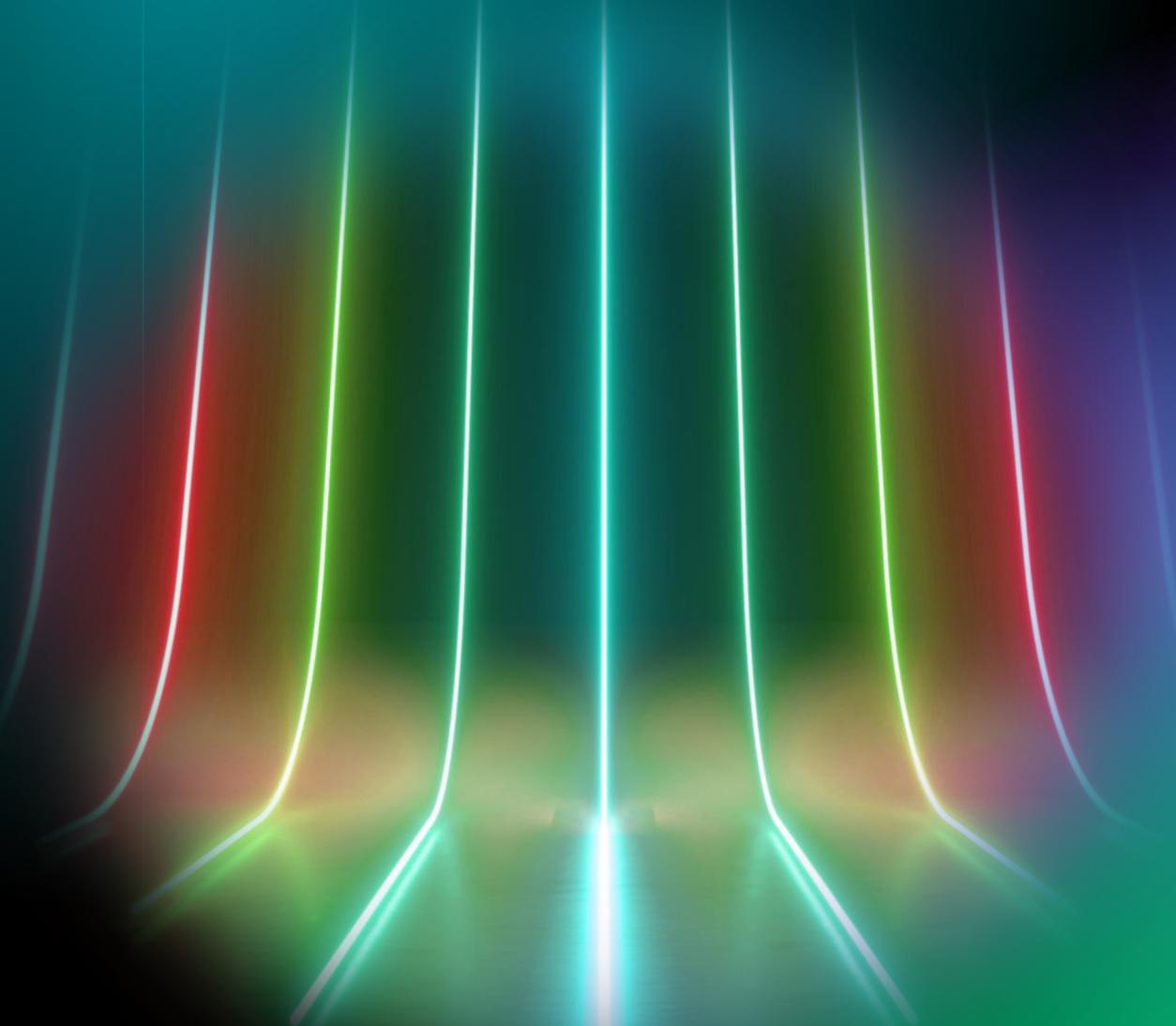


Arduino Interface Project

Musical Lamp

NARENDRA KONATHALA

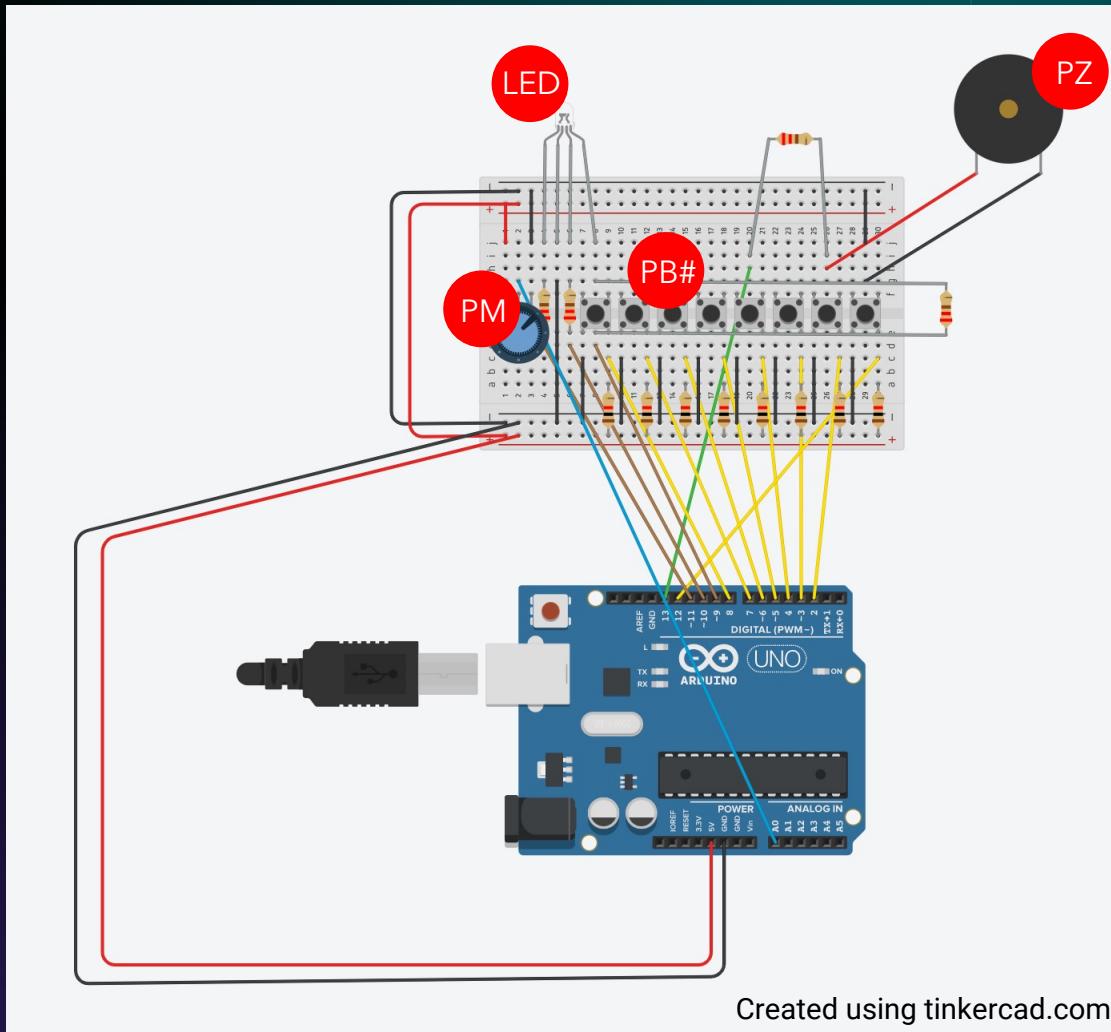


Description

User can play music(piezo) using notes (Do Re Mi Fa Sol La Ti Do) with help of the push buttons on the board. Alternatively, user can tune in to listen to three automated feed-in songs using the potentiometer. The lamp (RGB LED) in both manual and automated use case, changes its color based on the frequency of sound produced.



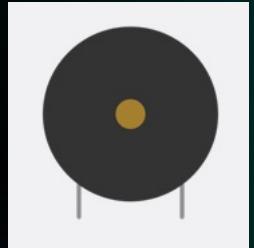
Circuit & Components



Name	Quantity	Component
UNO	1	Arduino Uno R3
PZ	1	Piezo
R1 R2 R4 R5 R6 R7 R8 R9	8	1 kΩ Resistor
PB1 PB2 PB3 PB4 PB5 PB6 PB7 PB8	8	Pushbutton
R10 R3 R11 R12	4	220 Ω Resistor
LED	1	LED RGB
PM	1	Potentiometer

https://www.tinkercad.com/things/8MzyAGi158I-piano/editel?lessonid=EHD2303J3YPUS5Z&projectid=OIJY88OJ3OPN3EA&sharecode=f5MWqPZiTld_BBDQ0rHaZvr8PqsJQhfC2Wq45Tp3B8

Sound

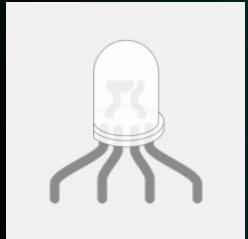


Piezo

A type of buzzer that makes noise at different frequencies

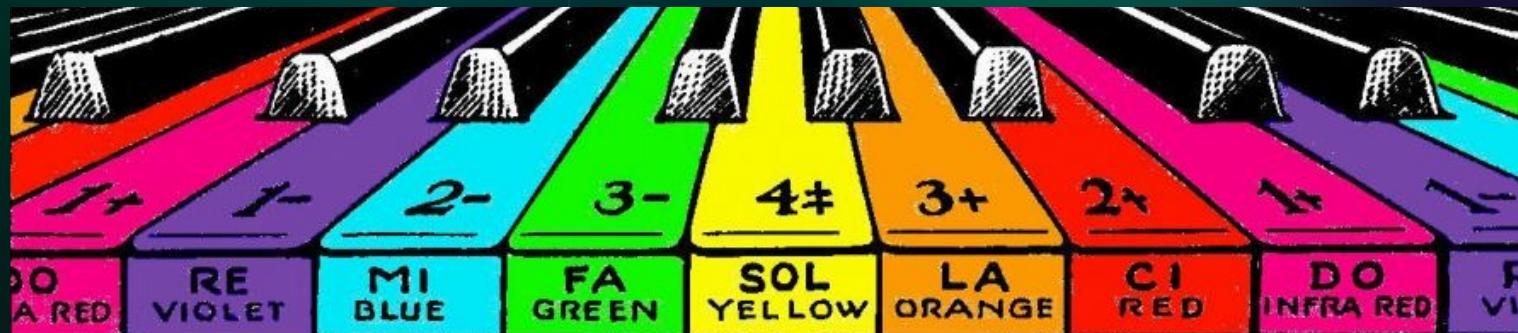
Note	Frequency (Hz)
Do	440
Re	493
Mi	554
Fa	587
Sol	659
La	739
Ti	830
Do#	880

Light



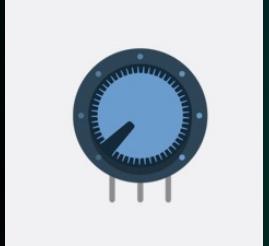
LED RGB

A type of LED that combines Red, Blue and Green to produce any color.



Color-tone combinations by Walter Bowman Russell

Music



Potentiometer

A type of resistor whose resistance changes at the turn of a knob

Slot	Reading	Song
1	0 - 255	Manual Mode
2	256 - 511	Ode to joy - Beethoven
3	512 - 767	Do-Re-Mi - Sound of music
4	768 - 1023	Kal-ho-na-ho - Hindi Music (Hindi)

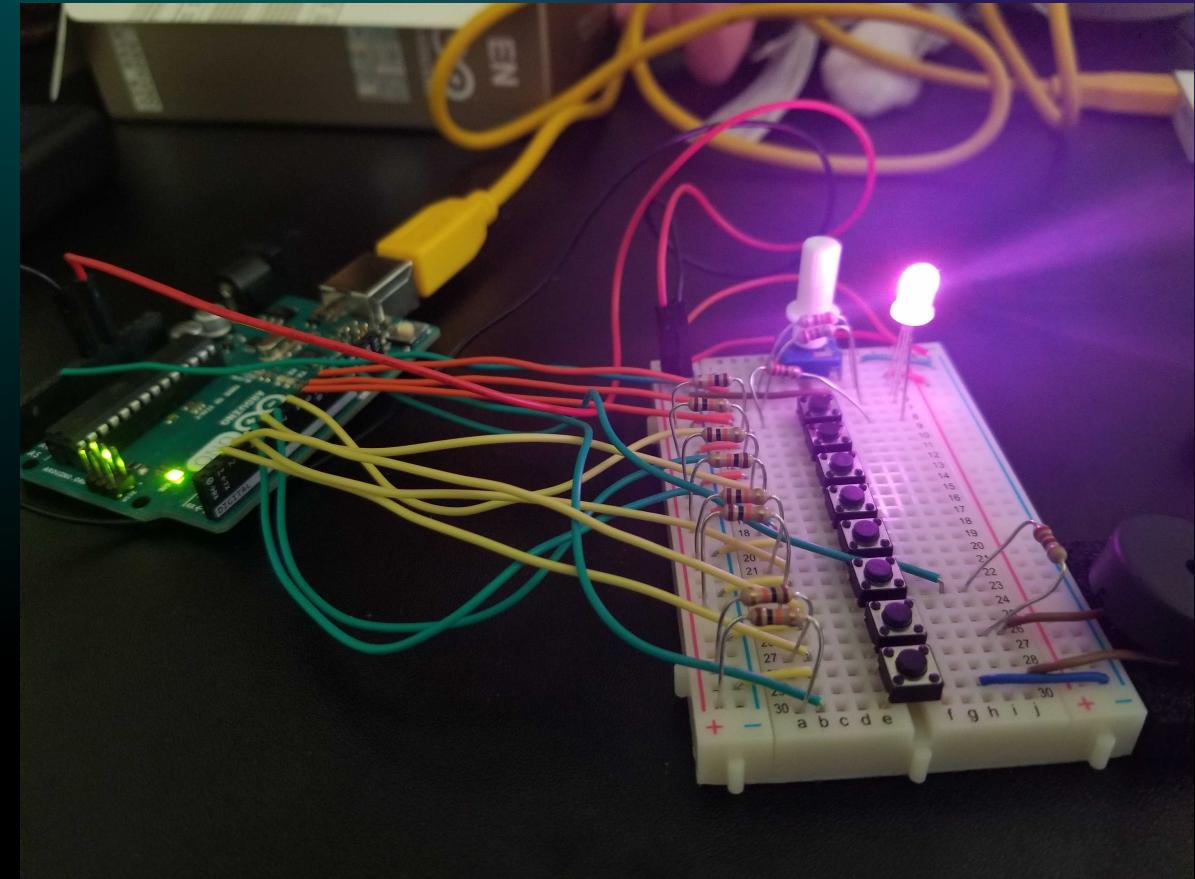
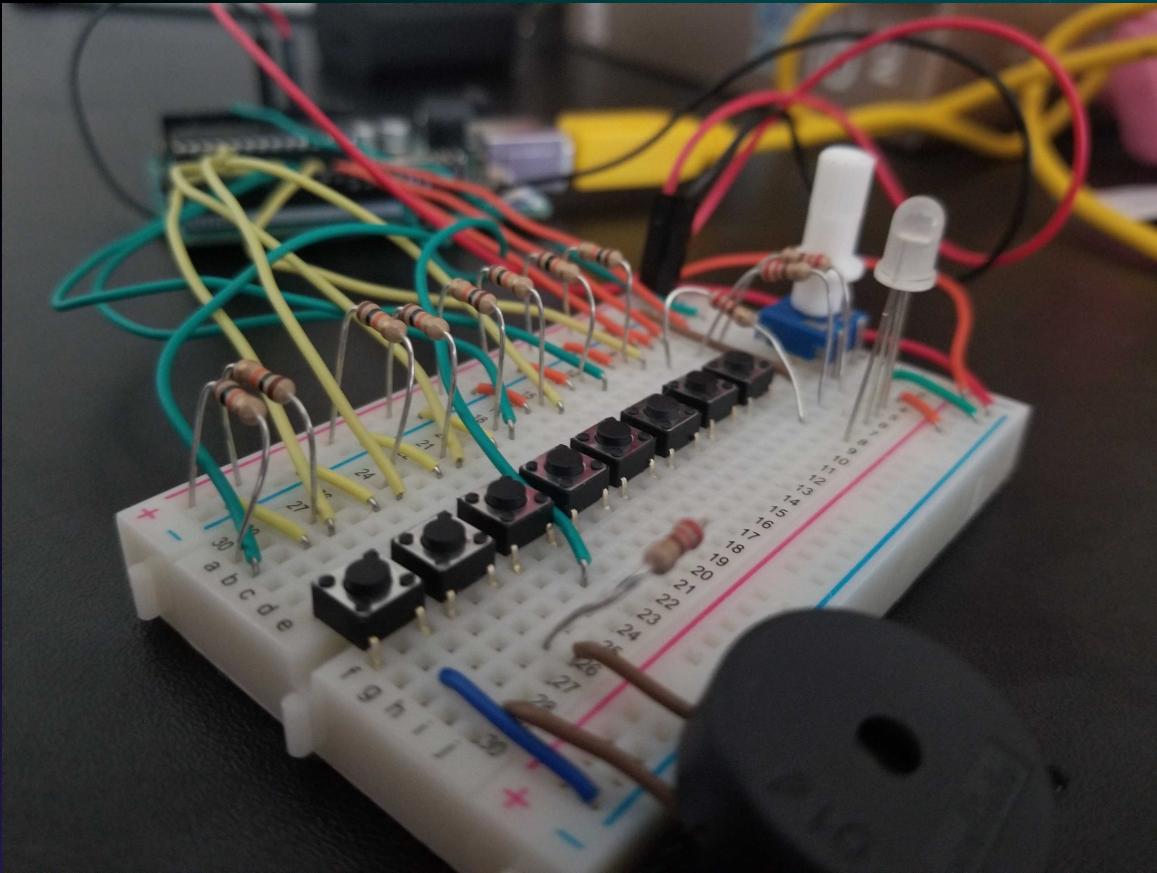
```
[{"name": "kal ho na ho (HINDI Song)", "bpm": 250, "bitsPerPage": 16, "pitchLevel": 0, "isComposed": "true", "songNotes": [{"time": 400, "key": "1Key7"}, {"time": 1000, "key": "1Key6"}, {"time": 1240, "key": "1Key7"}, {"time": 1840, "key": "1Key6"}, {"time": 2080, "key": "1Key7"}, {"time": 2620, "key": "1Key6"}, {"time": 2860, "key": "1Key7"}, {"time": 3100, "key": "1Key9"}, {"time": 3340, "key": "1Key8"}, {"time": 3580, "key": "1Key7"}, {"time": 3820, "key": "1Key6"}, {"time": 4420, "key": "1Key5"}]}
```



```
notes = { 6, 6, 7, 8, 8, 7, 6, 5, 4, 4, 5, 6, 6, 5, 5, 6, 6, 7, 8, 8, 7, 6, 5, 4, 4, 5, 6, 5, 4, 4, 5, 5, 6, 4, 5, 6, 7, 6, 4, 5, 6, 7, 6, 5, 4, 5, 1, 6, 6, 7, 8, 8, 7, 6, 5, 4, 4, 5, 6, 5, 4, 4 };
```

```
timing = {423, 450, 483, 428, 450, 456, 445, 460, 452, 458, 460, 467, 719, 237, 996, 473, 489, 427, 467, 436, 456, 436, 455, 456, 428, 446, 585, 761, 210, 977, 442, 483, 468, 453, 443, 233, 267, 449, 451, 421, 206, 281, 460, 484, 466, 493, 991, 450, 466, 431, 441, 442, 457, 454, 463, 469, 427, 521, 489, 773, 251, 500};
```

Result



```
1 /**
2  Music Lamp – Narendra Konathala
3  Maps tone to color
4
5  Do Re Mi Fa Sol La Ti Do
6  OR
7  Sa Re Ga Ma Pa Da Ni Sa
8
9  Frequencies used from
10 https://en.wikipedia.org/wiki/Saptak
11 https://licn.typepad.com/my\_weblog/2011/07/the-unsolvable-problem-john-dunn-consultant-ambertec-pe-pc.html
12
13 Colors obtained from – Based on tone and color – Walter Rusell – The
14 universal One
15 https://roelhollander.eu/en/tuning-frequency/sound-light-colour/
16
17 Music notes obtained from https://sky-music.herokuapp.com/
18 */
19
20 const int piezo = 13; //Restricted 11 or 3 – Per documentation
21
22 //Only using PWM digital pins for analog writes
23 const int redPWM = 11;
24 const int greenPWM = 9;
25 const int bluePWM = 10;
26
27
28 int pushButtons[] = {8, 7, 6, 5, 4, 3, 2, 12};
29
30 int frequencies[] = {440, 493, 554, 587, 659, 739, 830, 880};
31
32 int redVariant[] = {255, 127, 0, 0, 241, 255, 255, 255};
33 int greenVariant[] = {73, 0, 0, 255, 196, 165, 0, 73};
34 int blueVariant[] = {108, 255, 255, 0, 15, 0, 0, 108};
35
36
37 void setup() {
38   pinMode(piezo, OUTPUT);
39   pinMode(redPWM, OUTPUT);
40   pinMode(greenPWM, OUTPUT);
41   pinMode(bluePWM, OUTPUT);
42
43   for (int i = 0; i < 8; i++) {
44     pinMode(pushButtons[i], INPUT);
45   }
46
47   Serial.begin( 9600 );
48
49 }
50
51 void loop() {
52
53   int val = analogRead( A0 );
54   int mappedValue = map(val, 0, 1023, 1, 4);
55
56   if (mappedValue == 1) {
```

```

57     manualMode();
58 } else if (mappedValue == 2) {
59     automaticMode(1, val);
60 } else if (mappedValue == 3) {
61     automaticMode(2, val);
62 } else if (mappedValue == 4) {
63     automaticMode(3, val);
64 }
65
66 }
67 }
68
69 //Automatic song feed
70 void automaticMode(int song, int meterValue) {
71
72     //Ode to joy - Beethoven
73     int notes1[] = { 6, 6, 7, 8, 8, 7, 6, 5, 4, 4, 5, 6, 6, 5, 5, 6, 6, 7, 8,
74     8, 7, 6, 5, 4, 4, 5, 6, 5, 4, 4, 5, 5, 6, 4, 5, 6, 7, 6, 4, 5, 6, 7, 6, 5, 4,
75     5, 1, 6, 6, 7, 8, 8, 7, 6, 5, 4, 4, 5, 6, 5, 4, 4, 4 };
76     int timing1[] = {423, 450, 483, 428, 450, 456, 445, 460, 452, 458, 460,
77     467, 719, 237, 996, 473, 489, 427, 467, 436, 456, 436, 455, 456, 428, 446,
78     585, 761, 210, 977, 442, 483, 468, 453, 443, 233, 267, 449, 451, 421, 206,
79     281, 460, 484, 466, 493, 991, 450, 466, 431, 441, 442, 457, 454, 463, 469,
80     427, 521, 489, 773, 251, 500};
81
82     //Do-Re-Mi - Sound of music
83     int notes2[] = { 0, 1, 2, 0, 2, 0, 2, 1, 2, 3, 3, 2, 1, 3, 2, 3, 4, 2, 4,
84     2, 4, 3, 4, 5, 5, 4, 3, 5, 4, 0, 1, 2, 3, 4, 5, 5, 1, 2, 3, 4, 5, 6, 6, 2, 3,
85     4, 5, 6, 7, 7, 6, 5, 3, 6, 4, 7, 4, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7, 4, 7 };
86     int timing2[] = { 710, 219, 716, 281, 445, 466, 974, 662, 272, 205, 252,
87     210, 271, 1889, 733, 268, 777, 254, 469, 484, 933, 654, 292, 182, 263, 231,
88     247, 1816, 652, 254, 207, 230, 223, 270, 1991, 648, 229, 224, 228, 194, 268,
89     1896, 653, 244, 233, 213, 256, 228, 1322, 223, 286, 467, 442, 490, 524, 402,
90     475, 450, 491, 227, 207, 247, 193, 229, 221, 366, 498, 489, 450 };
91
92     //Kal-ho-na-ho - Hindi Music
93     int notes3[] = { 7, 6, 7, 6, 7, 6, 7, 9, 8, 7, 6, 5, 6, 5, 6, 7, 6, 7, 6,
94     7, 6, 7, 9, 8, 7, 6, 5, 6, 5, 6, 4, 5, 7, 5, 3, 4, 5, 4, 4, 5, 7, 7, 5, 4, 3,
95     3, 5, 5 };
96     int timing3[] = { 600, 240, 600, 240, 540, 240, 240, 240, 240, 240, 600,
97     240, 540, 240, 1230, 540, 240, 540, 240, 540, 240, 240, 240, 240, 240, 540,
98     240, 540, 240, 600, 240, 240, 240, 1080, 240, 240, 240, 240, 1260, 240, 240, 240,
99     240, 990, 240, 600, 480, 240, 400 };
100
101     int* notes;
102     int* timing;
103     int length; //Length of notes and timing array should be same
104     if (song == 1) {
105         notes = notes1;
106         timing = timing1;
107         length = sizeof(notes1)/sizeof(notes1[0]);
108     }else if (song == 2) {
109         notes = notes2;
110         timing = timing2;
111         length = sizeof(notes2)/sizeof(notes2[0]);
112     }else if (song == 3) {
113         notes = notes3;
114         timing = timing3;
115         length = sizeof(notes3)/sizeof(notes3[0]);

```

```
99
100
101 for (int i = 0; i < length; i++) {
102     int val = notes[i];
103
104     if (val > 7) {
105         val = val - 7;
106     }
107
108     int newVal = analogRead( A0 );
109     if (newVal != meterValue) {
110         int mappedValue = map(newVal, 0, 1023, 1, 4);
111         if (mappedValue == 1 || (mappedValue - 1) != song){
112             noTone(piezo);
113             setLEDColor(255, 255, 255);      //Default Color
114             return; //If meter is changed during the song
115         }
116     }
117
118 }
119
120     setLEDColor(redVariant[val],           //Writing Color based on song
121                 greenVariant[val], blueVariant[val]);
122     tone(piezo, frequencies[val], 220);
123     delay(timing[i]);
124 }
125
126 delay(2000); //Song ended
127
128 }
129
130
131 void manualMode() {
132     int j = 0;
133     bool pushed = false;
134
135     for (int i = 0; i < 8; i++) {
136
137         if ( digitalRead(pushButtons[i]) == LOW) {
138
139             //Writing Color based on button pressed
140             setLEDColor(redVariant[i], greenVariant[i], blueVariant[i]);
141
142             tone(piezo, frequencies[i], 220);
143             delay(220);
144
145             pushed = true;
146         }
147     }
148
149     if (pushed == false) {
150         noTone(piezo);
151         setLEDColor(255, 255, 255);      //Default Color
152     }
153
154 }
155
156
157 }
```

```
158 void setLEDColor(int r, int g, int b) {  
159     analogWrite(redPWM, r);  
160     analogWrite(greenPWM, g);  
161     analogWrite(bluePWM, b);  
162 }
```