

Plant Info Collector

- Nume: Croicia Constantin Rares Alexandru
- Grupa: 333CC

Introducere

- Plant Info Collector este un proiect care extrage date despre o planta si trimite date la o aplicatie web care mai departe se ocupa cu manipularea acestora.
- Scopul proiectului este acela de a imbina utilul cu placutul, invatand mai bine cum functioneaza un astfel de sistem.
- Ideea de baza a fost incercarea de a gasi ceva util, chiar daca nu e atat de complex.
- Toti avem plante in casa, uneori ne intereseaza sa stim daca trebuie udate/mutate din cauza luminii, asa ca putem lasa sistemul acesta pornit sa trimita date catre un server si sa le putem folosi cand dorim.

Descriere generală

Sistemul este amplasat langa planta, iar senzorii sunt pusi astfel incat sa extraga date despre planta si ambientul acesteia. Sistemul extrage date o data la un timp ajustabil, default fiind 1 minut, insa poate sa trimita date si la orice moment doreste utilizatorul. Exista 3 butoane si un led/afisaj LCD(Subject to be decided) instalate pe sistem, 2 dintre butoane sunt folosite pentru a ajusta intervalul la care se trimit date, iar al treilea buton este folosit pentru a trimite datele in momentul apasarii. Led-ul clipoceste de fiecare data cand sunt trimise date / LCD-ul va arata cate secunde mai sunt pana la urmatoarea trimitere a datelor. Sistemul se conecteaza Wireless/Serial prin USB(Subject to be decided) la o aplicatie web care primeste datele extrase de senzori si acestea pot fi folosite oricum. In cazul de fata, acestea vor fi afisate intr-un tabel cu optiunea de a putea fi descarcate ca CSV. Sistemul va merge pe baterie.



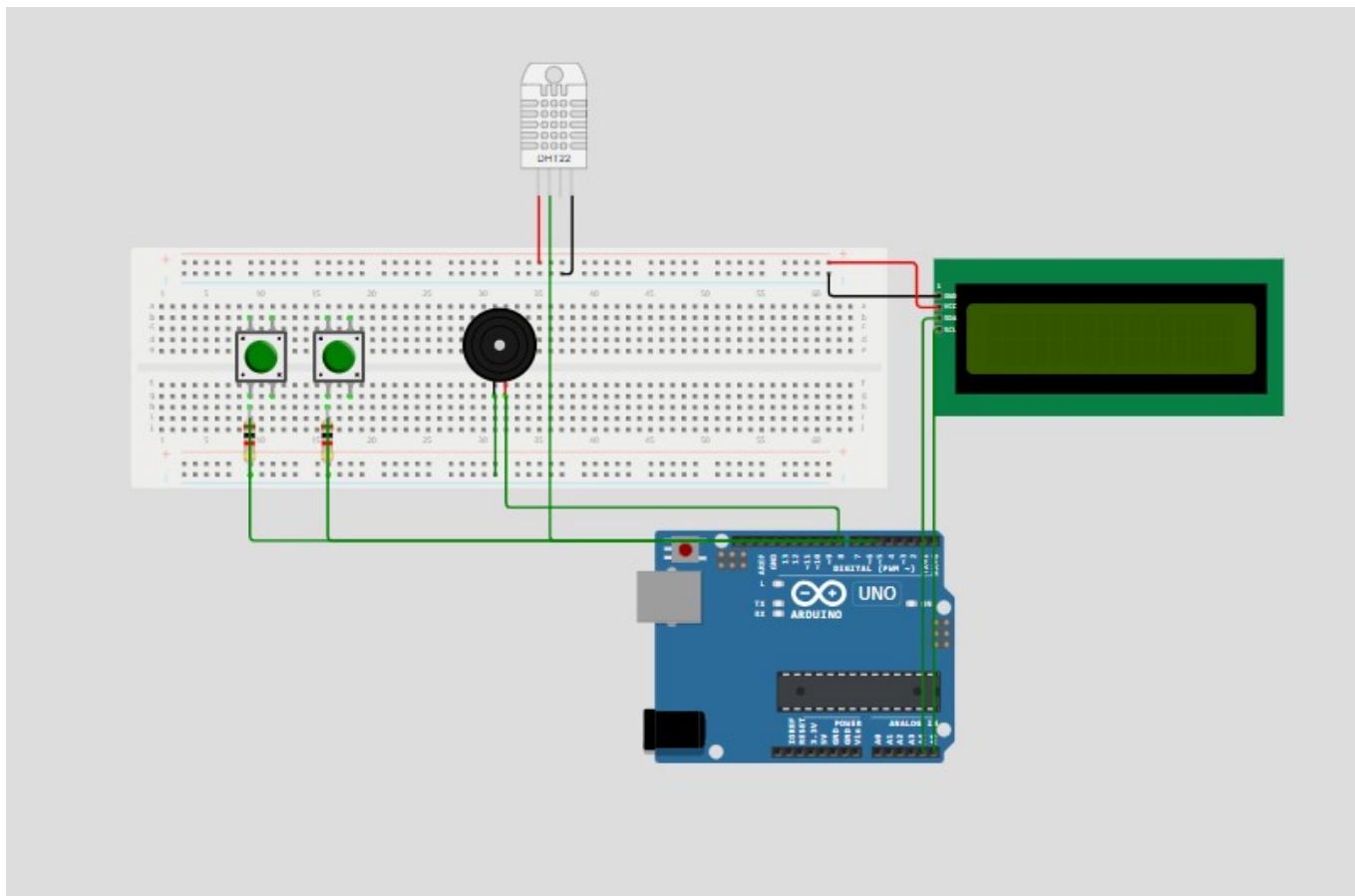
Hardware Design

Lista Piese:

- Arduino UNO R3
- Senzori Temperatura + Umiditate (ala de luminozitate mi-a venit stricat cu 3 zile inainte de PM Fair...)
- Butoane
- Afisaj LCD

- Buzzer
- Rezistente 1K
- Fire (Tata-Tata, Mama-Tata)

Schema proiectului:



Software Design

Mediu de dezvoltare folosit

- **Arduino IDE:** Dezvoltarea codului pentru arduino
- **VS Code:** Dezvoltarea codului pentru aplicatia Web
- **dht11.h:** Am folosit aceasta [biblioteca](#) pentru ca avea deja implementata citirea datelor de pe senzor
- **LiquidCrystal_I2C.h:** Am folosit biblioteca existenta in library manager-ul de la Arduino pentru a folosi implementarea oficiala pentru output pe LCD.

Detalii de implementare

- Pentru LCD am folosit un modul I2C pentru a fi mai usor de comunicat pe placuta.
- Exista 2 intreruperi legate la cele doua butoane, cand un buton este apasat, intervalul la care se trimit date scade sau creste cu 10 secunde.
- Buzzer-ul scoate sunet la fiecare apasare de buton sau cand se trimit date.
- Exista un timer bazat pe functia millis() care la fiecare secunda updateaza variabila de timp si

- verifica daca trebuie sa trimita date
- Datele se trimit serial prin cablul USB, existand un mic protocol prin care se trimit datele.
 - Datele sunt incapsulate astfel:
 - Sirul Begin care spune ca incepe mesajul
 - un numar care reprezinta temperatura in grade Celsius
 - Good/Bad in functie de temperatura
 - un numar care reprezinta nivelul de umiditate din zona
 - Good/Bad in functie de umiditate
 - Sirul End care spune ca s-a terminat mesajul.
 - Mai departe este treaba aplicatiei din celalalt end sa gestioneze parsarea acestor date si folosirea lor.
 - Senzorul citeste date o data la cat este intervalul. Initial acesta e 5 secunde pentru a fi usor la debug, intr-un mediu de productie ar varia acest numar.

Laboaratoare folosite

- Intreruperi - Pentru butoane
- SPI - Pentru trimiterea datelor Serial
- I2C - LCD
- Timere - Timerul variabil

Rezultate Obținute


Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Cam [asa](#) arata! Uite si un [video](#)!

Download

Proiectul se poate descarca apasand [aici](#). Poate fi vizualizat pe [repo](#).

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2024/aungureanu/constantin.croicia>



Last update: **2024/05/26 12:52**