

---

title: 单片机C语言知识用法之#define top: false cover: false toc: true  
mathjax: false date: 2019-08-31 00:53:41 password: summary: tags: C语言  
categories: C语言

## #define的定义:

---

#define是C语言中的一个预处理指令，其中的“#”表示这是一条预处理命令。凡是以“#”开头的均为预处理命令，“define”为宏定义命令，“标识符”为所定义的宏名。

```
#define TIME_NUM 1000 //定义一个时间数
```

一个简单的TIME\_NUM 就定义好了，它代表1000，如果在程序里面写:

```
if(i<TIME_NUM )  
{  
    ....  
    ....  
}
```

编译器在处理这个代码之前会对TIME\_NUM 进行处理替换为1000。这样的定义看起来类似于普通的常量定义CONST，但也有着不同，因为define的定义更像是简单的文本替换，而不是作为一个量来使用。

## #define的语法:

---

### 语法一:

```
#define      标识符      被标识符代表的字符串      //这种用法很简单,就是一种替换.
```

### 语法二:

```
#define      标识符[(参数1,.....,参数n)]      被标识符代表的字符串      //其中,在  
"被标识符代表的字符串" 中出现的形参将在使用时被实参替代. 就像写函数一样.
```

例如:

```
#define      ADD_NUM      10
```

也可以这样子用:

```
#define ADD(x,y) (x+y)
```

这个定义就将返回两个数中较大的那个，看到了吗？因为这个“函数”没有类型检查，就好像一个函数模板似的，当然，它绝对没有模板那么安全就是了。可以作为一个简单的模板来使用而已。但是这样做的话存在隐患，例子如下：

```
#define ADD(a,b) a+b
```

在一般使用的时候是没有问题的，但是如果遇到如： $c * \text{Add}(a,b) * d$ 的时候就会出现问題，代数式的本意是 $a+b$ 然后去和 $c, d$ 相乘，但是因为使用了`define`（它只是一个简单的替换），所以式子实际上变成了  $ca + bd$  所以，用`#define`要注意顺序

一般我个人用`#define`在单片机程序上的话，我一般只做简单的替换。

```
#define TIME_NUM (60*60*24)UL//定义一个一天时间有多少秒
```

另外举一个例子：

```
#define pin (int*);  
pin a,b;
```

本意是`a`和`b`都是`int`型指针，但是实际上变成`int* a,b`;`a`是`int`型指针，而`b`是`int`型变量。这是应该使用`typedef`来代替`define`，这样`a`和`b`就都是`int`型指针了。所以我们在定义的时候，养成一个良好的习惯，建议所有的层次都要加括号。

而且，宏在单片机代码中用的很多，常数的替换、地址的偏移，等等都用得上 用宏来修改移植代码更加便捷，代码更容易使人读懂。。。。

## 喜欢就关注我吧！

---



相关代码可以在公众号后台获取。