SOAR LinoRobot 03/08/2017

**Testing the motors**

**L298N Motor Driver**

Blue Connector (Middle) has 3 ports

1. 12V input (Connected directly to the battery)
2. Ground (Connected to the negative part of the battery)
3. 5V output

Blue Connector (2 symmetrical Sides)

1. Positive and Negative from the motor connected here
2. Supplies 5V

Headers and their functions

1. Enable A (ENA A). PWM input that goes to the motor. With a PWM signal, the voltage input can be varied from 0V to 5V. This is used to vary the speed of the motor and turn it on.
2. Inputs

* Inputs control the direction of rotation of the wheel. Hence there are 2 inputs per motor. A signal of 1:High and 2:Low gives the forward direction and when it is reversed, the motor turns in the opposite direction (Test and adjust using Arduino if it is not like that). High-high and low-low will cause the motor to brake, whereas to supply no power to the motor, turn off the Enable pin.
* Input 1 (IN 1).
* Input 2 (IN 2)

1. Note that IN 3, IN 4, EN B can be used for another motor and hence we will only need one L298N to control 2 motors, but for simplicity’s sake, we’ll assign one L298 motor driver per motor so we won’t have to worry about heat and voltage/current concerns with powering multiple motors.
2. Note: Speed of the motor cannot be changed with a potentiometer in the circuit (Explained later)

**Initial Testing**

First test was directly connecting the +/- of the motor to the 12V lead-acid battery. This managed to turn the wheels. The problem is that this could have destroyed the encoder. It was connected for a short while. Hence, to remedy this, we will test it with an Arduino Uno.

**Testing (II) Safer way of testing the motors at max speed**

Connections

1. Connections to the battery are the same as the first test
2. The 5V output from the middle blue connector is plugged into a breadboard
3. EN A and IN 1 connected to the 5V input on the Breadboard. This sends the 5V to the motors and creates the HIGH for the direction.
4. Another jumper is connected from IN 2 to a separate row on the breadboard. Another jumper is then linked from the GND from the middle blue connector to that same row. This creates the low
5. With the direction of rotation and the input voltage, the motor turns

**Testing (III) Failed attempt to provide PWM using a potentiometer**

* A potentiometer has 3 connections to it. The one in the middle varies the voltage. One side receives maximum voltage, while the other is connected to the ground. Connect the plus to the voltage and the minus to the GND.
* The encoder is connected to where the middle is connected.
* It fails to vary the speed because it only varies pulse amplitude, not frequency, which is what we’ll need to vary speed.

**Testing (IV) Use an Arduino board with all the existing components to test the encoder (IT WORKS!)**

Arduino is used to read the output from the encoder (value changes when the speed is varied using the potentiometer. In simple terms, the potentiometer input into the Arduino changes between a value 0 to 1023 which is then scaled down to 0-255 (via the map() function) which is the max speed of the motor.

This information is transferred to the motors via a PWM signal from the Arduino board to the L298 motor driver via the enable pin to control the speed of the motor. Notably, the L298N has to be grounded to the Arduino for the enable pin to work.

void setup() {

// put your setup code here, to run once:

pinMode(11, OUTPUT);

pinMode(A5, INPUT);

pinMode(A0, INPUT);

pinMode(A1, INPUT);

Serial.begin(9600);

}

int inputA0;

int inputA1;

int spd;

void loop() {

// put your main code here, to run repeatedly:

spd = analogRead(A5);

analogWrite(11, map(spd, 0, 1023, 0, 255));

inputA0 = analogRead(A0);

inputA1 = analogRead(A1);

Serial.print(spd);

Serial.print("\t");

Serial.print(map(spd,0,1023,0,255));

Serial.print("\t A0 \t");

Serial.print(inputA0);

Serial.print("\t A1 \t");

Serial.println(inputA1);

}