PREFACE

In the name of ALLAH, the most merciful and beneficent we have been assigned with a project of microprocessor system and interfacing in which we came up with an idea of making a smart safe lock system which requires password to get open and also we can access the safe using an app name dabble through Bluetooth.

We had a week to make this project and we managed everything in a week so we worked day and night ALHAMDULILAH we done it and our project came up according to our expectations and thanks to ALLAH almighty as without his help we cannot do anything.

A project report is submitted to the department of Electrical and Electronics Engineering, by Raza Haider, Raja Haseeb and Obaidullah khan in Comsats University, Islamabad in partial fulfillment of the requirement for the award of 50 percent marks in lab performances of microprocessor system and interfacing Bachelor of Electrical Engineering.

Raza haider
Raja Muhammad haseeb
Obaidullah khan

Microprocessor system and interfacing
Department of Electrical and Electronics Engineering

TABLE OF CONTENT

CHAPTER 1

1. INTRODUCTION

1.1. BACKGROUND

Securing our expensive things has become a major concern in the twenty first century everybody wants to feel safe about their expensive things. Smart safe lock system using Arduino and Bluetooth application is a project aimed at increasing and advancing the safety of valuables and property of the people. The project deals primarily with the protection of safe and making it more secured within our house. With the help of smart safe lock, when indoors or while you are away from your home and workplace, the safe are protected primarily to grant access to only authorized persons using their smartphone and Bluetooth application.

With the advancement of technology smartphone has become a household requirement. As of 2007 smartphones were not more than two percent of phone industries, but in 2009 the smartphone world has taken more than fifty percent of phone market (www.Wikipedia.com/androiddevelopers) With this it is quite logical to say that there is at least a smartphone owner in every house. With this smartphone we can simply interface it with an Arduino (microcontroller) to keep our valuables safer and more secure with a single click to lock or unlock the resulting safe. The microcontroller will be mounted on the safe which will then communicate with the device through a Bluetooth module which will be connected to the microcontroller.

This project is aimed at making security better and securing the properties of people at home or their work places issues of theft and burglary increases with high rate as such safe security is a necessity, safety is now of paramount importance which triggers the need for project ‘smart safe lock system’ to provide a more safe and secure place for valuables in generally.

1.2. PROBLEM STATEMENT

In order to make sure that every door is safe people now look for counter measures to protect their doors. With the vast and different forms of keeping doors locked from unauthorized persons many people tend to use low means of protection. Meanwhile the deployment is of more advanced technology like the use of smart door security is a step forward. This project aims to keep doors safe and also ease access. The latching of the door is solely controlled by smartphone via the Bluetooth connection established between the device and the microcontroller.

1.3. OBJECTIVES

The aim of this project is to design and implement a smart safe lock security system using Arduino and Bluetooth application that could help advance the protection of money and

expensive things like jewelry, gold, diamond etc.

- Designing of a smartphone application to control the movement of doors.
- Interfacing of Arduino UNO with the smartphone via Bluetooth module to establish a connection between the Arduino the smartphone.
- Controlling the lock to open or close a door which we controlled and processed due to the response of the microcontroller.

1.4. SCOPE OF PROJECT

The project is about interacting with component and devices with the help of HC-06 (Bluetooth module). The project is limited to performing the task of opening and closing of safe it is also used to save things from being theft as built in buzzer will turn on if it receives wrong password.

CHAPTER 2

2. LITERATURE REVIEW AND THEORITICAL BACKGROUND

2.1. REVIEW OF LITERATURE

Smart locks have been implemented using different methods such as Radio frequency identification (RFID) and Biometric lock to unlock and lock door or safe. Both the RFID and biometric lock are real ideal and smart ways to make a door smart, due to necessity and limitations such as cloning of biometric prints or card. The use of Bluetooth and smartphone is much simpler and easier to adapt and use. It gives you more access to communicate with the door and it also give access to physically challenged persons that might not have a finger to use for biometric lock or is crippled to use RFID but with respect to this project physically challenged can simply open their door by single click in device. Adarsh V Patil et al (2008) did a similar project Android based smart door locking system which also employed the use of android phone which is also a smartphone and also a GSM module to access the door. Also Agbo David et al (2017) did a somewhat similar project based on door locking system using android application. Shafarana A.R.F et al (2017) did android based automation and security system for smart homes.

There are many other projects done on smart door in different countries. They are all different from each other in terms of designs, features, devices, and algorithm. They are mostly designed according to specific needs and availability of components in the respective areas. Some of them are cheap; some of them are very expensive. Availability of both hardware and software is necessary to work. After a long searching, I have found a lot of articles. Searching for security purpose articles, also found some projects done for door security. These are mainly done in western countries. Many projects are done only for security purpose With Arduino or Raspberry Pi. Again, the projects are done only for controlling home Appliances using Arduino or Raspberry Pi. Most of the previous researches encountered problems in their design especially in terms of cloning by other third party and availability of components.

2.2. THEORATICAL BACKGROUND OF EQUIPMENTS USED

2.2.1. ATMEGA 328P

The Atmel AVR ATmega328P is low-power 8-bit microcontroller architecture. It has 131 Powerful Instructions. Most of them require single clock cycle for execution. It has 32K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1K bytes EEPROM, 2K bytes SRAM, 23 general purpose I/O pins and 32 general purpose working registers.

2.2.1.1. ATMEGA 328P PERIPHERAL FEATURES

- 3 flexible Timer/Counters with compare modes
- Six PWM Channels
- Internal and External Interrupts
- A serial programmable USART
- Two Master/Slave SPI Serial Interface
- 8 channel 10-bit ADC

2.2.1.2. ASSOCIATED REGISTERS

Each of the AVR Digital I/O port that is Port B, Port C and Port D are associated with three I/O registers. These registers are:

- **DDRx (Data Direction Registe**r) - Sets the direction of each pin as input or output.
- **PORTx** - Used to output/write values on a Port.
- **PINx** - Used to input/ read values from a Port.

Where, x is the port B, C or D. All of these registers are 8-bit registers.

2.2.1.3. ATMEGA 328P PIN CONFIGURATION

The 23 digital I/O pins are grouped into 3 ports named as Port B, C and D. Port B and D have 8 pins each whereas Port C has 7 pins. Pin 4 and 6 are required to connect with +ve (VCC) and pin 3, 5 and 21 need to connect with ground (GND).Pin 18 is the supply voltage pin for the A/D Converter. Pin 20 is the analog reference pin for the A/D Converter.

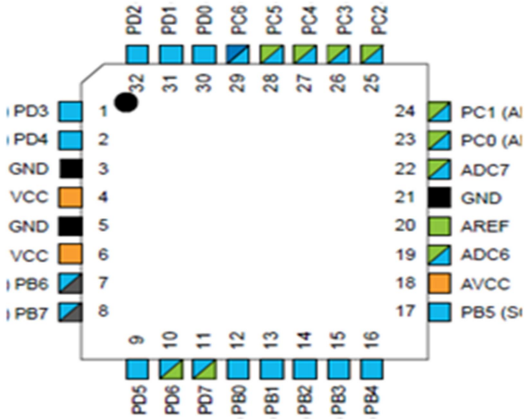


Figure 2.2.1.3 pin configuration of ATMEGA 328p

2.2.2. ARDUINO UNO

While it is possible to use a chip directly, it is sometimes much easier to use and program them when the controller is mounted on a board. It so happens, that the ATmega328p comes mounted on a number of boards. Students are recommended to use one of the two most popular variants of these boards, namely:Arduino NanoArduino Uno

The Arduino Uno is more flexible to use with a removable microcontroller, the Nano is a cheaper alternative that matches most relevant capabilities of the Uno.

The Arduino Uno is a microcontroller board based on the ATmega328p. It is simple, inexpensive, open source prototyping platform extensible to hardware and software. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller. We either need to connect it to a computer using a USB cable or power it with an AC-to-DC adapter. The Arduino circuit acts as an interface between the software part and the hardware part of the project.

2.2.2.1. SPECIFICATION

Please review figures below for Arduino Uno. Do note that both boards use the AVR Atmega328P microcontroller.

Figure 2.2.1 Arduino board

- Specification of Arduino Uno
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limit) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)

- PWM Digital I/O Pins 6
- Analog Input Pins 6
- DC Current per I/O Pin 20 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328P)
- SRAM 2 KB (ATmega328P)
- EEPROM 1 KB (ATmega328P)
- Clock Speed 16 MHz
- IO with built-in LED 1 (on pin #13)
- Length 68.6 mm
- Width 53.4 mm
- Weight 25 g

2.2.2.2. PIN CONFIGURATION

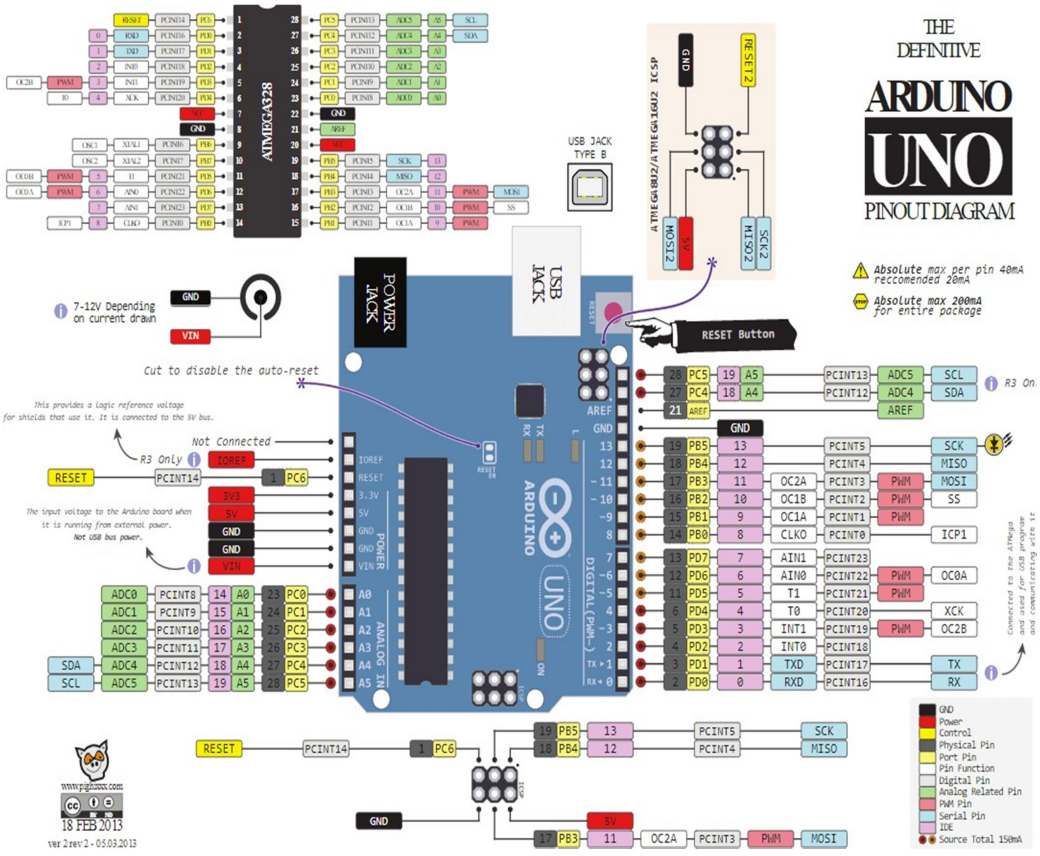


Figure 2.2.2.2 basic pin out of Arduino Uno development board

2.2.3. LIQUID CRYSTAL DISPLAY

LCD provides an output interface to AVR microcontroller. It comes in various configurations such as 20 x 2, 16 x 1, 40 x 1, 16 x 2 etc. with 5 x 8 or 5 x 11 dots resolution. Out of these, 16 x 2 is widely used.

LCD also has a microcontroller in it. This microcontroller is already coded to display characters on the screen. Therefor to use an LCD a controller to controller interface is required.

2.2.3.1. LCD CONTROL UNIT (CONTROLLER)

The LCD controller is responsible for all the operations with the LCD. These operations may include (but are not limited to):

- Movement of curser,
- Clearing display,
- Printing characters,
- Special character generation etc.

HD44780U and S6A0069 are common examples of LCD control unit. An LCD controller's internal memory comprises of Character Generator ROM (CGROM), Character Generator RAM (CGRAM) and Display Data RAM (DDRAM).d

- Character Generator ROM (CGROM) contain predefined character fonts table.

2.2.3.2. PIN CONFIGURATION

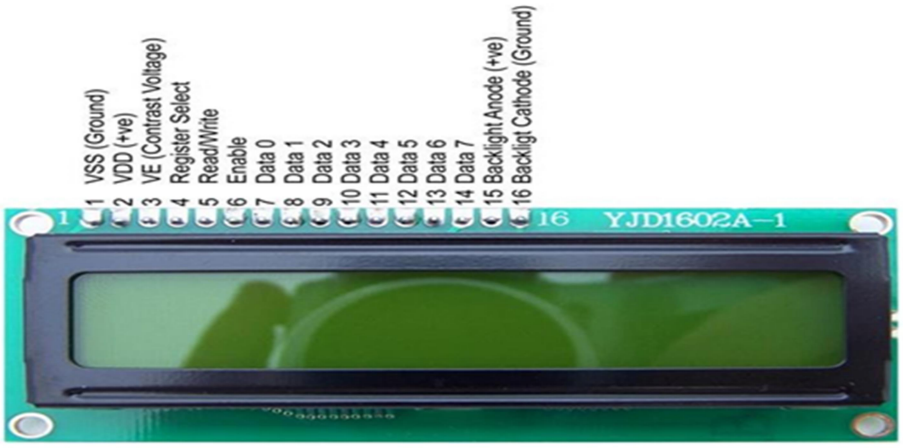


Figure 2.2.3.2 LCD pin configuration

2.2.3.3. LCD INTERFACING

LCD controller has two 8-bit registers:

- Command Register allows the user to send commands such as clear display, cursor at home etc.
- Data register allows the user to send the data to be displayed on LCD

Since same pins from D0 to D7 are used for both command and data, RS pin is used to identify between the command and data.

Following steps are used to send command or data to LCD:

- Initialize the LCD.
- Send commands from Table 4.3 to LCD.
- Send data to be displayed on LCD.

2.2.4. BLUETOOTH

2.2.4.1. HC – 06

The HC-06 module only can be a slave. This makes it only useful for say connecting a notebook as a master to a robot with a slave module e.g. for a wireless serial bridge. For most use cases the HC-06 is enough, as typically I want to have a wireless UART connection to my devices from my notebook.

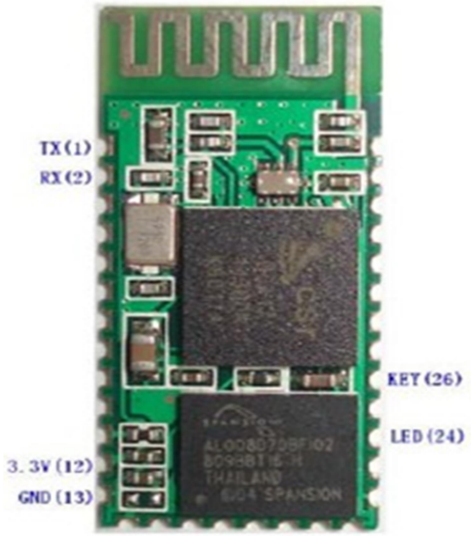


Figure 1 hc-06 module

2.2.4.2. JY-MCU V1.5 Module

Below is an image of the JY-MCU HC-06 (JY-MCU V1.5) module. The module came with a 4-pin header, and I have added the pins for STATE and KEY, and removed the plastic around the module to get access to the pins:

2.2.4.3. SPECIFICATIONS

- **KEY**: according to the data sheet, I need to pull - up this pin while power - on - reset of the module to enforce AT mode. I have not been able to verify this yet. I have been told that some modules have this pin not connected at all?
- **VCC** is indicated in the range of 3.6V-6V. The module worked for me both with 3.3V and 5V.
- **GND**: Ground
- **TXD**: serial output of the module, to be connected to RX of the microcontroller. Note that this signal is using 3.3V logic level
- **RXD**: serial input of the module, to be connected to the TX of the microcontroller. Note that this signal is using 3.3V logic levels.
- **STATE**: connected to LED2 (Pin32) of the module, but no meaning? At least on my module the pin was always low, regardless if paired or not.

2.2.4.4. PIN CONFIGURATION

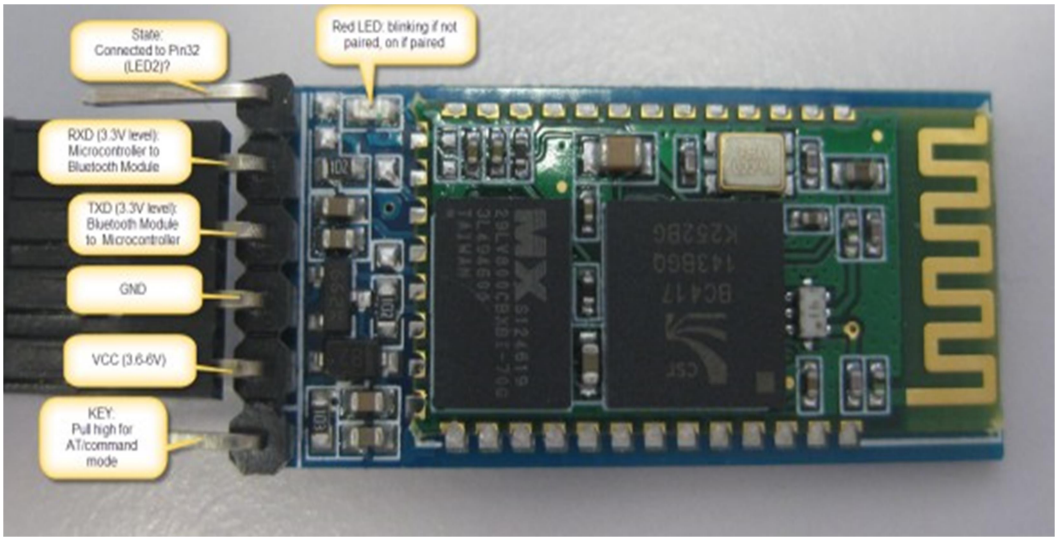


Figure 2.2.4.1 hc-06 module

2.2.5. KEYPAD

The Keypad 4x4 features 16 push buttons arranged in 4x4 matrixes to form standard alphanumeric keypad. It provides a useful human interface component for microcontroller projects. Matrix keypad uses a combination of four rows and four columns to provide button states to the microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in figure 2.2.5.

This keypad can be used in Security systems, Menu selection, Data entry for embedded systems and various other applications.

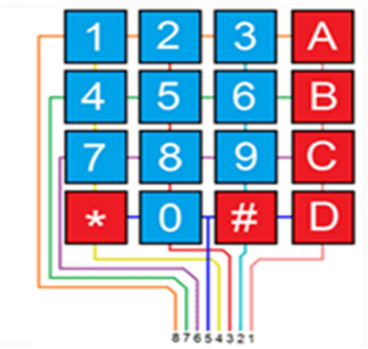


Figure 2.2.5. 4 X 4 keypad

2.2.5.1. WORKING OF KEYPAD:

As can be seen in figure 6.1, the buttons consist of 16 switches that are normally open. When one button is pressed, a pair of pins is connected. As shown in the figure, the R2 and C2 pins will help determine the button that is pressed.

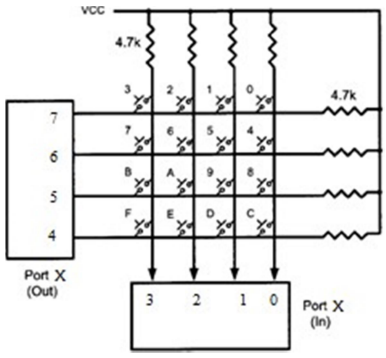


Figure 2 inner circuitry of keypad

2.2.5.2. SPECIFICATIONS

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions: Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm) Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

2.2.5.3. PIN CONFIGURATION

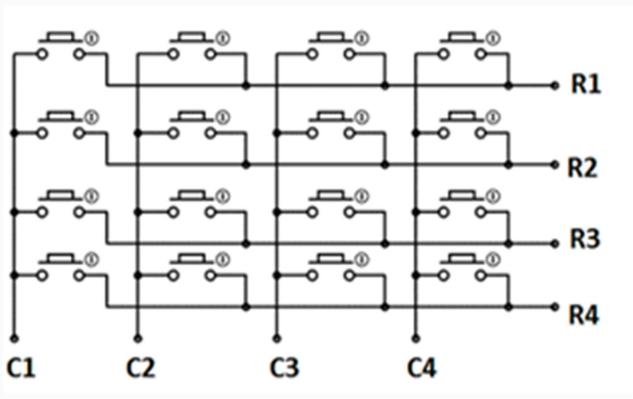


Figure 2.2.5.3. internal circuitry and pin configuration of keypad

2.2.6. RELAY

Basically, a Relay is a device with contacts that opens and closes a switch as the result of an input signal (voltage or current) applied to a coil.

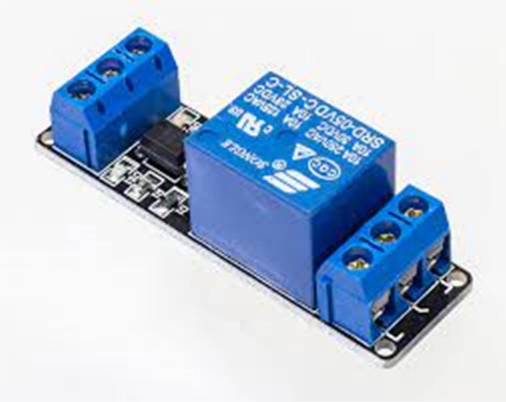
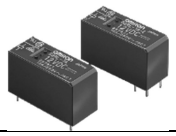


Figure 2.2.6. relay

2.2.6.1. SPECIFICATIONS

Type	Poin ts	Typical Relays
Relays for control panels · Control Relays · I/O Relays · Latching Relays, etc	Relays with 1 to 4 poles are mainly used for relay sequences or I/O applications.	MM(K), MK(S), MY, LY, G2R(S), G2RV, G7T, etc.
Built-in relays · High - capacity Relays, etc.	Built-in relays enable using a carry current of up to 40 A and are used for building into devices, e.g., to turn ON load power supplies.	G7Z, G7J, G7X, etc.
Work-saving relays · Terminal Relays · Relay Terminals, etc.	Work-saving relays are available in relay units and are used mainly for I/O applications for programmable controllers when downsizing and saving work are required.	G6D-F4B, G6B-4BND, etc.
Relays for special operations · Ratchet Relays, etc.	Relays for special operations are available as relays or relay units that are specified for a specific applications, such as alternative operation and stepping operation of pumps .	G4Q, G9B, MYA, etc.
Relays for PCBs *	These specialized relays are mounted on PCBs.	G5NB, G2RL, etc. 

2.2.7. VERO BOARD

This type of wiring board may be used for initial electronic circuit development, to construct prototypes for bench testing or in the production of complete electronic units in small quantity. Vero board was first used for prototype construction within Vero Electronics Department in 1961.

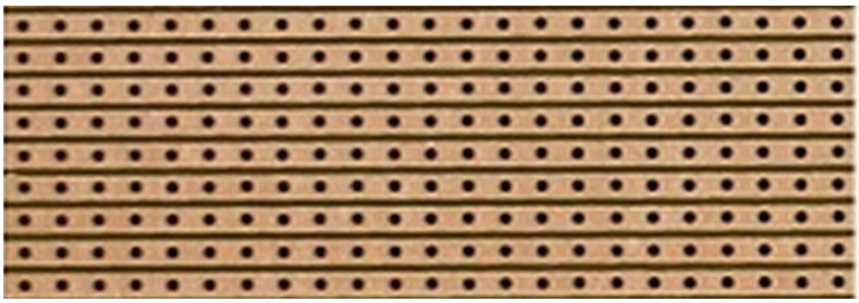


Figure 2.2.7 Vero board

2.2.8. BUZZER

Buzzer is a kind of voice device that converts audio model into sound signal. It is mainly used to prompt or alarm. According to different design and application, it can produce music sound, flute sound, buzzer, alarm sound, electric bell and other different sounds.

Figure 2.2.8 Buzzer

2.2.9. CONNECTING WIRES

Wire used to extend the firing line or leg wires in an electric blasting circuit. We use these wires connect different devices with each other so that data can be transferred and shared for easy processing.

CHAPTER 3

3. SOFTWARE AND APPLICATIONS

3.1. ARDUINO

Arduino is an open-source platform used for building electronics projects. It consists of both a physical programmable circuit board and an IDE (Integrated Development Environment) that runs on your computer. Most Arduino boards consist of an Atmel 8-bit AVR microcontroller with varying amounts of flash memory, pins and features. Arduino is programmed using the Arduino Software (IDE) which is a cross-platform application for windows, macOS and Linux. It is connected to a PC via USB cable to upload computer code to the physical board. This also provides power to the board, as indicated by a LED.

3.2. PROTEUS

The Proteus Design Suite is a complete software solution for circuit simulation and PCB design. Proteus can simulate electric/electronic and microcontroller based circuits. It supports number of microcontrollers available in the market.

3.3. DABBLE

It is an application used to assist our hardware design using a Bluetooth connection which can help us to open the safe using Bluetooth through this application we will send the password to Bluetooth which will be then passed to microcontroller and if it is correct safe will open and we will get access if not then LCD will display wrong password and buzzer will turn on.

CHAPTER 4

4. SOFTWARE IMPLEMENATATION AND DESIGN

4.1. PROCEDURE

Starting with the name of almighty ALLAH, we have designed a smart safe in which we have used atmega328p microcontroller, Bluetooth, LCD, keypad and buzzer. We have connected all the devices together in Proteus as well as in Vero board and soldered them. We have interfaced the keypad first using atmega328p and also we have connected LCD and a buzzer which will simply turn on if the code is incorrect.

We can see in the figure 4.3 that LCD is asking for password after inserting password if tis i correct it will accept the access and if it is wrong it say wrong password and buzzer will turn on and we have inserted three different passwords and on every password Bluetooth will also tell us that who opened the safe and we will get a buzzing voice in safe which is indication of safe is or safe is being accessed by unknown person who is trying to steal.

We can access the safe using Bluetooth and if someone will try to open the safe the buzzer will turn on and we will simply get a message on our Bluetooth interface and we can also get rid of robbery or something in this manner.

4.2. SOFTWARE PROGRAM

```
#define CUSTOM_SETTINGS
#define INCLUDE_TERMINAL_MODULE
#include <Dabble.h>
#include <Keypad.h>
#include <LiquidCrystal.h>
#define Password_Length 5
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
String Serialdata = "";
bool dataflag = 0;
int pos = 0;
char Data[Password_Length];
char Master[Password_Length] = "1190";
char Master1[Password_Length] = "1290";
char Master2[Password_Length] = "1390";
byte data_count = 0, master_count = 0;
bool Pass_is_good, break_loop=0;
bool door = false;
char customKey;
const byte ROWS = 4;
const byte COLS = 4;
```

```
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {4, 5, 6, 7};
byte colPins[COLS] = {8, 9, 10, 11};

Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);
void setup()
{
  lcd.begin(16, 2);
  Dabble.begin(9600);
  Serial.begin(9600);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void loop()
{
  Dabble.processInput();
  if (door == true)
  {
    customKey = customKeypad.getKey();
    if (customKey == '#')
    {
      lcd.clear();
      door = false;
    }
  }
  else
    Open();
}
void loading (char msg[]) {
  lcd.setCursor(0, 1);
  lcd.print(msg);

  for (int i = 0; i < 9; i++) {
    delay(1000);
    lcd.print(".");
  }
}
void clearData()
```

```
{
 while (data_count != 0)
 {
  Data[data_count--] = 0;
 }
 return;
}
void Open()
{
 lcd.setCursor(0, 0);
 lcd.print("Enter Password");
 customKey = customKeypad.getKey();
 if(Terminal.available())
 {
  while (Terminal.available() != 0)
  {
   Data[data_count] = Terminal.read();
   data_count++;
  }
  dataflag = 1;
 }
 else if (customKey)
 {
  do
  {
   if (customKey)
   {
    if (customKey == 'C')
    {
     clearData();
     break_loop = 1;
     lcd.clear();
     break;
    }
    Data[data_count] = customKey;
    lcd.setCursor(data_count, 1);
    lcd.print(Data[data_count]);
    data_count++;
   }
   customKey = customKeypad.getKey();
  }while (!(data_count == (Password_Length - 1)));
  if (!break_loop)
   dataflag = 1;
```

```
}
if (dataflag)
{
 if (!strcmp(Data, Master))
 {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Safe is Open");
  door = true;
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  lcd.setCursor(2, 1);
  lcd.print("HELLO RAZA");
   Terminal.print("Raza Opened the safe");
  _delay_ms(5000);
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print("Safe is Closed");
   digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  _delay_ms(500);
  door = false;
  dataflag = 0;
 }
 else if(!strcmp(Data, Master1))
 {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Safe is Open");
  door = true;
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  lcd.setCursor(2, 1);
  lcd.print("HELLO HASEEB");
  Terminal.print("Haseeb Opened the safe");
  _delay_ms(4500);
  lcd.clear();
```

```
  lcd.setCursor(1, 0);
  lcd.print("Safe is Closed");
   digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  _delay_ms(500);
  door = false;
  dataflag = 0;
}
 else if(!strcmp(Data, Master2))
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Safe is Open");
  door = true;
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  lcd.setCursor(2, 1);
  lcd.print("HELLO OBAID");
  Terminal.print("Obaid Opened the safe");
  _delay_ms(4500);
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print("Safe is Closed");
   digitalWrite(13, HIGH);
  _delay_ms(500);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  _delay_ms(500);
  door = false;
  dataflag = 0;
}
else
{
  lcd.clear();
  lcd.print(" Wrong Password ");
  digitalWrite(13, HIGH);
  _delay_ms(3000);
  digitalWrite(13, LOW);
  door = false;
```

```
    dataflag = 0;
   }
   delay(1000);
   lcd.clear();
   clearData();
  }
}
```

4.2.1. DISCUSSION

- First of all we have inserted libraries of Bluetooth, LCD, Keypad and dabble app.
- Then we have defined the password length.
- Then we have assigned that which of 4 x 4 keypad pads are going to do which command and which pin of atmega328p is going to receive data from keypad.
- Basically we have defined the rows and columns of the keypad and used them with assigning pins of atmega328p.
- Then we designed three different passwords.
- We made a pin high for buzzer to get on if password is incorrect.
- We have sent the information to Bluetooth and designed an interface to access password using Bluetooth as well.
- Then we wrote message which Bluetooth will receive if the password is correct or incorrect.
- If password is correct Bluetooth will say that this person has opened the safe.
- If it is wrong Bluetooth will send message to dabble that password is incorrect.
- This is how our code works.
- We have also added the time delay.

We can access the safe with three different passwords as defined in the code and also we can access t through two means by Bluetooth as well as by using keypad and safe LCD will say this welcome to authorized personale only and if code is wrong the buzzer will turn on and we cannot stop it until we plug it out or force fully make it stop to not create voice.

4.3. SOFTWARE IMPLENTATION ON PROTEUS

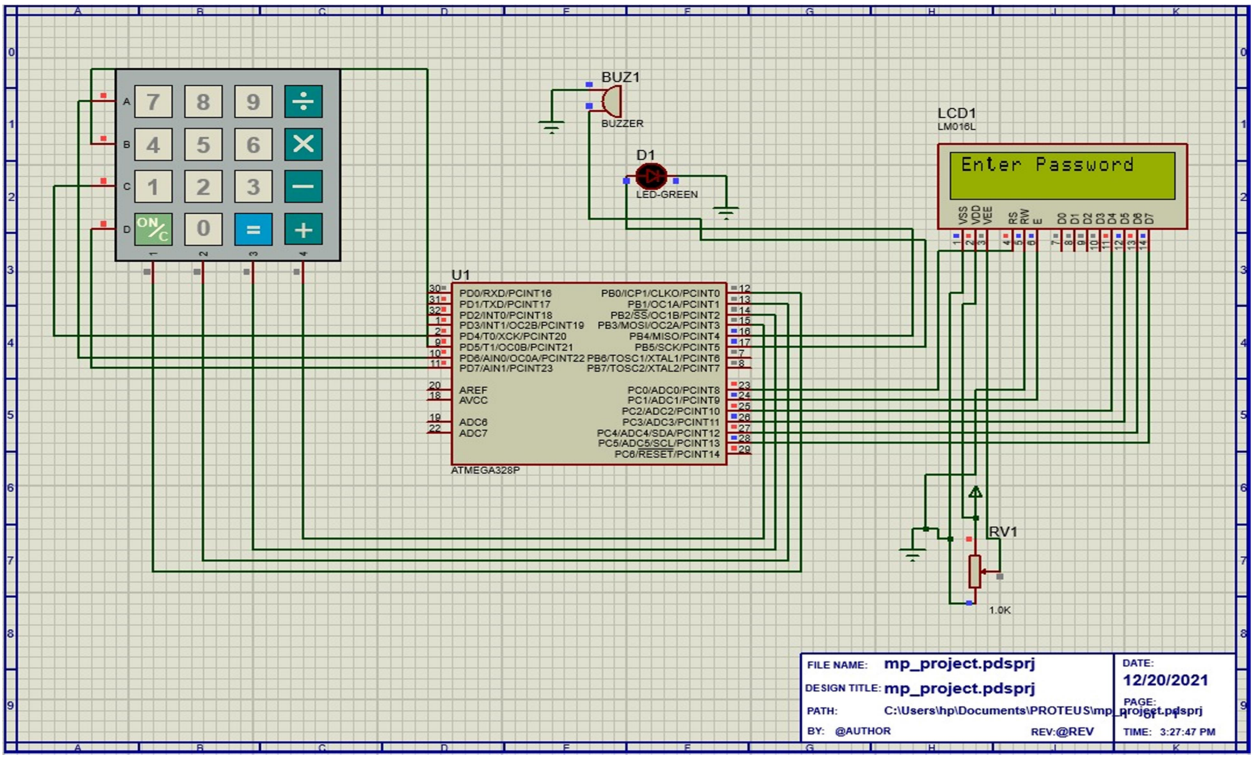


Figure4.3 software implementation on Proteus

4.3.1. DISCUSSION

- We have designed a smart safe in which we have used atmega328p microcontroller, Bluetooth, LCD, keypad and buzzer.
- We have connected all the devices together in Proteus as well as in vero board and soldered them.
- We have interfaced the keypad first using atmega328p and also we have connected LCD and a buzzer which will simply turn on if the code is incorrect.
- We can see in the figure 4.3 that LCD is asking for password after inserting password if it is correct it will accept the access and if it is wrong it say wrong password and buzzer will turn on and we have inserted three different passwords and on every password Bluetooth will also tell us that who opened the safe and we will get a buzzing voice in safe which is indication of safe is or safe is being accessed by unauthorized person.
- We can access the safe using Bluetooth and if someone will try to open the safe the buzzer will turn on and we will simply get a message on our Bluetooth interface and we can also get rid of robbery or something in this manner.

CHAPTER 5

5. HARDWARE IMPLEMENTATION AND DESIGN

5.1. PROCEDURE

We can see in the figure 5.4 that LCD is asking for password after inserting password if tis i correct it will accept the access and if it is wrong it say wrong password and buzzer will turn on and we have inserted three different passwords and on every password Bluetooth will also tell us that who opened the safe and we will get a buzzing voice in safe which is indication of safe is or safe is being accessed by unknown person who is trying to steal.

We can access the safe using Bluetooth and if someone will try to open the safe the buzzer will turn on and we will simply get a message on our Bluetooth interface and we can also get rid of robbery or something in this manner.

5.2. SOLDRING

Soldering is the process of a making a sound electrical and mechanical joint between certain metals by joining them with a soft solder. This is a low temperature melting point alloy of lead and tin. The joint is heated to the correct temperature by soldering iron. For most electronic work miniature mains powered soldering irons are used. These consist of a handle onto which is mounted the heating element. On the end of the heating element is what is known as the "bit", so called because it is the bit that heats the joint up.

Solder melts at around 190 degrees Centigrade, and the bit reaches a temperature of over 250 degrees Centigrade. This temperature is hot enough to inflict a nasty burn, consequently care should be taken. Good soldering is a skill that is learnt by practice. The most important point in soldering is that both parts of the joint to be made must be at the same temperature. The solder will flow evenly and make a good electrical and mechanical joint only if both parts of the joint are at an equal high temperature. Even though it appears that there is a metal to metal contact in a joint to be made, very often there exists a film of oxide on the surface that insulates the two parts. For this reason it is no good applying the soldering iron tip to one half of the joint only and expecting this to heat the other half of the joint as well.

5.3. WIRE CONNECTIONS

Here we have connected all the pins of LCD as in the figure 4.3 and figure 5.3 and all 8 pins of keypad are interfaced and we can use all its 16 pads for our needs as we have seen its working in the introduction of literature review in chapter 2 please review chapter 2.2.5. We have also seen how all other useful equipment in our project are working and we connected them all together and it is working as our requirements and hence we have reached our destination.

5.4. HARDWARE IMPLEMENTATION

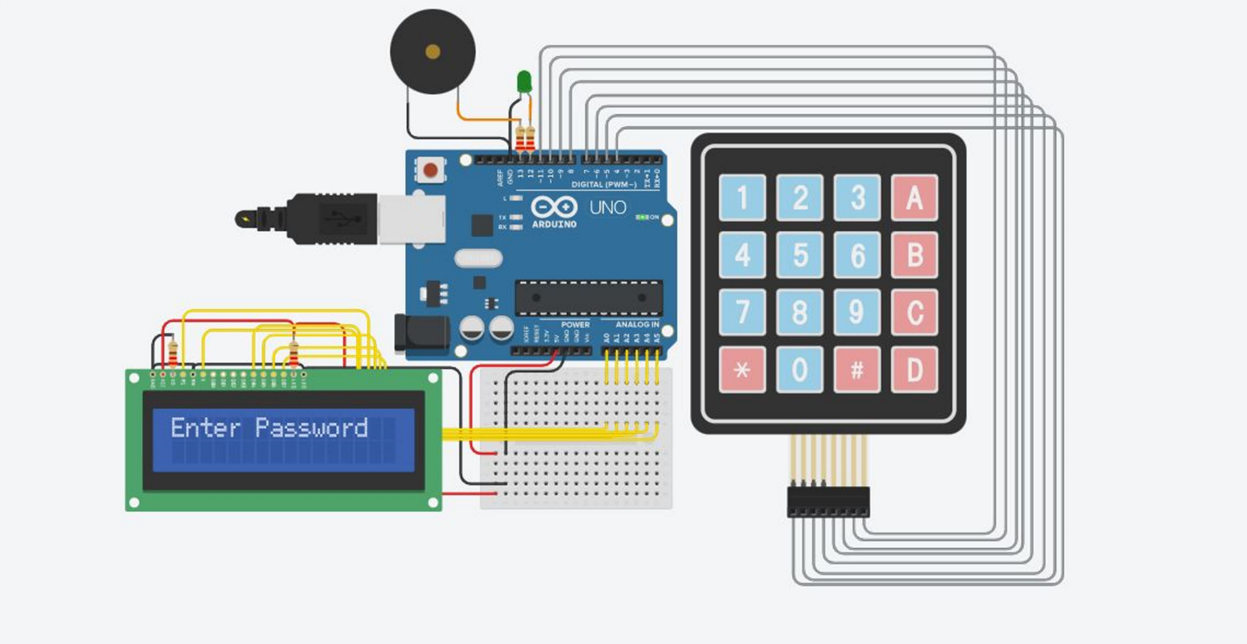


Figure 5.4 hardware implementation on Vero board

5.4.1. DISCUSSION

We made this circuitry on Vero board for smart safe in which we have used atmega328p microcontroller, Bluetooth, LCD, keypad and buzzer. We connected all the devices together on Vero board and soldered them. We have interfaced the keypad first using atmega328p and also we have connected LCD and a buzzer which will simply turn on if the code is incorrect. LCD will ask for password after inserting password if it is correct it will accept the access and if it is wrong it say wrong password and buzzer will turn on and we have inserted three different passwords and on every password Bluetooth will also tell us that who opened the safe and we will get a buzzing voice in safe which is indication of safe is or safe is being accessed by unauthorized person. We can access the safe using Bluetooth and if someone will try to open the safe the buzzer will turn on and we will simply get a message on our Bluetooth interface and we can also get rid of robbery or something in this manner.

CHAPTER 6

6. FLOW CHART, TESTING, RESULTS AND WOKING

6.1. FLOW CHARTS

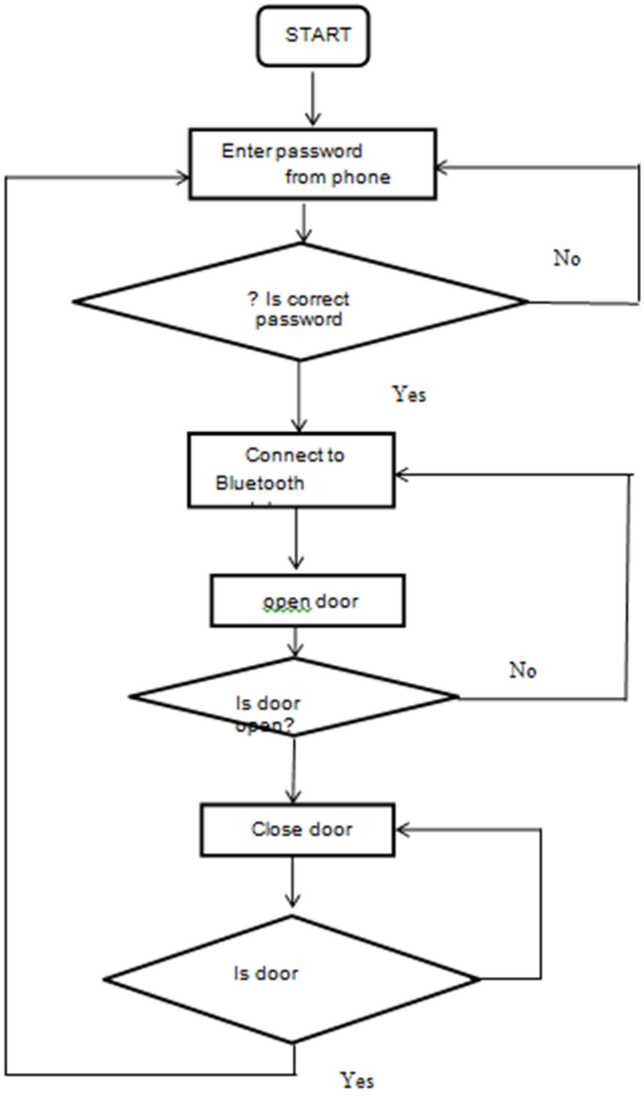


Figure 3 flow chart of system

6.2. TESTING

This chapter discuss the various test carried out during the cause of this project.

Various results from different test were gotten. And also brief discussion about the guiding principle and process of the entire project.

6.3. RESULTS

The program for the Arduino Uno microcontroller was written in C language and was then compiled into an executable file using the Arduino IDE. The executable file was then imported into the Proteus Design Suite, where the hardware circuit shown was designed and simulated. Figure shows the installed android app, the Proteus simulation of the door security system results for each process of entering the correct and wrong passwords respectively. Upon successful completion of the software simulation, the system hardware was constructed on a bread board and programming of the Arduino microcontroller was carried out using Arduino IDE. The hardware construction with connections and various operations of the system will be discussed in section 6.4.

6.4. WORKING

The response of the hardware of the security door when communicating with the Arduino board.

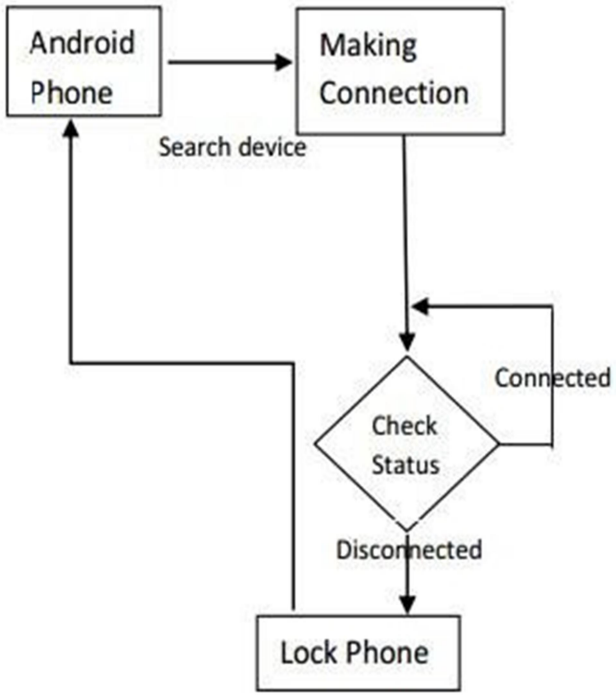


Figure 4 system flow

6.5. DISCUSSION

This project gives idea of how to control door locks. Carbon paper door lock system was used as a prototype for indoor and outdoor key lock system. It also provide a security and easy for smartphone users. This project based on smartphone and Arduino platform both of which are Free Open Source Software. So the implementation rate is inexpensive and it is reasonable by a common person. Accomplishment of wireless Bluetooth connection in microcontroller permits the system installation in more easy way. The system has been successfully designed and prototyped to control the door condition using smartphone Bluetooth-enabled phone and Bluetooth modules via Bluetooth HC-05. A simple prototype is discussed for this research purpose.

CHAPTER 7

7. SUMMARY AND REFERENCES

7.1. ABSTRACT

In early centuries security systems has been very poor. These research projects consist of a smart safe lock system which provides a great solution to improve the safety management of valuables. Arduino IDE software and a Bluetooth module hc-06 were used to connect between the smartphone, the microcontroller and the safe lock to give an easy access to authorized persons. The person with the authority to open the safe can have access within password by installing the required application which has open / close button. The hc-06 serves as a receiver and transmitter but also communicates with the microcontroller which serves as a processing unit in this project and decides whether the password entered by the user is right or wrong and then send the electric lock to either open or close the door. If the password is right the user can have access to the safe and when the password is wrong the user will have no access whatsoever enhancing the safety and security of secret documentations and money.

7.2. SUMMARY

Recently proposed safe lock systems based on Biometrics Techniques, Password Based and RFID have been studied and developed. This research project is centered on safe locking system via smartphone controlled locking system with the help of Arduino and Bluetooth module.

7.3. CONCLUSION

The main aim of this paper is to design a smart safe lock system using Arduino and Bluetooth application, so that people can feel safe about their valuables whether they are away from home or in the house. This project is based on Arduino, and the coding is done on Arduino ide platform using the Arduino application. At the end of this research the aim and objectives of the project was achieved. People can now feel more secure about their valuables and money all the time. Doors can be controlled conveniently to those with access. Physically challenged people can open or lock doors of safe from their password without asking help of anybody. It is safe to say that the main objectives and the aim of the project were achieved at the end of the project.

7.4. RECOMMENDATION

During this research some we had been developed a strong base over Arduino programming and microcontroller usage so we thought that we can also make door lock

system using the same procedure just we need a motor instead of lock and also we can add fingerprint module to access via fingerprint of authorized persons so for future research purpose we recommend other researchers to make the doors lock which can access through fingerprint and also from password as well as from Bluetooth connection also the door can only be accessed at a particular distance as we used Bluetooth module which can be accessed from a particular distance as discussed earlier in Bluetooth specifications so for future research the distance can be put into consideration and made to be longer using expensive and long distance access devices for better performance.

7.5. REFERENCE

http://circuitdigest.com
http://electronics.howstuffworks.com
http://www.creativeworld9.com/2011/04/abstract-andfullpaper-on-bluetooth.html
http://www.slideshare.net/kashyapshah11/bluetooth
http://www.techbitar.com/modify-the-hc-
https://en.wikipedia.org/wiki/androiddevelopers
https://www.bluetooth.com/whatisbluetooth technology/bluetooth-technology-basics
URL:http://.android.com/about/versions/index.html.

M.A.E. Mowad, A. Fathy and A. Hafez, "Smart Home Automated Control System Using Android Application and Microcontroller", International Journal of Scientific & Engineering Research, vol. 5, Issue. 5, pp 935-939, May 2014.

Shafana A.R.F.1, Aridharshan A. International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 5, Issue 3, pp: (26-30), Month: July - September 2017.

V. Madan and S.R.N. Reddy, "GSM-Bluetooth based Remote Monitoring and Control System with Automatic Light Controller", International Journal of Computer Applications, Vol. 46, No 1, pp 20-28, May 2012