

# Tachometer

MANUAL

SDMAY19-30

Client: Matt Post

Adviser: Matt Post

Team Members/Roles:

Jessica Bader - Chief Software Engineer / Communication Manager / Scribe

Meghna Chandrasekaran - Chief Software Engineer / Meeting Facilitator

Katrina Choong - Chief Hardware Engineer / Timeline Manager

Seth Noel - Chief Hardware Engineer

Kyle Zelnio - Project Manager

[sdmay19-30@iastate.edu](mailto:sdmay19-30@iastate.edu)

[Sdmay19-30.sd.ece.iastate.edu](http://Sdmay19-30.sd.ece.iastate.edu)

Revised: EDIT DATE - Version 1

# Table of Contents

<b>1. General User Guide</b>	<b>2</b>
How Open the Program	2
How to Start the Tachometer	2
How to Stop the Tachometer	2
How to Change the Communication Port	2
How to Close the Program	2
How to Tell What the RPM is	2
<b>2. Setup User Guide</b>	<b>3</b>
Required Parts	3
Mounting the Hall Effect Sensor	4
Testing the Hall Effect sensor	6
Setting up the Microcontroller	6
Setting up the Software	6
<b>3. Design</b>	<b>7</b>
3.1 Interface Specifications	7
GUI to Microcontroller	7
Microcontroller to Hardware circuit	7
3.2 Component Design	7

# 1. General User Guide

## HOW OPEN THE PROGRAM

Go to C:\EE448\Speed Control GUI\dist\gui\ and click to open gui.exe. A command window will pop up, which can be ignored. After several seconds, the GUI will pop up.

## HOW TO START THE TACHOMETER

Once the GUI has been opened, if calculations are not being made then the user can initiate RPM calculations by pressing the large, green “Start” button. The measurements will appear above the START and STOP buttons.

## HOW TO STOP THE TACHOMETER

Once the GUI has been opened, if calculations are being made then the user can pause new RPM calculations by pressing the large, red “STOP” button. The measurement on the screen will freeze where it is.

## HOW TO CHANGE THE COMMUNICATION PORT

To receive RPM calculations, the user must provide the valid communication port already chosen by the computer to communicate with the microcontroller. If the com port is not valid, a notice in red writing will appear on the bottom left of the GUI to request a valid com port. To find the correct com port, the user can go to *Device Manager*, open the tab labeled *Ports (COM & LPT)*, and choose the port listed. They will input the new port name in the input box next to the red writing and press the submit button below. The warning will disappear when the com port is valid and pressing the green “Start” button will cause the RPM value at the top to update.

Example:

COM5 would be entered as *com5*.

## HOW TO CLOSE THE PROGRAM

The user can close the program by pressing the X at the top right corner. This will close the communication port and exit the program.

## HOW TO TELL WHAT THE RPM IS

The most recent RPM value will be displayed in the middle at the top of the GUI. This will be updated every one second while the microcontroller is in ‘start’ mode. The RPM calculation is an average of the last ten values. If speed is changed, it may take up to ten seconds before the new value is valid. The tachometer can output calculations in steps of 6RPM; if it is oscillating frequently between two values, the true reading can be approximated as halfway between the two (the user may use their discretion as to whether to choose a specific value or halfway between two values). When the rotational speed of the motor changes, the user must wait until the tachometer picks a single value or oscillates between two values to read the RPM.

**NOTE: outside influences on the system (such as using a stroboscope) may influence the system. If the tachometer starts changing values across large ranges when the user knows no rotational acceleration has taken place, the measurements must be assumed to be invalid. Immediately remove any nearby electronics or other potential sources of EM. If this does not solve the problem, contact someone to search for the source of error.**

## 2. Setup User Guide

### REQUIRED PARTS

- Mount
- Filter circuit
- Arduino
- USB A/B cord
- Desktop computer
- Executable file
- 4mm Allen wrench
- Phillips screwdriver
- 7/16 inch wrench
- 3/8 inch wrench

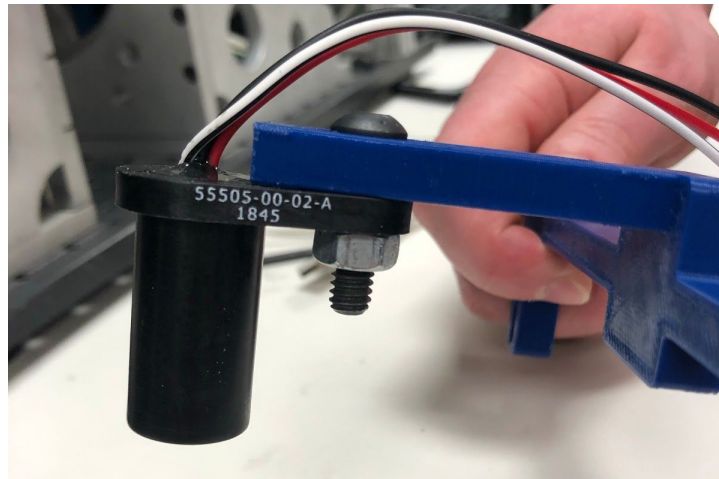
## MOUNTING THE HALL EFFECT SENSOR

Equipment:

- 8 screws
- 4 washers
- 1 3-hole plate
- 1 single-arm mount (pictured in blue)
- 1 double-arm mount (pictured in black)
- Hardware circuit board

The goal is here to mount the Hall Effect sensor about 3mm directly above the key on the motor shaft.

Step 1: First connect the Hall Effect sensor to the bottom of the single-arm mount with a screw above and a nut below (loosely for now). See *Figure 2.1*.



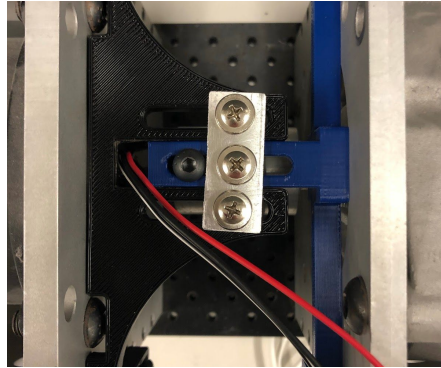
*Figure 2.1: Attaching the sensor to the single-arm mount*

Step 2: The single-arm mount should be screwed into the generator faceplate with two screws.

Step 3: Adjust the Hall Effect sensor along the single arm mount to place it above the key. Adjust the single-arm mount up and down on the face generator plate until the Hall Effect sensor is 3mm above the key.

Step 4: Attach the double-arm mount to the motor side with two screws.

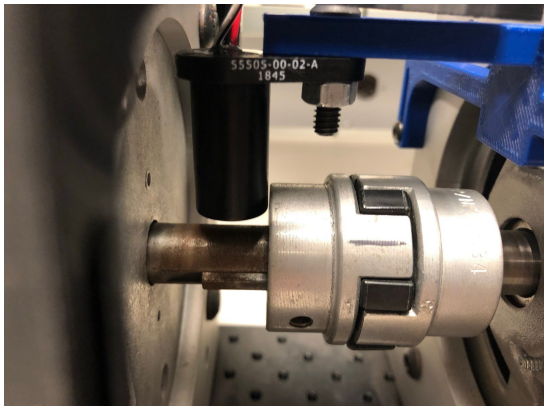
Step 5: Align the 3-hole plate across the three arms from the mounts. Use 3 screws to attach the plate. See *Figure 2.2*.



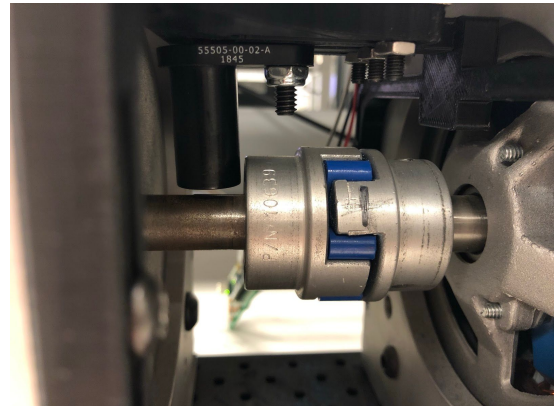
*Figure 2.2*

Step 6: Once correctly placed, all screws should be secured tightly to prevent wobbling of the sensor when the motor vibrates. Make sure the hall effect sensor is centered over the shaft. See *Figure 2.1: Mount and sensor setup*.

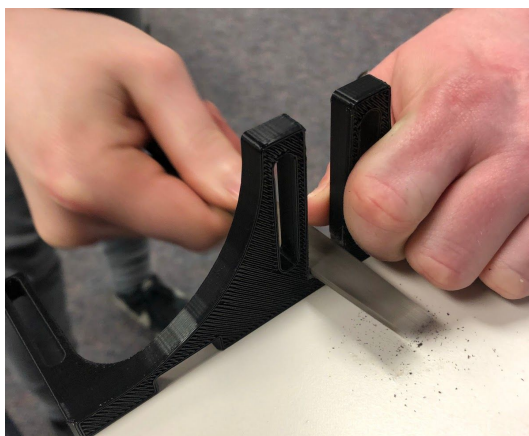
Step 7: If the sensor is touching/almost touching the shaft (see *Figure 2.3*), the double-arm mount may need to be filed down (see *Figure 2.4*). See *Figure 2.5* for where to file.



*Figure 2.3: sensor too close to shaft*

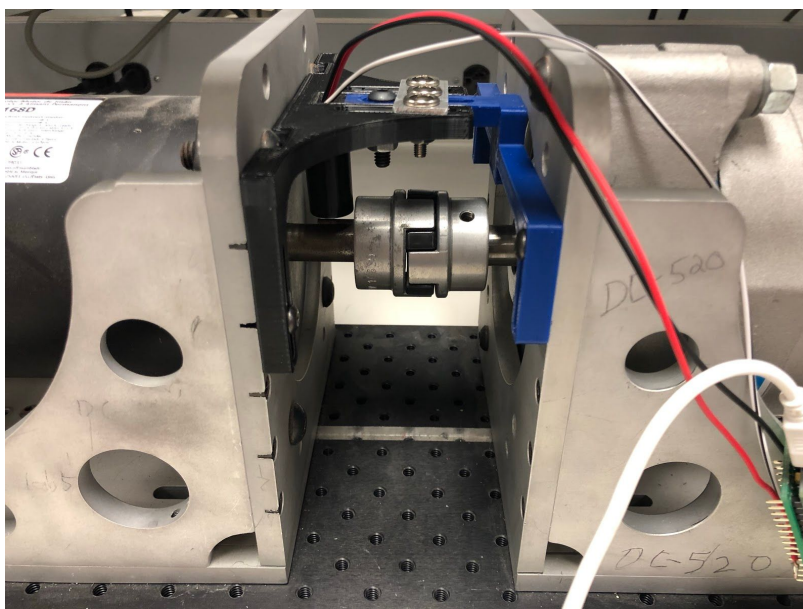


*Figure 2.4: sensor properly placed*



*Figure 2.5: Where to file the double-arm mount if the sensor is too close to the shaft*

The finished mount can be seen in *Figure 2.6*.



*Figure 2.6: Finished Mount*

### TESTING THE HALL EFFECT SENSOR

(previous part) The Hall Effect sensor should already sit directly above the key on the motor shaft. It should be about 3mm above the key. See *Figure 2.3*.

Step 1: Once the sensor is mounted, set up the oscilloscope to read from the D2 pin (K node). (Probing the D2 pin side of the capacitor is recommended).

Step 2: Ensure that the output of the sensor is clear. Move the sensor closer to the shaft (but not touching) if it is not clear. See *Figure 2.7: Sensor output* for an image of a clear output.

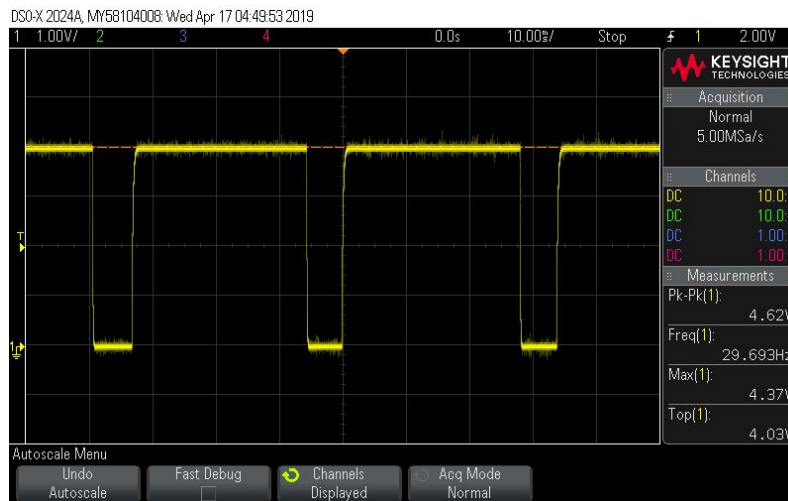


Figure 2.7: Sensor output

## SETTING UP THE MICROCONTROLLER

The microcontroller must be connected to the desktop with a USB A/B cord through a valid communication port.

## SETTING UP THE SOFTWARE

The executable GUI file must be on the computer, along with any other files required to run it (see *insert executable location here*). This should be put in C:\EE448\Speed Control GUI The desktop should be connected to the microcontroller through a USB A/B cord.

# 3. Design

## 3.1 INTERFACE SPECIFICATIONS

### GUI to Microcontroller

The GUI will need to be set up by the user to use the communication port already assigned by the computer. Instructions for this can be found in *1. General User Guide: How to Change the Communication Port*. The GUI is in charge of opening and closing the com port. It will open the serial port with the com port when a new port is submitted by the user and will close it upon exiting the GUI.

The GUI can give start and stop commands to the microcontroller, to start/stop the calculations (see *1. General User Guide: How to Start the Tachometer* and *How to Stop the*



*Tachometer*). The GUI will send a *1* to the microcontroller at any point to command a start and a *0* to command a stop.

When in start mode, the microcontroller will output the most up to date measurement every 1 second for the GUI to display directly. When in stop mode, nothing will be output from the microcontroller.

### Microcontroller to Hardware circuit

The hardware circuit should output a pulse. The speed of the pulse should match the rotational speed of the motor. At this point, the microcontroller uses an interrupt to count how many pulses it reads each second. That is then converted into an RPM measurement. The pulse output must be clear and consistent when the motor is not changing speed. See *Figure 2.4*.

## 3.2 COMPONENT DESIGN