Piano Board - Kullman Robert-Alexandru 333CA

Introducere

Acest proiect constă într-un mini pian, cu 8 butoane, ce poate înregistra sau reda melodii de pe un SD Card. De asemenea, există opțiunea de învățare intermediul unor LED-uri care se vor aprinde în funcție de “clapa” care trebuie apăsată.
Pentru a oferi o interfață utilizatorului, un ecran va afișa informații despre statusul board-ului şi despre fişierul curent accesat. Cu ajutorul acestui pian, se pot crea melodii simple sau poate fi considerat şi un mini-player.

Descriere generală

Schema bloc a proiectului:

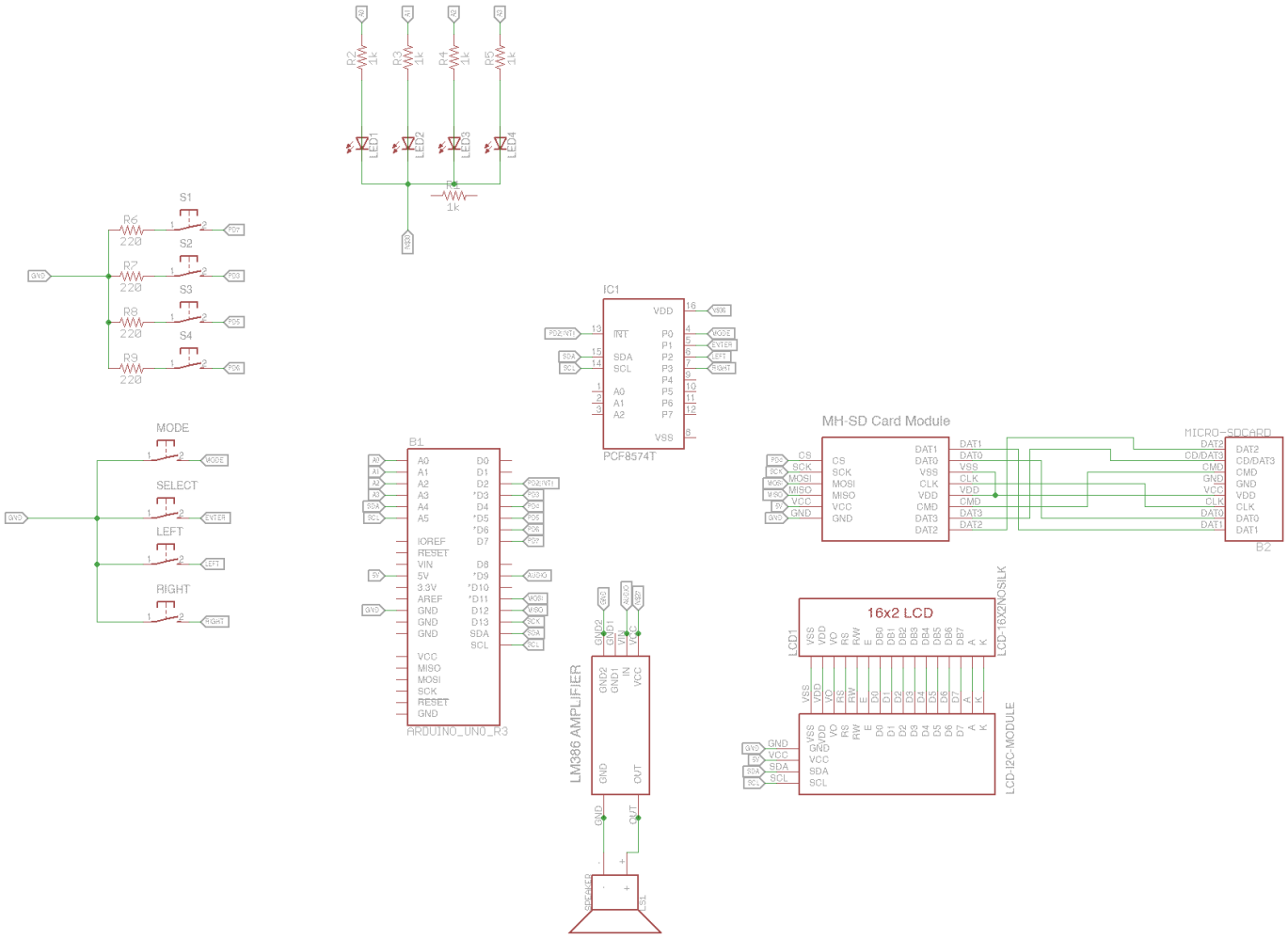
Placuţa de Arduino Uno este conectată la modulul pentru card şi la ecranul LCD. Protocoalele de comnunicare folosite fiind SPI, respectiv I2C. Butoanele vor pune la dispoziţie utilizatorului, în primul rând, posibilitatea de a selecta modul de funcţionare. Acesta poate fi modul de record, modul de player sau modul de teaching. În continuare, butoanele vor simula nişte clape ale unui pian pentru a înregistra anumite sunete.

Hardware Design

Componente folosite:

1. Arduino Uno (ATMega 328)
2. Butoane
3. Ecran LCD 1602 I2C
4. Buzzer
5. Modul Card SD
6. Breadboard
7. Fire de legătură
8. Rezistențe
9. LED-uri

Schema electrica



Functionalitate

Acest piano board este bazat pe 4 moduri de functionare: **FREE**, **RECORDING**, **LEARNING** SI **LISTEN**.

1. **FREE**: aici utilizatorul poate apasa cele 4 butoane rosii, fiecare avand asociata o anumita nota muzicala, neexistand restrictii
2. **RECORDING**: se va crea un nou fisier pe cardul micro SD, iar “clapele” apasate de catre utilizator vor fi retinute, cu o limita superioara de 15 note muzicale. In fisierul text creat, va exista pe fiecare linie cate o nota muzicala
3. **LEARNING**: se va folosi un fisier cu note muzicale de pe cardul microSD. Cate un led se va aprinde, corespunzator cu butonul care trebuie apasat. Dupa ce toate butoanele au fost apasate asa cum e necesar, se va trece din nou in FREE MODE.
4. **LISTEN**: in prezent, utilitatea acestui mod este ca o melodie descarcata in format .wav va fi redata de pe microSD. O functionalitate posibil ulterioara poate fi navigarea si alegerea fisierului prin cardul microSD.

Mod de utlizare

Se pot folosi cele 4 butoane rosii de pe breadbord pentru a imita clapele unui pian. Butonul galben “M” face referire la schimbarea intre modurile piano board-ului. Buton albastru are rol de selectare, de Enter Cele 2 butoane din dreapta au rolul de navigare: stanga-dreapta

Software Design

In functia de setup a programului, se vor apela functii corespunzatoare, ce vor initia LED-uri, butoane, comunicarea cu LCD sau vor activa intreruperi.

1. Butoane - Pentru fiecare buton, se declanseaza o **intrerupere**, in care vor fi setate flag-uri sau se vor face operatiile corespunzatoare apasarii. De asemenea, fiecare buton a fost setat ca **INPUT_PULLUP**. Intreruperile sunt de tipul PCINT:

```
void setup_buttons() {

  pcf8574.pinMode(P0, INPUT_PULLUP, HIGH);
  pcf8574.pinMode(P1, INPUT_PULLUP, HIGH);
  pcf8574.pinMode(P2, INPUT_PULLUP, HIGH);
  pcf8574.pinMode(P3, INPUT_PULLUP, HIGH);

  for (int i = 0; i < 4; i++) {
    // Pin Mode Input Pullup
    DDRD &= ~(1 << buttons[i]);
    // Set default button state to high
    PORTD |= (1 << buttons[i]);
  }
}


void setup_interrupts() {
  cli();

  // configurare intreruperi
  PCICR |= (1 << PCIE2);  // enable the pin change interrupt, set PCIE2 to
enable PCMSK2 scan

  PCMSK2 |= (1 << PCINT19);  // Turns on PCINT19 (PD3)
  PCMSK2 |= (1 << PCINT21);  // Turns on PCINT21 (PD5)
  PCMSK2 |= (1 << PCINT22);  // Turns on PCINT22 (PD6)
  PCMSK2 |= (1 << PCINT23);  // Turns on PCINT23 (PD7)

  sei();
}
```

2. LED-uri

```
void setup_leds() {
  for (int i = 0; i < 4; i++) {
```

```
    pinMode(leds[i], OUTPUT);
    digitalWrite(leds[i], LOW);
  }
}
```

3. LCD

```
void setup_lcd() {
  lcd.begin();      //initialize the lcd
  lcd.backlight();  //open the backlight
}
```

In implementarea piano board-ului, pentru intreruperile pe butoane, se tine cont de anumite flag-uri care ne spun ce mod de functionare este activat. Asadar, intr-o intrerupere, se va verifica ce buton a declansat intreruperea, fiind de tipul PCINT si se vor face operatiile corespunzatoare. In loc sa ne bazam pe un debouncer, deoarece dorim ca sunetul sa fie auzit, indiferent de intervalul la care e apasata clapa, ne bazam pe 2 variabile: buttonState si currentState. Daca acestea indica o schimbare, buttonPressed va fi activat pentru a sti ulterior daca trebuie ca butonul sa realizeze actiunea de care e “responsabil”.

```
  int currentState[4];
  for (int i = 0; i < 4; i++) {
    currentState[i] = PIND & (1 << buttons[i]);
  }

  for (int i = 0; i < 4; i++) {
    if (!currentState[i] && button_1_state == HIGH) {
      button_1_pressed = true;
      timeFromPress = actTime;
      lcdNote = notes[i];
    } else if (currentState[i] && button_1_state == LOW) {  // Check for
rising edge trigger
      button_1_pressed = false;
      timeFromPress = actTime;
    }
```

Functionare GPIO Expander PCF8574T. Pentru a adauga porturile necesare tuturor butoanelor, am avut nevoie de un GPIO Expander. La pin-ul de intrerupere al acestuia, am conectat PIN-ul 2 de pe Arduino UNO. In setup-ul intreruperilor, am avut nevoie sa resetez state-ul tuturor butoanelor la state-ul anterior pentru a evita un bug care facea ca unele butoane sa ramana blocate (sa nu mai fie detectate intreruperi).

```
void resetPCF(bool val0, bool val1, bool val2, bool val3) {
  if (val0 == HIGH) {
    pcf8574.digitalWrite(P0, LOW);
  } else {
    pcf8574.digitalWrite(P0, HIGH);
  }

  if (val1 == HIGH) {
```

```
    pcf8574.digitalWrite(P1, LOW);
  } else {
    pcf8574.digitalWrite(P1, HIGH);
  }

  if (val2 == HIGH) {
    pcf8574.digitalWrite(P2, LOW);
  } else {
    pcf8574.digitalWrite(P2, HIGH);
  }

  if (val3 == HIGH) {
    pcf8574.digitalWrite(P3, LOW);
  } else {
    pcf8574.digitalWrite(P3, HIGH);
  }
}



  if (pcf8574_interrupt) {
    bool val0 = pcf8574.digitalRead(P0);
    bool val1 = pcf8574.digitalRead(P1);
    bool val2 = pcf8574.digitalRead(P2);
    bool val3 = pcf8574.digitalRead(P3);


    if (val0 == LOW) {
      mode_change();
    }

    if (val1 == LOW) {
      enter_press();
    }

    if (val2 == LOW) {
      left_button();
    }

    if (val3 == LOW) {
      right_button();
    }

    resetPCF(val0, val1, val2, val3);
```

Navigare card microSD

Navigarea este posibila in modul de LISTEN, iar pentru a accesa meniul, trebuie ales modul, apoi apasat butonul SELECT/ENTER, iar pe LCD se vor afisa numele fisierelor. La o noua apasare a

butonului ENTER, acea melodie va fi redata cu ajutorul speaker-ului.

Format melodii card microSD

Pentru format-ul melodiilor originale de pe card, s-au folosit:

- extensia: .wav
- bit resolution : 8bit
- audio channels: Mono
- PCM format: U8

Video proiect

Concluzii

Desi nu eram asa de pasionat de partea de hardware si am intalnit multe probleme in timpul dezvoltarii acestui proiect, am lucrat cu placere si mi s-a parut foarte interesant cum poti dezvolta o idee si a o face realitate cu o placuta, niste module si un cod care s-a folosit de aproape toti cei 2K de memorie :)

Jurnal

- 26 aprilie - achiziționare piese de start
- 03 mai- creare pagină documentație
- 26 mai - achizitonare piese suplimentare (difuzor in loc de buzzer, fire difuzor, PCF8574 GPIO Expander)

Bibliografie/Resurse

Codul proiectului se poate gasi aici: piano_board_kullman.zip.

Export to PDF