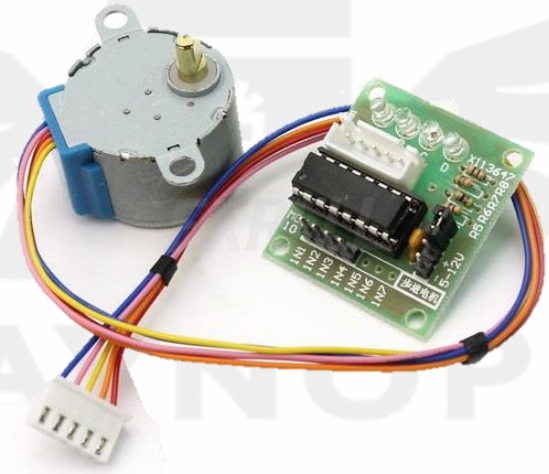
LAUNCH THE PROJECT - 12

Stepper Motor

(28BYJ-48 + ULN2003 Driver Board)

On AYNOP® UNO Launchpad Kit



Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 19-9-2025

Table of Contents

1. **Overview** 3
2. **Components Required** 3
3. **Software Required** 3
4. **Hardware Setup** 4
   4.1 **Wiring Diagram** 4
5. **Principle – How It Works** 5
6. **Procedure – Steps to Run** 6
   6.1 **How to Use the Serial Monitor:** 7
7. **Expected Output** 7
8. **Code** 8
   8.1 **Function References** 8
9. **Troubleshooting Tips** 9
10. **License** 10
11. **Support & Feedback** 10

1. Overview

This project introduces the **28BYJ-48 stepper motor** and its **ULN2003 driver board**. Using Arduino's built-in **Stepper library**, the motor will.

- Rotate one full revolution clockwise.
- Pause briefly.
- Rotate one full revolution anti-clockwise.

This experiment helps you to understand stepper motor control, direction reversal, and the concept of steps per revolution.

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- 28BYJ-48 stepper motor
- ULN2003 driver board
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended).
- Built-in **Stepper library** (included with Arduino IDE)
- No additional libraries required**.**

Note:

We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the components to drive the **Stepper motor**.

4.1 Wiring Diagram

- 28BYJ-48 stepper motor white connector → ULN2003 driver board.
- ULN2003 **IN1 → D8**
- ULN2003 **IN2 → D9**
- ULN2003 **IN3 → D10**
- ULN2003 **IN4 → D11**
- ULN2003 **VCC → 5V**
- ULN2003 **GND → GND**

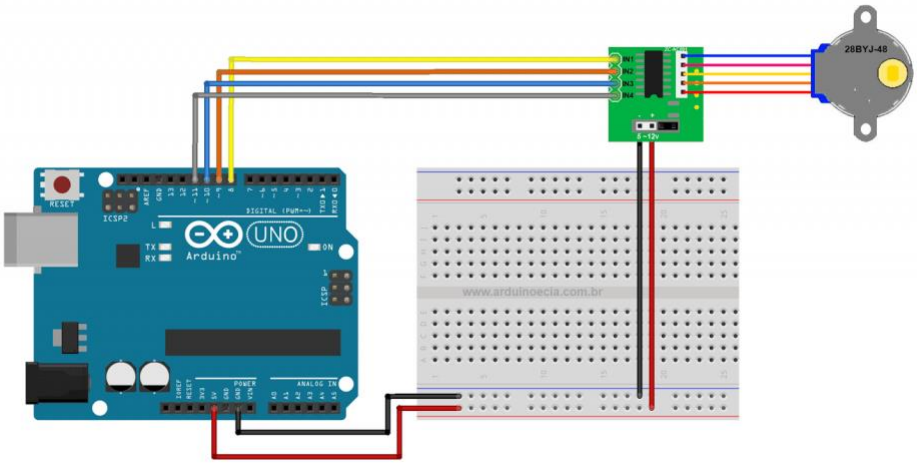


Figure 4.1 – Wiring diagram for Stepper Motor project

Tip: Always disconnect the USB cable before making or changing hardware connections.

5. Principle – How It Works

A **stepper motor** moves in small steps instead of continuous rotation.

- The **28BYJ-48 stepper motor** requires approximately **2048 steps for one full revolution** in full-step mode.
- The ULN2003 driver board provides current amplification, allowing Arduino to safely drive the motor coils.
- The Arduino **Stepper library** simplifies control by letting you specify the number of steps or revolutions and the rotation direction.

In this project, the motor:

- Completes one clockwise revolution.
- Pauses.
- Then completes one counter-clockwise revolution.

⚠ **Important Note on Speed:**

- The **28BYJ-48 stepper motor** is designed for **precision, not speed**.
- At the default setting in this project (myStepper.setSpeed(10)), the motor runs at about **10 RPM (revolutions per minute)**.
- In practice, this motor can usually handle up to **15–20 RPM** before it starts skipping steps or losing torque.
- If you increase the speed in code, test gradually and avoid pushing it too high, otherwise the motor may just vibrate instead of turning

6. Procedure – Steps to Run

1. **Build the Circuit**
   a. Connect the stepper motor to the ULN2003 driver board
   b. Connect the ULN2003 driver as shown in the **Wiring Diagram (Figure 4.1)**.

2. **Connect the Board**
   a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. **Open the Project Code**
   a. Simply **double-click** the file *12_Stepper_Motor.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. **Confirm Board Selection**
   a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
   b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
   c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

5. **Upload the Code**
   a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
   b. Wait until the console displays "**Done uploading**."

6. **Observe the Behaviour**
   a. The stepper motor will rotate **one full turn clockwise**, pause, and then rotate **one full turn counter-clockwise**.
   b. This sequence will repeat continuously.
   c. Observe real time direction of stepper motor **on** Serial Monitor.

6.1 How to Use the Serial Monitor:

The Serial Monitor is a **tool in the Arduino IDE** that lets you see text/data sent by the microcontroller (MCU) over the USB connection.

- In Arduino IDE, go to **Tools → Serial Monitor**, or press **Ctrl + Shift + M**.
- A new window will open showing real-time data from your UNO R4 Minima.
- At the bottom-right of the Serial Monitor window, set the baud rate to **9600**. This must match the *Serial.begin*(9600) in the code.

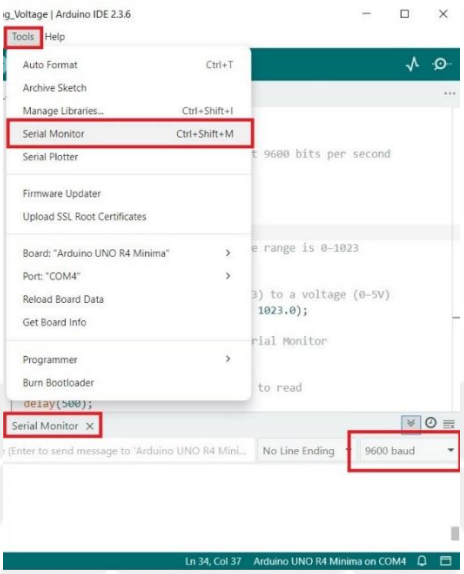


Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

Tip: If the Serial Monitor shows nothing, check that:

- The correct **COM port** is selected.
- The baud rate is set to **9600**

7. Expected Output

- The stepper motor physically rotates one full revolution clockwise, pauses, then rotates back anti-clockwise and This sequence will repeat continuously.
- Serial Monitor shows messages about the current direction:

```
AYNOP: Stepper Motor Demo
Rotating clockwise...
Rotating  anti clockwise...
```

8. Code

The source code for this project is included in the downloaded folder:

📂 uno-launchpad-kit/01_Basic_Projects/12_Stepper_Motor/12_Stepper_Motor.ino

👉 **Tip:**

- To open the project, simply **double-click the .ino file**. If the Arduino IDE is installed, it will launch automatically and load the code.
- If you **haven't installed the Arduino IDE yet**, please refer to:
  📂 uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima
  to **download and install it.**

8.1 Function References

- setup() – runs once when the board is powered on or reset.
- loop() – runs continuously after setup() finishes.
- Stepper(steps, pin1, pin2, pin3, pin4) – create Stepper object.
- setSpeed(rpm) – set motor speed in RPM.
- step(steps) – move the motor the specified number of steps.
- Serial.begin(baudrate) – start serial communication.
- Serial.print() / Serial.println() – print to Serial Monitor.
- delay(ms) – pause program for given milliseconds.

📚 **For more details and advanced usage, visit:**

🔗 Arduino Language Reference — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **Motor not moving?**
  - Ensure ULN2003 VCC → 5V and GND → GND.
  - Double-check IN1–IN4 wiring order.

- **Motor vibrating but not rotating?**
  - Ensure speed is set **at or below 20 RPM** in the code; higher values may cause vibration or missed steps.
  - The pin sequence may be incorrect.
  - Check that ULN2003 input wiring matches the code.

- **Motor rotates in wrong direction?**
  - Swap wiring order, or invert step direction in code (step(-STEPS_PER_REV)).

- **Motor speed too slow/fast?**
  - Adjust myStepper.setSpeed(rpm) value in the code.
  - Higher RPM may cause missed steps.

- **Upload error in Arduino IDE?**
  - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
  - Check that the correct **COM port** is chosen.

- **Board not detected via USB?**
  - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
  - Try reconnecting the cable or using a different USB port.

- **Board not listed in Arduino IDE?**
  - If you don't see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.
  - To fix this, follow the installation steps in:
    uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima

Tip: If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

Email: *support@aynop.com*

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.