

LPCXpresso 用户手册 V1.0

1、绪论

LPCXpresso 是来自 NXP 的一款新的、低成本开发平台。其软件部分包括增强型 IDE 开发环境、GNU C 编译器、连接器、库函数、增强型 GDB 调试器。硬件部分包括 LPCXpresso 开发板，该开发板包含两部分：LPC-Link 调试接口板、LPC ARM 微控制器目标板。LPCXpresso 是一个中断对终端解决方案，它可以帮助嵌入式工程师完成从产品的初始评估到最终产品的所有工作。

LPCXpresso IDE 是由 Code Red Technologies 公司开发的基于流行的 Eclipse 开发平台并且支持 LPC 系列器件。它是一个符合行业标准的 GNU 工具链，它的优化 C 库函数提供给工程师各种所需的开发工具，使得工程师能够获得快速、廉价的高质量软件解决方案。C 编程环境具有专业特色：语句/关键字颜色设置、源程序格式设置、展开/收缩功能、离线/在线帮助、自动项目管理。

LPCXpresso 目标板由 NXP、Code Red Technologies、Embedded Artists 共同合作开发。板载集成的 JTAG 调试器（LPC-Link），不用再另外配置单独的 JTAG 调试器。核心半部分提供了多种接口和 I/O 驱动方式，可以方便地进行功能扩展。板载 LPC-Link 调试器提供高速 USB 转 JTAG/SWD 接口连接到 IDE 开发软件，并且还可以作为调试器连接到其他的目标板进行调试。用户还可以从 Code Red Technologies 购买 Red ProbeJTAG 适配器在 LPCXpresso IDE 上进行开发。

LPCXpresso 支持下列 LPC 器件：

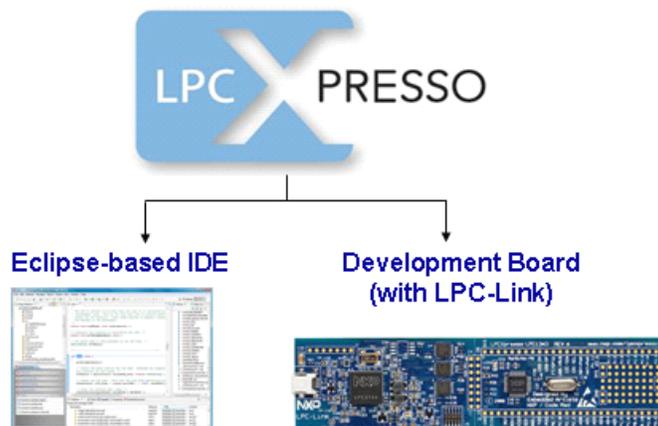
LPC11XX: 全系列

LPC13XX: 全系列

LPC17XX: LPC1751, LPC1752, LPC1754, LPC1756, LPC1758, LPC1764, LPC1765, LPC1766, LPC1767, LPC1768

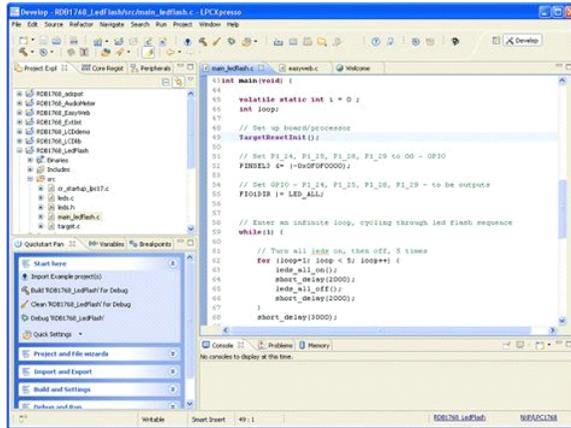
LPC2XXX: LPC2109, LPC2134, LPC2142, LPC2362

LPC3XXX: LPC3130



1.1 LPCXpresso IDE

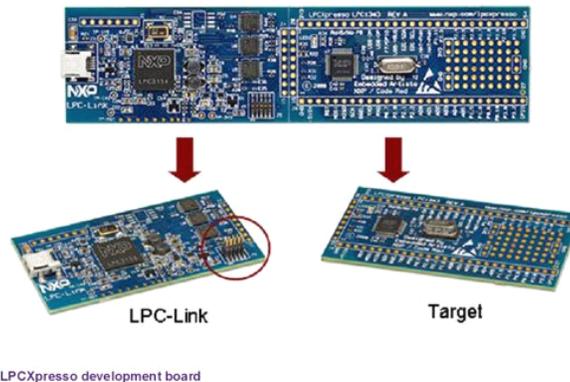
LPCXpresso IDE 是一个针对 LPC 微控制器的高度集成的软件开发环境，它包含要求快速、廉价方式软件解决方案所需要的所有工具。LPCXpresso 基于许多 LPC 器件增强的 Eclipse 技术。它的特征：低成本、符合最新版本的行业标准 GNU 工具、专业的 C 优化库函数。LPCXpresso IDE 可以编写任意长度的可执行代码，并且支持代码优化。注册后支持最大128K 的代码下载限制。LPCXpresso 可以在仿真板上进行开发，也可以在扩展目标板上进行开发。下图是 LPCXpresso IDE 的界面。



ig 1. LPCXpresso IDE

1.2 LPCXpresso 开发板

如下图：



LPCXpresso development board

1.3 LPC-Link JTAG/SWD 调试器

LPCXpresso 开发板包含一个被称为 LPC-Link 的 JTAG/SWD 调试器和一个目标 MCU。LPC-Link 包含一个10芯 JTAG 接头（上图中用红色圈起来的部分），经过 USB 接口与目标实现了无缝连接(由一片 NXP 公司的 ARM 芯片 LPC3154实现 USB 转换和调试信息的处理)。沿着 LPC-Link 和目标 MCU 中间的切割线把板子一分为二，就可以得到一个独立的 JTAG 调试器了。这样就可以使用这个调试器开发更多类型的 LPC 芯片了。

1.4 集成的目标板评估系统

目标板包括一个小的原型区域和便于扩展的接口。LPCXpresso 开发板的目标板可用于：自身的软件开发利用板载 MCU，连接到扩展板利用 JTAG 接口，连接到别的用户目标板。

第一个 LPCXpresso 目标板将是基于 LPC1000 系列的。

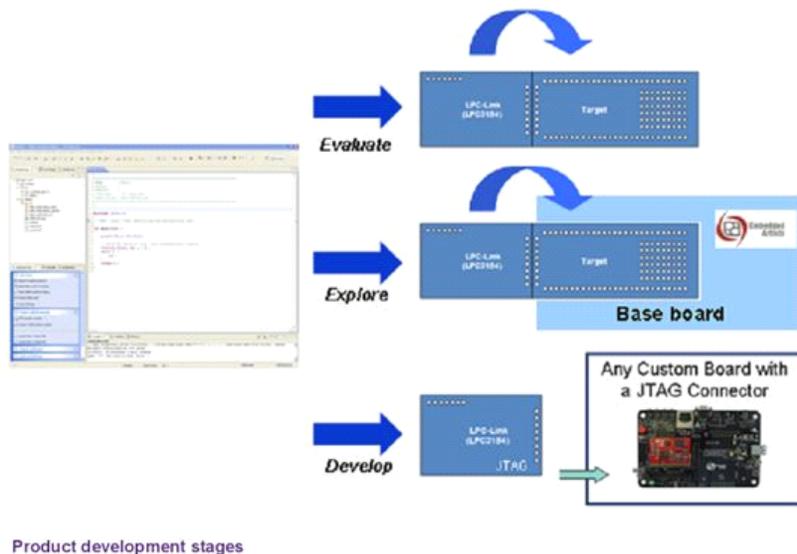
1.5 LPCXpresso 合作伙伴

LPCXpresso 目标板由 NXP、Code Red Technologies、Embedded Artists 共同合作开发。为增加适应性和更高的存储器配置，LPCXpresso 平台可以很容易地从 Code Red Technologies 获得升级，还可以从 Embedded Artists 获得更多的硬件增强开发包。请访问 LPCXpresso 网站获得更多信息。



2 评估、调试、开发

开发步骤如下图所示。



上图显示了使用 LPCXpresso 开发板的三种方式。用户可以使用板载的目标 MCU 和接口进行性能测试，也可以连接到扩展板上进行试验。还可以利用 JTAG 接口连接到其他目标板上进行开发。

3 安装

3.1 计算机配置要求

操作系统: windows xp

windows vista

windows7

内存: 最低512MB

硬盘: 最少225MB 可用空间

显示器: 最低分辨率1024*768

网络连接: 需要高速网络已完成软件下载和注册

Operating System	Microsoft® Windows - XP (SP2 or greater) Microsoft® Windows - Vista 32 bit Microsoft® Windows - Windows 7 32 bit
System RAM	512 MB minimum (1 GB recommended)
Hard Disk	225 MB of available space.
Screen/Display Adaptor	1024x768 minimum recommended
Internet Connection	High-speed internet is recommended to download and register the software

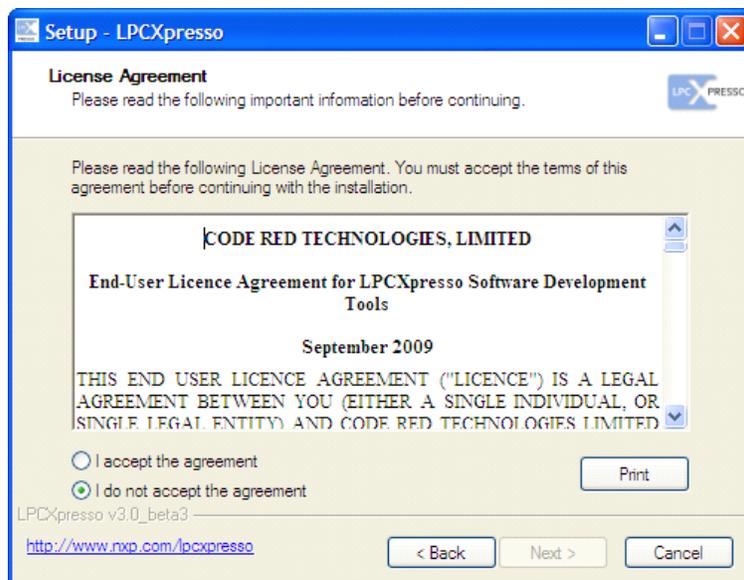
Note: Desktop virtualization tools with USB support can be used to run LPCXpresso on other computing platforms.

3.2 安装步骤

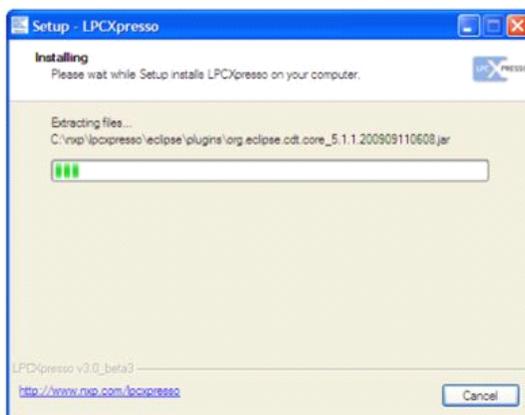
LPCXpresso 可以按照默认路径安装，也可以自定义安装路径。多种版本可以同时安装，不会互相冲突。软件下载完成后，双击安装文件开始安装，单击“下一步”，如下图。



选择“接受许可协议”，单击“下一步”继续安装。如下图。



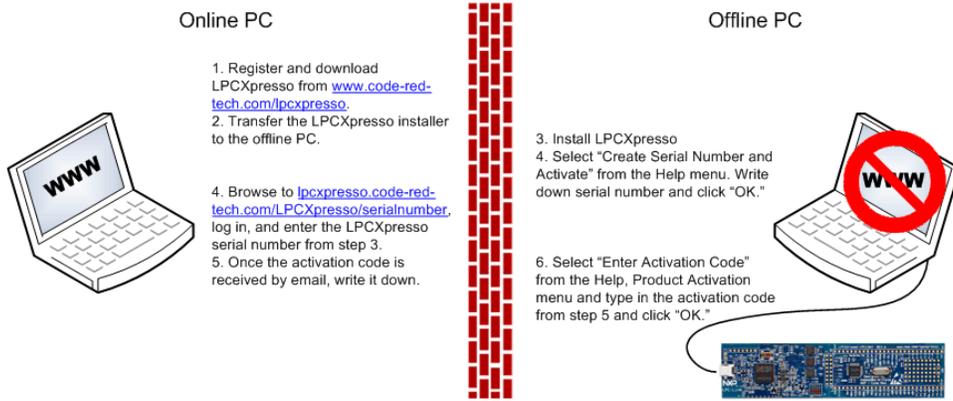
安装过程中会弹出一些选择页面，选择接受全部默认设置即可，如下图。



安装完成后，会弹出并显示一些相关自述文件、信息文件，不想阅读的话，关闭即可。单击“下一步”。恭喜！你已经成功安装 LPCXpresso 了！

3.3 激活

想要激活你的 LPCXpresso，首先运行该软件，按照如下步骤操作：Help->Product activation->Create Serial number and Activate 当页面打开后，单击“copy to clipboard”，复制 LPCXpresso 的序列号到 clipboard 中，这个序列号是基于你的计算机硬件和操作系统配置生成的，不过不包含个人隐私信息。然后单击按钮进入注册激活页面，激活页面以网页形式显示。完成这些内容后，稍等几分钟，你会在你的注册邮箱中收到包含激活码的邮件。从邮箱中复制激活码并放入 clipboard 窗口内。然后选择 Help->Product activation->Enter Activation code。将产品激活码填入激活码对话框中。最后单击“OK”按钮。将会弹出一个激活码确认对话框。对于不能上网的情况来说，也可以完成激活功能。参考下图。



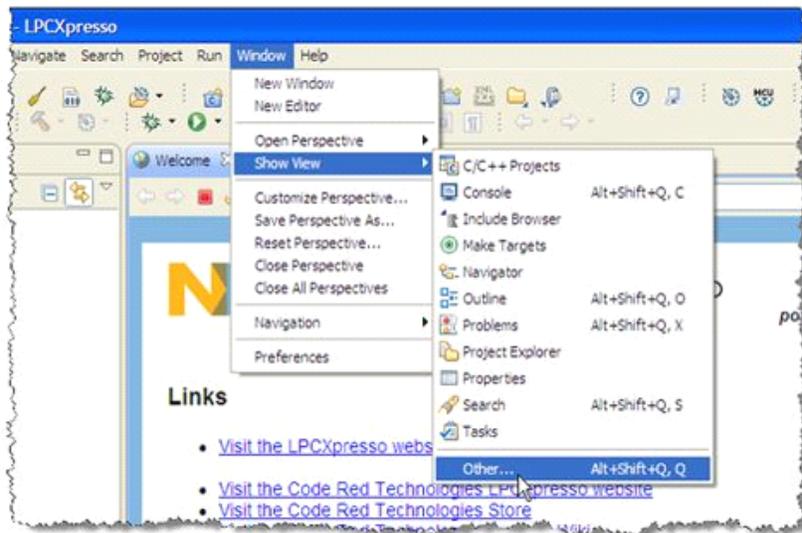
4 开始熟悉 LPCXpresso IDE

LPCXpresso IDE 基于 Eclipse IDE 框架，在 Eclipse 文档中有很多内核特征的详细描述，在帮助菜单中可以发现这些帮助文件。更多的用户文档可以到 Code Red 的网站上获得。

4.1 LPCXpresso 的桌面布局

LPCXpresso 桌面包含很多窗口，每一个窗口分别显示 LPCXpresso 环境一个特定的详细数据，因而被称为观察窗口，这些数据可以是源代码、hex 数据、反汇编、存储器内容等等。观察窗口可以打开、移动、固定、关闭，并且当前的窗口布局可以保存和恢复。一个特殊的窗口配置称之为“透视图”。在 LPCXpresso 中，无论代码编写还是调试都在一个界面下进行。这种方式使得开发工作变得简单和高效。

所有的观察窗口都可以随意拖动。如果一个观察窗口被无意关闭了，可以从“Show View”对话框中再次打开。“Show View”对话框可以从“Window” --- “Show View” --- “other”中打开，如下图所示。



4.1.1 代码编写

代码编辑界面如下图所示，由4个部分组成，功能介绍如下。

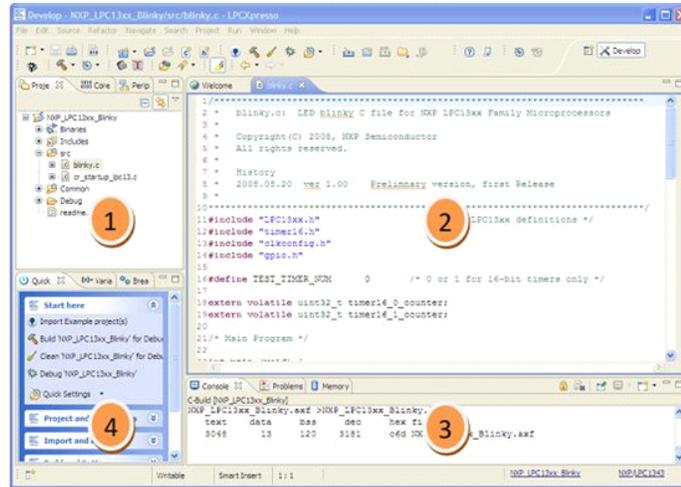


Fig 9. Single perspective (develop)

1、项目管理窗口：项目管理窗口显示当前“工作区域”下的所有项目文件。“工作区域”是指你在计算机中建立的项目文件所在的文件夹。

2、编辑窗口：代码编辑窗口用于输入、修改、保存代码文件。在调试的时候，还可以在该窗口中设置断点。

3、控制台问题显示窗口：控制台窗口显示程序输出时的编译和调试的状态信息。问题窗口（通过标签选项选择）显示编译的错误信息以及错误信息所在的位置。

4、快速启动窗口：快速启动窗口包含一些经常用到的使用选项，这是寻找编译、调试、输入等选项的最方便的位置。

4.1.2 调试

调试窗口由5部分组成，如下图所示。各部分功能分别是：

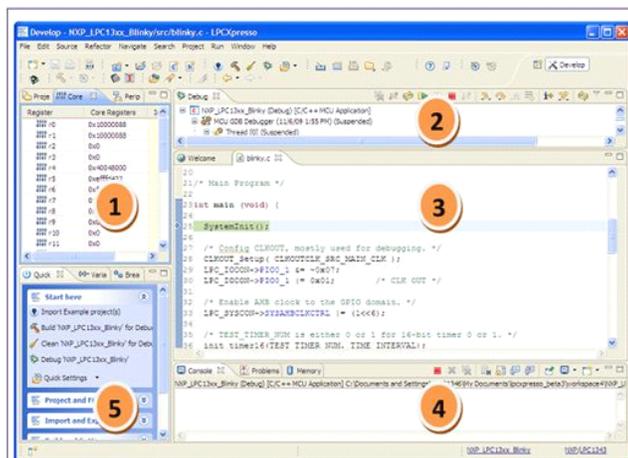


Fig 10. Single perspective (debug)

1、寄存器观察窗口：该窗口显示微处理器中的所有寄存器信息。调试过程中寄存器的值改变的时候，都会以高亮黄色字体显示出来。

2、调试窗口：该窗口显示堆栈和调试工具栏，可以使用快捷图标进行单步、全速等调试功能。还可以随时点击“停止”，以观察变量变化情况。

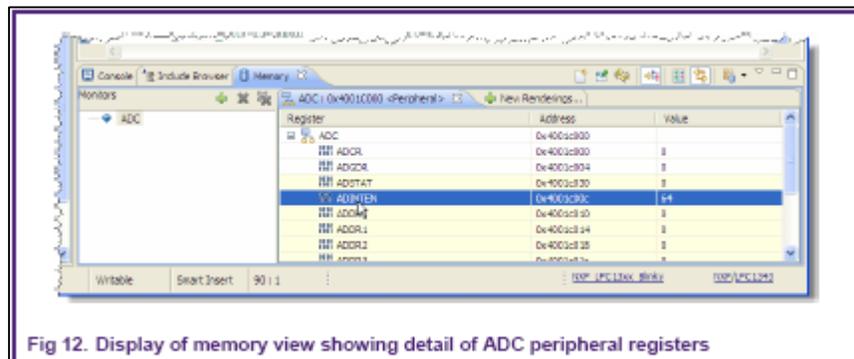
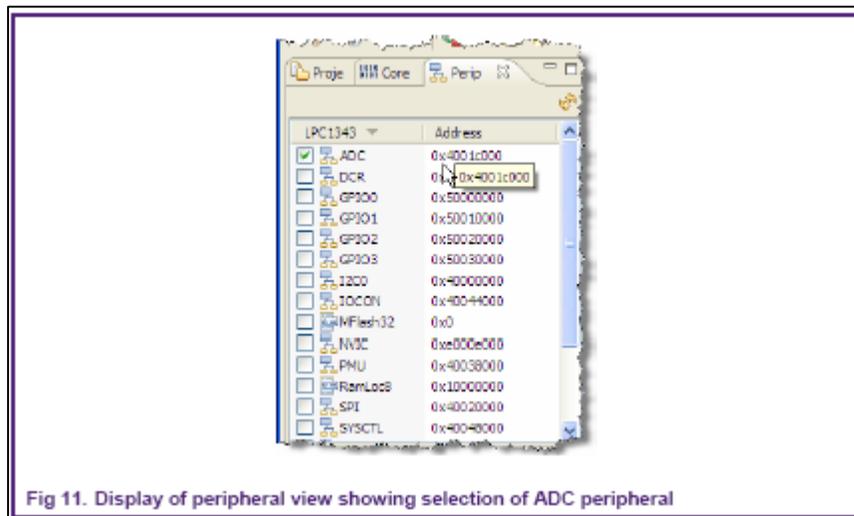
3、编辑窗口：可以在该窗口中观察所执行的代码。单击“i”图标，可以观察汇编指令的执行情况。还可以设置和删除断点。

4、控制台观察窗口：该窗口显示程序输出时的编译和调试的状态信息。

5、快速启动窗口：快速启动窗口可以快速的寻找到编译、调试、输入等选项。

4.1.2.1 外设观察窗口

LPCXpresso IDE 有一个包含所有外设的注释、交互式调试观察窗口。可以通过标签项选择寄存器观察窗口或者外设观察窗口。外设被选中后，会在控制台观察窗口显示“memory”观察窗口。分别如下图所示。



4.2 连接目标板

使用一根 USB2.0A/Mini-B 电缆将开发板和计算机连接起来，就可以进行开发了。电缆如下图所示：

**USB 2.0 A/Mini-B cable**

如果需要调试其他型号的 MCU，请参考附录中的相关文档。

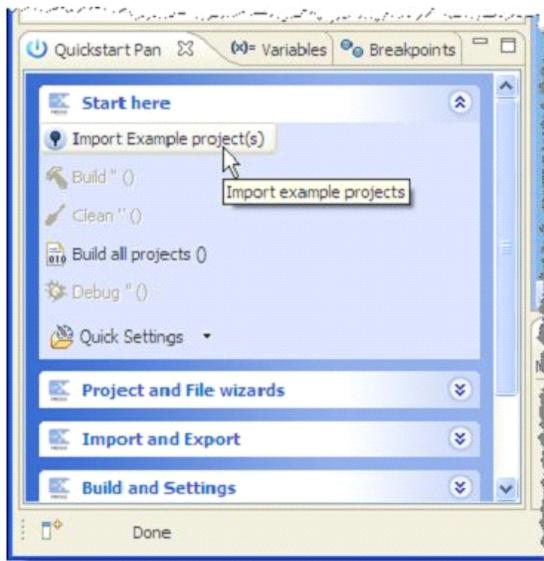
5 闪烁灯：编译、下载、调试

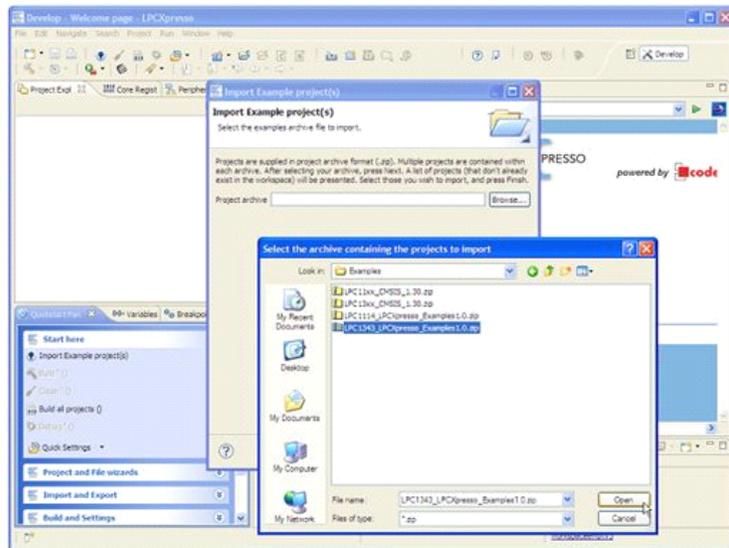
5.1 从快速启动面板导入闪烁灯项目

例子项目对于初始开发有很大的帮助作用。

在 LPCXpresso 开发界面里，快速启动面板里包含了使用最频繁的各种操作选项。它位于窗口的左下角。

- 在快速启动面板中，单击“Start Here”，然后选择“Import Example project(s)”。如下图所示。
- 在弹出的页面中，从“Browse”中选择例程所在的目录。根据提示打开例程即可。如下图。

**Quickstart panel**



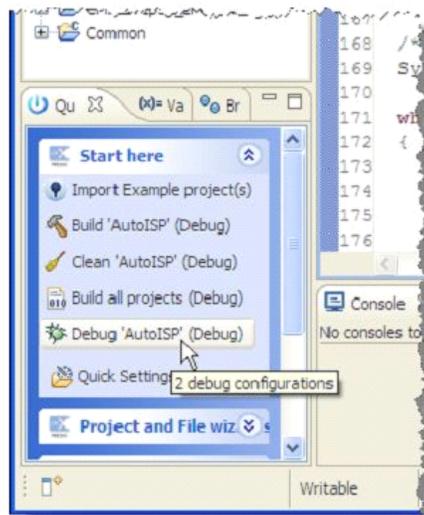
5. Examples directory

- 然后就可以从项目列表中选择例程项目。要确保“Blinky”和“CMSIS”这两个项目都存在。
- 单击“Finish”，这两个项目将会被导入到当前工作区。
- 在快速启动面板里单击“Build all projects(Debug)”开始编译 Blinky 例子和 CMSIS 库。

5.2 在开发板上调试/运行“blinky”程序

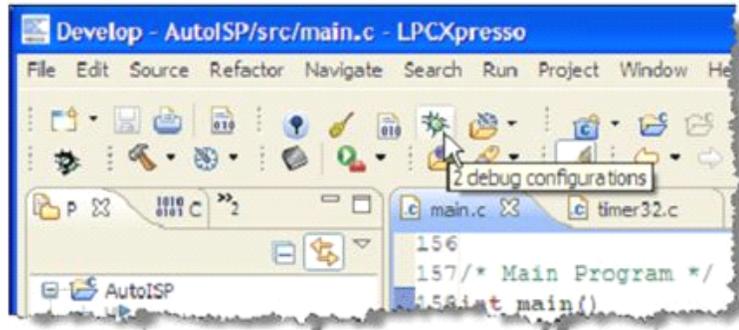
LPCXpresso 开发界面下，当开始调试的时候，程序会自动下载到目标 MCU 的 flash 中。

在项目管理区中选中将要调试的项目，然后点击快速启动面板中的“Debug project ‘Blinky’”。这时会弹出一个询问你使用哪种执行方式（发布或调试，Release or Debug）选项的对话框，选择“Debug（调试）”选项，然后单击“OK”完成程序下载和创建项目调试操作。如下图所示。



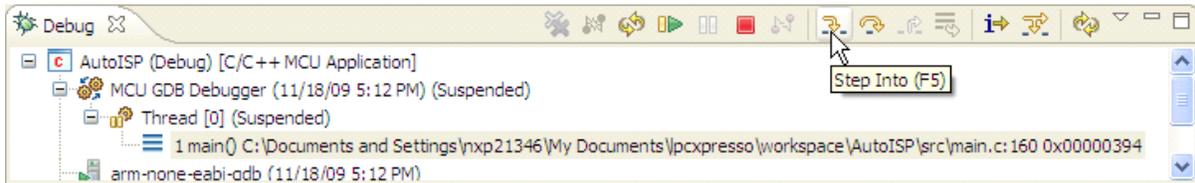
i. Debug

也可以通过点击快捷工具栏中的调试（BUG）图标来进入调试模式。如下图所示。

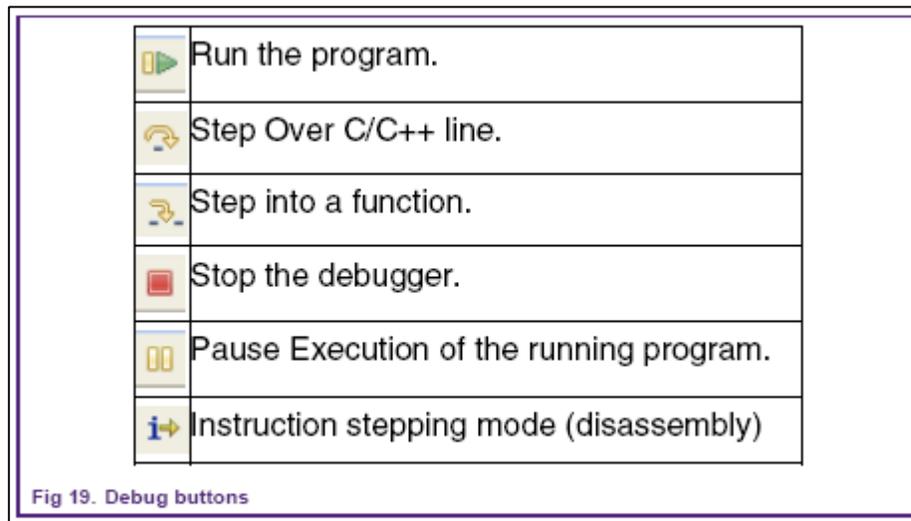


Bug icon

进入调试界面后，调试信息观察窗口以及调试工具栏都会显示出来。如下图所示。



现在，你可以使用下图所示的按钮进行调试操作了。



6 LPCXpresso IDE 调试系统和技巧

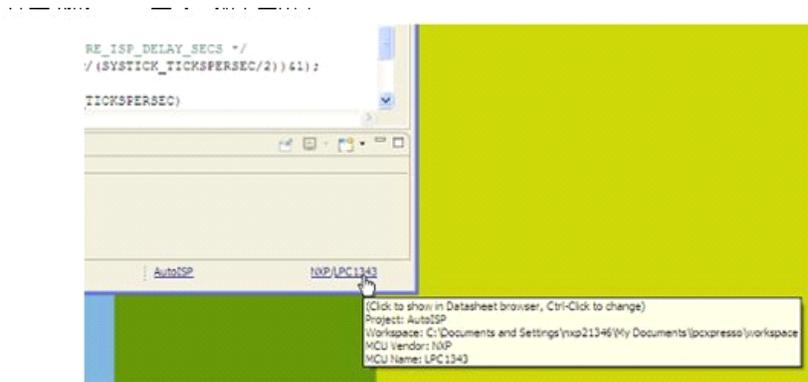
6.1 调试系统

6.1.1 调试属性未使能

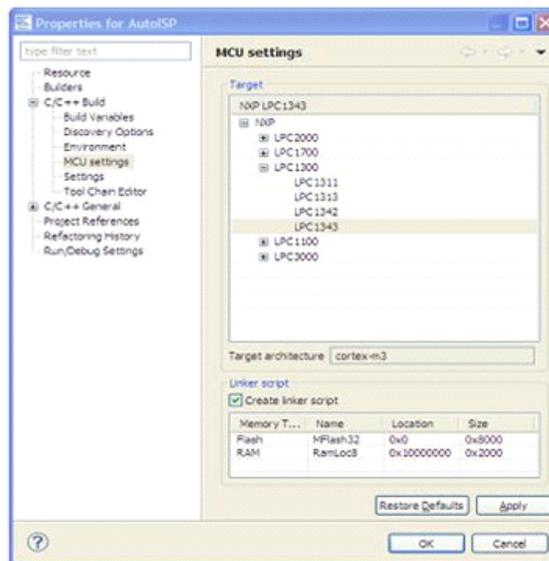
LPCXpresso 的所有属性都是上下文相关的。如果属性被禁止了，或者一些菜单和工具栏按钮被禁止了，双击打开项目管理器中的.c 文件进入。如果你正在编译库项目中的.c 文件，调试特性也是被禁止的。

6.1.2 寄存器显示错误或起始调试错误

确认 MCU 型号是否正确。当前选择的 MCU 型号在 LPCXpresso 界面窗口下面的状态栏中显示。通过按住键盘上的“Ctrl”键并点击可以修改 MCU 型号。在弹出的对话框中选择正确的 MCU 型号。如下图所示。



Current part number



I. Selecting correct part number

6.1.3 优化

优化选项使能后，代码将会被重新排序。也即意味着冗余的 C 代码行将会被重新整合。另外，初始化部分将会被放在程序最顶部，以使它们只被执行一次。这些改变会使用于调试的代码变得混乱。你会看到一些异常情况：断点只能在第一次运行的时候有效，调试的时候程序走向指针指示的位置不正确等。最好的解决办法是：在调试的时候，优化等级设置为 O0。代码优化能够使代码的大小和性能发生很大变化，用代码优化来测试你最终的产品是个很好的办法。

6.1.4 显示汇编指令

单击  图标，在当前指令周围将会显示反汇编观察窗口。

6.1.5 退出调试模式并停止调试

在调试窗口中的工具栏内点击”Stop”按钮（红色方形按钮），即可终止调试并退出调试模式。

6.1.6 程序下载

要想获得最好的编程和调试效果，计算机的操作系统应该是 Vista 或者 Windows 7，在 Windows XP 操作系统下，调试运行的速度明显要慢一些。但是并不影响代码的执行速度，只是软件的响应速度等变慢了。同样需要确认 LPCXpresso 开发板正确的直接连接到 PC 机上。通过 hub、网关或 USB 分线器连接的效果不是很好。

6.2 数据手册阅览器

LPCXpresso 内置了一个集成网页浏览器，在项目中直接点击右下角的 MCU 型号，就可以查看该型号 MCU 的数据手册。如下图所示。

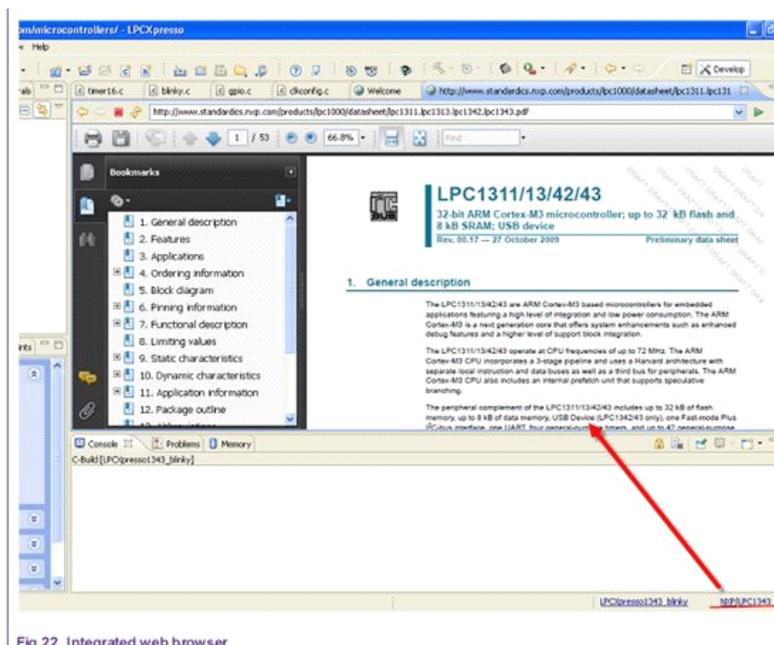
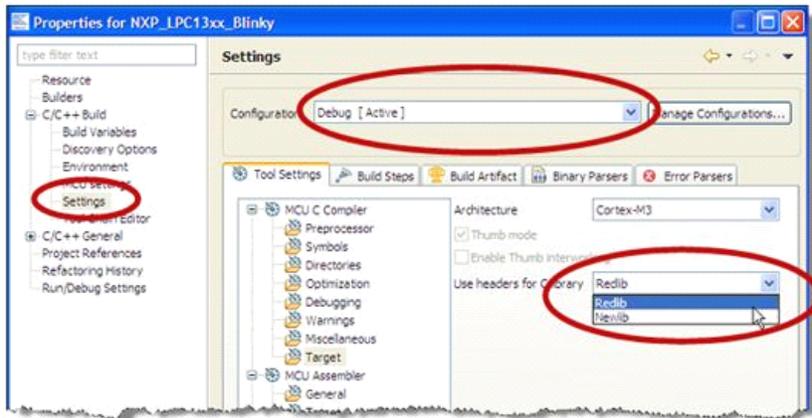


Fig 22. Integrated web browser

6.3 代码长度

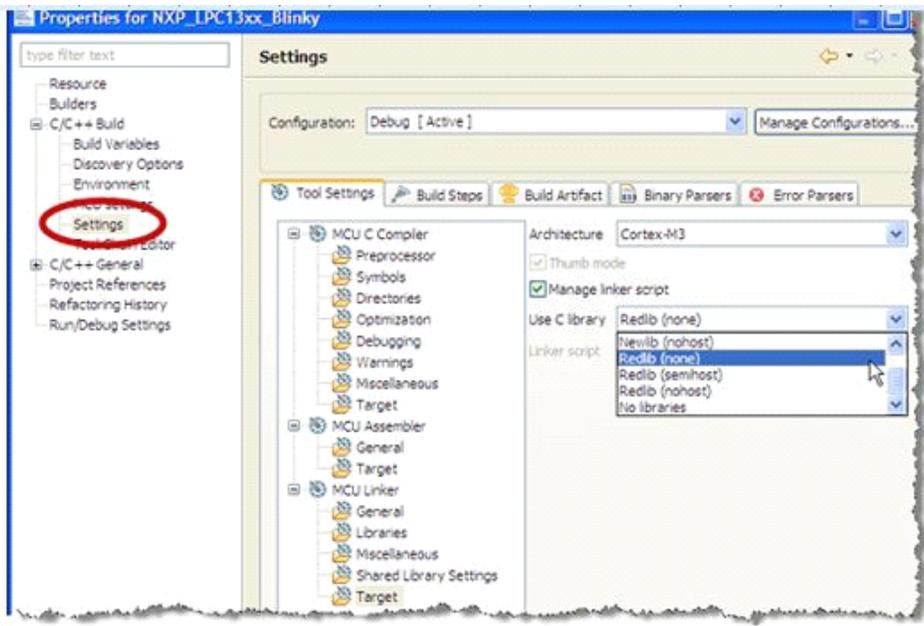
6.3.1 格式输出

当优化项目大小时，如果使用格式输出，必须在项目属性对话框中将 Redlib 选为标准库。这个选项必须在两个位置进行设置：头文件、库。最好也要在调试目标和发布目标这两处进行设置。头文件选项在左侧的树形文件列表的 C/C++ Build 文件夹中。需要确认工具设置标签是否处于激活状态，并且在右侧的树形文件列表中选择 MCU C Compiler 下面的 Target，在设置选框中选择调试或者发布目标，在 C 库头文件下选择 Redlib，这个通常是默认设置。如下图所示。



23. Header file option

在相同的对话框中，选择 MCU Linker 下面的 Target 选项，再次设置链接时的标准库函数为 Redlib。默认设置是 Redlib (semihost)，语序所有 I/O 通过 LPCXpresso 控制台连接到 PC。如下图所示。



24. Reconfigure library setting

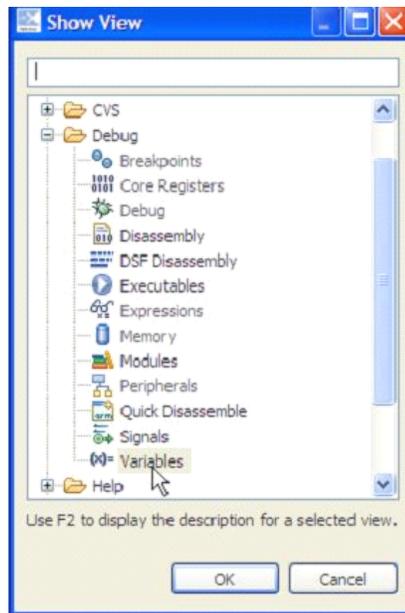
在 Redlib 中执行格式输出是 Newlib 格式输出的一半大小。如果不使用浮点格式字符串的话，可以在 Redlib 中使用更小的格式输出库。如果要选择更小的库，在编译器(i.e. `-DCR_INTEGER_PRINTF`) 中定义符号 `R_INTEGER_PRINTF` 。为保存更大的空间，避免使用格式输出或者任何 C 标准库函数功能，并且选择 Redlib (none) 。根据你的格式输出设置和代码，这样能节省10-20K 的 flash 存储器。

6.3.2 优化

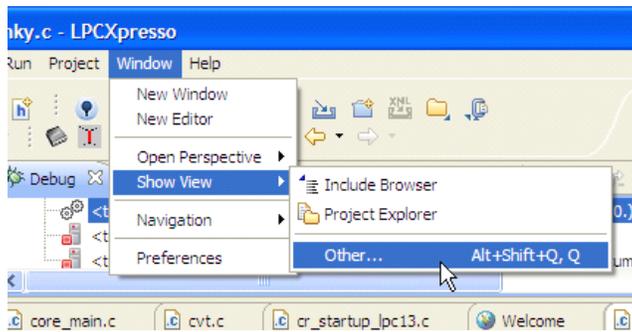
有话可以节省很多 flash 存储器空间。像设置 C 标准库一样在相同的对话框中进行优化设置。在 “Tool Settings” 标签中选择 “MCU C Compiler” 选项，然后在该选项下选择 “Optimization (优化)” 选项。高的优化水平会导致高的性能结果，但是也可能会导致代码量的大幅度增加。最好的方式是：在调试时选择-O0 优化，在发布时选择更高的优化选项。可以试着选择-Os `-mword -relocations` 进行代码的最大优化。使用 GCC 的时候，还有更多可用的优化选项，请访问以下网址获得更多信息：

6.4 显示隐藏窗口

观察窗口在 IDE 环境中显示各种各样的信息，可以显示源代码、项目树、调试窗口等。如果你不小心关闭了其中一些窗口，可以从 Window 菜单中选择 Showing View and Other，并从中选择打开相应窗口。将会弹出一个对话框允许你选择并显示一个窗口。如下图所示。



Show view window

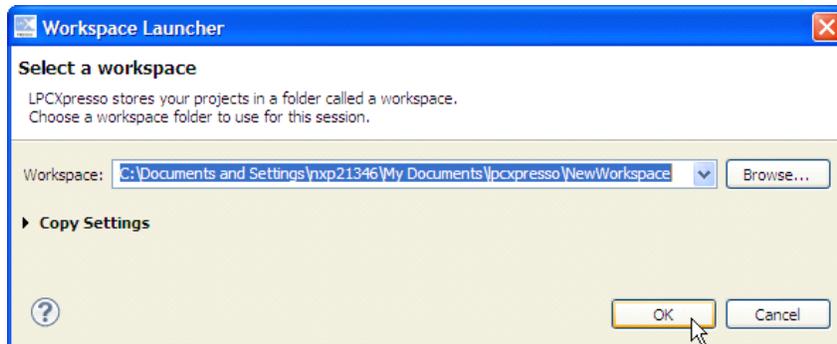


6.5 在新工作区创建一个 ‘skeleton’ 项目

LPCXpresso 包含多种项目模板以帮助快速建立项目文件。

6.5.1 创建一个新工作区

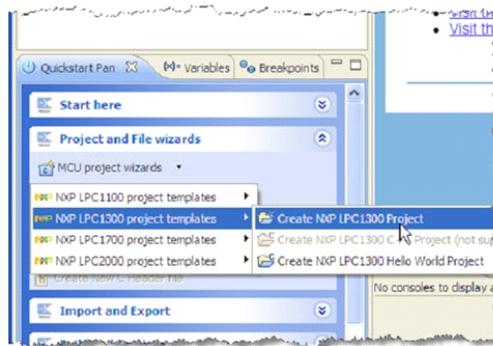
从“File”菜单中选择“Switch Workspace”,然后选择“Other...”。将会弹出一个“Workspace Launcher”对话框。输入或者选择一个新的路径,在此将这个新工作区命名为“NewWorkspace”。如下图所示。



然后单击“OK”完成创建并重新回到带有这个新工作区的 LPCXpresso 环境中。

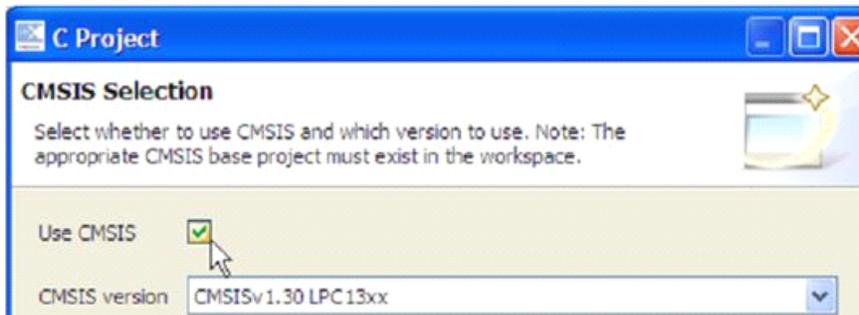
6.5.2 创建“Skeleton”项目

- 在快速启动面板单击“Project and File Wizards”标签。
- 单击“MCU project wizards”，并选择“Creat NXP Project”
- 在弹出的对话框中输入项目名字。将项目名字命名为“MyProject”，然后单击“Next”。



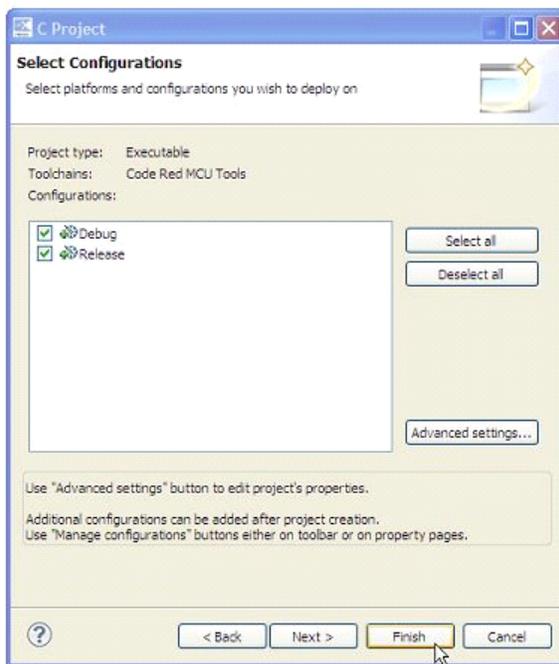
28. Enter project name

如果使用的是基于 Cortex-based 的部分，将会询问是否使用 CMSIS。CMSIS 支持 Cortex 控制器的软件接口标准。CMSIS 定义了一个通用的访问外设寄存器和定义中断的方式。选中“Use CMSIS”复选框并单击“Next”。如下图所示。



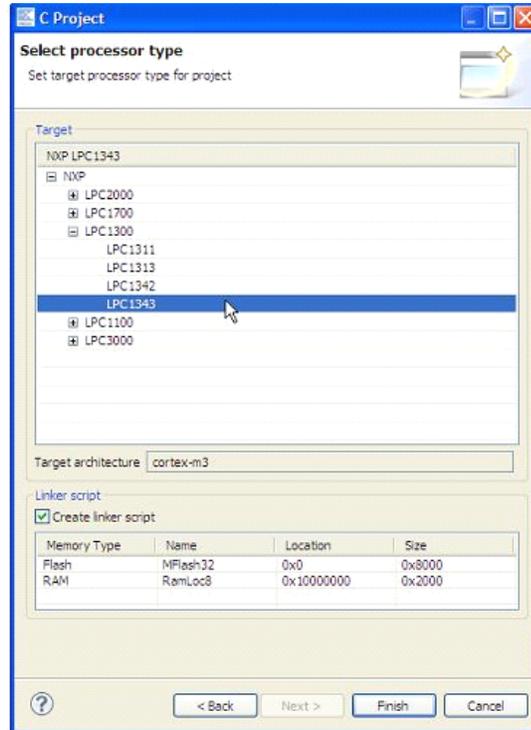
29. CMSIS selection

- 接着弹出询问“源文件信息”选项，输入作者及版本等文本信息。
- 然后弹出一个选择创建编译设置对话框。选择默认设置即可。如下图所示。



1. Select build configuration

- 最后，弹出“SelcetProcessorType”对话框。在这里选择 MCU 的型号。然后单击“Finish”。至此完成了一个项目的创建。本项目中 MCU 型号是 LPC1343。如下图所示。



1. Select processor type

● 在编译“skeleton”项目之前，还有最后一步工作。由于在这个例子中我们选择了CMSIS，所以必须在本工作区中输入CMSIS项目。再次点击“Import Example Project”，找到“CMSIS<version/part>.zip”。将这个项目加入到工作区中，在快速启动面板里单击“Build all projects(Debug)”。

恭喜！你成功的建立了第一个项目！

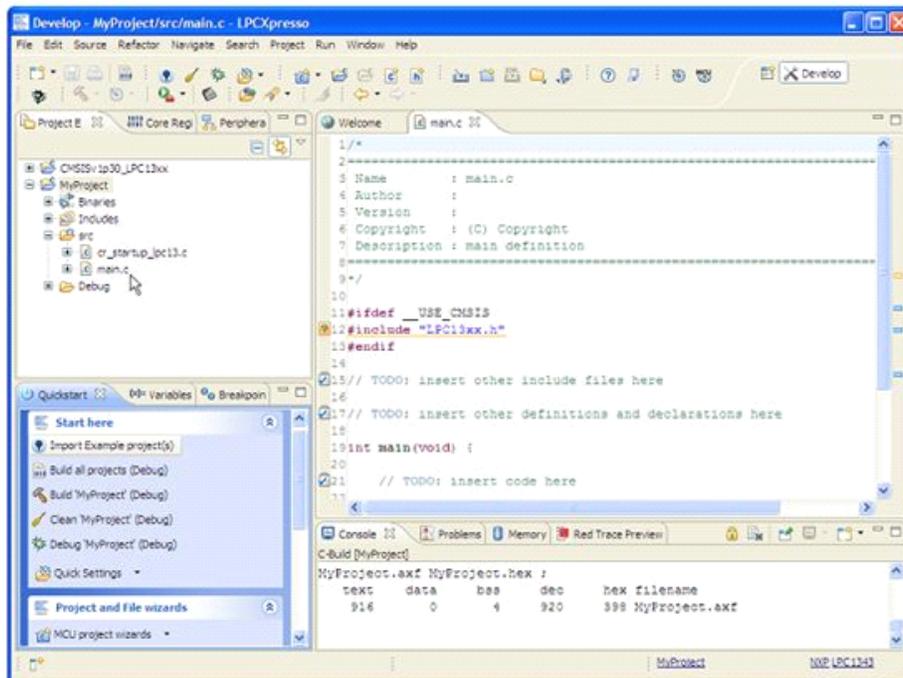
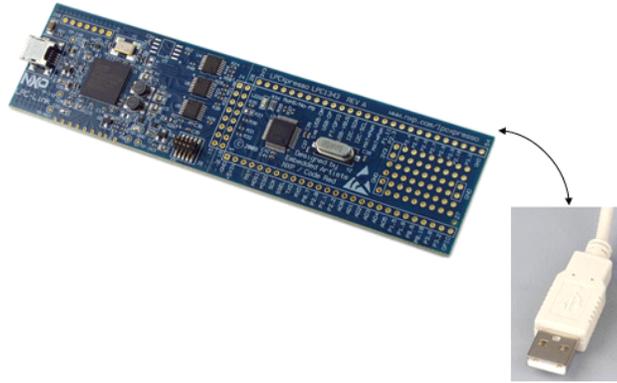


Fig 32. Project creation complete

7 附录

7.1 LPCXpresso PCB 的引线 and 尺寸

LPCXpresso 目标板和 LPC-LINK 的原理图如下所示（略）。LPCXpresso 开发板与 NXP 芯片引脚兼容。LPCXpresso 可以使用三种供电方式：mini-USB 端口供电，3.3V 电压直接向板子供电，5V 电源通过 USB 口供电。10芯 mini JTAG 电缆连接 LPC-LINK 和目标板，这个可以通过 Digi-key 购买。



尺寸：LPCXpresso LPC1343 的外形尺寸是 1.35*5.45 英寸。包含两排间距为 900mil 的焊孔。每一排包含 27 个间距为 100mil 的焊孔。