

ESP8266 Built-in LED Blink using ROS 2 Jazzy (Ubuntu 24.04 on WSL)

◆ PART 1: WINDOWS (WSL USB SETUP)

1 Check WSL

```
wsl --version
```

2 Install USB attach tool

```
winget install usbipd
```

→ Reboot Windows

3 Plug ESP8266 → list USB devices

```
usbipd list
```

Example output:

BUSID	VID:PID	DEVICE
1-9	1A86:7523	USB-SERIAL CH340

4 Bind device (ONE TIME ONLY)

```
usbipd bind --busid 1-9
```

5 Attach device to WSL (EVERY BOOT / REPLUG)

```
usbipd attach --busid 1-9 --wsl
```

◆ PART 2: UBUNTU (WSL SIDE)

6 Confirm device is visible

```
ls /dev/ttyUSB*
```

Expected:

```
/dev/ttyUSB0
```

7 Fix permission

```
sudo chmod 666 /dev/ttyUSB0
```

◆ PART 3: ARDUINO IDE + ESP8266

8 Install Arduino IDE

```
sudo apt update
```

```
sudo apt install arduino -y
```

```
arduino
```

9 Add ESP8266 board support

Arduino IDE →

File → Preferences → Additional Boards Manager URLs

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

10 Install ESP8266 boards

Arduino IDE →

Tools → Board → Boards Manager

Search esp8266 → Install

1 1 Select board & port

Tools → Board → NodeMCU 1.0 (ESP-12E)

Tools → Port → /dev/ttyUSB0

1 2 Upload ESP8266 code

```
#define LED_PIN LED_BUILTIN
```

```
void setup() {
```

```
pinMode(LED_PIN, OUTPUT);
digitalWrite(LED_PIN, HIGH);
Serial.begin(9600);

}

void loop() {
if (Serial.available()) {
    char c = Serial.read();
    if (c == '1') digitalWrite(LED_PIN, LOW);
    if (c == '0') digitalWrite(LED_PIN, HIGH);
}
}
```

✓ Test using Serial Monitor

1 → LED ON

0 → LED OFF

◆ PART 4: ROS 2 WORKSPACE SETUP

1 3 Create workspace

```
mkdir -p ~/ros2_ws/src
```

```
cd ~/ros2_ws/src
```

1 4 Create ROS package

```
ros2 pkg create led_control --build-type ament_python
```

◆ PART 5: PYTHON VIRTUAL ENV (Ubuntu 24.04 SAFE WAY)

1 5 Create venv

```
cd ~/ros2_ws/src/led_control
```

```
python3 -m venv venv
```

1 6 Activate venv

```
source venv/bin/activate
```

1 7 Install pyserial

```
pip install pyserial
```

Verify:

```
python -c "import serial; print(serial.__version__)"
```

◆ **PART 6: ROS NODE CODE**

1 8 Create ROS Python node

```
cd ~/ros2_ws/src/led_control/led_control
```

```
nano led_serial_node.py
```

```
import rclpy
```

```
from rclpy.node import Node
```

```
import serial
```

```
import time
```

```
class LedNode(Node):
```

```
    def __init__(self):
```

```
        super().__init__('led_node')
```

```
        self.ser = serial.Serial('/dev/ttyUSB0', 9600)
```

```
        time.sleep(2)
```

```
        self.state = False
```

```
        self.timer = self.create_timer(1.0, self.toggle)
```

```
        self.get_logger().info("LED control started")
```

```
    def toggle(self):
```

```
    self.ser.write(b'1' if not self.state else b'0')
    self.state = not self.state

def main():
    rclpy.init()

    node = LedNode()

    rclpy.spin(node)

    rclpy.shutdown()
```

1 9 Update setup.py

```
nano ~/ros2_ws/src/led_control/setup.py

entry_points={

    'console_scripts': [
        'led_serial = led_control.led_serial_node:main',
    ],
},
```

◆ PART 7: BUILD & RUN ROS NODE

2 0 Build workspace

```
cd ~/ros2_ws
colcon build --symlink-install
```

2 1 Source ROS

```
source install/setup.bash
```

2 2 Activate venv

```
source ~/ros2_ws/src/led_control/venv/bin/activate
```

2 3 Run ROS node 

```
ros2 run led_control led_serial
```

 **ESP8266 LED BLINKS**

 **ERRORS & SOLUTIONS**

 **/dev/ttyUSB0 not visible**

Reason: Using WSL

Fix:

```
usbipd bind --busid X-Y
```

```
usbipd attach --busid X-Y --wsl
```

 **usbipd: Device is not shared**

Fix:

```
usbipd bind --busid X-Y
```

 **externally-managed-environment**

Reason: Ubuntu 24.04 blocks pip globally

Fix:

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
pip install pyserial
```

 **No executable found**

Reason: Wrong executable name or not rebuilt

Fix:

```
colcon build
```

```
ros2 run led_control led_serial
```

(Case-sensitive!)