

Wrocław University of Science and Technology

Final Report **Basic piano using arduino**

Author: Vadson Culanda

Instructor: Witold Paluszyński

Class: Intermediate project- Embedded Robotics

Date: 01/02/2019

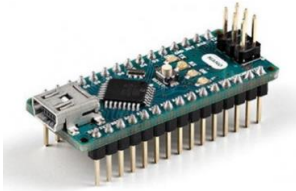
Summary: In this project i implemented a basic piano with only 7 notes, using push buttons, a buzzer and a led matrix. We can play using the push buttons and we can even listen to 3 different preset songs.

1. Introduction

In this project I tried to find an idea that was compatible with my interest and considering that I've been learning how to play the piano, I figured the best thing I could do is to combine both, electronic and music. And I decided to implement a simple piano using Arduino.

2. Components

1 arduino nano.



10 push buttons:



1 led matrix 8x8:

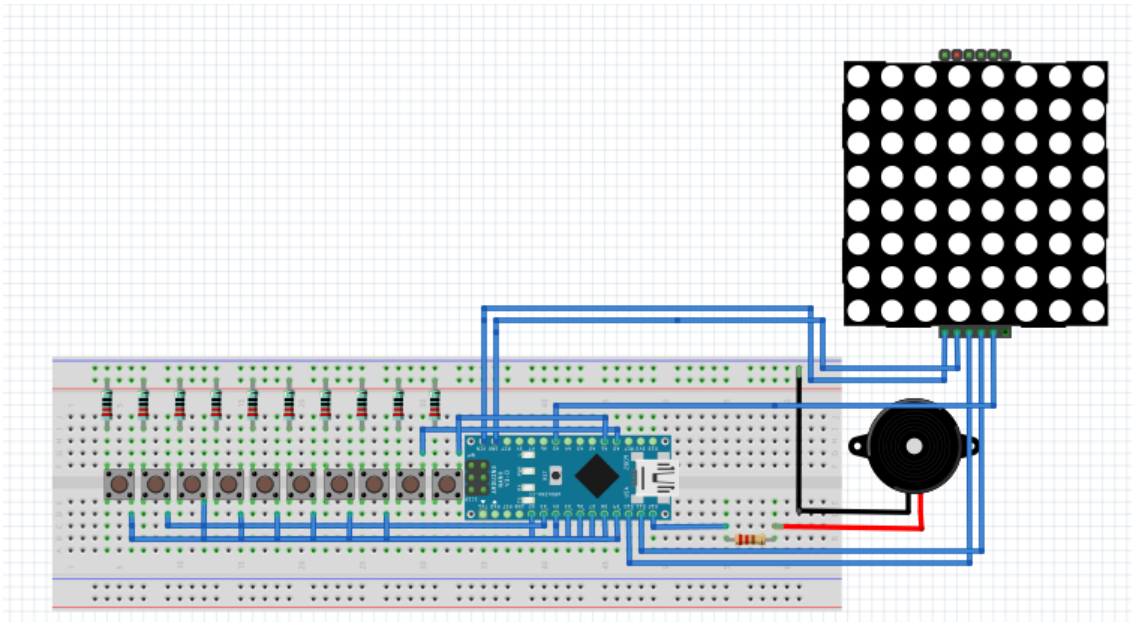


1 buzzer:



10 resistors of 10k Ω and 1 resistor of 10 Ω

3. Circuit set up



- The firsts 7 buzzers correspond to the notes, we you press a buzzer it sends the buzzer the information of which note to play.
- The 8th buzzer correspond to the song selection each time we press the buzzer a different song plays, who was previously implemented, plays. We can listen up to 3 different songs.
- The 9th and 10th buttons allows to change the scale of the notes, to an higher or lower octave.
- The led matrix shows which note we are currently playing, and the number of the song we are listening.
- The resistor in the buzzer allow us to change volume of the buzzer the lower the resistor the higher the volume.

4. Code explanation

In this part I'll explain how the code works using a few images with portions off my code. The programming was done using c++. I started by defining the frequency of each note. I defined more than 8 octaves of notes, which allow me to have enough variety to play around with. To use the led matrix I used a library called LedControl.h, which is available for anyone to use it as we can see in the resources [1].

```
#include <LedControl.h>

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
```

Then I declared the pins related to the buzzer and pushbuttons:

```

const int BUTTON_C = 2;
const int BUTTON_D = 3;
const int BUTTON_E = 4;
const int BUTTON_F = 5;
const int BUTTON_G = 6;
const int BUTTON_A = 7;
const int BUTTON_B = 8;
const int L_scale = A0;
const int H_scale = A1;
const int BUZZER = 12;

```

Then I declared the pins for the led matrix:

```

const int DIN = 10;
const int CS = 11;
const int CLK = A5;

```

Then I declared some variables and the byte arrays which represent characters or graphics which I would like to display.

```

int buttonSong = 9;
int song = 1;
int scale = 1;
byte one[8] = {0x18, 0x28, 0x48, 0x08, 0x08, 0x08, 0x18, 0x7E};
byte two[8] = {0x7C, 0x04, 0x04, 0x7C, 0x40, 0x40, 0x7C, 0x00};
byte three[8] = {0x3E, 0x02, 0x02, 0x3E, 0x02, 0x02, 0x3E, 0x00};
byte four[8] = {0x22, 0x22, 0x22, 0x22, 0x3E, 0x02, 0x02, 0x02};
byte c[8] = {0x00, 0x3C, 0x40, 0x40, 0x40, 0x40, 0x3C, 0x00};
byte d[8] = {0x00, 0x3C, 0x22, 0x22, 0x22, 0x22, 0x3C, 0x00};
byte e[8] = {0x00, 0x3C, 0x20, 0x20, 0x3C, 0x20, 0x20, 0x3C};
byte f[8] = {0x00, 0x3E, 0x20, 0x20, 0x3E, 0x20, 0x20, 0x20};
byte g[8] = {0x00, 0x3E, 0x20, 0x20, 0x2E, 0x22, 0x22, 0x3E};
byte a[8] = {0x00, 0x1C, 0x22, 0x22, 0x22, 0x3E, 0x22, 0x22};
byte b[8] = {0x00, 0x3C, 0x22, 0x22, 0x3C, 0x22, 0x22, 0x3C};

```

Then I initialize the LED control library by creating an object of the library after which I moved to the void setup function where we prepare the display and define the pins as inputs (in the setupbuttons() function)

```

void setup() {
  lc.shutdown(0, false);          //The led matrix is in power-saving mode on startup
  lc.setIntensity(0, 15);        // Set the brightness to maximum value
  lc.clearDisplay(0);            // and clear the display
  Serial.begin(9600);
  setupbuttons();                //setup of buttons
}

void setupbuttons() {
  pinMode(BUTTON_C, INPUT);
  digitalWrite(BUTTON_C, HIGH);
  pinMode(BUTTON_D, INPUT);
  digitalWrite(BUTTON_D, HIGH);
  pinMode(BUTTON_E, INPUT);
  digitalWrite(BUTTON_E, HIGH);
}

```

Then I define the arrays containing the notes in the song and the durations of each note:

```

int song1[] = {
NOTE_AS4, NOTE_C5, NOTE_AS4,
NOTE_DS4, NOTE_DS4, NOTE_F4,
NOTE_G3, NOTE_AS3, NOTE_C4,
NOTE_G3, NOTE_AS3, NOTE_C4,

int noteDurations1[] = {
4.5, 2.25, 2.25, 4.5, 2.25, 2.25,
4.5, 2.25, 4.5, 2.25, 1.125, 2.25
4.5, 4.5, 4.5, 2.25, 4.5, 4.5, 4.
4.5, 4.5, 4.5, 2.25, 4.5, 4.5,

```

Then we have the loop function in which I define what happens when I press each button, we have 3 different scales for the notes, depending on this variable value which is incremented/decremented each time we press the H_scale/L_scale button. And depending on the value of the song variable we play one of the 3 preset songs(this variable is incremented when we press the buttonSong). To implement the songs I based myself on youtube piano tutorials[5][6][7].

```

void loop()
{
    if(digitalRead(H_scale)== ACTIVATED){
        scale=scale+1;
    }
    if(digitalRead(L_scale)== ACTIVATED){
        scale=scale-1;
    }
    if(scale==0){

        while(digitalRead(BUTTON_C) == ACTIVATED)
        {
            tone(BUZZER,NOTE_C3);
            printByte(c);                //print the character c
        }
        while(digitalRead(BUTTON_D) == ACTIVATED)
        {
            tone(BUZZER,NOTE_D3);
            printByte(d);                //print the character d
        }

    if(digitalRead(buttonSong) == ACTIVATED) {
        if(song==1){
            int size = sizeof(song1) / sizeof(int);
            for (int thisNote=0; thisNote <size; thisNote++) {
                int noteDuration = 600 / noteDurations1[thisNote];
                tone(BUZZER, song1[thisNote], noteDuration);
                int pauseBetweenNotes = noteDuration * 1.50;
                delay(pauseBetweenNotes);
                noTone(BUZZER);
                printByte(one);          //print in display
            }
            lc.clearDisplay(0);
        }
    }
}

```

And finally I implemented the function that allows me to print each character in the led matrix:

```

void printByte(byte character [])
{
    int i = 0;
    for(i=0;i<8;i++)
    {
        lc.setRow(0,i,character[i]);
    }
}

```

5. Conclusion

I was able to achieve all my goals for the project, the circuit works as a piano, plays preset songs, displays the number of songs it's currently playing and the key pressed. There are a few limitations in my project, i can't play 2 notes at the same time, but that could be fixed by adding another buzzer that way we could listen to different sounds at the same time.

6. Resources

- [1]- <http://educ8s.tv/arduino-8x8-led-matrix-tutorial/>
- [2]- <http://www.ardumotive.com/how-to-use-a-buzzer-en.html>
- [3]- <https://www.instructables.com/id/How-to-use-a-Push-Button-Arduino-Tutorial/>
- [4]- <https://github.com/wayoda/LedControl>
- [5]- <https://www.youtube.com/watch?v=-AVWEtIG3iQ>
- [6]- <https://www.youtube.com/watch?v=OzNhUtHutIg>
- [7]- <https://www.youtube.com/watch?v=aQlZg0zenfQ>