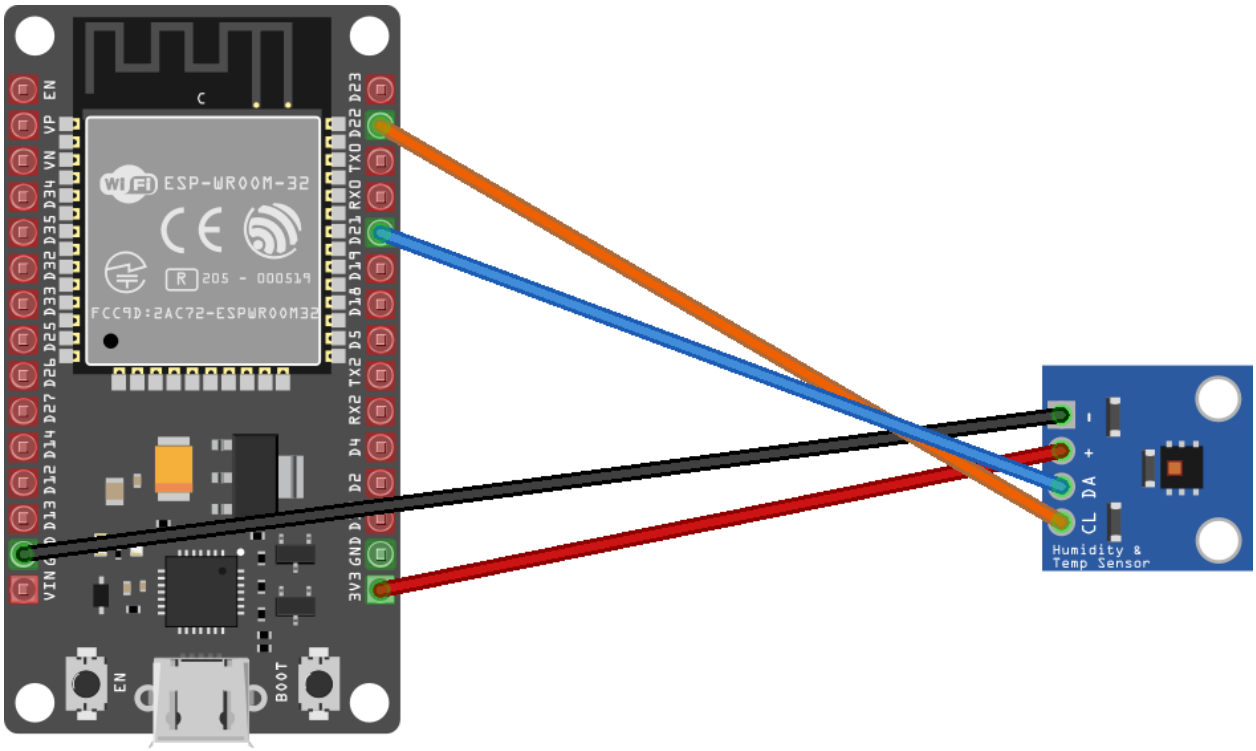
Interfacing HDC1080 RTH sensor with ESP32

Ebrahim Mamawala
25th May, 2024
Living Things: Embedded Interview Question 1

Steps taken to setup Arduino IDE to interface ESP32 with HDC1080 RTH sensor

1. Go to Board Manager, search ESP32 and install the board library
2. Go to Sketch > Include Library > Manage Libraries and search for “HDC1080" and install the HDC1080 library by Adafruit.
3. Go to Sketch > Include Library > Manage Libraries and search for “ArduinoJson” and install the library.
4. Write the code to interface with the HDC1080 sensor using the appropriate libraries and functions

Connection Diagram of ESP32 and HDC1080 RTH sensor



ESP 32	HDC1080
3.3V	VCC
GND	GND
GPIO21	SDA
GPIO22	SCL

Code to read temperature and sensor data from sensor and publish data through MQTT using Mosquitto open source MQTT Broker

```
#include <ArduinoJson.h>
#include <Wire.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include "ClosedCube_HDC1080.h" // Include the HDC1080 library

#define wifi_ssid “" // Add WiFi SSID
#define wifi_password “" // Add WiFi Password
#define mqtt_server “" //Provide the address or hostname of MQTT broker here

#define mqtt_topic "sensor/HDC1080"

WiFiClient espClient;
PubSubClient client(espClient);
ClosedCube_HDC1080 hdc1080; // Initialize HDC1080 sensor object

void setup() {
  Serial.begin(9600);
  Wire.begin(); // Initialize I2C communication
  hdc1080.begin(0x40); // HDC1080 address
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(wifi_ssid);

  WiFi.begin(wifi_ssid, wifi_password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");

    if (client.connect("ESP32Client")) {
      Serial.println("connected");
      client.subscribe(mqtt_topic);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
```

```
    }
  }
}

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  float temperature = hdc1080.readTemperature();
  float humidity = hdc1080.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from HDC1080 RTH sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  //Create a JSON object
  DynamicJsonDocument jsonDoc(256);

  // Add temperature and humidity data to JSON object
  jsonDoc["temperature"] = temperature;
  jsonDoc["humidity"] = humidity;

  // Serialize JSON object to a string
  char jsonString[100];
  serializeJson(jsonDoc, jsonString);

  // Publish JSON-formatted sensor data
  client.publish(mqtt_topic, jsonString, true);

  delay(2000); // Delay between readings
}
```

Block Diagram of the project