⚡ BlitzTech Learn

Building a Smart Weather Station with ESP32

Combine an ESP32, DHT22, and BMP280 to build a WiFi-connected weather station that publishes to MQTT and shows on your phone. End-to-end ~60 minutes.

Intermediate | 60 min | Sensors & IoT | Jun 26, 2026 | blitztechelectronics.co.zw/learn/smart-weather-station-with-esp32/

This is the canonical 'first real IoT project', temperature, humidity, pressure, all WiFi-connected, all visible on your phone. We use it as the starting template for our BlitzAgroTech smart weather stations deployed across Zimbabwean farms.

1 Parts and prep

- ESP32 DevKit-C (or any ESP32 board with WiFi)
- DHT22, temperature + humidity
- BMP280, barometric pressure + temperature (I2C)
- Breadboard, jumper wires, 10 kΩ resistor
- An MQTT broker, we recommend a free HiveMQ Cloud account (1 GB free, plenty for a personal weather station)

2 Wiring

ESP32 → DHT22:

3V3 → VCC

GPIO 4 → DATA (with 10 kΩ pull-up to 3V3)

GND → GND

ESP32 → BMP280 (I2C):

3V3 → VCC

GND → GND

GPIO 21 → SDA

GPIO 22 → SCL

3 Arduino IDE setup for ESP32

Arduino IDE → File → Preferences → Additional Boards URLs:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Then Tools → Board → Boards Manager → search 'esp32' → install Espressif esp32. Also install these libraries: 'DHT sensor library', 'Adafruit BMP280', 'PubSubClient' (for MQTT).

4 The firmware

Replace WIFI_SSID, WIFI_PASS, MQTT_HOST, MQTT_USER, MQTT_PASS with your values before uploading. Open Serial Monitor at 115200 baud to watch it connect and publish.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <Adafruit_BMP280.h>

#define WIFI_SSID  "YourWiFi"
#define WIFI_PASS  "YourPass"
#define MQTT_HOST  "xxxxx.s1.eu.hivemq.cloud"
#define MQTT_PORT  8883
#define MQTT_USER  "user"
#define MQTT_PASS  "pass"
#define DHTPIN 4
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP280 bmp;
WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  dht.begin();
  bmp.begin(0x76);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) { delay(500); Serial.print("."); }
  client.setServer(MQTT_HOST, MQTT_PORT);
}

void loop() {
  if (!client.connected()) {
    client.connect("blitztech-weather", MQTT_USER, MQTT_PASS);
  }
  client.loop();
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float p = bmp.readPressure() / 100.0;
  char buf[128];
  snprintf(buf, sizeof(buf),
    "{\"temp\":%.1f,\"humidity\":%.1f,\"pressure\":%.1f}", t, h, p);
  client.publish("blitztech/weather", buf);
  Serial.println(buf);
```

```
  delay(30000);
}
```

5 View on your phone

Install the free 'IoT MQTT Panel' app on your phone (Android/iOS). Add a connection to your HiveMQ broker with the same credentials. Subscribe to 'blitztech/weather' and add Number widgets bound to the temp / humidity / pressure JSON keys. Refresh every 30 seconds. That's a deployed IoT device.

6 Where this goes next

- Add a solar panel + LiPo battery for outdoor deployment.

- Add a soil-moisture sensor and run from low-power deep-sleep on a 6-hour cadence to monitor a Zimbabwean field for a year on one battery charge.

- This is the basic platform for BlitzAgroTech smart weather stations. Contact us if you want to deploy this at agricultural scale.