e-PG Pathshala

Subject : Computer Science

Paper: Embedded System
Module: Programming Embedded Systems in C
Module No: CS/ES/9
Quadrant 1 – e-text

In this module, we will discuss about the embedded C programming for microcontroller. C programs for time delay and I/O operations will be discussed**.** Programs for handling the I/O ports and I/O bit manipulation will also be written and discussed.

1.Need for programming 8051 in embedded C

There are several reasons for using a high-level language such as C to program the 8051 microcontroller. It is easier and less time consuming to write programs in Embedded C than Assembly. C is easier to modify and update. We can also use code available in function libraries. Also, C code is portable to other microcontrollers with little or no modification.

1.1 Flow of Execution

We will first look at a few common structures available in Embedded C. Figure1.1 shows the flow for sequential execution , selection procedure, and looping structure.

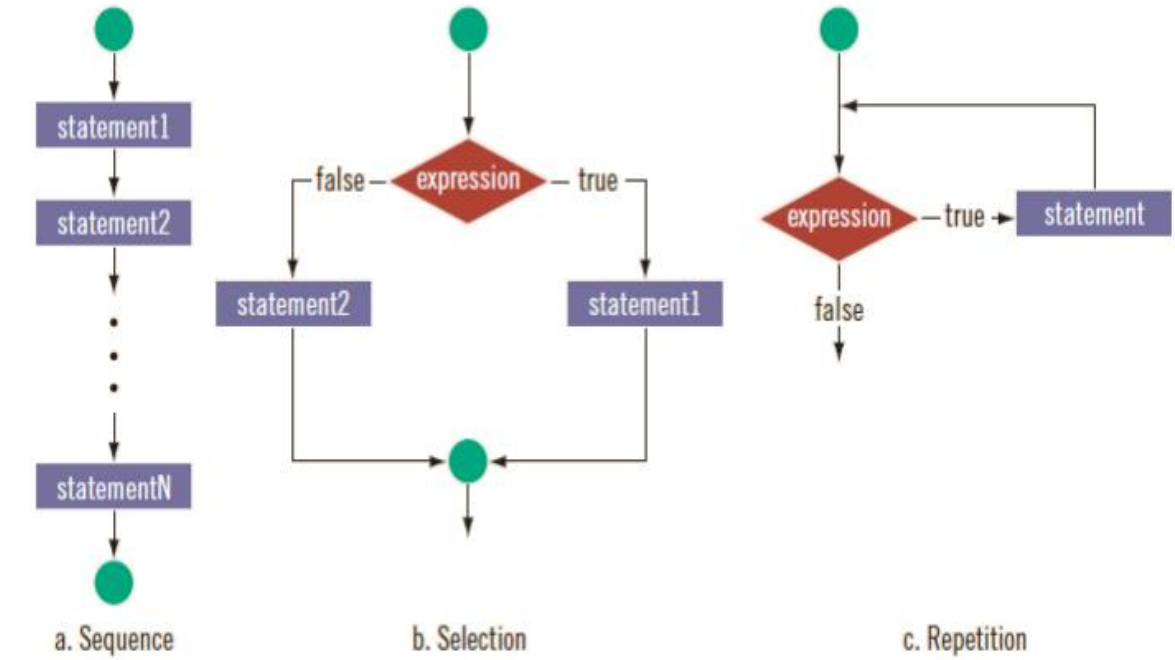
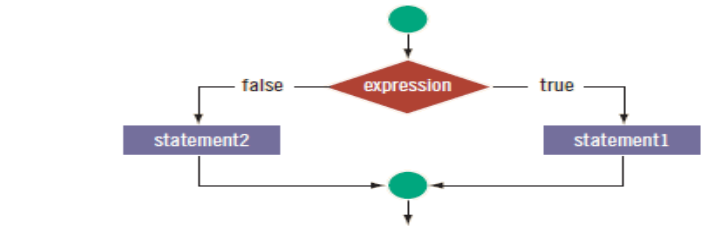


Figure 1.1 a. Sequential , b. Selection and c. Repetition structure

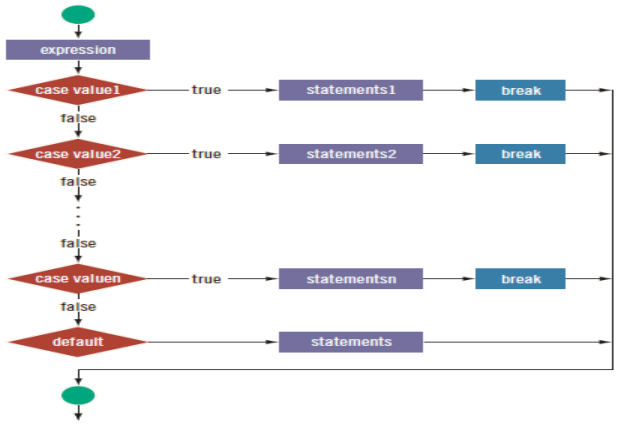
1.2 If and Switch statement(selection) structure

Fig1.2 shows the If -then-else and switch - case statement. Like normal C programming in embedded C also programmers will use if statements and switch case statements.




```
if (expression)
    statement1
else
    statement2
```

Two-way selection



switch statement

Figure 1.2 If-then else and Switch-case statement

```
switch (expression)
{
case value1:
     statements1
     break;
case value2:
     statements2
     break;
     .
     .
     .
case valuen:
     statementsn
     break;
default:
     statements
}
```

1.3 While and for looping(Repetition) structure

For looping fig 1.3 shows the structure of while loop and for looping structure .

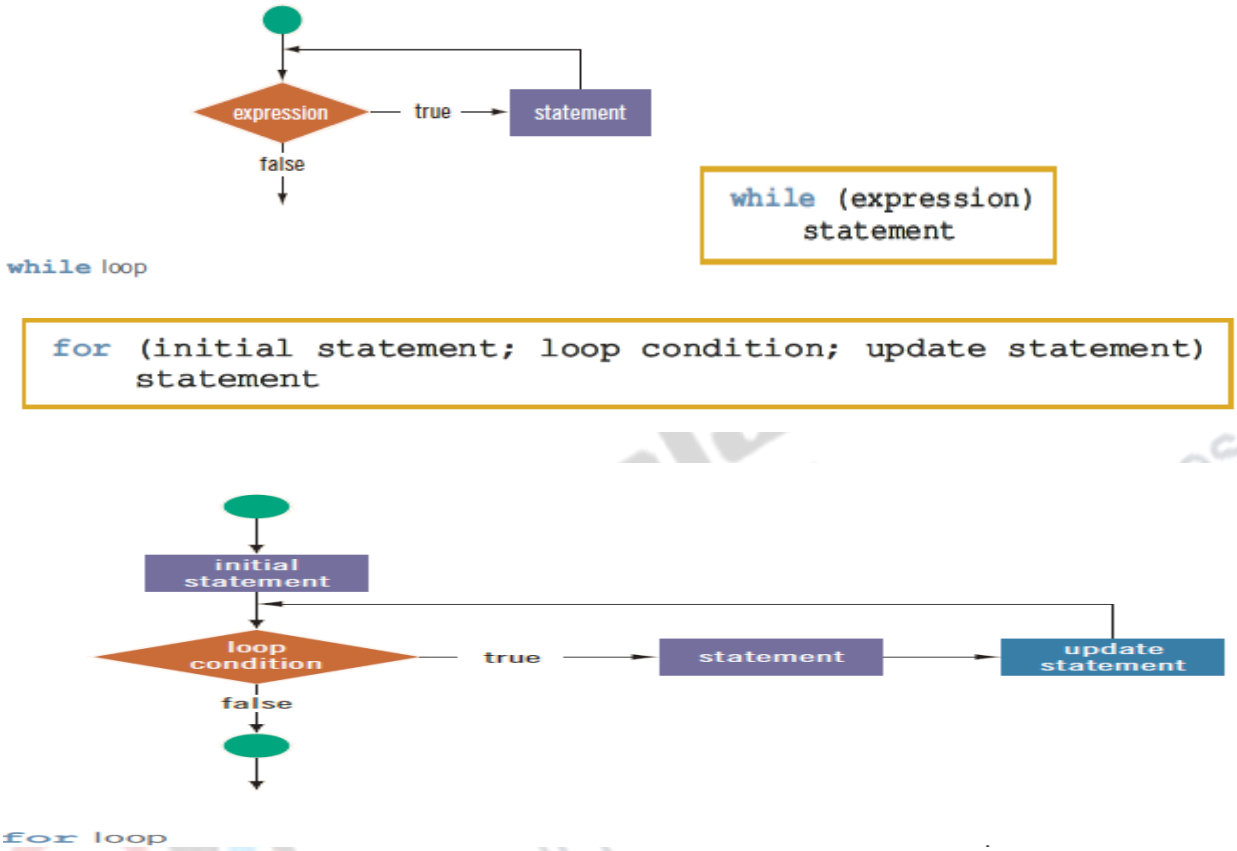


Figure 1.3 Looping Structure

2. Translation between C and Assembly Code

Following is the example on how an assembly program loop will be written in embedded "c" code

A loop in Assembly

```
        MOV R2,#255
ABC:    MOV P1,R2
        DJNZ R2,ABC
```

A for-loop in C

```
for (int z=255; z>0; z--)
P1=z;
```

3. Data Type and Time Delay in 8051 C

Specific C data types for 8051 can help programmers to create smaller hex files. They are

Unsigned char
Signed char
Unsigned int
Signed int
Sbit (single bit)
Bit and sfr

Figure 1.4 shows the number of bits needed by each data type. The character data type is the most natural data choice because 8051 is an 8-bit microcontroller. The C compiler will allocate RAM space for the variables based on their data type - char, int , bit etc.

Data Type	Size in Bits	Data Range/Usage
unsiged char	8-bit	0 to 255
(signed) char	8-bit	–128 to +127
unsigned int	16-bit	0 to 65535
(signd) int	16-bit	–32,768 to +32,767
sbit	1-bit	SFR bit-addressable only
bit	1-bit	RAM bit-addressable only
sfr	8-bit	RAM addresses 80 - FFH only

Figure 1.4 Some Widely used Data types for 8051 C

3.1 Unsigned and Signed Char

Unsigned Char:

The most widely used data type for the 8051 is unsigned char. It uses 8-bits. The range of values taken by unsigned char is 0-255(00-FFH).

In a microcontroller setup, we use unsigned char for the following:

- ✓ To set counter value
- ✓ To handle a string of ASCII characters

- ✓ For toggling ports.

Signed Char:

Signed char is an 8-bit data type. It uses a 2's complement representation. The range of unsigned char is -128 to +127 (00-FFH). It is used for the following purposes:

- ✓ To present a given quantity such as temperature which can take both positive and negative values. .

Following are examples of programs using unsigned and signed char.

Example 1 : Write an 8051 C program to send values 00 – FF to port P1.

Solution:

```
#include <reg51.h> // library code for changing the following code to 8051 assembly code
void main(void)
{
        unsigned char z;
        for (z=0;z<=255;z++)
        P1=z;
}
```

Write an 8051 C program to send hex values for ASCII characters of 0, 1, 2, 3, 4, 5, A, B, C, and D to port P1.

Solution:

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

Write an 8051 C program to send values of –4 to +4 to port P1.

Solution:

```
//Signed numbers
#include <reg51.h>
```

```
void main(void)
{
        char mynum[]={+1,-1,+2,-2,+3,-3,+4,-4};
        signed char z;
        for (z=0;z<=8;z++)
        P1=mynum[z];
}
```

4. Integer

The **unsigned int** is a 16-bit data type. It takes a value in the range of 0 to 65535 (0000 – FFFFH). It is used to

- ➢ Define 16-bit variables such as memory addresses.
- ➢ Set counter values of more than 256.

Since registers and memory accesses are in 8-bit chunks, the misuse of int variables will result in a larger hex file.

Signed int is a 16-bit data type. It uses the most significant bit D15 to represent the sign : – or + . We have 15 bits for the magnitude of the number, giving a range from –32768 to +32767.

Following code shows examples for using unsigned bit and sbit.

Write an 8051 C program to toggle bit D0 of the port P1 (P1.0) 50,000 times.

Solution:

```
#include <reg51.h>
sbit MYBIT=P1^0;
void main(void)
{
unsigned int z;
for (z=0;z<=50000;z++)
{
MYBIT=0;
MYBIT=1;
}
}
```

5. Time Delay

There are two ways to create a time delay in 8051 C : (i) using the 8051 timer and (ii) using a simple for-loop. When using a loop to create the time delay, there are three factors that can affect the accuracy of the time delay. They are: (i) Crystal frequency of the 8051 system, (ii) 8051 machine cycle timing and (iii) Compiler used for 8051 C.

Following are example programs for creating time delay:

Write an 8051 C program to toggle bits of P1 continuously forever with some delay.

Solution:

//Toggle P1 forever with some delay in between “on” and “off”

```
#include <reg51.h>
void main(void)
{
unsigned int x;
for (;;) //repeat forever
        {
        p1=0x55;
        for (x=0;x<40000;x++); //delay size unknown
        p1=0xAA;
        for (x=0;x<40000;x++);
        }
}
```

Write an 8051 C program to toggle bits of P1 ports continuously with a 250 ms delay.

Solution:

```
#include <reg51.h>
void MSDelay(unsigned int);
void main(void)
{
while (1) //repeat forever
{
p1=0x55;
MSDelay(250);
p1=0xAA;
MSDelay(250);
}
}

void MSDelay(unsigned int itime)
{
unsigned int i,j;
for (i=0;i<itime;i++)
for (j=0;j<1275;j++);
}
```

The operating modes of the 8051 can be changed by manipulating the values of the 8051's Special Function Registers (SFRs). SFRs are accessed as if they were normal Internal RAM. The only difference is that Internal RAM is from address 00h through 7Fh whereas SFR registers exist in the address range of 80h through FFh.Each SFR has an address (80h through FFh) and a name.

Write an 8051 C program to toggle all the bits of P0, P1, and P2 continuously with a 250 ms delay. Use the sfr keyword to declare the port addresses.

Solution:

```
//Accessing Ports as SFRs using sfr data type
sfr P0=0x80;
sfr P1=0x90;
sfr P2=0xA0;
void MSDelay(unsigned int);
void main(void)
{
while (1)
        {
        P0=0x55;
        P1=0x55;
        P2=0x55;
        MSDelay(250);
        P0=0xAA;
        P1=0xAA;
        P2=0xAA;
        MSDelay(250);
        }
}
```

6. I/O programming in 8051 C

The Figure 1.5 shows the address for the SFR registers. Through which you can access individual bit of SFR also.

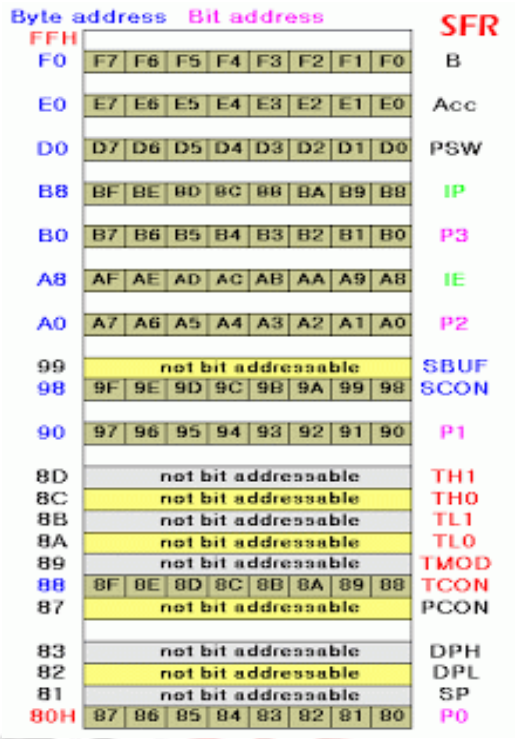


Figure 1.5 Special Function Register

Following program is an example for the use of sbit and name of SFR

#include <reg51.h>
Sbit MYBIT = P1^5; //D5 of P1
Use sbit to declare the bit of SFR and declare
Sbit MYBIT = 0x95; //D5 of P1
•reg51. h is not necessary.

Following figure 1.6 shows 8051 256 Byte RAM

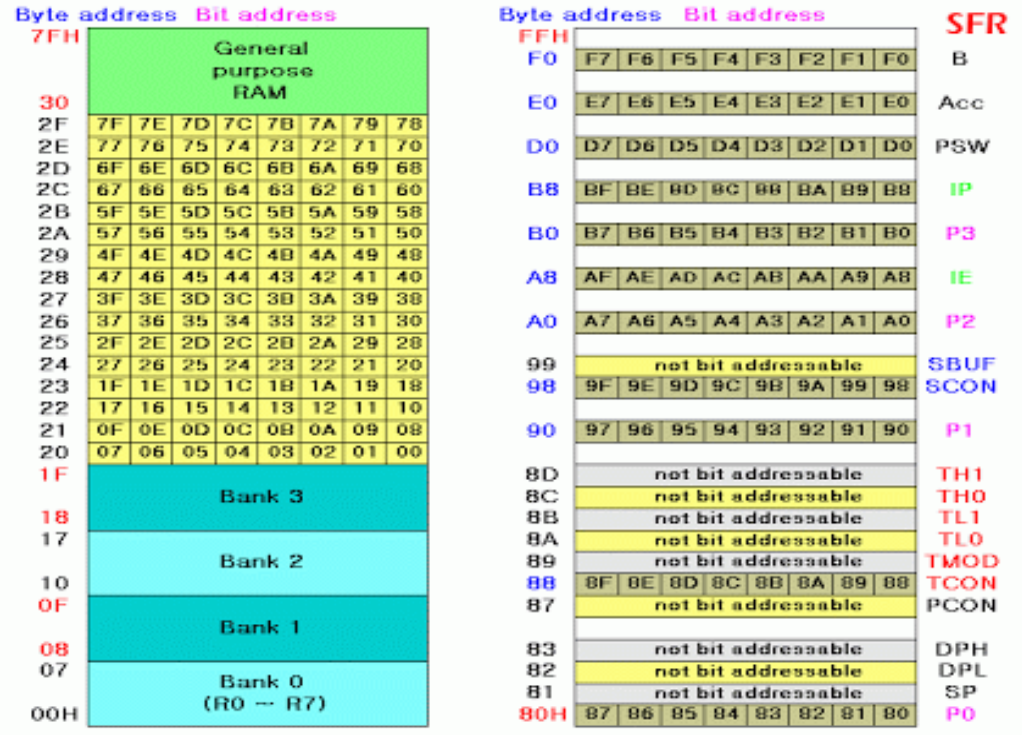


Figure 1.6 8051 256 byte RAM

Write an 8051 C program to toggle all the bits of P0, P1, and P2 continuously with a 250 ms delay. Use the sfr keyword to declare the port addresses.

Solution:

```
//Accessing Ports as SFRs using sfr data type
sfr P0=0x80;
sfr P1=0x90;
sfr P2=0xA0;
void MSDelay(unsigned int);
void main(void)
{
while (1)
{
P0=0x55;
P1=0x55;
P2=0x55;
MSDelay(250);
P0=0xAA;
P1=0xAA;
P2=0xAA;
MSDelay(250);
```

```
}
}
```

Access Single Bit of SFR

Way to access a single bit of SFR

- Use sbit and name of SFR

```
#include <reg51.h>

Sbit MYBIT = P1^5;       //D5 of P1
```

- Use sbit to declare the bit of SFR and declare by yourself

```
Sbit MYBIT = 0x95;      //D5 of P1
```

•reg51. h is not necessary.

Write an 8051 C program to turn bit P1.5 on and off 50,000 times.

Solution

```
Sbit MYBIT = 0x95;// P1^5
void main(void)
 {
unsigned intz;
for (z=0;z<50000;z++)
        {
        MYBIT=1;
        MYBIT=0;
        }
 }
```

This program is similar to Example for unsigned int.

Access Bit-addressable RAM

You can use bit to access one bit of bit-addressable section of the data RAM space 20H-2FH.

```
#include <reg51.h>
Sbit inbit= P1^0;
bit membit;//C compiler assign a RAM
                space for mybit
membit= inbit; //Read P1^0 to RAM
```

Write an 8051 C program to get the status of bit P1.0, save it, and send it to P2.7 continuously.

Solution:

```
#include <reg51.h>
sbit inbit= P1^0;
sbit outbit= P2^7;
bit membit;
void main(void)
{
while(1) { //repeat forever
membit= inbit;
outbit= membit
}
}
```

7.Summary

In this lecture we discussed C data type for 8051. Discussed C programs for time delay and I/O operations. Written C code for I/O bit and I/O port manipulations.

8.**References**

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, “The 8051 Microcontroller and Embedded Systems Using Assembly and C -Second Edition”, NewDelhi (2000).