Toolauth | Wing-Combo

Part of the Toolauth Project: https://github.com/Toolauth/

This device is intended to help simplify the process of setting up and managing tool authorization control, in a shared workshop environment.

Specifically, this is designed for use at MakeHaven, in New Haven, Connecticut, USA from a previous solution. The circuit board shown here includes a place for an Adafruit Feather ESP32 microcontroller (with wifi) to sit into place, and manage the functions of the board and ultimately control access to power a tool. This schematic is broken into several pages to better clarify the circuit.

Microcontroller

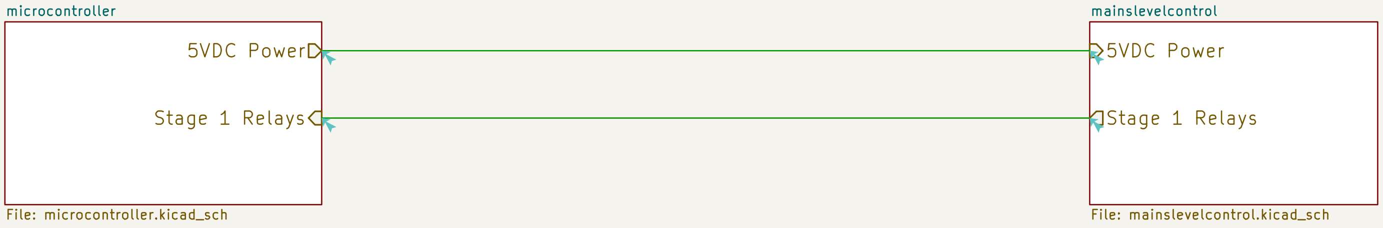
This section of the schematic covers everything that happens at 3.3VDC or 5VDC, including the microcontroller, peripherals, and some of the simple circuits needed to support the operation.

Power for this section is provided by a sealed switching transformer soldered onto the board.

Mains Level Control

This section covers everything that happens at the AC power level: anywhere from 100VAC to 240VAC, provided the corresponding configuration.

There is only a single relay changed to cover voltage this range. All the panel-mount buttons, relays and Emergency Stops that ensure primary function of the tool-control are included here.



Based on Adafruit Feather ESP32 boards

The main MCU board is a 'Feather', and they can have 'Wings' plugged into them.
This is the biggest wing we are aware of: does that make it an albatross?

feather specification: https://learn.adafruit.com/adafruit-feather/feather-specification

Highest Level Overview
Corey Rice & MakeHaven
Sheet: /
File: wing-combo.kicad_sch
Title: ToolAuth Hardware | Overview | ESP32-DEVKITC-V4
Size: USLetter Date: 2022-10-09 **Rev: 3**
KiCad E.D.A. 8.0.6 Id: 1/11

Microcontroller Overview

This shows the relationships between all components running at 3.3VDC or 5VDC on the board. These are all on the 'low voltage' side of the board. These are all united in their service to the Adafruit Feather board. Thinking about this design as a whole, it is our past experience and makerspace spirit that guided us towards making this a Feather Wing rather than integrate the ESP32 on-board. If the microcontroller is removable, it is easier to program, and you can replace a 'problematic feather' for any reason (without needing to rewire the tools).

ESP32 Core

esp32core

File: esp32core.kicad_sch

This shows a component to represent the Adafruit Feather board, as it will be seated in place. However, the component itself is not included on the circuit board, instead seating into some breakout headers for easy replacement. This page is for reference only, not design.

Experience implementing things like this on networks has taught us that it can be handy to be able to get a new device/MAC address without needing to rewire the entire 'box' that connects to the tool.

ADE7953

ade7953

File: ade7953.kicad_sch

The ADE7953 is a high accuracy current, voltage and power measurement chip that communicates with the ESP32 via I2C. There are three 3.1mm audio jacks to enable connecting CT split-ring clamp sensors. These CT sensors can be clamped around the wires that run power to the tool, to monitor the current and voltage draw throughout use. Additionally, there are a number of jumpers that can be soldered to modify the filtering circuits if necessary.

Breakout Board Connections

breakoutboards

File: breakoutboards.kicad_sch

These are the connections for I2C, SPI, and the Feather headers. There is also a 3.3V linear regulator to power these daughter boards, without placing more draw on the Feather's onboard linear regulator (this includes powering the ADE7953).

Lights, Sounds & Status

lightsounds

File: lightsounds.kicad_sch

This is a collection of the simple circuits that translate between voltage levels, break out the programming buttons for the Feather, control the buzzer, and control the indicator LEDs. Many of these circuits use a mix of 3.3v and 5 volts to be as bright/loud as possible.

USB-C & Solder | Daughter Power

usb-c-power-out

File: usb-c-power-out.kicad_sch

It is foreseeable that daughter baords will be added, and these connections allow for the ability to power them without needing a second AC/DC Power Supply. These are fused to not over-draw from the main baords/ uses.

Corey Rice & MakeHaven

Sheet: /microcontroller/

File: microcontroller.kicad_sch

Title:

Size: USLetter | Date: 2022-10-09 | **Rev: 3**

KiCad E.D.A. 8.0.6 | Id: 2/11

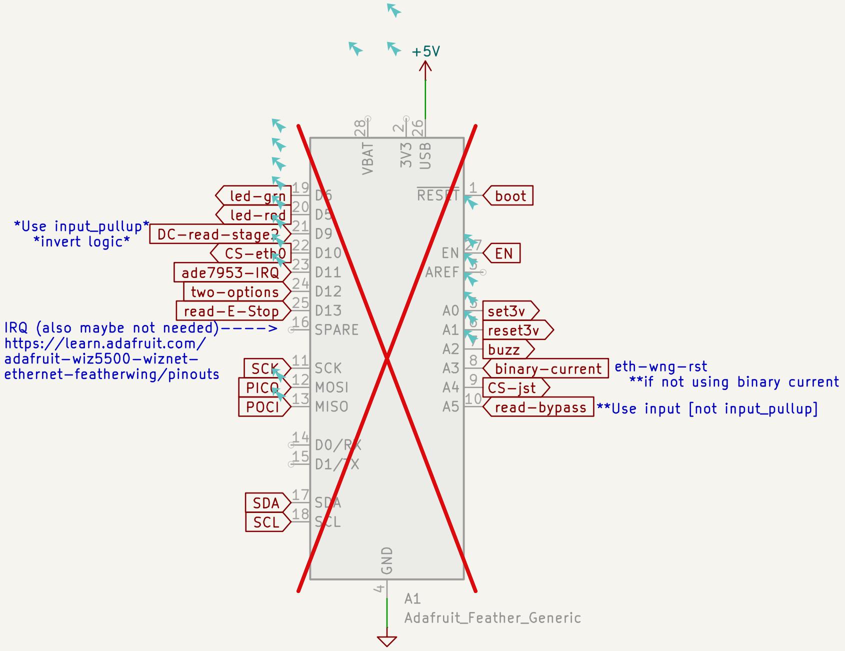
feather specification:
https://learn.adafruit.com/adafruit-feather/feather-specification

Feather Compatable Board

This shows a component to represent the Feather Footprint board, as it will be seated in place. However, the component itself is not included on the circuit board, instead seating into some breakout headers for easy replacement. This page is for reference only, not design.

———————————————

We love Feather boards because it is a specification, not just a single vendor (but we <3 Adafruit). That way we can more easily source and/or upgrade boards over the years we hope to implement this system.



See headers for actual connections
Connections shown, but part excluded from BOM.
Corey Rice & MakeHaven
Sheet: /microcontroller/esp32core/
File: esp32core.kicad_sch

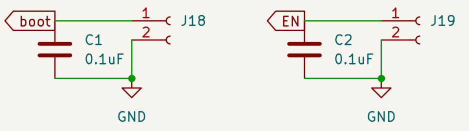
Title: Connections to ESP32 Board

Size: USLetter | Date: 2022-10-09 | **Rev: 3**
KiCad E.D.A. 8.0.6 | Id: 3/11

Lights, Sounds & Status

Programming Buttons for ESP32

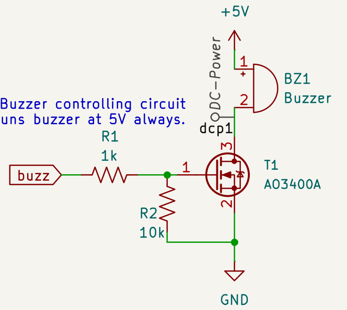
access to small buttons outside of case
[just for easy reprogramming of ESP32]



Piezo Buzzer Audio Feedback

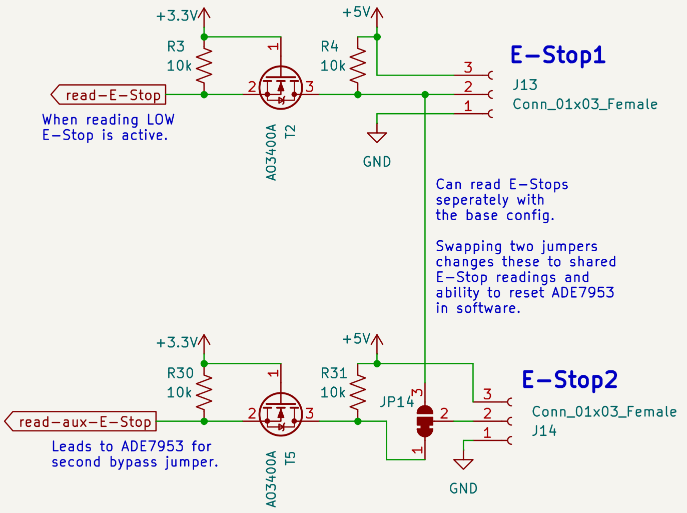
A small piezo buzzer is loud enough to provide
some auditory feedback through the toolauth box.

Buzzer controlling circuit
runs buzzer at 5V always.



Read the E-Stop

An E-Stop is an essential Safety Measure in any
workshop. This is a secondary switch on the E-Stop
assembly (totaling 3 switches and an LED in the
standard config). This circuit allows the ESP32 to
read the state of the E-Stop.
! A second E-Stop low voltage circuit can be put in !
! parallel on one JST if an Auxilary E-Stop in use. !



When reading LOW
E-Stop is active.

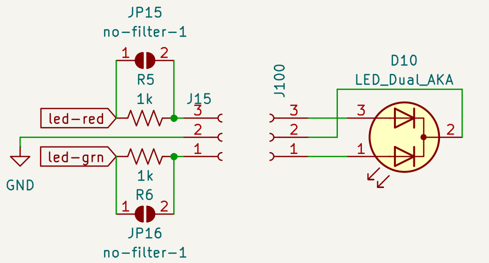
Can read E-Stops
seperately with
the base config.

Swapping two jumpers
changes these to shared
E-Stop readings and
ability to reset ADE7953
in software.

Leads to ADE7953 for
second bypass jumper.

Panel Mount Status LED

A single panel mounted Red/Green LED will indicate states
of the system for every card read. These LEDs are cheap
and relatively easy to find in pre-made panel mount packages.



Breakout connections for the EN & Boot pins, for possible panel mount
DIP switches to describe hardware state
Logic Level shifters to read states of E-Stop(s) and Bypass Key
Buzzer and Panel Mount LED for Audio feedback to user
Corey Rice & MakeHaven
Sheet: /microcontroller/lightsounds/
File: lightsounds.kicad_sch

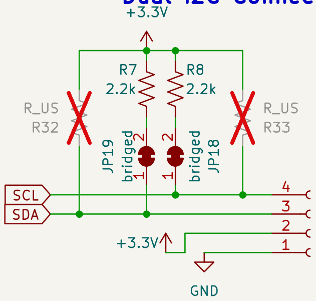
Title: Onboard Lights, Sounds and Logic Level Converters

Size: USLetter | Date: 2022-10-09 | **Rev: 3**
KiCad E.D.A. 8.0.6 | Id: 4/11

Breakout Board Connections

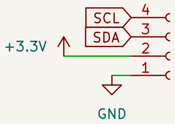
It takes more than just one board to make these controllers work.

Dual I2C Connections



The I2C bus connection is used by the ADE7953, but here are 2 more that add more extensibility to the board as a whole. One of these will may be used by a PN532 RFID Card Reader.

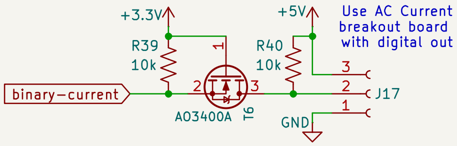
NOTE: Cut jumpers & solder in THT resistors if you find the I2C data has a bad rise-time/ connection.



Backup Plan for Current Measurement

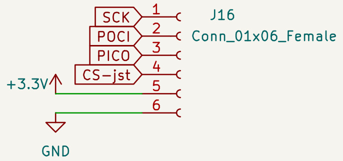
The ADE7953 can be tricky to fully implement, and we know that an old CT Clamp breakout board has worked for years. For that reason, we added this ‘backup plan’ header to easily attach an AC current sensor breakout board (with digital out) if you need a fallback.

[!WARNING] You must use an AC current sensor here, with digital output or this will not work. Try this one: https://a.co/d/aXBdB3J



Single SPI Bus

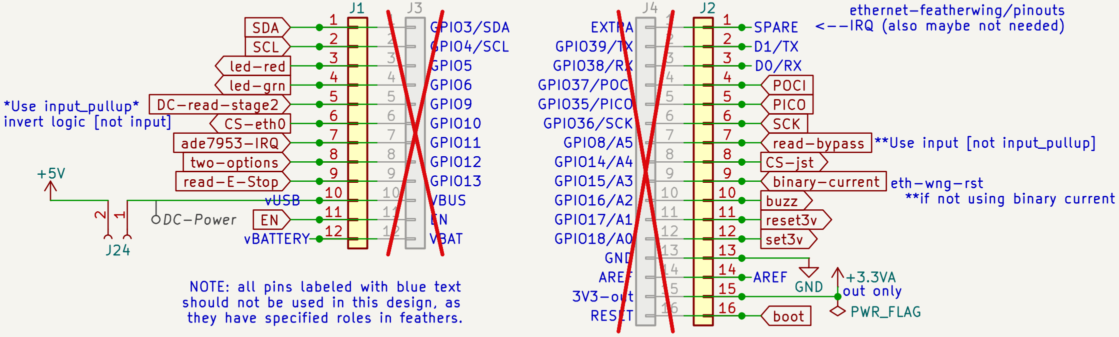
Here is a single SPI connection in a 6-pin JST configuration. You will need to verify the alignment of these pins with whatever you are connecting.
Also: be mindful of the new PICO notation versus the legacy MOSI notation for boards.



feather specification: https://learn.adafruit.com/adafruit-feather/feather-specification

Headers for easy access to all of the ESP32 Pins

All of the connections are broken out to secondary pins, so they can be accessed fairly easily. This will probably not be needed in normal operation, but it definitely helps to diagnose hardware problems or give easy access to the happy matter-hackers that live in our space :)

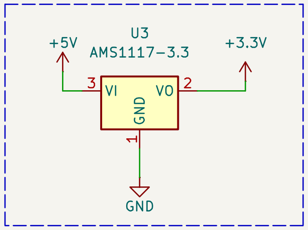


https://learn.adafruit.com/adafruit-wiz5500-wiznet-ethernet-featherwing/pinouts
<--IRQ (also maybe not needed)

NOTE: all pins labeled with blue text should not be used in this design, as they have specified roles in feathers.

Linear Regulator

Power demands on the Feather can be handled by its own linear regulator. This one is added to handle all breakouts on this board, including the ADE7953, in hopes that the draw of the microcontroller and any wings on top are isolated from the 3.3volt needs of this board.



>[!NOTE] The Feather will be powered by the onboard AC/DC power supply on the Wing by default. That means:
> NEVER plug your computer’s USB port into the Feather while the Wing is powered from the AC/DC power supply!
> This would backpower your Computer’s USB and can cause serious harm to the computer.
> -- Disconnect Jumper J24 if you need to attempt this, but it is NOT recommended [it is NOT completely safe]
>
> Here is what you SHOULD do:
>   1. Program the Feather over USB (first time only) while it is NOT SEATED in the Wing-Combo board.
>   2. Disconnect Feather from computer and seat in Wing-Combo board. It should turn on and connect to WiFi.
>   3. Reprogram Feather as needed Over-The-Air only. [Never put a USB from inside the box into a computer.]

If your Feather requires 5VDC via USB only (not advised for most Feather boards).
Step 1: Pull the jumper from J24, and leave it disconnected.
Step 2: Proceed with oddly powering the Feather from the USB-C port available on the Wing-Combo board.
This will require a short jumper USB (just a few centimeters) and be very goofy. Only do this for very good reason...
https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/power-management#alternative-power-options-3122391

Binary Current sensor hookup (backup plan)
A seperate Linear Regulator to better supply 3.3VDC where needed
Headers for the Feather to seat into the board
Breakouts for I2C and SPI communication with external boards.
Corey Rice & MakeHaven
Sheet: /microcontroller/breakoutboards/
File: breakoutboards.kicad_sch

Title: Breakout Connections & Power

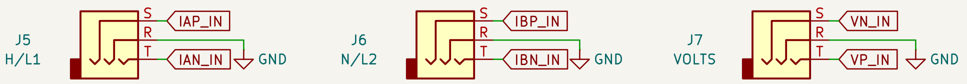
Size: USLetter   Date: 2025-09-25   **Rev: 4**
KiCad E.D.A. 8.0.6   Id: 5/11

ADE7953 Current Sensor

https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7953.pdf

This current sensor was selected because it was known to work with the ESP32 and ESPHome (the main platforms of this project at large). It took several test-boards to get to a useful integration, yet much of this still just follows the documentation linked above. There are jumpers that can be soldered over, if the circuit needs any kind of fine tuning from your own observations.

CT Clamp Connectors: 3.5mm Audio & two-pin backup



IAP_IN
IAN_IN
1
2
J23
H/L1

<-- This allows you to solder in a CT clamp instead of using the 3.5mm jacks.
While this can let you shorten the lines, it also can be a trap. NOT advised.

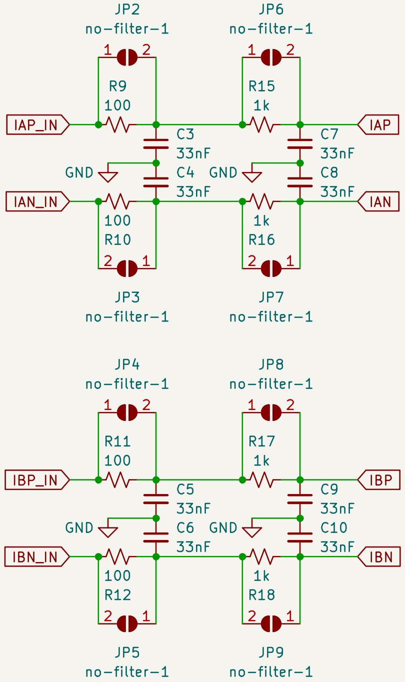
Measuring AC Current is hard.

If the ADE7953 (this page) is not working for you, try new I2C resistors. Or fully bail on the ADE7953 and use an AC current sensing breakout on our 'backup plan' bin-cur connector. [sometimes we do, no shame]
– previous page in schematic

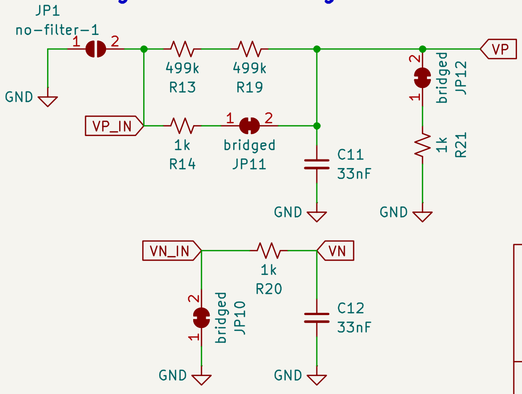
In the future, we may get a more elegant solution to monitor the electric current drawn by the tool. There are several possible future options to explore:

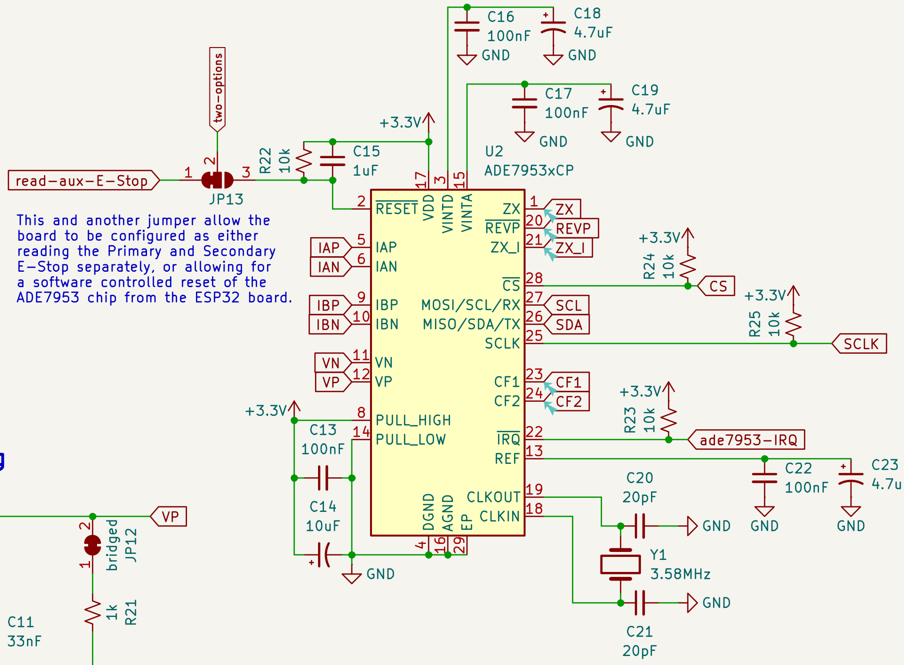
- the TI ADS1115 over I2C (like this)
- on-board shunt resistor options
- a WCS1800 on board, configured for AC

Current Sensors Filtering



Voltage Sensor Filtering





This and another jumper allow the board to be configured as either reading the Primary and Secondary E-Stop separately, or allowing for a software controlled reset of the ADE7953 chip from the ESP32 board.

3 connectors added, to read 2 current channels and 1 voltage
Pull-up resistors added to configure chip for I2C communication
ADE7953 implemented as in engineering development board
Corey Rice & MakeHaven
Sheet: /microcontroller/ade7953/
File: ade7953.kicad_sch

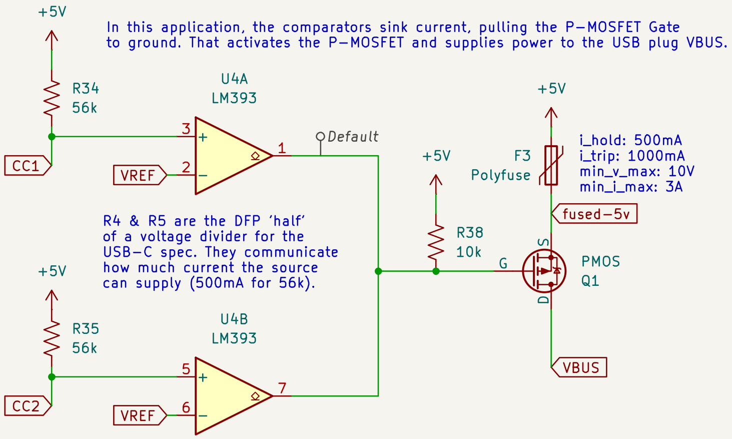
Title: ADE7953 Current Sensor

Size: USLetter | Date: 2024-12-13 | **Rev: 3**
KiCad E.D.A. 8.0.6 | Id: 6/11

5V Daughter Boards: USB-C Source and Pins

While it is not entirely clear how these will be used, they provide options for powering secondary daugher boards without needing to add a second AC/DC Power Supply unit to the tool-controller box. The easiest attachment is a USB-C source plug (technically a DFP) that is basically compliant with the proper USB-C specifications. Also, there is a pair of solderable holes that allow for a more-direct connection to the 5V line, but are still guarded by the resettable fuse. Either of these could be used to add a panel-mount USB-C port, and then attach a second MCU for any number of purposes: we think an E-Ink display for machine status could be really helpful :)

In this application, the comparators sink current, pulling the P-MOSFET Gate to ground. That activates the P-MOSFET and supplies power to the USB plug VBUS.

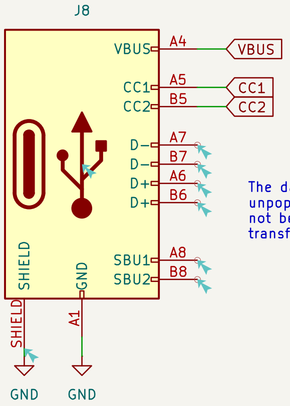


R4 & R5 are the DFP 'half' of a voltage divider for the USB-C spec. They communicate how much current the source can supply (500mA for 56k).

The Upstream-Facing Port (UFP) will supply 5.1k resistor(s) on at least one of the plug's `CC1` or `CC2` lines

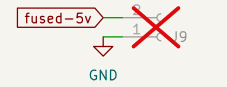
Power-only USB

This USB port only supplies power, not a data connection anywhere.
If you remove JP24, this can power the feather [NOT advised].
The USB can power secondary microcontrollers (like for e-Ink).

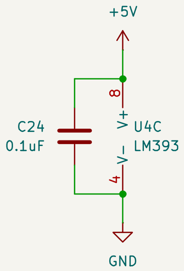


The data pins are all unpopulated, as we wil not be using them to transfer any data.

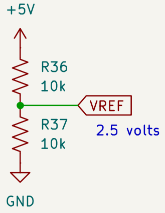
This allows for soldered-in items as well, that are also limited by the resettable fuse.



Power for Comparator



A stable voltage divider for the two Comparators' inverting inputs.



This is a minimum-viable USB-C source.

Connecting a USB-C device to this port will activate at least one comparator, and turn on power to the `VBUS` only after a connection is made. This behavior is called 'Cold Plugging' and is an expectation of USB-C Downside-Facing Ports (DFP).

The fuse will limit the current to roughly 500mA, and trip if 1A is ever pulled from this device. This is another expected behavior of USB-C DFPs: basic over-current protection.

Sheet: /microcontroller/usb-c-power-out/
File: usb-c-power-out.kicad_sch

Title: 5V USB-C and Pins | Daughter Board Power

Size: USLetter | Date: 2025-10-04 | **Rev:**
KiCad E.D.A. 8.0.6 | Id: 7/11

Mains Level Control

The original version of these tool controllers (before this design)
required that this mains–level circuit be built by hand, every time.

The goal here is to build a latching relay system from non–latching relays (for reset on power failure)
with the added benefit of available E–Stops. Although a bit redundant, this design breaks up the
control of power–access to a tool into three stages to manage voltage levels and needed current.

––> 1) Twin small relays that are controlled by a microcontroller, which allow access to use tool

––> 2) A larger relay that is triggered and controls mains–level voltage (also has 110 & 220 versions)

––> 3) An off–board contactor that actually transmits the full power needs of the tool

Stage 1

stage1relays

File: stage1relays.kicad_sch

This includes both logic–
level relays circuits.

The page also shows the
5VDC sealed transformer
and a simple 'sanity
check' power LED :)

Stage 2

stage2relay

File: stage2relay.kicad_sch

This includes the one
mains–controlled relay.

Also, the logic of the
main latching circuit is
outlined on this page.

The fuses are included here.

Stage 3

stage3contactor

File: stage3contactor.kicad_sch

With an off–board contactor
this shows the logical loop
that connects to the device.

The screw–terminal connectors
are also shown on this page.

Corey Rice & MakeHaven
Sheet: /mainslevelcontrol/
File: mainslevelcontrol.kicad_sch

Title:

Size: USLetter	Date: 2022–10–09	**Rev: 3**
KiCad E.D.A. 8.0.6		Id: 8/11

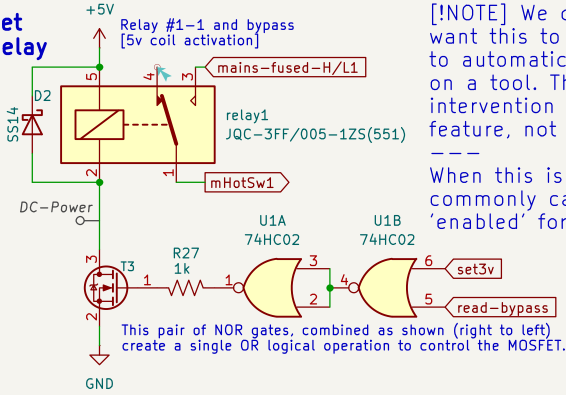
Stage 1 || Control from 5V & 'set' or 'reset' Stage 2 Latch

Relay1 [NOFF 'set' the Stage 2 relay's Latch] {5VDC coil activation}
Relay2 [NON 'reset' the Stage 2 relay's Latch] {5VDC coil activation}

This stage is responsible for allowing the Feather to control the actions of the mains level of power switching.

The 'SET' relay allows the tool to be turned on with the additional press of the green button by the human operator. Bypass locks this relay 'ON'

Set Relay

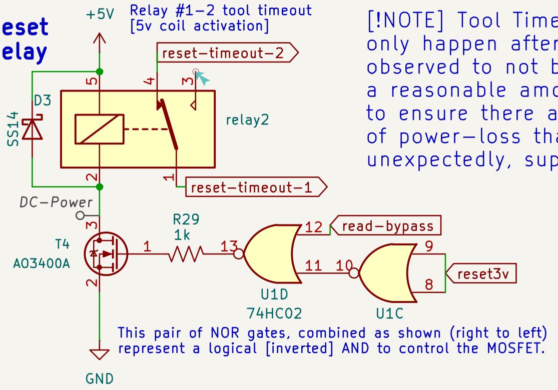


This pair of NOR gates, combined as shown (right to left) create a single OR logical operation to control the MOSFET.

[!NOTE] We didn't want this to be able to automatically turn on a tool. The human intervention is a feature, not a bug.
–––
When this is 'ON' we commonly call the tool 'enabled' for use.

The 'RESET' relay will break the electrical self–latch on the Stage 2 relay. Used as a way to auto shut–off the tool, in event of timeout. Bypass locks this relay 'OFF'

Reset Relay

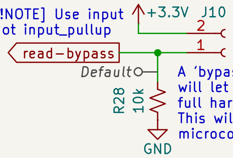


This pair of NOR gates, combined as shown (right to left) represent a logical [inverted] AND to control the MOSFET.

[!NOTE] Tool Timeouts should only happen after a tool is observed to not be in use for a reasonable amount of time to ensure there are no cases of power–loss that happen unexpectedly, suprising a user.

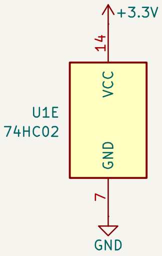
Physical 'Bypass Key' to lock–ON tool

[!NOTE] Use input not input_pullup



[!TIP] if key is ON, these connect

A 'bypass key' mounted on the case will let anyone with the key use a full hardware–override of the system. This will also be readable by the microcontroller, for feedback.



Boolean Logic for Hardware Bypass

[!NOTE] Experience has taught us that the 'Bypass' should always be predictable, and when you need to use it, you shouldn't have to worry about what a microcontroller or server is thinking. We just want the tools to work in bypass.

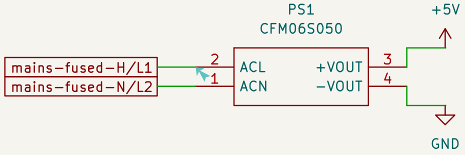
When `read–bypass` is ON this turns ON the `set` relay, but turns OFF the `reset` relay. By DeMorgan's Laws, that gives us the connections shown above. Note: the final NOT of the `reset` relay logic is in the relay itself.

check out our mock–up: tinkercad.com ~ sr–relays–bypass–74hc02–b1p3

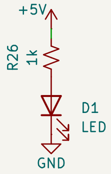
`set` relay: (ON if `set3v` is HIGH) OR (ON if read–bypass is HIGH [B–key is ON])
`reset` relay: ON if read–bypass is LOW [B–key is OFF] AND reset3v is HIGH
--> else: `reset` relay is OFF [B–key can block `reset` relay]

5V supply from mains (in box)

Sealed switching power supply, soldered in place.
Can accept 90–264VAC to generate 5V, up to 6W.
100mV ripple and 78% efficient, by datasheet.



sanity check power LED



A 'sanity check' LED is included to see that 5VDC is functioning
Mains voltage is converted to 5VDC for the microcontroller and more
Boolean Logic so Bypass Key can cause predictable behavior, blocking all else.
Twin Stage 1 relays 'set' and 'reset' the Stage 2 latching relay

Corey Rice & MakeHaven

Sheet: /mainslevelcontrol/stage1relays/
File: stage1relays.kicad_sch

Title:

Size: USLetter	Date: 2025–02–24	**Rev: 4**
KiCad E.D.A. 8.0.6		Id: 9/11

Stage 2 || Core Functions run at full 110-230VAC

Relay 2 [main function] {110VAC or 240VAC coil activation}

active self-latching of a relay   https://electronics.stackexchange.com/questions/223691/self-latching-relay

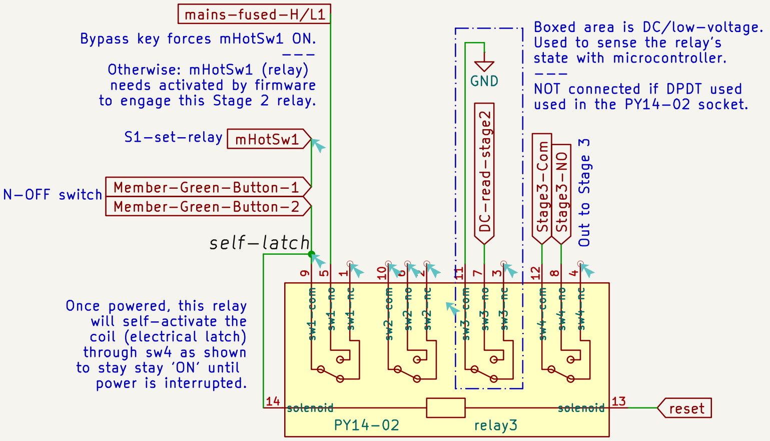
There are multiple variants of Omron's MY series relays, which allow for various coil voltage levels.
---> And this footprint also works with Omron's G2A relays, if there is ever a parts shortage...

Provides the majority of the Member's interface experience: This is the latch that powers the contactor.

The previous Tool-Controller design was just this section of the circuit (with simpler Microcontroller connection).

'Set' & 'Latch' circuit on Stage 2 relay

This portion of the circuit allows the Stage 2 Relay to turn ON & stay ON. When enabled [by Feather or Bypass] a person can turn ON the relay with the green button. Once activated, the link from relay pin 12-to-8 'latch' the relay ON while powered, until reset. As long as 4PDT relay is used, microcontroller can sense relay state.



Bypass key forces mHotSw1 ON.

Otherwise: mHotSw1 (relay)
needs activated by firmware
to engage this Stage 2 relay.

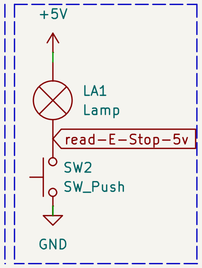
Boxed area is DC/low-voltage.
Used to sense the relay's
state with microcontroller.

NOT connected if DPDT used
used in the PY14-02 socket.

Once powered, this relay
will self-activate the
coil (electrical latch)
through sw4 as shown
to stay stay 'ON' until
power is interrupted.

Relay State Feedback to Microcontroller

The firmware is more manageable if we sense the state of the relay (instead of infer the state from any electrical current readings).
As long as 4PDT relay is used, microcontroller can sense relay state.
DPDT versions of Omron MY relays will fit, but not connect this circuit.



E-Stop on modular, panel-mount button

The E-Stop is connected from a 3-wire header:
>> 5V goes 'above' the E-Stop's lightbulb
>> Sig/Read goes 'between' E-Stop and switch
>> GND goes 'below' switch

The E-Stop light runs at 5V, yet is shown here
to be closer to the relevant 'reset' features.

This lamp-to-switch connection is done on the
modular E-Stop switches themselves, not any PCB.

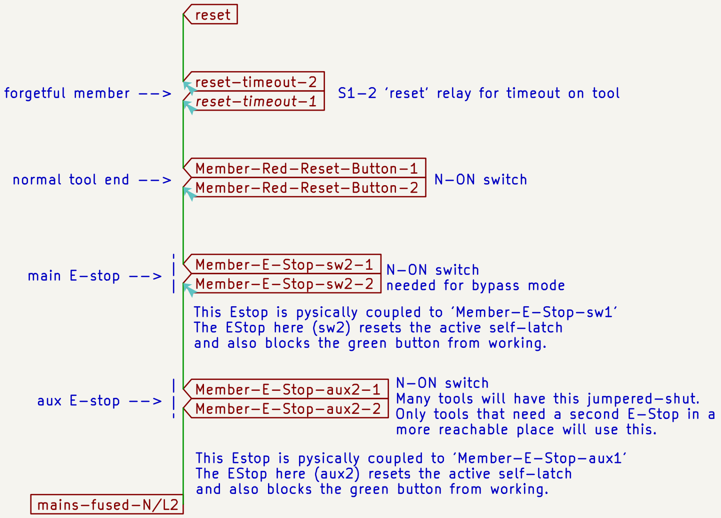
The middle 'read-E-stop' informs the microcontroller.

If LED lamp us used, put resistor ABOVE the 'read' location.

'Reset' circuit on Stage 2 relay

These are all the ways to 'reset' the Stage 2 relay.
- (Normally) a member will press the red button
- (Emergency) the E-Stop to quickly end function
- timeout could trigger Stage1 Reset relay (if programmed correctly...)
- overall loss of power, which also causes reset

The connections needed for the above, are shown below:



S1-2 'reset' relay for timeout on tool

N-ON switch

N-ON switch
needed for bypass mode

This Estop is pysically coupled to 'Member-E-Stop-sw1'
The EStop here (sw2) resets the active self-latch
and also blocks the green button from working.

N-ON switch
Many tools will have this jumpered-shut.
Only tools that need a second E-Stop in a
more reachable place will use this.

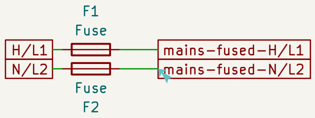
This Estop is pysically coupled to 'Member-E-Stop-aux1'
The EStop here (aux2) resets the active self-latch
and also blocks the green button from working.

Both E-Stops (primary & aux):
1) shut off the tool at two 'layers' for redundancy
2) resets the active self-latch on the stage 2 relay
3) block green button use by member, locking out restarts
All this works in redgular + bypass mode.

Fuse Protected Power Inputs

Mains power needs connected here, and will power the whole box.
Except, perhaps, the tool. If more exotic power is needed.

Labeled on PCB as H/L1 & N/L2 to cover the bases (that both may be live).



E-Stop low voltage light & switch logic shown for reference
Fuses added to power inputs, to protect board from shorts/ bad wiring
Can be interrupted by E-Stops or the S1 'reset' relay
Once activated, relay holds itself in an active state until reset
Corey Rice & MakeHaven

Sheet: /mainslevelcontrol/stage2relay/
File: stage2relay.kicad_sch

Title: Stage 2 | Core 'Active Self-Latching' relay

Size: USLetter   Date: 2022-10-09   **Rev: 3**
KiCad E.D.A. 8.0.6   Id: 10/11

Stage 3 ||Final Contactor & Screw Terminals

Output to control the Contactor, and a light to show the state.

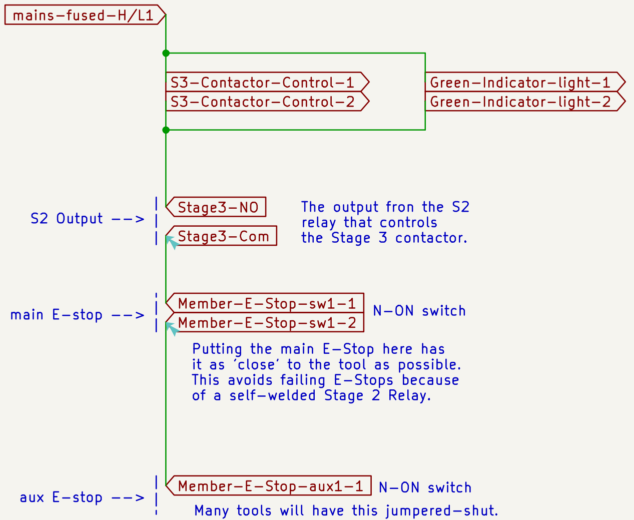
Control for Final Contactor

The contactor is not shown here, only the connection point labeled as "S3-Contactor-Control-{1or2}" which is a screw terminal connection point.

The Stage 3 contactor can supply power to the tool, directly from the mains input power. However, all that is shown here is for the lower-current control loop that activates the contactor. Tool-current should never paass through this PCB.

>[!WARNING] Not for use with latching contactors.

The E-Stop switches shown here can cut power to the contactor in the unlikely event that their twin switch (sw2 | each E-Stop has 2 switches) is fused shut, and the stage 2 relay stays powered. A backup to ensure the E-Stop is effective.



The output fron the S2 relay that controls the Stage 3 contactor.

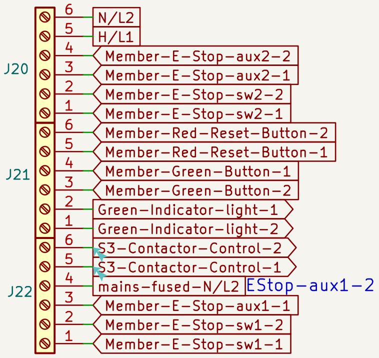
Putting the main E-Stop here has it as 'close' to the tool as possible. This avoids failing E-Stops because of a self-welded Stage 2 Relay.

Many tools will have this jumpered-shut. Only tools that need a second E-Stop in a more reachable place will use this.

Screw Terminals for all functions

The terminals are in pairs, and labeled on the board.

PCB screenprinting does not necessarily match the tracenames/ flags shown here. Instead, silkscreen labels are designed to be as user-friendly as possible. They are explained in this repo's wiki.



Screw-terminals shown here, to connect to mains-level devices
E-Stops to interrupt power from the Stage 2 relay to contactor coil
The contactor actually controls power access to the tool
Corey Rice & MakeHaven
Sheet: /mainslevelcontrol/stage3contactor/
File: stage3contactor.kicad_sch

Title: Stage 3 | Contactor to power the Tool

Size: USLetter | Date: 2022-10-29 | **Rev: 3**
KiCad E.D.A. 8.0.6 | Id: 11/11