



# PineconePi ONE

## 入门指南

日期	作者	内容	备注
2019.5.21	小董	完成文档架构	

# 目录

## 一, 关于松果派 ONE

### 1.什么是松果派 ONE?

### 2.松果派 ONE 可以干什么？

## 二, 开发环境配置

### 1.安装 KEIL PACK

### 2.新建工程

### 3.SVD 辅助调试

## 三, 烧录软件使用

### 1.ISP 烧录软件使用

### 2.J-link 烧录软件使用

## 四, 点亮一盏 LED 灯

## 五, 外设寄存器查看修改软件使用

## 六, HardFault 分析软件使用

## 七, 尝试使用 MicroPython 进行开发 (新手推荐)

# 一，关于松果派 ONE

## 1.什么是松果派 ONE?

松果派 ONE 开启你的极客之心：搭载华芯微特 SWM320 处理器；低成本，小体积，尺寸仅 48 mm x31 mm；引脚全引出；面包板直插；支持 MicroPython;ARM-Cortex-M4 内核；硬件单周期乘法；512KB FLASH,128KB SRAM；外设交叉映射；最高可支持分辨率 1024\*768TFT-LCD 驱动；丰富的硬件资源：1 组 32 位看门狗定时器，6 组 32 位通用定时器，1 组 32 位专用脉冲宽度测量定时器，12 通道 16 位 PWM 发生器，2 个 8 通道 12 位，IMSPS 的逐次逼近型 ADC 模块；板载 Ch330N。丰富的芯片资料；多扩展坞支持。

## 2.松果派 ONE 可以干什么？

一块入门 ARM 单片机的神奇开发板；

3D 打印机控制板；

激光雕刻机控制板；

写字机控制板；

一款四轴飞行器飞控；

一款 NES 掌上游戏机；


# 二，开发环境配置

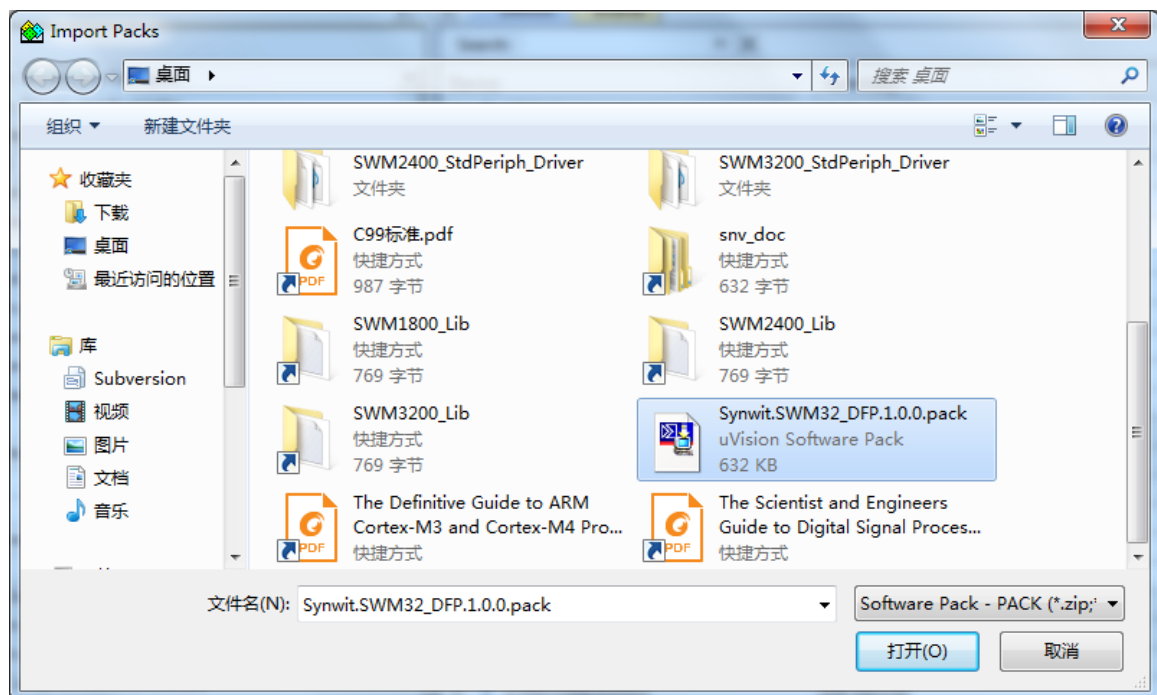
## 1.安装 KEIL PACK

Keil 从 5.0 版本之后通过安装对应单片机的 pack 提供对芯片的支持,安装 Synwit.SWM32\_DFP.1.0.0.pack 后在 Keil 5 中新建工程可以直接选择 Synwit SWM 系列芯片，且软件会根据所选型号自动正确设

置 Flash/RAM 大小、程序烧写算法、SVD 仿真文件等内容

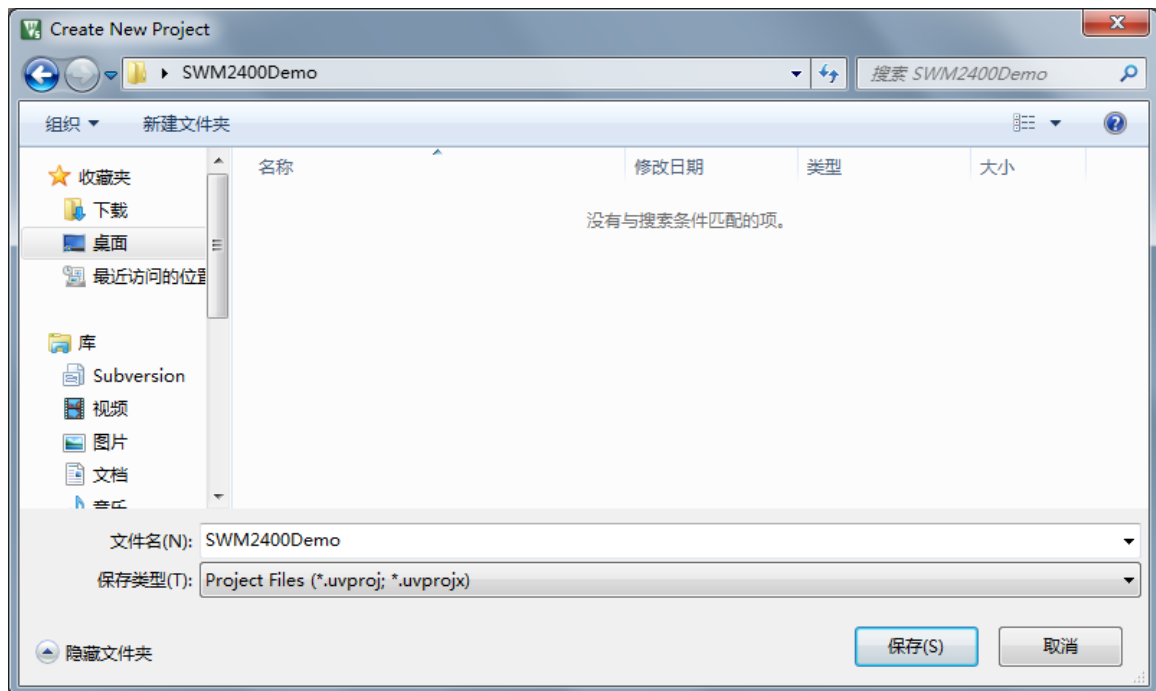
## 安装 Synwit.SWM32\_DFP.1.0.0.pack

点击工具栏“Pack Installer”按钮, 在弹出窗口中通过菜单“File→Import..”选中 Synwit.SWM32\_DFP.1.0.0.pack, 点击“打开”按钮安装

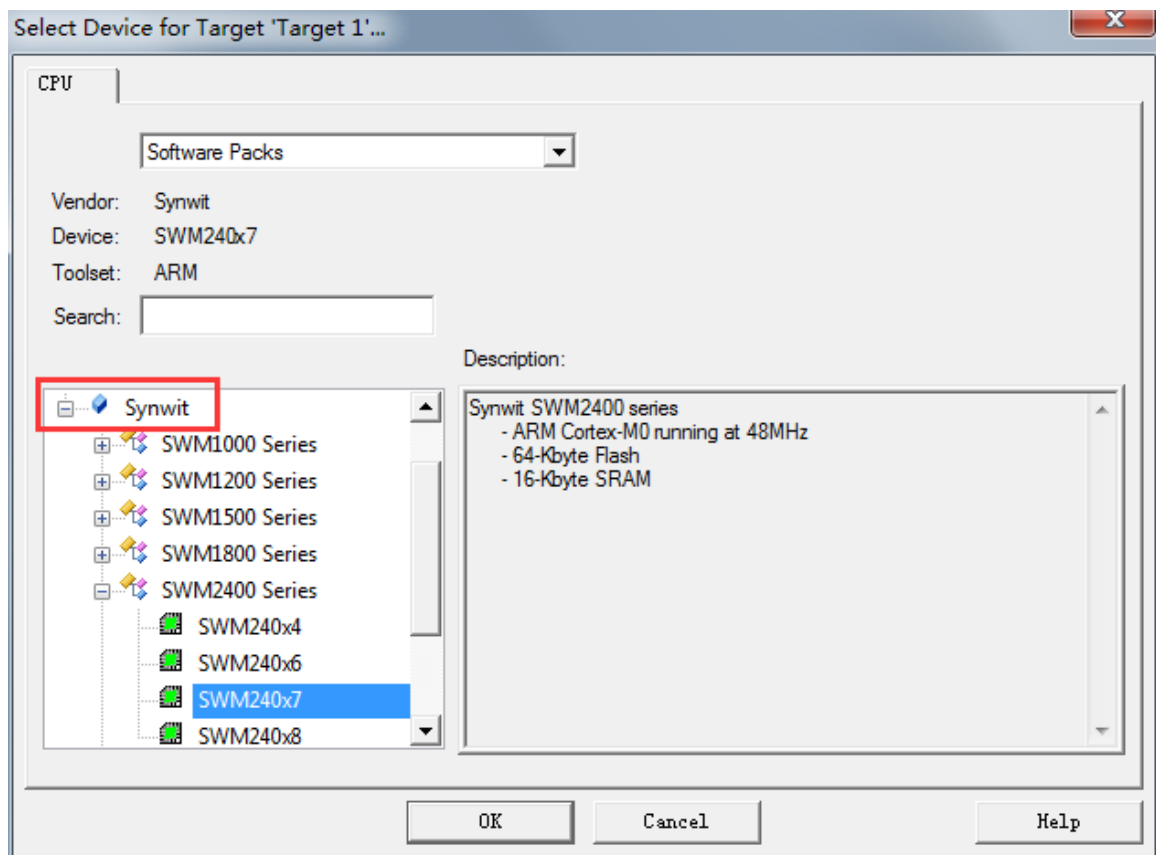


## 2.新建工程

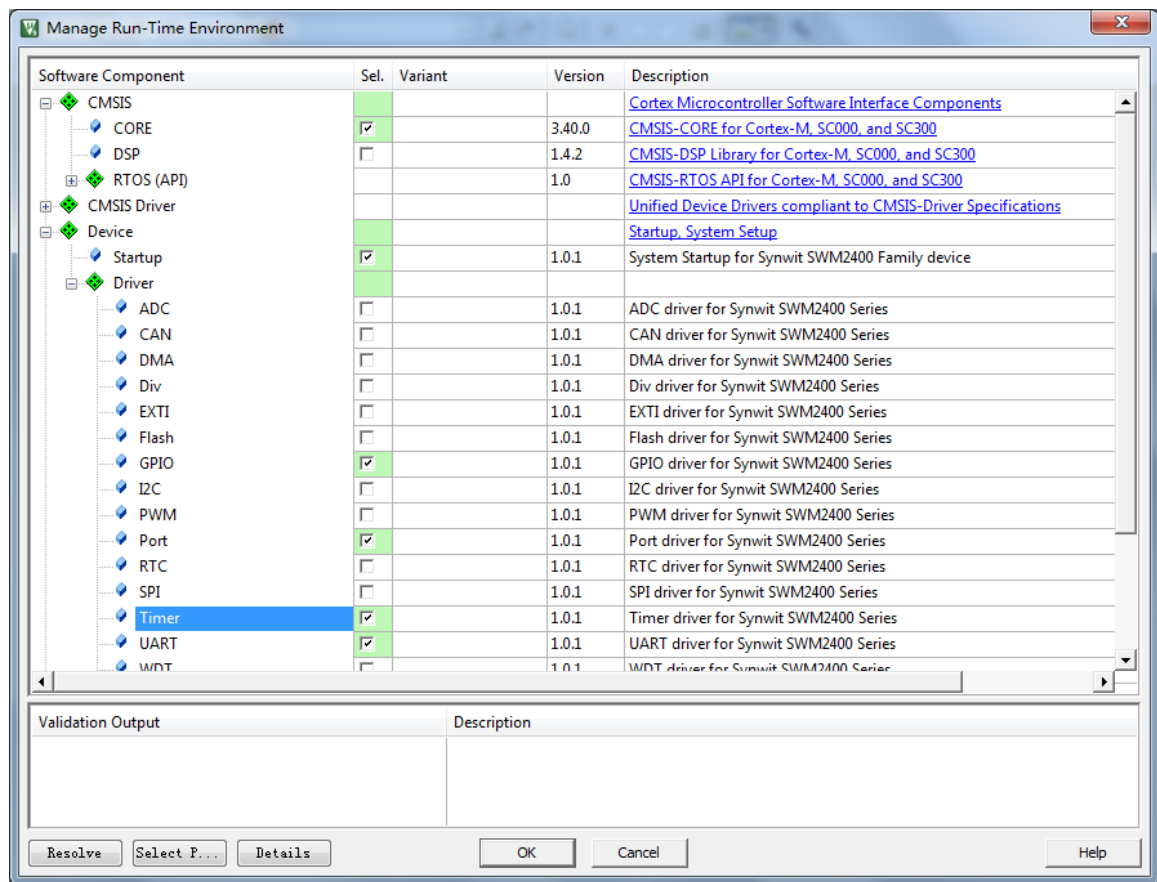
打开 Keil 5 软件, 选择菜单“Project→New uVision Project..”



点击“保存”后弹出器件选择窗口, 选择 Synwit 下的 SWM240x7



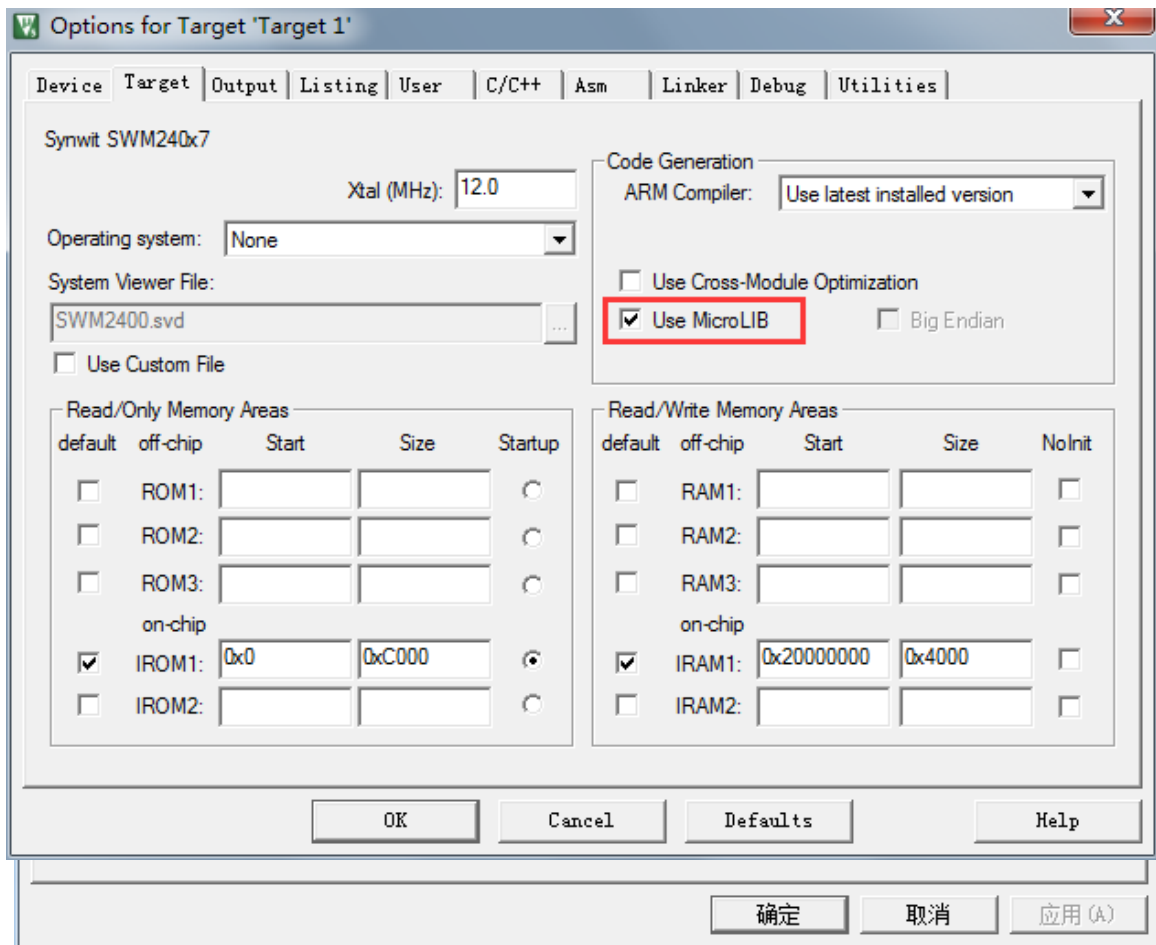
点击“OK”，在弹出的“运行时环境管理”窗口中选择需要的文件



其中 CORE 和 Startup、Port 是必选的

点击“OK”，完成项目创建和配置；但仍有一些设置需要手动勾选

## 1、打开“Option”窗口手动勾选上“Use MicroLib”

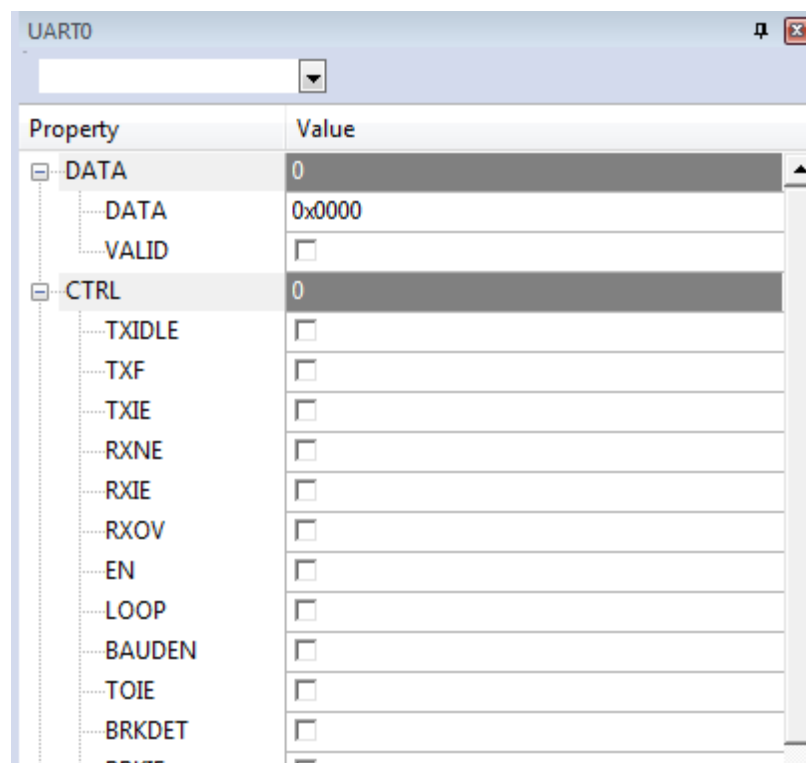
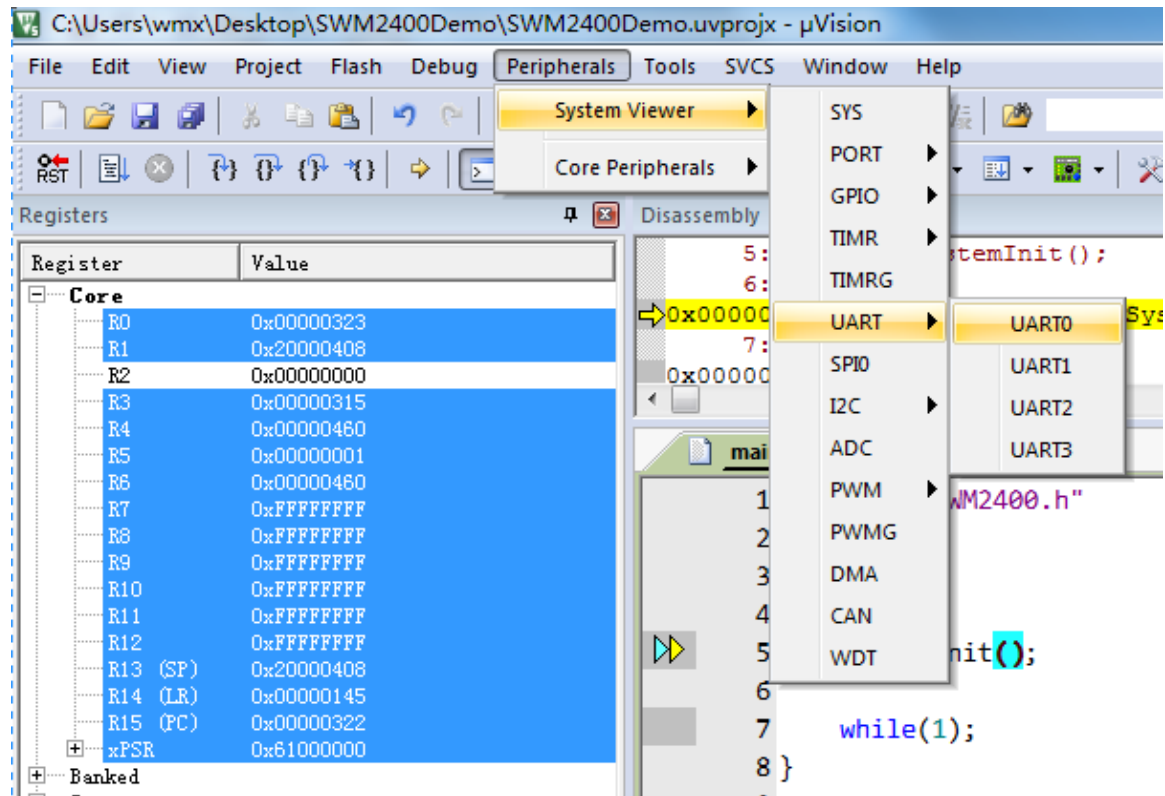


## 2、勾选“Reset and Run”



### 3.SVD 辅助调试

Synwit.SWM32\_DFP.1.0.0.pack 中包含有 SVD 文件, 可以在调试中按位域显示外设的状态, 并操作外设



## 三，烧录软件使用

### 1.ISP 烧录软件使用

功能说明：

- 1、握手后自动烧录：点击握手后自动执行一键下载。
- 2、生产模式：下载成功后自动执行握手操作，直到握手成功。
- 3、全部擦除：擦除整片 Flash。
- 4、扇区擦除：擦除所选程序大小所需 Flash。
- 5、一键下载：自动执行扇区擦除和程序下载操作。

其他说明：

- 1、加载 bin 文件过程中尽量不要进行其他操作。
- 2、为保证稳定性，建议用户按照界面提示的步骤顺序执行。

操作流程：

- 1、芯片断电情况下连接下载接口：

SWM180 -- RX-A0 、TX-A1

SWM240 -- RX-B11、TX-B12

SWM320 -- RX-A2 、TX-A3

SWM400 -- RX-A6 、TX-A5

- 2、把 BOOT 引脚拉高。

SWM180 -- BOOT-PB0

SWM240 -- BOOT-PD0

SWM320 -- BOOT-PB0

SWM400 -- BOOT-PB0

3、上电、执行相应操作。

下面让我们开始烧录程序吧！

第一步 :打开“SYNWIT\_ISP\_V2.2.3.exe”, 选择型号 SWM320VET7-50, 点击“确认”。



第二步 :选择文件, 并点击打开；

第三步 :选择操作模式 (一般情况下, 勾选“握手后自动下载”)。

第四步 :串口操作, 选择串口号和波特率, 并点击“打开串口”按钮。

第五步 :下载操作点击“一键下载”

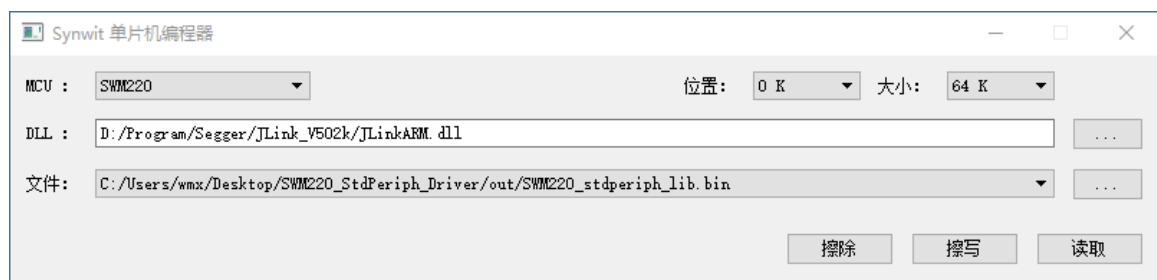
第六步 :等待下方下载提示框提升“下载成功”和进度条变为 100%。



## 2.J-Link 烧录软件使用

通过 JLINK 将\*.bin 或\*.hex 文件烧录到芯片指定位置

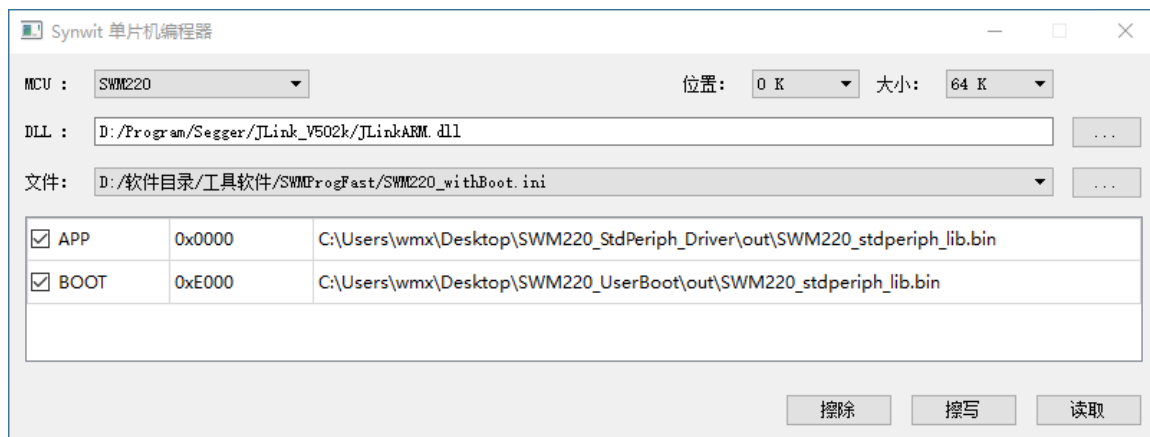
### 单文件烧录



位置：擦除、擦写、读取操作的起始地址

大小：擦除、读取操作的大小，擦写操作的大小由文件大小决定

## 多文件烧录



当指定的文件以“.ini”为后缀时，程序变成上图所示的多文件烧录界面

可双击表格中文件路径替换文件、修改“.ini”文件

## .ini 文件格式

[APP]

addr = 0x0000

path

C:\Users\wmx\Desktop\SWM220\_StdPeriph\_Driver\out\SWM220\_stdperiph\_lib.bin

[BOOT]

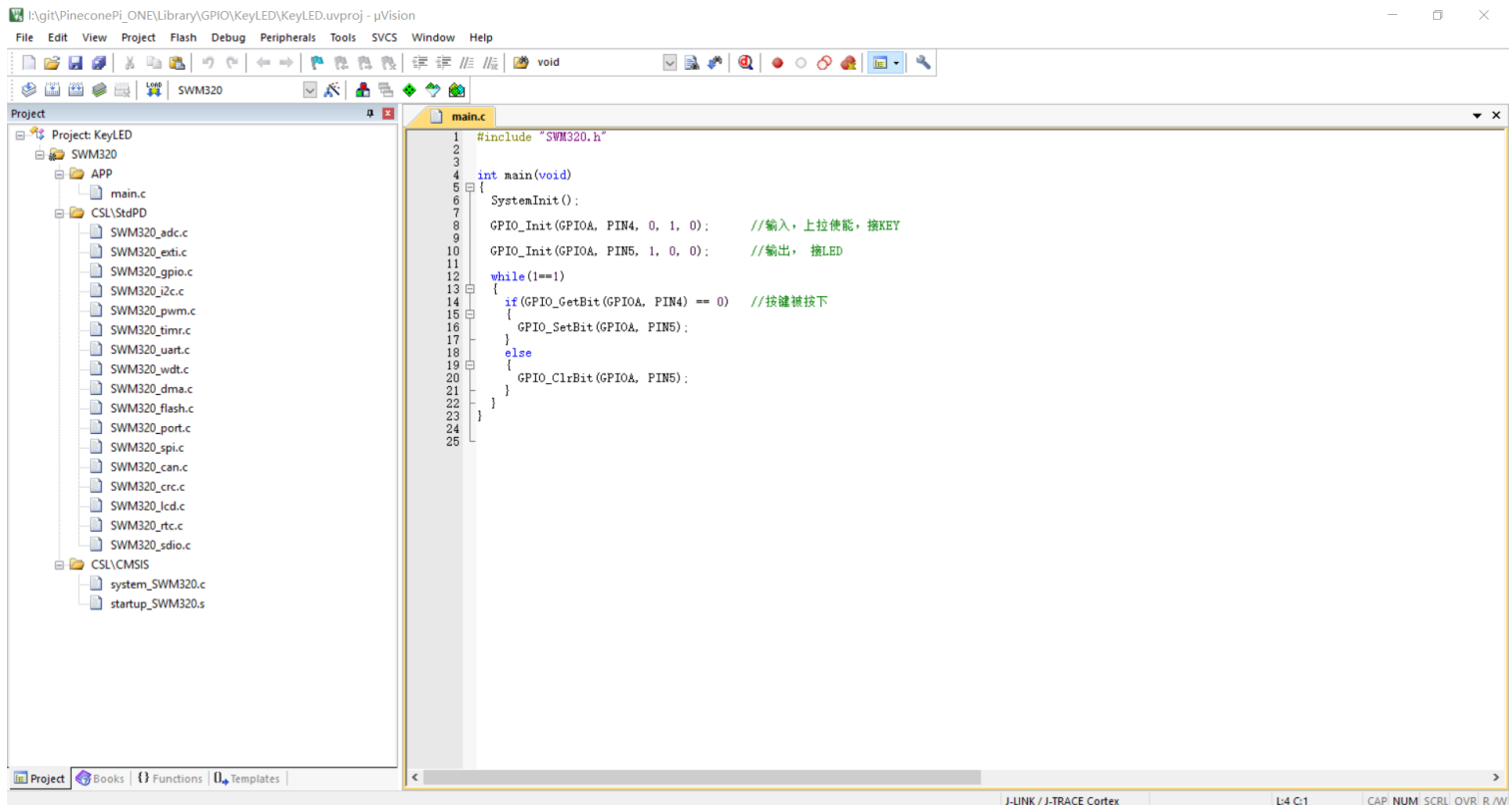
addr = 0xE000

path = C:\Users\wmx\Desktop\SWM220\_UserBoot\out\SWM220\_stdperiph\_lib.bin

注意：对于 SWM220，如果程序不是 UserBoot，请不要使用“BOOT”标签；因为给 SWM220 下载程序时，如果检测到有 BOOT 标签，会对程序做特殊处理

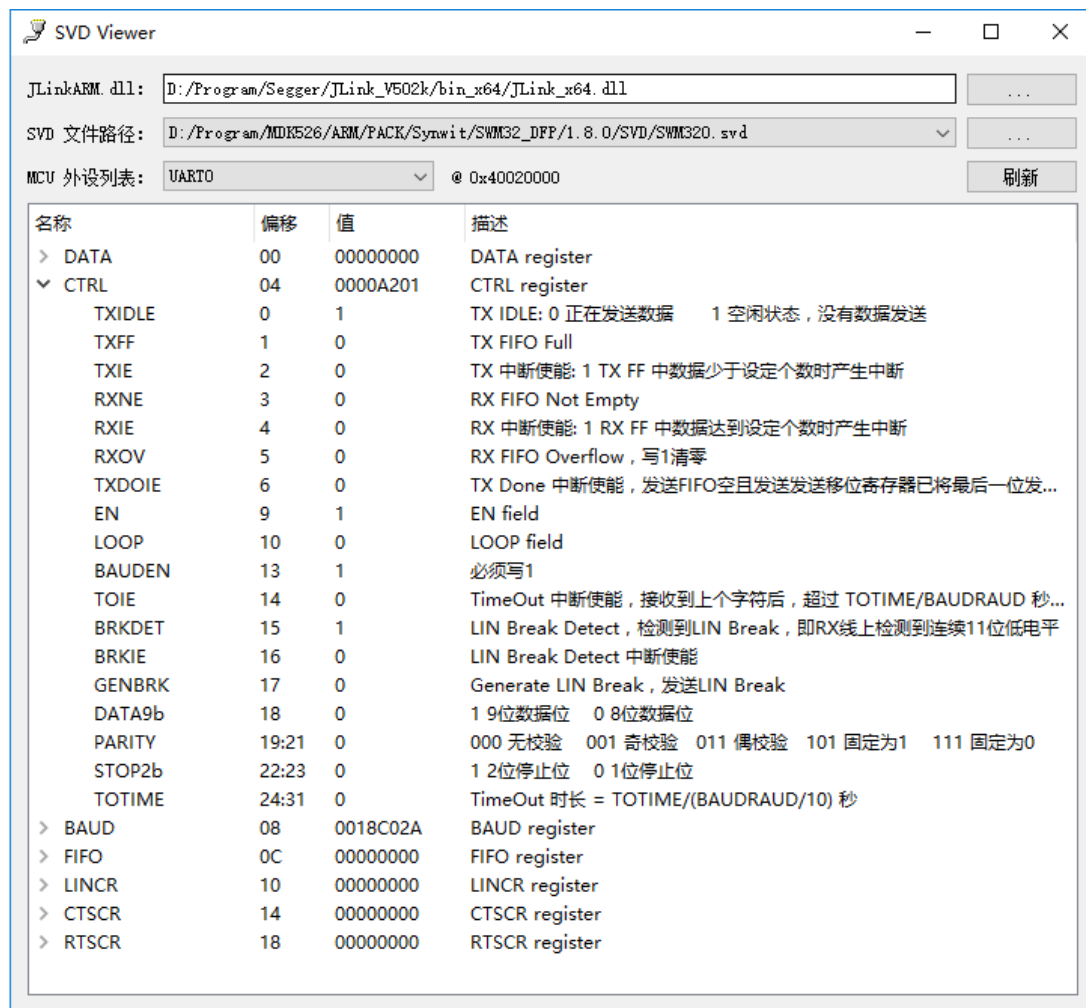
## 四，点亮一盏 LED 灯

打开“/Library/GPIO/KeyLED/”下的 KEIL 工程，根据工程定义完成 GPIO 输入输出管脚定义，即可完成点灯。



## 五，外设寄存器查看修改软件使用

概述：J-Link 连接芯片，此软件通过 J-Link 读取芯片外设寄存器，并根据 SVD 文件解析寄存器的位域定义，从而显示外设寄存器和位域的值，并且可以单击寄存器和位域的值进行实时修改，方便非调试状态下调试单片机外设



第一步、确保 J-Link 能连上芯片, 可使用 J-Link Commander 或 Keil 测试下连接

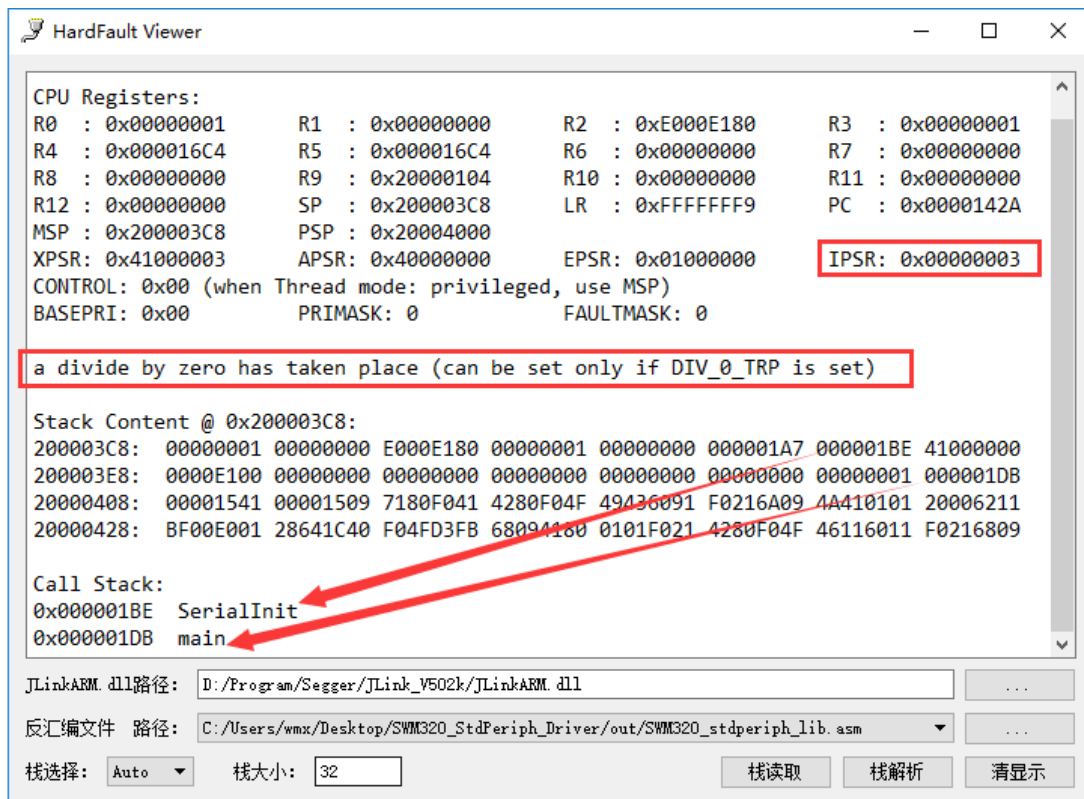
第二步、双击 SVDView.exe 运行, 设置正确的 JLink\_x64.dll 和器件 SVD 文件路径

第三步、从“外设列表”下拉框选择想要查看的外设

第四步、点击“刷新”按钮, 显示外设寄存器的当前值

第五步、单击寄存器/位域的值, 弹出对话框可修改寄存器/位域的值

## 六, HardFault 分析软件使用



发生 HardFault 后, 可用本软件通过 JLink 从芯片内存中读取栈中的内容, 然后根据栈内容和由反汇编文件中解析出的函数调用关系, 解析出 HardFault 发生的位置和执行路径。对于有故障原因寄存器的内核 (如 Cortex-M3、M4), 本软件还会分析故障原因寄存器、打印触发故障的原因

此外, 本软件还会将从反汇编文件中解析出的函数调用关系 (如函数 funcA 调用了哪些函数、哪些函数调用了函数 funcA) 保存到本软件同目录下的 CallStack.txt 文件中, 内容格式如下:

```

13 ▼ SerialInit @ 0x0000015C - 0x000001CE
14 | 0x00000168 PORT_Init
15 | 0x00000174 PORT_Init
16 | 0x0000019C UART_Init
17 | 0x000001A2 UART_Open
18 |
19 ▼ main @ 0x000001D0 - 0x00000230
20 | 0x000001D2 SystemInit
21 | 0x000001D6 SerialInit
22 | 0x000001E6 GPIO_Init
23 | 0x000001F6 GPIO_Init
24 | 0x000001FE GPIO_ClrBit
25 | 0x00000206 GPIO_SetBit
26 | 0x00000210 GPIO_InvBit
27 | 0x00000218 GPIO_InvBit
28 |
29 ▼ fputc @ 0x00000232 - 0x0000024E
30 | 0x0000023C UART_WriteByte
31 | 0x00000244 UART_IsTXBusy

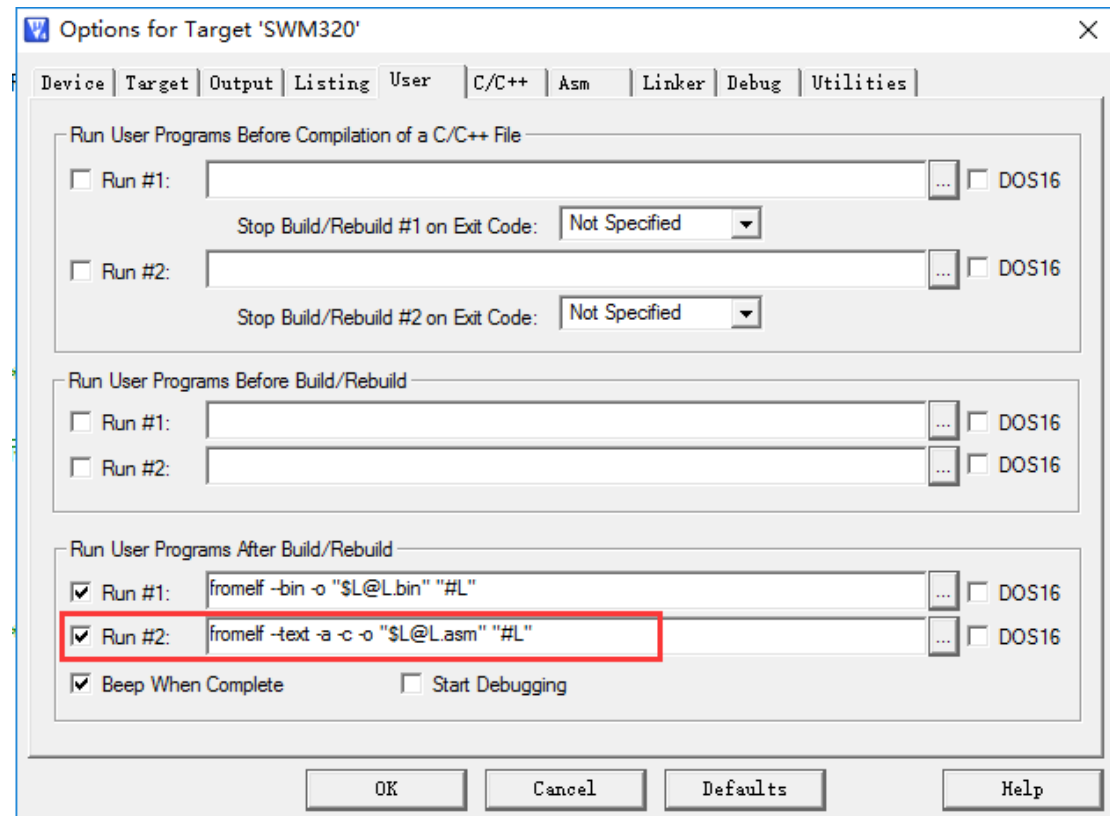
```

第零步、请确保 HardFault\_Handler() 中只有一个死循环, 没有额外代码

第一步、确保 J-Link 能连上芯片, 可使用 J-Link Commander 或 Keil 测试下连接

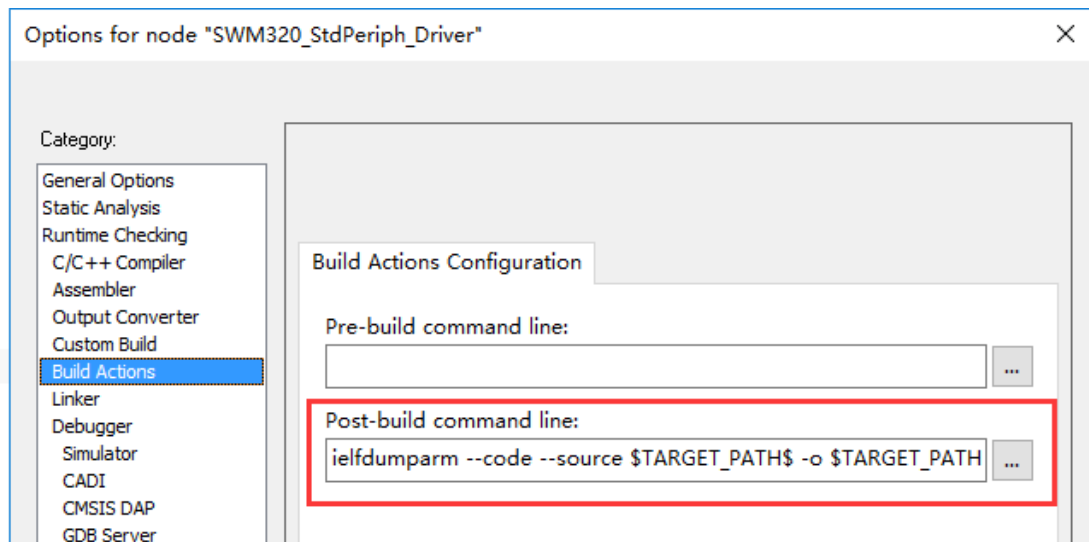


第二步、编译程序时生成程序反汇编文件，生成方法如下



生成命令 :fromelf --text -a -c -o "\$L@L.asm" "#L"

IAR 生成反汇编设置：



生成命令：ielfdumparm --code --source \$TARGET\_PATH\$ -o \$TARGET\_PATH\$.dis

第三步、双击 HFView.exe 运行，设置正确的 JLinkARM.dll 文件和生成的反汇编文件的路径

第四步、点击“栈读取”按钮，查看 IPSR 的值，若为 0x00000003，则说明发生了 HardFault，可点击“栈解析”分析发生 HardFault 的原因和位置；若 IPSR 的值不是 0x00000003，则说明没有发生 HardFault

## 七, 尝试使用 MicroPython 进行开发 (新手推荐)

MicroPython 教学文档见 Github,“/MicroPython/Teaching

Document/”文件夹。