

The Caddiellac

"In the pursuit of perfection on the course, your clubs should follow your lead"

EE41430

Senior Design Final Report

Ryan Bernhold

Matt DePorre

Rachel Drumm

Colleen Tigani

Table of Contents

1. Introduction.....	2
2. System Requirements.....	3
3. Detailed Project Description	
3.1 System Theory of Operation.....	4
3.2. System Block Diagram.....	5
3.3. Golfer Detection (Vision System)	6
3.4 Object Detection (Ultrasonic)	12
3.5. Manual Override (Bluetooth LE)	16
3.6. Self-Driving (Motor Control)	17
3.7 Interfaces.....	22
4. System Integration Testing.....	23
5. To-Market Design Changes.....	26
6. Conclusions.....	26
7. Appendices	
A.1 Hardware Schematics and Component Data Sheets.....	Appendix A
A.2. User Manual.....	Appendix B
A.3.Final Code.....	Appendix C

1. Introduction

This report outlines the design, testing, and construction of Team Caddiellac's final Senior Design project, which is a cumulation of the Electrical Engineering undergraduate experience and allows the team to utilize the knowledge and skills acquired the past four years at the University of Notre Dame. The group consists of Ryan Bernhold, Matthew Deporre, Rachel Drumm, and Colleen Tigani.

The Caddiellac is a motorized cart that follows behind a golfer on the golf course and carries their clubs. It is hands free to allow the golfer a relaxed and enjoyable golfing experience. It relies on a color code which is worn by the golfer to detect him and follow closely behind, and can be overridden and driven by a mobile application for all users with an Android phone. This report outlines the problem statement and proposed solution, system requirements, a system block diagram, subsystem breakdown and discussion, and system integration. It also includes a user manual and all code used in the product.

When golfing, there are two major issues that one has to deal with. The golfer has to have access to a large selection of heavy golf clubs at any given time on the course. Additionally, golf courses are expansive and transporting these clubs is tiresome. Currently, there are three ways that this problem can be resolved: the clubs can either be carried around by hand, the golfer can rent a golf cart or a pull cart, or hire a caddie for the day. To carry the clubs by hand is cumbersome and tiring for the golfer. The latter two options become expensive for regular golfers. There is a need for a method of transporting clubs which relieves the golfer from carrying them from stroke to stroke, and also eliminates the necessity of renting an expensive piece of equipment or hiring someone to walk with you.

The solution is to build a motorized cart that follows behind the golfer based on position tracking. The cart resembles a typical pull cart, and drives behind the golfer along the course. The cart has a visual detection system which follows the golfer by identifying a color code worn on his back. The cart has an accompanying mobile application which communicates via Bluetooth for Android users. This allows the user to turn the follow capability on and off, as well as manually override the movement of the cart. This also allows the golfer to take a swing, or use the restroom without his golf bag getting in the way. Additionally, if the cart gets in a tight spot, the user can control it back on track. The robotic cart accommodates and transports the golfer's personal golf bag in a secure manner.

The final prototype of the Caddiellac met all of the design requirements. It was self-driving and successfully ran on two motors. It incorporated golfer detection and following through the vision system, specifically the PixyCam and color code combination. It allowed for manual override through the bluetooth app. And lastly, it had a user friendly design in terms of function on the golf course. The mechanical design could definitely be improved in term of ease of transportation on and off the golf course, however it provided access to the clubs and the overall function of the Caddiellac does not interfere with one's game.

2. System Requirements

The system requirements for this project can be broken down according to the features The Caddiellac is expected to demonstrate. The Caddiellac must be self-driving, incorporate golfer detection and following, allow for manual override, and have a user friendly design. If all requirements are successfully met, The Caddiellac will solve the problem of transporting golf clubs along the course in a reasonable, simple, and inexpensive manner. All numbers used in the system requirements were taken from the calculations of the sub-system demo.

First and foremost, The Caddiellac must have the ability to be self-driving. This includes the capability to carry the weight of a bag of golf clubs and maneuver along the golf course. The Caddiellac must be able to maneuver along the terrain of the golf course, including up and down inclines, without tipping. The design must feature a low center of gravity to avoid tipping, along with the appropriate torque being provided by the motors to drive up inclines. The coefficient of friction on the wheels must also be great enough to resist slipping on surfaces. The Caddiellac must also have enough power to maintain itself through a full 18 hole game of golf.

The Caddiellac will drive at the average pace of a golfer, 3 mph. It will be expected to drive up inclines of up to 15 degrees. This will require a torque of 2.399 Nm per wheel. In order to ensure that the Caddiellac does not slip on inclines, the coefficient of friction of the wheels must be great enough to overcome sliding. The Caddiellac will be expected to last the entire duration of a golf game, 5 miles, which will require a battery of 19 Amp-hours.

The Caddiellac must also have the ability to follow the user. It must be capable of detecting the position of the user, and then adjusting its path accordingly in order to follow the user along the course. Additionally, the Caddiellac should have the ability to detect obstacles in the vicinity, including nearby trees, rocks, or golfers, and avoid collision with these obstacles.

The system must have the ability for a manual override. This will require a user interface in which the golfer has the ability to turn the following feature of the caddie on and off to stop the Caddiellac when desired. It will also require the ability to drive the cart forwards and turn it left and right, in order to avoid any situations in which the cart may get stuck.

The Caddiellac must have a user friendly design to facilitate ease of use for the golfer. The physical design must be sleek and clean, and hold the golf bag at an angle that allows for easy access of the clubs. The caddie must also be easily transported when desired, which imposes requirements on its height and weight.

In order to maintain transportability, The Caddiellac should be easily stored into the trunk of a car, which provides size limitations of 36" x 34" x 23". The weight of the caddie should allow for small amounts of lifting for the sake of storage. When the golf bag has been removed from the caddie, the maximum weight of the caddie will be restricted to 45 lbs. Additional wheels will also be provided for ease of transportation, so that the Caddiellac can be tipped back and wheeled from the car to the course.

3. Detailed Project Description

3.1. System Theory of Operation

The Caddiellac operates through the use of five key components: mechanical design, drive system, golfer following, object detection, and manual override. The Caddiellac is comprised of a sleek, three wheel cart design. The drive system has two drive wheels and a caster wheel for easy rotation. The drive wheels are powered by motor controllers which rely on a PWM signal. The motor controllers adapt to the desired speed by changing the duty cycle being sent to the motors.

The system itself is controlled by a board which contains a PIC32MX795F512H microcontroller. For golfer following, the golfer is detected through the use of a PixyCam, which identifies a specific color code worn by the golfer. The camera identifies the golfer's color code, determines the location of the color code, and then adjusts the motor speeds to follow the golfer along the course. If the color code disappears, The Caddiellac will stop. This camera also allows for object detection. If an object comes between the golfer and The Caddiellac, the color code will be lost and the system will come to a stop. Also, The Caddiellac determines the distance between itself and the golfer to ensure that it does not come too close, and will stop if the golfer is less than the proper following distance away. Manual override is provided through an Android application, which communicates with The Caddiellac through Bluetooth. When manual override is enabled on the application, The Caddiellac stops and awaits further instruction from the golfer. The golfer then has the ability to drive The Caddiellac straight, as well as turn it left and right. While the system is being controlled by the application, it will not respond to the presence of a color code to ensure that the application always overrides other functions.

3.2. System Block Diagram

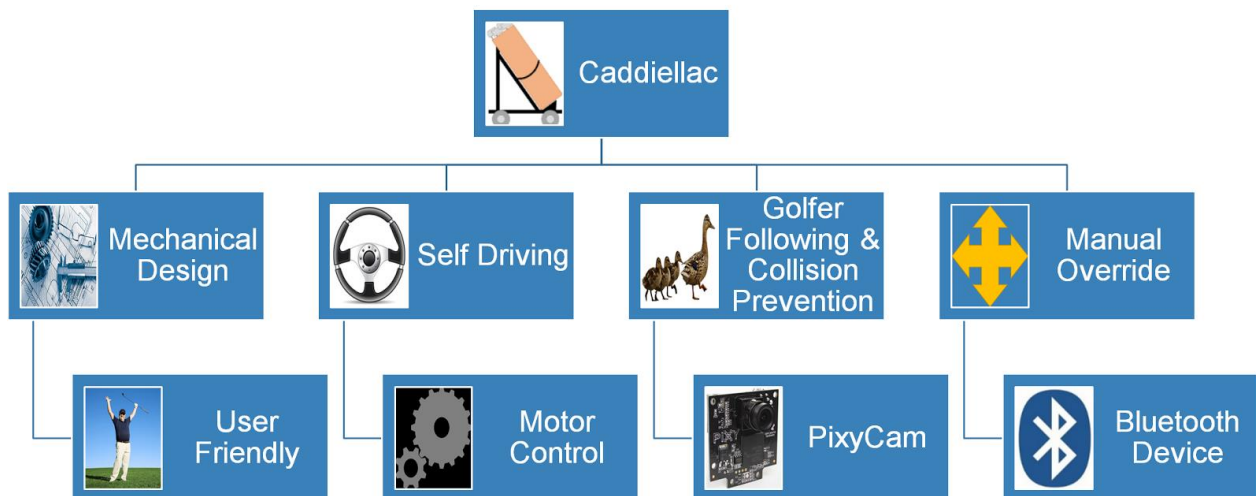


Figure 3.2.1. Caddiellac System Block Diagram

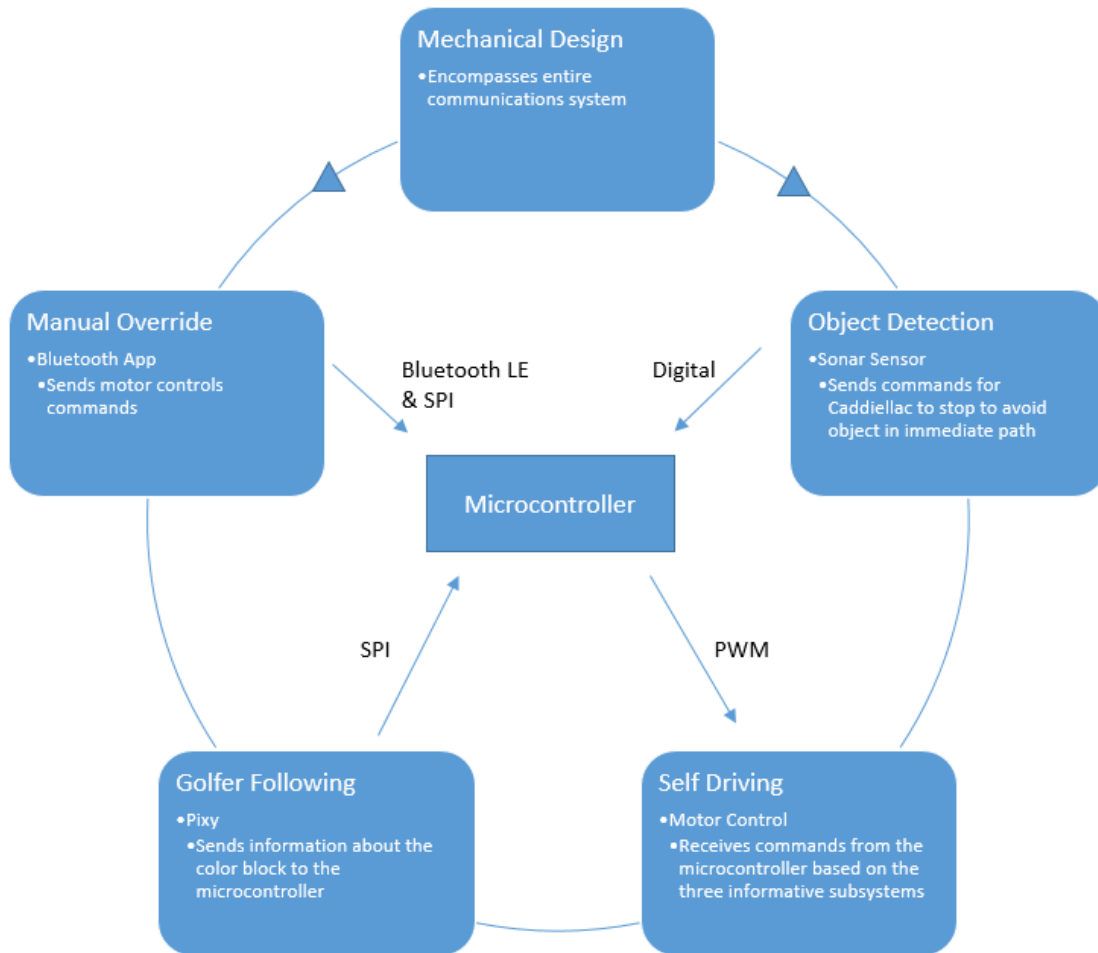


Figure 3.2.2. Caddiellac Subsystem Communication Diagram

3.3. Golfer Detection (Vision System)

The Caddiellac has the capability to follow the user along the golf course through a visual detection system. The vision system detects the location of the golfer through use of a color code that is worn by the user. The color code is detected as a block of pixels by the Pixy, and the x and y coordinates of this block are then sent to the microcontroller. The microcontroller uses this information to direct the motion of the Caddiellac towards the path of the golfer.

Pixy over Raspberry Pi

Two options seemed feasible to implement the golfer detection subsystem when design first began: the Raspberry Pi and the Pixy. While the Raspberry Pi has the capabilities of capturing and communicating image data, the Pixy was ultimately chosen due to its more advanced and developed features. The Pixy can be programmed to recognize color channels with the click of the button and with the additional option detecting of color codes, two block of colors next to each other, the Pixy has the ability to pinpoint a specific object if the color code is visible. These additional tracking features preprogrammed in the Pixy made it a better option for the Caddiellac's vision system. With the smart features of the camera, focus of this subsystem shifted from developing a tracking algorithm to developing communication with the Pixy.

Method of communication- SPI

The SPI method of communication was selected for many reasons. Primarily, SPI was one of the best options for transmitting data on the Pixy. Although the Pixy is compatible with various communication interfaces, including I2C and UART, SPI is the fastest option in terms of utilizing the full capture speed of the Pixy. In addition, SPI was chosen since it will be used for other subsystems, allowing for the possibility of slave select control. Since the Pixy outputs a string of information describing the parameters of the block of pixels detected, previously written SPI send functions and read codes could be used as a model for the communication between the Pixy and microcontroller.

Process of testing, from Arduino to Kit Board

Since the Pixy is preset to interface with an Arduino, the first approach to testing was to run the given helloworld code and use the USBee analyzer to determine the specific signals associated with the Pixy. The helloworld code captured and displayed several parameters of the detected color block outlined in the screenshot of the serial port below. Note that the Pixy is detected a single block of orange for these tests. The Pixy was previously set to recognize orange as signal 2.

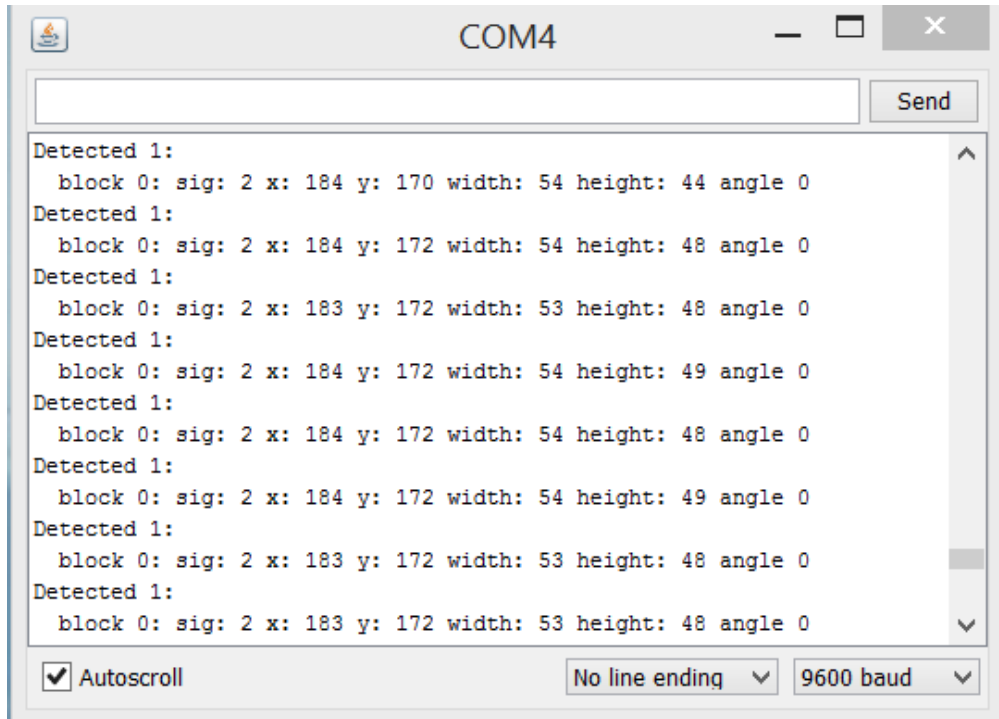


Figure 3.3.1. Serial Port of Helloworld Arduino Code

Then, hooking up the USBee analyzer, the MISO and MOSI signals can be seen to determine what the Arduino was specifically sending in the helloworld code.

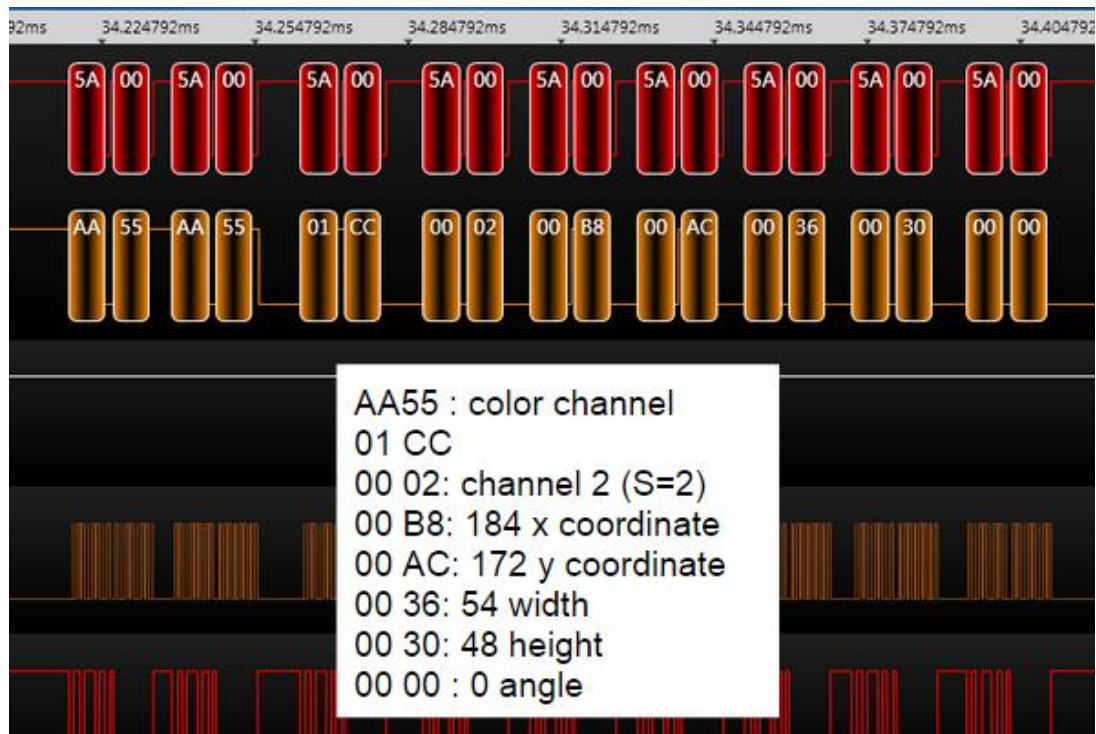


Figure 3.3.2. USBee Analyzer with Arduino

In the image above, MOSI is red, MISO is orange, and the clock signal is brown. The Arduino sends a continuous 5A 00 command and every 20ms, the Pixy sends a series of information about the color block. Since this information is sent about four times during the USBee capture times, testing consisted of several trial and error runs until this segment of information was found. If one examines a different part of the signals, the MISO from the Pixy simply sends zeros. Once the full captured data is taken into account, the segments of important MISO data can be seen.

The next step was to determine that the Pixy could communicate directly with the kit board, and that the desired data could be pulled from the MISO pin to be used for navigation. In order to test this, the Pixy was hooked up to the kit board by directly wiring pins from the camera to the SPI connection. Five connections were made: MOSI, MISO, chip select, clock, and ground.

The purpose of the test code is to pull desired data from the Pixy and display it on an LCD connected to the kit board. The full text can be found in Section 1 of the Appendix. In this code, both the SPI and LCD are first initialized. Then the write enable instruction is sent in order to allow communication with the Pixy, and the chip select is pulled high and back low. The write instruction is sent next to inform the Pixy that information will be written to it. After this, the code enters a conditional loop in which "0x5A 0x00" is sent. This command informs the Pixy that the kit board is searching for information on the color block. The Pixy will then begin sending information about the detected block of color approximately every 20 milliseconds. The first information received by the kit board is "0xAA 0x55," which informs the board that it is receiving a color code. At this point, the code breaks out of the loop and begins storing the information. It confirms that the next information received is also "0xAA 0x55." This ensures that the information is being collected and stored in the appropriate unsigned char locations. Because the color code is sent twice, this verifies that the second color code is not being caught in place of the first one.

The data collected includes the channel number, x coordinate of the color block, y coordinate of the color block, width, height, and angle. All information is then available for use. In order to confirm this, select information is sent to the LCD display. The information displayed is the color code, channel number, and x and y coordinates. For all testing performed, the color block being detected was on channel 1. Therefore, the outputs for color code and channel number were always "AA 55" and "00 01" respectively. As the color block

being detected was moved, changes in the x and y coordinates being displayed on the LCD were observed. When the block remained stationary, the only change observed was wavering by a single pixel, due to slight variance in the shape of the box detected. A screenshot of the PixyMon program, the Pixy's provided user interface, in Figure 2.3 shows the box detected during testing. The test setup can be observed in Figure 2.4, and the output display is shown in Figure 2.5.

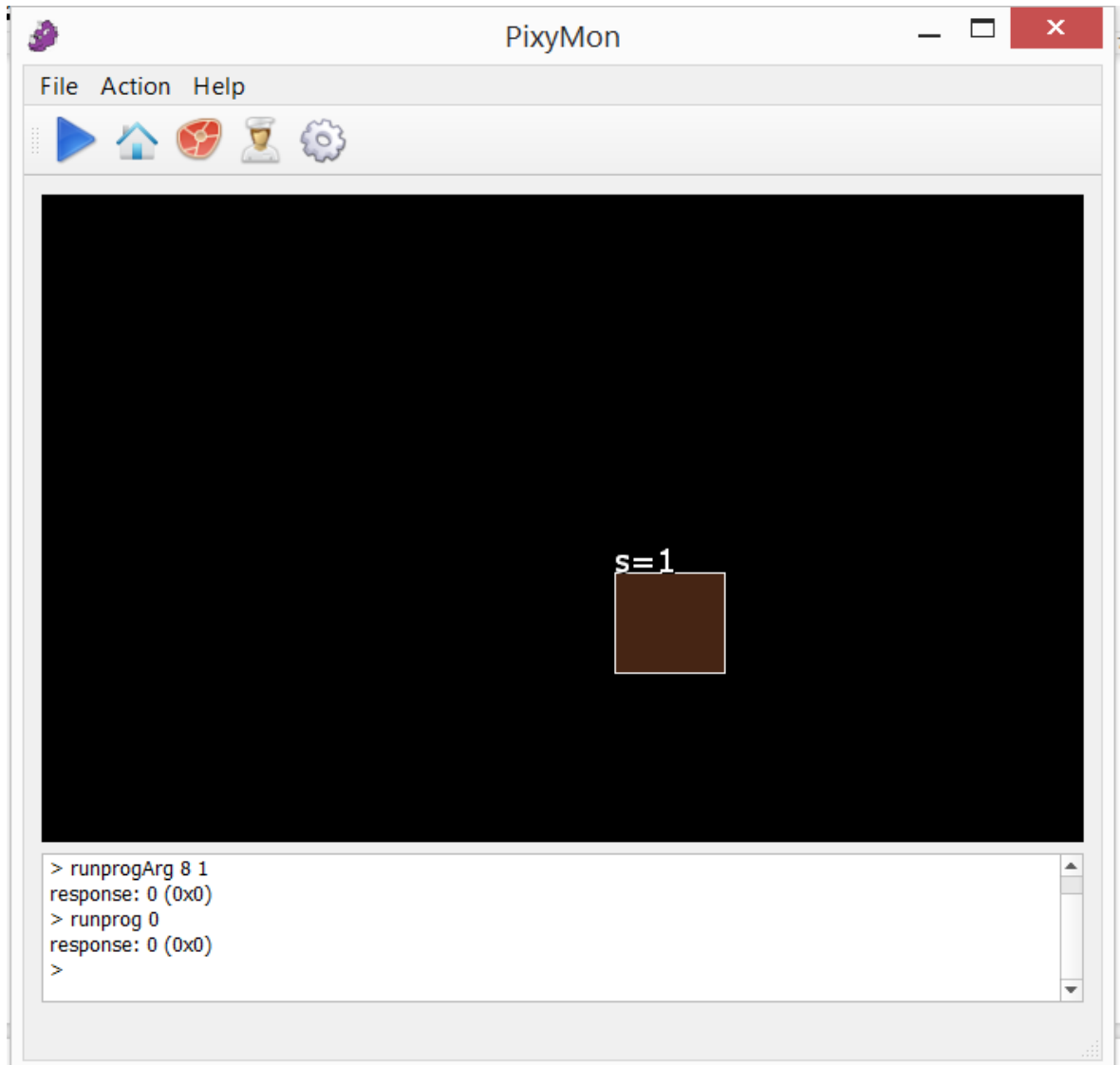


Figure 3.3.3. PixyMon Interface Detecting a Color Block

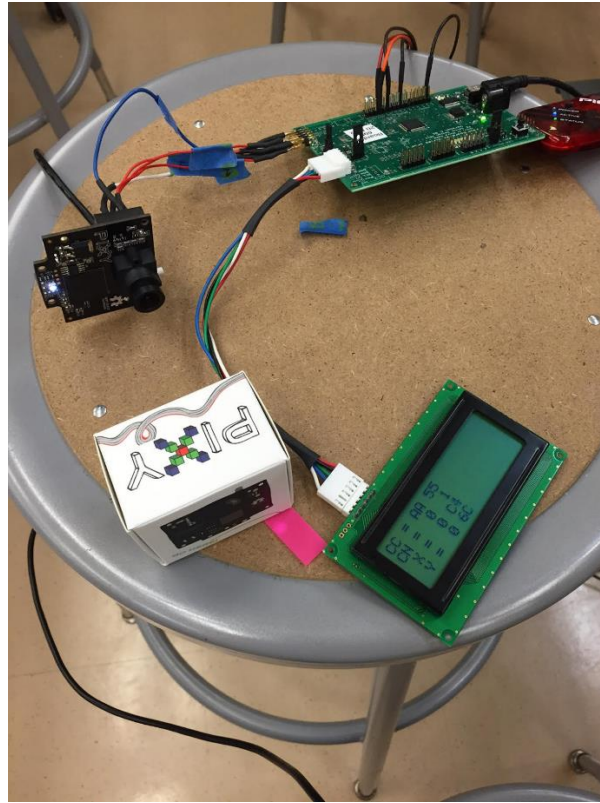


Figure 3.3.4. Test setup for vision system

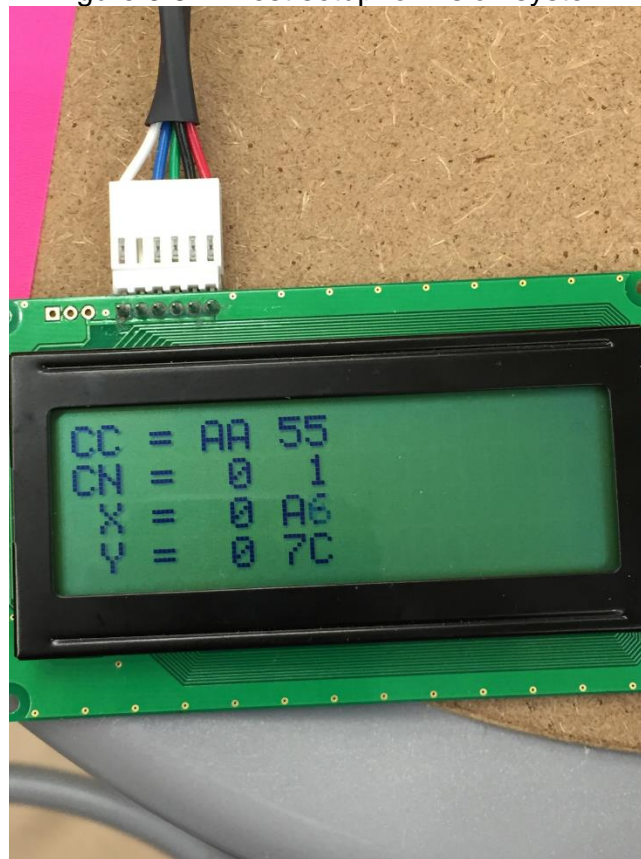


Figure 3.3.5. Output to LED display

Subsystem Updates

For the final design, the Pixy captures a color code rather than a singular color signature to improve detection and prevent false color detection. The code uses the x parameter of the color code to determine the relative direction of the golfer and sends the appropriate pwm signal to the motors to turn it right and left. After trial and error testing with the motors, the code was adjusted to prevent jerky movements and have smooth transitions.

Another adjustment to improve the vision system efficiency was tuning the specific color signatures of the pixy. The final color code was determined to be yellow and orange since it is unlikely that there will be other yellow and orange objects interfering with the signal on the golf course. After the vest was made, by using PixyMon, a final programming of the color codes was made to ensure following which accommodated the natural lighting found on a golf course.

3.4. Object Detection (Ultrasonic)

Ultrasonic Object Detection

The Caddiellac will have a built-in ultrasonic object detection feature in addition to the golfer tracking system. The purpose of this detection system is to recognize objects in the immediate area that could be a hazard to the Caddiellac. Such objects include nearby trees, rocks, or other golfers that may appear in the path of the Caddiellac. The ultrasonic sensing will detect hazards such as these that are within the immediate path of the Caddiellac. It will then notify the Caddiellac so that it can change its course slightly to avoid the object, wait for the obstacle to pass out of its path, or pause for a manual override from the user in particularly difficult situations. Additionally the Caddiellac will use the distance information from the ultrasonic to stop an appropriate distance behind the golfer as the golfer approaches his/her ball.

Ultrasonic Sensor Module: HC-SR04

The sensor HC-SR04 was chosen for this application due to its availability and low cost.



Figure 3.4.1. HC-SR04

The HC-SR04 requires 5V on vcc pin and ground on the gnd pin. To start a measurement, the Trig pin of the sensor must receive a high signal (5V) for at least 10us. The sensor transmits out 8 cycles of 40kHz ultrasonic burst and waits for the reflected ultrasonic burst. When the receiver detects the reflection, it sets the Echo pin to high (5V) for a period (width) proportional to distance. The distance is calculated using the period of the Echo signal and the speed of sound.

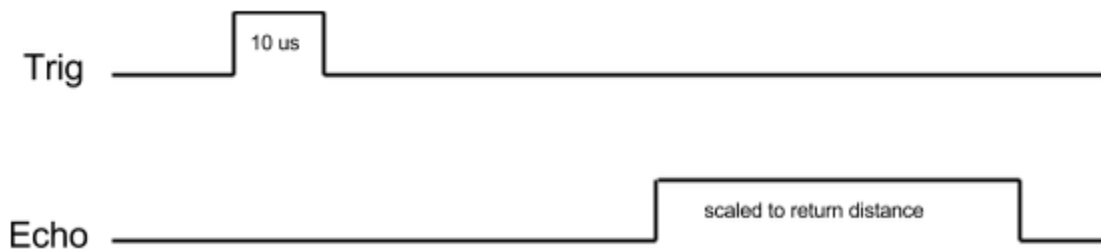


Figure 3.4.2. Signal Diagram of HC-SR04

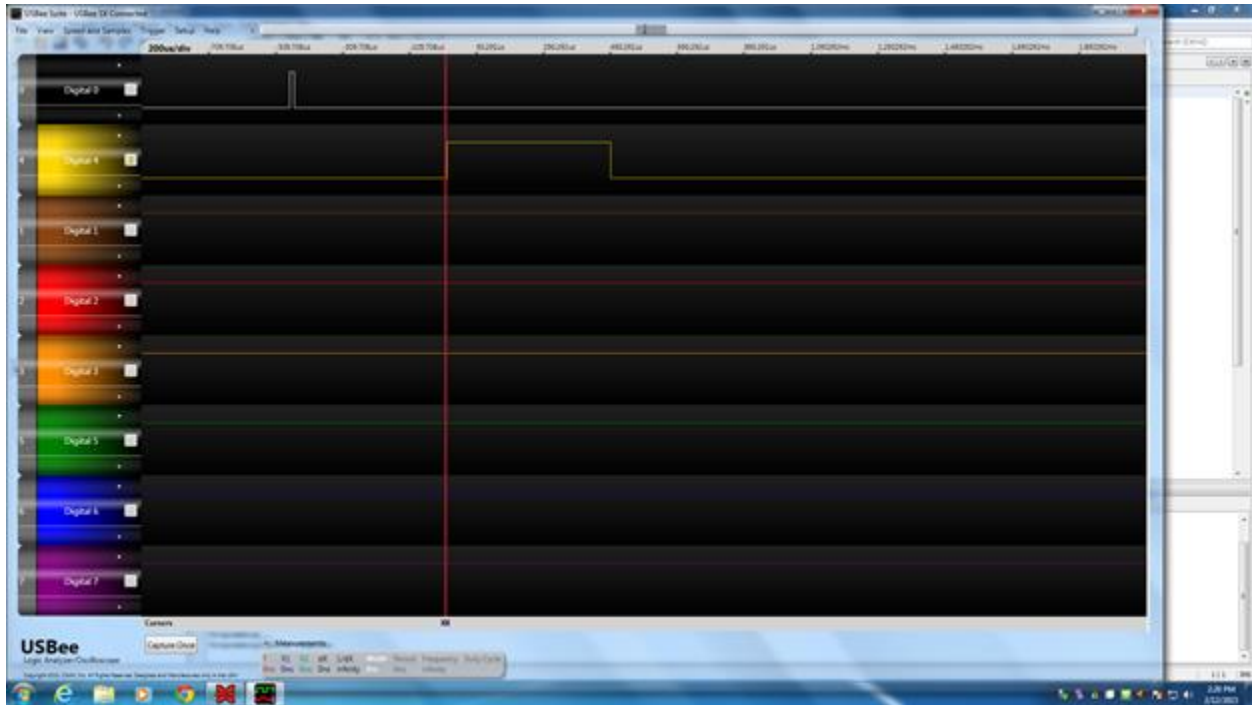


Figure 3.4.3. Measured Signal of HC-SR04

Timing the Echo Signal

In order to determine how long the echo pin is high for, interrupts are used. We have adjusted the prescaler values to set up Timer2 as a 10us counter. In our code, as soon as the Echo signal becomes high, a global variable named echocount is set to 0. After that, every time Timer2 interrupt becomes active 10 is added to echocount. This 10 corresponds to the 10us time that has passed. Once Echo turns low, the current echocount is used to determine distance. This method, however simple, loses precision. With this method there is room for error to be +/- 0.31cm. However, in such an application as we are using this is a very tolerable error margin.

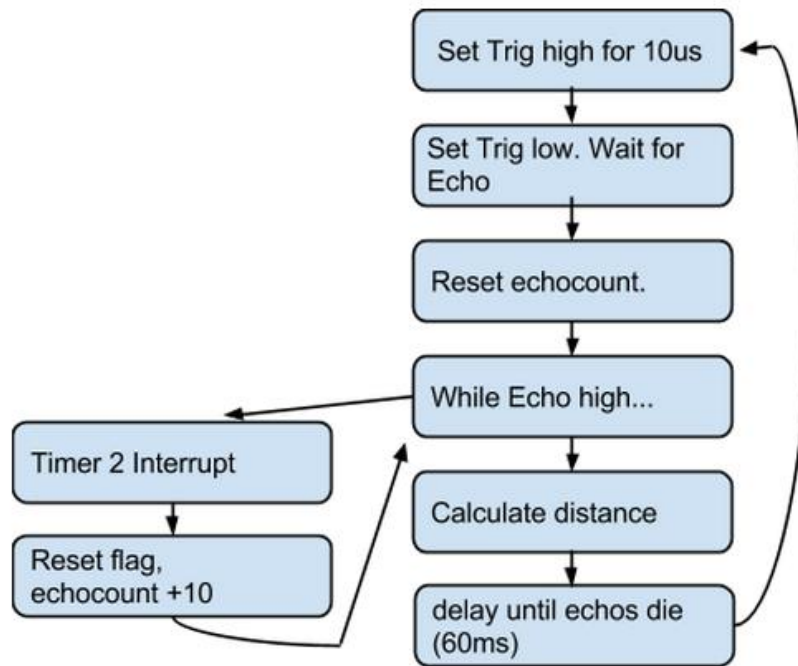


Figure 3.4.4. Flow chart of distance measurement

Testing

A program was written to correctly initialize the timer, activate the HC-SR04 sensor, and continuously calculate the distance in cm to the object in front of the sensor. The distance was displayed on the LCD display. The code is attached in the Appendix.

Subsystem Update

While sonar was working with the Pixy for the subsystem integration demo, overall, it was not the best option for a successful collision prevention subsystem. Since the Pixy grabs multiple parameters of the color code, the height parameter can determine how close the golfer is to the cart. By setting a conditional with a specific height that corresponds with an appropriate distance, the Caddiellac will only follow the golfer if he or she is far enough away.

In addition, the Pixy following system will ramp down the motors if the color code is not found, therefore if another golfer or object comes in between the golfer and Caddiellac, it will stop.

3.5. Manual Override (Bluetooth LE)

Bluetooth LE Device

An additional feature the Caddiellac will have is a manual override. This feature will be used for when the Caddiellac follows the golfer into tight spots, the golfer wants to suspend follow mode, or when the user wants to control it for other reasons (e.g. get out of the way of a shot). The Caddiellac will be Bluetooth enabled and an app will be available for download with a simple user interface to control and suspend the Caddiellac's movements.

Bluetooth LE Code

To set up the kit board to work with the Nordic nRFB001 chip, a basic Hello World code was taken from the Nordic website and modified to be used with a PIC chip. Once it was modified to work, the basic protocol was used and the Hello World program was modified to turn an LED on and off when a "1" and "0" were sent from a smartphone. The lines of code that were added are included in figure 2.10 and the rest of the code is available on Box.

Testing

Most of the testing was using the Hello World program, android Bluetooth terminal, and PuTTY to monitor and understand how the code was working. Once an understanding was achieved, the lines of code were added to take the Rx signal and control some aspect of the kit-board. This demonstrates the ability to send and receive information from a phone. For final implementation, the android app will send a single character on each button press, and this character will instruct the microcontroller to do a different action.

Subsystem Update

An app was written for Android smart phones to control the Bluetooth enabled Caddiellac. The app checks to make sure Bluetooth is enabled on the user's phone and if it is not sends a prompt to turn on the Bluetooth. The app has a simple GUI (figure 2.10) that automatically connects to a broadcasting Bluetooth LE device. This auto-connect feature was made with the consideration that very few devices use Bluetooth LE and the use of this cart will be on a golf course where individuals are very far apart from other groups. Additionally, given the function of this cart it would be tedious to connect manually to the Caddiellac

every time the user wishes to take manual control. The app connects to the NRF8001 via a simulated UART connection. The app sends an integer when each button is pressed down, and another when the button is released. In this way, we can tell when the user wants to enable Bluetooth override, turn left, right, or drive straight.

3.6. Self-Driving (Motor Control)

The team has chosen a basic configuration using two rear wheels (powered by two motors) and a caster wheel in the front of the Caddiellac to drive and turn. The best way to accomplish this is a motor control system. Some rudimentary background research was conducted in order to gain essential knowledge on how motor controllers operate. Since it was determined early on that purchasing a fabricated dual motor controller (approximately \$125.00) was out of the team's budget, a motor controller had to be designed and tested.

The motor controller subsystem requirements are as follows:

- Should be dual-driving (able to control two motors)
- Comply with power supply (12V battery)
- Able to successfully adjust motor speed from zero to maximum duty cycle
- Have a high enough resolution (PWM) for stable control of motor speed

After numerous hours of reading and research, it was determined that an H-Bridge motor controller could be a viable option for the Caddiellac. An H-bridge is essentially four power MOSFETs—two p-channel and two n-channel—configured so that two MOSFETs (one of the p-channel MOSFETs and one of the n-channel MOSFETs) are on at a time. Depending on the polarization of the voltage applied, the direction of current flow in the circuit would allow the motor to turn clockwise or counter-clockwise (corresponding to moving forward or reverse).

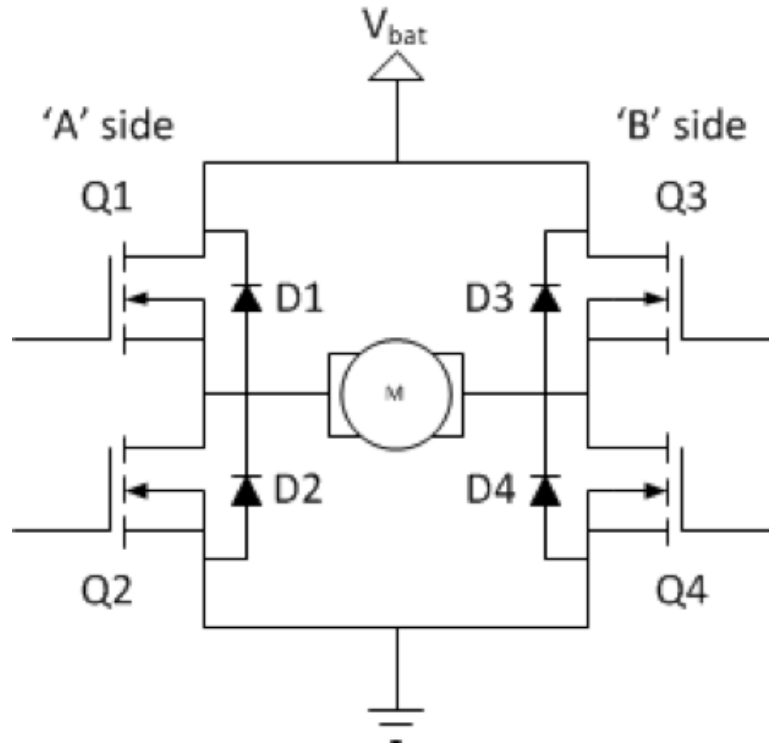


Figure 3.6.1. Simple H-Bridge

Image: <http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>

However, the team decided it was not necessary to drive the Caddiellac both forward and reverse (just forward required). It would add extraneous complexity to the project, and the same maneuverability is still attainable by rotating and driving forward. With this in mind, a simpler and more compact circuit would suffice in attaining our motor controller needs. Initially, a separate timer was a possibility to use pulse width modulation (PWM) to drive a single MOSFET, which switches on and off controlling the motor speed (one direction only); however, the PIC32MX695 microcontroller has the hardware built in to employ pulse width modulation. By driving the motors off the microcontroller, it would eliminate unnecessary parts and would help with communication with the Pixy camera (also being implemented through the microcontroller) in terms of distance and speed determinations. Thus, the decision was made to use the pulse width modulation mode from the microcontroller to drive a signal through a MOSFET driver and MOSFET, which would control the speed of the motor.

First, to get the PWM working from the microcontroller, a timer and output compare module had to be configured. After selecting a pre-scale (1:256) for the timer from the

peripheral bus clock (10 MHz) and configuring the resolution needed for a PWM frequency of approximately 2 kHz, the basic setup was ready to be implemented. By setting the OC2RS register with a number corresponding to the duty cycle of the PWM signal, the code was ready for motor control.

In determining a circuit for this application, simplicity and functionality were the main two priorities. Since the spec sheet of the microcontroller had a DC Motor Speed Control application circuit, this was the template used for the motor controller of the Caddiellac.

Figure 16-20: DC Motor Speed Control Application Schematic

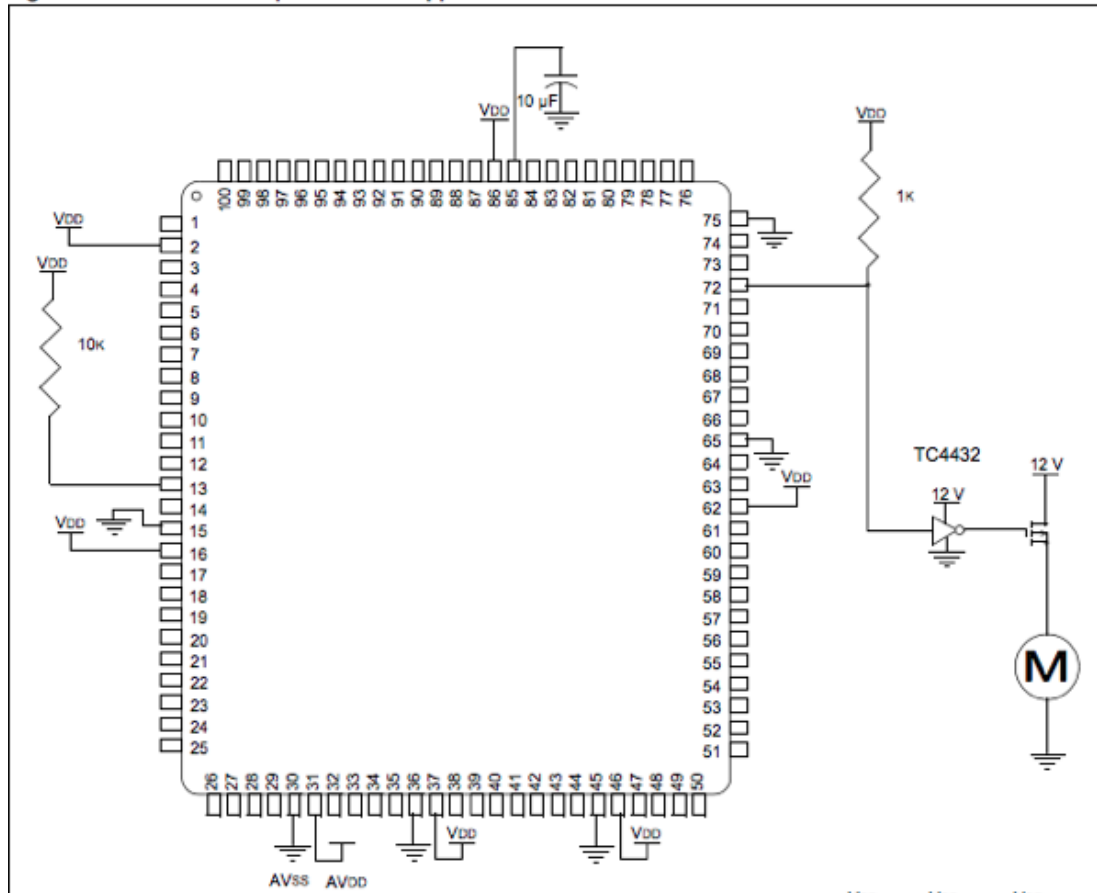


Figure 3.6.2: Schematic of a template DC Motor Speed Control from Microchip Spec Sheet
Image: <http://ww1.microchip.com/downloads/en/DeviceDoc/61111E.pdf>

The MOSFET and MOSFET driver had to be selected to fit current and voltage requirements. Since the team was supplied a RS-550 motor (motor chosen at conclusion of fall semester for our initial design), the components needed to comply with 12V. After finding a spreadsheet provided by Microchip, parameters were entered into the spreadsheet to find a suitable MOSFET driver to go along with a chosen MOSFET. After three MOSFETs

were entered (two were disregarded due to long factory lead times through suppliers), the IRLB8743 n-channel Power MOSFET was chosen to be the MOSFET of our circuit. The team chose an n-channel MOSFET because of its low $R_{DS(ON)}$ value (due to electrons having higher mobility than holes) and thus faster switching speed. The MOSFET driver that matched it was the TC427EPA Dual MOSFET driver. This was chosen for its ability to control two motors, which is needed for the Caddiellac.

The circuit above was implemented through a breadboard and after some trial and error, the team got the motor controller to work. Some issues encountered along the way included a malfunctioning TC427EPA (which the team simply replaced), and failing to realize the breadboard Vdd and Ground rails were split in the middle of the breadboard, half and half (rather than the entire vertical rail). This led to an easy fix, and the circuit functioned as it should. It was necessary to ramp up the motor (to turn on) and ramp down the motor (to turn off) in order for safe operation. After the first design review, it has been determined that the resolution of the PWM signal must be increased in order to have increased control and stability on the motor speed.

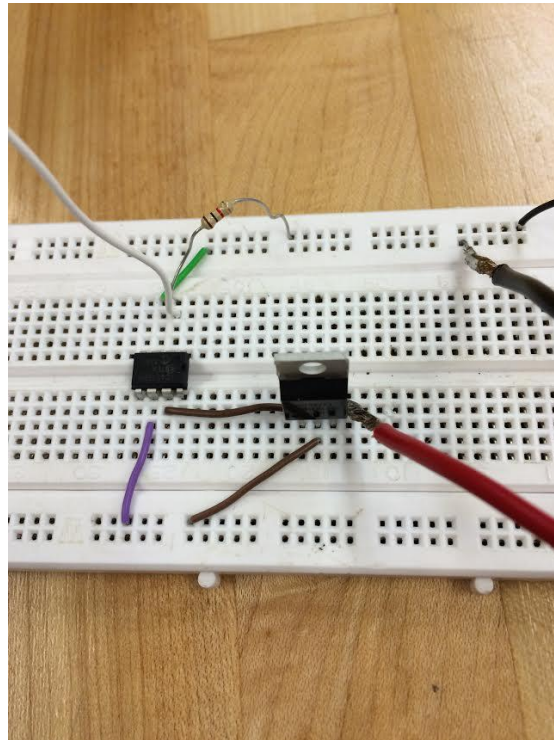


Figure 3.6.3: Motor Controller Circuit connections (with IRLB8743 MOSFET and TC427EPA MOSFET Driver)

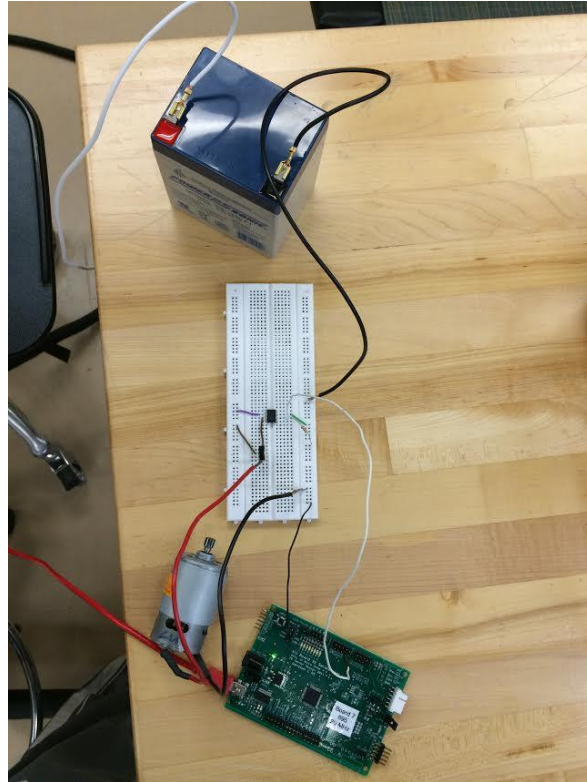


Figure 3.6.4. Entire Motor Controller Subsystem (with 12V motor and battery connected, PWM signal from PIC32MX695F512H Microcontroller)

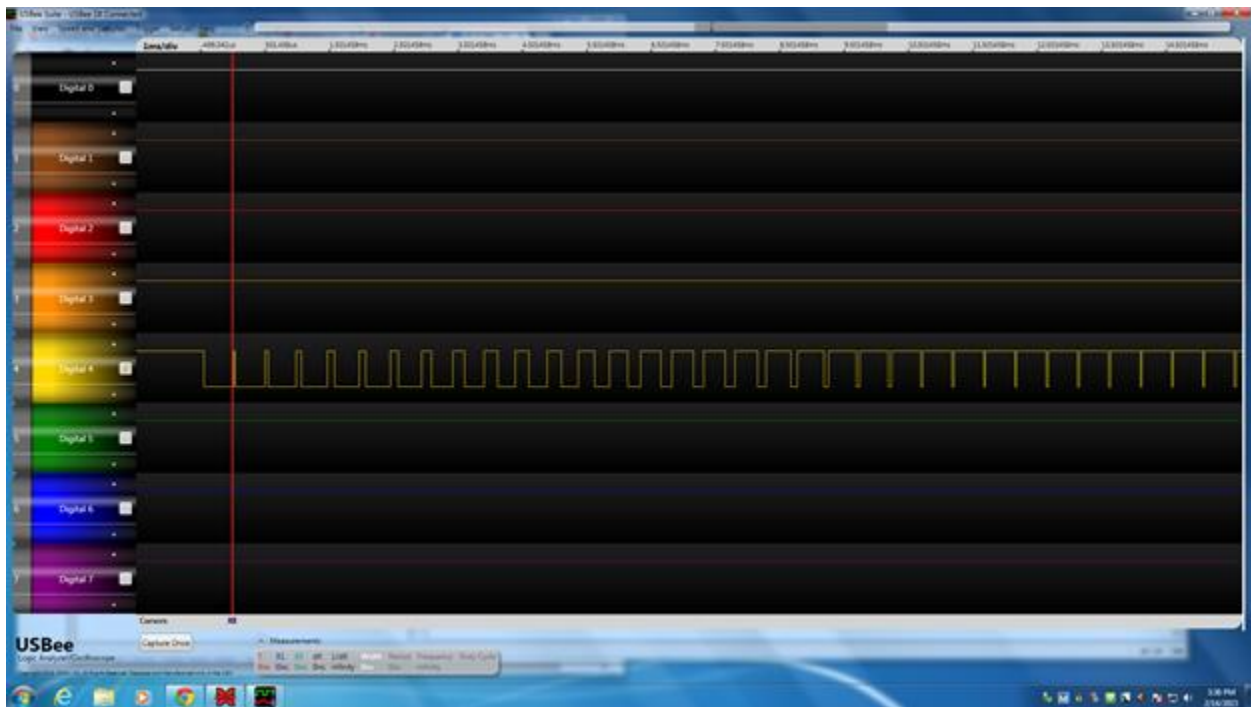


Figure 3.6.5. Ramp up Duty Cycle (PWM)

Subsystem Update

After thorough testing, it was found that the configuration used at the first subsystem demo was insufficient. The initial arrangement (using the TC427 MOSFET driver) sporadically worked and the team concluded that the driver was inadequate and could not handle the current being put through it. A new schematic was created for the motor controllers also employing bypass capacitors between Vdd and ground. Also, heatsinks were applied to the MOSFETs to control / minimize heat. A final addition to the motor controllers was a flyback diode. The function of the flyback diode is eliminate voltage spikes across an inductive load (motor) when the supply voltage is removed or shut off (battery).

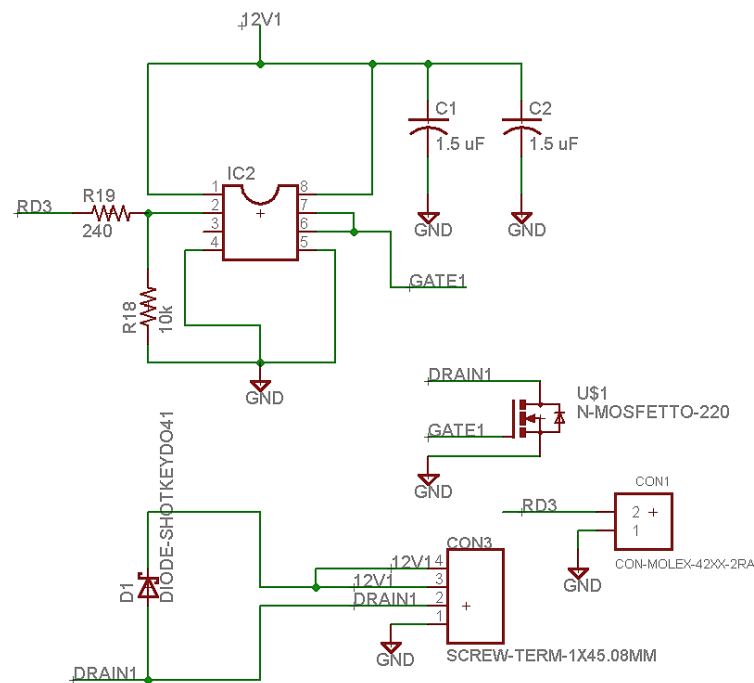


Figure 3.6.6. Final Motor Controller Design

3.7. Interfaces

The subsystems communicate with each other through a hierarchy of commands. The overarching control is the app, which communicates to the board via Bluetooth LE, and allows the golfer to enable and disable the Caddiellac movement, as well as control its direction once it is enabled. If the app is not enabled and the cart is switched on, the motors are automatically controlled by the Pixy over SPI. Since the pixy code accounts for following the

golfer as well as situations in which the color code is lost or the golfer is too close to the Caddiellac, it encompasses the vision system as well as collision prevention. Previously, the collision prevention categorized as object detection with the sonar sensor.

4. System Integration Testing

The integration of the subsystems of the Caddiellac are best thought of as a hierarchy of commands. The default mode is when the Caddiellac is controlled by the vision system and using the information from the PixyCam to determine the speed of the motors. However, if the sonar sensor detects an object within a certain range, the Caddiellac will stop and not follow the color code until the sonar detects a clear path. The app that communicates through Bluetooth overrides all other subsystems to provide the user with a failsafe controller to navigate the Caddiellac in any circumstance.

Integration testing began from the bottom up, starting with the integration of the PixyCam data and the motor controllers. The PixyCam captures a color code in a rectangle and outputs various information including the relative x location of the rectangle within the entire frame. This allows us to determine where the relative position of the color code is and from the value of x, we can determine if the Caddiellac needs to turn right or left. Within a series of conditional statements, depending on this x value, the motors will appropriately ramp up or down to turn and follow the color code. Since the motor controllers were not fully functioning at this point in testing, this subsystem integration was testing using the LCD screen by outputting the duty cycle values of the right and left motors while moving the color code. Once this worked, we were confident that the values of the motor controller pins were outputting the correct values and moved on to integrate the sonar subsystem.

The sonar sensor was integrated by defining a threshold distance in which the Caddiellac would ignore the vision system and automatically stop. Once the path is clear again, the Caddiellac will continue to follow the color code with the vision system controls. From the software perspective, this just required framing the existing vision system code with a conditional concerning the output of the sonar sensor.

The Bluetooth app, which was developed separately, provides the override commands for the Caddiellac. A command from the app will stop and steer the Caddiellac and cancel out

all other commands from the vision and sonar system, allowing the user with maximum control in any situation. Integration consisted of determining the specific commands for each button of the app as well as allowing the Caddiellac to be controlled by the vision and sonar system when no information is being sent from the app. The app sends number each time a button is pressed and a different number when the button is released, allowing the overall code assist in accuracy of the motor controls. The app must be enabled which restricts the code to just follow the app commands, otherwise, it returns to checking sonar and following signals from the vision system.

Various problems occurred during testing, mainly due to the limitations of the kitboard. Since we have 2 subsystems functioning through SPI, plus a primary testing aide of the LCD screen, several pins would overlap in function and cause problems in proper communication. Overall, with the addition of another kitboard, the subsystems all worked properly to control the motor in the correct hierarchy.

Once the team board was completed, testing mainly consisted of getting each subsystem to work with the board, followed by integrating the subsystems together. First, testing the Pixy required proper setup of SPI2 and correct cable wiring. The label on the team board for MOSI and MISO were switched, plus the code was attempting to run off of the external oscillator. Once these errors were corrected, the Pixy successfully communicated over SPI and could be integrated with the motor controls.

The first error with the motor control subsystem was that the code was not updated to reflect the change in output compare pin from 2 and 3 to 3 and 4. Once this adjustment was made, a pwm signal was observed on the USBee for both left and right motors. Using the code from the previous subsystem design, without the sonar functions, the Pixy successfully communicated the color code information with the motor controllers. Another detail that was taken care of was finalizing a color code and making a vest for testing.

Alongside this code update, the physical prototype of the Caddiellac was developed to a more robust design as seen in the figure below. It now includes a caster wheel and a platform for the battery and boards for an overall more stable design.

Once the prototype was completed, testing with the Pixy and motor controls began. It is important to note that in order for the Pixy and motors to successfully communicate, the speed must be initialized to zero. Once it followed the color code down the hallway, slight

changes to the motor functions were made to account for turning adjustments and better speed control. The pixy code was also updated to account for false color code detection if the golfer is too close.

The best demonstration of how the Caddiellac meets the design requirements can be seen in the videos on our website. The app successfully overrides all other commands and controls the Caddiellac. The vision system successfully maneuvers the cart to follow the golfer based on the relative location of the color code. The Caddiellac will stop if something comes in between the golfer and the cart, as well as if the golfer gets too close to the cart. All of these scenarios were tested by physically using the Caddiellac in a controlled environment.



Figure 4.1. Final Caddiellac Design

5. To-Market Design Changes

In order to bring The Caddiellac to market, some further development would be required. Due to budget and time limitations, the final project only represents a prototype of the product which would be made commercially available. Most notably, the mechanical design needs many additional features. The prototype is quite cumbersome, and provides some difficulty moving in and out of a vehicle. The market-ready product would include a more durable, lightweight, and collapsible design which could easily be stored in the trunk of a car. Additional design features would include a cup holder, cigar clip, and other added compartments.

Updates would also need to be made to the mobile application before it was released. The application is currently available for Android, but a version for Apple would also need to be developed to allow for the same level of use for iPhone consumers. Additional features could also be incorporated into the application, such as course maps and a scorecard. GPS could also be included into the application in order to detect the distance between the golfer and the hole, and then recommend clubs accordingly. More robust controls could also be developed for the application, including a joystick-inspired control instead of simply left, right, and forward arrows.

In a market-ready product, personalization of the color code would need to be incorporated. Each golfer would need to have a unique code in order to ensure that The Caddiellac does not become distracted by the code worn by a different golfer. The most robust solution to this problem would be to allow the camera to detect whatever the golfer was wearing that day when a button was hit on either the mobile application or on The Caddiellac itself, and that pattern or color would be stored in the code and followed along the course.

6. Conclusions

Overall, the Caddiellac was a success. While there is always room for improvement, it is important to keep in mind that this is just a first attempt at a prototype. While the path to a final product was filled with various challenges and setbacks, at the end of the day, the hard work paid off. The Caddiellac follows a golfer, avoids collisions, can be controlled

remotely by a smartphone app, and can last for a full round of golf. The basic requirements for project were completed, and while minor adjustments can be made for improvement, the overall functionality of the Caddiellac is effective and successful.



Appendices: Attached

Appendix A: Hardware Schematics and Component Data Sheets

Appendix B: User Manual

Appendix C: Final Code