OVERVIEW - SCHEMATIC AND COMMUNICATION PROTOCOL

Wiring Schematic

Items	Model	Quantity	Power Supply
NodeMCU ESP8266	ESP8266-12E	3	4.5 to 9 VDC
Raspberry Pi (RPi)	RPi 5	1	5 to 15 VDC (27W)
Photoresistor		3	With resistors
Light Emitting Diode	Red/Yellow/Green/White LEDs	4	With resistors
Resistor	220 Ω	4	
Resistor	10k Ω	3	
Push Button	Bread-board Mount	1	

Table 1. Equipment list

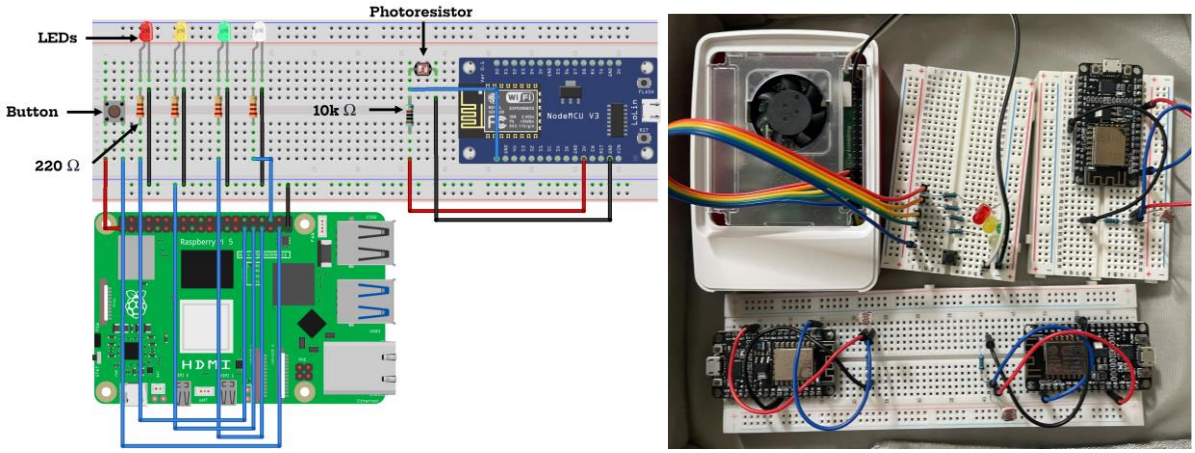


Figure 1. Wiring schematic for RPi5 and each ESP8266 module

- ESP8266:
    - 3.3V → Photoresistor (+)
    - A0 → Photoresistor (+)
- Raspberry Pi 5
    - 3.3V → Button (+)
    - GPIO 26 → Button (-)
    - GPIO 5 → Red LED (+)
    - GPIO 6 → Yellow LED (+)
    - GPIO 13 → Green LED (+)
    - GPIO 16 → White LED (+)

Communication Protocol/Message Description

Protocol Overview

- The system uses broadcast UDP messages over a mobile hotspot network to facilitate communication among multiple ESP8266 devices and a Raspberry Pi. Each message follows a specific format, allowing the devices to identify the type of message, sender, and content.

Message Format

- Each message contains:
    - Start Bit: Identifies the beginning of a message.
    - Payload: Contains specific information (e.g., sensor readings, commands).
    - End Bit: Marks the end of a message.
- General Message Structure:
    - Start Bit: "+++" | "~~~" → indicates the start of the message.
    - End Bit: "***" | "---" → indicates the end of the message.
    - Payload: Content varies depending on the message type (sensor data or control command).
- For example: +++"<PAYLOAD>"***

Communication Flow

- Communication Between ESP8266 Modules:
    - Each ESP8266 device broadcasts its Swarm ID and analog sensor reading periodically over UDP.
    - All ESP8266 devices listen to these broadcasts to compare their sensor readings and determine if they should be the Master based on the highest reading.
- Sensor Data Broadcast from ESP8266 to Raspberry Pi:
    - Each ESP8266 sends its Swarm ID (self-assigned from the last digit of its IP address) and analog sensor reading (from a light sensor) to the Raspberry Pi.
    - The RPi5 listens to incoming messages and processes each one by logging the data and triggering LED feedback based on the analog reading.
- System Reset Broadcast from RPi5 to ESP8266:
    - When the RPi5's button is pressed, it broadcasts a reset command "RESET_REQUESTED" to all ESP8266 devices.

- ESP8266 devices, upon receiving this command, reset their internal state as required (not explicitly coded here, but implied for the reset functionality).

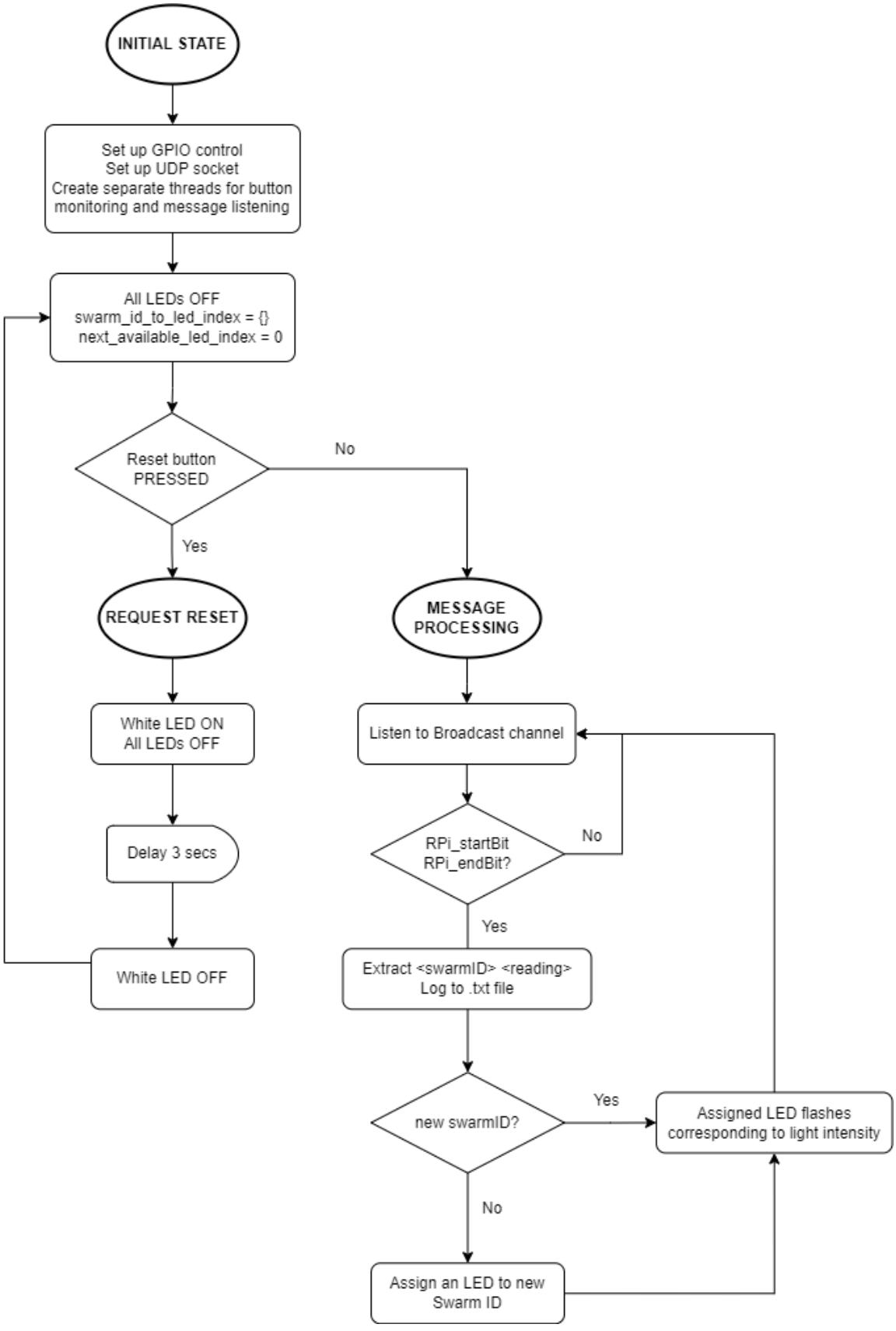
Detailed Message Descriptions

- Sensor Data Message (ESP8266 to ESP8266):
  - Start Bit: "~~~"
  - Payload: "<Swarm ID>,<Analog Reading>"
    - Swarm ID: Unique identifier for each ESP8266 (based on the last digit of its IP address).
    - Analog Reading: Light sensor reading as an integer, used to compare against other modules' readings.
  - End Bit: "---"
  - Example: If an ESP8266 with Swarm ID 4 has an analog reading of 750, the message would look like: "~~~4,750---"
- Sensor Data Message (from ESP8266 Master to RPi5):
  - Start Bit: "+++"
  - Payload: "<Swarm ID>,<Analog Reading>"
  - End Bit: "***"
- Reset Request Message (from RPi5 to all ESP8266): When the button connected to the RPi5 is pressed, it broadcasts a reset request message to all ESP8266 devices. This message has the following structure:
  - Start Bit: "+++"
  - Payload: "RESET_REQUESTED"
  - End Bit: "***"

Message Type	Direction	Start Bit	Payload	End Bit
Sensor Data	ESP <> ESP	"~~~"	"<SwarmID>,<AnalogReading>"	"---"
Master Data	ESP → RPi	"+++"	"<SwarmID>,<AnalogReading>"	"***"
Reset Request	RPi → ESP	"+++"	"RESET_REQUESTED"	"***"

PART 1 - RASPBERRY PI WIFI SETUP AND PACKET DELIVERY

Flowchart


INITIAL STATE
Set up GPIO control
Set up UDP socket
Create separate threads for button monitoring and message listening
All LEDs OFF
swarm_id_to_led_index = {}
next_available_led_index = 0
Reset button PRESSED
No
Yes
REQUEST RESET
MESSAGE PROCESSING
White LED ON
All LEDs OFF
Delay 3 secs
White LED OFF
Listen to Broadcast channel
RPi_startBit
RPi_endBit?
No
Yes
Extract <swarmID> <reading>
Log to .txt file
new swarmID?
Yes
Assigned LED flashes corresponding to light intensity
No
Assign an LED to new Swarm ID

State Descriptions

1. **Idle State:**

   The system remains idle, waiting for UDP messages from the ESP8266 swarm devices and monitoring the button. In this state, LEDs remain off unless a message is received, and the button remains unpressed.

2. **Message Processing State:**

   When a UDP message arrives from an ESP8266 device, the system parses the message, extracts the Swarm ID and sensor reading, logs the data, and maps the Swarm ID to an LED for flashing. The LED flashes at an interval calculated from the sensor reading.

3. **Reset State:**

   When the button on the Raspberry Pi is pressed, the system enters the reset state. It sends a broadcast reset request message to the ESP8266 devices, clears the sensor data log, resets LED assignments, and lights up the white LED for 3 seconds to signal reset completion. The system then returns to the Idle State.

Main Functionality

The Raspberry Pi code manages inputs from a button, processes UDP messages from ESP8266 devices to control LEDs, and outputs visual indicators via LEDs. The main functionalities are structured as follows:

1. **Input**

   - Button Monitoring:
     - The monitor_button function continuously checks the state of the button connected to GPIO pin 26.
     - When the button is pressed, it triggers the reset_system function, initiating a reset of the system.
   - UDP Message Reception:
     - The listen_for_messages function listens on port 4210 for incoming UDP messages.
     - Messages contain sensor data from ESP8266 devices, structured with delimiters (+++ and ***) and a payload of <Swarm ID>,<Analog Reading>.

2. **Process**

   - Message Parsing:
     - Upon receiving a message, the system verifies its format using start (+++) and end (***) delimiters.
     - It then extracts the Swarm ID and analog sensor reading from the payload.
   - Logging and Mapping:

- The sensor data is logged to sensor_readings.txt, with entries recorded as Swarm ID: <Analog Reading>.
- New Swarm IDs are assigned an RGB LED based on the swarm_id_to_led_index dictionary, which maps each unique Swarm ID to a specific LED.

- LED Control:
  - For each Swarm ID, the flashing interval of the assigned LED is calculated based on the analog sensor reading. Using a linear interpolation formula, the interval between LED flashes is set based on the sensor value.
  - LEDs toggle state at each calculated interval, creating a flashing effect that visually represents the sensor reading rate for each Swarm ID.
- System Reset:
  - When reset_system is triggered by a button press, the Raspberry Pi broadcasts a RESET_REQUESTED message to the ESP8266 devices.
  - It clears the sensor data log, resets LED assignments, turns off all RGB LEDs, and flashes the white LED for 3 seconds.

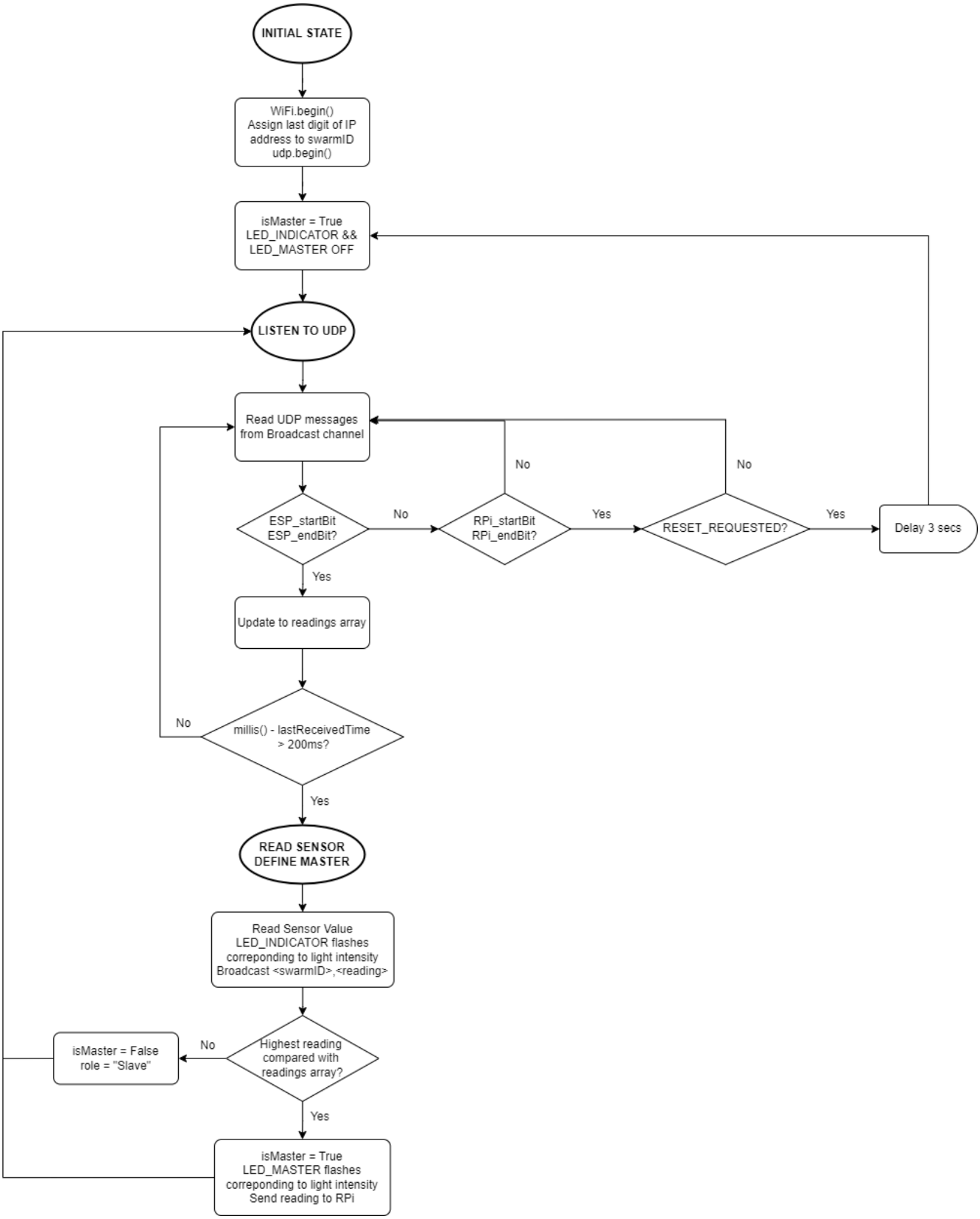
3. Output

- Log Output:
  - Sensor readings from the ESP8266 devices are appended to sensor_readings.txt, providing a continuous log of received data.
- Visual Output (LED Control):
  - The assigned RGB LED for each Swarm ID flashes at intervals corresponding to the analog reading, indicating activity and providing a visual cue of the sensor's intensity.
  - During a reset, the white LED is activated for 3 seconds to signal the reset's progress and completion.

This setup ensures that the RPi5 manages communication with ESP8266 devices, controls LED indicators based on sensor data, and handles system resets effectively in response to the button input.

PART 2 - ESP WIFI SETUP AND PACKET DELIVERY

Flowchart



State Descriptions

1. **State 1: Initial Setup**

   The ESP8266 connects to the WiFi network, assigns a unique swarmID based on the last digit of the IP address, and initializes UDP communication. The LED pins are set up, and the slope and intercept for flashing the LEDs based on light sensor readings are calculated.

2. **State 2: Sensor Reading and Master Determination**

   The ESP8266 continuously reads the light sensor value and flashes an indicator LED according to the sensor reading. If the device is the Master, the Master status LED flashes as well. It then checks for incoming UDP messages from other devices or the RPi5.

3. **State 3: Broadcasting and Receiving Data**

   The device broadcasts its sensor reading over UDP if no message has been received for a set time (silentTime). It also processes incoming messages from other ESP8266 devices (to update readings) and from the RPi5 (to handle reset requests). The Master device sends its reading and ID to the RPi5 and adjusts its status based on the highest sensor reading in the swarm.

Main Functionality

- Input:
  - WiFi credentials and UDP communication setup for network interaction.
  - Analog light sensor readings (analogValue) from the photoresistor connected to pin A0.
  - UDP packets from other ESP devices (light sensor readings) and from the Raspberry Pi (reset requests).
- Process:
  - WiFi Connection: The ESP8266 connects to the specified WiFi network and initializes UDP communication on a given port.
  - LED Flashing: Based on the light sensor readings (then deduce to be a linear equation), the code calculates an interval for flashing an indicator LED and a Master status LED. These LEDs toggle at a rate corresponding to the sensor input.
  - Swarm ID Assignment: Each ESP assigns itself a unique swarmID derived from its IP address, ensuring no conflicts in the swarm of devices.
  - Data Broadcasting: If no UDP messages are received for a specified silent period, the device broadcasts its light sensor reading along with its swarmID to other devices.
  - Role Determination (Master/Slave): Each ESP device compares its reading to others in the swarm. The device with the highest sensor reading becomes the

Master, and it broadcasts this information. If it's not the Master, it assumes the Slave role.

- Message Handling: The device listens to incoming UDP messages, processes ESP-to-ESP data (light sensor readings), and handles reset requests from the Raspberry Pi (resetting the Master role and LED states).

- Output:
  - LED Indicators: The LED_INDICATOR and LED_MASTER LEDs flash at intervals based on the light sensor readings, providing visual feedback on the sensor's intensity and Master role.
  - UDP Messages:
    - ESP-to-ESP: The device broadcasts its sensor reading, swarmID, and status.
    - RPi Communication: The Master sends its reading and ID to the RPi5 to log and display the data. If a reset is requested, the Master role is reset, and LEDs are toggled.