ECE 3331-302

Motor Mini-Project for TTU Robotics Lab 1

Rexford Brittingham

Mark Franzino

Kyle Hansz

Texas Tech University

September 2023

Abstract

Finding new and better ways to solve a pervious problem is the purpose of being an engineer. This report examines the process of the building and designing of a working and moving rover chassis, using the Basys3 board and the L298 dual H-bridge. This project was designed and made to demonstrate skills and practical knowledge of electrical and computer engineering. The circuit schematics, code snips, and other readings will demonstrate the skills and tools used to accomplish this task.

Acknowledgements

Lab Group:

Kyle Hansz, Mark Franzino, Rexford Brittingham

Funding:

Funded and managed by the Texas Tech Electrical and Computer Engineering department and Electrical and Computer Engineering stockroom

Equipment:

KEYSIGHT DMM, Triple Output DC Power Supply, 5 ½ Digital Multimeter, Waveform Generator, Digital Storage Oscilloscope

Technical Support:

Dr. Hemmert, Dr. Kai Wu, PhD, ECE Stockroom

Others:

Tyler Wooten, Erik Vaughn, Royce Black, Carter Diamond, and the rest of the ECE project lab 1 students

Table of Contents

List of Figures

List of Tables

1. Introduction

Tasked with the mission of enabling autonomous rover navigation, our team assumed the responsibilities of circuit design and code development. With a constrained budget of $30 and a short 4-week timeline, being compelled to optimize both our resource allocation and time management.

During the time of the assembly, this project allowed for free thinking and strategic decision making. Given the freedom of design and limited resources for this project, this report will demonstrate the steps taken to accomplish this task.

The consensus for this project was centered towards the thought of excellence and efficiency; basic tests that lead to more detailed outcomes, allowing for a better grip on the project and skills for the engineering field. This report stands as a systematic journey, documenting the steps taken to complete this challenge.

2. Hardware

The hardware for this project was completed with a LM339 comparator chip and a 1 A fast blow fuse. The LM339 comparator chip was used to create an overcurrent detection circuit, allowing for the rover to detect a 1 A current and shut itself off to ensure the safety of the Basys3 board. The 1 A fuse was put in place as a deterrent as a secondary measure to protect from a 1 A current if the detection were to fail.

2.1 Dual H-Bridge Motor Drive

A L298 dual H-Bridge was the main source of power for the 2 motors. The L298 allowed for the use of two H-Bridges at once, allowing for a dual output and the movement of two motors.

BLOCK DIAGRAM

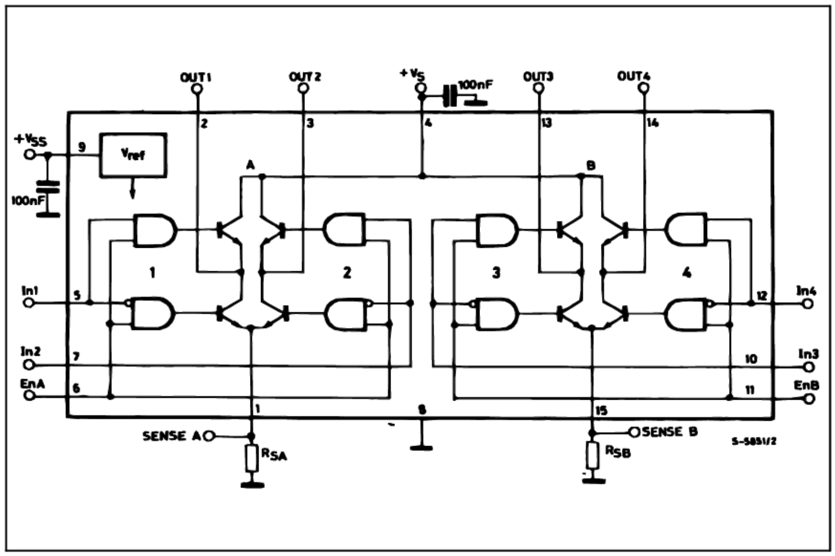


Figure 1: Dual l298 H-Bridge

In Figure 1, seen above, it is seen that there are 4 outputs, 2 for each H-Bridge. This allows the user to have control of 4 different outputs: motor 1 spinning clockwise or counterclockwise, and motor 2 spinning clockwise or counterclockwise [1]. This allowed for a choice of direction. Sense A allowed for control of outputs 1 and 2, whereas Sense B allowed for the control of outputs 2 and 3.

Using a 9.6 V rechargeable battery for the source of power, this ran through the 5 volt-regulator to provide power for the overcurrent protection circuit, which will be discussed later. The regulator dropped the 9.6 V down to 5 V, allowing for a proper Vcc value.

2.2 LM339 Comparator

The Lm339 comparator chip is a chip designed and manufactured by Texas Instruments and was the basis for the overcurrent detection circuit. Seen in Figure 2.

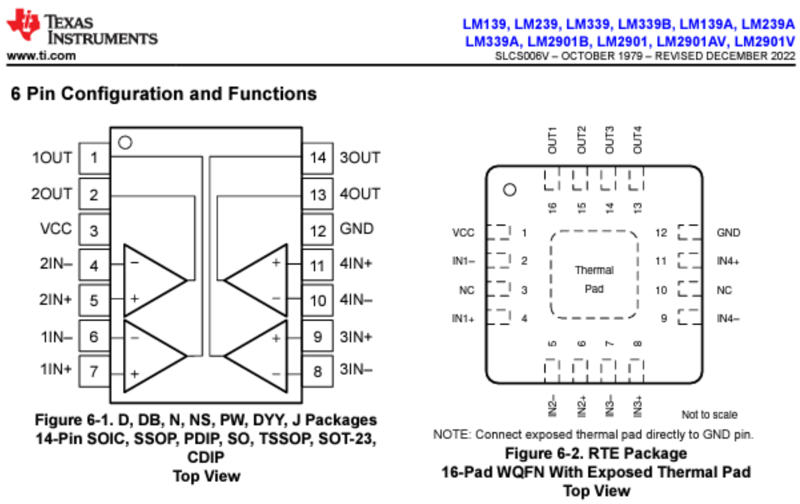


Figure 2: LM339 comparator diagram

The Lm339 chip has 4 internal comparators, each with one IN- and IN+ and a corresponding OUT, having an absolute maximum input of 36 V. This chip is also equipped with a VCC and GND, allowing for personal choice in rails, with an absolute maximum of +36 V [2].

The negative terminal was to be a 1 V input, with a 10% error, and the positive terminal was to be compared to the corresponding negative terminal. The Vcc terminal was measured at 5 V, allowing it to be within the suggested range as well as providing enough voltage to have the chip functioning as intended. The objective of using this chip was to achieve a logic level high when the positive terminal was to reach above the 1 V input at the negative terminal, providing data of overcurrent being detected. While the positive terminal was below the 1 V input, it would read a logic level low, providing the data of no overcurrent being detected.

2.3 First Tests

Simulations of this idea were made, providing a simple way to test how the comparator should behave based on specific voltage inputs. A 3.3 V input connected to a 3.3 Ω resistor feeding into the negative terminal was used as the reference voltage. As well as an AC voltage feeding into the positive terminal. Attached to Vcc was an 11 KΩ resistor, acting as a pulldown resistor.

The idea was to give the comparators' Vcc a higher voltage than its' Vref, allowing the Vref input not to exceed the rail. The 11 KΩ pulldown was added to pull down the output in the physical design if overcurrent was detected [4]. The circuit was designed to read a logic level high when Vin, the positive terminal, read above 3.3 V. This is illustrated in Figure 3.

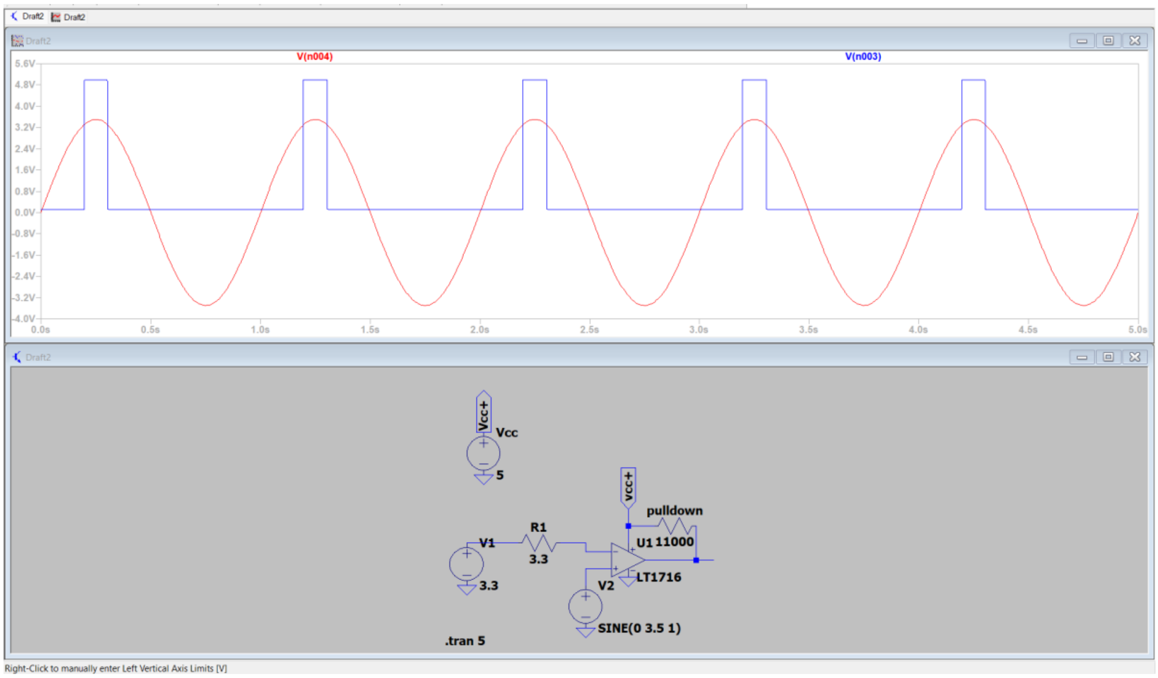


Figure 3: Comparator Simulation 1

Seeing the simulation provide promising outcomes, a physical test was made. It used the same components seen in Figure 4.

Figure 4: Comparator Reading Low

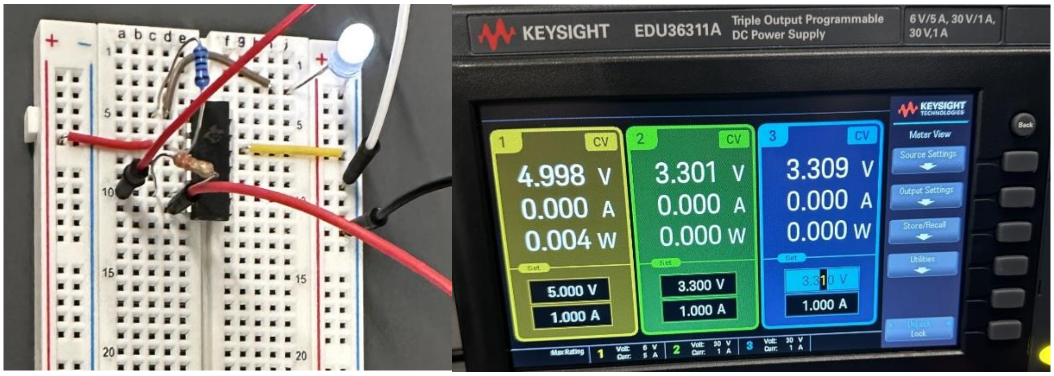


Figure 5: Comparator Reading High

Figures 4 and 5 demonstrate the comparator working at the expected values. Figure 4 shows the comparator reading a logic level low when the positive, Vin, terminal feeds a lower voltage than the compared negative, Vref, terminal. This is demonstrated by the LED indicator not being lit. Figure 5 shows the comparator reading a logic level high when the positive terminals voltage exceeds the negative terminals voltage. This is demonstrated by the LED indicator being lit, showing overcurrent being detected.

These simulations and physical tests, while not having the necessary requirements, allowed for a better understanding of how the component being used was to work and react in real life. As mentioned above the negative terminal was to have a 1 V, with a 10% error, input, thus giving the 1 to 1 relation of voltage to current.

2.4 Final Tests

With the project needing a 1 V input to the negative terminal, a drop in voltage from the 3.3 V output to the Vref input was needed. A voltage divider was used to accomplish this effectively [3]. Demonstrated in Figure 7.

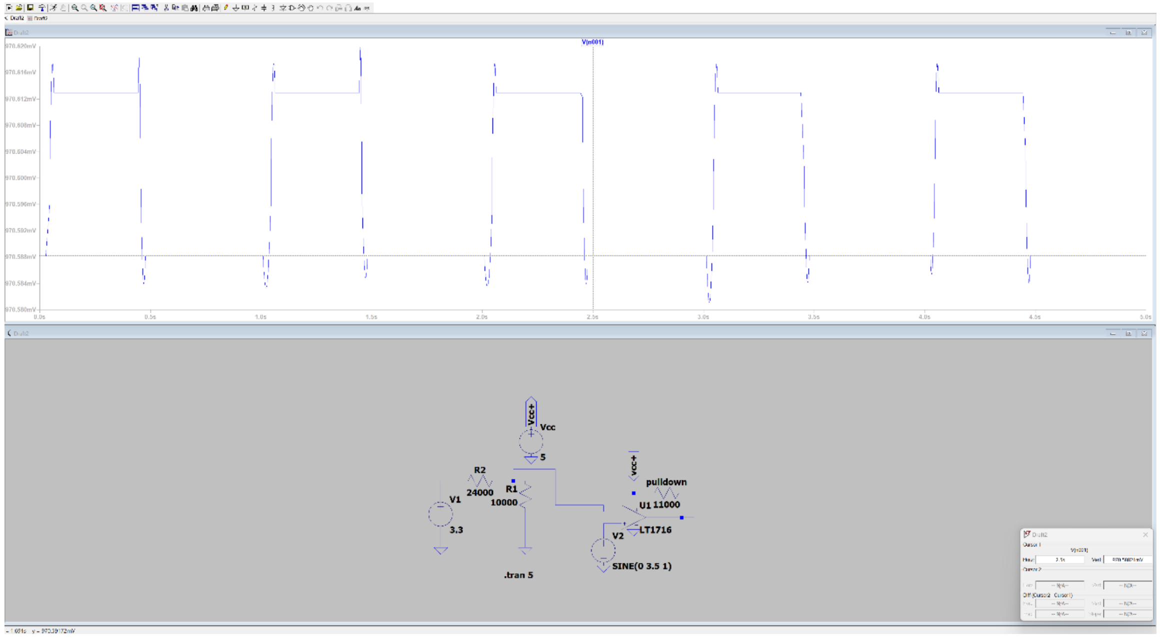


Figure 6: Voltage Divider Input

Figure 6 shows the simulated voltage divider providing an average 970.6 mV output, feeding into the negative terminal. The peek output being 970.618 mV and the lowest being 970.583 mV, the output could be considered 970.6 mV. In the grand scheme of the simulations, Vref will be considered the above, 970.6 mV.

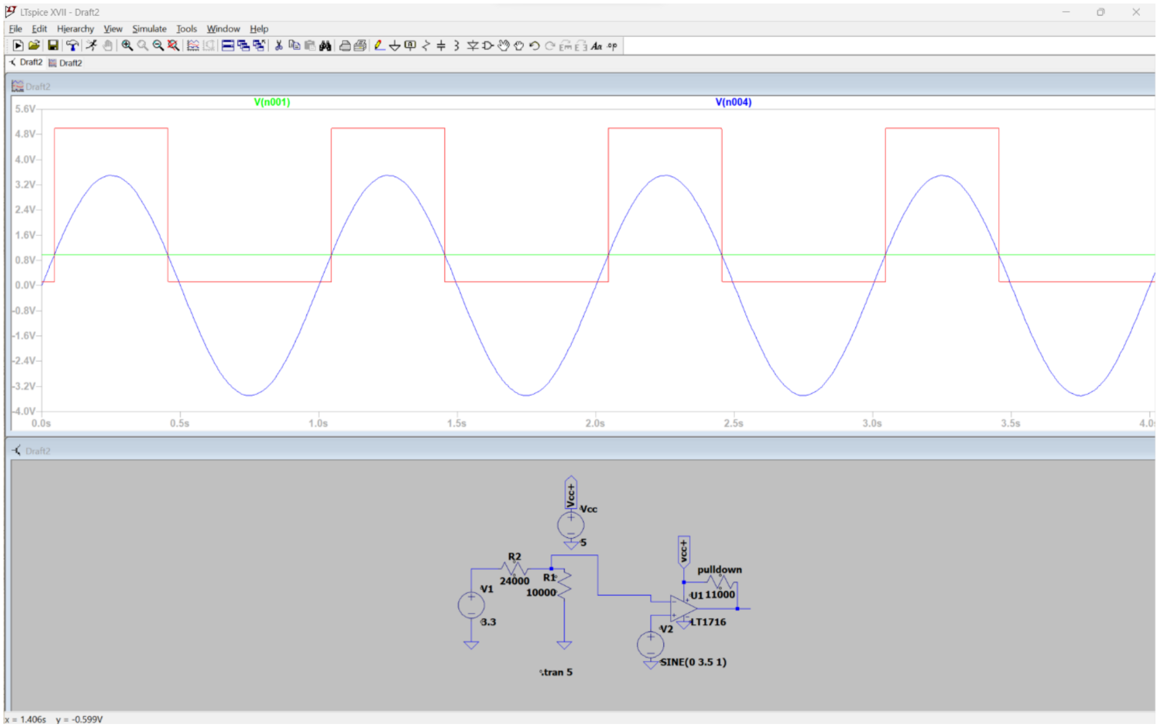


Figure 7: Final Overcurrent Simulation

Figure 7 shows the circuit in whole preforming as intended. The output, red square wave, shows that when the Vin, blue sin wave, is above the Vref, green DC value, it outputs a logic level high. When the Vin is below the Vref voltage value it outputs a logic level low.

A new design using the parameters from the simulation in Figure 7, gave the footprints to follow to recreate this circuit for physical testing. Figure 8 shows what the new circuit looked like. The new design incorporates two out of the four comparators in the lM339 chip. This was done because there are two motors to consider for the rover chassis, meaning two locations where overcurrent could happen and needs to be detected. A more durable design was made, seen in Figure 8, allowing for a more compact and sturdy circuit. The 5-volt regulator was added to limit the voltage to Vcc to the needed 5

V, this was later removed due to the use of an external rechargeable 9.6 V battery as the main source of power. The battery was put through the 5 V regulator and connected to the Vcc input.

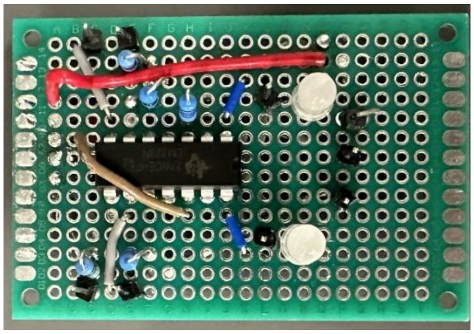


Figure 8: Protoboard Overcurrent Design

The proto board overcurrent detection circuit, seen in Figure 8, was tested, and preformed as intended, a second output was made as well, used for the second motor. Seen in the side-by-side comparison of Figure 9, the circuit now detects overcurrent at a Vin value of above 0.97 V. Having a 1 to 1 relationship of voltage to current meant that when it was above the approximate 1 V, read as 1 A, overcurrent was detected.

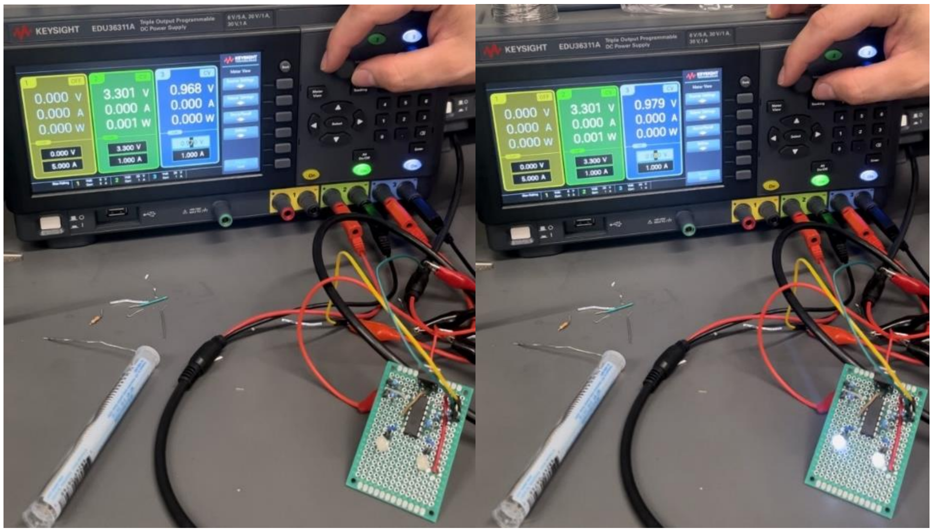


Figure 9: Dual Input Overcurrent Detection

3. Software

The software for this project was done by using Vivado to code an FPGA board: the Basys3 board. The use for this was to control the output speeds and direction of the rover and program a 7-segment display. The Basys3 board consists of 16 singular switches, a 7-segment display, Jumper pins, and I/O ports. Switch 0 is mapped to be the main power for the board, switch 1 controls the direction the motors spin, and switches 2 through 7 control the motor output speed. Each of the 2 jumper pins on the board have their own use as well. The power jumper pin allows for the Basys3 to be able to have power come from a USB cable, or internal power. For the code jumper, this allows for the code to be QSPI, having it code from its own memory. Illustrated in Figure 10.

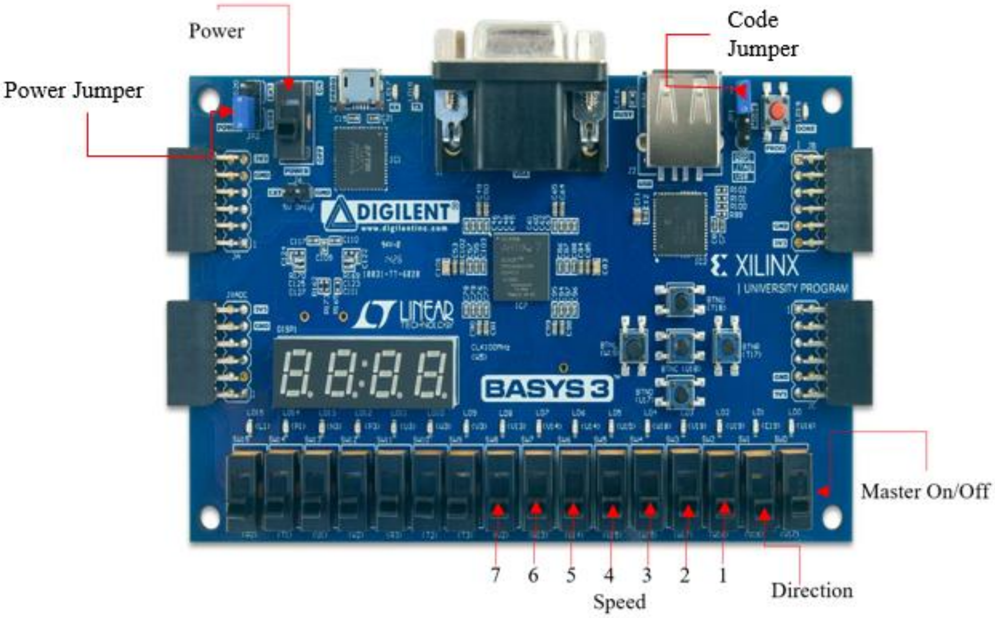


Figure 10: Basys3 Switch Controls

3.1 Speed Control

As stated above switches 2 through 7 determine the speed at which the motors spin. To determine the speed, the master on/off switch is to be turned on and the direction switch is to be set either forward or backwards.

To assign the switches to the correct outputs, a register was created, and they were assigned to that registry as a binary number. These were then set to a max counter using a divided clock signal.

```
module ClockDivider(
    input clk,  // FPGA base clock signal
    input reset // Global reset signal for all internal dffs
    );

    // Connections for the DFFs to make the clock divider
    wire [19:0] dbus;
    wire [19:0] clkbus; // clkbus[10] is going to be the output for top modules calling this

    // Instantiate the first DFF manually
    DFF dff_inst0 (.clk(clk), .reset(reset), .D(dbus[0]), .Q(clkbus[0]));

    // Generate the rest for easy changes to amount of division later
    genvar i;
    generate for (i = 1; i < 20; i = i + 1) begin : DFF_gen
        DFF dff_inst (.clk(clkbus[i-1]), .reset(reset), .D(dbus[i]), .Q(clkbus[i]));
    end endgenerate

    // Insert inverters
    assign dbus = ~clkbus;

endmodule
```

Figure 11: Code of the Clock Divider

Using a 6-bit counter, 63 different speeds were able to be obtained. The counter started at 3 so our speeds were starting at 0 to 61, getting rid of those 3 speeds gave a more precise frequency for the PWM signal. With the use of having a 6-bit selection and 61 different speeds, the rover was able to have precise speed control. The speeds range from 0-0 up to 3-F, in a hexadecimal format. While having 61 speed options, speeds 2-0 (50% duty cycle) up to 3-F are the only options that work with the motors. Figures 11 through 13 demonstrate the output signal for 50%, 75%, and 100% duty cycle being ran at 100 Hz frequency.

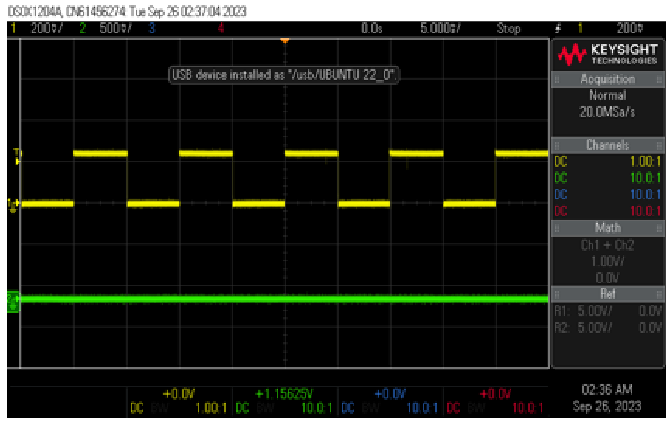


Figure 12: 50% Duty Cycle

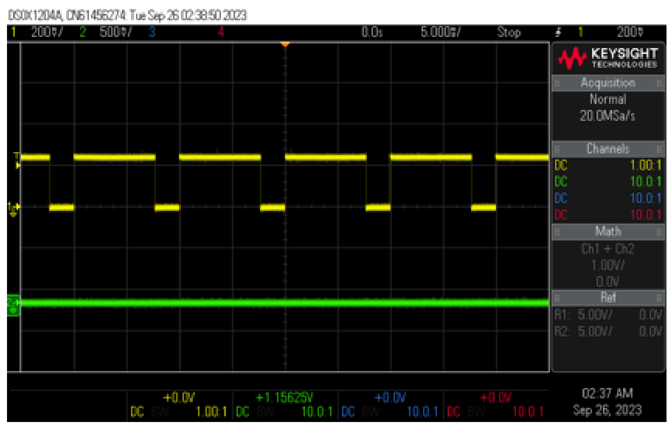


Figure 13: 75% Duty Cycle

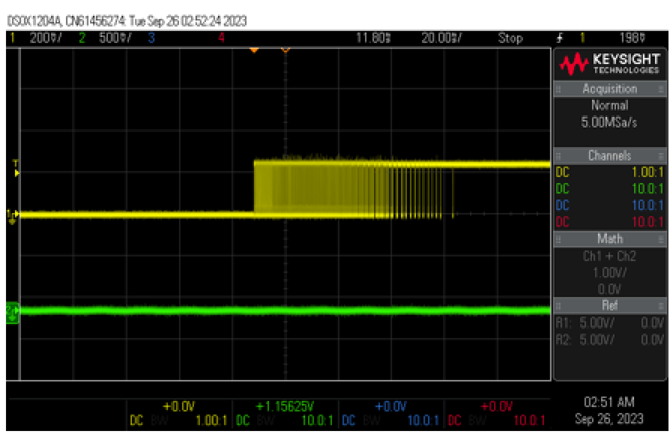


Figure 14: 100% Duty Cycle

3.2 7-Segment Display

The 7-segment display was used to show the multitude of information the Basys3 provided. Having 4 digits to use, the display was used; accordingly, digit 0 showed the direction of the spinning motor, digits 1 and 2 were used to display the speed in hexadecimal format, and digit 3 was used to indicate overcurrent being detected or not. Along with digit 3 indicating overcurrent, the decimal point of digit 3 shows a direct indicator from the overcurrent protection circuit.

```
// Set display digit every cycle of the given clock, at 190Hz
always @ (posedge(divdclk)) begin
    // Rotate the selected digit
    anout <= {anout[0], anout[3], anout[2], anout[1]};

    // Based on the digit, assign output
    case(anout)
    4'b0111: begin
            // 1 - First Digit of Speed, 0-3
            BtoH = 0;
            BtoH[1:0] = sw[7:6];
        end
    4'b1011: begin
            // 2 - Second Digit of Speed, 0-F
            BtoH = sw[5:2];
        end
    4'b1101: begin
            // 3 - Overcurrent Detection, 1 (with a decimal to look like !) or 0
            BtoH = JB[0];
        end
    4'b1110: begin
            // 0 - Directional display, F or B
            BtoH = sw[1] ? 4'b1111 : 4'b1011;
        end
    endcase
end
```

Figure 15: Assignment of 7_Segment

Each speed was shown on the 7-segment display with a corresponding Hexadecimal value, seen in Figure 14. The values ranged from 00 to 3F, meaning speed is set to minimum up to maximum, respectively.

4. Results

The overcurrent protection allows for the detection of a 1 A current being detected across the Basys3 board. There are two inputs and outputs available for both SENSE-A and SENSE-B. The LEDs connected to the output of the overcurrent circuit and the 4th digit of the 7-segment display both show when overcurrent is detected.

Based on the multiple tests run with the overcurrent and the many lines of code, the overall Mini Motor project works, and meets all the expectations required. The PWM signals going into the motors allow us to accurately control the output speed. Each speed

can be triggered and changes on the Basys3 board, from 0% to 100%, while the rovers' motors start to spin at a 25% PWM signal, the ability to choose between 0% and 100% is available.

When overcurrent is detected, the motor that it is detected in shuts off and waits till the 1 A current is gone, then turns itself back on.

5.Summary and Conclusions

The objective was to have 1 of the 2 motors moving forward and backwards with variable speeds, show a working use of overcurrent detection and protection, and show Oscilloscope readings of the different PWM signals, proving different speeds were achieved.

As a group, every objective was completed, and extra work was done to improve the initial design of the Mini Motor project.

6. References

1. STMicroelectronics. "Dual Full-Bridge Driver - Sparkfun Electronics." *DUAL FULL-BRIDGE DRIVER*, 2000, www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf.
2. TI. "LM339B, LM2901B, LM339, LM239, LM139, LM2901 Quad Differential ... - Ti.Com." *LM339b Data Sheet*, Dec. 2022, www.ti.com/lit/ds/symlink/lm339b.pdf?ts=1674127236769.
3. Tyler Wooten, Face to Face Conversation, September 15, 2023
4. Bishop , Josh. "How an Op-Amp Comparator Works - Electronics Tutorials." *CircuitBread*, 14 Oct. 2022, www.circuitbread.com/tutorials/how-an-op-amp-comparator-works.

7. Appendices

7.1 Budget Sheet:

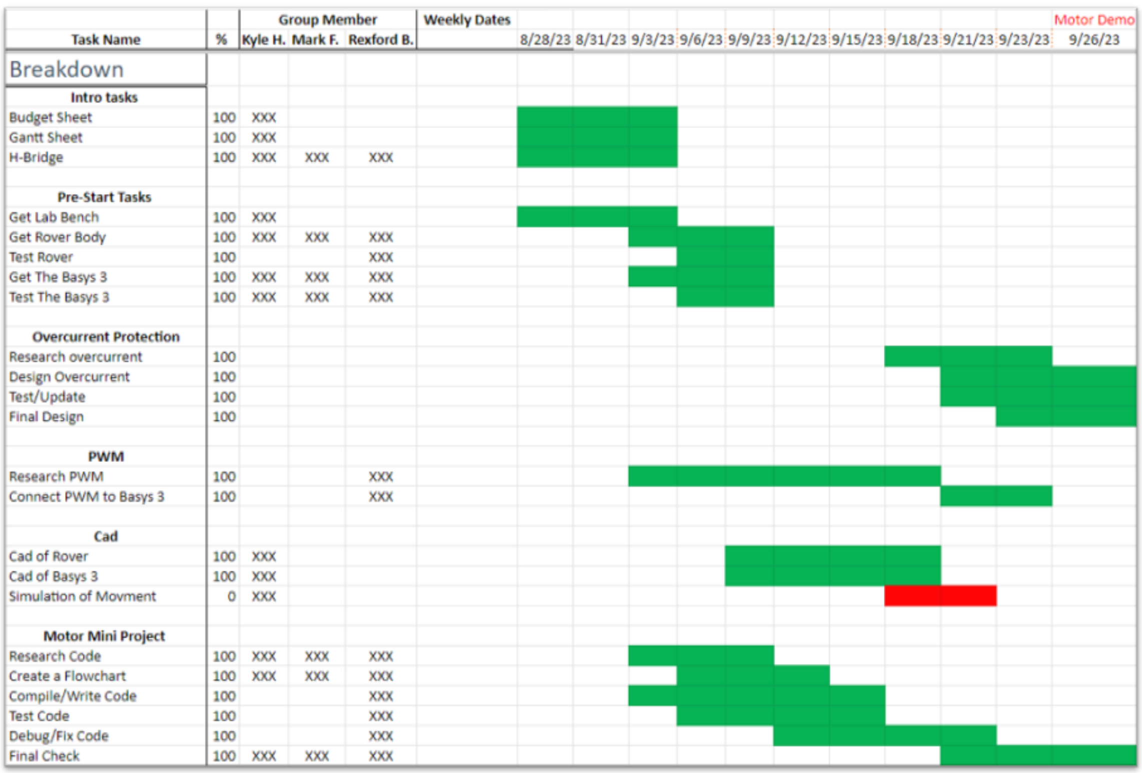
Our budget shows that the team performed all the required tasks under the estimated timeframe originally estimated. Having extra working hours allowed for more help from others in and around the lab. $3,794.37 was the final running total, this is $1591.31 under what was expected.

Labor	Hourly Rate	Hours	Total	Hourly Rate	Hours	Total
Kyle Hansz	$15.00	52	$780.00	$15.00	29.5	$442.50
Mark Franzino	$15.00	52	$780.00	$15.00	24	$360.00
Rrexford Brittingham	$15.00	52	$780.00	$15.00	29	$435.00
Labor Total	Rate	Total Hours	Total Labor Cost	Rate	Total Hours	Total Labor Cost
	100.00%	156	$2,340.00	100.00%	82.5	$1,237.50
Contract Work	Hourly Rate	Hours	Total	Hourly Rate	Hours	Total
ECE 3332	$18.00		$0.00	$18.00	4	$72.00
ECE4333	$25.00		$0.00	$25.00		$0.00
Class TA/Lab Assistant	$50.00		$0.00	$50.00		$0.00
Professors	$200.00		$0.00	$200.00		$0.00
Contract Work Total		Total Hours	Total Labor Cost		Total Hours	Total Labor Cost
		0	$0.00		4	$72.00
Parts	Cost Per Unit	Total Units	Total cost	Cost Per Unit	Total Units	Total cost
H-bridge	$47.81	3	$143.43	$47.81	3	$143.43
Rover	$49.00	1	$49.00	$49.00	1	$49.00
Basys 3	$162.00	1	$162.00	$162.00	1	$162.00
LM339B	$0.44		$0.00	$0.44	2	$0.88
11k Resistor	$0.55		$0.00	$0.55	1	$0.55
10k Resistor	$0.59		$0.00	$0.59	2	$1.18
24k Resistor	$0.55		$0.00	$0.55	2	$1.10
LED	$0.07		$0.00	$0.07	2	$0.14
Parts Total	Total Parts Cost			Total Parts Cost		
	$354.43			$358.28		
Equipment Rental Cost	Cost To Rent	Rental Fee	Total Cost	Cost To Rent	Rental Fee	Total Cost
O-Scope	$957.00	0.20%	$191.40	$957.00	0.20%	$191.40
DC Power Supply	$1,000.00	0.20%	$200.00	$1,000.00	0.20%	$200.00
Function Generator	$1,100.00	0.20%	$220.00	$1,100.00	0.20%	$220.00
Multimeter	$844.00	0.20%	$168.80	$844.00	0.20%	$168.80
Equipment Cost Total	Total Rental Cost			Total Rental Cost		
	$780.20			$780.20		
	Overhead	Total Expences	Total Cost	Overhead	Total Expences	Total Cost
	55.00%	$3,474.63	$5,385.68	55.00%	$2,447.98	$3,794.37

7.2 Gantt Chart:

The Gantt Chart shows that the team completed all the required tasks. Designs, testing, and final adjustments all went well and performed as expected, as well as meeting every expectation.

The only task that wasn't completed was a CAD simulation of movement, this was due to the fact that it was not required and unnecessary to the main goal.



7.3 Code

10. References