

DC1168C001 开发板快速入门指南

2022 年 3 月 - 版本 1

概述

DC1168C001 开发板使用 SPC1168APE48 作为主控制器。开发板可使用仿真器或外部 3.3V 电源为其提供 3.3V 供电电压。本开发板可方便使用者开发调试 GPIO, AES, CRC, TIMER, WDT, PWM, ECAP, ADC, T-SENSOR, PGA, COMP, DAC, UART, I2C, SSP 等外设资源。

本文主要描述该开发板的硬件设计、软件开发工具、例程使用方法，旨在帮助开发人员快速地进行应用程序的开发工作。

目录

1	快速导览	7
1.1	开发板介绍	7
1.2	文件夹结构	8
1.2.1	DOCUMENTS 文件夹结构	9
1.2.2	FIRMWARE 文件夹结构	9
2	软硬件环境使用	12
2.1	软件开发环境	12
2.2	硬件器件介绍	12
2.3	开发板供电接口	14
3	开发及调试方法	15
3.1	拷贝 Flash 下载算法程序	15
3.2	Keil 配置导引	15
3.2.1	版本检测	15
3.2.2	打开项目文件	16
3.2.3	配置检查 Optional	16
3.3	使用 JLINK 进行开发及调试	20
3.4	使用 ULINK 进行开发及调试	27
3.5	使用 UART 进行开发及调试	29
4	进阶说明	37
4.1	Demo 程序进阶说明	37
4.2	Keil IDE 进阶说明	38
5	修订记录	错误！未定义书签。

表格列表

表 2-1: SW 接口信号定义	13
表 5-1: 文档修订记录	6

图片列表

图 1-1:	开发板实物图	7
图 1-2:	开发板功能介绍图	8
图 1-3:	主文件夹结构	9
图 1-4:	DOCUMENTS 文件夹结构	9
图 1-5:	FIRMWARE 文件夹结构	10
图 1-6:	Libraries 子文件夹	10
图 1-7:	Project 子文件夹	11
图 2-1:	USB 转串口实物图	12
图 2-2:	JLINK 实物图	13
图 2-3:	JLINK 接口	13
图 2-4:	J-Link OB 实物图	14
图 2-5:	ULINK2 实物图	14
图 3-1:	拷贝 Flash 算法程序至 ARM 文件夹	15
图 3-2:	检查 Keil 版本	16
图 3-3:	点击 Keil Project	16
图 3-4:	Options window	17
图 3-5:	检查 Device	17
图 3-6:	检查 Target	18
图 3-7:	检查 Output	18
图 3-8:	检查 Listing tab	19
图 3-9:	检查 C/C++ tab	19
图 3-10:	GPIO 管脚信息	20
图 3-11:	JLINK 与开发板连接方式	20
图 3-12:	启动模式与 Debug 功能选择	21
图 3-13:	JLINK 下 Debug 配置界面	22
图 3-14:	JLINK 设置对话框	23
图 3-15:	Flash Download 设置	24
图 3-16:	Add Flash Programming Algorithm	24
图 3-17:	Cortex Jlink/JTrace Target Driver Setup	25
图 3-18:	Options 保存设置	25
图 3-19:	编译与编译结果	26
图 3-20:	用 JLINK 下载程序	26
图 3-21:	Debug window	27
图 3-22:	ULINK2 下 Debug 配置界面	28
图 3-23:	ULINK2 设置对话框	28
图 3-24:	ISP download tool	29
图 3-25:	编译工程	29
图 3-26:	编译后产生新的 Objects 文件夹	30
图 3-27:	ISP Tool 配置	30
图 3-28:	FT232 与开发板连接方式	31
图 3-29:	UART 波特率配置	32

图 3-30:	打开 PC 串口	33
图 3-31:	将 BOOT 模式选为 ISP	34
图 3-32:	开始下载程序	35
图 3-33:	BOOT 模式改为 NORMAL	35
图 3-34:	ISP Tool UART Communication	36
图 4-1:	GPIO_Level_Test 工程界面图	37
图 4-2:	ISR 文件图	38
图 4-3:	程序加载域与执行域配置	38
图 4-4:	编译时搜索路径	39

版本历史

表 5-1: 文档修订记录

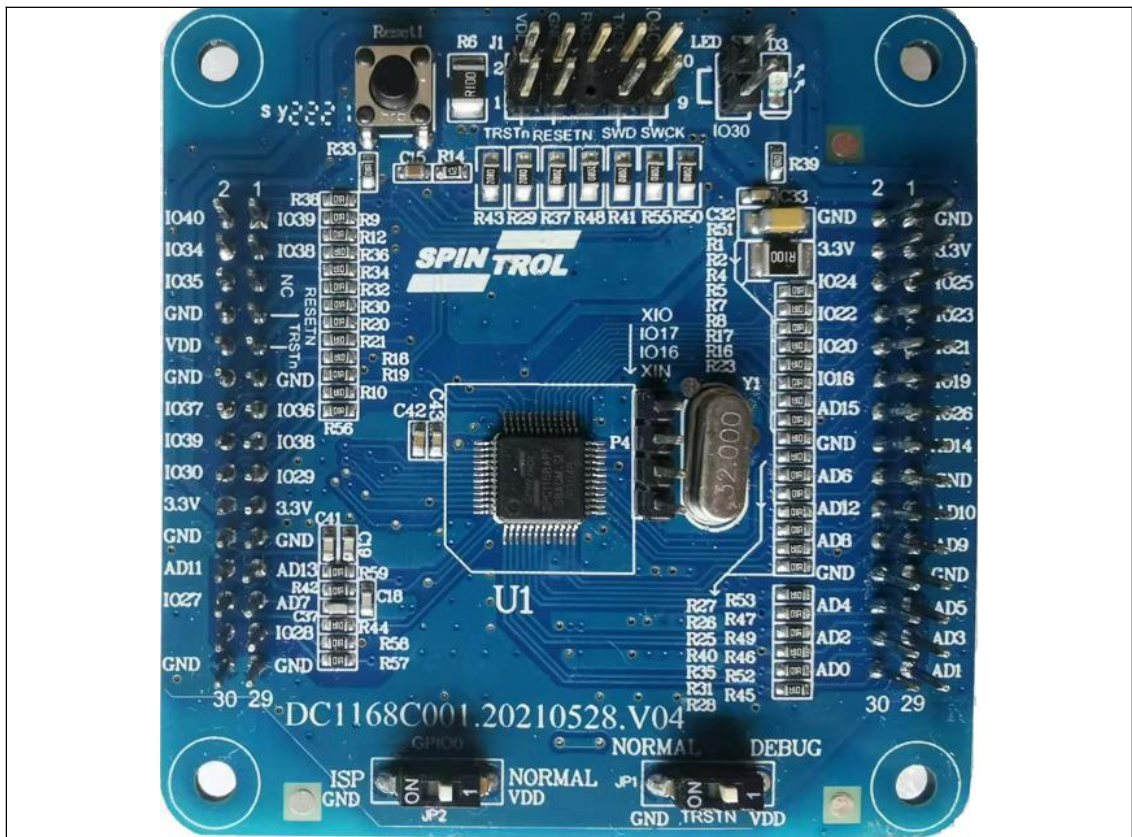
日期	版本	修改内容	作者
2022-03-22	1	初始版本	Feichen (fei.chen@spintrol.com)

1 快速导览

本指南包含了 DC1168C001 开发板的硬件介绍，软件开发环境，与开发调试方法。在本文件中，提供了手把手的教学，协助用户：

- (1) 下载一个基本的代码到 SPC1168
- (2) 使用 Keil 编辑器，并通过 JLINK 或 ULINK2 进行下载与调试
- (3) 使用串口 UART 进行代码下载
- (4) 了解 Demo 程序结构及 Keil 常用功能

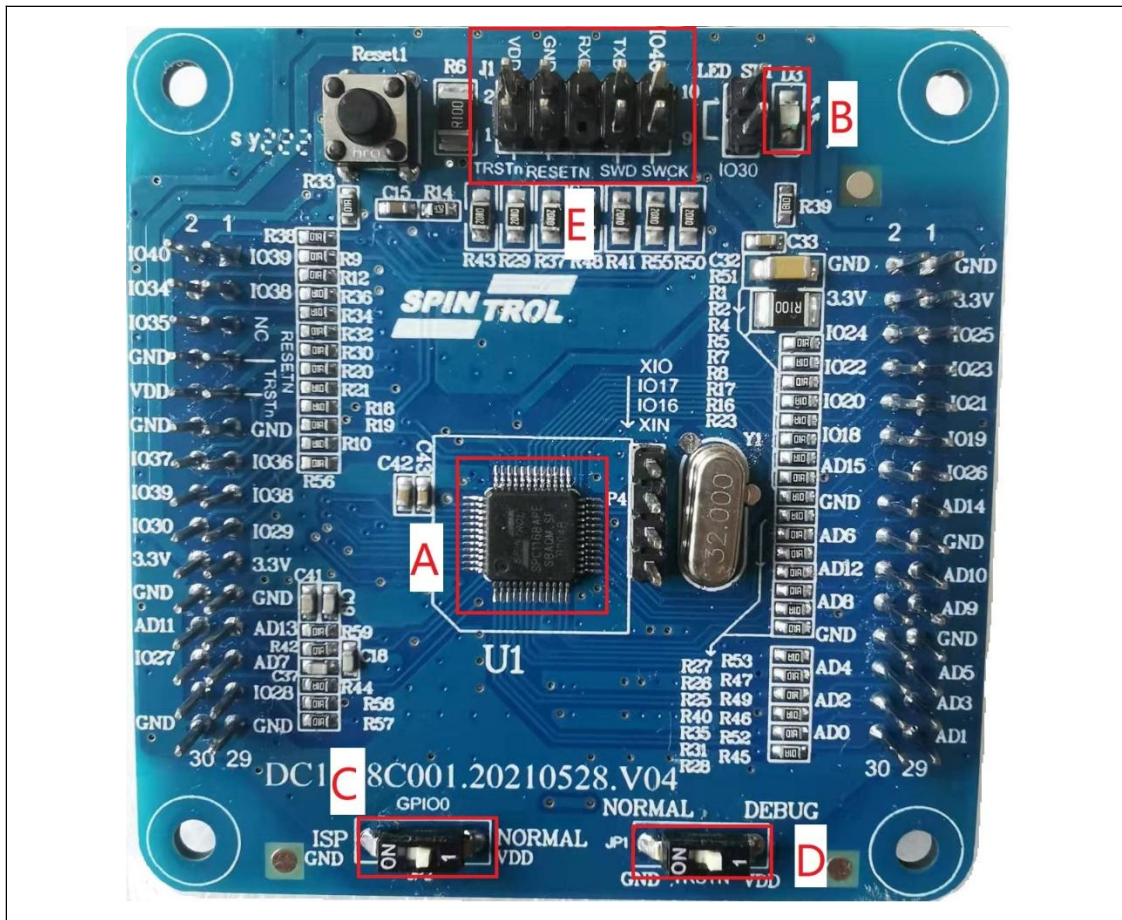
图 1-1: 开发板实物图



1.1 开发板介绍

图 1-2 蓝色开发板，代号为 DC1168C001，带有 SPC1168 芯片，LED 灯，芯片引脚均有拉出标准的排针，并附带兼容 JLINK 与 ULINK2 仿真器的接口，方便使用者开发与调试。

图 1-2: 开发板功能介绍图

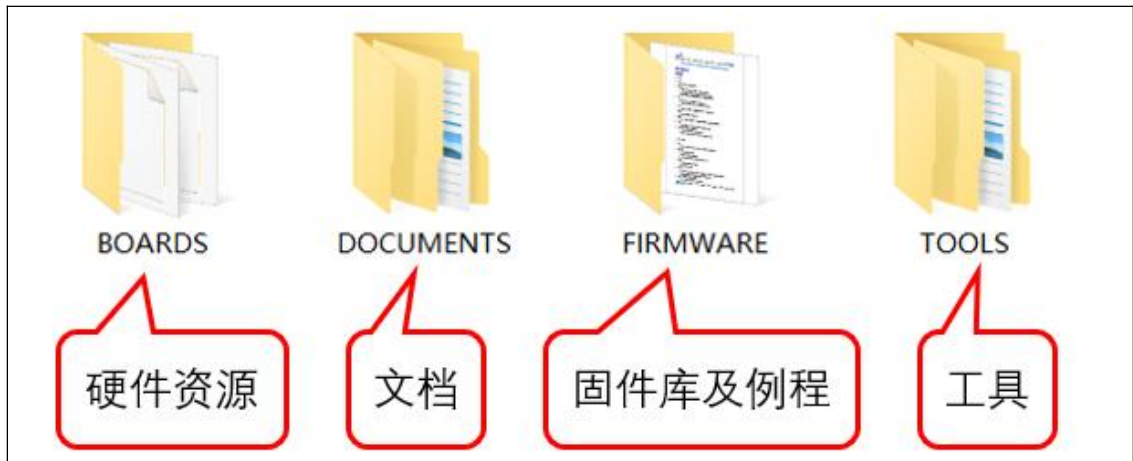


- A. SPC1168 芯片
- B. LED 灯，插上跳帽后可由 GPIO30 控制
- C. 芯片 BOOT(GPIO40)选择开关
向右拨(NORMAL)，芯片 Reset 后程序直接执行
向左拨(ISP)，Reset 后进入 UART 串口编程模式
- D. Debug 使能开关
向右拨(DEBUG)，芯片 Debug 功能开启，兼容 SWD 与 JTAG
- E. SWD 与 JTAG 调试接口，UART 接口
可使用 JLINK (或 J-Link OB)，ULINK2 进行调试，并相容于 Spintrol 研发之量产工具与调试器；推荐使用 SWD 接口，因为更省引脚而且调试功能不受影响

1.2 文件夹结构

针对本开发板，使用者会获取到对应的原理图及 PCB，主芯片手册，使用指南，ISP 下载工具等开发资料，可将获取到的资料存于不同文件夹中，便于管理。图 1-3 的主文件夹目录结构供使用者参考。

图 1-3：主文件夹结构

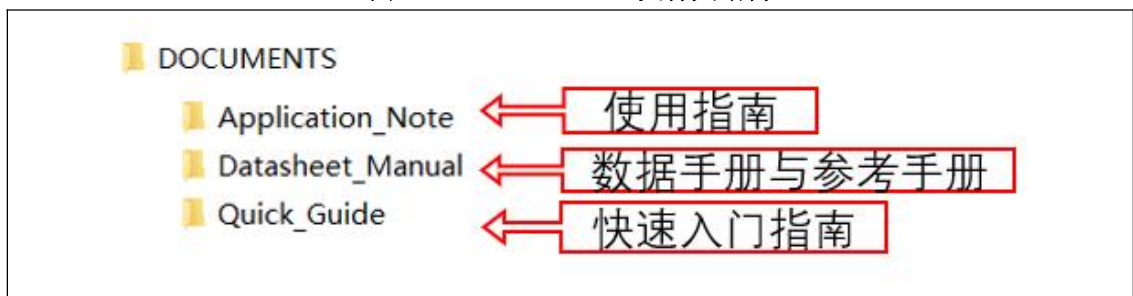


- **BOARDS:** 存放开发板对应的原理图和 PCB。
- **DOCUMENTS:** 存放开发板所使用的主芯片的使用指南，数据手册和参考手册，快速入门指南等文档。
- **FIRMWARE:** 存放固件库及所有 Demo 程序。
- **TOOLS:** 存放开发过程中用到的工具，比如 ISP 代码下载工具。

1.2.1 DOCUMENTS 文件夹结构

图 1-4 是其 DOCUMENTS 文件夹结构，供使用者参考。

图 1-4：DOCUMENTS 文件夹结构



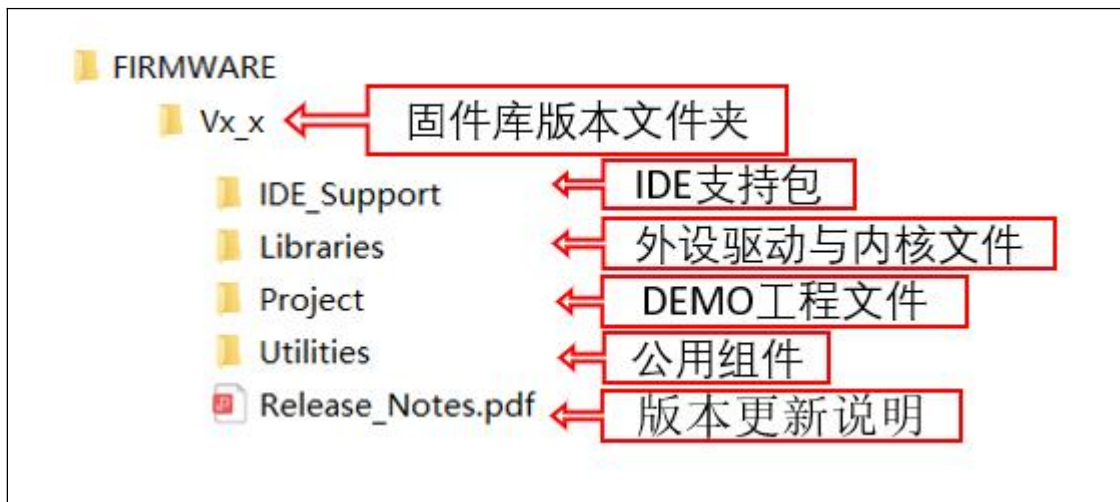
- **Application_Note:** 使用指南即应用文档。针对不同外设做出的一些解决方案的应用实例，都会在该文档里面进行说明。文档中既包含功能性的描述，也有搭配的程序示例。
- **Datasheet_Manual:** 数据手册中包含芯片引脚定义、电气特性、机械封装等信息。参考手册包含对芯片每个模块的具体描述和功能介绍。
- **Quick_Guide:** 存放该开发板的快速入门指南文档。

1.2.2 FIRMWARE 文件夹结构

所有的 Demo 程序都支持 Keil 及 IAR 两种 IDE 开发工具，在之后介绍的所有目录中，名字为“EWARM”的子目录是支持 IAR 的软件文件夹，名字为“MDK-ARM”的子目录是支持 Keil 的软件文件夹。

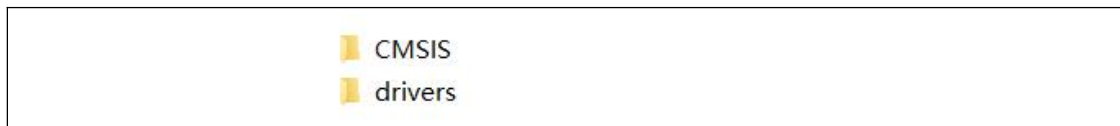
图 1-5 是 FIRMWARE 文件夹结构。其中 Vx_x 为第一级子目录文件夹，存放对应版本的固件库。

图 1-5: FIRMWARE 文件夹结构



- **IDE_Support:** 包括了对 Keil 以及 IAR 进行支持所需的软件，例如 Keil 中所需的 Flash 烧录算法文件（FLM 文件）及 SFR 文件，这些文件属于比较高阶的玩法，新手用户可以不用理解其作用机理。
- **Libraries:** 此文件夹包含了产品所支持的外设驱动，以及 ARM 核的启动文件及一些特殊函数接口实现等。此文件夹又包含了如图 1-6 两个文件夹：

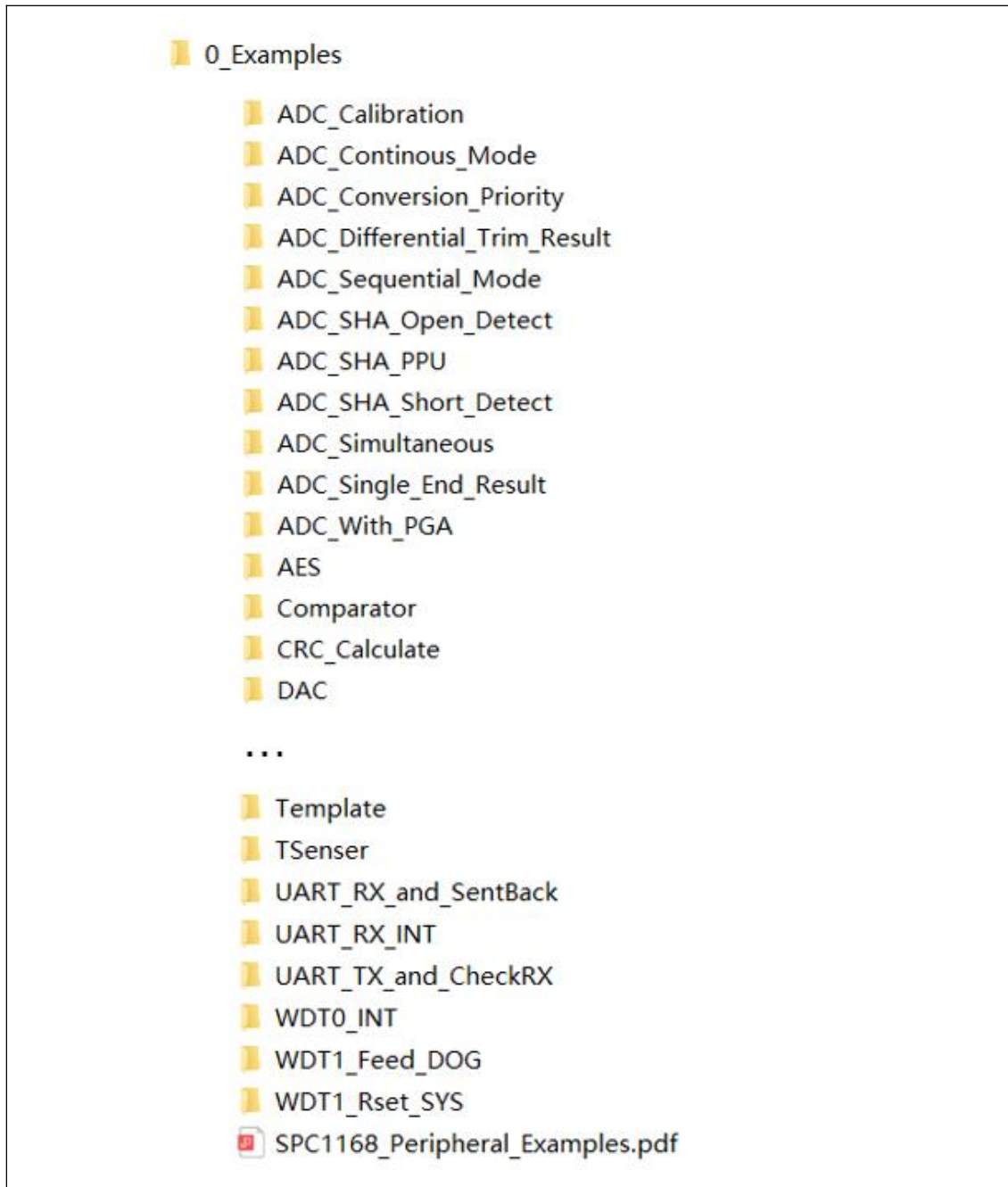
图 1-6: Libraries 子文件夹



其中 CMSIS 包含了 ARM 核启动文件及特殊函数的实现，drivers 文件夹包含了产品所支持的外设的驱动函数。

- **Project:** 文件夹下是用驱动库写的针对官方发行 demo 板的例子和工程模板。如图 1-7 所示，列出了部分 demo 例子。其中 Tmeplate 为工程模板，SPC1168_Peripheral_Examples.pdf 文件是对示例工程的说明文档。

图 1-7: Project 子文件夹



- Utilities: 包含了延时函数实现及打印信息输出重定向接口的实现。
- Release_Notes.pdf: 库的版本更新说明。

2 软硬件环境使用

2.1 软件开发环境

所有例程均提供了 Keil 和 IAR 两个版本。
建议使用 Keil 集成开发环境且安装 Keil5 之后的版本。

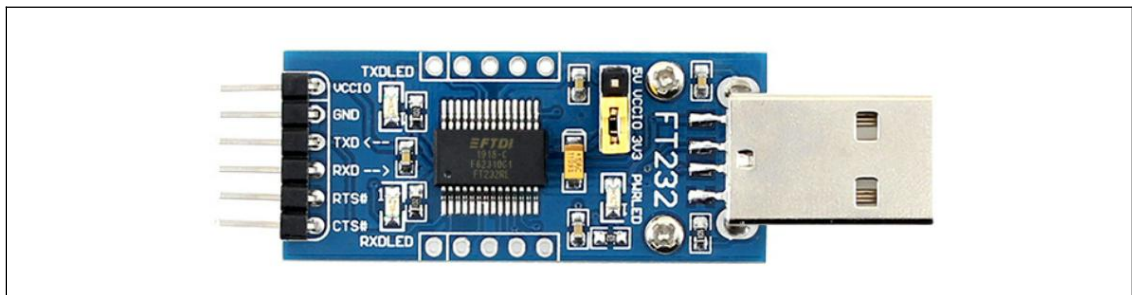
2.2 硬件器件介绍

最基础的硬件开发环境，除了个人 PC 之外，还包括：3.3V 供电、开发板、USB 转串口、JLINK（或 J-Link OB）或 ULINK2。有了这些硬件之后就可以进行开发工作了，下面就这些硬件器件做简单的介绍。

- 3.3V 供电：因为开发板输入电压为 3.3V，所以需要有一个 3.3V 的供电电源。
- 开发板：对于不同的型号的产品，都有一块供用户进行开发使用的最小系统开发板，这些开发板都引出了绝大部分的芯片管脚，用户可根据这些管脚开发自己需求的功能。
- USB 转串口：由于 USB 接口的便利特性，几乎所有的个人 PC 机都带有 USB 接口；而传统的 RS232 串口由于接口物理体积大，越来越少的出现在个人 PC 上，所以强烈建议用户使用 USB 转串口工具，本文档中使用 FT232 模块工具。

在进行开发工作时，USB 转串口可进行程序烧录，以及信息打印，后续将会做详细用法介绍。USB 转串口工具上每个排针脚都标注有名字，如图 2-1，其中 VCCIO 用于输出 3.3V 或 5V（模块由 USB 供电，针对本开发板须将跳线帽跳到 3.3V），GND 接开发板的 GND，TXD 接开发板 MCU 的 RXD，RXD 接开发板 MCU 的 TXD。实际开发中，只需要 VCCIO、RXD、TXD、GND 四根连接线即可。

图 2-1：USB 转串口实物图



- JLINK：JLINK 是 SEGGER 公司为支持仿真 ARM 内核芯片推出的 JTAG 仿真器。简单地说，是给你一个 JTAG 协议转换盒。其连接到计算机用的是 USB 接口，而到目标板内部用的还是 JTAG 协议。它完成了一个从软件到硬件转换的工作。JLINK 支持 ARM7/ARM9/ARM11，Cortex M0/M1/M3/M4，Cortex A4/A8/A9 等内核芯片，支持 IAR、KEIL 开发。与 IAR，Keil 等编译环境无缝连接，操作方便、连接方便、简单易学，是学习开发 ARM 最好最实用的开发工具。

图 2-2 为 JLINK 实物图。JLINK 支持 2 种接口，如图 2-3 所示。推荐使用 SWD 接口，因为更省引脚而且调试功能不受影响。表 2-1 列出了 SW 接口信号定义。

图 2-2: JLINK 实物图



图 2-3: JLINK 接口

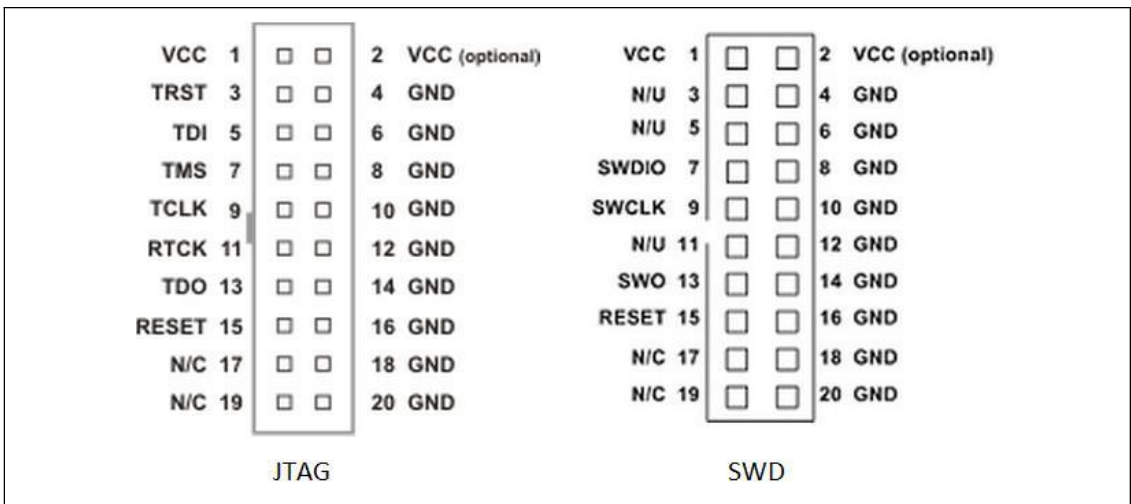
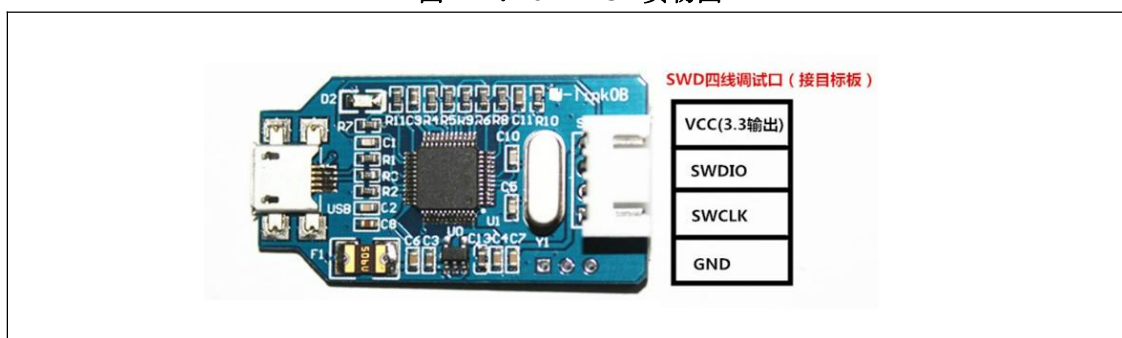


表 2-1: SW 接口信号定义

Signal	Connects to...
SWDIO	Data I/O pin
SWCLK	Clock pin
VCC	Positive Supply Voltage, the pin is optional
GND	Digital ground
RESET	RSTIN PIN, the pin is optional

- J-Link OB: J-Link OB 是一种便携式在线调试工具。J-Link OB 上每个排针脚标注有名字（在 J-Link OB 的背面），图 2-4 对四个排针脚作了标注，分别是 VCC（3.3V 输出），SWDIO（数据线），SWCLK（时钟线），GND（地线）。

图 2-4: J-Link OB 实物图



- ULINK2: ULINK2 是一款通用的 ARM 核调试仪器，可以通过 JTAG/SWD 接口与 ARM 核通信，手册中后续将会演示如何使用 SWD 接口调试程序。

图 2-5: ULINK2 实物图



2.3 开发板供电接口

开发板的输入电压为 3.3V，用户需要将外部 3.3V 电源接到标有 3.3 和 GND 的任意两个管脚上。这里需要说明的是，虽然用户拿到的开发板可能会由于产品型号的不同而形状各异，但总能在板子上找到类似的 3.3V 供电区域。

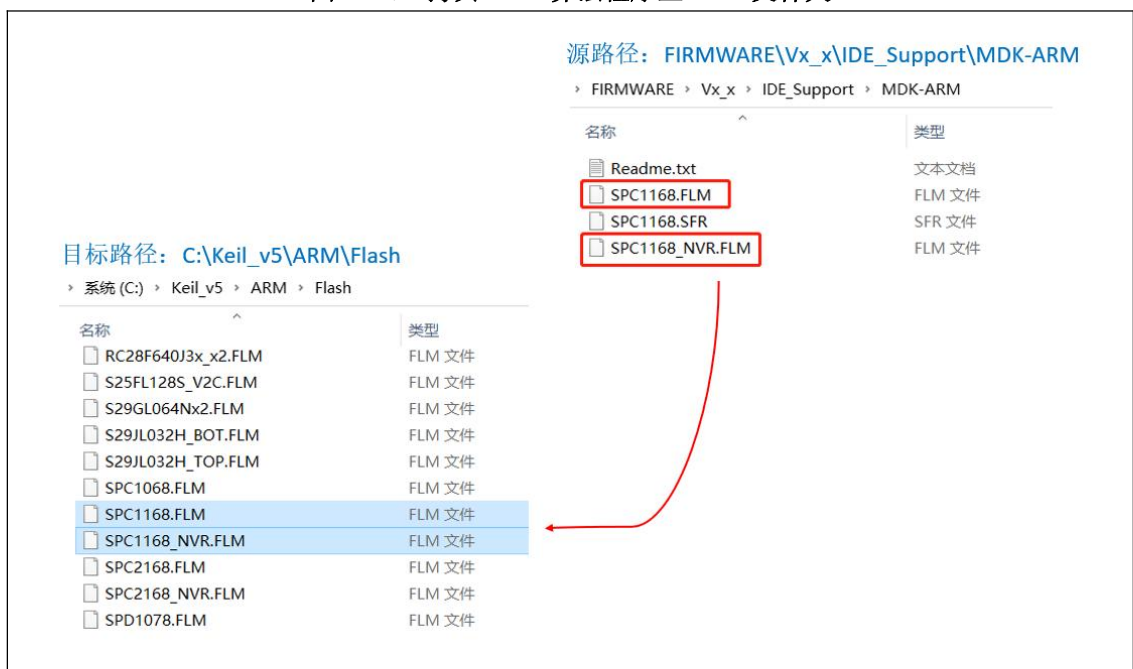
3 开发及调试方法

SPC1168 的软件开发包中，已经为用户准备好了一个标准嵌入式系统框架 demo 范例码，内有常用模块使用范例。通过本章节，使用者将了解如何从无到有设定好开发环境，并且将此代码编译后烧入芯片中，快速开始第一个专案。

3.1 拷贝 Flash 下载算法程序

如图 3-1 所示，请将 FIRMWARE\Vx_x\IDE_Support\MDK-ARM 文件夹下的所有 FLM 文件拷贝至 C:\Keil_v5\ARM\Flash 文件夹下。其中 Keil_v5 为安装 Keil5 的路径，若用户并非安装在预设 C:位置，请自行修正。

图 3-1: 拷贝 Flash 算法程序至 ARM 文件夹



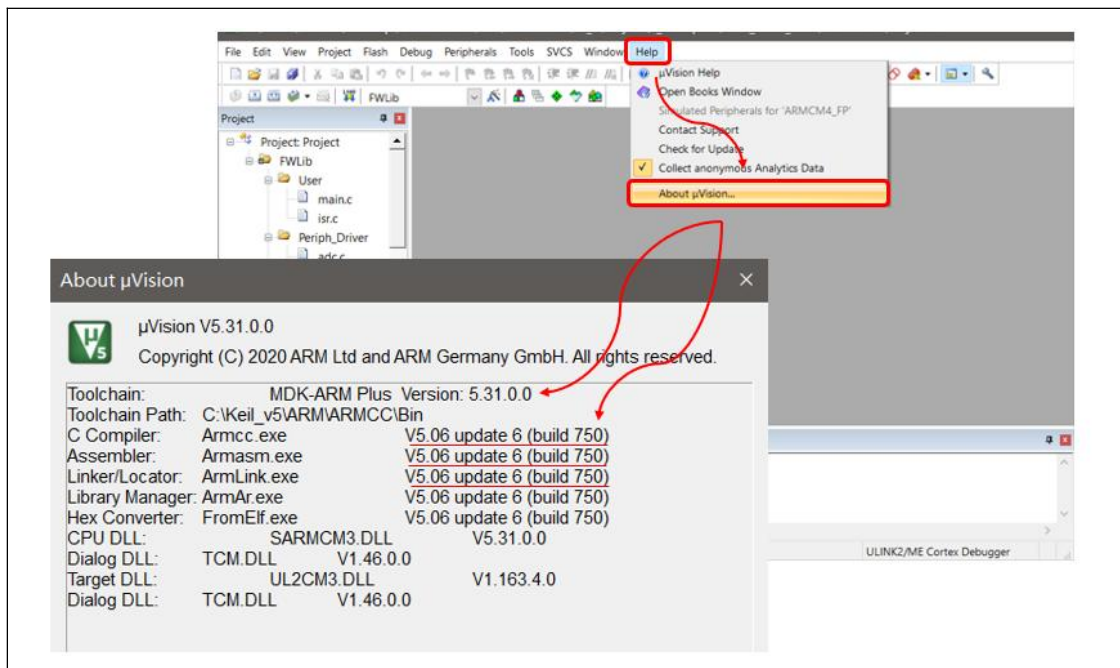
3.2 Keil 配置导引

建议使用者参考以下描述，检查是否所有设置均一致。

3.2.1 版本检测

建议安装 Keil 5 之后的版本。

图 3-2: 检查 Keil 版本

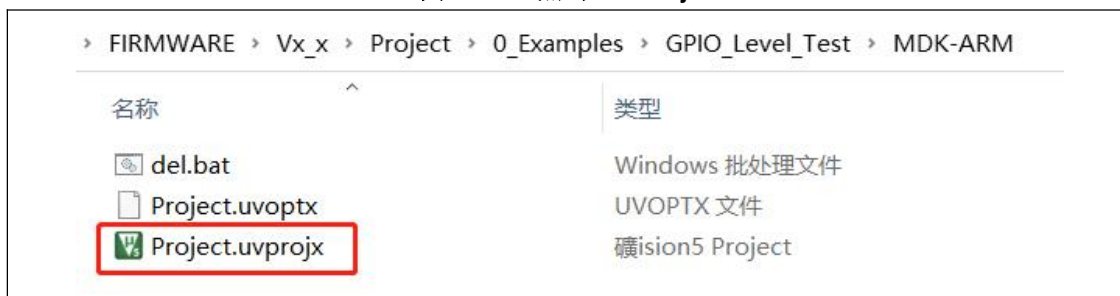


3.2.2 打开项目文件

本章节均以 GPIO_Level_Test 例程作为演示示例。

打开以下路径的项目文件夹：FIRMWARE\Vx_x\Project\0_Examples\GPIO_Level_Test\MDK-ARM，点击 Project.uvprojx 即可打开项目工程。

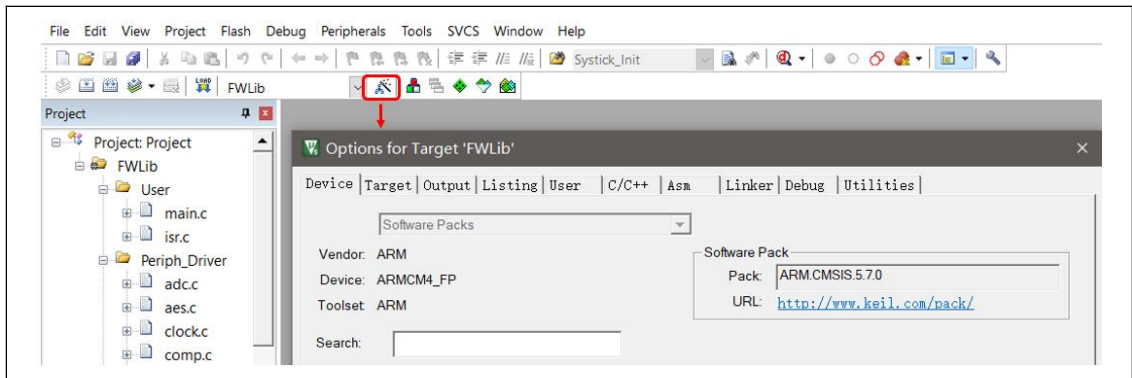
图 3-3: 点击 Keil Project



3.2.3 配置检查 Optional

请点击魔法棒，开启 Optional 设定窗口。

图 3-4: Options window



请检查 Keil 配置如下列组图中所述。

图 3-5: 检查 Device

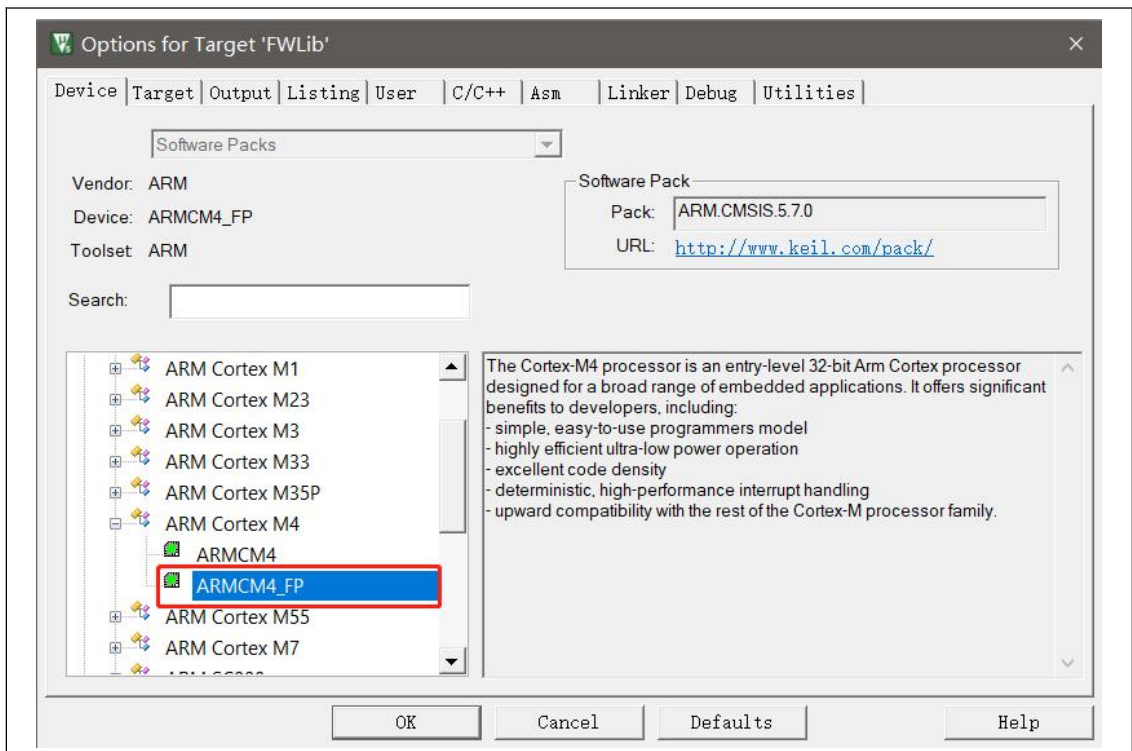
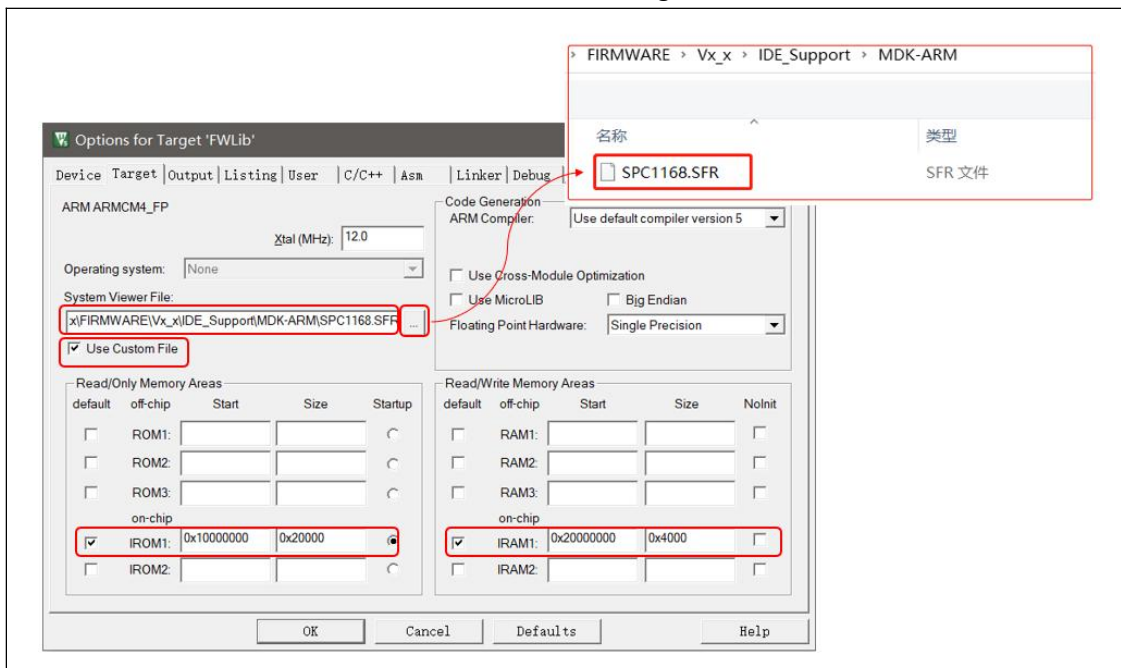


图 3-6: 检查 Target



SFR 文件所在的位置: FIRMWARE\Vx_x\IDE_Support\MDK-ARM\SPC1168.SFR。

在开发过程中, 难免会碰到需要查看芯片寄存器实时数值的情况, 使用 SFR 文件, 就为开发者提供了一个标准查看寄存器数值的方法。

图 3-7: 检查 Output

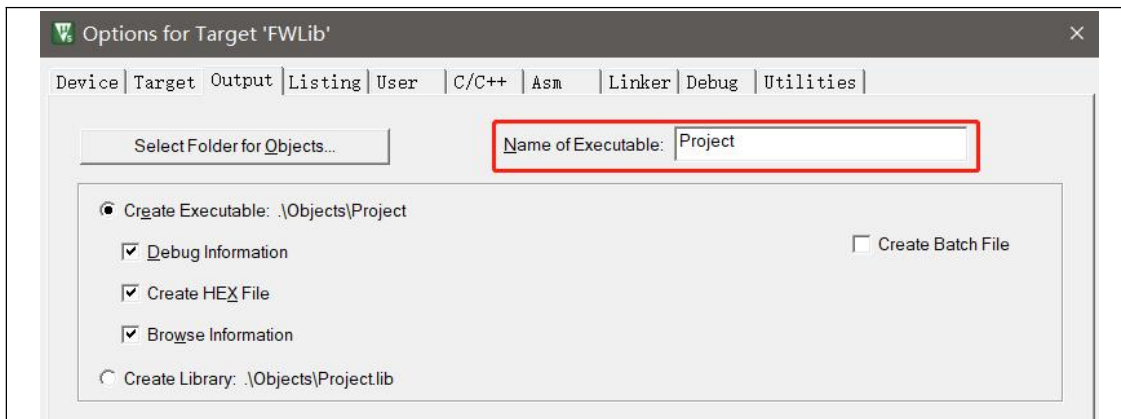


图 3-8: 检查 Listing tab

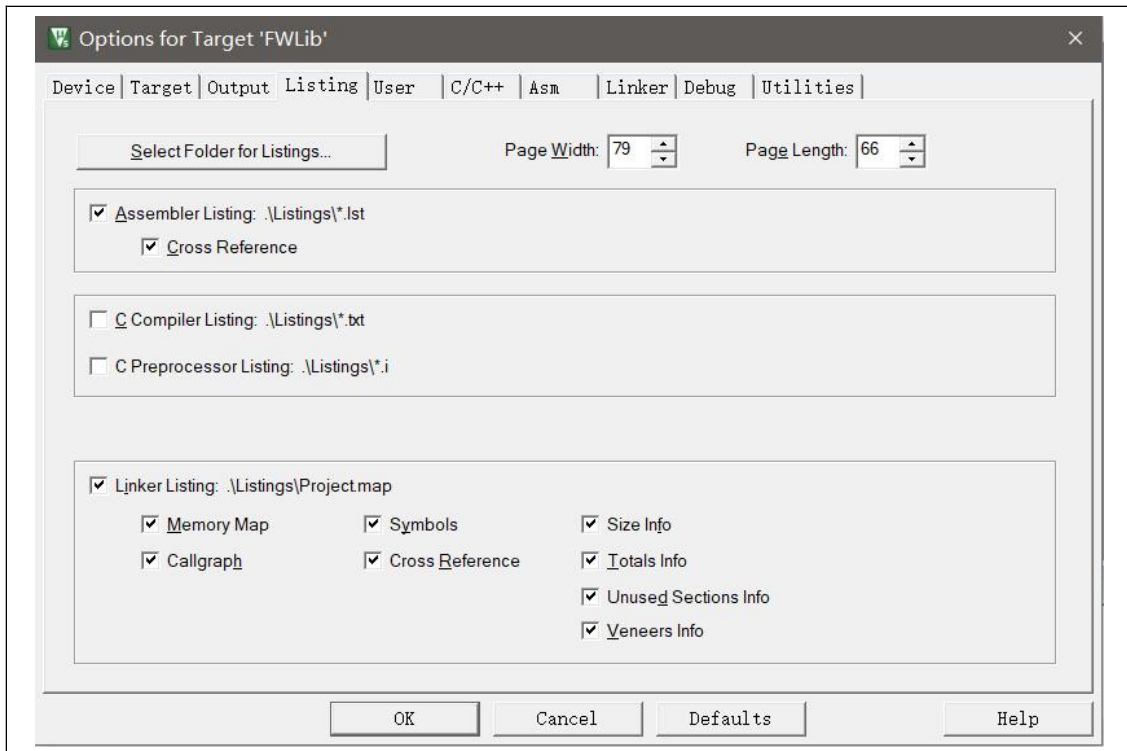
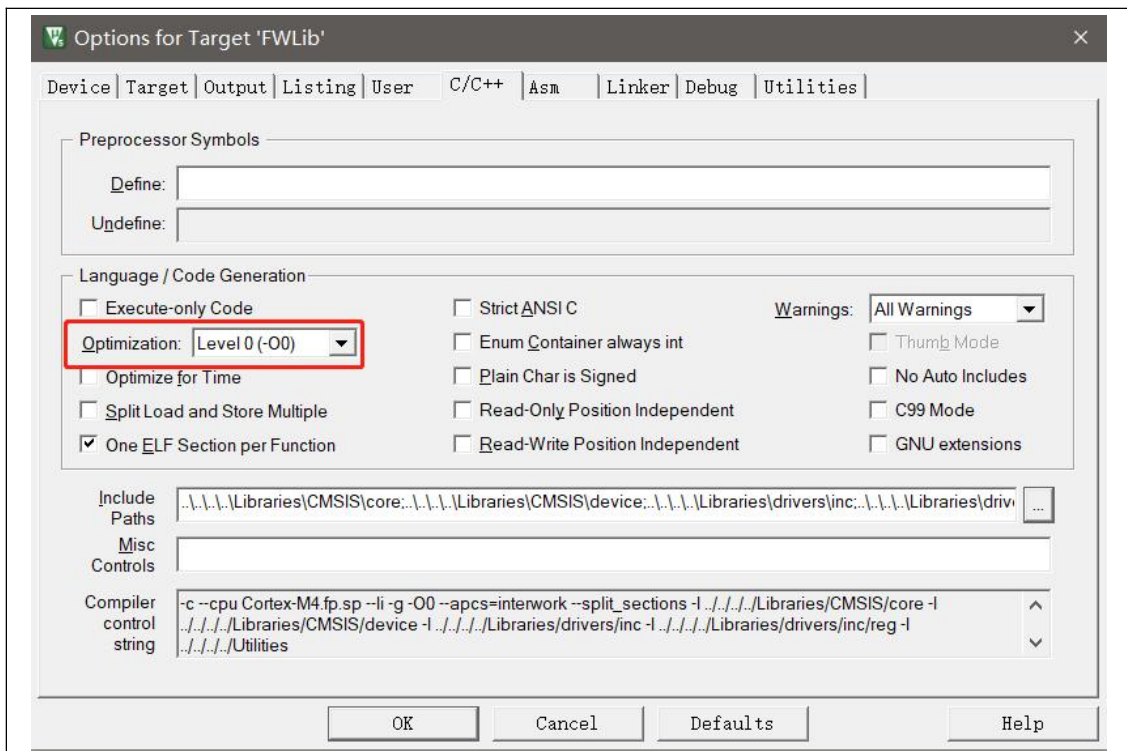


图 3-9: 检查 C/C++ tab



注：开发初期，C 的 Optimization 建议选择 O0。使用 O2 与 O3 虽然有较佳的代码大小或是运算效率，但是在调试时变量数值不一定与预期相符合，即使运算结果正确，却容易造成开发困难。

3.3 使用 JLINK 进行开发及调试

JLINK 是市面上常见的 Debug 调试器，有众多版本，本范例中使用 J-Link OB。

Step1

J-Link OB 支持 SWD 调试，使用上只需要把 3.3V，GND，SWCLK，SWDIO 四条线接好即可，请依照 J-Link OB 上的标志与开发板上的标志对接。

注：J-Link OB 支持 3.3V 电源供应，不需额外使用电源。

用户可以在《SPC1168 技术参考手册》中第 4 章节找到以下表格的信息，如图 3-10 所示，GPIO38/39 可以用作 SWD 接口的 SWD（数据线）/SWCK（时钟线）。

图 3-10: GPIO 管脚信息

引脚	通道 (红色表示默认通道)							
	0	1	2	3	4	5	6	7
GPIO37	GPIO37	TDI	UART_TXD	SPI_SFRM	PWM5B	PWM1B	SIO0_15	I2C_SCL
GPIO38	GPIO38	TMS/SWD	I2C_SDA	SPI_MOSI	SPI_MISO	PWM2A	SIO0_16	ECAPO
GPIO39	GPIO39	TCK/SWCK	I2C_SCL	SPI_MISO	SPI_MOSI	PWM2B	SIO0_17	ECAPO
GPIO40	GPIO40	SPI_SCLK	UART_TXD	DCLK			SIO0_0	ECAPO

图 3-11: JLINK 与开发板连接方式

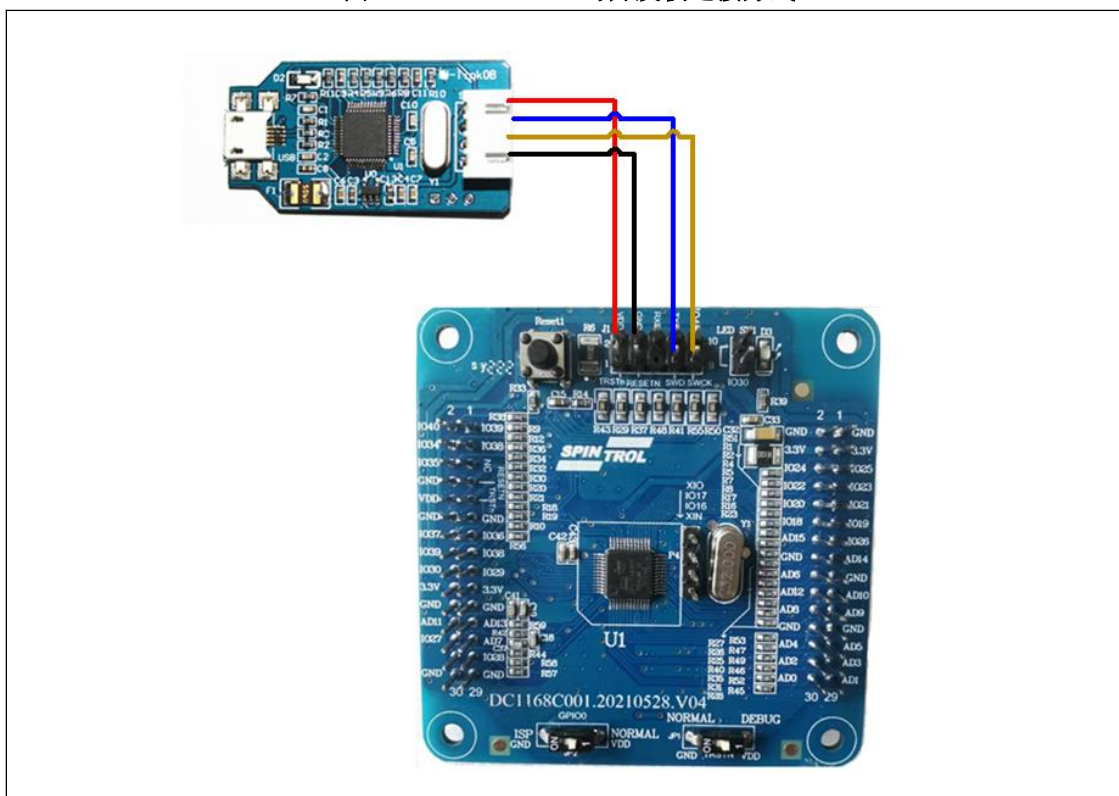


图 3-13: JLINK 下 Debug 配置界面

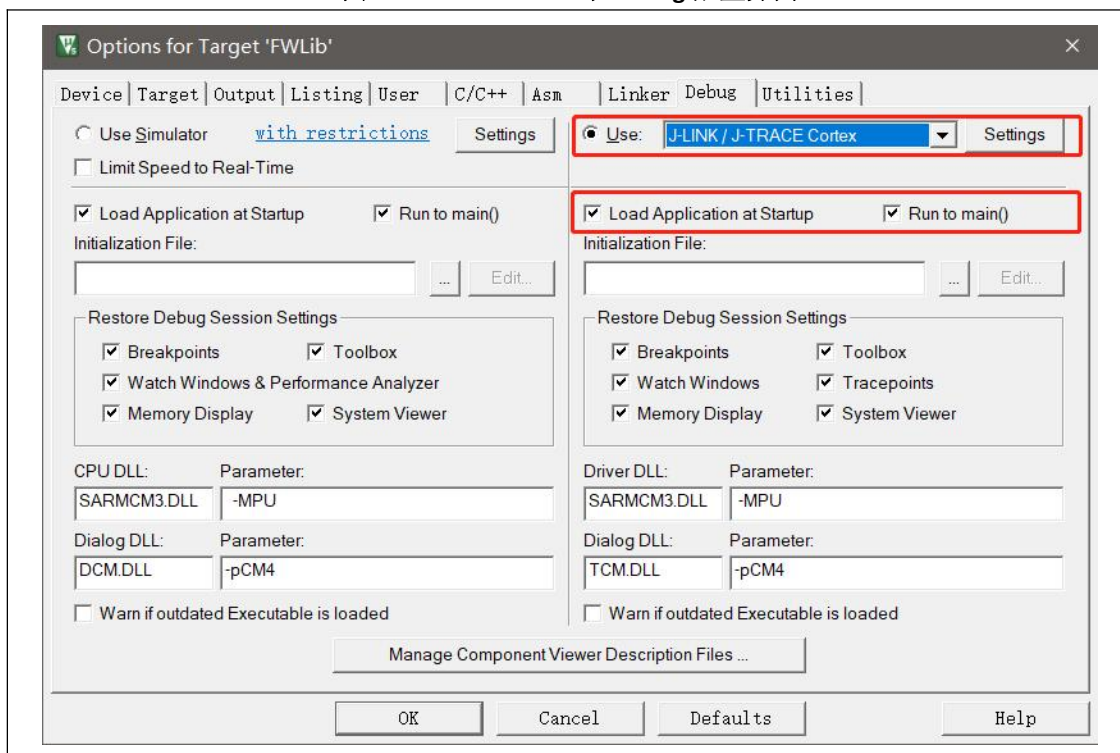


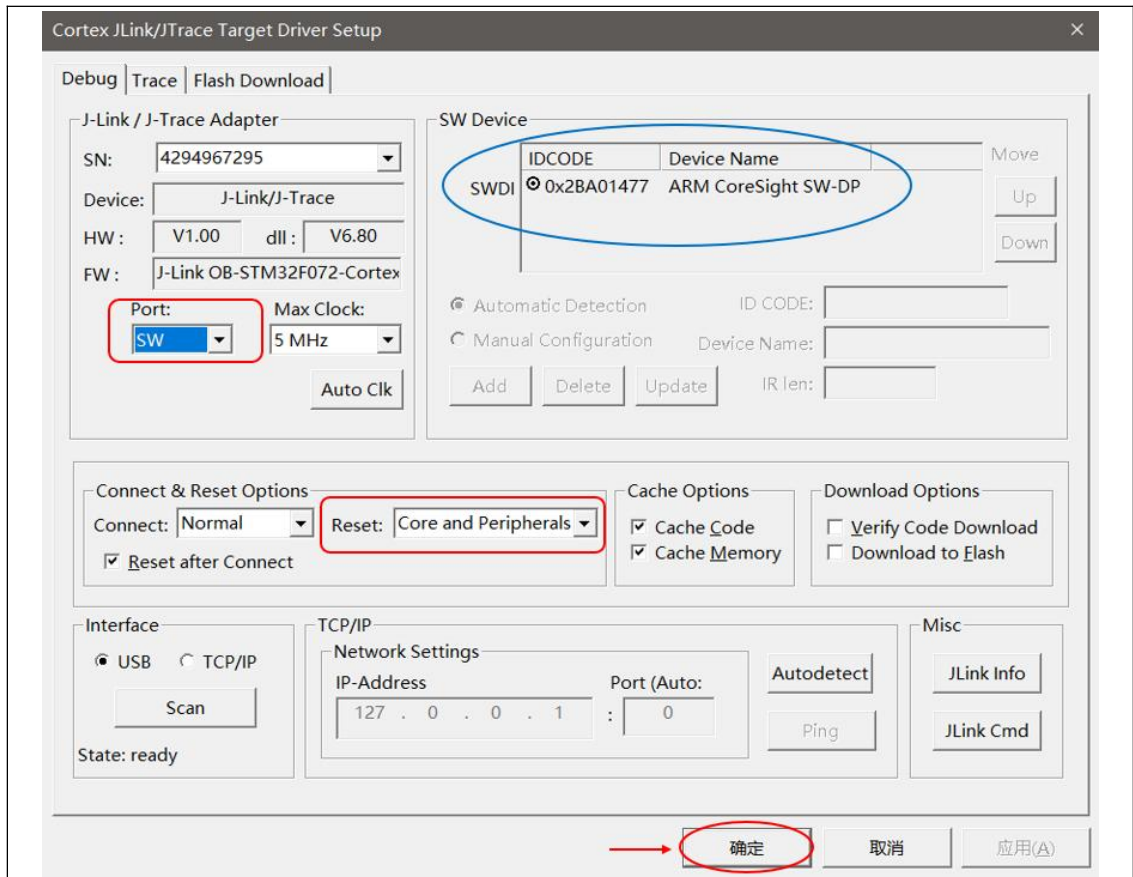
图 3-13 所示界面中，左侧是仿真调试相关的配置选项，右侧则是与硬件调试相关的选项。

本节使用 JLINK 硬件进行调试，所以选择使用 J-LINK/J-TRACE Cortex 选项。

下方还有两个选项：Load Application at Startup 和 Run to main()，Load Application at Startup 选项是必须要勾选的，Run to main()选项根据需要决定要不要勾选。

点击 Settings 按钮后，会弹出与 JLINK 相关的设置，如图 3-14 所示。可以看到，蓝色椭圆形框中出现相应设备号及名称，只有当 SW Device 显示芯片 IDCODE 时才能算是连接成功。此外，建议用户按照图 3-14 配置 Connect & Reset Options，Reset 方式选择 Core and Peripherals。此外，SPC1168 芯片支持 JTAG 和 SWD 两种 Debug 协议，用户可以根据需要进行配置，本节使用 SWD 协议。

图 3-14: JLINK 设置对话框



注:

此步骤最常出现无法与芯片连接的情况，若发生该问题，可以检查：

- (1) 芯片的 TRSTn 是否没有拉到高电位，拉到高电位后，按开发板上的 Reset 键；
- (2) JTAG 或 SWD 走线的滤波电容是否过大，建议可先去除滤波电容。

Step4

在使用 JLINK 调试程序之前，还需要设置 Flash Download 选项，如图 3-15 所示。

使用仿真器调试程序时，必须保证 Flash 存储器的程序与当前程序一致，这就需要用户每次修改代码后，都要点击 LOAD 按钮将程序下载到 Flash 存储器中。而 KEIL 软件提供了一个功能，可以自动完成上述动作。用户只需勾选 Update Target before Debugging 选项，那么在每次启动 Debug 会话时，KEIL 软件会自动通过仿真器设备将程序下载到 Flash 中，从而保证 Flash 中的程序与当前调试的程序一致。

Programming Algorithm 可以通过点击 Add 按钮来添加对应的 Flash 算法程序，如图 3-16 所示。

(注：必须将 FIRMWARE\Vx_x\IDE_Support\MDK-ARM 目录下的 SPC1168.FLM 文件复制到 KEIL 软件安装路径下的目录 C:\Keil_v5\ARM\Flash，否则不会出现 SPC1168 相关选项。)

图 3-15: Flash Download 设置

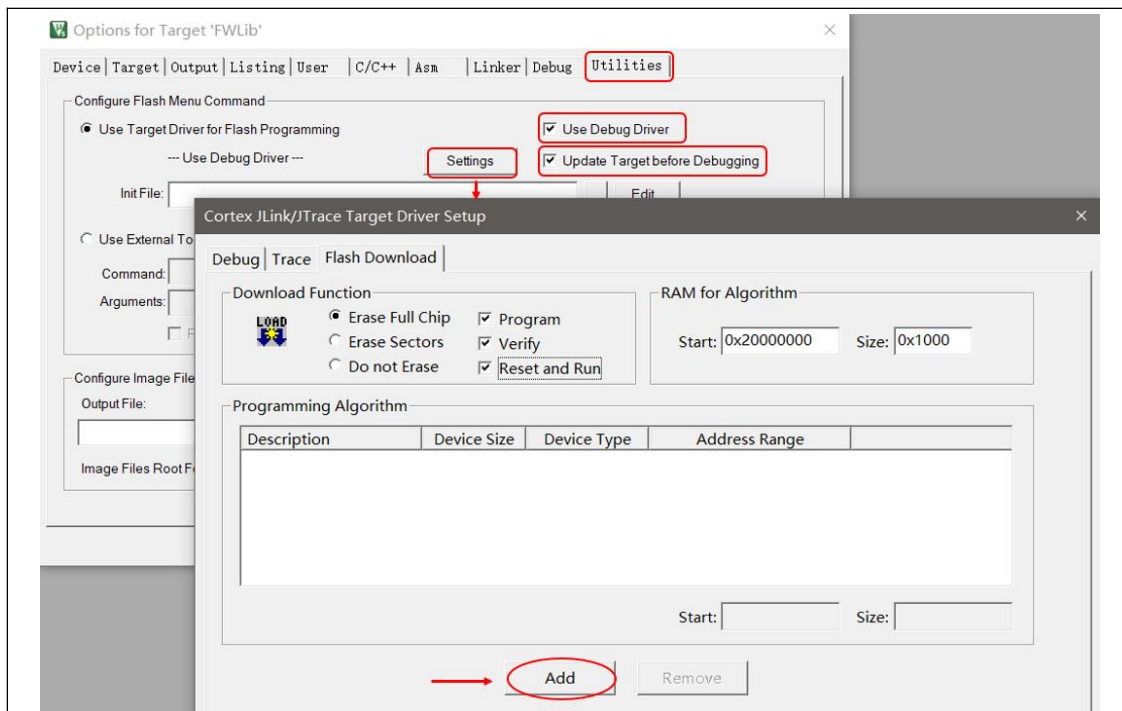
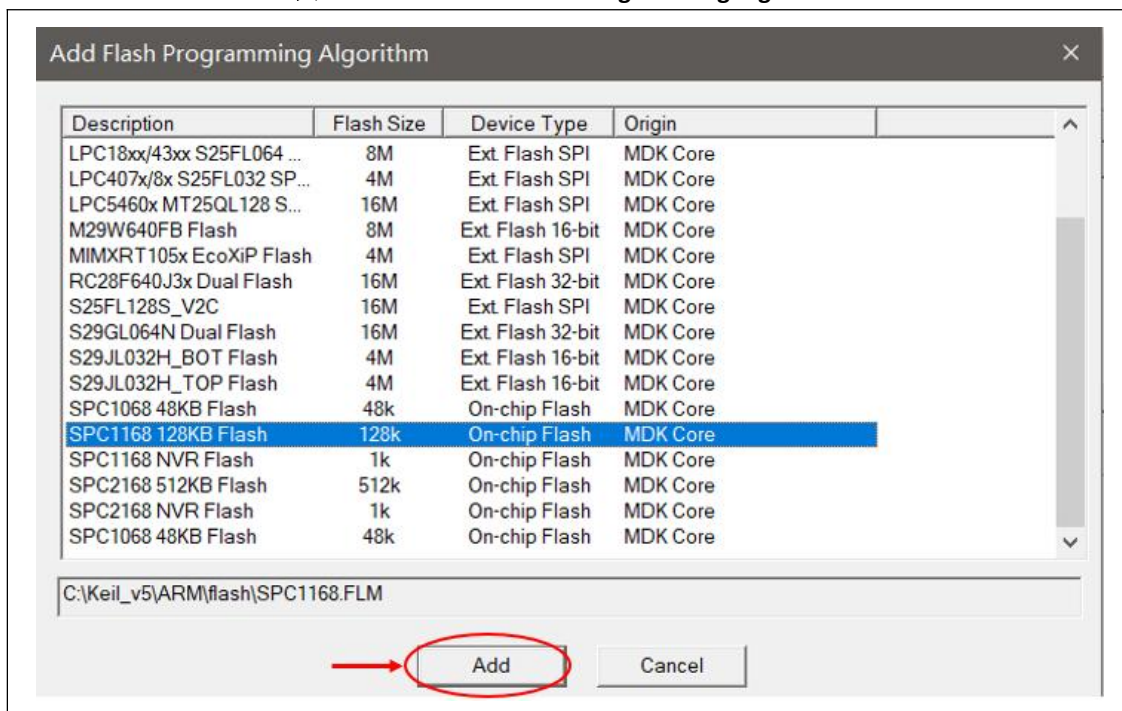


图 3-16: Add Flash Programming Algorithm

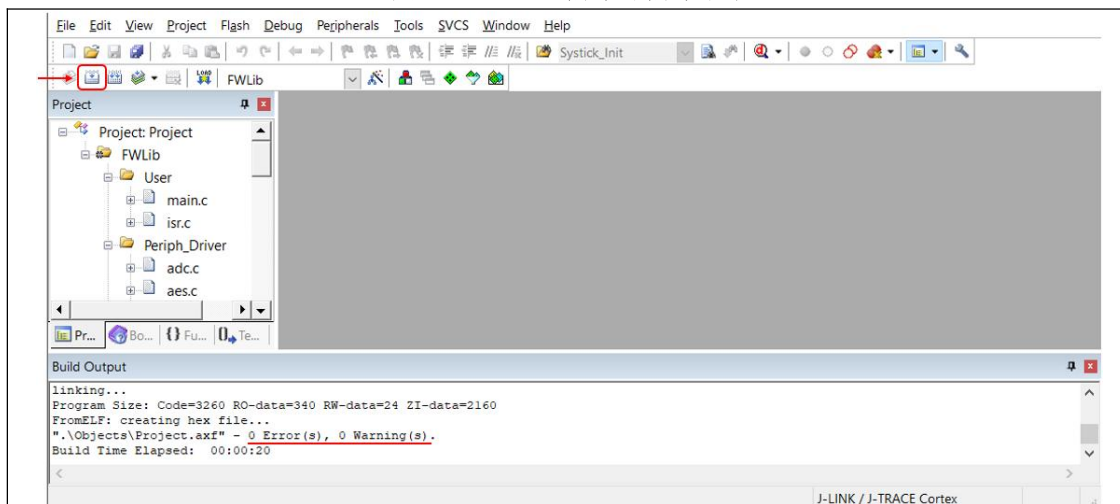


点击 Add 按钮后，会返回到 Cortex Jlink/JTrace Target Driver Setup 界面，如图 3-17 所示，可见 SPC1168 算法已被加入。将实际使用芯片的 FLASH 和 RAM 的位置和大小，填入红色矩形框后，点击确定。

Step5

执行完以上步骤之后，即可正式进入调试阶段。
首先编译代码。

图 3-19: 编译与编译结果



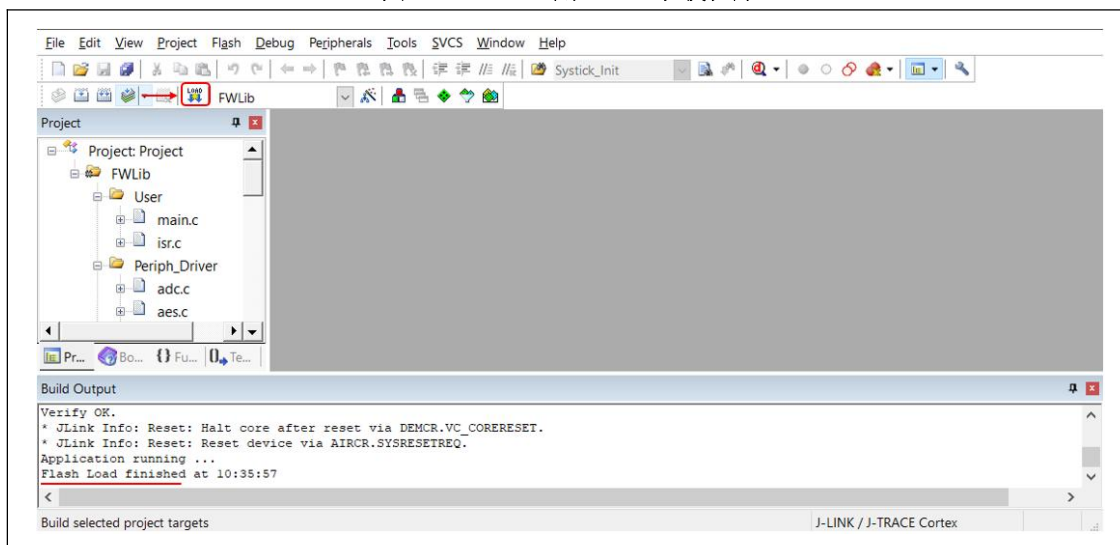
注:

编译(compile)后，Keil 的编译程序会在当前工程目录下产生大量的 **object file**，工程文件夹的大小可能从数十 KB 变成数十 MB，除了占据硬盘空间外，也难以在电邮内夹带，若想解决该问题，可执行当前工程目录下的 **del.bat** 脚本，删除多余的 **obj** 文件。

Step6

按下 Keil tool bar 上的 Download 键。

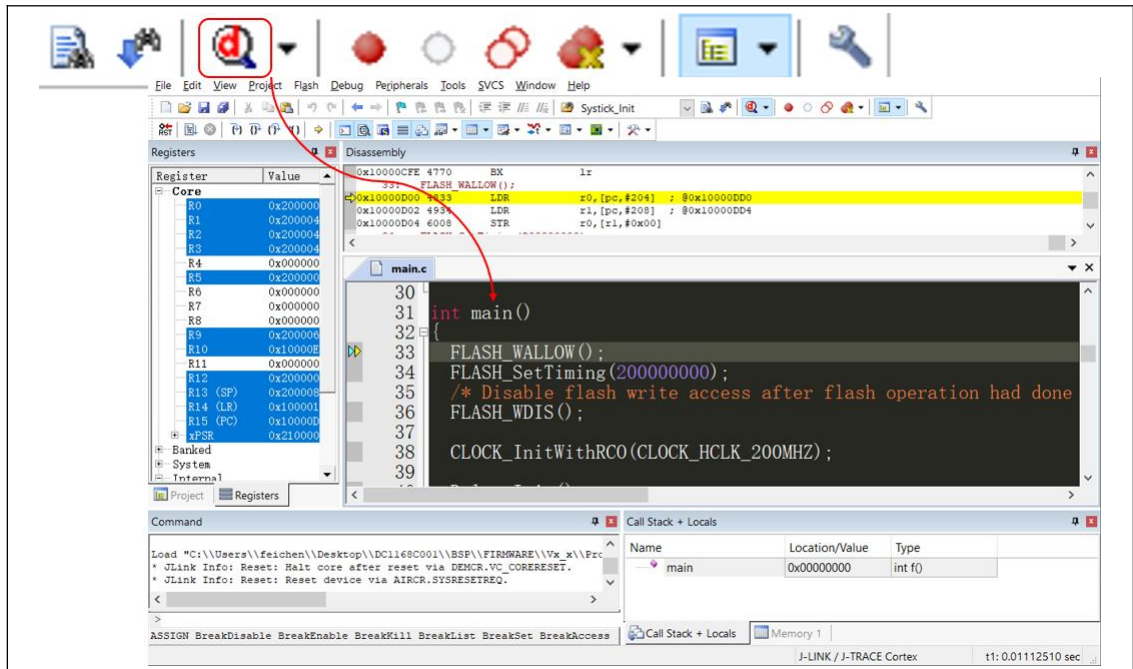
图 3-20: 用 JLINK 下载程序



Step7

按下 Debug 键，则可以进入 Debug 模式开始调试。

图 3-21: Debug window



注：若成功进入 Debug 模式，但是却无法单步调试，请检查 BOOT 是否拨到 NORMAL 模式（BOOT 拉高），若拉到地，芯片进入 UART 串口模式，会造成 Debug 异常。

3.4 使用 ULINK 进行开发及调试

常用的调试工具除了 JLINK 之外，还有一款叫做 ULINK 的工具，其调试程序的方法与 JLINK 类似。本节只简单描述使用 SWD 调试时，ULINK 与 JLINK 不同的地方：

- (1) 一般 ULINK2 不带有源 3.3V，请使用外部 3.3V 电源为开发板供电；
- (2) 如图 3-10 所示，GPIO38/39 可以用作 SWD 接口的 TMS (Test Mode State Pin)/TCLK (Test Clock Pin)。由于 3.3V 由外部电源提供，所以只需要把 GND，SWCLK，SWDIO 三条线接好即可；
- (3) Debug 配置界面，请选择使用 ULINK2/ME Cortex Debugger 选项，如图 3-22 所示；点击 Settings 按钮后，会弹出与 ULINK2 相关的设置，如图 3-23 所示。可以看到，蓝色椭圆形框中出现相应设备号及名称，只有当 SW Device 显示芯片 IDCODE 时才能算是连接成功。

图 3-22: ULINK2 下 Debug 配置界面

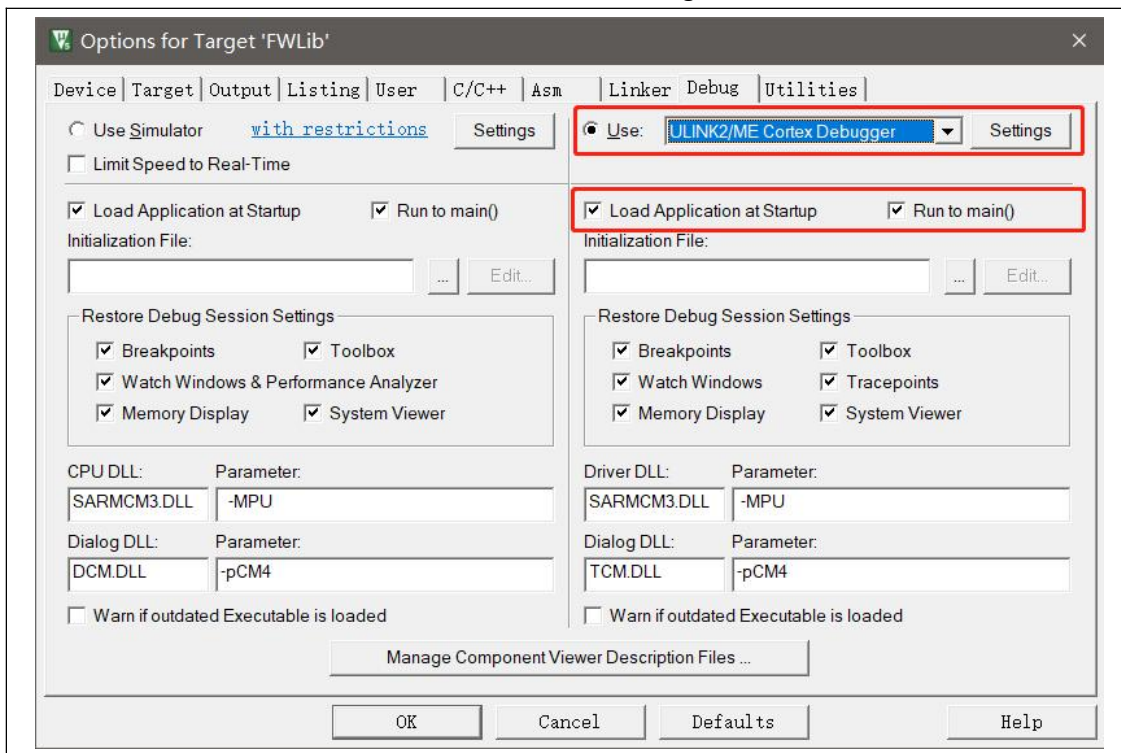
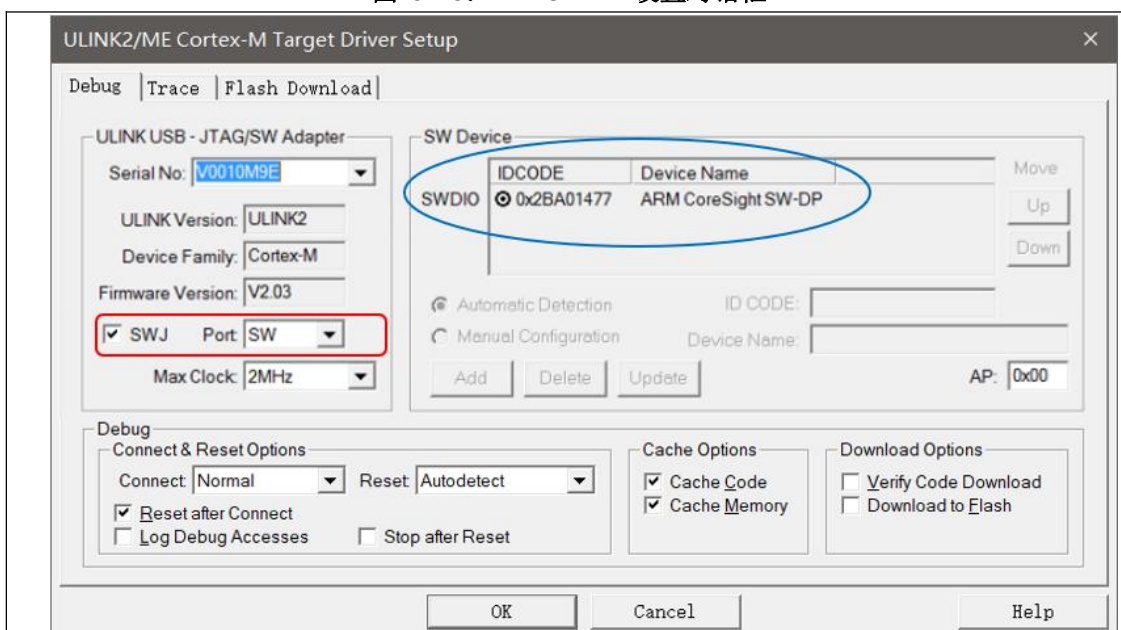


图 3-23: ULINK2 设置对话框



关于 ULINK2 的其它调试步骤，与 JLINK 相同，请参阅章节 3.3。

3.5 使用 UART 进行开发及调试

SPC1168 可通过串口(UART)将程序烧写入芯片的 Flash，当 BOOT(GPIO40)脚为低电平时，搭配 Spintrol 提供的 ISP Tool，可将代码以加密或不加密的形式烧写入芯片中。ISP Tool 中也包含常用串口终端与常用小工具，方便使用者通过串口进行开发与调试。

Step1

打开 TOOLS\Download_tool 文件夹下的 ISP Tool 工具。

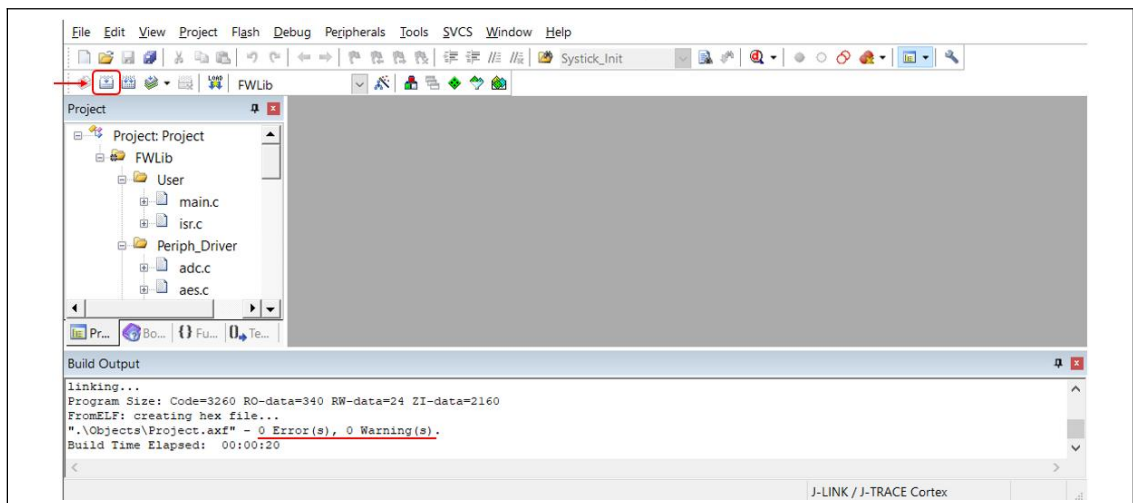
图 3-24: ISP download tool



Step2

打开“GPIO_Level_Test”示例工程，并进行编译。

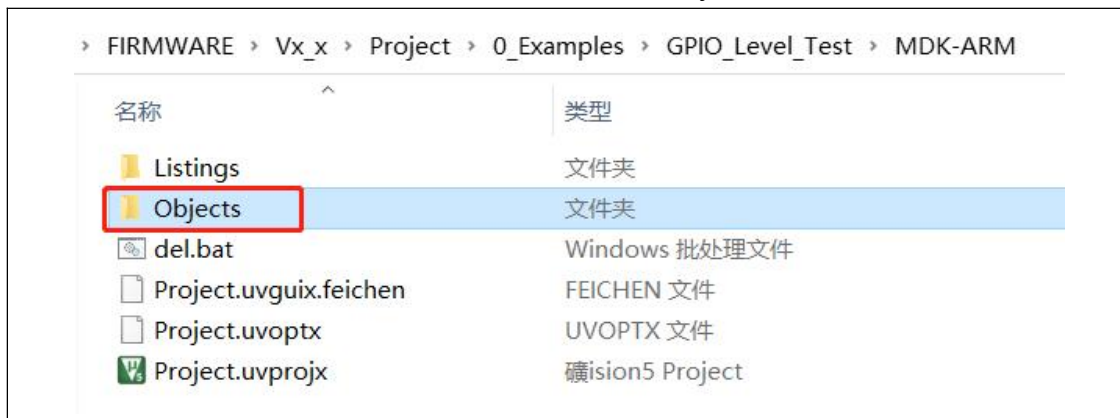
图 3-25: 编译工程



Step3

编译成功后，可以发现项目内多了个文件夹。

图 3-26: 编译后产生新的 Objects 文件夹

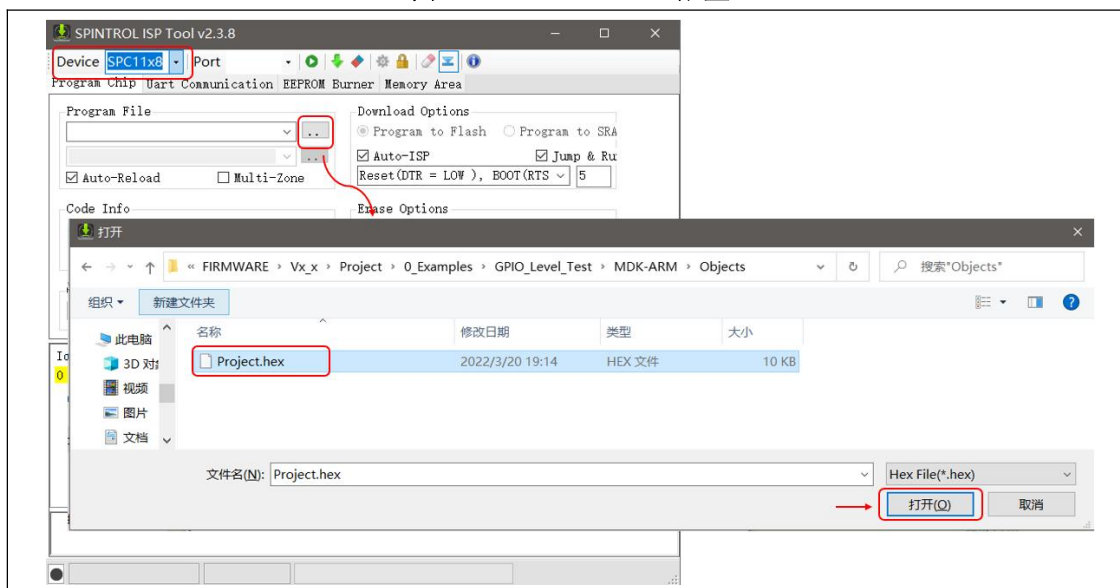


Step4

开启 ISP Tool，Device 选项请选择 SPC11x8；Program File 请选择位于项目文件夹下的 Objects 文件夹下的 hex 文件。以 GPIO_Level_Test 为例，hex 文件存放于如下位置：

FIRMWARE\Vx_x\Project\0_Examples\GPIO_Level_Test\MDK-ARM\Objects。

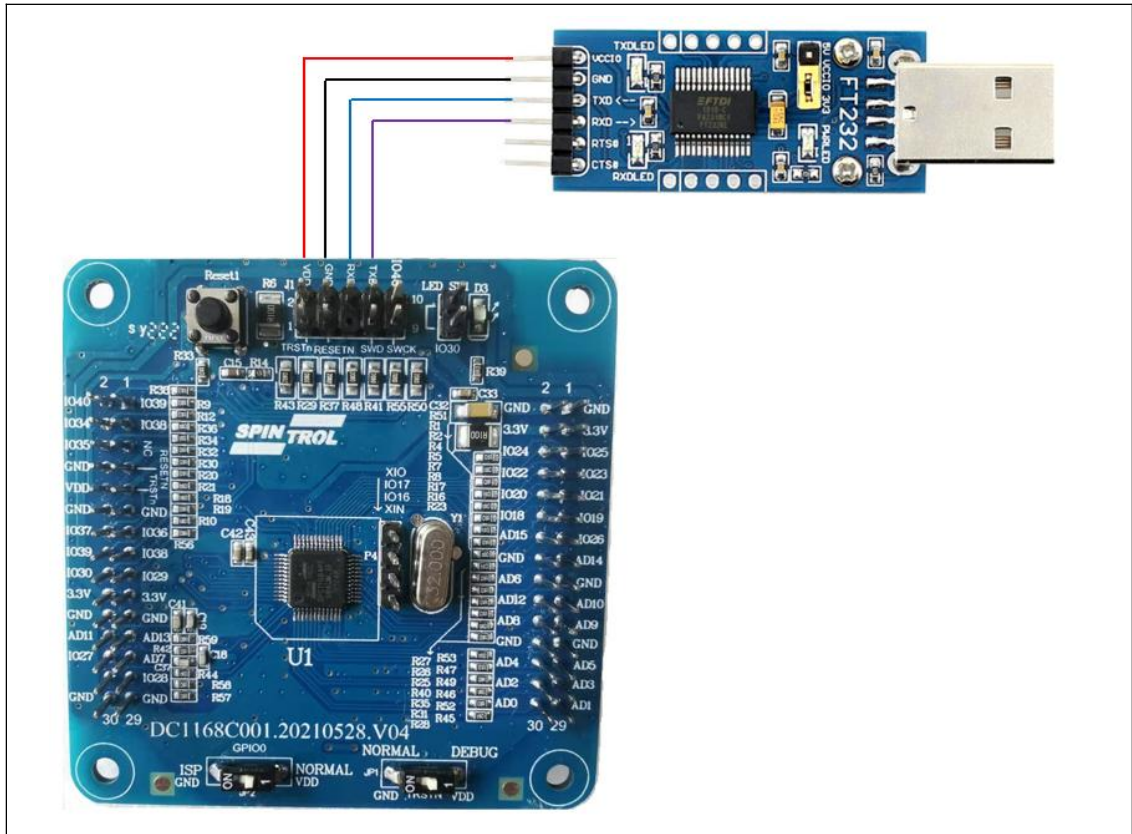
图 3-27: ISP Tool 配置



Step5

将 FT232 串口工具与开发板连接。VCCIO 用于输出 3.3V 或 5V（模块由 USB 供电，针对本开发板须将跳线帽跳到 3.3V），接开发板的 VDD；GND 接开发板的 GND；TXD 接开发板 MCU 的 RXD；RXD 接开发板 MCU 的 TXD。FT232 另一端接入电脑。

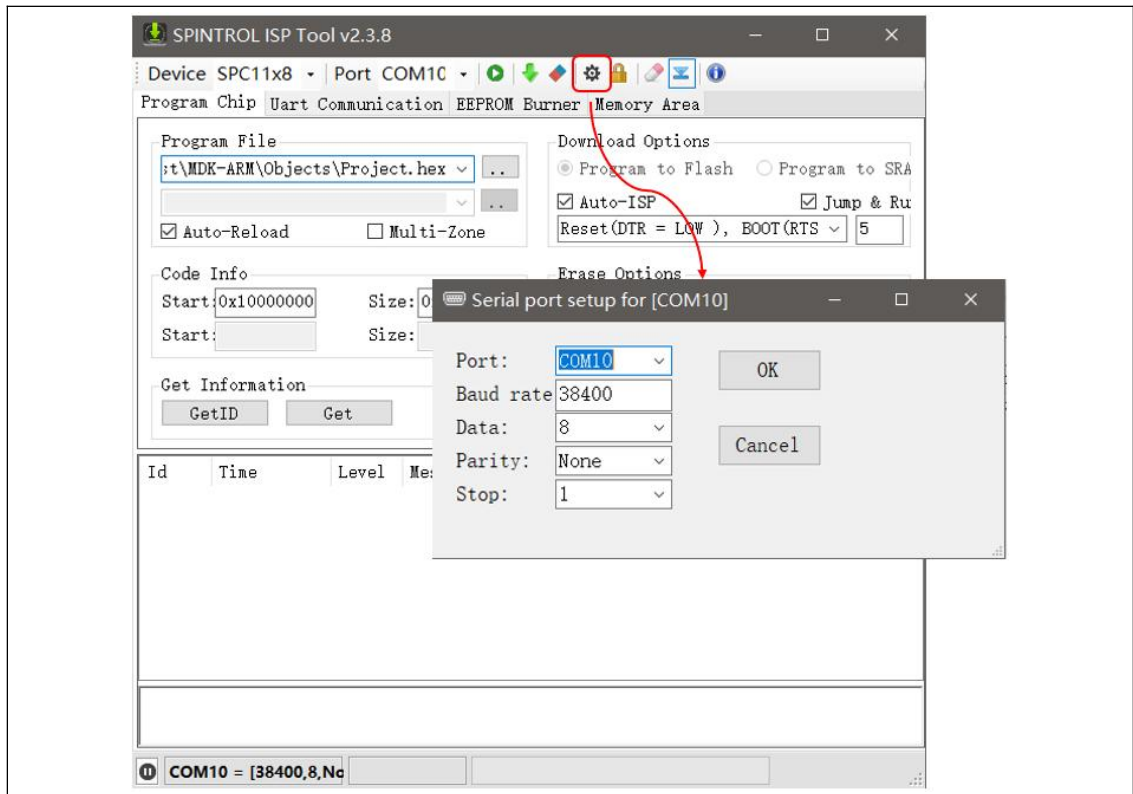
图 3-28: FT232 与开发板连接方式



Step6

选择对应的端口 Port 与速率，SPC1168 可自动适应不同串口速率，因为范例代码中 UART 速率为 38400，因此这里将速率也设为 38400，以便观察程序下载后打印的数据。

图 3-29: UART 波特率配置



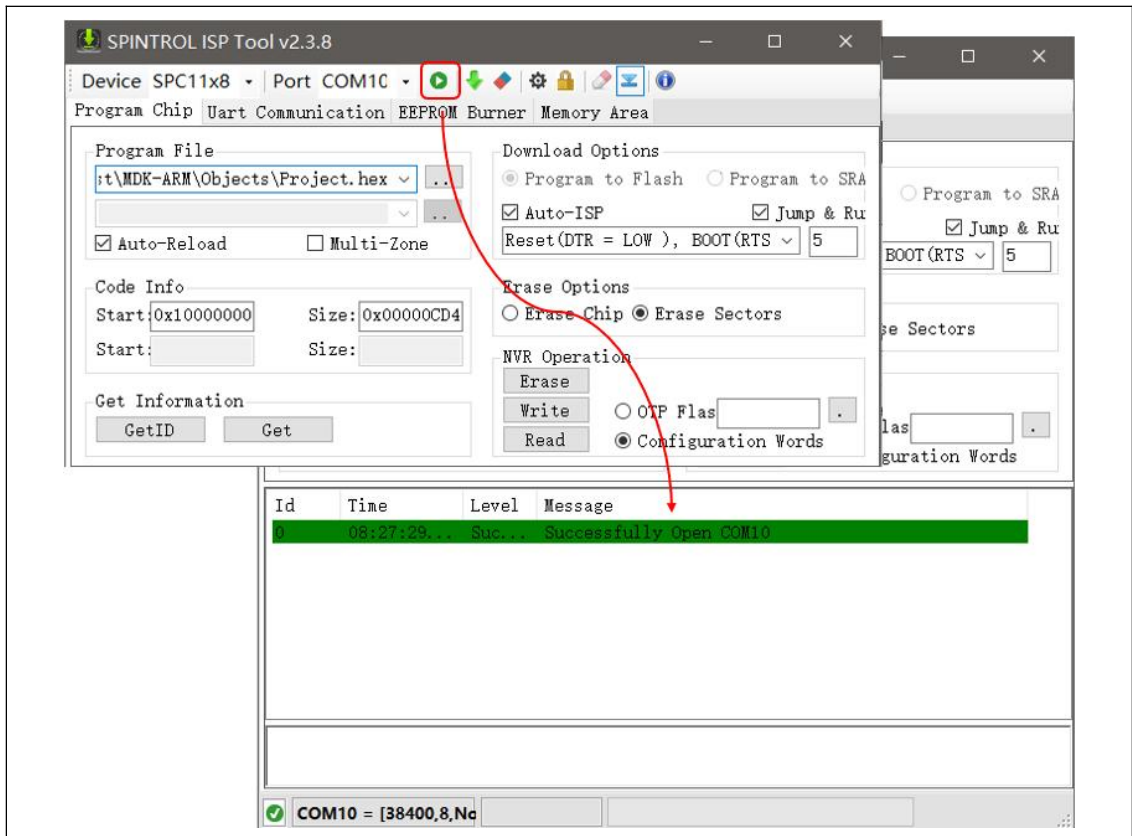
注:

- (1) 若无法识别出 Port, 请检查或更换 FT232 模块。
- (2) Download tool 会自动识别 Port 端口。若端口太多, 不清楚 Port 编号, 请在电脑设备管理器进行查询。

Step7

开启串口 COM port。

图 3-30: 打开 PC 串口



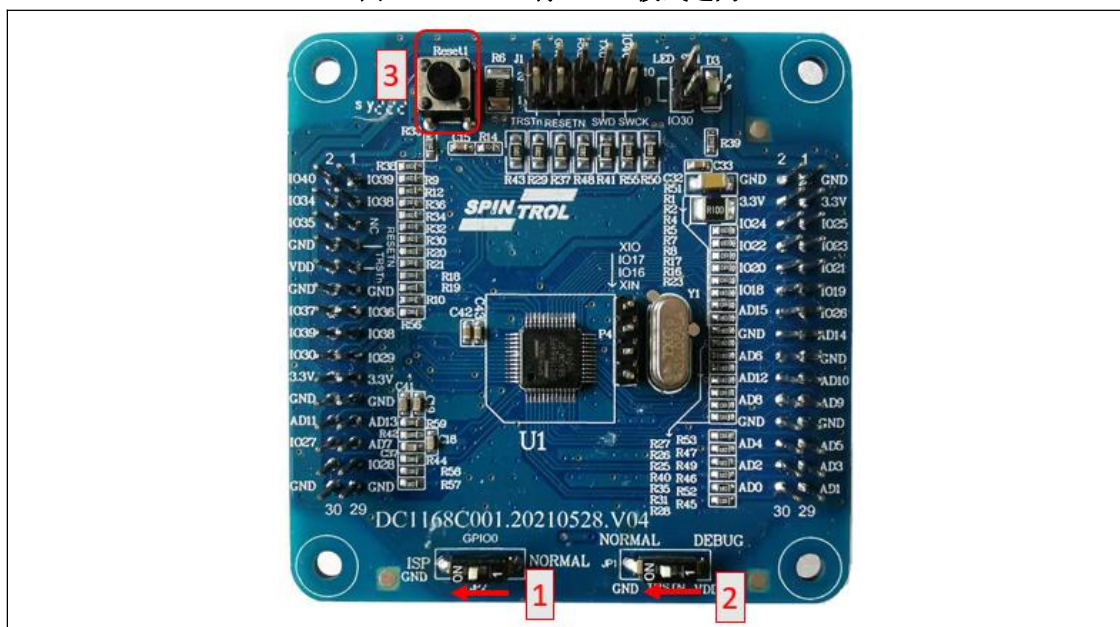
Step8

将开发板上的 BOOT(GPIO40)管脚选项拨至 ISP，会将芯片上的 BOOT 拉低，即选择使用 UART 接口对 Flash 存储器进行重新编程。

将开发板上的 TRSTn 管脚选项拨至 NORMAL，会将芯片上的 TRSTn 拉低，禁止芯片的 Debug 功能。需要注意的是，当 BOOT 管脚为低电平时，必须保持 TRSTn 管脚为低电平，否则芯片将进入工程测试模式。换句话说，在使用 UART 进行开发及调试时，TRSTn 保持为低即可，不需改动。

按下 Reset 键。

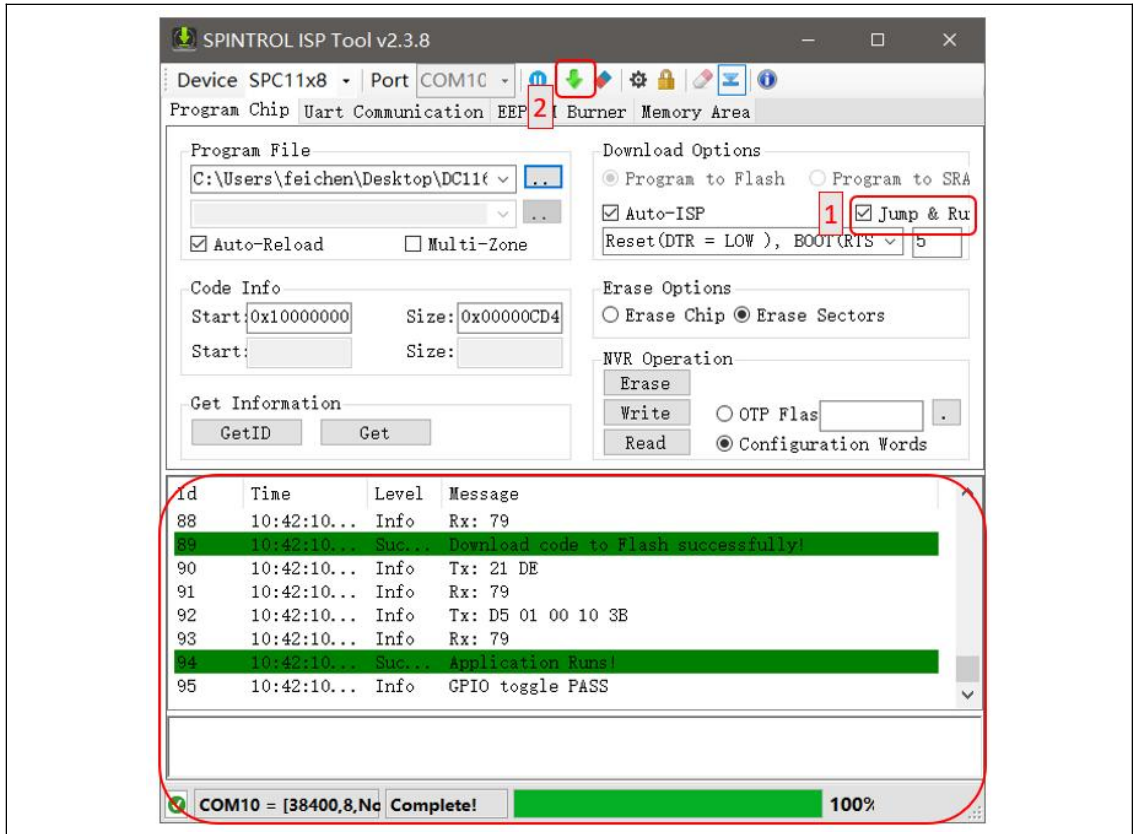
图 3-31: 将 BOOT 模式选为 ISP



Step9

回到计算机端，按下 ISP Tool 的 Download 按钮。ISP 工具还提供了 Jump & Run 的功能。如果勾选该功能，在程序下载完成之后，Boot Loader 会直接跳转到程序的开始处并执行程序；如果未勾选该功能，ISP 工具只是将程序下载到 Flash 或者 SRAM 中。

图 3-32: 开始下载程序

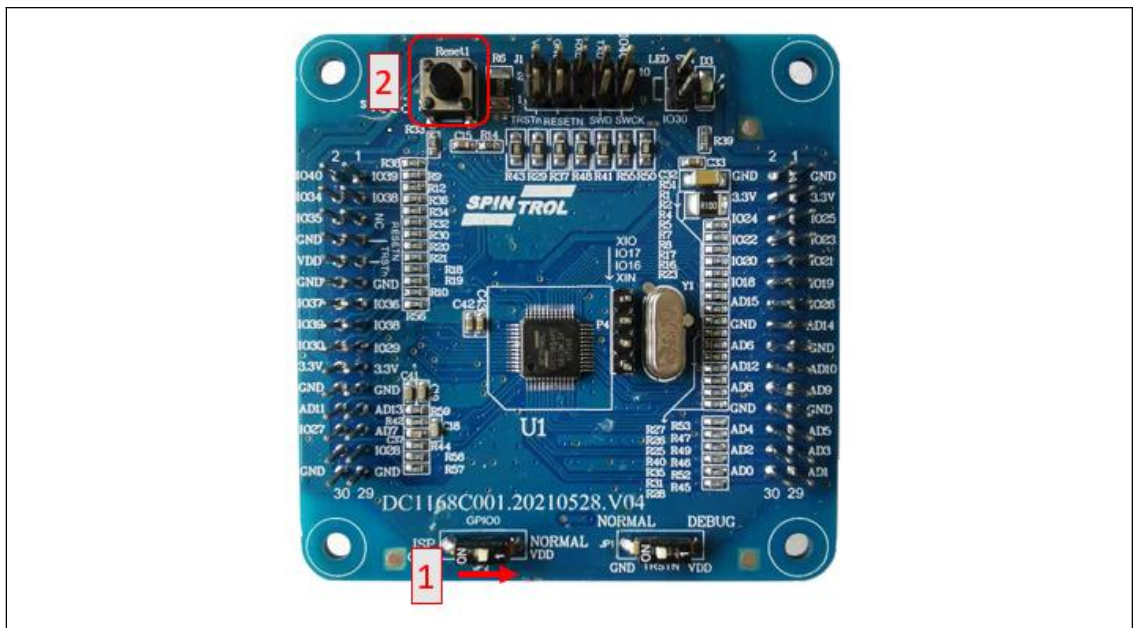


Step10

烧写成功后，程序开始运行。

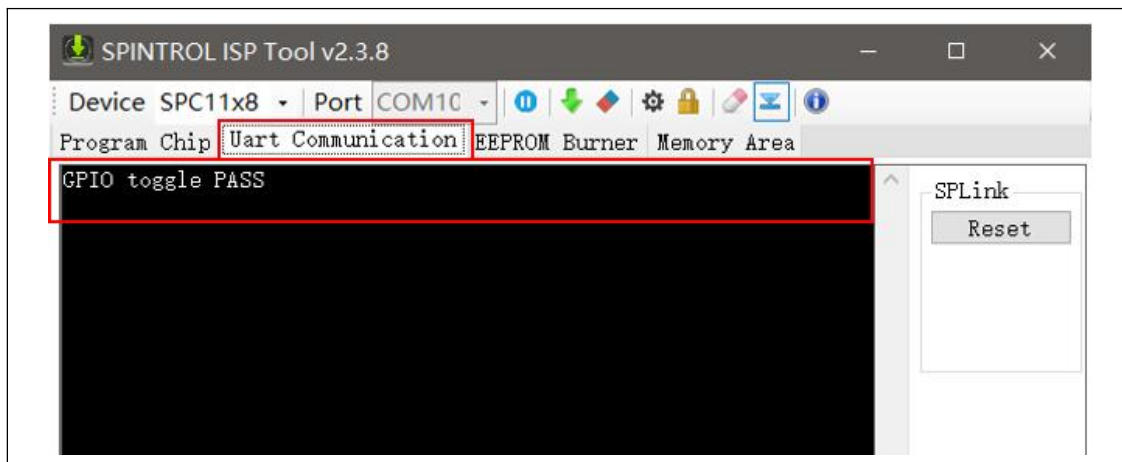
如果未勾选 Jump & Run 功能，需将 BOOT 开关选项拨回 NORMAL，并且按下 Reset 键，程序方可运行。

图 3-33: BOOT 模式改为 NORMAL



打开串口终端接口，可看到打印的信息，说明程序已经在运行。

图 3-34: ISP Tool UART Communication



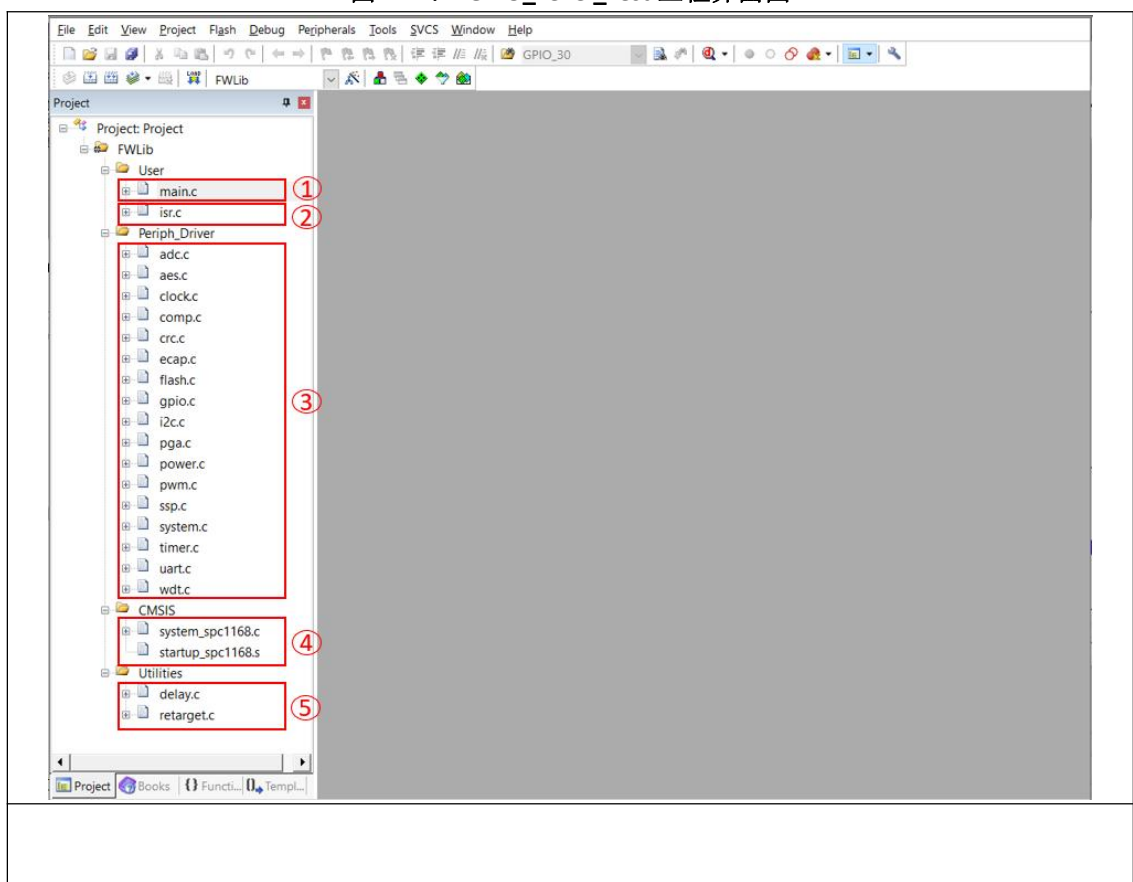
4 进阶说明

4.1 Demo 程序进阶说明

前面章节已经对 Demo 的大体结构分布做了介绍，此章节就“GPIO_Level_Test”示例程序做更进一步的说明。

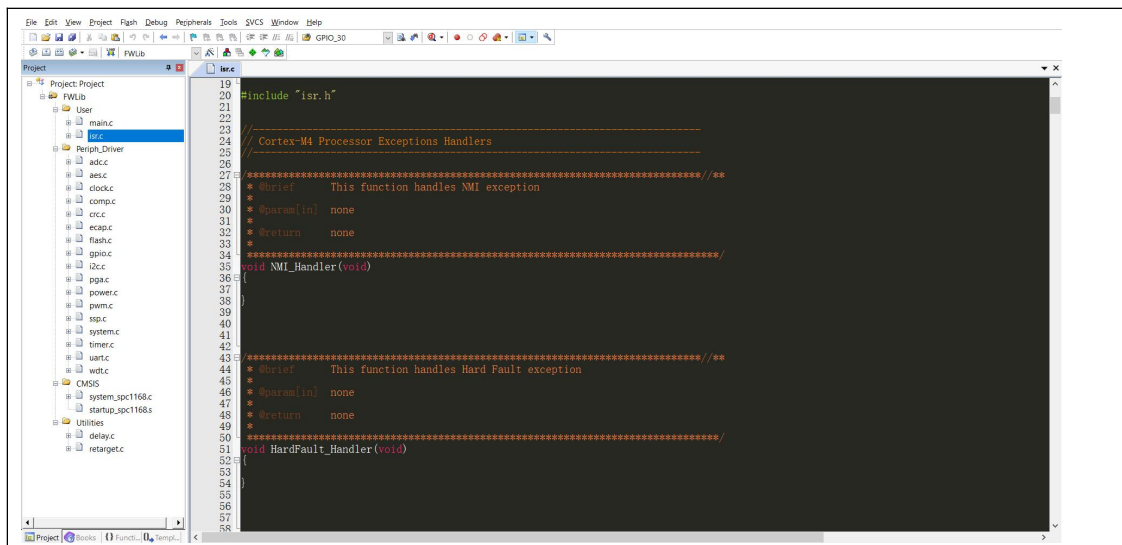
打开“GPIO_Level_Test”工程之后，能看到如图 4-1 所示的界面。

图 4-1: GPIO_Level_Test 工程界面图



- ①是程序的主体部分。
- ②实现了所有中断函数，但函数体全部为空，如图 4-2 所示，这些函数的入口地址在④中的启动文件（.s 文件）中定义。若使用者在 main.c 中实现了中断函数，那么就需要将②中的相应函数注释，否则编译时会出现重定义的错误。

图 4-2: ISR 文件图



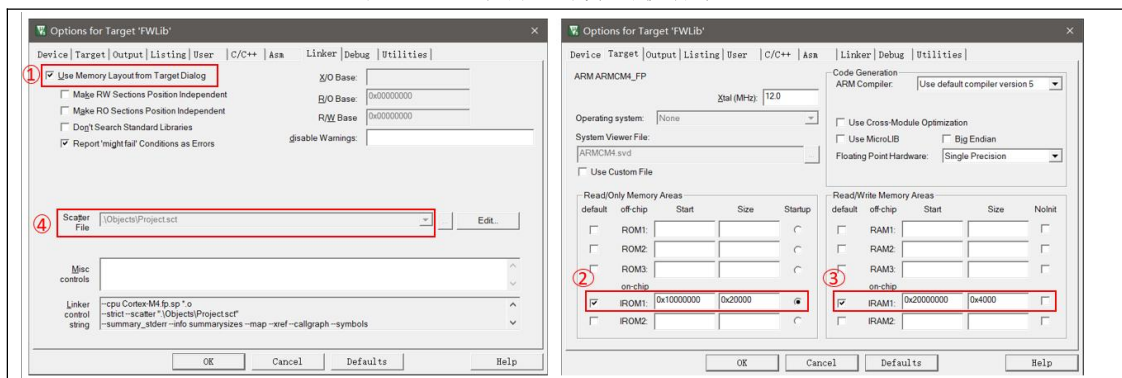
- ③是所有外设的驱动接口文件。
- ④是芯片启动文件，以及 ARM 核特殊函数接口文件。
- ⑤是延时函数接口，以及输出重定向文件。

4.2 Keil IDE 进阶说明

此小节将介绍一些日常开发过程中使用频率非常高的 Keil IDE 使用方法，供开发者使用。

- 程序加载域与执行域。程序的加载域就是程序运行前在 flash 中具体分区情况，执行域就是程序运行后，程序在 flash 和 ram 中的分区情况。如图 4-3 所示。若①被选中，则编译器根据②③的设置值自动处理加载域与执行域的地址范围。若①没有被选中，则采用④分散文件中设置的加载域与执行域地址范围。

图 4-3: 程序加载域与执行域配置



- 编译搜索路径设定。

图 4-4: 编译时搜索路径

