

## 第七十三节： 结构体数据的传输存储和还原。

### 【73.1 结构体数据的传输存储和还原。】

结构体本质是一个数组，数组内可能包含了许多不同数据长度类型的成员，当整个结构体数据需要存储或者传输（通信）给另外一个单片机时，这时候有两种选择，一种是一个成员一个成员的挨个处理，这种“以成员为单位”的处理方式比较繁琐麻烦，另外一种是把整个结构体变量当作一个“以字节为单位”的普通数组来处理，但是有两个关键的问题来了，第一个问题是如何把结构体“拆分”成“以字节为单位”来进行搬运数据，第二个问题是假如我们把整个结构体数据以“字节为单位”的方式“整体打包”传递给另外一个单片机，当这个接收方的单片机接收到我们这一组数据后，如何把这“一包”以字节为单位的数组再“还原”成相同的结构体变量，以便在程序处理中也能直接按“结构体的方式”来处理某个具体的成员。其实，这两个问题都涉及到“指针的强制转换”。具体讲解的例子，请直接阅读下面 73.2 段落的源代码例子和注释。

### 【73.2 例程练习和分析。】

现在编写一个练习程序，把一个结构体变量“以字节的方式”存储到另外一个普通数组里，然后再把这个“以字节为单位”的普通数组“还原”成“结构体的方式”，以便直接操作内部某个具体的成员。

```
/*---C 语言学习区域的开始。-----*/

struct StructMould_1
{
    unsigned char  u8Data_A;
    unsigned long  u32Data_B;
    unsigned int   u16Data_C;
};

struct StructMould_1  GtMould_1;  // “生成” 一个变量。

unsigned char  Gu8Buffer[sizeof(GtMould_1)]; //定义一个内存跟结构体变量大小一样的普通数组
unsigned char *pu8;    //定义一个把结构体变量“拆分”成“以字节为单位”的指针
struct StructMould_1  *ptStruct; //定义一个结构体指针，用于“还原”普通数组为“结构体”
unsigned int i;        //定义一个用于 for 循环的变量

void main() //主函数
{
    //先把该结构体变量内部具体成员分别以“成员的方式”初始化为 5, 6, 7
    GtMould_1.u8Data_A=5;
    GtMould_1.u32Data_B=6;
    GtMould_1.u16Data_C=7;

    pu8=(unsigned char *)&GtMould_1;    //把结构体变量强制转换成“以字节为单位”的指针
    for(i=0;i<sizeof(GtMould_1);i++)
    {
```

```

        Gu8Buffer[i]=pu8[i];    //把结构体变量以字节的方式搬运并且存储到普通数组里。
    }

    ptStruct=(struct StructMould_1 *)&Gu8Buffer[0]; //再把普通数组强制“还原”成结构体指针
    ptStruct->u8Data_A=ptStruct->u8Data_A+1;    //该变量从 5 自加 1 后变成 6。
    ptStruct->u32Data_B=ptStruct->u32Data_B+1; //该变量从 6 自加 1 后变成 7。
    ptStruct->u16Data_C=ptStruct->u16Data_C+1; //该变量从 7 自加 1 后变成 8。

    View(ptStruct->u8Data_A); //在电脑端观察结构体成员 u8Data_A 的数值。
    View(ptStruct->u32Data_B); //在电脑端观察结构体成员 u32Data_B 的数值。
    View(ptStruct->u16Data_C); //在电脑端观察结构体成员 u16Data_C 的数值。

    while(1)
    {
    }
}
/*---C 语言学习区域的结束。-----*/

```

在电脑串口助手软件上观察到的程序执行现象如下：

开始...

第 1 个数

十进制:6

十六进制:6

二进制:110

第 2 个数

十进制:7

十六进制:7

二进制:111

第 3 个数

十进制:8

十六进制:8

二进制:1000

分析：

结构体成员 u8Data\_A 的数值是 6。

结构体成员 u32Data\_B 的数值是 7。

结构体成员 u16Data\_C 的数值是 8。

### 【73.3 如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。