

Sistema Ventilatore con ESP32



Introduzione

In questo progetto ho creato un sistema per controllare automaticamente un ventilatore in base alla temperatura ambiente. Quando fa troppo caldo, il ventilatore si accende. Inoltre, un display mostra sempre temperatura e umidità, mentre un LED ci indica visivamente lo stato del sistema.



Componenti Necessari

Hardware Principale

- ESP32
- Sensore DHT11
- Display OLED 128x64
- Ventilatore
- LED bianco
- Resistenza 220Ω

Materiali di Collegamento

- Breadboard o PCB per i collegamenti
- Cavi jumper maschio-maschio e maschio-femmina
- Alimentazione per ESP32 e ventilatore

Schema Collegamenti

Componente	Pin ESP32	Note
DHT11 (Data)	D4	Sensore temperatura/umidità
OLED (SDA)	D21	Comunicazione I2C
OLED (SCL)	D22	Comunicazione I2C
LED + Resistenza	D18	Indicatore stato
Ventilatore (+)	D15	Controllo accensione

Note sui Collegamenti

- Il DHT11 ha bisogno di alimentazione a 3.3V
- Il display OLED comunica via I2C, quindi solo 2 fili dati
- Il LED va collegato con la resistenza in serie
- Il ventilatore può richiedere un transistor o relay se consuma troppo

Librerie Necessarie

1. DHT sensor library di Adafruit
2. Adafruit GFX Library
3. Adafruit SSD1306 per il display OLED

Come Funziona il Sistema

1. Lettura continua - Ogni secondo legge temperatura e umidità
2. Controllo automatico - Se $T > 24^{\circ}\text{C}$ → ventilatore ON, altrimenti OFF
3. LED acceso quando tutto normale, spento quando ventilatore attivo
4. Display - Mostra sempre i valori

Comportamento LED

- Ventilatore SPENTO → LED acceso fisso (tutto OK, temperatura normale)
- Ventilatore ACCESO → LED spento (sistema attivo, ventilatore in funzione)

Display OLED

Mostra:

- Titolo del sistema
- Temperatura con 1 decimale
- Umidità senza decimali
- Stato ventilatore (ON/OFF)

Personalizzazioni Facili

Cambiare la Soglia di Temperatura

Modifica questa riga nel codice:

```
const float TEMP_SOGLIA = 24.0; // Cambia 24.0 con la tua temperatura che vuoi
```

Frequenza Aggiornamenti

Cambia il delay nel main loop:

```
delay(1000); // 1000ms = 1 secondo tra ogni controllo
```

Io ho scelto ogni secondo, per non sovraccaricare, è ideale perchè gli sbalzi di temperatura interni di un edificio come casa mia sono veramente lenti e non richiedono una frequenza elevata per un progetto semplice

Risoluzione Problemi che ho avuto

Display Non Si Accende

- Controlla i collegamenti SDA/SCL
- Verifica l'indirizzo I2C (dovrebbe essere 0x3C)
- Assicurati che il display sia alimentato

Sensore DHT11 Non Legge

- Controlla il pin dati (D4)
- Verifica alimentazione 3.3V o 5V esterna

Ventilatore Non Si Accende

- Controlla il collegamento al pin D15
- Verifica che l'alimentazione sia sufficiente
- Se il ventilatore consuma molto, usa un relay

NOTA :

Per questo progetto non ho avuto la necessità di un Relay, ma per progetto futuri ne avrò bisogno poichè ho intenzione di collegarci ventilatori grandi veri e propri o addirittura lampadine che richiedono un Voltaggio > 3.3 dell'esp32



Idee per Miglioramenti Futuri

- Controllo remoto - Aggiungere WiFi per controllo via smartphone
 - Più sensori
 - Impostare orari di funzionamento
 - Notifiche se temperatura troppo alta/bassa
 - Salvare temperature su SD card
 - Relay
-



Note Finali

Questo progetto è perfetto per iniziare con ESP32, è stato semplice sia dal lato di backend che il montaggio.

Lo consiglio a chi è alle prime armi