

lot:

INDEX:

CODI IOT NO Base de Dades	4
CODI iot SI BD	7
PHP PER LA BD:	11

Codis:

CODI IOT NO Base de Dades

Aquest codi crea una mini estació meteorològica amb un ESP8266, però sense connexió Wi-Fi ni enviament de dades. El que fa és recollir i mostrar dades pel monitor sèrie.

Què fa el codi:

1. Inicialitza sensors:
 - El DHT11 mesura temperatura i humitat (connectat al pin D6).
 - Un sensor de vent (anemòmetre) connectat al pin D5 genera impulsos quan bufa el vent.
2. Calcula la velocitat del vent:
 - Cada vegada que el sensor genera un impuls, una interrupció incrementa un comptador.

Cada 2 segons, es calcula la velocitat del vent amb la fórmula:

$\text{velocitat_kmh} = (\text{impulsos} * 2.4) / \text{segons}$

3. Mostra les dades pel monitor sèrie:
 - Si les lectures del DHT11 són vàlides, mostra:
 - Temperatura (en °C)
 - Humitat (en %)
 - Velocitat del vent (en km/h)
4. Torna a començar:
 - Es reinicia el comptador d'impulsos i es torna a esperar 2 segons.

En resum:

És un codi senzill per mesurar i mostrar dades meteorològiques localment (sense enviar-les a cap servidor). Ideal per fer proves amb sensors abans d'afegir connexions a internet o bases de dades.

```

#include "DHT.h"

#define DHTPIN D6

#define DHTTYPE DHT11

#define WINDPIN D5

DHT dht(DHTPIN, DHTTYPE);

volatile int comptador_impulsos = 0;

unsigned long temps_anterior = 0;

float velocitat_kmh = 0;

void IRAM_ATTR incrementalImpuls() {

    comptador_impulsos++;

}

void setup() {

    Serial.begin(9600);

    dht.begin();

    pinMode(WINDPIN, INPUT_PULLUP);

    attachInterrupt(digitalPinToInterrupt(WINDPIN), incrementalImpuls, FALLING);

    temps_anterior = millis();

}

void loop() {

    unsigned long temps_actual = millis();

    unsigned long interval = temps_actual - temps_anterior;

    if (interval >= 2000) {

        float h = dht.readHumidity();

        float t = dht.readTemperature();

        velocitat_kmh = (comptador_impulsos * 2.4) / (interval / 1000.0);

        if (isnan(h) || isnan(t)) {

            Serial.println("Error llegint del sensor DHT!");

        } else {

            Serial.print("Temperatura: ");

            Serial.print(t);


```

```
Serial.print(" °C | Humitat: ");  
Serial.print(h);  
Serial.print(" % | Vent: ");  
Serial.print(velocitat_kmh);  
Serial.println(" km/h");  
}  
comptador_impulsos = 0;  
temps_anterior = temps_actual;  
}  
}
```

CODI iot SI BD

Aquest codi està fet per una placa ESP8266 (com un NodeMCU) i fa una petita estació meteorològica. La seva funció és llegir dades i enviar-les a un servidor web local (XAMPP).

Què fa exactament:

1. Connecta a la Wi-Fi amb el nom i contrasenya indicats.
2. Cada 2 segons:
 - Llegeix la temperatura i humitat amb el sensor DHT11.
 - Calcula la velocitat del vent amb un sensor connectat al pin D5, comptant els impulsos amb una interrupció.
 - Mostra les dades pel monitor sèrie.
 - Si hi ha connexió, fa una petició HTTP POST a una pàgina PHP (guardar.php) que hauries de tenir en un servidor XAMPP a la IP indicada (10.93.254.11).
 - Envia les dades en format temperatura=xx&humitat=yy&vent=zz.
3. Després, reinicia el comptador d'impulsos i espera 2 segons més.

Serveix per fer un projecte de recollida de dades ambientals de manera automàtica i enviar-les a un servidor perquè quedin registrades.

EL CODI:

```
#include <ESP8266WiFi.h>
```

```
#include <ESP8266HTTPClient.h>
```

```
#include "DHT.h"
```

```
#define DHTPIN D6
```

```
#define DHTTYPE DHT11
```

```
#define WINDPIN D5
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
const char* ssid = "Cicles Aula3";      // Canvia si cal
```

```
const char* password = "Cicles@ul@3";   // Canvia si cal
```

```
const char* servidor = "10.93.254.11";   // IP del teu PC amb XAMPP
```

```
volatile int comptador_impulsos = 0;
```

```
unsigned long temps_anterior = 0;
```

```
float velocitat_kmh = 0;
```

```
void IRAM_ATTR incrementalImpuls() {
```

```
    comptador_impulsos++;
```

```
}
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    dht.begin();
```

```
    pinMode(WINDPIN, INPUT_PULLUP);
```

```
    attachInterrupt(digitalPinToInterrupt(WINDPIN), incrementalImpuls, FALLING);
```

```
    WiFi.begin(ssid, password);
```

```
    Serial.print("Connectant a Wi-Fi");
```

```
    while (WiFi.status() != WL_CONNECTED) {
```

```
        delay(500);
```

```
        Serial.print(".");
```

```
    }
```

```
    Serial.println("\nWi-Fi connectat!");
```

```
    Serial.print("IP local: ");
```

```
    Serial.println(WiFi.localIP());
```



```
    temps_anterior = millis();  
}
```

```
void loop() {  
    unsigned long temps_actual = millis();  
    unsigned long interval = temps_actual - temps_anterior;  
  
    if (interval >= 2000) {  
        float h = dht.readHumidity();  
        float t = dht.readTemperature();  
        velocitat_kmh = (comptador_impulsos * 2.4) / (interval / 1000.0);  
  
        if (isnan(h) || isnan(t)) {  
            Serial.println("Error llegint del sensor DHT!");  
        } else {  
            Serial.print("Temp: "); Serial.print(t);  
            Serial.print(" °C | Hum: "); Serial.print(h);  
            Serial.print(" % | Vent: "); Serial.print(velocitat_kmh);  
            Serial.println(" km/h");  
  
            if (WiFi.status() == WL_CONNECTED) {  
                HTTPClient http;  
                WiFiClient client;  
                String url = "http://" + String(servidor) + "/meteodades/guardar.php";  
                http.begin(client, url);  
                http.addHeader("Content-Type", "application/x-www-form-urlencoded");  
  
                String dades = "temperatura=" + String(t) + "&humitat=" + String(h) + "&vent=" + String(velocitat_kmh);  
                int httpCode = http.POST(dades);
```

```
if (httpCode > 0) {  
    Serial.print("Resposta del servidor: ");  
    Serial.println(http.getString());  
} else {  
    Serial.print("Error HTTP: ");  
    Serial.println(httpCode);  
}
```

```
    http.end();  
}  
}
```

```
comptador_impulsos = 0;  
temps_anterior = temps_actual;  
}  
}
```

PHP PER LA BD:

El fitxer PHP que has creat (guardar.php) és el que s'encarrega de rebre les dades enviades pel **Wemos** mitjançant una petició POST i de desar-les a la base de dades **estacio_meteo**. Aquí tens una explicació breu de què fa aquest codi:

```
<?php

$servername = "localhost"; // Nom del servidor (en aquest cas localhost perquè estàs treballant amb XAMPP)

$username = "root";        // Usuari de la base de dades

$password = "";            // Contrasenya de la base de dades (a XAMPP normalment és buida)

$dbname = "estacio_meteo"; // Nom de la base de dades


// Connexió a la base de dades

$conn = new mysqli($servername, $username, $password, $dbname);


// Comprovem si hi ha algun error de connexió
if ($conn->connect_error) {
    die("Error de connexió: " . $conn->connect_error); // Si hi ha error, s'atura l'execució
}


// Recollir les dades enviades pel Wemos

$temp = $_POST['temperatura']; // Dada de temperatura

$hum = $_POST['humitat'];      // Dada d'humitat

$vent = $_POST['vent'];        // Dada de velocitat del vent


// SQL per afegir les dades a la taula 'dades'

$sql = "INSERT INTO dades (temperatura, humitat, vent) VALUES ('$temp', '$hum', '$vent')";


// Executar la consulta
if ($conn->query($sql) === TRUE) {
    echo "Dades guardades"; // Si l'insert es fa bé, es mostra aquest missatge
} else {
    echo "Error: " . $sql . "<br>" . $conn->error; // Si hi ha error, mostra l'error
}
```

```
}
```

```
// Tancar la connexió a la base de dades
```

```
$conn->close();
```

```
?>
```

Explicació breu del codi:

- **Connexió a la base de dades:** El codi estableix una connexió amb la base de dades **estacio_meteo** utilitzant el **mysqli**. El nom d'usuari per defecte és **root**, i la contrasenya és buida a **XAMPP**.
- **Rebre dades del Wemos:** El codi rep les dades enviades des del Wemos (temperatura, humitat i velocitat del vent) mitjançant **\$_POST**. Aquestes dades es guarden en les variables **\$temp**, **\$hum** i **\$vent**.
- **Guardar les dades a la base de dades:** Després de rebre les dades, s'executa una consulta SQL que les inserta a la taula **dades** dins de la base de dades **estacio_meteo**. Si l'operació es fa amb èxit, es mostra el missatge "**Dades guardades**". Si hi ha algun error, es mostrarà l'error.
- **Tancar la connexió:** Finalment, es tanca la connexió amb la base de dades mitjançant **\$conn->close()**.

Ubicació del fitxer:

Aquest fitxer s'ha de col·locar dins la carpeta **htdocs** de XAMPP per poder accedir-hi des del navegador o des del microcontrolador (Wemos). Aquesta és la ruta on XAMPP busca els fitxers web per poder-los executar. La carpeta **htdocs** està situada normalment a:

```
C:\xampp\htdocs\
```

Resum:

Aquest script PHP rep les dades de temperatura, humitat i velocitat del vent enviades pel **Wemos** mitjançant una petició POST i les desa a la base de dades. Un cop guardades, mostra un missatge de confirmació o bé un missatge d'error en cas de problemes.