Spring 2015

Project RGB LED Grid: An Interactive LED Wall

Logan Adams
University of Portland

University of Portland

Shiley School of Engineering
5000 N. Willamette Blvd.
Portland, OR 97203-5798

Phone 503 943 7292
Fax 503 943 7316

Senior Honors Project

Project RGB LED Grid: An Interactive LED Wall

Logan Adams Spring 2015

Team Members:

Logan Adams

Jordan Garcia

Dan Nelson

Alex Varvel

Faculty Advisor:

Dr. Aziz Inan

Industry Advisor:

Bret Aguirre (Bonneville Power Administration)

Client:

Dean Sharon Jones, Shiley School of Engineering

Revision History

Rev.	Date	Author	Reason for Changes
0.1	04 Sep 2014	Logan Adams	Submission of initial SHP to Dr. Orr for approval.
0.2	27 Sep 2014	Logan Adams	Formatting of document. Adding information from Functional Spec.
0.3	28 Oct 2014	Logan Adams	Updating overall project information and table of contents.
0.4	11 Nov 2014	Logan Adams	Adding information from Design Document.
0.5	06 Mar 2015	Logan Adams	Updating Introduction, High Level Architecture and Figures.
0.6	15 Mar 2015	Logan Adams	Editing to make figures more clear for lay audience.
0.7	19 Mar 2015	Logan Adams	Adding additional information on FPGA and microcontroller.
0.8	26 Mar 2015	Logan Adams	Incorporated edits from faculty advisor.
0.9	03 Apr 2015	Logan Adams	Continued editing and revisions for spelling and grammar.
1.0	10 Apr 2015	Logan Adams	Founder's Day presentation for SHP, paper submitted to Dr. Orr.

Table of Contents

List of Figures

Introduction

Team LOL (Lots of LEDs) of the University of Portland, consisting of Logan Adams, Jordan Garcia, Dan Nelson, and Alex Varvel embarked on a challenging engineering design project. The team chose to build an RGB LED Grid, explained below in the body of the report. This report serves as the Senior Honors Paper accompanying the Founders' Day presentation.

Background

The Red-Green-Blue (RGB) Light Emitting Diode (LED) Grid is a large display, reminiscent of a Jumbotron that might exist at a stadium or the electronic signs that exist at the entrances to the University of Portland campus. As is true in these types of displays, the RGB LED Grid is based on the principles of low Pixels per Inch (PPI). In many small displays, such as those found on phones and computers, the users are so close to the display that the individual display elements, called pixels, need to be so close together that they appear continuous across the entire screen. This process of packing the pixels very close together is quite expensive, and is not financially reasonable with larger displays such as those mentioned above.

Although the human eye cannot perceive any increase beyond approximately 300 PPI as is common in phones, the eye can also stitch together an image at very low PPI, as long as the display is placed a greater distance away from the user. This is what allows large displays to function. As long as the users are far away, the human eye can stitch together the empty space between the pixels to create an image.

The pixels that make up the RGB LED Grid are LEDs that are capable of displaying any color in the Red-Green-Blue colorspace, the same colors that can be shown on computers, phones, and televisions. Additionally, the Grid is made up of 1200 individual LEDs arranged in a configuration of 40 columns by 30 rows. The LEDs were mounted on Printed Circuit Boards (PCBs) that were designed specifically for this project. A Field Programmable Gate Array (FPGA) and a microcontroller were used to generate and display the patterns. Overall, the entire project follows a design pattern known as "process-connect-display." This format consists of: *processing* the image to be displayed, *connecting* the data to the LEDs over the PCBs, and *displaying* the desired image on the grid.

The major design challenges posed by this project included designing PCBs, controlling and powering the LEDs, generating patterns that can be sent to the LEDs, and designing a stable structure for the entire apparatus. However, these difficulties were overcome through careful planning and design.

The report that follows in the remaining sections provides an overview of the major systems that make up the RGB LED Grid. Minimal electrical engineering or computer science skills are assumed of the document reader. Additionally, an image of the final Grid is shown in Figure 1.

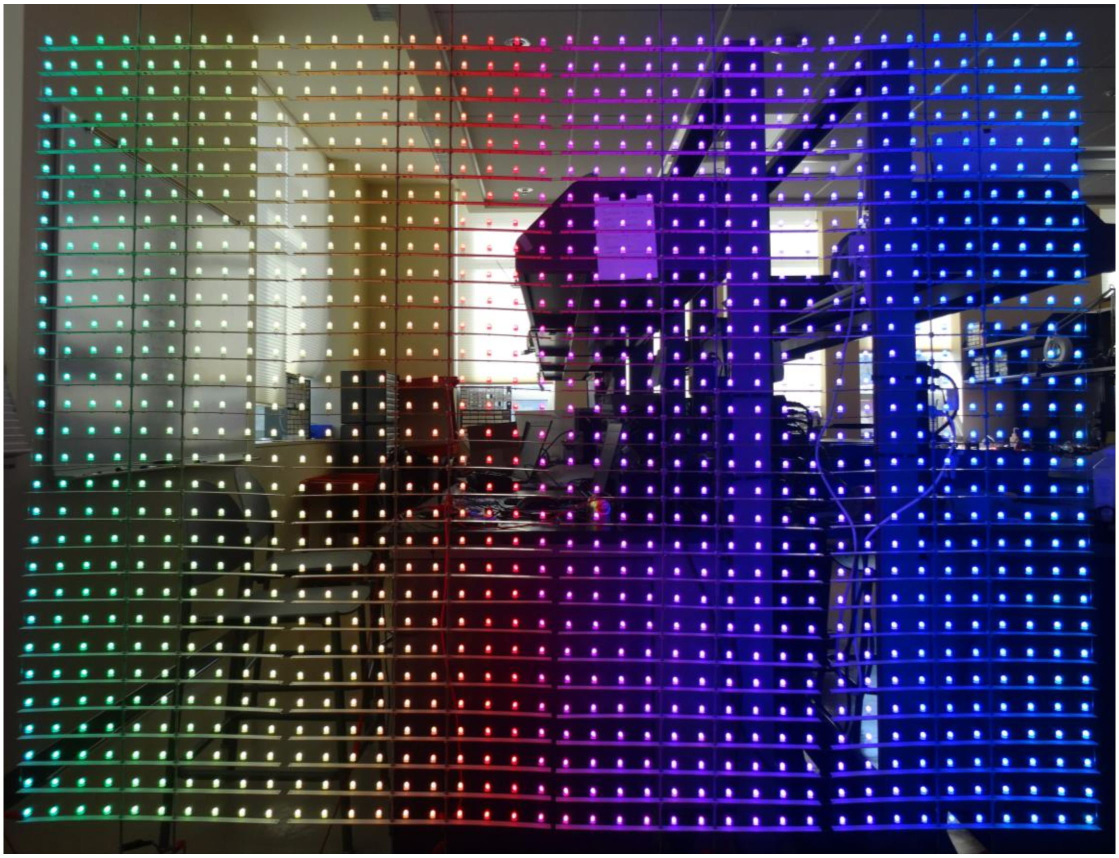

Figure 1. RGB LED Grid.

Design Process

In order to successfully complete the project of realizing a complete RGB LED Grid, a "process-connect-display" was employed to compartmentalize the design. This allowed for various parts of the project to be developed in parallel to maximize time allocated to design and construct the Grid.

Overview

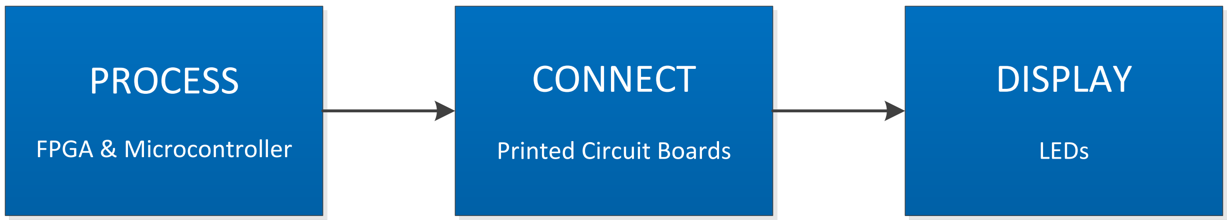

Figure 2. RGB LED Grid system diagram.

The system functionally contains three major conceptual components: process-connect-display. A block diagram of the top level system is shown in Figure 2. Each major component is comprised of smaller components, which are described below. The system in greater detail is depicted in Figure 3.

The block diagram that is shown in Figure 3 shows the blocks in greater detail, and shows the subcomponents and their interactions that help produce the RGB LED Grid as a whole. The individual details will be explained in detail below.

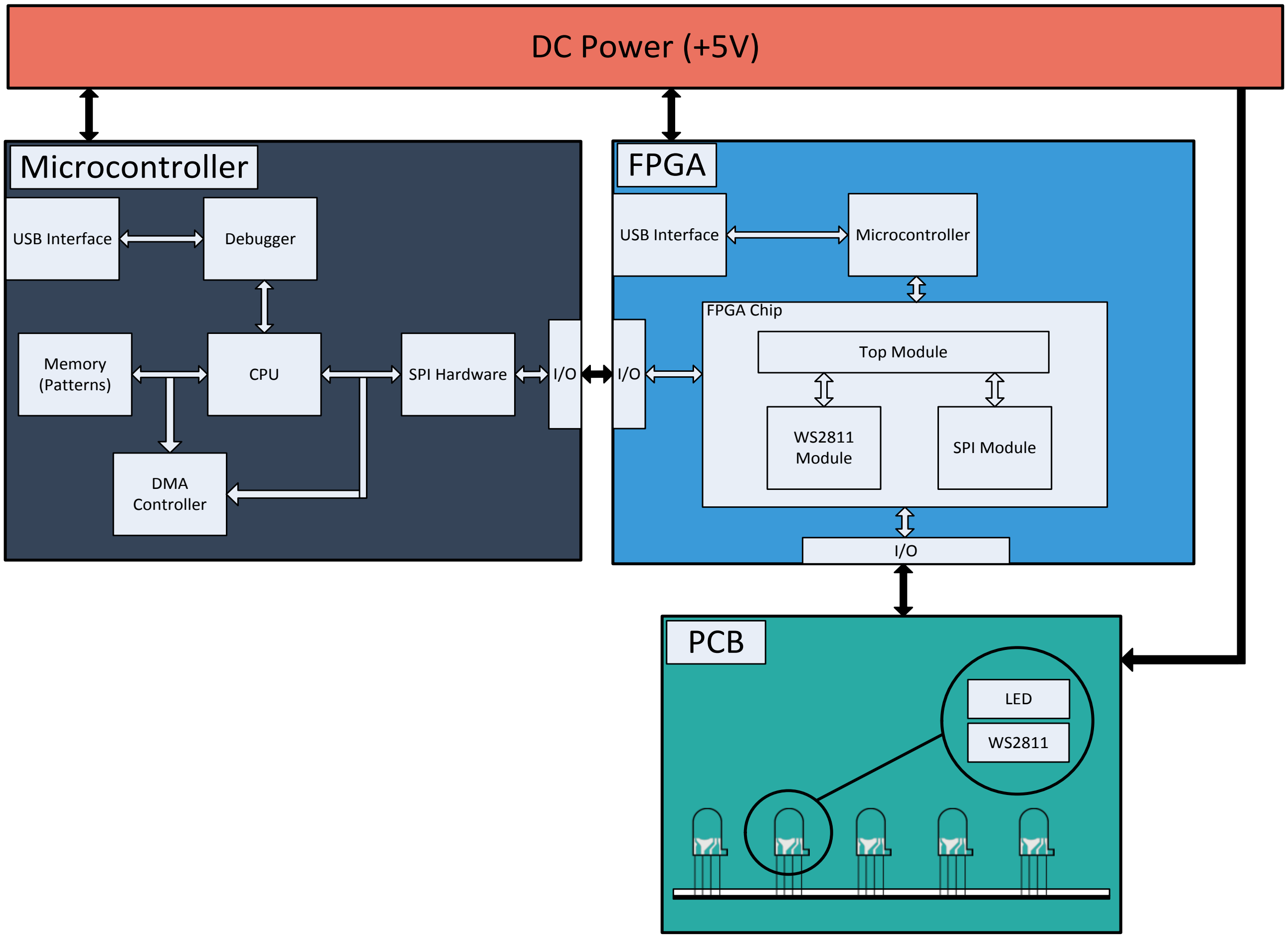

Figure 3. System Block Diagram

LEDs

The RGB LED Grid's main goal is to display patterns that are visually appealing. In order to do this, the most important components are the LEDs, as they are the pixels that make up the overall display.

The LEDs used in this project are RGB LEDs that are contained in an 8mm package, meaning that each LED measures 8mm in diameter. The LEDs each contain 4 pins, used for power, ground, data in, and data out. To control the individual red, green, and blue LEDs that exist within the package, there is also an integrated circuit (IC) known as a WS2811 chip. This chip takes enough data to control each of the three inner LEDs, and then passes the remaining data to the next LED in the chain via the data out pin. Because of this, all of the LEDs are connected together in a snaking pattern that covers the entire Grid. Additionally, this allows for each LED to be controlled and addressed individually. The circuitry that makes up the packaged LED is shown below.

Each LED is also frosted slightly on the outside, allowing the RGB colors produced by the individual inner LEDs to combine together and appear as a single color, instead of appearing to have more of one color on a given side.

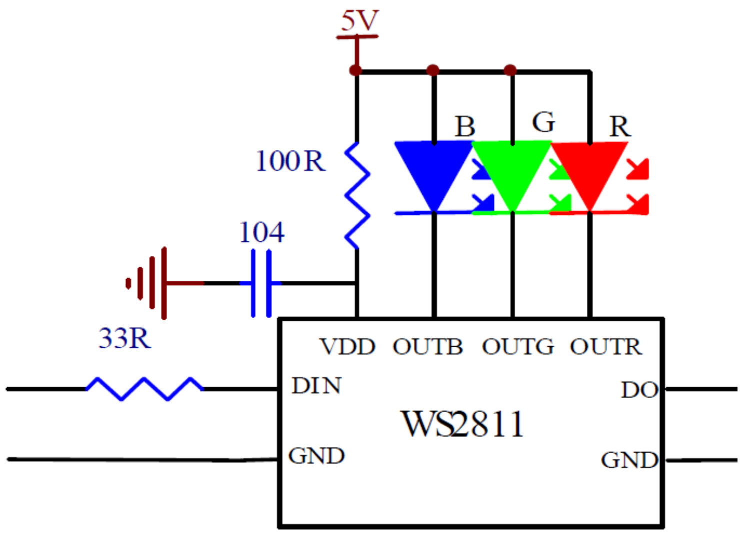

Figure 4. LED Diagram

PCBs

The printed circuit boards are integral to the project as they reduce the overall number of wires, and serve as the mounting hardware that supports the LEDs. PCBs are usually made of copper, and include etched connections into the copper that replace physical wires. PCBs are usually coated with a green soldermask that makes the top layer appear a shiny green color instead of the metallic finish of the copper used internally. The PCBs that make up the RGB LED Grid are covered in a white soldermask instead to reflect the colors of the LEDs better. Overall, the PCBs minimize the number of wires needed as well as provide a way for the LEDs to be supported in a grid fashion. Each PCB has two cutouts, one for power and one for ground that will also work to structurally hold the PCB at the appropriate height. The PCBs are one sided for cost reasons, and will contain all of the circuitry for powering the LEDs as well as routing the data signal. Due to limitations in manufacturing technology, the PCBs cannot be longer than 18” in any direction. As a result of this, the PCBs measure 18” by 0.65”. This results in 10 LEDs per board. The final configuration includes 4 PCBs per row, each with 10 LEDs, and 30 rows of PCBs. In total, there will be 120 PCBs supporting the 1200 LEDs.

Shown in Figure 5, the PCB layout is shown below with 10 LEDs per board. The layout also shows the location of the cutout holes for power and ground.

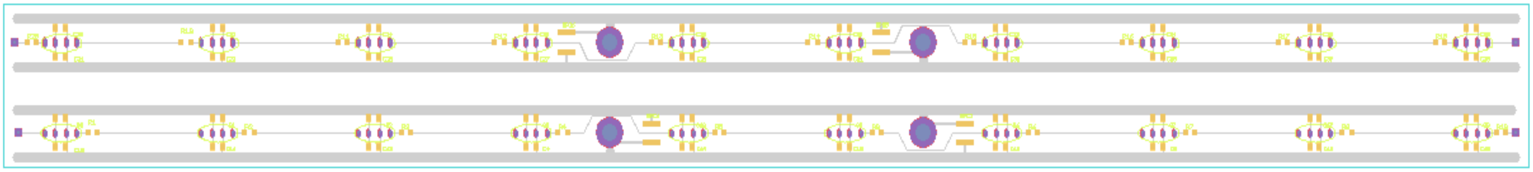

Figure 5. PCB Layout

FPGA

Field Programmable Gate Arrays (FPGAs) are hardware devices that are reprogrammable, allowing the programmer to achieve any logic configuration. These devices are well suited for use cases where the hardware may need to change quickly over time to meet the needs of other parts of the system.

As mentioned above in the LED section, the LEDs require data to come serially into their data in pin. The LEDs also require data to come in with a very rigid timing requirement. This requires that the device sending data to the LEDs must be able to operate at very high speeds. For this reason, an FPGA was chosen. The FPGA is used to translate the data that is in a human readable format into the timing specification used by the LEDs. However, is it not possible to program patterns and images in the FPGA due to the way that it is programmed via a hardware description language instead of a normal programming language. To accommodate this, the FPGA also decodes the data that is receives into the correct format, and interfaces with the LEDs. It is also capable of signaling when it needs more data. In the event that it is not able to get data that it needs as fast as it needs it, it will continue sending the data that it currently has, creating a lagging effect.

In order to implement the FPGA correctly for use in the RGB LED Grid, it needed to consist of a decoder module, a memory module to store the data, and an output module that sends the data out to the LEDs. The schematic depicting this process is located below in Figure 6.

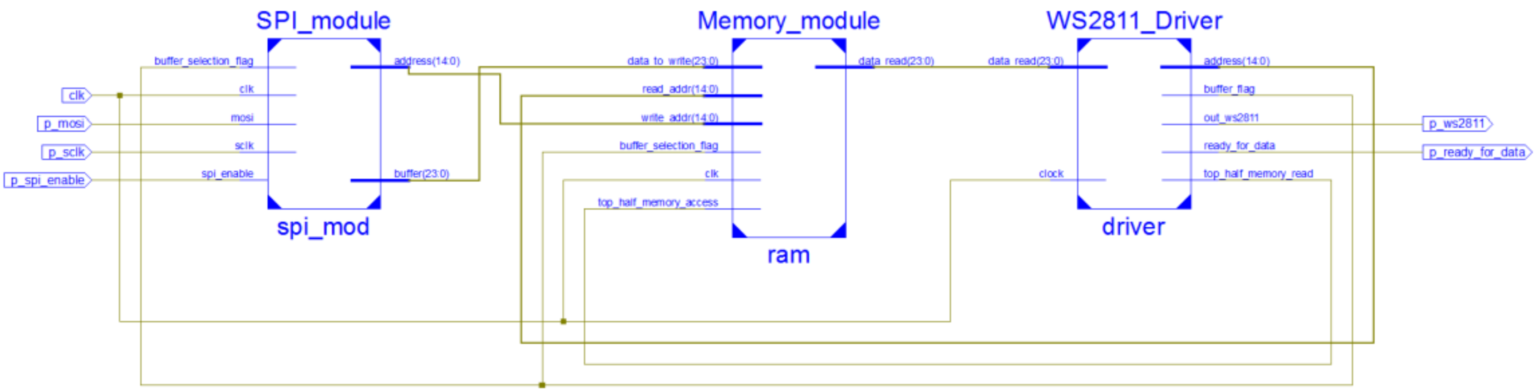

Figure 6. FPGA Schematic

Microcontroller

Microcontrollers are small computers that are often used to handle inputs and outputs as well as perform large amounts of computation. The microcontroller in the RGB LED Grid is needed to perform all of these tasks. The microcontroller performs all computation of generating patterns and sending them out in a format that the FPGA can read. This intermediary format is known as the Serial Peripheral Interface (SPI). The microcontroller generates a two dimension array of data that is then converted into a single long string of data. This data is sent via SPI to the FPGA that sends it to the LEDs. However, in order to maximize the amount of time that is spent generating patterns, Direct Memory Access (DMA) is utilized. DMA is the process of using built in hardware support to take pattern data that has been generated and move it to the SPI subsystem so it can be transmitted automatically as soon as it is generated.

Power Supply

In addition to the 2 data pins on the LEDs, each LED has a power and ground pin. These pins must be powered to +5 V DC. However, most power supplies are not capable of powering 1200 LEDs, as the voltage would drop over the course of the Grid. To alleviate this problem, the current draw of an individual LED was measured, and this value was scaled across the grid. The result of this experiment was the determination that roughly 80Amps would be needed to power the Grid. However, 80A coming from a single power supply is potentially enough to kill a human, or at least injure someone. The decision was made to break the Grid up into 4 panels, each panel being powered by a safer 20A power supply with backup fusing to ensure that if someone were to accidently connect the power supplies incorrectly that a fuse would break instead of sparking dangerously

.

Physical Apparatus

The physical apparatus of the Grid must be strong enough that is can support the LEDs and PCBs, while also enclosing the FPGA and microcontroller from tampering. Additionally, the two cutouts that exist in the PCBs that allow for power and ground connections are served by aluminum rods that are conductive. These rods are isolated from the remainder of the frame through electrical insulators. The physical apparatus is quite large, measuring almost 6' by 7' and is shown in Figure 7.

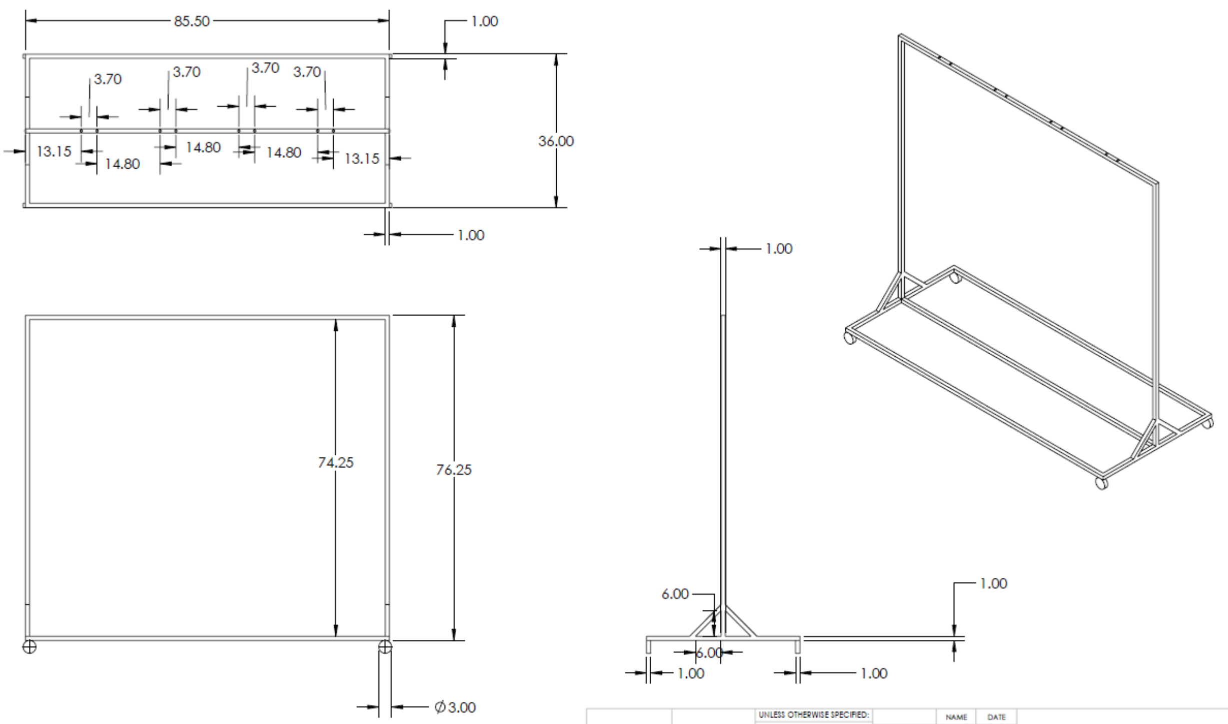

Figure 7. Supporting Apparatus of the Grid

All parts of the apparatus were carefully measured before being sent off to the UP Engineering Machine Shop for construction. Additionally, the physical dimensions of the inner part of the Grid that includes the PCBs and the LEDs is shown below in Figure 8.

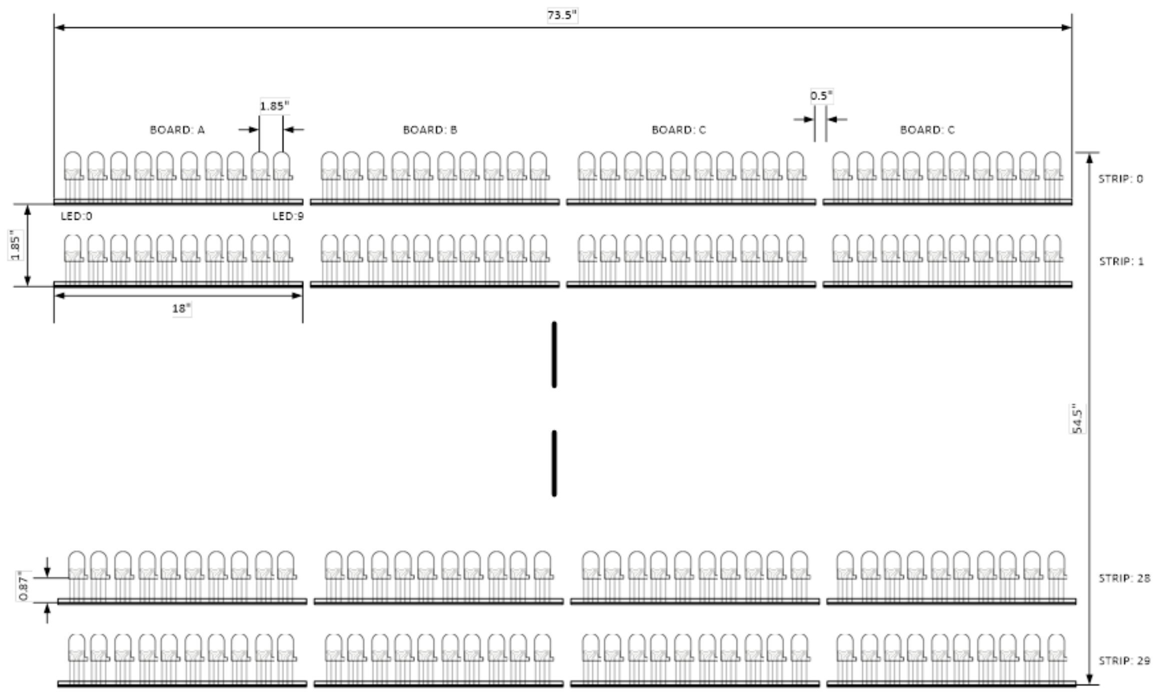

Figure 8. Physical Dimensions of the Grid.

Although the dimensions of the spacing between the PCBs were changed many times during the design process, these dimensions were finalized as they produced even spacing in the horizontal and vertical directions for all LEDs. The end result produces a display that is appealing on the eye and does not stretch in one direction. Additionally, the final resolution of the display is 4:3, meaning the ratio between the number of LEDs per row and column is 4:3. This was chosen as it is the standard display size for non-widescreen displays.

Patterns

In addition to the physical hardware that makes up the RGB LED Grid, software must be written that produces visually appealing patterns that show off the capabilities of the device. The first patterns that were written were quite simple, often just turning all LEDs on or off with all being the same color. However, these patterns soon developed into a snaking pattern of LEDs turning on to test the entire display.

Other patterns were needed, and the team was able to draw on patterns that are commonly used to demonstrate theater lighting and TVs in stores. This meant that there was a focus on mathematical based patterns.

In order to test the full functionality of the RGB LEDs, rainbow patterns that slowly faded across the grid were first developed, which soon evolved into fading pixels in and out. Soon, more complex patterns were needed, resulting in patterns showing exploding shapes, and balls bouncing around the display. Additionally, John Conway's Game of Life, a mathematical simulation was added, as were patterns depicting advertising for the University of Portland and the Shiley School of Engineering that could be displayed on Founders' Day.

Most of the patterns are cycled through based on either time limits or completion of the pattern. However, cycling through patterns can get boring once the cycle is repeated, so the team decided to have a constant "screensaver" like pattern that would only switch into a demonstration mode when a user walked up to the Grid. As a result, a motion sensor was added to enable this functionality. However, this left the choice of what to display whenever no one was looking at the Grid. This problem was solved by the idea of displaying a clock on the Grid when it was in an inactive state.

In recognition of the team's advisor, Dr. Aziz Inan, who loves palindromes, the clock has a special functionality that turns on for the minutes when the clock is displaying a palindrome time, such as 05:50, 12:21, or 20:02.

Simulation Software

Time constraints limited the amount of time that was available for pattern generation once the Grid was completed. This meant that patterns had to be created and tested in advance of having a fully complete Grid to test on. Software to simulate patterns was developed by the team that allowed for this. The software runs on windows, and allowed the team to directly copy and paste their code into the software and view what the patterns would look like on the full scale Grid. This software proved to be invaluable to the team, as it permitted the development of many patterns that would not have been possible without it.

A screenshot of the program displaying a pattern known as fading pixels that colors in random LEDs and slowly fades them back to black is shown below.

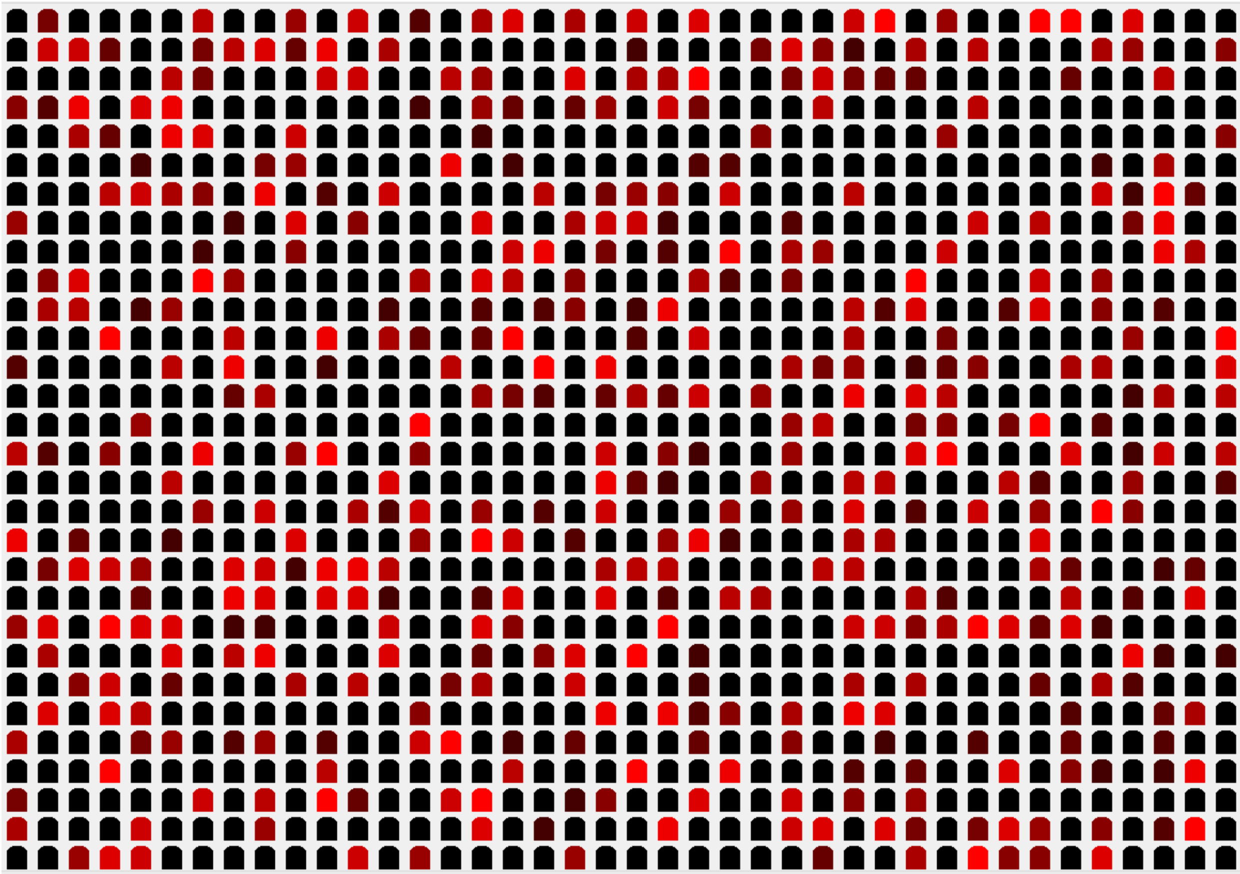

Figure 9. Screenshot from the Pattern Simulation Software

Challenges

Over the course of design, construction, and debugging of the RGB LED Grid, a number of issues were discovered. These issues were remedied with various engineering solutions taking cost, time, and other factors into account.

The first of the major issues that was discovered was with the metal that was used to power the Grid. Our original specification called for using stainless steel for all conductive parts. However, due to the chemical properties of stainless steel, is it not well suited to conducting electricity over large distances (> 3'). Since the overall Grid needs to move power 6', aluminum, which is a much better conductor, was used in place of the stainless steel. However, aluminum is far more malleable than stainless steel is, so the aluminum needed to be reinforced by additional support at the top of the Grid.

In addition, the physical support structure that was built to hold the Grid was designed to be short enough to fit through doors, and long enough to fit into an elevator for transportation. It was, however, not designed to be small enough in width to fit through doors. Due to this restriction, the Grid needed to be made thinner, which meant that fewer LEDs were used, as the Grid could not be made as tall as was originally planned.

However, there were also issues with the software that displays patterns. The FPGA was incredibly difficult to work with, especially in terms of scaling. In order to complete the project, small prototypes were built and tested over the course of the project. However, when scaling from 300 LEDs to 1200 LEDs, the FPGA was unable to complete this task due to its lack of internal hardware resources. This meant that a large amount of the code had to be rewritten so that the FPGA could store values in memory and read them later instead of handling all values at once. This meant that an additional memory module needed to be added, as well as all interfacing with the other modules. This was a large task that involved a significant amount of changing code.

The final challenge that occurred in the project was the designing of patterns. Creating patterns is difficult on its own, but it made harder due to constraints of the microcontroller which generates the patterns. Since the microcontroller is not as fast as a phone or laptop computer, only more simplistic patterns are possible. This eliminates the possibility of creating any patterns that would require the cosine of sine mathematical functions. Without this, it is impossible to create any rounded edges or circles. In order to fix this, a look up table with precalculated values was entered into the code that could be referenced to prevent the microcontroller from needing to calculate the value each time, instead being able to look this up in a table.

One pattern that was desired specifically was the ability to display clock time on the Grid. Although there are many modules that will keep precise time once set, these become incorrect with Daylight Savings Time or power outages. Since the desire was for a clock that could automatically set itself, the team looked into other options. One of the first that came up was using a cell chip like those found in phones. However, due to the low cellular reception in Shiley Hall, this was not optimal. Another alternative that was explored was GPS. The GPS signal includes the current time and date, and sends data that can be received in Shiley Hall. As a result, a GPS module was connected to the microcontroller and code was written to translate from GMT, which is transmitted by GPS to local time.

Conclusions

Despite these challenges, Team LOL was able to successfully implement the project, despite being slightly over budget. Overall, the project should be regarded as a success, and will be put on permanent display in the Shiley Hall first floor lobby.

If more time was allotted for this project, or a future team was going to continue this project, the changes that would include adding additional patterns to improve user experience, or adding a feature for more user interaction with the device than what exists currently. To improve the process of design and building, the team could create more documentation for each step including taking more pictures and explaining all useful construction techniques that were learned through experience. This documentation would be beneficial to others so that they could more easily understand the overall design to add new features or repair damages

The author and the team area proud of each step of the design and building process because they demonstrate a culmination of undergraduate studies at the University of Portland. Various technologies were utilized and skills learned in a wide range of classes were used. Some of the course topics that were applied include digital logic design, Verilog HDL, circuit analysis, programming, technical writing, public speaking, and team collaboration. As graduation approaches, this project represents the academic growth made during the last four years at the University of Portland in growing towards the Public Intellectual.

Additional information about the project is located on the team's website, found at the following URL:

engineering.projects.up.edu/TeamLOL

Bibliography

[1] L. Adams, J. Garcia, D. Nelson, A. Varvel. Functional Specifications. Fall 2014.

[2] L. Adams, J. Garcia, D. Nelson, A. Varvel. Design Document. Fall 2014.

[3] L. Adams, J. Garcia, D. Nelson, A. Varvel. Final Report. Spring 2015.

[4] Burlison, Alan. “Driving the WS2811 at 800KHz,” December 2012. [Online]. Available: http://bleaklow.com/2012/12/02/driving_the_ws2811_at_800khz_with_a_16mhz_avr.html. [Accessed: 01 October 2014].

[5] “Educational Circuit Board Project Program,” 2014. [Online]. Available: http://www.sunstone.com/news-resources/sponsorships. [Accessed: 14 October 2014].

[6] “Tiva Microcontroller Data Sheet,” March 2014. [Online: PDF] Available: http://www.ti.com/lit/ug/spmu365a/spmu365a.pdf. [Accessed: 14 October 2014].

[7] “TivaWare Peripheral Driver Library,” Febraury 2014. [Online: PDF] Available: http://www.ti.com/lit/ug/spmu298a/spmu298a.pdf. [Accessed: 14 October 2014].

[8] “Xilinx Spartan-6 FPGA Data Sheet,” January 2015. [Online: PDF] Available: http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf. [Accessed: 03 February 2015].

[9] “WS2811 Data Sheet – World Semiconductor,” August 2013. [Online: PDF] Available: http://www.adafruit.com/datasheets/WS2811.pdf. [Accessed: 4 September 2014].

[10] Analog Devices “Decoupling Techniques – What is it and why is it necessary,” March 2009. [Online: PDF] Available: http://www.analog.com/media/en/training-seminars/tutorials/MT-101.pdf. [Accessed: 02 February 2015].

[11] Schmitz, Tamara and Mike Wong. “Choosing and Using Bypass Capacitors,” October 2011. [Online]. Available: https://www.intersil.com/content/dam/Intersil/documents/an13/an1325.pdf. [Accessed: 29 January 2015].

Glossary

FPGA: A Field Programmable Gate Array is a type of programmable logic device. The functionality of the FPGA is designed to be programmed by the customer "in the field" and is commonly used for prototyping application specific ICs. FPGAs are programmed in a hardware description language such as Verilog or VHDL.

FPS: Frames per second is a measure of how many times the images is updated in a second.

IC: Integrated Circuit. An electrical circuit manufactured on a thin substrate of silicon using advanced photolithographic patterning techniques.

LED: Light Emitting Diode. This is a semiconductor light source, as well as a diode acting as a one way valve for current.

Microcontroller: A small, programmable computer on a single IC designed for embedded applications. In the case of the Arduino Microcontroller, this is an entire board consisting of the microcontroller IC and peripheral devices to allow easy programming and use.

PCB: A Printed Circuit Boards physically supports and electrically connects electronic components using conductive tracks, pads, holes, and other features etched from copper sheets laminated onto a non-conductive substrate with a silkscreen on top.

PPI: Pixels per Inch is a measure of the number of pixels that are shown on 1 square inch of the display. The maximum that a human eye can perceive is 300 PPI.

RGB: Red-Green-Blue. This refers to the red green blue color model that allows for the addition of red, green, and blue light to make each color within its spectrum.

SHP: Senior Honors Project. This project is completed for the University of Portland Honors Program and submitted to Dr. John Orr.

SPI: Serial Peripheral Interface is a serial synchronous data link that was created by Motorola. It uses a master/slave mode and is a four-wire serial bus.

WS2811: The IC that drives the LEDs. It follows its own data format that is referred to as the WS2811 proprietary format.