Arduino – Tethered Car

Contents

[Parts List 2](#_Toc445536474)

[Instructions 6](#_Toc445536475)

[Theory 9](#_Toc445536476)

[Practical 10](#_Toc445536477)

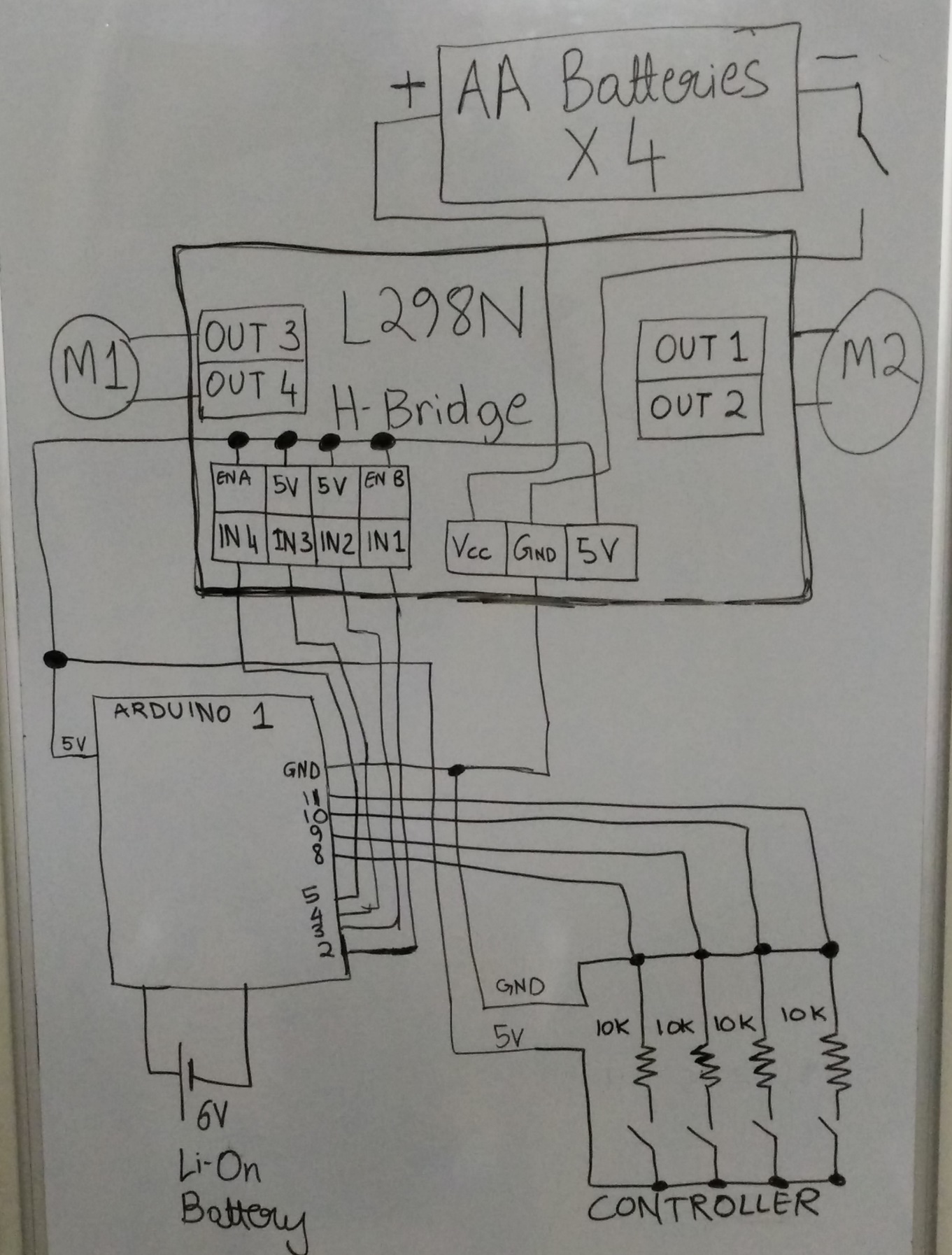
## Parts List

|  |  |
| --- | --- |
| Part | Image |
| Arduino UNO | Arduino Uno - R3.jpg |
| L298N H-Bridge (Motor Driver) | https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRCzpwz8KjIdyNwKWA-7AkMBDq3ccxHVdG8LT5jMQrIauXiByC4aQ |
| Toy Car with 2 motors | C:\Users\Rabbani\SkyDrive\George Brown\Programming\Pictures\IMG_20141202_214252.jpg |
| Batter case with 4 AA Batteries | C:\Users\Rabbani\SkyDrive\George Brown\Programming\Pictures\IMG_20141202_213504.jpg |
| Wires | C:\Users\Rabbani\SkyDrive\George Brown\Capstone\Documentation\Images\IMG_20141213_052959.jpg |
| 4 Push buttons |  |
| Vcc to Motors |  |

|  |  |
| --- | --- |
| Li-on Batter (6V and 3000mAh) with a connector pin (the connector is optional. Wires alone would suffice) | C:\Users\Rabbani\SkyDrive\George Brown\Programming\Pictures\IMG_20141202_213931.jpg |
| Breadboard x 2 | http://www.nexuscyber.com/content/images/thumbs/0000164_half-size-solderless-breadboard.jpeg |
| UTP Cable (At least 2m) | https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQ6MmwFdJNGZ3xBjagiwCourj-BHi7-0nPc2dpqv8BJK0u9T_y7 |
| Electrical Tape | https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3o4ZWZg-6DyjuNDu59mTI_F2qj9RCol4_8V4S0DOkFsrIVrj4SQ |
| 10K resistors x 4 | http://nootropicdesign.com/ve/images/10K_resistor.jpg |
| Male to Female Header Wires | http://www.mikroe.com/img/development-tools/compontents/wire-jumpers/wire_jumpers_female_to_male.jpg |

## Instructions

1. Set up the circuit as shown below:



1. Upload this code onto Arduino:

//A toy car with two DC motors is operated with an H-Bridge

//buttons to go forward, backward, left and right.

int switch1;

int switch2;

int switch3;

int switch4;

void setup() {

//setup pins

pinMode(0, OUTPUT);

pinMode(1, OUTPUT);

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

//car starts in neutral .i.e. stationary position.

neutralFR();

neutralRL();

}

void loop () {

//read digital value of switches

switch1 = digitalRead(8);

switch2 = digitalRead(9);

switch3 = digitalRead(10);

switch4 = digitalRead(11);

//if switch1 is pressed, go forward, if not, see if switch2

// is pressed. If not, remain in neutral with respects to forwards and backwards.

if (switch1 == 1) {

forward();

} else if (switch2 == 1) {

reverse();

} else {

neutralFR();

}

//if switch3 is pressed, go right, if not, see if switch4

// is pressed. If not, remain in neutral with respects to left and right.

if (switch3 == 1) {

right();

} else if (switch4 == 1) {

left();

} else {

neutralRL();

}

}

//configure H-bridhge for left

void left() {

digitalWrite(4, LOW);

digitalWrite(5, HIGH);

}

//configure H-bridhge for right

void right() {

digitalWrite(4, HIGH);

digitalWrite(5, LOW);

}

//configure H-bridhge for forward

void forward() {

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

}

//configure H-bridhge for reverse

void reverse() {

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

}

//configure H-bridhge for neutral with respects to left and right.

void neutralRL() {

digitalWrite(4, LOW);

digitalWrite(5, LOW);

}

//configure H-bridhge for neutral with respects to forward and backward.

void neutralFR() {

digitalWrite(2, LOW);

digitalWrite(3, LOW);

}

## Theory

The L298N H-Bridge is used to control the motors by much higher currents than supported by the Arduino. The Arduino sends signals as following:

* There are four enable pins (two for each motor) which must be set to high before the motor can be used.
* Input pins 1 and 2 are used to control motor 1. If input 1 and 2 are of the same value (.i.e. both high or both low) then the motor will not move. If one is high and the other is low, the motor will rotate. If we switch the values, the motor will rotate in the opposite direction.
* The UTP cable is used in order to minimize EMI.
* The push buttons are used to send signals to the Arduino of forward, backward, left and right. Based on the signal, the Arduino changes the values of the inputs.

## Practical

This is the entire project:



