

2

Clepsidra Digitala

Introducere

Proiectul constă în crearea unei clepsidre digitale care va fi reprezentata de doua matrici 8x8 cu LED-uri. Scopul principal al proiectului este de a crea o reprezentare digitală a clepsidrei clasice, oferind o experiență vizuală atractivă și interactivă. Ideea de bază a pornit de la fascinația pentru clepsidre ca instrumente de măsurare a timpului

Descriere generală



O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Proiectul este realizat cu ajutorul unei placute Arduino Nano ATmega328P. Se poate seta timpul de rulare al clepsidrei cu ajutorul butoanelor. Se va implementa un engine de particule pentru a rula animatia clepsidrei.

Cu ajutorul unui accelerometru vom detecta pozitia in spatiu pentru a determina directia de deplasare a particulelor.

Hardware Design



Piese necesare pentru acest proiect

Arduino Nano ATmega328P

Controlerul principal al proiectului. Arduino Nano va coordona funcționarea tuturor componentelor conectate, inclusiv afișarea pe matricile LED, citirea datelor de la accelerometru și gestionarea intrărilor de la butoane.

MODUL MATRICE LED MAX7219

Afișează “nisipul” clepsidrei digitale. Fiecare modul matrice LED de 8×8 va afișa o jumătate a clepsidrei.

Pinii utilizați:

- VCC: Conectat la alimentare(5v).
- GND: Conectat la masa.
- DI: Pinul pentru transferul datelor conectat la pinul D4 al microcontrolerului.
- CS: Pinul pentru selectarea chipului conectat la pinul D6 al microcontrolerului.
- CLK: Pinul pentru semnalul de ceas conectat la pinul D5 al microcontrolerului.

Butoane

Permite utilizatorului să interacționeze cu clepsidra digitală, de exemplu, pentru a seta timpul de rulare.

Pinii utilizați: Am conectat butoanele la pinii D2, D3 și la GND.

Modul accelerometru și giroscop MPU6050

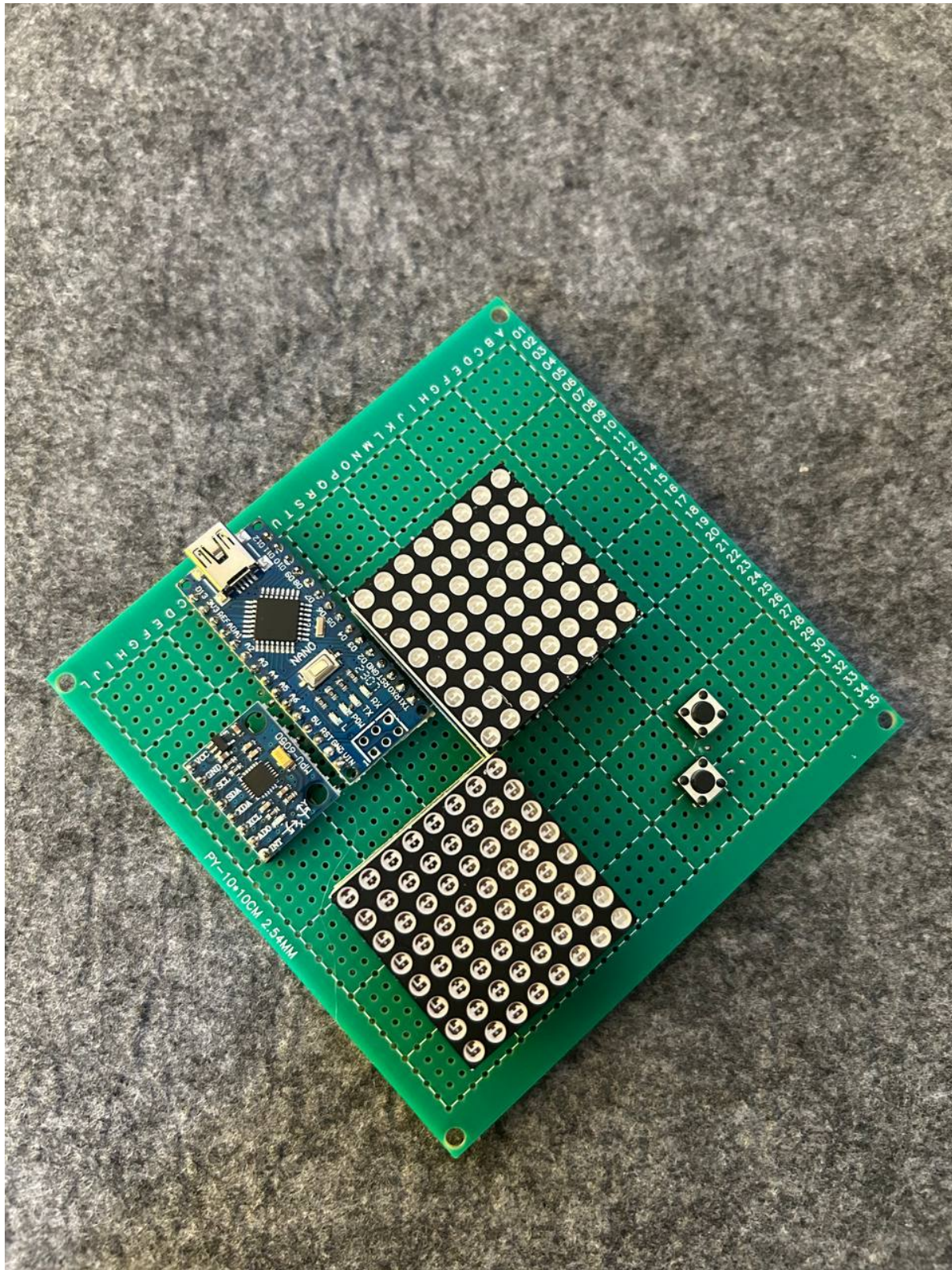
Detectează orientarea și mișcarea dispozitivului. Folosit pentru a simula efectul de întoarcere al clepsidrei.

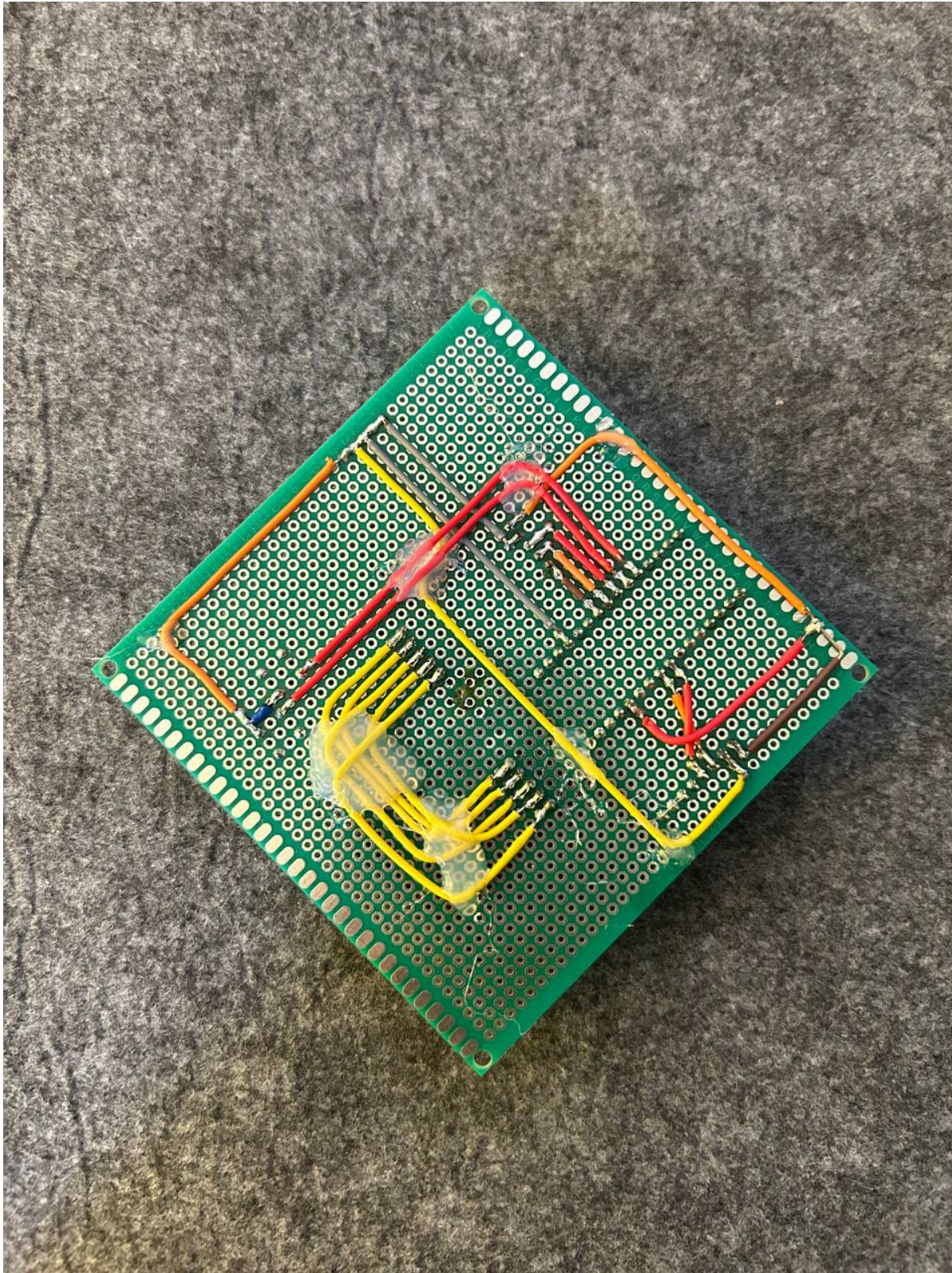
Pinii utilizați:

- VCC și GND: Alimentare (3.3V sau 5V și masă).
- SCL (Serial Clock Line): Conectat la pinul A5 (SCL) al Arduino Nano.
- SDA (Serial Data Line): Conectat la pinul A4 (SDA) al Arduino Nano.

Breadboard

Facilitarea conexiunilor electrice între componente fără a fi nevoie de lipire permanentă. Mi-as dori să printez la o imprimantă 3d o carcasa pentru clepsidra pentru un aspect estetic mai plăcut.





Software Design

Biblioteci folosite

Pentru realizarea proiectului am folosit biblioteci pentru a facilita interactiunea cu accelerometrul si matricile LED.

- **Adafruit_MPU6050 si Adafruit_Sensor**, pentru interactiunea cu accelerometrul MPU6050
- **LedControl**, pentru a controla matricile LED MAX7219
- De asemenea, din cauza faptului ca proiectul este Software heavy,

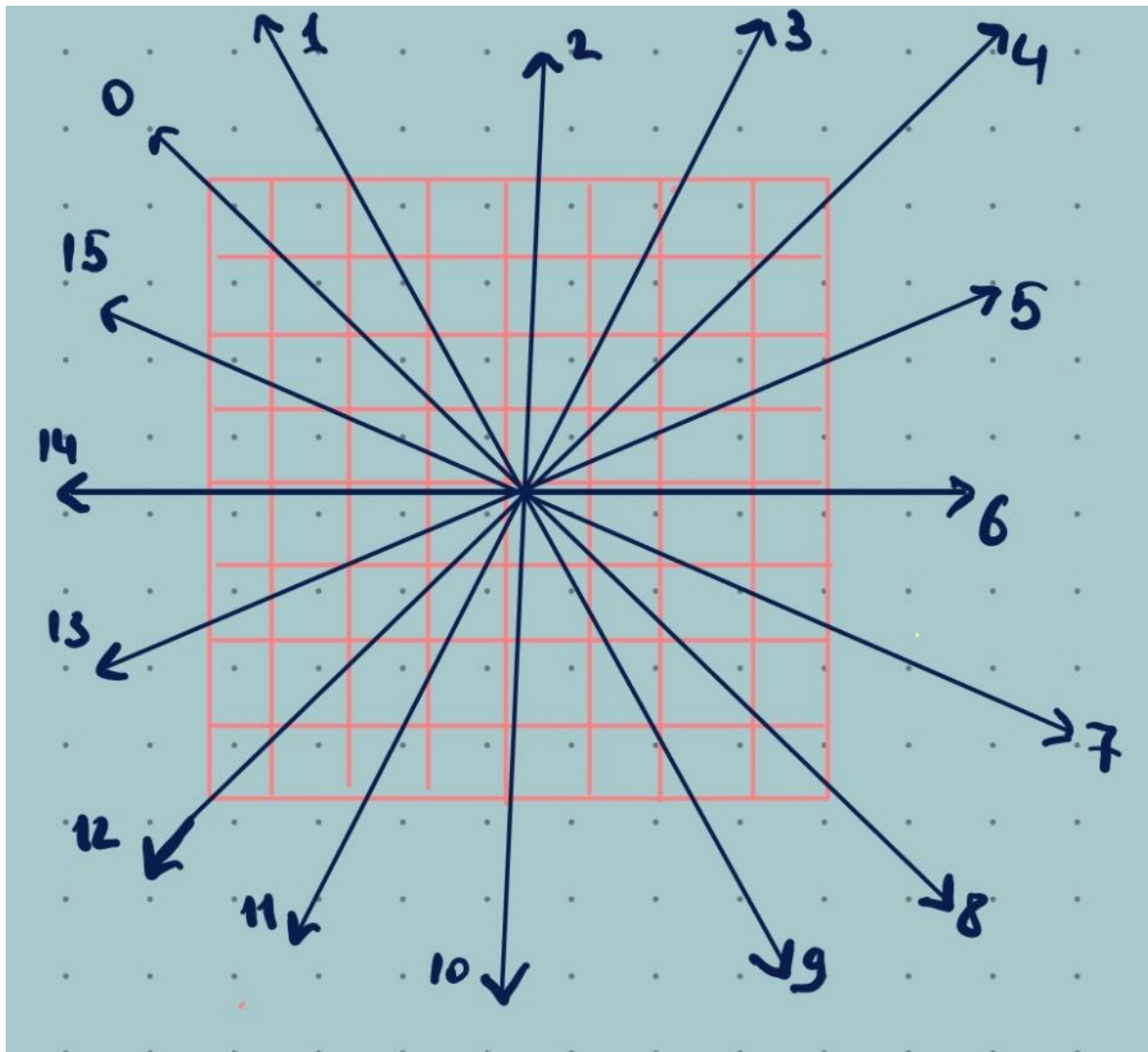
am folosit biblioteci pentru timere(**Timer.h**), interactiunea cu butoanele(*EncButton*), pentru a abstractiza lucrul cu acestea si a scadea din complexitatea codului.

Elementul de noutate. Engine de particule

Am mai vazut video-uri pe internet cu implementari de astfel de clepsidre digitale, insa acestea aveau implementata miscarea particulelor de nisip foarte simplist, care nu semana deloc cu comportamentul nisipului in viata reala. De aceea elementul de noutate al proiectului il constituie engine-ul de particule, care incearca sa replice un comportament mai realist al nisipului digital.

Detalii de implementare

Reprezentarea logica nisipului in programul meu, este realizata printr-un array de bytes, fiecare byte reprezentand o linie din matricea 8x8 al matricei de LED-uri. Ideea de la care am pornit a fost de a detecta directia in care este inclinata clepsidra si de a misca particulele de nisip in directia respectiva, tinand cont de coliziunea dintre particule. Numarul optim de directii pe care le detecteaza accelerometrul este 16. Astfel putem obtine o miscare cat de cat fluida a particulelor.



Pentru asigurare miscarii impredictibile a particulelor, atunci cand o particula poate merge in mai multe celule se alege aleator urmatoarea pozitie.

Functii implementate

- Pentru setarea timpului de scurgere a clepsidrei am folosit doua butoane si am implementat o functie **printDig**, care printeaza orice cifra la o pozitie data in matrice. De asemenea avem functia `changeTime`, pentru a modifica timpul de scurgere a clepsidrei.
- Din cauza ca matricea este de fapt un array de bytes, am implementat niste functii pentru interactiunea cu aceasta: `setXY`, `getXY`, `goDir`, `canGoDir`, `clearMatrix`, `fillMatrix`, `clearMatrix`.
- Pentru interpretarea datelor de pe accelerometru si determinarea directiei de deplasare al particulelor am definit functiile: `getRightDirByAngle`, `getLeftDirByAngle`, `getDownDirByAngle`
- Pentru rularea engine-ului de particule am implementat functiile: `fall`(functia care permite scurgerea nisipului dintr-o matrice in alta), `updateMatrix` si `resetSand`.

Rezultate Obținute

In continuare voi prezenta cum functioneaza clepsidra.

Concluzii

Download

Repository: <https://github.com/alexandru-trofim/arduino-hourglass>

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/prj2024/fstancu/alexandru.trofim>



Last update: **2024/05/27 18:54**