



UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

E344 Assignment 1

Lee Johnson

24058661

Report submitted in partial fulfilment of the requirements of the module
Design (E) 344 for the degree Baccalaureus in Engineering in the Department of Electrical
and Electronic Engineering at Stellenbosch University.

September 15, 2023

Plagiaatverklaring / Plagiarism Declaration

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.

I also understand that direct translations are plagiarism.

4. Dienooreenkomsdig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingsstekens (selfs al word die bron volledig erken) plagiaat is.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

24058661	
Studentenommer / Student number	Handtekening / Signature
L. Johnson Voorletters en van / Initials and surname	September 15, 2023 Datum / Date

Contents

Declaration	i
List of Figures	iv
List of Tables	vi
Nomenclature	vii
1. Literature survey	1
1.1. Operational amplifiers	1
1.2. Current sensing	3
1.3. Interfacing with and using the ultrasonic range sensor	4
1.4. Converting PWM signals to analogue	5
1.5. Fundamental operation of the range sensor	6
1.6. Converting Digital Values to Analogue Equivalents	7
1.7. Lead-acid battery voltages and currents	7
2. Detail design	9
2.1. Current sensor	9
2.2. Analogue range sensor	11
2.3. Digital to Analogue Converter	12
2.4. Motor control signal and driver	13
2.4.1. Motor Control	13
2.4.2. Emitter Follower Buffer	14
2.5. Digital left wheel control	15
2.5.1. Current Sensing	15
2.5.2. Low-side switch	16
2.5.3. Firmware (Range Sensor and PWM control)	17
2.6. Battery	18
2.6.1. Battery Charger	18
2.6.2. Under-voltage Protection	19
2.6.3. Battery Voltage Signal Conditioning	20
2.7. Bluetooth Connection Protocol and Firmware Design	21
2.8. Graphical User Interface Design	22
3. Results	24
3.1. Current sensor	24
3.2. Analogue Range Sensor	27

3.3.	Digital to Analogue Converter	29
3.4.	Motor Control	30
3.5.	Battery	31
3.5.1.	Battery Charger	31
3.5.2.	Under-voltage Protection	32
3.5.3.	Battery Voltage Signal Conditioning	34
3.6.	Bluetooth Connection Protocol and Firmware Results	35
3.7.	Graphical User Interface Results and Screenshots	37
4.	Physical implementation	38
4.1.	Current sensor	38
4.2.	Digital to Analogue converter	39
4.3.	Motor Control and Driver circuit	40
4.4.	Complete RC Car Circuit	41
Bibliography		42
A. Social contract		44
B. GitHub Activity Heatmap		45

List of Figures

1.1.	Equivalent Circuit of an Ideal Operational Amplifier, as provided by	1
1.2.	Inverting and Non-inverting Op-Amp configurations and their gain equations .	2
1.3.	Differential and Summing Op-Amp configurations and their gain equations .	2
1.4.	Inverting and Non-inverting Op-Amp configurations and their gain equations .	2
1.5.	Current Sensing Methods	4
1.6.	Timing Diagram of the HC-SR04 Ultrasonic Range Sensor	5
1.7.	Frequency Spectrum of a pulse at 3.3V, showing the DC component in green and carrier and harmonic frequencies in red.	6
1.8.	Operation of an Ultrasonic Transmitter and Receiver Pair	7
1.9.	Inverting Summing Amplifier Circuit	8
2.1.	Difference amplifier used to measure current	9
2.2.	Current Sensing Circuit Diagram	10
2.3.	Active Low Pass filter with Amplification	11
2.4.	Digital to Analogue Converter circuit Diagram	13
2.6.	Hardware Implementation of the Digitally Controlled Left Wheel	16
2.7.	Flow Diagram of Firmware Implementation to control the left motor digitally .	17
2.8.	Battery Charger Circuit	19
2.9.	Schmitt Trigger Under-Voltage Protection Circuit	20
2.10.	Battery Voltage Signal Conditioning Circuit	21
2.11.	Flow diagram illustrating the process of instructing and sensing data over bluetooth	22
2.12.	Flow Diagram illustrating the interaction between Front end, API and Car for GUI implementation	23
3.1.	Simulation output vs input voltage	24
3.2.	Circuit Current Draw at 200mA input	25
3.3.	Current sensor circuit output voltage when motor is free running, slightly loaded or momentarily stalled respectively.	26
3.4.	Step input vs Step response of analogue range sensor displaying met requirements of lower than 0.3V at low and higher than 3V at high ranges.	27
3.5.	Noise level on analogue input is around 45mV	27
3.6.	Current Draw of the analogue range sensor circuits remains lower than $100\mu A$.	27
3.7.	Step input vs Output Response of analogue range sensor	28
3.8.	Simulation output (blue) vs input (grey) demonstrating requirement compliance as well as current draw (red) at a max of $260\mu A$	29

3.9.	Simulated motor control output (green) vs Range Sensor (Blue) and Torque (red) input demonstrating requirement compliance	30
3.11.	Voltage output of current sensing circuit when the wheel is turning freely	30
3.10.	Measured results of left wheel control pulse entering switch at different conditions	31
3.12.	Battery Charger Circuit Output Voltage (green) and Output Current (blue) . .	31
3.13.	Charger output over a $1k\Omega$ and a 50Ω resistor respectively	32
3.14.	Simulated Schmitt Output (blue) vs Test Battery Voltage (Green)	32
3.15.	Simulated Motor Voltage Output (blue) vs Test Battery Voltage (Green) . . .	33
3.16.	Signal Conditioning Output (Left Multimeter) and Schmitt Trigger Output (Right Multimeter), at a battery voltage of 5.8V and 6.1V respectively	33
3.17.	Battery Voltage Signal Conditioning Simulation Result with Circuit Output (Green) and Test input (Blue)	34
3.18.	Signal Conditioning Output (on left multimeter) at a voltage of 7V	34
3.19.	Measurement setup for bluetooth connection protocol and firmware	35
3.20.	Graphical User Interface displaying measured sensor data as well as optional controls for a user	37
4.1.	Current Sensing Circuit with labels	38
4.2.	DAC circuit with labels	39
4.3.	Low-side switch and current sensor physical implementation for left motor control	40
4.4.	Physical implementation for the complete RC car circuit	41

List of Tables

1.1.	Amplifier Key Specifications and their Descriptions	3
2.1.	Differential Output Voltage when Range sense and Torque provided as inputs .	14
2.2.	ESP32 GPIO Pin Configurations	21
3.1.	DC motor currents under various conditions	24
3.2.	Measurements of DC motor currents under various conditions	25
3.3.	Range input vs Output	28
3.4.	Measured output voltage of DAC converter vs nibble input via a DIP switch .	29
3.5.	Full Speed with No Objects Close By	35
3.6.	Hard turn right with no objects close by	36
3.7.	Full speed with object near by left wheel sensor	36
3.8.	Full speed with object near by right wheel sensor	36

Nomenclature

Variables and functions

V	Voltage
A	Amperes
A	op-amp gain
f_c	Cut-off Frequency
τ	Time Constant
Hz	Hertz or per second

Acronyms and abbreviations

Op-Amp	Operational Amplifier
IC	Integrated Circuit
DAC	Digital to Analogue Converter
AH	Ampere Hours
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NMOS	N-Channel Metal Oxide Semiconductor
PMOS	P-Channel Metal-Oxide Semiconductor

Chapter 1

Literature survey

In this chapter we will build a foundation for operational amplifier configurations and current sensing methods that will allow us to design and implement a current sensing circuit for the right wheel of our RC Car. We will discuss basic operational amplifier configurations and their most important properties before diving into it's implementation into current sensors as well as analysing the best op-amp configurations for them.

1.1. Operational amplifiers

Operational amplifiers: Basic Principles

Op-amps are linear devices that have a vast usage in analogue electronics. They are fundamentally voltage amplifying devices, that when in combination with specific networks of resistors and capacitors around it, can perform a multitude of operations, and is thus termed an "Operational Amplifier". They are commonly used for filtering, basic amplification and mathematical operations such as addition, subtraction, integration and differentiation. Thus, it is clear to see why op-amps are as popular as they are useful.

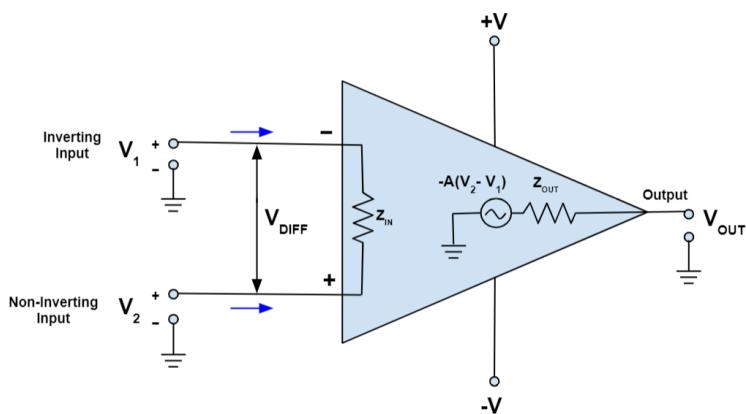


Figure 1.1: Equivalent Circuit of an Ideal Operational Amplifier, as provided by

Operational amplifier configurations

As per [1], there are 6 main op-amp configurations that we can use to amplify the current sensing signal. Each of these configurations may use the same op-amp IC but are differentiated by the resistor and capacitor network around them.

The inverting and non-inverting amplifiers are the most basic and create a negative or positive voltage gain respectively.

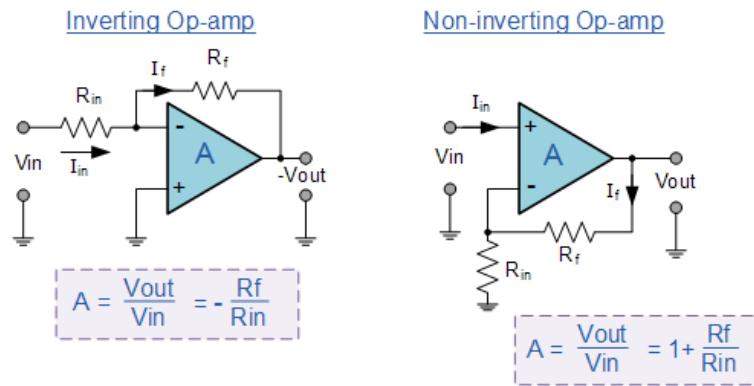


Figure 1.2: Inverting and Non-inverting Op-Amp configurations and their gain equations

Differential and summing amplifiers allow us to amplify the difference between or the sum of multiple signals.

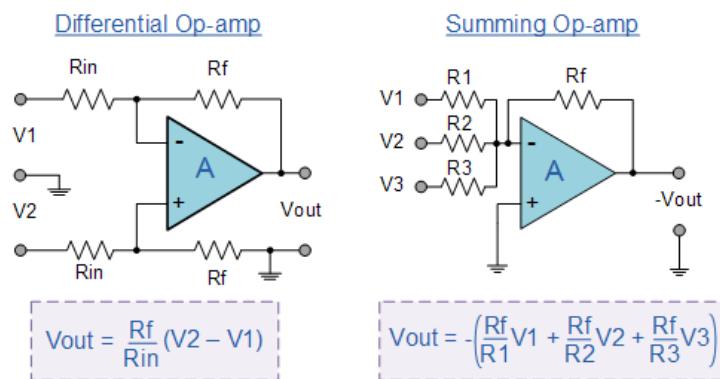


Figure 1.3: Differential and Summing Op-Amp configurations and their gain equations

The integrator and differentiator amplifier outputs the mathematical integral and derivative of the input signal respectively.

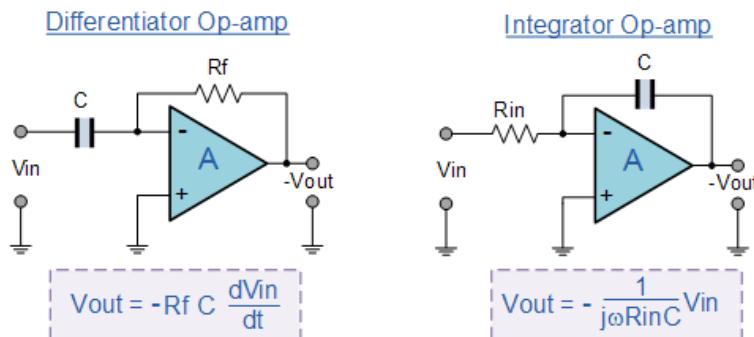


Figure 1.4: Inverting and Non-inverting Op-Amp configurations and their gain equations

Operational amplifiers: Key Specifications

Some of the key specifications of an op-amp are discussed in table 1.1. There are many more op-amp properties but the following are the most relevant to us in our search for the best op-amp configuration for our current sensing circuit. An expanded table with more properties is available on [2].

Specification	Description
CMRR	An amplifier's ability to reject input voltages common to both of its terminals
Bandwidth	The frequency range within which an amplifier is able to provide maximum output voltage
Slew Rate	The rate at which the output of an amplifier responds to its input
Gain	The ratio between the amplifier's output voltage and its input voltage

Table 1.1: Amplifier Key Specifications and their Descriptions

1.2. Current sensing

There are many different techniques for measuring current. Some of the most popular include:

Hall Effect Sensor: This technique produces an output voltage that is proportional to a magnetic field. The current is measured by measuring this magnetic field. Although this sensor can be used in higher frequencies, it is more costly than the shunt based technique. This is known as a Non-Invasive current sensing method (also known as a non-contact based method). In terms of power, the output voltage is very low and needs to be amplified, but these sensors can sense current from the mA to kA range. A Hall Effect sensor can be used for both AC and DC.

The Rogowski Coil: The Rogowski Coil (a helical shaped core coil) is wrapped around a conductor and provides an output voltage depending on the rate of current change in the conductor. The Rogowski Coil is low cost but only suitable for AC. The Rogowski coil is also non-invasive. Rogowski coil's are known to be excellent at dealing with high power modules. The Rogowski coil can only be used for AC.

The Shunt Resistor: A shunt resistor uses Ohm's Law in order to sense the current which flows through it. When the current flows through the resistor, it produces a voltage across it, and this voltage is used to measure the current flowing through the resistor. The shunt resistor method is low cost, works for AC and DC and requires no additional equipment. However, due to heat dissipation it is not suitable for higher current operation. Due to this, shunt resistors are better used for low power circumstances. A shunt resistor is an example of an invasive current sensor. The shunt resistor can be used for both AC and DC.

Some other examples of current sensors are the Flux Gate Sensor and the Current

Transformer method as described in [3], which I won't look at here.

High side current sensing occurs when the shunt resistor is placed between the supply and the load. Low side current sensing occurs when the shunt resistor is placed between the load and ground. Figure 1.5 gives a clear visual description of these two methods.

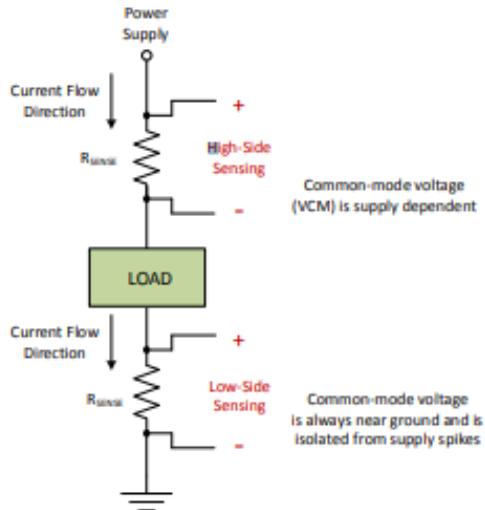


Figure 1.5: Current Sensing Methods

1.3. Interfacing with and using the ultrasonic range sensor

We will be using the HC-SR04 Ultrasonic Sensor in order to perform object detection with our vehicle. The sensor transmits sound waves and uses how long the sound waves take to return as a way to measure the distance to an object in front of it.

As per the HC-SR04 datasheet [4], we can see that the ultrasonic has an operating voltage of 5V DC and an operating current of 15mA. Thus, the sensor's power rating is given by $P = VI = 75mW$. The trigger pin on the sensor is used to "trigger" the ultrasonic sound pulses transmitted from the sensor. In order to do this, the trigger pin needs to be set high for $10\mu s$. In order to provide this trigger, we will be making use of the ESP32 micro controller to send these pulses, and thus a high will be 3.3V.

After this is done, the sensor transmits 8 40kHz pulses (these are ultrasonic pulses since it's above audible frequency for humans), and the echo pin of the sensor goes high [5]. If the pulses are not reflected back to the sensor, it times out after 38ms. However if an object is within range, it will reflect the pulses back to the sensor and after the sensor receives it, the echo pin goes low, leaving us with a pulse from the echo pin, whose length can be used to determine distance. The timing diagram of this process can be seen in figure 1.6 as seen in [5].



Figure 1.6: Timing Diagram of the HC-SR04 Ultrasonic Range Sensor

To get the distance, we can use the good old speed-distance-time relationship. In this case, we are working out for distance and our speed is the speed of sound, 340m/s. Our time is determined from the length of the echo pulse, but we need to remember to divide this by two as the echo pulse is the distance to the object and back. This will give us $d = vt = \frac{340t}{2}$.

1.4. Converting PWM signals to analogue

In a standard PWM signal, the frequency is fixed (inverse of fixed pulse period) and the pulse width is varied [6]. The pulse width is also referred to as the duty cycle.

We know that the output of the range sensor is going to be a pulse, whose width will give us information on the distance an object is from the range sensor. We want to use this information in order to create an analogue signal which can then be used in further subsequent circuits. We can do this by taking the pulse in as an input, and creating an analogue voltage output with a value of $V_O = A \times \text{DutyCycle}$ [7], where A is the voltage amplitude of the high state of the pulse. So if we have a pulse with a logical high at 3.3V and the duty cycle is 0%, we will get an output of 0V. If the duty cycle is 100%, the output will be 3.3V.

As we can see in figure 1.7, in order for us to get this desired DC output at $V_O = A \times \text{DutyCycle}$, we need to filter out the large spikes at the carrier frequency and harmonics and only leave the DC component of the pulse. In order to do this we can use a simple RC low pass filter. We can also follow this low pass filter with a gain section in order to get our output to be in whichever voltage range we need. Our filter will thus need to have a cut off frequency lower than the carrier frequency of the pulse.

From this we can see that as the duty cycle of the pulse increases, our output voltage will also increase proportionally, as we would want it to.

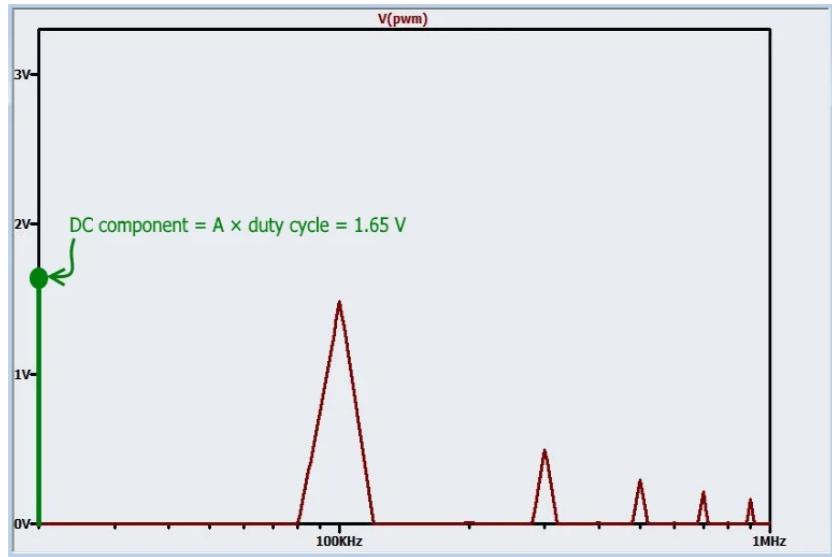


Figure 1.7: Frequency Spectrum of a pulse at 3.3V, showing the DC component in green and carrier and harmonic frequencies in red.

1.5. Fundamental operation of the range sensor

An ultrasonic sensor uses ultrasonic sound waves (sound waves higher than the audible frequency range) to measure the distance to an object. This is done by using the equation $Distance = Speed \times Time$, where the time will be given by the width of the pulse on the echo line, and the speed will be the speed of sound. Ultrasonic range sensors can use a single transducer to send and receive these ultrasonic sound waves, or individual, independent transmitters and receivers.

Most ultrasonic sensors have a trigger pin and an echo pin. The trigger pin is set to high (which could be 3.3V or 5V depending on the module) and this "triggers" the sensor to transmit the ultrasonic sound waves. At this point the echo is also set high (Also 3.3V or 5V depending on module) and is only set low once it times out, or when the sound waves return from bouncing off the object. This process is shown in figure 1.8 from [8]. This pulse width indicates the time it took the sound wave to reach the object and return. The inverse of this time will give us the carrier frequency of the pulse.

Ultrasonic sensors are extremely useful for presence detection, as it can be used to detect objects regardless of surface, colour or material, which is where some other optical technologies may fail [9].

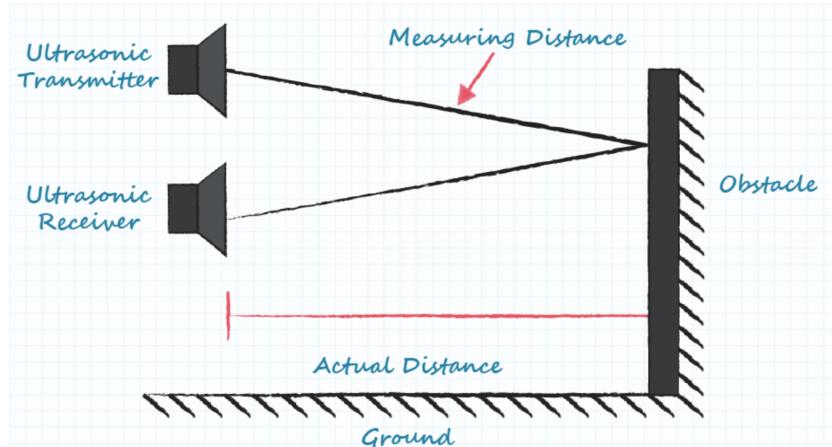


Figure 1.8: Operation of an Ultrasonic Transmitter and Receiver Pair

1.6. Converting Digital Values to Analogue Equivalents

In order to convert digital values to an analogue equivalent, we can make use of a binary weighted DAC. This can be implemented with the use of a summing amplifier. An example is shown in figure 1.9 which can be found on electronics-tutorials [1].

Whether or not the summing amplifier is inverting or non-inverting, with a zero DC offset applied to the op-amp, the output will be zero if all the inputs are zero. The difference occurs for none-zero values - For an inverting amplifier, none-zero inputs would lead to a negative voltage at the output; while for a non-inverting amplifier, none-zero inputs would give a positive voltage at the output.

As given in the datasheet for the op-amp, [10], the recommended common-mode input voltage will have a minimum of V_{DD-} and a maximum of $V_{DD+} - 1.5V$, with a maximum rail voltage of $V_{DD+} = 8V$. This means, per recommendation, we can select a minimum rail voltage higher than $-8V$ and a maximum rail voltage lower than $6.5V$. As we have a 5V and a 3.3V regulator, it would make sense to connect the output of one of these regulators to the positive rail, and the negative rail to ground, thus staying within the recommended common-mode input.

The output impedance of the source of the binary (digital) signals will be determined by R_1, R_2, R_3, R_4 as shown in figure 1.9. These values are very important as they will be used to determine the gain of the summing amplifier, and hence will ensure the amplifier's outputs will remain within given limits.

1.7. Lead-acid battery voltages and currents

A lead-acid battery is a type of rechargeable battery. It was invented by the French physicist Gaston Plante in 1859, and was the first rechargeable battery for commercial use [11]. If the battery is fully discharged, it could cause damage, and every discharge/charge cycle leads to the battery having a smaller amount of capacity. These batteries, for deep-cycle applications,

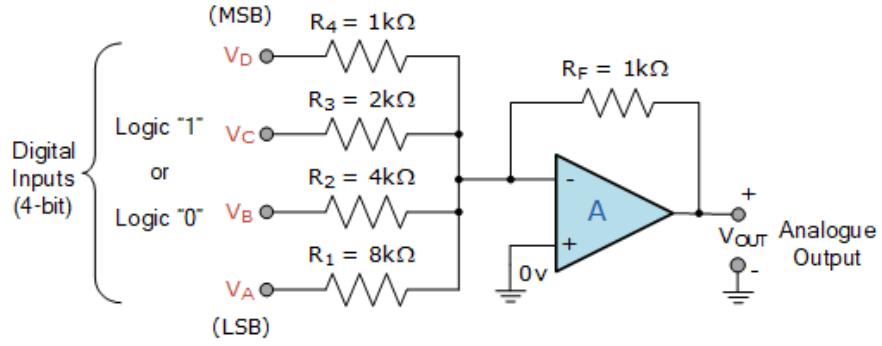


Figure 1.9: Inverting Summing Amplifier Circuit

provides about 200 to 300 discharge/charge cycles.

There are four methods for charging a lead-acid battery [12].

1. Constant Voltage
2. Constant Current
3. Taper Current
4. Two Step Constant Voltage

The best choice would be to go with Constant Voltage-Current Limited charging as it will lead to us having maximum battery service life and capacity, as well as an acceptable recharge time. Throughout the rest of this survey, I will be speaking about Constant Voltage-Current Limiting charging. As the battery voltage increases during charging, the battery's current acceptance decreases. When the current stabilises at a low level, we know that the battery is charged. When evaluating whether the battery is fully charged or not, we look at the final current level as well as the peak charging voltage while this current flows.

Batteries can be charged either on a continuous or non-continuous basis. The constant voltage charge method applies a constant voltage to the battery and limits the initial charge current. Over-charge or under-charge can be caused by inaccurate voltage settings.

The charging efficiency, η , of a battery is given by 1.1.

$$\eta = \frac{AH\text{ Discharged After Fully Charged}}{AH\text{ Delivered to Battery During Charge}} \quad (1.1)$$

Chapter 2

Detail design

2.1. Current sensor

By setting a DC power supply to 6V and connecting it to the motor with the wheel attached, we were able to stall the motor (Apply force and stop the wheel from spinning) and see the motor draw a maximum stall current of 1A. When the motor was running with no forces acting onto it, the motor was drawing 150mA.

We are using low side current sensing, so the average voltage at the measurement inputs is near zero. Due to this fact, we often use this method of current sensing when dealing with high or possibly spiking voltages. After these near zero voltages are passed into the filter/op-amp, they are amplified to a range from 0 to about 3V. The reason for this is so that our micro controller can take in the voltage level and use it to accurately measure the current through the shunt resistor, hence the amplification is needed - for accuracy purposes. However, the signal can not be amplified by too much as the micro controller has a maximum voltage input.

As described in [13], a very good op-amp configuration to use for low side current sensing is the difference amplifier. This is because it provides a very high level of accuracy, as any voltage drops from the resistor to ground is removed. Thus, I will be using a differential op-amp as shown in figure 2.1, with low-pass filter configuration in this design. I have also added a RC filter stage before the op-amp to attenuate the higher signals even further, and minimise noise.

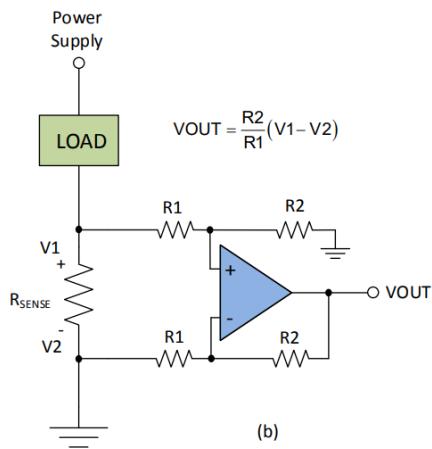


Figure 2.1: Difference amplifier used to measure current

As discussed, the input voltage is near zero (around 10mV) and we need an output voltage

that is larger than 3V but also lower than 3.3V, which is the maximum voltage allowed to be input in the ESP32 module. We can thus see we need an amplifier gain of about $A = 310$. Using equation 2.1 we can see that the ratio of $\frac{R_2}{R_1} = 310$ and by choosing resistors as small as possible (due to our circuit having a current limit of 150 μ A), we obtain $R_2 = 100k\Omega$ and $R_1 = 270\Omega$. This will give us a gain higher than needed, which will give us room to design the low pass filter.

Using equation 2.2 we can choose values for two capacitors such that they have the same value and are added in parallel to R_2 in the figure 2.1. We aim to greatly attenuate signals over 1kHz, and by making use of simulation, it was found that the best noise reduction came when the cut-off frequency was designed to be 20Hz. Using all the known values, we were able to determine that $C_1 = C_2 = 82nF$. Equation 2.3 was used to ensure that the 90% response time remained below 100ms.

It was found from simulation that the noise levels were still too high even after the low pass filter was added. It was thus decided to add an RC module before the low pass differential op-amp to attenuate the noise even further. Designing for a cut-off frequency of 150Hz (This is a sufficient cut off frequency as the low-pass filter cuts off from even lower), I was able to use equation 2.2 to obtain a resistance value of $R = 15k\Omega$ when a capacitor of 82nF was chosen (chosen to be the same value as other capacitors for simplicity). This yielded a low time constant per equation 2.3 which meant the slew rate remained fast, and was also small enough not to draw too much current.

$$V_{out} = \frac{R_2}{R_1}(V_1 - V_2) \quad (2.1)$$

$$f_c = \frac{1}{2\pi RC} \quad (2.2)$$

$$\tau = RC \quad (2.3)$$

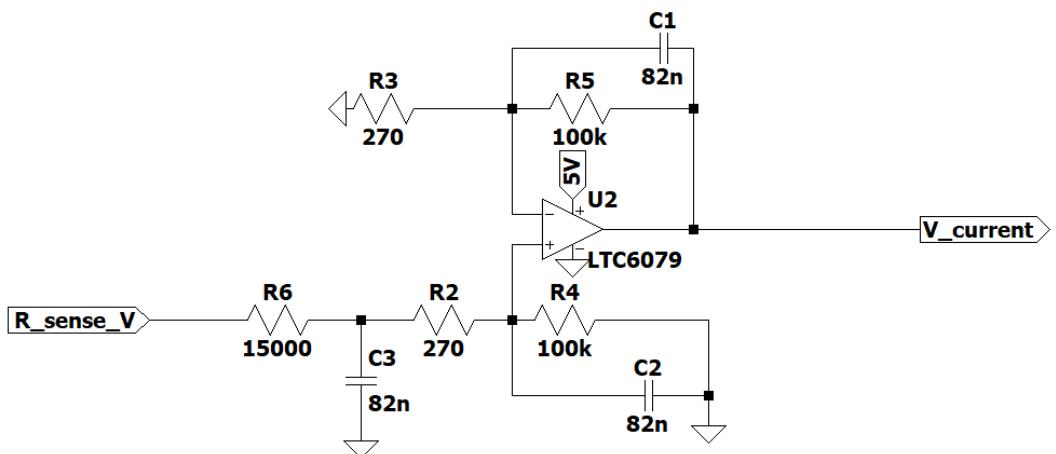


Figure 2.2: Current Sensing Circuit Diagram

2.2. Analogue range sensor

From the data sheet for the HC-SR04 Ultrasonic Ranging Sensor [14], we know that our module operates at DC 5V and 15mA. We also see that the sensor operates at a working frequency of 40Hz, and from our Arduino code, we can see that it will be outputting a frequency of 16Hz. The ultrasonic sensor will output a pulse wave, with the duty cycle of the pulse wave being proportional to the distance an object is from in front of the sensor - the further away the object, the higher the pulse duty cycle. The amplitude of this output pulse will be constant at 5V.

From the requirements stated in the Assignment 2 specifications, we can see that the two main objectives of the active filter is to firstly act as a low pass filter, by filtering out higher frequencies and providing an analogue signal from the pulse input, and secondly to amplify the filtered signal to the specified level. From these requirements, I decided to use an active low pass filter with amplification as shown in figure 2.3 from [1].

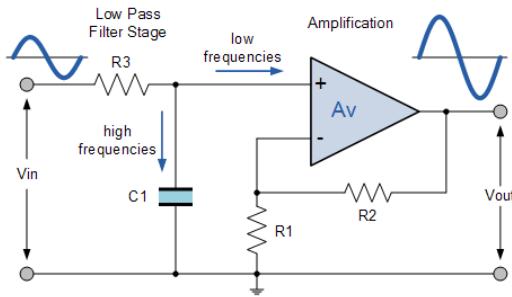


Figure 2.3: Active Low Pass filter with Amplification

The corner frequency for the low pass filter was chosen by first taking half of the ultrasonic sensor's output frequency and then lowering the corner frequency until a suitable analog signal that met specification was found. After learning in the first assessment about the close relationship between the corner frequency and the rise time, I needed to ensure that the corner frequency wasn't made so low that it resulted in an inadequate rise time. Taking into account the various duty cycles of the pulses which could be input into the filter, I found that the best way to achieve low noise and fast rise time was by using a two-stage RC filter, designing the first stage for a 1Hz cut-off frequency and the second stage for a 0.8Hz cut off frequency. These cut-off frequencies were found to be optimal after many iterations of trial and error within SPICE simulation.

By making use of equations 2.2 and 2.3, I was able to find $R_{L1} = 16k\Omega$, $C_{L1} = 10\mu F$, $R_{L2} = 200k\Omega$ and $C_{L2} = 1\mu F$, adjusting these values for practical resistors found in the lab, I ended up with $R_{L1} = 15k\Omega$ and $R_{L2} = 180k\Omega$. Through simulation these values also led to a filter that met specifications.

Next I had to focus on amplifying this filtered signal to a suitable level. From the

datasheet for the TLC2272CP op-amp [10], I was able to gauge the limitations of the op-amp. The maximum and minimum rail voltages are limited to +8V and -8V, and a maximum differential mode of 16V. The minimum input voltage is limited to the negative rail voltage -0.3V. For this application I will either use 5V or 3.3V for the positive rail and ground for the other, so I am well within the limits of the op-amp.

By running some tests, I was able to determine that the input voltage of my op-amp from my filter would be at a maximum around 550mV after the 5V pulses were filtered down. We also know from the ESP32 specifications, that the maximum allowable output of the op-amp is 3.3V. Using this information, I decided to use a rail voltage of 3.3V, which would hard limit my op-amp output to this value. As I need to amplify to a maximum of higher than 3V, I would need a gain $G = \frac{3.2}{0.55} \approx 6$. Using equation 2.4 for the gain of an active low pass filter with amplification, and the required gain of about $G \approx 6$, I chose $R_2 = 100k\Omega$ and then calculated for $R_1 = 20.7k\Omega$. I decided to use a potentiometer for my R_1 as this way I could change my gain continuously if my theoretical values were not suitable to a practical environment. Luckily this was not the case and my R_1 remained at $20.7k\Omega$

$$G = 1 + \frac{R_2}{R_1} \quad (2.4)$$

We know from basic electronic principles that a circuit with higher resistance will allow less charge to flow through it, thus I decided to use very high resistance values to provide a high impedance to charge flow and thus current, which ensured my circuit used as little current as possible.

2.3. Digital to Analogue Converter

As per [1], we can make a Binary Weighted DAC with an Inverting Summing Amplifier Circuit. In order to reach some specifications, I also decided that it would be best to add a DC offset to the op-amp.

Once again, from the datasheet of the TLC2272CP op-amp, [10], we can take the op-amp's limits into consideration. Some of these would include the maximum and minimum rail voltages at 8V, the maximum differential mode of 16V and the minimum input voltage being limited to $V_{ss} - 0.3V$. The digital input voltages will be 3.3V TTL signals (Anything from 0V to 0.8V representing a low input, and anything from 2V to 3.3V representing a high input).

The maximum allowable output voltage of the circuit is limited to 3.3V. I will use a potentiometer at both my gain stage and offset stage so that I can easily adjust the offset or gain of my op-amp if my theoretical calculations are not suitable to a practical environment.

I was able to find resistor values by making use of equations 2.5 and 2.6. I first chose values $R_f = 10k\Omega$ and $R_7 = 100k\Omega$. The latter choice was made for lower current draw. Knowing that the output needs to be 3.3V or less when all the inputs are 0, means I needed an offset

voltage of 3.3V. From there I was able to determine $R_8 = 51.5k\Omega$. Now, when all inputs are high, we need an output voltage of 0V. This leads us to choose values for R_5, R_4, R_2, R_1 in equation 2.5 that would give us our desired output when all inputs are high.

$$V_{out} = -\left(\frac{R_f}{R_5} * V_{b3} + \frac{R_f}{R_4} * V_{b2} + \frac{R_f}{R_2} * V_{b1} + \frac{R_f}{R_1} * V_{b0}\right) \quad (2.5)$$

$$V_{Offset} = 5\left(\frac{R_7}{R_7 + R_8}\right) \quad (2.6)$$

figure 2.4 shows the completed circuit diagram for the design. In section 4.2 see figure 4.2 for the physical implementation.

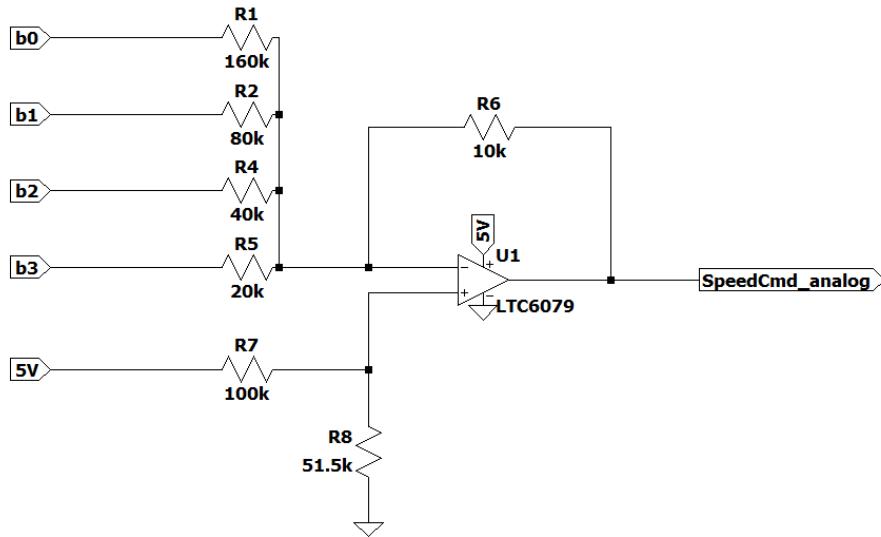


Figure 2.4: Digital to Analogue Converter circuit Diagram

2.4. Motor control signal and driver

For the motor control signal and driver, we will build the circuit in two stages. Firstly, a control stage that will take the torque and range sensor measurements as inputs and decide whether or not the motor should turn the wheel, and if it does, how quickly. The second stage will be an emitter follower, that will act as a buffer between the motor control stage and the motor, which will provide a layer of protection between the two.

2.4.1. Motor Control

The range sensor allows us to do object detection, and depending on how far away an object is from the front of the car, will provide us with a voltage input from 0 to 3.3V. When an object is near, the motor output needs to be low, and vice versa for when an object is far away or no object is detected.

The torque on the other hand is given as direct input to the car by the user. When the output of the torque is low the motor output is low and vice versa for when the torque input is high. As per 2.3, we know that a low torque can be measured as a 3V output and a high torque value is at 0V.

Since these are the only two inputs we are taking into consideration for the motor control, this is easily implemented with a differential operational amplifier. This can be shown by pseudo values in table 2.1 to hopefully provide a greater description of how this will be achieved.

Range	Torque	Differential Output Voltage
Far (High voltage)	High (Low Voltage)	High-Low = High
Close (Low voltage)	High (Low Voltage)	Low-Low = Low
Far (High voltage)	Low (High Voltage)	High-High = Low
Close (Low voltage)	Low (High Voltage)	Low-High = Low

Table 2.1: Differential Output Voltage when Range sense and Torque provided as inputs

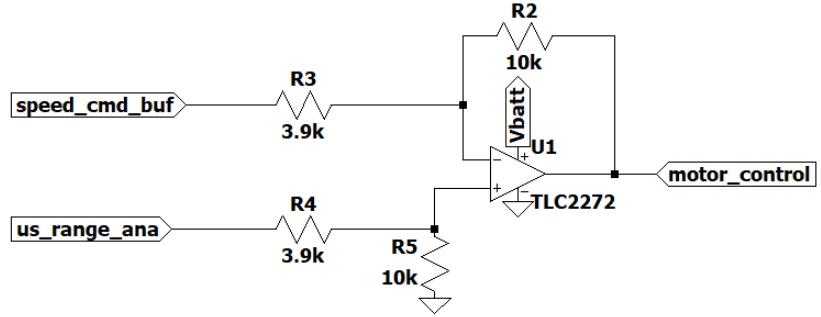
From the last row of the table we can see that we have a problem. When we subtract the high voltage from the low voltage, we will inevitably get a negative voltage output, which we don't want. To counteract this, we will connect the negative op-amp rail to ground, clamping the lowest output voltage to 0V.

The final design for this circuit is shown in figure 2.5a. By choosing $R_2 = R_5$ and $R_3 = R_4$ the transfer function becomes equation 2.7 as per [2]. By choosing $R_2 = 10k\Omega$ and designing for a gain of around $G = \frac{6.9}{3.3} = 2.1$ to get an output of higher than 6.2V for a differential input of higher than 3V, we get $R_3 = R_4 = 3900\Omega$

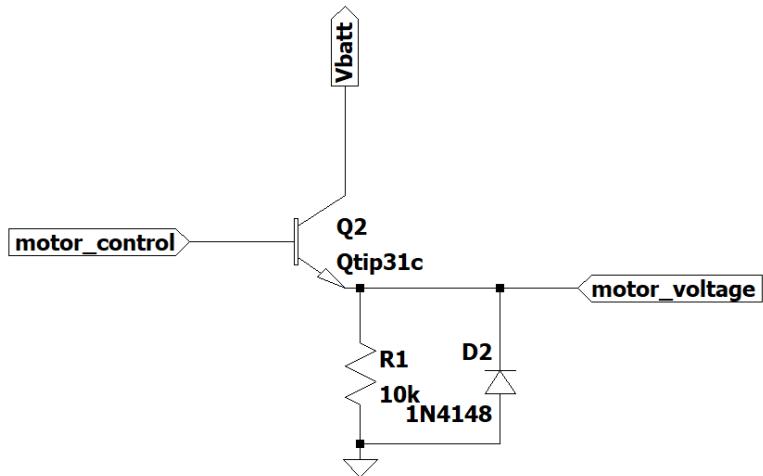
$$V_{out} = \frac{R_2}{R_3}(V_{rangeSens} - V_{torque}) \quad (2.7)$$

2.4.2. Emitter Follower Buffer

This stage acts as a buffer with a high input impedance, low output impedance and an approximately unity gain. We also include a diode for Back EMF protection for the control circuit. In general the buffer provides an extra layer of protection between the control circuit and the actual motor, so that if anything goes wrong, the one does not break the other. We choose a value of $R = 10k\Omega$ for the emitter resistor. The full circuit diagram is shown in figure 2.5b



(a) Motor Control Circuit



(b) Emitter Follower Buffer

2.5. Digital left wheel control

2.5.1. Current Sensing

The design for the current sensor for the left wheel is for the most part the same as the design for the current sensor for the right wheel, as discussed in section 2.1. However, this current sensing circuit will follow a low-side MOSFET switch and thus we have to consider the noise that will be introduced into the circuit due to the switching frequency. The design of the current sensor is shown in figure 2.2

The MOSFET switching frequency will be around the range of 1kHz, meaning that the first current sensing circuit design will work perfectly here as well, because it has been designed with a cut off frequency of 20Hz, with an additional RC filter stage that cuts off at 150Hz. This means that the circuit will filter out most of the noise introduced due to the MOSFET switching frequency at 1kHz.

We will be using a difference amplifier to measure the current. The input voltage will be near zero (around 10mV) and we want a max output voltage in the range of 3V - 3.3V.

For this we need to find a gain of about 310. using the transfer function equation 2.1, and selecting $R_2 = 100k\Omega$, we can obtain $R_1 = 270\Omega$. By using equation 2.2, and designing for a cut off frequency of 20Hz, we were able to obtain $C_1 = C_2 = 82nF$. For our additional filter stage we once again used equation 2.2 to design for a cut off frequency of 150Hz and obtained $R_6 = 15k\Omega$ when choosing $C_3 = 82nF$, the same value as the rest of the circuit capacitors. By using equation 2.3, we ensured the 90% response time remained below 100ms.

2.5.2. Low-side switch

For the design of the low-side switch, the first and most important design decision we make is deciding which MOSFET to use for the low side switch. Ultimately, I decided to use the FQD13NO6L N channel MOSFET for reasons discussed in the following paragraphs.

Firstly, from testing we know that our Motor will draw anywhere up to 1.2A of current, thus our selected switch needs to be able to handle this. From the data sheet [15], we can see that the FQD13NO6L can handle up to 11A of drain current which is more than sufficient. The MOSFET will be switched on or off by a voltage provided from our ESP32, which will be digitally controlling the speed of the motor. We know that our ESP32 will provide a low voltage around 0V and a high voltage at 3.3V, thus we need the gate threshold voltage of the chosen MOSFET to be larger than 0 and smaller than 3.3V. The data sheet of the FQD13NO6L shows that it is perfectly capable of this with $V_{GS(th)min} = 1V$ and $V_{GS(th)max} = 2.5V$.

We also add a series resistor to the gate. When the MOSFET is turned on, the gate-source acts like a fully charged capacitor that doesn't allow any current to flow through it [16], but when it is in the process of turning on, the gate-source (capacitor) is charging and thus allows current to flow through it. To prevent too much current from being drawn from the ESP32 during this turn on phase, we connect a series resistor between the ESP32 output and the gate. I also had to keep in mind that the higher this resistor value, the slower the MOSFET switching frequency will be. Thus I decided to use a small 220Ω resistor. I also decided to use a $10k\Omega$ pull down resistor to ensure that the MOSFET gate terminal isn't left floating as the ESP32 is busy turning on.

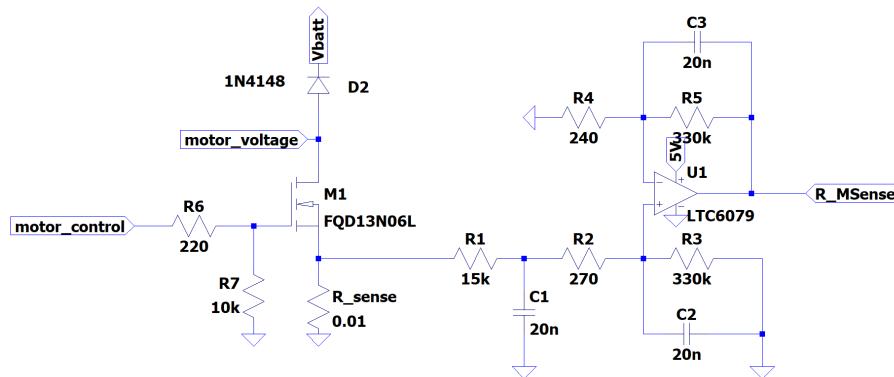


Figure 2.6: Hardware Implementation of the Digitally Controlled Left Wheel

2.5.3. Firmware (Range Sensor and PWM control)

For the firmware, I will be using two inputs, the speed nibble and the range sensor. From these two inputs I will be outputting a pulse, whose duty cycle will determine the length that the low side switch on the left wheel is on for, and hence control the speed of the left wheel.

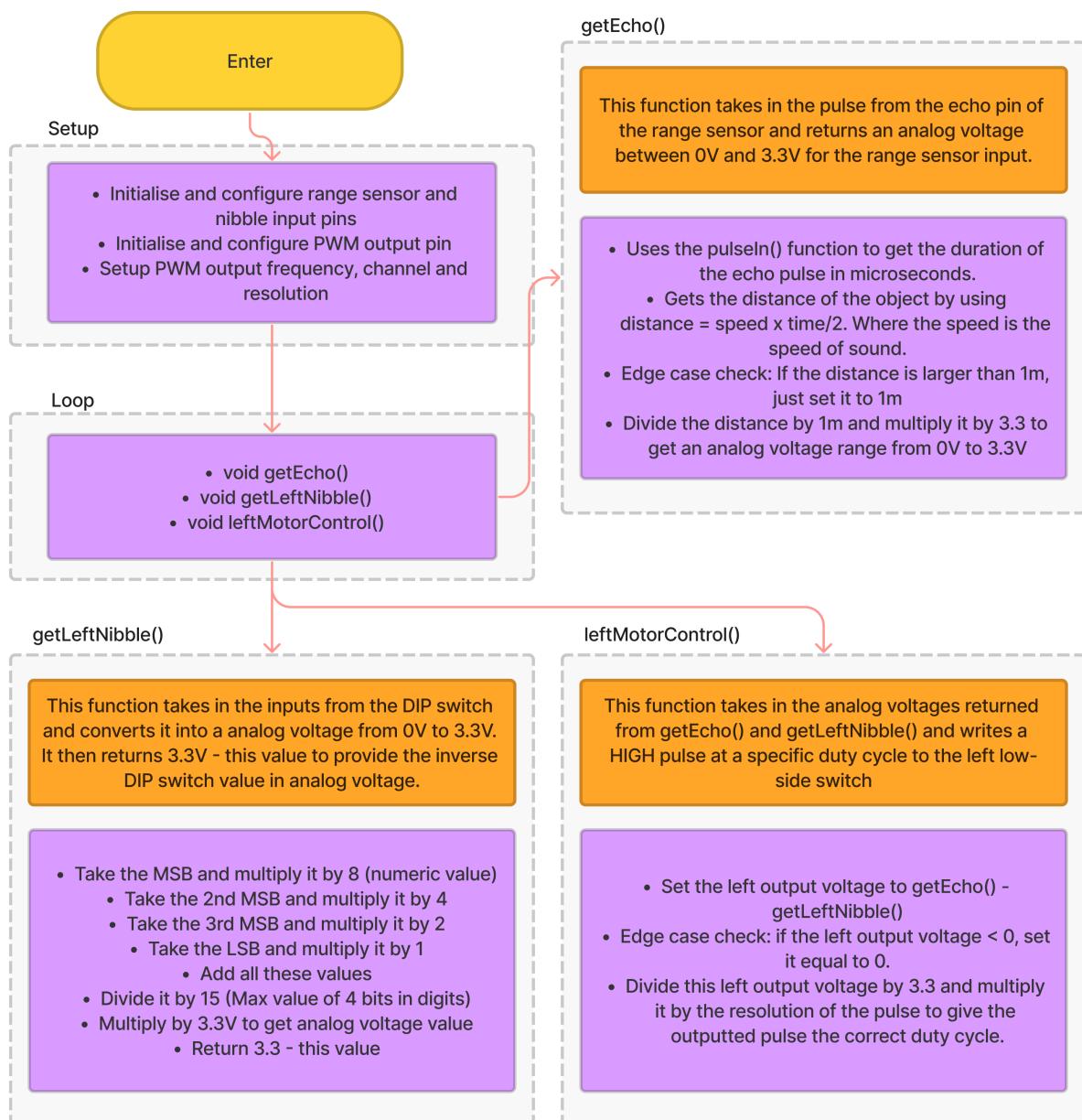


Figure 2.7: Flow Diagram of Firmware Implementation to control the left motor digitally

2.6. Battery

2.6.1. Battery Charger

The completed design for the battery charger is shown figure 2.8. Our goal is to make use of the LM317T voltage regulator [17] as well as MOSFET transistors to create a battery charger that can turn on or off with the use of a switch. More background of this process is discussed in section 1.7.

We'll be using the LM317T as both a constant voltage source, as well as a current limiter simultaneously. From [12] we know that a constant DC voltage of between 2.3 V/cell and 2.45 V/cell needs to be applied to the terminals of the battery when charging it, and this should roughly equal to the maximum voltage rating of the battery [18]. Since we are using the battery in cycle mode, we need 2.45 V/cell. From the data sheet for the battery [19] we know that the cycle use voltage is between 7.3V and 7.4V, thus using 2.45V/cell for 3 cells, we can apply a constant voltage output of 7.35V to the battery when charging.

A resistive voltage divider circuit is used at the output of the IC to produce an output voltage, following equation 2.8. The value of R_2 is chosen to be 220Ω for regulator stability. Since we want a value of 7.35V for V_{out} and we have R_2 , we can solve for R_1 by making use of equation 2.8 and get $R_1 = 1080\Omega \approx 1.1k\Omega$. We round up to take into account the diode we will add later.

As we've discussed, the LM317T also needs to act as a current limiter circuit, and for this we need a constant current source, which can be made by connecting a resistor, R_3 between the output of LM317T and its adjustment pin, as shown in figure 2.8. A requirement for the circuit is to limit the current into the battery to 0.1C. Given that the battery operates at a nominal capacity of 7 AH, we then have that $C = 7$ and hence we need to limit the current into the battery to $700mA$.

When there is a constant demand of current at the output of the IC, we will have a 1.25V at the adjustment pin. Taking the voltage division into account, this means we will have a 6.9V voltage drop across R_3 . To determine the value of R_3 becomes difficult as we are not exactly sure how much current the MOSFET-switch will draw. Instead of doing these calculations by hand, I've decided to first design the rest of the circuit and then use LTSpice simulation to determine for which value of R_3 we get a current draw of under $700mA$. This value ended up being $R_3 = 0.3\Omega$.

$$V_{out} = 1.25 * \left(1 + \frac{R_1}{R_2}\right) \quad (2.8)$$

The second stage of the battery charger circuit includes a dual-MOSFET switch. Two MOSFETs are used here so as to allow the voltage being switched and the digital input voltage to be different. This is of vital importance, as the voltage being switched is approximately 6.9V, while our digital input voltage comes from the ESP32 microcontroller which can not

produce an I/O output at a voltage that high. To prevent ringing while switching, we implement a small resistor ($R_4 = 10\Omega$) in series with the gate. R_5 and R_6 are pull down and pull up resistors respectively, which ensures that the charge is removed from the gate whenever the gate voltage is removed. I decided to choose $R_5 = R_6 = 10k\Omega$. Now when the ChargeOn signal is high, the battery charging circuit is applied to the battery terminal.

Capacitor C_2 is a decoupling capacitor which protects the battery charging circuit by isolating it from the voltage supply. Two Schottky diodes are used in the circuit and provide protection. Diode D_2 blocks any reverse current from the battery when the circuit is in an off state. The diode performs a similar task at the input of the circuit.

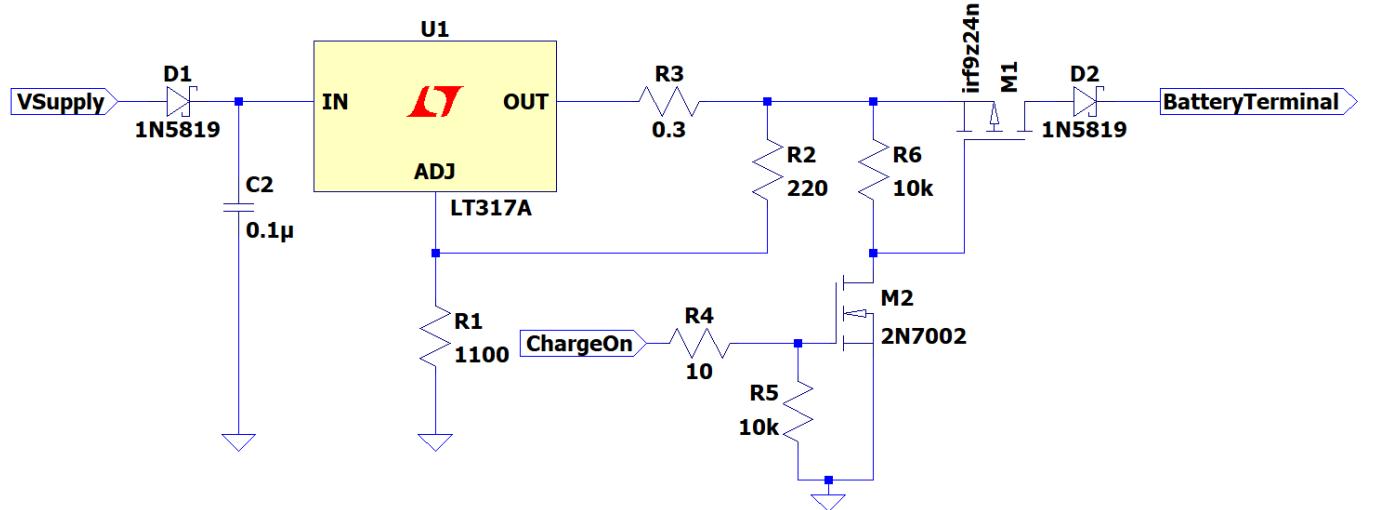


Figure 2.8: Battery Charger Circuit

2.6.2. Under-voltage Protection

We want to design an under-voltage protection circuit to turn off the circuit supply when the battery voltage is lower than 6.0V and only turns it on again when the battery voltage is over 6.1V. We can implement this in two main stages, a Schmitt Trigger [20] circuit as well as a high side PMOS switch which will control the battery supply to the motor.

The full design, as discussed, is shown in figure 2.9 below. The Schmitt trigger takes in the battery voltage and determines whether or not to output a digital high signal based on the given battery voltage. In this case, if our battery is under 6V, we output a low signal, and if it is over 6.1V, we output a high (5V) signal. In order to deal with this functionality, we provide 5V on the positive rail and ground on the negative rail. These rail values also mean that the battery voltage needs to be scaled down before being input into the op-amp due to input-rail relationship limitations.

I decided to design for a max input of 5V when the battery voltage is 7.2V. Using simple voltage division $V_{out} = V_{in} * \frac{R_3}{R_3+R_2}$, I was able to obtain $R_2 = 1k\Omega$ and $R_3 = 2.3k\Omega \approx 2.2k\Omega$. We also add a buffer stage after this voltage division to prevent current draw and hence

prevent the voltage divider from being biased and inaccurate.

To obtain the rest of the values of the Schmitt trigger, I used a Schmitt trigger calculator [21]. After voltage division, we have a high voltage threshold of $V_{th} = 6.1 * \frac{2.2}{3.2} = 4.2V$ and a low voltage threshold of $V_{tl} = 6 * \frac{2.2}{3.2} = 4.13V$. I can use any reference voltage lower than V_{tl} , thus I decide to use 3.3V, as I have already implemented a 3.3V regulator. I decided to limit the current input to 2mA, which yields realizable resistor values as well as keeps the current draw from the battery low. Our digital output high and lows were discussed in the first paragraph. From these inputs, I obtain $R_4 \approx 470\Omega$, $R_5 \approx 1800\Omega$ and $R_6 \approx 33k\Omega$.

We need to use a high side switch to cut off the supply to the motors, as a low side switch would effect our current sensor 2.1 readings. We can implement this with a NMOS in series with a high side PMOS switch. When we have a low output from the Schmitt trigger, our NMOS is off and thus the NMOS output is left floating. Hence the gate of the PMOS is pulled up to the battery voltage and $V_{PGS} \approx 0V$ and hence the PMOS is off, and the battery can't supply the motor, as required. When the Schmitt trigger output is high, we have an output from the NMOS of about $V_{NDS} = 5V - V_{NGS(on)}$ and hence for the PMOS switch, we'd have $V_{PDS} = V_{NDS} - V_{bat} < V_{PGS(th)}$ which would turn the PMOS on, meaning the battery would supply the motor as required.

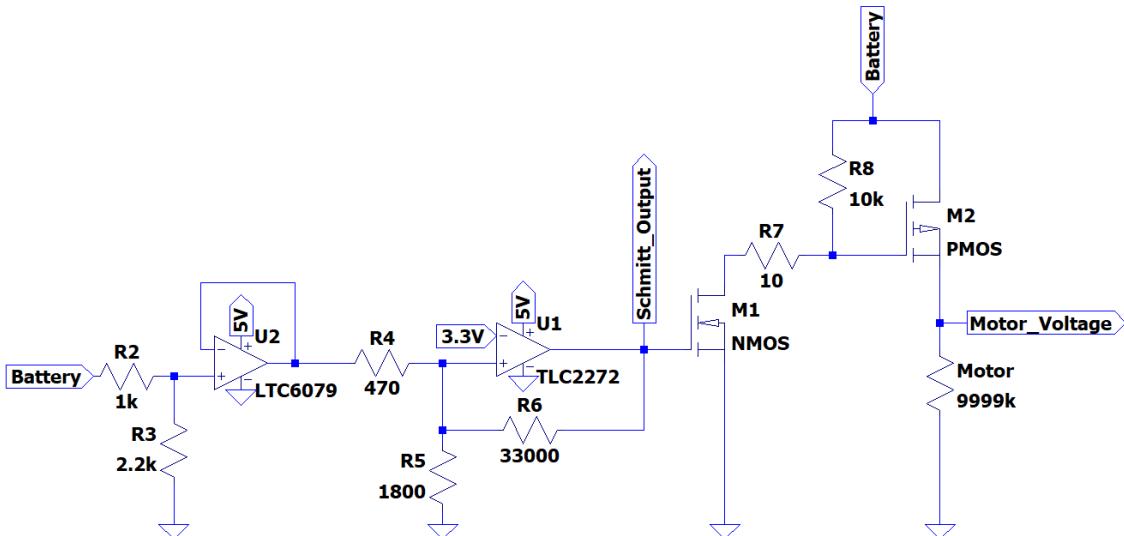


Figure 2.9: Schmitt Trigger Under-Voltage Protection Circuit

2.6.3. Battery Voltage Signal Conditioning

Our aim for this design is to produce an analogue voltage that represents the voltage, such that the analogue voltage is in a range of 0.1V to 3.3V for a battery voltage range of 5V to 7.2V. Since this will be read by the ESP32, the output of this circuit can not be above 3.3V, which will already be satisfied if we correctly design the circuit to peak at 7.2V, which would be the peak battery voltage.

In order to implement this circuit, I decided to use a differential amplifier, which will shift our desired battery voltage range down, as well as allow us to apply a gain to the differential

input to achieve our desired 3.3V range. The complete design is shown in figure 2.10. The battery voltage input enters the non-inverting input but is first passed through a voltage divider to half the value. This is done due to positive op-amp rail only being 5V. There is also a 5V input at the inverting pin, but this value is also divided by two to match the other input's scaling. By choosing $R_1 = R_2$ and $R_3 = R_4$ we were able to use a similar form of equation 2.7 to obtain the values shown in the diagram.

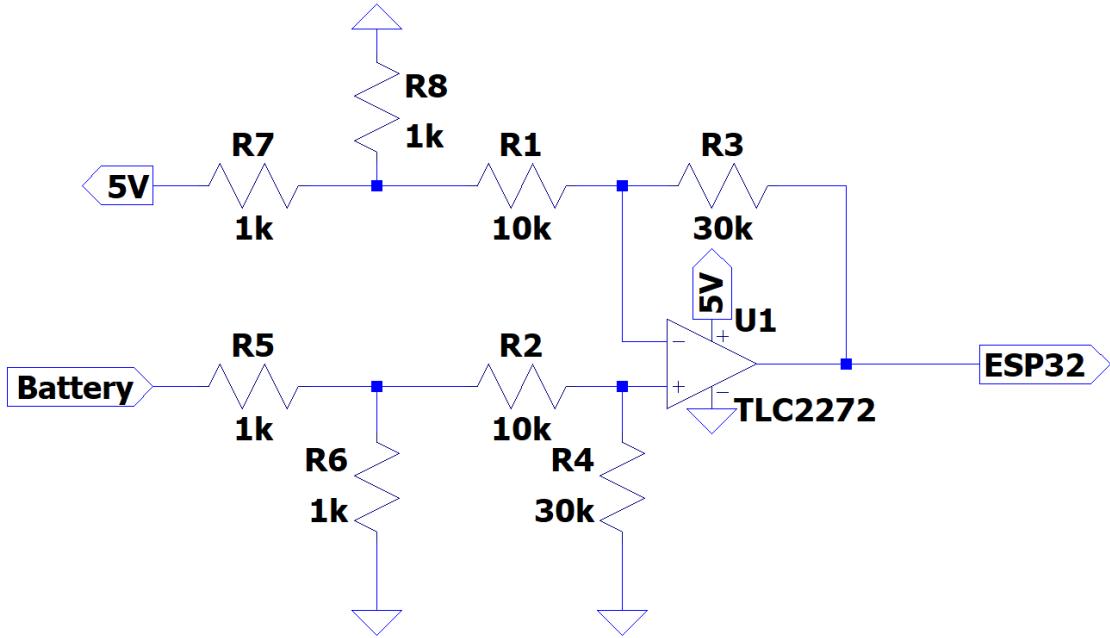


Figure 2.10: Battery Voltage Signal Conditioning Circuit

2.7. Bluetooth Connection Protocol and Firmware Design

In order to control the car, I will be making use of a Bluetooth connection to send instructions to the car, and receive sensor measurements from it. This, for the most part will be implemented in software.

Pin Number	Pin Usage
19	Right Wheel Speed Control Output MSB
18	Right Wheel Speed Control Output
5	Right Wheel Speed Control Output
17	Right Wheel Speed Control Output LSB
4	left Wheel Range Sensor Measurement
0	Right Wheel Range Sensor Measurement
12	Left Wheel Current Sensor Input
14	Right Wheel Current Sensor Input
27	Battery Voltage Signal

Table 2.2: ESP32 GPIO Pin Configurations

At a frequency of 1Hz, I will be displaying sensor data from the car as well as the instructions sent to it, thus it's important that the various sub-circuits are well organised

amongst the various GPIO pins on the ESP32 microcontroller. This organisation is shown in table 2.2. The flow chart shown in figure 2.11 provides the basic outline of the software implementation of the Bluetooth configuration.

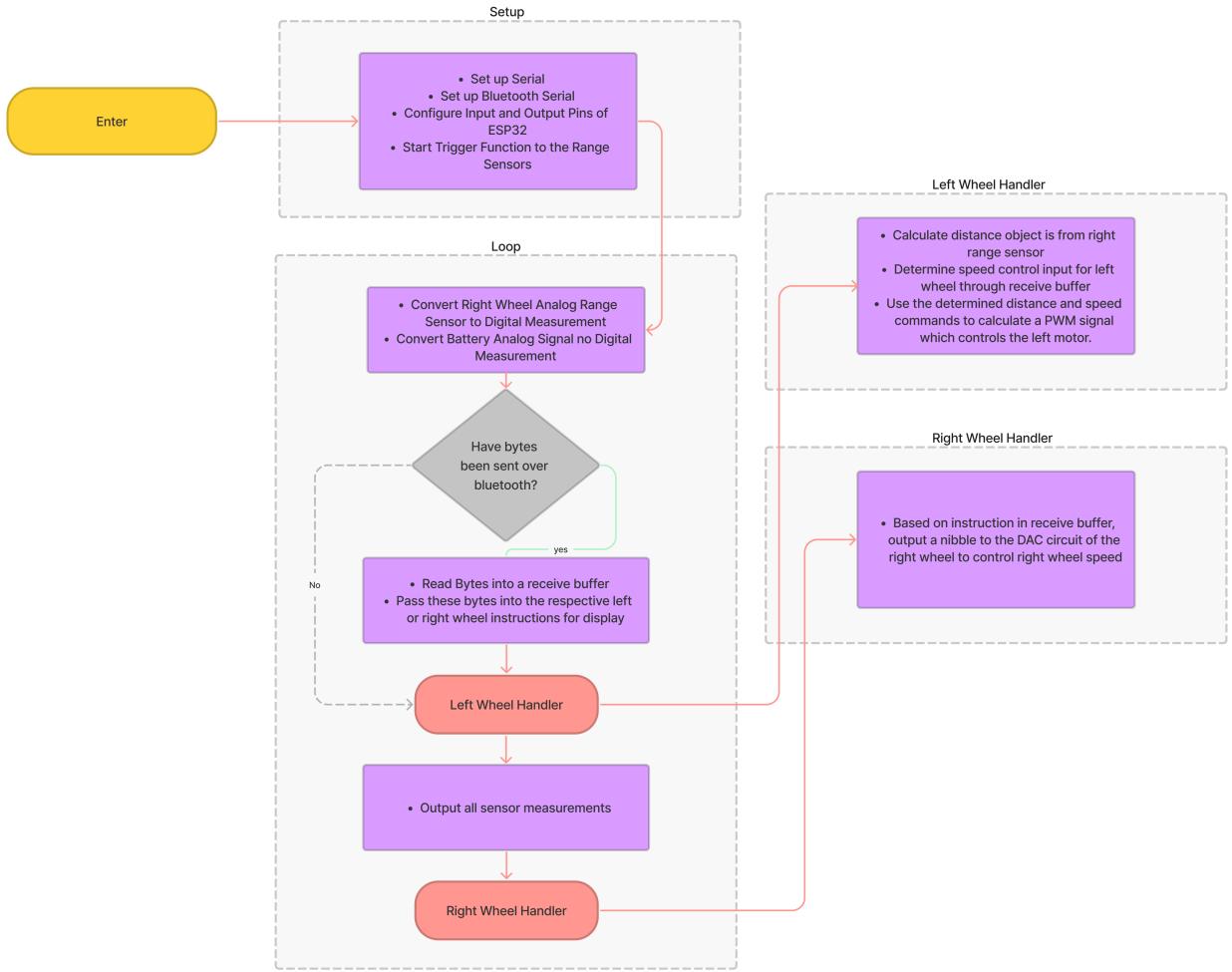


Figure 2.11: Flow diagram illustrating the process of instructing and sensing data over bluetooth

2.8. Graphical User Interface Design

For the implementation of the graphical user interface, I will be developing a simple web API application which will allow the user to control the car via Bluetooth remotely, in a user friendly manner. For the front end design (what the user will see), I first created a few mock ups in Figma, before implementing it in code using React java script and Tailwind CSS frameworks. I will be using axios to perform requests from the frontend to the API. I will be using Django (Python) and Django Rest Framework to handle the API requests that will be coming from the user on the front end, and to send requests via Bluetooth to the car. This is shown in better detail in figure 2.12

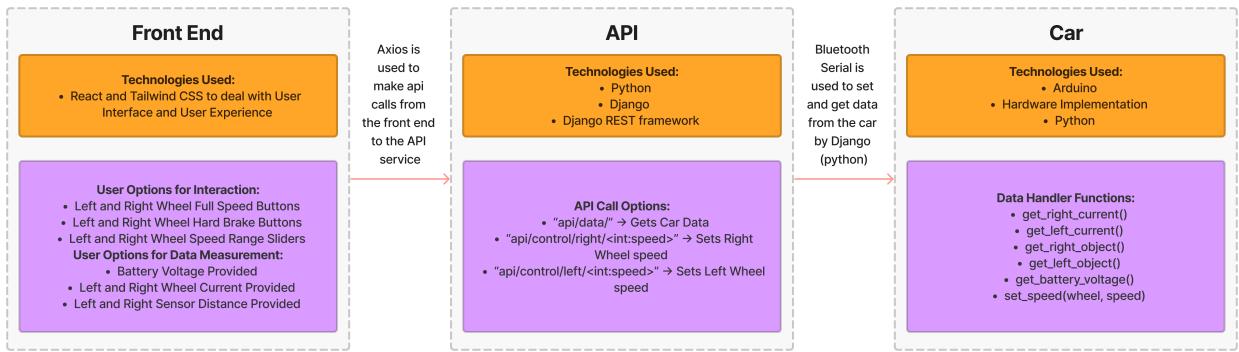


Figure 2.12: Flow Diagram illustrating the interaction between Front end, API and Car for GUI implementation

Chapter 3

Results

3.1. Current sensor

Table 3.1 shows the different measurements obtained for the current draw when testing the DC motor.

Condition	Current Draw (A)
Stall	1
Running	0.15
Slight Load	0.51 terminals

Table 3.1: DC motor currents under various conditions

Figure 3.1 shows the simulation output (green) vs the input (blue). We can see that with the step, the output reaches 643mV. This was done with an input current of 200mA, which is 1/5th of the max. By multiplying this output voltage by 5, we obtain 3.2V, which is in the output range we want for maximum current. We can also see how well the noise is reduced by the circuit compared to the input. This shows that our circuit requirements have been met.

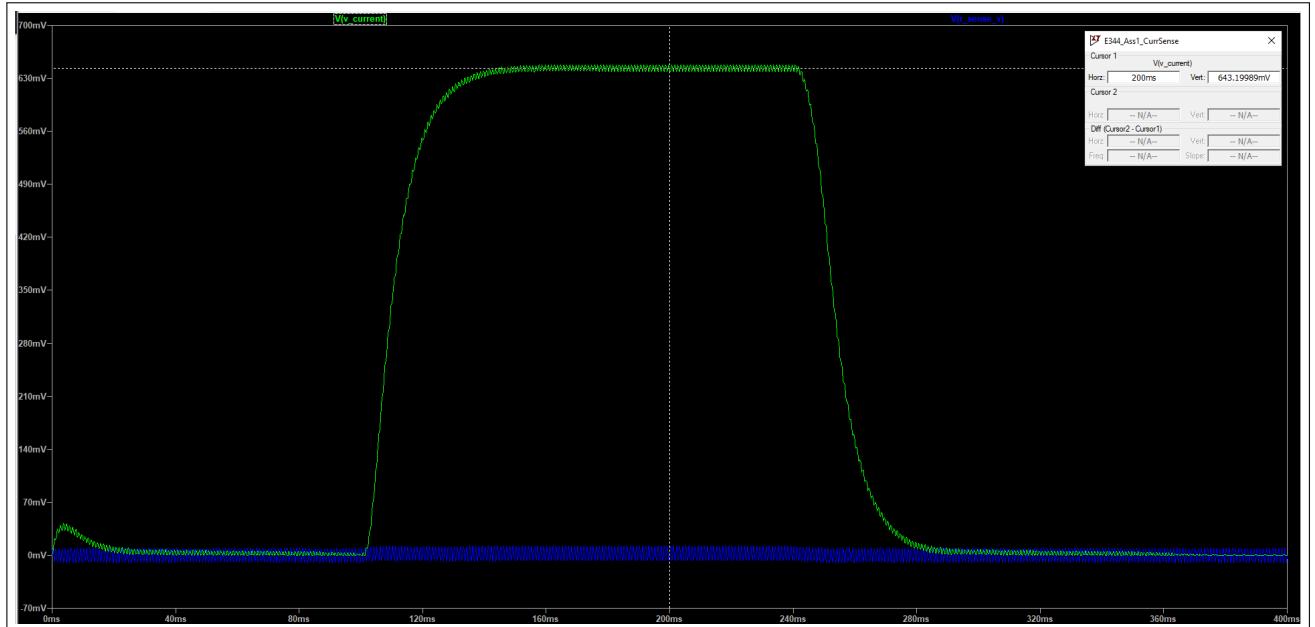


Figure 3.1: Simulation output vs input voltage

As seen from figure 3.2, by measuring across the sense resistor when a 200mA input is provided, the circuit only draws .5 uA. Which is much less than the 150uA limit the circuit has.

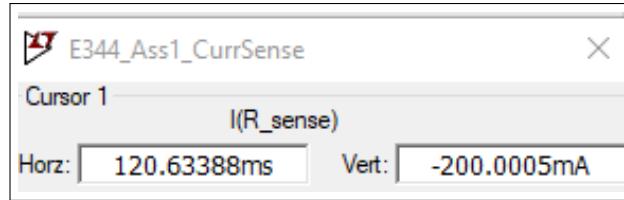


Figure 3.2: Circuit Current Draw at 200mA input

Current Sensor: Measured Results

While at stall and slight load, the current sensor provides sensible results that match my simulated circuit's output; however, at free run I have a major deviation: My current sensor provides a near 0V output at free run, which should be around 640mV according to my simulated results. The exact reason is still unknown and I am currently debugging and researching in order to find a solution. Possible reasons for deviation could be due to faulty or incorrect connections on the circuit, or a bad general design. As my circuit is meeting most of the requirements, it's difficult to blame the problem at hand on these exact reasons.

From table 3.2 we can see that the DC motor currents drawn were as expected.

Condition	Current Draw (A)
Stall	1.2
Running	0.16
Slight Load	0.3 - 0.4

Table 3.2: Measurements of DC motor currents under various conditions

In figure 3.3, in the oscilloscope readings you can clearly see the circuits response to various inputs and loads on the motor.

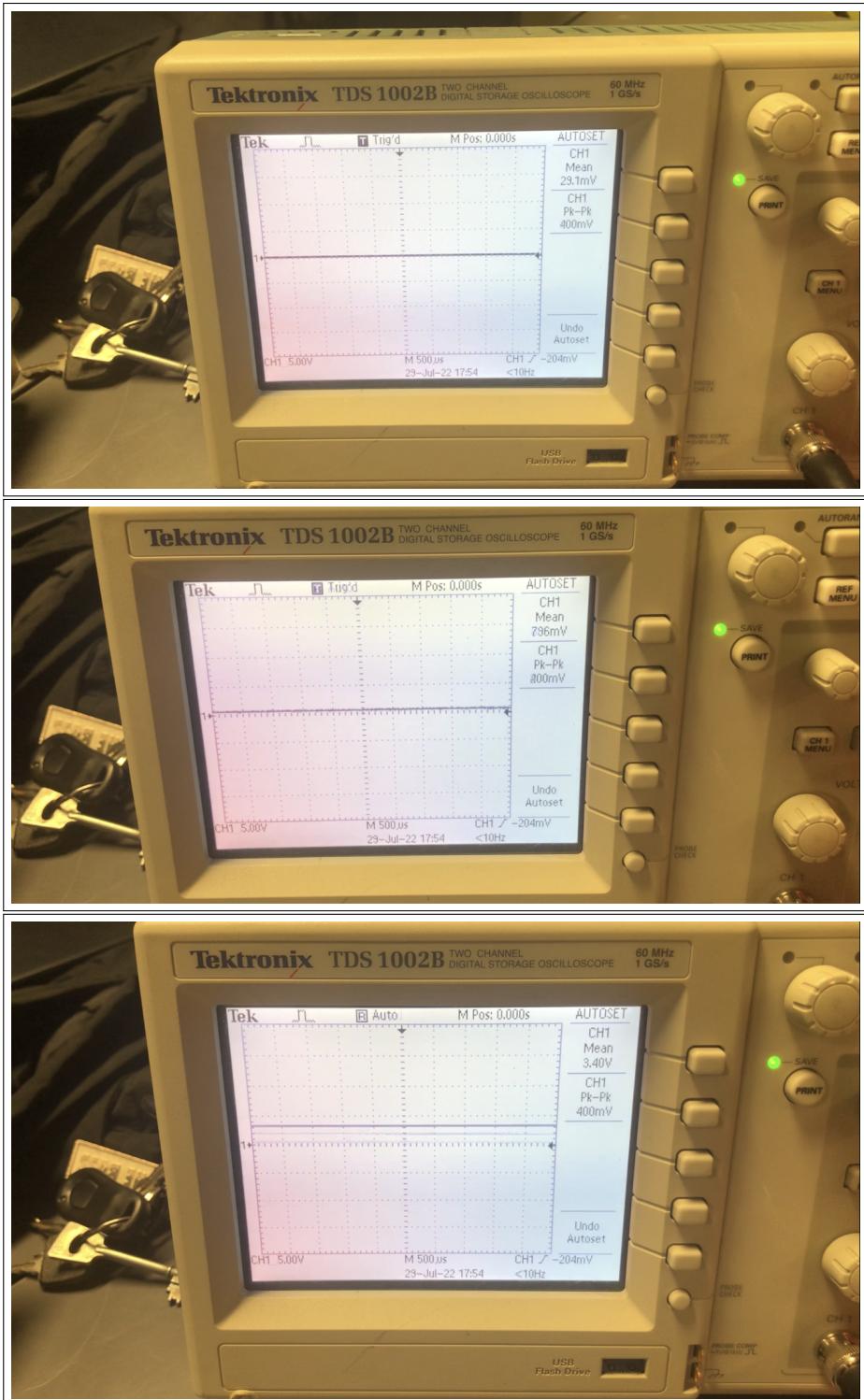


Figure 3.3: Current sensor circuit output voltage when motor is free running, slightly loaded or momentarily stalled respectively.

3.2. Analogue Range Sensor

Range Sensor: Simulation Results

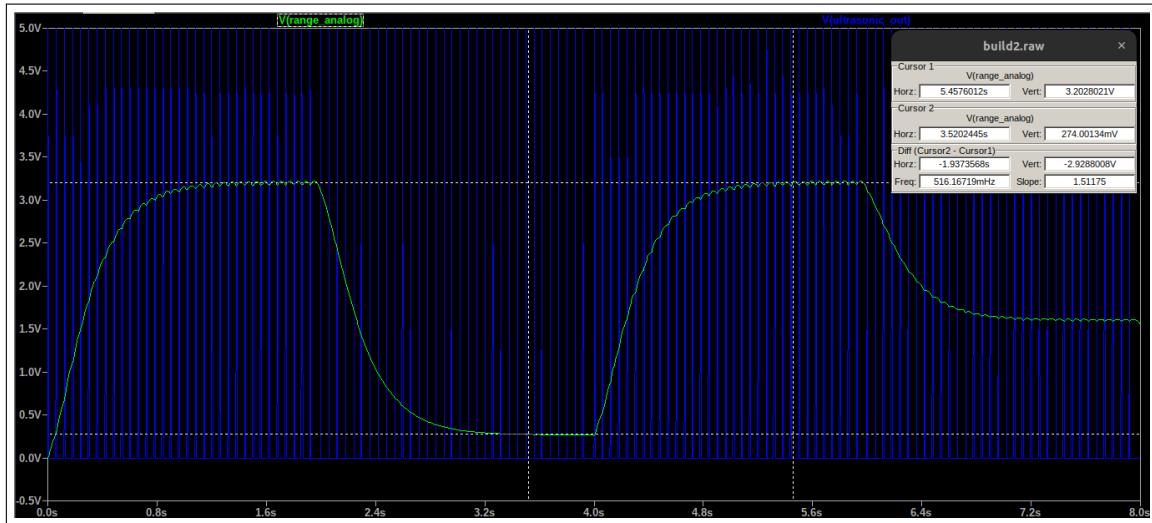


Figure 3.4: Step input vs Step response of analogue range sensor displaying met requirements of lower than 0.3V at low and higher than 3V at high ranges.

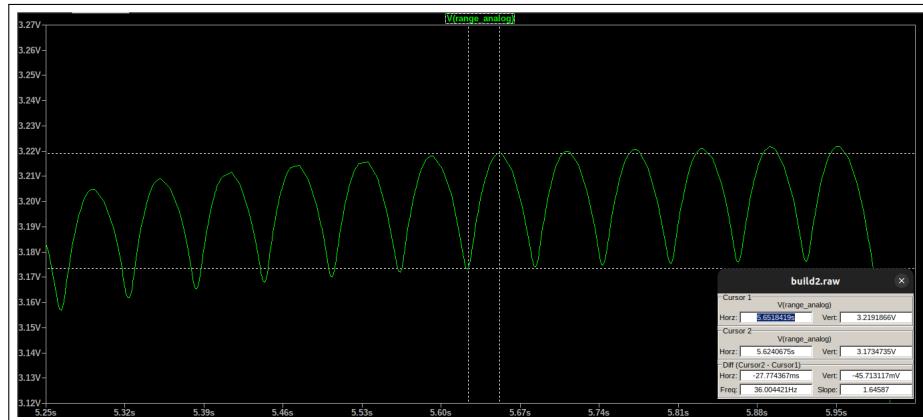


Figure 3.5: Noise level on analogue input is around 45mV

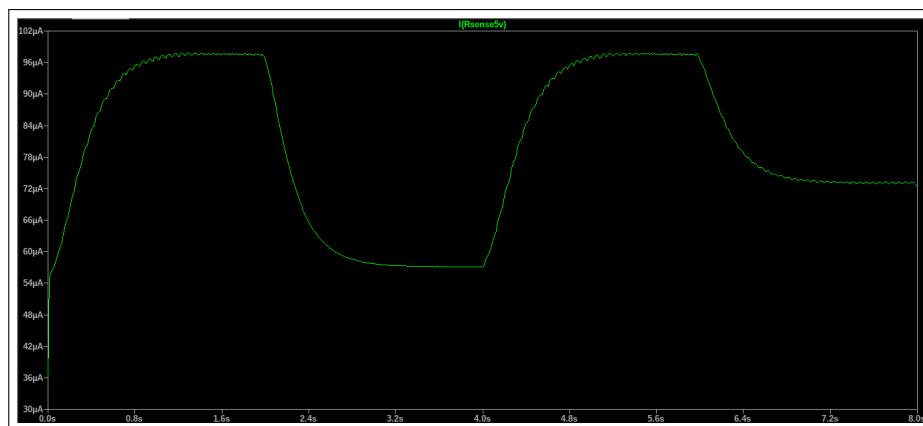


Figure 3.6: Current Draw of the analogue range sensor circuits remains lower than $100\mu A$

Range Sensor: Measured Results

Range input (cm)	Output Voltage (V)
1	0.114
15	0.606
30	1.08
1000	3.294
> 1000	3.3

Table 3.3: Range input vs Output

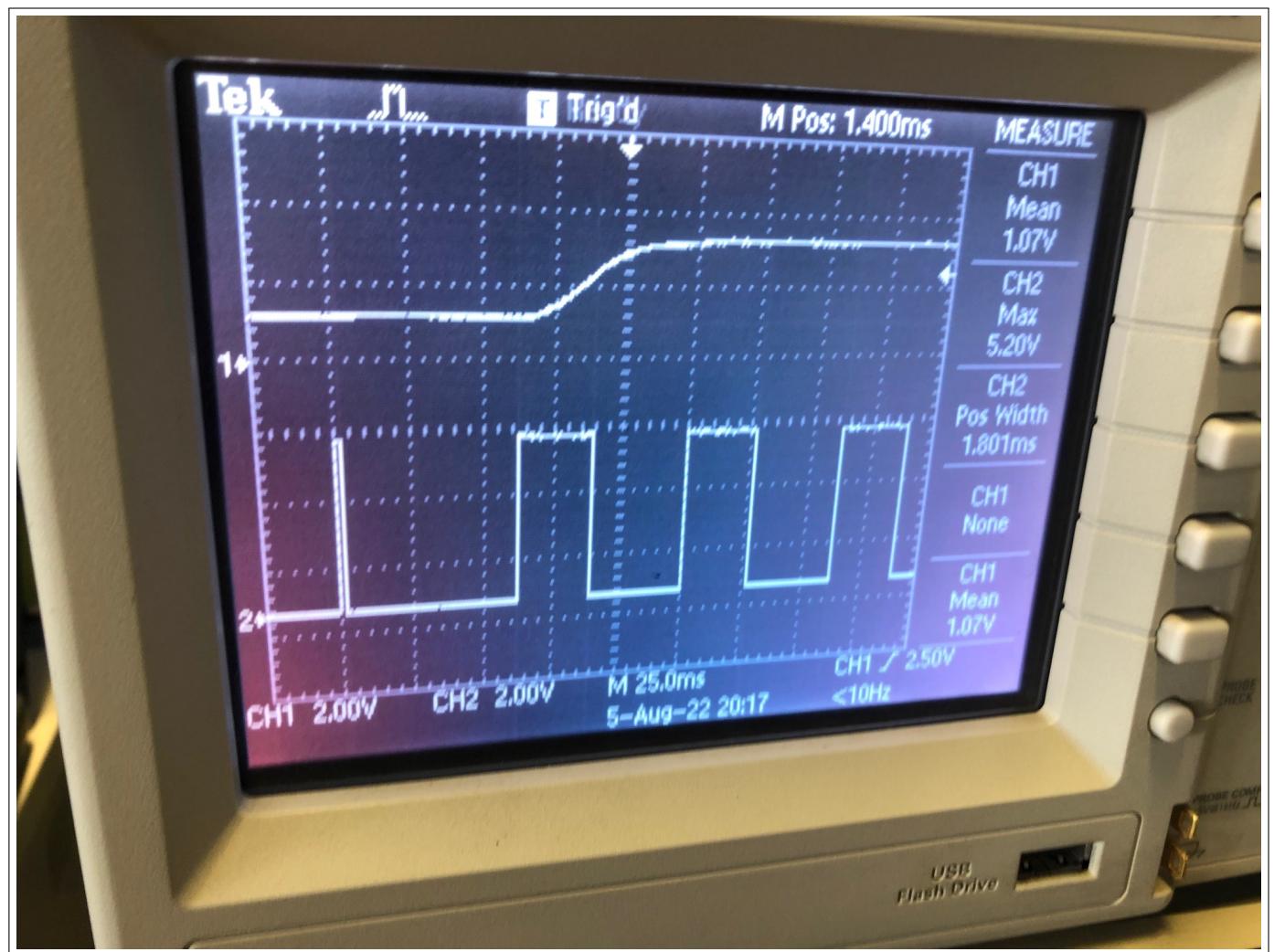


Figure 3.7: Step input vs Output Response of analogue range sensor

3.3. Digital to Analogue Converter

DAC: Simulation Results

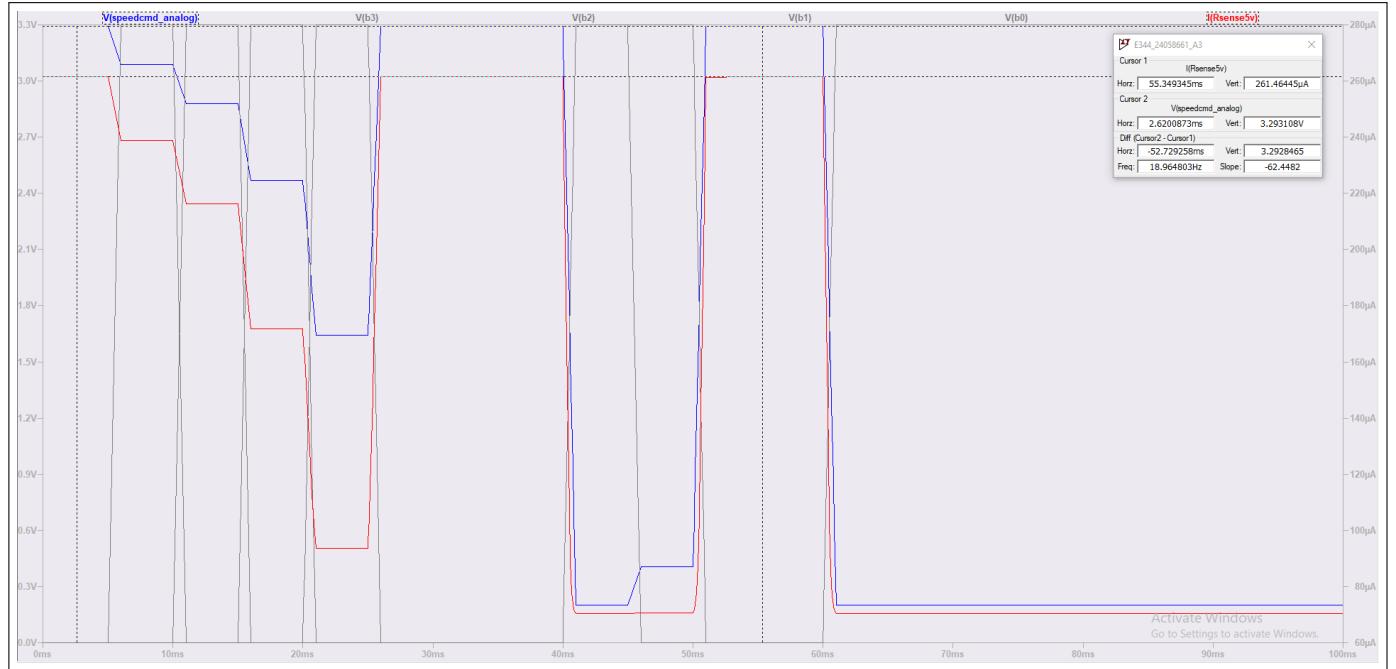


Figure 3.8: Simulation output (blue) vs input (grey) demonstrating requirement compliance as well as current draw (red) at a max of $260\mu A$

DAC: Measured Results

Nibble Input	Output Voltage (V)
0000	3.08
0001	2.89
1110	0.39
1111	0.20

Table 3.4: Measured output voltage of DAC converter vs nibble input via a DIP switch

3.4. Motor Control

Simulation

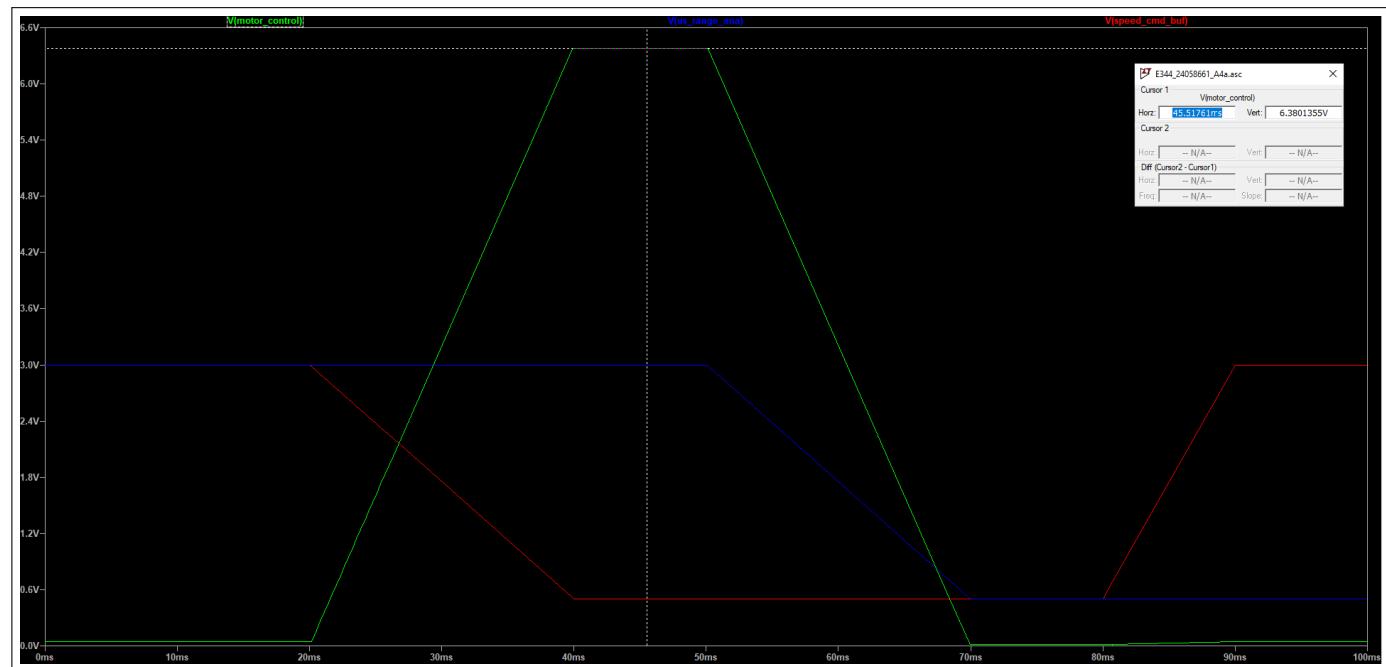


Figure 3.9: Simulated motor control output (green) vs Range Sensor (Blue) and Torque (red) input demonstrating requirement compliance

Measurements

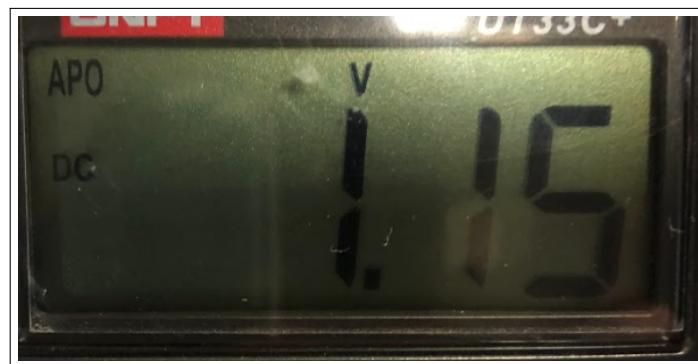


Figure 3.11: Voltage output of current sensing circuit when the wheel is turning freely

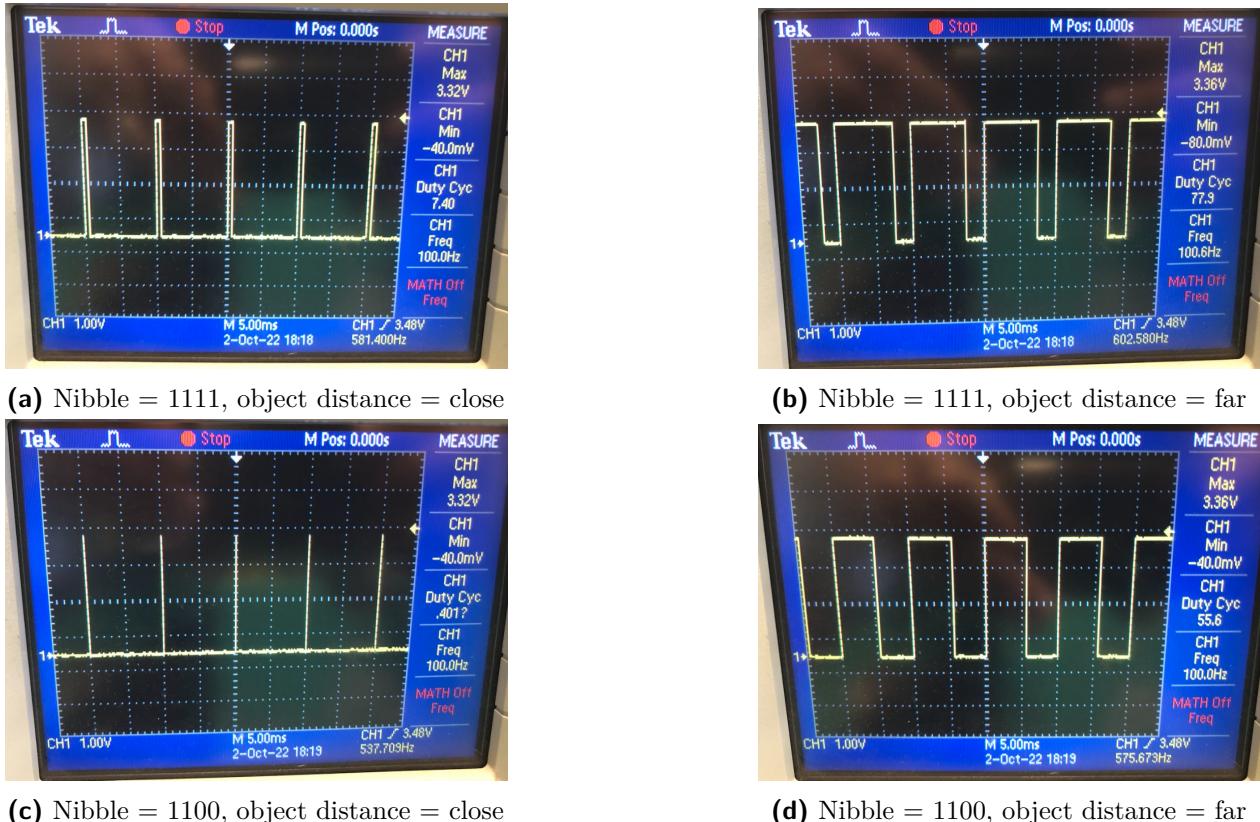


Figure 3.10: Measured results of left wheel control pulse entering switch at different conditions

3.5. Battery

3.5.1. Battery Charger

Simulation Results

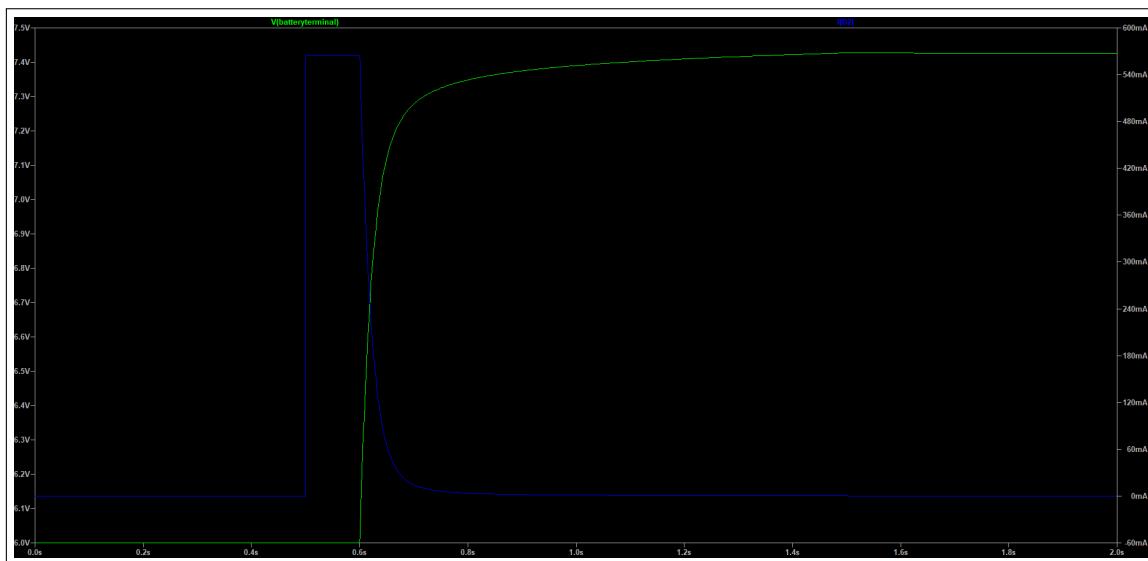


Figure 3.12: Battery Charger Circuit Output Voltage (green) and Output Current (blue)

From this, we can see that the voltage at the battery terminal peaks at about 7.42V, which is close enough to the 7.35V that we've designed for. The current delivered to the battery terminal peaks at 569mA, which is lower than the 0.1C limit that is required.

Measured Results



(a) Charger Output over $1k\Omega$ resistor



(b) Charger Output over 50Ω resistor

Figure 3.13: Charger output over a $1k\Omega$ and a 50Ω resistor respectively

3.5.2. Under-voltage Protection

Simulation Results

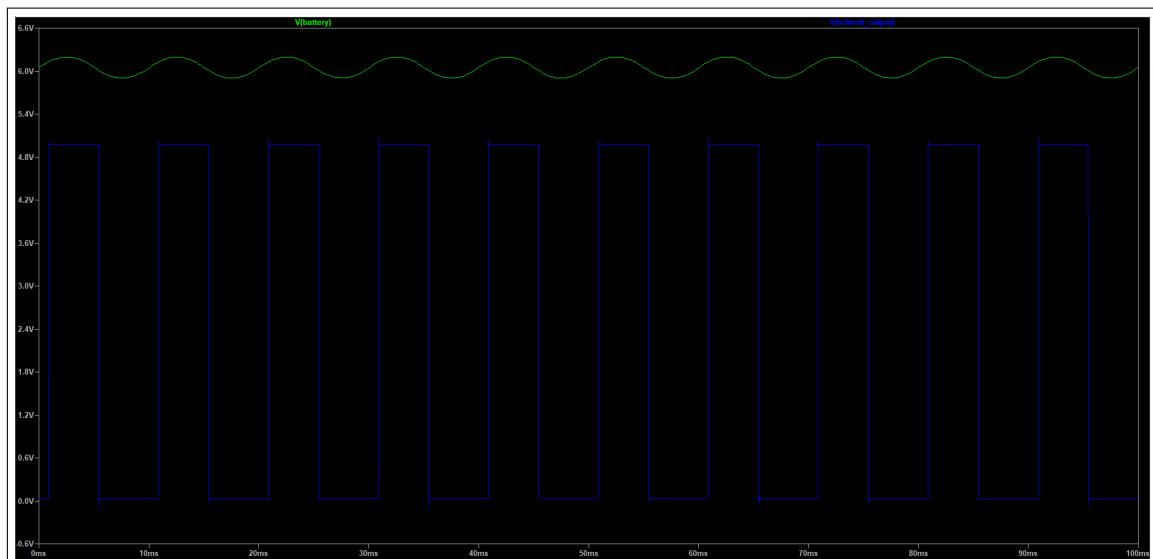


Figure 3.14: Simulated Schmitt Output (blue) vs Test Battery Voltage (Green)

We can see here that when the test battery voltage drops below 6V, then our Schmitt trigger outputs low, and when our test battery voltage goes above 6.1V, then our Shcmitt trigger outputs high. As designed

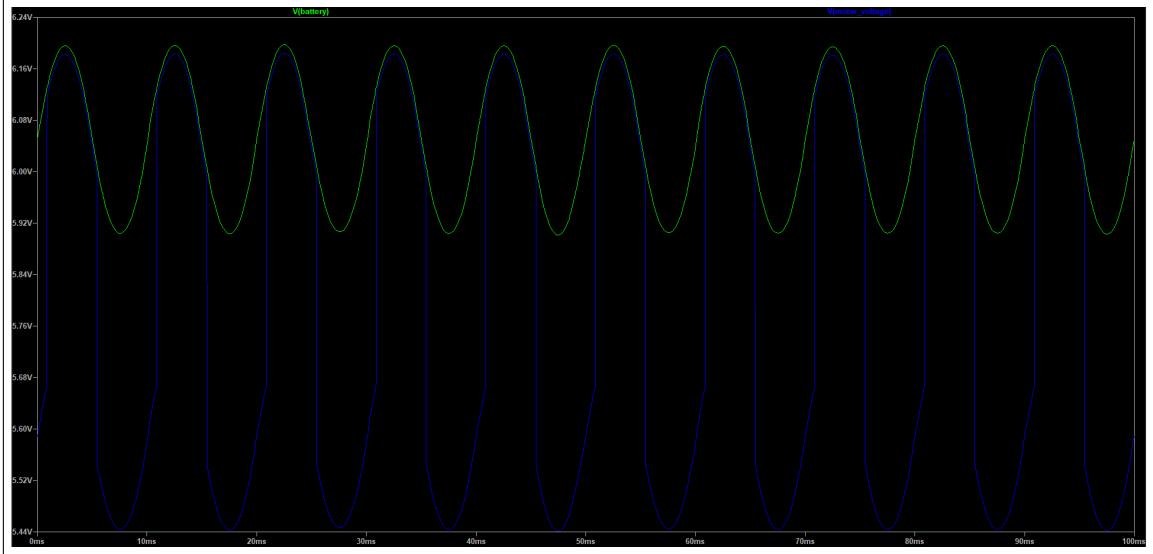


Figure 3.15: Simulated Motor Voltage Output (blue) vs Test Battery Voltage (Green)

Here we see that when our battery voltage is high (higher than 6.1V), so is our motor voltage as designed. However when the battery voltage goes low (lower than 6V), our motor voltage doesn't drop to 0 as we have wanted. This could be due to the incorrect NMOS and PMOS switches being used in the simulation or because of incorrect threshold voltage calculations.

Measured Results

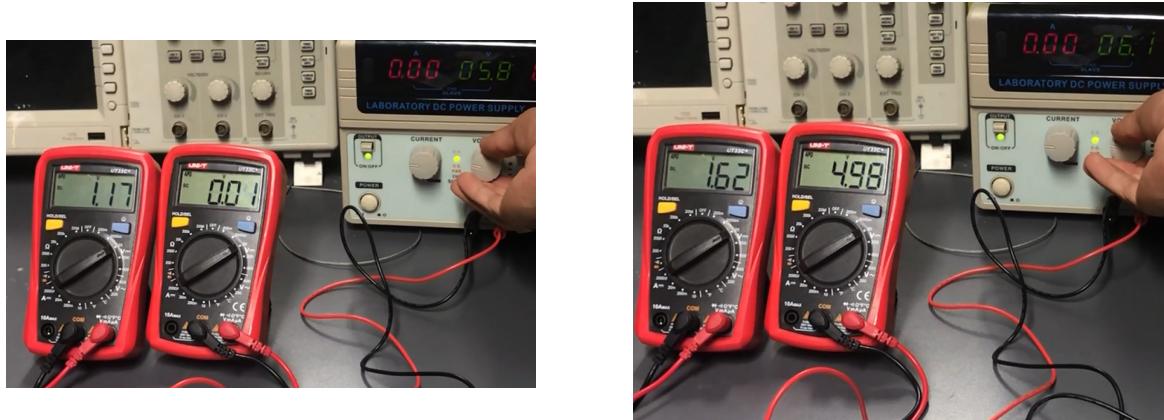


Figure 3.16: Signal Conditioning Output (Left Multimeter) and Schmitt Trigger Output (Right Multimeter), at a battery voltage of 5.8V and 6.1V respectively

3.5.3. Battery Voltage Signal Conditioning

Simulation Results

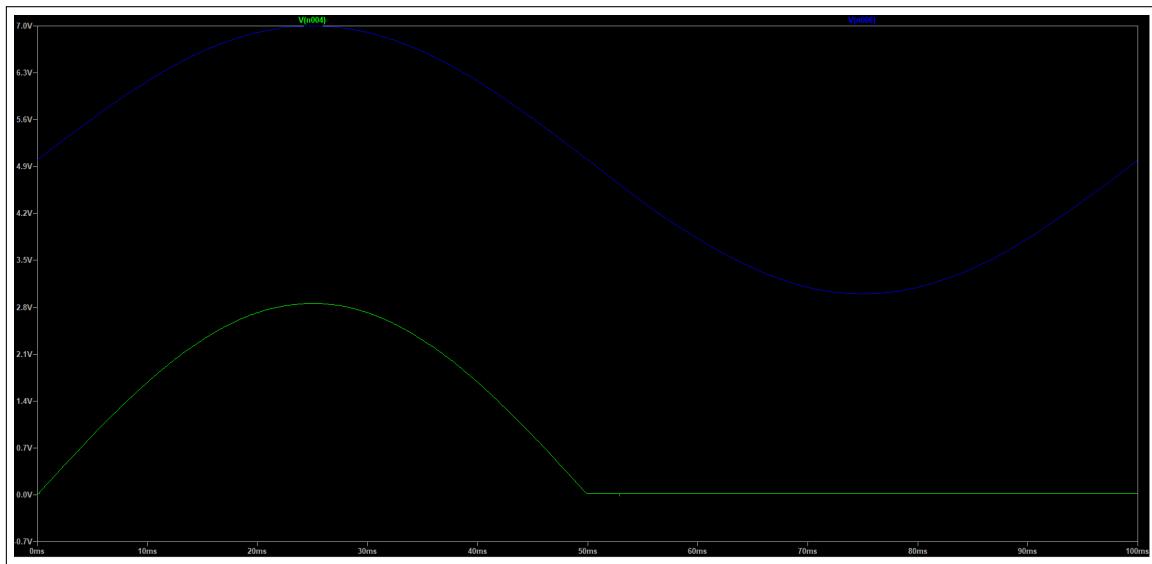


Figure 3.17: Battery Voltage Signal Conditioning Simulation Result with Circuit Output (Green) and Test input (Blue)

From this, we can see that when the blue test input (battery voltage) is between 5V and 7.2V, our green circuit output is between 0.1V and 3.3V as designed.

Measured Results

The measured results for the battery signal conditioning can be seen in figure 3.16 on the left multimeter at a battery voltage of 5.8V and 6.1V respectively. We can also see the battery signal conditioning output at a battery of 7V in figure ?? below



Figure 3.18: Signal Conditioning Output (on left multimeter) at a voltage of 7V

3.6. Bluetooth Connection Protocol and Firmware Results

To demonstrate the results of this section, I provided the measurement setup as seen in figure 2.11 as well as four tables under four different conditions to indicate the sent and received data over bluetooth, to and from the car. As you can see in the measurement setup, the data is monitored through the use of a serial monitor while the car is held floating above the ground to properly test performance. Keep in mind a low write to the left wheel is the equivalent of making the speed high and vice versa. A high right to the right wheel is the equivalent of making the speed high.

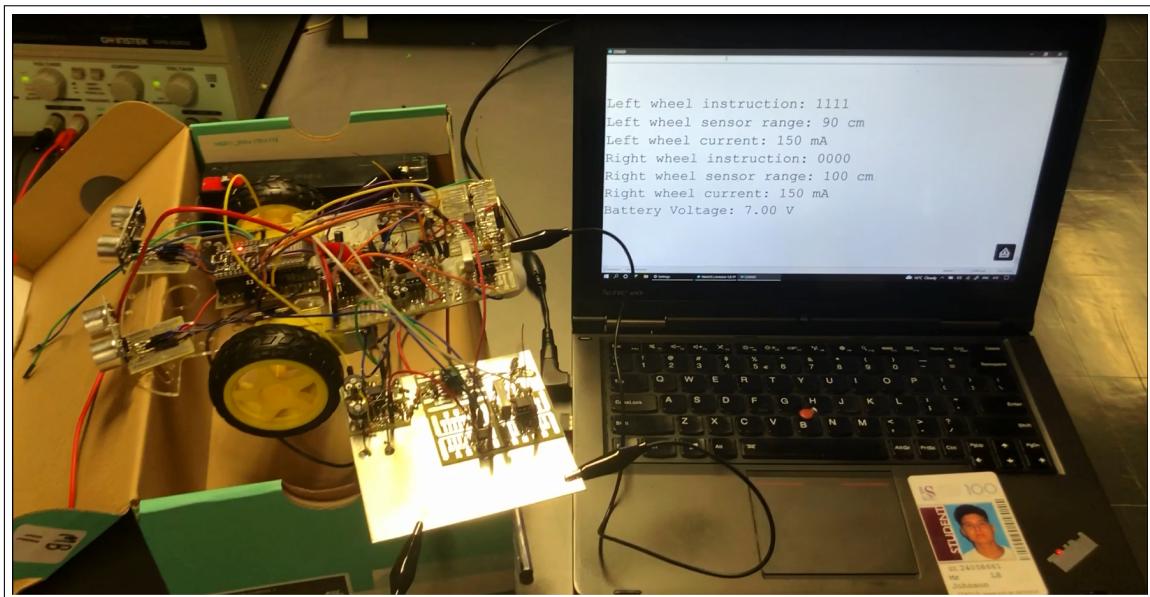


Figure 3.19: Measurement setup for bluetooth connection protocol and firmware

Parameter	Measured Data
Left Wheel Instruction	0000
Left Wheel Sensor Range	89cm
Left Wheel Current	150mA
Right Wheel Instruction	1111
Right Wheel Sensor Range	100cm
Right Wheel Current	150mA
Battery Voltage	7V

Table 3.5: Full Speed with No Objects Close By

Parameter	Measured Data
Left Wheel Instuction	0000
Left Wheel Sensor Range	89cm
Left Wheel Current	150mA
Right Wheel Instruction	0000
Right Wheel Sensor Range	100cm
Right Wheel Current	150mA
Battery Voltage	7V

Table 3.6: Hard turn right with no objects close by

Parameter	Measured Data
Left Wheel Instuction	0000
Left Wheel Sensor Range	5cm
Left Wheel Current	150mA
Right Wheel Instruction	1111
Right Wheel Sensor Range	89cm
Right Wheel Current	150mA
Battery Voltage	7V

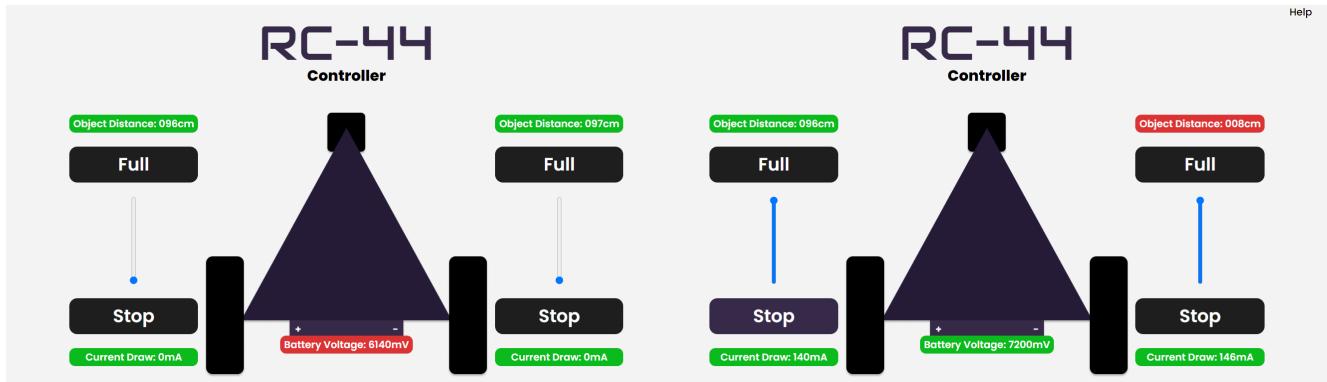
Table 3.7: Full speed with object near by left wheel sensor

Parameter	Measured Data
Left Wheel Instuction	0000
Left Wheel Sensor Range	90cm
Left Wheel Current	150mA
Right Wheel Instruction	1111
Right Wheel Sensor Range	76cm
Right Wheel Current	150mA
Battery Voltage	7V

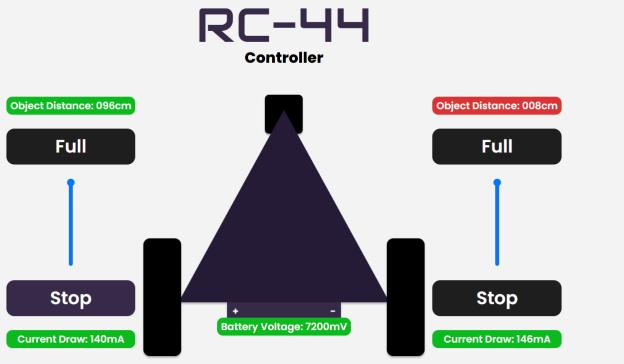
Table 3.8: Full speed with object near by right wheel sensor

Due to my current sensor not functioning correctly, I decided to hard code a value of 150mA, which is what it's supposed to be. Thus all the current results are not accurate depictions of the true current draw.

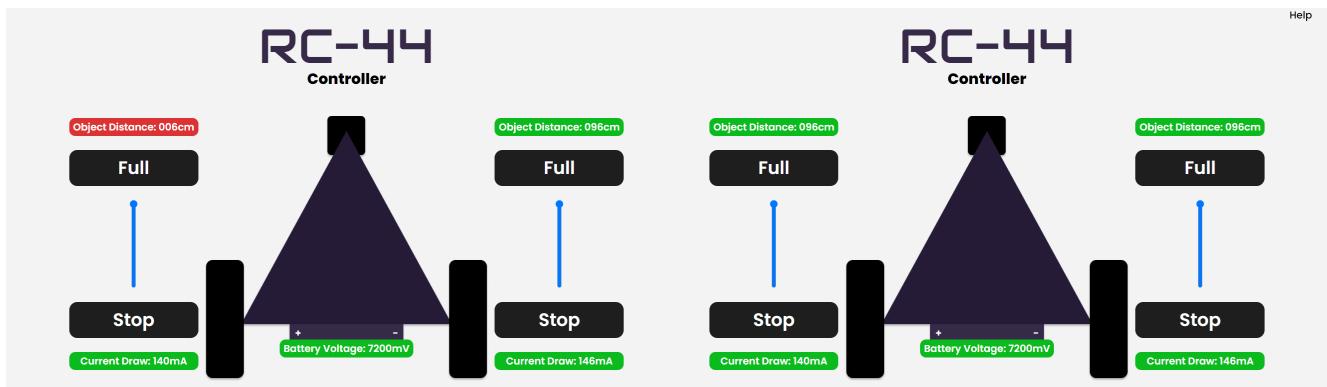
3.7. Graphical User Interface Results and Screenshots



(a) Low battery and Both Wheels Not Turning



(b) Objected near right side at full speed



(c) Objected near left side at full speed

(d) Full speed with all sensors ok

Figure 3.20: Graphical User Interface displaying measured sensor data as well as optional controls for a user

Chapter 4

Physical implementation

4.1. Current sensor



Figure 4.1: Current Sensing Circuit with labels

4.2. Digital to Analogue converter

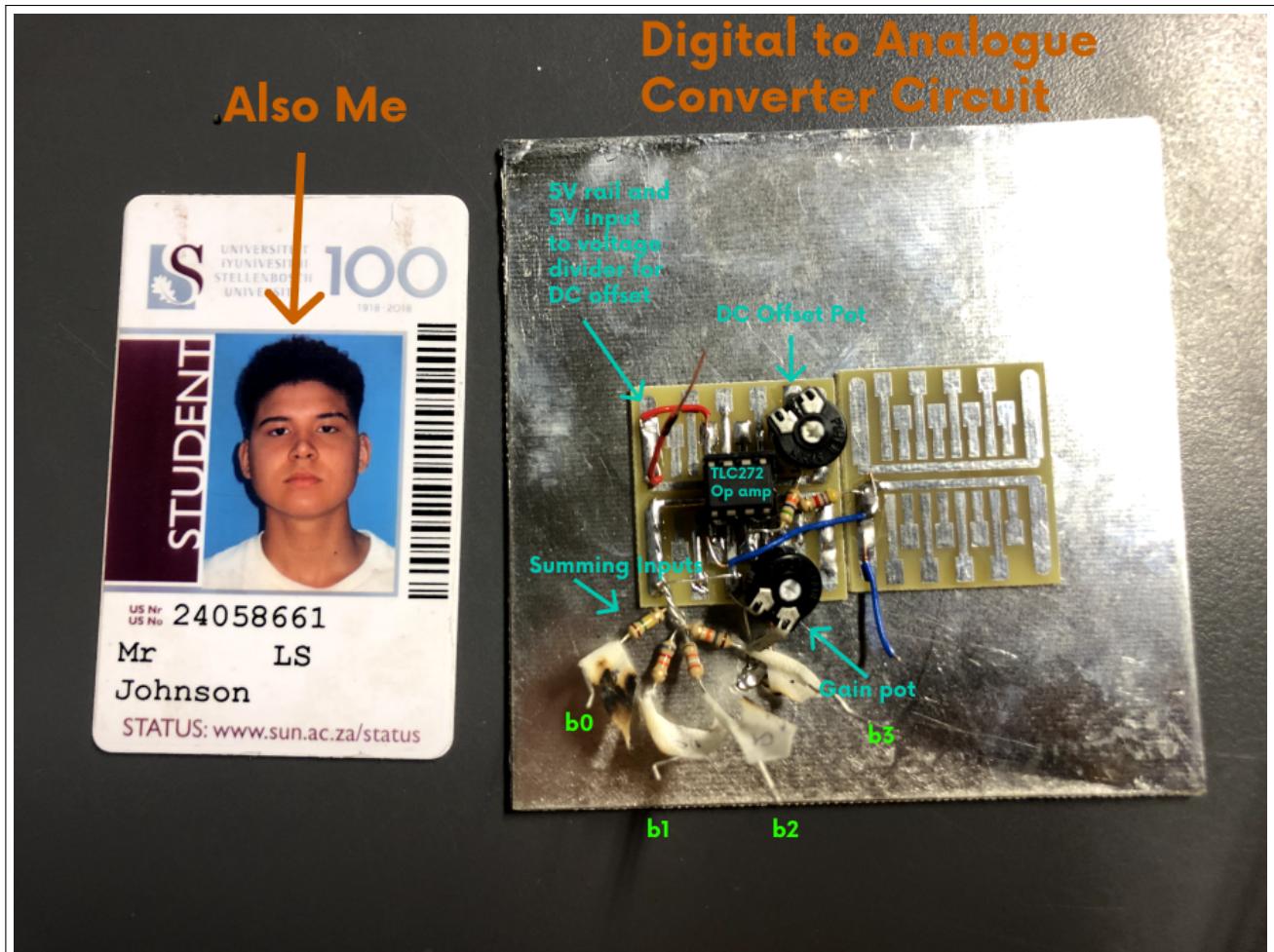


Figure 4.2: DAC circuit with labels

4.3. Motor Control and Driver circuit



Figure 4.3: Low-side switch and current sensor physical implementation for left motor control

4.4. Complete RC Car Circuit

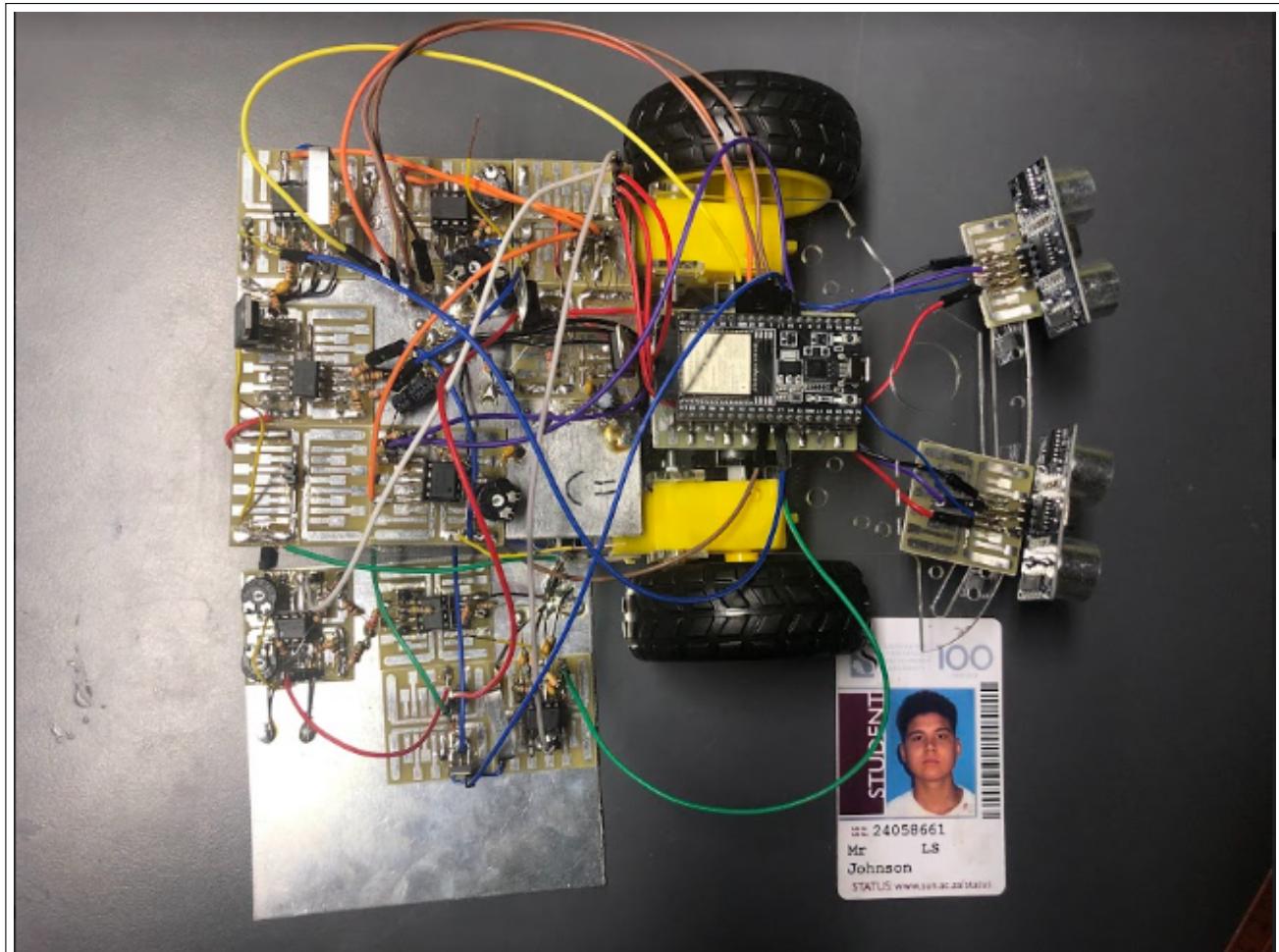


Figure 4.4: Physical implementation for the complete RC car circuit

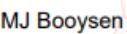
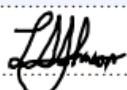
Bibliography

- [1] BBC, “How to make opamps amp op,” 2018. [Online]. Available: www.electronics-tutorials.ws
- [2] “Operational amplifiers specifications.” [Online]. Available: https://www.globalspec.com/specsearch/searchform/semiconductors/analog_linear_devices/amplifier_chips/operational_amplifier_chips
- [3] C. C. S. <https://circuitcellar.com/author/staff/>, “Non-invasive current sensor,” Jan 2021. [Online]. Available: <https://circuitcellar.com/research-design-hub/non-invasive-current-sensor/>
- [4] [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [5] L. M. Engineers, “How hc-sr04 ultrasonic sensor works amp; how to interface it with arduino,” Jun 2022. [Online]. Available: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>
- [6] “Using pwm to generate an analog output.” [Online]. Available: <https://ww1.microchip.com/downloads/en/Appnotes/90003250A.pdf>
- [7] “Low-pass filter a pwm signal into an analog voltage - technical articles.” [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/>
- [8] “The basics of ultrasonic sensors,” Apr 2021. [Online]. Available: <https://www.cuidevices.com/blog/the-basics-of-ultrasonic-sensors>
- [9] R. Burnett, “Understanding how ultrasonic sensors work,” Mar 2021. [Online]. Available: <https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm#:~:text=An%20ultrasonic%20sensor%20is%20an,information%20about%20an%20object's%20proximity.>
- [10] [Online]. Available: <https://www.ti.com/lit/ds/symlink/tlc2272.pdf>
- [11] B. University, “Bu-201: How does the lead acid battery work?” Mar 2022. [Online]. Available: <https://batteryuniversity.com/article/bu-201-how-does-the-lead-acid-battery-work>
- [12] “Complete guide on how to charge a lead acid battery,” Mar 2021. [Online]. Available: <https://www.power-sonic.com/blog/how-to-charge-a-lead-acid-battery/#:~:text=To%20obtain%20maximum%20battery%20service,the%20terminals%20of%20the%20battery.>

- [13] D. Hudgens, “Low-side current sense circuit integration - ti.com.” [Online]. Available: <https://www.ti.com/lit/an/sboa190/sboa190.pdf>
- [14] “Ultrasonic ranging module hc - sr04 - sparkfun electronics.” [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [15] “Fairchild fqd13n06l/fqu13n06l 60v logic n-channel mosfet manual : Free download, borrow, and streaming.” [Online]. Available: <https://archive.org/details/manuallib-id-2686269/page/n1/mode/2up?view=theater>
- [16] N. Dahl, “Mosfet gate resistor,” Jan 2022. [Online]. Available: <https://www.build-electronic-circuits.com/mosfet-gate-resistor/>
- [17] “1.2 v to 37 v adjustable voltage regulators - rs components.” [Online]. Available: <https://docs.rs-online.com/0054/0900766b813862fa.pdf>
- [18] Diksha, “Designing 12v lead-acid battery constant voltage limited current charger for ups (part- 2/17).” [Online]. Available: <https://www.engineersgarage.com/designing-12v-lead-acid-battery-constant-voltage-limited-current-charger-for-ups-part-2-17/>
- [19] “E-rt-rt670-v21a-0 - ritar power.” [Online]. Available: <https://www.ritarpower.com/uploads/ueditor/spec/RT670.pdf>
- [20] “Introduction to schmitt trigger.” [Online]. Available: <https://components101.com/articles/schmitt-trigger-introduction-working-applications>
- [21] W. J. Highton. [Online]. Available: <https://chemandy.com/calculators/schmitt-trigger-calculator.htm>

Appendix A

Social contract

 <p>UNIVERSITEIT•STELLENBOSCH•UNIVERSITY jou kennisvenoot • your knowledge partner</p>	
<p style="text-align: center;">E-design 344 Social Contract</p>	
<p style="text-align: center;">2022</p>	
<p>The purpose of this document is to establish commitment between the student and the organisers of E344. Beyond the commitment made here, it is not binding.</p>	
<p>In the months preceding the term, the lecturer (Thinus Booyens) and a few paid helpers (Rita van der Walt, Keegan Hull, and Michael Ritchie) spent countless hours to prepare for E344 to ensure that you get your money's worth, that you are enabled to learn from the module, and demonstrate and be assessed on your skills. We commit to prepare the assignments, to set the assessments fairly, to be reasonably available, and to provide feedback and support as best and fast we can. We will work hard to give you the best opportunity to learn from and pass analogue electronic design E344.</p>	
<p>I, Lee Johnson have registered for E344 of my own volition with the intention to learn of and be assessed on the principals of analogue electronic design. Despite the potential publication online of supplementary videos on specific topics, I acknowledge that I am expected to attend the scheduled lectures to make the most of these appointments and learning opportunities. Moreover, I realise I am expected to spend the additional requisite number of hours on E344 as specified in the yearbook.</p>	
<p>I acknowledge that E344 is an important part of my journey to becoming a professional engineer, and that my conduct should be reflective thereof. This includes doing and submitting my own work, working hard, starting on time, and assimilating as much information as possible. It also includes showing respect towards the University's equipment, staff, and their time.</p>	
Prof. MJ (Thinus) Booyens Signature:  Date: 1 July 2022	Student number: 24058661 Signature:  Date: 24 July 2022

Appendix B

GitHub Activity Heatmap

