

Електронска клавијатура за пијано со претходно поставени песни

1. Планирање и анализа

Електронските клавијатури се способни да создадат широк опсег на звуци на инструменти (пијано, оргули, виолина итн.) и тонови на синтисајзер.

Првата клавијатура која работела на струја била креирана во 1876 година, од страна на Elisha Gray кој бил електроинженер во САД. Тој открил дека може звукот да го контролира од самовибрирачко електромагнетно коло, со тоа го измислил осцилатор со една нота. Па така ја создал првата клавијатура која работела на струја и ја нарекол „Musical Telegraph“ поради тоа што осцилациите кои ги креирал овој инструмент се пренесувале преку телефонска линија со помош на електромагнет. Овој инструмент содржел доволно едно тонски осцилатори за да може да се отсвират две октави.

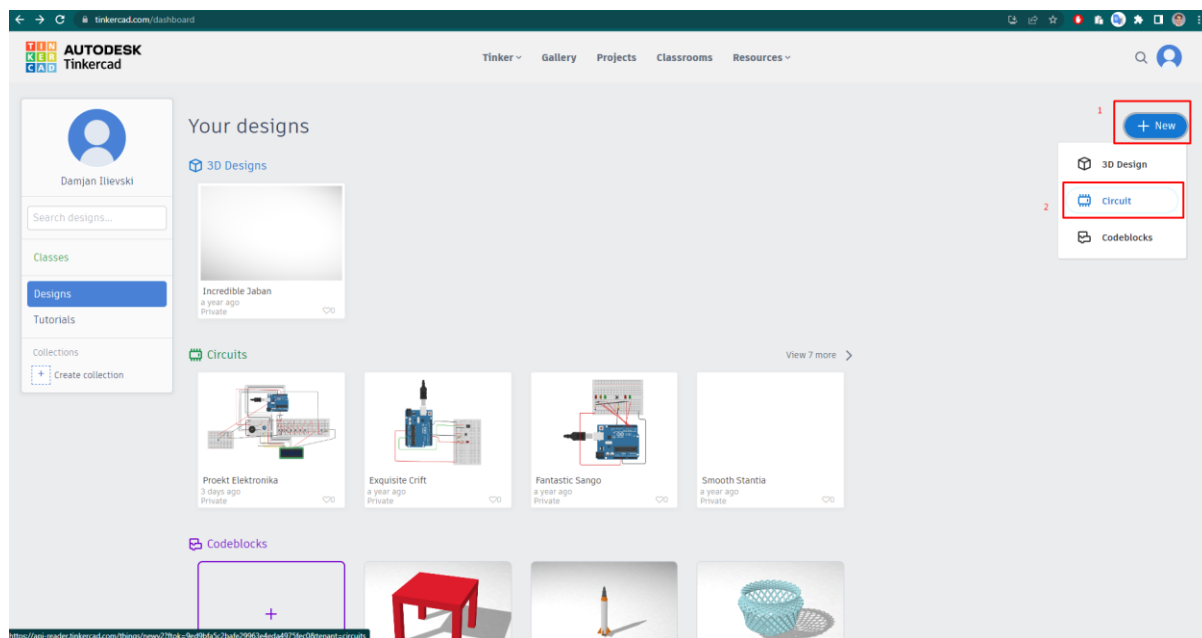
Со напредување на технологијата, така овие инструменти биле унапредувани што дури и во самите инструменти биле вградени песни што инструментот самиот може да ги отсвири. Па така денеска со неколку едноставни електронски компоненти ние самите можеме да креираме еден ваков инструмент.

Побарувања:

1. Копчиња со кои со секое копче ќе се свири една нота
2. Звучник кој ќе произведува звук со притискање на копче
3. Автоматско свирење на претходно дефинирани песни
4. Дисплеј кој ни покажува која песна се свири во моментот

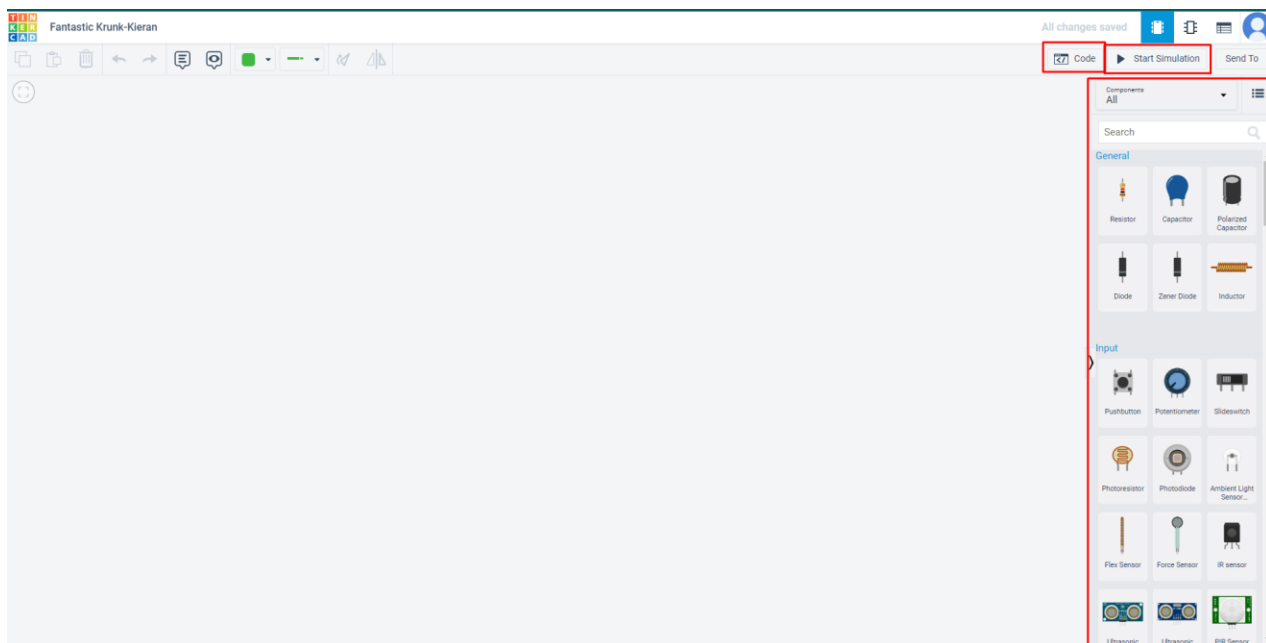
2. Дизајнирање

Овој проект ќе биде реализиран во онлајн симулатор наречен „TinkerCard“. За да се користи овој симулатор потребно е претходно да се креира корисничка сметка. По креирање на корисничката сметка и кога ќе се логирате во горен десен агол постои копче на кое што пишува „New“ откако ќе кликнете избирате Circuit прикажано на сликата подолу.

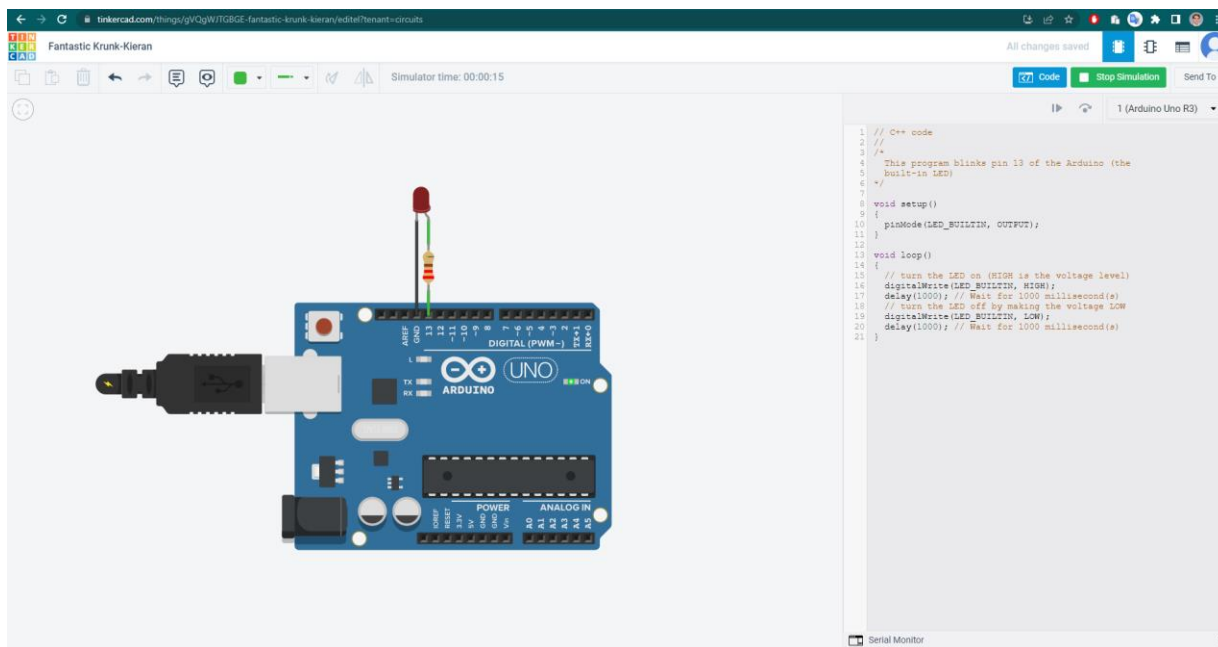


Слика 1. Креирање на нов проект во TinkerCard .

Откако ќе креирате нов проект од десната страна можете да ги видите компонентите кои можете да ги користите, исто така може и да пребарувате доколку ви е потребна некоја специфична компонента. Исто така во горен десен агол можете да видите копче за код едитор наречен „CODE“, тука се пишува код кој подоцна автоматски се внесува во Arduino (или друга програмабилна компонента), и последно потребно копче е копче за започнување на симулација, односно откако ќе ги поврзете сите компоненти и ќе го внесете кодот (доколку е потребен), симулаторот ги симулира компонентите во реално време, и може да има интеракција со компонентите доколку компонентите овозможуваат интеракција (пример копче, потенциометар, сензор за температура, итн.).



Слика 2. Креиран проект, компоненти, копче за кодирање и копче за започнување на симулација.



Слика 3. Пример како изгледа еден проект со започната симулација и кликнато копче за код едитор.

Компоненти кои се користат за овој проект се следните:

Name	Quantity	Component
Uarduino	1	Arduino Uno R3
U1	1	LCD 16 x 2
S1 S2 S3 S4 S5 S6 S7 S8	8	Pushbutton
R1 R2 R4	3	10 k Ω Resistor
R3	1	1 M Ω Resistor
R5	1	4.7 k Ω Resistor
R6	1	560 Ω Resistor
R7	1	1 k Ω Resistor
R8 R9 R10 R11	4	220 Ω Resistor
PIEZ01	1	Piezo
Rpot1	1	10 k Ω Potentiometer
D1	1	Red LED

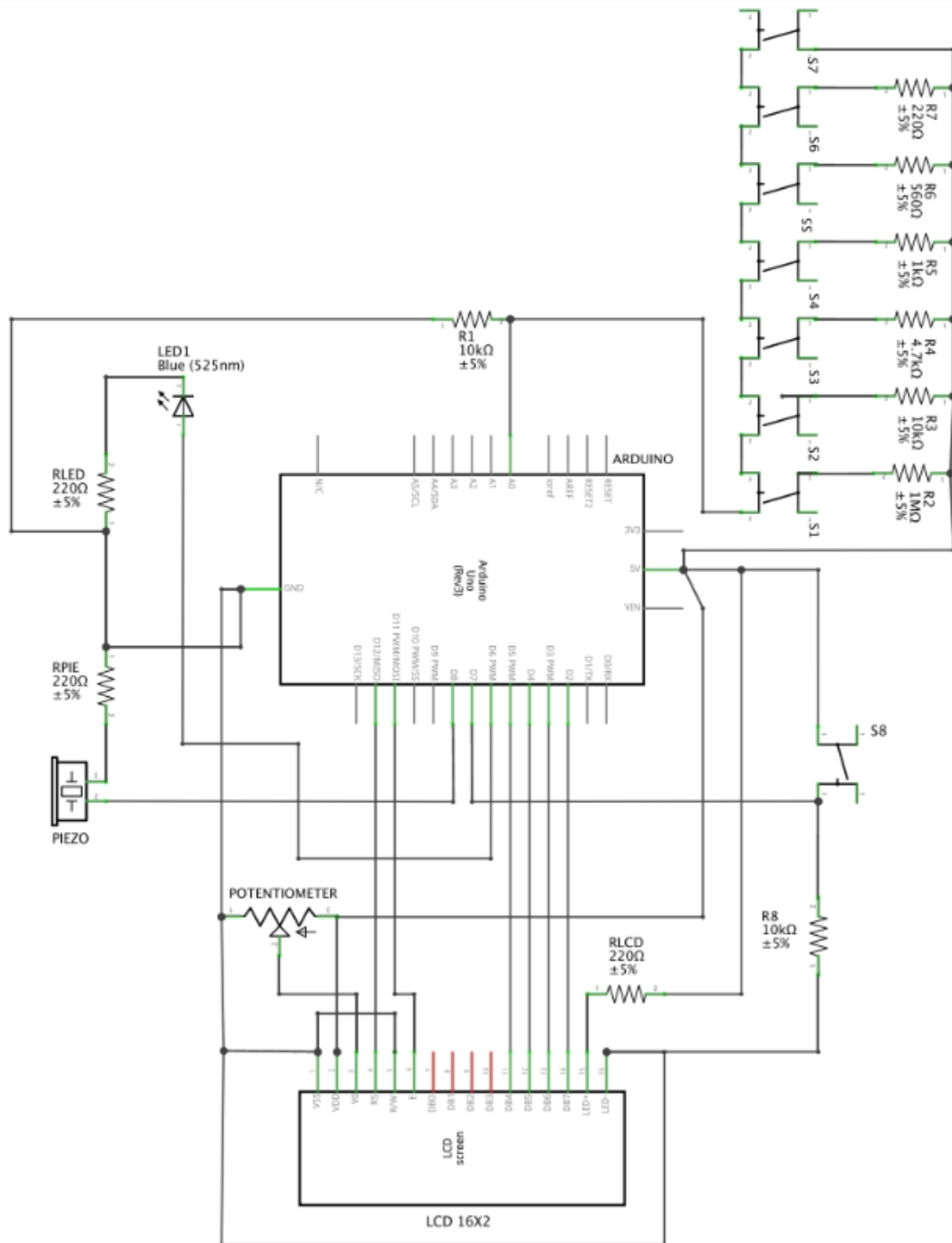
Софтвер кој се користи:

Оперативен систем - Windows 10.

Симулатор - TinkerCard, www.tinkercad.com.

Програмски јазик - C++.

Шема за поврзување на компонентите:



Слика 4. Шема за поврзување на компонентите

3. Имплементација

3.1 Поврзување на компонентите

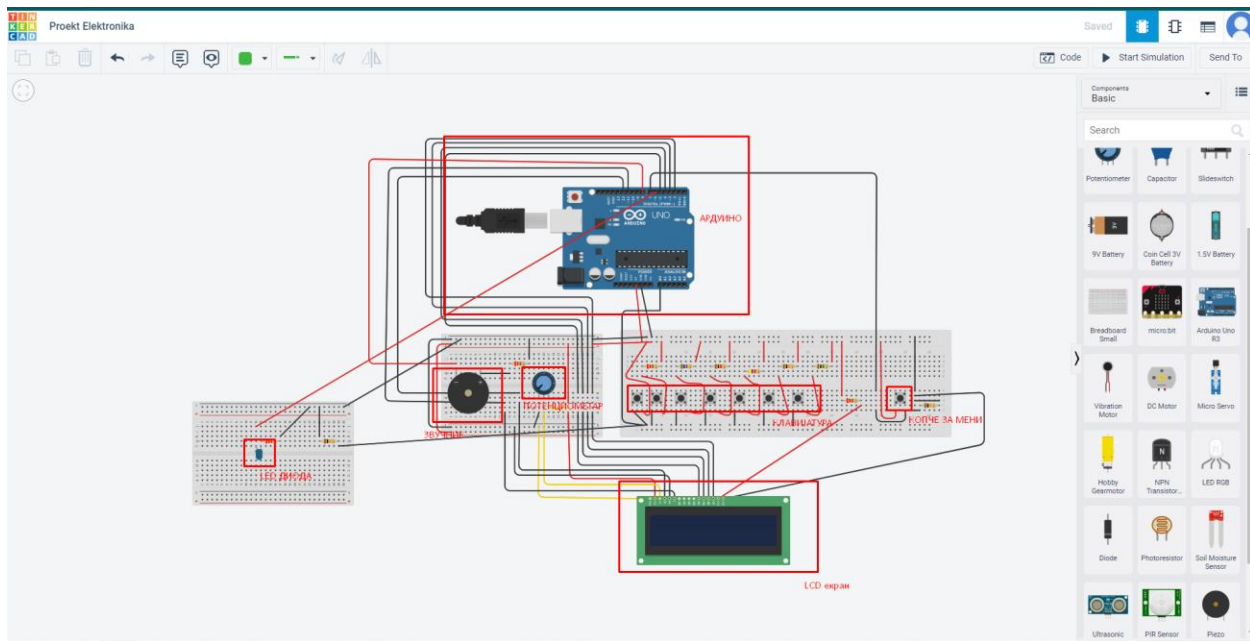
Копчињата на пиано - означени во шемата од S1 до S7, тие се поврзани заедно со отпорниците. Отпорниците се во растечки редослед (220 оми, 560 оми, 1 кило ом , 4.7 кило оми, 10 кило оми, 1 мега ом) и се поврзани во секвенцијален редослед поврзувајќи го секое копче на напојување. Првото копче е поврзано на 5V напојување од едната страна од другата е поврзано со друго копче, второто копче е поврзано со најмалиот отпорник и секое наредно копче е поврзано со следен отпорник кој е најмал од останатите. Последното копче е поврзано на 10 кило оми отпорник и ги поврзува сите останати копчиња на аналогниот пин на Ардуино A0.

LCD екран - пиновите VSS, R/W - се поврзани на заземјување, пинот V0 е поврзан на потенциометар за контрола на јачина на светлоста на екранот, пинот RS - е поврзан на дигитален пин 12 на Ардуино, пинот E е поврзан на дигиталниот пин 11 на Ардуино, пиновите од DB7 до DB4 се поврзани на дигиталните порти на Ардуино од D2 до D5 соодветно и пиновите LED + е поврзан на напојување од 5V, а LED - на заземјување.

Копчето за мени за песни - означено со S8 е поврзано на напојување од 5V и од другата страна на заземјување преку 10 кило оми отпорник, и исто така е поврзано до дигиталниот пин 7 на Ардуино.

LED диода - која ни означува кога сме во „мени режим“ и дека кога сме во овој режим на работа тогаш копчињата на пиано се оневозможени за користење. Оваа диода се вклучува кога ќе се притисне копчето за мени. Анодата е поврзана на дигитален пин 6 на Ардуино и на заземјување преку 220 оми отпорник.

Piezo звучник - поврзан на дигиталниот пин 8 на Ардуино и од другата страна на заземјување преку 220 оми отпорник. Функцијата „tone“ во кодот ќе го бара овој пин за свирење на нотите.



Слика 5. Поврзани компоненти

3.2 Код

3.2.1 Подесување

- Кодот започнува со „импортирање“ на библиотеката „LiquidCrystal.h“, која служи за контрола на LCD екранот.
- Следно се дефинирани имиња на влезно/излезни пинови и дефинирање на ноти така што за секоја фреквенција се дефинира име на нота.
- Потоа се дефинирани песните во посебни класи
- И дефинирани се фреквенциите на пијано копчињата

3.2.2 Главниот циклус (main loop)

- Прв пат кога ќе се вклучи клавијатурата LED диодата е исклучена и сите копчиња за пијаното се активни и може да се свири со нив
- Откако циклусот ќе детектира дека копчето за мени било притиснато ќе ги оневозможи копчињата за пијано и LED диодата ќе се вклучи со што ни укажува дека сме во „мени режим“.

3.2.3 Навигација во „мени режим“

- Откако сме во „мени режим“ на работа со притискање на копчето за мени повторно се движиме низ песните, и LCD екран ни покажува моментално на која песна сме, откако ќе избереме песна со притискање на било кое копче за клавијатура звучникот ќе започне со свирење на песната која е прикажана на екранот.

- По завршување со свирењето на песната, LCD екранот повторно ја покажува истата песна и во тој момент можеме пак да кликнеме на копчињата за клавијатура за потврдоно да се отсвири песната или пак да продолжиме со листање на песните.
- Двоен клик на копчето за мени ќе го рестартира менито од почеток.
- За да излеземе од овој режим на работа треба да ги излистаме сите песни и екранот и диодата ќе се исклучат со што се укажува дека може да ги користиме копчињата за пијано.

Детално објаснување на кодот е наведено во коментари на самиот код.

КОД:

```
// LCD Screen, Switch & LED подесувања
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int LCDswitchPin = 7; // Дефинирање на кој пин е switch-от за мени за песни
int LCDswitchState = 0; // Дефинирање на состојба switch-от за мени за песни
int prevLCDswitchState = 0; //Предходна состојба на switch-от за мени за песни
int blueLED = 6; //Пин на сината лед диода
int LCDswitchCounter = 0; //Бројач кој брои колку пати е притиснато
копче(switch) за мени за песни,

    // според тоа колку пати е кликнато копчето се листа низ менито,

    // односно бројачот почнува од нула, кога прв пат ќе го притиснеме копчето за
мени се вклучува LCD дисплејот,

    // кога прв пат ќе го претиснеме копчето за мени ОДКАКО се вклучил дисплејот
бројачот се зголемува на 1

    // и се прикажува насловот на првата песна,кога ќе го притиснеме копчето за
мени втор пат се прикажува насловот

    // на втората песна итн. Кога ќе стигне бројачот до последната песна и го
притиснеме копчето пак, бројачот се

    // ресетира и LCD дисплејот ќе се исклучи. Кога е исклучен дисплејот тогаш
можеме да ги користиме копчињата

    // на пијано односно седумте копчиња кои се користат за свирење на пијано
односно на седумте ноти.
```



```
//*****  
/** Дефинирање на музички ноти, односно доделување на фреквенција  
//*****  
  
#define NOTE_B0 31  
#define NOTE_C1 33  
#define NOTE_CS1 35  
#define NOTE_D1 37  
#define NOTE_DS1 39  
#define NOTE_E1 41  
#define NOTE_F1 44  
#define NOTE_FS1 46  
#define NOTE_G1 49  
#define NOTE_GS1 52  
#define NOTE_A1 55  
#define NOTE_AS1 58  
#define NOTE_B1 62  
#define NOTE_C2 65  
#define NOTE_CS2 69  
#define NOTE_D2 73  
#define NOTE_DS2 78  
#define NOTE_E2 82  
#define NOTE_F2 87  
#define NOTE_FS2 93  
#define NOTE_G2 98  
#define NOTE_GS2 104  
#define NOTE_A2 110
```

```
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
```

```
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
```

```

#define NOTE_G5 3322
#define NOTE_A5 3520
#define NOTE_AS5 3729
#define NOTE_B5 3951
#define NOTE_C6 4186
#define NOTE_CS6 4435
#define NOTE_D6 4699
#define NOTE_DS6 4978

//***** ПЕСНА 5 НОТИ *****

void Song5() {

    lcd.clear();

    // Ноти во мелодијата:
    int melody[] = {
        NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
    };

    // Времетраење на нотите: 4 = четврт нота, 8 = осмина нота, итн.:
    int noteDurations[] = {
        4, 8, 8, 4, 4, 4, 4, 4
    };

    //Принтање на дисплејот

    lcd.setCursor(1, 0);

```

```
lcd.print("Now playing...");  
lcd.setCursor(0, 1);  
lcd.print("  SONG 5 TITLE");
```

```
int thisNote = 0;
```

```
    // За да се пресмета реалното времетраење на нотите,  
    // треба една секунда да ја поделиме со времетраењето на нотата.  
    // (дефинирани погоре во низата noteDurations)  
    // четврт нота = 1000 / 4, осмина нота = 1000/8, итн.  
for (int thisNote = 0; thisNote < 8; thisNote++) {  
  
    int noteDuration = 1000 / noteDurations[thisNote];  
    tone(8, melody[thisNote], noteDuration);  
  
    //За да можеме да ја чуеме нотата односно да не се мешаат ноти туку да се  
свират една по една,  
  
    // ќе треба да дефинираме време помеѓу нотите и да почекаме да се  
одсвири нотата  
  
    // пред да започне друга нота. Затоа тоа време ќе биде 30% плус од  
времетраењето на една нота  
  
    int pauseBetweenNotes = noteDuration * 1.30;  
    delay(pauseBetweenNotes);  
    noTone(8); //Со оваа функција се стопира свирењето на нота од звучникот:  
}
```

```
Song5title();
```

```
} // крај на Song5 циклусот
```

```
/** ***** ПЕСНА 4 НОТИ *****
```

```
void Song4() {
```

```
    lcd.clear();
```

```
    // Ноти во мелодијата:
```

```
    int melody[] = {
```

```
        NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
```

```
    };
```

```
    // Времетраење на нотите: 4 = четврт нота, 8 = осмина нота, итн.:
```

```
    int noteDurations[] = {
```

```
        4, 8, 8, 4, 4, 4, 4, 4
```

```
    };
```

```
    lcd.setCursor(1, 0);
```

```
    lcd.print("Now playing...");
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("  SONG 4 TITLE");
```

```

int thisNote = 0;

// За да се пресмета реалното времетраење на нотите,
// треба една секунда да ја поделиме со времетраењето на нотата.
// (дефинирани погоре во низата noteDurations)
// четврт нота = 1000 / 4, осмина нота = 1000/8, итн.
for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    //За да можеме да ја чуеме нотата односно да не се мешаат ноти туку да се
    свират една по една,

    // ќе треба да дефинираме време помеѓу нотите и да почекаме да се
    одсвири нотата

    // пред да започне друга нота. Затоа тоа време ќе биде 30% плус од
    времетраењето на една нота

    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8); //Со оваа функција се стопира свирењето на нота од звучникот:
} // крај на for

Song4title();

} // крај на Song4 циклусот

```

```

//***** ПЕСНА 3 НОТИ *****

void Song3() {

    lcd.clear();

    // Ноти во мелодијата:
    int melody[] = {
        NOTE_C2, NOTE_F3, NOTE_C3, NOTE_A2, NOTE_A2,
        NOTE_C3, NOTE_F3, NOTE_C3,
        NOTE_C3, NOTE_F3, NOTE_C3, NOTE_F3,
        NOTE_AS3, NOTE_G3, NOTE_F3, NOTE_E3, NOTE_D3, NOTE_CS3,
        NOTE_C2, NOTE_F3, NOTE_C3, NOTE_A2, NOTE_A2, // repeat line 1 and 2
        NOTE_C3, NOTE_F3, NOTE_C3,
        NOTE_AS3, 0, NOTE_G3, NOTE_F3,
        NOTE_E3, NOTE_D3, NOTE_CS3, NOTE_C3};

    // Времетраење на нотите: 4 = четврт нота, 8 = осмина нота, итн.:
    int noteDurations[] = {
        4,    4,    4,    8,    8,
        4,    4,           2,
        4,    4,    4,    4,
        3,    8,    8,    8,    8,    8,
        4,    4,    4,    8,    8,
        4,    4,           2,
        4,    8,    8,    4,    4,

```



```
4,    4,    4,    4,  
0};
```

```
lcd.setCursor(1, 0);  
lcd.print("Now playing...");  
lcd.setCursor(0, 1);  
lcd.print("    JEOPARDY");
```

```
for (int thisNote = 0; noteDurations[thisNote] != 0; thisNote++) {  
  
    // За да се пресмета реалното времетраење на нотите,  
    // треба една секунда да ја поделиме со времетраењето на нотата.  
    // (дефинирани погоре во низата noteDurations)  
    // четврт нота = 1000 / 4, осмина нота = 1000/8, итн.  
    int noteDuration = 2000/noteDurations[thisNote];  
    tone(8, melody[thisNote],noteDuration * 0.9);  
  
    //За да можеме да ја чуеме нотата односно да не се мешаат ноти туку да се  
    свират една по една,  
  
    // ќе треба да дефинираме време помеѓу нотите и да почекаме да се одсвири  
    нотата  
  
    // пред да започне друга нота. Затоа тоа време ќе биде 30% плус од  
    времетраењето на една нота  
    delay(noteDuration);  
}
```

```
Song3title();
```

```
    } // крај на Song3 циклусот
```

```
//***** ПЕСНА 2 НОТИ *****
```

```
void Song2() {
```

```
    lcd.clear();
```

```
// Ноти во мелодијата:
```

```
int melody[] = {
```

```
    NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
```

```
    NOTE_AS3, NOTE_AS4, 0,
```

```
    0,
```

```
    NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
```

```
    NOTE_AS3, NOTE_AS4, 0,
```

```
    0,
```

```
    NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
```

```
    NOTE_DS3, NOTE_DS4, 0,
```

```
    0,
```

```
    NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
```

```
    NOTE_DS3, NOTE_DS4, 0,
```

```
    0, NOTE_DS4, NOTE_CS4, NOTE_D4,
```

```
NOTE_CS4, NOTE_DS4,  
NOTE_DS4, NOTE_GS3,  
NOTE_G3, NOTE_CS4,  
NOTE_C4, NOTE_FS4, NOTE_F4, NOTE_E3, NOTE_AS4, NOTE_A4,  
NOTE_GS4, NOTE_DS4, NOTE_B3,  
NOTE_AS3, NOTE_A3, NOTE_GS3,  
0, 0, 0  
};
```

// Времетраење на нотите: 4 = четврт нота, 8 = осмина нота, итн.:

```
int noteDurations[] = {  
    12, 12, 12, 12,  
    12, 12, 6,  
    3,  
    12, 12, 12, 12,  
    12, 12, 6,  
    3,  
    12, 12, 12, 12,  
    12, 12, 6,  
    3,  
    12, 12, 12, 12,  
    12, 12, 6,  
    6, 18, 18, 18,  
    6, 6,  
    6, 6,  
    6, 6,  
    18, 18, 18, 18, 18, 18,  
    10, 10, 10,
```

```
10, 10, 10,  
3, 3, 3  
};
```

```
lcd.setCursor(1, 0);  
lcd.print("Now playing...");  
lcd.setCursor(0, 1);  
lcd.print("MARIO UNDERWORLD");
```

```
int size = sizeof(melody) / sizeof(int);  
for (int thisNote = 0; thisNote < size; thisNote++) {  
  
    // За да се пресмета реалното времетраење на нотите,  
    // треба една секунда да ја поделиме со времетраењето на нотата.  
    // (дефинирани погоре во низата noteDurations)  
    // четврт нота = 1000 / 4, осмина нота = 1000/8, итн.  
    int noteDuration = 1000 / noteDurations[thisNote];  
  
    tone(8, melody[thisNote], noteDuration);  
  
    //За да можеме да ја чуеме нотата односно да не се мешаат ноти туку да се  
    свират една по една,  
  
    // ќе треба да дефинираме време помеѓу нотите и да почекаме да се одсвири  
    нотата  
  
    // пред да започне друга нота. Затоа тоа време ќе биде 30% плус од  
    времетраењето на една нота
```

```

    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    // Прекинување на свирењето:
    tone(8, 0, noteDuration);

}

```

```

Song2title();

```

```

} // крај на Song2 циклусот
//***** ПЕСНА 1 НОТИ *****

```

```

void Song1() {

```

```

    lcd.clear();

```

```

// Ноти во мелодијата:

```

```

int melody[] = {
    NOTE_E7, NOTE_E7, 0, NOTE_E7,
    0, NOTE_C7, NOTE_E7, 0,
    NOTE_G7, 0, 0, 0,
    NOTE_G6, 0, 0, 0,

```

```

NOTE_C7, 0, 0, NOTE_G6,
0, 0, NOTE_E6, 0,
0, NOTE_A6, 0, NOTE_B6,
0, NOTE_AS6, NOTE_A6, 0,

NOTE_G6, NOTE_E7, NOTE_G7,
NOTE_A7, 0, NOTE_F7, NOTE_G7,
0, NOTE_E7, 0, NOTE_C7,
NOTE_D7, NOTE_B6, 0, 0,

NOTE_C7, 0, 0, NOTE_G6,
0, 0, NOTE_E6, 0,
0, NOTE_A6, 0, NOTE_B6,
0, NOTE_AS6, NOTE_A6, 0,

NOTE_G6, NOTE_E7, NOTE_G7,
NOTE_A7, 0, NOTE_F7, NOTE_G7,
0, NOTE_E7, 0, NOTE_C7,
NOTE_D7, NOTE_B6, 0, 0
};

```

// Времетраење на нотите: 4 = четврт нота, 8 = осмина нота, итн.:

```

int noteDurations[] = {
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
}

```

```
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
  
9, 9, 9,  
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
  
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
  
9, 9, 9,  
12, 12, 12, 12,  
12, 12, 12, 12,  
12, 12, 12, 12,  
};
```

```
lcd.setCursor(1, 0);  
lcd.print("Now playing...");  
lcd.setCursor(0, 1);  
lcd.print("MARIO THEME SONG");
```

```

int size = sizeof(melody) / sizeof(int);
for (int thisNote = 0; thisNote < size; thisNote++) {

    // За да се пресмета реалното времетраење на нотите,
    // треба една секунда да ја поделиме со времетраењето на нотата.
    // (дефинирани погоре во низата noteDurations)
    // четврт нота = 1000 / 4, осмина нота = 1000/8, итн.
    int noteDuration = 1000 / noteDurations[thisNote];

    tone(8, melody[thisNote], noteDuration);

    //За да можеме да ја чуеме нотата односно да не се мешаат ноти туку да се
    свират една по една,

    // ќе треба да дефинираме време помеѓу нотите и да почекаме да се одсвири
    нотата

    // пред да започне друга нота. Затоа тоа време ќе биде 30% плус од
    времетраењето на една нота

    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    // Прекинување на свирењето:
    tone(8, 0, noteDuration);

}

Song1title();

```



```
} // Крај на Song1 циклусот
```

```
int notes[] = {262, 294, 330, 349, 392, 440, 494}; //Дефинирање на ноти за  
клавијатурата C, D, E, F, G, A, B
```

```
////////////////////////////////////
```

```
//Иницијализација
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    lcd.begin(16, 2);
```

```
    pinMode(LCDswitchPin, INPUT);
```

```
    pinMode(blueLED, OUTPUT);
```

```
    digitalWrite(blueLED, LOW); // LED диодата е исклучена додека копчињата за  
клавијатура вклучени
```

```
} // крај на setup
```

```
////////////////////////////////////
```

```
// Главен циклус (main loop)
```

```
void loop() {
```

```

LCDswitchState = digitalRead(LCDswitchPin);

int keyVal = analogRead(A0); //Читање на копчињата за клавијатура
Serial.println(keyVal);

lcd.noDisplay(); //Исклучување на екранот

digitalWrite(blueLED, LOW); // Исклучување на LED диодата секогаш кога
започнува овој циклус


// ***** Започнување "LCDstart" кога ќе се притисне копчето за мени
*****


if (LCDswitchState == HIGH) {

    digitalWrite(blueLED, HIGH); // Вклучи ја LED диодата за да знаеме дека сме
    во „мени режим“ на работа и дека копчињата за клавијатура се оневозможени

    lcd.display();

    lcd.clear();

    LCDstart();

}


// *****

/////ИСКАЗИ ЗА НОТИ НА КЛАВИЈАТУРАТА И ФРЕКВЕНЦИИ /////

// Скалата од отпорници се користи за да ја смениме волтажата на секое копче од
клавијатурата

```

```
// 220 оми = C5 // 560 оми = D5 // 1K оми = E5 // итн...
```

```
if(keyVal == 1023) {  
    tone(8, notes[0]); // C  
} // крај if
```

```
else if(keyVal >= 990 & keyVal <=1010) {  
    tone(8, notes[1]); // D  
} // крај else if
```

```
else if(keyVal >= 960 && keyVal <= 980) {  
    tone(8, notes[2]); // E  
} // крај else if
```

```
else if(keyVal >= 900 && keyVal <= 945) {  
    tone(8, notes[3]); // F  
} // крај else if
```

```
else if(keyVal >= 650 && keyVal <= 710) {  
    tone(8, notes[4]); // G  
} // крај else if
```

```
else if(keyVal >= 500 && keyVal <= 550) {  
    tone(8, notes[5]); // A  
} // крај else if
```

```
else if(keyVal >= 5 && keyVal <= 10) {  
    tone(8, notes[6]); // B  
} // крај else if
```

```
        else if(keyVal <= 1) { // ако не е притиснато копче, звучникот не свири  
ниту еден звук
```

```
        noTone(8);
```

```
    } // крај else if
```

```
}
```

```
////////////////////////////////////
```

```
void LCDstart() {
```

```
    int keyVal = analogRead(A0);
```

```
    LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето  
за мени
```

```
    while (LCDswitchState == LOW) { // продолжи со циклусот додека копчето за  
мени не биде притиснато
```

```
        LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на  
копчето за мени секогаш кога ќе започне циклусот
```

```
        //***** ЗАПОЧНИ ГИ ЦИКЛУСОТ ЗА НАСЛОВОТ НА ПРВАТА ПЕСНА  
*****
```

```
        if (LCDswitchState == HIGH) { // Ако е притиснато копчето за мени,  
тогаш...
```

```
            LCDswitchCounter++; // зголеми го бројачот, кој ни кажува колку пати е  
кликнато копче за мени, за 1
```

```
            Serial.print("number of button pushes: ");
```

```
            Serial.println(LCDswitchCounter);
```

```
            delay(300); // Потребно време за да се зголеми бројачот
```

```
Song1title();  
lcd.clear();  
break;  
}
```

```
//***** ELSE ПРИКАЖИ ГО МЕНИТО*****
```

```
else { // ако копчето за мени не е притиснато, тогаш...  
    LCDswitchCounter=0; // бројачот го ставаме на 0  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("WHICH SONG WOULD");  
    lcd.setCursor(0, 1);  
    lcd.print("YOU LIKE TO PLAY");  
    delay(50);  
}
```

```
} // крај на while
```

```
} // крај на LCDstart циклусот
```

```
// ***** SONG 1 ЦИКЛУС ЗПОЧНУВА ТУКА *****
```

```
void Song1title() {  
    int keyVal = analogRead(A0);
```

```
LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето  
за мени повторно
```

```
while (LCDswitchState == LOW) { // продолжи со циклусот се додека не се  
притисне копчето за мени
```

```
LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на  
копчето за мени повторно
```

```
//***** ЗАПОЧНУВАЊЕ НА SONG 2 TITLE ЦИКЛУС *****
```

```
if (LCDswitchState == HIGH) { // ако копчето за мени е притиснато,  
тогаш...
```

```
LCDswitchCounter++; // сетирај го бројачот на 2
```

```
Serial.print("number of button pushes: ");
```

```
Serial.println(LCDswitchCounter);
```

```
delay(300); // Потребно време за да се зголеми бројачот
```

```
Song2title();
```

```
lcd.clear();
```

```
break;
```

```
}
```

```
else { // ако копчето за мени не е притиснато, тогаш...
```

```
int keyVal = analogRead(A0);
```

```
Serial.println(keyVal);
```

```
if(keyVal >= 2) {
```

```
digitalWrite(blueLED, LOW);
```

```
Song1();
```

```
break;
```

```

        } // end if

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SUPER MARIO BROS");
        lcd.setCursor(0, 1);
        lcd.print("  THEME SONG");
        delay(50);
    }
} // крај на while песна 1 -> 2

} // крај на Song1title циклус

// ***** SONG 2 ЦИКЛУС ЗАПОЧНУВА ОВДЕ *****

void Song2title() {
    int keyVal = analogRead(A0);

    LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето
    за мени повторно

    while (LCDswitchState == LOW) { // продолжи со циклусот се додека не се
    притисне копчето за мени

        LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на
        копчето за мени повторно

        //***** ЗАПОЧНУВАЊЕ НА SONG 3 TITLE ЦИКЛУС *****

        if (LCDswitchState == HIGH) { // ако копчето за мени е притиснато,
        тогаш...

```

```
LCDswitchCounter++; // сетирај го бројачот на 3
Serial.print("number of button pushes: ");
Serial.println(LCDswitchCounter);
delay(300); // Потребно време за да се зголеми бројачот
```

```
Song3title();
lcd.clear();
break;
}
```

```
else { // ако копчето за мени не е притиснато, тогаш...
```

```
int keyVal = analogRead(A0);
Serial.println(keyVal);
```

```
if(keyVal >= 2) {
    digitalWrite(blueLED, LOW);
    Song2();
    break;
} // end if
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("SUPER MARIO BROS");
lcd.setCursor(0, 1);
lcd.print("    UNDERWORLD");
delay(50);
}
```



```

    } // крај на while песна 2 -> 3

} // крај на Song2title циклусот

// ***** SONG 3 ЦИКЛУС ЗАПОЧНУВА ОВДЕ *****

void Song3title() {
    int keyVal = analogRead(A0);

    LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето
    за мени повторно

    while (LCDswitchState == LOW) { // продолжи со циклусот се додека не се
    притисне копчето за мени

        LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на
        копчето за мени повторно

        //***** ЗАПОЧНУВАЊЕ НА SONG 4 TITLE ЦИКЛУС *****

        if (LCDswitchState == HIGH) { // ако копчето за мени е притиснато,
        тогаш...

            LCDswitchCounter++; // сетирај го бројачот на 4

            Serial.print("number of button pushes: ");
            Serial.println(LCDswitchCounter);

            delay(300); // Потребно време за да се зголеми бројачот

            Song4title();

            lcd.clear();

            break;

        }
    }
}

```

```

else { // ако копчето за мени не е притиснато, тогаш...

    int keyVal = analogRead(A0);
    Serial.println(keyVal);

    if(keyVal >= 2) {
        digitalWrite(blueLED, LOW);
        Song3();
        break;
    } // end if

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("    JEOPARDY");
    lcd.setCursor(0, 1);
    lcd.print("    THEME SONG");
    delay(50);
}

} // крај на while песна 3 -> 4

} // крај на Song3title циклус

```

```

// ***** SONG 4 ЦИКЛУС ЗАПОЧНУВА ОВДЕ *****

void Song4title() {
    int keyVal = analogRead(A0);

```

```
LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето  
за мени повторно
```

```
while (LCDswitchState == LOW) { // продолжи со циклусот се додека не се  
притисне копчето за мени
```

```
LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на  
копчето за мени повторно
```

```
//***** ЗАПОЧНУВАЊЕ НА SONG 5 TITLE ЦИКЛУС *****
```

```
if (LCDswitchState == HIGH) { // ако копчето за мени е притиснато,  
тогаш...
```

```
LCDswitchCounter++; // сетирај го бројачот на 5
```

```
Serial.print("number of button pushes: ");
```

```
Serial.println(LCDswitchCounter);
```

```
delay(300); // Потребно време за да се зголеми бројачот
```

```
Song5title();
```

```
lcd.clear();
```

```
break;
```

```
}
```

```
else { // ако копчето за мени не е притиснато, тогаш...
```

```
int keyVal = analogRead(A0);
```

```
Serial.println(keyVal);
```

```
if(keyVal >= 2) {
```

```
digitalWrite(blueLED, LOW);
```

```
Song4();
```

```
break;
```

```

        } // end if

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("    SONG 4");
        lcd.setCursor(0, 1);
        lcd.print("    TITLE HERE");
        delay(50);
    }

} // крај на while песна 4 -> 5

} // крај на Song4title циклусот

// ***** SONG 5 ЦИКЛУС ЗАПОЧНУВА ОВДЕ *****

void Song5title() {
    int keyVal = analogRead(A0);

    LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на копчето
    за мени повторно

    while (LCDswitchState == LOW) { // продолжи со циклусот се додека не се
    притисне копчето за мени

        LCDswitchState = digitalRead(LCDswitchPin); // провери ја состојбата на
        копчето за мени повторно

    }

    //***** ИЗЛЕЗ ОД „МЕНИ РЕЖИМ“ *****

```

```

    if (LCDswitchState == HIGH) { // ако копчето за мени е притиснато,
    тогаш...

        LCDswitchCounter=0; // сетирај го бројачот на 0

        Serial.print("number of button pushes: ");

        Serial.println(LCDswitchCounter);

        delay(300); // // Потребно време за да се зголеми бројачот


        lcd.clear();

    }

    else { // ако копчето за мени не е притиснато, тогаш...

        int keyVal = analogRead(A0);

        Serial.println(keyVal);


        if(keyVal >= 2) {
            digitalWrite(blueLED, LOW);

            Song5();

            break;
        } // end if


        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("    SONG 5");

        lcd.setCursor(0, 1);

        lcd.print("    TITLE HERE");

        delay(50);

    }

} // крај на while песна 5 -> излез од „мени режим“

```

```
} // крај на Song5title циклусот
```