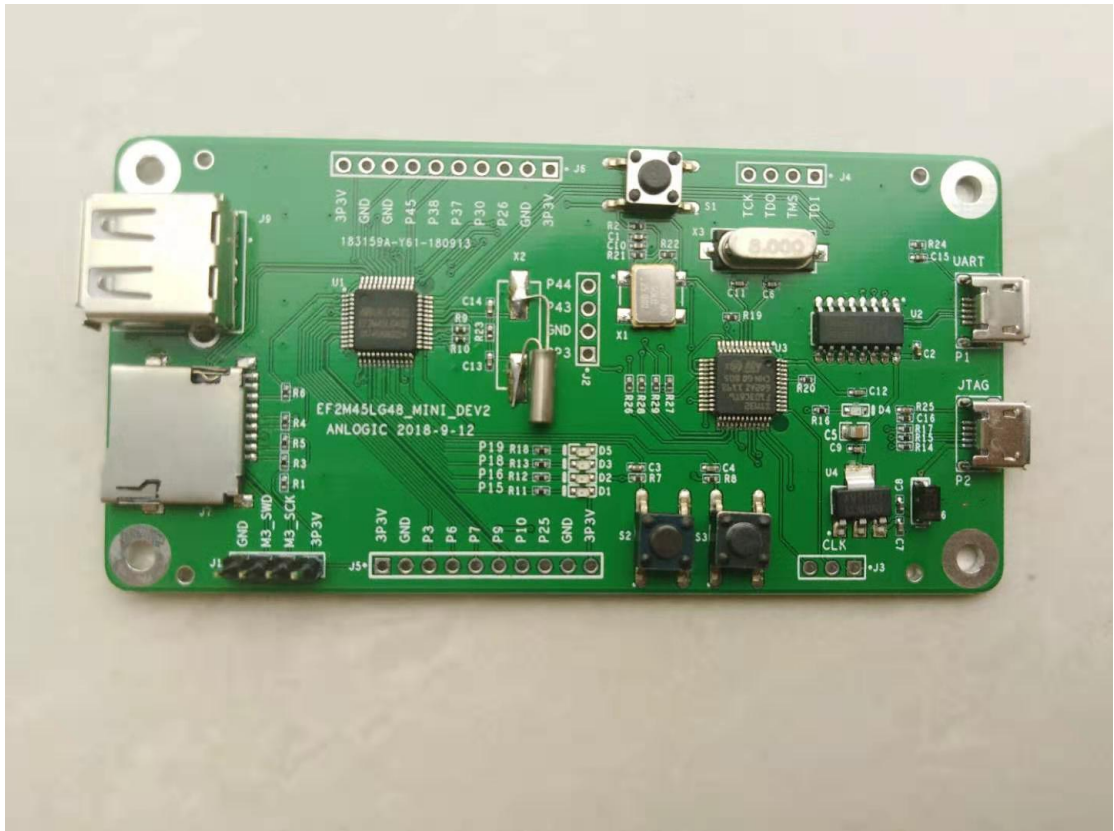


# 基于 EF2M45 运行 Arduino 移植测试

## 1.可测试的功能

目前已经移植的功能有 I/O、PWM，以及 UART 通信的部分功能。适用于安路的板子 EF2M45LG48\_MINI\_DEV2。



## 2.可用函数说明

### 2.1 I/O 函数

Arduino 函数库自带的 I/O 函数包括 `pinMode()` 函数、`digitalWrite()` 函数、`digitalRead()` 函数。

#### **`pinMode(pin, mode)`函数语法**

`pin`：引脚的编号，`pin` 可以使用数字 0~31，也可以使用 `gpio0_0~gpio0_15`、`gpio1_0~gpio1_15`，需要结合 FPGA 工程共同设置。

`mode`：INPUT 或 OUTPUT。

---

#### **digitalWrite(pin, value)函数语法**

pin：引脚的编号，pin 可以使用数字 0~31，也可以使用 gpio0\_0~gpio0\_15、gpio1\_0~gpio1\_15，需要结合 FPGA 工程共同设置。

value：HIGH 或 LOW。

#### **digitalRead(pin)函数语法**

pin：引脚的编号，pin 可以使用数字 0~31，也可以使用 gpio0\_0~gpio0\_15、gpio1\_0~gpio1\_15，需要结合 FPGA 工程共同设置。

## **2.2 脉冲宽度调制 PWM**

关于 PWM 基本原理这里不过多讲述。Arduino 内自带 analogWrite()函数，将模拟值（PWM 波）写入引脚。它可用于以不同的亮度点亮 LED 或以各种速度驱动电机。本人自定义 PWM 初始化 pwmInit()函数。

#### **analogWrite (pin, value)函数语法**

pin：引脚的编号，pin 可以使用数字 0~9。

mode：占空比 0~255。

#### **pwmInit()函数**

可将 0~9 号 PWM 输出占空比初始化为 0。

## **2.3 UART 通信**

Arduino 自带 Serial 对象，以及常用函数 begin()函数和 println()函数。

#### **Serial.begin(baud)函数语法**

baud：波特率

#### **Serial.println(res)函数语法**

res：字符串

目前还不支持 Serial.read()函数。

## **3.测试方法**

Arduino IDE 和 Anlogic TD 工具的使用方法，在安路的官方资料中有详细的说明，可参考：

---

<https://github.com/AnlogicInfo/EF2M45-SOC>

<https://github.com/AnlogicInfo/EF2M45duino>

这里不再过多讲述, 只讲述本例的测试方法。安装 Arduino IDE 和 Maple 库, 将 MapleLibrary 中的文件复制到 Arduino IDE 安装目录中 Arduino\hardware\Arduino\_STM32 位置, 覆盖原有文件。

 drivers	2020/4/8 16:05	文件夹
 STM32F1	2020/4/8 16:05	文件夹
 tools	2020/4/8 16:05	文件夹

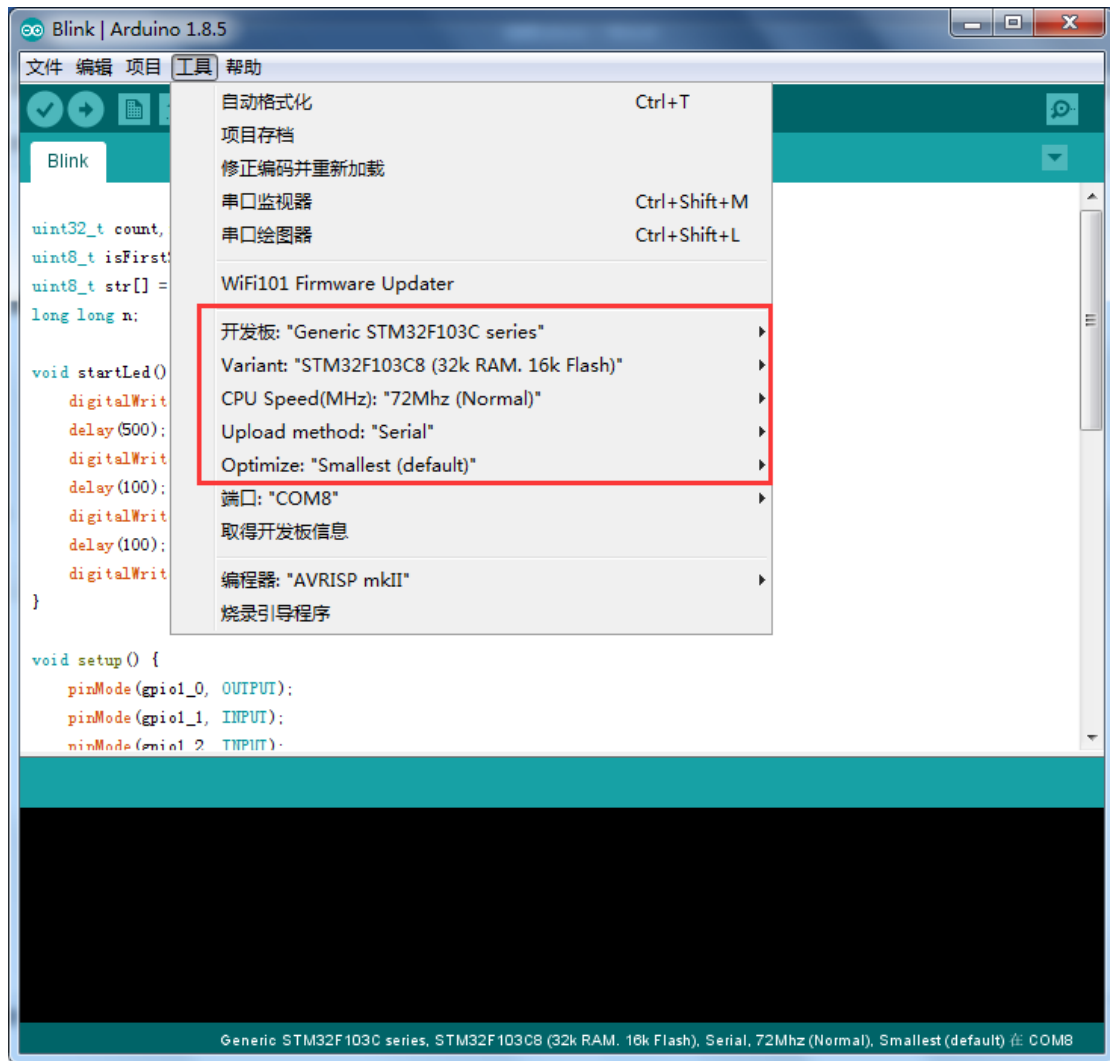
本例中需要用的程序包括：

 ArduinoProject	2020/2/11 11:00	文件夹
 FPGAProject	2020/4/8 13:19	文件夹

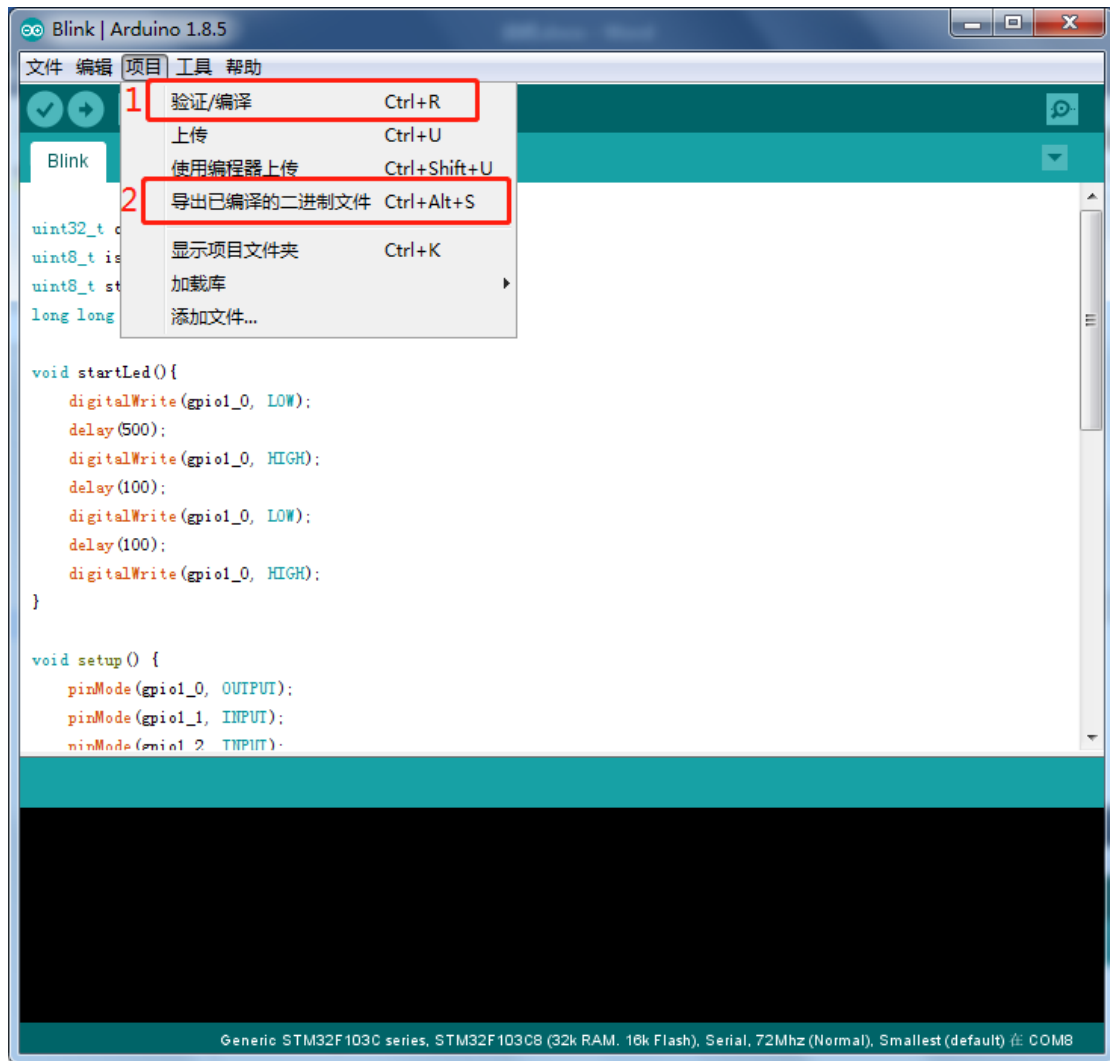
打开 ArduinoProject\Blink 目录中的 Blink.ino

名称	修改日期	类型	大小
 Blink.ino	2020/4/8 13:23	Arduino file	1 KB
 Blink.ino.generic_stm32f103c.bin	2020/4/8 13:19	BIN 文件	16 KB

在工具中做如下配置：



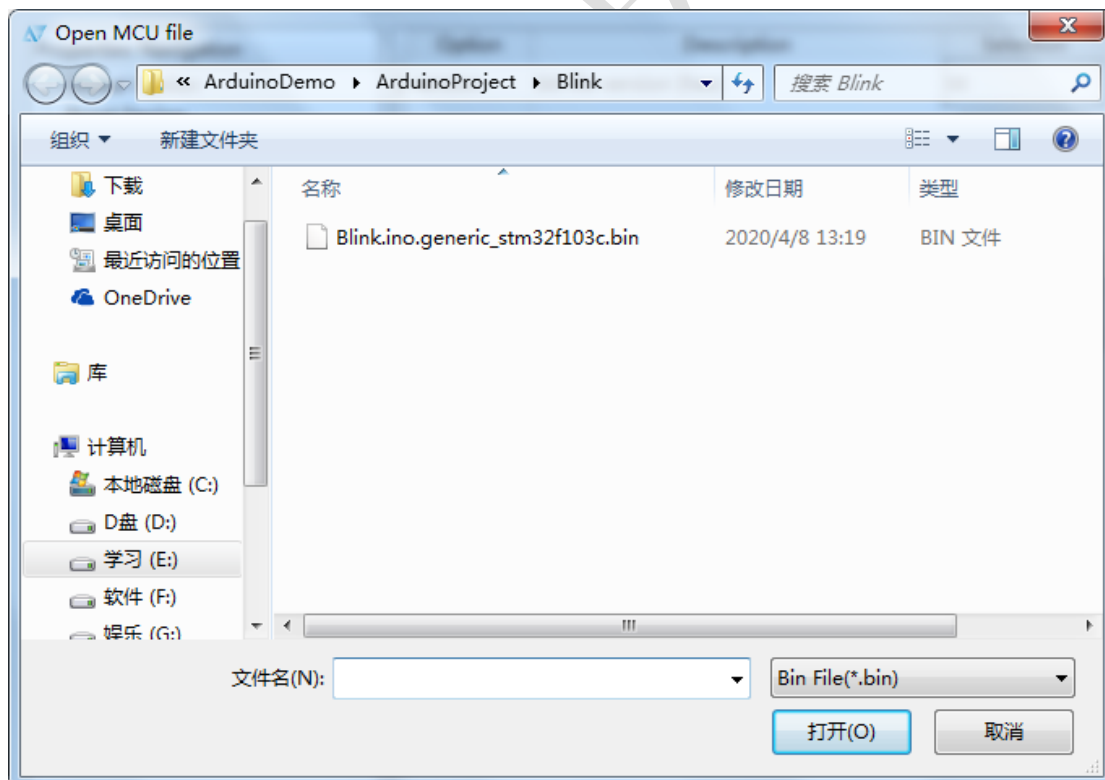
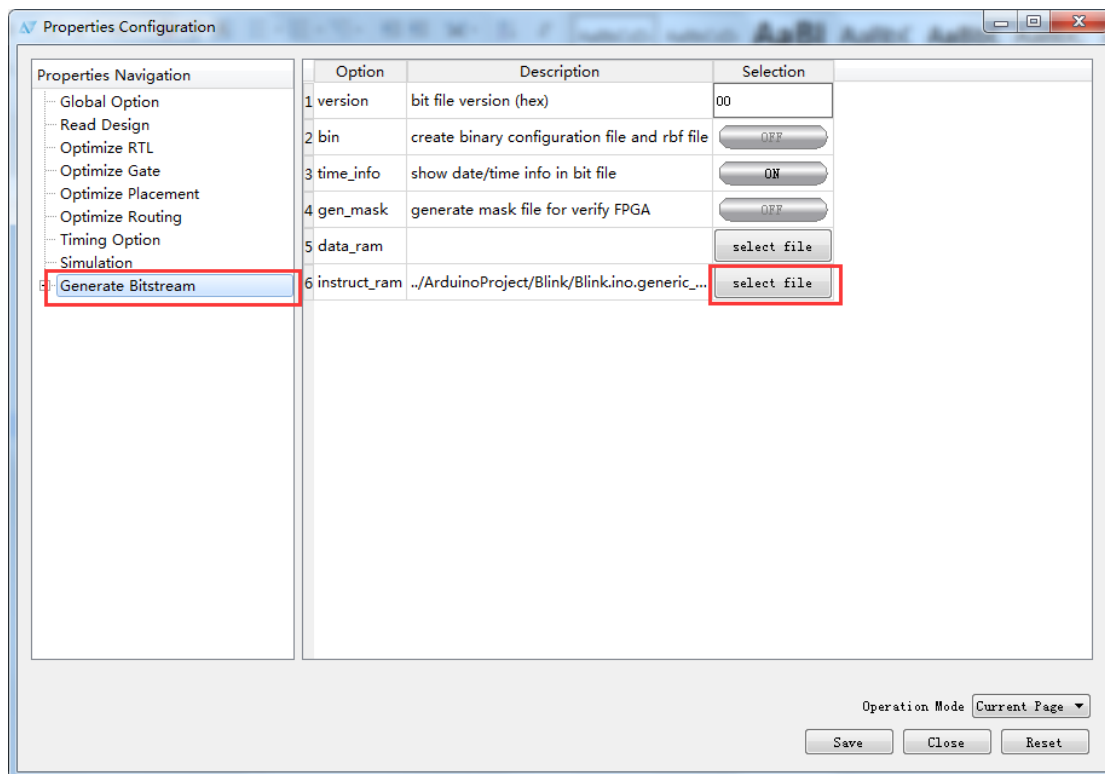
点击项目→编译，等待编译完成后导出已编译的二进制文件：



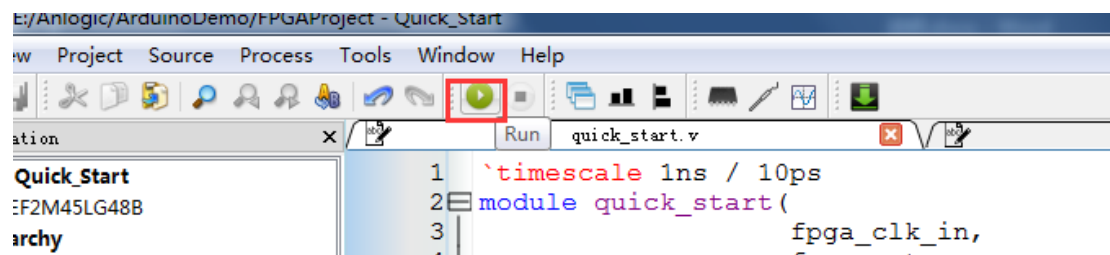
然后会在 Arduino 工程文件所在的位置生成.bin 文件，后面需要用到。

Blink.ino	2020/4/8 13:23	Arduino file	1 KB
Blink.ino.generic_stm32f103c.bin	2020/4/8 13:19	BIN 文件	16 KB

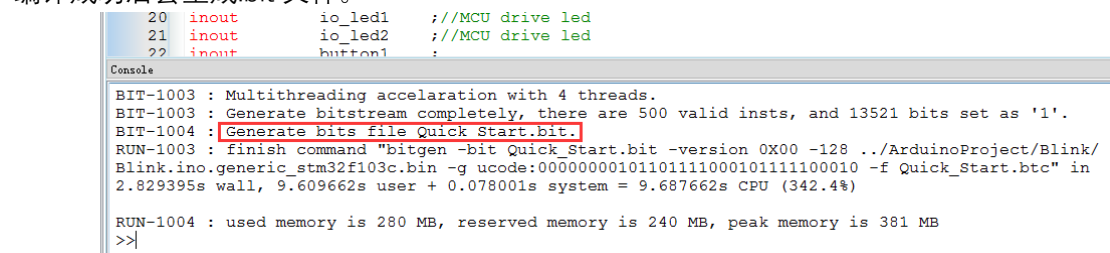
然后打开 FPGAProject 工程文件。点击 Process→Properties 进行配置，在 Generate Bitstream 中点击 6instruct\_ram 右侧的 select file，选择刚才生成的 Blink.ino.generic\_stm32f103c.bin 文件，点击 save 并退出来。



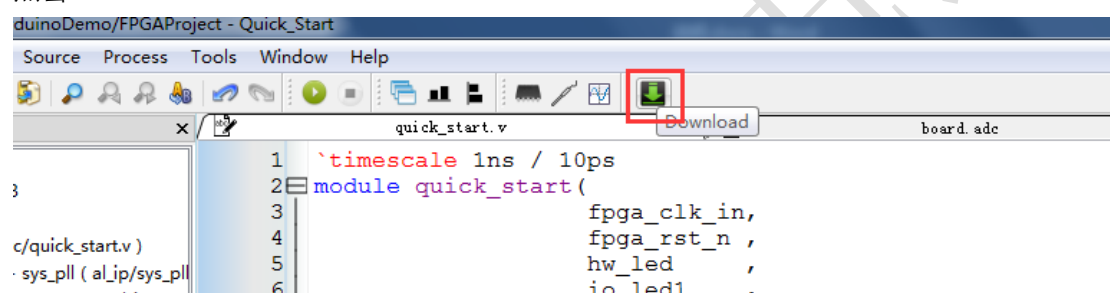
点击 RUN。



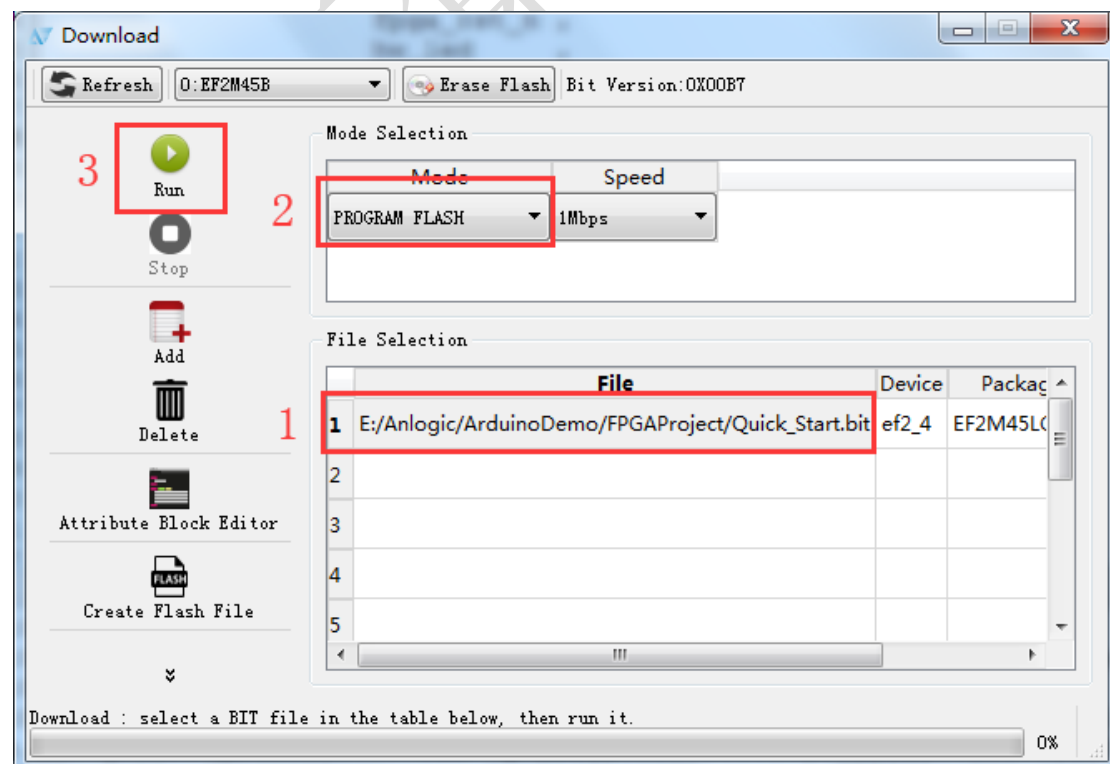
编译成功后会生成.bit 文件。



点击 Download



在 Download 界面中进行如下操作。



The screenshot displays the Anlogic IDE interface for the 'Anlogic/ArduinoDemo/FPGAProject/Quick\_Start' project. The 'Hierarchy' pane on the left shows the project structure, including 'quick\_start (src/quick\_start.v)', 'U.SYS.PLL - sys\_pll (a1ip/sys\_pll.v)', 'U.AL.MCU - AL\_MCU (a1ip/AL\_MCU.v)', 'u\_bram256kbit - bram256kbit (u\_bram256kbit.v)', 'constns/board.adc', and 'simulation/ChipWatcher.cwc'. The 'Project' pane shows 'User Constraints', 'HDL2Bit Flow', 'Read Design', 'Optimize RTL', 'Optimize Gate', 'Optimize Placement', 'Optimize Routing', 'Generate Bitstream', 'Download', and 'Design Summary'. The 'Download' dialog box is open, showing 'Mode Selection' (PROGRAM FLASH) and 'Speed' (1Mbps). The 'File Selection' table lists the file 'E:/Anlogic/ArduinoDemo/FPGAProject/Quick\_Start.bit' for device 'ef2\_4' (EP2M4510). The console output shows the download progress and completion:

```

RUN-1004 : used memory is 305 MB, reserved memory is 267 MB, peak memory is 390 MB
GUI-1001 : Download success!

```



---

## 4.注意事项

由于安路的板子上仅有 D1、D2、D3、D5 四个 LED，S2、S3 两个按键，本测试例程将 gpio1\_1 和 gpio1\_2 映射至 S2 和 S3 作为输入用，将 gpio1\_0 和 gpio1\_3 映射至 D2 和 D5 作为输出用，pwm 功能只是用了 9 号引脚作为输出，映射至 D3。所以如果想要使用其他 gpio 口作为控制口的话需要对 ArduinoProject 和 FPGAProject 同时进行相应更改。

