Car Algorithm

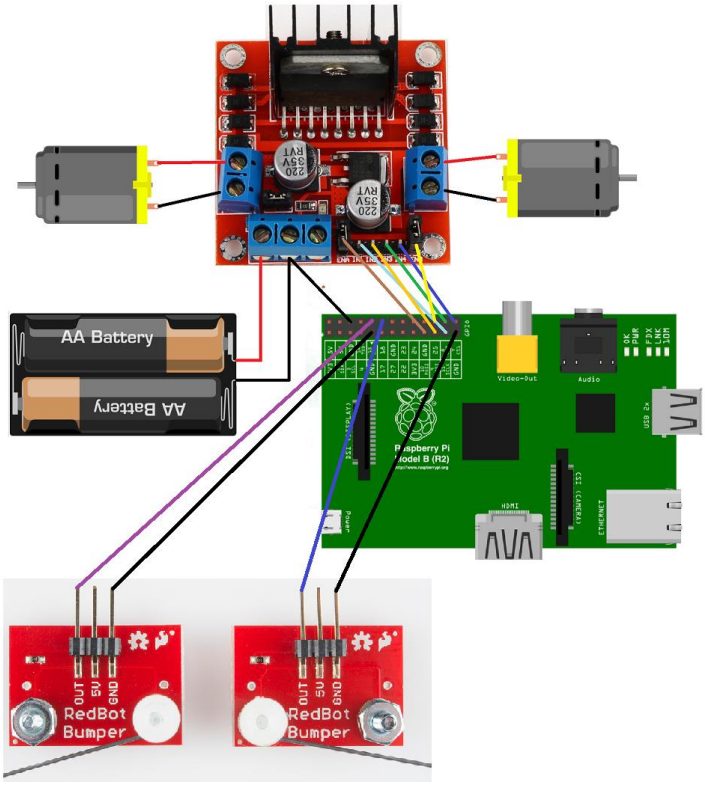
Introduction

This project builds upon the Car project by creating an algorithm, that takes inputs from two bumpers, to navigate around. The algorithm will make the car move forwards until one of its bumpers is triggered, then it will back up the car, turn away, and then move forwards. The Bumper Class in this project is very similar to Class Button and while it isn't necessary, its recommended that you check out our Button project before attempting this one.

Assembly

Parts:

- 10x Female to female jumper cables
- 1x Male to female jumper cable
- 2x RedBot Bumpers
- H-bridge L298N
- Batteries (with case)
- 2x Motors
- Raspberry Pi
- (Optional) A base
- (Optional) 2x Wheel (for the motors)
- (Optional) Stabilising wheel(s)



Build Instructions:

1. Power off the Pi completely.
2. Connect pins IN1, IN2, IN3, IN4, ENA and ENB on the H-bridge to pins 14, 13, 10, 11, 12 and 6 respectively on the Pi with female to female jumper cables.
3. Loosen the OUT 4 and OUT 3 screws (the two screws in the blue plastic on the right side of the board) and connect the motors power and

ground wires their connectors respectively and secure them in place by tightening the screws.
4. Do the same for the second motor but with OUT 1 and OUT 2.
5. Loosen the 12V and GND screws (the two left most of the screws in the blue plastic at the bottom of the board) and connect the ground pin on the Pi to the GND connector, with a male to female jumper cable.
6. Connect the power wire from the battery pack to the 12V connector and the ground wire to the GND connector and tighten both screws. A red LED on the H-bridge should now light up.
7. Connect the OUT pins on the left and right RedBot Bumpers to pins 1 and 16 respectively on the Pi.
8. Connect the GND pins on the RedBot Bumpers to ground pins on the Pi.
9. (Optional) Attach wheels to the motors and the motors to either side of the base as well as any stabilizing wheels and connect the bumpers to the front.

Exercises

Exercise 1: Fill out the main() method in class Algorithm. This method should move the car forwards until one the bumpers are triggered, it should then back up the car, turn it away from the direction of the triggered button and finally continue on forwards. RightBumper.isPressed() will return true if the right bumper is currently making contact with anything.

There are several ways of tackling this exercise, so don't worry if your solution doesn't match up exactly to how we have done it.

Notes

- Remember, Pi4J using something called WiringPi to manage GPIO pins. This means that the pin numbers do not actually correlate with what is written on the board. Use this website to convert: http://pi4j.com/pins/model-b-plus.html

- The method sleepMillisec(millisec) will allow you to pause your program for an amount of time, this can be useful if you want to move the car backwards for a given amount of time before moving it forwards again

- Once you have completed this project have ago at coming up with a move advanced algorithm.

- Remember to comment your code and that if your car isn't moving you may have forgotten to set its speed.

- An assembly guide for the Redbot Bumper can be found here: https://learn.sparkfun.com/tutorials/redbot-assembly-guide-rev-02/redbot-sensor---mechanical-bumpers?_ga=1.58372968.359569331.1458658716