Dispenser de mâncare pentru animale

Autor

Mihăilă Iuliana-Raluca

Introducere

Proiectul are ca obiectiv principal ușurarea modului de hrănire a animalelor de companie.

Acest produs se adresează oricărui deținător de pisici sau câini și face parte din ramura produselor destinate pentru segmentul de “Smart-Home”.

Descriere generală

Problema reală apare atunci când un posesor de pisici sau câini trebuie sa părăsească locuința pentru un timp mai îndelungat, iar animalul trebuie să aibă mereu condițiile minime necesare pentru a trăi, printre acestea aflându-se și hrana. Soluția pe care o propun este implementarea unui dispenser de mancare ușor de folosit si simplist, construit din materiale reciclabile: două conserve cilindrice și un recipient din plastic unde va ajunge mâncarea.

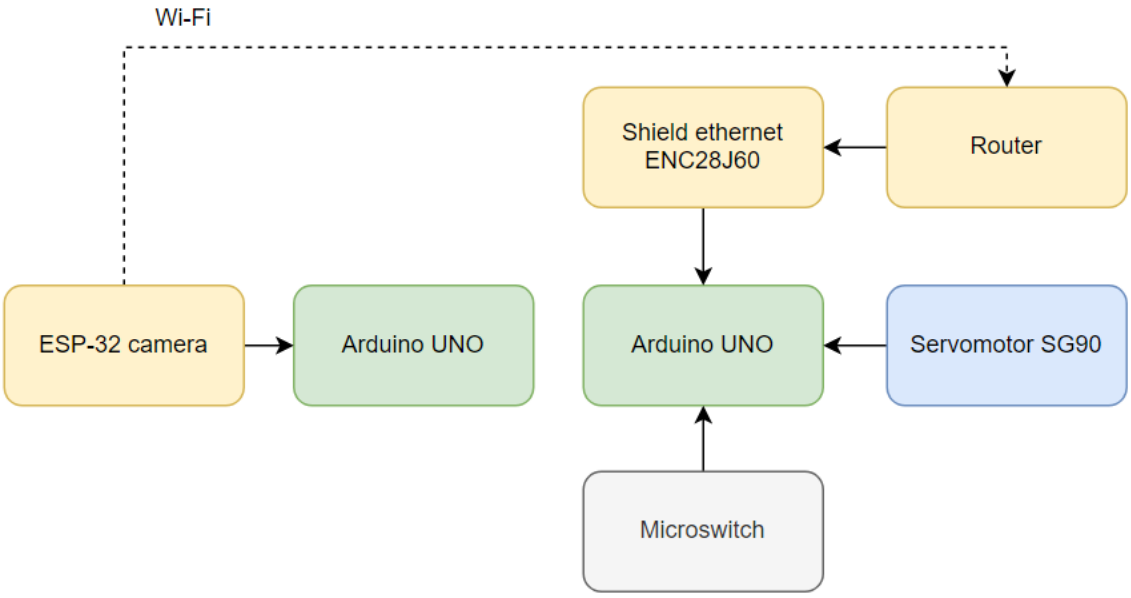
Cele trei metode prin care feeder-ul va putea fi controlat:

1. Animalul de companie apasă pe un buton, acționând servo-motorul, iar utilizatorul primește o notificare pe telefon pentru a fi înștiințat de acest lucru

2. Utilizatorul trimite comanda /feed pe telefon, cu ajutorul unui bot de Telegram, servo-motorul fiind acționat de un server (serializarea si deserializarea unui byte)

3. Utilizatorul trimite comanda /photo pe telefon, ce va determina trimiterea unei poze, pe server cu starea actuala a feeder-ului

Schema bloc:



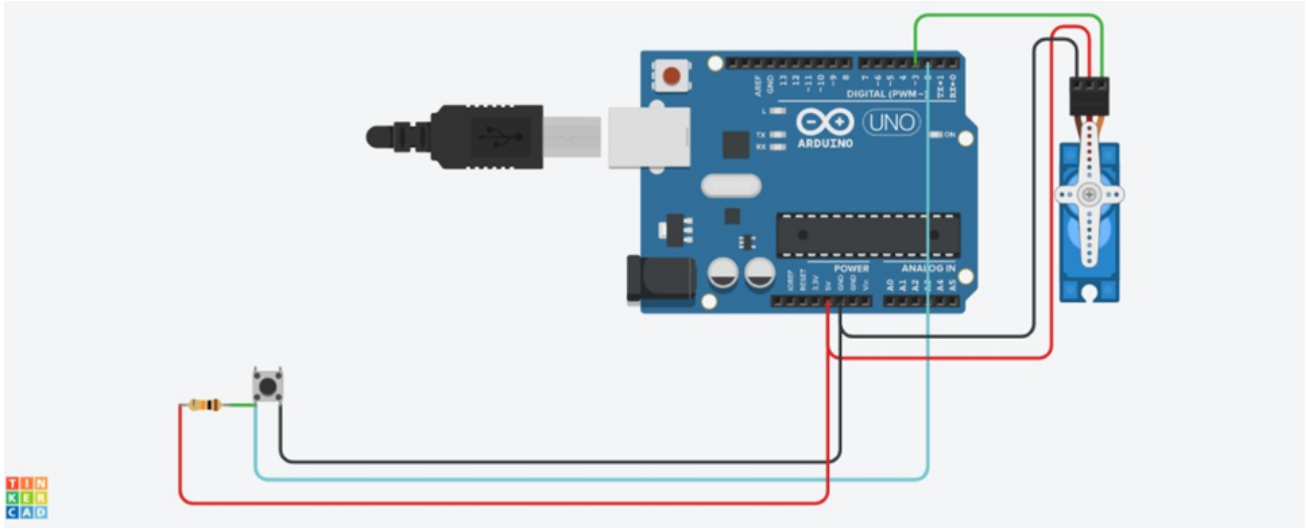
Hardware Design

Componentele utilizate pentru construirea dispenserului:

- Placă de dezvoltare UNO R3 compatibil Arduino x 2
- Servomotor SG90
- Fire pentru conexiune
- Microswitch Type: Snap Action
- Mini-breadboard
- Shield rețea ENC28j60 ethernet
- ESP-32 camera

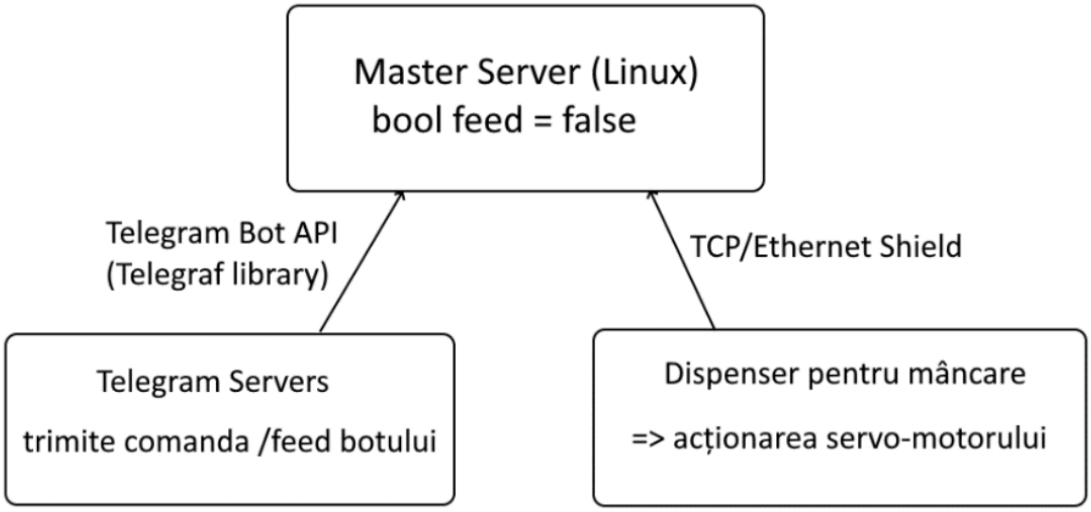
Materiale reciclabile folosite:

- O conservă de suc de tomate și o cutie de Pringles
- O sticlă de plastic pentru rampă si buton
- O cutie de plastic pentru susținerea butonului
- Recipient de mâncare pentru animale



Software Design

În ceea ce privește partea software a acestui proiect, servo-motorul va putea fi acționat prin intermediul aplicației Telegram și al unui buton, aspecte ce vor fi realizate prin prisma protocolului TCP.

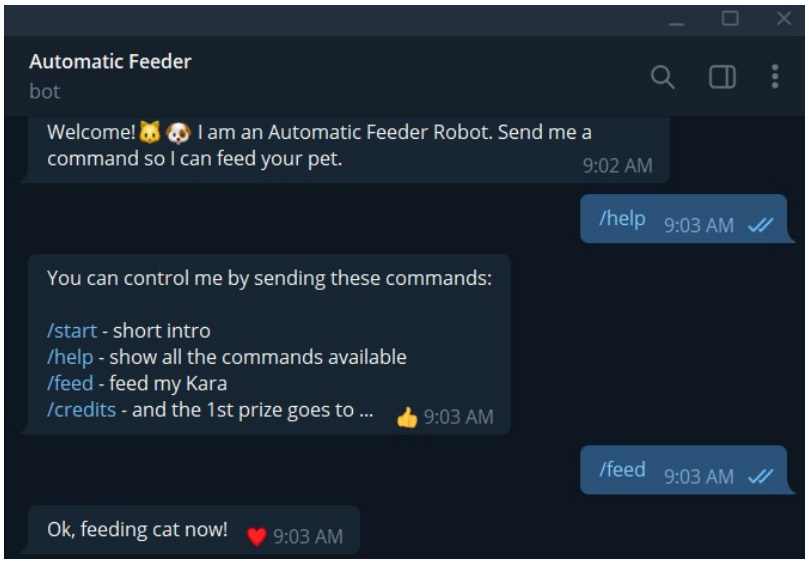


Așadar, va fi necesar un router care se va conecta prin intermediul unui cablu de ethernet la modulul Wi-Fi, conexiunea cu laptopul fiind posibilă și prin wireless. Totodată, codul de Arduino conține o funcție de reconectare automată, reconnect(), în cazul în care nu se mai face conexiunea la master server, prevenind astfel comportamentul imprevizibil generat de potențialele erori sau de fluctuații ale vitezei cu care se transmit datele prin internet.

```
void reconnect() {
  bool reconnected = false;
  while (!reconnected) {
    if (client.connect(IPAddress(192,168,1,121), 1337)) {
      Serial.println("connected");
      reconnected = true;
    }
  }
}
```

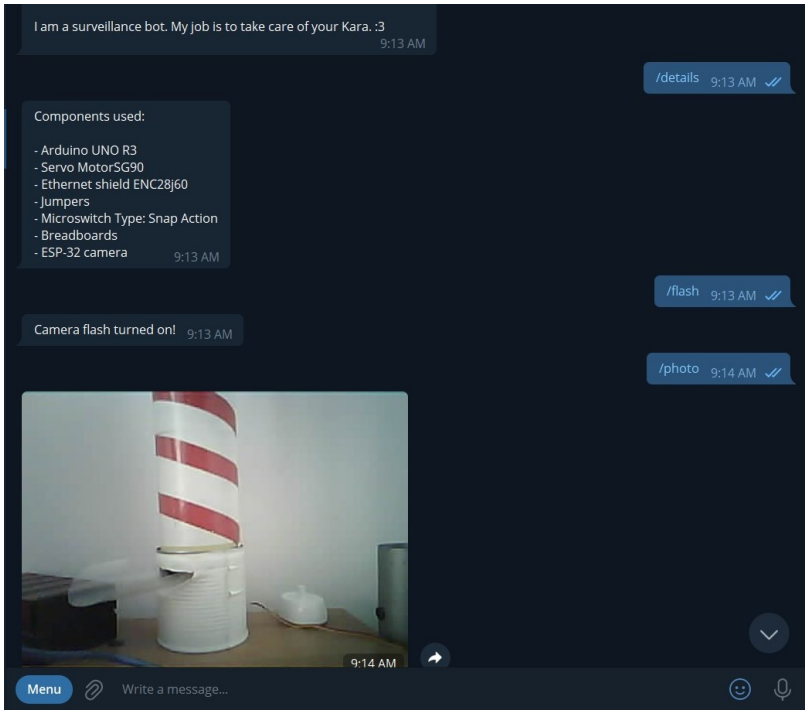
Inițializare botului de Telegram se face prin intermediul unui BotFather cu comanda /newbot, astfel că, utilizatorul trimite niște comenzi pe această aplicație pentru a-și configura propriul chatbot care, va căpăta sens + utilitate practică in momentul in care va fi programat pe calculator. Odată ce botul de Telegram a fost inițializat, vom folosi biblioteca Telegraf împreună cu NodeJS care este un framework pentru siteuri, servicii și aplicații scrise in Javascript.

Comenzile trimise de pe Telegram acționeaza callback-uri lambda pe serverul realizat prin protocolul TCP. Comanda de hrănire, de exemplu, verifică dacă feederul este conectat la serverul de control și ii trimite un byte pentru a determina actionarea servomotorului. Totodată, utilizatorul primește un mesaj pe telefon pentru a fi înștiințat dacă nu au fost erori.



Atunci când feederul primește un byte, acționează servo motorul, iar când serverul primește un byte de la feeder, știm că pisica a apăsat pe butonul fizic, hrănindu-se singură. Pentru pornirea efectiva a serverului se dă comanda node index.js. Codul de pe placa de dezvoltare Arduino presupune conectarea la server cu ajutorul IP-ului local al laptopului si al adresei MAC precum și implementarea celor două scenarii.

Pentru implementarea functionalitatii camerei ESP32 CAM, ce presupune conectarea acesteia la bot-ul de Telegram, am folosit biblioteca Universal Arduino-Telegram Bot.



Comenzile botului de Telegram

- start - short intro (scurta descriere)
- help - show all the commands available (afișarea tuturor acestor comenzi)
- details - show all the components used
- feed - feed my Kara (hrănirea pisicii)
- credits - and the 1st prize goes to ...
- photo - take a photo of the current state of the feeder
- flash - toggles the flash of the camera

Rezultate Obţinute

Testarea solutiei s-a realizat prin intermediul robotului fizic conectat la calculator prin intermediul unui cablu de ethernet. Asigurându-ne de faptul că avem o conexiune Wi-Fi stabilă, am pornit serverul cu ajutorul comenzii node index.js. De mentionat este faptul ca ESP32 CAM dispune de un modul Wi-Fi, conectarea acesteia la bot-ul de Telegram realizându-se wireless.

```
38  /* /feed command  */
39  bot.command('feed', ctx => {
40      if (GLOBALSOCKET != undefined) {
41          sendReply('f');
42          ctx.reply("Ok, feeding cat now!");
43      }
44      else {
45          ctx.reply("Bot not connected!");
46      }
47  })
```

În prima instanță, s-a verificat funcționalitatea de acționare prin apăsare pe buton, eveniment ce produce transmiterea unui anumit tip de date (un byte) către server, ca mai apoi să fie preluat si transformat într-o comandă ce prevede evacuarea mâncării.

Funcționalitatea ce are ca subiect conectarea la aplicația de Telegram prin intermediul API-ului open source, a fost testată, cu serverul pornit, comunicând cu botul pe care l-am creat. Astfel, după ce am setat lista de comenzi (/setcommands) prin BotFather si după ce le-am atribuit câte o semantică (primirea unui mesaj text ca raspuns), am testat comanda /feed ce acționează motorul în mod instant. Codul ce presupune redarea pozei a fost implementat in fisierul TelegramCamera. Acesta primeste token-ul corespunzator bot-ului de Telegram, informatii despre conexiunea Wi-Fi si se foloseste de un client TCP pentru a transmite reda poza, respectiv pentru a porni flash-ul camerei.

Concluzii

Proiectul a fost o experiență plăcută și interesantă în care am aprofundat mai bine anumite noțiuni, am întâmpinat probleme și am încercat să găsesc soluții. Printre problemele pe care le-am întâmpinat, montarea shieldului de Ethernet a prezentat dificultăți deoarece nu am putut folosi o bibliotecă standard. Având în vedere faptul că s-a folosit un shield care nu este original Arduino, am folosit biblioteca UIPEthernet.h de pe github: https://github.com/UIPEthernet/UIPEthernet. În ceea ce privește implementarea camerei ESP-32, nu am reușit să programez pe aceeași plăcuță și partea care ține de feeder și transmiterea pozei prin Telegram, motivul fiind insuficiența pinilor.

Download

feeder-bot.zip

Jurnal

- 18.04.2022 - Alegerea temei proiectului
- 21.04.2022 - 20.05.2022 - Realizarea proiectului
- 26.06.2021 - Intocmirea documentatiei

Bibliografie/Resurse

https://www.youtube.com/watch?v=5MHisFC-_dE

https://how2electronics.com/interface-enc28j60-ethernet-module-with-arduino/

https://www.youtube.com/watch?v=hCvOzzDa1ms&t=124s

https://www.youtube.com/watch?v=kEG8cd32fb0

https://www.chatcompose.com/ro/telegram.html

https://randomnerdtutorials.com/telegram-esp32-cam-photo-arduino/

dispenser_de_mancare_pentru_animale.pdf