*Arduino Music Station*

Facultatea de Automatică și Calculatoare

An 3, grupa 30236

Student:

*Petrea Irina-Alexandra*

Cuprins:

1. Specificația proiectului

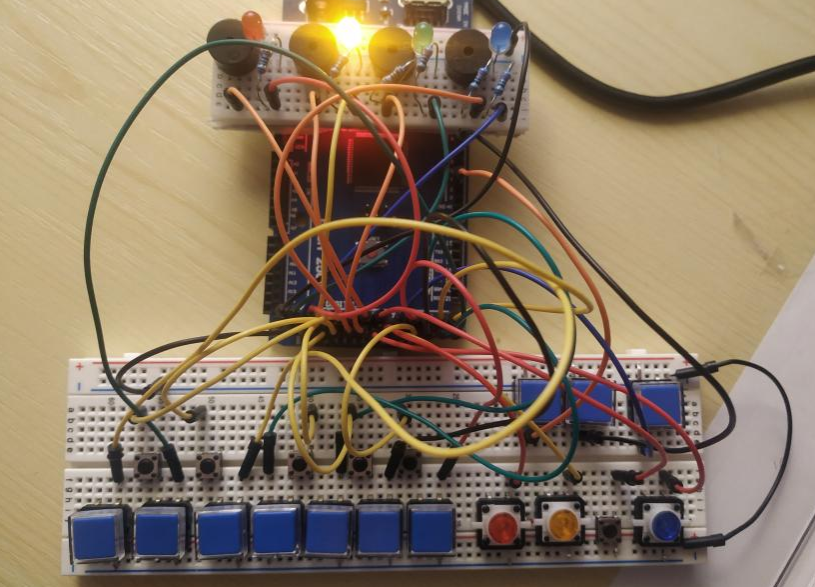
2. Proiectare (Scheme si explicatii).

3. Justificarea soluției alese.

4.Testare

5. Posibilități de dezvoltare ulterioară.   
 6. Referinte bibliografice

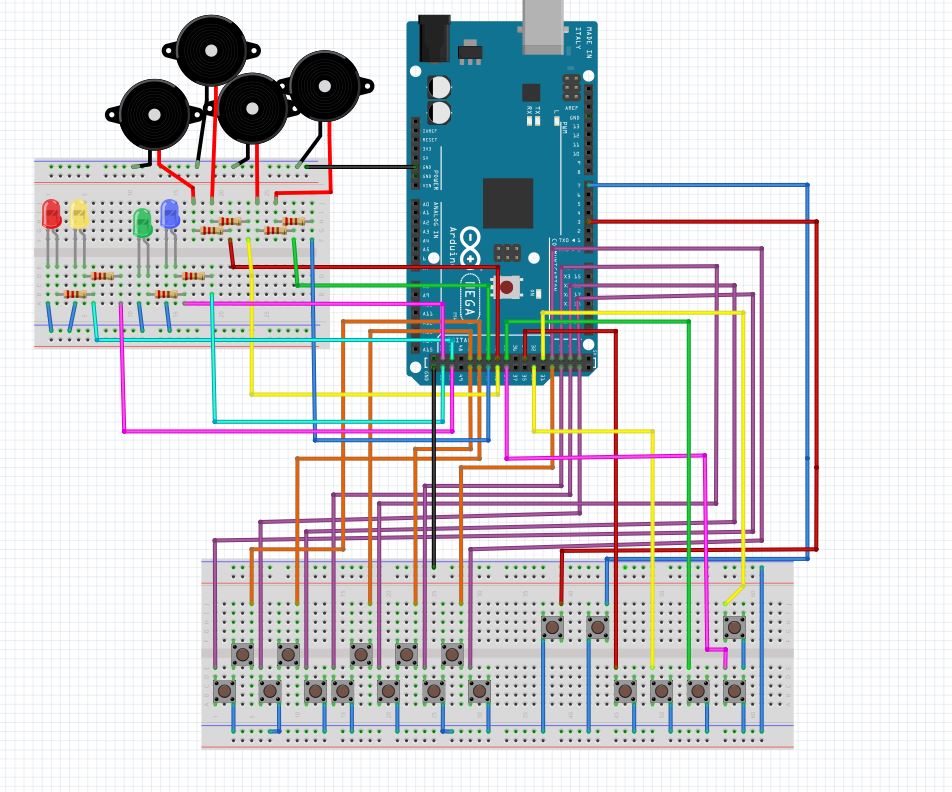
1. Specificatia Proiectului

Arduino Music Station este un proiect muzical, de divertisment. Se poate vedea in imagine faza de prototipozare a proiectului, cu ajutorul placilor de tip breadboard.  
  
  
  
Se foloseste o placa de dezvoltare Arduino Mega2560, precum si diferite butoane, leduri, buzzere piezoelectrice pasive, fire si rezistori.

Se urmareste implementarea urmatoarelor functionalitati:

* redarea de sunete corespunzatoare notelor dintr-o anumita octava – pe o claviatura normala, o octava apare ca o succesiune de 7 clape albe si 5 clape negre.
* schimbarea octavei in care se reda
* schimbarea elementului piezoelectric (buzzer pasiv) pe care se reda la un moment dat melodia
* posibilitatea inregistrarii unei succesiuni de sunete si redarea ei in bucla
* posibilitatea inregistrarii mai multor succesiuni care sa fie redate in acelasi timp pe diferite elemente piezoelectrice.

1. Proiectare

Se poate vedea schema circuitului, realizata cu ajutorul utilitarului Fritzing:  
  
  
Continutul programului Arduino este explicat pe larg in cardul videoclipului de prezentare, dar se vor detalia si aici functiile principale folosite in program, precum si principala biblioteca folosita : Tone.

Principalele functii implementate si folosite in program:

* pin\_config() – este functia care incapsuleaza configurarea tuturor semnalelor, prin legarea lor la pinii digitali ai placii Arduino Mega
* pin\_read() – se incapsuleaza citirea tuturor semnalelor care se vor utiliza in cadrul proiectului
* track\_set() – se face schimbarea pistei de redare/inregistrare prin verificarea starilor butoanelor corespunzatoare. Se aprind ledurile corespunzatoare fiecarul element piezoelectric activ la un moment dat. De asemenea, se seteaza poantorii variabilelor corespunzatoare vectorului de note inregistrate, a indexului in acest vector si a obiectului Tone corespunzator elementului piezoelectric activ.
* octave\_set() – se face schimbarea intre vectorii de frecvente corespunzatoare notelor muzicale ale octavelor. Momentan, aceasta functie modifica octava doar daca se tine apasat (HOLD) butonul aferent.
* process\_rec\_pause() – se face interschimbarea intre modul inregistrare (RECORD) si modul redare si cantec (PAUSED).
  + RECORD: in acest mod de functionare, se tine cont de butoanele apasate, durata apasarii lor, dar si de perioadele de pauza dintre apasari. La trecerea in acest mod de functionare, se initializeaza temporizatoarele corespunzatoare pistelor si se alege pista de inregistrare ca fiind pista curenta (lucru indicat din punct de vedere vizual prin starea de aprins ledului aferent).
  + PAUSED: in acest mod de functionare, nu este decat complementarul modului RECORD, se inregistreaza ultima pauza si se permite redarea sunetelor de pe pistele unde s-a inregistrat deja o melodie.
* process\_keys() – aceasta functie se ocupa de redarea sunetelor la apasarea unui buton. De asemenea, se ocupa si cu inregistarea timpilor de press/release in modul RECORD, spre a adauga aceste valori in vectorul de note corespunzator pistei curente.
  + key\_pressed() – verifica daca a fost apasata una din tastele pentru generare de sunete (clape)
* play() – aceasta functie se ocupa de redarea simultana, cu ajutorul elementelor piezoelectrice, a melodiilor inregistrate pe piste. Acest lucru se realizeaza utilizand o secventa de cod echivalenta cu utilizarea functiei delay(ms). Totusi, foarte important este faptul ca aceasta secventa de cod nu blocheaza programul - nu utilizeaza intreruperile interne -, lucru esential pentru redarea simultana a sunetelor pe buzzere.
  + still\_playing() – functia care verifica daca mai exista vreo pista pe care nu s-a terminat de redat melodia. Nu putem incepe redarea in bucla pe alta pista pana cand nu s-a terminat redarea pe toate celelalte piste.
  + start\_all() – determina inceperea redarii pe toate pistele unde exista o secventa inregistrata.
  + all\_finished() – returneaza valoarea logica TRUE daca s-a finalizat redarea pe toate pistele.

1. Justificarea solutiei alese

Pentru redarea de sunete pe un element piezoelectric, se folosesc, de obicei, functiile tone() si noTone(). Dezavantajul folosirii acestor functii este ca nu putem reda la un moment dat sunete decat pe un singur pin conectat la un element piezoelectric. Acest lucru are loc intrucat functia tone se foloseste de un temporizator intern al placii, care este mereu acelasi.

Pentru a reda sunete pe mai multe elemente piezoelectrice in acelasi timp, am cautat o librarie care sa permita selectarea la nivel AVR a temporizatoarelor folosite. In acest sens, libraria Tone-library [1] ofera functionalitatea necesara.

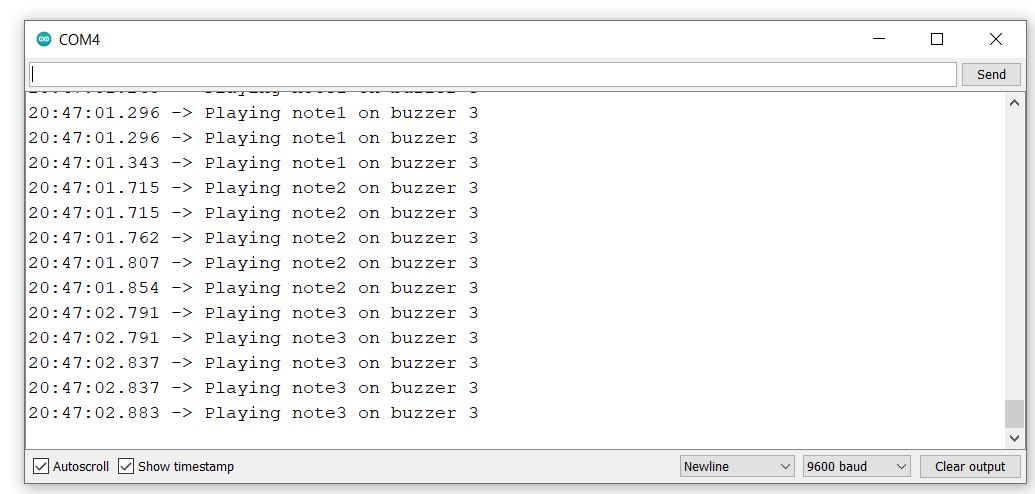
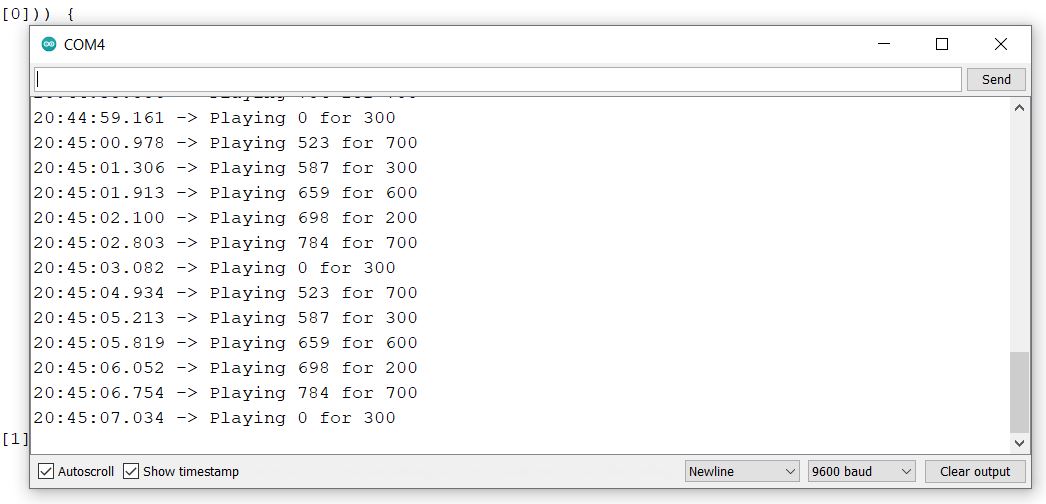
Placa de dezvoltare Arduino Mega2560 pune la dispozitie 6 temporizatoare interne, dintre care unul (cel cu numarul de ordine 0) este rezervat pentru functia PWM si millis(). Deci, se pot reda sunete pe cel mult 5 pini, spre redarea de sunete simultan pe diferite elemente piezoelectrice. Am ales sa folosesc 4 elemente piezoelectrice, pentru a nu utiliza la maximum de capacitate placa de dezvoltare, dar, in acelasi timp, sa mentin un nivel de complexitate relativ ridicat. Am scris codul puternic scalabil, asa incat el se poate extinde la mai multe elemente piezoelectrice si taste functionale corespunzatoare clapelor (de exemplu, o claviatura intreaga).

1. Testare

Testarea si depanarea acestui proiect s-a facut prin interactiune directa cu el si prin interfata seriala (Serial monitor) oferita de mediul de dezvoltare Arduino IDE. Dupa cum s-a prezentat si in videoclipul atasat acestui proiect, functia de inregistrare nu se comporta asa cum ne-am astepta. Pentru a nu ma opri in acest punct al dezvoltarii proiectului, am ales sa implementez un player de test, care sa exemplifice redarea unei melodii compuse din mai multe succesiuni de note pe toate pistele, precum si redarea de sunete prin apasarea butoanelor in timpul redarii secventelor deja inregistrate. Aceste aspecte sunt expuse in amanunt in videoclipul de prezentare.

Am inclus mai multe capturi de ecran din procesul de testare a proiectului:

(notele se redau din vectorii hard-coded in playerul de test)



(functia de record nu functionaza cum ne asteptam)

1. Posilibitati de dezvoltare ulterioara

Placa Arduino Mega nu este special proiectata pentru astfel de proiecte muzicale. Un microcontroller care sa puna la dispozitie mai multe temporizatoare interne ar mari capacitatea de redare.

Utilizarea de difuzoare in loc de elemente piezoelectrice ar mari diversitatea de sunete ce pot fi redate de catre un astfel de proiect.

1. Referinte bibliografice

* biblioteca Tone: <https://github.com/bhagman/Tone>
* biblioteca Tone-Library (folosita): https://github.com/daniel-centore/arduino-tone-library
* echivalent delay(ms): <https://dzone.com/articles/arduino-using-millis-instead-of-delay#:~:text=In%20many%20cases%2C%20it%20might,accurate%2C%20non-blocking%20alternative.&text=Timing%20issues%20are%20often%20present,milliseconds%20specified%20as%20a%20parameter>.
* Datasheet chip placa de dezvoltare: http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\_datasheet.pdf