

Okosvirág

Készítők:

Juszkú Dorina

Nagy Réka Zsuzsa

Lévai Dominika

A projektben a programozás nagyobb részét Réka vállalta, az összeszerelést Dorina a kivitelezés pedig Dominika feladata volt.

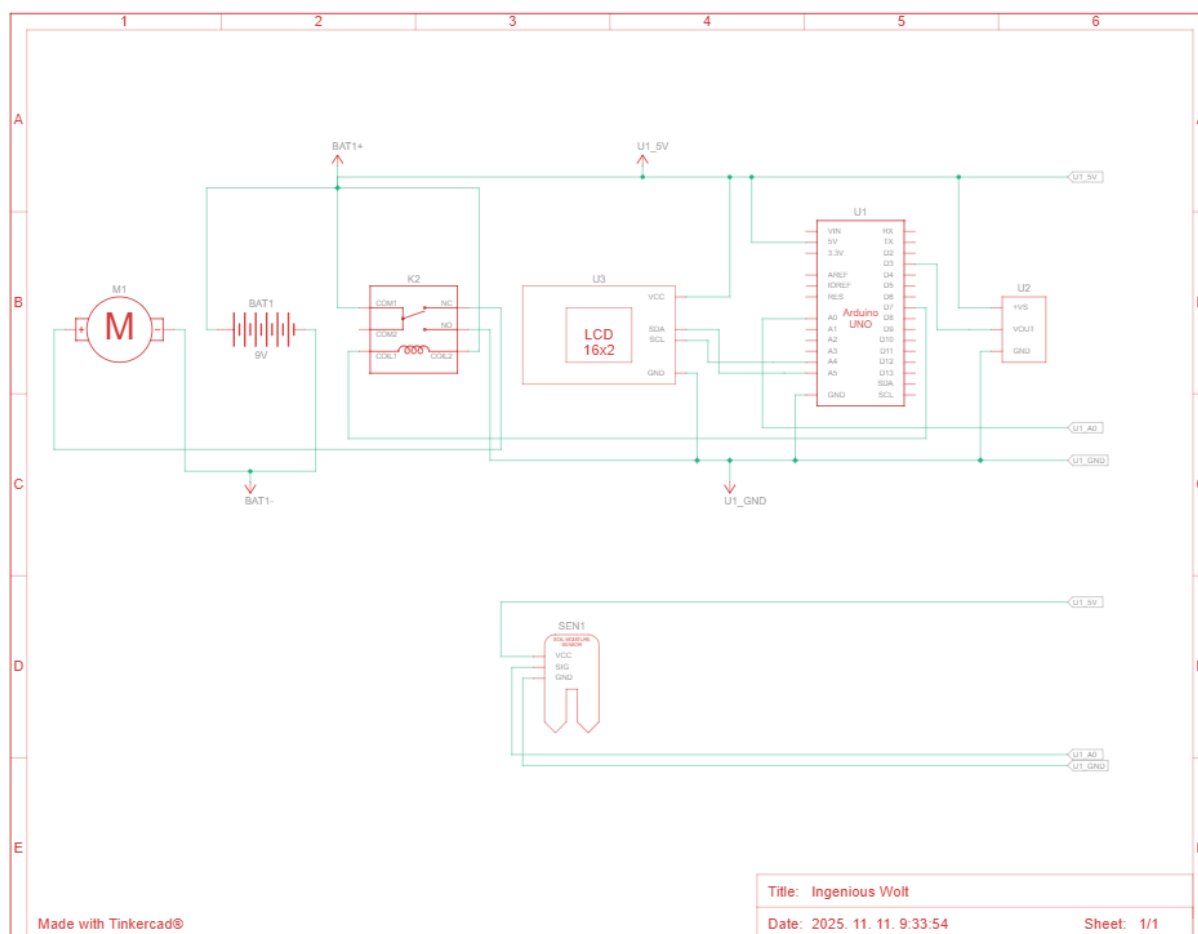
A projekt rövid leírása

A projektünk célja egy növénymegfigyelő rendszer kialakítása volt. A rendszer tudatja a felhasználóval, hogy milyen a talajnedvességszázalék a páratartalom és a hőmérséklet. A rendszerbe beépítettünk egy öntözőt is, amely működését a talajnedvességszázalék dönti el.

Hardverelemek, elemek feladata

Tervünk megvalósításához egy Arduino Uno R3 mikrovezérlőt használtunk, amely a rendszer központi irányítóegységeként biztosítja az egyes komponensek közötti kommunikációt és a folyamatok vezérlését. Felhasználtuk még a talajnedvességmérőt, ami meghatározza, hogy megfelelően elláttuk-e a növényünket, illetve egy DHT11 modult, ami a hőmérsékletet és a páratartalmat méri. Az adatokat egy I2C LCD kijelzőn jelenítettük meg. Az öntözőrendszerhez szükségünk volt egy relay modulra és egy 5V-os mini merülőszivattyúra, ami biztosítja a növény vízellátását.

Áramköri rajz



Bom lista

Név	Mennyiség	Megnevezés
U1	1	Arduino Uno R3
BAT1	1	Powerbank (5V)
M1	1	5 V-os merülőszivattyú
U3	1	I2C LCD kijelzőn
SEN1	1	Talajnedvesség-érzékelő
U2	1	Hőmérséklet-érzékelő (DHT11)
K2	1	Relé (SRD-05VDC-SL-C)

A forráskód és magyarázata

```
#include <dht11.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// eszkoz
const int soilPin = A0;
const int relayPin = 7;
#define DHT11PIN 4

// kijelzo
dht11 DHT11;
LiquidCrystal_I2C lcd(0x27, 16, 2);

// beallitasok
const int dryThreshold = 20;
const unsigned long pumpTime = 1000UL;

// allapot
bool pumpOn = false;
unsigned long pumpStart = 0;

void setup() {
  Serial.begin(9600);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);

  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("Okosvirag ON");
  lcd.setCursor(0,1); lcd.print("Udvozol a virag!");
  delay(1500);
  lcd.clear();
}

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

  //meres
  int sensorValue = analogRead(soilPin);
  int moisturePercent = map(sensorValue, 1023, 0, 0, 100);

  int chk = DHT11.read(DHT11PIN);
  float humidity = (float)DHT11.humidity;
  float temperature = (float)DHT11.temperature;

  Serial.print("Soil:"); Serial.print(moisturePercent); Serial.print("% ");
  Serial.print("T:"); Serial.print(temperature); Serial.print("C ");
  Serial.print("H:"); Serial.print(humidity); Serial.println("%");

  // pumpa kezdes off
  if (!pumpOn && moisturePercent < dryThreshold) {
    // pumpa on
    digitalWrite(relayPin, HIGH);
    pumpOn = true;
    pumpStart = now;
    Serial.println("Pumpa ON");
  }

  if (pumpOn) {
    if (now - pumpStart >= pumpTime) {
      digitalWrite(relayPin, LOW);
      pumpOn = false;
      Serial.println("Pumpa OFF");
      // kijelzo frissites
      delay(50);
    } else {
    }
  } else {
    //kijelzo ha pumpa off
    lcd.setCursor(0,0);
    lcd.print("T:");
    lcd.print(temperature,1);
    lcd.print((char)223);
    lcd.print("C ");

    lcd.setCursor(10,0);
    lcd.print("H:");
    lcd.print(humidity,0);
    lcd.print("% ");

    lcd.setCursor(0,1);
    lcd.print("Soil:");
    lcd.print(moisturePercent);
    lcd.print("% ");

    lcd.setCursor(11,1);
    lcd.print(" ");
  }

  delay(2000); // kesleltetes a meresek kozott
}
```

A program elején a szükséges könyvtárak vannak megadva: a dht11.h a hőmérséklet- és páratartalom-érzékelő kezeléséhez kell, a Wire.h és a LiquidCrystal_I2C.h pedig az LCD kijelző működéséhez. Ezután a kódban beállításra kerültek a csatlakozási pontok: a talajnedvesség-mérő szenzor az A0 analóg bemenetre, a relé a 7-es digitális kimenetre, a DHT11 szenzor pedig a 3-es pinre csatlakozik.

Létrehozzuk az eszközök vezérléséhez szükséges objektumokat: egy DHT11 nevű példányt a szenzorhoz, és egy lcd nevű objektumot a kijelzőhöz. A beállítások között meg van adva, hogy mi számít száraz talajnak (`dryThreshold = 20`), vagyis ha a nedvesség 20% alá esik, akkor a program be kapcsolja a pumpát. A pumpa működési ideje 1000 ezredmásodperc, vagyis 1 másodperc.

A program elején két változó tárolja a pumpa állapotát: a `pumpOn` jelzi, hogy éppen megy-e a pumpa, a `pumpStart` pedig elmenti, mikor indult el.

A `setup()` rész csak egyszer fut le, amikor az Arduino bekapcsol. Itt beállítja a kommunikációt a számítógéppel (`Serial.begin(9600)`), a relét kimenetként konfigurálja, és kezdetben kikapcsolt állapotban hagyja (LOW). Ezután elindítja az LCD kijelzőt és egy rövid üdvözlő üzenetet ír ki („Okosvirag ON”, „Udvozol a virag!”). Másfél másodperc után törli a kijelzőt, hogy készen álljon az adatok megjelenítésére.

A `loop()` rész folyamatosan ismétlődik, itt történik a mérés és az automatikus öntözés vezérlése. Először a `millis()` függvénnyel elmenti az aktuális időt milliszekundumban. Ezzel tudja majd mérni, mennyi ideje megy a pumpa. Ezután beolvassa a talajnedvesség-érzékelő jelét az A0 lábról, és az értéket százalékos formába alakítja a `map()` függvénnyel. A legtöbb talajnedvesség-szenzor 0–1023 közötti analóg értéket ad vissza, ahol a 1023 általában a száraz, a 0 pedig a vizes állapotot jelenti.

Ezután a program kiolvassa a DHT11 szenzor adatait, és eltárolja a hőmérsékletet és páratartalmat lebegőpontos számként. Ezeket az értékeket a soros monitorra is kiírja, így a felhasználó a számítógépen is láthatja, hogy milyen az aktuális talajnedvesség, hőmérséklet és páratartalom.

A következő rész dönt arról, hogy be kell-e kapcsolni a pumpát. Ha a pumpa jelenleg nem működik és a talajnedvesség kisebb, mint a beállított küszöb (20%), akkor a program bekapcsolja a relét, ezzel elindítja a pumpát. Ekkor a `pumpOn` változó `true` lesz, és elmenti az indulás időpontját. A soros monitorra kiírja, hogy „Pumpa ON”.

A kód ezután folyamatosan figyeli, hogy mennyi idő telt el. Ha a pumpa már működik, és az indulás óta eltelt idő nagyobb vagy egyenlő, mint az engedélyezett 1 másodperc, akkor kikapcsolja a relét (LOW), és ezzel a pumpa is leáll. A soros monitorra kiírja, hogy „Pumpa OFF”.

Ha a pumpa nem működik, a kijelző frissül: az első sorban megjelenik a hőmérséklet fokban, valamint a páratartalom százalékban. A második sorban látható a talaj nedvessége százalékos formában. A program néhány szóközzel biztosítja, hogy a régi szöveg ne maradjon a kijelzőn.

A program végén egy delay(2000) parancs található, ami azt jelenti, hogy 2 másodpercet vár a következő mérés előtt. Ez lassítja a frissítést, hogy a kijelző és a soros monitor olvasható maradjon, és a szenzorokat se terhelje túl gyors lekérdezés.