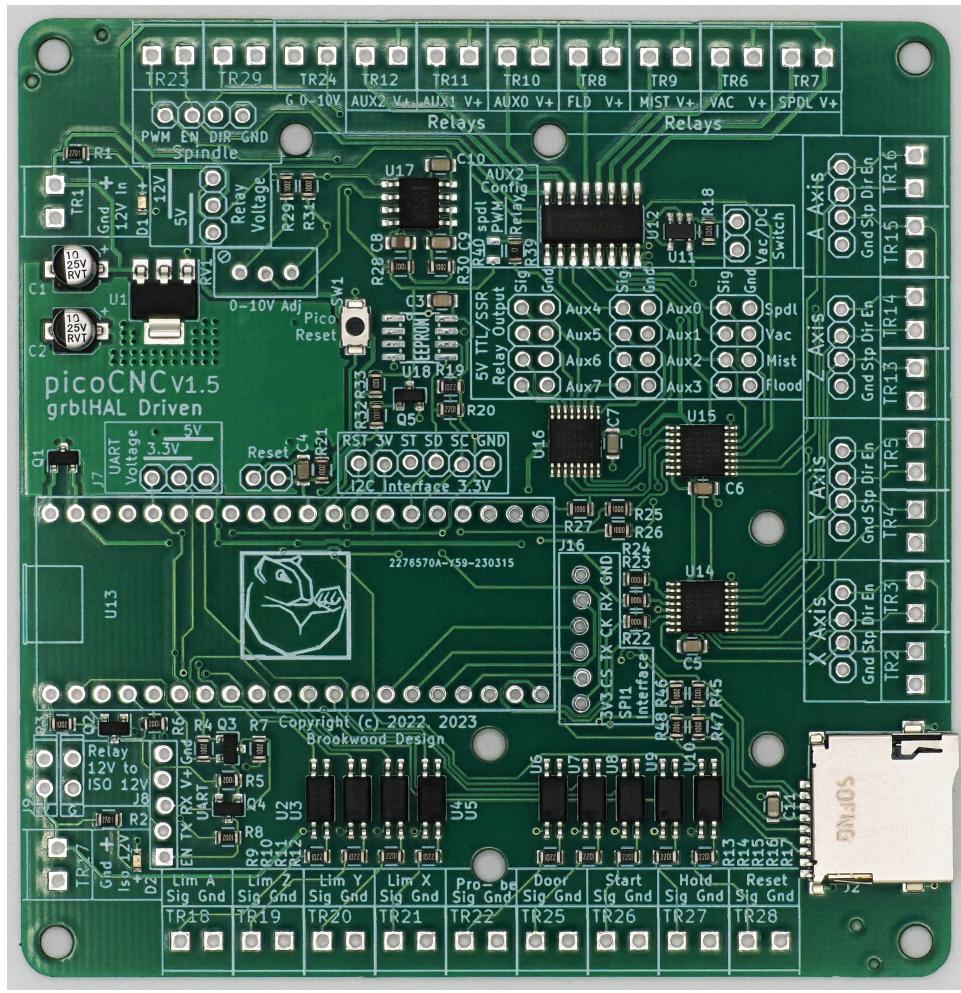


grblHAL PicoCNC Motion Controller Breakout Board User Manual

Document Version 1.02

Applicable PCB Revisions: 1.5, 1.51, 1.52



The PicoCNC PCB is a CNC controller board designed to run grblHAL.

Features:

- Compatible with Raspberry PI Pico and Pico W.
- 4 Axis control outputs.
 - 5V compatible.
 - Screw Terminal and Pin Header.
 - Independent enables for each axis.
- Spindle Control.
 - PMM (5V).
 - Direction (5V).
 - 0-10V speed control.
 - 12V PWM supported (open collector output).
- Limit Inputs for all 4 axes.
 - Opto-isolated
 - Support switches or powered proximity sensors
 - 12V for better EMI immunity
 - Separate power input for full isolation
- Opto-isolated inputs standard Grbl controls (Cycle Start, Feed Hold, Estop and Safety Door).
- Opto-isolated Probe input.
- Relay Support
 - Directly drive up to 7 relay coils via open collector outputs.
 - Relay Coil voltage selectable between +5V or +12V.
 - Drive up to 11 SSR or Logic Level relays.
- Dust Extraction support.
 - Activated by spindle enable.
 - Directly drive relay coil, powered relay board or SSR
 - Separate control of Dust Extraction via pin header/external switch.
- Screw Terminals for reliable connections available for most I/O.
- Communication
 - USB, via Pico, Pico W
 - WiFi, via Pico W
- LED Indicators
 - 12V Power (both inputs)
 - 5V Power
 - Spindle On
- Requires 12V supply.
- On-board 5V regulator.
- Flexible 12V power for isolated inputs. May be powered separately for isolation or off main board 12V.
- microSD Card socket
- I2C header with daughter card mounting holes.
- SPI header with daughter card mounting holes.
- UART header with daughter card mounting holes. Supports 5V or 3.3V.
- EEPROM footprint. Supports EEPROM or FRAM.

Stepper Driver outputs.

There are 4 sets of standard Grbl stepper driver outputs: X, Y, Z and B axes. Each has 4 pins: step, direction, enable and Gnd. Each are 5V compatible and capable of driving opto-isolated stepper driver inputs. Each axis has a set of screw terminals as well as pin headers (0.1"/2.54 mm spacing).

Limit Pins.

There are 4 limit inputs: X, Y, Z and A. All are opto-isolated and available via screw terminals. Powered by 12V and compatible with numerous limit sensors and switches. Fully compatible with mechanical switches or 12V powered proximity switches. When powered via an isolated 12VDC supply, they are fully isolated from the main processor and associated logic.

Control Input Pins.

Standard Grbl control input pins are supported via screw terminals: Feed/Hold, Cycle/Start, Halt/Stop and Safety Door open. These are opto-isolated. When powered via an isolated 12VDC supply, they are fully isolated from the main processor and associated logic.

Probe Input

Probe input is opto-isolated and available via screw terminals. When powered via a separate 12VDC supply, it is fully isolated from the main processor and associated logic.

Spindle Control.

Outputs for Spindle Enable, Spindle Direction and PWM signals available via screw terminal and pin header. These are 5V TTL logic compatible. There are relay outputs for the spindle (direct relay coil and 5V TTL). See Relay section for more details. Standard 0-10V output for spindle control is provided via a screw terminal. One auxiliary output can be set up to support 12V PWM output, open collector.

Relay outputs.

Standard Grbl Flood and Mist relay coil outputs are supported via screw terminals. Spindle relay output via a screw terminal is supported and controlled by Spindle Enable. In addition, a separate screw terminal relay output for Dust Collection is activated by either the Spindle Enable output or a separate pin header.

3 auxiliary relay coil outputs, Aux0 - Aux2 are supported via screw terminals and used via M62-M65 grblHAL pin control.

All relay coil outputs are configured as open collector drivers and capable of directly driving relay coils. Each relay driver is capable of up to 100 mA. All 7 are limited to 500 mA in total. Above 250 mA, forced air cooling is recommended.

All relay coil drivers have open collector outputs.

Each relay also has a 5V TTL output that can be used to drive solid state relays, separate relay boards or other low voltage activated devices. In addition, there are 5 TTL/SSR outputs, Aux3 - Aux7.

Relay coil voltage is selectable between 5 and 12V/24V via the pinheader labeled Relay Voltage in the upper left corner of the PCB or, alternatively, via a solder jumper on the bottom the PCB.

One auxillary output (Aux 2) may be connected to the spindle PWM signal to drive 12V devices. It can be used as an open collector driver or with an optional pull up resistor.

All relay coil outputs have an optional pull up resistor pad.

Digital Outputs

Name	Open Collector Screw Terminal	TTL/SSR Pin Header*	Notes
Spindle	✓	✓	TTL* signal also available via Screw Terminal
Dust Extraction/Vacuum	✓	✗	Triggered via Spindle or Pin Header
Mist	✓	✓	
Flood	✓	✓	
Aux 0	✓	✓	
Aux 1	✓	✓	
Aux 2	✓	✓	Assignable as a 5V or 12V spindle PWM output
Aux 3	✗	✓	
Aux 4	✗	✓	
Aux 5	✗	✓	
Aux 6	✗	✓	
Aux 7	✗	✓	

* A note on TTL. Technically, this should be referred to as “Logic Level” but we use TTL as a convenient, if imprecise, shorthand.

I2C Header

An I2C header is provided. 3.3V only, there is no translation to/from 5V and the inputs are not 5V tolerant. A strobe line is provided on the header. In addition, a line for an E-Stop signal is routed directly to the Grbl Halt input. Currently, only a limited number of devices are supported. Check the grblHAL github site for more details.

SPI Header

An SPI header is provided. 3.3V only, there is no translation to/from 5V and the inputs are not 5V tolerant. The header pins are also connected to a microSD socket. It is not recommended that the header is used when a microSD card is inserted in the socket.

UART Header

A UART header is provided. RX, TX and an enable pin are available. Supports 3.3V/5V translation, the interface voltage is selected via the UART Voltage pin header.

External Connections

12V, Input, main power.
Supply for relays, 0-10V output amplifier and 5V regulator.

12V, Input, isolated input.
Supply for opto-isolated input switches.

Stepper motor, Output
5V Logic Signals.

Pin	Signal
G	Ground
Stp	Step Pulse. Positive.
Dir	Direction
En	Enable

Limit Switches, Input
One input for each axis. Opto-isolated, 12V.

Halt, Door, Cycle/Start and Feed/Hold, Input
Opto-isolated, 12V.

Probe
Opto-isolated, 12V.

Spindle Group, Output.
PWM, Direction, Enable. 5V compatible.
0-10V.

Relay Coil Group, Output
These are open collector sinks. They run at the selected relay voltage.
Flood, Mist, Spindle, Vac/Dust Collector, Aux0, Aux1, Aux2

Relay TTL Group, Output.
These are 5V, TTL compatible outputs suitable for SSR, powered relay modules or any 5V compatible logic.
Spindle, Flood, Mist, Vac/Dust Collector, Aux0, Aux1, Aux2, Aux3, Aux4, Aux5, Aux6, Aux7.
Note: Enable in the Spindle Group is capable of driving an SSR, powered relay modules or 5V compatible logic.

Vac/DC trigger.
External Switch to activate a vacuum or dust collector.

I2C Header, Input/Output.

Pin	Signal
G	Ground
SC	I2C SCL
SD	I2C SDA
St	Strobe, used for signaling outside of I2C
3V	3.3VDC, sourced by Pico
Hlt	Halt Signal. Connected to the Halt input Pico pin.

SPI Header Input/Output

Pin	Signal
Gnd	Ground
RX	SPI Device output, MISO
CK	SPI Clock
TX	SPI Device input, MOSI
CS	Component (Device) Select *
3V3	3.3VDC, sourced by Pico

* The SPI header is shared with the microSD Card reader.
Concurrent use of both SPI and microSD reader is not supported.

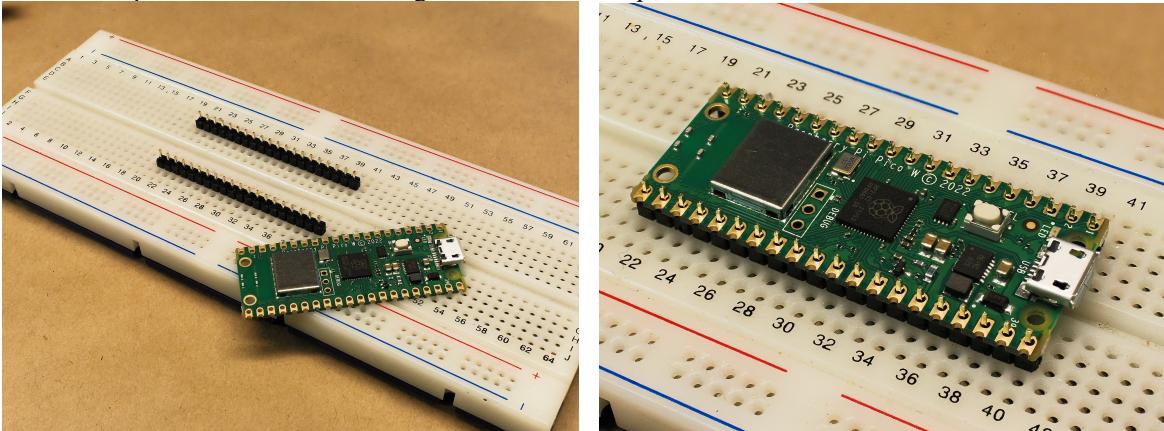
UART Header Input/Output

Pin	Signal
Gnd	Ground
V+	3.3V or 5V, selected via UART Voltage header
RX	Receive, 3.3V or 5V
TX	Transmit, 3.3V or 5V
EN	Enable, 3.3V or 5V can be used for signaling or switching half duplex logic

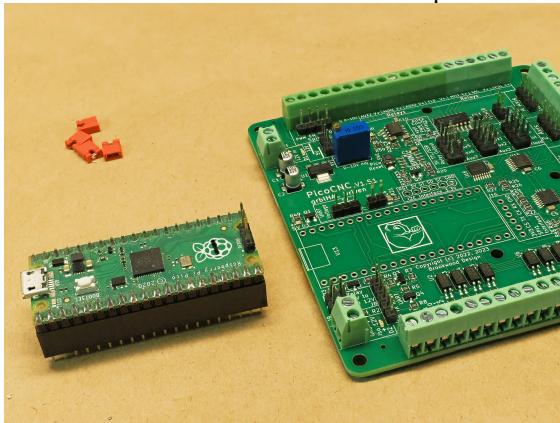
PCB Assembly

The PicoCNC board is provided with all standard surface mount components soldered in place. You are responsible for installing the through-hole components, either from the option assembly kit or sourced yourself (See BOM section). Assemble the PCB in the following order:

1. Install 1x2 and 1x3 pin headers: Vac/DC Trigger, 5/12V selection, 12V Iso (V1.52 and later) and UART Voltage. Carefully snap each one off the longer pin-header strip. The headers should snap into place but if they are loose, use tape (painter's tape works well) to hold them in place. Solder 1 pin and then verify the alignment. Each header should be perpendicular to the board. If it is not, reheat the soldered pin and gently press it into place (watch the heat on the short headers!). Be sure to be gentle with heating headers as the plastic spacer softens quickly.
2. Install longer 1x headers : Spindle (1x4), I2C (1x6), SPI (1x6) and 4 Axis Headers (1x4). Use same technique as in 1).
3. Install 2X headers: TTL Relay 3X 2x4 headers. Isolated 12V header 2x2. Use same technique as in 1).
4. If you bought a Pico or Pico W without headers, install pin headers. A solderless breadboard makes a great way to stabilize the pin headers while soldering the 0.1" header strips.



5. Install the 3 pin potentiometer. You can bend the leads to ensure it stays in place. But, also follow the the above approach of soldering 1 pin first, verifying alignment and then finishing.
6. Install the Pico. Place the sockets on the pin headers you just soldered onto the Pico.



Now, gently insert the socketed Pico assembly into the indicated holes on the breakout board. Verify that all socket pins are correctly in place. Secure the entire assembly with tape on the component side of the board and then solder one pin on each of the two sockets and verify alignment. Once you are sure that the alignment is correct, solder the other pins, leaving the pins you initially soldered to last.

7. Mount and solder screw terminals. Assemble a complete row before soldering. The terminals lock together but are sometimes hard to get completely aligned. If that is the case, place them upside down (with the pins pointing up) on a hard surface and press each terminal in place with a screwdriver or similar. When you are satisfied that the tops are

flush, insert the assembly into the holes on the board. The fit can be a bit tight so you may need to use a bit of force to get them full inserted. Solder one pin one each end. Make sure the row is properly aligned; reheat and adjust if not. It is sometimes helpful to solder the pin of middle a middle terminal while moving the row into proper alignment. Then solder remaining terminals while ensuring that the row is straight. Be sure to install the 2 power terminals.

Relay Voltage

Relay Coil Voltage source is selectable between 5V and 12V. Use a “suitcase jumper” as indicated on the PCB to select the desired voltage.

The user is responsible for ensuring that all relays are rated for the voltage they select.

Note: it is recommended to not use the stepper motor supply for the relay supply. It could be a source of EMI and cause erratic operation of the system.

Dust Collector/Vacuum Control

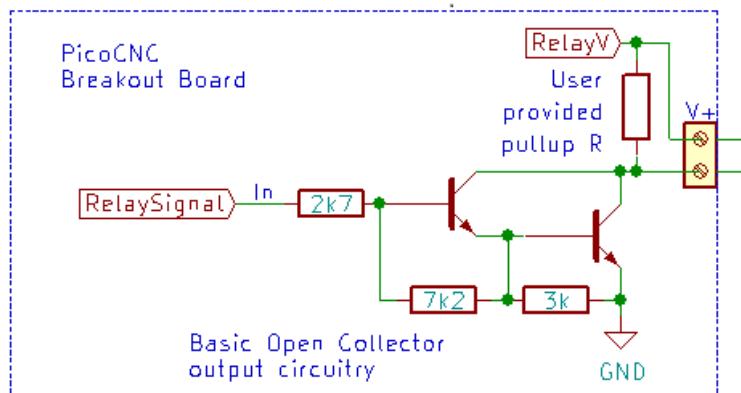
An additional driver is provided for a dust collector relay. It is activated whenever the spindle is turned on (M3). In addition, this relay output can be independently activated by a switch that connects to the pin header labeled Vac/DC Trigger. This makes it easy to do clean up with the same Vacuum or Dust Collector used for dust extraction while running your CNC machine.

Board Level Configuration Options

There are several options to modify the behavior of the PicoCNC board.

Open Collector Output Pull Up Resistor.

The relay outputs use an open collector to drive relay coils. Pads for pull up resistors are available if needed. They are located on the bottom of the PCB and are labeled for their associated output. Use an 805 SMD resistor. 10K ohms is a common pullup value but it is up to the user to determine the appropriate resistor value. Refer to the diagram to understand how the output works. Note that the attached device needs to share a common ground with the PicoCNC board.



12V Spindle PWM Output

Spindle PWM can be directed to Aux2 output. This is controlled by R39 and R40. The default is to have Aux2 controlled by M62-M65 G Codes (technically, M Codes...). To have the Aux2 output controlled by the spindle PWM signal, you need to move the 0 ohm resistor on the R39 pad to the R40 pad. A small piece wire can be used if the 0R resistor is lost when desoldering. Note that a maximum PWM frequency of 50K is supported. Maximum current draw is 250 mA. Aux2 can not be used for M62-M65 control when configured for PWM.



I2C Pullup Resistors

If you are using an I2C device, you may need pullup resistors on the communication lines. There are pads on the bottom of the board to allow you to use 805 SMD resistors.

Firmware.

The [grblHAL Web Builder](#) site is used to create the proper firmware for your PicoCNC board.

Select PR2040 (Pi Pico and Pi Pico W) in the Driver box and PicoCNC in the Board box.

In the General panel, select the Number of axes and the appropriate axis configurations for your CNC machine.

In the Plugins panel, select Enable or Ymodem in the SD card option. Ymodem will speed up transmission of large files.

If you have a Pico W, in the Network/WebUI panel select Networking WiFi and check Telnet, WebSocket and FTP.

Then press the Generate and Download button. Save the firmware file for use in the next section. Also, press the Save board button so that you will have the configuration if you need to later make changes.

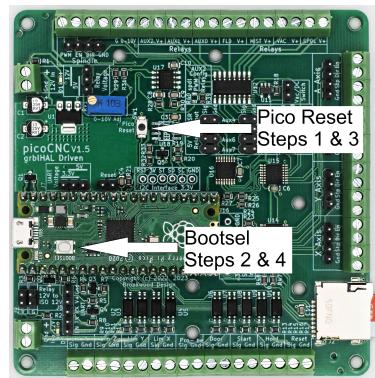
Installing the firmware is very easy! Once you have a firmware file, connect a USB cable from your computer to the Pico installed on your PicoCNC board. Then:

1. Press and hold the Pico reset button on the PicoCNC board.
2. Press and hold the Bootsel button on the Pico board.
3. Release the Pico reset button on the PicoCNC board.
4. Release the Bootsel button on the Pico board.

Windows

The first time you do this, a USB configuration notification will come up. For options, select “Open File Explorer Window”. After that, all subsequent Reset/Bootsel operations will cause a file explorer window to open. Drag and drop your firmware file onto the window. You will see a progress indicator as the firmware file is uploaded to the Pico.

When done, you can start your GCode sender application and connect to the com port associated with the Pico. Use the Device Manager to determine which one it is if you are having difficulty finding it.



Mac OS

A drive called RPI-RP2 will appear on your desktop. Double click on RPI-RP2 and a finder window will open. Drag and drop the firmware document onto the window. When done, you can start your GCode sender and connect to the Pico.

If you are interested in building grblHAL from source, you can download the source code at <https://github.com/grblHAL>. Note that there are multiple project and libraries that you will need to download. A complete discussion of installing and building the source code is beyond the scope of this document. Start with the core wiki to understand how to build grblHAL – <https://github.com/grblHAL/core/wiki/Compiling-grblHAL>.

Running grblHAL on your machine

Once you have finished board assembly and installed the firmware, your grblHAL PicoCNC based Motion Controller is ready to be tested in your CNC machine. Attach it to a PC via a USB cable and run a compatible GCode sender application on the PC. We recommend this one: <https://github.com/terjeio/ioSender/releases/tag/2.0.42>. Check for the most recent release. Next, you will need to set up Grbl to reflect your hardware. We will use ioSender for our examples. Actual Grbl setting values are also shown. Follow these steps:

1. Hook up your control buttons: Feed Hold, Cycle Start and EStop.
2. Open the **Settings: Grbl** panel. You will see a list of parameters that control grblHAL.
3. In the Spindle section, \$16, Invert spindle signals, check the PWM box. \$16=4.
4. In the Stepper section, \$4, Invert stepper enable pins (s), Check all boxes. \$4=7 for a 3 Axis machine. \$4=15 for a 4 Axis machine.
5. In the Control signals section, \$14, Invert control pins. Check Feed Hold, Cycle Start, EStop and Safety Door (if present). \$14 = 15.
6. Press the save button and return to the main tab (**Grbl** panel, you should see the DRO section).
7. Press the Red Reset button if an ALARM state is showing.

8. Open the **Jog** panel and click on any of the jog direction buttons. The DRO should show movement. If you have drivers and motors hooked up, they should show movement.
9. If you have any control buttons attached, test them. It should show activity. Note the associated “LED” in ioSender when you press a button. If nothing happens, you should verify your connections.
10. If everything works to your satisfaction, you should now proceed to full testing of your machine.

Note, if you have not connected 12V to the Iso 12V terminal (or installed jumpers on the 2x2 header next to the terminal), the input signals will not be active. You will also not see a lit LED near the Iso 12V screw terminal. This means that you won't see any of those signals in ioSender.

There is a lot more information in the grblHAL wiki at <https://github.com/grblHAL/core/wiki>. In particular, we recommend you check the First Run Grbl Settings section.

Some Initial Grbl Settings

Grbl settings are dependent on your hardware so you might have to experiment a bit. Here are some key settings that will help you get going faster.

\$ID	Name/meaning	Suggested Value	Comments
0	Step Pulse time in uS	5	Default is 10 but 3 or 5 should work ok. Using the default may cause problems for step rates above 80 kHz.
4	Step Enable bit mask, one per axis	7 (3 motors) or 15 (4 motors)	A 1 bit means invert. A 3 axis machine would have 7 as the value, a 4 axis machine would have 15. Bit 0 is X, 1 is Y, 2 is Z and A is 3.
14	Control pin inversion bit mask, one per pin.	15	A 1 bit means invert. Reset is bit 0, Feed Hold – 1, Cycle Start – 2, Safety Door – 3. This assumes you are using NO switches for input.
16	PWM Inversion	4	Inversion required for PWM and 0-10V. Note Spindle Indicator LED will be lit if not inverted.
30	Max spindle speed in RPM	1000 or VFD max.	Default is 1000. If you have a VFD, you will need to set this to the spindle max. In G-Code, the Sxxx parameter controls speed and to get full speed, you will need to have xxx match this parameter (S1000, in the case of the default).
100, 101, 102, 103	Steps/mm. One for each Axis. 100 – X, 101 – Y, and so on.	varies	Set these based on your hardware – lead screw/belt ratios, microstepping. The Grbl defaults are likely wrong for your machine but you will still get movement. On a per axis basis, determine the number of steps (including microsteps) it will take to make one full rotation of the lead screw. Determine how many mm that axis will move with one full rotation of the lead screw, pinion gear or belt driver pulley. Divide the first number by the second to get the value you will use.
110, 111, 112, 113	Max Rate	2500 mm/Min	This is a reasonable starting point for a hobby machine. Higher performance machines can easily support 5000 or higher.
120, 121, 122, 123	Acceleration in mm/Sec ²	500 mm/Sec ²	This is a reasonable starting point for a hobby machine. High performance stepper motors and servos can support significantly higher acceleration.

You should spend some time studying all the Grbl settings. We suggest you start conservatively and tune your machine to higher performance one parameter at a time.

Compatible G-Code Senders

Most senders will work with grblHAL. If your preferred sender is having problems, it may need to use a version of grblHAL built with compatibility settings. Use the grblHAL Web Builder interface described in the Firmware section above. Compatibility Level is set in the Advanced features panel. [note: this is increasingly becoming unnecessary.]

Setting up WiFi with a Pico W.

Follow all the above steps when setting up to use a Pico W. When using the the web builder to create the firmware, be sure to select WiFi on the Network/WebUI panel, Networking box. Also check Telnet, WebSocket and FTP. Note that your screen may differ slightly from the one pictured here.

The screenshot shows the grbl HAL Web Builder v0.9d software interface. At the top, it says "grbl HAL Web Builder v0.9d - work in progress". Below that, there's a note about generating and downloading firmware. The main menu bar includes General, Plugins, Network/WebUI (which is selected), Advanced features, and 3rd party plugins. Under the Network/WebUI tab, the Networking section is open, showing options for WebUI (disabled) and WiFi. Under WiFi, Telnet server, WebSocket server, and FTP server are checked, while others like HTTP server, WebDAV protocol, mDNS protocol, and SSDP protocol are unchecked. Buttons at the bottom include "Generate and download firmware", "Save board", and "Load board". A note for first-time users points to a Wiki page.

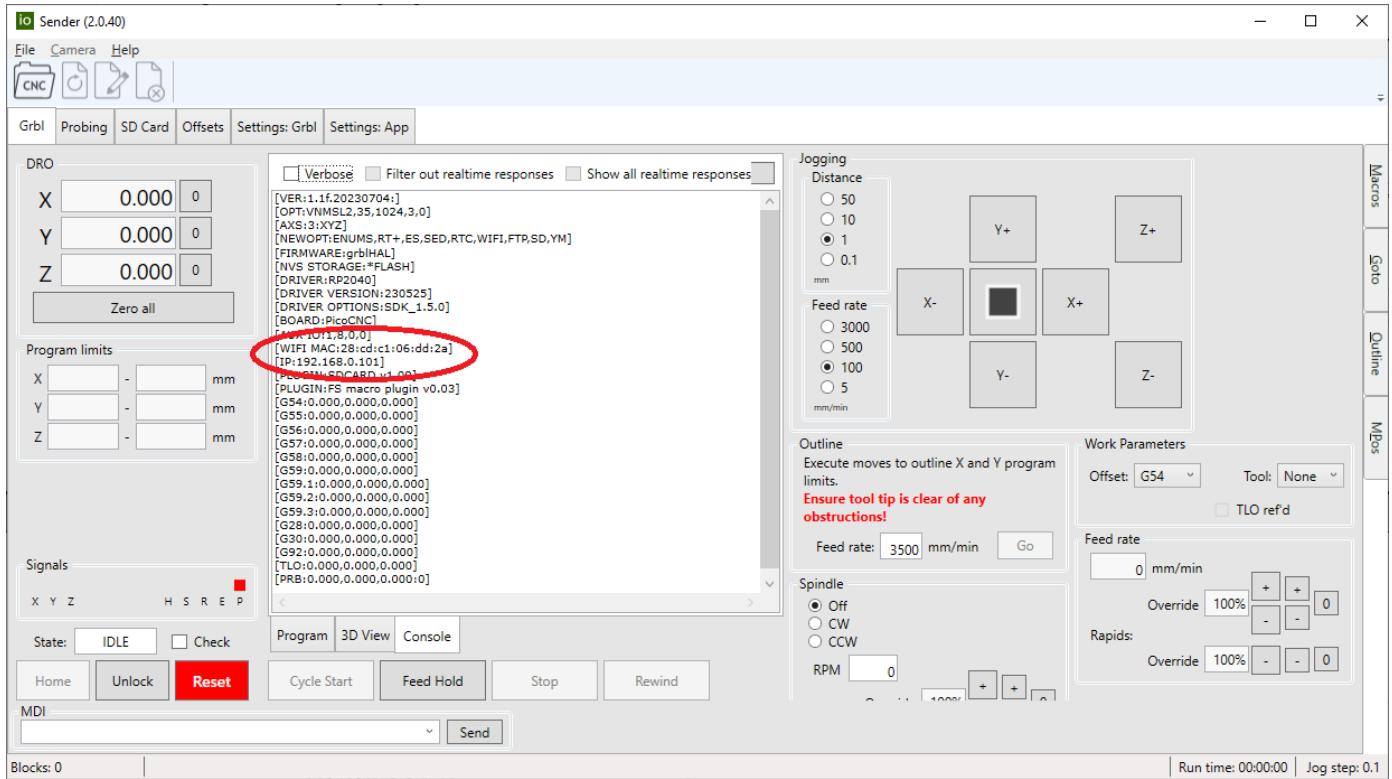
Generate and download the firmware file. Install it as directed above.

Once you are able to connect to the PicoCNC board via USB, configure WiFi by the following steps.

1. In ioSender, open the Settings: Grbl panel.
2. In the Networking section, click on the Network Services entry (\$70). Make sure that Telnet, Winsocket and FTP boxes are checked.
3. Open the WiFi section and fill out the WiFi network name/SSID (\$74) and password (\$75) for your WiFi network.
4. Make sure that the WiFi Mode (\$73) is set to Station.
5. Press the Save button, close ioSender and press and release the Pico Reset button on the PicoCNC board.
6. Close and Restart ioSender. Click on the Console panel. Look for the following lines:

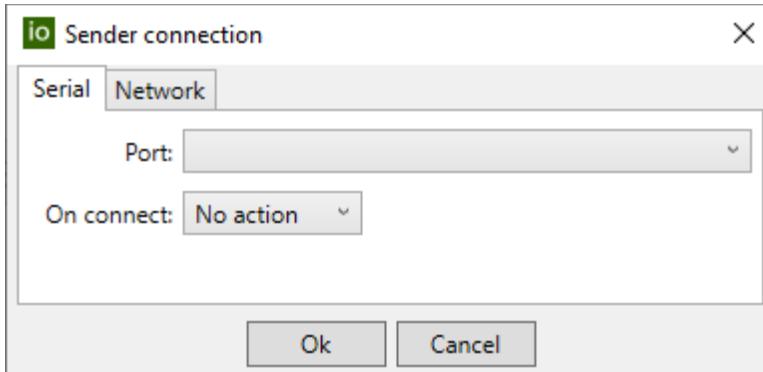
```
[WIFI MAC:xx:xx:xx:xx:xx:  
[IP:xxx.xxx.xxx.xxx]
```

Where x is a digit or letter. If IP is 0.0.0.0, then you are not connected, otherwise the PicoCNC is now connected to your WiFi network. Write down the IP address numbers, you will use them shortly.

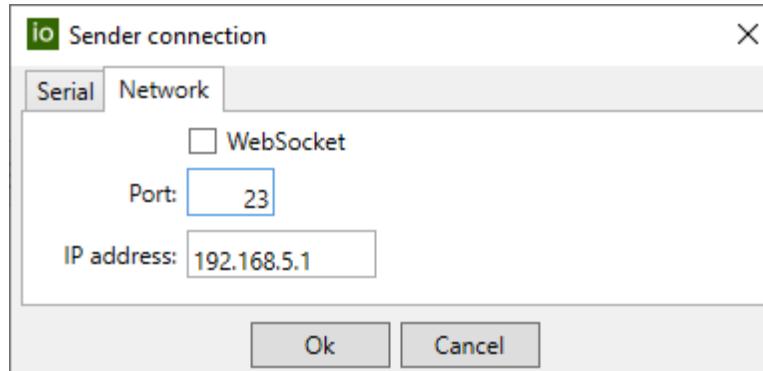


To set up communication via WiFi (rather than the current USB connection),

1. Close ioSender
2. Press and hold the Pico Reset button on the PicoCNC board (alternatively, disconnect the USB cable from the Pico)
3. Restart ioSender. It will put up a dialog asking you to configure the connection.

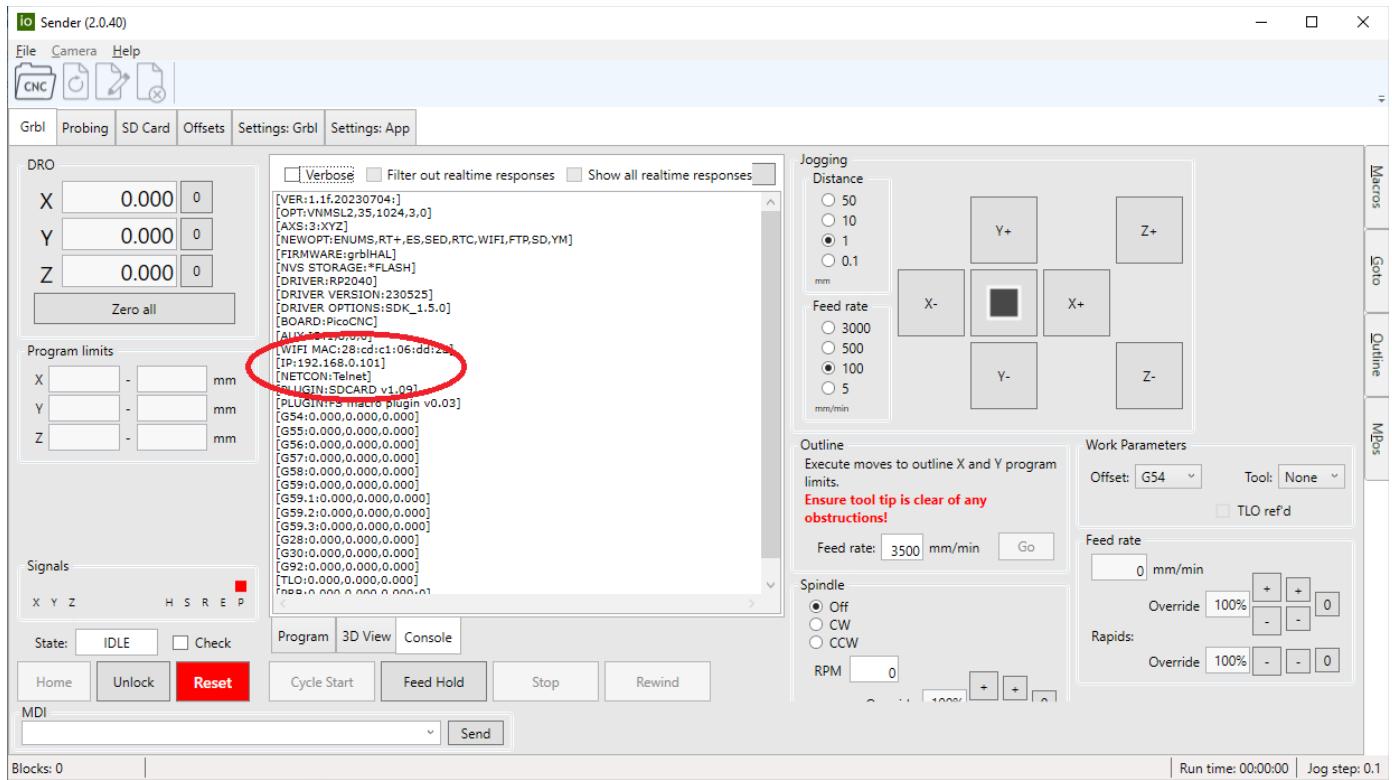


4. Release the Pico Reset button. (alternatively, plug the USB cable back into the Pico)
5. Click on the Network tab.



6. Enter the IP address numbers you wrote down in place of 192.168.5.1. Click on Ok and you should be rewarded with ioSender running and connected via WiFi to the PicoCNC.

To verify you are connected via WiFi, open the console panel and look for a line that reads [NETCON:Telnet].



0-10V Spindle output setup

The 0-10V circuit works by using an opamp to filter a 5V PWM signal and amplify it to 10V. As such it needs 12V or higher input which the user must provide via the 12V screw terminals. A second order filter with an aggressive low pass cut off frequency is used to provide a stable signal for the spindle controller.

Because source voltage level can vary, the amplifier must be adjusted. You can do this by turning on your spindle in GCode (via M3Sxxxx) to full speed and adjusting RV1 (the blue box with the screw) until the voltage on the 0-10V terminal reads 10V. Here is a step-by-step method you can use:

- Disconnect your spindle or VFD.
- In the **Settings: Grbl** tab, set \$30, **Maximum Spindle Speed**, to 1000.
- Send the following command to grblHAL: **M3 S1000**. This turns on the spindle and causes the speed to be set to 100%.
- With a digital multimeter, measure the voltage between the 0-10V output and G terminals.
- With a screwdriver, adjust the RV1, labeled “0-10V Adjust”, until the voltage is as close to 10V as you can get. Anything in the range 9.990V to 10.010V is acceptable.
- Send the following command to grblHAL: M5. This turns off the spindle and you should read 0V.
- Reconnect your spindle or VFD and test.
- Change the max spindle speed setting, \$30, to something more appropriate for your spindle.

You may want to repeat the above process with the spindle connected and running to fine tune the maximum speed. Most VFDs will show you the actual spindle speed.

Powering the PicoCNC.

The PicoCNC requires 12V input. It will draw up to approximately 500 mA though any directly driven relay coils or other devices will add to that.

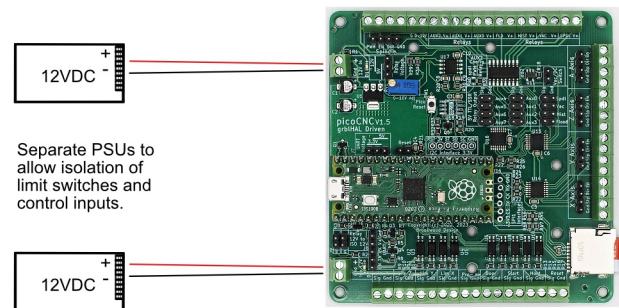
Warning

- Do not use a PSU that may go above 12V. Unregulated wall adapters (aka “wall warts”) can go quite high. These can damage the PicoCNC. Use a regulated PSU that outputs 12V even with no load.
- Do not reverse polarity. Connect + on the PSU to + on the PicoCNC board. Connect - on the PSU to - on the PicoCNC board.

There are 3 ways to power the PicoCNC board.

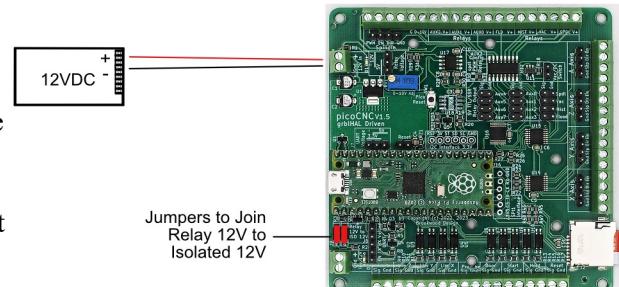
1) Dual PSUs, recommended.

This provides separate power for the relay and spindle sections and isolated power for the limit sensors/switches. The digital section – the Pico board itself and all the digital logic on the PicoCNC board – are powered by an on-board 5V regulator fed from the 12V relay input. 1 Amp PSUs are acceptable for both supplies. If you have other 12V items to power, use the relay section PSU and size accordingly to the additional load.



2) Single PSU.

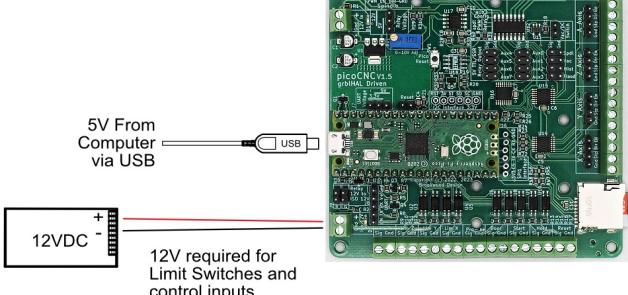
For budget conscious builders, there is the option to use a single PSU. The 2x2 pin header near the 12V Iso input can be jumpered to join the two 12V sections together and thus have one 12V PSU power the entire board. This does not provide true isolation for the limit, probe and control input section but may be acceptable, especially if good EMI practices are followed. The opto-isolators provide some EMI protection even without a fully isolated PSU. See “EMI Considerations” below.



WARNING. PCB V1.51 jumpers must be oriented vertically. PCB version V1.52 and later have 2 separate 1x2 headers that need to be jumped, rather than a 2x2 pin header.

3) Logic Powered from USB

The core digital section of the PicoCNC can be powered via USB power. This is acceptable if you do not power any relays or other devices. However, the Limit, Probe and Input section requires 12V power to function. If you do not supply 12V to the PicoCNC board, grblHAL may boot up in an error state. See the First Run Settings in the wiki for how to deal with that.



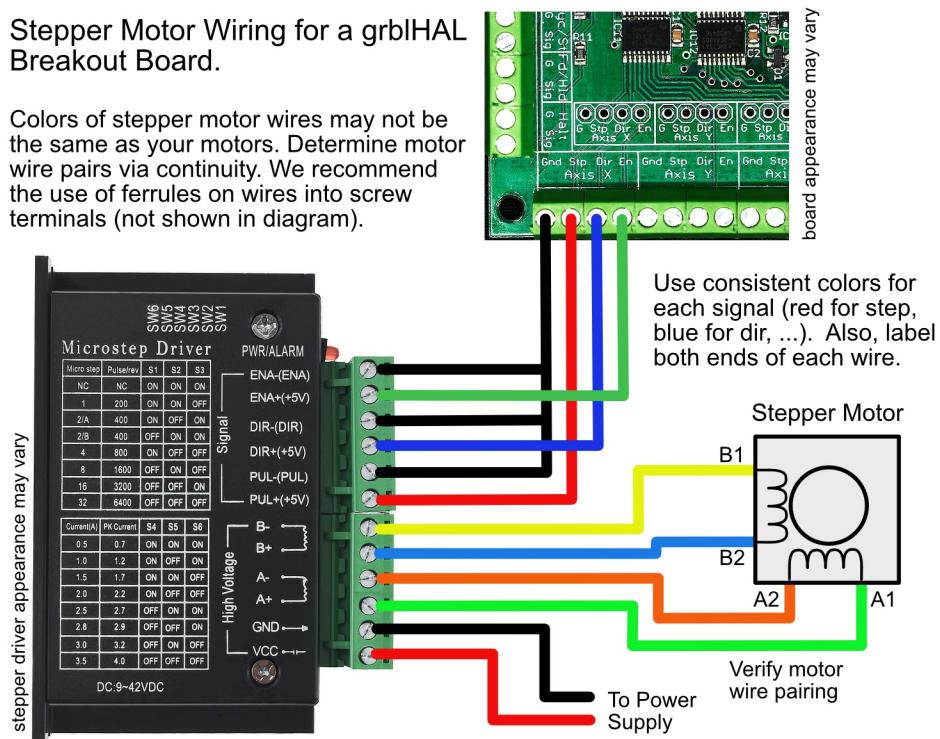
Connecting Stepper Drivers to the PicoCNC

Using stepper motors with the PicoCNC is fairly simple. Typically, a Stepper Motor Driver is used to turn the PicoCNC's Step and Direction signals into voltage and current that cause the desired step motions of the stepper motor. Servo drivers do a similar operation so, for simplicity, we will just talk about steppers here but the following applies to servos as well.

Most stepper drivers use opto-isolators for input so you will see a plus (+) and minus (-) terminal for each signal. In our case, we will connect the minus (-) inputs to ground on the PicoCNC board and the plus inputs (+) to the named signals (Step, Direction and Enable) on the PicoCNC board. Typically the Step terminals on drivers are labeled PUL (pulse), the Direction terminals are labeled DIR (direction) and the Enable terminals are labeled ENA (enable). Other stepper drivers may have variations on these names.

Stepper Motor Wiring for a grblHAL Breakout Board.

Colors of stepper motor wires may not be the same as your motors. Determine motor wire pairs via continuity. We recommend the use of ferrules on wires into screw terminals (not shown in diagram).



Using servos with the PicoCNC board.

The connection are similar to the above diagram. Servos also have an error output, often labeled ER or similar. This signal indicates when the servo has failed to operate as directed, usually due to lost steps. Typically, it is an open collector signal. You can connect it to the Halt input on the PicoCNC board in parallel to an Estop switch. This will cause your GCode program to stop when an error condition occurs. It is also possible to connect it in parallel with an NO switch on the associated axis limit input. You will need to configure your Grbl Settings to use hard limits and have correct inversion on the limit input.

Connecting to a spindle

There are several different ways you can connect the PicoCNC board to control a spindle:

- On/Off control via spindle enable (En) or Relay.
- PWM control of a speed controller or VFD.
- 0-10V control of a VFD.
- Modbus/RS485.

On/Off control

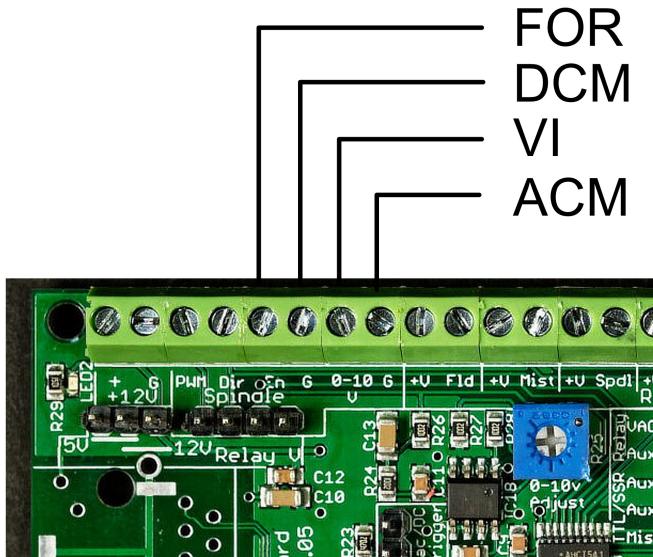
There are 3 places to control spindle On/Off: the En screw terminal in the spindle group (a 5V logic level output), the Spdl in the TTL/SSR pin header group (a 5V logic level output) or the SPDL screw terminal (an Open Collector output). Any of these can be used to control a relay to turn a spindle on or off. This is how you would use a router as the spindle.

PWM control

Some spindle controllers take 5V PWM control. This is common in DC motor controllers. You can use the PWM screw terminal output for this. If your DC motor controller requires a different output voltage, Aux2 can be configured to drive 12V PWM. See the 12V Spindle PWM section. If other than 5V or 12V are needed, an external voltage can be used in conjunction with Aux2 – see the General Purpose Open Collector section.

VFD Control via 0-10V output

VFDs may be controlled via 0-10V output. A typical connection is shown below. The VFD must be configured to take an external voltage for control and set to 0-10V. Some VFDs will also take PWM control. The actual configuration steps for a VFD are beyond the scope of this manual. Consult with your VFD's user manual.



Modbus/RS485

Not directly supported on the PicoCNC board but you may use a serial to Modbus adaptor with the UART header.

Using relay outputs with the PicoCNC board

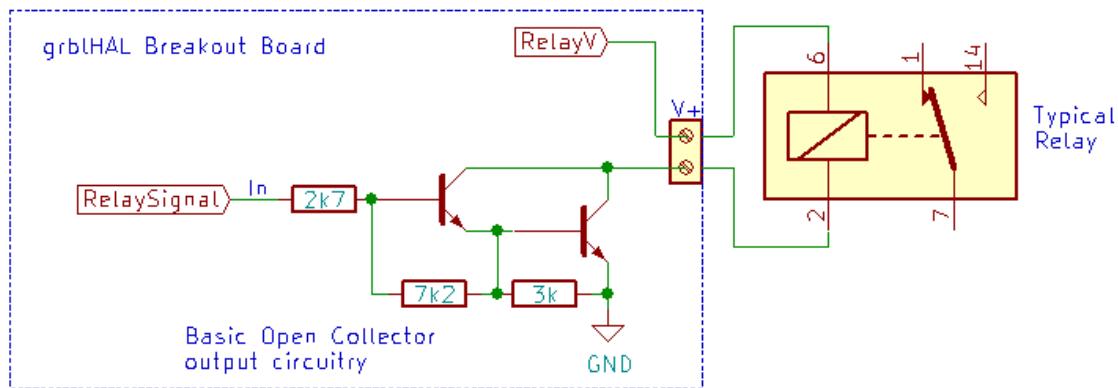
There are several ways to use the relay outputs of the PicoCNC board:

- To directly drive external 5V or 12V relay coils.
- Drive solid state relays (SSR).
- Drive powered relay boards (PRB) with 5V control inputs.
- Drive 5V logic (TTL) devices.
- As a general purpose open collector (OC) driver.

A note on using relays and other devices to switch 110/220 VAC: Please be aware of the risks associated with using relays to control high voltage devices like motors. If you are not familiar with high voltage, we recommend you consult a professional electrician for help in selecting the right devices and proper wiring techniques.

Directly driving external relay coils.

The relay screw terminal outputs of the PicoCNC board are used to drive relay coils. Automotive and DIN Rail mountable relays are commonly available. The relay coil is wired across the two terminals as shown below. Relay coil current up to 100 mA is supported. If only one relay is connected, up to 300 mA coil current is acceptable. The 12V relay section PSU supplies the current for relay coils. Add up total relay current draw plus 100 mA. Make sure that your PSU is capable of delivering that amount of current.



Driving solid state relays.

SSRs are driven from the TTL/SSR Relay section pin headers. The output voltage level is +5V which is compatible with most SSRs but verify that yours is as well. DIN rail mountable SSRs are readily available.

Driving powered relay boards.

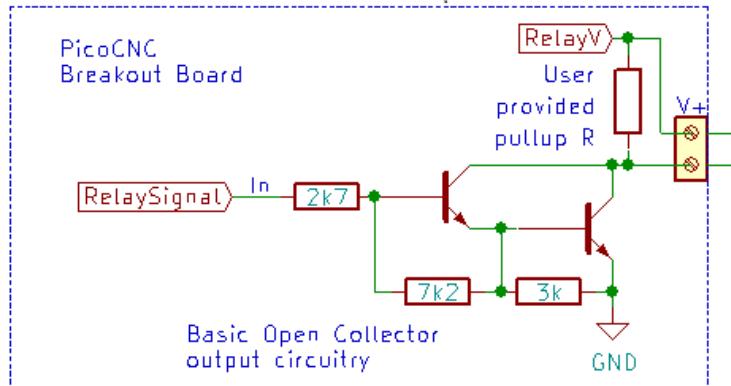
Powered relay boards are driven from the TTL/SSR Relay section pin headers. The output voltage level is +5V – make sure that your PRB is compatible. PRB modules are readily available. Be aware that some cheaper modules use optimistically specified relays. We recommend derating relays contact ratings by at least 20% for safety.

Driving 5V logic devices.

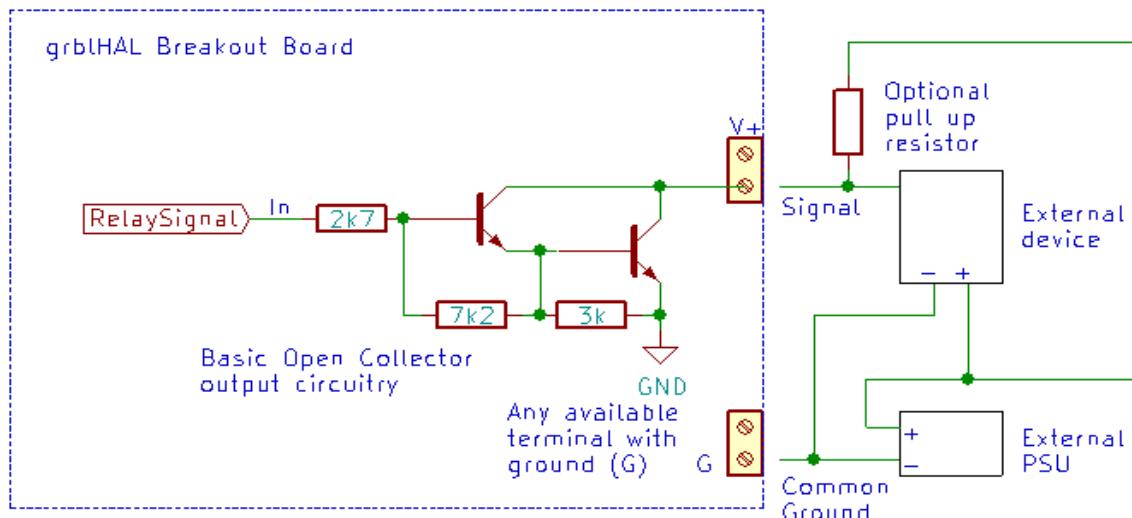
The TTL/SSR Relay output pin headers are usable for general 5V logic signaling. For Aux0-7, use the M62-M65 commands. They can be issued from GCode, a macro in a GCode Sender or from direct GCode input via a GCode sender.

General Purpose Open Collector driver.

The open collector outputs of the relay screw terminals are available for general use. The current is limited to a sum total of 300 mA through all terminals at a 100% duty cycle. The v+ terminals supply either +5V or +12V, determined by the jumper on the Relay Voltage jumper. A common ground may be needed. Any unused ground terminal on the relay or stepper output section the PicoCNC can be used. Do not use a ground pin from the Limit or Control input sections.



You may use any external voltage up to 48V however, we recommend an upper limit of 36V. To use an external voltage, the v+ terminal is not used. Consult the below diagram for details. It is the user's responsibility to determine suitability for their application.



EMI Considerations

Electro Magnetic Interference can cause significant disruptions to your CNC machine. It is best to build your system using good practices. Often your wiring can act as antennas for EMI and conduct it back into your electronics box. In designing the PicoCNC board, we used good practices to avoid EMI but further precautions are recommended. High voltage electrical devices are often a source of EMI and you should take additional steps to ensure that these sources are managed. Finding the source of EMI later can be difficult – we recommend that you design your system with EMI avoidance it in mind.

These are general recommendations, you don't need to follow all of them but the more you do the better chance you stand of having EMI free operations.

- Do not power 5V relay coils from the USB voltage supplied by to the Pico. Relays can produce EMI and power fluctuations. Use external power (see above).
- Do not power relay coils directly from the Stepper Motor power supply. It can be a significant source of EMI.
- Use shielded wires for signals. Make sure the shields are grounded in a star configuration at the controller box or other common ground.
- Use ferrite filters on all wires entering or exiting the PicoCNC.
- Avoid routing wires near EMI sources such as a VFD, Stepper motors, electrical motors or power supplies. Avoid co-routing unshielded wires with wires that are connected to those devices as well.
- If you are connecting to the PicoCNC via a USB cable, use one with a ferrite filter on it or add a clamp-on one close to the electronics box end of the cable.

Ferrules

We recommend that all stranded wires have ferrules crimped on the end that goes into a screw terminal. This is a good practice. It ensures the best possible contact and helps to avoid stray “cat whiskers” - strands of wire that could cause errors. Kits with ferrules and a crimper are readily available for \$20 to 30 USD. It is a worthwhile investment. Use 22 Ga ferrules for the assembly kit screw terminals on the PicoCNC. Larger ones may not fit.

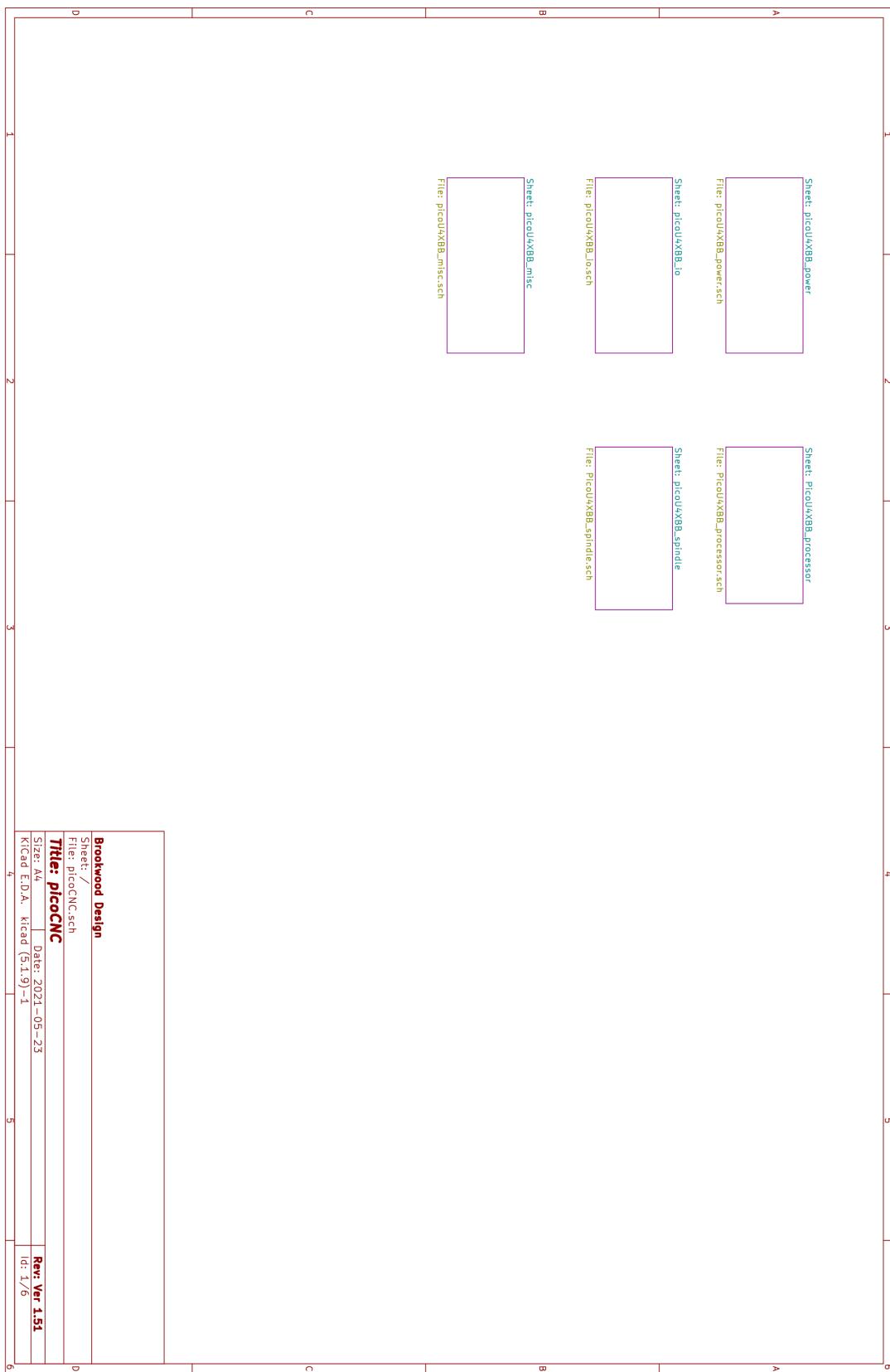


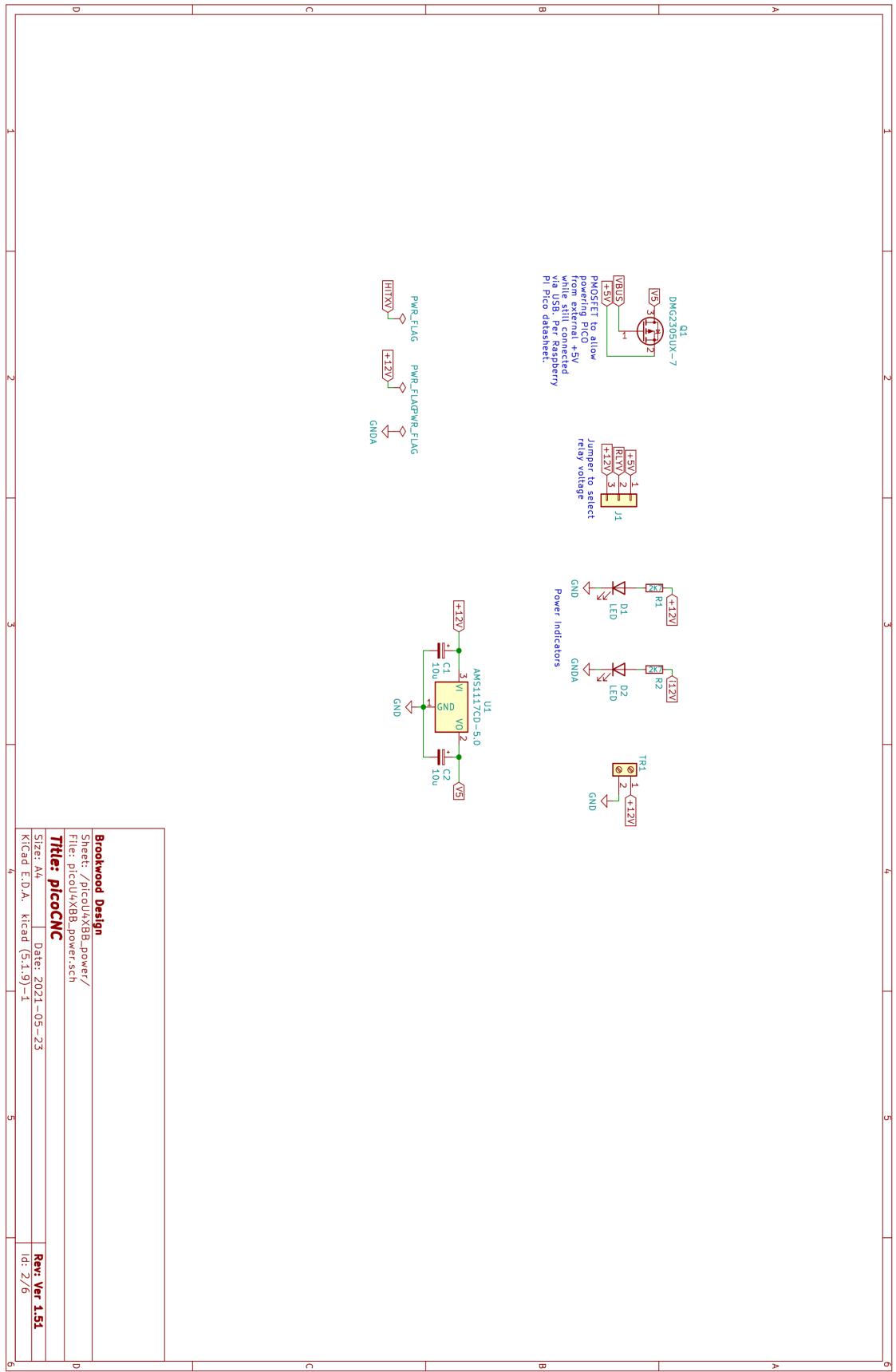
Errata

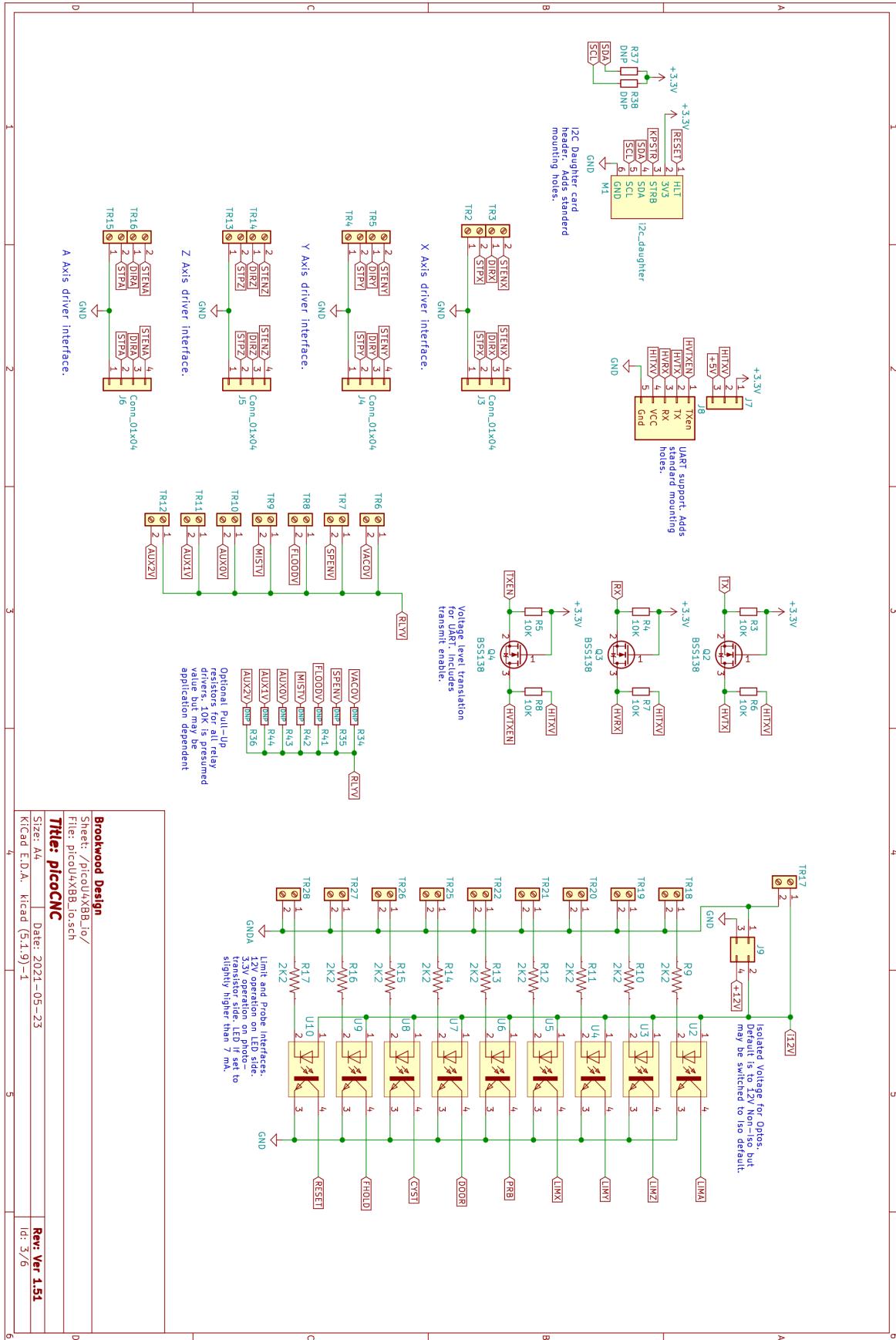
V1.51 PicoCNC

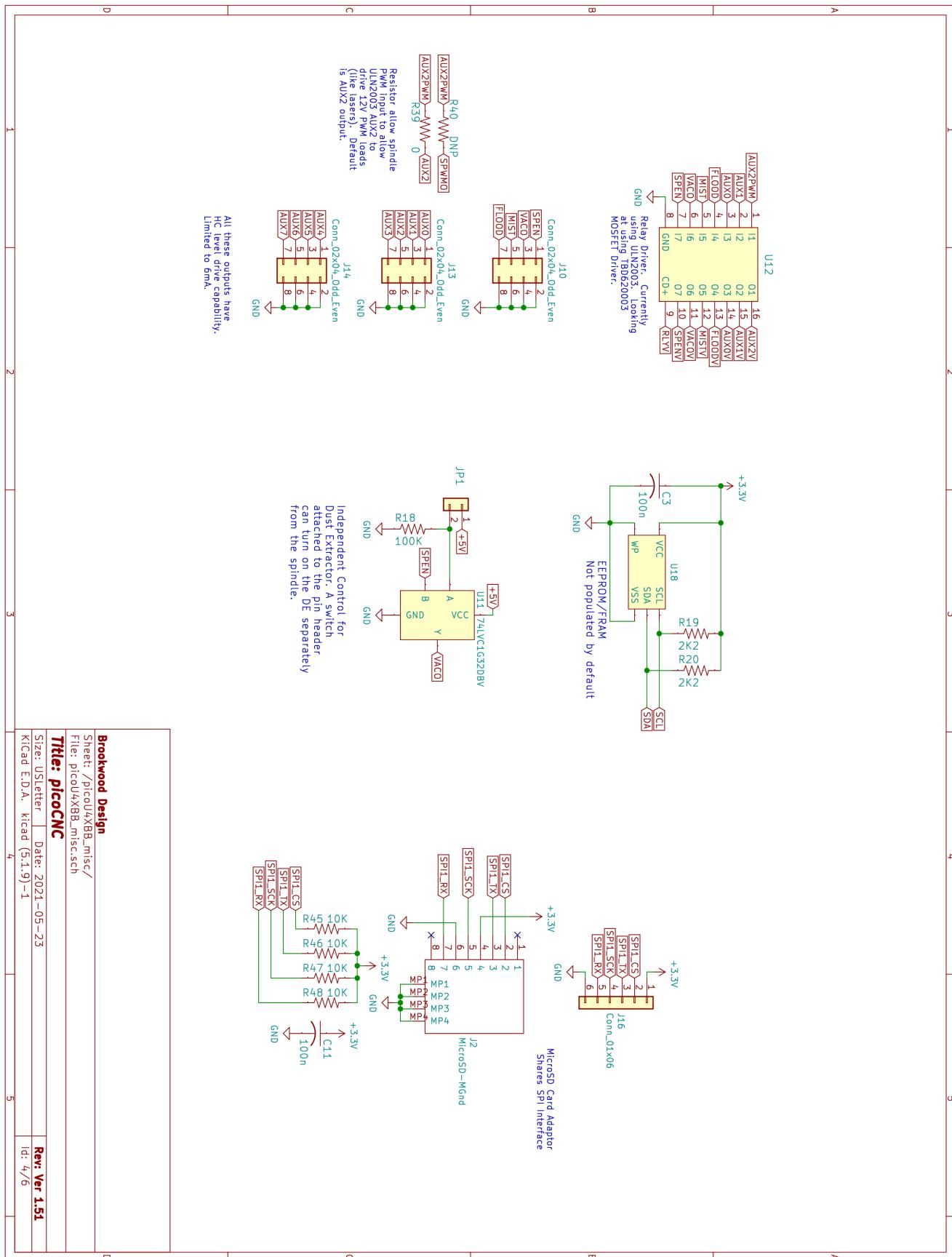
5V Indicator LED incorrectly tied to 5V, not VBUS. Fixed in V1.52.

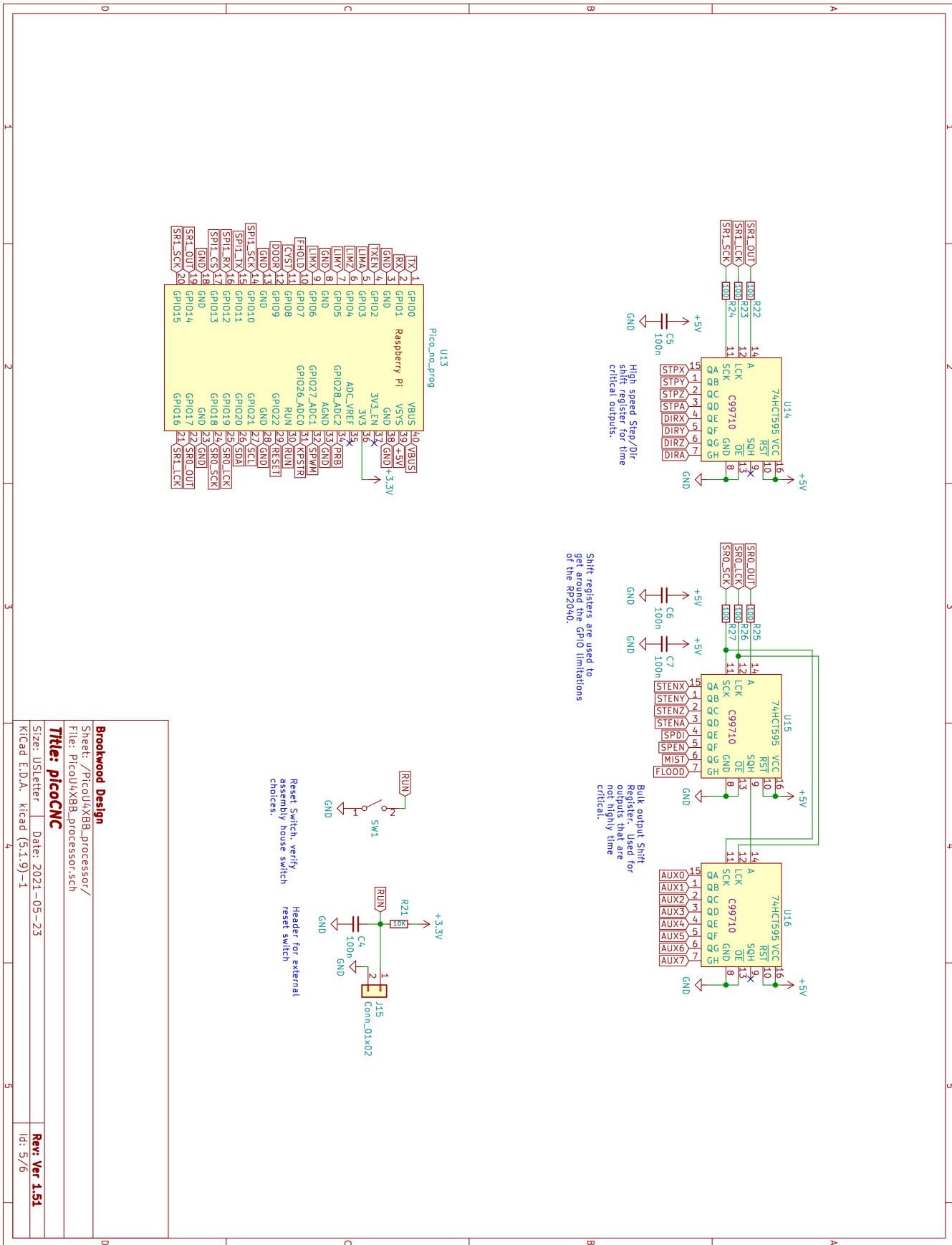
Schematics:

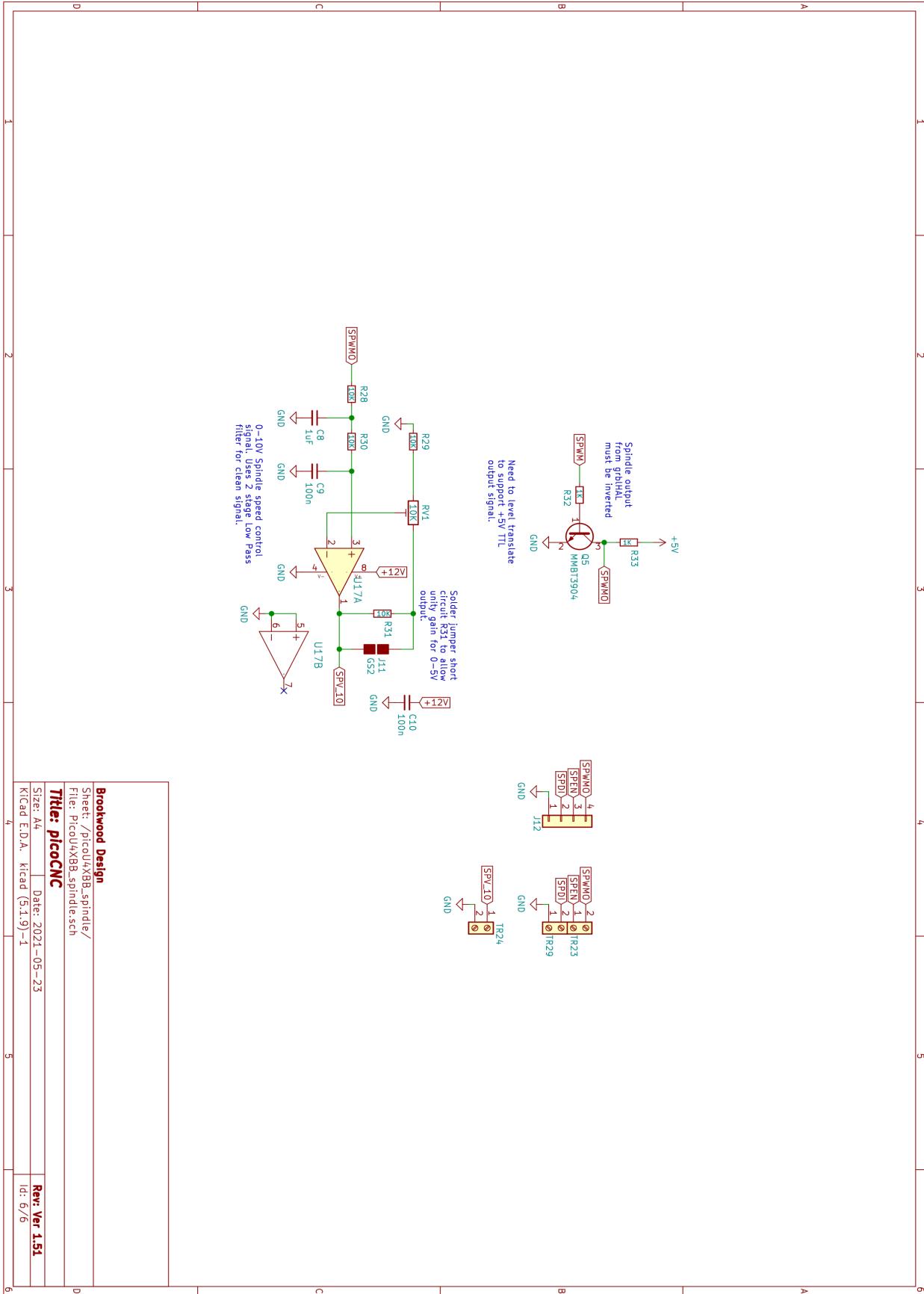




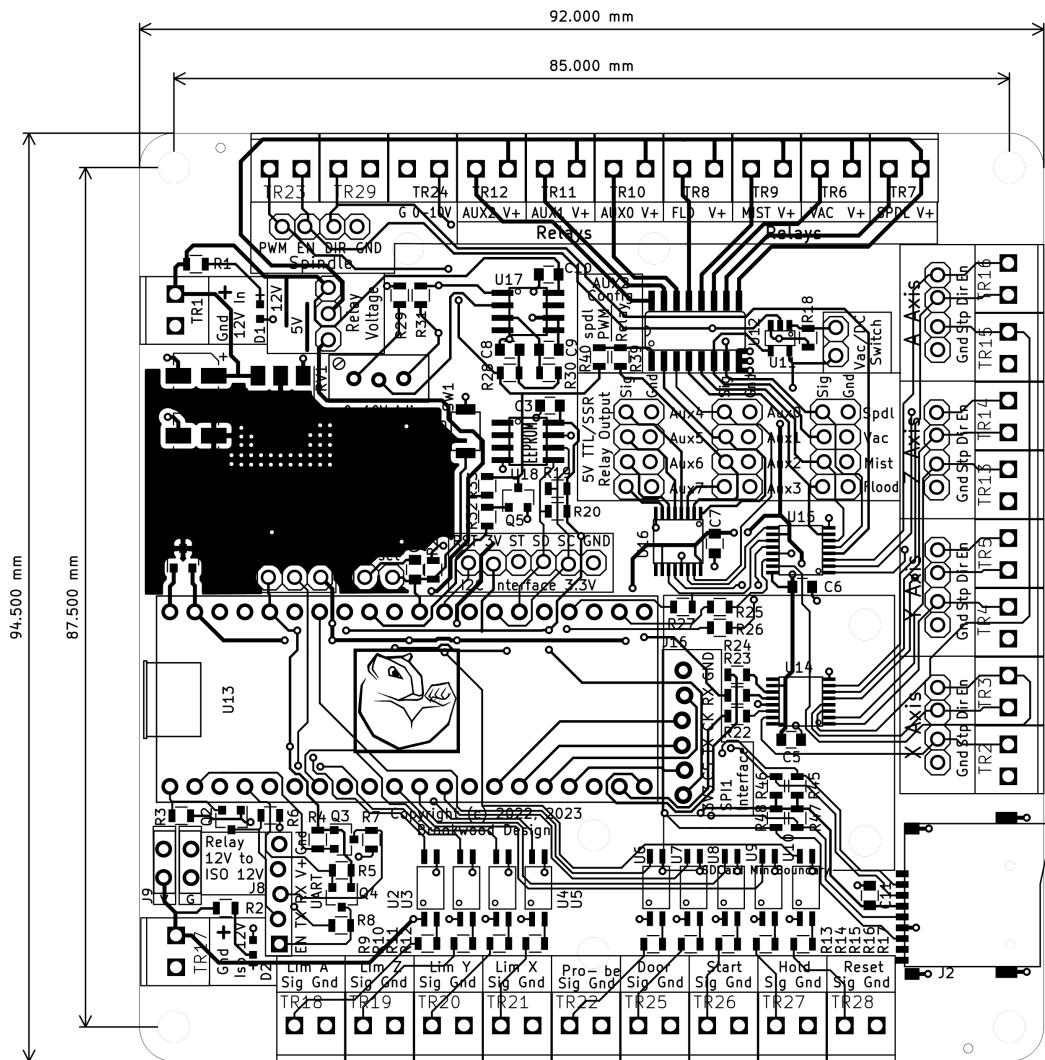








Board and mounting hole dimensions



Assembly Kit Bill of Materials.

Item	Number	Description	Part nums
Pico sockets	2	1x20 pin 2.54mm pitch socket	Various Suppliers. Mouser: 571-1-535541-8
Screw terminals	29	3.5mm pitch screw terminals	KF350-3.5-2P Various Suppliers. LCSC: C474892
2x4 pin header	3	2x4 2.54mm pitch header	Various suppliers. Mouser 538-70287-1051
2x2 pin header	1	2x2 2.54mm pin header	Various suppliers. Mouser: 571-826656-2
1x40 pin header	1	1x40 2.54mm pitch header	Various suppliers. Mouser: 571-9-146280-0
10K Trim Pot	1	10K trim potentiometer	Bochen 3296W-1-103, LCSC: C118954
Shorting Jumper	4	Shorting Jumper 2.54mm pitch.	Various suppliers. Mouser: 855-M7581-05

BOM for surface mount components.