**DIGITAL DOOR LOCK WITH 4 DIGIT CODEUSING 8051 MICROCONTROLLERS**

by

ANUNOY MAJI 18BEC1168

KSHITIJ SHARMA 18BEC1199

SHREYANSH JAMUAR 18BEC1224

PRAJWAL BONDARDE 18BEC1279

A project report submitted to

**Prof. ANITA** **ANGELINE**

**SCHOOL OF ELECTRONICS ENGINEERING**

in partial fulfilment of the requirements for the course of

**ECE3003– MICROCONTROLLER AND ITS APPLICATIONS**

in

**B.Tech. ELECTRONICS AND COMPUTER ENGINEERING**

****

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**JUNE 2020**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**DIGITAL DOOR LOCK WITH 4 DIGIT CODE USING 8051 MICROCONTROLLERS”**is a bonafide work of **ANUNOY MAJI– 18BEC1168, KSHITIJ SHARMA - 18BEC1199,SHREYANSH JAMUAR – 18BEC1224 andPRAJWAL BONDARDE - 18BEC1279”**who carried out the Project work under my supervision and guidance for**ECE3003-Microcontroller and its Applications.**

**Prof. ANITA ANGELINE**

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

**ABSTRACT**

This Digital Door Lock – is simply a password based electronic code lock designed using 8051 micro controller, a keypad and a 12 volt dc relay. In this project, we have designed a simple digital door lock using 8051 -which can be used as a security checking system to limit access to an area/room only for certain individuals with the password. So our digital door lock project can be called with a very wide range of names like a digital combination lock using 8051 or a digital security code lock using 8051 microcontroller or a password security system using 8051 or an electronic code lock or a digital code lock using 8051. Extra features like automatic door lock/opening facility, sound alarm, gsm based sms alert can also be appended in this project.

Our Digital Code Lock project – is a simple electronic number lock system or an electronic combination lock using 8051 – which has a preset 5 digit password stored inside the program. The system collects 5 digit user input, compares the user input with the preset password inside program and if the user input and stored password matches, access will be granted (by opening the door with the help of relay for a few seconds and closing it automatically after stipulated time) . If there is a mismatch between user input and stored password, access will be denied (by not opening the closed door – that is by keeping the relay in OFF position)

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide,**Prof Anita Angeline,** Associate Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We express our thanks to our Head of the Department**Dr. Vetrivelan. P**forhis support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**ANUNOY MAJI KSHITIJ SHARMA**

**SHREYANSH JAMUAR PRAJWAL BONDARDE**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SL. NO.** |  | **TITLE** | **PAGE NO.** |
|  |  | ABSTRACT | 3 |
|  |  | ACKNOWLEDGEMENT | 4 |
|  |  |  |  |
| 1 |  | INTRODUCTION | 6 |
|  |  |  |  |
|  | 1.1 | OBJECTIVES | 6 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 2 |  | DIAGRAMS AND SIMULATION |  |
|  | 2.1  2.2 | BLOCK DIAGRAM OF SYSTEM DESIGN  CIRCUIT DIAGRAM | 7  7 |
|  | 2.3  2.4  2.5  2.5  2.6 | COMPONENTS REQUIRED  SIMULATION  HARDWARE DESCRIPTION  SOFTWARE AND CODING  WORKING PRINCIPLE | 8  8-10  11-13  14-25  25-26 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 3 | 3.1  3.2  3.3  3.4 | **ADVANTAGES OF DOOR LOCK SYSTEM APPLICATION OF DOOR LOCK SYSTEM LIMITATION OF DOOR LOCK SYSTEM INTERFERENCE** | 27  27  27  28 |

**CHAPTER 1**

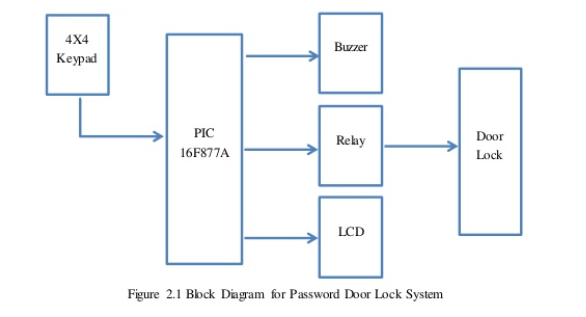
**INTRODUCTION**

* 1. **OBJECTIVES:**
* Initially, declare the PORT1 to LCD data pins and control pins (RS and E) to P3.0 and P3.2. Also, declare PORT2 to keypad. Also use P0.0 and P0.1 for motor driver.
* Then, display the message “enter password” on LCD.
* Now read the five-digit password from the user.
* Compare the entered password with the stored password.
* If password is correct, then make P0.0 pin HIGH and P0.1 pin LOW to open the door. During this time, display *“Door opening”* on LCD.
* After some time, make P0.0 pin LOW and P0.1 pin HIGH to close the door and after this display *“Door closing”* on LCD.
* If the password is wrong, then display *“Wrong Password”* on LCD.
* After some delay again ask to enter password.

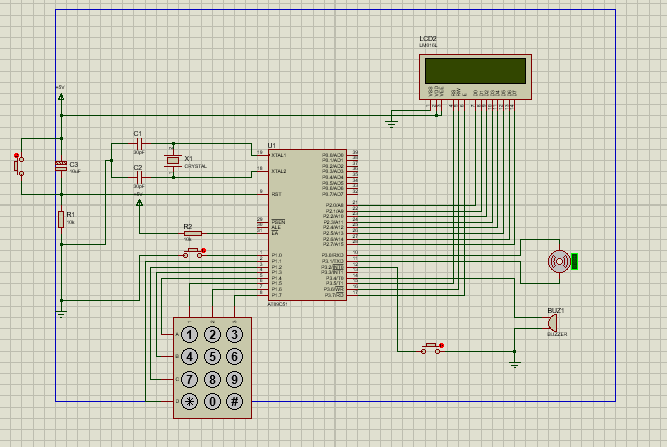
**CHAPTER 2**

**2.1 BLOCK DIAGRAM**

The Block diagram of the system is given below:



* 1. **CIRCUIT DIAGRAM**

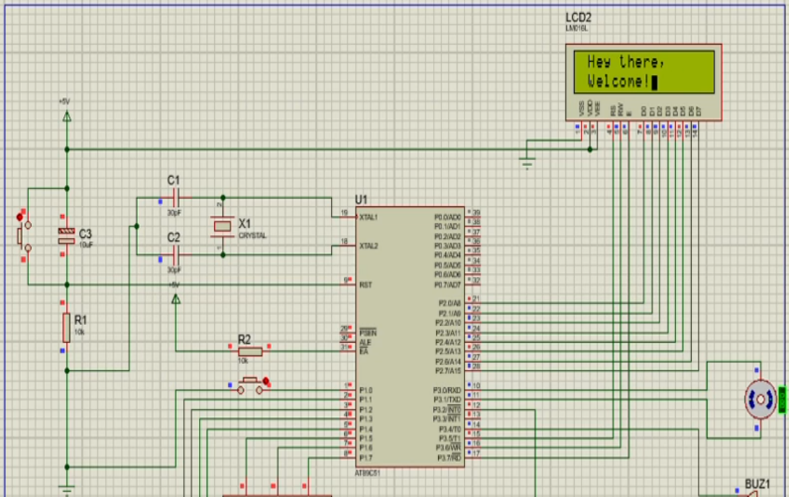


**2.3 COMPONENTS REQUIRED**

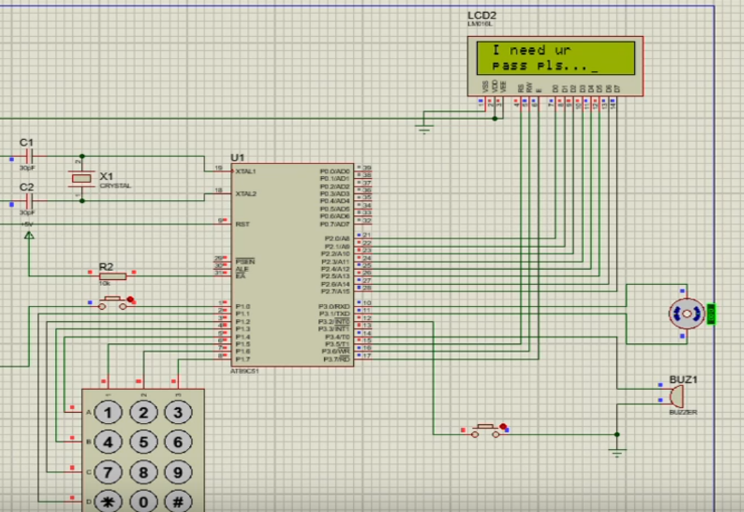
* LCD (LM016L)
* AT89C52 Microcontroller
* Switches
* LED
* Resistors (10K)
* Capacitor (10uF,30uF)
* Crystal oscillator
* Voltage regulator(7805)
* Buzzer
* Number-Pad
* Keil
* Proteus
* Connecting wires

**2.4 SIMULATION:**

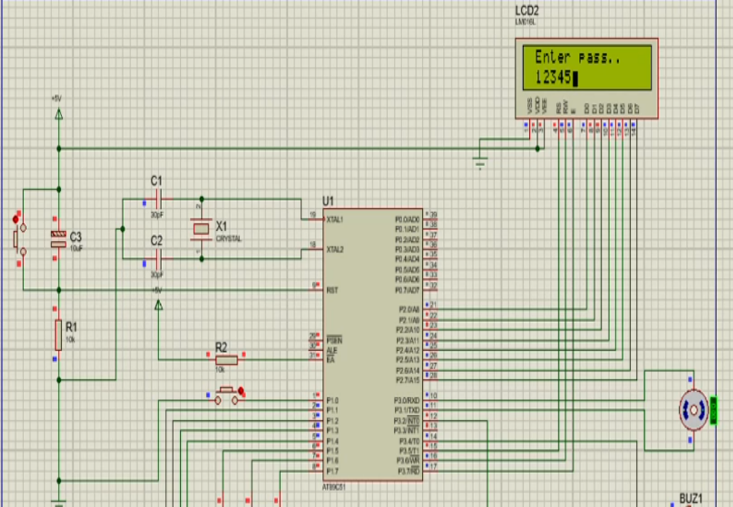
1. Initial state of the system when we start the machine.



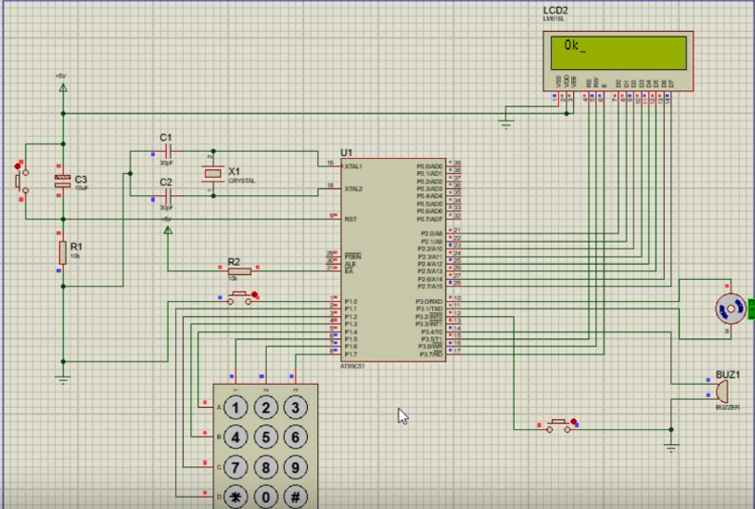
1. The LED will as the password form the user.



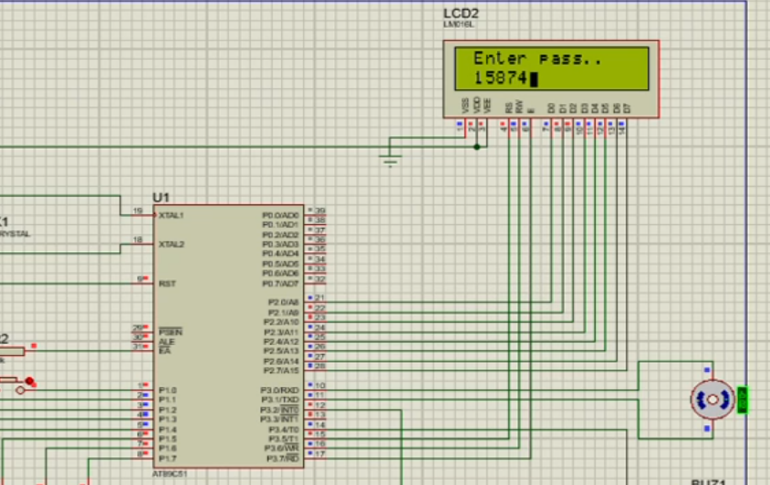
1. Enter the password LED will display the password which you display.



1. If the password is correct than it will display “Ok”. And if the password is correct than motor will turn and open the lock.



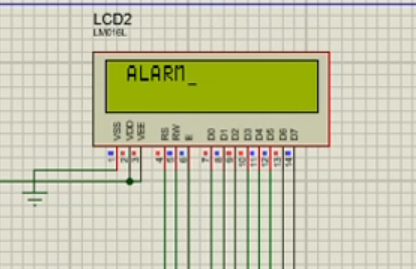
1. If you write wrong password continuously for 2 times than speaker will beep.



**6.)** If the password is wrong than LED will display “Pls try Again”.



**7.)**And after 2 wrong password alarm will start.



**2.5 HARDWARE DESRIPTION:**

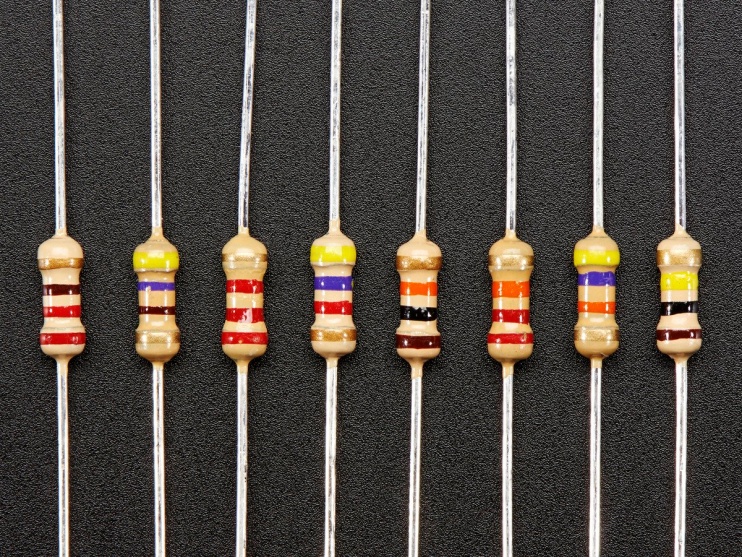
**1.) LED:**



**2.) AT89C52 MICROCONTROLLER:**



**3.) RESISTORS:**



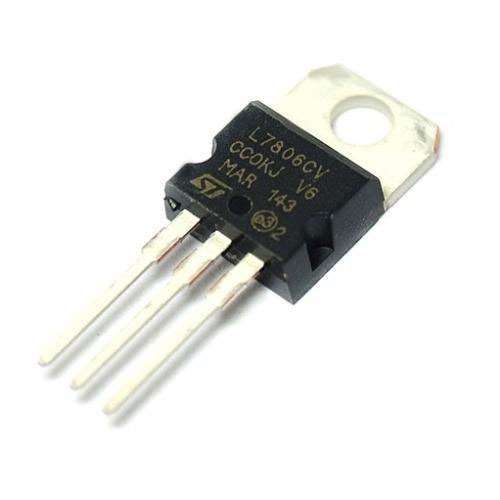
1. **CAPACITOR:**



**5.) CRYSTAL OSCILLATOR:**



**6.) VOLTAGE REGULATOR**



**7.) BUZZER:**



**8.) NUMBER-PAD:**



**2.6 SOFTWARE AND CODING**

***CODE:***

#include <reg51.h> /\* define 8051 registers \*/

void delay\_50ms(unsigned int);

void init\_Sys(void);

void set\_sett(void);

void set\_data(void);

void s\_latch(unsigned char );

void clr(void);

void prep\_Wrt(void);

void prep\_Wrt2(void);

unsigned char rowFinder1(void);

unsigned char rowFinder2(void);

unsigned char rowFinder3(void);

unsigned char rowFinder4(void);

void postKyData(unsigned char);

void class\_init(void);

void prepKeyb(void);

void prepLCD(unsigned char );

#define LCD P2

sbit RS=P3^5;

sbit RW=P3^6;

sbit EN=P3^7;

sbit R1=P1^4;

sbit R2=P1^3;

sbit R3=P1^2;

sbit R4=P1^1;

sbit C1=P1^5;

sbit C2=P1^6;

sbit C3=P1^7;

sbit user\_switch=P1^0;

sbit MOTOR\_1=P3^0;

sbit MOTOR\_2=P3^1;

sbit ALARM\_PIN=P3^4;

sbit EXT\_INT=P3^2;

unsigned char ALARM\_TRIG=0,

defPass[6]="123456",

asc\_Alarm[5]="ALARM",

asc\_Greet1[10]="Hey there,",

asc\_Greet2[8]="Welcome!",

asc\_EnterPass[12]="Enter pass..",

asc\_Misc[14]="Pass mismatch!",

asc\_Redo[15]="Pls try again..",

asc\_Success[2]="Ok",

asc\_Ndpass1[9]="I need ur",

asc\_Ndpass2[11]="pass pls...",

asc\_changePass[9]="pass rst?",

asc\_openDoor[2]="Op",

asc\_closDoor[2]="Sh";

void ex0\_isr (void) interrupt 0

{

EXT\_INT=1;

ALARM\_PIN=0;

}

void main(void)

{

init\_Sys();

class\_init();

while (1){

}}

void init\_Sys(void){

RW=0; //R/W=0 for write

RS=0; //RS=0 for command

EN=1; //E=1 for high pulse

delay\_50ms(1); //give LCD some time

set\_sett();

LCD=0x38; //;INIT. LCD 2 LINES, 5X7 MATRIX

s\_latch(4);

LCD=0x0F; //Display on, cursor blinking

s\_latch(1);

MOTOR\_1=0;

MOTOR\_2=0;

ALARM\_PIN=0;

IT0 = 1; // Configure interrupt 0 for falling edge on /INT0 (P3.2)

EX0 = 1; // Enable EX0 Interrupt

EA = 1; // Enable Global Interrupt Flag

}

void set\_sett(void){

RW=0; //R/W=0 for write

RS=0; //RS=0 for command

}

void set\_data(void){

RW=0;//R/W=0 for write

RS=1;

}

void s\_latch(unsigned char x){

unsigned char z;

for(z=0;z<x;z++){

EN=1; //E=1 for high pulse

delay\_50ms(3); //give LCD some time

EN=0; //E=0 for H-to-L pulse, i.e latch in

}

}

void delay\_50ms(unsigned int x){

unsigned char c;

for(c=0;c<x;c++){

TMOD&=0x0F; //;Timer 0, mode 1(16-bit mode

TMOD|=0x01; //;Timer 0, mode 1(16-bit mode

TL0=0xFD; //the low byte

TH0=0x4B; //the high byt

TR0=1; //start the timer 0

while (TF0==0);//monitor timer flag 0 until it rolls over

TR0=0; //stop timer 0

TF0=0; //clear timer 0 flag

}

}

void prep\_Wrt(void){

set\_sett();

LCD=0x80; //LCD Command that Forces cursor to beginning to 1st line

s\_latch(1);

LCD=0x0;

set\_data();

s\_latch(1);

}

void prep\_Wrt2(void){

set\_sett();

LCD=0xC0; //LCD Command Force cursor to beginning to 2nd line

s\_latch(1);

LCD=0x0;

set\_data();

s\_latch(1);

}

void clr(void){

delay\_50ms(30); //give LCD some time

set\_sett();

LCD=0x1; //Clear display screen

s\_latch(1);

}

void userInit(void){

unsigned char c;

ALARM\_TRIG++;

clr();

prep\_Wrt();

if(ALARM\_TRIG>3){

ALARM\_PIN=1;

for(c=0;c<5;c++){

LCD=asc\_Alarm[c];

s\_latch(1);

}

while(1);

}

for(c=0;c<12;c++){

LCD=asc\_EnterPass[c];

s\_latch(1);

}

prep\_Wrt2();

prepKeyb();

}

void prepKeyb(void){

unsigned char chk=0,c,d;

unsigned char alpha[6];

do{

C1=C2=C3=1;

R1=R2=R3=R4=0; //ground all rows at once

if(C1==0){ //see if any key is pressed

alpha[chk]=rowFinder1();

chk++;

}

else if(C2==0){ //see if any key is pressed

alpha[chk]=rowFinder2();

chk++;

}

else if(C3==0){ //see if any key is pressed

alpha[chk]=rowFinder3();

chk++;

}

}while(chk<6); //loop until codes are entered 6 digits

for(d=0;d<6;d++){

if(defPass[d]!=alpha[d]){

clr();

prep\_Wrt();

for(c=0;c<14;c++){

LCD=asc\_Misc[c];

s\_latch(1);

}

prep\_Wrt2();

for(c=0;c<14;c++){

LCD=asc\_Redo[c];

s\_latch(1);

}

userInit();

break;

}}

clr();

prep\_Wrt();

for(c=0;c<2;c++){

LCD=asc\_Success[c];

s\_latch(1);

}

MOTOR\_1=1;

MOTOR\_2=0;

clr();

prep\_Wrt();

for(c=0;c<2;c++){

LCD=asc\_openDoor[c];

s\_latch(1);

}

delay\_50ms(100);

MOTOR\_1=0;

MOTOR\_2=1;

clr();

prep\_Wrt();

for(c=0;c<2;c++){

LCD=asc\_closDoor[c];

s\_latch(1);

}

delay\_50ms(100);

MOTOR\_1=0;

MOTOR\_2=0;

}

unsigned char rowFinder1(void){

R1=R2=R3=R4=1;

C1=C2=C3=0;

if(R1==0){

postKyData('1');

return '1';

}

else if(R2==0){

postKyData('4');

return '4';

}

else if(R3==0){

postKyData('7');

return '7';

}

else if(R4==0){

postKyData('\*');

return '\*';

}

}

unsigned char rowFinder2(void){

R1=R2=R3=R4=1;

C1=C2=C3=0;

if(R1==0){

postKyData('2');

return '2';

}

else if(R2==0){

postKyData('5');

return '5';

}

else if(R3==0){

postKyData('8');

return '8';

}

else if(R4==0){

postKyData('0');

return '0';

}

return '$';

}

unsigned char rowFinder3(void){

R1=R2=R3=R4=1;

C1=C2=C3=0;

if(R1==0){

postKyData('3');

return '3';

}

else if(R2==0){

postKyData('6');

return '6';

}

else if(R3==0){

postKyData('9');

return '9';

}

else if(R4==0){

postKyData('#');

return '#';

}

return '$';}

void postKyData(unsigned char dt){

while(R1==0||R2==0||R3==0||R4==0){}

LCD=dt;

s\_latch(1);

LCD=0x6; //shift cursor right

delay\_50ms(1);//give LCD some time

}

void class\_init(void){

unsigned char c;

user\_switch=1;

ALARM\_TRIG=0;

while(1){

prep\_Wrt();

for(c=0;c<10;c++){

LCD=asc\_Greet1[c];

s\_latch(1);

if(user\_switch==0)

break;

}

if(user\_switch==0){

clr();

break;

}

prep\_Wrt2();

for(c=0;c<8;c++){

LCD=asc\_Greet2[c];

s\_latch(1);

if(user\_switch==0)

break;

}

if(user\_switch==0){

clr();

break;

}

clr();

}

prep\_Wrt();

for(c=0;c<9;c++){

LCD=asc\_Ndpass1[c];

s\_latch(1);

}

prep\_Wrt2();

for(c=0;c<11;c++){

LCD=asc\_Ndpass2[c];

s\_latch(1);

}

userInit();

s\_latch(1);

s\_latch(1);

clr();

set\_sett();

LCD=0x02; //Command that cauuses the LCD Return home

s\_latch(1);

P1=0;

class\_init();

}

**2.6 WORKING PRINCIPLE:**

Password based door lock system using 8051 microcontroller circuit design uses five major components – a Microcontroller, an L293D Motor Driver, a DC Motor, a 4×4 Matrix Keypad and a 16×2 LCD. Here, an AT89C52 Microcontroller is used and it is an 8-bit controller. This controller requires a supply voltage of +5V DC. In order to provide regulated 5V DC voltage to the controller we need to use 7805 power supply circuit. We can use 9V DC battery or 12V, 1A adaptor as a power source.

**Reset Circuit Design:**The reset pin of the microcontroller is kept active till the power supply is in the specified range and a minimum oscillation level is maintained.  In other words to ensure the supply voltage does not falls below the threshold level of 1.2V and the reset pulse width is greater than 100ms (recommended for 89C52),  we need to select the values of resistor and capacitor such that RC >=100ms.  Hence, we selected a 10KΩ resistor and a 10µF electrolytic capacitor.

**Oscillator Circuit Design:** An 11.0592MHz crystal oscillator is used to provide external clock signal to the microcontroller. To ensure smooth operation, we need to connect two ceramic capacitors in the range of 30pF to 40pF. This crystal oscillator is connected between pin 18 and 19 of the microcontroller. Here, we used two 33pF capacitors.

**Interfacing LCD, Keypad and Motor Driver:** First, a 10KΩ Potentiometer is connected to the LCD Display’s Contrast Adjust Pin (Pin 3). RS, RW and E of LCD are connected to P3.0, GND and P3.2 pins respectively. The eight data lines of the LCD are connected to PORT1.

The four ROW pins of the Keypad are connected to P2.0 to P2.3 and the four COLUMN pins of the Keypad are connected to P2.4 to P2.7 pins respectively. The IN1 and IN2 of (1A and 2A) of the L293D Motor Driver are connected to PORT0 pins P0.0 and P0.1. Motor is connected between OUT1 and OUT2 (1Y and 2Y) pins of L293D.

**Compilation of Microcontroller Code:**Once the circuit is designed and drawn on a piece of paper, the next step is to write and compile the code. Here, we used the Keil µVision software to write the program in C language.

Prior to writing the code, general steps needs to be followed like creating a new project and selecting the target device or the required microcontroller. Once the code is written, we need to save it with .c extension and then add it to the source file group under the target folder.  The code is then compiled by pressing F7 key.

Once the code is compiled, a hex file is created.  In the next step, we use Proteus software to draw the circuit. The code is dumped into the microcontroller using an external programmer and Willar Software.

**CHAPTER-3**

### **Advantages of Password Based Door Lock System**

* This project provides security
* Power consumption is less
* Used commonly available components
* Project is simple and easy

### **Applications of Password Based Door Lock System**

* 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.

### **Limitations of Password Based Door Lock System**

* It is a low range circuit, i.e. it is not possible to operate the circuit remotely.
* If you forget the password it is not possible to open the door.

**INFERENCE:**

1. The stepped-down AC voltage is rectified by bridge rectifier and filtered by capacitor before it is fed to the IC.
2. To change the password no change in the circuit is required.
3. The motor is controlled by an electrical impulse, which is triggered by the keypad.
4. Digital door locks are powered by electricity, if your house or building has a power failure, then the door lock will not work which restricts the user from entering the locked place.
5. The system will tell the customer to replace the battery.We should replace the battery before the battery is completely discharged, thus ensuring that the battery does not drain and damage the boiler
6. We must keep the door lock away from corrosive substances,any corrosive solution shouldn’t be used

to clean the lock.