EE4216 Hardware for Internet of Things
Lab2

Men Jiying
A0239631A

1. Introduction

The objectives of this lab are:

- Interface with various sensors including DHT22, light sensor, and HC-SR04, and a relay module.
- Calibrate sensor readings and ensure accurate measurements.
- Display sensor data in the serial port terminal.

This report will document the steps taken to achieve these goals, along with design schematics, code, and results.

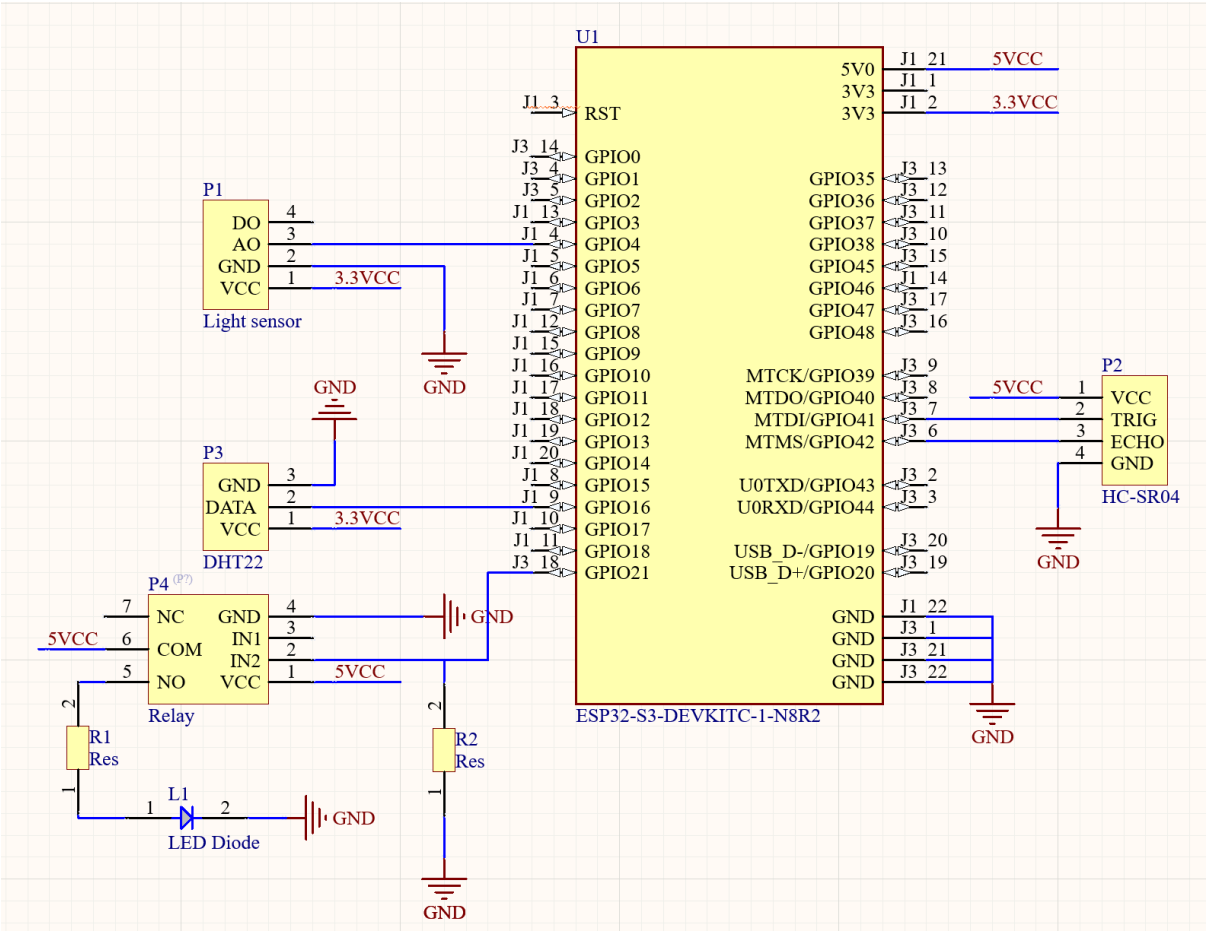
---

2. Materials and Setup

Materials Used:

- ESP32-S3 devkit board
- DHT22 sensor
- Light sensor
- HC-SR04 ultrasonic sensor
- Relay module
- Resistors (10KΩ, 1KΩ)
- Red LED diode
- Breadboard, Dupont wires, and USB cable

Circuit Design:



3. Lab Activities

3.1. HC-SR04 Ultrasonic Sensor

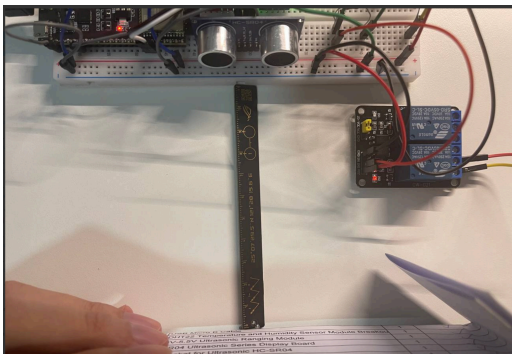
- Objective: Measure distance using the HC-SR04 ultrasonic sensor and ESP32-S3.
- Code:

```
#define TRIG_PIN 41  // Trigger pin for HC-SR04

#define ECHO_PIN 42  // Echo pin for HC-SR04

// Variables for calibration

float calibration_factor = 1.05;  // calibrated factor with 0.5cm precision

void setup() {

  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);

}

float measureDistance() {
```

```
  // Send a 10us pulse to the trigger pin
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // Measure the time for the echo pulse
  long duration = pulseIn(ECHO_PIN, HIGH);
  // Calculate distance in cm (speed of sound is 343m/s)
  float distance = (duration * 0.034) / 2;
  // Apply calibration factor
  distance *= calibration_factor;
  return distance;
}
void loop() {
  float distance = measureDistance();
  Serial.print("Measured Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);  // Delay 1 second between measurements
}
```

- Code Results

```
Measured Distance: 12.08 cm
Measured Distance: 12.08 cm
Measured Distance: 12.07 cm
Measured Distance: 12.42 cm
Measured Distance: 11.35 cm
Measured Distance: 11.39 cm
Measured Distance: 11.35 cm
Measured Distance: 11.71 cm
Measured Distance: 11.73 cm
Measured Distance: 11.71 cm
Measured Distance: 11.71 cm
Measured Distance: 11.35 cm
Measured Distance: 10.64 cm
Measured Distance: 11.35 cm
Measured Distance: 11.37 cm
Measured Distance: 11.35 cm
Measured Distance: 11.37 cm
```



3.2. Relay Module and LED Control Integrated with Light Sensor and DHT22

- Objective: Control the relay module to turn on/off the red LED based on room illumination, Meanwhile update humidity and temperature data every 5s and on demand.
- Code:

```
#include <Adafruit_Sensor.h>

#include <DHT.h>

// Define sensor and relay pins

#define DHTPIN 16

#define DHTTYPE DHT22

#define LIGHT_SENSOR_ANAPIN 4

#define RELAY_PIN 21

DHT dht(DHTPIN, DHTTYPE);

// Timer variables

unsigned long previousMillis = 0;  // Stores the last time temperature was updated

const long interval = 5000;       // Interval at which to read temperature (5 seconds)

void setup() {

  Serial.begin(115200);

  dht.begin();

  pinMode(RELAY_PIN, OUTPUT);
```

```
}

void loop() {
  unsigned long currentMillis = millis();  // Current time
  // Read the light sensor and control the relay (respond immediately)
  int lightValue = analogRead(LIGHT_SENSOR_ANAPIN);
  // Immediately control relay based on light conditions
  if (lightValue < 500) {
    digitalWrite(RELAY_PIN, HIGH);  // Turn on relay if low light detected
  } else {
    digitalWrite(RELAY_PIN, LOW);   // Turn off relay if high light detected
  }
  // Read humidity, temperature, and light values every 5 seconds
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature()+ temperatureOffset;
    if (isnan(humidity) || isnan(temperature)) {
      Serial.println(F("Failed to read from DHT sensor!"));
    } else {
      float heatIndex = dht.computeHeatIndex(temperature, humidity, false);
      // Update the sensor data
      Serial.print("Humidity: ");
      Serial.print(humidity);
      Serial.print(" %\t Temperature: ");
      Serial.print(temperature);
```

```
      Serial.println(" °C");
    }
    // Print the light sensor values every 5 seconds
    Serial.print("Analog Light Value: ");
    Serial.println(lightValue);
  }
  // Check for request to display temperature on demand
  if (Serial.available() > 0) {
    String request = Serial.readString();
    request.trim();  // Remove any extra spaces or newline characters
    if (request == "temperature") {
      float temperature = dht.readTemperature()+ temperatureOffset;
      Serial.print(F("Current Temperature: "));
      Serial.print(temperature);
      Serial.println(F("°C"));
    }
  }
}
```

- Code Results

```
Analog Light Value: 1100
Humidity: 53.20 %        Temperature: 28.90 ° C
Analog Light Value: 1117
Humidity: 53.20 %        Temperature: 28.80 ° C
Analog Light Value: 1116
Humidity: 53.30 %        Temperature: 28.80 ° C
Analog Light Value: 1109
Current Temperature: 28.80° C
```