Angelina Melnychuk

Student ID: 200330873

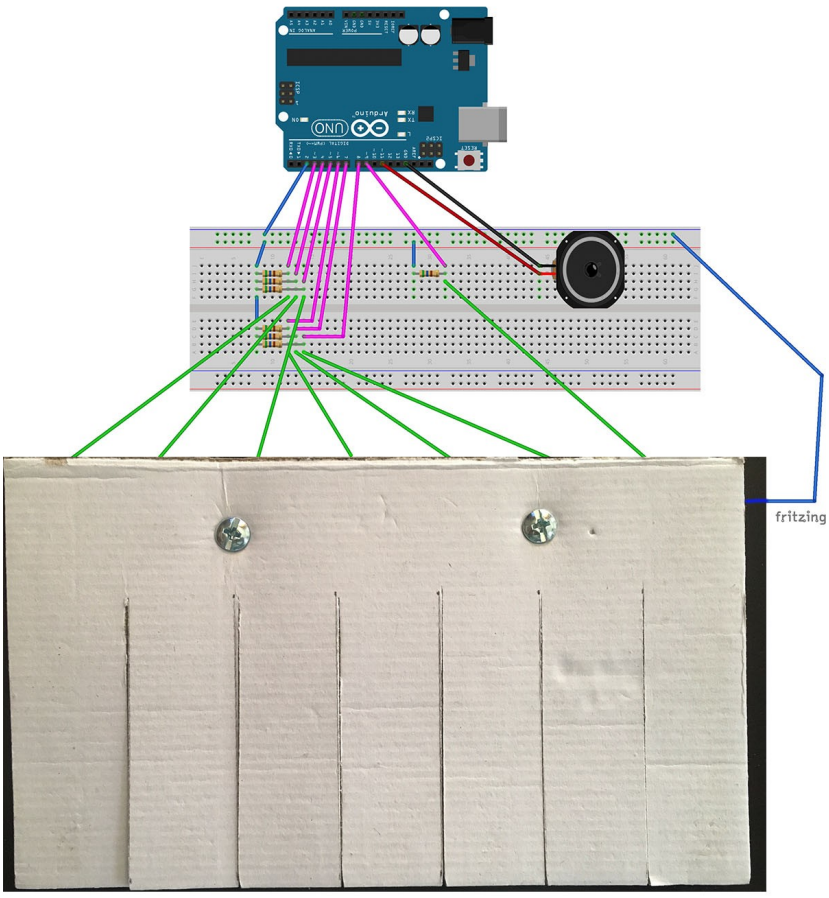
CS 207 – Building Interactive Gadgets

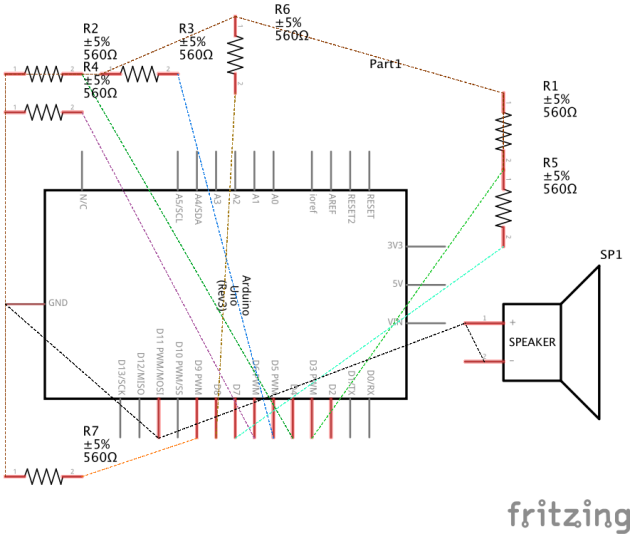
Professor Travor

04 December 2018

Conductive Fabric Piano

Circuit designs (breadboard and schematic)[1]





Components:

- Arduino Uno
- Resistors (7 for each keynote- 560 Ohm)
- Jumper wires
- Breadboard
- Speaker 0.25W 8 ohms
- Conductive fabric and cardboard
- Scissors
- Arduino program
- <CapacitiveSensor.h> Library.
- <pitches.h> Library

Introduction

Do you have tension to the music and instruments but do not have musical grammar background? Imagine that you can play piano even without knowing notes. Using Arduino, a speaker, a handful of resistors, and pieces of conductive fabric, you can assemble your own piano in just a few steps. On the Arduino Uno you get a device with 18 keys, and on the Arduino Mega as much as 60! The original project was done by Ikhsan Ismail [2] on hackster.io.

This project was chosen to demonstrate to the audience that you do not need to be a musician to perform the music piece. Also, with the conductive fabric piano project, I try to convey the idea of simplicity in terms of music in order to demonstrate that computer science (Arduino Uno board) can be used as a part of such media as music.

The initiate build/circuit

This project uses 7 resistors each of 560 Ohm to represent 7 keynotes on the piano board. Piano board is a combination of cardboard and conductive fabric. Lower board has the long rectangular strip placed on double tape whereas upper board is a combination of seven strips placed accordingly to represent keynotes. Each resistor connects both to the pin number (from pin 3 to 9) on a Arduino Uno board and to the actual key note. For instance, pin 3 is connected to the first resistor on the breadboard and to the first keynote on the piano board. This means that if we have a board with 20 pins, 14 of which are digital and 6 analog, we can create a piano with 20 keys. If you have an Arduino Mega or another board with a large number of outputs, you can safely make more keys.

Pin 2 responsible to send Capacitive Sensor library to the Arduino. This project uses the idea of Capacitive sensing a technology that measure and detect the conductive material. Conductive fabric is commonly used material in the world of DIY projects. Conductive fabric is a fabric, the conductance of which is provided by the thread, containing fibres with micro metalled particles.

How does it work

Over the past decade or so, it has become really difficult to imagine a world of electronics without touch sensors. Smartphones are the most visible and common example, but, of course, there are many other devices and systems that have touch sensors. To build touch sensors, both capacitance and resistance can be used.

The sensor keyboard and fingers form a capacitor that stores charge. The amount of charge that can be stored depends on the value of the capacitance. The closer finger to the board, more capacitance receives. So why does the presence of a finger change capacity? There are two reasons: the first includes the dielectric properties of the finger, and the second includes its conductive properties.

However, it should be mention that that this method is not suitable for measuring the absolute distance between the sensor and the finger; Capacitive sensors do not provide the data necessary to perform accurate calculations of absolute distances. Then, it follows that this technology is particularly suitable for detecting changes in distances, when a finger approaches or moves away from the sensor.

The original project uses 1mOhm resistors, however, for my project I have chosen 560 Ohm resistors. First of all, this type of resistor is used because we have them in Arduino kit. Secondly, resistor with less resistance actives absolute and precise touch that makes each press on the piano key to respond simultaneously. For instance, if you have decided to use 1mOhm resistors for the project or more, the keyboard might respond couple cm. away as becomes to sensitive that makes the whole process of playing meaningless.

Capacitive Sensor Library

In order to make the conductive material works, we need to download Capacitive Sensor Library that will do almost all work for us, for instance sets the Send Pin and Receive Pin. Once the library is downloaded, the sketch code will indicate at the beginning #include <CapacitiveSensor.h>.

Physical hardware designs

7 resistors are placed on the breadboard in a such way that it will be able to insert wires that both both are connected to the board and to the Arduino's pins. The first resister is connected with wire to ground. The third and fourth resistors are connected with wire to continue the circuit. The last seventh resistor is also grounded. Each resistor joins with the pin number on the Arduino Uno board. One pin (in this case pin number 2) is used to send a signal (the pin state simply changes from 0 to 1 or vice versa), and the second pin is used to register the altered state. The detection of the contact of the conductive fabric occurs due to the fact that the reception time of the sent signal depends on the

capacitance in the circuit, which, in turn, depends on the touch of the conductive fabric. The keyboard is also grounded by yellow wire (as shown in the video) to the breadboard.

How the code works

The sketch begins by importing the <CapacitiveSensor.h> libraries and <pitches.h> Library. The speaker is defined through pin 11. Capacitive Sensor sets the Send Pin (pin 2) and Receive Pins (from 3 to 9). When hand touches the sensor, the speaker produces a tone. For this project the sensitivity of 200 was chosen, to make the board less sensitive while touching the keyboard. In void loop if statement defines the sensitivity and tone that is measured in numbers. Each frequency tone corresponds to the specific note. For this project such frequencies were chosen:

- 104-NOTE_GS2
- 233-NOTE_AS3
- 262-NOTE_C4
- 311-NOTE_DS4
- 370-NOTE_FS4
- 440NOTE_A4

Brett Hagman [3] has created this preset of pitch frequencies that simplified the whole process of setting the appropriate value to the tone that send this or that frequency to the speaker. It also should be mention that frequency values are represented in Hz. As s result, it is possible to modify tone in range from 31 to 4978. The if function either send the signal when the sensitivity is less or equal to 200 once the one of seven keynotes is pressed or, when hand didn't touch on it, no tone is produced. Arbitrary delay 10 is declared to limit data to serial port.

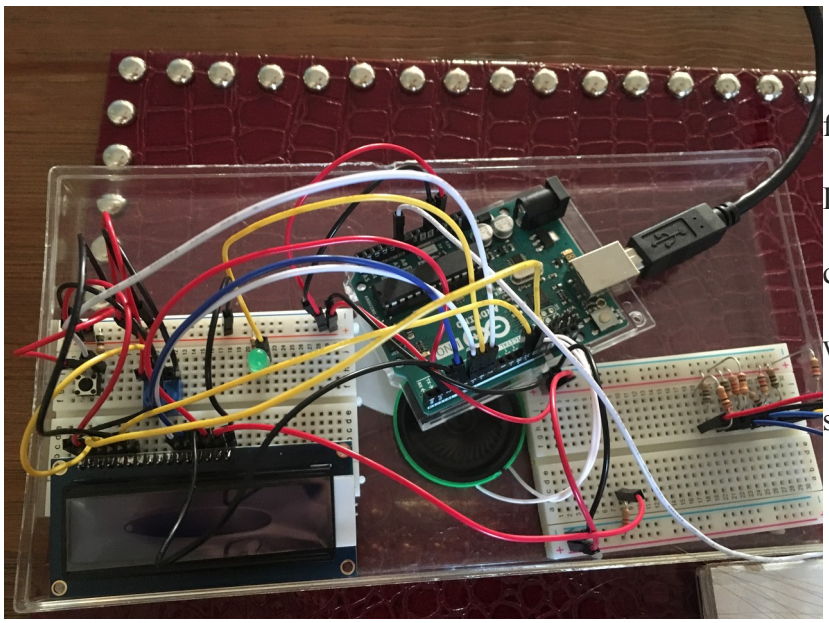
Code

```
 #include <pitches.h>

// Import the CapacitiveSensor Library.

#include <CapacitiveSensor.h>

// Name the pin as led.

#define speaker 11

// Set the Send Pin & Receive Pin.

CapacitiveSensor   cs_2_3 = CapacitiveSensor(2,3);        // 560 OhM resistor

CapacitiveSensor   cs_2_4 = CapacitiveSensor(2,4);        // 560 OhM resistor

CapacitiveSensor   cs_2_5 = CapacitiveSensor(2,5);        // 560 OhM resistor

CapacitiveSensor   cs_2_6 = CapacitiveSensor(2,6);        // 560 OhM resistor

CapacitiveSensor   cs_2_7 = CapacitiveSensor(2,7);        // 560 OhM resistor

CapacitiveSensor   cs_2_8 = CapacitiveSensor(2,8);        // 560 OhM resistor

CapacitiveSensor   cs_2_9 = CapacitiveSensor(2,9);        // 560 OhM resistor

void setup()

{

  cs_2_3.set_CS_AutocaL_Millis(0xFFFFFFFF);     // turn off autocalibrate on channel 1 - just as an

example

  // Arduino start communicate with computer.

  Serial.begin(9600);

}

void loop()

{

  // Set a timer.
```

```
long start = millis();
// Set the sensitivity of the sensors.
long total1 =  cs_2_3.capacitiveSensor(200);
long total2 =  cs_2_4.capacitiveSensor(200);
long total3 =  cs_2_5.capacitiveSensor(200);
long total4 =  cs_2_6.capacitiveSensor(200);
long total5 =  cs_2_7.capacitiveSensor(200);
long total6 =  cs_2_8.capacitiveSensor(200);
long total7 =  cs_2_9.capacitiveSensor(200);
Serial.print(millis() - start);        // check on performance in milliseconds
Serial.print("\t");                    // tab character for debug windown spacing
Serial.print(total1);                  // print sensor output 1
Serial.print("\t");                    // Leave some space before print the next output
Serial.print(total2);                  // print sensor output 2
Serial.print("\t");                    // Leave some space before print the next output
Serial.print(total3);                  // print sensor output 3
Serial.print("\t");                    // Leave some space before print the next output
Serial.print(total4);                  // print sensor output 4
Serial.print("\t");                    // Leave some space before print the next output
Serial.print(total5);                  // print sensor output 5
Serial.print("\t");                    // Leave some space before print the next output
Serial.print(total6);                  // print sensor output 6
Serial.print("\t");                    // Leave some space before print the next output
Serial.println(total7);                // print sensor output 7
```

```
                                    // "println" - "ln" represent as "line", system will jump to next line after print
the output.
   // When hand is touched the sensor, the speaker will produce a tone.
  // I set a threshold for it, so that the sensor won't be too sensitive.
  if (total1 > 200) tone(speaker,104);   //NOTE_GS2
  if (total3 > 200) tone(speaker,233);   //NOTE_AS3
  if (total4 > 200) tone(speaker,262);   //NOTE_C4
  if (total5 > 200) tone(speaker,311);   //NOTE_DS4
  if (total6 > 200) tone(speaker,370);   //NOTE_FS4
  if (total7 > 200) tone(speaker,440);   //NOTE_A4
  // When hand didn't touch on it, no tone is produced.
  if (total1<=200  &&  total2<=200  &&  total3<=200 && total4<=200  &&  total5<=200  &&
total6<=200 &&  total7<=200)
   noTone(speaker);
  delay(10);                          // arbitrary delay to limit data to serial port
}
```

Discussion of milestones



Initially, the aim was to create conductive fabric piano board[2]with preset songs [4], LCD screen that will show the song chosen, and green LED with button that will be on once the button is pressed to show that LCD is also on. The milestone of

parts order was complete even earlier as I ordered the speaker and LCD from Adafruit. I did not have any problems with the circuit build and assemble parts also in the scheduled time. I tried to use resistors in the kit, as a result, combined them to represent different pitches. However, the whole idea collapsed when the code did not work successfully. The only thing that had made working was the LED, as seen above on the picture and speaker made weird noses. As my main was to present to the audience the actual project that works and offer an ability of the interactivity, I decided to simplify my project just to the conductive fabric piano.

Conclusion

In conclusion, I can say that although the initial project has failed, I can bring it to life in the future and create my personal list of preset songs with the help of Brett Hagman's tone table. However, I also satisfied with finished project because I have learn a lot about additional libraries in Arduino, examine the capacitive sensor technology, and experienced an exclusive opportunity of interactivity as the actual board with limited 7 keyboards could be pressed in order to perform the music.

References

[1]"Fritzing." Fritzing Fritzing, fritzing.org/home/.

[2] Ismail, Ikhsan. "Paper Piano with Arduino." Hackster.io, 16 Oct. 2018,

[3]Arduino - Introduction, www.arduino.cc/en/Tutorial/toneMelody.

[4] www.hackster.io/San_Ismail/paper-piano-with-arduino-e27da7.Arduino Play Melody, www.arduino.cc/en/Tutorial/PlayMelody.