Memory Game

Introducere

Proiectul presupune un joc de memorie. Microcontroller-ul va genera secvente aleatoare de litere pe care o va afisa pe ecran litera cu litera. Utilizatorul are la dispozitie un set de butoane pe care trebuie sa le apese in ordinea corecta pentru a reprezenta secventa. Daca greseste, atunci se va genera un sunet specific si se va scadea o viata. Jocul va avea un singur mod : Endless, care presupune ca jocul tine pana cand jucatorul ramane cu 0 vieti. Pentru modul Endless va exista un sistem de score si highscore si un sistem de nivele in care creste dificultatea in mod liniar : secvente mai lungi, timp mai scurt de raspuns. Jocul va dispune si de un meniu, care va contine cateva setari de baza : exit, play (1 singur mode momentan deci nu e necesar mai mult), time (se afiseaza ora si data pentru un interval de 5 secunde), hscore (se afiseaza highscore-ul pentru un interval de 5 secunde). De asemenea, in spate se vor crea fisiere de logging pentru a putea retine toate interactiunile dintre jucator si calculator : input-uri, output-uri, intreruperi etc.

Scopul acestui joc este de a testa memoria vizuala a jucatorului sau memoria auditiva a acestuia si de a te relaxa dupa o zi foarte grea de munca.

Am pornit de la ideea de a face o reprezentare la pian ale unor soundtrack-uri(MP3 to Piano Version), Ideea mi s-a parut interesanta dar cum o simpla convertire din note muzicale in afisaj pe display proiectul ar fi fost unul 95 % soft si fara nicio interactiune cu utilizatorul / mediul extern, ceea ce nu ar fi fost ideal pentru un proiect de PM, asa ca am ales sa gandesc un proiect care sa combine joaca, dar si care sa aduca putin spre muzica.

Acest proiect nu este util pe domeniul tehnic, dar este util pe domeniul psihologic deoarece te poate scapa de stres, te poate ajuta sa iti testezi memoria si coordonarea creier - maini si multe altele.

Descriere generală

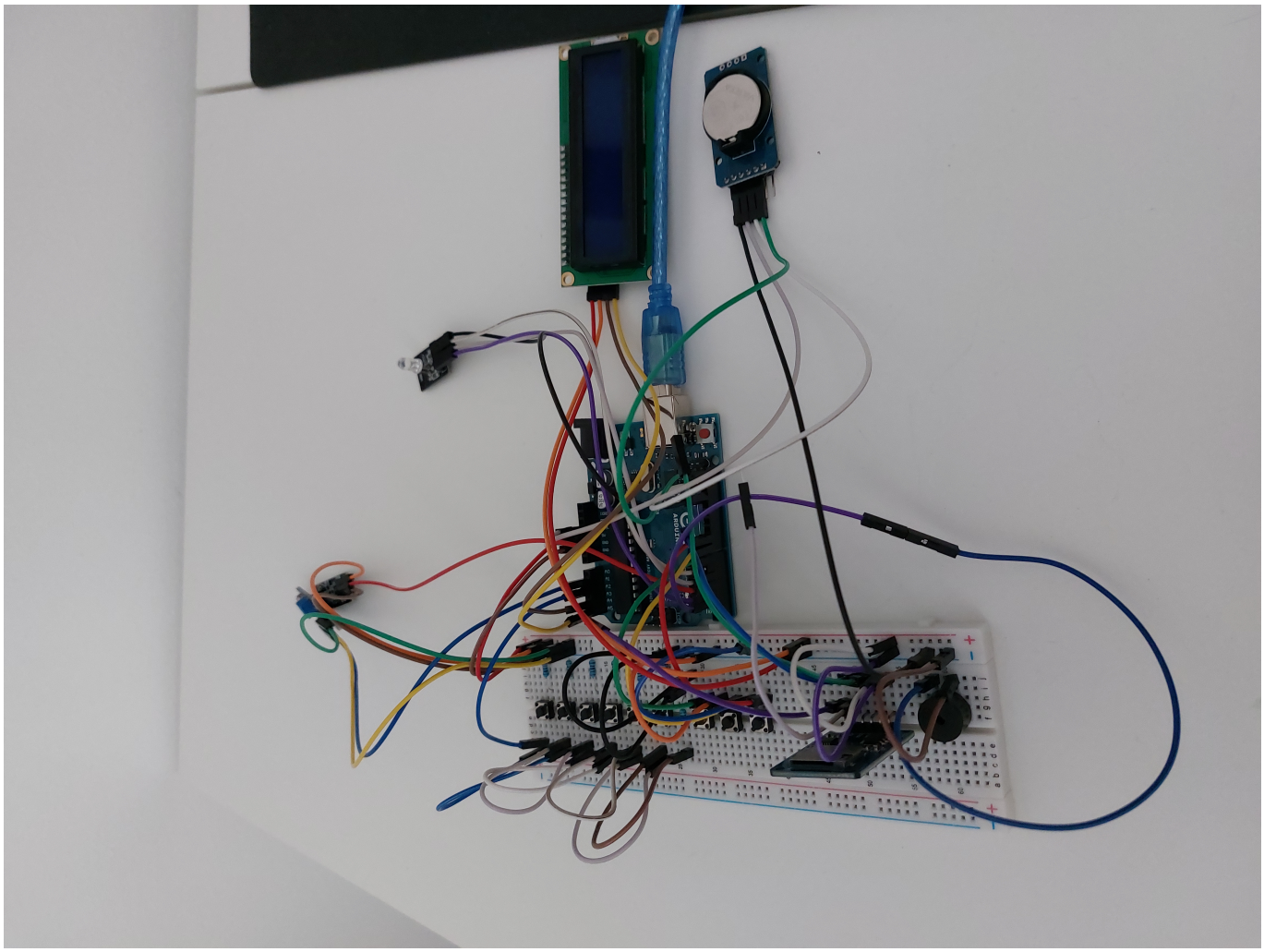
Schema bloc

Jocul ofera o comunicare prin semnal analog intre user si microcontroller, user-ul folosind o mini tastatura formata din 6 butoane. De asemenea, user-ul poate comunica si prin folosirea a 3 butoane conectate digital la Arduino, butoane ce sunt pentru a naviga cu usurinta prin meniul de setari.

Jucatorul poate naviga printr-un meniu simplu cu 4 setari, de unde poate sa si porneasca jocul apasand pe optiunea de “play”. Odata apasata aceasta optiune jocul incepe de la nivelul 0. Microcontroller-ul va genera o secventa de caractere random pe baza unui functii f(x, y) ce primeste ca input valoarea data de senzorul de temperatura si valoarea senzorului de lumina. Vom trata aceasta functie ca un blackbox in spatele jocului deoarece ne dorim ca jocul sa nu redea o secventa

previzibila pentru jucator. Jucatorul trebuie sa tasteze, folosind cele 6 butoane, in ordinea corecta literele generate de Arduino. In caz contrar se scade o viata din totalul de vieti al jucatorului si se repeta secventa generata. Jocul se termina cand player-ul ramane cu 0 vieti, urmand ca acesta sa fie intampinat de un meniu ce ii ofera posibilitatea de a relua jocul sau de a merge inapoi in meniul de setari.

Acesta este proiectul complet in format fizic:



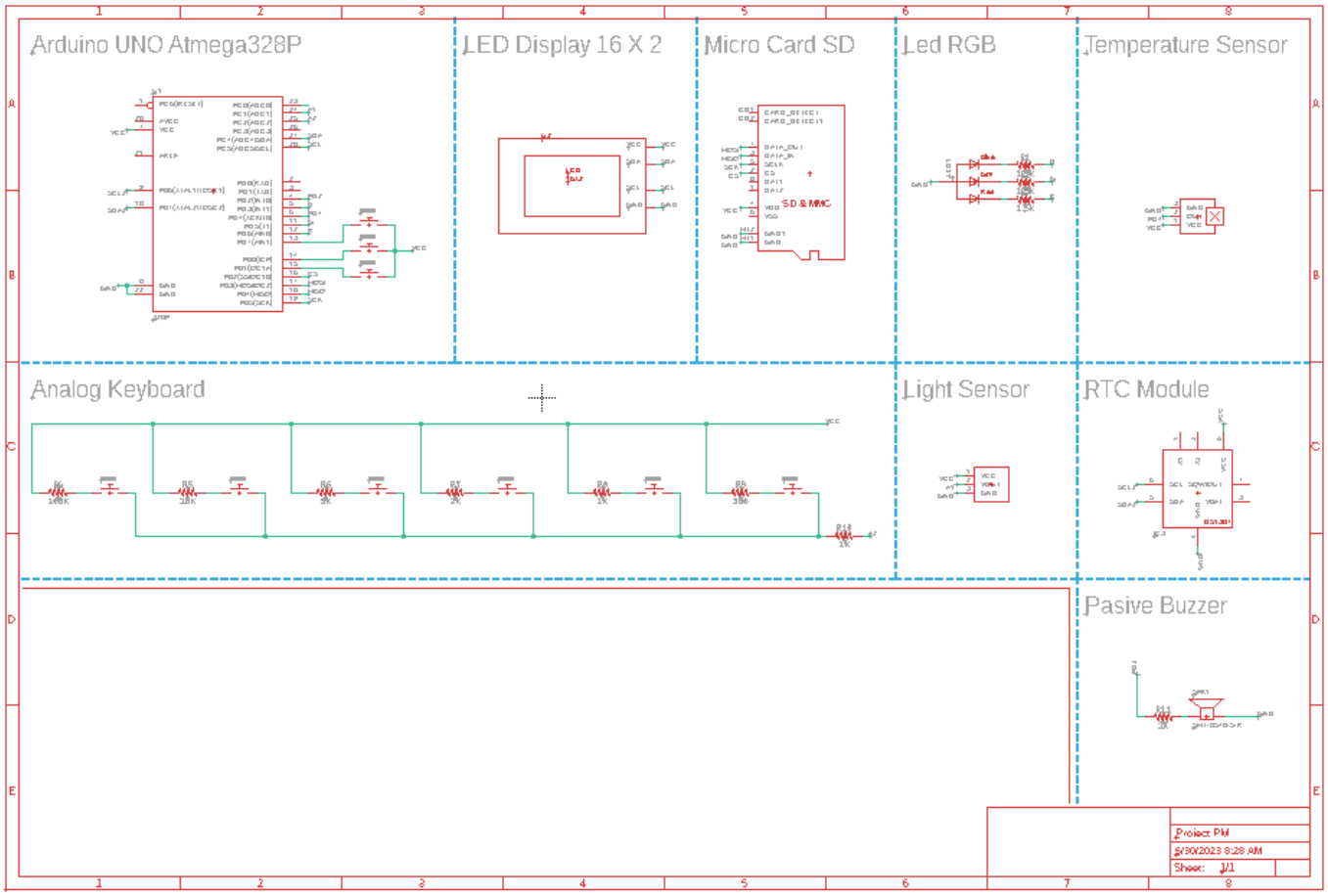
Hardware Design

Lista de piese:

- Arduino UNO ATMega 328P
- Buzzer pasiv Piezzo
- Modul Micro SD
- Modul RTC DS3231
- Display 16×2 LCD
- Modul I2C
- Led RGB KY-016
- Senzor temperatura DS18B20
- Senzor lumina LDR

- Sase butoane conectate analog pentru joc
- Trei butoane conectate digital pentru setari
- Baterie externa
- Baterie Litium 3V
- Rezistente
- Fire
- BreadBoard

Schematic:



Acest schematic este mai mult o forma de prezentare a proiectului. Unele componente sunt doar niste placeholdere la componentele folosite in proiect dar sunt folosite pentru a evidentia legaturile facute cu microcontroller-ul.

Aici evidentiez o rulare a jocului care afiseaza pe seriala cat si pe cardul usb in fisierul de log aferent mesaje de logging:

Software Design

Pentru acest proiect am dedcis sa lucrez cu Arduino IDE. De asemenea, pentru a putea rula jocul cu usurinta tot ce trebuie facut e sa descarcati arhiva cu proiectul, sa includeti libraria in folder-ul

“libraries” din folder-ul Arduino si sa rulati exemplul sugerat de IDE “Run game”. Astfel, pentru a facilita utilizarea si rularea cu usurinta a proiectului codul este structurat in 2 fisiere:

- MemoryGame.h - header ce contine toate functiile si variabilele definite
- MemoryGame.cpp - implementarea functiilor propriu zise

Pentru implementare am folosit urmatoarele biblioteci:

- Wire.h si LiquidCrystal_I2C.h pentru display-ul LCD
- DS3231.h pentru modulul RTC
- DS18B20.h pentru modulul de senzor de temperatura
- SPI.h si SD.h pentru modulul de micro card SD

Pentru a putea organiza mai usor structura proiectului am facut 3 clase majore :

- Time
- GameState
- MemoryGame - clasa de baza a proiectului

Scurta descriere a claselor

- Time - aceasta clasa este folosita pentru a retine date legate de ora si ziua curenta. Aceasta dispune de 3 functii importante:
  1. SetValues() → preia datele oferite de modulul RTC si le stocheaza in campurile aferente clasei.
  2. PrintTimeMessage() → ofera un mesaj sugestiv ce afiseaza ora curenta in format de 24h.
  3. PrintDayMessage() → ofera un mesaj sugestiv ce afiseaza data curenta in care ne aflam.
- GameState - aceasta clasa ofera informatii legate de toate starile posibile prin care poate trece jocul nostru. Aceste stati sunt:
  1. WAITING → starea de sleep a jocului in care nu se intampla nimic inafara de afisarea unui mesaj pe ecran si modificarea intensitatii led-ului RGB
  2. MENU → starea in care jocul se afla in meniul de setari principal.
  3. RUNNING → starea in care se joaca jocul si utilizatorul poate juca.
  4. WAITING → starea in care jocul a generat secventa de litere, iar acesta asteapta ca jucatorul sa reproduca aceasta secventa.
  5. RETRY → starea in care jocul a detectat o litera tastata gresit de catre utilizator, asa ca jocul va repeta secventa.
  6. GAME_OVER → starea in care jocul s-a terminat (jucatorul a ajuns la 0 vieti) si se afiseaza pe ecranul LCD un meniu de sfarsit de joc.
  7. TIME_SHOW → starea in care jocul afiseaza pe LCD ora si data.
  8. SCORE_SHOW → starea in care jocul afiseaza pe LCD scorul maxim obtinut.
- MemoryGame - clasa principala care face ca jocul sa ruleze. Aceasta pune la dispozitie variabile constante definite (pinii folositi de pe arduino), 1 variabila globala folosita ca un contor pentru un timer local si 4 functii, restul functiilor si variabilelor fiind private fate de utilizator:
  1. init() → aceasta functie initializeaza toate structurile si modulele hardware folosite de arduino : seteaza pinii ca fiind de input sau output, seteaza modulele rtc, lcd, seteaza buzzer-ul ,senzorii.
  2. draw() → aceasta functie tine separata logica jocului de partea de afisare pe LCD. Asadar, aici se vor realiza toate afisarile datelor pe LCD in functie de starea in care se afla jocul.

3. update() → aceasta functie se ocupa de modificarea unor variabile interne si stari ale jocului.
4. input() → aceasta functie se ocupa de a prelua input de la utilizator (analog si digital).

Acestea sunt toate functiile si variabilele ce se pot accesa:

Fisierul in care se executa codul principal (“main-ul” din Arduino IDE) este unul foarte simplu avand in vedere organizarea codului:

```
#include <MemoryGame.h>
MemoryGame *game = new MemoryGame();
ISR(TIMER1_COMPA_vect) {
    // Interrupt code
    game->g_sleepPeriod = game->g_sleepPeriod + 1;
    game->setGreenValue(game->getGreenValue() - 30);
}
void setup() {
    game->init();
}
void loop() {
    game->draw();
    game->update();
    game->input();
}
```

Jocul este mai mult un state machine. Asteapta input si pe baza acestuia schimba sau nu starea jocului.

Exista un mic bug de proiectare in aceasta schema a state machine-ului :))). Jocul nu se duce din starea TIME_SHOW in starea RUNNING cand se apasa play, ci se duce din starea MENU in starea RUNNING cand se apasa play.

Dupa cum se poate observa, meniul de setari se acceseaza pur si simplu apasand butonul digital din mijloc. Aici ne putem plimba cu un cursor care selecteaza o setare din cele 4 disponibile : exit, play, time, hscore. Apasarea din nou a butonului digital din mijloc va duce jocul in starea aferenta setarii. Jocul sta doar 5 secunde in starile TIME_SHOW si SCORE_SHOW, iar in starea RUNNING incepe jocul. Aici e mai mult un ciclu de stari care se repeta la nesfarsit sau pana cand jucatorul ramana cu 0 vieti:

- jocul genereaza o secventa de litere bazata pe un algoritm random de switch ce foloseste senzorii de temperatura si de lumina. Bazat pe aceste valori se verifica de cate ori trebuie facut un swap intre 2 litere dintr-un vector de litere prestabilit.
- jocul asteapta input de la utilizator. Daca utilizatorul introduce un caracter gresit, atunci jocul scade o viata jucatorului si repeta secventa generata, altfel jocul trece la urmatorul nivel si genereaza alta secventa de caractere.

- jocul se termina cand jucatorul ajunge la 0 vieti si intreaba utilizatorul daca vrea sa reinceapa sau sa se intoarca in meniul de setari.

Rezultate Obţinute

Jocul functioneaza conform asteptarilor, fara bugg-uri mari la input, dar exista niste probleme pe care nu am stiut cum sa le abordez:

1. memorie limitata, fapt care m-a adus la putine mesaje de log definite
2. biblioteca card sd nu functioneaza corect cand vreau sa rescriu un fisier de la 0, fapt ce m-a dus la renuntarea stocarii scorului maxim intr-un fisier.

Overall, iata un video care arata modul in care functioneaza jocul:

Concluzii

A fost frumos sa lucrez la acest proiect. Am invatat cum comunica componentele intre ele si acesta este si primul proiect fizic pe care il fac. Codul este usor de modificat in caz ca s-ar dori sa se adauge noi moduri de joc si noi functionalitati, dar trebuie optimizat din cauze lipsei de memorie pe care microcotroller-ul o duce (in momentul actual codul ocupa fara sa ruleze in jur de 80% din memorie). As fi dorit sa am un buzzer mai puternic deoarece accesta pe care l-am achizitionat nici in video-ul de prezentare nu se aude, dar singura varianta era un buzzer pasiv deoarece aveam nevoi de sunete diferite in functie de ce tasta analog s-a apasat.

Download

Cod sursa

Schematic proiect

Tot ce trebuie facut e sa descarcati arhiva, sa o importati din Arduino IDE si sa deschideti exemplul oferit de IDE “Game” din libraria “Memory Game”

Jurnal

- 30 mai 2023 - modificare schematic + arhiva zip schematic
- 30 mai 2023 - adaugare specificatii software
- 30 mai 2023 - adaugare README in arhiva
- 30 mai 2023 - update descriere proiect

- 29 mai 2023 - adaugare specificatii hardware, software, concluzii, rezultate, bibliografie
- 29 mai 2023 - modificare introducere + lista componente
- 26 mai 2023 - adaugare schematic proiect
- 7 mai 2023 - adaugare descriere, schema bloc si schematic provizoriu tinkercad
- 6 mai 2023 - creare pagina ocw pentru proiect

Bibliografie/Resurse

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

Resurse hardware:

- https://www.instructables.com/How-to-Multiple-Buttons-on-1-Analog-Pin-Arduino-Tu/

Resurse software:

- https://docs.arduino.cc/built-in-examples/digital/Debounce
- https://arduinogetstarted.com/tutorials/arduino-micro-sd-card
- https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/
- https://arduinogetstarted.com/faq/how-to-use-special-character-on-lcd
- https://docs.arduino.cc/learn/electronics/lcd-displays
- https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023

Export to PDF

From:
http://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2023/gpatru/memory_game

Last update: **2023/05/30 09:13**