



专注于高端实时控制芯片设计

# Xplorer-GS32F37X\_38X\_075 系列 生态板用户手册

2025 年 05 月 20 日

格见构知（上海）半导体有限公司 | 深圳格见构知半导体有限公司（简称“格见半导体”）是一家专注于高端实时控制芯片设计公司，在芯片产品定义、设计研发、量产导入、销售运营等领域都具备丰富经验。

GS32-DSP 是格见半导体自主研发的实时控制微处理器系列产品。公司致力于为数字能源、数字电源、工业自动化、智能汽车、机器人、高端家电等领域提供芯片解决方案。

更多产品资料: [sales@gejian-semi.com](mailto:sales@gejian-semi.com); [support@gejian-semi.com](mailto:support@gejian-semi.com)

更多详情访问: [www.gejian-semi.com](http://www.gejian-semi.com)

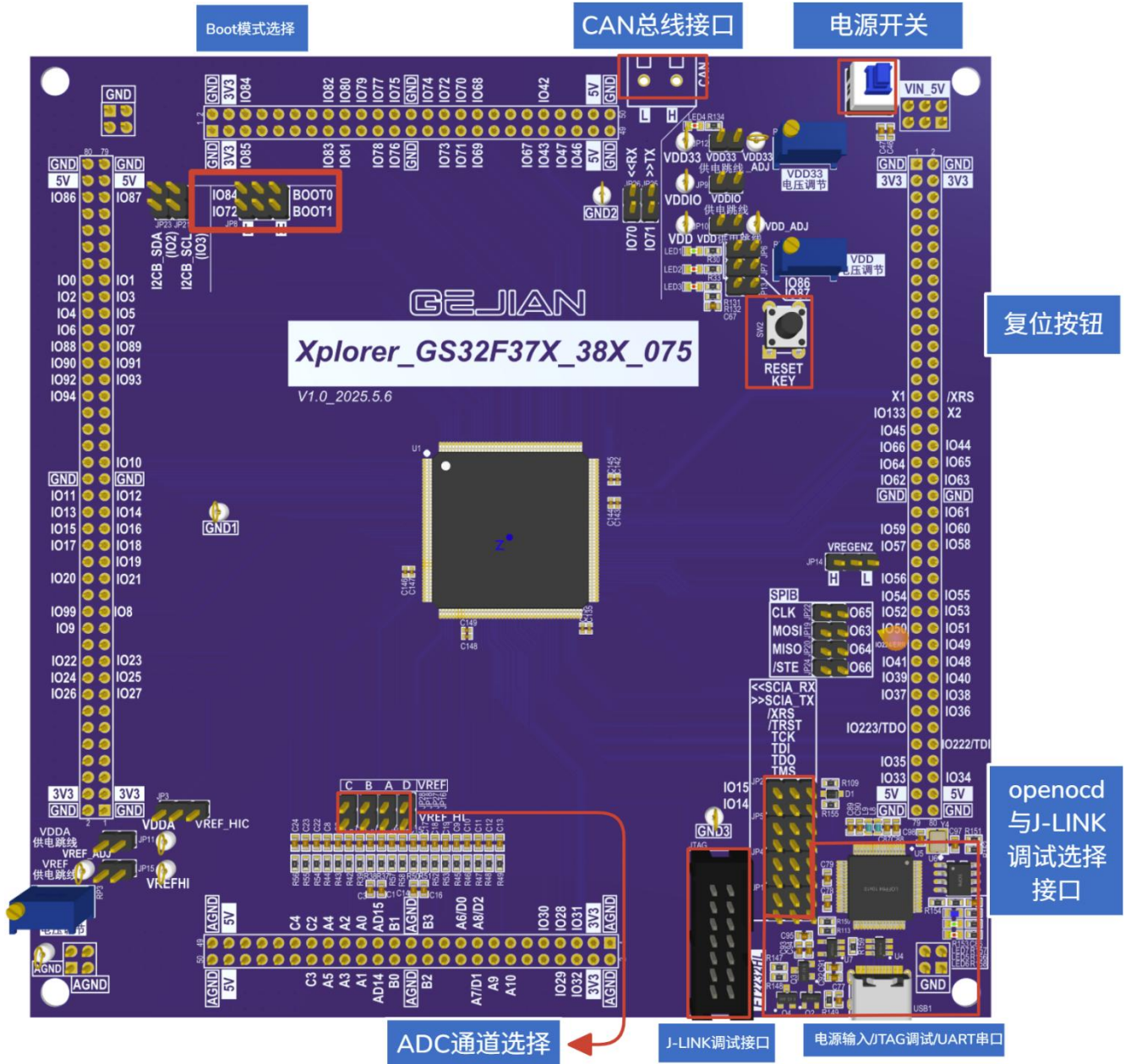


更多官方资讯

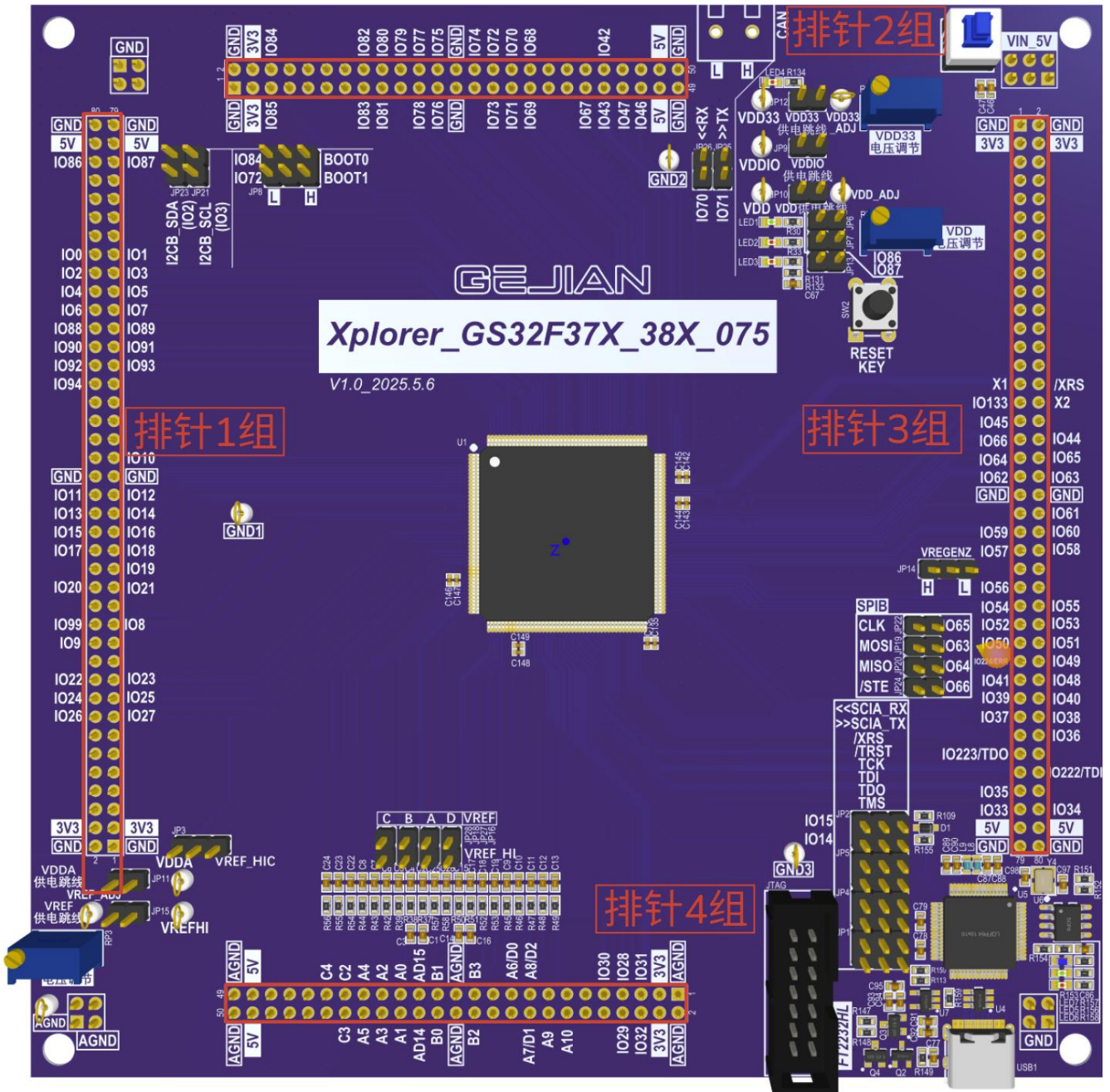
## 目录

1 总体框图 .....	1
2 排针组图 .....	2
2.1 引脚定义图 .....	3
3 功能介绍 .....	7
3.1 open OCD 或 J-link 调试 (跳线帽选择) .....	7
3.2 使用 type-c 供电和外接供电 (JP12 跳线帽选择) .....	8
3.3 VDDIO 供电跳线 (给电路中的输入/输出部分 (I/O) 提供的电压) .....	9
3.4 VDD 跳线 (芯片内部数字电路供电) .....	9
3.5 VDDA 跳线 (芯片内部模拟电路供电) .....	10
3.6 VREGENZ 跳线 .....	11
3.7 BOOT 方式选择(芯片的启动方式).....	12
3.8 ADC 测试引脚连接 .....	13
3.9 晶振 .....	14
3.10 I2C 引脚选择 .....	16
3.11 CAN 引脚选择 .....	17
3.12 SPI 管脚连接 .....	17
3.13 LED 引脚连接 .....	18
4 基本操作 .....	19
4.1 上电与连接 .....	19
4.2 复位 .....	19
4.3 常见问题 .....	19
4.3.1 板载 openocd .....	19
4.3.2 J-LINK 连接 .....	20
5 修订历史 .....	21

1 总体框图



2 排针组图







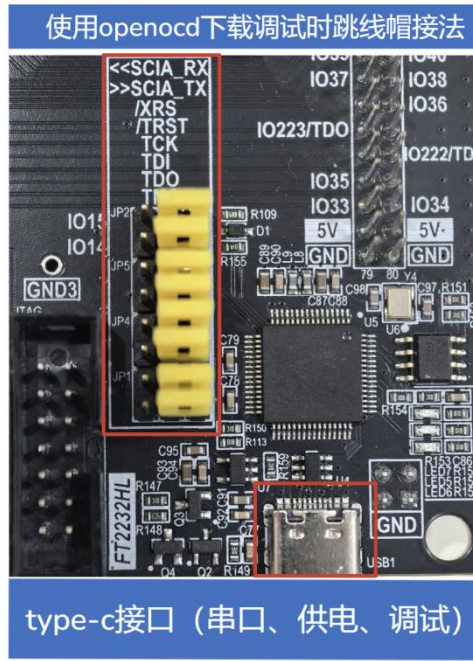
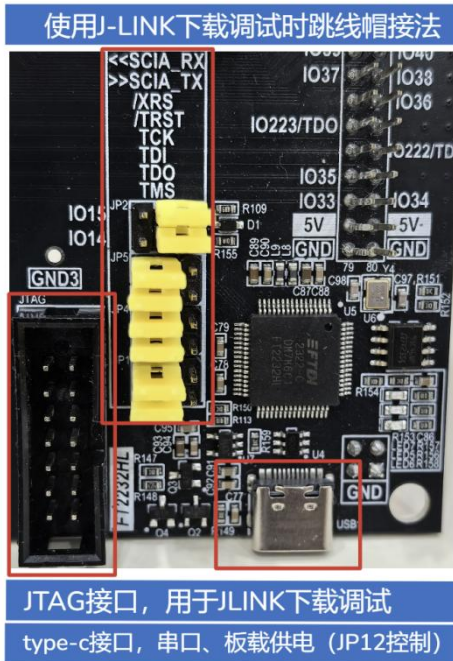




### 3 功能介绍

在使用生态板时，需要先将硬件的管脚通过跳线帽连接，以便正常调试。

#### 3.1 open OCD 或 J-link 调试 (跳线帽选择)



### 3.2 使用 type-c 供电和外接供电 (JP12 跳线帽选择)

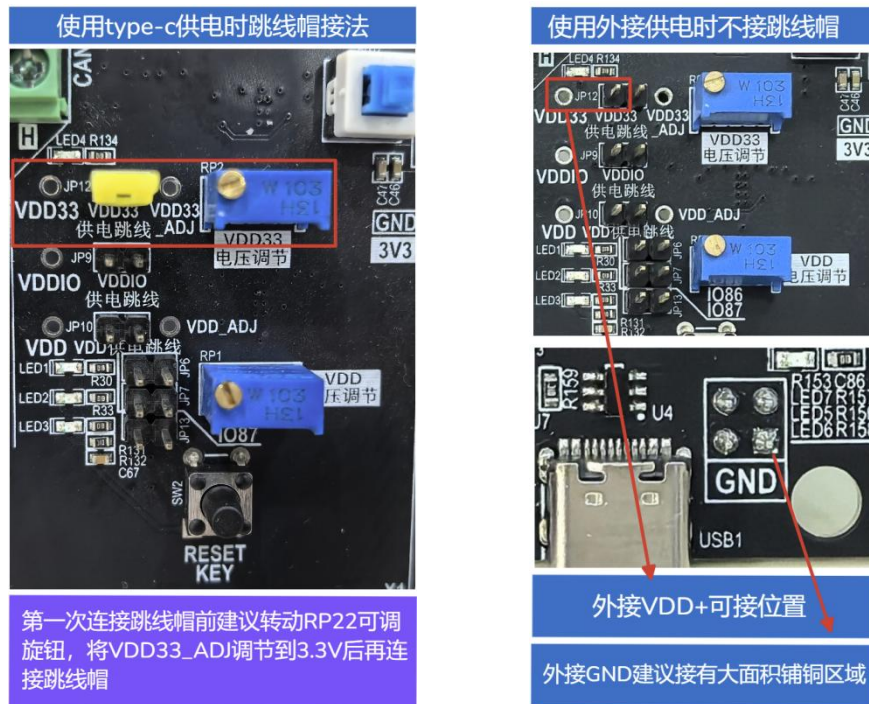
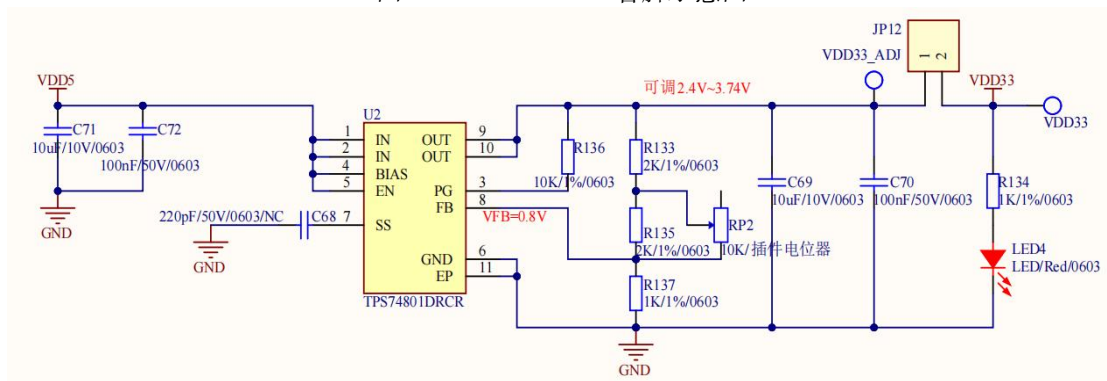


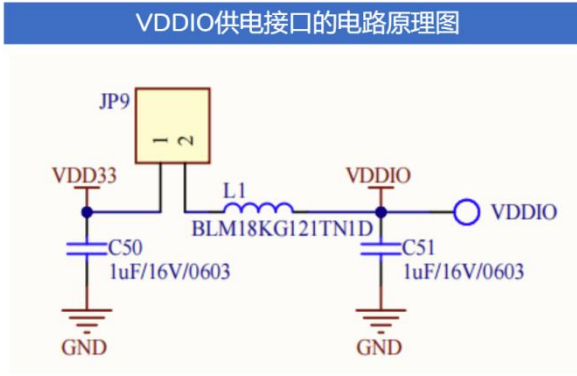
图 2-2-1 JP12 PCB 管脚示意图



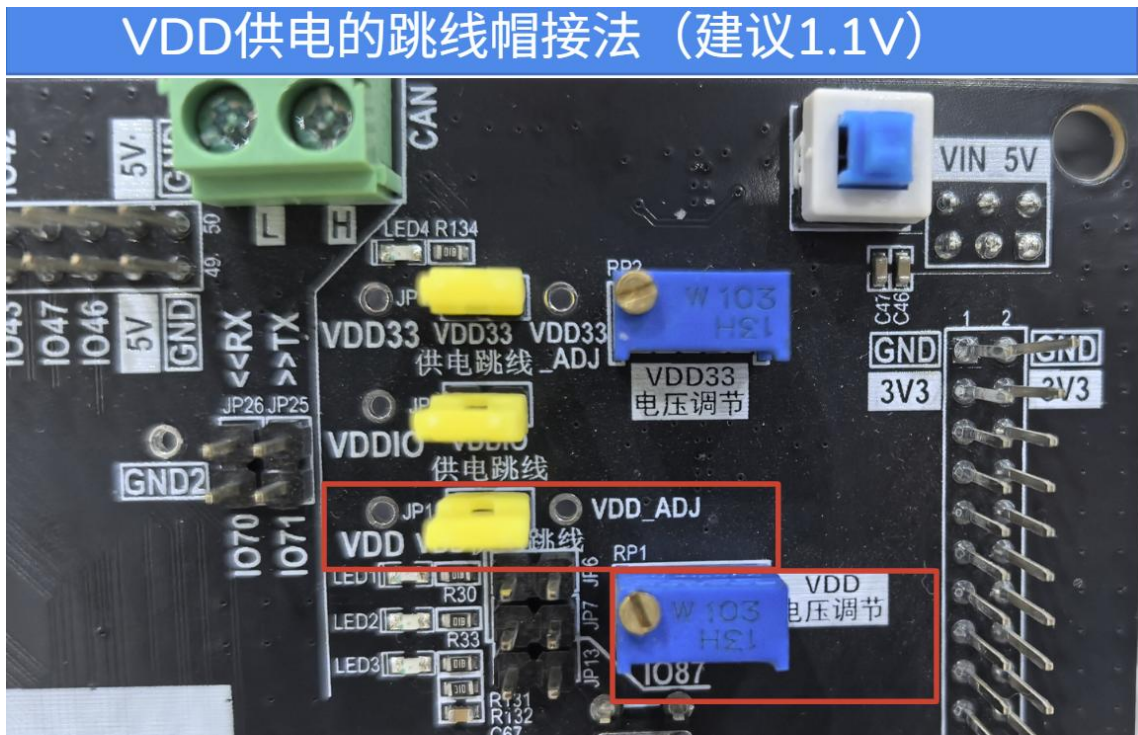
JP12 硬件原理图

JP12 跳线端子	生态板供电
连通	USB (type-C) 供电
断开	外部供电 (需要自行连接供电端口)

### 3.3 VDDIO 供电跳线 (给电路中的输入/输出部分 (I/O) 提供的电压)

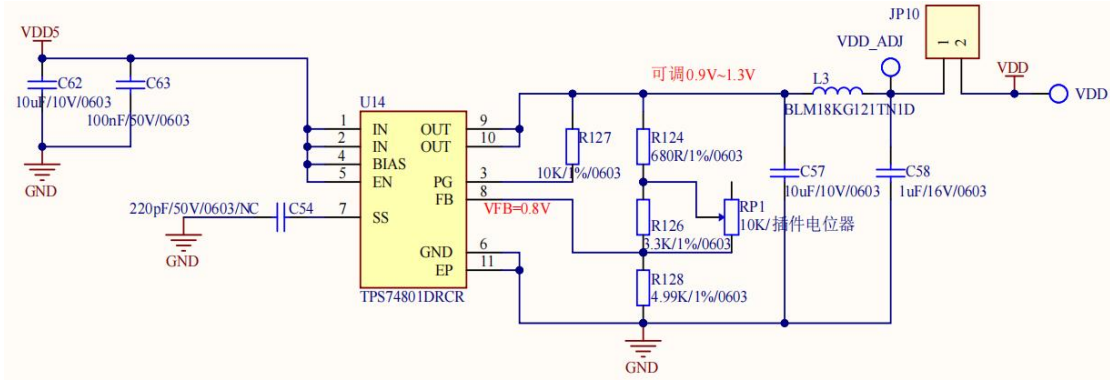


### 3.4 VDD 跳线 (芯片内部数字电路供电)



第一次连接跳线帽前建议转动RP1可调旋钮，将VDD\_ADJ调节到1.1V后再连接跳线帽(建议在1.21V内)

JP10 PCB 管脚示意图



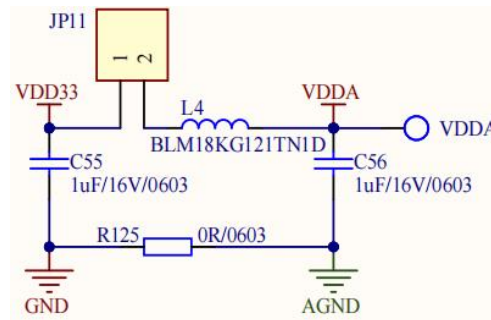
3.4 图一 JP10 硬件原理图

图为硬件原理图。其作用是给电路中的数字电路部分提供的主电源电压。

### 3.5 VDDA 跳线 (芯片内部模拟电路供电)

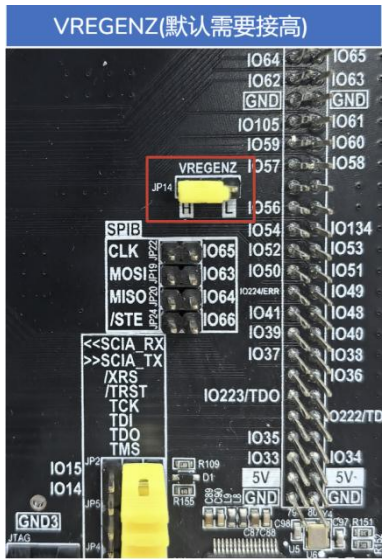


3.5 图一 JP11 管脚示意图

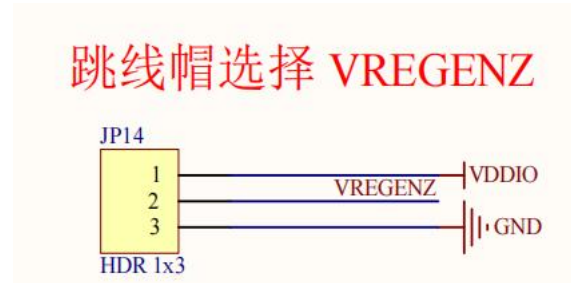


图为硬件原理图。其作用是给电路中模拟部分提供的电源电压。

### 3.6 VREGENZ 跳线



JP14 PCB 管脚示意图



图为硬件原理图

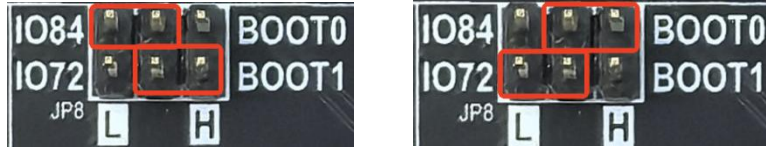
37X、38X、VREGENZ 必须上拉到 VDDIO,选择外部 1.1V 供电

075 上拉到 VDDIO 选择为外部 1.1V 供电, VREGENZ 下拉到 GND, 选择为内部 1.1V 供电

### 3.7 BOOT 方式选择(芯片的启动方式)



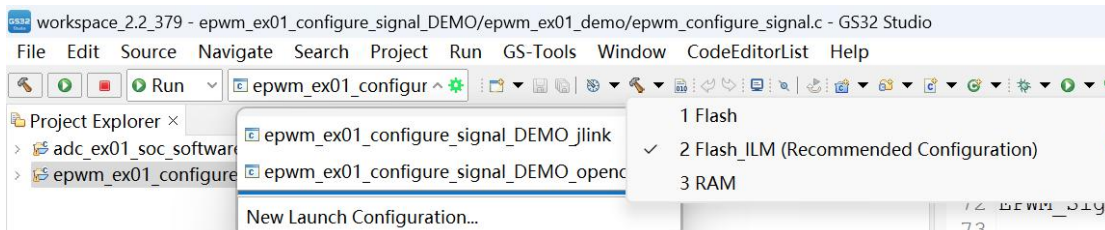
以上这两种 bootpin 的连接方式表示芯片从 flash 启动如图 2-9-1 所示。



图二和图三展示的是芯片分别从 bootrom 和 sram 启动 bootpin 的连接方式。启动模式配置如下表所示:

	丝印 (BOOT0)	丝印(BOOT1)	启动模式
引脚	GPIO84	GPIO72	\
状态	LOW	LOW	FLASH
状态	HIGH	HIGH	FLASH
状态	LOW	HIGH	BOOTROM
状态	HIGH	LOW	SRAM

#### 3.7.1 软件配置



1. 选中需要编译运行的文件，在 **launch configure** 窗口选择调试和连接的方式，**openocd/J-LINK**,并在编译选项中选择编译程序运行的区域

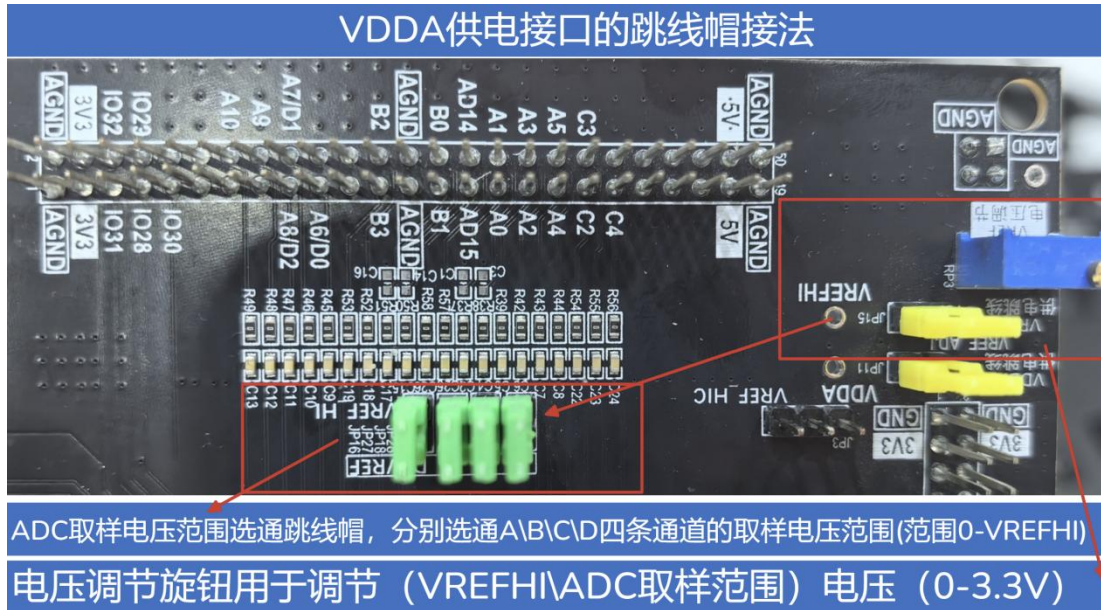


- 在 RAM 区域运行程序时，点击 **launch configure** 右边绿色的小齿轮打开 **Edit Configuration** 界面在 **Startup** 界面勾选 **Debug in RAM** 选项与 **Restart reset** 选项，才能进行 **Debug** 调试
- 在 **FLASH** 区域运行时，**Debug** 调试需要取消勾选 **Debug in RAM** 选项。

### 3.8 ADC 测试引脚连接

ADC 测试引脚可以选择外部参考电压与内部参考电压两种方式

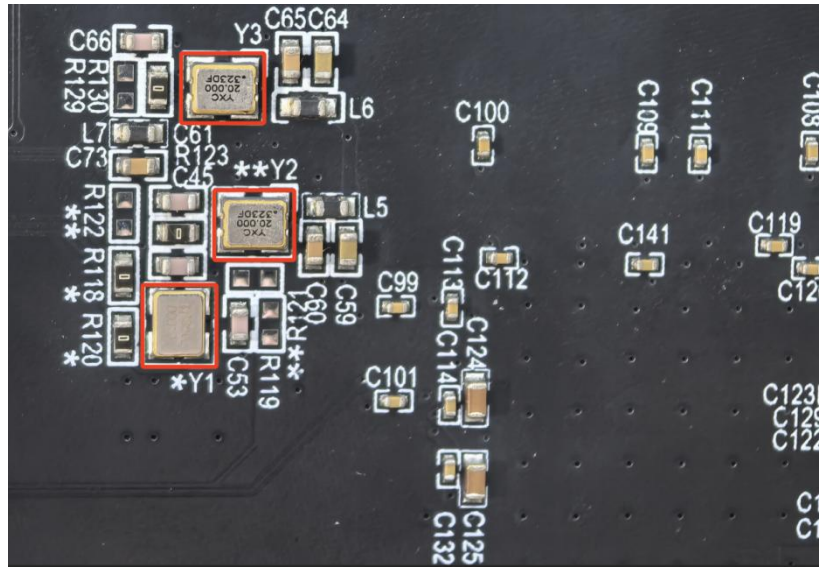
ADC 外部参考电压跳线



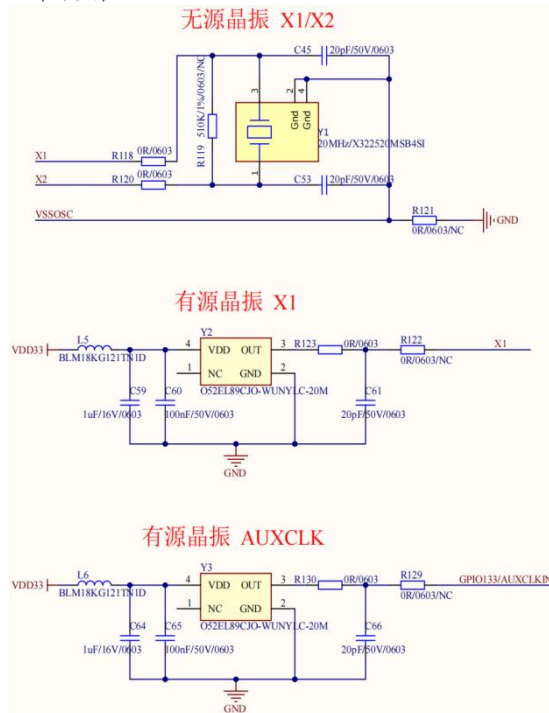
#### ADC 硬件接口原理图

图中的跳线是否接通，取决于是否使用 ADC 通道，从左到右依次是通道 D、A、B、C，每个通道对应的 I/O 口如原理图所示。

### 3.9 晶振



3.9 图一：PCB 晶振位置示意图（背面）



3.9 图二：晶振原理图

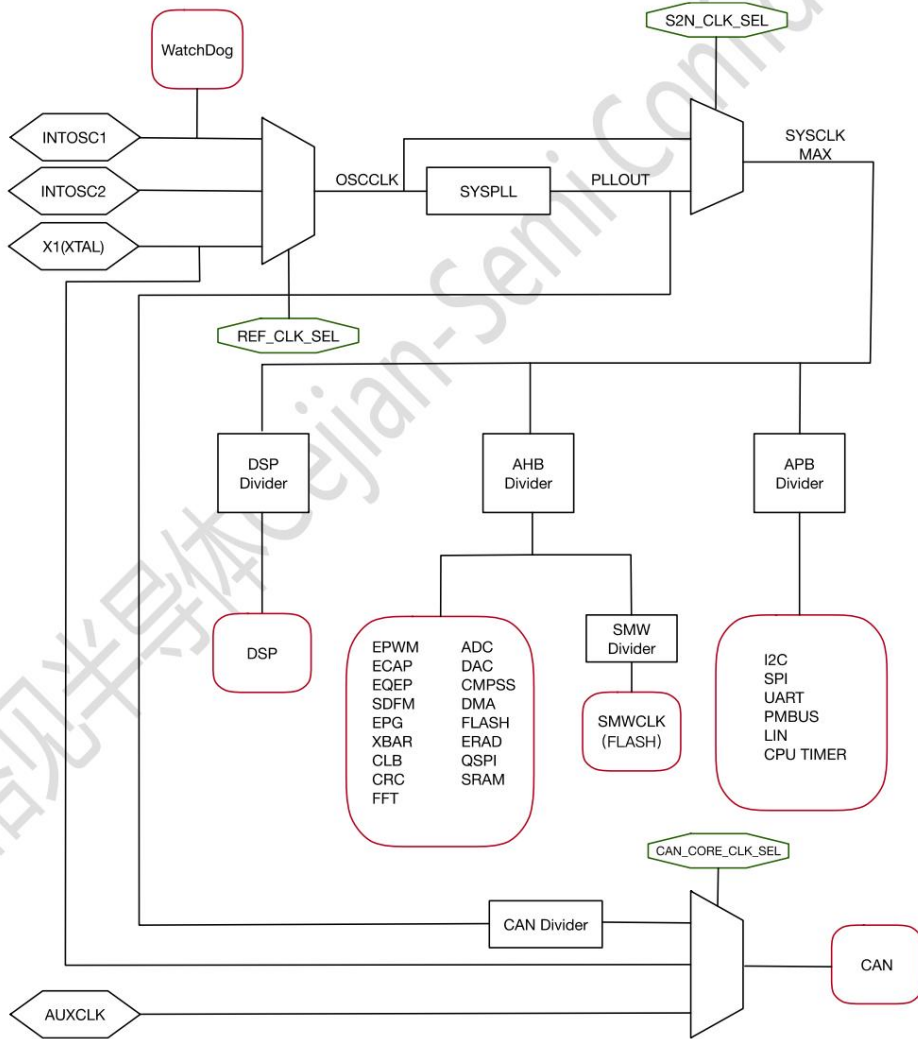
图一和图二分别是晶振的 PCB 和原理图。无源晶振 X1/X2 和有源晶振 X1 是为电路提供稳定的 20M 的时钟信号，有源晶振 AUXCLK 是为 CAN 通信提供时钟信号。当前电路板上没有焊接 R122，表示没有使用有源晶振 X1。没有焊接 R129，表示没有使用有源晶振 AUXCLK。如果需要使用，要焊接这两个电阻。如果不使用无源晶振 X1/X2，可以取下 R118 和 R120。

频率设置参考图:

表6-4. 可选的参考时钟源

时钟源	描述
INTOSC1	内部振荡器 1。 无需引脚连接的 10 MHz 内部振荡器。
INTOSC2 <sup>(1)</sup>	内部振荡器 2。 无需引脚连接的 10 MHz 内部振荡器。
X1 (XTAL)	连接在 X1 和 X2 引脚之间的外部晶体或谐振器，或连接到 X1 引脚的单端时钟。
AUXCLKIN	单端 3.3V 电平时钟源。GPIO133/AUXCLKIN 引脚应用于提供输入时钟。

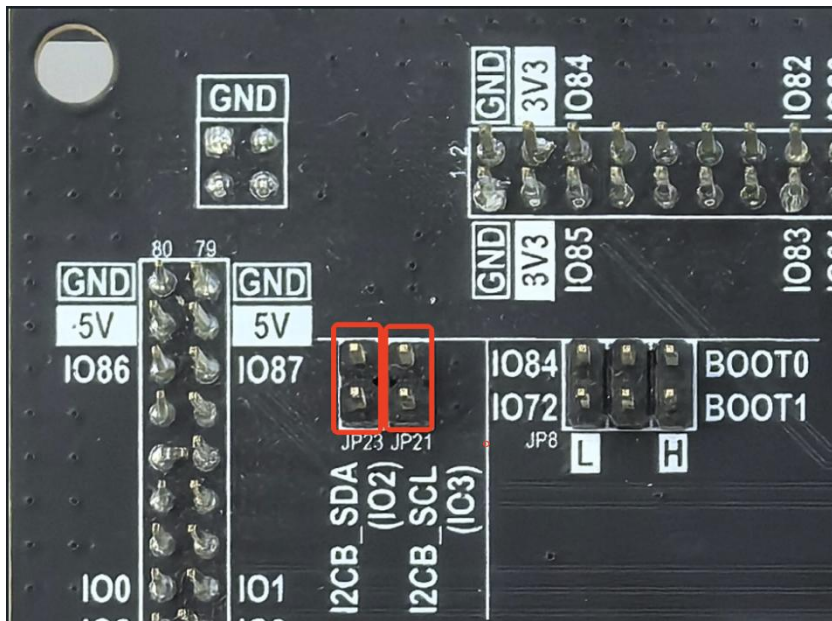
(1) 复位时，内部振荡器 2 (INTOSC2) 是 PLL (OSCCLK) 的默认时钟源。



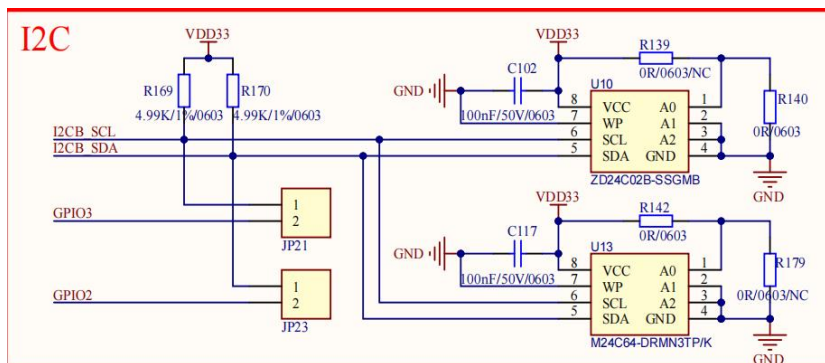
3.9 图三: 时钟示意图

例如: 设置为 X1 为时钟源 (外部) 时, 进入为 20MHZ, 倍频 30 倍, 为 600MHZ, 设置分频为 2 分频, 系统的时钟频率就为 300MHZ

### 3.10 I2C 引脚选择



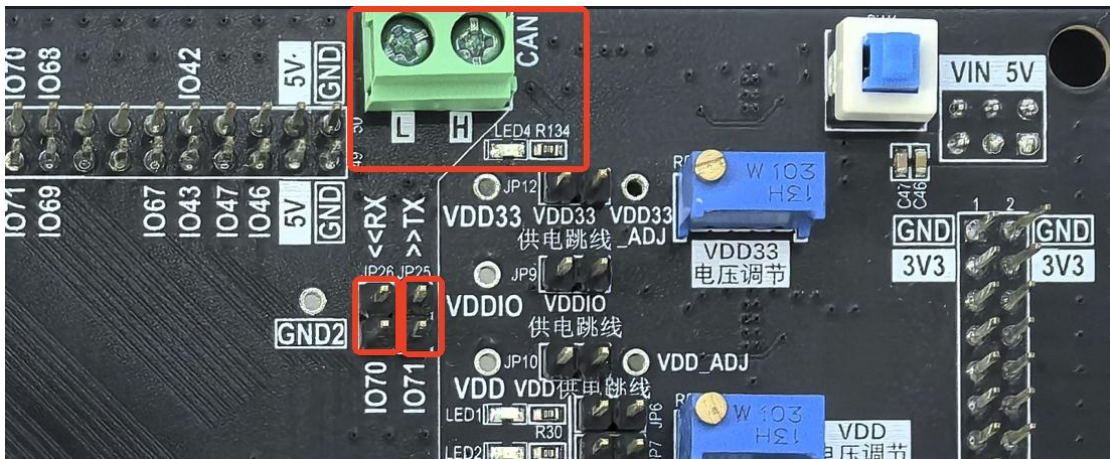
3.10 图一 I2C 的 PCB 引脚示意图



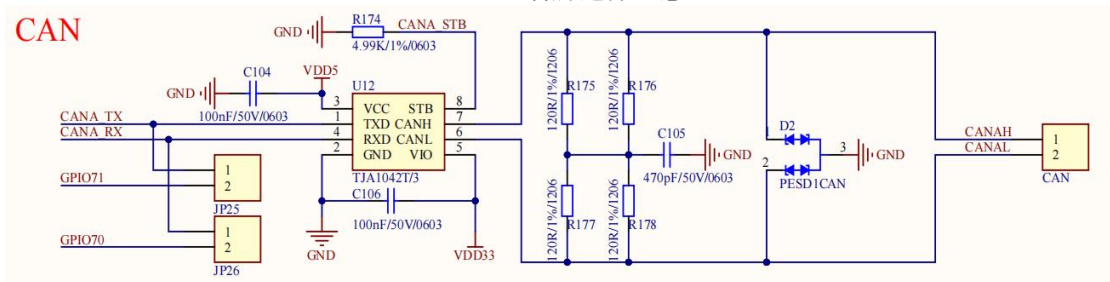
3.10 图二 I2C 原理图

按如图一所示连接跳线，GPIO2(I2C\_SDA)、GPIO3(I2C\_SCL)与板上排针相连

### 3.11 CAN 引脚选择



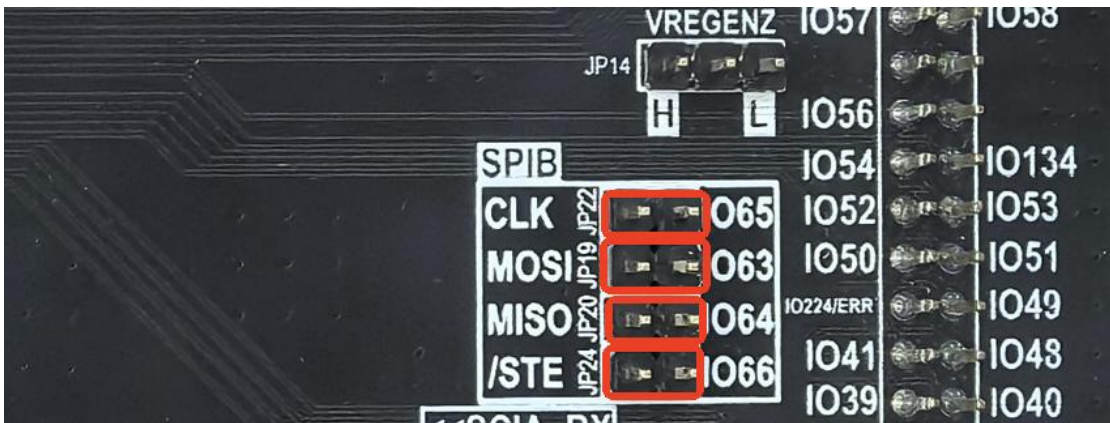
3.11 图一 引脚选择示意图



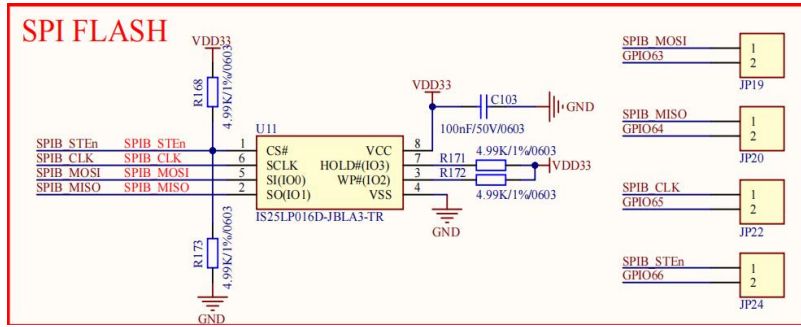
3.11 图二 CAN 硬件原理图

该引脚按照上图连接，GPIO70 连接 CANTX，GPIO71 连接 CANRX。

### 3.12 SPI 管脚连接



3.12 图一: -SPI PCB 硬件管脚连接



3.12 图二 SPI 硬件原理示意图

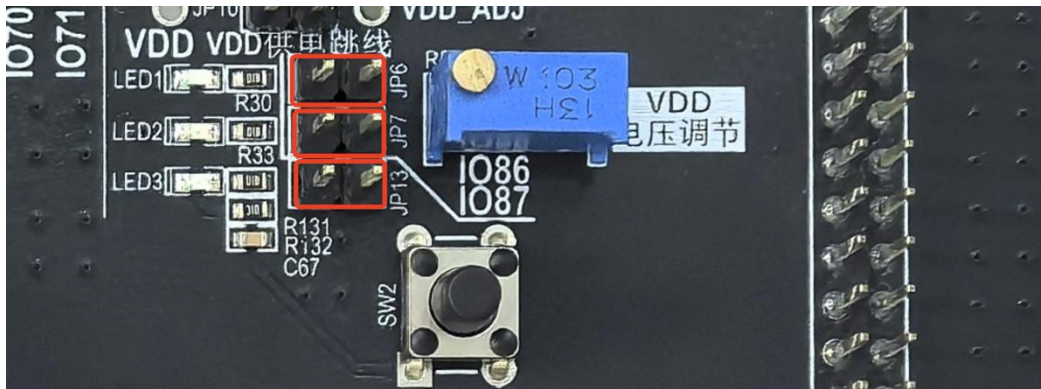
SPI 管脚的硬件连接如图一所示，硬件原理图如图二所示。

连通——连接板上的 SPI-Flash IS25LP016D-JBLA3-TR

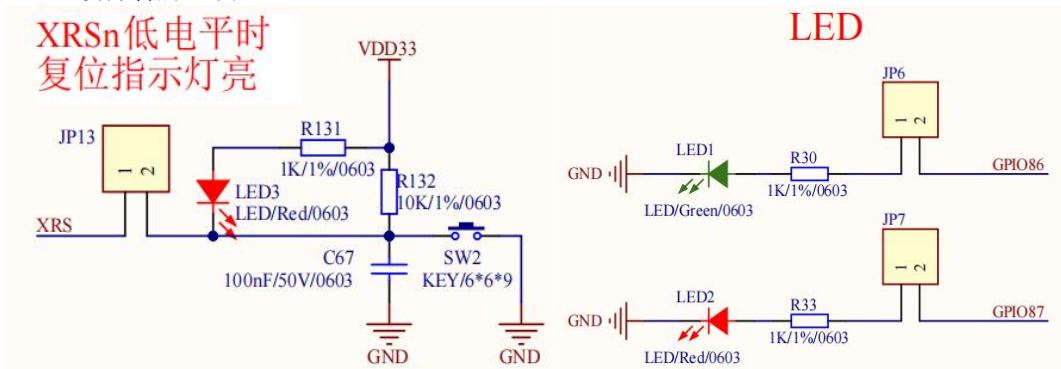
断开——图片右侧引脚作为普通 GPIO 使用，与板上排针相连。

按照该图进行跳线帽的连接默认使用的是 SPIB 读写内部 FLASH，硬件内部是将 SPIB 连接到 SPI FLASH.

### 3.13 LED 引脚连接



3.13 图一 LED 硬件管脚连接



3.13 图二 LED 硬件原理图

LED1 和 LED2 是调试灯对应 GPIO86 和 GPIO87，高电平时 LED1 为绿色 LED2 为红色。LED3 是复位指示灯，按下 RESET KEY 时，XRSn 为低电平，复位指示灯亮。使 PCB 恢复到上电时的初始状态。

## 4 基本操作

### 4.1 上电与连接

通过该 PCB 的 type-c 进行 USB 供电(typec 接口也可以用作串口连接上位机使用)。使用板载 OpenOCD 调试器, 或者外部 JLink 调试器

### 4.2 复位

在该 PCB 上, 可以通过 XRSn 按钮进行硬件复位。

### 4.3 常见问题

#### 4.3.1 板载 openocd

##### ①驱动安装

打开 GS32\_Studio IDE 工具-->tools-->HBird\_Driver。双击 HBird\_Driver 进行驱动安装, 若安装失败, 右键点击, 以管理员身份安装。遇到问题也可以在控制面板中删除驱动, 重新进行安装

##### ②连接电脑

请使用带数据通信功能的 typeC 数据线, 确保电脑可以正常识别到端口 (在设备管理器中查看), 可以通过插拔 USB 数据线来对比端口的变化, 如图为 openocd 映射的 USB 端口



##### ③调试设置

确认端口已经正确连接.在 IDE 中先点击项目文件然后在 launch configuration 中选择 项目文件名 openocd 在运行模式中选择 DEBUG, 后进行 DEBUG 调试。(遇到弹出 openocd 报错可尝试摁下复位按钮解决) 故障排查, 可在工程师的协助下快速找到问题:

##### ④openocd 查看版本号

例如: 在 GS32\_Studio IDE 工具-->toolchain-->openocd 目录下打开批处理文件 gs32f00xx\_v22\_standard

gs32f00xx_v12_attach	2025/2/24 18:50	Windows 批处理文件	1 KB
gs32f00xx_v12_standard	2025/2/24 18:51	Windows 批处理文件	1 KB
gs32f00xx_v22_attach	2024/9/25 16:12	Windows 批处理文件	1 KB
gs32f00xx_v22_standard	2024/9/25 16:12	Windows 批处理文件	1 KB

```
PS C:\Users\Administrator\Desktop\GS32\GS32_Studio_v1.1.0_Pre\toolchain\openocd> .\bin\gs-openocd.exe -s ..\..\packages\gs32_fw\utilities\scripts\ -f gs32f0039_ftdi_dsp_flash.cfg
GS-DSP OpenOCD 0.11.0+dev-02405-ga383d1d03-dirty (20250327-06:19:58)
Licensed under GNU GPL V2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
0.020 - Info: INIT target_init: resethalt=0
0.030 - Info: algorithm find[..\..\packages\gs32_fw\utilities\scripts\GS32F0039.bin]
0.034 - Info: expect_model:GS32F0039 expect_series:cs12 support.hardware:DEV_Seq[0x39] CORE_Type[0x0] SYS_Freq[0xf0] SPECGrade[0x1] Temp[0x1] Target[0x1] FLASHSize[0x5] MEMSize[200K]
0.042 - Info: algorithm find[..\..\packages\gs32_fw\utilities\scripts\GS32F0039.bin]
0.045 - Info: expect_model:GS32F0039 expect_series:cs12 support.hardware:DEV_Seq[0x39] CORE_Type[0x0] SYS_Freq[0xf0] SPECGrade[0x1] Temp[0x1] Target[0x1] FLASHSize[0x5] MEMSize[200K]
0.052 - Info: algorithm find[..\..\packages\gs32_fw\utilities\scripts\GS32F0039.bin]
0.055 - Info: expect_model:GS32F0039 expect_series:cs12 support.hardware:DEV_Seq[0x39] CORE_Type[0x0] SYS_Freq[0xf0] SPECGrade[0x1] Temp[0x1] Target[0x1] FLASHSize[0x5] MEMSize[200K]
0.063 - Info: algorithm find[..\..\packages\gs32_fw\utilities\scripts\GS32F0039.bin]
```

如图所示 Openocd 版本为 0.11 版本, 修订日期为 20250324。

### 4.3.2 J-LINK 连接

GS32F2837X\_38X\_075 生态板 J-LINK 端口为 14PIN J-TAG 接口，可使用转接板转换为 14PIN T-TAG 端口。

调试器需使用 V11 版本及以上的 ARM V11 仿真器，既支持 ARM 内核，也支持 RISC-V 内核。

J-Link 官网的介绍页面如下：

[https://kb.segger.com/UM08001\\_J-Link\\_/J-Trace\\_User\\_Guide#Target\\_interfaces\\_and\\_adapters](https://kb.segger.com/UM08001_J-Link_/J-Trace_User_Guide#Target_interfaces_and_adapters)

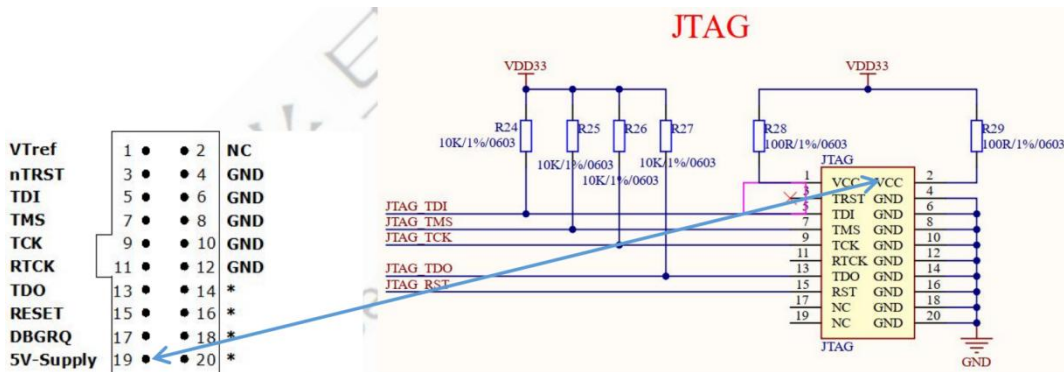
查看该页面的如下内容：

“11 Target interfaces and adapters”

=> “20-pin J-Link Connector (e.g. Flasher/J-Trace/J-Link)”

=> “[https://kb.segger.com/20-pin\\_J-Link\\_Connector](https://kb.segger.com/20-pin_J-Link_Connector)”

这里的 20pin 调试器接口引脚定义参考下图：



关于调试器接口信号的简单说明：

①必要的信号包括 VTref (3.3V 参考电压)、GND、TDI、TDO、TMS、TCK；  
无用的信号包括 Pin3 nTRST、Pin11 RTCK、Pin17 DBG RQ

②Pin15 RESET 是调试器输出的高低电平信号，用于强制复位目标器件，通常不用该信号；

③Pin1 VTref = Pin 1 VCC，它对调试器是输入引脚，目标板应在该管脚给调试器提供 3.3V 参考电压输入；

④J-Link 官方调试器的 Pin19 5V-Supply 用于调试器给目标板供电，但是在我们实际使用的 J-Link 调试器通过 Pin2 给目标板供电，Pin19 是 NC (无连接)；且我们通常不使用调试器给目标板供电 (供电能力不足，会带来其它问题，且可能损坏调试器)；

⑤JLINK 软件的安装，打开 GS32\_Studio IDE 工具-->tools-->JLINK\_Windows\_V798a\_x86\_64.exe 程序，进行软件安装，

⑥让 J-Link 支持 GS32 DSP 系列器件：

需要将 SDK 路径 “gs32\_fw\utilities\flashloader\Jlink\_addon” 中的 “Devices” 目录与 “JLinkDevices.xml” 文件复制到 JLink 安装目录下。

其他：

该软件包安装之后有如下软件： (安装该软件包之后即完成 J-Link 驱动的安装)

● J-Link Commander v7.98a

-常用来检查 JTAG 连接，检查 MCU 状态，学习 J-Link Commander 的各种命令：

[https://kb.segger.com/J-Link\\_Commander](https://kb.segger.com/J-Link_Commander)

● J-Flash v7.98a

用来擦写目标器件内部的 Flash；J-Flash 的使用说明：

<https://kb.segger.com/J-Flash>

● J-Link GDB Server v7.98a

-被 IDE 调用，用于下载与调试；相应的使用说明与支持的 GDB 命令：

[https://kb.segger.com/J-Link\\_GDB\\_Server](https://kb.segger.com/J-Link_GDB_Server)

## 5 修订历史

版本	时间	内容描述	审核人	批准人
v0.1	2025.5.20	初始版本	Jason	