

מדריך פרויקט מקיף: תחנת ניטור סביבתי ואינטרנט של הדברים (IoT) עם ESP32

1. סקירת הפרויקט ומטרותיו

פרויקט זה מציע פתרון מקצה לקצה לבניית מערכת ניטור סביבתי ובקרה חכמה לבית (Smart Home / Environment Monitoring System) המבוססת על בקר **ESP32**.

מטרת הפרויקט:

- מדידה מדויקת של טמפרטורה ולחות בחדר בזמן אמת.
- הצגת הנתונים שנמדדו באופן מקומי על גבי מסך LCD 16x2.
- שליטה ובקרה על מכשירי חשמל (או עומסים לדוגמה) באמצעות מודול ממסר (Relay) כפול.
- אפשרות למעבר מצבים ידני/אוטומטי בעזרת לחצן רגעי.
- חיבור רשת תקשורת אלחוטית (Wi-Fi) לשידור נתונים לשרת ענן או ממשק ניטור מרחוק.

2. רשימת רכיבי הערכה ומפרט טכני

רכיב	תיאור ומפרט טכני	תפקיד במערכת
ESP32 Development Board	מיקרו-בקר עוצמתי בעל מעבד כפול ליבה, תקשורת Wi-Fi Bluetooth-ו 802.11 b/g/n משולבים, פני GPIO מרובים.	מוח המערכת – מעבד נתונים, מפעיל את המסך והממסר, ומנהל תקשורת רשת.
חיישן טמפרטורה ולחות (DHT22 (AM2302	חיישן דיגיטלי בעל דיוק גבוה. טווח מדידת טמפרטורה: -40° עד 80°C (דיוק $\pm 0.5^{\circ}\text{C}$), טווח לחות: 0-100% (דיוק $\pm 5\%$).	דגימת הנתונים הסביבתיים.
מסך תצוגה LCD 16x2 עם	מסך אלפא-נומרי להצגת 2 שורות	הצגת נתוני הסטטוס,

רכיב	תיאור ומפרט טכני	תפקיד במערכת
מתאם I2C	של 16 תווים, כולל מתאם I2C של PCF8574)) המאפשר חיבור ב-2 קווי תקשורת בלבד (SDA, SCL).	הטמפרטורה, הלחות ומצב הממסרים ברמה המקומית.
מודול ממסר כפול (Channel-2) (Relay Module)	מודול ממסר המאפשר מיתוג מתח גבוה (עד 250VAC / 10A). בעל מגננון הטיית/אופטוקאפלר להפרדה גלוונית ובטיחות.	הפעלה וניתוח של עומסי חשמל (כגון מאוורר, תאורה, או חימום).
מודול כפתור רגעי (Keyestudio Button)	כפתור לחיצה הכולל נגד משוך (Pull-down/Pull-up) מובנה על הלוח לתפוקה נקייה ללא רעשים.	שינוי מצבי עבודה (כגון מיתוג ידני של הממסרים או החלפת מסכי תצוגה).
ערכת חוטי גישור Dupont 20cm	חוטים גמישים מסוג זכר-נקבה, נקבה-נקבה, וזכר-זכר לחיבור סדיר בין הרכיבים ללוח הפיתוח.	חיווט והעברת אותות ומתחים בין רכיבי המערכת.

3. תרשים חיבורים ומפת פינים (Wiring Diagram)

יש לבצע את החיווט לפי הפינים המפורטים בטבלה הבאה. שים לב לחיבורי המתח (5V / 3.3V) וההארקה (GND).

רכיב	פין ברכיב	חיבור בלוח ESP32	הערות
חיישן DHT22	+ / VCC	3.3V (או 5V)	מתח הפעלה לחיישן
	- / GND	GND	הארקה משותפת
	OUT / DATA / S	GPIO 4	קו נתונים דיגיטלי

רכיב	פין ברכיב	חיבור בלוח ESP32	הערות
מסך LCD I2C	GND	GND	הארקה משותפת
	VCC	(5V (VIN	אספקת מתח לתאורה האחורית והמסך
	SDA	GPIO 21	קו נתונים I2C תקני ב-ESP32
	SCL	GPIO 22	קו שעון I2C תקני ב-ESP32
מודול ממסר (Relay)	VCC	(5V (VIN	מתח הפעלה לסלילי הממסר
	GND	GND	הארקה משותפת
	IN1	GPIO 18	פין פיקוד ערוץ 1 (מאורר/קירור)
	IN2	GPIO 19	פין פיקוד ערוץ 2 (תאורה/חימום)
מודול כפתור	V / VCC	3.3V	מתח הפעלה לכפתור
	G / GND	GND	הארקה משותפת
	S / OUT	GPIO 13	אות לחיצה דיגיטלי

דגש בטיחות: בעת עבודה עם מתח רשת גבוה (220V) דרך הממסר, יש לנקוט בכל אמצעי הזהירות הדרושים. מומלץ להתחיל בניסויים עם עומסי מתח נמוך (כגון נורות LED או מאורר DC קטן) לפני חיבור

4. הגדרת סביבת העבודה (Software Setup)

1. הורד והתקן את תוכנת **Arduino IDE** (גרסה 2.0 ומעלה).
2. הוסף את תמיכת ESP32 ל-Arduino IDE:
 - כנס ל-File -> Preferences.
 - בשדה *Additional Boards Manager URLs* הוסף את הקישור:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_e_sp32_index.json
 - פתח את Boards Manager -> Board -> Tools, חפש **esp32** והתקן את החבילה.
3. התקנת ספריות נדרשות דרך ה-Library Manager (Tools -> Manage Libraries):
 - **DHT sensor library** מאת Adafruit (כולל Adafruit Unified Sensor).
 - **LiquidCrystal_I2C** מאת Frank de Brabander.

5. קוד המקור המלא (Arduino / ESP32 Code)

הקוד הבא קורא את נתוני הטמפרטורה והלחות מחיישן ה-DHT22, מציג אותם על גבי מסך ה-LCD, מנהל בקרת אקלים אוטומטית (הפעלת מאוורר בטמפרטורה גבוהה), ומאפשר שליטה ידנית בערוץ הממסר השני באמצעות הכפתור.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"

// הגדרת פנים
#define DHTPIN 4           // פין חיבור חיישן DHT22
#define DHTTYPE DHT22     // סוג החיישן

// (פין ממסר ערוץ 1) בקרת טמפרטורה / מאוורר
#define RELAY1_PIN 18
// (פין ממסר ערוץ 2) תאורה / שליטה ידנית
#define RELAY2_PIN 19
// פין כפתור רגעי
#define BUTTON_PIN 13

// הגדרת אובייקטים
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); // 0x27: כתובת I2C, 16: רוחב, 2: גובה

// משתנים גלובליים
bool relay2State = false;
```

```

int lastButtonState = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

unsigned long lastSensorRead = 0;
const long sensorInterval = 2000; // קריאה כל 2 שניות

void setup() {
    Serial.begin(115200);

    // הגדרת פניי ממסרים
    pinMode(RELAY1_PIN, OUTPUT);
    pinMode(RELAY2_PIN, OUTPUT);

    // LOW Active-הגדרה ראשונית של ממסרים כבויים (רוב מודולי הממסר פועלים ב
    digitalWrite(RELAY1_PIN, HIGH);
    digitalWrite(RELAY2_PIN, HIGH);

    // הגדרת פין כפתור
    pinMode(BUTTON_PIN, INPUT);

    // אתחול החיישן והמסך
    dht.begin();
    lcd.init();
    lcd.backlight();

    // הודעת פתיחה במסך
    lcd.setCursor(0, 0);
    lcd.print("Smart Home System");
    lcd.setCursor(0, 1);
    lcd.print("Initializing...");
    delay(2000);
    lcd.clear();
}

void loop() {
    unsigned long currentMillis = millis();

    // --- קריאת נתוני חיישן כל 2 שניות ---

```

```

    if (currentMillis - lastSensorRead >= sensorInterval) {
        lastSensorRead = currentMillis;

        float humidity = dht.readHumidity();
        float temperature = dht.readTemperature();

        if (isnan(humidity) || isnan(temperature)) {
            Serial.println("Failed to read from DHT sensor!");
            lcd.setCursor(0, 0);
            lcd.print("Sensor Error!  ");
            return;
        }

        // הדפסה לסריאלי
        Serial.print("Temp: ");
        Serial.print(temperature);
        Serial.print(" C | Humidity: ");
        Serial.print(humidity);
        Serial.println(" %");

        // LCD-הצגה על גבי מסך ה
        lcd.setCursor(0, 0);
        lcd.print("Temp: ");
        lcd.print(temperature, 1);
        lcd.print((char)223); // סימן מעלות
        lcd.print("C  ");

        lcd.setCursor(0, 1);
        lcd.print("Hum:  ");
        lcd.print(humidity, 1);
        lcd.print("%  ");

        // --- --- (ערוץ ממסר 1) לוגיקת בקרת אקלים אוטומטית
        // אם הטמפרטורה עולה על 28 מעלות - הפעל קירור/מאוורר
        if (temperature > 28.0) {
            digitalWrite(RELAY1_PIN, LOW); // הפעלת ממסר 1
        } else {
            digitalWrite(RELAY1_PIN, HIGH); // כיבוי ממסר 1
        }
    }

```

```

{
  --- ( ) --- // שלטה ידנית באמצעות כפתור (ערוץ ממסר 2
  int reading = digitalRead(BUTTON_PIN);

  if (reading != lastButtonState) {
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    static int buttonState = LOW;
    if (reading != buttonState) {
      buttonState = reading;
      if (buttonState == HIGH) {
        relay2State = !relay2State;
        digitalWrite(RELAY2_PIN, relay2State ? LOW : HIGH);
        Serial.print("Relay 2 toggled: ");
        Serial.println(relay2State ? "ON" : "OFF");
      }
    }
  }
  lastButtonState = reading;
}

```

6. שלבי בדיקה, הפעלה ודיבאג

1. **בדיקת חיבורים פיזיים:** לפני חיבור כבל ה-USB, ודא כי קווי המתח וההארקה אינם מקוצרים.
2. **טעינת הקוד:** חבר את לוח ה-ESP32 למחשב באמצעות כבל Micro-USB, בחר בפתרון הנכון תחת Tools -> Board -> ESP32 Dev Module ובחר בפורט המתאים (COM Port). לחץ על Upload.
3. **כיוון ניגודיות המסך:** אם הטקסט במסך ה-LCD אינו מופיע בבירור, השתמש במברג קטן כדי לסובב את הפוטנציומטר הכחול הנמצא בגב מתאם ה-I2C של המסך עד לקבלת תמונה חדה.
4. **סריקת כתובת I2C (במידת הצורך):** אם המסך נדלק אך לא מציג טקסט, הריץ סקריפט I2C Scanner כדי לוודא שכתובת המסך היא אכן 0x27.
5. **בדיקת עבודה:** נשוף בעדינות על חיישן ה-DHT22 כדי להעלות את החום והלחות – ודא כי הנתונים במסך מתעדכנים, וכי ערוץ הממסר מגיב בהתאם לסף שנקבע. לחץ על הכפתור הצהוב לבדיקת מיתוג ערוץ הממסר השני.