



UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

E344 Assignment 8

Richard Knipe

23540427

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.

October 17, 2022

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.

23540427	
Studentenommer / Student number	Handtekening / Signature
R.Knipe Voorletters en van / Initials and surname	October 17, 2022 Datum / Date

Contents

Declaration	i
List of Figures	v
List of Tables	vii
Nomenclature	viii
1. Literature survey	1
1.1. Operational Amplifiers	1
1.2. Interfacing with and using the Ultrasonic Range Sensor	3
1.3. Converting PWM signals to analogue	4
1.4. Converting digital values to analogue equivalents	5
1.5. Lead-acid Battery Voltages and Currents	6
1.5.1. Comparison of different rechargeable battery types	6
1.5.2. Charging a lead-acid battery	6
2. Detail design	7
2.1. Current sensor	7
2.2. Ultrasonic Range Finder	9
2.3. Digital to analogue converter	11
2.4. Analogue Control of the Right Wheel	12
2.4.1. Requirements	12
2.4.2. Emitter follower	12
2.4.3. Analogue subtractor	12
2.4.4. Voltage regulator	13
2.4.5. Flyback diode	13
2.4.6. Future improvements	13
2.5. Digital Control of the Left Wheel	14
2.5.1. Low-side Switch	14
2.5.2. Current Sensing	15
2.5.3. LTSpice Circuit Diagram	15
2.5.4. Firmware (Range Sensor and PWM Control)	16
2.6. Battery Charger and Undervoltage Control	17
2.6.1. Requirements	17
2.6.2. Battery Charging Circuit	17
2.6.3. Undervoltage Protection Circuit	18

2.6.4. Battery-voltage Signal Conditioning Circuit	19
2.7. Graphical User Interface	20
2.7.1. Why Kivy	20
2.7.2. Functionality of the GUI	20
2.7.3. Details of the Software	20
3. Results	21
3.1. Current Sensor	21
3.1.1. DC Motor Current Measurements	21
3.1.2. LTSpice Simulation Outputs	21
3.2. Ultrasonic Range Finder	22
3.2.1. LTSpice Simulation Outputs	22
3.2.2. Measured Outputs	22
3.3. Digital to Analogue Converter	23
3.3.1. LTSpice Simulation Outputs	23
3.3.2. Measured Outputs	23
3.4. Analogue Control of Right Motor	24
3.4.1. LTSpice Simulation Outputs	24
3.4.2. Measured Outputs	25
3.5. Digital Control of Left Motor	26
3.5.1. LTSpice Simulation Outputs	26
3.5.2. Measured Outputs	27
3.6. Battery Charger and Undervoltage Control	28
3.6.1. LTSpice Simulation Outputs	28
3.6.2. Measured Outputs	29
3.7. Graphical User Interface	31
3.7.1. Problems and Future Improvements	31
4. Physical implementation	32
4.1. Current Sensor	32
4.1.1. Measured Results	32
4.1.2. Circuit Photograph	33
4.2. Ultrasonic Range Finder	34
4.2.1. Circuit Photograph	34
4.3. Analogue Control of Right Motor	35
4.3.1. Circuit Photograph	35
4.4. Digital Control of Left Motor	36
4.4.1. Circuit Photograph	36
Bibliography	37

A. Social contract	40
B. GitHub Activity Heatmap	41
C. MOSFET Comparison	42
D. GitHub Version Control	43

List of Figures

1.1.	Types of op amp circuits.	2
1.2.	Simulated noise in LTSpice	2
1.3.	Pinout and timing diagram of the US.	3
1.4.	PWM and its associated average voltages. [1]	4
1.5.	The Fourier spectrum of a PWM signal. [2]	4
1.6.	DAC types using inverting op amp circuits [3]	5
1.7.	Comparison chart and CCCV charging cycle.	6
2.1.	Current sensor: Amplifier cascaded to a filter.	7
2.2.	Ultrasonic range finder: Schematic diagram in LTSpice.	9
2.3.	4-bit DAC: Schematic diagram in LTSpice.	11
2.4.	Motor control circuit: Schematic diagram in LTSpice.	13
2.5.	Low-side switch and current sensor: Schematic diagram in LTSpice.	15
2.6.	A schematic of the provided charging circuit.	17
2.7.	An LTSpice schematic of the Schmitt trigger circuit.	18
2.8.	An LTSpice schematic of the signal conditioning circuit.	19
2.9.	Flow diagram showing operation of the Python GUI.	20
3.1.	Total current draw of current sensor in LTSpice	21
3.2.	Input signal with noise vs output signal without noise.	21
3.3.	Total current draw of US range finder in LTSpice	22
3.4.	Input and output step responses and noise.	22
3.5.	Ultrasonic range finder sensing objects at different distances.	22
3.6.	Total current draw of DAC in LTSpice.	23
3.7.	Digital step input and analogue step output responses of DAC.	23
3.8.	DAC outputs for four different digital inputs.	23
3.9.	Analogue subtractor: Input voltages, output motor control voltage, and current draw.	24
3.10.	Emitter follower: Voltage over motor and at base of TIP31C, and current draw.	24
3.11.	Motor output voltage for a medium and a medium-to-high speed command for close range.	25
3.12.	Motor output voltage for a low and a medium-to-high speed command for far range.	25
3.13.	Digital control circuit: Control voltage (red), current sensor output voltage (blue), and current draw through sense resistor (light purple).	26
3.14.	Zero speed close and far, and full speed close and far.	27

3.15. Current sensor output voltages for zero- and full range on full speed.	27
3.16. The charger readings in LTSpice.	28
3.17. Undervoltage protection circuit in LTSpice.	28
3.18. Signal conditioning circuit in LTSpice.	28
3.19. Charger output voltage when a 50Ω resistor is connected at its output.	29
3.20. Input (blue) vs output (yellow), showing the working Schmitt trigger.	29
3.21. Signal conditioning circuit outputs for 5V and 7.2V.	30
3.22. Screenshot of the GUI used to control the car.	31
4.1. DC motor under various test conditions.	32
4.2. Current sensor: built circuit.	33
4.3. Ultrasonic range finder sensor: built circuit.	34
4.4. Analogue Control of Right Motor: built circuit.	35
4.5. Digital Control of Left Motor: built circuit.	36

List of Tables

3.1. Motor current at 6.4 V	21
3.2. More motor voltages for various test cases.	25
C.1. Parameters considered when choosing the MOSFET. [4] [5] [6] [7]	42

Nomenclature

Electrical symbols and units

GND	Ground / Zero volts reference
RMS	Root Mean Square
$V_{BE}(ON)$	The Base-Emitter turn-on voltage of a BJT
VDD	Voltage Drain Drain
VSS	Voltage Source Source

Abbreviations

ADC	Analogue to Digital Converter
EMF	Electromotive Force
BJT	Bipolar Junction Transistor
$GPIO$	General Purpose Input Output
$HIGH$	Logical high. Usually the rail voltage of a circuit. Represents a digital “1”
LOW	Logical low. Usually the 0 V reference voltage. Represents a digital “0”
LPF	Low-Pass Filter
LSB	Least Significant Bit
MSB	Most Significant Bit
$Op\ Amp$	Operational Amplifier
PWM	Pulse Width Modulation
TTL	Transistor Transistor Logic
US	Ultrasonic Sensor

Chapter 1

Literature survey

In this section the provided operational amplifier is observed in more detail, and an operational amplifier configuration that will work best for the current sensor is chosen.

1.1. Operational Amplifiers

Operational amplifiers: limitations and considerations

The IC used in this project is the MCP6242. Parameters of it that needs to be considered for this design are [8]:

- Common-mode voltage (-0.3 V to 0.3 V)
- Power supply voltage (1.8 V to 5.5 V)
- Maximum output voltage swing (35 mV to 4.965 V)

Note that the above ratings are calculated using the power supply for this project, namely $V_{DD}=5\text{ V}$ and $V_{SS}=0\text{ V}$.

Operational Amplifier Configurations

Seeing that the purpose of the op amp circuit should be to amplify the voltage over the sense resistor to a voltage ranging from 0 V to just over 3 V , this left three configurations to choose from. These are the inverting, non-inverting and differential amplifiers. Others amplifiers exist too, but these are not applicable here (e.g., summing amplifiers, integrators, differentiators, etc.).

The inverting amplifier (Figure 1.1a) is single-ended, meaning that it amplifies a single input with respect to ground. The transfer function is $V_s = -V_e \frac{R_2}{R_1}$, where V_e denotes the input voltage and V_s the output voltage. Since we are working with DC voltages and the output can not be a different sign than the input, this configuration would not work.

The non-inverting amplifier (Figure 1.1b) is also single-ended, but the polarity of the output is not opposite to that of the input. It has a voltage transfer function of $V_s = V_e(1 + \frac{R_1}{R_2})$. This is thus an option to consider.

The differential amplifier (Figure 1.1c) is double-ended, meaning it takes two inputs and amplifies their difference, as the transfer function indicates: $V_s = \frac{R_2}{R_1}(V_2 - V_1)$. The reason

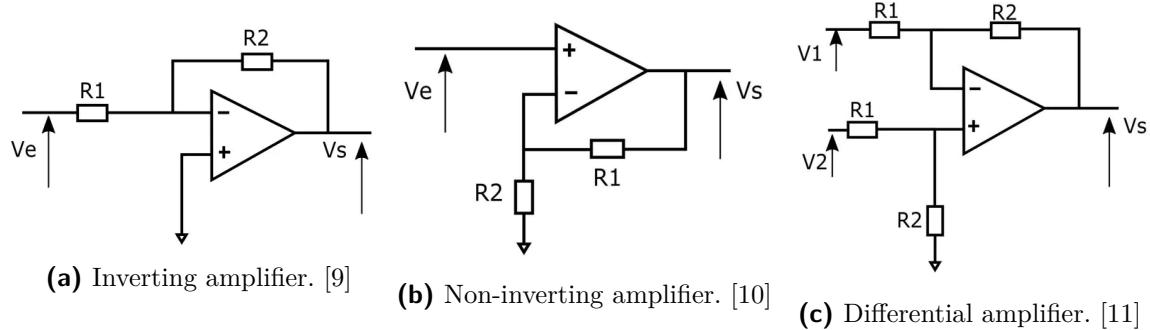


Figure 1.1: Types of op amp circuits.

it does this is to eliminate a voltage common to both terminals, called the common-mode voltage.

According to the requirements in the assignment pdf [?], a noise signal with amplitude 10 mV and frequency 1 kHz has to be eliminated.

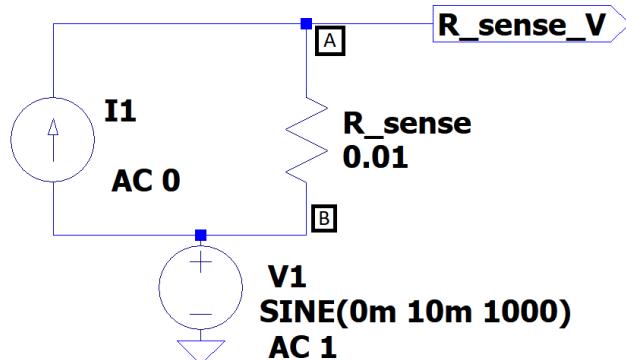
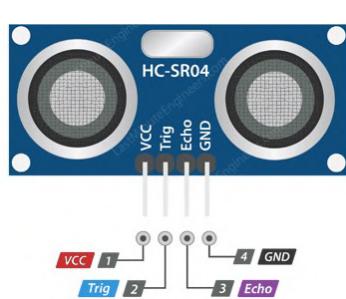


Figure 1.2: Simulated noise in LTSpice

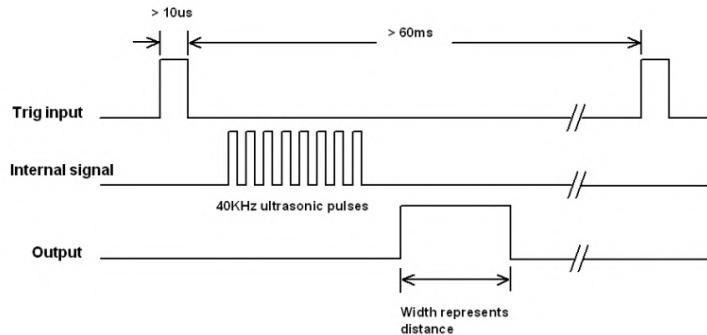
This noise signal was introduced at terminal B of the sense resistor in a provided LTSpice file to simulate the noise that would be present in real life (Figure 1.2). This effectively means that there is no noise on terminal B, and the noise is only present at terminal A. Therefore, one could either use a non-inverting amplifier with a filter before it that takes the voltage at A as input, or one could use a differential amplifier and introduce the same noise at the other terminal of this amplifier. It would then subtract the voltage common to both inputs (the noise), and the output would be noise-free. However, this would only work if the noise at B and at the other terminal of the differential amplifier are of exactly the same amplitude and also in phase, which is difficult to know for sure in real life. One can also not choose a cutoff frequency for a differential amplifier. Therefore, it was decided that a non-inverting amplifier would be good enough for this purpose.

1.2. Interfacing with and using the Ultrasonic Range Sensor

The US used in this project is the HC-SR04 (Figure 1.3a).



(a) Pinout. [12]



(b) Timing diagram. [13]

Figure 1.3: Pinout and timing diagram of the US.

The sensor has an operating voltage of 5 V and typically draws around 15 mA [14], resulting in a power usage of 75 mW.

The sensor makes a single measurement by receiving a trigger signal, which is a pulse with a width of 10 μ s at 5 V, and then emitting a burst of ultrasound (8 pulses at 40 kHz), while setting Echo HIGH [14]. It then sets Echo low when the sound wave is reflected back. Multiple measurements are done using a pulse train with a period of at least 60 ms [14]. See Figure 1.3b. The US operates with TTL voltage levels [14], and therefore an input trigger of only 3.3 V would be acceptable. This would not have been the case had the sensor worked with CMOS logic, which requires an input of at least 3.5 V for a HIGH [15]. Therefore, the ESP32's 3.3 V GPIO output voltage [16] would suffice as the trigger input.

The range of this sensor is 2 cm to 400 cm [14]. This distance is directly related to the width of the received Echo pulse signal by Equation 1.1, which utilizes the speed of sound in the air:

$$\text{distance to an object} = \frac{343 \text{ m/s} \times \text{time}}{2} 1001[17] \quad (1.1)$$

Using the above formula and substituting 400 cm for the distance, this yields 23.3 ms - the maximum (approximate) length of the pulse for a detected object. If no object is detected, the pulse width will be 38 ms [18]. Furthermore, the Echo pin provides a 5 V TTL output [14].

A note on optimization - If one is short on GPIO pins, the sensor can be used with three pins only, connecting Trig and Echo. This is feasible since Trig and Echo are not used at the same time [12]. However, since there are plenty of pins available, all four pins will be used for simplicity.

1.3. Converting PWM signals to analogue

A PWM signal is a series of pulses occurring at a constant frequency, but the widths of the individual pulses are not necessarily the same. The time for which the pulses are high divided by the time for which they are low is known as the duty cycle, and is usually expressed as a percentage. A special case kind of PWM signal is the square wave, which has a constant duty cycle. Common uses of PWM include the brightness control of a light, or the speed control of a DC motor [19].

Like many other wave types, PWM can be used to create an average voltage by multiplying the duty cycle by the maximum output of the signal (Figure 1.4). For example, when the maximum voltage is 5 V, the duty cycle can be set to 50 % to obtain an average voltage of 2.5 V (for each period).

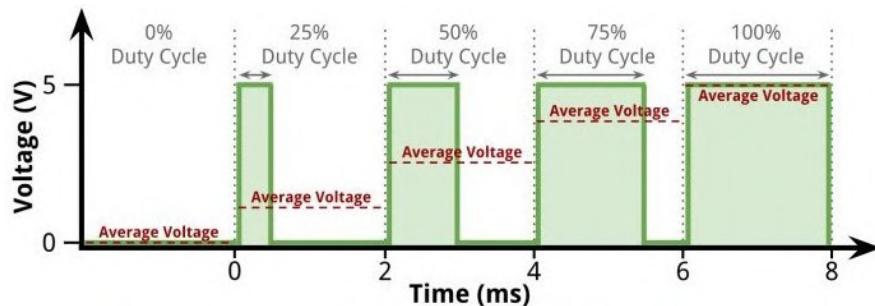


Figure 1.4: PWM and its associated average voltages. [1]

A Fourier analysis of the PWM signal shows that the spectrum of it consists of a fundamental frequency the same size as the frequency of the PWM signal, and higher order harmonics, which are at integer multiples of the fundamental frequency. These higher order harmonics have to be filtered out (hence the low-pass filter) to obtain the desired average value of the signal (the DC component in Figure 1.5). If the cutoff frequency is too low the rise time would be too long, and if it is too high, there would be too much ripple in the output. A careful midway between these two has to be found, and the process of determining the perfect cutoff frequency is not straight-forward.

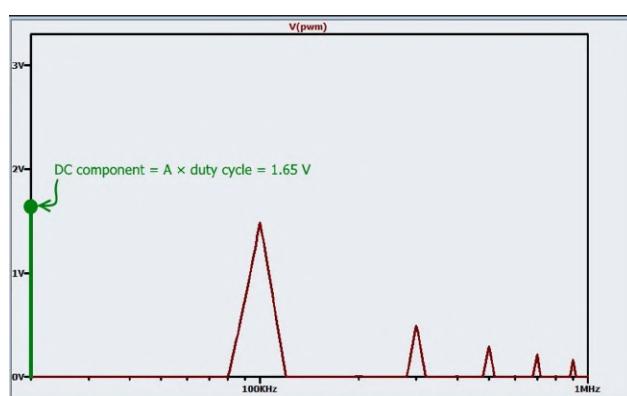


Figure 1.5: The Fourier spectrum of a PWM signal. [2]

1.4. Converting digital values to analogue equivalents

Since a binary number consists of bits of which some weigh heavier than others, this can be exploited by attributing gains (which correspond to the bit weights) to each individual bit, and then summing them together to obtain an analogue voltage proportional to the digital input. This can be done using a summing op amp, and the circuit is commonly known as the weighted resistor DAC.

Another common implementation is the R-2R ladder DAC [3], which uses a voltage divider and cascades bits in the DAC in a modular fashion.

Reasons to use the R-2R DAC include its simplicity (since identical resistors used), and it is also modular (DAC's resolution is increased simply by adding another R and 2R). A reason to use the weighted resistor DAC is that it uses less resistors*. It is also accurate when using low-resolution DAC's.

* Using a 4-bit DAC as an example, the R-2R approach would use 13 resistors. Since the R-2R would need three R resistors, five 2R resistors and two (using standard value ones to get an exact value) feedback resistors, you need $(3 + 2^5 + 2)$ 15 resistors in total (*2 since there are no standard value resistor for which another standard value one has exactly twice the resistance). The weighted sum circuit would need one standard value resistor for R, and two for 2R, 4R, and 8R, meaning $(1 + 2^3 + 2)$ 9 resistors in total.

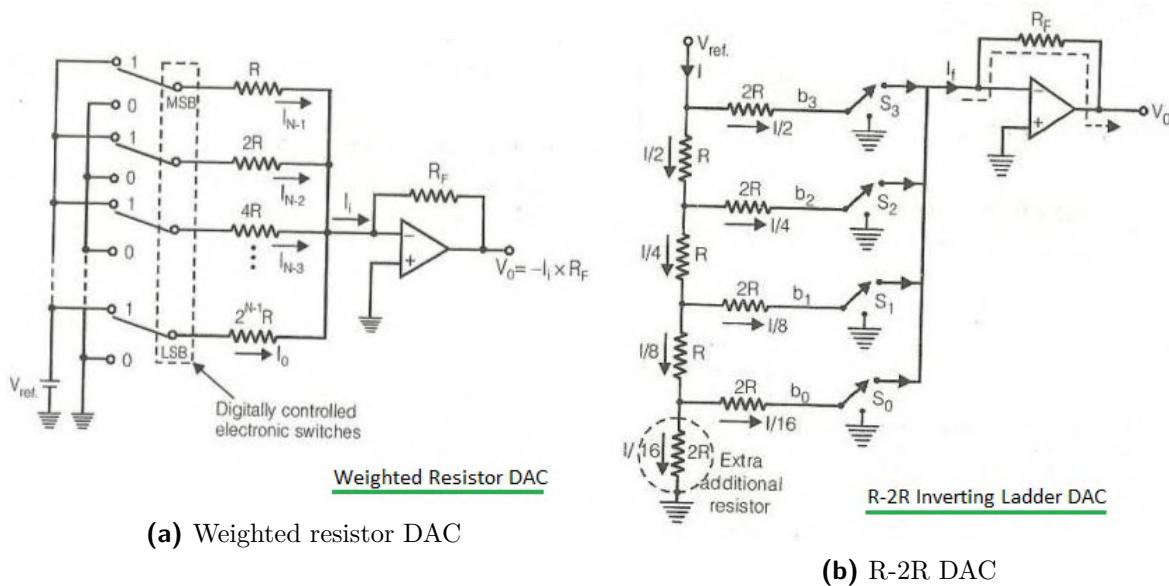


Figure 1.6: DAC types using inverting op amp circuits [3]

Figure 1.6 shows the R-2R and the weighted resistor approaches. Here only the inverting configurations are shown. An inverting configuration would be used if one wants to invert the binary representation, such that a logical low would show a high analogue signal. This is however only feasible with a single power supply. As will be seen in the design of the DAC, this can be side-stepped by adding an offset voltage to the non-inverting input terminal. The main reason for using the lead-acid batteries is probably it's low cost.

1.5. Lead-acid Battery Voltages and Currents

1.5.1. Comparison of different rechargeable battery types

Among other types of rechargeable batteries, such as Li-ion, LiPo, NiMH and NiCd batteries, to name the popular ones, lead-acid (Pb) will be used in this project. NiCd batteries are being phased out due to them being environmentally troublesome [20], so only the others are compared in Figure 1.7a.

1.5.2. Charging a lead-acid battery

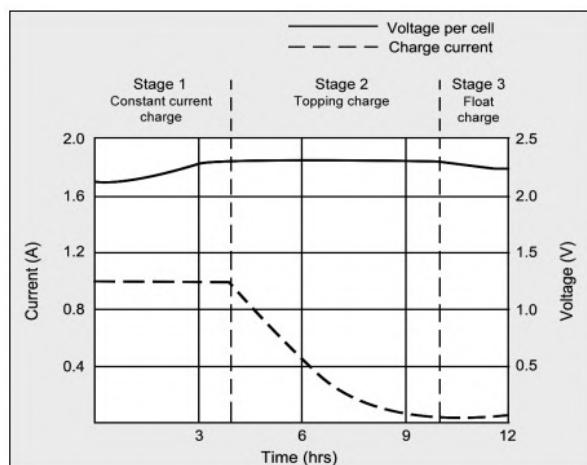
According to [21], the CCCV (constant current constant voltage) method is used. It takes about 12-16 hours to charge a Pb battery. It takes 5-8 hours to charge to 70%, and a slower topping charge then takes another 7-10 hours, which is essential for a prolonged lifetime (See Figure 1.7b). The charging current must be limited to 0.1C, which is 10% of the Ah-rating, and hence 700 mA for a 7 Ah battery. A single cell has 2V, so a 6V battery has 3 cells. The applied charging voltage should be between 2.3V and 2.45V per cell. For a 6V battery, this is thus between 6.9V and 7.35V.

	Example applications	Cons	Pros	Cycle life*	Typical cost**	Gravimetric energy density (Wh)
NiMH	HEVs, RC toys, phones	Low energy density, relatively expensive	Safe, ideal for portable projects for beginners	300-500	R1080	60-120
Pb	Cars, UPSs, robotics, heavy machinery	Big, heavy	Safest, high energy density, cheap	200-300	R450	30-50
Li-ion	EVs, toys, power banks	Expensive, dangerous	Best energy density, ideal for portable projects for advanced users	500-1000	R1800	110-160
LiPo	Drones, phones, RC toys	Expensive, dangerous	Best energy density, ideal for portable projects for advanced users	300-500	R1800	100-130

* Cycle life - number of charge and discharge cycles that a battery can complete before losing performance.

** for 7.2V, and 6V lead-acid (US, converted)

(a) A chart comparing four popular battery types. [22] [23]



Stage 1: Voltage rises at constant current to V-peak
 Stage 2: Current drops; full charge is reached when current levels off
 Stage 3: Voltage is lowered to float charge level

(b) Charge stages of a lead-acid battery. [21]

Figure 1.7: Comparison chart and CCCV charging cycle.

Chapter 2

Detail design

2.1. Current sensor

Since the provided DC motor had no brand, another was found looking exactly like it. According to its product information [24], the expected free-running current of the given generic DC motor is 200 mA, while the stall (maximum) current is 1.5 A. After physically testing these ratings (see Chapter 3), a maximum current of 1.2 A was decided upon.

Since these currents have to correspond to a voltage range of less than 0.1 V for 1 A and greater than 3.3 V for maximum current, 3.3 V was chosen to represent 1.2 A. This implies a voltage across the sense resistor ($10 \text{ m}\Omega$) of 12 mV. This was the voltage to be amplified to 3.3 V. The reason for the 3.3 V is that the ESP32's analogue input pins can handle voltages between 0 V and 3.3 V [16].

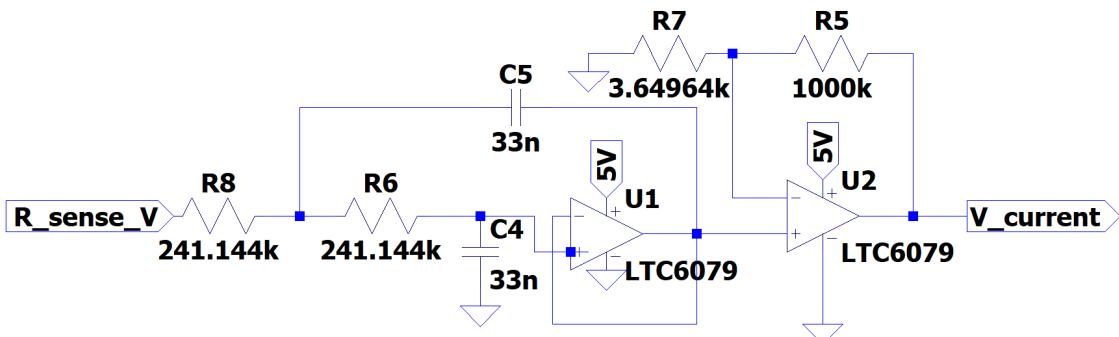


Figure 2.1: Current sensor: Amplifier cascaded to a filter.

As stated earlier, the non-inverting op amp configuration was chosen, and the design of the components are as follows [25]:

$$\begin{aligned} A_v &= 1 + \frac{R_2}{R_1} = 275 \\ \Rightarrow R_2 &= 274R_1 \\ \text{let } R_2 &= 1 \text{ M}\Omega \\ \Rightarrow R_1 &= 3.64964 \text{ k}\Omega \end{aligned} \tag{2.1}$$

And the design of the second order low-pass filter:

$$\begin{aligned}
f_c &= \frac{1}{2\pi\sqrt{R_3R_4C_1C_2}} \\
&\text{let } R_3 = R_4 = R \\
&\text{let } C_1 = C_2 = C \\
&\text{let } f_c = 20 \text{ Hz} \\
&=> 20 = \frac{1}{2\pi RC} \\
&\text{let } C = 33 \text{ nF} \\
&=> R = 241.144 \text{ k}\Omega
\end{aligned} \tag{2.2}$$

The components were chosen such that the resistors were all in the high $\text{k}\Omega$ -range, where possible. This was to limit the current through the device to be well under the $150 \mu\Omega$ requirement. To meet the rise time requirement (90% of the output voltage reached in less than 100 ms when going from minimum to maximum current) while still achieving a decent filtering quality, many values for the cutoff frequency were tried. Since it had to filter out the 1 kHz noise, the cutoff frequency needed to be below this of course, and since DC was used it would need to be ideally at around 0 Hz, but this would significantly increase the rise time since it would result in a larger RC time constant. A cutoff frequency of 20 Hz was found to be a good compromise.

A passive filter could be added to the input of the non-inverting amplifier (and this would essentially then be an active filter with a gain), but after many attempts to find a first-order filter that would give both the required settling time and filtering of the noise, it was decided that a second order filter had to be used instead. Initially, this was attempted using a single op amp and the circuit at [25], but this resulted in the output of the system clipping to the 5 V rail and forming a non-periodic square wave. After playing around with the filter wizard from [26], the conclusion was made that for such a high gain (275 V/V), op amps had to be cascaded since they are not able to provide this high gain alone while also filtering. It is however feasible if an op amp filter with unity gain is cascaded with a non-inverting amplifier at its output, that can provide that gain. This method would also make the system more modular.

2.2. Ultrasonic Range Finder

The range finder circuit had to be designed to meet the following specifications:

1. Filter PWM-type output from distance of 1 m down to 5 cm, such that distance ≥ 1 m gives output > 3 V, and distance ≤ 5 cm gives output < 0.3 V.
2. Response time must be such that if PWM changes (step change) from 5 cm to 1 m, the output must reach 90 % of final within 1.5 s.
3. The noise on the analogue signal must be less than 70 mVpp.
4. The whole circuit must use less than 750 μ A.
5. It is recommended that you make the output level adjustable with a potentiometer.

As stated in Chapter 1.3, the US operates on 5 V and typically draws 15 mA of current. As also stated in that section, the ESP32 can be used to power it, even though its GPIOs only output 3.3 V. Furthermore, the maximum current draw from the pins is 40 mA, which means it can control the US directly [16].

The device needed a filter stage to get the analogue waveform, and a gains stage to boost the signal to a range of 0 - 3.3 V, according to requirement 1 in .

Since the cutoff frequency of the LPF is not a straight-forward choice, an iterative approach was taken. Firstly, a design for a single, passive LPF was done, which resulted in an output with unacceptable ripple / rise time. Then the second-order filter was designed, but using an active filter, to add isolation between stages. It was attempted to include a gain into the same op amp, but this was found to influence the wave shape in an unpredictable manner, and a second, non-inverting op amp was therefore used for that. It was found that even after using a second order filter, requirement 2 was still not satisfied. Therefore, a third filter was added after the Sallen-Key section. This can be seen in Figure 2.2 It was therefore redundant to use the first op amp, but this was only realized close to the submission. Also, cost was not a requirement and there were many op amps available.

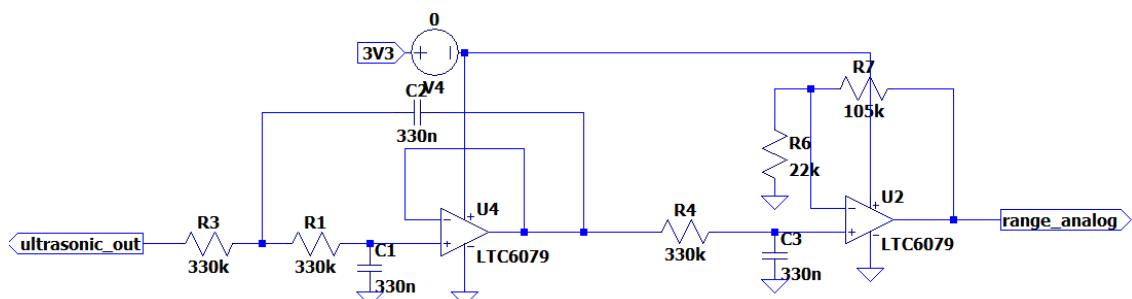


Figure 2.2: Ultrasonic range finder: Schematic diagram in LTSpice.

For calculating the component values of the Sallen-Key section, a method in an online tutorial [25] was used. A relatively small capacitor was chosen, since they have less variability

in capacitance over time, because they use dielectric materials which has a better performance than the bigger capacitors [27]. Therefore, 330 nF capacitors were chosen. Using Eq. 2.2 with a cutoff frequency of 1.5 Hz (after many iterations), the resistor values in the filter was found to be $330\text{ k}\Omega$.

Using this filter with the added-on first-order passive LPF on the output of the op amp, the minimum voltage of the analogue signal in the provided LTSpice file was observed, and found to be 539.22 mV. This also corresponded to a distance of 1.02 metres. This was then mapped to 3.3 V, and this meant a gain of 6.12. When substituted into Eq. 2.1 and choosing R_2 large ($110\text{ k}\Omega$), R_1 was found to be $21.48\text{ k}\Omega$. To meet requirement 5, a potentiometer of $220\text{ k}\Omega$ was used for R_2 .

2.3. Digital to analogue converter

As discussed in Section 1.4, the weighted resistor approach uses less resistors than the R-2R approach. Since the DAC is also only 4 bits, the accuracy of the weighted sum approach would not be an issue. As a power supply, 5 V was chosen over 3.3 V since it would mean that the common-mode voltage of the op amp would not be reached as quickly. Since the output had to be inverted (requirement 2) and a single power supply was to be used, an offset voltage was added to shift the negative result of the inverting amplifier to a positive range. One would think that this offset would need to be 3.3 V, but when deriving the transfer function for the summing amplifier with the offset voltage (V_{ref}) applied to the non-inverting terminal of the op amp, the following equation is obtained:

$$V_{out} = -\left(\frac{R_f}{R_0}V_0 + \frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2 + \frac{R_f}{R_3}V_3\right) + V_{ref}\left(\frac{R_f}{R_0} + \frac{R_f}{R_1} + \frac{R_f}{R_2} + \frac{R_f}{R_3} + 1\right) \quad (2.3)$$

Note: In this equation, V_0 represents the LSB and V_3 the MSB. Also, R_0 represents the resistor at V_0 , and R_3 the resistor at V_3 . Now, V_{ref} was chosen as 1.65 V. This meets Eq. 2.3. R_0 was chosen to be 22 kΩ. This means $R_2 = 44$ kΩ, $R_1 = 88$ kΩ, and $R_0 = 176$ kΩ. Substituting these values and the input of 3.3 for an output of zero volts into Eq. 2.3, R_f was found to be 11.73 kΩ. A voltage divider was designed to obtain 1.65 V from the supply voltage of 5 V. A 47 kΩ potentiometer was chosen as part of the bottom resistor (R_5) (connected to ground). The top resistor is R_4 . Using voltage division with a chosen range of 1.3 V to 2 V, two equations were obtained and solved to yield $R_4 = 149.057$ kΩ and $R_4 = 52.3714$ kΩ.

The ESP32 will supply 3.3 V from its GPIO pins, according to its data sheet [16]. Since the inputs are not known to be floating or grounded when 3.3 V is not applied, pull-down resistors had to be implemented. Furthermore, the output impedance of the ESP32's pins are 35 ohms, which may load the circuit. Therefore, it is important to use the pull down resistors to ensure predictable behaviour. Different values for these resistors were tried while ad the same time adjusting the gain resistor (R_f) to get the desired behaviour, and a value of 5.6 kΩ was decided on.

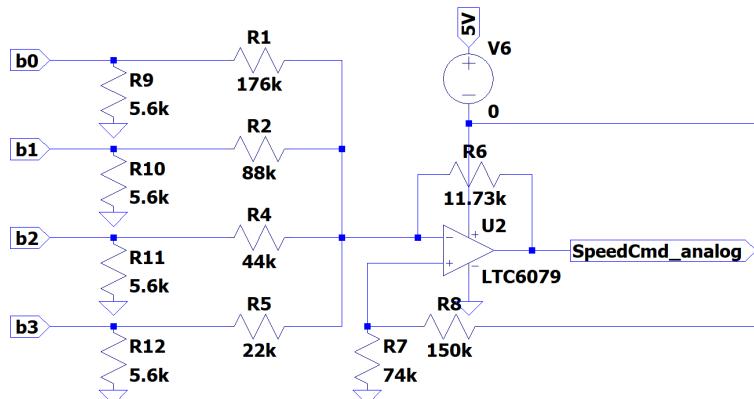


Figure 2.3: 4-bit DAC: Schematic diagram in LTSpice.

2.4. Analogue Control of the Right Wheel

2.4.1. Requirements

The requirements of the motor driver are a 7.2 V power supply, a voltage range of 0 - 6.2 V at least when using this supply, a low voltage over the DC motor when an object is near, a high voltage over the motor when an object is far and the torque command is HIGH and a low voltage over it when an object is far and the torque command is LOW, at most 0.5 V over the motor as a low voltage, at least 1 V below the battery voltage over the motor as a high voltage, a current consumption of the 5 V regulator of less than 500 μ A, a current consumption of a 7.2 V battery supply of less than 1.5 A, and finally also provision for motor back-emf.

2.4.2. Emitter follower

The DC motor of the right wheel will be connected directly to the battery supply to ensure that it can receive the highest voltage possible and draw as much current as needed. This will be done using an emitter follower controlled by the ESP32. An emitter follower is a kind of buffer that has about the same input- and output voltage, but it allows for a small base current to control a much larger collector current. See Figure 2.4b. Another nice feature of it is that the current draw of the load (connected to the emitter) is not limited by the base current, but rather draws as much current as needed from the collector supply, and the base current then matches with this via the β current gain ($I_C = \beta I_B$). The TIP31C NPN power transistor was chosen for this task, since it has a maximum collector current of 3 A [28] and the motor will only draw about 1.5 A maximum. $V_{BE}(ON)$ was assumed to be 0.7 V. To comply with the requirements, this meant that the voltage at the base of the TIP31C was to be 6.9 V. However, 6 V was later decided to be sufficient over the motor, and the voltage needed was thus 6.7 V.

2.4.3. Analogue subtractor

To achieve a voltage of 6.7 V at the base of the power transistor from 3.3 V, a gain of 2.03. This voltage had to come from the DAC and analogue range sensor outputs, which are also in the 0 - 3.3 V range. The output was thus to be $V_{base} = 2.03 \times (V_{US} - V_{DAC})$. To achieve this equation, a differential amplifier was used while choosing the two input resistances equal and the feedback resistor and pull-down resistor at the non-inverting input equal. This meant that the following (simplified) design formula $V_{out} = \frac{R_a}{R_b}(V_{US} - V_{DAC})$ could be used, where R_a is the resistor at the inverting input. Choosing high resistor values to limit current draw, $R_a = 270 \text{ k}\Omega$ and $R_b = 560 \text{ k}\Omega$. See Figure 2.4a.

2.4.4. Voltage regulator

A voltage regulator has to supply a constant 5 V to the circuit from the battery. The options were the AP7384, LD1117 and LM317T. The AP7384 has an output current limit of 50 mA [29] and was therefore not suitable. According to their data sheets, LM317T had a maximum output current of more than 1.5 A [30] and the LD1117 of maximum 1.3 A [31], but due to the difference in their dropout voltages one has to use 6.7 V to power the LM317T but can get away with 6 V for the LD1117, so it was chosen. The application note on page 26/46 of the data sheet of the LD1117 [31] was followed and the third capacitor in Figure 11 was added to improve ripple rejection. After using the given formula ($V_{OUT} = V_{REF}(1 + \frac{R_1}{R_2})$) with $V_{REF} = 1.25$ V, the component values were: $R_1 = 120\Omega$, $R_2 = 360\Omega$, $C_1 = C_2 = C_3 = 10\mu F$.

2.4.5. Flyback diode

Since the motor behaves like an inductor, it creates a back-emf that has to reduce to zero when the motor is switched off. The only path for this current to flow to is back through the control circuit, which may damage the transistor. A flyback- or "snubber" diode is used to create a path for this current to flow from the one motor terminal to the other (Figure 2.4b). It is connected with the anode to the positive side of the motor to ensure that it passes no current when the motor is running. The 1N4148 was chosen for this, and was soldered directly onto the motor's terminals.

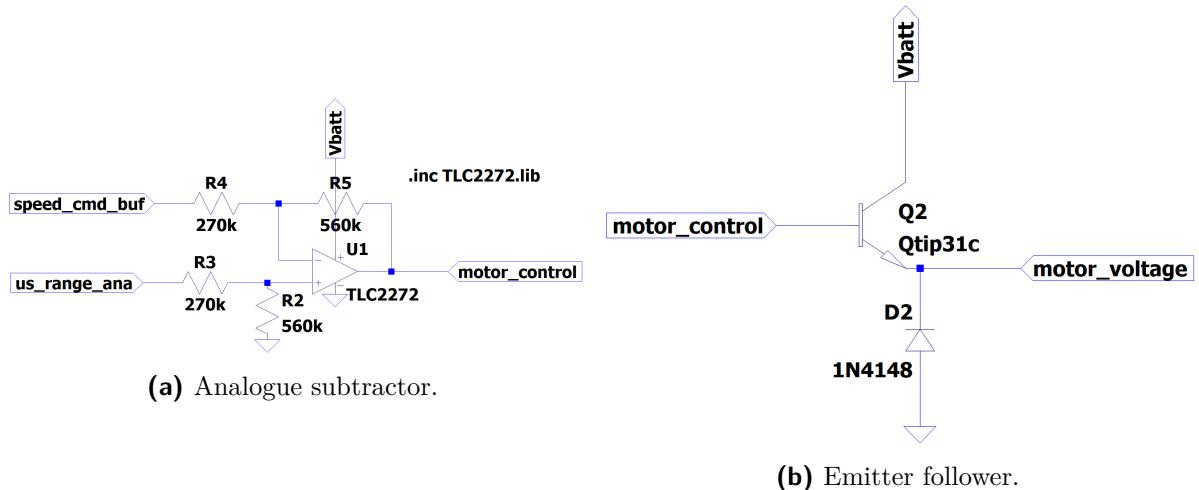


Figure 2.4: Motor control circuit: Schematic diagram in LTSpice.

2.4.6. Future improvements

In hindsight, a Darlington pair could have been used to control the motor instead of a NPN transistor, since this would mean a much lower current draw from the op amp subtractor, especially with the motor stalled. Note that the subtractor's gain would just have to be increased in this case, since the total voltage drop of a Darlington pair is about 1.4V.

2.5. Digital Control of the Left Wheel

2.5.1. Low-side Switch

According to the requirements, the left wheel motor had to be controlled with PWM, using a low-side switch. The positive terminal of the motor would be connected to the battery voltage, and the negative terminal to the drain of the MOSFET. The MOSFET's source pin would then be connected to the sense resistor of the current sensor (see Section 2.5.2), and the other terminal of this resistor to ground. By varying the PWM duty cycle, the amount of current that would flow through the motor would be controlled while keeping the voltage over it constant. This is because the motor is essentially an inductor, and its current can not change instantaneously. By switching the motor on and off rapidly, the current does not have enough time to flow through if the duty cycle is low, and it does for a higher duty cycle. No pull-down resistor was needed since the transistor would

Choosing a MOSFET

There were four MOSFETs available in the provided packet. To turn a MOSFET on, a threshold voltage, V_{GS} , has to be overcome. This is the voltage difference between the gate and the source. For an NMOS device, the gate has to be at a higher potential than the source. This would be easily achieved on the low-side, since the source would then be effectively connected to ground. For a PMOS device, the source voltage has to be greater than that of the gate. This transistor would be more suited as a high-side switch. Therefore, the provided PMOS transistor was eliminated as an option. After comparing a few more parameters of the remaining MOSFETs (See Table C.1), it was concluded that the FQU13N06L would be the most suited.

MOSFET gate- and pull-down resistor

According to [32], a gate resistor had to be added to reduce the initial current spike due to that may damage the GPIO pin of the ESP32. This is because the gate acts like a small capacitor. A widely-used value of 100Ω was used. A pull-down resistor of $10\text{ k}\Omega$ (also widely-used) was also used to prevent a floating input.

Choosing a PWM switching frequency

The sensor had to be able to filter the PWM noise would be used to control the left motor with a low-side switch (See section 4.4.1). Ideally, the PWM frequency was not to be heard (i.e., outside the human hearing range of 20 Hz to 20 kHz). When choosing a frequency below 20 Hz the motor would be switched on and off too slowly, preventing it from running continuously. For determining the maximum PWM frequency that the ESP32 can handle, the formula $f_{max} = \frac{80000000}{2^{PWM_bit_resolution}}$ [33] was used. Inserting 20 kHz, the bit number was 11.9658.

Therefore, 12 bits was used. This yielded a maximum frequency of 19.531 kHz, which was close enough.

2.5.2. Current Sensing

A current sensor had to be built to output a voltage that is a direct representation of the RMS (i.e., DC) current drawn by the left DC motor. Since this current will be read by the ESP32, the output voltage was chosen to be limited to 3.3 V, since the GPIO pins can not tolerate an input voltage of above 3.6 V [16].

Improvements on the previous current sensor

The current sensor which was designed in section 4.1 was used, since it filters out frequencies above 1 kHz and is used for an identical motor as discussed in previous sections. However, it was slightly modified. Firstly, it was now designed to be powered with 5 V instead of 3.3 V, to have a larger range to work with. Consequently, a voltage divider had to be implemented to step the 5 V down to 3.3 V, which can be read by the ESP32. Large resistors were used to limit the current. The maximum current was decided to be 500 mA instead of 1.2 A, to increase accuracy. Figure 2.5 shows the newly-designed current sensor in the full circuit. Take note that the previous current sensor was also changed to be identical to the newly designed one, since it works better.

2.5.3. LTSpice Circuit Diagram

Figure 2.5 shows a circuit diagram of the current sensor connected to the low-side switch. As before, a flyback-diode had to be placed across the motor to protect the MOSFET, and the 1N4148 was used as before.

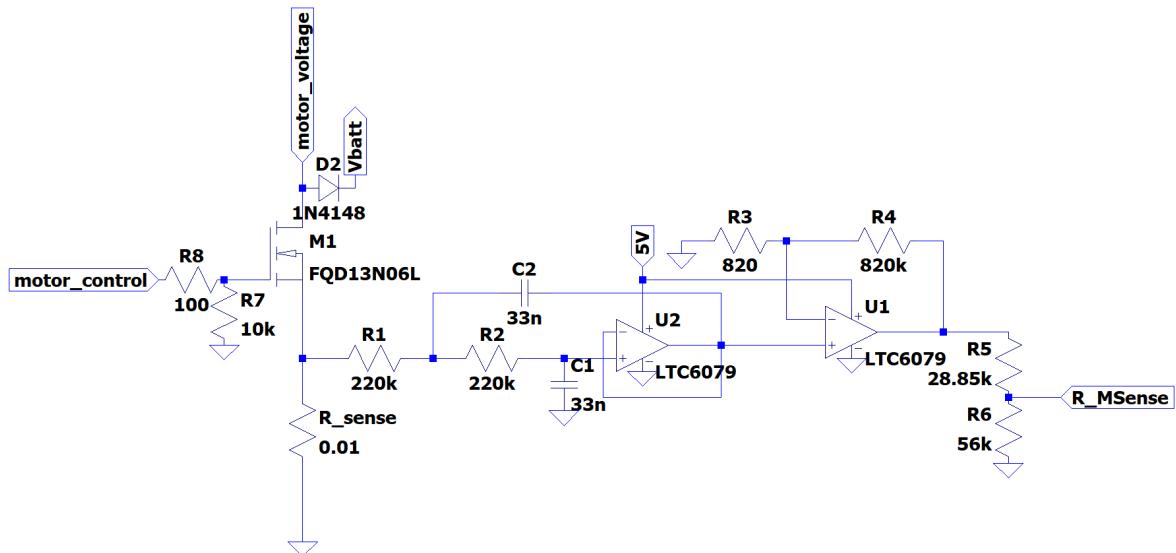


Figure 2.5: Low-side switch and current sensor: Schematic diagram in LTSpice.

2.5.4. Firmware (Range Sensor and PWM Control)

Range measurement

The HC-SR04 was used as the right US as well. The main source of the code for its control was [34]. However, the trigger of the US was not manual as in the tutorial but instead the same pin that was used for the right US was soldered onto the TRIG of the right US, which made use of the "ledc" functions. As in the tutorial, the distance was obtained from the "pulseIn" function, which returns the length of the pulse (either HIGH or LOW; HIGH in this case) in microseconds. This was then multiplied by the speed of sound and divided by two (to account for there-and-back travel time) to get the distance in centimetres.

PWM control of MOSFET

The main source of the code for the PWM control was [35], which also used the "ledc" functions. Firstly, the distance was found as explained in Section 2.5.4. It was then capped to 100 cm with an "if"-statement, since that was the maximum range of the analogue wheel's US. The distance was then converted to a "double" value between 0 and 1. The proximity was then obtained as $1 - \text{distance}$.

Since a remote control is not yet implemented, a DIP switch was used to input the speed command as a binary value to the ESP32. This was then converted to a digital value with the following line of code:

```
left-speed-duty-cycle = (8*in-3-new + 4*in-2-new + 2*in-1-new + in-0-new) / (double)  
15;
```

The final duty cycle as a fraction was then obtained by subtracting the proximity from the speed command. Another "if"-statement was used to make sure that the values were capped to zero if the proximity was greater than the speed command.

To change the PWM duty cycle applied to the MOSFET, the following line of code was used:

```
ledcWrite(ledChannelLeftWheel, (int) (left-duty-cycle * 4096));
```

Since the PWM resolution is 12 bits, 4096 is the maximum value. "left-duty-cycle" is the final duty cycle as a fraction. "ledChannelLeftWheel" was chosen as channel 2, since channel 0 and 1 were already used for other purposes (the ESP32 has 16 independent PWM channels).

Furthermore, there was no delay time in the main loop, to read and write as fast as possible.

2.6. Battery Charger and Undervoltage Control

2.6.1. Requirements

A battery charger had to be designed and soldered to a pre-made PCB, to limit the charging current to 0.1C. Secondly, an under-voltage protection circuit had to be designed that would cut power to the wheels when the battery voltage is less than 6.0 V, and turn it on again when the voltage rises above 6.1 V. Finally, a circuit had to be built to represent the battery voltage, to be read by the ESP32. Additionally, one could optionally design the circuit to produce an output of 0.1 V to 3.3 V for voltages from 5.0 V to 7.2 V.

2.6.2. Battery Charging Circuit

Since the current had to be limited to 0.1C and a battery of 7 Ah was used, this meant a charging current of 700 mA. The provided circuit can be seen in Figure 2.6. As can be seen, only the resistors had to be designed. The MOSFETs are there to include a functionality of enabling and disabling the charger, but since this is not required in this assignment (the requirement was scratched), pin 2 and 3 of M1 was simply connected, and the MOSFETs were omitted from the design. Thus, R3, M1, M2 and R5 were all not part of the final circuit.

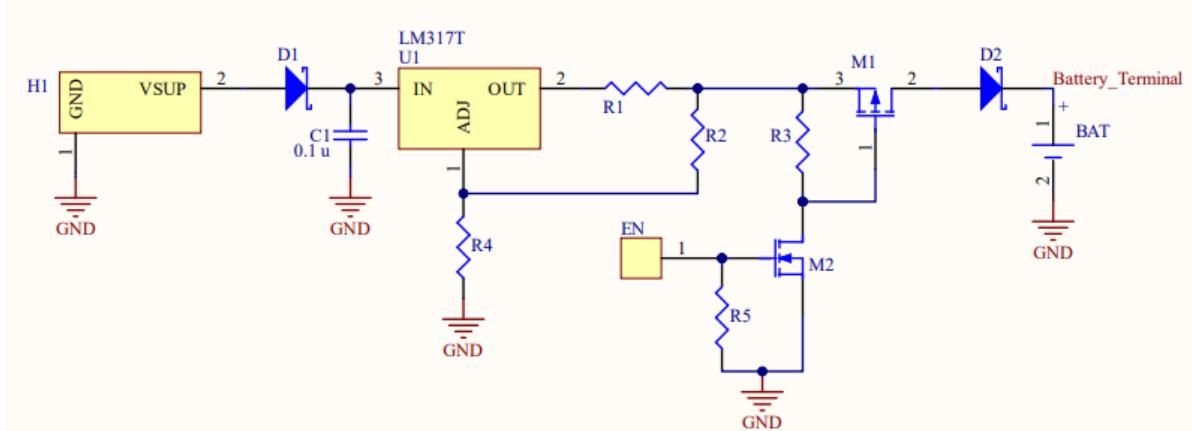


Figure 2.6: A schematic of the provided charging circuit.

Note that lead-acid batteries can be charged with a lab bench power supply, and according to [] it is recommended to stop charging the battery at 3% of the charging current, which is 21 mA in this case. According to its datasheet [], the 1N5819 Schottky diode has a forward-voltage drop of 0.25 V at 21 mA. Doing a KVL with the voltage at "Battery_terminal" equal to 7.2 V, this yielded an output voltage at pin 2 of the LM317T of 7.45 V.

Resistors R2 and R3 are used to control the output voltage of the LM317T variable voltage regulator according to the following equation: $V_O = V_{REF}(1 + R_4/R_2) + I_{ADJ}R_2$, where the last term can be neglected according to the datasheet, and $V_{REF} = 1.25 \text{ V}$. Choosing $V_O = 7.45 \text{ V}$ and $R_2 = 220 \Omega$, R_4 was found to be 1.0912.

R1 is used to control the maximum current that the regulator can output to the load. Since this is complex to determine, very small resistance values were tried in LTSpice to give an output current of 700mA. The value of 0.23Ω produced the desired output current whilst having very little effect on the output voltage, and was thus chosen for R1.

2.6.3. Undervoltage Protection Circuit

A Schmitt trigger is a circuit that switches an output on and off based on the voltage that it senses. If the sensed voltage is above a certain threshold, the Schmitt trigger turns the output on, and if the voltage is below another threshold, the output is turned off. The design tool at [36] was used (indirectly) to design an op-amp-based Schmitt trigger. Two equations on the site were used to solve the resistances analytically. The final design of the circuit can be seen in Figure 2.7.

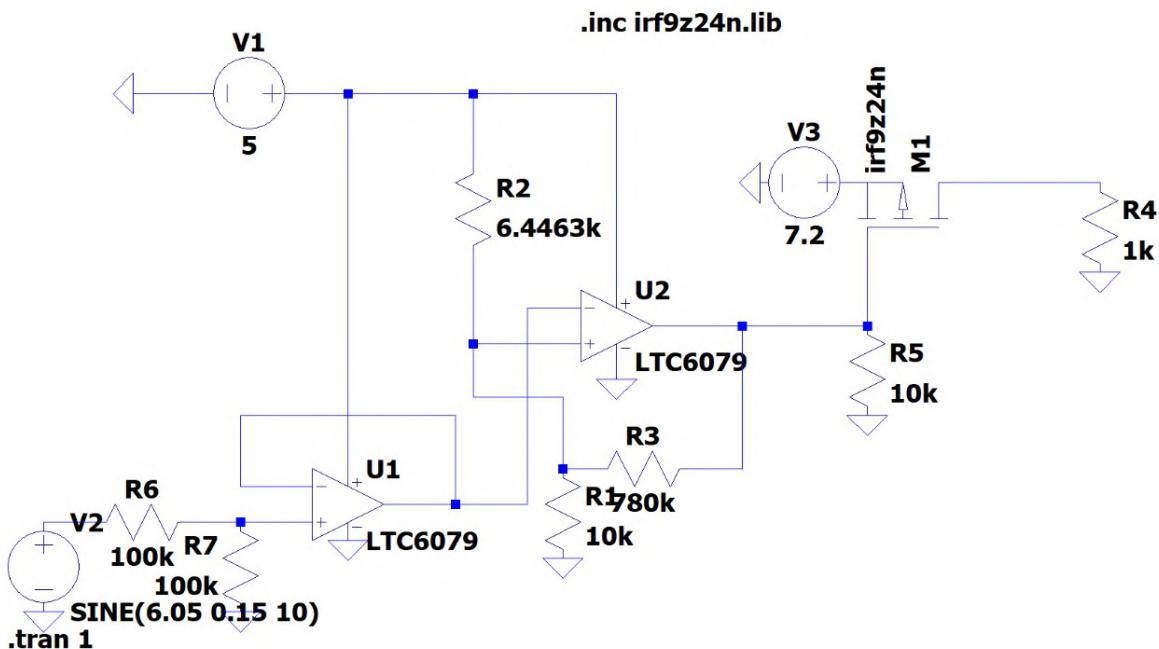


Figure 2.7: An LTSpice schematic of the Schmitt trigger circuit.

Choosing the upper threshold voltage as 3.05 V, the lower one as 3.0 V*, and R1 as $10\text{k}\Omega$, R2 and R3 were found to be 6.4463 and $780\text{k}\Omega$, respectively.

As can be seen in the circuit, the IRF9Z24N PMOS power transistor was used as a high-side switch (since it was the only provided PMOS transistor), and NMOS transistors are rather used as low-side switches, as discussed in Section 2.5.1. In this case it was easier to use a high-side switch for the motors, since the low-side is already occupied by other electronics, such as current sensors and even a low-side switch. Also, using a high-side switch would mean only a single switch is needed, since the same supply voltage can be used for the motors, but only a single switch would not suffice to connect the grounds of the motors. A general value of $10\text{k}\Omega$ was chosen as a pull-down resistor for the MOSFET, to prevent it from turning on at unpredictable times. Note that the $1\text{k}\Omega$ load is just there for testing the simulation.

* Since the op-amp is powered with 5 V and the input signal is up to 7.2 V, a voltage divider is used right after the input to divide the input by 2, thereby staying in the power supply range of the op-amp. This is the reason why 3.05 V is used instead of 6.1 V in the equations earlier. The input buffer is there to avoid interference between the Schmitt trigger and the voltage divider.

2.6.4. Battery-voltage Signal Conditioning Circuit

To design the circuit that would read the battery voltage that falls into the specified range of 5.0 V to 7.2 V, a differential amplifier was implemented. The idea was to subtract 5 from 7.2 and to amplify that difference into a range of 0 V to 3.3 V*. However, it was decided to rather amplify it to a range of 0 V to 5 V, and then to voltage-divide that to 3.3 V. This would ensure that even though the resistors are not exactly the same as in the design or to account for any imperfections regarding the op-amp (and provided that the voltage divider at the output had accurate-enough resistor values in real life), the output voltage would never exceed 3.3 V, since the op-amp can not deliver more than 5 V, because that is its supply voltage. Using the equation of a differential amplifier (as done in Section 2.4.3) and substituting the output voltage of 5 V and the input voltage of 7.2 V, a gain ratio of $R_b/R_a = 2.27$ was obtained, where R_a is the value of the resistors at the input terminals of the op-amp, and R_b the value of the other ones. Choosing R_a as 10 k Ω , R_b was found to be 22.7 k Ω . As for the design of the voltage divider, the top resistor was chosen as 10 k Ω , and the bottom one came out as 19.41 k Ω , according to the votlage divider equation.

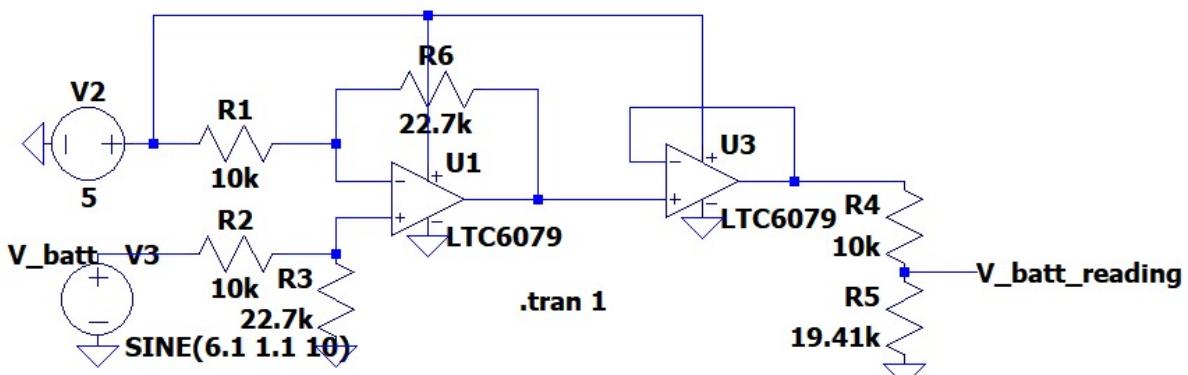


Figure 2.8: An LTSpice schematic of the signal conditioning circuit.

* The circuit was initially designed to produce the optional range of 0.1 V to 3.3 V (and it worked perfectly in LTSpice), but it was then decided to build the circuit without the voltage buffer at the input, thereby saving an op amp, and also changing the lower output voltage to 0 V. In real life, one would never get exactly 0 V, but slightly more due to common-mode voltages. For this reason, only the simplified design is instead discussed in this section.

2.7. Graphical User Interface

A graphical user-interface was to be designed using Python. The Kivy and KivyMD (“Material Design”) libraries were used for this.

2.7.1. Why Kivy

KivyMD was chosen for its professional look since it complies with Google’s standards for how widgets are displayed on Android. Kivy was chosen above Tkinter, since the Kivy code for a desktop application can be used for cross-platform mobile apps as well. This might be done in the future.

2.7.2. Functionality of the GUI

Sliders were used to send speed commands of 0-15 to each wheel independently. The speed command read by the car was then sent back and is also displayed alongside other telemetry readings for each wheel. A melody could also be played on a buzzer mounted on the car when pressing a button. See Section 3.7 for a screenshot of the GUI.

2.7.3. Details of the Software

Kivy works with a .kv file that contains all the parameters for each widget, like their position and size, and then the usual .py file. A drawback of Kivy is that it allows for no loops inside the .py file, otherwise the GUI freezes. A function named “schedule_interval” (from the Clock library) was tried in an attempt to update readings every second, but this only froze the sliders on the screen. This is the reason for the “Update Readings” button, which would allow the user to update the readings at any time. All initialization code was put in a function named “on_enter”, which is called automatically when the screen is created. This included the line of code to connect to the ESP32. The sliders were used to call the function to send the speed command to one of the wheels wheel at the moment the mouse was released. When the mouse was being held down, the slider would display the current value. See Figure 2.9.

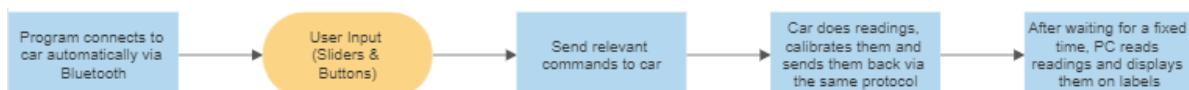


Figure 2.9: Flow diagram showing operation of the Python GUI.

Chapter 3

Results

3.1. Current Sensor

3.1.1. DC Motor Current Measurements

Table 3.1: Motor current at 6.4 V.

Motor current (A)	
Free-running	0.19
Stalled	1.18

3.1.2. LTSpice Simulation Outputs

Figure 3.1 shows the total current draw of the current sensor circuit, measured through the 5 V power supply. It complies with the requirement of a current draw less than $150 \mu\text{A}$.

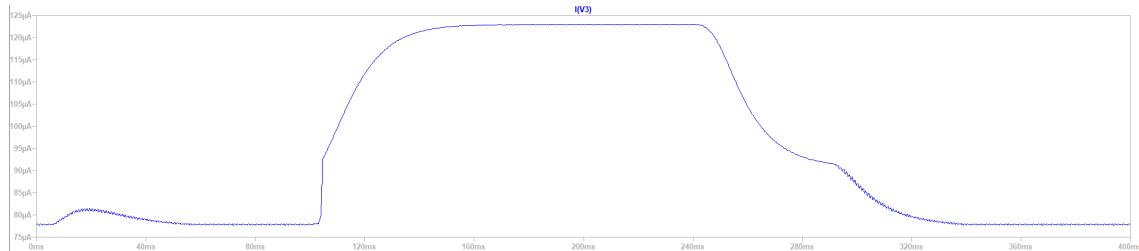


Figure 3.1: Total current draw of current sensor in LTSpice

Figure 3.2 shows that the noise-free output voltage of the sensor peaks at 3.3 V, for a 12 mV signal with a 10 mV sinusoidal noise wave superimposed on it. Clearly the noise levels are less than 250 mV on the output. The circuit also has the required rise time. Finally, the output voltage of the circuit is less than 0 V for 0 A.

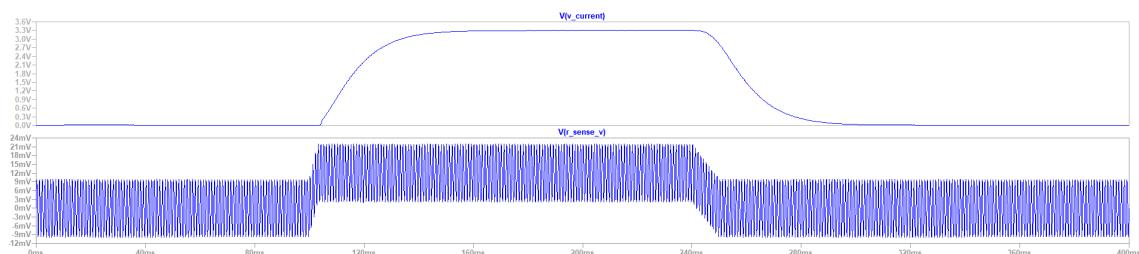


Figure 3.2: Input signal with noise vs output signal without noise.

3.2. Ultrasonic Range Finder

3.2.1. LTSpice Simulation Outputs

Figure 3.3 shows the total current draw of the range sensor circuit, measured through the 5 V power supply. It complies with requirement 4 (a current draw of less than $750 \mu\text{A}$).

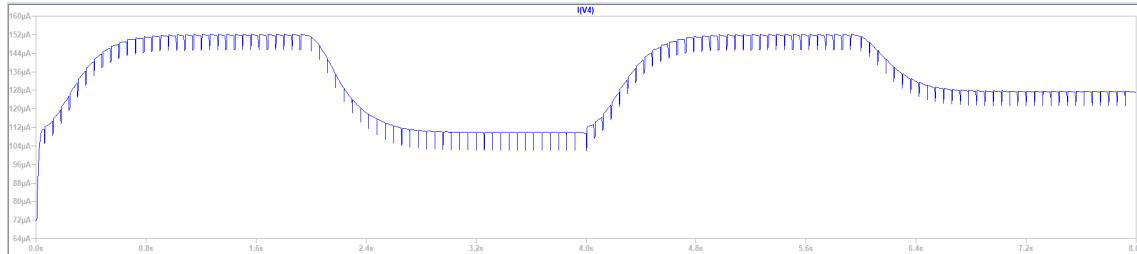


Figure 3.3: Total current draw of US range finder in LTSpice

Figure 3.4 shows the ultrasonic sensor's Echo pin (the PWM signal at the top), and the corresponding analogue signal at the bottom. Clearly the analogue signal's noise is less than 70 mVpp (meeting requirement 3), and it also meets the rise time requirement of 1.5 s for a full step change in voltage. Clearly it also meets requirement 1, as the output voltage is above 3.3 V for full range and below 0.3 V for zero distance.

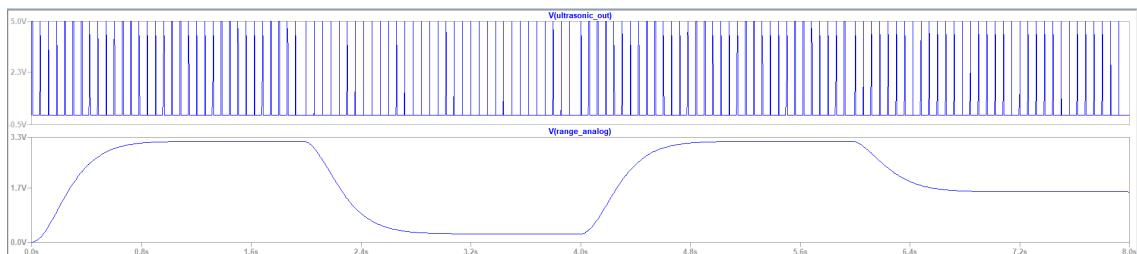


Figure 3.4: Input and output step responses and noise.

3.2.2. Measured Outputs

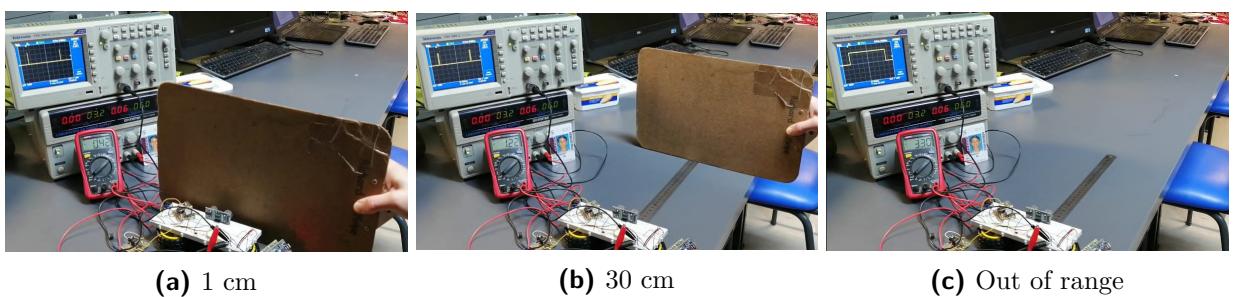


Figure 3.5: Ultrasonic range finder sensing objects at different distances.

Figure 3.5 shows partial compliance with the requirements. The voltage (see multimeter) is a bit too high at closer distances, but is perfect at 1 meter and when out of range.

3.3. Digital to Analogue Converter

3.3.1. LTSpice Simulation Outputs

Figure 3.6 shows the total current draw of the DAC circuit, measured through the 5 V power supply. Although not a requirement, the current draw is less than $200 \mu\text{A}$.

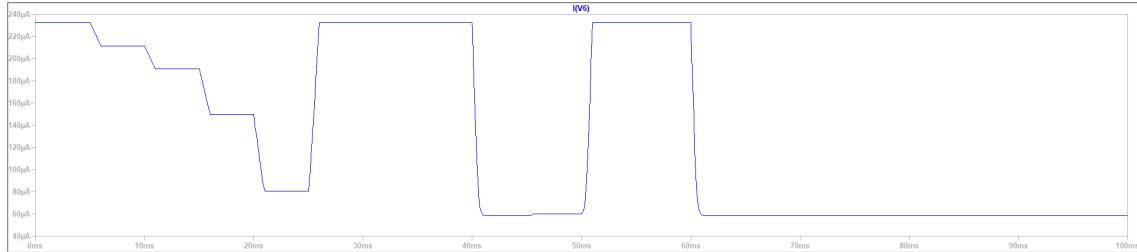


Figure 3.6: Total current draw of DAC in LTSpice.

Figure 3.7 shows the four digital inputs applied, and the corresponding step response of the analogue output voltage of the DAC. The design meets the requirements of a range of 0 - 3.3 V, and the output is also the inverted version of the digital input.



Figure 3.7: Digital step input and analogue step output responses of DAC.

3.3.2. Measured Outputs

Figure 3.8 shows compliance to the requirements. The theoretical values for (a) - (d) are 3.30, 3.08, 0.22, and 0.00 V, which are close to the measured ones.

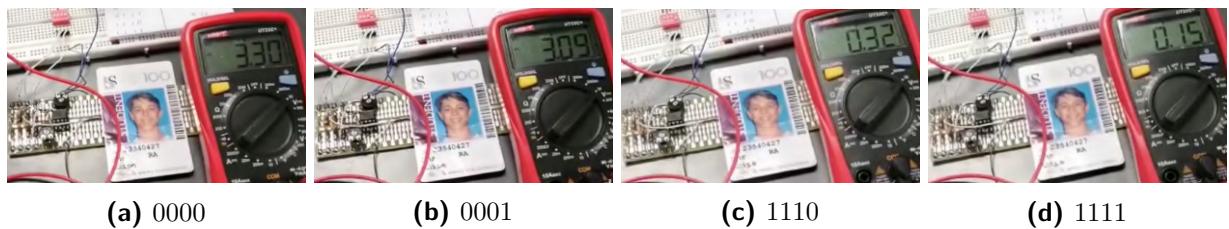


Figure 3.8: DAC outputs for four different digital inputs.

3.4. Analogue Control of Right Motor

3.4.1. LTSpice Simulation Outputs

Figure 3.9 shows the simulated analogue subtractor. The DAC output is subtracted from the speed command, as designed. The current draw from the battery is less than 2.5 mA. A low voltage is less than 0.5 V. The other requirements are also satisfied.

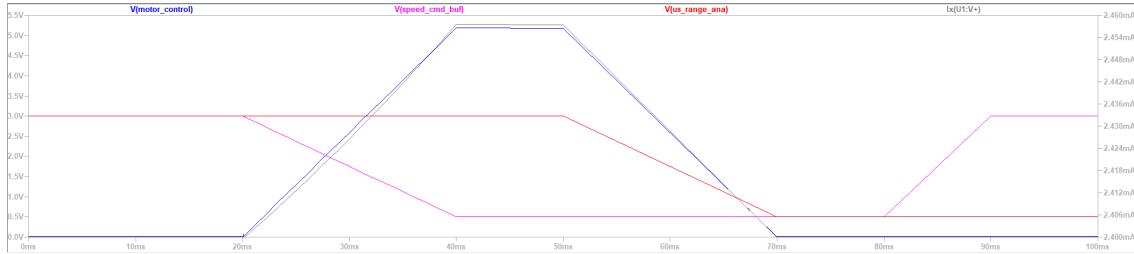


Figure 3.9: Analogue subtractor: Input voltages, output motor control voltage, and current draw.

The emitter follower simulation can be seen in Figure 3.10. The voltage over the motor is higher than 6.5 V (the simulated battery voltage) - 1 V. The current draw from the battery is also less than 1.5 A. Provision is also made for the back-emf of the motor. The simulation shows conformance with the other requirements as well.

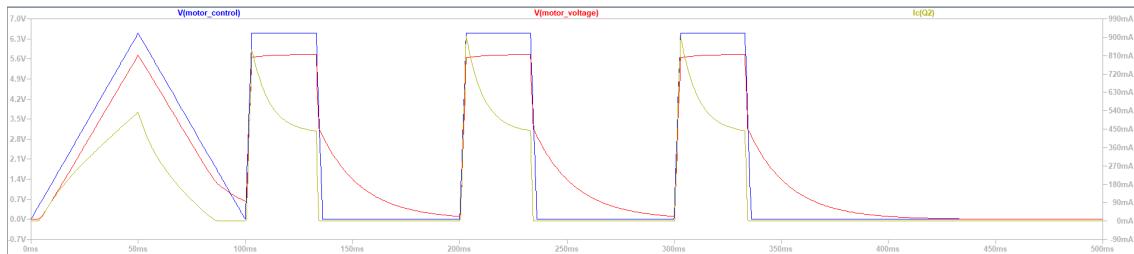
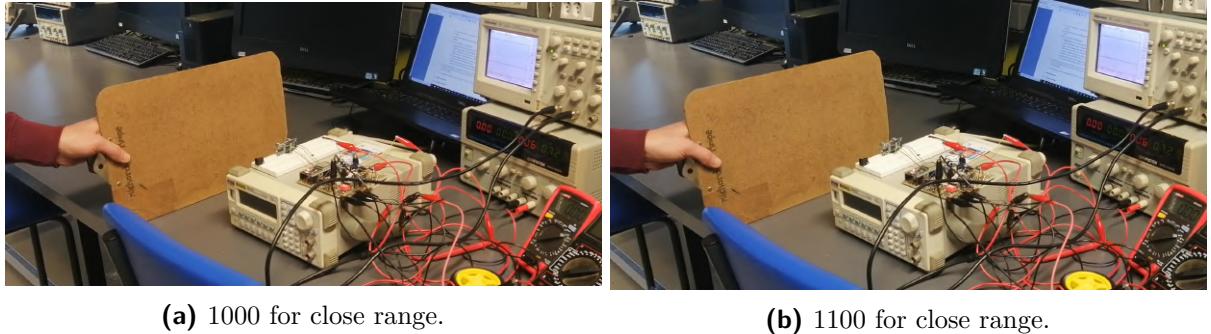


Figure 3.10: Emitter follower: Voltage over motor and at base of TIP31C, and current draw.

3.4.2. Measured Outputs

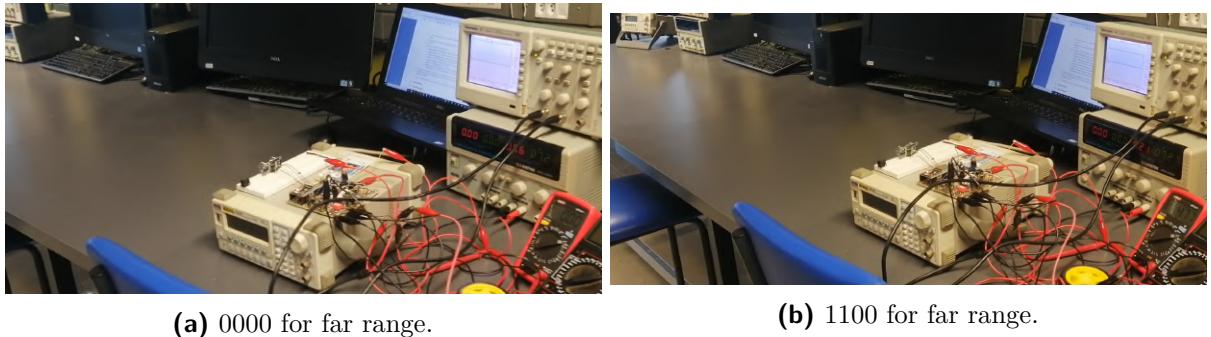
Figures 3.11 and 3.12 shows some sample outputs of the motor controller. Since it is unpractical to test all possible scenarios, table 3.2 contains a larger assortment of measurements.



(a) 1000 for close range.

(b) 1100 for close range.

Figure 3.11: Motor output voltage for a medium and a medium-to-high speed command for close range.



(a) 0000 for far range.

(b) 1100 for far range.

Figure 3.12: Motor output voltage for a low and a medium-to-high speed command for far range.

Table 3.2: More motor voltages for various test cases.

Scenario	Motor voltage (V)
Close, 1111	0.46
Far, 0000	0.00
Far, 0101	2.92
Far, 0111	3.83
Far, 1000	4.40
Far, 1100	4.77
Far, 1111	5.77

From these results, it is clear that the output voltage is always zero when the object is (reasonably) close to the sensor, irrespective of the input. This is not exactly as designed, but serves as a good precaution. The rest of the voltages all make sense.

Furthermore, the maximum current draw from the lab bench power supply in all of the above tests was 0.24 A, which is less than 1.5 A.

For the above reasons, the physical implementation complies with the requirements.

3.5. Digital Control of Left Motor

3.5.1. LTSpice Simulation Outputs

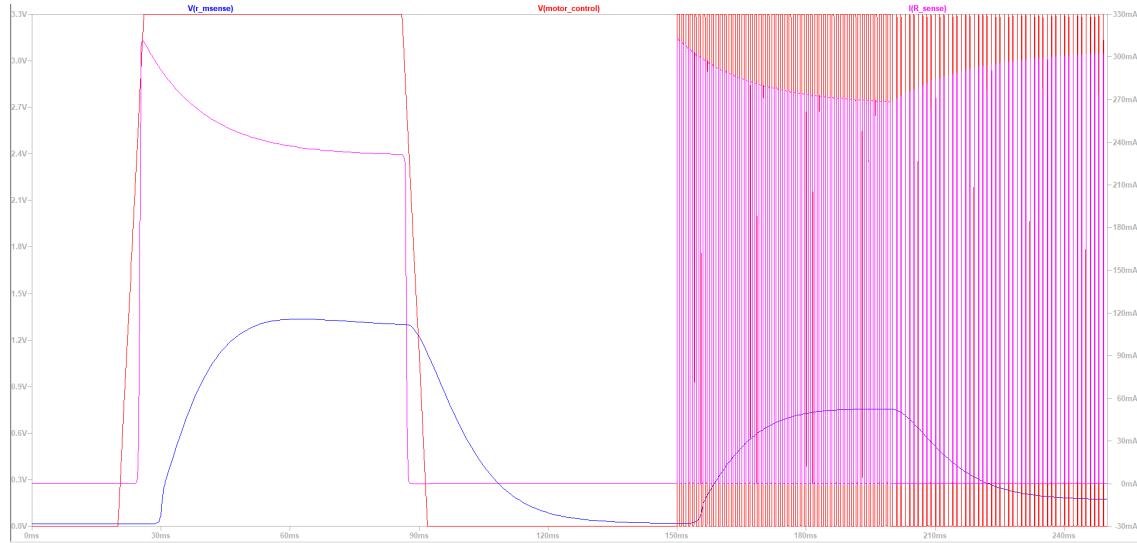


Figure 3.13: Digital control circuit: Control voltage (red), current sensor output voltage (blue), and current draw through sense resistor (light purple).

The frequency used in the SPICE simulation is 1 kHz. If it works for 1 kHz it will work for 20 kHz as well. As can be seen from Figure 3.13, the current sensor output voltage does not contain any ripple, as designed. At higher PWM duty cycles, the current sensor's output voltage is higher, as expected. The current draw from the motor is about 230 mA, which is similar to the full-power voltage over the motor as discussed earlier. For the design of 500 mA at 3.3 V, this would mean that we expect about 1.5 V at the output of the current sense circuit, but the output is only about 1.3 V. This is not a big problem however, since it can just be calibrated using software when having to read the current with the ESP32.

3.5.2. Measured Outputs

Important note: The PWM control duty cycle is multiplied by 0.7 in all the below measurements, since the analogue motor only had 5 V over it at full speed and full range, and the digital motor had 7.2 V over it. This was done to test if the car was able to drive straight.

PWM control

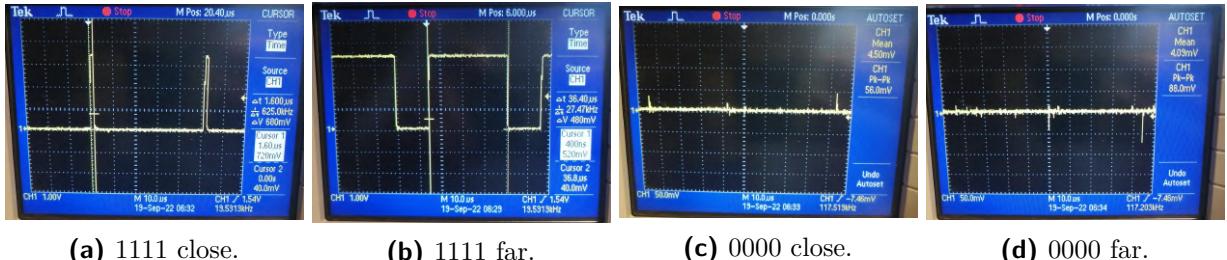


Figure 3.14: Zero speed close and far, and full speed close and far.

According to Figure 3.14, the PWM is what is expected at the ranges and speed commands. One can see in Figure 3.14b that $36.4 \mu\text{s} / 50 \mu\text{s}$ (for 20 kHz) = 0.728, which is close to the 0.7 scaling, as stated in the important note.

Current sensor



(a) 1111 with closest range. Current sensor input (blue) vs output (yellow) **(b)** 1111 with full range. Current sensor input (blue) vs output (yellow)

Figure 3.15: Current sensor output voltages for zero- and full range on full speed.

Figure 3.15 shows that the current sensor outputs about 600 mV at a motor command of zero, which is not ideal, but can be worked around in software by things like calibration. The voltage for a full-speed current (about 200 mA) is 1.6 V, which is close enough to the expected value of 1.32 V. The full range image also shows noise rejection of the current sensor at high frequencies.

3.6. Battery Charger and Undervoltage Control

3.6.1. LTSpice Simulation Outputs

Battery charging circuit

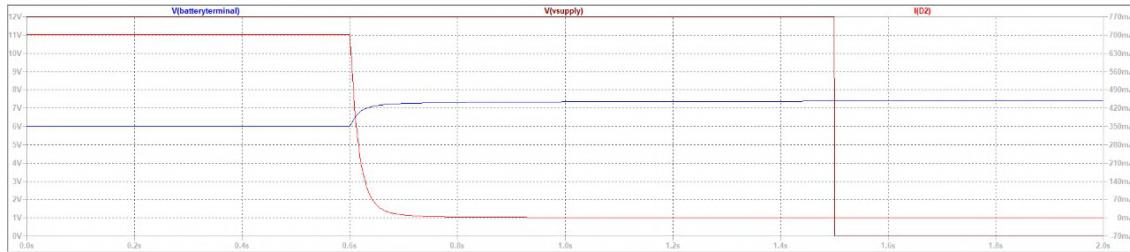


Figure 3.16: The charger readings in LTSpice.

Undervoltage protection circuit

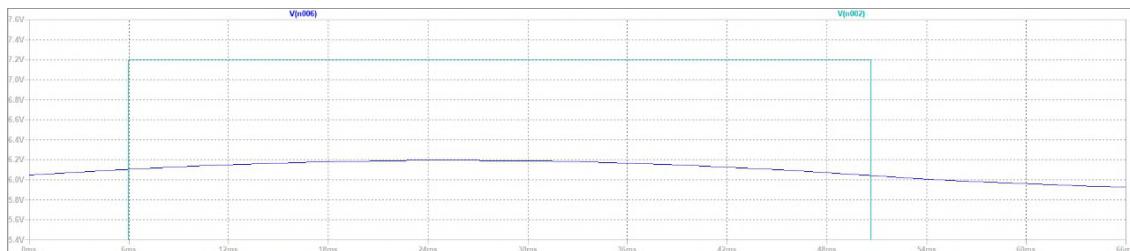


Figure 3.17: Undervoltage protection circuit in LTSpice.

Battery-voltage signal conditioning circuit

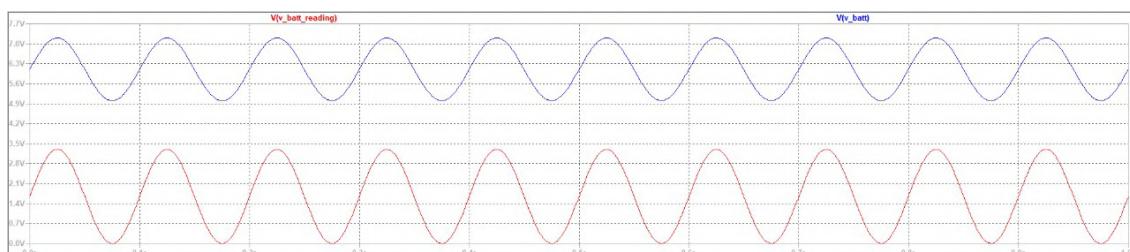


Figure 3.18: Signal conditioning circuit in LTSpice.

As can be seen above, the LTSpice simulations conform to the requirements. Note that the charger's output is higher than what was designed for, since the current in LTSpice is 0 mA at full charge, and this will never happen in real life.

3.6.2. Measured Outputs

Battery charging circuit

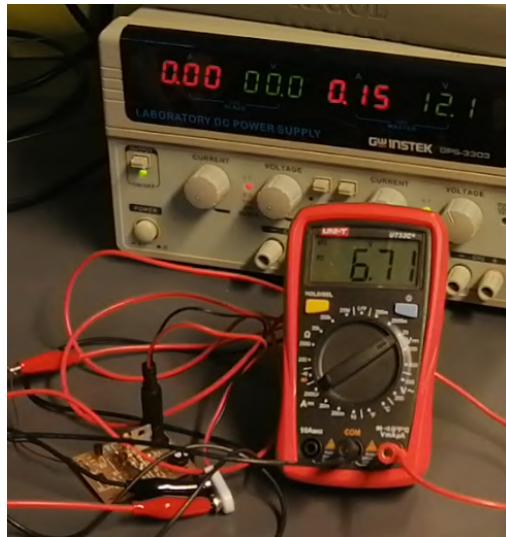


Figure 3.19: Charger output voltage when a 50Ω resistor is connected at its output.

Figure 3.19 shows that the current draw through a 50Ω resistor is 150 mA, which is small compared to what was designed. The output voltage at no load is exactly 7.2 V, however. Small resistor values were tried for R1 but the attempt to produce the 700 mA in practice was futile.

Undervoltage protection circuit

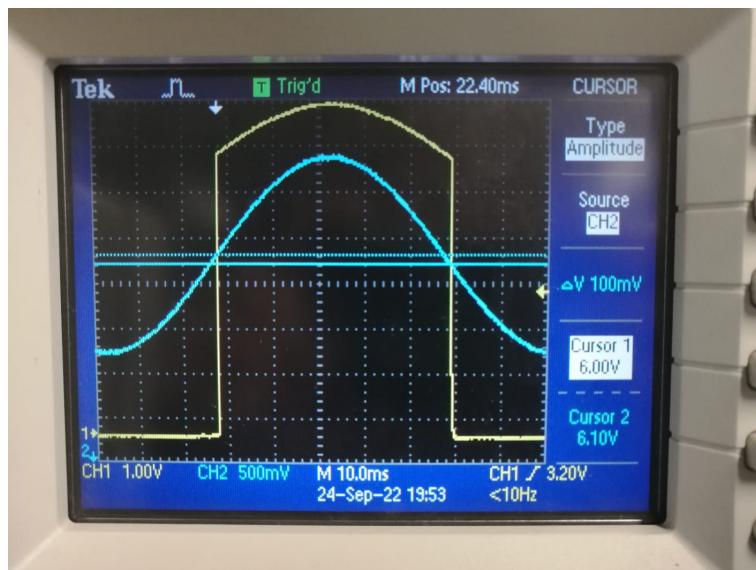


Figure 3.20: Input (blue) vs output (yellow), showing the working Schmitt trigger.

As can be seen in Figure 3.20, the Schmitt trigger turns on the output when the input voltage rises above 6.1 V, and it turns the output off when the input voltage dips below 6.0 V, as designed.

Battery-voltage signal conditioning circuit

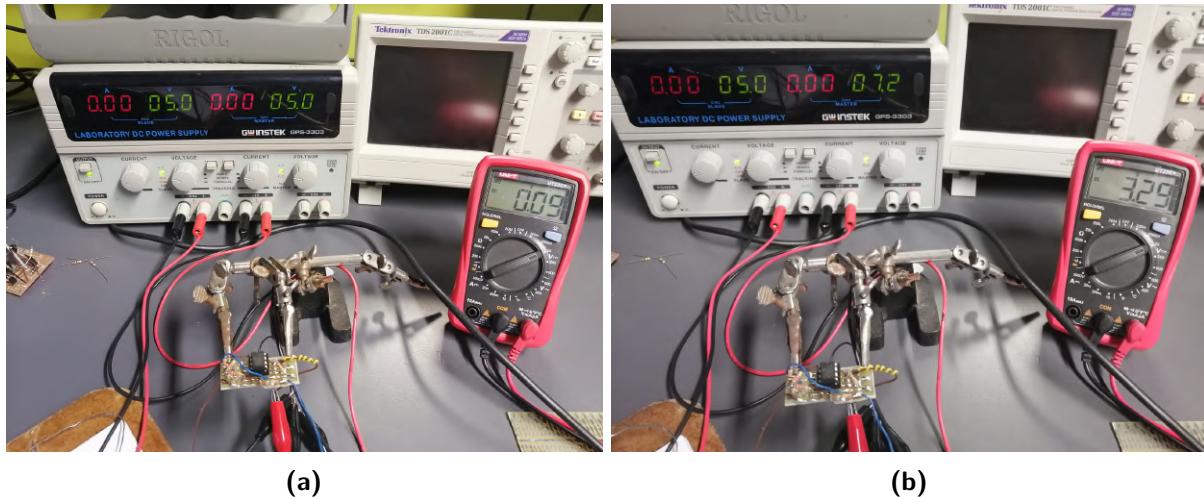


Figure 3.21: Signal conditioning circuit outputs for 5V and 7.2V.

Figure 3.21 shows the voltage readings for the signal conditioning circuit. The output voltage is 0.09 V at 5 V input, and 3.29 V at 7.2V input, which is close enough to the designed values.

3.7. Graphical User Interface

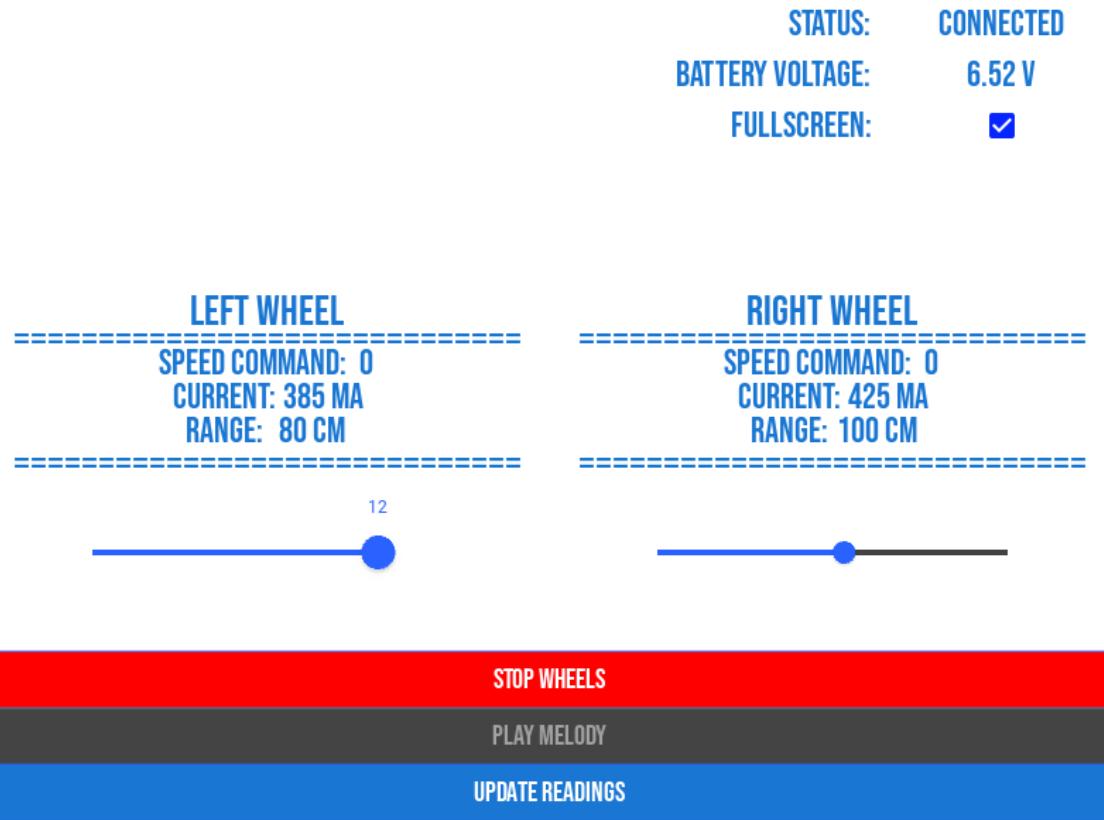


Figure 3.22: Screenshot of the GUI used to control the car.

3.7.1. Problems and Future Improvements

There were some other problems apart from the readings not being able to refresh automatically. For some reason the current readings from the wheels were very similar, even when the one wheel was at a complete standstill and the other spinning at full speed. The only possible reason for this could be an analogue problem when they are combined, since the two circuits worked perfectly when tested individually. Unfortunately, the buzzer would only be played when the wheels were not turning. Therefore, the button is greyed out for this case. In hindsight this is not a big problem, since the wheels are quite loud. Except for the aforementioned problems, the GUI was programmed to connect automatically to a port that was set up in the code. This is not optimal since different computers will use different COM ports. Furthermore, Tkinter would rather be used for similar tasks in the future, since it is older and therefore has a lot more support on the Internet. Unexplainable bugs are also less prevalent with Tkinter.

Chapter 4

Physical implementation

4.1. Current Sensor

4.1.1. Measured Results

The current sensor was redesigned to be used with a 3.3 V regulator instead of the planned 5 V regulator. This was to limit the voltage going into an ADC of a microcontroller to 3.3 V. To test the motor, 7.2 V was applied and the output voltage was checked to see if it corresponded to the designed value.

The motor current (far-left reading on the bench power supply), voltage over the motor and sense resistor (left-most multimeter), and output voltage of the current sensor (right-most multimeter) was observed simultaneously.

When the circuit was redesigned it was designed for 300 mA corresponding to 3.3 V. Only the gain and supply voltage was altered, and not the rest of the original design. The voltage supply of 3.3 V to the op amps then clamped the output voltage of the sensor to 3.3 V when the motor is stalled (which worked as designed according to Figure 4.1b). The free-running current of 190 mA should yield an output voltage of $(190 / 300 * 3.3) = 2.09$ V. The result is 2.67 V (Figure 4.1a), which is not too accurate.



(a) Free-running

(b) Stalled

Figure 4.1: DC motor under various test conditions.

4.1.2. Circuit Photograph

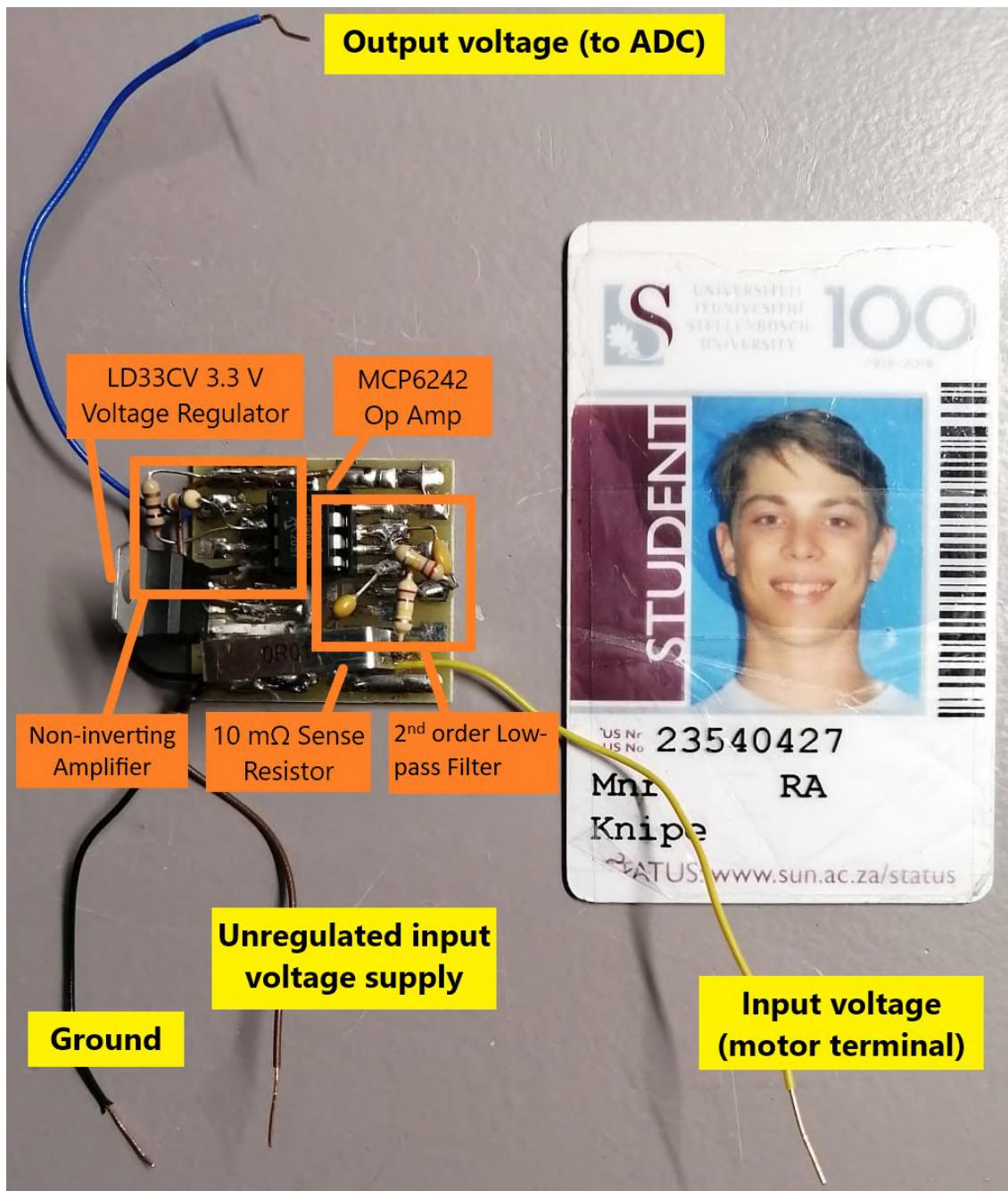


Figure 4.2: Current sensor: built circuit.

4.2. Ultrasonic Range Finder

4.2.1. Circuit Photograph

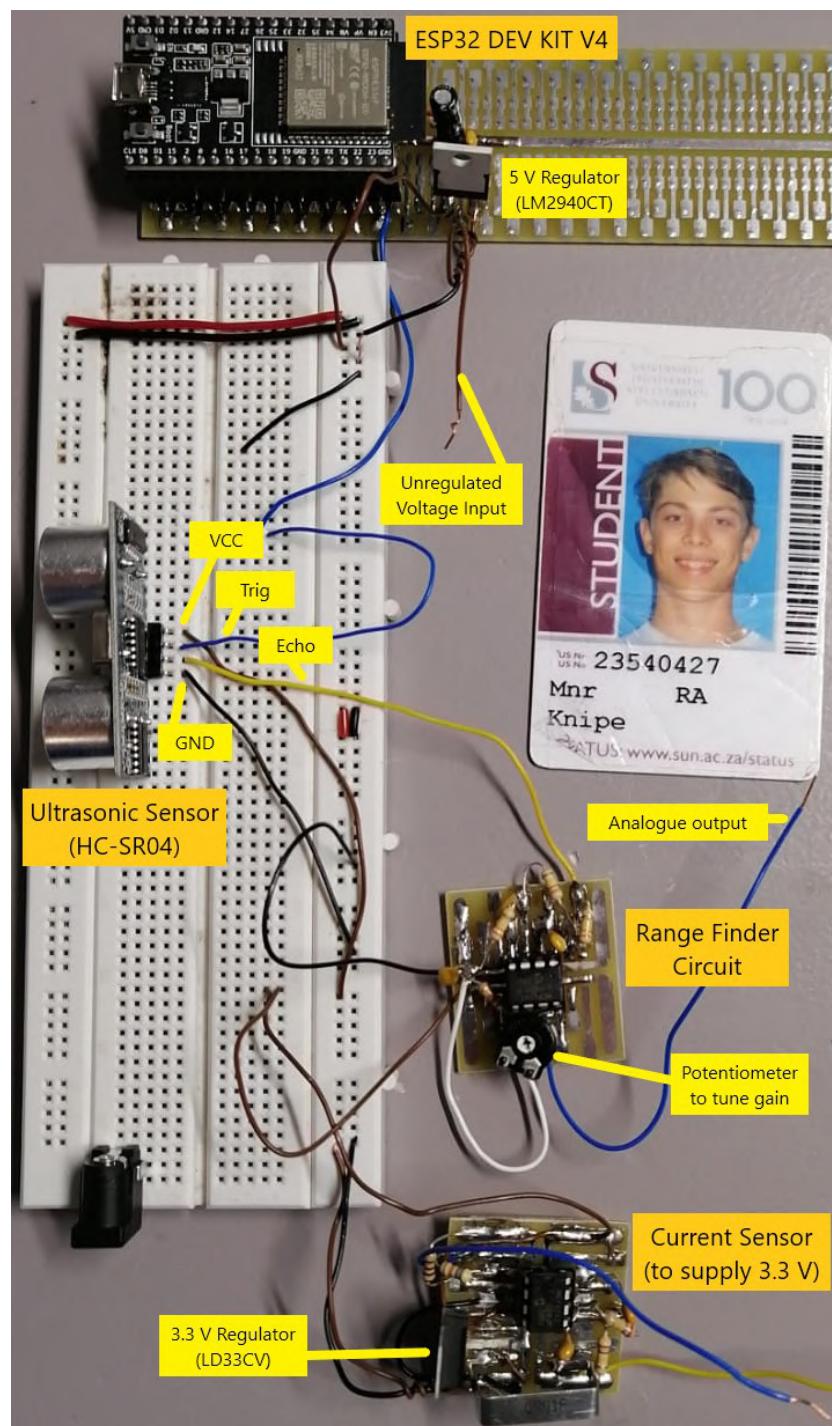


Figure 4.3: Ultrasonic range finder sensor: built circuit.

4.3. Analogue Control of Right Motor

4.3.1. Circuit Photograph

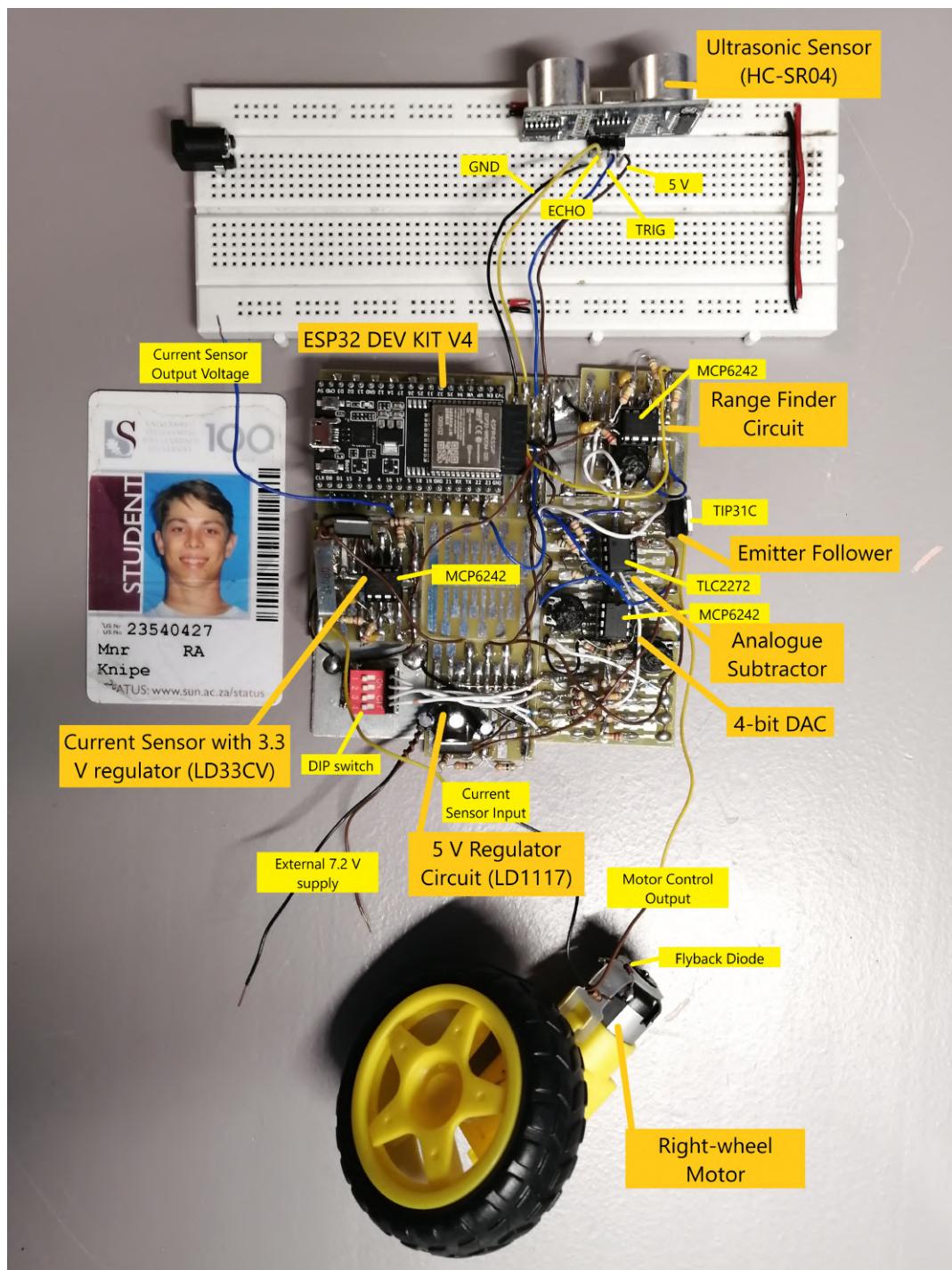


Figure 4.4: Analogue Control of Right Motor: built circuit.

4.4. Digital Control of Left Motor

4.4.1. Circuit Photograph

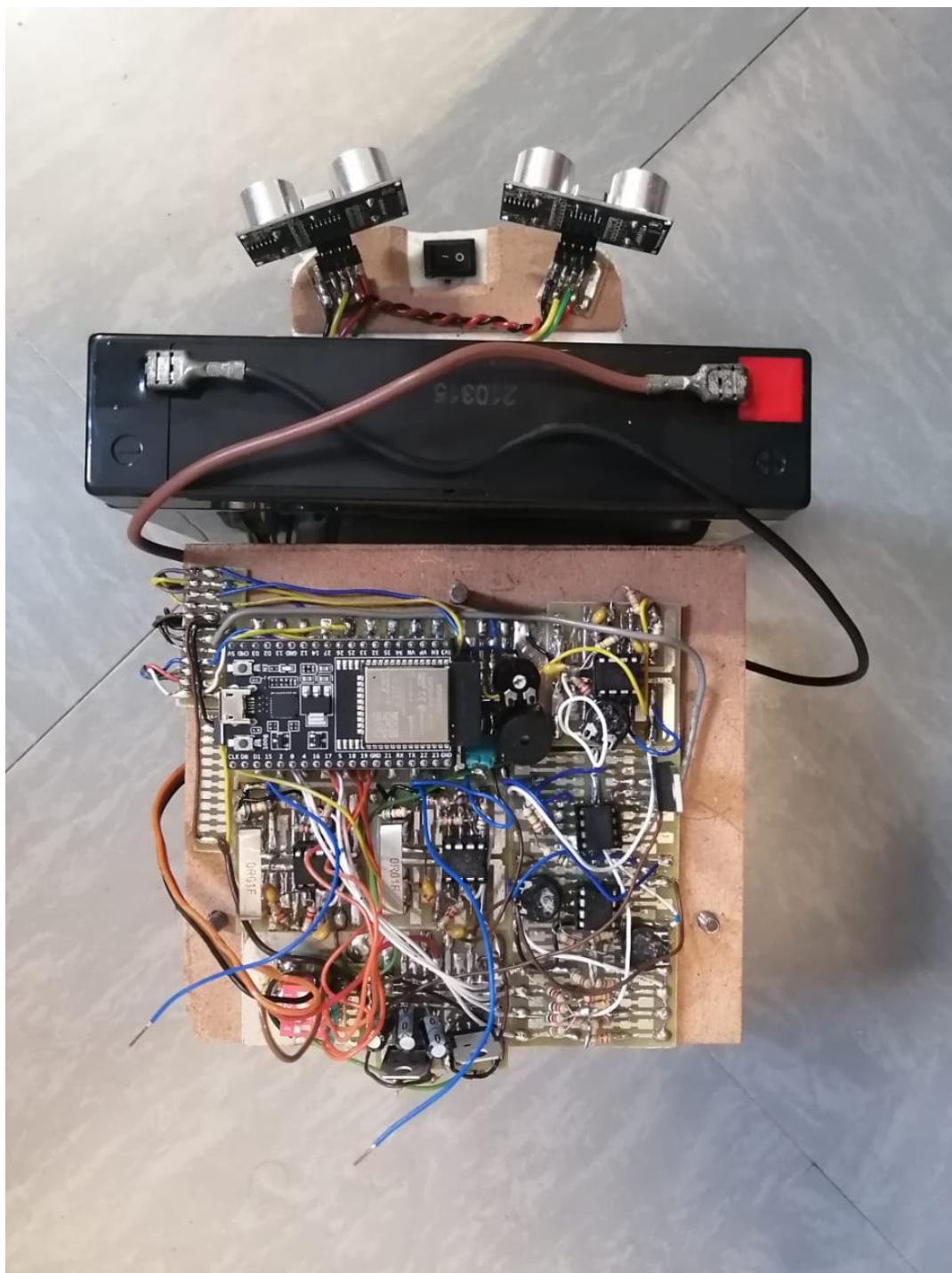


Figure 4.5: Digital Control of Left Motor: built circuit.

Bibliography

- [1] L. Loflin, “Pulse width modulation power control.” [Online]. Available: https://www.bristolwatch.com/ele/pulse_width_modulation.htm
- [2] R. Keim, “Low-pass filter a pwm signal into an analog voltage,” 2016. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/>
- [3] R. W. World, “Difference between dac types-weighted resistor,r-2r ladder,” 2012. [Online]. Available: <https://www.rfwireless-world.com/Terminology/Difference-between-DAC-types.html>
- [4] I. Rectifier, “Irf530npbf hexfet® power mosfet.”
- [5] F. Semiconductor, “2n7000 / 2n7002 / nds7002a n-channel enhancement mode field effect transistor.”
- [6] ——, “Fqd13n06l / fqu13n06l 60v logic n-channel mosfe.”
- [7] I. Rectifier, “Irf9z24npbf hexfet® power mosfet.”
- [8] Microchip, “50 a, 550 khz rail-to-rail op amp,” 2008, p. 3. [Online]. Available: <https://www.microchip.com/en-us/product/MCP6242>
- [9] J. Meneu, “Top 10 fundamental op amp circuits,” 2016. [Online]. Available: <https://static4.arrow.com/-/media/arrow/images/miscellaneous/1/1116-op-amp-fun-image-5.jpg?la=it-it&h=320&w=537&hash=44E9C4C198DD210F6B621E977AE7BD20>
- [10] ——, “Top 10 fundamental op amp circuits,” 2016. [Online]. Available: <https://static4.arrow.com/-/media/arrow/images/miscellaneous/1/1116-op-amp-fun-image-7.jpg?la=it-it&h=412&w=600&hash=F6A114CDBF70162F675AFE4A54129442>
- [11] ——, “Top 10 fundamental op amp circuits,” 2016. [Online]. Available: <https://static4.arrow.com/-/media/arrow/images/miscellaneous/1/1116-op-amp-fun-image-13.jpg?la=it-it&h=473&w=600&hash=04073DFFDF2BB46D4864118BC3CDECE3>
- [12] Lastminuteengineers, “How hc-sr04 ultrasonic sensor works interface it with arduino,” 2022. [Online]. Available: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>
- [13] G. Anjitha, “Obstacle avoidance robot using fpga,” 2020. [Online]. Available: https://www.researchgate.net/figure/Timing-Diagram-for-HC-SR04-Ultrasonic-Sensor_fig2_341480237

- [14] ElecFreaks, “Ultrasonic ranging module hc - sr04.”
- [15] R. Keim, “Logic signal voltage levels.” [Online]. Available: <https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/>
- [16] Espressif, “Esp32 series.”
- [17] R. Santos, “Complete guide for ultrasonic sensor hc-sr04 with arduino,” 2013. [Online]. Available: <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- [18] Dejan, “Ultrasonic sensor hc-sr04 and arduino – complete guide,” 2017. [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
- [19] T. W. Rob Toulson, “Pulse width modulation,” 2012. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/pulse-width-modulation>
- [20] C. Hope, “What are the advantages and disadvantages of nicad batteries?” 2022. [Online]. Available: <https://www.computerhope.com/issues/ch000351.htm>
- [21] I. Buchmann, “Bu-403: Charging lead acid,” 2022. [Online]. Available: <https://batteryuniversity.com/article/bu-403-charging-lead-acid>
- [22] GreatScott, “Battery type comparison —— lead acid vs nimh vs li-ion vs lipo,” 2016. [Online]. Available: <https://www.youtube.com/watch?v=LqgP16JQ24I&t=228s>
- [23] I. Buchmann, “What’s the best battery?” 2022. [Online]. Available: <https://batteryuniversity.com/article/whats-the-best-battery>
- [24] Adafruit, “Dc gearbox motor - ”tt motor” - 200rpm - 3 to 6vdc,” 2022. [Online]. Available: <https://www.adafruit.com/product/3777#description>
- [25] E. Tutorials, “Active low pass filter,” 2019. [Online]. Available: https://www.electronics-tutorials.ws/filter/filter_5.html
- [26] A. Devices, “Analog filter wizard,” 2022. [Online]. Available: <https://tools.analog.com/en/filterwizard/>
- [27] T. Photon, “Why does the capacitor need to be small in rc filter circuits?” 2020. [Online]. Available: <https://electronics.stackexchange.com/questions/478778/why-does-the-capacitor-need-to-be-small-in-rc-filter-circuits>
- [28] S. Microelectronics, “Power transistors.”
- [29] D. Inc, “Wide input voltage range, 50ma uldo regulator.”
- [30] S. Microelectronics, “1.2 v to 37 v adjustable voltage regulators.”

- [31] ——, “Adjustable and fixed low drop positive voltage regulator.”
- [32] J. Lewis, “7 mosfet myths and misconceptions addressed,” 2019. [Online]. Available: <https://www.baldengineer.com/7-mosfet-myths-and-misconceptions-addressed.html>
- [33] loboris, “Esp32 maximum pwm frequency,” 2015. [Online]. Available: <https://forum.micropython.org/viewtopic.php?t=3717>
- [34] R. Santos, “Esp32 with hc-sr04 ultrasonic sensor with arduino ide,” 2021. [Online]. Available: <https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/>
- [35] ——, “Esp32 pwm with arduino ide (analog output),” 2018. [Online]. Available: <https://randomnerdtutorials.com/esp32-pwm-arduino-ide/>
- [36] R. Nave, “The schmitt trigger,” 2022. [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/schmitt.html>

Appendix A

Social contract



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

E-design 344 Social Contract

2022

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.

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.

I, 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.

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.

Prof. MJ (Thinus) Booyens

MJ Booyens
Signature:

Student number: 23540427

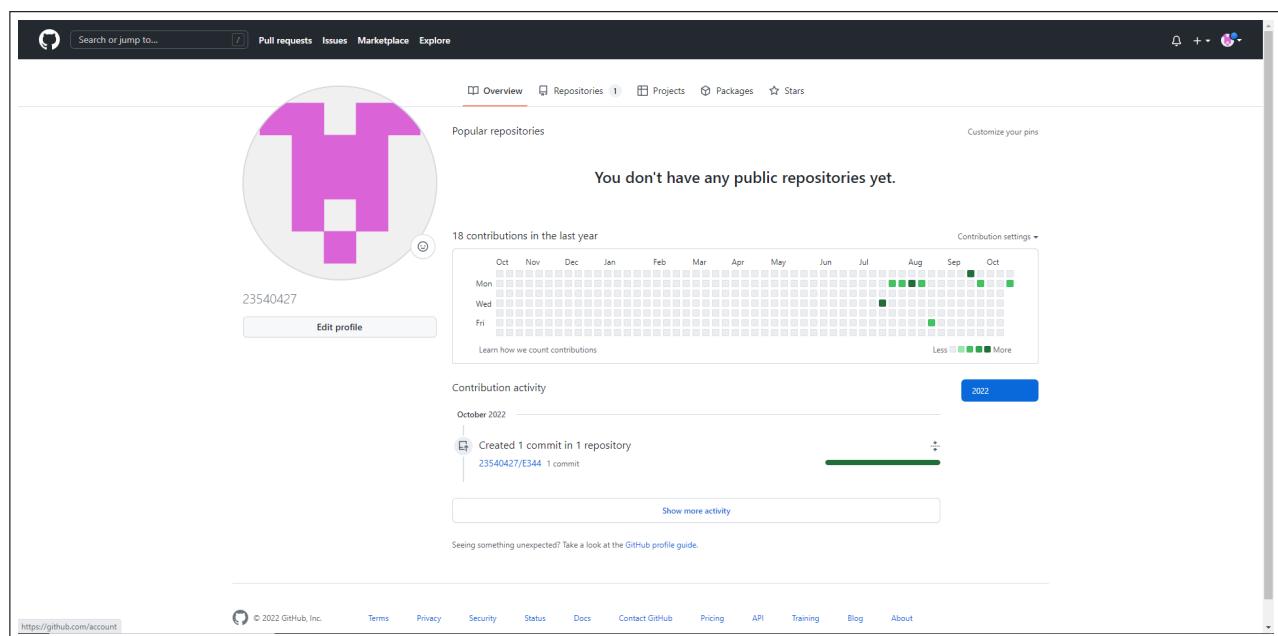
Digital signature by Richard Knipe
Richard Knipe
Date: 2022.07.20
17:00:45 +02'00'

Date: 1 July 2022

Date: 20 July 2022

Appendix B

GitHub Activity Heatmap



Appendix C

MOSFET Comparison

	IRF530	2N7000	FQU13N06L	IRF9Z24N
Type	NMOS	NMOS	NMOS	PMOS (Hence not an option)
Min $R_{DS(on)}$ [mΩ]	90	1200	115	
Max I_D (pulsed) [A]	60	0.5	44	
Max V_{DSS} [V]	100	60	60	
Min $V_{GS(on)}$ [V]	4	3	2.5	

Table C.1: Parameters considered when choosing the MOSFET. [4] [5] [6] [7]

The lowest $R_{DS(on)}$ for the lowest voltage drop (and hence less voltage over the motor), a max I_D above 1.5 A (the stall current of the motor), a max V_{DSS} above the battery voltage of 7.2 V, and a min $V_{GS(on)}$ of below 3.3 V (the output voltage of a GPIO pin on the ESP32) were all things to consider.

Appendix D

GitHub Version Control

The screenshot shows the GitHub repository history for branch `main`. The commits are organized by date:

- Commits on Oct 17, 2022:**
 - AB (commit `2f677fe`) - 23540427 committed 6 minutes ago
- Commits on Sep 26, 2022:**
 - a6 (commit `5d358ef`) - 23540427 committed 21 days ago
- Commits on Sep 18, 2022:**
 - A5 Included gate- and pull-down resistors (commit `95ff57c`) - 23540427 committed 28 days ago
 - A5 circuit and code (commit `0542063`) - 23540427 committed 28 days ago
- Commits on Aug 26, 2022:**
 - A4 (commit `ac4509c`) - 23540427 committed on 26 Aug
- Commits on Aug 15, 2022:**
 - Updated and uploaded SPICE files ... (commit `38fadf0`) - 23540427 committed on 15 Aug
- Commits on Aug 8, 2022:** (No commits shown)