Entry manual

ARM Cortex-M0 LPC1114 入门手册

V1.0 10/29/2013 瑞生电子

版本	日期	说明
V1.0	2013/10/29	首次发布

瑞生电子官网：http://www.rationmcu.com

Ration

第一章 简介

LPC1114 是 NXP 公司推出的一款 ARM Cortex-M0 内核的 32 位单片机。它的主频最大可达 50MHz，内部集成时钟产生单元，不用外部晶振也可以工作。内部集成 32KB FALSH 程序存储器、8K SRAM 数据存储器、一个快速 I2C 接口、一个 RS485/EIA485 UART、两个带 SSP 特征的 SPI 接口、4 个通用定时器、1 个系统定时器、1 个带窗口功能的看门狗定时器、功耗管理模块、1 个 ADC 模块和 42 个 GPIO。截至 Ration 写稿时，一片 LPC1114 的零售价只需 5.9 元，批量价更便宜。如此强大的处理器，如此低廉的价格，可谓是性价比无敌，其低功耗、简单易用、高能效和低成本相结合，必然会在市场中占有一席之地。

LPC1114 是 ARM 入门级的单片机，使用起来非常简单，只要会 51 单片机，就可以快速的使用 LPC1114。幸运的是，即使你不会 51 单片机，Ration 也可以带领你彻底征服这个看似复杂实则简单的单片机。

不管是什么单片机，本质上都一样，对外表现为 N 个引脚，用引脚的高低电平变化来完成各种控制通信工作。内部由若干个功能模块构成，例如串口模块、ADC 模块等，有些单片机集成的功能模块相对较多，有些单片机集成的功能模块相对较少。我们要学习的，即如何配置单片机内部的各个模块，来完成我们所需要的目的。

不管是学习单片机，还是学习其它与单片机配合的其它硬件，学习方法都一样。从大局上看，它们都是由外部引脚和内部功能模块构成的。内部功能模块会有一些寄存器，我们了解了它的每个寄存器的功能，就可以通过它的用户手册配置寄存器，达到所需的要求。

例如：给 51 单片机中的寄存器 P1 写 0x01，将会使得引脚 P1.0 电平为高，P1.1~P1.7 引脚为低。给 51 单片机中的寄存器 TMOD 写 0x20，将会配置定时器 0 为 16 位模式，定时器 1 为 8 位自动重载模式。

从学习角度讲，LPC1114 与普通 51 单片机的主要区别：

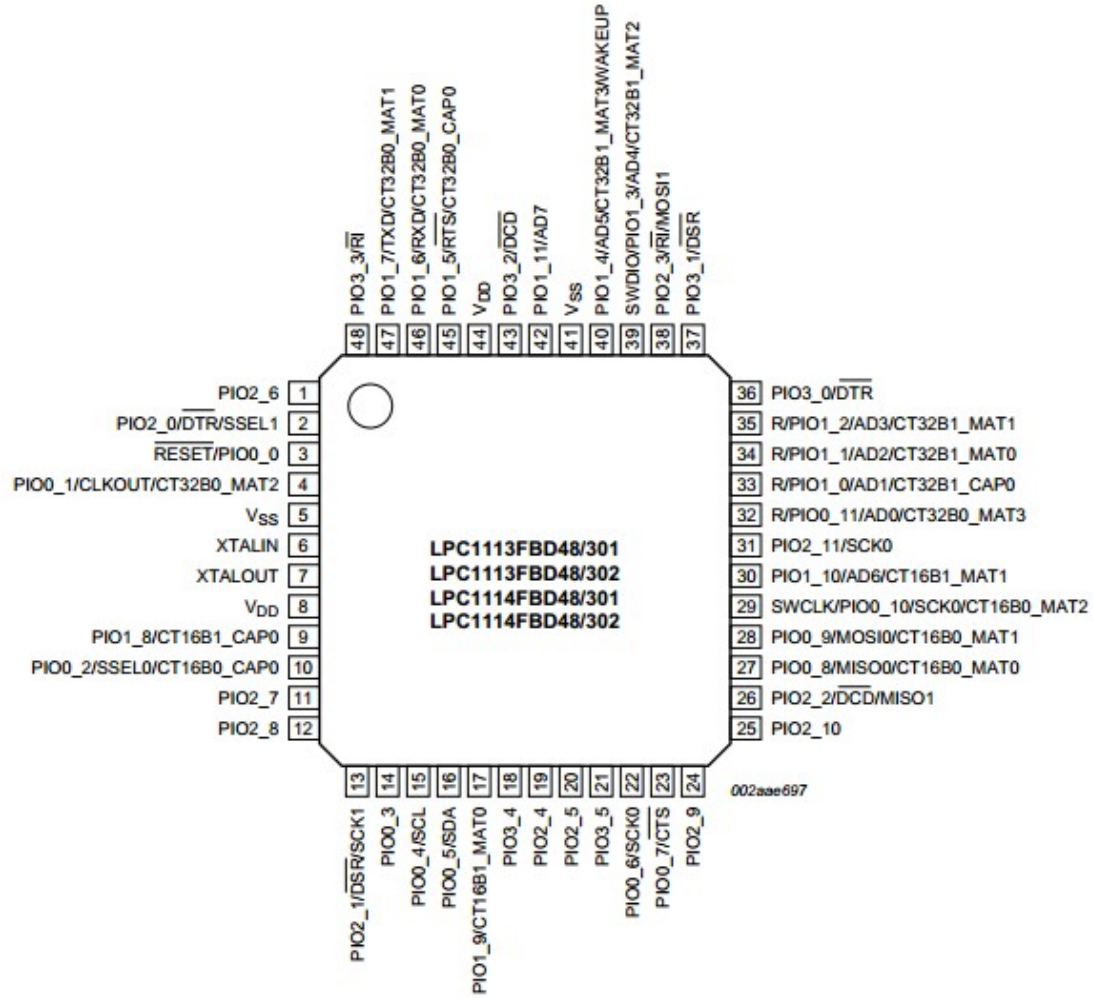
1. LPC1114 寄存器是 32 位的，普通 51 单片机寄存器是 8 位的
2. LPC1114 内部功能模块比普通 51 单片机多

只看到上述两点区别，你对 LPC1114 学习的压力是否减轻许多！

第二章 初识 LPC1114

2.1 观引脚识功能

我们以 LQFP48 封装为例进行介绍。



从图中引脚上的描述可以看出，它的几乎每一个引脚上都复用了若干个功能。例如，第 9 脚：PIO1_8/CT16B1_CAP0，代表，第 9 脚既可以作为通用的输入输出引脚 P1.8，也可以作为 16 位定时器 1 的捕获引脚。（关于什么是捕获引脚，请看 Ration 的《RATION LPC1114 基础篇手册》）。

引脚作为什么功能，需要通过 IOCON 模块来配置。

现在，让我们把所有的引脚描述都看一遍吧！看完了引脚描述，你就会对它有一个基本的认识了。

GPIO 模块引脚：

PIO0_0~PIO0~11

PIO1_0~PIO1~11

PIO2_0~PIO2_11

PIO3_0~PIO3_5

P0 口，P1 口，P2 口各有 12 个引脚，P3 口有 6 个引脚，一共 42 个 GPIO 口。

电源引脚：(3.3V 供电)

PIN5: V_{SS}

PIN8: V_{DD}

PIN41：V_{SS}

PIN44：V_{DD}

时钟振荡器引脚：

PIN6：XTALIN

PIN7：XTALOUT

接外部晶振。

时钟输出引脚：

PIN4：CLKOUT

复位引脚：

PIN3：RESET

I2C 模块引脚：

PIN15：SCL

PIN16：SDA

UART 串口引脚：

PIN46：RXD 串行数据输入引脚

PIN47：TXD 串行数据输出引脚

除了 RXD 和 TXD 引脚，还有 9 针全功能串口中的其它握手信号引脚。

PIN45： RTS

PIN2、PIN36：DTR

PIN23： CTS

PIN13\PIN37：DSR

PIN26\PIN43：DCD

PIN38\PIN48：RI

其中，DTR、DSR、DCD、RI 引脚复用到了两个引脚上，可以根据实际需要选择一个引脚作为对应功能。

SPI 模块引脚：（LPC1114 内部有 2 个 SPI 模块，分别用 SPI0 和 SPI1 表示）

PIN10： SSEL0

PIN27： MISO0

PIN28： MOSI0

PIN22\PIN29：SCK0

PIN2： SSEL1

PIN26： MISO1

PIN38： MOSI1

PIN13： SCK1

SCK0 复用到了两个引脚上，可以通过 IOCON_LOC 寄存器配置到其中一个引脚上。

ADC 模块引脚：

PIN32：AD0

PIN33：AD1

PIN34：AD2

PIN35：AD3

PIN39：AD4

PIN40：AD5

PIN30：AD6

PIN42：AD7

LPC1114 内部有一个 ADC 模块，可以通过 8 个引脚采集模拟信号。

通用定时器模块引脚：(共有 4 个定时器，2 个 16 位定时器，2 个 32 位定时器)

PIN10：CT16B0CAP0
PIN27：CT16B0MAT0
PIN28：CT16B0MAT1
PIN29：CT16B0MAT2
“16 位定时器 0”有一个捕获引脚，3 个匹配输出引脚。

PIN9：CT16B1CAP0
PIN17：CT16B1MAT0
PIN30：CT16B1MAT1
“16 位定时器 1”有一个捕获引脚，2 个匹配输出引脚。

PIN45：CT32B0CAP0
PIN46：CT32B0MAT0
PIN47：CT32B0MAT1
PIN4：CT32B0MAT2
PIN32：CT32B0MAT3
“32 位定时器 0”有一个捕获引脚，4 个匹配输出引脚。

PIN33：CT32B1CAP0
PIN34：CT32B1MAT0
PIN35：CT32B1MAT1
PIN39：CT32B1MAT2
PIN40：CT32B1MAT3
“32 位定时器 1”有一个捕获引脚，4 个匹配输出引脚

捕获引脚可以计数，可以测频率，类似于普通的中断引脚。
匹配输出引脚可以输出 PWM 脉宽调制信号。

SWD 调试模块引脚：
PIN29：SWCLK
PIN39：SWDIO

上面，把所有的引脚名称都归了一下类。上面提到的各个功能模块都是对外表现出引脚的，在 LPC1114 内部，还有没有对外表现出引脚的模块，例如功耗管理模块，看门狗模块等。

2.2 寄存器配置

32 位的单片机内部各种数据寄存器和控制寄存器都是 32 位的，同理，8 位单片机内部的数据和控制寄存器都是 8 位的。

例如:

AT89C51 单片机的“中断控制寄存器”IE 定义如下图所示：

bit7	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
EA	保留	ET2	ES	ET1	EX1	ET0	EX0

LPC1114 的“AHB 总线时钟控制寄存器”SYSAHBCLKCTRL 定义如下图所示：

bit31	**bit30**	**bit29**	**bit28**	**bit27**	**bit26**	**bit25**	**bit24**
保留	保留	保留	保留	保留	保留	保留	保留
bit23	**bit22**	**bit21**	**bit20**	**bit19**	**bit18**	**bit17**	**bit16**
保留	保留	保留	保留	保留	SSP1	保留	IOCON
bit15	**bit14**	**bit13**	**bit12**	**bit11**	**bit10**	**bit9**	**bit8**
WDT	保留	ADC	UART	SSP0	CT32B1	CT32B0	CT16B1
bit7	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
CT16B0	GPIO	I2C	FLASH2	FLASH1	RAM	ROM	SYS

以上两个控制寄存器，一个是 8 位的，一个是 32 位的，它们的相同之处都是每一位决定了一项任务。例如，给 AT89C51 单片机的 IE 寄存器的 bit4 写 1 可以打开串口中断，写 0 可以关闭串口中断。给 LPC1114 单片机的 AHBCLKCTRL 寄存器的 bit6 写 1 表示打开 GPIO 的工作时钟，写 0 表示关闭 GPIO 的工作时钟。

上面所讲的 IE 寄存器，可以用 IE=0x80 开启总中断，也可以直接写 EA=1 开启总中断。用 EA=1 来开启的方式就是“位操作”。“位操作”与直接写寄存器

值相比，直接写寄存器将会改变整个寄存器的值，而“位操作”不会改变寄存器中的其它值。

LPC1114 单片机的寄存器不支持“位操作”，为了使得操作某位的同时，不影响其它位的值，我们需要运用一下 C 语言的逻辑“或”“与”操作。

例如：

对 SYSAHBCLKCTRL 寄存器的 bit6 写 1：

```
LPC_SYSCON->SYSAHBCLKCTRL |= (1<<6);
```

对 SYSAHBCLKCTRL 寄存器的 bit6 写 0：

```
LPC_SYSCON->SYSAHBCLKCTRL &= ~(1<<6);
```

在头文件 lpc11xx.h 中，各个寄存器是由各个模块的结构体定义的，所以我们要给某个寄存器写值的时候，要用到给结构体成员变量赋值符号“->”。上式中，SYSAHBCLKCTTL 寄存器位于结构体 LPC_SYSCON，所以给寄存器赋值的时候，要这么写。

1<<6 就是 1 向左移 6 下的意思，即：

32 位数 1 用二进制表示 00000000000000000000000000000001

32 位数 1 左右 6 下以后为 00000000000000000000000001000000

把这个左移好的数据与 SYSAHBCLKCTRL 中的值“或”一下，“或”的逻辑为 0“或”任何数都是任何数，1“或”任何数都是 1，所以结果只把 bit6 置 1.

同理，可以分析一下给 bit6 写 0 的语句。都是 C 语言的基础知识。请相信，高手并不是拥有了特殊的技能，而是掌握了扎实的基础。

特别提示：上述两条写寄存器的语句，初看有些复杂，实则简单至极！当我们以后要给某个寄存器的某个位写值的时候，例如,要给 AA 模块的 BB 寄存器的 bit n 写 1，套用上式，即：

```
AA->BB |=(1<<n);
```

同理，要给 bit n 写 0，即：

```
AA->BB&=~(1<<n)
```

亲！恭喜你，你已经学会一大半了。

第三章 LPC1114 开发环境建立与使用

3.1 安装开发软件

支持 LPC1114 的开发软件有 KEIL、IAR、TKStudio、LPCXPresso、CoIDE 等。这里我们选用最长用的 KEIL 为例进行讲解，版本号 4.60 评估版。

MDK 4.60 评估版代码限制 32K，对于 LPC1114 来说，就是一款免费的开发工具了。

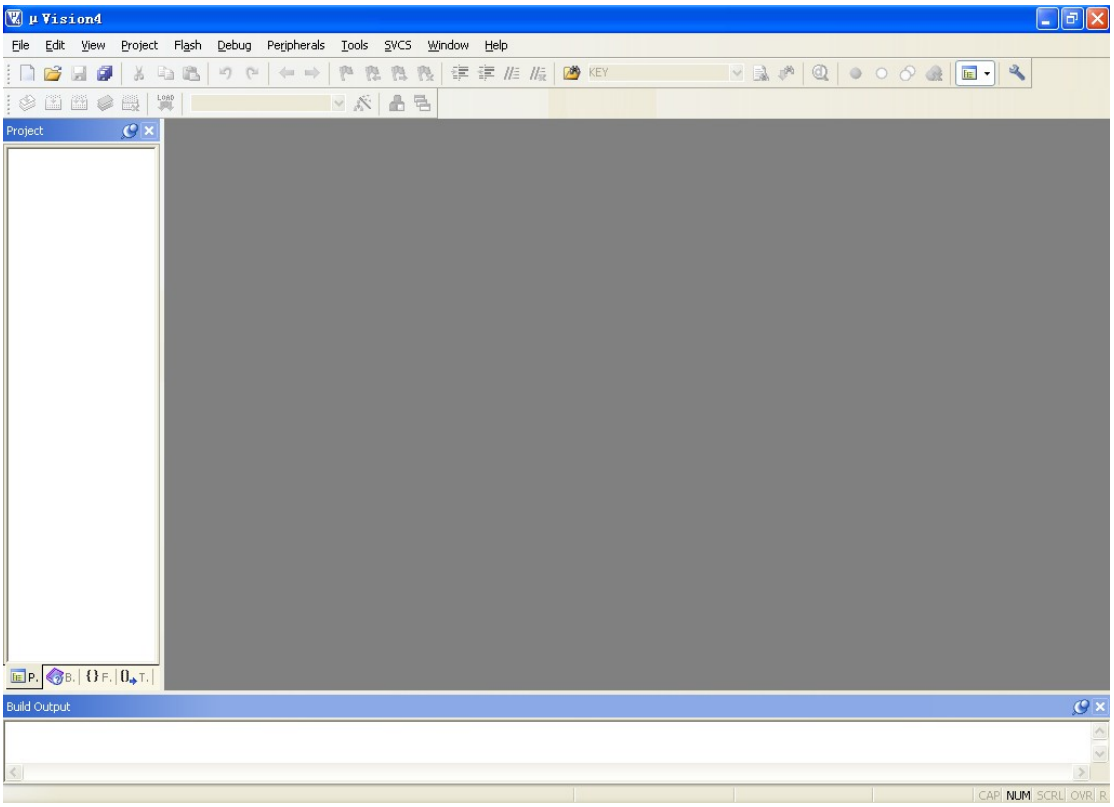
MDK460.exe 文件，自身大小 487M，安装到硬盘，占用 4.77G 硬盘空间。
系统：WINDOWS XP、WIN7 32 位、WIN7 64 位系统均可使用
内存：建议 2G 以上
CPU：建议双核以上

双击打开安装程序，和大多数的 WINDOWS 程序类似，一路点击 NEXT 即可安装完成。安装时间，由 CPU 速度决定，普通的电脑配置，大概需要 8 分钟的时间，所以建议电脑配置越高越好。

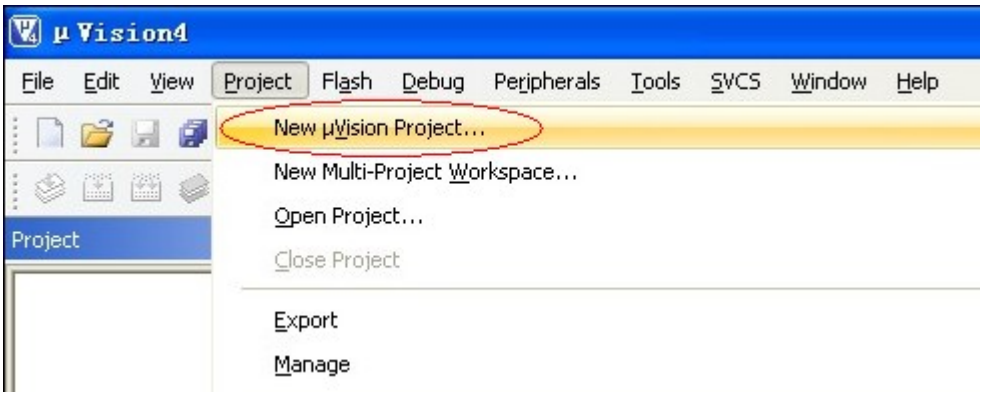
完成后，会在桌面上产生一个图标。

3.2 新建一个 LPC1114 工程

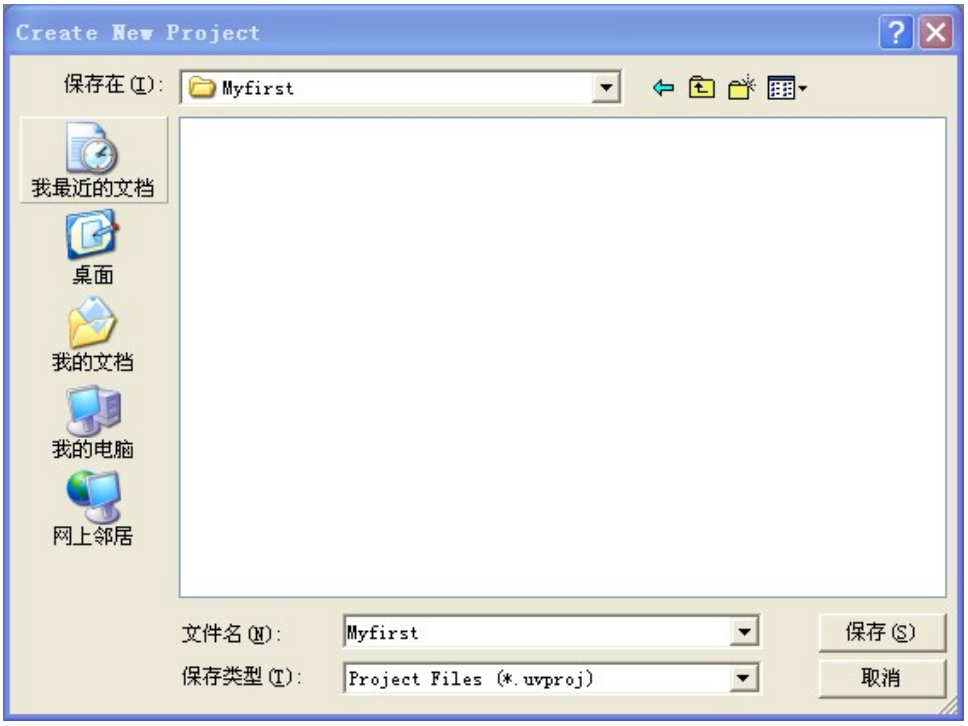
双击桌面快捷方式，打开 KEIL，如下图所示：



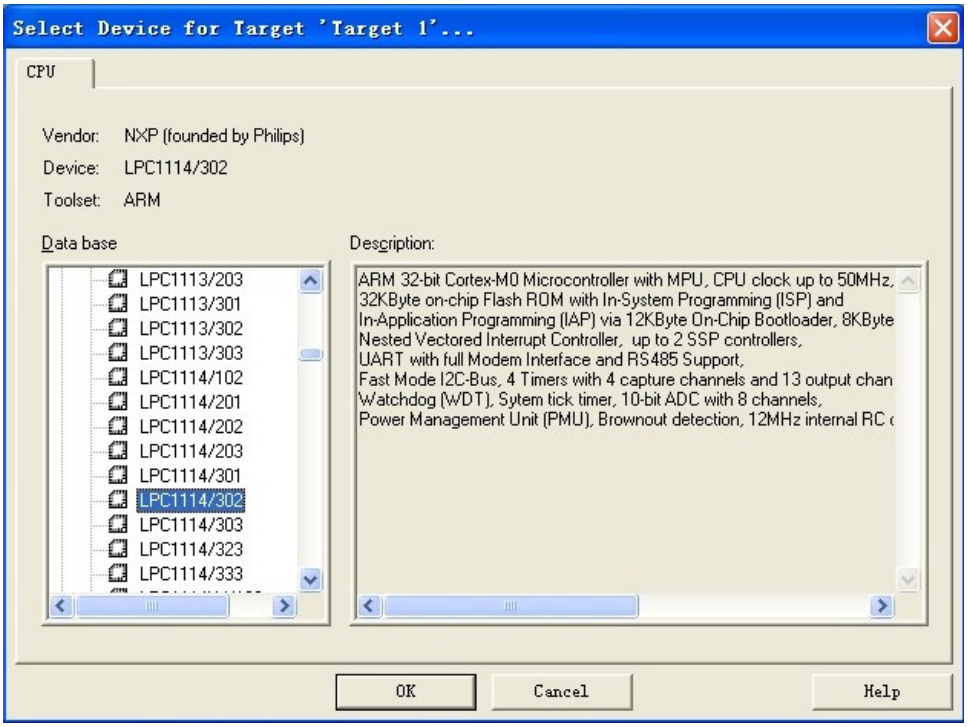
单击菜单“Project”,在下拉菜单中选择“New µVision Project”



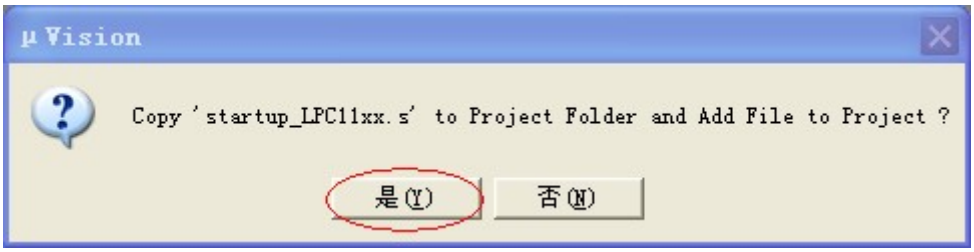
在弹出的窗口中，选择工程保存路径，例如把工程保存到 E 盘的 Myfirst 文件夹，并把工程名命名为 Myfirst。如下图所示：



点击“保存”按钮，弹出另一个窗口，选择 NXP 公司的 LPC1114/302，如下图所示：

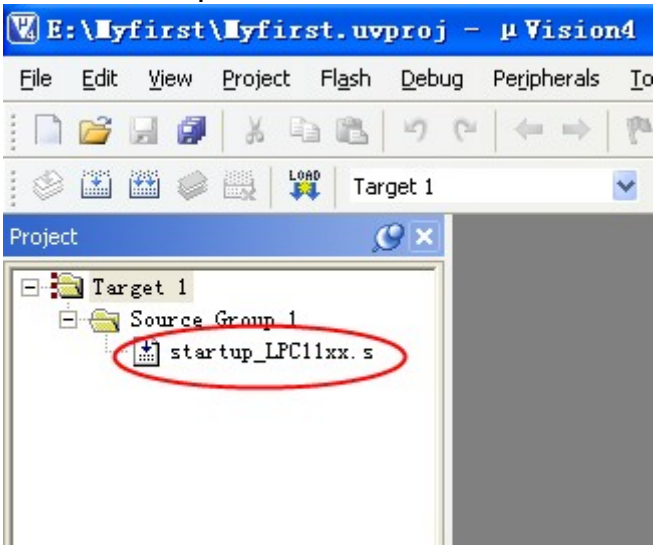


在上图中右边的文本框里面是 LPC1114/302 芯片的介绍。单击“OK”按钮，会弹出一个窗口，询问是否加载 startup_LPC11xx.s 文件到工程文件夹。这里我们选择“是”。

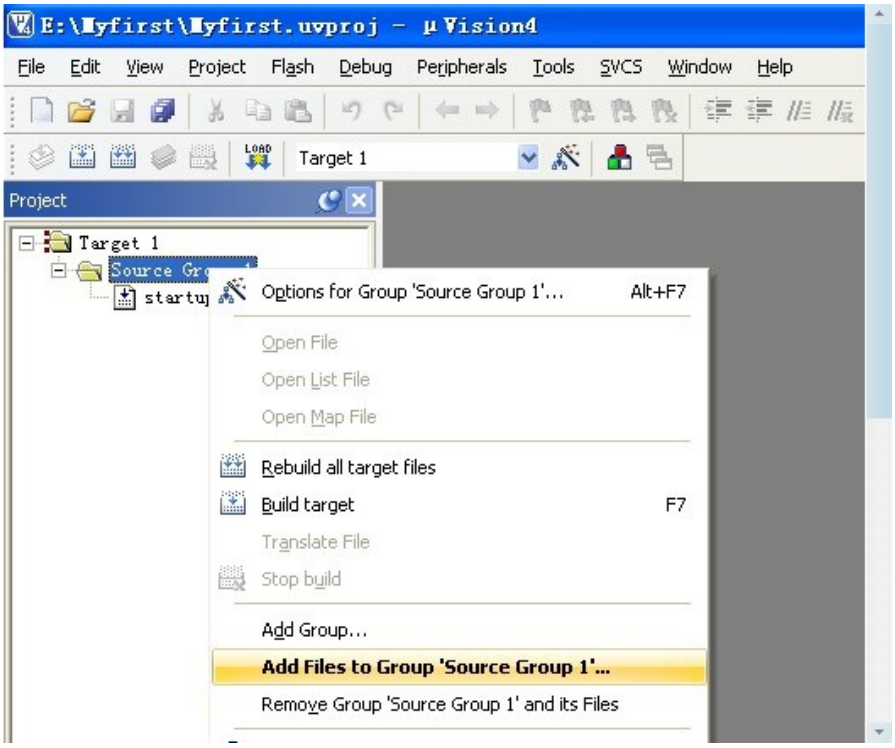


此时，在 Project 窗口中，会显示如下图所示的 Target 1。

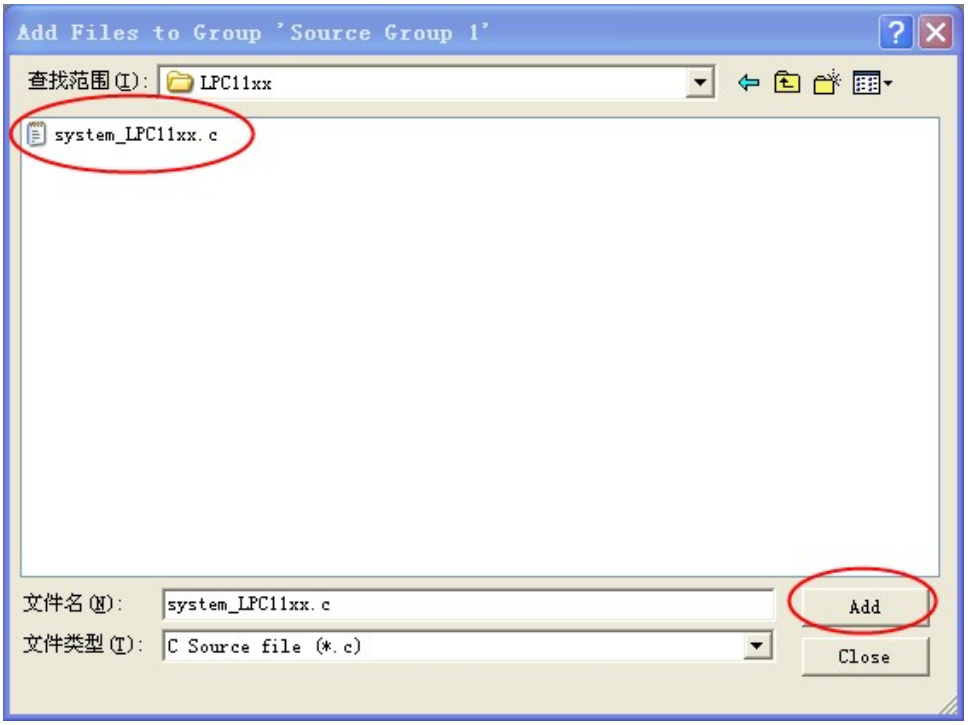
单击它前面的加号，出现 Source Group 1，然后再单击 Source Group 1 前面的加号，我们就可以看到现在整个工程里面，只有一个文件，就是我们刚才加载的 startup_LPC11xx.s 文件。如下图所示：



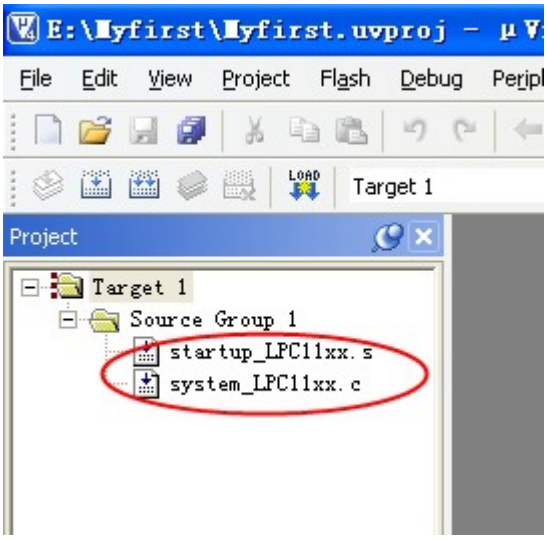
把鼠标放到 Source Group 1 上面，单击右键，在弹出的菜单中选择“Add Files to Group‘ Source Group 1’ ”，如下图所示：



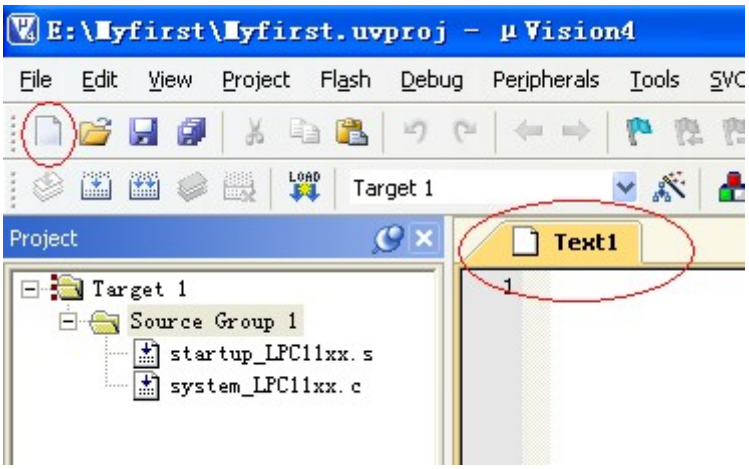
在弹出的窗口中，根据 KEIL 的安装路径，找到 system_lpc11xx.c 文件。例如在 RATION 电脑上的 KEIL 安装到了 E 盘，system_lpc11xx.c 的查找路径即为 E:/KEIL/ARM/Startup/NXP/LPC11xx，如下图所示：



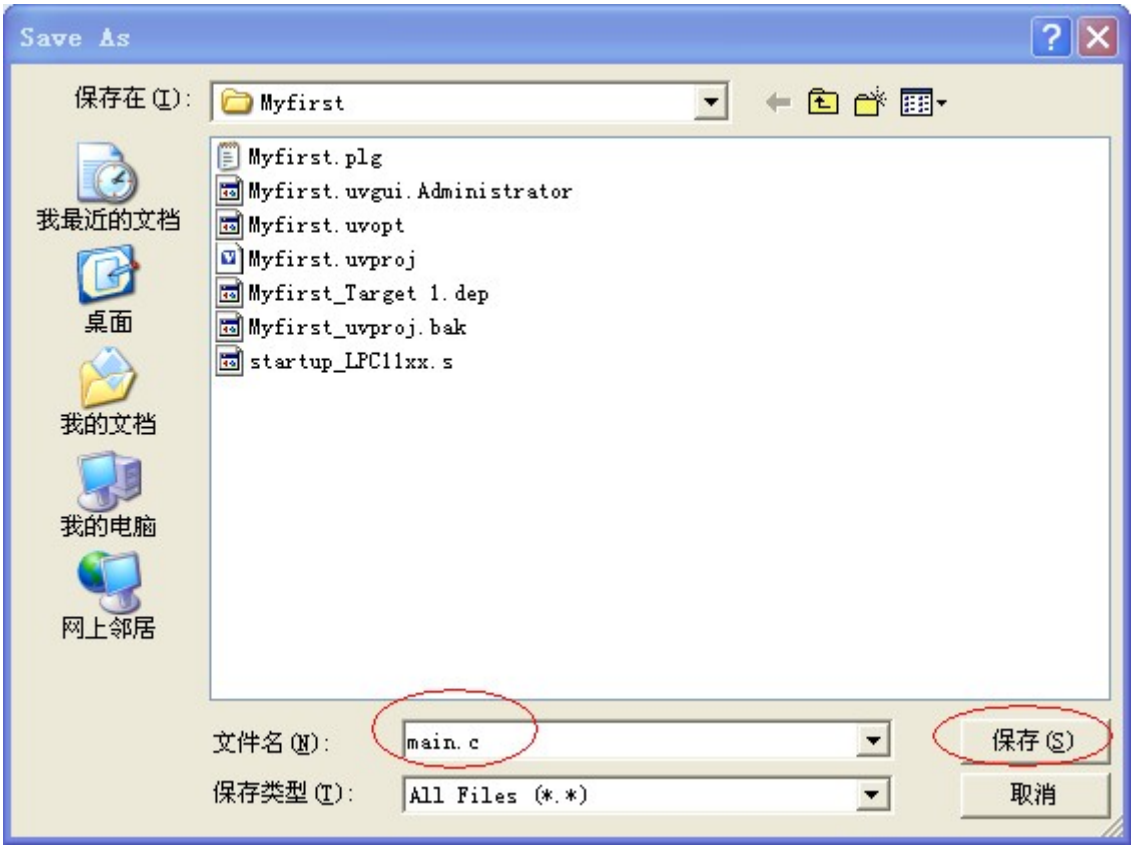
单击文件 system_LPC11xx.c，然后单击“Add”按钮。System_LPC11xx.c 就会添加到工程中去。也可以直接双击 system_LPC11xx.c 文件添加到工程中去。点“Close”退出窗口，回到工程。添加进去后，工程里面的文件就有两个了，如下图所示：



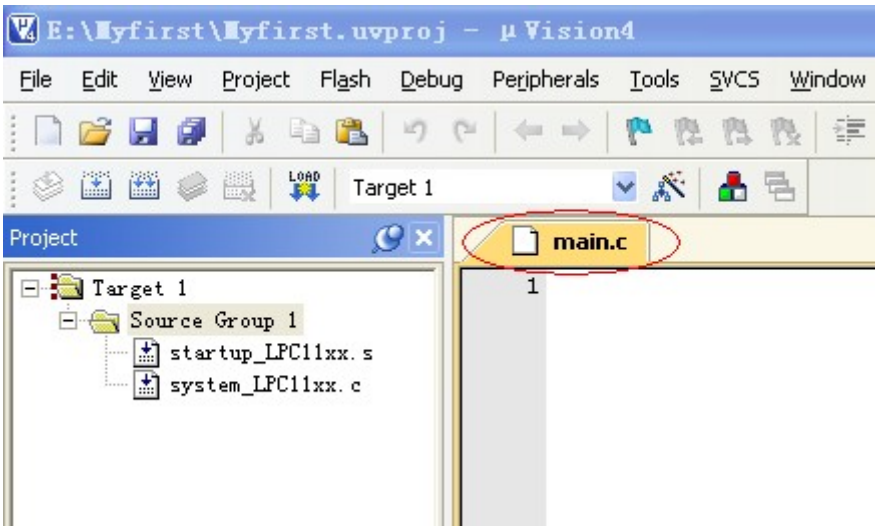
接下来需要新建一个.c 文件存放主函数，单击新建文件图标，或者通过菜单 File->New 新建一个文件。如下图所示：



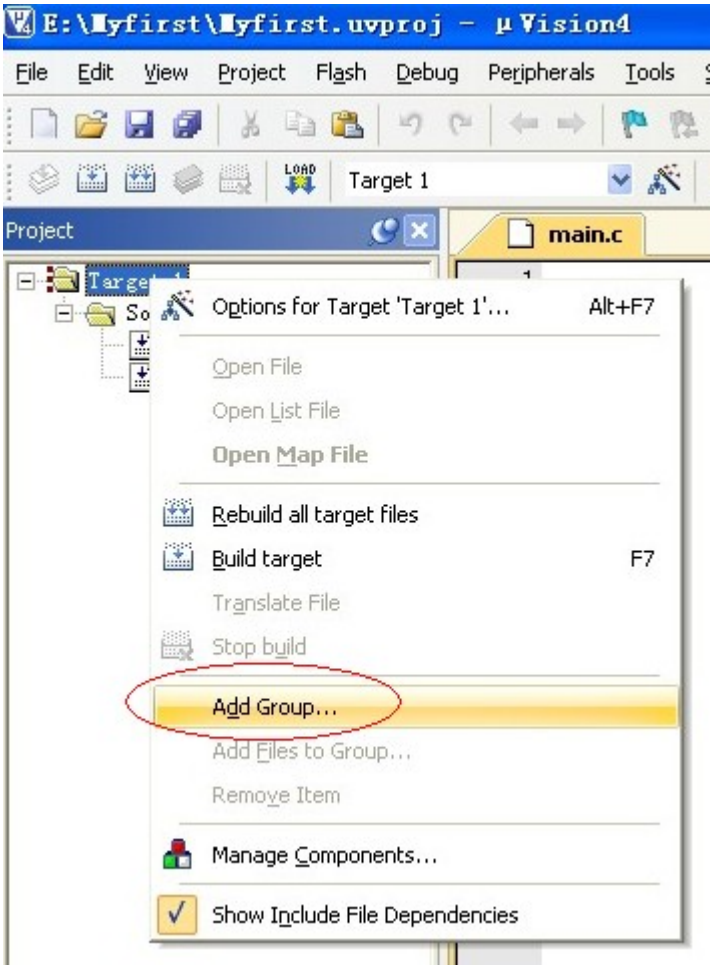
单击保存按钮图标，或者 File->Save,把新建的文件保存到工程文件夹目录下，给此文件起名为 main.c，可以起自己喜欢的名字，但是文件后缀一定要是.c。如下图所示：



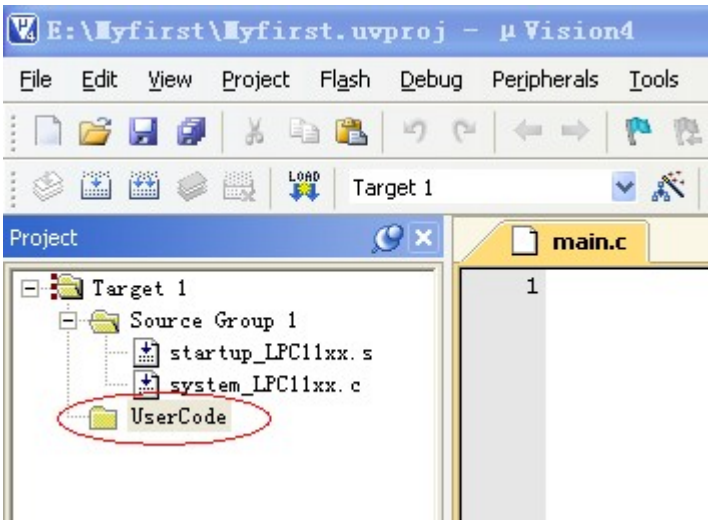
单击“保存”按钮，刚才的文件就保存到了工程目录下。文件名也会显示在工程中。如下图所示：



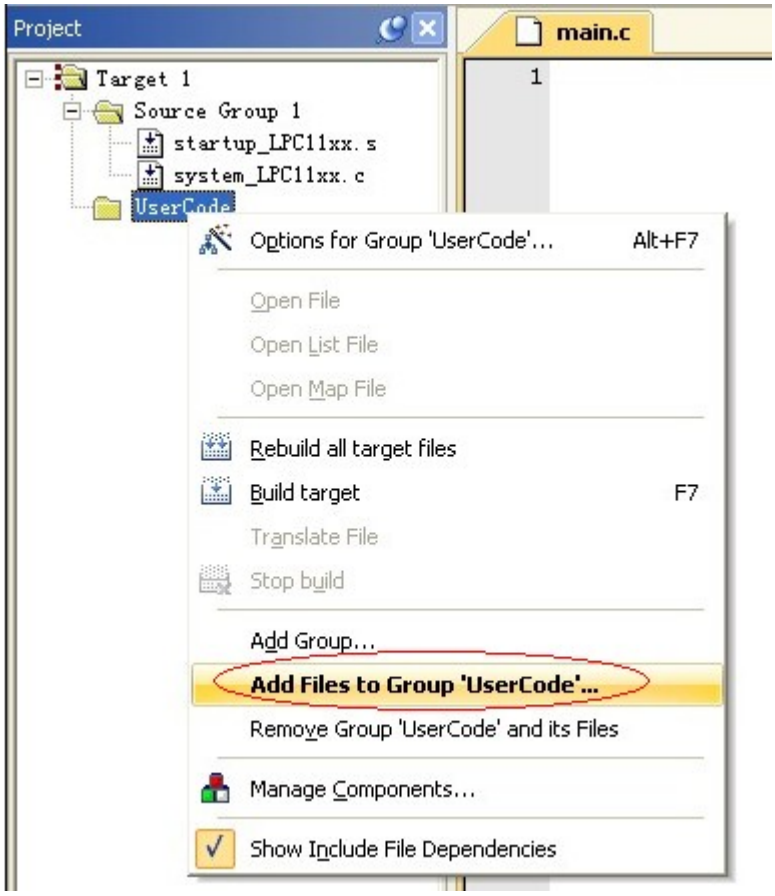
把鼠标放到 Target 1 上面，单击右键，在弹出的窗口中选择“Add Group”



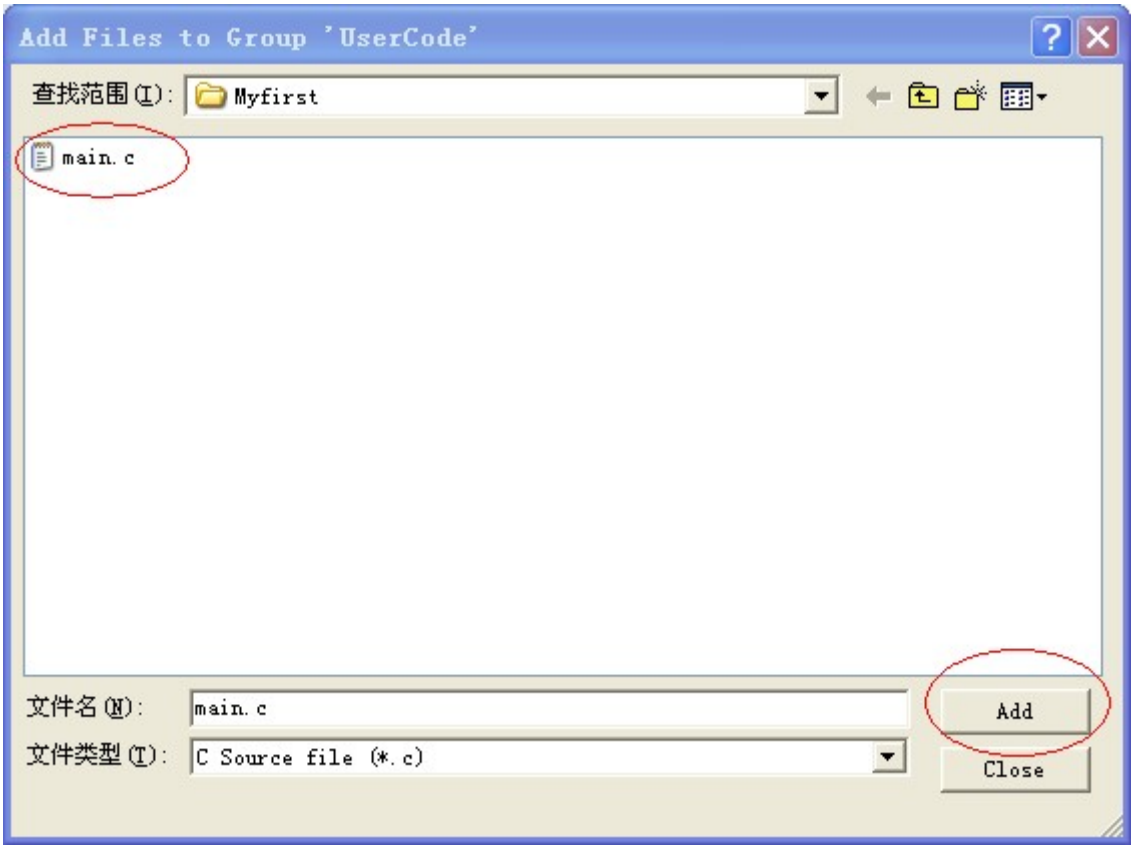
在 project 窗口中，会出现一个 New Group 的组，我们把它的名称修改为 UserCode，这里的名字也是一样，可以修改为你喜欢的名称。修改完以后，在 project 中就有两个组了，在 Source Group 1 组里面已经有了两个文件，我们新建了一个 UserCode 组，目的是为了把 main.c 加载到这个组里面。如下图所示：



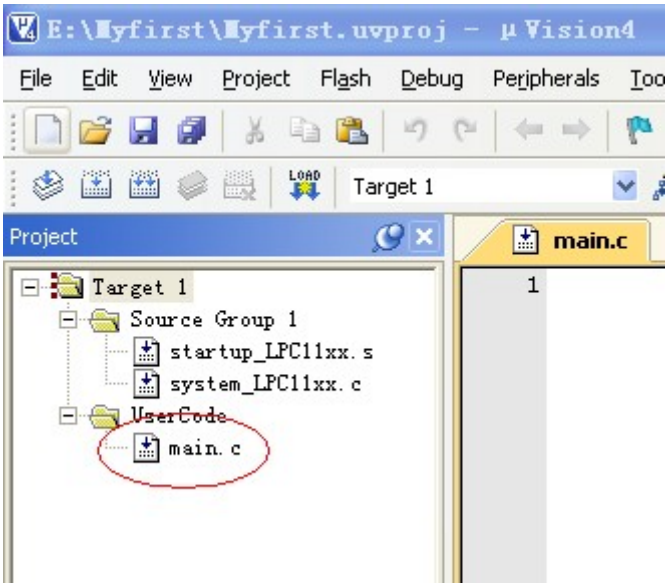
把鼠标放到 UserCode 上面，单击右键，在弹出的菜单中选择“Add Files to Group‘ UserCode’ ”。如下图所示：



弹出一个窗口，我们选择刚才新建的文件 main.c，然后单击“Add”按钮。如下图所示：



这时，main.c 文件就添加到工程的 UserCode 组了。单击 UserCode 前面的“+”号展开这个组的文件，如下图所示：



现在，我们就可以给 main.c 里面写代码了。

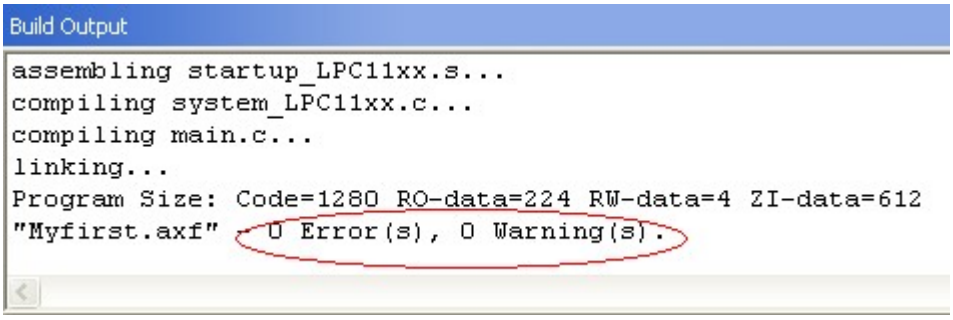
首先，我们要把 lpc11xx.h 头文件包含到文件当中，然后写一个主函数 main()。写完后，基本的格式如下图所示：

main.c

```
1 #include <lpc11xx.h>
2
3
4
5 int main()
6 {
7   while(1)
8   {
9     ;
10  }
11 }
12
13
```

单击工具栏中的“编译”按钮，编译按钮如下图所示：

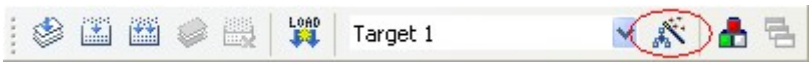
之后会在 Build Output 窗口中看到编译结果如下图所示：



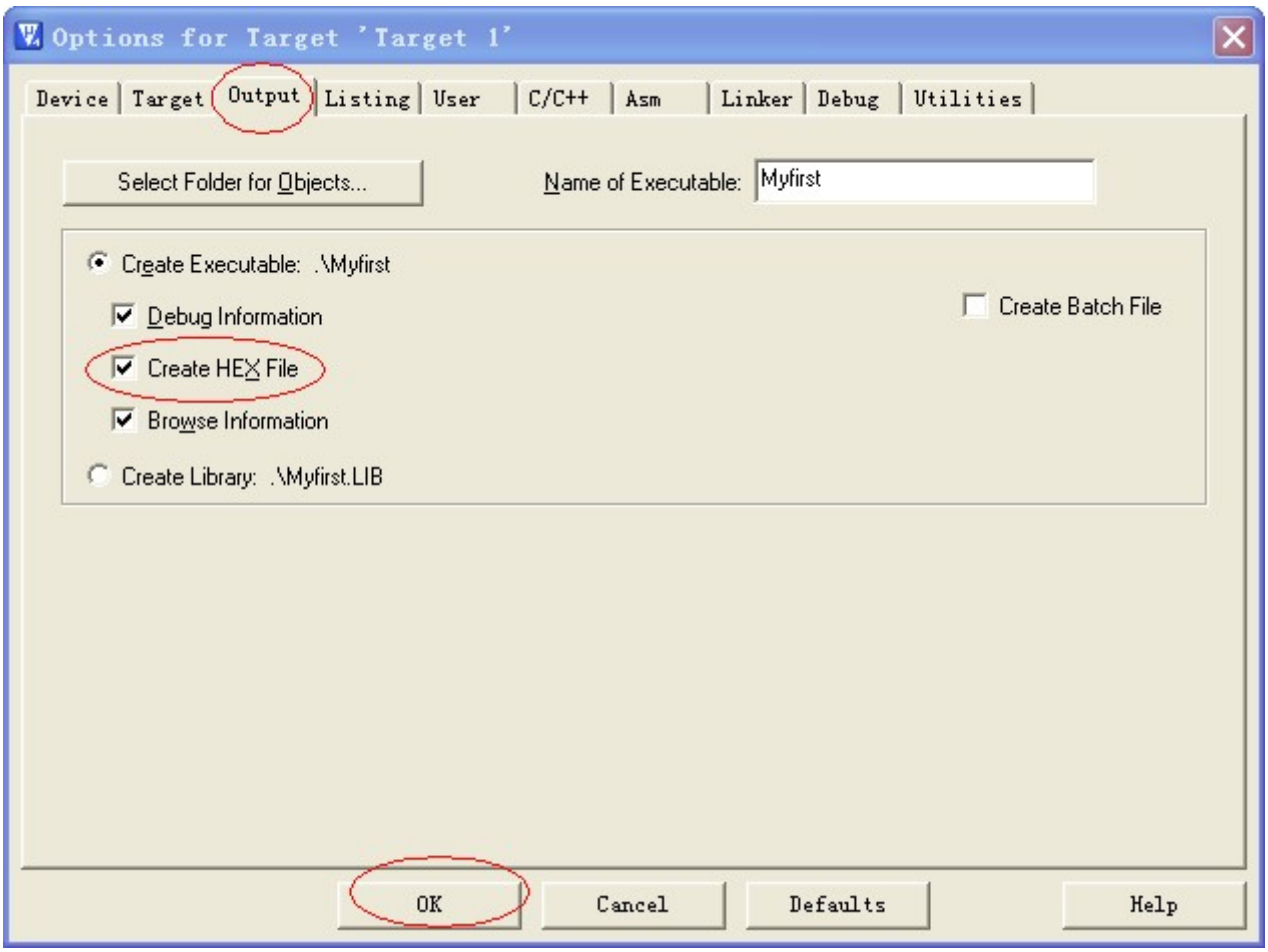
没有错误，没有警告！一个简单的工程就建成了。

3.3 生成 Hex 文件

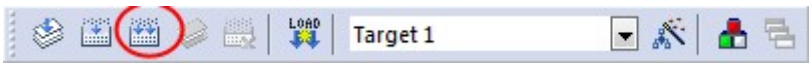
如果要生成可以下载到单片机的 HEX 文件，还需要进行如下设置。在工具栏中，单击 Target Options 图标，图标如下所示：



弹出一个窗口，单击 Output 选项卡，在 Create HEX File 前面打钩。如下图所示：



这时，在工具栏中单击编译按钮，将会生成 HEX 文件。



3.4 程序下载

LPC1114 程序的下载有两种方式，一种是通过串口，一种是通过 JTAG 接口。

串口下载程序，即通过 LPC1114 的 RXD 和 TXD 下载程序。需要借助 ISP 下载软件。这里我们用 FLASH MAGIC 软件下载。

手动下载：

LPC1114 的 P0.1 脚，即 BOOT 引脚，用来控制程序的下载，在 LPC1114 上电之前，把 BOOT 接地，上电后，单片机会等待程序的下载，下载好程序后，单片机断电，BOOT 与地断开，单片机重新上电后，会运行刚才下载进去的程序。也就是说，当单片机上电后，会首先检测 BOOT 引脚是否接地，如果接地，等待程序下载；如果 BOOT 没有接地，将运行用户程序。

自动下载：

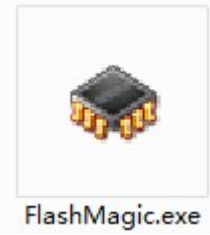
利用串口的 DTR 和 RTS 分别连接 LPC1114 的 RESET 和 BOOT 引脚，可以免去手动下载的麻烦，直接点击 FLASH MAGIC 的下载按钮，即可下载程序，下载完程序后自动从新复位运行刚刚下载的程序，使得开发更得心应手。

JTAG 接口下载程序，实质上是用 SWD 串行口下载，SWD 是 JTAG 的精简版本，专为 Cortex 系列处理器而生，只需要两条线，一条时钟线，一条数据线。需要借助 JLINK V8 或 ULINK2 仿真器。仿真器下载程序，直接在 KEIL 里面点击下载按钮即可下载。

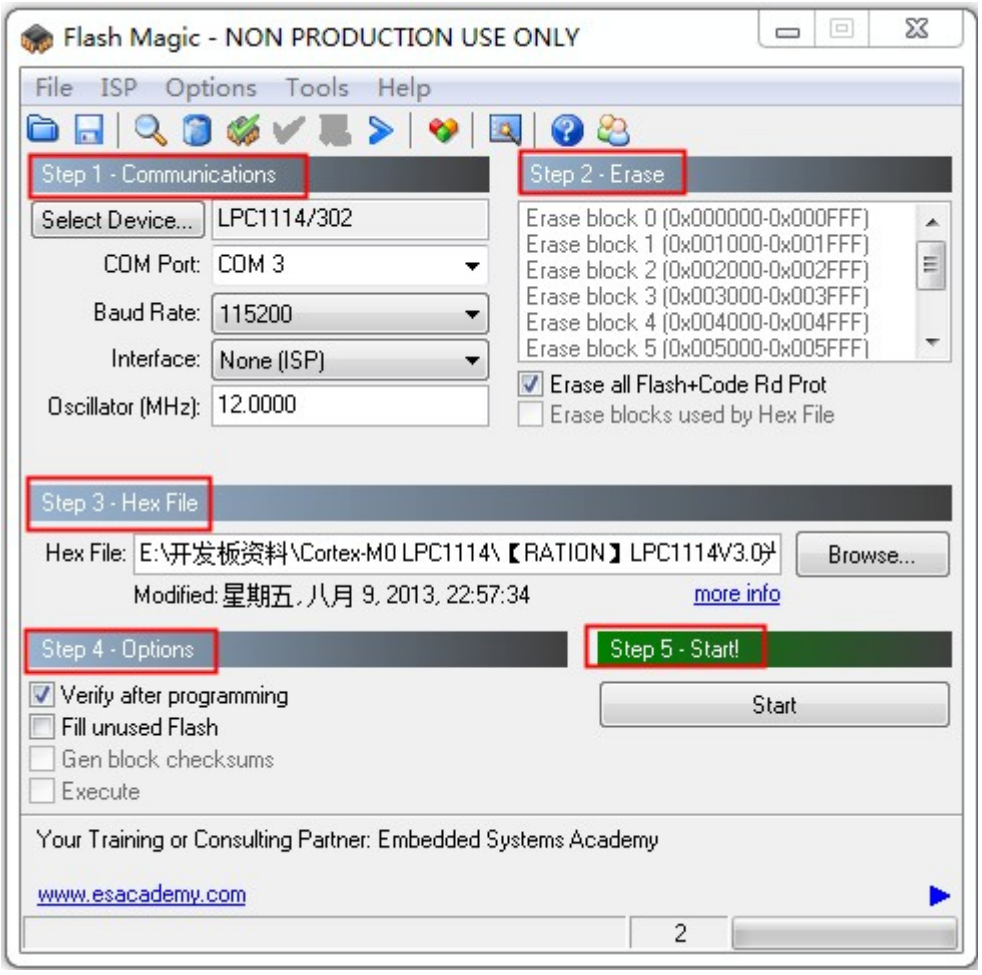
3.41 串口下载

FLASH MAGIC 软件最新版本下载地址：http://www.flashmagictool.com
安装软件如下所示：

双击文件，一路 NEXT，直到完成。完成后在桌面上产生一个图标：



双击图标打开，按照如下步骤配置：



一共有 5 个步骤：

第一步：选择正确的器件和串口号

第二步：勾选 Erase all Flash+Code Rd Prot

第三步：选择要下载的 Hex 文件

第四步：勾选 Verify after programming

第五步：点击“Start”按钮开始下载

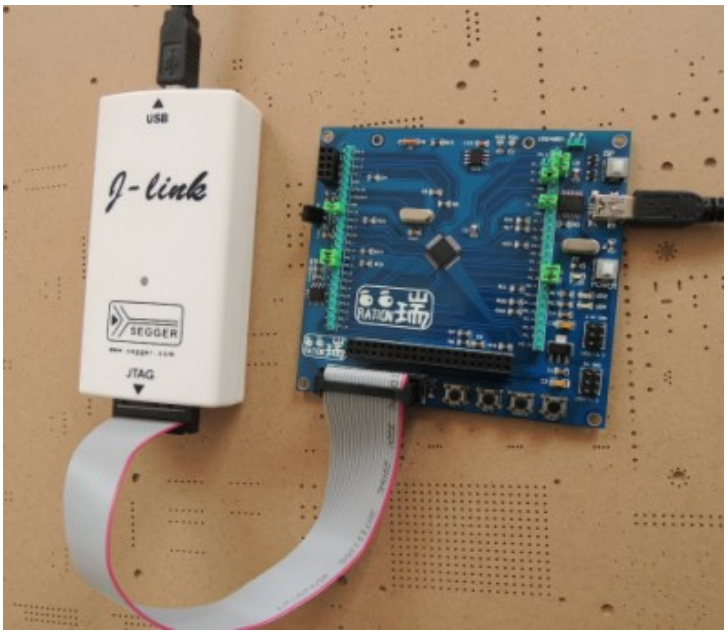
3.42 JTAG 接口下载（以 JLINK V8 为例进行介绍）

1. 安装 JLINK V8 驱动

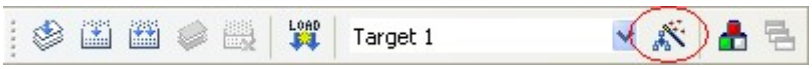
这里我们选用稳定版的驱动程序：408。安装好驱动以后，再把 JLINK V8 插到电脑 USB 口。安装成功的话，会在电脑“设备管理器”看到 JLINK V8，如下图所示：

2. 配置 KEIL

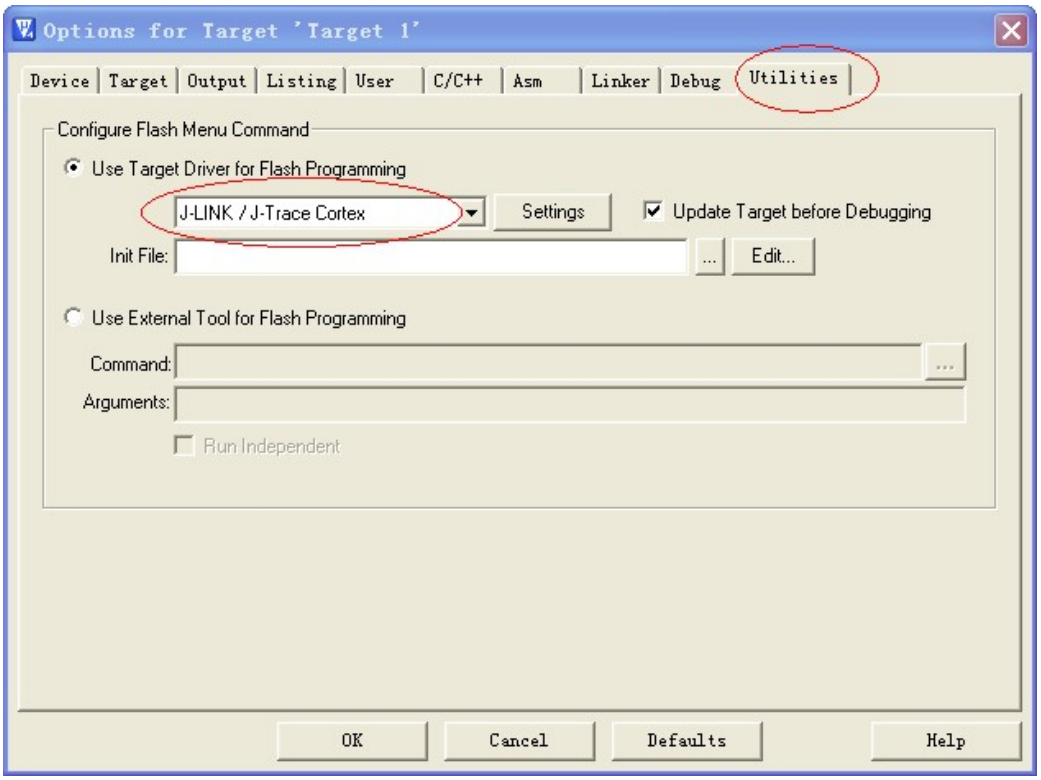
将 JLINK V8 一端与单片机目标板连接，一端与电脑连接。如下图所示：



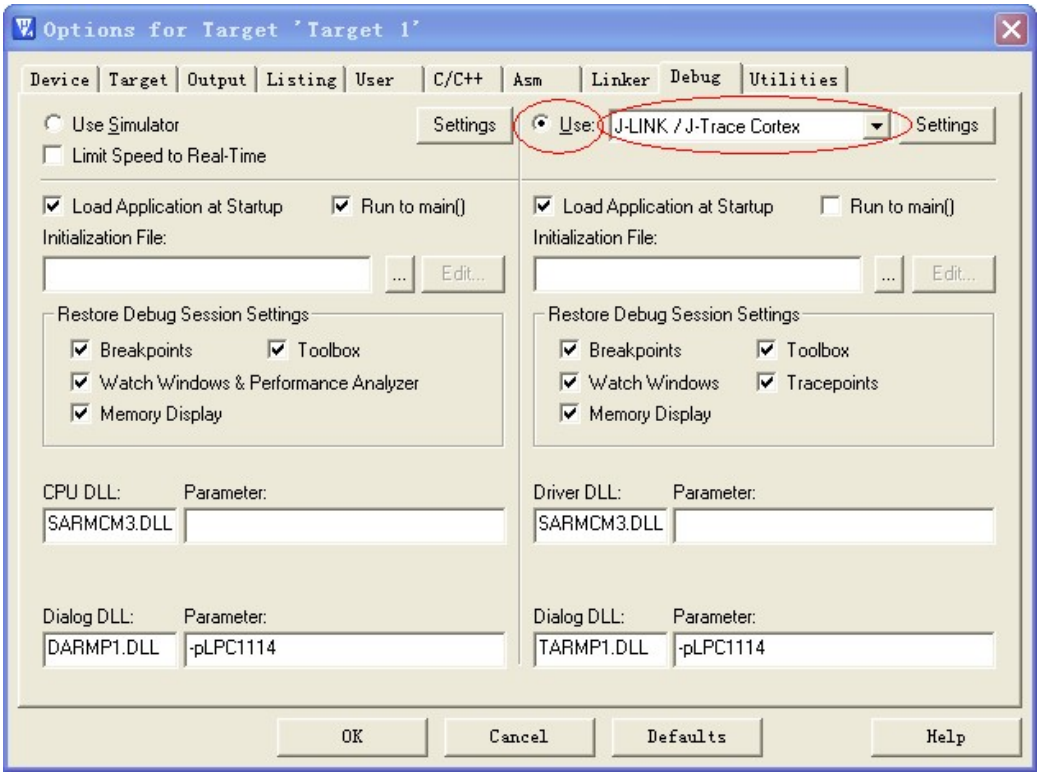
打开 LPC1114 工程文件，在工具栏中，点击 Target Options 按钮。



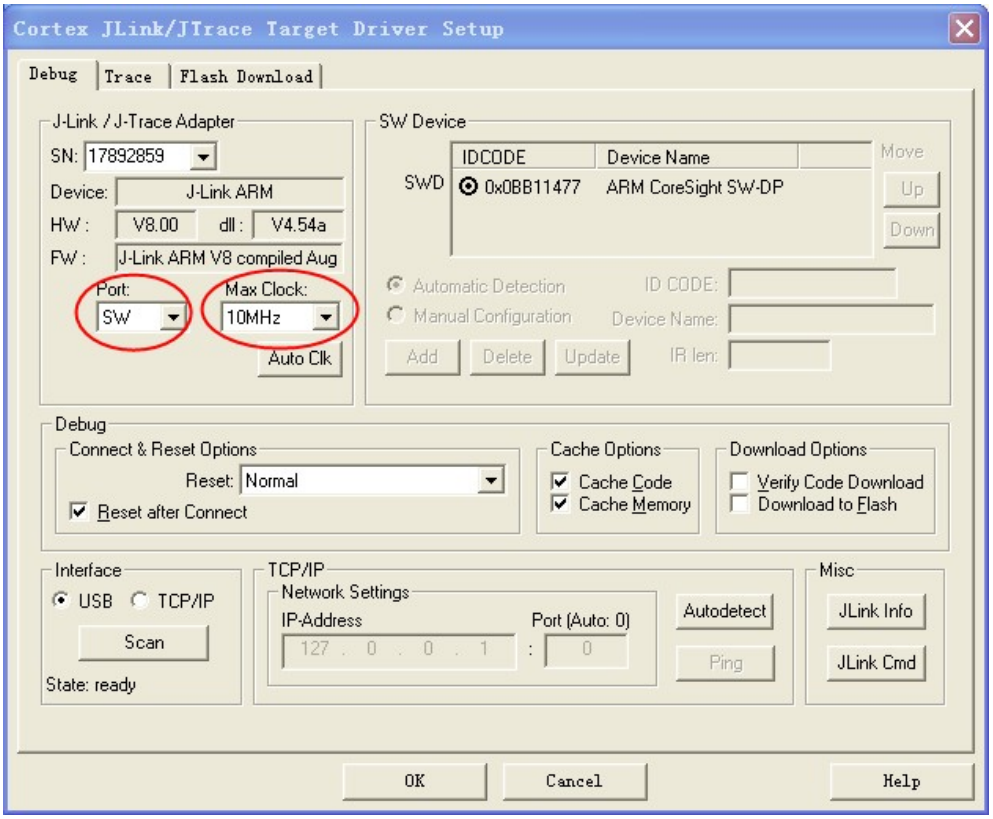
在弹出的窗口中，选择 Utilities 选项卡，按照下图配置好。



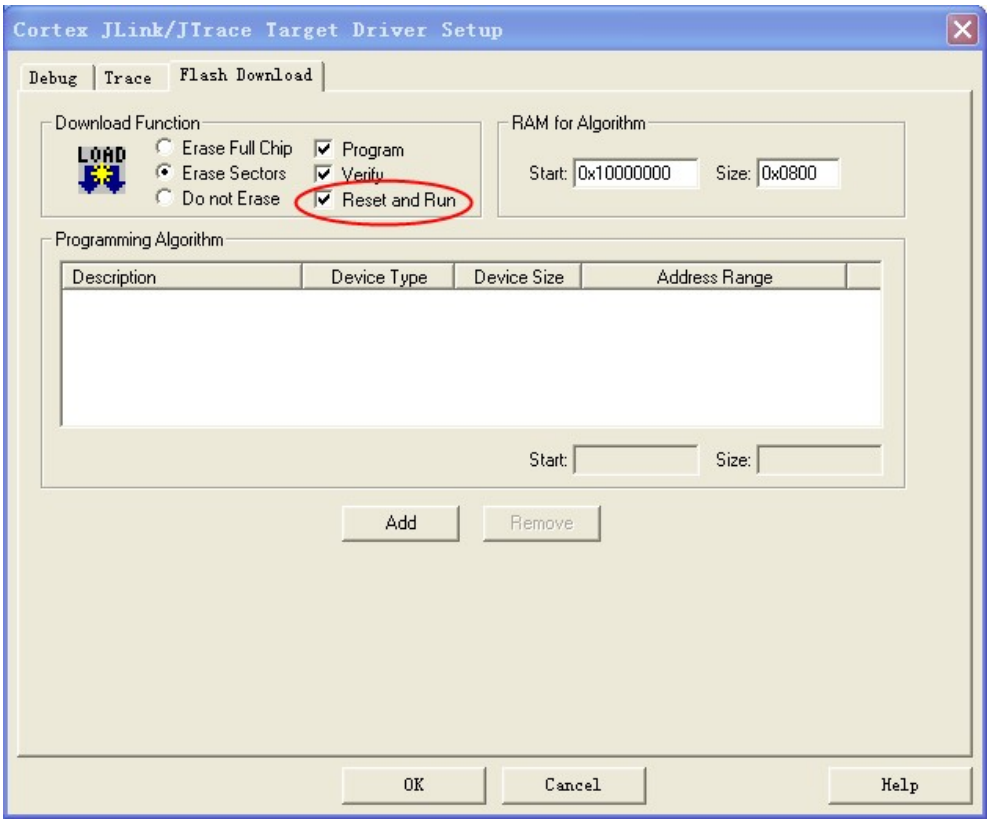
然后选择 Debug 选项卡，按照下图配置：



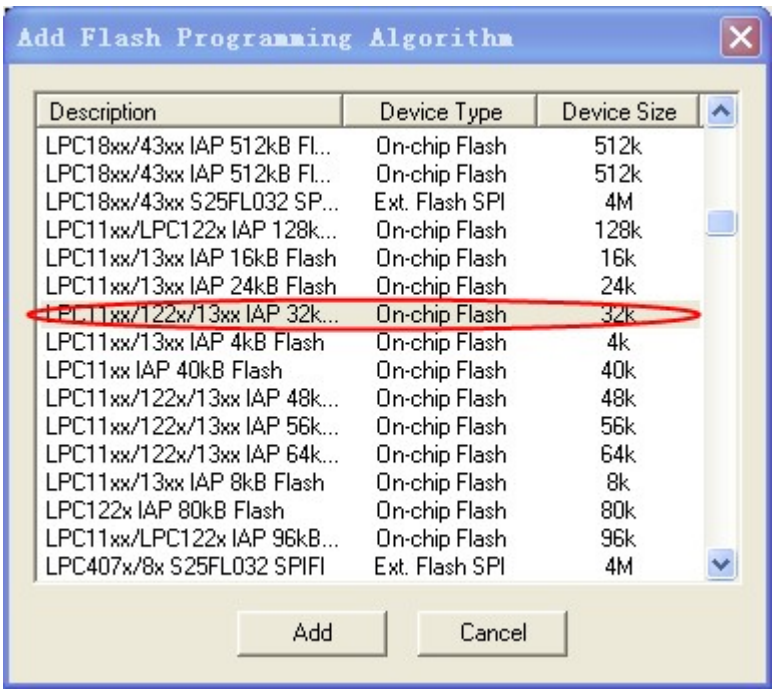
按照上图中画圆圈的部分设置好以后，点击“Setting”按钮。在弹出的窗口中，按照下面的图片配置：



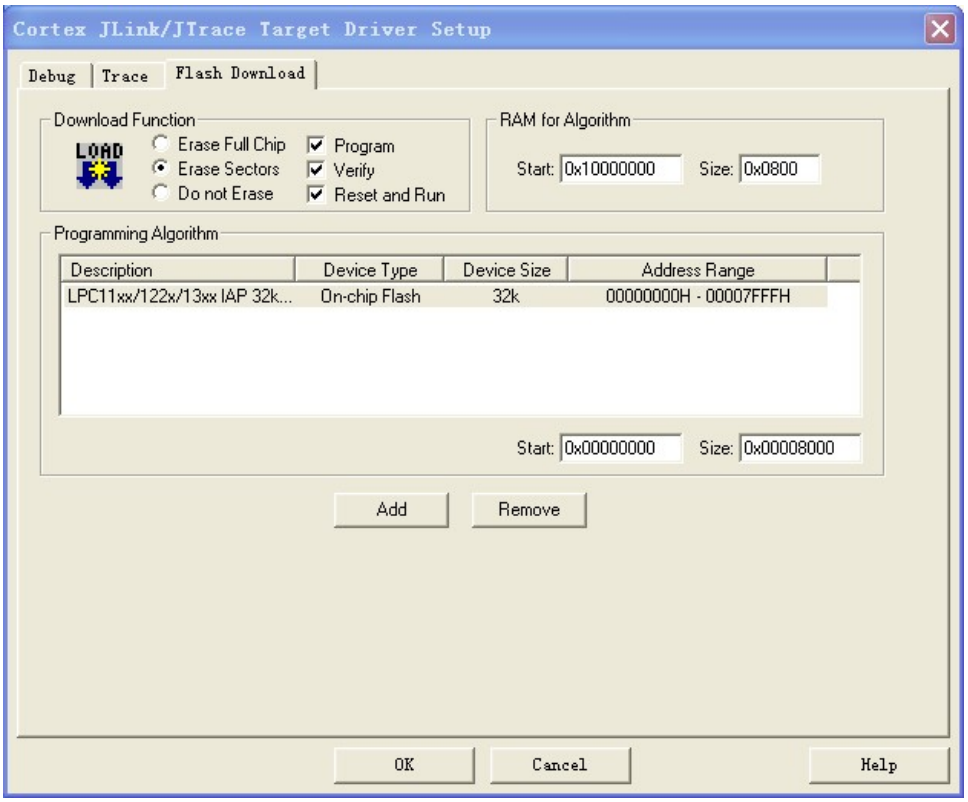
在上面的窗口中，单击 Flash Dowload 选项卡，如下图所示配置：



点击“Add”按钮，弹出一个窗口，如下图所示：



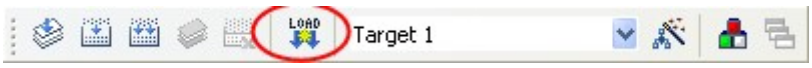
选择 LPC11xx/122x/13xx IAP 32k，单击 Add 按钮。回到原来的窗口，如下图所示：



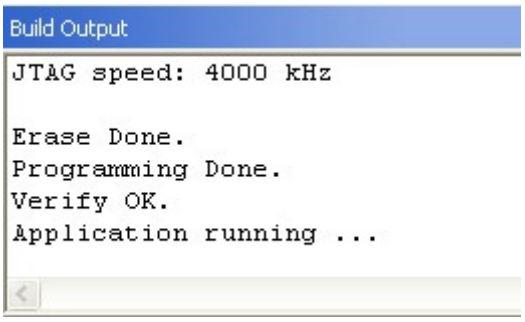
点击“OK”按钮，完成设置。回到刚才的窗口中，再次点击“OK”按钮回到编译环境中。

这时候，JLINK V8 在 KEIL 中就可以仿真和下载程序了。

点击工具栏中的 Dowload 按钮，如下图所示：



在 Build Output 窗口中，会看到 JLINK 正在下载程序，下载完以后，显示如下图所示：



提示程序下载成功，并且开始运行。