

SCONS 自动化构建工具入门教程

mousie-yu

2012-1-7

本教程是作者在学习使用 RT-Thread V1.0.0 版本时，对 SCONS 的入门级别的经验总结。
主要内容来自：

<http://www.rt-thread.org/dokuwiki/doku.php?id=rt-thread%E4%B8%AD%E7%9A%84scons%E4%BD%BF%E7%94%A8%E8%AF%A6%E8%A7%A3>

一、安装工具

需要安装 python 2.7.x 以及 scons 2.x。注意，并不推荐安装软件的最新版本。

下载链接：

Python: <http://www.python.org/ftp/python/2.7.1/python-2.7.1.msi>

SCONS: <http://www.rt-thread.org/download/scons-2.0.1.win32.exe>

必须先安装 python，再安装 scons。安装路径不建议使用中文和空格之类的符号。

采用默认安装即可，假设把 python 安装到了 D:\Python27.

则 scons 的默认安装路径为 D:\Python27\Scripts

为免去敲命令行的苦楚，推荐一个图形化工具 ArdaSConsUI，下载链接：

http://www.ourdev.cn/bbs/bbs_content.jsp?bbs_sn=4773281&bbs_id=3066&bbs_page_no=1

请自行安装编译工具。

以 RealView MDK 为例，安装路径必须是纯英文且不能带空格等符号。

建议直接安装到盘符根目录，如：

D:\Keil\

下载 RT-Thread V1.0.0，里面已经针对 scons 写好了脚本。

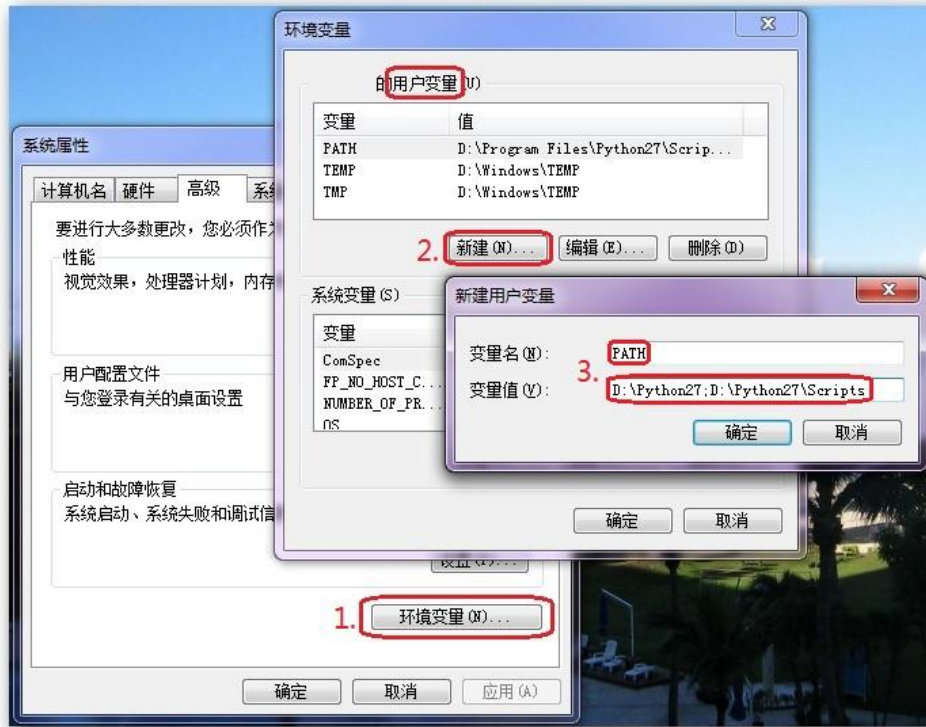
<http://rt-thread.googlecode.com/files/RT-Thread%201.0.0.7z>

至此，所有软件的准备工作完成了。

二、设置环境变量

为了能让系统识别 `scons` 指令，还需要设置一下工作路径。设置方式有多种，使用命令行指令 `path`；设置环境变量；通过注册表修改。此处介绍如何设置环境变量。

通过 我的电脑右键->属性->系统属性->高级，可以看到如下页面：



1. 在系统属性高级页面下，点击“环境变量”。
2. 在弹出的环境变量窗口下，在用户变量下点击“新建”。
3. 弹出新建用户变量窗口，输入如下信息：
变量名：PATH
变量值：D:\Python27; D:\Python27\Scripts
4. 一路确认到底即可完成设置。

三、使用 SCONS 编译 RT-Thread

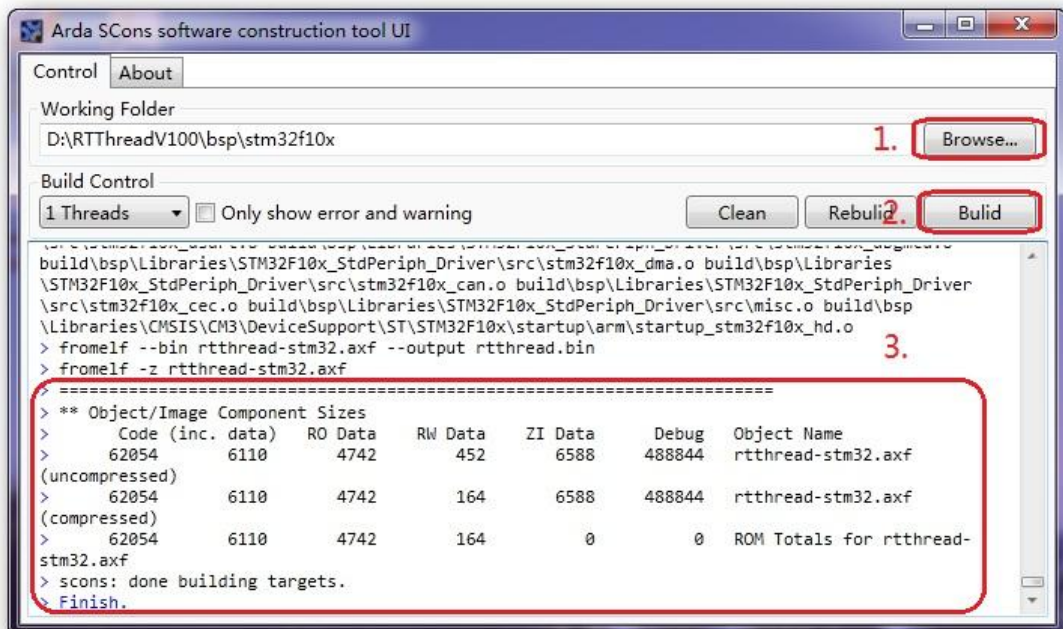
在安装完成 Python 和 SCons 后, 就可以使用 scons 编译 RT-Thread 了, 但是默认配置文件中并不知道你主机上安装的编译器位置, 所以在编译 RT-Thread 之前, 需要手动编辑 `rtconfig.py` 配置文件告诉 scons 编译器在哪里。用文本编辑器打开相应 bsp 目录下的 `rtconfig.py` 文件, 编辑其中的:

```
CROSS_TOOL='keil' # 这里需要选择你使用的编译器

# EXEC_PATH 指向了你编译器安装的位置
if CROSS_TOOL == 'gcc':
    PLATFORM      = 'gcc'
    EXEC_PATH      = 'D:/SourceryGCC/bin'
elif CROSS_TOOL == 'keil':
    PLATFORM      = 'armcc'
    EXEC_PATH      = 'D:/Keil'
elif CROSS_TOOL == 'iar':
    PLATFORM      = 'iar'
    IAR_PATH       = 'C:/Program Files/IAR Systems/Embedded Workbench 5.4
Evaluation_0'
```

作者用的是编译器 keil, 修改 keil 的工作路径为 “D:/Keil”, 注意路径符号用的是反斜杠 “/”, 而非 windows 的 “\”。

修改好上述信息后, 请出 Arda SConsUI 软件。使用起来非常方便。



1. 设置工作路径, 该路径同样不允许有中文和空格之类的符号存在。
2. 点击 Build。
3. 查看编译结果, 编译成功完成。

四、增减组件及生成 RVMDK 工程

增减组件非常简单。只需在 `rtconfig.h` 中宏定义某个组件。以文件系统为例：

在 `rtconfig.h` 中，宏定义 `RT_USING_DFS`。然后用 Arda SConsUI 软件编译即可。

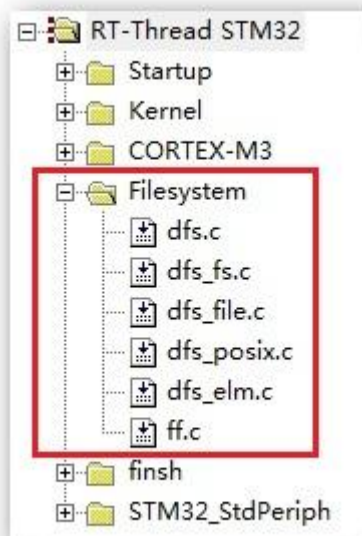
```
#define RT_USING_DFS // 使能文件系统
```

此时可能还需要使用 RVMDK 工程文件进行调试。使用 `scons` 可以自动生成 RVMDK 的工程文件。确保工程目录下包含一个名为 `template.Uv2` 或 `template.uvproj` 的模板工程。其中 `.Uv2` 是给 RVMDK V3.xx 版本用的，`.uvproj` 是给 RVMDK V4.xx 版本用的，当然，RVMDK V4.xx 是可以打开 `.Uv2` 工程并转换为 `.uvproj` 工程的。

运行 `cmd`，进入命令提示符窗口。



1. 进入工程目录。
2. 输入指令：`scons --target =mdk4 -s` 创建 RVMDK V4.xx 版本的工程。
如果需要 RVMDK V3.xx 版本的工程，输入：`scons --target =mdk -s`
3. 此时在工程目录下，可以看到 RVMDK 工程文件 `project.uvproj` 或 `project.Uv2`。



打开工程即可看到文件系统已加入。

五、Sconsstruct 和 Sconsscript

SConstruct 文件是 **scons** 所接受的编译脚本主文件，它包含了一些基本的变量信息。

如果要关联用户自定义组件，需要在 **Sconsstruct** 中增加内容。已 STM32 固件库为例，有如下内容：

```
# STM32 firemare library building script
objs = objs + SConscript(RTT_ROOT + '/bsp/stm3210/Libraries/SConscript',
variant_dir='bsp/Libraries', duplicate=0)
```

其含义为，让一个 **SConstruct** 文件再关联一个新的组件（用户自行定义的组件，而不是 RT-Thread 系统的组件，此处是 STM32 的固件库）。

Sconsscript 分散在各个组件目录中，可理解为组件脚本。以下的是一个脚本范例：

```
Import('RTT_ROOT')
from building import *

src = Glob('*.c')
CPPPATH = [RTT_ROOT + '/components/threads']
group = DefineGroup('threads', src, depend = ['RT_USING_THREADS'], CPPPATH =
CPPPATH)

Return('group')
```

注意：如果是在 **components** 目录下放置一个新目录，并附上类似上面的正确 **SConscript** 文件，那么在 RT-Thread 编译的过程中，**scons** 将会自动寻找这个组件，并检查它的依赖关系，如果依赖关系满足，将自行把这个组件添加到编译过程中。

可以加入的参数包括：

- **CCFLAGS** - C 源文件编译的参数；
- **CPPPATH** - 应该额外包含的头文件路径；
- **CPPDEFINES** - C 源文件编译时额外的宏定义；
- **LINKFLAGS** - 连接时应该添加的参数。

六、懒人必备之批处理

笔者是懒人，对于重复劳动这种吃力不讨好的事情要坚决杜绝!!!

于是使用批处理将整个过程简化到了极致。环境变量、ArdaSConsUI 都忽略掉吧，不需要他们了。只需将批处理放在工程目录下，双击运行即可！

批处理代码如下。保存为 .bat 格式即可。

```
@echo off
@Rem 设置 Python27 及 Scripts 的路径
set python=D:\Program Files\Python27
set scripts=D:\Program Files\Python27\Scripts

@Rem 设置编译器，mdk/mdk4/iar
set complier=mdk4

@Rem 设置编译方式，0 或 1 表示普通编译 >2 表示多核编译
set core=2

@Rem 将本批处理文件放到工程目录下点击运行即可
@Rem 注意：工程目录不可有“中文”“空格”等字符出现

@Rem 下面命令可自动实现编译及生成 RVMK
@Rem 加入 script 工作路径
path %path%;%python%;%scripts%
cd %~dp0

@Rem 编译或生成工程
:Start
echo .
echo .
echo .
echo *****
echo * author: mousie-yu *
echo * date: 2012.1.10 *
echo * brief: scons 自动化批处理 *
echo * path: %~dp0
echo *****

choice /c:bme /m Build-编译工程,Make-生成工程,Exit-退出
if errorlevel 3 goto End
if errorlevel 2 goto Make
if errorlevel 1 goto Build
```

:Build

call scons.bat -c

call scons.bat -j %core%

goto Start

:Make

call scons.bat --target=%complier% -s

goto Start

:End

exit