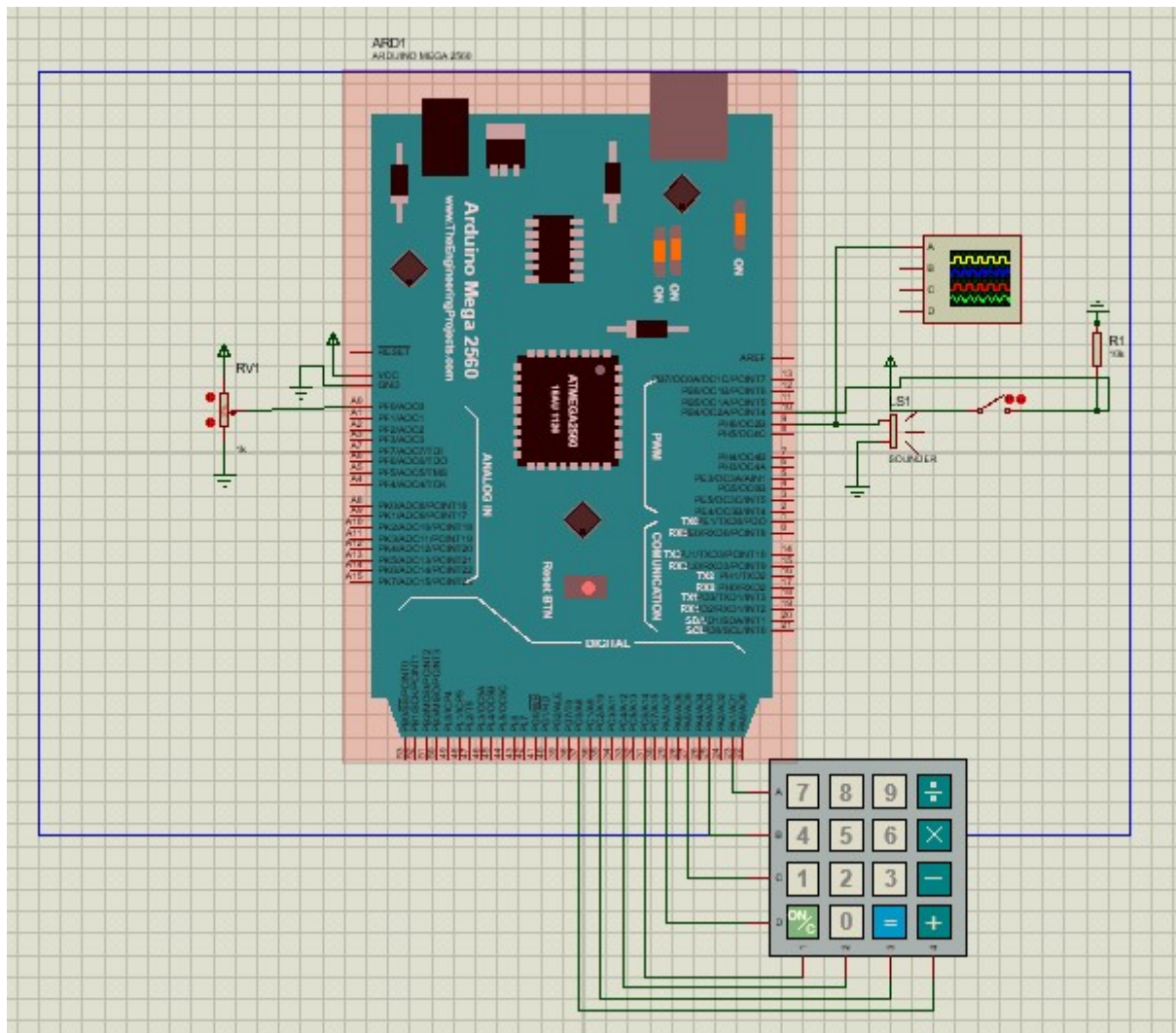


مدار پیاده سازی شده به صورت زیر است



خروجی ها هم با توجه به شماره ی کپی به صورت صدا پخش می شود. در اصل با فشردن هر کدام از دکمه های کپی یک فایل از ملودی پخش می شود.

1- برنامه ای بنویسید که هنگام فشردن هر کلید فایل ملودی متناظر با آن را پخش کند.

برای انجام ایم کار فایل هایی که در کانال برای مشخص کردن فرکانس نت ها و ملودی ها بود استفاده کردیم و کد زیر را نوشتیم.

ابتدا keypad را تعریف کردیم و پین اسپیکر را مشخص کردیم و کتابخانه های مورد نیاز را include کردیم. در تابع setup تابع keypadEvent را به eventlistener در keypad اضافه می کنیم و پین مربوط به پنتاسیومتر را ورودی تعریف می کنیم. در تابع loop، برنامه همواره منتظر ورود یک عدد از سمت keypad است تا بعد از دریافت ورودی از keypadEvent را اجرا کند. در تابع keypadEvent با توجه به عدد انتخاب شده از keypad تابع playMelody برای آهنگ متناظر با آن ورودی صدا شده می شود. در تابع playMelody در یک for loop یکی یکی نت ها را به کمک تابع tone اجرا می کند.

sketch_may25a

pitches.h

themes.h

```
#include "pitches.h"
#include "themes.h"
#include <Keypad.h>

const byte rows = 4; //four rows
const byte cols = 4; //four columns
char keys[rows][cols] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'0','0','=','+'}
};

byte rowPins[rows] = {23, 25, 27, 29}; //connect to the row pinouts of the keypad
byte colPins[cols] = {31, 33, 35, 37}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols );

#define PIEZO 9

void setup() {
  // put your setup code here, to run once:
  keypad.addEventListener(keypadEvent);
  pinMode(10,INPUT);
}

void loop() {
  char key = keypad.getKey();
}

void playMelody(int melody[], float noteDurations[], int m_size, int num) {
  for (int thisNote = 0; thisNote < m_size; thisNote++) {
    int key=digitalRead(10);
    float scale = analogRead(A0) / 512.0;
    if(key == LOW){
      scale = 1.0;
    }
  }
}
```

```

    int noteDuration = num / noteDurations[thisNote];
    tone(PIEZO, (int)(melody[thisNote] * scale), noteDuration);
    delay((int)noteDuration * 1.2);
}

noTone(9);
}

void playImperial(int melody[], int noteDurations[], int m_size, int num) {
// float scale = 0;
for(int thisNote = 0; thisNote < m_size ;thisNote++){
    int key=digitalRead(10);
    float scale=analogRead(A0) / 512.0;
    int noteDuration;
    if(key == LOW){
        scale = 1.0;
    }
    if(noteDurations[thisNote] < 0){
        noteDuration = num / (-1 * noteDurations[thisNote]);
        noteDuration = noteDuration * 1.5;
    }
    else{
        noteDuration = num / noteDurations[thisNote];
    }
    tone(9, (int)(melody[thisNote] * scale),noteDuration);
    delay((int)(noteDuration * 1.3));
}
    noTone(9);
}

void keypadEvent(KeypadEvent key){
    if(key == '1'){
        playMelody(odeToJoy, odeToJoyDurations, sizeof(odeToJoy) / sizeof(int), 800);
    }
    if(key == '2'){
        playMelody(jingleBells, jingleBellsDurations, sizeof(jingleBells) / sizeof(int), 600);
    }
    if(key == '3'){
        playImperial(imperialMarch, imperialMarchDurations, sizeof(imperialMarch) / sizeof(int), 500);
    }
    if(key == '4'){
        playMelody(PiratesOfCaribbean, PiratesOfCaribbeanDurations, sizeof(PiratesOfCaribbean) / sizeof(int), 800);
    }
    if(key == '5'){
        playMelody(mario, marioDurations, sizeof(mario) / sizeof(int), 600);
    }
    if(key == '6'){
        playMelody(iran, iranDurations, sizeof(iran) / sizeof(int), 800);
    }
}
}

```

2- فرکانس کلاک مورد را پایین بیاورید تا به جایی برسید که موسیقی به نرمی اجرا شود متوجه می شوید که فرکانس تنها هم عوض می شود پس با ایجاد تغییری کوچک در کد آن را هم رفع کند.

این کار انجام شده است و برای روان تر اجرا شدن موسیقی باید متغیر num در ورودی تابع playMelody را تغییر میدادیم.

3- یک پنتاسیومتر به برد وصل کنید همانطور که قبلا تابع map ورودی آنالوگ را استفاده کردید اکنون برنامه ای بنویسید که هر دفعه که دکمه فشرده می شود بنا به وضعیت پنتاسیومتر یک نت زیر تر یا بم تر اجرا کند.

این کار در قسمتی از کد در تابع playMelody به کمک متغیر scale انجام میشود. اگر دکمه فشرده شده باشد متغیر scale با استفاده از ورودی پنتاسیومتر تعریف می شود و عددی بین 0 تا 2 خواهد داشت. اگر کلید فشار داده نشده باشد این متغیر مقدار 1 خواهد داشت. و در نهایت مقدار این متغیر در ورودی فرکانس تابع tone ضرب می شود و با تغییر فرکانس صدا را بم تر یا زیر تر میکند.

```
float scale = analogRead(A0) / 512.0;
if(key == LOW){
    scale = 1.0;
}
int noteDuration = num / noteDurations[thisNote];
tone(PIEZO, (int)(melody[thisNote] * scale), noteDuration);
```

4- با دانشی که دارید برنامه ای بنویسید که ملودی **Ode to Joy** بتهوون را پخش کند.

این بخش همراه با قسمت اول انجام شده است. برای این کار در فایل **themes.h** کدهای زیر را اضافه کرده ایم.

```
int odeToJoy[] = {
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4,
    NOTE_G4, NOTE_F4, NOTE_E4, NOTE_D4,
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,
    NOTE_E4, NOTE_D4, NOTE_D4,
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4,
    NOTE_G4, NOTE_F4, NOTE_E4, NOTE_D4,
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,
    NOTE_D4, NOTE_C4, NOTE_C4,
    NOTE_D4, NOTE_D4, NOTE_E4, NOTE_C4,
    NOTE_D4, NOTE_E4, NOTE_F4, NOTE_E4, NOTE_C4,
    NOTE_D4, NOTE_E4, NOTE_F4, NOTE_E4, NOTE_D4,
    NOTE_C4, NOTE_D4, 0, 0
};

float odeToJoyDurations[] = {
    2, 2, 2, 2,
    2, 2, 2, 2,
    2, 2, 2, 2,
    1.33, 4, 1,
    2, 2, 2, 2,
    2, 2, 2, 2,
    2, 2, 2, 2,
    1.33, 4, 1,
    2, 2, 2, 2,
    2, 4, 4, 2, 2,
    2, 4, 4, 2, 2,
    2, 2, 2, 2
};
```

5-ملودی امتیازی:

```
int imperialMarch[] = {
    NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_F4, 0,
    NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_F4, 0,
    NOTE_A4, NOTE_A4, NOTE_A4, NOTE_F4, NOTE_C5,

    NOTE_A4, NOTE_F4, NOTE_C5, NOTE_A4,
    NOTE_E5, NOTE_E5, NOTE_E5, NOTE_F5, NOTE_C5,
    NOTE_A4, NOTE_F4, NOTE_C5, NOTE_A4,

};

int imperialMarchDurations[]={
    -4,-4,16,16,16,16,8,8,
    -4,-4,16,16,16,16,8,8,
    4,4,4,-8,16,

    4,-8,16,2,
    4,4,4,-8,16,
    4,-8,16,2,

};
```