ENEL 387 Project Final Report

Due: April 8th

By: Ammar Alvi (200374049), Shannon D’souza (200378926)

Table of Contents

1.0 The Design Process:....3
1.1 New Objectives and Course Design....3
1.2 Hardware Design....4
1.2.1 Requirements/Specification....4
1.2.2 Component Selection....4
1.2.3 High-Level Block Diagram....5
1.2.4 Detailed Design....6
1.6 Robot Navigation Process....7
1.7 Verification and Testing....8
2.0 Deviation From Functional Specification:....9
2.1 Unused Sensors....9
2.2 Logic Changes....9
2.3 Power Supply Change....9

1.0 The Design Process:

1.1 New Objectives and Course Design

The couch surfer robot is designed to navigate around a 1.7m by 0.7m couch starting from a corner on a white surface. Along its route, it should interact with the boxes that are flushed against the couch which come in various sizes. The minimum width of the boxes is 15cm width and 25cm length to account for the size of robot chassis. The couch surfer should autonomously be able to go around the object and the count the number of white strips placed adjacent to the boxes. The number of stripes counted correspond to the box number. The couch surfer should successfully be able to navigate around the couch and stop at the same white surface it started from. This should be all sensor based, the robot cannot be pre-programmed to meet these objectives. See Figure 1 below for a top view of the course.

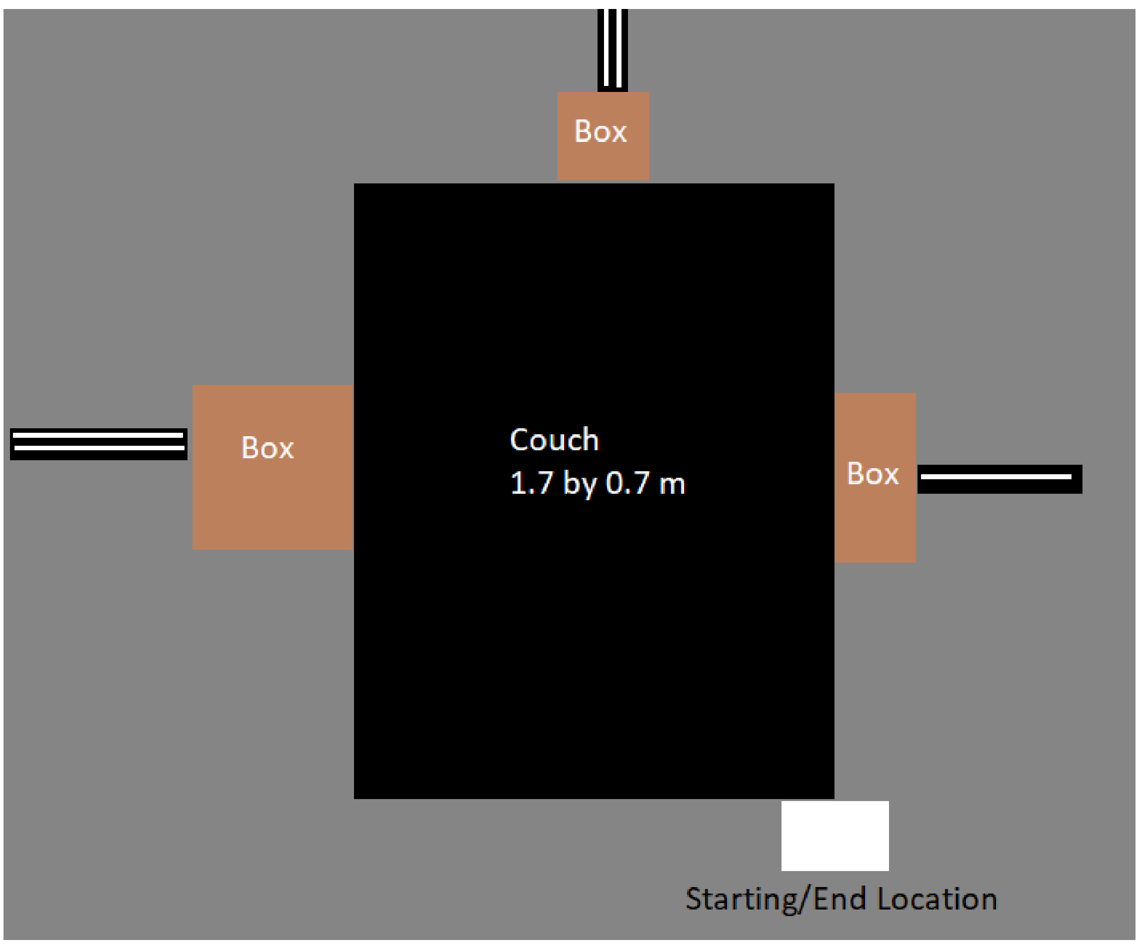

Figure 1: Top view of the navigation course

1.2 Hardware Design

1.2.1 Requirements/Specification

As part of the design process, we first needed to determine what requirements the robot was subjected to. The following initial requirements for the Robot were laid out before the COVID 19 pandemic:

- must accept dynamic inputs from the outside world.
- must provide controlled outputs to the outside world.
- serve some real world function or solve a real world problem.
- must incorporate at least two sensors in addition to those provided on the ENEL 384 board.
- must include both digital and analog sensors.
- must incorporate at least two output devices in addition to those provided on the ENEL 384 board.
- must have at least two different output interface systems (PWM, SPI, UART, GPIO to external drivers).
- must have at least one output device with a current requirement greater than the STM32F100 rated output current.

Due to the change in the project objectives from the COVID 19 Pandemic, we no longer needed to use the analog temperature sensor which was ordered for the previous design to meet the analog sensor requirement. Therefore for this project, we only used digital sensors to accomplish our new objectives.

1.2.2 Component Selection

The perseids Robot Car 4WD Chassis Kit from amazon.ca was used. It included 4 DC Gear motors (TT Motors) and 4xAA battery box for motor power supply which gives us around 6V. The current consumption of those motors is around 200mA at 0.15Nm ~0.60Nm torque.

To drive these motors, we used a two H-Bridge motor driver DRV8833 with 1.2A output per channel. It runs at 2.7 to 10.8 V logic/motor power so therefore it would be able to accept 3.3V PWM signals from the STM32 board.

To sense the distance from the robot to the couch/boxes for navigation, we decided to use three HC-SR04 Ultrasonic sensors. On its trigger input, it accepts a 10us pulse width which sends 8 cycle sonic bursts from the module. On its echo pin, it outputs a pulse width that is proportional to the distance measured (uS / 58 = centimeters).

In order to uniquely sense white surfaces, we determined an infrared sensor would be a suitable choice. We chose the TCRT5000 infrared sensor module that works at very small distances. It constantly transmitts infrared light and the strength of the reflected light measured determines if the object was black or white. The sensor can be calibrated at certain heights using potentiometer and comparator on the module to provide a digital logic 1 if it detects a dark surface and a logic of 0 if it detected a white surface.

To display the information gathered by the robot, we decided to use a Bluetooth serial monitor. An HC-05 module was selected to wireless transmit our Bluetooth data. The HC-05 has a UART serial interface that can be used to connect to the STM32 board.

The table below shows the components, quantity and cost of the project.

Component Name:	Quantity:	Cost: ($)
DRV8833 Dual H-Bridge Motor Driver	x1	3.49
HC-SR04 Ultrasonic Sensor	x3	5.59
TCRT5000 IR Line Sensor	x1	1.99
HC-05 Bluetooth Module	x1	13.93
Perseids Robot Car 4WD with Motor	x1	22.99
6 x AA Battery holder	X1	4.19
8 inch Premium Female/Male Jumper Extension 40pc	X1	5.95
Ultrasonic Sensor Acrylic Mounting Bracket	X3	4.17
Total:		62.30

1.2.3 High-Level Block Diagram

A high level block diagram for the connection of the system is shown below in figure 2.

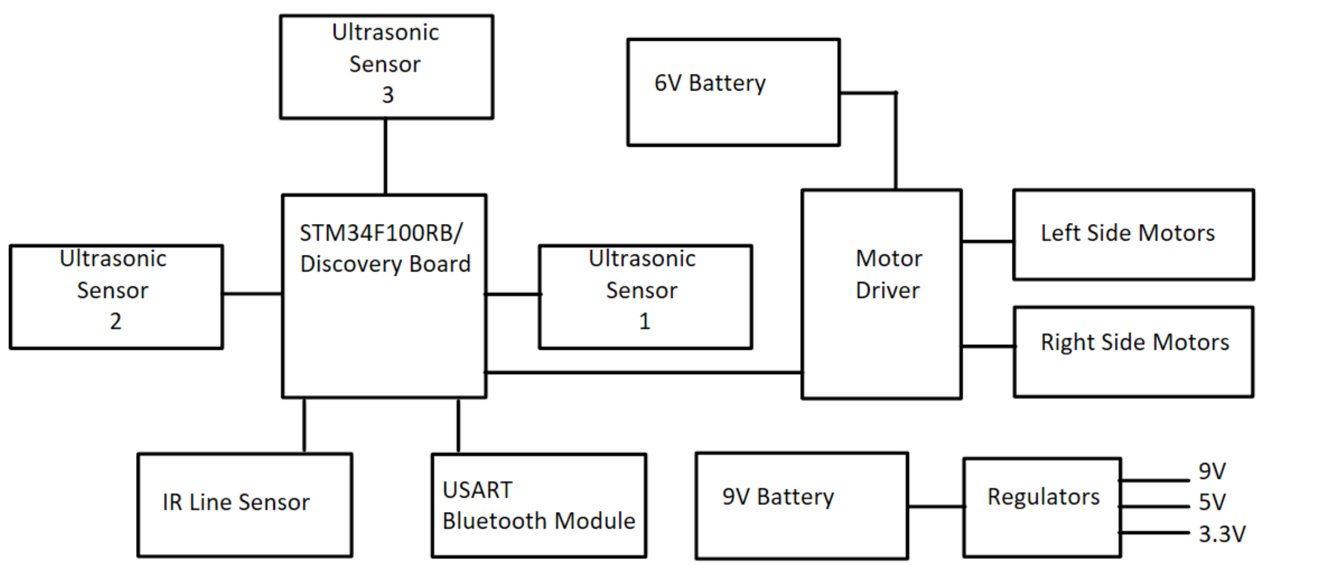

Figure 2: System Block Diagram

1.2.4 Detailed Design

Port selection was an important step in our design as certain ports on the STM32VDL were only available for our sensor inputs. Setting ports PA8-11 as alternate functions enabled us to use TIM1 for sending PWM output signals to our H-Bridge Motor Driver. Our Bluetooth module was connected to PB10 since it's alternate function is USART3. Since our Bluetooth module only needed to transmit data to another device, we only used the TX pin.

A lot of the GPIO pins on our discovery board would only tolerate up to 3.3V inputs and since most of our external components operated in 5V, we looked to the STM32 reference manual to determine which 5V pins were available. A lot of the GPIO pins were also connected externally to the ENEL 384 PCB, and so after carefully considering all of these constraints, we determined which GPIO pins were best for use.

Below is the schematic representation of all the ports used for our system.

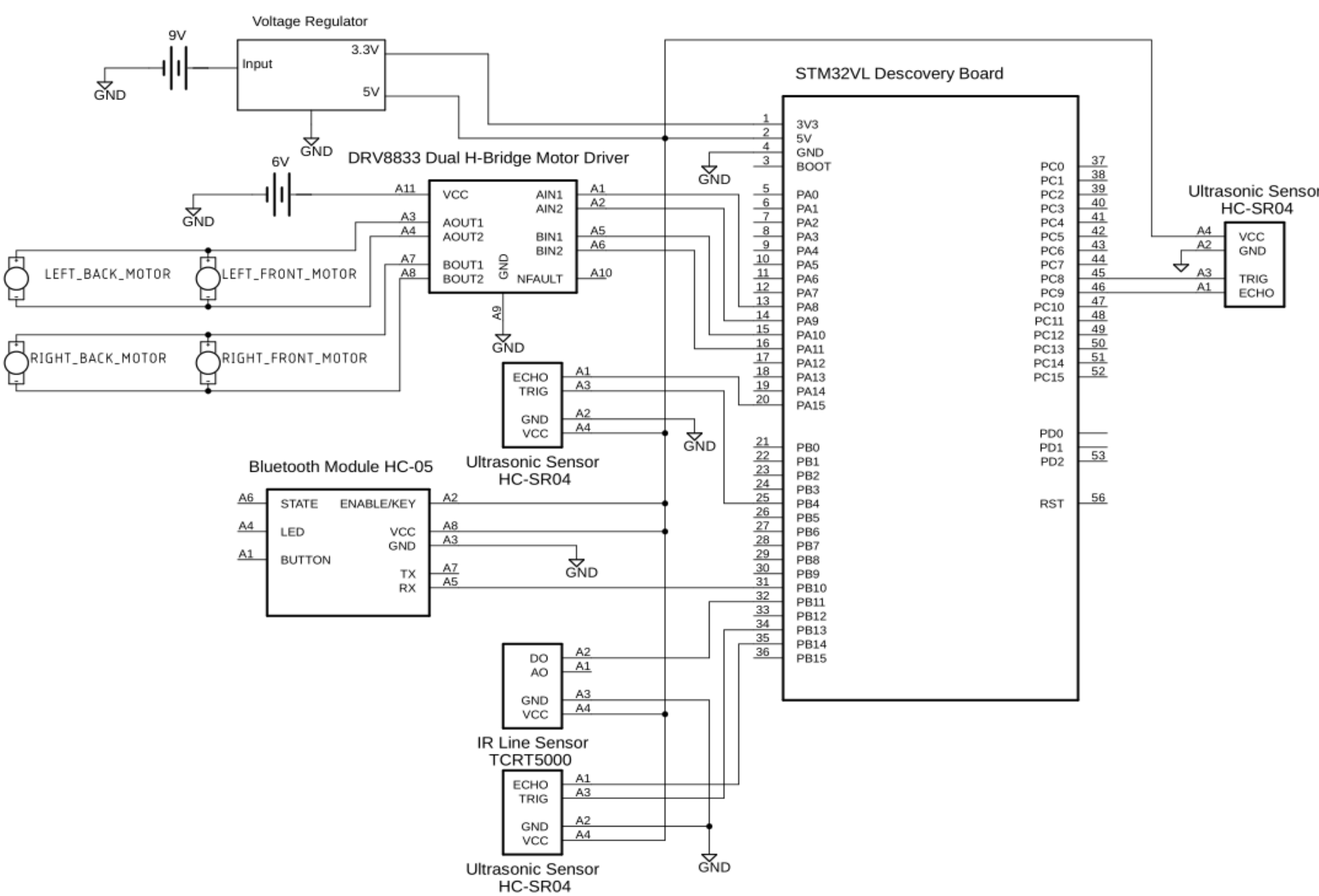

Figure 3: Schematic Diagram of System

1.6 Robot Navigation Process

In order to conceptualize the process on how the robot would navigate the course and how our main code would flow, we came up with a flow chart in the form of a pseudo state diagram, see figure 1.3 below. It gave us a logical understanding on how to go about implementing the main application code. The flow of the main function in our application somewhat flows like the chart where each of the nodes represent a while loop that runs until a certain condition has been met. The conditions are sensed inputs that are represented by the flow line.

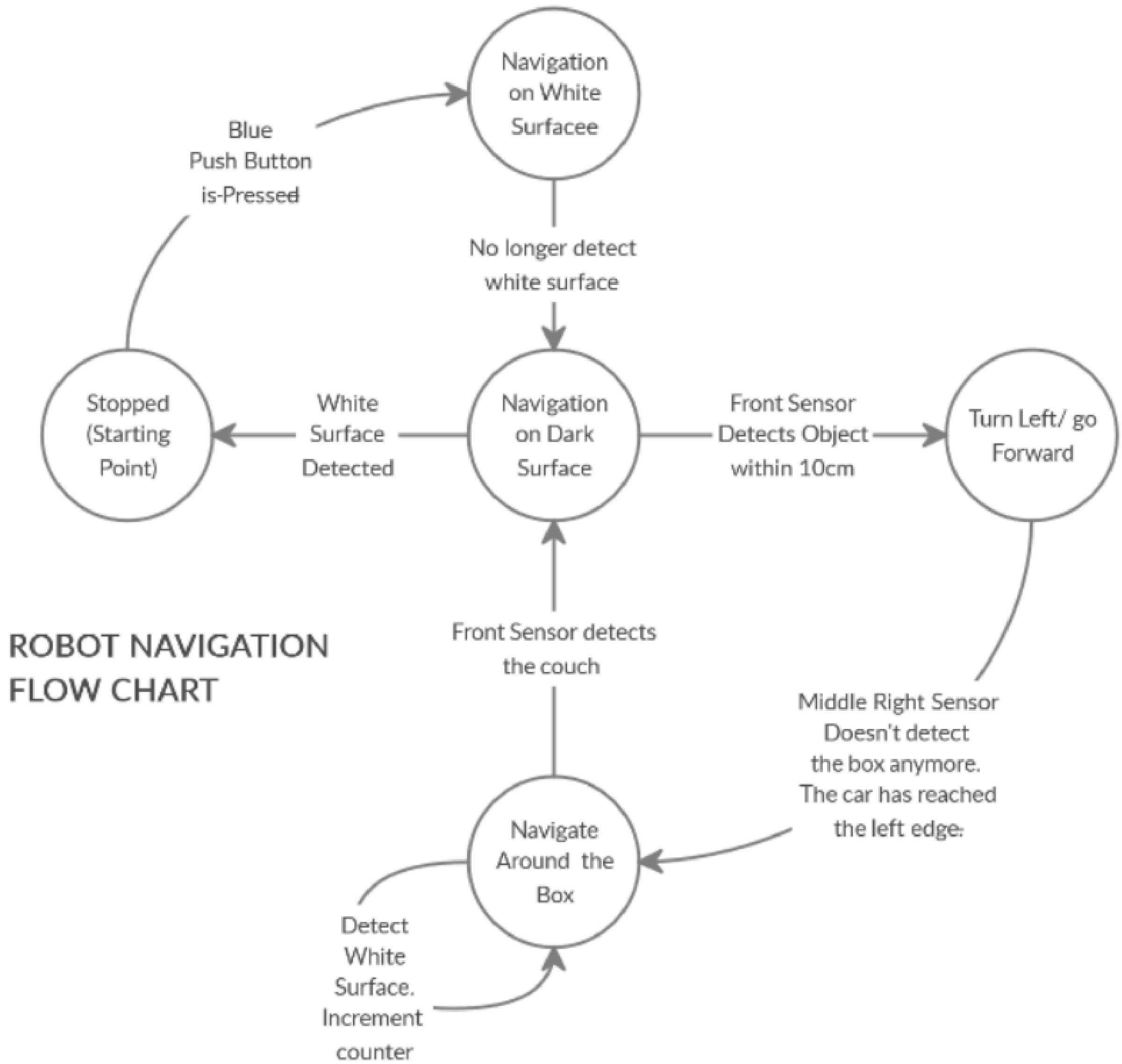

Figure 4: Robot Navigation Flow Chart

1.7 Verification and Testing

The testing for our system was performed using a bottom up approach where the lowest level components were tested first. The low level units were combined to form software functions which were used to test the entire system.

We individually tested the ultrasonic distance sensors by comparing the actual distance to the measured distance to ensure accuracy. The bluetooth module was tested by transmitting combination of characters that formed a string. We tested the white stripe counter using the infrared sensor at changing forward speed of the car and at different angles.

After making adjustments at the component level, we combined individual units and started testing them as a group. For example from figure 4, we individually tested every processes individually in a controlled environment such that it would work every time. One of the main problems we faced in this step was finding a dark enough surface so that the infrared sensor would not detect the color white. We also needed a smooth surface so that the wheels would be able to slide when making turns. This portion of our testing was the most time consuming because we had to make several adjustments, tweak a lot of our distance thresholds, and add extra delays when making turns to prevent the car from getting stuck. We used the LCD display to constantly display the measured distance from every sensor which would help us tweak the threshold values in our code. At every step in our code, we outputted a unique string to the LCD to indicate when our code fails.

Finally we were ready to combine the code for all the different states from figure 4 and test our entire system. Using a similar scheme as above, we tested our robot on the course noting all the points of failure and ways to circumvent those issues. Majority of our problems was when the robot was making turns. Since we did not have a way of making accurate 90 degree turns, we relied mostly on our distance sensor inputs. If a specific ultrasonic distance measurement was inaccurate in our loop, it would completely disrupt the flow of our p`rogram. So in this part of our testing, we implemented additional logic requirement and extra steps as a workaround. When we started seeing consistent results, we decided to start recording. See the link below for our result.

https://www.youtube.com/watch?v=_aDb7OR65XA

2.0 Deviation From Functional Specification:

Due to COVID-19 there was deviations from the original functional specification document. We also had other changes to the project that we not COVID related; all of which will be discussed in this section.

2.1 Unused Sensors

Our original design included 2 additional infrared sensors that were to be used to detect and follow navigational lines on navigation course. Our original design was to achieve navigation through the use of these lines, however due to the change in course we opted to make our navigation ultrasonic sensor based.

We also neglected to use 1 more infrared sensor that would have been used to detect wall targets as we opted to not use wall patterns in our course design.

Similarly the temperature sensor and infrared receiver were also not included in our present design due to the change in course targets excluding a light source.

Our initial sketch and block diagram of the system was also changed with the previous statements in mind. The placement of the remaining components changed slightly to accommodate the new course

2.2 Logic Changes

Our initial design included logic of how we detected hallways and rooms and also recorded what room was entered. We combined the logic in our final design to instead use walls to navigate, detecting when there is an open space or obstacle to turn or avoid. Also using only one infrared sensor to detect pointed down to count the amount of lines the car crosses indicating what obstacle number the car has avoided.

2.3 Power Supply Change

Our initial design used a 6 1.5V batteries in series to power our micro-controller and sensors. What we found was that the additional weight from the battery pack was enough to prevent the robot from moving. We then decided to change into a lighter 9V battery. This alleviated the weight issues however power consumption issues then became prevalent as the battery voltage dropped significantly during operation. We then had to change the power supply to a wall connected outlet to correct the issues found.