



AC7802x 开发板入门指南

文档版本： 1.0

发布日期： 2022-11-30

© 2013 - 2022 杰发科技

本文档包含杰发科技的专有信息。未经授权，严禁复制或披露本文档包含的任何信息。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。

修订信息

版本	日期	作者	修订说明
1.0	2022-11-30	AutoChips	文档初版

版权声明

本文档包含 AutoChips 公司的机密信息。禁止未经授权使用或披露本文档包含的信息。对因未经 AutoChips 公司授权而全部或部分披露此文档内容而给 AutoChips 公司带来的任何损失或损害，AutoChips 公司将追究责任。

AutoChips 公司保留对此处任何信息进行更改的权利，此处的信息如有变更，恕不另行通知。AutoChips 公司对使用或依赖此处包含的信息不承担任何责任。

本文档的所有信息均“按原样”提供，不提供任何形式的明示，暗示，法定或其他形式的保证。AutoChips 公司明确拒绝对适销性，非侵权性和针对特定用途的适用性方面的所有暗示保证。AutoChips 公司对本文档可能使用、包含或提供的任何第三方软件不提供任何担保，并且用户同意仅向该等第三方寻求与此相关的任何担保索赔。AutoChips 公司对于根据用户规格或为符合特定标准或公开论坛而产生的任何交付物，也不承担任何责任。

文档目录

修订信息	2
版权声明	3
文档目录	4
插图目录	6
表格目录	8
1 开发板平台简介	9
1.1 AC7802x 通用开发板资源简介	9
1.2 AC7802x 通用开发板资源说明	9
1.3 开发板使用注意事项	17
2 芯片开发资料介绍	18
3 MDK5 软件入门	19
3.1 新建基于固件库的 MDK5 工程	19
3.2 程序的下载与调试	24
3.3 MDK5 仿真调试技巧	27
3.3.1 变量监控	27
3.3.2 查看 memory	28
3.3.3 查看外设寄存器	28
3.3.4 查看 CPU 寄存器	28
3.3.5 查看函数嵌套	29
4 AC7802x 开发基础知识介绍	30
4.1 系统架构	30
4.2 地址分配	30
4.3 时钟	31

4.4	端口复用和重映射	31
4.5	中断优先级管理	32
5	AC7802x 驱动库简介	33
5.1	驱动库结构	33
5.2	Device 文件夹介绍	33
5.2.1	启动文件	33
5.2.2	debugout_ AC7802x.c 文件	34
5.2.3	系统文件	34
5.2.4	寄存器定义	34
5.3	外设驱动	34
5.3.1	中断回调机制	34
5.4	UART 初始化示例	35

插图目录

图 1-1 AC7802x 开发板资源图	9
图 1-2 MCU 部分原理图	10
图 1-3 引出 GPIO 引脚示意图	10
图 1-4 接口及特殊功能引脚	11
图 1-5 12V 转 5V 电路图	12
图 1-6 5V 转 3.3V 电路图	12
图 1-7 供电切换电路图	13
图 1-8 串口转 USB 电路图	13
图 1-9 Nor Flash 原理图	14
图 1-10 EEPROM 原理图	14
图 1-11 调试接口	14
图 1-12 Reset 按键电路图	15
图 1-13 LED 电路图	15
图 1-14 KEY 电路图	16
图 1-15 NMI 电路	16
图 1-16 电位器原理图	17
图 3-1 设置工程保存路径	19
图 3-2 新建工程	19
图 3-3 选择芯片型号	20
图 3-4 Manage Run-Time Environment 界面	21
图 3-5 工程目录	22
图 3-6 添加 main 文件	23
图 3-7 输出代码	23
图 3-9 debug 选项卡设置	25
图 3-10 debug 参数配置	25
图 3-11 Flash Download 选项卡设置	26
图 3-12 下载程序	26
图 3-14 进入仿真	26
图 3-15 设置仿真断点	27
图 3-16 变量监控	27
图 3-17 查看 memory	28

图 3-18 查看外设寄存器	28
图 3-19 查看 CPU	29
图 3-20 函数调用嵌套	29
图 4-1 AC7802x MCU 系统架构	30
图 4-2 时钟树	31
图 5-1 AC7802x 驱动库结构	33
图 5-2 UART 初始化流程	35

表格目录

表 1-1 LVD/LVR 设置	12
表 4-1 复用功能配置	32

1 开发板平台简介

1.1 AC7802x 通用开发板资源简介

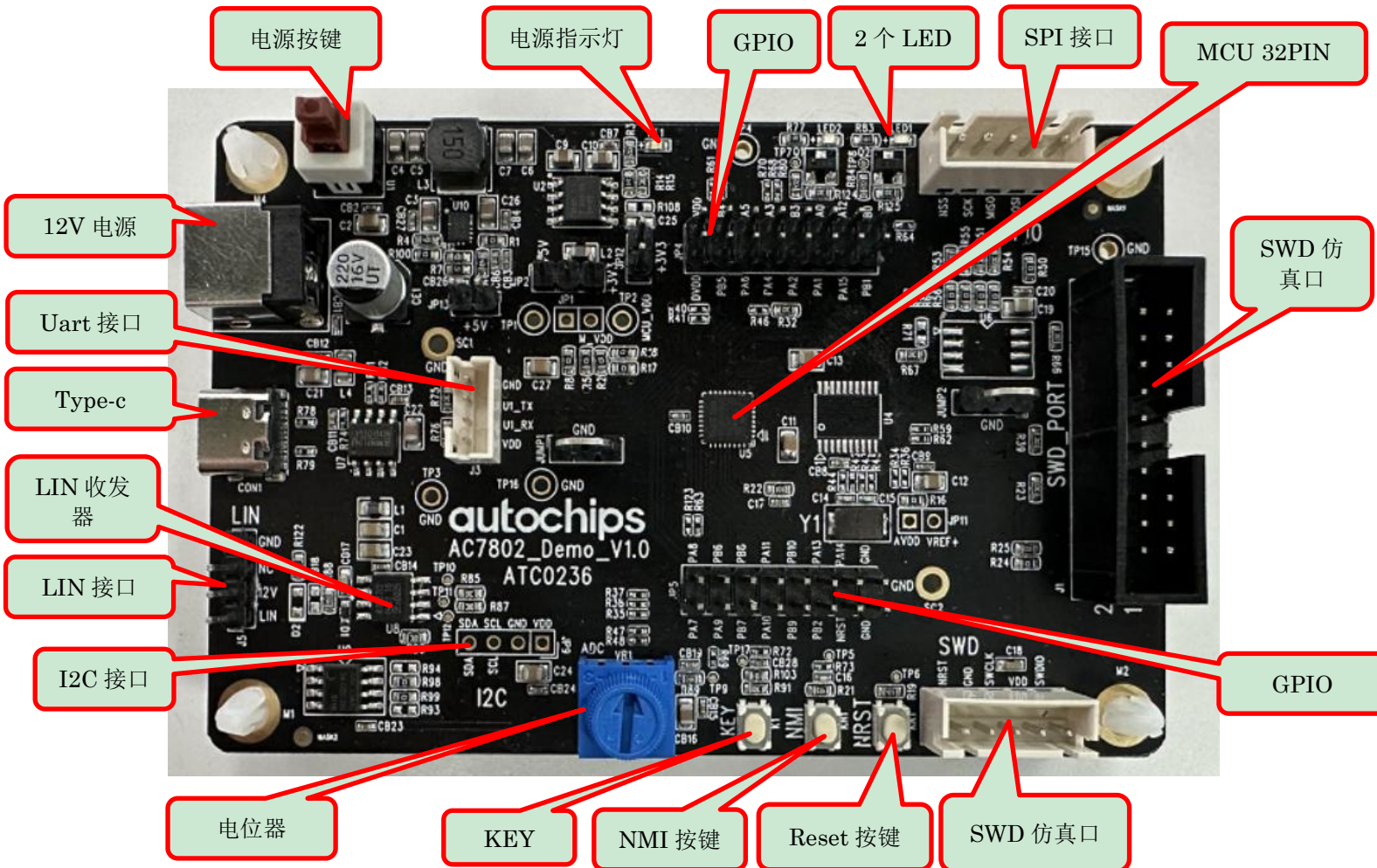


图 1-1 AC7802x 开发板资源图

1.2 AC7802x 通用开发板资源说明

1. MCU

芯片规格：请参见《ATC_AC7802x_ReferenceManual_CH》表 1-1 AC7802x 模块概述

MCU 部分原理图：

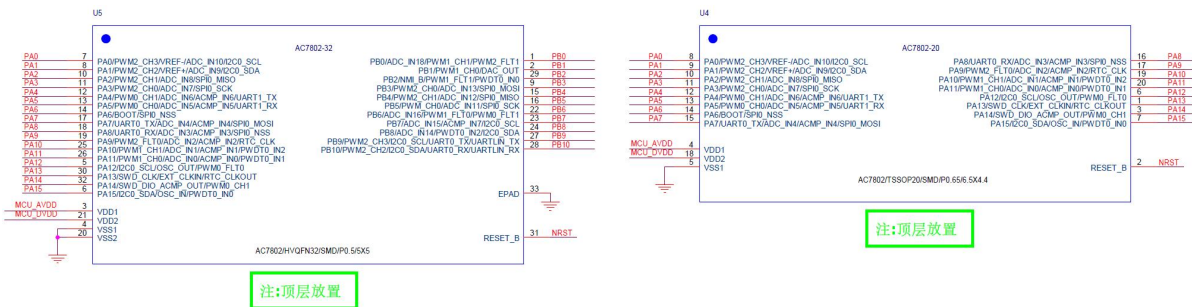


图 1-2 MCU 部分原理图

- AC7802x_demo 板支持两种封装型号的 MCU，分别为 20PIN 和 32PIN，高 PIN 脚对低 PIN 脚兼容，客户可以根据需求选择合适的封装型号。
- 芯片可选内部 RC 振荡电路或外部无源晶体作为芯片的时钟源，AC7802x_demo 板的外部晶振选择 16M 无源晶体。
- 芯片的供电分为 VDDA 和 VDD，分别用于芯片内部的数字部分和模拟部分供电，二者的供电电压相差不能超过 0.3V。详细信息可以参考《ATC_AC7802x_Datasheet_CH》表 5-4 电压和电流操作额定值。

2. GPIO

GPIO

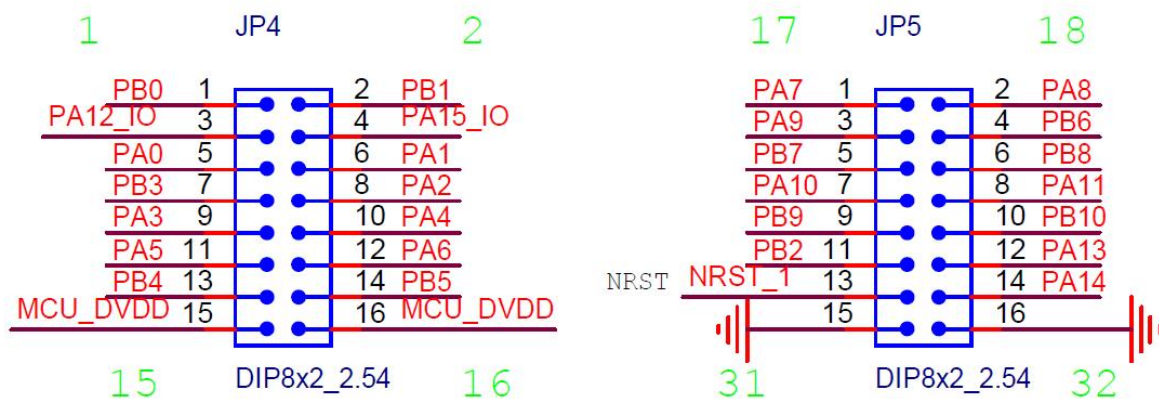


图 1-3 引出 GPIO 引脚示意图

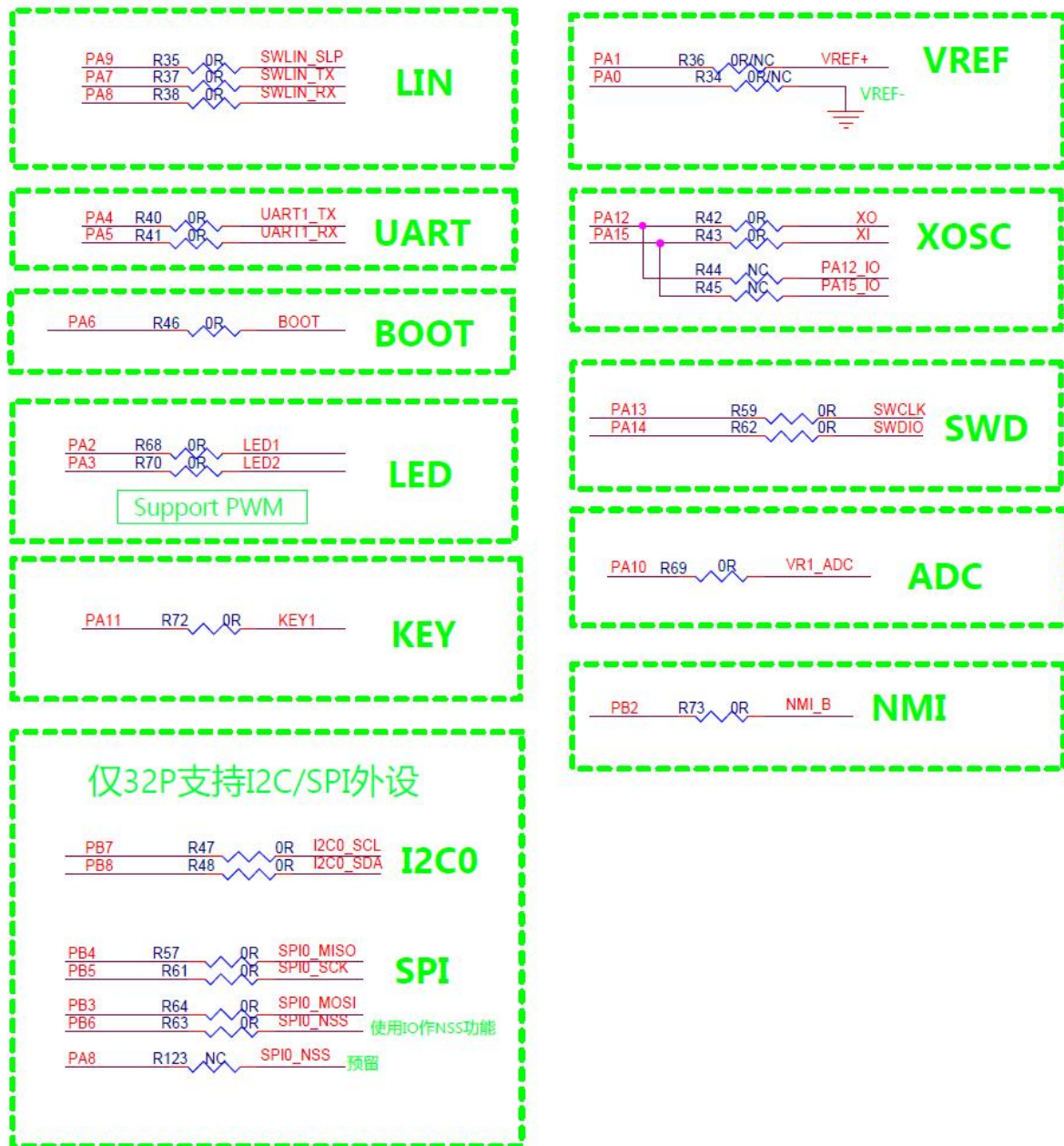


图 1-4 接口及特殊功能引脚

3. 12V/5V/3.3V 电源

板载外部 12V-DC 电源输入，标准直流电源插座，开发板板载 DC-DC 芯片 LN10X42Q1 和 LN20045Q1，可以将 12V 电源转换为 3.3/5V，用于给开发板提供高效、稳定的电源。因为使用了 DC-DC 芯片，所以开发板的供电范围非常宽，方便大家找到合适的电源给开发板供电。在使用了 LIN 通讯的情况下，建议使用外部电源供电，以保证 LIN 通信正常工作。

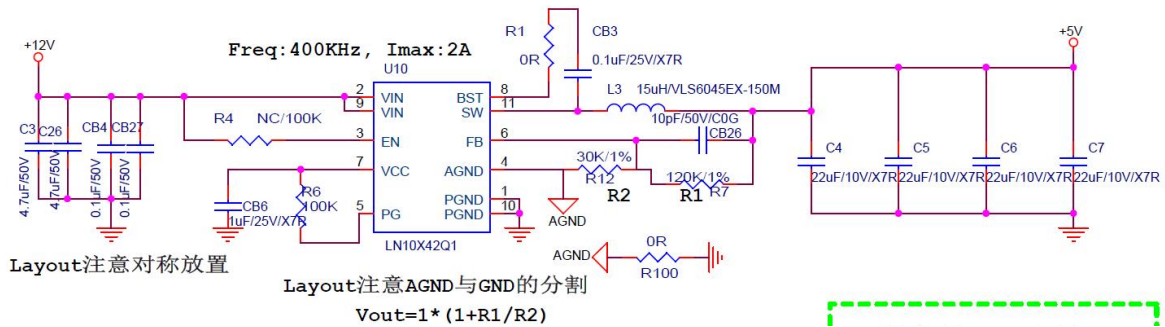


图 1-5 12V 转 5V 电路图

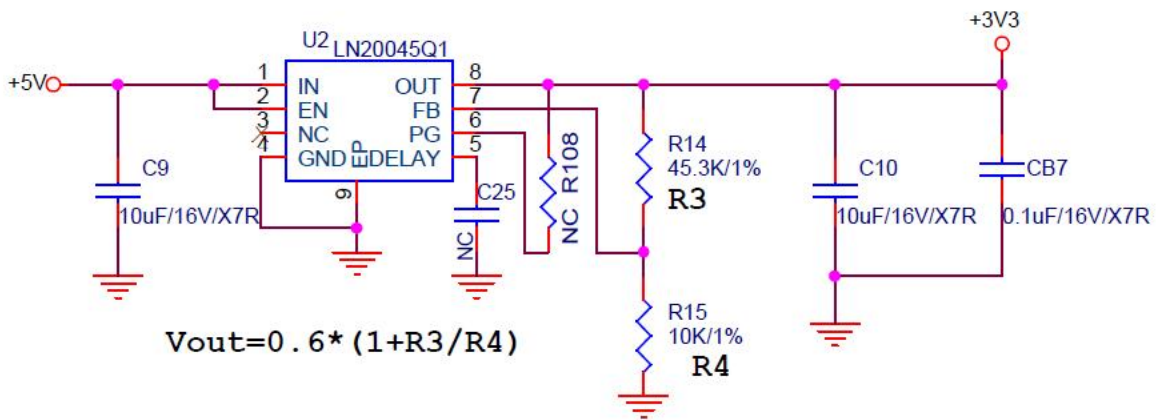


图 1-6 5V 转 3.3V 电路图

4. 5V/3.3V 选择口

AC7802x 支持 2.7~5.5V 宽电压输入，AC7802x_demo 板默认情况下是使用 3.3V 供电，若要使用 5V 供电系统，需要断开 L2 磁珠，使用跳线帽短接 JP2 的 1-2 引脚。若电压低于 LVR_THL 寄存器设定的 LVR 电压阈值会触发 LVR 低电压复位，也可以通过 LVD_THL 寄存器设置 LVD 电压阈值，触发 LVD 中断。

表 1-1 LVD/LVR 设置

寄存器数值	LVD 阈值选择	LVR 阈值选择
0	低档 2.9V	低档 2.6V
1	高档 4.6V	高档 4.3V

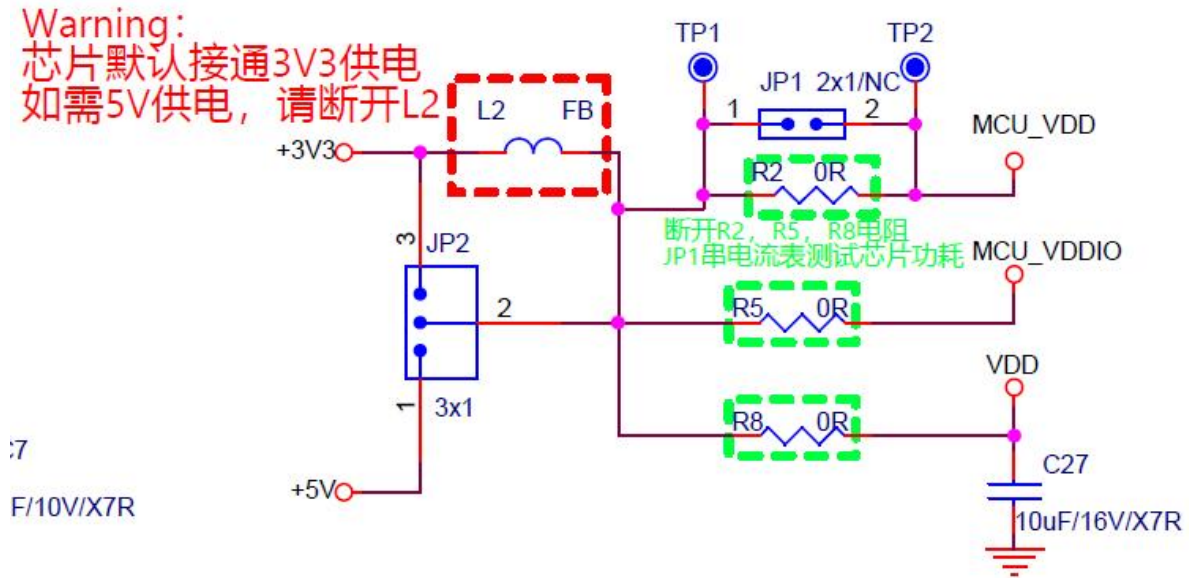


图 1-7 供电切换电路图

5. USB 转串口/UART1

USB 转串口座选用 Type-C 类型, AC7802x_demo 板选择用的 USB 转串口芯片为 CH340, win10 下可以直接识别到串口, win7 需要安装 (CH340) 驱动, 用户也可以通过 USB 口对开发板提供 5V 供电。

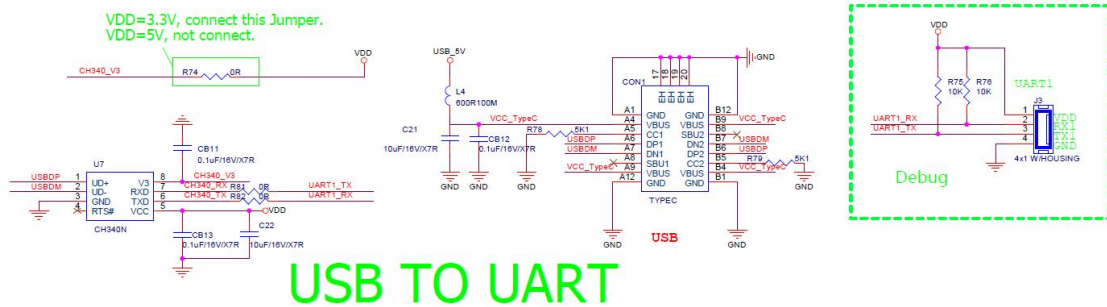


图 1-8 串口转 USB 电路图

6. 电源指示灯

电源指示灯使用板载 3.3V 供电, 该指示灯 PL1 亮说明电源芯片都工作正常。

7. serial flash

预留 SPI 接口, 通过 SPI0 通信, 可用于扩展存储。

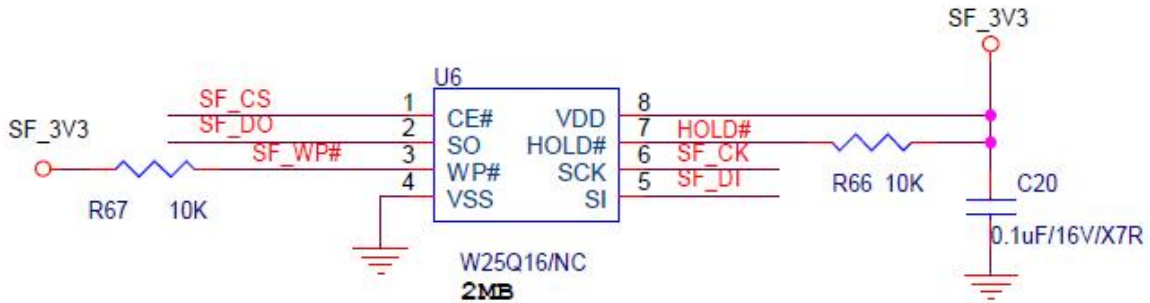


图 1-9 Nor Flash 原理图

8. 24C02 eeprom

预留 IIC 接口，通过 I2C0 通信，可用于扩展存储。

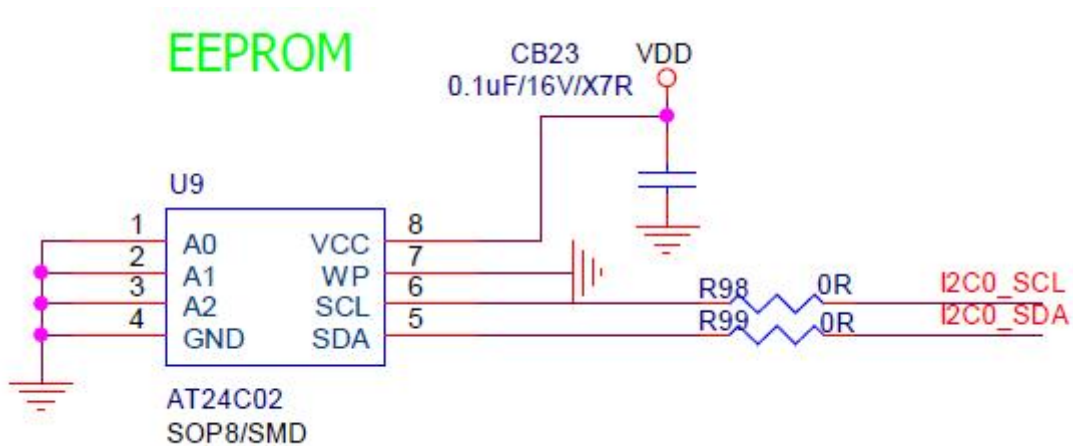


图 1-10 EEPROM 原理图

9. SWD 仿真口

预留 SWD 接口，分别有 5PIN 和 20PIN 调试底座，方便在线调试。

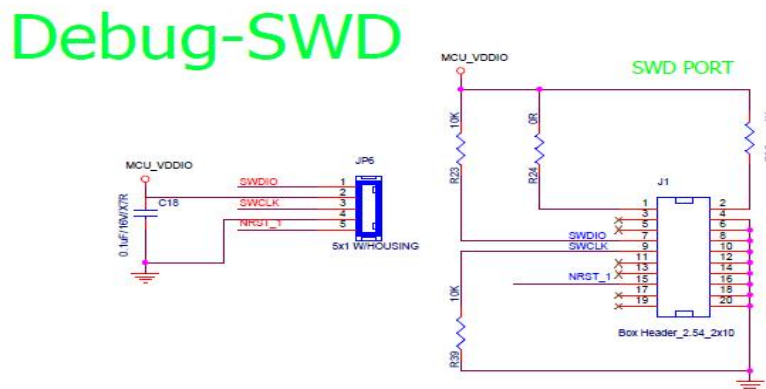


图 1-11 调试接口

10. 复位按键

单独引出芯片复位脚至按键 KR1，按下可使芯片复位，低有效。

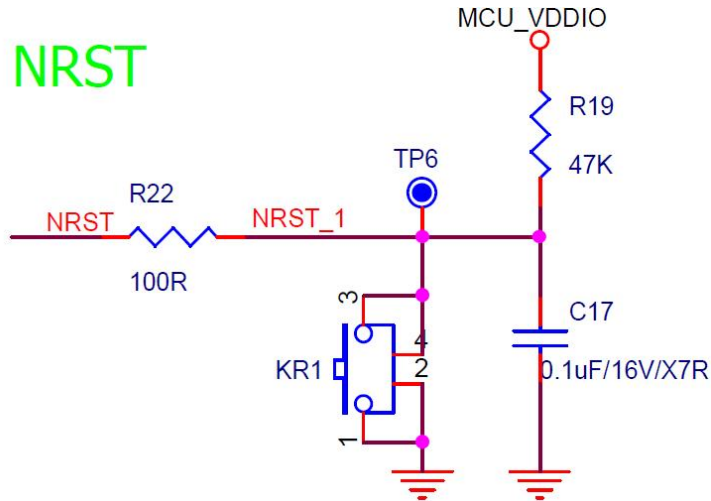


图 1-12 Reset 按键电路图

11. 2 个 LED

开发板提供了两个 LED（PA2/PA3）用于指示程序运行状态，两个灯都为绿色，两个 LED 分别通过一个三极管连接到 IO 口，IO 口通过输出高点亮 LED，同时 LED1/LED2 连接的 IO 口均具有 PWM 功能，可以通过 PWM 输出来调节 LED 亮度。

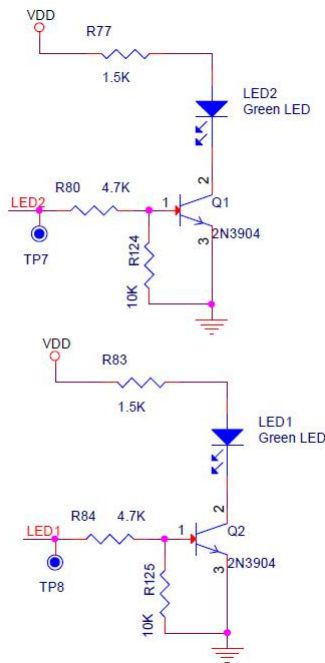


图 1-13 LED 电路图

12. 1 个按键

如图 1-14，单独引出芯片 1 个 GPIO 引脚至 K1(PA11)，按键可做输入检测用，也可用于 STOP 模式下的唤醒。

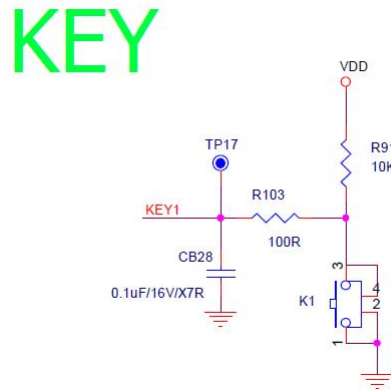


图 1-14 KEY 电路图

13. NMI 按键

如图 1-15，单独引出芯片 NMI 引脚至 KN1(PB2)，PB2 脚可以复用为 NMI 功能，为低电平时可触发 NMI 中断。

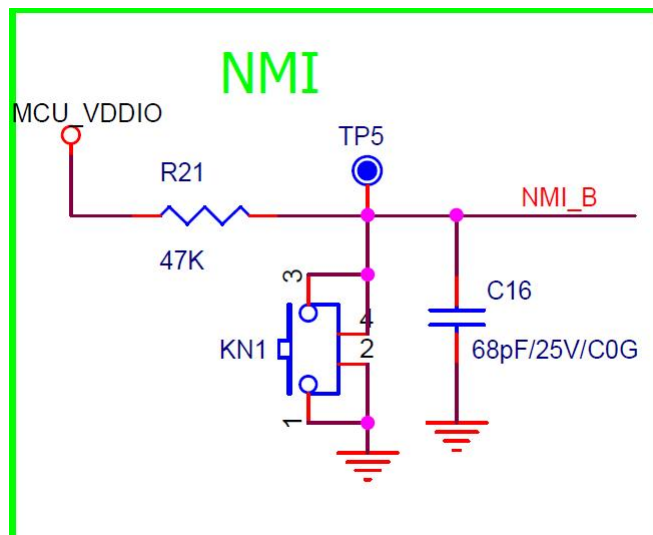


图 1-15 NMI 电路

14. 启动模式配置

AC7802x 支持 2 种启动模式：eflash 启动和 SRAM 启动。

1. eflash 启动：正常运行代码应该将代码烧录在 eflash，并从 eflash 启动运行。

2. SRAM 启动：芯片从 SRAM 起始地址（0x20000000）启动运行，用于仿真测试情况下，将代码写入到 SRAM 中启动运行。

15. 电位器

接到芯片 ADC_IN1(PA10)通道，通过 ADC 采样获取旋钮电压，用于 AD 检测功能测试。

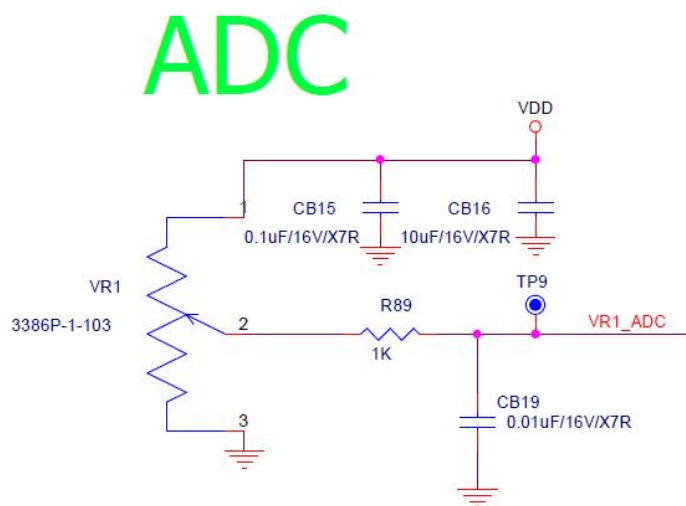


图 1-16 电位器原理图

1.3 开发板使用注意事项

为了便于大家更好的使用开发板，本节总结了一些使用过程中常见的需要注意的问题，大家在使用的时候需要注意：

- 1, 如需测试 LIN 模块，必须在连接 12V 电源环境下测试，因为 LIN 控制器工作在 12V，仅使用 USB 5V 供电无法测试 LIN 模块。
- 2, 在测试 LIN 模块时，需要使能 LIN 收发器，给收发器休眠引脚正确的电平（CAN 收发器为低电平，LIN 收发器为高电平）。
- 3, AC7802x 的 demo 板在 12V 直流电源和 USB 5V 同时存在的时候，MCU 为 12V 经 LDO 转换输出的电压供电。

2 芯片开发资料介绍

AC7802x 的 demo 板的软硬件资料，包括开发板硬件资料、芯片选型手册、参考手册、数据手册、CMSIS 驱动包，及基于 demo 板的应用例程，后续都会释放到官网上，链接如下：

http://www.autochips.com/jk/MCU1/info_225.aspx?pid=530&typeid=225

基于开发 AC7802x 芯片的相关开发工具也会释放到官网，链接如下：

http://www.autochips.com/jk/kfzzy/index_287.aspx

AutoChips 技术支持论坛（21ic），链接如下：

<https://bbs.21ic.com/iclist-864-1.html>

3 MDK5 软件入门

3.1 新建基于固件库的 MDK5 工程

在搭建基于固件库的 MDK5 工程之前，需要先准备如下资料：

- MDK5 开发环境安装，建议使用 5.23 版本（除 5.30）及以上版本。MDK5 安装包可以从 keil 官网下载：<https://www2.keil.com/mdk5/install>
- AutoChips.AC7802x_DFP.0.0.3.pack 驱动包安装（官网下载官网待更新），下载链接如下：http://www.autochips.com/jk/MCU1/info_225.aspx?pid=530&typeid=225

上述环境准备好后就可以开始新建工程了：

新建工程之前，建立一个新的文件夹，用于存放所有的工程文件，如图 3-1 所示。

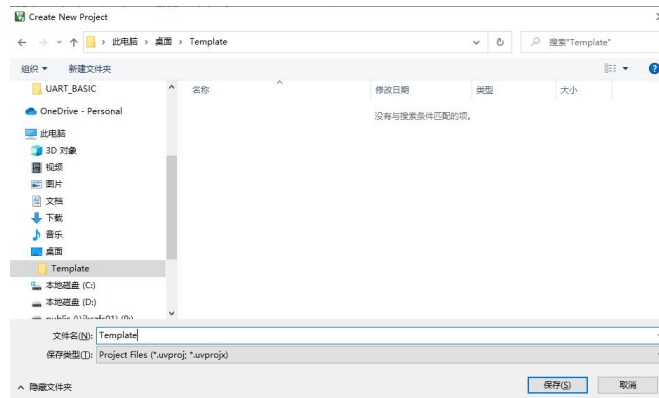


图 3-1 设置工程保存路径

打开 MDK 软件，按照以下步骤 Project->New uVision Project 新建工程，在弹出的界面定位到之前建立的 Template 文件加下，然后在该文件夹下新建 Project 文件。

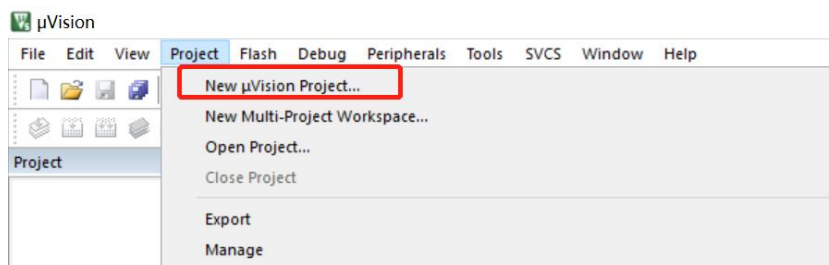
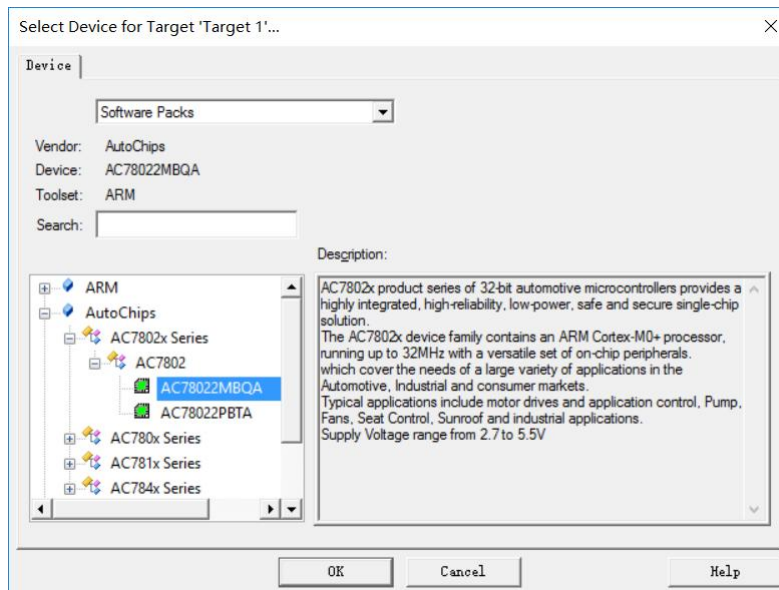


图 3-2 新建工程

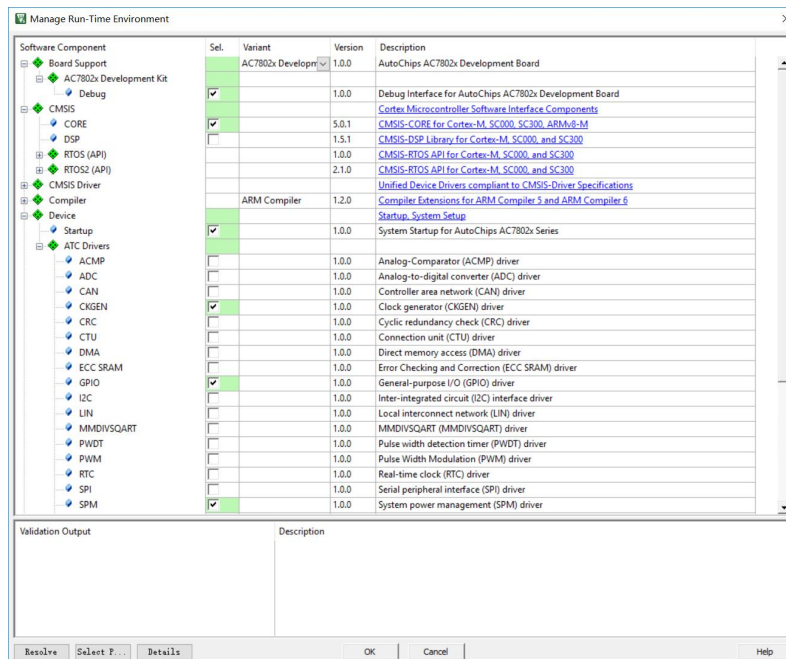
选择所使用的芯片型号：在完成上述步骤后，接下来会弹出一个芯片型号选择界面，如图 3-3 所示，按照 AutoChips->AC7802x Series-> AC7802 -> AC78022MBQA 步骤选择 demo 板对应的芯片型号（如果使用的其他型号芯片，选择对应的型号即可）。



注意： 只有正确安装了上面的 AutoChips.AC7802x_DFP.0.0.3.pack（版本会基于 pack 版本有所变动）驱动包，才能找的到 AC7802x 的芯片型号，如果没有该选项，请重新安装改驱动包。

图 3-3 选择芯片型号

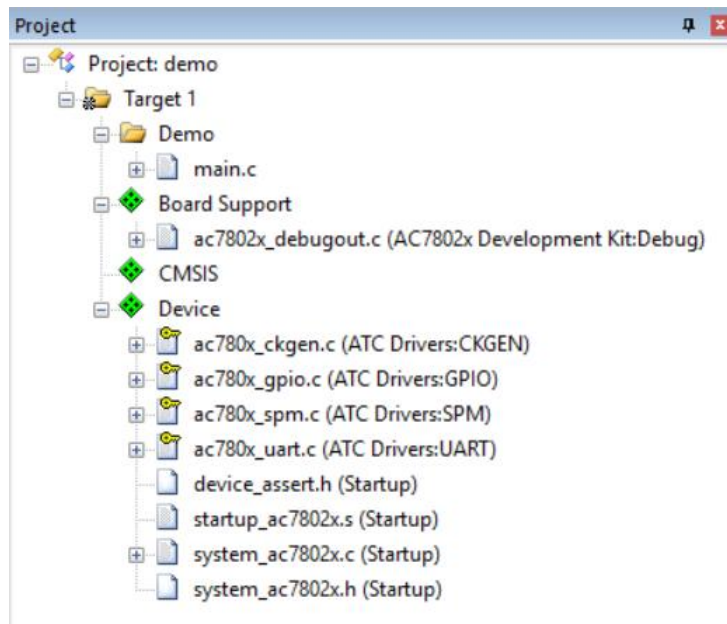
完成芯片选型后，会弹出 Manage Run-Time Environment 窗口，如图 3-4 所示，可以在该对话框中添加启动文件、内核文件及所需要的外设驱动。如果 driver 没有正确勾选，会在 Validation Output 窗口提示依赖的驱动，按提示勾选就可以。



说明：这是 MDK5 新增的功能，方便用户添加自己需要的组件，如果想要自己添加驱动等文件，可以直接点 Cancel

图 3-4 Manage Run-Time Environment 界面

此次开发入门例程仅示范简单的 LED 闪烁功能，需添加 Debug 及 GPIO 等.c 文件，添加完成后可在 Project 窗口显示已添加的 driver。



注意：图中部分文件有钥匙样式图案，提示该文件为只读属性文件，这是因为在 RTE 环境下新建工程添加的驱动是从安装的 pack 目录下引用的，在该环境下新建的工程共用一份 driver，如果修改的话，会影响到其他工程。如需修改，需将相应的文件 copy 到工程本地路径下，修改只读属性，然后通过手动添加的方式加入到工程中。并去掉 RTE 环境中的文件，避免重复定义。

图 3-5 工程目录

完成上述步骤后，开发环境基本已经搭好，接下来添加项目的开发代码就可以了。在 Template 文件夹下新建一个 User 文件夹，新建一个 main.c 文件，然后将其添加到工程中的 User 文件夹下。

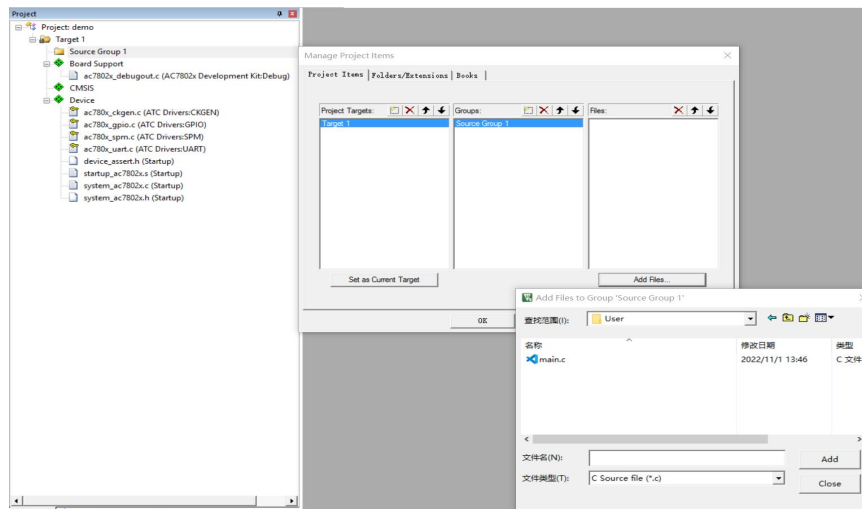


图 3-6 添加 main 文件

在 main.c 文件中编写 LED 闪烁的代码，调用 Initdebug 和 InitDelay 接口，编译可以使用 printf 函数和 mdelay 延时函数，代码如下。

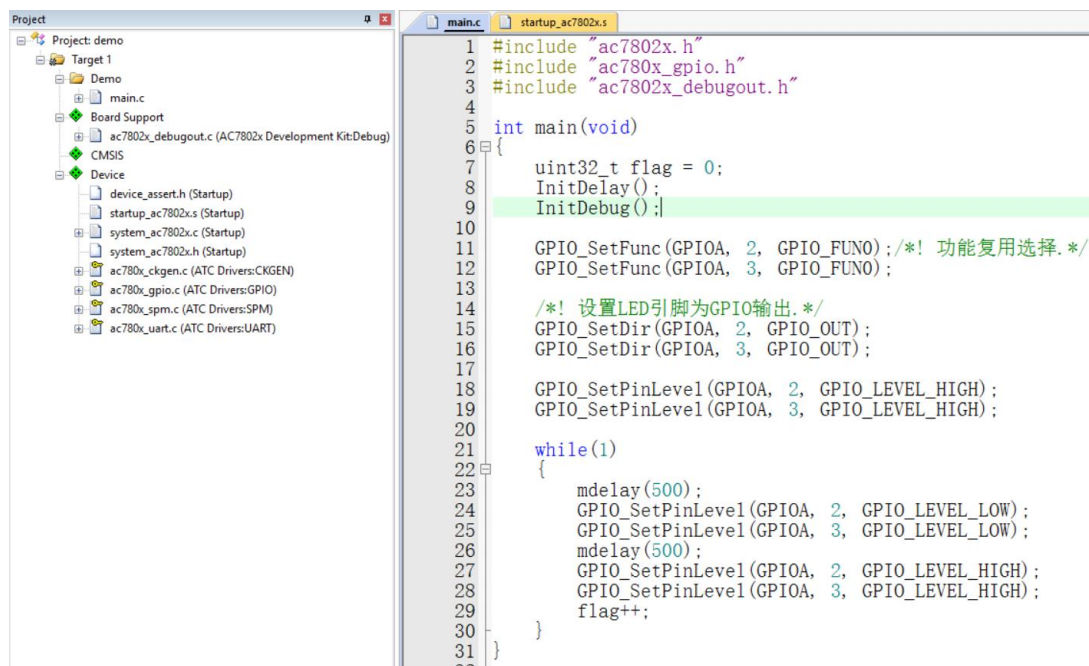


图 3-7 输出代码

附代码：

```

#include "ac7802x.h"
#include "ac780x_gpio.h"
#include "ac7802x_debugout.h"

```

```
int main(void)
{
    Uint32_t flag = 0;
    InitDelay();
    InitDebug();
    /*! 功能复用选择.*/
    GPIO_SetFunc(GPIOA, 2, GPIO_FUNC0);
    GPIO_SetFunc(GPIOA, 3, GPIO_FUNC0);
    /*! 设置 LED 引脚为 GPIO 输出.*/
    GPIO_SetDir(GPIOA, 2, GPIO_OUT);
    GPIO_SetDir(GPIOA, 3, GPIO_OUT);
    /*! LED 默认点亮.*/
    GPIO_SetPinLevel(GPIOA, 2, GPIO_LEVEL_HIGH);
    GPIO_SetPinLevel(GPIOA, 3, GPIO_LEVEL_HIGH);

    while(1)
    {
        mdelay(500);
        GPIO_SetPinLevel(GPIOA, 2, GPIO_LEVEL_LOW);
        GPIO_SetPinLevel(GPIOA, 3, GPIO_LEVEL_LOW);
        mdelay(500);
        GPIO_SetPinLevel(GPIOA, 2, GPIO_LEVEL_HIGH);
        GPIO_SetPinLevel(GPIOA, 3, GPIO_LEVEL_HIGH);
        flag++;
    }
}
```

3.2 程序的下载与调试

完成新工程的创建后，本小节介绍如何将编译成功的 demo 程序下载到开发板上并进行调试。需要准备以下环境：

- AC7802x_demo 板
- 12V 直流电源
- 调试器（ARM 仿真器都可以，包括但不限于：ATC-LINK、JLINK、ULINK、ST-LINK）

- 串口转接板
- PC

连接好仿真器后，点击工程界面上面的魔术棒，然后在 Debug 界面，选择所使用的仿真器型号，ATC-LINK 对应的仿真器型号选择 CMSIS-DAP Debugger。



图 3-9 debug 选项卡设置

点击 Setting 按钮，设置一些仿真参数，Port 可以选择 SW 或 JTAG 模式，AC802x 只支持 SW 模式。

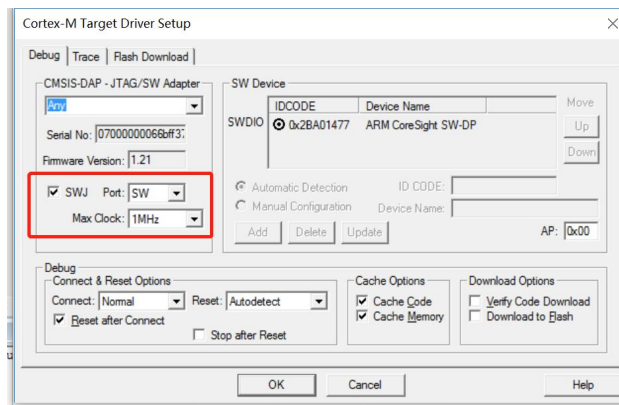


图 3-10 debug 参数配置

然后进入 Flash Download 选项卡，配置芯片下载算法和动作，一般选择默认配置，建议勾选 Reset and Run，这样可以在下载程序后，软件自动复位运行。不勾选则下载后程序不会运行，需要手动按下复位按钮程序才会运行。

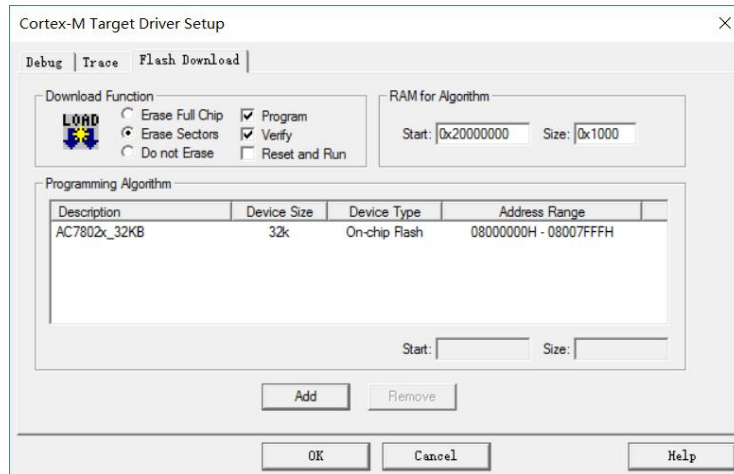


图 3-11 Flash Download 选项卡设置

配置完成后点击 OK，然后点击 Load 按钮，将程序下载到开发板中：



图 3-12 下载程序

➤ 程序下载成功后，就已经开始运行了，此时观察 AC7802x_demo 板，LED 以 1Hz 频率闪烁。

点击 Debug 按钮进入仿真，然后通过左上角的按钮，可以控制芯片复位，全速运行，停止，单步运行、运行到断点处等操作。

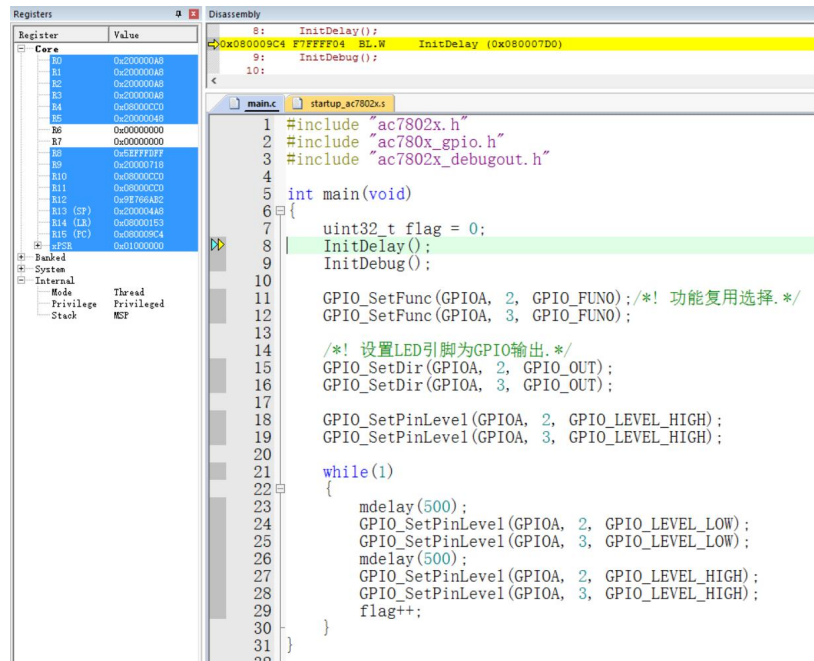
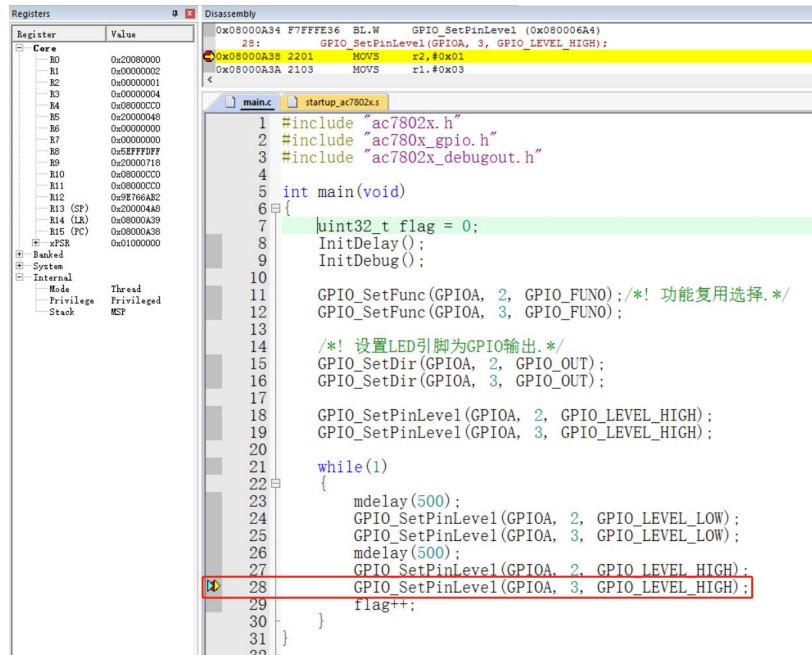


图 3-14 进入仿真



注意： win10 环境下 ATC-LINK 免驱，win7 需要安装驱动，可以在 AutoChips 官网上下载 ATC-LINK 的驱动包。下载链接如下：http://www.autochips.com/jk/kfzzy/index_287.aspx

图 3-15 设置仿真断点

3.3 MDK5 仿真调试技巧

3.3.1 变量监控

通过 Watch 界面，可以对一些变量进行监控，图 3-16 中添加了 flag 变量的监控：

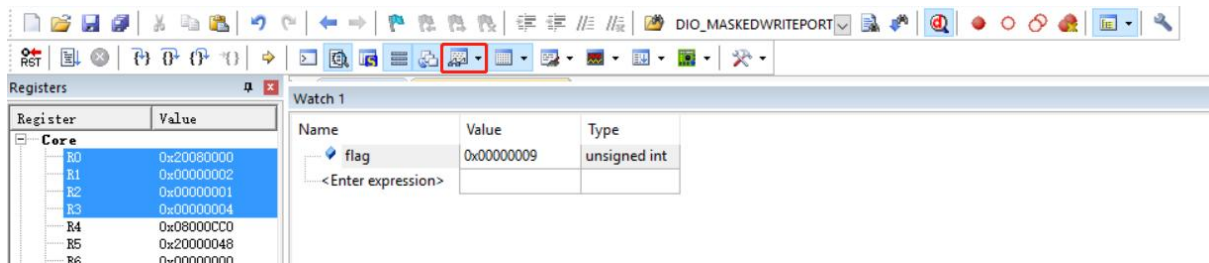


图 3-16 变量监控



注意： 变量监控最好对全局变量进行监控，一些局部变量或静态变量有时候会监控不到他的值。

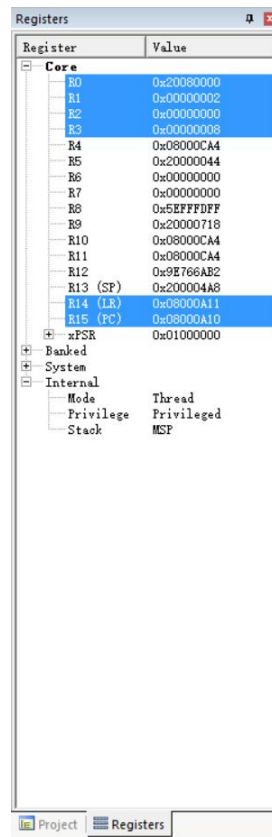


图 3-19 查看 CPU

3.3.5 查看函数嵌套

通过 call stack windows，可以看到函数调用的嵌套关系。

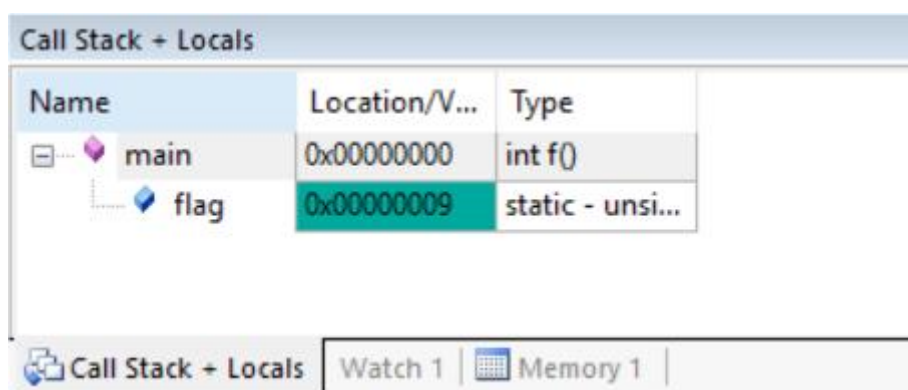


图 3-20 函数调用嵌套

通过此窗口，可以看到当前代码停在哪个函数，以及从哪个函数进入的这个函数，逐级往上。在调试程序的时候较为重要。例如在程序进入 hardfault handler 时可通过此窗口，查询是从那个函数跳转进入 hardfault handler 的。

4 AC7802x 开发基础知识介绍

4.1 系统架构

AC7802x 系统架构图如图 4-1 所示。

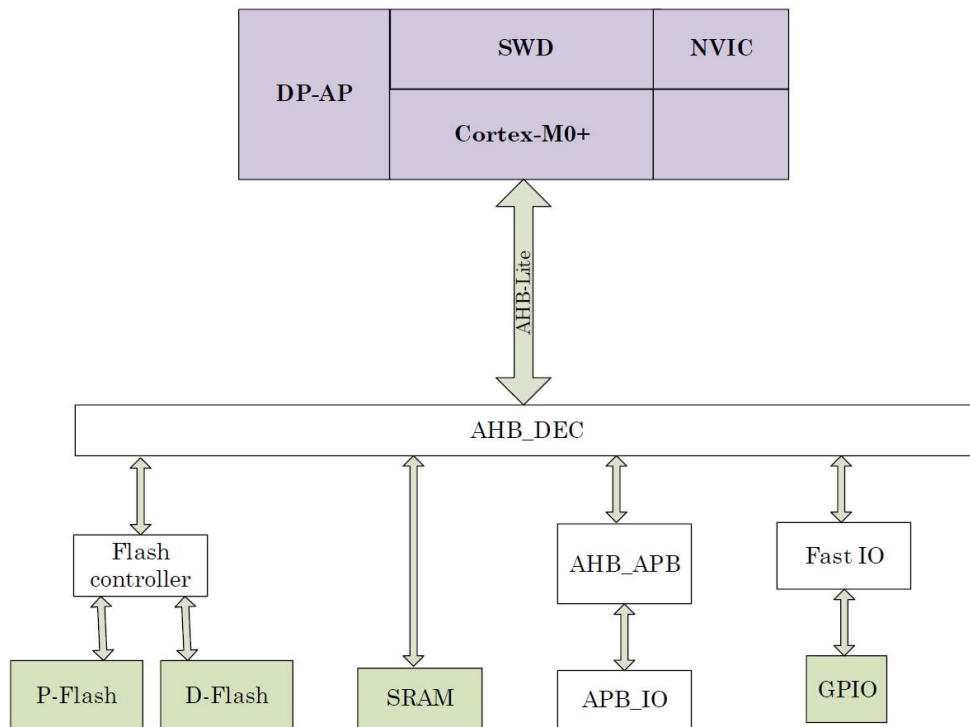


图 4-1 AC7802x MCU 系统架构

4.2 地址分配

AC7802x 采用的 cortex-M0+内核，支持 eFlash 启动和 SRAM 启动，数据总线为 32 位宽度，程序存储器(P-Flash)、数据存储器(D-Flash)、寄存器和输入输出（I/O）接口统一编址在 4G 字节的线性地址空间里。详细分配请参见《ATC_AC7802x_ReferenceManual_CH》表 2-2 存储器映射表。

4.3 时钟

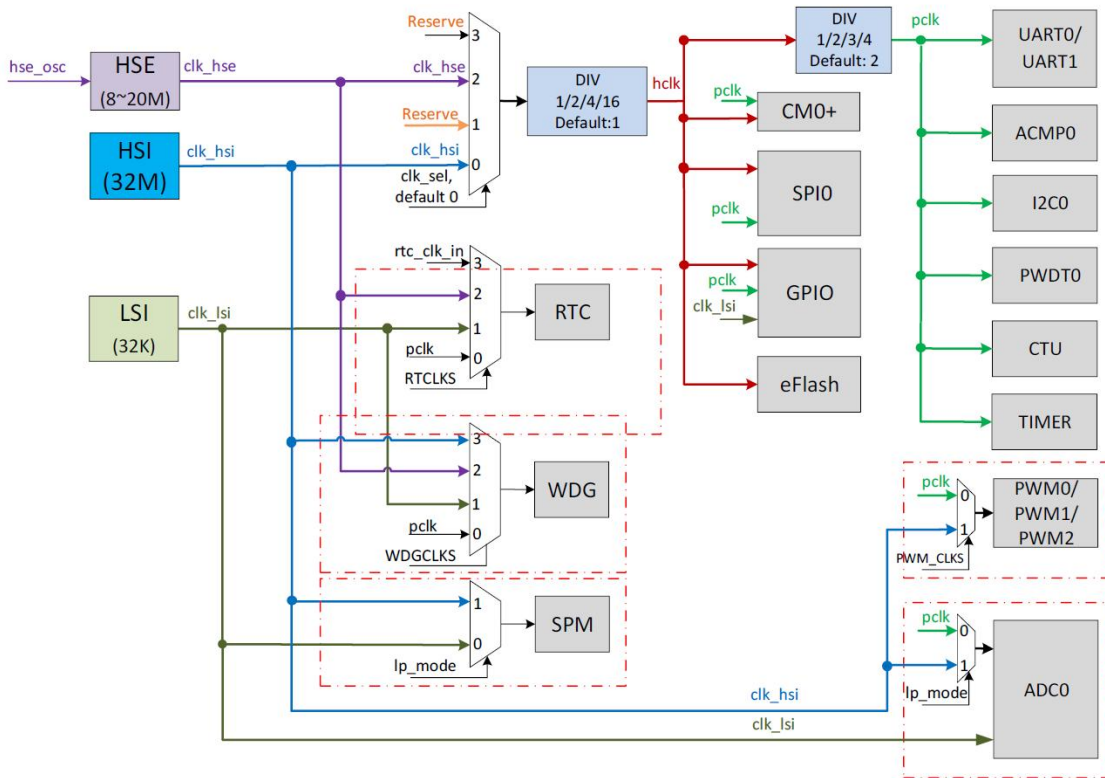


图 4-2 时钟树

AC7802x 共有 3 个时钟源：

- 高速内部时钟（HSI）：内部 RC 振荡器提供 32MHz 时钟源。
- 外部高速时钟（HSE）：外部 OSC 提供 8MHz ~20MHz 晶振。
- 低速内部时钟（LSI）：内部低速 RC 振荡器提供 32KHz 时钟源。

4.4 端口复用和重映射

AC7802x 有丰富的外设资源，为了方便用户的硬件设计，AC7802x 的 GPIO 引脚有 4 种复用功能，不同的外设可以被复用到不同的 GPIO 引脚上。具体 GPIO 支持的复用功能可以查阅《ATC_AC7802x_ReferenceManual_CH.pdf》表 14-2 GPIO 复用功能描述。

表 4-1 复用功能配置

32 Pi n	20 Pi n	引脚 名称	功能 0	功能 1	功能 2	功能 3	复用功能配 置	GPIO (序号)
12	12	PA4	GPIO	PWM0_CH1	ADC_IN6/ ACMP_IN6	UART1_TX	PMUX0[14: 12]	4
13	13	PA5	GPIO	PWM0_CH0	ADC_IN5(I)ACMP_IN 5(I)	UART1_RX	PMUX0[17: 15]	5

将 PA4, PA5 脚设置为 UART 功能示例代码如下:

```
GPIO_SetFunc(GPIOA, GPIO_PIN4, GPIO_FUN3);
GPIO_SetFunc(GPIOA, GPIO_PIN5, GPIO_FUN3);
```

4.5 中断优先级管理

AC7802x 支持 0~3 级中断优先级配置, 0 级最高。可以通过内核头文件 core_cm0plus.h 提供的 NVIC_SetPriority 函数来设置中断优先级, 例如设置 UART1 中断优先级为 2:

```
NVIC_SetPriority(UART1_IRQn, 2);
```


5 AC7802x 驱动库简介

5.1 驱动库结构

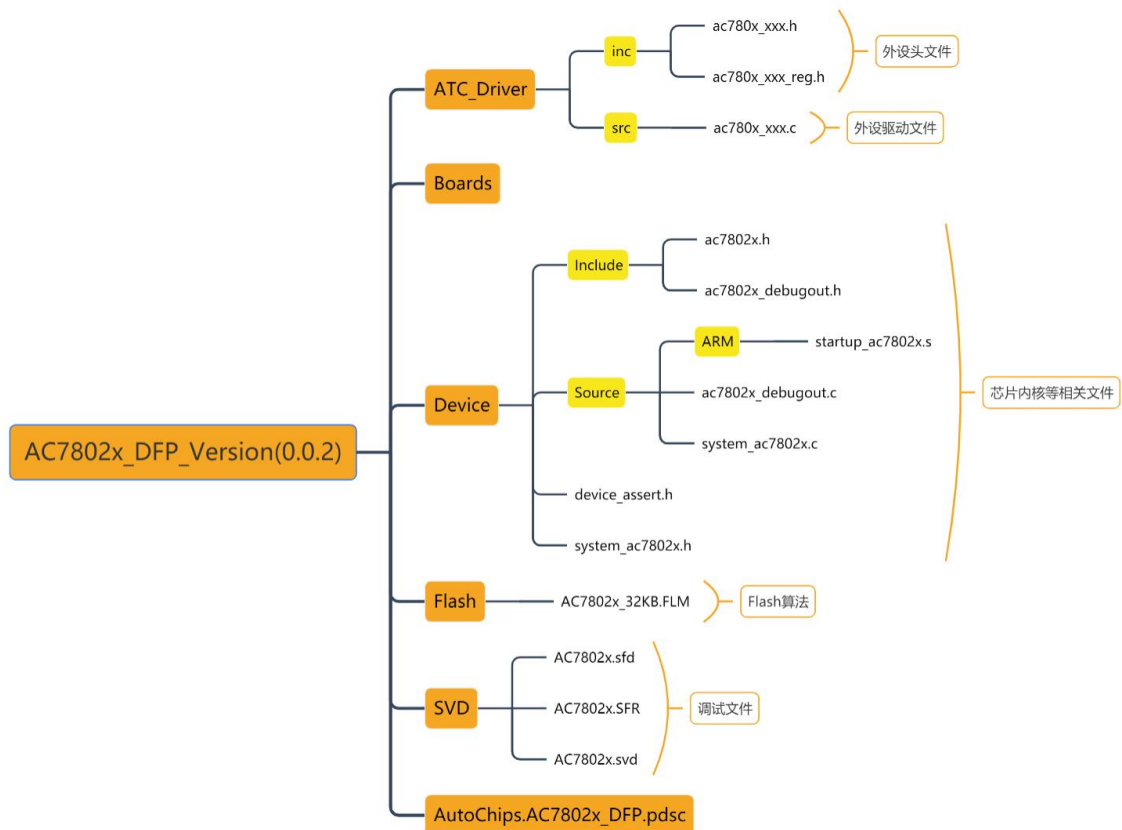


图 5-1 AC7802x 驱动库结构

5.2 Device 文件夹介绍

5.2.1 启动文件

启动文件 `startup_ac7802x.s` 存放在 `Device->Source->ARM` 下，在启动文件中定义了栈空间的大小，还有中断向量表等相关相关定义。在芯片上电后，会首先执行 `Reset_Handler` 函数，在该函数中会调用 `SystemInit` 函数对 MCU 进行初始化，然后跳转到 `_main` 库，最后在 `_main` 库中会跳转到 `main` 函数中运行。

5.2.2 debugout_AC7802x.c 文件

在 Device->Source 路径下，存放了 ac7802x_debugout.c 文件，可用于实现格式化打印输出的重定向，通过调用其初始化 InitDebug() 函数，将 printf 输出重定向到芯片的 UART1 输出（PA4，PA5）。



注意：如果代码中使用到了 printf 相关的函数，必须要包含此文件，或者自己实现重定向功能，否则会导致程序运行不正常。

5.2.3 系统文件

在 Device->Source 文件夹下存放有 system_ac7802x.c 文件，该文件实现了对芯片系统寄存器相关的操作，例如读取芯片的专属 UUID。

5.2.4 寄存器定义

芯片所有的寄存器都定义在 ac7802x.h 这个文件中，如果用户不想使用官方提供的驱动库，而想自己通过寄存器操作的方式来实现功能，只需要引入此文件即可。

5.3 外设驱动

如图 5-1 介绍，每个外设模块包含三个文件，具体文件功能如下：

1. ac780x_xxx.c: 封装了底层的部分功能接口，一般包括初始化函数 xxx_Init，反初始化函数 xxx_DeInit，以及一些相应的操作接口。
2. ac780x_xxx.h: 对外的函数，类型等申明。
3. ac780x_xxx_reg.h: 对寄存器的相关操作，在该文件中会以 inline 函数的方式实现。

5.3.1 中断回调机制

为了便于对中断的管理，每个驱动文件中已经实现了对应外设的中断 IRQ handler，因此，在编写应用代码的时候，不需要自己去实现中断入口函数。下面是 UART 驱动代码中的中断入口函数：

```
/*!
 * @brief UART0 Interrupt Handler
 * @param[in] none
 * @return none
 */
void UART0_IRQHandler(void)
{
```

```
UART_InterruptHandler(UART0);
}
```

考虑到应用上会需要在中断中执行一些操作以及判断一些事件标志。因此提供了以回调函数的方式实现应用层执行中断代码。中断回调函数的原型统一如下：

```
typedef void (*DeviceCallback_Type)(void *device, uint32_t wpara, uint32_t lpara);
```

注册回调函数的方式有两种，一可以在初始化结构体中，将回调函数赋值给 callback 成员，或者可以通过调用 xxx_SetCallBack 函数。下面是 UART 注册回调函数的方式。

```
void UART_SetCallback(UART_Type *UARTx, const DeviceCallback_Type callback)
```

5.4 UART 初始化示例

UART 初始化流程如图 5-2 所示。

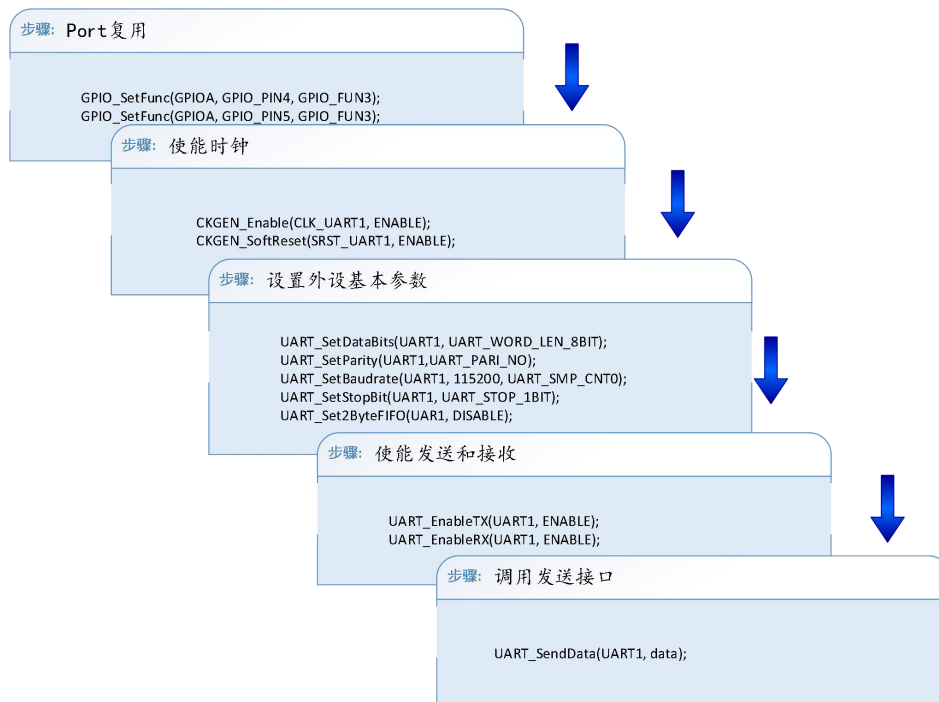


图 5-2 UART 初始化流程

示例代码：

```
UART_ConfigType s_uartUserConfig = { //uart 初始化结构体
    .baudRate = 115200U,
    .dataBits = UART_WORD_LEN_8BIT,
    .stopBits = UART_STOP_1BIT,
```

```
.parity = UART_PARI_NO,  
.fifoByteEn = DISABLE,  
.sampleCnt = UART_SMP_CNT0,  
.callBack = NULL,  
};  
  
UART_Init(UART1, &s_uartUserConfig); // 调用初始化接口  
  
UART_SendData(UART1, data); // 调用发送接口
```