

Whac a Mole

Introducere

Dispozitivul are ca scop testarea vitezei de reacție prin intermediul unui joc inspirat din ideea clasică a jocurilor de tip „Whac-A-Mole”.

Scopul jocului este să nimeri sau să atingi (cu degetul, ciocan de jucărie sau alt obiect) gaura din care iese „viermele” cât mai repede, înainte ca acesta să dispară.

Dispozitivul poate fi folosit și în educația copiilor care implică învățarea culorilor, literelor sau numerelor – viermii având etichete pe ei.

Tabla de joc are mai multe găuri (în cazul nostru 4), aranjate într-o rețea.

În fiecare gaură poate apărea, la întâmplare, un „vierme”

Viermele apare doar pentru o perioadă scurtă, apoi dispare.

Jucătorul trebuie să fie rapid și atent, să „lovească” sau să atingă gaura exact atunci când viermele apare.

Descriere generală

Punctaj: este contorizat un punct pentru fiecare vierme nimerit.

Durata: jocul are o durată fixă (60 de secunde), iar scopul este obținerea celui mai mare scor posibil.

Gradul de dificultate poate fi crescut sau scăzut prin modificarea frecvenței cu care apare viermele.

Modul de realizare

Se aplica cunoștințele din:

Laboratorul 2- Întreruperi, rutine de tratare a întreruperilor

Laboratorul 3 - Lucru cu timere; PWM

Laboratorul 4- Utilizarea conversiei analog-digitală

Laborator 5 - Utilizarea comunicatiei cu un periferic prin intermediul protocolului SPI.

Placa de dezvoltare Arduino comanda aleatoriu 4 servomotoare care simuleaza iesirea din gaura a viermelui.

Viteza cu care este generata comanda de iesire a viermelui este variabila fiind reglata cu un potentiometru. Tensiunea variabila dependenta de pozitia cursorului potentiometrului este folosita de convertorul analog digital al microcontrolerului care va seta prescalarul ce da intreruperea folosita la comanda servomotoarelor.

Starile de apasat sau neapasat a celor 4 butoane sunt folosite ca semnale de intrare si daca este coincidenta intre apasare si aparitie se incrementeaza un contor.

Numarul de aparitii ale viermelui si numarul de reusite se memoreaza . Aceste sunt prelucrate (se calculeza rata de reusite) si sunt afisate in timp real prin intermediul protocolului SPI la un display

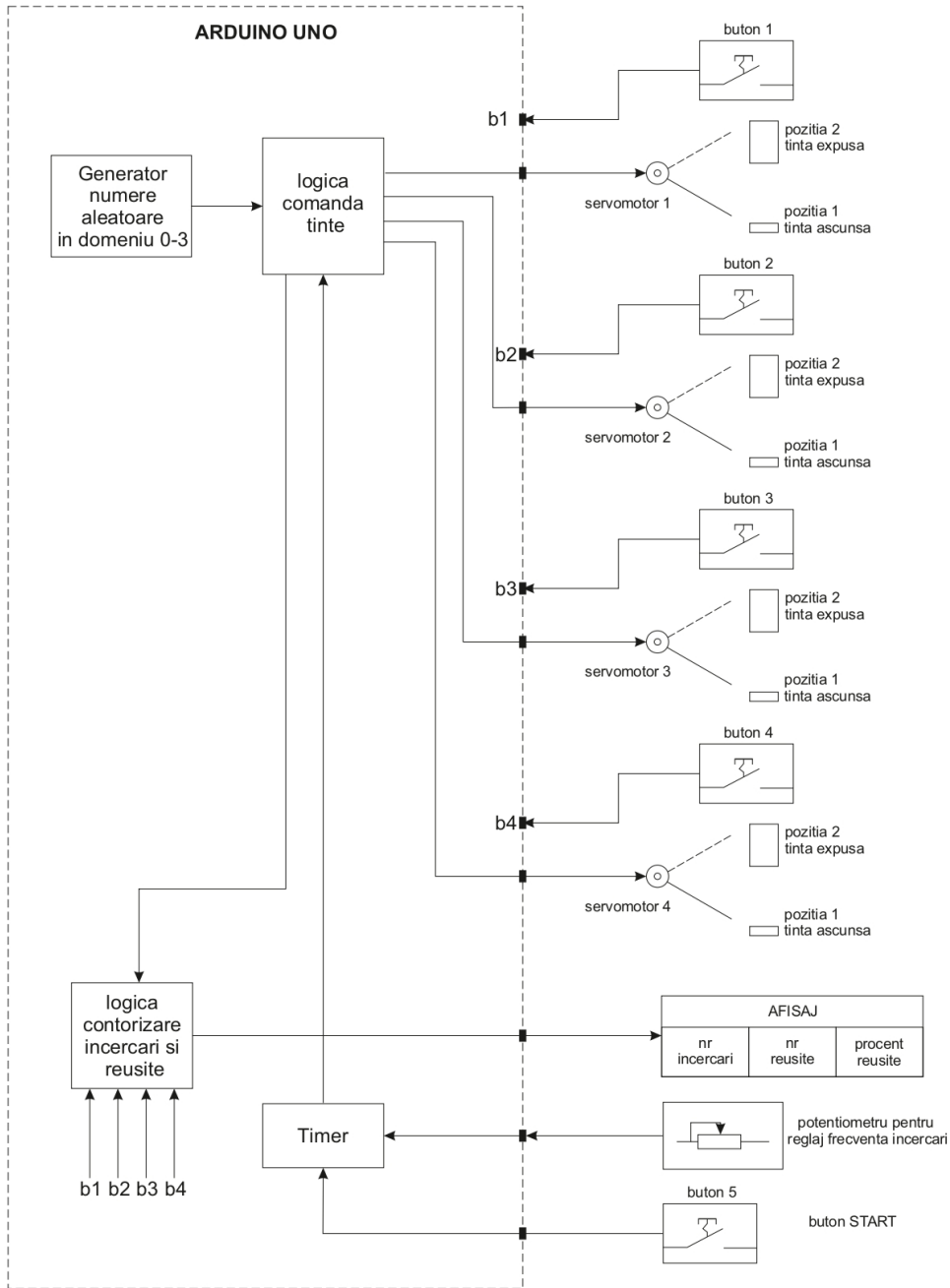
Bill of Materials

Componente:

```
* servomotor
https://www.bitmi.ro/electronica/servomotor-sg90-180-grade-9g-10496.html
* placuta
https://www.bitmi.ro/electronica/placa-de-dezvoltare-originala-arduino-uno-r-ev3-11135.html
* shield
https://www.bitmi.ro/electronica/shield-cu-mini-breadboard-pentru-prototipuri-compatibil-arduino-uno-r3-10387.html*
* butoane
https://www.bitmi.ro/electronica/modul-4-butoane-compatibil-arduino-11739.html
* alimentator https://www.bitmi.ro/produse?c=alimentator+5v
* display
https://www.bitmi.ro/modul-cu-ecran-lcd-1-8-128x160-tft-spi-serial-10996.html?gad\_source=1&gbraid=0AAAAADLag-l-6kiM00NKNe48A8ls224S&gclid=Cj0KCQjwoNzABhDbARIsALfY8VMnb-XmQVW9C1-ztcAgU3pBvpWdiGxeQS6am921yAykY5ywtsl8uXAaAnVXEALw\_wcB
```

Schema bloc

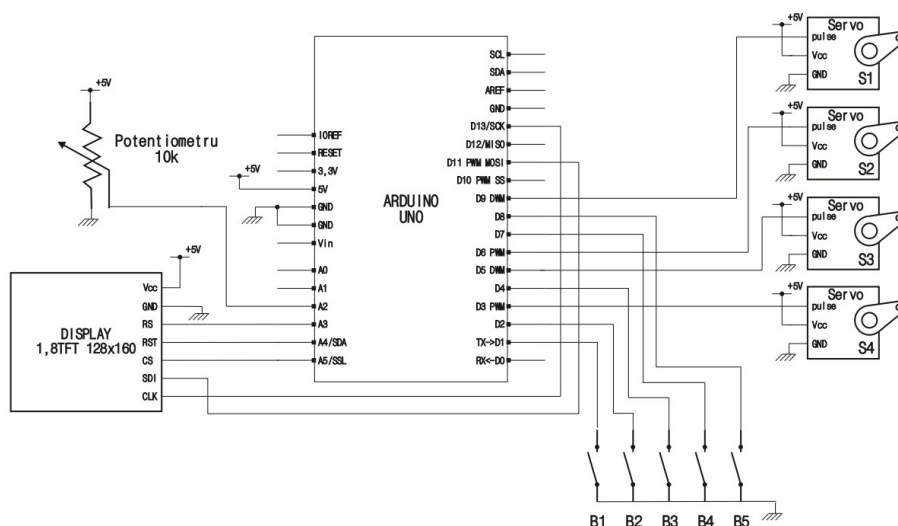
DISPOZITIV DE TESTARE-MASURARE A RASPUNSULUI LA STIMULI - joc "Prinde viermii"



Cablaj



Schema electrica



Descriere functionalitate hardware

- S1,S2,S3,S4 – servomotoare comandate cu modulatie PWM de tip SG90 care vor fi programate sa ridice bratul 90 de grade simuland astfel aparitia viermelui.
- B1,B2,B3,B4,B5 - butoane de tip punch cu revenire
- Potentiometru - pentru reglarea vitezei aparitiei viermelui de la 0.5s la 2s; este de tip liniar cu valoarea de 10Kohmi.
- Displayul este tip LCD de 1.8 inch cu rezolutia maxima de 128x160 , interfata SPI care afiseaza 3 informatii: 1. Nr de aparitii vierme(Na), 2. Nr de reusite(coincide numarul gaurii unde a aparut viermele cu apasarea butonului asignat) (Nr), 3. Procentul de reusite $\% = (Nr/Na) \times 100$.
- Curentul maxim pe care il consuma montajul este suma urmatoarelor consumuri partiale:
 Arduino(datasheet) = 100mA + Potentiometru(5V/10kohmi) = 0,5mA + Servomotor (datasheet = 250mA) Observatie importanta: cu toate ca sunt 4 la numar, ele se manevreaza aleatoriu, niciodata simultan + Display(datasheet) = 100mA TOTAL= 450.5mA. Avand in vedere ca valoarea maxima recomandata pentru alimentarea din portul USB 2 (laptop)este de cca 500mA rezulta ca montajul poate functiona in siguranta alimentat din portul USB. Pentru siguranta in functionare , protejarea laptopului in caz de scurtcircuit in montaj si realizarea functionarii independente a montajului , acesta dupa incarcarea si testarea programului va fi alimentat de la o sursa externa de 5V si curent maxim mai mare de 1A.

Justificare alegere pini

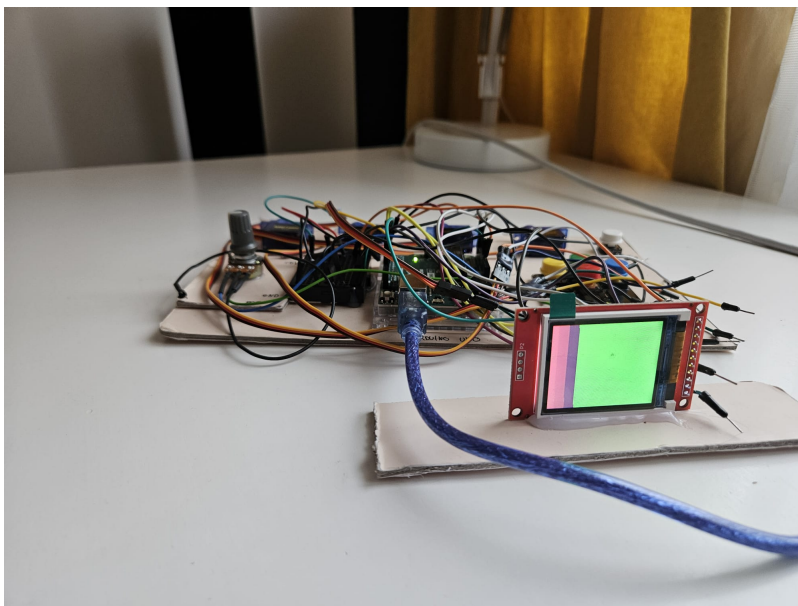
Componente:

1. → S1,S2,S3,S4 - servomotoare
2. → B1,B2,B3,B4 - butoane utilizate in joc, corespunzand “viermilor” indicati de S1,S2,S3,S4
3. → B5 - buton de on/off
4. → placuta Arduino Uno

5. → mini breadboard
6. → potentiometru
7. → display SPI

- Fiecare dintre S1-S4 are 3 fire: maro(GND), rosu(VCC) si portocaliu(cablu de date). Toate cele 4 cabluri de date au fiecare nevoie de cate un pin digital(0-13) la care sa se conecteze pe placuta, dar din cauza principiului PWM utilizat de catre servomotoare, pentru aceste cabluri de date am doar 6 pini digitali disponibili(cei marcati cu ~ pe placuta):3,5,6,9,10,11. Am ales urmatoarea asignare: S1 cu 3, S2 cu 5, S3 cu 6 si S4 cu 9.
- Pini digitali ramasi liberi sunt: 0,1,2,4,7,8,10,11,12 si 13.
- Display-ul de SPI prezinta urmatorii pini: VCC,GND,CLK,SDA,RS,RST si CS. In datasheetul sau avem impuse urmatoarele corespondente impuse pentru Arduino Uno: CLK la 13, SDA la 11, RS la A3, RST la A4 si CS la A5. Pini analogici de pe placuta sunt A0,A1,A2,A3,A4,si A5.
- Pini digitali liberi ramasi sunt: 0,1,2,4,7,8,10,12.
- Pini analogici liberi ramasi sunt: A0,A1,A2.
- Butoanele B1- B5 trebuie conectate la niste pini digitali liberi ramasi pe placuta. Am ales urmatoarea asignare: B1 la 1, B2 la 2, B3 la 4, B4 la 7 si B5 la 8.
- Pini digitali liberi ramasi sunt: 0,10,12.
- Potentiometrul are 3 pini: GND, VCC si unul care trebuie conectat la un pin analogic de pe placuta; aleg sa il conectez la A2.
- Pini analogici liberi ramasi sunt A0 si A1.
- B1-B4 se afla toate pe aceeaasi componenta fizica, deci au GND-ul comun, in rest fiecare avand cate un pin ce trebuie conectat la un pin digital(0-13) de pe placuta de dezvoltare. B5, la randul sau, are 2 pini: unul conectat la GND, altul la un pin digital.
- Bineinteles, vom conecta pini de 5V si GND de pe placuta cu cate un rand din mini breadboard.

Dovada functionare



- Dupa conectarea componentelor conform cu ce am mentionat anterior, pentru verificarea functionalitatii am ales sa afisez pe display ceva la intamplare, considerand ca orice e diferit de un ecran alb (cum afisa display-ul fara sa fi rulat vreun cod pentru el) este o justificare valida a functionarii.

Motivare alegere biblioteci folosite

- Servo.h : am ales folosirea Servo.h deoarece este biblioteca oficiala Arduino pentru a controla servomotoare, permitand o manevrare facila a celor 4 servomotoare utilizate prin functiile attach(pin_x) pentru configurarea servo-ului cu pinul pin_x de pe placuta Arduino si write(unghi_x) pentru pozitionarea bratului de plastic al servomotorului la unghi_x grade, pentru miscarile de "aparitie", respectiv "revenire in vizuina" a unui "Mole"
- Adafruit_GFX.h: uitandu-ma in datasheet-ul display-ului SPI, fiind vorba de un controller ST7735, prima sugestie a library manager-ului din Arduino a fost Adafruit. Am folosit aceasta biblioteca pentru functionalitatile de desenare.
- Adafruit_ST7735.h: desi toate functionalitatile necesare pentru desenare sunt implementate in Adafruit_GFX.h, a trebuit sa folosesc si Adafruit_ST7735.h deoarece este specializat pentru afisari pe ST7735, Adafruit_GFX.h avand implementari doar la modul general, nestiind specificatiile dispozitivului pe care se va face concret desenarea.
- SPI.h: o folosesc pentru transmiterea datelor catre afisor(ecran) pentru afisarea datelor.

Evidentiere element de noutate al proiectului

- Se evidentiaza ca element de noutate reglarea gradului de dificultate(viteza de aparitie a unui "Mole") prin intermediul unui potentiometru (intre 0.5s pentru dificultate mare si 2s pentru dificultate mai scazuta).

Justificare utilizare functionalitati din laborator in cadrul proiectului

- Laboratorul 2: Intreruperi ==> Am setat timer-ul 2 sa genereze o intrerupere care are ca efect oprirea jocului dupa un timp stabilit(60 de secunde) si care poate fi modificat prin inlocuirea timpului de divizare a frecventei de numarare(prescaler-ul).
- Laboratorul 3: Timere. PWM ==> Comanda servomotoarelor s-a facut prin stabilirea unghiului de rotatie de la 0 la 90 de grade si implicit a gradului de modulare a factorului de umplere(duty cycle) corespunzator unghiului stabilit. De asemenea, pinii de comanda ai servomotoarelor au trebuit obligatoriu sa faca parte din grupa pinilor care suporta operatii PWM(pinii cu ~ de pe placuta Arduino Uno: 3,5,6,9,10,11).
- Laboratorul 4: ADC ==> Am folosit conversia analog-digitala(ADC-ul pe Arduino Uno e pe 10 biti, deci valori intre 0 si 1023) pentru a avea o corespondenta intre pozitia butonului potentiometrului si valoarea frecventei de aparitie a viermelui.
- Laboratorul 5: SPI ==> Am folosit materia din acest laborator deoarece am ales sa afisez datele corespunzatoare starii jocului pe un display folosind SPI, care e compatibil cu biblioteci accesibile din library manager in Arduino IDE.

Calibrare elemente de senzoristica

- Pentru pozitia de minim a potentiometrului(maxim stanga, 0V), conform maparii facute prin program, am obtinut frecventa cea mai mare de aparitie a viermelui($1 / 0.5s = 2Hz$), iar pentru pozitia de maxim a potentiometrului(5V, maxim dreapta) am obtinut frecventa cea mai mica de aparitie($0.5 Hz = 1 / 2s$)

Explicare schelet si interactiune intre functionalitati

- In partea de set-up configuram butoanele in stare de pull-up si initializam afisajul pe display-ul SPI si timer 2 pentru a cauza mai tarziu terminarea jocului dupa 60 de secunde printr-o intrerupere.
- Din punct de vedere logic, jocul este gandit la nivel de 6 stari, avand urmatoarele semnificatii:
 - START_GAME = starea initiala; e responsabila doar de contorizarea momentului in care a inceput propriu-zis jocul, dupa care trecem in starea CHOOSE_MOLE
 - CHOOSE_MOLE = se asteapta alegerea unui numar aleator care va desemna "Mole-ul" care iese la suprafata. Mai intai se verifica daca jucatorul a ales sa reseteze jocul(a apasat pe B5) sau nu. Daca se continua jocul curent, alegem un "Mole" aleator (diferit intre iteratii consecutive pentru un joc mai interesant), ridicam steguletul corespunzator acestuia, crestem numarul de aparitii si tranzitionam in starea SURFACE_MOLE
 - SURFACE_MOLE = un "Mole" este la suprafata si trebuie sa fie lovit pentru a obtine puncte(trebuie ca butonul corespunzator sa fie apasat in tr-un timp mai mic decat cel de reactie). Cand apasam pe buton sau expira timpul, "Mole-ul" intra in vizuina si mergem in starea de actualizare a tabelii(REFRESH_SCORE).
 - REFRESH_SCORE = aici actualizam statisticile, dupa care ne intoarcem in CHOOSE_MOLE
 - RESET_WAIT = ajungem in aceasta stare daca se apasa pe B5(RESET_BUTTON de la pin 8); se reseteaza jocul dupa ce tin apasat macar jumătate de secunda, dupa care eliberez
 - GAME_OVER = stare finala; ajung aici dupa ce expira timpul
- Explicare setare durata la 60 de secunde
 - Frecventa pe Arduino Uno este $16MHz = 16\,000\,000 Hz$. Prin setarea prescaler-ului la 1024 rezulta $16000000 / 1024 = 15625$ de tick-uri pe secunda. Prin setarea registrului OCR2A la 249, programul va itera prin 250 de valori(de la 0 la 249) pentru a produce a intrerupere \Rightarrow o intrerupere are loc periodic la $250 / 15625 = 0.016s \Rightarrow$ este nevoie de $60s / 0.016s = 3750$ de intreruperi ca sa pot sti ca a trecut un minut, timp dupa care trebuie sa opresc jocul.

Rezultate Obținute

- Un joc interactiv pe care o sa il mai joc in viitor.

Demo video proiect cu explicatii

- <https://www.youtube.com/watch?v=UiAcehrmnwM>

Link Github

- <https://github.com/dariuuss/software-pm-milestone-3>

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/avaduva/darius.ignatescu>



Last update: **2025/05/25 06:13**