

Lab-7

1. Write an 8051 C program to send values 00H - FFH in port P1.

```
#include <reg51.h>
void main (void)
{
    unsigned char z;
    for (z=0; z<=255; z++)
    {
        P1 = z;
    }
}
```

2. WAP in C to send hex values from ASCII characters 0, 1, 2, 3, 4, 5, A, B, C, D to port P1.

```
#include <reg51.h>
void main (void)
{
    unsigned char num[] = "012345ABCDEF";
    unsigned char z;
    for (z=0; z<=10; z++)
    {
        P1 = num[z];
    }
}
```

3. Toggle all the bits of P1 continuously.

```
#include <reg51.h>
void main(void)
{
    for( ; ; )
    {
        P1 = 0x55;
        P1 = 0xAA;
    }
}
```

4. WAP in C to send value -4 to +4 to port P1.

```
#include <reg51.h>
void main(void)
{
    signed char z;
    for(z = -4 ; z <= 4 ; z++)
    {
        MYBIT = 0; P1 = z;
        MYBIT = 1; P1 = z;
    }
}
```

5. WAP in 8051 C to toggle the bit DO of port

P1 50,000 times.

```
#include <reg51.h>
sbit MYBIT = P1^0
```

```

void main (void)
{
    unsigned int z;
    for (z = 0 ; z < 50000 ; z++)
    {
        MYBIT = 0;
        MYBIT = 1;
    }
}

```

6. WAP in 8051 C to toggle all the bits of port P1 and P2 continuously with a delay of 250 ms.

```

#include <reg51.h>
void msDelay (unsigned int);
void main (void)
{
    for ( ; ; )
    {
        P1 = 0x55;
        P2 = 0x55;
        msDelay (250);
        P1 = 0xAA;
        P2 = 0xAA;
        msDelay (250);
    }
}

```

void msDelay (unsigned int num)

```

{
    unsigned int i, j;
    for (i = 0 ; i < num ; i++)
        for (j = 0 ; j < 1275 ; j++);
}

```

Lab 8

Q.1) Write an 8051 C program to toggle all the bits of P2 continuously every 500ms delay. Using timer 1, mode 1
XTAL = 11.0592 MHz.

```
#include <reg51.h>
void delay (void);
void main (void) {
    unsigned char i;
    P2 = ~P2;
    for (i = 0; i < 7; i++)
        delay ();
}
void delay (void) {
    TMOD = 0x10;
    TL1 = 0x00;
    TH1 = 0x00;
    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}
```

Q.2) Write a C program to create a freq 2.5 kHz from P2.7

```
#include <reg51.h>
sbit MYBIT = P2^7;
void main (void) {
    TMOD = 0x20;
    TH1 = -184;
    TR1 = 1;
```

```

for(;;){
    MYBIT = ~MYBIT;
    while(TF1 == 0);
    TF1 = 0;
}

```

Q.3) Write 8051C program to toggle all the bits of P0, P1, P2 continuously with 250ms delay using SFR to declare port address, without #include <reg51.h>

```

#include <reg51.h>
sfr P0 = 0x80;
sfr P1 = 0x90;
sfr P2 = 0xA0;
void main (void){
    while(1){
        P0 = ~P0;
        P1 = ~P1;
        P2 = ~P2;
        msDelay(250);
    }
}

void msDelay() {
    unsigned char i;
    for (i = 0; i < 25; i++) {
        unsigned char j;
        for (j = 0; j < 100; j++);
    }
}

```

Lab - 8 g

Q.1) Write an ALP to generate a square wave of 2 kHz frequency on pin P1.5. You are required to use a timer in mode 1. Assume that XTAL = 11.0592 MHz.

```
#include <reg51.h>
sbit MYBIT = P1^5;
void delay (void) {
    TMOD = 0x10;
    TL1 = 0x1A;
    TH1 = 0xFF;
    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}
```

```
void main (void) {
    P1 = 0x00;
    for ( ; ; ) {
        MYBIT = ~MYBIT;
        delay ();
    }
}
```

Q.2) Two 16 bit numbers are stored in the consecutive memory space starting from location 0250H - perform a 16-bit multiplication of these 2 numbers and store the result in RAM space starting from 50H. [Note: All the numbers (multiplicand, multiplier and product) are stored in INT. memory following big endian architecture].

ORG 0000H
 MOV DPTR, #0250H
 MOV R3, #30H
 CLR A
 MOV R4, #04H
 BACK: MOV C A, @A+DPTR

MOV R3, A
 INC DPTR
 INC R3
 CJNZ R4, BACK

MOV A, 31H

MOV B, 41H

MUL AB

MOV 52H, B

MOV 53H, A

MOV A, 30H

MOV B, 40H

MUL AB

MOV 50H, B

MOV 51H, A

MOV A, 30H

MOV B, 41H

MUL AB

MOV A, 52H

MOV 52H, A

MOV A, B

ADC A, 51H
JNC SKIP
INC 50H
SKIP: MOV A, 31H
MOV B, 40H
MUL AB
ADD A, 52H
MOV 52H, A
MOV A, B
ADC A, 51H
MOV 51H, A
JNC SKIP2
INC 50H
SKIP2 : SJMP SKIP2.

Lab - 10

Q.1) A square wave is fed on pin P3.4. Design a counter to count the number of pulses in each second and display the count on port P1 and P2. (Assumptions: ① P2 and P1 will display the higher and lower byte respectively, ② the square wave have a frequency not more than $(2^{16} - 1)$, ③ XTAL = 22 MHz).
ORG 0000H

RPT: MOV TMOD, #15H

SETB P3.4

MOV TLO, #00H

MOV THO, #00H

SETB TR0

MOV R2, #28

BACK: MOV TLI, #00H

MOV THI, #00H

SETB TRI

AGAIN: JNB TFI, AGAIN

CLR TRI

CLR TFI

DJNZ R2, BACK

MOV A, TLO

MOV P1, A

MOV A, THO

MOV P2, A

SJMP RPT

END

Q.2) Write an ALP to serially transmit the character 'Y' at a Baud rate 9600.

```

ORG 0000H
LJMP START
START: MOV TMOD, #20H
        MOV TH1, #0FDH
        MOV SCON, #50H
        SETB TRI
AGAIN: MOV SBUF, #'Y'
WAIT: JNB TI, WAIT
        CLR TI
        SJMP AGAIN
END

```

*Sh
4/11/25*