Howard University Department of Electrical and Computer Engineering

Team S.L.A.M Final Report

Morganne Veal, Cameron Lewis, Clifford Peeples, Jarrett Cunningham

April 25, 2019

Background

The problem with autonomous navigation is that we are reaching a computation limitation for our systems to detect objects, path plan, and make decisions in real-time. Dr. Amoo has done research in the field of autonomous navigation where she has explained what would be the best solution for this problem. In her literature review, she has concluded the Field Programmable Gate Arrays (FPGAs) are the best integrated circuit design to solve the main problem with autonomous navigation. For her to prove her theory, Dr. Amoo needs to first create baselines that will be used for comparisons to the final FPGA based autonomous platform. As members of her senior design VIP team, we have taken on the take of creating these baselines for her during this year.

Introduction

For this school year, our overall goal was to design two autonomous platforms, each having different control algorithms (Bang-Bang and Proportional Integral Derivative). To construct our autonomous platforms, we wanted to use commercial off the shelf (COTS) parts and design all of the circuits that complement each component. In this paper, we break down all the components of each design and what we learned. In the end, we want the reader to have a better understanding of how each component plays a role in autonomous navigation.

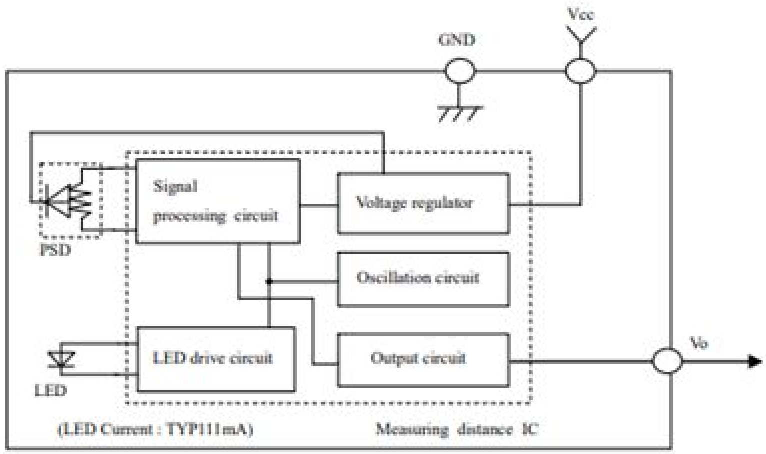
Sensors

IR Sensors

The IR Sharp Sensor GP2Y0A41SK0F is designed to measure distances of objects varying in size, color, and shape. The distance that the sensor can view ranges from 4cm to 30cm. This sensor can be used for autonomous platforms and in applications such as cleaning robots, self-driving cars, and more. For this specific project the sensor is used to help navigate a

Bang-Bang or PID controlled car. The purpose of the sensor is to direct the vehicle to either stop, turn left, turn right, or continue moving based on the distance of the object. The sensor analyzes the distance and based on how far it is it sends a signal to the motor controls. If the object is close to the sensor, the voltage will increase to tell the Bang-Bang or PID controller to decide which direction to move.

Schematic Design



- Vcc Input Voltage: The input voltage comes in from here and goes to the voltage regulator.
- Voltage Regulator: The voltage regulator takes the input voltage and disperses it through the circuit into usable voltage for the sensor.
- Oscillation Circuit: This part of the circuit takes the voltage from DC to AC and turns it into a signal that can be processed. It receives information from the signal processing circuit This signal is turned into a format that is able to be read as the voltage output
- Signal Processing Circuit: This processing circuit gathers information from the lenses of the sensor and sends it to the oscillation circuit which turns the information into data that tells the voltage to either increase, decrease, or stay the same based on the distances of the objects the lenses see.
- Output Circuit: The output circuit gives out information as voltage.
- LED Drive Circuit: The LED shows the user if the sensor is receiving power.

Sensor Testing

To test the sensor in this project, multiple objects were used that varied in size, color, and properties. The objects tested were:

- Green plastic soda bottle
- Sheet of glass
- Glass cup with water
- Glass cup without water
- Red t-shirt with white designs
- Aluminum foil
- Books- hardback (orange/black)
- Gold reflective surface (used in photography)
- Mirror with ring light
- Plain white sheet of paper

The sensor is set up by first connecting to power, then being propped up so it will remain stationary. Before testing, the object is moved to the very front of the face of the sensor to create a control value of zero then moved further 2cm at a time.

Sensor Results

The results of the sensor are displayed on figure 1.2. The data shows for objects that can be detected from farther distances output a lower voltage, and vice versa with the objects at a closer distance. The distances can be seen on the x-axis and the voltage values along the y-axis. The closer objects would prove to be an **obstruction in the vehicles path, therefore would send more power to the Bang-Bang or PID controller to decide what the next steps for the vehicle are.

Figure 1.2

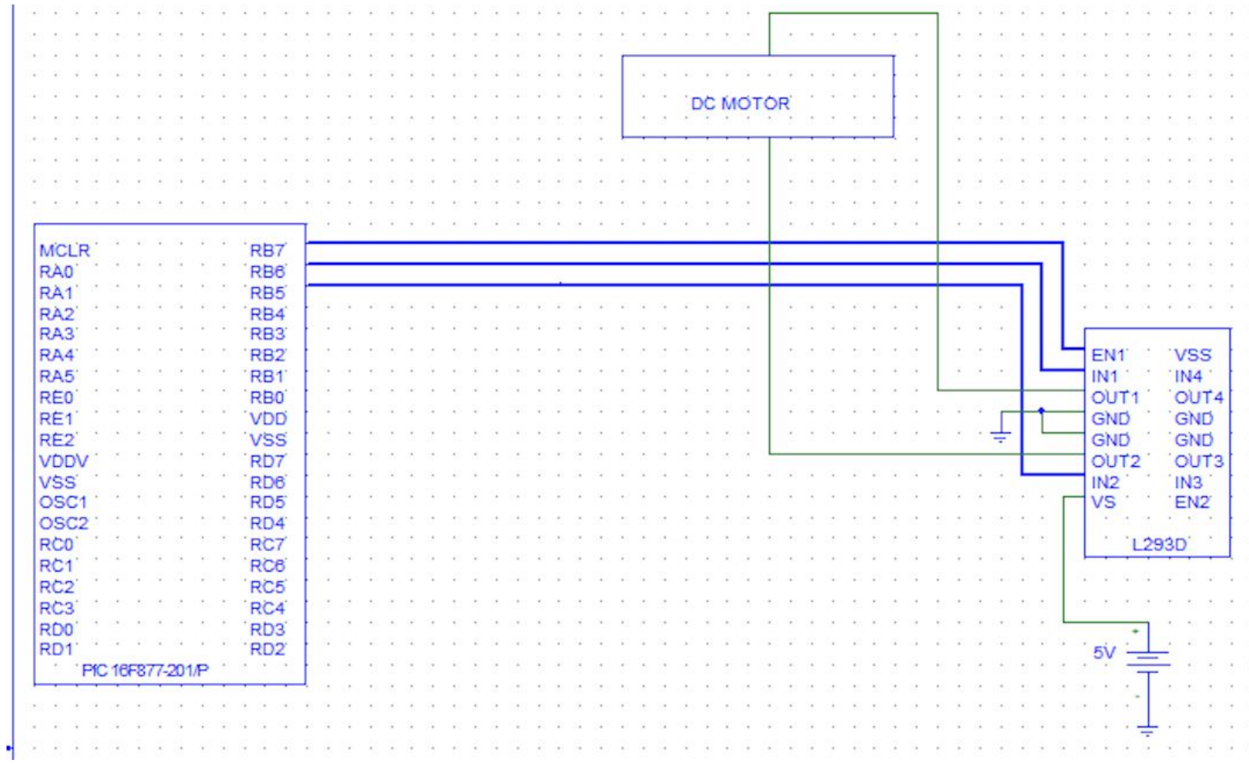
The results of the data vary depending on the object; meaning the size, reflection of the sensor, color and texture can affect the reading. For example, the glass of water had the lowest voltage reading when closest to the sensor, instead of producing the desired outcome of the hardcover books. The glass of water was also unable to give a reading after about 16cm. This is most likely due to the refraction of the water and glass, and the sensor's detection being lost in the reflection after the 16cm distance. Unlike the glass of water, the hardcover books showed a linear regression. As the books were taken further, the output voltage also decreased, simulating no visible obstructions to the vehicles path.

For most of the tested objects, the sensor is rated to have a distance from 4cm to about 30cm. Objects like the gold reflective surface start like other objects with a high voltage but this object's values fluctuate because of its reflection. At certain points the light reflects differently on the surface, which makes the sensor see the object at some distances and not at others which makes for a non-linear result. As the results above show once the object is at about 28cm the minimum voltage that can be outputted is less than or equal to 0.5V.

The sensor continuously detects objects that may come into view. It records distance measurements so that it can send an increase in voltage to the PID or Bang-Bang controller. The increase in voltage signals to the other sensors to detect if there are objects in the path, so the direction the vehicle is moving can be changed.

DC Motors

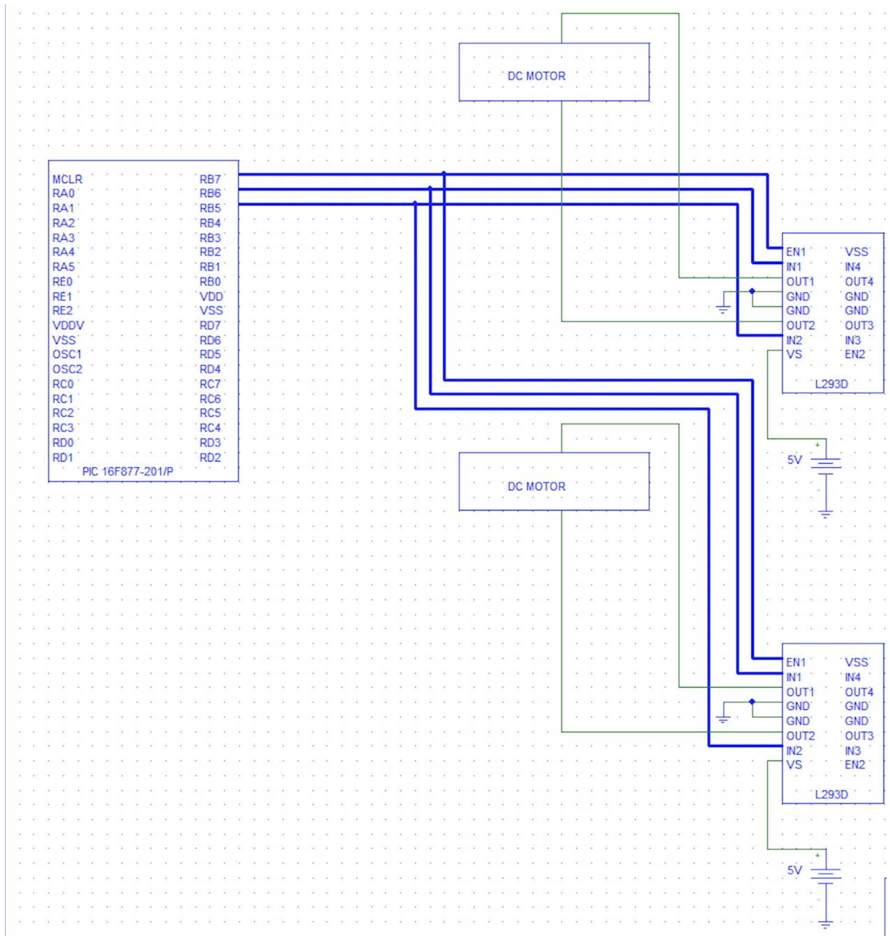
To control the DC motor, the L293D chip uses an enable(EN1) and two directional control logic inputs (IN1, IN2). The each of the three inputs will be provided by three bi-directional I/O ports (RB7, RB6, RB5) from the PIC 18F877-201/P. These I/0 ports are outputting voltages to the L293D chips which in turn drive the DC motors. The power supply for the L293D comes from a 5V battery pack.



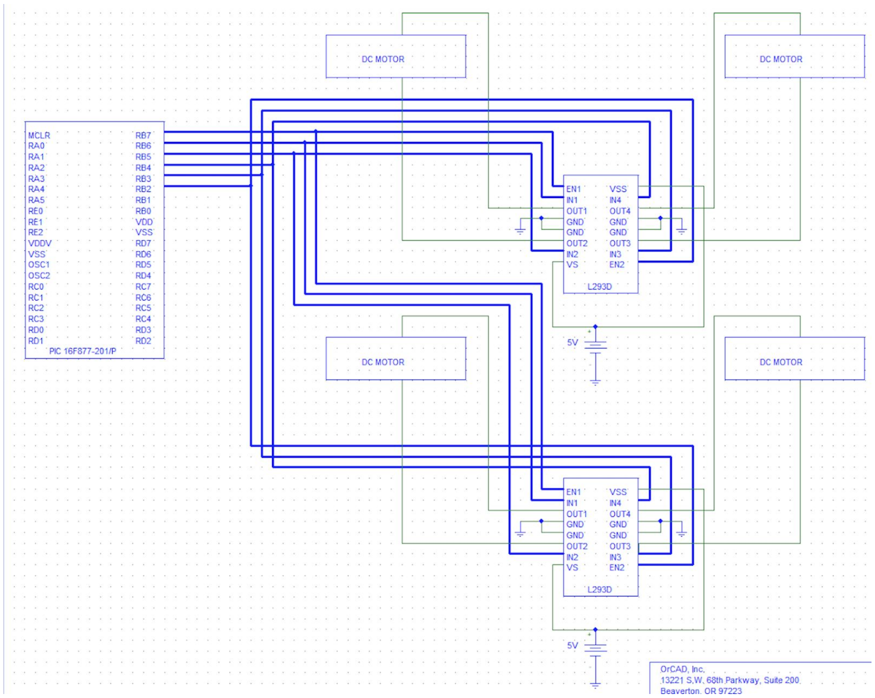
Using digital logic principles, there are eight ($2^3 = 8$) possible outcomes for the motor circuit. However, we are only concerned with three specific outcomes as they are related to our project. For the motor to run, the enable voltage must be at or exceeding the minimum enable high voltage (2.3 V) along with one of the input voltages exciting the minimum input high voltage (2.3V) and the other input voltage below the maximum low voltage (1.5V). We will use the truth table below to arrange our inputs so that they produce our desired outcomes of forward motion, reverse motion, and no motion or stop.

EN1	IN1	IN2	Functions
High	Low	High	Forward
High	High	Low	Reverse
Low	X	X	Stop

Using our previous schematic and truth table, we add another chip to the circuit to have both motors on the same line. With this schematic, our intention is to have both motors moving in sync by attaching the three inputs of the new chip to the same corresponding I/O ports as our original motor. For our final product, we want to have two wheel drive laterally.



Since we are connecting both motors so that they run on the same line, the truth table for their operation is the same as if it was just one motor connected to one single chip. Next, we will make the name design using the other side of both chips. On the other side, we will connect the motors to separate pins on the PIC. When we separate the inputs for the pairs of motors by different pins, we create a lateral two wheeled drive for our platform.



Now that we have a lateral two wheeled drive for our platform, we can toggle our inputs to make our platform turn left to right instead of just moving forwards and backwards. When trying to turn the platform, one side of motors must be turned off while the other side of motors continues to keep moving. The truth table below illustrates how to toggle the inputs so that you can make the car travel in the desired direction.

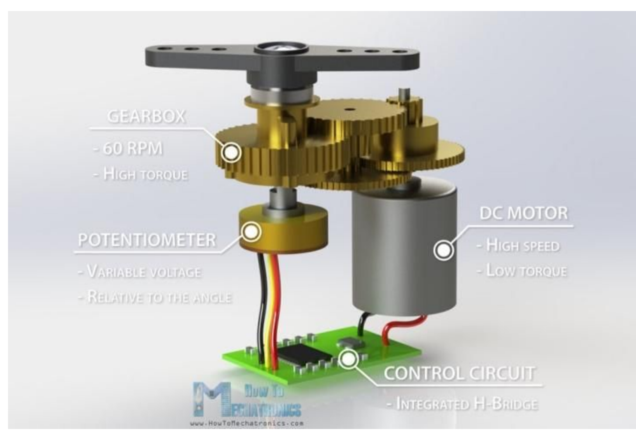
LS EN	LS IN1	LS IN2	RS EN	RS IN4	RS IN3	Function
H	L	H	H	L	H	Forward
H	H	L	H	H	L	Reverse
H	L	H	L	X	X	Turn Right
L	X	X	H	L	H	Turn Left
L	X	X	L	X	X	STOP

Now that we understand how the PIC is controlling the output of our motors from the hardware, we will look at the software that speaks directly to both chips. Below is some pseudo code that accesses the pins connected to the L293D chip and send high voltages so that we can bring our truth table to life. In this code, we are moving an 8-bit binary literal into an I/O port. The binary literal correlates to the pins of the port that move onto the L293D Chip. The literal can also be decoded by using the truth table to match which pins are going to be set to high/low or in our software case, 0 or 1.

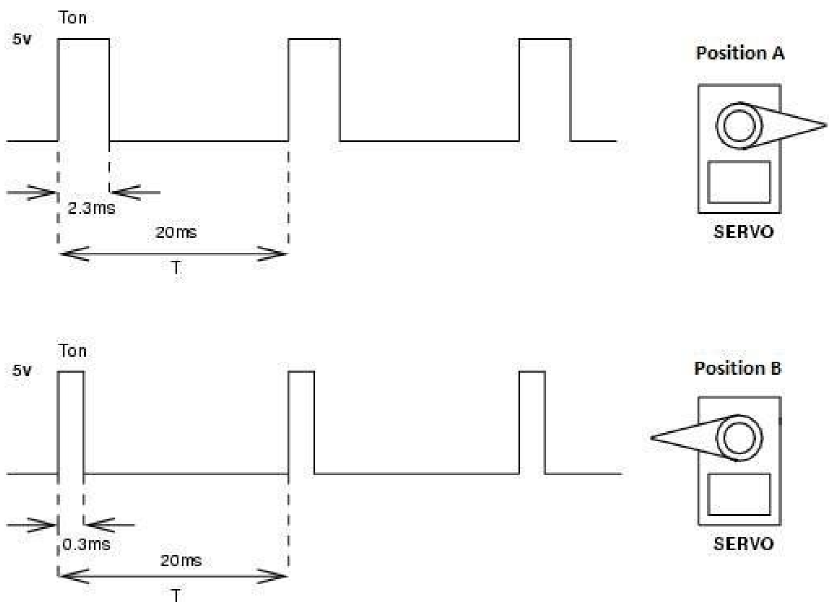
```
go_fwd      movlw       B'01010101'
            movwf       PORTD
            return

go_bwd      movlw       B'00000000'
            movwf       PORTD
            return

r_turn      movlw       B'01100110'
            movwf       PORTD
            return

l_turn      movlw       B'10011001'
            movwf       PORTD
            return

stop        movlw       B'10101010'
            movwf       PORTD
            return
```

Servo Motors

The Servomotor is a rotary motor used for precise angular movements and velocities. It is commonly used in autonomous steering and robotic operations with an emphasis on precision. It consists of a DC or AC motor which is connected to an internal potentiometer and position feedback sensor. The position of the internal potentiometer is constantly compared to the position of the controller. Any difference will produce an error signal and the motor will move in the desired direction (forwards or backwards) to correct the error. When the motor has reached the desired position,



the error signal reduces to zero and the servo will stop moving. In our case, we used a potentiometer to act as our commanding controller.



The Servo motor works through Pulse Wave Modulation. It is controlled by sending a series of wave pulses at a frequency of approximately 50 Hz. While the frequency of the waves are constant, the width of the pulses can be altered to change how far the motor will move.

Conclusion

When the semester ended, we completed the circuits for the PIC chip, DC motors, and Servo Motors. We also completed all the testing for the IR Sensor and LIDAR Sensor. For the autonomous cars to be complete, we need to integrate the Assembly code onto the PIC 8 bit and PIC 32 bit along with the PID algorithm . After the code is completed, the last step is full system integration with all components. Although we did not meet our goal of finishing both of our cars, we obtained a lot of data and recorded enough information for the incoming graduate students to finish the cars in the summer so that they will be ready by the fall, The goal is for the incoming graduate students to teach the next senior design team how to do this work as well along with getting into the development of the FPGA based autonomous car.