7TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Boulevard, Ermita, Manila

CIT-ELECTRONICS DEPARTMENT

CPET11L-M – Microprocessor and Microcontroller Systems, Lab

1st Semester, SY 2-24-2025

Name: Joseph C. Arenas	**Instructor:** Prof. Tristan Mopas
Course & Section: BET-CPET- 3A	**Date Submitted:** October 14, 2025

Activity 6

I. OBJECTIVES

- To apply practical knowledge in using the ESP32
- To explain the functions and components of the related topics
- To implement the ESP32, Relay Module, Push button, resistors, 20x4 LED Screen, Soil Moisture Sensor, MQ-2 Gas Sensor, HC-SR04 Ultrasonic Sensor and LED components in a working circuit.
- To develop and enhance problem-solving skills related to the topics

II. EQUIPMENT AND MATERIALS

HARDWARE

- ESP32
- Breadboard
- Jumper Wires (Male-to-Male, Female-to-Male & Female-to-Female)
- Laptop
- Push button
- 220Ω Resistor x2
- 2-Channel Relay Module
- Red and Green LEDs

- 20x4 LED Screen
- Soil Moisture Sensor
- MQ-2 Gas Sensor
- HC-SR04 Ultrasonic Sensor

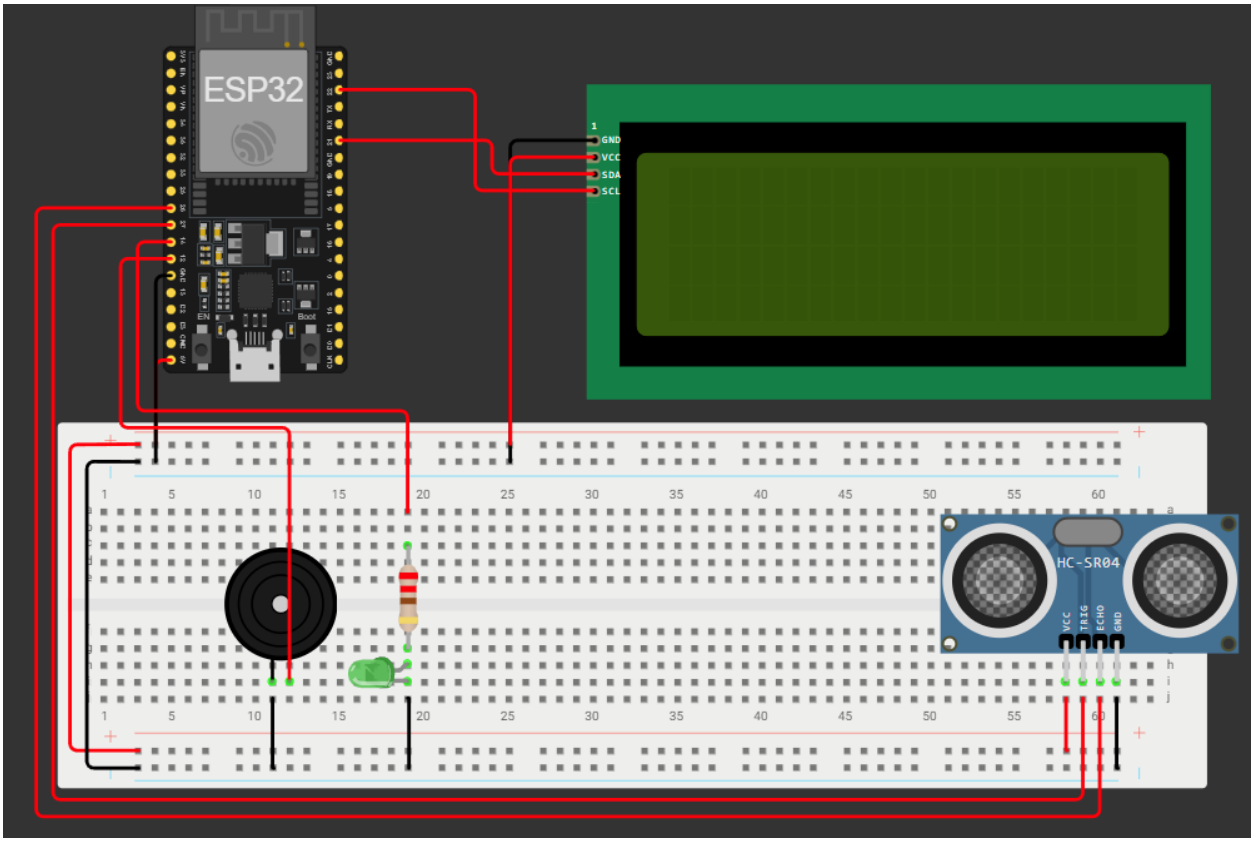
SOFTWARE

- Arduino IDE with libraries for specified components.
- MS Word
- Wokwi
- Blynk IOT Application

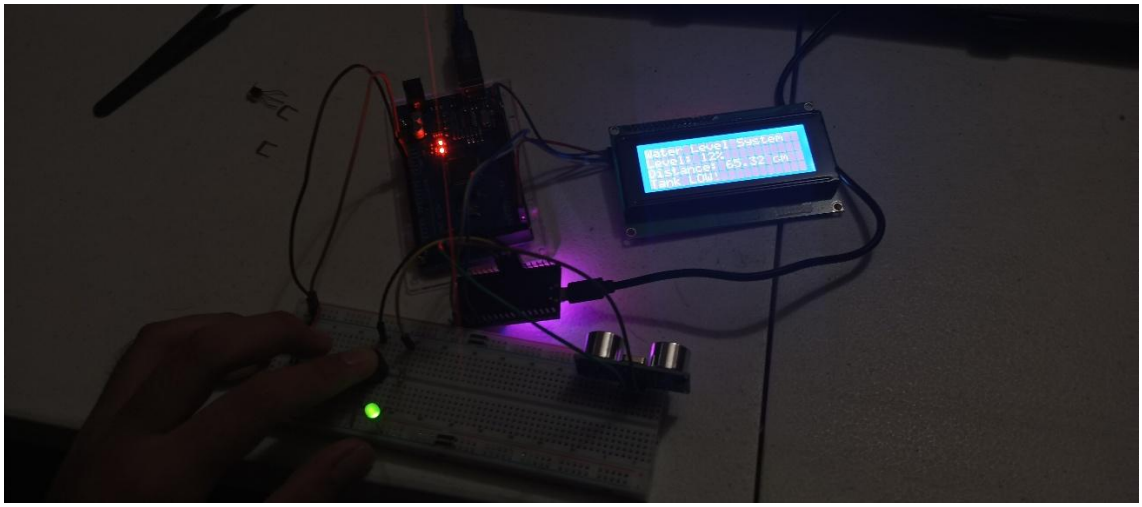
III. DIAGRAM

====== TOPIC 1: IoT Based Water Level Indicator using Ultrasonic Sensor ======

A. Wokwi Simulation



B. Breadboard



C. Source Code

```
#define BLYNK_TEMPLATE_ID "TMPL6C59Tss2O"

#define BLYNK_TEMPLATE_NAME "ESP32 WaterLevel"

#define BLYNK_AUTH_TOKEN "stEj6vFxp0WwxMm6vQaXXpVr_BFRy1J8"


char ssid[] = "1234";

char pass[] = "hello123";


// Water level limits (in cm)

int emptyTankDistance = 70;  // distance when tank is empty

int fullTankDistance  = 30;  // distance when tank is full


// Pins

#define TRIGPIN   27  // D27

#define ECHOPIN   26  // D26

#define wifiLed   2  // D2

#define BuzzerPin  12  // D12

#define GreenLed  14  // D14


#define VPIN_BUTTON_1   V1

#define VPIN_BUTTON_2   V2


#include <LiquidCrystal_I2C.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>


LiquidCrystal_I2C lcd(0x27, 20, 4); // LCD address and size
```

```
float duration;

float distance;

int waterLevelPer;


char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer;


// Blynk connection check

void checkBlynkStatus() {

 bool isconnected = Blynk.connected();

 digitalWrite(wifiLed, isconnected ? HIGH : LOW);

}


BLYNK_CONNECTED() {

 Blynk.syncVirtual(VPIN_BUTTON_1);

 Blynk.syncVirtual(VPIN_BUTTON_2);

}


// Display function for LCD

void displayData(int value) {

 lcd.clear();

 lcd.setCursor(0, 0);

 lcd.print("Water Level System");

 lcd.setCursor(0, 1);

 lcd.print("Level: ");

 lcd.print(value);

 lcd.print("%");

 lcd.setCursor(0, 2);

 lcd.print("Distance: ");
```

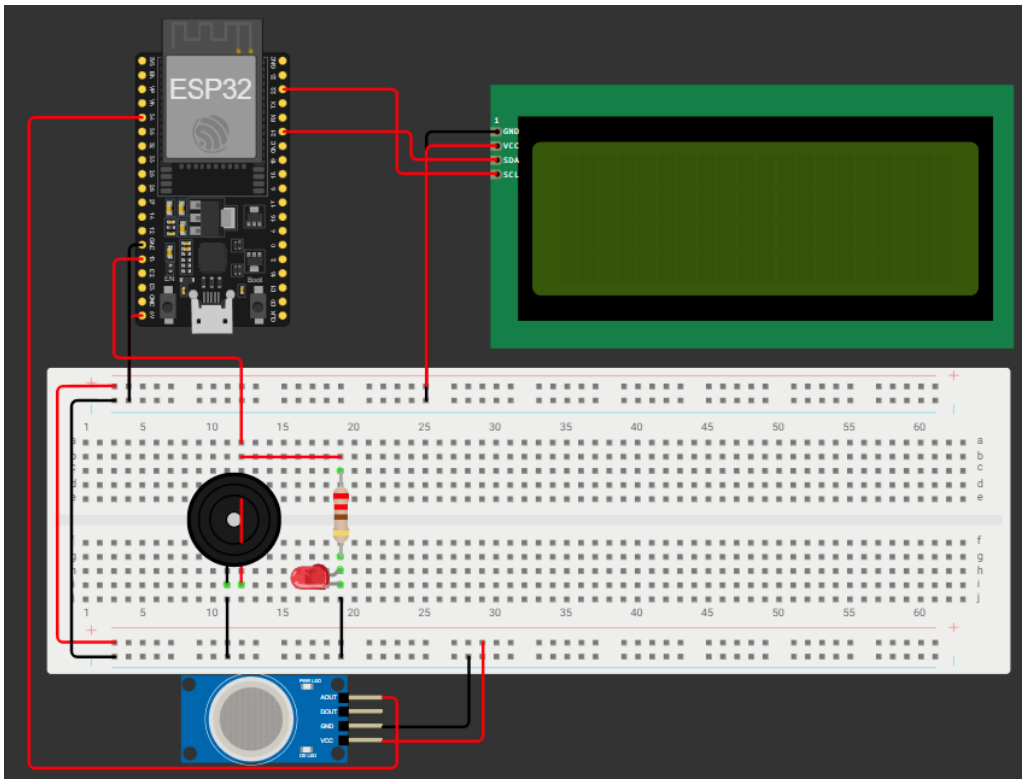
```
lcd.print(distance);
 lcd.print(" cm");
 lcd.setCursor(0, 3);
 if (value >= 100) {
  lcd.print("Tank FULL!");
 } else if (value <= 25) {   //  🔹 changed from 10% to 25%
  lcd.print("Tank LOW!");
 } else {
  lcd.print("Normal Level");
 }
}

// Measure distance and control outputs
void measureDistance() {
 digitalWrite(TRIGPIN, LOW);
 delayMicroseconds(2);
 digitalWrite(TRIGPIN, HIGH);
 delayMicroseconds(20);
 digitalWrite(TRIGPIN, LOW);
duration = pulseIn(ECHOPIN, HIGH);
 distance = ((duration / 2) * 0.343) / 10;

 if (distance < emptyTankDistance && distance > (fullTankDistance - 10)) {
  waterLevelPer = map((int)distance, emptyTankDistance, fullTankDistance, 0, 100);
  displayData(waterLevelPer);
Blynk.virtualWrite(VPIN_BUTTON_1, waterLevelPer);
  Blynk.virtualWrite(VPIN_BUTTON_2, String(distance) + " cm");
```

```
Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print(" cm | Water Level: ");
  Serial.print(waterLevelPer);
  Serial.println("%");

  // Control logic (UPDATED)
  if (waterLevelPer >= 100 || waterLevelPer <= 25) {
//  🔹 changed from 10% to 25%
   digitalWrite(BuzzerPin, HIGH);
   digitalWrite(GreenLed, HIGH);
  } else {
   digitalWrite(BuzzerPin, LOW);
   digitalWrite(GreenLed, LOW);
  }
 }

 delay(300); // faster refresh rate
}
void setup() {
 Serial.begin(115200);

 pinMode(ECHOPIN, INPUT);
 pinMode(TRIGPIN, OUTPUT);
 pinMode(wifiLed, OUTPUT);
 pinMode(BuzzerPin, OUTPUT);
 pinMode(GreenLed, OUTPUT);

 digitalWrite(wifiLed, LOW);
 digitalWrite(BuzzerPin, LOW);
 digitalWrite(GreenLed, LOW);
```

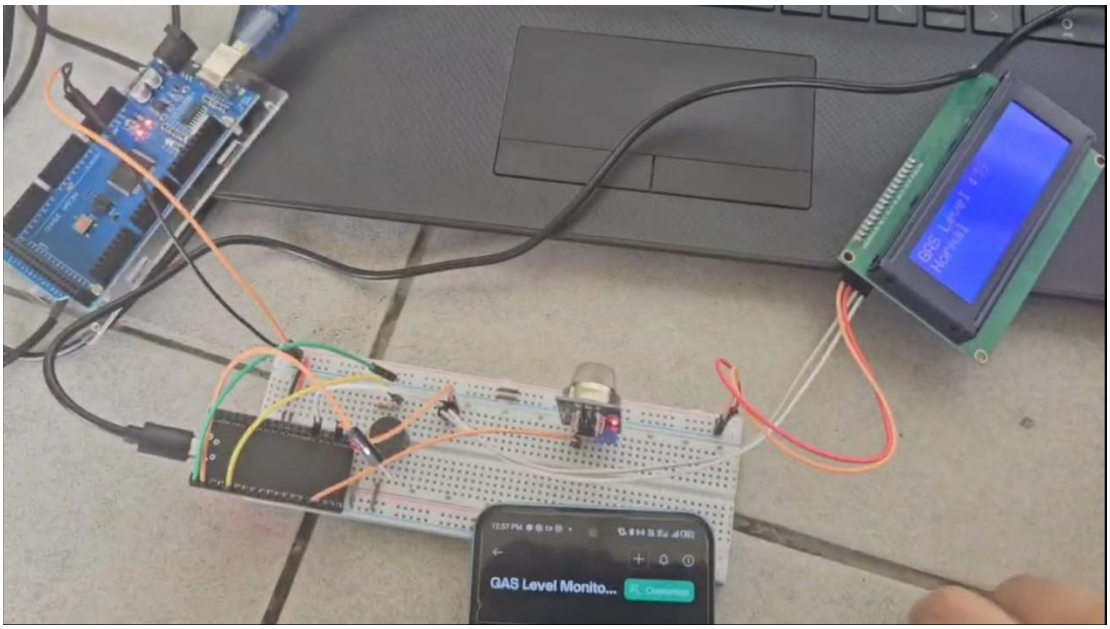
```
 lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Water Level System");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
  delay(1500);

  WiFi.begin(ssid, pass);
  timer.setInterval(2000L, checkBlynkStatus);
  Blynk.config(auth);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Connecting WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    lcd.print(".");
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi Connected!");
  delay(800);
  lcd.clear();
 }

 void loop() {
  measureDistance();
  Blynk.run();
  timer.run();
 }
```

====== TOPIC 2: GAS-Level Monitoring System with ESP32 Board ======

A. Wokwi Simulation



B. Breadboard



C. Source Code

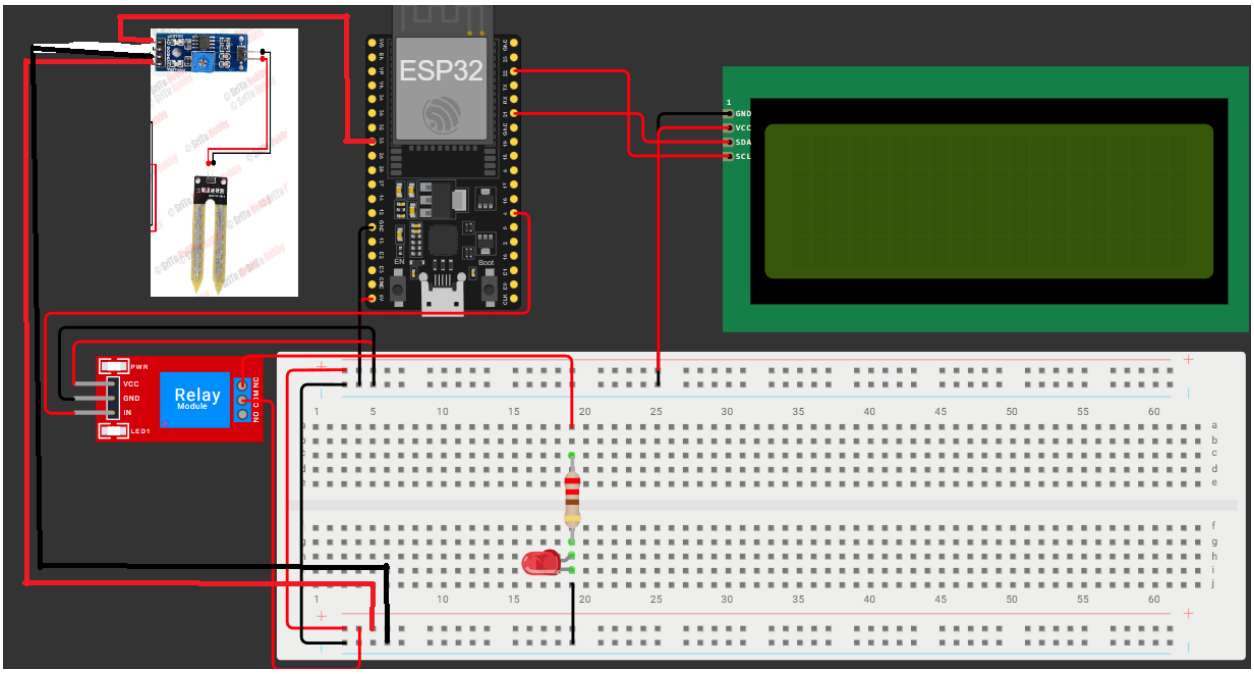
```
#define BLYNK_TEMPLATE_ID "TMPL6fkO4INWC"

#define BLYNK_TEMPLATE_NAME "GAS level monitoring system"

#define BLYNK_AUTH_TOKEN "GO27ivSmIl6l_KydsKM46MrCIgforkX3"


//Include the library files

#include <LiquidCrystal_I2C.h>

#include <Wire.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>


#define sensor 34

#define buzzer 13

//Initialize the LCD display

LiquidCrystal_I2C lcd(0x27, 16, 2);

BlynkTimer timer;

//Enter your WIFI SSID and password

char ssid[] = "1234";

char pass[] = "hello123";

WidgetLED LED(V1);

void setup() {

 // Debug console

 Serial.begin(115200);

 Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud", 80);

 lcd.init();

 lcd.backlight();

 pinMode(buzzer, OUTPUT);
```

```
lcd.setCursor(1, 0);

 lcd.print("System Loading");

 for (int a = 0; a <= 15; a++) {

  lcd.setCursor(a, 1);

  lcd.print(".");

  delay(200);

 }

 lcd.clear();


 timer.setInterval(1000L, GASLevel); // Run GASLevel every 1 sec

}


//Get the gas sensor values

void GASLevel() {

 int value = analogRead(sensor);

 value = map(value, 0, 4095, 0, 100);


 if (value >= 50) {

  digitalWrite(buzzer, HIGH);

  lcd.setCursor(0, 1);

  lcd.print("Warning!  ");

  LED.on();

 } else {

  digitalWrite(buzzer, LOW);

  lcd.setCursor(0, 1);

  lcd.print("Normal   ");

  LED.off();

 }
```

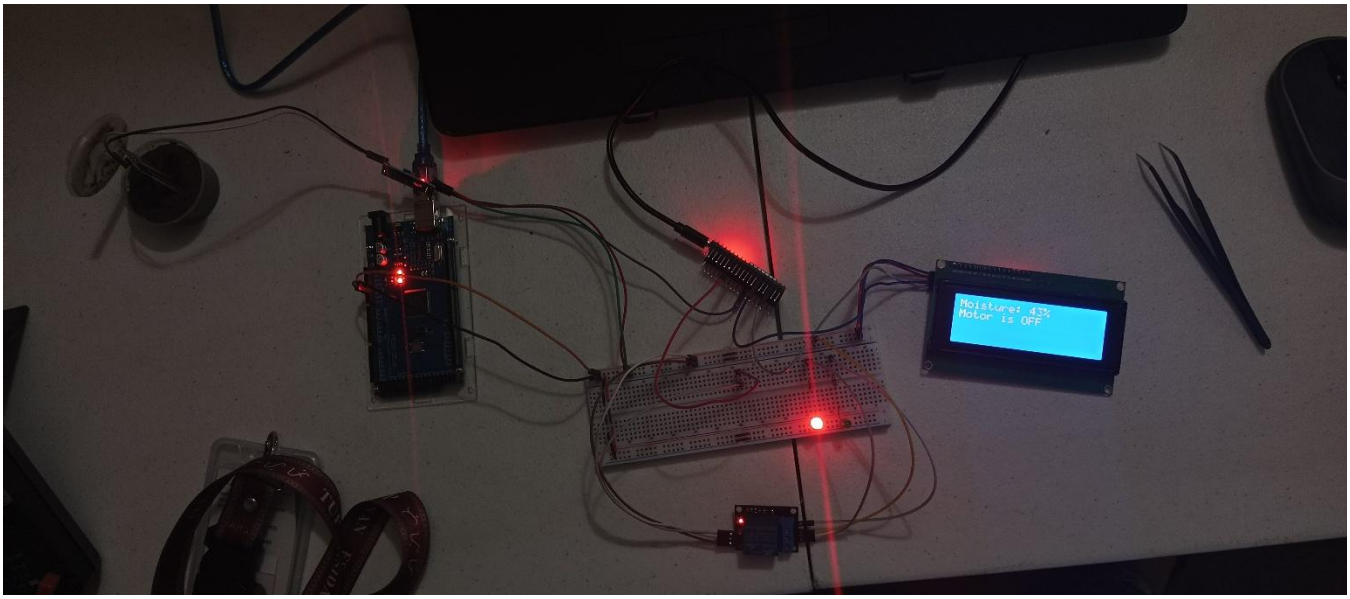
```
Blynk.virtualWrite(V0, value);
  Serial.println(value);
  lcd.setCursor(0, 0);
  lcd.print("GAS Level :");
  lcd.print(value);
  lcd.print("   "); // Spaces to overwrite old digits
}

void loop() {
  Blynk.run();   // Run the Blynk library
  timer.run();   // Run the timer
}
```

====== TOPIC 3: Plant Watering System with ESP32 Board and Blynk App======

A. Wokwi Simulation



B. Breadboard



C. Source Code

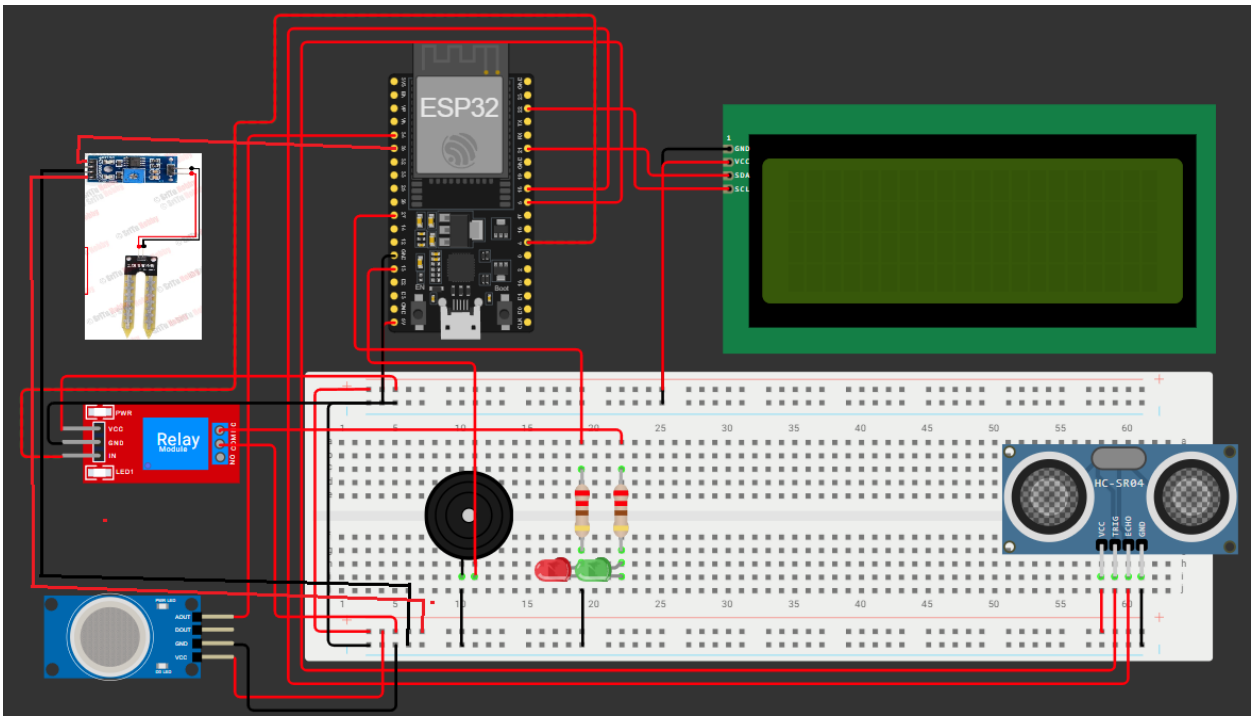
```
#define BLYNK_TEMPLATE_ID "TMPL618GSH8_F"

#define BLYNK_TEMPLATE_NAME "Plant watering system"

#define BLYNK_AUTH_TOKEN "EmG3-yCYv8iLg-TqsDeS755HlW6ZJuz5"

#include <LiquidCrystal_I2C.h>

#include <Wire.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#define sensor 33   // Soil moisture sensor pin

#define relay 4    // Relay pin for water pump

LiquidCrystal_I2C lcd(0x27, 16, 2);

BlynkTimer timer;

Char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "1234";

char pass[] = "hello123";

void setup() {

// Debug console

Serial.begin(115200);


// Connect to Blynk cloud

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

// Setup LCD

lcd.init();

lcd.backlight();
```

```
// Relay setup

pinMode(relay, OUTPUT);

digitalWrite(relay, HIGH); // OFF by default

lcd.setCursor(1, 0);

lcd.print("System Loading");

for (int a = 0; a <= 15; a++) {

lcd.setCursor(a, 1);

lcd.print(".");

delay(150);

}

lcd.clear();

timer.setInterval(1000L, soilMoisture);

}

void soilMoisture() {

int value = analogRead(sensor);

value = map(value, 0, 4095, 0, 100);   // Convert raw to percentage

value = (value - 100) * -1;          // Invert so dry = low %

Blynk.virtualWrite(V0, value);

Serial.print("Soil Moisture: ");

Serial.print(value);

Serial.println("%");


// Show on LCD

lcd.setCursor(0, 0);

lcd.print("Moisture: ");

lcd.print(value);

lcd.print("%   ");

}
```

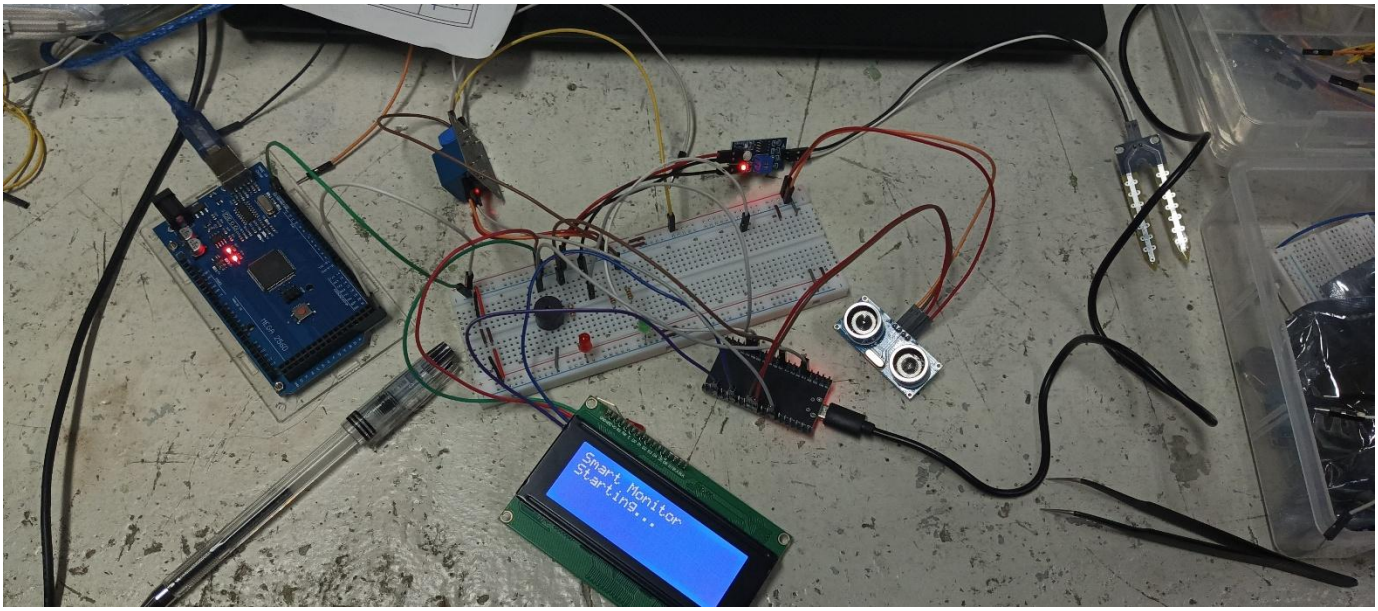
```
// Function to control motor from app button (V1)
BLYNK_WRITE(V1) {
bool Relay = param.asInt();
if (Relay == 1) {
digitalWrite(relay, LOW);   // Turn ON motor
lcd.setCursor(0, 1);
lcd.print("Motor is ON  ");
} else {
digitalWrite(relay, HIGH);  // Turn OFF motor
lcd.setCursor(0, 1);
lcd.print("Motor is OFF ");
}
}

void loop() {
Blynk.run();   // Run Blynk
timer.run();   // Run timer tasks
}
```

====== TOPIC 4: Combined Smart Monitoring System======

A. Wokwi Simulation



B. Breadboard



C. Source Code

```
#define BLYNK_TEMPLATE_ID
"TMPL6zbi2xukB"

#define BLYNK_TEMPLATE_NAME
"Activity6Topic4"

#define BLYNK_AUTH_TOKEN
"3ySQjbx_EHKsG5DgEKsaJyHSs1hZmnkd"


char ssid[] = "1234";

char pass[] = "hello1234";


// ==== LIBRARIES ====

#include <Wire.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <LiquidCrystal_I2C.h>

BlynkTimer timer;

LiquidCrystal_I2C lcd(0x27, 20, 4);

#define TRIGPIN  5

#define ECHOPIN  18

#define GAS_PIN  34

#define BUZZER   13

#define WARN_LED 27


// Soil Moisture + Pump

#define SOIL_PIN 35

#define PUMP_PIN 4
```

```
// ==== Tank Setup ====

int emptyTankDistance = 70;  // cm (empty)

int fullTankDistance  = 30;  // cm (full)


// ==== Variables ====

float distance;

int waterLevelPer;

int gasLevel;

int soilMoist;

int lcdPage = 0;   // To cycle LCD pages

bool buzzerState = false;


// ==== Blynk V-pins ====

#define VPIN_WATER  V1

#define VPIN_DIST   V2

#define VPIN_GAS    V3

#define VPIN_WARNLED V4

#define VPIN_SOIL   V5

#define VPIN_PUMP   V6

float getDistance() {

 digitalWrite(TRIGPIN, LOW);

 delayMicroseconds(2);

 digitalWrite(TRIGPIN, HIGH);

 delayMicroseconds(20);

 digitalWrite(TRIGPIN, LOW);

 long duration = pulseIn(ECHOPIN, HIGH,
30000);

 float d = ((duration / 2.0) * 0.0343);

 return d; // cm

}
```

```
// Water level monitoring
void waterLevelCheck() {
  distance = getDistance();
  if (distance > 3 && distance < 400) {
    waterLevelPer = map((int)distance, emptyTankDistance, fullTankDistance, 0, 100);
    if (waterLevelPer > 100) waterLevelPer = 100;
    if (waterLevelPer < 0) waterLevelPer = 0;

    Blynk.virtualWrite(VPIN_WATER, waterLevelPer);
    Blynk.virtualWrite(VPIN_DIST, distance);
  }
}
// Gas monitoring
void gasCheck() {
  gasLevel = analogRead(GAS_PIN);
  gasLevel = map(gasLevel, 0, 4095, 0, 100);
  Blynk.virtualWrite(VPIN_GAS, gasLevel);
}

// Soil moisture monitoring
void soilCheck() {
  soilMoist = analogRead(SOIL_PIN);
  soilMoist = map(soilMoist, 0, 4095, 100, 0); // Invert
  Blynk.virtualWrite(VPIN_SOIL, soilMoist);
}
```

```
// Pump control from Blynk
BLYNK_WRITE(VPIN_PUMP) {
  int state = param.asInt();
  if (state == 1) {
    digitalWrite(PUMP_PIN, LOW); // Relay ON
  } else {
    digitalWrite(PUMP_PIN, HIGH); // Relay OFF
  }
}

// Shared buzzer + warning LED logic
void warningCheck() {
  bool waterAlert = (waterLevelPer >= 90);
  bool gasAlert   = (gasLevel >= 90);
  if (waterAlert || gasAlert) {
    digitalWrite(BUZZER, HIGH);
    digitalWrite(WARN_LED, HIGH);
    Blynk.virtualWrite(VPIN_WARNLED, 1);
  } else {
    digitalWrite(BUZZER, LOW);
    digitalWrite(WARN_LED, LOW);
    Blynk.virtualWrite(VPIN_WARNLED, 0);
  }
}
void lcdDisplay() {
  lcd.clear();
  if (lcdPage == 0) {
    lcd.setCursor(0, 0); lcd.print("Water: ");
    lcd.print(waterLevelPer); lcd.print("%");
    lcd.setCursor(0, 1); lcd.print("Dist: ");
    lcd.print(distance); lcd.print("cm");
  }
```

```
 else if (lcdPage == 1) {
   lcd.setCursor(0, 0); lcd.print("Gas Level: ");
   lcd.print(gasLevel); lcd.print("%");
   lcd.setCursor(0, 1);
   if (gasLevel >= 90) lcd.print("Warning!!!");
   else lcd.print("Normal");
  }
  else if (lcdPage == 2) {
   lcd.setCursor(0, 0); lcd.print("Soil: ");
   lcd.print(soilMoist); lcd.print("%");
   lcd.setCursor(0, 1); lcd.print("Pump: ");
   if (digitalRead(PUMP_PIN) == LOW)
lcd.print("ON ");
   else lcd.print("OFF");
  }
  lcdPage = (lcdPage + 1) % 3; // Next page
}
void setup() {
  Serial.begin(115200);

  pinMode(TRIGPIN, OUTPUT);
  pinMode(ECHOPIN, INPUT);
  pinMode(GAS_PIN, INPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(WARN_LED, OUTPUT);
  pinMode(PUMP_PIN, OUTPUT);
  digitalWrite(PUMP_PIN, HIGH);
lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0); lcd.print("Smart
Monitor");
  lcd.setCursor(0, 1); lcd.print("Starting...");
  delay(2000);
```

```
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass,
"blynk.cloud", 80);

  // Timers
  timer.setInterval(2000L, waterLevelCheck);
  timer.setInterval(2000L, gasCheck);
  timer.setInterval(2000L, soilCheck);
  timer.setInterval(2000L, warningCheck);
  timer.setInterval(3000L, lcdDisplay);
}

// ==================== LOOP
====================
void loop() {
  Blynk.run();
  timer.run();
}
```

IV. PROCEDURE

A. Preparation

- Gather the required components as stated in Chapter 2.
- Prepare the ESP32 on your preferred device.
- Review the problem for each topic and formulate or search for a circuit diagram given the problem of each topic.
- Simulate your circuit diagram to a circuit simulator such as Wokwi.

B. Actual

- For topic 1, connect the 5v and GND pins to the “+” and “-“ of the breadboard. Connect the VCC and GND pins of the 20x4 LED screen to the “+” and “-“ of the breadboard. Connected the SDA and SCL pins of the LED pins 21 and 22 of the ESP32. Connect the buzzer to pin 12 and the LED to pin 14. Connect the Trigger and Echo Pin of the ultrasonic sensor to pins 27 and 26 respectively. Connect the ESP32 to your laptop. Create your blynk template with ESP32 as the hardware and the connection type being WIFI. Create two data streams in Blynk with the first being pin V1, Integer data type, minimum value of 0 and maximum value of 100. The second datastream should be V2 pin, and the datatype being string. Automations should be added for these two data streams with the first being a sensor and the second being color. Set the first data stream to be online for conditions and actions. Set up the web dashboard by adding 1 gauge widget and 1 label widget, linking them both two the first and second datastreams respectively. You then add a device in Blynk and integrating this template to it. Afterwards, do the same for the web dashboard in mobile so that one could also control the circuit from the mobile application.
- For topic 2, connect the 5v and GND pins to the “+” and “-“ of the breadboard. Connect the VCC and GND pins of the 20x4 LED screen to the “+” and “-“ of the breadboard. Connected the SDA and SCL pins of the LED pins 21 and 22 of the ESP32. Connect the buzzer and the led with 220 ohms resistor to pin 13 of the ESP 32. Connect the VCC and GND of the MQ-2 Gas Sensor to the “+” and “-“ of the breadboard. Connect to Aout pin to pin 34 of the ESP32. Connect the

ESP32 to your laptop. Create your blynk template with ESP32 as the hardware and the connection type being WIFI. Create two data streams in Blynk with the first being pin V0, minimum value of 0 and maximum value of 100. The second datastream should be V1 pin and min and max being 0 and 1 respectively. Set up the web dashboard by adding 1 gauge widget and 1 LED widget, linking them both two the first and second datastreams respectively. You then add a device in Blynk and integrating this template to it. Afterwards, do the same for the web dashboard in mobile so that one could also control the circuit from the mobile application.

- For topic 3, connect the 5v and GND pins to the "+" and "-" of the breadboard. Connect the VCC and GND pins of the 20x4 LED screen to the "+" and "-" of the breadboard. Connected the SDA and SCL pins of the LED pins 21 and 22 of the ESP32. Connect the VCC and GND of the Soil Moisture Sensor to the "+" and "-" of the breadboard. The same goes for the VCC and GND of the Relay module. Connect the AO pin of the soil moisture sensor to pin 33. For the data pin of the relay module, connect it to pin 4. Create your blynk template with ESP32 as the hardware and the connection type being WIFI. Create two data streams in Blynk with the first being pin V0, minimum value of 0 and maximum value of 100. The second datastream should be V1 pin and min and max being 0 and 1 respectively. Set up the web dashboard by adding 1 gauge widget and 1 switch, linking them both two the first and second datastreams respectively. You then add a device in Blynk and integrating this template to it. Afterwards, do the same for the web dashboard in mobile so that one could also control the circuit from the mobile application.
- For topic 4, it simply is a combination of all three topics for both the circuit and blynk template. Take note for the pins specified in the code for the proper pins to be utilized and connected for topic 4.

C. Checking

- For Topic 1, ensure that when ultrasonic sensor is close or far enough the water, the led will show it is near or far the water, the warning, the percentage and distance as well as the LED and buzzer on high when these phenomena happen.
- For topic 2, ensure that upon a set percentage, the buzzer and led would go high while the gauge on the app shows the percentage and the led lighting up if the gas levels reach the threshold.
- For topic 3, ensure that the gauge from the blynk application and the 20x4 led shows the percentage of moisture. Furthermore, the switch from the blynk app should be able to switch the relay and open and close the LED connected at NC.
- For topic 4, ensure that all of the components and web dashboard elements from the Blynk IOT app are working.

D. Uploading

- Upload the final and corrected code to the ESP32.
- Ensure that topic 1, 2, 3, and 4 works properly.

V. CONCLUSION

In conclusion, this activity enhanced my understanding of how the ESP32 can integrate multiple sensors and modules to create efficient IoT-based monitoring systems. It strengthened my technical and problem-solving skills through hands-on circuit design, programming, and system troubleshooting. Overall, I learned how IoT technology bridges hardware and software to provide smart, automated, and remotely accessible solutions for real-world applications.

Name: Arenas, Joseph C.

Section: BET-CPET 3A

Activity No: 6

Topic	Date	Time	Signature
Water Level Sensor Blynk App & LCD	10/08/2025	12:25PM	
Gas Level Monitoring System with ESP 32	10/08/2025	12:25PM	
Plant Watering System with ESP32 & Blynk App.	10/08/2025	12:25PM	
Combination of water level, Gas level, & Plant watering system.	10/08/2025	1:15PM	R.N.A