



—— BF531 开发入门

Getting started with BF531



BF531 开发入门

版本 V 0.0 日 期 6/22/2010

作者：Xiaoyu / XiaomaGee

深入交流 QQ 群：82344271

E-Mail:

Xiaoyu0622@Gmail.com

XiaomaGee@Gmail.com

免责声明

Xiaoyu / XiaomaGee 保留本教程的最终解释权，并不对由于因阅读本文所带来的一切后果（包括商业目的）负责，请大家慎重使用。

文档记录：

持续更新中，请大家关注..... Xiaomagee 2010.6.22

文档提纲：

第一章 BF531 简介

第二章 MS531 介绍

第三章 开发环境

第四章 BF531 内存映射、时钟及电源管理

第五章 可编程标志口

第六章 PPI 接口

第七章 SPI 接口

第八章 Uart 接口

第九章 定时器

第十章 总线

第十一章 DMA

第十二章 Serial Port

第十三章 中断系统

目 录

目 录	2
第一章 BF531 简介	3
一、 概述	5
二、 存储器	5
三、 外设	6
第二章 MS531 介绍	7
一、 工程简介	9
二、 资源介绍	9
三、 SFL 下载器	10
第三章 开发环境	13
一、 引言	15
二、 VISUALDSP++ 简介	15
三、 建立工程 —— LED 实验	21

第一章 BF531 简介

BF531 简介

本章介绍了 DSP 芯片的特点、结构及性能。

本章分为以下几个部分：

- 一、概述
- 二、存储器
- 三、外设

一、概述

Blackfin DSP 处理器是新型 16~32 位嵌入式处理器，基于 ADI 与 Intel 联合开发的微信号架构 (MSA)，将一个 32 位 RISC 型指令集和双 16 位乘法累加 (MAC) 信号处理功能与通用型微控制器所具有的易用性组合在了一起。Blackfin DSP 的体系结构不仅特别适合于完成视频、图像、音频、语音和数据通信的数字信号处理，同时还提供综合的控制能力。

Blackfin 系列 DSP 目前公布的有 BF535，BF531，BF532，BF533。其中 BF535 接口丰富，性能优良，而 BF531/2/3 则增加了视频处理接口，性价比很高。由于其用量很大，BF531 目前在国内价格便宜，并且容易购买。另外由于 BF531 支持外挂 SDRAM，所以在很多大容量数据处理的应用场合，其表现更佳，本文就以 BF531 为主要对象，逐步阐述他的特征、资源及应用。BF531 的特点为：

- 16 位定点 DSP 内核，可以实现 400MHz 的持续工作；
- 高达 600MHz 高性能 Blackfin 处理器：2 个 16 位 MAC，2 个 40 位 ALU，4 个 8 位视频 ALU，以及 1 个 40 位移位器；
- RISC 式寄存器和指令模型，编程简单，编译环境友好，先进的调试、跟踪和性能监视；
- 内核电压 V_{DD} 0.8V~1.2V；
- 片内调压器支持从 3.3V~2.5V 的输入电压；
- 灵活的软件控制动态电源管理；
- 160 引脚 Mini-BGA 封装；169 引脚 PBGA 封装；176 引脚 LQFP 封装。

二、存储器

存储器的性能决定了整个 CPU 运行的效率。BF53X 内部集成了多种可配置的 SRAM，可以用做程序、数据缓存，这样就较好的弥补了 SDRAM 读写速度慢的问题。BF53X 系列 CPU 的存储器特征为：

- 4GB 统一寻址空间；
- 高达 148Kbytes 片内存储器：
 - 16KBytes 指令 SRAM/Cache
 - 64KBytes 指令 SRAM
 - 32KBytes 数据 SRAM/Cache
 - 32KBytes 数据 SRAM
 - 4KBytes 存放中间结果的 SRAM
- L1 数据存储器包括 1 个 16KB 的 Bank，可配置成 SRAM 或 Cache；
- 4KB L1 暂存 SRAM，访问速度与 L1 存储器同样为处理器最快速度；
- 两个双通道存储器 DMA 控制器；
- 存储器管理单元提供存储器保护；

BF531 简介

- 存储器控制器可与 DSRAM、SRAM、Flash 和 ROM 无缝连接；
- 灵活的存储器引导模式，可以选择从 SPI 口或外部存储器启动。

三、 外设

BF531 定位为数字信号处理器，即 DSP，所以其外设没有通用的 CPU（例如现在当红的 ARM）丰富。但是与通用处理器相比，只能说各有长短。BF531 更多的注重于处理音频、视频流。当然他也包含了目前主流的外设如 UART/SPI 等。关并行外设接口（PPI）/GPIO 支持 ITU-R656 视频数据格式：

- 2 个双通道全双工同步串行接口（SPORT），支持 8 个立体 I²S 通道；
- 2 个存储器间的直接存储器存取（DMA）；
- 8 个外围的 DMA；
- SPI 兼容端口；
- 3 个定时/计数器，支持 PWM；
- 实时时钟和“看门狗”定时器；
- 32 位内核定时器；
- 16 个 GPIO；
- 支持 IrDA 的 UART。

对于 BF531 的内部结构，有瘾的兄弟，可以参阅相关文档，这里不作介绍。



LQFP 封装的 BF531，非常方便大家应用，即使两层电路板，也可以跑得很欢。

第二章 MS531 介绍

MS531 工程介绍

本章对 MS531 工程的系统框架、资源及如何下载程序进行了介绍。通过本章的学习，大家可以对 MS531 有个大致的了解，并学会如何通过 SFL 进行程序下载。

本章分为以下几个部分：

- 一、系统框图
- 二、资源介绍
- 三、SFL 下载器

一、工程简介

即使 BF531 简单易用，但是由于其昂贵的开发工具，使很多初学者望而却步（此章有钱人可以绕道，哈哈）。作者去年接了一个项目，摸索出一套方便、廉价的 BF531 开发模式，这时本工程就迫不及待的诞生了。

MS531，取自于 Mini System BF531，即 BF531 的最小系统。虽然是最小系统，但是五脏俱全，系统不但包含了电源管理、大容量存储，更包含了方便快捷的下载器。用户只需要一块 9~24V 直流电源，一个 USB 线即可做 BF531 开发，当然你得有电脑。图 2-1 为 MS531 系统的框图。

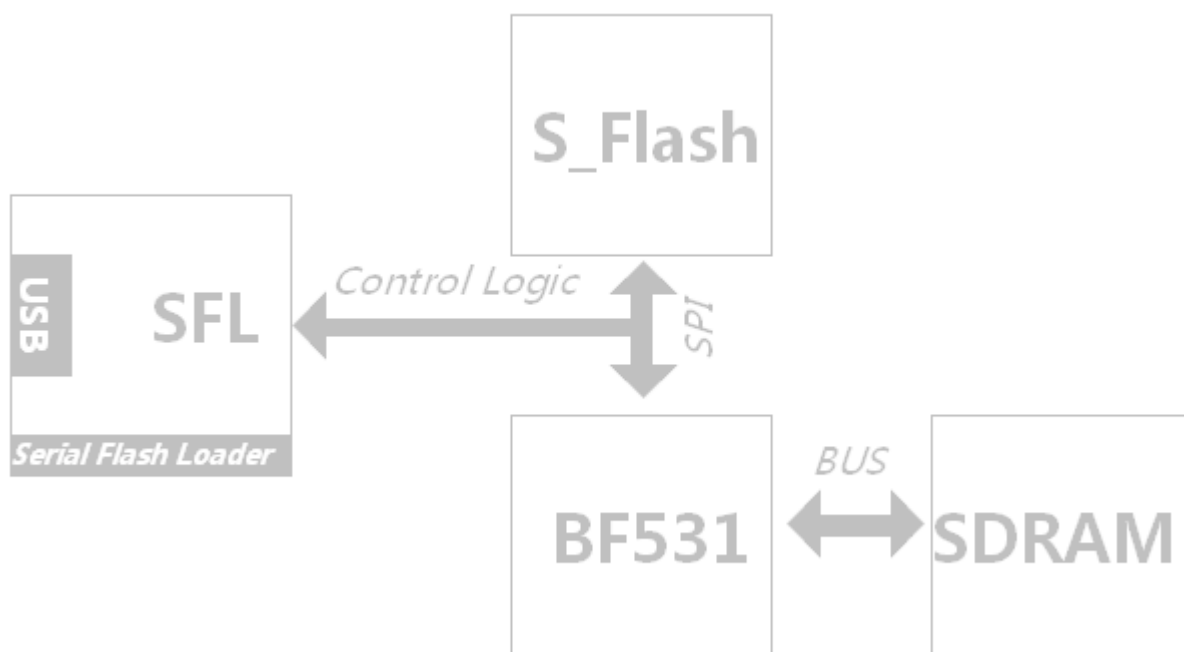


图 2-1 系统框图

二、资源介绍

MS531 系统，定位为“体验”，所以没加什么冗余的功能，不过为了方便大家做实验，大部分硬件资源都用跳线扩展出来了。板子特征为：

- 紧凑设计，只比名片大一点点；
- 独立设计的下载系统，USB 接口即插即用，速度快、效率高；提供 PC 机烧写程序，烧写过程中自动控制 BF531 并在烧写完成后自动复位处理器；具有 USB 连通和烧写指示 LED；
- 绿色电源：宽电压（9~24V）输入开关电源，效率高，不发热；
- 32MByte SDRAM，2M Byte Flash，加上 400M 的主频，轻松体验高速、大容量存储；
- 预留 3.3V 电源和地外扩，方便做实验；

- SPI 接口引出；
- PPI 接口引出；
- PF0~11 接口引出；
- SPORT0 和 SPORT1 接口引出；
- T0~T2 定时器引脚引出；
- UART 接口引出，可通过转换板连接计算机。烧录 Uboot 或 ucLinux 后，可作为终端。

三、 SFL 下载器

SFL (Serial Flash Loader) 下载器是利用 CH341H USB 转 SPI 接口芯片为主控器，利用简单的逻辑控制，达到烧写 SPI flash 的目的。烧写完成后，程序可以立即执行。



SFL.exe 图标如图所示：。它是用 GCC 的 windows 移植版本 Mingw32 开发的基于命令行的软件，软件支持整片擦除、文件烧写、特定扇区烧写等功能。如果嫌繁琐，可以写个批处理文件完成一系列的操作，当然也可以集成在 Visual DSP++ IDE 内，达到编译后立即烧写的目的。软件源代码清晰易懂，并且开源，大家可以在这个基础上增加支持的芯片。

加入把软件 SFL.exe 拷贝到 D:盘根目录，我们可以从命令行操作它。点击“开始” - “运行”，输入 CMD.EXE 后，我们就得到了命令行窗口。

键入 D:，在 D 盘下找到 SFL.EXE。

键入 `sfl --help` 后，得到软件的一些编译日期信息以及简单的帮助。

```
D:\>sfl --help

Serial Flash Loader 0.04, Mon Jun 21 17:58:18 2010

By XiaomaGee. XiaomaGee@Gmail.com. // QQ:15959622

Usage: sfl [--help] [--version] [--download filename <-sx>] [--erase
all]
Note: download filename in specified sector, Example "sfl --download
a.bin -s5"
```

键入 `sfl --version` 后, 得到软件的版本。

```
D:\>sfl --version

Serial Flash Loader 0.04, Mon Jun 21 18:07:37 2010

By XiaomaGee. XiaomaGee@Gmail.com. // QQ:15959622

Serial Flash Loader.
Version: 0.04 / Mon Jun 21 18:07:37 2010
By XiaomaGee. XiaomaGee@Gmail.com.
```

如果我们要烧写一个由 Visual DSP ++ 生成的 .ldr 二进制文件, 可以先把它拷贝到 D: 盘根目录下, 然后键入下面的命令。

```
D:\>sfl --download a.ldr

Serial Flash Loader 0.04, Mon Jun 21 18:11:00 2010

By XiaomaGee. XiaomaGee@Gmail.com. // QQ:15959622

-----
Flash id is: 0x202015
File size=3164 Byte.
Erase Sector:
0 Done!
Starting write.....
Write done!
Starting verify.....
Verify ok!
-----
```

软件分别把 SPI flash 的 ID 号, 文件大小均打印出来。并先擦除扇区、然后写入最后再校验, 分三步完成一次烧写。具体擦除哪个扇区, 是由起始扇区和文件大小决定的, 用户不用考虑此问题。

MS531 介绍

如果用户需要下载一个汉字库到第 20 个扇区，用来被程序调用，我们可以采取如下的方法：

```
D:\>sfl --download hzk12 -s20

Serial Flash Loader 0.04,Mon Jun 21 18:16:22 2010

By XiaomaGee. XiaomaGee@Gmail.com. // QQ:15959622
-----
Flash id is: 0x202015
File size=261696 Byte.
Erase Sector:
20 21 22 23 Done!
Starting write.....
.....
.....
.....
Write done!
Starting verify.....
.....
.....
.....
Verify ok!
-----
```

如果要全部擦除，需要这样操作。整片擦出是个漫长的过程，大概需要十几秒，当然每种芯片需要的时间都不一样。

```
D:\>sfl --erase all

Serial Flash Loader 0.04,Mon Jun 21 18:28:03 2010

By XiaomaGee. XiaomaGee@Gmail.com. // QQ:15959622
-----
Flash id is: 0x202015
erasing flash...Done!
```

第三章 开发环境

开发环境

通过本章，您可以了解到实验板的 VisualDSP++ 软件开发部分，通过 VisualDSP++ 5.0 的简介，以及 LED 实验的工程建立、配置、编译下载等，每一步都有图片配合指导，就算大家没有任何基础，都可以顺利的完成软件开发部分的内容。

本章分为以下几个部分：

- 一、引言
- 二、VisualDSP++ 简介
- 三、建立工程 —— LED 实验

一、引言

在建立工程之前，将我们的实验板通过 USB 线连接到电脑上，也许这个时候会弹出 USB 不能用的窗口，如果遇到这个问题，直接从群共享里下载 CH341PAR.exe，点击安装，再次插上 USB 线即可开始正常使用该实验板了。

二、VisualDSP++ 简介

VisualDSP++ 是 ADI 公司针对 ADI 公司的 DSP 器件而专门开发的一种使用方便的开发平台，它支持 ADI 公司所有系列的 DSP 处理器，包括 Blackfin 系列和 ADSP-21XX 系列定点处理器、SHARC 系列和 TigerSHARC 系列的浮点处理器的各种型号处理器，是一个安装简便、易于操作的集成软件开发调试环境（IDDE）。它可以在单一界面中对软件项目自始至终地进行有效管理。它能够在编辑、构建和调试活动中进行快速、轻松的切换。关键特性包括：自带的 C/C++ 编译器、先进的绘图工具、特性统计图表和 VisualDSP++ Kernel（VDK）。其他性能还包括汇编器、链接器、库、分离器、精确周期及精确功能编译的模拟器、仿真器支持等多性能。其主要结构包括：

- 集成了 VisualDSP++ 内核的集成编译和调试环境（IDDE）；
- 带实时运行库的 C/C++ 优化编译器；
- 汇编器和链接器；
- 仿真软件和程序例程。

下面主要针对 VisualDSP++ 5.0 的安装及图形界面做一简单介绍。

VisualDSP++ 5.0 的下载地址：

http://www.analog.com/en/embedded-processing-dsp/software-and-reference-designs/content/visualdsp_tools_upgrades/fca.html

运行安装程序 VisualDSP++ 5.0，出现图 3-1 所示的确认安装对话框。

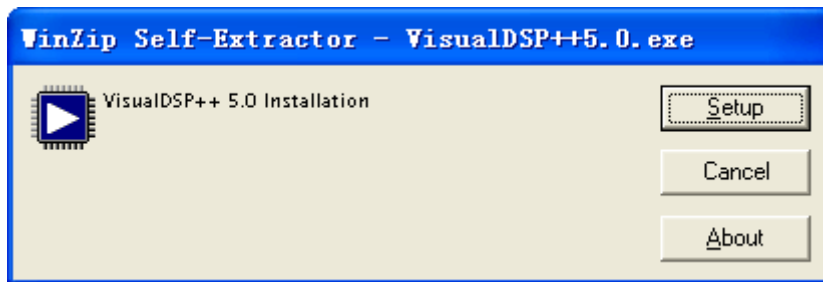


图 3-1 确认安装对话框

单击“Setup”后，将会出现图 3-2 所示的解压对话框。

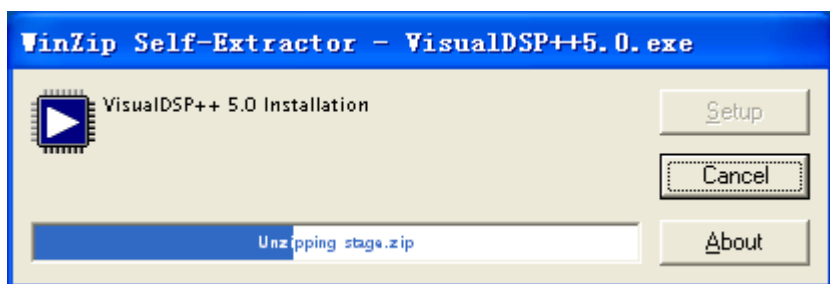


图 3-2 解压对话框

解压完后，将弹出图 3-3 所示的欢迎安装对话框。

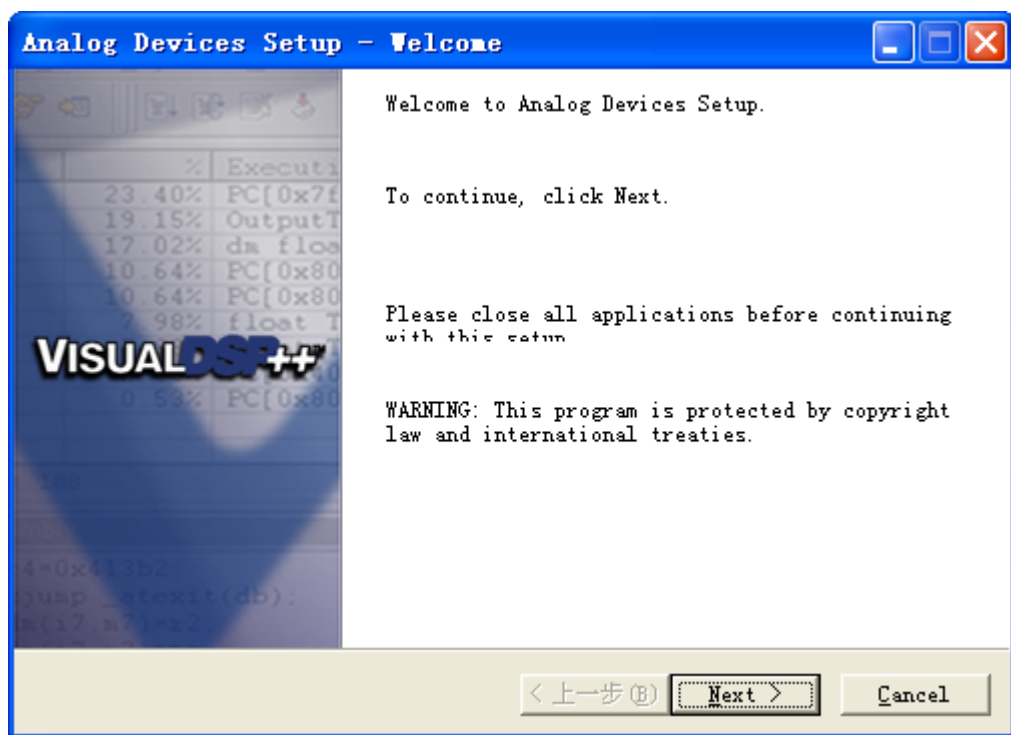


图 3-3 欢迎安装对话框

单击“下一步”后，又将出现一个对话框，如图 3-4 所示单击“Next”。，将出现确认接受协议对话框，如下图所示。选中“I accept the terms in the license agreement”后。

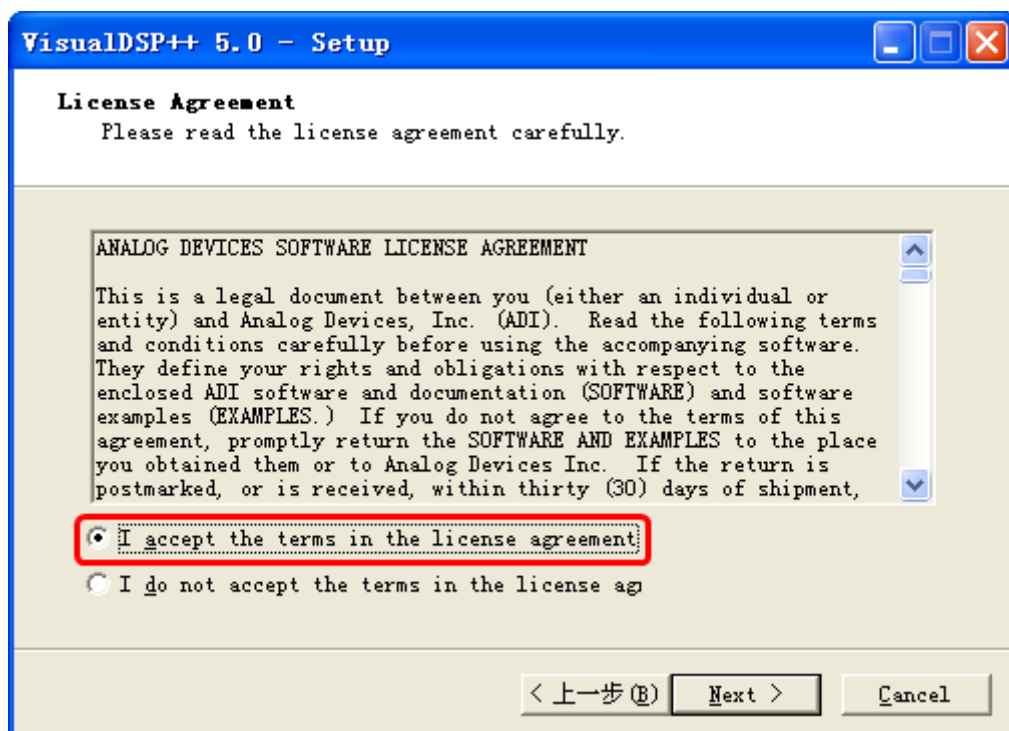


图 3-4 确认接受协议对话框

单击“Next”，弹出选择用户名及单位名对话框，如图 3-5 所示，填入用户名及单位名后，继续单击“Next”。

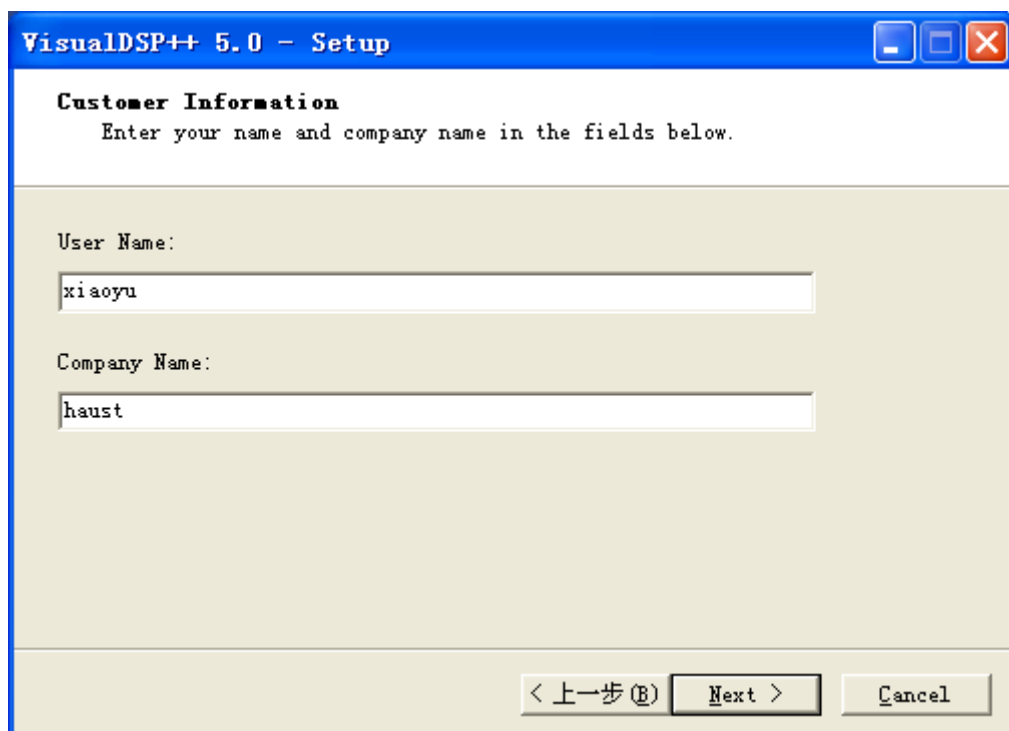


图 3-5 选择用户名及单位名对话框

弹出如图 3-6 所示的选择安装路径对话框,选择安装路径或者使用默认路径,单击“Next”。

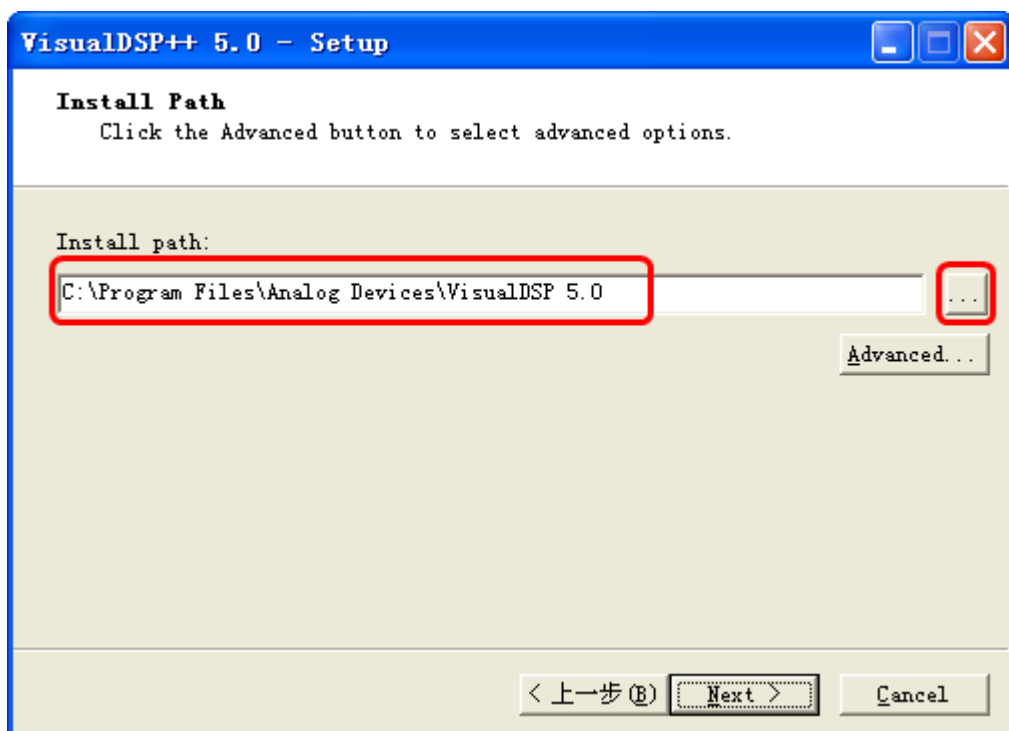


图 3-6 选择安装路径对话框

弹出图 3-7 所示的对话框。

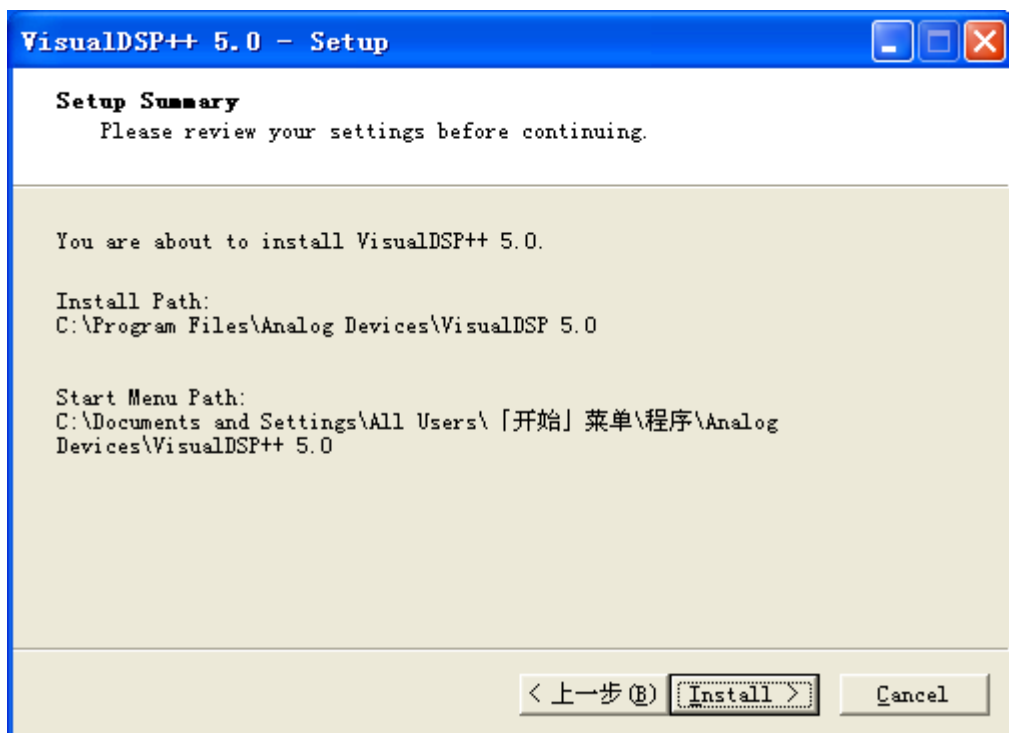


图 3-7

点击其中的“Install”，便开始安装 VisualDSP++ 5.0。直到出现完成安装对话框，如图 3-8 所示。

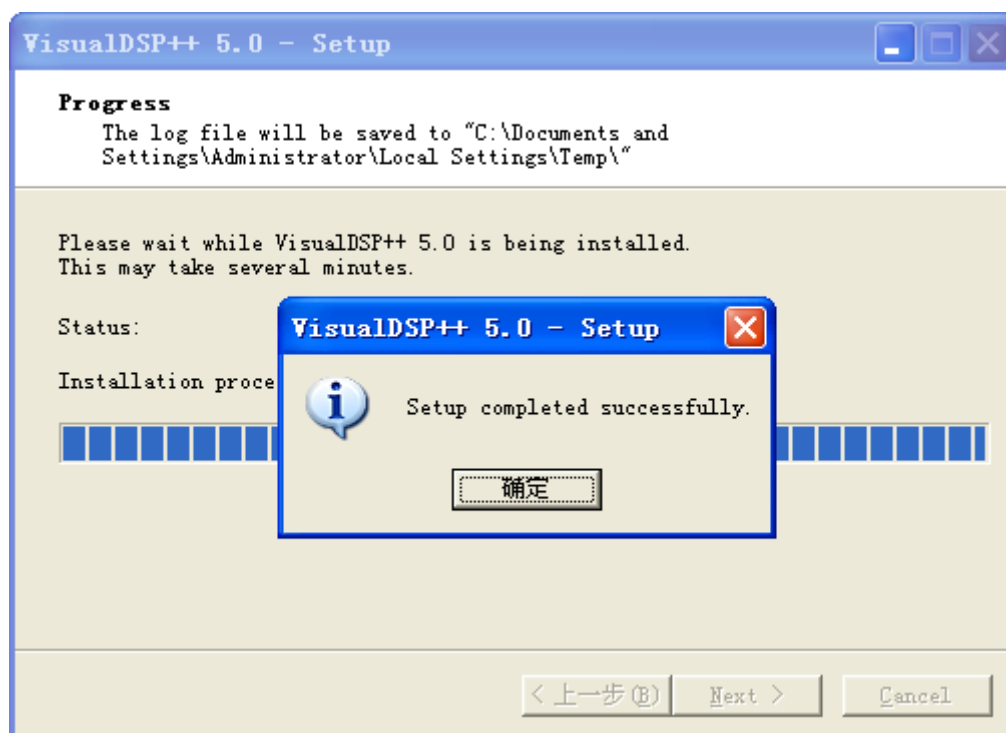


图 3-8 完成安装对话框

单击“确定”后，弹出安装完成窗口，直接单击“Next”和“完成”，完成 VisualDSP++ 5.0 的安装。

至此然后直接双击开始菜单中的 Analog Devices->VisualDSP 5.0-> VisualDSP++ Environment 即可进入 VisualDSP++ 的编程界面，为了使用方便，一般将 VisualDSP++ Environment 发送到桌面快捷方式，如图 3-9 所示。

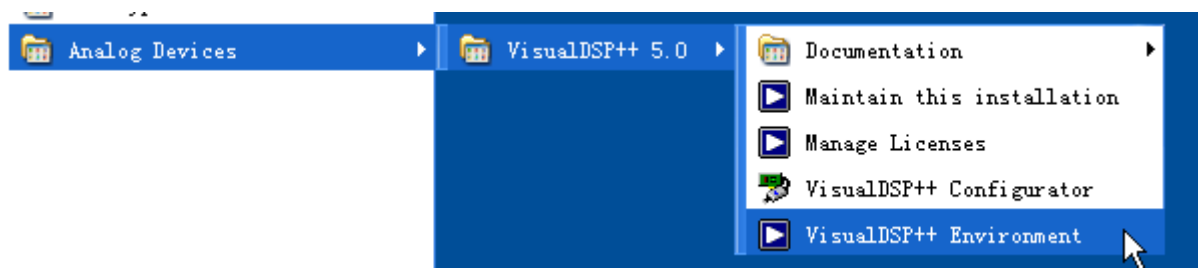


图 3-9 VisualDSP++ 菜单

Analog Device 对 VisualDSP++ 一直做更新包，更新包的扩展名为 *.vdu，大家下载更新包后，可以通过菜单来的 Maintain this installation 升级您的软件。如图 3-10 和 3-11 所示。

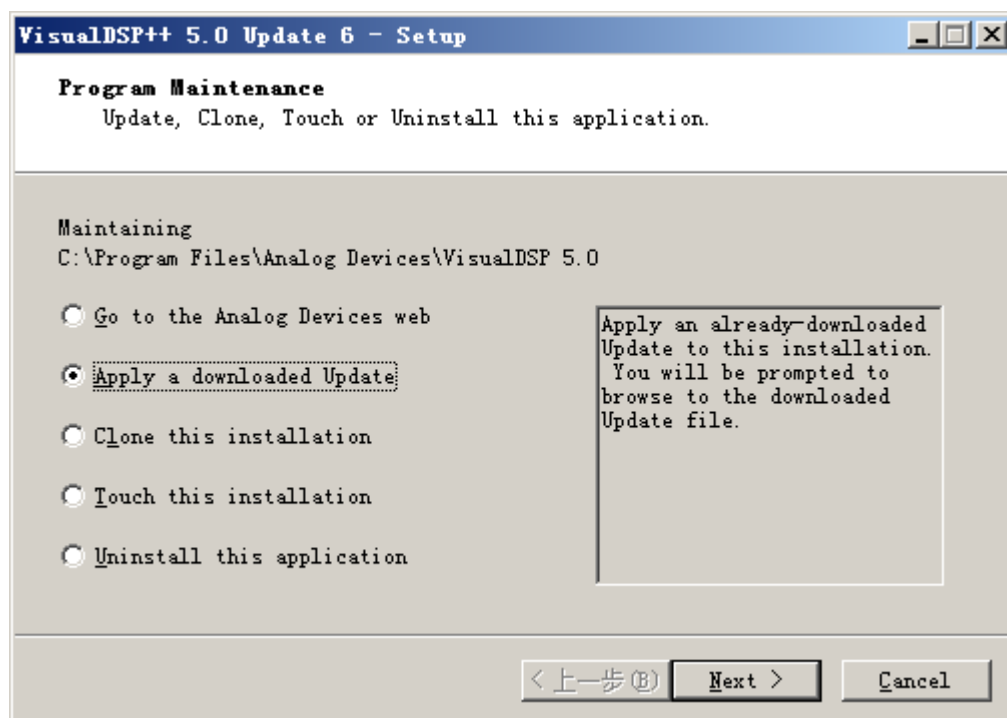


图 3-10 选择应用一个更新包

点击“next”，并选择更新包的路径，一路“next”即可。

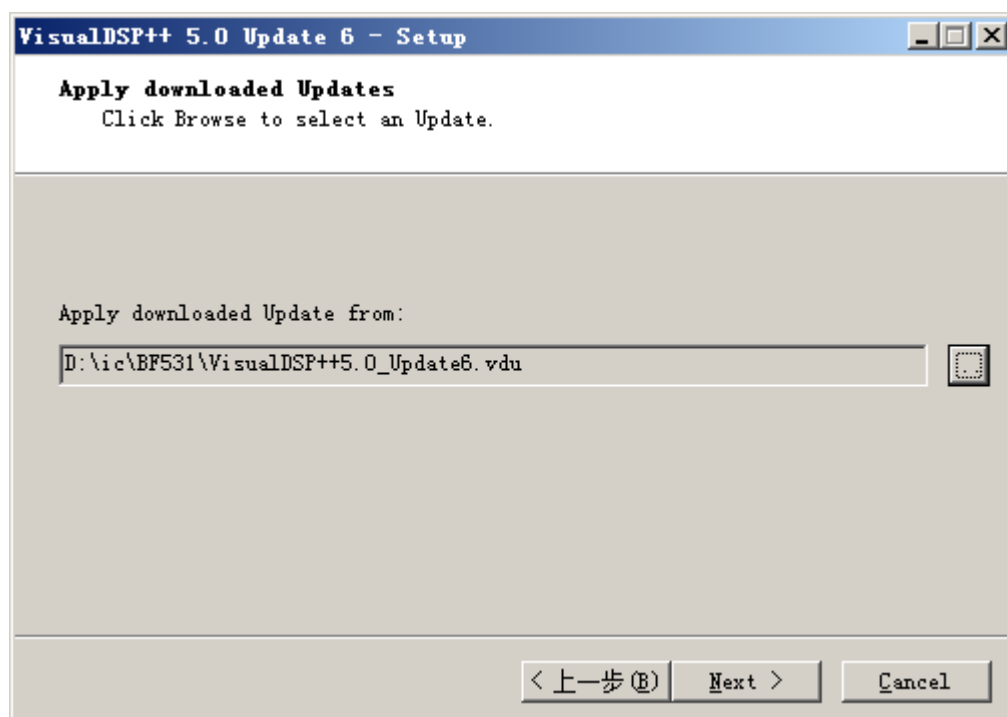


图 3-11 选择应用一个更新包

三、建立工程 —— LED 实验

首先，将 VisualDSP++ 5.0 IDE 软件打开，打开后 VisualDSP++ 的界面赫然显现在我们面前，界面很简单，跟其他的 IDE 没什么太大的区别，特别是熟悉 VC++ 的朋友，看这个界面肯定更亲切。我们首先需要做的就是建立一个软件工程，操作方式如图 3-12 所示，点击 File->New->Project，VisualDSP++ 将启动新建工程向导，帮助用户逐步建立新工程。

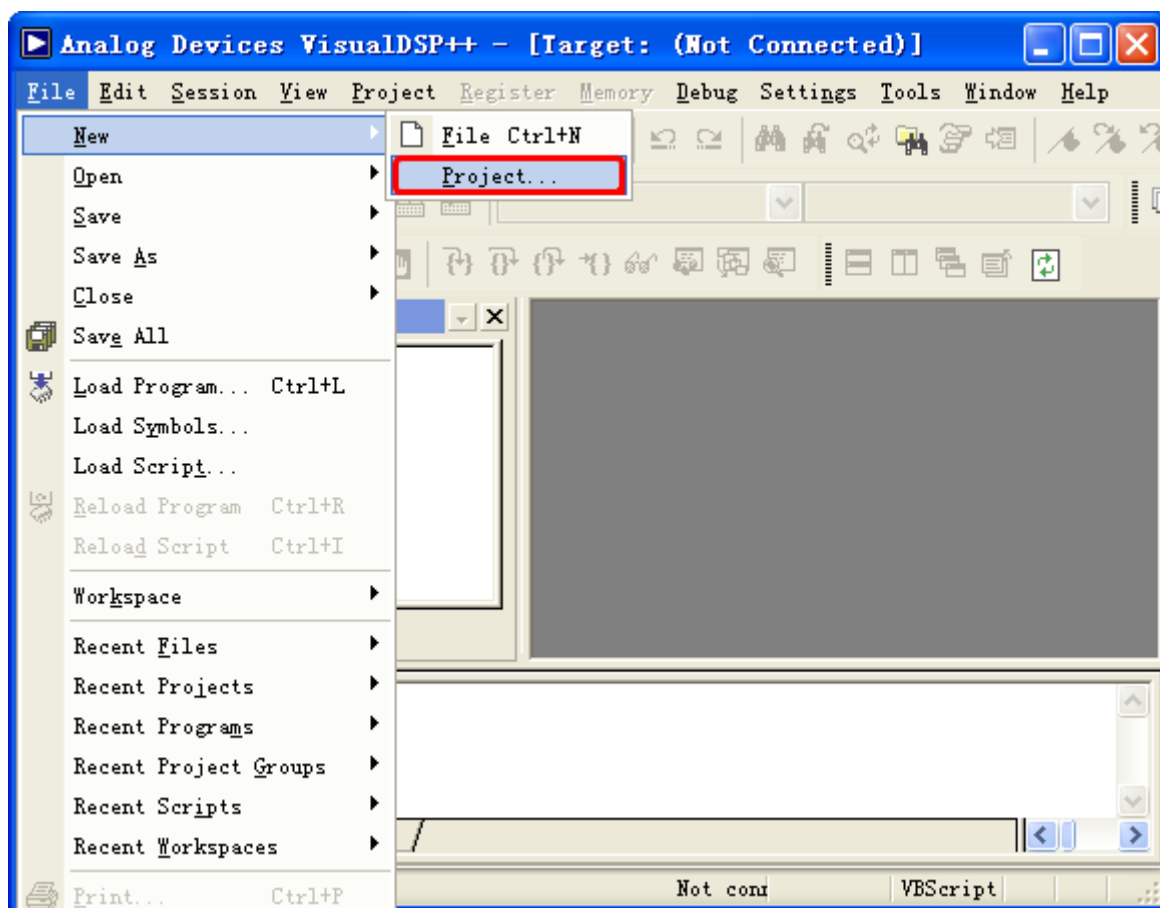


图 3-12

点击后，会出现工程向导界面，新建工程向导主要包含两个部分：“常规(General)”和“输出类型(Output Type)”。新建工程向导的常规信息窗口如图 3-13 所示，主要包含了所建工程的名称、路径和工程类型等。红圈 1 和红圈 2 处是工程类型选择，工程的类型有四种，分别为：“Standard application”、“Library”、“Multi-threaded application using VDK”和“TCP / IP Stack application using LwIP and VDK”四种，用户根据自己需要进行选择，系统默认为“Standard application”，本书也将按照“Standard application”进行讲解。工程的名称和路径由用户自己设置，红圈 3 处是工程名称，我将其修改为 LedTest，红圈 4 处是工程存放的位置，我将其存放在 F:\ms531\LED_TEST\LedTest 处。

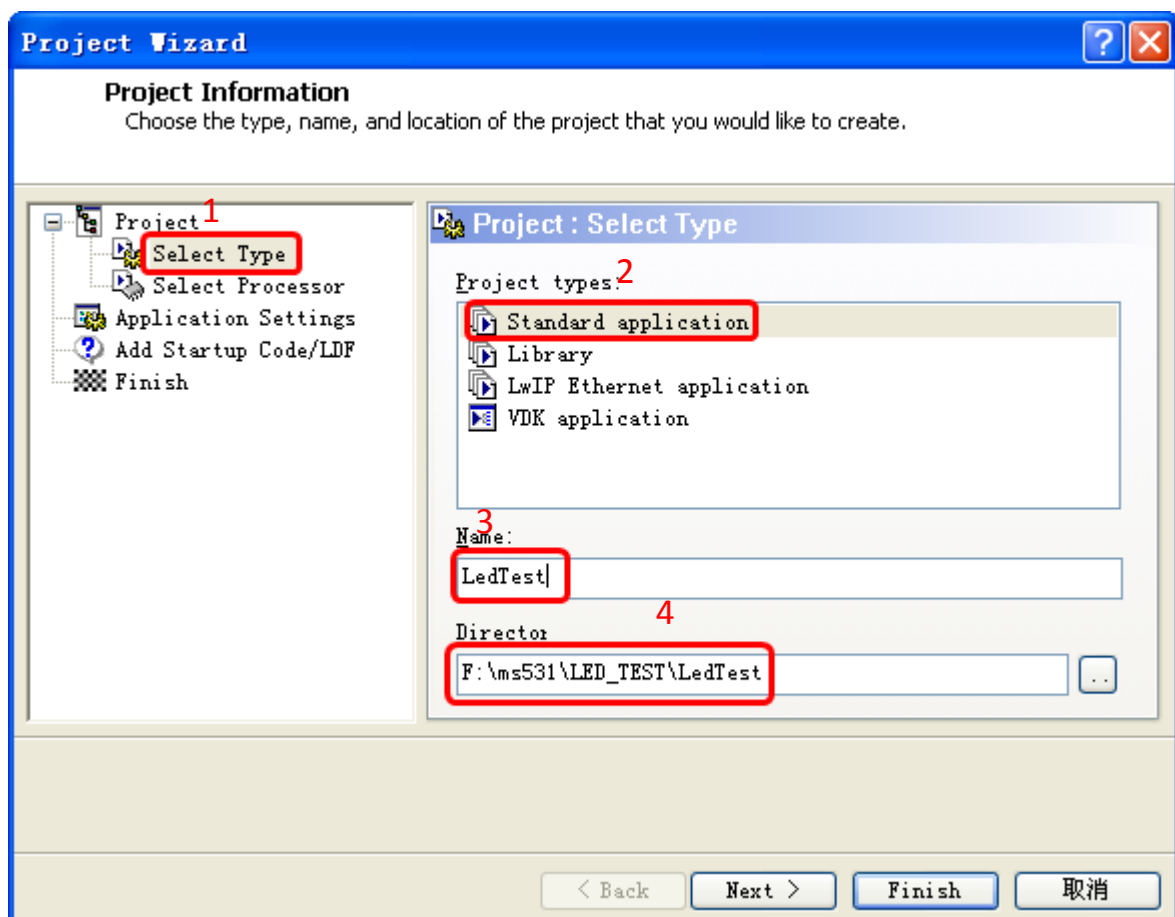


图 3-13 工程向导界面

在将工程名称、路径和类型设置完成后，单击“NEXT”按钮，会出现图 3-14 的确认窗口。

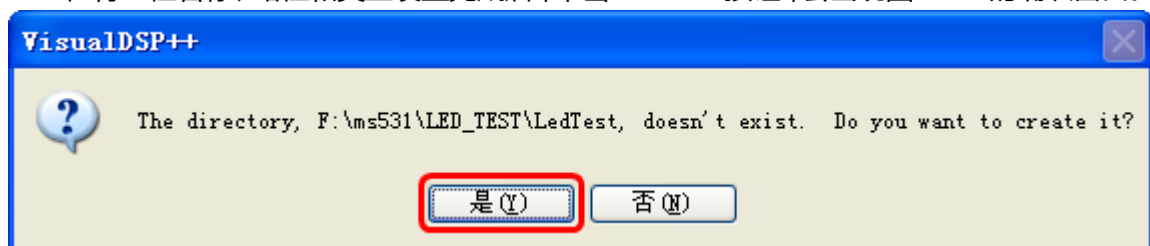


图 3-14 确认窗口

然后直接点击“是”，创建存放项目的位置，VisualDSP++将显示工程选项设置窗口，如图 3-15 所示。在输出类型窗口中主要是设置工程采用的处理器类型(Processor types)、芯片版本号(Silicon)和工程输出文件类型(Project output)等。

处理器类型窗口用于可以选择相关的处理器，该窗口中包含了 ADI 公司 2007 年以前的所有处理器类型。

芯片版本号选项将随所选的处理器信号变化而变化，包含 2007 年所有处理器的芯片版本

号，另外还附加了自动(Automatic)、无(NONE)和任意(any)三个选项，在芯片版本号未知的情况下可以选择自动或者任意。

由于我们实验板选用的是 BF531，所以在此选中红圈 1 中的 Blackfin 和红圈 2 处的 ADSP-BF531。

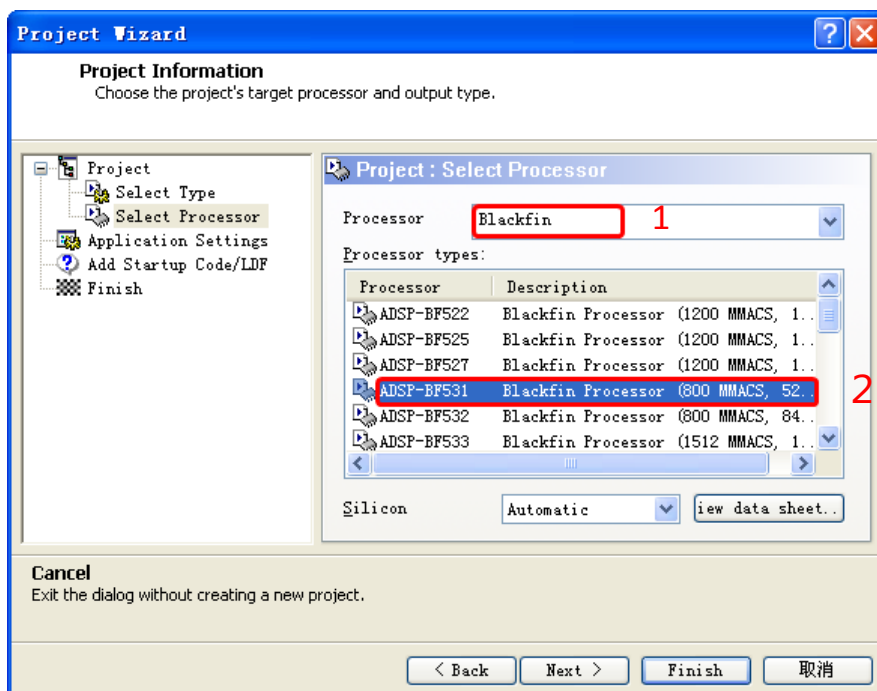


图 3-15 处理器选择界面

点击 Next，进入设置界面，如图 3-16 所示。红圈 1 处模板代码的语言选择上，仍保留原来的 C 语言，此处不做修改。红圈 2 处为工程输出文件选项，该选项是用来设置工程输出的文件为处理器可执行的文件(Executable File)或者处理器加载文件(Load File)。如果用户的工程处在调试阶段，那么一般将该选项设置为处理器可执行的文件，以方便模拟器或者仿真器进行调试，而如果用户的工程处于开发完成阶段，那么一般将该选项设置成加载文件，加载文件可以用于对处理器进行程序加载。本项目将输出类型选为 Load file (.ldr)。

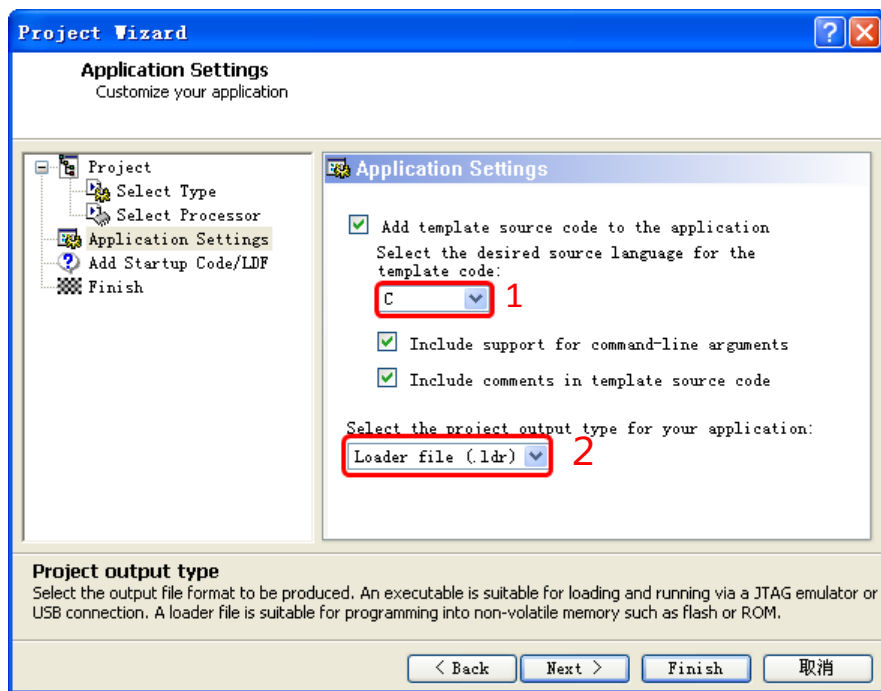


图 3-16 设置界面

点击 Next，进入添加开始代码界面，如图 3-17 所示，此处我们选中红圈处的内容。

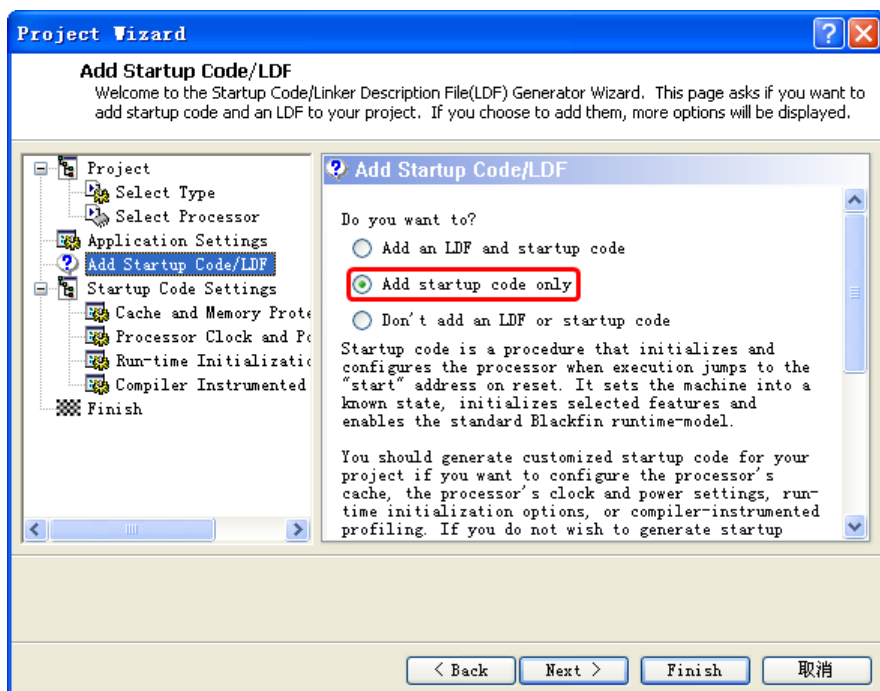


图 3-17 添加开始代码界面

此处可以直接点击 Finish 完成工程向导；也可以一步步的按照默认效果进行设置。如果选择一步步设置，那么点击 Next，直接进入缓存和存储器保护的设置页面，如图 3-18 所示。

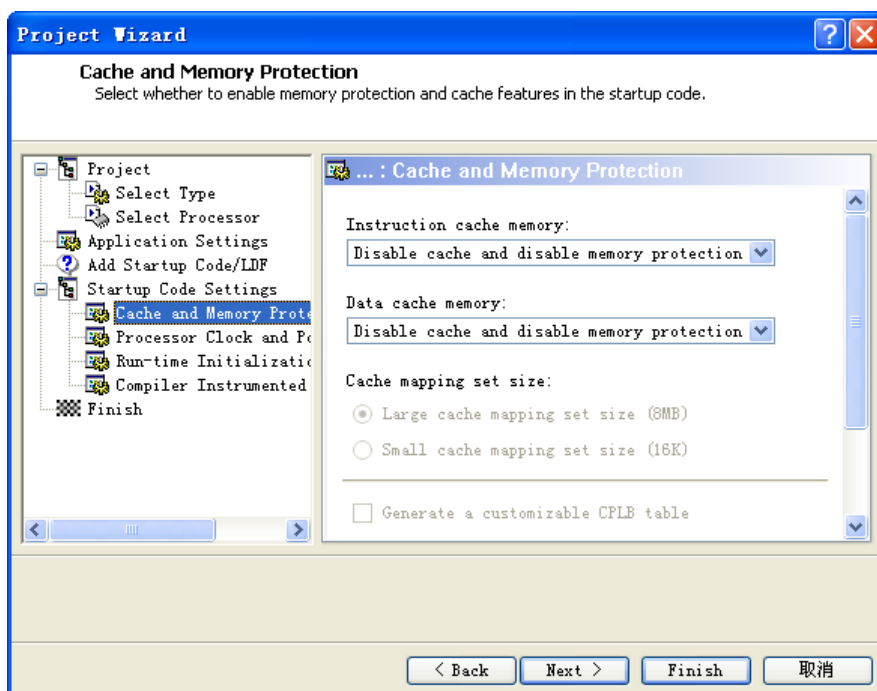


图 3-18 缓存和存储器保护的设置页面

这里不做修改，直接点击 Next，进入处理器时钟及电源设置页面，如图 3-19 所示。

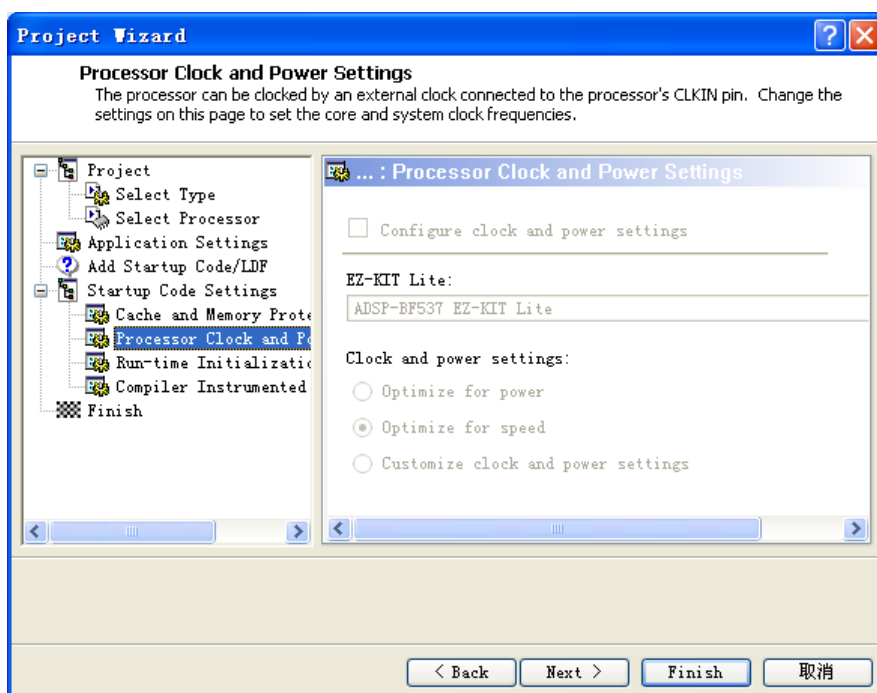


图 3-19 处理器时钟及电源设置页面

这里仍不做修改，直接点击 Next，进入初始化设置页面，如图 3-20 所示。

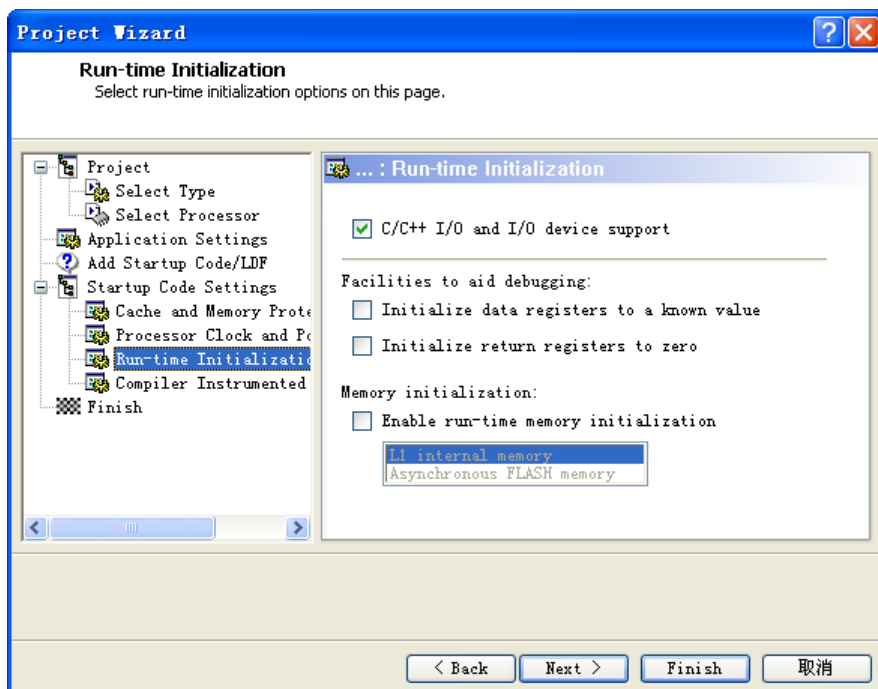


图 3-20 初始化设置页面

这里仍不做修改，直接点击 Next，进入编译器设置界面，如图 3-21 所示。

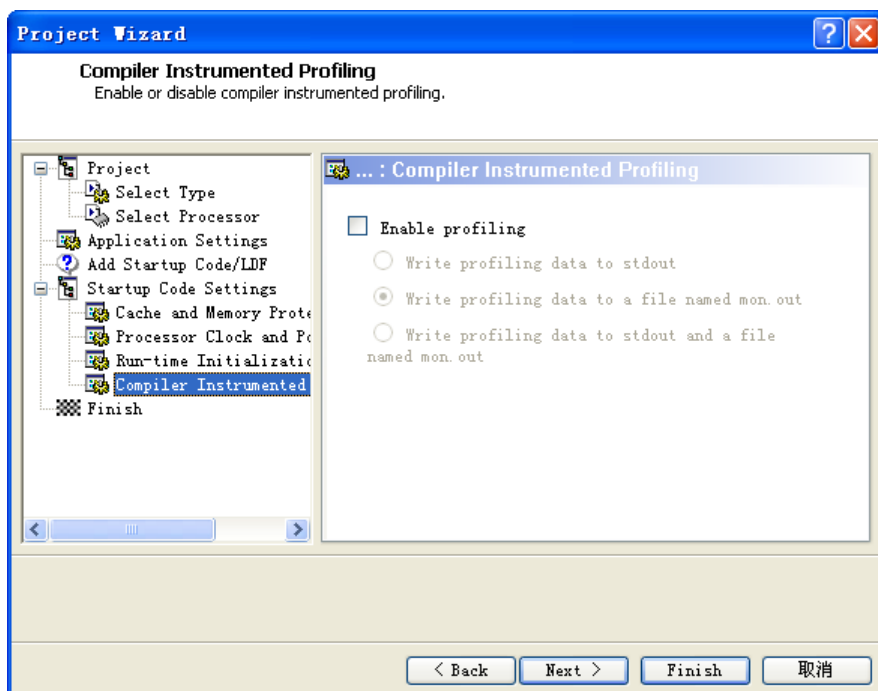


图 3-21 编译器设置界面

这里仍不做修改，单击“NEXT”按钮则将显示出用户建立的工程信息，如图 3-22 所示。如果用户确认信息无误，则单击该窗口中的“Finish”按钮将完成工程的建立，如果用户需要

修改工程的某些参数，那么通过单击“Back”按钮可以回到前面的窗口重新进行设置。

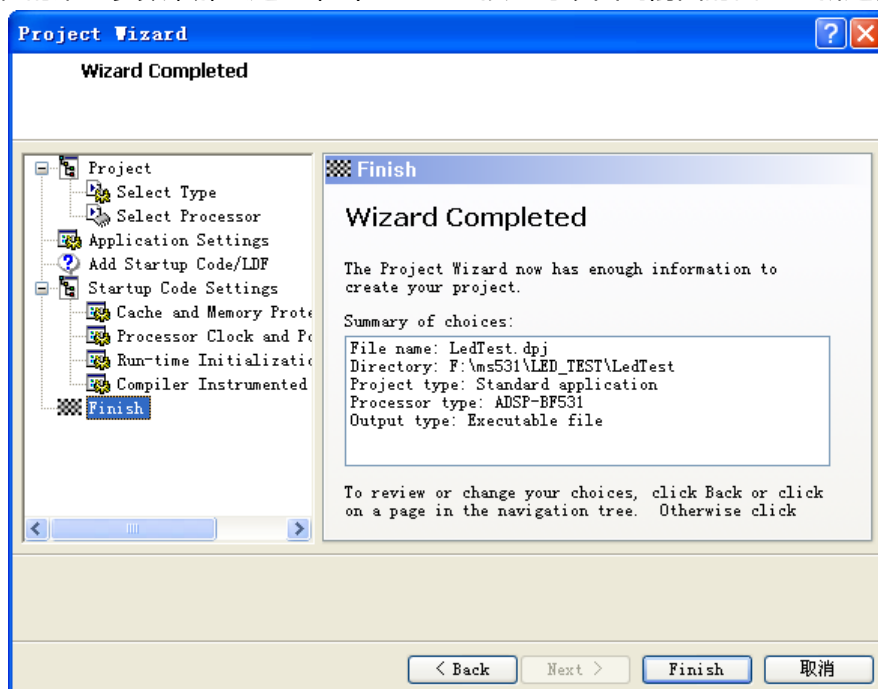


图 3-22 工程完成界面

点击 Finish，完成工程向导，正式进入 Visual DSP++ 的界面了，如图 3-23 所示。主要介绍三个部分，按功能将这三部分分别命名为工程目录窗口、代码窗口及信息输出口。

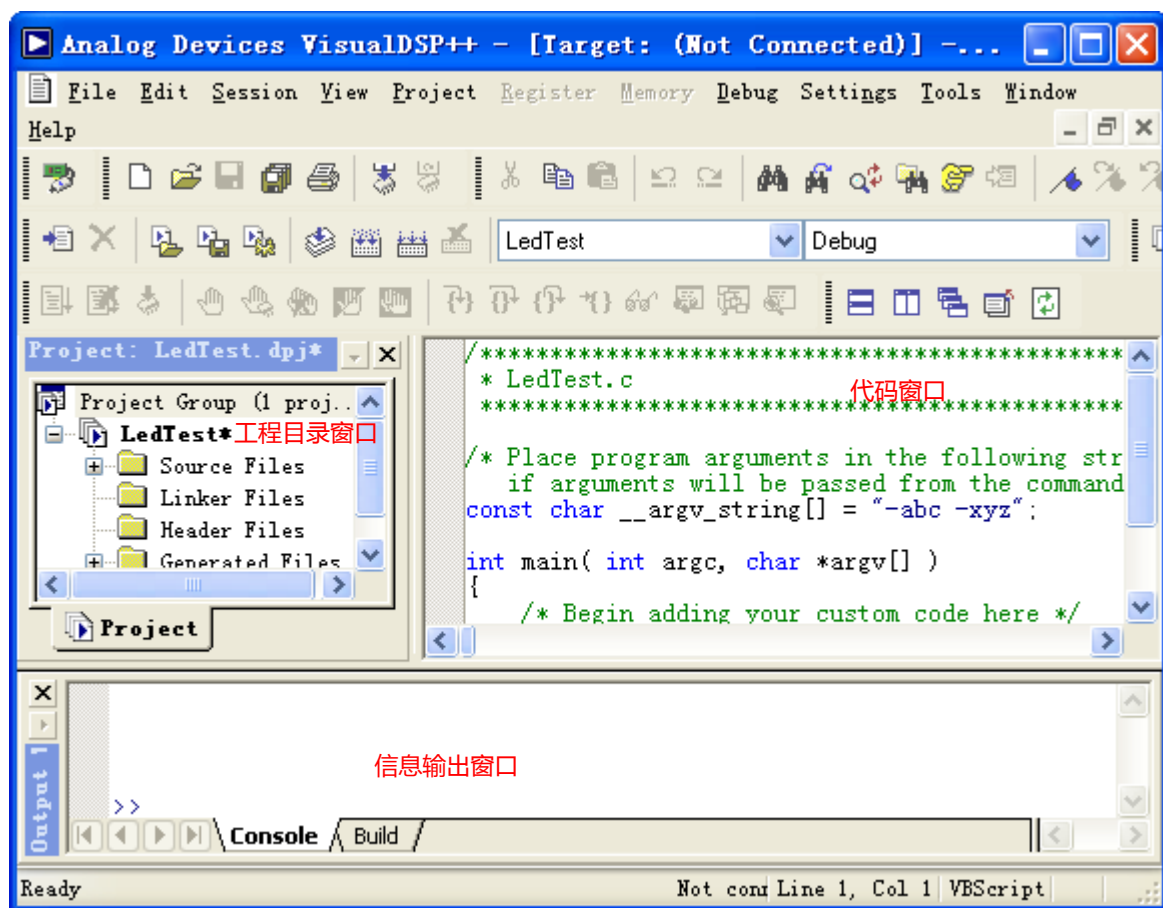


图 3-23 Visual DSP++界面

很明显，代码窗口就是显示代码的，工程目录窗口显示所有与工程有关系的文件，跟我们有关系有.c 文件。信息输出窗口有两个栏，一个是 Console，一个是 Build。第一个是编译信息显示区，这里先不多说了，后面的工程编译中我们再做介绍。

接下来，进行工程编译，如图 3-24 所示，点击 1 处的红圈中的图标，或者选中菜单 Project->Rebuild Project 对整个工程进行编译。红圈 2 中的信息窗口表明工程编译通过。

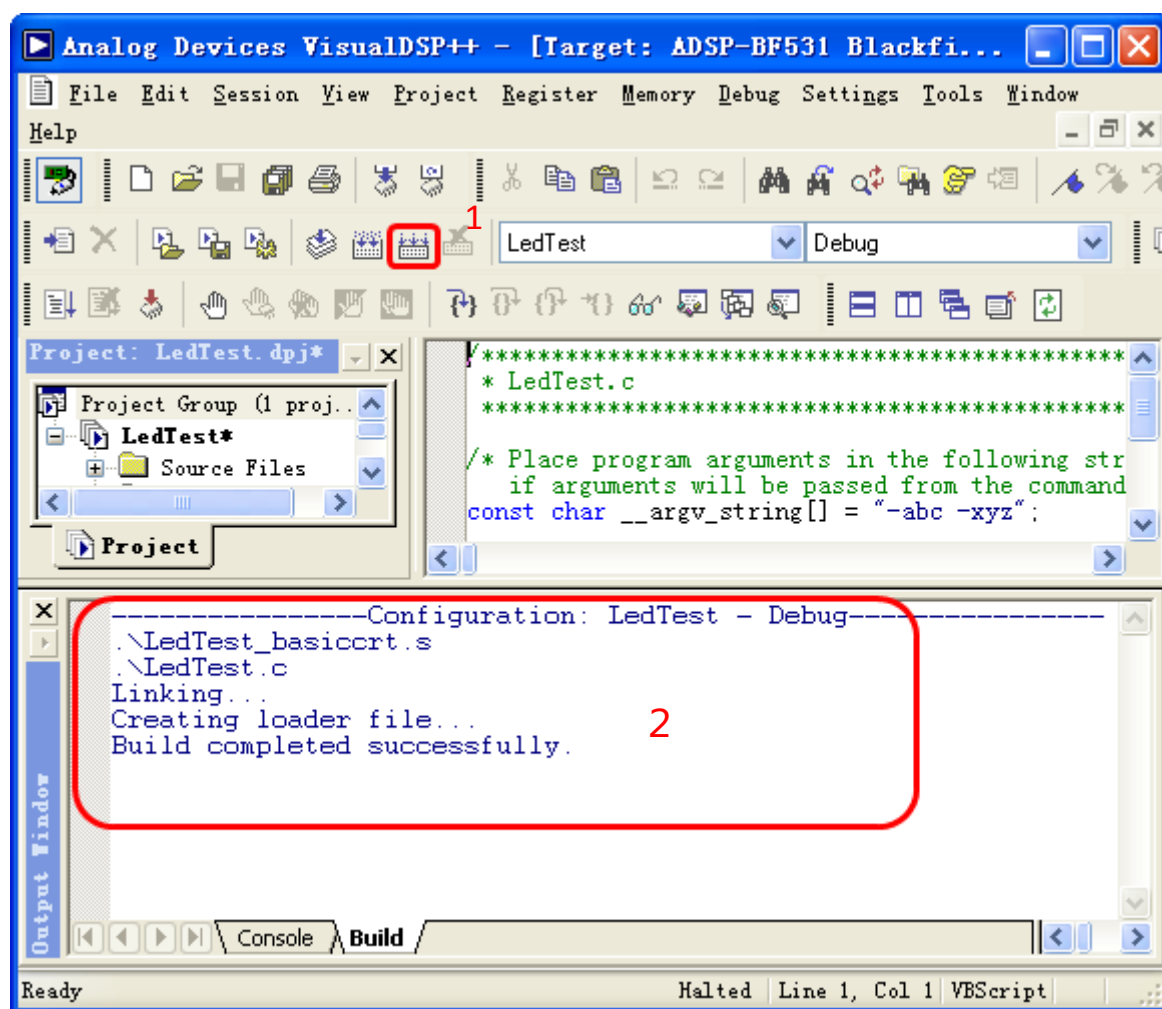


图 3-24 工程编译图

接下来，我们将给定的 DSP 初始化文件及下载文件拷贝到工程所在目录下，如图 3-25 所示，红圈 1 表示工程所在目录，红圈 2 是我们将要拷过来的 DSP 初始化文件及下载文件。关于 DSP 初始化文件，属于高级内容，我们在以后的章节里会介绍。

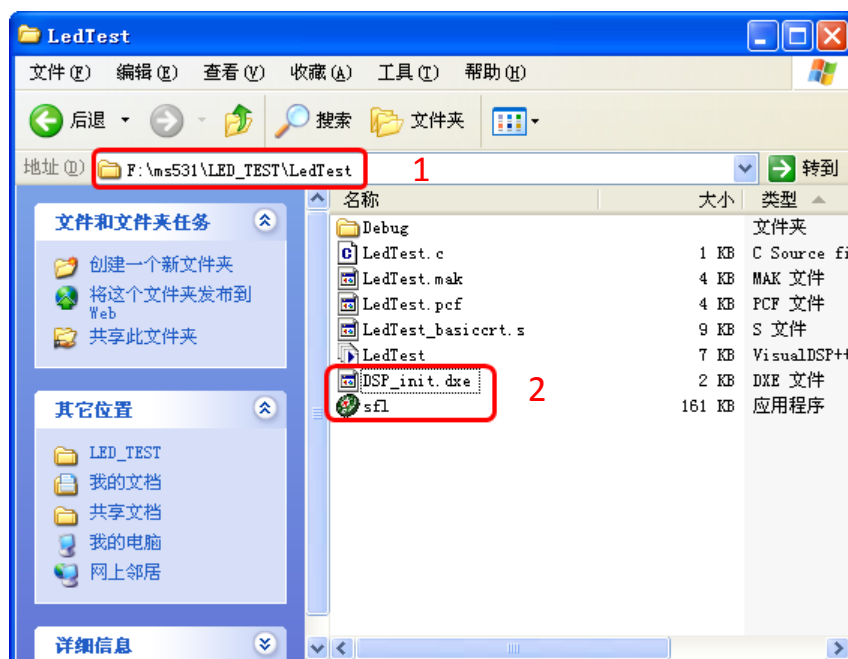


图 3-25 拷贝 DSP 初始化文件及下载文件

在新的工程建立完成后,用户可以通过工程选项窗口对工程修改,并对工程设置参数。用户通过单击 visualDSP++ 主界面中的“Project”下拉菜单,然后选择“Project Options”或者通过键盘快捷方式“Alt+F7”如图 3-26 所示。

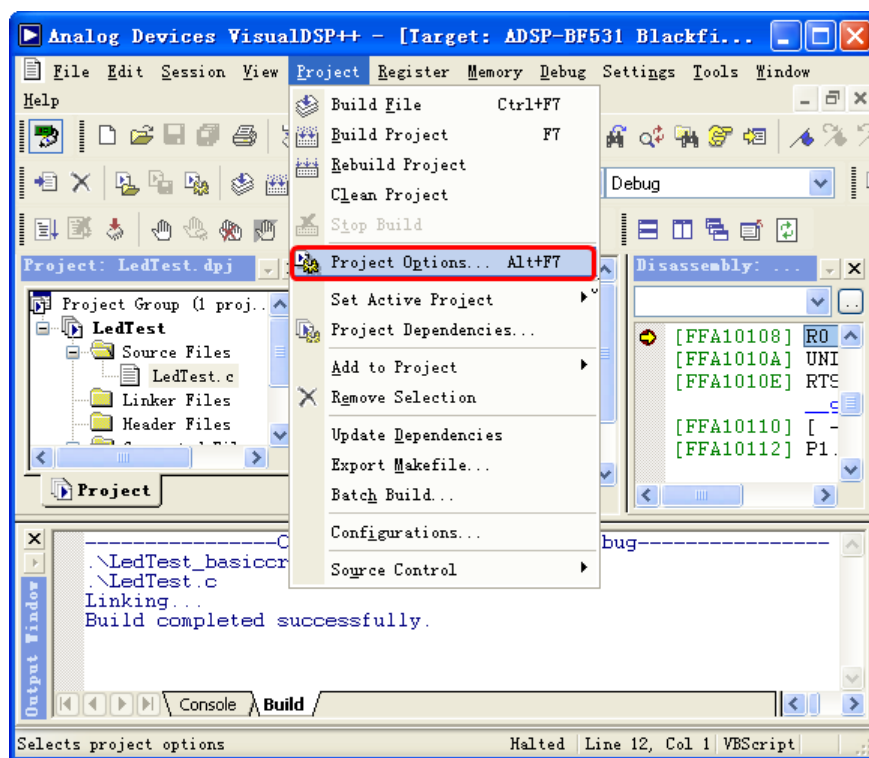


图 3-26 工程配置图

然后将弹出工程选项窗口，如图 3-27 所示，该窗口中主要包含 9 个部分，分别是：工程(Project)、常规(General)、编译(Compile)、汇编(Assemble)、链接(Link)、分割(Split)、加载(Load)、预编译(Pre-build)和后编译(Post-Build)等。

大部分选项栏的选项可以使用默认值，主要是 Project 选项栏的设置。它用于选择处理器类型和工程输出类型，其余选项可以使用默认值。Project 选项栏的各选项的意义如下。

目标(Target)

- Processor：用于设置该工程中使用的处理器的类型，包含了 ADI 公司的绝大多数处理器(如 ADSP-21160、ADSP-2106i、ADSP-21062、ADSP-21065L 等)。
- Type IDDE 的输出文件类型，包括可执行文件(Executable File)、库文件(Library File)、加载文件(Load File)、目标文件(Object File)和分割文件(Split File)等。
- Name：输出文件的文件名，如 LedTest。

工具链组(Tool Chain)

- Compiler：指定 C 编译器；
- Assembler：指定汇编器；
- Linker：指定链接器；
- Loader：指定加载器；
- Splitter：指定加载方式的镜像文件管理器。
- 该组参数基本上使用默认值即可。

设置(Settings for)

指定一个输出类型，有以下两种输出类型：

- Debug 类型，编译链接的工程文件可用来进行 Debug 调试。
- Release 类型，生成具有限制的或不能进行 Debug 调试的工程文件，Release 类型通常用来进行优化程序性能。

一般在调试过程中选择 Debug 类型，当程序调试好以后，选择 Release 类型。在此配置窗口，将红圈处的类型选为 Loader file，其它均不做修改。

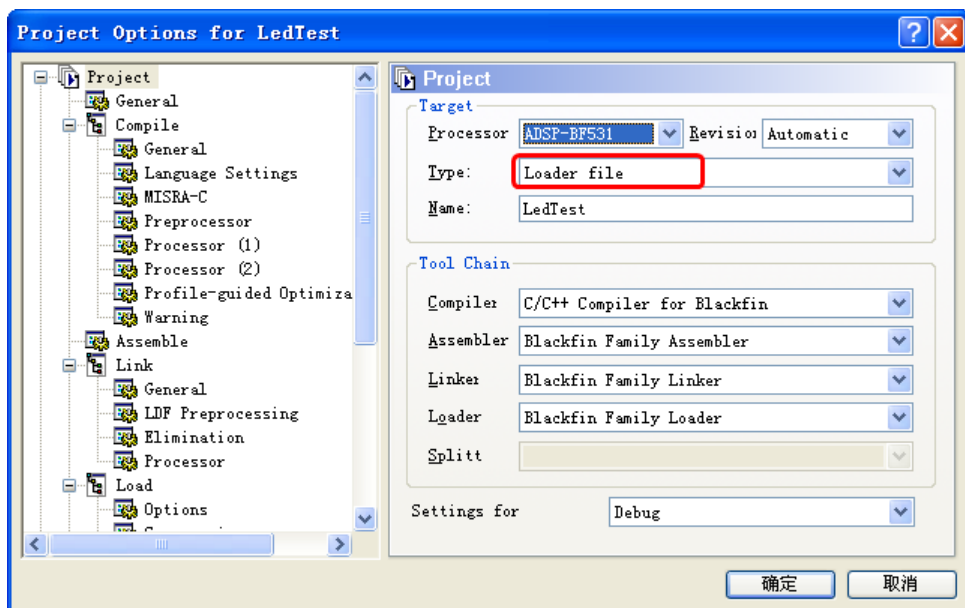


图 3-27 配置窗口

如图 3-28 所示，找到红圈 1 中 Load 下面的 Options 页面，红圈 2 中的选择 SPI，红圈 3 中选择 Binary，在红圈 4 处，找到刚才我们拷入的初始化文件 DSP_init.dxe，该文件就是对 DSP 的一些配置，初学者先不要追究里面的内容，直接使用我们提供的这些文件即可。

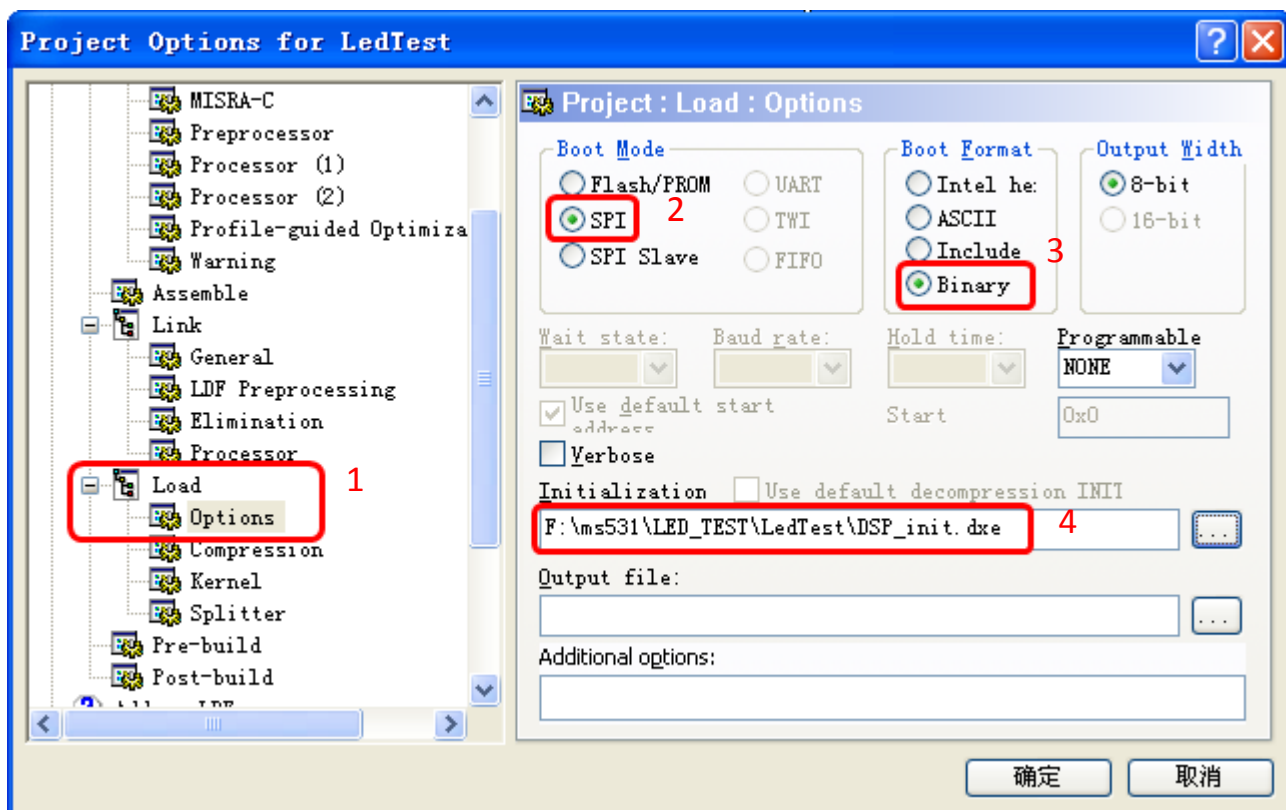


图 3-28 Load Options 的配置页面

其它页面不做修改，如图 3-29 所示，直接找到红圈 1 中的 Post-build 页面，然后在红圈 2 处输入 `sfl --download .\debug\ledtest.ldr`，其中 `ledtest` 是和工程对应的名称，如果工程名不是 `ledtest` 的话，请根据自己的工程做相应的修改。此设置的目的是为编译软件后，自动下载代码到 MS531 板子，添加此功能后，请确保每次编译工程前，您的 MS531 板子已经连接到了计算机，不然会出一点点小错误。

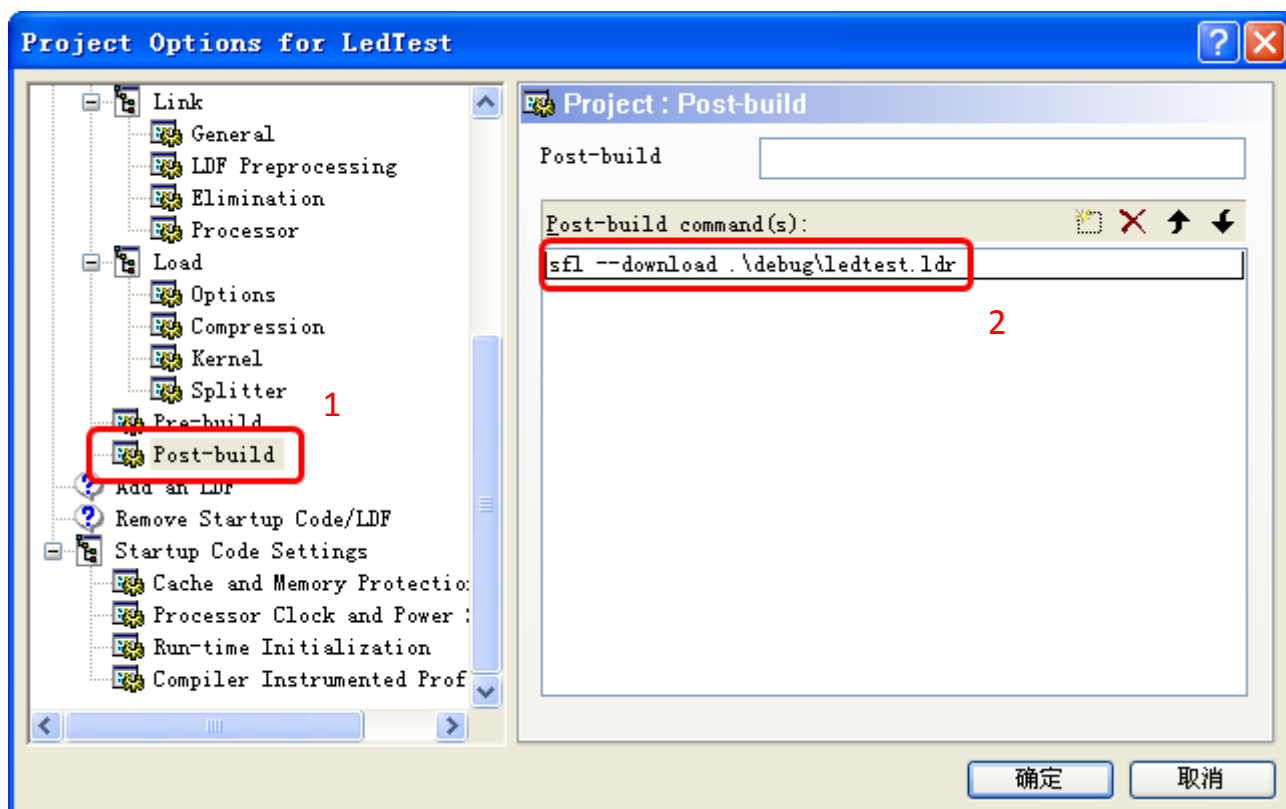


图 3-29 Load Post-build 的配置页面

直接点击“确定”，整个工程配置完成。

下面开始编译工程，将下面 led 灯闪烁的程序直接拷到代码区，如图 3-30 所示，覆盖原来代码区中的程序。

```
#include <cdefbf531.h>
#include <sysreg.h>
#include <math.h>
//-----define-----//
#define LED 11

#define initialize_led()  *pFIO_DIR|=1<<LED;

#define LED_ON      *pFIO_FLAG_S=1<<LED
#define LED_OFF     *pFIO_FLAG_C=1<<LED
#define LED_TOGGLE  *pFIO_FLAG_T=1<<LED

int main( int argc, char *argv[] )
{
    int i;

    initialize_led();
    while(1){
        LED_TOGGLE;
        for(i=0;i<1000000;i++);
    }
}
```

图 3-30 LED 灯闪烁代码

直接再次进行编译，编译结果如图 3-31 所示，编译结束后，会自动调用 SFL 进行目标程序下载，千万别忘了连上 ms531 板子到计算机上。这个时候实验板上的 led 灯开始闪烁。

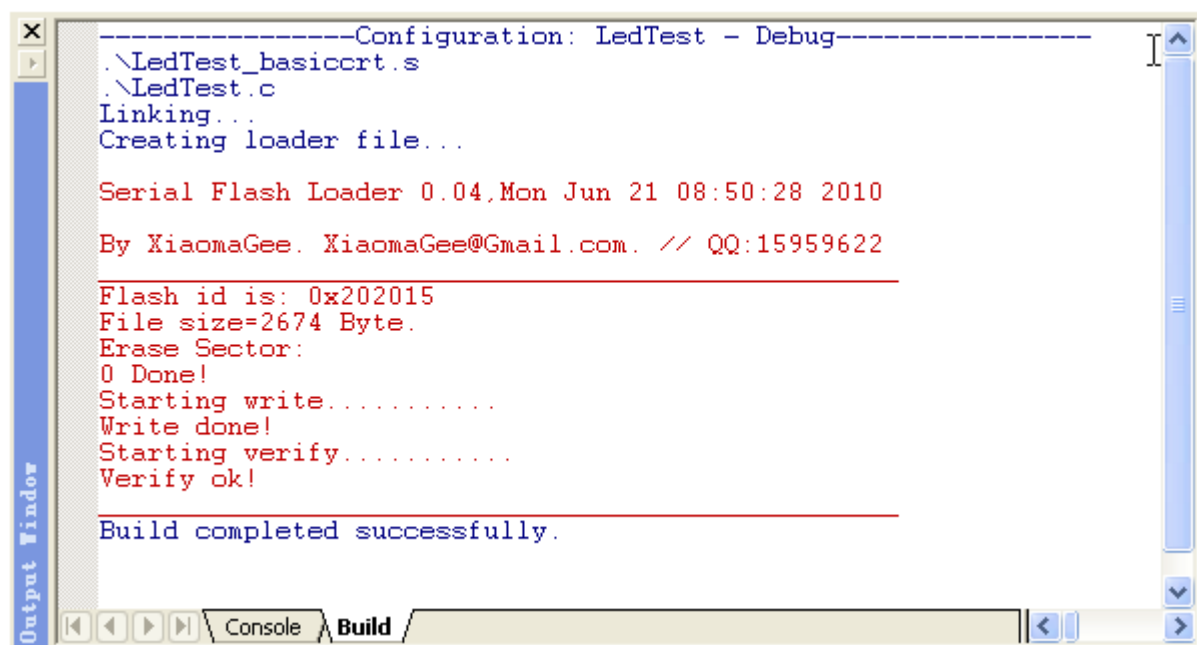


图 3-31 LED 灯闪烁编译结果图

到此为止，整个 LED 实验的工程建立完毕。