VIT®

Vellore Institute of Technology

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

Fall 2020-21

ECE3003 - Microcontroller and its ApplicationsProject Report

TITLE

Password Based Circuit Breaker

Team Members

1. **K.Tarun Sai Chowdary – 18BEC0052**
2. **R. Goutham – 18BEC0019**

Faculty

Padmini T.N

Abstract:

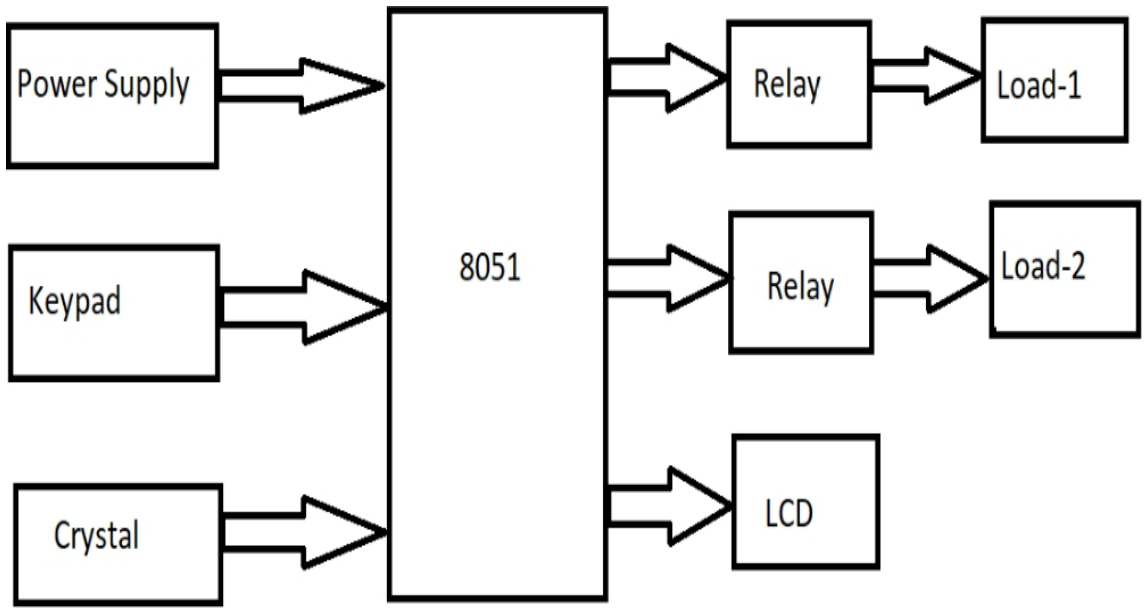
Nowadays, electrical accidents to the line man are increasing,while repairing the electrical lines due to the lack of communication between the electrical substation and maintenance staff. This project gives a solution to this problem to ensure line man safety. In this proposed system, the control (ON/OFF) of the electrical lines lies with line man.This project is arranged in such a way that maintenance staff or line man has to enter the password to ON/OFF the electrical line.

Now, if there is any fault in electrical line, then the line man will switch off the power supply to the line by entering password and comfortably repair the electrical line, and aftercoming to the substation line man switch on the supply to the particular line by entering the password. Separate passwords are assigned for each electrical line.

Introduction:

Nowadays, the current power system deals with huge powernetwork as well as associated electrical equipment. During the electrical fault or short circuit, the power network will suffer from a high stress of fault current in them which may harm the equipment permanently. For conserving the power networks and equipment, the fault current should be very cleared from the system as fast as possible. To overcome this problem, the proposed system password based circuitbreaker gives a solution to ensure lineman security.

Block-Diagram:

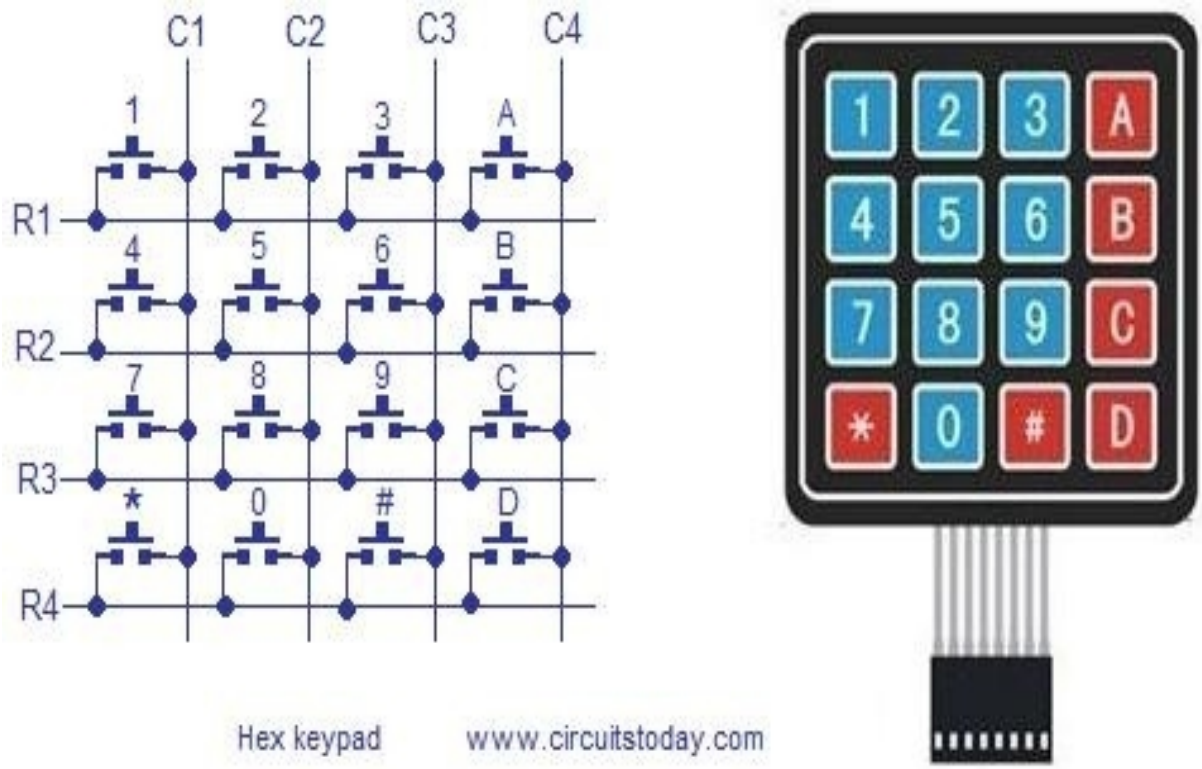


Overview:

Hex keypad:

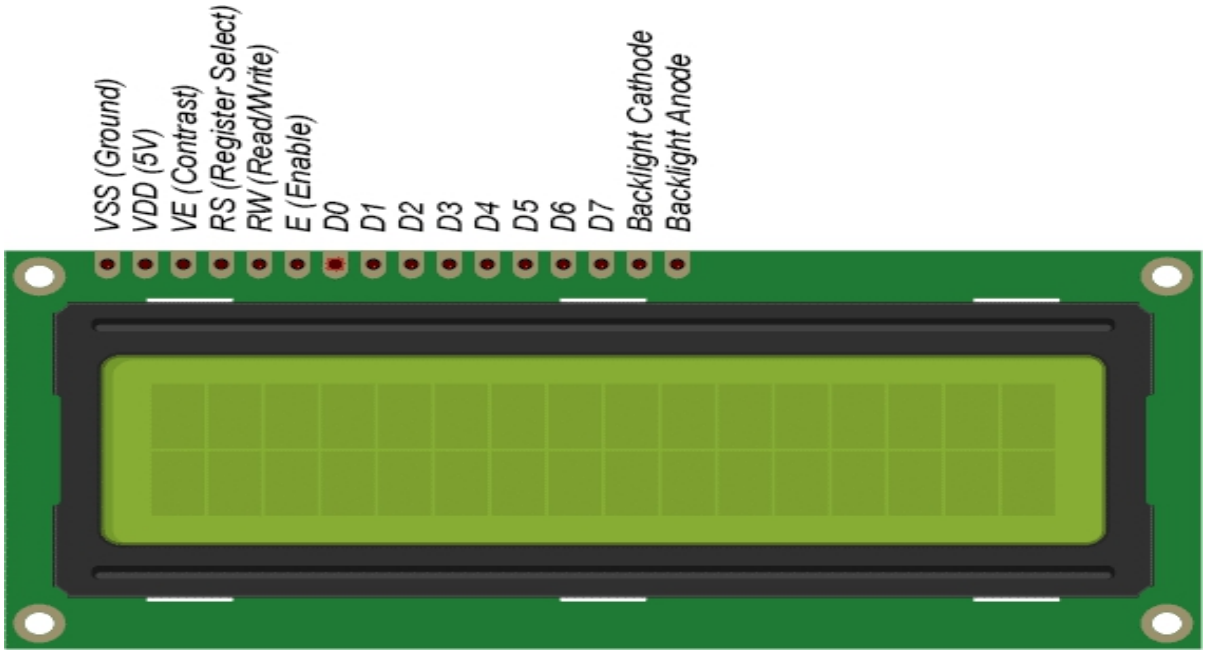
Hex key pad is essentially a collection of 16 keys arranged in the form of a 4×4 matrix. Hex key pad usually have keys representing numerics 0 to 9 and characters A to F. The simplified diagram of a typical hex key pad is shown in thefigure below.

The hex keypad has 8 communication lines namely R1, R2, R3, R4, C1, C2, C3 and C4. R1 to R4 represents the four rows and C1 to C4 represents the four columns. When a particular key is pressed the corresponding row and column to which the terminals of the key are connected gets shorted. For example if key 1 is pressed row R1 and column C1 gets shorted and so on. The program identifies which key is pressed by a method known as column scanning. In this method a particular row is kept low (other rows are kept high) and the columns are checked for low .If a particular column is found low then that means that the key connected between that column and the corresponding row (the row that is kept low) is been pressed. For example if row R1 is initially kept low and column C1 is found low during scanning, that means key 1 is pressed.

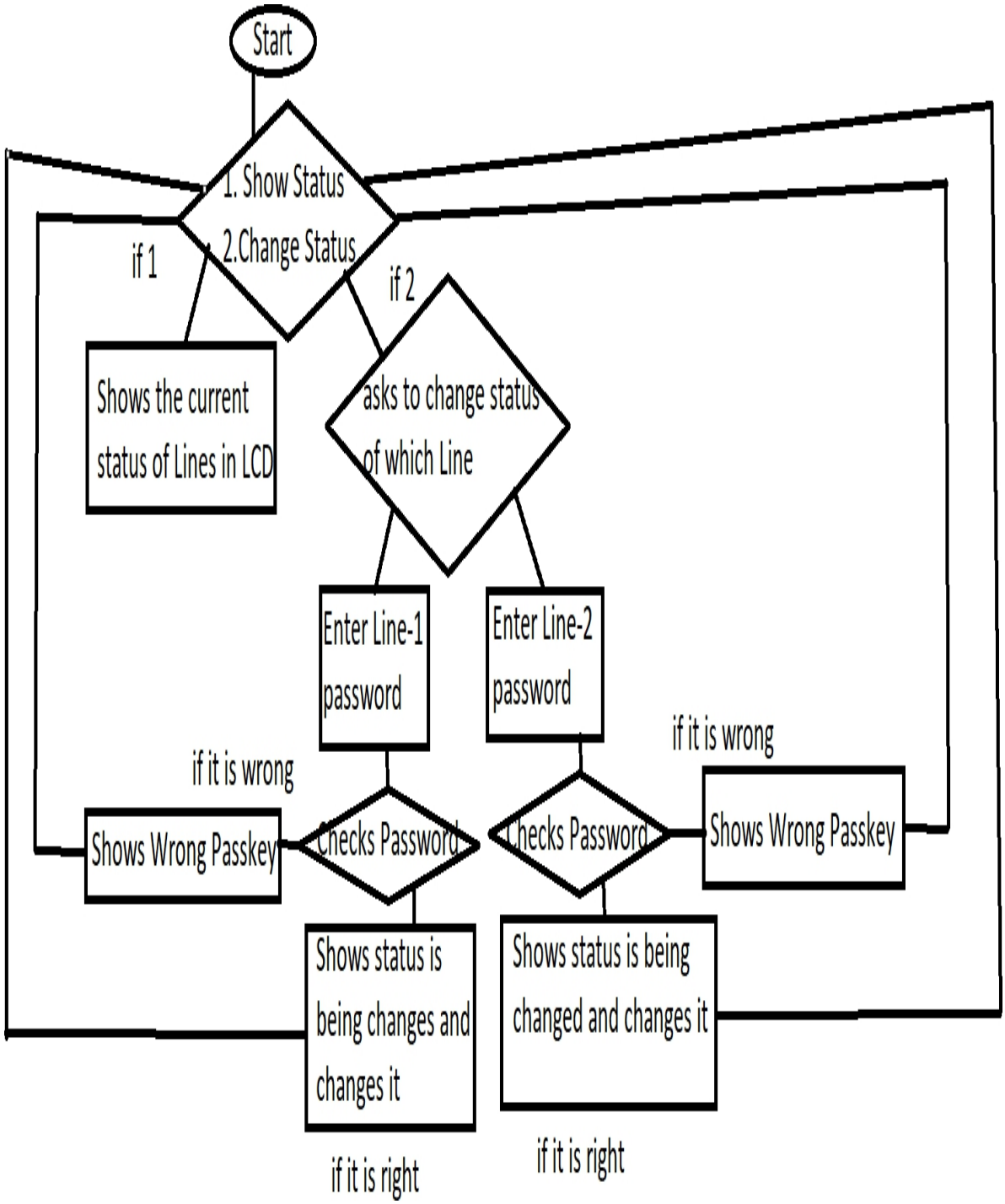


16×2 LCD module:

16×2 LCD module is a very common type of LCD module that is used in 8051 based embedded projects. It consists of 16 rows and 2 columns of 5×7 or 5×8 LCD dot matrices. The module were are talking about here is type number JHD162A which is a very popular one . It is available in a 16 pin package with back light ,contrast adjustment function and each dot matrix has 5×8dot resolution.



Flow Chart:


Start
1. Show Status
2.Change Status
if 1
if 2
Shows the current status of Lines in LCD
asks to change status of which Line
Enter Line-1 password
Enter Line-2 password
if it is wrong
if it is wrong
Shows Wrong Passkey
Checks Password
Checks Password
Shows Wrong Passkey
Shows status is being changes and changes it
Shows status is being changed and changes it
if it is right
if it is right

Code:

```
ORG 0000H;

        CLR P2.3;CLR
        P2.4;
        MOV TMOD, #00000001B;ACALL
        LCD_INT;
        ACALL LINE1;

        MOV DPTR, #TEXT1;ACALL
        LCD_OUT; ACALL LINE2;
        MOV DPTR, #TEXT2;ACALL
        LCD_OUT; ACALL  DELAY;
        ACALL DELAY; SAI1:ACALL
        CLRSCR;ACALL LINE1;
        MOV DPTR, #TEXT3;ACALL
        LCD_OUT; ACALL LINE2;

        MOV DPTR, #TEXT4;ACALL
        LCD_OUT;
        ACALL READ_KEYPRESS1;SJMP
        SAI1;


        READ_KEYPRESS1:ACALL KEY_SCAN1;RET


        KEY_SCAN1:MOV P1, #0FFH;CLR P1.2;
        JB P1.4, JUMP1;

        ACALL CLRSCR;
        ACALL LINE1;
        MOV DPTR, #TEXT7;ACALL
        LCD_OUT; MOV A, #06H;
        ACALL LCD_COMMAND;JB P2.3,
        SPRINT1;
```

```
MOV DPTR, #TEXT6;ACALL
LCD_OUT; SJMP SPRINT2;

SPRINT1:MOV DPTR, #TEXT5;ACALL
LCD_OUT;

SPRINT2:

ACALL LINE2;

MOV DPTR, #TEXT8;ACALL
LCD_OUT; MOV A, #06H;
ACALL LCD_COMMAND;JB P2.4,
SPRINT3;
MOV DPTR, #TEXT6;ACALL
LCD_OUT; SJMP SPRINT4;

SPRINT3:MOV DPTR, #TEXT5;

ACALL LCD_OUT;

SPRINT4:ACALL  DELAY; ACALL
DELAY;
ACALL        DELAY;
ACALL DELAY;RET

JUMP1:

JB P1.5, EXIT2; ACALL
READ_LINE;RET

EXIT2:LJMP KEY_SCAN1;

READ_LINE:ACALL CLRSCR;ACALL
LINE1;
MOV DPTR, #TEXT9;ACALL
LCD_OUT;
```

```
ACALL LINE2;
MOV DPTR, #TEXT0;ACALL
LCD_OUT; ACALL
KEYSCAN3; ACALL DELAY;
RET

KEYSCAN3:MOV P1, #0FFH;CLR P1.2;
JB P1.4, LUMP1;
ACALL READ_KEYPRESS2;ACALL
CLRSCR;
ACALL LINE1;
MOV DPTR, #CHKMSG;ACALL
LCD_OUT; ACALL DELAY;
ACALL CHECK_PASSWORD1;RET
LUMP1:JB P1.5, LUMP2;
ACALL READ_KEYPRESS2;ACALL
CLRSCR;
ACALL LINE1;
MOV DPTR, #CHKMSG;ACALL
LCD_OUT; ACALL DELAY;
ACALL CHECK_PASSWORD2;RET
LUMP2: LJMP KEYSCAN3;

READ_KEYPRESS2:ACALL CLRSCRACALL
LINE1
MOV DPTR,#IPMSGACALL
LCD_OUT ACALL LINE2
MOV R0,#5 MOV
R1,#160
ROTATE:ACALL KEYSCAN2MOV
@R1,A
```

```
ACALL LCD_DATA
ACALL DELAY
INC R1
DJNZ R0,ROTATERET

KEYSCAN2:MOV P1, #0FFH;CLR P1.0;
JB P1.4, NEXT1;
MOV A, #55D;
RET
NEXT1: JB P1.5, NEXT2;MOV A,
#56D;
RET
NEXT2: JB P1.6, NEXT3;MOV A,
#57D;
RET
NEXT3: SETB P1.0;CLR
P1.1;
JB P1.4, NEXT4;
MOV A, #52D;
RET
NEXT4: JB P1.5, NEXT5;MOV A,
#53D;
RET
NEXT5: JB P1.6, NEXT6;MOV A,
#54D;
RET
NEXT6: SETB P1.1;CLR
P1.2;
JB P1.4, NEXT7;
```

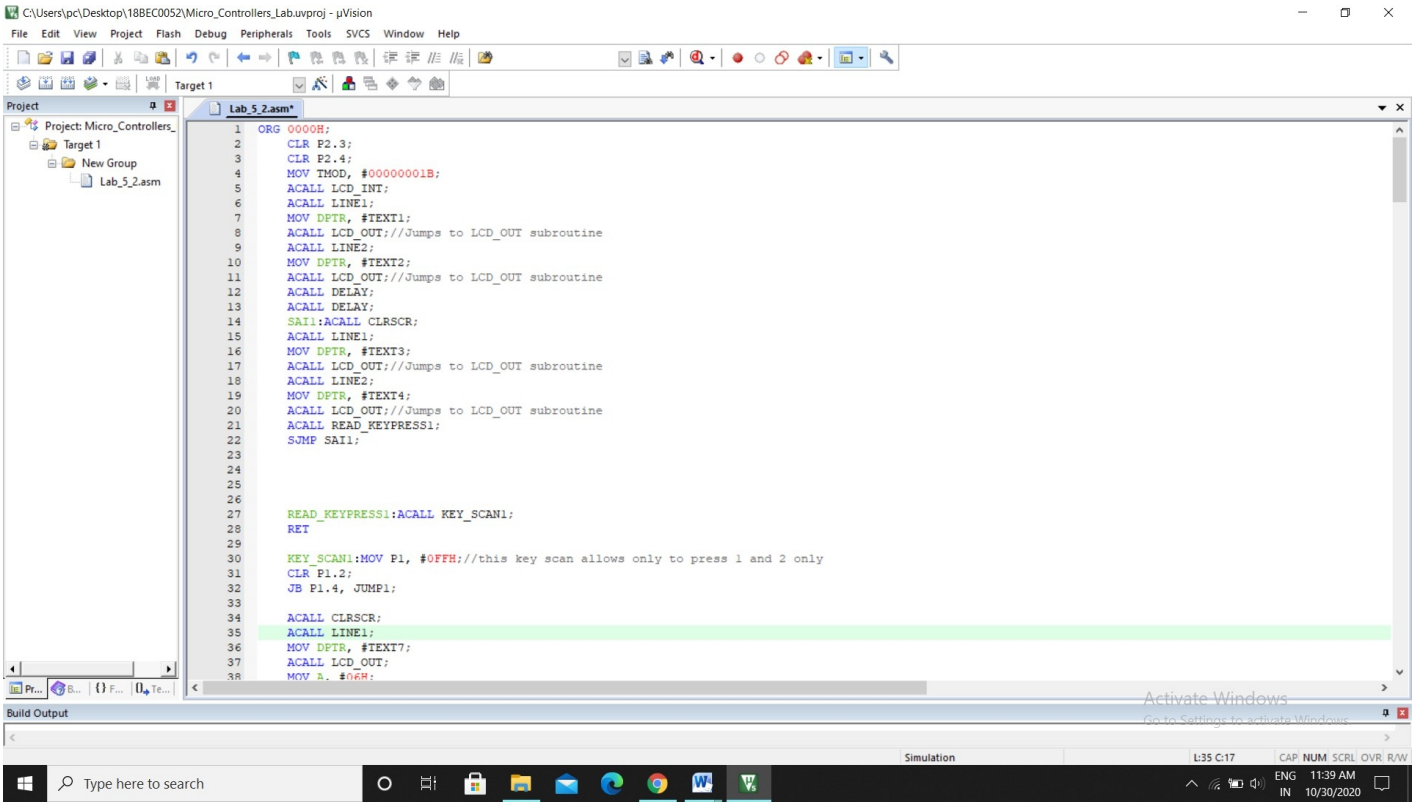
```
MOV A, #49D;

RET

NEXT7: JB P1.5,NEXT8;MOV A,

#50D;

RET

NEXT8: JB P1.6,NEXT9;MOV A,

#51D;

RET

NEXT9: SETB P1.2;CLR

P1.3;

JB P1.5, NEXT10;MOV A,

#48D; RET

NEXT10:LJMP KEYSCAN2;


CHECK_PASSWORD1:MOV R0,#5;MOV R1,#160;

MOV DPTR,#PASSWORD1;RPT:CLR A;

MOVC A,@A+DPTR;XRL

A,@R1;

JNZ FAIL;

INC R1;

INC DPTR;

DJNZ R0,RPT;

ACALL CLRSCR;
ACALL LINE1;

MOV DPTR,#TEXT_S1ACALL

LCD_OUT

JNB P2.3, SUCCESS;CLR

P2.3;
```

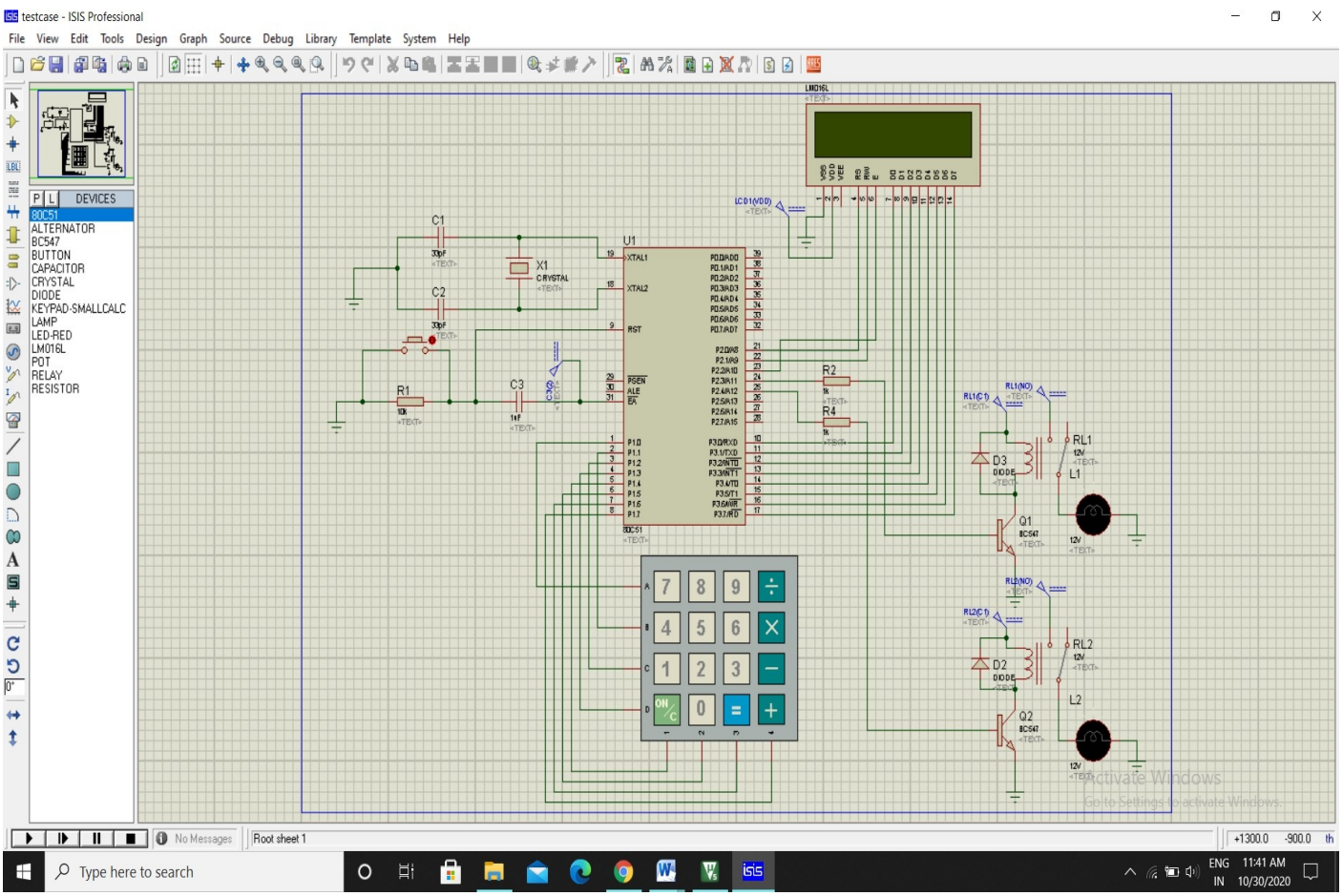
```
SJMP GOBACK; SUCCESS:
SETB P2.3;SJMP GOBACK
FAIL:ACALL CLRSCR ACALL
LINE1
MOV DPTR,#TEXT_F1ACALL
LCD_OUT ACALL DELAY
ACALL LINE2
MOV DPTR,#TEXT_F2ACALL
LCD_OUT ACALL DELAY
GOBACK:RET
SJMP GOBACK1
FAIL1:ACALL CLRSCR
ACALL LINE1
MOV DPTR,#TEXT_F1ACALL
LCD_OUT ACALL DELAY
ACALL LINE2
MOV DPTR,#TEXT_F2ACALL
LCD_OUT ACALL DELAY
GOBACK1:RET

LCD_INT:MOV DPTR, #INIT_COMMANDS;ACALL LCD_IN;
RET

LCD_IN:MOV A,#00H;MOVC A,
@A+DPTR; JZ EXIT0;
```

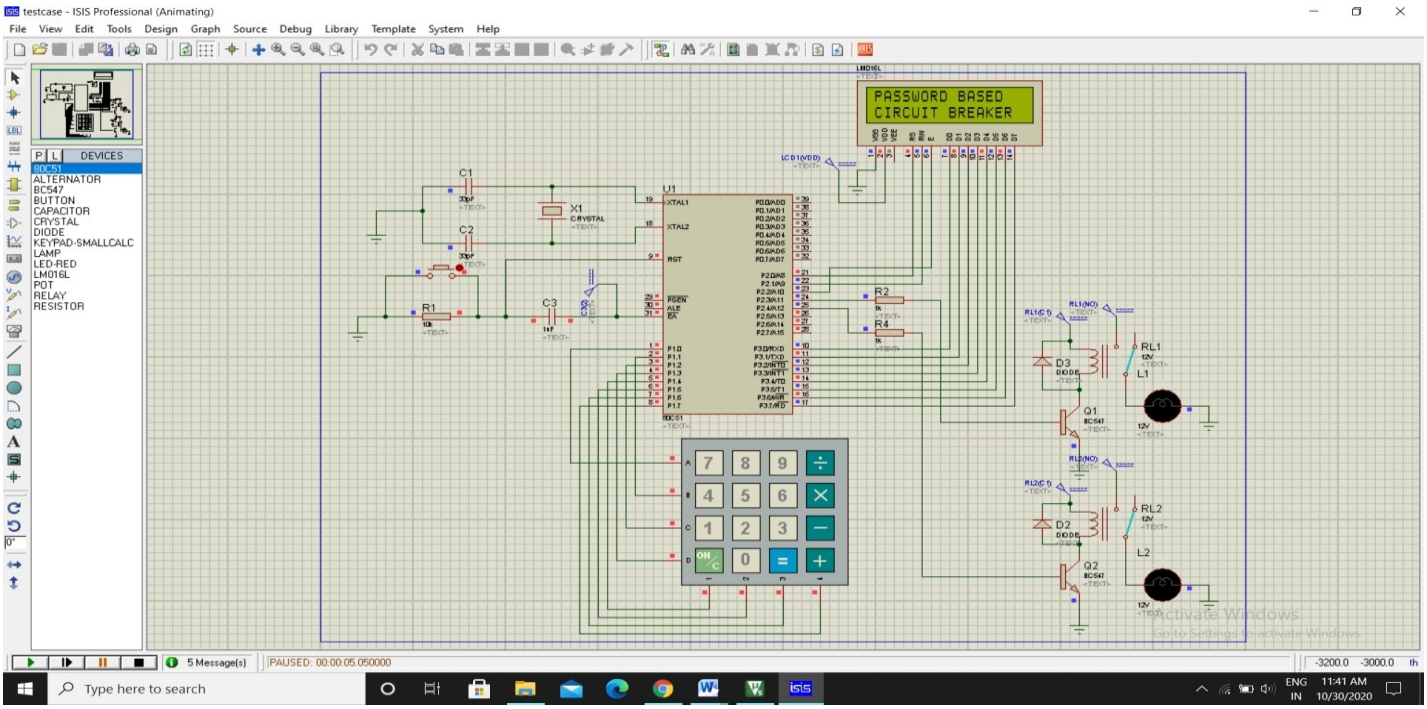
```
ACALL LCD_COMMAND;INC
DPTR;
SJMP LCD_IN;EXIT0:
RET

LCD_OUT:MOV A, #00H;MOVC A,
@A+DPTR;
JZ EXIT1;
ACALL LCD_DATA;INC
DPTR;
SJMP LCD_OUT;
EXIT1: RET

LCD_COMMAND: MOV P3,A;CLR P2.0;
CLR P2.1; SETB
P2.2;
ACALL DELAY;
CLR P2.2;
RET
LCD_DATA:MOV P3,A;SETB
P2.0;
CLR P2.1; SETB P2.2;
ACALL DELAY;CLR
P2.2;
RET
CLRSCR:MOV A, #01H; ACALL
LCD_COMMAND;RET

LINE1:MOV A, #80H; ACALL
LCD_COMMAND;RET
```

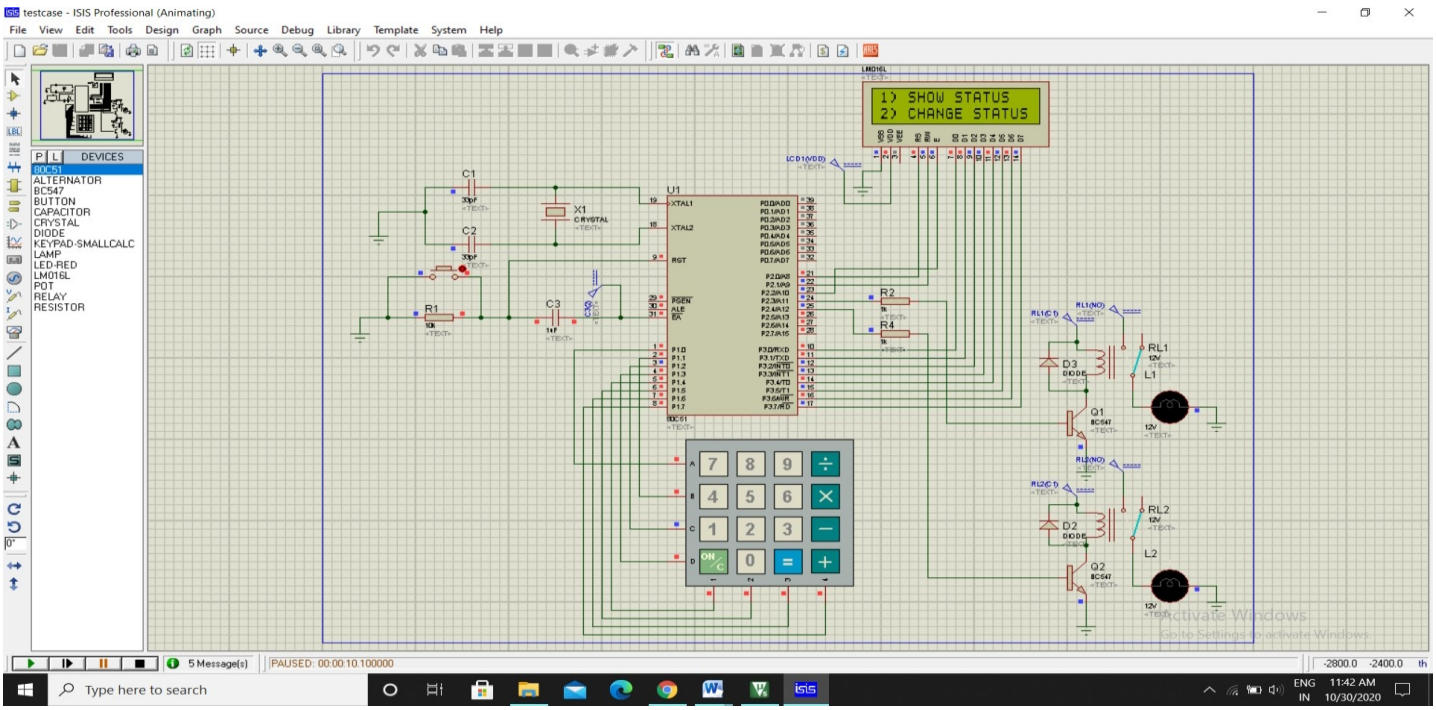
```
LINE2:MOV A, #0C0H; ACALL
LCD_COMMAND;RET

DELAY:MOV R2, #250;

LL1:MOV R3, #250;
LL2:DJNZ R3, LL2; DJNZ R2,
LL1;
RET

INIT_COMMANDS:  DB 0CH,01H,06H,80H,3CH,0TEXT1: DB
"PASSWORD BASED",0
TEXT2: DB "CIRCUIT BREAKER",0TEXT3: DB
"1) SHOW STATUS", 0
TEXT4: DB "2) CHANGE STATUS", 0TEXT5: DB
"ON", 0
TEXT6: DB "OFF", 0

TEXT7: DB "LINE1 : ", 0TEXT8:
DB "LINE2 : ", 0

TEXT9: DB "1) CHANGE LINE1", 0

TEXT0: DB "2) CHANGE LINE2", 0 IPMSG: DB "ENTER
PASSKEY", 0 CHKMSG: DB "CHECKING PASSKEY", 0
TEXT_S1: DB "STATUS CHANGED",0 TEXT_F1: DB
"WRONG PASSKEY",0 TEXT_F2: DB "ACCESS
DENIED",0

PASSWORD1: DB 49D, 50D, 51D, 52D, 53D, 0

PASSWORD2: DB 48D, 50D, 53D, 56D, 48D, 0

END
```

Results:

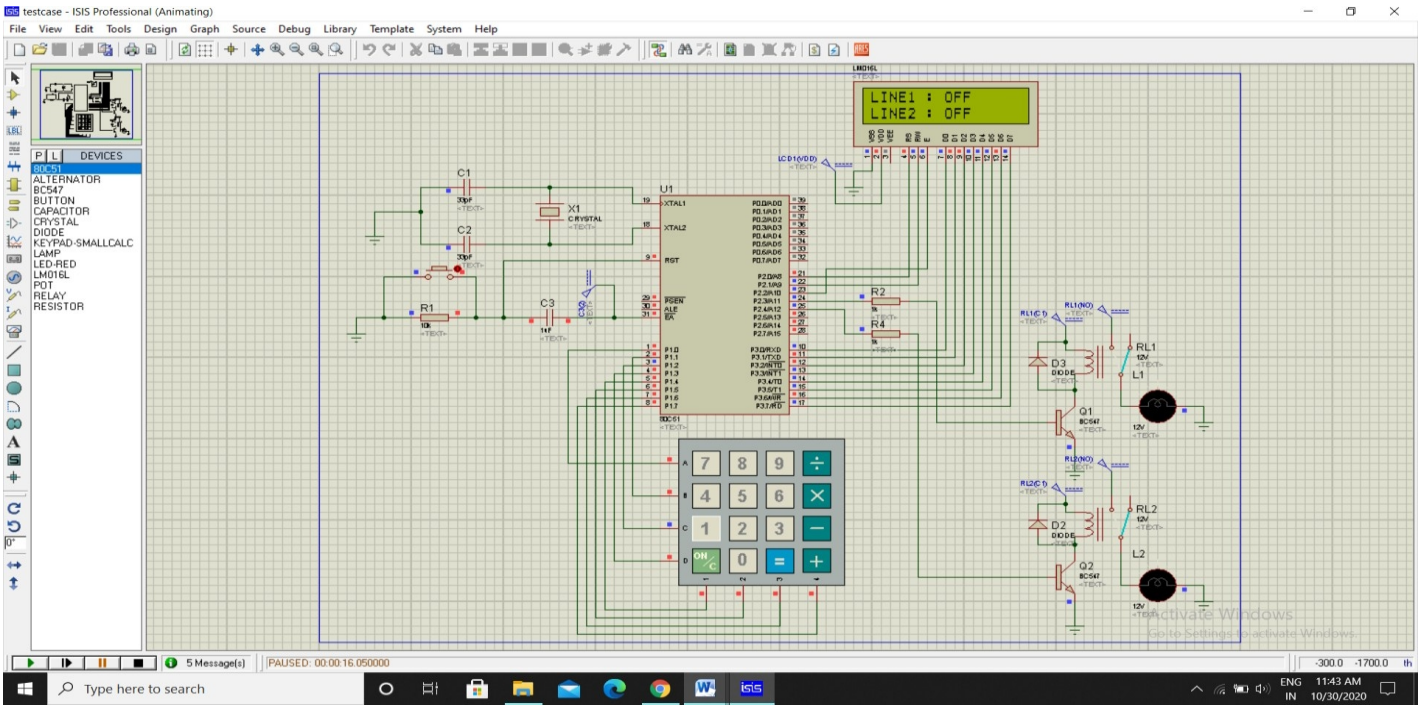

C:\Users\pc\Desktop\18BEC0052\Micro_Controllers_Lab.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: Micro_Controllers_
Target 1
New Group
Lab_5_2.asm
Lab_5_2.asm*
1 ORG 0000H;
2 CLR P2.3;
3 CLR P2.4;
4 MOV TMOD, #00000001B;
5 ACALL LCD_INT;
6 ACALL LINE1;
7 MOV DPTR, #TEXT1;
8 ACALL LCD_OUT;//Jumps to LCD_OUT subroutine
9 ACALL LINE2;
10 MOV DPTR, #TEXT2;
11 ACALL LCD_OUT;//Jumps to LCD_OUT subroutine
12 ACALL DELAY;
13 ACALL DELAY;
14 SAI1:ACALL CLRSCR;
15 ACALL LINE1;
16 MOV DPTR, #TEXT3;
17 ACALL LCD_OUT;//Jumps to LCD_OUT subroutine
18 ACALL LINE2;
19 MOV DPTR, #TEXT4;
20 ACALL LCD_OUT;//Jumps to LCD_OUT subroutine
21 ACALL READ_KEYPRESS1;
22 SJMP SAI1;
27 READ_KEYPRESS1:ACALL KEY_SCAN1;
28 RET
30 KEY_SCAN1:MOV P1, #0FFH;//this key scan allows only to press 1 and 2 only
31 CLR P1.2;
32 JB P1.4, JUMP1;
34 ACALL CLRSCR;
35 ACALL LINE1;
36 MOV DPTR, #TEXT7;
37 ACALL LCD_OUT;
Build Output
Activate Windows
Go to Settings to activate Windows.
Simulation
L:35 C:17
CAP NUM SCRL OVR R/W
Type here to search
ENG IN 11:39 AM 10/30/2020


testcase - ISIS Professional
File View Edit Tools Design Graph Source Debug Library Template System Help
DEVICES
80C51
ALTERNATOR
BC547
BUTTON
CAPACITOR
CRYSTAL
DIODE
KEYPAD-SMALLCALC
LAMP
LED-RED
LM016L
POT
RELAY
RESISTOR
U1
C1
C2
C3
X1
R1
R2
R4
RL1
RL2
D3
D2
Q1
Q2
L1
L2
Activate Windows
Go to Settings to activate Windows.
No Messages
Root sheet 1
Type here to search
ENG IN 11:41 AM 10/30/2020

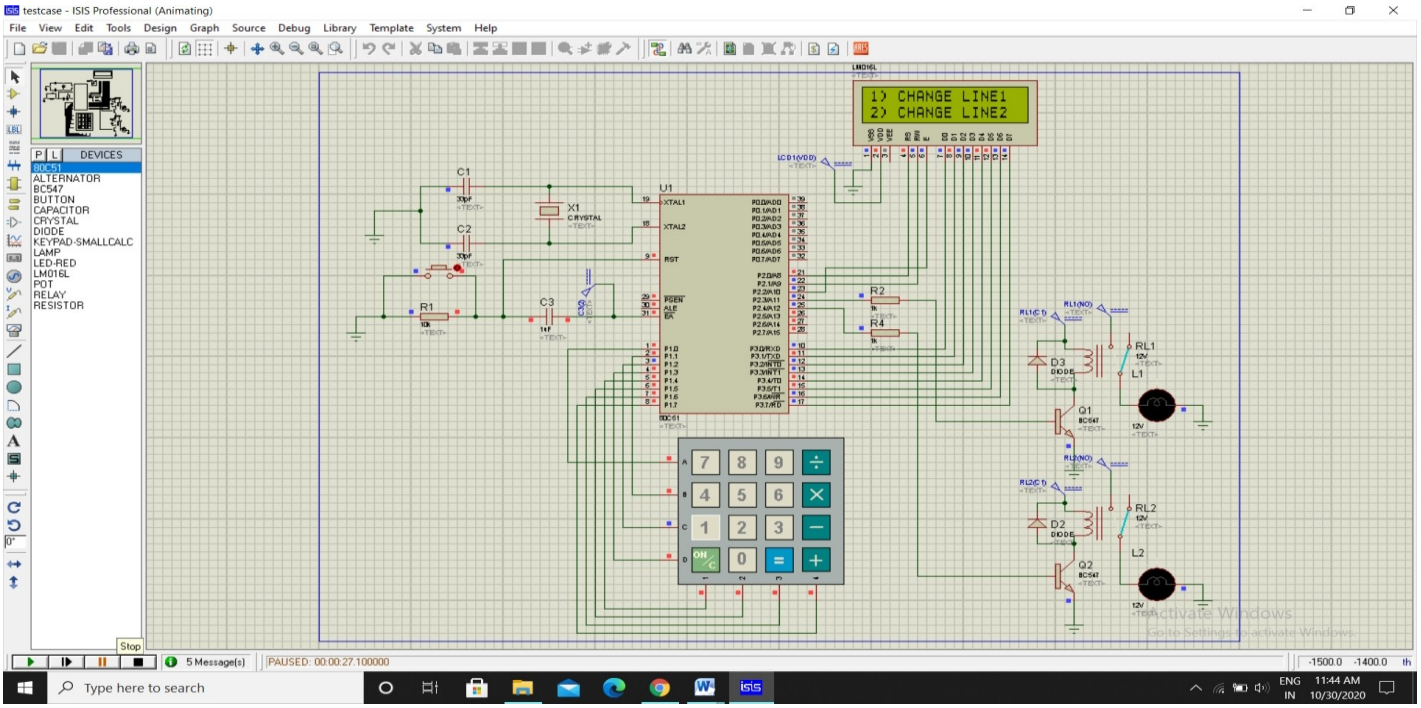




Now If we press key1 it shows the current status of the line.

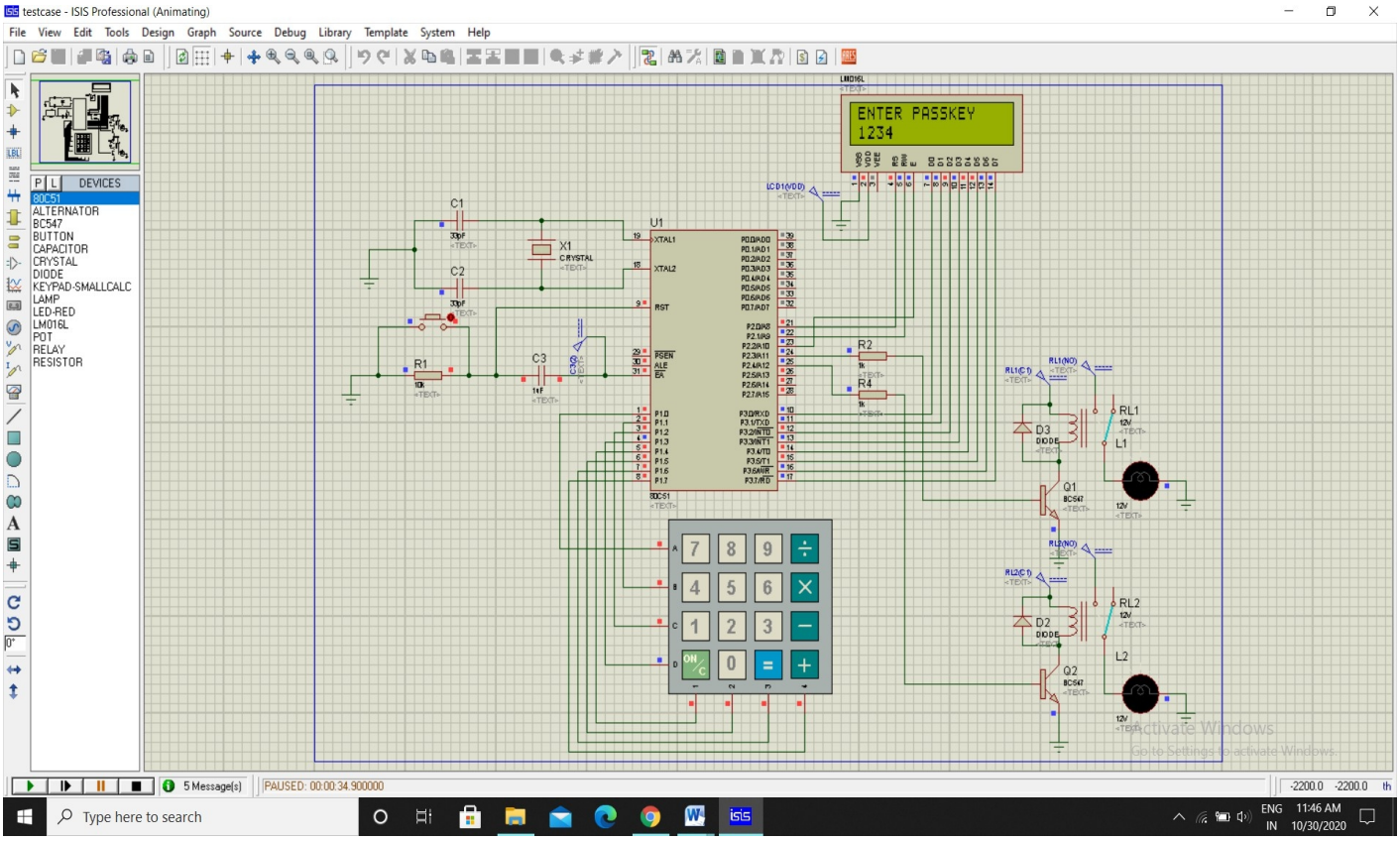


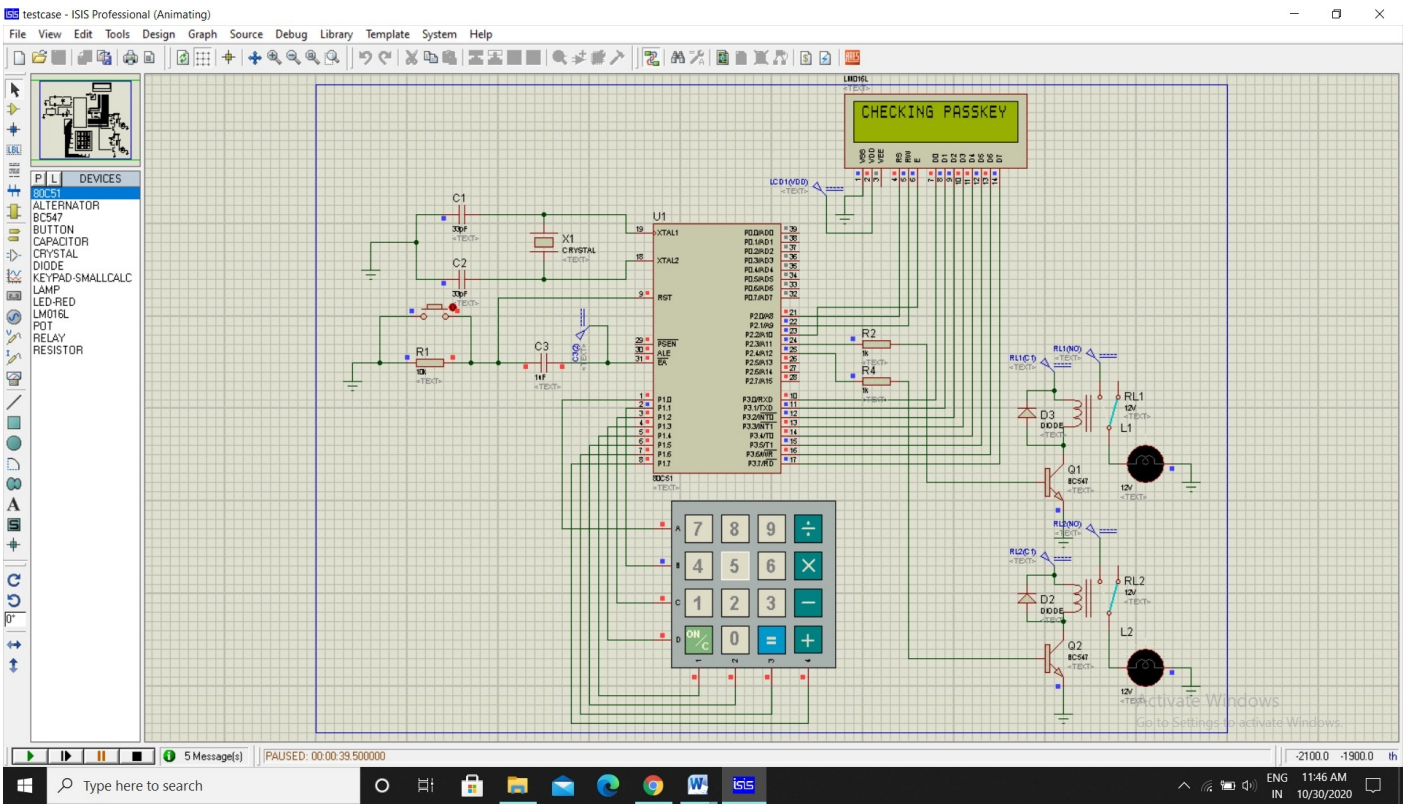
Now if we press key2 we will be asked to change which line status.

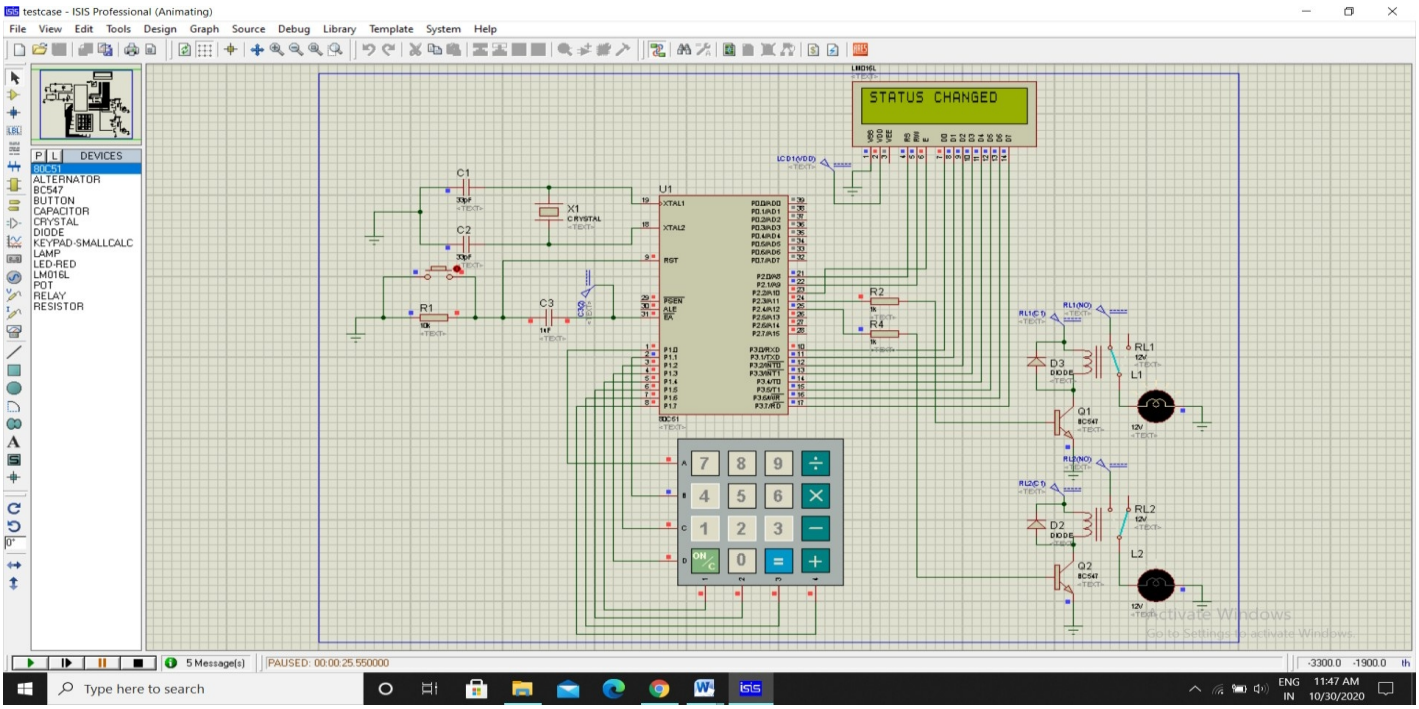


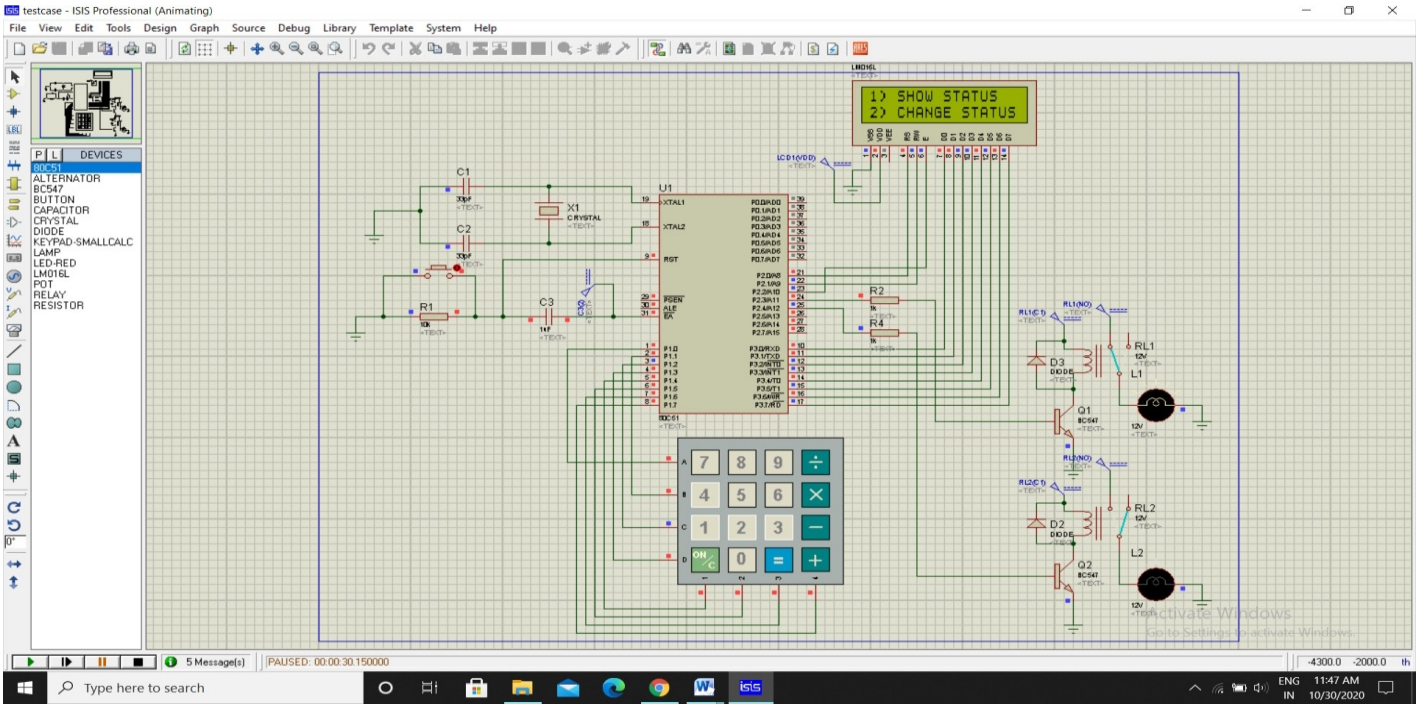
Now if we press key1 it asks us to enter the password

For Line-1 it is given ‘12345’ as password and ‘02580’ for Line-2


testcase - ISIS Professional (Animating)
File View Edit Tools Design Graph Source Debug Library Template System Help
ENTER PASSKEY
1234
DEVICES
80C51
ALTERNATOR
BC547
BUTTON
CAPACITOR
CRYSTAL
DIODE
KEYPAD-SMALLCALC
LAMP
LED-RED
LM016L
POT
RELAY
RESISTOR
U1
X1
CRYSTAL
C1
C2
C3
R1
R2
R4
RL1
RL2
D2
D3
Q1
Q2
L1
L2
5 Message(s)
PAUSED: 00:00:34.900000
Type here to search
ENG
IN
11:46 AM
10/30/2020


testcase - ISIS Professional (Animating)
File View Edit Tools Design Graph Source Debug Library Template System Help
CHECKING PASSKEY
DEVICES
80C51
ALTERNATOR
BC547
BUTTON
CAPACITOR
CRYSTAL
DIODE
KEYPAD-SMALLCALC
LAMP
LED-RED
LM016L
POT
RELAY
RESISTOR
U1
X1
CRYSTAL
C1
C2
C3
R1
R2
R4
RL1
RL2
D2
D3
Q1
Q2
L1
L2
5 Message(s)
PAUSED: 00:00:39.500000
Type here to search
ENG
IN
11:46 AM
10/30/2020





Now we can see that the Line-1 is ON

If we again follow the same procedure we change the line-2 status andcan check the current status by pressing key1.

Conclusion:

Hence we created the Password based circuit breaker circuitsuccessfully.

References :

Took the basic idea to interface LCD and Keypad with 8051 from onlineWebsites and the remaining code was created by us own.

Website references:

1. https://www.circuitstoday.com/interfacing-16x2-lcd-with-8051
2. https://www.circuitstoday.com/interfacing-hex-keypad-to-8051