Keyboard Piano Arduino - Pedolu Carmen-Petronela 332CAa

Pedolu Carmen-Petronela 332CAa

Introducere

Proiectul presupune implementarea unui pian cu 8 clape si afisaj electronic al notelor muzicale. Fiecare clapa va fi simulata printr-un buton, iar pe afisajul electronic va aparea nota muzicala asociata clapei. Se vor putea reda si melodii de pe un cardSD, daca este apasat un anumit buton.

Scopul proiectului este de a putea canta la pian sau de a asculta melodii de pe card SD.

Ideea mi-a venit la laboratorul 4 in care am vazut cum se poate citi valoarea potentiometrului doar atunci cand se apasa un buton. Astfel mi-a venit ideea de a avea mai multe butoane care redau fiecare alt sunet si nota muzicala sunetului sa o afisez pe un monitor. Pentru a complica un pic ideea m-am gandit sa pot reda melodii de pe un un card MicroSD.

Consider ca acest proiect va fi util atat pentru mine ca voi acumula si lega cunostinte, dar si pentru cei care doresc sa cante la pian sau sa asculte muzica de pe un card MicroSD, fara a fi nevoie de un pian fizic, care este mai costisitor.

Descriere generală

Utilizatorul apasa pe un buton, care este conectat in mod direct la placuta. Aceasta va interpreta ce buton este apasat si va reda sunetul specific butonului si va afisa pe monitor ce sunet se reda. Sunetul va fi redat doar cat timp este butonul apasat. Cand nu este niciun buton apasat, atunci se va afisa doar “Hello world!” pe ecranul LCD.

Pentru a reda muzica, va exista un buton special, care odata apasat va incepe sa cante o melodie de pe cardul SD pana la final. In timpul in care melodia este redata pe monitor se va afisa un mesajul “Play music!” pe monitor si un countdown de la 1000ms in timpul redarii melodiei. Daca timpul expira in locul countdown-ului va scrie “Timer expirat!”.

Daca in timpul in care o melodie este redata, se apasa o tasta a unei clape, atunci aceasta nu va fi luata in considerare pana cand melodia nu se opreste.

Cand nu se reda nimic, pe monitor nu este afisat nimic

Hardware Design

Componentele necesare pentru realizarea proiectului sunt:

- **Modul Slot Card Compatibil cu MicroSD** - pentru a reda melodii
- **LCD 1602 cu Interfata I2C si Backlight Albastru** - pentru a afisa text
- **Modul cu Buzzer activ** - pentru a reda muzica
- **9 Butoane** - 8 butoane vor fi pentru notele muzicale DO RE MI FA SOL LA SI DO si un buton pentru a reda muzica sau a opri muzica de pe cardul SD
- **Plăcuță Arduino Uno**
- **Breadboard**
- **Rezistente**
- **Fire**

Schema electrica

Software Design

Mediu de dezvoltare: Arduino IDE

Librarii:

- SPI.h si SD.h - pentru citirea de pe microSD
- Wire.h si LiquidCrystal_I2C.h - pentru afisarea pe monitorul LCD I2C

Descrierea functiilor:

- void myCallback()- functia e apelata atunci cand are loc o intrerupere.

In aceasta functie se citesc notele muzicale de pe cardul SD si durata fiecarei note, valorile fiind retinute intr-un array de valori integer, iar apoi se seteaza state-ul 9, pentru a stii ca are loc o intrerupere.

```
void myCallback() {
  if (SD.exists("MELODY.TXT") == 0) {
    Serial.println("MELODY.TXT");
  }
  if (SD.exists("DURATI~1.TXT") == 0) {
    Serial.println("DURATI~1.TXT");
  }

  File melody_file = SD.open("MELODY.TXT", FILE_READ);
  nr_elem = 0;
  while (melody_file.available()){
    String nr = melody_file.readStringUntil(',');
    melody[nr_elem] = nr.toInt();
    nr_elem++;
```

```
  }
  melody_file.close();

  File duration_file = SD.open("DURATI~1.TXT", FILE_READ);
  for (int i = 0; i < nr_elem; i++) {
    String nr = duration_file.readStringUntil(',');
    noteDurations[i] = nr.toInt();
  }
  duration_file.close();
  state = 9;
```

- void setup() - setez pinii pentru pentru butoane, initializez ecranul LCD si afisez “Hello world!”, atasez intreruperea pentru butonul de play, initializez cardul SD, setez timer-ul

```
void setup() {
  Serial.begin(9600);
  nr_elem = 0;
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Hello, world!");
  pinMode(pushButton1, INPUT_PULLUP);
  pinMode(pushButton2, INPUT_PULLUP);
  pinMode(pushButton3, INPUT_PULLUP);
  pinMode(pushButton4, INPUT_PULLUP);
  pinMode(pushButton5, INPUT_PULLUP);
  pinMode(pushButton6, INPUT_PULLUP);
  pinMode(pushButton7, INPUT_PULLUP);
  pinMode(pushButton8, INPUT_PULLUP);
  pinMode(pushButton9, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(pushButton9), myCallback, FALLING);

  if (!SD.begin(10)) {
    Serial.println("initialization failed!");
  }

  cli();
  OCR2A = 254;
  TCNT2 = 0;
  TCCR2A |= 0;
  TCCR2B |= 0;
  TCCR2A |= (1<<WGM20) | (1<<WGM21);
  TCCR2B |= (1<<WGM22);
  TIMSK2 |= (1<<TOIE2);

  TCCR2B |= (1<<CS22);
  sei();
}
```

- ISR(TIMER2_OVF_vect) - functia e apelata de timer pentru a decrementa countdown-ul

```
ISR(TIMER2_OVF_vect) {
  if (countdown > 0) {
    countdown--;
  }
}
```

- void loop()

Pentru a sti ce se intampla in cazul celor 10 stari, starile fiind 0 (in care se afiseaza pe ecran “Hello world!” si nu se intampla nimic), de la 1 la 8, daca se apasa un buton al unei clape si se afieaza nota specifica clapei si se reda sunetului asociat clapei, si 9 daca se apasa butonul de play de pe cardul SD am folosit o variabila globala volatile int state.

Verific starea fiecarui buton specific unei clape, daca pentru un buton functia digitalRead intoarce valoarea 0, cat timp aceasta valoare este 0, atunci se va afisa pe monitorul LCD nota specifica butonului si se va reda sunetul specific butonului si state-ul va fi numarul specific butonului.

```
  int buttonState1 = digitalRead(pushButton1);
  while (buttonState1 == 0) {
    if (state != 1) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("nota DO1!");
      state = 1;
    }
    tone(buzzer, 440);
    buttonState1 = digitalRead(pushButton1);
  }
```

La fel se intampla si pentru celelalte 7 butoane, fiind printat alt mesaj si redat alt sunet si setat alt state.

Apoi daca state-ul este 9, se afiseaza “play music!”, si sa se redea melodia salvata. In timpul redarii melodiei se va afisa pe ecranul LCD un countdown de la 1000ms, daca acest timp trece se va afisa “timer expirat!”.

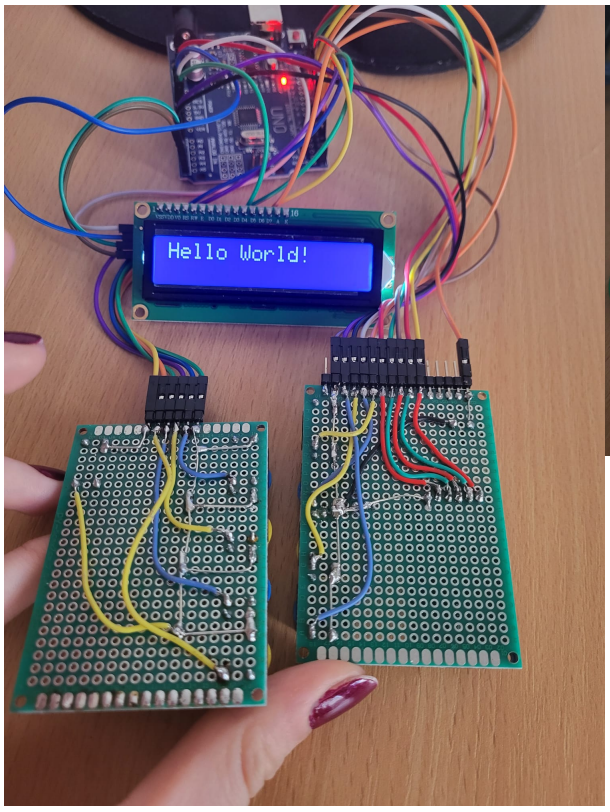
```
if (state == 9) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("play music!");
    countdown = 2000;
    for (int i = 0; i < nr_elem; i++) {
      int noteDuration = 1000 / noteDurations[i];
      tone(buzzer, melody[i], noteDuration);
      int pauseBetweenNotes = noteDuration * 1.30;
      delay(pauseBetweenNotes);
      if (countdown == 0) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("play music!");
        lcd.setCursor(0, 1);
```

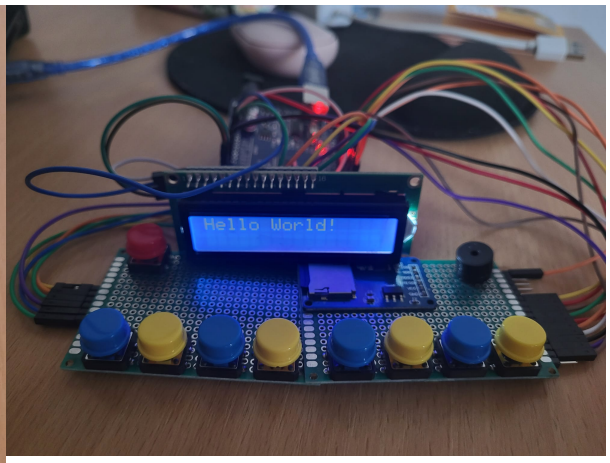
```
        lcd.print("Timer expirat!");
      } else {
        lcd.setCursor(0, 1);
        lcd.print("Time left: ");
        lcd.print(countdown/2);
        lcd.print("ms");
      }
      noTone(buzzer);
    }
  }
```

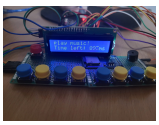
Iar la final daca state-ul este diferit de 0 sunetul este oprit si se afiseaza doar textul “Hello world!” pe ecranul LCD.

```
  if (state != 0) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Hello World!");
    noTone(buzzer);
    state = 0;
  }
```

Rezultate Obţinute







Concluzii

Proiectul mi s-a parut interesant, fascinant si util. Libertatea de a face ceea ce imi doresc m-a atras la inceput. Apoi mi-a placut sa invat sa lipesc componente. Mi s-a parut interesant sa caut in datasheet diverse informatii despre intreruperi, timere si pinii la care trebuie conectati piesele. Consider ca am invatat multe despre lucratul cu arduino si conectarea hardware-ului cu software-ul.

Download

Surse

Jurnal

25.04.2023 - Alegere tema proiect

03.05.2023 - Comanda piese

05.05.2023 - Primire piese si Realizare schema electrica

09.05.2023 - Lipire butoane

12.05.2023 - Lipirea celorlalte componente

22-23.05.2023 - Implementare Software

26-27.05.2023 - Completare documentatie

Bibliografie/Resurse

Melodie card SD: https://docs.arduino.cc/built-in-examples/digital/toneMelody

Datasheet arduino:
https://docs.arduino.cc/static/24a60576d4397e398a1de23a9f2c492e/A000066-datasheet.pdf

Folosire I2C: https://arduinogetstarted.com/tutorials/arduino-lcd-i2c

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2023/vstoica/carmenpedolu

Last update: **2023/05/29 18:18**