

Nitehawk-SB 头板



介绍：

Nitehawk-SB 是 LDO 公司专为 Voron Stealthburner 打印头设计的一款工具板。它是一块集成了 RP2040 MCU、TMC2209 步进驱动和 ADXL345 加速计的工具头 PCB。Nitehawk 外形是根据 HartK 分体式 Stealthburner 打印头双板的 PCB 来设计的。

Nitehawk 采用 USB 与树莓派(RPI)进行通信,而非 CAN 端口。因此,对于大多数安装过 Klipper 打印机的玩家来说,软件和硬件设置都要简单和熟悉得多。

特征：

布线方便, 无需复杂且容易出错的分线电缆, Nitehawk 只需接入 24V 电源并通过 USB 连接到 RPI 主机即可。

通过 USB Klipper 连接, 无需额外的软件或硬件设置, 相比 CAN 更加简便。

定制工具头电缆, 采用单一的 USB 数据与电源合成电缆, 不仅能搭配拖链使用, 也适用 umbilical 配置。

便捷的共振补偿技术, 利用板载加速计可实时校准。

风扇配备转速计, 热端和部件风扇端口均兼容三针转速计, 可提供附加的诊断功能和安全保障。

系统概述：

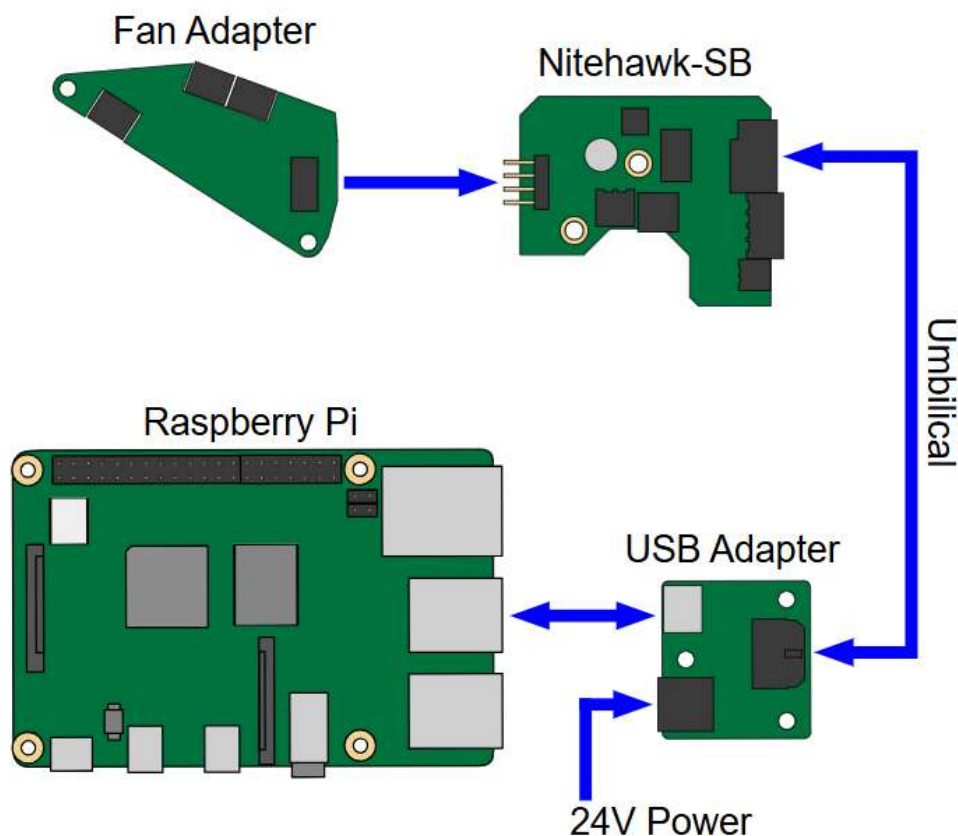
Nitehawk 系统由三块 PCB 板和一条 Umbilical 电缆组成，简化接线图如图所示：

1.Nitehawk-SB，作为核心 PCB 板，集成 MCU 微控制器、步进电机驱动器、风扇驱动电路及其他功能电路。

2.Fan Adapter，是一块简易的 PCB 板，负责连接部件冷却风扇、热端风扇和 NeoPixel 灯带，通过排针与主控板对接。此分体式设计主要目的是使 Stealthburner 喷头前端可快速拆卸，便于维护作业。

3.Umbilical 电缆，专为拖链应用设计的定制柔性线缆，同时具备向主控板传输 24V 电源及承载 USB 数据信号的功能。

4.USB Adapter，是一块功能整合型 PCB 板，将电源 24V 供电与树莓派(Raspberry Pi)USB 数据信号合并，通过 umbilical 电缆实现与主控板的统一连接。



打印件：

Nitehawk-SB 适配标准 Stealthburner 打印头，同时兼容 Clockwork 2 和 Galileo 2 挤出机。

但需额外打印少量定制结构件。相关模型文件可在相关随附文件中找到。

1.USB Adapter Mount，是用于安装 USB Adapter PCB 板的支架。它可与标准 Voron DIN 卡扣兼容，并支持两种不同的安装方向。同时还配备一个防护盖板，用于减少静电放电对 PCB 板的影响。

2.Cable Door，此打印件在 Hartk 原始设计的基础上略作修改，新增了一个用于外部腔体热敏电阻的扎带固定点。该门板还采用 M3 防丢螺丝作为扣件设计，避免在打开门板时容易松动。

3.Chain Anchor Tilted，此打印件可以使拖链略微倾斜，以帮助其避开左侧 XY 关节，并防止拖链与工具板上的 CT 接头摩擦。

Klipper 配置文件

请参考附带的另一个文档。

如果你不打算使用转速计（Tachometer）的风扇，请务必注释掉以下配置参数：

tachometer_pin（转速计信号引脚）

tachometer_ppr（转速计每转脉冲数）

否则，Klipper 在运行时可能会报错或无法正常控制风扇。

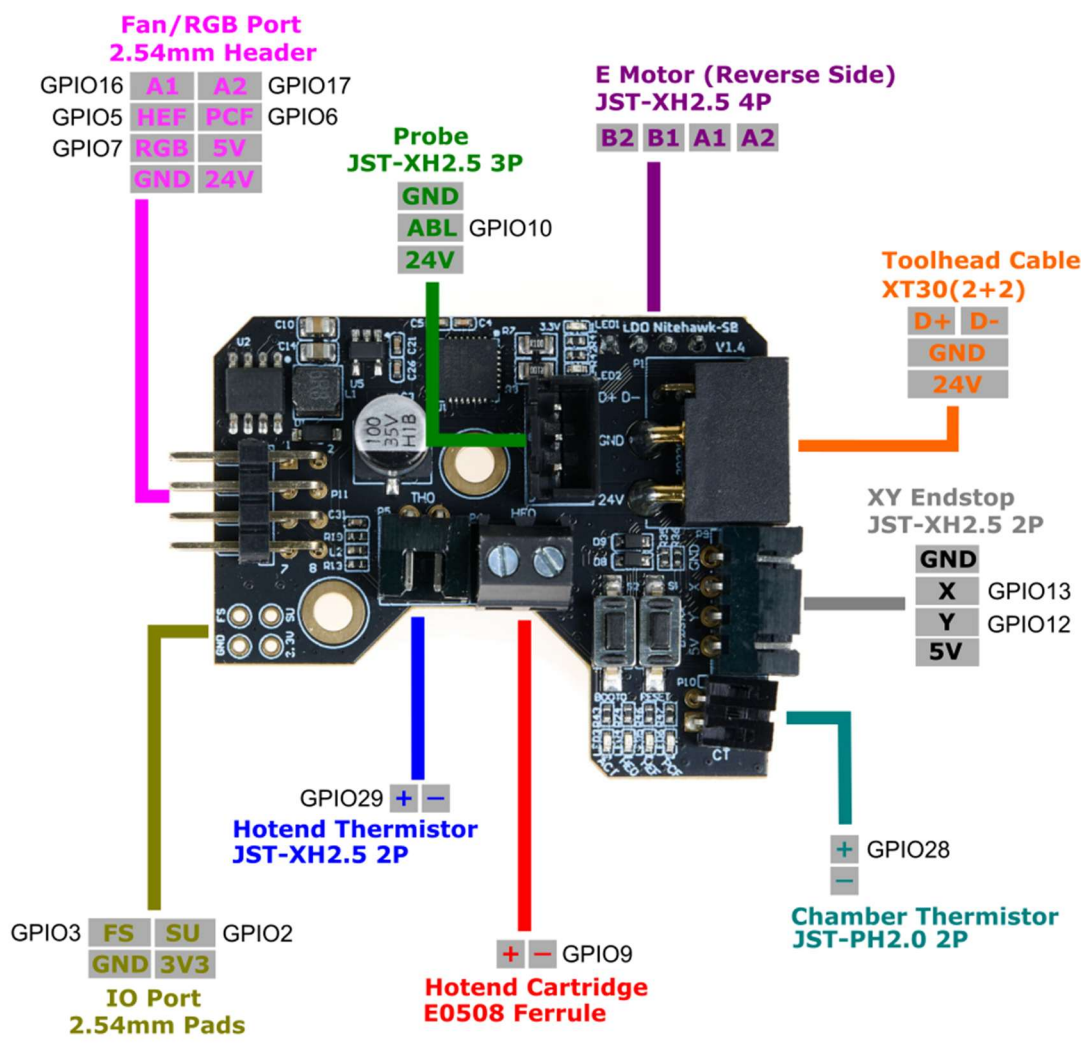
部件风扇检查宏

在 GitHub 代码库中提供了一个可选的 tacho_macros.cfg 文件，该文件包含宏指令

“PREFLIGHT_CHECK”。该宏会短暂启动部件风扇，并通过转速计信号检测风扇是否正常运转。

要使用该宏指令，在 printer.cfg 文件中添加 [include tacho_macros.cfg] 以引入配置，并在 “PRINT_START” 宏指令的开头调用 “PREFLIGHT_CHECK” 。

端口及引脚定义

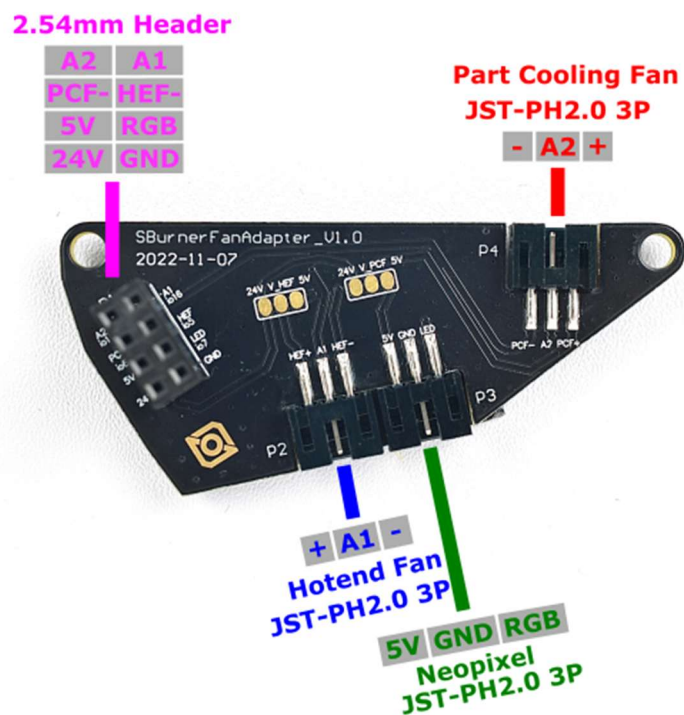


| 设备/端口 | PCB 板标签 | 端子类型 | RP2040 引脚 | 描述 |
|--------|---------|-----------|-----------------|---|
| X/Y 限位 | Endstop | JST-XH 4P | gpio13/12 (X/Y) | 连接 X 和 Y 限位器。X 端限位是 gpio13，Y 端限位是 gpio12。 |

| | | | | |
|--------|-----------------------------|-----------------|-----------------------------|---|
| 热端加热器 | HE0 | 螺丝端子 / E0506 | gpio 9 | 连接热端加热器。 |
| 热端热敏电阻 | TH0 | JST-XH 2P | gpio 29 | 连接热端热敏电阻。 使用 2.2k Ω 上拉电阻。 |
| 部件冷却风扇 | PCF (Fan adapter PCB) | JST-PH 3P | gpio6/17 (fan/tacho) | 连接部分冷却风扇 (通过 fan adapter PCB)。风扇控制引脚 为 gpio6, 转速传感 器引脚为 gpio17。 |
| 热端风扇 | HEF (Fan adapter PCB) | JST-PH 3P | gpio5/16 (fan/tacho) | 连接热端风扇 (通过 fan adapter PCB)。 风 扇 控 制 引 脚 为 gpio5, 转速传感器引 脚为 gpio16。 |
| 腔体热敏电阻 | CT | JST-PH 2P | gpio 28 | 腔体热敏电阻端口。 连接外部热敏电阻。 使用 4.7k Ω 上拉电 阻。 |
| 探头 | PROBE | JST-XH 3P | gpio 10 | 探头用于床身调平/或 Z 轴探测。仅 24V 电 源。 |

| | | | | |
|-----------------|------------------------------|-----------|--|---|
| 电机 | MOTOR | JST-XH 4P | gpio23/24/25/0/1 (step/dir/ena/uart/ tx) | 挤出机步进电机驱动 端口, 搭载 TMC2209 驱动芯片, 配置 100mΩ 电流采样电 阻, 使能信号低电平 有效。 |
| Neopixel 灯 带 | LED (Fan adapter PCB) | JST-PH 3P | gpio 7 | 连接 Neopixel 灯带 (通过 fan adapter PCB) 。 |
| 活动指示灯 | ACT | N/A | gpio 8 | 板载 LED (低电平有 效) 。 |
| 加速计 | N/A | N/A | gpio21/18/20/19 (cs/clock/mosi/miso) | ADXL345 加速度计 (软件 SPI 控制) , 用于共振补偿功能。 |

Fan Adapter PCB 的引脚



本表格列明了工具头电路板各端口的详细功能说明。

| PCB 板标签 | 名称 | 端子类型 | 描述 |
|--------------|------------|-----------|------------------------------------|
| HEF+/A1/HEF- | 热端风扇* | JST-PH-3P | 用于冷却热端散热器的风扇端口。可选 24V/5V, 默认为 24V。 |
| PCF+/A2/PCF- | 部件冷却风扇* | JST-PH-3P | 部件冷却风扇端口。可选 24V/5V, 默认为 24V。 |
| 5V/GND/LED | Neopixel 灯 | JST-PH-3P | 连接 Stealthburner 的 Neopixel 灯带接口。 |

| | | | |
|--|-------|--------------|------------------|
| | 工具头电缆 | 2x8 F-Header | 连接至主打印头电路板公端接插件。 |
|--|-------|--------------|------------------|

*热端风扇与部件风扇可独立切换至 5V 供电，操作分两步：

- 1.切断 24V 连接 - 划断 24V 焊盘与 V_HEF/V_PCF 焊盘间的走线。
- 2.建立 5V 连接 - 在 5V 焊盘与 V_HEF/V_PCF 焊盘间进行桥接焊接。

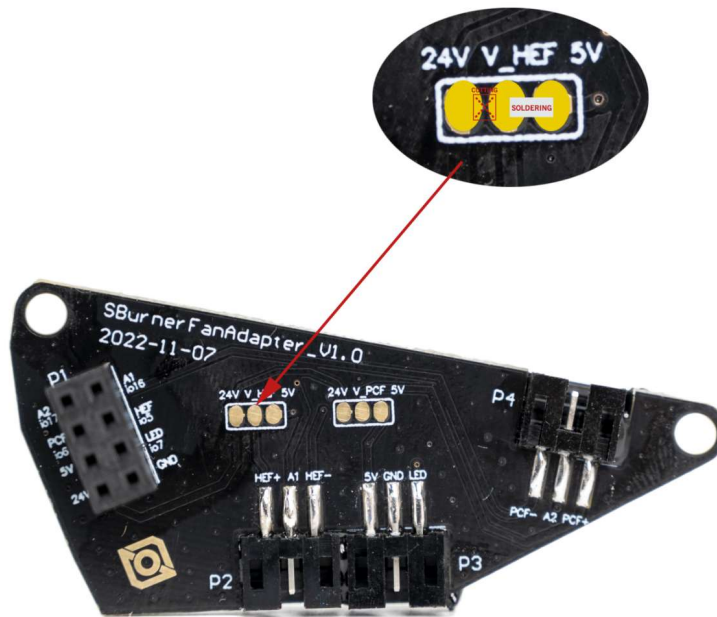
*步进电机电流设置建议：

若电机额定电流为 1A，建议运行电流设置为 0.7A 以下；如需超过 0.7A，请为控制板增加散热措施。

如需选择 5V 供电，请如根据图示进行操作：

- 1.划断 24V 与 HEF 间的走线
- 2.将 HEF 与 5V 焊盘进行桥接

(PCF 操作同理)



Umbilical 电缆

接头类型（工具板端）

工具板端所使用的连接器为 Amass XT30(2+2)-F。目前市面上存在三种不同的变体，均与 Nitehawk-36、Nitehawk-SB 和 Orbitool 工具头板兼容，但在外观结构和柔韧性方面存在差异：

1. **全包胶 (Fully Overmolded)**，此版本为 Nitehawk-36 Kit **Rev C** 及更高版本的默认配置，具备良好的防拉扯能力，但在接头末端的柔性不如第 2 和第 3 种变体。

2. **半包胶 (Partially Overmolded)**，此为 Nitehawk-SB 和 Voron 套件的默认出厂版本。该版本在接头端保留一段裸露导线，使其更容易在工具板端弯折（适用于 Stealthburner 打印头搭配线缆拖链使用）

3. **灌封型 (Potted)** , 此为早期 Nitehawk-36 V1.2 和 Nitehawk-SB 套件随附的版本。现已由上述第 1 和第 2 种变体取代。

*电缆的额定弯曲半径为 28mm；电缆的额定工作温度为 105°C 。



电缆引脚 (工具板端)

下图展示 XT30 接头详细引脚分布：



| Pin# | 名称 | 颜色 | 描述 |
|------|-------------------|-------|---|
| 1 | GND/Shielding（屏蔽） | 黑色 | 电缆屏蔽层与 PCB 板地线共用此引脚。 |
| 2 | 24V | 红色 | 工具板 24V 电源输入接口。 |
| 3 | D+ | 棕色/绿色 | USB 数据正极。该导线的颜色可能因生产批次不同而有所差异，可能为棕色或绿色。 |
| 4 | D- | 白色 | USB 数据负极。 |

电缆引脚分布（适配器端）

Micro-Fit 3.0 是 umbilical 电缆中 USB 适配器端所使用的连接器。该连接器易于压接，支持电缆长度灵活剪裁。**切勿在设备通电状态下插拔此连接器！** 否则可能会对 Nitehawk 或主控设备（如 Raspberry Pi）造成损坏。下图展示了该连接器的详细引脚定义。



| Pin# | 名称 | 颜色 | 描述 |
|------|---------------------------|-------|---|
| 1 | NC/GND | / | 该引脚不连接电缆端，但在 PCB 板上与接地端相连。 |
| 2 | NC/Shielding (屏蔽) /GND | / | 线缆屏蔽层连接至 PCB 板地线。新版设计中该引脚将不再使用。 |
| 3 | GND | 黑色 | 新版设计中，地线与屏蔽层共用同一导线。 |
| 4 | D+ | 棕色/绿色 | USB 数据正极。该导线的颜色可能因生产批次不同而有所差异，可能为棕色或绿色。 |
| 5 | D- | 白色 | USB 数据负极。 |

| | | | |
|---|-----|----|-----------------|
| 6 | 24V | 红色 | 工具板 24V 电源输入接口。 |
|---|-----|----|-----------------|

电气规格

| 参数 | 符号 | 最小值 | 标准值 | 最大值 | 单位 | 注解 |
|----------------------|-----------|-----|-----|-----|----|--|
| 输入电源 | V_{in} | 20 | 24 | 28 | V | 工 具 板 输 入电源 |
| 5V 电流 | I_{rpi} | | | 5 | A | 5V 降压转 换 器 的 输 出电流 |
| 风 扇 电 流 (HEF,PCF) | I_{fan} | | | TBD | A | 各 风 扇 接 口 电 流 额 定值 (HEF 与 PCF 端 口) |
| 热端电流 | I_{he} | | | TBD | A | 受 MOSFET 最 大 持 续 电流限制 |

固件设置及更新

Nitehawk 固件由两个主要组件组成：Katapult 和 Klipper。

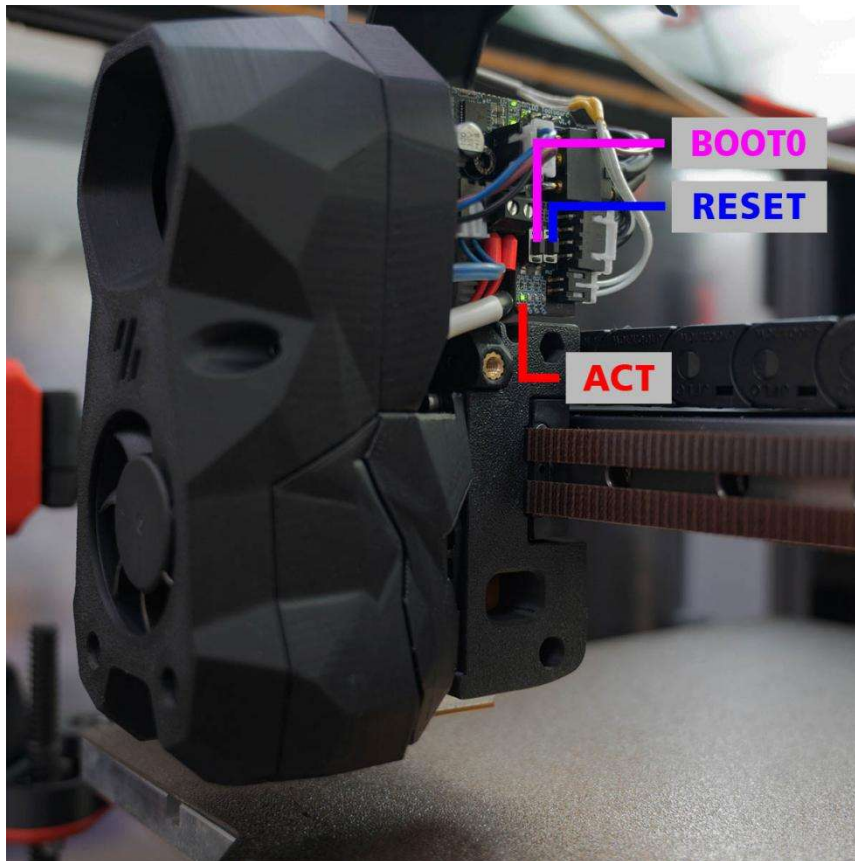
Katapult 是专为 Klipper 设计的引导加载程序 (Bootloader)，其主要功能是确保 RP2040 微控制器 (MCU) 上的软件能够顺利启动，并支持对 Klipper 固件的便捷更新。您可以在[此处](#)了解更多关于 Katapult 的信息。

Klipper 是运行在 RP2040 微控制器上的核心固件，负责控制 3D 打印机的主要功能。它以高性能和灵活的配置方式广泛应用于当前的 FDM 打印设备中。您可以在[此处](#)了解更多关于 Klipper 的信息。

Nitehawk 出厂时已预装 Katapult 和 Klipper 两个组件。通常情况下，用户仅需定期更新 Klipper 固件，无需对 Katapult 进行任何操作。如果 Katapult 引导程序被意外擦除或缺失，可参考本手册相关章节，按照说明重新烧录 Katapult。

硬件设置

在安装 Klipper 或 Katapult 时，无需进行任何特殊设置。Nitehawk 作为 3D 打印机中的常规部件，只需通过 USB 适配器将工具板连接至 Klipper 主控设备 (例如 Raspberry Pi) 即可。同时，需要能够操作工具板上的两个按钮 (RESET 和 BOOT0)，通常是在将打印头移动至前方并打开打印头盖进行操作。此外，请注意按钮下方的四个 LED 指示灯——最左侧的 LED 为 ACT 状态指示灯，后续操作中 will 起到关键作用。请参考下方照片，以识别按钮及 LED 的具体位置。



编译 Klipper 固件

以下是 Nitehawk 工具板编译及上传新版 Klipper 固件的操作说明。如果您希望将 Klipper 固件更新至最新版本，或者您是首次安装且已上传 Katapult（详见前文章节），则需要执行以下步骤。在编译固件之前，请确保您的主控设备（例如 Raspberry Pi）已成功安装 Klipper。

- 通过 SSH 登录至您的 Klipper 主机。Windows 用户可使用 Putty 或其他任意 SSH 客户端，Mac 与 Linux 用户则可直接在终端中使用 ssh 命令连接。登录后，运行以下命令以打开固件配置界面：

```
1 | cd ~/klipper
2 | make menuconfig
```

- 在配置器中启用附加的低层级配置选项，选择 “Raspberry Pi RP2040” 作为目标平台，并根据下方所示截图对其余参数进行一致配置。

```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
  Micro-controller Architecture (Raspberry Pi RP2040) --->
  Bootloader offset (16KiB bootloader) --->
  Communication interface (USB) --->
  USB ids --->
(!gpio8) GPIO pins to set at micro-controller startup

[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)    [ESC] Leave menu
```

- 最重要的一点：请务必设置 16KiB 的引导加载程序偏移地址。否则，将会擦除 Katapult 引导加载程序！
- 完成后，按 Q 退出，并在系统提示时选择 Yes 以确认保存设置。随后，运行以下命令以生成固件文件：


```
1 | make clean
2 | make
```

- 固件文件将被生成并保存在目录 `~/klipper/out` 中。接下来，您可以将该固件上传至 Nitehawk 工具板。推荐的固件上传方式是使用 `make flash` 命令。

上传 Klipper (通过 make flash)

- 运行 `ls /dev/serial/by-id` 查找 Nitehawk 工具板的 USB ID。USB ID 的格式应类似于：
`usb-Klipper_rp2040_1234567890000000-if00`。
- 接下来运行以下命令，这将安装 Python、pip 以及 pyserial 模块（若尚未安装）。在执行最后一条命令时，您可能会收到 “externally managed environment”（外部环境管理）的错误提示。这是正常现象，表示 pyserial 模块可能已预先安装，您可直接跳过该步骤并继续后续操作。

```
1 | sudo apt install python3 python3-pip
2 | pip install pyserial
```

- 运行以下指令以将固件直接上传至微控制器（MCU）。

```
1 | cd ~/klipper
2 | sudo service klipper stop
3 | make flash FLASH_DEVICE=/dev/serial/by-id/<your USB ID>
4 | sudo service klipper start
```

- 如果在刷新固件后遇到连接问题，请重启打印机。此时，您的 Frame PCB 板应已成功更新为最新固件版本。如果刷新过程失败，建议尝试使用第二种方法，通过 Katapult 上传 Klipper 固件。

上传 Klipper (通过 Katapult)

在本节中，我们将使用 Katapult 引导加载程序作为替代方法来上传 Klipper 固件。如果您的工具板由于某种原因缺少 Katapult 引导加载程序，请先参考下一节内容安装该引导程序。

- 首先，我们将检查并在必要时安装 Katapult 软件包。

```
1 | test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone https://github.com/Arksine/katapult) ; cd ~
```

```
test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone  
https://github.com/Arksine/katapult) ; cd ~
```

- 要上传 Klipper 固件，我们将使用 Python 脚本与 Nitehawk 工具板内的 Katapult 引导加载程序进行通信。首先需要搭建一个 Python3 环境。请运行以下命令：

```
1 | virtualenv -p python3 ~/katapult-env  
2 | ~/katapult-env/bin/pip3 install pyserial
```

- 此步骤将在 ~/katapult-env/ 路径下创建一个 Python3 环境，并安装上传脚本所需的 pyserial 模块。

- 接下来，需要强制 Nitehawk 进入 Katapult 引导加载程序并获取 USB 串行地址。请快速双击复位 (**RESET**) 按钮两次，此时应观察到活动指示灯 (**ACT**) 缓慢闪烁。这表明设备已成功进入 Katapult 引导加载模式。

- 然后，执行 `ls /dev/serial/by-id/` 命令。正常情况下将显示类似 `usb-katapult_rp2040_A1234567898D1234-if00` 的地址（注意地址中包含 "katapult" 标识）。若未显示，则表明设备未安装 Katapult 或未成功进入 Katapult 引导加载程序。请记录该地址用于后续操作，此时先不要退出引导加载程序。

- 最后，运行以下命令（需将命令中的地址替换为上一步获取的实际地址）。若操作无误，您将看到写入和验证进度提示，并在最终显示 "Flash Success" 烧录成功的信息。

```
1 | ~/katapult-env/bin/python3 ~/katapult/scripts/flashtool.py -d /dev/serial/by-id/usb-katapult_rp2040_A1234567898D1234-if00
```

```
~/katapult-env/bin/python3 ~/katapult/scripts/flashtool.py -d  
/dev/serial/by-id/usb-katapult_rp2040_A1234567898D1234-if00
```

- 作为最终验证，再次执行 `ls /dev/serial/by-id/`，此时应显示 Klipper 的 USB 串行地址，格式为 `usb-Klipper_rp2040_E1234567A12D9835-if00`。

安装 Katapult 引导加载程序

本节将指导您完成 Katapult 引导加载程序的编译与烧录流程。请注意：您的 Nitehawk 工具板出厂时已预装 Katapult 系统，仅当 Katapult 被意外擦除或缺失时，才需要执行以下操作。

- 登录 Raspberry Pi。我们将检查并在必要时安装 Katapult 软件包：

```
1 | test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone https://github.com/Arksine/katapult) ; cd ~
```

```
test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone
```

```
https://github.com/Arksine/katapult) ; cd ~
```

- 现在，我们需要配置一些选项：

```
1 | cd ~/katapult
2 | make menuconfig
```

- 进入配置菜单。确保设置如下选项：

```
(Top)
Katapult Configuration v0.0.1-64-g3e23332
Micro-controller Architecture (Raspberry Pi RP2040) --->
Flash chip (W25Q080 with CLKDIV 2) --->
Build Katapult deployment application (Do not build) --->
Communication interface (USB) --->
USB ids --->
() 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
(!gpio8) Status LED GPIO Pin

[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)    [ESC] Leave menu
```

- 完成后，按 Q 退出，并在系统提示时选择 Yes 以确认保存设置。随后，运行以下命令以

生成 Katapult 二进制文件：

```
1 | make clean
2 | make
```

- 此时将在 ~/katapult/out/ 路径下生成名为 katapult.uf2 的二进制文件。下一步需将该文件烧录至 Nitehawk 工具板搭载的 RP2040 微控制器 中。

- 现在需要将 Nitehawk 工具板重启进入系统引导模式。操作分为三个步骤：

1.同时长按 **RESET** 和 **BOOT0** 按钮

2.先放开 **RESET** 按钮

3.再放开 **BOOT0** 按钮

- 如果操作无误，您的 Nitehawk 此时应已进入 Boot 模式并模拟为"半 U 盘设备"。请执行命令 `ls /dev/sda*` 验证，正常情况下应显示类似 `/dev/sda /dev/sda1` 的分区节点。若显示 `ls: cannot access '/dev/sda*': No such file or directory` 错误，则表明：

Nitehawk 未成功进入 Boot 模式或 Raspberry Pi 与 Nitehawk 之间的物理连接存在故障。

- 现在可以开始上传 Katapult 程序了，请执行以下指令：

```
1 | sudo mkdir -p /mnt/pico
2 | sudo mount /dev/sda1 /mnt/pico
3 | sudo cp ~/katapult/out/katapult.uf2 /mnt/pico
4 | sudo sync
5 | sudo umount /mnt/pico
```

- 上述命令本质上将 Nitehawk 挂载为存储驱动器，并将 Katapult 二进制文件复制到该驱动器中，随后卸载驱动器。若操作顺利，此时应可观察到 **BOOT0** 按钮下方的 **ACT** 指示灯进入慢速闪烁状态。为确认 Katapult 安装状态，请执行 `ls /dev/serial/by-id` 命令。正常情况下将显示类似以下格式的地址：`usb-katapult_rp2040_A1234567898D1234-if00`，此即为运行 Katapult 的 Nitehawk 的 USB 串行地址。需特别注意以下技术细节：

1.该地址仅在 Nitehawk 处于 Katapult 引导加载模式时可见,系统启动模式或正常运行 Klipper 固件时不会显示。

2.可通过快速双击 Nitehawk 工具板上的 **RESET** 按钮强制进入 Katapult 引导加载模式(此模式下 **ACT** 指示灯呈前述慢速闪烁状态)。

3.后续上传 Klipper 固件时,必须确保设备处于 Katapult 引导加载模式并获取该 USB 串行地址。

4.按照上述说明操作后，你将成功上传 Katapult 程序，但会擦除所有其他固件，包括之前安装的任何 Klipper 固件。

5.单按 **RESET** 按钮可退出 Katapult 引导加载模式（正常情况下应进入 Klipper 固件，但由于刚刚擦除了 Klipper，设备将再次返回 Katapult 引导加载模式）

现在就可以编译并上传 Klipper 了--请参阅此处的上一节。