Appendix A Arduino Code for ESP8266 NodeMCU with ThingSpeak Integration

The following code demonstrates how to use the ESP8266 NodeMCU to collect temperature data from a DS18B20 sensor and send simulated Temperature and Resistance (R2) values to ThingSpeak. The code includes Wi-Fi connection management and a calibration adjustment for the temperature sensor.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

#define ONE_WIRE_BUS D2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

const float Vin = 3.3;
const float Rfixed = 330.0;
const float analogPin = A0;

// WiFi Setup
const char* ssid = "Pie";
const char* password = "Napie001";

// ThingSpeak Configuration
// Ensure your ThingSpeak channel
has at least 2 fields enabled for this version:
// Field 1: Measured Temperature
// Field 2: Measured Sample Resistance
unsigned long myChannelNumber = 2958251;
const char* myWriteAPIKey = "JJLT5AF5B5CMCZ8C";
WiFiClient client;
```

```
void setup() {
  Serial.begin(9600);
  delay(1000);

  sensors.begin();


  // Connect to Wi-Fi
  Serial.print("Connecting to WiFi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("Time (s) \tTemperature (C)
  \tResistance (Ohm) \tAnalog Value (ADC)");

  // Initialize ThingSpeak
  ThingSpeak.begin(client);
}

void loop() {
  unsigned long currentTime = millis();

  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);

  int adcVal = analogRead(analogPin);
  float Vout = (adcVal* Vin / 1023.0) ;
  float Rsample_measured = (Vout * Rfixed) / (Vin - Vout);

  Serial.print(currentTime / 1000.0, 2);
  Serial.print("\t \t \t");
  Serial.print(temperatureC, 2);
  Serial.print("\t \t \t");
  Serial.print(Rsample_measured, 3);
  Serial.print("\t \t \t");
  Serial.println(adcVal);

  // Write to ThingSpeak
  // Set Field 1 to temperatureC
  ThingSpeak.setField(1, temperatureC);
  // Set Field 2 to Rsample_measured
  ThingSpeak.setField(2, Rsample_measured);

  // Write the fields to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);


  delay(20000);
}
```

Appendix B Data For Titanium

B.0.1 Temperature and Resistance Data (Primary Set)

Temperature (°C)	Experimental Resistance (Ω)	Theoretical Resistance (Ω)
43.50	12.731	14.5640
45.50	13.079	14.6656
48.50	13.428	14.8180
52.75	13.428	15.0339
61.25	14.775	15.4657
69.94	16.664	15.9412
79.94	16.747	16.5120
87.19	15.528	16.9271
92.38	15.537	17.2111
93.81	15.537	17.2793

Table B.1 Comparison of experimental and theoretical resistance values of Titanium as a function of temperature.

B.0.2 Resistance Data (Around 20 °C)

Temperature (°C)	Resistance(R (Ω))
20.00	11.345
20.13	11.345
20.25	11.345
20.38	10.656
20.50	10.656
20.63	10.656
20.75	10.656

Table B.2 Measured resistance values for Titanium around 20 °C used for calculating R_0.

Appendix C   Data For Tungsten

C.0.1 Temperature and Resistance Data (Primary Set)

Temperature (°C)	Experimental Resistance (Ω)	Theoretical Resistance (Ω)
29.81	5.910	0.8375
29.94	5.910	0.8380
30.13	6.006	0.8387
33.63	6.245	0.8513
41.69	6.246	0.8804
48.63	6.379	0.9056
51.44	6.247	0.9161
56.75	7.043	0.9363
62.94	7.246	0.9588
78.75	8.006	1.0155
86.25	8.396	1.0414
92.38	8.006	1.0632
93.94	9.006	1.0690

Table C.1   Comparison of experimental and theoretical resistance values of Tungsten as a function of temperature.

C.0.2 Resistance Data (Around 20 °C)

Temperature (°C)	Resistance(R (Ω))
20.13	6.245
20.25	6.245
20.38	6.245
20.50	6.245
20.63	5.910
20.81	5.910
20.94	5.910

Table C.2   Measured resistance values for tungsten around 20 °C used for calculating R_0.