

Manas Kolaskar

Minor in Embedded Systems and IOT

Seat Number : 31011223036

MicroController 8051 programming

Semester 4

Course : BCA

Practical Journal

Edsim 51

1. Solve an equation (Example: $Y = A + B - C$)

```
MOV A, #10H ; Load A with 10H
ADD A, #05H ; Add 05H to A
SUBB A, #03H ; Subtract 03H from A
MOV 30H, A ; Store result in memory 30H
HERE: SJMP HERE ; Infinite loop
```

2. Find the GCD (Greatest Common Divisor)

```
MOV R0, #18H ; First number
MOV R1, #24H ; Second number
AGAIN:
    MOV A, R0
    SUBB A, R1
    JNC SKIP
    XCH A, R1
    SJMP AGAIN
SKIP:
    MOV R0, A
    JNZ AGAIN
    MOV 30H, R1 ; Store GCD in memory
HERE: SJMP HERE
```

3. Addition of sum of series (1+2+3+...+N)

```
MOV R0, #0 ; Sum
MOV R1, #5 ; N (example: sum of 1 to 5)
MOV A, #0
LOOP:
    ADD A, R1
    DJNZ R1, LOOP
MOV 30H, A ; Store result
HERE: SJMP HERE
```

4. Swap the contents of an 8-bit number and add them

```
MOV A, #56H ; Example number
MOV B, A
SWAP A
ADD A, B
MOV 30H, A
HERE: SJMP HERE
```

5. Find the number of 1's in a given number

```
MOV A, #0F3H ; Example: 11110011
MOV R0, #0
MOV R1, #8
COUNT:
    RRC A
    JNC SKIP
    INC R0
SKIP:
    DJNZ R1, COUNT
MOV 30H, R0 ; Store count
HERE: SJMP HERE
```

6. Transfer series to memory location

```
MOV DPTR, #40H ; Memory location
MOV R1, #10
MOV A, #1
LOOP:
    MOVX @DPTR, A
    INC A
    INC DPTR
    DJNZ R1, LOOP
HERE: SJMP HERE
```

7. Exchange numbers in memory

```
MOV A, 30H
MOV B, 31H
MOV 30H, B
MOV 31H, A
HERE: SJMP HERE
```

8. Find the greatest among a series of memory locations

```
MOV R0, #40H ; Start of series
MOV A, @R0
MOV R1, #10 ; Number of elements
NEXT:
    INC R0
    MOV B, @R0
    CJNE A, B, CHECK
    SJMP SKIP
CHECK:
    JNC SKIP
    MOV A, B
SKIP:
    DJNZ R1, NEXT
MOV 30H, A
HERE: SJMP HERE
```

Keil and Proteus

1. Generate display 0 to 9 on a Seven Segment

```
#include <REGX51.H>
void delay(int time) {
    int i, j;
    for(i=0; i<time; i++) {
        for(j=0; j<1275; j++);
    }
}
void main() {
    unsigned char numbers[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,
    0x7D, 0x07, 0x7F, 0x6F};
    while(1) {
        for(int i=0; i<10; i++) {
            P1 = numbers[i];
            delay(500);
        }
    }
}
```

2. Generate display 0 to 99 on a Seven Segment

```
#include <REGX51.H>
void delay(int time) {
    int i, j;
    for(i=0; i<time; i++) {
        for(j=0; j<1275; j++);
    }
}
void main() {
    unsigned char numbers[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,
    0x7D, 0x07, 0x7F, 0x6F};
    while(1) {
        for(int i=0; i<10; i++) {
            for(int j=0; j<10; j++) {
```

```

        P2 = numbers[i]; // Tens place
        P1 = numbers[j]; // Ones place
        delay(500);
    }
}
}
}

```

3. Generate Square Wave

```

#include <REGX51.H>
void delay() {
    int i;
    for(i=0; i<50000; i++);
}
void main() {
    while(1) {
        P1 = 0xFF; // High
        delay();
        P1 = 0x00; // Low
        delay();
    }
}

```

4. Generate Triangle Wave

```

#include <REGX51.H>
void delay(int time) {
    int i, j;
    for(i=0; i<time; i++) {
        for(j=0; j<1275; j++);
    }
}
void main() {
    int i;
    while(1) {
        for(i=0x00; i<=0xFF; i++) {
            P1 = i;
            delay(10);
        }
    }
}

```

```

    }
    for(i=0xFF; i>=0x00; i--) {
        P1 = i;
        delay(10);
    }
}
}

```

5. Generate Sawtooth Wave

```

#include <REGX51.H>
void delay(int time) {
    int i, j;
    for(i=0; i<time; i++) {
        for(j=0; j<1275; j++);
    }
}
void main() {
    int i;
    while(1) {
        for(i=0x00; i<=0xFF; i++) {
            P1 = i;
            delay(10);
        }
    }
}

```

6. Generate Sine Wave

```

#include <REGX51.H>
unsigned char sine_wave[8] = {0x7F, 0xBF, 0xDF, 0xFF, 0xDF, 0xBF, 0x7F,
0x3F};
void delay(int time) {
    int i, j;
    for(i=0; i<time; i++) {
        for(j=0; j<1275; j++);
    }
}

```

```
void main() {
    int i;
    while(1) {
        for(i=0; i<8; i++) {
            P1 = sine_wave[i];
            delay(10);
        }
    }
}
```

7. Time Delay

```
#include <REGX51.H>
void delay() {
    int i;
    for(i=0; i<50000; i++);
}
void main() {
    while(1) {
        P1 = 0xFF;
        delay();
        P1 = 0x00;
        delay();
    }
}
```

8. LED Blinking Pattern

```
#include <REGX51.H>
void delay() {
    int i;
    for(i=0; i<50000; i++);
}
void main() {
    while(1) {
        P1 = 0x55; // 01010101
        delay();
        P1 = 0xAA; // 10101010
    }
}
```



```
    delay();  
  }  
}
```

9. Four LEDs ON and Four LEDs OFF

```
#include <REGX51.H>
```

```
void delay(unsigned int time) {  
    unsigned int i, j;  
    for (i = 0; i < time; i++)  
        for (j = 0; j < 255; j++)  
            ;  
}
```

```
void main() {  
    while (1) {  
        P1 = 0xF0; // 11110000 (First 4 LEDs ON, Last 4 LEDs OFF)  
        delay(500);  
        P1 = 0x0F; // 00001111 (First 4 LEDs OFF, Last 4 LEDs ON)  
        delay(500);  
    }  
}
```

10. Two LEDs ON and Two LEDs OFF

```
#include <REGX51.H>
```

```
void delay(unsigned int time) {
```

```
    unsigned int i, j;
```

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

```
        for (j = 0; j < 255; j++);
```

```
}
```

```
void main() {
```

```
    while (1) {
```

```
        P1 = 0xCC; // 11001100 (Two ON, Two OFF, Repeat)
```

```
        delay(500);
```

```
        P1 = 0x33; // 00110011 (Opposite Pattern)
```

```
        delay(500);
```

```
    }
```

```
}
```

11. Interface 8 LEDs at the Input-Output Port & Create Different Patterns

```
#include <REGX51.H>
```

```
void delay(unsigned int time) {  
    unsigned int i, j;  
    for (i = 0; i < time; i++)  
        for (j = 0; j < 255; j++)  
            ;  
}
```

```
void main() {  
    while (1) {  
        P1 = 0xAA; // 10101010  
        delay(500);  
        P1 = 0x55; // 01010101  
        delay(500);  
        P1 = 0xF0; // 11110000  
        delay(500);  
        P1 = 0x0F; // 00001111  
        delay(500);  
    }
```

```
}  
}
```

12. Create a Binary Counter

```
#include <REGX51.H>
```

```
void delay(unsigned int time) {  
    unsigned int i, j;  
    for (i = 0; i < time; i++)  
        for (j = 0; j < 255; j++);  
}
```

```
void main() {  
    unsigned char count = 0;  
    while (1) {  
        P1 = count; // Display binary count on LEDs  
        delay(500);  
        count++;  
    }  
}
```

13. Stepper Motor Clockwise and Anti-Clockwise Rotation

```
#include <REGX51.H>
```

```
void delay(unsigned int time) {
```

```
    unsigned int i, j;
```

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

```
        for (j = 0; j < 255; j++);
```

```
}
```

```
void main() {
```

```
    while (1) {
```

```
        // Clockwise rotation sequence
```

```
        P1 = 0x09; delay(500);
```


```
        P1 = 0x0C; delay(500);
```

```
        P1 = 0x06; delay(500);
```

```
        P1 = 0x03; delay(500);
```

```
        // Anti-clockwise rotation sequence
```

```
        P1 = 0x03; delay(500);
```



```
P1 = 0x06; delay(500);  
P1 = 0x0C; delay(500);  
P1 = 0x09; delay(500);  
}  
}
```