

Pian electric

Autor: Pantelimon Andreea 331CC

Introducere

Proiectul are ca si scop implementarea unui pian care sa aibe doua module. Modulul "freestyle" si modulul "profesor". Acest pian poate fi folositor pentru utilizatorii care doresc sa invete melodii usoare sau care doresc sa aibe o mica jucarie cu care sa se distreze.

Descriere generala

Pianul este implementat astfel incat sa contina doua module pe care utilizatorul le poate selecta prin apasarea unui buton. Primul mod, cel de "freestyle", permite utilizatorului sa cante ceea ce doreste, iar prin apasarea fiecărei dintre clape, un led se va aprinde astfel incat sa semnaleze clapa care a fost apasata. Cel de-al doilea mod, "profesor", va aprinde mai intai un led care sa semnaleze utilizatorului clapa care trebuie apasata. Daca utilizatorul va apasa pe clapa gresita, atunci clapa nu va fi luata in considerare si se va astepta pana cand clapa potrivita a fost apasata. Clapele vor fi realizate din hartie.

Schema bloc:



Hardware Design

Lista de componente:

- Arduino UNO x 1
- Rezistente (10 kohmi) x 5
- Rezistente (1,2 kohmi) x 5
- Butoane x 1
- Speaker x 1
- LED-uri x 5
- Breadboard x 1
- Fire tata-tata x 24

Schema electrica:



Software Design

Pentru implementare am folosit Arduino IDE ca si mediu de dezvoltare. Bibliotecile externe pe care le-am folosit sunt urmatoarele:

- pitches.h (ce contine notele cu frecventele corespunzatoare)
- [CapacitiveSensor.h](#) pentru depistarea capacitatii electrice din corpul uman.

La pornirea pianului, acesta va intra automat in modul de "freestyle", astfel incat utilizatorul poate sa foloseasca clapele cum doreste. Prin apasarea butonului, utilizatorul va intra in modul de "profesor".

Variabile globale:

- b1, b2, b3, b4, b5 - folosite pentru initializarea clapelor de pian;
- lpin1, lpin2, lpin3, lpin4, lpin5 - folosite pentru initializarea led-urilor;
- professor - folosit pentru initializarea butonului de mod "profesor";
- state - in care se va retine statusul butonului (daca este apasat sau nu);
- count - un counter ce ma va ajuta sa parcurg in ordine melodiile memorate din modul "profesor";

Functii:

- keyWait(CapacitiveSensor bpin) - este o functie folosita in modul "profesor". Aceasta opreste melodia si asteapta pana cand utilizatorul va apasa clapa potrivita. Odata ce clapa a fost apasata, se va iesi din functie si melodia va continua;
- Song1(), Song2(), Song3() - aceste functii au ca si rol derularea melodiilor pentru modul "profesor" (sunt trei melodii implementate, fiecarei functii atribuire-se una). In aceste functii, se va aprinde ledul ce indica clapa ce trebuie apasata, se va apela keyWait si o data iesita din aceasta functie, sunetul va fi produs iar ledul va fi inchis, trecandu-se la urmatoarea nota. Am grija sa am un delay astfel incat trecerea de la o nota la alta sa nu fie prea rapida;
- setup() - initializez pinii pentru led-uri si butonul pentru profesor;
- loop() - la inceput setez senzitivitatea pentru fiecare clapa in parte. Dupa care verific daca una din clape a fost atinsa. Daca da, atunci se va aprinde ledul ce va indica clapa atinsa, iar in caz contrar ma asigur ca nu este scos niciun sunet si ca ledurile sunt inchise. Dupa care verific daca butonul de profesor a fost apasat. Cu ajutorul counter-ului, voi verifica de cate ori a fost apasat acest buton. Daca a fost apasat de 0 ori, atunci se va apela Song1() si count-ul creste. Daca a fost apasat o data, atunci se va apela Song2() si count-ul creste. Daca a fost apasat de 2 ori, atunci se va apela Song3() si count-ul va creste. Daca a fost apasat de 3 ori, count-ul se va reseta la 0.

Setul de melodii:

Asa cum am specificat si mai sus, modul "profesor" are implementat 3 melodii. Aceste melodii sunt inregistrate in 4 array-uri: melodyx, noteDurationsx, ledsx, buttonsx, unde x reprezinta numarul melodiei. In aceste array-uri sunt salvate urmatoarele:

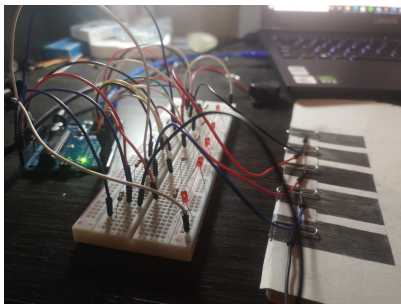
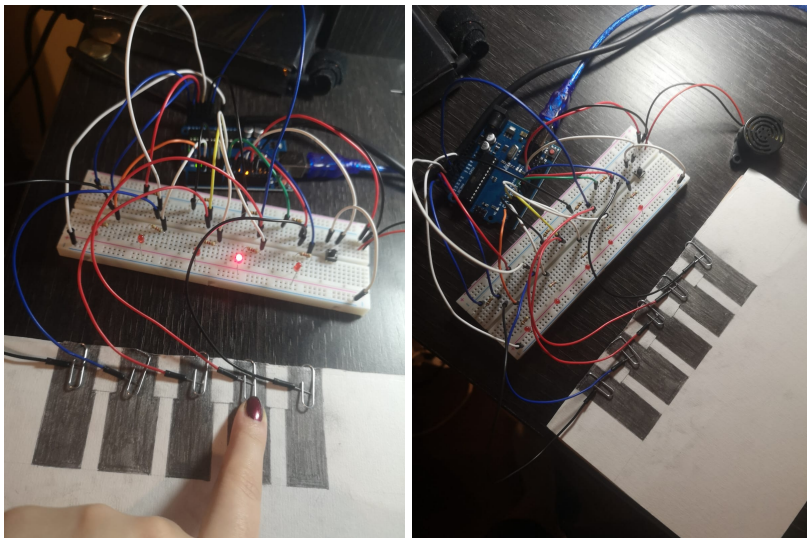
- melodyx: sunt salvate variabile de tip int ce reprezinta notele muzicale (frecventa acestora luata din pitches.h);
- noteDurationsx: sunt salvate variabile de tip int ce vor reprezenta durata unei note;

- leds: sunt salvate variabile de tip int ce reprezinta ce leduri vor fi aprinse pentru fiecare nota
- buttonsx: sunt salvate variabile de tip CapacitiveSensor ce reprezinta ce clape vor trebui apasate;

Rezultate obtinute

Am obtinut un pian electric, a caror clape sunt facute din hartie si prin atingerea acestora sunetul este produs si care are doua module implementate. Una de "freestyle" in care utilizatorul poate sa cante cum doreste si una de "profesor" in care utilizatorul poate invata mai multe melodii.

Overview al proiectului:




Mai jos puteti gasi un mic demo al proiectului:

[DEMO](#)

Concluzii

- In principiu am reusit sa implementez cam tot ce mi-am propus.
- Am invatat multe lucruri interesante, mai ales despre biblioteca CapacitiveSensor si ceea ce ofera aceasta.
- M-am distrat in elaborarea proiectului si simt ca informatia din laboratoare a fost mai bine fixata

odata ce am lucrat la acest proiect.

- Probleme aparute:
 - Mi-as fi dorit ca in loc de led-urile ce arata ce clape sunt apasate sa fi reusit sa introduc o mica banda de led-uri, inasa nu am reusit si am fost nevoita sa ma rezum la led-uri simlpe.
 - Am incercat sa folosesc cati mai putini pini pentru clape, inasa am constat ca programul rula mai incet si detectarea degetului se facea cu intarziere. Prin urmare am folosit mai multi pini astfel incat detectarea sa se faca instantaneu si redarea sunetului sa fie rapida.
 - Din cauza minei de creion, te murdaresti rapid pe degete .

Donwload

Atasez mai jos arhiva cu codul sursa.

[Arhiva](#)

Jurnal

01.05.2021 - Crearea paginii de wiki.

03.05.2021 - Comandarea componentelor.

17.05.2021 - Testarea componentelor.

02.06.2021 - Finalizarea testarii partii hardware si software.

03.05.2021 - Finalizarea paginii de wiki.

Bibliografie / Resurse

[Export PDF](#)

<https://www.arduino-libraries.info/libraries/capacitive-sensor/>

<https://www.arduino.cc/reference/en/libraries/capacitivesensor/>

<https://gist.github.com/mikeputnam/2820675>

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/pian_electric



Last update: **2021/06/04 09:15**