SAN FRANCISCO STATE UNIVERSITY

SCHOOL OF ENGINEERING

ENGR 478-02
Final Project Lab Report
Spring 2018

By
Ahmad El Shakoushy

Alexander Louie

Table of Contents

Introduction: While brainstorming possible project ideas; we had a vision of designing and implementing a semi autonomous robotic car. Our reason for wanting to design such a vehicle was to create a small-scale foundational prototype that could in theory be translated and ameliorated into an actual autonomous vehicle. We eventually ended up adding additional functionality on top of what we had envisioned in order to make it more realistic. For example, we added a manual control mode, UART interfacing to communicate wirelessly, and wireless dual mode switching. These specific additions will be discussed in great detail in the 'Design and Implementation' section of this report.

We conducted a lengthy amount of research regarding possible configurations and sensors in the beginning of March. Our visions started to solidify themselves two weeks later and we are able to work together in order to plan out a finalized project plan. This was a crucial point in our project, because we had to scour datasheets in order to determine which pins were available and how we would integrate our components. We firmly believe that it is our firm commitment to this planning phase that allowed our project to turn out as successful as it did.

The planning and testing phases of our project allowed us to ensure that our project had a certain amount of sustainability attached to it. One of the factors that showcase the sustainability of our design is that our HC-SR04 sensors are freely placed on individual mini breadboards instead of being physically mounted to the chassis. This may not sound like it makes a difference but it makes a huge difference because a user can swap the sensors out within seconds if needed. There are many other demonstrations of sustainability within our project, and we feel that listing them out here would be too tedious with regards to getting to the main point across. Fundamentally, the sustainability of our design allows users to quickly swap components out and replace them with others. However, it is important to note that we could add even more sustainability to our design. A prime example of this would be the design of a printed circuit board to reduce the amount of wires and breadboards used in the project. An added advantage of using a PCB board would mean that users can debug the circuitry and track issues with the hardware in a rapid manner. We would most definitely undertake the design and implementation of a PCB, enclosure, among other sustainability additions if we were to bring our device to market. It is of utmost importance to mention that the aforementioned features would provide our project with an added degree of sustainability, but their exclusion does not diminish the functionality of our project as it is meant to serve as a functioning foundational prototype.

After doing our research during the planning phase; we determined that there have been similar projects in the past. However, most of them were done with Arduinos and we did not find a single project that implemented the wireless dual mode switching logic that we added to our initial vision. We believe that this unique, but important feature is what sets us apart from similar projects. The wireless dual mode switching logic is important because it allows the user to almost instantaneously take over control of the car in the rare cases where that may be necessary. This may not seem to matter much for our small-scale model, but it has massive implications when it comes to the testing of an actual full-sized autonomous vehicle. It would be prohibitively expensive and possibly dangerous if such vehicles were to not have the capability of allowing the user to quickly switch to manual control mode and prevent a collision. Not having a system that includes the aforementioned switching mechanism would imply that an actual driver would need to be present in the car in order to take over from the autonomous system. Such a vehicle could be potentially deadly to the individual in the driver's seat. Our design could provide manufacturers with a safer and more streamlined model to test their autonomous vehicles.

Design and Implementation: This section of the report serves as both an introduction to the methodologies that we utilized in designing and implementing the hardware component of our project. We will give an overview of our design, then, discuss our specific design decisions in greater detail within the 'Hardware Implementation' section of this report.

i) Hardware Design: Before this stage, we had already solidified our ideas and prepared the following schematics for each board:

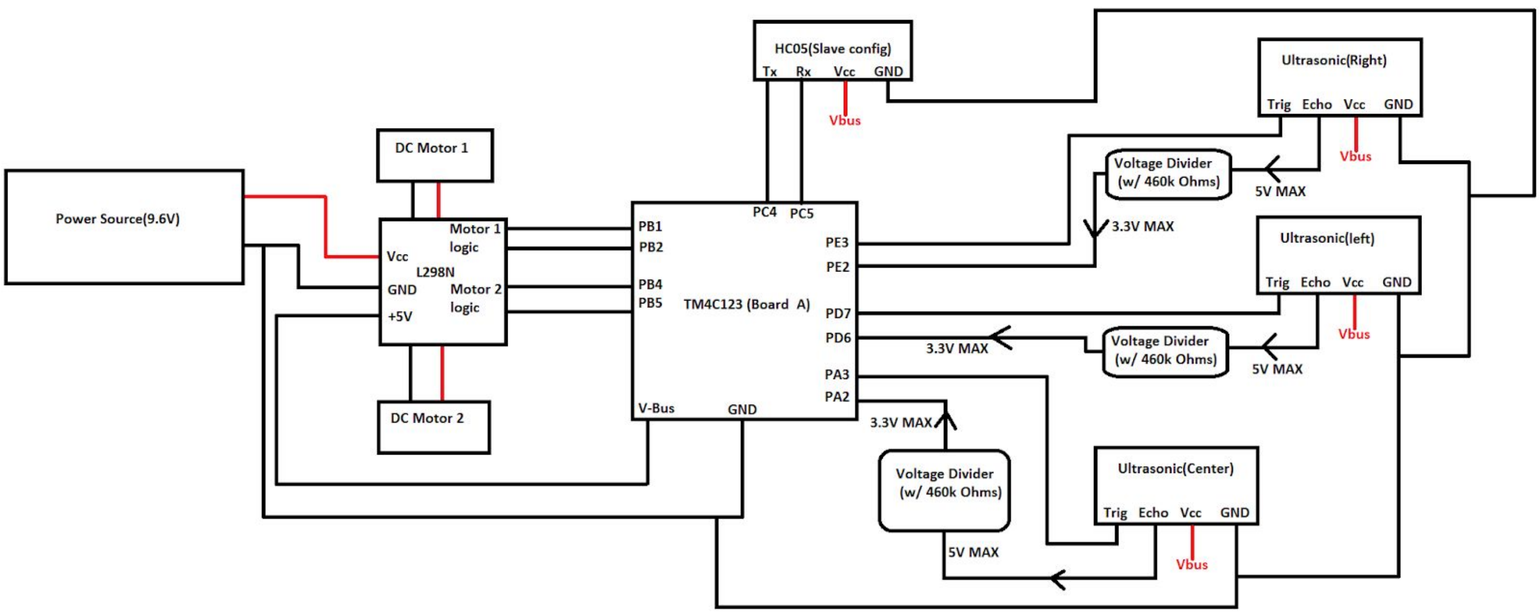
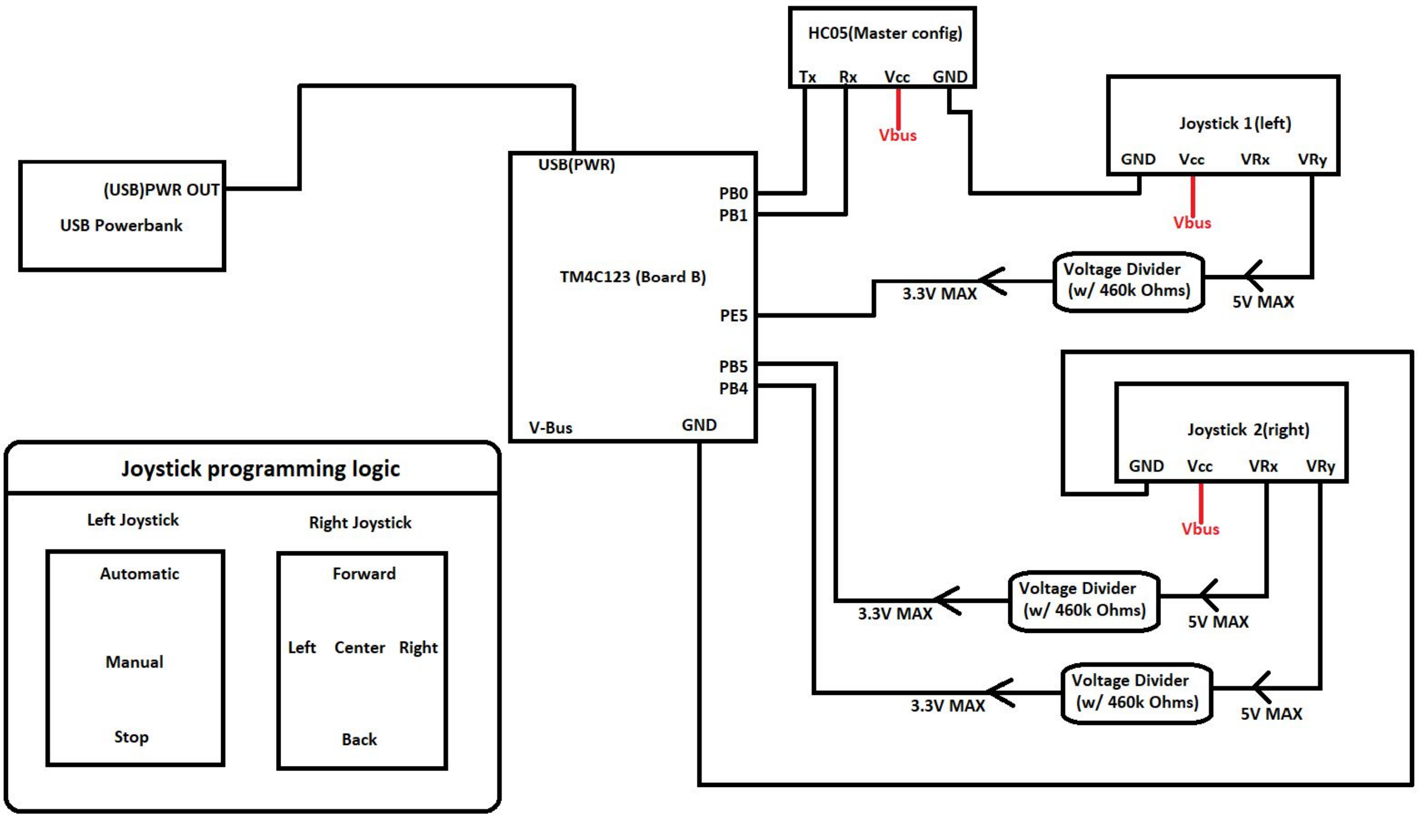


Figure (A): This figure shows the schematic for the actual robot car. All the components shown on this schematic were mounted onto our chassis. The heart of this schematic is a TM4C123 microcontroller which we designated as 'Board A' for simplicity. We have a 9.6V power source which consists of a 6 AA battery holder pack. To the right of our power source, we have the L298N dual H-Bridge which is connected to each of the motors. The L298N is an integral part of this project because it allows us to supply the required DC current to run both of our brushless DC motors. Our TM4C123 microcontroller does not supply enough current to drive the motors, so the L298N was a necessity. We also have an HC-05 bluetooth module in slave configuration to communicate with our secondary board(Board B). The rightmost section of our schematic showcases the three ultrasonic sensors and their respective voltage dividers. We have come to the conclusion that the voltage dividers are unnecessary, and we will elaborate on this point in the Hardware Implementation section of this report.



Figure(B): This figure shows the schematic of our setup for the secondary(manual mode) part of the project. As in Figure (A), the heart of this schematic is a TM4C123 microcontroller. We have decided to designate the board used in this part of the project as Board B. The power source for Board B is a USB Power Bank which is connected via microUSB. The HC-05 bluetooth module shown in this schematic is in Master configuration in order to send data to the Slave module shown in Figure(A) via motion of the joysticks. The two joysticks on the right of the diagram are analog joysticks, and each one requires a voltage divider because the maximum voltage of the on-board ADCs used to sample the joysticks is 3.3V. Programming the joysticks was a very difficult task, so we have added our joystick programming logic in the bottom-left corner of the schematic.

ii) Hardware Implementation: This section of the report serves as an in-depth analysis and discussion section in which we offer a visual and textual representation of how we actually implemented the hardware from the 'Hardware Design' section.

a) **Primary Board:** We first started off by testing our 2 brushless DC motors by applying the proper voltage difference across the wires. This was done to ensure that our motors were functioning properly before we would interface them with the L298N. Our next course of action was to use the screws of the terminal block to secure the wires of each motor to their respective positions. It is important to note that it wouldn't matter which wire of each motor went into which pin for that specific motor on the L298N because the logic could simply be switched in our software implementation. The next step was to connect wires from logic pins(PB1, PB2, PB4, PB5) into the logic input pins of the L298N. It is important to note that the L298N has an enable input which could use pulse width modulation for speed control. These pins on the board had jumpers which fixed them to HIGH(which we measured at 5V with a DMM). We decided to keep these jumpers on and to use our car as a fixed speed vehicle because the PWM wouldn't offer much of an advantage for our goal. Using PWM on a small scale model like this wouldn't offer much of an advantage in terms of speed control for an actual autonomous car that would be based off our foundational design. Figure (C) below shows the previously mentioned connections of the L298N along with the voltage divider for the center ultrasonic sensor.

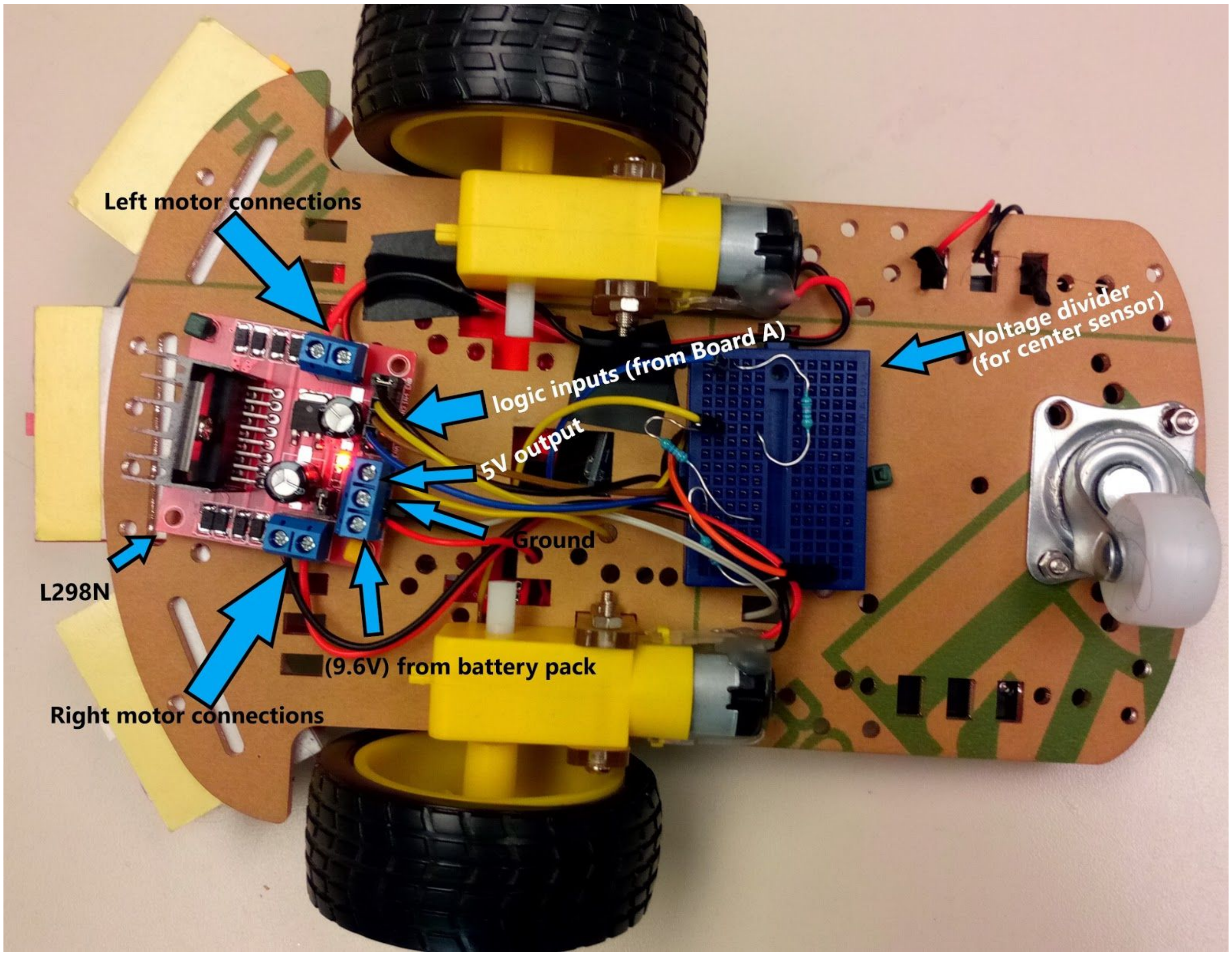


Figure (C): This figure shows a visual representation of our aforementioned L298N hardware wiring configuration. It is important to note that we determined that the voltage divider is not necessary for the ultrasonic sensors because the TM4C123 can actually take a maximum input of 5V as opposed to a maximum of 3.3V as we thought. Our reason for this misstep is because the onboard ADCs have a maximum input of 3.3V and we naively assumed that this would be the same for the GPIO pins of the board. Nonetheless, the presence of the voltage dividers does not diminish from the effectiveness of our project as it functions as expected in all manners.

The rest of the hardware connections for Board A were rather simple in nature: HC-05(slave configuration), ultrasonic sensors(left, right), and voltage dividers for the ultrasonic sensors. As mentioned previously, the voltage dividers for the sensors are unnecessary and can thus be removed without any danger to Board A. Figure (D) below highlights our hardware connections for the aforementioned components.

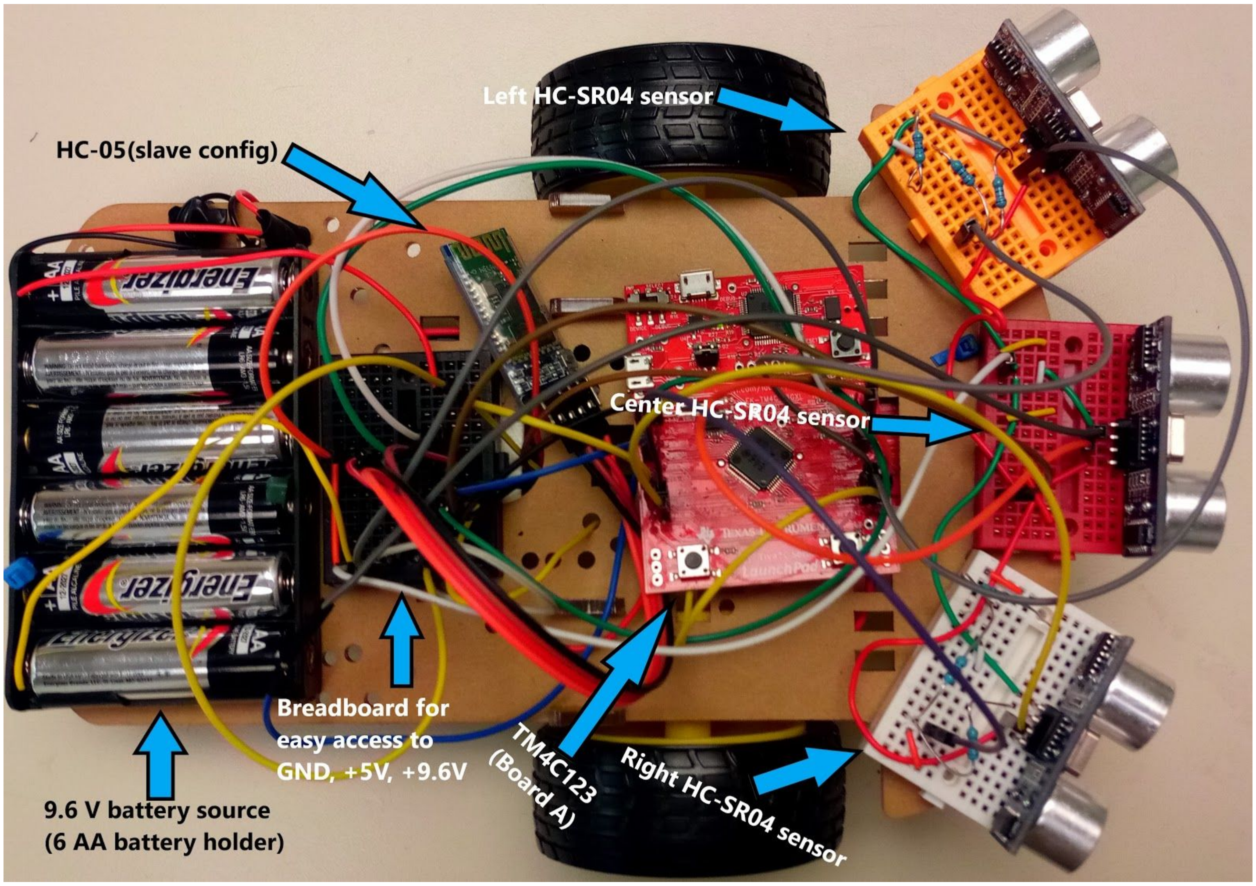


Figure (D): This figure shows a visual representation of our previously mentioned connections/components. We would have ziptied everything to make it neater but some of our wires were short and we didn't want to risk accidentally unplugging something. It may look like our connections are disorganized but that is not simply the case because we tried to connect everything as efficiently as possible(even going as far as utilizing an additional breadboard as shown in the figure). In conclusion, we feel that our wire organization is sufficient for the purposes of a base prototype that could be modified for future development.

b) Secondary Board: The setup for this board was significantly simpler than that of the Primary Board because the purpose of this board was to utilize UART interfacing in order to transmit directions for manual control to the car. A very important feature that we incorporated in our design was to allow for wireless dual mode switching, so the user can alternate between semi-autonomous and manual control mode almost instantly. We firmly believe that this single feature is what makes our design unique when compared to other similar projects. Figure (E) below serves as a visual representation of the connections that we made.

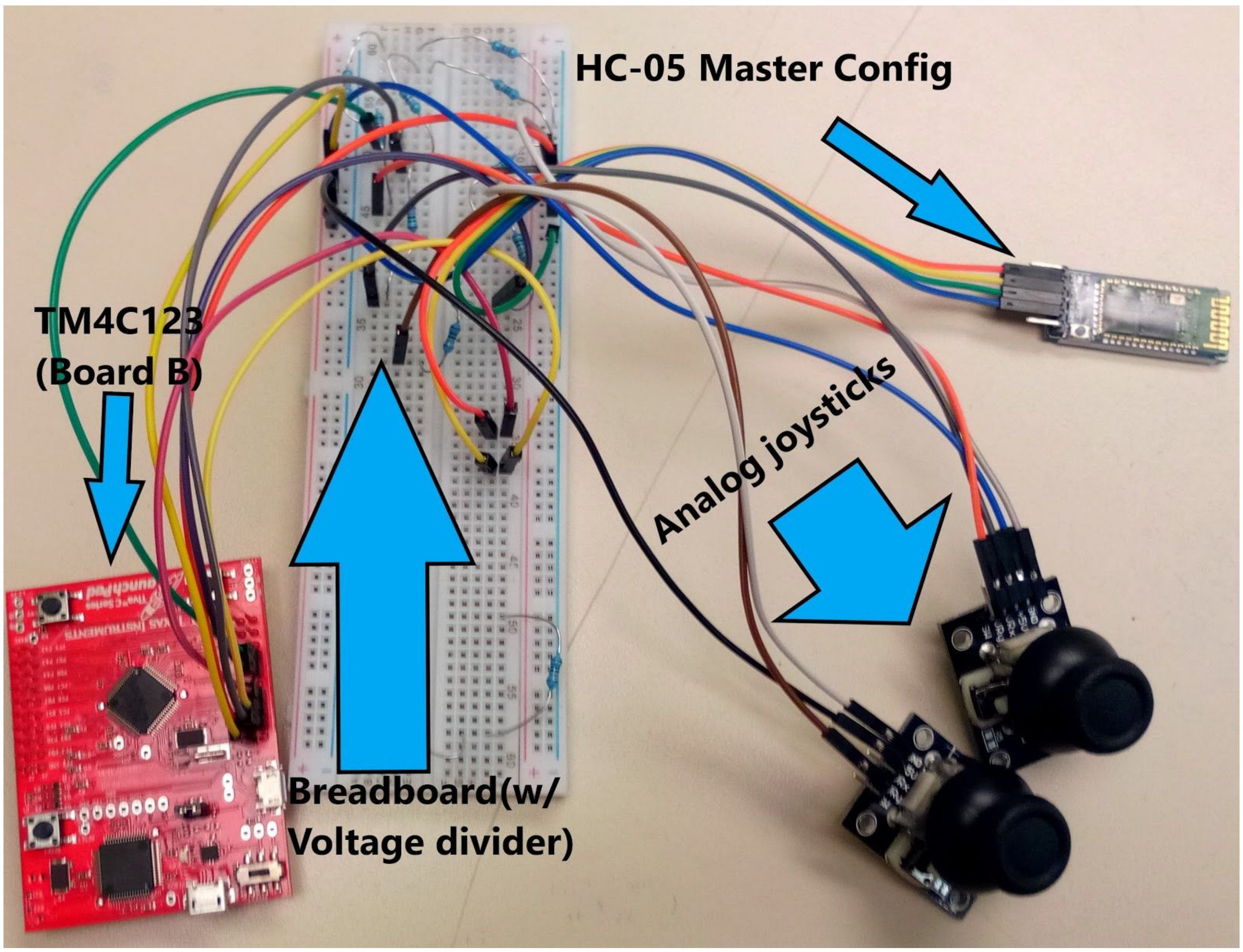


Figure (E): This figure serves as a visual representation of our connections for Board B and its respective components. It is important to note that there is a stray resistor on the bottom-right of our breadboard that was used for testing, so it was removed before our final presentation and serves no purpose for our setup. Our HC-05 for this secondary board was in master configuration ,so it could transmit characters such as ‘F’, ‘B’, and ‘S’ to our Slave HC-05. The specific characters transmitted and their meaning will be discussed in the ‘Software Design’ section of this report. The analog joysticks that we used have two analog output pins(1 for x-axis and 1 for y-axis, respectively). We did not utilize the x-direction on the left joystick as it was simply used to change states(automatic, manual, or stop) in a rapid manner. The right joystick serves the purpose of allowing for 360-degree swivel movement to control the car when it is in manual mode. We are using on-board ADCs in order to sample each of the joysticks' analog outputs. It is of utmost importance to mention that we built voltage dividers for each analog output of the joysticks in order to step the voltage down from 5V to 3.3V, so the ADC doesn't get damaged. We decided to utilize 460 kΩ resistors for this purpose because they are large enough to prevent a large draw of unneeded current from the board. There were some difficulties that we had to overcome in the programming of the analog joysticks. The aforementioned difficulties and our solutions to them will be discussed in great detail within the 'Results and Discussions' section of this report.

iii) Software Design: In this section of the report, we will showcase our software development flowcharts and give a summary of their meaning in order to provide readers with a wide-scoped view of our programming methodologies.

a) Primary Board(Main):

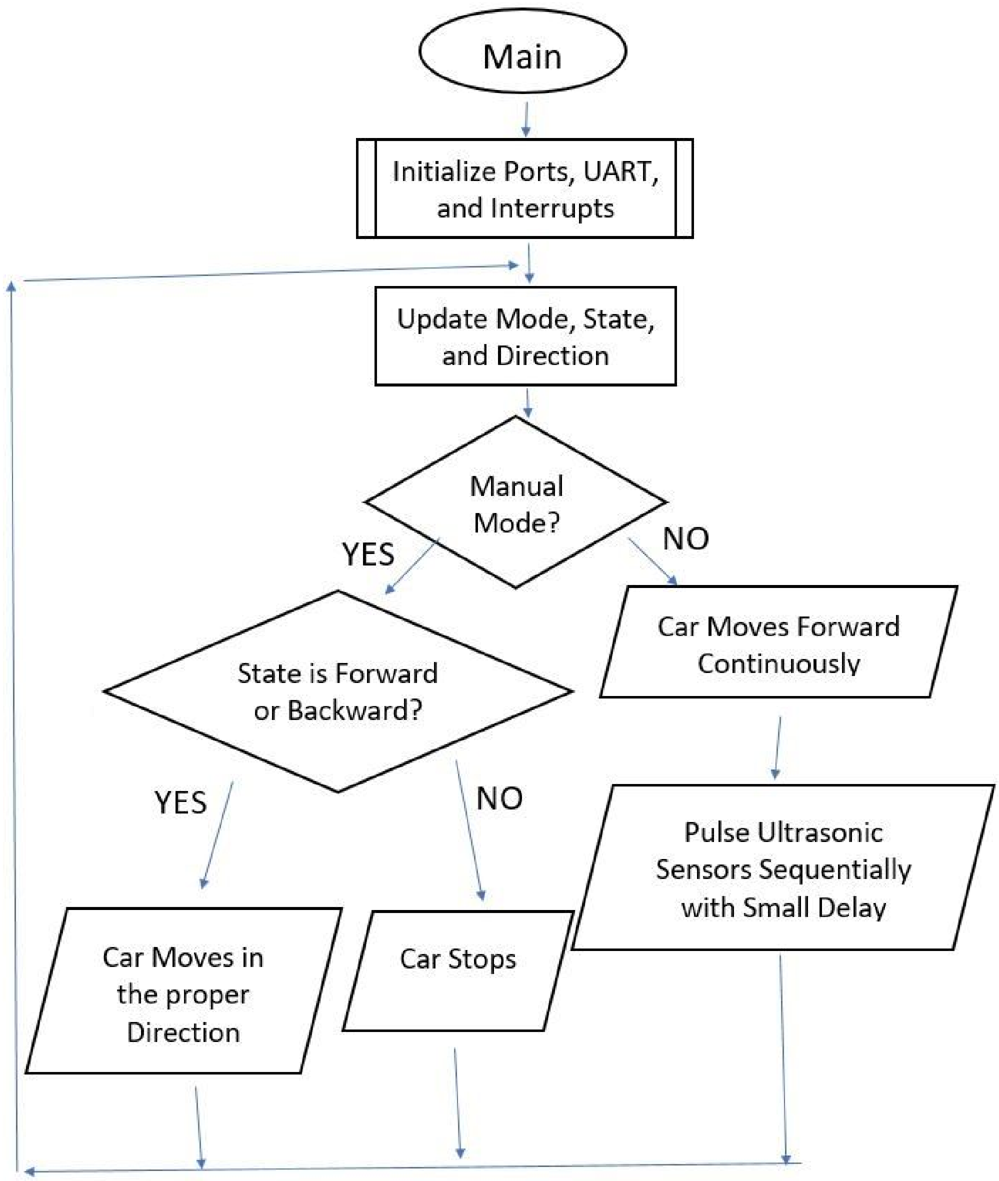


Figure (F): This figure shows the flowchart of our main method for our primary board(Board A). We first call our initialization functions for our ports, UART, and interrupts, respectively. The next step is to update our Mode, State and Direction char variables. It is crucial to note that the Mode refers to (Manual('M'), Automatic('A')), State refers to (Forward('F'), Backward('B'), Stop('S')), and Direction refers to (Left('L'), Right('R'), Center('C')), respectively. We then check if the car is manual mode(State = ‘M’). If the car is in Manual mode, we check if the State variable is either forward(‘F’) or backward(‘B’). The car will stop if the State variable’s value is neither of the aforementioned choices. Otherwise, it will move in the appropriate direction(s) based on the output of the ADC which is used to sample the output of the analog joysticks. If the car is not in Manual mode(‘M’) it would continue going forward, and the ultrasonic sensors would be pulsed sequentially with a short delay.

b) Primary Board(Handlers):

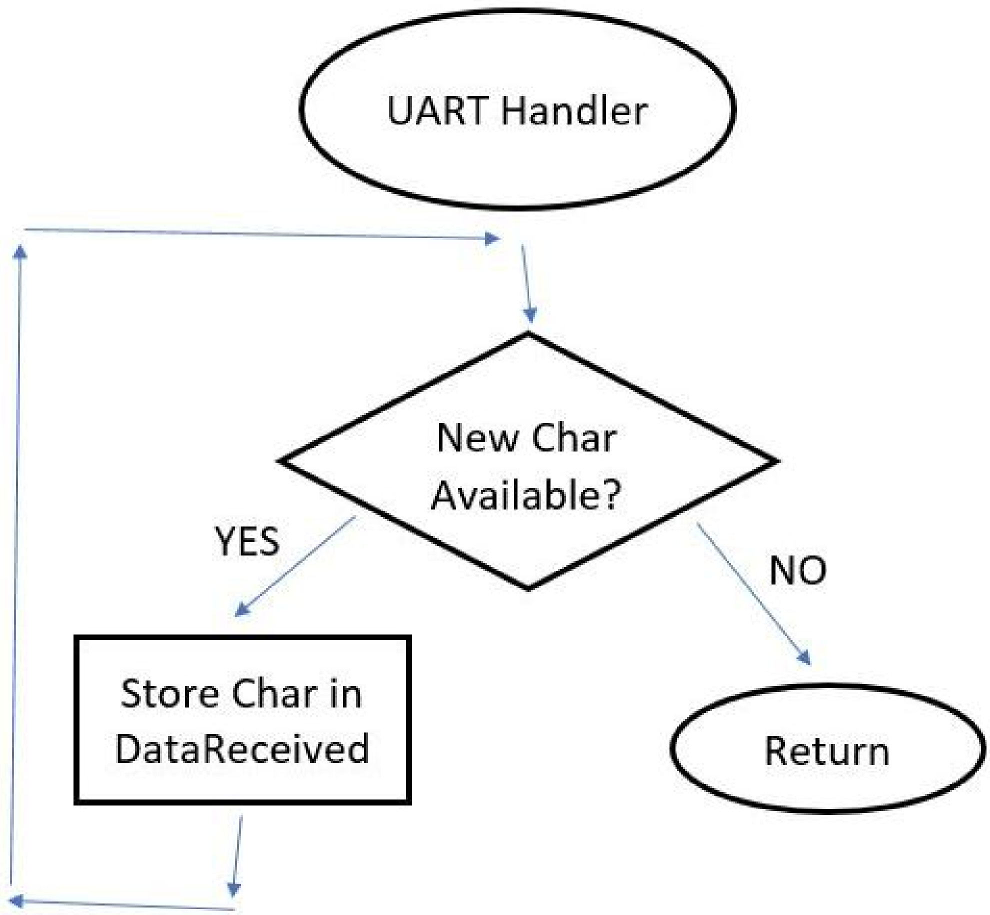


Figure (G): This figure shows our UART Handler(ISR). It is important to note that the code that this flowchart is showing is still on our primary board(Board A). We check if a new character is available once we enter the handler, and store that character into the DataReceived variable if it is available. The most recently received character is the most important. Therefore we are not worried about overriding a previous character stored in the DataReceived variable. Otherwise, we return from the Handler.

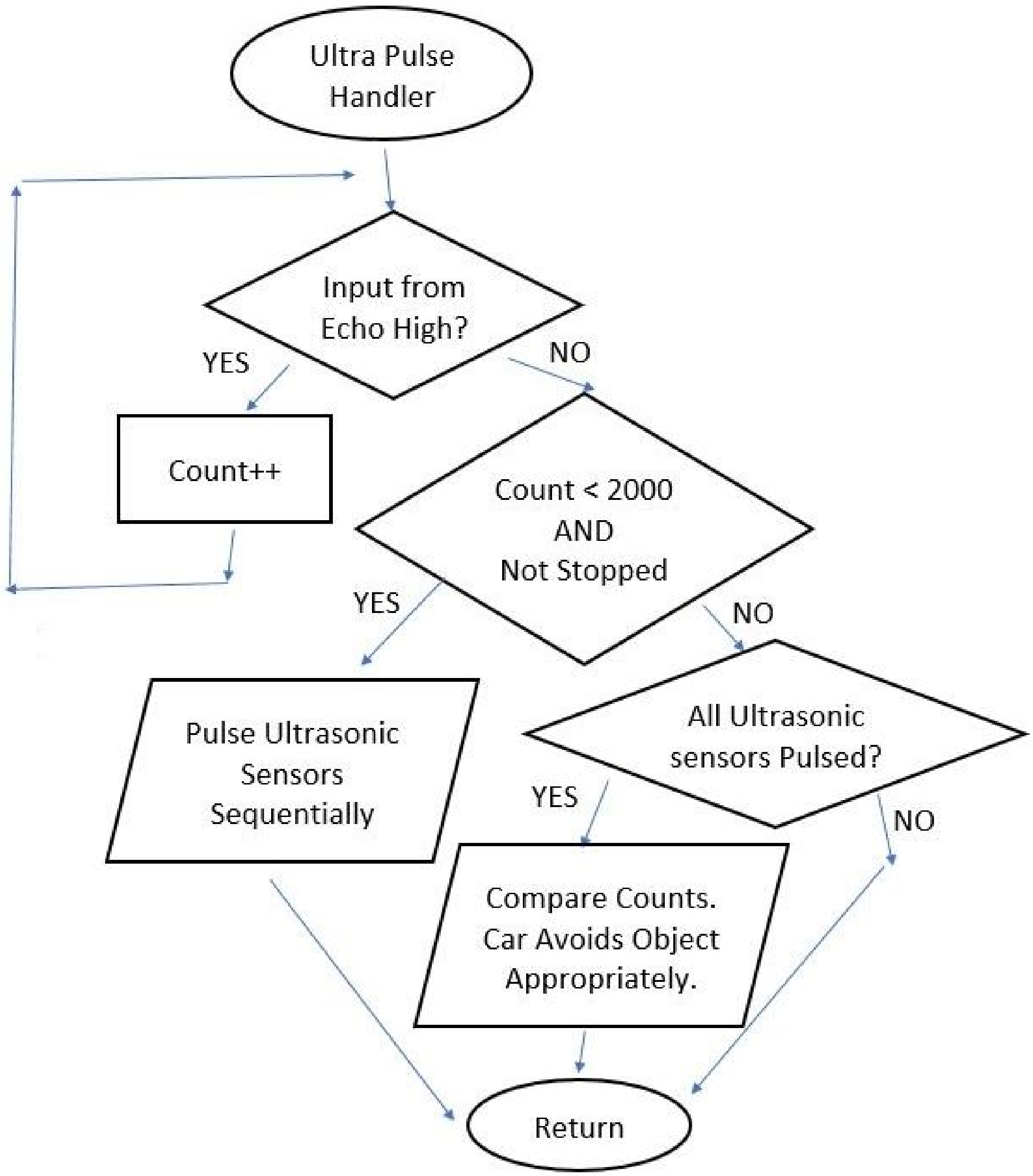


Figure (H): This figure showcases our software design logic for one of the Ultrapulse handlers which resides on Board A. We will first introduce when this handler would be called, and then we will explain how our HC-SR04 ultrasonic sensors function. The handler will respond to the input from the ultrasonic sensors by making appropriate adjustments to move the car or measure distances. The corresponding handler would only run if there is a rising edge on either PE2, PA2, or PD6. The significance of these pins is that they are the GPIO input pins that we are using to receive inputs from the sensors. The way the HC-SR04 sensor works is that it receives a 3.3V pulse of width 10 µs as input(on trig pin), and the sensor then sends 8 rapid ultrasonic pulses.

The receiver part of the sensor will receive any of those pulses if they bounce back meaning that an object is present. It is important to note that the required input pulse width of 10 µs, and the fact that 8 ultrasonic pulses are sent out each time we 'pulse' the sensor was set by the manufacturer, and not ourselves. The sensor will output a pulse from its 'Echo' pin only if it receives some of the ultrasonic pulses back; meaning that the sensor will only send back a pulse if it detects an obstacle. An important fact about the pulse that we receive back from the sensor is that its width is proportional with the actual distance of the detected object. We utilize this fact to calculate the proper distances from the count variable.

c) Secondary Board:

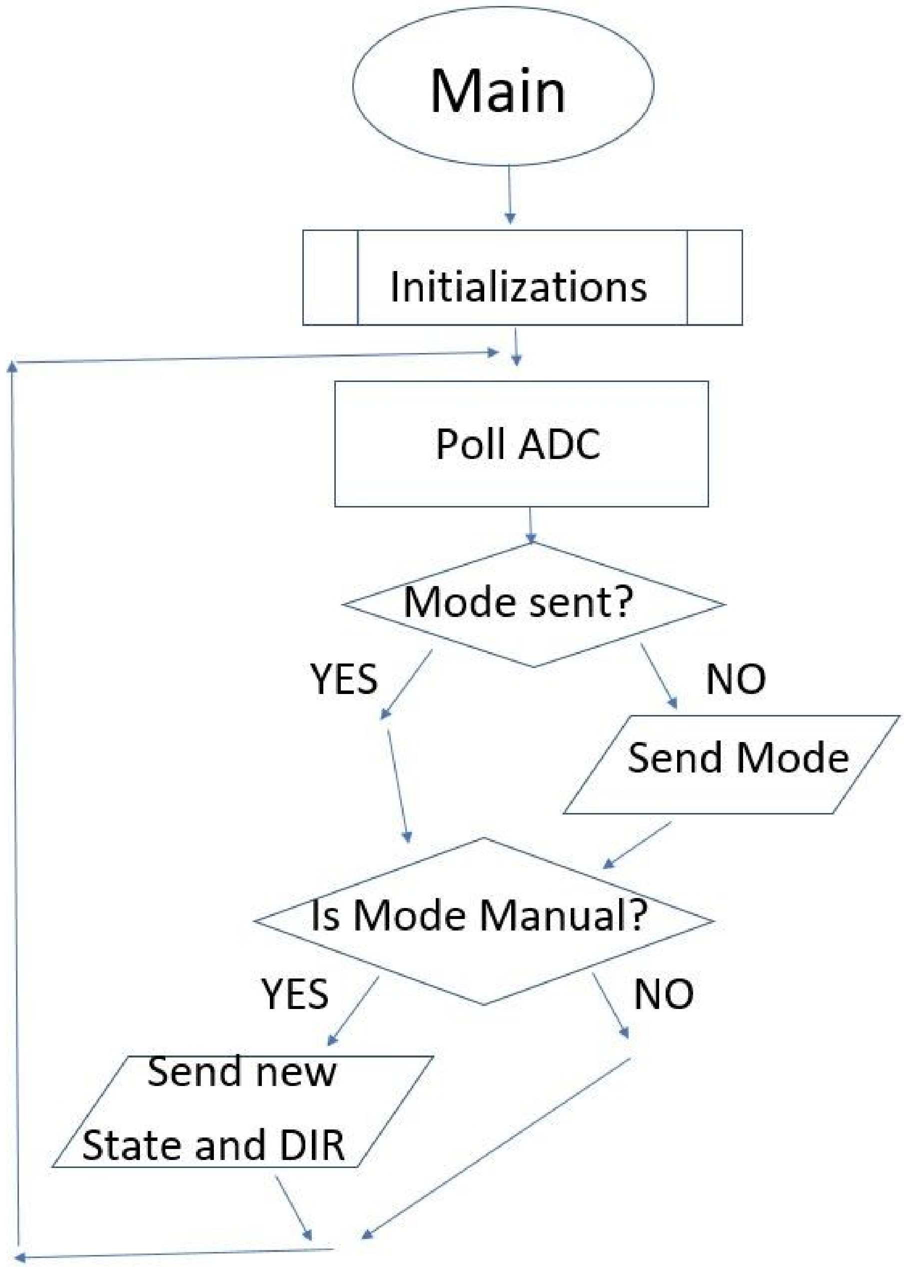


Figure (I): This figure shows our flowchart for the secondary board(Board B). Our first step in main is to initialize everything that needs initialization. We then poll the ADC, and check if the mode was sent previously to the slave HC-05 module on Board A. The mode is sent if it is not sent already. The next step is to check if we are in manual mode, and send the new State and Direction if that is the case.

Results and Discussion: We are proud to say that we have fully achieved all of the goals that we outlined in our research proposal. Furthermore, we were able to add additional functionality such as the manual control mode(w/ two analog joysticks) and the feature of wireless dual mode switching.

We faced an array of obstacles and setbacks during our journey of working on this project. One problem that we faced was that our ADC wasn't showing the proper number of levels when sampling one of our joysticks. An oscilloscope, DMM, and various other equipment was used to probe and debug this issue. We also reached out to Dr. Zhang for some guidance regarding this problem. After having tried using different ADCs, sample sequencers, circuit configurations, and joysticks; we determined that the issue was likely a hardware problem with the board that we were using. Our eventual solution for this issue was to realize that the number of levels on the problematic side was changing but in a limited range, so we picked appropriate values that would allow us to work around the problem. We faced many difficulties in completing this project and listing all of them out would probably take more space than we would need to get the main point across. While our project is a success; it was definitely not an easy one. Many hours were spent in the planning phase of the project, and we also set aside 7 hours per week for the last 6 weeks to get together and debug any issues that we faced individually. In conclusion, we were able to excel at this project because we had a well-structured plan in place regarding how to work together and utilize our personal strengths.

Various equipment/tools such as an oscilloscope, the debug watch window of Keil µVision, and a DMM were used to test and verify many aspects of our design. We worked in conjunction with each other to ensure with a great degree of certainty that all components were tested on an individual level before being integrated with the rest of our system. This was done to prevent us from running into a nightmare scenario where we have a fully assembled project with many components and wires, yet have no idea where a problem could be coming from. We feel that our methodical approach in testing each component individually helped us in achieving our goals sooner.

We feel that our project was very successful but could be improved via the addition of certain features, or the optimization of our existing hardware/circuitry. Firstly, we could determine the HC-SR04 output pulse width as a function of time rather than the number of clock cycles. This would ensure that our methodology for determining the object distance would work for any processor clock speed. Another possible improvement would be to use level shifters in place of the voltage dividers that we built for the joystick circuits in order to have a smaller form factor, less wires, and to ensure a small current draw. As previously mentioned, we remedied the issue of a potentially large current being drawn from the board by building the voltage divider circuits with large resistances.

Conclusion: Our project consisted of the design and implementation of a semi-autonomous dual mode motorized car. We encountered many difficulties along the way, but we were able to overcome all of them and work together as a team to add even more features than we had initially planned out. Regardless of how others may perceive it; we feel that this project was an immense success. We don't feel this way simply because everything worked at the end of the day, but rather because we had to challenge ourselves and think of possible solutions that we were faced with. This process of facing challenges and overcoming them has allowed us to bolster our knowledge of the TM4C123 microcontroller and of fundamental Electrical and Computer Engineering principles. This course, and especially this project has personally piqued our interest in the field of Embedded Systems and of Design with Microprocessors.

References:

1. ENGR 478 labs
2. UART demonstration by Dr. Zhang
3. TI Workshop Workbook
4. Dr. Valvano's ADC sample project
5. TM4C123 Datasheet