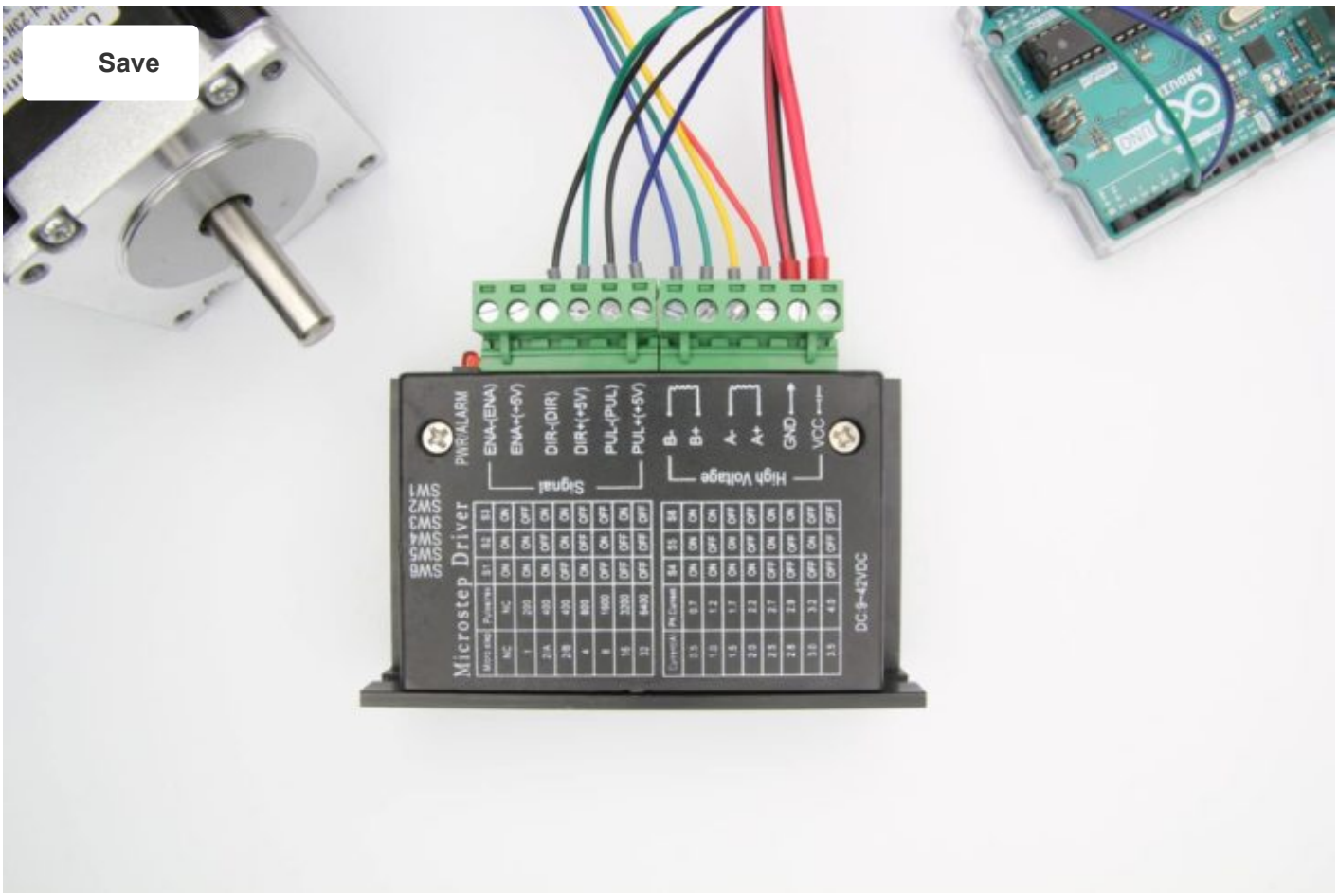
Makerguides.com

TB6600 Stepper Motor Driver with Arduino Tutorial

Written by Benne de Bakker (https://www.makerguides.com/author/benne-de-bakker/)



In this tutorial, you will learn how to control a stepper motor with the TB6600 microstepping driver and Arduino. This driver is easy to use and can control large stepper motors like a 3 A NEMA 23 (https://amzn.to/2HpVx6b).

I have included a wiring diagram and 3 example codes. In the first example, I will show you how you can use this stepper motor driver without an Arduino library. This example can be used to let the motor spin continuously. In the second example, we will look at how you can control the speed, number of revolutions, and spinning direction of the

stepper motor. Finally, we will take a look at the AccelStepper library. This library is fairly easy to use and allows you to add acceleration and deceleration to the movement of the stepper motor.

After each example, I break down and explain how the code works, so you should have no problems modifying it to suit your needs.

If you have any questions, please leave a comment below.

If you would like to learn more about other stepper motor drivers, then the articles below might be useful:

- TB6560 Stepper Motor Driver with Arduino Tutorial (https://www.makerguides.com/tb6560-stepper-motor-driver-arduino-tutorial/)
- How to control a stepper motor with A4988 driver and Arduino (https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/)
- 28BYJ-48 Stepper Motor with ULN2003 Driver and Arduino Tutorial (https://www.makerguides.com/28byj-48-stepper-motor-arduino-tutorial/)
- How to control a Stepper Motor with Arduino Motor Shield Rev3 (https://www.makerguides.com/arduino-motor-shield-stepper-motor-tutorial/)

Supplies

Hardware components

	TB6600 stepper motor driver (https://www.amazon.com/s/ref=as_li_ss_tl?k=tb6600&ref=nb_sb_noss_1&linkCode=ll2&tag=makerguid 20&linkId=fd1433071a741d50f702b5db6f423fef&language
	NEMA 23 stepper motor (https://amzn.to/2tKqhBR)

	Arduino Uno Rev3 (https://amzn.to/374aJjX)
(https://amzn.to/374aJjX)	
	Power supply (https://amzn.to/2PKWwT7) (24/36 V)
	Jumper wires (https://amzn.to/2EG9wDc)
	USB cable type A/B (https://amzn.to/34SBuXf)

Tools

	Wire stripper (https://amzn.to/2SiDQMg)	Amazon (https://amzn.to/2SiDQMg)
	Small screwdriver (https://amzn.to/2SkE0ms)	Amazon (https://amzn.to/2SkE0ms)
	Self-adjusting crimping pliers (https://amzn.to/376HZXV) (recommended)*	Amazon (https://amzn.to/376HZXV)
	Wire ferrules assortment (https://amzn.to/2s4W6hD) (recommended)*	Amazon (https://amzn.to/2s4W6hD)

*Hackaday (https://hackaday.com/2018/04/12/to-ferrule-or-not-to-ferrule/) wrote a great article on the benefits of using wire ferrules (also known as end sleeves).

Software

Arduino IDE (https://www.arduino.cc/en/Main/Software)

About the driver

The TB6600 microstepping driver (https://amzn.to/2Vks43D) is built around the Toshiba TB6600HG IC and it can be used to drive two-phase bipolar stepper motors.

With a maximum current of 3.5 A continuous, the TB6600 driver can be used to control quite large stepper motors like a NEMA 23. Make sure that you do not connect stepper motors with a current rating of more than 3.5 A to the driver.

The driver has several safety functions built-in like over-current, under-voltage shutdown, and overheating protection.

You can find more specifications in the table below. Note that the exact specifications and dimensions can differ slightly between manufacturers. Always take a look at the datasheet of your particular driver, before connecting power.

TB6600 Specifications

Operating voltage	9 – 42 V
Max output current	4.5 A per phase, 5.0 A peak[1]
Microstep resolution	full, 1/2, 1/4, 1/8 and 1/16[2]
Protection	Low-voltage shutdown, overheating and over-current protection
Dimensions	96 x 72 x 28/36 mm
Hole spacing	88, ⌀ 5 mm
Cost	Check price (https://amzn.to/2Vks43D)

[1] These are the specifications for the TB6600HG IC, the driver itself has a maximum current rating of 3.5 A and 4.0 A peak.
[2] See comment on fake/upgraded TB6600 drivers below.

For more information, you can check out the datasheet and manual below:

Toshiba TB6600 Datasheet

TB6600 Manual

Fake or ‘upgraded’ TB6600 drivers

I recently took apart one of the TB6600 drivers I ordered and found out that it didn't actually use a TB6600HG chip. Instead, it used a much smaller TB67S109AFTG chip, also made by Toshiba. The performance and specifications of these chips are similar, but the TB6600HG does have a higher peak current rating (up to 5 A) and it is just a much larger chip with better heatsinking overall.

There is a very simple way to check if your driver uses a TB6600HG chip or a TB67S109AFTG chip, **the TB6600HG only supports up to 1/16 microstepping** (see datasheet), whereas the TB67S109AFTG goes to 1/32. The main reason manufacturers switched over to this other chip is probably price. Below you can find links to the chips on LCSC.com which shows that the TB67S109AFTG is around $1.50 cheaper.

TB6600HG: https://lcsc.com/product-detail/Motor-Drivers_TOSHIBA_TB6600HG_TB6600HG_C66042.html (https://lcsc.com/product-detail/Motor-Drivers_TOSHIBA_TB6600HG_TB6600HG_C66042.html)
TB67S109AFTG: https://lcsc.com/product-detail/Motor-Drivers_TOSHIBA_TB67S109AFTG_TB67S109AFTG_C92125.html (https://lcsc.com/product-detail/Motor-Drivers_TOSHIBA_TB67S109AFTG_TB67S109AFTG_C92125.html)

You can buy genuine TB6600 drivers on Amazon, like this 4-axis driver board (https://amzn.to/32YNh5v) but most use the TB67S109AFTG chip. You can tell it uses the TB6600HG chip from the pins sticking out of the PCB and it also only goes up to 1/16 microstepping.

Jim from embeddedtronicsblog did some testing on the TB67S109AFTG drivers (https://embeddedtronicsblog.wordpress.com/2018/11/07/tb67s109aftg-stepper-motor-driver-testing/) and found that the stepper motors ran nicer than with the TB6600 drivers. So should you be going for a genuine TB6600 or the 'upgrade'? I would say it depends on whether you really need the high current output or if you rather prefer up to 1/32 microstepping.

You can find the datasheet for the TB67S109AFTG below.

TB67S109AFTG Datasheet

Alternatives

Note that the TB6600 is an analog driver. In recent years, digital drivers like the DM556 (https://amzn.to/2Zsqtc8) or DM542 (https://amzn.to/2ZdNm7W) have become much more affordable. Digital drivers usually give much better performance and quieter operation. They can be wired and controlled in the same way as the TB6600, so you can easily upgrade your system later.

I have used the DM556 drivers for my DIY CNC router and they have been working great for several years.

TB6600 vs TB6560

When shopping for a TB6600 stepper motor driver, you will probably come across the slightly cheaper TB6560 driver (https://amzn.to/30Ktl50) as well. This driver can be controlled with the same code/wiring, but there are some key differences.

	TB6560	TB6600
Operating voltage	10 – 35 VDC, 24 VDC recommended	9 – 42 VDC, 36 VDC recommended
Max output current	3 A per phase, 3.5 A peak	3.5 A per phase, 4 A peak
# Current settings	14	8
Microstep resolution	full, 1/2, 1/8 and 1/16	full, 1/2, 1/4, 1/8, 1/16 and 1/32*
Clock frequency	15 kHz	200 kHz
Cost	Check price (https://amzn.to/30Ktl50)	Check price (https://amzn.to/2Vks43D)

*Drivers using TB67S109AFTG chip.

So the main differences are the higher maximum voltage, higher maximum current, and up to 1/32 microstepping. The TB6600 also has a better heatsink and a nicer overall form factor. If you want to control larger stepper motors or need a higher resolution, I recommend going with the TB6600.

Wiring – Connecting TB6600 to stepper motor and Arduino

Connecting the TB6600 stepper motor driver to an Arduino and stepper motor is fairly easy. The wiring diagram below shows you which connections you need to make.

TB6600 stepper motor driver with Arduino UNO and stepper motor wiring diagram

In this tutorial, we will be connecting the driver in a common cathode configuration. This means that we connect all the negative sides of the control signal connections to ground.

The connections are also given in the table below:

TB6600 Connections

TB6600	Connection
VCC	9 – 42 VDC
GND	Power supply ground
ENA-	Not connected
ENA+	Not connected
DIR-	Arduino GND
DIR+	Pin 2 Arduino
PUL-	Arduino GND
PUL+	Pin 3 Arduino
A-, A+	Coil 1 stepper motor
B-, B+	Coil 2 stepper motor

Note that we have left the enable pins (ENA- and ENA+) disconnected. This means that the enable pin is always LOW and the driver is always enabled.

How to determine the correct stepper motor wiring?

If you can not find the datasheet of your stepper motor, it can be difficult to figure out which color wire goes where. I use the following trick to determine how to connect 4 wire bipolar stepper motors:

The only thing you need to identify is the two pairs of wires which are connected to the two coils of the motor. The wires from one coil get connected to A- and A+ and the other to B- and B+, the polarity doesn't matter.

To find the two wires from one coil, do the following with the motor disconnected:

1. Try to spin the shaft of the stepper motor by hand and notice how hard it is to turn.
2. Now pick a random pair of wires from the motor and touch the bare ends together.
3. Next, while holding the ends together, try to spin the shaft of the stepper motor again.

If you feel a lot of resistance, you have found a pair of wires from the same coil. If you can still spin the shaft freely, try another pair of wires. Now connect the two coils to the pins shown in the wiring diagram above.

(If it is still unclear, please leave a comment below, more info can also be found on the RepRap.org wiki (https://reprap.org/wiki/Stepper_wiring))

TB6600 microstep settings

Stepper motors typically have a step size of 1.8° or 200 steps per revolution, this refers to full steps. A microstepping driver such as the TB6600 allows higher resolutions by allowing intermediate step locations. This is achieved by energizing the coils with

intermediate current levels.

For instance, driving a motor in 1/2 step mode will give the 200-steps-per-revolution motor 400 microsteps per revolution.

You can change the TB6600 microstep settings by switching the dip switches on the driver on or off. See the table below for details. Make sure that the driver is not connected to power when you adjust the dip switches!

Please note that these settings are for the 1/32 microstepping drivers with the TB67S109AFTG chip. Almost all the TB6600 drivers you can buy nowadays use this chip. Typically you can also find a table with the microstep and current settings on the body of the driver.

Microstep table

S1	S2	S3	Microstep resolution
ON	ON	ON	NC
ON	ON	OFF	Full step
ON	OFF	ON	1/2 step
OFF	ON	ON	1/2 step
ON	OFF	OFF	1/4 step
OFF	ON	OFF	1/8 step
OFF	OFF	ON	1/16 step
OFF	OFF	OFF	1/32 step

Generally speaking, a smaller microstep setting will result in a smoother and quieter operation. It will however limit the top speed that you can achieve when controlling the stepper motor driver with an Arduino.

TB6600 current settings

You can adjust the current that goes to the motor when it is running by setting the dip switches S4, S5, and S6 on or off. I recommend starting with a current level of 1 A. If your motor is missing steps or stalling, you can always increase the current level later.

Current table

Current (A)	Peak current	S4	S5	S6
0.5	0.7	ON	ON	ON
1.0	1.2	ON	OFF	ON
1.5	1.7	ON	ON	OFF
2.0	2.2	ON	OFF	OFF
2.5	2.7	OFF	ON	ON
2.8	2.9	OFF	OFF	ON
3.0	3.2	OFF	ON	OFF
3.5	4.0	OFF	OFF	OFF

Basic TB6600 with Arduino example code

With the following sketch, you can test the functionality of the stepper motor driver. It simply lets the motor rotate at a fixed speed.

You can upload the code to your Arduino using the Arduino IDE (https://www.arduino.cc/en/main/software). For this specific example, you do not need to install any libraries.

In the next example we will look at controlling the speed, number of revolutions and spinning direction of the stepper motor.

You can copy the code by clicking on the button in the top right corner of the code field.

```
/* Example sketch to control a stepper motor with TB6600 stepper motor driver and Arduino without a library: continuous rotation. More info: https://www.makerguides.com */

// Define stepper motor connections:
#define dirPin 2
#define stepPin 3

void setup() {
  // Declare pins as output:
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);

  // Set the spinning direction CW/CCW:
  digitalWrite(dirPin, HIGH);
}

void loop() {
  // These four lines result in 1 step:
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(500);
}
```

As you can see, the code is very short and super simple. You don't need much to get a stepper motor spinning!

Code explanation

The sketch starts with defining the step (PUL+) and direction (DIR+) pins. I connected them to Arduino pin 3 and 2.

The statement `#define` is used to give a name to a constant value. The compiler will replace any references to this constant with the defined value when the program is compiled. So everywhere you mention `dirPin`, the compiler will replace it with the value 2 when the program is compiled.

```
// Define stepper motor connections:
#define dirPin 2
#define stepPin 3
```

In the `setup()` section of the code, all the motor control pins are declared as digital OUTPUT with the function `pinMode(pin, mode)`. I also set the spinning direction of the stepper motor by setting the direction pin HIGH. For this we use the function `digitalWrite(pin, value)`.

```
void setup() {
  // Declare pins as output:
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);

  // Set the spinning direction CW/CCW:
  digitalWrite(dirPin, HIGH);
}
```

In the `loop()` section of the code, we let the driver execute one step by sending a pulse to the step pin. Since the code in the loop section is repeated continuously, the stepper motor will start to rotate at a fixed speed. In the next example, you will see how you can change the speed of the motor.

```
void loop() {
  // These four lines result in 1 step:
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(500);
}
```

---

2. Example code to control rotation, speed and direction

This sketch controls both the speed, the number of revolutions and the spinning direction of the stepper motor.

```
 /* Example sketch to control a stepper motor with TB6600 stepper motor driver and Arduino without a library: number of revolutions, speed and direction. More info: https://www.makerguides.com */

// Define stepper motor connections and steps per revolution:
#define dirPin 2
#define stepPin 3
#define stepsPerRevolution 1600

void setup() {
  // Declare pins as output:
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}

void loop() {
  // Set the spinning direction clockwise:
  digitalWrite(dirPin, HIGH);

  // Spin the stepper motor 1 revolution slowly:
  for (int i = 0; i < stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2000);
  }

  delay(1000);

  // Set the spinning direction counterclockwise:
  digitalWrite(dirPin, LOW);

  // Spin the stepper motor 1 revolution quickly:
  for (int i = 0; i < stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }

  delay(1000);

  // Set the spinning direction clockwise:
  digitalWrite(dirPin, HIGH);

  // Spin the stepper motor 5 revolutions fast:
  for (int i = 0; i < 5 * stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
```

```
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
  }

  delay(1000);

  // Set the spinning direction counterclockwise:
  digitalWrite(dirPin, LOW);

  // Spin the stepper motor 5 revolutions fast:
  for (int i = 0; i < 5 * stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
  }

  delay(1000);
}
```

How the code works:

Besides setting the stepper motor connections, I also defined a `stepsPerRevolution` constant. Because I set the driver to 1/8 microstepping mode I set it to 1600 steps per revolution (for a standard 200 steps per revolution stepper motor). Change this value if your setup is different.

```
// Define stepper motor connections and steps per revolution:
#define dirPin 2
#define stepPin 3
#define stepsPerRevolution 1600
```

The `setup()` section is the same as before, only we don't need to define the spinning direction just yet.

In the `loop()` section of the code, we let the motor spin one revolution slowly in the CW direction and one revolution quickly in the CCW direction. Next, we let the motor spin 5 revolutions in each direction with a high speed. So how do you control the speed, spinning direction and number of revolutions?

```
15.   // Set the spinning direction clockwise:
16.     digitalWrite(dirPin, HIGH);
17.
18.     // Spin the stepper motor 1 revolution slowly:
19.     for(int i = 0; i < stepsPerRevolution; i++)
20.     {
21.       // These four lines result in 1 step:
22.       digitalWrite(stepPin, HIGH);
23.       delayMicroseconds(2000);
24.       digitalWrite(stepPin, LOW);
25.       delayMicroseconds(2000);
26.     }
```

Control spinning direction:

To control the spinning direction of the stepper motor we set the DIR (direction) pin either HIGH or LOW. For this we use the function `digitalWrite()` . Depending on how you connected the stepper motor, setting the DIR pin high will let the motor turn CW or CCW.

Control number of steps or revolutions:

In this example sketch, the for loops (https://www.arduino.cc/reference/en/language/structure/control-structure/for/) control the number of steps the stepper motor will take. The code within the for loop results in 1 (micro)step of the stepper motor. Because the code in the loop is executed 1600 times (stepsPerRevolution), this results in 1 revolution. In the last two loops, the code within the for loop is executed 8000 times, which results in 8000 (micro)steps or 5 revolutions.

Note that you can change the second term in the for loop to whatever number of steps you want. `for(int i = 0; i < 800; i++)` would result in 800 steps or half a revolution.

Control speed:

The speed of the stepper motor is determined by the frequency of the pulses we send to the STEP pin. The higher the frequency, the faster the motor runs. You can control the frequency of the pulses by changing `delayMicroseconds()` in the code. The shorter the delay, the higher the frequency, the faster the motor runs.

Installing the AccelStepper library

The AccelStepper library written by Mike McCauley is an awesome library to use for your project. One of the advantages is that it supports acceleration and deceleration, but it has a lot of other nice functions too.

You can download the latest version of this library here (https://www.airspayce.com/mikem/arduino/AccelStepper/index.html) or click the button below.

AccelStepper-1.59.zip

You can install the library by going to **Sketch > Include Library > Add .ZIP Library...** in the Arduino IDE.

Another option is to navigate to **Tools > Manage Libraries...** or type Ctrl + Shift + I on Windows. The Library Manager will open and update the list of installed libraries.

You can search for ‘accelstepper’ and look for the library by Mike McCauley. Select the latest version and then click Install.

3. AccelStepper example code

With the following sketch, you can add acceleration and deceleration to the movements of the stepper motor, without any complicated coding. In the following example, the motor will run back and forth with a speed of 1000 steps per second and an acceleration of 500 steps per second squared.

Note that I am still using the driver in 1/8 microstepping mode. If you are using a different setting, play around with the speed and acceleration settings.

```
 /* Example sketch to control a stepper motor with TB6600 stepper motor driver, AccelStepper library and Arduino: acceleration and deceleration. More info: https://www.makerguides.com */

// Include the AccelStepper library:
#include <AccelStepper.h>

// Define stepper motor connections and motor interface type. Motor interface type must be set to 1 when using a driver:
#define dirPin 2
```

```
#define stepPin 3
#define motorInterfaceType 1

// Create a new instance of the AccelStepper class:
AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin, dirPin);

void setup() {
  // Set the maximum speed and acceleration:
  stepper.setMaxSpeed(1000);
  stepper.setAcceleration(500);
}

void loop() {
  // Set the target position:
  stepper.moveTo(8000);
  // Run to target position with set speed and acceleration/deceleration:
  stepper.runToPosition();

  delay(1000);

  // Move back to zero:
  stepper.moveTo(0);
  stepper.runToPosition();

  delay(1000);
}
```

Code explanation:

The first step is to include the library with `#include <AccelStepper.h>` .

```
// Include the AccelStepper library:
#include <AccelStepper.h>
```

The next step is to define the TB6600 to Arduino connections and the motor interface type. The motorinterface type must be set to 1 when using a step and direction driver. You can find the other interface types here (https://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html#a608b2395b64ac15451d16d0371fe13ce).

```
// Define stepper motor connections and motor interface type. Motor interface type must be set to 1 when using a driver:
#define dirPin 2
#define stepPin 3
#define motorInterfaceType 1
```

Next, you need to create a new instance of the AccelStepper class with the appropriate motor interface type and connections.

In this case, I called the stepper motor 'stepper' but you can use other names as well, like 'z_motor' or 'liftmotor' etc. `AccelStepper liftmotor = AccelStepper(motorInterfaceType, stepPin, dirPin);` . The name that you give to the stepper motor will be used later to set the speed, position, and acceleration for that particular motor. You can create multiple instances of the AccelStepper class with different names and pins. This allows you to easily control 2 or more stepper motors at the same time.

```
// Create a new instance of the AccelStepper class:
AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin, dirPin);
```

In the setup(), besides the maximum speed, we need to define the acceleration/deceleration. For this we use the function `setMaxSpeed()` and `setAcceleration()` .

```
void setup() {
  // Set the maximum speed and acceleration:
  stepper.setMaxSpeed(1000);
  stepper.setAcceleration(500);
}
```

In the loop section of the code, we let the motor rotate a predefined number of steps. The function `stepper.moveTo()` is used to set the target position (in steps). The function `stepper.runToPostion()` moves the motor (with acceleration/deceleration) to the target position and blocks until it is at the target position. Because this function is blocking, you shouldn't use this when you need to control other things at the same time.

```
    // Set the target position:
    stepper.moveTo(8000);
    // Run to target position with set speed and acceleration/deceleration:
    stepper.runToPosition();
```

If you would like to see more examples for the AccelStepper libary, check out my tutorial for the A4988 stepper motor driver:

- How to control a stepper motor with A4988 driver and Arduino (https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/)

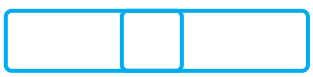
Conclusion

In this article, I have shown you how to control a stepper motor with the TB6600 stepper motor driver and Arduino. I hope you found it useful and informative. If you did, please **share it with a friend** who also likes electronics and making things!

I would love to know what projects you plan on building (or have already built) with this driver. If you have any questions, suggestions, or if you think that things are missing in this tutorial, **please leave a comment down below**.

Note that comments are held for moderation to prevent spam.

Beginner


What to read next?

LM35 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/lm35-arduino-tutorial/)

TMP36 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/tmp36-arduino-tutorial/)

Arduino Nano Board Guide (Pinout, Specifications, Comparison) (https://www.makerguides.com/arduino-nano/)

The complete guide for DS18B20 digital temperature sensors with Arduino (https://www.makerguides.com/ds18b20-arduino-tutorial/)

How to use an IR receiver and remote with Arduino (https://www.makerguides.com/ir-receiver-remote-arduino-tutorial/)

Comments

John Doe says
January 18, 2021 at 12:39 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-6057)

Phenomenal,

Great work, much appreciated, very well explained

Reply

Robert Born says
January 16, 2021 at 1:57 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-6020)

Very nice explaination!
I have a simple question .. I want to control multiple (actually 4) 23-frame steppers same speed/direction for a conveyor application. I want these to be synchronized exactly as possible for smooth operation.
Can the four tb4400 drivers be daisy chained to the same Arduino GPIO pins? Is there a limit to how many times (number of drivers) I can do this? I'm wondering about current draw from the 5volt Arduino GPIO spread out over many drivers.
Thanks for your help!
Bob

Reply

Robert Born says
January 16, 2021 at 1:59 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-6021)

Oops I meant tb6600

Reply

Amina says
December 31, 2020 at 12:15 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-5726)

Hi,
Thank you for this tutorial which is very useful.
I want to use the DM556 driver for my stepper motors and I want to add an absolute encoder to the motor. It is possible to command the stepper motor in function of values that I will receive from the encoder using this code?
Thank you

Reply

Rick Morley (http://United%20States) says
December 10, 2020 at 6:52 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-5271)

I am using the stepper motor to control the angle of a solar panel so that it always points towards the sun. I needed this tutorial to tell me the basic operation of the TB6600 stepper motor driver. You did a great job doing that.

Reply

João Correia says
November 19, 2020 at 6:08 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-4887)

Awesome!! Thanks for all this great information! Very, very useful.

Reply

EddyCurr says
October 3, 2020 at 3:07 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-4051)

I suggest a correction as follows. Substitute (PUL+) and (DIR+) in place of (PUL-) and (DIR-) where these appear in the text snippet below:

“Code explanation

The sketch starts with defining the step (PUL-) and direction (DIR-) pins. I connected them to Arduino pin 3 and 2."

Earlier in the "Wiring" section of the tutorial, a diagram displays step (PUL-) and direction (DIR-) pins connected to Arduino "Gnd", this is followed by text stating that "we will be connecting the driver in a common cathode configuration."

That said, I join the other posters in complimenting you for a VERY helpful tutorial about the TB6600 driver. Well done.

In place of an Arduino, I am a PWM Signal Generator device to control a stepper motor. I came across the tutorial while searching for knowledge about what determines when the TB6600 driver (and others like it) should be connected in: a) Common Cathode config and b) Common Anode config.

While I did not find the answer to my question, I learned a number of other things that improve my understanding about drivers.

Reply

Benne de Bakker says
October 3, 2020 at 3:17 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-4052)

Thank you for the suggestion!

Reply

Mike says

August 24, 2020 at 3:48 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-3599)

Benne,

I notice if the unit sits powered, the motors get a bit warm. Not hot. I have a 2.8amp NEMA 23 and switches at 2.8-2.9 amps set. Assuming this is due to the EN+ and EN- on all the time. If I wanted to have a on/off switch, but leaving the Power Supply on. Would I run all the EN+ to a pull-down resistor to ground? I have not added a power switch yet. Maybe I will just do that.

Reply

Benne de Bakker says
August 24, 2020 at 3:58 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-3601)

Hi Mike,

If you want to disable the driver, connect EN- to GND and EN+ to one of the digital pins of the Arduino. To disable the driver, you can use digitalWrite(pin,HIGH); in your Arduino code. To enable it again, simply set the pin low with digitalWrite(pin,LOW);. Note that you need to replace ‘pin’ with the pin number to which EN+ is connected.

Benne

Reply

Konrad says

June 27, 2020 at 4:22 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-3050)

Hello!

Thank you for the tutorial, it was very helpful however I do have a problem. I have no access to a 36V power supplier and the strongest one I have is 19V which I suspect is not really a problem for the motor that I use which is the sl42sth40-1504a.
While everything seems to work, the motor jitters in a random fashion instead of spinning and so what I'd like to know is if this is a problem with my wiring (which I've done exactly as you described above) or is it simply the voltage that's incorrect please?
I would very much appreciate an answer as I'm in the middle of a project ☺

Many thanks!

Reply

Benne de Bakker says
June 29, 2020 at 10:59 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-3070)

Hi Konrad,

19 V should be enough to run the stepper motor with the TB6600 driver. You could try to increase the current settings with the dip switches which might solve the problem that you are having.

Benne

Reply

Harry Zuzan says
June 26, 2020 at 5:46 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-3033)

Thank you for this straight forward tutorial. Made things easy for me. Other tutorials complicate things with potentiometers and switches that I can worry about later. This way is better.

Reply

Michael says
March 27, 2020 at 2:34 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-2190)

Dear Ladies, dear gentlemen,
since I'm an absolute beginner I would like to thank you for the work and time you spend building this webpage. Your explanations were very helpful for my project. However, you stated: "The wires from one coil get connected to A- and A+ and the other to B- and B+, the polarity doesn't matter." For the stepper motor I used it was very important since the sketches you posted will not work. I wired the stepper motor like you displayed in your drawing. I figured out which wires belong to coil A and to coil B. After hours rewirering the connections between TB6600 and the Arduiono board again and again without any success to get the motor spinning I changed the wires of the power supply of the motor. Means I changed A+ to A- and B+ to B-. This makes at last the motor spinning like it should. I only therefore can assume that correct wiring is important.
May you would like to know that I'm using a NEMA 17 stepper Motor assembled by

Quimat. By the plate attached it is identified as 17HS441.
Best regards
Michael

Reply

Bob Kubichek says
January 23, 2020 at 1:06 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1661)

Really Excellent. Thanks!

Reply

Martin Rowan (http://www.martinrowan.co.uk) says
December 17, 2019 at 4:20 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1364)

Thanks, I'll do that.

Reply

Martin Rowan (http://www.martinrowan.co.uk) says
December 17, 2019 at 3:45 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1362)

Do you have or can direct me at the source for the TB6600 Fritzing Part? I'm having some issues with a project I've built and wanted to use Fritzing to show what I've created. Thanks

Reply

Benne de Bakker says
December 17, 2019 at 3:55 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1363)

Hi Martin,

I couldn't find a TB6600 Fritzing Part online, so the graphic you see in this tutorial was partially created with illustrator.

A quick way to add a non-existing part to Fritzing is to insert it as an image (you can find the insert button under Core > Breadboard view). So if it's just for a quick diagram to troubleshoot I would do that.

Creating your own custom parts is not super easy and a bit time consuming, so I generally try to avoid it.

Benne

Reply

Obi says
December 8, 2019 at 8:52 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1277)

Awesome! The article cleared some air and paved out ways against procrastination...thanks. You should make a youtube video if you havent already. if you have please help me with the link.

Reply

Jason Blubaugh says
November 26, 2019 at 2:46 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1175)

The power supply says 9-42VDC. Do you have a more specific recommendation?

Reply

Benne de Bakker says
November 26, 2019 at 11:53 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1179)

Hi Jason,

I recommend using a 36-volt power supply when using larger stepper motors (NEMA 23) and 24 volts when using small NEMA 17 stepper motors.

Benne

Reply

Peter Smart says
December 5, 2019 at 5:02 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-1260)

Hi Benne,

The selection of supply voltage to drive the stepper does not actually feature in the article.

Would be interested to know why you make the recommendation of 36V for NEMA 23 and 24 for NEMA 17.

Thanks

Peter

Reply

David Kauppi says
October 11, 2019 at 2:24 am (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-810)

Awesome!! This got me up and running in very short order. This helped me so much. Thank you!!!

Reply

Aschwin Rutgers says
October 8, 2019 at 2:21 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-784)

Thank you verry much.
Was a clear demonstration for a noob in stepper motors.

Reply

Benne de Bakker says

October 8, 2019 at 2:40 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-785)

Thank you for the feedback!

Reply

Trackbacks

Stepper Motor with DRV8825 and Arduino Tutorial (4 Examples) (https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/) says:
October 4, 2019 at 6:56 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-736)
[...] TB6600 Stepper Motor Driver with Arduino Tutorial [...]

Stepper Motor with A4988 and Arduino Tutorial (4 Examples) (https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/) says:
October 4, 2019 at 6:55 pm (https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/#comment-735)
[...] TB6600 Stepper Motor Driver with Arduino Tutorial [...]

(https://www.makerguides.com/maxbotix-mb1240-arduino-tutorial/)

MaxBotix MB1240 ultrasonic distance sensor Arduino tutorial (https://www.makerguides.com/maxbotix-mb1240-arduino-tutorial/)

(https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/)

How to control a stepper motor with A4988 driver and Arduino (https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/)

(https://www.makerguides.com/fsr-arduino-tutorial/)

Force Sensing Resistor (FSR) with Arduino Tutorial (https://www.makerguides.com/fsr-arduino-tutorial/)