Password based circuit breaker

A report submitted in fulfilment of the requirements for J component of the ECE4002 Advanced Microcontrollers" course.

Under the super vision of

Vidyapathi C.M

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)Vellore Institution of Technology

Vellore

2021

DECLARATION BY THE CANDIDATES

18BEC0052 – TARUN SAI CHOWDARY

18BEC0031 - JAVED SHAIK

18BEC0087 – AKULA VAMSI

We hereby declare that the project report entitled “Password based circuitbreaker

” submitted to VIT University, Vellore is a record of J-Component project workcarried out by us under the guidance of **VIDHYAPATHI C.M**

. We further declare that the report has been written in our own words and haveprovided proper references whenever we referred to other articles or the internet.

ACKNOWLEDGEMENT

We would like to express our gratitude to, our faculty **VIDHYAPATHI C.M**

for the course Advanced Microcontroller and its applications and for giving us anopportunity to work in this project and for guiding us through it. We would also like to thank VIT University Management, Dean Academics and all the facultiesof SENSE school for providing us an opportunity to carry out this study at the university.

ABSTRACT

Our project mainly focuses on lineman protection system who works on the electric wires. If some unknown person attempted to change the wires then itwould be great problem to the person and also the entire area. So arranging apassword-based system to it is our main idea.

Table of Contents:

Chapter 1

Introduction

1.1 Problem Statement

We often see in news about the sudden death of person who works onthe electric wires or some on else nearby touching the wires or changing them experience a sudden shock that may leads to death

1.2 Importance of solving the stated problem

Therefore, it is important enough to find out some solution to this otherwisethese type of scenarios are more likely to repeat. Therefore, we arranged a Password based system to it to so that expect lineman

Chapter 2

2.1 Components Required

The components required for this work are very cheap and easily available.List of components:

AVR MICROCONTROLLER (ATmega32)

- LCD DISPLAY
- Matrix Keypad
- Lamps
- POWER SOURCE DC
- RESISTORS
- POTENTIOMETER

2.2 Block Diagram

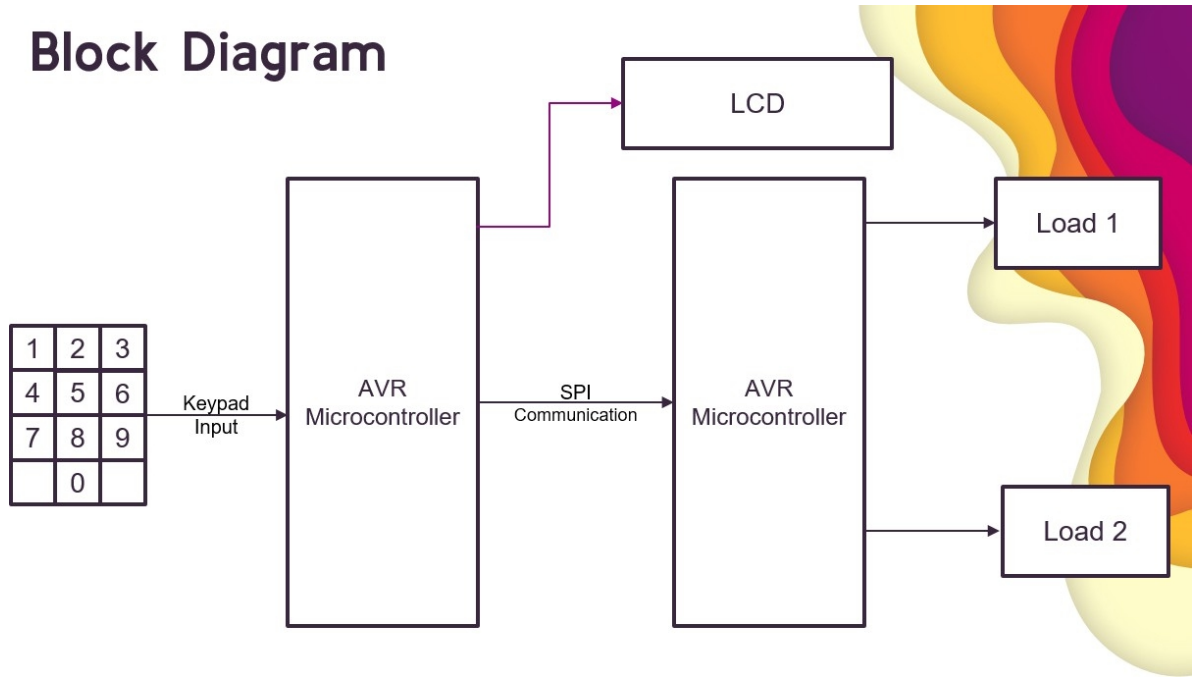
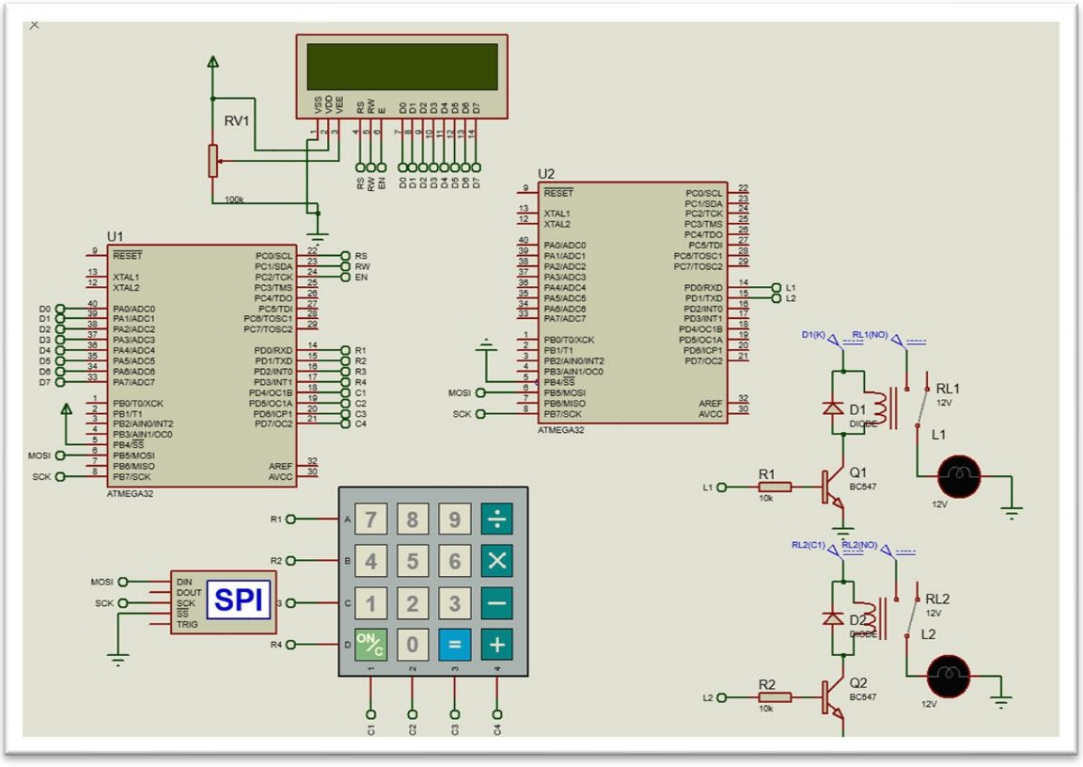


Fig 2.1

The figure 2.1 shows the block diagram of our work.

2.3 Circuit Diagram



2.4 Methodology

- When a user approaches the system, they are asked to enter their choice ifthey want to check or change the status of machine.
- If checking the status is chosen, then the current status of loads are shownon the LCD screen.
- If changing the status is chosen they are asked to enter the correct keycombination.
- If the entered key combination is correct, then the corresponding load ischanged.
- If the entered key combination is wrong, then the user is asked to tryagain later.

Chapter 3

Results and Discussion

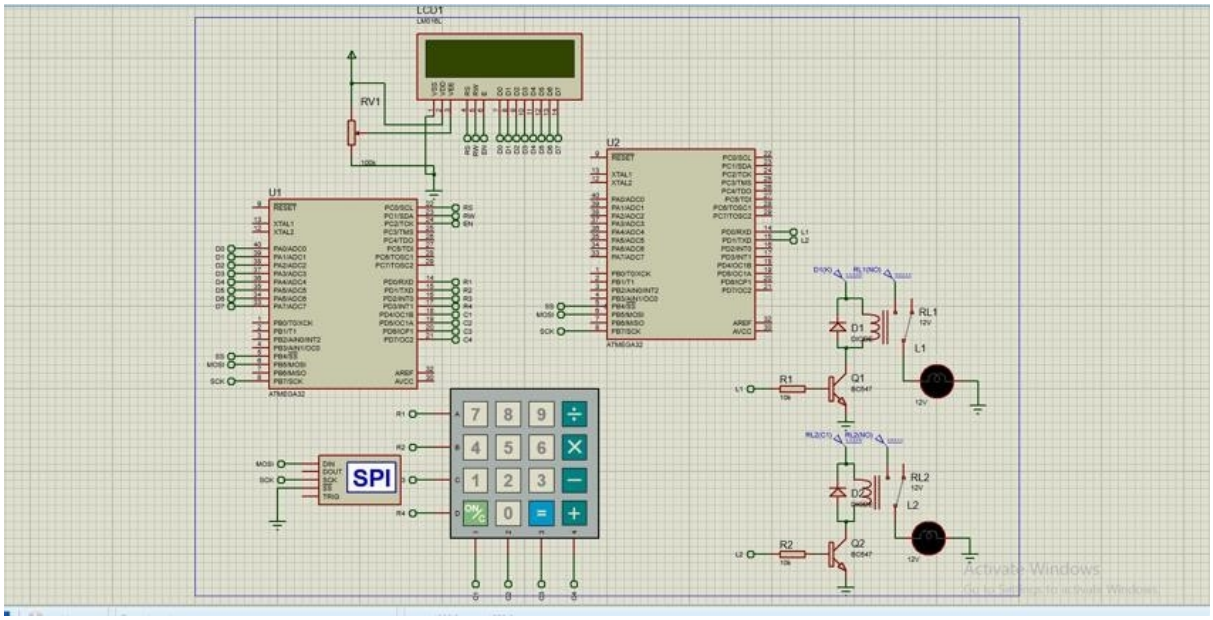


Fig 3.1.1

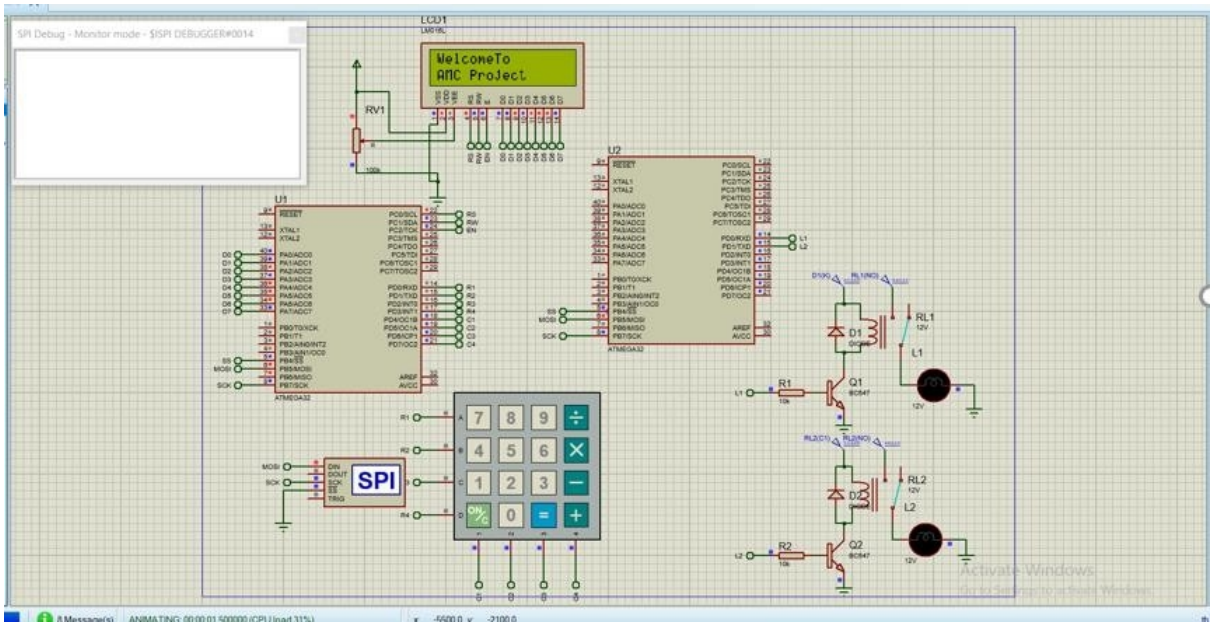


Fig 3.1.2

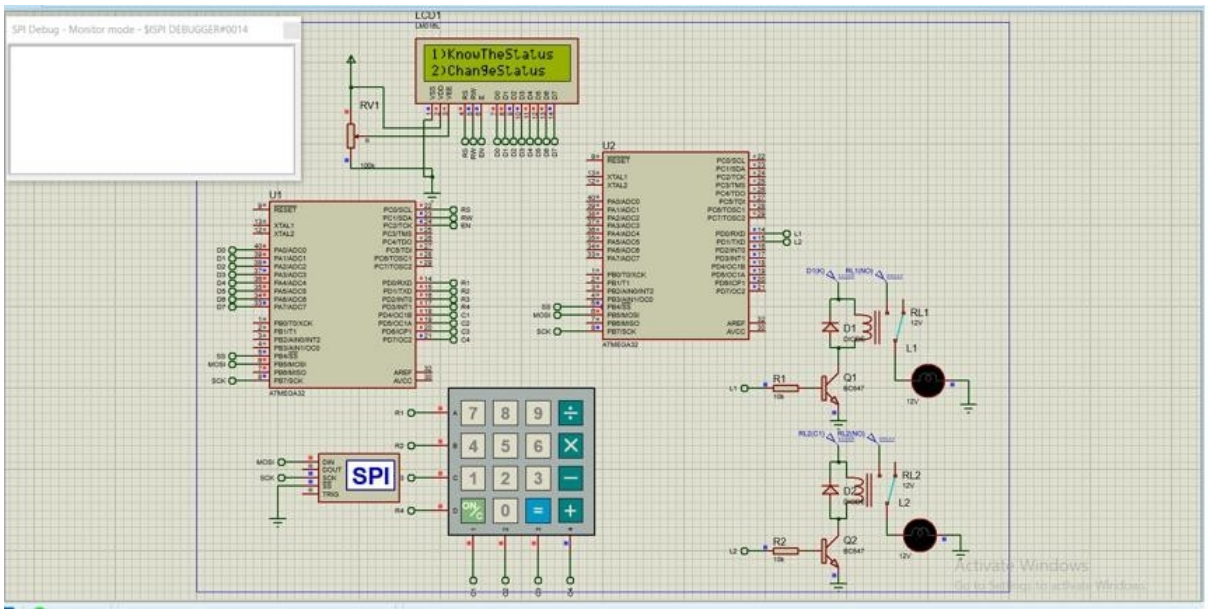
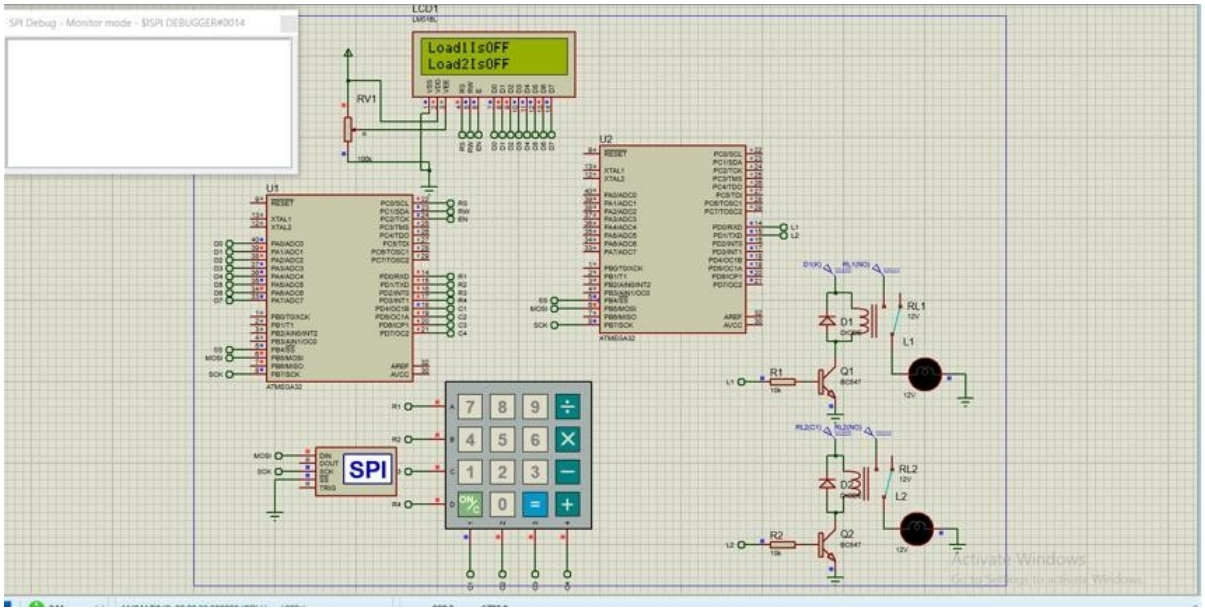
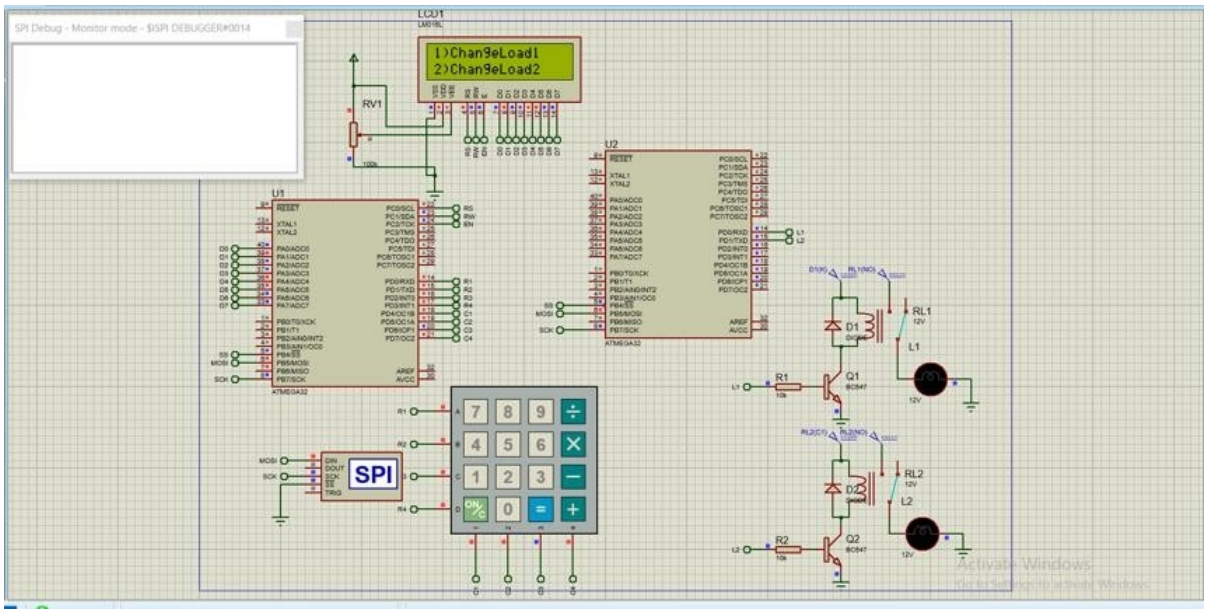


Fig 3.1.3



By default it displays both loads are off

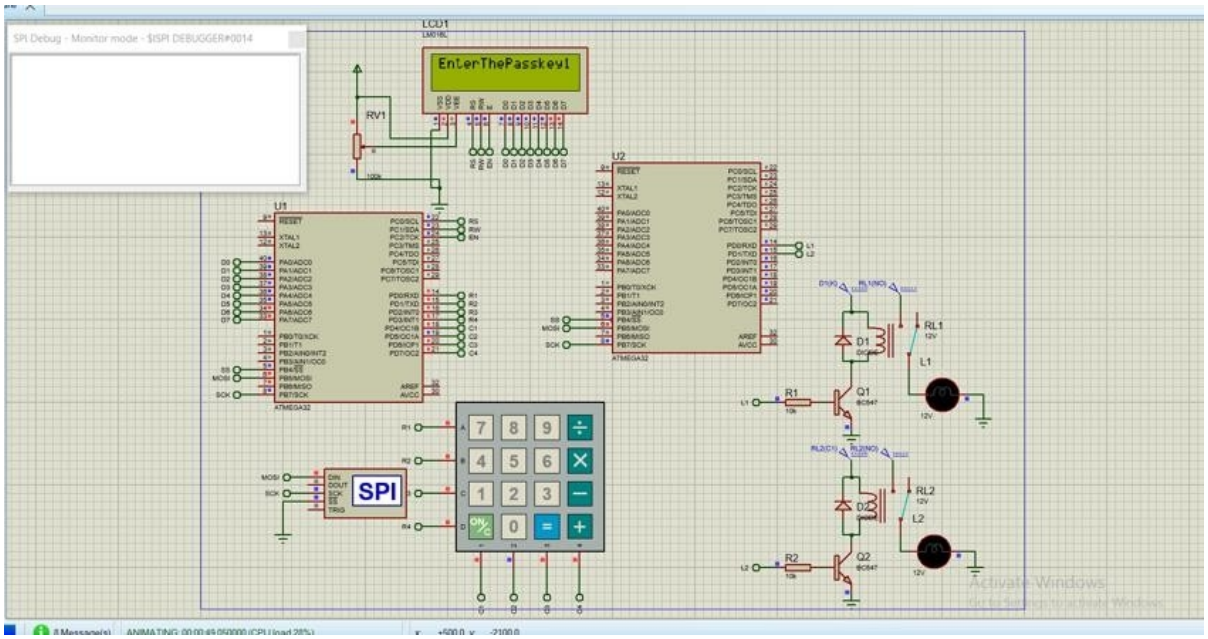
Fig 3.1.4



Press 1- To change the load 1

Press 2 – To change the load 2

Fig 3.1.5



Now enter the password

Fig 3.2.1

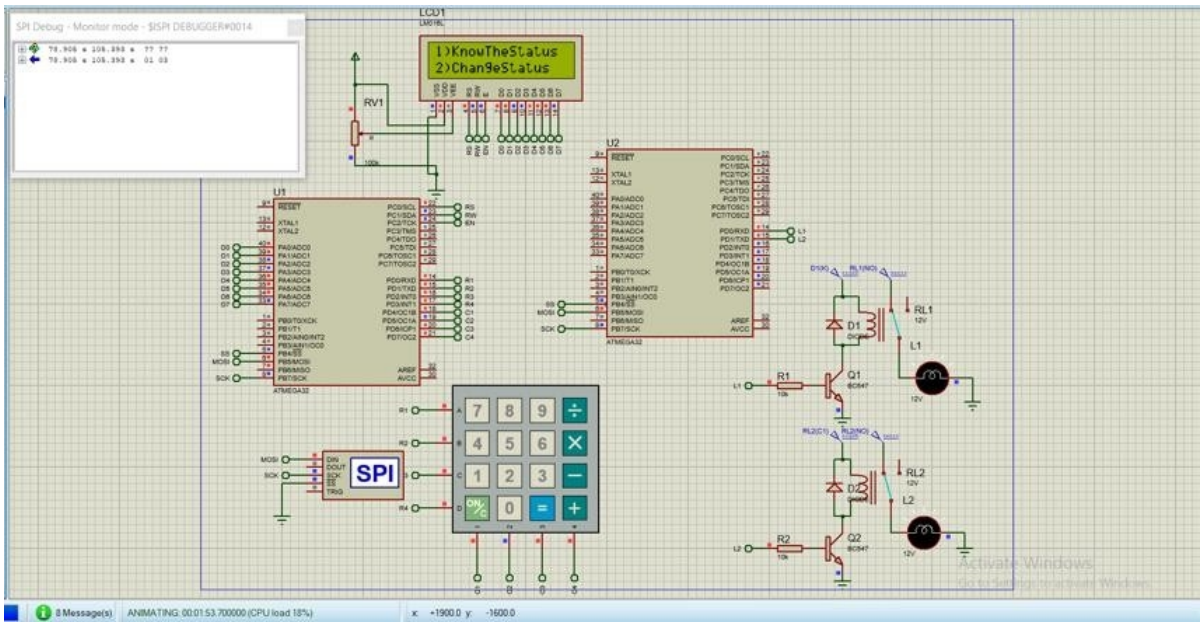
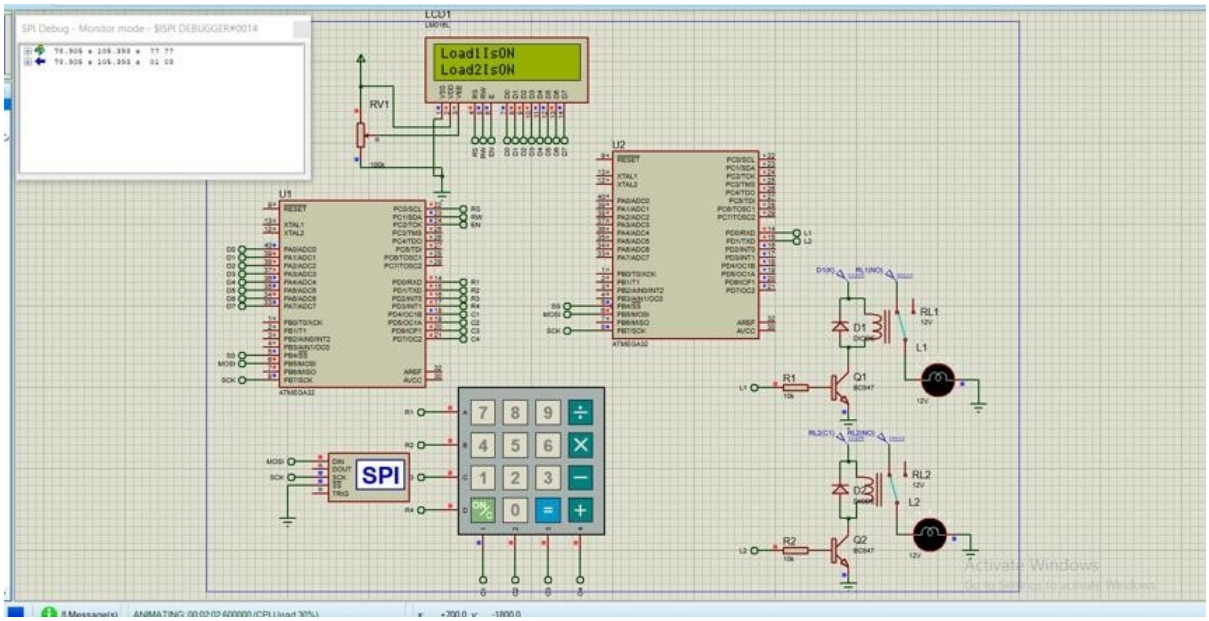


Fig 3.2.2



Hence if it right then load is on

Fig 3.3.1

Chapter 4

Conclusions and Recommendations

4.1 Conclusion

The model will first check if the password is valid. If valid, then he can changethe status of the load it will check if the user has enough credentials to pass through Else, it will ask to enter the password again

4.2 Advantages and Disadvantages

The advantage is that person can only change the wires if entered password is right otherwise not and disadvantage is that we have to add the one more optionif he forgets the password .

Appendix

Code:

```
# define F_CPU 16000000UL
#include<avr/io.h>
#include<util/delay.h>

#define ctrl PORTC
#define en PC2
#define rw PC1
#define rs PC0
int Load1=0;
int Load2=0;

void LCD_init(void);
void LCD_clear(void);
void LCD_cmd(unsigned char cmd);
void LCD_data(unsigned char data);
void LCD_string();
void SPI_master_init(void);
void SPI_transmit(unsigned char data);
void keypad1(void);
void keypad2(void);
void keypad3(void);
void keypad4(void);
void keypad5(void);
void passkey1(void);
void passkey2(void);
char keyscan(void);
char keycheck(void);
```

```
int main(void)
 {
        DDRA = 0XFF;
        DDRC = 0X07;
        DDRD = 0XF0;
        SPI_master_init();
        LCD_init();
        LCD_string("WelcomeTo");
        LCD_cmd(0XC0);
        LCD_string("AMC Project");
        _delay_ms(10);
        while(1){
                LCD_clear();
                LCD_string("1)KnowTheStatus");
                LCD_cmd(0XC0);
                LCD_string("2)ChangeStatus");
                keypad1();
        }
        return 0;
 }

void keypad1(void)
 {
                char key1;
                key1=keyscan();i
                if(key1=='1'){
                 LCD_clear();
                 LCD_string("Status is");
                 LCD_clear();
```

```
        if (Load1==0 &&
                Load2==0){ LCD_string("
                Load1IsOFF");
                LCD_cmd(0XC0);
                LCD_string("Load2IsOFF");
        }

        if (Load1==1 &&
                Load2==0){ LCD_string("L
                oad1IsON");
                LCD_cmd(0XC0);
                LCD_string("Load2IsOFF");
        }
        if(Load1==0 &&
                Load2==1){ LCD_string("
                Load1IsOFF");
                LCD_cmd(0XC0);
                LCD_string("Load2IsON");
        }
        if(Load1==1 &&
                Load2==1){ LCD_string("L
                oad1IsON");
                LCD_cmd(0XC0);
                LCD_string("Load2IsON");
        }
        _delay_ms(100);
}
```

```
	if(key1=='2'){
		keypad2();
	}
	return;
}
void keypad2(void)
{
	LCD_init();
	LCD_string("1)ChangeLoad1");
	LCD_cmd(0XC0);
	LCD_string("2)ChangeLoad2");
	keypad3();
	return;
}
void keypad3(void)
{
	char key2;
	key2=keyscan();
	if(key2=='1'){
		LCD_clear();
		LCD_string("EnterThePasskey1");
		LCD_cmd(0XC0);
		passkey1();
	}
```

```
        if(key2=='2'){
                LCD_clear();
                LCD_string("EnterThePasskey2");
                LCD_cmd(0XC0);
                passkey2();
        }
        return;
}
void passkey1(void)
{
        int i;
        int count1;
        char arr1[4];
        char original1[4]={'1','2','3','4'};for(i=0;
        i<4; i++){
                arr1[i]=keyscan();
                LCD_data(arr1[i]);
                LCD_cmd(0X06);
        }
        for(i=0; i<4; i++){
                if(arr1[i]==original1[i]){cou
                        nt1 += 1;    }
                }
        if(count1 == 4){
                LCD_clear();
                LCD_string("PasskeyRight");
                _delay_ms(100);
                LCD_clear();
                LCD_string("SendingSignal");
```

```
        _delay_ms(10);
        if(Load1==0){
                if(Load2==0){
                        SPI_transmit(0X01);
                        Load1=1;
                }
                else{
                        SPI_transmit(0X03);
                        Load1=1;
                }
        }
        else{
                if(Load2==0){
                        SPI_transmit(0X00);
                        Load1=0;
                }
                else{
                        SPI_transmit(0X02);
                        Load1=0;
                }
        }

}
```

```
if(count1 == 4){
        LCD_clear();
        LCD_string("PasskeyRight");
        _delay_ms(100);
        LCD_clear();
        LCD_string("SendingSignal");
        _delay_ms(10);
        if(Load1==0){
            if(Load2==0){
                SPI_transmit(0X01);
                Load1=1;
            }
            else{
                SPI_transmit(0X03);
                Load1=1;
            }
        }
        else{
            if(Load2==0){
                SPI_transmit(0X00);
                Load1=0;
            }
            else{
                SPI_transmit(0X02);
                Load1=0;
            }
        }

 }
```

```
        else{
                LCD_clear();
                LCD_string("PasskeyWrong");


                _delay_ms(100);
                LCD_clear();
                LCD_string("TryAgainLater");
                _delay_ms(100);
        }
        return;
}

void passkey2(void)
{
        int i,count2;
        char arr2[4];
        char original2[4]={'8','5','2','0'};for(i=0;
        i<4; i++){
                arr2[i]=keyscan();
                LCD_data(arr2[i]);
                LCD_cmd(0X06);
        }
        for(i=0; i<4; i++){
                if(arr2[i]==original2[i]){cou
                        nt2 += 1;
                }
        }
```

```
if(count2 == 4){
        LCD_clear();
        LCD_string("PasskeyRight");
        _delay_ms(50);
        LCD_clear();
        LCD_string("SendingSignal");
        if(Load2==0){
                if(Load1==0){
                        SPI_transmit(0X02);
                        Load2=1;
                }
                else{
                        SPI_transmit(0X03);
                        Load2=1;
                }
        }
        else{
                if(Load1==0){
                        SPI_transmit(0X00);
                        Load2=0;
                }
                else{
                        SPI_transmit(0X01);
                        Load2=0;
                }
        }
}
```

```
        else{
                LCD_clear();
                LCD_string("PasskeyWrong");
                _delay_ms(100);
                LCD_clear();
                LCD_string("TryAgainLater");
                _delay_ms(100);
        }
        return;
}
char keyscan(void)
{
        char key = 'a';
        while(key=='a')
        {
                key = keycheck();
        }
        return key;
}

char keycheck(void)
{
        PORTD=0b11101111;   //ground row 0 (A-key row)
        _delay_ms(3);
        if((PIND&0b00000001)==0)
        {
                return '7';
        }
```

```
        if((PIND&0b00000010)==0)
        {
                return '4';
        } if((PIND&0b00000100)==0)
        {
                return '1';
        }
        PORTD=0b11011111;   //ground row 1 (B-key row)
        _delay_ms(3);
        if((PIND&0b00000001)==0)
        {
                return '8';
        } if((PIND&0b00000010)==0)
        {
                return '5';
        } if((PIND&0b00000100)==0)
        {
                return '2';
        } if((PIND&0b00001000)==0)

        return ‘0’;
}

        PORTD=0b10111111;   //ground row 3 (C-key row)
        _delay_ms(3);
        if((PIND&0b00000001)==0)
        {
                return '9';
        }
        if((PIND&0b00000010)==
```

```
        0)
        {
                return '6';
        }
        if((PIND&0b00000100)==
        0)
        {
                return '3';
        }
        PORTD=0b01111111;   //ground row 4 (D-key row)
        _delay_ms(3);
        if((PIND&0b00000001)==0)
        {
                return '/';
        } if((PIND&0b00000010)==0)
        {
                return '*';
        } if((PIND&0b00000100)==0)
        {
                return '-';
        } if((PIND&0b00001000)==0)
        {
               return '+';
        }
        //if no key pressed
        return 'a';
}
```

```
void LCD_init(void)
{
        LCD_cmd(0x38);
        _delay_ms(1);
        LCD_cmd(0x3
        8);
        _delay_ms(1);
        LCD_cmd(0x3
        8);
        _delay_ms(1);
        LCD_cmd(0x0
        6);
        _delay_ms(1);
        LCD_cmd(0x0
        c);
        _delay_ms(1);
        LCD_cmd(0x0
        1);
        _delay_ms(1);
        LCD_cmd(0x0
        2);
        _delay_ms(1);
        return ;
}
```

```
void LCD_clear(void)
{
        LCD_cmd(0X01);
        _delay_ms(1);
        LCD_cmd(0X8
        0);
        _delay_ms(10);
}

void LCD_cmd(unsigned char cmd)
{
        PORTA=cmd;
        ctrl=(0<<rs)|(0<<rw)|(1<<en);
        _delay_ms(1);
        ctrl=(0<<rs)|(0<<rw)|(0<<en);
        _delay_ms(2);
        return;
}
void LCD_data(unsigned char data)
{
        PORTA=data;
        ctrl=(1<<rs)|(0<<rw)|(1<<en);
        _delay_ms(1);
        ctrl=(1<<rs)|(0<<rw)|(0<<en);
        _delay_ms(2);
        return;
}
```

```
void LCD_string(unsigned char* str)
 {
        int i=0;
        while(str[i]!='\0')
        {
                LCD_data(str[i]);i++;
        }
        return;
 }
 void SPI_master_init(void)
 {
        DDRB =
        ((1<<PB5)|(1<<PB7)|(1<<PB4));
        PORTB |= !(1<<PB4);
        SPCR = ((1<<SPE)|(1<<MSTR)|(1<<SPR0));
 }

 void SPI_transmit(unsigned char
        data){SPDR=data;
        while(!(SPSR & (1<<SPIF)));
          return;
 }
```

References

Most part of the project was done by referring various websites and tutorials. For literature review and further understanding, some articles and journals werereferred. Some of them are:

[1] P. Dehghanian, M. Kezunovic, G. Gurrala and Y. Guan, "Security-based circuit breaker maintenance management," 2013 IEEE Power & Energy SocietyGeneral Meeting, 2013, pp. 1-5, doi: 10.1109/PESMG.2013.6672293..

[2] Password Based Circuit Breaker Control to Ensure Electric Line Man'sSafety And Load Sharing by C.Pearline Kamalini, A.Kokila, S.Jesimabanu, V.Jayalakshmi