

Időjárás állomás automata öntöző rendszerrel

Projekt kivitelezői:

Laczó Bálint

Projekt megvalósításának helyszíne:

Vas Vármegyei SZC III. Béla Technikum.
9970 Szentgotthárd. Honvéd utca 10.

Projekt ismertetése:

Időjárás állomás mely méri a páratartalmat és a hőmérsékletet ezzel párhuzamosan egy talajnedvességmérővel mérjük a föld víztartalmát és ha a minimális szint alá csökken egy relével kapcsoljuk a locsolóberendezést. A beérkező adatokat, soros monitoron és/vagy egy LCD kijelzőn követhetjük nyomon. Adatok megtekintésére használhatjuk még a Blynk szoftvert, amin keresztül kapcsoljuk a relét.

Komponensek hiányában felhasznált eszközök: ESP WROOM-32-ESP32S és a hozzá tartozó tápkábel, talajnedvesség-érzékelő, 16x2 LCD panel, 2 csatornás relé modul (5V), DHT-22 pára- és hőmérséklet-szenzor, breadboard, jumper kábel

Hiányzó alkatrészek: elektromos mágnesszelep víz-levegő szelep kapcsoló, kapcsolóüzemű DC-DC konverter

Felhasznált szoftverek:

Arduino IDE, Blynk



ESP WROOM-32-ESP32S

Mikrovezérlő, a projekt agya,
ezen keresztül fut a program



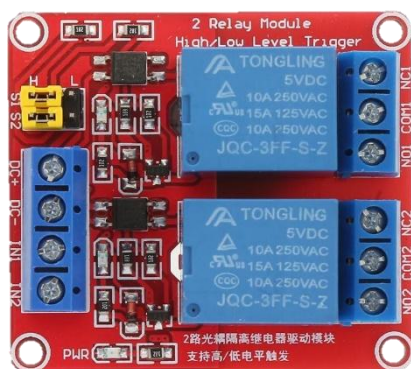
Talajnedvesség-érzékelő

Talajnedvességnek mérésére
szolgáló eszköz



16x2 LCD panel

A kapott talajnedvesség és hőmérséklet kijelzése



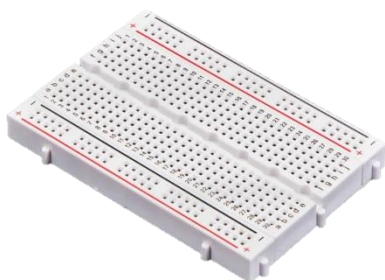
2 csatornás relé modul (5V)

Mágnesszelep kapcsolásáért felelős relé (alkatrész hiánya miatt a szelep nem került beépítésre)



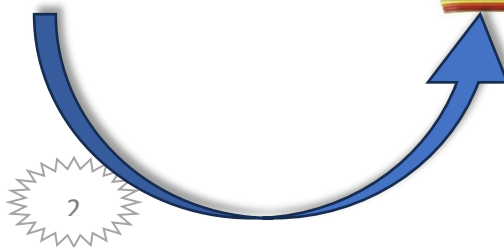
DHT-22 szenzor

Hőmérséklet és páratartalom mérése



Breadboard

Eszközök ráhelyezése, majd jumper kábelvel való összekötése



Programkód másolható formában

```
#define BLYNK_TEMPLATE_ID "TMPL4M0zVFNsh"
#define BLYNK_TEMPLATE_NAME "ontozo"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
```

```
#define crit_min_hom 10
#define min_hom 15
#define atlag_hom 20
#define max_hom 25
#define crit_max_hom 30
```

```
#define crit_min_nedv
#define min_nedv
#define atlag_nedv
#define max_nedv
#define crit_max_nedv
```

```
#define sensor 33
#define relay 4
#define DHTPIN 5
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```

```
LiquidCrystal_I2C lcd(0x27, 22, 21);
```

```
BlynkTimer timer;
```

```
char auth[] = "mssqNTr_VMygeyALfuLeixMqi6SifJMi";
char ssid[] = "TM4_Wifi";
char pass[] = "Server_01";
```

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.init();
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  lcd.backlight();
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);
```

```
  lcd.setCursor(1, 0);
  lcd.print("System Loading");
  for (int a = 0; a <= 15; a++) {
```

Könyvtárak
meghívása

Definiálás

Blynk
beállítása

LCD-re való
kiíratás

```

    lcd.setCursor(a, 1);
    lcd.print(".");
    delay(200);
}
lcd.clear();

}

void soilMoisture() {
    int value = analogRead(sensor);
    value = map(value, 0, 4095, 0, 100);
    value = (value - 100) * -1;
    Blynk.virtualWrite(V0, value);
    Serial.println(value);
    lcd.setCursor(0, 0);
    lcd.print("Talajnedv.: ");
    lcd.print(value);
    lcd.print("%");
}

void temperature(){
    int value = dht.readTemperature();
    lcd.setCursor(0, 1);
    lcd.print("Homerseklet: ");
    lcd.print(value);
    lcd.print("C");
}

BLYNK_WRITE(V1) {
    bool Relay = param.asInt();
    if (Relay == 1) {
        digitalWrite(relay, LOW);
        lcd.setCursor(0, 1);
        lcd.clear();
        lcd.print("Motor is ON ");
        delay(2000);
    } else {
        digitalWrite(relay, HIGH);
        lcd.setCursor(0, 1);
        lcd.clear();
        lcd.print("Motor is OFF");
        delay(2000);
    }
}

void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();

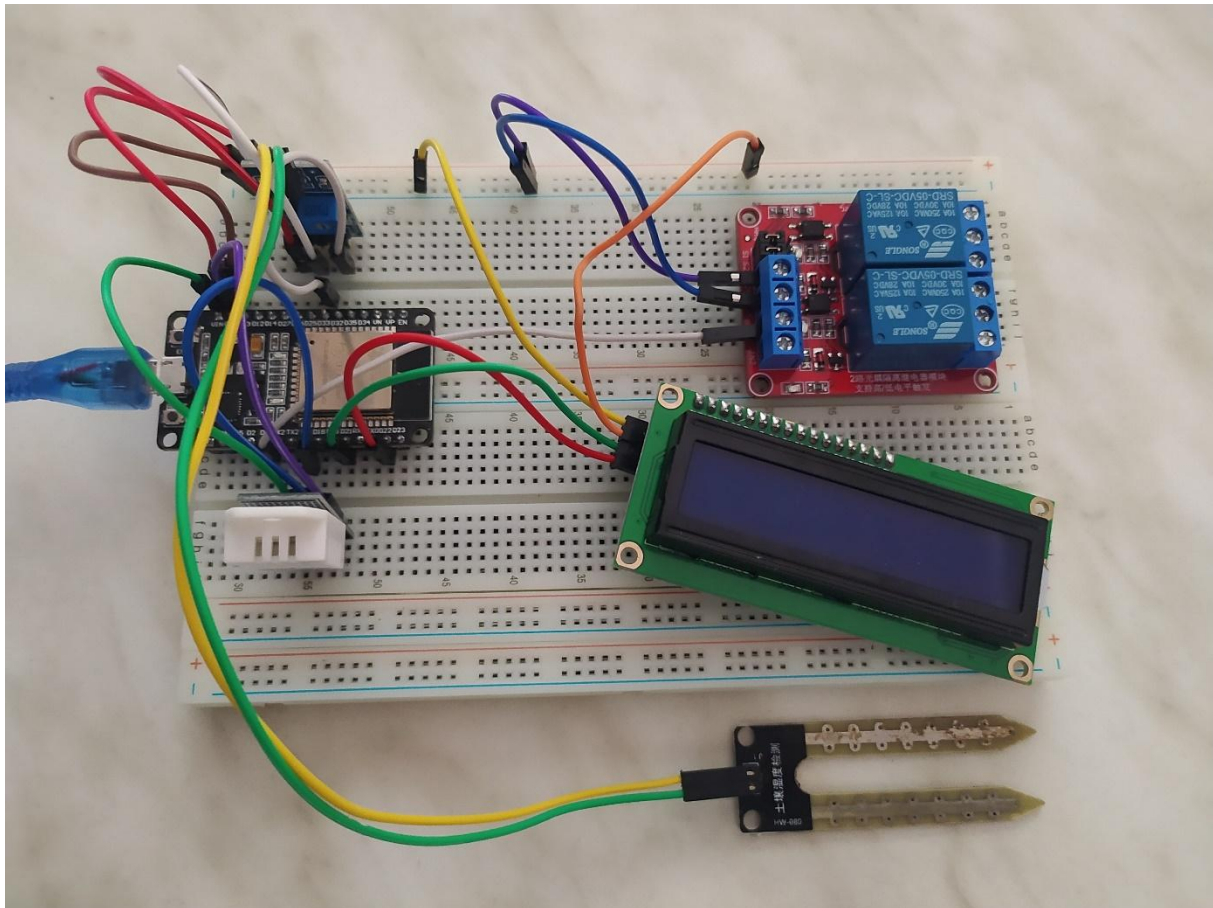
    Blynk.virtualWrite(V2, h);

```

LCD-re való
kiíratás

LCD-re való
kiíratás

```
Blynk.virtualWrite(V3, t);  
soilMoisture();  
temperature();  
Blynk.run();  
lcd.write(t);  
  
delay(200);  
  
}
```



Projekt jelenlegi állapota

A projekthez inspirációként egy YouTube videót használtunk.