

Paws Watch

- Autor: Aradoaie Ioana-Maria
- Grupa: 332CA
- Email: ioana.aradoaie@stud.acs.upb.ro

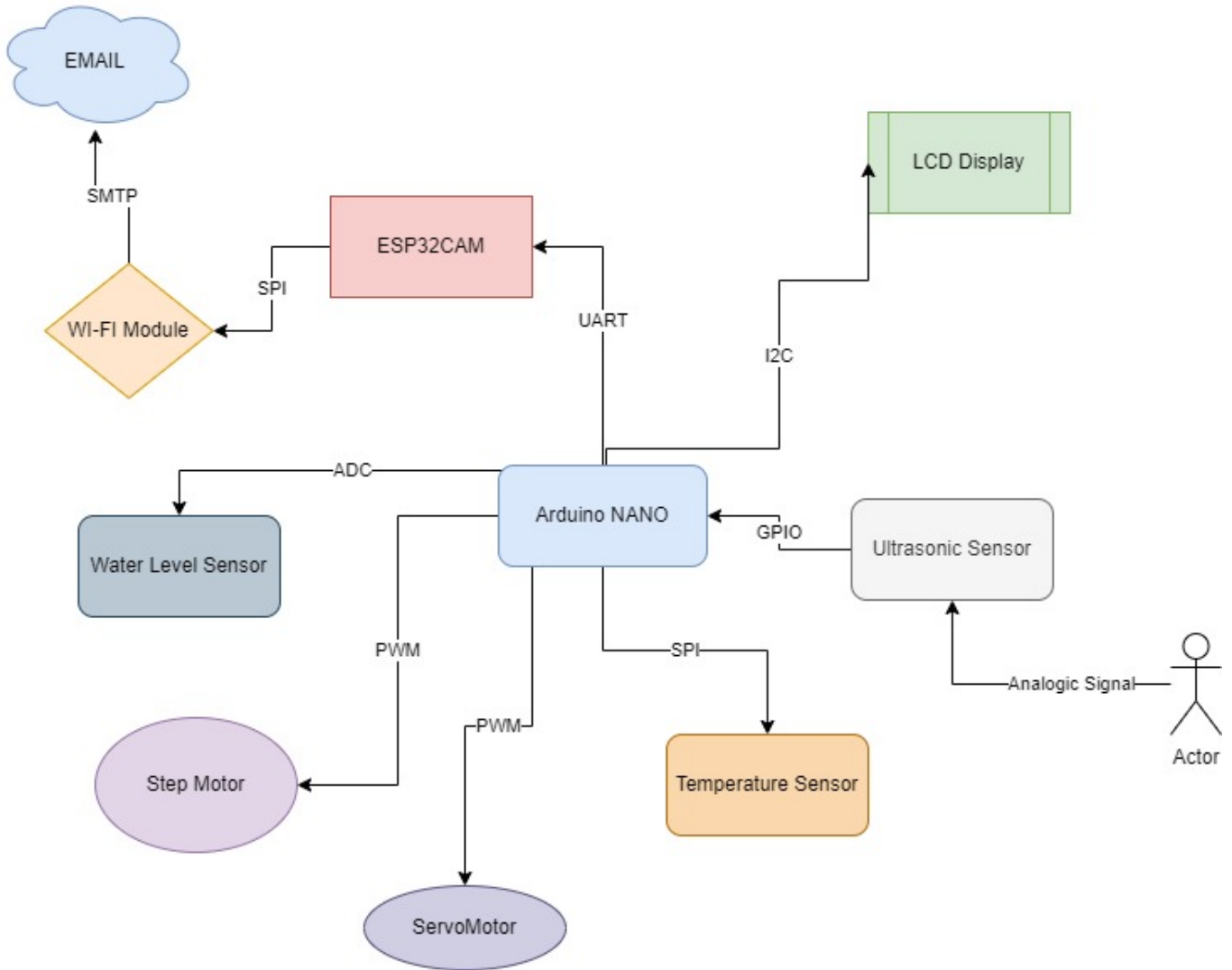
Introducere

Proiectul are ca scop realizarea unui sistem de monitorizare a bunastarii animalului de companie. Acesta vine in ajutorul oamenilor care detin un caine/pisica pe care sunt nevoiti sa il lase nesupraveghiat o perioada indelungata de-a lungul zilei. Ceea ce isi propune acest proiect este sa se asigure ca animalutul dispune in permanenta de apa, facand posibila actiunea de reumplere a bolului dupa ce acesta a baut toata apa.

A avea un animal de companie este o responsabilitate foarte mare pe care multi oameni si-o asuma, dar, din nefericire, acestia nu le pot oferi mereu atentia si grija de care au nevoie. Prin urmare, un astfel de sistem va substitui din atributiile stapanului, in absenta acestuia, oferindu-i si lui o stare de liniste.

Descriere generală

Sistemul se bazeaza pe un controler ESP32 si integreaza o serie de senzori folositi pentru detectarea prezentei cainelui, monitorizarea consumului de apa si a temperaturii mediului inconjurator. Datele colectate sunt afisate pe un ecran LCD si transmise prin e-mail prin intermediul modulului WiFi si protocolului SMTP, permitand proprietarilor sa fie permanent informat cu privire la starea animalutului lor. Sistemul va incorpora si un Step Motor controlat prin PWM pentru a reumple automat bolul de apa atunci cand nivelul acesteia scade sub un prag prestabilit.



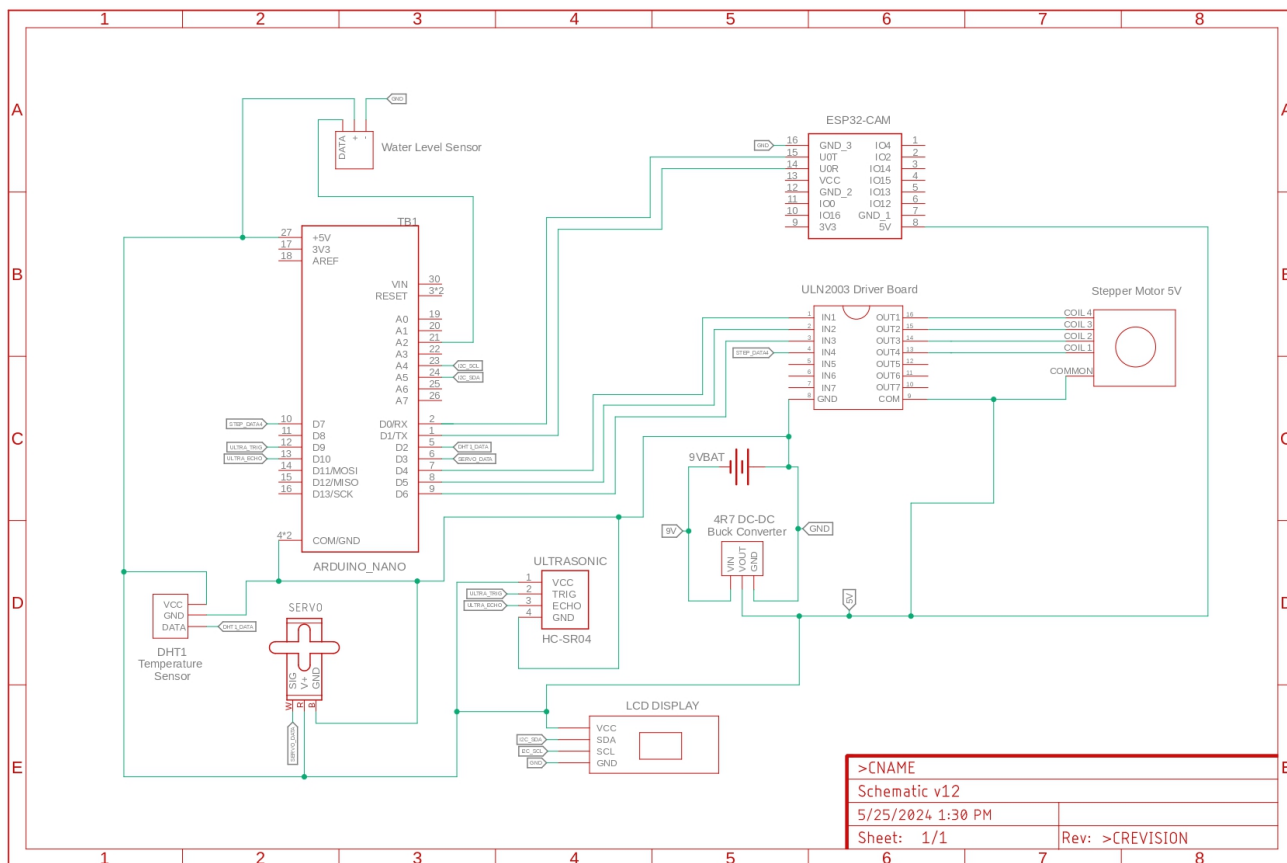
Hardware Design

Lista de piese:

- Arduino Nano ATMEGA328PB
- ESP32-CAM WIFI+BT
- senzor ultrasonic HC-SR04
- senzor de temperatura si umiditate DHT11
- senzor de nivel al apei
- display LCD 16x2
- modul I2C
- stepper motor 5v
- micro servo SG90
- interface board
- sursa 5v
- breadboard
- fire si jumpere

Schema electrica:

Aceasta a fost realizata in Fusion. Am cautat simbolurile componentelor din ansamblu, am importat bibliotecile corespunzatoare in Fusion si am realizat legaturile (de tip NET, cu labeluri pe ele) pentru o mai buna intelegere a diagramei.



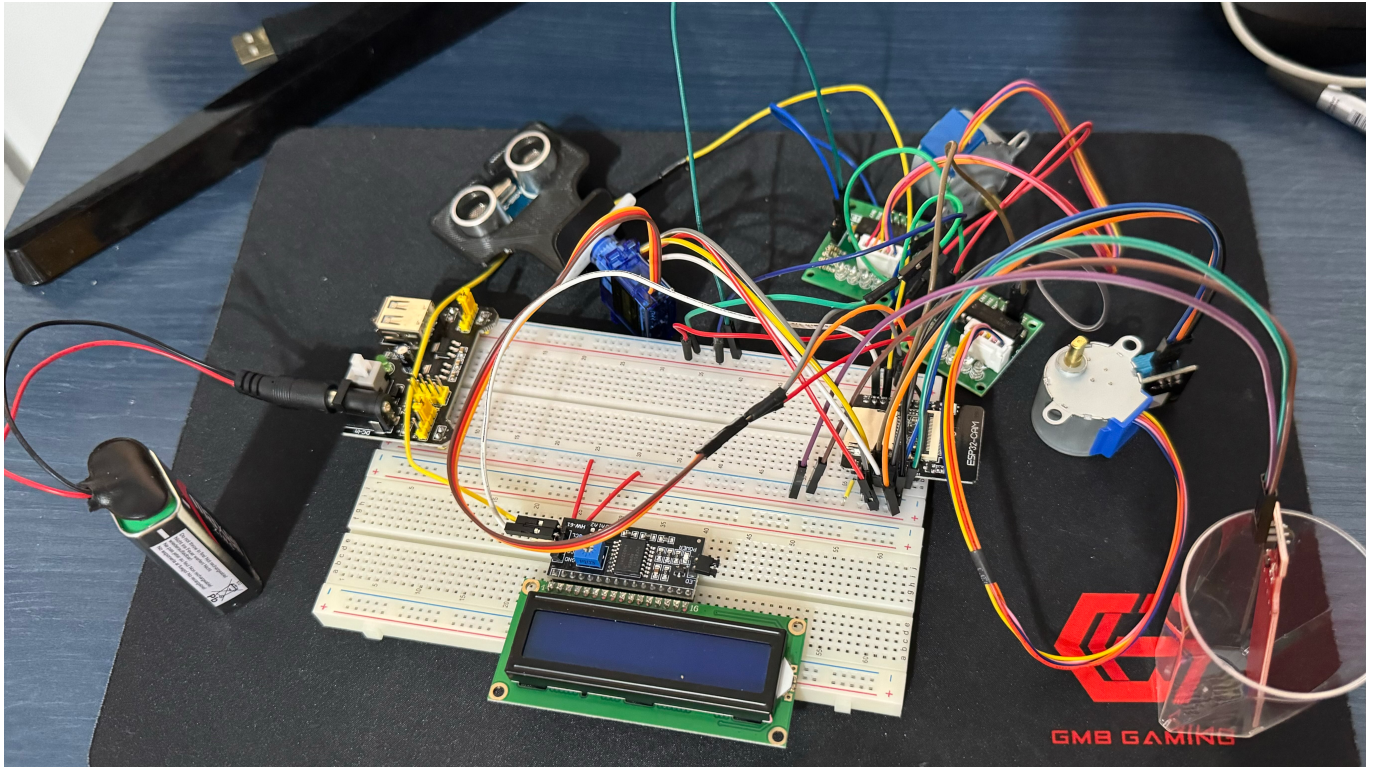
Implementare fizica:

Structura legaturilor dintre componente:

Componenta 1	Protocol	Componenta 2
RX ESP32-CAM	UART	TX Arduino Nano
TX ESP32-CAM	UART	RX Arduino Nano
TRIGGER_PIN (9)	GPIO	Senzor ultrasonic (Trigger)
ECHO_PIN (10)	GPIO	Senzor ultrasonic (Echo)
DHTPIN (2)	GPIO	Senzor DHT11 (Data)
SERVO_PIN (3)	PWM	SG90 Servo motor
SDA (A4)	I2C	LCD I2C (SDA)
SCL (A5)	I2C	LCD I2C (SCL)

A0	ADC	Senzor de nivel al apei (Data)
----	-----	--------------------------------

Implementare:



Software Design

Mediul de dezvoltare

PlatformIO

Motivarea alegerii bibliotecilor folosite în cadrul proiectului:

Bibliotecile folosite au fost necesare pentru a realiza atât comunicatia dintre Arduino și ESP, cât și între acestea și senzori, motoare și ecranul LCD.

Elementul de noutate al proiectului:

Este ieftin, ușor de înțeles și de implementat în propria casă, fără a necesita cunoștințe avansate de electronica.

Utilizarea funcționalităților din laborator:

- **Laboratorul 0: GPIO** - Utilizat pentru senzorul ultrasonic, senzorul de nivel al apei și controlul servomotorului.
- **Laboratorul 1: UART** - Comunicare serială între ESP32-CAM și Arduino Nano.
- **Laboratorul 2: Interruperi** - TimerOne pentru declanșarea periodică a funcției de actualizare a afișajului LCD.
- **Laboratorul 3: Timere. PWM** - Controlul poziției servomotorului cu biblioteca Servo.h și TimerOne pentru actualizarea periodică a afișajului LCD.
- **Laborator 4: ADC** - Citirea nivelului apei cu funcția analogRead() de pe pinul analogic.
- **Laboratorul 5: SPI** - Utilizat indirect prin ESP Mail Client pentru trimiterea email-urilor.
- **Laboratorul 6: I2C** - Controlul afișajului LCD prin interfața I2C folosind biblioteca LiquidCrystal_I2C.

Calibrarea elementelor de senzorică:

• Senzor ultrasonic:

Detectarea distanței: Testarea senzorului la diferite distanțe pentru a asigura măsurători corecte.

Setarea pragurilor: Stabilirea unui prag (ex. 20 cm) pentru a detecta prezența câinelui.

• Senzor de nivel al apei:

Măsurarea nivelului: Testarea senzorului cu diferite niveluri de apă pentru a obține valori precise.

Determinarea pragurilor: Stabilirea unui prag (ex. valoare ADC 500) care indică un nivel scăzut al apei.

• Senzor de temperatură și umiditate (DHT11):

Validarea citirilor: Compararea măsurătorilor cu un termometru și un higrometru de referință.

Ajustarea măsurătorilor: Aplicarea corecțiilor în cod pentru a compensa orice erori constatate.

Librării și surse 3rd-party

NewPing.h: Am folosit biblioteca NewPing.h pentru a gestiona senzorul ultrasonic. Aceasta oferă funcții pentru a măsura distanțele utilizând senzorul ultrasonic într-un mod eficient.

DHT.h: Biblioteca DHT.h este utilizată pentru a citi datele de temperatură și umiditate de la senzorul DHT11. Aceasta gestionează protocoalele de comunicare necesare pentru a obține citiri precise de la senzor.

Servo.h: Pentru controlul servomotorului, am folosit biblioteca Servo.h, care permite poziționarea acestuia la un anumit unghi cu ușurință și precizie.

Wire.h: Bibliotecă standard în Arduino, folosită pentru a comunica cu dispozitive periferice care

utilizează protocolul I2C.

LiquidCrystal_I2C.h: Aceasta simplifică utilizarea ecranului LCD cu interfață I2C, permițându-mi să comunic cu ușurință cu ecranul fără a fi nevoie să implementez manual protocolul I2C.

TimerOne.h: Biblioteca TimerOne.h este folosită pentru a gestiona un timer hardware pe Arduino Nano, pentru a actualiza periodic afișajul LCD.

Algoritmi și structuri implementate

Citirea și maparea datelor de la senzori: Citirea datelor de la senzorul ultrasonic, DHT11 și senzorul de nivel al apei, maparea acestora în variabile utile pentru procesare.

Controlul servomotorului și motorului stepper: Utilizarea bibliotecii Servo.h pentru a mișca servomotorul și a motorului stepper pentru a inclina paharul.

Afișarea mesajelor pe ecranul LCD: Afișarea pe un LCD I2C a temperaturii, umidității și timpului care a trecut de la ultima data când a baut apă.

Gestionarea timpului: Utilizarea funcției millis() pentru a măsura timpul care a trecut de la ultima băutură și afișarea acestuia pe LCD.

Flow firmware

- **Initializare:** Sistemul inițializează senzorii, servomotorul, motorul stepper și LCD-ul.
- **Monitorizarea senzorului ultrasonic:** Dacă câinele se apropie (detectat de senzorul ultrasonic), se măsoară nivelul apei și, dacă este necesar, se adaugă apă în bol.
- **Actualizarea LCD-ului:** La fiecare secundă, LCD-ul este actualizat cu temperatura, umiditatea și timpul care a trecut de la ultima băutură.
- **Trimiterea datelor:** Datele sunt trimise către ESP32-CAM pentru a fi trimise prin email cu ajutorul protocolului SMTP.

Optimizări

Evitarea busy-waiting-ului: Sistemul nu verifică în continuu senzorii, ci doar atunci când este necesar, ceea ce economisește resurse și evită efectuarea unor acțiuni inutile.

Gestionarea stării aplicației: Variabilele booleene dogDrank și start_wash sunt folosite pentru a urmări starea curentă a aplicației și pentru controlul fluxului programului (determina acțiunea următoare în funcție de starea curentă).

Rezultate Obținute

Link Repo GitHub: <https://github.com/aradoai3/PawsWatch/>

Jurnal

- **15 aprilie-30 aprilie** - comenzi de piese
- **1 mai** - primul draft de documentatie
- **10 mai** - logistica pentru hardware (schema electrica, lipituri, cablaj)
- **15 mai** - adaugare implementare harware + documentatia aferenta
- **26 mai** - adaugare repo GitHub si finalizare cod

DEMO

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/mdinica/ioana.aradoaie>



Last update: **2024/05/27 00:58**