

Özyeğin University Electrical and Electronics  
Engineering Department

Hotel Safe  
EE321 Term Project

DATE: 08/06/2021

M. İbrahim Kılıç

S014912

# Contents

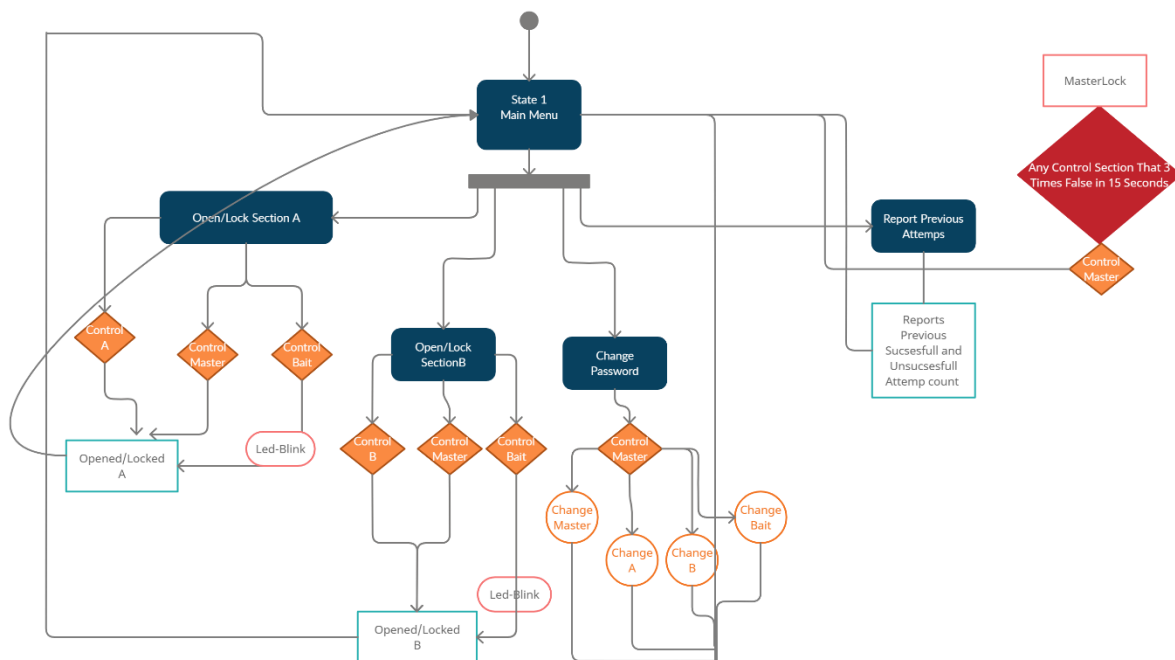
FRONT COVER

CONTENTS

1 INTRODUCTION	3
2 COMPONENTS	3
2.1 Arduino Mega2560 R3 .....	3
2.2 Keypad Phone .....	3
2.3 LM016L .....	4
2.4 Motor .....	4
2.5 L293D .....	4
2.6 Led-Red .....	4
3 FUNCTIONS	5
3.1 Setup .....	5
3.2 Main Menu .....	6
3.3 Entering A-B-Master .....	6
3.4 Control A-B .....	7
3.5 Control Master .....	8
3.6 Change A-B-Master-Bait .....	8
3.7 Master Lock .....	9
3.8 Open/Close Motor A-B .....	9
3.9 Report Previous Attempts .....	10
3.10 Interrupt .....	10
4 CONCLUSION	11

# 1 INTRODUCTION

In this project, I created a Hotel Safe lock/unlock mechanism. It has 2 sections with individual passwords (A and B), a master password to open any section and a bait password that when entered will alert the police and informs us with a blinking led. It is a simplified application of an actual safe system. In this project I used ARDUINO MEGA2560 R3 board which uses the microcontroller ATMEGA2560-16AU. I will give the details of the materials used in the Materials section. I arranged the processors clock frequency as 16MHz. Detailed explanation of how the project works will be with the code explained function by function. Here is a simplified state diagram that will show how should it work like.



## 2 MATERIALS USED (Proteus)

### 2.1 Arduino Mega2560 R3

I worked on an Arduino before and I know how to code in C so I choose Arduino as my platform and because I need a lot of output pins which this board provides me so many of them so, I choose this board.

### 2.2 Keypad Phone

Keypad Phone is the one with the desired numpad content.

## 2.3 LM016L

2x16 LCD that will display our conditions.

## 2.4 Motor

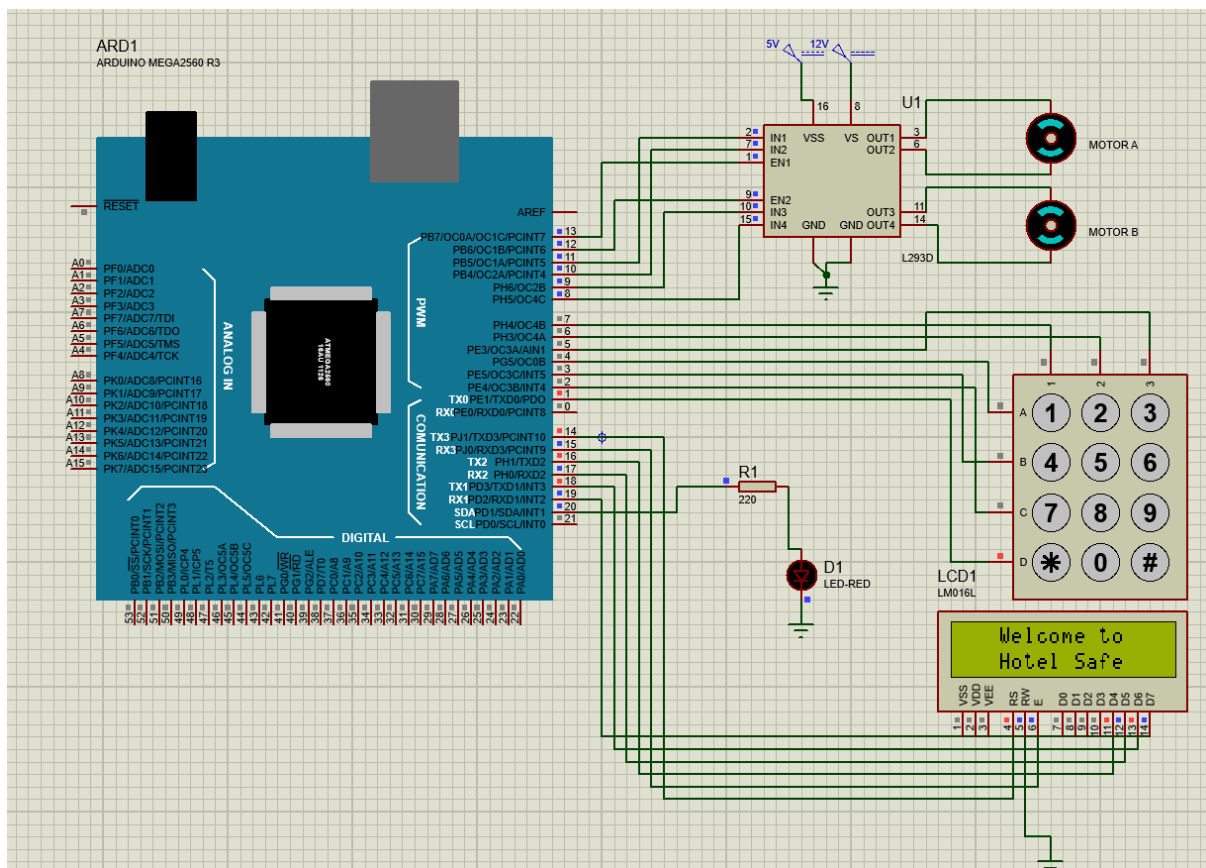
Simple DC Motor model from the library ACTIVE. Connected with to L239D.

## 2.5 L293D

Push-Pull four channel driver with diodes. Connected to PWM outputs of the board and its own outputs are connected to Motor A and Motor B.

## 2.6 Led-Red

Emergency LED that will blink for 1 seconds.



## 3 FUNCTIONS

### 3.1 Setup

Before the function setup, I declared global parameters and needed codes for Numpad to work. In setup I activated the lcd and I have set the necessary output pins for the motors to work properly. After that I reset the Timer1 control register A, set the CTC mode, set prescaler to 1024, reset Timer1 and set the compare value of 46875 (calculation will be at the interrupt part of the report), enable the Timer1 compare interrupt and then enable global interrupts. The user will see "Welcome to" at the first row of LCD and "Hotel Safe" at the second row of the LCD for 2.5 seconds. Then the function main menu will operate.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(Led, OUTPUT);
  pinMode(EN1, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  //Reset Timer1 Control Reg A
  TCCR1A = 0;
  //Set CTC mode
  TCCR1B &= ~(1 << WGM13);
  TCCR1B |= (1 << WGM12);
  //Set prescaler of 1024
  //CSn2 CSn1 CSn0
  //1 0 1 clkI/O/1024 (From prescaler)
  TCCR1B |= (1 << CS12);
  TCCR1B &= ~(1 << CS11);
  TCCR1B |= (1 << CS10);
  //Reset Timer1 and set compare value
  TCNT1 = 0;
  OCR1A = 46875;
  //Enable Timer1 compare interrupt
  TIMSK1 |= (1 << OCIE1A);
  //Enable global interrupts
  sei();

  lcd.setCursor(0, 0);
  lcd.print(" Welcome to");
  // entPasw[4] = '\0';
  lcd.setCursor(0, 1);
  lcd.print(" Hotel Safe");
  delay(2500);
  statel();
}
```

### 3.2 Main Menu

This is the main menu function, when every process ends, we will turn back to main menu. In this function I am setting L293D's EN1 and EN2 pins high. I am displaying "Main Menu" and "1-A 2-B 3-M 4-R" in LCD screen. I am getting the pressed key and according to what user pressed, user will go to that stage. For getting the key I am using Arduino's `#include <Keypad.h>` functions.

```
void statel () {
    digitalWrite(EN1, HIGH);
    digitalWrite(EN2, HIGH);
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("Main Menu");
    lcd.setCursor(1, 1);
    lcd.print("1-A 2-B 3-M 4-R");
    while (1) {
        char key = keypad.getKey();
        if (key) {

            if (key == '1') {
                enteringA();
            }
            if (key == '2') {
                enteringB();
            }
            if (key == '3') {
                enteringMaster();//change Password
            }
            if (key == '4') {
                reportPrevious();
            }
        }
    }
}
```

### 3.3 Entering A-B-Master

In this function I declared a String `pressedKey`, that will be my string to collect what user pressed in the numpad. We have a while loop that runs 4 times. This is the allowed number length for password. After every numpad press `keypad.getKey()` will get the char collected and assigns it to the key. `pressedKey = pressedKey + key;` Using this assignment we will have a string as entered password. This function has the same processes with `enteringB()` and `enteringMaster()`.

```
void enteringA() {
    i = 1;
    String pressedKey;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter Password A");
    lcd.setCursor(0, 1);
    while (i < 5) {
        char key = keypad.getKey();
        if (key) {
            lcd.print("");
            pressedKey = pressedKey + key;
            i++;
        }
    }
    controlA(pressedKey);
}
```

### 3.4 Control A-B

We were redirected here after enteringA(). Here we check our password, which we obtained as a string. We check whether the entered password matches password A (bool x) or master password (bool y). If it is equal to either of these, it controls the engine according to its previous state than turns the engine off (closeMotorA) or on (openMotorA). If the given password is the Bait password (bool z), the motors will work same as others but there will be red-led blinking and letting the police known. This blinking stage will be 1 seconds as demanded so that there will be a delay(1000) in the other two outcomes. If the given password by user is incorrect the LCD will display "Wrong Password", "Try again" and the user will go to the function enteringA(). If the user enters the wrong password 3 times in a row within 15 seconds, the program will be redirected to the masterLock() function. controlB() operates same as controlA().

```
void controlA(String givenPasw) {
    bool x = givenPasw.equals(paswA);
    bool y = givenPasw.equals(paswMaster);
    bool z = givenPasw.equals(paswBait);
    if (x) {
        if (Status_MotorA == 0)
            openMotorA();
        else if (Status_MotorA == 1)
            closeMotorA();
        delay(1000);
        SFA = 0;
        sucesfullAttempts++;
        EEPROM.write(addressSucesfullAttempts, sucesfullAttempts);
        statel();
    }
    else if (y) {
        if (Status_MotorA == 0)
            openMotorA();
        else if (Status_MotorA == 1)
            closeMotorA();
        delay(1000);
        SFA = 0;
        sucesfullAttempts++;
        EEPROM.write(addressSucesfullAttempts, sucesfullAttempts);
        statel();
    }
    else if (z) {
        if (Status_MotorA == 0) {
            openMotorA();
            digitalWrite(Led, HIGH);
            delay(250);
            digitalWrite(Led, LOW);
            delay(250);
            digitalWrite(Led, HIGH);
            delay(250);
            digitalWrite(Led, LOW);
            delay(250);
        }
        else if (Status_MotorA == 1) {
            closeMotorA();
            digitalWrite(Led, HIGH);
            delay(250);
            digitalWrite(Led, LOW);
            delay(250);
            digitalWrite(Led, HIGH);
            delay(250);
            digitalWrite(Led, LOW);
            delay(250);
        }
        SFA = 0;
        sucesfullAttempts++;
        EEPROM.write(addressSucesfullAttempts, sucesfullAttempts);
        statel();
    }
    else if (SFA < 2) {
        unsucesfullAttempts++;
        EEPROM.write(addressUnsucesfullAttempts, unsucesfullAttempts);
        SFA++;
        lcd.clear();
        lcd.print("Wrong Password");
        lcd.setCursor(0, 1);
        lcd.print("Try again");
        delay(1500);
        enteringA();
    }
    else {
        unsucesfullAttempts++;
        EEPROM.write(addressUnsucesfullAttempts, unsucesfullAttempts);
        masterLock();
    }
}
```

### 3.5 Control Master (Change Password)

User pressed 3 in numpad to get here this means that user wants to change a password. In this function we are using the features we used in Main Menu and Control A. Whether the user enters the correct or wrong password, the function will work as same as the previous one (controlA). The user will decide which password he/she wants to change with pressing one of the top 4 keys (1 for Master, 2 for A, 3 for B, 4 for Bait).

For changing the password, the program will go to the other stage which is one of these changeMaster, changeA, change or changeBait.

```
void controlMaster(String givenPasw) {
    bool y = givenPasw.equals(paswMaster);
    bool z = givenPasw.equals(paswBait);
    if (y) {
        succesfullAttempts++;
        EEPROM.write(addressSuccesfullAttempts, succesfullAttempts);
        lcd.clear();
        lcd.print("CP 1M 2A 3B 4T");
        SFA = 0;
        while (1) {
            char key = keypad.getKey();
            if (key) {
                lcd.setCursor(0, 0);
                lcd.print("    Main Menu");

                if (key == '1') {
                    changeMaster();
                }
                if (key == '2') {
                    changeA();
                }
                if (key == '3') {
                    changeB();
                }
                if (key == '4') {
                    changeBait();
                }
                delay(100);
            }
        }
    }
    else if (SFA < 2) {
        unsuccessfullAttempts++;
        EEPROM.write(addressUnsuccessfullAttempts, unsuccessfullAttempts);
        SFA++;
        lcd.clear();
        lcd.print("Wrong Password");
        lcd.setCursor(0, 1);
        lcd.print("Try again");
        delay(1500);
        enteringMaster();
    }
    else {
        unsuccessfullAttempts++;
        EEPROM.write(addressUnsuccessfullAttempts, unsuccessfullAttempts);
        masterLock();
    }
}
```

### 3.6 Change A-B-Master-Bait

When the product Hotel Safe first opened, the 4 passwords are set. This function's goal is to change the original passwords if the user wants to change them. When the user presses "3" in the main menu and correctly enters the master password and enters "2" there. The program will get into this function, LCD will display "Enter New Pass A". The function will work same as enteringA() when getting the keys pressed together. User presses 4 buttons of his/her choice and password A will be that chosen numbers. After that LCD will let user know, password is changed by displaying "Pasw is changed". The text will be visible for 1 seconds and program goes back to the main menu.

```
void changeA() {
    i = 1;
    String pressedKey;
    lcd.clear();
    lcd.print("Enter New A");
    lcd.setCursor(0, 1);
    while (i < 5) {
        char key = keypad.getKey();
        if (key) {
            lcd.print("*");
            pressedKey = pressedKey + key;
            i++;
        }
    }
    paswA = pressedKey;
    // EEPROM.write(addressAPassword, paswA);
    lcd.clear();
    lcd.print("Pasw is changed");
    delay(1000);
    statel();
}
```



### 3.7 Master Lock

When the user enters wrong password 3 times in a row within 15 seconds, we will enter this function. This function will operate with the same methods when getting the keys and looking if they are correct or not.

When the user comes to this stage user needs to enter the master password, the other two password A and password B wont work here. If user enters the master password correctly user will go to the main menu but if user enters wrong password then the program goes to timeLock() function which locks the system while displaying a count down from 20 seconds to 0.

```
void masterLock() {
    i = 1;

    String pressedKey;
    lcd.clear();
    lcd.print("System Locked MP");
    lcd.setCursor(0, 1);
    while (i < 5) {
        char key = keypad.getKey();
        if (key) {

            lcd.print("*");
            pressedKey = pressedKey + key;
            i++;
        }
    }
    bool y = pressedKey.equals(paswMaster);
    if (y) {
        sucesfullAttemps++;
        EEPROM.write(addressSucesfullAttemps, sucesfullAttemps);
        statel();
    }
    else {
        unsucesfullAttemps++;
        EEPROM.write(addressUnsucesfullAttemps, unsucesfullAttemps);
        timeLock();
    }
}

void timeLock() {
    for (int j = 20; j >= 1; j--) {
        lcd.clear();
        lcd.print("    No access");
        lcd.setCursor(0, 1);
        lcd.print("    " + String(j));
        delay(1000);
    }
    masterLock();
}
```

### 3.8 Open/Close Motor A-B

When the program comes into this function the motor will turn two times counter clockwise while LCD displays "Sect A opening", after two cycles completed (calculated as 973 ms) LCD displays "Sect A opened" Then sets the status of motor as 1 meaning the A section is open. To close the section A, user needs to enter the section A's password (or master). As seen on the right, when the motor is shutting down, it moves in the opposite direction and turns clockwise twice.

```
void openMotorA() {
    lcd.clear();
    lcd.print("Sect A opening");
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    delay(973);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    lcd.clear();
    lcd.print("Sect A opened");
    delay(500);
    Status_MotorA = 1;
}

void closeMotorA() {
    lcd.clear();
    lcd.print("Sect A locking");
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    delay(973);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    lcd.clear();
    lcd.print("Sect A locked");
    delay(500);
    Status_MotorA = 0;
}
```

### 3.9 Report Previous Attempts

```
void reportPrevious() {
  lcd.clear();
  //lcd.print(String(sucesfullAttemps) + " Suc " + String(unsucesfullAttemps) + " Unsuc");
  // I am getting the attemp counts from EEPROM
  lcd.print(String(EEPROM.read(adressSucsesfullAttemps)) + " Suc " + String(EEPROM.read(adressUnsucsfullAttemps)) + " Unsuc");
  delay(3000);
  state1();
}
```

The function is simple it only displays the successful and unsuccessful attempts when “4” pressed in main menu. We were collecting these numbers from other functions and registering them to the EEPROM registers. We can see the EEPROM.write() every function we entered a password (not the changing password because we cannot enter a wrong or correct password there it is a new one). In this function I used EEPROM.read() to get the successful and unsuccessful counts from EEPROM registers. If needed we can get these number even, we close and open the system because EEPROM is electrically erasable programmable read-only memory meaning if the number in the registered address is not deleted manually, it will remain there. If I erase the following two lines in my code,

```
int sucsefullAttemps = 0;
```

```
int unsucsfullAttemps = 0;
```

and not declare as 0, we can count the total successful and unsuccessful attempts even if we close the system and open it again. Because it was not given as a need in project pdf I didn't do that.

### 3.10 Interrupt

timer1 max = 65535,  $1/16\text{MHZ} = 62.5\text{ms}$ , (presc)  $8/16\text{MHZ} = 500\text{ms}$

$15 / (1/16\text{Mhz}) = 240\text{m}$ , (presc)1024 -> 234,375

3 seconds = 46,875 (these are my interrupt calculations) and as you can

see right side this is how I use interrupt. My calculation was 3 seconds

every time I enter the interrupt but although my calculations were correct it worked

differently so with the help of TA Cengiz Emre Dedeagac I realized I go into interrupt every 0.45 sec so I chanced the original plan to  $s == 34$  it will give me 15.3 seconds (so its roughly 15 secons). It will interrupt every 0.45 seconds and keep increasing s until s equals to 34.

When S equals to 34, it will reset SFA and s. In this way, SFA will be reset every 15.3 seconds.

```
int s = 0;
ISR(TIMER1_COMPA_vect) {
  if (s == 34) {
    SFA = 0;
    s = 0;
  } else
    s++;
```

## 4 CONCLUSION

It was a project that covered all the topics we covered in lab classes. We had the opportunity to apply what we have seen in the labs so far in one big project. When I first started the project, I planned to use a servo motor and wrote a function accordingly, but the servo motor library gave an error due to the version of my Arduino ide. When I faced this problem, I decided to use dc motor and L293 motor driver and it worked. I prepared the methods and functions that I created and used within the scope of the project in accordance with the project document given to us. I also performed testing to observe how it works in different scenarios. Thanks to these operations, a few errors appeared, and I solved them as well. In my final test run, the project was working exactly as requested.