

嵌入式C语言之- 预处理指令#define

讲师：叶大鹏

助力你成为优秀的电子工程师！



预处理指令#define

- #define叫做宏定义，语法格式为：

#define 名字 值

#define PI 3.14159

1. 结尾没有分号；，这一点和#include以及所有预处理指令一样，都不是C语句；
2. 值可以是数字、表达式、代码语句等；
3. 和#include一样，在预处理阶段执行，文本替换；

- ✓ **宏定义的好处是，便于程序的阅读和维护：**

- 使用宏定义可以用宏的 名字 代替一个在程序中经常使用的 值，这样，当需要改变这个值时，就不需要对整个程序一个一个进行修改，只需修改宏定义中的值即可；

#define TEM_MAX_VALUE 55.0f

- 当值比较长时，使用宏定义就可以用较短的名字来代替它，这样方便程序的编写；

#define MIN(a, b) (((a) < (b)) ? (a) : (b))

带参数的宏定义

- `#define MIN(a, b) (((a) < (b)) ? (a) : (b))`

1. 和函数不同，宏的参数没有数据类型，因为它是文本展开；
2. 因为是文本展开，所以相比函数没有执行调度的开销，效率要高；
3. 但是也同样是因为文本展开，所以在展开时可能会出现意想不到的情况。

带参数的宏定义

- 一切都要 () :

1. 整个值的表达式都要 () ;
2. 参数出现的每个地方都要 () 。

```
#define CEL_TO_FAH(t) ((t) * 9.0f / 5.0f + 32)
```

没有值的宏定义

- **#define 名字**

1. 这种宏定义，是用于条件编译的，配合其他的预处理指令来检查这个宏是否被定义过：

```
#ifndef _TEM_H_
```

```
#define _TEM_H_
```

```
float GetCelTem(void);
```

```
#endif
```

#include预处理指令

- #include是一个预处理指令，预处理这个动作发生在编译之前：



- #include的作用是，在预处理时，将文件中的全部文本内容全部复制粘贴到#include所在的位置；

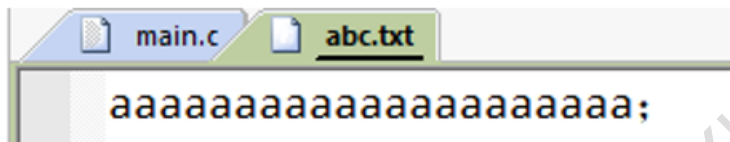
```
1 float GetCelTem(void);
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "ARMCM4.h"
4 #include "tem.h"
```

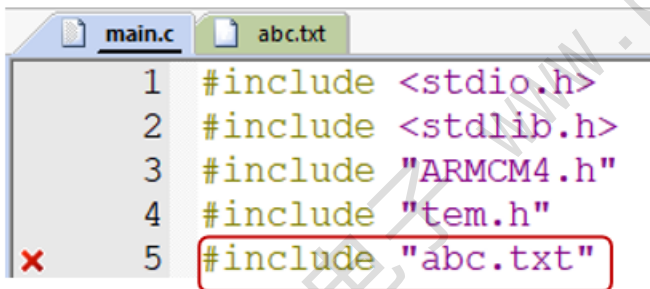


```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "ARMCM4.h"
4 float GetCelTem(void);
```

#include预处理指令



```
main.c  abc.txt
aaaaaaaaaaaaaaaaaaaaaa;
```



```
main.c  abc.txt
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "ARMCM4.h"
4  #include "tem.h"
x 5  #include "abc.txt"
```

compiling main.c...

abc.txt(1): error: #7: unrecognized token
aaaaaaaaaaaaaaaaaaaaaa;

#include预处理指令

- <>和""有什么区别?

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "tem.h"
```

- 使用尖括号< >, 编译器会到标准库路径下查找头文件;
- 使用双引号" ", 编译器首先在当前目录下查找头文件, 如果没有找到, 再到标准库路径下查找。

```
#include <stdio.h>
#include <stdlib.h>
#include "tem.h"
```

```
#include "stdio.h" ?
#include <stdlib.h>
#include "tem.h"
```

```
#include "stdio.h"
#include <stdlib.h>
#include <tem.h> ?
```


#include预处理指令

```
#include <stdio.h>
#include <stdlib.h>
#include "tem.h"
```

```
#include "stdio.h" ✓
#include <stdlib.h>
#include "tem.h"
```

```
#include "stdio.h"
#include <stdlib.h>
#include <tem.h> ✓
```

规范用法：

- 标准库的头文件使用尖括号< >;
- 自定义的头文件使用双引号" "。

#include预处理指令

名称	修改日期	类型	大小
Listings	2023/1/12 16:39	文件夹	
Objects	2023/1/16 17:04	文件夹	
RTE	2023/1/16 16:59	文件夹	
tem			
abc.txt			1 KB
all.dis			31 KB
app.h	2023/1/12 16:55	C Header 源文件	1 KB
EventRecorderStub.scvd	2023/1/16 15:51	SCVD 文件	1 KB
main.c	2023/1/16 17:04	C 源文件	1 KB
tem.c	2023/1/16 16:59	C 源文件	1 KB
template.uvguix.Administrator	2023/1/16 17:04	ADMINISTRATO...	176 KB
template.uvguix.yedap	2022/12/29 15:00	YEDAP 文件	177 KB
template.uvoptx	2023/1/16 16:41	UVOPTX 文件	8 KB
template.uvprojx	2023/1/16 16:41	碘ision5 Project	17 KB

将tem.h放到新建的tem文件夹下

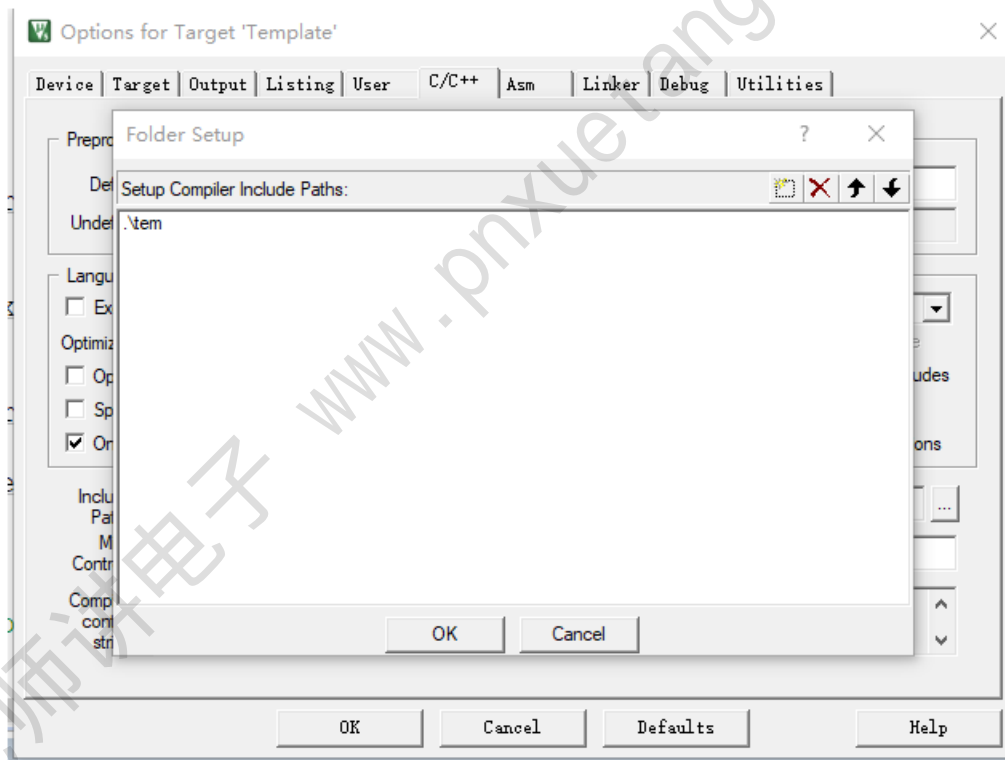
```
#include <stdio.h>
#include <stdlib.h>
#include "tem/tem.h"
```

#include预处理指令

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "tem.h"
```



还有哪些预处理指令

- 宏定义指令：

#define、#undef

- 条件编译指令：

#if、#else、#elif、#endif、#ifdef、#ifndef

- 其它指令：

#error、#line、#pragma

THANK YOU!