SCHOOL OF COMPUTING
UUM COLLEGE OF ARTS AND SCIENCES

STTHK3113 SENSOR-BASED SYSTEMS (A)
SEMESTER 6 (A242)

MIDTERM EXAM:
TEMPERATURE AND HUMIDITY MONITORING WITH RELAY TRIGGER AND NEAR REAL-TIME GRAPH

28/5/2025 08:30 AM – 30/5/2025 8:30 AM
48 HOURS

PREPARED FOR :
AHMAD HANIS BIN MOHD SHABLI

PREPARED BY

NAME	MATRIC NO.
MOHAMMED UMAIR BIN MOHAMMED SUHAIMEE	295498

SUBMISSION DATE: 30TH MAY 2025

1.0 SYSTEM ARCHITECTURE

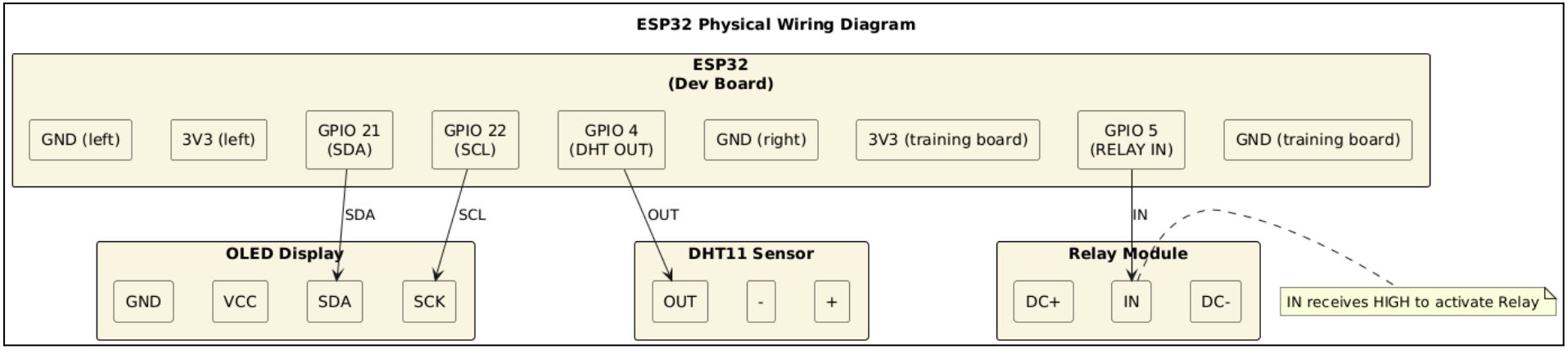

Figure 1: ESP32 Sensor Wiring Diagram

2.0 SETUP STEPS

REQUIRED COMPONENTS	
Component	**Purpose**
ESP32 Dev Board	Microcontroller
OLED Display (0.96" I2C)	Show readings (SDA/SCL)
DHT11 Sensor	Temperature & humidity sensor
Relay Module (DC 5V)	To trigger external device
Training Board	To share 3V3 and GND pins
Jumper Wires	To make all physical connections

OLED Display (I2C) Connection	
OLED Pin	**ESP32 Pin**
GND	GND (left side)
VCC	3V3
SCK	GPIO 22
SDA	GPIO 21

DHT11 Sensor Connection	
DHT11 Pin	**ESP32 Pin**
+	3V3
OUT	GPIO 4
SCK	GND (right side)

Relay Module Connection	
Relay Pin	**ESP32 Pin**
DC+	3V3 (shared with DHT11)
DC-	GND (training board ground)
IN	GPIO 5

3.0 SCREENSHOTS

readings

	id	device_id	temperature	humidity	alert	created_at
Edit Copy Delete	1	ESP32-01	31.9	80	0	2025-05-28 20:35:10
Edit Copy Delete	2	ESP32-01	31.8	80	0	2025-05-28 20:35:20
Edit Copy Delete	3	ESP32-01	31.8	80	0	2025-05-28 20:35:30
Edit Copy Delete	4	ESP32-01	31.8	80	0	2025-05-28 20:35:40
Edit Copy Delete	5	ESP32-01	31.9	80	0	2025-05-28 20:35:50
Edit Copy Delete	6	ESP32-01	31.9	80	0	2025-05-28 20:36:00
Edit Copy Delete	7	ESP32-01	31.9	80	0	2025-05-28 20:36:10
Edit Copy Delete	8	ESP32-01	31.9	80	0	2025-05-28 20:36:20
Edit Copy Delete	9	ESP32-01	31.8	80	0	2025-05-28 20:36:30
Edit Copy Delete	10	ESP32-01	31.9	80	0	2025-05-28 20:36:40
Edit Copy Delete	11	ESP32-01	31.9	79	0	2025-05-28 20:36:50
Edit Copy Delete	12	ESP32-01	31.9	79	0	2025-05-28 20:37:00
Edit Copy Delete	13	ESP32-01	31.9	79	0	2025-05-28 20:37:10
Edit Copy Delete	14	ESP32-01	31.9	79	0	2025-05-28 20:37:20
Edit Copy Delete	15	ESP32-01	31.9	79	0	2025-05-28 20:37:30

Figure 2: PHPMyAdmin readings table stored values

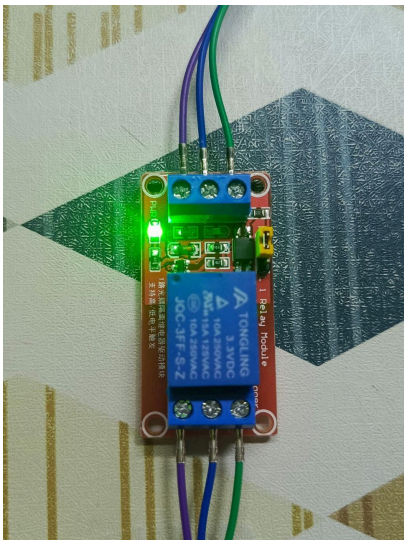

Figure 3a: Relay OFF Mode

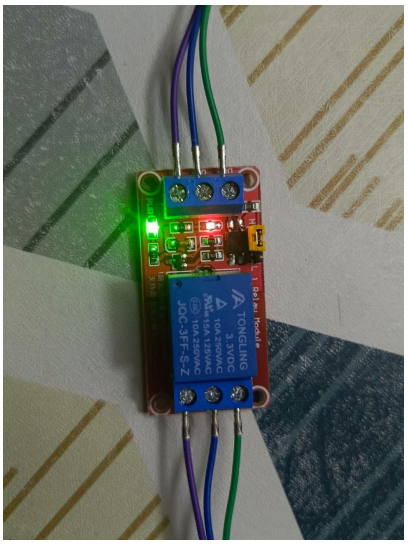

Figure 3b: Relay ON Mode

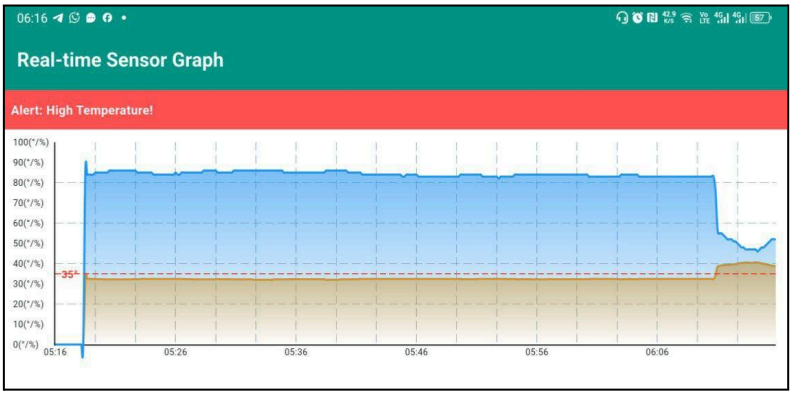

Figure 4: Near Real-time Sensor App

4.0 CHALLENGES & IMPROVEMENTS

During the development of my midterm assignment project, I faced several challenges that tested both my technical skills and problem-solving abilities. One of the main difficulties was ensuring smooth communication between the ESP32 microcontroller and the connected components such as the OLED display, DHT11 sensor, and the relay module. Initially, some components like the OLED would not display data as expected. After thorough troubleshooting, I discovered that setting the correct I2C parameters and using the appropriate text display methods such as `setTextWrap(false)` played a key role in resolving the display issue.

Another challenge was handling the data readings from the DHT11 sensor. The readings were sometimes unstable or returned as `NaN`, which required me to implement retry logic to get consistent temperature and humidity values. Additionally, the relay, which was controlled based on the temperature threshold, needed precise logic to avoid false triggering, especially when readings fluctuated close to the alert level.

Integrating real-time data into a Flutter mobile app introduced a new set of difficulties. One issue was parsing and displaying the sensor data in an interactive graph. I had to ensure that missing data points, especially during network hiccups, were handled gracefully. I also learned how to cache the last good state of the graph to avoid displaying an empty or misleading graph when fresh data was unavailable.

Time synchronization was another critical issue. The readings in the database were not aligning with the current local time due to timezone differences between PHP and MySQL. To fix this, I updated the `post_reading.php` script to manually set the timestamp using the "Asia/Kuala Lumpur" timezone, which ensured all data was correctly recorded and matched the graph's expectations.

Through these challenges, I improved my understanding of both hardware and software integration. I also learned how important it is to handle edge cases and errors gracefully. These improvements made my project more stable, responsive, and user-friendly, which I consider a valuable experience for my future in IoT development.