

Doza de sănătate

I. Utilitate practică

În urma unei sesiuni de brainstorming, am realizat că un număr ridicat de persoane care fac parte din comunitatea noastră pot beneficia de un ajutor atunci când vine vorba de medicație. Am observat că există multe persoane care au nevoie de administrarea unor diferite pastile la anumite ore, astfel am creat un robot care îmbunătățește procesul.

Robotul nostru, care poartă numele de EMY – acronim de la „Emergency”, este gândit pentru a ușura și eficientiza acest proces, astfel persoana care îl folosește nu mai este nevoită să stea cu această grijă. La începutul zilei, utilizatorul inserează pastilele prin partea superioară a robotului și cu ajutorul ecranului tactil setează ora individuală la care trebuie să ia fiecare pastilă.

II. Mecanică

Robotul utilizează plăcuța de control Arduino MEGA 2560 la care este atașat un ecran LCD TFT cu diagonală de 2.4 inch prin intermediul unui Shield care reduce voltajul oferit de Arduino – 5V la voltajul acceptat de ecran – 3.3V. În adăția acestui ecran, la placuță, este atașat un motor Stepper controlat de o placuță de driving EasyDriver ce acționează dispenserul cu pastile. De asemenea, la plăcuța Arduino este conectat un Tiny RTC (Real Time Clock), un modul buzzer și un servomotor SG90. Tot acest sistem este alimentat cu ajutorul a doi acumulatori care se conectează la o placuță de alimentare și care oferă întregului ansamblu un voltaj de 7V.

INDEX	NUME PIESĂ	TIP
1.	Arduino MEGA 2560	Microprocesor
2.	TFT LCD 2'4''	Display / Senzor Touch
3.	LCD Shield	Componenta activă
4.	Stepper Motor	Motor
5.	Easy Driver	Microcontroller
6.	Servomotor	Motor
7.	Tiny RTC	Componenta activă
8.	Plăcuța alimentare	Componenta activă

III. Electronică

Arduino MEGA 2560

Plăcuța Arduino MEGA 2560 are integrat un microprocesor, dar scopul ei principal este acela de a se comporta ca un controller pentru toate componentele electrice din ansamblul ce formează robotul, astfel încât aceasta controlează ecranul LCD, motorul Stepper și servomotorul.

Ecran LCD TFT 2'4"

Ecranul LCD TFT permite atât afișarea unui text/imagini pe acesta cât și transmiterea datelor către Arduino, acesta fiind un ecran tactil. Am ales să folosim acest tip de ecran pentru a nu fi nevoiți să implementăm altfel de butoane, dar și pentru a face interfața robotului mult mai prietenoasă pentru utilizator.

Shield pentru ecran

Rolul Shield-ului este de a reduce voltajul pe care plăcuța Arduino îl oferă la 3.3V - numărul de volți acceptați de ecranul TFT, fără acest Shield ecranul s-ar fi ars.

Motor Stepper

Motorul Stepper sau „motor pas cu pas” funcționează pe baza a 2 faze care îi dau rotația astfel încât atunci când curentul trece printr-una dintre cele două faze motorul avansează cu un pas. Numărul total de pași pentru o revoluție completă a acestui model de motor este de 1600 de pași.

Plăcuța EasyDriver

Plăcuța EasyDriver este un microcontroller care se conectează la Arduino, la motor și la sursa de alimentare. Aceasta funcționează în următorul fel: atunci când primește un semnal de la Arduino, lasă curentul de la sursă să treacă alternant prin cele 2 faze astfel rotind motorul cu un pas.

Servomotorul

Pentru servomotor nu este nevoie de o plăcuță de driving, el poate funcționa conectat direct la Arduino prin intermediul a 3 cabluri (5V/VCC, GND, PWM). Astfel motorul se rotește atunci când primește un semnal de la cablul conectat la pinul de tip PWM.

Placă de alimentare

Cum acumulatorul nu poate fi conectat direct la placa EasyDriver, am folosit o placă de alimentare pentru a adapta modul de transmitere al curentului, astfel acumulatorul este conectat la placa de alimentare care la rândul ei este conectată la microcontrollerul EasyDriver.

Tiny RTC

Placă Tiny RTC funcționează ca un ceas chiar și când placa Arduino nu este alimentată deoarece este conectată la un acumulator separat. Astfel pe ecranul robotului este afișată ora exactă indiferent de cât de mult timp robotul a fost oprit.

IV. Software

Robotul se folosește de diferite librării, unele deja implementate în mediul de lucru Arduino IDE - **Wire.h**, **Servo.h**, altele importate manual de către programator - **UTFT.h**, **URTouch.h**, **DS1307RTC.h**, **TimeLib.h**, pentru a putea acționa toate componentele și pentru a permite să primească informații de la anumite componente cum ar fi ecranul sau RTC-ul.

În ceea ce privește autonomia robotului, acesta nu are nevoie de niciun fel de asistență externă pentru a funcționa în afară de interacțiunile utilizatorului cu el care constau în introducerea pastilei și setarea orei la care trebuie administrată.

Programul este construit eficient, acesta folosind structuri iterative în loc de cele recursive pentru a reduce cât mai mult posibil spațiul ocupat în stivă și timpul executării. În același timp, pentru a optimiza și mai mult timpul de rulare, structurile repetitive au fost folosite doar când erau necesare și au fost găsite alte metode de rezolvare a unei probleme acolo unde acest tip de structuri păreau să fie eficiente.

Scriptul este organizat pe module, astfel încât un anumit proces al robotului poate fi modificat individual fără a risca ca modificarea să dăuneze altui proces. Pe lângă această structură modulară a codului, în fișierul sursă sunt prezente comentarii care explică fiecare funcție implementată de programator și rolul fiecărei variabile.

**MAI MULTE DETALII PRINTRE CARE
ȘI CODUL, CAD-UL, IMAGINI CU
ROBOTUL SE GĂSESC AICI:**

