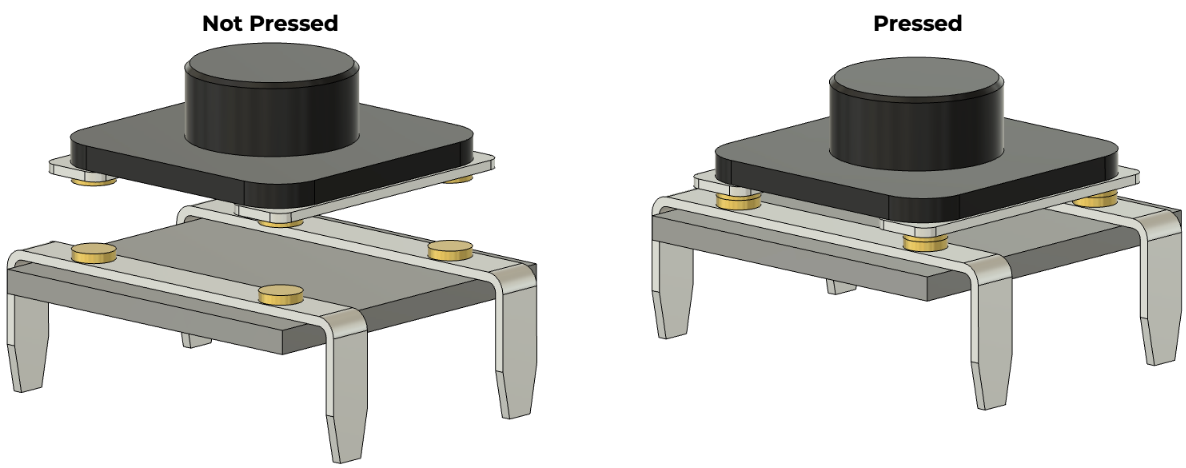
PIPER

What are Pull-Up and Pull-Down Resistors?

How pushbuttons work

Before we talk about Pull-Up and Pull-Down Resistors, it's important to know how a pushbutton works.

A pushbutton is a simple device that makes an electrical connection. Here's a diagram that shows what is happening inside of a pushbutton when you press the button:

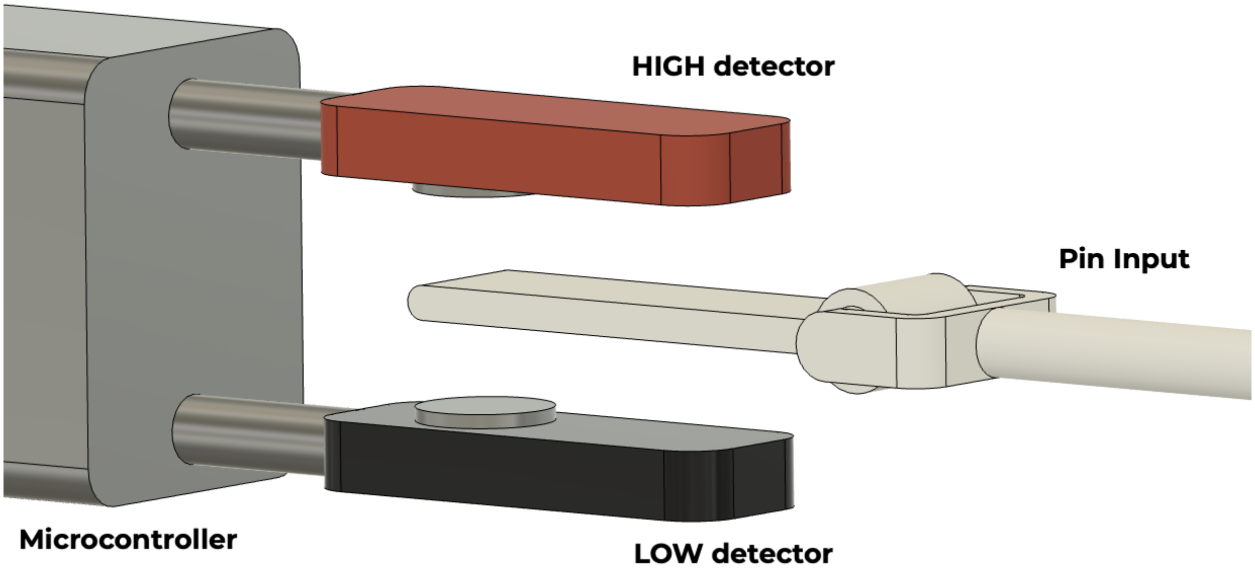


When the button is not pressed there is not an electrical connection between the pins on the left and the pins on the right side of the button. When the button is pressed, all the pins are connected electrically.

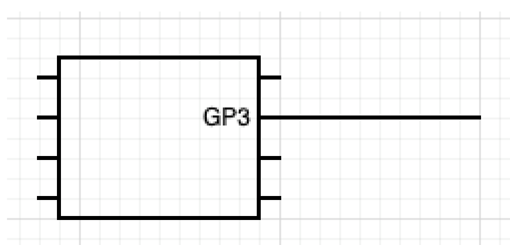
How microcontroller digital inputs work

In a microcontroller like the Raspberry Pi Pico, the Pins on the microcontroller can be used as **inputs**. This means that they can detect whether the voltage of whatever is connected to the pin is LOW (approximately 0 Volts or Ground) or HIGH (approximately 3.3 Volts).

Below is a graphic that we will use to help visualize what is going on with the electricity in a microcontroller pin when it is configured to be an input. This is not what it looks like – instead, this graphic helps explain what is happening inside the microcontroller's pin.

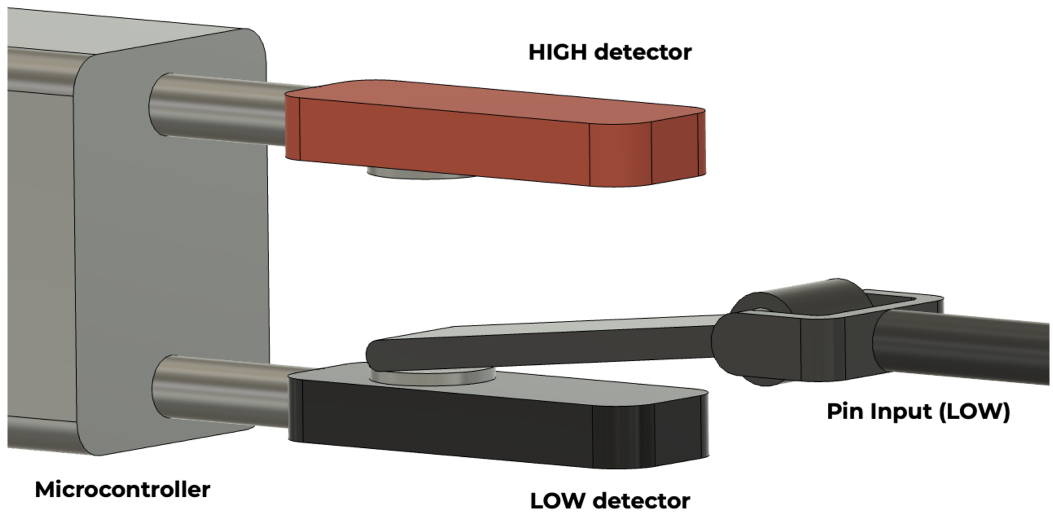


Here's a schematic diagram of what it looks like then nothing is connected to the input pin:

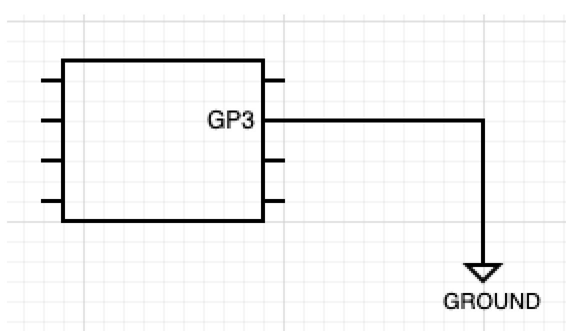


When nothing is connected to the Pin's input, we call that "floating". It's not clear what the microcontroller should detect, so it often flickers between LOW and HIGH. These flickering readings can really mess with your program – more on that in a bit!

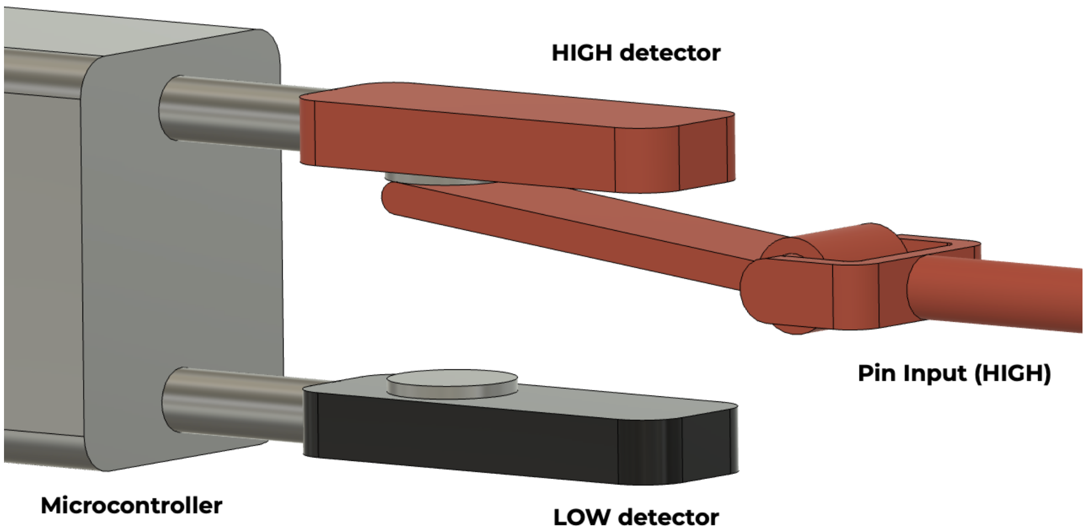
If you connect the Pin's input to Ground (0 Volts), it will trigger the LOW detector in the microcontroller, and the microcontroller reads that as LOW:



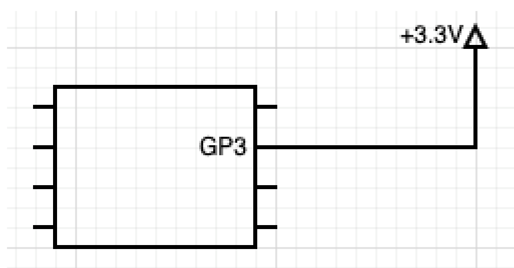
Here's a schematic diagram showing the input pin is connected to Ground (0 Volts):



And if you connect the Pin's input to 3.3 Volts, it will trigger the HIGH detector in the microcontroller, and the microcontroller reads that as HIGH:



Here's a schematic diagram showing the input pin is connected to 3.3 Volts:

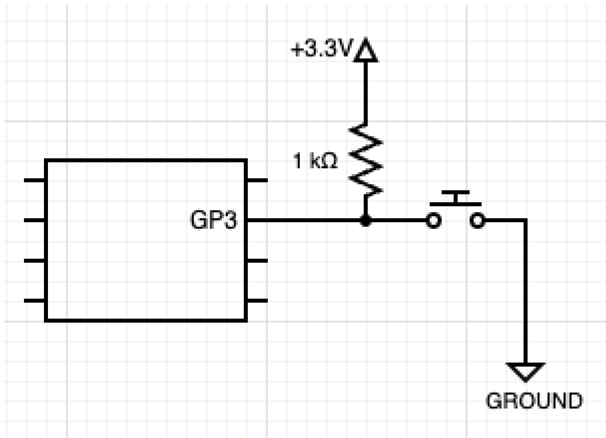


Connecting a Pushbutton to an Input

When you connect a pushbutton to a microcontroller's input and the button is not pressed, nothing is connected, and that pin is **floating**. The problem with floating pins is that they can flicker between high and low, which can create all kinds of problems in your code.

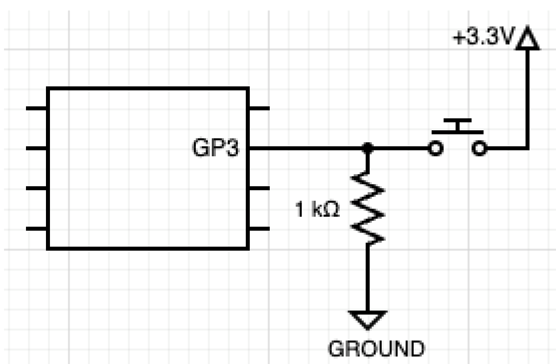
The good news is that there is a solution to this problem! We can use a resistor to "pull" on the pin, which will hold it in one position while the microcontroller waits for the button to set it to the other position.

Here's a schematic showing a pushbutton that is connected to Ground (0 Volts) and the microcontroller's input connected to both the button and a resistor that is connected to 3.3 Volts:



In the schematic above, when the button is NOT pressed, the only thing connected to the microcontroller's input is a resistor that is connected to 3.3V. This **pulls** the input HIGH, but the pull on the input pin is *weak* - the resistor prevents most electrical current from flowing through it. This means that when the button is pressed, the pin is now also connected directly to Ground (0 Volts). That connection is *strong*, which overcomes the pull from the resistor and pulls the input LOW. It's important that resistors used to pull have a lot of resistance – these resistors are often 10000 or more Ohms.

You can also flip this arrangement. You can use a *pull-down* resistor and connect the pushbutton to 3.3V:

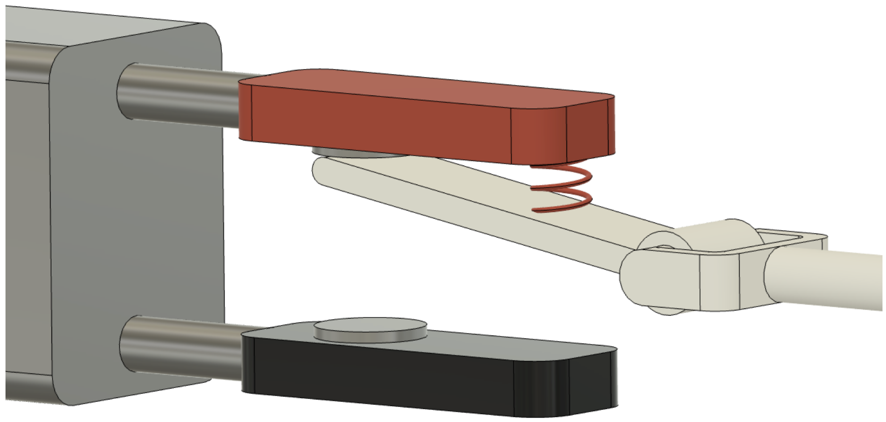


Internal Pull-up and Pull-down resistors

A cool feature of most modern microcontrollers is that they often have *internal* pull-up and pull-down resistors. This means that you don't need to connect a resistor to your circuit, you can simply program your microcontroller to use the resistor it already has!

These resistors work the same way, kind of like little springs in our model:

If you tell the microcontroller to use a pull-up resistor, it keeps the input HIGH when nothing else is connected:



But when you connect the input to ground by pressing your pushbutton, the connection pulls harder to Ground (0 Volts), overcoming the internal pull-up: