VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi - 590 018, Karnataka, India.

MICROCONTROLLERS AND EMBEDDED SYSTEMS MINI PROJECT REPORT

On

"PASSWORD BASED DOOR LOCK SYSTEM USING LPC2148"

Submitted in partial fulfilment of the requirements for the award of the Degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

Submitted by

1.	**SUGAM CHAND M**	**1BI19CS155**
2.	**SUMAN TD**	**1BI19CS159**
3.	**VEERA SATWIK**	**1BI19CS183**
4.	**VISHAL SACHAN**	**1BI19CS188**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
BANGALORE INSTITUTE OF TECHNOLOGY

SIGN OF THE FACULTY IN-CHARGE

(Dr. HARISH KUMAR B T)

Index

Contents

Introduction

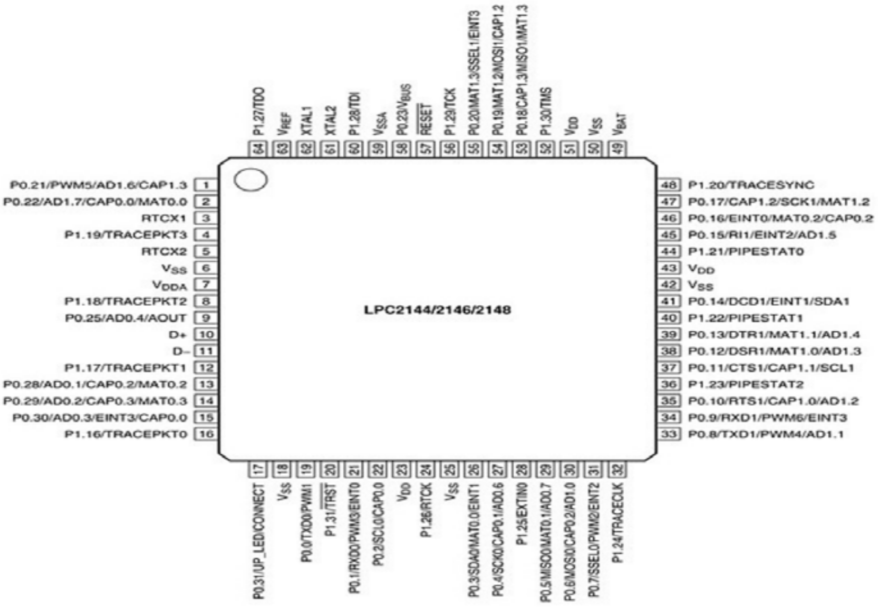
Password Based Door Open System Using LPC2148:

In the present scenario of the world, security is a major concern for all, and the security problem is being faced by every person. The usual means of securing anything is through mechanical locks, which operate with a specific key or a few keys, but, for locking a large area many locks are necessary. However, conventional locks are heavy and do not offer the desired protection as they can be easily broken down by using some tools. Therefore, security breaching problems are associated with the mechanical locks. However to decide the electronic based locking system problems that are associated with the mechanical locks.

Nowadays, many devices' operations are based on digital technology. For example, token based digital identity device. Here we will see Password Based Door Open System Using LPC2148.

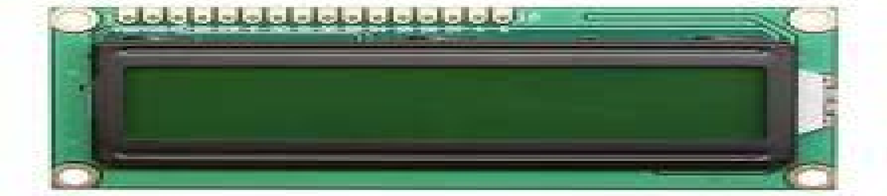
LPC2148 Micro controller:

The LPC2148 microcontroller is designed by Philips (NXP Semiconductor) with several in-built features and peripherals. Due to these reasons, it will make more reliable as well as the efficient option for an application developer. LPC2148 is a 16-bit or 32-bit micro controller based on ARM7 family.



LCD Module:

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers.



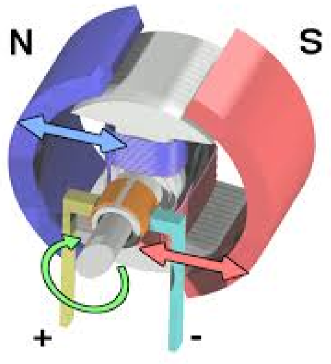
A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling the display, etc. The data register stores the data to be displayed on the LCD.

4 X 3 Keypad:

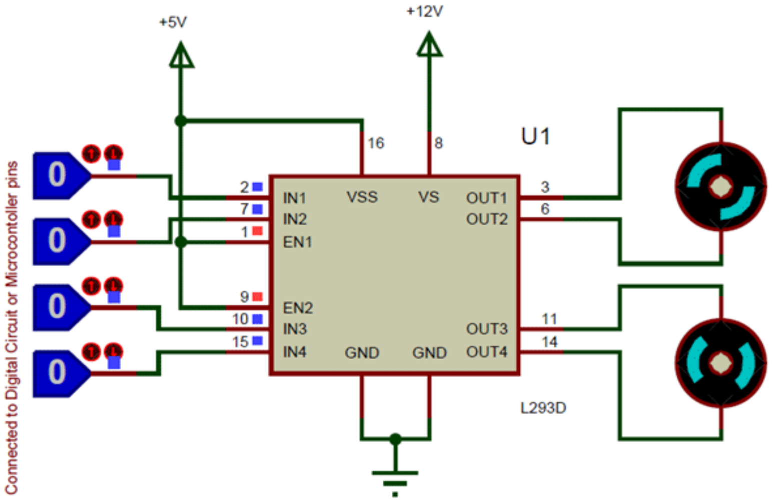
This Matrix Keypad 4 X 3 has **12 buttons**, arranged in a telephone-line 3x4 grid. The keys are connected into a matrix, hence only 7 microcontroller pins (3-columns and 4-rows) are required to scan the pad.

DC Motor:

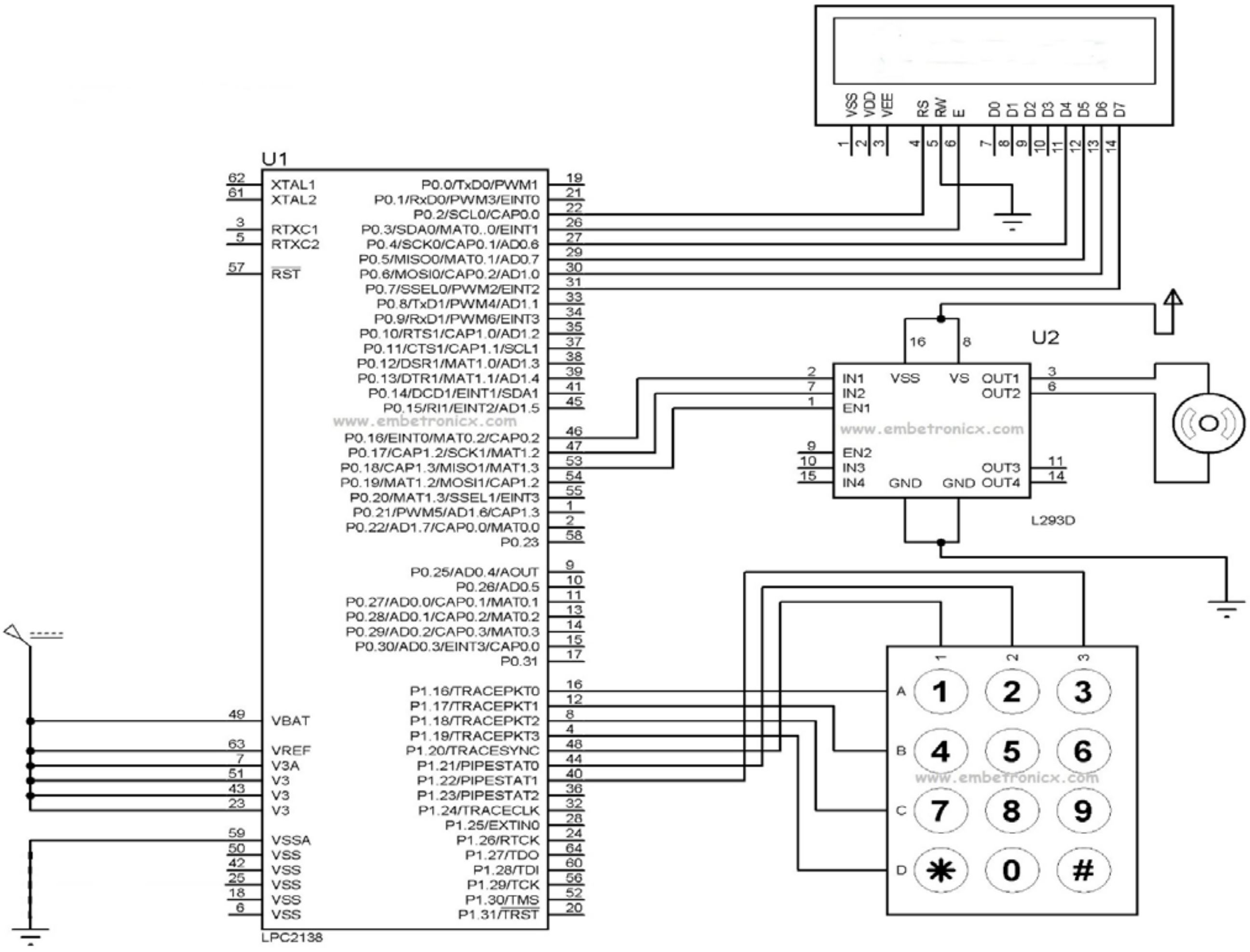
A **DC motor** is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.



L293D Motor Driver:



The L293D is a popular 16-Pin **Motor Driver IC**. As the name suggests it is mainly used to drive motors. A single **L293D IC** is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Microcontrollers like Arduino, PIC, ARM etc.



Problem Statement

This system demonstrates a Password based Door Lock System using LPC2148 Micro controller, wherein:

- once the correct code or password is entered, the door is opened and the concerned person is allowed access to the secured area.
- Again, if another person arrives, it will ask to enter the password.
- If the password is wrong, then door would remain closed, denying access to the person.

Module wise explanation along with logic

#include is a way of including a standard or user-defined file in the program and is mostly written at the beginning of any C program. This directive is read by the pre-processor and orders it to insert the content of a user-defined or system header file into the following program. These files are mainly imported from an outside source into the current program.

LPC214X.H is a built-in header file for all the Phillips LPC214 series microcontrollers that contains the basic definitions of important Pins that exist in LPC2148 and other members of the series.

The program contains custom functions which can be classified based upon which component they control. Below are the categories with the functions along with their descriptions:

- **DC Motor**:
  1. **void forward(void)** : Controls the DC Motor by setting the pins connected to it such that we observe a forward rotation. It returns nothing but the state of the motor is set to running forward.
  2. **void reverse(void)** : Controls the DC Motor by setting the pins connected to it such that we observe a reverse rotation. It returns nothing but the state of the motor is set to running in reverse.
  3. **void stop(void)** : Stops the DC Motor by clearing all pins that run the motor. It returns nothing but the state of motor is set to stop.
- **LCD:**
  1. **void lcd_init(void)** : Initialises the LCD using another method cmd() which readies the LCD to display the designated content by the manipulation of existing Pins corresponding to LCD.
  2. **void cmd(unsigned char a)** : A sub process which handles the initialisation of the LCD display by the manipulation of certain IO pins.
  3. **void dat(unsigned char b)** : Displays a character on the LCD by transmitting the information to the LCD
  4. **void show(unsigned char *s)** : Using dat(), this function helps to print a string on the LCD display.
  5. **void lcd_delay()** : Provides a small delay in between transmissions to ensure that the message transmissions do not collide during any of the above data transmission methods.

- **Keypad:**
    1. c**x** : These are some important macros that inform about the current state of the IOpin1. **'x'** can take values of 1, 2 and 3 each corresponding to the port pins P1.20, P1.21 and P1.22 respectively
    2. **unsigned char r_loc, c_loc** : These global variables store the value of the row and column of the keypad button respectively that has been pressed.
    3. **unsigned char keypad(void)** : Senses the key pressed by checking the states of the pins P1.16-P1.22. Each column's state is checked and then the column number is noted in c_loc. Each row is then checked and then row number is noted in r_loc. The keypad button that is pressed is linked to a unique key (For example, the button with r_loc=0 and c_loc=0 corresponds to '1'). The value of the key pressed is returned.
- **Miscellaneous:**
    1. **void delay()** : Causes a short delay that is required in between certain operations.
    2. **bit(x)** : Left shifts 1 by x times, this corresponds to the bit x being 1 while the rest are 0.
- **Main:** The main body of the program that runs indefinitely and infinitely until stopped. The keypad input is taken and stored temporarily and also changes the state of LCD. If the password entered is correct, uses suitable DC Motor functions to run the motor in order to open/close the lock. In case of a wrong password, the system resets and gets ready to take another password.

Implementation

Source Code:

```
#include<lpc214x.h>
#define bit(x) (1<<x)

void delay ()
{
    unsigned int temp, ct;
    for(ct=0; ct<30; ct++) {
        for(temp=0; temp < 65000; temp++);
    }
}
unsigned int range=0,i;

/* --------- DC Motor ----- */
void forward(void);
void reverse(void);
void stop(void);
/* --------- LCD ---------*/
void lcd_init(void);
void cmd(unsigned char a);
void dat(unsigned char b);
void show(unsigned char *s);
void lcd_delay(void);

/* ---------- Keypad ------*/
#define c1 (IOPIN1&1<<20)
```

```
#define c2 (IOPIN1&1<<21)
#define c3 (IOPIN1&1<<22)

unsigned char r_loc,c_loc;
unsigned char keypad(void);

/* ------- Main -------*/
int main()
{
                    char a, b, c, d;
   VPBDIV=0x01;             // PCLK = 60MHz
   IO1DIR |= 0x0f<<16;
   IO0DIR |= 0xf00fc;

   lcd_init();
   while(1) {
      cmd(0x80);
      show("#Enter Password#");

      cmd(0xc5);

                                        a = keypad();
                                        dat('*');
                                        b = keypad();
                                        dat('*');
                                        c = keypad();
                                        dat('*');
                                        d = keypad();
                                        dat('*');
```

```
        if(('5' == a) && ('5' == b) && ('5' == c) && ('5' == d) ) {
            cmd(0xc0);
            show("   Thank You!   ");
            forward();
            delay();
            stop();
            cmd(0xc0);
            show("  Come Again!!  ");
            delay();
            reverse();
            delay();
            stop();
        } else {
            cmd(0xc0);
            show("~Wrong Password~");
            delay();
        }
        cmd(0x01);
    }
}

/* ----- Keypad Function ----*/
unsigned char keypad()
{
    IO1PIN &= ~(0xff<<16);
    IO1PIN |= 0xf0<<16;
    while(c1 && c2 && c3);
    while(!c1 || !c2 || !c3) {
```

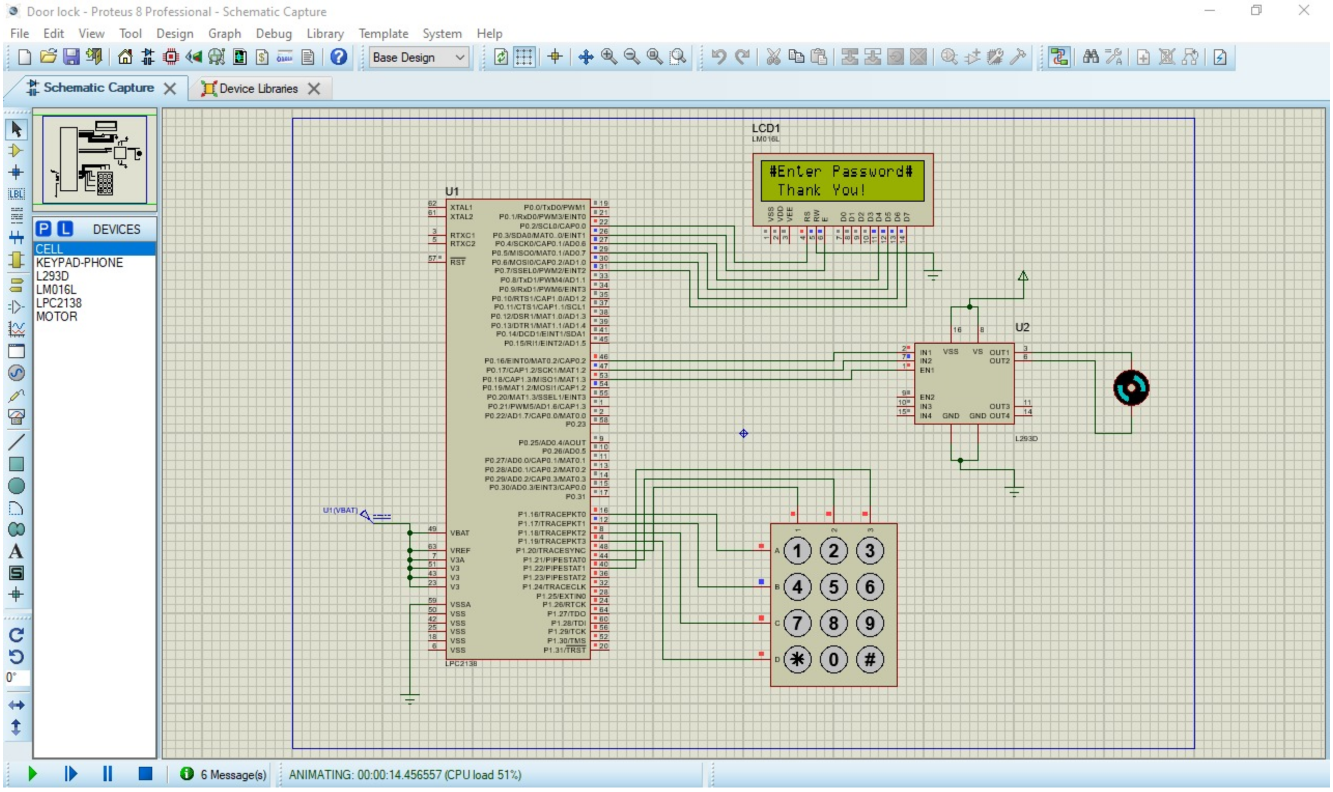
```
if(!c1 && c2 && c3)     c_loc=0;

else if(c1 && !c2 && c3)    c_loc=1;

else if(c1 && c2 && !c3)    c_loc=2;


IO1CLR = 1<<16;

IO1SET = 0x0e<<16;

if(!c1 || !c2 || !c3) {

   r_loc=0;

   break;

}


IO1CLR = 1<<17;

IO1SET = 0x0d<<16;

if(!c1 || !c2 || !c3) {

   r_loc=1;

   break;

}


IO1CLR = 1<<18;

IO1SET = 0x0b<<16;

if(!c1 || !c2 || !c3) {

   r_loc=2;

   break;

}


IO1CLR = 1<<19;

IO1SET = 0x07<<16;

if(!c1 || !c2 || !c3) {

   r_loc=3;
```

```
        break;
    }
}
while(!c1 || !c2 || !c3);
switch(c_loc){
                case 0: switch(r_loc){
                                        case 0: return '1';
                                        case 1: return '4';
                                        case 2: return '7';
                                        case 4: return '*';
                                }
                                break;
                case 1: switch(r_loc){
                                        case 0: return '2';
                                        case 1: return '5';
                                        case 2: return '8';
                                        case 4: return '0';

                                }
                                break;
                case 2: switch(r_loc){
                                        case 0: return '3';
                                        case 1: return '6';
                                        case 2: return '9';
                                        case 4: return '#';

                                }
                                break;
        }
        return '\0';
```

```
}

/* ----- LCD Functions -------*/

void lcd_init()
{
    cmd(0x02);
    cmd(0x28);
    cmd(0x0c);
    cmd(0x06);
    cmd(0x80);
}

void cmd(unsigned char a)
{
    IO0PIN &= 0xffffff03;
    IO0PIN |= (a & 0xf0) << 0;
    IO0CLR |= bit(2);               //rs=0
    IO0CLR |= bit(1);               //rw=0
    IO0SET |= bit(3);                //en=1
    lcd_delay();
    IO0CLR |= bit(3);                //en=0

    IO0PIN &= 0xffffff03;
    IO0PIN |= ((a << 4)  & 0xf0) << 0;
    IO0CLR |= bit(2);               //rs=0
    IO0CLR |= bit(1);               //rw=0
    IO0SET |= bit(3);                //en=1
    lcd_delay();
    IO0CLR |= bit(3);                //en=0
}

void dat(unsigned char b)
{
    IO0PIN &= 0xffffff03;
    IO0PIN |= (b & 0xf0) << 0;
    IO0SET |= bit(2);              //rs=1
    IO0CLR |= bit(1);              //rw=0
    IO0SET |= bit(3);          //en=1
    lcd_delay();
    IO0CLR |= bit(3);          //en=0

    IO0PIN &= 0xffffff03;
    IO0PIN |= ((b << 4)  & 0xf0) << 0;
    IO0SET |= bit(2);              //rs=1
    IO0CLR |= bit(1);              //rw=0
    IO0SET |= bit(3);          //en=1
    lcd_delay();
    IO0CLR |= bit(3);          //en=0
}

void show(unsigned char *s)
{
    while(*s) {
        dat(*s++);
    }
}
```

```
void lcd_delay()
   {
   unsigned int i;
   for(i=0;i<=1000;i++);
}

/* ---------- DC Motor ---------*/
void forward()
{

   IO0SET = bit(16) | bit(18);
   IO0CLR = bit(17);
}

void reverse()
{

   IO0SET = bit(17) | bit(18);
   IO0CLR = bit(16);
}

void stop()
{
   IO0CLR = bit(18);
}
```

Snapshots

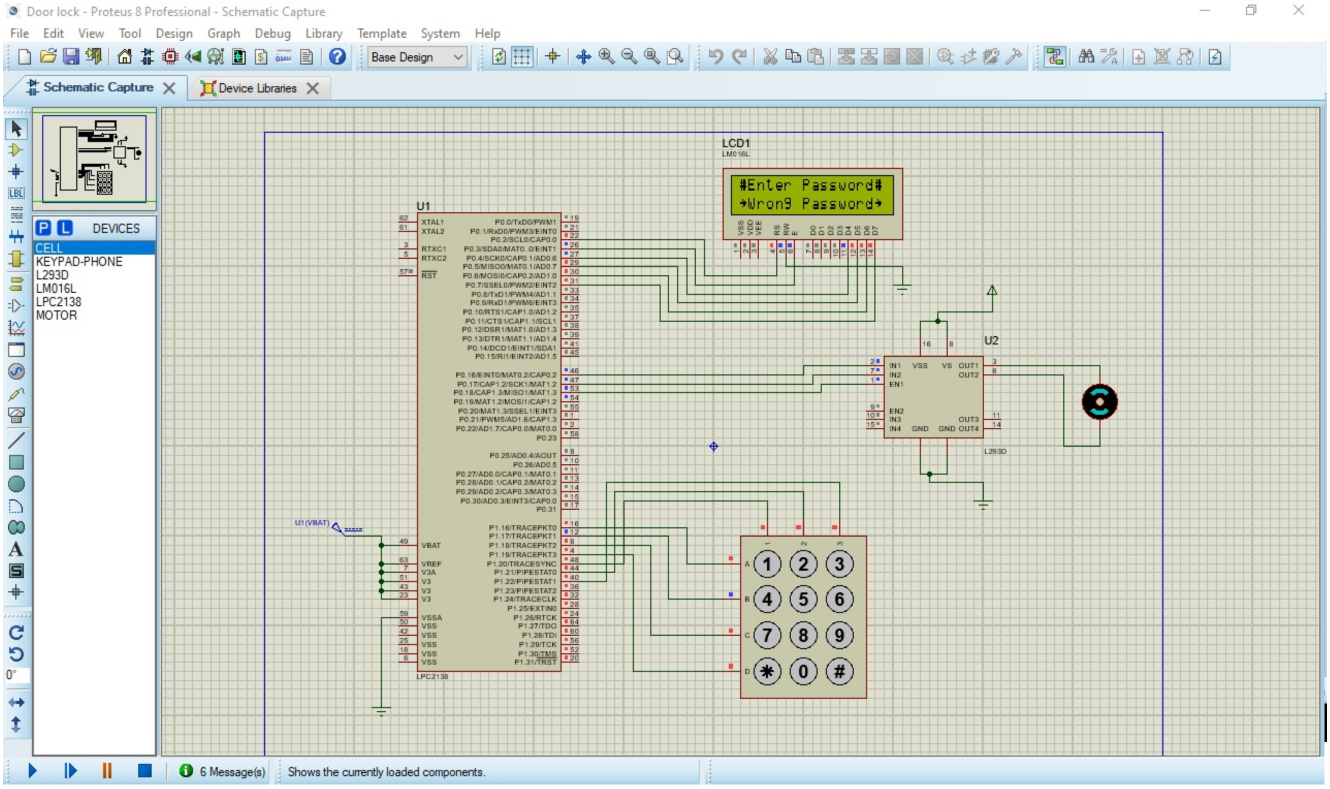


Door lock - Proteus 8 Professional - Schematic Capture

```
#Enter Password#
 Thank You!
```

6 Message(s)

ANIMATING: 00:00:14.456557 (CPU load 51%)



Door lock - Proteus 8 Professional - Schematic Capture

```
#Enter Password#
→Wrong Password→
```

6 Message(s)

Shows the currently loaded components.

Applications

- This simple circuit can be used at residential places to ensure better safety.
- It can be used at organizations to ensure authorized access to highly secured places.
- With a slight modification this Project can be used to control the switching of loads through password.
- Can be used in various rooms like seminar hall, conference room, and study rooms in college.
- The project (password detector) can be used to automate the door locking process, so the user need not to carry the door lock keys along with them, they need to just remember the password and use it later to open the door.
- It will add the good security.
- Gives an indication for unauthorized entry.
- User don’t have to carry keys along with him.

CONCLUSION:

This automated gate locking system performed as expected. We are able to implement all of the functions specified in this proposal. The biggest hurdle we had to overcome with this project was interfacing the Microcontroller with the hardware components. This product is very marketable because it is easy to use, comparatively inexpensive due to low power consumption, and highly reliable.

Future Development:

- We can monitor parameters like fire, overheat.
- We can provide voice feedback system.
- We can interface GSM modem which will send sms if invalid attempt is made to open the lock.