

如何将 IAR Embedded Workbench for ARM v7.x 版本工程移植到 v8.x 版本

2017年3月 IAR Systems 发布了 IAR Embedded Workbench for ARM (简称 EWEARM)全新的 v8.10.1 版本。v8 版本做了重大升级，本文将基于此次升级内容介绍如何将旧版本移植到新版本上。

语言标准的升级

EWARM v7 的版本支持的 C 语言标准是 C89 和 C99, C++ 语言标准是 C++03, EWARM 8.10.1 支持 C89 和 C11, C++ 标准升级到 C++14。编译器采用不同的语言标准，对编译生成的机器码可能会有一些影响。C++14 语言标准带来了一些新的特性，如自动类型推断、新的二进制常量和空指针常量等等，新语言标准对于旧项目的影响如下：

移植现有的 C/C++ 应用程序非常简单，需要关注的点是 IAR 不再支持 Embedded C++ 和 Extended Embedded C++，及不再支持库模式。Embedded C++ 和 Extended Embedded C++ 是一些针对嵌入式应用涉及的 C++ 语言子集。不支持的原因是显而易见的，因为最终用户可以定义自己的 C++ 子集并将其扩展到完整的 C++ 语言。

(1) 移植嵌入式 C++ 应用程序时，C++ 库符号现在位于 std 命名空间中可能是主要的一个问题。用户必须在对这些符号的引用之前使用 "std::" 前缀修饰，或者必须在每个 C++ 源文件和部分头文件中插入如下语句：

```
using namespace std;
```

该语句插入的位置是在 include 头文件包含指令之后，但在任何 C++ 库函数符号被引用之前。

(2) 类似于上文，放弃对 Extended Embedded C++ STL 的支持的副作用是 STL 只能驻留在默认内存中并以默认指针运行。更具体地信息请参考《C/C++ Development Guides》中的“Porting code from EC++ or EEC++”章节。

wchar_t 宽度的变化

在 EWARM 7.80 及更早版本，wchar_t 的大小是 2 字节，而在 EWARM 8.11 及以后的版本是 4 字节。如果在 v8 版本上使用 v7 版本编译的库，在链接时就会报“Warning [Lt009]: Inconsistent wchar_t size”警告。

解决方法：

如果遇到上述编译警告，必须要在 v8 的版本上重新编译 v7 版本上编译的库，否则可能会有潜在的风险。强烈推荐在升级到 8.x 版本时，将所有源码和库在 8.x 版本上重新编译一次。

工程选项设置参数丢失

差异较大版本迁移，可能会遇到配置中很多的设置项都丢失了，比如器件型号，头文件路径都变成了空白，这种情况在编译的时候就会报错。解决的方法是，重新手动设置相关的设置项。

如果有旧工程对应的 EWARM 版本，可以用此版本打开旧的工程，将头文件路径、全局宏定义字符等复制到新版本工程丢失的设置项中。

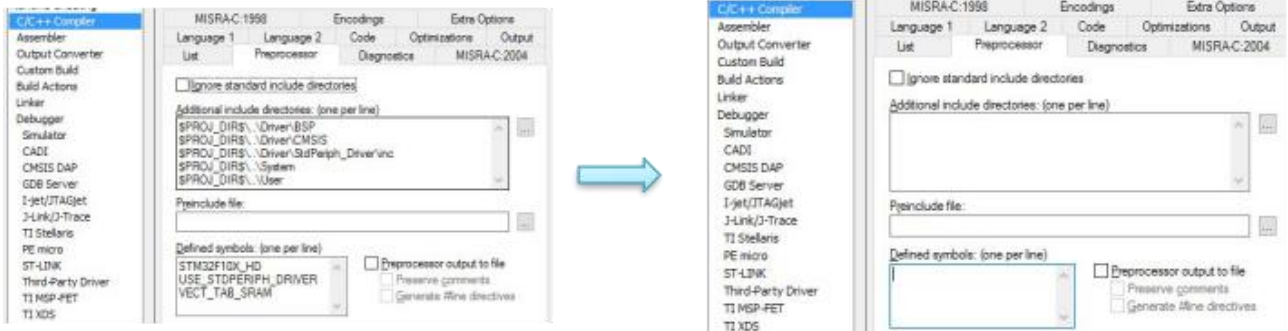


图 1

然后对照之前工程的设置，将新工程其余的设置项依次设置一遍，经过对工程重新设置，一般都能够正常编译和调试。

比较坏的一种情况。经过对比旧工程，重新对新工程进行设置之后，在编译和调试的时候还是会有问题，甚至是会遇到没有任何报错的未知问题。这时候如果工程必须升级到新版本上来的话，比较可行的方式是基于新版本重新创建工程。但是工作量可能会比较大，而且如果工程中有汇编的代码，有时候可能还需要修改汇编的部分，用户还是需要根据实际情况评估一下。

创建新工程如果源码和目录比较多的情况，手动一个个添加会比较的耗费时间，目前 IAR 也没有提供直接导入工程和源码的工具。有一个方法可以减轻工作量，即直接通过修改*.ewp 工程描述文件快速在工程中添加目录和头文件路径，以下是示例。

1). 首先在新版本 EWARM 中新建一个工程，并将源码目录复制到工程文件夹，目录结构要保持与旧版本工程的一致，建议直接复制之前的工程目录，删除目录中与工程有关的文件。

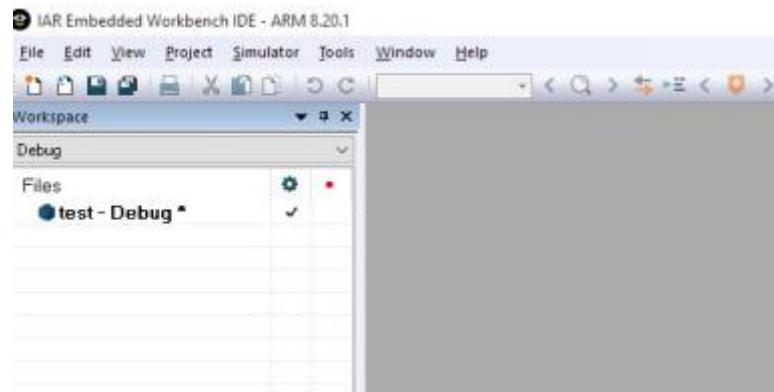


图 2

2). 用文本编辑工具打开新、旧两个工程的 ewp 工程文件，将旧版本 ewp 文件中<group> </group>的内容复制到新工程 ewp 文件的最后，保存文件。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project>
3   <fileVersion>3</fileVersion>
4   <configuration>
1025 </configuration>
2045 <group>
2046   <name>BSP</name>
2047   <file>
2048     <name>$PROJ_DIR$..\Driver\BSP\BSP_led.c</name>
2049   </file>
2050   <file>
2051     <name>$PROJ_DIR$..\Driver\BSP\BSP_usart.c</name>
2052   </file>
2053 </group>
2054 <group>
2055   <name>CMSIS</name>
2056   <file>
2057     <name>$PROJ_DIR$..\Driver\CMSIS\startup_stm32f10x_cl.s</name>
2058   </file>
2059   <file>
2060     <name>$PROJ_DIR$..\Driver\CMSIS\system_stm32f10x.c</name>
2061   </file>
2062 </group>

```

图 3

然后就可以看到目录和源码已经添加新的工程中。

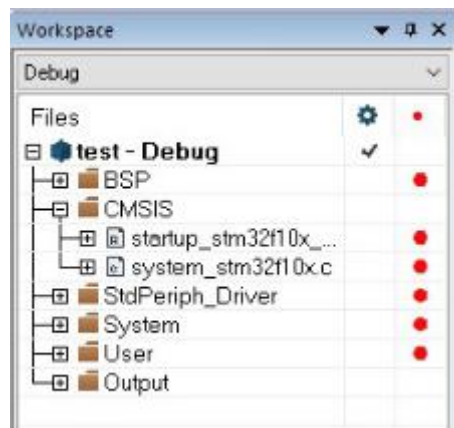


图 4

3).在 ewp 文件中搜索“CCIncludePath2”字段，将此内容复制到新版本 ewp 工程文件对应的位置。这些内容是头文件路径的定义。

```

<option>
  <name>CCIncludePath2</name>
  <state>$PROJ_DIR$..\Driver\BSP</state>
  <state>$PROJ_DIR$..\Driver\CMSIS</state>
  <state>$PROJ_DIR$..\Driver\StdPeriph_Driver\inc</state>
  <state>$PROJ_DIR$..\System</state>
  <state>$PROJ_DIR$..\User</state>
  <state>$PROJ_DIR$..\LibSection</state>
</option>

```

图 5

4).在 ewp 文件中搜索“CCDefines”字段，将内容复制到新工程的 ewp 对应位置。这些内容是全局宏定义(如果工程有使用)。

```
<option>  
  <name>CCDefines</name>  
  <state>STM32F10X_CL</state>  
  <state>USE_STDPERIPH_DRIVER</state>  
</option>
```

图 6

保存 ewp 工程文件，头文件路径和全局宏定义字符就完成了设置。

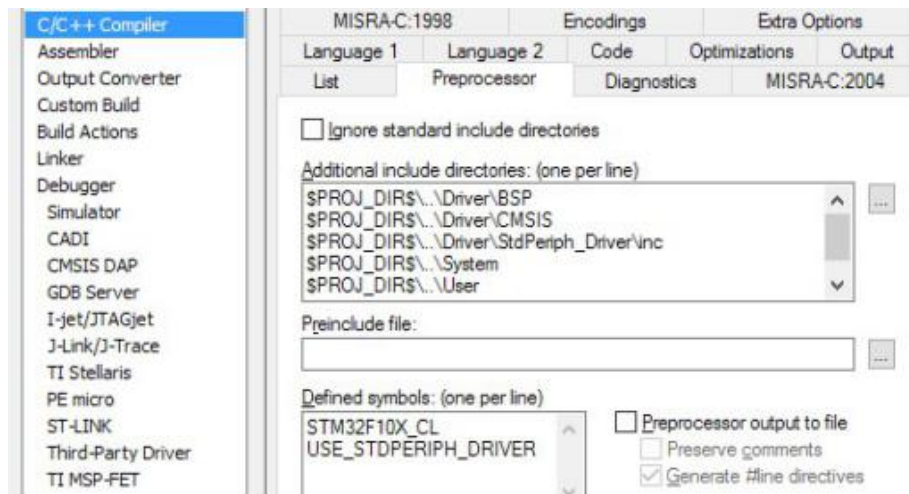


图 7

5). 其他是工程选项再参考之前的工程，手动进行设置。

汇编代码差异

如果有汇编代码(包括内联汇编)，有可能编译器版本的差异报错则也需要修改。这部分的话需视具体出现的问题而定如何对汇编代码进行修改。