

## LAUNCH THE PROJECT - 18

# Joystick Module

*(Basic Reading)*

*On AYNOP® UNO Launchpad Kit*



*Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 03-9-2025*

## 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 demonstrates how to use a **joystick module** with Arduino® UNO R4 Minima. The joystick has two potentiometers (for X and Y axes) and one push button (SW).

You will learn to:

- Connect the joystick module to Arduino.
- Read analog values for X and Y axes.
- Detect button press using digital input with internal pull-up.
- Print results to the Serial Monitor.

## 2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- Joystick module (5 pins: VCC, GND, VRx, VRy, SW)
- Breadboard
- Jumper wires

## 3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- No additional library is 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 **joystick** module.

### 4.1 Wiring Diagram

- Joystick **VCC** → **5V**.
- Joystick **GND** → **GND**.
- Joystick **VRx** → **A1** (Arduino).
- Joystick **VRy** → **A0** (Arduino)
- Joystick **SW** → **D9**(Arduino)

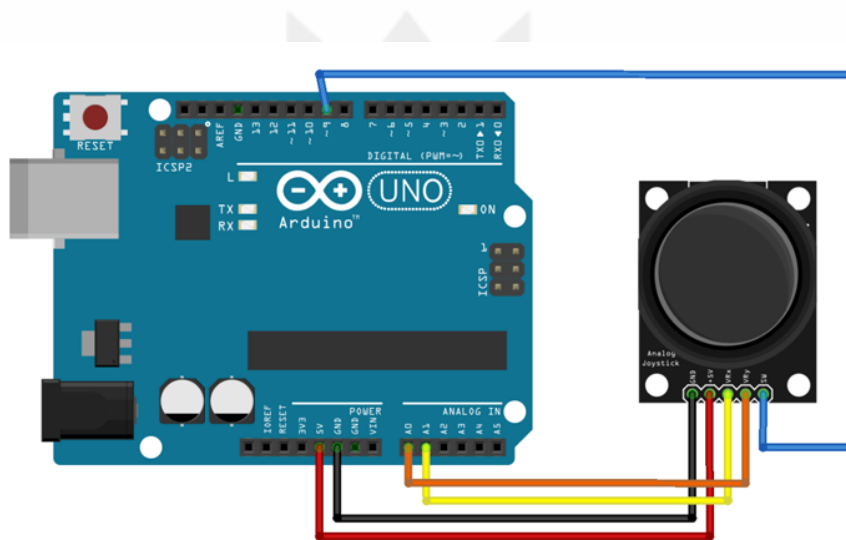


Figure 4.1 – Wiring diagram for joystick module with Arduino UNO R4 Minima

#### 💡 Tip:

- Always disconnect the USB cable before making or changing hardware connections.
- The SW pin uses Arduino's internal pull-up, so button reads **LOW** when pressed.

## 5. Principle – How It Works

The joystick module contains two potentiometers and a push button:

- **X-axis (VRx):** Varies voltage as you move left/right. Arduino reads this on A1.
- **Y-axis (VRy):** Varies voltage as you move up/down. Arduino reads this on A0.
- **SW button:** Connects to GND when pressed. With Arduino configured as INPUT\_PULLUP, the pin reads **HIGH** when released and **LOW** when pressed.

Arduino continuously reads these signals and prints them to the Serial Monitor.



## 6. Procedure – Steps to Run

### 1. Build the Circuit

- a. Connect the joystick 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 *18\_Joystick\_Module.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. Open the Serial Monitor (set baud rate to **9600**).
- b. Move the joystick: observe **X** and **Y** values change between 0–1023.
- c. In center position, values are around 512 (approximate).
- d. Press the joystick: the Serial Monitor shows **PRESSED**
- e. Release the joystick: it shows **RELEASED**.

## 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.

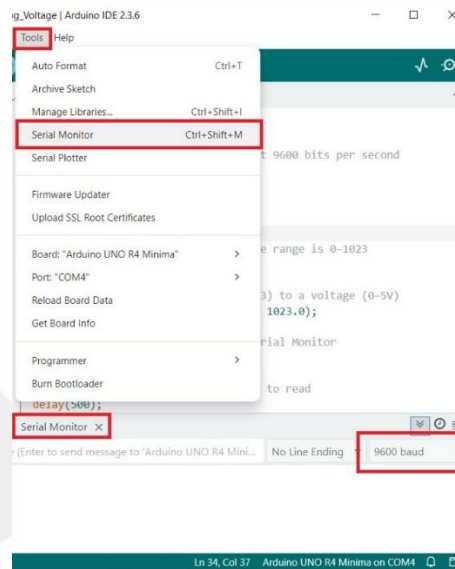


Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

**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 Serial Monitor will print X, Y values and button state of joystick:

```


AYNOP: Joystick Started
X: 516 | Y: 500 | Button: RELEASED
X: 516 | Y: 500 | Button: RELEASED
X: 515 | Y: 499 | Button: RELEASED
X: 514 | Y: 500 | Button: PRESSED
X: 515 | Y: 500 | Button: PRESSED
X: 515 | Y: 500 | Button: RELEASED
    
```

## 8. Code

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

 `uno-launchpad-kit/01_Basic_Projects/18_Joystick_Module/18_Joystick_Module.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.
- `analogRead(pin)` — read analog input (0–1023).
- `digitalRead(pin)` — read HIGH/LOW from a pin.
- `pinMode(pin, mode)` — configure a pin as input or output.
- `Serial.begin(baud)` — start serial communication.
- `Serial.print()` / `Serial.println()` — print text to Serial Monitor.
- `delay(ms)` — pause program.

 For more details and advanced usage, visit:

 [Arduino Language Reference](#) — The official guide for all Arduino functions.

## 9. Troubleshooting Tips

- **No output on Serial Monitor?**
  - Check the baud rate (must be 9600).
  - Ensure Serial Monitor is open after upload.
- **Values stuck at 0 or 1023?**
  - Check wiring of VRx/VRy pins and confirm VCC/GND.
- **Button always PRESSED or RELEASED?**
  - Verify SW wiring (D9 and GND).
- **Noise or fluctuating readings?**
  - Analog values may vary slightly; this is normal.
- **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

This content (source code and documentation) is licensed under **Creative Commons BY-NC-SA 4.0**.  
© 2025 AYNOP. You may use, modify, and share for personal and educational purposes only.  
Commercial use or redistribution without prior written permission is strictly prohibited.  
Refer to the **LICENSE** file for complete details.

## 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](mailto: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.

