



摘要

本技术说明文档提供关于 CPW3101 开发板 UFCS_EVB_V1.2 功能介绍、特性说明、使用说明、IAP 升级、使用注意事项等，旨在帮助用户更好地使用 CPW3101 开发板_V1.2，进行芯海科技 CPW3101 SOC 应用产品的设计、开发和调试。

适用范围

| 类型 | 适用产品型号或系列 | 说明 |
|----|--------------------------------|----|
| | 芯海科技 CPW3101 开发板 UFCS_EVB_V1.2 | |

版本

| 历史版本 | 修改内容 | 日期 |
|------|------|------------|
| V1.0 | 初版生成 | 2023-02-28 |

芯海科技CHIPSEA

目 录

| | |
|----------------------------|-----------|
| 1. 概述 | 4 |
| 1.1 方案概述..... | 4 |
| 1.2 cpw3101 芯片概述..... | 4 |
| 1.2.1 主要特性..... | 4 |
| 2. EVB 介绍 | 6 |
| 2.1 开发板驱动安装..... | 6 |
| 2.2 开发板硬件资源介绍..... | 6 |
| 2.3 开发板原理图 EVB 说明..... | 7 |
| 2.3.1 MCU 部分原理图..... | 7 |
| 2.3.2 电源电路..... | 8 |
| 2.3.3 CPW3101 部分电路..... | 9 |
| 2.3.4 UART 转 USB 部分电路..... | 10 |
| 3. IAP 固件升级 | 11 |
| 4. 硬件接口 | 12 |
| 4.1 通信接口..... | 12 |
| 4.2 硬件连接..... | 12 |
| 4.3 进入升级模式..... | 13 |
| 5. 升级流程 | 14 |
| 5.1 固件下载流程..... | 14 |
| 5.2 跳转流程..... | 15 |
| 6. 命令协议 | 17 |
| 6.1 通信协议..... | 17 |
| 6.2 命令包格式..... | 17 |
| 6.2.1 命令数据包..... | 17 |
| 6.2.2 响应数据包..... | 17 |
| 6.3 命令详解..... | 18 |
| 6.3.1 命令列表..... | 18 |
| 6.3.2 CMD_UPGRADE 命令..... | 18 |
| 6.3.3 CMD_VERSION 命令..... | 18 |
| 6.3.4 CMD_ERASE 命令..... | 19 |
| 6.3.5 CMD_PROGRAM 命令..... | 19 |
| 6.3.6 CMD_CRC 命令..... | 20 |
| 6.3.7 CMD_JUMP 命令..... | 20 |
| 7. 时序特性 | 21 |
| 8. KEIL 工程搭建 | 22 |
| 9. 仿真调试 | 24 |
| 10. 注意事项 | 26 |

1. 概述

1.1 方案概述

本说明文档介绍的 UFCS_EVB_V1.2 开发板，为使用芯海科技 CPW3101 芯片开发设计的一块仿真评估板，开发板由 CPW3101 芯片加一颗主控 MCU CS32F035 组成，主控 MCU 和上位机软件采用 uart 通信，符合通信协议主控的 MCU CS32F035 和 CPW3101 采用 IIC 通信。开发板的 TypeA 自带 USB 转串口控制芯片，通过该端口，用户可以通过上位机跟开发板上的主控 MCU 进行串口通信，主控 MCU 可以模拟 Sink 设备或 Source 设备。开发板的 TypeC 端口可以连接 TypeC 线缆，并通过 TypeC 线缆的 D+和 D-与 CPW3101 的 uart TX 和 uart RX 进行连接。

1.2 cpw3101 芯片概述

CPW3101 是一个可编程 UFCS 协议控制芯片，最高可运行在 24MHz，内置 32K 字节 Flash，2K 字节 SRAM，支持 UART、I²C、定时器等外设，具有高整合度、高抗干扰、高可靠性的特点。它的封装为 WLCSP8，工作温度范围为-40 度~85 度，工作电压范围 2.2V~5.5V。芯片提供多种低功耗模式，以满足不同的应用。

CPW3101 具备“一芯双向”能力，在应用中既能作为 sink 端设备又能作为 source 端设备，集成双向 UFCS 协议、可实现双向电源应用场景，满足多种终端产品的设计需求，能够满足智能手机、笔记本电脑、移动电源等“充电+放电”等双向电源应用场景需要。

1.2.1 主要特性

1.通信

1 个通用异步收发器 (uart)

支持波特率自适应 (发送 0xAA)

UART 的 RX/TX 集成上拉 RPULL_UP (10K Ω \pm 10%)

USART 的 RX/TX 支持交换

I²C 支持 1.8V/1.2V 通信

2.UFCS 协议支持

支持 UFCS 快速握手协议

支持 UFCS 硬件复位检测

UFCS 协议通信通过 UART 通信实现

3.外设

有 6 个通用 (GPIO) 脚

1 路通用的 16 位定时器, 支持捕获功能、PWM 输出

1 路低功耗定时器

1 路独立看门狗定时器(FWDT): 8 位预置数和 12 位自减型计数器

1 个系统时间定时器: 24 位自减型计数器

1 组 UART

1 个 I²C, 支持主/从模式, 支持: 标准模式(100k bit/s), 快速模式(400k bit/s)

4. 调试口

串行调试(2 线 SWD), 与串口 RX/TX 复用, 默认为 SWD

2. EVB 介绍

2.1 开发板驱动安装

若要使用开发板上的 USB 转串口功能，则需先安装 USB 转串口驱动 Silicon Labs CP210x，也可以使用驱动精灵自动安装 USB 转串口工具。驱动安装完成后，在计算机设备上，可以看到此测试机的 COM 口。如：



图 1 COM 口检测

2.2 开发板硬件资源介绍

图 2 为开发板实物图

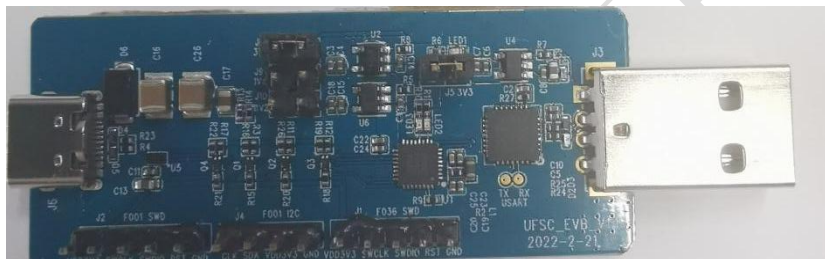


图 2 开发板实物图

根据图 3 进行各个模块分析：

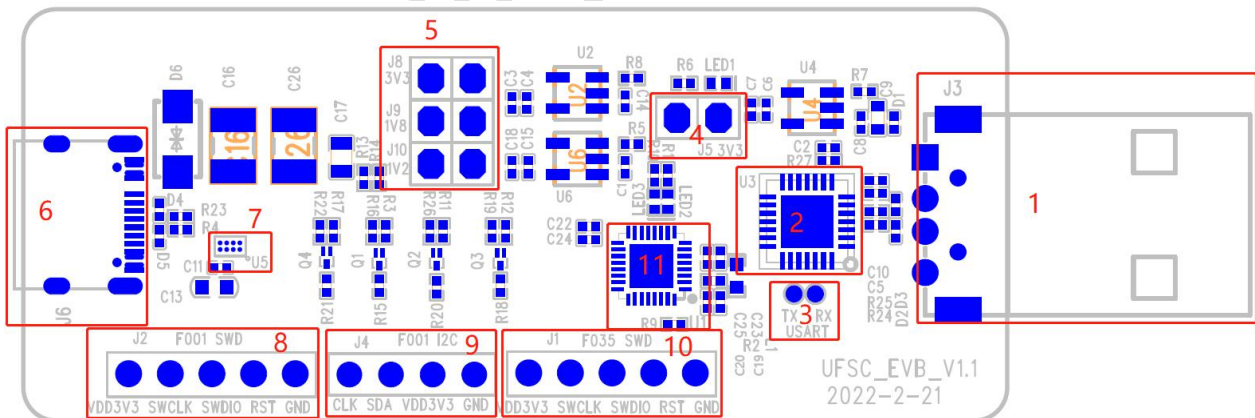


图 3 开发板

- J3 为 Type A 接口，MCU CS32F035 的 UART 经过 CP2102 转为 USB，经过 J3 可连电脑通信。
- U3 为 CP2102，将 MCU CS32F035 的 UART 转为 USB。
- MCU CS32F035 UART 信号测试点。

- J5 为板上 3.3V 供电跳线帽，板子工作需要此跳线帽。
- J8、J9、J10 为 CPW3101 I2C、RST 和 INT 电平选择（跳线帽三选一，若短接 J8，IO 上拉电源选择 3.3V；若短接 J9，IO 上拉电源选择 1.8V；若短接 J10，IO 上拉电源选择 1.2V）
- Type C 插座，用于 CPW3101 USB 通信，同时 VBUS 电源取样后送入 U1 CS32F035 芯片。
- U5 为 CPW3101 芯片。
- J2 为 CPW3101 的 SWD 接口，用于 CPW3101 调测。
- J4 为 CS32F035 的 I²C 接口，用于连接外部 I²C 从设备。
- J1 为 CS32F035 的 SWD 接口，用于 CS32F035 调测。
- U1 为 MCU CS32F035 芯片。

2.3 开发板原理图 EVB 说明

2.3.1 MCU 部分原理图

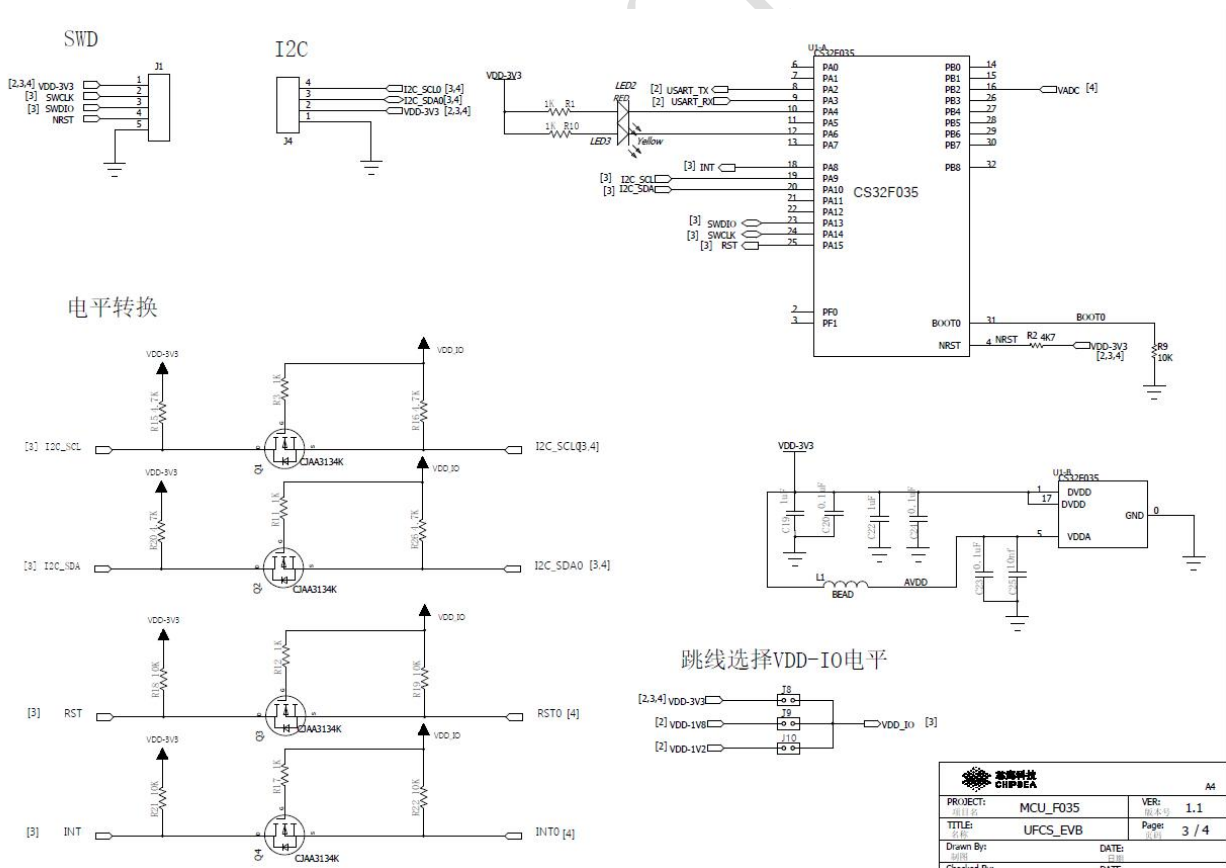


图 4 MCU 原理图

- MCU 留有 SWD 接口和 I²C 接口用于调测和外部通信。

| | | |
|-----------------|----------|-------------|
| 芯海科技 CHIPSEA | | 44 |
| PROJECT: | MCU_F035 | VER: 1.1 |
| TITLE: | UFCS_EVB | Page: 3 / 4 |
| Drawn By: | DATE: | DATE: |
| Checked By: | DATE: | DATE: |

- 电平转换电路可以保证 CPW3101 的 3.3V、1.8V 和 1.2V 电平信号与 CS32F035 的 3.3V 电平信号正常通信。
- 预留两颗 LED 指示灯，分别为红色和黄色，可用于指示芯片状态。

2.3.2 电源电路

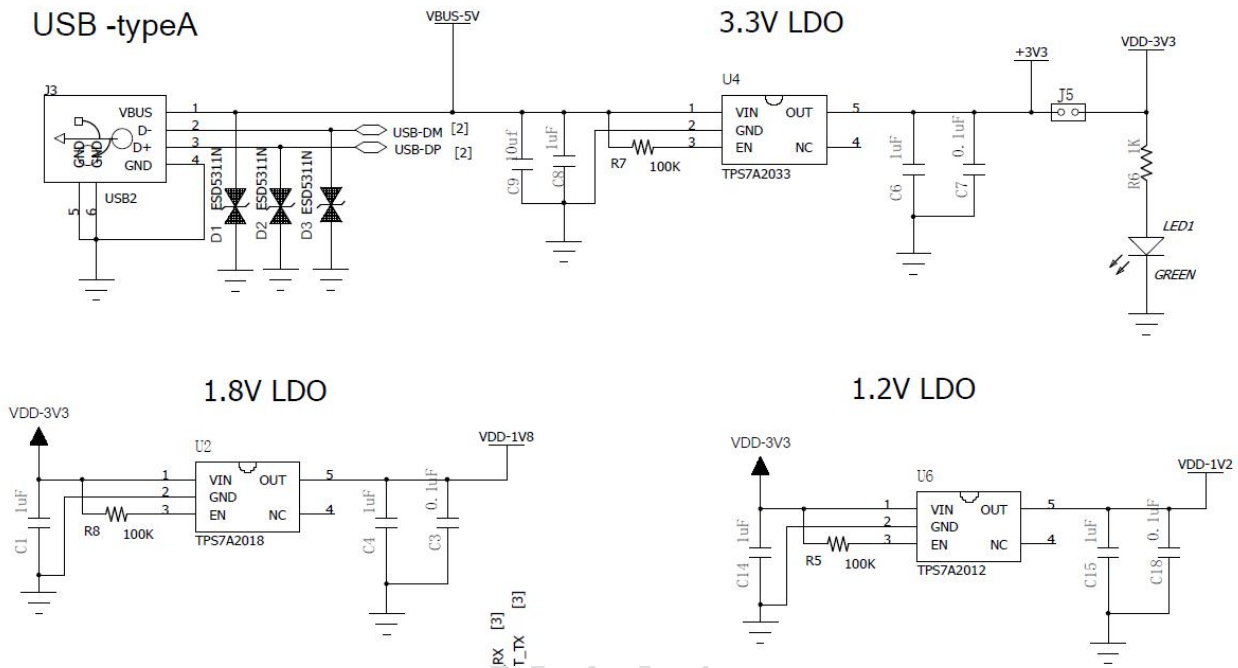


图 5 电源电路

- 电源接口配置：
- 整个 EVB 板通过 Type A USB 口供电，VBUS_5V 进入后经 LDO TPS7A2033 转换为 3.3V，VDD_3V3 给整个板子供电，电源上预留一颗绿色 LED，用于指示 VDD_3V3 电源已开启。
- VDD_3V3 经过两个 LDO TPS7A2018 和 TPS7A2012 转换为 1.8V 和 1.2V，作为 CPW3101 I²C、RESET 等接口上拉电源。

2.3.3 CPW3101 部分电路

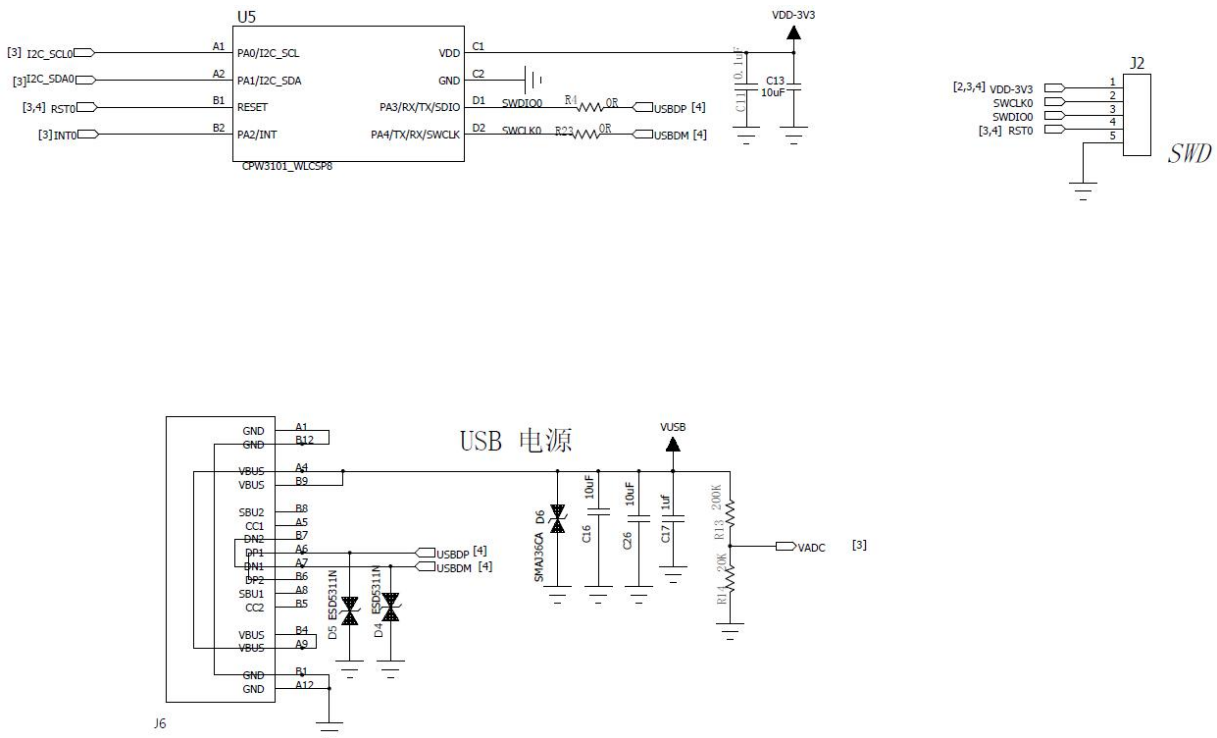


图 6 CPW3101 部分电路

- CPW3101 留有 USB 接口，通过 type C 插座连接。
- CPW3101 留有 SWD 接口用于调测。
- Type C 接口进来的电源 VUSB 经过分压电阻后进入 CS32F035 的 ADC。

2.3.4 UART 转 USB 部分电路

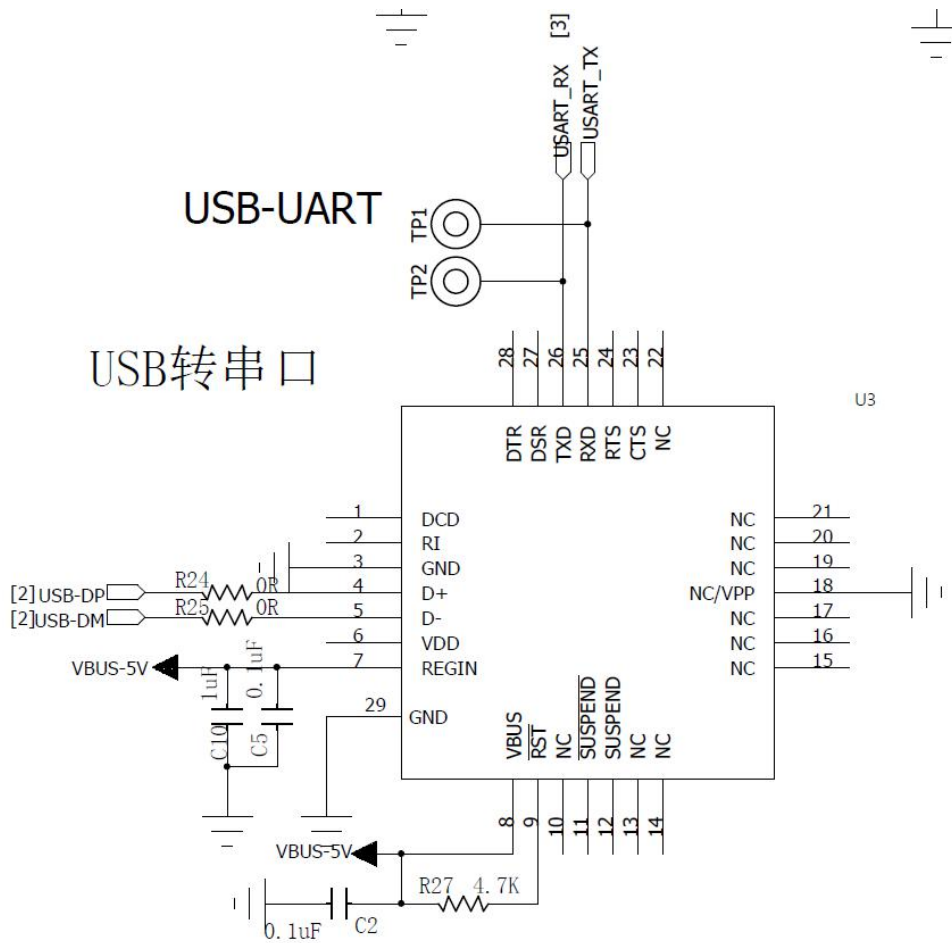


图 7 uart 转 usb 部分电路

- CS32F035 的 UART 信号经 CP2102 转为 USB 信号，通过 type A 插头可连接电脑通信。

3. IAP 固件升级

后续将详细介绍 CPW3101 芯片 IAP 固件升级程序的使用，详细介绍固件升级的硬件接口，协议命令与升级流程。升级协议包含的特性如下：

- 适用于 CPW3101 芯片的应用代码升级，29KB（0x08000C00 - 0x08007FFF）
- I²C 通信接口，地址 0x50（7bit），最高支持 400KHz 通信速率
- 通过发送升级命令进入升级模式，执行 FLASH 擦写
- FLASH 页擦除，单页 1024 字节
- 单次编程最大长度 128 字节（必须 4 字节对齐）
- 采用 CRC32 对升级固件进行校验
- IAP 程序占用 CPW3101 3KB FLASH 空间（0x08000000 - 0x08000BFF）
- 上电/复位超时跳转应用代码，超时时间为 1s

4. 硬件接口

4.1 通信接口

CPW3101 MCU 的固件升级程序采用 I²C 通信接口，地址 0x50，速率最高 400KHz，所使用到的 I²C 相应引脚如错误!未找到引用源。所示。

表 1 I²C 引脚复用

| 通信接口 | SCL 引脚 | SDA 引脚 |
|------------------|--------|--------|
| I ² C | PA0 | PA1 |

主机写时序

| | | | | | | | | | | |
|---|-----|---|---|------|---|------|---|------|---|---|
| S | ADD | W | A | D[0] | A | D[1] | A | D[n] | A | P |
|---|-----|---|---|------|---|------|---|------|---|---|

主机读时序

| | | | | | | | | | |
|---|-----|---|------|---|------|---|------|----|---|
| S | ADD | R | D[0] | A | D[1] | A | D[n] | NA | P |
|---|-----|---|------|---|------|---|------|----|---|

4.2 硬件连接

进行 I²C 接口固件升级时，上位机与待升级 MCU 的 I²C SCL/SDA 连接，进行命令数据包的发送，以及响应数据包的接收。同时，主机需要能够改变 MCU 复位引脚（NRST）的电平，控制 MCU 在芯片复位后 1s 钟内写入升级命令，以启动位于系统存储区固件升级程序。硬件连接如错误!未找到引用源。所示。

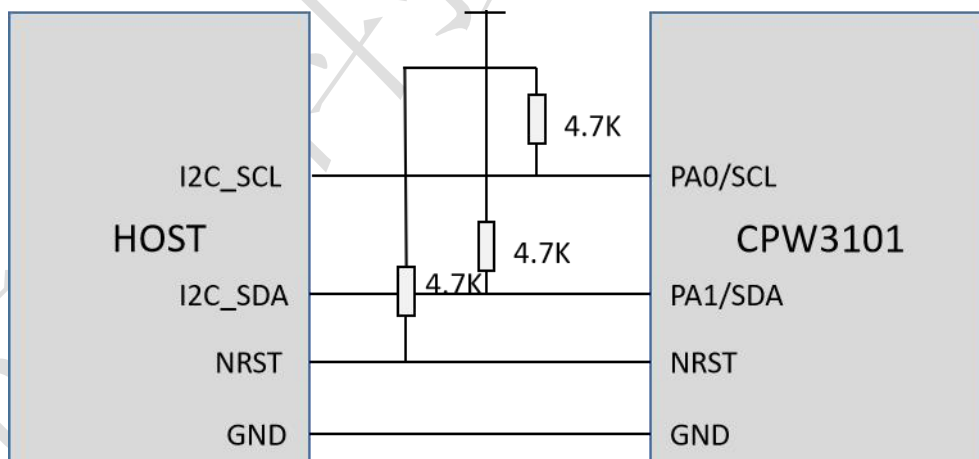


图 8 硬件连接示例

4.3 进入升级模式

CPW3101 芯片 FLASH 空间为 32KB (0x08000000-0x08007FFF)，前 3KB 为 IAP 程序占用，后 29KB 为应用代码空间。SRAM 空间为 2KB (0x20000000-0x200007FF)，0x20000000-0x200000B7 为中断向量表占用，应用代码需要避开。芯片上电或复位后从 FLASH 0x08000000 地址处启动，首先进入 IAP 程序。在 IAP 程序中将超时等待升级命令，升级完成或者直接发送跳转命令均可使芯片从 (0x0800 0000) 跳转至 APP 地址执行 (0x0800 C000)。

5. 升级流程

5.1 固件下载流程

烧入 BOOT 程序以后，按上述要求完成配置。芯片复位以后，开始计时，1 秒钟内未收到固件升级指令，便自动跳转至 APP 区，若在 1 秒钟收到升级固件命令，则停止计数，后续操作是获取固件 BOOT 代码版本号，擦除 FLASH 指定页，写入固件，计算写入固件的 CRC 校验值，并与上位机计算出的 CRC 校验值进行对比，最后跳转，完成全部升级流程。

上述流程开始前需先完成 FLASH, I²C, SYSTICK 等初始化配置。I²C 配置完成后，固件升级程序将循环等待上位机串口命令数据包，如果命令数据包校验无误，则执行相应命令并返回响应数据包，否则返回包含数据包错误信息的响应数据包。具体流程见错误!未找到引用源。。

若用户需要将固件下载到芯片 Flash，需要按顺序执行擦除、写存储、CRC 比较等命令。待下载结束，可根据需要，设定 Flash 读写保护级别。

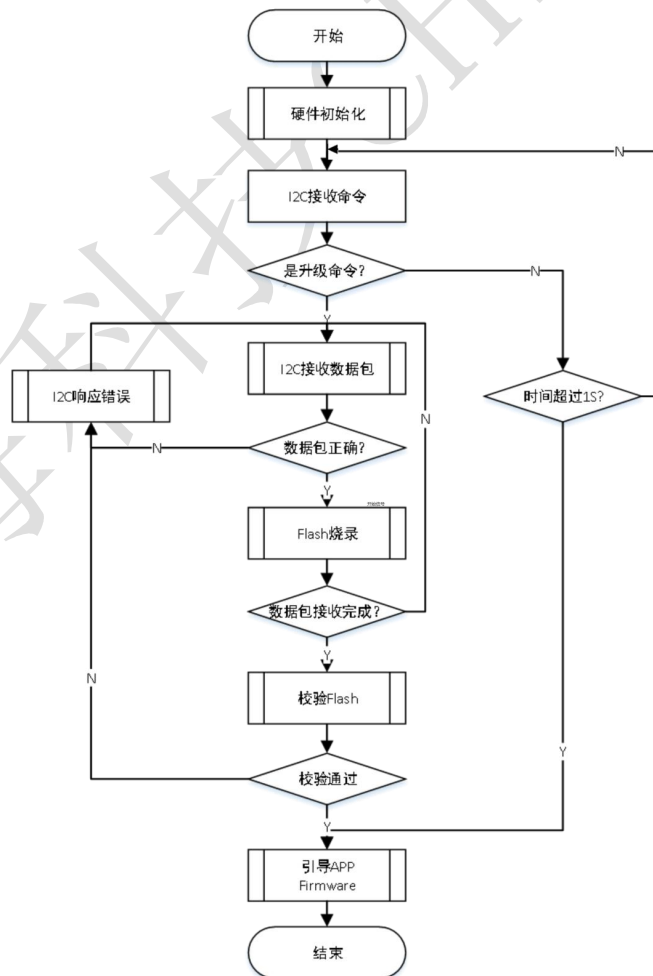


图 9 固件升级程序流程

5.2 跳转流程

CPW3101 芯片跳转流程分 4 步：

- 将 APP 地址（0x0800 0C00）中断向量表复制到 SRAM（0x2000 0000）处
- 设置中断向量表映射到 SRAM 处
- 将 SP 指向 APP 的堆栈栈顶（0x0800 0C00）处
- 将 PC 指向 APP 的复位向量（0x0800 0C04）处，实现 BOOT 跳转到 APP

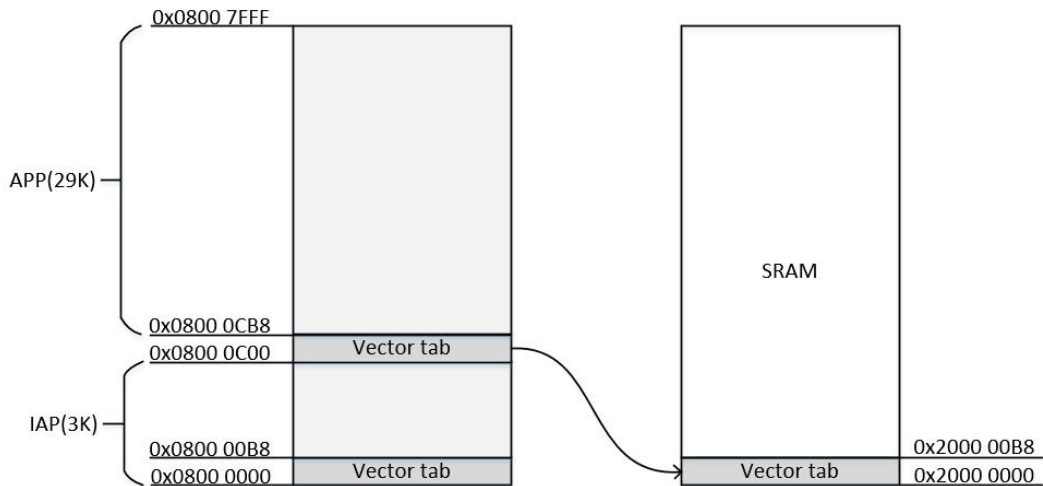


图 10 IAP 和应用代码存储空间划分

在实现此部分内容之前，需要先对 RAM 进行分区，分区的意义在于防止向量表被破坏。具体操作如下。

打开 KEIL，选择 Option for Target -> Target, 对 RAM 进行分区。如下图右栏所示。

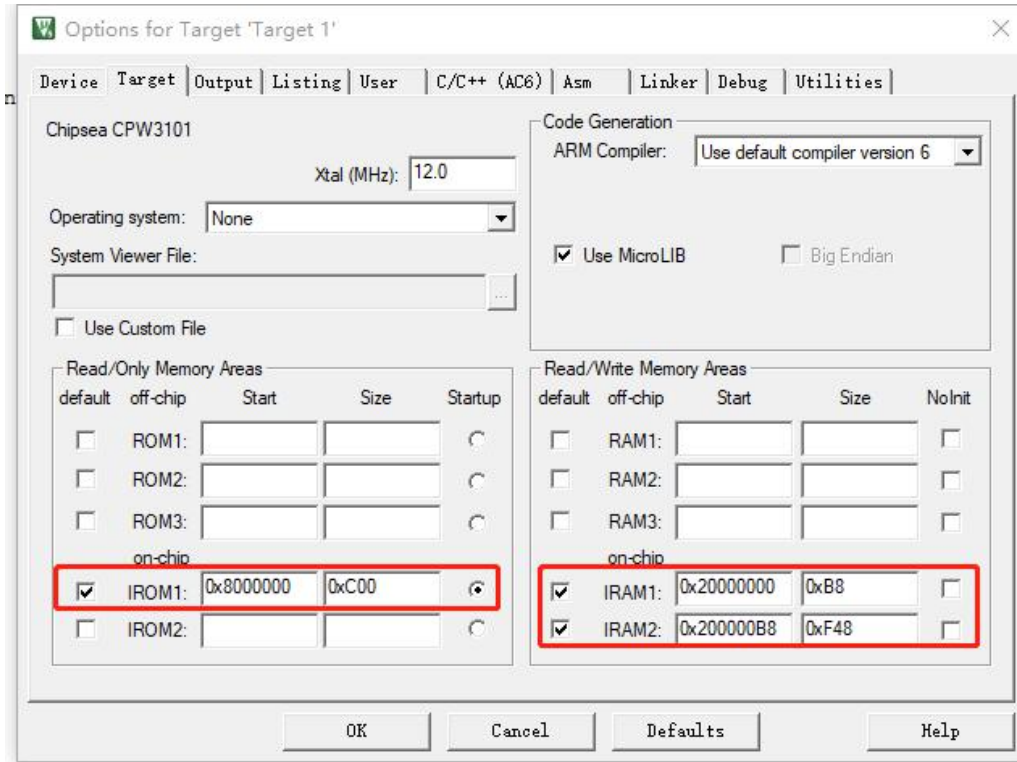


图 11 IAP 程序 Memory 空间占用

其中，0xB8 是向量表大小，0xF48 是 2KB RAM 剩余大小，应用程序需要避免占用 SRAM 前 0xB8 字节。前面 default 选择勾选。左栏中选择从 0x0800 0000 启动，0xC00 为 IAP 空间大小。

6. 命令协议

CPW3101 芯片固件升级程序，采用基于命令接收与处理的升级流程，数据包分为命令包及对应返回包。命令和返回包采用带校验的不定长数据包格式，从机会检查接收数据包的帧头、长度、校验，确保数据包的完整性和正确性，防止通信异常或误码导致的错误命令被执行。同时，上位机发送命令后，需要检查响应包，确定命令包被正确接收和执行，识别失败错误情形，执行命令重发，可以有效提高命令协议的可靠性和鲁棒性。

6.1 通信协议

CPW3101 的 IIC 通信协议如下：

- 主机写数据格式：

0xB0+0x01+命令包

其中，B0 为 CPW3101 的器件地址，01 为固定数据，命令为 IAP 各类命令数据包。

- 主机读数据格式：

0xB1+0x01+命令包

其中，B1 为 CPW3101 的器件地址，01 为固定数据，命令为 IAP 各类命令数据包。

6.2 命令包格式

6.2.1 命令数据包

| Header | Length | Command Area | Checksum |
|--------|--------|--------------|----------|
| 0xAA | N | CMD + DATA | SUN 8 |

其中，Length 指示 Command Area 区域数据长度；命令数据包校验采用和校验，对 Length 和 Command Area 区域执行和校验计算，计算结果附在数据包末尾 Checksum 字节。

Command Area 区域包含 1 字节命令码以及命令参数，如页擦除命令（Erase Page）的命令参数指示 32-bit 待擦除页首地址，某些命令不包含参数。

6.2.2 响应数据包

| Header | Length | Response Area | | Checksum | |
|--------|--------|---------------|------------|----------|-------|
| 0xAA | N | CMD | Error Code | DATA | SUN 8 |

其中，Length 指示 Command Area 区域数据和 Response Area 区域数据长度；CMD 为当前接收到的命令的命令码；Error Code 表示命令接收执行情况；DATA 为命令返回数据，根据具体命令不同，有或者无。响应数据包与命令数据包采用相同校验方式，对 Response

Area 区域执行校验和计算，计算结果附在数据包末尾 Checksum 字节。

| Error Code | 描述 | 备注 |
|------------|-------------|----|
| 0x03 | 数据包接收或者执行失败 | |
| 0x08 | 数据包接收并且执行成功 | |

6.3 命令详解

6.3.1 命令列表

命令定义采用命令结构体数组，支持方便地扩展或裁剪。

| 命令 | 命令码 | 描述 | 返回包 | 备注 |
|-------------|------|-------------------|-----|---|
| CMD_UPGRADE | 0xF0 | 升级命令 | 有 | 上电/复位后 1s 内写入，否则跳转。 |
| CMD_VERSION | 0xF5 | 获取固件升级程序版本号 | 有 | |
| CMD_PROGRAM | 0xF2 | 向 Flash/SRAM 区写数据 | 有 | 4 字节对齐，最大 128 字节，执行 CMD_UPGRADE 命令后方可执行 |
| CMD_CRC | 0xF3 | 对已下载固件执行 CRC 校验 | 有 | 校验区域为 CMD_ERASE 命令指定擦除的页（1024 字节对齐，FLASH 未写区域为 0xFF），执行 CMD_UPGRADE 命令后方可执行 |
| CMD_ERASE | 0xF1 | 擦除多页（1KB/页） | 有 | 执行 CMD_UPGRADE 命令后方可执行 |
| CMD_JUMP | 0xF4 | 强制跳转到应用程序 | 有 | 成功跳转无返回 |

6.3.2 CMD_UPGRADE 命令

命令描述 写入固件升级指令

命令示例

| Header | Length | CMD | Checksum |
|--------|--------|------|----------|
| 0xAA | 0x01 | 0xF0 | 0xF1 |

命令响应示例

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF0 | 0x08/0x03 | SUN 8 |

其中，Response Area = 0x08 指示成功进入 BOOT 模式，Response Area = 0x03 指示进入 BOOT 模式失败。

6.3.3 CMD_VERSION 命令

命令描述 获取固件升级程序版本号 Version（32bit）

命令示例

| Header | Length | CMD | Checksum |
|--------|--------|------|----------|
| 0xAA | 0x01 | 0xF5 | 0xF6 |

命令响应示例 - 成功

| Header | Length | CMD | Response Area | DATA[0]~DATA[3] | Checksum |
|--------|--------|------|---------------|-----------------|----------|
| 0xAA | 0x06 | 0xF5 | 0x08 | Version[0:31] | SUM_8 |

命令响应示例 - 失败

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF5 | 0x03 | 0xFA |

其中，Version[0:31]为返回的 32-bit 版本号，低字节在前。

6.3.4 CMD_ERASE 命令

命令描述 根据指定页首地址，擦除该地址及之后共 Page_num 数目的页（1K/页）。下载代码到 Flash 某页，需要首先执行擦除操作。页擦除时间为 23ms(max)，上位机需要延时等待擦除命令执行完成，接收到响应数据包，才能执行下一个命令。

命令示例

| Header | Length | CMD | DATA[0]~DATA[3] | Page_num[L] | Page_num[H] | Checksum |
|--------|--------|------|-----------------|-------------|-------------|----------|
| 0xAA | 0x07 | 0xF1 | ADDR[0:31] | Page[L] | Page[H] | SUM_8 |

其中，ADDR[0:31]为指定的 32-bit Flash 页地址，低字节在前。

命令响应示例

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF1 | 0x08/0x03 | SUN_8 |

其中，Response Area = 0x08 指示擦除执行成功，Response Area = 0x03 指示擦除执行失败。

6.3.5 CMD_PROGRAM 命令

命令描述 向指定地址写入指定数目数据，执行对 Flash 或者 SRAM 的编程。

命令示例

| Header | Length | CMD | DATA[0]~DATA[3] | NUM[0]-NUM[15] | DATA[4]~DATA[N+3] | Checksum |
|--------|--------|------|-----------------|----------------|-------------------|----------|
| 0xAA | N + 7 | 0xF2 | ADDR[0:31] | N | CODE[0]~CODE[N-1] | SUN_8 |

其中，ADDR[0:31]为指定的 32-bit 待写入地址，低字节在前；N 指示待写入有效数据长度，需要按 4 字节对齐，最大单次写入 128 字节；CODE[0]~CODE[N-1]为待写入 Flash 的代码，低字节在前。

命令响应示例

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF2 | 0x08/0x03 | SUN_8 |

其中，Response Area = 0x08 指示写操作成功，Response Area = 0x03 指示写操作失败。

6.3.6 CMD_CRC 命令

命令描述 回读升级固件并计算 CRC32 校验值，与上位机计算的的校验值比对。

命令示例

| Header | Length | CMD | CRC[0]~CRC[3] | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x05 | 0xF3 | CRC[0:31] | SUM_8 |

其中，CRC[0:31]为上位机计算的 CRC32 校验值。

命令响应示例

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF3 | 0x08/0x03 | SUN_8 |

其中，Response Area = 0x08 指示固件校验成功，Response Area = 0x03 指示固件校验失败。

CRC 校验采用多项式 0x04C11DB7，初始值 0xFFFFFFFF，异或值 0x00000000，输入输出不翻转，校验结果低字节在前。其中，CRC32 校验值是从 CMD_ERASE 命令指示的代码升级区域进行回读并迭代计算得到的。

6.3.7 CMD_JUMP 命令

命令描述 强制 CPU 跳转应用代码执行。

命令示例

| Header | Length | CMD | Checksum |
|--------|--------|------|----------|
| 0xAA | 0x01 | 0xF4 | 0xF5 |

命令响应示例 - 成功

程序已强制跳转，无响应数据包。

命令响应示例 - 失败

| Header | Length | CMD | Response Area | Checksum |
|--------|--------|------|---------------|----------|
| 0xAA | 0x02 | 0xF4 | 0x03 | 0xF9 |

7. 时序特性

CPW3101 I²C 固件升级的关键时序特性如下表所示，主机连续发送命令时需要考虑命令执行耗时，命令执行结束通过 I²C 回读的返回数据包才是有效的。在命令执行未结束期间，通过 I²C 读返回数据包，将会触发 CPW3101 的 I²C slave clock stretching 特性，拉低 SCL，主机需要支持 clock stretching 的处理或者保证足够 I²C 读写间延时。

| 描述 | 最小值 | 典型值 | 最大值 | 备注 |
|-----------------------------|-----|--------|-----|------------------|
| 上电/RESET 后进入命令接收状态的时间 | | 189us | | |
| 上电/复位后从 IAP 超时跳转应用时间 | | 1s | | 无 CMD_UPGRADE 命令 |
| CMD_PROGRAM 命令 FLASH 写入执行时间 | - | 5.41ms | - | 128 字节时间 |
| CMD_CRC 命令回读和 CRC 计算执行时间 | - | 566us | - | 128 字节回读和计算时间 |
| CMD_ERASE 命令页擦除执行时间 | - | 5.16ms | - | 单页 1024 字节时间 |

8. keil 工程搭建

- 1、安装 keil pack 包<Chipsea.CPW3101_DFP.1.0.1.pack>
- 2、打开 keil，创建新工程，设备选择“CPW3101”，见图 12

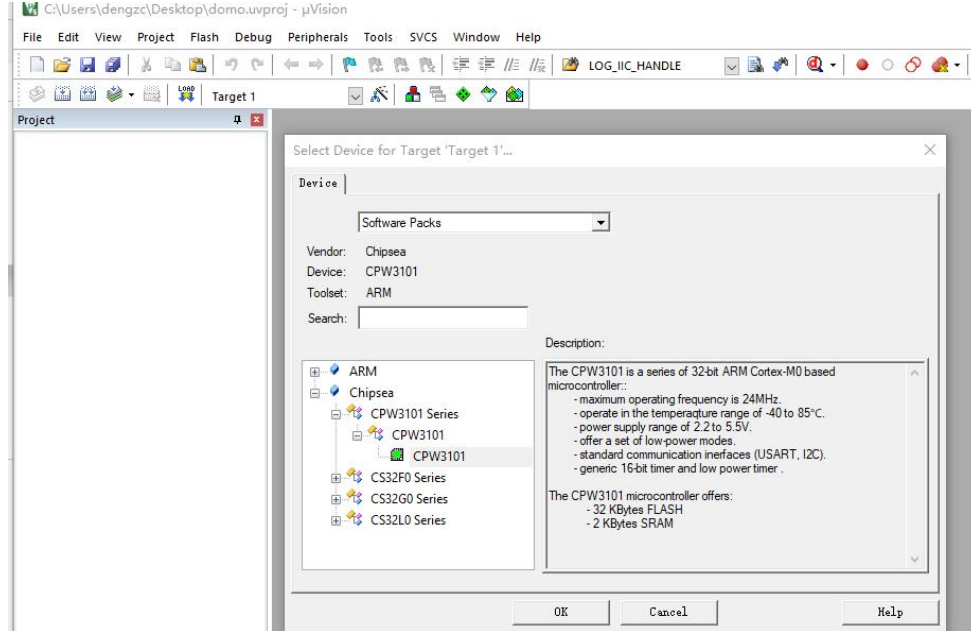


图 12 设备选择

- 3、芯片选择完成后，在<Manage Run-Time Environment>中，勾选 CORE、Startup 以及 HAL 下的所有外设。

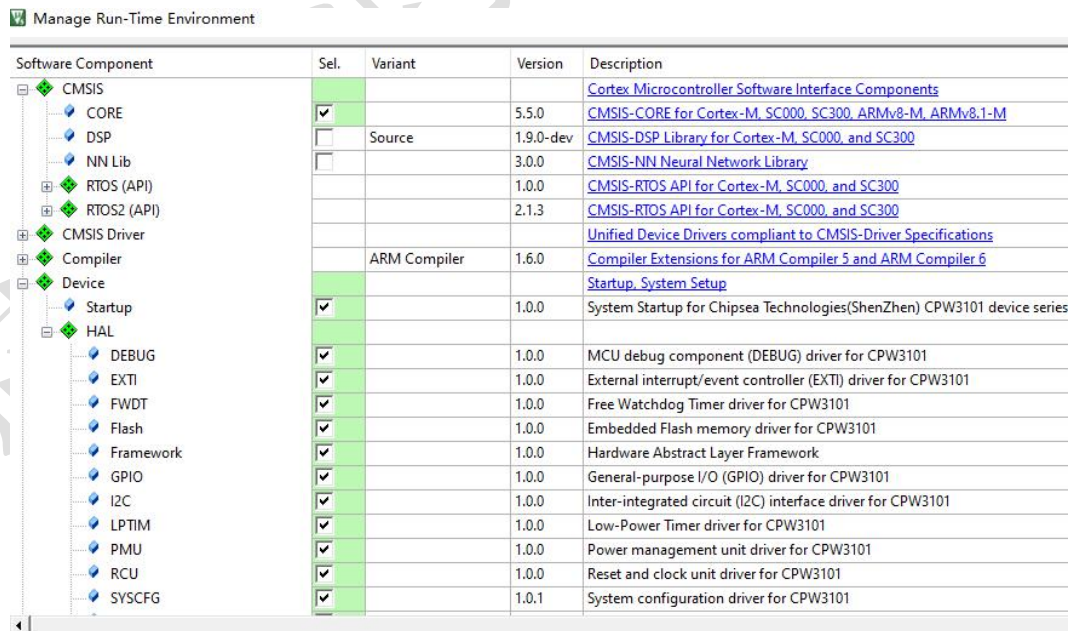


图 13 HAL 外设选择

4、在工程配置中，勾选“Use MicroLib”

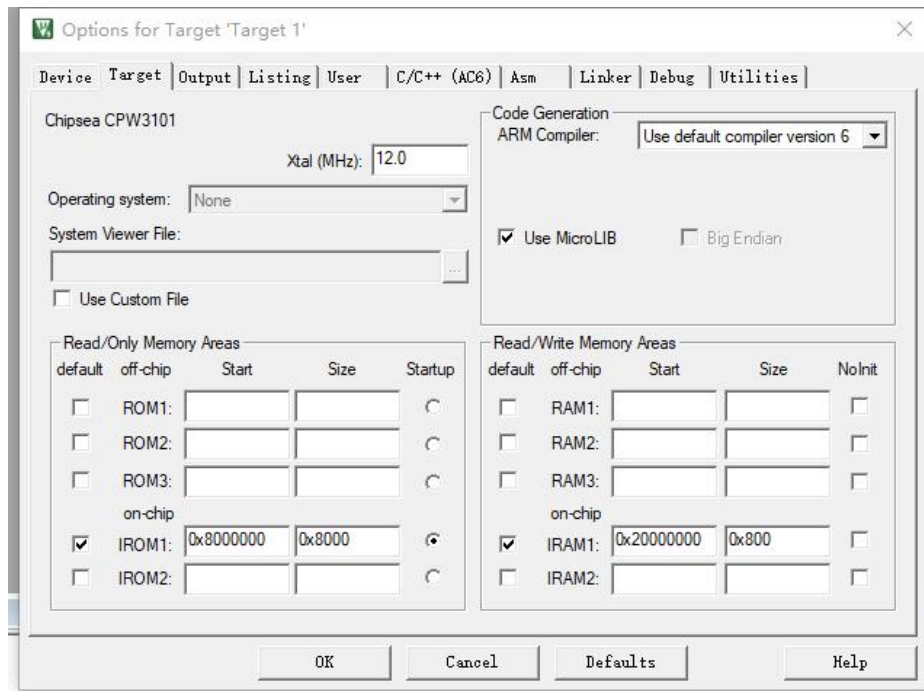


图 14 工程配置

9. 仿真调试

CPW3101 支持 SW 调试端口。用户可以使用 Jlink 工具，进行仿真和调试。

1、硬件接口

将 Jink 的 SW 接口对应连接至开发板 J2 的 SWD 和 SDCLK 端口。

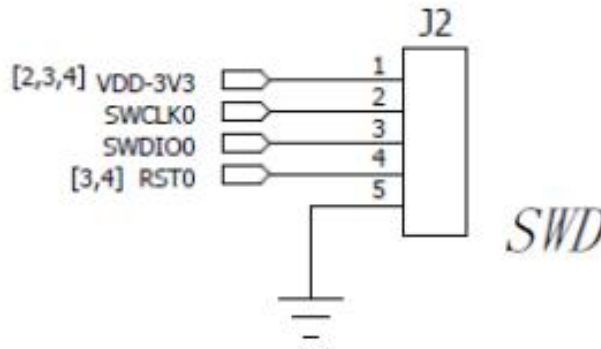


图 15 SW 连接

2、keil 工程配置的<Debug>中，使用“J-LINK/J-TRACE Cortex”

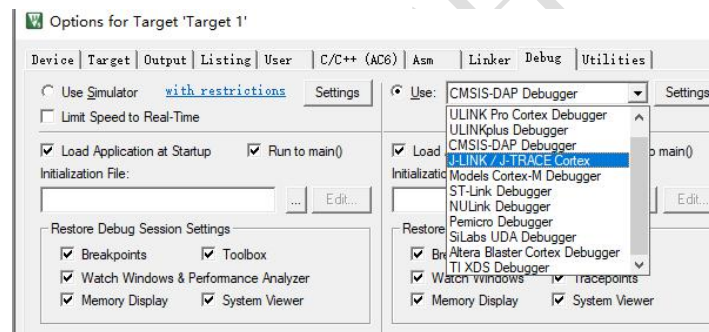


图 16 JLINK 仿真器选择

3、在 Jlink 的 Device 选择中，选择“Unspecified Cortex-M0”

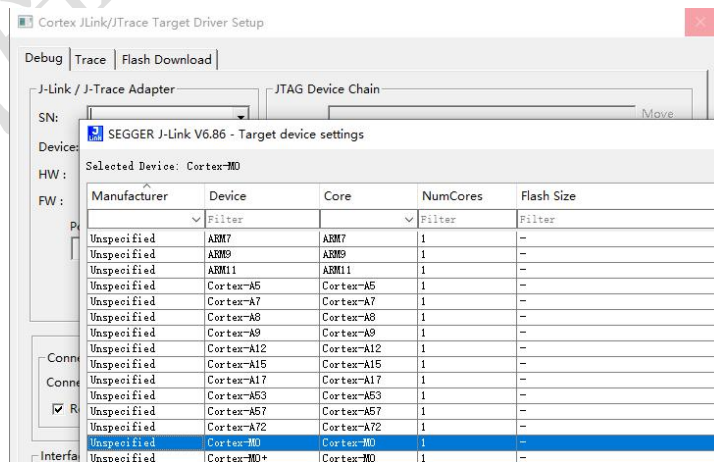


图 17 JLINK 设备选择

4、仿真端口 Port 选择 “SW”

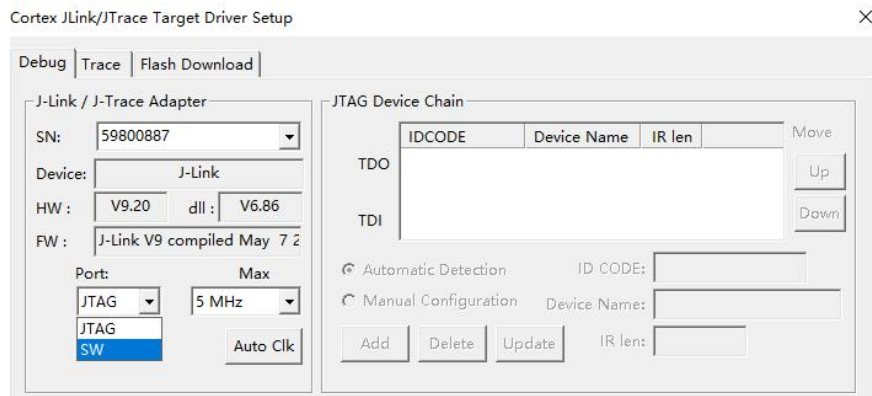


图 18 SW 模式选择

5、配置完成了，即可使用 keil 自带的 debug 功能。支持 full memory access、system register access 以及 watch point、break point、step run 等功能。

10. 注意事项

1、PA3、PA4 端口默认为 SWDIO 以及 SWCLK 功能，因此在初始化时，PA3 默认上拉高电平，PA4 默认下拉低电平。

2、PA3、PA4 端口默认为 SWDIO 以及 SWCLK 功能，若将这两个端口在应用时复用普通 GPIO 或者 uart 功能，则无法继续使用 SW 调试功能。

若端口复用以后，仍需使用 SW 的 DownLoad 功能，则需将 keil 的 Debugger 配置为 Under nreset。并将 Jlink 的 NRESET 引脚连接至 CPW3101 的 NREST 引脚。

Keil 的配置如下：

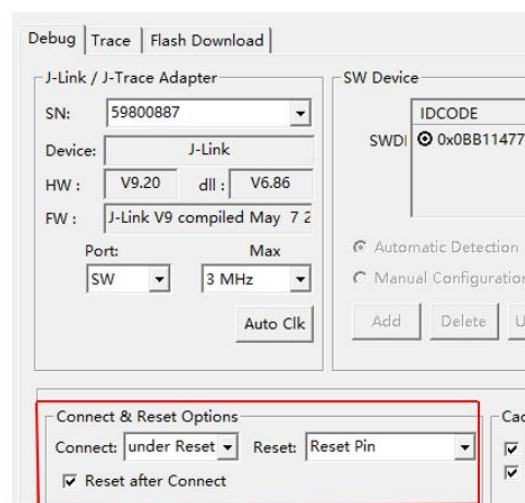


图 19 keil 配置

3、PA0、PA1、PA2 口为开漏输出口，不支持推挽输出，没有上拉电阻和下拉电阻，没有驱动能力。

4、ufcs 外设时钟必须配置为 500K，否则会导致 ufcs 外设无法正常接收、发送 ufcs 硬件信号，如复位脉冲、握手脉冲等。

5、ufcs 外设接收复位信号的精度为 250us。使用时，需保证外部复位信号符合协议要求，且通信线路对信号的畸变在可控范围内。

6、ufcs 外设通过 D-端口发送完握手脉冲后，会自动将 D-的控制权交给 uart 外设，此时 D-端口会被 uart 外设以外拉高。

7、ufcs 握手脉冲序列必须通过复位 ufcs 外设才能停止。

8、LPTIM1 为低功耗时钟，在使用 LRC 为时钟源时，该时钟可以在低功耗下继续运行。该时钟的更新中断为“Autoreload match Interrupt”

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2023 芯海科技（深圳）股份有限公司，保留所有权利。



芯海科技
CHIPSEA

股票代码:688595