bit、sbin、sfr、sfr 16区别分析
2009-11-12 09:08
1. bit和sbit都是C51扩展的变量类型。

bit和int char之类的差不多，只不过char=8位, bit=1位而已。都是变量，编译器在编译过程中分配地址。除非你指定，否则这个地址是随机的。这个地址是整个可寻址空间，RAM+FLASH+扩展空间。bit只有0和1两种值，意义有点像Windows下VC中的BOOL。

sbit是对应可位寻址空间的一个位，可位寻址区：20H～2FH。一旦用了sbi xxx = REGE^6这样的定义，这个sbit量就确定地址了。sbit大部分是用在寄存器中的，方便对寄存器的某位进行操作的。

2. bit位标量
bit位标量是C51编译器的一种扩充数据类型，利用它可定义一个位标量，但不能定义位指针，也不能定义位数组。它的值是一个二进制位，不是0就是1，类似一些高级语言中的Boolean类型中的True和False。

3. sfr特殊功能寄存器
sfr也是一种扩充数据类型，点用一个内存单元，值域为0～255。利用它可以访问51单片机内部的所有特殊功能寄存器。如用sfr P1 = 0x90这一句定P1为P1端口在片内的寄存器，在后面的语句中我们用以用P1 = 255（对P1端口的所有引脚置高电平）之类的语句来操作特殊功能寄存器。

sfr P1 = 0x90; //定义P1 I/O 口,其地址90H
sfr 关键定后面是一个要定义的名字,可任意选取,但要符合标识符的命名规则,名字最好有一定的含义如P1 口可以用P1 为名,这样程序会变的好读好多.等号后面必须是常数,不允许有带运算符的表达式,而且该常数必须在特殊功能寄存器的地址范围之内(80H-FFH),具体可查看附录中的相关表.

sfr 是定义8 位的特殊功能寄存器而sfr16 则是用来定义16 位特殊功能寄存器,

如8052 的T2 定时器,可以定义为:
sfr16 T2 = 0xCC; //这里定义8052 定时器2,地址为T2L=CCH,T2H=CDH
用sfr16 定义16 位特殊功能寄存器时,等号后面是它的低位地址,高位地址一定要位于物
理低位地址之上.注意的是不能用于定时器0 和1 的定义.
sbit 可定义可位寻址对象.如访问特殊功能寄存器中的某位.其实这样应用是经常要
用的如要访问P1 口中的第2 个引脚P1.1.我们可以照以下的方法去定义:
(1) sbit 位变量名=位地址
sbit P1_1 = 0x91;
这样是把位的绝对地址赋给位变量.同sfr 一样sbit 的位地址必须位于

80H-FFH 之间.
(2) sbit 位变量名=特殊功能寄存器名^位位置
sft P1 = 0x90;
sbit P1_1 = P1 ^ 1; //先定义一个特殊功能寄存器名再指定位变量名所在的位置,当可
寻址位位于特殊功能寄存器中时可采用这种方法
(3) sbit 位变量名=字节地址^位位置
sbit P1_1 = 0x90 ^ 1;
这种方法其实和 2 是一样的,只是把特殊功能寄存器的位址直接用常数表示.
在 C51
存储器类型中提供有一个 bdata 的存储器类型,这个是指可位寻址的数据存储器,位于单
片机的可位寻址区中,可以将要求可位录址的数据定义为 bdata,如:
unsigned char bdata ib; //在可位录址区定义 ucsigned char 类型的变量 ib
int bdata ab[2]; //在可位寻址区定义数组 ab[2],这些也称为可寻址位对象
sbit ib7=ib^7 //用关键字 sbit 定义位变量来独立访问可寻址位对象的其中一位
sbit ab12=ab[1]^12;
操作符"^"后面的位位置的最大值取决于指定的基址类
型,char0-7,int0-15,long0-31.

sfr 并标准 C 语言的关键字，而是 Keil 为能直接访问 80C51 中的 SFR 而提供了一个新
的关键词，其用法是：
sfrt 变量名=地址值。
2）符号 P1_0 来表示 P1.0 引脚。
在 C 语言里，如果直接写 P1.0，C 编译器并不能识别，而且 P1.0 也不是一个合法的 C
语言变量名，所以得给它另起一个名字，这里起的名为 P1_0，可是 P1_0 是不是就是 P1.0
呢？你这么认为，C 编译器可不这么认为，所以必须给它们建立联系，这里使用了 Keil C
的关键字 sbit 来定义，sbit 的用法有三种：
第一种方法：sbit 位变量名＝地址值
第二种方法：sbit 位变量名＝SFR 名称^变量位地址值
第三种方法：sbit 位变量名＝SFR 地址值^变量位地址值
如定义 PSW 中的 OV 可以用以下三种方法：
sbit OV=0xd2 （1）说明：0xd2 是 OV 的位地址值
sbit OV=PSW^2 （2）说明：其中 PSW 必须先用 sfr 定义好
sbit OV=0xD0^2 （3）说明：0xD0 就是 PSW 的地址值
因此这里用 sfr P1_0=P1^0;就是定义用符号 P1_0 来表示 P1.0 引脚，如果你愿意也可以
起 P10 一类的名字，只要下面程序中也随之更改就行了。

＊AT89C51 的特殊功能寄存器表请看附录二

4. sfr16 16 位特殊功能寄存器

sfr16 占用两个内存单元，值域为 0～65535。sfr16 和 sfr 一样用于操作特殊功能寄存器，所不同的是它用于操作占两个字节的寄存器，好定时器 T0 和 T1。

5. sbit 可录址位

sbit 同位是 C51 中的一种扩充数据类型，利用它可以访问芯片内部的 RAM 中的可寻址位或特殊功能寄存器中的可寻址位。如先前我们定义了
sfr P1 = 0x90; //因 P1 端口的寄存器是可位寻址的，所以我们可以定义
sbit P1_1 = P1^1; //P1_1 为 P1 中的 P1.1 引脚
//同样我们可以用 P1.1 的地址去写,如 sbit P1_1 = 0x91;
这样我们在以后的程序语句中就可以用 P1_1 来对 P1.1 引脚进行读写操作了。通常这些可以直接使用系统提供的预处理文件，里面已定义好各特殊功能寄存器的简单名字，直接引用可以省去一点时间，我自己是一直用的。当然您也可以自己写自己的定义文件，用您认为好记的名字。

data 表明数据在片内数据存储区；
xdata 表明数据在片外数据存储区；
code 表明数据在程序存储区；
extern 定义的数据是在另外一个模块，当引用其它文件中的变量时要加上 extern。extern 的重要意义在于表明要定义的数据已经在其他地方定义过，此处只是引用，所以编译器不会另外开辟内存。