



# Arduino Joystick Control: LED & Buzzer

Join us as we delve into a beginner-friendly Arduino project that leverages the power of joysticks to control LEDs and buzzers.



by **Shashwat Singh**

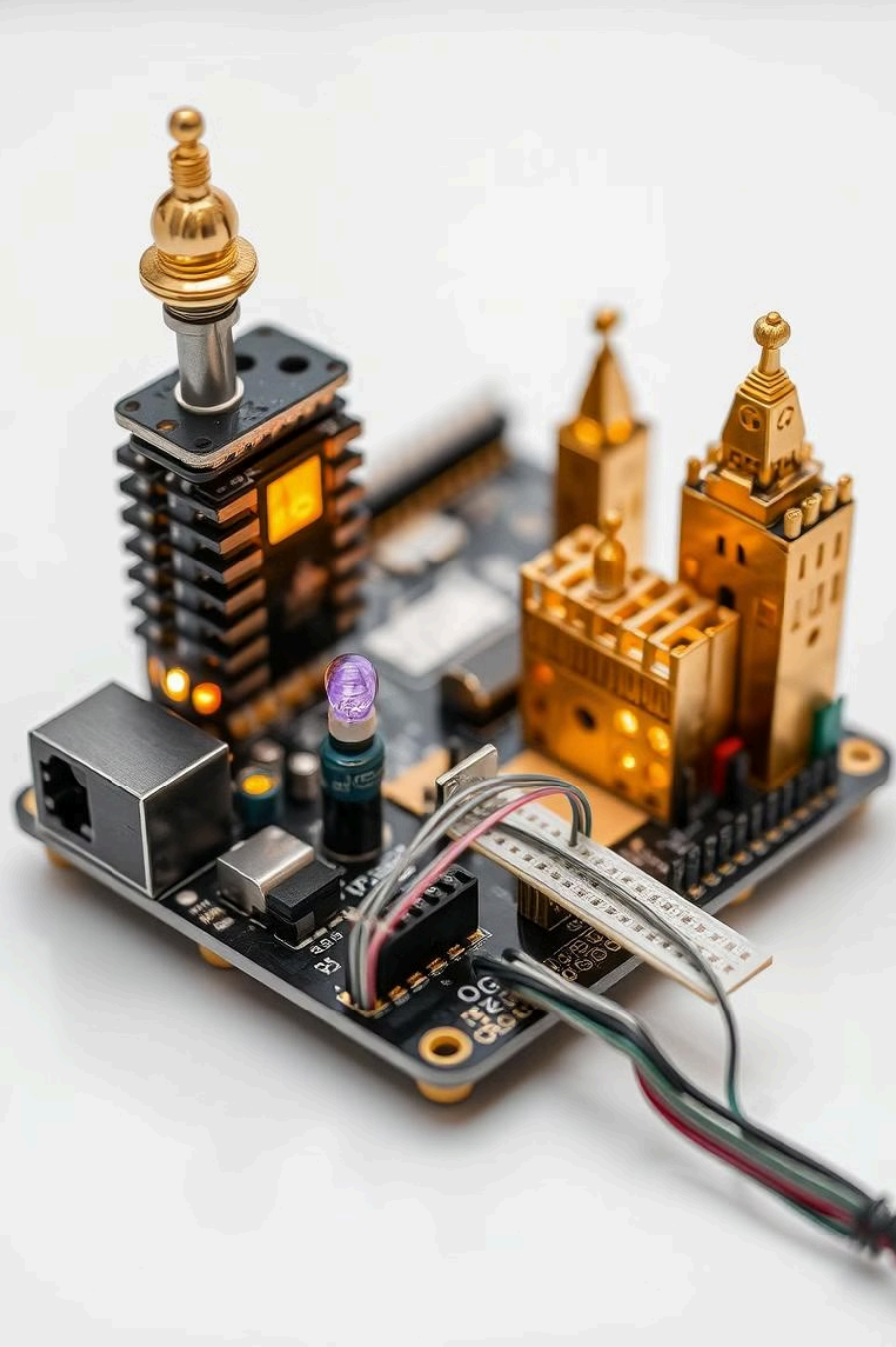
# Project Overview

## Objective

Control an LED's brightness and a buzzer's sound frequency using the joystick's X and Y axes.

## Benefits

Learn how to read analog inputs, manipulate outputs, and create interactive projects with Arduino.



# Essential Components

## Arduino Uno

The microcontroller brain of our project. It processes the joystick's signals and controls the LED and buzzer.

## Joystick Module

A versatile input device with two analog axes (X and Y) and a push-button switch. It translates your movements into electrical signals.

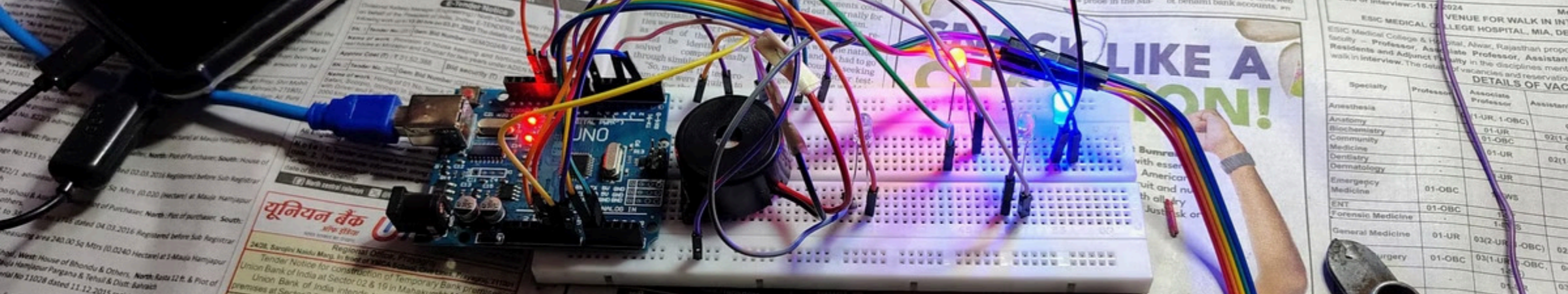
## LED, Resistor, and Buzzer

The output devices we'll be controlling. The LED provides visual feedback, while the buzzer generates sound.

## Breadboard and Jumper Wires

The platform for connecting all components and forming our interactive circuit.





# Connecting the Joystick



## VRx to Analog Pin A0

Connect the joystick's horizontal axis (VRx) to the Arduino's analog input pin A0.



## VRy to Analog Pin A1

Connect the joystick's vertical axis (VRy) to the Arduino's analog input pin A1.



## SW to Digital Pin 2

Connect the joystick's switch (SW) to the Arduino's digital pin 2.

# Reading Analog Inputs

1

Use the `analogRead()` function to get values from the joystick's X and Y axes.

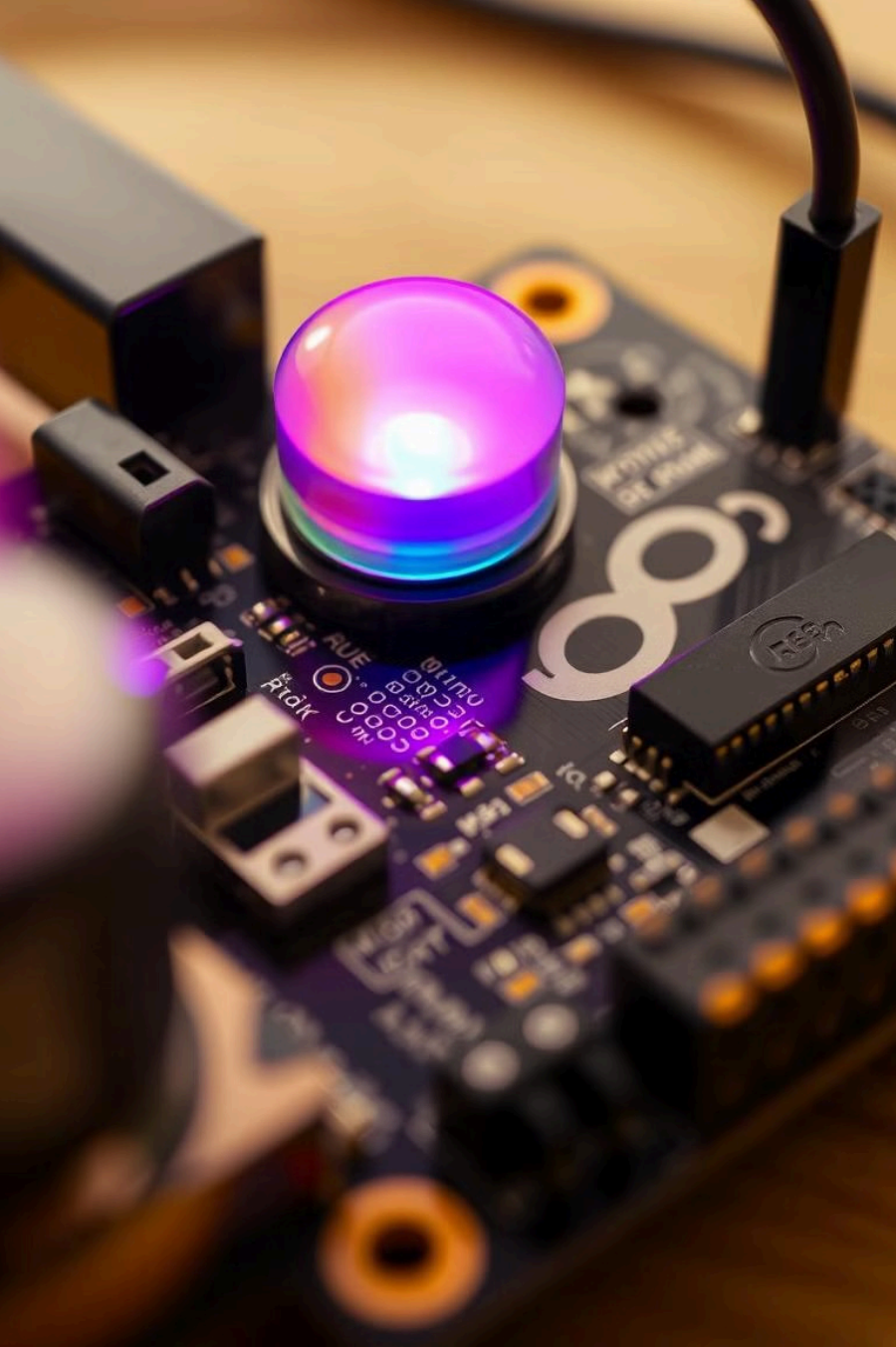
2

These values range from 0 to 1023, representing the full range of motion of the joystick.

3

We'll use these raw values to map them to the desired brightness and frequency of the LED and buzzer.

```
Value analogRead()
1 {Farnesct: Adddino[DE 49143]
2   Ceatlcpmmeet Farbylapd)
3   fortantingst farnesanerty, vetldeing joystick)
4     Nots hot <
6     Resilnot vging cal joyvl((992.ac1)
10    Real but viing cal joysstck, 'inster= walef2)
17    cood)
19    Pnelingt telermfatory, pol;
17    Resilontt walngice, amped (011440)
18    Real hot wlist,buufffy,vissel; Sofares to (09d2 4009;
23    Resillogt joriaraiserire:
27    Resileinde Ar, <andue =lop
23    Resilofine juning ferericat fisster= 49020r;
23    Real Agt-vilst-aurzer( = (12220019);
23    coming
23    restingst-Ranqinalceads usl)>
23 }
```



# Controlling the LED

1

## Mapping Values

Map the joystick's X-axis value to the LED's brightness.

2

## Analog Write

Use the `analogWrite()` function to control the LED's brightness based on the mapped value.

3

## Visual Feedback

The LED will brighten as you push the joystick to the right and dim as you push it to the left.

# Triggering the Buzzer

1

## Frequency Mapping

Map the joystick's Y-axis value to the buzzer's frequency.

2

## Tone Function

Use the `tone()` function to generate sound at the mapped frequency.

3

## Sound Output

The buzzer will produce a higher-pitched sound as you push the joystick up and a lower-pitched sound as you push it down.



# Project Demo and Next Steps

Witness the magic of your creation as the LED illuminates and the buzzer beeps in response to the joystick's movement.

