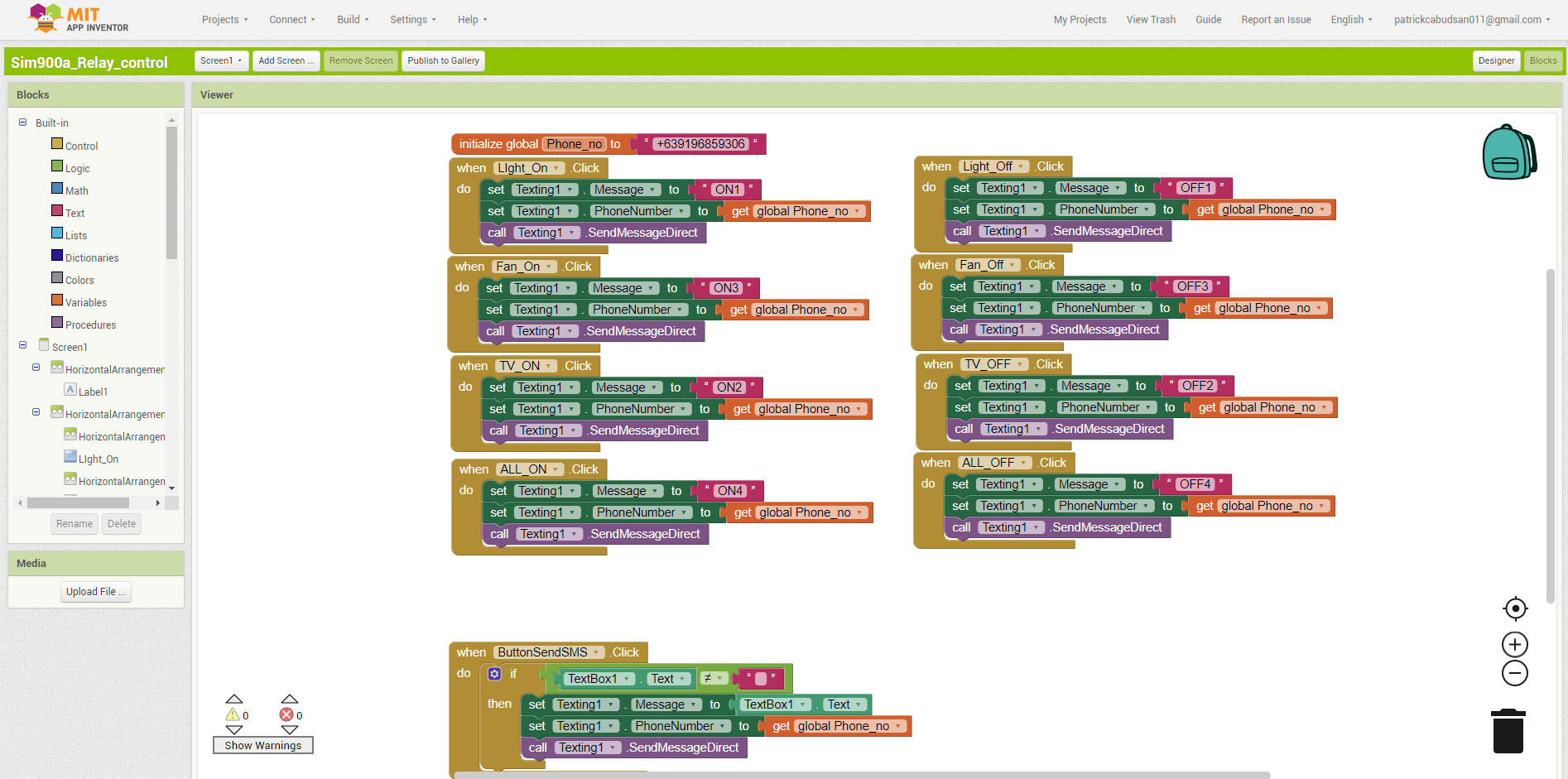
# 4.3 INTEGRATION OF EXISTING WEB APPLICATION

The web application is the App Inventor. This web application features a visual smartphone wherein the researchers design a phone application to control appliances through mobile phones. The interface of the mobile application is designed to be simple, noticeable, and friendly to the user. It comprises 4 sets of on and off remote buttons for the three home appliances and one selection for switching all appliances. For the sample application, see Figure 22.

For the construction of the phone application, the building blocks feature of the App Inventor is presented in Figure 23 for the visualization of the construction of the syntax. It includes setting up the given mobile number through the GSM module and the conditions set through each home appliance, and button functions.

The mobile application where the researchers focus in the future is Utility Mobile application, like when having a meeting to identify the weather conditions outside and inside there has a certain built-in application to be informed . This mobile application, for just a short period of time, is like a calculator we normally use in a restaurant to compute the bill of a certain order.

Figure 22: The Design of Web application



### Figure 23: The Blocks of the Web Application

# 4.4 OTHER FEATURES OF THE SYSTEM

The scheduling timer is a feature of the home automation system wherein the user will input by using the set of push button switches to set time, set clock, increment and decrement the time scheduling to automatically switch on/off the appliances in a given input of the user.

The Temperature Monitoring using RTC Module, wherein the data is retrieved from the module using a standard command from the DS3231 Library. Whenever the actual temperature is below 0 degree Celsius the RTC temperature appears to be equal to the real temperature plus 255. Figure 24 shows the interface of the scheduling timer.



### Figure 24: Scheduling timer from the LCD interface.

# 

# 4.5 USER EVALUATION

Evaluation is the process of closely analyzing the project by collecting and evaluating information about its activities, features, and outcomes in order to test the acceptance of the product as well as make judgments and improve the program's quality. This section focuses on user evaluations of the product. The researchers collect this information using a variety of home automation system approaches. Here, user evaluation is conducted by selecting a sample of 10 individuals and requesting that they employ the system and provide feedback using an online questionnaire. The researchers utilized Google Forms application for the evaluation creation and collection. Refer to Appendix A for the User Evaluation Questionnaire.

## 4.5.1 User Evaluation Results

After gathering all of the information obtained from the questionnaire, the following is an overview of the pertinent data:

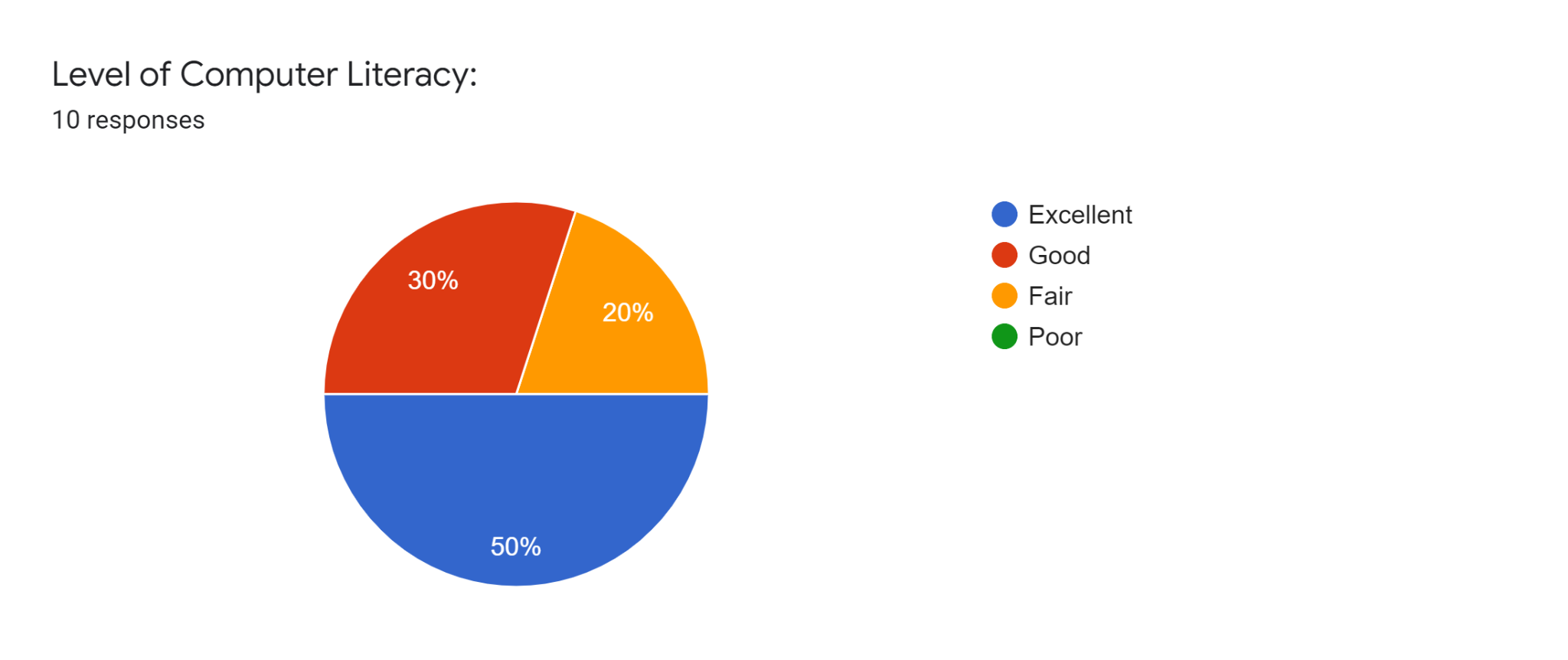


Figure 25: Results about Computer Literacy

Figure 25 shows a graphic summary of the data about how much individuals agreed on the level of computer literacy. In the results accumulated, 50% of the respondents are excellent in computer literacy. Meanwhile, 30% are good and the rest have fair knowledge about computers.

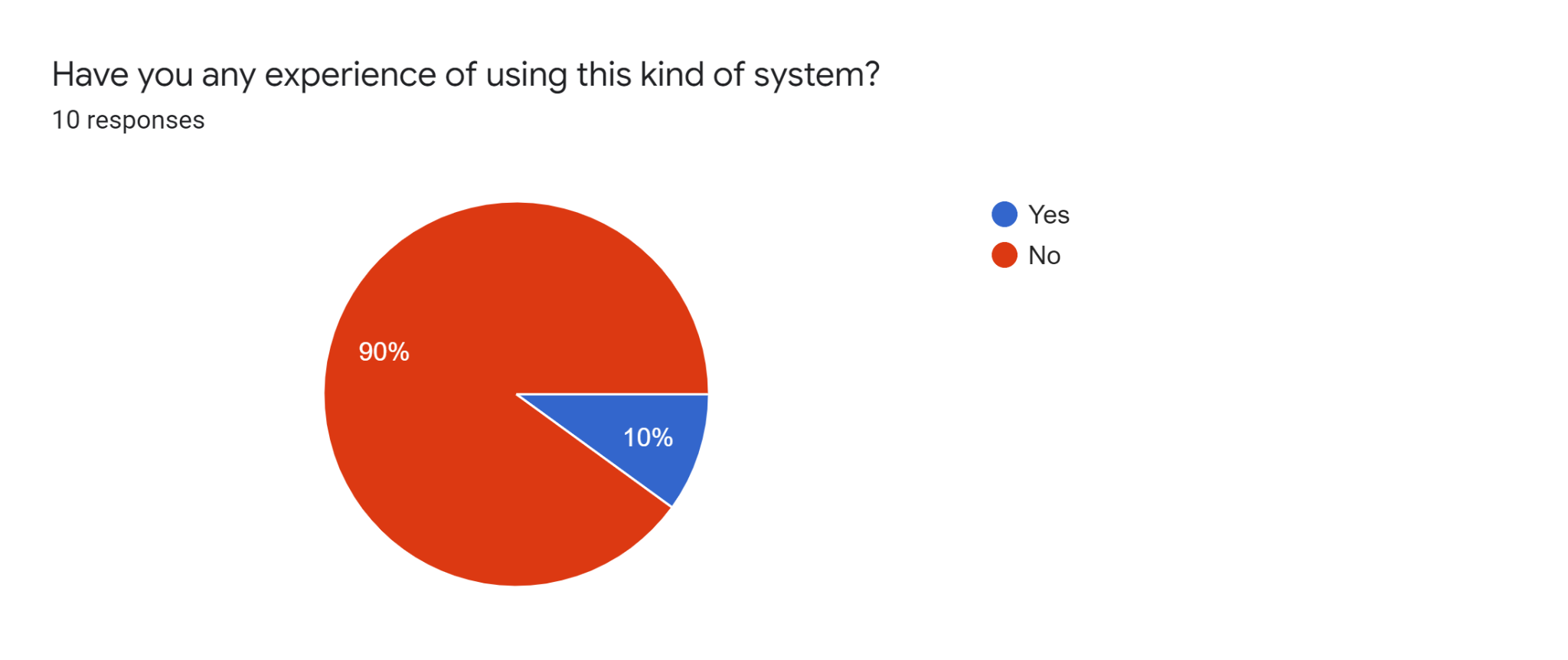


Figure 26: Results about Awareness of the System

Figure 26 shows a graphic summary of the data about how individuals responded to their awareness of using a home automation system. According to the acquired data, nine out of ten of the respondents do not have any prior experience utilizing a system like this.

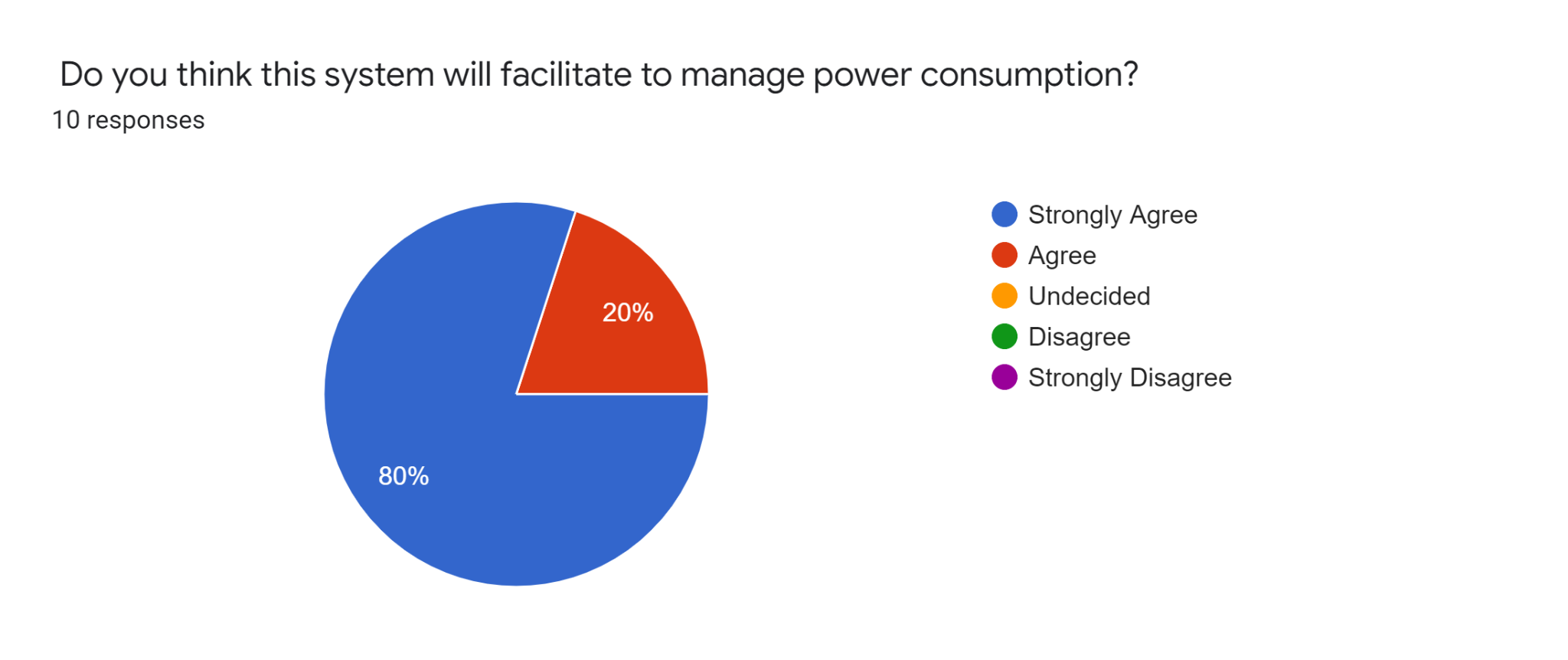


Figure 27: Results about Power Consumption

Figure 27 shows a graphic summary of the data about how individuals agreed that the system can facilitate in managing their power consumption. As per the accumulated results, 80% of the respondents are in complete agreement that the system assists in the management of power usage.

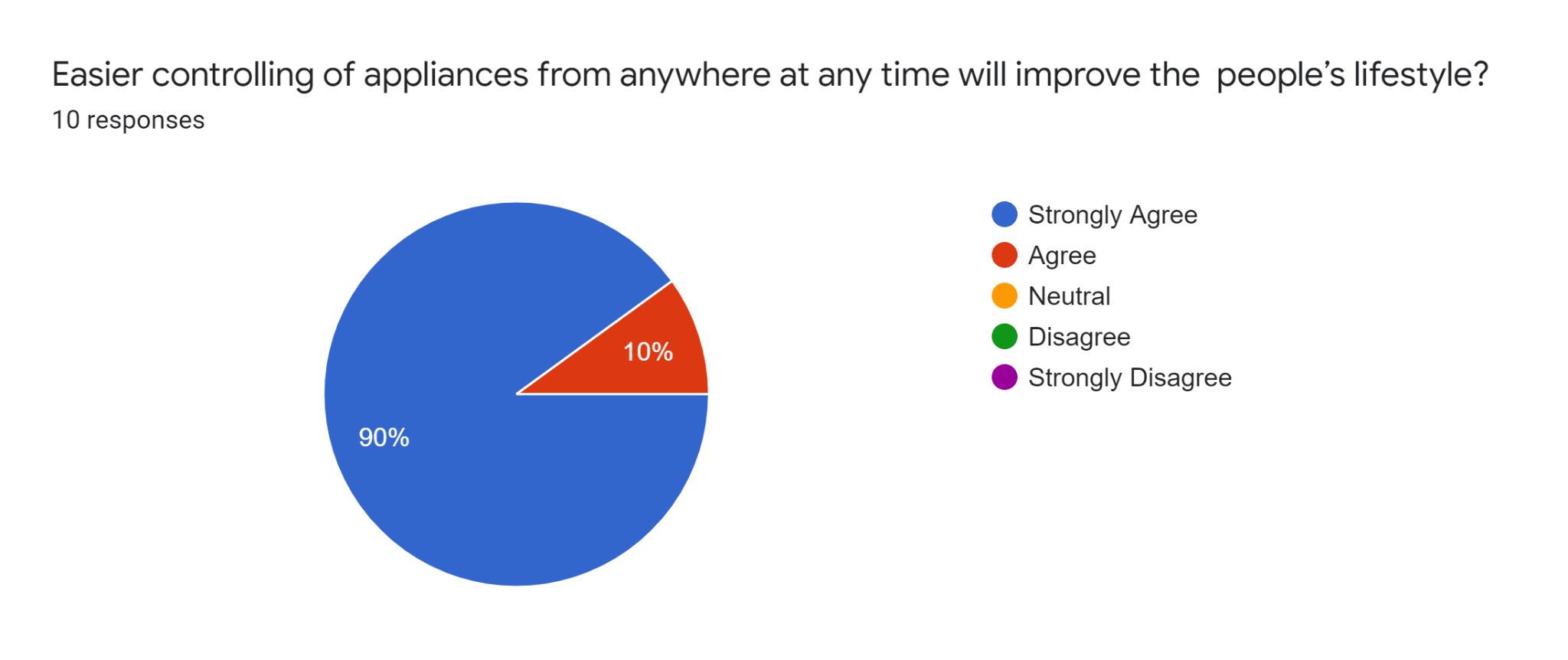


Figure 28: Results about People’s Lifestyle

Figure 28 shows a graphic summary of the data about how individuals agreed the ease of controlling appliances away from home at any given time would improve their lifestyle. Based on the outcome, 90% of the people who answered strongly agreed that being able to control their appliances anywhere can make their lives better.

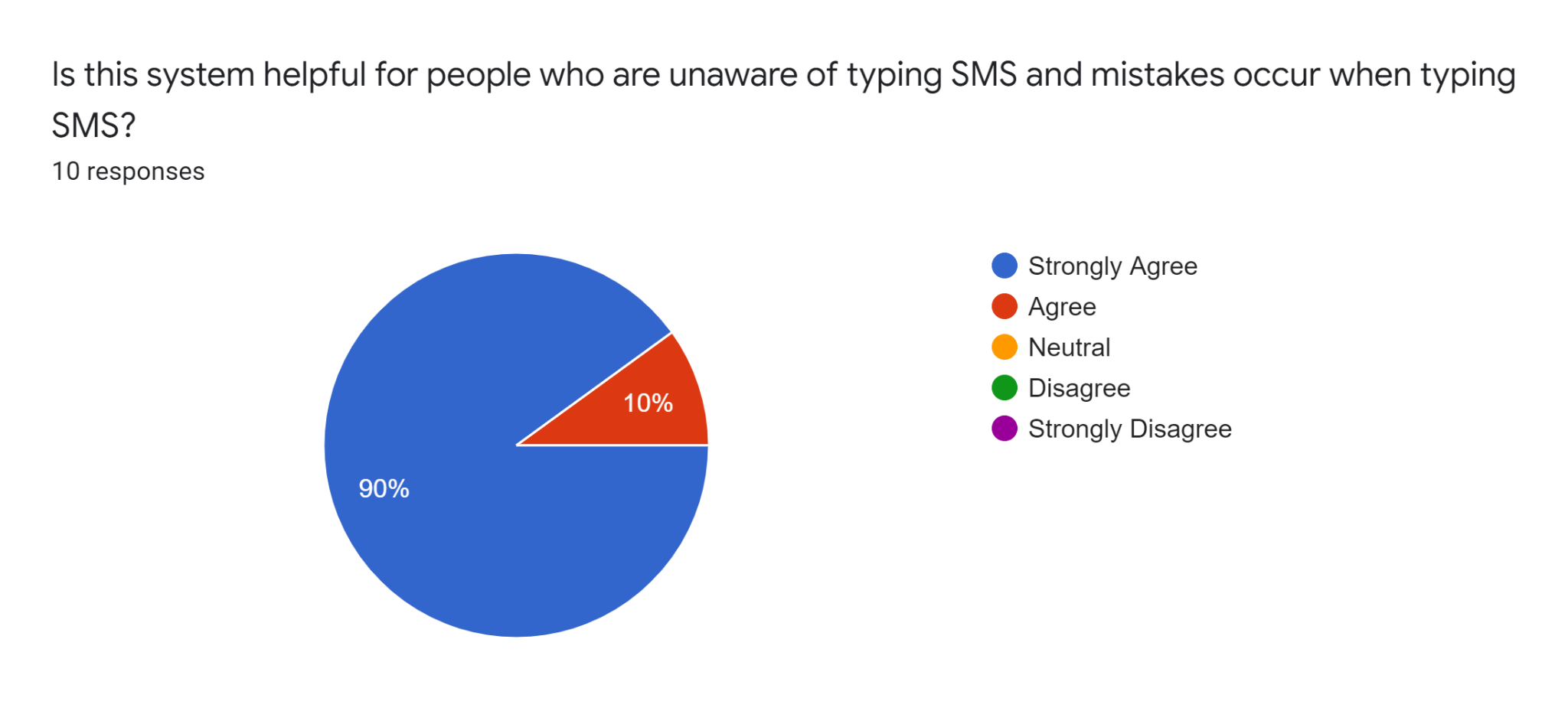


Figure 29: Results about SMS Awareness

Figure 29 shows a graphic summary of the data about how individuals agreed that the system is helpful for the people with less awareness about SMS and those who make mistakes while using SMS. According to data that have been recorded, 90% of everyone who answered concurred that the system is helpful to persons who are not acquainted with SMS.

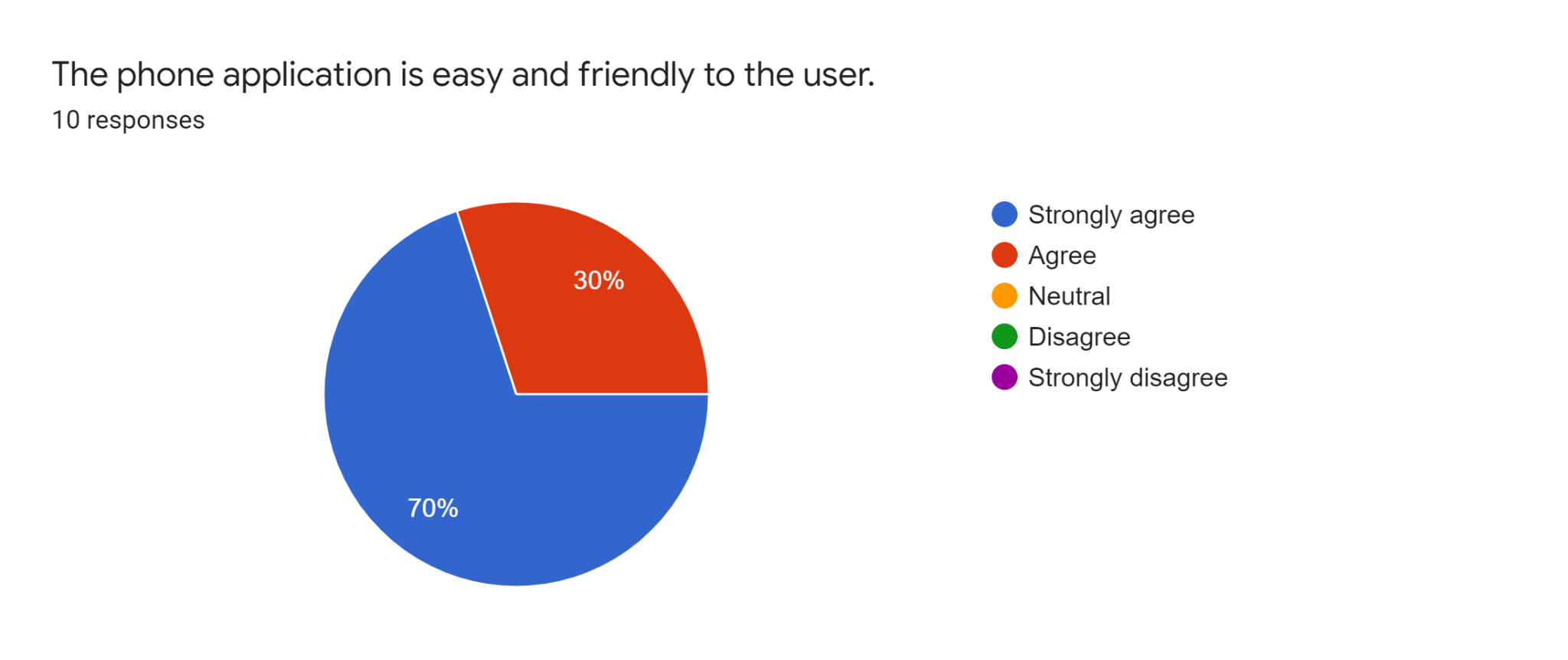


Figure 30: Results about the Phone Application

Figure 30 shows a graphic summary of the data about how individuals agreed that the phone application is easy and user-friendly. In which, 70% of the respondents strongly agreed while 30% have agreed in response to their phone application experience.

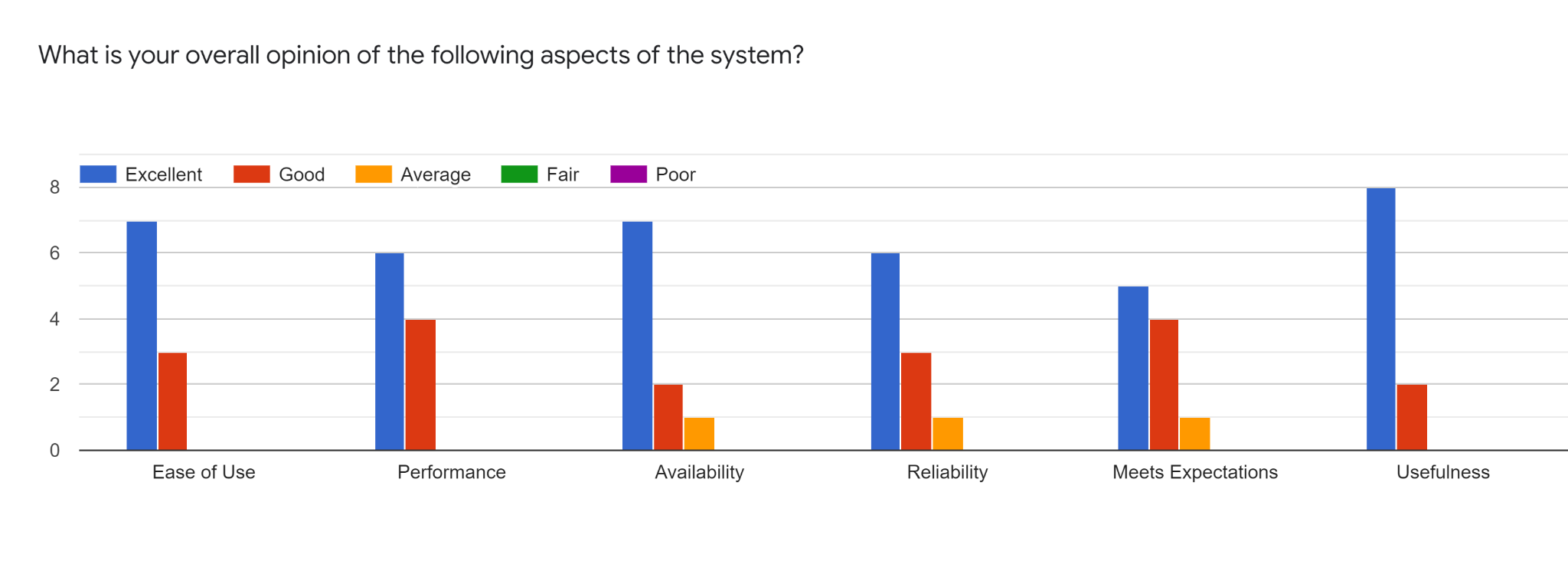


Figure 31: Results about System Aspects

Figure 31 shows a graphic summary of the data about how individuals agreed the overall opinion of the different aspects of the system. The questionnaire consists of six aspects as follows: Ease of Use, Performance, Availability, Reliability, Meets Experience, and Usefulness. Majority of the respondents have strongly agreed with the system characteristics abovementioned 80% for the Usefulness, 70% for the Availability and Ease of Use, 60% for Reliability and 50% for Meets Expectations.

CHAPTER V

SUMMARY, CONCLUSION AND RECOMMENDATION

This chapter presents the summary, conclusions, and recommendations of the project.

# 5.1 SUMMARY

Through the phase of planning and building the system the researchers have obtain the following:

* Learned how to test programs to control various appliances.
* Learned how to operate with SMS technology.
* Learned how to make a remote-control phone application.
* Make additional features for the system.
* Enhanced knowledge in electronic components.
* Improved writing skills and learning methods.
* Learn the process of proper documentation methods while writing this thesis.

# 5.2 CONCLUSION

The GSM-Based Home Automation Using Arduino can be deemed a success. This system includes an Arduino Mega board, GSM Module, a mobile device, electrical and other electronic components, and home appliances. It is both user-friendly, accessible, and economical. Furthermore, it can be determined that this project's requirements are fulfilled, and they are as follows: Built a home automation system that will automatically switch (on & off) an appliance. Designed a switch box that can be controlled remotely using a mobile device. Integrated an existing mobile application for home appliance monitoring, and designed other features suitable for the automation system.

# 5.3 RECOMMENDATIONS

Listed below are the areas in which the researchers recommended making improvements to the system:

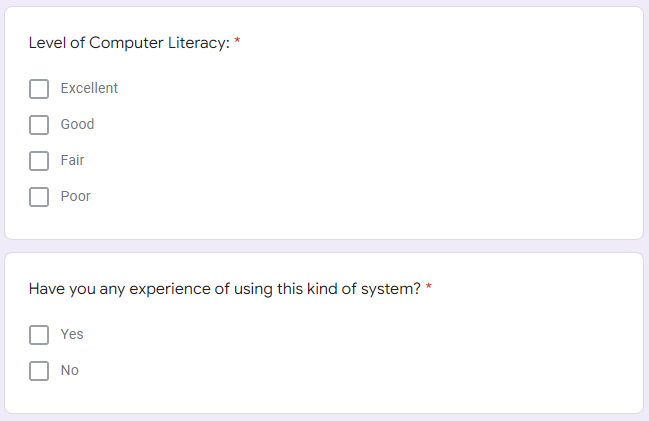
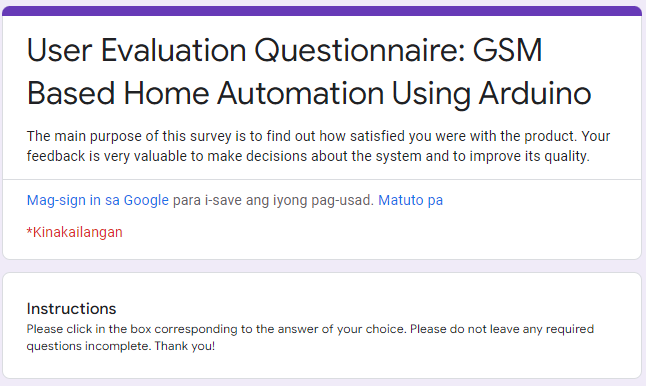
* This system is limited to functionalities such as switching ON and OFF appliances and does not support extended controls such as speed of the fan and the brightness of lights.
* Building a utility mobile application for home automation is a research topic to consider.
* Security measures to minimize infiltrating the system to improve home automation systems.
* The design of the switchbox can be further improved with additional materials that support the strength and compatibility of the system.

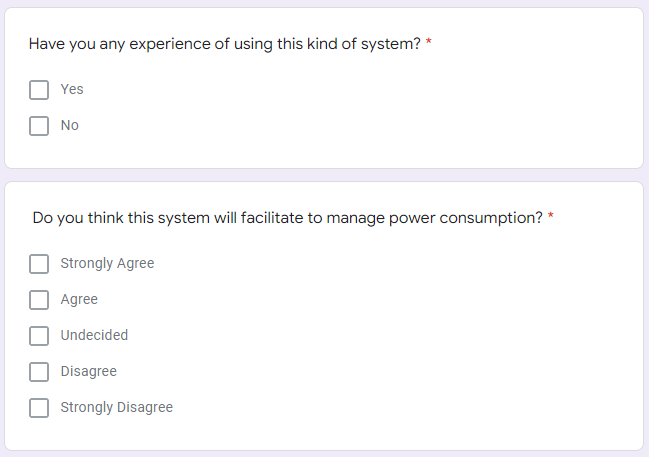
REFERENCES

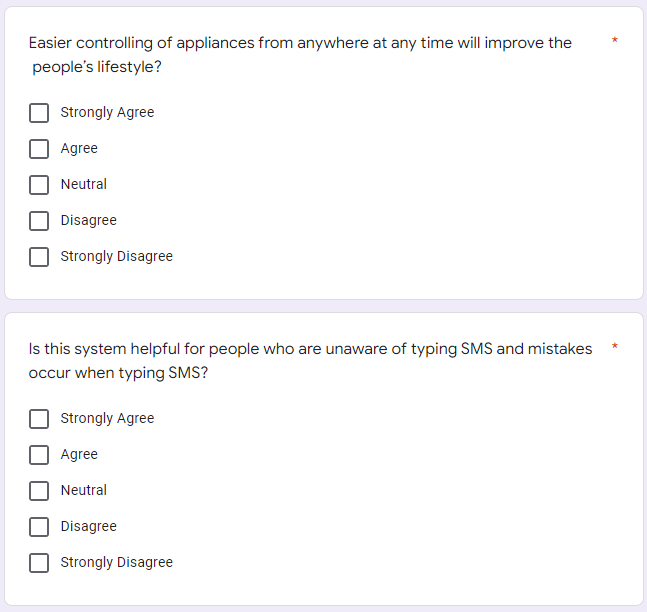
1. Abdul Raheem, A. K. K. (2017). Bluetooth Based Smart Home Automation System using Arduino UNO Microcontroller. Al-Mansour Journal, 27, 139–156.
2. Dubey, A. (2012). Design & Development of ARM7TDMI-S based GSM Mobile for Home Automation & Security. E-Journal of Science & Technology, 7(4), 67–74.
3. Gunawan, T. S., Yaldi, I. R. H., Kartiwi, M., & Mansor, H. (2018). Performance Evaluation of Smart Home System using Internet of Things. International Journal of Electrical & Computer Engineering (2088-8708), 8(1), 400–411. https://doi.org/10.11591/ijece.v8i1.pp400-411
4. Henry, O. (2013). CONSTRUCTION OF A GSM BASED HOME AUTOMATION USING AVR [Review of CONSTRUCTION OF A GSM BASED HOME AUTOMATION USING AVR]. pub.abuad.edu.ng
5. Hussain, Y. M., Hanoosh, H. O., Zakaria, Z., Al-Dhief, F. T., Saare, M. A., Jawad, M. M., Omran, A. H., & Abdulbari, A. A. (2021). Smartphone’s off grid communication network by using Arduino microcontroller and microstrip antenna. Telkomnika, 19(4), 1100–1106. https://doi.org/10.12928/TELKOMNIKA.v19i4.15949
6. Isa, E., & Sklavos, N. (2017). Smart Home Automation: GSM Security System Design & Implementation. Journal of Engineering Science & Technology Review, 10(3), 170–174. https://doi.org/10.25103/jestr.103.22
7. Janani, K., Harini, S., Abirami, B., & Vijayalakshmi, V. J. (2021). IOT based Home Automation using Cloud. *Grenze International Journal of Engineering & Technology (GIJET)*, *7*(1), 927-933. Computers & Applied Sciences Complete.
8. Jivani, M. (2014). GSM Based Home Automation System Using App-Inventor for Android Mobile Phone. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 3. 12121-12128. 10.15662/ijareeie.2014.0309042.
9. Karkera, T., Dubey, A., Kamalnakhawa, S., & Mangale, S. (2018). GPS-GSM based Vehicle Tracking System. *International Journal of New Technology and Research (IJNTR)*, *4*(3), 140–142. Retrieved 2022, from https://media.neliti.com/media/publications/263098-gps-gsm-based-vehicle-tracking-system-5b708ef1.pdf.
10. Kerem, A. and Yazgan, A. (2021), "Design and prototyping of GSM-bluetooth based solar energy remote monitoring system", COMPEL - The international journal for computation and mathematics in electrical and electronic engineering, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/COMPEL-10-2021-0384
11. Malik, N., & Bodwade, Y. (2017). Literature Review on Home Automation System. International Journal of Advanced Research in Computer and Communication Engineering, 6, 733-737.
12. Mohd Annuar, K. A., Maharam, M. F., Ab Hadi, N. A., Harun, M. H., & Ab Halim, M. F. M. (2019). Development of wireless and intelligent home automation system. Telkomnika, 17(1), 32–38. https://doi.org/10.12928/TELKOMNIKA.v17i1.9075
13. Monowar, M. I., Kafi, A. H., Shakil, S. R., & Rhaman, M. K. (2019). Framework of an Intelligent, Multi Nodal and Secured RF Based Wireless Home Automation System for Multifunctional Devices. *Wireless Personal Communications*, *105*(1), 1–16. https://doi.org/10.1007/s11277-018-6100-z
14. Rohith Sai, V. (2019). GSM based Home Automation System. Grenze International Journal of Engineering & Technology (GIJET), 5(2), 177–182.
15. Singh, P., Chotalia, K., Pingale, S., & Kadam, S. (2016). A Review Paper on Smart GSM Based Home Automation System.
16. Sowah, R. A., Boahene, D. E., Owoh, D. C., Addo, R., Mills, G. A., Owusu-Banahene, W., Buah, G., & Sarkodie-Mensah, B. (2020). Design of a Secure Wireless Home Automation System with an Open Home Automation Bus (OpenHAB 2) Framework. Journal of Sensors, 1–22. https://doi.org/10.1155/2020/8868602
17. Tirian, G.-O. (2017). Smart Home Automation with Arduino. Annals of the University of Petrosani Electrical Engineering, 19, 5–14.

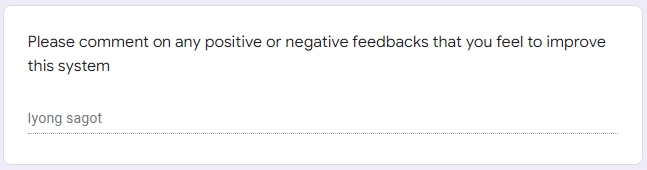
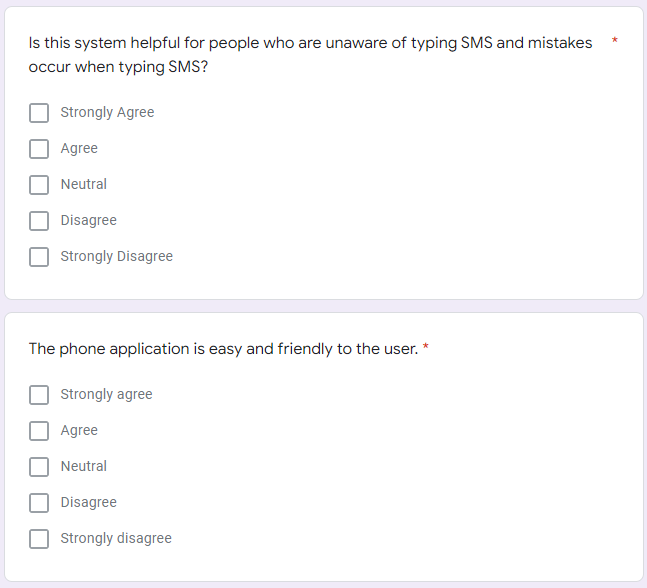
APPENDIX

# Appendix A: User Evaluation Questionnaire Sample









# APPENDIX

# Appendix B: Arduino IDE Source Code

/\*SIM800L< ------ >ARDUINO Board

Vcc < ------ > 5V

Gnd < ----- > Gnd

Rx < ------- > Pin no. 10 (SoftwareSerialTx)

Tx < ------- > Pin no. 9 (SoftwareSerialRx)

\*/

#include <DS3231.h>

#include <EEPROM.h>

#include <Wire.h>

#include <SoftwareSerial.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,20,4); //0x27 is the i2c address, while 16 = columns, and 2 = rows.

SoftwareSerial mySerial(9,10);

DS3231 rtc(SDA, SCL);

Time t;

#define bt\_clock A0

#define bt\_up A1

#define bt\_down A2

#define bt\_timer A3

#define Light 7

#define Fan 8

#define TV 11

const int OnHour = 12;

const int OnMin = 24;

const int OffHour = 12;

const int OffMin = 25;

#define buzzer 13

int hh = 0, mm = 0, ss = 0, dd = 0, bb = 0, set\_day;

int yy = 0;

String Day = " ";

int StartHH = 0, StartMM = 0, FinishHH = 0, FinishMM = 0, setMode = 0, setAlarm = 0, alarmMode=1;

int Start1HH, Start1MM, Finish1HH, Finish1MM;

int Start2HH, Start2MM, Finish2HH, Finish2MM;

int Start3HH, Start3MM, Finish3HH, Finish3MM;

int Start4HH, Start4MM, Finish4HH, Finish4MM;

int timer1, timer2, timer3, timer4;

int stop =0, mode=0, flag=0;

char incomingByte;

String incomingData;

bool atCommand = true;

int index = 0;

String number = "";

String message = "";

void setup()

{

lcd.init(); //Init the LCD

lcd.backlight(); //Activate backlight

rtc.begin();

Serial.begin(9600);

mySerial.begin(9600);

pinMode(bt\_clock, INPUT\_PULLUP);

pinMode(bt\_up, INPUT\_PULLUP);

pinMode(bt\_down, INPUT\_PULLUP);

pinMode(bt\_timer, INPUT\_PULLUP);

pinMode(Light, OUTPUT);

pinMode(TV, OUTPUT);

pinMode(Fan, OUTPUT);

pinMode(buzzer, OUTPUT);

lcd.setCursor(0,0);

lcd.print("GSM Based");

lcd.setCursor(0,1);

lcd.print("Home Automation");

delay(2000);

lcd.clear();

// Check if you're currently connected to SIM800L

while(!mySerial.available()){

mySerial.println("AT");

delay(1000);

Serial.println("connecting....");

}

Serial.println("Connected..");

mySerial.println("AT+CMGF=1"); //Set SMS Text Mode

delay(1000);

mySerial.println("AT+CNMI=1,2,0,0,0"); //procedure, how to receive messages from the network

delay(1000);

mySerial.println("AT+CMGL=\"REC UNREAD\""); // Read unread messages

Serial.println("Ready to received Commands..");

lcd.print("System Ready!");

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Fan TV Light");

lcd.setCursor(0,1);

lcd.print("OFF OFF OFF");

delay(2000);

lcd.clear();

stop=EEPROM.read(50);

if(stop==0){

}else{

mode=1;WriteEeprom ();delay (20);

mode=2;WriteEeprom ();delay (20);

mode=3;WriteEeprom ();delay (20);

mode=4;WriteEeprom ();delay (20);

mode=0;

}

EEPROM.write(50,0);

ReadEeprom();

}

void loop()

{

lcd.setCursor(0,0);

lcd.print("Fan TV Light");

if (mySerial.available()){

delay(100);

// Serial buffer

while(mySerial.available()){

incomingByte = mySerial.read();

incomingData += incomingByte;

}

delay(10);

if(atCommand == false){

receivedMessage(incomingData);

}else{

atCommand = false;

}

//delete messages to save memory

if (incomingData.indexOf("OK") == -1){

mySerial.println("AT+CMGDA=\"DEL ALL\"");

delay(1000);

atCommand = true;

}

incomingData = "";

}

t = rtc.getTime();

Day = rtc.getDOWStr(1);

if (setMode == 0){

hh = t.hour,DEC;

mm = t.min,DEC;

ss = t.sec,DEC;

dd = t.date,DEC;

bb = t.mon,DEC;

yy = t.year,DEC;

//if(t.hour>12){hh=t.hour-12;}// for removing 24 hours

//else{hh=t.hour;}

}

if(setAlarm==0){

lcd.setCursor(0,2);

lcd.print((hh/10)%10);

lcd.print(hh % 10);

lcd.print(":");

lcd.print((mm/10)%10);

lcd.print(mm % 10);

lcd.print(":");

lcd.print((ss/10)%10);

lcd.print(ss % 10);

lcd.print(" T:");

lcd.print(rtc.getTemp(),0);

lcd.write(223);

lcd.print("C");

lcd.print(" ");

lcd.setCursor(1,3);

lcd.print(Day);

lcd.print(" ");

lcd.print((dd/10)%10);

lcd.print(dd % 10);

lcd.print("/");

lcd.print((bb/10)%10);

lcd.print(bb % 10);

lcd.print("/");

lcd.print((yy/1000)%10);

lcd.print((yy/100)%10);

lcd.print((yy/10)%10);

lcd.print(yy % 10);

}

setupClock();

setTimer();

delay (100);

blinking();

//Timer1 ON

if (timer1==1 && alarmMode==1 && hh==Start1HH && mm==Start1MM) {

digitalWrite(Light, HIGH);

lcd.setCursor(11,1);

lcd.print("ON");

}

//Timer1 OFF

if (timer1==1 && alarmMode==1 && hh==Finish1HH && mm==Finish1MM){

digitalWrite(Light, LOW);

lcd.setCursor(11,1);

lcd.print("OFF");

lcd.clear();

}

//Timer2 ON

if (timer2==1 && alarmMode==1 && hh==Start2HH && mm==Start2MM) {

digitalWrite(Fan, HIGH);

lcd.setCursor(0,1);

lcd.print("ON");

}

//Timer2 OFF

if (timer2==1 && alarmMode==1 && hh==Finish2HH && mm==Finish2MM){

digitalWrite(Fan, LOW);

lcd.setCursor(0,1);

lcd.print("OFF");

lcd.clear();

}

//Timer3 ON

if (timer3==1 && alarmMode==1 && hh==Start3HH && mm==Start3MM) {

digitalWrite(TV, HIGH);

lcd.setCursor(6,1);

lcd.print("ON");

}

//Timer3 OFF

if (timer3==1 && alarmMode==1 && hh==Finish3HH && mm==Finish3MM){

digitalWrite(TV, LOW);

lcd.setCursor(6,1);

lcd.print("OFF");

lcd.clear();

}

if (timer4==1 && alarmMode==1 && hh==Start4HH && mm==Start4MM) {

digitalWrite(Light, HIGH);

digitalWrite(Fan, HIGH);

digitalWrite(TV, HIGH);

lcd.setCursor(0,1);

lcd.print("ON ON ON");

}

//Timer4 OFF

if (timer4==1 && alarmMode==1 && hh==Finish4HH && mm==Finish4MM){

digitalWrite(Light, HIGH);

digitalWrite(Fan, HIGH);

digitalWrite(TV, HIGH);

lcd.setCursor(0,1);

lcd.print("OFF OFF OFF");

lcd.clear();

}

delay (100);

digitalWrite(buzzer, LOW);

}

void receivedMessage(String inputString){

//Get The number of the sender

index = inputString.indexOf('"')+1;

inputString = inputString.substring(index);

index = inputString.indexOf('"');

number = inputString.substring(0,index);

Serial.println("Number: " + number);

//Get The Message of the sender

index = inputString.indexOf("\n")+1;

message = inputString.substring(index);

message.trim();

Serial.println("Message: " + message);

message.toUpperCase(); // uppercase the message received

//turn LED ON or OFF

if (message.indexOf("ON1") > -1){

digitalWrite(Light, HIGH);

lcd.setCursor(13,1);

lcd.print("ON ");

delay(200);

Serial.println("Command: Light Turned On.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The Light is On!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

if (message.indexOf("OFF1") > -1){

digitalWrite(Light, LOW);

lcd.setCursor(13,1);

lcd.print("OFF ");

delay(200);

Serial.println("Command: Light Turned Off.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The Light is OFF!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

delay(50);

if (message.indexOf("ON2") > -1){

digitalWrite(Fan, HIGH);

lcd.setCursor(0,1);

lcd.print("ON ");

delay(200);

Serial.println("Command: Fan Turn On.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The Fan is On!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

if (message.indexOf("OFF2") > -1){

digitalWrite(Fan, LOW);

lcd.setCursor(0,1);

lcd.print("OFF ");

delay(200);

Serial.println("Command: Fan Turned Off.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The Fan is Off!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

delay(50);

if (message.indexOf("ON3") > -1){

digitalWrite(TV, HIGH);

lcd.setCursor(7,1);

lcd.print("ON ");

delay(200);

Serial.println("Command: TV Turned On.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The TV is On!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

if (message.indexOf("OFF3") > -1){

digitalWrite(TV, LOW);

lcd.setCursor(7,1);

lcd.print("OFF ");

delay(200);

Serial.println("Command: TV Turned Off.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("The TV is OFF!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

delay(50);

if (message.indexOf("ON4") > -1){

digitalWrite(TV, HIGH);

digitalWrite(Fan, HIGH);

digitalWrite(Light, HIGH);

lcd.setCursor(0,1);

lcd.print("ON ON ON");

delay(200);

Serial.println("Command: Turned On All.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("All Appliances are On!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

if (message.indexOf("OFF4") > -1){

digitalWrite(TV, LOW);

digitalWrite(Fan, LOW);

digitalWrite(Light, LOW);

lcd.setCursor(0,1);

lcd.print("OFF OFF OFF");

delay(200);

Serial.println("Command: Turned Off All.");

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+639761656784\"\r");

delay(1000);

mySerial.println("All Appliances are OFF!!");

delay(100);

mySerial.println((char)26);

delay(1000);

}

delay(50);

}

void blinking (){

//BLINKING SCREEN

//Set Clock

if (setAlarm <2 && setMode == 1){lcd.setCursor(0,2); lcd.print(" ");} //0,1

if (setAlarm <2 && setMode == 2){lcd.setCursor(3,2); lcd.print(" ");} //3,0

if (setAlarm <2 && setMode == 3){lcd.setCursor(6,2); lcd.print(" ");} //6,0

if (setAlarm <2 && setMode == 4){lcd.setCursor(1,3); lcd.print(" ");} //1,1

if (setAlarm <2 && setMode == 5){lcd.setCursor(5,3); lcd.print(" ");} //5,1

if (setAlarm <2 && setMode == 6){lcd.setCursor(8,3); lcd.print(" ");} //8,1

if (setAlarm <2 && setMode == 7){lcd.setCursor(11,3); lcd.print(" "); }//11,1

//Set Timer

if (setMode == 0 && setAlarm == 1 && mode==0){lcd.setCursor(2,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 2 && mode==0){lcd.setCursor(6,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 3 && mode==0){lcd.setCursor(10,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 4 && mode==0){lcd.setCursor(13,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 1 && mode>0){lcd.setCursor(11,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 2 && mode>0){lcd.setCursor(14,2); lcd.print(" "); }

if (setMode == 0 && setAlarm == 3 && mode>0){lcd.setCursor(11,3); lcd.print(" "); }

if (setMode == 0 && setAlarm == 4 && mode>0){lcd.setCursor(14,3); lcd.print(" "); }

}

void setupClock (void) {

if (setMode == 8){

lcd.setCursor (0,2);

lcd.print ("Set Time Finish ");

lcd.setCursor (0,3);

lcd.print ("Set Date Finish ");

delay (1000);

rtc.setTime (hh, mm, ss);

rtc.setDate (dd, bb, yy);

lcd.clear();

setMode = 0;

}

if (setAlarm >0){alarmMode=0;}

if(digitalRead (bt\_clock) == 0 && flag==0) {flag=1;

if(setAlarm>0){WriteEeprom(); setAlarm=1; mode =5;}

else{setMode = setMode+1;}

digitalWrite(buzzer, HIGH);

}

if(digitalRead (bt\_timer) == 0 && flag==0){flag=1;

if(setMode>0){setMode=8;}

else{

setAlarm = setAlarm+1;

if(setAlarm>4){setAlarm=1;

WriteEeprom ();

mode=mode+1;

ReadEeprom();

}

}

lcd.clear();

digitalWrite(buzzer, HIGH);

}

if(setAlarm == 1 && mode==5){

lcd.setCursor (0,2);

lcd.print ("Set Timer Finish");

lcd.setCursor (0,3);

lcd.print ("-EEPROM Updated-");

delay (2000);

lcd.clear();

setAlarm=0;

mode =0;

alarmMode=1;

}

if(digitalRead (bt\_clock) == 1 && digitalRead (bt\_timer) == 1){flag=0;}

if(digitalRead (bt\_up) == 0){

if (setAlarm<2 && setMode==1)hh=hh+1;

if (setAlarm<2 && setMode==2)mm=mm+1;

if (setAlarm<2 && setMode==3)ss=ss+1;

if (setAlarm<2 && setMode==4)set\_day=set\_day+1;

if (setAlarm<2 && setMode==5)dd=dd+1;

if (setAlarm<2 && setMode==6)bb=bb+1;

if (setAlarm<2 && setMode==7)yy=yy+1;

//Timer

if (mode==0 && setMode==0 && setAlarm==1)timer1=1;

if (mode==0 && setMode==0 && setAlarm==2)timer2=1;

if (mode==0 && setMode==0 && setAlarm==3)timer3=1;

if (mode==0 && setMode==0 && setAlarm==4)timer4=1;

if (mode>0 && setMode==0 && setAlarm==1)StartHH=StartHH+1;

if (mode>0 && setMode==0 && setAlarm==2)StartMM=StartMM+1;

if (mode>0 && setMode==0 && setAlarm==3)FinishHH=FinishHH+1;

if (mode>0 && setMode==0 && setAlarm==4)FinishMM=FinishMM+1;

if(hh>23)hh=0;

if(mm>59)mm=0;

if(ss>59)ss=0;

if(set\_day>7)set\_day=0;

if(dd>31)dd=0;

if(bb>12)bb=0;

if(yy>2030)yy=2000;

if(StartHH>23)StartHH=0;

if(StartMM>59)StartMM=0;

if(FinishHH>23)FinishHH=0;

if(FinishMM>59)FinishMM=0;

rtc.setDOW(set\_day);

digitalWrite(buzzer, HIGH);

}

if(digitalRead (bt\_down) == 0){

if (setAlarm<2 && setMode==1)hh=hh-1;

if (setAlarm<2 && setMode==2)mm=mm-1;

if (setAlarm<2 && setMode==3)ss=ss-1;

if (setAlarm<2 && setMode==4)set\_day=set\_day-1;

if (setAlarm<2 && setMode==5)dd=dd-1;

if (setAlarm<2 && setMode==6)bb=bb-1;

if (setAlarm<2 && setMode==7)yy=yy-1;

//Timer

if (mode==0 && setMode==0 && setAlarm==1)timer1=0;

if (mode==0 && setMode==0 && setAlarm==2)timer2=0;

if (mode==0 && setMode==0 && setAlarm==3)timer3=0;

if (mode==0 && setMode==0 && setAlarm==4)timer4=0;

if (mode>0 && setMode==0 && setAlarm==1)StartHH=StartHH-1;

if (mode>0 && setMode==0 && setAlarm==2)StartMM=StartMM-1;

if (mode>0 && setMode==0 && setAlarm==3)FinishHH=FinishHH-1;

if (mode>0 && setMode==0 && setAlarm==4)FinishMM=FinishMM-1;

if(hh<0)hh=23;

if(mm<0)mm=59;

if(ss<0)ss=59;

if(set\_day<0)set\_day=7;

if(dd<0)dd=31;

if(bb<0)bb=12;

if(yy<0)yy=2030;

if(StartHH<0)StartHH=23;

if(StartMM<0)StartMM=59;

if(FinishHH<0)FinishHH=23;

if(FinishMM<0)FinishMM=59;

rtc.setDOW(set\_day);

digitalWrite(buzzer, HIGH);

}

}

void setTimer (){

//Timer

if (setMode == 0 && setAlarm >0 && mode>0){

lcd.setCursor (0,2);

lcd.print("Load ");

lcd.print(mode);

lcd.print(" On :");

lcd.setCursor (11,2);

lcd.print((StartHH/10)%10);

lcd.print(StartHH % 10);

lcd.print(":");

lcd.print((StartMM/10)%10);

lcd.print(StartMM % 10);

lcd.setCursor (0,3);

lcd.print("Load ");

lcd.print(mode);

lcd.print(" Off:");

lcd.setCursor (11,3);

lcd.print((FinishHH/10)%10);

lcd.print(FinishHH % 10);

lcd.print(":");

lcd.print((FinishMM/10)%10);

lcd.print(FinishMM % 10);

}

if (setMode == 0 && setAlarm >0 && mode==0){

lcd.setCursor (0,2);

lcd.print(" L1 L2 L3 L4 ");

lcd.setCursor (0,3);

if(timer1==1){lcd.print(" A");}

else{lcd.print(" D");}

if(timer2==1){lcd.print(" A");}

else{lcd.print(" D");}

if(timer3==1){lcd.print(" A");}

else{lcd.print(" D");}

if(timer4==1){lcd.print(" A");}

else{lcd.print(" D");}

}

}

void ReadEeprom() {

Start1HH=EEPROM.read(11);Start1MM=EEPROM.read(12);Finish1HH=EEPROM.read(13);Finish1MM=EEPROM.read(14);

Start2HH=EEPROM.read(21);Start2MM=EEPROM.read(22);Finish2HH=EEPROM.read(23);Finish2MM=EEPROM.read(24);

Start3HH=EEPROM.read(31);Start3MM=EEPROM.read(32);Finish3HH=EEPROM.read(33);Finish3MM=EEPROM.read(34);

Start4HH=EEPROM.read(41);Start4MM=EEPROM.read(42);Finish4HH=EEPROM.read(43);Finish4MM=EEPROM.read(44);

if(mode==1){StartHH=Start1HH, StartMM=Start1MM, FinishHH=Finish1HH,FinishMM=Finish1MM;}

if(mode==2){StartHH=Start2HH, StartMM=Start2MM, FinishHH=Finish2HH,FinishMM=Finish2MM;}

if(mode==3){StartHH=Start3HH, StartMM=Start3MM, FinishHH=Finish3HH,FinishMM=Finish3MM;}

if(mode==4){StartHH=Start4HH, StartMM=Start4MM, FinishHH=Finish4HH,FinishMM=Finish4MM;}

timer1=EEPROM.read(1);

timer2=EEPROM.read(2);

timer3=EEPROM.read(3);

timer4=EEPROM.read(4);

}

void WriteEeprom() {

if(mode==1){EEPROM.write(11,StartHH);EEPROM.write(12,StartMM);EEPROM.write(13,FinishHH);EEPROM.write(14,FinishMM);}

if(mode==2){EEPROM.write(21,StartHH);EEPROM.write(22,StartMM);EEPROM.write(23,FinishHH);EEPROM.write(24,FinishMM);}

if(mode==3){EEPROM.write(31,StartHH);EEPROM.write(32,StartMM);EEPROM.write(33,FinishHH);EEPROM.write(34,FinishMM);}

if(mode==4){EEPROM.write(41,StartHH);EEPROM.write(42,StartMM);EEPROM.write(43,FinishHH);EEPROM.write(44,FinishMM);}

EEPROM.write(1,timer1);

EEPROM.write(2,timer2);

EEPROM.write(3,timer3);

EEPROM.write(4,timer4);

}