

Error: L6200E

Error: L6200E: Symbol temp multiply defined (by wenshidu.o and main.o). 在编译的时候出现了这个问题，但是检查不出来，希望各位大侠帮帮忙

什么变量你给付了两次值

你看看是不是那个外部变量你又给赋值了

main.c 申明，其他.c 文件对应的.h 文件中用 `extern` 引用

error: #20

error: #20: identifier "TIM2_IRQChannel" is undefined 谁能说说，哪里错了

你的固件库里的库文件没有添加进工程里面，所以出现未定义的情况。

`TIM2_IRQChannel` 指定时器 2 的中断通道没有定义，其实在固件库对这些参数都有定义，宏定义代替了一串寄存器地址数据。需要将.C 文件添加到工程文件中

warning: #1-D

main.c(7): warning: #1-D: last line of file ends without a newline

当使用 keil 编译时，弹出这样的警告信息：`main.c(7): warning: #1-D: last line of file ends without a newline`

这个是由于在 `main` 函数的"}"后，没有加回车。

只要在 `main` 函数的"}"后加回车键，此警告信息即可消除。

error:#65

... test_menu.c(27):error:#65:expected a ";"

分数送你了，问题在你回答之前已经解决了，头文件里的结构体定义里的最后一行没有加";"，如 `NB menu{..};`

error: #1113:

折腾了大半天，才搞明白一个空操作的指令

先在网上查有的说是__asm{NOP;}, 从 intrins.h 里调用，可犄角旮旯全找了，也没看到什么 intrint.h 的文件。如果直接用，就出现

error: #1113: Inline assembler not permitted when generating Thumb code

最后搜索这条错误，知道是因为__asm("指令");这种语法是内联汇编（inline assembly）的语法。而 RMDK 下，内联汇编仅支持 ARM 汇编语言，不支持 Thumb 或者 Thumb-2 汇编语言；但内嵌汇编器支持 Thumb 和 Thumb-2。

__asm 放到一个单独的子函数再被调用就没问题了

如下：

```
__asm void nop(void)
{
    NOP
}
```

然后在之后的 C 代码中调用该函数：

```
void main()
{
    ...

    nop();

    ...
}
```

最新更新

来自 KEIL 官网

warning: #223-D

```
..\..\source\CCxx00_New.C(718): warning: #223-D: function "_NOP_" declared implicitly
```

在使用的文件中添加 `extern void _NOP();` 既可

warning: #1295-D

```
..\..\include\CCxx00_New.h(20): warning: #1295-D: Deprecated declaration CC_XCal - give arg types
```

没有用形参 定时时用 `void CC_XCal(void);` 即可

Error: L6218E: Undefined symbol

```
.\Obj\output.axf: Error: L6218E: Undefined symbol FSMC_NORSRAMCmd (referred from tft_lcd.o).
```

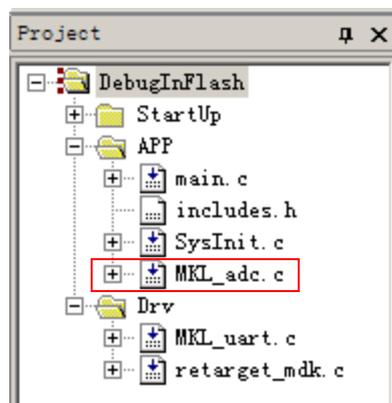
```
.\Obj\output.axf: Error: L6218E: Undefined symbol FSMC_NORSRAMInit (referred from tft_lcd.o).
```

Target not created

请教高人，我该如何处理。

你看看 `tft_lcd` 里面有没有添加 `fsmcXXX.h` 之类的头文件，这个错误是说你使用的函数没有被定义。

一般只要添加相应的头文件即可



error: #101:

error: #101: has already been declared in the current scope

将

```
#ifndef __STM32F10x_LIB_H
```

```
#define __STM32F10x_LIB_H
```

#endif 调整到最后!

```
233  
300 #endif /* __STM32F10x_LIB_H */  
301  
302 /***** (C) COPYRIGHT 2008 STMicroelectronics *****/  
303
```

error: #247: function "DelayUs" has already been defined

有一个同名的 DelayUs 类已经被定义过了

error: #109: expression must have (pointer-to-) function type

这个问题就是那 19 个 error 中的大部分，意思是表达式需要一个（指针）函数参数，我一开始以为是自己函数声明或者调用上的错误，但是看了下发现并没有错，后来查了好久发现原来是犯了一个很低级的错误：将宏定义和函数名重名了。因为我一开始想的是每一个宏定义对应一个函数名，这样做起来就比较清晰，但是我却很傻逼地将函数名每次直接复制到宏名，导致了这种蛋碎的结果。

warning: #61-D

warning: #68-D:

在 KeilARM 的 LPCARM,存在(1<<31)编译警告问题

```
main.c(174): warning: #61-D: integer operation result is out of range
```

```
main.c(174): warning: #68-D: integer conversion resulted in a change of sign
```

由于编译器默认 signed int 即 32 位有符号整数类型,而 1<<31 实际为 0x80000000,这样就有可能改写了符号位(最高位)

依此类推,(2<<30),(3<<29)...等都会出现编译警告问题.

解决办法为: ((unsigned int)1<<31),((unsigned int)2<<30),...

warning: #1295-D:

```
warning: #1295-D: Deprecated declaration ShowSendTime - give arg types
```

解决方法: 将 void ShowSendTime()改为 void ShowSendTime(void)

warning: #550-D:

```
warning: #550-D: variable "d" was set but never used
```

描述:变量'd'定义但从未使用,或者是,虽然这个变量你使用了,但编译器认为变量 d 所在的语句没有意义,编译器把它优化了.

解决:仔细衡量所定义的变量 d 是否有用,若是认定变量 d 所在语句有意义,那么尝试用 volatile 关键字修饰变量 d,若是真的没有用,那么删除掉以释放可能的内存.

error: #159:

```
.error: #159: declaration is incompatible with previous "wr_lcd" (declared at line 40)
```

```
void a(void) //函数 a 的实体
```

```
{  
    b(); //调用函数 b  
}
```

```
void b(void) //函数 b 的实体
{
    ...
}
```

这样如果点编译,就会产生 error: #159 的错误,因为当函数 a 调用函数 b 时,发现在这之前都没有函数 b 的任何声明.

解决方法:在函数 a 调用函数 b 之前,对函数 b 进行声明,如:

```
void b(void); //对函数 b 进行声明
```

```
void a(void) //函数 a 的实体
{
    b(); //调用函数 b
}
```

```
void b(void) //函数 b 的实体
{
    ...
}
```

error: #70:

error: #70: incomplete type is not allowed

原来是重复定义了,包含了两次的 `psock` 的定义,所以才会出现这种情况。因为我发现 `psock` 和 `pt` 是一样定义的,但是 `pt` 是不报错的,所以我就试图删除头文件中 `include` 头文件的那一行,错误消除了,谢谢你了,还是实践出真知啊,有些时候不能死编,要思考,哈哈。

keil MDK 编译器警告和错误详解(不定期更新)

工作后从单片机转成 ARM,刚开始用 ADS1.2 编译器,用了一段时间,因为我接手的项目的老程序正是用 ADS 编译的,部门也大多在用.在学单片机的时候用的是 keil c51 编译器,ads 和这个编译器在易用性上真是无法比较.后来渐渐知道 keil 已经被 arm 公司收购,现在 keil MDK 成为了 arm 官方编译器,所以决定重新投奔 keil,利用平时的时间,将原程序重新用 mdk 编译.mdk 的优点就没必要说了,在这里把平时遇到的编译器给出的警告和错误信息给出详解,希望给初学者一点帮助,发现错误,需要补充的欢迎留言.

warning: #550-D:

1.warning: #550-D: variable "d" was set but never used

描述:变量'd'定义但从未使用,或者是,虽然这个变量你使用了,但编译器认为变量 d 所在的语句没有意义,编译器把它优化了.

解决:仔细衡量所定义的变量 d 是否有用,若是认定变量 d 所在语句有意义,那么尝试用 volatile 关键字修饰变量 d,若是真的没有用,那么删除掉以释放可能的内存.

warning: #1-D:

2.warning: #1-D: last line of file ends without a newline

描述:文件最后一行不是新的一行.编译器要求程序文件的最后一行必须是空行,想了半天没想通为什么要这样.

解决:可以不理睬.若是觉得出现警告不爽,那么在出现警告的文件的最后一行敲个回车,空出一行.

warning: #111-D:

3. warning: #111-D: statement is unreachable

描述:声明不可能到达.多出现在这种场合:

```
int main(void)
{
...

while(1) //无限循环,这在不使用操作系统的程序中最常见
{
...

}

return 0; //这句声明在正常情况下不可能执行到,编译器发出警告
}
```

解决:不理睬.

warning: C3017W:

4. warning: C3017W: data may be used before being set

描述:变量'data'在使用前没有明确的赋值.如:

```
uint8 i,data; //定义变量 i 和 data,二者都没有明确赋值
```

```
for ( i = 0; i < 8; i++) //变量'i'在语句中被赋值 0
{
if ( IO1PIN & SO_CC2420 )
data |= 0x01; //变量'data'在使用前没有明确赋值,编译器发出警告
else
data &= ~0x01;
}
```

解决:应仔细衡量该变量的初始值是否为 0,若是,可以不理睬这个警告,因为 MDK 编译器在程序执行前,会将使用到的数据区初始化为 0,但若是该变量的初始值不应该是 0,忽略这个警告可能会引起致命错误.这个警告应引起足够重视.应养成变量赋初值的习惯,好在有编译器给把关.

warning: #177-D:

5. warning: #177-D: variable "temp" was declared but never referenced

描述:变量'temp'进行了声明但没有引用.多出现在声明了一个变量,但却没有使用它,它和 warning: #550-D:

variable "temp" was set but never used 不同之处在于 temp 从没有使用过.

解决:若是定义的变量确实没有用,删除掉;若是有用,则在程序中使用.

与该警告类似的还有 warning: #177-D: function "MACProcessBeacon" was declared but never referenced

warning: #940-D:

6. warning: #940-D: missing return statement at end of non-void function "DealwithInspect2"

描述:返回非空的函数"DealwithInspect2"的最后缺少返回值声明.如:

```
int DealwithInspect2(uint32 test)
{
...
...
...
//此处应该是 return x;返回一个 int 型数据,若是没有返回值,编译器产生警告
}
```

.warning: #1295-D:

7..warning: #1295-D: Deprecated declaration lcd_init - give arg types

描述:在定义函数的时候,如果你写上函数参数,就会有这个警告,比如 `void timer_init();` 这里就没有形参,如果这样的话,编译器会给出警告.

error: #65:

1. error: #65: expected a ";"

描述:缺少分号.大多是漏忘';'.

解决:双击错误行,在定位到错误点的附近找到没加';'号的语句,加上分号.并不一定在定位到的错误行才却分号,可能是这行的上一行,也可能是下一行.

error: #65: error: #20

2. error: #65: expected a ";"和 error: #20: identifier "xxxx" is undefined 一块出现,而且后面的 error: #20 错误可能一大堆

描述:这个错误对于第一次遇上的人来说绝对是个噩梦,当错误出现,满怀希望的双击错误提示,来到错误行时却愕然发现,错误行绝对没有错,于是找找错误行的上一行,下一行,没有错误,再找上上行,下下行...让人无比郁闷的事情出现了:编译提示的所有错误行都不可能没有错误出现.其实这最可能是你在.h文件声明外部变量或者函数时,没有在声明语句的最后加分号!如果你有很多模块,如 `main.c`,`lcd.c`,`key.c`...有很多头文件,如 `lcd.h`,`key.h`,若是在 `lcd.h` 文件声明函数时没有加分号,那么这种错误可能定为到 `main.c` 中,所以要检查所有头文件.

解决:仔细检查.h文件,将分号补上.

Error: L6200E:

3. Error: L6200E: Symbol flagu multiply defined (by uart0.o and main.o).

描述:变量(也是一种符号)flagu 多处定义(在 `uart0.c` 中和 `main.c` 都定义了).通常错在全局变量定义重复.比如:在 `main.c` 中定义全局变量 `flagu`:

```
uint8 flagu=0;
```

在 `uart0.c` 中也用到该变量,于是声明此变量,我通常都是先复制定义的变量再在变量前面加关键字 `extern` 修饰:

```
extern uint8 flagu=0;
```

然后编译,就会出现上面的连接错误,原因在于,我在 `uart0.c` 中是又定义了一个变量,而不是声明变量,因为我给变量赋了初值 "`flagu=0`",这样就重复定义了变量 `flag`.正确的声明方法是去掉赋值部分:

```
extern uint8 flagu;
```

解决办法:找到重复定义的变量,看情况修改一处.

error: #159:

4.error: #159: declaration is incompatible with previous "wr_lcd" (declared at line 40)

描述:在 `wr_lcd` 函数还没有声明之前就已经使用了.多出现在两种情况:第一种,`wr_lcd` 函数体还没有写,就已经用到了它,这种情况多出现在写一个程序的大体结构中,只是简单写一下框架.第二种情况比较常见,函数 `a` 调用函数 `b`,但函数 `b` 的函数体在函数 `a` 的下面:

```
void a(void) //函数 a 的实体
{
b(); //调用函数 b
}
```

```
void b(void) //函数 b 的实体
{
...
}
```

这样如果点编译,就会产生 `error: #159` 的错误,因为当函数 `a` 调用函数 `b` 时,发现在这之前都没有函数 `b` 的任何声明.

解决方法:在函数 `a` 调用函数 `b` 之前,对函数 `b` 进行声明,如:

```
void b(void); //对函数 b 进行声明
```

```
void a(void) //函数 a 的实体
{
b(); //调用函数 b
}
```

```
void b(void) //函数 b 的实体
{
...
}
```

error: #137:

5. error: #137: expression must be a modifiable lvalue

描述:表达式必须是一个可以修改的左值.主要出现在这种现象:

```
a=NUM;
```

NUM 是一个数值或表达式,a 为一个变量,但 a 被定义为像 `const` 这种不可更改的类型,导致 NUM 不能赋值给变量 a.

解决方法:要么放弃赋值,要么修改变量属性.

.error: #18:

6.error: #18: expected a ")"

如果是出现在 c 文件中,多半是因为少了一个")",或者错误行有编译器不识别的字符

如果出现在头文件中,错误行又是一个函数声明,多半是因为在函数声明中有编译器不认识的字符