**Final Project: Bluetooth Controlled Car**

**CS M117 with Professor Revaz P. Dzhanidze**

**Date: June 6th, 2018**

**Team Name: “The Procrastinators”**

**Team Members:**

Nathan Knight 004 749 179

Kate Baldwin 804 847 859

David Feng 604 756 204

Nick Bruce 504 675 519

Harrison Yuan 504 634 331

Gwendolyn Hammons 104 584 253

**Motivation**

The objective of this the lab is to learn about utilizing Bluetooth wireless technology to design a physical object that responds to commands and learn about how this popular wireless technology is often implemented in everyday devices. Our team decided to to explore the effectiveness and limitations of Bluetooth technology through a simple, fun, and engaging project of designing a remote controlled car.

In the designing phase, it was determined that given the abundant resources to work with Arduino software and hardware alongside with the compatibility of Arduino products with Android, the Arduino platform provided a solid foundation to base the project. A physical object that responded to different wireless commands at a distance in order to carry out Bluetooth experiments was used in this project. As a result, a bluetooth controlled car seemed to best satisfy our objectives and specifications for the project.

**Functionality**

The vehicle will be made using 2 DC motors, Arduino Uno, Bluetooth HC05, wheels, and a chassis. Our device has three key features. Bluetooth connectivity was implemented using the Google Pixel Android in conjunction with an app developed using resources from the MIT App Inventor. Android application communicates with our car wirelessly using the bluetooth 2.0 protocol. Bluetooth 2.x is the most popular variant, especially 2.1. Bluetooth 2.x introduces the optional Enhanced Data Rate (EDR) capability. Devices with basic rate + EDR (optional) have theoretical maximum data rate of 3 Mbps. The 2.1 version makes it easier to pair with devices from different manufacturers and increases the reliability of the pairing process. Both motors on our car work in coordination to perform 4 basic functions: move forward, move backwards, turn to the right, turn to the left. Along with controlling the motors, the app controls a light and sound button. The light button triggers a series of flashing red and blue lights similar to a police car with a sound button ringing sirens.

**Wireless Technology**

Bluetooth was the wireless technology utilized in this project. The HC05- bluetooth module connected to the Arduino Uno. This acts as the interface between the Arduino and the Android phone. As the user inputs commands to the phone, the phone generates a signal that is received by the bluetooth module. This information is then relayed to the DC motors which instructs the motors to rotate in a particular direction. Specifically, the C/C++ code incorporated specific functions that details how the wheels move in order for the car to move “forward”, “backwards”, “left” , and “right”.

The device exchanges data over short distances using electromagnetic waves in the 2.4 GHz range. It can be configured as either slave or master mode to relay data. The chip is connected via a serial port uses CMOS technology with Adaptive Frequency Hopping Feature used in bluetooth technologies.

**Hardware** Two sets of 4 pack AA batteries are connected in series two increase the voltage supplied

into the electronics. This allows for the car to move faster. These batteries primarily power the two servo motors but also provide secondary power to the Arduino Uno which acts as the main frame for the system. These materials are physically attached to the chassis that holds the components together. The dual DC motors are attached below the chassis for motor functions giving the car the capability to move. A third caster wheel acts as the back wheel to balance the chassis but has no engine functionality.

**Design**

In our initial design of this project, the group anticipated 3D printing many of the different aspects of our car including the body of the car itself and the wheels that would go along with it. Early on, it became clear that this wasn’t going to be a time or cost effective method for us to continue with. Our second attempt was to just start out with the basics: an Arduino Uno, Bluetooth HC05, and two stepper motors. While the first prototype was able to connect over bluetooth and control the motors, the motors were rotating very slowly and at the same time, the group was struggling to find wheels that would work well for our car and the stepper motors. This lead to the decision to purchase UCTRONICS Smart Robot Car Kit for Arduino. It solved three of our main issues: the DC motors ran faster, it came with wheels and the chassis was already constructed. This allowed us to focus more on the Bluetooth research and experimentation that we’d set out to do.

In order to send out bluetooth signals to control our remote car, we decided to build a simple app on Android and utilize the built-in bluetooth functionality of all Android phones. Our initial plan was to build the app from scratch, but this proved to be impractical given the time frame for the project. Instead, we were able to use MIT’s App Inventor tool to quickly prototype an app with a functional user interface that could be tested on the Android phone immediately. This proved very effective for the iterative design process of going through trial and error in ensuring the controls in the app corresponded to the desired movement behavior in the car. The app contains a bluetooth search list that allows us to look for all available bluetooth connections nearby. Once the connection with the car’s Arduino bluetooth system is established, the app sends out various string messages that is interpreted by the Arduino code to move the car.

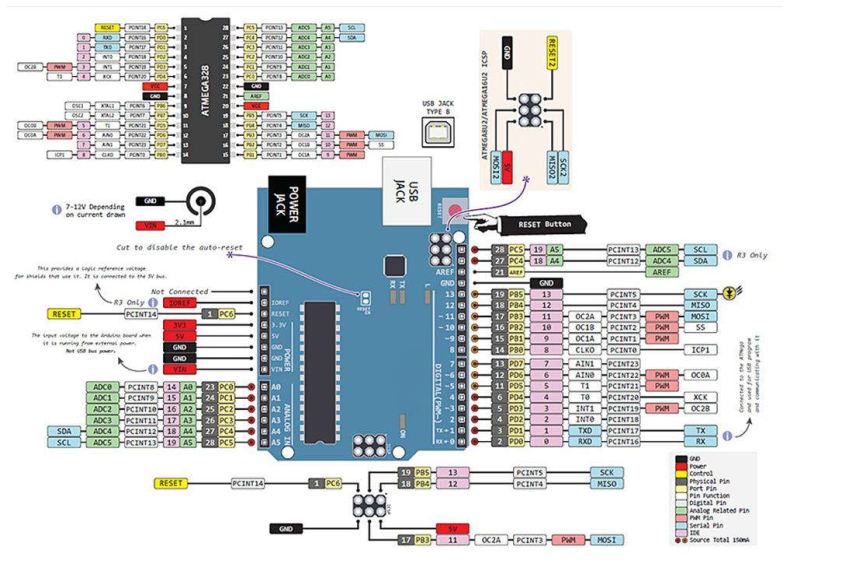
**Implementation**

The group built the UCTRONICS car using the pieces supplied inside the kit. Once everything was connected and put together, the implementation of the code and app began. As mentioned in the design portion of this report, the original code controlled two stepper motors that ran extremely slowly. The reason for this slowness is because stepper motors are normally used for precise rotations such as robotics in an assembly line. The stepper motors were controlled via the Stepper library provided by Arduino. This was another reason the team originally went with the stepper motors. Once the new DC motors came in the mail, the team implemented the new motors with the AFMotor library provided by a third party named Adafruit. The car is mainly controlled by a switch statement that is based on the given input from Bluetooth. In other words, if the user presses a particular button on the app, the app will send a character to the Bluetooth shield and the Uno will read this in via the Serial.read() command. Once a character is read in, the switch statement will execute based off of that character. Characters ‘0’, ‘1’, ‘2’, ‘3’, and ‘4’ will control the motors and ultimately the movement of the car. Characters ‘5’ and ‘6’ will control the lights and sound of the vehicle; ‘0’ makes the car stationary with no lights and no sound, ‘1’ moves the car forward, ‘2’ moves the car backward, ‘3’ turns the car left, ‘4’ turns the car right, ‘5’ turns the lights on, and ‘6’ turns the buzzer on.

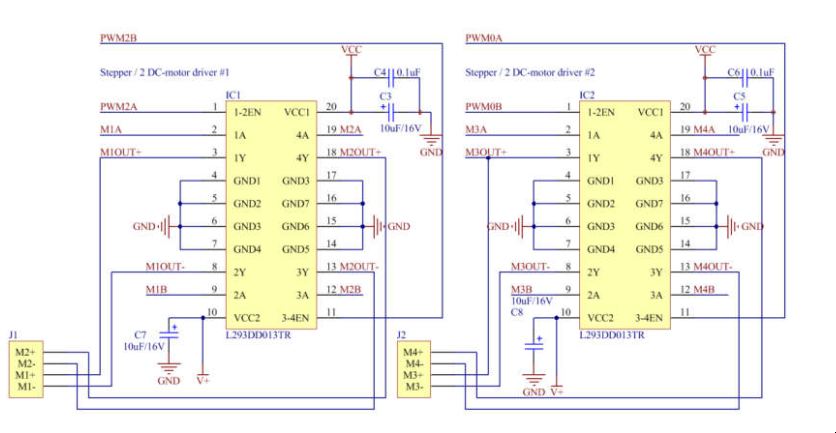
As stated above, the app was made in MIT’s App Inventor 2. The app provided basic connection events to map to button presses. First, we had to create the basic user interface layout of buttons corresponding to the different directional movements. The app included a method to load up the list of all available Bluetooth connections within the Android phone. Once the connection was established, we were able to use the different Bluetooth events that the app provided to map to button presses we created in the user interface. Using the block design diagram for connection events, we transmitted different command strings through a Bluetooth connection that would be interpreted by the Arduino software.

**Conclusion**

In summary, a robot with motor functions was designed with bluetooth capability to demonstrate wireless technology in application. By controlling a robot wirelessly with an Android phone, it is demonstrated that the signal can be transmitted across 2.4 GHz band to communicate with an Arduino. This signal is translated and used to move the servo motors in a direction.

****

**Figure 1: Pinout Arduino Uno**

****

**Figure 2: Pinout Motor Shield**

**Contributions**

Mechanical aspects / Troubleshooting Functionality:

1. Nick Bruce
2. David Feng

Implementation of Android App

1. Harrison Yuan
2. Gwendolyn Hammons

Implementation of the Arduino Code

1. Kate Baldwin
2. Nathan Knight