

Arduino Notebook 2

Traffic Light

Prakriti Bansal
Phanith DSV

Table of Contents

Table of Contents	2
Introduction	3
Abstract	3
Experiments	3
Hito 2.1	3
Hito 2.2	4
Hito 2.3	4
Hito 2.4	5
Hito 2.5	5
Hito 2.6	6
Hito 2.7	6
Hito 2.8	7
Traffic Light System Simulation	8
Working Hours	10
Team Experience	10

Introduction

The task allotted to the team was to simulate a traffic light control system using Arduino circuitry with the help of TinkerCad simulator.

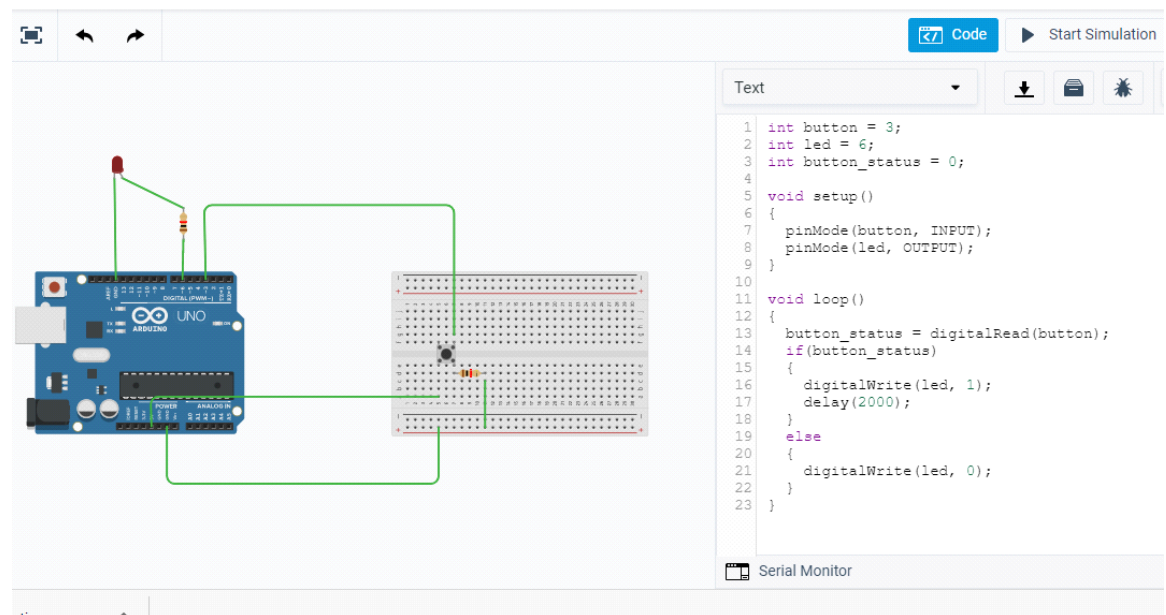
Abstract

The system will include basic functionality like changing of red, yellow and green light signals depending upon the traffic flow along with an additional functionality of triggering a buzzer for visually impaired pedestrians.

Experiments

Hito 2.1

c Bruticus

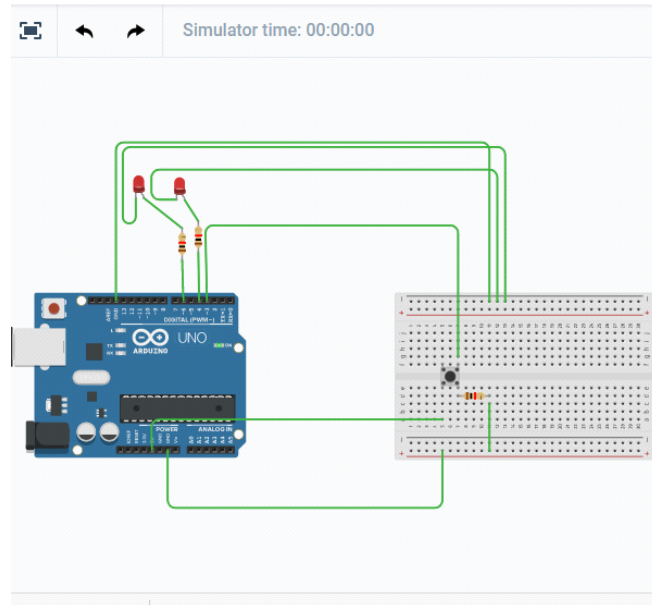


The experiment enabled us to control an element like LED by reading values from a Push button.

Hito 2.2

c Bruticus

All changes s



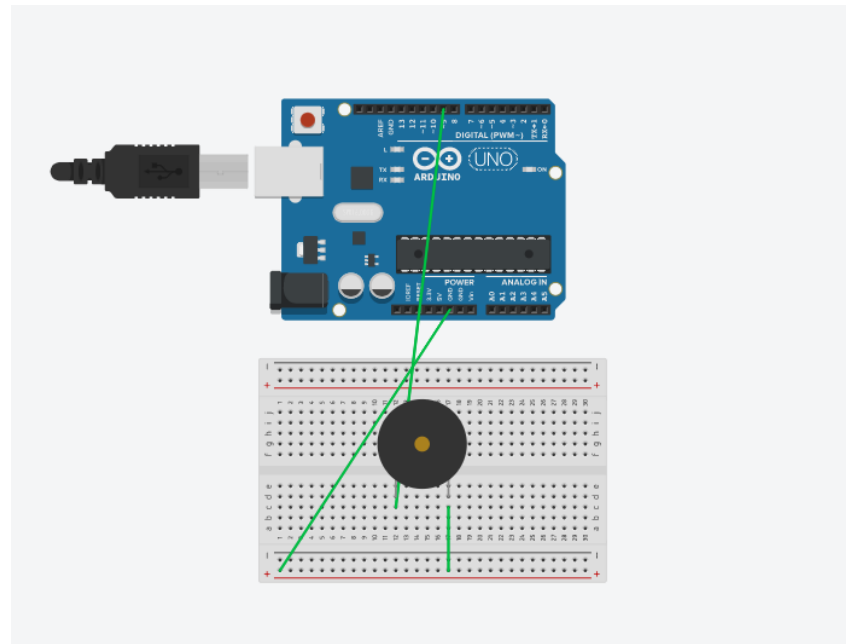
Text

```
1 int brightness = 0;
2 int brightness2 = 0;
3 int fadeAmount = 5;
4
5 void setup()
6 {
7   pinMode(6, OUTPUT);
8   pinMode(4, OUTPUT);
9 }
10
11 void loop()
12 {
13   analogWrite(6, brightness);
14   analogWrite(4, brightness2);
15
16   brightness = brightness + fadeAmount;
17   brightness2 = brightness + fadeAmount;
18
19   if(brightness==0 || brightness==255)
20   {
21     fadeAmount = -fadeAmount;
22   }
23   if(brightness2==0 || brightness2==255)
24   {
25     fadeAmount = -fadeAmount;
26   }
27 }
```

Serial Monitor

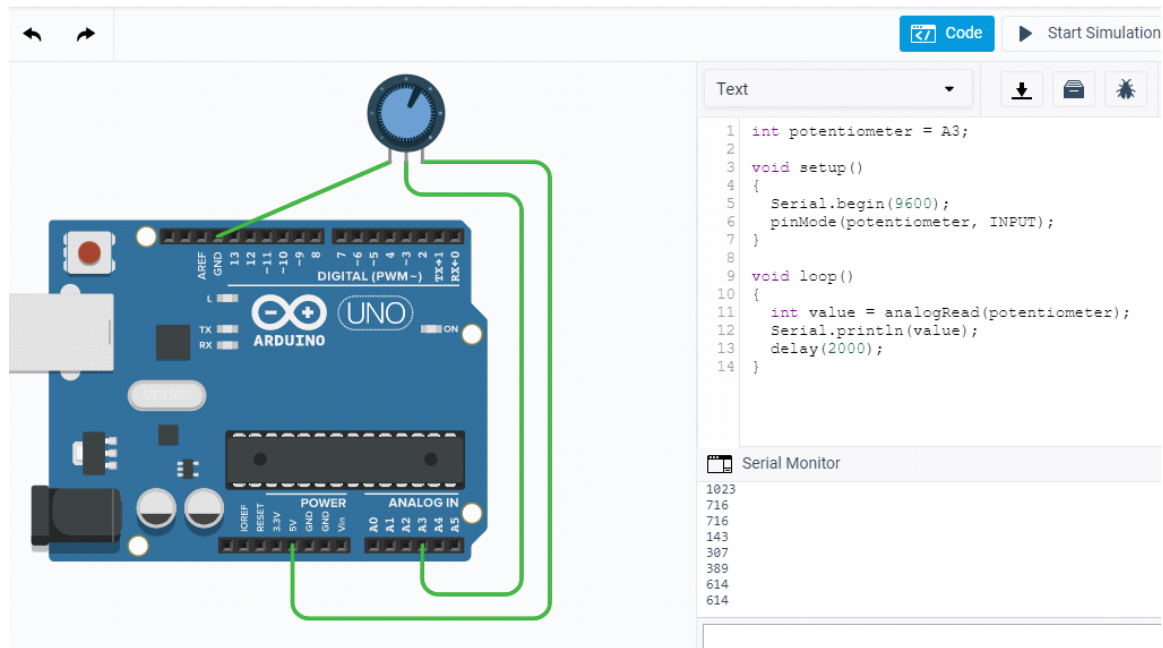
The experiment helped us to understand how to control more than two elements (like LEDs) together by modifying the code slightly.

Hito 2.3



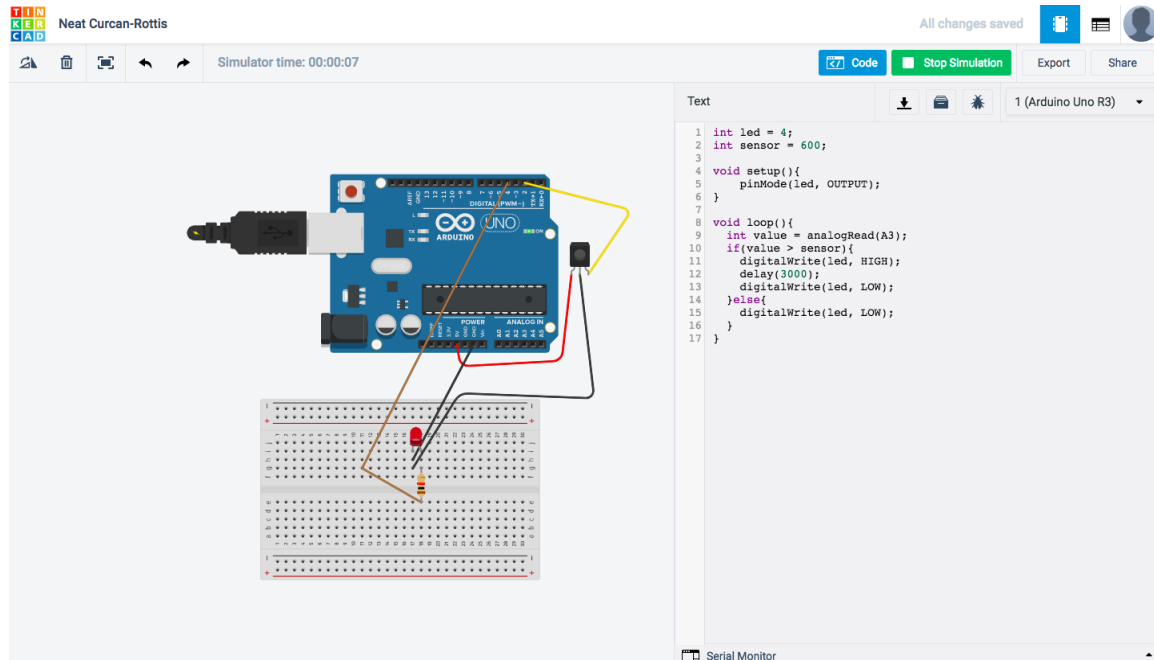
By performing this experiment, we understood how to create notes and music using Arduino and thus, implement a buzzer.

Hito 2.4



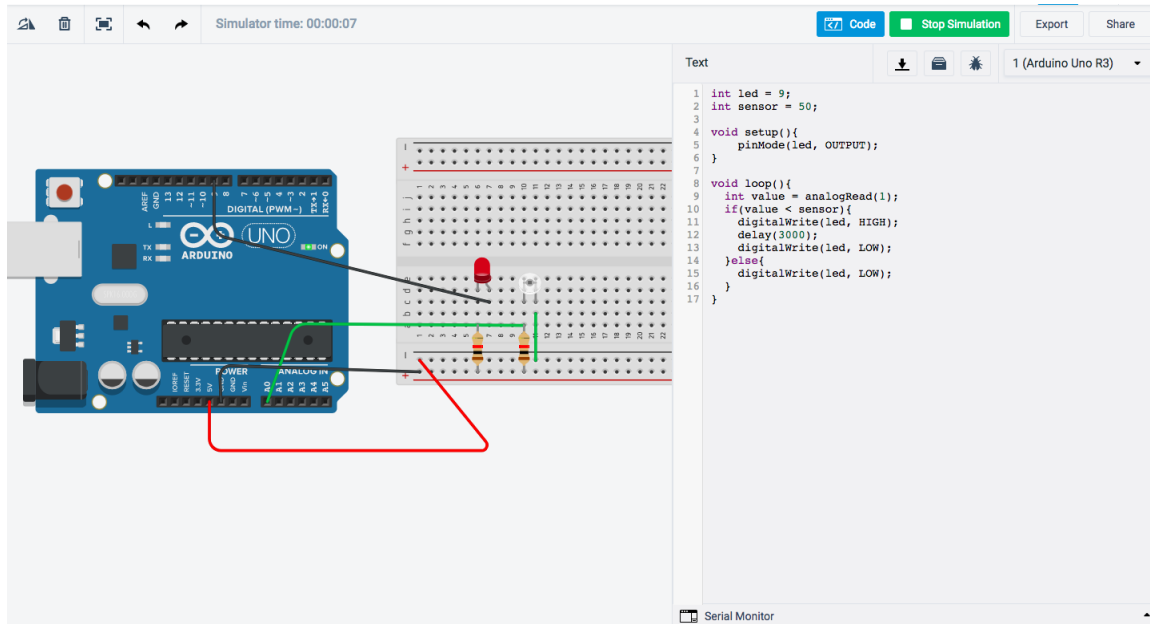
The experiments helped us understand the basic concept of using a potentiometer in order to find rotation angle which can further be translated to the rotatory motions like opening doors.

Hito 2.5



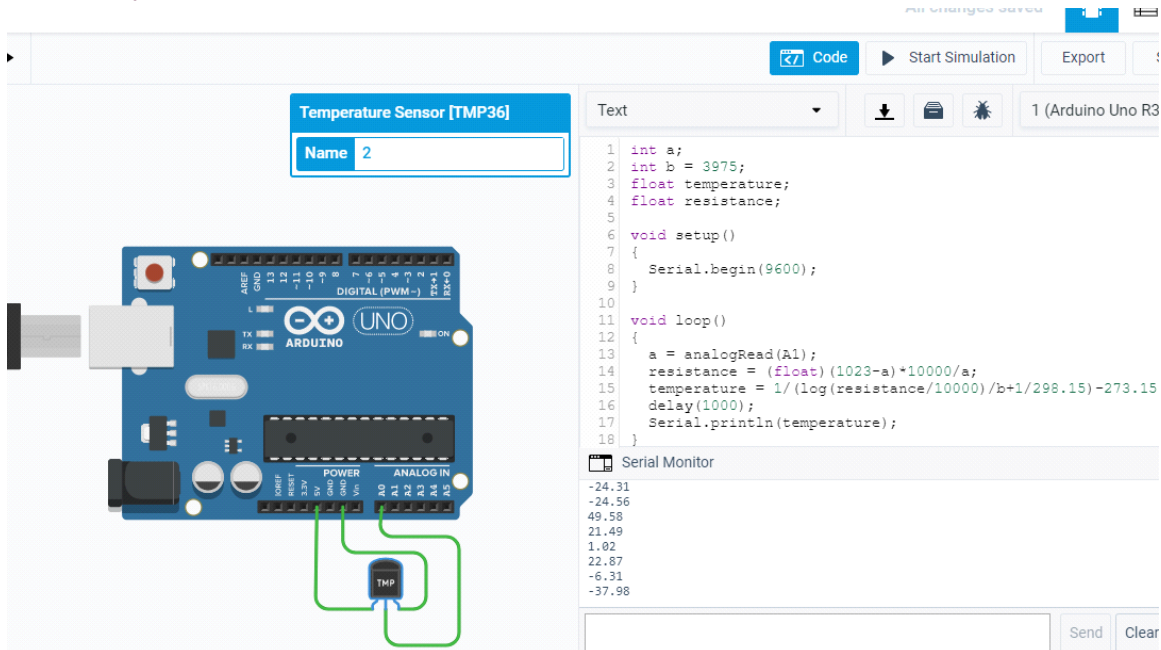
Using the sound sensor helps control other elements and produce outputs depending upon the different tones detected by the sensor.

Hito 2.6



The light sensor detects the intensity of ambient light in the surrounding and depending upon it, controls other devices enabling the circuit to lower the blinds and turn on the lights.

Hito 2.7



The experiment helped us gain understanding of the working of a temperature sensor which can be used to detect temperature in a room and display its value in our choice of

scale(Celsius or Fahrenheit).

Hito 2.8

Simulator time: 00:00:30

Code Stop Simulation

Potentiometer

Name 1

Resistance 250 kOhm

```
4 int potentiometer = 0;
5 int shaft;
6
7 void setup()
8 {
9   Serial.begin(9600);
10  myServo.attach(3);
11  pinMode(potentiometer, INPUT);
12 }
13
14 void loop()
15 {
16   shaft = analogRead(potentiometer);
17   shaft = map(shaft, 0, 500, 0, 179);
18   myServo.write(shaft);
19   delay(1500);
20   Serial.println(myServo.read());
21 }
```

Serial Monitor

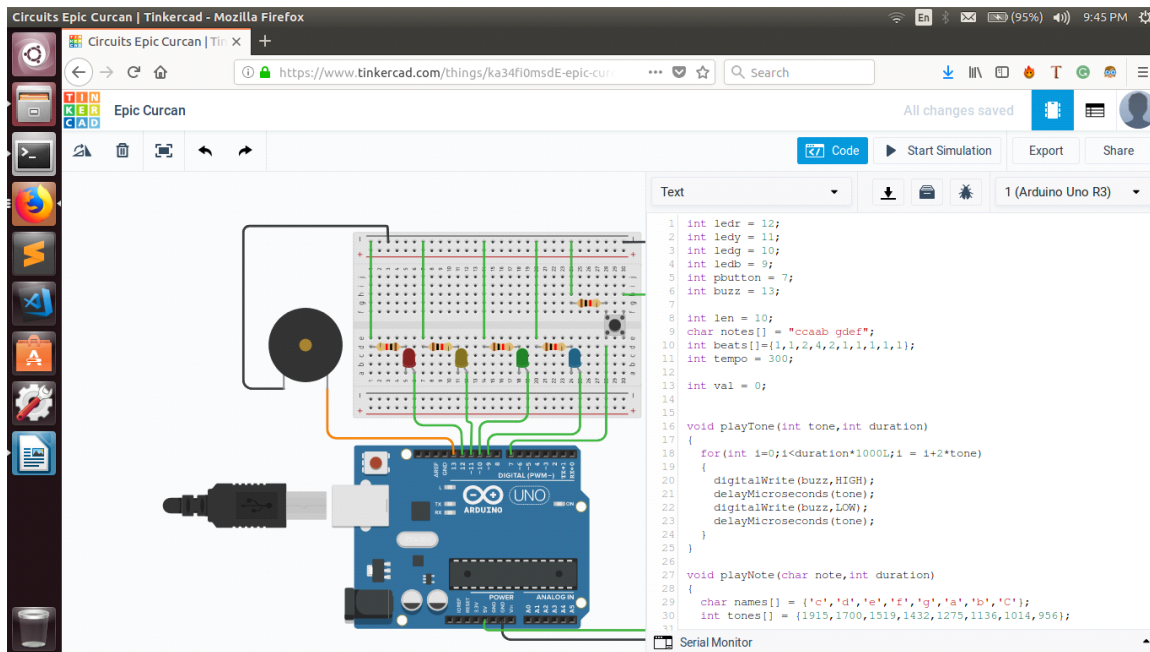
180
95
88
44
36
0
0
0

The concept of Servo might take some time to wrap our heads around but once understood, it can prove to be one of the most useful tools in the Arduino circuitry. It maps the value of potentiometer rotation to the a mechanical shaft that helps rotate it.

Traffic Light System Simulation

The traffic light system can be described using a state transition diagram which depicts which lights to turn ON/OFF depending upon the current condition which can be registered using a Push button.

Working on this task was a bit challenging as it was our first fully fledged experiment using various functionalities and components that we have been learning up till now.



Circuits Epic Curcan | Tinkercad - Mozilla Firefox

https://www.tinkercad.com/things/ka34fi0msdE-epic-cur...

Epic Curcan

All changes saved

Code Start Simulation Export Share

Text 1 (Arduino Uno R3)

```
28 {
29   char names[] = {'c','d','e','f','g','a','b','c'};
30   int tones[] = {1915,1700,1519,1432,1275,1136,1014,956};
31
32   for(int i=0;i<8;i++)
33   {
34     if(names[i]==note)
35       playTone(tones[i],duration);
36   }
37 }
38
39
40
41
42
43
44 void setup()
45 {
46   pinMode(ledr,OUTPUT);
47   pinMode(ledy,OUTPUT);
48   pinMode(ledg,OUTPUT);
49   pinMode(buzz,OUTPUT);
50   pinMode(pbutton,INPUT);
51 }
52
53
54
55
56 void loop()
57 {
58 }
```

Serial Monitor

Circuits Epic Curcan | Tinkercad - Mozilla Firefox

https://www.tinkercad.com/things/ka34fi0msdE-epic-cur...

Epic Curcan

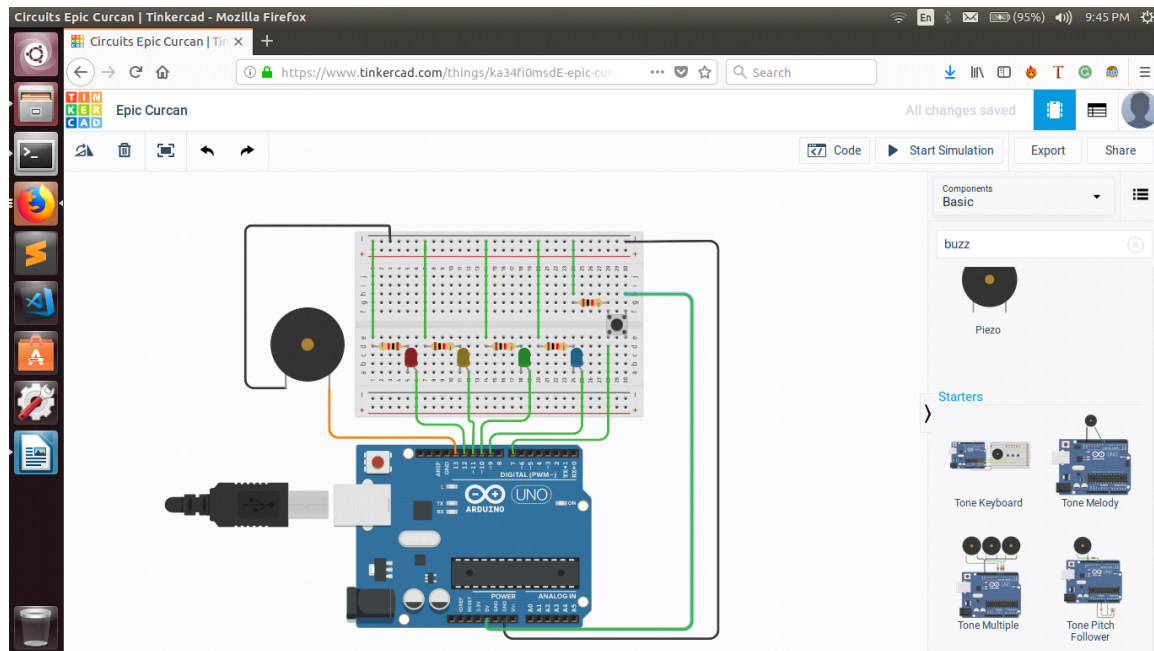
All changes saved

Code Start Simulation Export Share

Text 1 (Arduino Uno R3)

```
58
59 digitalWrite(ledg,HIGH);
60 val = digitalRead(pbutton);
61 if(val==1)
62 {
63   digitalWrite(ledg,LOW);
64   digitalWrite(ledb,HIGH);
65   for(int i=0;i<10;i++)
66   {
67     digitalWrite(ledy,HIGH);
68     delay(500);
69     digitalWrite(ledy,LOW);
70     delay(500);
71   }
72
73   digitalWrite(ledy,HIGH);
74   delay(4000);
75   digitalWrite(ledy,LOW);
76   digitalWrite(ledb,LOW);
77
78
79   digitalWrite(ledr,HIGH);
80   for(int i=0;i<len;i++)
81   {
82     if(names[i]!=' ')
83       delay(beats[i]*tempo);
84     else
85       playNote(names[i],beats[i]*tempo);
86   }
87   delay(tempo/2);
88 }
```

Serial Monitor



Working Hours

This time we spent the time as recorded below in completing this notebook.

Prakriti - 5 hours

Phanith - 2.5 hours

Team Experience

It took a while to complete the task but it helped us gain hands-on knowledge of using the components and writing the code that would enable the circuit to perform in sync and change state according to the users' requirements.