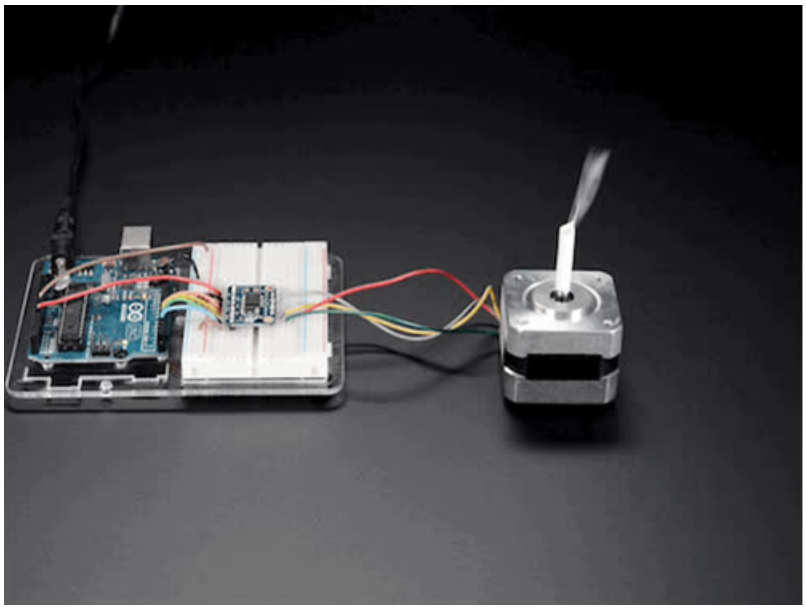
adafruit learning system

Adafruit TB6612 1.2A DC/Stepper Motor Driver Breakout Board

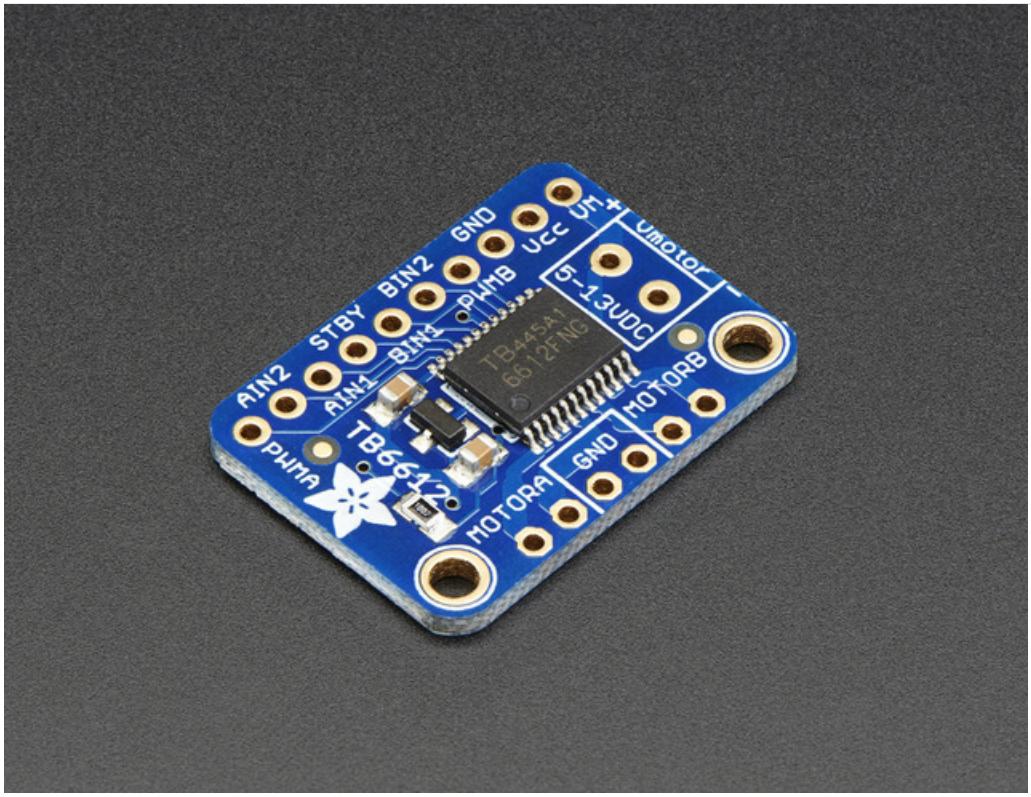
Created by lady ada



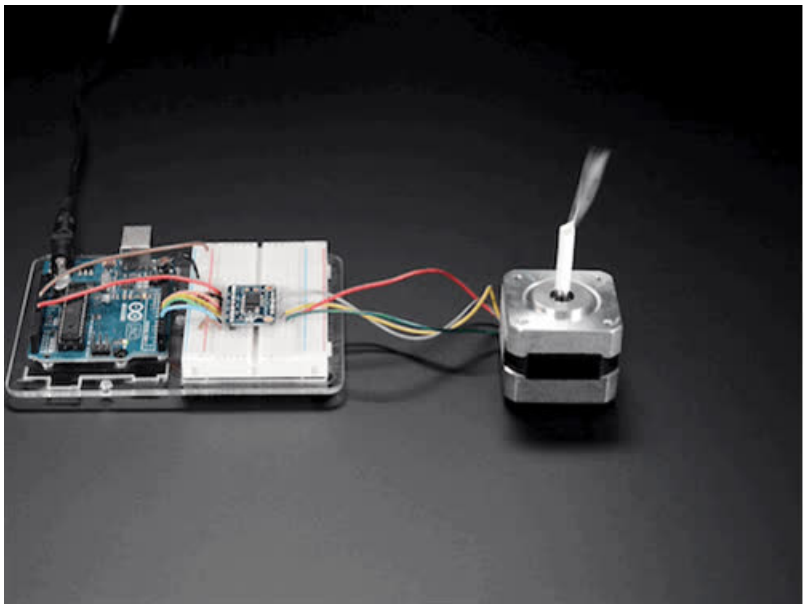
Last updated on 2018-08-22 03:47:23 PM UTC

Guide Contents

Overview

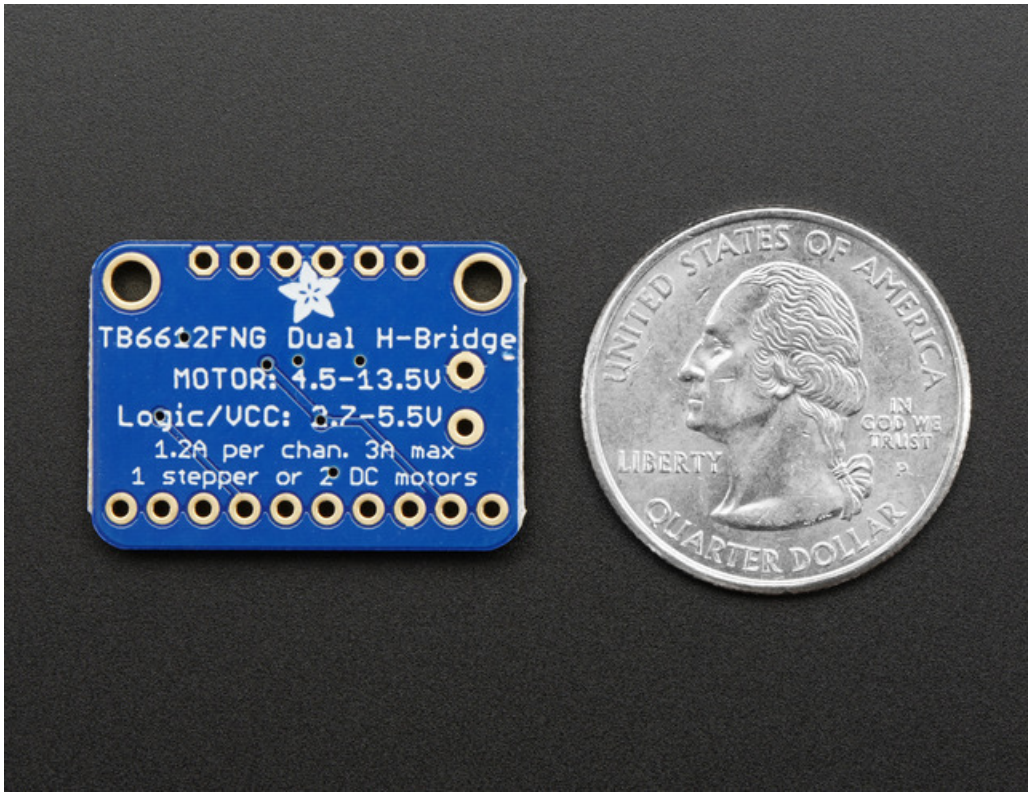


Fire four solenoids, spin two DC motors or step one bi-polar or uni-polar stepper with 1.2A per channel using the TB6612. These are perhaps better known as "the drivers in our assembled Adafruit Motorshield (http://adafru.it/1438) or Motor HAT (https://adafru.it/eRq)" We really like these dual H-bridges, so if you want to control motors without a shield or HAT these are easy to include on any solderless breadboard or perma-proto.

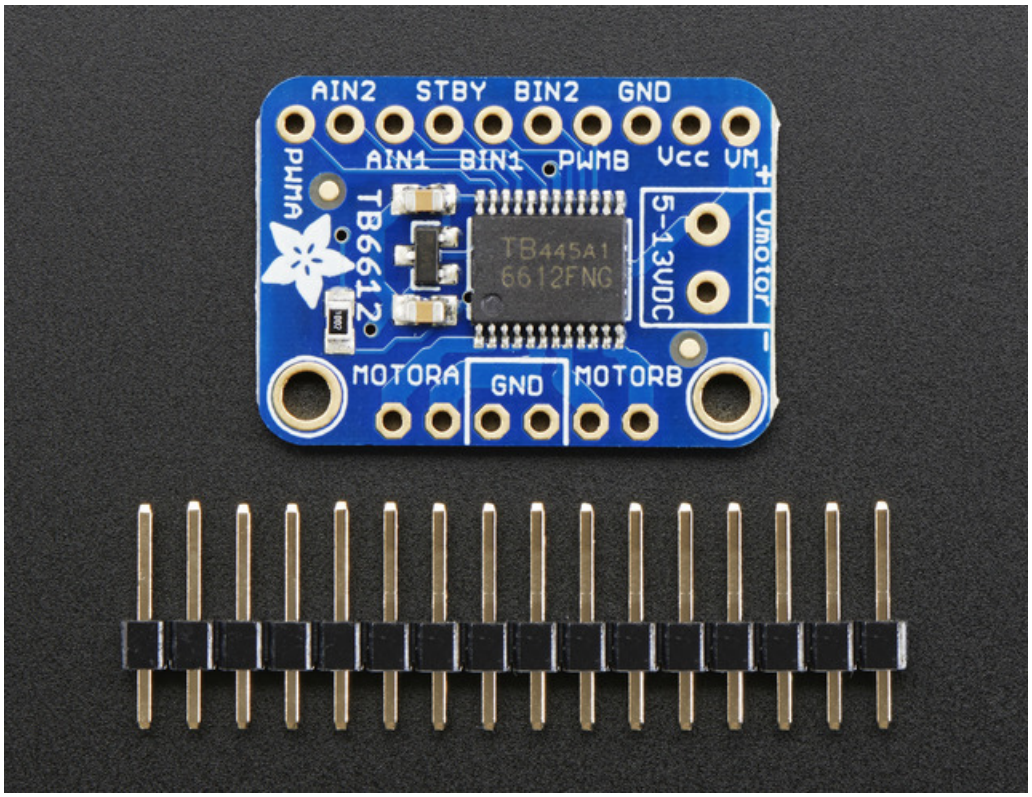


We solder on TB6612 onto a breakout board for you here, with a polarity protection FET on the motor voltage input and a pullup on the "standby" enable pin. Each breakout chip contains two full H-bridges (four half H-bridges). That means you can drive four solenoids, two DC motors bi-directionally, or one stepper motor. Just make sure they're

good for 1.2 Amp or less of current, since that's the limit of this chip. They do handle a peak of 3A but that's just for a short amount of time, about 20 milliseconds. What we like most about this particular driver is that it comes with built in kick-back diodes internally so you dont have to worry about the inductive kick damaging your project or driver!

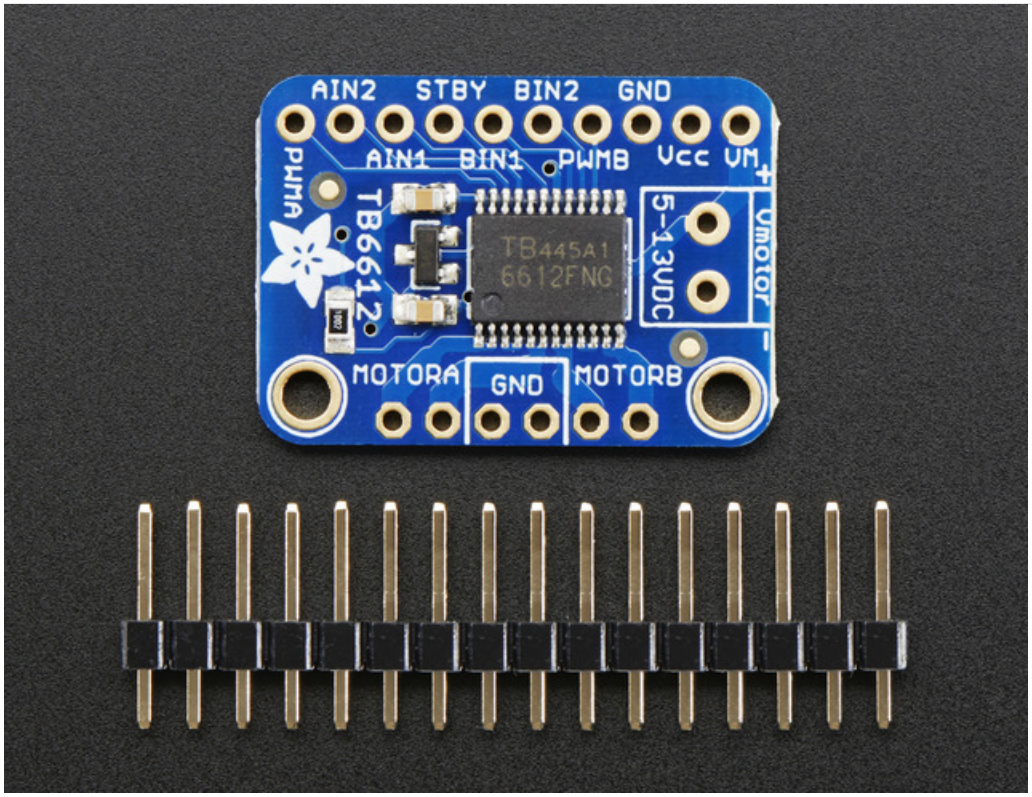


There's two digital inputs per H-bridge (one for each half of the bridge) as well as a PWM input per driver so you can control motor speed. Runs at 2.7V-5V logic. The motor voltage is separate from the logic voltage. Good for motor voltages from 4.5V up to 13.5V! This wont work well for 3V motors.



Comes as one assembled and tested breakout plus a small strip of header. You'll need to do some light soldering to attach the header onto the breakout PCB. Arduino, motors, and power supply not included.

Pinouts



This motor driver is a fairly simple breakout of the TB6612 motor chip, so do check out the datasheet for the TB6612 for any details you need about pin voltage limits, capacitance, etc! (https://adafru.it/emK)

Power Pins

- **Vmotor** - This is the voltage for the motors, not for the logic level. Keep this voltage between 4.5V and 13.5V. This power supply will get noisy so if you have a system with analog readings or RF other noise-sensitive parts, you may need to keep the power supplies seperate (or filtered!)
- **Vcc** - this is the voltage for the logic levels. Set to the voltage logic you'll be using on your microcontroller. E.g. for Arduinos, 5V is probably what you want. Can be 2.7V to 5.5V so good for 3V or 5V logic
- **GND** - This is the shared logic and motor ground. All grounds are connected

Signal in Pins

These are all 'Vcc logic level' inputs

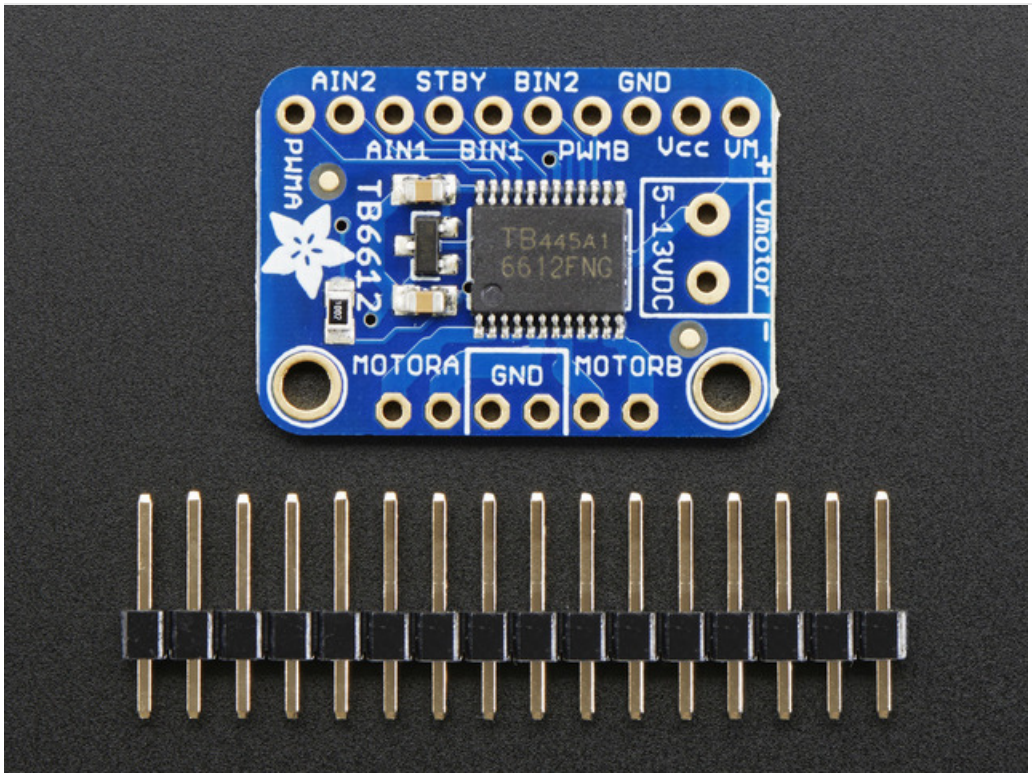
- **INA1, INA2** - these are the two inputs to the Motor A H-bridges
- **PWMA** - this is the PWM input for the Motor A H-bridges, if you dont need PWM control, connect this to logic high.
- **INB1, INB2** - these are the two inputs to the Motor B H-bridges
- **PWMB** - this is the PWM input for the Motor B H-bridges, if you dont need PWM control, connect this to logic high.
- **STBY** - this is the standby pin for quickly disabling both motors, pulled up to Vcc thru a 10K resistor. Connect to ground to disable.

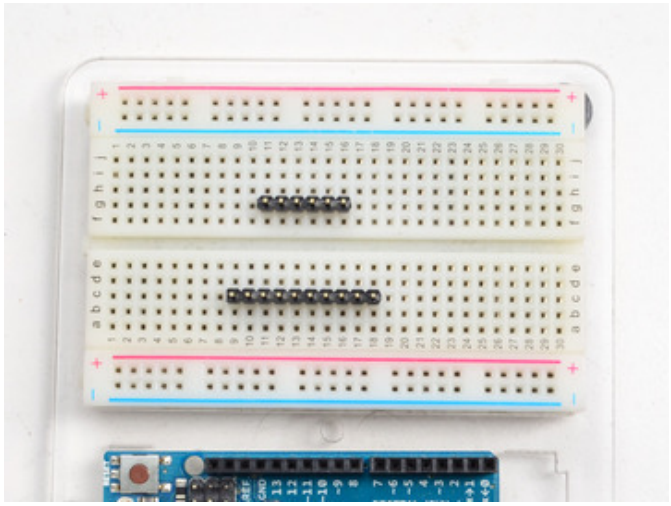
Motor Out Pins

These are 'Vmotor level' power outputs

- **Motor A** - these are the two outputs for motor A, controlled by INA1, INA2 and PWMA
- **Motor B** - these are the two outputs for motor B, controlled by INB1, INB2 and PWMB

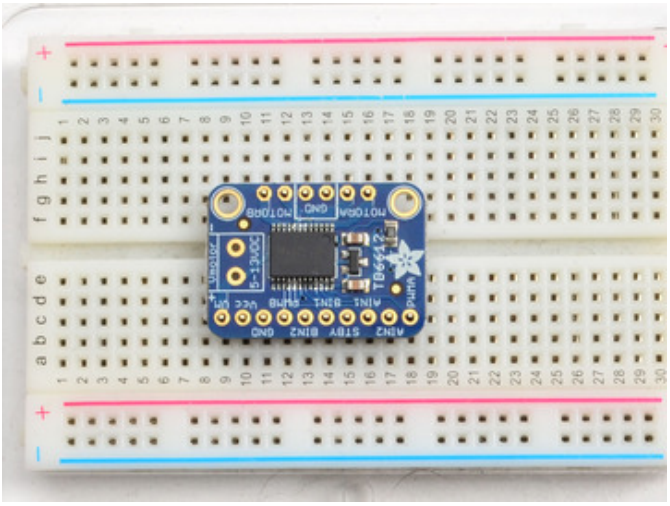
Assembly





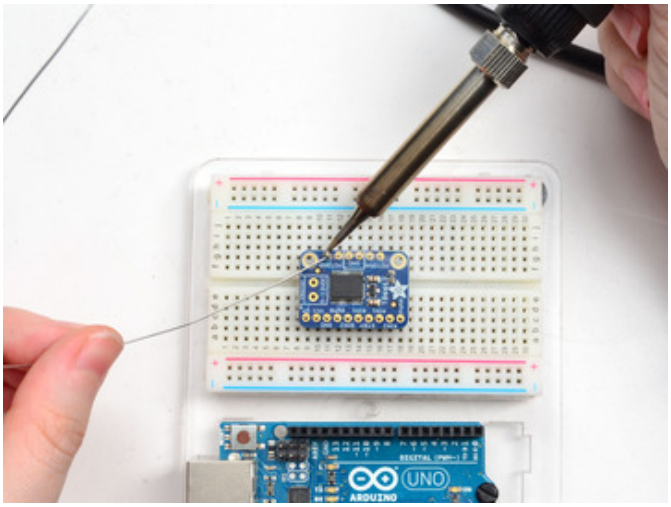
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



Add the breakout board:

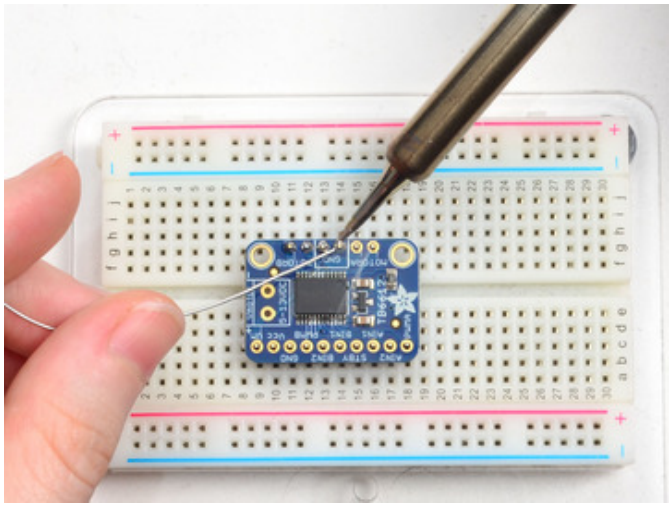
Place the breakout board over the pins so that the short pins poke through the breakout pads

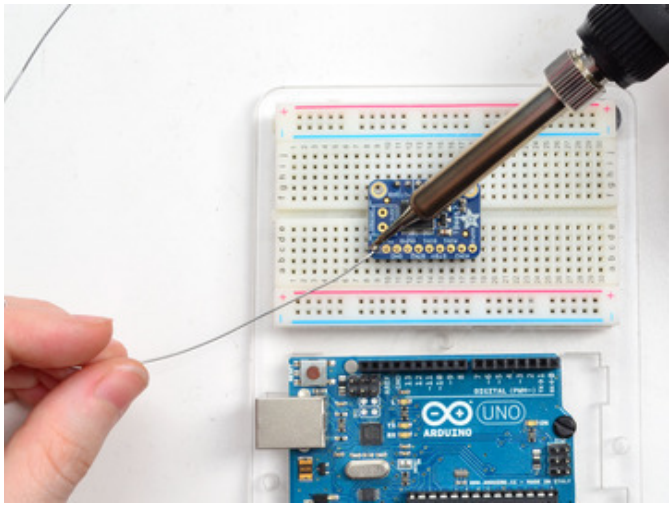


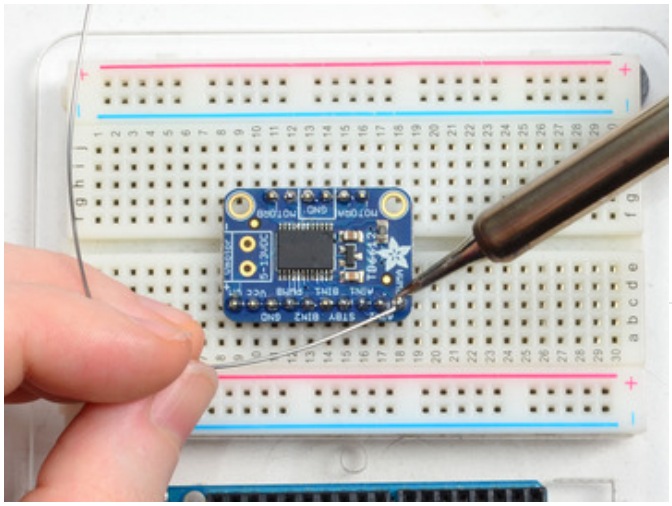
And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering (https://adafru.it/aTk).

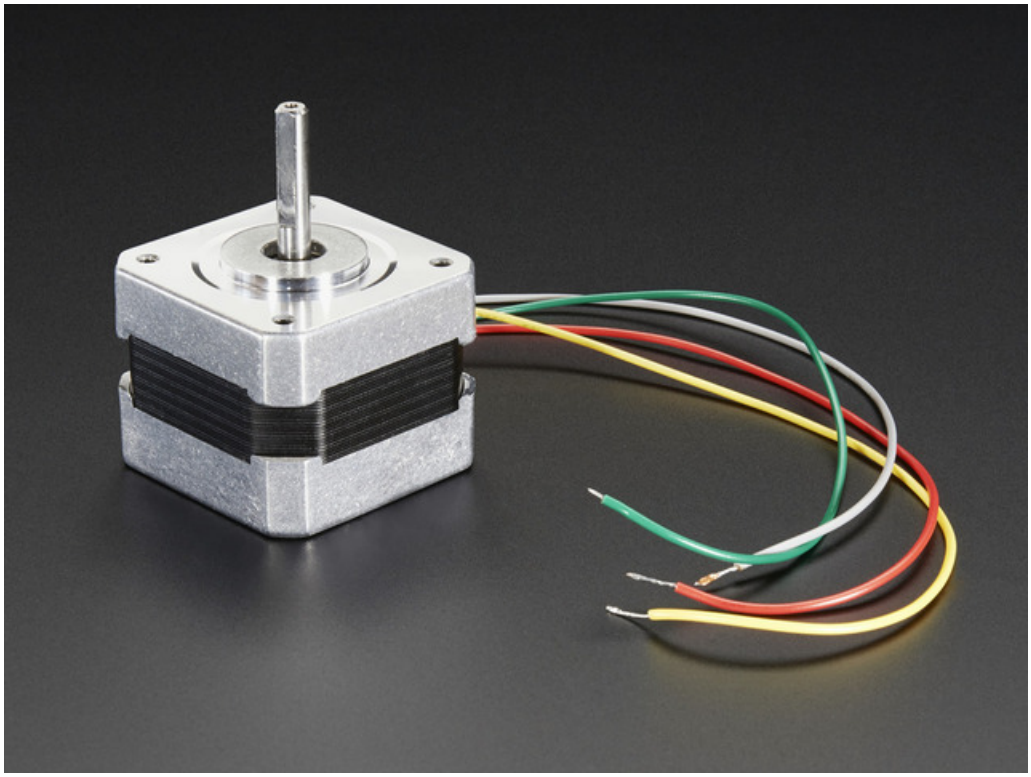



UNO
ARDUINO



Using Stepper Motors

In this example we'll wire up and use a bi-polar stepper motor with recommended 12V motor voltage, and 200 steps per rotation.



Wiring

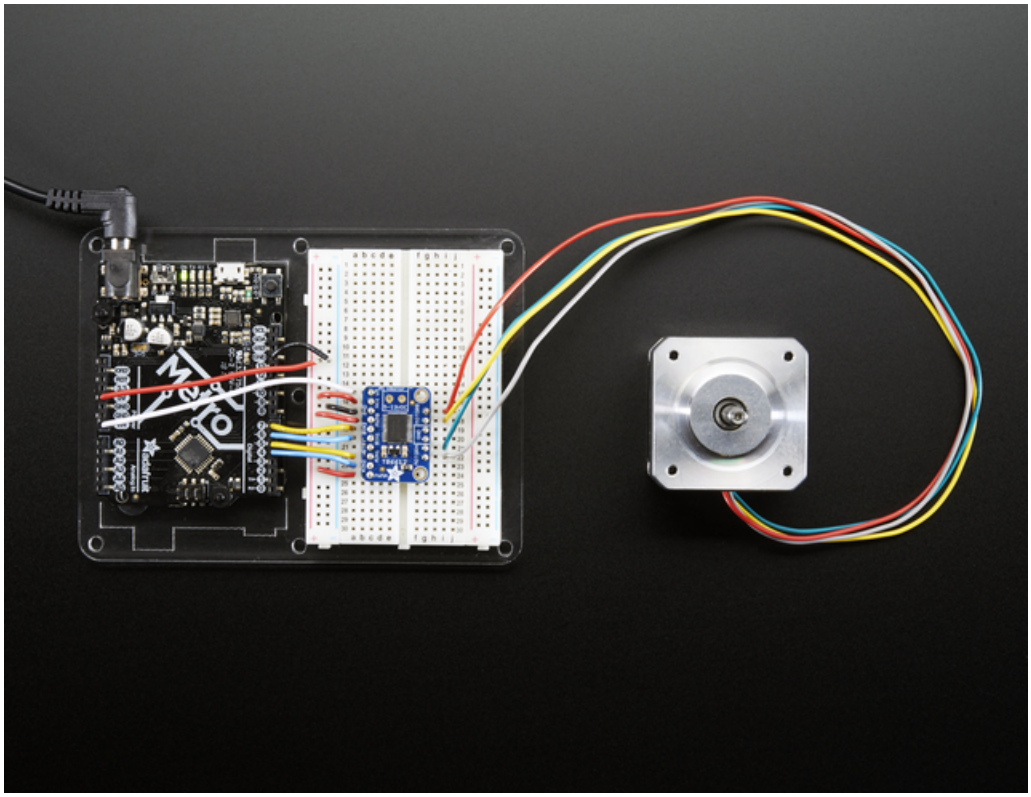
We'll wire it to a Metro, but you can use any microcontroller you like!

Connect:

- **Vmotor** to 12V (red wire)
- **Vcc** to 5V (orange wire)
- **GND** to ground
- **AIN2** to Digital 4
- **AIN1** to Digital 5
- **BIN1** to Digital 6
- **BIN2** to Digital 7
- **PWMA** and **PWMB** to Vcc (orange wire)

Then hook one stepper motor coil to **Motor A** (red and yellow) and the second coil to **Motor B** (green and gray/brown). If you have another motor, you'll need to experiment a little to figure out which wires are which coil. Check any documentation you have! You can use a multimeter to measure between wires, the ones with a small resistance between them are a pair to a coil, for example. If the motor is vibrating but not spinning, check all wires are connected and try flipping around a pair or rechecking the wire pairs.

If you have a unipolar motor, there will be a 5th or 6th wire that is the 'common' wire. Connect these wires to the GND pins in between the Motor A and B outputs on the breakout.



Software

We'll use the built-in Arduino Stepper library (https://adafru.it/eRw), but you can manually toggle the AIN1/AIN2/BIN1/BIN2 pins with your own favorite microcontroller setup

```
#include <Stepper.h>

// change this to the number of steps on your motor
#define STEPS 200

// create an instance of the stepper class, specifying
// the number of steps of the motor and the pins it's
// attached to
Stepper stepper(STEPS, 4, 5, 6, 7);


void setup()
{
  Serial.begin(9600);
  Serial.println("Stepper test!");
  // set the speed of the motor to 30 RPMs
  stepper.setSpeed(60);
}

void loop()
{
  Serial.println("Forward");
  stepper.step(STEPS);
  Serial.println("Backward");
  stepper.step(-STEPS);
}
```

Basically after you make the **Stepper** object with the 4 control pins, you can set the rotational speed (in RPM) with **setSpeed(*rpm*)** and then step forward or backwards with **.step(*steps*)** where *steps* is positive for 'forward' and negative for 'backward'

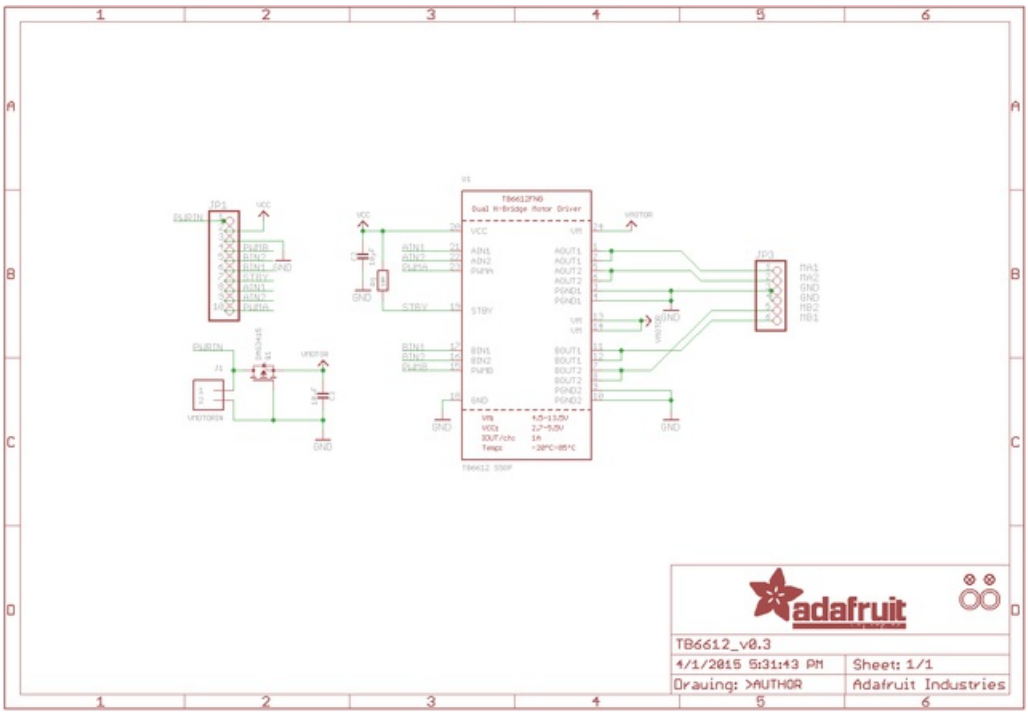
For more details, check out the Stepper library (https://adafru.it/eRw)

Downloads

Files

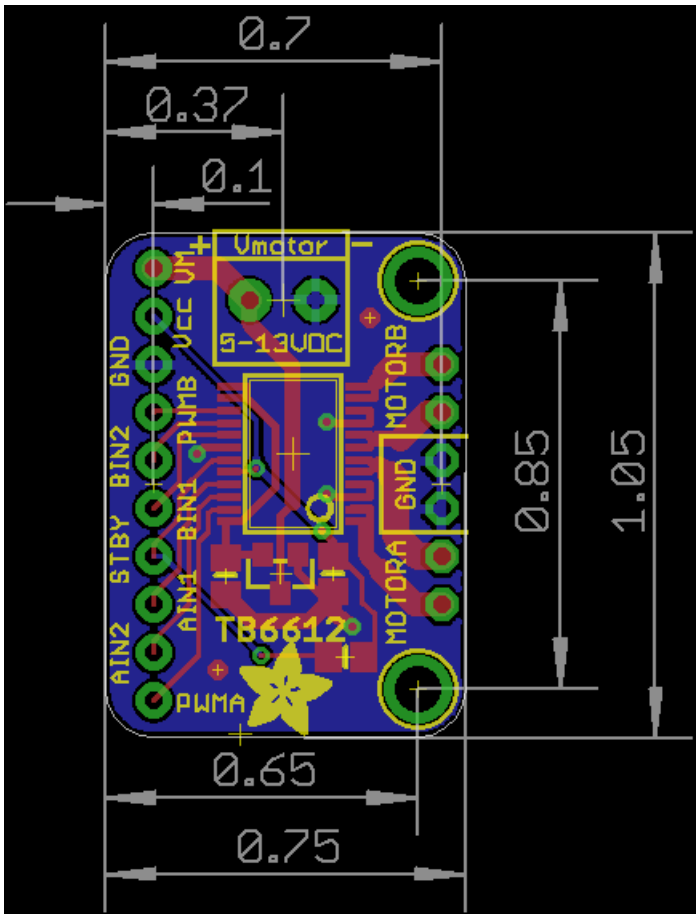
- This motor driver is a fairly simple breakout of the TB6612 motor chip, so do check out the datasheet for the TB6612 for any details you need about pin voltage limits, capacitance, etc! (https://adafru.it/emK)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/aP3)
- EagleCAD PCB files (https://adafru.it/rF1)

Schematic



Fabrication print

Dimensions in inches


0.7
0.37
0.1
Vmotor
5-13VDC
VM
VCC
GND
PWMB
BIN2
BIN1
STBY
AIN1
AIN2
PWMA
TB6612
MOTORB
GND
MOTORA
0.85
1.05
0.65
0.75