CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Within the ambit of wireless technology, appearance of the remote control based devices and appliances have become the order of the day. It reduces human effort and increases the efficiency; hence every sector needs automation ranging from home to industries. Automation systems perform by allowing a number of devices to communicate all information to the user or the owner of the systems per the instructions and structure of the system. The applications of such automation systems could be in the areas such as heating, lighting, defense, energy management, audio and video systems, health monitoring and entertainment.

Being an emerging area of research, many works on home automation using different technologies have being carried out different technologies used in home automation includes Infra-Red, Bluetooth, Radio Frequency, Global System of Mobile Communication (GSM), Dual Tone Multi-frequency (DTMF) etc. The technology used depends on several factors like safety, cost, flexibility, reliability and easy usage.

Keeping all these facts in mind, this proposed project proposes a system which is based on Graphical User Interface (GUI) control through a Personal Computers (PC) or Laptop. This project proposes an automation of appliances like Fan, Bulb, Television, etc. The automation is done by making use of Global System of Mobile (GSM) technology and MATLAB-GUI. This is implemented with the help of Arduino microcontroller which can communicate with MATLAB GUI using serial communication.

1.2 Statement of the Research Problem

Presently, there is a huge advancement in the communication and indeed, in the automation sector. That is to say, different technologies have come to stay in the area of home automation and indeed, industrial automation. Each of the technology has its own advantages and shortcomings. While some technologies are secured, others are prone to attack. Sophisticated knowledge is required in using some of the available technologies hence only experts in this area can work on such devices or components. It is against this backdrop that we propose a simple and secured home automation system which makes use of MATLAB GUI and a controller (Arduino) to address the challenge of sophistication required in other areas of automation.

1.3 Aim and Objectives of the Project

The aim of this project is to design and construct a GUI based wireless home appliances switching system. The general objectives of this project are:

1. To send and receive text using Arduino and GSM module.
2. To design a Graphical User Interface using MATLAB software.
3. To control home appliances using GUI and GSM technology.

1.4 Significance of the Study

The significance of this project is as follow:

1. Acquisition of a better understanding of Telecommunication and Electronic Engineering.
2. Development of a device that is capable of saving power and providing safety to its users.
3. Development of a device that offers flexibility, security and privacy to its users.
4. Demonstration of one (Graphical User Interface) out of the numerous functions inherent in MATLAB as a computational and simulation tool for engineers, technologist and scientists.

1.5 Scope of the Project Work

Several technologies are available in the field of automation in general and home automation in particular; each of these technologies has its own advantages and short comings. However, this project is limited to the design and construction of a GUI based wireless home appliances switching system. Complex and sophisticated MATLAB function is outside the scope of this work.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Theoretical Framework

Automation is simply the performance of a system without human intervention. Automation could be home or industrial automation. Home automation or "Domestics" is simply building automation for residential areas (Hill, 2015). A home automation system will control lighting, climate, entertainment systems and appliances. It may also include home security such as access control and alarm systems (Consumer Reports, 2016). Many industries like brewery companies, steel company etc. are automated as most of the machine being used work perfectly by themselves without human intervention. Many institutions like banks use automated teller machine (ATM) to dispense cash and to carry out other transactions. Some restaurant now uses robots as the receptionist hence it is believed that in the nearby future, virtually all facets of human endeavor would be automated. Early home automation began with labor-saving machines; self-contained electric or gas powered home appliances became viable in the 1900s. The introduction of electric power distribution led to the invention of washing machine in 1904, Water Heater in 1889 (James, 1999). In 1975, the first general purpose home automation network technology, X10 was developed. It is a communication protocol for electronic devices (Rye, 1999). According to Li et.al (2016), there are three generation of home automation, these are: First generation (wireless technology with proxy server e.g Zigbee), Second generation (Artificial Intelligence controls electrical devices e.g amazon echo) and Third generation (robot buddy who interacts with human e.g Robot Rovio, Roomba).

The major component of home automation is the controller; it carries out all the controlling functions as per instruction given to it. Many technologies can be interfaced with the controller to automate home, some of such technologies includes: GSM technology, Bluetooth technology, RF technology and the hosts of others.

2.1.1 Microcontrollers

A Microcontroller is a self-contained system with peripherals, memory and a processor that can be used as an embedded system. Most programmable microcontrollers that are used today are embedded in other consumer product or machinery including phones, peripherals, automobiles and household appliances. Microcontroller could also be called "embedded systems" (Ejeh, 2016). Some embedded systems are more sophisticated, while others have minimal requirements for memory and programming length and a low software complexity. Input and output devices that can be interfaced with a microcontroller include: solenoid, LCD display, relays and sensors for data like humidity, temperature or light level among others. Microcontrollers are classified by bits, memory size, Random Access Memory (RAM) size, number of inputs and output lines, packaging type, supply voltage and frequency (speed). Examples of Microcontrollers are Peripheral Interface Controllers (PIC) series, Advanced Virtual RISC (AVRs) series and self-contained microcontrollers like Arduino series.

1. **Arduino series controllers**

   Arduino is an open source hardware and software company, project and user community that designs and manufactures single board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and remotely. The Arduino project was started at the Interactive Design Institute Ivrea (IDII) in Ivrea, Italy (David, 2011). The goal of the Arduino was to create simple, low cost tools for creating digital projects by non-engineers. The wiring platform consists of a Printed Circuit Board (PCB) with an atmega series microcontroller depending on the Arduino series; an Interactive Development Board (IDE) based on processing and library functions to easily program the microcontroller (Hernando, 2016). The microcontrollers are typically programmed using some functions from C and C++ programming languages. Some available Arduino series or family includes: Arduino Uno (R3), lilypad Arduino, Arduino Mega (R3), Arduino Leonardo, Arduino Nano among others. Arduino Nano is used for this project.

   Some common features present on the Arduino board are explained below (Arduino Nano as a reference):

   a. **Power (USB/Barrel Jack)**: Every Arduino board needs a way to be connected to a power source. The Arduino nano can be powered from a Universal Serial Bus (USB) cable coming from computer or wall power supply that is terminated in the Barrel Jack. The USB connection is also used to load code onto Arduino board. The recommended voltage for most Arduino models is between 6 and 12 Volts.
   b. **Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**: the Arduino has several kinds of pins, each of which is labeled on the board and used for different functions.
      i. **GND**: short for ground.
      ii. **5V and 3.3V**: 5V pin applies to 5Volts of power and 3.3Volts pin applies to 3.3Volts of power.

iii. **Analog**: The areas of pins under the analog pins label (A0 through A5 on the nano) are analog input pins. These pins can be used to read signals from analog sensor like a temperature sensor and convert it into a digital value that we can read.

iv. **Digital**: there are digital pins on the board also; there are 14 pins (0 through 13 on the nano). These pins can be used for both digital inputs like telling the state of a Push Button and digital output like powering a Light Emitting Diode (LED).

v. **PWM**: Some of the digital pins (3,5,6,9,10 and 11 on the nano) can be used for Pulse Width Modulation (PWM) other than digital functions.

vi. **AREF**: AREF stands for Analog Reference. Most of the time, it is left alone. It is sometimes used to set External Reference Voltage (between 0 to 5Volts) as the upper limits for the analog input pins.

vii. **Reset button**: This is used for resetting or to restart any code that is loaded on the Arduino.

viii. **Power LED indicator**: This LED should light up whenever the Arduino board is powered.

ix. **T_X, R_X LEDS**: These are the pins responsible for serial communication on the Arduino nano, T_X and R_X appears in two places: one by digital pins 0 and 1 and a second time next to the T_X and R_X indicator LEDs. These LEDs will give some visual indication whenever the Arduino is receiving or transmitting data.

x. **Main IC**: This is the brain of the Arduino. The main IC on the Arduino varies from board to board, but it is usually from the Atmega family from the ATMEL Company. Atmega328P is on Arduino nano.

xi. **Voltage regulator**: This controls the amount of voltage that is let into the Arduino board.

xii. **In-Circuit Serial Programmer (ICSP):** This is the programmer that is used to program the microcontroller

The figure below shows an Arduino nano with its parts:

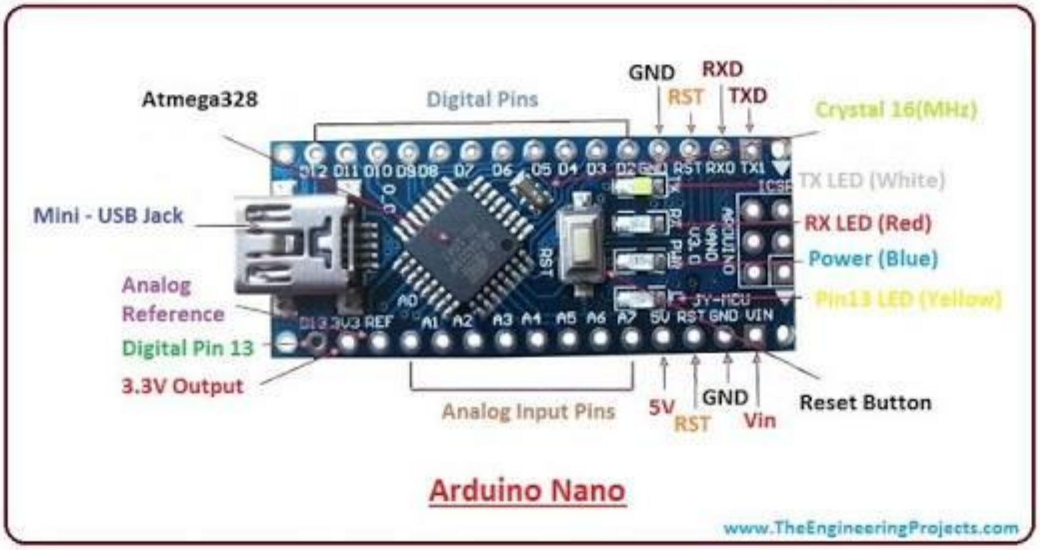


Figure 1: Arduino nano with its Parts.

(Source: https://encrypted-tbno.gstatic.com/images?q)

2.1.2 MATLAB

MATLAB is an acronym for Matrix Laboratory. It is a multi paradigm numerical computing environment and proprietary programming language developed by Mathworks. MATLAB allows matrix manipulations plotting of functions and data implementation of algorithms, creation of user interface and interfacing with programs written in other programming languages including C, C++, C#, JAVA, FORTRAN and PYTHON. Despite being used primarily for numerical computing, an optional tool box uses the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, simulink adds graphical multi-domain simulation and model based design for dynamic and embedded systems. MATLAB has more than 3 million users worldwide as of 2018 (Mathworks, 2018). MATLAB users come from different backgrounds of engineering, science and economics.

Cleve Moler, the chairman of the computer science department at the University of New Mexico started developing MATLAB in the late 1970s (Cleve, 2004). He designed MATLAB to give his students access to LINPACK and EISPACK without having to learn FORTRAN. MATLAB soon spread to other universities and found a strong audience within the applied mathematics community. Jack Little, an engineer was exposed to MATLAB during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. In 1984, Mathworks was formed and MATLAB was rewritten in C to continue its development.

1. **MATLAB Graphical User Interface (GUI)**

MATLAB GUI is an interactive platform whereby a user can communicate with a computer system (hardware or software), it is an interface between a user of a computer and the computer system. For this project, MATLAB GUI is just an interface whereby a user of appliances at home can communicate with the appliances through a controller.

MATLAB GUI has many useful components like Edit Text Box, Sliders, Push Buttons, Graph Axes, Static Text Box, e.t.c. by using these elements; one can make a professional program like VC++ or VB. This is especially useful to electronics engineers as they are not familiar with window based programming language. To launch a MATLAB GUI on a system that MATLAB software is installed, the following steps is followed

a. Type “guide” at the command prompt
b. Select “create New GUI” tab
c. Select “blank GUI (Default)”
d. Then press OK

Then the GUI design editor appears with title “untitled.fig”, any desired title can be given while saving. When a GUI is created in MATLAB; MATLAB automatically generates functions or codes relating to the used components. The codes for the GUI can be added under the “callback” function for each of the components. “Callback” functions are the instructions that will be followed when the user pushes the buttons or does something with the components that was created in the MATLAB GUI (Quan, 2007).

2. **Arduino and MATLAB**

The versatility of MATLAB as a fourth world programming language; the developer, Mathworks has developed software that makes MATLAB to communicate with microcontrollers like Arduino for use in embedded systems (Pawan, 2017). The software is called “MATLAB and simulink support for Arduino” or “Arduino IO package”. The following procedures are followed in installing Arduino IO package on MATLAB for communication with Arduino:

a. Download the Arduino IO package from mathwork’s websites; log in if u are directed.
b. Burn the adioe.pde file to the Arduino using Arduino IDE. This adioe.pde file can be found in Arduino IO package. Arduino IO\pde\adioe\adioe.pde
c. Open the MATLAB software, go through the Arduino IO folder. Open the installed arduino.m file and run it in MATLAB. A message showing “Arduino folders added to the path” in the command window of MATLAB means path is updated to Arduino folders.

This is how Arduino can be made to communicate with MATLAB. For higher version of MATLAB (like R2015b or R2016a), you can directly click “Get hardware support packages” from where you can install Arduino packages for MATLAB. After the files are installed, GUI can be created. The installation of the support packages enable the code to be written in MATLAB GUI instead of the Arduino IDE. The coding in Arduino IDE is only slightly different from that in MATLAB GUI; for example, to make a pin HIGH in Arduino, we write code as digital Write (pin, HIGH). In MATLAB, we use this function with the help of an object or variable like a digital Write (pin, HIGH) and likewise so on (Pawan, 2017). Nonetheless, the project was coded using serial communication between Arduino and MATLAB.

2.1.3 GSM transmission

Basically, GSM (Global System for Mobile communication) is an open, digital cellular technology used for transmitting mobile voice and data services. GSM differs from first generation wireless system in that it uses digital technology and Time Division Multiple Access (TDMA) transmission methods. GSM is a circuit-switch system (can transmit and receive simultaneously) that divides each 200Hz channel into eight 25KHz time slots. GSM supports data transfer speed of up to 9.6Kbits/sec, allowing the transmission of basic data services such as SMS (Short Messaging Services). It also allows international roaming capability, allowing users to access the same services when traveling abroad as at home. GSM bands may include, 850MHz, 900MHz, 1800MHz and 1900MHz bands depending on the parts of the wind the user finds himself or herself. This technology is used in mobile phones and GSM modules (that work like normal handsets). Mostly, all GSM modules obey the standard ‘ATTENTION’ command called ‘AT command’. This is the language that is understood by GSM modules; hence they are controlled using these commands. Depending on its brand name, some well known GSM modules are: SIM900, SIM800, A6, A7, A9 among others. SIM products are manufactured by SIMcom while ‘As’ products are manufactured by AI-Thinker.

2.2 Conceptual Framework

The concept behind the operation of this work is discussed here. The project is divided into blocks and each block has its own peculiar function as discussed below. Moreover, there is a transmitting and receiving section.

2.2.1 Development of block diagram

1. **Transmitting section**

The purpose of this section is to increase the range of control of the appliances. The transmitting section is made up of the following:

a. **MATLAB GUI**: This is a set of Push Buttons designed in the MATLAB GUI using MATLAB software. It is so designed using MATLAB software installed on a PC. In addition to these Push Buttons and Edit Text Box is also created to take password from an authorized user. It is from these Buttons on the MATLAB GUI that the appliances are controlled from.

b. **USB-to-TTL converter:** this is used for serial communication between the MATLAB GUI and the GSM module. The summary of its connection with the GSM module is presented below:

i. USB-to-TTL converter 5V pin to V_{cc} of GSM module

ii. USB-to-TTL converter TXD pin to RXD of GSM module

iii. USB-to-TTL converter GND pin to GND of GSM module

iv. USB-to-TTL converter RXD pin to TXD of GSM module

c. **GSM module:** This is basically used to send text message to the receiving end.

2. **Receiving section**

This is the section that receives the intelligence that was sent by the transmitter. It is this section that the appliances to be controlled are connected to. It is made up of the following:

a. **Arduino nano microcontroller:** The Arduino controller at this end turns it output pin HIGH or LOW based on the received message from the module.

b. **GSM module:** This receives the text sent by the transmitting module; it then sends the message through the software serial part to the Arduino controller.

The Arduino controller at this end turn it output pin HIGH or LOW based on the received message from the module.

c. **Relay Shield**: This is a small PCB sub-assembly with one or more relays. Four channel relay shield is used for this project. The usage of this shield is due to its flexibility and inherent advantages over four single relays which needs a relay driver circuitry.

d. **Appliances:** This is the last block in this project. It is the block that is made up of the appliances to be controlled. The figures show the block diagram for this project.

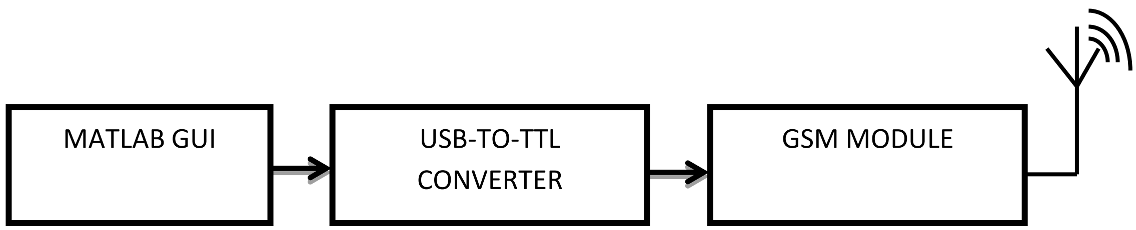


Figure 2. Transmitting Section

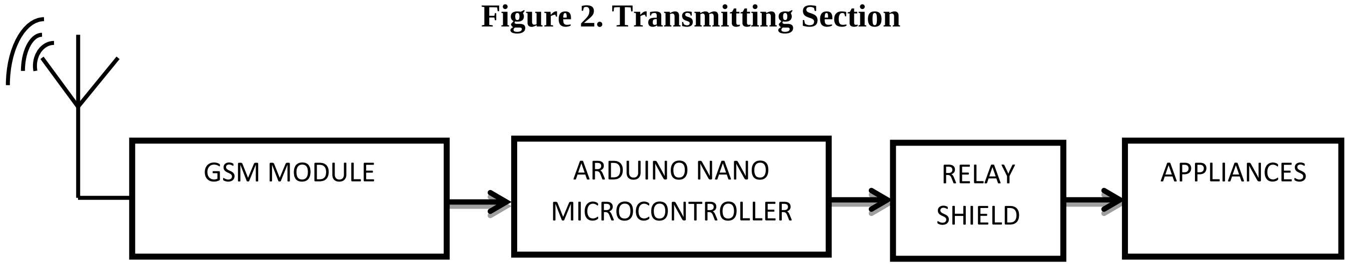
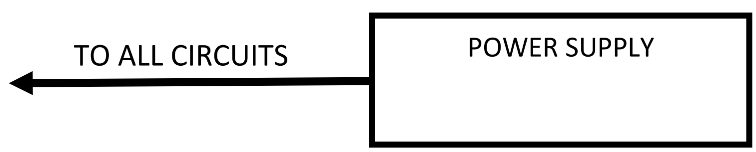


Figure 3. Receiving Section



2.3 Review of Related Empirical Studies

Many works have been carried out in the aspect of home automation using different available technology. Some of the works done in this regard are presented here for more information or research in this area.

Attaih and Hamdy (2011) stated that power systems quality is affected by energy saving issues, which in turn has effect on the global environment. The technology of smart phone (that is, promising area) can be considered as one of the most important tools that deal with the demand for power consumption hence, producing various benefits such as security, safety and consequently good comfort.

Mirza *et al*. (2014) carried out comparative analysis of most common and recent techniques that have been implemented in the field of home automation systems along with their advantages and disadvantages.

In order to ensure scalability and management of home appliances, Shafee and Hamed (2012) proposed a design and prototype implementation of new home automation system that uses Wi-Fi technology as a network infrastructure connecting its parts. The proposed project consists of two main components: the first part is the web server that manages, controls and monitors user's home and Hardware interface module.

Dinesh P. *et al.* (2013) proposed a system of home automation using voice command through GSM, Microsoft Speech Recognition engine, speech SDK 5.1 is used to understand the voice command of the user. They further said that the proposed system is cost effective and more secure than existing system.

In a bid to incorporate feedback notification in home automation systems, Johri *et al*. (2015) proposed the design and prototype implementation of home automation systems that uses Wi-Fi technology and android operating system. User can control electrical appliances in home or office via smart phones in this proposed system.

With the advance in technology, a novel area like Internet of Things (IoT) has emerged. This technology is expected to rule the world within a few years. Home automation system uses the Internet of things for monitoring and controlling of the electrical and electronic appliances at home from any remote location by simply using a smart phone (Pooja *et al.*, 2016).

System architecture developed by NazmulHasan (2013) consists of Atmega16 as the brain of the system along with different supporting hardware like Remote controller, touch screen, temperature and humidity sensor, speed regulator. As per command forward by user through touch screen, all home appliances can be controlled manually. The proposed system also works with complete automotive mode of detecting presence of human beings according to the given command.

Generally, old people have more needs than young people. Thus efforts are made to improve home automation system by using Z-wave technology to transfer data in home network to have control over devices (Prashanth and Gaurav, 2014).

Speech processing using MATLAB and Android application plays vital role to support home automation systems, a system presented by Parameshachar *et al.* (2013) and

Tharaniyasoundhari (2015) uses speech processing and speech recognition to control home appliances.

On the basis of the versatility and compatibility of the fourth world programming (MATLAB) with controllers like Arduino Series, Vijaya and Surender (2016) proposed a system which focuses on the remedial measures made to reduce human interactions with machine by using an industrial monitoring system. The proposed system uses robot hanged over the ropes, to measure the parameters of the machines in aerial mode. The control room interface according to them makes use of MATLAB GUI model (which shows the parameters from the robot and then plot it on a graph).

In 21st century, various system implementations are present for home automation with wired as well as wireless communication as key element. Home automation is not new! Throughout history, we have continuously strived to automate tasks in the home in order to make our lives easier. Pankaj *et al.* (2014). A comparative analysis on the most common and recent techniques that have been implemented in the field of home automation systems are given below.

Wireless technologies like Bluetooth and Wi-Fi have been used in contemporary home security systems using low cost, low power, less complexity RF module. It uses multi-hop communication for data transfer. This multi-hop technique gives out an unlimited range of communication thus giving it an edge over the other wireless technologies. Here the radio frequency transmission system employs Amplitude Shift Keying (ASK) technique with transmitter and receiver operating at 433MHz. (Krishna *et al.,* 2015).

Pankaj *et al* (2014) proposed a project of smart home system because of our humans' bad attitude itself. Lazy to turn OFF/ON home appliances are common problem among us. A better smart home system is able to overcome such a serious problem. The automation centers on recognition of voice commands and uses low-power RF ZigBee wireless communication modules. This technology is relatively cheap, and is made for those with special needs like the elderly and the disabled (Benjamin Arul S, 2014).

Naresh *et al.* (2013) opined a paper to put forwards the design of home automation and security system using ARM7 LPC2148 board. The design is based on a standalone embedded system board ARM7 LPC2148 at home. Home appliances are connected to the ARM7 and communication is established between the ARM7 and ARM9 with Bluetooth device. The home appliances are connected to the input / output ports. Bluetooth technology is used to provide remote access from PC/laptop or smart phone (Ramlee *et al.,* 2013).

Nowadays, the remote home automation turns out to be more and more significant and appealing. It improves the value of our lives by automating various electrical appliances or instruments. Harinath R. *et al*. (2015) proposed remote monitoring through mobile phone involving the use of spoken commands. The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task.

With the openness, flexibility and features that Android offers, it has been widely adopted in applications beyond just smart phones. Anwaarullah S. (2013) proposed the design and implementation of a low cost yet compact and secure Android smart phone based home automation system. The controlling application which has been developed for Android devices can also be easily developed on other popular smart phone operating systems like Apple's iOS, Microsoft's WP7/8 and BlackBerry OS.

Kumar A. (2013) proposed a project to develop the Graphical User Interface of device control through MATLAB, interface the MATLAB GUI that consist of transmitter and receiver program with hardware via serial communication and control the various devices. By using MATLAB GUIDE, the process of laying out and programming GUIs and interface with microcontroller via serial communication port to control the devices will be easier because it is already providing a set of tools.

Ahmed O. and Karim, (2012) suggested the design and prototype implementation of new home automation system that uses WiFi technology as a network infrastructure connecting its parts. The proposed system consists the server (web server), which presents system core that manages, controls, and monitors users' home unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface modules.

Consistent with the marketing research firm ABI regarding 4 million home automation systems were subscribed globally in 2013. Avadhoo *et al.* (2007) opened that Image acquisition device can be interfaced to MATLAB that will continuously show the status of household equipments on Graphical User Interface (GUI) designed in MATLAB. Proper commanding is done from MATLAB GUI, household equipments can be turned ON or OFF which are interfaced to Arduino through relay board.

Ibhaskha *et al. (*2015) presents a low cost and flexible home control and monitoring system using an embedded microprocessor and microcontroller, with IP connectivity for accessing and controlling devices and appliances remotely using Smart phone application. The proposed system does not require a dedicated server PC with respect to similar systems and offers a novel communication protocol to monitor and control the home environment with more than just the switching functionality.

Rajdev D. (2011) opined to show the benefits of a smart house and the areas of usage of smart living systems. Details about the technical substructure and application of the designed home automation system through gesture recognition are described. This report presents the extension of existing vision based gesture recognition system using an algorithm that combines the classical and novel approach for gesture recognition.

2.4 Summary of Literature Review

From the reviewed literatures, it is pertinent to note that each of the work carried out has its own shortcomings. The shortcoming that this work tends to address as regards to home automation project is flexibility in operation. This work is flexible in the sense that, only clicking of the Push Buttons in the GUI is the only task hence anybody can handle the task. As a way of preventing an intruder, only authorized user can operate each of the appliances from the User Interface (UI). This measure to a large extent adds more Security and Uniqueness to this work. An alternative mobile phone can also be used instead of the GUI.

CHAPTER THREE

3.0 CIRCUIT ANALYSIS AND DESIGN

3.1 Introduction

The analysis and design in this work is divided into three, viz:

1. Analysis of the Power Supply Unit (PSU)
2. Analysis of the transmitting section
3. Analysis of the receiving section

3.1.1 Analysis of the Power Supply Unit (PSU)

Two power supply units will be used in this project: The PSU for the transmitter section and the PSU for the Receiver section. However, the design follows the same procedure hence, PSU to be design here is general. However, references will be made to device or components peculiar to any section.

1. **Transforme**r: A 220/12V transformer was used in our project to convert AC input from public supply to 12V for our circuit.
2. **Rectification stage**: The rectifier makes use of four IN4007 diodes which acts as a full wave rectifier to convert the ac input to a pulsating DC. The choice of IN4007 diode is based on their Peak inverse voltage (PIV) which is the maximum possible voltage across the diode when it is reverse biased; it is 1000V for IN4007. Also it has low forward voltage drop of 0.7V and high surging current.

   The peak voltage from Rectifiers output and the current is given by

$$\mathrm{V_p} = \sqrt{2}\ \mathrm{x}\ V_{rms} - 2V_d$$

$$I_p = \sqrt{2}\ \mathrm{x}\ I_{rms}$$ Source: Gupta JB, (2014)

Where: $\mathrm{V_p} = Peak\ Voltage$

$\mathrm{V_{r.ms}}$ = Root Mean Square RMS value of the secondary voltage

$\mathrm{V_d}$ = diode forward voltage drop

$\mathrm{I_p}$ = peak current and $\mathrm{I_{r.m.s}}$ = RMS current

$$The\ rating\ of\ the\ transformer\ is\ 220/12V, 300mA = 0.3$$

$$\Rightarrow \ V_{peak} = \sqrt{2} * 12 - 2 * 0.7$$

$$V_{peak} = 15.571Volts$$

For the full cycle, the average dc voltage is given by

$$V_{dc} = \frac{2Vpeak}{\pi} = \frac{2 \text{ x } 15.571}{\pi}$$

$$V_{dc} = 9.913Volts, \quad similarly,$$

$$I_{peak} = \sqrt{2} \text{ x } I_{rms}\,; I_{rms} = 300mA = 0.3A$$

$$I_{peak} = \sqrt{2} \text{ x } 0.3 = 0.424A$$

$$I_{dc} = \frac{2I_{peak}}{\pi} = \frac{2 \text{ x } 0.424}{\pi} = 0.270A$$

3. **Filter stage**: This makes use of a capacitor. It value is chosen to provide low impedance to ripples or AC voltage and high impedance to DC voltage; hence ripples are shunted to ground while only DC component are passed to the load (Gupta, 2014).

   The filtering or smoothening capacitor is chosen to have a Ripple Factor (ϒ) of 3% of the DC voltage

$$Ripple\ factor,\ \Upsilon = \frac{V_{ac\,(rms)}}{V_{dc}}$$

$$V_{ac(r.m.s)} = RMS\ rippple\ voltage$$

$$V_{dc} = average\ DC\ voltage$$

If the discharging time is small; the output wave from the filter could be assumed to be triangular wave, hence the peak RMS voltage is given by:

$$V_{ac\,(rms)} = \frac{V_r}{2\sqrt{3}};$$

$$V_r = ripple\ voltage\ and$$

$$V_r = \frac{I_{dc}}{2FC},\ \ for\ a\ full\ cycle\ hence$$

$$V_{ac\,(rms)} = \frac{I_{dc}}{4\sqrt{3} \text{ x } F \text{ x } C}$$

$$C = \frac{I_{dc}}{4\sqrt{3} \text{ x } F \text{ x } \Upsilon \text{ x } Vdc},\ \ F\ is\ frequency = 50Hz, \mathrm{I_{dc}} = 0.27A\ from\ above$$

$$\Upsilon = 3\% = 0.03, V_{dc} = 9.91V\ from\ above$$

⇨ $C = \frac{0.27}{4\sqrt{3} \text{ x } 50 \text{ x } 0.03\ x\ 9.91} = 2622\mu F.\ Standard\ value\ of\ 2500\mu F\ was\ chosen$

4. **Regulator stage**: A three terminal IC voltage regulator of the 78XX family, IC7805 is used to provide an output voltage of 5V.
5. **Red LED power indicator limiting Resistor**: The value of the forward voltage drop and the maximum current are 2.2V and 20mA respectively from the datasheet. The output voltage from the Regulator output is 5V. the resistor value is given by

$$R_{led} = \frac{V_{dc}\ -\ V_{led}}{I_{max}}$$

$$R_{led} = limiting\ resistor, V_{dc} = Output\ DC\ voltage$$

$$V_{led} =\ LED\ forward\ voltage\ drop,$$

$$\mathrm{I}_{max} = Maximum\ led\ current$$

$$R_{led} = \frac{5-2.2}{0.02} = 140\Omega.$$

$$standard\ value\ of\ 330\Omega\ with\ 5\%\ tolerance\ was\ chosen$$

3.1.2 Analysis of the transmitting section

The design and analysis of this section is made up of the following:

1. **MATLAB GUI**: The GUI is designed using a PC on which MATLAB software is installed. The GUI is made up of one edit text box to collect user password and ten sets of push buttons. Each push button has its own peculiar function to perform. On designing the GUI, the MATLAB automatically generates a code for the GUI with two files (the fig file and the m file). The code for appliance control is added under the callback function of the components created. It is an object oriented type of coding. The MATLAB GUI communicates with the Arduino over serial communication hence it is necessary to note down the Com Port for the Arduino as it will be referred to when writing the code in the MATLAB environment.
2. **Arduino Microcontroller**: This is used as the controller for this project; it receives command from the GUI through the help of serial communication between MATLAB and Arduino. It sets it output pin HIGH or LOW based on the signal it receives. The digital data is then sent to the receiving GSM module using the transmitting GSM module.
3. **GSM module unit:** GSM module is an electronic sub-assembly that works like a GSM mobile phone, That is, it can receive text messages, send text messages, receive and make

calls respectively. It comprises 5V input/output voltages, antenna, SIM jacket (which houses the GSM SIM card and the R_x and T_x. It communicates with Arduino over serial communication. A9 GSM module is used in this project at both the receiving and sending end. SIM card was fixed to both modules. Some basic features of A9 GSM/GPRS module are:

i. Walking voltage of 5V (USB) or LIPO battery (3.3V - 4.2V) (3.3V – 4.2V) on the V_{BAT} pin.
ii. Supports the GSM/GPRS four bands, including 850, 900, 1800 and 1900MHz.
iii. Support digital audio and analog audio, support HR, FR, GFR, AMR, Voice coding.
iv. Support voice calls and SMS messages.
v. Embedded network service protocol stack.
vi. Support standard GSM 07.07.07 AT command.
vii. The data sheet of A9 GSM/GPRS mode is on Appendix ‘A’.

3.1.3 Analysis of the receiving section

The analysis and design is made up of the following blocks:

1. **Receiver GSM module**: This module receives the command sent by the transmitting module, process it and send it to the Arduino which sets it output pin HIGH based on the command received.
2. **Relay shield**: Relay shield is a relay sub-assembly that houses one or more relays. In this project, four channel relay shield is used based on the number of appliances to be automated. Four channel 5V (DC) relay shield is used in this project.
3. **Appliances**: Four different home appliances will be controlled in this project.

3.2 Flow Chart

The flow chart for the project is presented in figure 4:

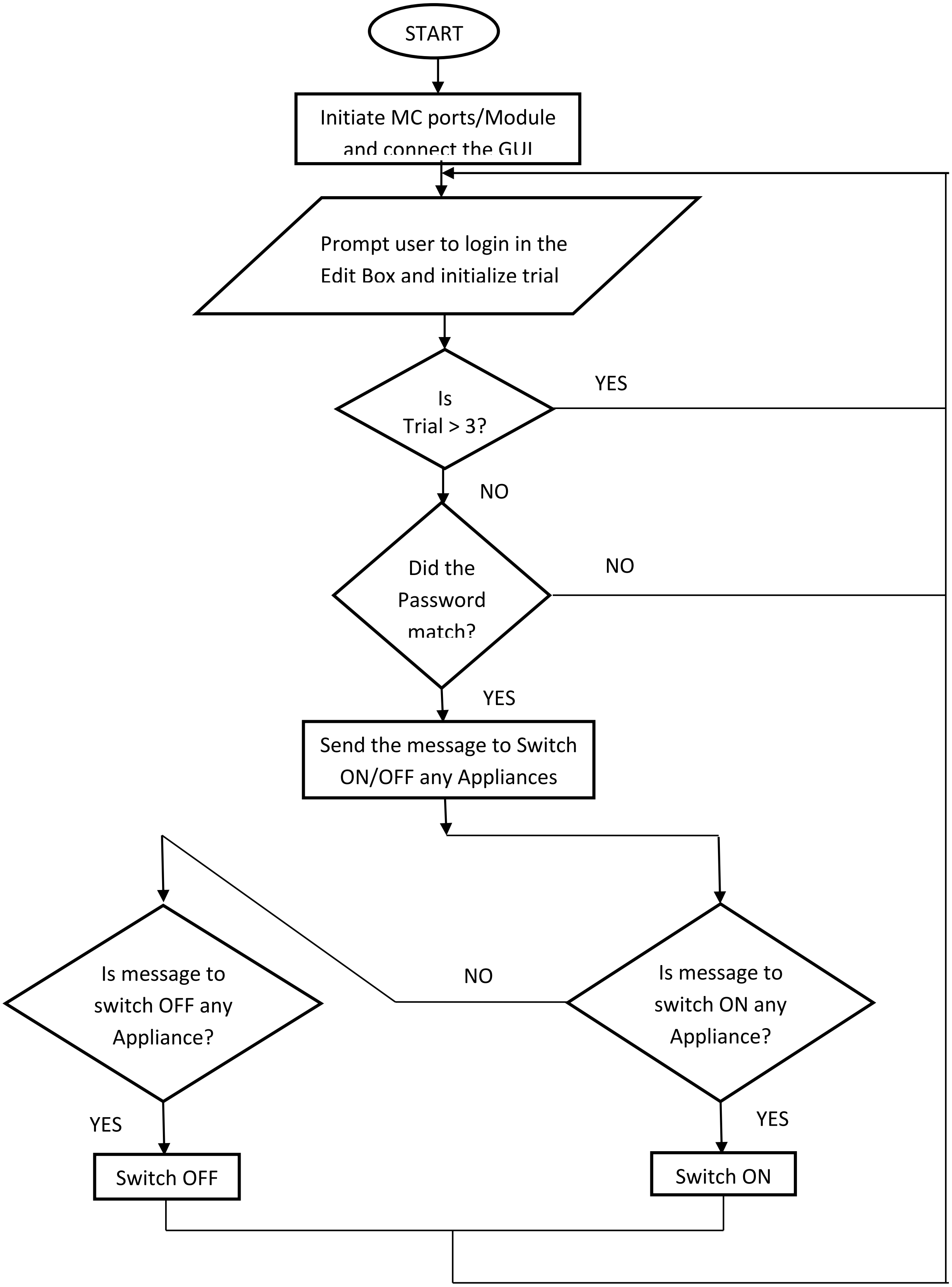


Figure 4: Project's Flow Chart

3.3 Programming:

The program for this project is divided into two:

1. MATLAB coding for the GUI design.
2. Arduino coding using the Arduino IDE.

3.3.1 MATLAB coding

In the project; ten (10) push buttons were designed in the MATLAB GUI. The instruction to be performed by each of the buttons is:

i. Two (2) buttons will be used for appliance 1 control
ii. Two (2) buttons will be used for appliance 2 control
iii. Two (2) buttons will be used for appliance 3 control
iv. Two (2) buttons will be used for appliance 4 control
v. Two (2) buttons will be used for all appliance control (ON/OFF)

There is also a login message box to collect valid password (By default, this will show first) information of user (whether is authorized or unauthorized). After these components are created in the GUI, the MATLAB automatically generates the code. The instruction to be performed by each of the buttons is added under the 'call back' function of the components.

3.3.2 Arduino based coding:

Arduino Integrated Development Environment (Arduino IDE) was used to perform the Arduino controller. This allows the programmer to write code in embedded C language. The program is burnt to the Arduino through a USB port connected from the system to the Arduino Board. It is to be noted that, the Arduino code is first burnt to the Arduino before running the MATLAB GUI. MATLAB code is on Appendix A while Arduino code is on Appendix B.

3.4 Simulation Model

The Simulation Models for the project are presented below:

1. The power supply simulation model (presented in figure 5)
2. The transmitting section simulation model (presented in figure 6)
3. The receiving section simulation model (presented in figure 7)

3.4.1 The power supply simulation model:

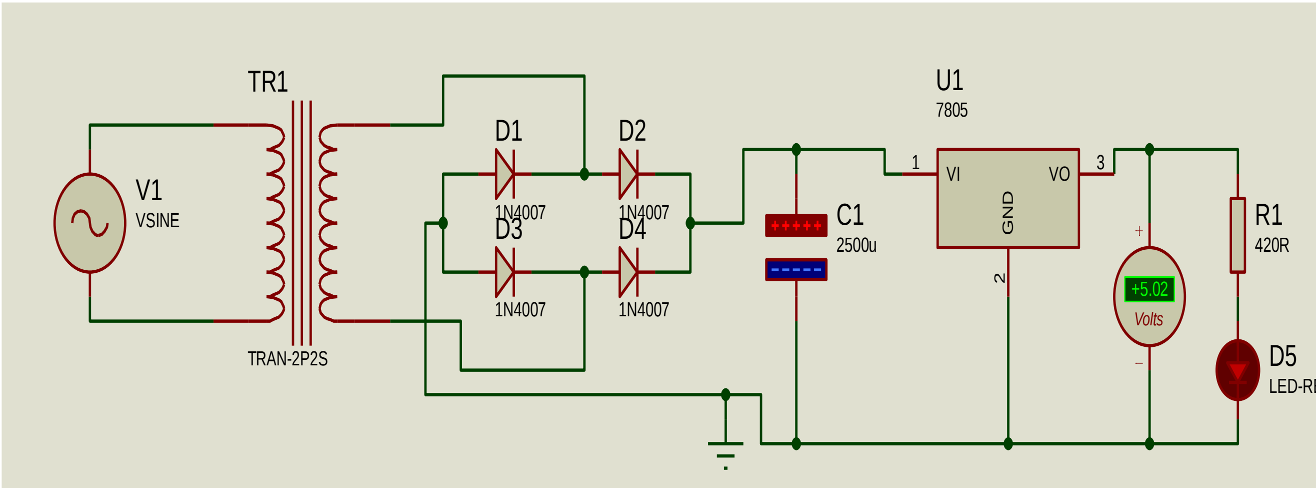


Figure 5: Designed Power Supply

3.4.2 The transmitting section simulation model

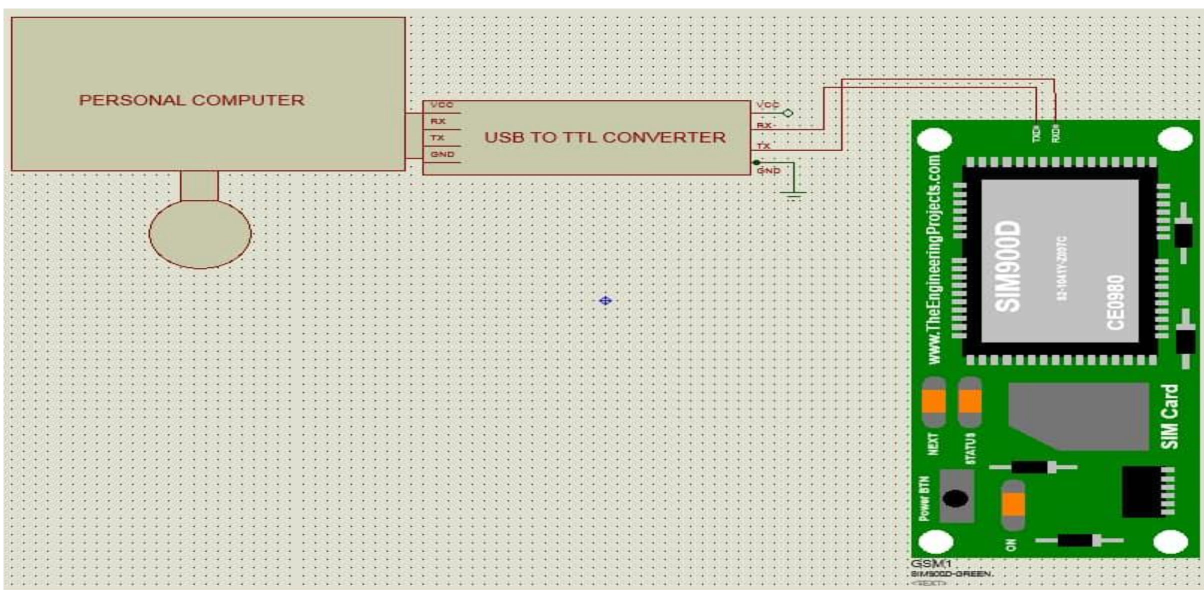


Figure 6: Transmitting Section

3.4.3 The receiving section simulation model

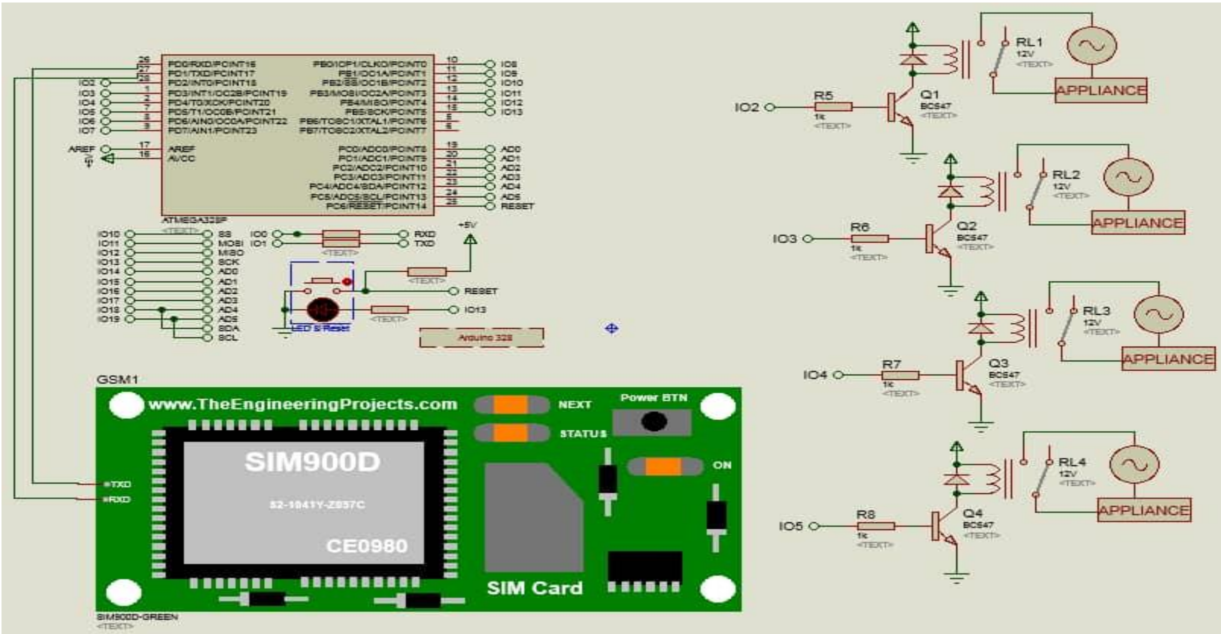


Figure 7: Receiving Section

3.5 Circuit Diagram

The overall circuit diagram is presented in figure 8.

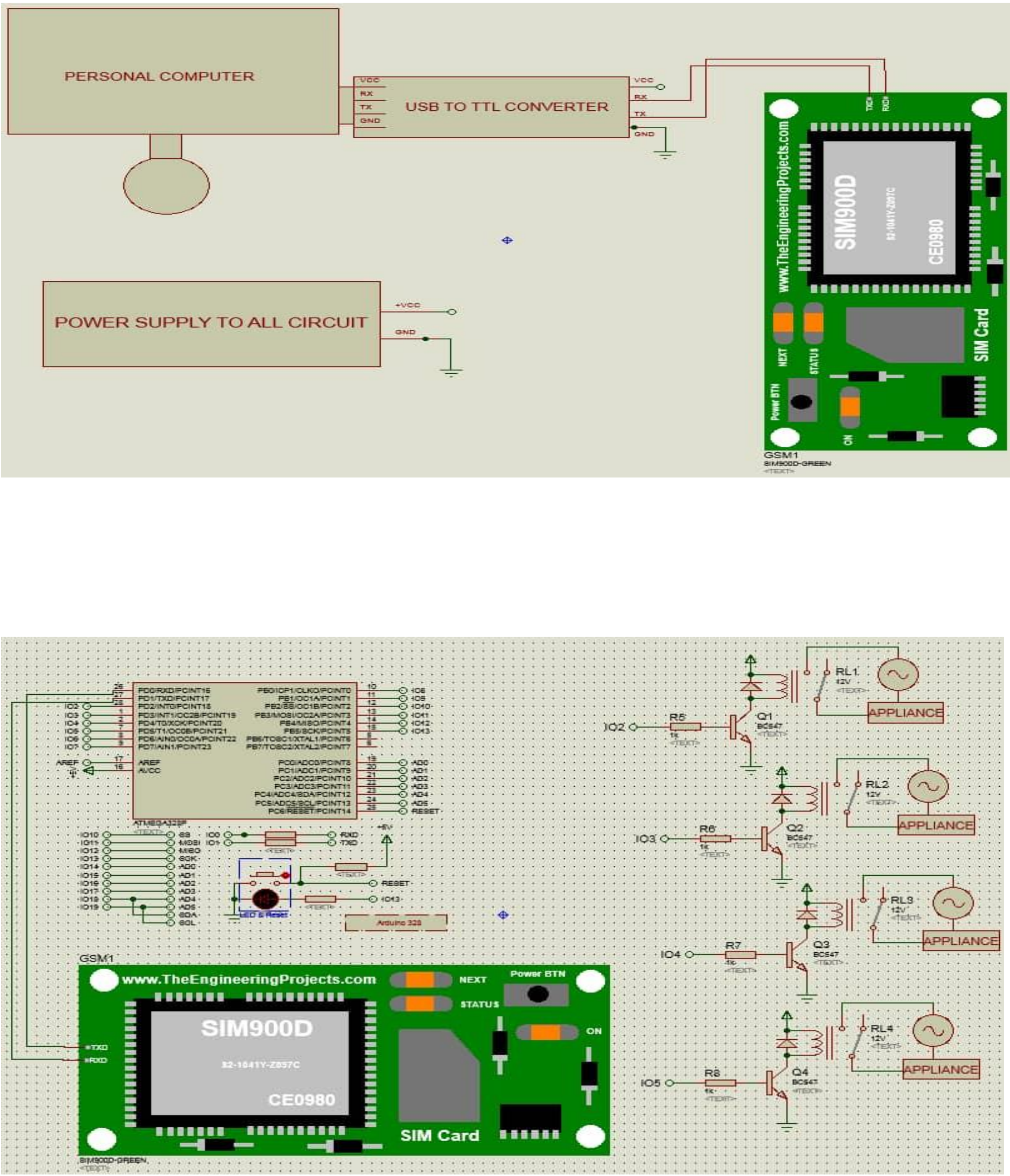


Figure 8: Overall Circuit Diagram

CHAPTER FOUR

4.0 CONSTRUCTION, TESTING, RESULTS AND DISCUSSION

4.1 Construction

After simulating the circuit and it was discovered to be working normally; we then implemented the circuit on project board. The construction is divided into:

1. Construction on bread board
2. Construction on Vero board

4.1.1 Construction on bread board

The components of the designed circuit were first of all implemented on the bread board in order to test and ensure the workability of the design. After it was found to be working, it was transferred to Vero board. The construction was done in successive stages; starting with the power supply unit and terminating at the output unit.

4.1.2 Construction on Vero board

This is the final construction where all necessary connections and soldering were done permanently before encasing. It entails transferring the construction on the bread board to the Vero board where it is permanently soldered, tested and encased. However, the components: GSM module, Arduino and relay shield all come with their respective boards.

4.2 Testing

Testing entails verifying that at all stage, the construction conform to the projected aim and objectives of the project work. In essence, every component was tested before and after soldering using the relevant test instruments in the laboratory. At every stage, continuity test out with the aid of multimeter and all lapses were before proceeding to the next stage.

4.3 Results

The results of the measurement taken are presented as follows:

Input DC voltage	12.22V
Input to Arduino	12.22V
Input to Relay Shield	8.22V
Input to GSM Module	4.98V

4.4 Discussion

The results of the transmitting section and receiving section are discussed as follows:

4.4.1 The transmitting section

The transmitting section comprises the GUI (which is made up of the login section and the appliances control unit), a USB-to-TTL converter (for serial communication between GUI and GSM module), and the transmitting GSM module. The wiring of this section is as shown in figure 9. By default, the login dialog box will show first as shown in figure 10. After successful login, the appliances switching unit will pop up as shown in figure 11. User(s) can only enter a password three times, after which (if unsuccessful), a dialog box will pop up as shown in figure 14. From the information displayed, a user will have to call on the login function as directed to continue with the login after the minimum number of trials is exceeded.

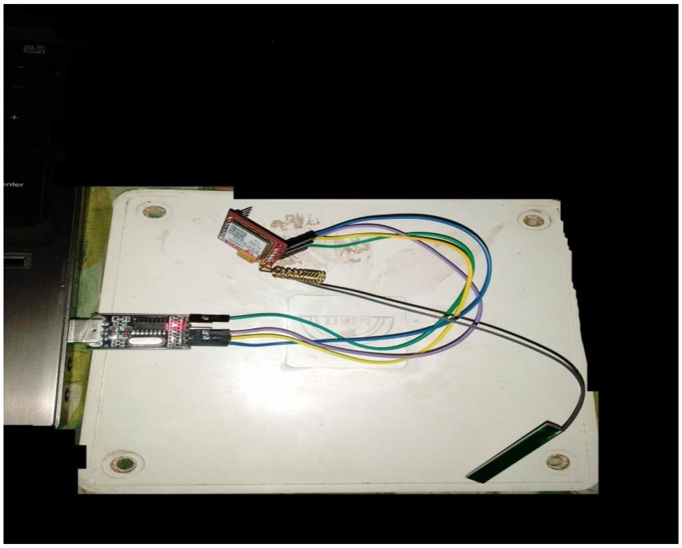


Figure 9: The Transmission Section Testing.

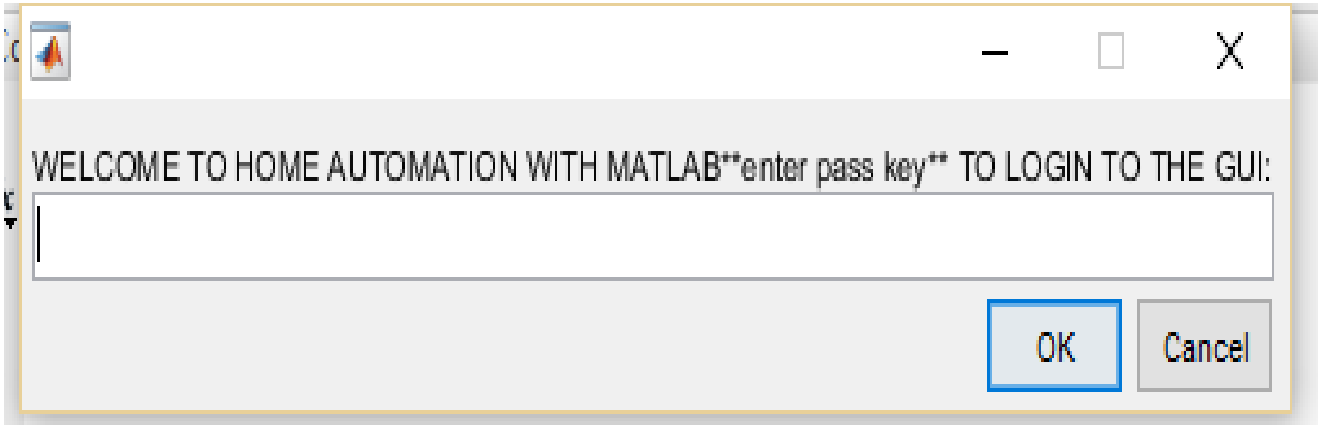


Figure 10: Password prompt

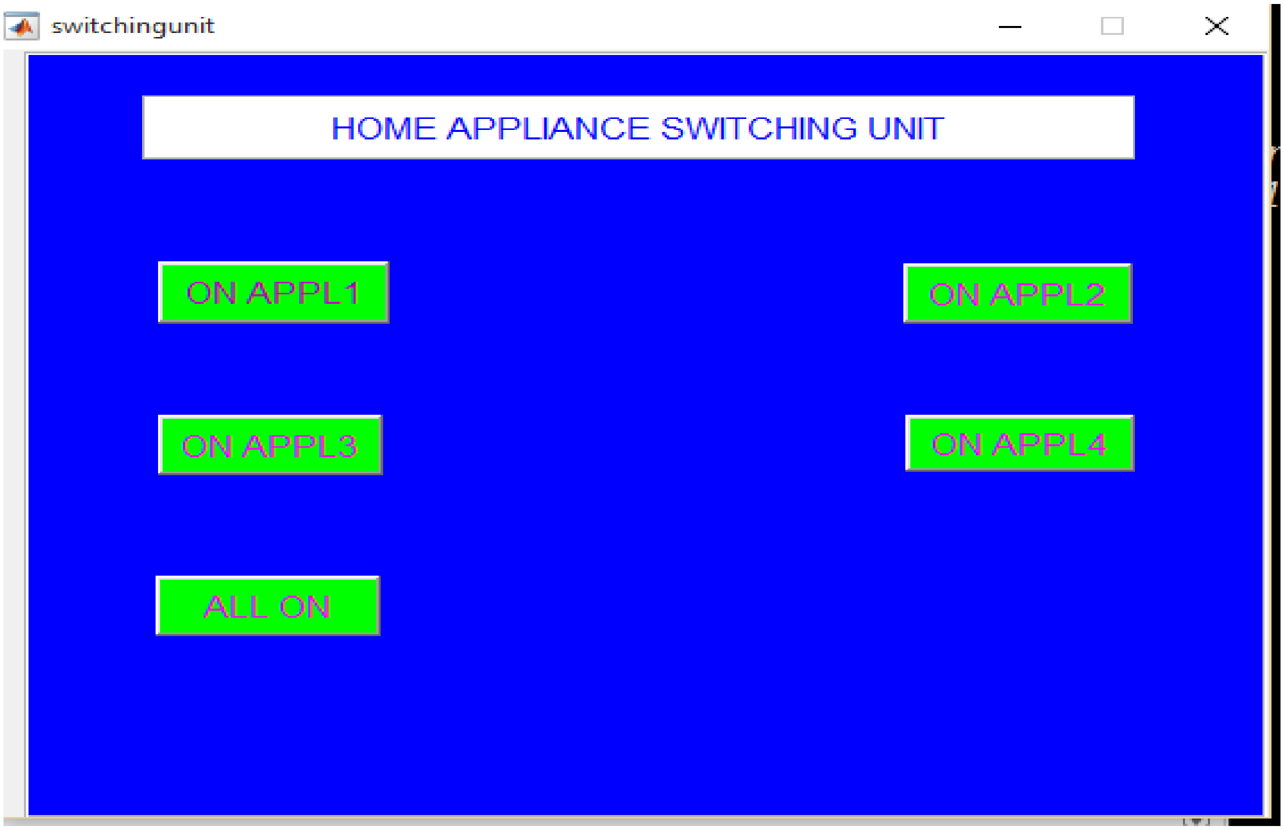


Figure 11: Appliance switching unit:

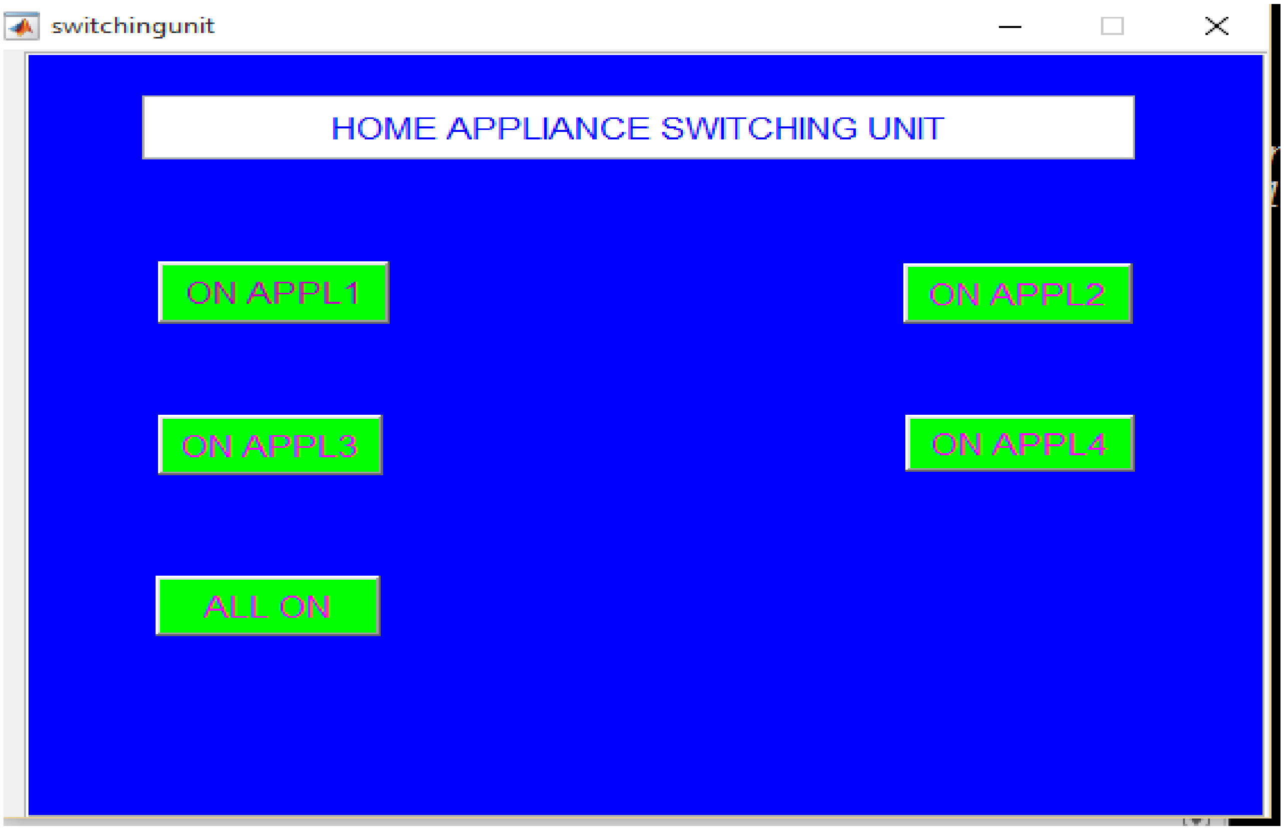


Figure 12: When all appliances are switched OFF

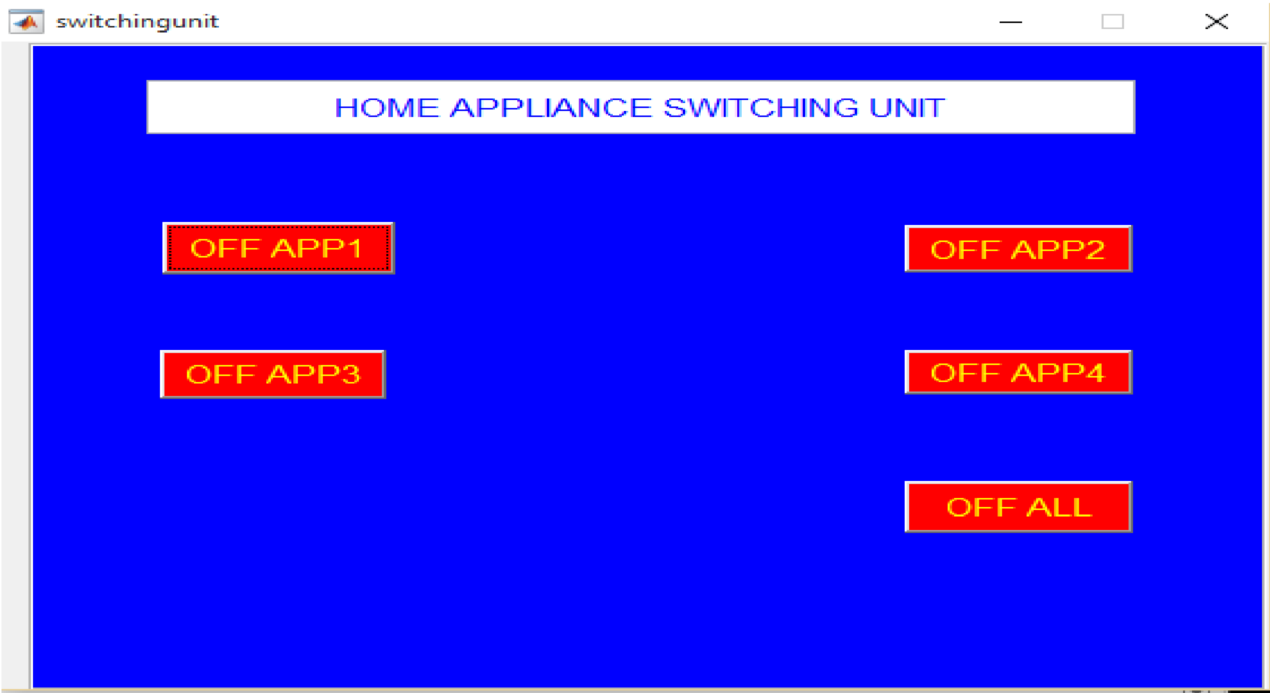


Figure 13: When all appliances are switched ON

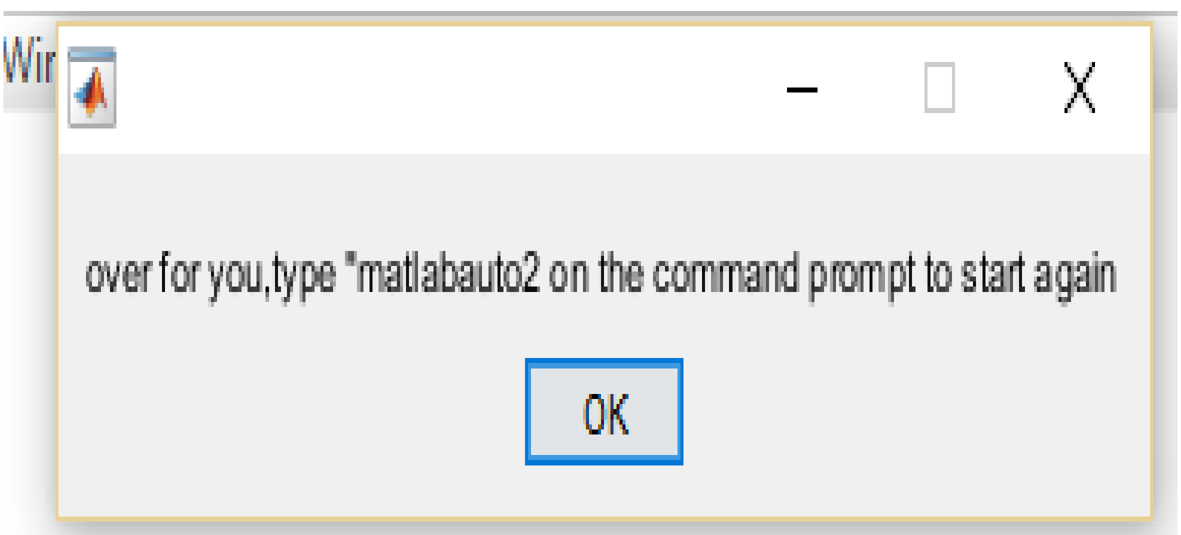
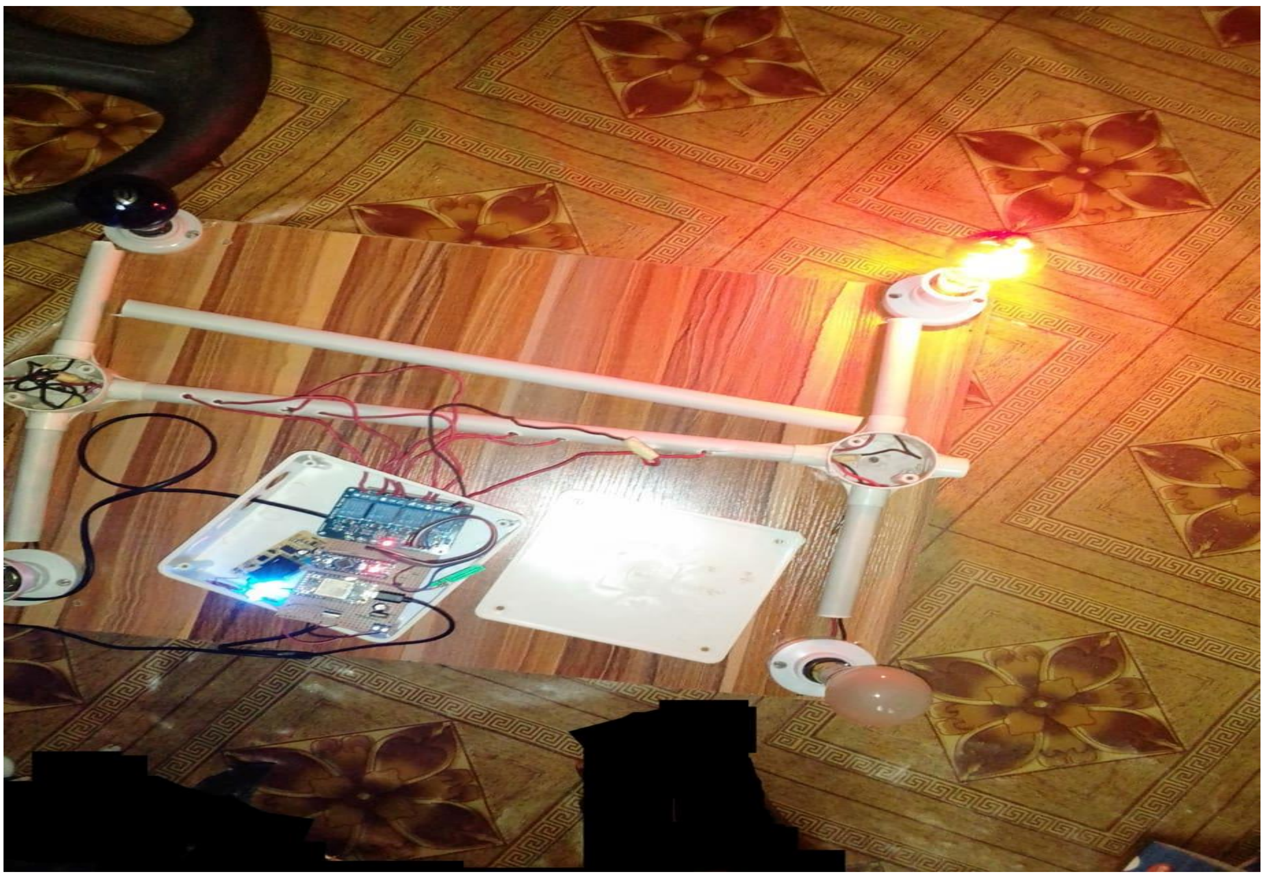


Figure 14: Incorrect password dialog box after three trials

The push buttons in the GUI is designed in such a way that when for example a button for ON1 is pressed, the button for OFF1 will pop up and vice-versa. The same is applicable to all the other buttons. Figure 13 shows the case where all the appliances were switched ON. Similarly, figure 12 shows the case where all the appliances were successively switched off.

4.4.2 The receiving section

This is the unit where the appliances to be remotely controlled are connected to. It comprises the receiving GSM module (which receives the texts sent by the transmitting GSM module and forward it to the controller), controller (which processes the texts sent and execute it by making its output pins HIGH or LOW depending on the command), four channel relay shield (which is used to switch on the AC appliances) and the appliances to be controlled (which comprises four different coloured incandescent bulbs (for demonstration purpose). The testing of the receiving section is shown in figure 15.



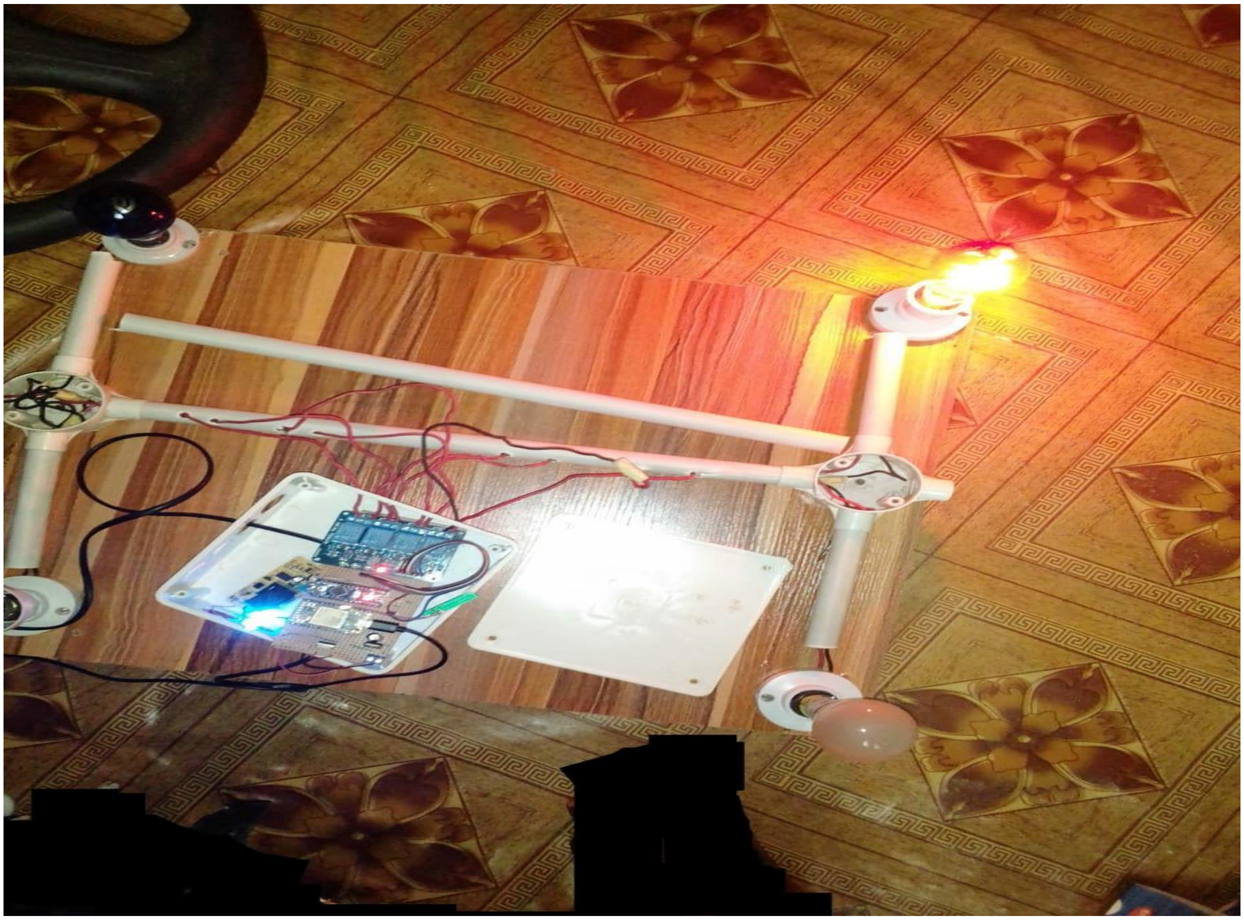


Figure 15: Receiving End Testing

4.5 Project Packaging

Packaging entails encasing the construction on the vero board in order to protect it. Packaging can be done using metallic, plastic or wooden materials.

In this project work, factors of portability, weight, cost and safety were considered and plastic material was used to package the construction. The plastics case was purchased from Electrical material shop to suit the packaging and the various circuit boards were firmly screwed to it.

4.6 Precautions

The following Precautions were taken during the project construction, testing and packaging

1. The datasheet of every equipment and components used during construction were duly consulted before handling them to avoid mishandling and damages.
2. Continuity test is always carried out after fixing each component
3. Soldering was carefully done to avoid short circuiting.
4. Every component was tested/debugged to prevent early failure
5. Soft lead was used for soldering for easy de-soldering if need be
6. Basic Electrical safety precautions were also adhered to

4.7 Components used in the Construction

Table 1: The Components used in the Construction

No.	Items	Quantity
1	Arduino Microcontroller	1
2	GSM Module	2
3	Relay Shield	1
4	USB-to-TTL Converter	1
5	Appliance (Bulbs)	3

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

During the course of this project work, there are lots of challenges that came up ranging from financial to components malfunctioning but to God be the glory, it was successfully tackled. There is no gain saying that the aim and objectives of this project work was achieved. The project was tested and found to work properly as per design. The project is very flexible in operation and can be controlled by those who has little or no knowledge in programming; this is made possible by the GUI. Furthermore, the initial cost of installing to this type of home appliance switching system could be high considering the fact that this requires a personal computer (PC) but the cost of maintenance will drastically reduce if installed. Despite being peculiar to home appliances, this work could also be employed in industrial setting.

5.2 Recommendations

This project work is peculiar to one way switching of home appliances only. With this in mind, it will be better if the following suggestions are incorporated in future design:

1. A feedback system could be incorporated in the receiving end to constantly monitor the state of the home appliances and forward same to the GUI for necessary decisions.
2. In case of system failure at the transmitting end, a backup system could be provided to keep the system in check.
3. Protective devices and sensors could be incorporated in the appliances to be controlled to constantly monitor Electrical parameters like power, current and voltage.

REFERENCES

Arduino – Introduction, Retrieved from, https://www.arduino.cc/en/guide/introduction.

Attaiah, I. and Hamdy, A. (2011).Energy Saving through Smart Home. *The Online Journal on Power and Energy Engineering* (OJPEE), 2(3), Pp. 223-227

Best Home Automation System: Consumer Reports. Retrieved from, https://www.consumerreports.org

Bhaskar P.R. and Uma S.K (2015). Raspberry Pi Home Automation with Wireless Sensor Using Smart Phone. *International Journal of Computer Science and Mobile Computing*, Vol.4 Issue 5, Pp. 797-803

Cleve, M. (2004). "The origins of MATLAB".

David, K. (2011). " The making of Arduino". IEEE Spectrum.

Dinesh, P., Kumar, D., Rahul, K. and Prathyush, C. (2013). Automation to Home Appliances through Voice Command *International Journal of Scientific Research*, 2(5), Pp. 100-103

ElShafee A. and Hamed K.A. (2012). Design and Implementation of a WiFi Based Home Automation System. *International Journal of Computer and Information Engineering* 6(8).

Ejeh, G. (2016). Design and Construction of GSM and Microcontroller Based Home Appliances Switching System. B.Eng. Project. Department of Electrical and Electronics Engineering, University of Agriculture, Makurdi. Nigeria. Pp. 10-36.

Encoders and Decoders – Introduction and Working with different ICs, Retrieved from, https://www.elprocus.com/encoders-and-decoders/.

Ghirardi, A.A. (1932). Radio Physics Course. 2^{e}, Rinehart Books. Pp.249. New York.

Gupta J.B (2014). A course in Electrical Technology: Electronic devices and circuits. 13^{e}, S.K Kataria and Sons. Pp. 135. New Delhi.

Harinath R. and Dr. Santhi S. (2015). GSM Based Home Automation System Using App-Inventor Android Mobile Phone. *International Journal of Computer Science and Mobile Computing*, Vol. 4, Issue 4, Pp. 158-167.

Hernando, B. (2016). *The untold History of Arduino*. Retrieved from, https://arduinohistory.githum.io

Hill, J. (2015). The smart home: a glossary guide for the perplexed.

James, G. (1999). Home Automation & wiring. 1[e], McGraw-Hill publisher co., Pp.338. London

Johri, E., Pradnya, B., Vidhi, T., Sheetal, G. and Vrusha, S. (2015). Remote Control Home Automation Using Android Application and WiFi Connectivity. *International Journal on Recent Innovation Trendsin Computing and Communication*, 3(3), Pp. 1489-1492

Kumar A. (2013). GUI Based Device Controller Using MATLAB *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 6.

Mirza, J., Muhammed, H., Bin, T.A. and Tamin, A.K. (2014). A Comparative Analysis on Home Automation Techniques.*Second International Conference on Artificial Intelligence, Modelling and* Simulation. Pp. 109-114

Naresh, B. Chakradhar, S. Krishnaveni, (2013). Bluetooth Based Home Automation and Security System Using ARM9. *International Journal of Engineering Trends and Technology* (IJETT) Vol. 4, Issue 9.

NazmulHasan, Abdullah, A.K., Nezan, U. and Abu, F.M. (2013). Design and Implementation of Touch Screen and Remote Control based Home Automation System. *Proceedings of 2nd InternationalConference on Advances in Electrical Engineering* (ICAEE). Pp. 347-352

Pankaj Jadhav ,Amit Chaudhari , SwapnilVavale. (2014). Home Automation using ZigBee Protocol. *International Journal of Computer Science and Information Technologies*, 5(2), Pp 1778-1780.

Parameshachar B., D. Sawan, K., GopyGoonesh, W., Hurry, T. and Gopal T. (2013). A Study on Smart Home Control System through Speech.*International Journal of Computer Applications*, 69(19), Pp. 0975-8887

Pawan, K. (2017). Home Automation using MATLAB and ArduinoIO. Retrieved from, https://www.engineersgarage.com/contribution/home-automation-using-matlab-and-arduinoio.

Pooja N., Pawar, Shruti, R., Nisha, P.S. and Varsha, V.W. (2016). A Home Automation UsingInternet of Things.*International Journal of Innovative Researchmin Computer and Communication Engineering,* 4(4), Pp. 6555-6563

Prasshanth, C. and Gaurav, S. (2014). Economic Home Automation Systems in Rural India for Senior Citizens.*IETE 45thMid Term Symposium on Broadband Technologies Services for Rural India*. MTS 14

Priyadarshni P. and Sharma S. (2014). GUI-MATLAB Based Home/Industrial Automation Using MCU89S52. *International Journal of Science and Research* (IJSR) Vol. 3 Issue 9, September.

Quan, Q. (2007). MATLAB GUI (Graphical User Interface) Tutorial for Beginners. Retrieved from, https://blinkdagger.com/matlab/matlab-gui-graphical-user-interface-tutorial-for-beginners/.

Rajdev D. (2011). Smart and Energy-Efficient Home Automation. *Undergraduate Project*.

Rita Yiman L., Herru Ching L., Yu, Chokei M., Tony Beiqi T. (2016). Sustainable Smart Home and Home Automation: Big Data Analytics Approach. *International Journal of Smart Home.* 10(8):177-198.

Rye, D. (1999). My life at X10. AV and Automation Industry e-Magazine. Archived form the original on September 30, 2014.

Saddam (2016). GUI Based Home Automation System using Arduino and MATLAB, Retrieved from, https://circuitdigest.com/microcontroller-projects/gui-based-automation-using-matlab-and-arduino.

Shafee, A. and Hamed, K. (2012). Towards a Wi-Fi based Home Automation Systems. *International Journal of Computer and Electrical Engineering*, Pp. 803-808

Syed Anwaarullah and S.V. Altaf (2013). RTOS based Home Automation System using Android. *International Journal of Advanced Trends in Computer Science and Engineering*, 2(1), Pp. 480-484.

Tharaniyasoundhari M., Brilly MS.S. S. (2015). Intelligent Interface based Speech Recognition for Home Automation Using Android Application. *IEEE Sponsored 2nd International Conference onInnovations in Information Embedded and Communication Systems* (ICIIECS).

The Mathworks (2018). “Company Overview” (PDF).

Vijaya, K.K. and Surender (2016). Industry Monitoring Robot Using Arduino Uno with Matlab Interface. *Adv Robot Autom* 5:150. Doi 10.4172/2168-9675. 1000150

What is Radio Frequency (RF)? – Definition from Technopedia, Retrieved from, https://www.technopedia.com/definition/5083/radio-frequency-rf.

APPENDICES

Appendix A: MATLAB code for password prompt

```
clc;
clear all;
delete(instrfind({'port'},{'COM3'}));
pass='mvs';
attempt=0;
while(attempt<3)
a=inputdlg('WELCOME TO HOME AUTOMATION WITH MATLAB**enter pass key** TO
LOGIN TO THE GUI:');
if(strcmpi(a,pass))
   switchingunit;
   break;
else
    attempt=attempt+1;

    if(attempt==3)
        msgbox('over for you,type "matlabauto2 on the command prompt to
start again');
        close all;
    end
end
end
```

Appendix B: MATLAB code for switching unit:

```
function varargout = switchingunit(varargin)
% SWITCHINGUNIT MATLAB code for switchingunit.fig
%      SWITCHINGUNIT, by itself, creates a new SWITCHINGUNIT or raises the existing
%      singleton*.
%
%      H = SWITCHINGUNIT returns the handle to a new SWITCHINGUNIT or the handle to
%      the existing singleton*.
%
%      SWITCHINGUNIT('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in SWITCHINGUNIT.M with the given input arguments.
%
%      SWITCHINGUNIT('Property','Value',...) creates a new SWITCHINGUNIT or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before switchingunit_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to switchingunit_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help switchingunit

% Last Modified by GUIDE v2.5 02-Sep-2019 11:53:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @switchingunit_OpeningFcn, ...
                   'gui_OutputFcn',  @switchingunit_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
```

```
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before switchingunit is made visible.
function switchingunit_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to switchingunit (see VARARGIN)

% Choose default command line output for switchingunit
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes switchingunit wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% Starting of main function
delete(instrfind({'port'},{'COM17'})); % delete everything on COM17
clear a;
global a; % declaring global variable for Serial communication
a=serial('COM17','BAUD',115200); % Serial communication on COM17
```

```
fopen(a); % Opening the Serial port



% --- Outputs from this function are returned to the command line.
function varargout = switchingunit_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in on1.
function on1_Callback(hObject, eventdata, handles)
% hObject    handle to on1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Callback function for ON button 1
global a;
   fprintf(a,'AT+CMGF=1'); % AT command for SMS enabled mode
fprintf(a,'AT+CMGS="+2349028781605"'); % AT command to send SMS
%change to receiving end phone number
fprintf(a,'on1');
fprintf(a,26); % ASCII CODE for CTRL+Z to show end of SMS
  set(handles.on1,'Visible','off'); % Make ON button 1 Invisible
  set(handles.off1,'Visible','on'); % Make OFF button 1 Visible

% --- Executes on button press in off1.
function off1_Callback(hObject, eventdata, handles)
% hObject    handle to off1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global a;
  fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'off1');
fprintf(a,26);
  set(handles.off1,'Visible','off');
  set(handles.on1,'Visible','on');

% --- Executes on button press in on2.
function on2_Callback(hObject, eventdata, handles)
% hObject    handle to on2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'on2');
fprintf(a,26);
  set(handles.on2,'Visible','off');
  set(handles.off2,'Visible','on');

% --- Executes on button press in off2.
function off2_Callback(hObject, eventdata, handles)
% hObject    handle to off2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'off2');
fprintf(a,26);
  set(handles.off2,'Visible','off');
  set(handles.on2,'Visible','on');
```

```
% --- Executes on button press in on3.
function on3_Callback(hObject, eventdata, handles)
% hObject    handle to on3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
  fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'on3');
fprintf(a,26);
  set(handles.on3,'Visible','off');
  set(handles.off3,'Visible','on');

% --- Executes on button press in off3.
function off3_Callback(hObject, eventdata, handles)
% hObject    handle to off3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'off3');
fprintf(a,26);
  set(handles.off3,'Visible','off');
  set(handles.on3,'Visible','on');

% --- Executes on button press in on4.
function on4_Callback(hObject, eventdata, handles)
% hObject    handle to on4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
```

```
fprintf(a,'on4');
fprintf(a,26);
  set(handles.on4,'Visible','off');
  set(handles.off4,'Visible','on');


% --- Executes on button press in off4.
function off4_Callback(hObject, eventdata, handles)
% hObject    handle to off4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'off4');
fprintf(a,26);
  set(handles.off4,'Visible','off');
  set(handles.on4,'Visible','on');


% --- Executes on button press in allon.
function allon_Callback(hObject, eventdata, handles)
% hObject    handle to allon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'allon');
fprintf(a,26);
  set(handles.allon,'Visible','off');
  set(handles.alloff,'Visible','on');
```

```
% --- Executes on button press in alloff.
function alloff_Callback(hObject, eventdata, handles)
% hObject    handle to alloff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
   fprintf(a,'AT+CMGF=1');
fprintf(a,'AT+CMGS="+2349028781605"');
fprintf(a,'alloff');
fprintf(a,26);
  set(handles.alloff,'Visible','off');
  set(handles.allon,'Visible','on');



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Appendix C: Arduino code

```
#include <na9.h>
A9Modem modem;
void setup() {
Serial.begin(38400);
modem.begin(&Serial);
for(int pin=2;pin<6;pin++)
  {
pinMode(pin,OUTPUT);
digitalWrite(pin,1);
  }
pinMode(13,OUTPUT);
modem.on_message([](const char* number, const char* message)
  {
    String msgg=String(message);
    String numBER=String(number);
msgg.trim();
numBER.trim();
if(numBER=="+2348168201475" )//08109523661
    {
Serial.println(msgg);
if(msgg=="on1" )
      {
digitalWrite(2,0);
      }
else if(msgg=="off1")
      {
digitalWrite(2,1);
      }
else if(msgg=="on2")
      {
digitalWrite(3,0);
      }
else if(msgg=="off2")
      {
digitalWrite(3,1);
      }
else if(msgg=="on3")
      {
```

```
digitalWrite(4,0);
       }
else if(msgg=="off3")
       {
digitalWrite(4,1);
       }
else if(msgg=="on4")
       {
digitalWrite(5,0);
       }
else if(msgg=="off4")
       {
digitalWrite(5,1);
       }
 if(msgg=="allon" )
       {
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(4,0);
digitalWrite(5,0);

       }
if(msgg=="alloff" )
       {
digitalWrite(2,1);
digitalWrite(3,1);
digitalWrite(4,1);
digitalWrite(5,1);

       }

     }
  });
}

void loop() {
modem.run();
```