LAUNCH THE PROJECT - 04

Passive Buzzer Sound

On AYNOP® UNO Launchpad Kit

Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 11-9-2025

Table of Contents

1. Overview

This project introduces the **passive buzzer**, which can generate different tones by varying the frequency of a square wave signal from the Arduino® UNO R4 Minima.

You will learn to:

- Connect and use a **passive buzzer** with Arduino.
- Generate sound using the **tone()** function**.**
- Understand frequency (Hz) and duration in sound generation.

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- 1 × Passive Buzzer
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- 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 for the **Passive Buzzer Sound** project.
It includes a **Wiring Diagram** and a **Circuit Schematic**.

4.1 Wiring Diagram

- Connect one pin of the passive buzzer to Arduino **D8**.
- Connect the other pin of the buzzer to **GND.**

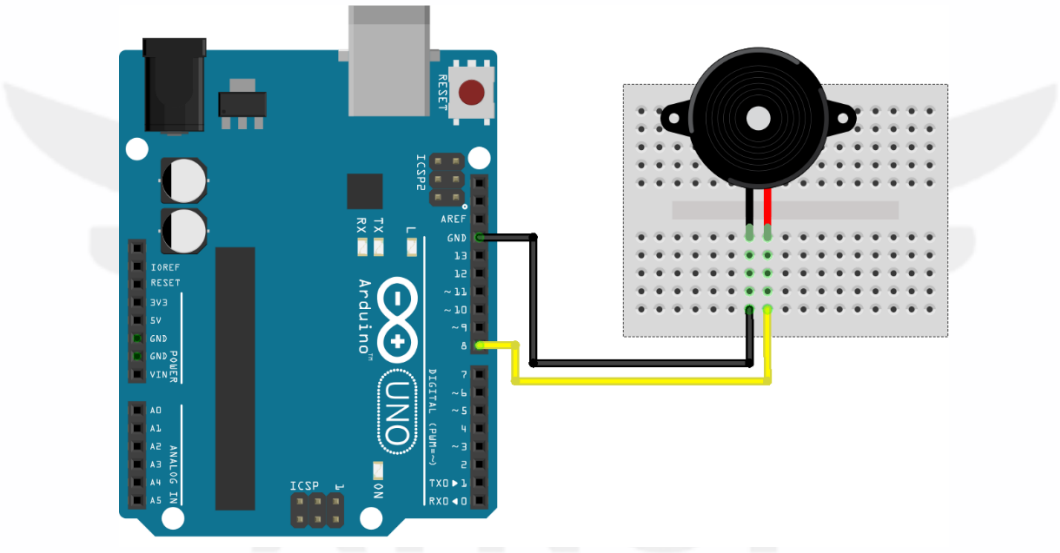

Figure 4.1 – Wiring diagram for Passive Buzzer Sound project

Tip: Unlike an **active buzzer**, which has fixed polarity and produces only one tone, a **passive buzzer** is not polarity-sensitive and can generate a wide range of tones depending on the frequency of the signal.

4.2 Circuit Schematic

- **BZ1 = Passive buzzer**
  - Pin 1 → Arduino D8
  - Pin 2 → GND

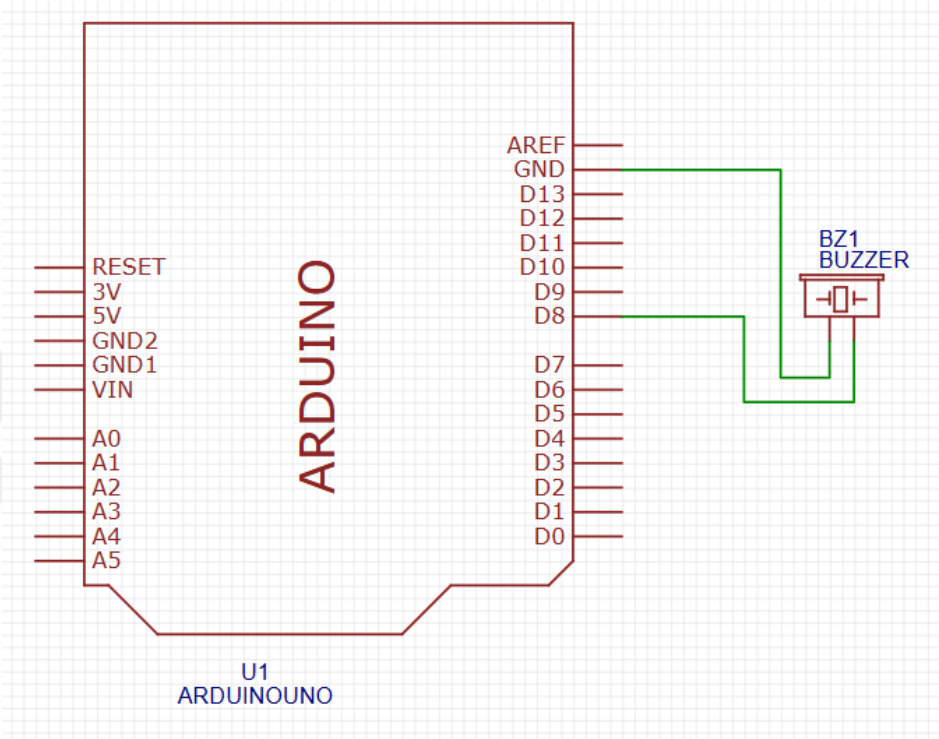

Figure 4.2 – Circuit schematic for Passive Buzzer Sound project

5. Principle – How It Works

A **passive buzzer** does not produce sound on its own. It requires a driving signal (a square wave) from the Arduino.

- The Arduino's **tone()** function generates a square wave at a given frequency (measured in Hz).
- The buzzer vibrates at that frequency, producing sound.
- Example frequencies:
  - 262 Hz → Note C4
  - 294 Hz → Note D4
  - 330 Hz → Note E4
- The **delay()** function controls how long each note is played.
- The **noTone()** function stops sound output.

6. Procedure – Steps to Run

1. **Build the Circuit**
   a. Assemble the buzzer connections 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 *04_Passive_Buzzer_Sound.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 buzzer will play three tones (C, D, and E), each lasting 500 ms.
   b. After a short pause, the sequence repeats continuously**.**

7. Expected Output

- The buzzer plays three musical notes in sequence:
  - **C4 (262 Hz) → D4 (294 Hz) → E4 (330 Hz)**
- Each note lasts for 500 ms, with short pauses in between.
- After a pause, the melody repeats.

8. Code

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

📂 uno-launchpad-kit/01_Basic_Projects/04_Passive_Buzzer_Sound/04_Passive_Buzzer_Sound.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.
- tone(pin, frequency, duration) – generates a square wave of the given frequency (Hz) on a pin for a set duration (ms).
- noTone(pin) – stops the sound on the specified pin.
- delay(ms) – pauses the program for the specified number of milliseconds.

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

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

9. Troubleshooting Tips

- **No sound from buzzer?**
  - Ensure one leg is connected to Arduino D8 and the other to GND.
  - Verify that you are using a **passive buzzer**, not an active buzzer.

- **Sound is too faint or distorted?**
  - Check that the buzzer is firmly connected to the breadboard.
  - Reduce jumper wire length for cleaner signal.

- **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?**
  - 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.