

嵌入式C语言之- 结构体类型的规则及使用

讲师：叶大鹏

助力你成为优秀的电子工程师！



应用案例

- 在之前代码例程基础上，温度传感器还具备检测输出湿度数据功能，现在需要管理传感器的数据包包括温度、湿度、ID(SN):

- 采用基础数据类型的方案:

```
float temp = 20.5f;  
uint8_t humi = 40;  
uint32_t id = 0x1234;
```

- 如果要向其他接口函数传递这些数据，需要定义3个参数Display(id, temp, humi);
- 如果要管理3个点位的传感器数据，则需要定义3个数组，此时还需要在大脑中维护3个传感器对应这些数组关系的表格;

```
float temp[3];  
uint8_t humi[3];  
uint32_t id[3];
```

数据类型声明

- 结构体类型和枚举类型一样，是一种用户自定义的数据类型，它可以使用一个变量来描述事物的多种属性，便于数据的管理，数据类型声明的语法格式：

struct 结构体名称

{

成员列表

};

struct TempHumiSensor

{

uint32_t id;

float temp;

uint8_t humi;

};

struct StudentInfo

{

char name[10];

uint8_t id;

char sex[10];

uint8_t age;

float score;

};

成员列表里的成员并不是变量，
在数据类型声明时，不会分配内存空间。

变量定义

- 结构体类型变量定义几种常见形式:

- `struct TempHumiSensor tempHumiData;`

- `struct TempHumiSensor`

- `{`

- `uint32_t id;`

- `float temp;`

- `uint8_t humi;`

- `} tempHumiData1, tempHumiData2;`

变量定义

- 结构体类型变量定义几种常见形式:

```
• typedef struct TempHumiSensor  
{  
    uint32_t id;  
    float temp;  
    uint8_t humi;  
} TempHumiSensor;  
TempHumiSensor tempHumiData;
```

只有在定义结构体变量时，才会为成员列表里的成员分配内存空间。

变量初始化

- 结构体变量初始化类似于数组初始化，要注意初始化时数据类型的匹配：

```
typedef struct
```

```
{
```

```
    char name[10];
```

```
    uint8_t id;
```

```
    char sex[10];
```

```
    uint8_t age;
```

```
    float score;
```

```
} StudentInfo;
```

```
StudentInfo stuInfo = {"令狐冲", 1, "男", 25, 95.5};
```

变量使用，访问成员

- C语言规定了两种运算符可用于访问结构体成员：一种是成员运算符，也称为圆点运算符(.); 另一种是配合指针使用的指向运算符，也称为箭头运算符(->):

结构体变量.成员名

- typedef struct

{

uint32_t id;

float temp;

uint8_t humi;

} TempHumiSensor;

TempHumiSensor tempHumiData;

tempHumiData.temp = 20.5f;

变量使用，整体赋值

- typedef struct

```
{  
    uint32_t id;  
    float temp;  
    uint8_t humi;  
} TempHumiSensor;
```

```
TempHumiSensor tempHumiData1, tempHumiData2;
```

```
tempHumiData1.temp = 20.5f;
```

```
tempHumiData1.humi = 50;
```

```
tempHumiData1.id = 0x1234;
```

```
tempHumiData2 = tempHumiData1;
```

数组不支持这种赋值方式

THANK YOU!