JAMIA MILLIA ISLAMIA, NEW DELHI

EMBEDDED SYSTEMS LAB

SUBJECT CODE: CEN 691

SEMESTER: 6th

COURSE: B.TECH.(COMPUTER ENGG.)

DEPT: DEPT OF COMPUTER ENGG.

SUBMITTED BY:

BUSHRA SHAHZAD

(21BCS046)

JAZIB ANSARI (21BCS065)

RAMSHA ANWAR (21BCS066)

RUMMAN AHMAD (21BCS067)

SUBMITTED TO:

PROF BASHIR ALAM

DR. WASEEM AHMED

MR HANNAN MANSOOR

INDEX

	Title	Date	Sign
1	Study of Keil Micro-vision IDE and Flash magic tool.	31-01-2024	
2	Design and implement Embedded System for blinking Two LEDs alternately with some delay in between, using 8051 Microcontroller and Keil.	31-01-2024	
3	To design counter based on four switches as input using 8051 board.	07-02-2024	
4	To design a modulo-16 counter based on four switches that counts from given input and produces buzzer sound, at bit 0000, to indicate the end of counting using 8051 board.	14-02-2024	
5	Design and implement an embedded system that interfaces an 8051 Micro-controller that displays the name and roll no on the LCD screen.	20-03-2024	
6	Design and implement an embedded system that interfaces an 8051 Micro-controller that takes input of a number from switches and displays its factorial on LCD screen.	24-04-2024	

EXPERIMENT 1

AIM

Study of Keil Micro-vision IDE and Flash magic tool.

Hardware Used

PC, 8051 Microcontroller kit.

Software Used

Keil, FlashMagic, and USB Drivers.

Pins Used

Rxd pin is used for serial communication and LED blinking.

Procedure

1. Write C code in Keil µVision.
2. Compile it to Hex Code.
3. Use Debug Menu to test the code and see everything is working as intended.
4. Connect 8051 to PC via USB wire.
5. Open Flash Magic software.
6. Program the 8051's memory by uploading the previously saved Hex code.
7. Disconnect the device and switch on to see if code is working.

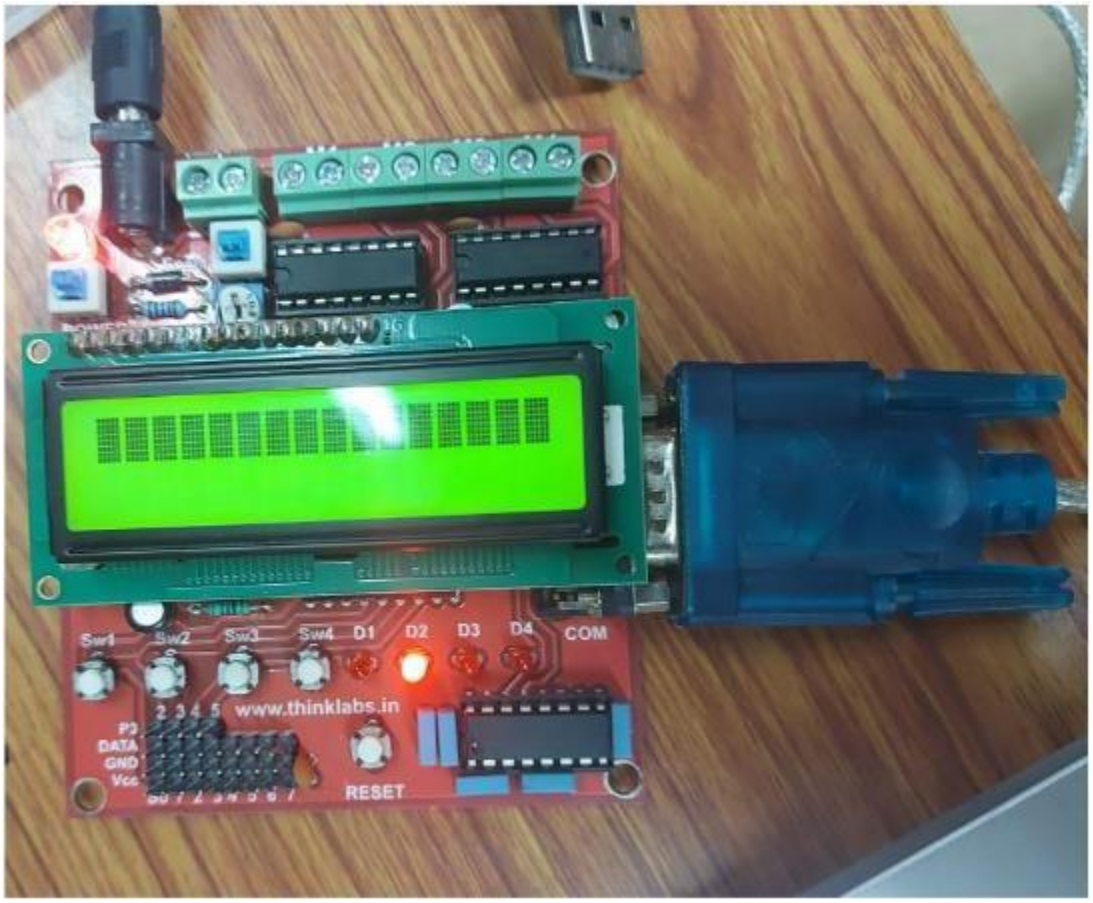
C Code

```
#include "p89v51rx2.h" void
delay(unsigned int x)
{     unsigned int i,
j;
    for (i = 0; i <= 1000; i++)
    {         for (j = 0; j <=
x; j++)
        {
        }
    } } void
main(void)
{     while
(1)
    {

        RxD = 0;
delay(30);
        RxD = 1;
    }
}
```

Result

First LED on 8051 will blink.

Output


Sw1
Sw2
Sw3
Sw4
D1
D2
D3
D4
COM
www.thinklabs.in
P2
DATA
GND
Vcc
RESET

EXPERIMENT 2

AIM

Design and implement Embedded System for blinking Two LEDs alternately with some delay in between, using 8051 Microcontroller and Keil.

Hardware Used

PC, 8051 Micro-controller kit

Software Used

Keil, FlashMagic, and USB Drivers.

Pins Used

LED	PORT	VARIABLE	USE
D1	P3_0	RxD	Serial Data Receive Pin
D2	P3_1	TxD	Serial Data Transmit Pin

Procedure

1. Write C code using Keil µVision software..
2. Compile the code to generate Hex Code.
3. Use the Debug Menu to test the code and ensure it functions as expected.
4. Connect 8051 to PC via USB wire.
5. Open Flash Magic software.
6. Program the 8051's memory by uploading the previously saved Hex code.
7. Disconnect the device and switch on to see if code is working.

C Code

```
#include "p89v51rx2.h" sbit LED3 = P3
^ 6; sbit LED4 = P3 ^ 7;
 void delay(unsigned int
x)
{     unsigned int i,
j;
    for (i = 0; i <= 1000; i++)
    {         for (j = 0; j <=
x; j++)
        {
        }
    } } void
main(void)
{     while
(1)
    {
        RxD = 0;
TxD = 1;
delay(50);
TxD = 0;         RxD
= 1;
delay(20);
    }
}
```

Result

First and second LED on 8051 blink alternatively.

Output

D1 On, D2 Off

D1 Off, D2 On

EXPERIMENT3

AIM

To design counter based on four switches as input using 8051 board.

APPARATUS

- Computer System,
- 8051 Microcontroller kit ● USB connector ● Adapter.

SOFTWARESUSED

● Keil µVision ●
Flash Magic

PINSUSED

LED	PORT	VARIABLE	USE
D1	P3.0	l1	Serial Data Receiver Pin
D2	P3.1	l2	Serial Data Transmit Pin
D3	P3.6	l3	External Memory Write
D4	P3.7	l4	External Memory Read
SW1	P3.2	s0	Switch 1
SW2	P3.3	s1	Switch 2
SW3	P3.4	s2	Switch 3
SW4	P3.5	s3	Switch 4

PROCEDURE

1) This C code is designed to monitor the configuration of switches on an 8051 microcontroller, executing specific operations based on their states.

2) When Switch 1 is activated, it triggers a 1-bit counter along with a predefined delay. Similarly, switches 2 through 4 initiate 2-bit, 3-bit, and 4-bit counters respectively.
3) The code manages interrupts from other switches, executing relevant operations corresponding to each switch's status.

CCODE

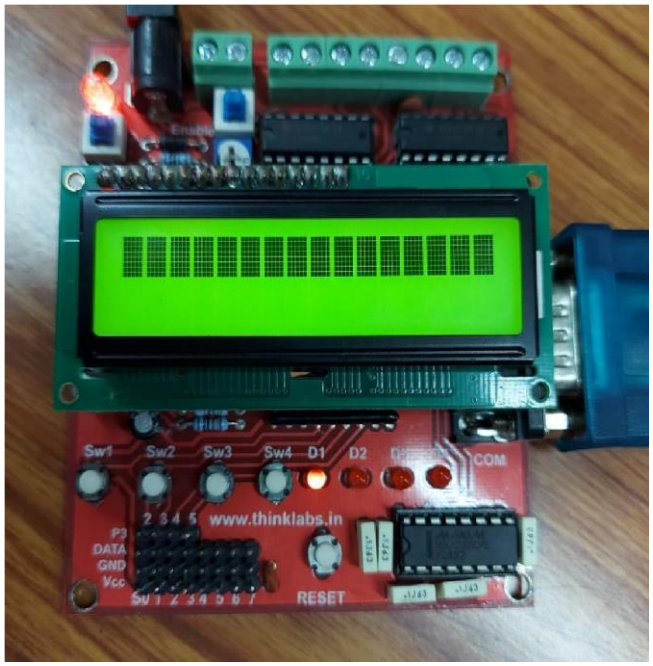
```
#include <p89v51rx2.h>

sbit l1 = P3^0; sbit l2

= P3^1; sbit l3 = P3^6;

sbit l4 = P3^7; sbit s0

= P3^2; sbit s1 = P3^3;

sbit s2 = P3^4; sbit s3

= P3^5;

void delay(unsigned int x){

      unsigned int i,j; for

      (i=0;i<=1000;i++){ for

      (j=0;j<=x;j++){

            }

      }

}


void counter(unsigned int n){

      int i=0; for(i =n-

      1;i>=0;i--){


            if(i & (1){

                  l1 = 0;

            }else{

            l1 =1;

            } if(n>=4){

            if(i & (2)){

            l2 = 0;

            }else{
            l2 =1;

            }
```

```
      } if(n>=8){ if(i &
            (8)){ l3 = 0;
            }else{
            l3 =1;
            }
      }

if(n>=16){ if(i &
      (16)){ l4 = 0;
            }else{
            l4 =1;
            }
      }
delay(200);
}

}


void main(void){

      while (1){ if(s0==0)
      counter(2); if(s1==0)
      counter(4); if(s2==0)
      counter(8); if(s3==0)
      counter(16);

      }

}
```

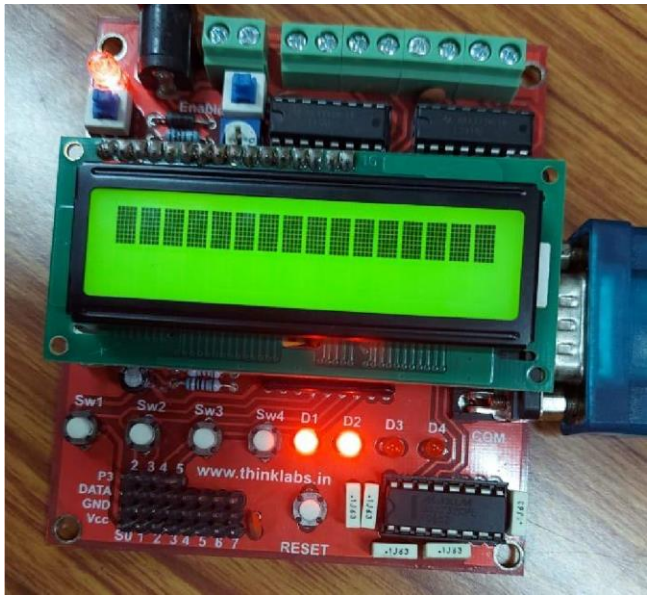
RESULT

This C code is designed to monitor the configuration of switches on an 8051 microcontroller, executing specific operations based on their states. When Switch 1 is activated, it triggers a 1-bit counter along with a

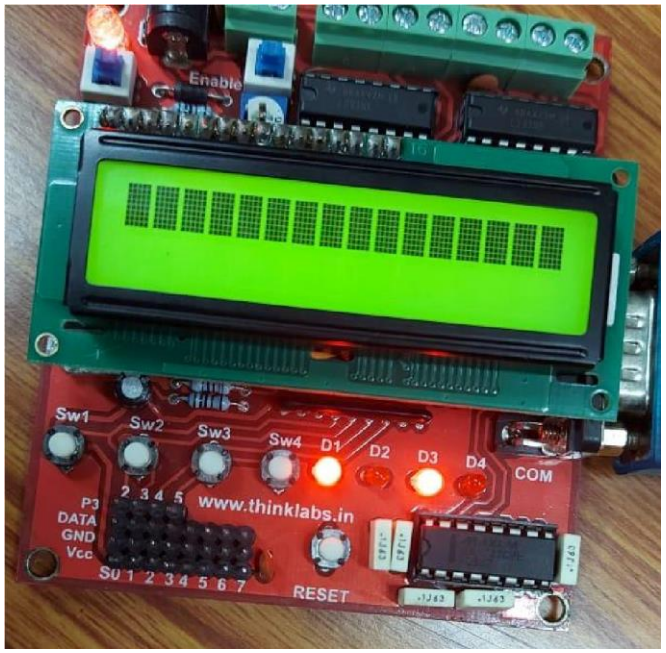
predefined delay. Similarly, switches 2 through 4 initiate 2-bit, 3-bit, and 4-bit counters respectively. Furthermore, the code manages interrupts from other switches, executing relevant operations corresponding to each switch's status.



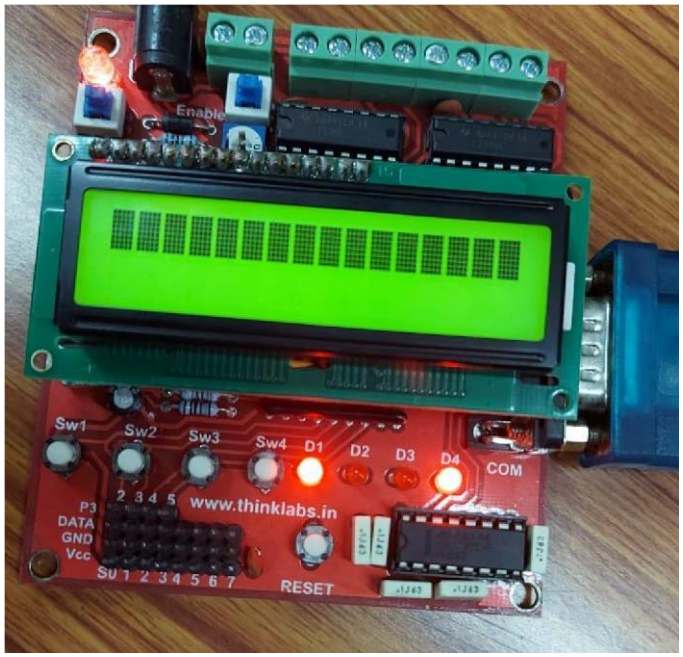
1-bit Counter for switch 1



2-bit Counter for switch 2



3-bit Counter for switch 3



4-bit Counter for switch 4

EXPERIMENT 4

AIM

To design a modulo-16 counter based on four switches that counts from given input and produces buzzer sound, at bit 0000, to indicate the end of counting using 8051 board.

APPARATUS

- Computer System
- 8051 Microcontroller kit
- USB connector
- Adapter

SOFTWARES USED

- Keil µVision
- Flash Magic

PINS USED

LED	PORT	VARIABLE	USE
l1	P3.0	l1	Serial Data Receiver Pin
l2	P3.1	l2	Serial Data Transmit Pin
l3	P3.6	l3	External Memory Write
l4	P3.7	l4	External Memory Read
s0	P3.2	s0	Switch 1
s1	P3.3	s1	Switch 2
s2	P3.4	s2	Switch 3

s3	P3.5	s3	Switch 4

PROCEDURE

1) LEDs (`l1` to `l4`) are connected to port P3, and switches (`s0` to `s3`) are also connected to the same port.
2) The `counter` function takes an integer input `n` and cycles through the LEDs based on the binary representation of `n`, lighting each LED according to its corresponding bit.
3) In the `main` function, an infinite loop continuously monitors the state of the switches.
4) Depending on the combination of switch states, a specific value is passed to the `counter` function, which then controls the LEDs accordingly.
5) The delay function introduces a delay between LED state changes.
6) When the microcontroller is powered on, it waits for switch inputs to determine the LED pattern to display.
7) Different switch configurations trigger different LED counting patterns.
8) After each count sequence, there is a brief pause with the buzzer sounding to indicate the end of the count.

C CODE

```
#include <p89v51rx2.h>

sbit  l1 = P3^0;

sbit  l2 = P3^1;

sbit  l3 = P3^6;

sbit  l4 = P3^7;

sbit  s0 = P3^2;

sbit  s1 = P3^3;

sbit  s2 = P3^4;

sbit  s3 = P3^5;
sbit buzzer = P0^3; void

delay(unsigned int x){

unsigned int i,j; for

(i=0;i<=1000;i++){ for

(j=0;j<=x;j++){

            }

      }

}


void counter(unsigned int n){
```

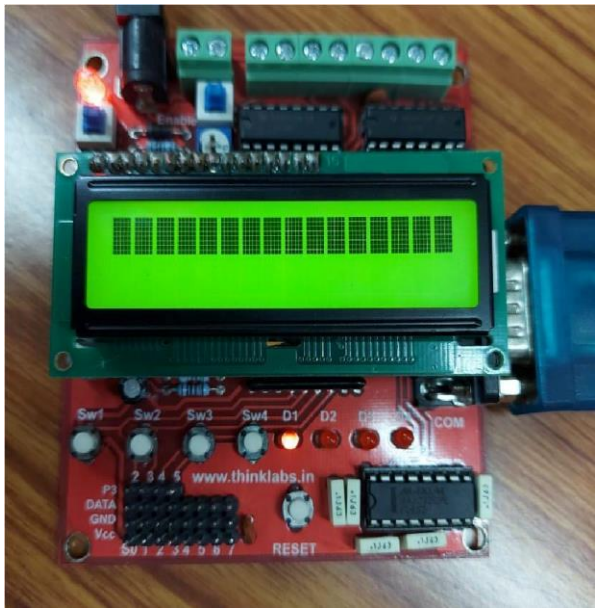
```
      int i=0; while(1){
      for(i =n;i<=16;i++){
      if(i & (1)){ l1 = 0;
            }else{
            l1 =1;
            } if(i &
            (2)){ l2 =
            0;
            }else{
            l2 =1;
            } if(i
      & (4)){ l3 =
      0;
            }else{
            l3 =1;
            } if(i
      & (8)){ l4 =
      0;
            }else{
            l4 =1;
            }

delay(200);
} n=1;
buzzer = 0;
delay(300);
buzzer = 1;
}
}


void main(void){
      while(1)
```

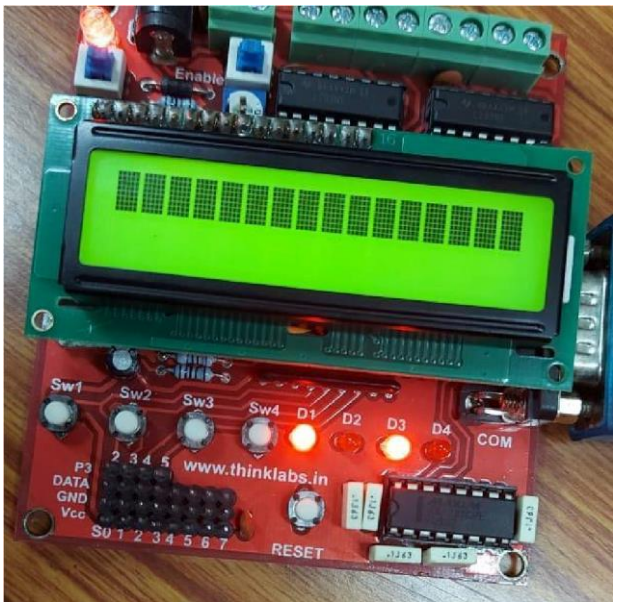
```
    { delay(300);

    if(s0==0 && s1== 1 && s2==1 && s3==1){
     counter(15);
} else if( s0==1 && s1==0 && s2==1 &&
s3==1){
     counter(14);
} else if( s0==0 && s1==0 && s2==1 &&
s3==1){
     counter(13);
} else if( s0==1 && s1==1 && s2==0 &&
 s3==1){
     counter(12);
} else if( s0==0 && s1==1 && s2==0 && s3==1){
     counter(11);
} else if( s0==1 && s1==0 && s2==0 && s3==1){
     counter(10);
} else if( s0==0 && s1==0 && s2==0 && s3==1){
     counter(9);
} else if( s0==1 && s1==1 && s2==1 && s3==0){
     counter(8);
} else if( s0==0 && s1==1 && s2==1 && s3==0){
     counter(7);
} else if( s0==1 && s1==0 && s2==1 && s3==0){
     counter(6);
} else if( s0==0 && s1==0 && s2==1 && s3==0){
     counter(5);
}
        else if( s0==1 && s1==1 && s2==0 && s3==0){
     counter(4);
} else if( s0==0 && s1==1 && s2==0 && s3==0){
     counter(3);
} else if( s0==1 && s1==0 && s2==0 && s3==0){
     counter(2);
```

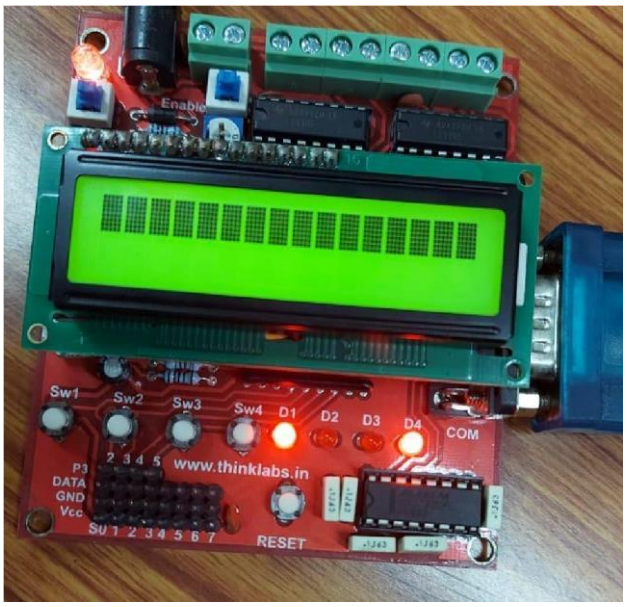
```
 } else if( s0==0 && s1==0 && s2==0 && s3==0){
       counter(1);
  }
}

}
```

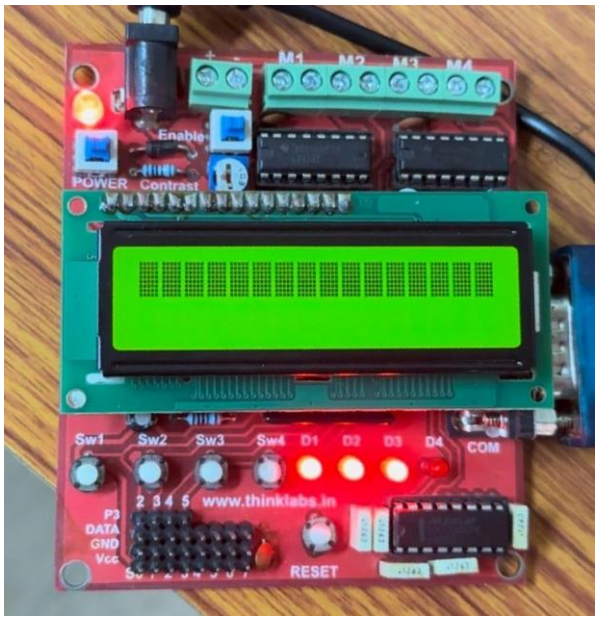
RESULT

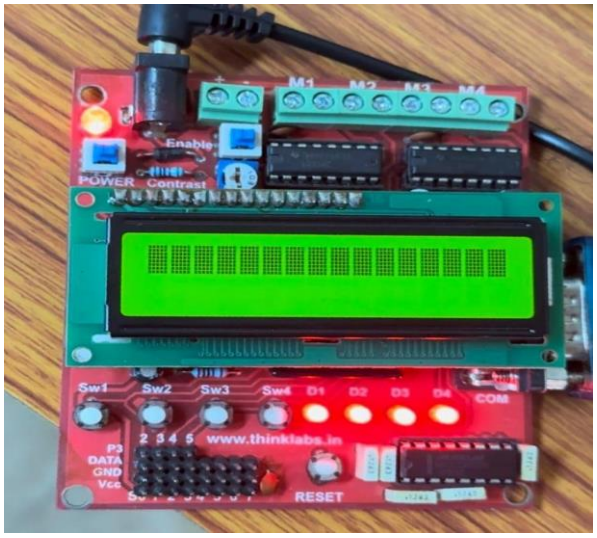
When the microcontroller is powered on, it continuously waits for the state of the switches. Depending on the switch configuration, a specific LED pattern is successfully displayed.

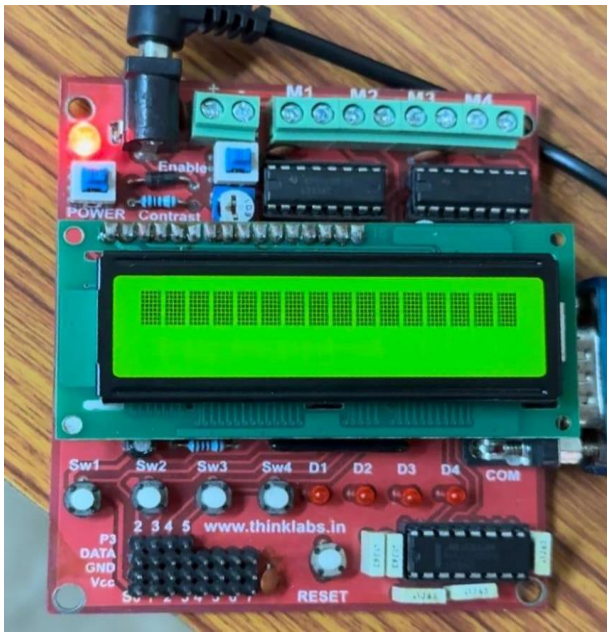



Enable
Sw1
Sw2
Sw3
Sw4
D1
D2
D3
D4
COM
P3
DATA
GND
Vcc
2 3 4 5
www.thinklabs.in
SU 1 2 3 4 5 6 7
RESET


Enable
Sw1
Sw2
Sw3
Sw4
D1
D2
D3
D4
COM
P3
DATA
GND
Vcc
2 3 4 5
www.thinklabs.in
SU 1 2 3 4 5 6 7
RESET


M1
M2
M3
M4
Enable
POWER
Contrast
Sw1
Sw2
Sw3
Sw4
D1
D2
D3
D4
COM
P3
DATA
GND
Vcc
2 3 4 5
www.thinklabs.in
RESET


M1 M2 M3 M4
Enable
POWER Contrast
Sw1 Sw2 Sw3 Sw4 D1 D2 D3 D4 COM
2 3 4 5 www.thinklabs.in
P3
DATA
GND
Vcc
RESET


M1 M2 M3 M4
Enable
POWER Contrast
Sw1 Sw2 Sw3 Sw4 D1 D2 D3 D4 COM
2 3 4 5 www.thinklabs.in
P3
DATA
GND
Vcc
RESET

Experiment 5

1. Aim

Design and implement an embedded system that interfaces an 8051 Micro-controller that displays the name and roll no on the LCD screen.

2. Hardware Used

8051 Microcontroller Kit, Computer System, USB Connector, Adapter.

3. Software Used

Keil Microvision, Flash Magic.

4. Procedure

1. Open Keil Microvision.
2. Create new project.
3. Setup all the required settings and create a C file for the source code.
4. Write the required source code in C.
5. Press F7 to build the code and generate the hex file.
6. After checking the code, we'll connect the USB cable to the Port.
7. We'll connect the Adapter to the 8051 microcontroller and power on the board.
8. Now, we'll launch Flash Magic.
9. We'll setup all the required settings.
10. After setting up everything in Flash Magic, we'll burn the hex code of the source code in the 8051 board.
11. Then, we'll power off the board and unplug the USB.
12. Now, power on the board.

6. C Code

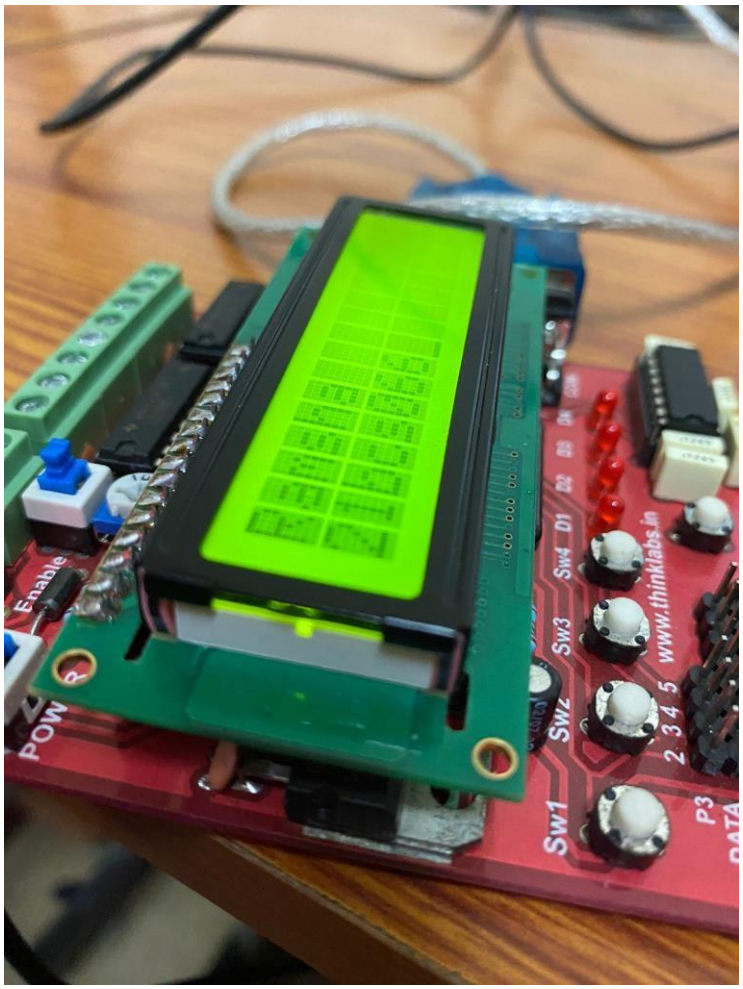
```
#include<LCD.h>
```

```
 unsigned char
x=2; unsigned char
y=0 ;

void main(void) {

LCD INIT();

while (1) {
 y=0;
x=2;

LCD_WRITE("Bushra",y,x);

DELAY (600);
 x=7;

LCD_WRITE ("Ramsha",y,x);

DELAY (600);
 y=1;
x=2;

LCD_WRITE("Rumman",y,x);

DELAY (600);

x=9; LCD_WRITE("46, 66, 67",y,x);

DELAY (1000);

}
}
```

7. Result

The C program successfully implements an embedded system that interfaces an 8051 Micro-controller to display name and roll number on LCD screen.

8. Output



Experiment 6

1. Aim

Design and implement an embedded system that interfaces an 8051 Micro-controller that takes input of a number from switches and displays its factorial on LCD screen.

2. Hardware Used

8051 Microcontroller Kit, Computer System, USB Connector, Adapter.

3. Software Used

Keil Microvision, Flash Magic.

4. Procedure

1. Open Keil Microvision.
2. Create new project.
3. Setup all the required settings and create a C file for the source code.
4. Write the required source code in C.
5. That is, configure the switches to take input. This is a binary number.
6. Corresponding decimal number is calculated and we calculate its factorial which is then displayed on the LCD screen.
7. Press F7 to build the code and generate the hex file.
8. After checking the code, we'll connect the USB cable to the Port.
9. We'll connect the Adapter to the 8051 microcontroller and power on the board. Now, we'll launch Flash Magic.
10. We will setup all the required settings.
11. After setting up everything in Flash Magic, we'll burn the hex code of the source code in the 8051 board.
12. Then, we'll power off the board and unplug the USB.
13. Now, power on the board.

6. C Code

```
#include "LCD.h"
 sbit s0
=P3^2; sbit s1
=P3^3; sbit s2
=P3^4; sbit s3
=P3^5;


int multiply(int x, int res[], int res_size);

void factorial(int n, char * ans)
{   int res[17];
int res_size = 1;
      int i,x;
res[0] = 1;
  for (x=2; x<=n; x++)
    res_size = multiply(x, res, res_size);
   for   (   i=res_size-1;   i>=0;   i--)
ans[res_size-i-1] = res[i] + '0';

      ans[res_size] = '\0';
}
int multiply(int x, int res[], int res_size)
{   int carry =
0;
      int i;
  for ( i=0; i<res_size; i++)
  {      int prod = res[i] * x +
carry;     res[i] = prod % 10;
carry = prod/10;
  }    while
(carry)   {
     res[res_size] = carry%10;
carry = carry/10;
res_size++;   }   return
res_size;
}

  void delay1(int
x){
      int i,j;
      for(i=0;i<x;i++){
            for(j=0;j<x;j++){
            }
}
}   unsigned int calFact(int
x){
```

```
      unsigned int ans;
 int i;           ans
=1;
       for(i=1;i<=x;i++){
             ans = ans * i;
       }
       return ans;
}
void calFactStr(int x, char * ans){


}
void iot(unsigned int num, char * str){
      unsigned int i,n;
int rem;
            int len;
n=num;
            len =0;
            while(n !=0){
      len++;
n = n/10;
            }
            for(i=0;i<len;i++){
            rem = num%10;
      num = num/10;
                  str[len-i-1] = rem + '0';
            }
            str[len] = '\0';
}  void main(void){
unsigned int fact;
char str[16];    char
numStr[3];
int x;
      LCD_INIT();
while(1){
                  LCD_WRITE("Enter Num:",0,2);
      delay1(500);           x = 0;
if(s3==0){                    x = x | 1;
            }
            if(s2 == 0){
      x = x| 1<<1;
            }
            if(s1 == 0){
                  x = x| 1<<2;
            }
            if(s0 == 0){
                   x = x | 1<<3;
            }

                  iot(x,numStr);
```

```
                    LCD_WRITE(numStr,0,12);
                factorial(x,str);
//iot(fact,str);
               LCD_WRITE("fact: ",1,0);
        LCD_WRITE(str,1,7);
delay1(500);              LCD_CMD(0x01);
        }


}
```

7. Result

The C program successfully implements an embedded system that interfaces an 8051 Micro-controller that takes input of a number from switches and displays its factorial on LCD screen.

8. Output