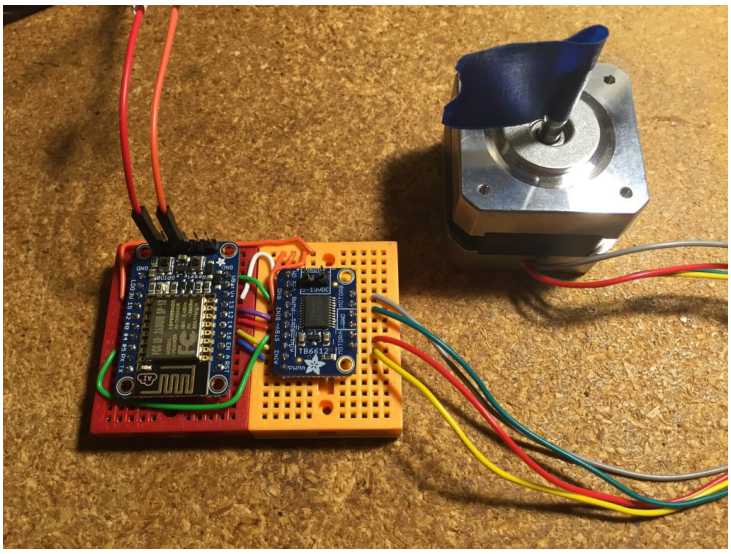
Running a Stepper Motor from an Adafruit HUZZAH ESP8266

Version 1.1 – 2015.10.01 R. Grokett

Overview

Surprisingly, you can run a powerful stepper motor (via motor controller) from the Adafruit Huzzah ESP8266 WiFi module and thus control the stepper via browser or web service, turning your stepper into an Internet of Things (IoT) device. This makes for a compact and low-cost remotely controlled motor, good for robots, mechanical arms and more.



Requirements

I used the following Adafruit components as they support the 12V and 3.3V requirements of the Stepper Motor and ESP8266 right out of the box. Otherwise, you would need to add regulators and related components to make the voltages compatible.

Hardware

- Adafruit HUZZAH ESP8266 Breakout https://www.adafruit.com/products/2471
- Adafruit TB6612 1.2A DC/Stepper Motor Driver Board https://www.adafruit.com/products/2448
- Stepper motor NEMA-17 size 200 step, 12V 350mA https://www.adafruit.com/products/324 or equiv
- 12 VDC 1000mA power supply https://www.adafruit.com/products/798 or equiv
- FTDI or USB console cable https://www.adafruit.com/products/954 or equiv
- Breadboard, wire, etc.

Software

- Arduino IDE with ESP8266 extension package installed *(see Initial Setup below)*
- Download the ESP8266_Stepper web server app from GitHub
  https://github.com/rgrokett/ESP8266_stepper/

Important Initial Setup of ESP8266

Before beginning the modification, you should become familiar with the Adafruit HUZZAH board and programming it using the Arduino IDE. The best way is to use the excellent Adafruit tutorial:

https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/overview

You must be able to program your ESP8266 and connect wirelessly to it via browser.
Once completed, THEN continue below.

Software

First you should program and test the ESP8266 before adding its hardware wiring.

1. **STOP**! Be sure you have already completed the preliminary software setup of the Arduino IDE and tested the ESP8266 with your WiFi network as described in the Initial Setup above!

2. Ok, download the ESP8266_stepper software from GitHub
   (https://github.com/rgrokett/ESP8266_stepper/)

3. Copy the **StepperWebServer.ino** program to your Arduino library area.

4. Using Arduino IDE, edit the StepperWebServer.ino and insert your SSID and PASSWORD into the appropriate places.

5. Compile and Upload the program using the FTDI or USB console cable just like shown in the Adafruit tutorial.

6. When the program finishes loading, open a Serial Monitor, set to 115,200 baud, and press the ESP8266 RESET button to restart the program running.

7. It should display the IP address once connected to your Wifi.
   Also, the onboard Red LED should blink 4 times signifying it's successfully connected.

8. Use a browser to go to http://{your_ip_addr}/stepper/stop
   This should respond back with "OK: MOTORS OFF" in browser
   Go to http://{your_ip_addr}/stepper/rpm?1 and see "OK: RPM = 1"
   Go to http://{your_ip_addr}/stepper/steps?10 and see "OK: STEPS = 10"
   Go to http://{your_ip_addr}/stepper/start and see the onboard LED blink once

9. Unplug your ESP8266 and remove the FTDI/USB cable. It's now programmed.

Wiring

1. Connect up the Stepper Motor to the TB6612 Driver board:

   - Hook one stepper motor coil to **Motor A** (red and yellow)
   - Hook the second coil to **Motor B** (green and gray/brown).

2. Next, wire up the Huzzah ESP8266 board to the TB6612 Driver board:

TB6612 Driver	HUZZAH ESP8266	Wire color in photo
VM	V+ (12V)	white
Vcc	3V	orange wires
GND	GND	green
AIN2	GPIO 16	blue
AIN1	GPIO 14	yellow
BIN1	GPIO 12	red
BIN2	GPIO 13	violet
PWMA and PWMB	Vcc	orange wires
STBY	GPIO 5	green

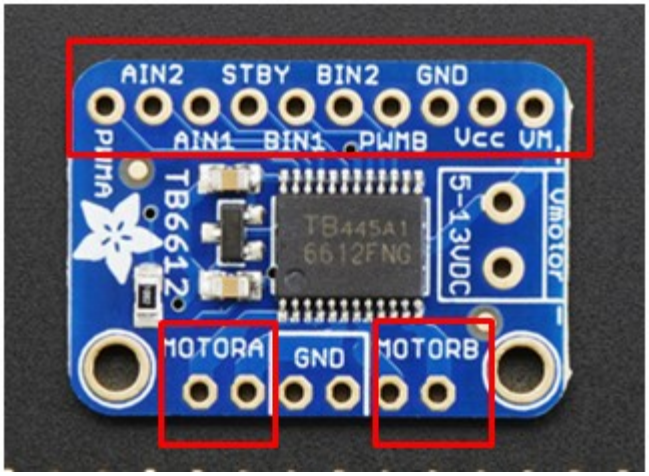
3. Connect up the 12v DC power supply to GND and V+ pins on the side of the Huzzah board (same side that the FTDI USB programming cable connected to in the software load, above.)

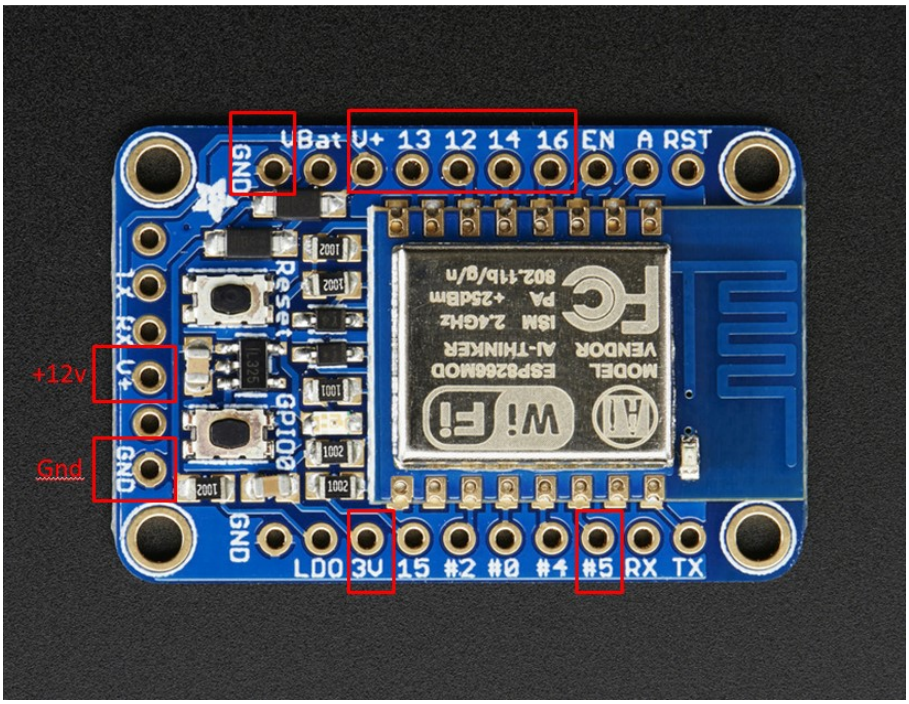
NOTE ON POWER:

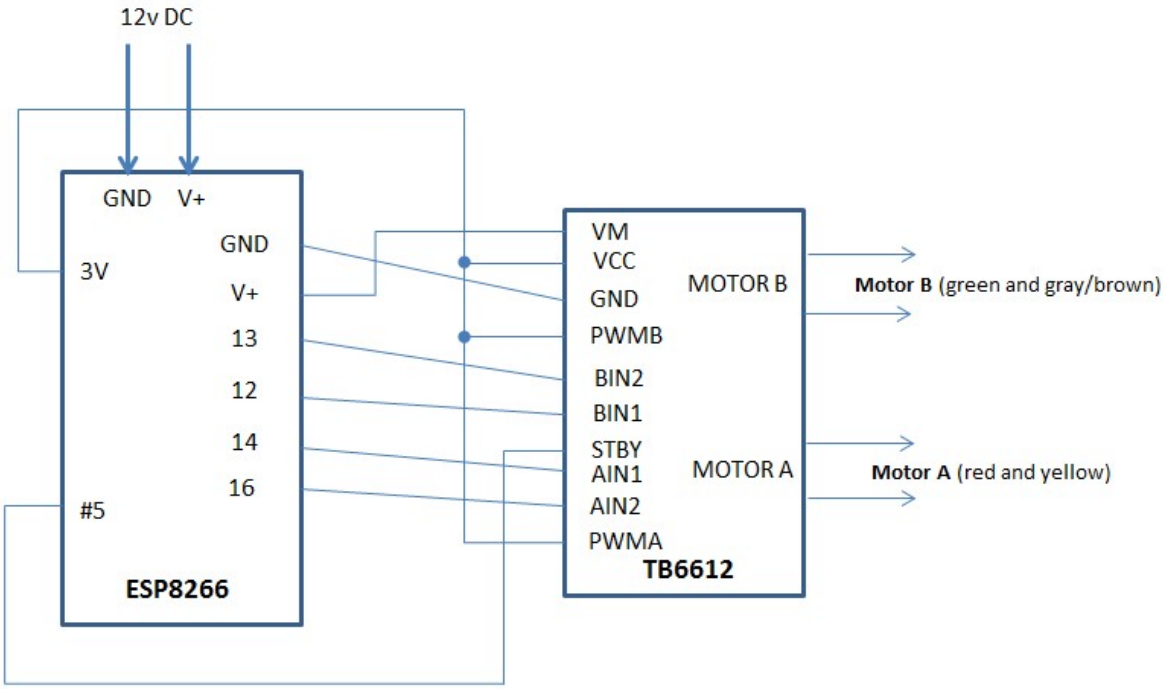
If your power supply in insufficient, you could cause the ESP8266 to reboot repeatedly or not be able to drive the Stepper Motor.

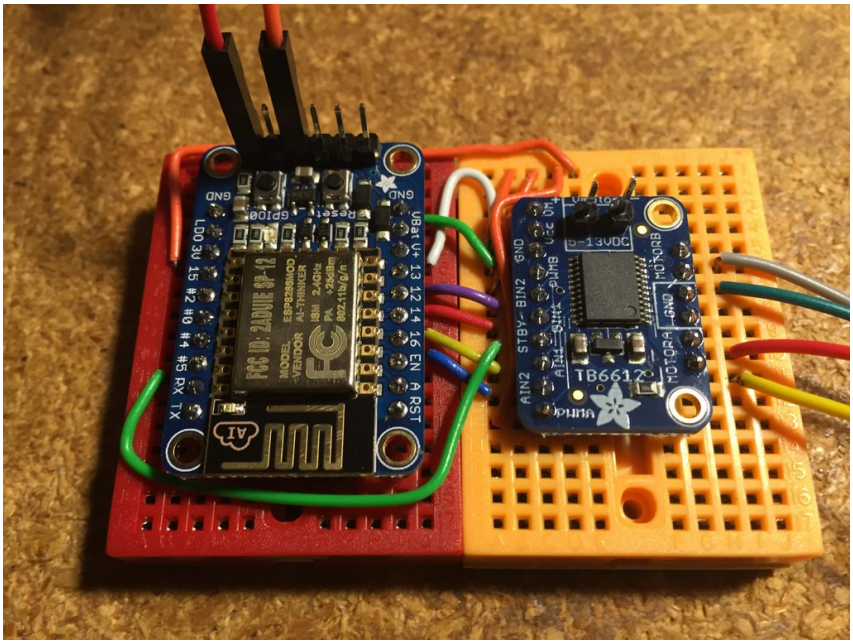
Also, Steppers can get really hot w/o current limiting. You may wish to use a separate current limited 12v power supply directly on the TB6612 **Vmotor** power leads. If so, then disconnect the WHITE wire between VM and V+ and attach a 5v-16v supply to the HUZZAH ESP8266. Be sure to leave the GND and 3v leads all in place.

Alternately, be sure to send a "STOP" command *(see below)* to switch the TB6612 to standby which removes power from the Stepper. Use "START" command to reapply power.


AIN2
STBY
BIN2
GND
AIN1
BIN1
PWMB
Vcc
VM
TB6612
TB445A1
6612FNG
5-13VDC
Vmotor
MOTORA
GND
MOTORB


VBat
V+
13
12
14
16
EN
A
RST
GND
Reset
GPIO0
+12v
Gnd
GND
LDO
3V
15
#2
#0
#4
#5
RX
TX





Testing

I used the built-in Arduino Stepper Library https://www.arduino.cc/en/Reference/Stepper to control the stepper motor. I added this to a simple HTTP control web server to provide remote control of the stepper.

Use a PC with web browser or Linux cURL command to try the functions below:

Functions available:

- http://{your_ip}/stepper/steps?5 -- Step 5 steps forward (values 1 to 200)
- http://{your_ip}/stepper/steps?-10 -- Step 10 steps backwards (-1 to -200)
- http://{your_ip}/stepper/rpm?6 -- Run at 6 RPM (1 to 60)
- http://{your_ip}/stepper/stop -- Stop the stepper motor (Removes power)
- http://{your_ip}/stepper/start -- Start the stepper motor (Restores power)

If you have a Linux machine (such as a Raspberry Pi) you can use the following simple automation example:

```
$ nano example.sh
```

Change ESP_IP to your ESP8266's IP address.

```
$ bash example.sh
```

Have Fun!