Left Sidebar

Add color control to your dimmable QuadBot lamp.

Overview

Reading the button and setting the color

Setting the brightness

Difficulty rating: 3

Fun Rating: 4

Time taken: 25 mins

Supporting activity - QuadBot Color

Color Controlled Dimmer

Upgrade the dimmable QuadBot lamp so that you can control not only the brightness, but the color as well.



We’ll combine using the dial and the button on QuadBot to let use the dial to set either the brightness or the color of the QuadBot. First let’s adapt our previous code to make the color change only when the button is pressed.

**Add button read into our previous code**

We can use the readButton() function to decide whether to change the color of the LEDs. We use a new function here called colorWheel() that converts a number from 0 to 255 into a color on the color wheel.

#include <QuadBot.h> *//Include the QuadBot Library*

long color; *//A variable to store the color value*

int dial;  *//A variable to store the value of the dial*

void setup(){

QuadBot.begin() *//Initialize QuadBot*

}

void loop(){

dial = map(analogRead(A0),0,1023,0,255); //Map dial value to maximum 255

if(QuadBot.readButton()){ //If the button is pressed

color = colorWheel(dial); //Set dial to be the colorwheel position of dial

writeLEDs(color); //Set the color of the LEDs using the color wheel

}

}

Now whenever you press the button, you should be able to change the color by turning the dial!

**Setting the brightness**

Finally let’s make the dial control the LED brightness when the button isn’t pressed. We can just add an else statement to our code, that runs when the button isn’t pressed!

#include <QuadBot.h> *//Include the QuadBot Library*

long color; *//A variable to store the color value*

int dial;  *//A variable to store the value of the dial*

void setup(){

QuadBot.begin() *//Initialize QuadBot*

}

void loop(){

dial = map(analogRead(A0),0,1023,0,255); //Map dial value to maximum 255

if(QuadBot.readButton()){ //If the button is pressed

color = colorWheel(dial); //Set dial to be the colorwheel position of dial

writeLEDs(color); //Set the color of the LEDs using the color wheel

}

else{

setLEDsBrightness(dial); //Set the brightness of the LEDs

}

delay(1) //Add a small loop delay

}

And now you should be able to get the exact color and brightness you want on QuadBot! Great work!

**Conclusion**

Fantastic progress! You’re starting to combine multiple inputs and make decisions for outputs. This is the essense of robotics! Let’s try some of the other outputs on QuadBot!