

USER MANUAL

Advanced Stepper Motor Driver



Introduction:

This Stepper motor driver is based on A3967 IC, which is generally used in 3D printers & other CNC machines and incorporates reduced audible motor noise, increased step accuracy & reduced power dissipation. As a minimum, this stepper driver requires only **one digital pin** of a microcontroller to control a bipolar stepper motor. Advanced control requires two additional digital pins to control the direction of rotation and to put the driver into sleep mode to save battery power.

Current control, which is a very important concern while driving a stepper motor, can be achieved very easily by this **chopper driver**. You just have to rotate a potentiometer placed on this board to set the output current limit. You can also choose from various step sizes by adjusting the onboard DIP switches.

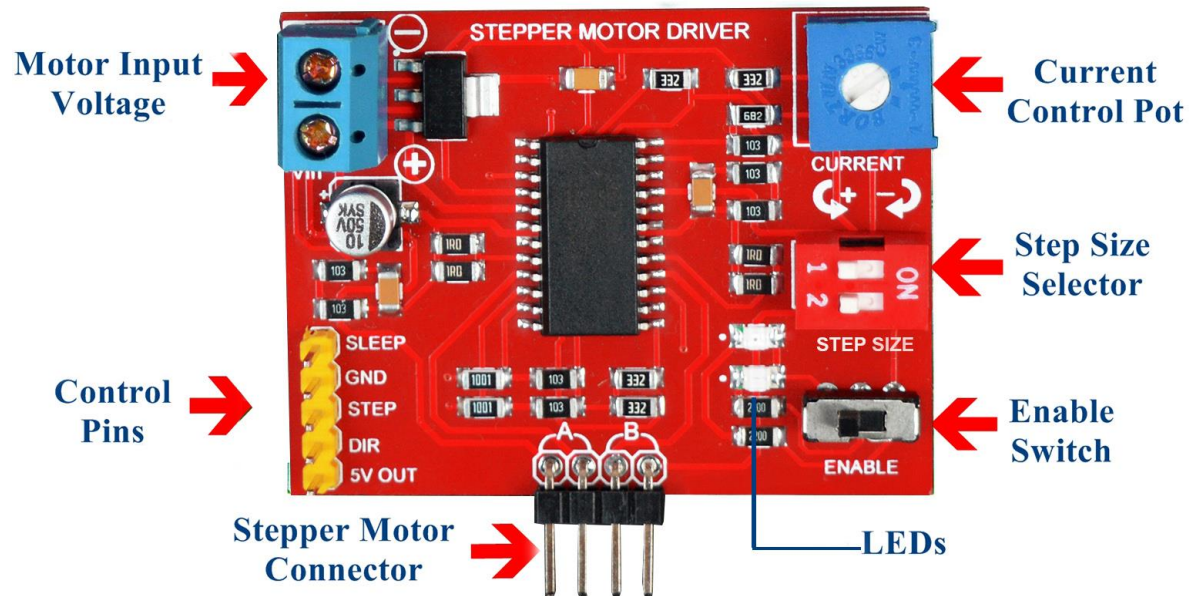
Features:

1. 2 phase bipolar stepper motor of 4, 6 & 8 wires can be controlled.
2. Minimum control requires pulses (square waves) that can be generated by only one pin of a microcontroller or any pulse generator IC (like NE555)
3. Motor rotation direction can be controlled.
4. Access to “ENABLE” and “SLEEP” feature of the A3967 IC.
5. Motor current (maximum output current) can be controlled between 300mA and 1.5A.
6. Full, half, quarter or micro stepping can be chosen.
7. On-board 5V voltage regulator to provide logic voltage to the controller IC.
8. Power & Enable LEDs. Enable switch.

Specifications:

1. Supported motors: 2 phase bipolar stepper motors of 4, 6 or 8 wires.
2. No. of control pins: 3 (only 1 required for minimum control)
3. Output current control limit: 300mA (min) to 1.5 A (max).
4. Input voltage (V_{in}): 7V (min) to 30V (max).
5. Logic supply voltage: 5V.
6. Logic voltage (signal pins): 3.5V min (for logic 1), 1.5V max (for logic 0).
7. Max pulse frequency at STEP pin: 500 KHz (1 uS high, 1uS low, time period 2 uS minimum).
8. Max controller IC temperature while in operation: 150 °C.

Components:



Control Pins:

- STEP:** To provide pulse of **equal** high & low time. The frequency should be **less** than 500 KHz (1 μ S high, 1 μ S low). This frequency should be adjusted depending on your stepper motor and your required speed.
- DIR:** Put digital 0 or leave disconnected to rotate the motor in the primary direction. Put digital 1 here to rotate the stepper motor to opposite direction.
- SLEEP:** Active LOW. Will put the driver IC to sleep mode and reduce current consumption to the least possible amount even if the “ENABLE” switch is active.
- GND:** For ground shorting.
- 5V OUT:** To provide 5V, generated by the onboard voltage regulator, to other devices for example to power up an Arduino. **Don't draw more than 100mA from here. Don't provide external 5V voltage to this pin as this is a voltage output pin, not input.**

Motor Input Voltage:

Provide the operating voltage of the stepper motor in this screw terminal (7V-30V). There's '+' and '-' marking on the PCB. **Be really careful about the polarity. The driver IC might get permanently damaged in case of reverse polarity.**

Moreover, you need to give the driver IC some time to initialize by providing motor input voltage before putting any pulse to the "STEP" pin. **Not doing so might damage the IC.** If you're using an arduino to generate pulse for the STEP pin and chose to power it up from the "5V OUT" pin, the bootloader delay of arduino should be enough to provide the driver IC sufficient time to initiate.

Stepper Motor Connector: It's where you connect the wires of your stepper motor. Please go through the "**Preparing the motor**" section before connecting anything here. You can take off the white plastic of the motor connector if you want.

Always connect the motor first and then provide power to the driver board. Never disconnect the motor while the driver board is powered up. **If you put pulse at "STEP" pin and motor input voltage at the screw terminal without attaching the motor, the driver IC will be fried.**

Enable Switch: To enable/ disable the motor driver.

Step size selector: Two separate slide switches (1&2) control step size.

Switch 1 state	Switch 2 state	Step Size
on	on	Micro Step
on	off	Quarter Step
off	on	Half Step
off	off	Full Step

Current Control Pot: Set maximum current the stepper motor is allowed to draw between 300mA to 1.5A (150mA to 750mA per coil). Rotate clockwise to decrease and anti-clockwise to increase output current limit.

A high current limit increases motor torque but drains the battery very fast and heats up the controller. So, always start from the lowest current set point and increase it until the motor torque is sufficient for your application.

LEDs: There's a red power LED and a blue enable indicator LED.

Choosing the motor:

Any 2 phase bipolar stepper motor, of 4, 6 or 8 wires, having motor operating voltage (V_{in}) up to 30V can be driven properly with this controller. As this is a chopper driver (controlled output current), you don't have to worry about motor coil resistance. It's good if the rated current PER COIL is less than 750mA (or 1.5A for both of the coils). But if not, you'll get less torque from your motor.

Motor Type	Bipolar
Phase	2
No. of wires	4, 6, 8
Motor operating Voltage	30V max
Motor rated current	750mA max (per phase), 1.5A max (total)
Motor coil resistance	Anything

Preparing the motor:

Only 4, 6 or 8 wire 2 phase bipolar steppers can be controlled. Never connect a 5 wire stepper or any unipolar stepper motor. **Connecting a wrong motor or motor wires in a wrong way may fry the driver IC.**

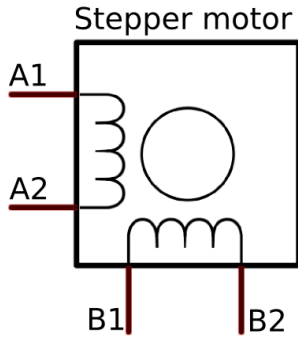
4 wire stepper:

There're two coils inside bipolar stepper motors. Each of the coils have two wires coming outside the motor. While connecting the wires to the 4 pin motor connector of this driver, we should remember this simple rule:

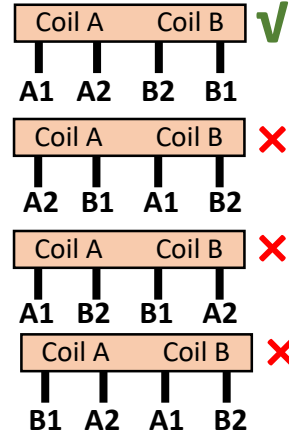
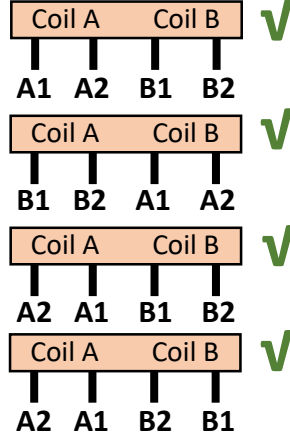
Always keep the wires of the same coil together.

Interchanging the coils or wires of the same coil doesn't matter. Let's call the coils as Coil A & Coil B. And the wires as A1, A2, B1, B2. See the examples of correct and wrong connections for easy understanding illustrated in next page.

So, **How to know which wire is which of your 4 wire stepper?** Just take a multimeter and measure resistance between any two wires. If the resistance is very high (> 1 Mega ohm), then they're of different coils. If there's less than 100 ohm resistance between 2 wires, then they're from the same coil.



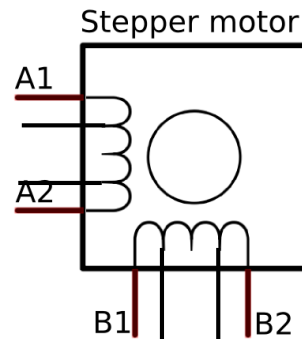
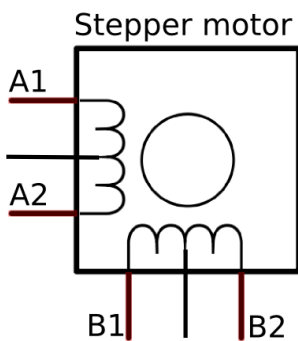
Motor Connector



Example of correct and Wrong Connection

6/8 wire stepper:

Each of the two coils have 3/4 wires coming out of the motor. What you have to do is just keep the middle one(s) disconnected. Take a multimeter, measure resistance between any 2 wires and continue doing so to detect two wires which have the greatest resistance but aren't disconnected.



Usage instruction (step by step):

STEP 1: Connect stepper motor to motor driver properly. You can take off the white plastic of the motor wire connector (from the driver board) if you want. Be careful of loose connection.

STEP 2: Provide Vin and GND to the blue screw terminal. Vin is your motors operating voltage. Be really careful about the '+' & '-' polarity.

STEP 3: Upload test code to Arduino. Then disconnect the usb cable from Arduino board.

STEP 4: Rotate the Current control pot to the extreme clockwise position (lowest current).

STEP 5: Connect wires to Arduino. The motor driver will power up your Arduino.

STEP 6: Enable the motor driver by pressing the Enable button.

STEP 7: The motor will start to rotate if everything is okay.

STEP 8: Change position of the red DIP switch to observe variation in step size. You can also rotate the current control pot anti-clockwise a bit to increase motor torque. The driver IC will get hotter (which is okay up to 150 °C). Do it only if the motor fails to rotate properly.

STEP 9: To turn off, disable the motor driver by pressing the Enable button again. Don't disconnect the motor.

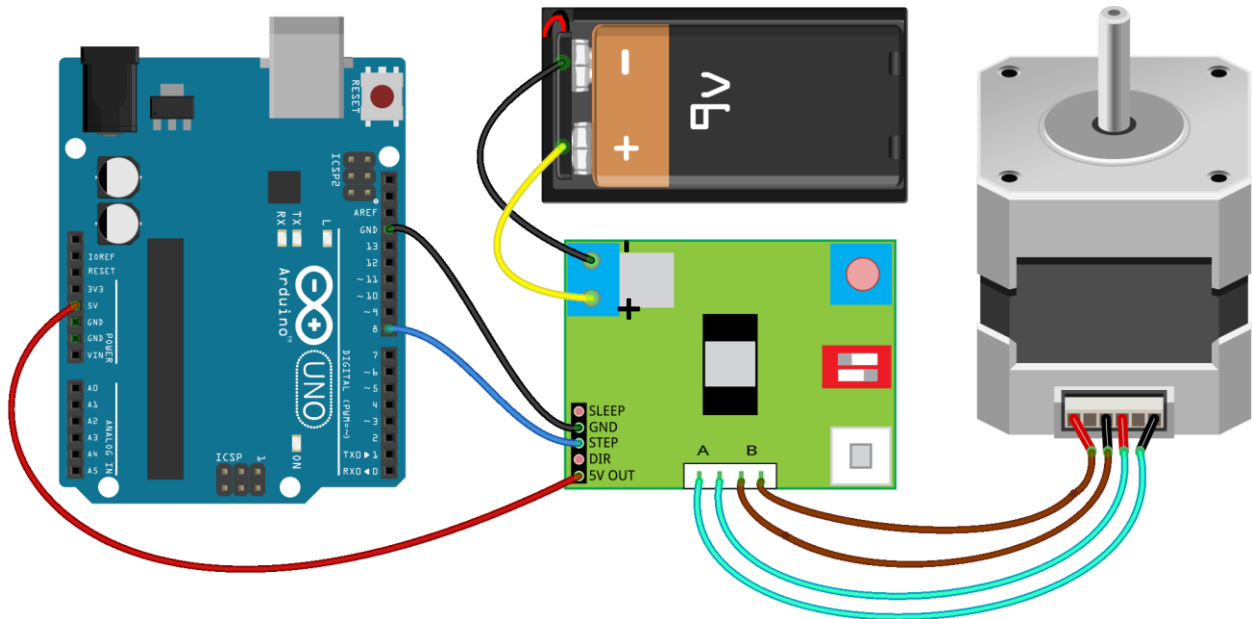
STEP 10: If you want to disconnect the motor from the driver board, first disconnect all connections from Arduino, then disconnect motor operating voltage (Vin) and GND wire and then disconnect the motor.

If you disconnect the motor while the driver board is powered up (Red LED on), the driver IC may get damaged permanently.

Basic Control:

Basic control lets you rotate the motor using only 1 digital pin of your Arduino. You will be able to control motor speed, step size and maximum current drawn by the motor.

Arduino Connection:



Test Code:

```
#define stepPin 8
int stepDelay=5; /* change it according to your need */

void setup()
{
  pinMode(stepPin, OUTPUT);
}

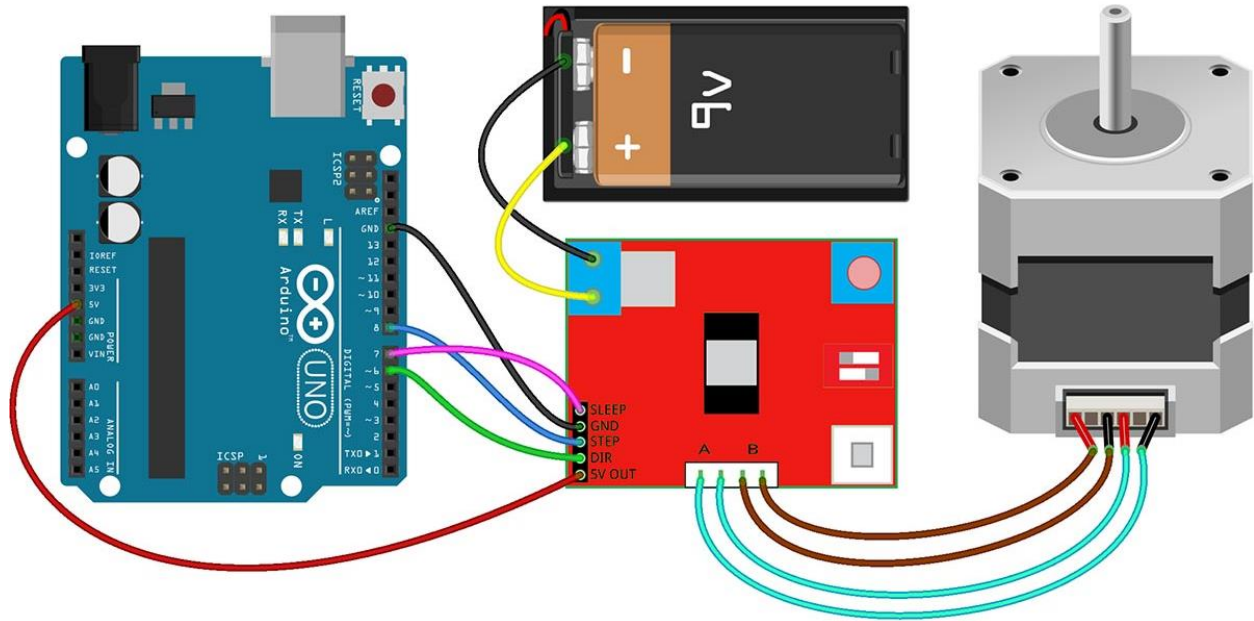
void loop()
{
  runMotor(1000);
  delay(1000);
  runMotor(500);
  delay(2000);
}

void runMotor(int steps)
{
  for (int i=0;i<abs(steps);i++)
  {
    digitalWrite(stepPin, HIGH);delay(stepDelay);
    digitalWrite(stepPin, LOW);delay(stepDelay);
  }
}
```


Advance Control:

Advanced control requires 3 digital pins of Arduino and lets you control motor speed, direction of rotation, step size and max motor current. Your motor will draw current only while it's rotating and save power when not rotating by entering sleep mode.

Arduino Connection:



Test Code:

```
#define stepPin 8
#define sleepPin 7
#define dirPin 6
int stepDelay=5; /* change it according to your need */

void setup()
{
  pinMode(stepPin, OUTPUT);
  pinMode(sleepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}

void loop()
{
  runMotor(1000);
  delay(1000);
  runMotor(-500); /* '-' for rotating in opposite direction */
  delay(2000);
}
```

```
void runMotor(int steps)
{
    digitalWrite(dirPin, steps>0?HIGH:LOW);
    digitalWrite(sleepPin, HIGH);

    for (int i=0;i<abs(steps);i++)
    {
        digitalWrite(stepPin, HIGH);delay(stepDelay);
        digitalWrite(stepPin, LOW);delay(stepDelay);
    }
    digitalWrite(sleepPin, LOW);
}
```

Do's and Don'ts:

1. Always connect motor to the driver board first. Then provide motor power (Vin & GND). And finally connect to a microcontroller (signal pins). When removing connections, disconnect the motor at the very end.
2. Never disconnect the motor while the driver board is powered on.
3. Prepare the motor correctly (correct motor wire sequence) before connecting to the board to avoid permanent damage of the driver.
4. Ensure correct polarity if the motor voltage input (Vin and GND).
5. Don't rotate the current control pot counter clockwise unless you feel that motor torque is not sufficient.