Documentatie

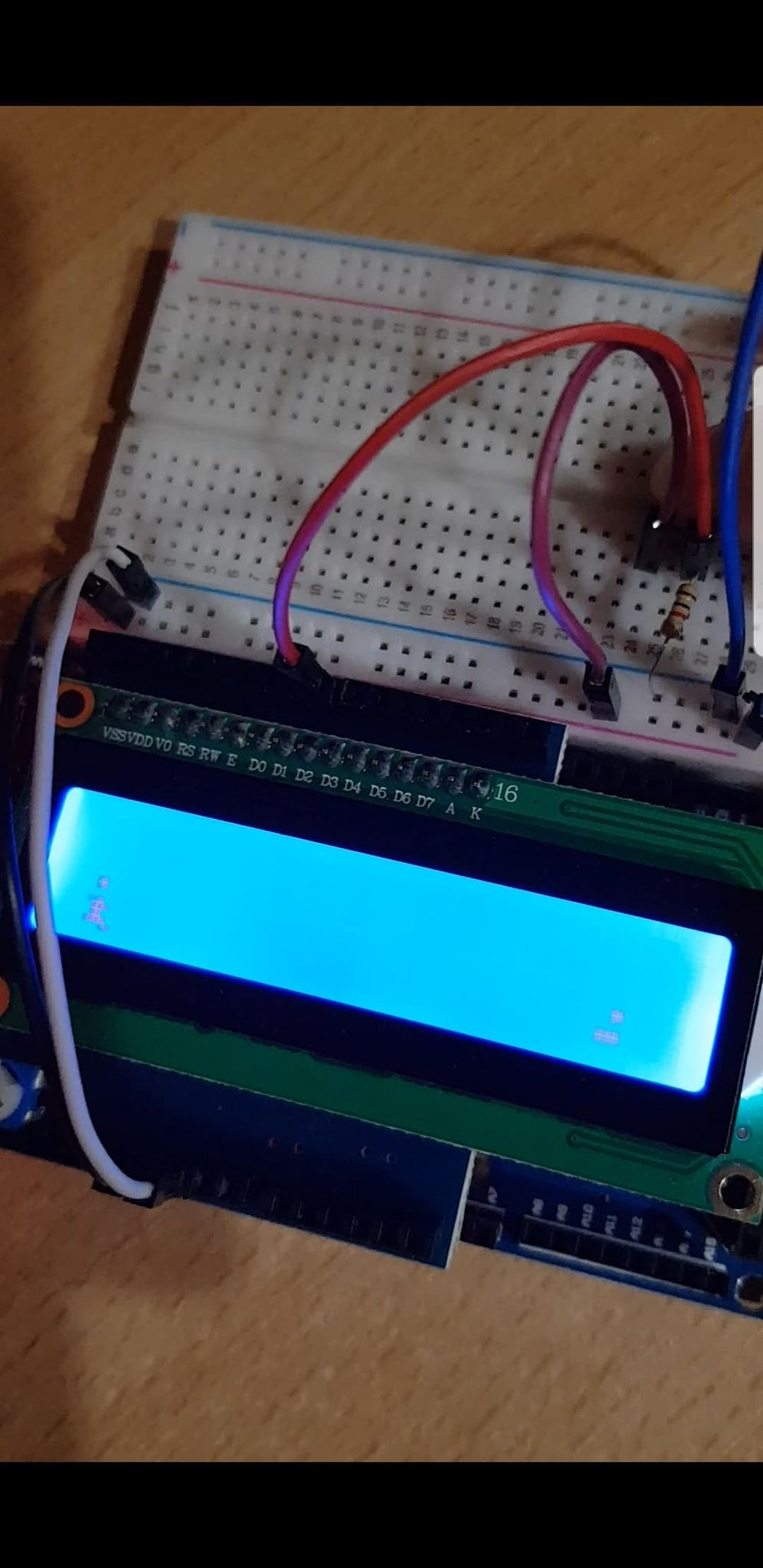
Proiect PMP

Man-Defender(Obstacle)

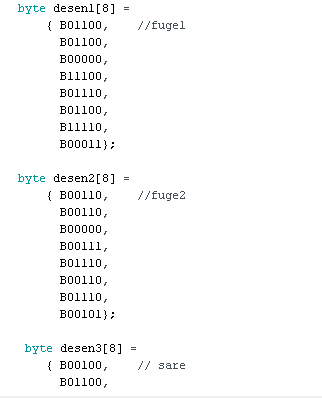
Nemes Daniel – Valentin grupa 30238, an3, CTI

Prof.Coordonator:Mircea Paul Muresan

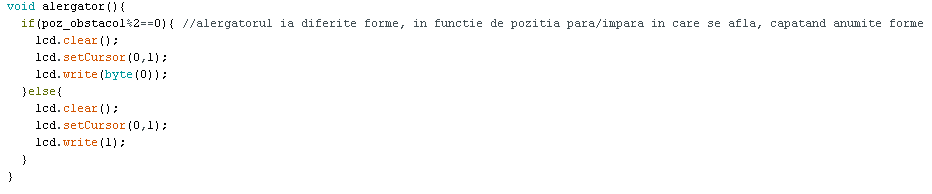
Proiectul meu este un joc prin care trebuie sa il evit pe cel care imi vine in fata mea. Am folosit un LCD 16x2, pentru a face posibil acest lucru. Mai jos se vede alergatorul si aparatorul(obstacolul) in partea dreapta.



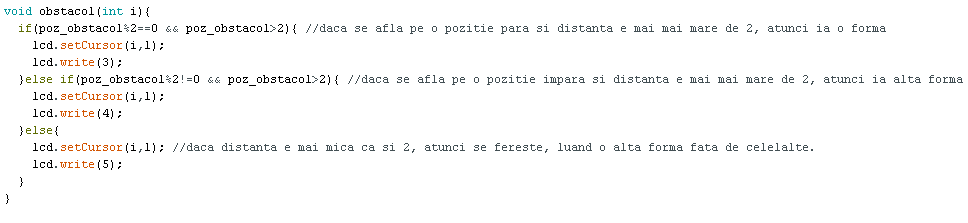
Pentru acest proiect, eu am avut nevoie de un buton, de un LCD ce trebuie impregnat la placa Arduino mega 2560, de un resistor si cateva fire tata-tata, la fel si de un cablu USB.

Am inceput prin a crea cateva matrici de caractere pentru alergatorul meu, aparatorul meu si de pozitia ghemuit al aparatorului atunci cand alergatorul sare peste el. 

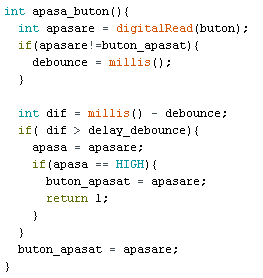
In continuare, am realizat cateva functii pentru alergator, aparator, pentru mesaje initiale, una pentru afisarea scorului si actiunea in sine la apasarea unui buton. Am inceput cu functia start() prin care am afisat mesaje initiale, in care se asteapta apasarea butonului. Am folosit o variabila joc, prin care incep jocul, atunci cand e pe 1. Jocul se bazeaza pe pozitia aparatorului(coloana in care se afla in acel moment). Am gandit in felul urmator: in functie de pozitia para/impara a aparatorului, atat aparatorul cat si alergatorul ia anumite forme, care dau impresia ca alearga. Am creat o functie pentru alergator in care, daca pozitia e para,ia o forma, daca e impara, ia alta forma, dand senzatia ca alearga.



Acelasi lucru am facut si pentru obstacol, cu mici adaugari. Cand pozitia obstacolului e mai mare decat 2, ia formele cum ca ar umbla, iar cand e mai mica decat 2, ia forma de ghemuit.



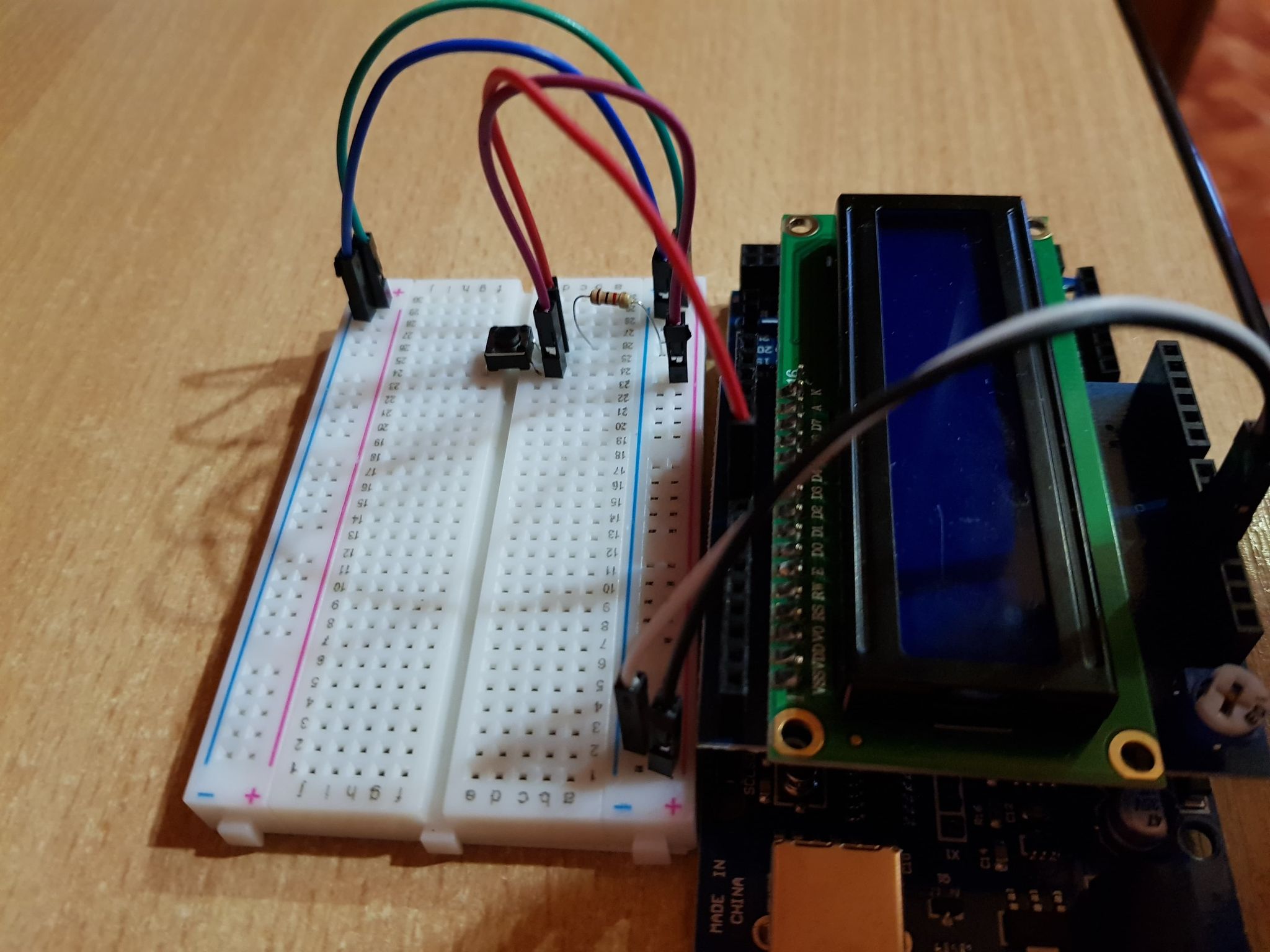
In continuare am facut o functie pentru apasarea butonului. Vom declara o variabila ce va tine cont de buton(low/high). Daca nu e apasat butonul se reseteaza debouncing time-ul. Am initiat delay\_debounce-ul cu 50 in cazul meu, iar daca diferenta este mai mare ca acel numar, vom lua valoarea butonului si daca este apasat va returna 1, adica alergatorul va sari .



Acum va trebui sa ne ocupam de actiunea in sine, si de aplicare a functiilor. Am realizat o functie in care initiez delayul obstacolului la o valoare, pozitia obstacolului va fi pe pozitia 15, in partea dreapta. Cat timp joc este 1, vom afisa alergatorul. Daca apasam butonul, alergatorul va fi afisat pe randul de sus, iar aparatorul va fi afisat tot jos. Altfel, daca nu e apasat butonul, in functie de pozitie para/impara, ni se vor afisa alergatorul si aparatorul in diferite forme. Daca pozitia este 0 , inseamna ca a fost loc de coliziune si se termina jocul. Intre timp, pozitia se tot scade , la fel si delayul cu o anumita valoare, ceea ce va mari viteza jocului. Daca pozitia obstacolului este mai mic ca zero, inseamna ca am ocolit cu succes si scorul creste la fiecare obstacol apasat, pozitia refacandu-se iar la 15. Aici depinde de fiecare cum vrea sa initieze delayul. Poate fi orice numar, dar preferabil e sa avem un numar cat mai mare, si sa se scada de fiecare data cu o anumita valoare.

In functia setup(), am pus matricele de caractere si am retinut butonul.

Mai jos, avem o imagine cu componentele necesare efectuarii cu succes a acestui proiect. Observam ca avem placuta Arduino, la care ii este pus un LCD. Avem un breadboard, in care am facut legatura cu placuta Arduino prin fire. Pe aceasta am pus un buton, iar cu ajutorul firelor, am conectat la pinul 8 al placii. Am mai pus un resistor.



Proiectul functioneaza cu succes, dar nu exact cum voiam eu. Pe video se va vedea faptul ca atat alergatorul cat si aparatorul au mereu aceeasi forma, in sensul ca, in functie de pozitia para/impara a obstacolului, ar trebui sa ia forma diferita, sa dea impresia ca alearga/umbla.