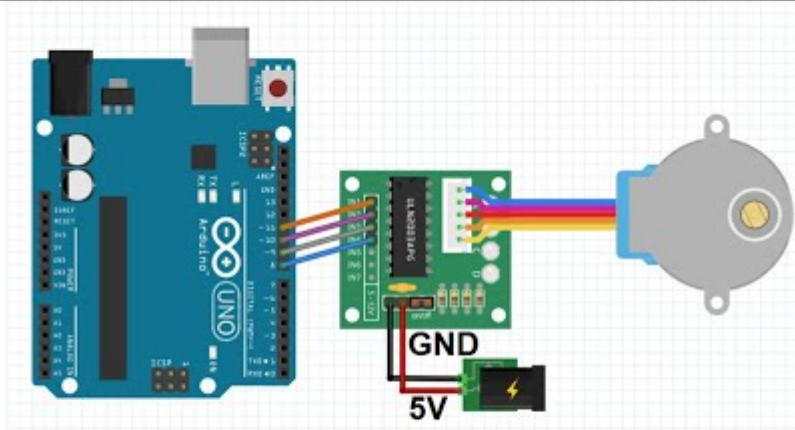


# 28BYJ-48 & ULN2003 – Stepper Motor Control

DIY [diyengineers.com/2021/11/25/28byj-48-uln2003-stepper-motor-control](https://diyengineers.com/2021/11/25/28byj-48-uln2003-stepper-motor-control)

## DIY Engineers

In this post we will go over how to use the 28byj-48 stepper motor with the ULN2003 motor driver. We'll focus on basics, module pins, how to connect to Arduino, programming in Arduino IDE, and testing.



## ULN2003 STEPPER MOTOR CONTROL

Watch Video At: <https://youtu.be/STCctT6JyPg>

### What is a Stepper Motor?

Stepper motors are brushless DC electric motors that rotate its shaft in discrete steps of equal angular rotation. This is possible due to the fact that the motor has coils in groups and can we to energize them in a specific sequence to get one step (i.e. an incremental rotational step). This very nature of stepper motors allow us to move a specific number of steps without having a position/displacement sensor, while maintaining good accuracy on the position and speed of rotation of the motor.

### 28BYJ-48 Stepper Motor

The 28BYJ-48 stepper motor is a great stepper motor for a wide range of applications. It's also a great motor for anyone looking to learn about stepper motors.

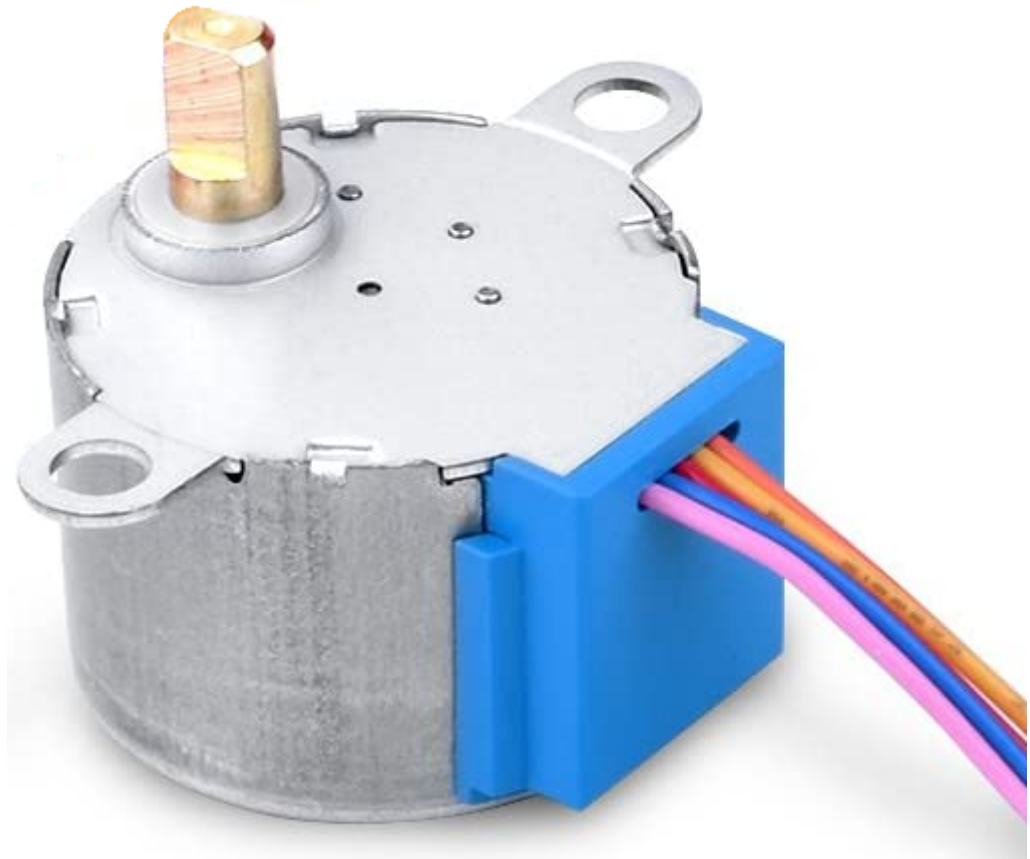
**Rated Voltage:** 5V DC

**Gear Ratio:** 64:1

**Net steps per revolution:** 2048 (~0.1758deg/step)

\*Motor has 32 steps/revolution, and the gear ratio is 64:1, resulting in a net steps/revolution of  $32 \times 64 = 2048$ .

**Coil Type:** Unipolar



**ULN2003 Motor Controller**

---

The ULN2003 motor controller is used to interface your stepper motor (in this case the 28BYJ-48) and your microcontroller (in this case an Arduino UNO). The motor controller accepts four digital inputs from the microcontroller (IN1 – IN4) and the 5 cables coming from the 28BYJ-48 stepper motor. There is also the power input, which can be set to 5V or 12V. For the specific motor in this example we would go with 5V. The motor controller also has 4 step indicator LEDs so you can see the state of the coils of the motor as it rotates. When running fast it can be hard to see.

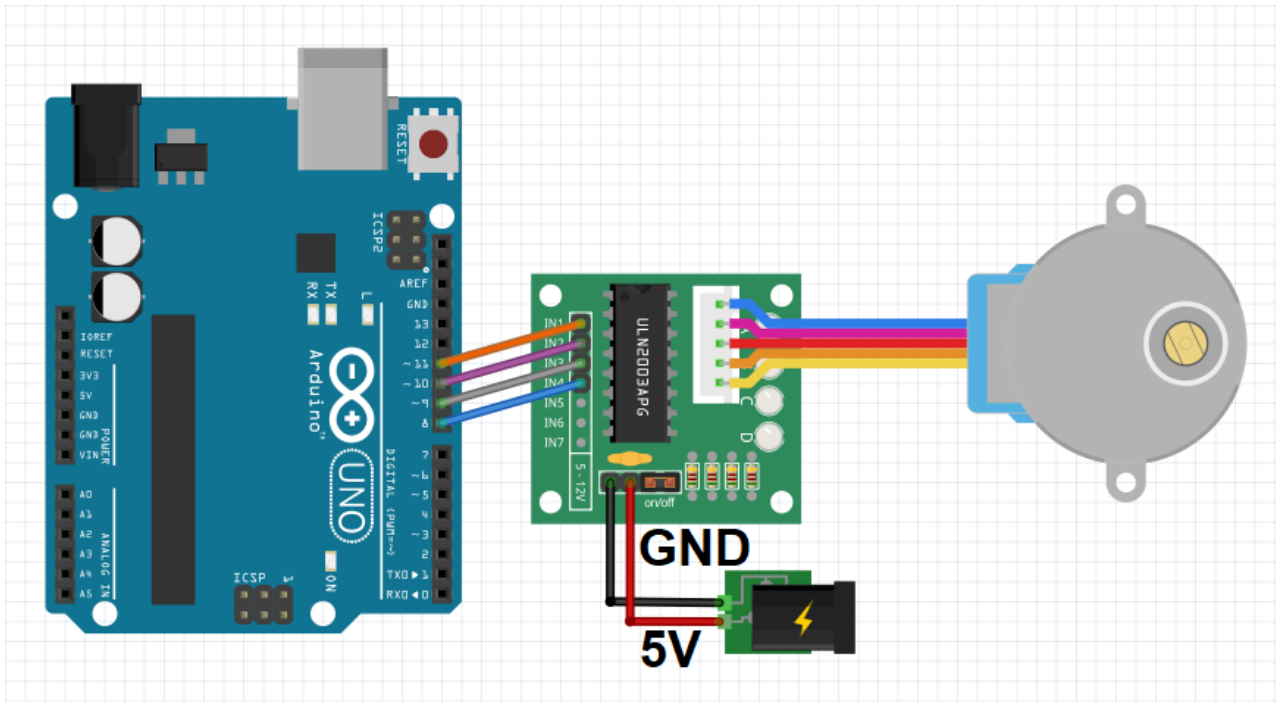


## Connecting ULN2003, 28BYJ-48, and Arduino UNO

**ULN2003 – IN1 through IN 4:** Used to drive motor phases. Connect to Arduino UNO digital input/output pins 8, 9, 10, and 11.

**ULN2003 Power In:** Connect to 5V and ground

**ULN2003 Motor Pins:** Connect the motor cables. The 28BYJ-48 normally comes with a connector that matches the ULN2003 connection terminal. They only go in on a specific orientation, so it's hard to make a mistake.



## ULN2003 / 28BYJI Examples

---

### List of Examples

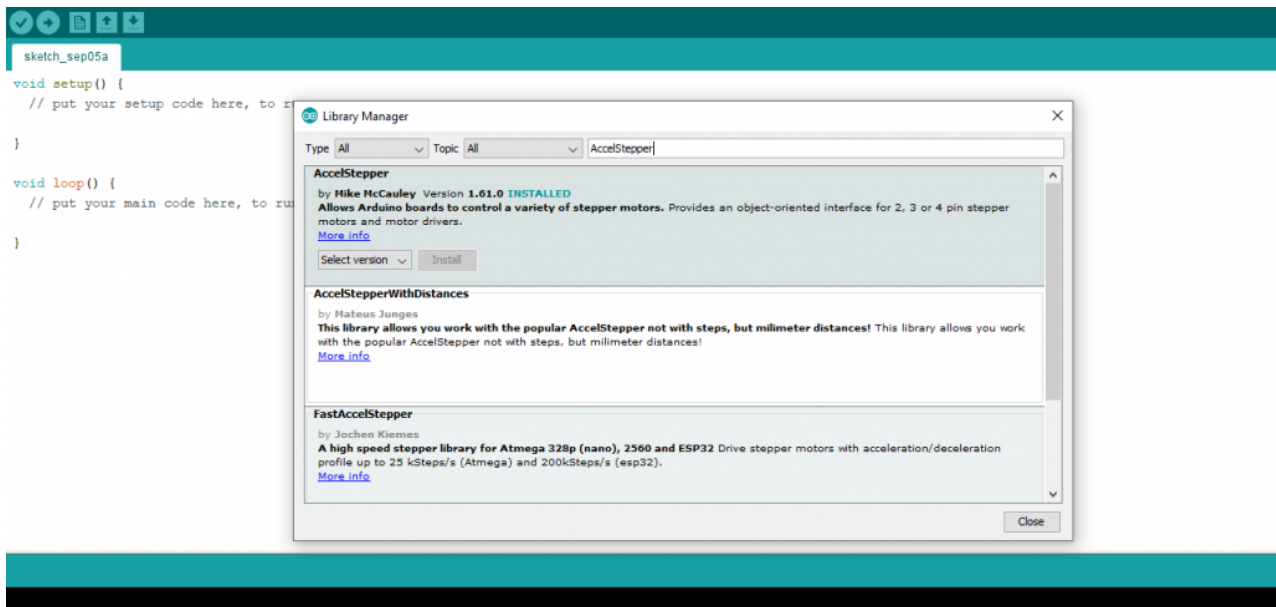
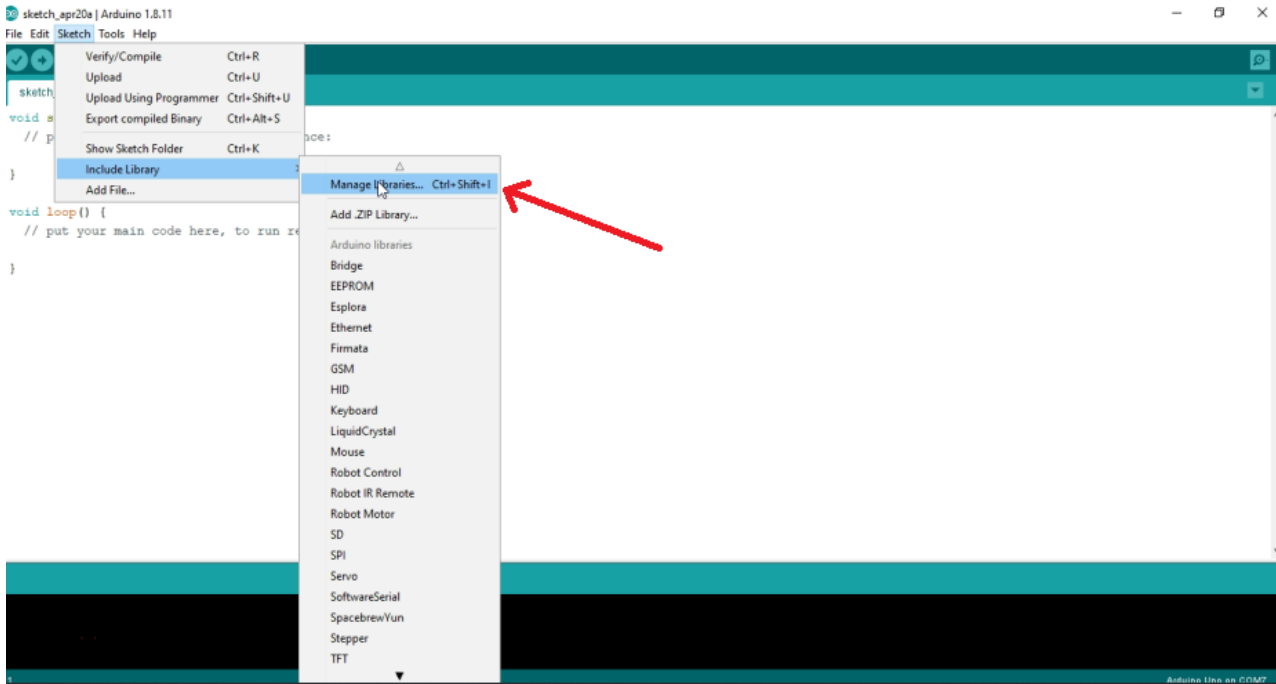
---

- Required Arduino Libraries
- Example 1 – Move from A to B by a specific number of steps with specific speed and acceleration
- Example 2 – Human input control (via rotary encoder)

### Required Arduino Libraries

---

To complete the examples below you will need to install the **AccelStepper** Arduino library (written by Mike McCauley). To install it, open Arduino IDE, go to the top menu, click on “Sketch”, then “Include Library”, “Manage Libraries...”. Then type “AccelStepper”, then click install on the row for the AccelStepper library.



## Example 1 – Move from A to B by a specific number of steps

In this example we will define the speed and acceleration at which we want the stepper motor to rotate and then make it rotate by a specified number of steps, stop, then rotate in the opposite direction in a specified number of steps. Specifically, we will do 3 rotations in one direction (meaning total steps will be the steps per revolution multiplied by 3 revolutions), stop and complete 3 rotations in the opposite direction.

Arduino Code - Move from A to B by a specific number of steps with set a acceleration  
 Arduino

```

1 //Arduino Code - Move from A to B by a specific number of steps with set speed
2 and acceleration
3
4 // Include the Arduino Stepper.h library:
5 #include <AccelStepper.h> //Include the AccelStepper library
6
7 // Define the motor pins:
8 #define MP1 8 // IN1 on the ULN2003
9 #define MP2 9 // IN2 on the ULN2003
10 #define MP3 10 // IN3 on the ULN2003
11 #define MP4 11 // IN4 on the ULN2003
12
13 #define MotorInterfaceType 8 // Define the interface type as 8 = 4 wires * step
14 factor (2 for half step)
15 AccelStepper stepper = AccelStepper(MotorInterfaceType, MP1, MP3, MP2,
16 MP4); //Define the pin sequence (IN1-IN3-IN2-IN4)
17 const int SPR = 2048; //Steps per revolution
18
19 void setup() {
20   stepper.setMaxSpeed(1000); //Set the maximum motor speed in steps per
21   second
22   stepper.setAcceleration(200); //Set the maximum acceleration in steps per
23   second^2
24 }
25
26 void loop() {
27   stepper.moveTo(3*SPR); //Set the target motor position (i.e. turn motor for 3 full
28   revolutions)
29   stepper.runToPosition(); // Run the motor to the target position
30   delay(1000);
31   stepper.moveTo(-3*SPR); //Same as above: Set the target motor position (i.e. turn
32   motor for 3 full revolutions)
33   stepper.runToPosition(); // Run the motor to the target position
34   delay(1000);
35 }

```

## **Example 2 – Move the stepper motor by controlling with a rotary encoder**

---

As the example title says, in this example we will control the rotation of a stepper motor using a rotary encoder. For details on how to use a rotary encoder, please take a [look at my earlier post](#).

Arduino Code - Move stepper motor based on movement of rotary encoder  
 Arduino

```

1 //Arduino Code - Move stepper motor based on movement of rotary encoder
2
3 // Include the Arduino Stepper.h library:
4 #include <AccelStepper.h> //Include the AccelStepper library
5
6 // Define the motor pins:
7 #define MP1 8 // IN1 on the ULN2003
8 #define MP2 9 // IN2 on the ULN2003

```

```

9  #define MP3 10 // IN3 on the ULN2003
10 #define MP4 11 // IN4 on the ULN2003
11
12 #define MotorInterfaceType 8 // Define the interface type as 8 = 4 wires * step
13 factor (2 for half step)
14 AccelStepper stepper = AccelStepper(MotorInterfaceType, MP1, MP3, MP2,
15 MP4); // Define the pin sequence (IN1-IN3-IN2-IN4)
16 const int SPR = 2048; // Steps per revolution
17
18 int DT = 4; // GPIO #4-DT on encoder (Output B)
19 int CLK = 5; // GPIO #5-CLK on encoder (Output A)
20
21 int counter = 0;
22 int angle = 0;
23 int aState;
24 int aLastState;
25
26 void setup() {
27   stepper.setMaxSpeed(1000); // Set the maximum motor speed in steps per
28   second
29   stepper.setAcceleration(1000); // Set the maximum acceleration in steps per
30   second^2
31
32   pinMode(CLK, INPUT_PULLUP);
33   pinMode(DT, INPUT_PULLUP);
34   aLastState = digitalRead(CLK);
35 }
36
37 void loop() {
38   aState = digitalRead(CLK);
39   // Encoder rotation tracking
40   if (aState != aLastState) {
41     if (digitalRead(DT) != aState) {
42       counter++;
43       angle++;
44     }
45     else {
46       counter--;
47       angle--;
48     }
49     if (counter >= 100) {
50       counter = 100;
51     }
52     if (counter <= -100) {
53       counter = -100;
54     }
55     aLastState = aState;
56   }

```

stepper.moveTo(200\*counter); // Set the target motor position (multiply by whatever number you want to define how much to move per encoder pulse)

```

   stepper.runToPosition(); // Run the motor to the target position
}

```

