**Whether monitoring using ESP32**

# Introduction:

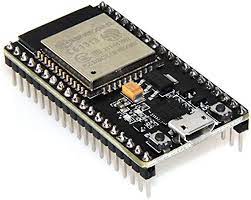
Using a single ESP32 board and two sensors, we will create a full-fledged weather station that monitors the weather conditions with micro python.

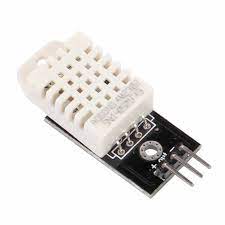
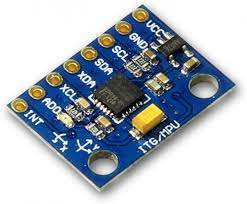
**Components:**

# mpu6050 6-Axis Accel & Gyro Sensor

* DHT 22 Sensor (2nos)
* ESP32

ESP 32



DHT22

Gyro sensor

### Connection:

Both VCC of the DHT22 sensors are connected to the esp:3v3 and also VCC of gyro sensor connected to esp:3v3

SDA of DHT22 (1) is connected into esp:D15

SDA of DHT22 (2) is connected into esp:D2

GND pin of the both DHT22 and gyro sensor is connected into the esp:GND.1

Imu1 :SCL is connected to the esp:D22&imu:SDA is connected to the esp :D21

Program:

below python program is used for this simulation

Below python program is using for this simulation import network

import time

from machine import Pin

import dht

import ujson

from umqtt.simple import MQTTClient

# MQTT Server Parameters

MQTT\_CLIENT\_ID = "demoErick"

MQTT\_BROKER    = "mqtt-dashboard.com"

MQTT\_USER      = ""

MQTT\_PASSWORD  = ""

MQTT\_TOPIC     = "iotunam"

sensor1 = dht.DHT22(Pin(15))

sensor2 = dht.DHT22(Pin(2))

# WIFI Connection

print("Connecting to WiFi", end="")

sta\_if = network.WLAN(network.STA\_IF)

sta\_if.active(True)

sta\_if.connect('Wokwi-GUEST', '')

while not sta\_if.isconnected():

  print(".", end="")

  time.sleep(0.1)

print(" Connected!")

# MQTT Server connection

print("Connecting to MQTT server... ", end="")

client = MQTTClient(MQTT\_CLIENT\_ID, MQTT\_BROKER, user=MQTT\_USER, password=MQTT\_PASSWORD)

client.connect()

print("Connected!")

# Build the message in JSON format and send the message only if there is a change

prev\_weather = ""

while True:

  print("Measuring weather conditions... ", end="")

  sensor1.measure()

  sensor2.measure()

  message = ujson.dumps({

    "temp1": sensor1.temperature(),

    "humidity1": sensor1.humidity(),

    "temp2": sensor2.temperature(),

    "humidity2": sensor2.humidity(),

  })

  if message != prev\_weather:

    print("Updated!")

    print("Reporting to MQTT topic {}: {}".format(MQTT\_TOPIC, message))

    # Send the message

    client.publish(MQTT\_TOPIC, message)

    prev\_weather = message

  else:

    print("No change")

  time.sleep(1)

MicroPython is a tiny open source Python programming language interpretor that runs on small embedded development boards. With MicroPython you can write clean and simple Python code to control hardware instead of having to use complex low-level languages like C or C++ (what Arduino uses for programming).

The simplicity of the Python programming language makes MicroPython an excellent choice for beginners who are new to programming and hardware. However MicroPython is also quite full-featured and supports most of Python's syntax so even seasoned Python veterans will find MicroPython familiar and fun to use.

Beyond its ease of use MicroPython has some unique features that set it apart from other embedded systems:

##### Schematic diagram :

###### 

This is a connection model of the simulation

###### Output :

###### 

The see this working of my project

<https://wokwi.com/projects/378753690804281345>