MIT World Peace University

Basics of Electronics and Electrical Engineering
ECE1022A, School of ECE
First Year B. Tech, Trimester 2
Academic Session 2021-22

8 Bit Mp3 Player

Project Report
For Project Based Learning Activity

Prepared By

Krishnaraj Thadesar
Division 9, Roll. 109054 (I3)
PRN. 1032210888

February 21, 2022

Contents

1 Introduction

An Mp3 Player is a device that has only one purpose, which is to play Mp3 files or songs. It can sometimes include a Display to let you know which song is being played, and what the next songs are.

8 Bit Music got its name from Music generated from *PSG(programmable Sound Generator)* Chip, located usually on 8-bit Microprocessors as opposed to Modern 64 Bit ones. It uses limited Frequencies to create any sound or tune.

In This project we will use **Arduino UNO** which uses an **ATMega328** chipset, which is an *8 Bit processor* combined with an LCD Screen to make a very simple Music Player.

2 Circuit Diagram

2.1 Schematic Diagram

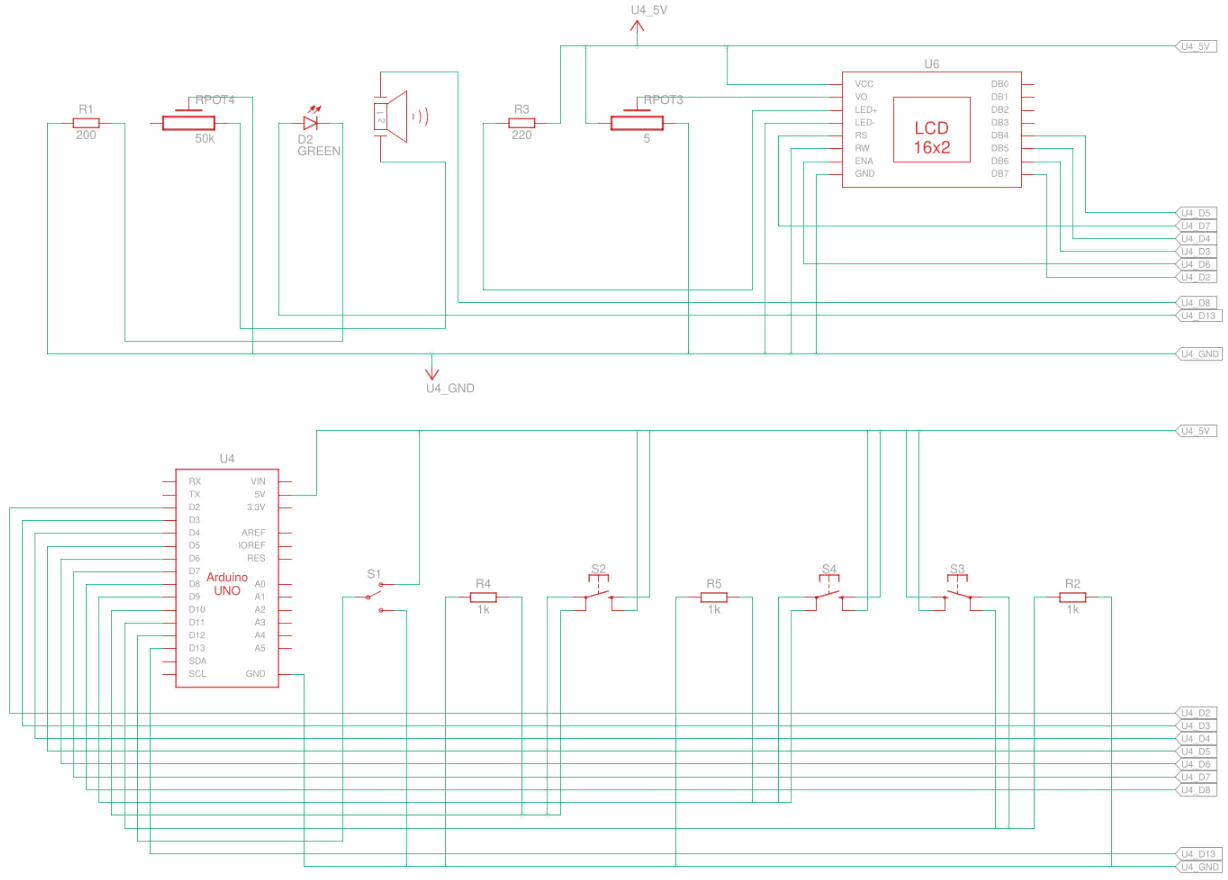


Figure 1: Schematic Diagram

2.2 Simulation

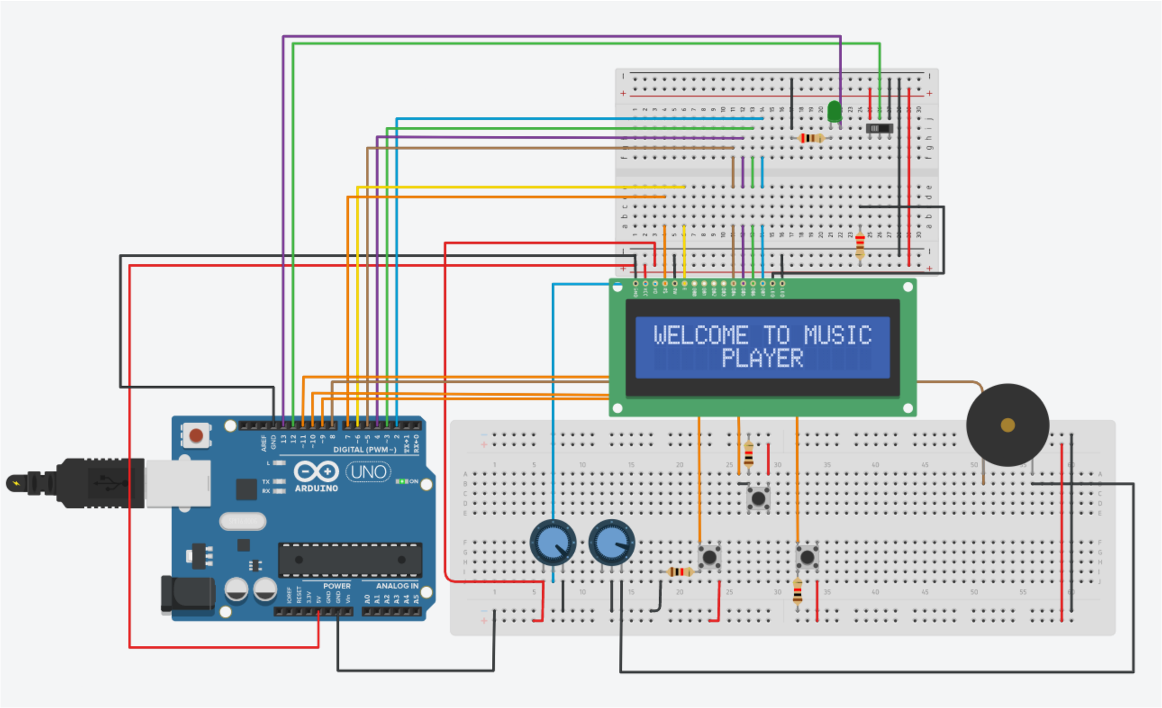


Figure 2: MP3 Player Simulation

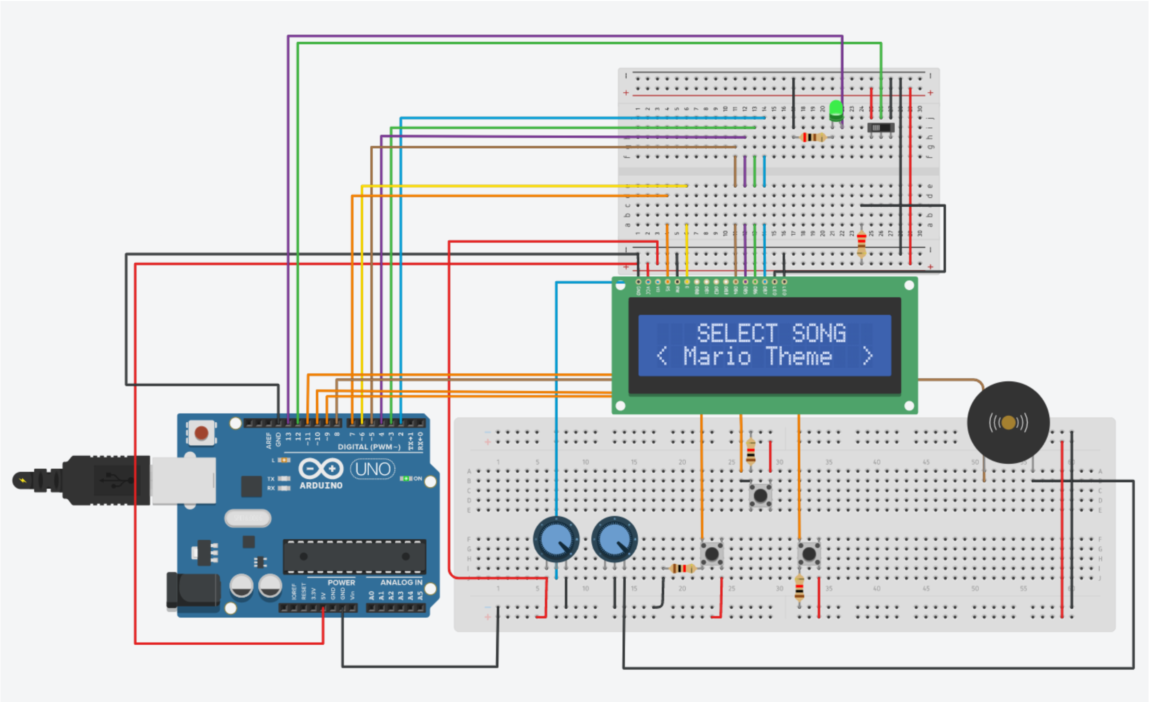


Figure 3: MP3 Player Simulation while working

3 Component List

The components Used For making the Circuit are listed Below

Table 1: Component List for 8 Bit MP3 Player

Name	Quantity	Component
U4	1	Arduino Uno R3
U6	1	LCD 16 x 2
PIEZO2	1	Piezo
D2	1	Green LED
S2, S3, S4	3	Pushbutton
Rpot3	1	5 Ω Potentiometer
Rpot4	1	50 kΩ Potentiometer
R3	1	220 Ω Resistor
R1	1	200 Ω Resistor
R4, R2, R5	3	1 kΩ Resistor
S1	1	Slideswitch

4 Circuit Explanation and Working

4.1 Power

Power to the Arduino is given from a standard Li-Ion 3.7V Battery or a 5V Battery or Wall Adapter that outputs 5V 2A. Power to the LCD Screen and the Buzzer is then supplied from the Arduino.

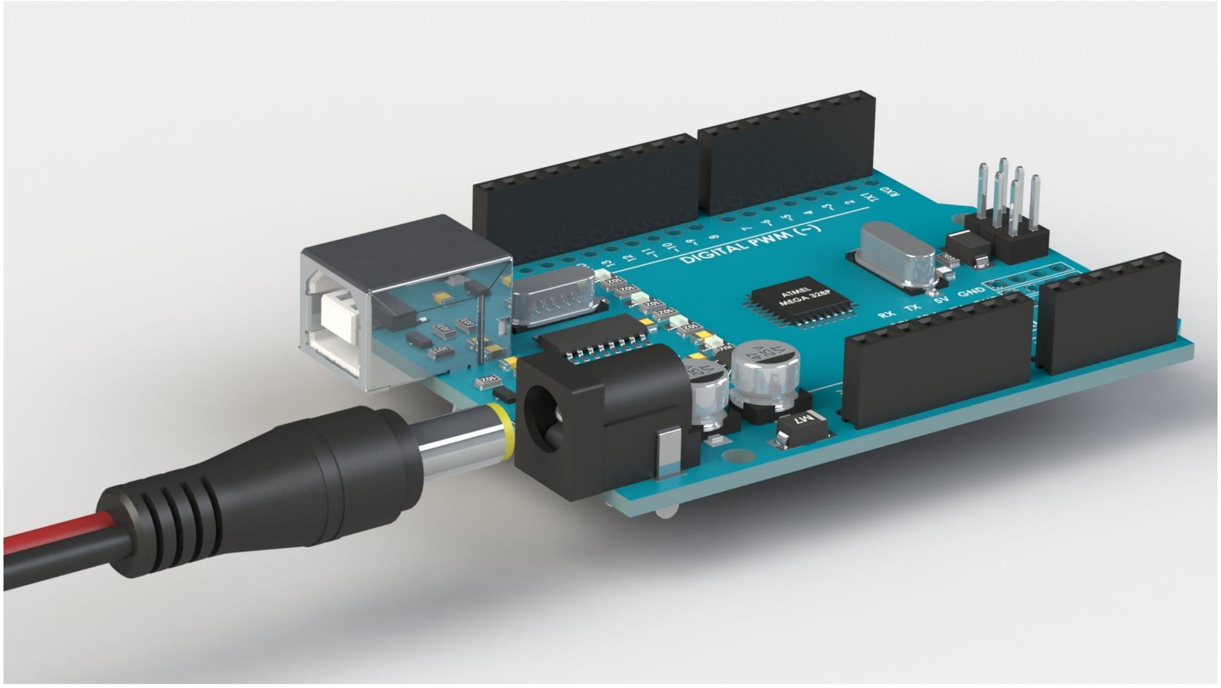


Figure 4: Power To the Arduino

4.2 Inputs

Inputs are taken from The Main switch, that controls the Entire working, and push buttons that control the playback. The Volume of the buzzer is controlled by a Potentiometer that simply increases the resistance, and thereby reducing the current passing in the Buzzer, resulting in control of sound.

Another Potentiometer controls the Brightness of the LCD Screen in the same way.

Figure 5: Inputs from Buttons are given to Digital PINS of the Arduino

4.3 Processing

All of the Processing is done with the Arduino, and the ATMEGA8U2 Chip inside it. If the play button is pressed, it sends a HIGH signal to pin 11 of the Arduino, which is detected, and the Arduino then sends calculated power processed in milliseconds by taking instructions from the code to PIN 8, where the Buzzer is connected.

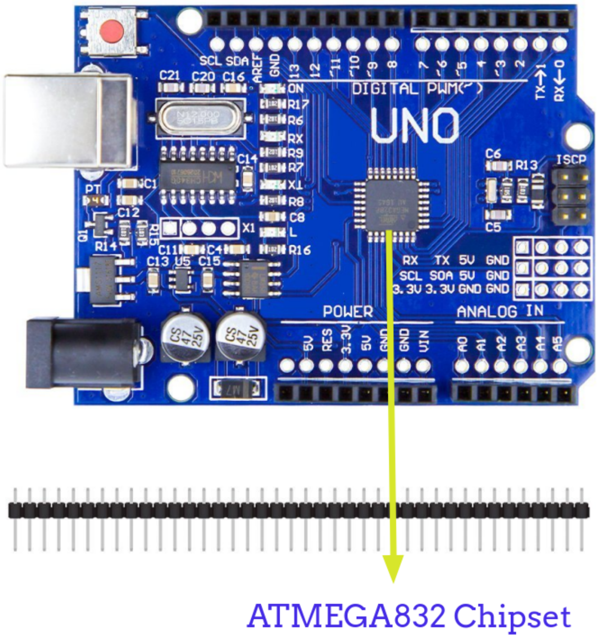


Figure 6: Chip of the Arduino

Similarly, When buttons are pressed, they send a voltage to another PIN of the Arduino that is detected, and the song is changed accordingly.

4.4 Code

The Complete code is not shown here due to space restrictions, only the main function names are given to get a crude idea.

```
#include <LiquidCrystal.h>
#include <string.h>

// Defining the Basic Notes
#define SA 256
#define RE 280
#define GA 312
#define MA 346
#define PA 384
#define DHA 426
#define NI 480

// Defining the Arduino Input Pins
#define PREV_PIN 10
#define FORWARD_PIN 11
#define PLAY_PIN 9
#define ON_PIN 13
#define SWITCH_PIN 12
#define melodyPin 8

int basic_frequencies[7] = {SA, RE, GA, MA, PA, DHA, NI};
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

// Declaring Basic Variables and Booleans.
int song = 0, end = 0;
int playing = 0, pause = 1;

// List of Songs
String songs[9] = {
    "Mario Theme",
    "Mario Theme 2",
    "Frequency Sa",
    "Frequency Re",
    "Frequency Ga",
    "Frequency Ma",
    "Frequency Pa",
    "Frequency Dha",
    "Frequency Ni",
};

// Mario main theme melody
int melody[] = {};

// Mario main them tempo
int tempo[] = {};
```

```
// Mario Theme 2 melody
int mario_theme_2_melody[] = {};

// Underworld tempo
int mario_theme_2_tempo[] = {};

// Control which notes to play
void sing(int s) {}

// Play those notes.
void buzz(int targetPin, long frequency, long length) {}

// Main Setup Function
void setup()
{
    // Print WELCOME TO MUSIC PLAYER on the LCD Screen
    // Print First Song after 3 Seconds

    // Declare Output PINS
    pinMode(melodyPin, OUTPUT); // buzzer
    pinMode(ON_PIN, OUTPUT); // LED

    // Declare INPUT PINS
    pinMode(SWITCH_PIN, INPUT); // main switch
    pinMode(PLAY_PIN, INPUT); // Play/Pause
    pinMode(PREV_PIN, INPUT); // Previous song
    pinMode(FORWARD_PIN, INPUT); // Next song
}

void loop()
{
    // Check if The Main switch is turned on, only then we do anything.
    if (digitalRead(SWITCH_PIN) == 1)
    {
        // Switch on the LED indicating that we have turned on.
        digitalWrite(ON_PIN, HIGH);

        // If user presses forward, change song and show it.
        if (digitalRead(FORWARD_PIN) == 1) {}

        // If user presses Previous, change song and show it.
        if (digitalRead(PREV_PIN) == 1) {}

        // If user presses play, then play the song
        if ((digitalRead(PLAY_PIN) == 1) && (playing == 0) && (pause = 1)) {}
        end = 0;
    }

    // If the main switch is off, then turn of the LED and Print Thank You.
    else if ((digitalRead(SWITCH_PIN) == 0) && (end == 0)){}
}
```

4.5 Outputs

PIN 8 of the Arduino sends Power to the buzzer or alternatively a speaker that will produce the sound at the desired Frequency.

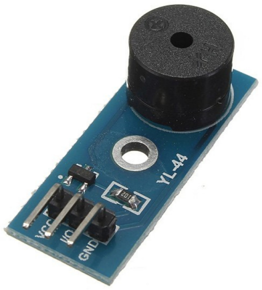
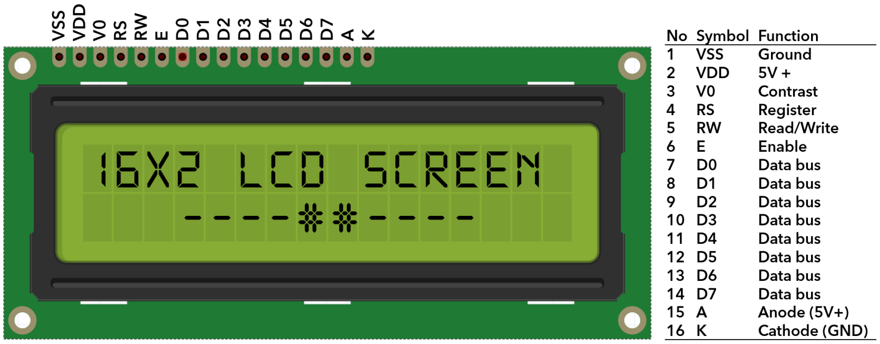


Figure 7: The Speaker or Buzzer in User

Digital PIN 2 - Digital PIN 7 of the Arduino, along with 5V Power and Ground connects to various inputs of the LCD Screen which then outputs the current song that is being played.



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

Figure 8: 16 Pin LCD Screen

5 Applications

1. An Arduino being far superior than required for a project like this one, would make it impractical to buy and use, but the Chipset, and certain music ICs paired with the same arrangement as in this project can be used to make a good 8-Bit Music Player.

2. The Arduino itself has numerous Applications like:

   (a) Weighing Machines.

   (b) Traffic Light Count Down Timer.

   (c) Parking Lot Counter.

   (d) Home Automation.

   (e) Industrial Automation.

   (f) Medical Instrument.

   (g) Emergency Light for Railways.

3. The LCD Screen can also be used to display anything from a clock, a timer, weights, Temperatures, or any needed value, cheaply and easily.

4. More Songs can be programmed and added to then put this into something very small and portable for a quick, simple and cheap Music Player.

5.1 Improvements

1. The Expensive Arduino can be replaced with the chipset used inside or another Music IC coupled with another cheaper Microprocessor IC to reduce the cost (while increasing the complications) of the Project.

2. A Better Buzzer can be used for better sound Quality

3. A Program can be written that automatically converts any song into its 8 Bit code version so it becomes easy to add new songs, instead of hard coding them into the Aruduino.

4. An External Storage of some sort can be added like an SD Card to store Thousands of other Songs.

5. All The components can be put in a small Enclosing for increasing Portabitlity.

6 Conclusion

6.1 Learning Experience

1. I got to interact with the Basic LCD Screen, and so now that can be used in many other places, which is an important output device.

2. Exposure to a Microprocessor and working on such a low level of input and output gave a better and more open idea about the working of a computer.

3. 2 Terminal Switches are basic, but 3 terminal switches with a ground pin make use of more Electrical concepts which were put to use by actually using them. This also forced me to learn the basics of the Potentiometer and its detailed working.

4. Programming the Basics of Music also made me learn some Music Theory, reading some music notes, and an in depth working of a speaker.

6.2 Conclusion

A very simple 8-bit Music Player was made, programmed and Built in Tinkercad successfully. The Working, Programming, and basic functioning of the Arduino, a Buzzer, an LCD Screen, Switches and Potentiometer was studied and Applied in detail.

References

[1] Tinkercad Website for Circuit Design
`https://www.tinkercad.com/`

[2] Mario Theme Song on Tinkercad
`https://www.tinkercad.com/things/26ARFLeA0mg-mario-theme-song`

[3] Music Notes for the Mario Theme Song
`https://www.youtube.com/watch?v=soTO2ywdViA`

[4] Working of the Potentiometer
`https://www.electrical4u.com/potentiometer`

[5] Working of the 3 Way Switches
`https://matthews.sites.wfu.edu/courses/p230/switches/SwitchesTut.html`