Arduino based Mini Piano

Components Required:

- **Arduino UNO R3**
- **Push Button**
- **Buzzer**

Buzzer:
A buzzer is an audio signaling device that converts electrical signals into sound. In this project, a piezoelectric buzzer is used to produce musical notes by generating square wave signals at specific frequencies using the tone() function. The frequency of the wave determines the pitch of the sound. When no button is pressed, the noTone() function stops the buzzer, ensuring that only the desired notes are played.

Fig 23.1: Buzzer

Push Button:
A push button is a simple mechanical switch that allows or interrupts the flow of current when pressed. In this project, each button is connected to an Arduino digital input pin with an internal pull-up resistor enabled, ensuring a default HIGH state. When pressed, the button completes the circuit and changes to a LOW state, triggering the corresponding musical note through the buzzer.

Fig22.2: Push Button

Project:

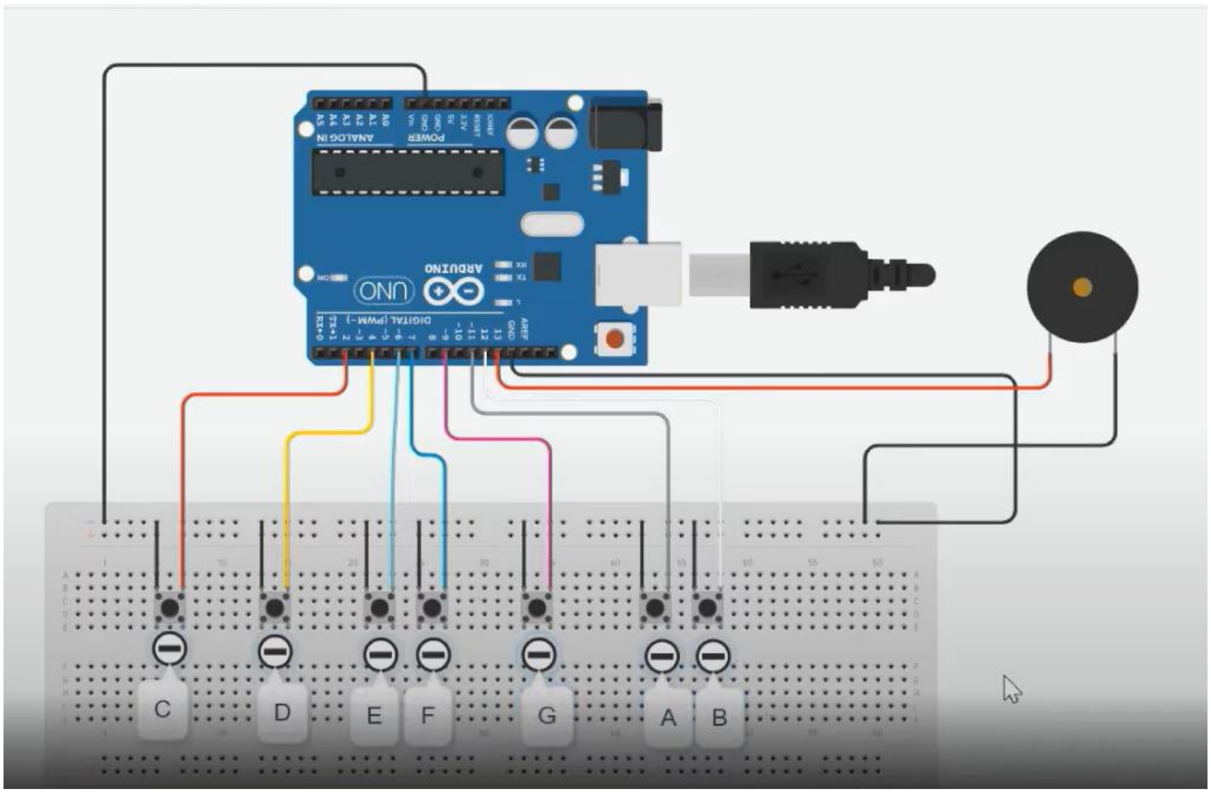

Fig23.3: Arduino based Piano

Explanation:

An Arduino-based piano is a simple yet effective project that allows users to play musical notes using push buttons. The system consists of a set of buttons, each corresponding to a musical note, and a buzzer that produces the sound. The implementation utilizes an Arduino microcontroller to detect button presses and generate the corresponding frequencies.

At the core of the system is a series of predefined tone frequencies that represent musical notes in a standard scale. The constants T_C, T_D, T_E, T_F, T_G, T_A, and T_B define the frequencies for the notes C, D, E, F, G, A, and B, respectively. These frequencies are essential for the tone() function, which generates square wave signals to drive the buzzer.

The hardware setup consists of seven push buttons, each connected to a dedicated digital input pin on the Arduino (pins 2 to 8). The buzzer is connected to pin 9, which serves as the output for generating sound. In the setup() function, all button pins are configured as inputs with internal pull-up resistors enabled. This ensures that each button remains in a HIGH state when not pressed and switches to LOW when pressed. The buzzer pin is set as an output to produce sound.

The loop() function continuously checks the state of each button using the digitalRead() function. When a button is pressed, the corresponding tone is played using the tone() function, which takes the buzzer pin and the note's frequency as parameters. If no buttons are pressed, the noTone() function stops any sound output. The if-else if structure ensures that only one note is played at a time, preventing overlapping sounds and ensuring smooth operation.

This Arduino-based piano effectively demonstrates fundamental concepts of embedded systems, such as digital input handling, signal generation, and basic circuit design. It can be further expanded with additional features, such as adding more notes, implementing LED indicators for visual feedback, or incorporating an LCD display to show the currently played note. With further modifications, the project can evolve into a more complex musical instrument, supporting polyphony or even MIDI integration.