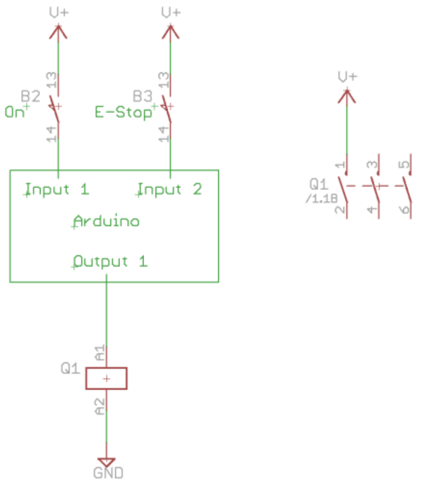
Emergency-Stop and You

You often see one ore multiple Emergency-Stops mounted to a simulator. Their purpose is to stop a dangerous motion. This can be your Simulator moving, or your Direct-Drive Wheel going crazy. Every Motion Simulator should have an Emergency-Stop mounted in reach of the user!
But often people underestimate the danger and do not implement an Emergency-Stop, or implementing it wrong.
In this text I would like to clear up the concepts of an E-Stop.

But first a disclaimer:
The material and information contained in this post is for general information purposes only. You should not rely upon material or information in this post as a basis for making any decisions. I make no representations or warranties of any kind, express or implied about the completeness, accuracy, reliability, suitability or availability with respect to the post or the information, or related graphics contained in the post for any purpose. Any reliance you place on such material is therefore strictly at your own risk.

An E-Stop has to stop a dangerous situation, put a machine into a safe state. This is not always as easy as "Turn the Power off". In some situations, you have to bring a motion to a safe stop before you can cut the power. An example would be a car. You have to brake first, before you cut the power.
In our hobby, it should be enough to simply cut the power to the motors. Best practice is to cut Life and Neutral (Never ever switch the Protective Earth!).
But how do you do this reliable? Let's look at some examples.

We live in a modern world! With cheap microcontrollers and all that stuff.



This solution will work, but **__is not safe!__**

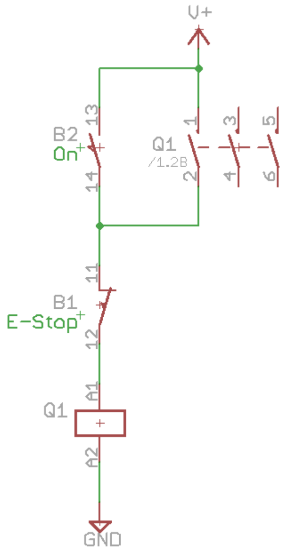
- The software has to take care that B3 will always overwrite B2.
- If the wire from B3 breaks, your E-Stop is useless!
- If the controller crashes, your E-Stop is useless!

Even if you replace B3 with a NC (normally closed) contact, you still face the possibility of a software crash. Things normally go crazy when software behaves bad. If this same software that is causing your issue now has to release Q1… You see where this is going.
The industry standard is two parallel working units which compare their results at the end. If they mismatch, a safe-state is triggered (again, with redundancy). For a hobbyist this is not the way to go.

Stay away from DIY E-Stop systems with microcontrollers!

The microcontroller can always only be an addition!

One easy solution could look like this:



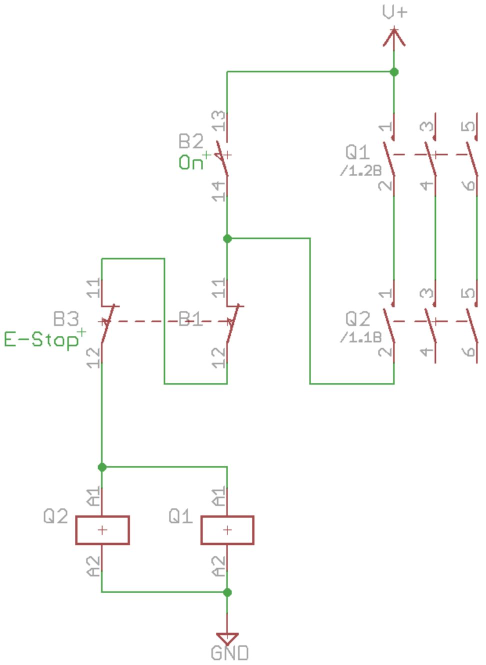
If you press B2 “On”, Q1 will engage. All contacts from Q1 will close. If you now let go of B2, Q1 will power itself. This is a simple latching circuit.
To unlatch it, press B1. Now Q1 will lose power and all contacts will open, cutting the power to your Simulator.

It’s a simple circuit, but it already covers some important points:

- Even if you hold down B2, the power will be cut.
- If the wire to B1 breaks, Q1 will return to its safe state.

This is way better than the microcontroller solution, but there is room for more.

Let's try to do better:



In this version a second safety layer got added.
Note: B1 and B3 are the same switch, they are mechanicaly connected.

- If a contact from Q1 or Q2 goes bad, you still have a second one that works.
- If B1 or B3 goes bad, you still have a second contact that works.

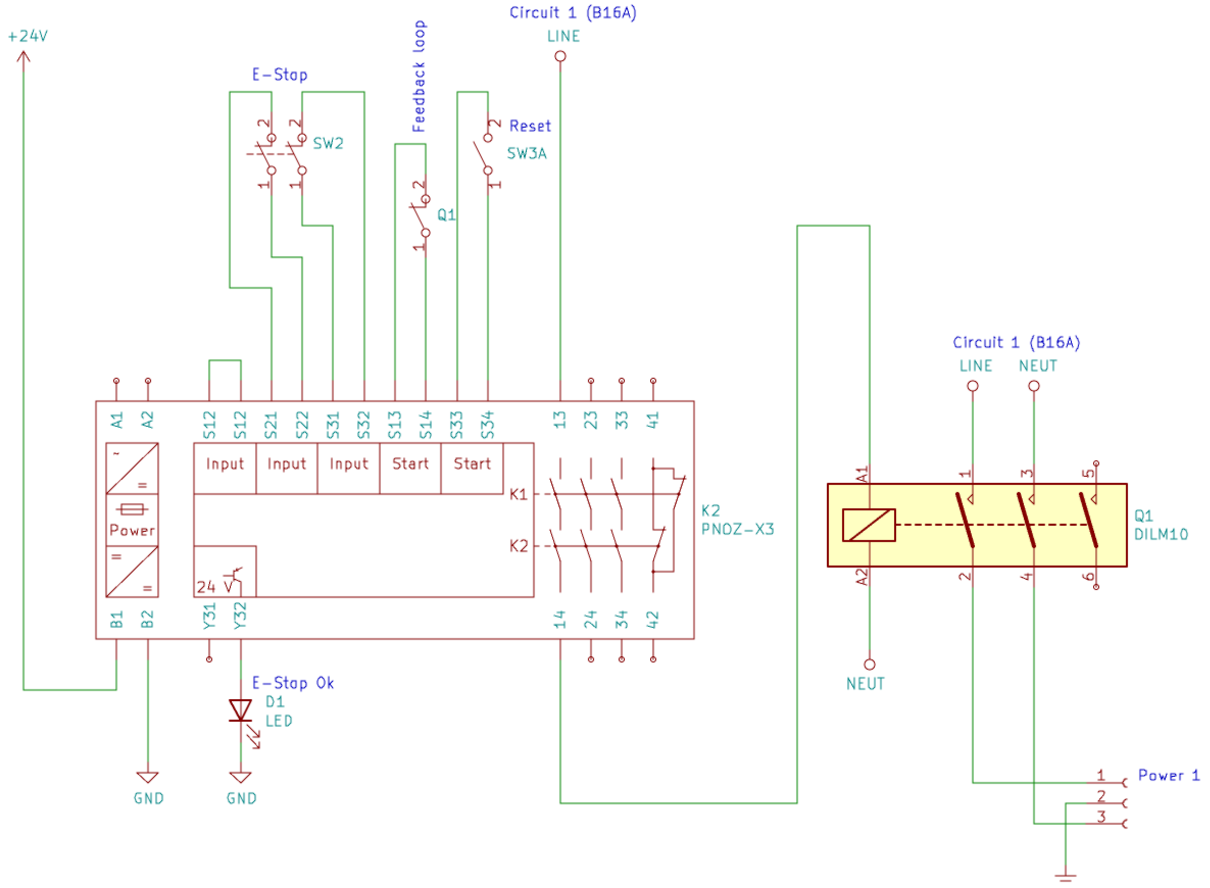
Looks good so far. But there are still things that can go wrong:

- How to note that a contact from Q1 or Q2 went bad?
- What if the wire to B1 and B3 is cut open on the metal frame and shorten?

Instead of writing down a bunch of other solutions, I jump directly to the professional solution.

For the next step, you need to know one term:

What does force guided contacts mean for Contactors and Relays?
Mechanical guiding (forced guiding) makes it impossible to close the normally closed and normally open contact simultaneously. If a normally closed contact becomes welded, it must be impossible for the normally open contacts to close when the coil is energized.
If a normally open contact becomes welded, it must be impossible for the normally closed contacts to close when the coil is de-energized.



Wow, there is a lot going on here! But stay with me a bit longer!
This is a professional E-Stop relay PnozX3.2 from Pilz. It covers all possible errors that can come up. As you can see, K1 and K2 are in series like we had Q1 and Q2 before. Everything else is a block box (for now).

Let us go over the schematic:
SW2 is the E-Stop. As in our drawing before, both contacts are mechanical connected. But instead of only two wires, we run four wires. Additional E-Stops would simply be wired in series.
SW3A is the Reset button to reset the E-Stop.
Q1 is a force guided auxiliary contact from the Contactor Q1.
D1 is a LED to simply signal that everything is ok.

You can easily add more contactors besides Q1. Just wire their force guided auxiliary contacts in series between S13 and S14.

The normal default state is as follows:

- E-Stop is not pressed
- Q1 is powered off

Enable the circuit:

1. Press SW3A
2. K1 and K2 will energize
3. Q1 will energize

The E-Stop relay covers the following fault conditions:

- If at least one contact from SW2 is interrupted, the E-Stop will trigger

- If the wires to SW2 are shorted, the E-Stop will trigger
- If the wires to SW2 are wired crossover, the E-Stop will trigger
- If the resistance from SW2 and its wires is greater than 300 Ohm, the E-Stop will trigger
- If the force guided auxiliary contact from Q1 is not closed, the E-Stop will not reset (i.e. because of welded contacts in Q1)
- You have to let go of SW3A before you can reset the E-Stop after it tripped.

You can find these relays relatively cheap on eBay. There are many other manufactures, like Siemens, Rockwell, Phoenix Contact, Eaton and so on.
They are all a bit different, but they cover the same basics. Downside is, you have to look into the manual to find out how to wire your exact model of safety relay.