		
Laboratory 3: Integration of Sensors with ESP32	School of Applied Digital Technology	
Name: Name:	ID: ID:	Section: Section:
Date:	Due date:	

Objectives

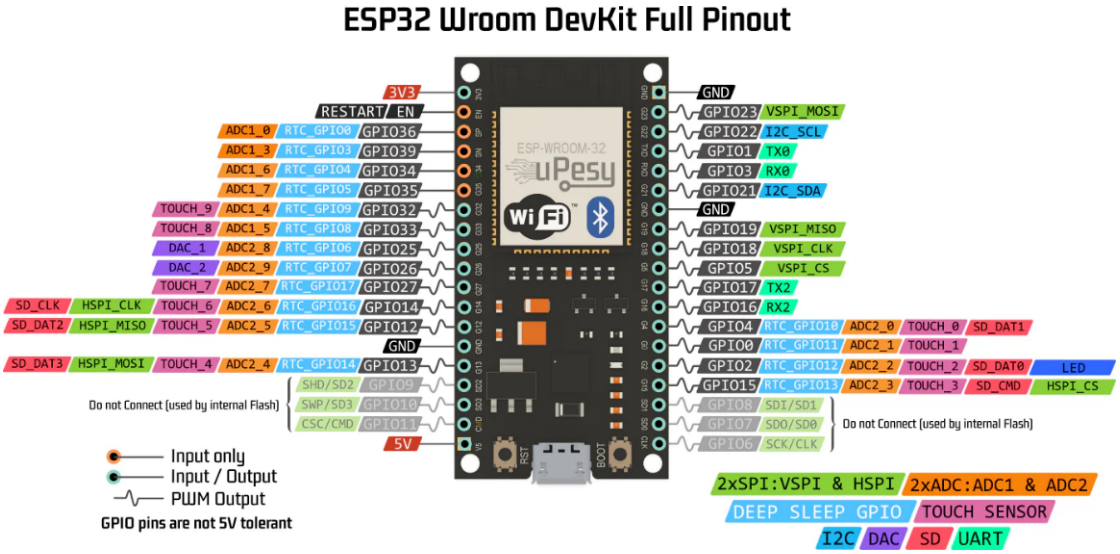
- Connect multiple sensors (LDR, LED, PIR) to the ESP32 board.
- Write configuration in ESPHome to read sensor data and control the LED.
- Display sensor data on Home Assistant.

Experiment 1: Integration of Sensors with ESP32

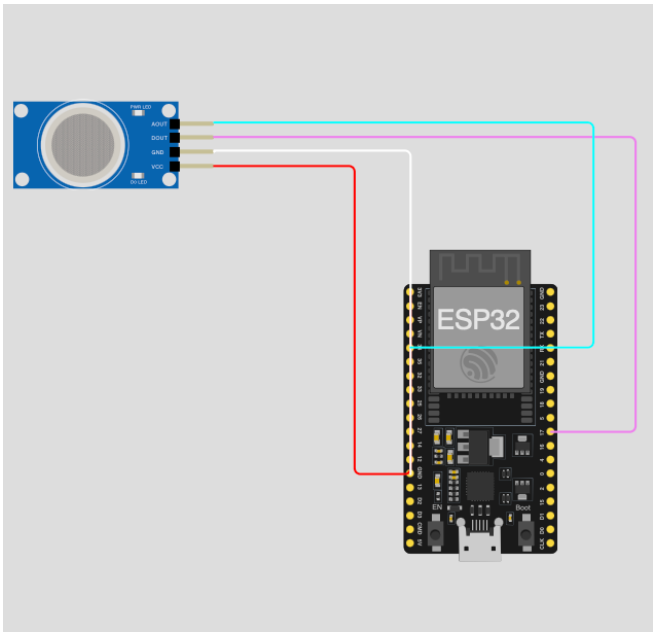
Equipment:

	Quantity
ESP32	1
MQ-2	1
Active Buzzer	1

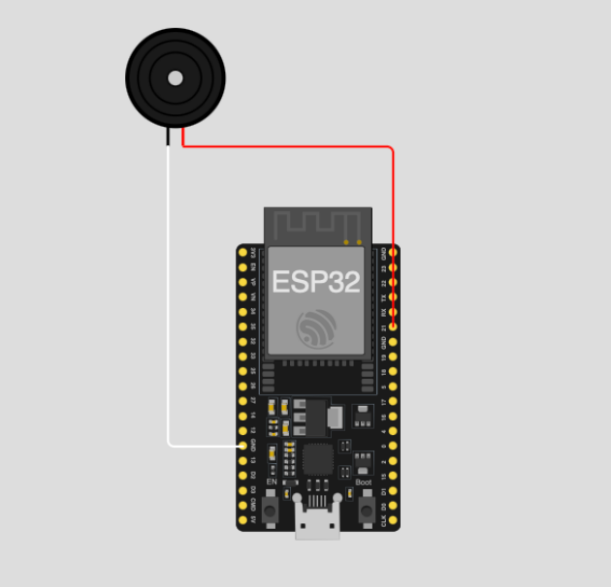
1. Wiring the Sensors to the ESP Board



- MQ-2
  - VCC ⟶ 5V
  - GND ⟶ GND
  - A0 ⟶ GPIO34 (ADC input)
  - D0 ⟶ GPIO17 (digital input)



- **Active Buzzer**
    - VCC ⟶ VCC
    - AOUT ⟶ GPIO21
    - GND ⟶ GND



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:

- **MQ-2**

```
sensor:
  - platform: adc
    pin: GPIO34
    name: "MQ-2 Gas Sensor Analog"
```

```
    id: mq2_sensor
    update_interval: 2s
    attenuation: 12db
    filters:
      - multiply: 3.3

binary_sensor:
  - platform: gpio
    pin: GPIO17
    name: "MQ-2 Gas Sensor Digital"
    id: mq2_digital
```

- Buzzer Module Active

```
output:
  - platform: gpio
    pin: GPIO21
    id: buzzer_output
    inverted: true

switch:
  - platform: output
    name: "Buzzer"
    output: buzzer_output
```

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:
  - Gas Detection Level from MQ-2
  - Buzzer Module Active

Take 1: Design and build a gas alarm system using MQ-2 sensors and buzzers.

Requirements:

1. Use an ESP32/ESP8266 board and an MQ-2 Gas Sensor.
2. Connect an Active Buzzer to indicate gas levels.
3. Read MQ-2 Analog values every 2 seconds.
4. Combine the sensor reading and buzzer control code into a single ESPHome YAML file.
5. Implement automation so the buzzer operates according to the following conditions:
   - MQ-2 Analog ≥ 1.5V ⟶ Buzzer turns ON (alert)
   - MQ-2 Analog < 1.5V ⟶ Buzzer turns OFF
6. Display MQ-2 Analog values (and optionally Digital output) in Home Assistant.

Continued on the next page.

CODE:

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

ANSWER:

```
# ---------- MQ-2 Analog ----------
sensor:
  - platform: adc
    pin: GPIO34
    name: "MQ-2 Gas Sensor Analog"   # ชื่อเต็ม
    id: mq2_sensor
    update_interval: 2s
    attenuation: 12db
    filters:
      - multiply: 3.3   # แปลงค่าเป็นโวลต์

# ---------- Active Buzzer ----------
output:
  - platform: gpio
    pin: GPIO21
    id: buzzer_output
    inverted: true

switch:
  - platform: output
    name: "Active Buzzer"
    output: buzzer_output

# ---------- Automation ----------
interval:
  - interval: 1s
    then:
      - lambda: |-
          const float threshold = 2;  // กำหนดค่ามาตรฐาน
          if (id(mq2_sensor).state > threshold) {
            id(buzzer_output).turn_on();   // เกินค่า ⟶ ดัง
          } else {
            id(buzzer_output).turn_off();  // ไม่เกิน ⟶ เงียบ
          }
```