

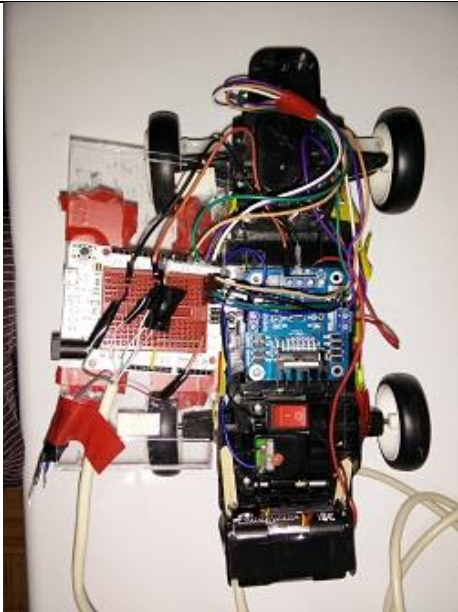


Arduino – Tethered Car

Contents

Parts List.....	2
Instructions	6
Theory	9
Practical.....	10

Parts List

Part	Image
Arduino UNO	 A blue Arduino UNO R3 microcontroller board. It features a USB Type-B port, a DC power jack, a reset button, and a large integrated circuit (ATmega328P). The board is populated with various electronic components like resistors, capacitors, and a crystal oscillator.
L298N H-Bridge (Motor Driver)	 An L298N H-Bridge motor driver module. It is a blue printed circuit board with a large black heat sink, several green terminal blocks for motor and power connections, and a red potentiometer. The module is branded with the SEED STUDIO logo.
Toy Car with 2 motors	 A photograph of a completed toy car assembly. It shows a black chassis with four wheels. An Arduino Uno is connected to an L298N motor driver via jumper wires. The driver is connected to two DC motors. A red battery pack is attached to the side of the car. The entire setup is mounted on a white surface.

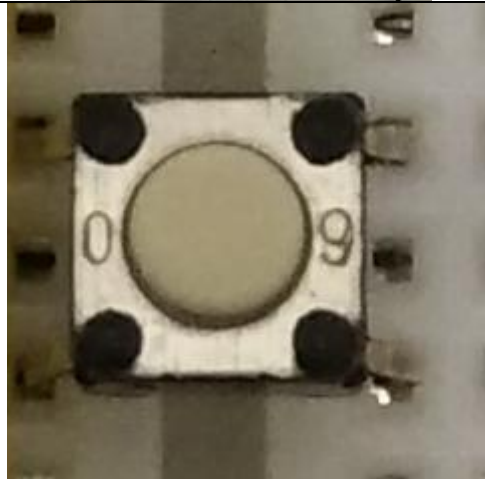
Batter case with 4 AA Batteries



Wires



4 Push buttons



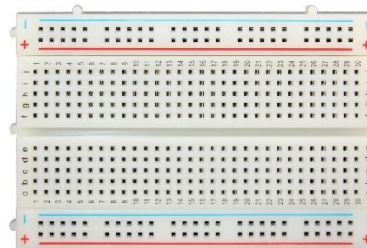
Vcc to Motors



Li-on Batter (6V and 3000mAh) with a connector pin (the connector is optional. Wires alone would suffice)



Breadboard x 2



UTP Cable (At least 2m)



Electrical Tape



10K resistors x 4

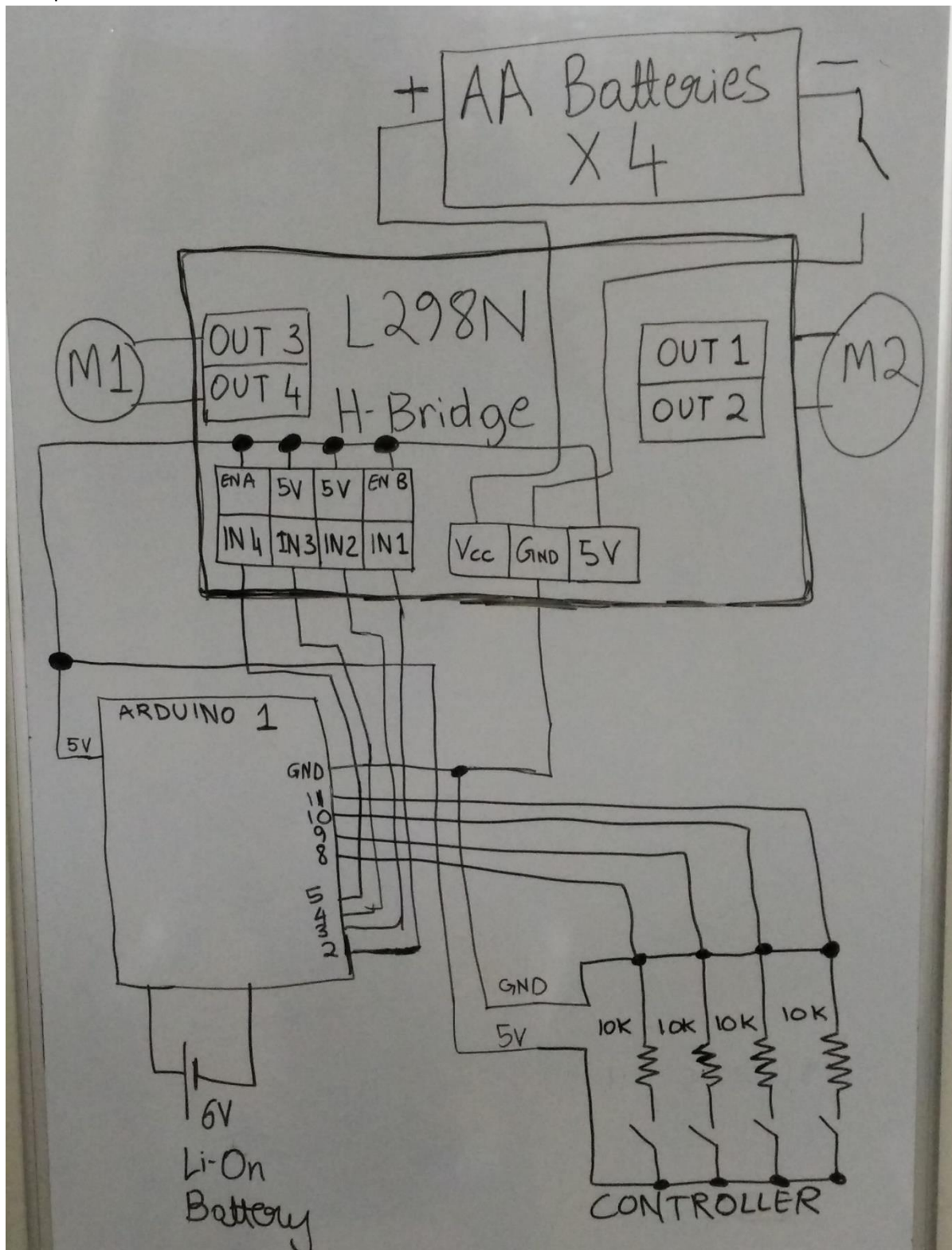


Male to Female Header Wires



Instructions

1. Set up the circuit as shown below:



2. 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:



