IOT DEVELOPMENT USING EMBEDDED C

INTRODUCTION TO LED INTERFACING

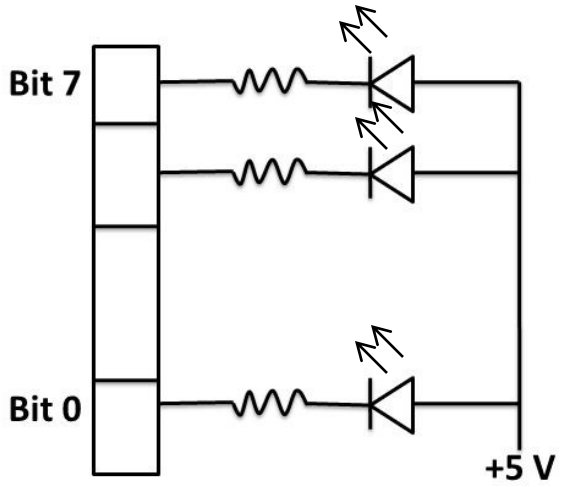
- **The LEDs are connected to pin 2 and pin 3 of port 3 of the microcontroller(P3.2 and P3.3) as shown in the figure. Inputs are given to glow the LEDs.**

- Common Anode

  [Port bit = 0
  LED Conducts
  i.e glows]

  Anodes of all LED's connected together to 5V

  Cathodes of LED's connected to port lines



Microprocessor interface to LED (common anode).

FIG 1: LED CONNECTION TO 8051 PORT

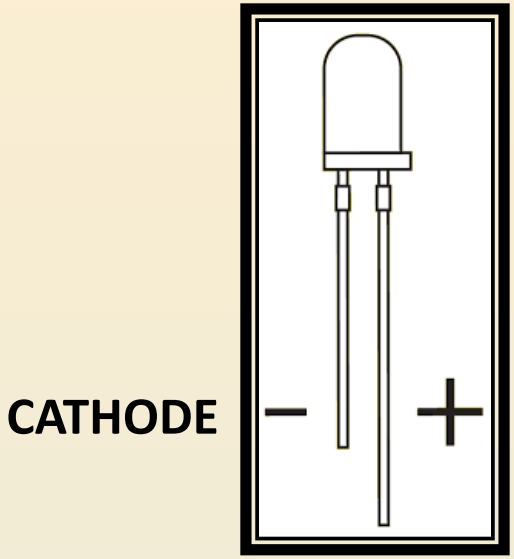


FIG 2: PHYSICAL STRUCTURE OF LED

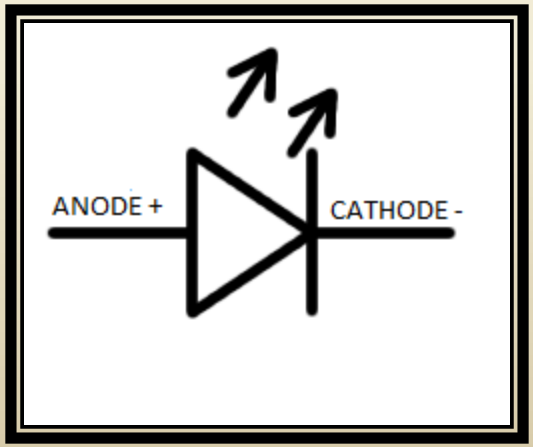


FIG 3: SYMBOL OF LED

EXP1:TO TOGGLE LED

- Input 0 and 1 are given to these pins and status of LED is observed.

```
while(1)          //FOR INFINTE LOOP
        {

          LED1=0; LED2=1;       //LED1 is ON and LED2 is OFF
          delay(75);            //calling the delay function with 75 input
          LED1=1; LED2=0;       //LED1 is OFF and LED2 is ON
          delay(75); //calling the delay function with definition in code
        }
```

INPUT (LED1)	INPUT (LED2)	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
0	1	ON	OFF
1	0	OFF	ON

- **CODE:**

experiments\P1-toggle LED\test.uvproj

DATA TYPES

TYPE	MEMORY	DECLARATION EXAMPLES
SBIT	1 BIT	sbit LED=P3^2;
INT	2 BYTES	int b=6;
CHAR	1 BYTE	char c='a';
FLOAT	4 BYTES	float d=1.6;
DOUBLE	8 BYTES	double d=0.2;
SPECIAL FUNCTION REGISTER(SFR)	1 BYTE=8 BITS	sfr j=0x80; //points to P0.0

1. **SBIT** data type used to toggle LEDs at P3.2 and P3.3

```
sbit LED1=P3^2;    //single bit data type
//value of P3.5 is given to variable LED1
sbit LED2=P3^3;
LED1=0;       LED2=1;      //LED1 is ON and LED2 is OFF
delay(150);                //calling the delay function defined in code
LED1=1;LED2=0;    //LED1 is OFF and LED2 is ON
delay(150);
```

EXAMPLES:

INPUT (LED1)	INPUT (LED2)	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
0	1	ON	OFF
1	0	OFF	ON

experiments\P2-data types\P1.uvproj

2. **INT** data type used to toggle LEDs

```
int b;
for(b=0;b<=7;b++) //for loop to give values to variable b from 0-7
            {          P3=61;  //both the LEDs on port 3 are off
                        delay(75);              //calling the delay function
                        P3=b; // binary code stored in port
                        delay(150);         //calling the delay function
            }
```

NOTE: If input in port is 61, it is stored in the binary form as follows. Pin 2 and 3 of port 3 have values 1 and hence the LEDs are off.

PORT 3

P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
0	0	1	1	1	1	0	1

experiments\P2-data types\P1.uvproj

- **EXAMPLES:**

DECIMAL NUMBER (INPUT)	BINARY NUMBER (STORED)	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
0	0000	ON	ON
1	0001	ON	ON
6	0110	OFF	ON
7	0111	OFF	ON

experiments\P2-data types\P1.uvproj

3. **CHAR** data type used to toggle LEDs at port 3

- **CAPITAL LETTERS**

```
char c;
for(c='A';c<='Z';c++)                    //for loop to give values to variable c
from 'A' to 'Z'
        {
                P3=61  //both LEDs are OFF
                delay(75);
                P3=c;    //binary equivalent of ASCII code stored in port 3
                delay(150);
        }
```

experiments\P2-data types\P1.uvproj

- **EXAMPLES:**

CHARACTER (INPUT)	ASCII CODE	BINARY CODE	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
A	65	1000001	ON	ON
D	68	1000100	OFF	ON
H	72	1001000	ON	OFF
L	76	1001100	OFF	OFF
P	80	1010000	ON	ON
T	84	1010100	OFF	ON
Z	90	1011010	ON	OFF

experiments\P2-data types\P1.uvproj

- **SMALL LETTERS**

```
char c;
for(c='a';c<='z';c++)  //for loop to give values to variable c from 'a' to 'z'
        {
                P3=61;
                delay(75);
                P3=c;   //binary equivalent of ASCII code stored in to port 3
                delay(150);
        }
```

- **EXAMPLES:**

CHARACTER	ASCII CODE	BINARY CODE	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
b	98	01100010	ON	ON
g	103	01100111	OFF	ON
k	107	01101011	ON	OFF
o	111	01101111	OFF	OFF
s	115	01110011	ON	ON
w	119	01110111	OFF	ON
z	122	01111010	ON	OFF

experiments\P2-data types\P1.uvproj

4. **FLOAT** data type used to toggle LEDs at port 3

```
float a;
for(a=0.6;a<=7.6;a++)//for loop gives values to variable a from 0.6 to 7.6
                    { P3=61;  //Both LEDs are OFF
                     delay(75);
                     P3=a;          //binary code stored in port 3
                     delay(150); }
```

- **EXAMPLES**

NUMBER (INPUT)	WHOLE NUMBER OF INPUT	BINARY NUMBER (STORED)	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
2.6	2	0010	ON	ON
3.6	3	0011	ON	ON
4.6	4	0100	OFF	ON
5.6	5	0101	OFF	ON

experiments\P2-data types\P1.uvproj

- **DOUBLE** data type used to toggle LEDs at P3.2 and P3.3

```
double d;
for(d=0.2;d<=7.2;d++)
        {
                P3=61;
                delay(75);
                P3=d;                   //binary code stored in port 3
                delay(150);
        }
```

- **EXAMPLES:**

NUMBER (INPUT)	WHOLE NUMBER OF INPUT	BINARY NUMBER (STORED)	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
1.2	1	0001	ON	ON
2.2	2	0010	ON	ON
5.2	5	0101	OFF	ON
6.2	6	0110	OFF	ON

experiments\P2-data types\P1.uvproj

- **SPECIAL FUNCTION REGISTER** data type used to assign value of P0.0 to port 3 to toggle LEDs

```
sfr j=0x80;  //points to P0.0
for(f=0;f<=61;f+=61)  //loop gives value 0 and 61
            {
              P0=f;  //port 0 is given the value 0 or 61
              P3=j; //port 3 is given value of port 0 through sfr data type
             delay(150);
            }
```

INPUT	STATUS (LED1 AT P3.2)	STATUS (LED2 AT P3.3)
0	ON	ON
61	OFF	OFF

- CODE:

experiments\P2-data types\P1.uvproj

EXP3:TO DEMONSTRATE STATUS OF LEDs USING EXPRESSIONS

- LOGICAL OPERATORS (&&,||,!)
- BOOLEAN OPERATORS (&,|,^)
- ARITHMETIC OPERATORS(+,-,*,/)
- RELATIONAL OPERATORS(<,>,<=,>=)

- **LOGICAL AND OPERATOR (&&)**-true if both operands are non zero. It takes value 1 if true and 0 if false.

```
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {
                        //LOGICAL AND OPERATOR
                        LED1=(a && b);   //LED1 assigned value of result a&&b
                        LED2=0;      //LED2 ON
                        delay(150);  //delay function called with input 150
                        LED2=61;     //LED2 OFF
                        delay(75);  //delay function called with input 75
                    }
            }
```

experiments\P3-expressions\P3.uvproj

EXAMPLES:

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	a &&b	VALUE IN LED1 (a && b)	STATUS OF LED1
0	0	FALSE	0	ON
0	1	FALSE	0	ON
1	0	FALSE	0	ON
1	1	TRUE	1	OFF

experiments\P3-expressions\P3.uvproj

- **LOGICAL OR OPERATOR(||)-true if one of the operands is non zero. It takes value 0 if false and 1 if true.**

```
for(a=0;a<=1;a++)
{
    for(b=0;b<=1;b++)
      {
          LED2=a||b;  //LED2 assigned value of result a||b
           LED1=0;
           delay(150);
           LED1=61;
           delay(75);
       }
}
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	a || b	VALUE IN LED2 (a || b)	STATUS OF LED2
0	0	FALSE	0	ON
0	1	TRUE	1	OFF
1	0	TRUE	1	OFF
1	1	TRUE	1	OFF

experiments\P3-expressions\P3.uvproj

- **LOGICAL NOT OPERATOR(!)**

```
for(a=0;a<=1;a++)  //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {
                        //LOGICAL AND OPERATOR
                        LED1=!(a && b);     //LED1 assigned value of result a&&b
                        //following code so as to observe variations in LED1 clearly
                        LED2=0;     //LED2 ON
                        delay(150);  //delay function called with input 150
                        LED2=61;     //LED2 OFF
                        delay(75);  //delay function called with input 75
                  }
          }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	!(a&&b)	VALUE IN LED2 (a || b)	STATUS OF LED2
0	0	TRUE	1	ON
0	1	TRUE	1	OFF
1	0	TRUE	1	OFF
1	1	FALSE	0	OFF

experiments\P3-expressions\P3.uvproj

• BOOLEAN AND OPERATOR(&) experiments\P3-expressions\P3.uvproj

```
//BOOLEAN AND OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {

                        LED1=a&b; //LED1 assigned value of result a&b as per truth table
                        //following code so as to observe variations in LED1 clearly
                        LED2=0;      //LED2 ON
                        delay(150);  //delay function called with input 150
                        LED2=61;     //LED2 OFF
                        delay(75);  //delay function called with input 75
                  }
         }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED1 (a & b)	STATUS OF LED1
0	0	0	ON
0	1	0	ON
1	0	0	ON
1	1	1	OFF

• BOOLEAN OR OPERATOR(|)

```
//BOOLEAN OR OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {

                        LED2=a|b;  //LED2 assigned value of result a|b as per truth table
                        LED1=0;
                        delay(150);
                        LED1=61;
                        delay(75);
                    }
            }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED1 (a | b)	STATUS OF LED1
0	0	0	ON
0	1	1	OFF
1	0	1	OFF
1	1	1	OFF

experiments\P3-expressions\P3.uvproj

• BOOLEAN XOR OPERATOR(^)

```
//BOOLEAN XOR OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {

                        LED1=a^b;  //LED1 assigned value of result a^b as per truth table
                        LED2=0;
                        delay(150);
                        LED2=61;
                        delay(75);
                    }
            }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED1 (a ^ b)	STATUS OF LED1
0	0	0	ON
0	1	1	OFF
1	0	1	OFF
1	1	0	ON

experiments\P3-expressions\P3.uvproj

- **ARITHMETIC '-' OPERATOR**
- **All values in LED1 except 0 are taken as OFF**

experiments\P3-expressions\P3.uvproj

```
//ARITHMETIC '-' OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {

                        LED1=a-b;  //all other values except 0 are taken as OFF
                        LED2=0;
                        delay(150);
                        LED2=61;
                        delay(75);
                    }
            }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED1 (a-b)	STATUS OF LED1
0	0	0	ON
0	1	-1	OFF
1	0	1	OFF
1	1	0	ON

- **ARITHMETIC ‘/’ OPERATOR**

```
//ARITHMETIC '/' OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {

                        LED2=a/b;  //all other values except 0 are taken as OFF
                        LED1=0;
                        delay(150);
                        LED1=61;
                        delay(75);
                  }
          }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED2 (a/b)	STATUS OF LED2
0	0	0/0	OFF
0	1	0	ON
1	0	1/0	OFF
1	1	1	OFF

experiments\P3-expressions\P3.uvproj

- **RELATIONAL '>' OPERATOR**

```
//RELATIONAL > OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {
                        LED1=0;
                        if(a>b)
                                LED2=b;
                        else
                                LED2=a;
                        delay(150);
                        LED1=61;
                        delay(75);
                    }
         }
```

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED2 (b if a>b)	STATUS OF LED2
0	0	0	ON
0	1	0	ON
1	0	0	ON
1	1	1	OFF

experiments\P3-expressions\P3.uvproj

• RELATIONAL '>=' OPERATOR

```
//RELATIONAL >= OPERATOR
for(a=0;a<=1;a++)   //for loop gives value 0 and 1 to 'a' variable
        {
                for(b=0;b<=1;b++)  //for loop gives value 0 and 1 to 'b' variable
                {
                        LED2=0;
                        if(a>=b)
                                LED1=a;
                        else
                                LED1=b;
                        delay(150);
                        LED2=61;
                        delay(75);
                 }
         }
```

experiments\P3-expressions\P3.uvproj

- **EXAMPLES:**

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	VALUE IN LED1 (a if a>=b)	STATUS OF LED1
0	0	0	ON
0	1	1	OFF
1	0	1	OFF
1	1	1	OFF

- CODE:

experiments\P3-expressions\P3.uvproj

EXP4:TO EXPLORE CONTROL STRUCTURES

- FOR LOOP
- WHILE LOOP
- SWITCH CASE
- IF ELSE

- **FOR LOOP**

```
//for loop
        for(a=0;a<=5;a++)
        {
                P3=61;  //both LEDs on port 3 are OFF
                delay(75);  //calling the delay function with input 75
                P3=a;  //assigning number to the port
                delay(150);  //calling the delay function with input 150
        }
```

VALUE OF VARIABLE 'a'	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
0	0000	ON	ON
1	0001	ON	ON
2	0010	ON	ON
3	0011	ON	ON
4	0100	OFF	ON
5	0101	OFF	ON

experiments\P4-loops\P4.uvproj

• WHILE LOOP

experiments\P4-loops\P4.uvproj

```
//while loop
        a=0;
        while(a<=5)     //till a is less than or equal to 5
        {
                P3=61;
                delay(75);
                P3=a;
                delay(150);
                a++;  //incrementing value of a by 1
        }
```

VALUE OF VARIABLE 'a'	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
0	0000	ON	ON
1	0001	ON	ON
2	0010	ON	ON
3	0011	ON	ON
4	0100	OFF	ON
5	0101	OFF	ON

• SWITCH CASE

```
//switch
for(b=0;b<=3;b++)   //for loop for different
cases
        {
          switch(b)
          {
                  case 0:
                   {
                          P3=61;
                           delay(75);
                          P3=b;
                          delay(150);
                   }
                   case 1:
                   {
                           P3=61;
                           delay(75);
                           P3=b;
                           delay(150);
                   }
                   case 2:
                   {
                           P3=61;
                           delay(75);
                           P3=b;
                           delay(150);
                   }
                   case 3:
                   {
                           P3=61;
                           delay(75);
                           P3=b;
                           delay(150);
                   }
                   default: P3=0;
                          }//switch ends
                  }//for loop ends
```

experiments\P4-loops\P4.uvproj

- **EXAMPLES:**

VALUE OF VARIABLE 'a'	BINARY CODE	CASE CALLED	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
0	0000	0	ON	ON
1	0001	1	ON	ON
2	0010	2	ON	ON
3	0011	3	ON	ON

experiments\P4-loops\P4.uvproj

- **IF-ELSE**

```
for(a=0;a<=1;a++)
{
        if(a==0)
        {
                P3=61;   delay(75);
                P3=a;    delay(150);
        }
        else
        {
                P3=61;    delay(75);
                P3=0;     delay(150);
        }
}
```

VALUE OF VARIABLE 'a'	VALUE PASSED TO P3	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
0	a i.e.0	ON	ON
1	0	ON	ON

- **CODE:**

[experiments\P4-loops\P4.uvproj]

P5:TO EXPLORE FUNCTIONS

- Different operations such as add, subtract, multiply, divide and recursion are performed using functions. The output is observed by observing the status of LEDs.

FUNCTION	VALUE GIVEN TO PORT 3	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
ADD()	4	0100	OFF	ON
SUBTRACT()	3	0011	ON	ON
MULTIPLY()	1	0001	ON	ON
DIVIDE()	1	0001	ON	ON
RECURSIVE()	0	0000	ON	ON

- **CODE:**

experiments\P5-functions\P5.uvproj

EXP6:TO EXPLORE ARRAYS

- Various values are stored in single dimensional and multi dimensional arrays. Then values of array is given to port 3.
- **1D ARRAY**

```
for(i=0;i<5;i++)
a[i]=i;
for(i=0;i<5;i++)
{
        P3=61;  //both LEDs OFF
        delay(75);  //calling delay() function with input 75
        P3=a[i];  //assigning values to port 3
        delay(150);  //calling delay() function with input 150
}
```

experiments\P6-arrays\p6.uvproj

- **OBSERVATIONS**

VALUE GIVEN TO PORT 3	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
a[0]=0	0000	ON	ON
a[1]=1	0001	ON	ON
a[2]=2	0010	ON	ON
a[3]=3	0011	ON	ON
a[4]=4	0100	OFF	ON

experiments\P6-arrays\p6.uvproj

• 2D ARRAY

```
for(j=0;j<2;j++)
{
        for(k=0;k<3;k++)
        {
                b[j][k]=j*k;  //assigning values to array
        }
}
for(j=0;j<2;j++)
{
        for(k=0;k<3;k++)
        {
                P3=61;
                delay(75);
                P3=b[j][k];  //assigning values to port 3
                delay(150);
        }
}
```

experiments\P6-arrays\p6.uvproj

- **OBSERVATIONS**

VALUE GIVEN TO PORT 3	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
b[0][0]=0	0000	ON	ON
b[0][1]=0	0000	ON	ON
b[0][2]=0	0000	ON	ON
b[1][0]=0	0000	ON	ON
b[1][1]=1	0001	ON	ON
b[1][2]=2	0010	ON	ON

- CODE:

experiments\P6-arrays\p6.uvproj

EXP7:TO EXPLORE POINTERS

```
void main()
{
          int i=1,j=6,k=72;  //integer variable declaration
           int *a;  //pointer variable declaration
           a=&i;  //pointer stores the address of variable i
          P3=61;  //both LEDs OFF
         delay(75); //calling delay() function with input 75
          P3=*a;  //the port is given the value of pointer
         delay(150);  //calling delay() function with input 150
         a=&j; //pointer stores the address of variable j
           P3=61;
         delay(75);
         P3=*a;
         delay(150);
         a=&k;  //pointer stores the address of variable k
          P3=61;
         delay(75);
          P3=*a;
         delay(150);
}
```

- **OBSERVATIONS:**

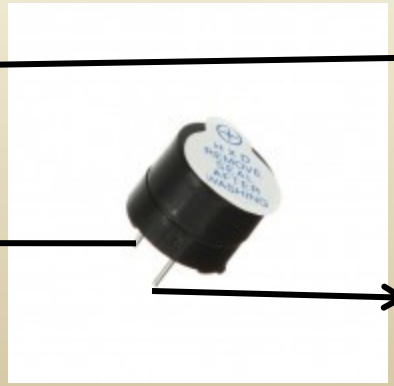
VALUE GIVEN TO PORT 3	BINARY CODE	STATUS OF LED1 (AT P3.2)	STATUS OF LED2 (AT P3.3)
1	00000001	ON	ON
6	00000110	OFF	ON
72	01001000	ON	OFF

- CODE:
- experiments\P7-pointers\P7.uvproj

INTERFACING IR SENSOR

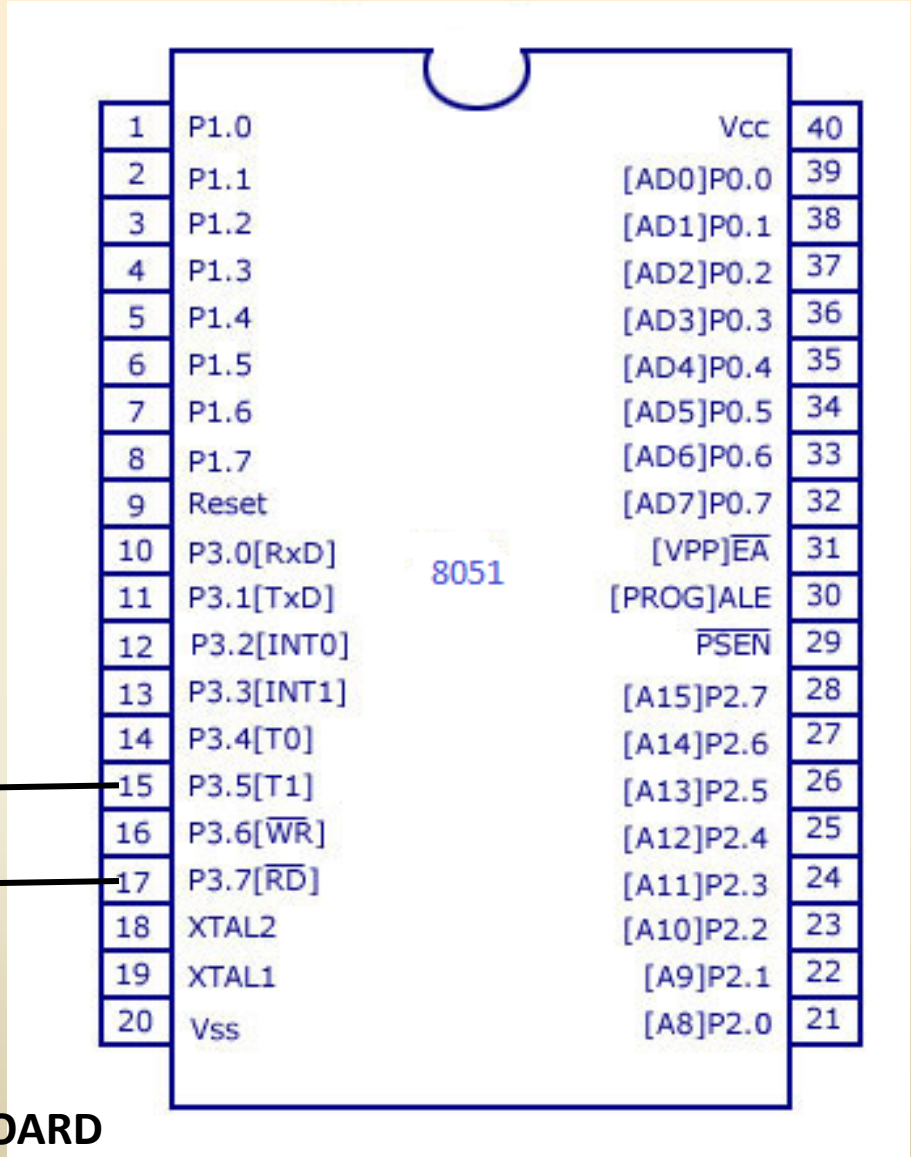
TO GND OF BOARD

TO VCC OF BOARD



TO VCC OF BOARD

BUZZER



PIN DIAGRAM

INTERFACING SENSORS

```
#include<reg51.h>  //include library to use registers defined in it
sbit sensor=P3^5;  //sensor is connected to pin 5 of port 3
sbit LED1=P3^2;  //
sbit LED2=P3^3;
sbit buzzer=P3^7;  //buzzer is connected to pin 7 of port 3
void main()
{
        if(sensor)   //if change is sensed
        {
        LED1=0;  //ON
        LED2=0;  //ON
        buzzer=0;   //buzzer sounds
        }
        else
        {
                LED1=1;  //OFF
                LED2=1;  //OFF
                buzzer=1;
        }
}
```

CODE:

experiments\P8-sensors\P8.uvproj

EXERCISES

Interface the following sensors such that when change is detected,

i. LEDs should be ON
ii. Buzzer should be ON
iii. Both LEDs and Buzzer should be ON

- Sensors:

i. IR sensor
ii. Touch sensor
iii. Clap sensor
iv. Push button

INPUT IN P3	BINARY CODE	STATUS OF LED 1 AT P3.2	STATUS OF LED2 AT P3.3
3			
5			
Q			
S			
G			
V			
l			
r			
x			
t			
6.9			
3.5			
1.0			
2.5			
59	00111011		
61	00111101		
35	00100011		
47	00101111		

VALUE OF VARIABLE 'a'	VALUE OF VARIABLE 'b'	EXPRESSION	STATUS OF LED (P3.2)
0	0	a&&b	
1	1	!a	
0	1	a||b	
0	0	!b	
1	1	a-b	
1	0	a*b	
0	0	a+b	
1	1	a/b	
0	1	if(a<=b) LED=b;	
1	0	if(a>b) LED=a;	
1	0	if(a<b) LED=a;	
1	1	if(a>=b) LED=b;	
0	0	if(a==b) LED=a;	
0	1	if(a!=b) LED=b;	
0	1	if(a!=b)	

REFERENCES

- **FIG1:**
  https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjImsis96_UAhWEtI8KHXR2ADwQjRwIBw&url=http%3A%2F%2Fslideplayer.com%2Fslide%2F3422738%2F&psig=AFQjCNHEva3UGecux_9bo3gOpq-xlofU_g&ust=1497069157431406
- **FIG 2:**

  https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiTwJTo96_UAhXERY8KHVlfBpYQjRwIBw&url=http%3A%2F%2Fmakertech.blogspot.com%2F2015%2F05%2Finterface-switch-with-8051.html&psig=AFQjCNHEva3UGecux_9bo3gOpq-xlofU_g&ust=1497069157431406
- **FIG 3:**

  https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiatu_d96_UAhUJpo8KHYosBDwQjRwIBw&url=http%3A%2F%2F8051programming.blogspot.com%2F2013%2F12%2F8051-seven-segment-display-ssd-interface.html&psig=AFQjCNHEva3UGecux_9bo3gOpq-xlofU_g&ust=1497069157431406
- **Pin diagram**

  https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiHqOfH_LLUAhWKuI8KHeaIDVsQjRwIBw&url=http%3A%2F%2Fworld4tronix.blogspot.com%2F2013%2F08%2Ffunctional-pin-diagram-of-8051.html&psig=AFQjCNHutZH7sVOumLy_BleadzcGaTWmOg&ust=1497173604441811