THE UNIVERSITY OF TEXAS AT ARLINGTON

COMPUTER SCIENCE AND ENGINEERING

MECHATRONICS LAB 4 REPORT

ELECTROMECHANICAL SYSTEMS & SENSORS

Submitted toward the partial completion of the requirements for CSE 5355-001

Submitted by,

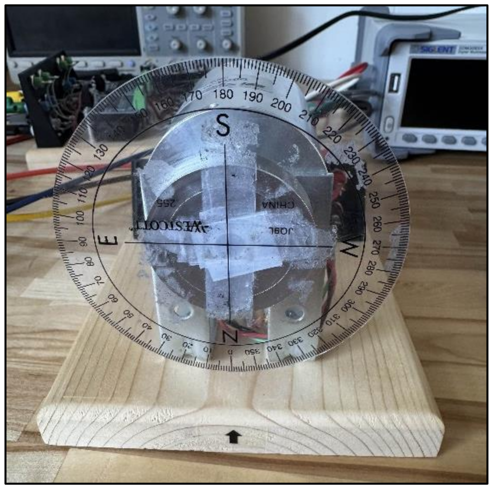
Jennifer Hernandez &

Servando Olvera

Date 10/20/2024

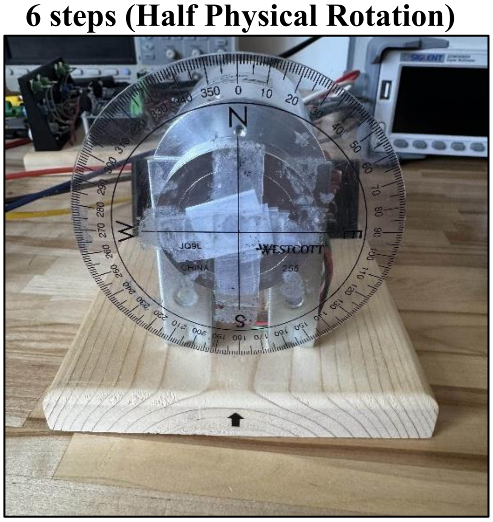
1. For this lab, we will use a voltage of 6V on the bench motor power supply unless otherwise noted. The 0V output will serve as a ground reference for the circuits and the oscilloscope leads throughout the lab. The BLDC motor has a coil resistance of 0.13 ohms, but the 10A power supply will limit current to the motor. The delta-wired BLDC A-B-C windings are on the RED-BLACK-GREEN banana jacks. A virtual common is created by summing voltages from all phases through 100k ohm resistors and is available on the WHITE jack.

2. First, energize one of the coil pairs. Attach a compass protractor to the belt pully on the motor, so that 0 degrees lines up with the arrow on the base.

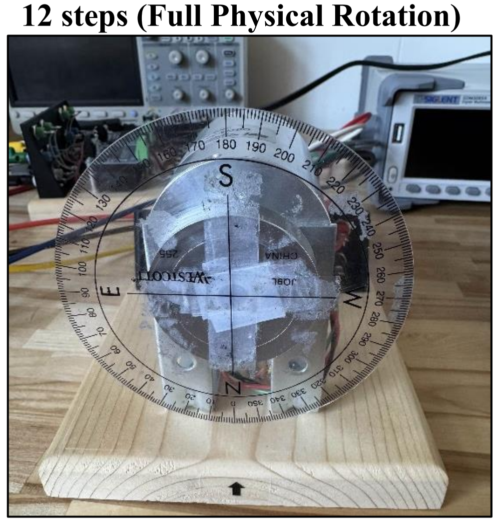


3. Next, energize the coils through the 6 steps to complete one electrical rotation. What is the physical rotation of the disc? Continue until one complete physical rotation is complete.

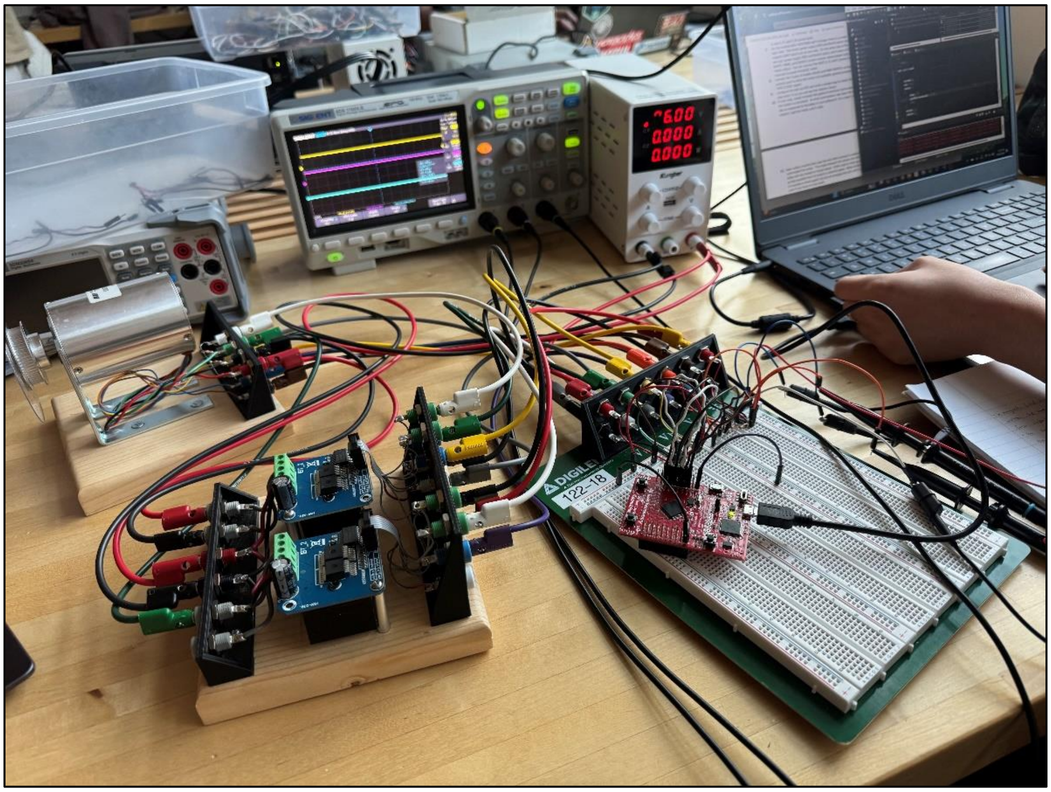
6 steps (Half Physical Rotation)



12 steps (Full Physical Rotation)



4. Power the hall effect sensors with 5V on the YELLOW jack and 0V on GREY jack. The outputs of the A1212E hall sensors 1-3 are on the BROWN-BLUE-ORANGE jacks. Since the outputs are open drain, connect the outputs through 10kohm resistors to 5V.

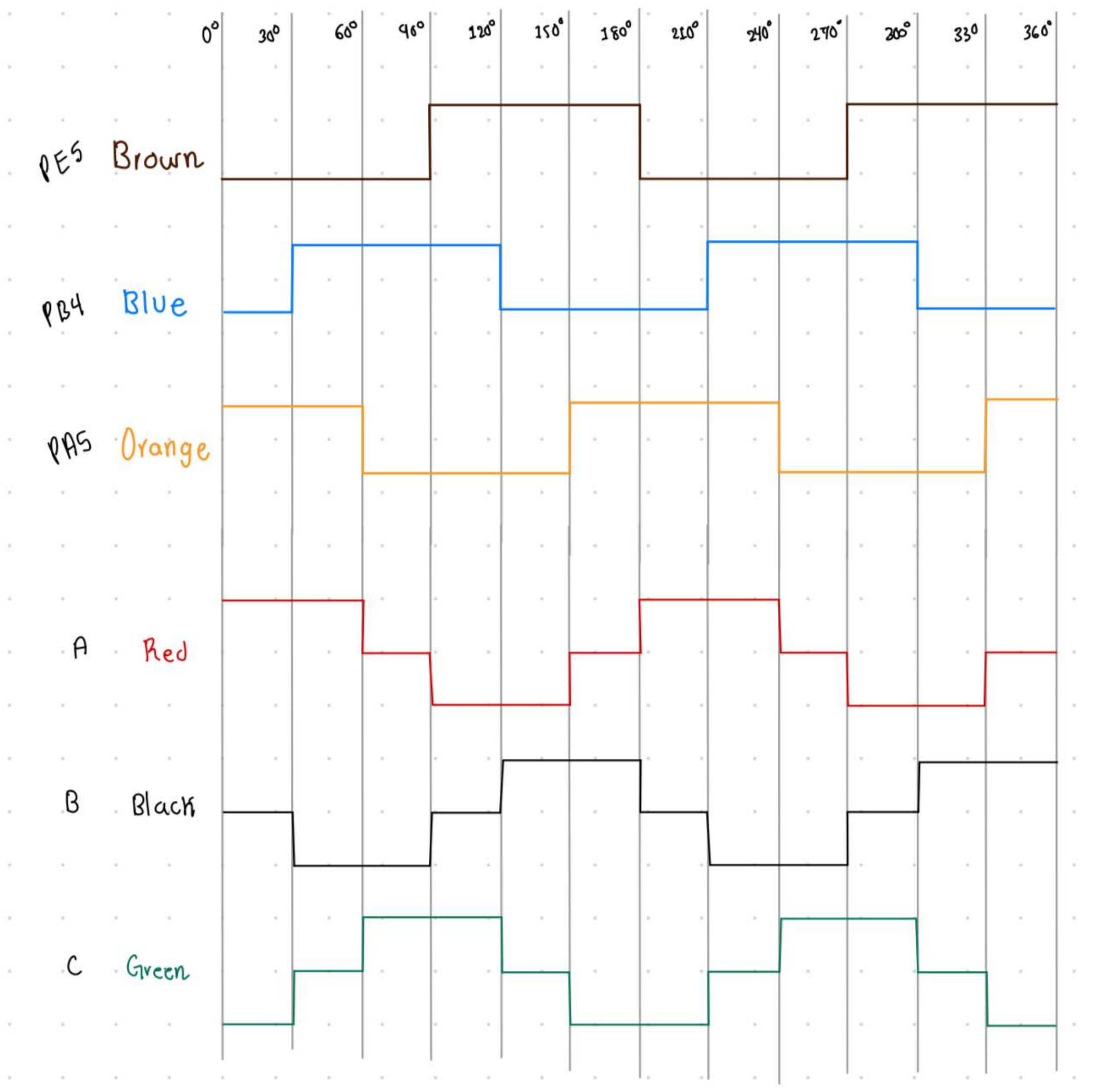


5. Slowly rotate the disc and note the angle at which the hall effect sensors turn on and off. Note the hall effect sensor outputs vs physical rotation relative to 0 degrees as marked in step 2.

Every 90˚ degrees the hall effect sensors would turn off and on.

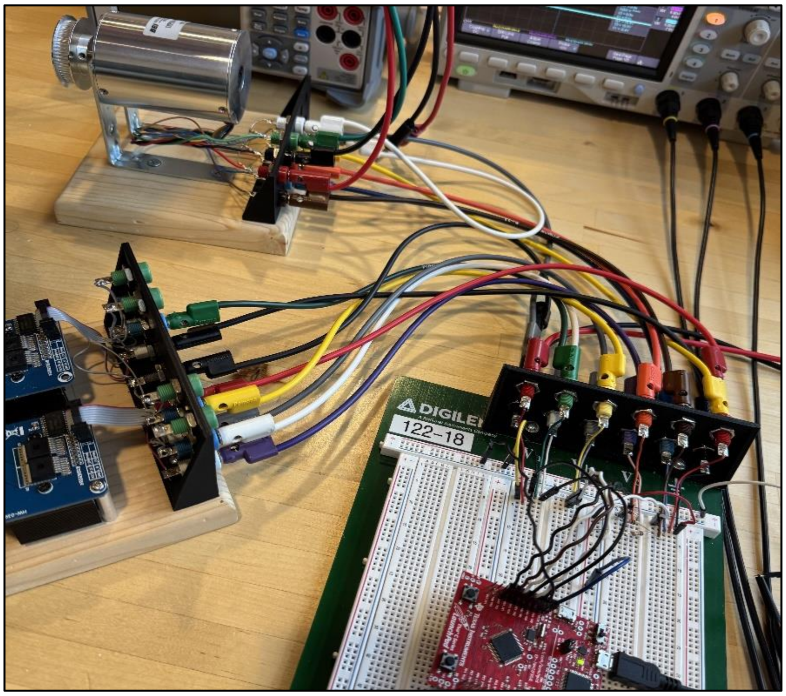
6 steps equate to 180˚ physical.

6. Now, plot the hall effect sensor outputs and their relationship to the disc angle when each of the coils was energized. You should have a plot of sensors 1, 2, 3, and A, B, and C (6 curves total).



7. On the module with 4 BTS7960 half H-bridges there is a motor power and a logic side. The motor power supply (RED and BLACK jacks) and 4 motor outputs (GREY jacks) are on one side. On the other side of the module, there are logic power supply (RED and BLACK jacks), and 4 pairs of BLUE (enable), WHITE (PWM), and GREEN (current resistor sense output) jacks, one for each motor output. You can connect the BLDC A, B, and C phases to any 3 of the 4 channels to power the BLDC motor.

8. Connect the 3 pairs of enable (BLUE) and PWM (WHITE) corresponding to the 4 outputs used to drive the motor to 6 available general-purpose outputs on the TM4C123GXL board.



9. Configure the pins as GPO and write a simple loop to rotate the motor by one step every 1s so you can see the rotation clearly.

Each coil of the motor had an enable and direction GPIO output pin which powered the motor's internal magnets and enabled commutation.

The code was designed to call a step function and wait an amount of microseconds after doing so. The step function has 6 hard-code steps available to be called according to the chart seen on part 6. Each step sets the direction outputs as necessary before setting the enable outputs. This ensures motor will be powered correctly during each step.

After each step, waitMicrosecond ensures the coils have had time to move the motor accordingly. The value used in this function is slowly increased after each step until the motor slips. This slip value was found empirically.

10. In your loop, slowly increase the speed of electrical commutation and ensure the motion is consistent. At what commutation rate does the motor start to slip as measured by the delay in your software?) At a lower consistent rate of rotation, load the motor until it slips.

The motor began slipping at a communication rate of about 2,700 ms.

11. Now, write a routine that uses the hall effect sensors to determine when to commutate the motor. The angle between the sensor and coil is fixed, but the delay will vary based on target speed. Write code to ramp up the motor speed as fast possible from full stop to a desired speed. When at the fixed speed, try to maintain a fixed speed, but do not energize the coil until it is time to do so, even if you have to slow down.

Three GPIO inputs were configured to take in readings from the motor's hall effect sensors. The motor's position can be calculated by continuously checking these readings. This position can also be assigned to a step from part 10.

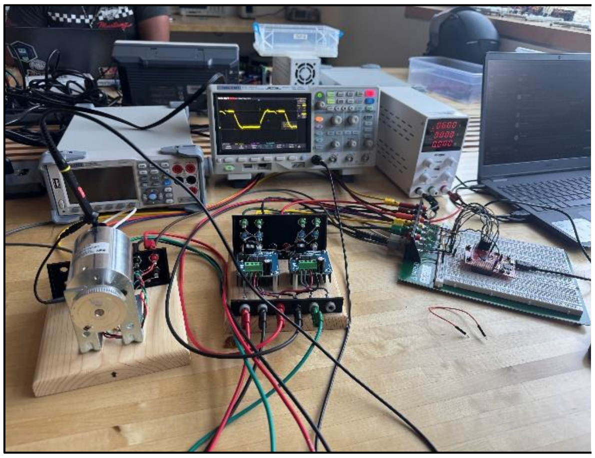
Therefore, according to the hall effect readings, the redboard enabled different GPIO output pins corresponding to the enable and direction inputs of the motor.

This code had the option to start at a slow speed of commutation, with a waitMicrosecond function call after every step. The value used in this function call slowly decreased to 100 and still function with a value of 1. The routine also contained the option to input a wait value using putty. The motor then slowly adjusted its speed to match the inputted value.

Using waitMicrosecond with a value of 0 blocked the motor's commutation, but removing the function entirely from the code caused the motor to commutate with no waiting time between each step, and therefore reach its fastest possible speed.

12. **Next, look at phase C as the commutation is occurring. What is the range of voltages you see on phase C when phase C is not energized (this is the backemf)?**

Circuit



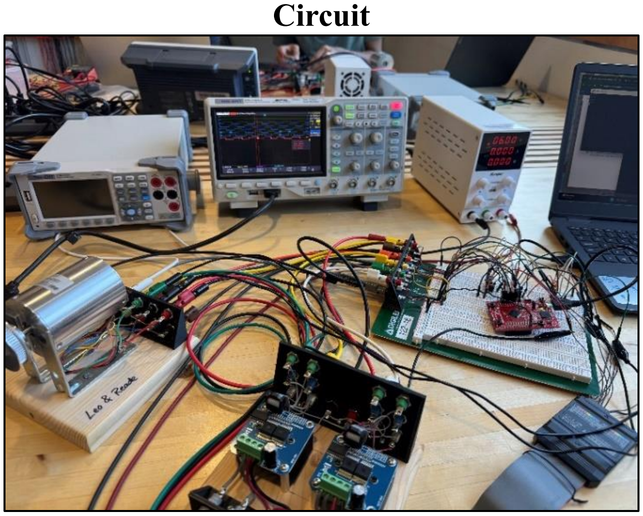
Reading

When C is first turned off, there is a voltage drop of about 7.44 V.

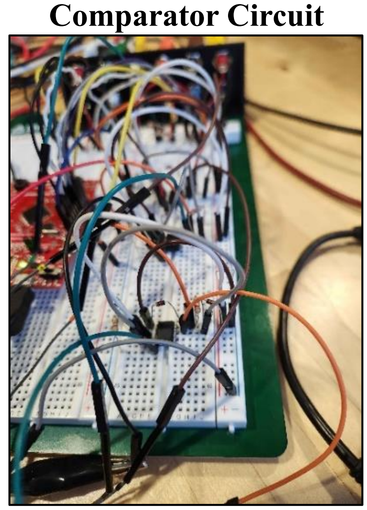
During the floating phase, or when C is not energized, there is a voltage drop of about 6.5 V with a voltage spike when set to undriven and another when set to driven.

13. **OPTIONAL: Now, connect each of the phases through a 47kohm current limiter to the positive input of 3 comparators. Connect the negative input of the 3 comparators to the virtual common. On all 4 comparator signals (from the 3 phases and virtual common), add diodes to ground and the comparator positive supply to protect the comparator.**

Circuit



Comparator Circuit



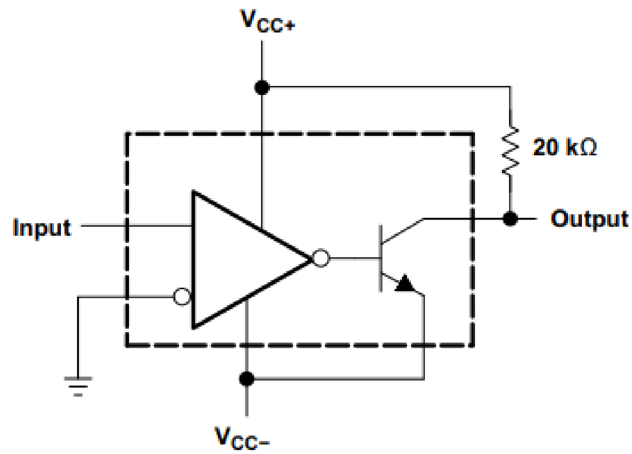
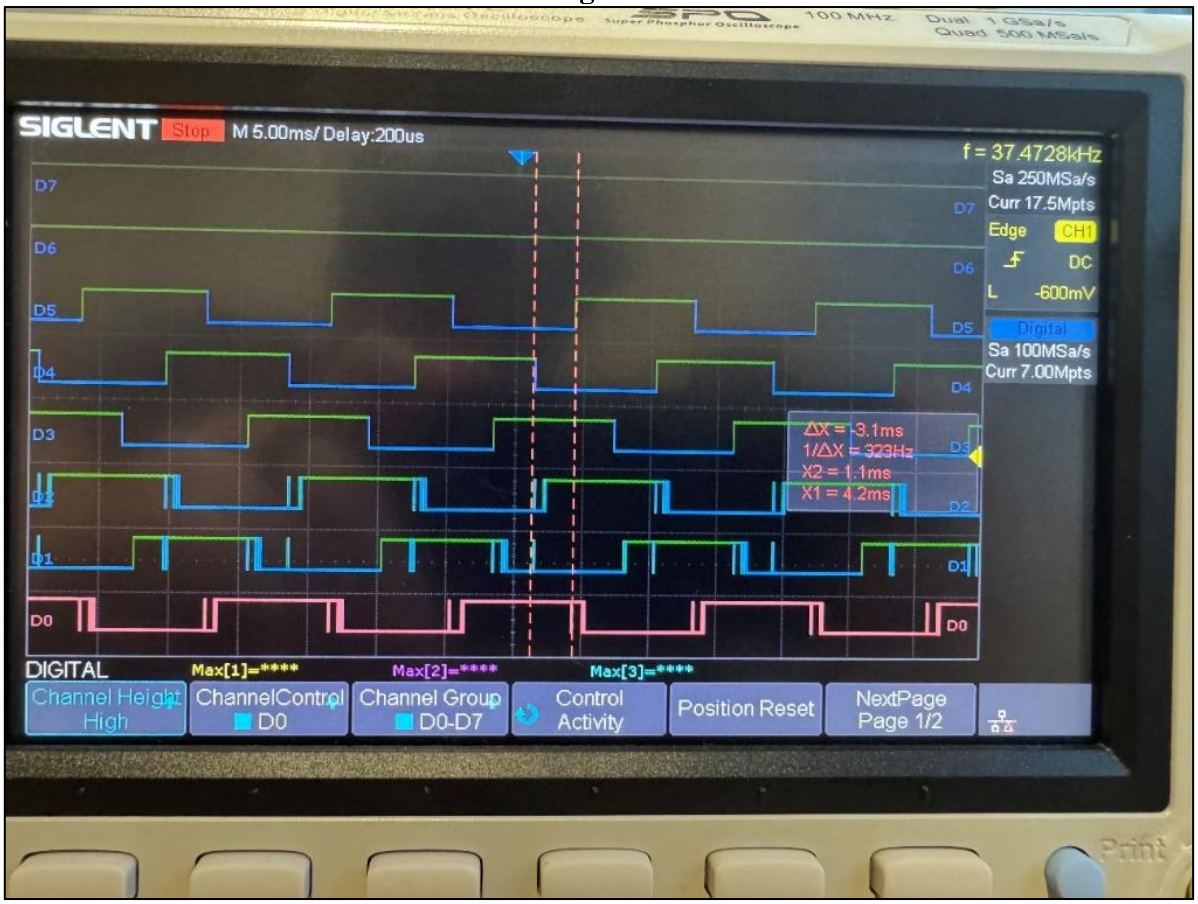


Figure 13. Zero-Crossing Detector

We used the above diagram to build the 3 comparator circuits.
Diodes were placed on ground and the comparator positive supply to protect the comparator.
The positive input on each comparator was phase A, B, & C, taken directly from the motor and connected through a 47kohm resistor.
Negative input was the virtual common ground taken directly from the motor.
On comparator output:
The emitter was connected to the general ground (Vcc-).
The collector was connected to VBUS(5V) through a 20kohm resistor (Vcc+).

14. OPTIONAL: The 3 comparator outputs should correlate with the hall sensors with some phase offset. Plot a hall effect sensor vs a comparator output to see how they are aligned. What is the minimum speed required to get accurate comparator outputs?

Readings at 0ms



By testing different wait values in the code, the minimum speed found to still create clear outputs on the oscilloscope was 11000 microseconds between each rotation. After this point, the comparator outputs become noisy and imprecise.