Experiment No : 7

Aim : Input output port programming.

Theory :

8051 microcontrollers have 4 I/O ports each of 8-bit, which can be configured as input or output. Hence, total 32 input/output pins allow the microcontroller to be connected with the peripheral devices.
•Pin configuration, i.e. the pin can be configured as 1 for input and 0 for output as per the logic state.
•Input/Output (I/O) pin– All the circuits within the microcontroller must be connected to one of its pins except P0 port because it does not have pull-up resistors built-in.
•Input pin– Logic 1 is applied to a bit of the P register. The output FE transistor is turned off and the other pin remains connected to the power supply voltage over a pull-up resistor of high resistance.
•Port 0– The P0 (zero) port is characterized by two functions –
•When the external memory is used then the lower address byte (addresses A0A7) is applied on it, else all bits of this port are configured as input/output.
•When P0 port is configured as an output then other ports consisting of pins with built-in pull-up resistor connected by its end to 5V power supply, the pins of this port have this resistor left out.
Input Configuration
If any pin of this port is configured as an input, then it acts as if it “floats”, i.e. the input has unlimited input resistance and in-determined potential.
Output Configuration
When the pin is configured as an output, then it acts as an “open drain”. By applying logic 0 to a port bit, the appropriate pin will be connected to ground (0V), and applying logic 1, the external output will keep on “floating”.
In order to apply logic 1 (5V) on this output pin, it is necessary to build an external pullup resistor.
Port 1
P1 is a true I/O port as it doesn’t have any alternative functions as in P0, but this port can be configured as general I/O only. It has a built-in pull-up resistor and is completely compatible with TTL circuits.
Port 2
P2 is similar to P0 when the external memory is used. Pins of this port occupy addresses intended for the external memory chip. This port can be used for higher address byte with addresses A8-A15. When no memory is added then this port can be used as a general input/output port similar to Port 1.
Port 3
In this port, functions are similar to other ports except that the logic 1 must be applied to appropriate bit of the P3 register.

Complete the following 8051 assembly programs in EDSIM
• 1) Assume that is a key press input is given whenever an oven is hot.
• Monitor the bit continuously.
• Whenever the oven is hot, send a high-to-low pulse to turn ON an LED.
• 2) WAP to continuously send out to port 0 values that toggle every bit.
3) Write a program to accept data at port 0 and send it to display a pattern on the LED display
4) W.A.P. to accept data at Port 1 and save it in R5,R6,R7.
5)Write a program to perform the following:
(a) Keep monitoring the keypad bit until there is a click
(b) When there is a click write value 45H to port 0
(c) Send a high-to-low (H-to-L) pulse to P2.3
6) A switch is connected to the kaypad.

Write a program to check the status of SW and perform the following:
(a) If SW=0, send letter 'N' to P2
(b) If SW=1, send letter 'Y' to P2
7) A switch is connected to pin P1.7.
Write a program to check the status of SW and perform the following:
(a) If SW=0, send letter 'N' to P2
(b) If SW=1, send letter 'Y' to P2
Use the carry flag to check the switch status.

1.

```
CLR P0.0
 KEYCHECK: JNB P0.4, PULSE
                    JMP KEYCHECK
 PULSE:SETB P1.0
                     CLR P1.0
                    JMP KEYCHECK
```

Output :

2.

```
HERE:
        CPL P0.0
        CPL P0.1
        CPL P0.2
        CPL P0.3
        CPL P0.4
        CPL P0.5
        CPL P0.6
        CPL P0.7
        JMP HERE
```

Output :

pins	bits	
0x7F	0xFF	P3
0xFF	0xFF	P2
0xFF	0xFF	P1
0x37	0x3F	P0

3.

```
MOV A, #11111110
MOV P0, A
ACCEPT: MOV A, P0
                MOV P1, A
                JMP ACCEPT
```

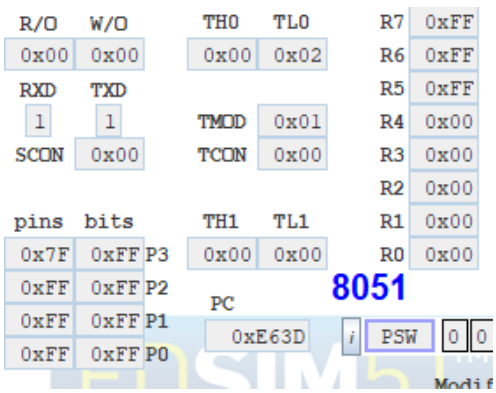
Output :

4.

```
MOV TMOD,#01
ACCEPT: MOV A, #0FFh
                MOV P1, A
                MOV R5, P1
                ACALL DELAY
                MOV A, P1
                MOV R5, A
                ACALL DELAY
                MOV A, P1
                MOV R6, A
                ACALL DELAY
                MOV A, P1
                MOV R7, A
                JMP STOP
DELAY:MOV TL0, #0B4h
;;Delay for 75 counts
              MOV TH0, #0FFh
              SETB TR0
              ACALL REPEAT
              RET
REPEAT: JNB TF0, REPEAT
                CLR TF0
                CLR TR0
                RET

STOP: NOP
```

Output :



5.

```
CLR P0.0
CHECKKEY: JNB
P0.4, PRESSED
JMP CHECKKEY
PRESSED:MOV A, #45h
        MOV P0, A

SETB P2.3

CLR P2.3
```

Output :

pins	bits	
0x7F	0xFF	P3
0xF7	0xF7	P2
0xFF	0xFF	P1
0x44	0x45	P0

6.

```
CLR P0.0
CHECKSWITCH: JNB P0.4, PRESSED
MOV A, #'Y'
MOV P2, A
JMP CHECKSWITCH
PRESSED: MOV A, #'N'
MOV P2, A

MOV CHECKSWITCH
```

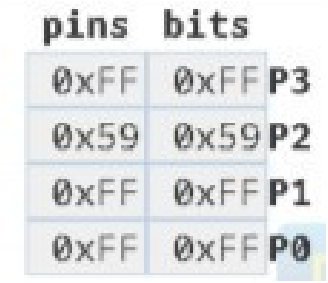
Output :

pins	bits	
0xFF	0xFF	P3
0x4E	0x4E	P2
0xFF	0xFF	P1
0xEE	0xFE	P0

7.

```
CHECKSWITCH:JB P1.7, PRESSED
MOV A, #'N'
MOV P2, A
JMP CHECKSWITCH
PRESSED: MOV A, #'Y'
MOV P2, A
JMP CHECKSWITCH
```

Output :



pins	bits	
0xFF	0xFF	P3
0x59	0x59	P2
0xFF	0xFF	P1
0xFF	0xFF	P0

Conclusion : Input/Output port programming was studied ans the codes were implemented successfully.

Deepraj Bhosale

181105016