CS Capstone Technology Review

November 16, 2017

Winter is Coming...

Prepared for

Oregon State University

Victor Hsu

Prepared by

Group 64

PlanteR-GB

Travis Hodgin

Abstract

This will be filled in during the final draft.

CONTENTS

0.1 References

[1] *K20 Sub-Family*, Freescale Semiconductor, 12 2012, available at https://www.pjrc.com/teensy/K20P64M50SF0.pdf, Rev. 4.5.

[2] *ATME 8-but microcontroller with 4/8/16/32kBytes*, Atmel, 11 2015, available at http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf.

[3] "Arduino uno," https://store.arduino.cc/usa/arduino-uno-rev3, 2017, rev. 3.

[4] "Introducing the raspberry pi zero," https://cdn-learn.adafruit.com/downloads/pdf/introducing-the-raspberry-pi-zero.pdf, 2017.

[5] "Adafruit neopixel digital rgb led strip white 60 led," https://www.adafruit.com/product/1138, 2017.

[6] "Adafruit dotstar digital led strip black 30 led - per meter," https://www.adafruit.com/product/2237, 2017.

[7] "12mm diffused thin digital rgb led pixels (strand of 25)," https://www.adafruit.com/product/322, 2017.

[8] D. Garcia, "Fast 'led," https://github.com/FastLED/FastLED/wiki/Overview, 2016.

[9] "Arduino library use," https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use, 2017.

1 MICROCONTROLLER

1.1 Overview and Criteria

The microcontroller will run parts or all of our software. With this in mind, we need to find a controller that fits within certain criteria. The controller must

- Controller is reprogramable
- Controller contains I/O pins

The Controller must be reprogramable as to allow the user to change files; This also facilitates testing. The I/O pins will allow the LED strips to interface with the controller, as well as allow additional features to be added.

Following these criteria, we have a few additional features that would be benifical. These include:

- Controller allows multiple applications to run simultaneously
- Controller allows connection via a wireless network
- Controller is capable of hosting a web page

Allowing the controller to run multiple applications will facilitate multiple features to run on a single controller; For example the controller will run the LED interface, as well as hosting a web page. Allowing connection via a wireless network will help with additional features on a single controller.

1.2 Potential Choices

During our research we have found three commonly used microcontrollers that fit our criteria. The first is the Teensy 3.2. The second is the Arduino Uno, and the third is the Raspberry Pi Zero W. These three were chosen for their ease of use, price, and documentation.

1.2.1 Teensy 3.2

The Teensy 3.2 is a commonly used microntroller for small scale LED projects. it uses an ARM processor running at 72MHz [1, Pg 2]. Storage consists of 256KB of flash memory that will contain the Arduino or C program you flash to it. The Teensy has 34 I/O pins, and is powered by 5V supplied by a micro USB cable.

The Teensy 3.2 is a fairly small board, being only 1.4 inches long by 0.7 inches wide, which is great for projects where space is limited. The Teensy 3.2 is priced at $20.

Advantages of the Teensy 3.2:

- Small form factor
- Runs Arduino and C programs
- Compatible with many LED libraries

Disadvantages of the Teensy 3.2:

- Not robust enough for much more than the LEDs
- Documentation is lacking with regards to LEDs

1.3 Arduino Uno

The Arduino Uno is a popluar microcontroller for many small electroncs projects. The Uno contains an ATmega328 controller running at 20 million instructions per second (MIPS) at 20MHz with 32KB of flash memory[2, Pg 2]. Containing 14 digital I/O pins of which 6 can be used for PWM poutputs, and 6 analog inputs [3, Pg 2]. With an input voltage of 5v off a barel jack. The Uno has a large community with a lot of documentation.

The Arduino Uno is 2.7 inches by 2.1 inches making it more ideal for projects where size isnt a limiting factor.

Advantages of the Arduino Uno:

- Compatible with many LED libraries
- Addon shields allow for further features to be added
- Interfacing with the LEDs is easiest with this device
- Documentation is fantastic

Disadvantages of the Arduino Uno:

- Shields sold seperatly
- Not robust enough to run more than LEDs and sensors
- Wireless connections require more hardware.

1.4 Raspberry pi Zero W

The Raspberry Pi Zero W is more of a microcomputer than a microcontroller. It is capable of running a full linux operating systems. The Pi Zero W contains an ARM processory at 1GHz with 512MB of RAM Making this the most powerful option we are looking at. This device uses 40 GPIO pins for external interface, uses a micro SD card for storage, and supports digital video out via a mini hdmi[4, Pg 2].

Additionally, the Pi Zero W contains a wireless and bluetooth chip built in. From a size standpoint the Raspberry Pi Zero W is another small controller being 2.5 inch by 1.2 inch. This makes this controller another great choice for smaller projects. Raspberry Pi Zero W is priced at $10, and has a huge community with plenty of documentation.

While the Raspberry Pi can run many RGB LEDs, specific models use internal clock timings time the LED signals. This makes running these on the Pi, which uses a multi-tasking linux operating system, a pain.

Advantages of the Raspberry Pi Zero W:

- Can host local web pages
- Can run a lite linux operating system
- Great documentation for

Disadvantages of the Raspberry Pi Zero W:

- Multitasking operating system makes clock based LEDs a pain to work with.

1.5 Discussion

Keeping our criteria in mind, each of the controllers listed here meet our requirements. The Teensy, being our least powerful device, is a great choice for small LED projects. It appears the Teensy will not be able to handle both setting the LEDs and changing the time the LEDs will turn on and off at the same time. This makes the Teensy a great option for only a part of the project.

The Arduino Uno, being our second most powerful device on the list, is capable of handling both setting the LEDs and changing the time the LEDs will turn on and off. This makes it a great option for this project. Since it is an Arduino, we are limited to two coding languages; Arduino, and C. The only way for the Uno to connect to a wireless network, is by using an external shield or USB wireless adapter. This means we will need additional hardware to connect to any network.

The Raspberry Pi Zero W is the most powerful controller option on this list.

1.6 Conclusion

Given our criteria, and our additional features, we have decided to use the Raspberry Pi Zero W and an Arduino Uno. Because of the power of the Pi Zero W, the ability to run a full linux operating system, price, size, and number of GPIO pins, this controller fits everything we need and more. The Arduino Uno LED libraries make using the Uno as the LED driver much more attractive. I will discuss LED Libraries in another section. The Pi Zero W's ability to host web pages, and connect to wireless networks with no additional hardware, we can achieve much more of our additional features without the need for additional hardware.

2 LEDs

2.1 Overview and Criteria

The LEDs are the centeral point of this project. As such, we need to pick a type that suits our needs. For this we need the LEDs to be a certain type. Our criteria are:

- The LEDs must be RGB
- The LEDs strips must be modular

The LEDs we choose have to be RGB, that is the point of this project. With modular LEDs, we will be able to create any size kit we want, depending on how many LEDs we want to use.

2.2 Potential Choices

We have three great choices for LEDs. The first is the Adafruit WS2812 Neopixel LED strip. The second is the DOTSTAR APA102 LED strip. Lastly we have the WS2801 diffused LED strand. Each of these fits all of our criteria.

2.3 Neopixel WS2812

The WS2812 Neopixel LEDs from Adafruit is popular for many LED projects. These LEDs are digitally addressable, meaning we can set each LED color individualy; Each LED has a shift-registers, chained throughout the strip which allows us to shorten the strip, or add more to the end [5, Pg 2]. Once you set the color, you can disconnect the strip from the controller, and as long as its still being powered, it will remain thanks to the build in PWM into each LED-chip. Powering the LEDs comes from solderpads on the side of each LED that will provide 5V and up to 2A, ground, and a data line.

Neopixel LEDs come in several size price combonations, the most popular being 60 LEDs for $24.99, across a full meter.

Advantages of the Neopixel WS2812:

- Documentation is great
- Digitally addressable
- Modular
- Many libraries

Disadvantages of the Neopixel WS2812:

- Clock controlled

2.4 DOTSTAR APA102

The DOTSTR APA102 are an alternative to adafruits Neopixel LEDs. Instead of working on a single data pin, these work on a 2-wire SPI, meaning data can traverse the strip much faster than the Neopixel PWM system. These do not require timing meaning clock cycles will not affect these [6, Pg2]. These contain 30 LEDs per meter with 24-bit color, 8 bits for each red, green, and blue. Like the Neopixel, each LED acts Like a shift register, which means we can, if not hardware limited, control an infinite number of LEDs. The full meter costs $19.95.

Advantages of the Dotstar APA102:

- Not clock based.

Disadvantages of the Dotstar APA102:

- less LED per meter than Neopixel
- SPI can be hard to set up
- Documentation is lacking

2.5 WS2801 Diffused LED Strand

The WS2801 LED strand runs similarly to the Neopixel LEDS as they are clock based. Instead of a strip, they are in a strand of weatherproof ”dots” that fit through a 12mm hole[7, Pg 2]. These are diffused LEDs meaning the light is spread more evenly, reducing hard edges and shadows. Like the other two options, these are 24-bit colors. Similarly to the Neopixel’s thes are PWM driven and must be clocked by the controller.

These only come in strands of 25, making these not very modular without modifications. They are run off of 5V at up to 2A, just like the previous choices. They are priced at $39.95 for 25 LEDs.

Advantages of the WS2801 Diffused LED Strands:

- Diffused lighting helps reduce hard shadows
- Dot based LED gives more options for hiding wiring
- Many libraries

Disadvantages of the WS2801 Diffused LED Strands:

- Not modular
- clock controlled
- Forced to use 25 pixels

2.6 Discussion

The WS2812 and the WS2801 use the same clock based control, making the only difference in the style of LED. Both the WS2812 and the DOTSTAR LEDs are strips of flat LEDs, with the Dotstar being about $5 cheaper.

The DOTSTAR uses a non clock based SPI connection from the controller, making it an interesting choice. The major complaint is the documentation. Very few people have posted projects using the DOTSTAR, where the WS281

2.7 Conclusion

Because of the documentation, the libraries supported, and size, we have chosen to go with the WS812 Neopixel LEDs. These are the LEDs we have seen for many different projects, has great documentation from both adafruit, and users, is supported by many libraries, and can be split up into smaller strips if needed makes this our best choice.

3 LED Libraries

3.1 Overview and Criteria

LED libraries are what the microcontroller uses to interface with the LEDs. We need to pick a library that fits with our microcontroller, and LED choice. It is important to note that there exists libraries to run clock based LEDs on the raspberry pi using hardware PWM. These, however are created by other users, and are not updated often. Because of this, our criteria are:

- Library must work with either the Raspberry Pi, or Arduino Uno
- Library must work with the WS2812 Neopixel LEDs

We need to be sure the library we want to use is compatible with the controller we are using, otherwise the whole project will not work. Similarly, we need to make sure the library can work for the LEDs of our choice.

3.2 Potential Choices

For libraries, we have several choices. The first is fastLED. The second is Adafruits Neopixel library, and the third choice is to build our own library. Each of these options will fit our criteria.

3.3 FastLED

FastLED is an arduino based library created to make programming LEDs faster and easier. This library has support for both SPI, and 3-wire chipsets[8, Pg 2]. Many projects found using the Neopixel LEDs used this library.

This library is written in C++ and can be used in for anything that can run arduino, such as the Arduino Uno and Teensy 3.2. Supporting both SPI and 3-wire chipsets allow for many different types of LEDs to interface, however it cannont be used on multi-tasking devices such as the raspberry pi. This is mostly due to the LEDs supported have a problem with clock timing, with the exception of LEDs such as the APA102.

Advantages to FastLED:

- Supports many different LED structures
- Great documentation

Disadvantages to FastLED:

- limited to running on Arduino

3.4 Adafruit Library

The Adafruit Neopixel Library is specifically designed for their Neopixel LEDs. This library is written in C++ and is made for use for Arduino projects. That means this can be run on anything that can run Arduino such as the Arduino Uno and the Teensy 3.2. Adafruit has an article discussing the use of the library[9, Pg 2].

past this the library has little in the way of documentation. The article discusses basic use, but doesn't go into much past their test script. This will lead to more time spent digging into the code base. In our research we have found many different projects using this library.

Advantages of the Adafruit Neopixel Library:

- Built specifically for Neopixel
- Built for use on Arduino devices

Disadvantages of the Adafruit Library:

- Can only use Neopixel LEDs
- Documentation is a little lacking

3.5 Build our Own

Building our own library will reduce a lot of our problems of other libraries. We can code specifically for the LED and controller we want. This reduces the amount of added code the controller will use. Additionally, we will have better knowledge of how to write the project, as we are the ones who wrote it.

The major problem that comes with building our own LED library is time. Havig to write our own library will require extensive resarch, writing, and testing time that can detract from our original project.

3.6 Discussion

FastLED is a very popular LED library used in many projects we found during our research. With its broad LED support, and great documentation, it is an appealing option. The Neopixel library, while only made for use on the Neopixel LEDs, isn't bloated by the addition of these other LED types. Building our own library would allow us to intergrate only features we need for the specific LED type we are using. This however, will require more time for us to complete.

3.7 Conclusion

Because of the documentation and LED support, we have decided to use FastLED as our LED library. FastLED has great support, and allows many LED types for future expansion if needed.