RC Car Project Documentation

The embedded system project that was implemented was a remote control car with an LED matrix display, range detection, and collision avoidance.

The following features were implemented as part of the embedded system:

1. DC motors in order to spin the wheels as well as turn left and right (using a gearbox).

2. Universal Synchronous/Asynchronous Receiver/Transmitter (also known as USART) communication protocol.

3. A Joystick input to direct the car movement.

4. Distance of objects in front of the car using the UltraSonic Ranging Module HC-04.

5. An LED matrix to display distance of an object in front of the car by using the UltraSonic Ranging Module HC-04 input.

6. Wireless/Bluetooth Communication between the controller and the car via the HC-05 Bluetooth Module.

7. Collision detection and avoidance to stop the car from hitting objects.

8. L293D motor driver to power DC motors.

9. 2 SN74HC595N shift registers used to power LED matrix.

How it Works (Software/Functionality General Overview)

A high level description of the code for the controller is a follows. The joystick uses two ADC (Analog Digital) inputs, one for left and right, the other for up and down. The up and down on the joystick corresponds to forward and backward on the RC car. The two values from the joystick are scaled to a value between 1 and 10. Next there is a two-bit encoding scheme placed on the two leading bits to allow the receiver to distinguish between forward/backward and left/right instructions given from the

controller. This is then placed into an array and sent to the receiver slave bluetooth module via the USART protocol.

The receiver saves the transmission sent from the controller to a variable using USART protocol. This data is then decoded by checking the two leading bits used for the encoding scheme. If the data sent was a left/right instruction, then it is used to turn the car left/right. If the data sent was a forward/backward instruction, then it is used to move the car forward/backward. Both the turning the wheels left/right and turning the back wheels to move forward/backward are done by DC motors. The turning motor uses a gearbox in order to turn the wheels. The outputs for the DC motors come out of the atmega 1284 microcontroller and are inputs to a L293D motor driver. The motor driver then uses these inputs to drive the two motors.

Lastly, an ultrasonic ranging module HC-04 was used to detect the distance of an object in front of the car. The ranging module is another ADC input the value was scaled down then used to display the distance of the object on the 8x8 LED matrix, which is visible on top of the car. The distance matrix display works like this; The matrix's rows correspond to the distance. If all 8 of the matrix rows are lit, then the object is far away. If only 1 row is lit, then the object is as close as the ranging module can properly detect. When an object is too close, then the car no longer allows moving forward, only reverse.

Circuit Description:

Figure 1 is a picture of the controller circuit. It contains a 2D joystick, a HC-05 bluetooth module and the atmega1284 microcontroller. The joystick produces two analog values that are for left/right and up/down. These were each designated a pin on PORTA of the atmega because it have analog to digital conversion on said port. Next the HC-05 bluetooth module was induced into a "master" state by using an arduino to send "AT" commands to via the serial port. After this was accomplished the HC-05 was hooked up to the RXD1 and TXD1 ports on the atmega such that they went to TX and RX on the bluetooth module respectively.

The receiver is illustrated in Figure 2, which contains two atmega 1284's, 2 SN74HC595N shift registers, a L293D motor driver, a HC-05 bluetooth module activated as a "slave", the HC-SR04 ultrasonic rangefinder and an 8x8 LED Matrix. The first "receiver" atmega receives bluetooth signals via RXD1 and TXD1 wired in a fashion similar to the controller. The receiver also controls two motors using a two bit signal sent from PB0 and PB1 for L/R control and from PA1 and PA0 for F/R control. The reciever recieves a collision signal on PC0 from the second atmega, which informs the forward motor of an occurrence of a collision. The second atmega controls two shift registers on from PA0-PA3, and the other wired on PC0-PC3. This effectively controls the LED

matrix. Finally the HC-SR04 range finder has the trigger and echo wired to PD0 and PD2 respectively.

User Guide:

The user interface is very intuitive and natural. First one must turn the controller on then the car. After waiting for the Bluetooth to pair the car is active. At this point one can turn left and right by moving the joystick left and right, forward and reverse work in the same intuitive fashion. The led matrix on the cab of the truck acts as a gauge of the range finder. Once the rangefinder is within the predetermined “collision range” the car will no longer be able to move forward and only the reverse functionality is available.

Known Bugs and Issues:

A known bug is that when the range sensor is at a distance greater than zero and less than the collision distance than there is a flickering that happens between the default state and the hazard state. Another issue is lack of speed. Even when twelve volts was applied to the motor driver the motors moved the car very slowly.

Resources Used:

Credit and Link to functions used to read multiple ADC inputs:

https://hekilledmywire.wordpress.com/2011/03/16/using-the-adc-tutorial-part-5/

Credit and Link to some code used in Ultrasonic Rangefinder function:

http://circuitdigest.com/microcontroller-projects/distance-measurement-using-hc-sr04-avr

Video Link:

https://youtu.be/Wt7KeqgwinU

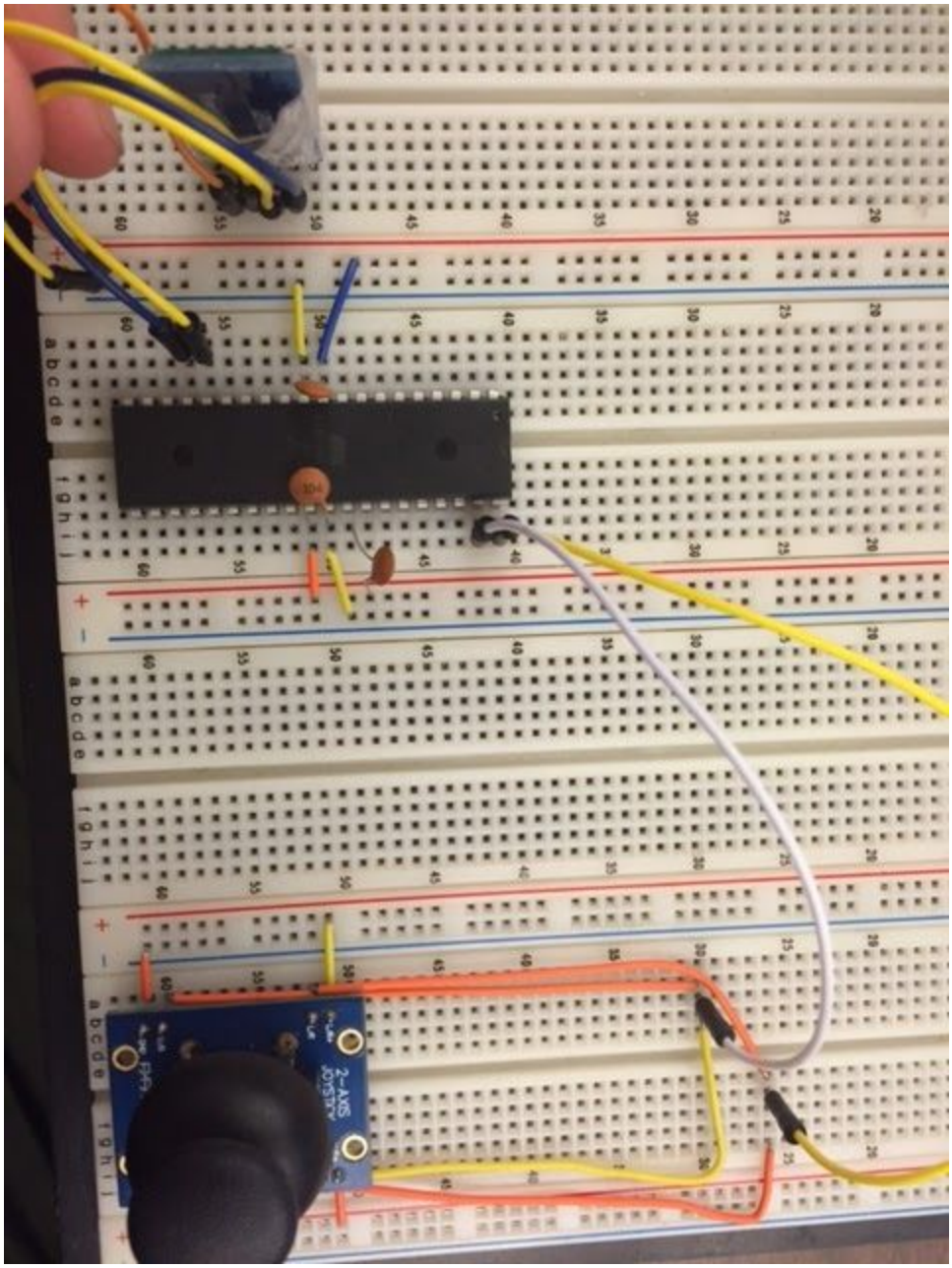

Figure 1: Controller Wiring on Breadboard

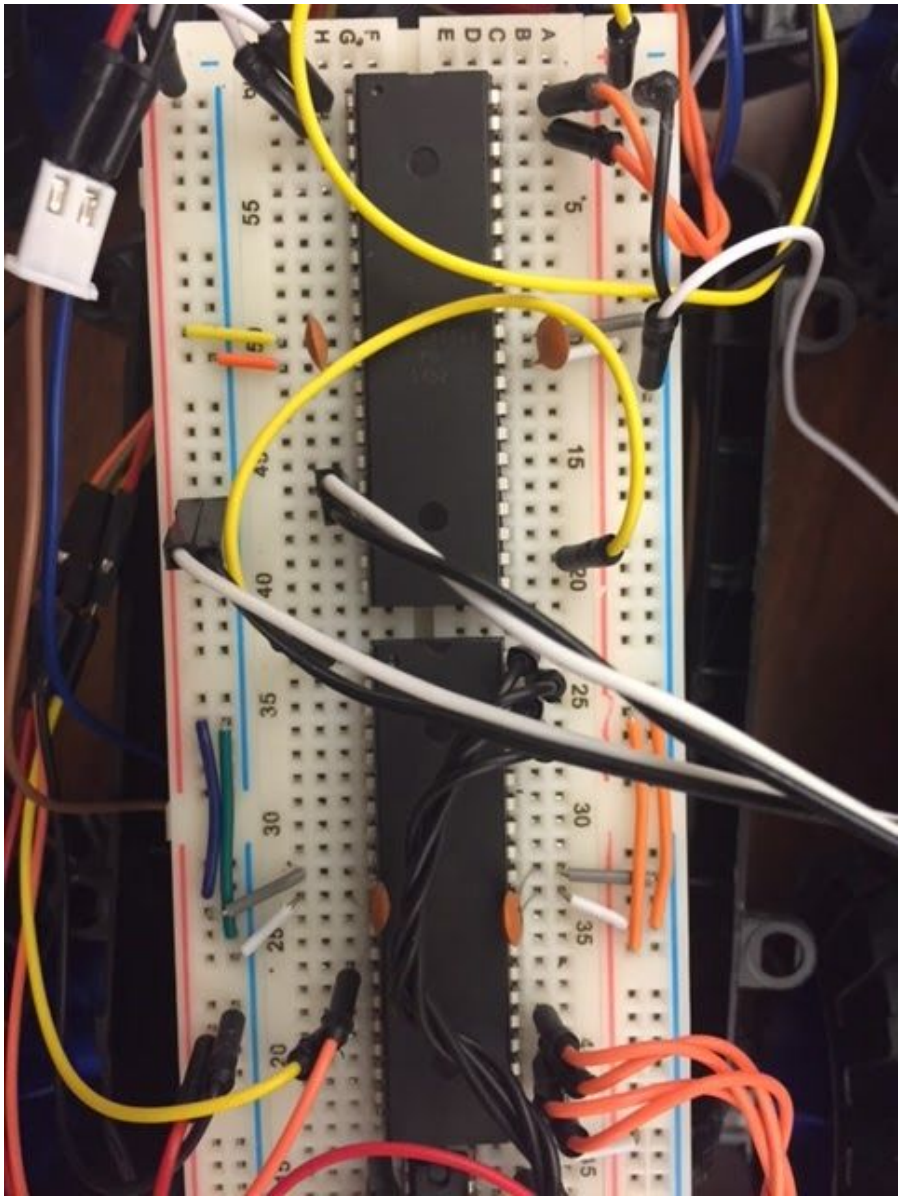

Figure 2: Receiver and Sonar/Matrix MicroController Wiring

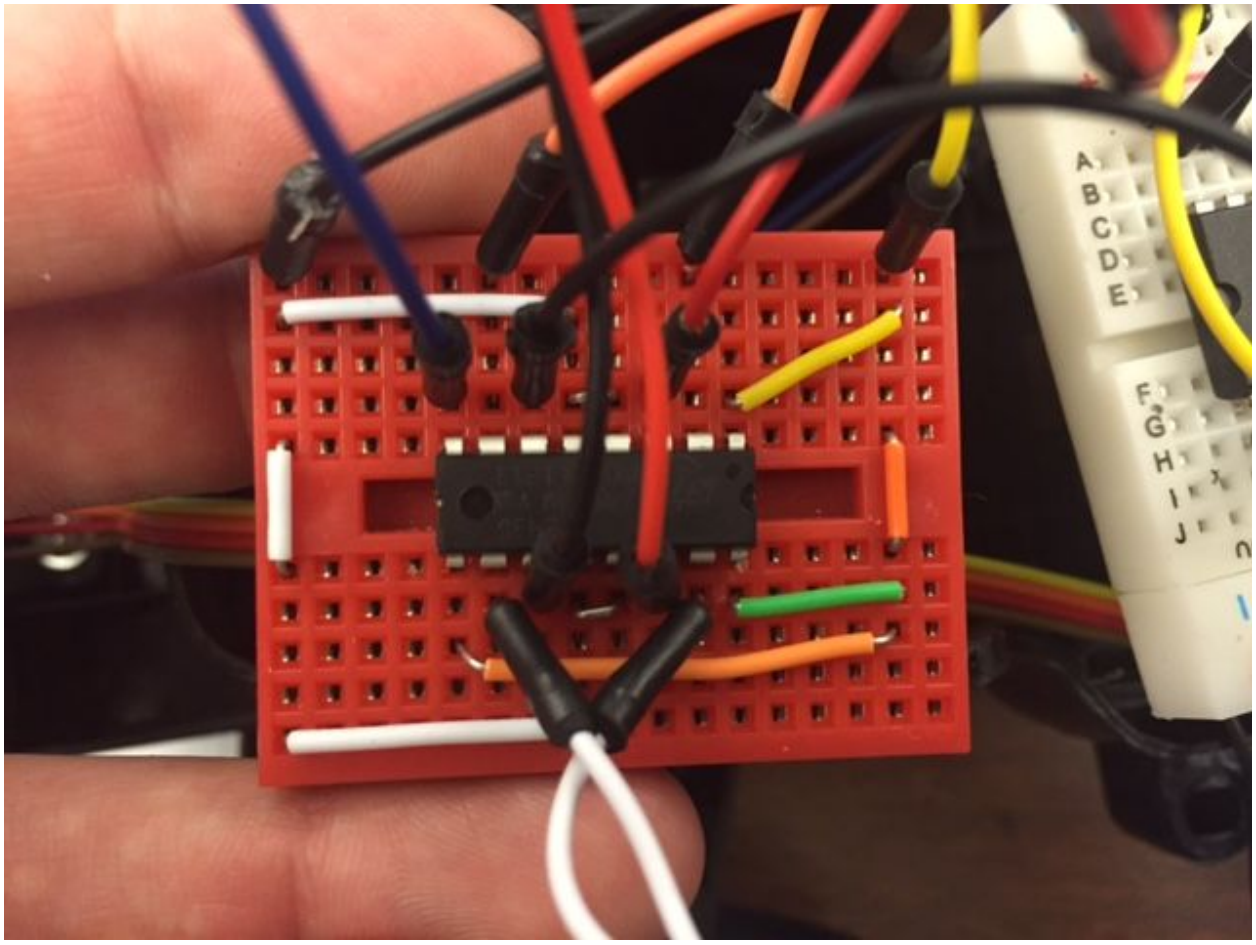

Figure 3: L293D Motor Driver Wiring

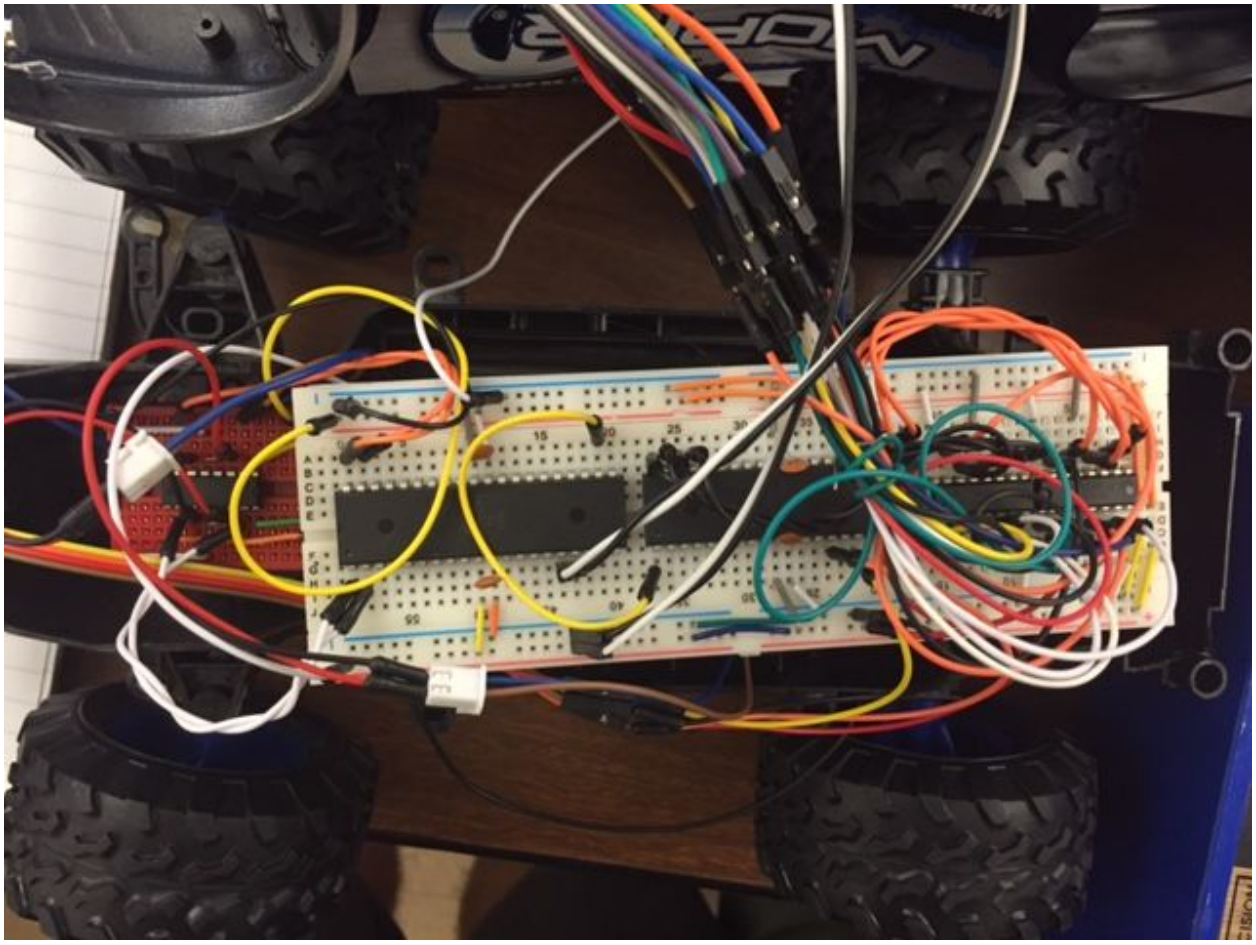

Figure 4: Overview of the entire car wiring circuit

Figure 5: Sonar Range Finder Module HC-04 Mounted on Car

Figure 6: LED matrix mounted on Car/ Bluetooth HC-05 Module.

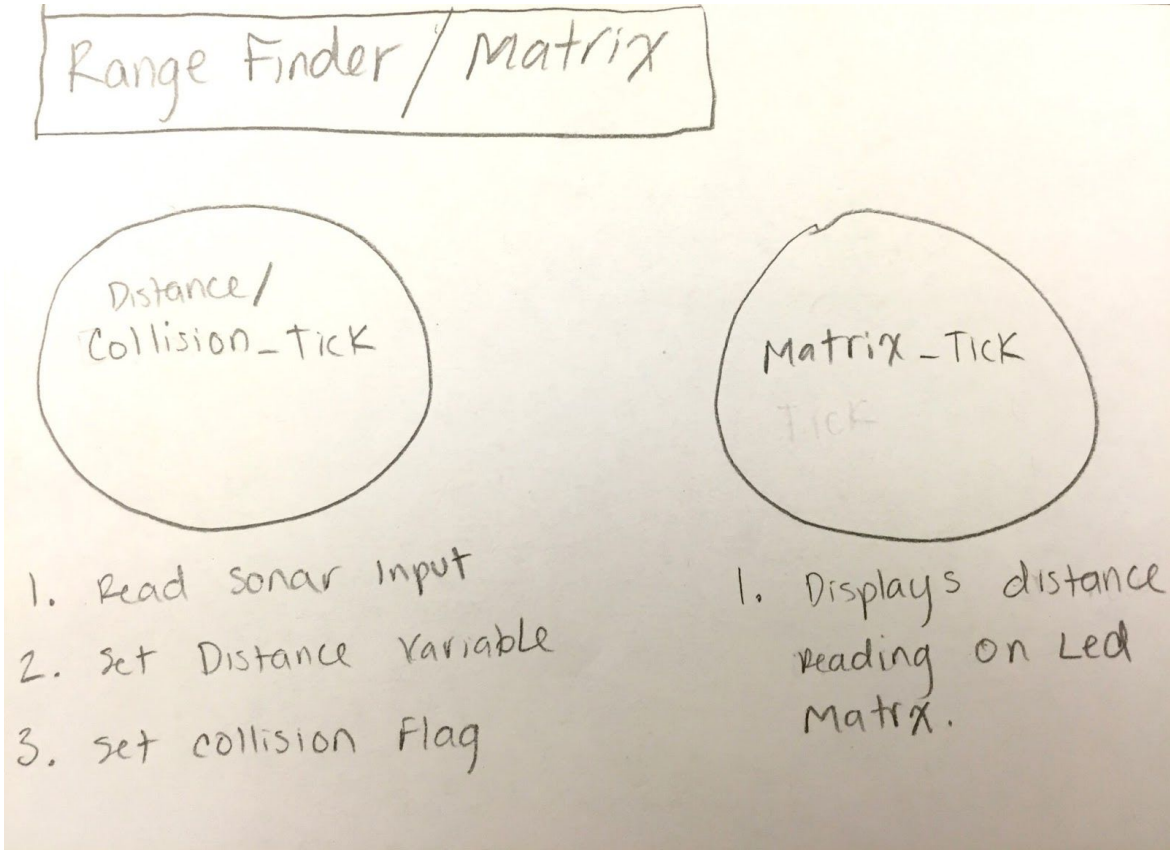

Figure 7: Range Finder/LED matrix State Machine Functions

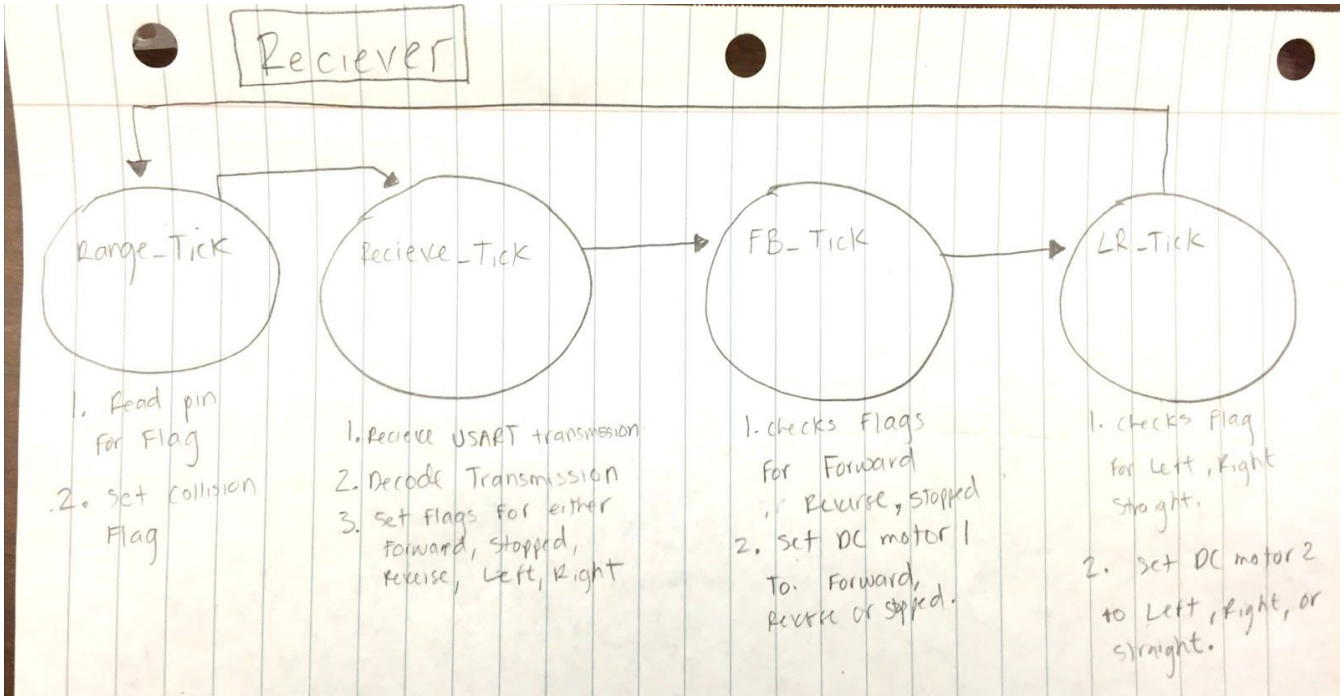

Figure 8: Receiver state machine Functions

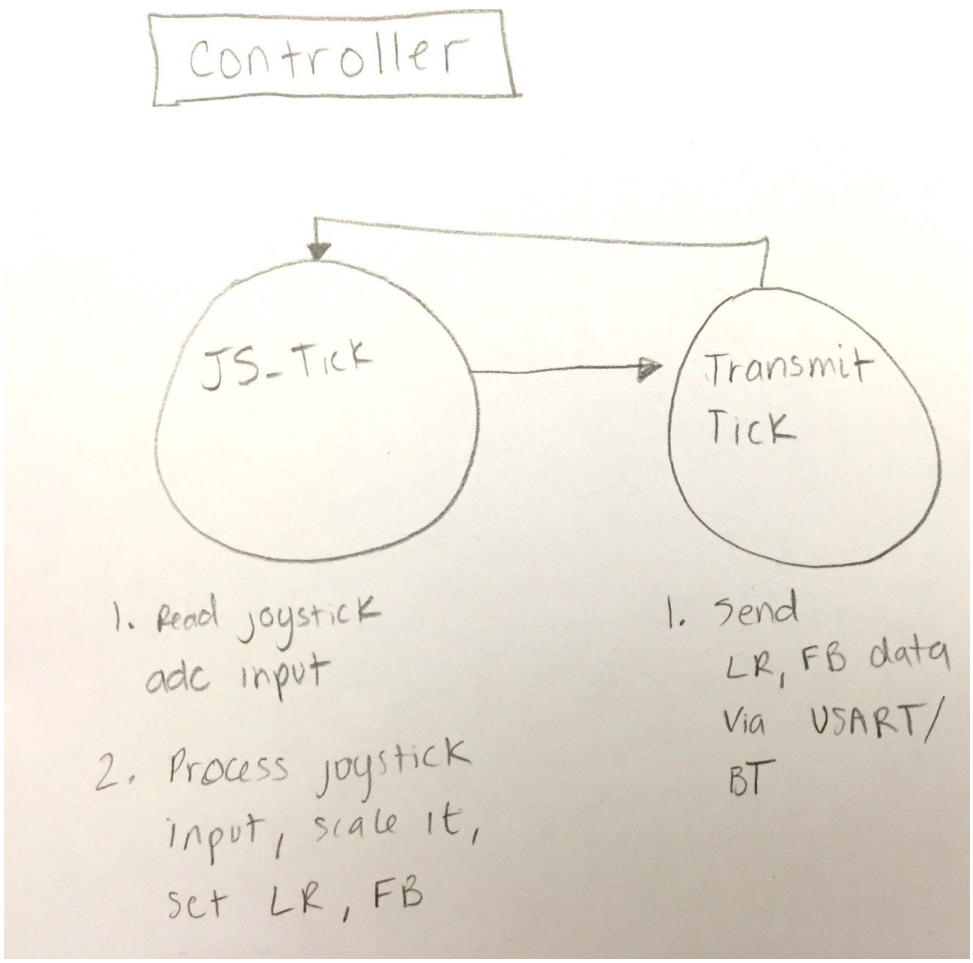

Figure 9: Controller State machine Functions

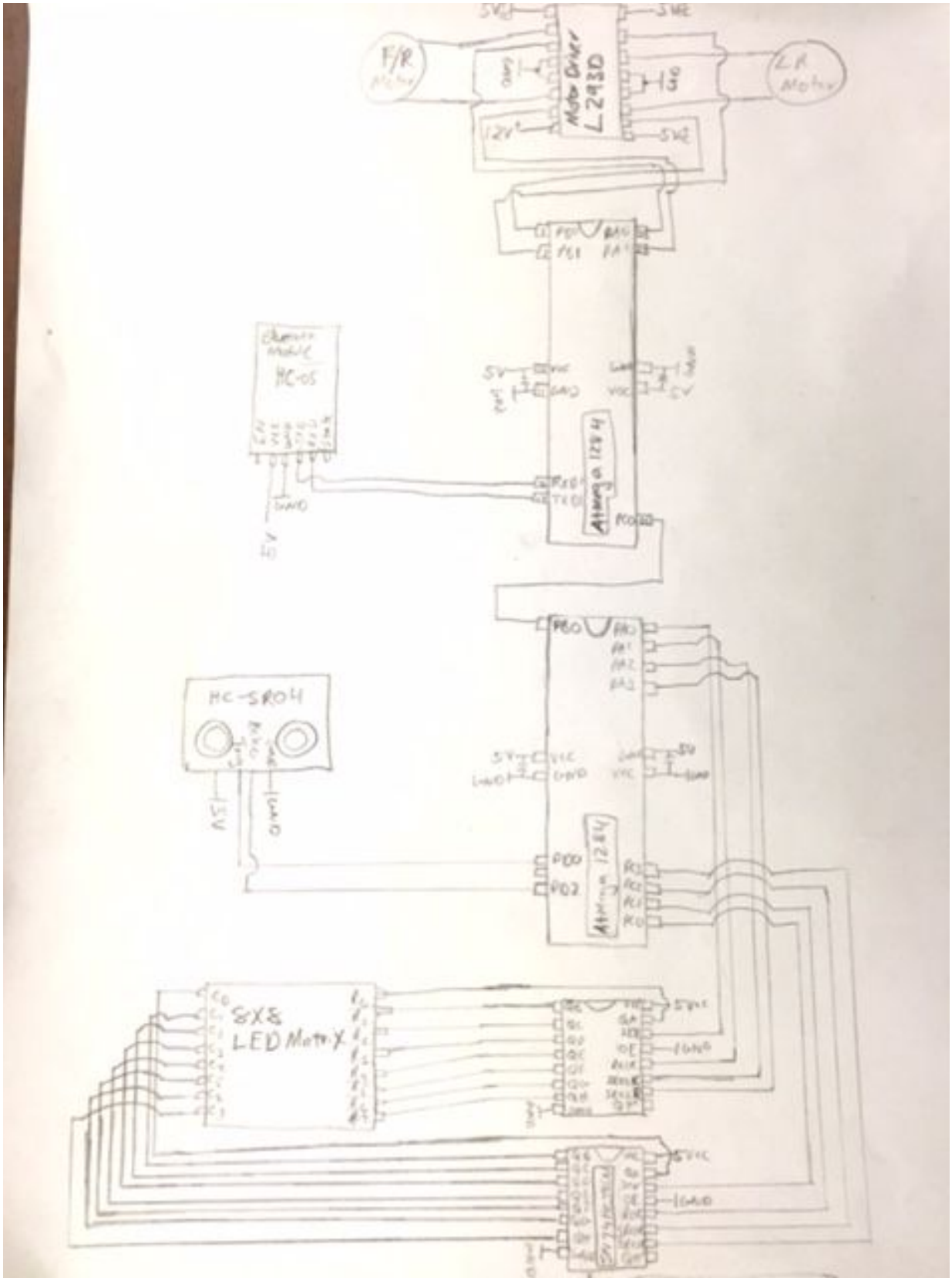

Figure 10: Wiring Diagram for Car Circuit.

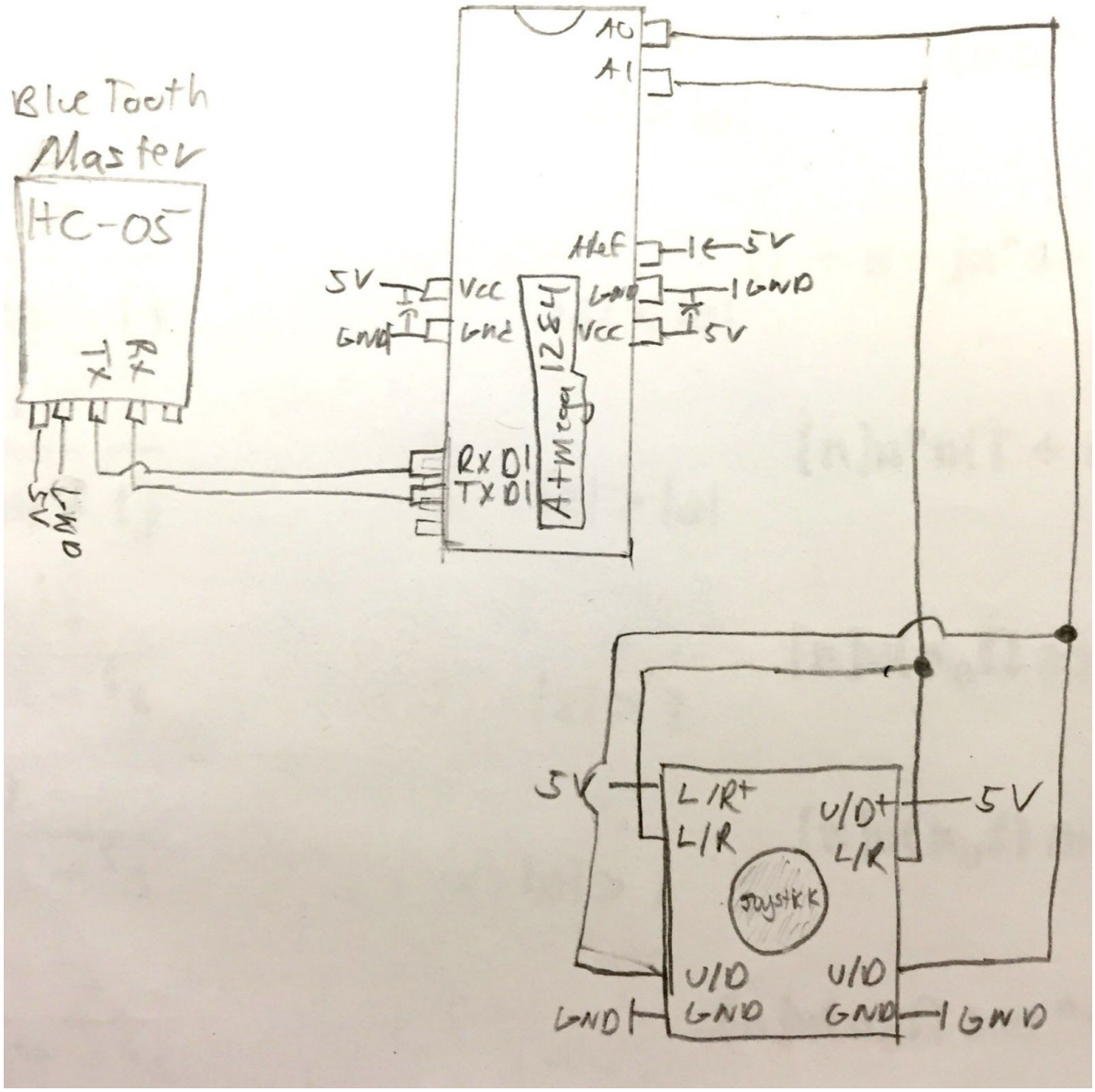

Figure 11: Wiring Diagram for Controller

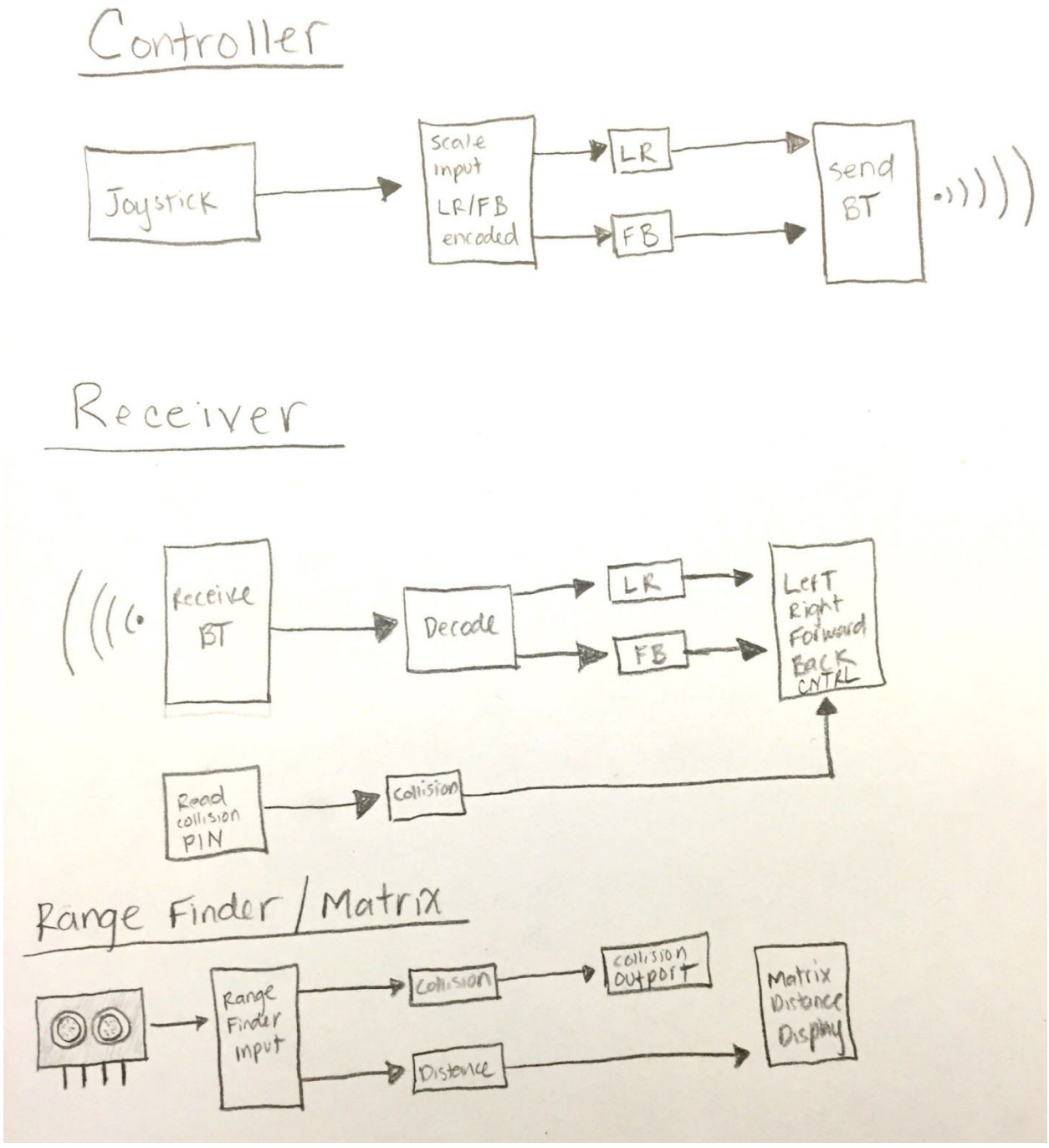

Figure 12: High Level Task Diagram