```
#define BLYNK_TEMPLATE_ID "your_template_id"
#define BLYNK_TEMPLATE_NAME "your_template_name"
#define BLYNK_AUTH_TOKEN "your_auth_token"

#include <Wire.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include "Adafruit_HTU21DF.h"

// WiFi credentials
char ssid[] = "Your_ssid";
char pass[] = "Your_pass";

// Pin definitions
#define MQ2_PIN 34
#define FAN_PIN 23

Adafruit_HTU21DF htu = Adafruit_HTU21DF();
BlynkTimer timer;

bool manualFan = false;  // Stato del controllo manuale
bool fanState = false;   // Stato effettivo della ventola

void sendSensorData() {
  float temp = htu.readTemperature();
  float hum = htu.readHumidity();
  int gasRaw = analogRead(MQ2_PIN);
  float gasPercent = map(gasRaw, 0, 4095, 0, 100); // conversione a %

  Serial.print("Temp: "); Serial.print(temp);
  Serial.print(" °C | Hum: "); Serial.print(hum);
  Serial.print(" % | Gas: "); Serial.print(gasPercent); Serial.println(" %");

  // Invia dati a Blynk
  Blynk.virtualWrite(V0, gasPercent);  // Gas
  Blynk.virtualWrite(V1, temp);        // Temp
  Blynk.virtualWrite(V2, hum);         // Umidità

  // Controllo automatico ventola
  if (temp >= 40.0 || gasPercent >= 15.0 || manualFan) {
    digitalWrite(FAN_PIN, HIGH);
    fanState = true;
  } else {
    digitalWrite(FAN_PIN, LOW);
    fanState = false;
  }

  // Stato ventola su Blynk
  Blynk.virtualWrite(V3, fanState ? 1 : 0);
}
```

```
// Riceve lo stato dello switch manuale (V4)
BLYNK_WRITE(V4) {
  manualFan = param.asInt();  // 1 = ON, 0 = OFF
}

void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22); // SDA, SCL

  if (!htu.begin()) {
    Serial.println("HTU21D non trovato. Controlla collegamenti!");
    while (1);
  }

  pinMode(FAN_PIN, OUTPUT);
  digitalWrite(FAN_PIN, LOW);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(2000L, sendSensorData); // ogni 2 secondi
}

void loop() {
  Blynk.run();
  timer.run();
}
```