**Week 4 Lab 2 Report**

**Name:** Javon Teo Tze Kai

**Student ID:** A0233706J

**Lab Group:** B02

**Task 1:**

myservo.setPeriodHertz(50);

myservo.attach(servoPin, 1000, 2000);

void loop() {

  for (pos = 0; pos <= 90; pos += 2) {

// for left to center

    myservo.write(pos);

    delay(15);

  }

delay(1000);

<for loop is repeated for 90-180 (center to right), then goes down in 180-90 (right to center), then 90-0 (center to left)> }

**Task 2:**

The RGBC Data register in the first version of the program stores the clear light intensity value measured.

All of the values of the RGBC Data register are (2313, 1028, 2057, 1542), representing the clear, red, green, and blue light intensity data measured by the APDS9960 respectively.

    for (uint8\_t reg = 0x94; reg < 0x9C; reg+=2) {

      Wire.beginTransmission(address);

      Wire.write(reg);

      Wire.endTransmission();

      Wire.requestFrom(address, 1);

      if(Wire.available()){

          LSByte = Wire.read();

      }

      Wire.beginTransmission(address);

      Wire.write(reg);

      Wire.endTransmission();

      Wire.requestFrom(address, 1);

      if(Wire.available()){

          MSByte = Wire.read();

      }

regValue = (MSByte<<8) + LSByte;

      Serial.print(reg);

      Serial.print(" Register Value: ");

      Serial.println(regValue);

      // Wait 1 second before next reading

      delay(1000);

    }

**Task 3:**

I added additional code to move the servo motor to the left (at position 180) when APDS9960 detects a “LEFT”, and to move the servo motor to the right (at position 0) when APDS9960 detects a “RIGHT”.

case DIR\_LEFT:

        Serial.println("LEFT");

        myservo.write(0);

        break;

      case DIR\_RIGHT:

        Serial.println("RIGHT");

        myservo.write(180);

        break;

**Task 4:**

I added this code in the setup() function:

  server.on("/toggleLED", []() {

    digitalWrite(led, !digitalRead(led));

    if (digitalRead(led) == HIGH) {

      server.send(200, "text/plain", "LED On");

    } else {

      server.send(200, "text/plain", "LED Off");

    }

  });

**Task 5:**

I used the DHT11 sensor to measure the current temperature and humidity detected by the sensor. I created a getMeasurements function to measure the data and return in a struct measurements (containing temp and humidity). Then this getMeasurements function will be called in my loop function:

    struct measurements mmts = getMeasurements();

    String url = laptopAt;

    url += "/data?data1={temp}&data2={humidity}";

    // Replace the placeholders with actual values

    url.replace("{temp}", String(mmts.temp));

    url.replace("{humidity}", String(mmts.humidity));

This way, the values for temp and humidity are the new payload, replacing the original hardcoded values of 31.2 and 76 respectively. The payload is sent using the HTTP GET request from the ESP32 to the laptop. Hence, the laptop is now able to receive the recorded temperature and humidity.