**Jetson Nano LED Control using Sockets and Tkinter GUI**

# Overview:

This project allows users to remotely control an LED connected to a Jetson Nano using a client-server architecture. The Jetson Nano acts as a server, receiving commands from a client device that has a graphical user interface (GUI) built with Tkinter. The communication between the server and client is handled using Python sockets.

# Features :

* Remote LED control over a local network
* Simple client-side GUI with a toggle button
* Uses Python sockets for communication
* Works with Jetson Nano's GPIO pins

## **Components Required**

* Jetson Nano
* LED
* 220Ω resistor
* Breadboard and jumper wires
* Tkinter (for GUI on the client side)
* Jetson.GPIO (for controlling the GPIO on the Jetson Nano)

## **System Architecture**

1. The **Jetson Nano** acts as a server, listening for incoming commands ("on" or "off").
2. The **client device** (another computer or Raspberry Pi) runs a GUI application to send LED control commands.
3. Communication happens over a **local network** using Python sockets.

**Circuit**

1. Connect the **long leg (anode) of the LED** to a GPIO pin (e.g., GPIO 12) on the Jetson Nano through a **330Ω resistor**.
2. Connect the **short leg (cathode)** to the **ground (GND)** pin of the Jetson Nano.

## **Packages need to be installed:**

Install the following packages on Jetson nano

1. Pip newer version
2. Jetson.GPIO

install the following package on client system

1. sudo apt install python3-tk

Connect the server and client in same local network

## **Server-side Code (Jetson Nano)**

**import** Jetson.GPIO **as** GPIO

**import** socket

**import** time

# Initialize GPIO

GPIO.setmode(GPIO.BOARD)

GPIO.setup(12, GPIO.OUT)

# Create a socket object

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

​

# Get the IP address of the Jetson Nano

host = '0.0.0.0'  # Listen on all available network interfaces

port = 12345  # Port to bind to

​

# Bind the socket to the address

server\_socket.bind((host, port))

​

# Listen for incoming connections

server\_socket.listen(5)

​

print(f"Server is running on {host}:{port}")

​

**while** **True**:

   # Accept a new connection

   client\_socket, client\_address = server\_socket.accept()

   print(f"Connection from {client\_address}")

​

**try**:

       # Receive the data sent by the client

       data = client\_socket.recv(1024).decode()

​

**if** data == 'on':

           GPIO.output(12, GPIO.HIGH)  # Turn LED on

           print("LED ON")

**elif** data == 'off':

           GPIO.output(12, GPIO.LOW)  # Turn LED off

           print("LED OFF")

**else**:

           print(f"Invalid command: {data}")

​

**except** Exception **as** e:

       print(f"Error: {e}")

**finally**:

       client\_socket.close()

​

# Clean up GPIO on exit

GPIO.cleanup()

## **Client-side Code (GUI with Tkinter):**

**import** socket

**import** tkinter **as** tk

​

# Function to send command to the server

**def** send\_command(command):

**try**:

       # IP address of the Jetson Nano (Server)

       host = '192.168.1.4'  # Replace with your Jetson Nano's IP address

       port = 12345  # Port used by the server

​

       # Create a socket object

       client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

​

       # Connect to the server

       client\_socket.connect((host, port))

​

       # Send the command (either 'on' or 'off') to control the LED

       client\_socket.send(command.encode())

       client\_socket.close()

**except** Exception **as** e:

       print(f"Error: {e}")

​

# Function to toggle LED state

**def** toggle\_led():

**if** toggle\_button["text"] == "Turn LED ON":

       send\_command("on")  # Send 'on' command

       toggle\_button.config(text="Turn LED OFF")  # Update button text

**else**:

       send\_command("off")  # Send 'off' command

       toggle\_button.config(text="Turn LED ON")  # Update button text

​

# Create the GUI window

root = tk.Tk()

root.title("LED Control")

​

# Add a toggle button to control the LED

toggle\_button = tk.Button(root, text="Turn LED ON", command=toggle\_led, height=2, width=20)

toggle\_button.pack(pady=20)

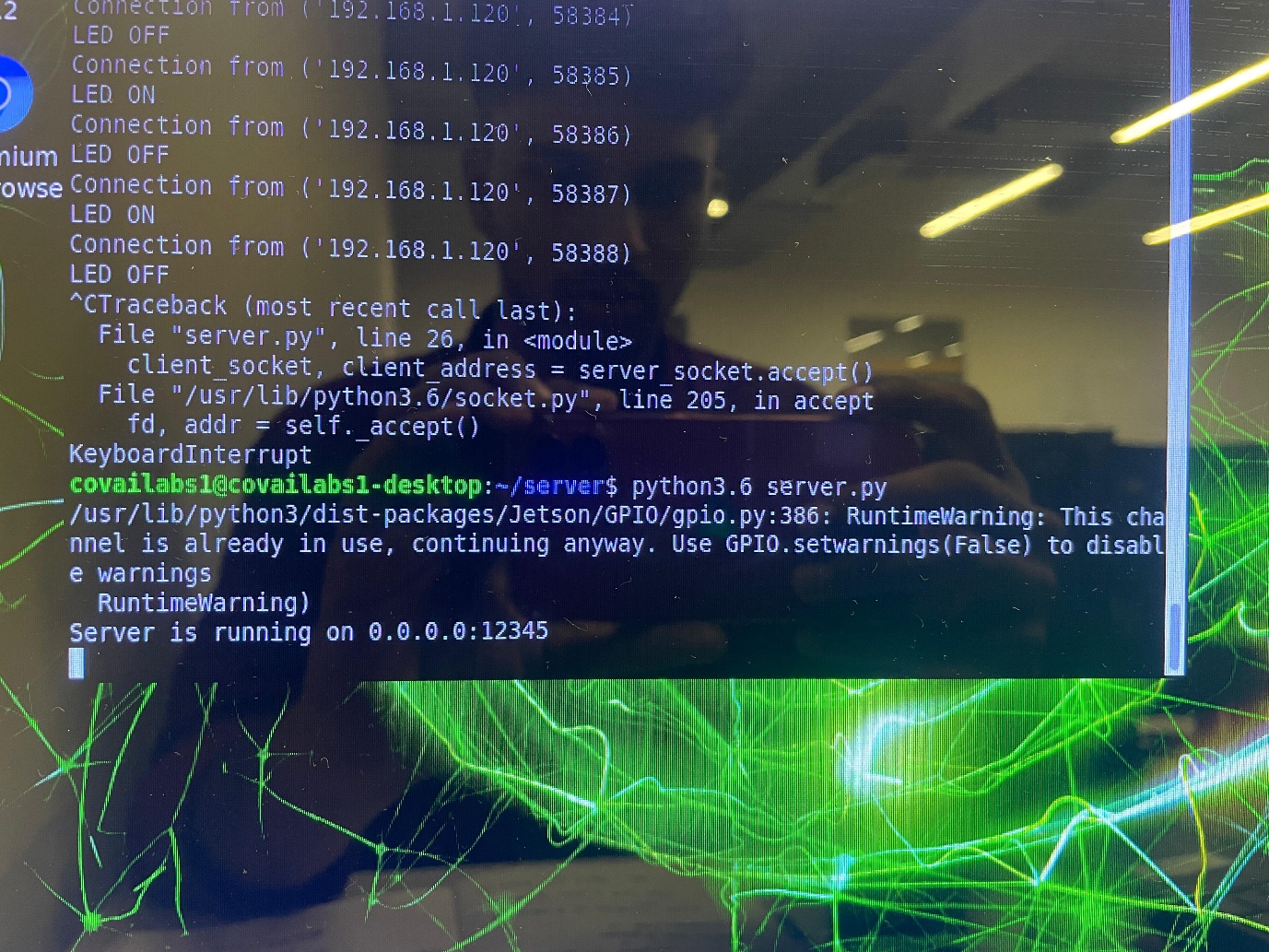
​

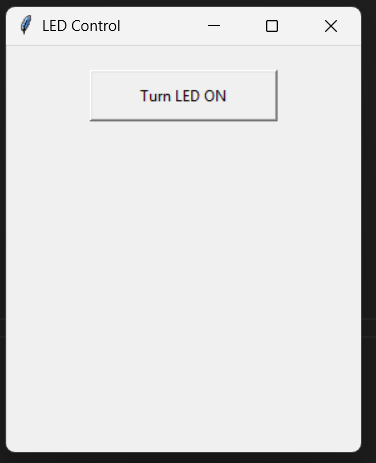
# Run the GUI loop

root.mainloop()

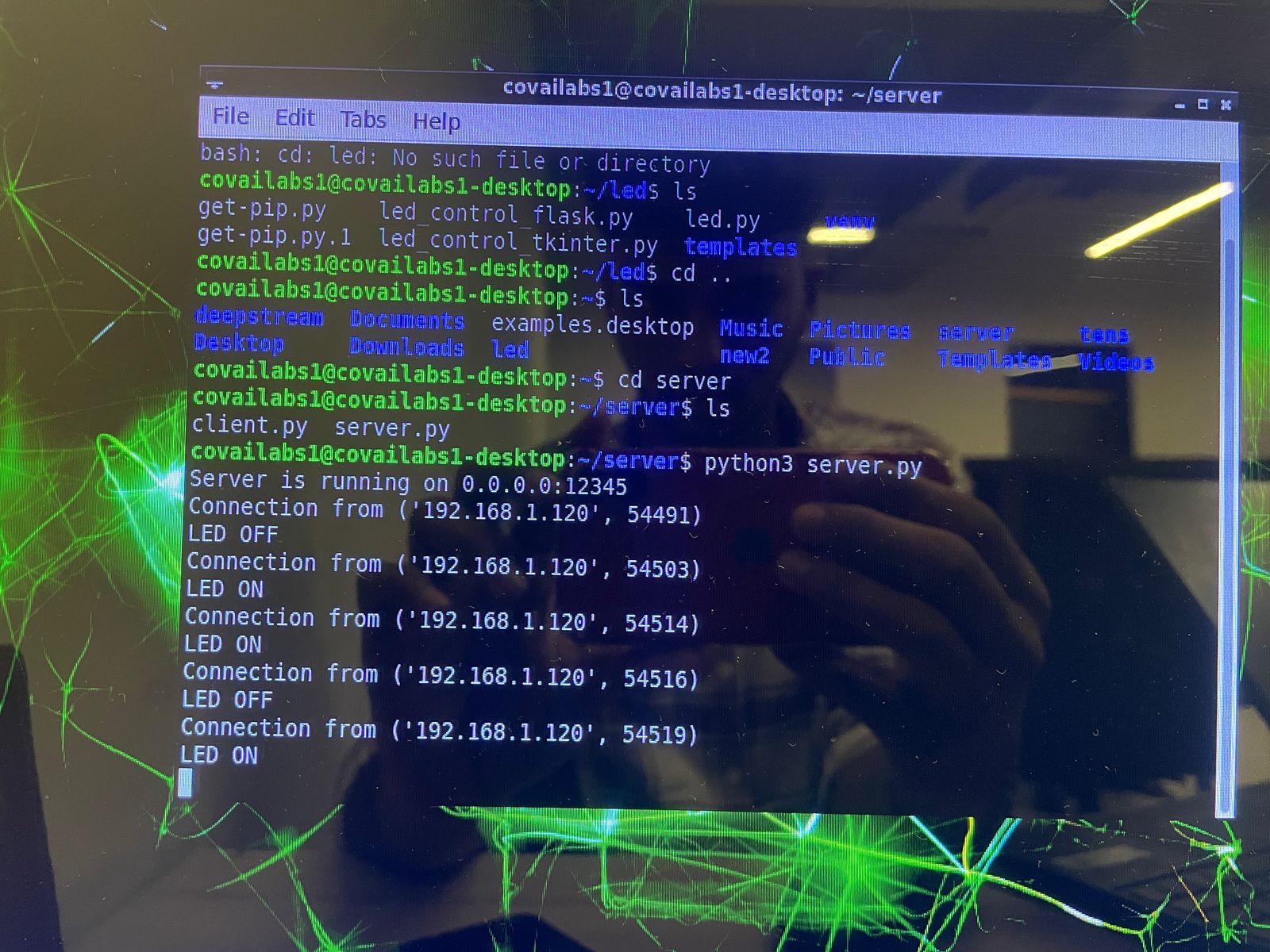
​

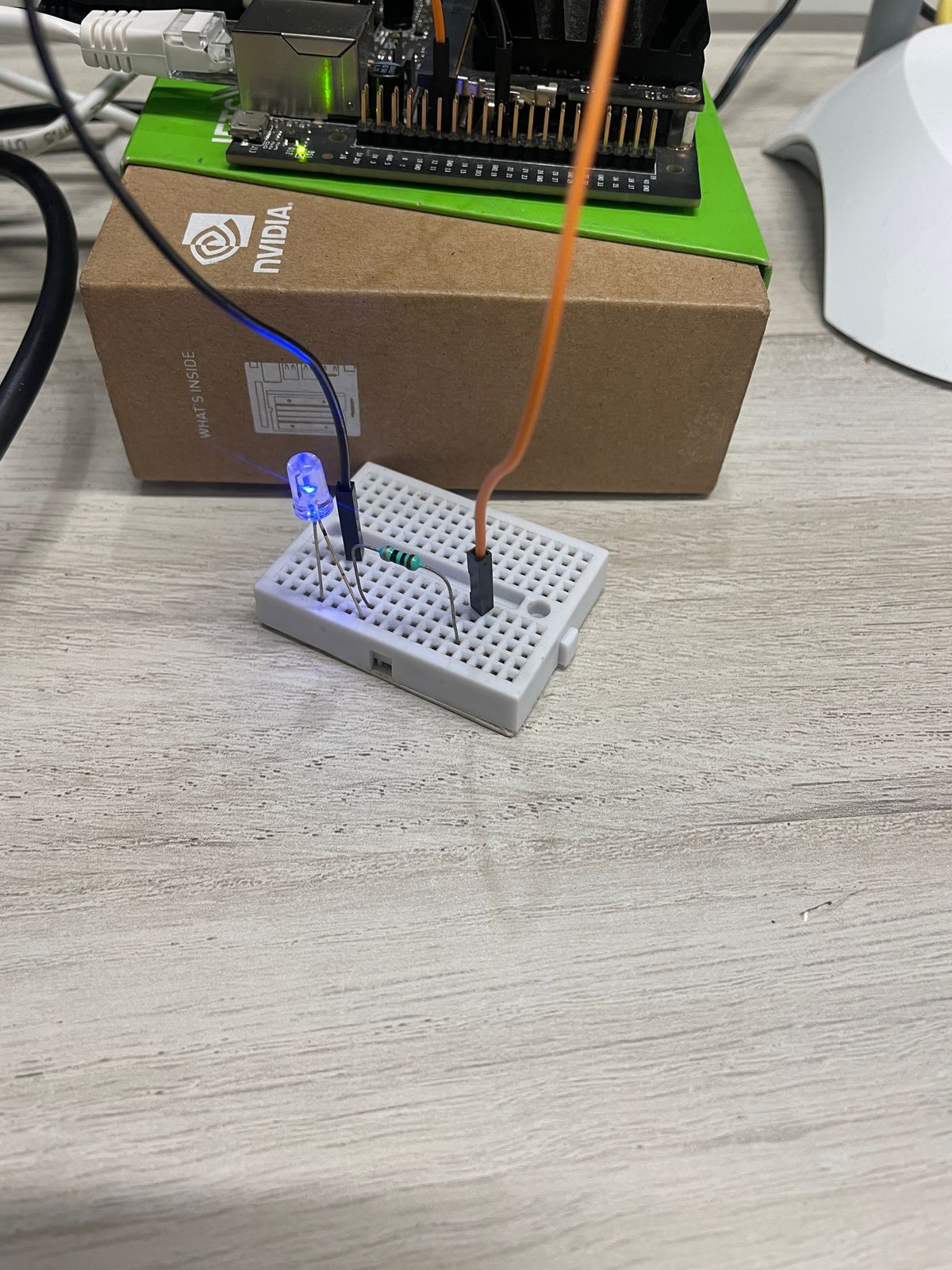
### How to Run the Project:

* 1. Run the Server on Jetson Nano
  2. Run the Client on Any Device
  3. Use the GUI



* + - Click the “LED ON” button to turn the LED on
    - Click the “LED OFF” button to turn the LED off





.