

# Instructor Guide – Module:

## Microcontrollers Explained: Build a Legal WiFi Detector with ESP32

### Before the Session

- Prepare ESP32, LEDs with current-limiting resistors, breadboards, USB cables, and Arduino IDE.

- Have a working demo ready to show.

#### **1. Hook: Connect to Everyday Life (5–10 mins)**

- Ask: “*What do a washing machine, a smart TV, and a car have in common?*”

- Show a quick demo of the WiFi detector LEDs responding to network density.

- State the goal: “*Today, we’ll build the ‘tiny brain’ behind this system.*”

#### **2. What is a Microcontroller? (10–15 mins)**

- Define: A small computer on a single IC designed for **specific tasks**, running one program repeatedly.

- Show internal components on a diagram or physical board:

- CPU (executes instructions)
- RAM (temporary data)
- Flash (permanent program storage)
- GPIO pins (connect to LEDs, sensors)
- WiFi/Bluetooth modules

#### **3. What Does a Microcontroller Do? (10 mins)**

- Introduce the **input-process-output model**:

1. **Read inputs** (WiFi signals)
2. **Process data** (count networks, compare thresholds)
3. **Control outputs** (light LEDs)

- Map it to the project:

- Input = WiFi scan
- Process = count + compare
- Output = LED color

#### **4. Microcontroller vs Microprocessor (5–10 mins)**

- Use the table to compare:

- Purpose (embedded control vs general computing)
- Memory (built-in vs external)
- Examples (ESP32 vs laptop CPU)

#### **5. Inside the ESP32 (5–10 mins)**

- Highlight key components that make it ideal for this project:

- Dual-core processor
- WiFi module (passive scanning)
- GPIO pins (LED control)
- ADC (for future sensor expansion)
- Flash memory (stores firmware)

## **6. Why ESP32 for This Project (5 mins)**

- Built-in WiFi
- GPIO for LEDs
- Affordable and programmable with Arduino IDE
- **Passive detection only** — legally compliant, no interference

## **7. The Control Loop: How It Works (15–20 mins)**

### **From Section 6:**

- Walk through the 7-step loop:
  1. Power via USB
  2. Setup initializes peripherals
  3. WiFi scan begins
  4. Count stored in RAM
  5. Conditional logic:
    - Green  $\leq 5$
    - Yellow 6–10
    - Red  $> 10$
  6. Activate LED
  7. Repeat every 5 seconds
- Show the code and trace through the loop.
- Emphasize **real-time embedded system** behavior.

## **8. Hands-On Activity (20–30 mins)**

- Students build and upload the project.
- Encourage experimentation:
  - Change thresholds
  - Add a buzzer
  - Test in different locations

### **Instructor Role:**

- Assist with wiring (remind about **current-limiting resistors** for LEDs).
- Reinforce the **control loop** as they test.

## **9. Wrap-Up and Discussion (10 mins)**

### **From Section 7 (Instructor Notes):**

- Revisit the “tiny brain” analogy.
- Emphasize:
  - Embedded systems concept
  - Legal/ethical responsibility (passive listening only)
  - Hardware limits (GPIO current, resistor use)