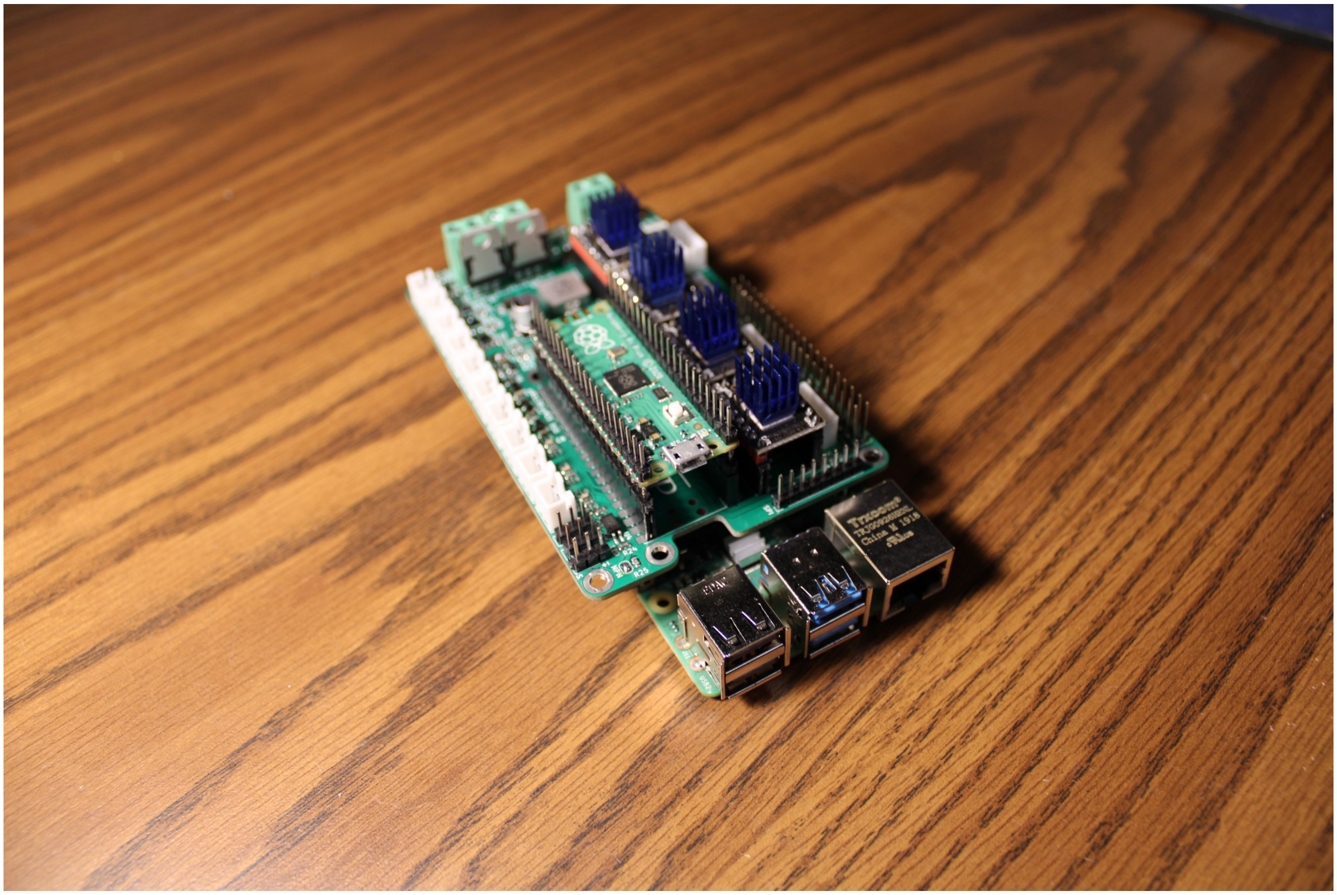
Strawberry Cookie Manual

Introduction

Hello and thank you for choosing the Strawberry Cookie as your next 3D print board! The Strawberry Cookie is vastly different from other boards out there. This is a fully Klipper-compatible HAT for the Raspberry Pi - a first of its kind, and is meant to be a drop-in replacement for many different types of 3D printers, including popular Cartesian and Delta-style printers. This can also be used as an expansion board, allowing for users to use their existing boards in conjunction with the Strawberry Cookie for a multi-MCU approach to control.



Hardware

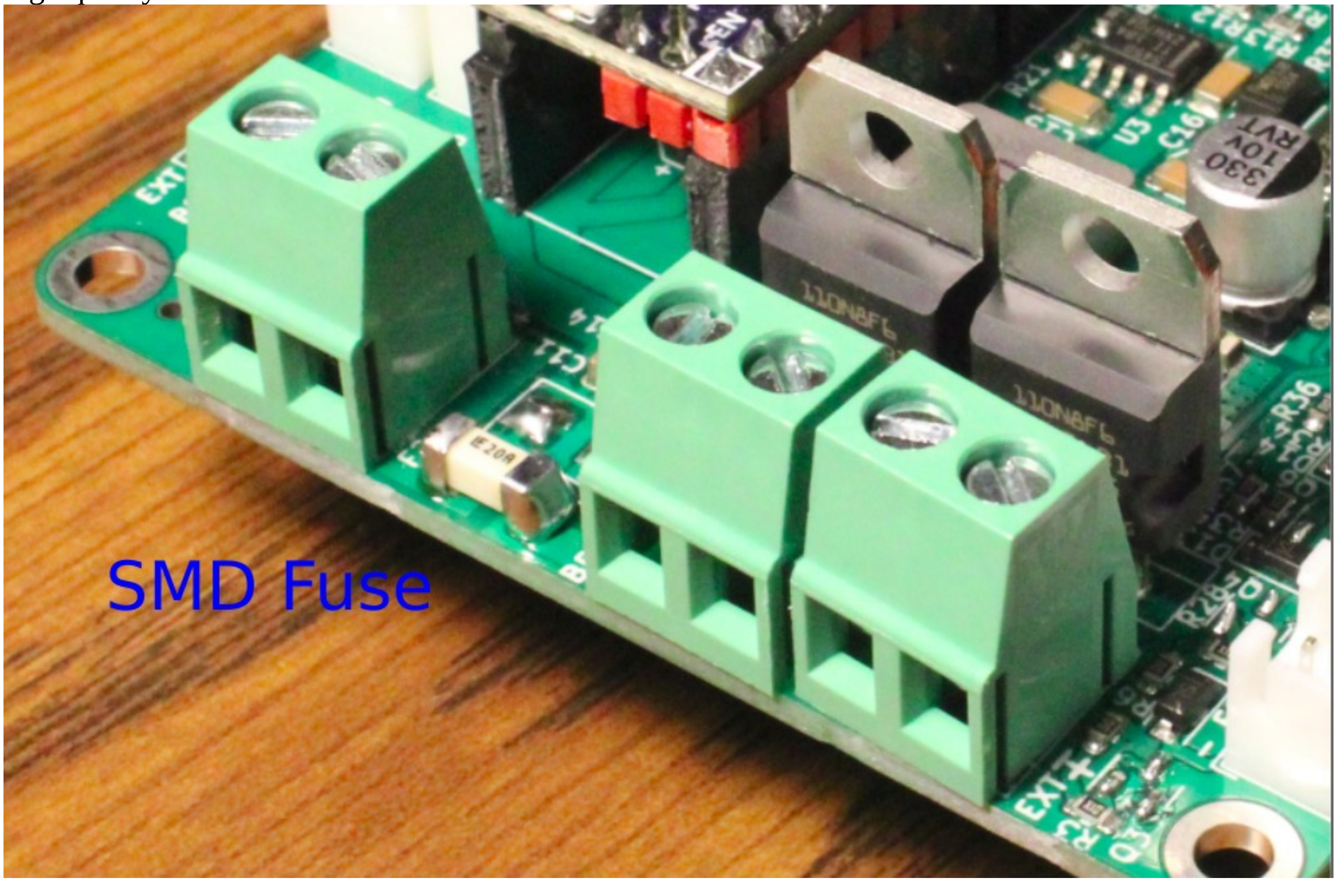
All necessary hard-ware related information can be found on the GitHub: https://github.com/LRFPV2/Strawberry_Cookie

Your Strawberry Cookie comes with the following features:

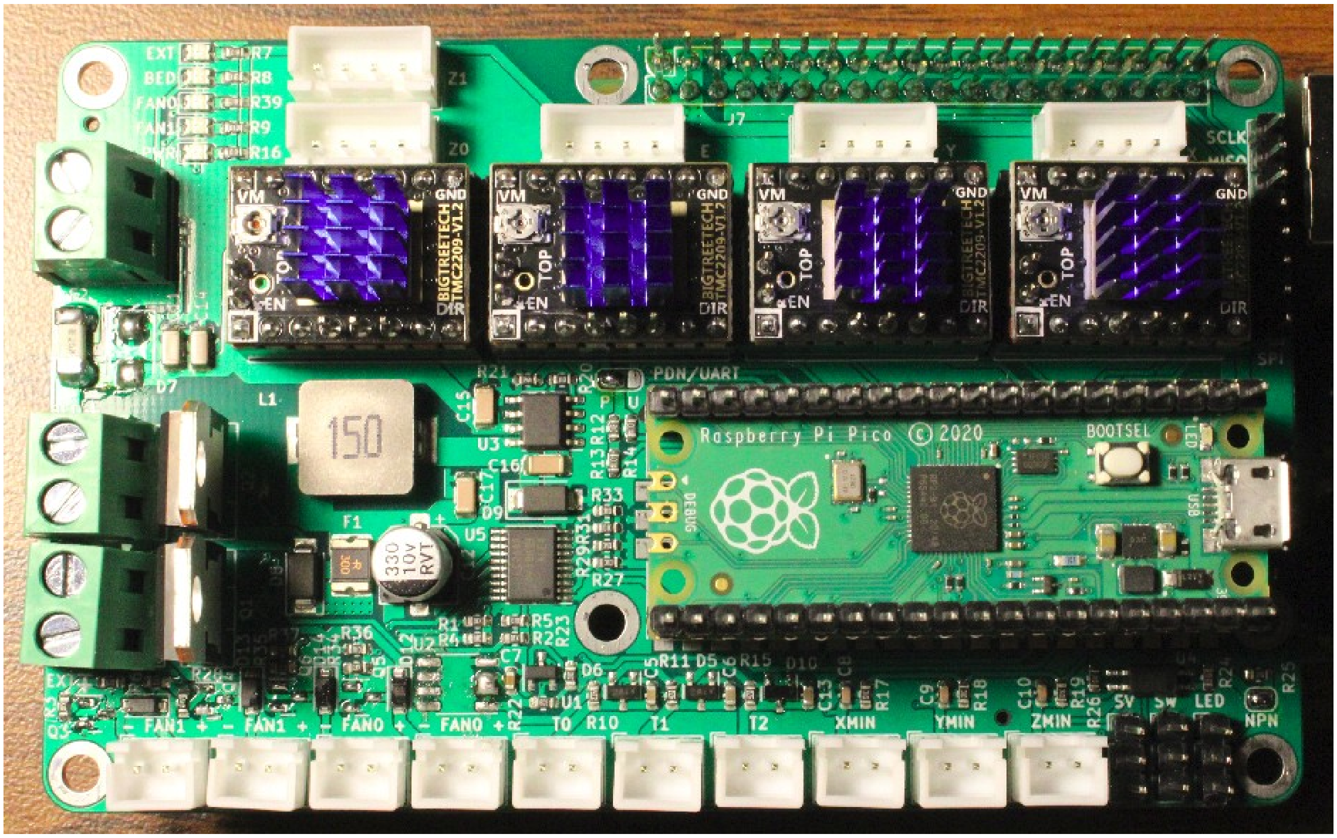
4x replaceable TMC 2209 support
4x fan ports (2 PWM channels)
3x temperature sensors
2x heaters
X, Y, Z min endstops
Dual Z-axis motor support
High quality 200W MOSFETs
BlTouch support
PL08 probe support (NPN/PNP)

LED strip support
SPI breakout
24V compatible
5V, 3A regulator

Assembly Notes
1) Solder 20A SMD Fuse. This should be labeled as “F2”, and is located at the edge of the board, between the power connector and heat bed power output. Make sure to use plenty of solder to ensure a high quality connection.



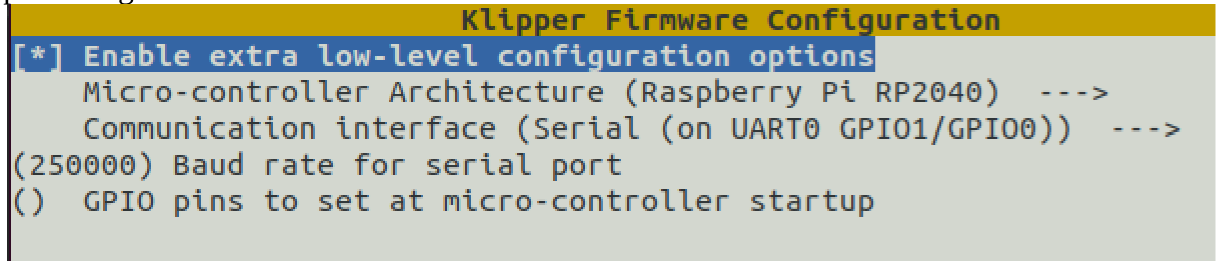
2) Solder terminal block connections. At this point, it would be a good idea to do a power-on test. Simply provide up to 24V to the power input (polarity should be labeled). If the red LED, labeled “PWR” in the upper left-hand corner turns on, that means that you are good to move on.
3) Solder in stepper driver headers, Pi headers servo, probe, and LED connectors. Add in JST-XH connections. SPI connection optional.
4) Solder MOSFETs. Be sure to check that orientation is correct. The metal side of the MOSFET should be pointing *away* from the terminal block connectors.
5) The Raspberry Pi Pico can be mounted in a number of ways. I prefer directly soldering it to the board. This yields a very slim profile. However, some people may want to remove the Pico from time to time. The use of additional female headers will work as well.



Software Setup

Please refer to the sample printer.cfg file as well and the schematic PDF printout for the most accurate wiring information.

The Strawberry Cookie has a built-in UART interface between the Pico and Raspberry Pi. Before you will be able to communicate to the Pico, you will need to flash it via USB for UART communication. With the shield unplugged from the Raspberry Pi, flash your Pico with the following Klipper configuration:



```
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040)  --->
    Communication interface (Serial (on UART0 GPIO1/GPIO0))  --->
(250000) Baud rate for serial port
()  GPIO pins to set at micro-controller startup
```

This will output a “klipper.uf2” file. Copy this to the Pico as it appears as a mass storage device. If you are not seeing the Pico as an option, make sure that you first hold the white “BOOTSEL” button while plugging the Pico via USB.

Next, here is the basic setup for UART communication in Klipper.

1)

sudo raspi-config

Interface options → Serial port → would you like a login shell to be accessible over serial port → No

Interface options → Serial port → would you like the serial hardware port to be enabled → Yes

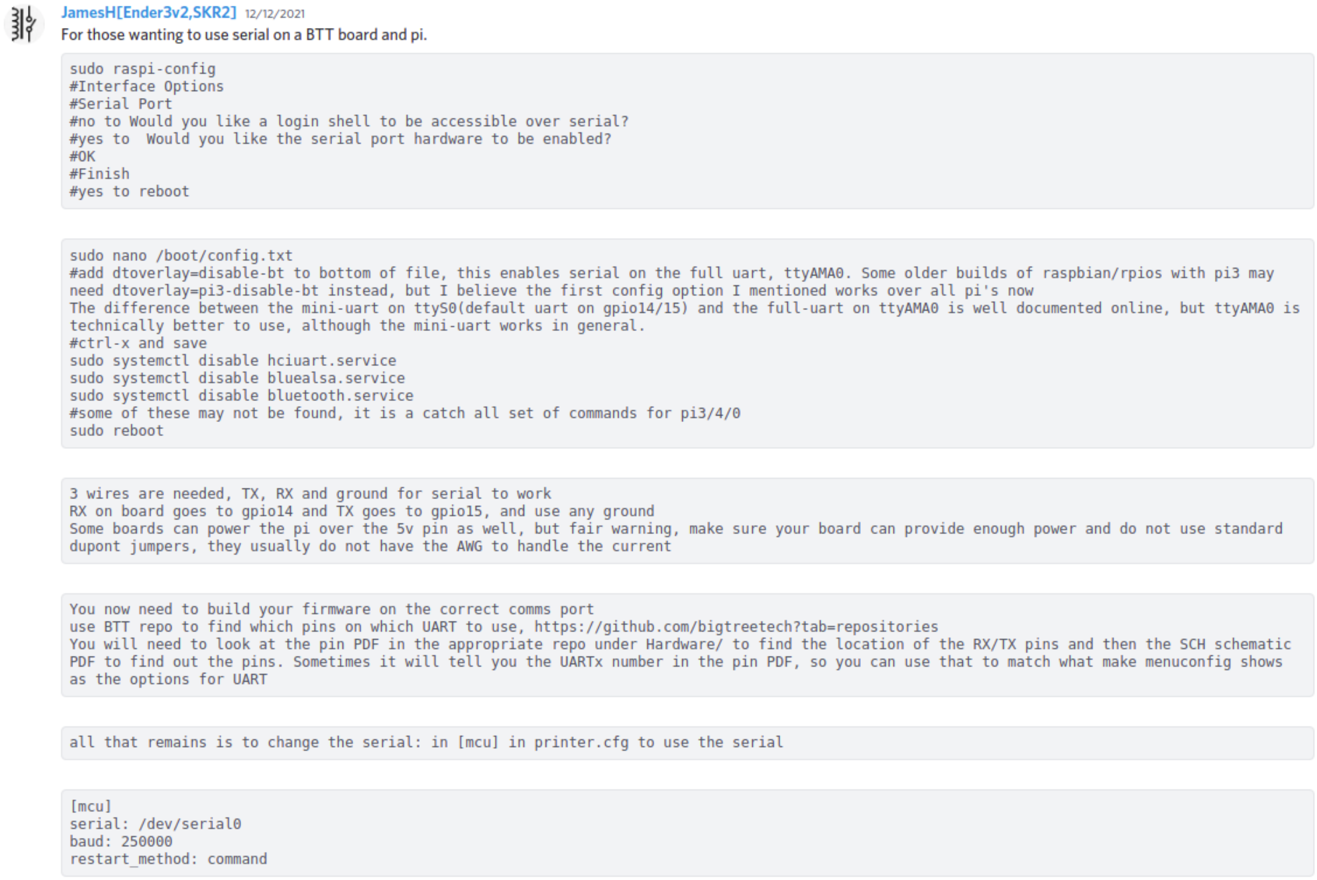
Reboot

2)

sudo vim /boot/config.txt

Add this to the bottom of the file:
dtoverlay=disable-bt
3)
sudo vim /boot/cmdline.txt
Remove the following from the string of text: “console=serial0,115200”
Reboot
4) Serial port for Klipepr communication should be “/dev/ttyAMA0” (in sample printer.cfg)

For more information, Nero3DP made a great video outlining this with a Raspberry Pi Zero:
https://www.youtube.com/watch?v=AtW3GqkKUz8
Additionally, JamesH on the Klipper Discord has posted some wonderful instructions:



JamesH[Ender3v2,SKR2] 12/12/2021
For those wanting to use serial on a BTT board and pi.

```
sudo raspi-config
#Interface Options
#Serial Port
#no to Would you like a login shell to be accessible over serial?
#yes to  Would you like the serial port hardware to be enabled?
#OK
#Finish
#yes to reboot
```

```
sudo nano /boot/config.txt
#add dtoverlay=disable-bt to bottom of file, this enables serial on the full uart, ttyAMA0. Some older builds of raspbian/rpios with pi3 may
need dtoverlay=pi3-disable-bt instead, but I believe the first config option I mentioned works over all pi's now
The difference between the mini-uart on ttyS0(default uart on gpio14/15) and the full-uart on ttyAMA0 is well documented online, but ttyAMA0 is
technically better to use, although the mini-uart works in general.
#ctrl-x and save
sudo systemctl disable hciuart.service
sudo systemctl disable bluealsa.service
sudo systemctl disable bluetooth.service
#some of these may not be found, it is a catch all set of commands for pi3/4/0
sudo reboot
```

```
3 wires are needed, TX, RX and ground for serial to work
RX on board goes to gpio14 and TX goes to gpio15, and use any ground
Some boards can power the pi over the 5v pin as well, but fair warning, make sure your board can provide enough power and do not use standard
dupont jumpers, they usually do not have the AWG to handle the current
```

```
You now need to build your firmware on the correct comms port
use BTT repo to find which pins on which UART to use, https://github.com/bigtreetech?tab=repositories
You will need to look at the pin PDF in the appropriate repo under Hardware/ to find the location of the RX/TX pins and then the SCH schematic
PDF to find out the pins. Sometimes it will tell you the UARTx number in the pin PDF, so you can use that to match what make menuconfig shows
as the options for UART
```

```
all that remains is to change the serial: in [mcu] in printer.cfg to use the serial
```

```
[mcu]
serial: /dev/serial0
baud: 250000
restart_method: command
```

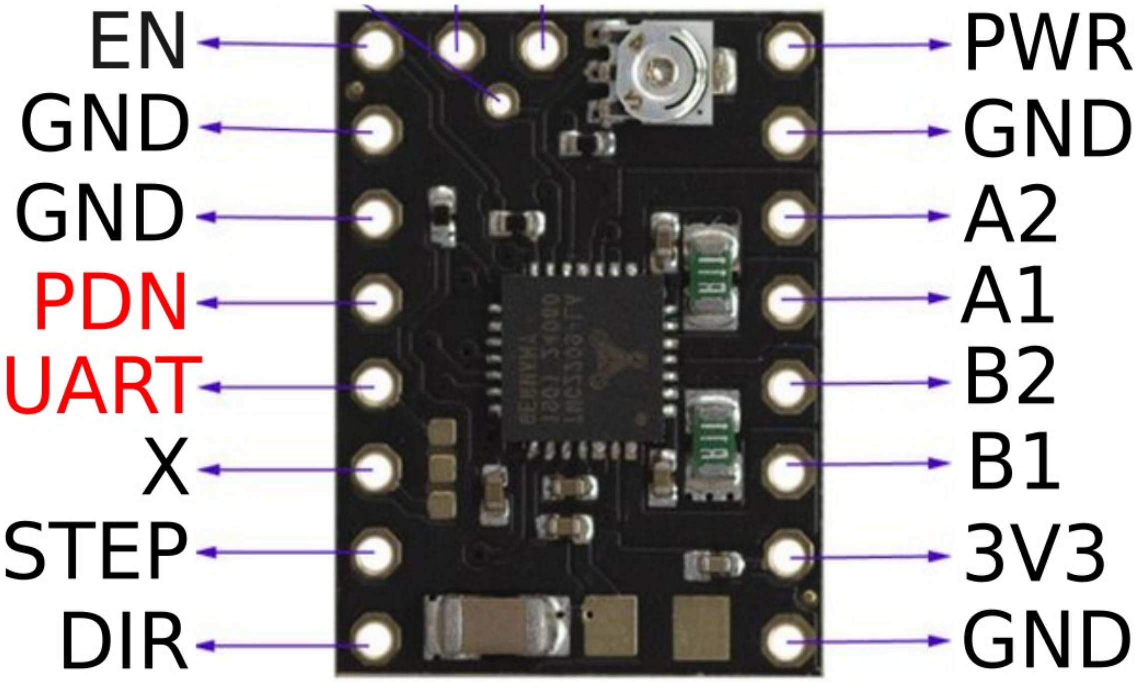
Usage Notes

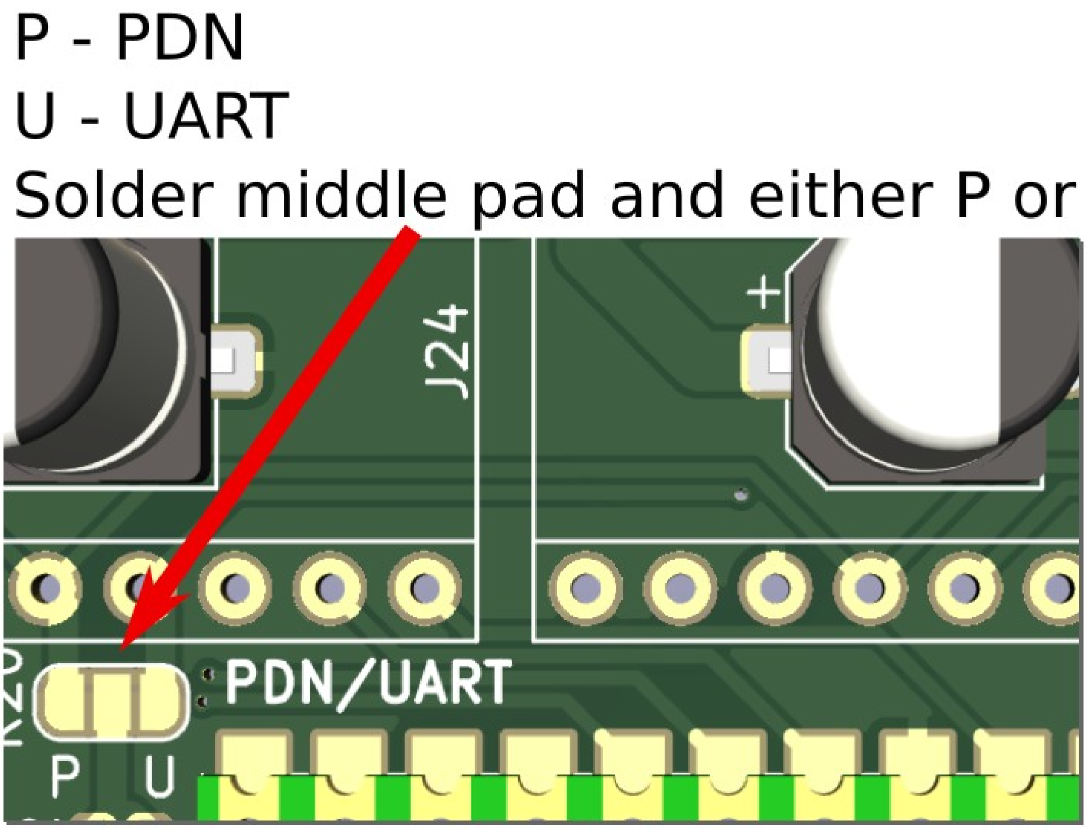
LEDs

5 LEDs in the corner are for status:
Extruder active
Bed active
Fan 0 channel active
Fan 1 channel active
5V power OK

Stepper Drivers

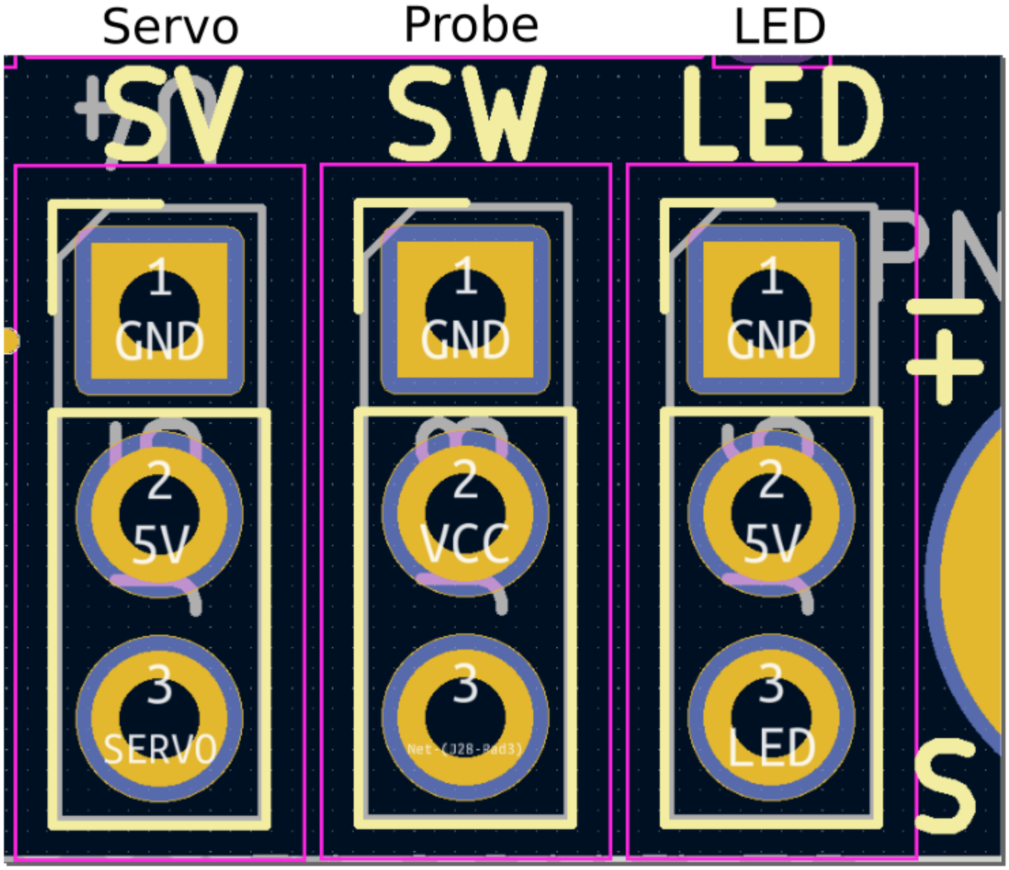
Stepper drivers can be configured to have UART communication through the PDN pin, or the UART pin. Refer to your stepper driver’s pin-out for more information. But this means that you don’t have to do any micro soldering of PDN/UART selection pads anymore!




P - PDN
U - UART
Solder middle pad and either P or U

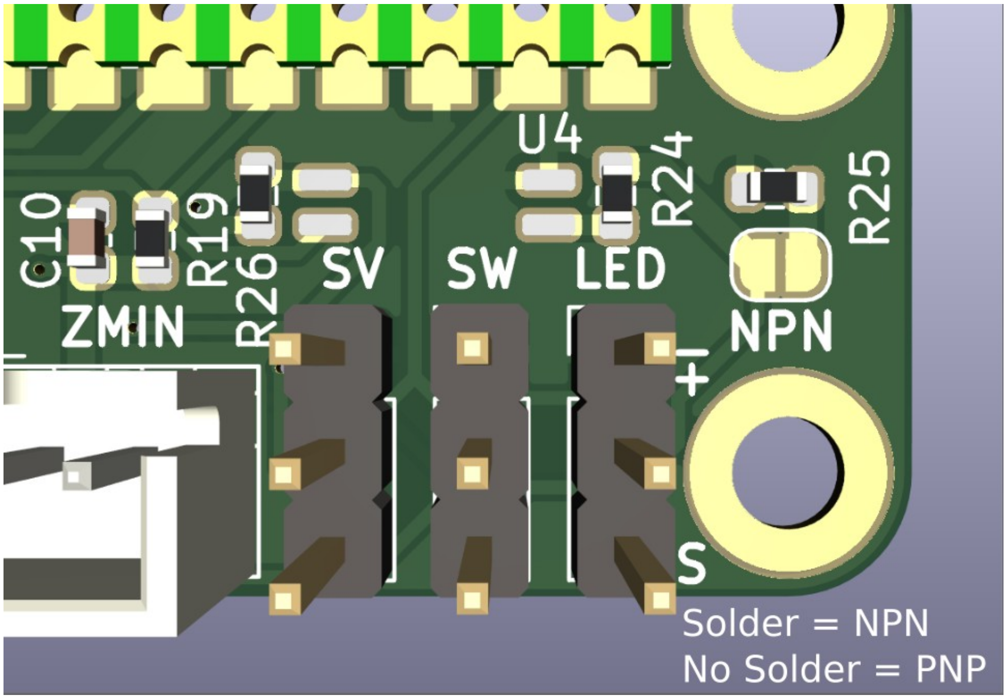
BLTouch

BlTouch probes should use Z-min endstop and Servo pin (pinout labeled in silkscreen for Servo connector) for probing. An example is in the printer.cfg file.

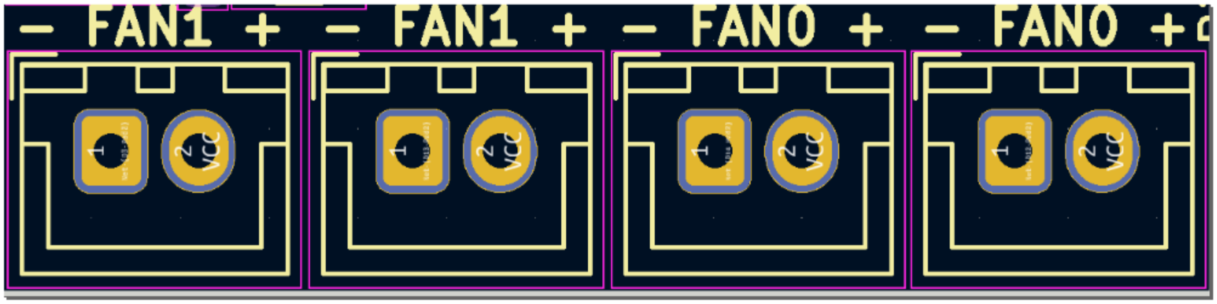


PL08 Probe

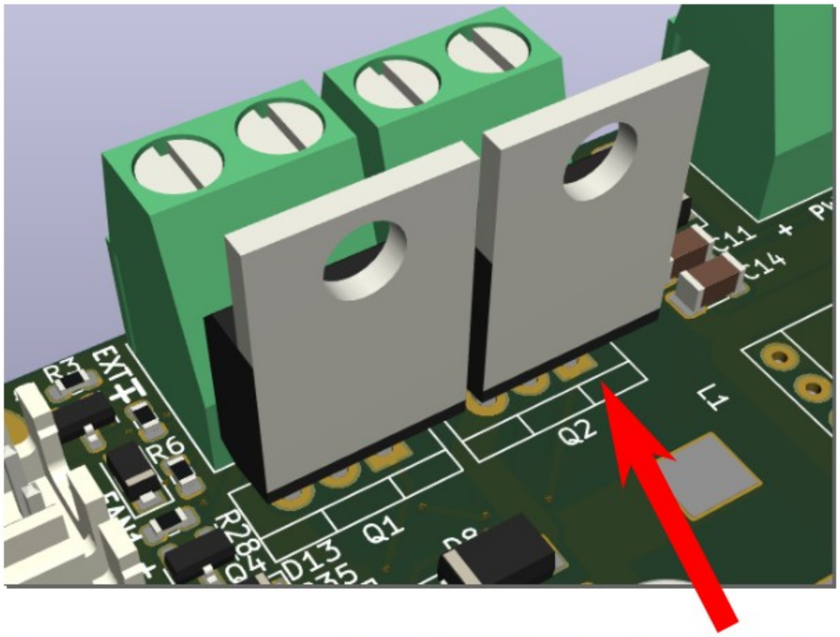
Simply plug in your PL08 probe to the SW header. Pinout is consistent with Servo and LED header, in silkscreen. Voltage is supplied directly from Power input. Optocoupler used, so no need for diode soldering “weirdness”. Solder pad if using NPN switch (PL082N).



Fans
There are two fan channels. Optimally, one would be used for cooling the hotend, while the other for print cooling. There are 2 outputs per channel, so you can mix and match fans to get things working that way you want them to. Please be aware that some machines may have the fan connector backwards. In this case, simply switch around the wires in the fan's connector and you should be good to go.


Pay attention to fan polarity!

Heaters
There should be silkscreen indicators for polarity. MOSFETs will not usually get very hot, even during heating. As always, ensure proper airflow while printing.



Ensure proper MOSFET orientation

If you have any further question or comments, please reach out to me at vitapartz0@gmail.com
Thank you, and Happy Printing!
Lawrence
VitaPartz