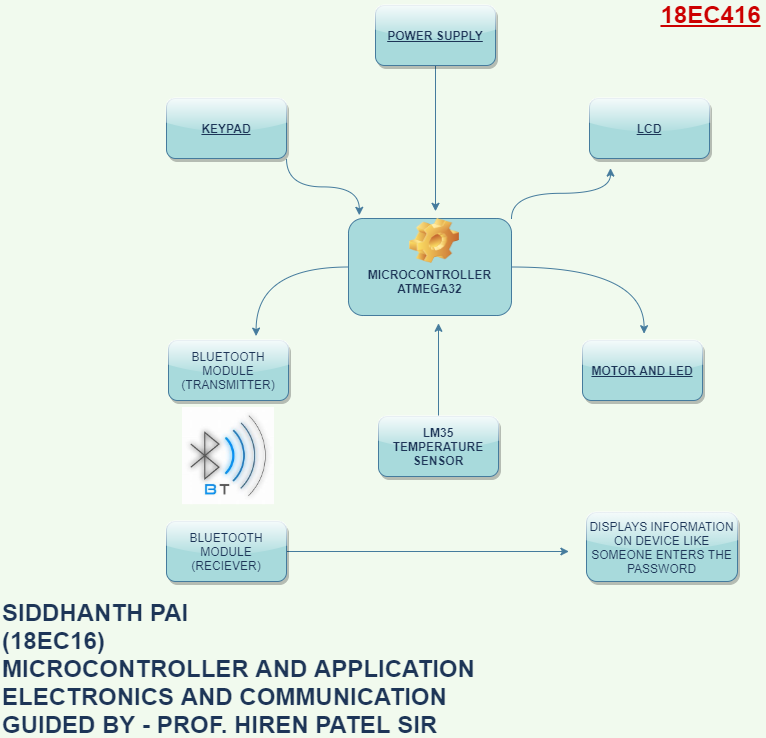
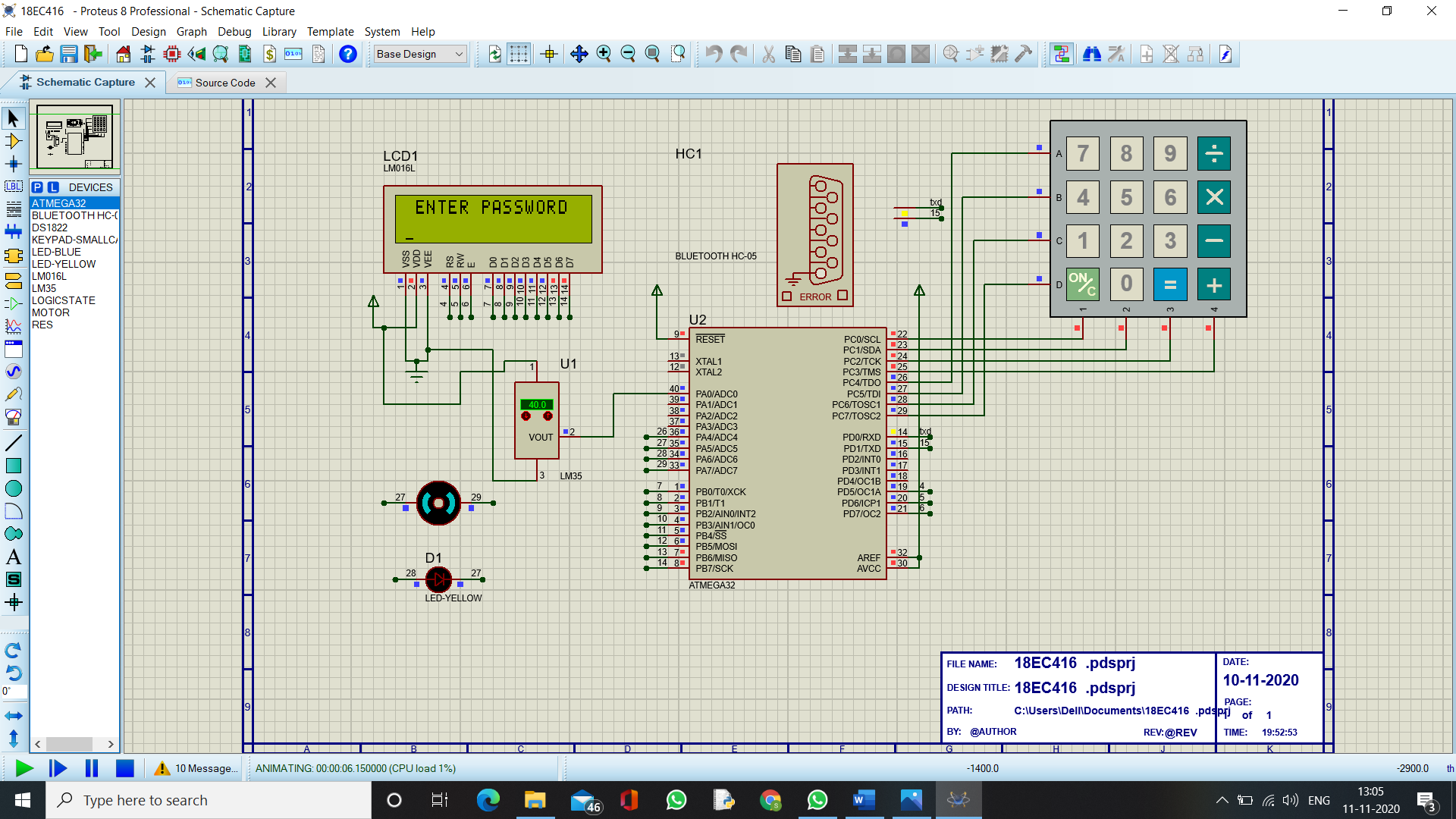
|  |  |
| --- | --- |
|  |  |

**PASSWORD BASED DOOR LOCK SECURITY SYSTEM**

**BLOCK DIAGRAM:**



**CIRCUIT DIAGRAM:**



**CODE:**

#include<avr/io.h> ////////////////////////////////////// **CORRECT PASSWORD = 5678** ///////////////////////////////////////////////

#include<util/delay.h>

///////////////////////////////////////// **LCD PART** ///////////////////////////////////////////////////////////////////////

#define LCD\_BPRT PORTB //FOR DATA PORT A

#define LCD\_BDDR DDRB

#define LCD\_BPIN PINB

#define LCD\_CPRT PORTD //FOR COMANDS PORT B

#define LCD\_CDDR DDRD

#define LCD\_CPIN PIND

#define LCD\_RS 5 /\* Define Register Select (data/command reg.)pin \*/

#define LCD\_RW 6 /\* Define Read/Write signal pin \*/

#define LCD\_EN 7 /\* Define Enable signal pin \*/

#define KEY\_PRT PORTC //initializing keyboard PORT

#define KEY\_DDR DDRC //initializing keyboard DDR

#define KEY\_PIN PINC //initializing keyboard PIN

void lcd\_cmd(unsigned char cmnd)

{

LCD\_BPRT= cmnd; // PORTB = cmnd

LCD\_CPRT &= ~(1<<LCD\_RS); /\* RS=0 command reg. \*/ // RS=0 for command

LCD\_CPRT &= ~(1<<LCD\_RW); /\* RW=0 Write operation \*/

LCD\_CPRT |= (1<<LCD\_EN); /\* Enable pulse high\*/

\_delay\_us(1);

LCD\_CPRT &= ~(1<<LCD\_EN); // enable low pulse

\_delay\_us(100);

}

void lcd\_data(unsigned char cmnd)

{

LCD\_BPRT= cmnd; // PORTB = cmnd

LCD\_CPRT |= (1<<LCD\_RS); /\* RS=1 for data. \*/

LCD\_CPRT &= ~(1<<LCD\_RW); /\* RW=0 Write operation \*/

LCD\_CPRT |= (1<<LCD\_EN); /\* Enable pulse \*/

\_delay\_us(1);

LCD\_CPRT &= ~(1<<LCD\_EN);

\_delay\_us(100);

}

void ini\_lcd(void) /\* LCD Initialize function \*/

{

LCD\_CDDR = 0xFF; /\* Make LCD command port direction as o/p \*/

LCD\_BDDR = 0xFF; /\* Make LCD data port direction as o/p \*/

LCD\_CPRT &=~(1<<LCD\_EN); // enable low

\_delay\_us(2000); /\* LCD Power ON delay always >15ms \*/

lcd\_cmd(0x38); /\* Initialization of 16X2 LCD in 8bit mode \*/ //00111000

lcd\_cmd(0x0E); /\* Display ON Cursor OFF \*/ //00001110

lcd\_cmd(0x06); /\* Auto Increment cursor \*/ //00000110

lcd\_cmd(0x01); /\* Clear display \*/ //00000001

lcd\_cmd(0x80); /\* Cursor at home position \*/

}

void lcd\_print(char \*str)

{

unsigned char i = 0;

while(str[i] != 0)

{

lcd\_data(str[i]);

i++;

}

}

//////////////////////////////////////////////// **KEYPAD PART** ///////////////////////////////////////////////////////////////////////////

#define key\_port PORTC

#define key\_ddr DDRC

#define key\_pin PINC

unsigned char keypad[4][4]= {{'7','8','9','/'},

{'4','5','6','\*'},

{'1','2','3','-'},

{'c','0','=','+'}};

char takekey()

{

unsigned char row,colum;

char key;

key\_ddr=0xf0; //11110000

key\_port=0xff; //11111111

do

{

key\_port&=0x0f; //PORTA 00001111

colum=(key\_pin&0x0f); //PIN 00001111

}while(colum!=0x0f); //COLUN 00001111

do

{

do

{

\_delay\_ms(1);

key\_port&=0x0f;

colum=(key\_pin&0x0f);

}while(colum==0x0f);

\_delay\_ms(1);

key\_port&=0x0f;

colum=(key\_pin&0x0f);

}while(colum==0x0f);

while(1)

{

key\_port=0xef;

colum=(key\_pin&0x0f);

if(colum!=0x0f)

{

row=0;

break;

}

key\_port=0xdf;

colum=(key\_pin&0x0f);

if(colum!=0x0f)

{

row=1;

break;

}

key\_port=0xbf;

colum=(key\_pin&0x0f);

if(colum!=0x0f)

{

row=2;

break;

}

key\_port=0x7f;

colum=(key\_pin&0x0f);

row=3;

break;

}

if(colum==0x0e)

key=keypad[row][0];

else if(colum==0x0d)

key=keypad[row][1];

else if(colum==0x0b)

key=keypad[row][2];

else

key=keypad[row][3];

return(key);

}

//////////////////////////////////// **HC05 PART** //////////////////////////////////////////////////////////////////////////////////////

void USART\_Init(unsigned long BAUDRATE) /\* USART initialize function \*/

{

UCSRB |= (1<<RXEN)|(1<<TXEN)|(1<<RXCIE);

UCSRB &= ~(1<<UCSZ1);

UCSRC |= (1<<UCSZ0) | (1<<UCSZ1);

UCSRA |= (1<<U2X);

UBRRL = 12;

UBRRH = 0; /\* Load UBRRH with upper 8 bit of prescale value \*/

}

char USART\_RxChar() /\* Data receiving function \*/

{

while (!(UCSRA & (1 << RXC))); /\* Wait until new data receive \*/

return(UDR); /\* Get and return received data \*/

}

void USART\_TxChar(char data) /\* Data transmitting function \*/

{

UDR = data; /\* Write data to be transmitting in UDR \*/

while (!(UCSRA & (1<<UDRE))); /\* Wait until data transmit and buffer get empty \*/

}

void USART\_SendString(char \*str) /\* Send string of USART data function \*/

{

int i=0;

while (str[i]!=0)

{

USART\_TxChar(str[i]); /\* Send each char of string till the NULL \*/

i++;

}

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void adc\_init()

{

ADMUX = (1<<REFS0);

ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

}

uint16\_t adc\_read(uint8\_t ch)

{

ch &= 0b00000111; // AND operation with 7

ADMUX = (ADMUX & 0xF8)|ch;

ADCSRA |= (1<<ADSC);

while(ADCSRA & (1<<ADSC));

return (ADC);

}

/////////////////////////////////////// **INT MAIN** ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int main()

{

uint16\_t adc\_result0;

int temp;

adc\_init();

char buffer[10];

unsigned int pass=5678,key;

unsigned char ch=0,i,w[]=" WRONG PASSWORD",E[]=" ENTER PASSWORD",s[]="CORRECT PASSWORD",a[]=" WELCOME :) ";

DDRA=0XFF;

DDRB=0XFF;

\_delay\_ms(500);

ini\_lcd();

lcd\_cmd(0x80);

for(i=0;i<15;i++)

lcd\_data(a[i]);

\_delay\_ms(1000);

lcd\_cmd(0x01);

while (1)

{

key=0; // for storing full password

ch=0; // for storing individial value of key press

lcd\_cmd(0X80); //force cursor to beginning of 1st line

for(i=0;i<15;i++)

lcd\_data(E[i]); // displays ENTER PASSWORD on first line

lcd\_cmd(0XC0); //force cursor to beinning of 2nfdline

while(ch!='=') //get password from visitor i.e takes value , = is for going to next value

{

ch=takekey(); //takes integer one by one from user

if(ch!='=') // pressed key should not be = or else it will go to next frame for saying the password is correct or not

{

lcd\_data(ch);

key=((key\*10)+(ch-0x30)); //key like 5678 converting it into BCD

}

}

lcd\_cmd(0X01); //clear display screen for going to next frame for saying if password is correct or not

if(key!=pass) //wrong password

{

for(i=0;i<15;i++)

lcd\_data(w[i]);

PORTA=0b11110000;

USART\_Init(9600); /\* initialize USART with 9600 baud rate \*/

UBRRL = 12;

\_delay\_ms(500);

USART\_SendString(" someone enters wrong password ");

adc\_result0 = adc\_read(0);

temp=adc\_result0/2.01;

itoa(temp,buffer,10);

lcd\_cmd(0XC0);

lcd\_print("Temp=");

lcd\_cmd(0xC7);

lcd\_print(buffer);

lcd\_cmd(0xC9);

lcd\_print("C");

\_delay\_ms(900);

lcd\_cmd(0x01);

//\_delay\_ms(5000);

ch=takekey(); // if not then it will come back to enter password terminal automatically

}

else //correct password

{

PORTA=0b11001100;

USART\_Init(9600); /\* initialize USART with 9600 baud rate \*/

UBRRL = 12;

\_delay\_ms(500);

USART\_SendString(" someone has been entered");

for(i=0;i<15;i++)

{

lcd\_data(s[i]);

}

adc\_result0 = adc\_read(0);

temp=adc\_result0/2.01;

itoa(temp,buffer,10);

lcd\_cmd(0XC0);

lcd\_print("Temp=");

lcd\_cmd(0xC7);

lcd\_print(buffer);

lcd\_cmd(0xC9);

lcd\_print("C");

ch=takekey(); // if not then it will come back to enter password terminal automatically

PORTA=0; // to stop motor and led

lcd\_cmd(0x01);

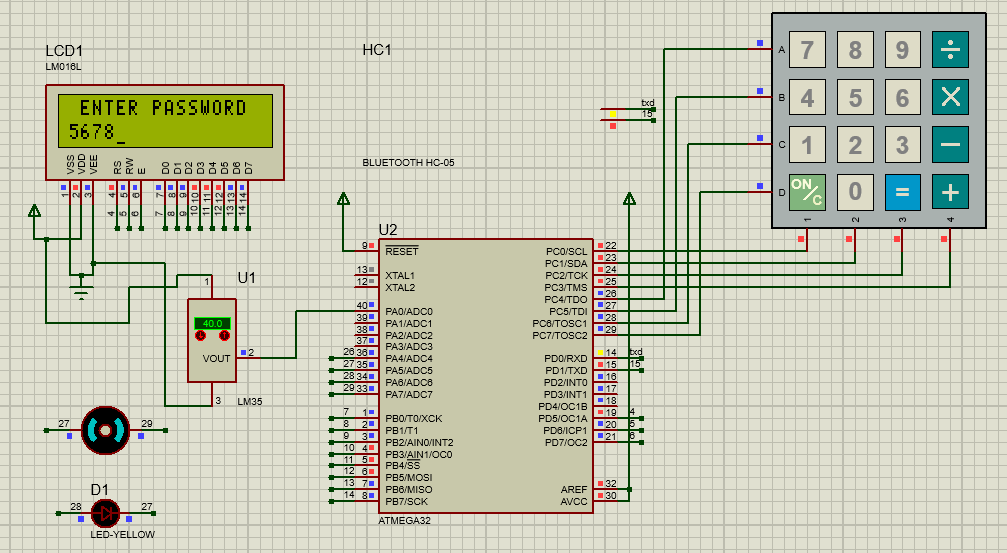
}

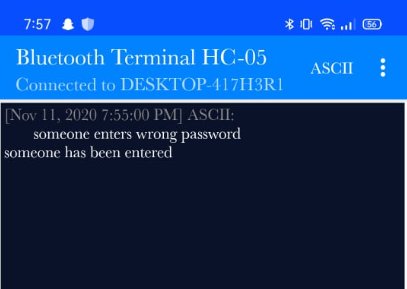
}

return(0);

}

**SOME OF SCREEN SHOTS:**





**Video link:**

