A screen shot of a graph

Description automatically generated

The picture above was from the current sensing, back when it was working correctly. The blue line is a binary indicator of if the current threshold (0.1 V difference at the time) is exceeded, and the orange line is a counter related to how far the motor position has rotated from start. There is a brief period at the beginning of starting the motor in the forward direction that the current threshold will be exceeded (about less than 200 counts), then the current settles and can be properly thresholded. The code should be configured to exclude this brief initial period before telling the motor to stop based on threshold. This is because the motor has to accelerate initially, and therefore requires more current.

As of 12/6/23, the current sensing was not working. This is diagnosed as being because of the ESP32 improperly reading the voltage at the current sensing pins. The voltages can be manually read with a DMM. With the DMM and when the ESP32 was working properly, the voltage difference at the pins over the current sensor should be about 0.09V with no resistance applied to the motor and the motor running steady. Don’t take that value for exact. Regardless, it should be relatively steady when the motor is running steadily. Currently, the voltages measured by the ESP32 vary wildly at each of the nodes of the current sensor. We need a reliable way to measure this difference. Possible solutions are to verify that the intended ESP32 pins for current sensing are properly measuring voltages, or maybe implementing a voltage differential op amp to minimize the work that the ESP32 must do.

Other problem to possibly explore: ensure that the 3.3V pins on the ESP32 are outputting DC voltage as opposed to AC. I briefly hooked PWM to a 3V3 pin and it went weird. Could have just been connecting something I shouldn’t have. Also, why does the motor make a high pitched noise when it’s just connected and not moving? If PWM is connected to a rail, it doesn’t make this noise.

Negating the current sensing, the EMG input was able to allow the motor to toggle between two arbitrary positions, as seen in Discord on 12/9/23.

The circuit was attached as shown below:

A circuit board with wires

Description automatically generatedA circuit board with wires

Description automatically generated

A circuit board with wires

Description automatically generated

Future optimization should be focused around:

* Ensuring current sensing works (notes earlier in this document)
* Making sure the motor can move fast enough between the proper beginning and end positions
* Reducing jitter around motor stop positions
* Ensure pins used are not the ones listed as bad from testing (see document in this folder)
* Ensure everything was connected back as pictured above (I disconnected everything at the end of the session 12/9/23 so there’s a chance I didn’t put everything back exactly correctly)
* Current input code in the merge file is not what was originally in V9 input code (the nice optimized stuff), and V9 was saved over, so check out the input code again and may have to redo that
* Soldering when finalized