**CS209-210 Mini Project Report**

By Aadit Sharma (2101AI02)

# Overview

I teamed up with Archit to build a smart piano for this CS209-210 mini project. The piano works in three modes- free play where the user can play any song he/she wants to, Learn to play mode where the user can learn to play one of the 5 songs implemented and Auto Play mode where the Arduino plays the song selected. An Android App is used to control the behaviour of the piano. For detecting the key pressed by the user capacitive sensor is created using graphite on paper. Graphite being a conductor is used to figure out when a particular key is pressed and the sound is produced by a buzzer. A HC-05 module is used to maintain a Bluetooth connection with the android app.

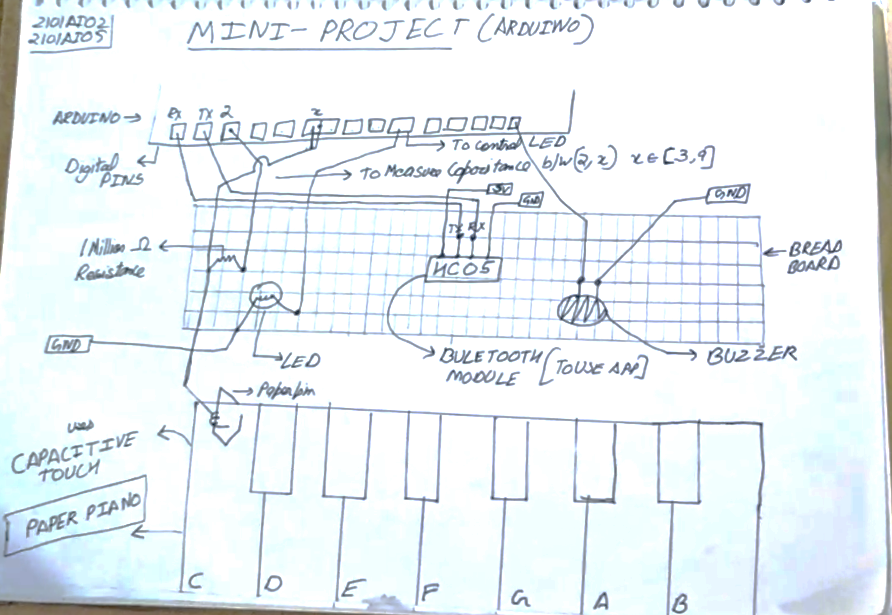
# My Contribution

1. Circuit Design- The circuit was completed and was designed by me. The circuit consists of 7 LEDs, 7 paper clips, jumper wires, a HC05 Bluetooth module and one buzzer. The connection of the various components and their inner workings will be explained in detail in the upcoming sections.
2. Arduino Code- For the Arduino code, I implemented Learn to Play and Auto Play modes. The notes for the five songs were searched from various sources which are mentioned at the end.

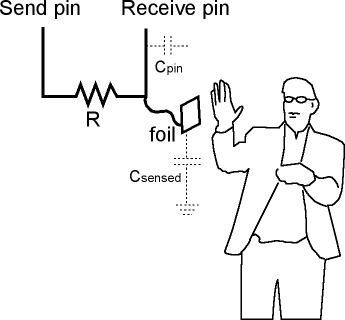
The next parts of this report elaborate the above sections.

# The Circuit

A schematic representing the diagram is below-



A brief overview of different components and connections is given below-

* **HC05 Bluetooth Module**- The RX pin of HC05 is connected to TX pin of Arduino and the TX pin of Arduino is connected to RX pin of Arduino. The VCC of the pin is connected to 5V and ground is connected to Arduino’s ground. The module communicates with the Arduino at 9600 bits/second and it is capable of receiving only 8-bit characters. We use it to send a character to the Arduino where further processing is done.
* **LEDs**- LEDs are connected from Digital pin 10 to 12 and Analog pin A0 to A3, these light up to indicate the current or future note to play.
* **Resistors**- One Million Ohm Resistors are connected with the capacitive touch sensors to decrease the sensitivity of the sensor.
* **Capacitive Touch**- The sensor is made of a paper covered with graphite. When a human touch is sensed, it gives a higher reading than usual and this is exploited for triggering various function calls. A high value resistor is used to decrease the sensitivity of the sensor. The adjoining diagram explains its functioning. The receive pin is connected to digital pin 3 to pin 9 for the 7 sensors, while the send pin of all the 7 sensors are connected to Pin 2
* **Piezo buzzer**- This is connected to digital pin 13 and is used to play tones at specified frequencies. The Arduino sends a square wave to pin 13 to produce sound of the specified frequency. Since we have control only over sound, the result is not well produced.

# Arduino Code

My contribution included implementing auto play and learn to play mode. For each of the 5 songs implemented, the loop function calls appropriate functions to play the song. For the purpose of this report I will explain some of the functions called in doing so-

### findKeyPressed()

This function is called to return the key currently pressed by the user by reading the capacitive sensors’ values.

### lightLed and turnoffLed

This function takes in character of note pressed and sets the relevant LED to the correct state.

### autoPlay

This function takes in character of note to play the sound, the octave to play the note at and the delay in milliseconds after which to stop the sound.

### aiPlaySong{x}

The sequence of notes to play have been added as character array in the program. When such a function is called, the CPU loops through the relevant loop and plays the relevant sound by calling auto play function, switches the relevant LED on and off by invoking lightLed and turnoffLed function. After the loop is terminated, the program returns to default state of free play.

### song\_{x}

This function is called to play song ‘x’ in tutorial mode, since the correct sequence of notes is known, the function loops through relevant loops lighting up LEDs corresponding to the note and if the same key is pressed, the board then reads the next note and so on until the array is exhausted at which point, the program returns to default state of free play. If an incorrect key is pressed, the piano stays in that loop.

# Sources

The notes for the songs were taken from the following sources-

1. Twinkle Twinkle Little Star- <https://www.letsplaykidsmusic.com/twinkle-twinkle-little-star-easy-piano-music/>
2. Jingle Bells- <https://www.notationsworld.com/jingle-bells-piano-notes.html>
3. Happy Birthday- <https://www.synthesizernotes.com/happy-birthday-to-you-piano.html>
4. The Scientist- <https://www.pianomint.com/english-hits/the-scientist-coldplay-piano-notes/>
5. Old Nokia Ringtone- <https://jeeteasypiano.blogspot.com/2018/06/nokia-tune-piano-notes.html>

Some parts of the songs were edited and/or shortened for the project.