

ADS集成开发环境及 EasyJTAG-H仿真器的使用

ADS 1.2集成开发环境

- 概述:

ADS集成开发环境是ARM公司推出的ARM核微控制器集成开发工具，英文全称为ARM Developer Suite，成熟版本为ADS1.2。

- 特点:

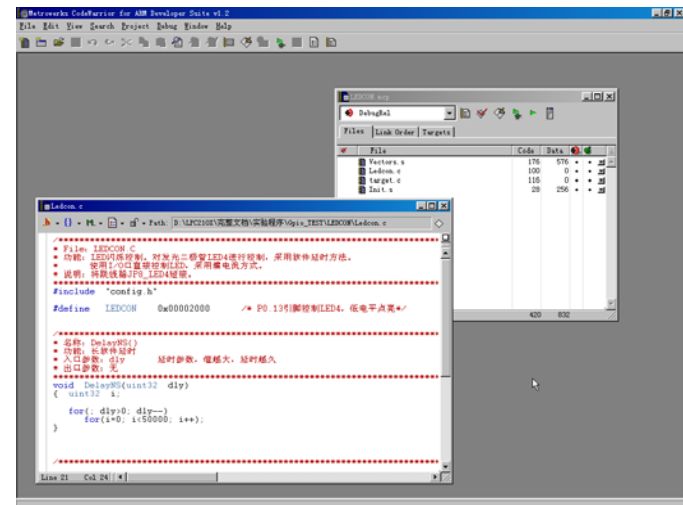
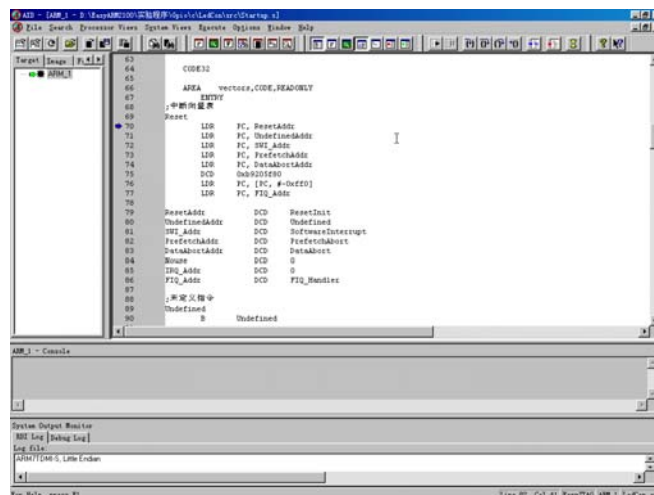
1. 支持软件调试及JTAG硬件仿真调试;
2. 支持ARM10之前的所有ARM系列微控制器;
3. 支持汇编、C、C++源程序，编译效率高，系统库功能强;
4. 可以在Windows98、Windows XP、Windows2000以及RedHat Linux上运行;
5. 支持单步、全速和断点等调试功能。



ADS 1.2集成开发环境

- 集成开发环境主要组成部分
CodeWarrior IDE

AXD



ADS 1.2集成开发环境

- CodeWarrior IDE简介

ADS 1.2使用了CodeWarrior IDE集成开发环境，并集成了**ARM汇编器**、**ARM的C/C++编译器**、**Thumb的C/C++编译器**、**ARM连接器**，包含工程管理器、代码生成接口、语法敏感（对关键字以不同颜色显示）编辑器、源文件和类浏览器等。



ADS 1.2集成开发环境

- AXD调试器简介

AXD调试器为ARM扩展调试器（ARM eXtended Debugger），包括ADW/ADU的所有特性，支持**硬件仿真**和**软件仿真**(ARMulator)。AXD能够装载映像文件到目标内存，具有**单步**、**全速**和**断点**等调试功能，可以观察变量、寄存器和内存的数据等。



在开发板上运行第一个程序

- “从做中学”

目标:

对8个LED进行花样控制

步骤:

安装ADS 1.2软件

专用模板建立工程

使用EasyJTAG-H仿真器

AXD工程调试

脱机运行

程序固化程序



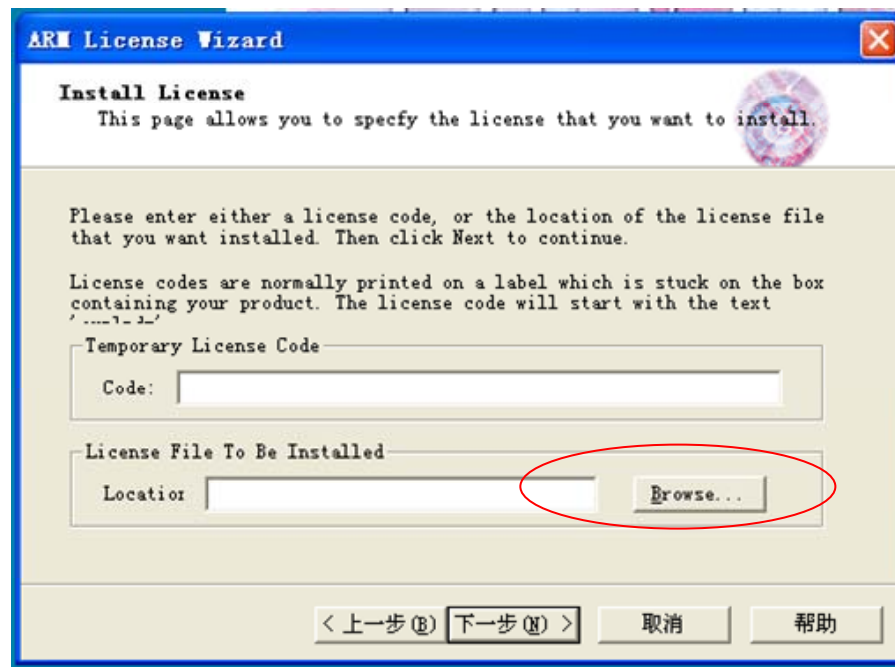
在开发板上运行第一个程序

- 安装ADS 1.2注意要点

安装完ADS1.2集成开发环境后，要继续安装许可证添加向导。在出现如图所示对话框时，注意选择选择Location项，正确的选择为

：

ADS1.2安装软件包中CRACK文件夹里的LICENSE文件。

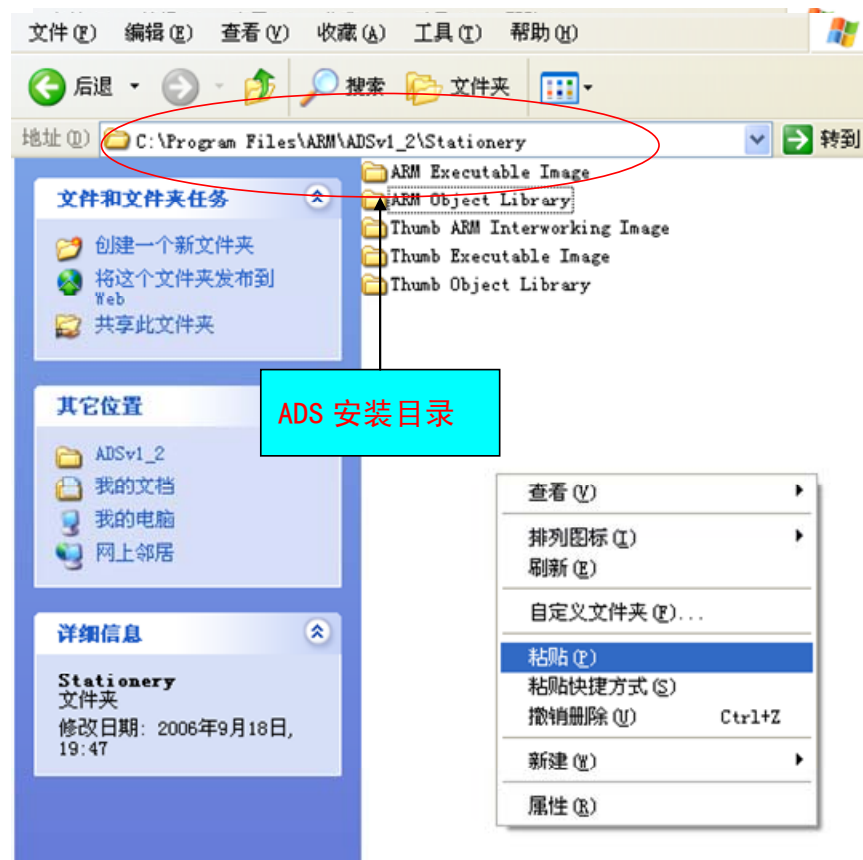


在开发板上运行第一个程序

- 工程建立具体操作步骤
 - ▶ 增加LPC2131专用工程模板; 🤪
 - ▶ 使用LPC2131专用工程模板建立工程; 🤪
 - ▶ 编写流水灯控制程序; 🤪
 - ▶ 编译连接工程。 🤪

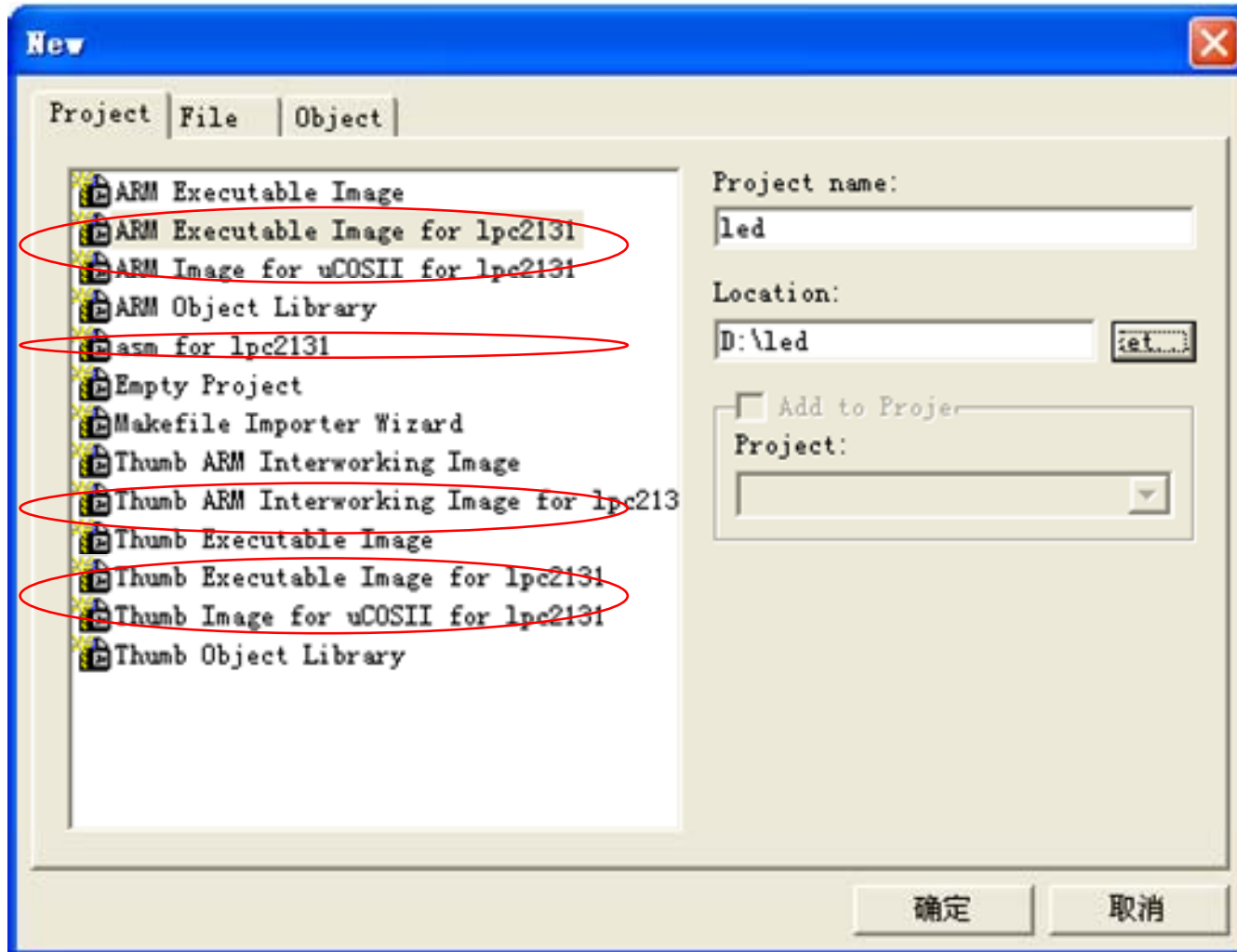
在开发板上运行第一个程序

将产品光盘中带有的LPC2131工程模板拷贝到“<ADS1.2安装目录>\Stationery\”下。



选择复制的文件到目标目录

在开发板上运行第一个程序



增加的工程模板

在开发板上运行第一个程序

• 8个LED花样控制源代码

```
#include "config.h"
#define LED8 0xff << 18
// P1.18~P1.25控制LED1~LED8
// 低电平有效（宏定义为了便于引用与修改）
```

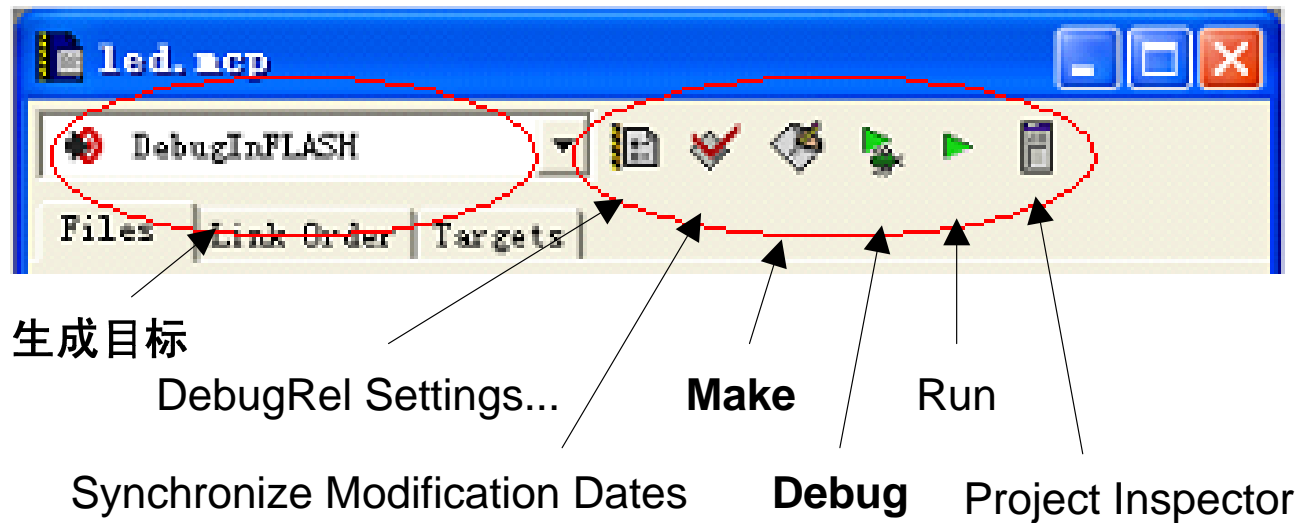
```
void DelayNS(uint32 dly)
// 建立纳秒级延时
{
    uint32 i;
    for( ; dly>0; dly--)
        for(i=0; i<50000; i++);
}
```

```
const uint32 LED_TBL[] = {
// 流水灯花样
0x01,0x02,0x04,0x08,
0x10,0x20,0x40,0x80,
};
```

```
int main (void)
{
    uint8 i;
    PINSEL1 = PINSEL1 & (~0x80);
    // 选择P1.16~P1.25为GPIO功能
    IO1DIR = LED8;
    // 设置LED控制口为输出
    while(1)
    {
        for(i=0; i<8; i++)
        {
            IO1SET = ~((LED_TBL[i]) << 18);
            DelayNS (10);
            IO1CLR = ((LED_TBL[i]) << 18);
            DelayNS (10);
        }
    }
    return 0;
}
```

在开发板上运行第一个程序

- 工程窗口



DebugRel Settings...

Synchronize Modification Dates

Make

Debug

Run

Project Inspector

工程设置

同步修改日期

编译连接

启动AXD进行调试

启动AXD进行调试，并直接运行程序

工程检查

在开发板上运行第一个程序

- EasyJTAG-H仿真器的使用步骤


- ▶ 安装H-JTAG软件

H-JTAG Server 用于检测芯片内核

H-Flasher 用于完成Flash的烧写

- ▶ 设置H-JTAG软件

在H-JTAG Server窗口中点击图标，检测芯片内核；Flasher项选中 Auto Download

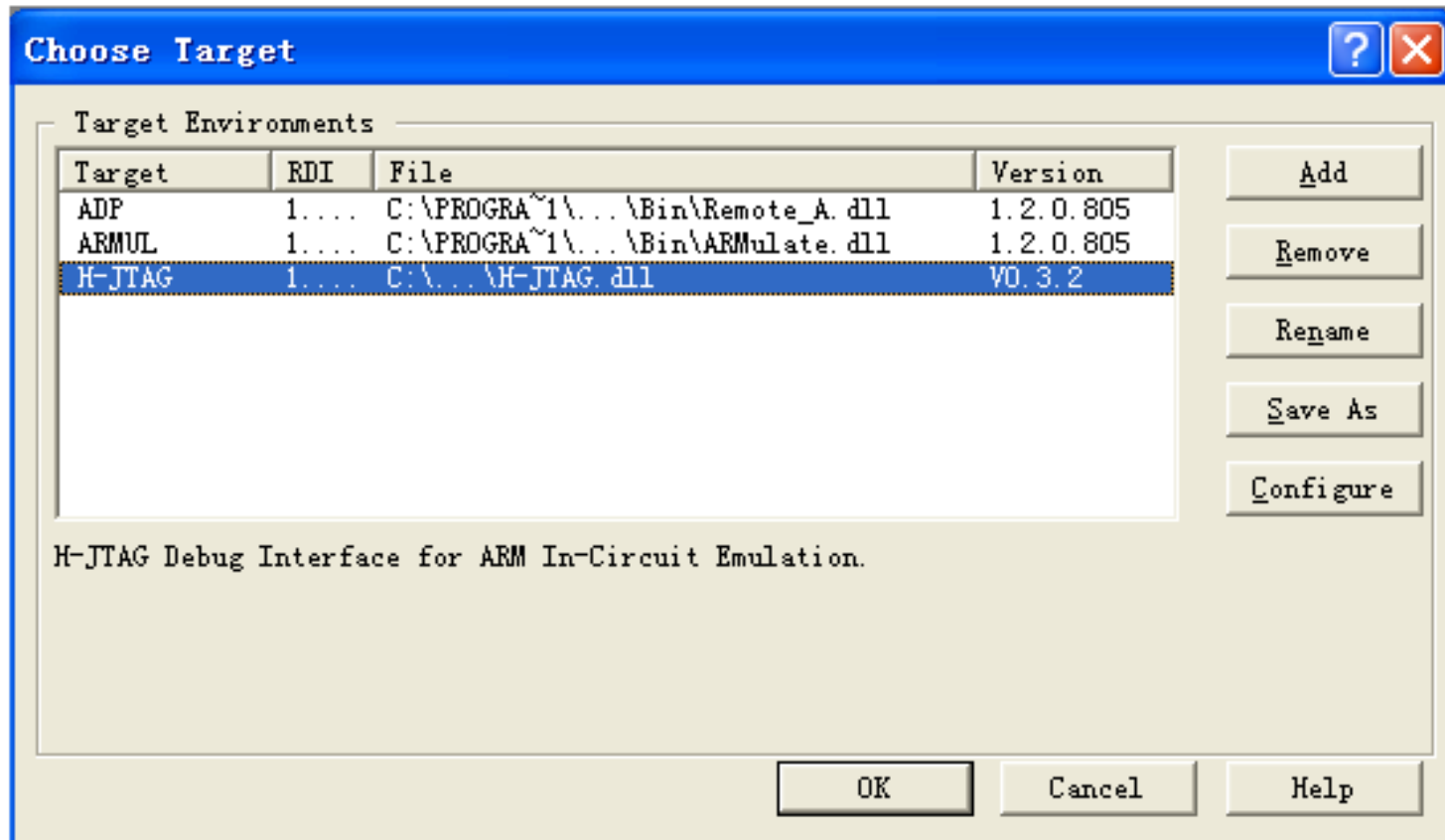
在H-Flasher窗口 Flash Selection 项选择CPU型号并保存设置。

在开发板上运行第一个程序

- **AXD工程调试步骤**
 - ▶ 启动**AXD**调试软件
 - ▶ 选择调试目标
 - ▶ 在线调试

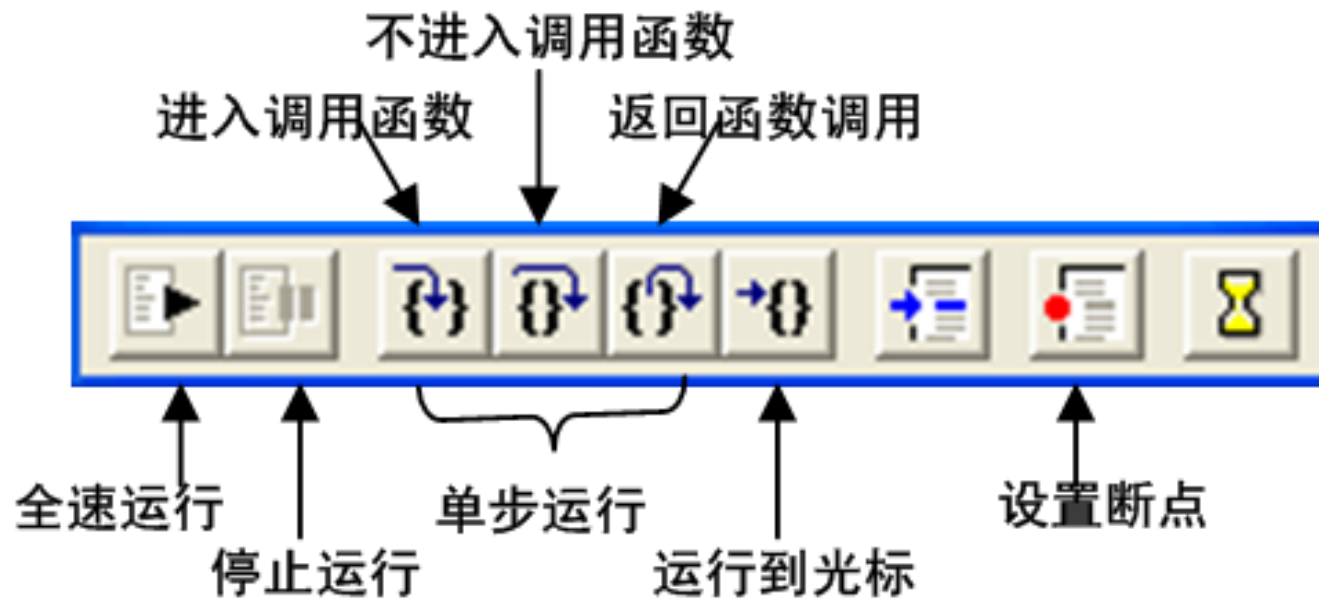
在开发板上运行第一个程序

- 选择添加H-JTAG驱动



在开发板上运行第一个程序

- 调试工具条



运行调试工具条

在开发板上运行第一个程序

- 调试工具条



调试观察窗口工具条

在开发板上运行第一个程序

- 调试工具条



文件操作工具条

在开发板上运行第一个程序

- 脱机运行程序
 - 将短接的ISP跳线取走
 - 开发板重新上电一次

在开发板上运行第一个程序

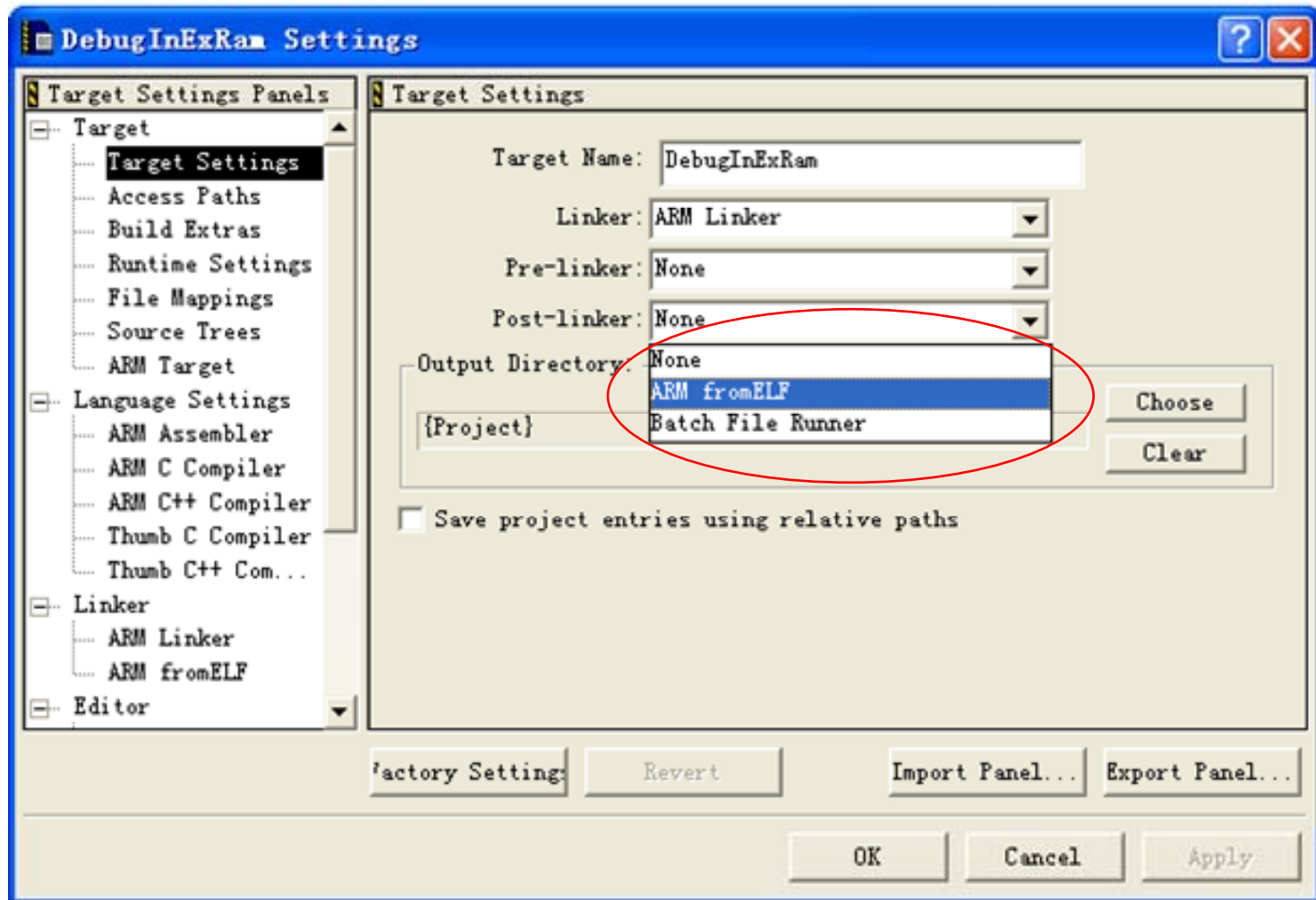
- 固化程序
 - 生成HEX文件
 - 使用JTAG接口下载
 - 使用ISP下载

在开发板上运行第一个程序

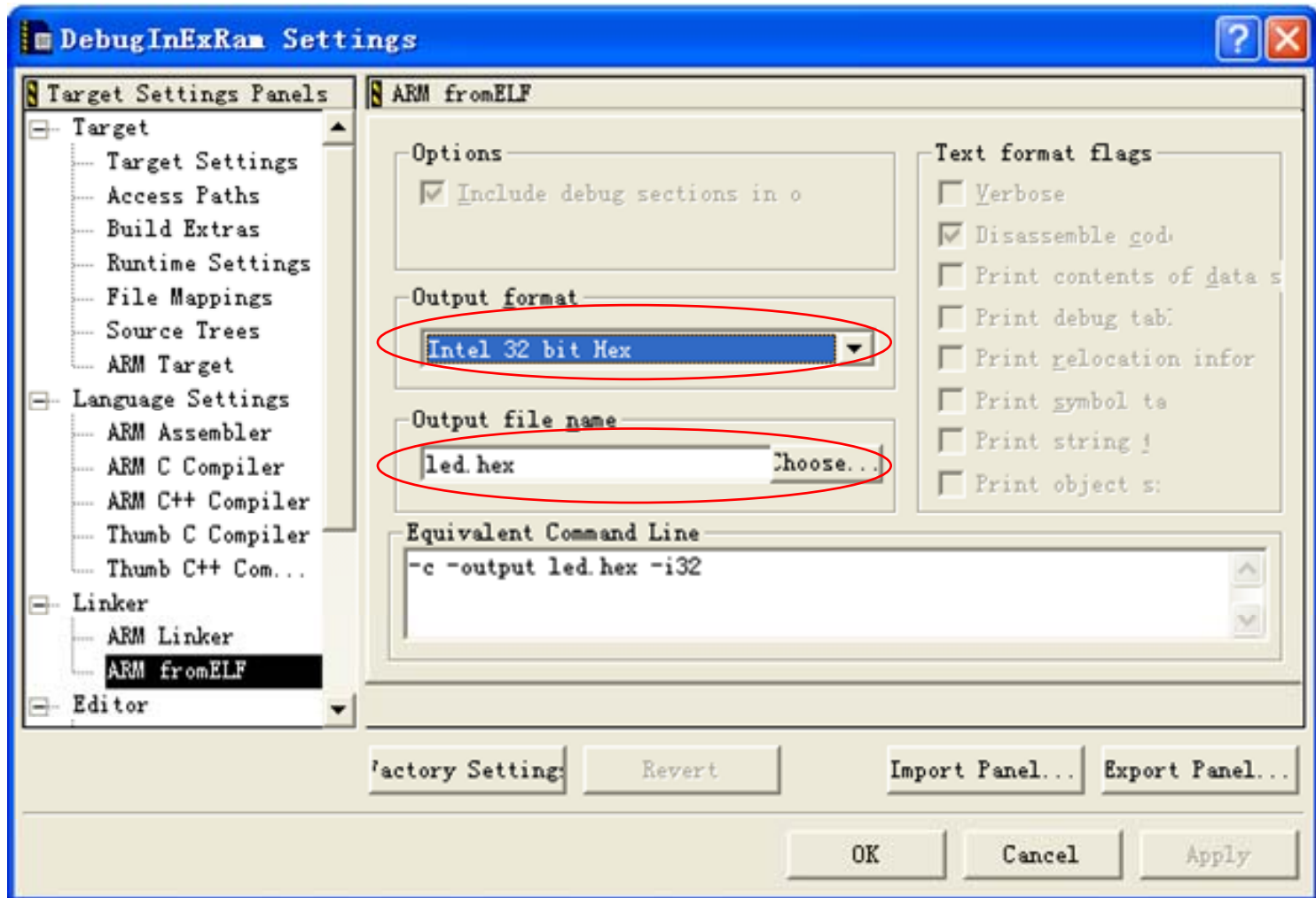
- **HEX文件的生成**

- ▶ 选择生成目标DebugInFLASH或RelInFLASH;
- ▶ 打开工程的DebugInFLASH Settings窗口;
- ▶ 在Target Settings项中设置Post-linker选取ARM fromELF; 🤪
- ▶ 在ARM formELF项中设置输出文件类型, 如设置为Intel 32 bit Hex, 然后设置输出文件名, 也可指定目录, 若不指定目录, 则生成文件存放在当前工程的目录中; 🤪
- ▶ 重新编译连接, 编译通过即会生成指定的输出文件。

在开发板上运行第一个程序









在开发板上运行第一个程序

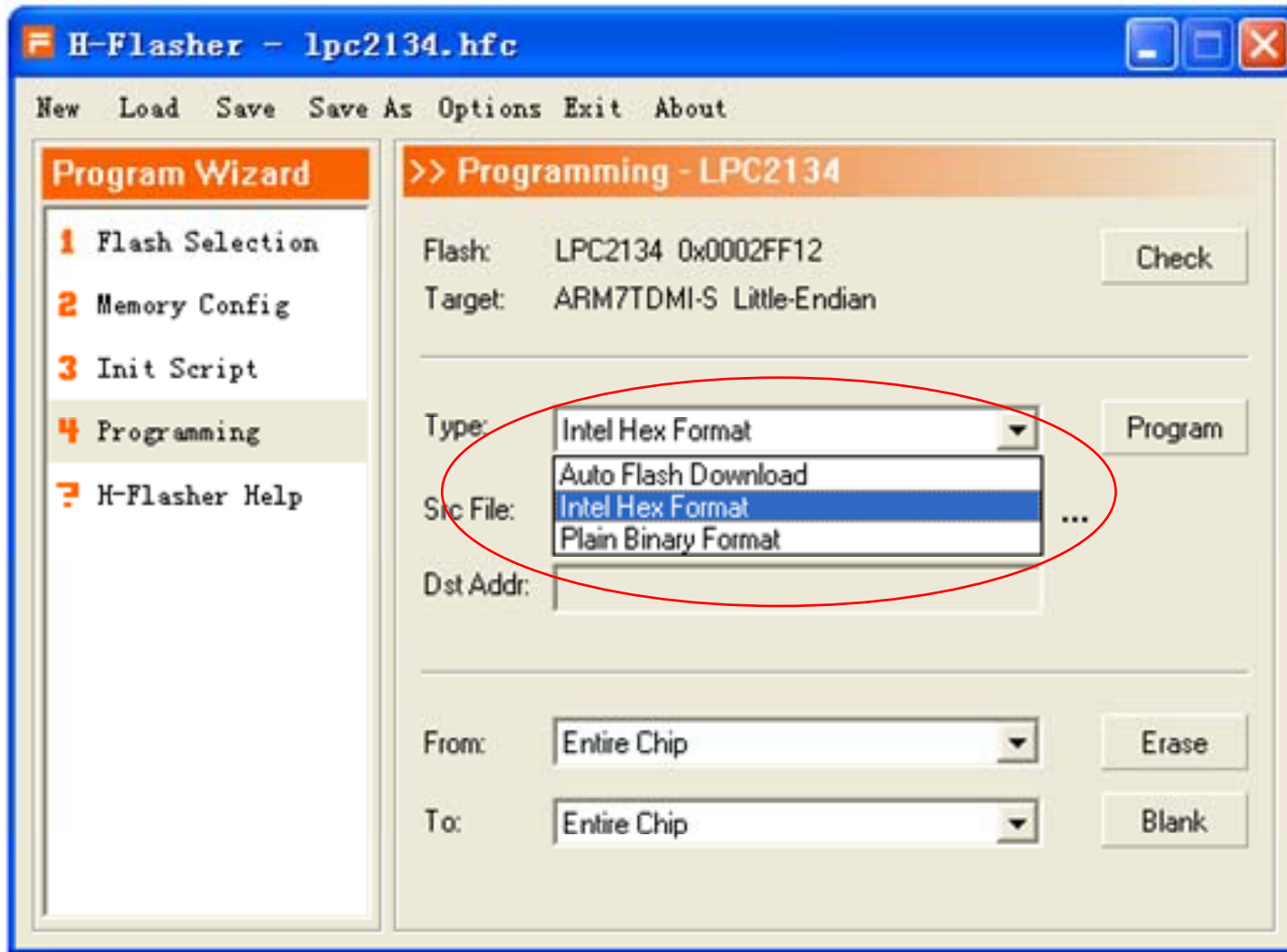


在开发板上运行第一个程序

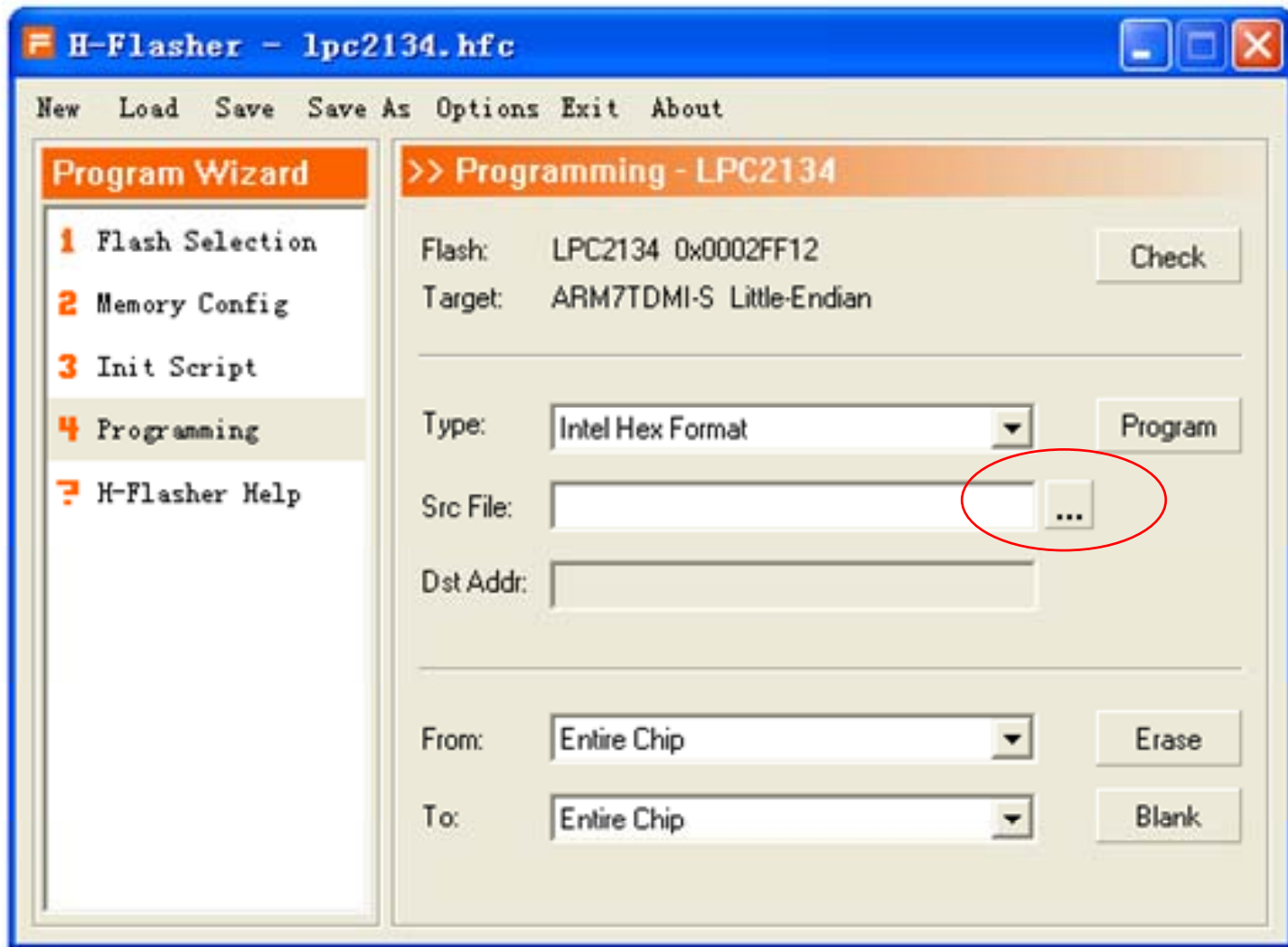
- 使用JTAG下载

- 连接开发板与PC串口，并将ISP跳线短接；
- 打开H-Flasher软件，选中  Programming 项，在Type栏选择下载文件类型为Inter Hex Format; 
- Src File栏选择文件路径; 
- 点击Erase按钮全片擦除; 
- 点击Black按钮擦除校验; 
- 点击Program按钮下载HEX文件。 

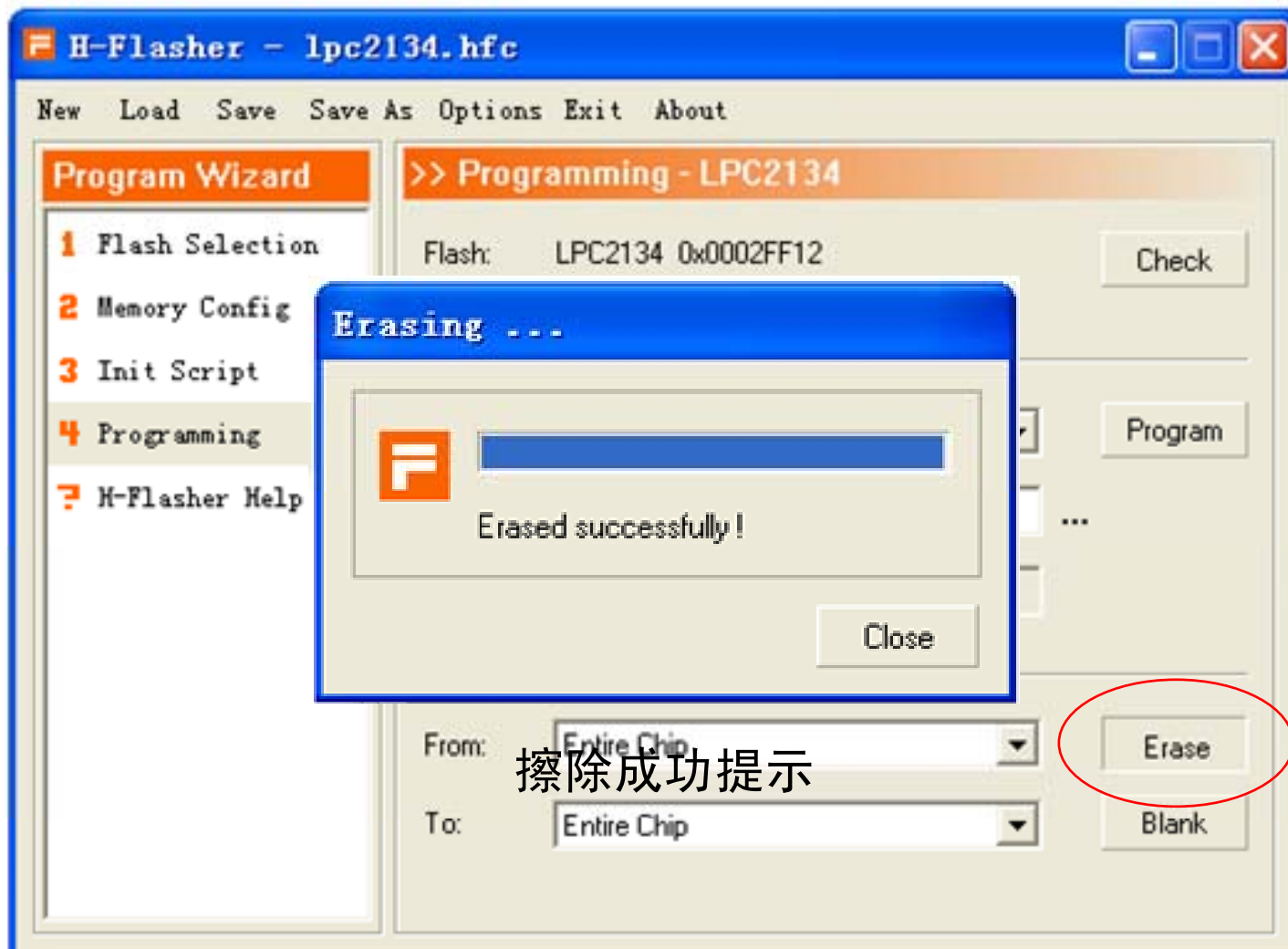
在开发板上运行第一个程序



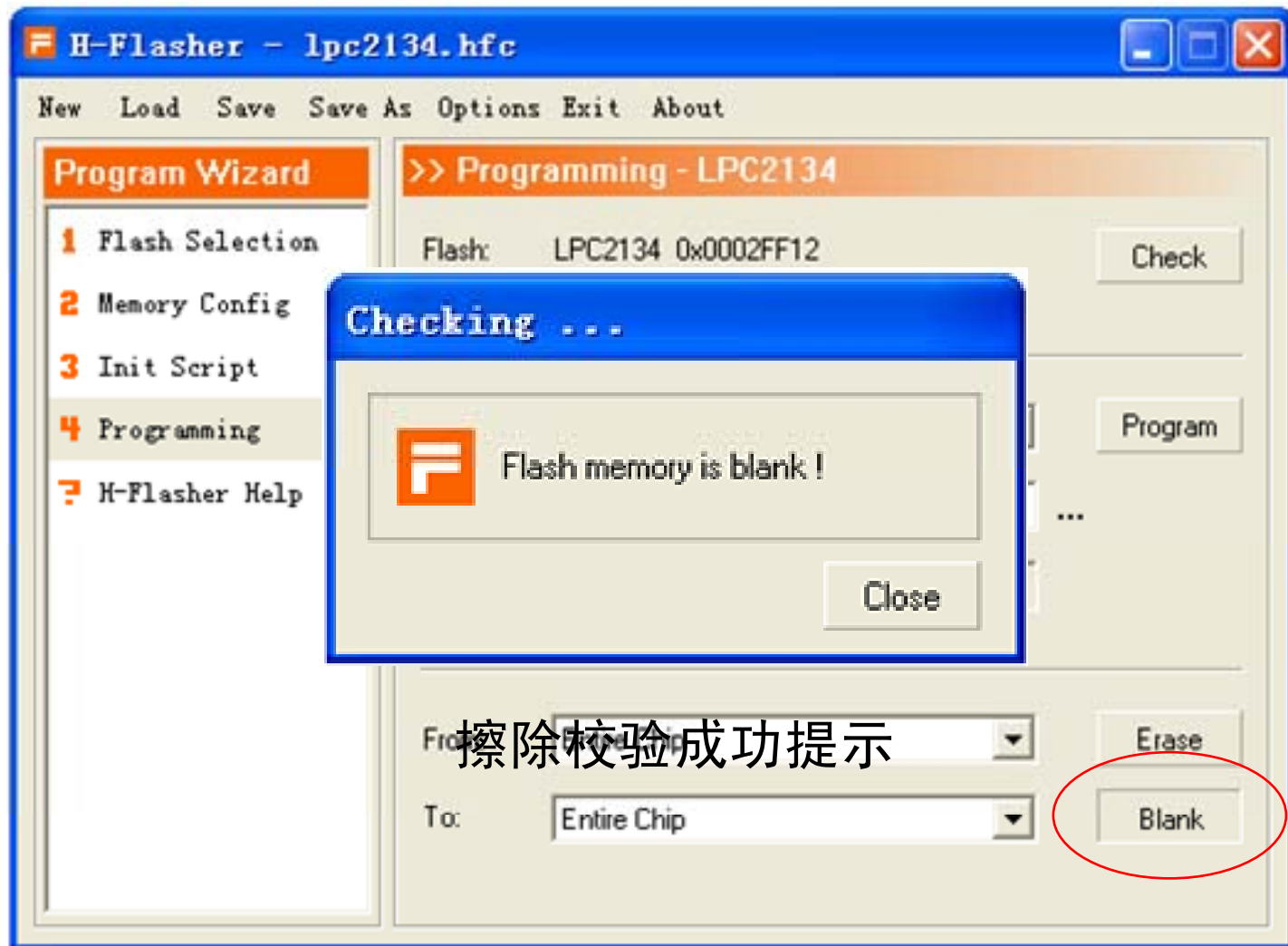
在开发板上运行第一个程序



在开发板上运行第一个程序



在开发板上运行第一个程序



擦除校验成功提示

在开发板上运行第一个程序



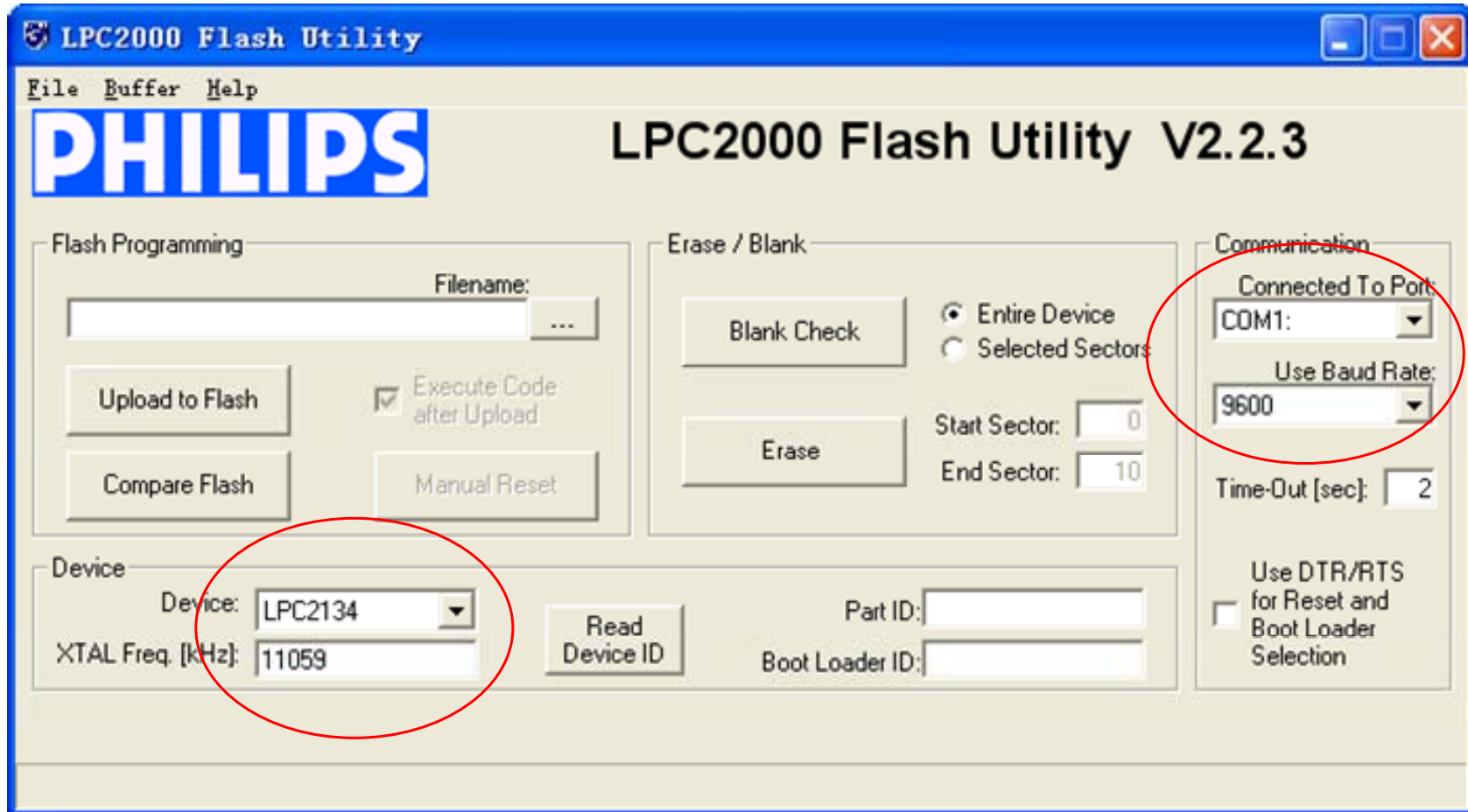
在开发板上运行第一个程序

- 使用ISP下载

- 安装LPC2000 Flash Utility软件;
- 连接开发板与PC串口, 并将ISP跳线短接;
- 打开LPC2000 Flash Utility软件, 设置参数; 🤪
- 点击Read Device ID按钮, 提示复位LPC2000信息; 🤪
- 再次点击Read Device ID按钮, 读取芯片ID号; 🤪
- 点击Erase按钮擦除FLASH; 🤪
- 点击Black Check按钮确认擦除成功; 🤪
- 在Filename项中选择要下载的HEX文件路径; 🤪
- 点击Upload to Flash按钮下载HEX文件。🤪



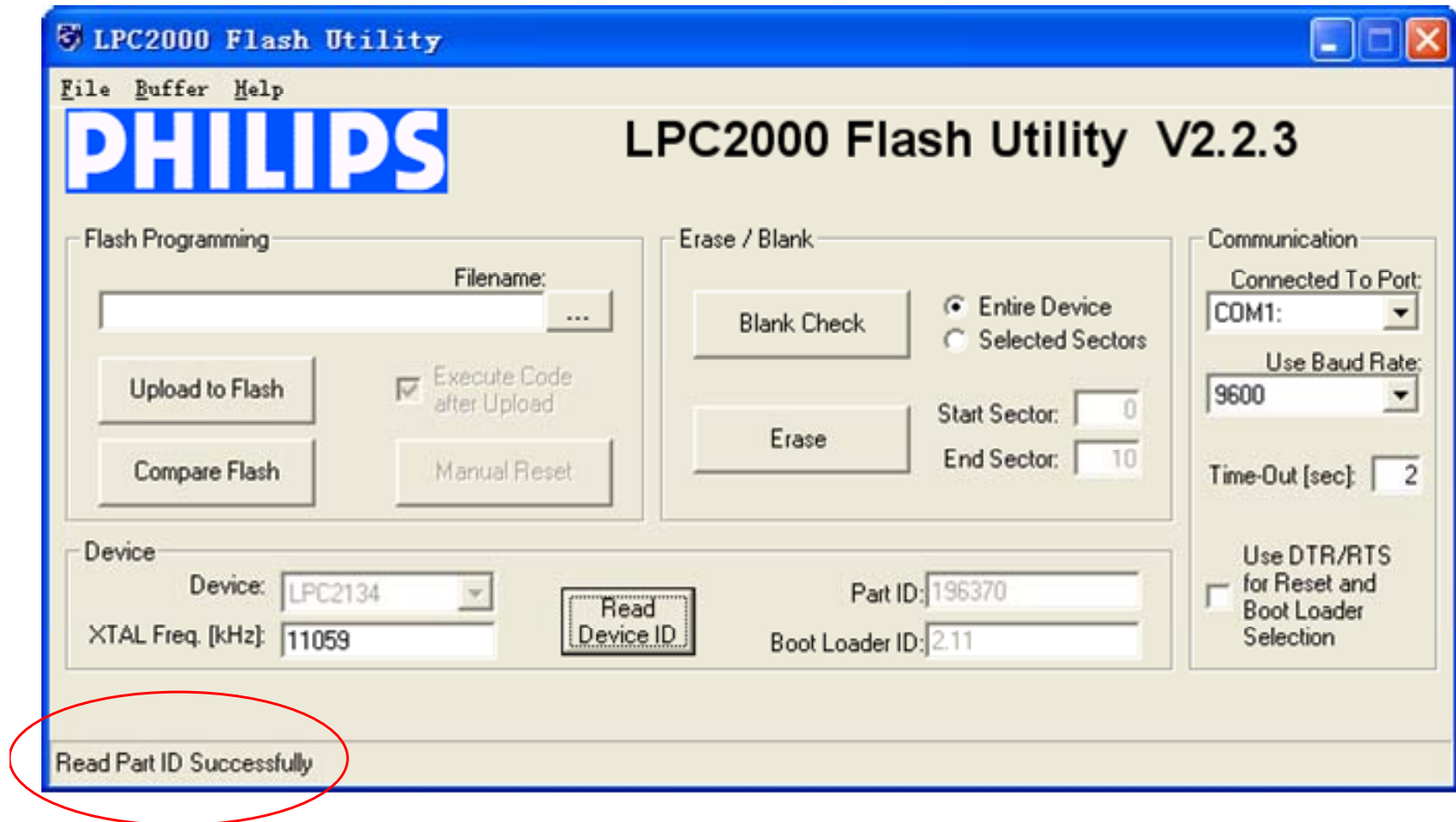
在开发板上运行第一个程序



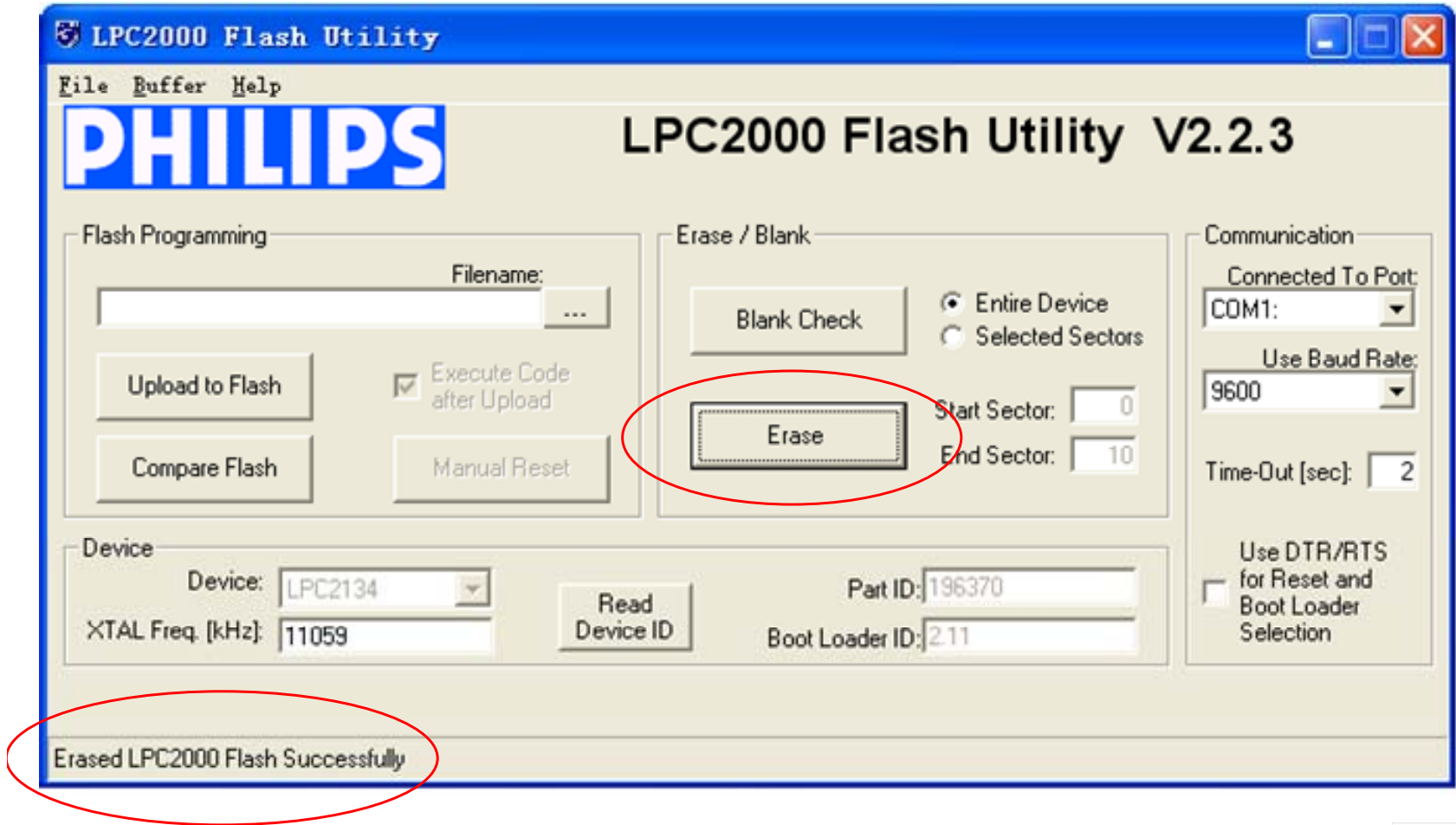
在开发板上运行第一个程序



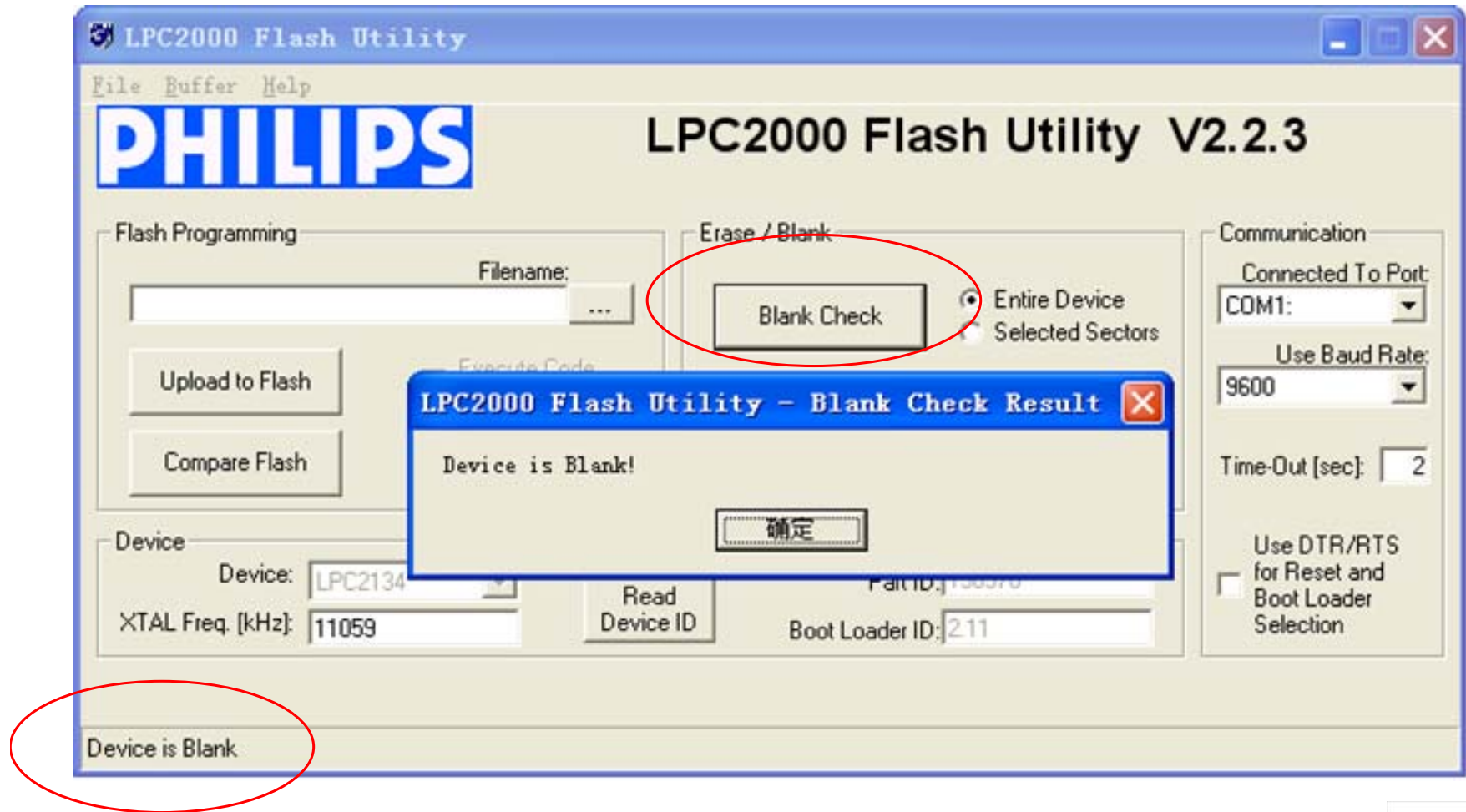
在开发板上运行第一个程序



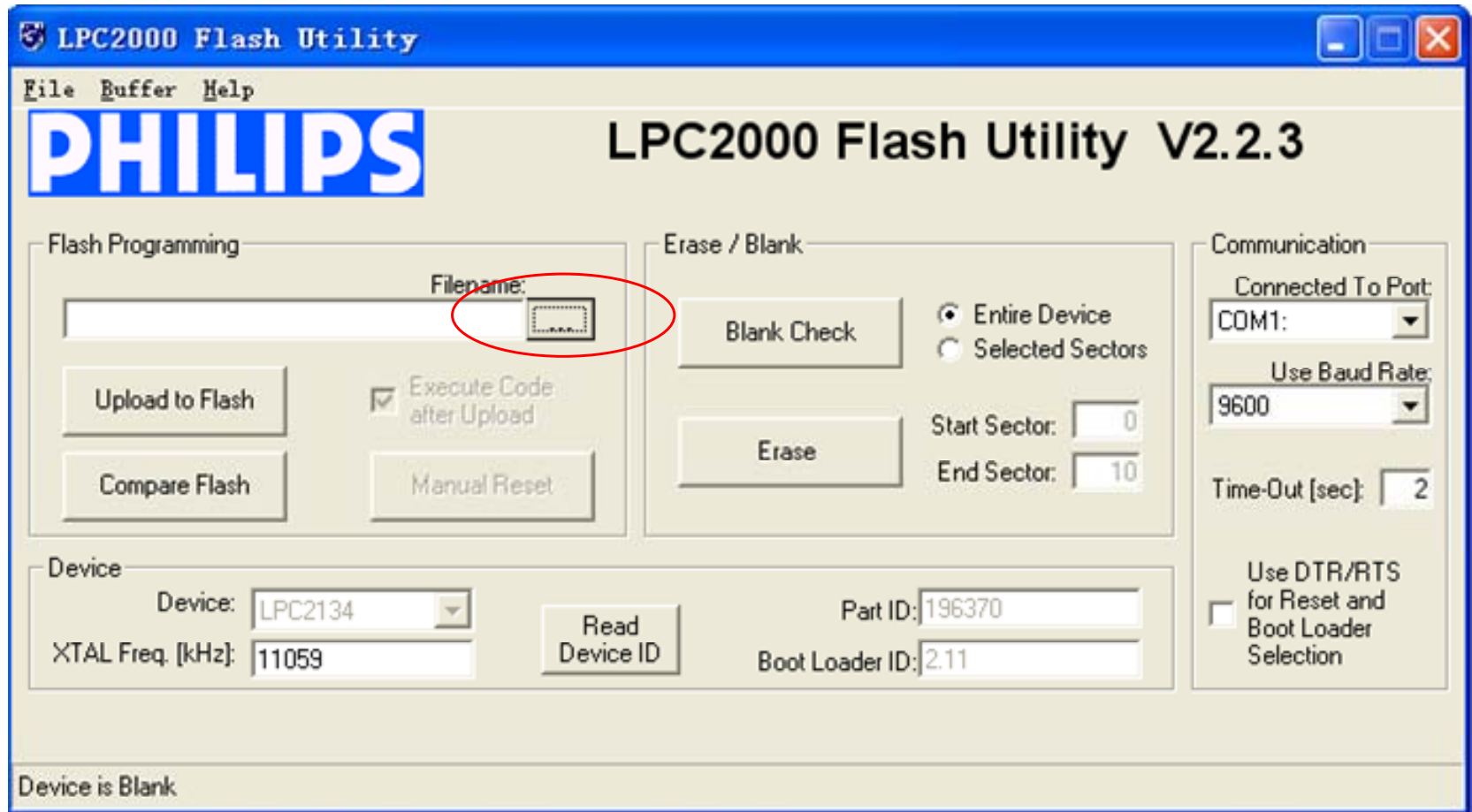
在开发板上运行第一个程序



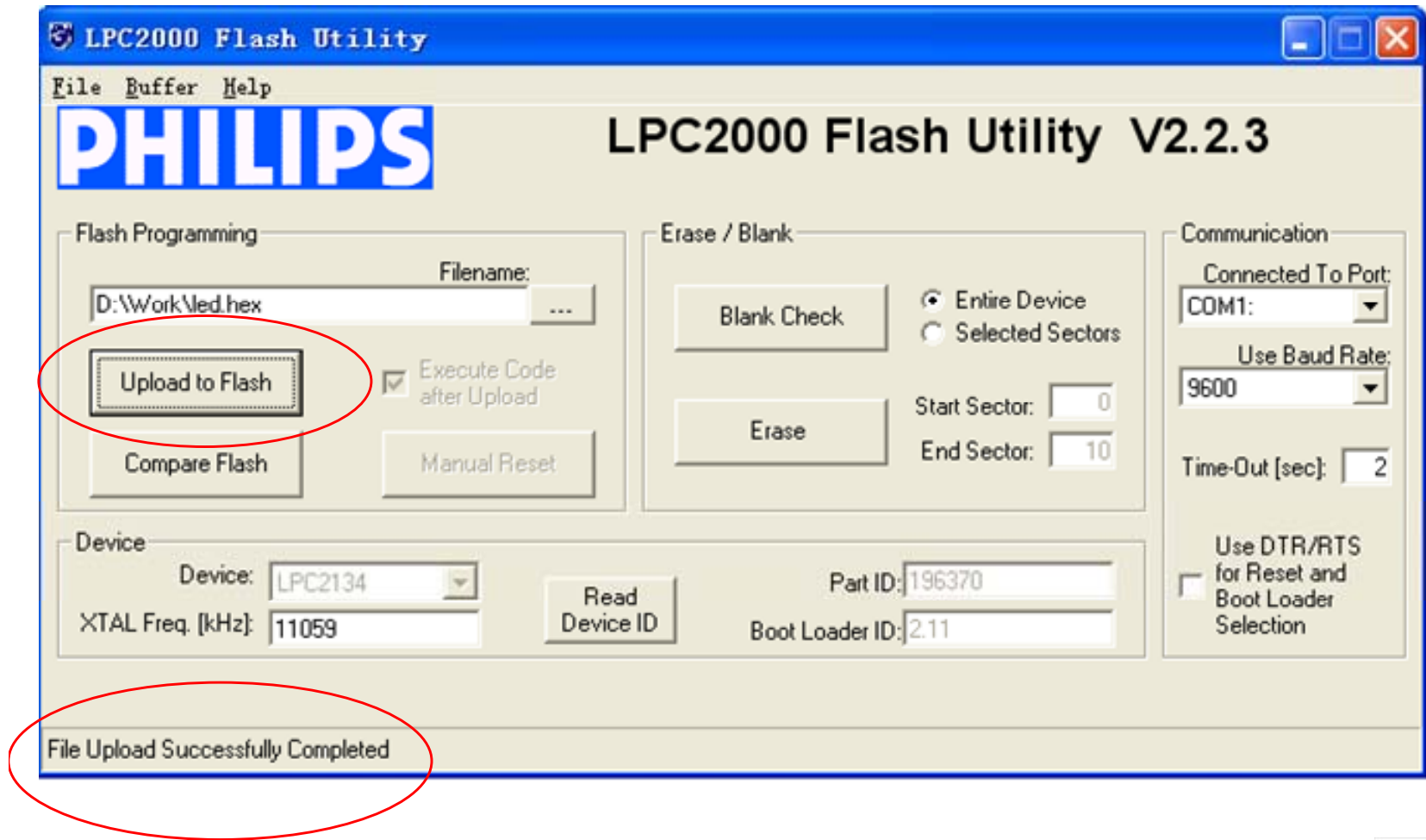
在开发板上运行第一个程序



在开发板上运行第一个程序



在开发板上运行第一个程序



专注的力量成就梦想！

谢谢大家！