

Pet Feeder

Introducere

Ai fost vreodata întrerupt de animalul de companie fiindcă voia să fie hrănit? Sau a trebuit să lipsești mai mult de acasă însă nu ai putut pentru că trebuia să îi oferi de mâncare animaluțului tău? Dacă răspunsul este pozitiv, nu ești singurul. Și eu mă aflu în această situație. Din acest motiv, m-am gândit ca pentru acest proiect să încerc să construiesc un automat ce îi oferă animaluțului porția necesară de mâncare la ore prestabilite. Astfel, tot ce mai rămâne de făcut este să setezi orele, să umpli recipientul de unde va cădea mâncarea și totul este pregătit. În plus, dispozitivul va trimite o notificare pe e-mail dacă masa a fost servită.

Acest proiect își propune să automatizeze puțin viața unui posesor de animale de companie prin eliminarea grijilor în ceea ce privește hrănirea lor.

Descriere generală

Utilizatorul va seta ora la care trebuie să fie hrănit animalul cu ajutorul tastaturii și al ecranului LCD. Când este timpul ca mâncarea să fie eliberată, utilizatorul o să primească pe e-mail o notificare trimisă cu ajutorul modului WiFi de pe ESP32.

Servomotorul este utilizat pentru a închide și a deschide recipientul unde se află depozitată mâncarea. Modulul RTC este folosit pentru a ține cont de timp.



Hardware Design

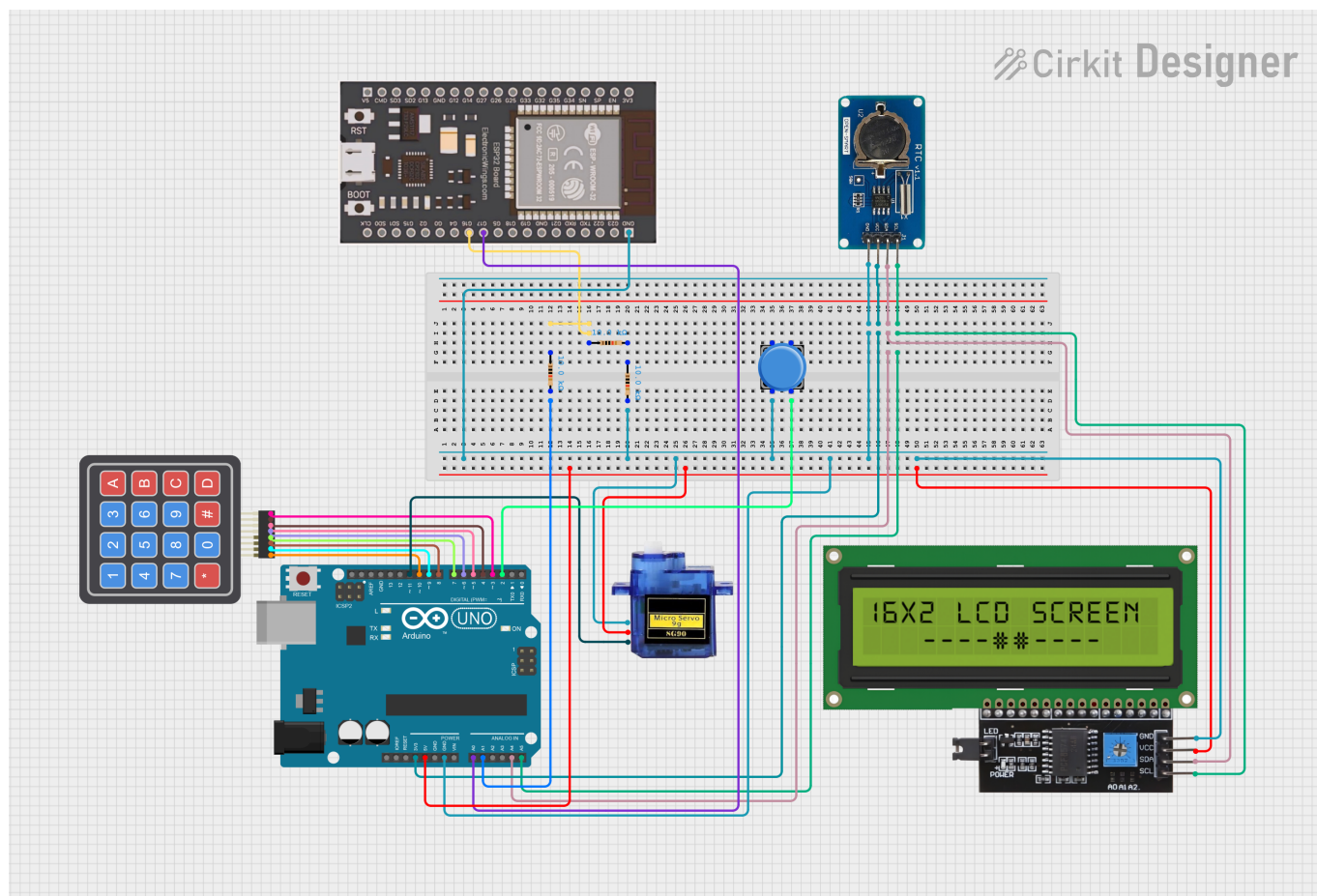
Lista de piese:

- Arduino Uno R3
- Servomotor SG90
- Modul LCD 1602
- Tastatura matriceala 4x4
- Placa de dezvoltare ESP32 cu WiFi și Bluetooth
- Modul RTC DS3231
- Breadboard
- Buton
- Fire pentru conexiuni

Componente folosite:

- Buton - folosit pentru o intrerupere, acesta fiind folosit cand utilizatorul vrea sa selecteze numarul de mese si orele
- Tastatura 4x4 - input mai simplu pentru ore/nr mese
- Ecran LCD - vizualizarea celor selectate
- Servomotor - deschide containerul cu mancare
- Modul RTC - ofera ora exacta, chiar daca a fost deconectat
- ESP32 - trimitere notificari prin email dupa ce animalul a fost hranit
- Rezistente (3 de 10kohmi) - divizor de tensiune pentru comunicarea arduino - esp32

Schema electrica:



Am realizat un divizor de tensiune cu 3 rezistente de 10kohmi pentru legatura UART dintre Arduino si ESP32, fiindca ESP32 functioneaza la 3.3V, iar Arduino la 5V. Este nevoie doar la ESP32 (RX) - Arduino (TX).

Deocamdata am niste probleme la alimentarea de la o baterie de 9V fiindca am observat ca ESP32 nu reuseste sa se conecteze la WiFi, inasa de la laptop nu exista probleme.

Update - o sa folosesc un power bank.

Pini folositi Arduino UNO:

- PD2 este folosit pentru conexiunea cu un buton, folosit pentru o intrerupere cand utilizatorul va selecta numarul de mese si timpul
- PD3-PD7 si PB0-PB2 sunt folositi pentru conexiunea cu keypad-ul 4x4 (GPIO)
- PB3 este folosit pentru functionarea servomotorului (PWM)
- PC4 (SDA) este folosit pentru conexiunile I2C de date cu LCD-ul si modulul RTC
- PC5 (SCL) este folosit pentru conexiunea I2C de clock cu LCD-ul si modulul RTC
- PC0 este folosit ca RX pentru UART pentru comunicarea cu ESP32
- PC1 este folosit ca TX pentru UART pentru comunicarea cu ESP32
- GND pentru conectarea la ground
- 3v3 este folosit pentru alimentarea modulului RTC cu 3.3V

Pini folositi ESP32:

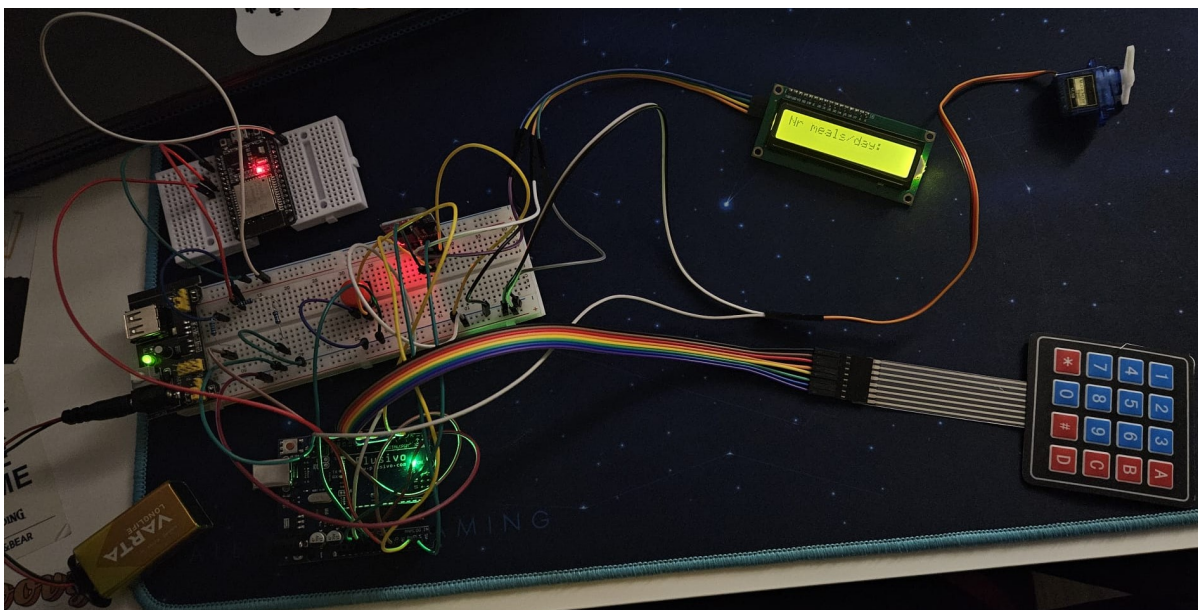
- G16 folosit ca RX pentru comunicarea UART cu Arduino
- G17 folosit ca TX pentru comunicarea UART cu Arduino
- GND pentru conectarea la ground

Functionalitate hardware:

[Functionalitate Servo cu modul RTC + LCD](#)

[Functionalitate keypad cu afisare pe ecran LCD](#)

Stadiul actual:



Software Design

Mediul de dezvoltare: PlatformIO

Bibliotecile folosite in cadrul proiectului:

Pentru Arduino Uno R3:

- Arduino.h
- DIYables_Keypad.h - pentru utilizarea keypad-ului 4x4
- Wire.h - pentru comunicatia prin I2C cu LCD-ul si modulul RTC DS3231
- LCD_I2C.h - pentru utilizarea LCD-ului 1602
- Servo.h - pentru utilizarea servo-ului
- DS3231.h - pentru utilizarea modulul RTC DS3231
- SoftwareSerial.h - pentru a comunica prin UART cu ESP32
- avr/interrupt.h - pentru utilizarea intreruperilor

Pentru ESP32:

- Arduino.h
- WiFi.h - pentru a realiza conexiunea la WiFi cu esp32
- ESP_Mail_Client.h - pentru a trimite notificari prin e-mail cand se elibereaza mancarea
- HardwareSerial.h - pentru a comunica prin UART cu Arduino

Elementul de noutate:

- Cand este timpul unei mese, iar feeder-ul elibereaza mancarea, Arduino ca trimite un mesaj catre ESP32 pentru a trimite pe e-mail o notificare catre proprietar.
- Exista un buton care este special folosit pentru setup (butonul rosu).

Justificare functionalitati din laboratoare:

- **GPIO** - Pentru a folosi keypad-ul 4x4 pentru a selecta timpul si butonul pentru setup.
- **UART** - Pentru comunicarea intre Arduino si ESP32, dar l-am folosit si pentru debug pe parcursul implementarii.
- **Intreruperi** - Pentru butonul rosu. Se va realiza o intrerupere cand utilizatorul vrea sa faca setup-ul feeder-ului, urmand sa se poata folosi si tastatura.
- **PWM** - Pentru servomotor, folosit cu ajutorul bibliotecii Servo.h.
- **I2C** - Pentru modulul RTC, cat si pentru ecranul LCD.

Descrierea codului aplicatiei:

Pentru Arduino:

1. In functia setup() sunt pregatite componentele.
2. In functia loop(), daca programul doar ce a fost pornit, setup_done o sa fie false pana cand utilizatorul face setupul.
3. Cand se realizeaza intreruperea, se apeleaza functia get_nr_meals() unde se va selecta numarul de mese.
4. Se apeleaza functia set_times() care va astepta ca userul sa scrie timpii doriti si ii va adauga in vectorul meals.
5. Functia sort_times() va sorta timpii dati de user, pentru a afla urmatorul timp la care trebuie servita masa. Acest lucru se va realiza convertind ora la minute si adaugand minutul, aflandu-se minutul din zi corespunzator.
6. Pe parcurs se verifica daca timpul curent este acelasi cu timpul la care trebuie eliberata mancarea. Pe ecran se afiseaza constant cat timp a ramas, in functie de ore si/sau minute. Daca este, se misca servo-ul, se schimba indexul din vectorul meals pentru urmatoarea masa si se trimite mesaj catre ESP32.

Pentru ESP32:

1. Sunt declarate datele necesare (id si parola WiFi, mail-ul sender-ului si cel al receiver-ului).
2. In setup() se va realiza conectarea la WiFi.
3. In loop() ESP32 asteapta pana cand primeste mesaj de la Arduino, iar apoi ii trimite un mesaj de "ACK" si trimite mail catre owner.

Demo video:

[Demo pet feeder](#)

Am setat primul timp de masa la 20:09. Mail-ul l-am primit la 20:10 (timpul setat pe modulul RTC este putin inapoi fata de laptop). Am taiat putin din video pana am primit mail-ul.

Optimizari:

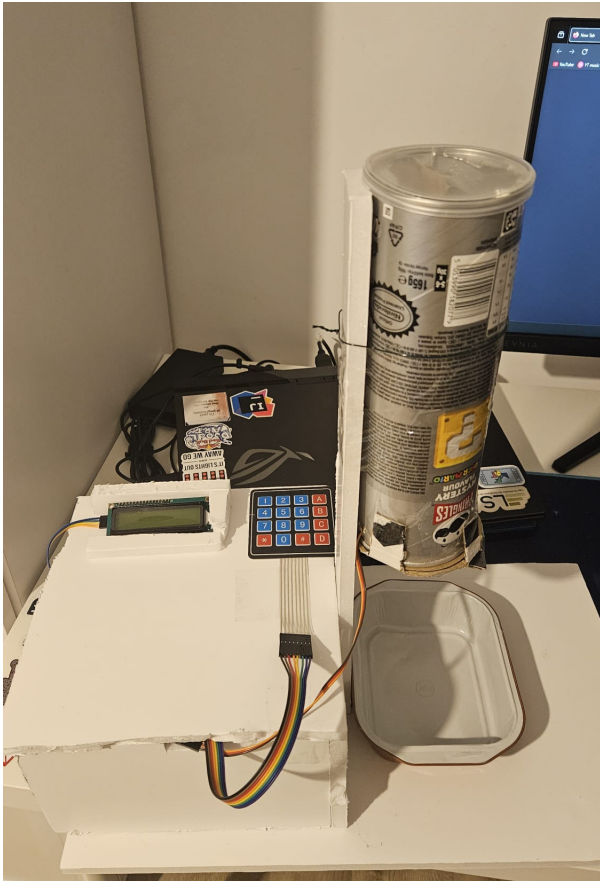
Am implementat o intrerupere la apasarea butonului rosu, moment in care se poate folosi keypad-ul pentru selectarea planului de mese. Astfel, nu se va testa la fiecare loop daca a fost apasata o tasta de pe keypad.

Algoritmi si structuri folosite:

Pentru stocarea timpilor la care trebuie sa se elibereze mancarea, am folosit un array de structuri. Structura de baza este alcatuita din 2 int-uri ce vor stoca ora si minutul. Timpii introdusi vor fi sortati imediat dupa introducerea tuturor pentru a fi mai usor de stiut care este urmatoarea masa si a nu fi nevoie sa cauti la fiecare pas care e cel mai apropiat timp.

Rezultate Obținute

Am reusit sa imi construiesc propriul pet feeder care elibereaza mancarea la ore fixe, alese de tine. Ca utilizator poti seta numarul de mese/zi (daca gresesti, poti sterge si rescrie), apoi orele si gata. Cand mancarea este eliberata, Arduino trimite mesaj catre ESP32 care trimite notificare catre utilizator pe mail. Pana cand este eliberata mancarea, pe LCD este afisat timpul ramas.



Concluzii

Proiectul a fost interesant, m-a ajutat sa inteleg mai bine conceptele studiate la PM si sa vad cum pot face debug pe hardware :). In plus, mi-a placut faptul ca am putut sa lucrez si la hardware, lucru pe care nu il mai facusem inainte.

[Sper sa le placa:\)](#)

Download

[GitHub Repo](#)

Arhiva proiect: [proiect_final.zip](#)

In pet_feeder este codul pentru arduino, in esp_part pentru esp + README.

Jurnal

- 10 Mai - am inceput sa conectez componentele
- 16 Mai - am terminat hardware-ul
- 20 Mai - am inceput sa scriu software-ul
- 29 Mai - ultimele modificari

Bibliografie/Resurse

Resurse Hardware

- [Divizor de tensiune pentru ESP32 si Arduino](#)
- [Documentatie esp32](#)
- [Documentatie Arduino Uno R3](#)

Resurse Software

- [Trimitere e-mail de pe ESP32](#)
- [Datasheet ATmega328P](#)

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/daria_maria.tanasie



Last update: **2025/05/29 21:25**