

7 segment display :

```
#include <reg51.h>
```

```
#define DATAPORT P0
```

```
Sbit DISP1 = P3^4;
```

```
Void msdelay (unsigned int time);          //Function to generate delay
```

```
Int main()
```

```
{
```

```
    Unsigned char seg_code[]=
```

```
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x88,0x83,0xC6,0x  
A1,0x86,0x8E};
```

```
    Int cnt;
```

```
    While (1)
```

```
    {
```

```
        For (cnt = 0; cnt <= 9; cnt++) // loop to display 0-9999
```

```
        {
```

```
            DISP1 = 0;
```

```
            DATAPORT = seg_code[cnt];
```

```
            Msdelay(500);
```

```
        }
```

```
    }
```

```
}
```

```
//Function Definitions
```

```
Void msdelay (unsigned int time) //Function to generate delay
```

```
{
```

```
    Unsigned int i, j;
```

```
    For (i = 0; i < time; i++)
```

```
        For (j = 0; j < 113; j++); //Calibrated for a 1 ms delay in MPLAB
```

```
}
```

LCD :

```
#include <reg51.h>
```

```
#define LCD P2
```

```
Sbit RS = P3^0;
```

```
Sbit RW = P3^1;
```

```
Sbit EN = P3^2;
```

```
Void delay_ms(unsigned int ms) {
```

```
    Unsigned int i, j;
```

```
    For(i = 0; i < ms; i++)
```

```
        For(j = 0; j < 1275; j++);
```

```
}
```

```
Void lcd_cmd(unsigned char cmd) {
```

```
    LCD = cmd;
```

```
    RS = 0;
```

```
    RW = 0;
```

```
    EN = 1;
```

```
    Delay_ms(1);
```

```
    EN = 0;
```

```
    Delay_ms(2);
```

```
}
```

```
Void lcd_data(unsigned char data_byte) {
```

```
    LCD = data_byte;
```

```
    RS = 1;
```

```
    RW = 0;
```

```
    EN = 1;
```

```
    Delay_ms(1);
```

```
    EN = 0;
```

```
    Delay_ms(2);
```

```
}
```

```
Void lcd_init() {  
    Lcd_cmd(0x38);  
    Lcd_cmd(0x0C);  
    Lcd_cmd(0x01);  
    Lcd_cmd(0x06);  
    Lcd_cmd(0x80);  
}
```

```
Void lcd_string(char *str) {  
    While(*str) {  
        Lcd_data(*str++);  
    }  
}
```

```
Void main() {  
    Lcd_init();  
    Lcd_string("Roll No T171000");  
    While(1);  
}
```

LED :

```
#include <reg51.h> // Header file for 8051 microcontroller
```

```
#define LED_PORT P2 // 8 LEDs connected to Port 2
```

```
Void delay_ms(unsigned int ms) {
```

```
    Unsigned int i, j;
```

```
    For(i = 0; i < ms; i++)
```

```
        For(j = 0; j < 1275; j++); // Roughly 1 ms delay for 12 MHz crystal
```

```
}
```

```
Void main() {
```

```
    While(1) {
```

```
        // Glow even-numbered LEDs: 2, 4, 6, 8 (binary 10101010)
```

```
        LED_PORT = 0xAA; // 0xAA = 10101010b
```

```
        Delay_ms(500);
```

```
        // Glow odd-numbered LEDs: 1, 3, 5, 7 (binary 01010101)
```

```
        LED_PORT = 0x55; // 0x55 = 01010101b
```

```
        Delay_ms(500);
```

```
    }
```

```
}
```

STM_ANTICLOCKWISE :

```
#include <reg51.h>
```

```
#define DATAPORT P0
```

```
Void msdelay (unsigned int time)    // Definition of delay subroutine
```

```
{
```

```
    Unsigned int i, j;
```

```
    For (i = 0; i < time; i++)        // Loop for itime
```

```
        For (j = 0; j < 113; j++);    // Calibrated for a 1 ms delay
```

```
}
```

```
Void main()
```

```
{
```

```
    While(1)                            // Endless loop
```

```
    {
```

```
        DATAPORT = 0x09;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x08;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x0C;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x04;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x06;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x02;
```

```
        Msdelay(2);
```

```
        DATAPORT = 0x03;
```

```
Msdelay(2);  
DATAPORT = 0x01;  
Msdelay(2);
```

```
}
```

```
}
```

STM_CLOCKWISE :

#include <reg51.h>

#define DATAPORT P0

Void msdelay (unsigned int time) // Definition of delay subroutine

{

 Unsigned int i, j;

 For (i = 0; i < time; i++) // Loop for itime

 For (j = 0; j < 113; j++); // Calibrated for a 1 ms delay

}

Void main()

{

 While(1)

 // Endless loop

 {

 DATAPORT = 0x01;

 // Stepper motor 4-

step sequence

 Msdelay(2);

 DATAPORT = 0x03;

 Msdelay(2);

 DATAPORT = 0x02;

 Msdelay(2);

 DATAPORT = 0x06;

 Msdelay(2);

 DATAPORT = 0x04;

 // Stepper motor 4-

step sequence

 Msdelay(2);

 DATAPORT = 0x0C;

 Msdelay(2);

 DATAPORT = 0x08;

 Msdelay(2);

 DATAPORT = 0x09;

 Msdelay(2);

}
}