Cezarmobilul

Introducere

V-ati dorit vreodata sa aveti un animal de companie loial, dar fara fite si nevoie sa ii dai de mancare? (excluzand bateriile) Vrei sa te ajute cu adusul romului si monstrilor, desi tu ai o mana ocupata cu tiktoku si cealalta cu tema la IA? Say no more, caci exista o solutie perfecta: Cezarmobilul

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât şi hardware însoţită de o descriere a acestora precum şi a modului în care interacţionează.

Exemplu de schemă bloc: http://www.robs-projects.com/mp3proj/newplayer.html

Acesta este un container umblator care poate fi controlat prin bluetooth cu mai multe moduri: sa seada la o anumita locatie sau sa isi urmareasca un stapan. Totusi, acesta trebuie cumva sa isi marcheze teritoriul, de aceea cand este pornit el va intra intr-o stare de recunoastere. Locatia din care o sa plece o sa fie cotetul lui si acolo se va intoarce cand intra in starea default.

In faza de recunoastere/follow el se va folosi de senzori ultrasonici pentru a detecta obstacolele/stapanul

Hardware Design

Aici puneţi tot ce ţine de hardware design:

- listă de piese
- scheme electrice (se pot lua şi de pe Internet şi din datasheet-uri, e.g. http://www.captain.at/electronic-atmega16-mmc-schematic.png)
- diagrame de semnal
- rezultatele simulării

Lista componente

- ESP32 Wroom (are si conectarea seriala la bluetooth builtin)
- 2-3 HC-SR04 (folosind intreruperi)
- 2 roti cu motor DC si 1 driver dual L298N (cu PWM)
- 1 I2C LCD 1602 (Labul cu I2C)
- Set 4 baterii AA + capsula
- Cutie metalica ptr sasiu, eventual niste piese 3D printate daca sparlesc de la unu care are asa cv
- SD card reader (utilizat in stocarea datelor despre obstacolele scanate)
- Un servo? (Probabil pentru a roti senzorii ultrasonici, most likely nu va fi folosit)

Conexiuni de pini

GPIO	Label	Pin componenta
36	Sensor_VP	ADC Bat
32	D32	L298N In1
33	D33	L298N In2
25	D25	L298N In3
26	D26	L298N In4
27	D27	PWM L298N EnA, EnB
13	D13	Trig S1, S2, S3
4	D4	Echo S1
16	D16	Echo S2
17	D17	Echo S3
5	SPI_CS (!)	CS SD_card
18	SPI_CLK	CLK SD_card
19	SPI_MISO	MISO SD_card
21	SDA	SDA LCD
22	SCL	SCL LCD
23	SPI_MOSI	MOSI SD_card

Schema electrica

Software Design

Descrierea mediului de dezvoltare

S-a folosit VScode cu PlatformIO. In cadrul frameworkului de ESP, am folosit Arduino. Personal, as fi

folosit ESP-IDF, doar ca acest lucru includea freeRTOS, iar incluziunea unui sistem de operare nu stiu daca era allowed in cadrul proiectului

Librariile folosite

Incluse in frameworkul Arduino:

- Arduino
- Wire
- Wifi
- ArduinoOTA

3rd party:

- WebSerial + dependencies
- LiquidCrystalI2C

Motivatia librariilor

Am folosit ArduinoOTA pentru a putea face flashing de la distanta, altfel ar fi insemnat sa ma tot car cu un fir si sa deschid la o frecventa foarte ridicata sasiul. Ar fi fost o motivatie parnaie sa ii dau si o “coada” catelului, dar oricum ar fi prezentat o incomoditate mare. On the same note, am folosit WebSerial pentru a putea avea un serial monitor, altfel degeaba aveam OTA. As fi putut sa fac printing pe LCD, dar unele chestii pot fi prea specifice/mari, si mai poti da si comenzi astfel.

For some whatever reason, nu exista un builtin pentru LCD_I2C, asa ca a trebuit sa se foloseasca o dependenta externa. E totusi frumos ca platformio face acest proces seamless

Elementul de noutate

undefined

Functionalitatile de laborator

Overall s-au folosit 4 laboratoare

1. (y) GPIO - god dammit speram sa se puna asta
2. (n) UART - not used
3. (y) Intreruperi - Folosit in cadrul echo-urilor de la senzorii ultrasonici pentru precizie ridicata (pacat ca e degeaba ca oricum e un senzor de 5 lei care functioneaza cand are el chef dar shhh)
4. (2*y) Timere/PWM - Timerele sunt folosite pentru a face periodic trigger pe ultrasonice, iar PWM a fost folosit pentru a controla cat de rapid se misca motoarele
5. (y) ADC - Masurarea voltajului bateriilor NiMH, s-a folosit un divizor 1k/(1k+10k) pentru a fi in

range-ul de 3.3V, de la maximul de 12V se ajunge la un teoretic 1.09V +- eroarea rezistorilor, si valoarea minima ar fi 0.87V (1.2V * 8 / 11)
6. (?) SPI - SD Card?
7. (y) I2C - LCD (Literalmente altfel nu mai intra pe pini un dinasta, am fost obligat de soarta aici)

Piesele de puzzle

Scheletul

S-a folosit un main care contine mai multe programe, OTA-ul si US-ul. In acelasi timp se updateaza si LCD-ul si motoarele. Motoarele si LCD-ul au fost facute pe principiul de singleton, LCD-ul contine mai multe parti ale statusului precum ar fi senzorii ultrasonici, ADC-ul, cat si in cazul in care se face flashing prin OTA care ar fi progresul pentru acesta. S-a si initializat WebSeriala pentru a putea face printuri generice cat si a putea da niste comenzi simplute pentru robot.

Interactiunea

In un program separat se pot folosi datele de senzori din LCD (Mi-a fost sincer lene sa creez o clasa diferita pentru asta cut me some slack), si se pot actiona motoarele. Datele sunt acquired automat cu ajutorul timerelor, dar setarea lor in starea globala se face doar la update-ul senzorului respectiv, pentru a sincroniza codul si a asigura ca datele senzorilor sunt luate cum trebuie.

Testarea componentelor

Am printat datele senzorilor pe LCD, si am vazut ca uneori dau valori, alteori nu. Pentru omul ghinionist care o sa aibe onoarea sa imi citeasca codul, o sa observe acolo o chestie '#ifndef PARNAIE_DE_LA_GPT'. Ce reprezinta asta? Senzorii astia functioneaza cand vor ei, dar exista o logica in existenta lor nefunctionala. Ideea e ca GPT-ul mi-a cacat un cod, si acela stiu clar ca functioneaza daca toti senzorii dau valoarea 0. Daca dau flash din nou la codu meu dupa, ca prin magie, (cel mai probabil!) functioneaza... daca nu, ia de scoate si baga senzorii la loc. 

Mda, din pacate, trebuia sa ma astept ca daca dai 5 lei, 5 lei o sa primesti inapoi, si mi-am invatat lectia fix cu senzorii astia. M-as fi gandit ca daca se rupe o masa de la senzorii astia, se intampla din cauza ca nu dau distanta la timp + accurately si imi intra robotu in ea, nu credeam ca o sa fie din cauza ca dau eu cu pumnu in masa mai rau ca la lol ca nu vor chinezariile astea astia sa mearga decat cand vor ele...

La restu din fericire a mers brici, functioneaza ca unse. Din pacate totusi speram ca rotile omnidirectionale sa preia o mare parte din loadul greutatii, dar tot se chinuia grav de la un amarat de borsec de 2 litri. Asta si un pic tot mai depinde de centrul de masa mobilitatea vehicului

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.
Momentan, nu am apucat sa implementez ce ma asteptam sa se poata implementa. Eram si eu baiat ambitios, credeam ca puteam sa il fac sa scaneze camera, sa salveze topologia, sa treaca prin ea la anumite checkpointuri... In schimb am ajuns la o janghina “open-loop” care depinde foarte mult de distributia de greutate, ca altfel ori mai deraiaza ori se misca foarte greu. De ce ii zic open loop daca are senzori ultrasonici? Simplu: ultrasonicii doar imi zic distantele, ca sa putem evita obstacole in drum drept. nu isi stie nici pozitia si nici rotatia, si chiar daca s-ar fi putut improviza ceva cu ultrasonicul pentru a se putea detecta “pozitia” in topologia curenta, la chinezariile astea de 5 lei sincer mai degraba ai lansa o racheta in drum drept si sa nimereasca pe marte.

Oricum, o chestie e clara: de frumos e frumos, de destept e frumos. Atat s-a putut.mp4

Concluzii

Sa mor eu de mai folosesc vreodata in viata mea ultrasonice in un proiect

Download

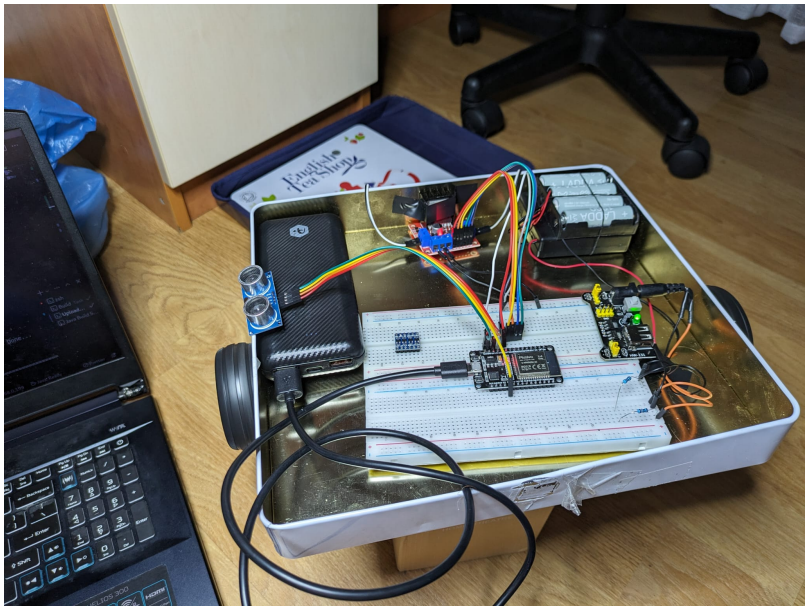
Atat s-a putut.zip

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

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele 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

Motorul care merge:



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

Bibliografie/Resurse

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

Export to PDF