

零死角玩转STM32



MDK的编译过程及文件类型全解

淘宝：firestm32.taobao.com

论坛：www.chuxue123.com



扫描进入淘宝店铺

主讲内容



01

编译过程

02

程序的组成、存储与运行

03

编译工具链

04

MDK工程的文件类型

05

实验：自动分配变量到外部SDRAM

06

**实验：优先使用内部SRAM并
分配堆到SDRAM**

MDK的编译过程及文件类型全解



MDK工程的文件类型

除了上述编译过程生成的文件，MDK工程中还包含了各种各样的文件，下面我们统一介绍，MDK工程的常见文件类型如下表：

后缀	说明
Project目录下的工程文件	
.uvguix	MDK5工程的窗口布局文件，在MDK4中.UVGUI后缀的文件功能相同
.uvprojx	MDK5的工程文件，它使用了XML格式记录了工程结构，双击它可以打开整个工程，在MDK4中.UVPROJ后缀的文件功能相同
.uvoptx	MDK5的工程配置选项，包含debugger、trace configuration、breakpooints以及当前打开的文件，在MDK4中.UVOPT后缀的文件功能相同
*.ini	某些下载器的配置记录文件
源文件	
*.c	C语言源文件
*.cpp	C++语言源文件
*.h	C/C++的头文件
*.s	汇编语言的源文件
*.inc	汇编语言的头文件(使用“\$include”来包含)



MDK的编译过程及文件类型全解

Output目录下的文件	
*.lib	库文件
*.dep	整个工程的依赖文件
*.d	描述了对应.o的依赖的文件
*.crf	交叉引用文件, 包含了浏览信息(定义、引用及标识符)
*.o	可重定位的对象文件(目标文件)
*.bin	二进制格式的映像文件, 是纯粹的FLASH映像, 不含任何额外信息
*.hex	Intel Hex格式的映像文件, 可理解为带存储地址描述格式的bin文件
*.elf	由GCC编译生成的文件, 功能跟axf文件一样, 该文件不可重定位
*.axf	由ARMCC编译生成的可执行对象文件, 可用于调试, 该文件不可重定位
*.sct	链接器控制文件(分散加载)
*.scr	链接器产生的分散加载文件
*.lnp	MDK生成的链接输入文件, 用于调用链接器时的命令输入
*.htm	链接器生成的静态调用图文件
*.build_log.htm	构建工程的日志记录文件
Listing目录下的文件	
*.lst	C及汇编编译器产生的列表文件
*.map	链接器生成的列表文件, 包含存储器映像分布
其它	
*.ini	仿真、下载器的脚本文件

这些文件主要分为MDK相关文件、源文件以及编译、链接器生成的文件。我们以“多彩流水灯”工程为例讲解各种文件的功能。

MDK的编译过程及文件类型全解



uvprojx、uvoptx、uvguix及ini工程文件

在工程的“Project”目录下主要是MDK工程相关的文件：

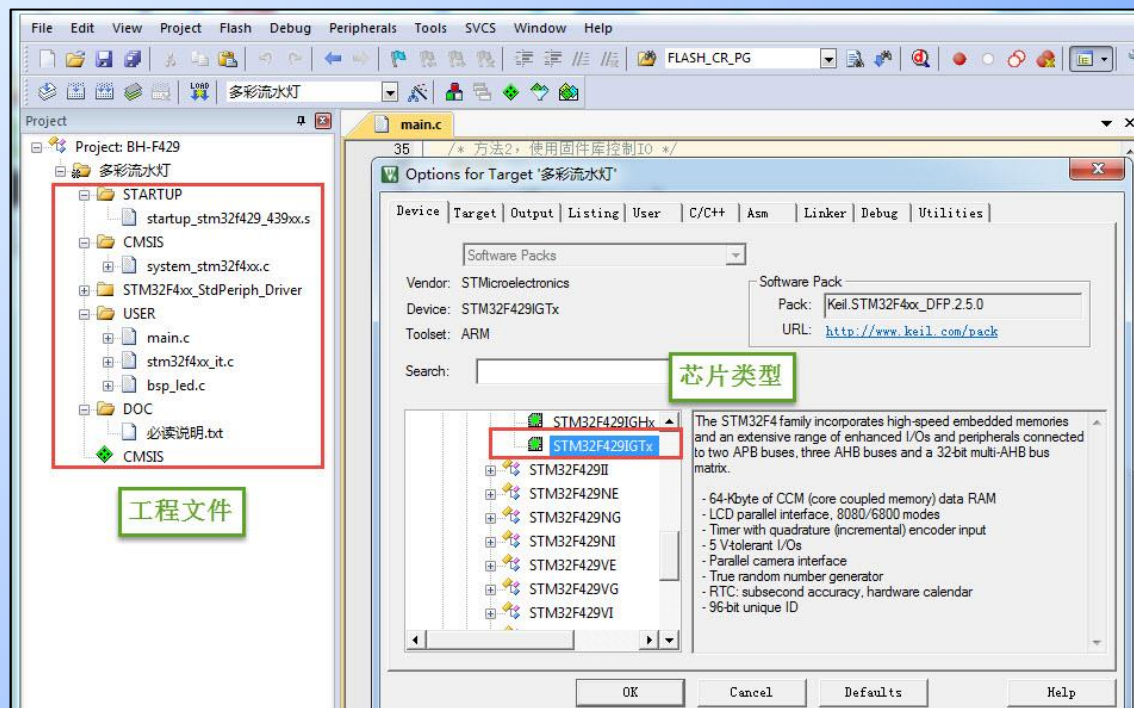
帮助(H)			
文件夹			
名称	修改日期	类型	大小
RTE	2015/11/18 19:18	文件夹	
BH-F429.uvguix.Administrator	2015/11/17 18:48	ADMINISTRATO...	140 KB
BH-F429.uvoptx	2015/11/2 21:42	UVOPTX 文件	28 KB
BH-F429.uvprojx	2015/11/2 21:42	vision5 Project	29 KB
JLinkSettings.ini	2014/11/20 10:05	配置设置	1 KB
WF-STM32F429.uvgui.Administrator	2015/7/31 11:45	ADMINISTRATO...	137 KB
WF-STM32F429.uvguix.Administrator	2015/10/31 14:08	ADMINISTRATO...	137 KB
WF-STM32F429.uvopt	2015/7/31 11:45	UVOPT 文件	30 KB
WF-STM32F429.uvoptx	2015/10/31 14:08	UVOPTX 文件	27 KB
WF-STM32F429.uvproj.saved_uv4	2015/1/8 10:49	SAVED_UV4 文件	26 KB

MDK的编译过程及文件类型全解



1.uvprojx文件

uvprojx文件就是我们平时双击打开的工程文件，它记录了整个工程的结构，如芯片类型、工程包含了哪些源文件等内容：



MDK的编译过程及文件类型全解



2.uvprojx文件

uvoptx文件记录了工程的配置选项，如下载器的类型、变量跟踪配置、断点位置以及当前已打开的文件等等：

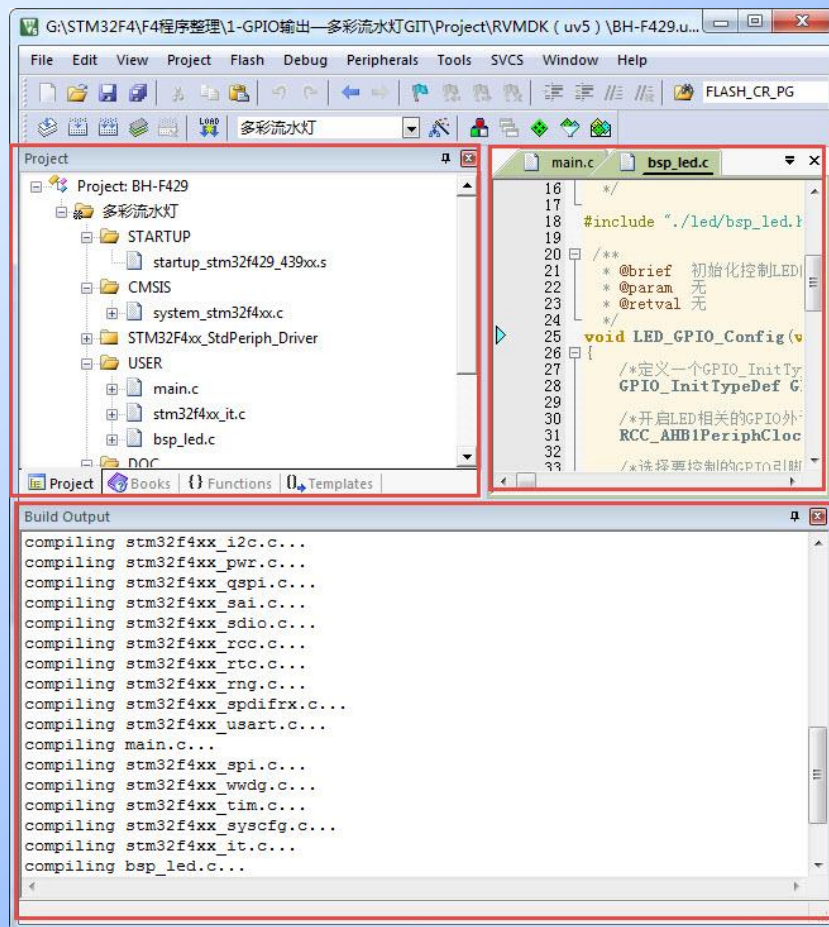
```
1 //**
2 *****
3 * @file      Pr
4 * @author    MC
5 * @version   V1.5.0
6 * @date      06-March-2015
7 * @brief     Main Interrupt Service Routines.
8 *            This file provides template for all excep
9 *            peripherals interrupt service routine.
10 *****
11 * @attention
12 *
13 * <h2><center>&copy; COPYRIGHT 2015 STMicroelectroni
14 *
15 * Licensed under MCD-ST Liberty SW License Agreement
16 * You may not use this file except in compliance wit
17 * You may obtain a copy of the License at:
```

MDK的编译过程及文件类型全解



3. uvguix文件

uvguix文件记录了MDK软件的GUI布局，如代码编辑区窗口的大小、编译输出提示窗口的位置等等。



MDK的编译过程及文件类型全解



uvprojx、uvoptx及uvguix都是使用XML格式记录的文件，若使用记事本打开可以看到XML代码。

而当使用MDK软件打开时，它根据这些文件的XML记录加载工程的各种参数，使得我们每次重新打开工程时，都能恢复上一次的工作环境。

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nam

<SchemaVersion>2.1</SchemaVersion>

<Header>### uVision Project, (C) Keil Software</Header>

<Targets>
  <Target>
    <TargetName>多彩流水灯</TargetName>
    <ToolsetNumber>0x4</ToolsetNumber>
    <ToolsetName>ARM-ADS</ToolsetName>
    <pCCUsed>5060020::V5.06 (build 20)::ARMCC</pCCUsed>
    <TargetOption>
      <TargetCommonOption>
        <Device>STM32F429IGTx</Device>
        <Vendor>STMicroelectronics</Vendor>
        <PackID>Keil.STM32F4xx_DFP.2.5.0</PackID>
        <PackURL>http://www.keil.com/pack</PackURL>
        <Cpu>IRCM(0x08000000,0x1000000) IRAM(0x20000000,0x300000) IRAM
        <FlashUtilSpec></FlashUtilSpec>
        <StartupFile></StartupFile>
        <FlashDriverDll>UL2CMS(-S0 -C0 -P0 -FD20000000 -FC1000 -FN
        <DeviceId>0</DeviceId>
        <RegisterFile>$$Device:STM32F429IGTx$Drivers\CMSIS\Device\T
        <MemoryEnv></MemoryEnv>
        <Cmp></Cmp>
        <Asm></Asm>
        <Linker></Linker>
        <OHString></OHString>
        <InfinionOptionDll></InfinionOptionDll>
        <SLE66CMisc></SLE66CMisc>
        <SLE66AMisc></SLE66AMisc>
        <SLE66LinkerMisc></SLE66LinkerMisc>
        <SFDFile>$$Device:STM32F429IGTx$CMSIS\SVD\STM32F429x.svd</S
        <CustSvd>0</CustSvd>
        <UseEnv>0</UseEnv>
        <BinPath></BinPath>
        <IncludePath></IncludePath>
        <LibPath></LibPath>
        <RegisterFilePath></RegisterFilePath>
      </TargetCommonOption>
    </TargetOption>
  </Target>
</Targets>

<Extensions>
  <Ext>*.c</Ext>
  <Ext>*.s*;</Ext>
  <Ext>*.obj</Ext>
  <Ext>*.lib</Ext>
  <Ext>*.txt;</Ext>
  <Ext>*.plm</Ext>
  <Ext>*.cpp</Ext>
  <Ext>*.nMigrate</Ext>
</Extensions>

<DveTm>
  <dwLowDateTime>0</dwLowDateTime>
  <dwHighDateTime>0</dwHighDateTime>
</DveTm>

<Target>
  <TargetName>多彩流水灯</TargetName>
  <ToolsetNumber>0x4</ToolsetNumber>
  <ToolsetName>ARM-ADS</ToolsetName>
  <TargetOption>
    <CLKADS>12000000</CLKADS>
    <OPTIT>
      <gFlags>1</gFlags>
      <BeepAtEnd>1</BeepAtEnd>
      <RunSim>0</RunSim>
      <RunTarget>1</RunTarget>
      <RunAbUc>0</RunAbUc>
    </OPTIT>
    <OPTRX>
      <HexSelection>1</HexSelection>
      <FlashByte>65535</FlashByte>
      <FlashRangeLowAddress>0</FlashRangeLowAddress>
      <FlashRangeHighAddress>0</FlashRangeHighAddress>
    </OPTRX>
  </TargetOption>
</Target>

<ViewPool>
  <View>
    <ViewName>System Viewer\GPIOH</ViewName>
    <UserString></UserString>
    <PropColWidth>136</PropColWidth>
  </View>
</ViewPool>

<SECTreeCtrl>
  <View>
    <WinId>38003</WinId>
    <ViewName>Registers</ViewName>
    <TableColWidths>115 115</TableColWidths>
  </View>
  <View>
    <WinId>346</WinId>
    <ViewName>Code Coverage</ViewName>
    <TableColWidths>1010 160</TableColWidths>
  </View>
  <View>
    <WinId>204</WinId>
    <ViewName>Performance Analyzer</ViewName>
    <TableColWidths>1170</TableColWidths>
  </View>
</SECTreeCtrl>

<TreeListPane>
  <View>
    <WinId>1506</WinId>
    <ViewName>Symbols</ViewName>
    <UserString></UserString>
    <TableColWidths>80 80</TableColWidths>
  </View>
  <View>
    <WinId>1506</WinId>
    <ViewName>Symbols</ViewName>
    <UserString></UserString>
    <TableColWidths>80 80</TableColWidths>
  </View>
</TreeListPane>
```

MDK的编译过程及文件类型全解



这些工程参数都是当MDK正常退出时才会被写入保存，所以若MDK错误退出时(如使用Windows的任务管理器强制关闭)，工程配置参数的最新更改是不会被记录的，重新打开工程时要再次配置。

根据这几个文件的记录类型，可以知道uvprojx文件是最重要的，删掉它我们就无法再正常打开工程了，而uvoptx及uvguix文件并不是必须的，可以删除，重新使用MDK打开uvprojx工程文件后，会以默认参数重新创建uvoptx及uvguix文件。(所以当使用Git/SVN等代码管理的时候，往往只保留uvprojx文件)

MDK的编译过程及文件类型全解



源文件

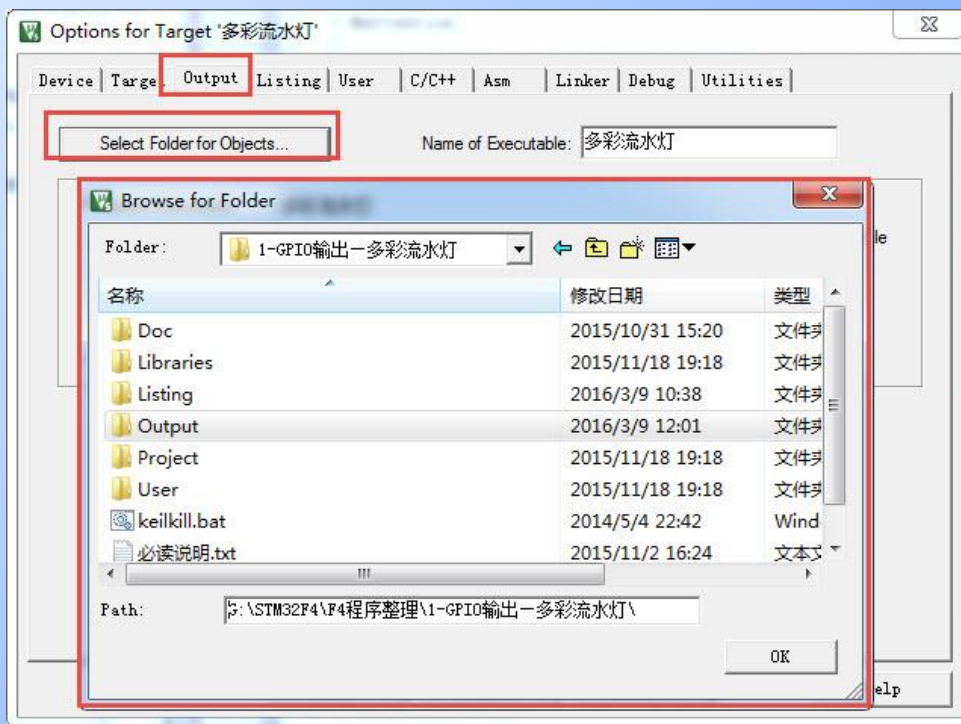
源文件是工程中我们最熟悉的内容了，它们就是我们编写的各种源代码，MDK支持c、cpp、h、s、inc类型的源代码文件，其中c、cpp分别是c/c++语言的源代码，h是它们的头文件，s是汇编文件，inc是汇编文件的头文件，可使用“`$include`”语法包含。编译器根据工程中的源文件最终生成机器码。

MDK的编译过程及文件类型全解



Output目录下生成的文件

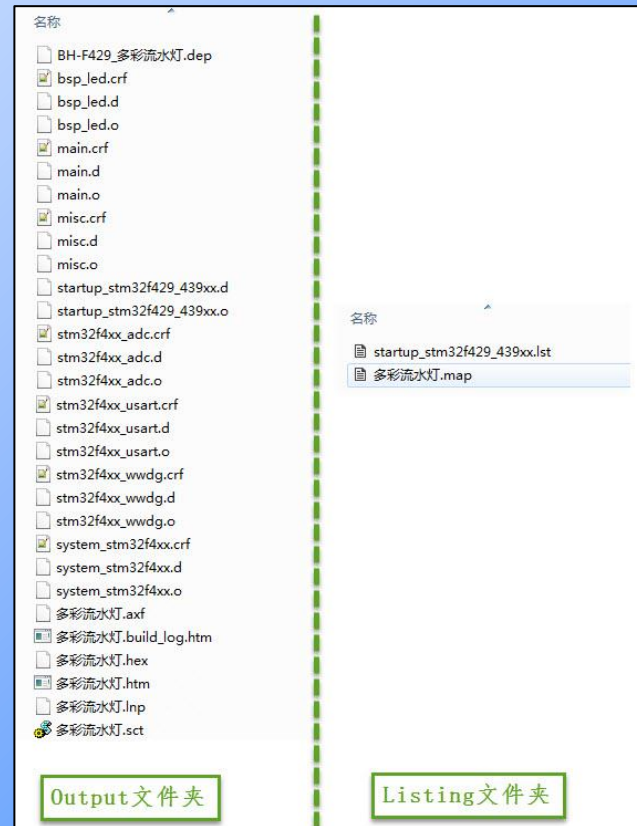
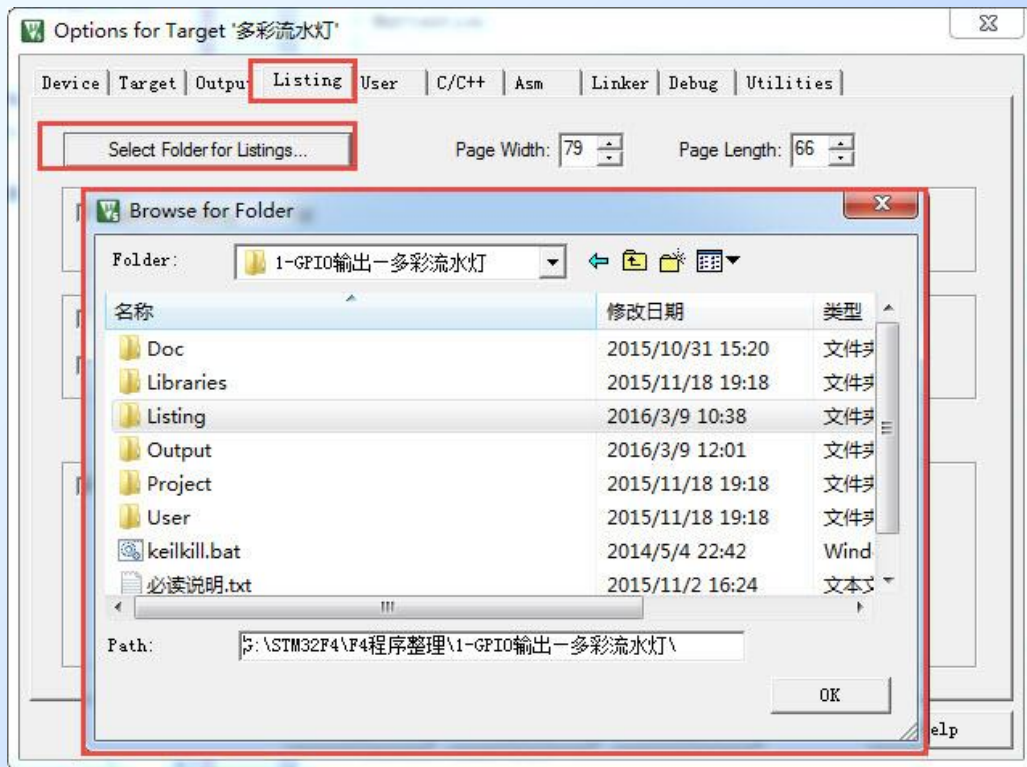
点击MDK中的编译按钮，它会根据工程的配置及工程中的源文件输出各种对象和列表文件，在工程的“Options for Target->Output->Select Folder for Objects”和“Options for Target->Listing->Select Folder for Listings”选项配置它们的输出路径：



MDK的编译过程及文件类型全解



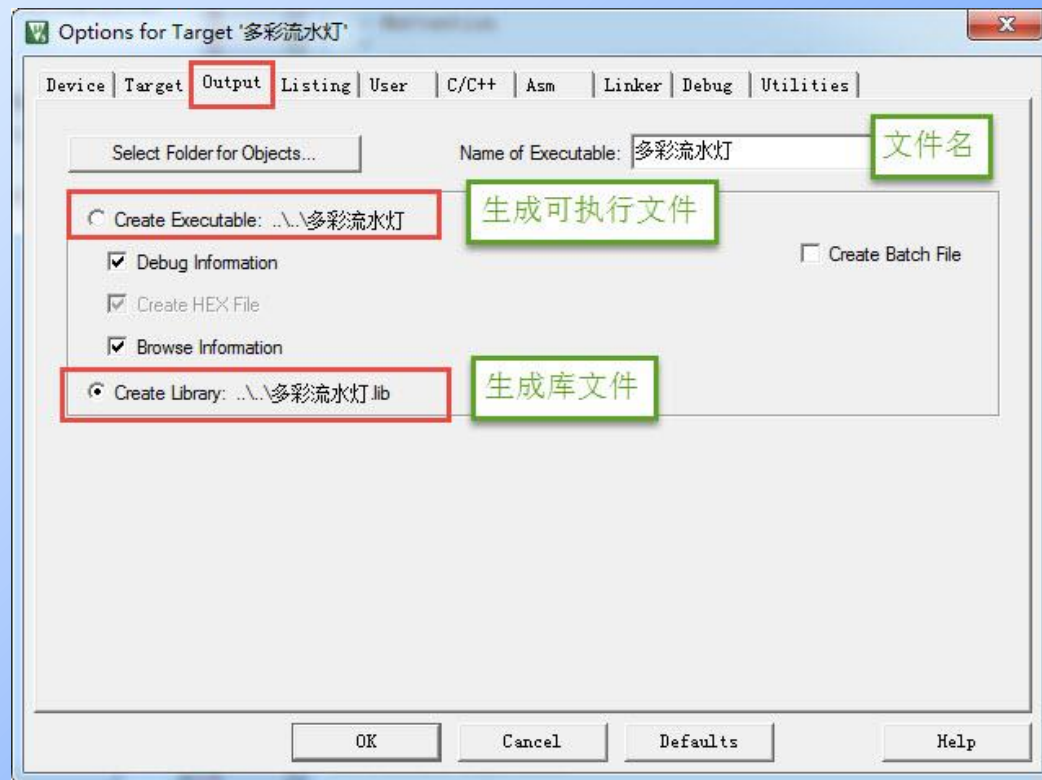
Output目录下生成的文件



MDK的编译过程及文件类型全解

1. lib库文件

在某些场合下可能不希望提供给第三方一个可用的代码库，但不希望对方看到源码，这个时候我们就可以把工程生成lib文件(**Library file**)提供给对方，在MDK中可配置“Options for Target->Create Library”选项把工程编译成库文件：



MDK的编译过程及文件类型全解



1. lib库文件

工程中生成可执行文件或库文件只能二选一，默认编译是生成可执行文件的，可执行文件即我们下载到芯片上直接运行的机器码。

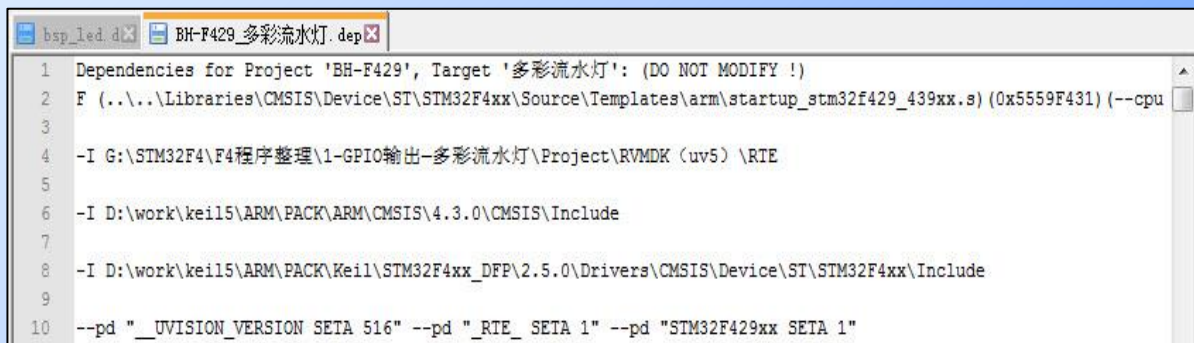
得到生成的*.lib文件后，可把它像C文件一样添加到其它工程中，并在该工程调用lib提供的函数接口，除了不能看到*.lib文件的源码，在应用方面它跟C源文件没有区别。

MDK的编译过程及文件类型全解

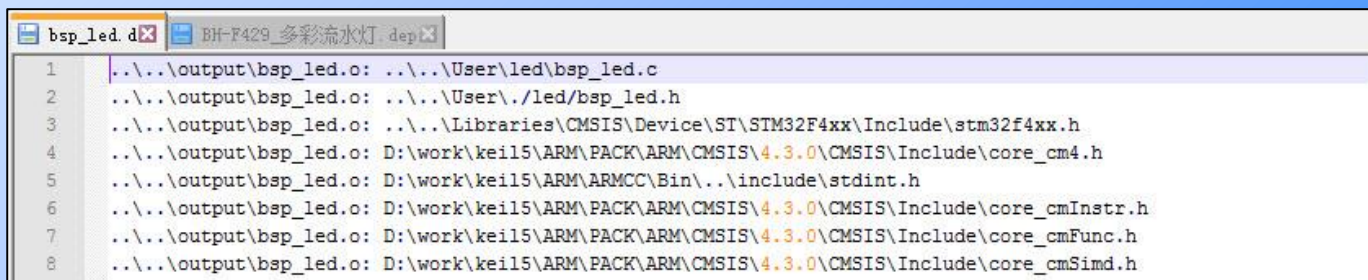


2.dep、d依赖文件

.dep和.d文件(Dependency file)记录的是工程或其它文件的依赖，主要记录了引用的头文件路径，其中*.dep是整个工程的依赖，它以工程名命名，而*.d是单个源文件的依赖，它们以对应的源文件名命名。这些记录使用文本格式存储，我们可直接使用记事本打开：



```
1 Dependencies for Project 'BH-F429', Target '多彩流水灯': (DO NOT MODIFY !)  
2 F (..\..\Libraries\CMSIS\Device\ST\STM32F4xx\Source\Templates\arm\startup_stm32f429_439xx.s) (0x5559F431) (--cpu  
3  
4 -I G:\STM32F4\F4程序整理\1-GPIO输出-多彩流水灯\Project\RVMDK (uv5) \RTE  
5  
6 -I D:\work\keil5\ARM\PACK\ARM\CMSIS\4.3.0\CMSIS\Include  
7  
8 -I D:\work\keil5\ARM\PACK\Keil\STM32F4xx_DFP\2.5.0\Drivers\CMSIS\Device\ST\STM32F4xx\Include  
9  
10 --pd "__UVISION_VERSION SETA 516" --pd "_RTE_ SETA 1" --pd "STM32F429xx SETA 1"
```



```
1 .....\output\bsp_led.o: .....\User\led\bsp_led.c  
2 .....\output\bsp_led.o: .....\User\./led/bsp_led.h  
3 .....\output\bsp_led.o: .....\Libraries\CMSIS\Device\ST\STM32F4xx\Include\stm32f4xx.h  
4 .....\output\bsp_led.o: D:\work\keil5\ARM\PACK\ARM\CMSIS\4.3.0\CMSIS\Include\core_cm4.h  
5 .....\output\bsp_led.o: D:\work\keil5\ARM\ARMCC\Bin\..\include\stdint.h  
6 .....\output\bsp_led.o: D:\work\keil5\ARM\PACK\ARM\CMSIS\4.3.0\CMSIS\Include\core_cmInstr.h  
7 .....\output\bsp_led.o: D:\work\keil5\ARM\PACK\ARM\CMSIS\4.3.0\CMSIS\Include\core_cmFunc.h  
8 .....\output\bsp_led.o: D:\work\keil5\ARM\PACK\ARM\CMSIS\4.3.0\CMSIS\Include\core_cmSimd.h
```

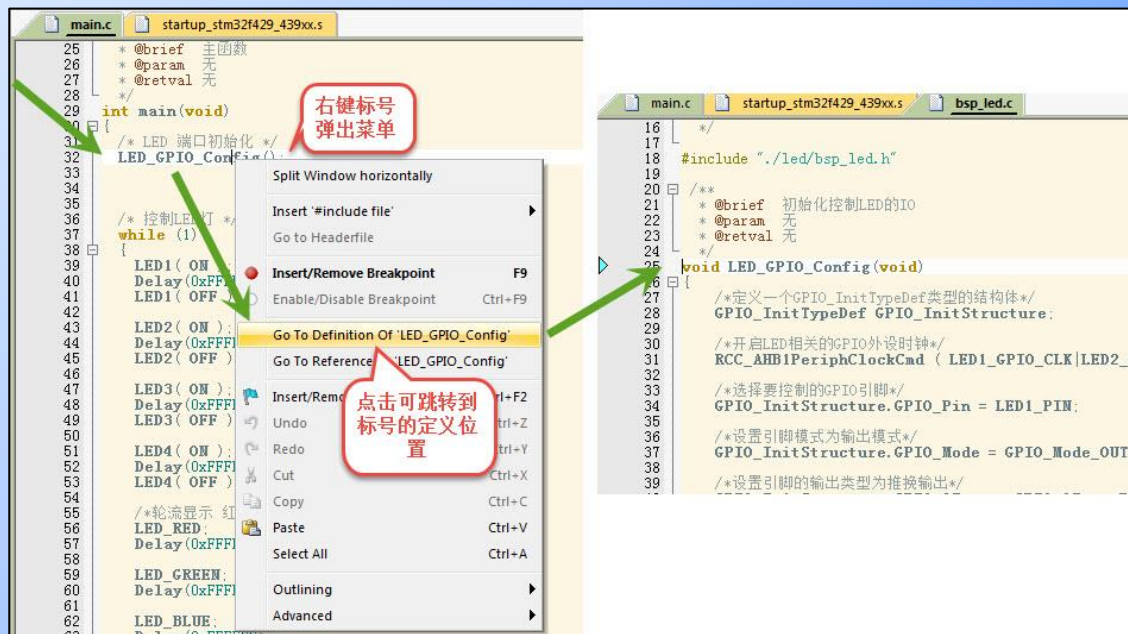

MDK的编译过程及文件类型全解



3.crf交叉引用文件

*.crf是交叉引用文件(Cross-Reference file)，它主要包含了浏览信息(browse information)，即源代码中的宏定义、变量及函数的定义和声明的位置。

我们在代码编辑器中点击“Go To Definition Of 'xxxx'”可实现浏览跳转，跳转的时候，MDK就是通过*.crf文件查找出跳转位置的。

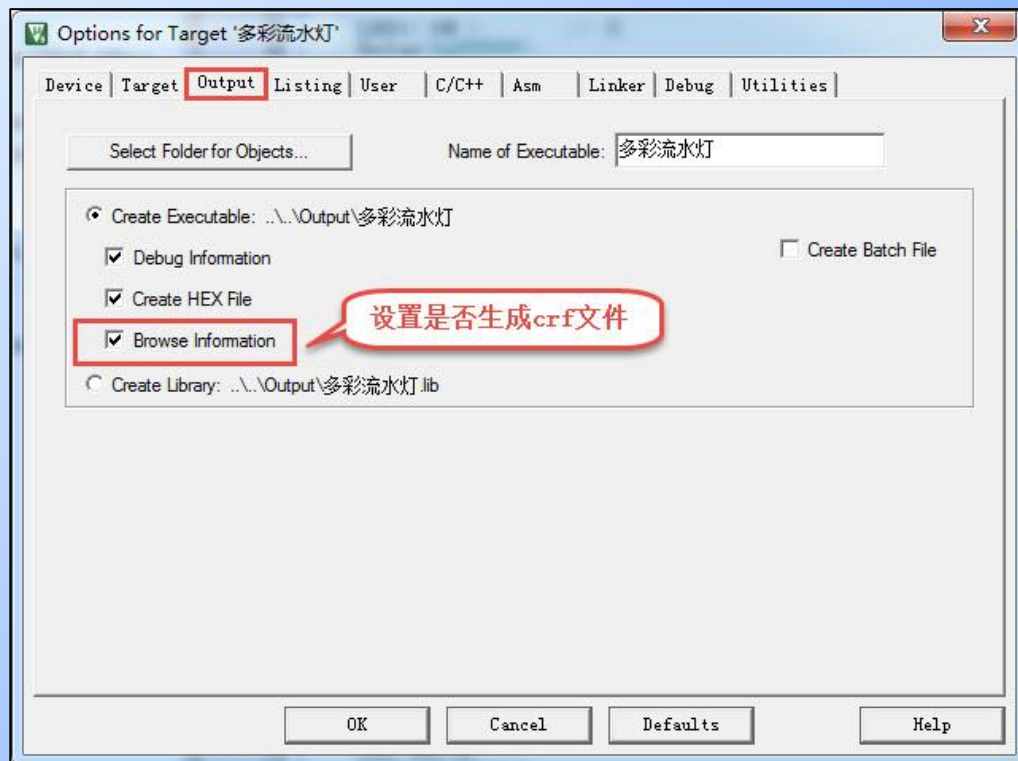


MDK的编译过程及文件类型全解



3.crf交叉引用文件

通过配置MDK中的“Option for Target->Output->Browse Information”选项可以设置编译时是否生成浏览信息，只有勾选该选项并编译后，才能实现上面的浏览跳转功能。



零死角玩转STM32



THANKS

论坛：www.chuxue123.com

淘宝：firestm32.taobao.com



扫描进入淘宝店铺