

Ioana-Adriana CHIRCA (78566) - Reflex Game

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Acesta este un joc 2-player care testeaza viteza de reactie a jucatorilor, inasa fara a ajunge la confruntari fizice sangeroase, cum se poate intampla la Jungle Speed. Exista 2 controllere, fiecare avand conectate la el un cate un modul Bluetooth, un display pentru starea jocului, o matrice de leduri si o suprafata de apasare. Pe fiecare matrice va aparea cate un pattern de leduri, iar in cazul in care cele doua sunt identice, jucatorul care apasa primul primeste un punct. Daca apasa la patternuri diferite, va suferi o penalizare - aceste informatii pot fi urmarite pe display. Jocul se termina dupa cateva runde.

Descriere generală



Cele doua controllere comunica intre ele prin Bluetooth. Inainte de a incepe jocul, ele trebuie sa se sincronizeze, pentru a putea compara 'timestampuri' de apasare. Unul dintre ele va fi desemnat master la inceput si pe tot parcursul jocului el va determina ce patternuri vor fi afisate pe fiecare dintre matrice.

FLOW

1. Controllerele se sincronizeaza.
2. Este desemnat masterul.
3. Sa zicem ca avem 3 runde. Masterul stabileste 3 perechi de cate doua patternuri (macar o pereche identica) pe care sa le afiseze fiecare la cate x secunde.
4. O runda decurge astfel: Masterul trimite un cod de pattern celuiilalt controller si un timestamp la care sa il afiseze. La acel moment de timp, modelele apar pe fiecare matrice. La apasarea pe master, acesta inregistreaza timestampul. La apasarea pe slave, acesta trimite timestampul la master. Masterul compara timestampurile si determina daca vreun jucator primeste puncte sau penalizari.
5. La terminarea rundelor este afisat castigatorul pe display-uri.

...Acesta a fost ideea initiala :)

Deoarece nu am reusit sa-i dau de cap comunicarii wireless, am simplificat putin abordarea, ajungand sa folosesc un singur microcontroller, care afiseaza pe un display grafic informatii despre ambii jucatori.

Diagrama bloc a proiectului arata acum asa:



Hardware Design

Lista de componente:

- placa de baza
- 2 x matrice leduri + registre de shiftare MAX7219 ([118-max7219-led-dot-matrix-module.html](http://www.tti.com/Products/Maxim-Integrated-Corporation/Max7219-LED-Driver-IC))
- 1 x display grafic 128 x 64 ([9351](#))
- 2 x buton

Conexiuni:

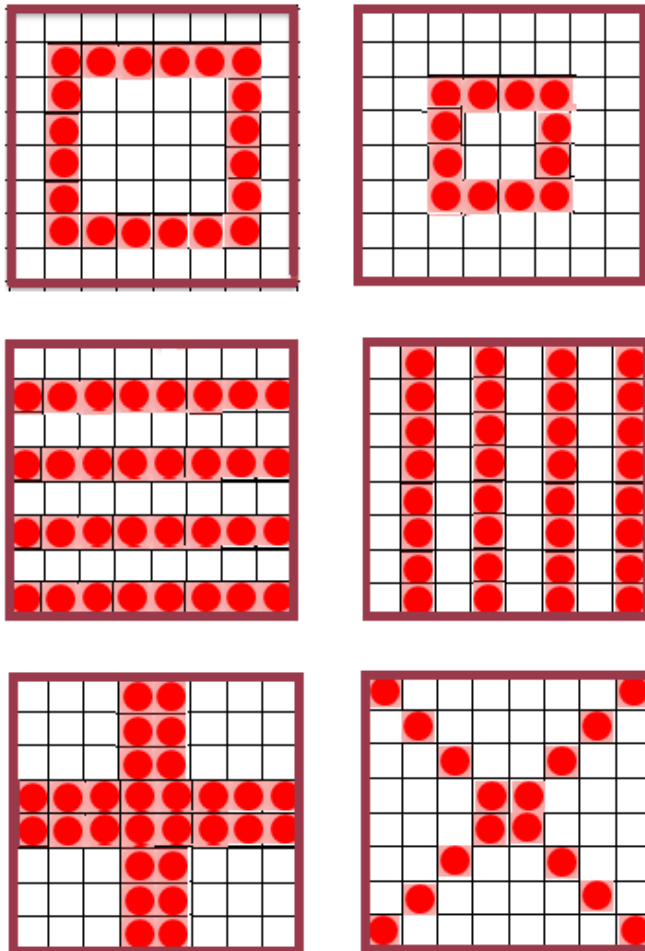
- buton player A: PB0
- buton player B: PA0
- LCD: TX → PD0, RX → PD1
- LED Matrix player A: CLK → PA5, CS → PA6, DIN → PA7
- LED Matrix player B: CLK → PD4, CS → PD5, DIN → PD6

Schema electrică

se intersecteaza (cruce sau diagonala). De asemenea, exista si posibilitatea ca modelul sa fie afisat *inverted* pe matrice (adica sa fie luminate ledurile complementare). Doua patternuri se considera la fel daca apartin aceleasi clase. De asemenea, la fiecare moment e updatat scorul pe display

- Dupa 15 runde (configurabil), jocul se termina si e afisat rezultatul final. Daca butoanele sunt apasate din nou, porneste un alt joc

Cele 3 clase se pot observa in figura: pe primul rand sunt patratele, pe al doilea liniile si pe ultimul



intersectiile.

Observatii legate de cod:

- Pentru a modifica usor numarul de secunde la care se schimba imaginile, am configurat Timer1 la 1Hz si numar datile in care intra in ISR (de exemplu, pentru 4 secunde trebuie sa fi incrementat o variabila de 4 ori in ISR)
- Pentru a obtine randomness la generarea imaginilor si totusi a pastra un numar fix de posibilitati, am definit in images.h un vector de clase de imagini - 3 clase cu cate 2 imagini. Asta inseamna ca trebuie generate clasa, indicele in clasa si proprietatea inverted in mod random. Dupa ce am luat in considerare mai multe optiuni (printre care conectarea unui senzor la ADC), am realizat ca mai simpla este sa pornesc Timer0 si sa folosesc valoarea din TCNT0 pe post de seed. Abordarea functioneaza pentru ca apelez *srand(seed)* dupa ce incepe jocul, adica dupa ce se apasa cele 2 butoane, iar ele nu au cum sa fie apasate in acelasi moment la fiecare joc
- Pentru interfatarea cu matricea de leduri prin registrul de shiftare MAX7219 am modificat un exemplu de cod gasit pe avrfreaks.com si am creat propria biblioteca in max7219.hpp

- LCD-ul grafic e comandat prin USART, iar functiile se gasesc in lcd.h si lcd.c
- Butoanele declanseaza intreruperi, nu fac polling.

Mediu de dezvoltare:

- Mac OS, avr-gcc, make
- La inceput am flash-uit codul cu bootloderul, dar apoi am reusit sa il suprascriu si am scris direct codul cu programatorul AVRISP mkII, folosind:

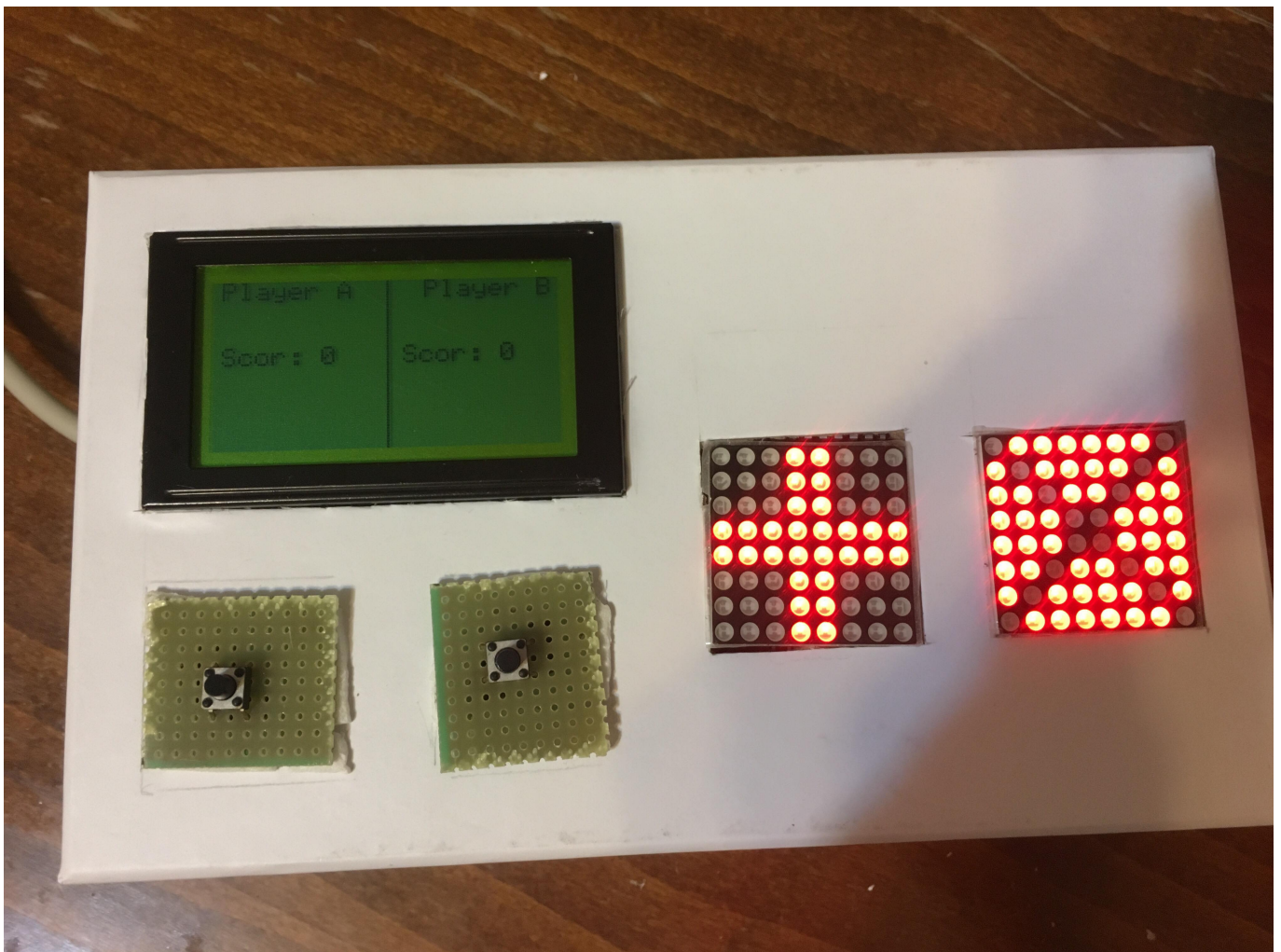
```
avrdude -p m324pa -P usb -c avrispmkII -U flash:w:main.hex -F
```

Rezultate Obținute

Prototip:



Rezultat final:



Concluzii

- Datasheet-ul e sfant si ar trebui verificat pentru fiecare componenta folosita.
- Planificarea atenta de la inceput a componentelor si a caracteristilor lor poate salva *mult* timp (Eu am avut o problema cu obtinerea de 3.3V pentru modulul wireless si am sfarsit prin a lipi un SMD pe o placa de test... not easy)
- Comunicarea wireless s-a dovedit a fi mult mai complicata decat m-am asteptat... dar poate ii voi da de cap in viitorul apropiat!
- Atentie mare la cum conectati din greseala VCC la GND si prajiti componente :)
- All in all, a fost o experienta interesanta - cand scrii cod pentru hardware trebuie sa fii mai ...responsabil

Download

Behold, the code: [reflex_game.zip](#)

Jurnal

- ~aprilie - lipit placa de baza, cumparat componente, schema electrica
- 18-24 mai - scris cod, prajit componente

Bibliografie/Resurse

- Documentația în format [PDF](#)

Resurse Software

- Laboratoare PM
- Datasheet ATmega324 [doc8272.pdf](#)
- Datasheet nrf24l01 [nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf](#)
- [max7219-8x8-dot-matrix-led-driver-attiny](#)
- [MAX7219.html](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2018/rbarbascu/waliays>



Last update: **2021/04/14 15:07**