1 RT-Thread/LM3S 配置指南

1.1 LM3S 系列芯片概述

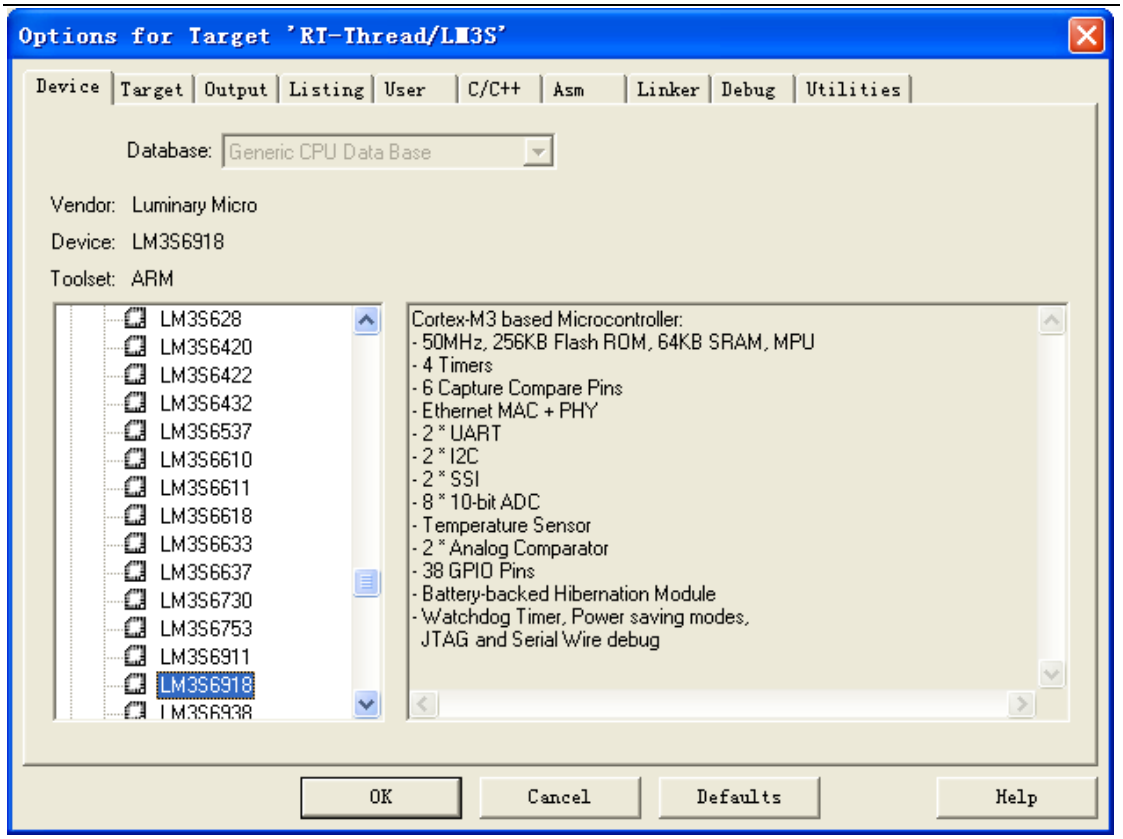
LM3S 系列芯片是美国德州仪器（TI）为汽车电子，工业控制，医疗电子等低成本嵌入式领域设计的 32 位计算能力的高性能芯片。
LM3S 平台包括 S100，S300，S600，S800，S1000，S2000，S3000，S5000，S6000，S8000，S9000 几大系列芯片，除 S100 和 S300 系列最大支持主频分别在 20MHz 和 25MHz 以外，其他所有系列芯片支持最大主频均为 50MHz。LM3S 系列芯片片内 FLASH 一般在 8KByte，16KByte，32KByte，64KByte，128KByte，256KByte，片内 SRAM 一般在 2Kbyte，8Kbyte，16Kbyte，32Kbyte，64Kbyte 不等。此外，S6000，S8000，S9000 系列集成了 100MHz 以太网控制器。

1.2 RT-Thread 的典型体积要求

下面看下 RT-Thread 几组典型的体积要求：
1）RT-Thread/LM3S 标准 Kernel（标准 Kernel 指得是没经过剪裁的内核）：
Program Size: Code=9150 RO-data=478 RW-data=120 ZI-data=1440
9.5K 只读数据和执行代码占用，1.5K 内存占用（通常只读数据和执行代码放置在闪存中）
从以上数据看出，RT-Thread/LM3S 标准 Kernel 可以运行在除 S100 系列以外的所有 LM3S 系列芯片中。
2）RT-Thread/LM3S 标准 Kernel + Lwip 协议栈
Program Size: Code=45654 RO-data=838 RW-data=308 ZI-data=6372
46K 只读数据和执行代码占用，7K 内存占用，另外，运行时会有 5K 左右动态内存占用。
从以上数据可以看出，RT-Thread/LM3S 标准 Kernel + Lwip 协议栈可以运行在所有集成以太网控制器的 LM3S 系列芯片上。
2）RT-Thread/LM3S 标准 Kernel + Lwip 协议栈 + File System + Finsh Shell + LM3S Lib
Program Size: Code=78826 RO-data=2298 RW-data=700 ZI-data=12972
标准 Kernel，Finsh Shell，文件系统，网络协议栈，需要 80K 只读数据和执行代码占用，13.5K 内存占用，当运行时，会有 5K 左右的动态内存占用。即如果芯片有 64K 的 SRAM，当系统运行时，大约剩余 45K 内存给用户使用。

1.3 LM3S 芯片类型的配置

当您使用 RT-Thread 时，请先确定您使用的芯片型号，在软件的配置上主要是选择芯片型号（在工程的选项中）：

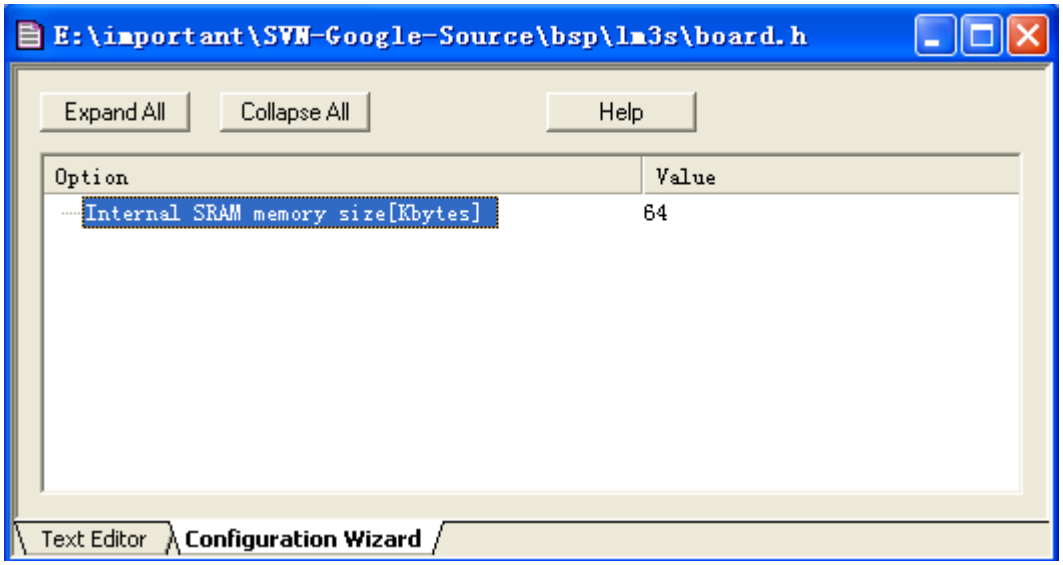


在上图中选择左边的芯片型号，例如 LM3S6918，STM32F6420 等。

1.4 STM32 开发板的配置

不同的 LM3S 芯片其差别体现在，片内 Flash 容量，片内 SRAM 容量，外设情况等。其中和操作系统密切相关的一个是，可用内存的多少。RT-Thread 针对 LM3S 系列芯片使用的是小型的内存管理算法，它需要知道它所能够使用、所能够管理的内存区域是哪一块区域（一段连续的地址范围）。

打开 RT-Thread 的 board.h 文件：



请在 Internal SRAM memory size[Kbytes)中填写芯片片内 SRAM 大小，单位是 Kbytes。例如 LM3S6918，填 64。
如果在 LM3S 系列芯片上，您只使用 RT-Thread 内核，那么到这里就算配置完成，可以使用了。

1.5 RT-Thread 组件的配置

RT-Thread 不仅仅是一个开源的实时操作系统，它也包括了周边的一些组件，主要包括几块：Finsh shell，TCP/IP 协议栈，文件系统，图形用户界面（RTGUI）。
图形用户界面暂时不包括在此次发布之内，所以这里主要着重介绍其他几个组件的配置。

1.5.1 Finsh shell 的配置

1.5.1.1Finsh shell 的使用与关闭

配置步骤：打开 rtconfig.h 文件：

```
069 /* SECTION: FinSH shell options */
070 /* Using FinSH as Shell*/
071 #define RT_USING_FINSH
072 /* Using symbol table */
073 #define FINSH_USING_SYMTAB
074 #define FINSH_USING_DESCRIPTION
```

使用网络模块功能时，将宏 RT_USING_LWIP 打开，如上图。

```
069 /* SECTION: FinSH shell options */
070 /* Using FinSH as Shell*/
071 /* #define RT_USING_FINSH */
072 /* Using symbol table */
073 /* #define FINSH_USING_SYMTAB */
074 /* #define FINSH_USING_DESCRIPTION */
```

关闭网络功能，将该宏注释掉。如上图：
该发布版本默认打开此宏。

1.5.2 TCP/IP 网络协议栈的配置

TCP/IP 网络协议栈需要实现的是网卡驱动，在 RT-Thread/LM3S 0.3.0 rc1 版本中已经提供该驱动，文件为 luminaryif.c，对应的是 LM3S 芯片内部集成的以太网卡。

1.5.2.1网络功能的使用与关闭

配置步骤：打开 rtconfig.h 文件，如下：

```
093 /* SECTION: lwip, a lighwight TCP/IP protocol stack */
094 /* Using lighweight TCP/IP protocol stack*/
095 #define RT_USING_LWIP
```

使用网络模块功能时，将宏 RT_USING_LWIP 打开。关闭网络功能，将该宏注释掉，即/* define RT_USING_LWIP */。该发布版本默认打开此宏。

1.5.2.2DHCP 功能的使用与关闭

配置步骤：打开 rtconfig.h 文件，如下：

```
123 /* Using DHCP*/
124 /* #define RT_LWIP_DHCP */
```

使用 DHCP 功能时，将宏 RT_LWIP_DHCP 打开。关闭网络功能，将该宏注释掉，即/* define RT_LWIP_DHCP */。该发布版本默认关闭此宏。

1.5.2.3网络地址的配置

网络地址的配置包括 IP 地址，网关地址，和子网掩码的配置，共有两种配置方式，一种是静态配置方式。配置步骤：打开 rtconfig.h 文件，如下：

```
128 /* ip address of target*/
129 #define RT_LWIP_IPADDR0 192
130 #define RT_LWIP_IPADDR1 168
131 #define RT_LWIP_IPADDR2 0
132 #define RT_LWIP_IPADDR3 30
```

默认 IP 地址为 192.168.0.30，需要在这里改成您需要设置的 IP 地址。

```
134 /* gateway address of target*/
135 #define RT_LWIP_GWADDR0 192
136 #define RT_LWIP_GWADDR1 168
137 #define RT_LWIP_GWADDR2 0
138 #define RT_LWIP_GWADDR3 1
```

默认网关地址为 192.168.0.30，需要在这里改成您需要设置的网关地址。

```
140 /* mask address of target*/
141 #define RT_LWIP_MSKADDR0    255
142 #define RT_LWIP_MSKADDR1    255
143 #define RT_LWIP_MSKADDR2    255
144 #define RT_LWIP_MSKADDR3    0
```

默认子网掩码为 192.168.0.30，需要在这里改成您需要设置的子网掩码。

第二种配置方式是在使用 Finsh Shell 的前提下，在系统运行时动态修改 IP 地址。如下图所示。

```
 \ | /
- RT - Thread Operating System
 / | \ 0.3.0 build Jan  3 2010
 2006 - 2009 Copyright by rt-thread team
part[0], begin: 122368, size: 477.392MB
finsh>>File System initialized!
TCP/IP initialized!

finsh>>set_if("192.168.0.30","192.168.0.1","255.255.255.0")
        536885396, 0x20003894
finsh>>
```

1.5.2.4网络应用程序

此外，我们已经开发和移植了一系列的网络应用程序，如 ftp server, tftp client, tftp server，chargen server 等，这部分代码不再另行发布，有兴趣的读者可以从我们的 SVN 中取得，SVN 地址是 http://code.google.com/p/rt-thread/

1.5.3 文件系统的配置

目前 RT-Thead/LM3S 主要支持的是 SD 卡，在发布版本中，实现了 SPI 接口的 SD 卡驱动，文件名为 sdcard.c。

1.5.3.1文件系统的使用与关闭

配置步骤：打开 rtconfig.h 文件，如下：

```
084 #define RT_USING_DFS
085 /* SECTION: DFS options */
```

使用文件系统功能时，将宏 RT_USING_DFS 打开。关闭文件系统功能，将该宏注释掉，即 /* define RT_USING_DFS */。该发布版本默认打开此宏。

1.5.4 其他更多配置

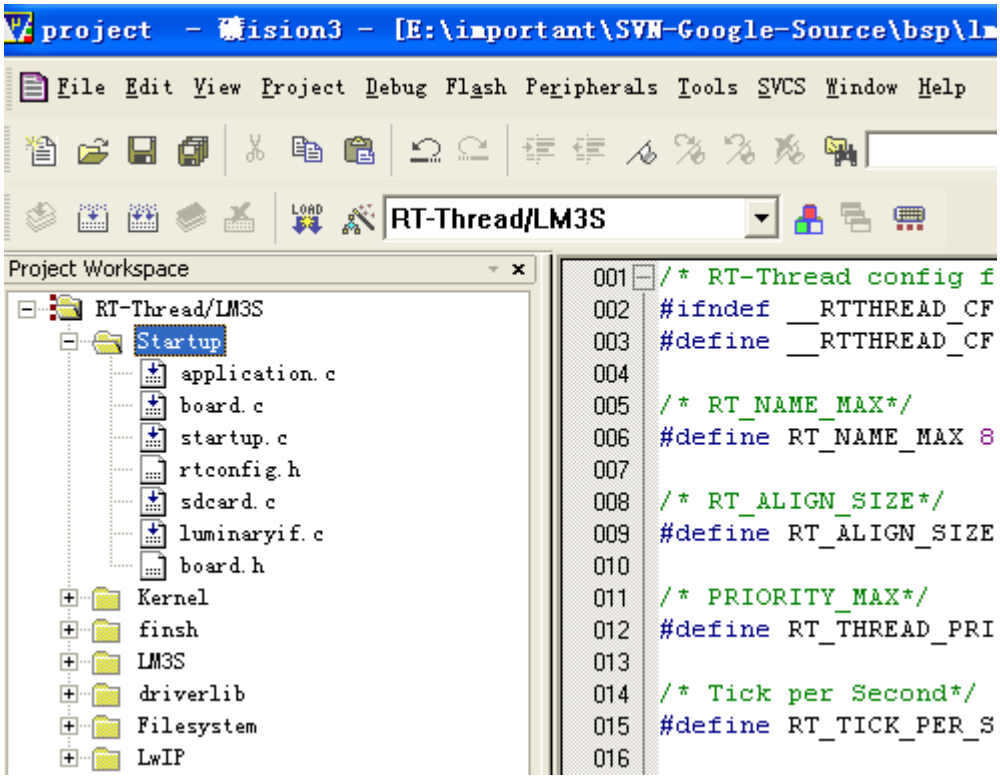
其他更多配置请参阅 RT-Thread 编程指南，或者直接登录论坛，http://www.rt-thread.org/phpbb 获得在线帮助。

2 RT-Thread/LM3S 编译调试环境

2.1 Keil MDK 开发环境

RT-Thread/LM3S 支持两种编译开发环境，第一种是 Keil MDK 集成开发环境，打开

RT-Thread/LM3S 0.3.0 rc1 版本软件包后，在 bsp\lm3s 目录下有个 project.Uv2 的文件，该文件是 Keil MDK 的项目文件，双击打开后会出现如下图所示的界面：



该项目文件涵盖了所有 RT-Thread/LM3S rc1 发布版本的所有功能，即包含了 RT-Thread Kernel 及其组件 Finsh Shell，File System，Lwip 协议栈，读者可以根据第一章 RT-Thread/LM3S 配置指南中讲述的配置方法来裁剪和修改 RT-Thread 及其组件。

2.2 GNU GCC 开发环境（scons 做为构建工具）

2.2.1 GNU GCC 开发环境搭建

工具准备：scons 构建工具需要 python2.5 和 scons1.2，GCC 编译工具需要 Sourcery G++
步骤一：安装 python 2.5
步骤二：安装 scons 1.2
步骤三：把 python2.5\scripts 加到你的执行目录中。具体方法如下：
打开 DOS 窗口，执行如下命令：set PATH=C:\python2.5\script;%PATH%
步骤四：安装 Sourcery G++

2.2.2 GNU GCC 开发环境下的配置和编译

2.2.2.1编译组件的配置

进入 rtt/bsp/lm3s 目录，打开 rtconfig.py 文件，如下图：

```
# component options
# finsh shell option
RT_USING_FINSH = True

# device file system options
RT_USING_DFS = True
RT_USING_DFS_EFSL = True
RT_USING_DFS_ELMFAT = False
RT_USING_DFS_YAFFS2 = False

# lwip options
RT_USING_LWIP = True

# rtgui options
RT_USING_RTGUI = False
```

RT_USING_FINSH = True 表示使用 Finsh 组件，RT_USING_FINSH = False 表示关闭 Finsh 组件，通过这些预编译开关来决定是否使用特定组件。默认配置使用 Finsh shell, File System, Lwip 协议栈组件。需要注意的是，在选择使用特定组件时，rtconfig.h 配置文件也应当被同步更新。

2.2.2.2编译器的配置

进入 rtt/bsp/lm3s 目录，打开 rtconfig.py 文件，如下图：

```
# toolchains options
ARCH='arm'
CPU='lm3s'
#PLATFORM = 'gcc'
#EXEC_PATH = 'E:/Program Files/CodeSourcery/Sourcery G++ Lite/bin'
PLATFORM = 'armcc'
EXEC_PATH = 'E:/Keil'
```

在此选择编译器环境及其在本地机器中的路径，默认使用 armcc 编译器，允许更改为 gcc 编译器。

2.2.2.3编译模式的配置

进入 rtt/bsp/lm3s 目录，打开 rtconfig.py 文件，如下图：

```
BUILD = 'release'
```

可以设置该项为 release 或者 debug 模式。

2.2.2.4开始编译

配置完毕后，打开 DOS 窗口，进入 rtt/bsp/lm3s 目录，执行 scons 命令，便开始编译，直到最后生成目标文件。

```
ject/Image Component Sizes

 Code (inc. data)   RO Data    RW Data    ZI Data      Debug

31922      11698       2634        964      13188      25174
31922      11698       2634        256      13188      25174
31922      11698       2634        256          0          0
: done building targets.
```