Bluetooth Joystick Controlled Car

Liping Hu

Advisor: Professor David Vallancourt, Electrical Engineering

Playing a wireless toy car is enjoyable for some people especially if they have younger kids around, but one might find it easy to get bored with the car. If one can build a wireless car themselves from scratch, and add layers of different functionalities on such as automatic obstacle detection and so on, the car would be much more satisfying and even convenient for life with the suitable car function. It is fun building a toy car and even more so watching it running on the floor.

My car mainly consisted of two parts, one part is the joystick-controlled car and the other one is the wireless communication. In order to get a working project, I decided to complete the car for starters. To have a properly running car, I bought four DC motors, Arduino Uno board, two LN298 motor driver shield, car chassis and a joystick module. To be more specific, I mounted two L298N motor drivers on the Arduino Uno board because each motor driver could drive 2 DC motors due to not enough PWM pins that it has. Where PWM (pulse width modulation) gives us varying analog outputs with digital inputs by changing the signal’s “on” time.  Therefore, I could use the motor driver to drive two DC motors which drive the wheels of my car.

My control in terms of the car is through a device called joystick.  A joystick is an input device that pivots on a base and reports its direction to the device it is controlling. I used the two inputs of the joystick, “x” and “y” to control directions.  In order for the joystick to work with the DC motors properly, I used the Serial monitor and learned that the joystick that I am using has x and y axis whose maximum values are both 1023 and minimum values are both 0 with central value 511 (See Figure 1.) But the problem was that the car had rash movements, it almost felt like the car wasn’t under my control.  Then I learned that it was because the input range (0 - 1023) from joystick didn’t match with the output range of the car (0 - 255). So, I used a function to map the inputs from joystick’s x pin to the corresponding value so that the car could respond.  Now that I got the correct input values from the joystick, it made programming much easier.

The logic behind my program: Firstly, I wrote out the code for moving the car forward and backward based on the x inputs from the joystick.  Then, in order to control the motor to go left or right, I used the y value from the joystick.

The second part is the wireless communication, I invented a lot of time on employing a chip called NRF24L01 since they are really cheap, and I was testing two NRF24L01 chips using different circuit schematics found online, but the two RF24L01 chips didn’t work properly, they communicated with each other briefly before they stopped working.  I tried putting a parallel capacitor across power source to stabilize it but it didn’t work,  I then decided to buy a more reliable chip. Therefore, I switch to the HC05 Bluetooth module. The HC05 chip is pretty straightforward, one can just connect them directly onto the Arduino Uno board according to its pinout and set the master and slave device using AT commands in the Arduino serial monitor. Then the rest is to program the devices to receive signals coming from each other.