

Projekt R

Sustav za automatsko navodnjavanje vrta pomoću

Opis projekta

Cilj ovoga projekta bio je simuliranje automatskog navodnjavanja vrta na temelju parametara dobivenih od nekolicine senzora. Senzori su spojeni na jedan mikrokontroler koji u određenim intervalima šalje informacije drugom mikrokontroleru putem ESP-NOW protokola. Na temelju dobivenih informacija drugi mikrokontroler simulira otvaranje ili zatvaranje solenoidnog ventila putem LED diode. Logiku može određivati i sam korisnik putem web sučelja tako da postavi sustav na ručno upravljanje i sam otvori odnosno zatvori ventil. Komunikacija između drugog mikrokontrolera i korisnika odvija se preko HTTP protokola gdje je mikrokontroler spojen na Internet i služi kao web poslužitelj.

Dijelovi sustava

Sustav se sastoji od 2 ESP32-32S mikrokontrolera od kojih jedan služi za skupljanje podataka senzora, a drugi za procesiranje podataka i prikazivanje istih korisniku preko web sučelja. Senzori korišteni u sustavu su:

- BMP280 – temperatura zraka
- HW-390 – vлага tla
- KY-018 – razina svjetlosti

Mikrokontroler 1

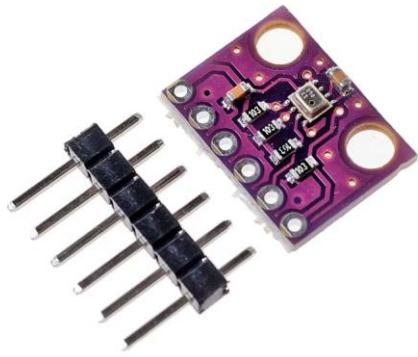
Funkcionalnost prvog mikrokontrolera može se podijeli na dva dijela, prikupljanje podataka sa senzora i slanje podataka.

Prikupljanje podataka sa senzora

BMP280

BMP280 senzor je koji služi za mjerjenje temperature i tlaka zraka ali se ovdje koristi samo kao senzor za temperaturu. Ima 6 pinova ali budući da se komunikacija vrši putem I²C protokola koristimo ih samo 4. GND spojen na ground, VCC spojen na 3.3V, SCL spojen na

Serial Clock pin (D22) i SDA spojen na Serial Data pin (D21). Za komunikaciju sa senzorom korištena je biblioteka Adafruit BMP280.



Slika 1: BMP280

HW-390

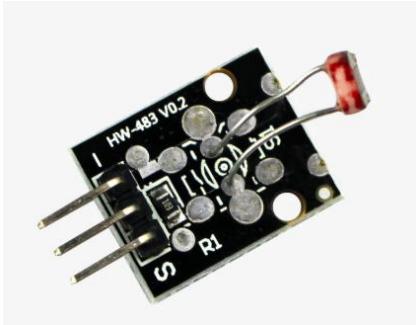
HW-390 je kapacitivni senzor za mjerenje postotka vlage u tlu. Ima 3 pina, GND, VCC (3.3V) i AOUT, informaciju o vlagi tla šalje u obliku napona preko AOUT pina koji je spojen na D34 pin mikrokontrolera s ugrađenim ADC-om (Analog to digital konverter). Analogno digitalni konverter pretvara analogni signal u digitalni koji potom možemo očitati u kodu i koristiti. Kako bi vrijednosti koje senzor daje bile ispravne potrebno ga je kalibrirati tako da se izmjeri signal koji daje na suhome i kada se uroni u slanu vodu (sol povećava vodljivost i daje točniji rezultat). Nakon što smo izmjerili maksimum i minimum sve vrijednosti mapiramo između njih, od 0% (suho) do 100% (vlažno).



Slika 2: HW-390

KY-018

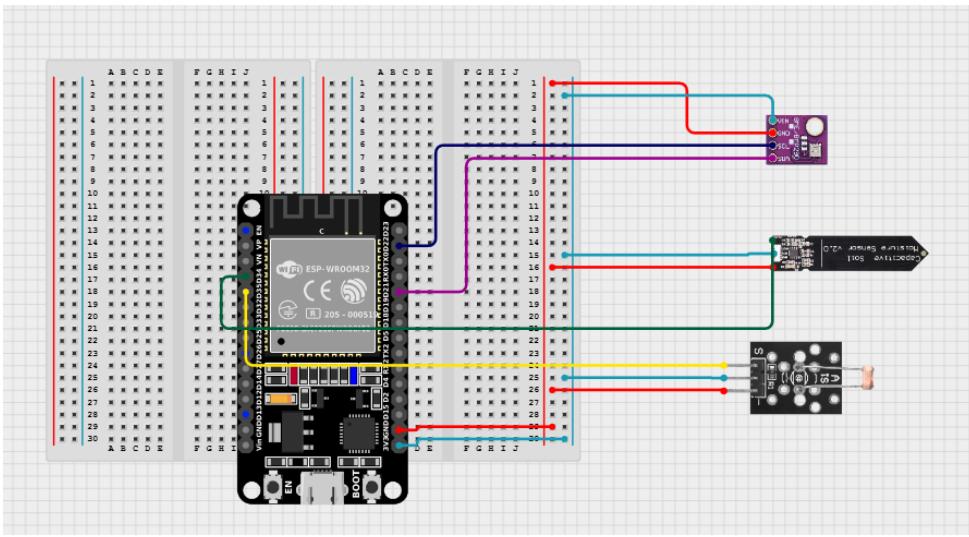
KY-018 je senzor s fotootpornikom za mjerenje intenziteta svjetla. Isto kao i HW-390 ima GND, VCC i AOUT pinove. Njegov AOUT pin je spojen na pin D35 mikrokontrolera koji također ima ugrađen ADC.



Slika 3: KY-018

Komunikacija s drugim mikrokontrolerom

Za komunikaciju se koristi ESP-NOW protokol. To je “peer-to-peer” protokol s niskom razinom kašnjena i malom potrošnjom energije koji ne ovisi o Wi-Fi-ju. Za inicijalizaciju nam je potrebna MAC adresa mikrokontrolera kojemu šaljemo podatke, a nakon slanja podataka poziva se callback funkcija koja nam govori je li paket stigao na odredište. Paket se šalje svakih 20 sekundi.



Slika 4: Shema spajanja senzora na mikrokontroler 1

```
void loop() {
    int raw_light_val = analogRead(LIGHT_SENSOR_PIN);
    float light_intensity = (float) (4095 - raw_light_val) / 4095;

    int raw_soil_moisture_val = analogRead(SOIL_MOISTURE_SENSOR_PIN);
    int clamped = constrain(raw_soil_moisture_val, WET_VALUE, DRY_VALUE);
    float soil_moisture = (float)(DRY_VALUE - clamped) / (DRY_VALUE - WET_VALUE);

    my_data.light_intensity = light_intensity;
    my_data.soil_moisture = soil_moisture;
    my_data.air_temp = bmp.readTemperature();

    // slanje poruke
    esp_err_t result = esp_now_send(broadcast_address, (uint8_t *) &my_data,
sizeof(my_data));

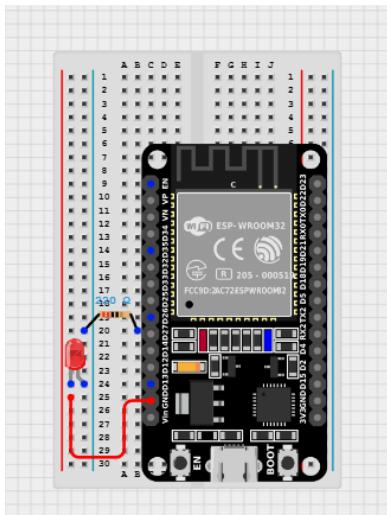
    if (result == ESP_OK) {
        Serial.println("Sent with success");
    }
    else {
        Serial.println("Error sending the data");
    }

    delay(20000);
}
```

Isječak koda 1: Glavna petlja prvog mikrokontrolera

Mikrokontroler 2

Svrha drugog mikrokontrolera je obrada podataka sa senzora te simulacija otvaranja ili zatvaranja ventila. Umjesto ventila i pravog izvora vode ovdje je korištena svijetleća dioda (LED). Uz to mikrokontroler služi i kao web server na kojem se mogu pregledavati podatci sa senzora i ručno upravljati ventilom.



Slika 5: Shema spajanja LED diode na mikrokontroler 2

Obrada podataka

Temperatura zraka, vлага tla i razina svjetlosti globalne su varijable koje se ažuriraju svaki put kada dođe paket s novim očitanjima senzora. Nije dovoljno da se samo zaljeva kada vлага tla bude niska već se mora pričekati da temperatura nije prevelika i svjetlost prejaka kako se biljke ne bi osušile nakon zalijevanja. Kada sve tri vrijednosti dosegnu optimalnu razinu i način rada sustava je postavljen na automatski ventil se otvara (o ručnom i automatskom načinu rada bit će objašnjeno u dijelu za web sučelje).

```
void OnDataRecv(const uint8_t * mac, const uint8_t *incoming_data, int len) {

    struct_message tmp;
    memcpy(&tmp, incoming_data, sizeof(tmp));

    //azuriramo globalne vrijednosti
    my_data = tmp;
    data_available = true;
    bool should_open;

    if (manual_mode) {
        //ako je manualni mod onda otvaramo ili ne prema tome sta je odabрано na webu
        should_open = valve_state;
    } else {
        should_open =
            my_data.soil_moisture < 0.3 &&
            my_data.light_intensity < 0.6 &&
            my_data.air_temp < 29;
    }
    valve_state = should_open;
    digitalWrite(VALVE_PIN, should_open ? HIGH : LOW);
}

}
```

Isječak koda 2: Callback funkcija za prijem ESP-NOW paketa

Spajanje na lokalnu mrežu

Kako bi se moglo pristupiti web sučelju mikrokontroler mora biti spojen na istu lokalnu mrežu kao i korisnik koji pristupa stranici.

WiFiManager

Za spajanje na mrežu korištena je biblioteka WiFiManager preko koje mikrokontroler pokrene vlastitu WiFi mrežu (Access Point) na koju se korisnik može spojiti te konfigurirati SSID i Password za spajanje na lokalnu mrežu. Ako su SSID i Password ispravni mikrokontroler se sam spoji na mrežu.

mDNS

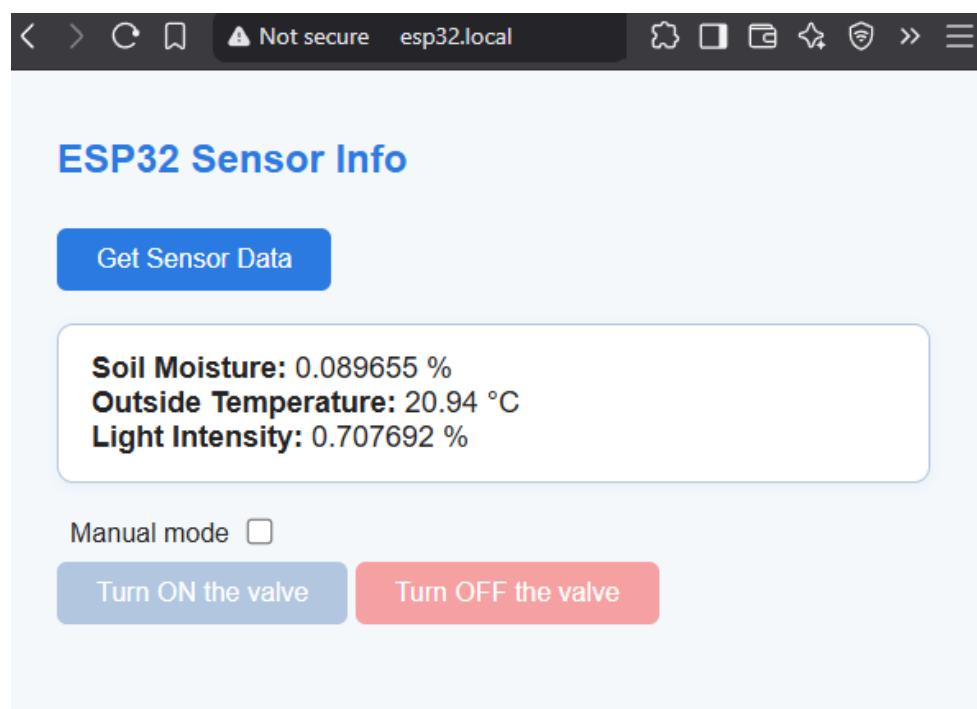
Kako korisnik ne bi trebao znati točan IP mikrokontrolera koristi se mDNS (Multicast DNS) koji omogućuje uređaju na lokalnoj mreži da ima ime umjesto IP adrese. Tako će se mikrokontroleru umjesto preko <http://192.168.1.101> pristupati preko <http://esp32.local>. Još jedan razlog zašto je ovakav pristup dobar je taj što je moguće da uređaj nakon svakog spajanja na WiFi dobije novu IP adresu (zbog DHCP-a) što nije problem ako koristimo ime umjesto adresu.

Web server

Nakon što se mikrokontroler uspješno spoji na lokalnu mrežu pokreće web server. Za implementaciju servera korištena je biblioteka ESPAsyncWebServer. Ona nudi asinkron rad servera. To znači da mikrokontroler može istovremeno posluživati web i raditi svoje zadatke u glavnoj petlji, ali budući da se obrada događaja odvija putem asinkronih povratnih funkcija, u ovom slučaju glavna petlja se ne koristi.

Web sučelje

Za pregled podataka i ručno upravljanje ventilom napravljeno je jednostavno web sučelje.



Slika 6: Web sučelje

LittleFS

U svrhu bolje organizacije podataka korištena je LittleFS biblioteka. LittleFS je datotečni sustav koji omogućuje spremanje datoteka u flash memoriju. Ovdje je korišten za spremanje HTML i CSS datoteka.

Dohvaćanje podataka

Web sučelje dobiva podatke sa senzora preko gumba „Get Sensor Data“ koji na server šalje HTTP GET zahtjev. Podaci očitani sa senzora strukturirani su u JSON objekt pomoću biblioteke ArduinoJson te poslani klijentskoj aplikaciji unutar tijela HTTP odgovora.

```
server.on("/api/status", HTTP_GET, [](AsyncWebRequest *request){  
    StaticJsonDocument<200> doc;  
  
    if (!data_available) {  
        doc["soil"] = nullptr;  
        doc["light"] = nullptr;  
        doc["temp"] = nullptr;  
    } else {  
        doc["soil"] = my_data.soil_moisture;  
        doc["light"] = my_data.light_intensity;  
        doc["temp"] = my_data.air_temp;  
    }  
  
    String out;  
    serializeJson(doc, out);  
    request->send(200, "application/json", out);  
});
```

Isječak koda 3: API endpoint za dohvata senzorskih podataka

Prikaz podataka

Prikaz podataka ostvaren je pomoću jednostavne HTML stranice, pri čemu je logika dohvaćanja i obrade podataka implementirana u JavaScriptu unutar <script> označke, dok je stilizacija stranice izdvojena u posebnu CSS datoteku.

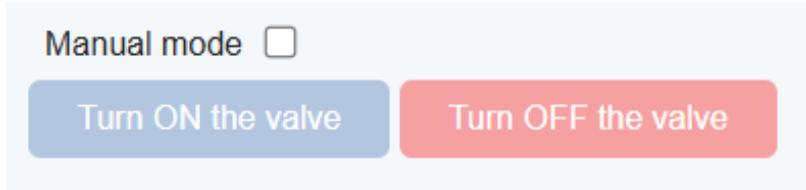
Načini upravljanja ventilom

Automatsko upravljanje

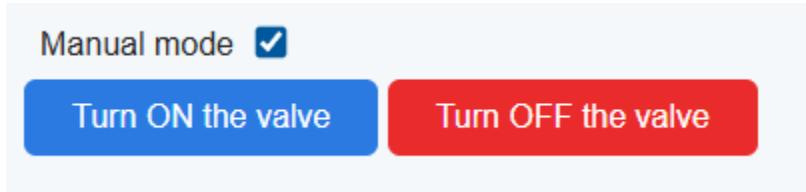
Ovo je zadani način upravljanja. Ventil se sam otvara i zatvara ovisno samo o podatcima koje dobiva od senzora.

Ručno upravljanje

Korisnik također može odabratи opciju ručnog upravljanja. U tom slučaju ventil se može otvoriti ili zatvoriti neovisno o vrijednostima koje pokazuju senzori. Da bi se ušlo u način ručnog upravljanja potrebno je označiti potvrđni okvir „Manual mode“.



Slika 6.1: Ručno upravljanje isključeno



Slika 6.2: Ručno upravljanje uključeno

```
server.on("/api/ventil/on", HTTP_POST, [](AsyncWebRequest *request){  
    if (!manual_mode) {  
        request->send(403, "text/plain", "Not in manual mode");  
        return;  
    }  
    valve_state = true;  
    digitalWrite(VALVE_PIN, HIGH);  
    request->send(200, "text/plain", "Valve ON");  
});
```

Isječak koda 4: API endpoint za ručno otvaranje ventila

Testiranje

Sustav je testiran simulacijom stvarnih uvjeta rada. Razina svjetlosti mijenjana je promjenom osvjetljenja, vlažnost tla simulirana je uranjanjem senzora u vodu, dok je promjena temperature ostvarena lokalnim zagrijavanjem senzora.

Zaključak

Projektom je uspješno realiziran sustav za simulaciju automatskog navodnjavanja temeljen na ESP32 mikrokontrolerima. Sustav omogućuje prikupljanje senzorskih podataka, njihovu obradu te upravljanje ventilom u automatskom i ručnom načinu rada putem web sučelja. Testiranjem su potvrđene ispravnost i funkcionalnost rješenja. Sustav se u budućnosti može dodatno proširiti ugradnjom stvarnog solenoidnog ventila, dodavanjem mogućnosti udaljenog pristupa izvan lokalne mreže te dodavanjem mogućnost konfiguriranja uvjeta rada. Time bi korisnik mogao prilagoditi pragove vlažnosti tla, temperature i razine osvjetljenja prema vlastitim potrebama.

Izvorni kod

U dokumentu su prikazani samo ključni dijelovi izvornog koda, dok je cijelokupan kod dostupan na sljedećoj poveznici: <https://github.com/LukaZadro/ESP32-automatic-irrigation>