Laboratory 4: Integration of Sensors with ESP32	**School of Applied Digital Technology**

Name: **Name:**	**ID:** **ID:**	**Section:** **Section:**
Date:	**Due date:**	

Objectives

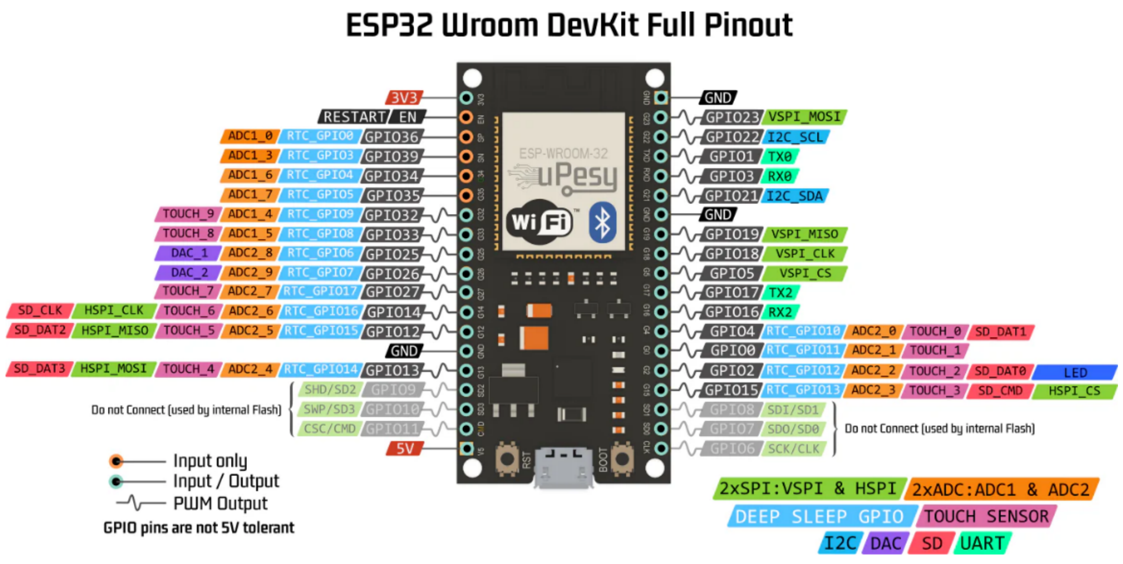
- Connect a Servo Motor and Buzzer to the ESP32 board.
- Configure ESPHome to control these components.
- Monitor and control the devices through Home Assistant.

Experiment 1: Integration of Sensors with ESP32

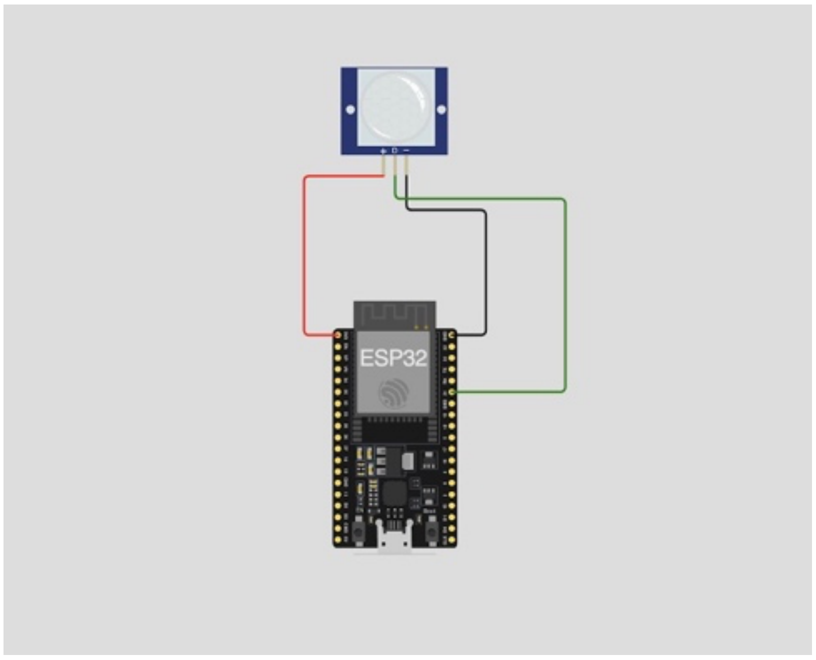
Equipment:

	Quantity
ESP32	1
Motion Sensor Detector Module HC-SR501	1
LDR Photosensitive Sensor Module Light Sensor	1

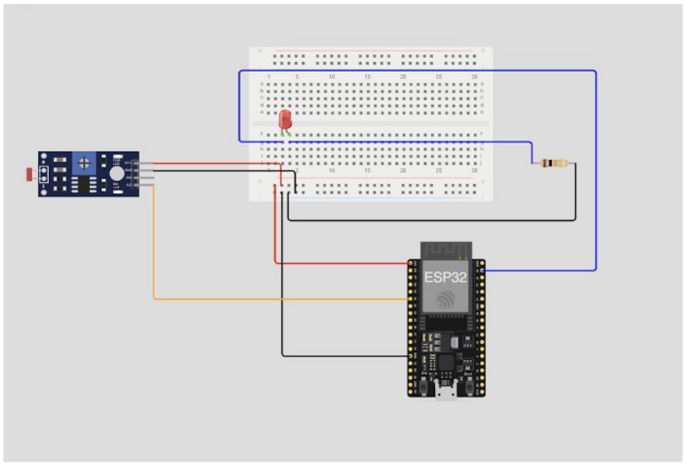
1. Wiring the Sensors to the ESP Board



- Motion Sensor Detector Module HC-SR501
  - VCC ⟶ 3V
  - GND ⟶ GND
  - OUT ⟶ GPIO14



- LDR Photosensitive Sensor Module Light Sensor
  - VCC ⟶ 3V
  - GND ⟶ GND
  - OUT ⟶ GPIO14



2. Open ESPHome and Select the Device

- Go to the ESPHome Dashboard
- Select your device (e.g., sensor)
- Click EDIT to modify the YAML configuration

3. Add Code for the Sensors in the YAML File

Add the following to your configuration file:

- Motion Sensor Detector Module HC-SR501

```yaml
# PIR Motion Sensor
binary_sensor:
  - platform: gpio
    pin:
      number: GPIO14
      mode: INPUT
    name: "Living Room Motion"
    device_class: motion
```

- LDR Photosensitive Sensor Module Light Sensor

```
# ----------------------------
# Analog LDR (AO)
# ----------------------------
sensor:
  - platform: adc
    pin: GPIO35
    name: "Living Room Light (Analog)"
    update_interval: 10s
    attenuation: 11db
    filters:
      - multiply: 3.3

# ----------------------------
# Digital LDR (DO)
# ----------------------------
binary_sensor:
  - platform: gpio
    pin: GPIO26
    name: "Bright Light Detected (Digital)"
    device_class: light
    filters:
      - delayed_on: 100ms
      - delayed_off: 100ms
```

4. Save and Install the Firmware

- Click SAVE
- Click INSTALL
- Choose Wirelessly or USB depending on your setup
- Wait for the upload to complete and let the device reboot

5. View Sensor Data in Home Assistant

- Go to your Home Assistant Overview Dashboard
- You should now see sensor readings such as:

  - Servo Motor (PWM)
  - Buzzer

Take 1: Design an LED control system using a light sensor (LDR) to automatically turn ON the LED when it is dark and OFF when it is bright.

Requirements:

1. Use an ESP32 board and an LDR sensor (Analog + Digital).
2. Connect an LED to indicate light/dark condition.
3. Read LDR Analog values every 1 second.
4. Combine the Analog/Digital sensor reading and LED control code into a single ESPHome YAML file.
5. Implement automation so the LED operates according to the following conditions:
   - LDR Analog < 2.0V ⟶ LED turns ON (dark)
   - LDR Analog ≥ 2.0V ⟶ LED turns OFF (bright)

Continued on the next page.

CODE:

----- Have a good day -----

ANSWER:

```
# LED Output
output:
  - platform: gpio
    pin: GPIO13
    id: led_output

# Analog LDR
sensor:
  - platform: adc
    pin: GPIO34
    name: "LDR Analog"
    id: ldr_analog
    update_interval: 1s
    attenuation: 11db
    filters:
      - multiply: 3.3  # ปรับค่าเป็นโวลต์ถ้าต้องการ

# Digital LDR
binary_sensor:
  - platform: gpio
    pin:
      number: GPIO26
      mode: INPUT
      inverted: false
    name: "LDR Digital"
    id: ldr_digital
    device_class: light

# Automation: LED control based on LDR analog
interval:
  - interval: 1s
    then:
      - if:
          condition:
            sensor.in_range:
              id: ldr_analog
              above: 2.0    # ปรับตามสภาพแสง
          then:
            - output.turn_off: led_output
          else:
            - output.turn_on: led_output
```