Piano using Arduino Uno, Bluetooth Module, and Android app

200453544 - Ravikumar Patel, *Computer Science dept, University of Regina*

***Abstract*—This report outlines the modifications made to an existing project. The existing project was modified by using a Bluetooth module to transmit input to Arduino Uno instead of Push buttons, Keypad, or Paper piano. Arduino Uno, Piezo speaker, Bluetooth module, and Android app were used to complete this project. The Bluetooth module is only used as a slave in this project as it only receives the data from the android app. The Designed piano was able to play 7 notes ('A", 'B', 'C', 'D', 'E', 'F', and 'G').**

***Index Terms*—Arduino Uno, Bluetooth module, Piezo speaker, Android app**

I. Introduction

THE use of digital media and devices has increased nowadays to gain knowledge and learn about analog devices or physical instruments like guitar, piano, keyboard, etc. The reason for this trend is the fact that digital media and devices are inexpensive and are available everywhere. The existing projects have used analog methods to take inputs like Push paper paino[1], keypad[2], and push buttons[3]. The main aim of the project was to replace analog inputs with digital input which I have achieved using the Bluetooth module and the Android app. As android devices are everywhere, we can scale this project in terms of users and added functionality. HC-05 Bluetooth module is used to transmit data to android. It has two modes, master and slave as we only wanted to receive the data from the android app and not transmit any, we have used it in slave mode. For the simplicity of the project android app was designed to play only 7 notes (A, B, C, D, E, F, and G). Piezo speaker is used to play the different notes when a particular note is played from the android app.

II. Problem Definition

The problem definition was to replace analog inputs for piano with digital inputs. One can achieve this using either a Wi-Fi module or a Bluetooth module. I have used the Bluetooth module as we don't want to transmit data from long range. Most of the Bluetooth modules are capable to transmit or receive data over 50 meters of distance which is more than enough for us. Additionally, we need to design an android app that can connect to the Bluetooth module and transmit necessary information to Arduino over Bluetooth.

III. Background

The main aim of this project was to replace analog input with digital input from existing projects. A total of 3 existing projects were considered for reference while building this project. Kashan used paper and lead pencil as shown in Fig. 8 in appendix A make a paper piano to pass input as it is a well-known fact that lead and body touch creates capacitive touch which can be used instead of buttons[1]. As shown in Fig. 9 in appendix A, Adewale has used a 4x4 keypad to pass inputs to the Arduino[2]. Whereas Arduino Enigma has used push buttons to pass inputs to the Arduino[3] as shown in Fig 10 in appendix A. A tutorial on instructables.com by Abdullah Al Mamun[4] was used to create an android app to transmit data to Arduino over Bluetooth suggesting which note to play. All the projects mentioned above were implemented and then modified to create this project. Wikipedia sources were used to understand the basic terms related to music like pitch, tempo, tone, amplitude, frequency, etc.

IV. Design Description

A. Overview

This project requires a Piezo speaker, a Bluetooth module, an Arduino Uno micro-controller, an Android device, and an app to play notes. Notes to be played would be directed by the android app over Bluetooth and would be played by a piezo speaker.

B. List of Materials Required

The list of Hardware components and software required to implement Piano with Arduino Uno, Bluetooth Module, and Android App are as mentioned below.

- Hardware Components
  - Arduino Uno
  - HC-05 Bluetooth Module
  - Piezo (Buzzer)
  - Breadboard
  - Jumper wires
  - Android Mobile
- Software
  - Arduino Software (https://www.arduino.cc/en/main/software)
  - Android App making platform (http://appinventor.mit.edu/)

C. Detailed Description

This device primarily uses Arduino Uno to combine all different parts and Android app to transmit data. As shown in Fig. 1, Arduino UNo, Bluetooth module and Piezo speaker are connected to create a circuit to play specific notes.

As shown in the circuit diagram in Fig. 1 and the Schematic diagram in Fig. 2, a breadboard is only used to connect

the different parts with one another to complete the circuit using jumper wires and a wire harness is used to connect the Bluetooth module with Arduino Uno. Digital pin number 7 is used to give input to the Piezo speaker in order to play notes with a specific frequency.

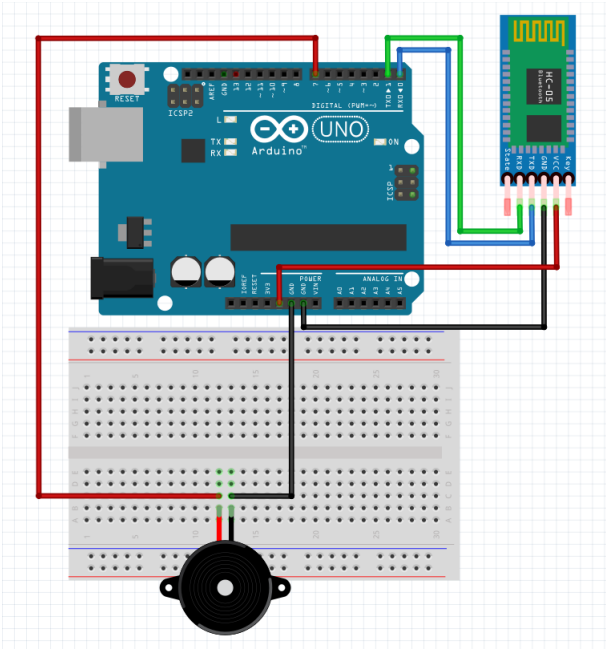

Fig. 1. Circuit of Piano

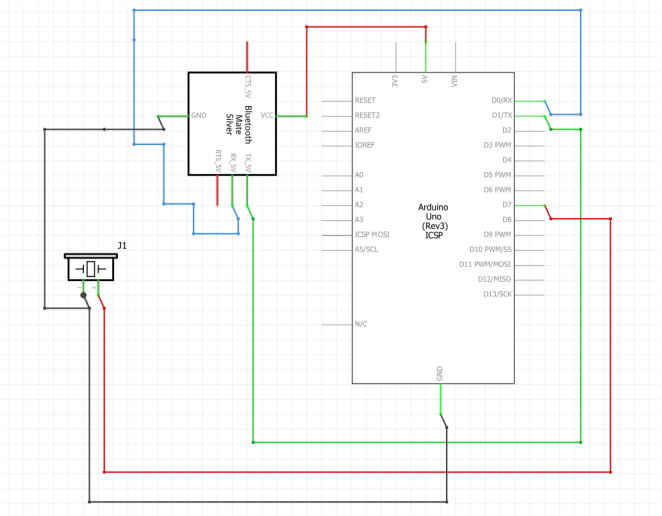

Fig. 2. Schematics of Piano

HC-05 Bluetooth module has 6 pins as shown in Fig. 1, Enable/key, Vcc, Ground, TX (Transmitter), RX (Receiver), and State. The description of each is shown below :

- **Enable/Key:** This pin is used in Data mode and Command mode. Data mode when set low and Command mode when set high. By default, it is in data mode.
- **Vcc:** As any other Vcc it is used for voltage input.
- **Ground:** This pin is used to connect with the ground.
- **TX:** This pin is used to transmit serial data.
- **RX:** This pin is used to receive serial data.
- **State:** This pin is used to connect to the onboard LED to check if Bluetooth is working properly.

Bluetooth is configured in Command mode, which includes changing the name, password, Baud rate, etc using Enable i. As the Bluetooth module is by default in slave mode and I needed it in slave mode so I need not use Enable pin. Bluetooth module comes with state-led which shows us the state of Bluetooth so State pin was also not used. Vcc pin of Bluetooth was connected to 5V voltage supply on Arduino board, the Ground pin was connected to GND of Arduino board, TX of Bluetooth was connected to RX of Arduino board and RX of Bluetooth was connected to TX of Arduino board. Above mentioned connections were the only connection needed throughout the project for Bluetooth.

Digital pin number 7 of the Arduino board was connected to the cathode of the Piezo speaker to take input and Anode was connected to the GRD of the Arduino board.

D. Build Process

The build process was difficult for developing an Android app to transmit data to a Bluetooth module connected with Arduino. I started developing an Android app using Android Studio which got tedious and was not able to achieve much. So, at last, I used an online platform (http://appinventor.mit.edu/) to develop the app. It allows users to drag and drop components needed in an app and uses code blocks to design logic for the app as shown in Fig. 3 and Fig. 4.

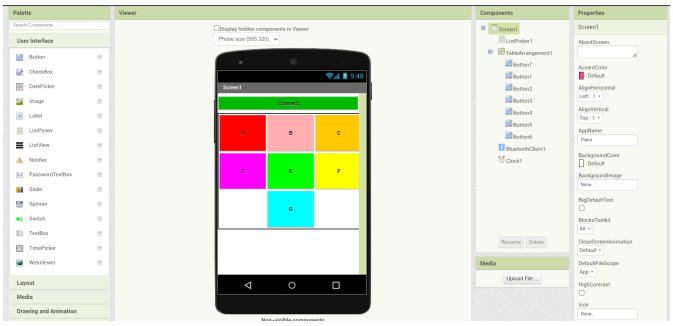

Fig. 3. App Development

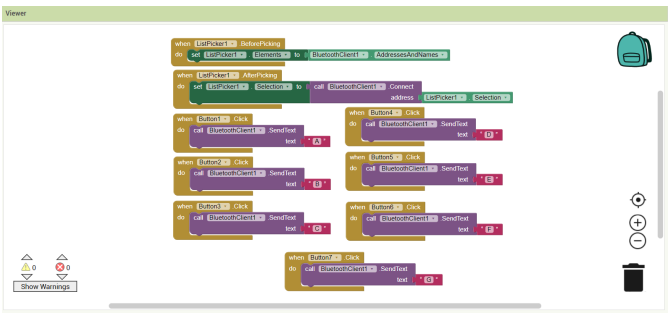

Fig. 4. App Logic Design

Implementing the circuit was easy but I did waste my time because I connected TX of Bluetooth with TX of the Arduino

board instead of RX and the same with RX of Bluetooth. Because of that, I was unable to receive any input from the android app and thought that the Bluetooth module was not working. But, later I figured that the RX of Bluetooth needs to be connected with the TX of Arduino to transmit data from Arduino, and the TX of Bluetooth should be connected to the RX of Arduino in order to receive data by Arduino.

E. Use

The user can use this project directly if he/she has an Arduino circuit implemented with code uploaded and an Android app. He/She needs to open the app and click on connect button shown in Fig. 5 which will bring the list of all Bluetooth devices paired with the device as shown in Fig. 6 and then the user needs to connect to the Bluetooth attached to the Arduino board simply by clicking on that. After connecting user can play any note form given 7 notes by clicking on the corresponding button.

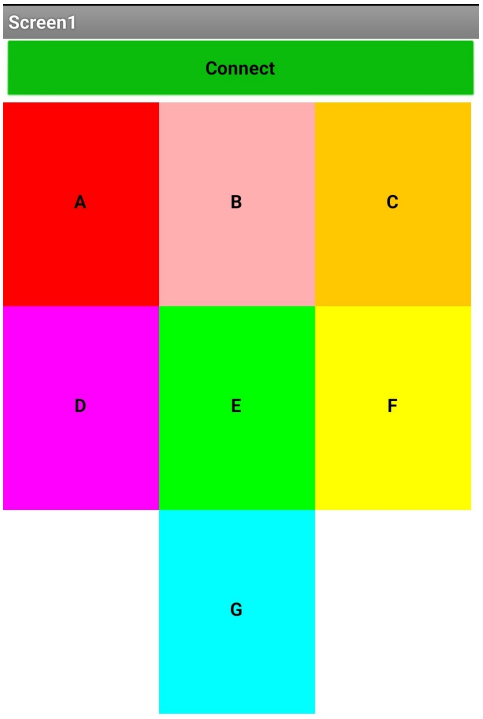

Fig. 5. App Home Screen

V. Evaluation

A. Overview

I started building this device by experimenting with individual components used in this device. After getting familiar with them, I combined them to achieve the solution to the problem. Details on experimenting with individual components are discussed further in the Milestone section.

There were some setbacks, as mentioned earlier I connected the Bluetooth module incorrectly with Arduino and due to this, I was unable to receive inputs from the Android app. Another setback was that only a single character, that too in

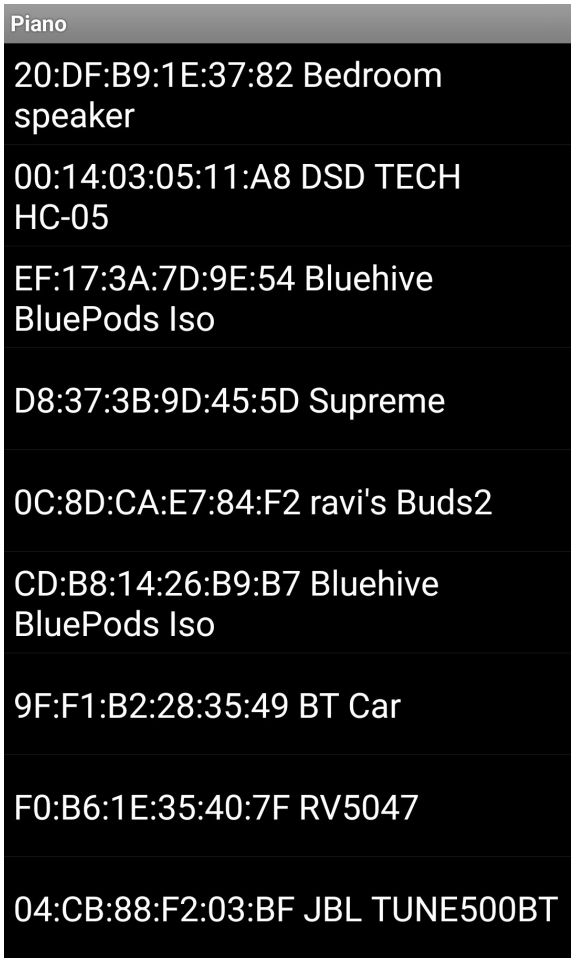

Fig. 6. Screen after clicking 'Connect' button

the form of ASCII code was transmitted at a time so I was not able to implement manual frequency tones. Only predefined frequencies are used to play sound.

B. Prototype

I only created one prototype which was getting me the desired outputs. My prototype is shown in Fig. 7 and the android app is shown in Fig. 5.

C. Testing and Results

Before creating the app using MIT online platform, I tried making the app using Android Studio and conventional methods but given the time constraint, I was not able to create what was needed for this project to work. So, finally, I used MIT online platform to create an app. A limitation with MIT online platform is that it provides limited code blocks and there is no way to add manual code, due to which app was only able to transmit one character at a time. So, I used that to pass the name of the note to be played ('A", 'B', 'C', 'D', 'E', 'F', and 'G') and was successful.

Bluetooth was able to transmit data from 50 meters, this was tested by playing the Piano which was placed in a hall and the user was standing in the parking lot which is 50 meters

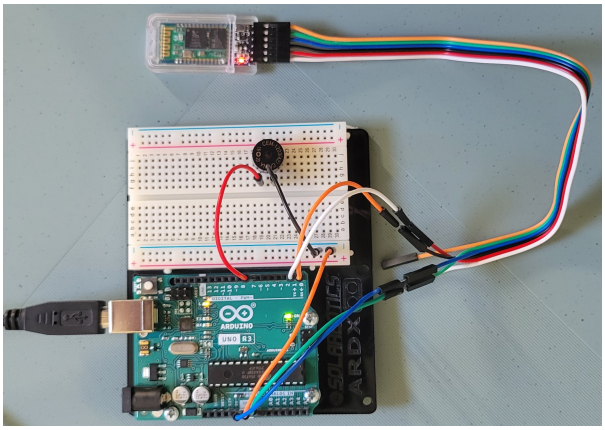

Fig. 7. Final Prototype

away. ”Happy Birthday to You” was played using a developed project for testing and the result was excellent.

D. Assessment

During, testing everything worked fine. There was one limitation though, the note was playing only when clicked and not while holding the button in-app. So, if someone has to play one note at a time, this is the perfect project to do so.

E. Next Steps

There are several improvements that can be made to this project, those are mentioned below.

- User interface can be made better, which looks like a piano with white and black keys.
- Two notes could be played together as in real piano.
- note should be played even while holding the button in the app.
- User could be able to set tempo and pitch of his/her desire.
- A display screen could be attached to an Arduino board to display notes playing.

VI. Milestones

Overall 5 milestones including stretch goals were required to complete this project.

- **Milestone 1:**
  To achieve milestone one, I gained knowledge related to sound waves, particularly related to music and instruments. I learned what is pitch, tempo, frequency, amplitude, and tone, and how they affect the sound produced.
  Along with all these, I started collecting the components required for this project, which were Arduino Uno, Jumper wires, Piezo speaker, Bluetooth module, and Display screen. I had all the components except the display screen and Bluetooth module so, I ordered them online. Additionally, I downloaded and installed Android Studio for app development.
- **Milestone 2:**
  To complete milestone 2, I created a simulation of 7 Push button piano on tinkercad.com as shown in Fig. 13 of Appendix A. Code for this simulation can be found in Appendix B as code 2. 7 push buttons, a Piezo speaker, jumper wire and an Arduino Uno micro-controller is needed to implement this simulation. The simulation was successful as the tone was produced by pressing push buttons. 7 Push button means 7 notes (’A”, ’B’, ’C’, ’D’, ’E’, ’F’, and ’G’) were designed for this piano.
- **Milestone 3:**
  To complete milestone 3, I implemented turning the LED on/off using Bluetooth to understand the working of the Bluetooth module. Additionally, I created an Android app to control LED (App can be found in the GitHub repository). I was not able to complete this milestone on time because of the late arrival of the Bluetooth module. Additionally, I tried making an app using Android Studio which was a failure, and then I created an app using MIT online platform (http://appinventor.mit.edu/).
- **Milestone 4:**
  To complete milestone 4, I combined the concepts from milestones 2 and 3 to create a piano that takes inputs from an Android app (App can be found in the GitHub repository). After completing this milestone, the project was ready which was a piano that is capable of playing 7 notes (’A”, ’B’, ’C’, ’D’, ’E’, ’F’, and ’G’) using the android app.
- **Milestone 5:**
  I was not able to complete this milestone due to time constraints. The display screen was lost in transit. Additionally, I was only able to transmit a single character over Bluetooth, due to which I was not able to design an android app that can give the user flexibility to play with the pitch and tempo of tones.

VII. Team Roles

This project was done by ”Ravi Patel” including project selection, research, setting milestones, hardware design, software design, prototyping, documentation, GitHub repository, and Video presentation.

VIII. Conclusion

The motivation for choosing this project was a long desire to learn some musical instruments and this project gave me the perfect opportunity to explore some concepts of music and instruments. During the process, I learned the basics of music, and the Bluetooth Module and found an easy but inefficient way to create an android app online without downloading anything. I was able to complete this project to a satisfactory level with room for improvement. The final prototype was able to play 7 different notes (’A”, ’B’, ’C’, ’D’, ’E’, ’F’, and ’G’) when directed by the user using the android app. The main takeaway from this project was that I was able to replace

analog input with digital input which was transmitted using Bluetooth. The developed project can be extended to a great extent because of the fact that devices like Android mobile and Arduino micro-controller are inexpensive and almost everyone has one.

APPENDIX A
DIAGRAMS AND PICTURES

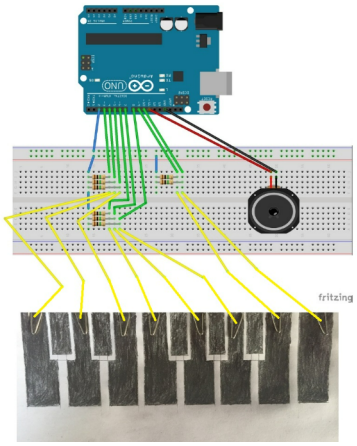

Fig. 8. Paper Piano with Arduino

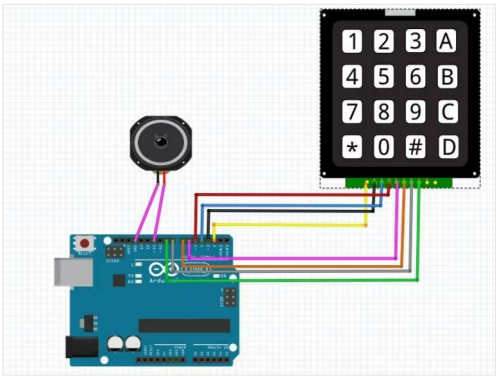

Fig. 9. Arduino Piano using 4x4 Keypad

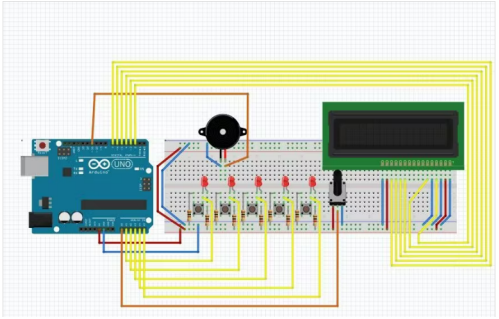

Fig. 10. A Simple, Five Button, Polyphonic, Arduino Uno-Based Piano

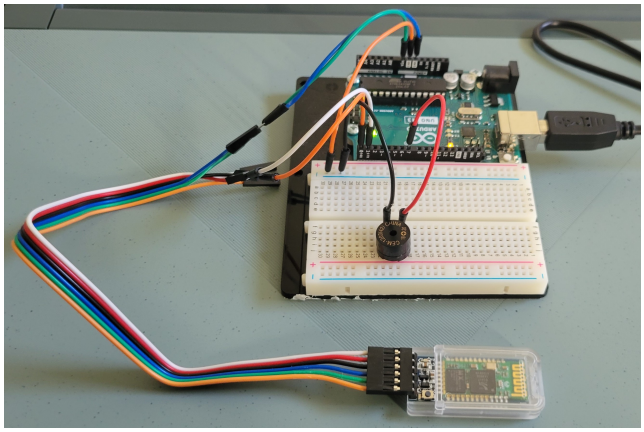

Fig. 11. Side view of main prototype

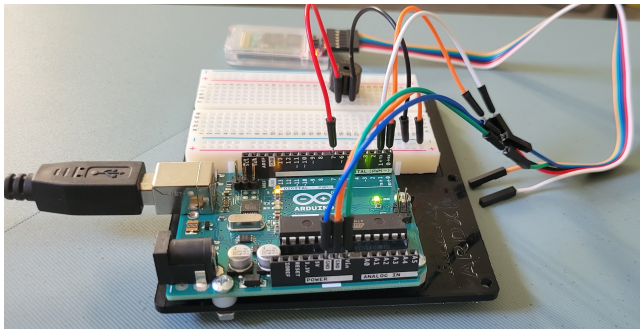

Fig. 12. Side view of main prototype

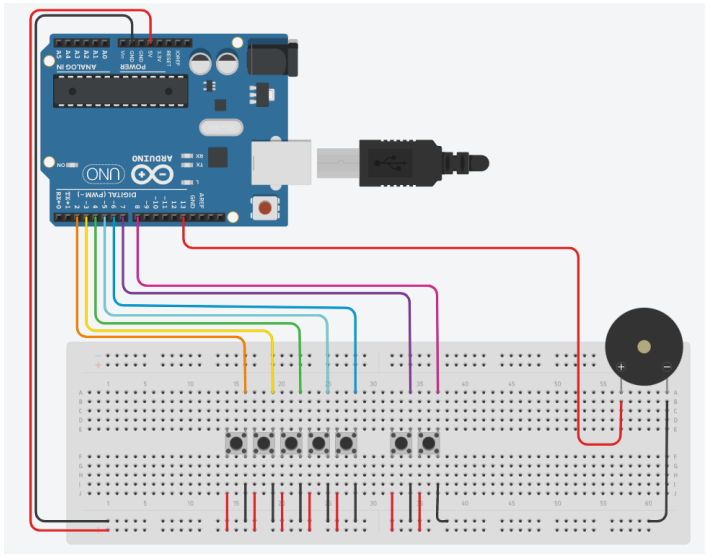

Fig. 13. Circuit of Push Button Piano

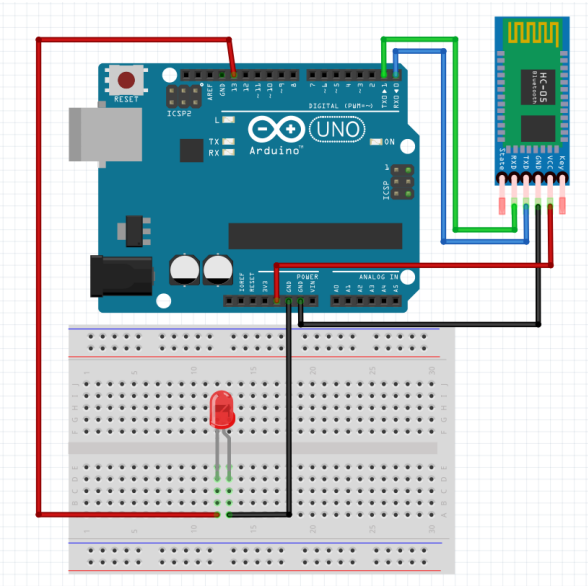

Fig. 14. Circuit of turning LED on/off using Bluetooth

APPENDIX B
SOURCE CODE

Code 1 : Main code for this project

```
/*
 * This sketch is code to recieve inputs from Android app over Bluetooth.
 * Inputs from Android app is notes to be played using Piezo speaker.
*/

int a;  // Variable to store ASCII value of Characters (Notes to be played)
int buzzerPin = 7;  // Input pin for Piezo

// Initialize varibales, pin modes etc.
void setup()
  {
  Serial.begin(9600); // Baud rate (exchange rate of 9600 bit per second)
  pinMode(buzzerPin,OUTPUT);  // Initialize pin mode
  }

// Infinite loop
void loop()
  {
  // When something is transmitted over Bluetooth
  while(Serial.available()) // Serial.available() returns size of message transmitted over bluetooth in
    BYTE
    {
    a=Serial.read();  // read message transmitted
    Serial.println(a);  // print to serial monitor for debugging

    // when note 'A' is played (ASCII for A = 65)
    if(a==65)
      {
        tone(buzzerPin, 262); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }

    // when note 'B' is played (ASCII for B = 66)
    if(a==66)
      {
        tone(buzzerPin, 294); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }

    // when note 'C' is played (ASCII for C = 67)
    if(a==67)
      {
        tone(buzzerPin, 330); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }

    // when note 'D' is played (ASCII for D = 68)
    if(a==68)
      {
        tone(buzzerPin, 349); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }

    // when note 'E' is played (ASCII for E = 69)
    if(a==69)
      {
        tone(buzzerPin, 392); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }

    // when note 'F' is played (ASCII for F = 70)
    if(a==70)
      {
        tone(buzzerPin, 440); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }
```

```
    // when note 'G' is played (ASCII for G = 71)
    if(a==71)
      {
        tone(buzzerPin, 494); // frequency of tone to be played with pin number
        delay(200); // delay of 200 ms
        noTone(buzzerPin);  // stop the tone
      }
    }
  }
```

Code 2: Code for Push Button Piano

```
/*
 * This sketch is code to play tones using Push Buttons
*/
int button1 = 2; // input pin for button 1
int button2 = 3; // input pin for button 2
int button3 = 4; // input pin for button 3
int button4 = 5; // input pin for button 4
int button5 = 6; // input pin for button 5
int button6 = 7; // input pin for button 6
int button7 = 8; // input pin for button 7

int buzzerPin = 13; // input pin for Piezo

// Initialize varibales, pin modes etc.
 void setup()
{
  //initialize button pins as input
  pinMode(but1,INPUT);
  pinMode(but2,INPUT);
  pinMode(but3,INPUT);
  pinMode(but4,INPUT);
  pinMode(but5,INPUT);
  pinMode(but6,INPUT);
  pinMode(but7,INPUT);

  //initialize buzzer pin as output
  pinMode(buzzer,OUTPUT);

}

// Infinite loop
void loop()
{
  //read the value from buttons
  int b1 = digitalRead(button1);
  int b2 = digitalRead(button2);
  int b3 = digitalRead(button3);
  int b4 = digitalRead(button4);
  int b5 = digitalRead(button5);
  int b6 = digitalRead(button6);
  int b7 = digitalRead(button7);

  // when button1 is pressed
  if( b1 == 1 ) // play note 'A'
    {
      tone(buzzerPin, 262); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
    }

  // when button2 is pressed
  else if( b2 == 1 ) // play note 'B'
    {
      tone(buzzerPin, 294); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
    }

  // when button3 is pressed
  else if( b3 == 1 ) // play note 'C'
    {
      tone(buzzerPin, 330); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
```

```
    }

  // when button4 is pressed
  else if( b4 == 1 ) // play note 'D'
    {
      tone(buzzerPin, 349); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
    }

  // when button5 is pressed
  else if( b5 == 1 ) // play note 'E'
    {
      tone(buzzerPin, 392); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the
    }

  // when button6 is pressed
  else if( b6 == 1 ) // play note 'F'
    {
      tone(buzzerPin, 440); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
    }

  // when button7 is pressed
  else if( b7 == 1 ) // play note 'G'
    {
      tone(buzzerPin, 494); // frequency of tone to be played with pin number
      delay(200); // delay of 200 ms
      noTone(buzzerPin);  // stop the tone
    }
}
```

Code 3: Code for turning LED on/off using Bluetooth

```
/*
 * This sketch is code to recieve inputs from Android app over Bluetooth.
 * Inputs from Android app is used to turn LED on/off.
*/

int a; // Variable to store ASCII value of Character
int led = 13; // input pin for LED

// Initialize varibales, pin modes etc.
void setup()
  {
    Serial.begin(9600); // Baud rate (exchange rate of 9600 bit per second)
    pinMode(led,OUTPUT);  // Initialize pin mode
  }

// Infinite loop
void loop() {
  // When something is transmitted over Bluetooth
  if(Serial.available() > 0)
    {
      a=Serial.read(); // read message transmitted
      //Serial.println("inside"); // print statement for debugging
      //Serial.println(a==78);  // print statement for debugging

      // when character 'O' is transmitted (ASCII for O is 78)
      if(a==78)
        {
          digitalWrite(led,HIGH); // turn on the LED
        }
      // when character 'F' is transmitted (ASCII for F is 70)
      if(a==70)
        {
          digitalWrite(led,LOW);  // turn off the LED
        }
    }
}
```

REFERENCES

[1] Ikhsan, *Paper Piano with Arduino*, 2018. Available at: https://www.hackster.io/Barqunics/paper-piano-with-arduino-e27da7
[2] Adewale Paul, *Arduino Piano Using 4x4 Keypad*, 2019. Available at: https://www.hackster.io/drwale2000/arduino-piano-using-4x4-keypad-f5ff8c
[3] Arduino Enigma, *A Simple, Five Button, Polyphonic, Arduino Uno-Based Piano*, 2018. Available at: https://www.hackster.io/arduinoenigma/a-simple-five-button-polyphonic-arduino-uno-based-piano-b367be
[4] Abdullah Al Mamun, *Build Android Bluetooth App for Arduino*. Available at: https://www.instructables.com/Build-Android-Bluetooth-App-for-Arduino/