

Tastatura Control media

E-mail contact: constantin.carp@stud.acs.upb.ro

Introducere

Ce face proiectul?

Proiectul reprezintă o tastatură conectată la laptop care prin intermediul unui buton poate da play/pause la o melodie de pe laptop, cu joystick-ul putem da next / previous song. Pe display v-a fi afișată melodia curentă și volumul actual care va putea fi modificat cu ajutorul unui potențiomtru. Și câteva led-uri.

Scopul proiectului

Implementarea unei tastaturi utile care o voi folosi zilnic.

Ideea de la care am pornit

Am vrut să fac ceva ce o să pot folosi și nu un proiect de care voi uita în scurt timp, plecând de la idea că ascult muzică majoritatea timpului petrecut în fața laptop-ului.

Utilitate

Comoditatea utilizării unui player la îndemână fără a căuta tabul cu muzica sau a deschide mereu aplicația de muzică și volum.

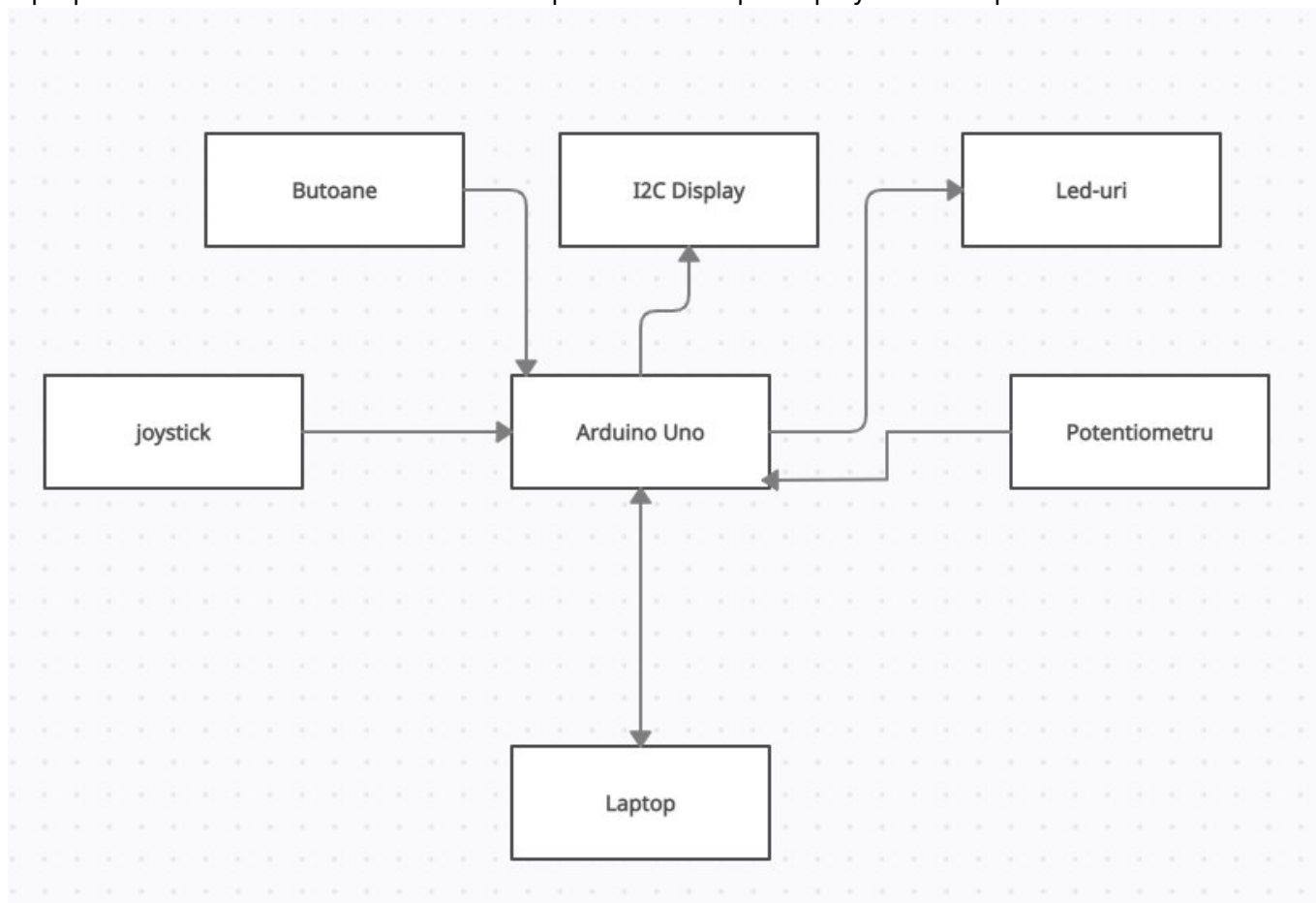
Descriere generala

Descrierea functionalitatii

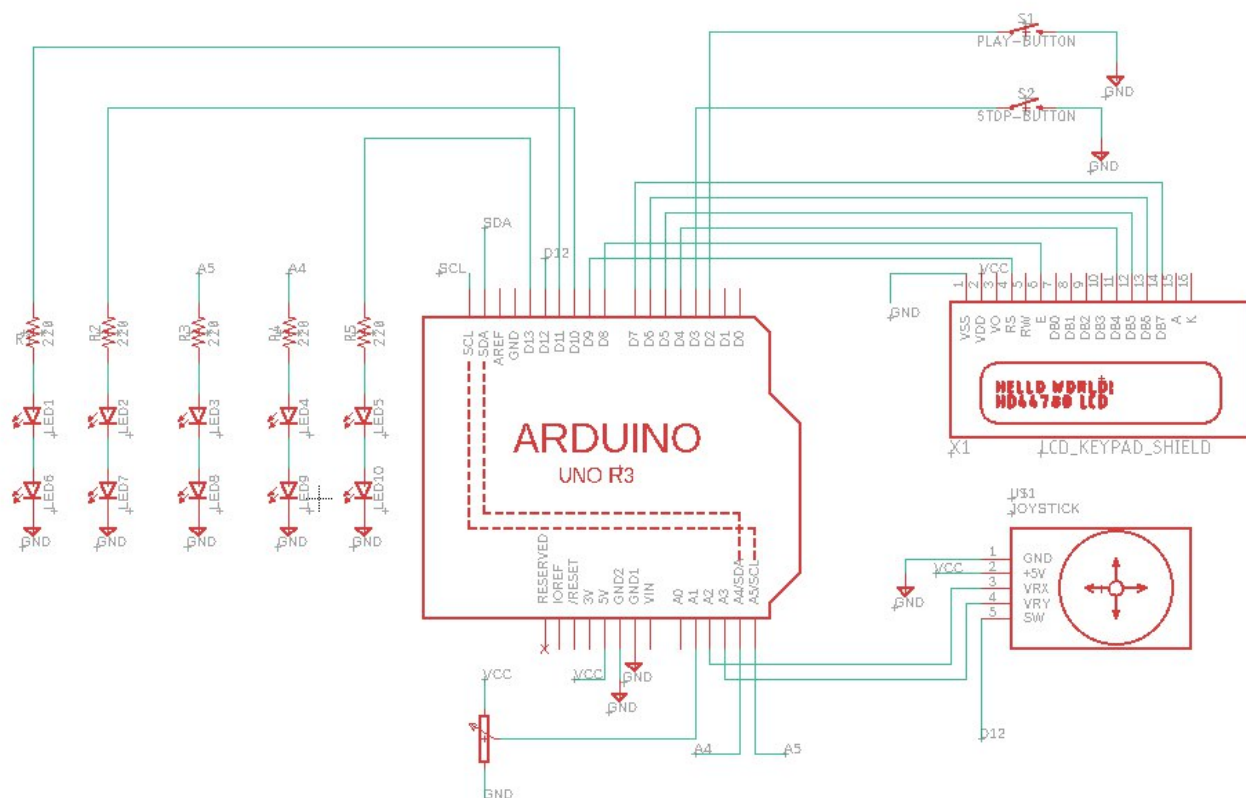
Placa este conectată la laptop, va fi un buton sau 2 pentru play/stop și un joystick care va fi înclinat în stânga pentru comanda previous song și în dreapta pentru next song, pe display în acest moment v-a fi afișat titlul melodiei curente și probabil și volumul sistemului la moment, volumul va putea fi controlat cu un potențiomtru/encoder. Vor fi prezente câteva led-uri sau un ledstrip pentru atmosferă.

Diagrama bloc

Butoanele, joystick-ul și potențiometrul vor trimite date către arduino care le va transmite mai departe pe serială pe laptop, unde vor fi prelucrate și efectuate instrucțiunile necesare, după care de pe laptop vor fi transmise date către arduino pentru a afișa pe display date și a porni led-urile.



Schema electrica



Hardware design

Listă de Componente:

Numar Piese	Nume Piesa	Link
1	Plăcuță Arduino UNO	https://robotica.md/Arduino-uno-CH340G
2	Buton cu capac rotund rosu/alb	https://robotica.md/buton-rotund
1	Potentiometru rotativ	https://robotica.md/potentiometru-rotativ
1	Capac Potentiometru	https://robotica.md/capac-potentiometru
10	LED-uri Mono culoare (2 de fiecare)	https://robotica.md/led-5mm
1	LCD DISPLAY (1602)	https://robotica.md/modul-lcd-cu-butoane
1	Breadboard 830 puncte	https://robotica.md/breadboard-830
10	Rezistente pentru leduri(220ohm)	https://robotica.md/rezistor
2	Fire Dupont mama-tata 10 fire in set	https://robotica.md/10-fire-dupont-mama-tata-20cm
2	Fire Dupont tata-tata 10 fire in set	https://robotica.md/10-fire-dupont-tata-tata-20cm
1	Fire Dupont mama-mama 4p	https://robotica.md/fire-mama-mama-4p
1	Joystick	https://robotica.md/modul-joystick-arduino

Software design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare : ArduinoIDE(1.8.15), IntelliJ PyCharm Community 2021.1.1
- librării și surse 3rd-party (LiquidCristal.h, pyautogui, pyserial, os, nircmd)

Python

Stabiliesc comunicatia pe seriala cu arduino cu ajutorul bibliotecii pyserial, o apasare de buton, rotire potentiometru sau actionare joystick trimite pe seriala un semnal, pe care il citesc in python si il tratez activand windows virtual key corespunzator semnalului cu ajutorul bibliotecii pyautogui. Primul mesaj pe seriala este valoare actuala a potentiometrului pentru a sincroniza cu volumul din windows cu ajutorul nircmd.

Butoanele

Folosec 2 butoane unul cu capac roșu și unul cu capac alb.

- Butonul alb - pause.
- Butonul roșu - resume.

Windows virtual keyboard nu oferă comenzi separate pentru pause și play așa că am făcut eu o separată astfel doar dacă apeși butonul alb se pune pauza și doar dacă apeși butonul roșu face resume, dacă apăsăm de mai multe ori același buton nu se schimbă starea.

JoyStick

Folosesc un joystick și fiecare aspect al său este implementat astfel:

- Înclinare stânga - previous track.
- Înclinare dreapta - next track.
- Înclinare în sus - stop.
- Înclinare în jos - win key.
- Apăsare - mute/unmute.

Folosesc o mapare a rezultatelor citite de pe joystick pentru simplitate.

LCD

Trebuia sa.

- Pe prima linie afiseaza melodia curenta din spotify.
- Pe a doua linie afiseaza ultima comanda executata.

Potentiometru

Folosesc un potentiometru rotativ cu valorile mapate(0,1023,0, 100) pentru a fi 1:1 cu volumul din windows:

- Rotire dreapta - volume up.
- Rotire stanga - volume down.

Led

Folosesc 10 leduri mono culoare, alb, rosu, galben, verde si albastru cate 2 in serie:

- Led-urile albe - indica starea pause.
 - Led rosu - indica starea resume
 - Led-urile galbene - actiune a joystick-ului.
 - Led-urile verzi - indica starea unmuted.
 - Led-urile albastre - indica starea muted.
 - Led rosu - indica nivelul de volum
1. Led-urile albe indica apasarea butonului alb, si raman active pana este apasat butonul rosu.
 2. Led rosu indica starea resume/play la apasarea butonului rosu si raman active pana este apasat butonul alb.
 3. Ledurile galbene efectueaza un efect de blink la orice actiune a joystick-ului.
 4. Ledurile verzi indica starea de unmuted(default), se dezactiveaza/activeaza cand apasam pe joystick.
 5. Ledurile albastre indica starea muted, se activeaza cand apasam pe joystick, si se dezactiveaza la urmatoarea apasare pe joystick sau la rotirea potentiometrului intru-cat volume-up si volume-down ne scot din starea de muted si se reaprinde ledul verde.
 6. Led rosu in functie de pozitia potentiometrului isi schimba luminozitatea indicand astfel volumul.

Rezultate obtinute

In principiu am realizat tot ce mi-am propus mai putin partea de de display pentru ca s-a ars.



Concluzii

Pentru realizarea acestui proiect Arduino Uno nu este cea mai buna alegere, o alegere mai buna ar fi de exemplu Arduino Leonardo sau Arduino micro, pentru ca au USB MCU, ceea ce ar permite tot ce am facut in python direct pe arduino, astfel rezultatul final ar fi un device USB plug&play fara a avea nevoie de software in plus.

De asemenea inainte de a va apuca de proiect asigurati-va din timp ca aveti toate piesele necesare in stock, din lipsa resurselor disponibile (un singur site de arduino), nu am putut obtine un display lcd i2c astfel am fost nevoit sa folosesc ce am putut, un lcd 1602 keypad shield care s-a ars cu success.

Download

https://github.com/numeoriginal/Media_Control_Keyboard

Bibliografie/Resurse

Export în [PDF](#)

https://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/tastatura_multimedia

- <https://forum.arduino.cc/t/debouncing-an-interrupt-trigger/45110/2>
- <https://www.arduino.cc/reference/en/language/functions/math/map/>
- <https://stackoverflow.com/questions/1969240/mapping-a-range-of-values-to-another>
- <https://stackoverflow.com/questions/16509569/python-readline-empty-lines>
- <https://www.arduino.cc/reference/en/language/functions/communication/serial/read/>
- <https://create.arduino.cc/projecthub/ansh2919/serial-communication-between-python-and-arduino-e7cce0>
- <https://stackoverflow.com/questions/65011660/how-can-i-get-the-title-of-the-currently-playing-media-in-windows-10-with-python>
- <https://pyautogui.readthedocs.io/en/latest/keyboard.html#keyboard-keys>
- <https://docs.microsoft.com/en-us/windows/win32/inputdev/virtual-key-codes>
- http://electronoobs.com/eng_arduino_tut132.php
- <https://forum.arduino.cc/t/serial-input-basics-updated/382007/2>
- <https://developer.spotify.com/console/get-users-currently-playing-track/>
- <https://www.deviceplus.com/arduino/the-basics-of-arduino-adjusting-led-brightness/>
- <https://www.nirsoft.net/utils/nircmd.html>

Libraries

1. LiquidCristal.h - <https://github.com/arduino-libraries/LiquidCrystal>
2. pyserial - <https://github.com/pyserial/pyserial>
3. pyautogui - <https://github.com/asweigart/pyautogui>
4. python os

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/tastatura_multimedia



Last update: **2021/06/05 18:08**