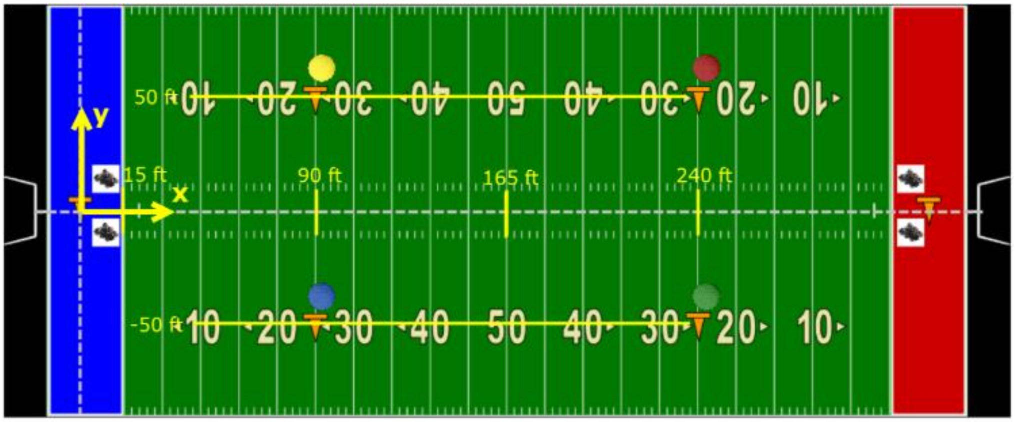
Problem Statement:

The Golf Ball Delivery project took place on a football field (Figure 1) where a mobile robot started in one of the end zones with three golf balls of different colors, determined the color of each golf ball, delivered each golf ball to its appropriate GPS location based on the color of the golf ball, and then returned safely back to the starting position.



(15,0)	Robot start location
(90, 50)	Yellow ball target cone
(90, -50)	Blue ball target cone
(165+, Y)	Beyond the 50 yard line where the White ball could go
(240, 50)	Red ball target cone
(240, -50)	Green ball target cone
(0, 0)	Home cone that you must come back to

Figure 1. The layout of the football field for the project.

Mobile Robot Hardware:

The mobile robot consisted of a Wild Thumper 6WD chassis (Figure 2), a Dagu robotic arm with five degrees-of-freedom and six servos (Figure 3), a Nexus 7 Android tablet, a custom-built golf ball stand, an Arduino Mega ADK 2560, and a regular Arduino Mega ADK.

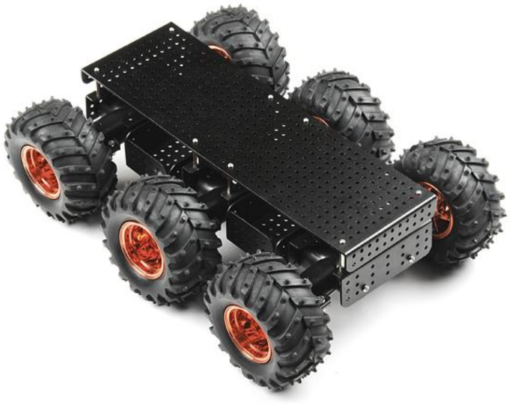


Figure 2. Wild Thumper 6WD chassis.

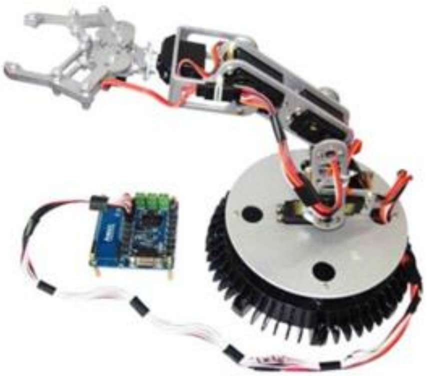


Figure 3. Dagu robotic arm.

The golf ball stand consisted of a wooden platform and three spherical holes for each of the golf balls to be placed in. The wooden platform contained 18 LED's of which six were red, six were blue, and six were green. The LED's were placed at six different locations on the platform with each location containing one red LED, one blue LED, and one green LED. The wooden platform also contained four photodiodes. One photodiode, one red LED, one blue LED, and one green LED were placed at the bottom of each spherical hole. The remaining LED's and fourth photodiode were placed outside the spherical holes (Figure 4).

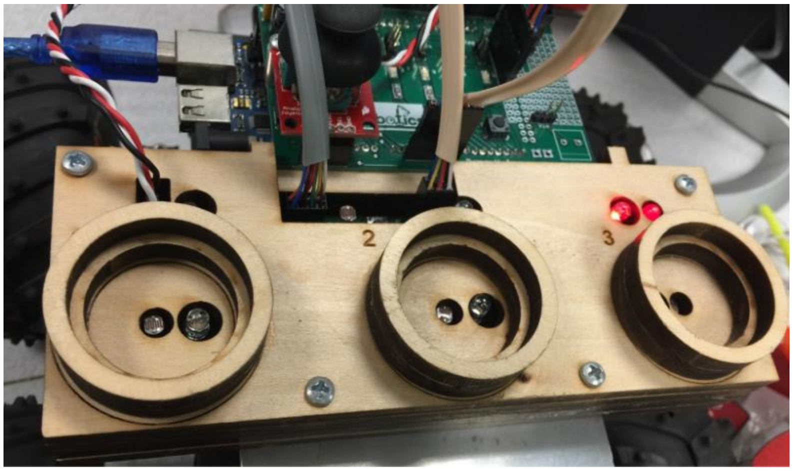


Figure 4. Custom-built golf ball stand.

Figure 5 displays all 18 LED's turned on simultaneously, with a white sheet of paper placed over top for better clarity.

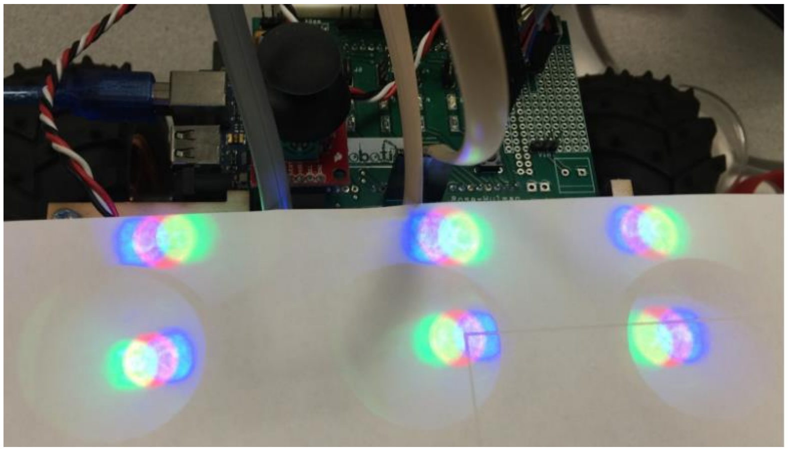


Figure 5. All 18 LED's on the golf ball stand turned on simultaneously.

A custom-made PCB was provided for connecting the components of the golf ball stand (Figure 6).

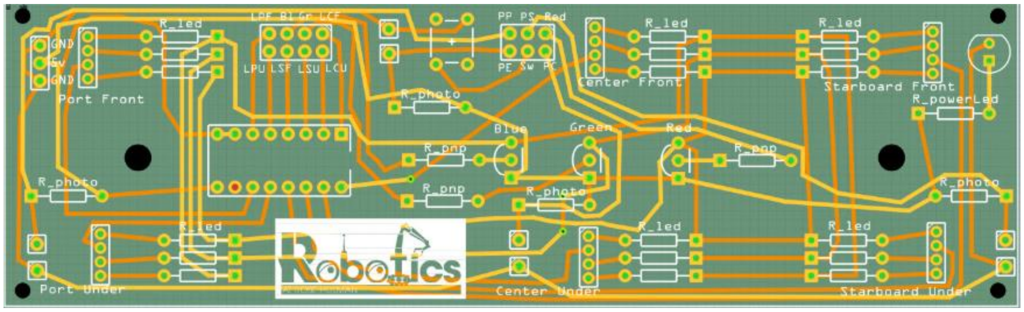


Figure 6. Custom-made PCB for golf ball stand components.

Schematics for the LED outputs and photodiode inputs for the golf ball stand are shown in Figures 7 and 8.

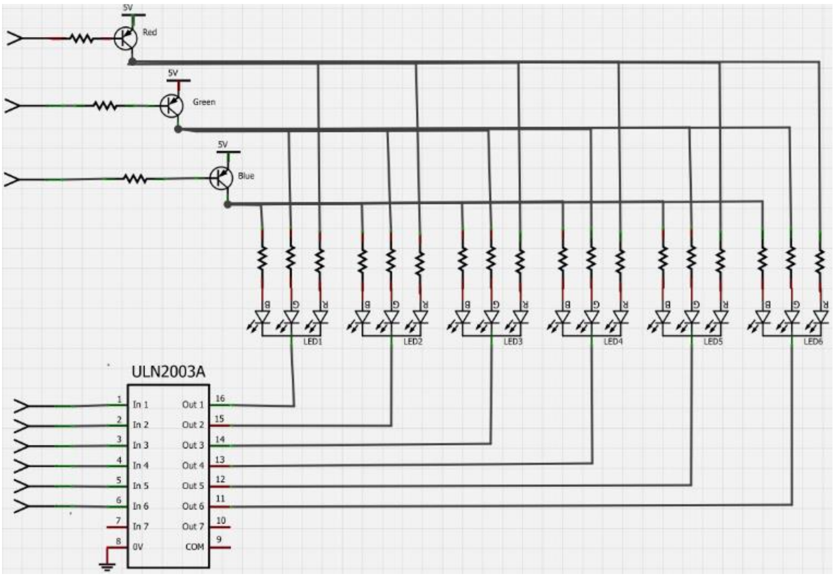


Figure 7. Schematic for LED outputs for golf ball stand.

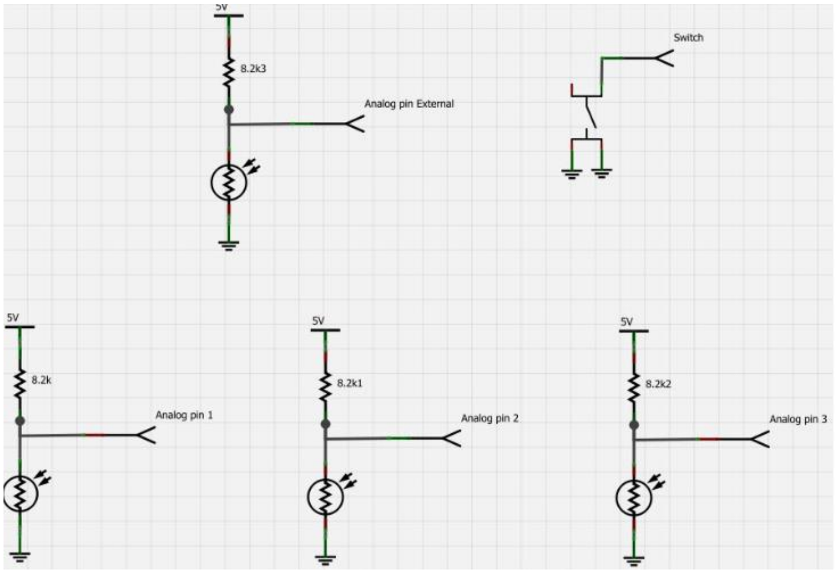


Figure 8. Schematic for photodiode inputs for golf ball stand.

The regular Arduino Mega ADK (Figure 9) was used to control the robotic arm and the Wild Thumper as well as communicate with the ADK 2560. The Arduino Mega ADK 2560 (Figure 10) was used to control the golf ball stand components as well as communicate with both the regular Arduino Mega ADK and the Android tablet.

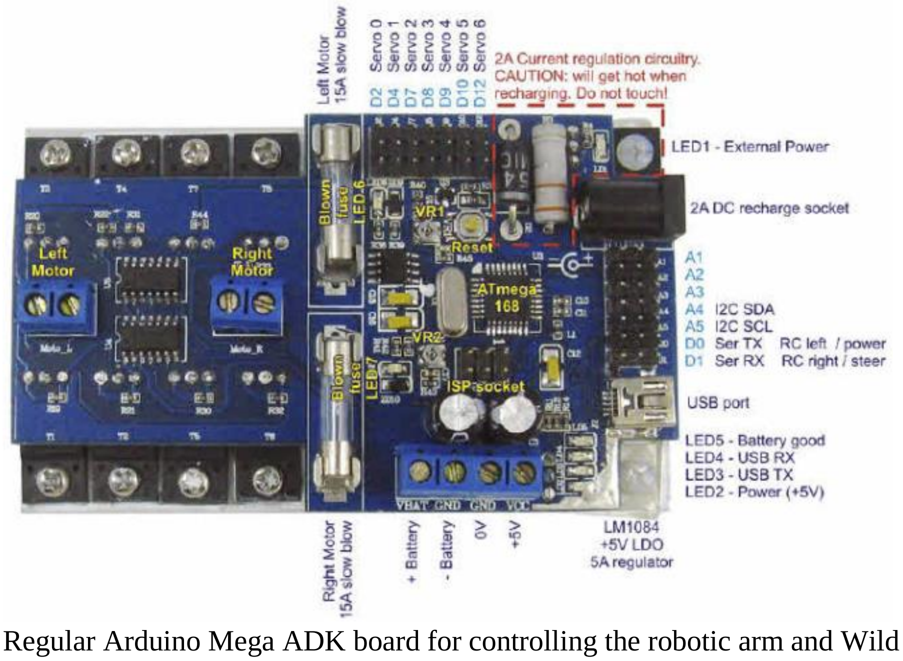


Figure 9. Regular Arduino Mega ADK board for controlling the robotic arm and Wild Thumper.

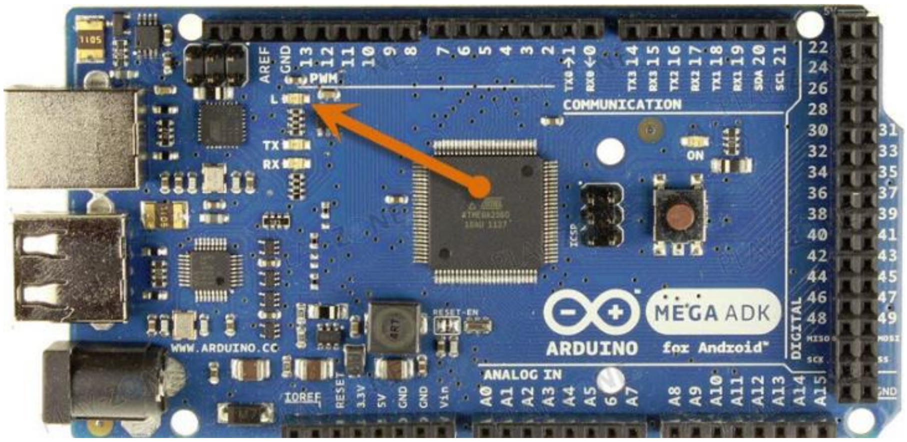


Figure 10. Arduino Mega ADK 2560 board for controlling the golf ball stand components and communicating with the regular Arduino ADK board and the Android tablet.

<u>Arduino Libraries:</u>
Several libraries were used with the Arduino boards for communication and hardware control; some of these libraries came installed with the Arduino IDE, others were custom-built. Below is a list of the libraries, with descriptions of each:

1) <u>RobotAsciiCom Library –</u> Custom library used on the Arduino Mega ADK 2560 for communication between it and the Android tablet. Communication is through messages that contain ASCII characters. The library parses messages received from the Android tablet, interprets them, and then the main code (ArduinoADKProjectCode.ino) performs the appropriate actions.

2) <u>WildThumperCom Library –</u> Custom library used on both Arduino ADK boards. Both boards use the library to communicate with one another via binary.

3) <u>GolfBallStand Library –</u> Custom library used on the Arduino Mega ADK 2560 to control the golf ball stand components. It provides methods that can be used to turn the LED's on and off as well as return the analog values of the photodiode sensors.

4) <u>Max3421, Usb, and AndroidAccessory Libraries –</u> Used on the Arduino Mega ADK 2560. Since Android tablets don't use serial communication, these three libraries make communication between the Android tablet and the ADK 2560 possible via SPI over the USB cable.

5) <u>Arduino Servo Library –</u> Built-in Arduino servo library that provides method for moving a servo a certain number of degrees.

6) ArmServos Library – Custom library that converts servo angles into Denavit-Hartenberg representation angles.

7) ArmServosSpeedControlled Library – Custom library that is a subclass of the ArmServos library. It contains all methods from ArmServos and additional methods that allow for better control of the speed of the robot's servos. It uses an ease-in ease-out algorithm to

move the joints slower at the beginning and end of a move, resulting in less jerk over the robot’s movement.

8) LiquidCrystal Library – Built-in Arduino library that is used for controlling an LCD and displaying messages on it, in this case controls the LCD on the Arduino Mega ADK 2560.

Android App:

NOTE: This file is still not finished. Had trouble getting the emulator to work in Android Studio and couldn't view app design in the design editor. Once the emulator is working and screenshots of the different menus within the app can be taken, the "Android App" section of this file will be updated and will display the various menus of the app as well as explain the purpose and functionality of each menu.