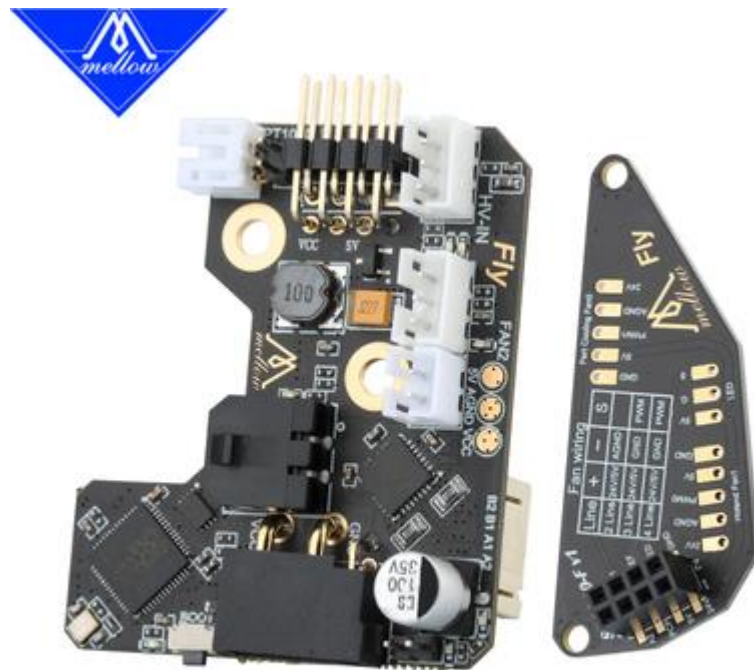


3D MELLOW

FLY CAN SB2040

工具板使用说明



Voron-SB2040 V1

修订历史

版本	修改说明	日期
1.00	初稿	2022.12.30
1.01	新增 3.2 上位机检测 章节，检测您的上位机内核是否支持 CAN；新增 3.5 加速度计的使用 章节，介绍加速度计的使用；新增 4. FAQ 章节	2022.01.05

目录

1.产品简介	1
1.1 产品特点	1
1.2 产品参数	1
2.硬件篇	3
2.1 接口说明	3
2.2 终端电阻配置	3
2.3 接线	4
2.3.1 电源接线	4
2.3.2 加热棒接线	5
2.3.3 热敏接线	5
2.3.4 风扇接线	6
2.3.5 RGB 灯	7
2.3.6 挤出机	8
2.3.7 限位开关	9
2.3.8 调平传感器	10
2.3.9 SB2040 与 UTOC	12
3.软件篇	14
3.1 上位机的配置	14
3.2 上位机检测	15
3.3 SB2040 固件刷写	15
3.3.1 不带 CanBoot 的固件烧录	15
3.3.2 带 CanBoot 的固件烧录	18
3.4 桥接 CAN	22
3.5 加速度计的使用	23
3.5.1 加速度计配置	23
3.5.2 加速度计的使用	23
4. FAQ	24
相关链接	24

1. 产品简介

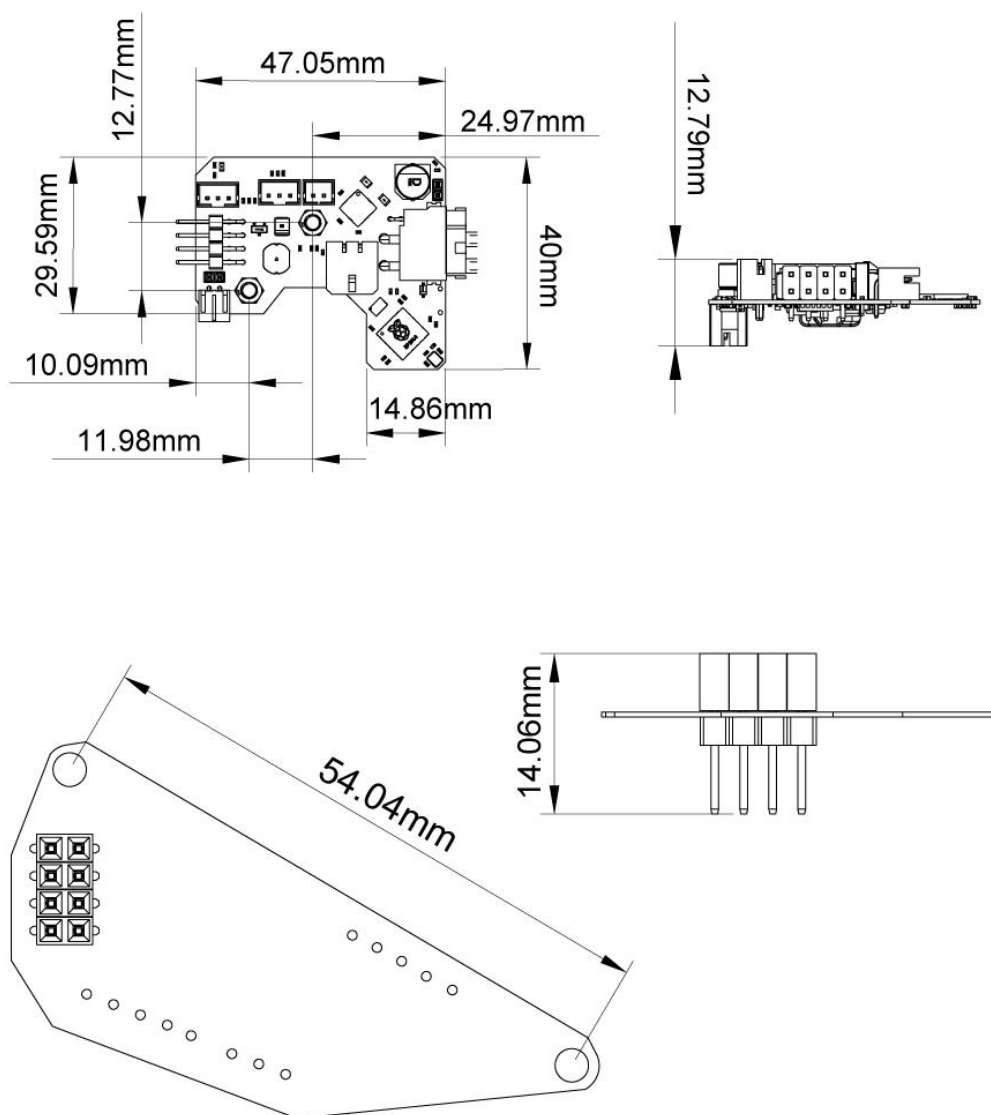
FLY SB2040 是广州镁伦电子科技有限公司针对 VORON Stealthburber 打印头推出的一款喷头工具板，使用该工具板，可以用四根线来替代原来主板与打印头之间繁杂的接线，优化布局。SB2040 适用于 36 步进电机，其集成了 CAN 收发器、USB 口、TMC2209 驱动、ADXL345 加速度传感器，两个可控风扇等其他功能的工具板。此工具板适用于 Klipper 固件。

1.1 产品特点

- 支持 CAN 总线连接，数据传输更稳定，延迟更小，连接更稳定
- 板载 TMC2209 驱动，使用 UART 模式，可用于挤出机
- 板载 ADXL345 加速度传感器
- 支持 BLTOUCH、PL08N 等调平传感器，集成 XY 限位开关
- 支持 12-24V 电压输入
- 主控使用 Raspberry PI rp2040
- 增加支持 4.7K 电阻(用于 ntc 热敏)或 1.1K 电阻(用于 PT1000)切换
- can 接口使用 XT30 (2+2)，支持 15A 电流，峰值 30A
- 增加板载 NTC100k 电阻，用于测仓温
- 增加 ADXL345 的 INT1 和 INT2 接到 mcu，给个别用户用于调平。
- 支持高压输入限位
- 赠送 3 米 CAN 连接线，避免用户无法正确压端子导致信号丢失的困扰

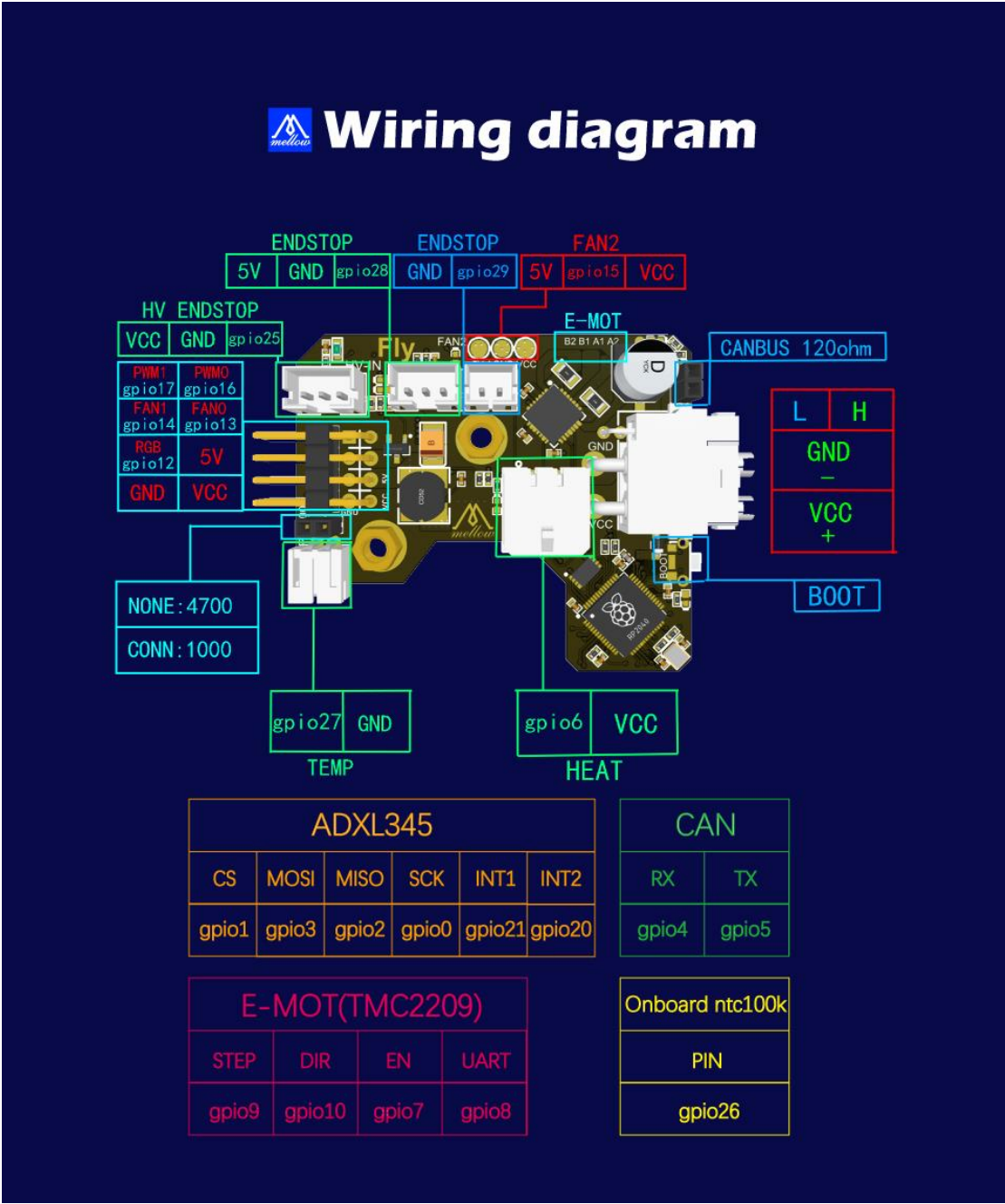
1.2 产品参数

- 1.外观尺寸： 47.05mm*40mm
- 2.安装尺寸： M3 安装孔径
- 3.处理器： Raspberry PI rp2040, Dual core ARM Cortex-M0+@133MHz
- 4.输入电压： DC12-24V 15A
- 5.加热棒最大输出电流： 5A
- 6.风扇接口最大输出电流 1A



2.硬件篇

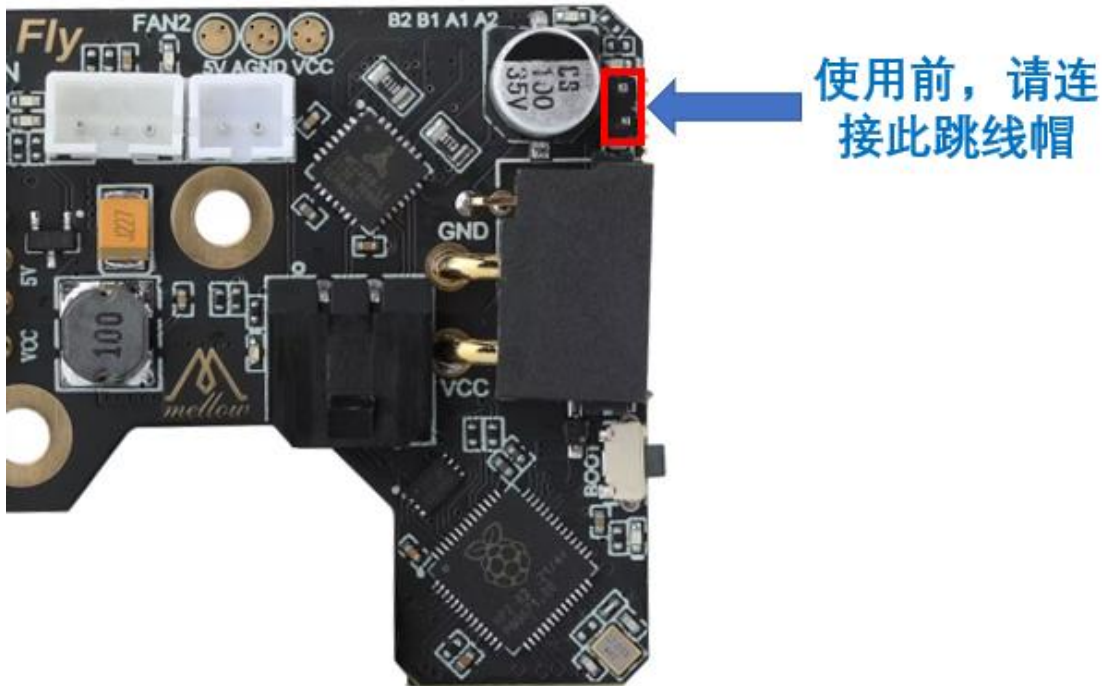
2.1 接口说明



2.2 终端电阻配置

- 使用 CAN 之前，请正确配置 CANBUS 终端电阻。
- CANBUS 总线协议一条总线中必须且只能有两个 120 欧姆电阻

- 不管你连接几个 USB 设备，只要是在一条总线就只配置两个 120 欧姆电阻。不用为每个设备加一个电阻
- 使用万用表测量 CAN H 与 CAN L 之间的阻值应该为 ≈ 60 欧姆



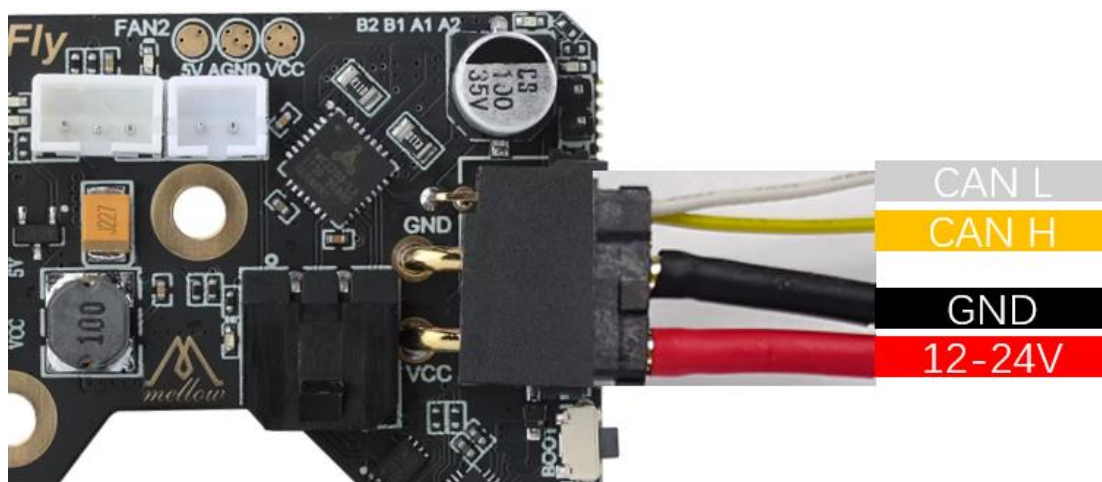
2.3 接线

首先，请确保您已经正确插上终端电阻的跳线帽。

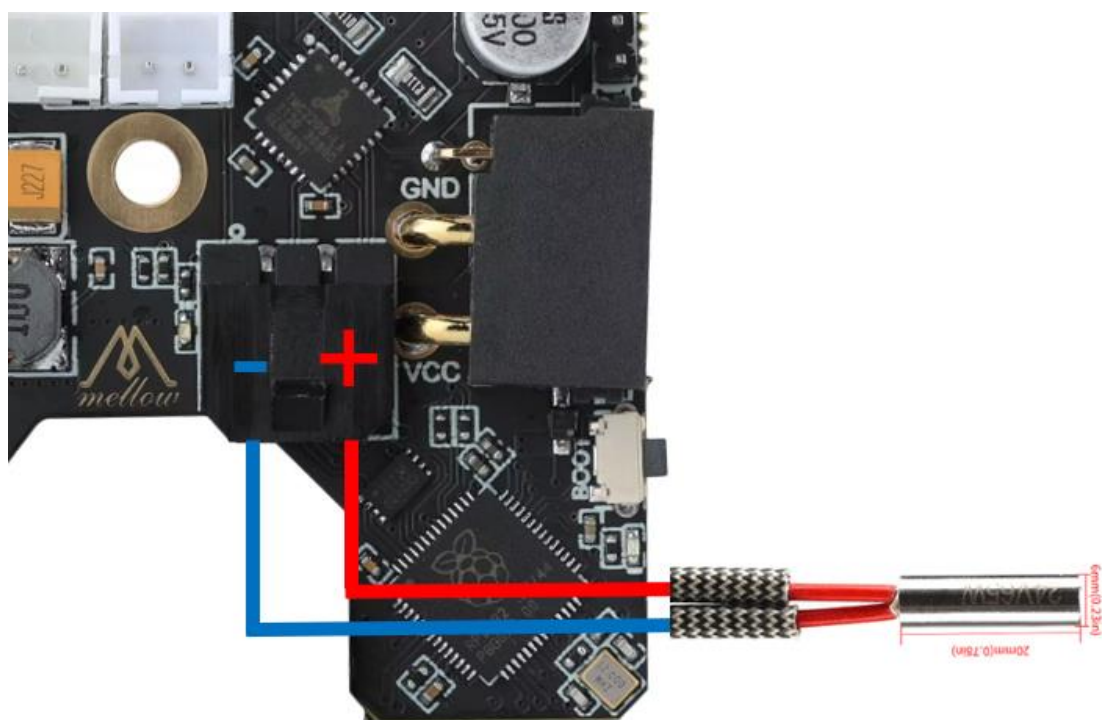
2.3.1 电源接线

SB2040 不支持电源防反接，在通电前请检查电源正负极接线是否正确，否则

会损坏您的 SB2040 工具板!!!



2.3.2 加热棒接线



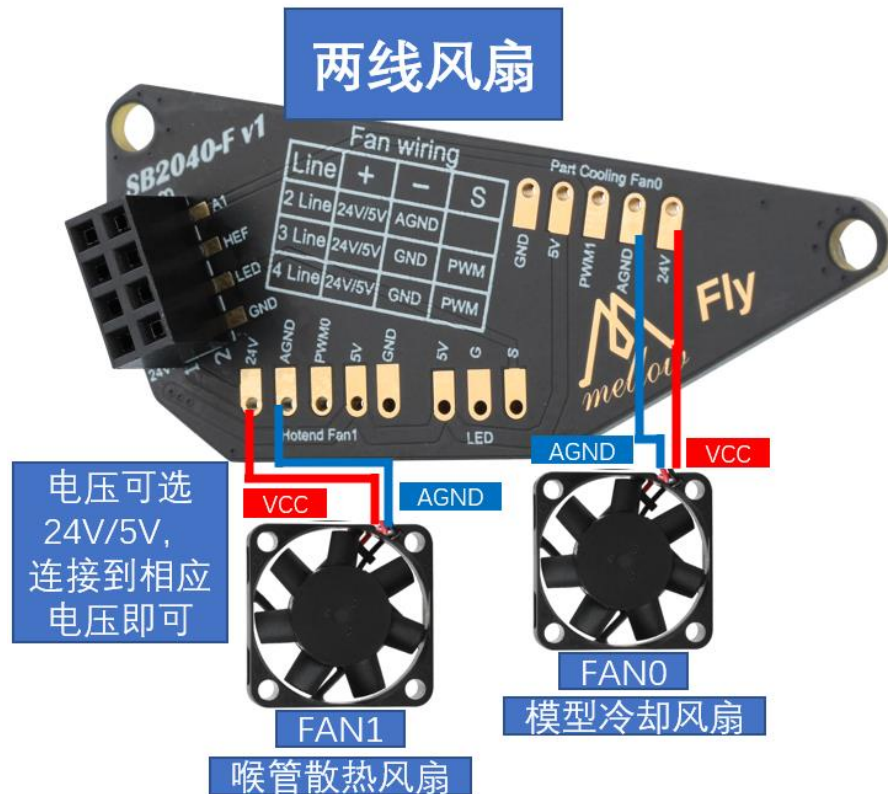
2.3.3 热敏接线

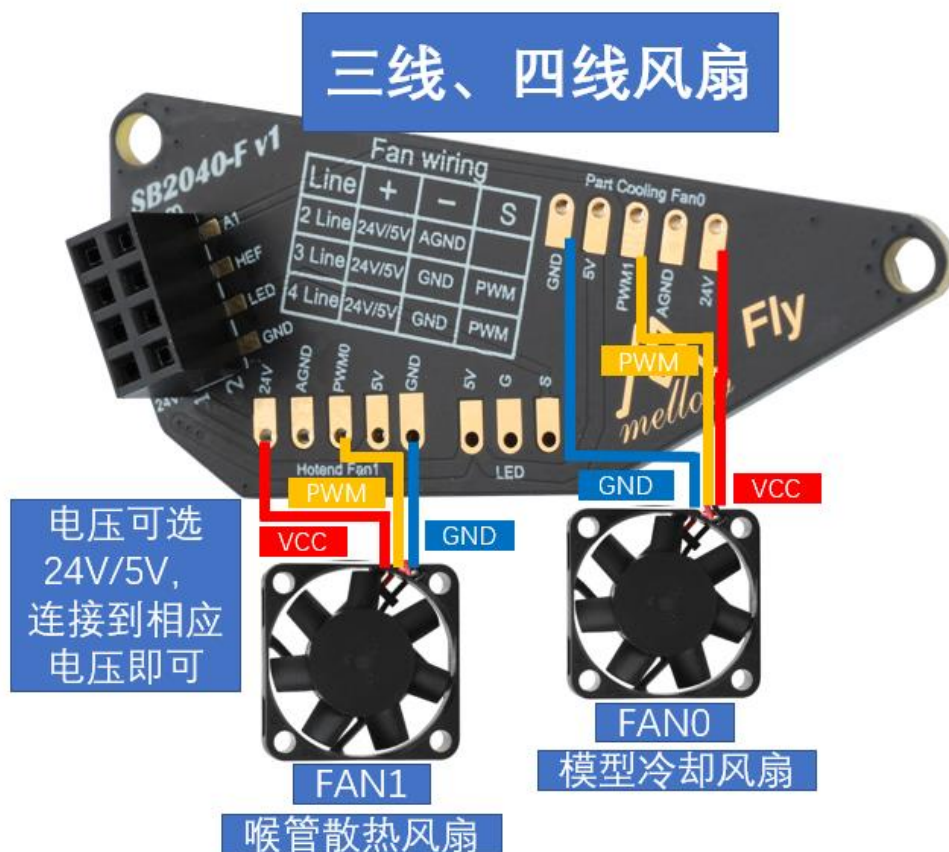
下图为普通热敏和 PT1000 的接线方法。注：SB2040 不支持 PT100。



2.3.4 风扇接线

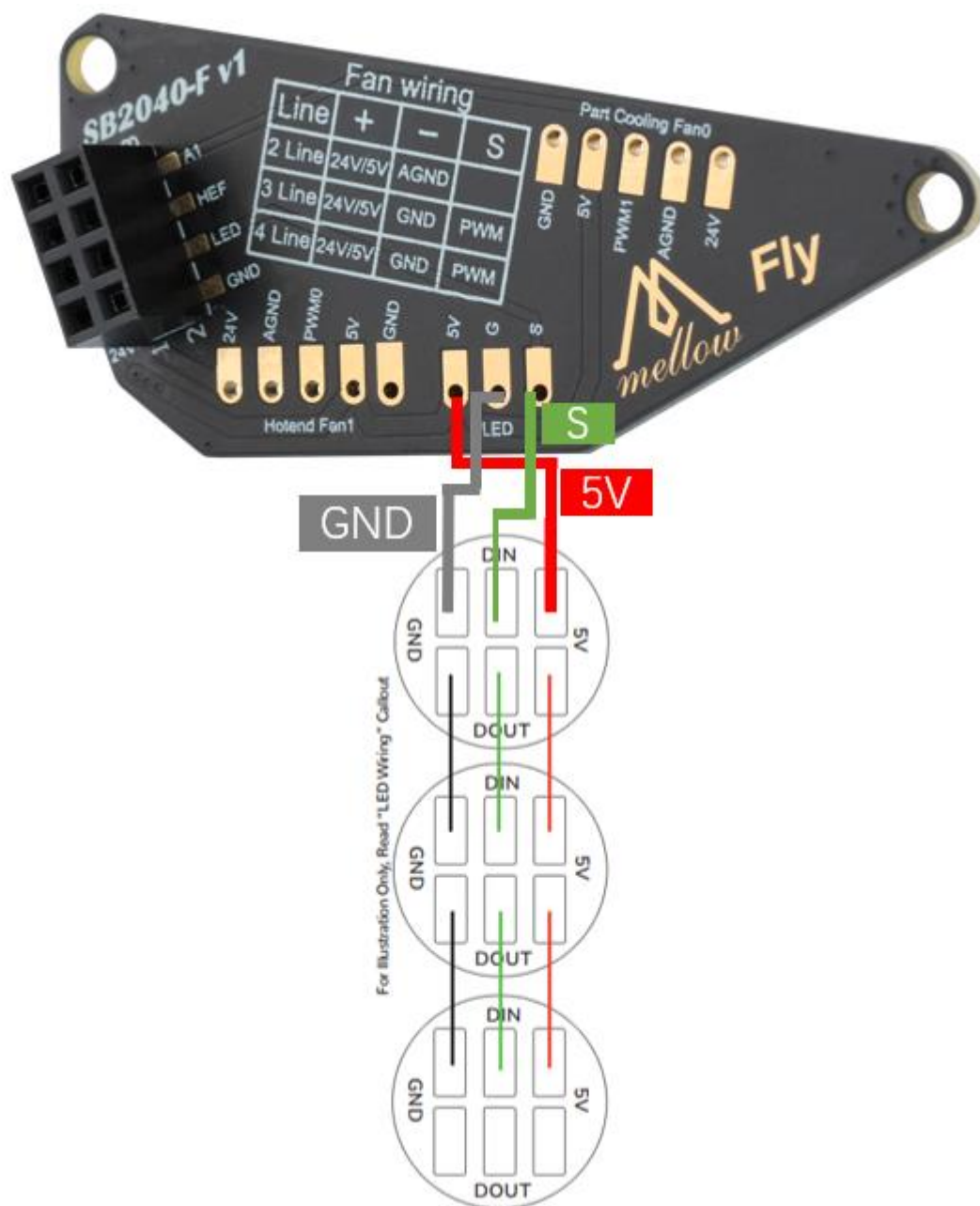
SB2040 风扇电压有 5V、24V (VCC) 可选，支持最多两个可控风扇，电压可以通过连接到相应电压来选择。另外，SB2040 支持 2 线、3 线、4 线风扇，连接方法如下图。





2.3.5 RGB 灯

接线时, 请注意 DIN 和 DOUT, 信号从 DIN 进, DOUT 出, 接反会造成不亮或者控制混乱等情况。



2.3.6 挤出机

挤出机接线完成后，请注意配置好驱动电流，校准好挤出机电机转向

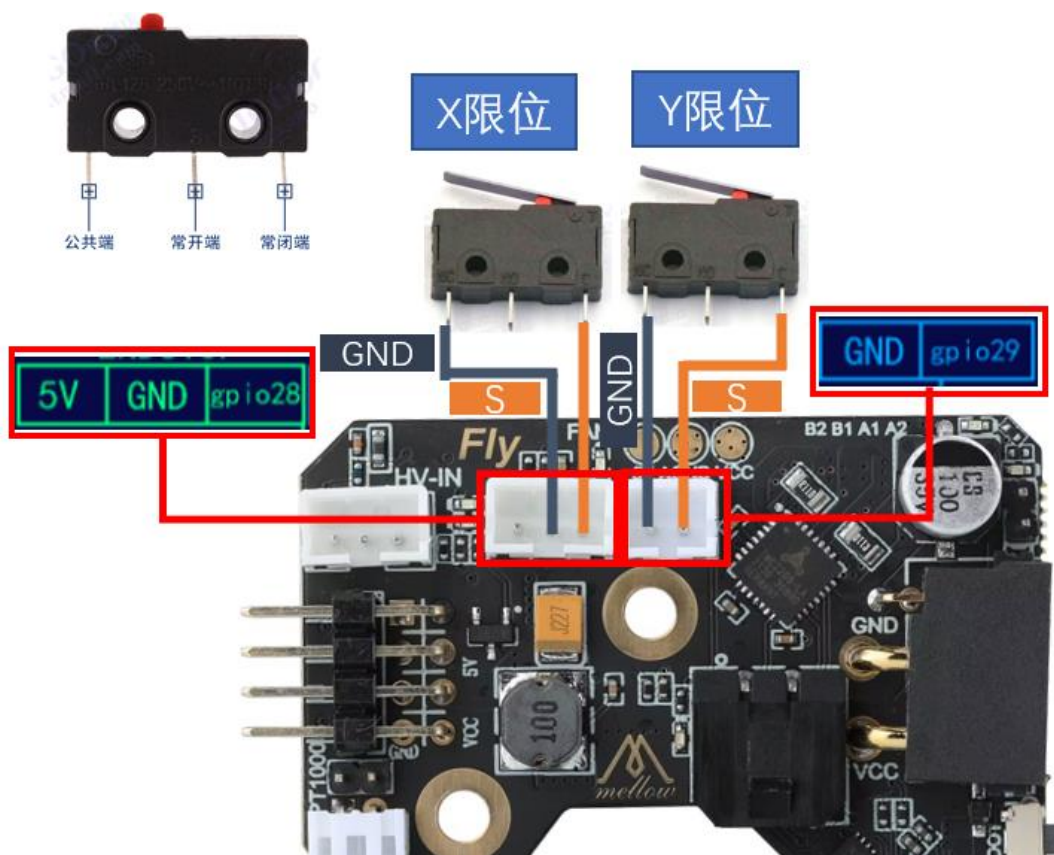


2.3.7 限位开关

限位开关有常开 (NO) 和常闭 (NC) 两种。一般在 3D 打印机上，建议使用常闭 (NC)，这样在限位开关线路出现问题时，系统会及时报错，可以避免一些不必要的撞车，避免损坏打印机。

如果为 VORON 机型，可以考虑更换限位的安装位置，在打印头滑车上安装 X 限位开关，A 电机座上安装 Y 限位开关。此安装方案，VORON 官方在打印文件中已提供。另外，在 SHT36 V2 上，建议限位开关前最好加上“^”，将信号上拉。例如：

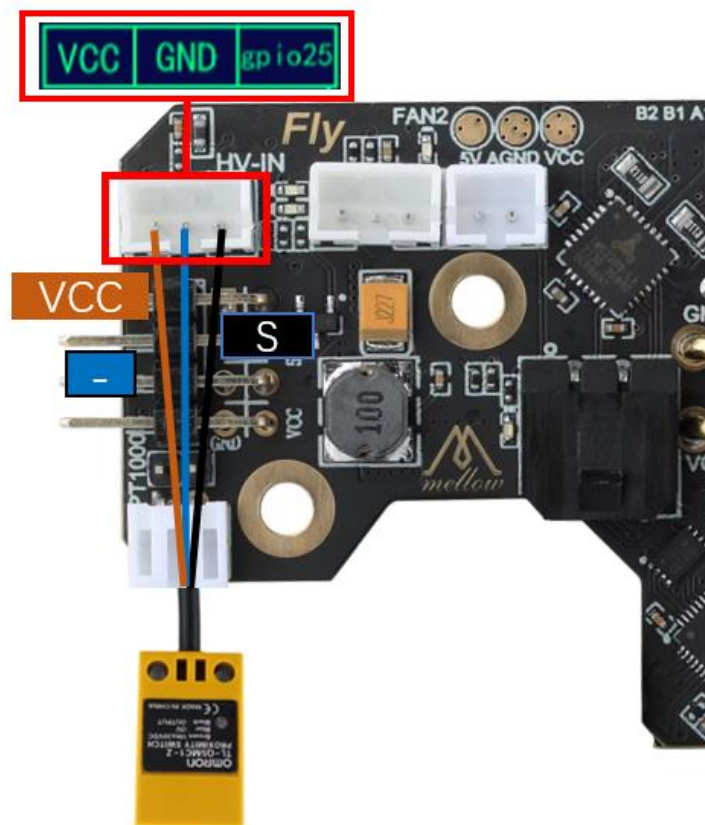
```
[stepper_x]
endstop_pin: ^sb2040:PA2
```



2.3.8 调平传感器

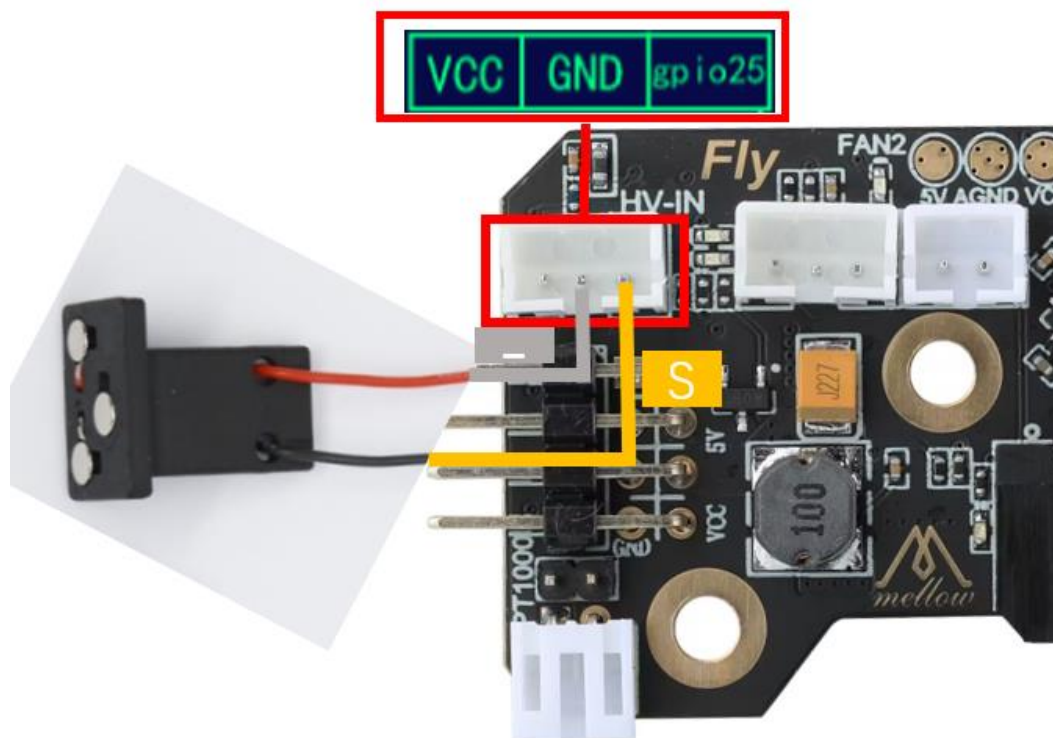
1) PL08N

VORON 官方建议是使用欧姆龙 Omron TL-Q5MC（之前官方推荐的是 PL08N，两者原理一样，只是探测距离不一样）传感器来进行热床调平。



2) Klicky

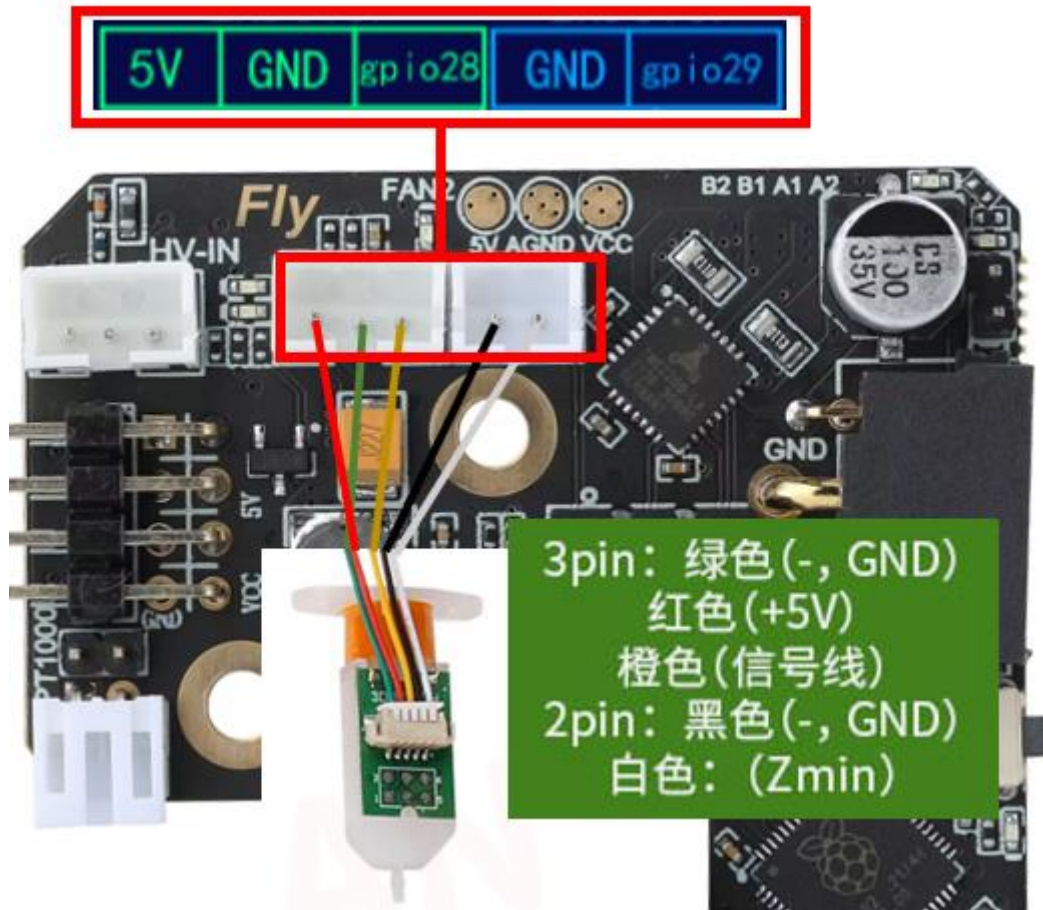
Klicky 为第三方的调平传感器，能够用极低的成本自己在家制作，且性能稳定，性价比非常高，推荐使用。接线方法如下图所示。



3) BLTOUCH

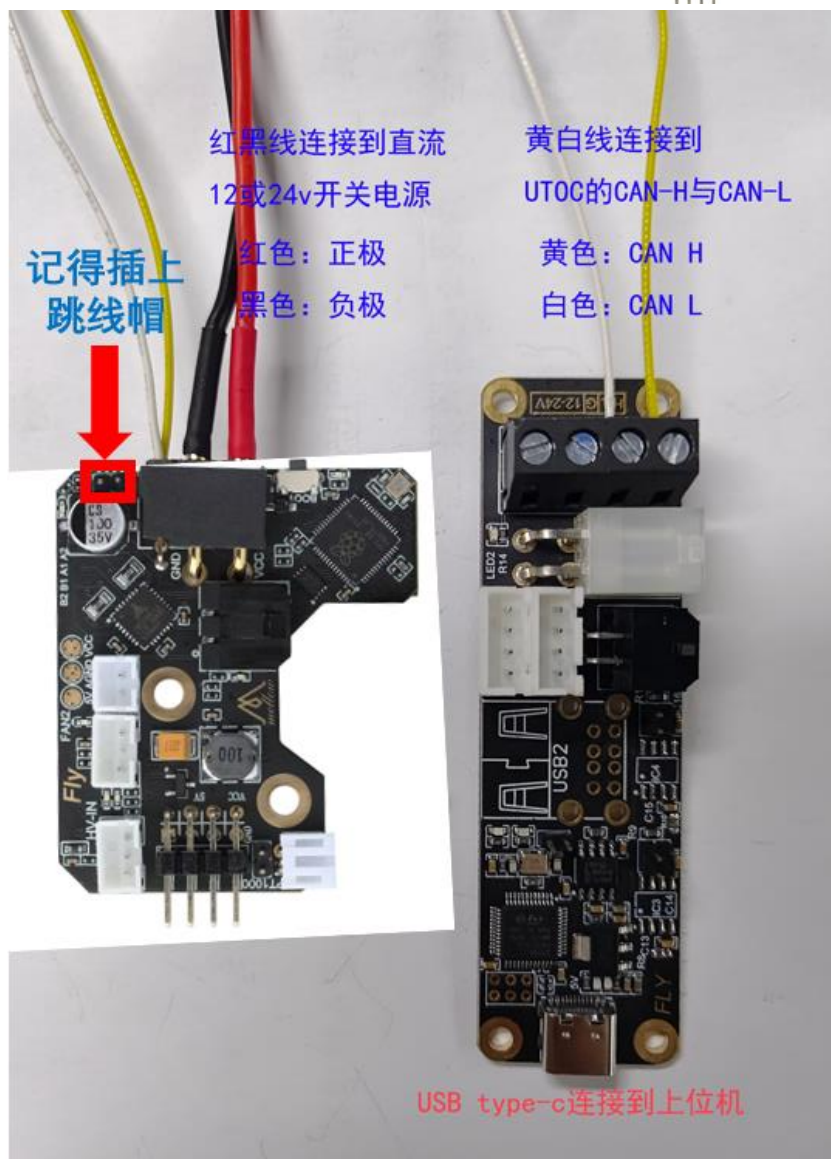
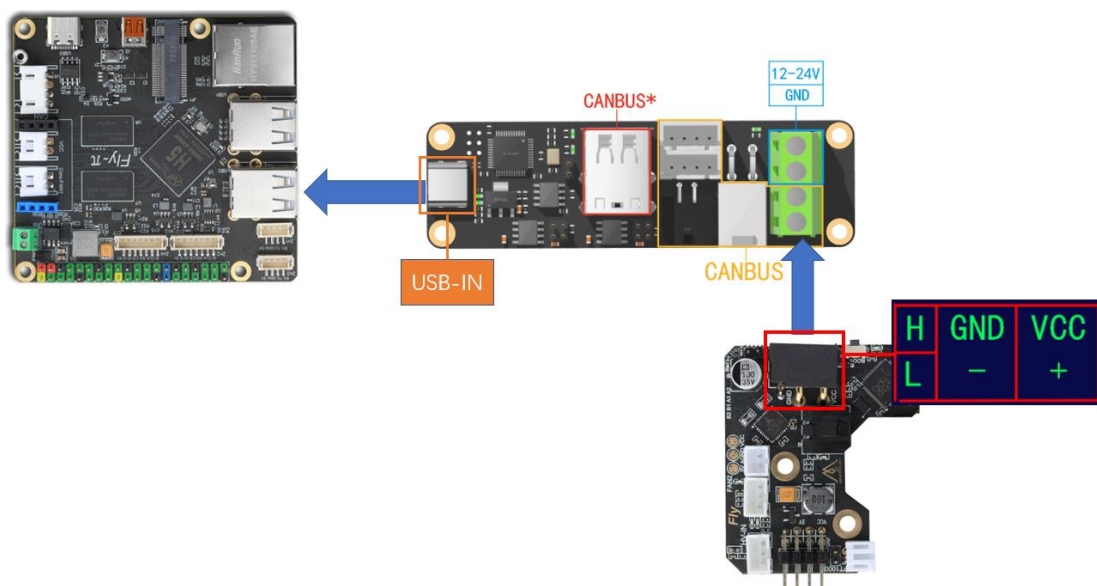
BL-touch 一共有五根线，三根为第一组，负责传感器的供电和探针的收放，第二组为地线 和信号线，输出限位信号。BL-touch 接线时请仔细核对线序，错误的接线可能会永久性损坏 传感器和主板!!! 接线方法如下图所示。

注：由于 SB2040 并未专门针对 BLtouch 设计接线座，因此使用 BLtouch 后，SB2040 至多只能接一个限位，即只能接到 HV-IN 接口上。



2.3.9 SB2040 与 UTOC

FLY UTOC 是一款 USB 转 CAN 总线的模块，通过它将 FLY π 的 USB 端口专为 CAN 总线，通过 CAN 总线连接 3D 打印主板、SHT36/42、SB2040 等 CAN 总线产品。FLY UTOC 板上拥有多种端子接口，使用灵活性好，可以适配不同的使用场景。另外，UTOC 在出厂时已经刷好固件，到手即用，无需再刷写固件。



USB-IN: USB 转 CAN 输入接口，连接到上位机

12-24v & GND: 电源接口

CANBUS: CAN 接口，连接到扩展主板和工具板等(连接到有板载 CAN 收发芯片的设备)

CANBUS*: CAN 接口，连接到扩展主板和工具板等(仅限连接到 STM32 设备的 USB 接口[PA11,PA12],请注意购买相应版本的 UTOC)

3.软件篇

- Gemini-v1, v1.1 版本不建议 CANBUS 和 KlipperScreen 同时启用，因为这两个进程的内存占用都比较大。Gemini-v2 有 1G 内存可以正常启用。
- 如遇到 Klipper 多 MCU 归位超时问题建议禁用 HDMI 输出，详情参考 <https://github.com/Klipper3d/klipper/issues/4861>
- 相关配置文件请前往 FLY 官方教程网站: [FLY-SB2040 \(klipper.cn\)](https://fly-sb2040.klipper.cn)

3.1 上位机的配置

注意!!! FLY π 、FLY Gemini 等 FLY 官方的上位机跳过此步骤。

非 FLY Gemini、 π 等上位机，需要先配置 CAN 才能正常使用 CAN。配置方法如下。

- 1) 执行下面的指令安装当前文档所需软件包

```
sudo apt update && sudo apt install nano wget -y
```

- 2) 创建配置文件，复制粘贴下面的指令到终端并回车即可

```
sudo /bin/sh -c "cat > /etc/network/interfaces.d/can0" << EOF
allow-hotplug can0
iface can0 can static
    bitrate 500000
    up ifconfig \${IFACE} txqueuelen 1024
EOF
```

- 3) 测试发现在部分设备中无法开机自动启用 CAN，所以建议都执行下面操作

- ①开机自动启用 CAN，复制粘贴下面的三段代码，每复制粘贴一段就回车一次

```
sudo wget https://cdn.mellow.klipper.cn/shell/can-enable -O /usr/bin/can-enable >
/dev/null 2>&1 && sudo chmod +x /usr/bin/can-enable || echo "The operation failed"
sudo cat /etc/rc.local | grep "exit 0" > /dev/null || sudo sed -i '$a\exit 0' /etc/rc.local
```

```
sudo sed -i '/^exit\ 0$/i \can-enable -d can0 -b 500000 -t 1024' /etc/rc.local
```

- ②重启设备

```
sudo reboot
```

- ③USB 转 CAN 模块在树莓派中无法即插即用，如果树莓派设备插拔过 USB 转 CAN 设备请所有**设备全部断电**，再重新上电设备或者执行下面的指令。

```
sudo can-enable -d can0 -b 500000 -t 1024
```

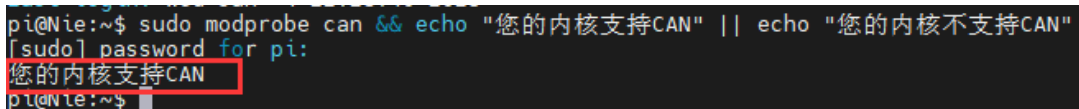
3.2 上位机检测

非 FLY 上位机的，请注意检查自己烧录的镜像内核是否支持 CAN。检测方法如下。

在 SSH 输入：

```
sudo modprobe can && echo "您的内核支持 CAN" || echo "您的内核不支持 CAN"
```

输入以上指令后，如果您的内核支持 CAN 就会返回：您的内核支持 CAN；如果不支持就会返回：您的内核不支持 CAN。



```
pi@Nie:~$ sudo modprobe can && echo "您的内核支持CAN" || echo "您的内核不支持CAN"
[sudo] password for pi:
您的内核支持CAN
pi@Nie:~$
```

3.3 SB2040 固件刷写

- SB2040 目前已支持 CanBoot。

什么是 CanBoot?

CanBoot 是 ARM Cortex-M mcu 的引导加载程序。这个引导加载程序最初是为 CAN 节点设计的，以便与 Klipper 一起使用。引导加载程序本身利用了 Klipper 的硬件抽象层，将内存占用最小化。除了 CAN, CanBoot 现在还支持 USB 和 UART 接口。

目前支持 lpc176x、stm32 和 rp2040 三种 MCU。CAN 支持目前仅限于 STM32 F 系列和 rp2040 设备。

Klipper 支持 CanBoot，通过 CANBUS 直接烧录固件。好处就是为 SB2040 工具板更新 klipper 固件无需再连接 USB 线；无需连接任何多余的线，保持现有的 CAN 连接的情况下可直接烧录固件。

3.3.1 不带 CanBoot 的固件烧录

SB2040 在出厂时并未刷写 Klipper 固件，因此在使用前需要为 CAN 工具板重新编译并烧写固件后，才能使用 CANBUS。

下面为 SB2040 编译固件。

1. 连接到 SSH。如何连接到 SSH 请参考：《[Super&VORON 修炼秘籍](#)》--3.4 连接 SSH
2. 进入固件编译配置界面。依次执行以下指令，即可进入固件编译配置界面。

```
cd ~/klipper && git pull
```

```
make menuconfig
```

2. 固件配置。如下图所示。

```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
  Micro-controller Architecture (Raspberry Pi RP2040) --->
  Bootloader offset (No bootloader) --->
  Flash chip (W25Q080 with CLKDIV 2) --->
  Communication interface (CAN bus) --->
(4) CAN RX gpio number (NEW)
(5) CAN TX gpio number (NEW)
(500000) CAN bus speed
(gpio24) GPIO pins to set at micro-controller startup
```

3.固件编译。SSH 输入以下指令并回车

```
make -j4
```

最后出现** Creating uf2 file out/klipper.uf2**则表示编译成功。

```
Compiling out/src/rp2040/hard_pwm.o
Compiling out/src/rp2040/spi.o
Compiling out/src/rp2040/i2c.o
Building rp2040 stage2 out/stage2.o
Preprocessing out/src/rp2040/rp2040_link.ld
Building out/compile_time_request.o
Version: v0.11.0-5-ga42f6158
Linking out/klipper_elf
Creating uf2 file out/klipper.uf2
fly@flygemini:~/klipper$
```

4.烧录固件。

(1) 烧录固件前，需要使 SB2040 工具板进入 Boot 烧录模式。方法如下。



(2) 输入以下指令，查看是否有 ID 2e8a:0003 Raspberry Pi RP2 Boot 这行，如没有请检查 USB 线(连接前记得按住 BOOT 键)。

```
lsusb
```

```
fly@flygemini:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 002: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 002: ID 1b3f:1167 Generalplus Technology Inc. WEB CAM
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 009 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

(3) 固件烧录。依次输入以下指令，出现下图则表示烧录成功。

```
cd ~/klipper/
```

```
make flash FLASH_DEVICE=2e8a:0003
```

```
fly@flygemini:~/klipper$ make flash FLASH_DEVICE=2e8a:0003
Building rp2040_flash
gcc -c -Wall -ggdb -I../rp2040/ `pkg-config libusb-1.0 --cflags` main.c -o main.o
gcc -c -Wall -ggdb -I../rp2040/ `pkg-config libusb-1.0 --cflags` picoboot_connection.c
gcc main.o picoboot_connection.o `pkg-config libusb-1.0 --libs` -o rp2040_flash
Flashing out/klipper.uf2 to 2e8a:0003
sudo lib/rp2040_flash/rp2040_flash out/klipper.uf2

[sudo] password for fly:
Loaded UF2 image with 114 pages
Found rp2040 device on USB bus 7 address 2
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
fly@flygemini:~/klipper$
```

5.检查。

如果固件配置正确并烧录成功，则下图中 SB2040 工具板的 LED 灯应该常亮。



3.3.2 带 CanBoot 的固件烧录

- 1.进入 SSH 终端。
- 2.依次执行以下指令。

```
git clone https://github.com/Arksine/CanBoot
cd CanBoot
```

```
fly@flygemini:~$ git clone https://github.com/Arksine/CanBoot
Cloning into 'CanBoot'...
remote: Enumerating objects: 534, done.
remote: Counting objects: 100% (135/135), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 534 (delta 100), reused 100 (delta 89), pack-reused 399
Receiving objects: 100% (534/534), 2.55 MiB | 1.61 MiB/s, done.
Resolving deltas: 100% (346/346), done.
Updating files: 100% (167/167), done.
fly@flygemini:~$
```

- 3.进入编译配置界面。输入以下指令，进入编译配置界面。

```
make menuconfig
```

参数配置如下图所示。

```
(Top)
Micro-controller Architecture (Raspberry Pi RP2040) --->
Flash chip (W25Q080 with CLKDIV 2) --->
Build CanBoot deployment application (Do not build) --->
Communication interface (CAN bus) --->
(4) CAN RX gpio number (NEW)
(5) CAN TX gpio number (NEW)
(500000) CAN bus speed
() GPIO pins to set on bootloader entry
[*] Support bootloader entry on rapid double click of reset button
[ ] Enable bootloader entry on button (or gpio) state
[*] Enable Status LED
(gpio24) Status LED GPIO Pin
```

- 4.编译 CanBoot 引导固件。配置完成后按“Q”键，然后再按“Y”键即可退出并保存。执行下面的命令编译固件。

```
make
```

出现下图中的 Creating hex file out/canboot.bin 或 Creating uf2 file out/canboot.uf2 即为编译成功。

```

Compiling out/src/rp2040/can.o
Compiling out/src/rp2040/chipid.o
Compiling out/src/./lib/can2040/can2040.o
Compiling out/src/generic/canserial.o
Compiling out/src/generic/canbus.o
Compiling out/src/./lib/fast-hash/fasthash.o
Building rp2040 stage2 out/stage2.o
Preprocessing out/src/rp2040/rp2040_link.ld
Building out/compile_time_request.o
Linking out/canboot.elf
Creating bin file out/canboot.bin
Creating uf2 file out/canboot.uf2
fly@flygemini:~/CanBoot$

```

5.烧录 CanBoot 引导固件。

(1) 烧录固件前，需要使 SB2040 工具板进入 Boot 烧录模式。方法如下。



(2) 输入以下指令，查看是否有 ID 2e8a:0003 Raspberry Pi RP2 Boot 这行，如没有请检查 USB 线(连接前记得按住 BOOT 键)。

lsusb

```

fly@flygemini:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 002: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 002: ID 1b3f:1167 Generalplus Technology Inc. WEB CAM
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 009 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

```

(3) 烧录。依次输入以下指令，出现下图则表示烧录成功（如果已经在 CanBoot 目录下，则第一条指令可以不执行）。

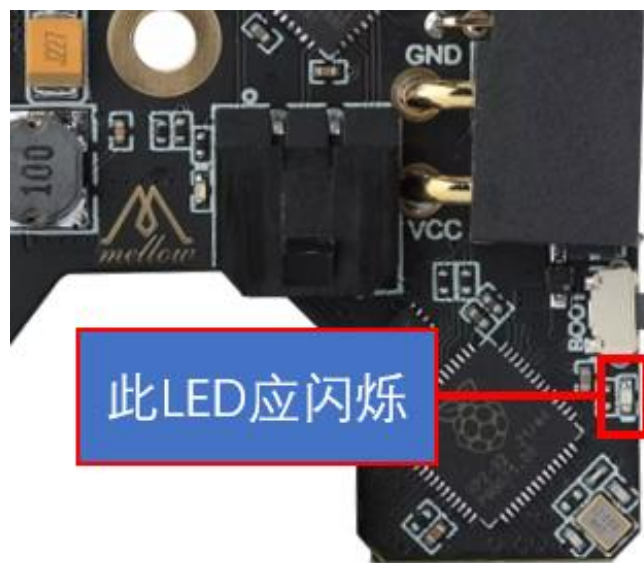
```
cd ~/CanBoot/
```

```
make flash FLASH_DEVICE=2e8a:0003
```

```
[sudo] password for fly:
Loaded UF2 image with 114 pages
Found rp2040 device on USB bus 7 address 2
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
fly@flygemini:~/klipper$
```

5.检查。

如果固件配置正确并烧录成功，则下图中 SB2040 工具板的 LED 灯应该为闪烁状态。



6.查看 SB2040 工具板的 UUID

在控制台输入以下指令。

```
python3 lib/canboot/flash_can.py -q
```

如下图中出现 Detected UUID: fea6a45462e9, Application: Klipper 则正常，(每一块 SHT36/42 的 UUID 都不相同)。

如果没有出现 UUID，请检查 CAN 接线及前面相关步骤是否成功。

```
fly@flygemini:~/klipper$ python3 lib/canboot/flash_can.py -q
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: fea6a45462e9, Application: Klipper
Query Complete
```

7.编译 Klipper 固件。

下面为 SB2040 编译 Klipper 固件。

(1) 连接到 SSH。如何连接到 SSH 请参考：《[Super&VORON 修炼秘籍](#)》--3.4 连接 SSH

(2) 进入固件编译配置界面。依次执行以下指令，即可进入固件编译配置界面。

```
cd ~/klipper && git pull
```

make menuconfig

(3) 固件配置。如下图所示。由于使用了 CanBoot 引导固件，固件配置有所不同。不同之处在下图中标出，编译时请注意修改!!!

```
(Top)
[*] Enable extra low-level configuration options
  Micro-controller Architecture (Raspberry Pi RP2040) --->
  Bootloader offset (16KiB bootloader) --->
  Communication interface (CAN bus) --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(500000) CAN bus speed
(gpio24) GPIO pins to set at micro-controller startup
```

(4) 固件编译。SSH 输入以下指令并回车

make -j4

最后出现** Creating uf2 file out/klipper.bin**则表示编译成功。

```
Compiling out/src/./lib/fast-hash/fasthash.o
Compiling out/src/rp2040/hard_pwm.o
Compiling out/src/rp2040/spi.o
Compiling out/src/rp2040/i2c.o
Preprocessing out/src/generic/armcm_link.ld
Building out/compile_time_request.o
Version: v0.11.0-5-ga42f6158
Linking out/klipper.elf
Creating bin file out/klipper.bin
fly@flygemini:~/klipper$
```

8.通过 CANBUS 烧录 Klipper 固件。

下面命令中的 fea6a45462e9 需要替换为你上一步中得到的 UUID

python3 lib/canboot/flash_can.py -i can0 -f ./out/klipper.bin -u fea6a45462e9

出现下图中的 CAN Flash Success 则表示烧录成功，

```
fly@flygemini:~/klipper$ python3 lib/canboot/flash_can.py -i can0 -f ./out/klipper.bin -u fea6a45462e9
Sending bootloader jump command...
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: fea6a45462e9, Application: CanBoot
Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8002000
MCU type: stm32f072xb
Verifying canbus connection
Flashing '/home/fly/klipper/out/klipper.bin'...

[#####]

Write complete: 12 pages
Verifying (block count = 361)...

[#####]

Verification Complete: SHA = E8B616ABAB17779CBAC3177A4F6DAEA36FBF826D
CAN Flash Success
fly@flygemini:~/klipper$
```

9. 以后如果需要更新 klipper 看下面操作即可。

(1) 拉取最新的 klipper。

```
cd ~/klipper
```

```
git pull
```

(2) 编译最新的 klipper。

```
make menuconfig
```

```
make
```

(3) 为 SB2040 烧录 klipper 固件。下面命令中的 fea6a45462e9 需要替换为你查询到的 UUID。

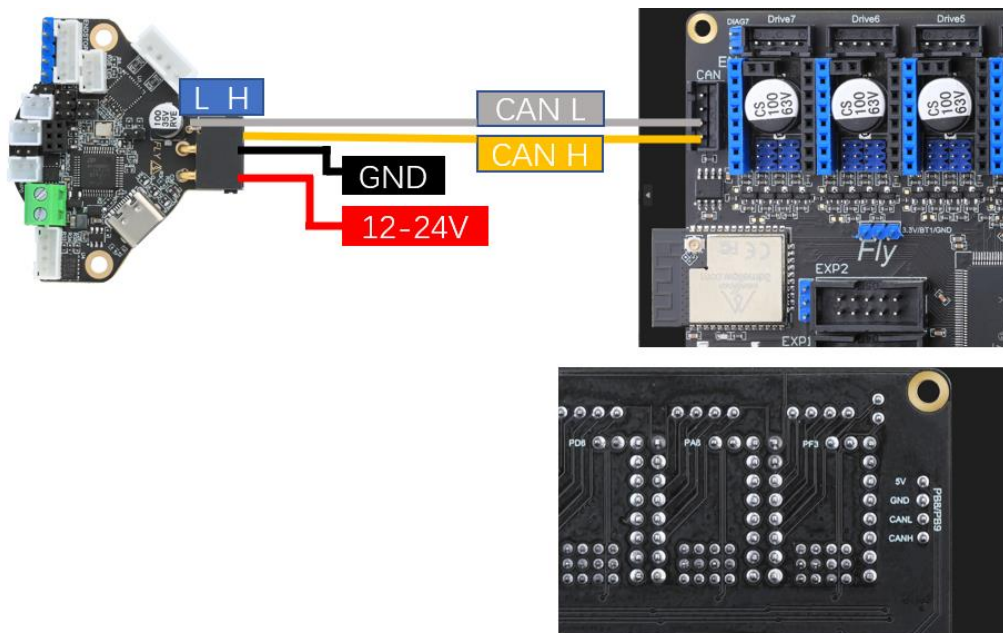
```
python3 lib/canboot/flash_can.py -i can0 -q
```

```
python3 lib/canboot/flash_can.py -i can0 -f ./out/klipper.bin -u fea6a45462e9
```

3.4 桥接 CAN

如果您购买的主板，例如 Super8、Gemini V3 等板载 CAN 收发器的主板，可以不需要使用 UTOC。只需要重新配置和编译固件，然后烧录编译好的固件即可。下面以 Super8 和 SHT36 V2 为例。

接线方法如下图。



Klipper 固件配置界面如下图。编译固件后，按照前面给的方法将固件刷入主板中。重新读取 CAN ID 即可。配置文件在：[FLY SB2040 工具板无需修改固件和接线。](#)

```
[*] Enable extra low-level configuration options
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32F405) --->
Bootloader offset (32KiB bootloader) --->
Clock Reference (8 MHz crystal) --->
Communication interface (USB to CAN bus bridge (USB on PA11/PA12)) --->
CAN bus interface (CAN bus (on PB8/PB9)) --->
USB ids --->
(500000) CAN bus speed
() GPIO pins to set at micro-controller startup (NEW)
```

3.5 加速度计的使用

3.5.1 加速度计配置

SB2040 板载 ADXL345 加速度传感器，只需要进行相应的配置就可以使用。配置方法如下。只需激活 adxl345 部分的配置（如下图所示）并配置好加速度计的位置，保存并重启即可。

```
## ADXL345加速度计
[adxl345]
cs_pin: sb2040:gpio1
spi_software_sclk_pin: sb2040:gpio0
spi_software_mosi_pin: sb2040:gpio3
spi_software_miso_pin: sb2040:gpio2
```

```
[resonance_tester]
accel_chip: adxl345
probe_points:
    100, 100, 20 # 在此处定义你的加速度计位置分别对应x,y,z
```

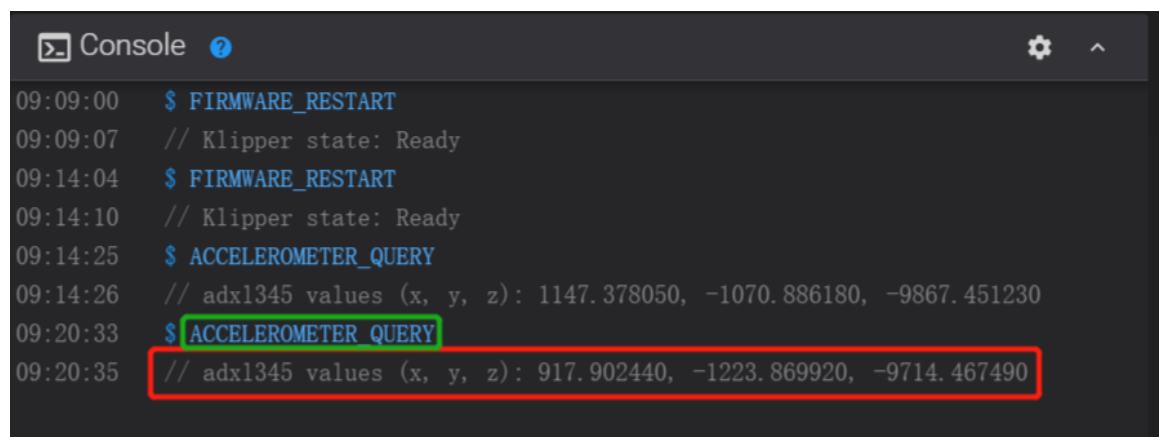
参考配置连接：[FLY-SB2040 \(klipper.cn\)](https://klipper.cn/FLY-SB2040)

3.5.2 加速度计的使用

修改配置并保存重启后，在控制台输入指令：

ACCELEROMETER_QUERY

如果出现报错请检查接线及配置，正常的输出如下图。出现下图中的输出后，便可以进行共振测量了。



```
09:09:00 $ FIRMWARE_RESTART
09:09:07 // Klipper state: Ready
09:14:04 $ FIRMWARE_RESTART
09:14:10 // Klipper state: Ready
09:14:25 $ ACCELEROMETER_QUERY
09:14:26 // adxl345 values (x, y, z): 1147.378050, -1070.886180, -9867.451230
09:20:33 $ ACCELEROMETER_QUERY
09:20:35 // adxl345 values (x, y, z): 917.902440, -1223.869920, -9714.467490
```

测试前需要将 X, Y, Z 全部归位。

测试 X 轴的命令如下：

```
TEST_RESONANCES AXIS=X
```

测试 Y 轴的命令如下：

```
TEST_RESONANCES AXIS=Y
```

注意！如果测试过程中打印机震动太剧烈请及时按紧急停止或发送 M112 来停止。过高的 accel_per_hz 会让振动变得剧烈，可以适当降低。

```
[resonance_tester]
accel_chip: adxl345
accel_per_hz: 50 # 默认值为75, 如果震动太剧烈可以添加这一行来降低震动, 默认不添加, 即保持默认值
probe_points:
    100, 100, 20 # 在此处定义你的加速度计位置分别对应x,y,z
```

4. FAQ

1. UTOC-1 和 UTOC-3 有啥区别啊

答：只能连接带有 CAN 收发器的下位机，3 可以连那些支持 CAN 但是没有 CAN 收发器的下位机（仅 stm32，并且 usb 链接的主控 PA11/PA12），只能连接两个不带收发器的下位机）

2. 板子上是用什么端子？

答：PH2.0, XH2.54, XT3.0 (2+2)

3. Can 支持什么上位机？

答：你的上位机系统内核支持 can 就可以用

4. 蜘蛛或者其他品牌的主板能用吗？

答：可以用。

5. 买 can 板必须搭配 UTOC 用吗？还是有什么需要搭配 UTOC？

答：上位机-usb-带 can 端口的主板-can 工具板，如果主板上没有 can 端口就不用 utoc 了

相关链接

FLY 官方教程网站：<https://mellow.klipper.cn>

FLY CAN 使用教程网站：<https://mellow.klipper.cn/#/advanced/can>

FLY SB2040 相关资料：https://mellow.klipper.cn/#/board/fly_sb2040/

FLY SB2040 原理图、3D 模型和支架：<https://github.com/Mellow-3D/Klipper-CAN-Toolboards>

FLY 3D 官方淘宝店：[首页-FLY 3D 打印主控板-淘宝网 \(taobao.com\)](#)

SB2040 详情及购买链接：[FLY3D 打印机 Voron 2.4 R2 Trident Stealthburner Can 工具头主板-淘宝网 \(taobao.com\)](#)

FLY 官方售后技术支持 QQ 群：621032883

