Programmer’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

PROGRAMMER'S MANUAL

TABLE OF CONTENTS

Page #

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 Adruino, 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 s 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.

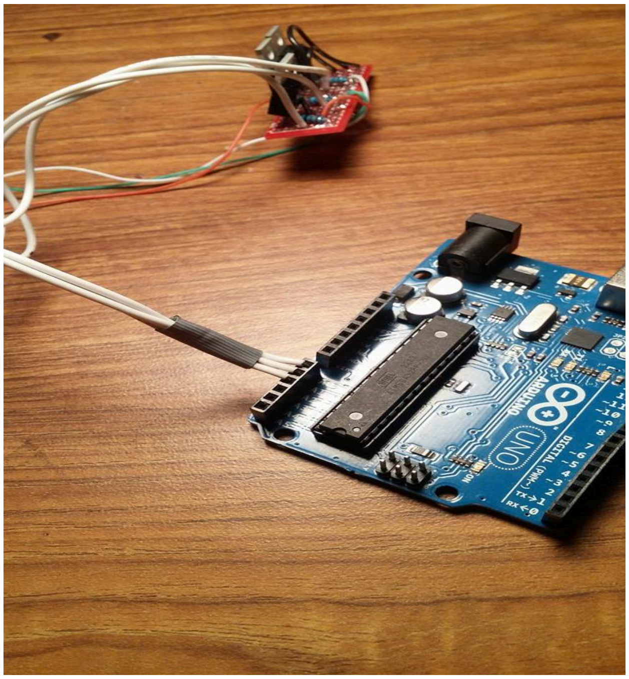
3.1.3 Installing Android Application onto Android Device

The user should download Android Studio onto his or her laptop is 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.

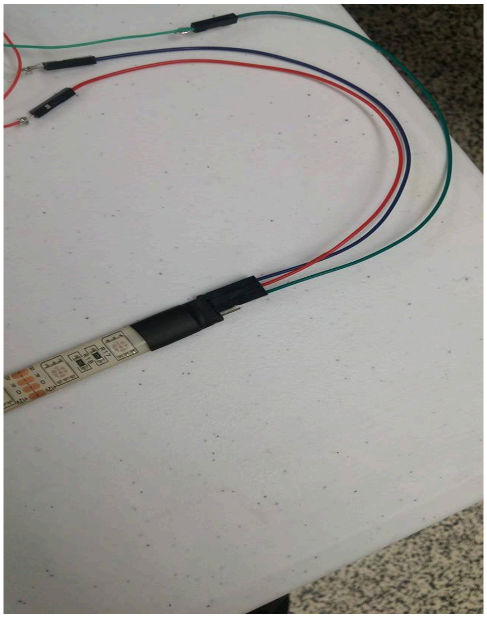
3.2 Connect Physical Devices

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

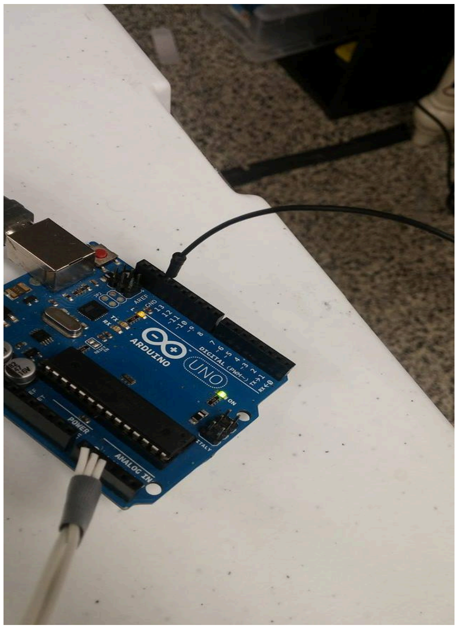
Using a 12 V power supply, the user should connect the power supply into the LED strip via jumper wire.

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

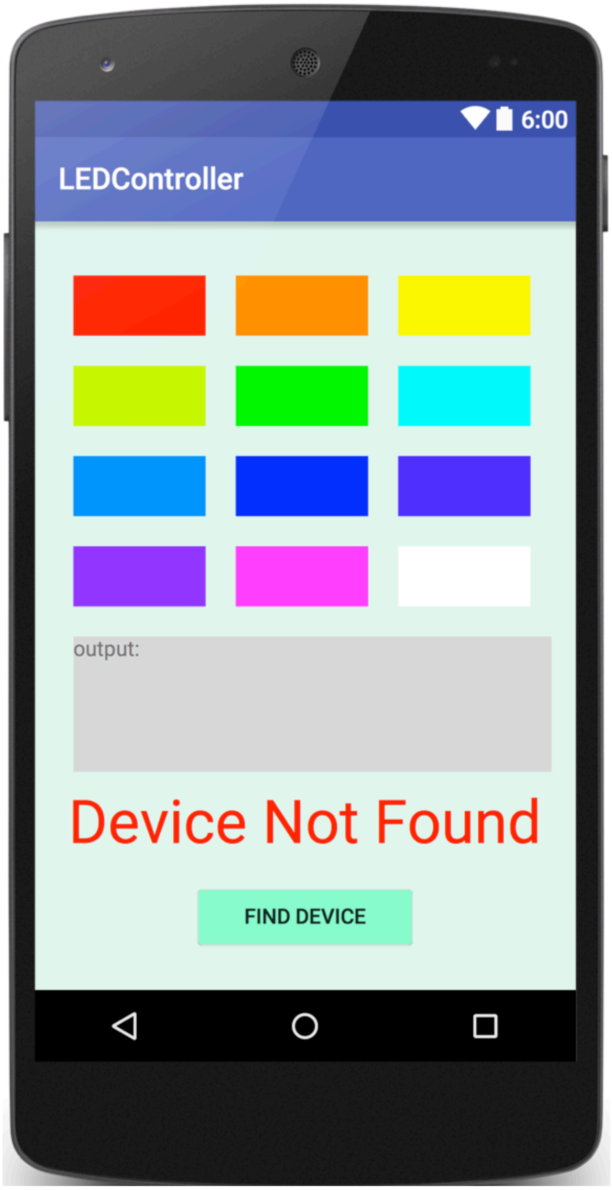
Please remember to ground both the power supply as well as the MOSFET transistors, this is essential to your safety.



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.

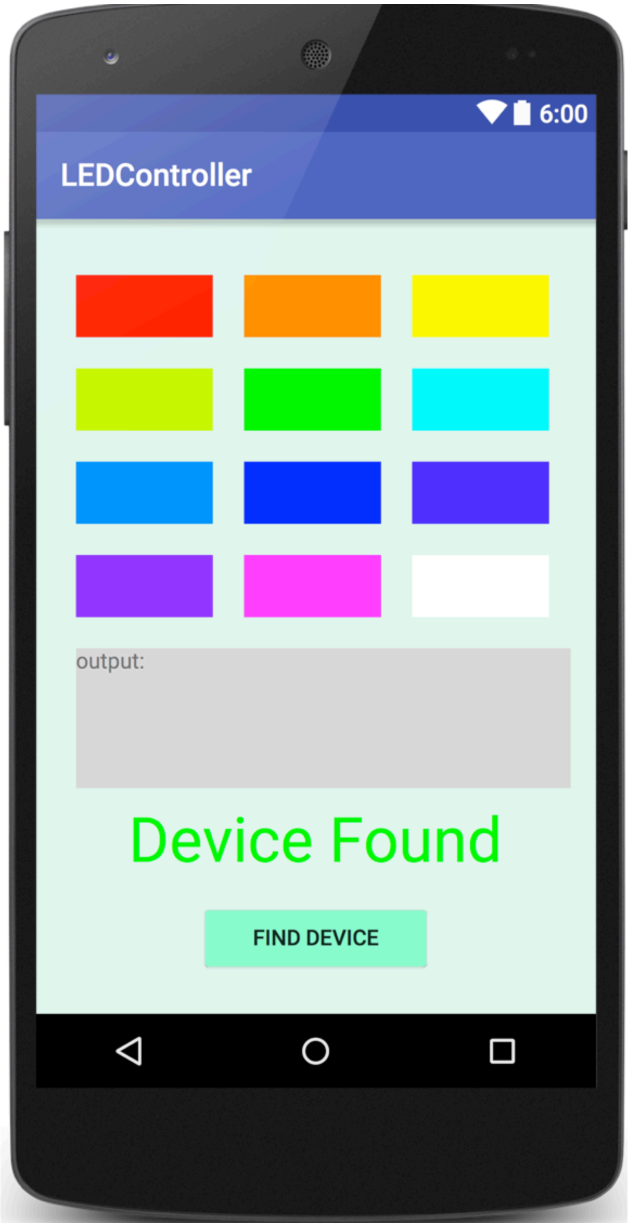
The following display will appear.



Device Not Found

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

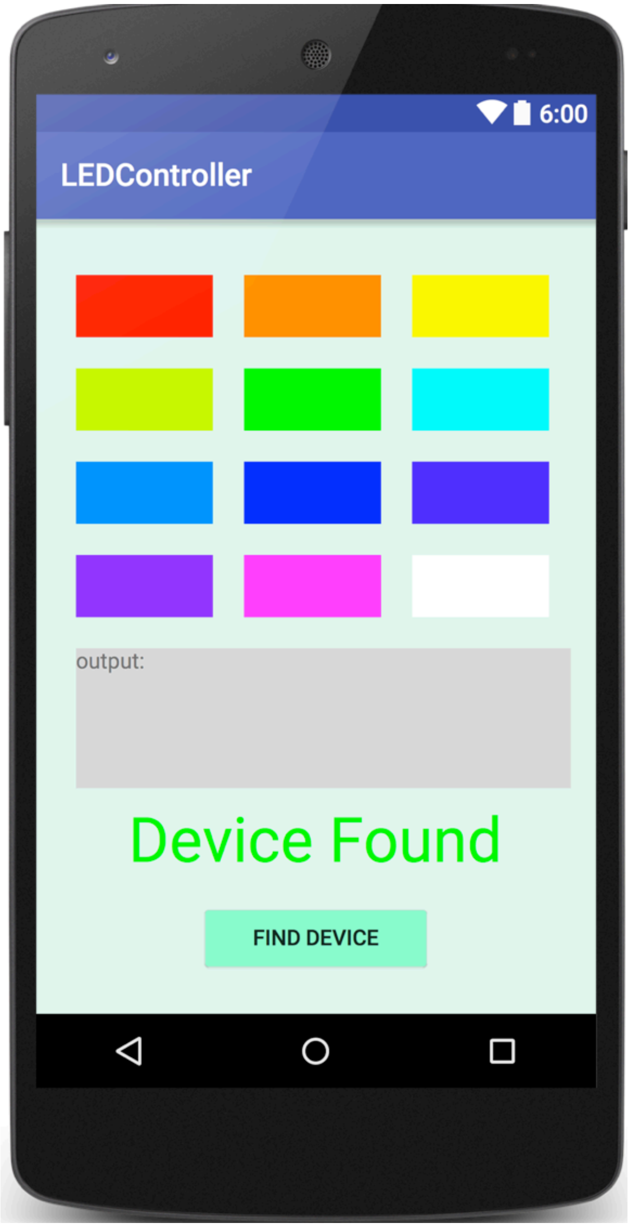
After pressing the “Find Device” button, the Text widget should not read “Device Found” as shown below. This results in enabling the colored buttons, making them clickable to send data over the OTG cable to the Arduino.



Device Found

4.0 Android Development

4.0 ANDROID DEVELOPMENT



4.1 XML Code

The purpose of the main activity XML file is to produce a user-friendly graphical user interface for our users.

4.1.2 Find Device

Using an OnClickListener to send permissions request to communicate with the Arduino device.

4.1.2 Colored Buttons

Using an OnClickListener to send bytes of color code over the USB serial port to the Arduino.

4.2 USB Protocol

Using Android UsbManager, we are able to find all the connected USB devices and open a connection between the Android device and the USB device. We also set our baud rate to 9600 and set our stop bits.

5.0 Arduino Development

5.0 ARDUINO DEVELOPMENT

The Arduino sets up communication to its pins: A0, A1, and A2 within the setup function. In the loop function, we read from our serial port and wait until we receive three bytes of data. Once this is complete, we now break up the data into three separate unsigned char's and send the data to our pins.