**‘‘Wireless home appliances controlling system’’**

# *Abstract*

*The aim of this project is to control home appliances like fan, light, door and windows wirelessly using GSM Module without having to make physical contact. Compared to the old system the system has a great significance, if one can control every appliance of its home wirelessly using a text message from his mobile, it saves time, energy, and also the risk of different electrical dangers like electric shock since there is no physical contact. Cost saving of different switching materials that are being used in the old system is also saved and we avoid the risk of durability of the materials.*

# Acknowledgement

First of all, we would like to thank the almighty, the creator of the universe GOD. We would like our next gratitude to go to our department for providing us with necessary tools like computers and internet connection. We also like to thank our advisor Mr. Aravind R. for his great and unlimited support throughout our project. We don’t want to miss this chance to thank all of our instructors for their role on sharing the lifetime knowledge’s, Mr. Asrat G (Assis. Proff.), Mr. Firew D (MSc), Mr. Samuel , and our department head Mr. Teshome (MSc).

**Table of Contents**

[Declarations I](#_Toc10793778)

[*Abstract* II](#_Toc10793779)

[Acknowledgement III](#_Toc10793780)

[List of figures VI](#_Toc10793781)

[List of tables VII](#_Toc10793782)

[List of acronyms’ VIII](#_Toc10793783)

[List of appendices IX](#_Toc10793784)

[Chapter 1 1](#_Toc10793785)

[Introduction 1](#_Toc10793786)

[Statement of Problem 2](#_Toc10793787)

[Objective 2](#_Toc10793788)

[General Objective 2](#_Toc10793789)

[Specific Objective 3](#_Toc10793790)

[Scope and Limitation 3](#_Toc10793791)

[Significance 3](#_Toc10793792)

[Chapter 2 5](#_Toc10793793)

[Introduction 5](#_Toc10793794)

[Literature Review 5](#_Toc10793795)

[conclusion 7](#_Toc10793796)

[Chapter 3 8](#_Toc10793797)

[Introduction 8](#_Toc10793798)

[Methodology 8](#_Toc10793799)

[System Design 9](#_Toc10793800)

[Arduino Pin Description 10](#_Toc10793801)

[Working principle of relays 13](#_Toc10793802)

[LCD Pin Description 14](#_Toc10793803)

[System program flow chart 19](#_Toc10793804)

[overall system flow 20](#_Toc10793805)

[Used software tools 21](#_Toc10793806)

[Conclusion 21](#_Toc10793807)

[Chapter 4 22](#_Toc10793808)

[Introduction 22](#_Toc10793809)

[Experiment and Result 22](#_Toc10793810)

[Conclusion 32](#_Toc10793811)

[Chapter 5 33](#_Toc10793812)

[Introduction 33](#_Toc10793813)

[Recommendation and feature work 33](#_Toc10793814)

[Conclusion 35](#_Toc10793815)

[References 36](#_Toc10793816)

# List of figures

[Figure 1: fan picture 9](#_Toc10793743)

[Figure 2: system design block diagram. 10](#_Toc10793744)

[Figure 3: Arduino Uno and its pin configuration on proteus. 11](#_Toc10793745)

[Figure 4: real Arduino diagram 12](#_Toc10793746)

[Figure 5: GSM module on Proteus. 12](#_Toc10793747)

[Figure 6 : Relay 13](#_Toc10793748)

[Figure 7: block diagram on working principle of relay. 14](#_Toc10793749)

[Figure 8: LCD display and it’s pin configuration. 15](#_Toc10793750)

[Figure 9: bulb photo. 15](#_Toc10793751)

[Figure 10 : potentiometer diagram. 16](#_Toc10793752)

[Figure 11 : transistor block diagram. 16](#_Toc10793753)

[Figure 12: ULN2003A pin diagram of transistor array on Proteus. 17](#_Toc10793754)

[Figure 13: virtual terminal layout on Proteus. 17](#_Toc10793755)

[Figure 14: schematic drawing of our design on Proteus. 18](#_Toc10793756)

[Figure 15: flow chart of the program. 19](#_Toc10793757)

[Figure 16: system user experience block diagram. 20](#_Toc10793758)

[Figure 17: The password correct message. 22](#_Toc10793759)

[Figure 18: Failed attempt to control appliances without first providing password. 23](#_Toc10793760)

[Figure 19: - The result of command ‘0’ which is list of help. 23](#_Toc10793761)

[Figure 20: - The output of command ‘1’ to turn on the fan. 24](#_Toc10793762)

[Figure 21: - The output of command ‘2’ to turn off the fun. 24](#_Toc10793763)

[Figure 22: - Result of command ‘3’ to turn off the light. 25](#_Toc10793764)

[Figure 23: - The output of command ‘4’ to turn off the bulb. 25](#_Toc10793765)

[Figure 24: - Output of command ‘5’ to unlock the door. 26](#_Toc10793766)

[Figure 25: - Output of command ‘6’ to lock the door. 26](#_Toc10793767)

[Figure 26: - Output of command ‘7’ unlocking the window. 27](#_Toc10793768)

[Figure 27: - Result of command ‘8’ to lock the window. 27](#_Toc10793769)

[Figure 28: - Output of command ‘9’ with respective reply. 28](#_Toc10793770)

[Figure 29: - Output of command ‘a’ while status of all device is off. 28](#_Toc10793771)

[Figure 30: - Output of command ‘a’ where some device is already on. 29](#_Toc10793772)

[Figure 31: - output of command ‘z’ to turn off light/fan and close window/door. 30](#_Toc10793773)

[Figure 32: - output of command ‘z’ where some devices are already off. 31](#_Toc10793774)

[Figure 33: - Error notification message for wrong command entry. 32](#_Toc10793775)

[Figure 34: - The improved diagram with sound or buzzer added to the system. 33](#_Toc10793776)

[Figure 35: - Improved design of the system for feature work with neural network implemented in it. 34](#_Toc10793777)

# List of tables

[Table 1: pin configuration of relay. 13](#_Toc10793738)

[Table 2: Possible commands with their respective output. 21](#_Toc10793739)

# List of acronyms’

GSM: global system for mobile communications.

IoT: Internet of things.

LCD: liquid crystal diode.

GPRS: general packet radio service.

SMS: short message service.

GND: ground.

AREF: analog reference.

PWM: pulse width modulation.

IDE: integrated development environment.

# List of appendices

[Appendix 1: System program code. 37](#_Toc10793728)

[Appendix 2: budget requirement 53](#_Toc10793729)

[Appendix 3: Activity Schedule. 54](#_Toc10793730)

# Chapter 1

## Introduction

It’s not long time ago that people were using different mechanism of classical control system weather its wired or wireless. But in these projects since we are going to use wireless communication it’s better to look at the evolution of wireless technology of mobile communication 1G, 2G, 3G, 4G, and 5G. Looking past, wireless access technologies have followed different evolutionary paths aimed at unified target: performance and efficiency in a high mobile environment [1]. The first generation (1G) has fulfilled the basic mobile voice, while the second generation (2G) has introduced capacity and coverage. This is followed by the third generation (3G), which has quest for data at higher speeds to open the gates for truly "mobile broadband" experience, which will be further realized by the fourth generation (4G) [1]. The Fourth generation (4G) will provide access to wide range of telecommunication services, including advanced mobile services, supported by mobile and fixed networks, which are increasingly packet based, along with a support for low to high mobility applications and wide range of data rates, in accordance with service demands in multiuser environment [1]. This paper provides a high-level overview of the evolution of Mobile Wireless Communication Networks from 3G to 4G. Is described LTE (Long Term Evolution) a fourth generation (4G) mobile network technology and recently the china has introduced the 5G network [1]. 5G networks are digital cellular networks, in which the service area covered by providers is divided into a mosaic of small geographical areas called cells. Analog signals representing sounds and images are digitized in the phone, converted by an analog to digital converter and transmitted as a stream of bits.

Time after time the endless desire of human being to make a material communication or internet of things(IoT) brought us different wireless communication system like Bluetooth, GSM, 3G, 4G, 5G, radio frequency, wi-fi, ZigBee, etc. in our case we are going to use GSM module as a communication means between the owner of the house and the appliances whenever and wherever regardless of location.

GSM is The Global System for Mobile Communications (GSM) is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile devices such as mobile phones and tablets. It was first deployed in Finland in December 1991. By the mid-2010s, it became a global standard for mobile communications achieving over 90% market share and operating in over 193 countries and territories.

2G networks developed as a replacement for first generation (1G) analog cellular networks. The GSM standard originally described a digital, circuit-switched network optimized for full duplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via General Packet Radio Service (GPRS), and Enhanced Data Rates for GSM Evolution (EDGE).

Generally, the aim of our project is to enable a house owner to control all of its appliances over a 2G network without having to move physically to the location of the equipment. To do this we needed a communication protocol, since there are a lot of types of communication protocol we are going to use a GSM module because of its advantages like long range, fast, reliable (regardless of the factor listed in limitation section below), and so on.

## Statement of Problem

As we know currently almost all home appliances in our country are being controlled or turn on and off using manual method which is the person who wants to control the material needs to make physical contact with the equipment. During this process there are a lot of problems that could rise like electric shock, the labor used to operate, material damage because of use of excess energy, time wasted to physically operate the switch and so on.

The problems listed above are were in terms of the persona. There are also a lot of problems on the side of the equipment being controlled. For example, the door with ordinary lock can be easily manipulated by a person with the right tools and technique, reduces cost that will be spent on materials like wire, switches etc., in the previous system since the wire is mostly buried in the wall there is a high probability of starting fire.

If we see at the problem in terms of use, the old system has no way of reporting operation that is going in home like turning on the fan, the door being opened, the window being opened and also weather the light is on or off unless the person is physical present at the place. Assume that there is a guest coming and the host is far away from home taking the key along with himself, which means there is no way that the guest could possibly enters to the house unless he has the key. Again, assume that the person is sleeping and the temperature is increasing the only way to maintain the temperature is to turn on a fan, in order to do this the person has to get off of his bed and then turn on the light and move to the ventilator to turn it on, while doing so there may be electric shock while turning on the equipment’s.

The problem of the old system in terms of security is that when someone opened house door, or window or turn the light on or off, or the fan is on while it’s not required to be without owner intention there is no mechanism to notify owners that weather the fan is on/off, the light turned on/off, the window is opened or the door is opened.

So, the system we are providing can solve many problems like we have listed above. In order to do this, we are using a wireless communication protocol to control the home appliances like fan, door, window and light regardless of the distance between the owner and the house.

## Objective

### General Objective

The general objective of this project is to communicate with home appliances like fan, door, window, and light wirelessly for the ease of control using mobile phone via SMS.

### Specific Objective

The specific objectives of this project are listed below as following:

* Switching lights on and off using text message.
* Switching the fan on and off by using a short message service.
* Opening door using text message.
* Locking door using SMS.
* Opening window with phone.
* To be aware of what’s going on in our house.
* To be notified whenever the light is on/off.
* To be notified whenever our door is opened or closed.
* To be notified whenever our window is opened or closed from anywhere.
* Reduce the electrical shock that could possibly happen in the previous system.
* Reduce the risk of the durability of objects.
* Reduce expenses related to wire and different switching materials.
* Eliminating the risk of fire from buried wire in the wall that may become short and start fire.
* To eliminate accidental energy waste because of a load that hasn’t been turned off.
* To control or on and off appliances regardless of distance.

## Scope and Limitation

Actually, the scope of this project is not limited to just control home appliances, we can also use it in different places like office, different institutions, supermarket, industry, etc. in general we can use the system presented in a place where there is a GSM network and electrical power regardless of the location of the controller and the appliances to be controlled. We can also use this system in different places that are out of reach of human to be operated manually for example a light in a chemical tanker, a hole, or a large tower that have risk of falling, etc.

Even though there is an enormous advantage of this project there is still some limitations. The limitations of the system are not because of the design, but they are because of different reason that are out of control of the designers (us). Those factors include the ISP (Internet Service Provider) since they are in control of the network that is being used by GSM Module, the other limitation rises from the user side by carelessness of the user to charge his phone, and the other factor is the limitation of the technology we have for example the GSM uses a 2G network which is low in speed, the battery capacity that we have currently is not efficient, the electric power provider also have a great role in order for the system to work effectively, failure to do so will categorize it under the list of factors resulting the limitation.

## Significance

Since the project we are proposing is going to solve all the problem listed in the section statement of problem, we can observe that its significance is great. Being able to operate our home appliance’s from wherever we are regardless of the distance between us and home is the closest thing we have to the magic. Although the other great significance of the system is that we can receive the information about our home like when the fan is on/off, the door is opened or window is opened or light is turned on without having to be at home. This are regarding ease of life.

The significance of the project can also be seen in terms of safety is that if we can control our home appliances wirelessly without having to make physical contact we can reduce the risk of electrical shock, material damage, labor expense, and also different fire related problems that could result while using the old system.

Our system also has great advantages regarding the security of one’s house. As we have mentioned in statement of problem section above; in old system there is no mechanism to identify weather the fan is on/off, the door/window is locked or opened, or whether the light is on/off without being present to the place or location of appliances. Which means there will not be energy wasting when the owner is not around [8].

Generally, we strongly recommend every house, offices, institutions, industry, supermarkets, etc. to the system we have proposed. Because once they used this system the user experience they will have toward their machine’s(appliances) is going to be completely changed from, like slave to master.

# Chapter 2

## Introduction

As we have mentioned earlier our project is to control our home appliances using one of wireless communication system called GSM. But its clear that there are a lot of researcher who have also worked in this interesting area to automate home appliances regardless of most of the limitation. So, before we rush into design and different process of this project we have realized that it’s necessary to review different literature in the area of field related to our project. Because it’ll help us understand the working of the system more but also after we have reviewed those research’s we have seen a lot of gaps and limitation that has to be improved and we have tried our best to improve them. The reviewed literature and their limitation has been discussed roughly on the next section below.

## Literature Review

Since we are not the first peoples to conduct a project in home automation field we have tried to look at the related project in order to understand the working principle of the system, the limitation, the strength and every information we could get. So, we have tried to summarize the paper we have seen on internet related to the project as following.

Delgado, Picking, and Grout (2006) consider the problems with the implementation of home automation systems. Furthermore, the possible solutions are devised through various network technologies. Several issues affecting home automation systems such as lack of robustness, com-partibility issue and acceptability among the old and disabled people are discussed. [5]

Ciubotaru-Petrescu, Chiciudean, Cioarga, and Stanescu (2006) present a design and implementation of SMS based control for monitoring systems. The paper has three modules involving sensing unit for monitoring the complex applications. A processing unit that is microcontroller and a communication module that uses GPRS modem or cell phone via serial port RS-232. The SMS is used for status reporting such as power failure. [7]

In this paper, Conte and Scaradozzi (2003) view home automation systems as multiple agent systems (MAS). In the paper home automation system has been proposed that includes home appliances and devices that are controlled and maintained for home management. The major task is to improve performance. [6]

In this paper, Alkar and Buhur (2005) propose an Internet Based Wireless Home Automation System for Multifunctional Devices. This paper proposes a low cost and flexible web-based solution but this system has some limitations such as the range and power failure.

Murthy (2008) explores primary health-care management for the rural population. A solution proposes the use of the mobile web-technologies providing the PHC services to the rural population. The system involves the use of SMS and cell phone technology for information management, transactional exchange and personal communication. [2]

Jawarkar, Ahmed, Ladhake, and Thakare (2008) propose 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. [4]

Potamitis, Georgila, Fakotakis, and Kokkinakis, G. (2003) suggest the use of speech to interact remotely with the home appliances to perform a particular action on behalf of the user. The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition. [3]

The work of L. Muhury and A.H.M.A Habib describes the design and implementation of a DTMF-based home automation system. The user calls a SIM number assigned to the home and presses the digits on their phone’s keypad to control the home’s devices by generating a DTMF tone. The tone is received and decoded by the GSM module at home using a DTMF decoder. The decoded instructions are passed to the microcontroller so that user commands can be implemented at home.

Home automation systems using DTMF are not very commonly implemented, maybe because there are other better options for communication available. Like all other systems, DTMF-based home security systems also have their security flaws. They are vulnerable to “fuzzing attacks,” as described by R. Sassi. In a fuzzing attack, a user exploits a vulnerability in DTMF processing algorithms by giving unusual input data, which results in triggering an exception. This could cause the entire home network to crash.

The work of A.Z. Alkar and U. Buhur implements a home automation system using Internet for enabling remote home access and infrared technology for device communication within the home. Internet or IP protocol-based communication in home automation systems is always a popular choice among researchers. The Internet is easily scalable, flexible when it comes to access and use, and very popular as a communication method in today’s world, so the hardware and the network required for access is readily available, offers high bandwidth and very low communication cost, and devices can connect to and disconnect from the network easily. These are some of the features that make t h e Internet such an attractive choice for researchers. Utilizing the Internet as a means to access and control the home seems to be the next logical step forward for home automation systems. From an end user’s point of view, using Internet to access their home is easy, convenient, cheap, flexible, and offers no complication of an added technology to learn. User interface devices like laptops, smartphones, PCs, and tablets are easily available in the market, and these devices are already a part of people’s daily lives. So, incorporating home automation into these already-popular user devices seems to be the natural progression.

The work of M. Gauger et al. proposes a decentralized approach to home automation control. They implement the decentralized approach by integrating actuators into the WSN of the home. The authors propose a distributed control or process architecture. The information from the sensors are received and processed by one or more control nodes, which in turn initiates the appropriate actuators to change or control the environment as previously specified by the user. The system thus eliminates the need for a central controller.

The work of N. Sriskanthan et al. shows the implementation of a home automation system using Bluetooth. They use a host controller implemented on a PC, which is connected to a microcontroller- based sensor and device controllers. The researchers even built a new protocol on top of the Bluetooth software stack, called Home Automation Protocol (HAP), to make the communication between devices possible. The device controller is connected to electronic devices through t h e I2C Bus. The system allows more than one device controller to be connected to the host controller.

The work of H. Kanma et al. also proposes a home automation system using Bluetooth that can be accessed remotely through GPRS. The researchers use a cellphone equipped with Bluetooth connectivity as a host controller and a GSM modem that provides Internet connectivity. Home devices are fitted with Bluetooth communication adapters so that they can communicate with the host controller phone via Bluetooth. The paper discusses remotely controlling and updating home devices along with fault diagnostics and detection. The work also talks about providing an electronics user manual on the phone using Bluetooth and Internet.

## conclusion

Even though we have reviewed the above papers, we can assure you that there is no single circuit design or methodology we have taken from them. Everything we are going to develop; all the design of the architecture, code, and algorithm is our own except those which are only patented and licensed to the public. Being said that we will also try to mention our references since those researches are public and they deserve the credit for the job they have done to make a work easier for home automation systems.

# Chapter 3

## Introduction

In this chapter we will try to discuss the approach we have followed to complete our projects by discussing about the methods we have used to gather data, to assemble our circuit, to write our program from start to end step by step. In order to do so we have used different diagrams as well as tables and figures to clearly show the procedures we have been through as understandable as possible. We have also tried to discuss all required materials as brief as possible including their pin configuration and other necessary data by referring to their manual or documentation provided by either the vendor of the tools or other third parties.

## Methodology

The general approach used to design system is not limited. Which means we have combined a different field of knowledge to design the system. For example, in order to choose the microcontroller, we had to look and study different available options of microcontroller, since there are thousands of them on the market, with their properties and we found that using Arduino Uno is better because of its flexibility and other different features. To be able to develop the program for the microcontroller on Arduino IDE we had to deeply understand different libraries that can let us communicate with our devices using high level programming language, in our case C Programming language, these libraries include liquidCrystal.h, softwareSerial.h we will discuss them deeply later. Like we have already discussed in introduction section of this report we were also required to select the better wireless communication protocols because as we now nowadays there are a lot of wireless communication system that can help us in IoT system like this one to communicate with things for example, Bluetooth, wi-fi, ZigBee, and GSM. From the listed communication protocols, we have selected GSM modules mainly because of its flexibility of location and distance.

Not only the hardware part but also there are a lot of software tools that are used during this project for the purpose like data collection, data analysis, to generate different charts, document writing, diagram drawing and so on. As we have tried to discuss in the above paragraph for the system design too we have used some tools like Proteus 8.7, and Arduino IDE. These tools are like brave browser, MS excel, MS word, EdrawMax, Internet download manager and so on.

For data collection since most of our source is an internet in order to get access to the internet we needed to use a browser. Specifically, the browser we have used is named “brave” based on chrome browser because of its ad blocking and tracker blocking features. The downloader we have used was a trial version of an “Internet download manager” because it’s downloading speed is outstanding compared to the other downloaders either third party downloader or browser downloader.

As we know currently Microsoft products like MS excel and MS word plays a great role in different project report writing. We also have taken advantage of their enormous functions to complete our project report and proposal writing. We have used MS excel for timeline Gantt chart generating purpose because it can generate automatically by only taking the list of date and some script as input. As we know MS word is well known for its amazing word processing ability. We have also used it to write both proposal and report for our project. There were also mind mapping technique known as flow chart that we have used to map or visualize our idea, then to draw those maps or to construct those diagrams we have used a tool named EdrawMax.

Since our system design is based on simulation only we have used some tools that are used to simulate an electric design virtually on a computer. One of these tools is known as Proteus and it’s used to design electric circuit on computer and simulate them to see the output on our monitor. Alongside Proteus we have used Arduino IDE (integrated development environment) to write the code which is a core to our system on which we instruct our Arduino and GSM to work according to the signal received.

## System Design

In this section we will discuss the overall procedure of system design, tools and equipment used to achieve our goal. The tools that will be used in our system are fan, micro controller (Arduino Uno), GSM module, relays, bulbs, resistors, transistor, LCD, motor-based door and window lock, and phone. Let’s take a look at each of them roughly;

Let’s try to discuss detailly the components of the above block diagram as following;

**Fan**: - is a rotating and powered machine used to create flow within a fluid, typically a gas such as air. Consists of a rotating arrangement of vanes or blades which act on the air. The rotating assembly of blades and hub is known as an impeller, a rotor, or a runner. Usually, it is contained within some form of housing or case. The figure shown below.

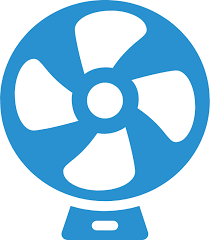


Figure 1: fan picture

**Power Supply**

**Arduino Uno**

**GSM module**

**Message from User**

**User send message**

**Transistor**

**Transistor**

**Transistor**

**Transistor**

**Relay**

**Relay**

**Relay**

**Relay**

**Door**

**Window**

**Fan**

**Bulb**

Figure 2: system design block diagram.

**Arduino Uno**: - is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits. Arduino is a tool for making computer that can sense and control more of physical world than desktop computer. It’s an open source physical computing platform based on a simple micro controller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking input from a variety of switches or sensors and controlling a variety of light, motors and other physical outputs. Arduino projects can be stand alone or they can be communicating software running on our computer. The Arduino programming language is an implementation of wiring, a similar physical computing plat form, which is based on the processing multimedia programming environment.

## Arduino Pin Description

1. **GND (3):** stands for ground. There are many ground pins on Arduino to connect network to the ground.



Figure 3: Arduino Uno and its pin configuration on proteus.

Now let’s take a look at the pin configuration of Arduino Micro controller as following;

1. **6(A0-A5):** Are stands for analog input pins. These pins can read a signal coming from analog sensors and convert it to digital value that the Arduino microcontroller can understand (read).
2. **7(0-13):** Digital output pin. This is used for both digital input like (telling if a button is pushed) and digital output like (powering LED).
3. **AREF:** Stands for analog reference (9) most of the time you can leave this pin alone. sometimes it used to set an external reference voltage between 0 and 5v as upper limit for analog input pins.
4. **PWM (8):** A digital pins (3, 5, 6,9,10 and 11). These pins are acts as normal digital pins but, they can also be used for pulse width modulation (PWM).
5. **5V (4) and3.3V (5):** 5V pin supplies 5V of power and 3.3V pin supplies 3.3V of power.

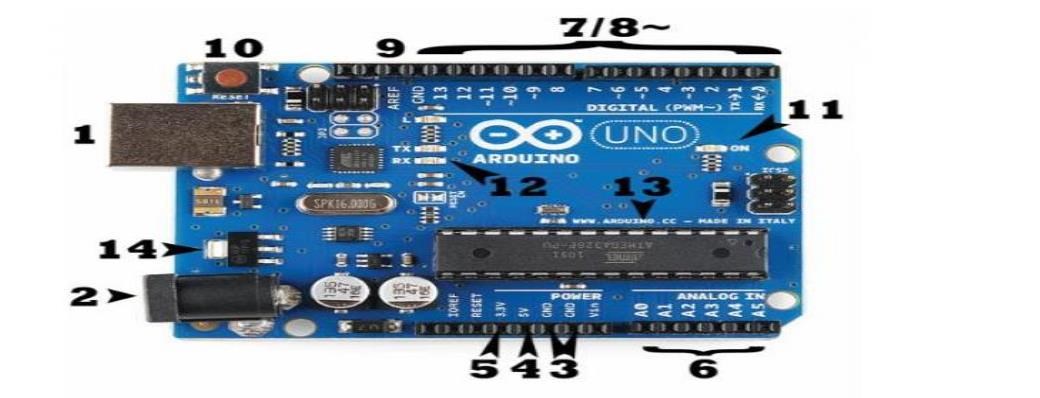


Figure 4: real Arduino diagram

**GSM module: -** A GSM module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The GSM stands for global system for mobile communication. It is nothing but, the system which accepts SIM CARD and operates over the desired user. This GSM modems interfaced or connected to a computer. The GSM modem frequently used to provide mobile network connectivity for sending and receiving SMS and MMS messages. To perform this operation the GSM modem must support an extended AT command sets for receiving and sending SMS messages.



Figure 5: GSM module on Proteus.

The GSM module requires a 12V input DC but there may be different voltage sources like a 15V and 5V. In this project the GSM module has directly powered by 5V output from Arduino booting up the GSM.

**Pin configuration of GSM**

The GSM module has 2 pins namely TX and RX which used for serial communication and it’s used to be interfaced with other devices that are capable of communicating serially like Arduino. To interface it with Arduino we have to connect TX and RX of GSM with RX and TX of GSM respectively. But in our case since we are going to simulate our design virtually we will use a tool named virtual terminal on Proteus and connect TX of the terminal to RX of the GSM and TX of the GSM to RX of the Arduino.

**Relay**: -is an electrically operated switch. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal.

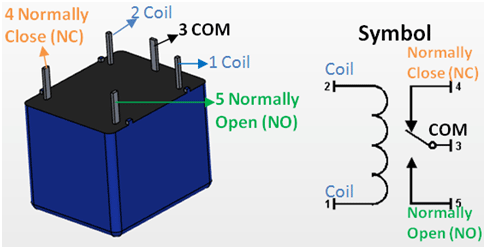


Figure 6 : Relay

**Pin configuration of relay**

Let’s try to summarize the pin configuration of the relay in the following table:

Table 1: pin configuration of relay.

|  |  |  |
| --- | --- | --- |
| R.NO | Pin name | Pin description |
| 1 | Coil end 1 | Used to trigger (on/off) the relay, normally one end is connected to 5V and the other end to ground. |
| 2 | Coil end 2 | Used to trigger (on/off) the relay, normally one end is connected to 5V and the other end to ground. |
| 3 | Common (COM) | Common is connected to one end of the load that is to be controlled. |
| 4 | Normally close(NC) | The other end of the load is either connected to NO or NC. If connected to NC, the load remains connected before trigger. |
| 5 | Normally open(NO) | The other end of the load is either connected to NO or NC. If connected to NO, the load remains connected before trigger. |

## 

## Working principle of relays

All relays contain a sensing unit, the electric coil, which powered by AC or DC current. When the applied current or voltage exceeds a threshold value, the coil activates the armature, which operates either to close the open contacts or to open the closed contacts. When a power is supplied to the coil, it generates a magnetic force that actuates the switch mechanism. The magnetic force effect is relaying the action from one circuit to another. The first circuit is called the control circuit.

Low Power Input

High Power Circuit

Figure 7: block diagram on working principle of relay.

There are three basic functions of a relay:

* **On/Off Control:** Example: Air conditioning control, used to limit and control a “high power “load, such as a compressor.
* **Limit Control:** Example: Motor Speed Control, used to disconnect a motor if it runs slower or faster than the desired speed.
* **Logic Operation:** Example: Test Equipment, used to connect the instrument to a number of testing points on the device under test.

Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Which are being interfaced with thyristors like SCR and TRIAC.

**LCD**: - A liquid-crystal display is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. [10]

**Reflective:** it uses external light reflected by reflector behind the display. Example: watch, Calculator, this is achieved by combining a reflector with rear polarizer.

**Backlight:** light source is from a back light and viewed from the front. Example: computer display built in fluorescent tubes above, besides and sometimes behind the LCD.

## LCD Pin Description

1. **Pin 1(VSS):** is a ground pin and it is certainly needed that this pin should be grounded for the LCD to work properly.
2. **VDD and VEE:** they are pin2 and pin3 and given to +5v normally. However, VEE may have a potentiometer voltage divider network to get contrast adjust. But, VDD is always at +5v.
3. **RS, R/W and E:** these three pins are numbered 4, 5 and 6 as shown above on LCD panel. The RS used to make the selection between data and command register. For RS = 0 the command register is selected. Whereas for RS = 1 the data register is selected. R/W gives us the choice between writing and reading mode. If R/W is set (R/W = 1) reading is enabled. If reset (R/W = 0) the writing mode is enabled.



Figure 8: LCD display and it’s pin configuration.

1. **D0-D7:** 8-bit data pins. Those are used to send information to the LCD or read the contents of the LCD internal register.

**Bulb**: - is the glass part of an electric lamp, which gives out light when electricity passes through it. [11]



Figure 9: bulb photo.

**Potentiometer:** - In our project potentiometer is used to provide us with a variable resistance while supplying source to our circuit while helping us adjust the contrast of LCD by connecting it to VEE pin.

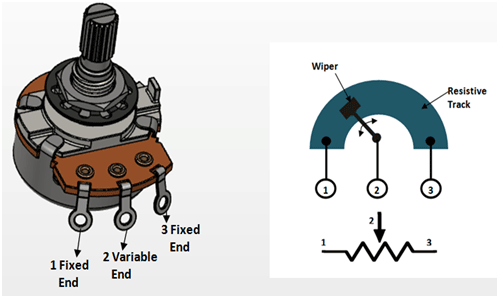


Figure 10 : potentiometer diagram.

**Transistor**: - A transistor is a semiconductor device used to amplify or switch electronic signals and electrical power. It is composed of semiconductor material usually with at least three terminals for connection to an external circuit. For this project we are going to use array of transistor named ULN2003A. The ULN2003A devices are high-voltage, high-current Darlington transistor arrays. Each consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads.

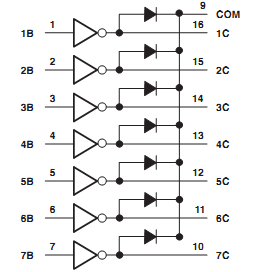


Figure 11 : transistor block diagram.

But in Proteus schematic drawing this specific type of transistor looks like the following.



Figure 12: ULN2003A pin diagram of transistor array on Proteus.

**Electronic door and window lock(motor):** - are a way to replace keys or to add additional automation features, like remote locking or unlocking. We will use a type of motor which will turn either clockwise or anti clockwise when either high or low is received.

**Serial Communication (Virtual Terminal): -** The serial communication from the GSM to Arduino microcontroller are done by connecting virtual terminal TX is to the GSM RX. The GSM TX pin is connected to Arduino Uno RX pin.



Figure 13: virtual terminal layout on Proteus.

As we have tried to design the system on Proteus for simulation purpose we will try to show it using picture as following;



Figure 14: schematic drawing of our design on Proteus.

The description of the connection is as following first the virtual terminal TX is connected to the GSM RX. The GSM TX pin is connected to Arduino Uno RX pin. We have initialized another serial communication on Arduino pin 12 and 13 to enable us add other virtual terminal. On this virtual based design, we don’t need to worry about power connection since Proteus will provide us an internal connection of both ground and power but in real implementation we have to connect GSM VSS pin to Arduino 5V output and GSM GND to Arduino GND.

Four pins of Arduino pin number 2, 3, 4, 5 are connected to the ULN2003A transistor pi of 4,3,2,1 respectively and the pin number 13, 14, 15, 16 are connected to the coil 2 of the relay R1, R2, R3, R4 respectively. On the load side AC voltage is used and one end of the AC voltage terminal is connected to the load and one side is grounded alongside with the NC terminal of the relays. The other end of the loads is connected to the NO open part of the relay.

The other connection on the above circuit is LCD interface with Arduino. The pin number 8, 9, 10, and 11 of Arduino is connected to the pin D4, D5, D6, And D7 of the LCD respectively. VDD Pin of the LCD is connected to 5V supply and VEE Is connected to potentiometer network to adjust the contrast of our LCD.

## System program flow chart

As we have mentioned earlier in order to make our program work we need to develop an algorithm in which it can performs all the functions that we are intended to accomplish through our device. This program will be written on Arduino IDE, because we are using a microcontroller named Arduino Uno to communicate devices and a GSM module. The language we are going to use is a C programming language it’s suggested by the Arduino vendor because of its robust feature like simplicity. But, before writing any code we have to first map the algorithm. There are actually a lot of technique to map algorithm like, pseudocode, and flow chart. In this case we are going to use a flow chart to show the process that the program will be through as shown in figure 3.

**Notify thewrongsender**

**Start**

**Receivemessage**

**Check sender**

**correct/incorrect**

**Identify theoperation**

**Load device asdescribed on UXflowchart**

**Show status onLCD**

**Terminate**

**Display error onLCD**

**correct**

**incorrect**

Figure 15: flow chart of the program.

## overall system flow

The system user experience diagram is shown on figure 2 below. overall process of the system combined with the user and also the interaction that will take place between the device and the user.

on

off

no

no

**Start**

**Recievemessagedata**

**identifymessage**

**is messageon/off**

**Show messagerecieved on LCD**

**Check ifdevice isOFF**

**Check ifdevice is ON**

**yes/no**

**yes/no**

**Turn ON thedevice**

**Send device isalready ON messae**

**send the device isalready OFFmessage**

**Turn OFFthe device**

**Send the divece isturned OFFmessage**

**Send the device isturned ONmessage**

**Terminate**

**Show the statuson the LCD**

**Show the status onthe LCD**

yes

yes

Figure 16: system user experience block diagram.

Basically, the system operates by accepting commands from the user and act based on the type of commands using the program written. Let’s see available commands as following;

Table 2: Possible commands with their respective output.

|  |  |  |  |
| --- | --- | --- | --- |
| **R.NO.** | **Command.** | **Function of the command.** | **Result of the command.** |
| 1) | ‘1’ | Used to turn on the fan. | Fun turned on! |
| 2) | ‘2’ | Used to turn off the fan. | Fun turned off! |
| 3) | ‘3’ | Used to turn on the light. | Light turned on! |
| 4) | ‘4’ | Used to turn off the light. | Light turned off! |
| 5) | ‘5’ | Used to unlock the door. | Door unlocked! |
| 6) | ‘6’ | Used to lock the door. | Door locked! |
| 7) | ‘7’ | Used to unlock the window. | Window unlocked! |
| 8) | ‘8’ | Used to lock the window. | Window locked! |
| 9) | ‘9’ | Used to check device status. | “Replies according the status”. |
| 10) | ‘0’ | Used to show the help manual. | “Shows possible commands with their function”. |
| 11) | ‘a’ | Used to turn on the all the equipment. | “Replies according to the result”. |
| 12) | ‘z’ | Used to turn off all the equipment. | “Replies according to the result”. |
| 13) | ‘x’ | To enable the system for the person who sent the letter x. | Access granted! |

We will discuss commands listed above detailly in chapter 4 result section.

## Used software tools

Actually, now a day everything is computable virtually so we are going to use a software called Proteus to develop the circuit and simulate. We are also using an Arduino programmer or Arduino IDE to write code for the system. The language we are using will be C programming language. Beside Proteus and Arduino IDE, we have also used a software like MS Excel to generate a time line in form of Gantt chart, MS word to write document, and EdrawMax to draw different flowchart’s and mind maps regarding our project.

## Conclusion

We hope that in these chapter we have clearly stated almost everything’s that are related our overall design which is the material used, the configuration of each materials, tools used, the flow of our program, structure of our circuit, and so on. We have also seen the possible commands to control our home appliances with their respective functions and the result that is expected to be sent to the owner and be displayed on LCD.

# Chapter 4

## Introduction

Now we have already designed our system and now it’s time to test our circuit for connection errors, as well as our code for bugs. Then, we will see what the result of our project will look like on different input and different conditions. For each of command and conditions we have taken the screen shoot of the output by snipping tool provided by Windows.

## Experiment and Result

Now we are going to look at each of our results by using all the 12 (13) including password commands one by one with their respective screenshot starting from ‘0’ to ‘z’.

**Command ‘x’:** - we have used this command to lock and unlock the whole system. Without first providing this command the user or any other third party cannot access any equipment. When we sent x to the system it will reply us with the “password correct” message as shown below in the next figure 17.

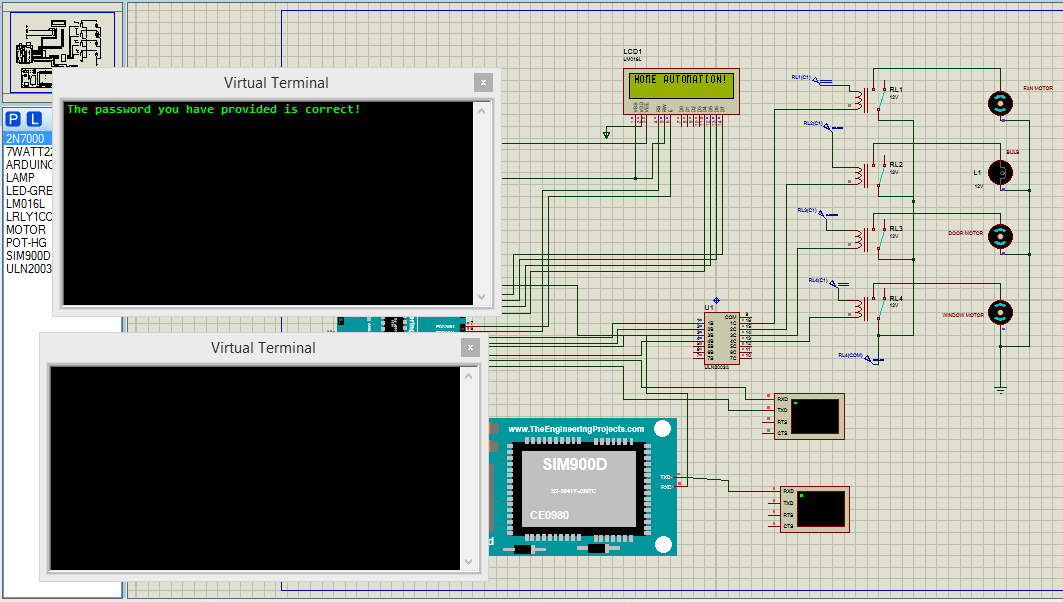


Figure 17: The password correct message.

But if the user tried to access any appliances to control them without first providing the pass word then the system will show the wrong password error notification and give hint on what to do. This is shown on the next figure 18 below as following.

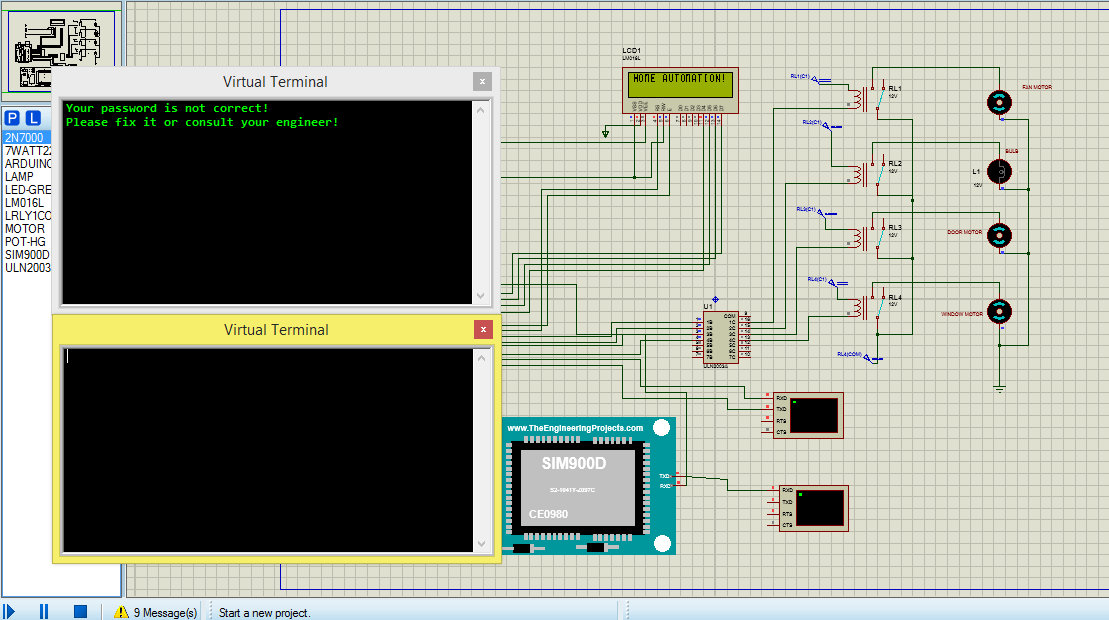


Figure 18: Failed attempt to control appliances without first providing password.

**Command ‘0’: -** is used to show available commands to be used on the system. When the user sends ‘0’ to the GSM module then the micro controller will reply back to the sender with information of the available commands as a help manual through GSM module. Figure 19 is the result of this command.

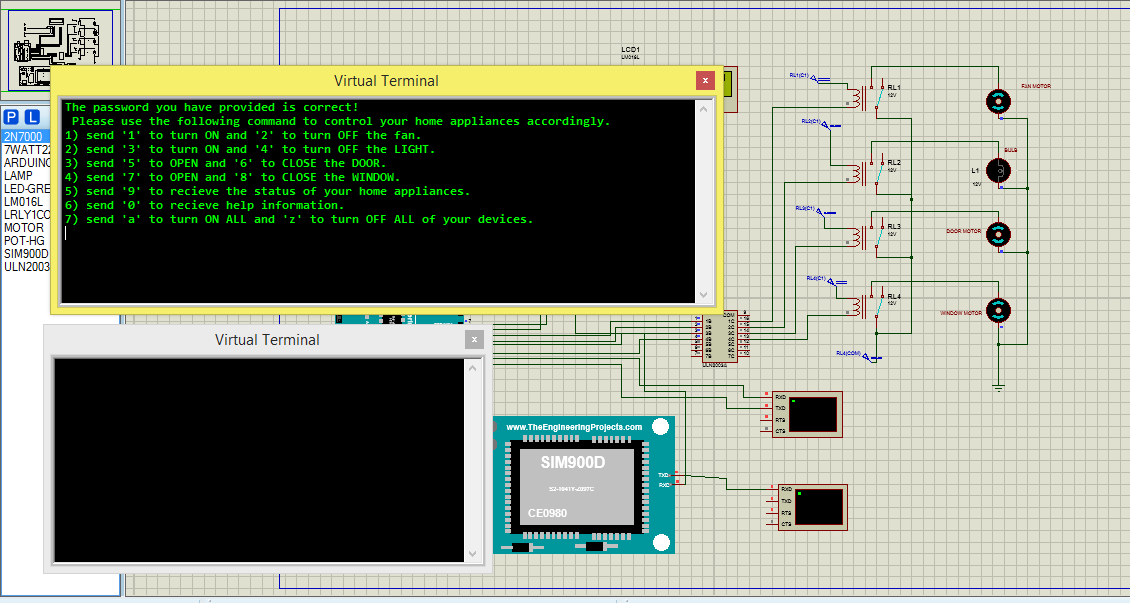


Figure 19: - The result of command ‘0’ which is list of help.

**Command ‘1’:** - used to control the fan which sends a HIGH signal to the relay in order to turn on the fan. Figure 20 is the result of this command.

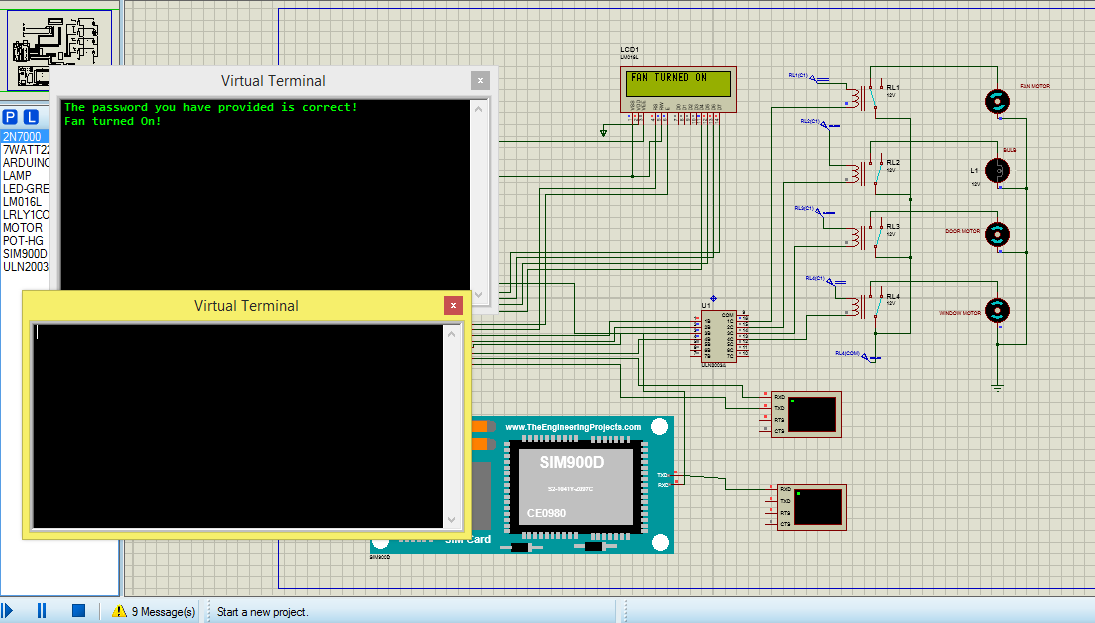


Figure 20: - The output of command ‘1’ to turn on the fan.

**Command ‘2’:** - used to control the fan which sends a LOW signal to the relay in order to turn off the fan. Figure 21 is the result of this command.

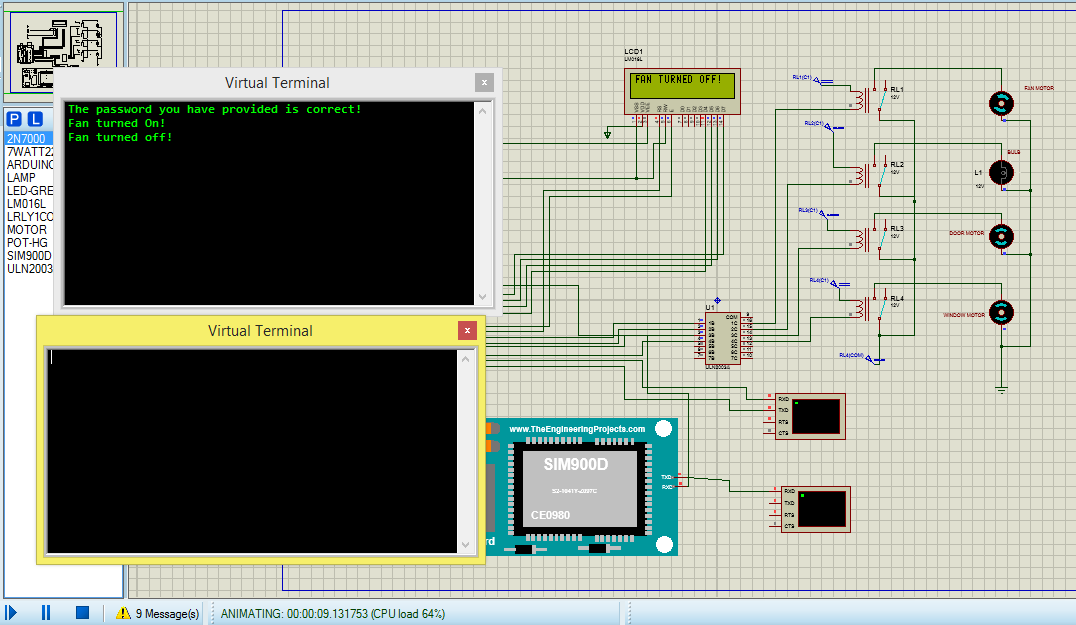


Figure 21: - The output of command ‘2’ to turn off the fun.

**Command ‘3’:** - used to control the light bulb which sends a HIGH signal to the relay in order to turn on the light. Figure 22 is the result of this command.

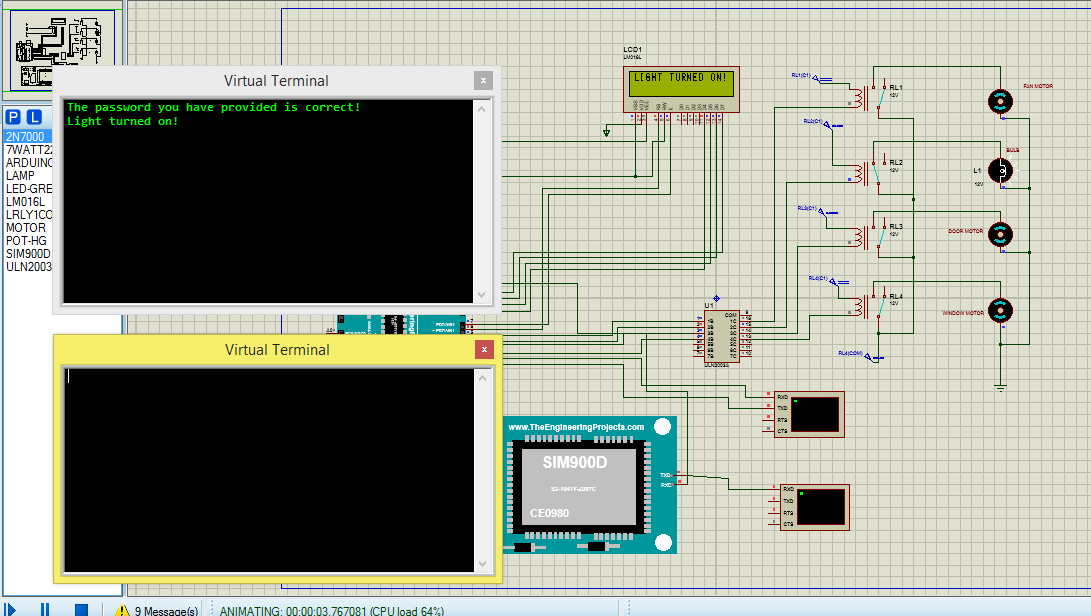


Figure 22: - Result of command ‘3’ to turn off the light.

**Command ‘4’:** - used to control the fan which sends a LOW signal to the relay in order to turn off the light. Figure 23 is the result of this command.

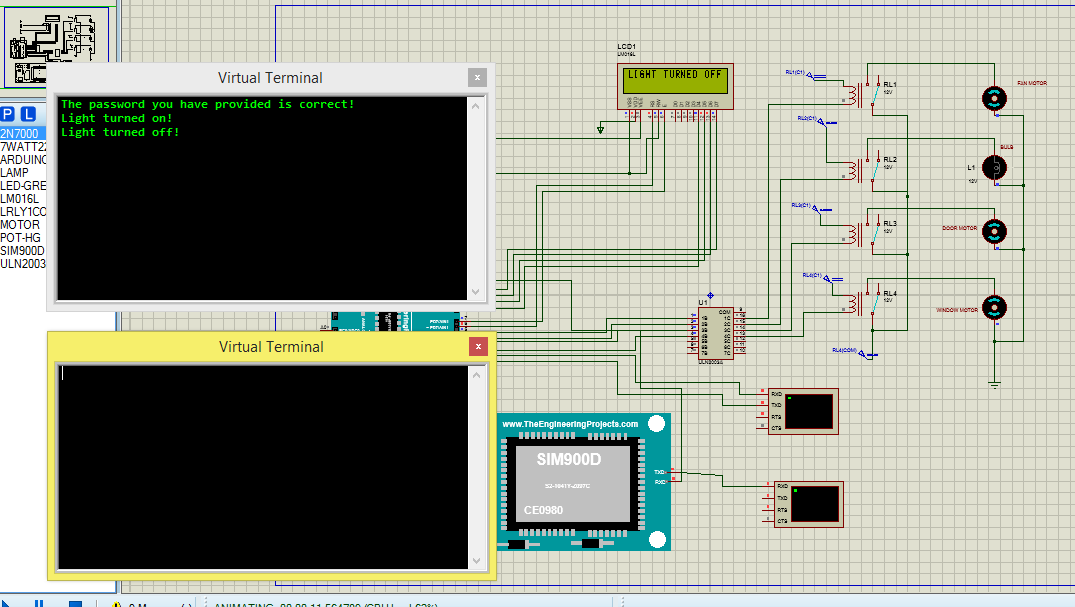


Figure 23: - The output of command ‘4’ to turn off the bulb.

**Command ‘5’:** - used to control the door which sends a HIGH signal to the relay in order to unlock the door. Figure 24 is the result of this command.

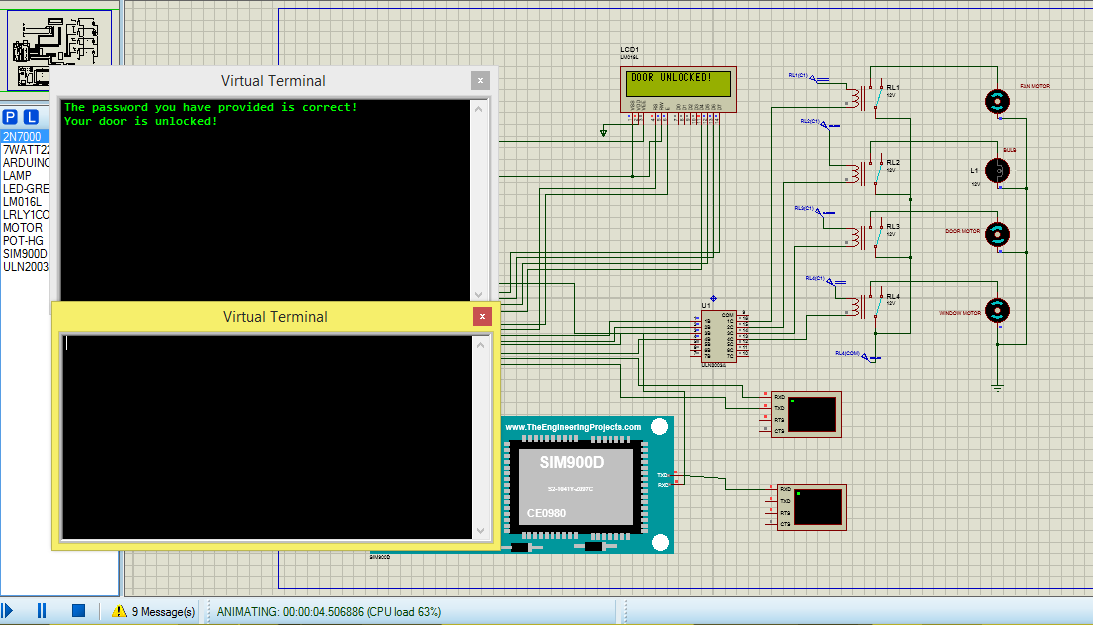


Figure 24: - Output of command ‘5’ to unlock the door.

**Command ‘6’:** - used to control the door which sends a LOW signal to the relay in order to lock the door. Figure 25 is the result of this command.

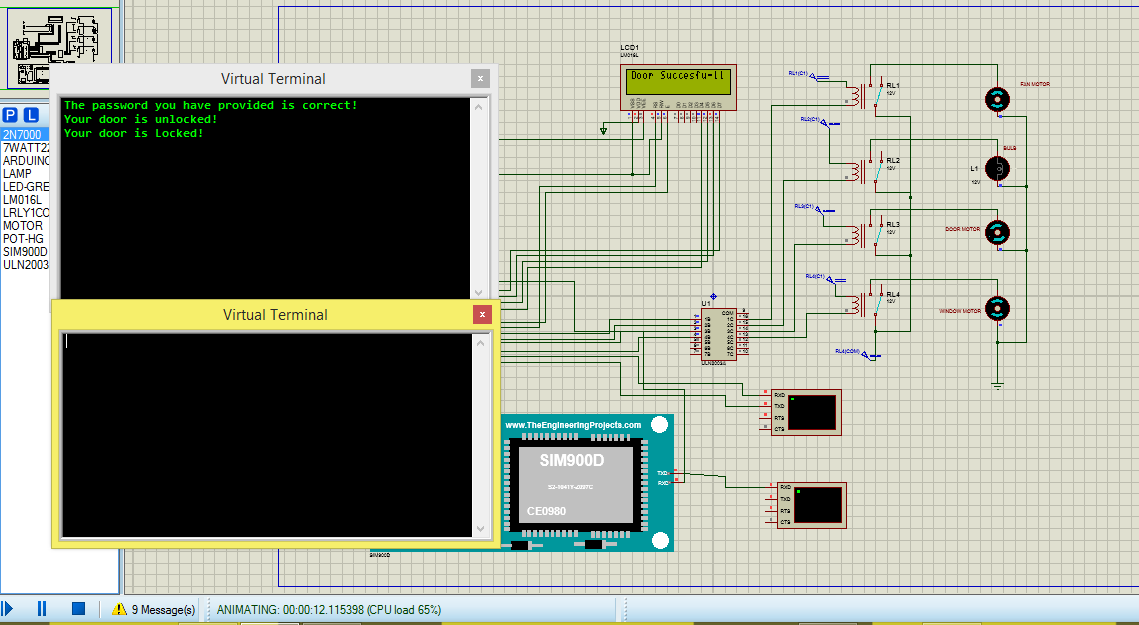


Figure 25: - Output of command ‘6’ to lock the door.

**Command ‘7’:** - used to control the window which sends a HIGH signal to the relay in order to unlock the window. Figure 26 is the result of this command.

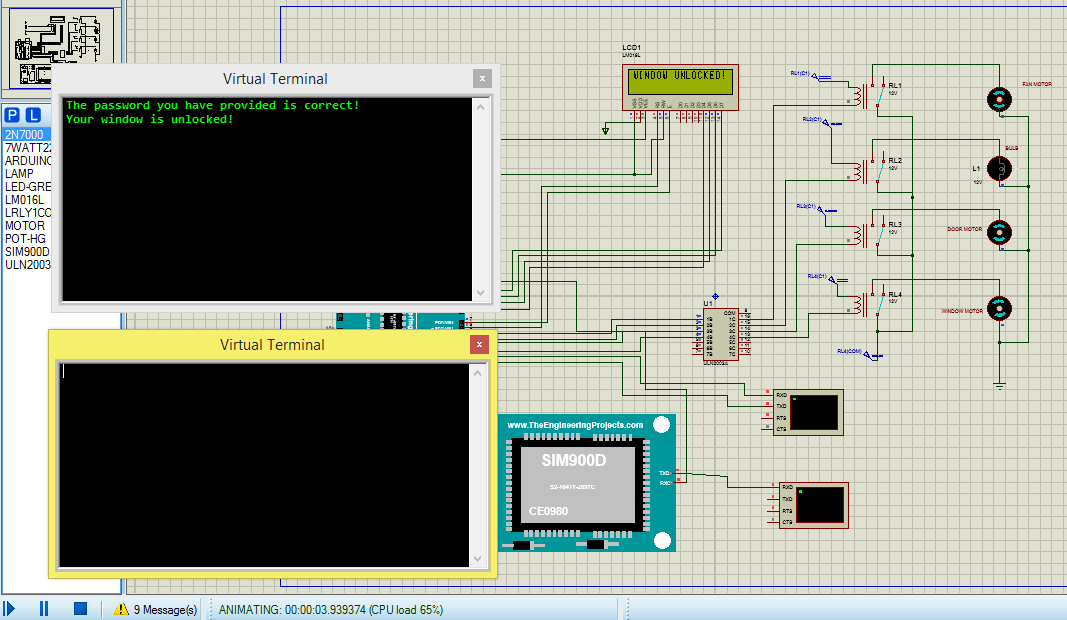


Figure 26: - Output of command ‘7’ unlocking the window.

**Command ‘8’:** - used to control the window which sends a LOW signal to the relay in order to lock the window. Figure 27 is the result of this command.

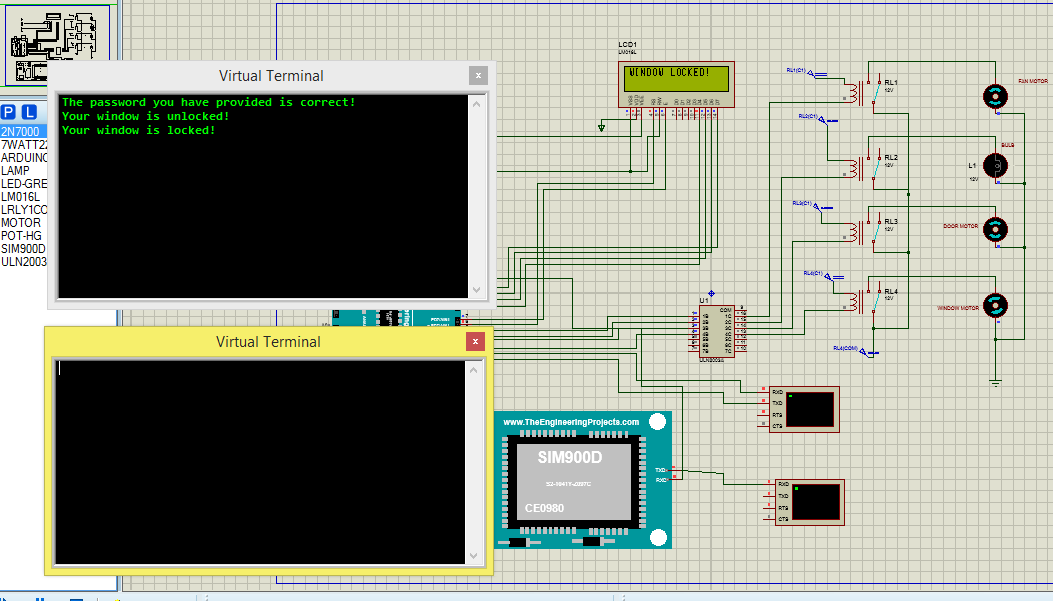


Figure 27: - Result of command ‘8’ to lock the window.

**Command ‘9’:** - used to acquire the status of the device depending on their current situation. Weather they are closed or open or on or off depending on the status it will reply the owner with a full information of all equipment.

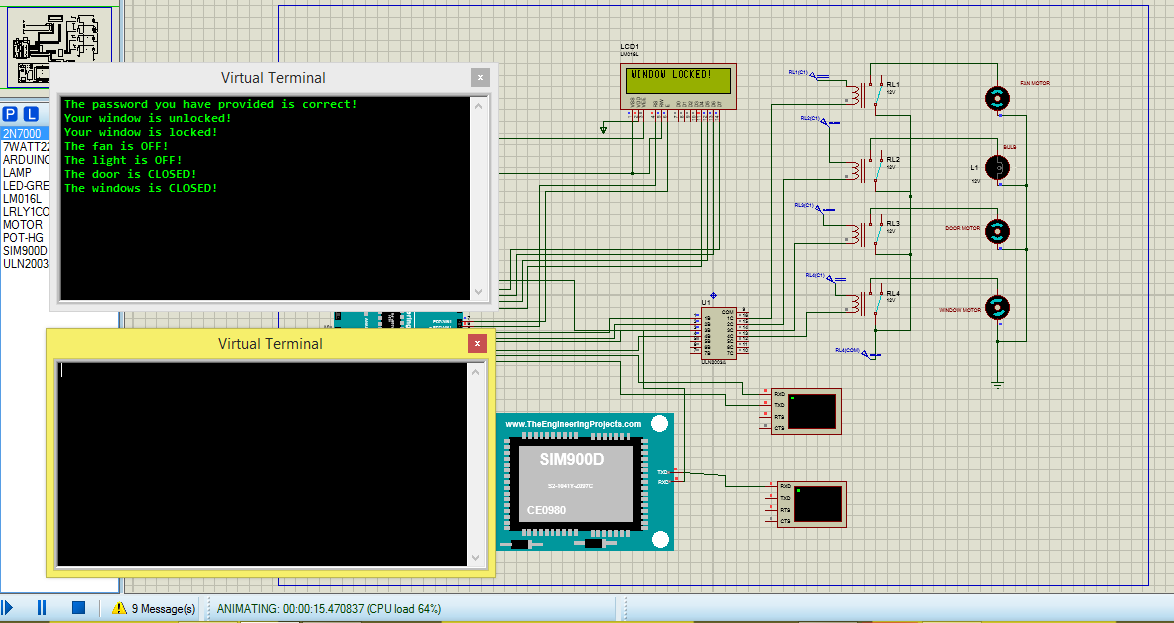


Figure 28: - Output of command ‘9’ with respective reply.

As we can see the after the system received command 9 it checked for the device current state and replied as accordingly. For example, we have tested our system

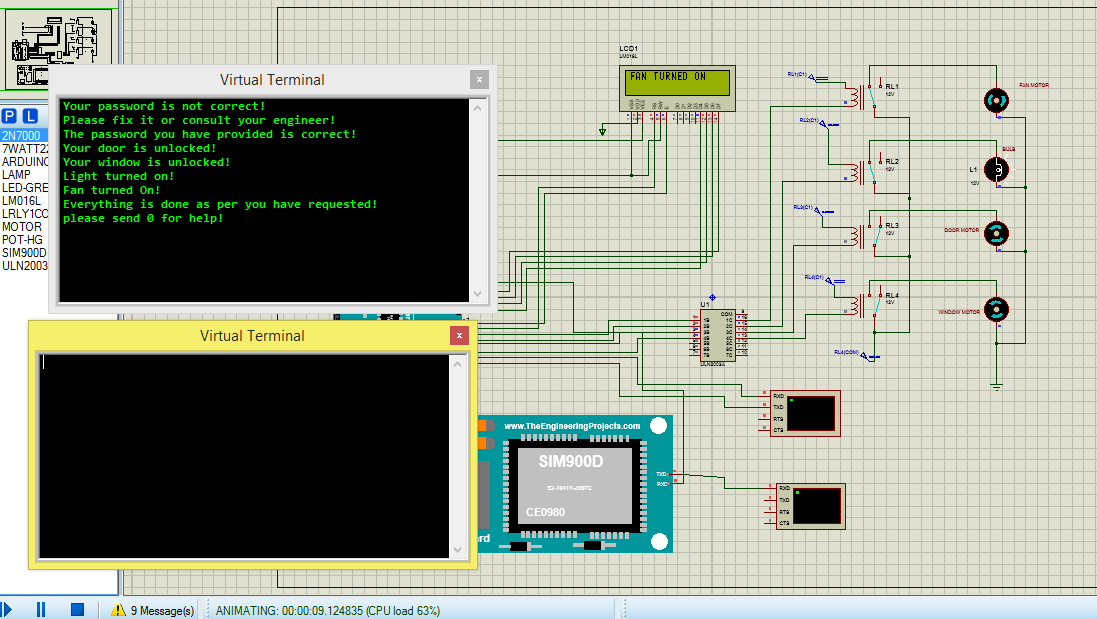
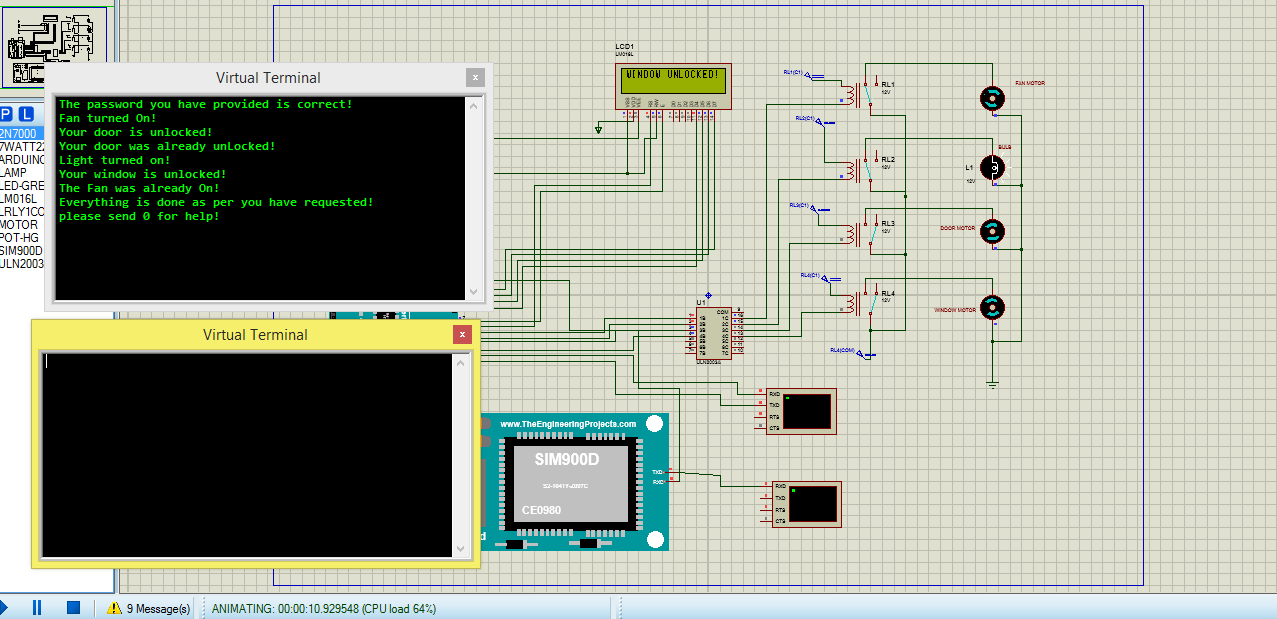
 **Command ‘a’:** - when we send this command to the system it will start all the devices that are

Figure 29: - Output of command ‘a’ while status of all device is off.

atached to the controller and reply us with status, the output has beedn shown on figure 29 above. But here the good feature of this thing is that if we have sent this command while some of the devices are already on then the system will first identify which devices are on and off or which door is closed or open and if there is any, will reply us with something like, (name of device) is on/off or closed or opened, and for those who are already on reply us with (name of device) is already on/off or open/close. See figure 30 for the result of this sutuation.

Figure 30: - Output of command ‘a’ where some device is already on.

As we can observe above the result of the command was based on the current status of the devices. In our case we have put our devices as following;

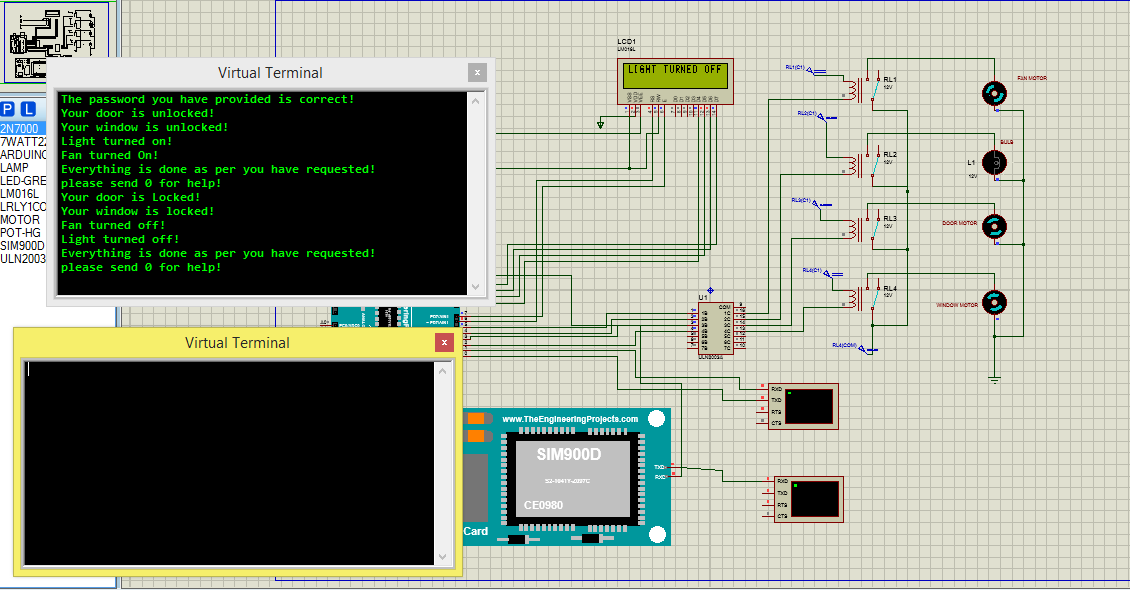
* Turned on fun.
* Unlocked door.

Then when we send command ‘a’ to the system it replied as with a report and action as following;

* Your door was **already** unlocked.
* Light turned on.
* Your window is unlocked.
* Your fan was **already** on.

If you have noticed the sequence its perfectly well thoughted to mimic human experience around house. When a person come to home it’ll first open the door, then turn on the light, open the window, and then turn on fan if necessary. The system is also programmed to do the same thing in a sequence that is mostly used by people.

**Command ‘z’:** - is used to turn off all the fan and light, close the window and door. the output has beedn shown on figure 31 above. But here the good feature of this thing is that if we have sent this command while some of the devices are already on then the system will first identify which devices are on and off or which door is closed or open and if there is any, will reply us with something like, (name of device) is on/off or closed or opened, and for those who are already on reply us with (name of device) is already on/off or open/close. See figure 32 for the result of this sutuation.

Figure 31: - output of command ‘z’ to turn off light/fan and close window/door.

As we can observe above the result of the command was based on the current status of the devices. In our case we have put our devices as following;

* Turned on fun.
* Unlocked door.

Then when we send command ‘a’ to the system it replied as with a report and action as following;

* Your door was **already** locked.
* Light turned off.
* Your window is locked.
* Your fan was **already** off.

Again, if you have noticed the sequence its perfectly well thoughted to mimic human experience around house. When a person is going to sleep or move out of house it’ll first lock the door, lock the window, turn off fan and then, then turn off the light if necessary. The system is also programmed to do the same thing in a sequence that is mostly used by people.

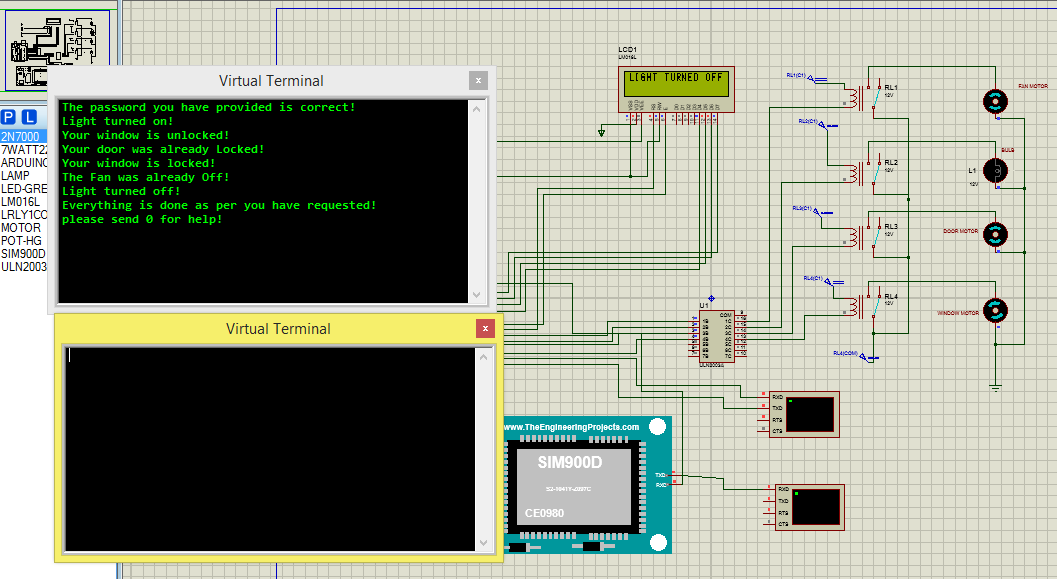


Figure 32: - output of command ‘z’ where some devices are already off.

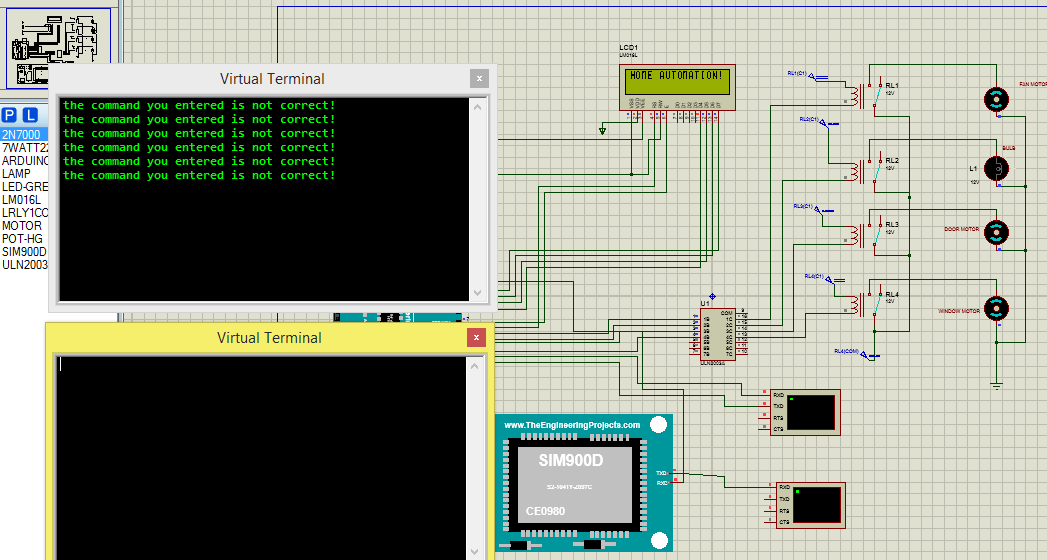
So far we have seen what we can do and can’t do with the system using visual graphic to illustrate each commands purpose and all possible outputs.

As we know there are a lot of keys available on our keyboard and that will pave the way for error in input. Means, if an owner of the house is In a rush and accidentaly sent the wrong command to the system, we found it reseonable to show the error notification that will tell the owner of the house that it’s the wrong command. See figure 33, below.

Figure 33: - Error notification message for wrong command entry.

## Conclusion

So far, we have seen what we can do with the system and what we can’t do by using screenshot of all possible outputs. And we can conclude that the system has worked the way we wanted. Except for the limitations stated in chapter 1 and other factors like incapability of the user. Actually, there is no system that is fully perfect on it’s first development. Ours also needs an update of both hardware and software program to support different features like password changing and multiple user identified by phone number.



# Chapter 5

## Introduction

It’s obvious that there is no system that is fully perfect on its first development. Which means every system in our world needs to be improved starting from the day of its development. The update to the device is not only because of the limitation and error but also to improve the user experience toward the system. And also, to meet the need of the situation at a time of update.

## Recommendation and feature work

The technology of home automation is a field which is highly emerging. And when we are working on this area there are a lot of works that need to be improved. These are included both in limitation and as a feuture work to improve the way human can communicate with it’s home appliances. Let’s see them by deviding them into limitation and as an update.

As a limitation in the feature we will implement a sound notification system that will make the person in the house to look at the lcd notification. The sound can be either beeping sound or a buzzer. The other thing is we will add the password changing function to the system. See figure 34,

**Power Supply**

**Arduino Uno**

**GSM module**

**Message from User**

**User send message**

**Transistor**

**Transistor**

**Transistor**

**Transistor**

**Relay**

**Relay**

**Relay**

**Relay**

**Door**

**Window**

**Fan**

**Bulb**

Figure 34: - The improved diagram with sound or buzzer added to the system.

On the improvement side we can implement the learning algorithm, deep learning (artificial intelligence) to our system to teach the system the user repeated action so that it can decide what to do and when to do with out the user input. To do this we have to also add different sensors like temperature sensor to control fan based on the current temperature and the user reaction toward the situation, motion sensor on doors with facial recognition feature to automatically lock or unlock the door. We can also implement the light sensor and user reaction to teach when to turn on and turn off the light. As shown in figure below.

Motion sensor

Camera

Light sensor

Temperature sensor

Learning algorithm with different parameters like,

* Time of the command.
* Temperature during the command.
* Time of entrance.
* Light intensity during light on/off command

**Power Supply**

**Arduino Uno**

**GSM module**

**Message from User**

**User send message**

**Transistor**

**Transistor**

**Transistor**

**Transistor**

**Relay**

**Relay**

**Relay**

**Relay**

**Door**

**Window**

**Fan**

**Bulb**

Figure 35: - Improved design of the system for feature work with neural network implemented in it.

## Conclusion

Generally, we have seen there is a lot more we can do with this technology by adding different sensors and also neural networks to make the system as better as possible. We have also showed how we can implement the sound notification system and different sensors using the block diagram on the above section.

# References

|  |  |
| --- | --- |
| [1] | R. Baby, "RF page," [Online]. Available: https://www.rfpage.com/evolution-of-wireless-technologies-1g-to-5g-in-mobile-communication/. [Accessed 16 May 2019]. |
| [2] | M. V. R. Murthy, "Mobile based primary hea lth care system for rural India.," 2008. |
| [3] | I. G. i. ,. K. F. k. N. &. K. G. Potamitis, " An integrated system for sma rt-home control of appliances based on remote speech interaction.," 2003. |
| [4] | N. P. A. V. L. S. A. &. T. r. R. D. Jawarkar, "M icro-controller based Remote Monitoring using Mobile through Spoken Commands," 2008. |
| [5] | A. R. P. k. R. &. G. V. Delgado, "Re mote-controlled home auto mation systems with different network technologies," 2006. |
| [6] | G. &. S. r. D. Conte, "Viewing ho me automat ion systems as multip le agents systems.," 2003. |
| [7] | B. C. i. D. C. R. &. S. D. Ciubotaru-Petrescu, " Wire less Solutions for Telemetry in Civ il Equip ment and Infrastructure Monitoring.," 2006. |
| [8] | A. J. H. A. K. Mohd Helmy Abd Wahab, "GSM Based Electrical Control System for Smart Home Application.," 2014. |
| [9] | A. R. S. K. M. P. S. M. Ankita KS, "Electronic For You," 1969. [Online]. Available: https://electronicsforu.com/resources/gsm-module. [Accessed 18 May 2019]. |
| [10] | M. Rouse, "Tech Target whatis.com," [Online]. Available: https://whatis.techtarget.com/definition/LCD-liquid-crystal-display. |
| [11] | "collins dictionary," [Online]. Available: https://www.collinsdictionary.com/dictionary/english/bulb. [Accessed 18 May 2019]. |
| [12] | J. Bezo, "Amazon Online Shop," [Online]. Available: https://www.amazon.co.uk. [Accessed 19 May 2019]. |
| [13] | S. Team, "Safe Wise," [Online]. Available: https://www.safewise.com/home-security-faq/electric-door-locks/. [Accessed 19 May 2019]. |

Appendix 1: System program code.

The following code is written to run on Arduino Uno, on Arduino IDE.

#include <SoftwareSerial.h>

#include<LiquidCrystal.h>

SoftwareSerial mySerial(12, 13);

LiquidCrystal lcd(6,7,8,9,10,11);

#define FAN 5

#define BULB 4

#define DOOR 3

#define WIN 2

boolean fans=false, doors=true, bulbs=false, windows=true, pass=false;

void setup()

{

mySerial.begin(9600); // Setting the baud rate of GSM Module Â

Serial.begin(9600); // Setting the baud rate of mySerial Monitor (Arduino) //mySerial

delay(100);

pinMode(FAN, OUTPUT);

pinMode(BULB, OUTPUT);

pinMode(DOOR,OUTPUT);

pinMode(WIN,OUTPUT);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("HOME AUTOMATION!");

delay(1000);

}

void loop() {

if (Serial.available()>0)

switch(Serial.read())

{

case 'x':

PassCheck();

break;

case '1':

FanOn();

break;

case '2':

FanOff();

break;

case '3':

BulbOn();

break;

case '4':

BulbOff();

break;

case '5':

OpenDoor();

break;

case '6':

CloseDoor();

break;

case '7':

OpenWindow();

break;

case '8':

CloseWindow();

break;

case '9':

DeviceStatus();

break;

case '0':

GetHelp();

break;

case 'a':

TurnAllOn();

break;

case 'z':

TurnAllOff();

break;

default:

mySerial.println("the command you entered is not correct!");

}

if (mySerial.available()>0)

Serial.write(Serial.read()); //mySerial at first

}

void PassCheck()

{

pass=true;

mySerial.println("The password you have provided is correct!");

}

void FanOn()

{

if (pass==true){

if (fans==true){

mySerial.println("The Fan was already On!");

}

else {

digitalWrite(FAN, HIGH);

fans=true;

delay(500);

mySerial.println("Fan turned On!");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("FAN TURNED ON");

delay(500);

}//every

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void FanOff()

{

if (pass==true){

if (fans==false){

mySerial.println("The Fan was already Off!");

}

else {

digitalWrite(FAN, LOW);

fans=false;

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("FAN TURNED OFF!");

delay(500);

mySerial.println("Fan turned off!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void TurnAllOn()

{

if (pass==true){

if (fans==false && bulbs==false && doors==true && windows==true) {

OpenDoor();

OpenWindow();

BulbOn();

FanOn();

mySerial.println("Everything is done as per you have requested!");

mySerial.println("please send 0 for help!");

}

else if (fans==false && bulbs==false && doors==true && windows==false) {

OpenDoor();

BulbOn();

FanOff();

mySerial.println("your window was already unlocked!");

mySerial.println("please send 0 for help!");

}

else if (fans==false && bulbs==false && doors==false && windows==false) {

BulbOn();

FanOff();

mySerial.println("Your door and Window were already unlocked!");

mySerial.println("please send 0 for help!");

}

else if (fans==false && bulbs==true && doors==false && windows==false) {

FanOff();

mySerial.println("Your door and Window were already unlocked!");

mySerial.println("The light was already On!");

mySerial.println("please send 0 for help!");

}

else if (fans==true && bulbs==true && doors==false && windows==false) {

mySerial.println("All of your fan and light devices are On!.your Door and Window are also already unlocked!");

mySerial.println("please send 0 for help!");

}

else {

OpenDoor();

BulbOn();

OpenWindow();

FanOn();

mySerial.println("Everything is done as per you have requested!");

mySerial.println("please send 0 for help!");

//i have no idea what to write here let's leave it empty for now ;-)

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void TurnAllOff()

{

if (pass==true){

if (fans==true && bulbs==true && doors==false && windows==false) {

CloseDoor();

CloseWindow();

FanOff();

BulbOff();

mySerial.println("Everything is done as per you have requested!");

mySerial.println("please send 0 for help!");

}

else if (fans==true && bulbs==true && doors==false && windows==true) {

CloseDoor();

FanOff();

BulbOff();

mySerial.println("your window was already locked!");

mySerial.println("please send 0 for help!");

}

else if (fans==true && bulbs==true && doors==true && windows==true) {

FanOff();

BulbOff();

mySerial.println("Your door and Window were already locked!");

mySerial.println("please send 0 for help!");

}

else if (fans==true && bulbs==false && doors==true && windows==true) {

FanOff();

mySerial.println("Your door and Window were already locked!");

mySerial.println("The light was already Off!");

mySerial.println("please send 0 for help!");

}

else if (fans==false && bulbs==false && doors==true && windows==true) {

mySerial.println("All of your fan and light devices are Off!.your Door and Window are also already locked!");

mySerial.println("please send 0 for help!");

}

else {

CloseDoor();

CloseWindow();

FanOff();

BulbOff();

mySerial.println("Everything is done as per you have requested!");

mySerial.println("please send 0 for help!");

//i have no idea what to write here let's leave it empty for now ;-)

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void BulbOn()

{

if (pass==true){

if (bulbs==true){

mySerial.println("The light was already On!");

}

else {

digitalWrite(BULB, HIGH);

bulbs=true;

delay(100);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("LIGHT TURNED ON!");

delay(500);

mySerial.println("Light turned on!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void BulbOff()

{

if (pass==true){

if (bulbs==false){

mySerial.println("The light was already Off!");

}

else {

digitalWrite(BULB, LOW);

bulbs=false;

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("LIGHT TURNED OFF");

delay(500);

mySerial.println("Light turned off!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void OpenDoor()

{

if (pass==true){

if (doors==false){

mySerial.println("Your door was already unLocked!");

}

else {

digitalWrite(DOOR, HIGH);

doors=false;

delay(1000);

digitalWrite(DOOR, LOW);

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("DOOR UNLOCKED!");

delay(500);

mySerial.println("Your door is unlocked!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void CloseDoor()

{

if (pass==true){

if (doors==true){

mySerial.println("Your door was already Locked!");

}

else {

digitalWrite(DOOR, HIGH);

doors=true;

delay(1000);

digitalWrite(DOOR, LOW);

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Door Succesfu-");

delay(500);

lcd.print("lly LOCKED");

delay(500);

mySerial.println("Your door is Locked!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void OpenWindow()

{

if (pass==true){

if (windows==false){

mySerial.println("Your Windodw was already unLocked!");

}

else {

digitalWrite(WIN, HIGH);

windows=false;

delay(1000);

digitalWrite(WIN, LOW);

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("WINDOW UNLOCKED!");

delay(500);

mySerial.println("Your window is unlocked!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void CloseWindow()

{

if (pass==true){

if (windows==true){

mySerial.println("Your window was already Locked!");

}

else {

digitalWrite(WIN, HIGH);

windows=true;

delay(1000);

digitalWrite(WIN, LOW);

delay(500);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("WINDOW LOCKED!");

delay(500);

mySerial.println("Your window is locked!");

}

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void DeviceStatus()

{

if (pass==true){

delay(500);

mySerial.print("The fan is ");

if (fans==true){

mySerial.println("ON!");

}

else if (fans==false){

mySerial.println("OFF!");

}

mySerial.print("The light is ");

if (bulbs==true){

mySerial.println("ON!");

}

else if (bulbs==false){

mySerial.println("OFF!");

}

mySerial.print("The door is ");

if (doors==false){

mySerial.println("OPEN!");

}

else if (doors==true){

mySerial.println("CLOSED!");

}

mySerial.print("The windows is ");

if (windows==false){

mySerial.println("OPEN!");

}

else if (windows==true){

mySerial.println("CLOSED!");

}

//mySerial.println((char)26);

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

void GetHelp()

{

if (pass==true){

delay(500);

mySerial.println(" Please use the following command to control your home appliances accordingly.");

mySerial.println("1) send '1' to turn ON and '2' to turn OFF the fan.");

mySerial.println("2) send '3' to turn ON and '4' to turn OFF the LIGHT.");

mySerial.println("3) send '5' to OPEN and '6' to CLOSE the DOOR.");

mySerial.println("4) send '7' to OPEN and '8' to CLOSE the WINDOW.");

mySerial.println("5) send '9' to recieve the status of your home appliances.");

mySerial.println("6) send '0' to recieve help information.");

mySerial.println("7) send 'a' to turn ON ALL and 'z' to turn OFF ALL of your devices.");

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" DEVICE READY ");

delay(500);

lcd.print("WAITING FOR CMD!");

}

else {

mySerial.println("Your password is not correct!");

mySerial.println("Please fix it or consult your engineer!");

}

}

Appendix 2: budget requirement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R.NO** | **Item** | **Quantity** | **Cost per item (ETB)** | **Total cost (ETB)** |
| 1) | Micro Controller (Arduino Uno) | 1 | 500.00 | 500.00 |
| 2) | GSM module SIM900D | 1 | 411.48 | 411.48 |
| 3) | Fan | 1 | 485.73 | 485.73 |
| 4) | Relays | 4 | 226.53 | 906.12 |
| 5) | Bulbs | 1 | 175.23 | 175.23 |
| 6) | Resistors | 50 pack | 32.13 | 32.13 |
| 7) | Transistor | 200 pack | 242.73 | 242.73 |
| 8) | LCD | 1 | 120.15 | 120.15 |
|  | | | **Total** | **2873.57** |

Appendix 3: Activity Schedule.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **R.NO** | **Task/Activity** | **Duration** | **Start date (G.C)** | **End date (G.C)** | 16-May-19 | 17-May-19 | 18-May-19 | 19-May-19 | 20-May-19 | 21-May-19 | 22-May-19 | 23-May-19 | 24-May-19 | 25-May-19 | 26-May-19 | 27-May-19 | 28-May-19 | 29-May-19 | 30-May-19 | 31-May-19 | 1-Jun-19 | 2-Jun-19 | 3-Jun-19 | 4-Jun-19 | 5-Jun-19 | 6-Jun-19 | 7-Jun-19 | 8-Jun-19 | 9-Jun-19 | 10-Jun-19 | 11-Jun-19 | 12-Jun-19 | 13-Jun-19 | 14-Jun-19 | 15-Jun-19 |
| 1) | Proposal writing | 4 days | 16-May-19 | 20-May-19 | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2) | Presentation | 1 day | 21-May-19 | 21-May-19 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3) | Installing required software’s | 2 days | 22-May-19 | 23-May-19 |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4) | Collecting different libraries | 1 day | 24-May-19 | 24-May-19 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5) | Understanding device libraries | 2 days | 25-May-19 | 26-May-19 |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6) | Designing the circuit. | 6 days | 27-May-19 | 1-Jun-19 |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7) | Develop Arduino program. | 10 days | 2-Jun-19 | 11-Jun-19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X | X | X | X | X | X |  |  |  |  |
| 8) | Simulating and debugging the design. | 3 days | 12-Jun-19 | 14-Jun-19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |  |
| 9) | Final documentation and presentation | 1 day | 15-Jun-19 | 15-Jun-19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |