

RoboScout

Introducere

RoboScout este o mașinuță controlată de pe telefon prin Bluetooth, care are montată o cameră video și mai mulți senzori de mediu (temperatură, presiune etc). Practic, e un mic robot mobil pe care îl poți trimite în locuri greu accesibile, iar el îți arată ce vede și ce condiții sunt acolo. De asemenea, acesta va avea posibilitatea de a detecta când urmează o coliziune și se va opri din mișcare înainte ca aceasta să aibă loc.

Scopul principal este ca acesta să fie folosit pentru a explora sau verifica zone în care oamenii nu pot ajunge ușor sau în siguranță, cum ar fi spații înguste, zone periculoase sau locuri unde există riscuri (gaze, temperaturi extreme etc).

Am pornit inițial de la o idee simplă: o mașinuță ce poate fi controlată de pe telefon, dar care încet încet a fost dezvoltată după ce mi-au venit mai multe idei.

Am considerat că ar fi o provocare interesantă, deoarece ar fi primul meu contact cu controlul wireless. De asemenea, va fi un exercițiu bun în montarea unui dispozitiv.

Descriere generală

Schema bloc:



Hardware Design

Componentă	Observații
4x Motoare DC 3-6V cu reductor	Trase de L298N
4x Roți de 65 mm	Pentru mișcare stabilă
L298N dual H-bridge	Control motoare DC
Senzor DHT11	Umiditate și temperatură
Senzor HC-SR04+	Ultrasonic
Senzor BMP280	Presiune atmosferică
ESP32-CAM	Pentru cameră, Bluetooth
Arduino Uno	Placă principală de control
Suport baterii 6xAA (9V total)	Sursa principală de alimentare
Fire, breadboard etc.	-

Schema Fritzing (WIP):



Dovadă funcționalitate cameră (capturat din apartamentul meu):

Toggle OV2640 settings

XCLK MHz Set

Resolution ▾

Quality 4 63

Brightness -2 2

Contrast -2 2

Saturation -2 2

Special Effect ▾

AWB ☐

AWB Gain ☐

AEC SENSOR ☐

AEC DSP ☐

AE Level -2 2

Exposure 0 1200

AGC ☐

Gain 1x 31x

BPC ☐

WPC ☐

Raw GMA ☐

Lens Correction ☐

H-Mirror ☐

V-Flip ☐

DCW (Downsize EN) ☐

Color Bar ☐

LED Intensity 0 255

Face Detection ☒

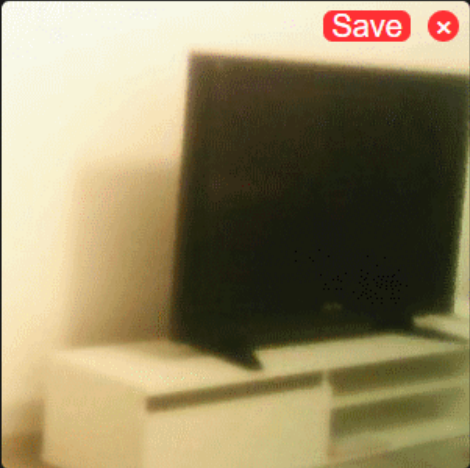
Face Recognition ☐

Get Still Stop Stream Enroll Face

Advanced Settings

≡ Register Get/Set

≡ CLK



Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

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/prj2025/apredescu/andrei.imireanu>



Last update: **2025/05/18 19:58**