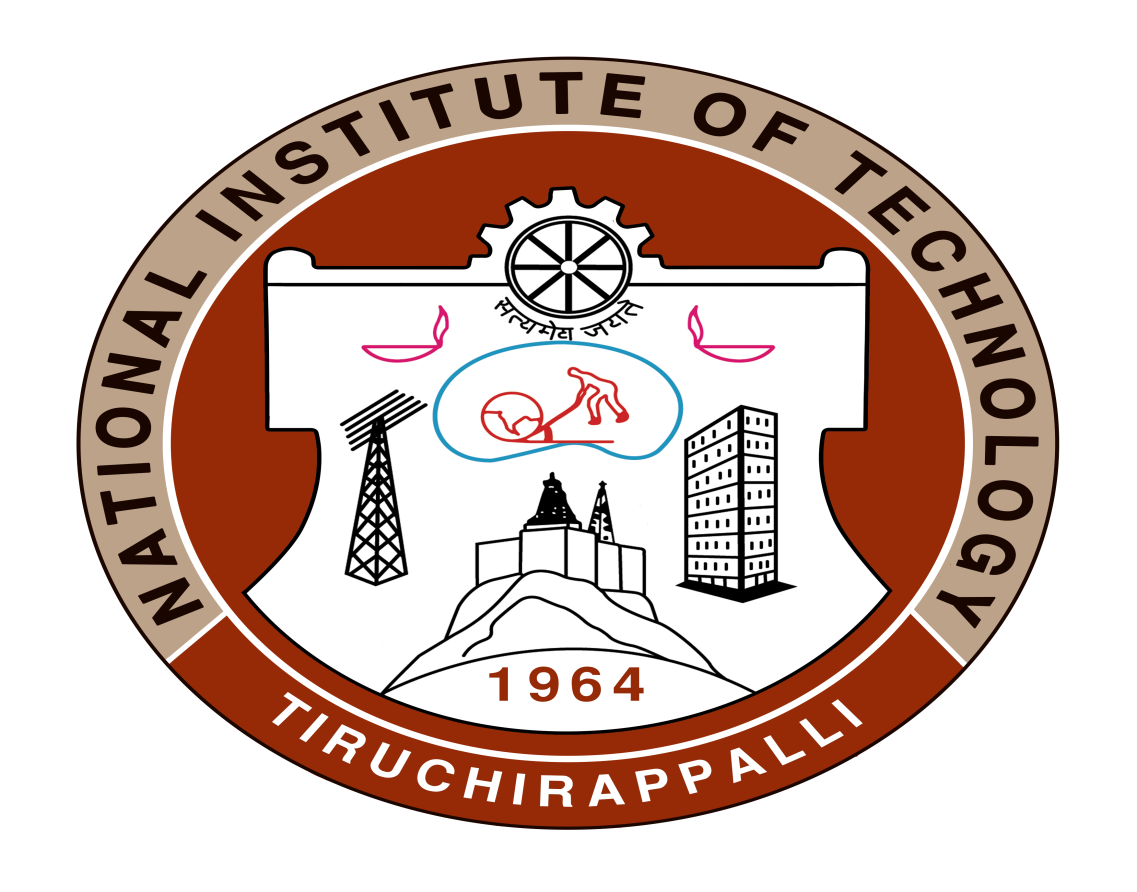
Department of Electronics and Communication Engineering

National Institute of Technology, Tiruchirappalli – 620 015



ECLR 11 – DIGITAL ELECTRONICS LABORATORY

PROFESSOR : Dr.R.Malmathanraj Sir

Mini Project Report

TITLE OF THE PROJECT:

DIGITAL MUSIC

(SIMULATION)

DONE BY :

S HARSHITHA - 108122105

SURUBOYINA HANVIK KUMAR - 108122121

THANDU SIRIVALLI - 108122125

Introduction:

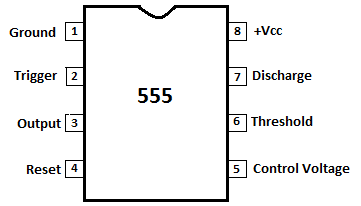
Creating a digital music device involves defining requirements for features like audio quality and user interface. Select a microcontroller for processing and audio components such as a DAC and amplifier for quality playback. Design a power supply system, considering portability with battery life and charging options. Choose a storage solution for storing music files and design a user interface, integrating buttons or touchscreens. Ensure seamless integration of hardware components for a cohesive and functional digital music device.

Components required:

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no | Apparatus required | Specification/Range | Qty |
| 1 | Power Supply | 0-5V | 1 |
| 2 | Breadboard | - |  |
| 3 | Resistors | 2K, 200, 400, 800, 1600, 3200, 6400, 12800 ohms | 36 |
| 4 | LED |  |  |
| 5 | Timer IC | IC 555 |  |
| 6 | Hexa Inverter gate | IC 7404 |  |
| 7 | Triple 3 input AND gate | IC 7411 |  |
| 8 | 4 Bit Synchronous binary counter | IC 74163 |  |
| 9 | Potentiometer | 25k ohm | 16 |
| 10 | Multi meter |  |  |
| 11 | Speaker |  |  |
| 12 | Capacitor |  |  |
| 13 | NPN Transistor(BJT) | BC107 | 1 |
| 14 | Potato battery |  |  |
| 15 | Dual 4 input AND gate |  |  |
| 16 | Breadboard small |  |  |
| 17 | Arduino Uno R3 |  |  |
| 18 | Push button |  |  |
| 19 | Connecting wires |  | As required |

IC 555 timer:

Pin diagram:



555 Timer IC is one of the most used IC in electronics, especially for triggering purpose. To learn more about it follow our various 555 Timer circuits. Here we are using 555 Timer IC in astable mode to create a beeping sound with Buzzer.

Pin 1. Ground: This pin should be connected to ground.

Pin 2. TRIGGER: Trigger pin is dragged from the negative input of comparator two. The Lower comparator output is connected to SET pin of flip-flop. A negative pulse (< Vcc/3) on this Pin sets the Flip flop and output goes High.

Pin 3. OUTPUT: This pin also has no special function. This is output pin where Load is connected. It can be used as source or sink and drive up to 200mA current.

Pin 4. Reset: There is a flip-flop in the timer chip. Reset pin is directly connected to MR (Master Reset) of the flip-flop. This is an active Low pin and normally connected to VCC for preventing accidental Reset.

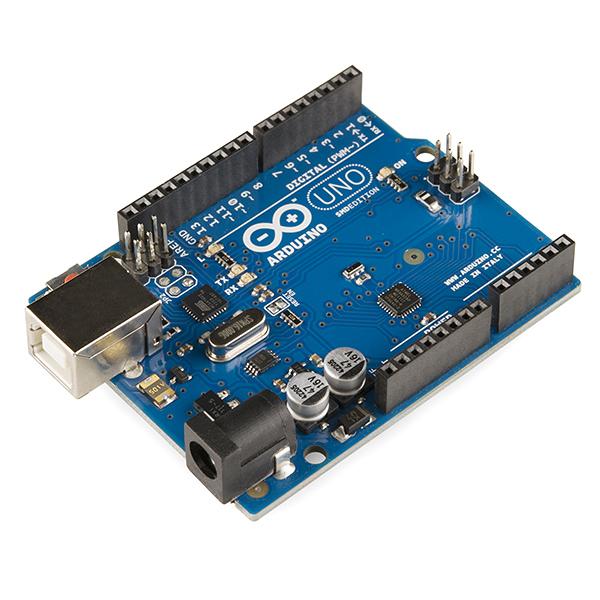
Pin 5. Control Pin: The control pin is connected from the negative input pin of comparator one. Output Pulse width can be controlled by applying voltage at this Pin, irrespective of RC network. Normally this pin is pulled down with a capacitor (0.01uF), to avoid unwanted noise interference with the working.

Pin 6. THRESHOLD: Threshold pin voltage determines when to reset the flip-flop in the timer. The threshold pin is drawn from positive input of upper comparator. If the control pin is open, then a voltage equal to or greater than VCC\*(2/3) will reset the flip-flop. So, the output goes low.

Pin 7. DISCHARGE: This pin is drawn from the open collector of transistor. Since the transistor (on which discharge pin got taken, Q1) got its base connected to Qbar. Whenever the output goes low or the flip-flop gets reset, the discharge pin is pulled to ground and capacitor discharges.

Pin 8. Power or VCC: It is connected to positive voltage (+3.6v to +9v).

Arduino Uno:



The **Arduino Uno** is an open-source microcontroller is equipped with sets of digital analog input/output. The board has 14 digital I/O pins,6 Analog I/O pins and is programmable with Arduino IDE.

### General pin functions

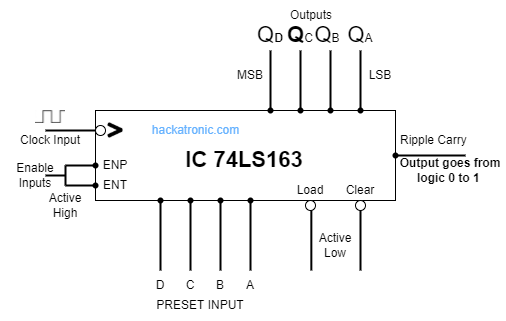
* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* **VIN**: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields that block the one on the board.

Some pins have specialized functions:

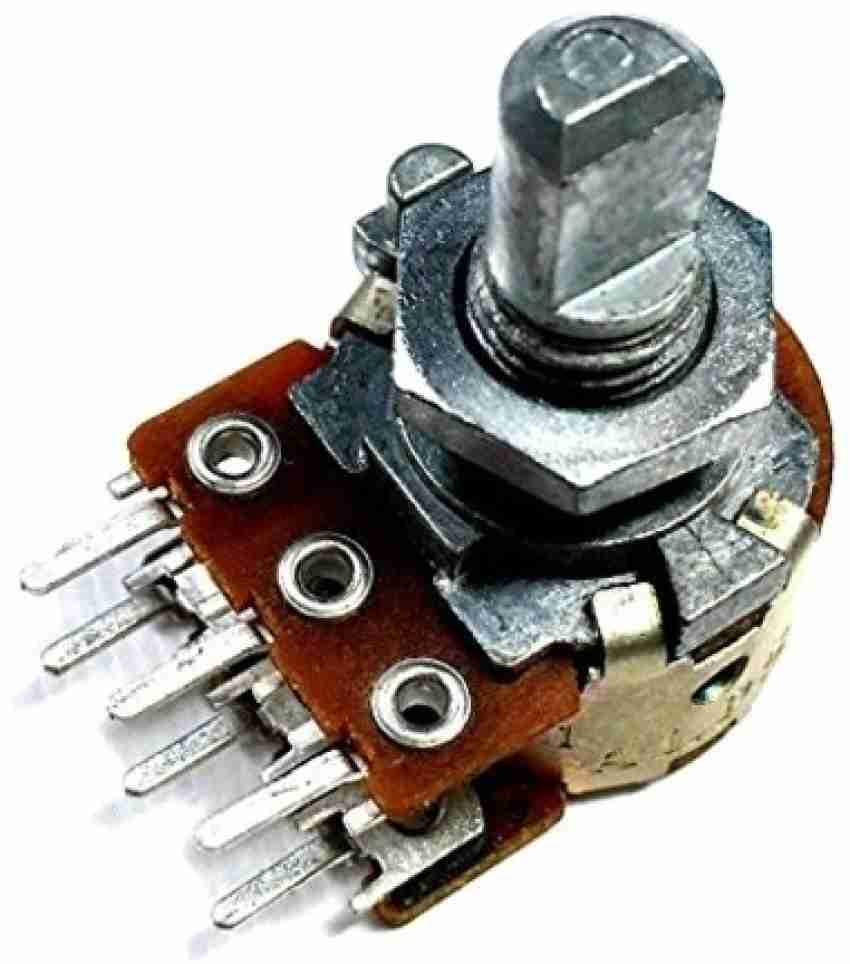
* **Serial** / [UART](https://en.wikipedia.org/wiki/UART): pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) [TTL serial](https://en.wikipedia.org/wiki/TTL_serial) data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* [**PWM**](https://en.wikipedia.org/wiki/Pulse-width_modulation) (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* [**SPI**](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.

4 bit synchronous counter:

This synchronous, presettable, 4-bit binary counter has internal carry look-ahead circuitry for use in high-speed counting designs. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable (ENP, ENT) inputs and internal gating. This mode of operation eliminates the output counting spikes that normally are associated with asynchronous (ripple-clock) counters. However, counting spikes can occur on the ripple-carry (RCO) output. A buffered clock (CLK) input triggers the four flip-flops on the rising (positive-going) edge of CLK. This counter is fully programmable. That is, it can be preset to any number between 0 and 15. Because presetting is synchronous, a low logic level at the load (LOAD) input disables the counter and causes the outputs to agree with the setup data after the next clock pulse, regardless of the levels of ENP and ENT.



Potentiometer:



A **potentiometer** is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a **variable resistor** or **rheostat**. Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment.

Multimeter:



A **multimeter** (also known as a **volt-ohm-milliammeter**, **volt-ohmmeter** or **VOM**) is a measuring instrument  that can measure multiple electrical properties. A typical multimeter can measure voltage, resistance and current , in which case can be used as a voltmeter, ammeter, ohmmeter. Some feature the measurement of additional properties such as temperature and capacitance.

Oscilloscope:



Project Steps:

1. Musical Note Representation: Define a binary representation for musical notes (e.g., C, D, E, F, G, A, B), their durations (quarter note, eighth note, etc.) and also their octaves.

2. Sequencer Interface: Create an interface for users to input musical notes and their durations using switches, buttons, or a digital keypad.

3. Sequencing Logic: Implement sequencing logic using counters. The counters determine the order in which the stored musical notes are played back. You can use different counters for different musical tracks.

4. Clock Timing: Use an oscillator or clock generator to provide precise timing for note playback. The clock signal controls the pace of the music.

5. Playback: When the sequencer is started, the counters will sequentially activate the flip-flops representing the notes and durations. Each activated flip-flop produces the corresponding musical note, and the timing creates the melody.

6. Audio Output: Connect the output of the flip-flops to a digital-to-analog converter (DAC) or a simple audio amplifier to produce audible music through a speaker or headphones.

7. Testing and Debugging: Test the sequencer with various musical compositions, ensuring that it accurately plays back the intended melodies.

8.Expansion: If desired, expand the project by adding more tracks, allowing for harmonies or polyphonic melodies.

Description and Working of the model:

The digital music generator is a device that generates artificial sound frequencies. It utilizes electrical signals and transforms them into audible sounds. The tone generator can create different sounds. However, this depends on what the application needs. Musical instruments, on the other hand, create simpler sounds—depending on the musical scale’s frequency. n modern testers, a modified DC is what supplies the electronic signal to the tone generator circuit. Plus, integrated circuits are responsible for modifying the DC. Additionally, there are different variations of tone generator circuits, including the sawtooth, triangle, sine, and square wave  generators. These periodic signals are capable of creating different sounds when connected to an audio transducer. The operation of this circuit is based on the working principle of a self-triggering oscillator ([astable multivibrator](https://circuits-diy.com/astable-multivibrator-mode-in-555-timer-ic/)), performed by a 555 precision timer circuit (**NE555**). When the circuit is powered on, The values of resistors (**R1**, **R2**) & capacitors (**C1**, **C2**) on the left side of the circuit set the pitch of the output tone coming from the audio transducer (loudspeaker), Including a variable resistor (potentiometer) serving as our pitch control.

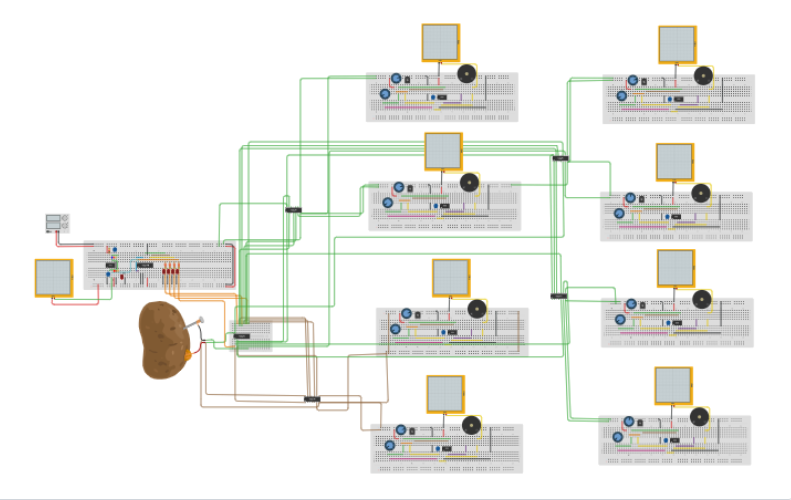
The capacitor (**C2**) to the far left filters out as much noise or undesired operation of the potentiometer due to which we get a smooth pitch change whilst adjusting. This circuit can be operated from as little as 6V to up to a 12 V power supply.

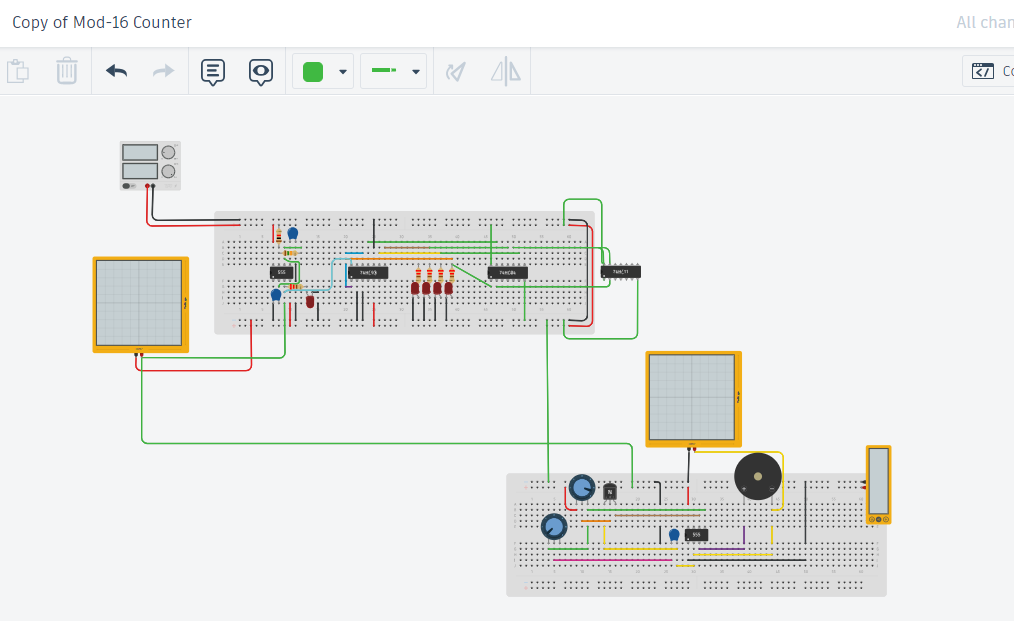
Use of Arduino uno as music generator:

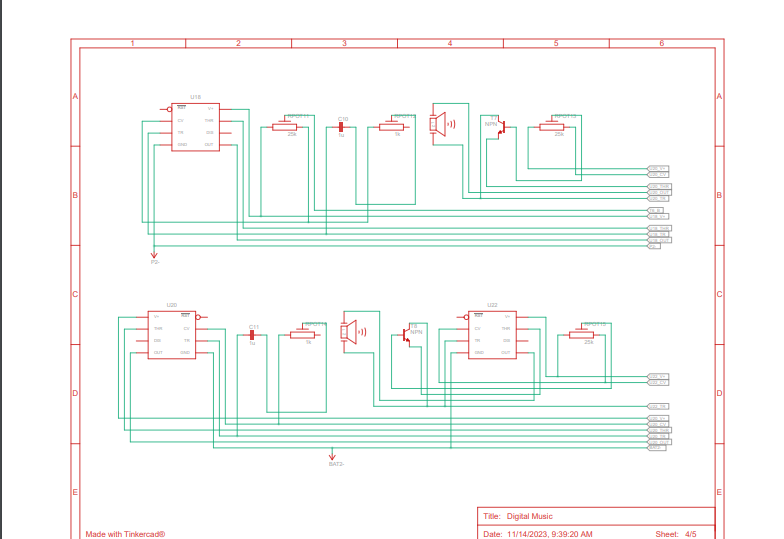
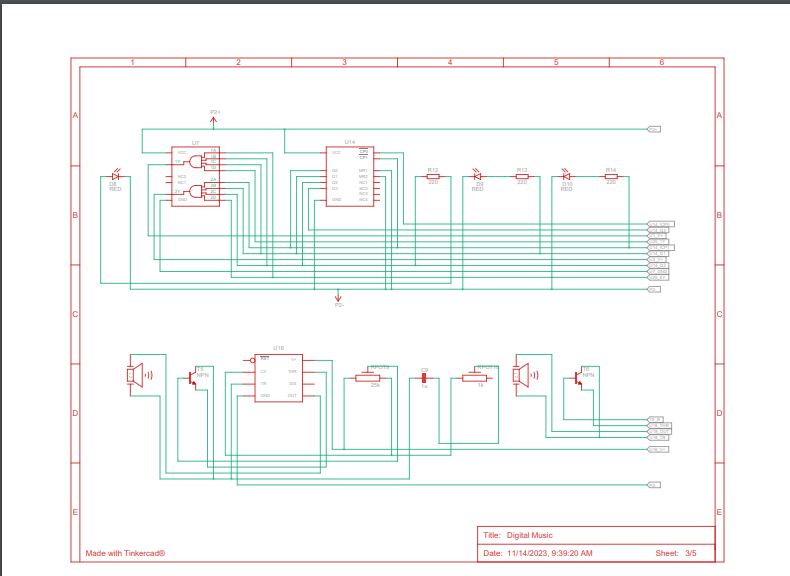
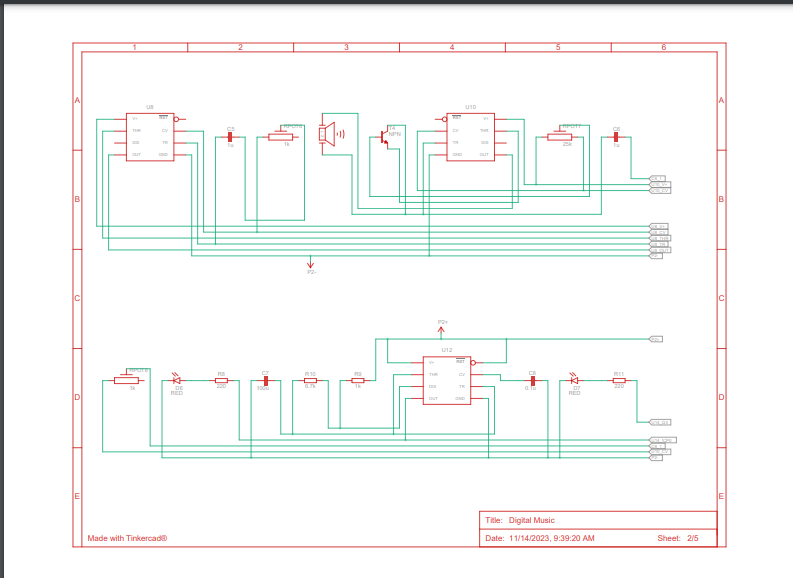
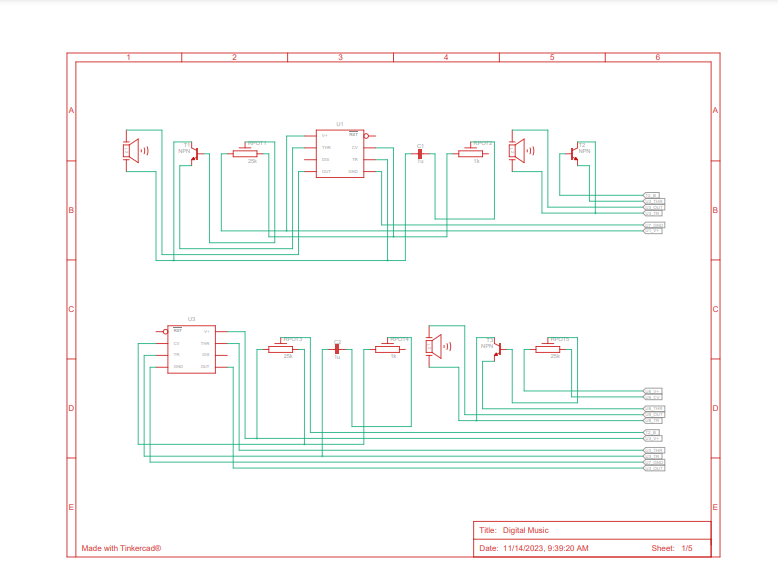
We can also use an Arduino uno to generate music tone in an easy manner. Here, we will turn the Arduino into an musical generator! This instrument have 2 function, 2 octaves usual and Practice mode, which user can practice hearing. This training are in the form of 3 type of question, Easy, Medium, and Hard. If the user presses one of these question buttons, the instrument will make sound of several notes, and the user must play the notes as they heard. In this project, Arduino use push button as the notes input through analog pin A3-A5. For the speaker, we will use Piezo buzzer in Tinkercad.

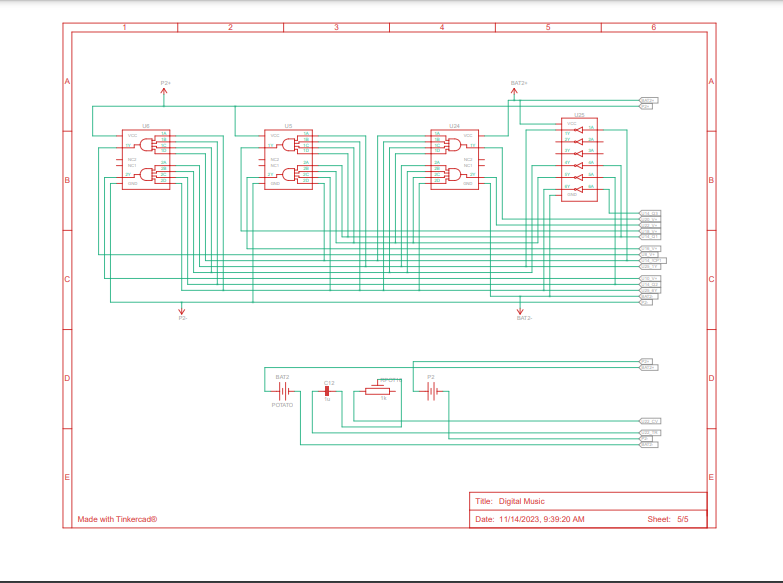
We have simulated the circuits in Tinkercad

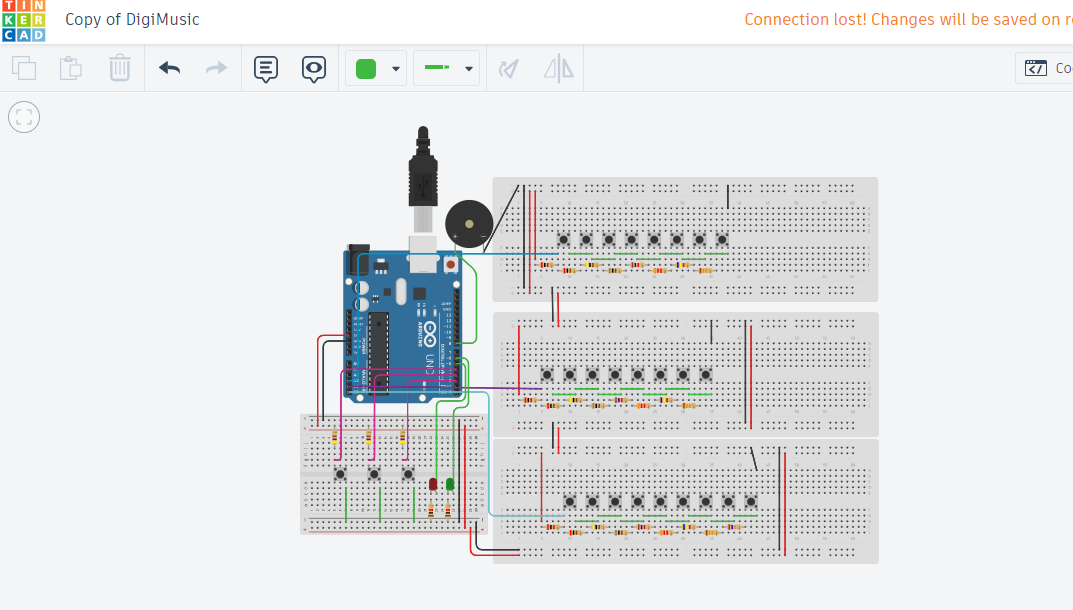
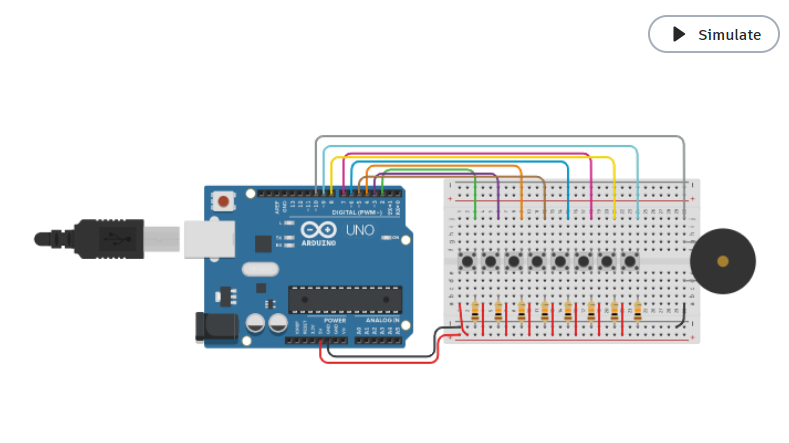
By using IC 555 timer:

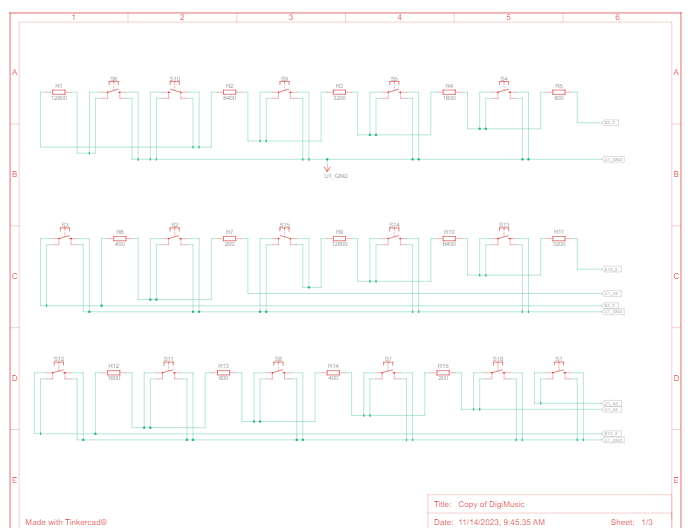


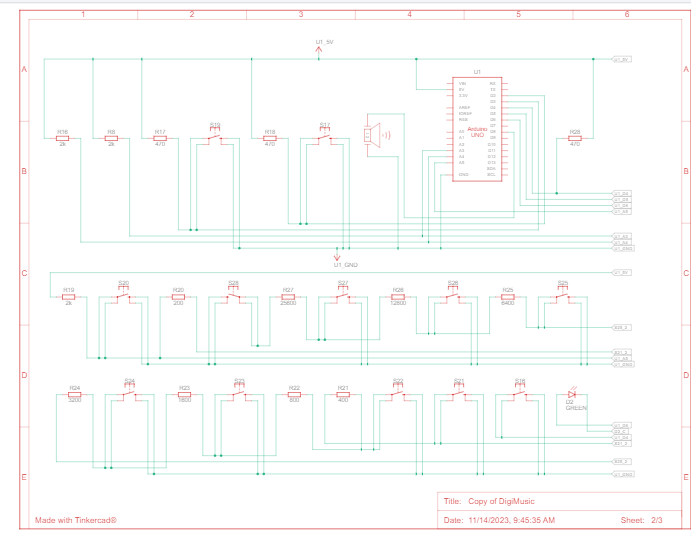












SIMULATION

<https://www.tinkercad.com/things/8PRRWpZVX7F-copy-of-digital-music/editel?sharecode=7HCl820fX2Et7ZdqmGjXAfTs8saym93GeKFaUQe1Ezw>

<https://www.tinkercad.com/things/lungAq41VwD-copy-of-digimusic/editel?sharecode=dR6bIBLjhywdUlu2-dNEQtnumLWzn9W9UmzbcW2bLHA>

Applications:

You can use circuit in the following applications:

Commonly used in home security setups such as burglar alarms & call bells.

Used to produce dial tone in telephonic devices.

Used to produce melodic tunes in devices such as toys & doorbells.

Conclusion:

Music generators are extremely versatile circuits. You can use them to test audio frequency equipment, create audio pulses, and even create sound in modern devices like mobile phones. In addition to their versatility, music generators work differently for different applications. Some types of generators include the 2 melody generator electronic circuits, ding-dong sound generator, two-transistor siren generator, and many more.

RESULT:

Thus, we designed a digital music sequencer using IC 555 timer and as well as using Arduino Uno R3 simulated using TinkerCad.