
Guide to LED and Hobby Lighting Projects Documentation

Release 0.1.2

Brian Luft

Nov 06, 2017

Contents

1	Set Your Goals and Expectations	3
1.1	Introduction	3
1.2	Learning Foundations	3
1.3	Determine Short-term and Long-term Goals	4
2	Materials and Tools	7
2.1	First steps	7
2.2	Nailed The Basics, Ready For More	7
2.3	Dedicated To Learning	8
2.4	Prototyping Tips	8
2.5	Diagnostics	8
3	Electricity and Digital Circuits	9
3.1	Basics	9
3.2	Components	9
3.3	About ICs	9
3.4	Exercises	10
4	Meet the LEDs	11
4.1	Form Factors (size, shape, type)	11
4.2	Properties	11
4.3	Hackable Consumer Products	12
5	Planning Your Project	13
6	LED circuits	15
6.1	Things You Should Not Do With LEDs	15
7	Indices and tables	17

Contents:

Set Your Goals and Expectations

1.1 Introduction

Your learning curve might depend on what your intellectual needs and goals are. Decide where you stand on the spectrum of using kits to make things for your own satisfaction vs. learning the fundamentals in order to be able to execute your creative whims. Temper that with the fact that mastering a technical field like digital circuits takes years of full-time dedication. Do you like songbooks or do you study music theory? Do you follow recipes or prefer baking from scratch?

I don't think you need to be a technical genius to be successful. An intimidating part of being a novice is the dizzying array of options with various tradeoffs to consider in how you accomplish your project.

- Cost of materials
- Form factor - both of the individual LEDs as well as the finished product as a whole.
- Ease of construction vs. design + programming + wiring + testing labor time.
- Willingness to compromise on any of the above in order to have something working.

1.2 Learning Foundations

The knowledge you'll need can vary greatly depending on your goals. If you plan to take any bit of DIY approach, you'll want to start with some basics.

Basic algebra is adequate for beginner digital circuit applications. You'll definitely want to spend some time going through intro to electronics and/or Arduino tutorials. You'll need some basic familiarity with a handful of components (resistors, capacitors, transistors, voltage regulators) and an understanding of how the properties of Ohm's law relate to each. I'd recommend this even if your goal is just to build schematics you find online.

If you're looking to program lights in fancy patterns you will run into some need to dip into various maths as well as some non-trivial programming. The more sophisticated your creative needs are, the more you'll need to be able to translate general mathematical concepts into specialized programming (changing light color and brightness according to the value of math equations over time).

Plan for a little time practicing with basic breadboard wiring. The simple act of wiring up your first circuits will help you realize how little you actually know.

The thing **about getting beyond a few lights** is:

- You **run out of pins** on the microcontroller. You can still use the single microcontroller but you'll need to learn some more principles.
- Your circuit and chips have to start dealing with (relatively) **a lot of current**. Combining a desire to keep minimal (and cheap) hardware, many lights, and nice aesthetics means spending some time learning some engineering.
- You'll want to consider **using some supporting chips** and components to help drive the LEDs. This means more learning time, design time, and build time. For some applications the extra hardware will be required.
- **Cost and time** become real factors. For a beginner who may just want to order roughly the parts necessary for some small projects, the cost per light can run from a couple to several dollars when you factor in the supporting components. A project with 25 RGB lights can easily run \$50-\$100 and at least several hours in design and build time for a novice. I'm just talking about the small 5mm lights here. If you want higher power lights - enough to make an impact in large rooms or small outdoor spaces - costs can run several times higher if you're planning to source and build everything yourself.

1.3 Determine Short-term and Long-term Goals

What follows is an attempt at a basic cross-section of options and considerations. Poke through these links and see what you identify with relative to your own goals and limitations.

Inspiration

- Look through some Instructables for inspiration and learning. Some are better than others. [LED Projects on Instructables](#)

Reality Check

This is a pretty advanced project and a source of inspiration for me. Reading through the specs a hundred times and researching what I didn't know helped me learn a lot. Note that this project uses a different microcontroller - the PIC. The operating principles show what is possible with minimal hardware and very tight design.

- [Aurora](#)
- [Aurora Mini](#)

This is an example of an installation done by people who really know their stuff and with many hours of professional labor

- [100 Lanterns and the blog post](#)

Temper your expectations and dreams accordingly :)

Find Your Path

A rough and opinionated breakdown of some options suitable for the novice:

Budget: Low

Learning: High

Labor: High

Creative Control: High

I want to be able to create unique projects with minimal hardware

Cost is very important

I have enough free time to dedicate to learning the engineering

Aesthetic is very important

- If you want to go the DIY route, you'll need to have an understanding of these PWM and multiplexing [Arduino wiki - Output techniques](#) (see [LED sections](#)) If you can understand how controlling a matrix works using these techniques, you should be able to achieve other form factors by grouping lights together in similar ways.
 - [ShiftPWM](#) is a nice option for getting started. Shift registers are cheap and the wiring is as easy as the registers, resistors and LEDs. The library is easy to get started with.
 - There are many LED driver ICs (chips). TLC5940 is a fairly popular one with a [nice library](#). The chips are a little more expensive than shift registers and offer some nice capabilities. This solution offers more power and flexibility than ShiftPWM with the tradeoff of more cost and more learning time to harness and understand the capabilities of the chip.
 - [FastSPI](#) gives you the flexibility to write your lighting routines and then change the supporting driver chips.
-

Budget: Medium-High

Learning: Medium

Labor: Medium

Creative: Medium

I'm sticking with Arduino and shields

I'm sticking with Arduino and a handful of cheap chips and components

There are Arduino-compatible boards that are designed with lighting projects in mind:

- [Rainbowduino](#)

Arduino LED shields

- [Color shield](#)
 - [Power LED shield](#) is a more advanced and expensive shield for controlling higher power LEDs.
-

Budget: High

Learning: Low-Medium

Labor: Low-Medium

Creative: Low-Medium

For bigger (in terms of number of lights and/or size of space to light) projects

For quick construction

Flexible form factor

- [ShiftBrite Shield](#) and [ShiftBrite lib](#)
- As others mentioned, LED strips are a nice option although they might not be the form factor you want. Some of the strips can be separated into multiple segments (one or two lights for example). The nice thing is the if you can accept the form factor limitations, the wiring construction is much simpler for being able to control up to dozens of nice lights.

- Something like [BlinkM](#) is the equivalent of an individual strip pixel but you have the freedom to build it into your own form factor.
- [DIY RGB LED pixel](#). You could start by building a few of these at a time as you learn.
- Various driver boards and kits that have their own user-friendly programming packages. These will cut way down on what you need to learn for circuit construction and programming - at the expense of creative and form factor control. A lot of these are designed for industrial / larger scale use, but that's probably closer to what you're ultimately after in public / larger spaces.

2.1 First steps

- Arduino
- A few LEDs
- A few resistors 220ohm
- Breadboard

2.2 Nailed The Basics, Ready For More

- voltage regulator
- trim pots
- switch / buttons
- More LEDs
- RGB LEDs
- multimeter
- pliers
- cutters
- wire stripper
- wall wart

2.3 Dedicated To Learning

- transistors
- 555
- Shift registers
- driver ICs

2.4 Prototyping Tips

Get some header pins so you can plug directly into breadboard Arduino's funky pin dimensions - adjust the pins

2.5 Diagnostics

Ways to test LED works

- Plug into pin 13
- Breadboard with resistor and 9v
- Testing common cathode and common anode RGB LEDs

3.1 Basics

- voltage, current, power
- Ohm's law
- voltage divider
- 2 transistor constant current
- source vs. sink (anode, cathode)

3.2 Components

- resistor
- LED
- transistor
- capacitor
- shift register
- ICs

Something that might trip up novices is that some components change behavior with current; others with voltage.

3.3 About ICs

ICs often come in different package types. You'll normally want to start with DIP packaging. Pay attention to whether the IC is a source or sink for current.

Don't work with mains power.

3.4 Exercises

alternate flashing lights fader chaser / sequence

4.1 Form Factors (size, shape, type)

Bulb

3mm 5mm

RGB

RGB is really 3 lights in one.

Common anode vs. common cathode: <http://flashgamer.com/arduino/comments/rgb-led-common-cathode-or-common-anode>

If you are mixing colors, diffused LEDs might be better otherwise you can see the individual diodes lit.

Pixel

BlinkM

Ghetto Pixel http://www.instructables.com/id/Ghetto-Pixels-Building-an-open-source-BlinkM/#commentactioncontainer_CF0Q2CTGP7IS5IH

Strip

Array / Matrix

7 segment

High Power

SMT

4.2 Properties

Operating Properties

- Forward Voltage

- Forward Current

Light properties

- Luminosity / Luminous Intensity / Brightness
- Wavelength
- Viewing Angle

4.3 Hackable Consumer Products

- GE light strings
- IKEA Dioder

Product Page: <http://www.ikea.com/us/en/catalog/products/20119418/> Tutorials:

- <http://marco-difeo.de/2012/03/28/irdioder-ikea-dioder-hack-mit-atmel-und-infrarotempfanger/>

Planning Your Project

http://www.reddit.com/r/electronics/comments/r42iu/need_help_with_choosing_driver_for_leds/ considerations: how many run simultaneously, adjust brightness, etc.

figure out limiting factors cost vs. design + build time vs. form factor requirements

components:

- resistor
- capacitor
- **transistor** mosfets are voltage components bjt are current components

http://www.pcbheaven.com/userpages/LED_driving_and_controlling_methods/?topic=worklog&p=3

IC

photoresistor

Make sure you understand

- Forward Voltage and Current

A nice intro tutorial to understanding the calculations and principles of basic LED wiring arrangements. <http://www.ladyada.net/learn/arduino/LEDs.html>

More advanced - learn about use of transistors and constant current circuits. http://www.pcbheaven.com/userpages/LED_driving_and_controlling_methods/?topic=presentation

6.1 Things You Should Not Do With LEDs

- Connect directly to battery
- Connect to a power source without a resistor (or a constant current driver)
- Connect to an Arduino pin without a resistor
- Connect in parallel without resistors
- Think that you don't have to limit current because you're using PWM

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`