

Keil 中的常见错误和警告

C51 编译器识别错类型有三种

- 1、致命错误：伪指令控制行有错、命令行指定的无效选项、访问不存在的原文件或头文件等。致命错误立即终止程序编译。
- 2、语法及语义错误：语法和语义错误都发生在源文件中。有这类错误时，给出提示但不产生目标文件，错误超过一定数量才终止编译。
- 3、警告：警告出现并不影响目标文件的产生，但执行时有可能发生问题，程序员应斟酌处理。

错误信息及可能发生的原因列表

*ERROR 100:unprintable character 0x??skipped

源文件中发现非法字符（注意，注解内的字符不做检查）。

*ERROR 101:unclosed string

字符串未用引号结尾。

*ERROR 102:string too long

字符串不得超过 511 个字符。为了定义更长的串，用户必须使用续行符 ‘\’ 逻辑的继续该字符串，在词汇分析时遇到以该符号结尾的行会与下行连接起来。

*ERROR 103: invalid character constant

试图再声明一个已定义的宏，已存在的宏可以用#undef 指令删除。预定义的宏不能删除。

*ERROR 104: identifier expected

预处理器指令期望产生一个标示符，如 #ifdef<name>。

*ERROR 105: unclosed comment

当注解无结束界定符（*/）时产生此错误。

*ERROR 106: unbalanced#if-endif controls

endif 的数量与 if 或 #ifdef 的数量不匹配。

*ERROR 107:include file nesting exceeds 9

include 指令后的文件名无效或丢失

*ERROR 108:expected string,如#error “string”

预处理器指令期望一个串变量。

*ERROR 109:<user error text>

由#error 伪指令引入的错误信息以错误信号形式显示。

*ERROR 110:missing directive

预处理行#后缺少伪指令。

*ERROR 111:unknown directive

预处理行#后不是伪指令。

*ERROR 112:misplaced ‘elif’

*ERROR 113:misplaced ‘else’

*ERROR 114:misplaced ‘endif’

指令 elif/else/endif 只有在 if、#ifdef、#ifndef 指令内才是合法的。

*ERROR 117:bad integer expression

if/elif 指令的数值表达式有语法错误。

*ERROR 118:missing' ('after macro identifier
宏调用中实参表的左括号丢失

*ERROR 119:reuse of macro formal parameter
宏定义形参名重复使用

*ERROR 120: 'C' unexpected in formal list
形参表中不允许有字符 'c' , 应用逗号代替

*ERROR 121: missing ')' after actual parameter
宏调用实参表的右括号丢失

*ERROR 122: illegal macro invocation
宏调用的实参表与宏定义中的形参表不同

*ERROR 123: missing macro name after 'define'
#define 伪指令后缺预定义的宏

*ERROR 124: expected macro formal parameter
宏定义要求形参名

*ERROR 125: declarater too complex
说明过于复杂

*ERROR 126: type-stack underflow
对象的声明至多只能包含 20 个类型修饰符 (**【】** , * , () ,) 错误 126 经常在错误 125 之前, 两者一起发生。

*ERROR 127: invalid storage class
对象用无效的存储类所说明。当在函数外用 auto/register 存储时会发生这种情况。

*ERROR 128: memory space: illegal memory space 'memory space' used
函数参数的存储类由存储模式 (SMALL LARGE COMPACT) 决定用户不能改变, 使用不同于存储模式的自动变量应该为静态的存储类

*ERROR 129: missing ';' before 'token'
该错误表示分号丢失, 通常该错误会引发一连串的错误, 引发的这些错误无关紧要。因为缺少分号后编译器不能做正确的语法分析

*ERROR 130: value out of range
using 或 interrupt 指令后参数越限。using 用的寄存器组号位 0-3, interrupt 需要 0-15 的中断号

*ERROR 131: duplicate function-parameter
函数中形参名重复, 形参名应彼此不同

*ERROR 132: not in formal parameter list
函数内参数声明使用的名字未出现在参数表中

*ERROR 133: char function(v0,v1,v2)
Char*v0,*v1,*v5;/ *' v5' 在形参中未出现

*ERROR 134: xdata/idata/pdata/data on function not permitted
函数总是驻留于 0x5xxxx 的 code 存储区, 不能位于 xdata/idata/pdata/data 空间

*ERROR 135: bad storage class for bit
位变量的定义可以接受 static 或 extern 的存储类, 用 REGIESTER 和 ALIEN 都是非法的

*ERROR 136: 'void' on variable
'void' 类型只允许作为函数的返回类型或与指针类型合用 (void*)

- *ERROR 137: illegal parameter type: 'function'
函数参数的类型不能是函数，然而函数指针可以作为参数
- *ERROR 138: interrupt () may not receive or return value (s)
中断函数既不能有参数又不能有返回值
- *ERROR 139: illegal use of 'alien'
关键字 alien 将函数定义为 PL/M51 规定的过程与函数结构。这意味着 C 函数中有参数的缩记符号（即 funct(…);）时是不能用 alien 的
- *ERROR 140: bit in illegal memory-space
位变量的定义可包含修饰符 DATA, 如果无修饰符则假定为 DATA。因为位变量始终位于 0x4xxx 的内部数据存储器中，当试图采用其他存储空间就会产生这个错误。
- *ERROR 141: NEAR<token>:expected<token>
编译器所见的单词是错误的。期望正确的单词
- *ERROR 142: invalid base address
Sfr 说明中的基址有错。有效基址为 0x80-0xff。如果声明采用 base^pos 形式，则基址是 8 的整数倍
- *ERROR 143: invalid absolute bit position
Sbit 说明中位地址必须在 0x80-0xff 之间
- *ERROR 144: base^pos: invalid bit position
Sbit 说明中位 pos 必须在 0~7 之间
- *ERROR 145: undeclared sfr
Sfr 未说明
- *ERROR 146: invalid sfr
绝对位地址说明 (base^pos) 包含无效的基地址。这个基地址必须与 sfr 名相对应
- *ERROR 147: object too large
对象不能超过 65536 (64k) 字节
- *ERROR 148: field not permitted in union
联合不能包含位成员，这个限制是由 8051 结构决定的。
- *ERROR 149: function member in struct/union
结构或者联合不能包含函数类型的成员。但是指向函数的指针是允许的
- *ERROR 150: bit member in struct/union
结构或者联合不能包含位类型的成员，这个限制是由 8051 结构决定的
- *ERROR 151: self relative struct/union
结构或者联合不能包含自身
- *ERROR 152: bit field type too small for number of bits
位域声明中指定的位数超过所给原型中位的数量
- *ERROR 153: named bit-field cannot have 0 width
命名的域宽度为 0 错误，只有未命名的位域允许是 0 宽度
- *ERROR 154: pro to field
无指向位域指针的类型
- *ERROR 155: char/int required for fields
位域基类型要求 char 或 int 类型，unsigned char 或 unsigned int 也有效
- *ERROR 156: alien permitted on function only

Alien 只能用于函数

*ERROR 157: var_parms on alien function

有变参数的函数不能用 alien , 因为 PL/M51 函数只能用固定数量的参数

*ERROR 158: function contains unnamed parameter

函数定义参数表中包含无名参数。无名参数只允许用于函数的原型中

*ERROR 159: type follows void

函数原型声明中可含一个空的参数表 f (void) 。Void 后不能再用其他类型定义

*ERROR 160: void invalid

Void 类型只能与指针合用或表明函数没有返回值

*ERROR 161: formal parameter ignored

函数内的外部函数引用声明使用了无类型的参数表。例如 “ extern(a, b, c); ” 要求形参表

*ERROR 162: duplicate function-parameter

函数内参数名重复

*ERROR 163: unknown array size

一般的不管是一维数组还是多维数组或外部数组, 都需要指定数组的大小, 这个大小是由编译器的初始化时计算, 这个错误表明试图为一个未定维的数组使用 sizeof 运算符, 会哦着一个多维数组的附加元素未定义。

*ERROR 164: ptr to null

这一个错误通常是由前一个错误造成的

*ERROR 165: ptr to bit

指向位的指针不是合法的类型

*ERROR 166: array of function

数组不能包含函数, 但可能包含指向函数的指针。

*ERROR 167: array of fields

位域不能安排为数组

*ERROR 168: array of bit

数组没有位类型

*ERROR 169: function returns function

函数不能返回函数, 但可以返回一个指向函数的指针

*ERROR 170: function returns array

函数不能返回数组, 但可返回指向数组的指针

*ERROR 171: missing enclosing switch

Break/continue 语句只能出现在 for, while, do while 或 switch 语句中间。

*ERROR 172: missing enclosing switch

Case 语句只能用在 switch 语句中

*ERROR 173: missing return-expression

返回类型不是 integer 的函数必须包含一条代表式的 return 语句。由于要与老版本兼容, 编译器对返回整形值的函数不做检查

*ERROR 174: return-expression on void-function

Void 函数不能返回值, 因此不能包含带表达式 return 的语句

*ERROR 175: duplicate case value

每个 case 语句必须包含一个常量表达式做其变量, 这个值不能在 switch 语句的各级中出现

多次

*ERROR 176: more than one default

Switch 语句中不能包含多于一个的 default 语句

*ERROR 177: different struct/union

赋值或参数传递中使用了结构/联合的不同类型

*ERROR 178: struct/union comparison illegal

根据 ANSI C, 两个结构或联合的比较是不允许的

*ERROR 179: cannot/cast from/to void-type

将 void 类型转化为其他类型数据或将其他类型转化为 void 类型都是非法的

*ERROR 180: cannot cast to 'function'

转化为 function 是非法的, 使用函数指针指向不同的函数

*ERROR 181: incompatible operand

在所给的运算符中至少有一个操作符类型是无效的

*ERROR 182: point to different object

报告指针使用不一致

*ERROR 183: unmodifiable value

预修改的对象位于 CODE 存储区, 因而不能修改

*ERROR 184: sizeof :illegal operand

Sizeof 运算符不能决定函数或位域大小

*ERROR 185: different memory space

对象说明的存储空间与前面的不一致

*ERROR 186: invalid dereference

这条错误信息可能由编译器内部问题产生的

*ERROR 187: not an lvalue

所需参量必须是可变对象的地址

*ERROR 188: unknown object size

无法计算对象的大小, 因为缺少数组的维数或因为通过 void 指针的间接访问

*ERROR 189: '&' on bit/sfr illegal

地址操作符 '&' 不允许用于位对象或 sfr

*ERROR 190: ' & ':not an lvalue

地址部是可变的对象, 不能作为左值

*ERROR 191: '&' on constant

试图为所列类型常数建立指针

*ERROR 192: '&' on array/function

地址操作符&不允许用于数组或函数, 函数和数组本身都代表了地址

*ERROR 193: illegal op-type(s)

*ERROR 193: illegal add/sub on ptr

*ERROR 193: illegal operation on bit(s)

*ERROR 193: bad operand type

当一个表达式使用给定运算符的非法操作类型时就会出现该错误, 使用个定运算符的非法操作类型的无效的表达式, 例如 bit+bit, ptr+ptr, 或 ptr*<any>。错误信息包括引起错误的运算符。

下列运算可使用位操作符

赋值 (=)

OR/复合 OR (|, |=)

AND/复合 AND (&, &=)

XOR/复合 XOR (^, ^=)

位或常数的按位比较 (=, !=)

取反 (~)

*ERROR 194: ' *' indirection to object of unknown size

间接操作符*不能用于 void 指针(void*)，因为指针所指的对象的大小时未知的

*ERROR 195: ' *' illegal indirection

间接操作符*不能用于非指针变量

*ERROR 196: mspace probably invalid

产生此警告是因为某些常数值赋给指针并且常数没有形成一个有效的指针值，有效的指针常数类型为 long/unsigned long。编译器对指针对象采用 24bit (3 字节)，低 16 位表示偏移，高 8 位表示存储类的选择，在低字节中，值从 1 到 5 表明了 xdata/pdata/idata/data/和 code 的存储类

*ERROR 197: illegal pointer assignment

试图将一个非法变量赋给指针，只有另一个指针或指针变量可以赋给指针

*ERROR 198: size of returns zero

求某些对象长度得到 0，如果对象是外部的或一个数组中并非所有维的大小都是已知时得到 0，这时候该值可能是错的。

*ERROR 199: left size of ' ->' requires struct/union pointer

->操作符的左边变量必须是结构或变量

*ERROR 200: left size of '.' requires struct/union

操作符的左边变量必须是结构/联合

*ERROR 201: undefined struct/union tag

所给的结构/联合标记名是未知的

*ERROR 202: undefined identifier

所给的标示符未定义

*ERROR 203: bad storage class(nameref)

该错误表示编译器的内部有问题

*ERROR 204: undefined member

所给的结构/联合成员名未定义

*ERROR 205: cannot call an interrupt function

中断函数不能像普通函数那样调用，因为这种函数的头端和尾端是为中断特殊编码的

*ERROR 206: missing function-prototype

调用的函数缺少原型说明

*ERROR 207: declared with 'void' parameter list

用 void 参数说明的函数不接受调用者传来的参数

*ERROR 208: too many actual parameter

函数调用包含了多余的实参

*ERROR 209: too few actual parameter

函数调用时传递的实参过少

*ERROR 210: too many nested calls

超过了 10 个函数嵌套调用的极限

*ERROR 211: call not to a function

函数调用时没有函数的地址或未对指向函数的指针赋值

*ERROR 212: indirect call with parameter

由于参数传递方法的限制，通过指针的间接函数调用不能直接作为实参。这种参数传递方法要求被调用的函数名已知，因为参数的写入要被写入调用函数的数据段。然而间接调用时函数的名字时未知的

*ERROR 213: left side of assign_op not an lvalue

在赋值操作符左边要求可变的对象

*ERROR 214: cannot cast non_pointer to pointer

非指针不能转化为指针

*ERROR 215: cannot cast pointer to not_int/pointer

指针可以转化为另一个指针或整数，但不能转化为其他类型

*ERROR 216: subscript on non_array or too many dimensions

对非数组使用了下标或数组维数过多

*ERROR 217: non_integral index

数组的下标表达式必须是整型类型

*ERROR 218: void_type in controlling expression

While, for 或 do while 语句中表达式不能是 void 类型

*ERROR 219: long constant truncated to int

企图把长整型常量截断为整型数是错误的

*ERROR 220: illegal constant expression

非法常量表达式

*ERROR 221: non_constant case/dim expression

Case 值或下标值 ([]) 要求用常量表达式

*ERROR 222: div by zero

*ERROR 223: mod by zero

编译器检测到 0 除或 0 模的错误

*ERROR 224: illegal operation on float/double

AND 和 NOT 一类的运算符不允许作用于 float/double

*ERROR 225: expression too complex ,simplify

表达式太复杂，必须简化

*ERROR 226: duplicate struct/union/enum tag

结构/联合/枚举类型中有重复标记

*ERROR 227: not a union tag

所给的标记名虽已定义，但不是联合的标记

*ERROR 228: not a struct tag

所给的标记名虽已定义，但不是结构的标记

*ERROR 229: not an enum tag

所给的标记名虽已定义，但不是枚举的标记

*ERROR 230: unknown struct/union/enum tag
所给的结构/联合/枚举标记名未定义

*ERROR 231: redefinition
所给的名字已经定义，不能再定义

*ERROR 232: duplicate label
所给的标号已经定义

*ERROR 233: undefined label
当对函数进行分析后，编译器检查到函数有未定义的标号，发出错误信息

*ERROR 234: '{' scope stack overflow(31)
超过了最大为 31 个的功能嵌套极限，多余的块被忽略

*ERROR 235: parameter<number>:different type
函数实参类型与函数原型中的不同

*ERROR 236: different length of parameter lists
所给的函数是参量与函数原型中的不同

Compling:C:\8051\LED.C

Error 237:' Led0n' :function already has a body // 试图定义已经定义过的函数
说明 Led0n()函数名称重复定义，即有两个以上一样的函数名称，解决方法修正其中的一个
函数名称使得函数名称都是独立的。

*ERROR 238: duplicate member

*ERROR 239: duplicate parameter
重复定义结构成员或函数参数

*ERROR 240: more than 128 local bit' s
位变量定义总数不能超过 128

*ERROR 241: auto segment too large
局部对象要求的空间超过了该模式的最大值。最大栈长定义如下：SMALL-128 字节, COMPACT-256 字节, LARGE-64k

*ERROR 242: too many initializers
初始化对象超限

*ERROR 243: string out of bounds
串中字符数超过了字符数组要求初始化的字符数

*ERROR 244: can' t initialize .bad type or class
试图初始化位或 sfr

*ERROR 245: unknown pragma, line ignored
未知的 pragma 语句，因此该行被忽略

*ERROR 246: floating point error
本错误发生在浮点变量超过 32 位有效字长时，32 位 IEEE 格式的浮点值的取值范围是±
1.75494E-38~±3.402832E+38

*ERROR 247: non_address +/-constant initializer
有效的初始化表达式必须是非地址量+/-常量

*ERROR 248: aggregate initialization needs curly braces
所有的组合变量（数组/结构或联合）初始化时要用花括号括起来

*ERROR 249: segment<name>:segment too large

编译器检测到过大的数据段，最大数据段长取决于存储器空间

*ERROR 250: ‘\esc’ ;value exceeds 255

串常数中\esc 转义序列的值超过有效域

*ERROR 251: illegal octal digit

不是有效的八进制数字

*ERROR 252: misplaced primary control、line ignored

一次性使用的编译控制伪指令必须在 C 模块开头指定，在#include 语句和变量说明之前

*ERROR 253: internal ERROR(ASMGEN\CLASS)

这种错误在以下情况下发生 (1)内部函数（如 testbit）被不正确激活。它发生在函数原型和实参表不存在匹配问题的时候。基于这个原因，头文件中的使用要适当（intrins.h, string.h）
(2) C51 识别出存在内部一致性错误，请向您的销售代理商查询

*ERROR 255: switch expression has illegal type

Switch 语句中的 case 语句必须具有类型 (u) char, (u) int 或 (u) short, 其他类型不允许 (如 bit)

*ERROR 256: conflicting memory model

Alien 属性的函数只能使用 SMALL 模式。函数的参数必须位于内部数据存储空间

*ERROR 257: alien function can not be reentrant

“alien” 属性的函数不能同时具有 “reentrant” 属性，函数的参数不能通过重入栈传递，这也适用于外部 “alien” 声明和 “alien” 函数

*ERROR 258: mspace illegal on struct/union member

不能为结构联合成员指定存储空间，但指向对象的指针可以

*ERROR 259: pointer: different mspace

当为指针赋值或做指针比较时，指针未指向存储在同一存储空间的对象时，会产生错误或者警告。如：

```
Char xdata *px; /*px to char in xdata memory*/
Char code *pc; /*pc to char in code memory*/
Void main()
{ char c;
  If(px==pc) ++c; /*warning 259*/
}
```

*ERROR 260: pointer truncation

指针转换时部分偏移量被截断，此时指针常量（如 char xdata）转为一个具有较小偏移区的指针（如 char idata）

*ERROR 261: bit in reentrant function

重入函数不能包含位变量，因为位变量不能存于重入栈，而只能位于 MCS51CPU 的可位寻址存储区中如：

```
Void test () reentrant
{ bit b0; /*illegal*/
  Static bit b1; /*legal*/
}
```

*ERROR 262: ' using/disable' : function returns bit

使用属性 using 选择寄存器组的函数或使用关中断（#pragma disable）功能的函数不能返回 bit 类型。如：

```

Bit test ()using 3/*error 261*/
{ bit b0;
return(b0);
}

```

*ERROR 263: save-stack overflow/underflow

“#pragma save” 最大嵌套级为 8 级。Save 和 restore 指令按 FIFO 原则工作

*ERROR 264: intrinsic<intrinsic_name>:declaration/ activation error

内部参数定义不正确

*ERROR 265: <name>recursive call to non_reentrant function

发现非重入函数被递归调用。直接递归用生成代码可有效查出，间接递归调用由 L51 发现
L51 连接定位器使用错误提示

Error 267 : ' DelayX1ms ' :requires ANSI-style prototype C:\8051\INPUT.C

说明：程序中有调用 DelayX1ms 函数, 但该函数没定义即未编写程序内容或函数已定义但未作声明
解决方法：编写 DelayX1ms 的内容，编写完后也要作声明或作外部声明，可在 delay.h 的包含文件声明成外部函数以便其它函数调用。

Error 268:任务定义错误任务 ID/优先级/using

任务声明错误

Error 271: asm/endasm 控制放错地方

asm 和 endasm 声明不能嵌套 endasm 要求一个汇编块前面用 asm 开头例如

```
#pragma asm
```

```
...
```

汇编指令

```
... #pragma endasm
```

Error 272: asm 要求激活 SRC 控制

在一个源文件中使用 asm 和 endasm 要求文件用 SRC 控制编译那么编译器就会生成汇编源文件然后
可以用 A51 汇编

Error 273: asm/endasm 在包含文件中不允许

在包含文件中不允许 asm 和 endasm 为了调试在包含文件不能有任何的可执行代码

Error 274: 非法的绝对标识符

绝对地址标识符对位目标函数和局部函数不允许地址必须和目标的存储空间一致, 例如下面的声明
是无效的, idata int _at_ 0x1000; 因为间接寻址的范围是 0x00 到 0xFF

Error 278: 常数太大

当浮点参数超出 32 位的浮点值范围就产生本错误 32 位 IEEE 值的范围是 $\pm 1.175494E-38$ 到 $\pm 3.402823E+38$

Error 279: 试图多次初始化一个目标

Error 280: 没有使用符号/标号/参数

在一个函数中声明了一个符号标号或参数但没有使用

Error 281: 非指针类型转换为指针

引用的程序目标不能转换成一个指针

Error 282: 不是一个 SFR 引用

本函数调用要求一个 SFR 作为参数

Error 283: asmparms 参数不适合寄存器

参数不适合可用的 CPU 寄存器

Error 284: <名称>: 在可覆盖空间函数不再可重入

一个可重入函数包含对局部变量的明确的存储类型标识符函数不再完全可重入

Error 300: 注释未结束

一个注释没有一个结束符*/

Error 301: 期望标识符

一个预处理器命令期望一个标识符

Error 302: 误用#操作符

字符操作符# 没有带一个标识符

Error 303: 期望正式参数

字符操作符# 没有带一个标识符表示当前所定义的宏的一个正式参数名

Error 304: 错误的宏参数列表

宏参数列表没有一个大括号逗号分开的标识符列表

Error 305: string/char 常数未结束

一个字符串活字符常数是无效的典型的后引号丢失

Error 306: 宏调用未结束

预处理器在收集和扩展一个宏调用的实际的参数时遇到输入文件的结尾

Error 307: 宏名称参数计算不匹配

在一个宏调用中实际的参数数目和宏定义的参数数目不匹配本错误表示指定了太少的参数

Error 308: 无效的整数常数表达式

一个 if/elif 命令的数学表达式包含一个语法错误

Error 309: 错误或缺少文件名

在一个 include 命令中的文件名参数是无效的或没有

Error 310: 条件嵌套过多 20

源文件包含太多的条件编译嵌套命令最多允许 20 级嵌套

Error 311: elif/else 控制放错地方

Error 312: endif 控制放错地方

命令 elif else 和 endif 只有在 if ifdef 或 ifndef 命令中是合法的

Error 313: 不能清除预定义的宏名称

试图清除一个预定义宏用户定义的宏可以用#undef 命令删除预定义的宏不能清除

Error 314: #命令语法错误

在一个预处理器命令中字符# 必须跟一个新行或一个预处理器命令名例如 if/define/ifdef

Error 315: 未知的#命令名称

预处理器命令是未知的

Error 316: 条件未结束

到文件结尾 endif 的数目和 if 或 ifdef 的数目不匹配

Compling :C:\8051\MANN.C

Error:318:can' t open file 'beep.h' // 不能打开文件文件名

说明: 在编译 C:\8051\MANN.C 程序过程中由于 main.c 用了指令 #include "beep.h", 但却找不到所致, 解决方法编写一个 beep.h 的包含文件并存入到 c:\8051 的工作目录中

Error 319:文件不是一个磁盘文件

指定的文件不是一个磁盘文件文件不能编辑

Error 320: 用户自定义的内容

本错误号未预处理器的#error 命令保留#error 命令产生错误号 320 送出用户定义的错误内容终止编译器生成代码

Error 321: 缺少<字符>

在一个 include 命令的文件名参数中缺少结束符例如#include<stdio.h

Error 325: 正参名称重复

一个宏的正参只能定义一次

Error 326: 宏体不能以## 开始或结束

Error 327: 宏名超过 50 个参数

每个宏的参数数目不能超过 50

L51 连接错误（与代码语法错误不同）

*ERROR101: SEGMENT COMBINATION ERROR

SEGMENT:segment_name

MODULE:filename(modulename)

由于连接错误所给段未能连入类型总段，并被忽略

*ERROR102:EXTERN ATTRIBUTE MISMATCH

SYMBOL:external_name

MODULE:filename(modulename)

所给外部符号名属性错，并被忽略

*ERROR103:EXTERN ATTRIBUTE DO NOTMATCH PUBLIC

SYMBOL:public_name

MODULE:filename(modulename)

所给外部符号名属性与公用符号名不匹配，并被忽略

*ERROR104:MUTIPULIC DEFINITION

SYMBOL:pulic_name

MODULE:filename(modulename)

所给公用符号重名

*ERROR105:PUBLIC REFERS TO IGNORED SEGMENT

SYMBOL:public_name

MODULE:filename(modulename)

所给外部符号名属性错，并被忽略

*ERROR106: SEGMENT OVERFLOW

SEGMENT:segment_name

所给段长超过 64，未处理

*ERROR 107:ADDESS SPACE OVERFLOW

SPACE: DATA

SEGMENT: _DATA_GOUP_

LENGTH: 0018H

说明: data 存储空间的地址范围为 0~0x7f, 当公用变量数目和函数里的局部变量数量过多时, 如果存储模式设为 SMALL 则局部变量先使用工作寄存器 R2~R7 作暂存, 当存储器不够用时则会以 data 型别的空间作暂存, 当个数超过 0x7f 时就会出现地址不够的现象, 解决方法将以 data 型类别定义的公共变量修改为 idata 型类别的定义。

*ERROR107:ADDRESS SPACE OVERLAP

SPACE:space_name

SEGMENT:segment_name

由于存储空间不够, 所给类型总段未能装入, 已被忽略

*ERROR108:SEGMENT IN LOCATING CONTROL CANNOT ALLOCATED

SEGMENT:segment_name

命令行定位控制中的段由于属性问题未能分配

*ERROR109:EMPTY RELOCATABLE SEGMENT

SEGMENT:segment_name

可在定位类型总段长度为零, 未定位

*ERROR110:CANNOT FIND SEGMENT

SEGMENT:segment_name

命令行所给的段在输入模块中未找到, 被忽略

*ERROR111:SPECIFIED BITADDRESS NOT ON BYTE MEMORY

SEGMENT:segment_name

位地址不在字界上, 位段被忽略

*ERROR112:SEGMENT TYPE NOT LEGAL FOR COMMAND

SEGMENT:segment_name

命令行所给的段类型非法, 被忽略

*ERROR114:SEGMENT DOES NOT FIT

SPACE:space_name

SEGMENT:segment_name

BASE:base_address

LENGTH: segment_length

由于所给段的长度或者基地址未定位, 故被忽略

*ERROR115:INPAGE SEGMENT IS GREATER THAN 256 BYTES

SEGMENT:segment_name

所给 INPAGE 属性的段长于 256 字节未能连入类型总段, 并被忽略

*ERROR116:INBLOCK SEGMENT IS GREATER THAN 2048 BYTES

SEGMENT:segment_name

所给 INBLOCK 属性的段长于 2048 字节未能连入类型总段, 被忽略

*ERROR117:BITADDRESSABLE SEGMENT IS GREATER THAN 16 BYTE

SEGMENT:segment_name

所给 BITADDRESSABLE 属性的段长于 16 字节未能连入类型总段, 被忽略

*ERROR118:REFERENCE MADE TO ERRONEOUS EXTERNAL

SYMBOL:symbol_name

MODULE:file_name(modulename)

ADDRESS:code_address

企图访问错误的外部程序地址，此错误常与 107 错误同时产生，详见 107 错误。

*ERROR119:REFERENCE MADE TO ERRONEOUS SEGMENT

SYMBOL:symbol_name

MODULE:file_name(modulename)

ADDRESS:code_address

企图访问错误段的程序地址

*ERROR120:CONTENT BELONGS TO ERRONEOUS SEGMENT

SEGMENT:segment_name

MODULE:file_name(modulename)

该内容属于有错误的段

*ERROR121:IMPROPER FIXUP

MODULE:file_name(modulename)

SEGMENTsegment_name

OFFSET:segment_name

根据所给段和偏移地址的到的是不当的地址

*ERROR122:CANNOT FIND MODULE

MODULE:file_name(modulename)

命令行所给的模块未能找到

L51 致命错误

*FATAL ERROR201:INVALID COMMAND LINE SYNTAX

Partial command line

命令行句法错。命令行显示到出错处。

*FATAL ERROR202:INVALID COMMAND LINE , TOKEN TOO LONG

Partial command line

非法命令行，单词太长。命令行显示到出错处

*FATAL ERROR203:EXPECTED ITEM MISSING

Partial command line

缺项。命令行显示到出错处。

*FATAL ERROR204:INVALID KEYWORD

Partial command line

非法关键字

*FATAL ERROR205:CONSTANT TOO LONG

Partial command line

常量大于 0xffff。命令行显示到出错处。

*FATAL ERROR206:INVALID CONSTANT

Partial command line

命令行常量无效（如 16 进制数以字母开头）。命令行显示到出错处。

*FATAL ERROR207:INVALID NAME

Partial command line

模块名或段名无效。命令行显示到出错处。

*FATAL ERROR208:INVALID FILENAME
Partial command line
文件名无效。命令行显示到出错处。

*FATAL ERROR209:FILE USED IN CONFLICTING CONTEXTS
FILE:filename
所给的文件名用于有矛盾之处。命令行显示到出错处。

*FATAL ERROR210:I/O ERROR ON INPUT FILE
System error message
FILE:filename
访问输入文件时检测到有错，并有后面的 EXCEPTION 给出具体的错误描述

*FATAL ERROR211:I/O ERROR ON OUTPUT FILE
System error message
FILE:filename
访问输出文件时检测到有错，并有后面的 EXCEPTION 给出具体的错误描述

*FATAL ERROR212:I/O ERROR ON LISTING FILE
System error message
FILE:filename
访问列表文件时检测到有错，并有后面的 EXCEPTION 给出具体的错误描述

*FATAL ERROR213:I/O ERROR ONWORK FILE
System error message
FILE:filename
访问工作文件时检测到有错，并有后面的 EXCEPTION 给出具体的错误描述

*FATAL ERROR214:I/O INPUT PHASE ERROR
MODULE:filename(modulename)
L51 在进行第二次扫描时遇到不同的数据发生该错误，可能是因汇编错误引起

*FATAL ERROR215:CHECK SUM ERROR
MODULE:filename(modulename)
校验和与文件内容不一致

*FATAL ERROR216:INSUFFICIENTMEMORY
MODULE:filename(modulename)
执行 L51 的内存空间不够

*FATAL ERROR217:NO MODULE TO BE PROCESSED
缺少应该被处理的模块

*FATAL ERROR218:NOTAN OBJECT FILE
FILE:filename
所给文件非目标文件

*FATAL ERROR219:NOTAN 8051 OBJECT FILE
FILE:filename
所给文件非 8051 目标文件

*FATAL ERROR220:INVALID INPUTMODULE
FILE:filename
所给输入模块无效，可能是由汇编错误引起的

*FATAL ERROR221:MODULE SPECIFIED MORE THAN ONCE
Partial command line
命令行上多次包含同一模块。命令行显示到出错处

*FATAL ERROR222:SEGMENT SPEXIFIED MORE THAN ONCE
Partial command line
命令行上多次包含同一段。命令行显示到出错处

*FATAL ERROR224:DUPLICATE KEYWORD OR CONFLICTING CONTROL
Partial command line
命令行上多次包含同一关键字或者存在相互矛盾的控制选项。命令行显示到出错处

*FATAL ERROR225:SEGMENTADDRESS ARE NOT IN ASCENDING ORDER
Partial command line
定位控制的段地址未按照升序显示。命令行显示到出错处

*FATAL ERROR226:SEGMENTADDRESS INVALID FOR CONTROL
Partial command line
定位控制的段的段地址无效。命令行显示到出错处

*FATAL ERROR227:PARAMETER OUT RANGE
Partial command line
所给 PAGEWIDTH 和 PAGELENGTH 参数越界。命令行显示到出错处

*FATAL ERROR228:PARAMETER OUT RANGE
Partial command line
命令行上 RAMSIZE 参数越界。命令行显示到出错处

*FATAL ERROR229:INTERAL PROCESS ERROR
Partial command line
L51 检测到内部处理错。请询问代理商

*FATAL ERROR230:STRARTADDRESS SPECIFIED MORE THAN ONCE
Partial command line
命令行上包含多个未命名组段的起始地址。命令行显示到出错处

*FATAL ERROR233:ILLEGAL USE OF *IN OVERLAY CONTROL
Partial command line
命令行 OVERLAY 定位选择非法使用了*号（如*! *或*~*）。命令行显示到出错处

E.5 异常信息

L51 某些错误的原因由系统的 EXCEPTION 给出。

*EXCEPTION 0021:PATH OR FILE NOT FOUND
路径名或文件名未找到。

*EXCEPTION 0026H:ILLEGAL FILE ACCESS
试图写或者删除写保护文件。

*EXCEPTION 0029H:ACCESS FILE DENIED
所给的文件实际是目录。

*EXCEPTION 002AH:I/O-ERROR
欲写的驱动器已满或未准备好。

*EXCEPTION 0101H:ILLEGAL CONTEXT

命令行的语意非法。如对打印机进行读操作。

警告

警告在列表文件中生成信息警告信息用下面的格式

*** WARNING number IN LINE line OF file: warning message

number: 错误号

line : 在源文件或包含文件中的对应行号

file : 错误产生的源或包含文件名

warning message: 警告的内容

*WARNING 1:UNRESOLVED EXTERNAL SYMBOL

SYMBOL:MUSIC3

MODULE:C:\8051\MUSIC.OBJ (MUSIC)

没把 C 文件添加到项目中，还有可能是因为没有被调用的已经定义的函数。

*WARNING 2:REFERENCE MADE TO UNRESOLVED EXTERNAL

SYMBOL:MUSIC3

MODULE:C:\8051\MUSIC.OBJ (MUSIC)

ADDRESS:0018H // ROM 空间地址

说明：程序中有调用 MUSIC 函数，但未将该函数所在的 C 文件加入到工程作编译和连接

解决方法：设 MUSIC3 函数在 MUSIC.C 里将 MUSIC.C 添加到工程文件中。

*WARNING1:UNSOLVED EXTERNAL SYMBOLS

SYMBOLS: external_name

MODULE: filename (modulename)

指定模块的外部符号在 PUBLIC 符号表中找不到

*WARNING2:REFERENCE MADE TO UNSOVED EXTERNAL

SYMBOLS: external_name (外部函数名)

MODULE: filename (modulename) (模块名)

ADDRESS:code_address

访问了未能匹配的外部符号 code 地址

*WARNING4:DATA SPACE MEMORY OVERLAP

FROM:byte, bit, address

TO: byte, bit, address

数据空间指定范围出现覆盖

*WARNING5:CODE SPACE MEMORY OVERLAP

FROM:byte, bit, address

TO: byte, bit, address

程序空间指定范围出现覆盖

*WARNING6 :XDATA SPACE MEMORY OVERLAP FROM : 0025H TO: 0025H

说明：外部 RAM 的 0025H 重复定义地址

解决方法：外部资料 RAM 的定义如下 Pdata unsigned char XFR_ADC _at_0x25，其中 XFR_ADC

变量的地址为 0x25, 请检查是否有其它的变量地址也是定义在 0x25 处并修正它。

*WARNING7:MODULE NAME NOT UNIQUE

MODULE:filename(modulename)

模块名重名。模块未处理

*WARNING8:MODULE NAME EXPLICITLY REQUESTED FROM ANOTHER FILE

MODULE:filename(modulename)

其他文件指名要求本模块名

*WARNING9:EMPTY ABSOLUTE SEGMENT

MODULE:filename(modulename)

本模块包含空的绝对段，因未定位，它可能在不通知的情况下随时被覆盖

*WARNING10:CANNOT DETERMINE ROOT SEGMENT

L51 对输入文件要求分辨是 C51 还是 PL/M 文件，然后进行流程分析，在无法确定的时候，发出本警告。它发生在主程序被汇编调用的时候，需要程序员用 OVERLAP 特殊控制选项进行干预

*WARNING11:CANNOT FIND SEGMENT OR FUNCTION NAME

NAME:overlap_control_name

在目标模块中找不到 OVERLAP 控制选项中规定的段或者函数间调用

*WARNING12:NO REFERENCE BETWEEN SEGMENTS

SEGMENT1:segment_name

SEGMENT2:segment_name

试图用 OVERLAP 控制选项删除本来不存在的段间访问或者函数间调用

*WARNING13:RECURSIVE CALL TO SEGMENT

SEGMENT:segment_name

CALLER:segment_name

CALLER 段递归调用 SEGMENT 段。PL/M51 和 C51 的非重入函数不允许递归调用

*WARNING14:INCOMPATIBLE MEMORY MODEL

MODULE:filename(modulename)

MODEL:memory_model

指定模块试图与以前不同的存储模式编译。

*WARNING15: MULTIPLE CALL TO SEGMENT

SEGMENT: ?PR?SPI_RECEIVE_WORD?D_SPI

CALLER1: ?PR?VSYNC_INTERRUPT?MAIN

CALLER2: ?C_C51STARTUP

该警告表示连接器发现有一个函数可能会被主函数和一个中断服务程序(或者调用中断服务程序的函数)同时调用，或者同时被多个中断服务程序调用。出现这种问题的原因之一是这个函数是不可重入性函数，当该函数运行时它可能会被一个中断打断，从而使得结果发生变化并可能会引起一些变量形式的冲突(即引起函数内一些数据的丢失，可重入性函数在任何时候都可以被中断打断，一段时间后又可以运行，但是相应数据不会丢失)。原因之二是用于局部变量或全局变量的内存区被其他函数的内存区所覆盖，如果该函数被中断，则它的内存区就会被使用，这将导致其他函数的内存冲突。

解决办法：将该函数设为重入型。例如：**void myfunc(void) reentrant { ... }**，这种方法消耗更多的 RAM 并会降低重入函数的执行速度。另一办法是复制两份该函数的代码，一份到主程序中，另一份复制到中断服务程序中。

*WARNING15: MULTICALL TO SEGMENT

SEGMENT:segment_name

CALLER1:segment_name

CALLER2:segment_name

两个函数调用同一个函数（如主函数和中断函数），参数和局部变量将被覆盖

*WARNING 16:UNCALLED SEGMENT, IGNORED FOR OVERLAY PROCESS

SEGMENT: ?PR?_DELAYX1MS?DELAY

说明 DelayX1ms()函数未被其它函数调用也会占用程序记忆体空间，解决方法删除 DelayX1ms()

函数或利用条件编译#if ...#endif, 可保留该函数并不编译。

*WARNING 173 缺少返回表达式

一个函数返回一个除了 int 类型以外的别的类型的值必须包含一个返回声明包括一个表达式为了兼容旧的程序对返回一个 int 值的函数不作检查

*WARNING 182 指针指向不同的目标

一个指针关联了一个不同类型的地址

*WARNING 185 不同的存储空间

一个目标声明的存储空间和前面声明的同样目标的存储空间不同

*WARNING 196 存储空间可能无效

把一个无效的常数值分配给一个指针无效的指针常数是 long 或 unsigned long 编译器对指针采用 24 位 3 字节低 16 位代表偏移高 8 位代表选择的存储空间

*WARNING 198 sizeof 返回零

一个目标的大小计算结果为零。如果目标是外部的或如果一个数组的维数没有全知道则值是错误的

Warning 206:' Music3':missing function-prototype // 缺少函数原型

说明 Music3()函数未作声明或未作外部声明所以无法给其他函数调用，解决方法将叙述 void Music3(void) 写在程序的最前端作声明，其它文件的函数则要写成 extern void Music3 (void), 即作外部声明。

*WARNING 209 实参太少

在一个函数调用中包含太少的实参

*WARNING 219 long 常数被缩减为 int

一个常数表达式的值必须能被一个 int 类型所表示

*WARNING 235: parameter 1: different types

第一个参数类型错误

*WARNING 245 未知的 pragma 本行被忽略

#pragma 声明是未知的因此整行程序被忽略

*WARNING 258 struct/union 成员的存储空间方法

参数的存储空间被忽略

一个结构的成员或一个参数不能指定存储类型但是指针所指的目标可以包含一个存储类型例如

struct vp{ char code c;int xdata i; }; 产生警告 258

struct vl{ char c;int xdata *i; }; 对 struct 是正确的声明

*WARNING 259 指针不同的存储空间

两个要比较的指针没有引用相同的存储类型的目标

*WARNING 260 指针折断

把一个指针转换为一个更小偏移区的指针转换会完成但大指针的偏移会折断来适应小指针

*WARNING 261 bit 在重入函数

一个 reentrant 函数不能包含 bit 因为 bit 标量不能保存在虚拟堆栈中

*WARNING 265 名称对非重入函数递归调用

发现对一个非重入函数直接递归这可能是故意的但对每个独立的情况进行功能性检查通过生成的代码间接递归由连接/定位器检

*WARNING 271 asm/endasm 控制放错地方

asm 和 endasm 不能嵌套 endasm 要求一个以 asm 声明开头的汇编块

例如

```
#pragma asm
```

```
...
```

汇编指令

```
...
```

```
#pragma endasm
```

*WARNING 275 表达式可能无效

编译器检测到一个表达式不生成代码例如

```
void test(void)
```

```
{
```

```
    int i1, i2, i3;
```

```
    i1, i2, i3; /* 死表达式 */
```

```
    i1 << i3; /* 结果未使用 */
```

```
}
```

*WARNING 276 常数在条件表达式

编译器检测到一个条件表达式有一个常数值，在大多数情况下是一个输入错误例如

```
void test(void)
```

```
{
```

```
    int i1, i2, i3;
```

```
    if( i1 = 1) i2 = 3; /* 常数被赋值 */
```

```
    while( i3 = 2); /* 常数被赋值 */
```

```
}
```

*WARNING 277 指针有不同的存储空间

一个 typedef 声明的存储空间冲突例如

```
typedef char xdata XCC; /* 存储空间 xdata */
```

```
typedef XCC idata PICC; /* 存储空间冲突 */
```

Warning 280: 'i' :unreferenced local variable // 符号或标号未使用

声明的局部变量 i 在函数中未作任何的存取操作，解决方法删除除函数中 i 变量的声明。

Warning C290: missing return value

函数定义时缺少返回值类型声明，例如：main()，应定义为 void main()或 int main()。

warning C291: not every exit path returns a value

不是每一个退出路径返回一个值，通常是在 switch 语句中出现的。

Warning 307 宏名称参数计算不匹配

一个宏调用的实参的数目和宏定义的参数数目不匹配表示用了太多的参数过剩的参数被忽略

*WARNING 317 宏名称重新定义无效

一个预定义的宏不能重新定义或清除参考 138 页的预定义宏常数

*WARNING 322 未知的标识符

在一个 #if 命令行的标识符未定义等效为 FALSE

*WARNING 323 期望新行发现多余字符

一个 # 命令行正确但包含多余的非注释字符例如 #include <stdio.h> foo

*WARNING 324 期望预处理器记号

期望一个预处理器记号但输入的是一个新行例如 #line 这里缺少 #line 命令的参数

*WARNING 329: single-line comment contains line-continuation

Keil c51 中第 194 行代码如下(最后字符为反斜杠, 如图):

```
while(ADOK){;}//1212\
```

C329 提示的中文意思是: 单行注释中有续行, 解决的办法, 把第 194 行最后的字符反斜杠“\”删去即可!