

ENVIROENMTEL MONITORING-DEVELOPMENT PART 1

Design a environmental monitoring system based on the esp32 using wokwi application can be a complex, but I can provide you with a high-level overview steps involved

1. Define Objectives and Requirements:

- Clearly state the purpose of your environmental monitoring system. In this case, you want to monitor temperature and humidity levels.

2. Select Sensors:

- Choose sensors capable of measuring temperature and humidity. Common choices include DHT11, DHT22, or similar sensors.

3. Select Microcontroller:

- Choose a microcontroller compatible with

4. Set Up the Wokwi Environment:

Launch the Wokwi platform and select the microcontroller you've chosen. Open a new project.

5. Connect Sensors to Microcontroller:

- Wire the temperature and humidity sensor to the microcontroller. Refer to the sensor's datasheet and ensure the correct pins are used.

6. Power Supply:

Ensure a stable power supply for

your microcontroller and sensors. This could be achieved through batteries, USB power, or other sources depending on your specific application.

7. Write the Code:

- Use the Arduino IDE or the Wokwi simulator's built-in code editor to write the code. Utilize a library compatible with your chosen sensor (e.g., DHT library for DHT sensors).

8. Code Functionality:

Write code to read data from the sensor and display it. You can choose to display the information on a virtual display in the simulator or even transmit it to a remote server if needed.

9. Test and Debug:

- Simulate the project in the Wokwi environment to ensure the sensors are providing accurate readings and the microcontroller is processing the data correctly.

10. Implement Physical Hardware:

- Once you're satisfied with the simulation, implement the same connections on real hardware using your selected microcontroller and sensors.

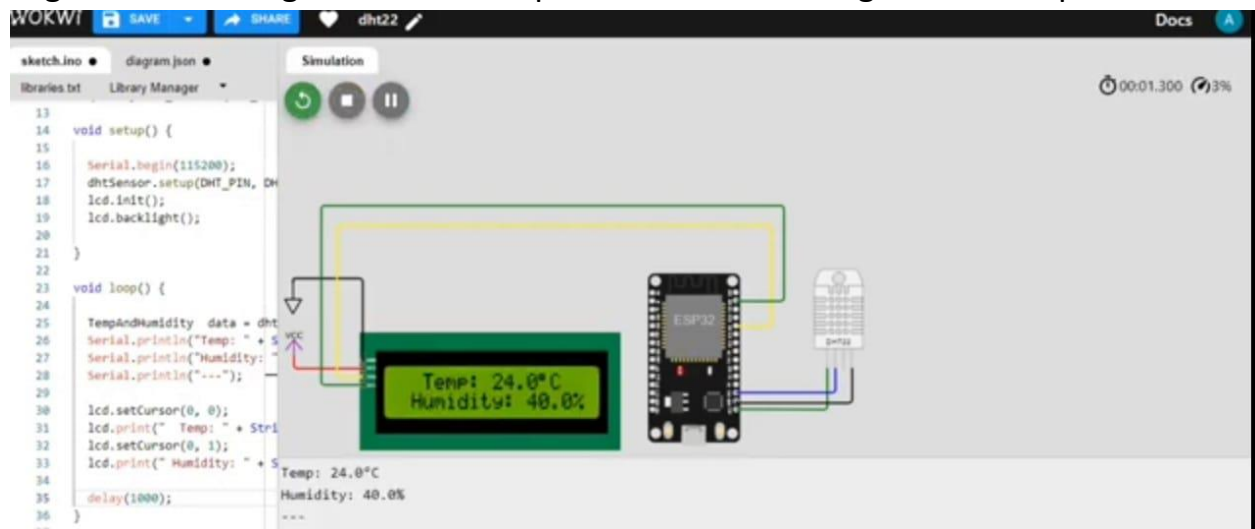
11. Monitor and Fine-tune:

- Observe the system's performance in the actual environment. Make any necessary adjustments to improve accuracy or reliability

12. Optional: Data Storage and

Visualization:

- If desired, you can add features to store and visualize the collected data. This might involve using additional components or connecting to external platforms



```
#include "DHTesp.h"
```

```
#include<liquidcrystal.12c.h>
```

```
#define 12c _ADDER 0*27
```

```
#define LCD _COLUMNS 20
```

```
#define LCD _LINES 4
```

```
Const int DHT_PIN=15;
```

```
DHTesp dhtsensor;
```

```
liquidcrystal_12c lcd(12c_ADDEr, LCD_COLUMNS,LCD_LINES);
```

```
VOID setup(){
```

```
Serial.BEGIN(115200);
```

```
dhtsensor.setup(DHT_PIN,DHTesp::DHT22);
```

```
lcd.init();
```

```
    lcd.backlight();  
}  
  
void loop(){  
    tempAndHumidity data = dhtsensor.getTempHumidity();  
    Serial.println("temp: " + string(data.temperature,1)+"c");  
    Serial.println("Humidity: " + string(data.Humidity,1)+"%");  
    Serial.println("----");  
    lcd.setCursor(0,0);  
    lcd.print("temp: " + string(data.temperature,1)+"\xDF"+"c");  
    lcd.println(0,1);  
    Serial.println("Humidity: " + string(data.Humidity,1)+"%");  
    lcd.print("wokwi online IOT");  
    delay(10000);  
}
```