To: Christopher Peters
From: Abdullah Khalid

Purpose

The purpose of this rework was to modify Assignment 5 to use Station Mode on the Arduino Uno R4 WiFi board instead of Access Point mode. The objective was to enable full control and interaction from a mobile phone browser, allowing the user to access live sensor data and control onboard LEDs wirelessly via a Wi-Fi network.

Methodology / Approach

In this version of the project, the Arduino connects to a Wi-Fi network in Station Mode and hosts a web server to serve dynamic interfaces. In Part A, the Arduino reads data from the GY-87 (IMU, barometer, magnetometer) and HC-SR04 ultrasonic sensor, then displays temperature, pressure, altitude, heading, and distance in a live HTML page accessed through a phone browser. In Part B, the Arduino serves a different HTML page with controls to toggle the blue LED on/off and a slider to adjust the brightness of the red LED via PWM. Both pages are responsive and controlled entirely through a mobile browser over the same Wi-Fi network, with all inputs processed using HTTP GET requests.

Results

The wireless system was successful overall. Every action on the website interacted perfectly and quickly with the Arduino board.

Appendix

```
Part A:

#include <WiFiS3.h>
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_BMP085.h>
#include <QMC5883LCompass.h>

#define TRIG_PIN 10
#define ECHO_PIN 9

Adafruit_MPU6050 mpu;
Adafruit_BMP085 bmp;
QMC5883LCompass compass;
```

```
char ssid[] = "Abdullah Khalid";
char pass[] = "QSPS2992@is2992";

WiFiServer server(80);

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.begin();
  bmp.begin();
  compass.init();

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("\nWiFi connected!");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (client) {
    String request = client.readStringUntil('\r');
    client.flush();

    String html = "<html><head><title>Sensor Data</title></head><body>";
    html += "<h1>Live Sensor Readings</h1>";
    html += getSensorData();
    html += "</body></html>";

    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.print(html);

    client.stop();
  }
}
```

```
String getSensorData() {
  sensors_event_t a, g, t;
  mpu.getEvent(&a, &g, &t);
  compass.read();
  float temp = bmp.readTemperature();
  float pressure = bmp.readPressure();
  float altitude = bmp.readAltitude();
  float distance = readDistanceCM();
  int heading = compass.getAzimuth();
  char dir[4]; compass.getDirection(dir, heading);

  String data = "";
  data += "Temperature: " + String(temp, 1) + " °C<br>";
  data += "Pressure: " + String(pressure / 100.0, 1) + " hPa<br>";
  data += "Altitude: " + String(altitude, 1) + " m<br>";
  data += "Distance: " + String(distance, 1) + " cm<br>";
  data += "Heading: " + String(heading) + "° (" + String(dir) + ")<br>";
  return data;
}

float readDistanceCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.0343 / 2;
}


Part B:

#include <WiFiS3.h>

const int redLED = 5;     // PWM capable
const int blueLED = 4;

char ssid[] = "Abdullah Khalid";
char pass[] = "QSPS2992@is2992";

WiFiServer server(80);

void setup() {
  Serial.begin(9600);
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("\nWiFi connected!");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (client) {
    String request = client.readStringUntil('\r');
    client.flush();

    // Handle red LED brightness
    if (request.indexOf("red=") != -1) {
      int valIndex = request.indexOf("red=") + 4;
      int ampIndex = request.indexOf('&', valIndex);
      String valStr = (ampIndex == -1) ? request.substring(valIndex) :
request.substring(valIndex, ampIndex);
      int redValue = valStr.toInt();
      analogWrite(redLED, constrain(redValue, 0, 255));
    }

    // Handle blue LED ON/OFF
    if (request.indexOf("blue=on") != -1) digitalWrite(blueLED, HIGH);
    if (request.indexOf("blue=off") != -1) digitalWrite(blueLED, LOW);

    // HTML Response
    String html = "<html><head><title>LED Control</title></head><body>";
    html += "<h1>Control LEDs</h1>";

    // Red LED Slider
    html += "Red Brightness (0-255):<br>";
    html += "<form method='GET'>";
    html += "<input type='range' name='red' min='0' max='255' value='128'
onchange='this.form.submit()'><br><br>";
    html += "</form>";

    // Blue LED buttons
    html += "<a href='/?blue=on'>Blue ON</a><br>";
    html += "<a href='/?blue=off'>Blue OFF</a><br>";
    html += "</body></html>";

    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.print(html);

    client.stop();
  }
}
```