

LED Glowing-Fading control

Sayan Sarkar*

July 10, 2021

Arduino Project

Department of Physics

Asutosh College

92 Shyamaprasad Mukherjee Road, Kolkata 700026,India

Affiliated to



Abstract

In this project "*Fading-Glowing Of LEDs*" are controlled using Serial Monitor and in addition to that corresponding texts are shown in LCD Panel. The Fading of LEDs are shown by reducing the brightness of the LEDs continuously by a small amount repeatedly until the LED finally shuts off. When finally the LED shuts then the brightness slowly gets increased until it gets full brightness. Thus, a LED continuously starts glowing and fading accordingly after User chooses the desired LED.

*equinoxxtreme@gmail.com

Contents

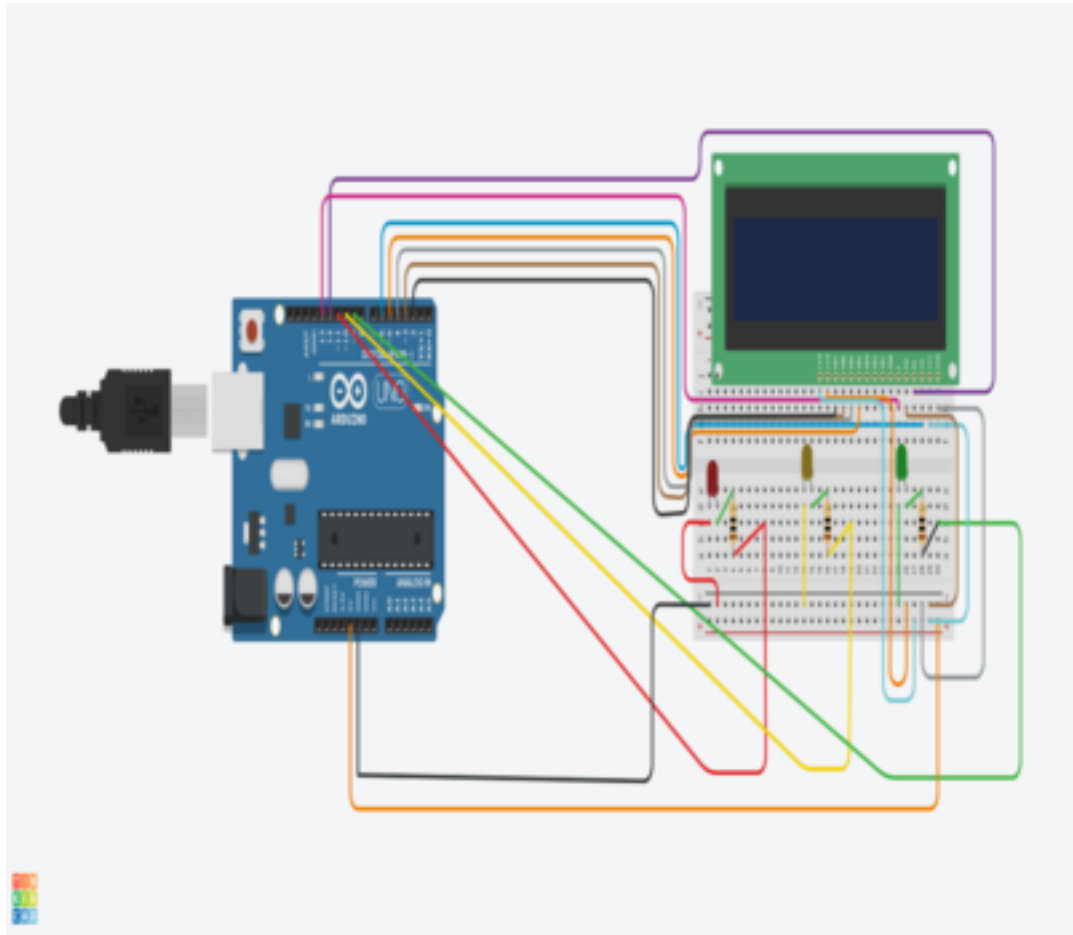
1	Introduction	3
2	Methodology	3
2.1	<u>CKT Diagram</u>	3
2.2	<u>Hardware Used</u>	4
2.3	<u>Arduino Code</u>	4
2.4	<u>Brief Description</u>	7
3	Results and Discussion	8
4	Conclusion	9
5	Acknowledgement	9

1 Introduction

Main Goal of this project is to take input from user in the serial monitor. The fading-Glowing can be of two types i.e. each LED will stop after glowing and fading for once OR we can keep the LED glowing and fading continuously unless User interrupts. We will continue with the latter. Through the input option User can select the desired LED that is to be Glow and Fade continuously. When we will choose another LED our previous LED should stop and newly selected LED must start glowing and fading. If the user gives any wrong input, an error message is to be shown.

2 Methodology

2.1 CKT Diagram



The Link of Tinkercad File : [Click Here](#).

2.2 Hardware Used

1. Breadboard
2. Jumper Wires
3. Red LED
4. Green LED
5. Yellow LED
6. 3 Resistors of 220 Ω
7. Arduino UNO
8. LCD Screen

2.3 Arduino Code

```
1 #include <LiquidCrystal.h>
2 int Contrast = 68;
3 LiquidCrystal lcd(12, 13, 5, 4, 3, 2);
4 const int redPin = 11;
5 const int yellowPin = 10;
6 const int greenPin = 9;
7 int brightness_1 = 0;
8 int fadeAmount_1 = 5;
9 int brightness_2 = 0;
10 int fadeAmount_2 = 5;
11 int brightness_3 = 0;
12 int fadeAmount_3 = 5;
13 bool val_1 = true;
14 bool val_2 = true;
15 bool val_3 = true;
16 char comdata;
17 void setup() {
18     // put your setup code here, to run once:
19     pinMode(redPin, OUTPUT);
20     pinMode(greenPin, OUTPUT);
21     pinMode(yellowPin, OUTPUT);
22     analogWrite(6, Contrast);
23     lcd.begin(16, 2);
24     Serial.begin(9600);
25     Serial.println("Press_B_to_start_the_program");
26     Serial.println("Then_press_R_-_RED_ON,_G_-_GREEN_ON,_Y_
        _YELLOW_ON,_r_-_TO_RESET,_X_-_ALL_is_OFF_");
27     Serial.print("choose_the_letter:_");
28 }
29
30 void loop() {
31     // put your main code here, to run repeatedly:
```

```
32 if (Serial.available() > 0) {
33   comdata = char(Serial.read());
34   if (comdata == 'R' || comdata == 'G' || comdata == 'Y'
       || comdata == 'X' || val_1 == true || val_2 ==
       true || val_3 == true || comdata == 'B' || Serial.
       available() == 0) {
35     if (comdata == 'B') {
36       Serial.println("Your_Program_Has_Started");
37       lcd.setCursor(0, 0);
38       lcd.print("Program_Started");
39     }
40     else if (comdata == 'X') {
41       Serial.println("ALL_LEDS_ARE_OFF");
42       digitalWrite(redPin, LOW);
43       digitalWrite(greenPin, LOW);
44       digitalWrite(yellowPin, LOW);
45       lcd.setCursor(0, 0);
46       lcd.clear();
47       lcd.print("TURNED_OFF");
48     }
49     while (comdata == 'R' || val_1) {
50       if (comdata == 'R')
51       {
52         Serial.println("Red_LED_is_ON");
53         lcd.setCursor(0, 1);
54         lcd.clear();
55         lcd.print("RED_LED_IS_ON");
56       }
57       analogWrite(redPin, brightness_1);
58       brightness_1 = brightness_1 + fadeAmount_1;
59       if (brightness_1 == 0 || brightness_1 == 255)
60       {
61         fadeAmount_1 = -fadeAmount_1;
62       }
63       delay(30);
64       if (Serial.available() > 0)
65       {
66         digitalWrite(redPin, LOW);
67         val_1 = !val_1;
68         break;
69       }
70     }
71     while (comdata == 'G' || val_2) {
72       if (comdata == 'G')
73       {
74         Serial.println("Green_LED_is_ON");
75         lcd.setCursor(0, 1);
76         lcd.clear();
77         lcd.print("GREEN_LED_IS_ON");
78         //delay(2000);
```

```
79     }
80     analogWrite(greenPin, brightness_2);
81     brightness_2 = brightness_2 + fadeAmount_2;
82     if (brightness_2 == 0 || brightness_2 == 255)
83     {
84         fadeAmount_2 = -fadeAmount_2;
85     }
86     delay(30);
87     if (Serial.available() > 0)
88     {
89         digitalWrite(greenPin, LOW);
90         val_2 = !val_2;
91         break;
92     }
93 }
94 while (comdata == 'Y' || val_3) {
95     if (comdata == 'Y')
96     {
97         Serial.println("Yellow_LED_is_ON");
98         lcd.setCursor(0, 1);
99         lcd.clear();
100         lcd.print("YELLOW_LED_IS_ON");
101         //delay(2000);
102     }
103     analogWrite(yellowPin, brightness_3);
104     brightness_3 = brightness_3 + fadeAmount_3;
105     if (brightness_3 == 0 || brightness_3 == 255)
106     {
107         fadeAmount_3 = -fadeAmount_3;
108     }
109     delay(30);
110     if (Serial.available() > 0)
111     {
112         digitalWrite(yellowPin, LOW);
113         val_3 = !val_3;
114         break;
115     }
116 }
117 }
118 else
119 {
120     Serial.println("Please_Use_Above_Mentioned_KeyWords");
121     lcd.setCursor(0, 1);
122     lcd.clear();
123     lcd.print("WRONG_INPUT");
124 }
125 }
126 }
```

2.4 Brief Description

One of our motto of this project is to show the corresponding states in an LCD Screen. So, for that we need to call a library named : *LiquidCrystal.h* After that, we are assigning arduinos digital pins : 12, 13, 5, 4, 3, 2 to lcd screen and Pin NO. 11, 10, 9 are assigned to respectively to RED, YELLOW, GREEN LED. For each LEDs brightness amount to 0 and fade amount to 5. And 3 Boolean variables are declared having TRUE. And another variable comdata of character datatype is declared.

Then the arduino code is to setup with pin mode of respective LEDs and lcd Screen needs to be started using *lcd.begin()* function. After that requirements to run the program is shown on the *Serial Monitor* and user is asked to give input accordingly. In the loop section we are going to check if user has given any input by the function *Serial.available()* . If any value is available in the serial monitor then it is stored in comdata variable.

Then comdata value is checked with certain conditions that is if it is any of B, R, G, Y, X then we are entering the loop. But the condition doesn't end here we need to check for the boolean values too and also if *Serial.available()* is 0. We're going to enter the loop because for each turn *Serial.available* is going to be zero so the loop will break resulting shutting of the LED after one Fading-Glowing.

Now User can check if the program has started yet or not by inputting 'B'. User can also turn off all the LEDs by giving 'X'. And for the main part in a while loop we are going to check if comdata has value R, G or Y and with each of them their corresponding assigned boolean variables and that's how the loop doesn't stop even if comdata has no value given.

And with these piece of code :

```
1 brightness = brightness + fadeAmount;
2 if (brightness == 0 || brightness == 255)
3 {
4   fadeAmount = -fadeAmount;
5 }
```

we're actually controlling the brightness of the LED . These is the main logic that makes the LEDs behave like fading and brightening continuously. Let's see what's the logic about:

Initially our brightness variable has value 0. Now we are increasing it's value by fadeamount that is 5. As our brightness variable has now value 5 it doesn't enter the if condition. And it gets increased with 5 amount each time the loop runs. When the overall sum of brightness and fadeamount reaches 255 i.e. when brightness becomes 255 it finally satisfies the if condition loop and enters it. Here the sign of fade amount gets changed . As a result fade amount is -5 now. and thus brightness decreases by 5 amount and when it finally reaches 0 it again satisfies the if condition and again enters the if and our fade amount's sign gets changed . So, it's +5 again and the first process of glowing the LED occurs and it repeats infinitely . So, now we need to stop this loop. And for that:

```
1 if (Serial.available() > 0)
2 {
3   digitalWrite(LEDPin, LOW);
4   val = !val;
```

```
5 break;
6 }
```

Within that while loop another if condition is introduced to check if user has given any input or not. And if any input is available this above mentioned condition gets satisfied and using *digitalWrite()* the used LED is stopped i.e. turns into low and our boolean value is reversed and thus the loop gets broken.

3 Results and Discussion

Our program looks so far so good it works fine. But I need to mention two points firstly when I wrote the code for the first time I decided to write using a for loop inside a if condition but while dealing with it I noticed that it stops after completing a full cycle of glowing-fading. And that is because when next time it starts running our serial.available is null then so it doesn't fulfill our condition and hence stops working. Then after a quite bit of thinking I changed my way of approaching the problem and rather using if then for I used a if statement to check if comdata has desired values or its null that's how I overcame the problem of running the program for second time. But the problem doesn't end here. Now I need a while loop to make the led to glow-fade infinitely. Then I thought inside while if I check only a particular condition that is not going to give me an infinite loop. Then an idea struck my mind few days before I was working on a Python Project of creating a snake game there I used an infinite while loop that checks if a variable has the boolean value true or not and inside that loop I used functions to detect if that snake has collided with a food or collided with itself or hits the wall and increase its length and I assign that variable to False when it satisfies certain condition. And that's how I came up with the idea of using a boolean variable to check along with the previous condition. Now, one can ask that we could also use the condition that is *series.available == 0* but the problem is then after completing a cycle our program will be confused because the same condition will be used for Red, Green, Yellow LED too. So, we simply can't use that. And then after reading the question again I realized that it is not mentioned clearly that the LED is needed to glow-fade continuously then I consulted our H.O.D S.S.B sir what should I do? then he suggested to go with the latter and keep the loop running unless another input is given by the User. Felt kinda relieved I didn't waste my time. Then sir suggested I should do something when the user gives some invalid input. So, Sir kind of did the debugging of my code on his own. Then I implemented a simple else statement corresponding to the if statement where I checked if user has given certain inputs or not. And then some tuning My code was ready to go. After that I thought why not use an LCD screen too to show corresponding outputs for each condition. For that I did quite a digging through internet because I had no idea how to use LCD. There in that instance I faced some debugging issues and finally finished this programming.

And I specially want to mention that while writing the PDF in L^AT_EX I faced an issue to use the arduino code. After quite digging I found a github file that allows arduino code to write in latex without any external problem and I used that. But I noticed maybe the code is written in certain way that in my print statements in arduino code the space between words filled by a weird shaped

character. I tried to remove that but I don't know how to interact with the mother file. So, I'm keeping it untouched like this.

4 Conclusion

In conclusion, Our project "*Fading-Glowing Of LEDs*" has been successfully made using some simple codes to control the LEDs. We finally achieve all of our goals. After running the program we can see in serial monitor that it is asking for inputs from user. After we press B it represents that the program is running and corresponding message is shown in both serial monitor and in lcd screen. After that let's assume user presses R which represents Red LED, red led starts glowing then fading and vice versa repeatedly until and unless user interrupts it with another input. If that another input is G or Y or X it executes the corresponding task i.e. on pressing G or Y it glows-fades just like previously mentioned. and on pressing X LED glowing-fading stops. Let's assume user gives some another input in that case Serial monitor and lcd screen shows to give only mentioned inputs by program.

5 Acknowledgement

I would like to express my special thanks of gratitude to our Physics Department H.O.D S.S.B Sir who gave me the golden opportunity to do this wonderful project on the topic *Arduino project of Controlling LED's using serial monitor*, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful for that. Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

References

- [1] Github Repo of trihedral (Kyle). It solved the issue of using arduino code directly in \LaTeX . Click Here.
- [2] Official Arduino Site of query of how to clear LCD Screen. Click Here.