

Midterm Proposal

...

Stella Kim



Problem

Children's Toy



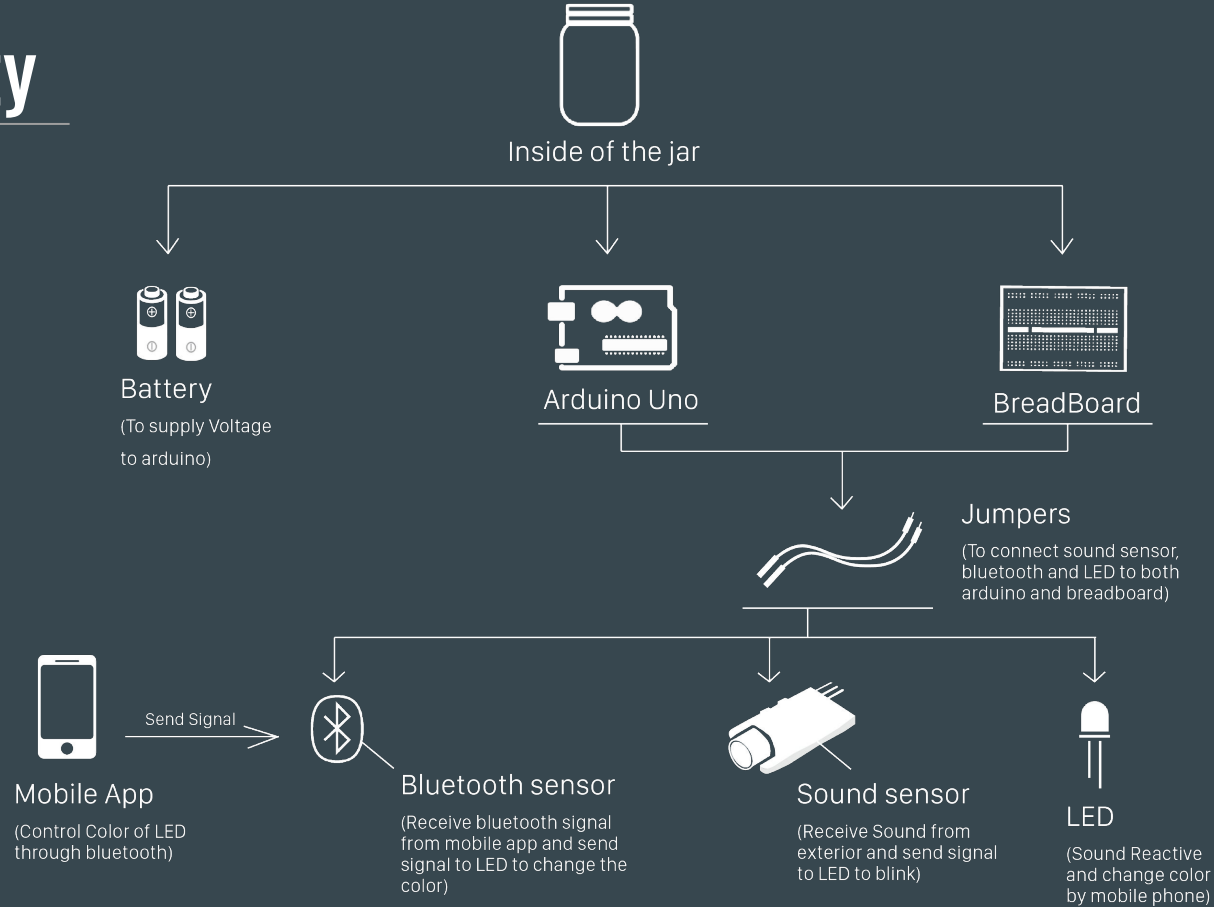
Children need a fun and realistic toy for upcoming Christmas Day. Adults want to give a fantasy feeling to their children to celebrate their holiday. This project is about both fun and interactive “Tinker bell Jar.” This Tinker bell Jar is coming with a music reactive LED light inside which increases the fairy tale mood. The background music of Tinkerbell will play and the light will follow the rhythm. Moreover, you can change your color of the light in mobile app. Kids can choose whatever color they want and display Tinker bell jar with different colors. Let Tinker bell presents children a fantasy!

Tinkerbell Jar

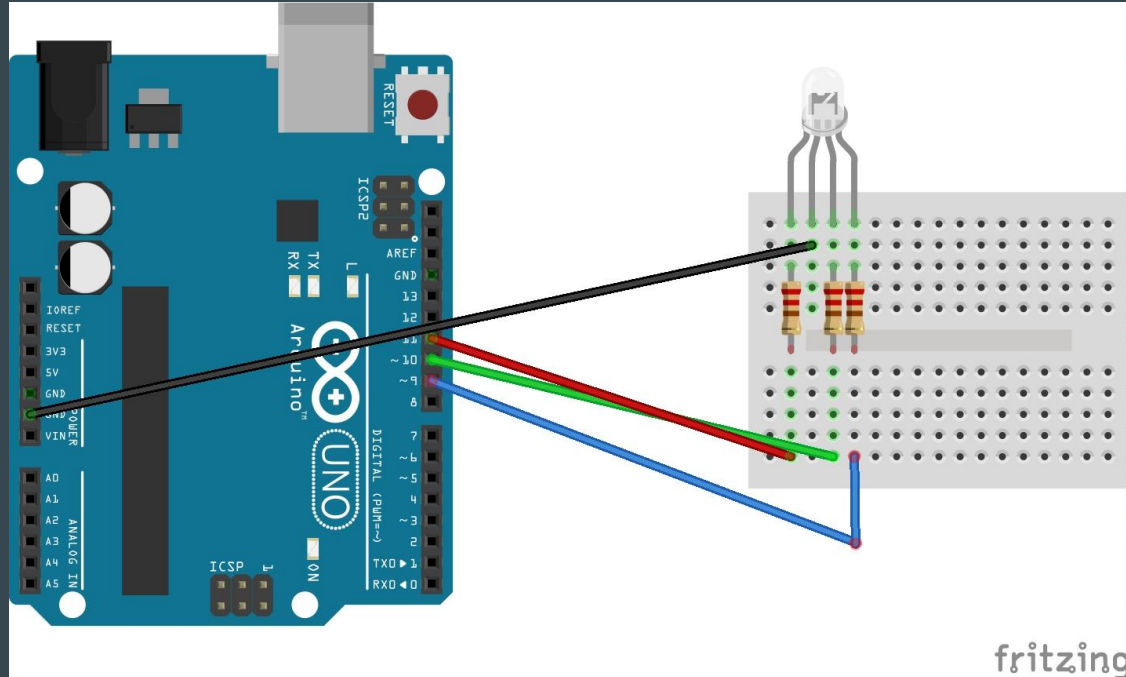


- A laptop computer
- Arduino Uno R3
- USB cable
- 5V power supply
- RGB LED
- Sound Sensor
- Bluetooth HC-05
- Three Resistors
- Joint Cables
- Glass mason jar & other elaborative papers

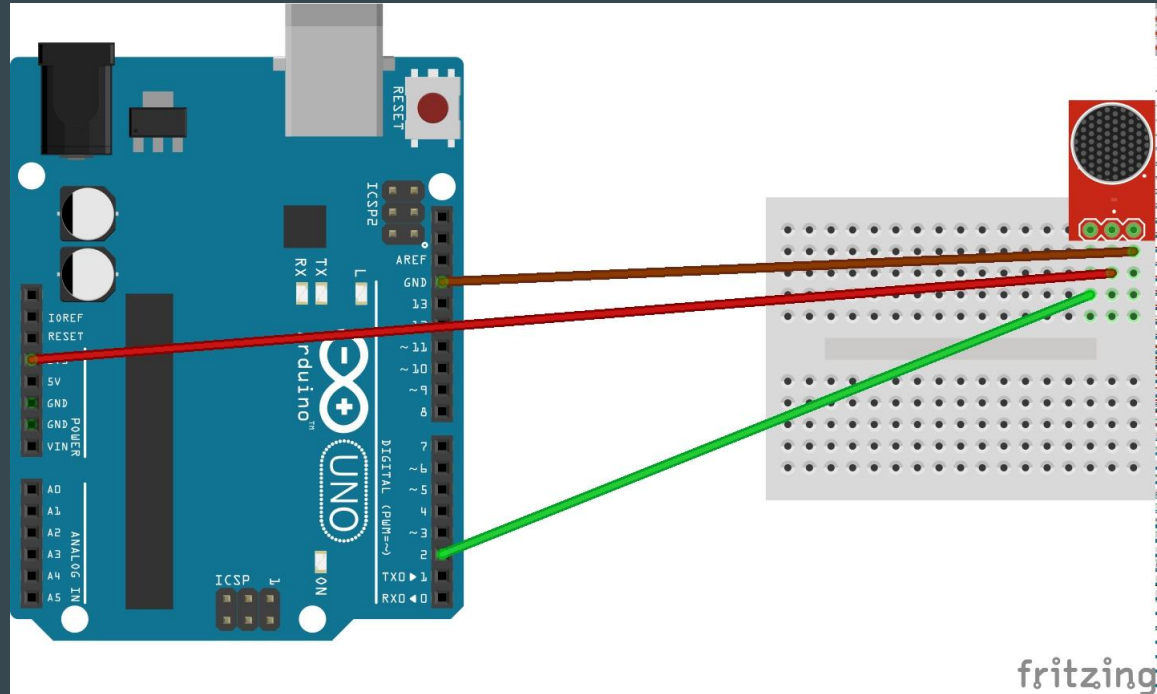
Connectivity



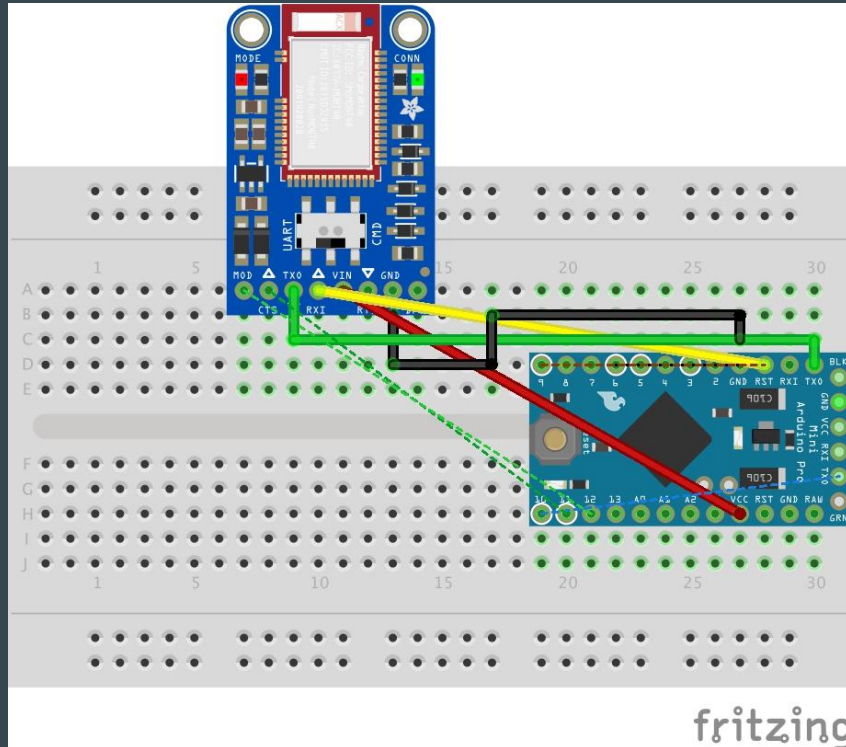
Low level hardware setup - LED



Low level hardware setup - Sound Sensor



Low level hardware setup - Bluetooth HC-05



5 Steps



1. Turn on Arduino
2. Lights RGB LED
3. Connects to the Sound Sensor
4. Connects to the Bluetooth
5. Combine all the three features

Steps Detail_1

1. Turn on Arduino

- Connect USB to laptop and Arduino.
- Turn on the Arduino app and upload the codes

2. Lights RGB LED

- Connect GND of LED to GND pin in Arduino. Red light connects to pin 8. Blue light connects to pin 9. Green light connects to pin 10.
- Connect three resistors to red, blue and green light parts.
- Upload the code for RGB LED
- Check if the LED turn on the lights

3. Turn on the Sound Sensor

- Connect input pin(S) to pin A5. Connect voltage pin(+) to 5v voltage part in the breadboard. Connect GND pin(-) to GND pin in Arduino.
- Now it's time to connect LED and sound sensor
- As the sound sensor gets high frequency, the LED will turn on color
- As the sound sensor gets low frequency, the LED will turn off the color
- Upload the code
- LED is sound reactive to the sound sensor

Steps Detail_2

4. Turn on the Bluetooth Sensor

- Connect RXD pin to pin 12. Connect TXD pin to pin 11. Connect GND pin to GND pin in Arduino. Connect VCC to 5V part in breadboard.
- Now it's time to connect bluetooth sensor and LED.
- This bluetooth sensor will connect mobile app and LED. This will enable user to control color of LED by just touching the color wheel displayed in mobile app.
- Upload the code
- LED is able to be controlled in mobile app.

5. Combine All the features

- When the music plays, the sound sensor will react to it. The LED starts to blink as the music plays. It follows the rhythm. As User sees the LED activates, he/she can change the color with the mobile app.
- The step in this part is just about the coding.

Code for LED



```
sketch_oct21a | 아두이노 1.6.13 Hourly Build
sketch_oct21a $
1 int redPin = 11;
2 int greenPin = 10;
3 int bluePin = 9;
4
5 void setup()
6 {
7   pinMode(redPin, OUTPUT);
8   pinMode(greenPin, OUTPUT);
9   pinMode(bluePin, OUTPUT);
10 }
11
12 void loop()
13 {
14   setColor(255, 0, 0); // red
15   delay(1000);
16   setColor(0, 255, 0); // green
17   delay(1000);
18   setColor(0, 0, 255); // blue
19   delay(1000);
20   setColor(255, 255, 0); // yellow
21   delay(1000);
22   setColor(80, 0, 80); // purple
23   delay(1000);
24   setColor(0, 255, 255); // aqua
25   delay(1000);
26   setColor(0x48, 0x0, 0x82);
27   delay(1000);
28
29 }
30
31 void setColor(int red, int green, int blue)
32 {
33   analogWrite(redPin, red);
34   analogWrite(greenPin, green);
35   analogWrite(bluePin, blue);
36 }
```

Code for Sound Sensor



```
sketch_oct21a | 아두이노 1.6.13 Hourly Build 2016/10/17 0...  
sketch_oct21a §  
1 int sound_sensor = 2;  
2 int RLED = 11;  
3 int GLED = 10;  
4 int BLED = 9;  
5  
6 void setup() {  
7   pinMode(sound_sensor, INPUT);  
8   pinMode(RLED, OUTPUT);  
9   pinMode(GLED, OUTPUT);  
10  pinMode(BLED, OUTPUT);  
11  
12 }  
13  
14 void loop() {  
15   int status_sensor = digitalRead(sound_sensor);  
16   if (status_sensor == 1){  
17     digitalWrite(RLED, LOW);  
18     digitalWrite(GLED, LOW);  
19     digitalWrite(BLED, LOW);  
20   }  
21   else {  
22     digitalWrite(RLED, HIGH);  
23     digitalWrite(GLED, HIGH);  
24     digitalWrite(BLED, HIGH);  
25   }  
26 }  
27  
28  
29 }  
30  
30  
Arduino/Genuino Uno on /dev/cu.usbmodem23
```

Code for BlueTooth



```
sketch_oct21a | 아두이노 1.6.13 Hourly Build 2016/10/17 0...
sketch_oct21a $
1 #include <SoftwareSerial.h>
2 SoftwareSerial BLU(0,1);
3 #define RLED 11
4 #define GLED 10
5 #define BLEED 9
6 void setup()
7 {
8   //Serial setup
9   Serial.begin(9600);
10  Serial.println("- HC-05 Bluetooth RGB LED =-");
11  BLU.begin(9600);
12  BLU.println("- HC-05 Bluetooth RGB LED =-");
13
14  pinMode(4, OUTPUT);
15  pinMode(RLED, OUTPUT);
16  pinMode(GLED, OUTPUT);
17  pinMode(BLEED, OUTPUT);
18  digitalWrite(4,HIGH);
19  setColor(255, 0, 0);
20  delay(500);
21  setColor(0, 255, 0);
22  delay(500);
23  setColor(0, 0, 255);
24  delay(500);
25  setColor(255, 255, 255);
26 }
27 void loop()
28 {
29   while (BLU.available() > 0)
30   {
```

```
sketch_oct21a | 아두이노 1.6.13 Hourly Build 2016/10/17 0...
sketch_oct21a $
27 void loop()
28 {
29   while (BLU.available() > 0)
30   {
31     int redInt = BLU.parseInt();
32     int greenInt = BLU.parseInt();
33     int blueInt = BLU.parseInt();
34     redInt = constrain(redInt, 0, 255);
35     greenInt = constrain(greenInt, 0, 255);
36     blueInt = constrain(blueInt, 0, 255);
37     if (BLU.available() > 0)
38     {
39       setColor(redInt, greenInt, blueInt);
40       Serial.print("Red: ");
41       Serial.print(redInt);
42       Serial.print(" Green: ");
43       Serial.print(greenInt);
44       Serial.print(" Blue: ");
45       Serial.print(blueInt);
46       Serial.println();
47       BLU.flush();
48     }
49   }
50 }
51 void setColor(int red, int green, int blue)
52 {
53   analogWrite(RLED, red);
54   analogWrite(GLED, green);
55   analogWrite(BLEED, blue);
56 }
```

Code for Combination



```
sketch_oct21b | 아두이노 1.6.13 Hourly Build 2016/10/17 0...
sketch_oct21b $
28 delay(500);
29 setColor(0, 0, 255);
30 delay(500);
31 setColor(255, 255, 255);
32
33
34 }
35
36 void loop() {
37
38   while (BLU.available() > 0)
39   {
40     int redInt = BLU.parseInt();
41     int greenInt = BLU.parseInt();
42     int blueInt = BLU.parseInt();
43     redInt = constrain(redInt, 0, 255);
44     greenInt = constrain(greenInt, 0, 255);
45     blueInt = constrain(blueInt, 0, 255);
46     if (BLU.available() > 0)
47     {
48       setColor(redInt, greenInt, blueInt);
49       Serial.print("Red: ");
50       Serial.print(redInt);
51       Serial.print(" Green: ");
52       Serial.print(greenInt);
53       Serial.print(" Blue: ");
54       Serial.print(blueInt);
55       Serial.println();
56       BLU.flush();
57     }
58   }
59 }
```

```
sketch_oct21b | 아두이노 1.6.13 Hourly Build 2016/10/17 0...
sketch_oct21b $
28 delay(500);
29 setColor(0, 0, 255);
30 delay(500);
31 setColor(255, 255, 255);
32
33
34 }
35
36 void loop() {
37
38   while (BLU.available() > 0)
39   {
40     int redInt = BLU.parseInt();
41     int greenInt = BLU.parseInt();
42     int blueInt = BLU.parseInt();
43     redInt = constrain(redInt, 0, 255);
44     greenInt = constrain(greenInt, 0, 255);
45     blueInt = constrain(blueInt, 0, 255);
46     if (BLU.available() > 0)
47     {
48       setColor(redInt, greenInt, blueInt);
49       Serial.print("Red: ");
50       Serial.print(redInt);
51       Serial.print(" Green: ");
52       Serial.print(greenInt);
53       Serial.print(" Blue: ");
54       Serial.print(blueInt);
55       Serial.println();
56       BLU.flush();
57     }
58   }
59 }
```

```
sketch_oct21b | 아두이노 1.6.13 Hourly Build 2016/10/17 0...
sketch_oct21b $
54   Serial.print(blueInt);
55   Serial.println();
56   BLU.flush();
57 }
58 }
59
60 int status_sensor = digitalRead(sound_sensor);
61 if (status_sensor == 1){
62   digitalWrite(4,LOW);
63   digitalWrite(BLED, LOW);
64   digitalWrite(GLED, LOW);
65   digitalWrite(BLED, LOW);
66 }
67
68 else {
69   digitalWrite(4,HIGH);
70   digitalWrite(BLED, HIGH);
71   digitalWrite(RLED, HIGH);
72   digitalWrite(GLED, HIGH);
73 }
74 }
75
76 }
77
78 void setColor(int red, int green, int blue)
79 {
80   analogWrite(RLED, red);
81   analogWrite(GLED, green);
82   analogWrite(BLED, blue);
83 }
```

Second_Attempt - Color Change



```
color_change_new
1 #include <SoftwareSerial.h>
2 #include <Wire.h> //Include libraries: SoftwareSerial & Wire
3 SoftwareSerial BT(11,12); //Define PIN11 & PIN12 as RX and TX pin
4 |
5 //RGB LED Pins
6 int R_LED = 8;
7 int G_LED = 10;
8 int B_LED = 9;
9 //RED LED at Pin 13
10 int RED_LED = 13;
11 String RGB = ""; //store RGB code from BT
12 String RGB_Previous = "255.255.255"; //preserve previous RGB color
13 String ON = "ON"; //Check if ON command is received
14 String OFF = "OFF"; //Check if OFF command is received
15 boolean RGB_Completed = false;
16
17 void setup() {
18   Serial.begin(9600); //Arduino serial port baud rate:9600
19   BT.begin(9600); //My HC-05 module default baud rate is 9600
20   RGB.reserve(30);
21
22   pinMode(RED_LED, OUTPUT);
23   //Set pin13 as output for LED,
24 }
25
26 void loop() {
27   // put your main code here, to run repeatedly:
28
29   //Read each character from Serial Port(Bluetooth)
30   while(BT.available()){
31     char ReadChar = (char)BT.read();
32
33     // Right parentheses ) indicates complet of the string
34     if(ReadChar == ' '){
35       RGB_Completed = true;
36     }
37   }
38
39   //When a command code is received completely with ' ' ending character
40   if(RGB_Completed){
41     //Print out debug info at Serial output window
42     Serial.print("RGB:");
43     Serial.print(RGB);
44     Serial.print("   PreRGB:");
45     Serial.println(RGB_Previous);
46
47     if(RGB==ON){
48       digitalWrite(13,HIGH);
49       RGB = RGB_Previous; //We only receive 'ON', so get previous RGB color
50       Light_RGB_LED();
51     }
52     else if(RGB==OFF){
53       digitalWrite(13,LOW);
54       RGB = "0.0.0"; //Send OFF string to turn light off
55       Light_RGB_LED();
56     }
57     else{
58       //Turn the color according the color code from Bluetooth
59       Light_RGB_LED();
60       RGB_Previous = RGB;
61     }
62   }
63   //Reset RGB String
64
65   RGB = "";
66   RGB_Completed = false;
67 }
```

```
color_change_new
33 // Right parentheses ) indicates complet of the string
34 if(ReadChar == ' '){
35   RGB_Completed = true;
36 }else{
37   RGB += ReadChar;
38 }
39 }
40
41 //When a command code is received completely with ' ' ending character
42 if(RGB_Completed){
43   //Print out debug info at Serial output window
44   Serial.print("RGB:");
45   Serial.print(RGB);
46   Serial.print("   PreRGB:");
47   Serial.println(RGB_Previous);
48
49   if(RGB==ON){
50     digitalWrite(13,HIGH);
51     RGB = RGB_Previous; //We only receive 'ON', so get previous RGB color
52     Light_RGB_LED();
53   }
54   else if(RGB==OFF){
55     digitalWrite(13,LOW);
56     RGB = "0.0.0"; //Send OFF string to turn light off
57     Light_RGB_LED();
58   }
59   else{
60     //Turn the color according the color code from Bluetooth
61     Light_RGB_LED();
62     RGB_Previous = RGB;
63   }
64   //Reset RGB String
65
66   RGB = "";
67   RGB_Completed = false;
68 }
```

```
color_change_new
64 //Reset RGB String
65 RGB = "";
66 RGB_Completed = false;
67
68 } //end if of check if RGB completed
69
70 } // end of loop
71
72 void Light_RGB_LED(){
73   int SP1 = RGB.indexOf('.');
74   int SP2 = RGB.indexOf('.', SP1+1);
75   int SP3 = RGB.indexOf('.', SP2+1);
76   String R = RGB.substring(0, SP1);
77   String G = RGB.substring(SP1+1, SP2);
78   String B = RGB.substring(SP2+1, SP3);
79
80   //Print out debug info at Serial output window
81   Serial.print("R=");
82   Serial.println( constrain(R.toInt(),0,255));
83   Serial.print("G=");
84   Serial.println( constrain(G.toInt(),0,255));
85   Serial.print("B=");
86   Serial.println( constrain(B.toInt(),0,255));
87   //Light up the LED with color code
88   analogWrite(R_LED, (R.toInt()));
89   analogWrite(G_LED, (G.toInt()));
90   analogWrite(B_LED, (B.toInt()));
91 }
92
93
94
95
96
97
```

Second_Attempt - Combination



color_change_new_2

```
1 #include <SoftwareSerial.h>
2 #include <Wire.h> //Include libraries: SoftwareSerial & Wire
3 SoftwareSerial BT(11,12); //Define PIN11 & PIN12 as RX and TX pins
4
5 int sound_sensor = 2;
6 //RGB LED Pins
7 int R_LED = 8;
8 int G_LED = 10;
9 int B_LED = 9;
10 //RED LED at Pin 13
11 int RED_LED = 13;
12 String RGB = ""; //store RGB code from BT
13 String RGB_Previous = "255.255.255"; //preserve previous RGB code
14 String ON = "ON"; //Check if ON command is received
15 String OFF = "OFF"; //Check if OFF command is received
16 boolean RGB_Completed = false;
17
18 void setup() {
19   Serial.begin(9600); //Arduino serial port baud rate:9600
20   BT.begin(9600); //My HC-05 module default baud rate is 9600
21   RGB.reserve(30);
22
23   pinMode(sound_sensor, INPUT);
24   pinMode(RED_LED, OUTPUT);
25   //Set pin13 as output for LED,
26 }
27
28 void loop() {
29   // put your main code here, to run repeatedly:
30
31   int status_sensor = digitalRead(sound_sensor);
32   if (status_sensor == 1){
33     digitalWrite(13, LOW);
34   }
35 }
```

color_change_new_2

```
36
37 //Read each character from Serial Port(Bluetooth)
38 while(BT.available()){
39   char ReadChar = (char)BT.read();
40
41   // Right parentheses ) indicates complet of the string
42   if(ReadChar == ')'){
43     RGB_Completed = true;
44   }else{
45     RGB += ReadChar;
46   }
47 }
48
49 //When a command code is received completely with ')' ending
50 if(RGB_Completed){
51   //Print out debug info at Serial output window
52   Serial.print("RGB:");
53   Serial.print(RGB);
54   Serial.print("   PreRGB:");
55   Serial.println(RGB_Previous);
56
57   if(RGB=="ON"){
58     digitalWrite(13,HIGH);
59     RGB = RGB_Previous; //We only receive 'ON', so get previous RGB code
60     Light_RGB_LED();
61   }else if(RGB=="OFF"){
62     digitalWrite(13,LOW);
63     RGB = "0.0.0"; //Send OFF string to turn light off
64     Light_RGB_LED();
65   }else{
66     //Turn the color according the color code from Bluetooth
67     Light_RGB_LED();
68     RGB_Previous = RGB;
69   }
70 }
```

color_change_new_2

```
71
72 //end if of check if RGB completed
73
74 }
75
76 else {
77   digitalWrite(13, HIGH);
78 }
79
80 // end of loop
81
82 void Light_RGB_LED(){
83
84   int SP1 = RGB.indexOf('.');
85   int SP2 = RGB.indexOf('.', SP1+1);
86   int SP3 = RGB.indexOf('.', SP2+1);
87   String R = RGB.substring(0, SP1);
88   String G = RGB.substring(SP1+1, SP2);
89   String B = RGB.substring(SP2+1, SP3);
90
91   //Print out debug info at Serial output window
92   Serial.print("R=");
93   Serial.println( constrain(R.toInt(),0,255));
94   Serial.print("G=");
95   Serial.println( constrain(G.toInt(),0,255));
96   Serial.print("B=");
97   Serial.println( constrain(B.toInt(),0,255));
98   //Light up the LED with color code
99   analogWrite(R_LED, (R.toInt()));
100   analogWrite(G_LED, (G.toInt()));
101   analogWrite(B_LED, (B.toInt()));
102 }
```


Thank You

