

Aim: Mini Piano

Objective: To explore the history, design, functionality, and cultural significance of mini pianos, highlighting their impact on music education, portability, and accessibility to aspiring musicians.

Components:

Arduino Uno: This will serve as the brain of your mini piano. It will process the input from the keys and generate the appropriate output to produce sound.

Breadboard: A breadboard will help you connect the various components without soldering, making it easy to prototype and modify your circuit.

Keys/Switches: You'll need buttons or switches to act as the keys of your piano. These will be the input devices that users press to play different notes.

Piezo Buzzer/Speaker: This will generate sound based on the signals sent from the Arduino. You can use a piezo buzzer for simple tones or a speaker for more complex sounds.

Resistors: Depending on the type of switches you use, you may need resistors to prevent issues like floating inputs or to create voltage dividers.

Wires/Jumpers: You'll need wires to connect the various components together on the breadboard and to connect the breadboard to the Arduino Uno.

Power Source: You can power your Arduino Uno via USB or an external power supply.

Setup : Connect buzzer and pushbuttons with Arduino Uno board.

Testing:

Connect the Hardware: Connect your Arduino Uno to your computer via USB.

Connect the buttons to the digital pins specified in the code (buttonPins[]).

Connect a speaker or buzzer to pin 8 (SPEAKER_PIN).

Upload the Code:

Open the Arduino IDE or any other compatible IDE.

Copy the code provided earlier into a new sketch.

Verify the code for any errors (click on the checkmark icon).

Upload the code to your Arduino Uno (click on the arrow icon).

Open Serial Monitor:

Once the code is uploaded, open the Serial Monitor in the Arduino IDE.

You should see the welcome messages and instructions printed in the Serial Monitor.

Test the Buttons:

Press each button one by one.

You should hear corresponding musical notes being played through the speaker.

Ensure that each button produces the correct note as specified in the code.

Test Volume Control:

If you have a volume control knob connected, turn it to adjust the volume.

Test whether turning the knob changes the volume of the sound produced by the speaker.

Play a Melody (Optional):

Modify the code to include a pre-programmed melody or sequence of notes.

Upload the modified code to your Arduino.

Press the "Play" button (if available) to start playing the melody.

Verify that the melody plays correctly and as expected.

Debugging:

If you encounter any issues during testing, check the connections of your hardware components.

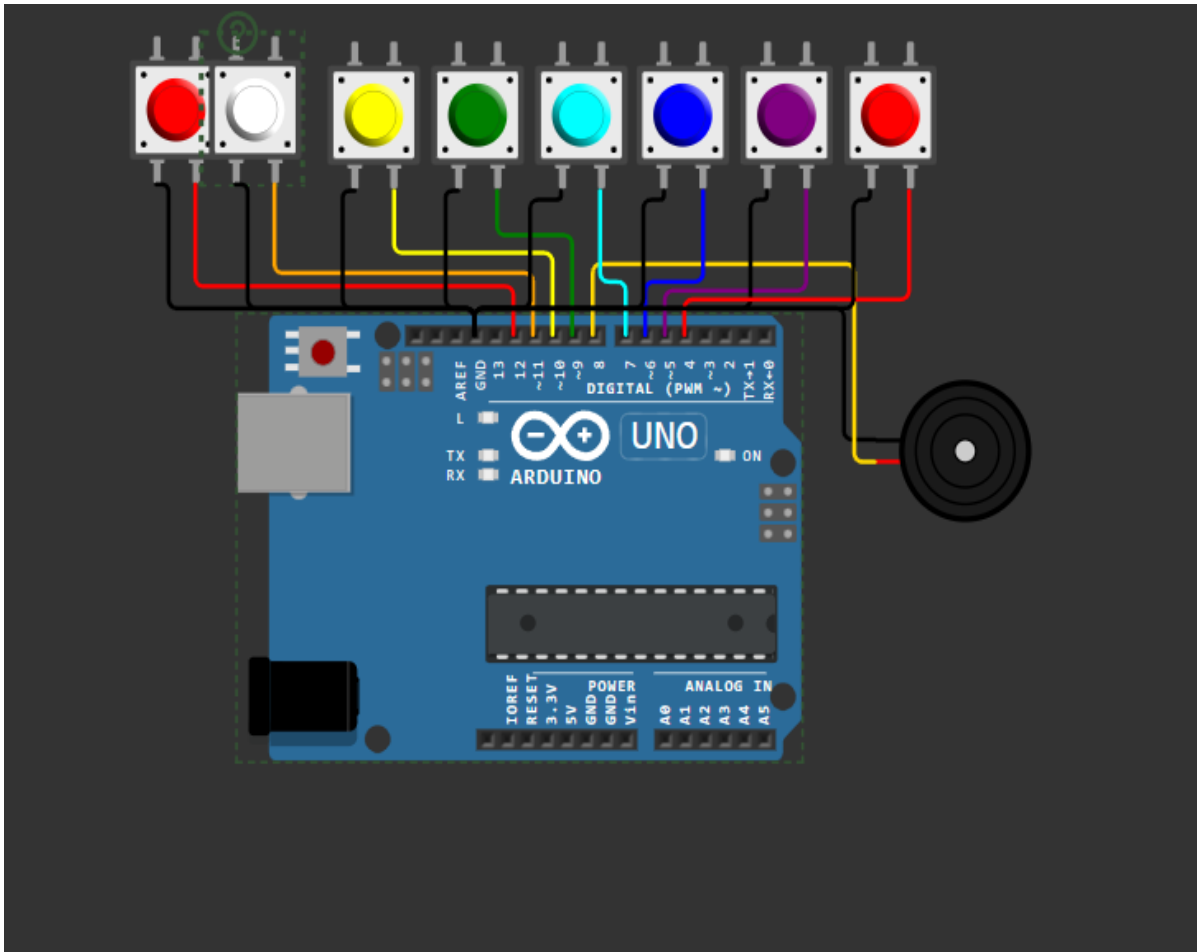
Use the Serial Monitor for debugging purposes to print debug messages or check the state of variables.

Enjoy Your Mini Piano:

Once everything works as expected, you can enjoy playing music on your mini piano!

By following these steps, you can effectively test your mini piano project and ensure that it functions as intended. Feel free to experiment with different melodies, button configurations, or additional features to customize your mini piano further.

DIAGRAM:



Observation:

This Arduino code facilitates the creation of a mini piano, employing push buttons to trigger various musical tones. The code begins with defining necessary pins and musical frequencies. It then proceeds to set up the pins, configuring them as inputs with pull-up resistors to ensure stable readings when buttons are not pressed. Additionally, the pin responsible for sound output is initialized.

In the main loop, the code continuously monitors each button. Upon detecting a button press (signaled by a LOW logic level), it assigns the corresponding tone frequency to a variable named pitch. Subsequently, if pitch holds a non-zero value (indicating a button press), the tone() function is invoked to generate sound at the specified frequency. Conversely, if no button is pressed, the noTone() function halts sound generation.

This setup allows users to produce different musical notes by pressing the corresponding buttons, making it a straightforward yet functional implementation for a basic electronic musical instrument using Arduino.