

一个用作标志的普通变量至少要用一个字节—8 位,而在某些计算机系统中则可能是 16 位。如果我们想在一个很大的表中存储很多标志,那么 “被浪费”的内存空间是很可观的。在 C 语言中,一种方法是用叫做位段的构造类型来定义一个压缩信息的结构。

我们通常想保持某个状态的Flag,就可以只用一个变量的某一个bit位来保存,尤其是做像modbus等通讯时,如果位变量较多;需要整合到一个字来传送;这样既可以节省空间又可以节省通讯数据量!

下面将对变量的位操作和位状态获取的方法。

位操作：

- 置位

```
#define BitSet(Number,pos) ((Number) |= 1<<(pos))  
//把变量Number的第pos位置1
```

- 复位

```
#define BitClr(Number,pos) (Number) &= ~(1<<(pos))  
//把变量Number的第pos位清零
```

- 取反

```
#define BitINV(Number,pos) ((Number) ^= 1<<(pos))  
//把变量Number的第pos位取反
```

位获取：

```
#define BitGet(Number,pos) ((Number) >> (pos)&1)  
//用宏得到变量Number的第pos位
```