



# USER'S MANUAL

*LED Strip Controller*

**for Controlling Color of LED Strip**

Prepared by: Daniel Puleio

November, 2016

---

## Revision Sheet

Release No.	Date	Revision Description
Rev. 0	11/28/16	User's Manual Created
Rev. 1	11/20/16	Connected LED Strip to Arduino
Rev. 2	11/25/16	Built android application to control Arduino

---

# USER'S MANUAL

## TABLE OF CONTENTS

	<u>Page #</u>
<b>1.0 GENERAL INFORMATION.....</b>	<b>1-1</b>
1.1 System Overview .....	1-1
<b>2.0 SYSTEM SUMMARY.....</b>	<b>2-1</b>
2.1 System Configuration.....	2-1
<b>3.0 GETTING STARTED.....</b>	<b>3-1</b>
3.1 Software Uploads and Driver Installs .....	3-1
3.1.1 Uploading C file onto Arduino .....	3-1
3.1.2 Installing ES File Explorer on Android Device .....	3-1
3.1.3 Installing Android Application onto Android Device .....	3-1
3.2 Connect Physical Devices.....	3-2
3.2.1 MOSFET Transistors.....	3-2
3.2.2 Power Supply.....	3-3
3.2.3 Android to OTG to Arduino .....	3-3
3.2.4 Ground.....	3-3
3.3 Finding Arduino on Android Application .....	3-4
<b>4.0 USING the Android APPLICATION.....</b>	<b>4-1</b>
4.1 Output.....	4-1
4.2 Find Device.....	4-1
4.3 3 x 4 Grid of Colored Buttons .....	4-2

## **1.0 GENERAL INFORMATION**

## **1.0 GENERAL INFORMATION**

### **1.1 System Overview**

This android application will allow users to control the colors of an LED strip with the click of a button, giving the user 12 different colors to choose from. Prior to running the app, the user must ensure that the LED strip is connected to the Arduino and that the android device being used has the proper USB driver: ES File Explorer.

## **2.0 SYSTEM SUMMARY**

## **2.0 SYSTEM SUMMARY**

### **2.1 System Configuration**

This application uses an Arduino microcontroller in order to interface the android device with our LED strip. To connect the android device with our Arduino, we use an OTG cable and download the proper driver to allow for compatibility. Our Arduino uses jumper cables, three MOSFET transistors as well as load resistors. The MOSFET transistors are placed as switches to amplify current coming out of the Arduino and cut off feedback current coming from our power supply. The android application will give the user the ability to control this system over a graphical user interface: an Android app.

## 3.0 GETTING STARTED



## **3.0 GETTING STARTED**

### **3.1 Software Uploads and Driver Installs**

#### **3.1.1 Uploading C file onto Arduino**

Simply decompress the tar file, and open the file named “arduino\_TermProject.c” on Arduino IDE linked below. To connect the Arduino to a laptop, one can use an Arduino USB cable. Once connected, the user should verify the C file, which is set up on the Arduino IDE, by clicking “verify”. After the verification has successfully completed, the user can go ahead and press, “upload”. Once finished, the user can unplug the Arduino from his laptop.

#### **3.1.2 Installing ES File Explorer on Android Device**

In order for the user’s Android device to communicate with the OTG cable, the proper drivers need to be installed on the user’s Android device. ES File Explorer is the most recommended driver for this device. This driver is available for Android users on Google Play, or any other app store.

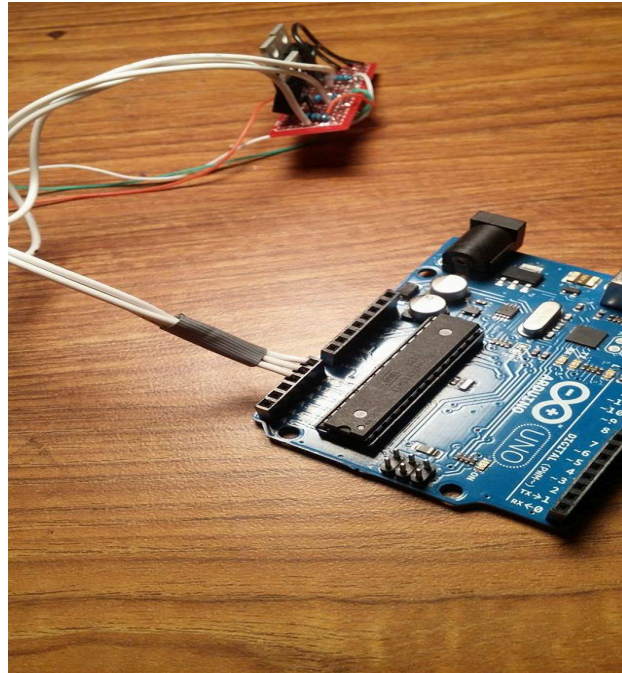
#### **3.1.3 Installing Android Application onto Android Device**

The user should download Android Studio onto his or her laptop if he or she does not have the application installed already. Move the folder named “LED Controller” into your “AndroidStudioProjects” folder and open the project using the IDE. Simply plug in your Android device into your laptop using a USB, and run MainActivity.java on your device. The application will now appear on your android device for future use.

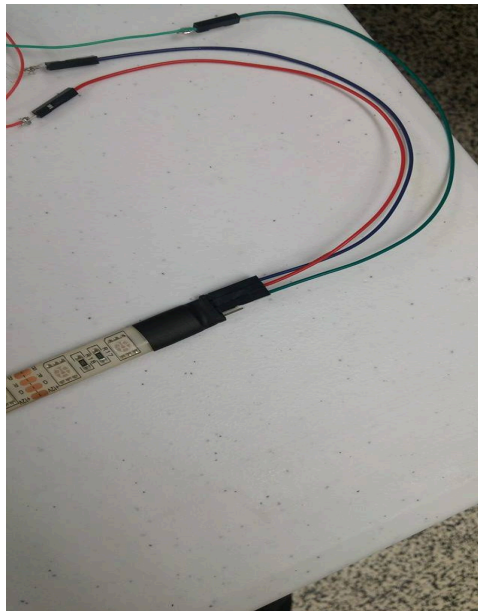
## 3.2 Connect Physical Devices

### 3.2.1 MOSFET Transistors

The user should connect the jumper wires on the input side of the transistors to Arduino pins A0, A1, and A2.



The user should connect the jumper wires coming from the output of our MOSFET transistors into the LED strip, respectively. Follow the color codes.



### **3.2.2 Power Supply**

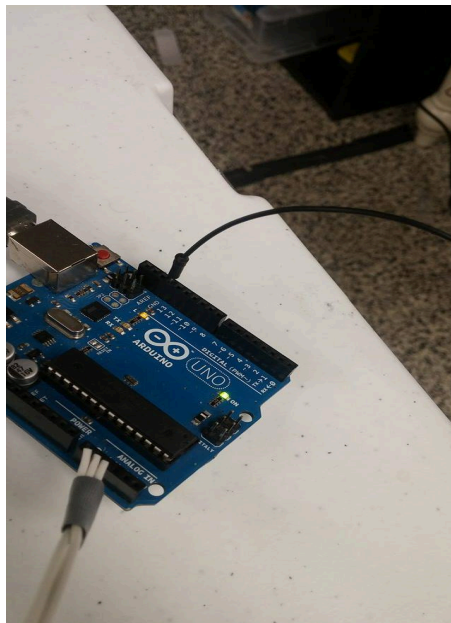
Using a 12 V power supply, the user should connect the power supply into the LED strip via jumper wire.

### **3.2.3 Android to OTG to Arduino**

The user should connect the Android device to the Arduino using an OTG cable, which will then be connected to an Arduino USB cable.

### **3.2.4 Ground**

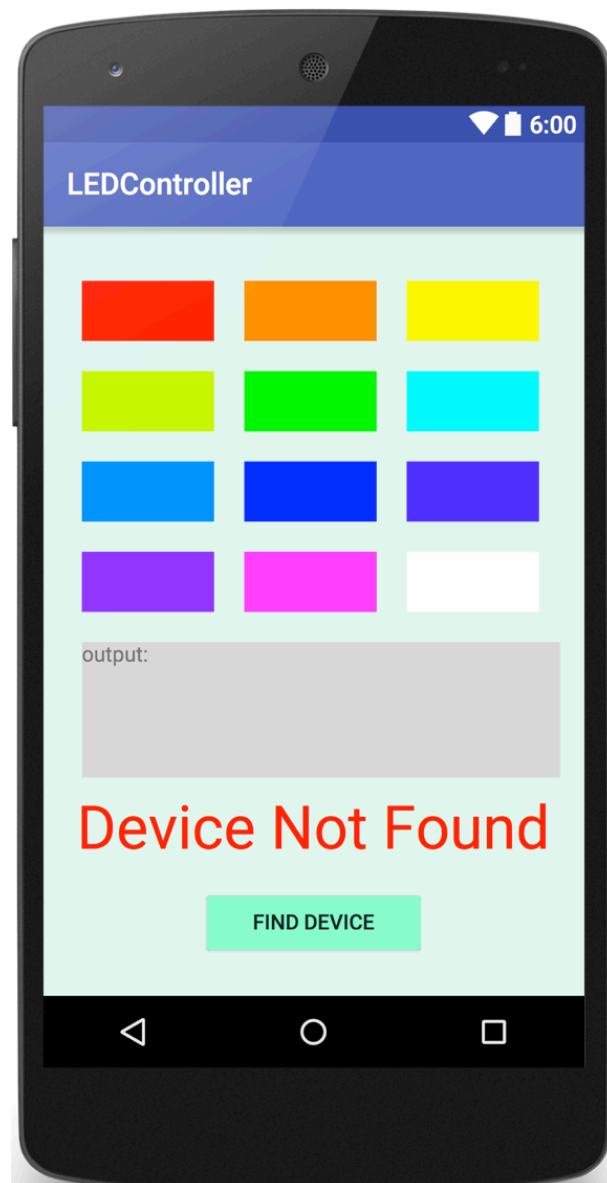
Please remember to ground both the power supply as well as the MOSFET transistors, this is essential to your safety.



### 3.3 Finding Arduino on Android Application

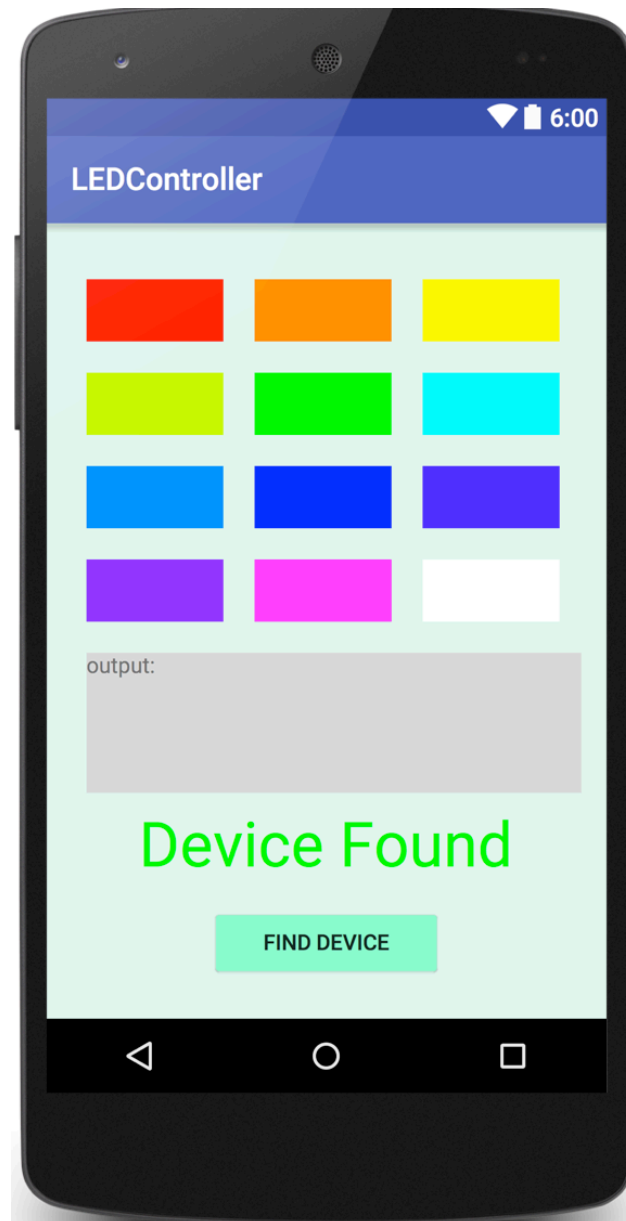
The assuming the user has successfully connecting all physical devices, he or she can is now reading run the android application. Click the short cut icon to run the android application.

The following display will appear.



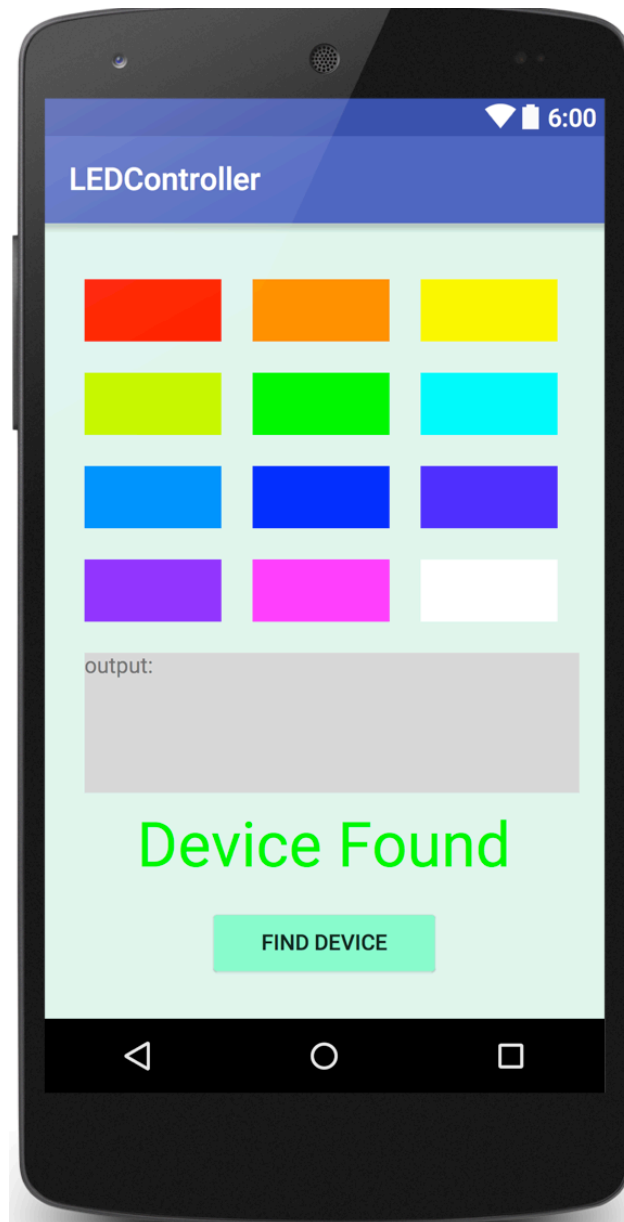
In order for the application to request permissions to communicate with our Arduino device, the user must click the button labeled “Find Device”.

After pressing the “Find Device” button, the Text widget should not read “Device Found” as shown below. This results in enabling the colored buttons, making them clickable to send data over the OTG cable to the Arduino.



## **4.0 USING THE ANDROID APPLICATION**

## 4.0 USING THE ANDROID APPLICATION



### 4.1 Output

The purpose of the output TextView widget is to receive clarification that our device has communicated with our Arduino, by printing to the serial in our C file and seeing if it is received by the android device by displaying the data in the TextView widget.

### 4.2 Find Device

The button labeled “Find Device” should be clicked when our Text widget displays “Device Not Found”. The Find Device will attempt to communicate with the device by requesting permissions to the Arduino device and wait for a response. If the communication is successful, the Text widget will display “Device Found”.

### **4.3 3 x 4 Grid of Colored Buttons**

These buttons will become enabled after the connection between the Android device and the Arduino is made. To make this connection, click “Find Device”. When enabled, clicking these buttons will write data to the serial port, making the Arduino switch the colors of the LED strip according to which colored button is pressed.