|  |  |  |
| --- | --- | --- |
| **المملكة العربية السعودية**  **وزارة التعليم العالي**  **جامعة الملك فيصل**  **كلية علوم الحاسب وتقنية المعلومات** | A logo of a university  Description automatically generated | **KINGDOM OF SAUDI ARABIA**  **Ministry of Higher Education**  **King Faisal University**  **College of Computer Sciences & Information Technology** |

**Arduino-based Wireless Weather Station**

**Mobile and Wireless Networks**

|  |  |
| --- | --- |
| Ali Ibrahim Burasis | 220027320 |
| Alabbas Alessa | 221441026 |

Instructors: Dr. Abdullah Albuali.

.

* **Introduction:**

think about As you are engineer at STC, you have been entrusted with the development project aimed at Benefit mobile and wireless technology. One of them Areas of interest in creating an Arduino based wireless weather station. Our project is designed to take real-time weather data using advanced sensors that project this information over the STC wireless network. The aim is to provide accurate, including climate monitoring.

* **Project Overview:**

In this project, we learn how to make a LoRa-based wireless weather station using an Arduino Pro Mini and an ESP32 Wi-Fi module. You can use sensors like the BME280 barometric pressure sensor along with a BH1750 light sensor and a rain sensor. Basically, this weather station can monitor environmental parameters like temperature, humidity, pressure, altitude, dew point, rainfall, and light intensity.

* **System Components:**

1. Arduino nano
2. Bmp180 sensor.
3. Raindrops Detection Sensor.
4. 2. LoRaWAN module.
5. Arduino: Esp32 development board.
6. Battery

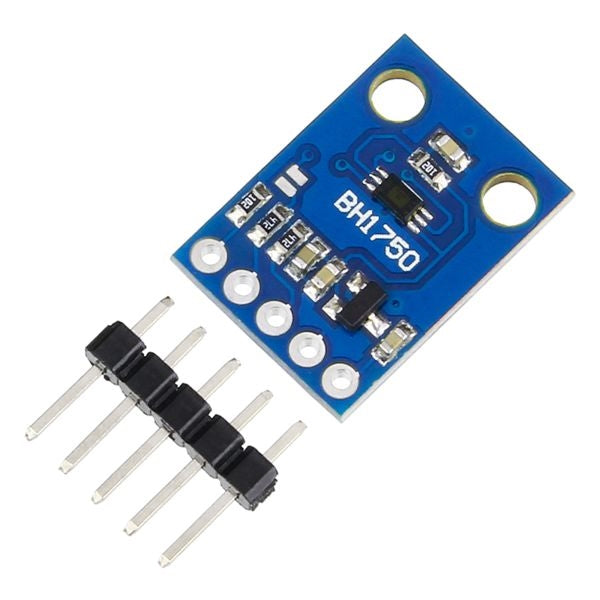
* **Circuit Documentation:**

This circuit documentation summarizes the integration of various components to perform specific, though unspecified, functions. The main controller for the circuit is an Arduino Nano, interfacing with several sensors and modules. An Adafruit BMP180 sensor is used for barometric pressure and temperature measurements, a YL-83 Rain Sensor detects rain, and a LoRa Ra-02 SX1278 module enables long-range communication. The entire circuit is powered by an 850mAh Polymer Lithium Ion Battery. The component list includes: the Arduino Nano, a compact microcontroller board based on the ATmega328P with digital and analog I/O pins and serial communication support; the Adafruit BMP180, a digital barometric pressure and temperature sensor that communicates over I2C; the YL-83 Rain Sensor, which includes a control board for processing signals and a detection board for sensing rain through changes in conductivity; the LoRa Ra-02 SX1278, a long-range communication module based on the SX1278 chip; and the 850mAh Polymer Lithium Ion Battery for power. The wiring details specify the connections among these components. The Arduino Nano connects to the Adafruit BMP180 via A5 (SCL) and A4 (SDA) pins, and to the YL-83 Rain Sensor's control board via A0 (AO). The LoRa module connects to various digital pins on the Arduino, including D10 (NSS), D11/MOSI, D13/SCK, D6 (RST), and D2 (DI00). The common ground and power connections ensure proper functioning of the entire circuit.

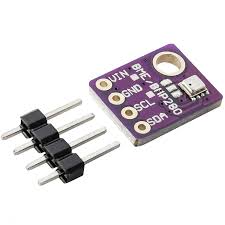
Software Setup:

Coding: writing a Arduino sketch to initialize sensors, read data, and send it to the LoRa module, Include error handling for sensor failures.

* Some figures for components;

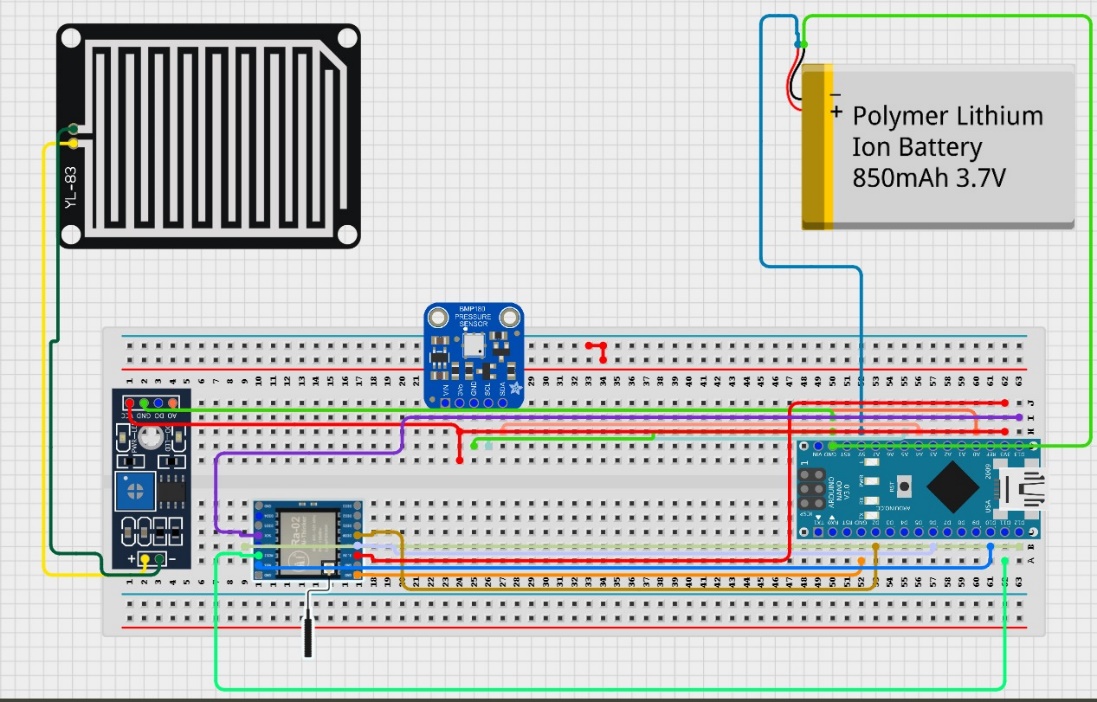
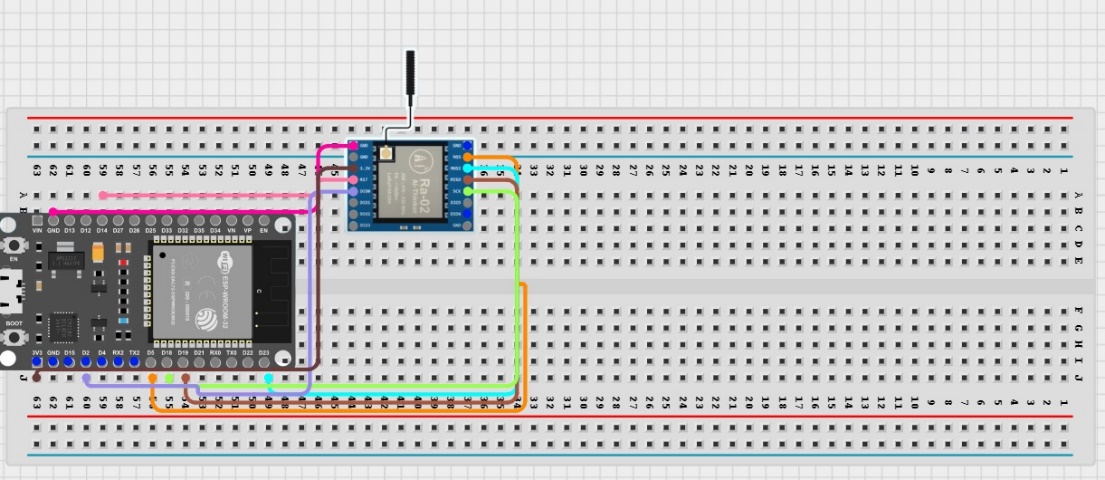
 

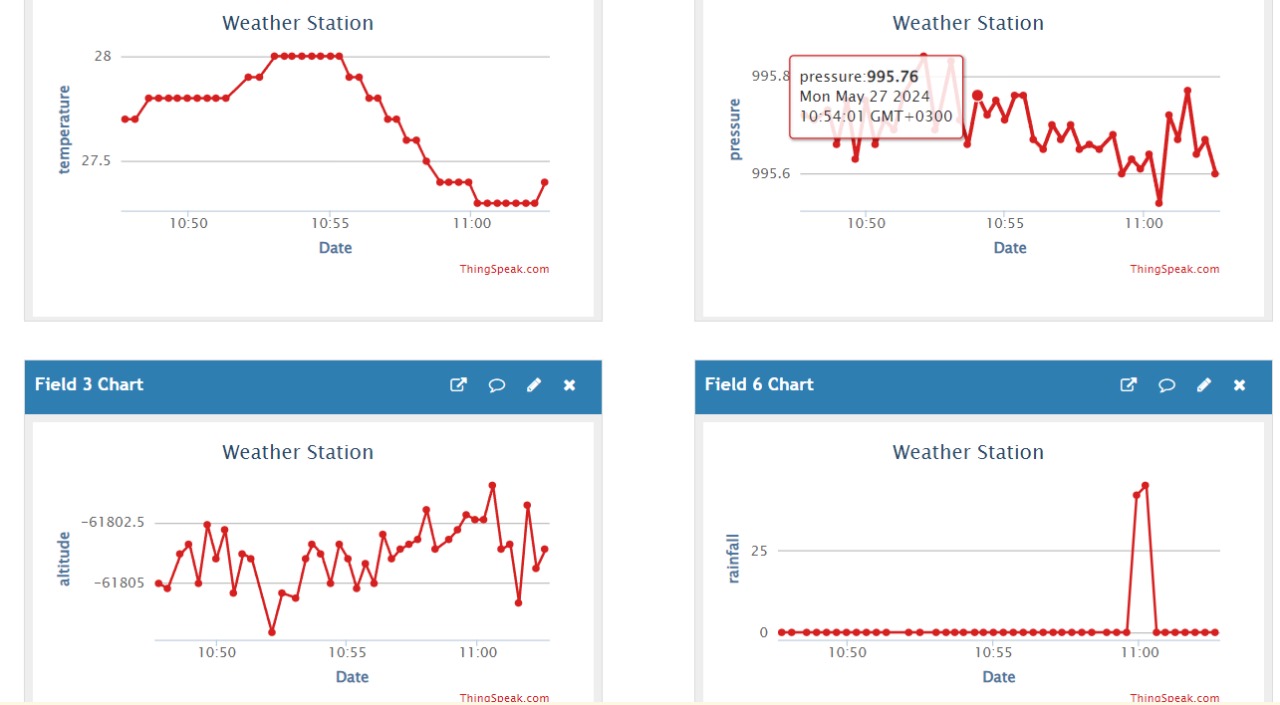
BH1750 Arduino Pro Mini

LoRa Module Sensors: BME280

* **Design :**

****

* **Output:**
* **Code:**

**Sender**

#include<Wire.h>

#include <SPI.h>

#include <LoRa.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BMP085.h> // Change to Adafruit\_BMP180 library

#define BAND 433E6

#define rain\_sensor A0

#define SEALEVELPRESSURE\_HPA (1013.25)

Adafruit\_BMP085 bmp; // Change to Adafruit\_BMP180

String LoRaMessage = "";

char device\_id[12] = "MyDevice123";

void setup() {

Serial.begin(115200);

Wire.begin();

pinMode (rain\_sensor, INPUT);

while (!Serial);

Serial.println(F("LoRa Sender"));

if (!LoRa.begin(BAND)) {

Serial.println(F("Starting LoRa failed!"));

while (1);

}

if (!bmp.begin()) { // Change to bmp.begin() for BMP180

Serial.println("Could not find a valid BMP180 sensor, check wiring!");

while (1);

}

}

void loop() {

float temperature = bmp.readTemperature(); // Change to bmp.readTemperature() for BMP180

float pressure = bmp.readPressure() / 100.0F; // Change to bmp.readPressure() for BMP180

float altitude = bmp.readAltitude();

int rainfall = map(analogRead(rain\_sensor), 780, 0, 0, 100);

if (rainfall >= 100) {

rainfall = 100;

}

if (rainfall <= 0) {

rainfall = 0;

}

Serial.print(F("Device ID: "));

Serial.println(device\_id);

Serial.print(F("Temperature = "));

Serial.print(temperature);

Serial.println(F("\*C"));

Serial.print(F("Pressure = "));

Serial.print(pressure);

Serial.println(F("hPa"));

Serial.print(F("Approx. Altitude = "));

Serial.print(altitude);

Serial.println(F("m"));

Serial.print(F("Rainfall = "));

Serial.print(rainfall);

Serial.println(F("%"));

Serial.println();

LoRaMessage = String(device\_id) + "/" + String(temperature) + "&" + String(pressure)

+ "#" + String(altitude) + "@"

+ "^" + String(rainfall) + "!" ;

// send packet

LoRa.beginPacket();

LoRa.print(LoRaMessage);

LoRa.endPacket();

delay(10000);

}

**Receiver**

#include <SPI.h>

#include <LoRa.h>

#include <WiFi.h>

String apiKey = "JL0D44OC494NWMYT"; // Enter your Write API key from ThingSpeak

const char\* ssid = "HUAWEI nova 3e";

const char\* password = "123456789";

const char\* server = "api.thingspeak.com";

#define SS 5

#define RST 14

#define DI0 2

//#define TX\_P 17

#define BAND 433E6

//#define ENCRYPT 0x23

String device\_id;

String temperature;

String pressure;

String altitude;

String rainfall;

WiFiClient client;

void setup()

{

Serial.begin(115200);

Serial.println("LoRa Receiver");

//LoRa.setTxPower(TX\_P);

//LoRa.setSyncWord(ENCRYPT);

LoRa.setPins(SS, RST, DI0);

if (!LoRa.begin(BAND))

{

Serial.println("Starting LoRa failed!");

while (1);

}

Serial.println("Connecting to ");

Serial.println(ssid);

//connect to your local wi-fi network

WiFi.begin(ssid, password);

//check wi-fi is connected to wi-fi network

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected..!");

Serial.print("Got IP: ");

Serial.println(WiFi.localIP());

}

void loop()

{

// try to parse packet

int pos1, pos2, pos3, pos4, pos5, pos6, pos7;

int packetSize = LoRa.parsePacket();

if (packetSize)

{

// received a packet

Serial.print("Received packet: ");

String LoRaData = LoRa.readString();

Serial.print(LoRaData);

// read packet

while (LoRa.available())

{

Serial.print((char)LoRa.read());

}

// print RSSI of packet

Serial.print("' with RSSI ");

Serial.println(LoRa.packetRssi());

pos1 = LoRaData.indexOf('/');

pos2 = LoRaData.indexOf('&');

pos3 = LoRaData.indexOf('#');

pos4 = LoRaData.indexOf('@');

pos5 = LoRaData.indexOf('$');

pos6 = LoRaData.indexOf('^');

pos7 = LoRaData.indexOf('!');

device\_id = LoRaData.substring(0, pos1);

temperature = LoRaData.substring(pos1 + 1, pos2);

pressure = LoRaData.substring(pos2 + 1, pos3);

altitude = LoRaData.substring(pos3 + 1, pos4);

humidity = LoRaData.substring(pos4 + 1, pos5);

dewPoint = LoRaData.substring(pos5 + 1, pos6);

rainfall = LoRaData.substring(pos6 + 1, pos7);

lux = LoRaData.substring(pos7 + 1, LoRaData.length());

Serial.print(F("Device ID = "));

Serial.println(device\_id);

Serial.print(F("Temperature = "));

Serial.print(temperature);

Serial.println(F("\*C"));

Serial.print(F("Pressure = "));

Serial.print(pressure);

Serial.println(F("hPa"));

Serial.print(F("Approx. Altitude = "));

Serial.print(altitude);

Serial.println(F("m"));

Serial.print(F("Rainfall = "));

Serial.print(rainfall);

Serial.println(F("%"));

Serial.println();

if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = apiKey;

postStr += "&field1=";

postStr += String(temperature);

postStr += "&field2=";

postStr += String(pressure);

postStr += "&field3=";

postStr += String(altitude);

postStr += "&field4=";

postStr += String(rainfall);

postStr += "&field7=";

postStr += "r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.println("Data Send to Thingspeak");

delay(500);

}

client.stop();

Serial.println("Waiting...");

}

}

* **Conclusion:**

Finally, the Arduino-based wireless weather station with ESP32 and LoRa is a versatile and auxiliary system for monitoring environmental conditions. Its long-range communication capabilities and variety of sensors make it suitable for remote applications. Ensures proper setup, calibration, and integration with data visualization tools to produce reliable and useful weather data reports.

* **References:**
* STC Wireless Network Specifications. (2023). STC Technical Documentation.
* <https://iotprojectsideas.com/lora-based-wireless-weather-station-using-arduino-esp32/>
* <https://github.com/Ali-Burasis/Arduino-based-Wireless-Weather-Station-Mobile-and-Wireless-Networks>