Synthesis of Arduino Based Musical Instrument

NABANEETA BANERJEE[1], ARIJIT SENGUPTA[2], NISHA MUNDA[3], ARITRA DAS[4], CHANDRIMA BANERJEE[4], ARITRA SARKAR[5]

[1, 2, 3, 4, 5] *Department of Electronics & Communication Engineering, Guru Nanak Institute of Technology, Kolkata, India*

I. INTRODUCTION

A research paper is a synthesis of research, critical thinking, source evaluation, organization, and writing. It serves two purposes: to advance the body of knowledge in the subject field and to give us, the student, an opportunity to enhance our own knowledge. When we are in creative mind for a long and rainy afternoon with our Arduino board, the making of electronic piano is to be a project for those wanting to have immediate output once the setup is complete [1]. Best suited for Arduino users with some prior experience, our proposed project includes detailed instructions with a circuit diagram, as well as code and photos, so we've got a wide array of material to work with. If we get stuck, the creator and other makers have quite a lot of support and responses to help us address our issues. Arduino is programmed to convert hand-waving above 3D gesture Flick board to musical notes and then synthesizes music producing audio output at GPIO pin. Flick Large board is shipped with male/female ribbon proto-cable, but this cable is too long for use with Arduino Uno [2-3]. One solution is to cut excessive length to around 100 mm, reconnect and isolate like on image below. Other solution is to order shorter ribbon proto-cable.

II. COMPONENTS OF OUR PROJECT

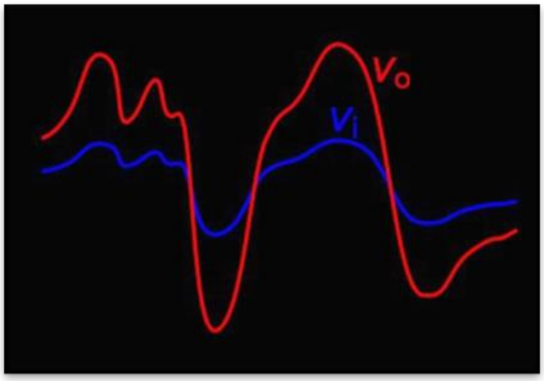
Serial number	Components name	list
01	Arduino Uno	1
02	Pi supply flick large	1
03	Pi supply flick large case	1
04	Capacitor (10Nf)	1
05	Resistor 4.7 k.ohm	1
06	Rca jack/plug cable	1
07	Usb type a to type b cable	1
08	amplifier	1
09	Bread board	1

a) Arduino UNO R3

Here, we have used Arduino UNO R3 (Fig.1) as microcontroller which has 20 input/output digital pins of which 6 are PWM outputs and 6 are analog inputs. We have loaded program into it by computer software. Arduino will control the full system. It will recieve signal from mobile application via bluetooth and the floor cleaning machine will work when we require.

b) Amplifier

An amplifier, electronic amplifier or (informally) amp is an electronic device that can increase the power of a signal (a time-varying voltage or current). It is a two-port electronic circuit that uses electric power from a power supply to increase the amplitude of a signal applied to its input terminals, producing a proportionally greater amplitude signal at its output. The amount of amplification provided by an amplifier is measured by its gain: the ratio of output voltage, current, or power to input. An amplifier is a circuit that has a power gain greater than one.

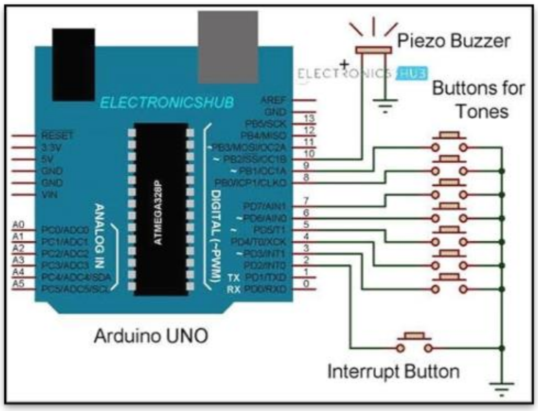


c) Pi supply flick large:

Flick Large is a 3D tracking and gesture board that lets you control your I2C enabled devices as if by magic. Flick uses technology that enables the PCB to detect your gestures from up to 15cm away in 3D space. Draw a circle, swipe your hand - Flick will track it.

III. CIRCUIT DIAGRAM OF PIANO USING ARDUINO-UNO

In the circuit diagram we have connected a piezo buzzer with the pin number 10 of Arduino-uno. According to the pin diagram, we have connected the interrupt button with the 2 number port of Arduino-uno pin. When we play the buttons the voltage converts into sound energy and we get the output as a sound.



CIRCUIT DIAGAM OF PIANO USING ARDUINO UNO

IV. EXPERIMENTAL DISCUSSIONS

Here we shall discuss about the set-up of our project. We can classify this part into some parts:

i. How to set up the hardware

- The piano keys:

The piano key switches were connected using a resistor ladder. The resistors are placed in sequential order, connecting each switch to power. For my project, I chose to use (in ohms) 1M, 10k, 4.7K, 1K, 560, and 220. The first piano switch (on the left) is connected to power, and also to the next switch in the circuit[2]. The second switch is connected to the smallest resistor, and each switch after that connects to the next largest resistor in the series. The last switch grounds the connection via a 10k ohm resistor and also connects all 7 switches to the "analog in" pin A0 on the Arduino. The main sketch defines which music note frequencies are associated with each piano key. For this project, I used C4, D4, E4, F4, G4, A4, & B4, with C4 being the leftmost switch. Change the frequency values, or add additional switches to fully customize your own project!

- The LCD screen:

Connect the LCD pins according to the circuit diagram below. Connect LCD pin 3 to the Potentiometer for contrast control of the screen. The outside pins of the Potentiometer are connected to power and ground to complete this circuit. LCD pin 15 is connected to power via a 220 ohm resistor. If any of these pins change on your own project, update the code accordingly.

- The menu switch:

Connect the menu switch to power, and ground it through a 10K ohm resistor. This switch should also be connected to pin 7 on the Arduino so you can control it via coding.

- The LED:

The LED is used to indicate when you're in "menu mode", and the piano keys have been disabled. The LED light turns on once the menu switch is pressed. Connect the anode (positive) of the LED to pin 6 of the Arduino, and ground the LED through a 220 ohm resistor.

- The Piezo buzzer:

To integrate the Piezo Buzzer, simply connect one end to the Arduino pin 8, and the other side to ground via a 220 ohm resistor. The "tone" function in the code will look for this pin in order to play the note called.

ii. How does the code work

The set up:

- The sketch begins by importing the "Liquid Crystal. h" and "pitche S.H" libraries so we can reference various items from them later on in the code
- Next, the code is set up by defining a name for the menu switch, determining which pins are inputs vs. outputs, and setting the frequency values for each of the 7 piano keys
- A new tab was created for each individual song in an effort to keep my code somewhat organized and easier to understand, and to expand on later. If you'd like to delete or add more songs, just make sure to change the code in the main "Electronic Keyboard" sketch as well [3].

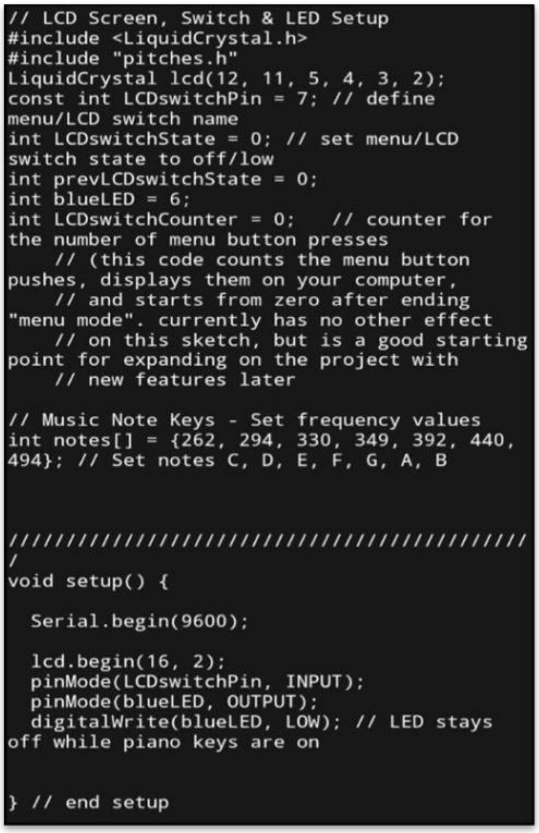
The main loop:

- The keyboard begins with the LED turned off, and all of the piano keys active
- Once the loop detects that the menu button has been pushed, it will disable the piano keys and turn the LED on to indicate you are in "menu mode"

Accessing the menu:

- Push the menu button to cycle through the present songs, and push any of the piano keys to start playing the song displayed on the LCD screen
- The LCD screen will show which song is currently playing, then return to the same point in the menu once the song has finished playing. From there, you can either replay or continue in the list of available songs.
- Double-clicking the menu button will restart the menu from the beginning
- To return to using the piano keys, cycle through the remainder of songs in the menu. Once the LCD screen in blank, and the LED is off, the piano keys will work again.

V. CODE DIAGRAM



```
// LCD Screen, Switch & LED Setup
#include <LiquidCrystal.h>
#include "pitches.h"
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int LCDswitchPin = 7; // define
menu/LCD switch name
int LCDswitchState = 0; // set menu/LCD
switch state to off/low
int prevLCDswitchState = 0;
int blueLED = 6;
int LCDswitchCounter = 0;   // counter for
the number of menu button presses
    // (this code counts the menu button
pushes, displays them on your computer,
    // and starts from zero after ending
"menu mode". currently has no other effect
    // on this sketch, but is a good starting
point for expanding on the project with
    // new features later

// Music Note Keys - Set frequency values
int notes[] = {262, 294, 330, 349, 392, 440,
494}; // Set notes C, D, E, F, G, A, B



/////////////////////////////////////////////
/
void setup() {

  Serial.begin(9600);

  lcd.begin(16, 2);
  pinMode(LCDswitchPin, INPUT);
  pinMode(blueLED, OUTPUT);
  digitalWrite(blueLED, LOW); // LED stays
off while piano keys are on


} // end setup
```

VI. ADVANTAGES OF OUR PROJECT

- Due to low distortion rate the output sound will be noiseless (accuracy level 97.78%)
- Our project is lower costly than the market cost

VII. PURPOSE OF OUR PROJECT

- As we are using here Arduino Uno there will be no electricity problem during, we play this piano...
- It can be accessed by our mobile phone also.
- It is easily portable for its size.
- We can reduce the noise and distortion level of the piano as we are using a Arduino uno and amplifier having good efficiency.

REFERENCES

[1] https://www.hackster.io/lanmiLab/make-musical-instrument-using-arduino-and-flick-large-e2890b

[2] https://www.instructables.com/How-to-Make-a-Arduino-HID-Keyboard/

[3] https://create.arduino.cc/projecthub/projects/tags/keyboard