

# Toolauth | Wing-Combo

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

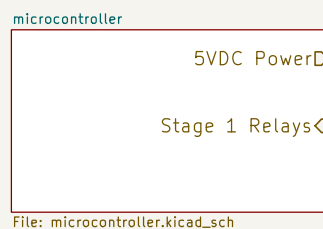
This device is intended to help simplify the process of setting up and managing tool authorization control, in a shared workshop environment.

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.

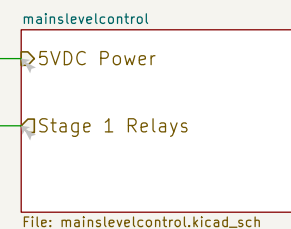
Power for this section is provided by a sealed switching transformer soldered onto the board.



## 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: 2025-11-02

KiCad E.D.A. 9.0.6

**Rev: 1.4.3**

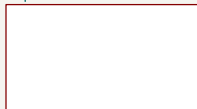
Id: 1/11

# 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



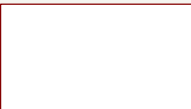
File: esp32core.kicad\_sch

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

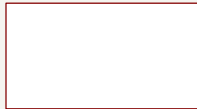


File: ade7953.kicad\_sch

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

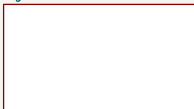


File: breakoutboards.kicad\_sch

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

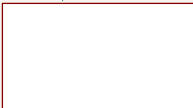


File: lightsounds.kicad\_sch

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.

All subpages for the Microcontroller level items

Corey Rice & MakeHaven

Sheet: /microcontroller/

File: microcontroller.kicad\_sch

Title:

Size: USLetter

Date: 2025-11-02

Rev: 1.4.3

KiCad E.D.A. 9.0.6

Id: 2/11

feather specification:

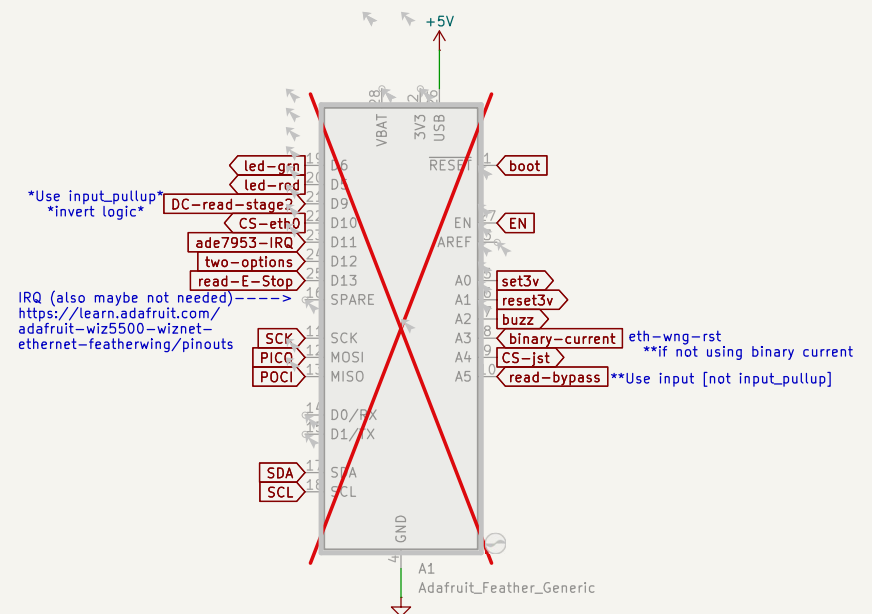
<https://learn.adafruit.com/adafruit-feather/feather-specification>

# 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 actual connections for the Feather board (by standard Feather footprint)  
Connections shown, but part excluded from BOM.

**Corey Rice & MakeHaven**

Sheet: /microcontroller/esp32core/  
File: esp32core.kicad\_sch

**Title: Connections to ESP32 Board**

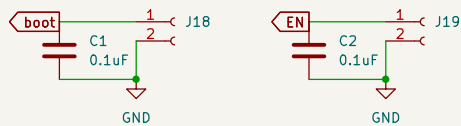
Size: USLetter Date: 2025-11-02  
KiCad E.D.A. 9.0.6

**Rev: 1.4.3**  
Id: 3/11

# Lights, Sounds & Status

## Programming Buttons for ESP32

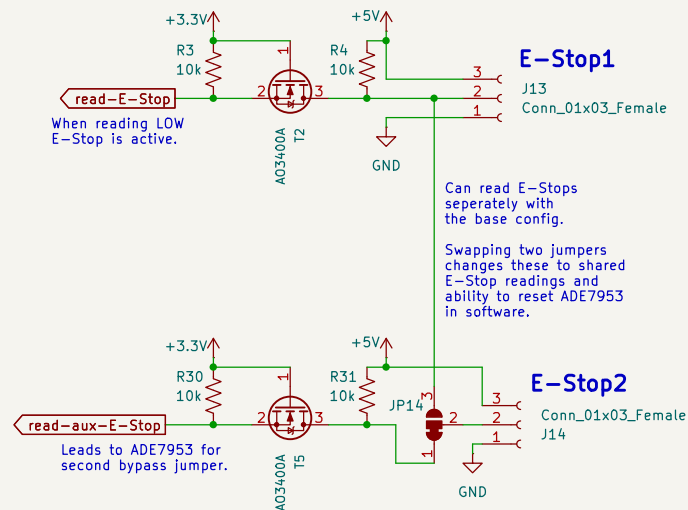
access to small buttons outside of case  
[just for easy reprogramming of ESP32]



## Read the E-Stop

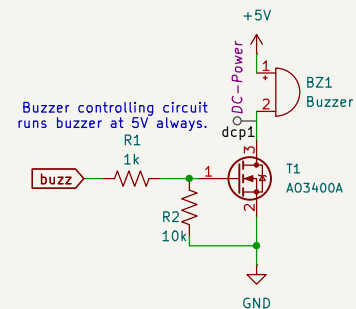
An E-Stop is an essential Safety Measure in any workshop. This is a secondary switch on the E-Stop assembly (totaling 3 switches and an LED in the standard config). This circuit allows the ESP32 to read the state of the E-Stop.

! A second E-Stop low voltage circuit can be put in !  
! parallel on one JST if an Auxiliary E-Stop in use. !



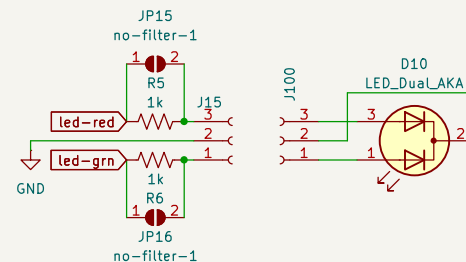
## Piezo Buzzer Audio Feedback

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



## 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: 2025-11-02

Rev: 1.4.3

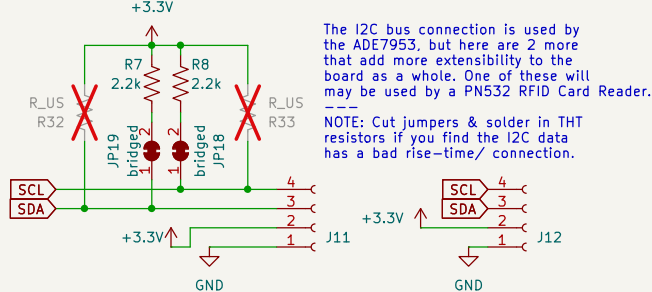
KiCad E.D.A. 9.0.6

Id: 4/11

# Breakout Board Connections

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

## Dual I2C Connections

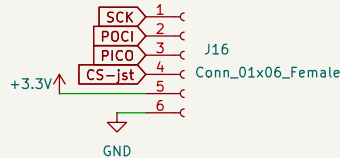


The I2C bus connection is used by the ADE7953, but here are 2 more that add more extensibility to the board as a whole. One of these will may be used by a PN532 RFID Card Reader.

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

## Single SPI Bus

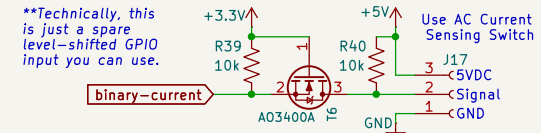
Here is a single SPI connection in a 6-pin JST configuration. You will need to verify the alignment of these pins with whatever you are connecting. Also: be mindful of the new PICO notation versus the legacy MOSI notation for boards.



## 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 flexible 'backup plan' to easily attach an AC current switch or breakout board (with digital out) if you need a fallback.

[!WARNING] Use an AC current sensor here, with digital output (NOT Analog) or this will not work. Try this: M3050 Current Sensing Switch



\*\*Technically, this is just a spare level-shifted GPIO input you can use.

The three-pin connector is there for flexibility. We don't use all 3. But you could if you found some awesome sensor that needed 5v to run and output digital signals.

Use AC Current Sensing Switch

For the M3050, you connect:

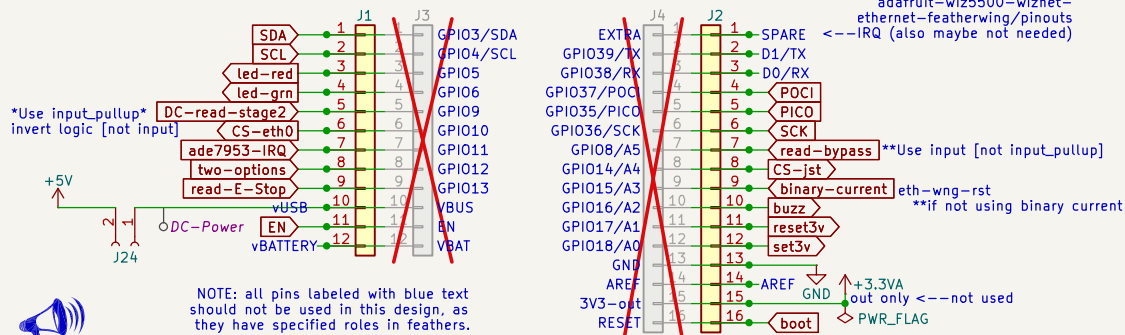
- K2 to J17-Pin\_1 (GND)
- K1 to J17-Pin\_2 (Signal)
- Do not connect J17-Pin\_3

This AC Current Switch simply connects signal to ground and gets powered by the AC current it observes. Invert your logic.

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 :)

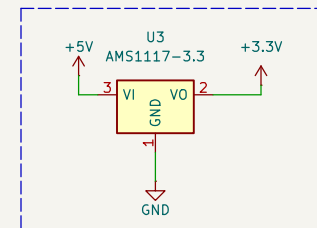


<https://learn.adafruit.com/adafruit-wiz5500-wiznet-ethernet-featherwing/pinouts>

---IRQ (also maybe not needed)

## 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 separate 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: 2025-11-02

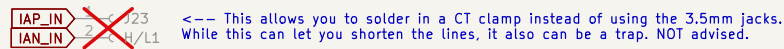
KiCad E.D.A. 9.0.6

Rev: 1.4.3

Id: 5/11

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

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



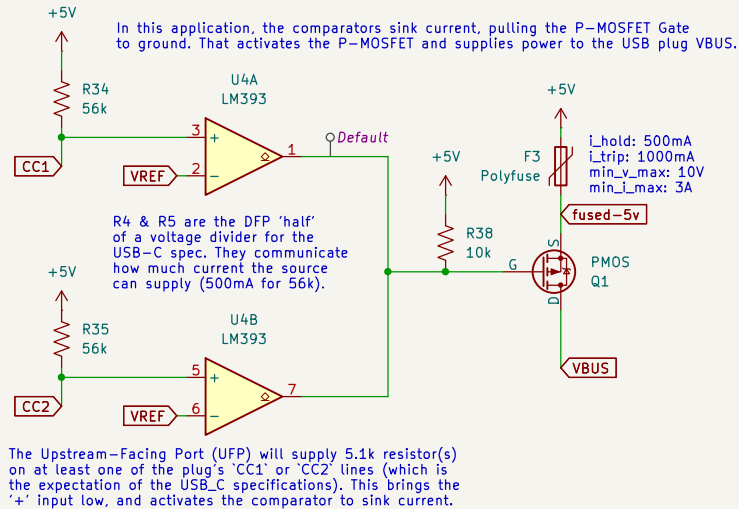
The schematic diagram illustrates the ADE7953 chip and its connections on the ESP32 board. The chip is a yellow rectangle with pins labeled IAP, IAN, IBP, IBN, VN, VP, PULL\_HIGH, PULL\_LOW, IRQ, CLKOUT, and CLKIN. The connections are as follows:

- Power and Ground:** The chip's VDD (pin 17) and GND (pin 4) are connected to the +3.3V and GND rails, respectively. The chip's AGND (pin 16) and EP (pin 29) are also connected to GND.
- Signal Connections:** The chip's SCL (pin 27) and SDA (pin 26) are connected to the SCL and SDA lines, respectively. The chip's IRQ (pin 13) is connected to the IRQ line. The chip's CLKOUT (pin 19) and CLKIN (pin 18) are connected to the CLKOUT and CLKIN lines, respectively.
- Configuration Options:** The chip's IAP (pin 5) and IAN (pin 6) are connected to the IAP and IAN lines, respectively. The chip's IBP (pin 9) and IBN (pin 10) are connected to the IBP and IBN lines, respectively. The chip's VN (pin 11) and VP (pin 12) are connected to the VN and VP lines, respectively.
- Other Components:** The diagram includes various capacitors (C11, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23), resistors (R21, R22, R23, R24, R25), and jumpers (JP12, JP13). The chip is labeled U2 ADE7953xCP.

Id: 6/11

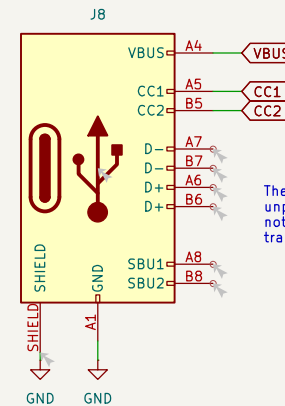
# 5V Daughter Boards: USB-C Source and Pins

While it is not entirely clear how these will be used, they provide options for powering secondary daughter 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 is still guarded by the shared 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 :)

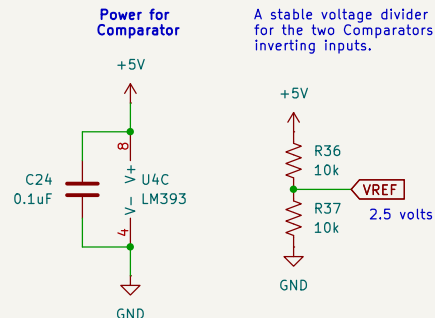
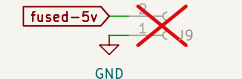


## Power-only USB

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).



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



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.

Bare-minimum USB-C downside facing port  
Features Cold-Plugging with an LM393 and over current protection  
Solderable connection also provided, still guarded by resettable fuse

Corey Rice / MakeHaven

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: 2025-11-02

KiCad E.D.A. 9.0.6

Rev: 1.4.3

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.

Overview of mains-level components

**Corey Rice & MakeHaven**

Sheet: /mainslevelcontrol/

File: mainslevelcontrol.kicad\_sch

**Title:**

Size: USLetter

Date: 2025-11-02

Rev: 1.4.3

KiCad E.D.A. 9.0.6

Id: 8/11



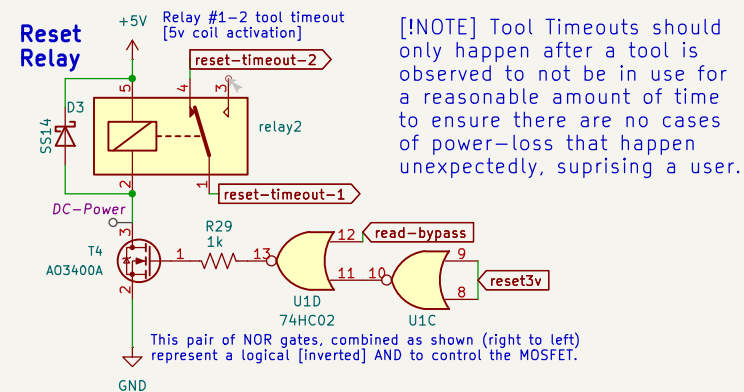
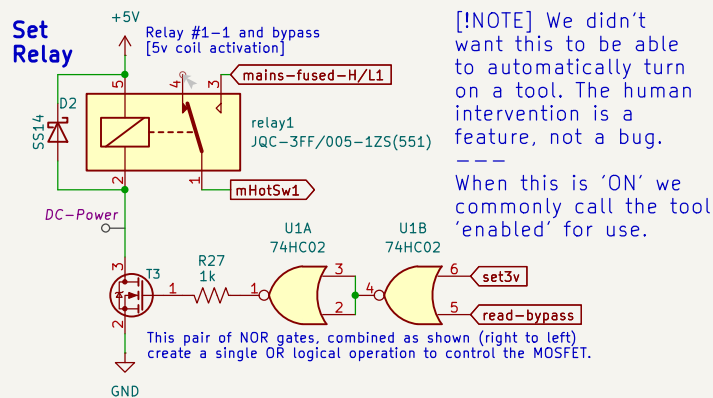
# Stage 1 || Control from 5V & 'set' or 'reset' Stage 2 Latch

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

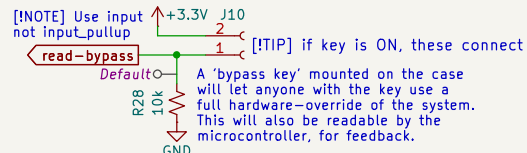
This stage is responsible for allowing the Feather to control the actions of the mains level of power switching.

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'

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'



## Physical 'Bypass Key' to lock-ON tool



## 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.

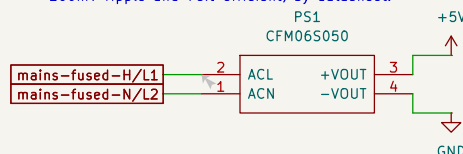
When 'read-bypass' is ON this turns ON the 'set' relay, but turns OFF the 'reset' relay. By DeMorgan's Laws, that gives us the connections shown above. Note: the final NOT of the 'reset' relay logic is in the relay itself.

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

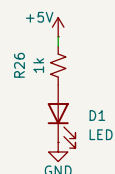
'set' relay: (ON if 'set3v' is HIGH) OR (ON if read-bypass is HIGH [B-key is ON])  
'reset' relay: ON if read-bypass is LOW [B-key is OFF] AND reset3v is HIGH  
--> else: 'reset' relay is OFF [B-key can block 'reset' relay]

## 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: 2025-11-02  
KiCad E.D.A. 9.0.6

Rev: 1.4.3  
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>

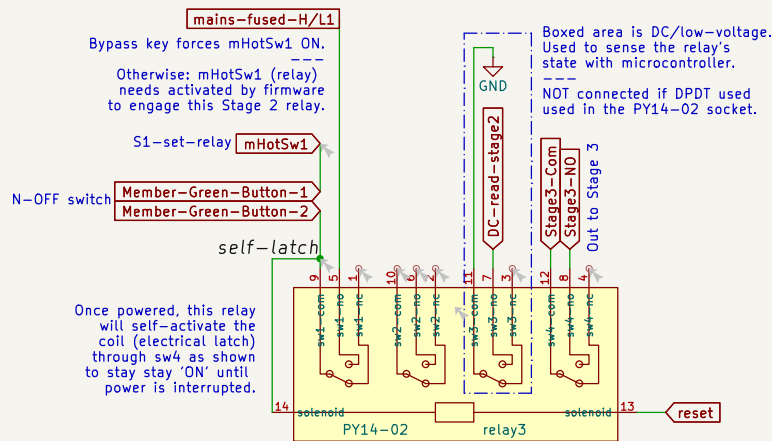
There are multiple variants of Omron's MY series relays, which allow for various coil voltage levels.  
----> And this footprint also works with Omron's G2A relays, if there is ever a parts shortage...

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).

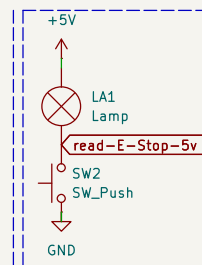
### 'Set' & 'Latch' circuit on Stage 2 relay

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.



### 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.

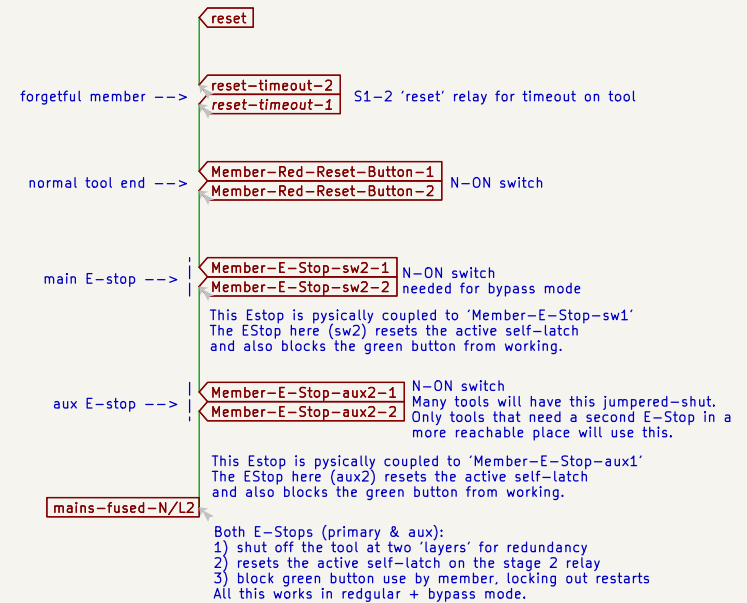
\*If LED lamp is used, put resistor ABOVE the 'read' location.

### '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

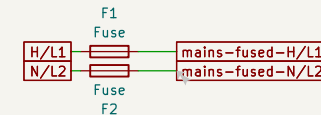
The connections needed for the above, are shown below:



### 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: 2025-11-02

KiCad E.D.A. 9.0.6

Rev: 1.4.3

Id: 10/11

# Stage 3 || Final Contactor & Screw Terminals

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

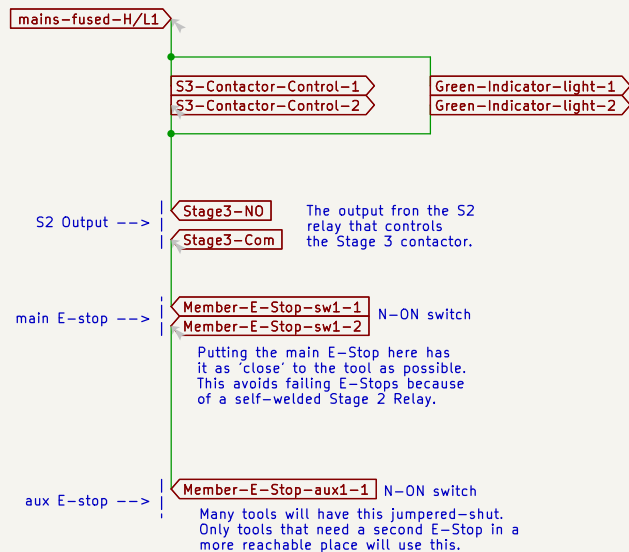
## Control for Final Contactor

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 pass through this PCB.

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

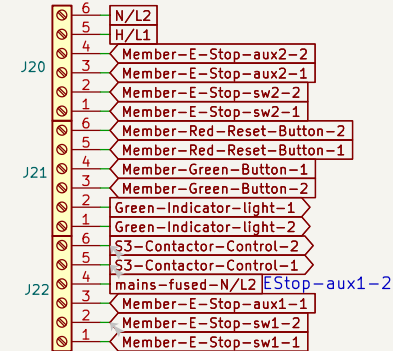
The E-Stop switches shown here can cut power to the contactor in the unlikely event that their twin switch (sw2 | each E-Stop has 2 switches) is fused shut, and the stage 2 relay stays powered. A backup to ensure the E-Stop is effective.



## 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: 2025-11-02

KiCad E.D.A. 9.0.6

**Rev: 1.4.3**

Id: 11/11