DESIGN DOCUMENT

***Team LED Mirror***

**EE 483 – Senior Design**

**Team Members:**

Nathan Lee - *Webmaster*

Jim Rowe - *Design Lead*

Dalton Faker - *Team Lead*

Kian Sadjadi - *Documenter*

**Faculty Advisor:**

Dr. Peter Osterberg

**Industry Advisor:**

Matt Clark

**Project Customer:**

University of Portland: Shiley School of Engineering

**University of Portland**

**Shiley School of Engineering**

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Version  | Author  | Date  | Reason for Change  |
| 0.9  | Team | 1/1/2019 | N/A |
| 0.95 | Kian Sadjadi | 11/8/2019 | Assigned for one voice |
| 1.0  | Kian Sadjadi | 11/15/2019 | N/A |

*Table 1- Revision History*

### Contents

[**REVISION HISTORY**............................................................................................................................................2](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25905)

[**LIST OF FIGURES**...............................................................................................................................................3](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25906)

[**LIST OF TABLES**.................................................................................................................................................3](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25907)

[**INTRODUCTION**.................................................................................................................................................3](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25908)

[**HIGH LEVEL ARCHITECTURE** ...............................................................................................................................4](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25909)

[Hardware Component..............................................................................................................................….6](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25910)

[Arduino.........................................................................................................................................................6](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25911)

[Zedboard………………………..............................................................................................................................6](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25912)

[LED Matrix....................................................................................................................................................6](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25913)

[Sensors..........................................................................................................................................................6](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25914)

[**SOFTWARE COMPONENTS**..............................................................................................................................7-8](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25915)

[**USER INTERFACE COMPONENT**............................................................................................................................8](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25917)

[**DIGITAL SYSTEMS**.............................................................................................................................................9](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25918)

[System Level Block Diagram.........................................................................................................................9](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25920)

 [I2C Timing Characteristics.............................................................................................................................9](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25921)

[Basic Power Analysis...................................................................................................................................10](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25922)

[Pin Definition..............................................................................................................................................10](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25924)

[Controller Software....................................................................................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25925)

**MECHANICAL COMPONENT**..............................................................................................................................11

[**IINTEGRATED CIRCUITS**....................................................................................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25926)

[**CONTROL SYSTEMS COMPONENT**......................................................................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25928)

[**SYSTEM TEST PLAN**.........................................................................................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25929)

[**DEVELOPMENT PLANS, MILESTONES, ASSUMPTIONS, RISKS, FACILITIES**..................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25930)

[**FINAL BUDGET**................................................................................................................................................11](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25931)

[**CONCLUSION**..................................................................................................................................................12](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25945)

[**GLOSSARY**.....................................................................................................................................................13](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25946)

[**BIBLIOGRAPHY**...............................................................................................................................................13](file:///C%3A/Users/ksadj/Desktop/Capstone%20Stuff/LED%20Mirror%20Func%20Spec%200.9.docx#_Toc25947)

LIST OF FIGURES

Figure 1: LED Mirror Example.......................................................................................................................4

# Figure 2: High Level Architecture Block Diagram..........................................................................................4

# Figure 3: User Interface Component............................................................................................................8

# Figure 4: I2C Data Transfer Clock Signal .......................................................................................................9

# LIST OF TABLES

# Table 1: Revision History.............................................................................................................................10

# Table 2: I2C Interface...................................................................................................................................10

# Table 3: Final Budget.................................................................................................................................11

# INTRODUCTION

The LED Mirror project is a creative take on a mirror utilizing modern and artistic elements. This idea was inspired by the Lots of LEDs project (a previous electrical engineering capstone project from 2012) with a plan to advance the display characteristics and add a user input. This project includes an LED/sensor array, Arduino, and Zedboard FPGA which allow the display to be interactive with its environment. The LEDs will also change color according to distance. Figure 1 is a fully fleshed out example of an LED Mirror. The user is meant to move themselves or an object over the sensors, and the corresponding LEDs attached to the sensor will light up, imitating a mirror. The final display will be an 8 x 8 matrix of LEDs and sensors encased in a 2ft x 2ft 3D printed frame, powered by a 5V DC power supply. The main points this document will address are the final design, high-level architecture of the device, the hardware components (sensors and LEDs), the software components (Arduino and Zedboard), the user interface, the plan on how to test the design, and lastly an update on the budget and overall development plan.



Figure 1: LED Mirror Example

HIGH LEVEL ARCHITECTURE



Figure 2: High-Level Architecture Block Diagram

HARDWARE COMPONENTS:

**Arduino:**

As seen in figure 2, the Arduino acts like a data buffer for the LED\_Data. The reason for this is because only the Arduino can perform the proper protocol to write data to the LEDs. In turn, the Arduino takes data from the FPGA (Field Programmable Gate Array) which takes it from the SPI (simple 4-bit signal) on the MOSI (Master input Slave output) pin and stores the data. It then sends it off to the DEMUX that decides which LED to send it to. The reason we chose the Neopixel LEDs is so that they can only be driven by an Arduino. This is because it is cheap and has one simple Data\_In and Data\_Out pin, making it very easy to use and control.

**ZedBoard (FPGA):**

The FPGA is the Master to all devices it communicates with. The reason it is the master is because it has the most logic to perform and needs to control the clock, so the other devices adhere to it. This is also where the programming of changing the colors of the LEDs based off distance will be stored and written.  The FPGA also performs the logic of incrementing the address and the type of data that is sent to the LED being addressed. Due to these responsibilities, the FPGA is best suited to be the Master of the system.

**LED Matrix:**

The LEDs that we will be using are the Adafruit Neopixel RBG LEDs. The single 64 LEDS will be laid out in an 8 x 8 matrix. Each LED is then connected to a sensor so that they can be addressed individually. Each LED takes a Data\_in signal from the Data\_out of a DEMUX whose signals are sent from the Arduino; as seen in figure 2. The LED takes in information from the sensor but not directly. The information passes through the FPGA then the Arduino before it reaches the LED so that each LED takes in the correct information and timing.

**Sensors:**

We will use SparkFun’s Distance Sensor Breakout boards which utilizes ST’s VL53L1X sensors. This board is an infrared time of flight sensor. It uses a vertical cavity surface emitting a laser to beam infrared light at an object and times how long it takes for it to reflect back. Although these diodes, or SPADs (Single-photon avalanche diodes), are in an array, they are individually independent of each other. This means the field of view of the sensors can be adjusted according to how many SPADs the device is programmed to use. They will be used to detect how far objects are in front of our LEDs and discriminate against objects that are too far or too close from them.

SOFTWARE COMPONENTS

**Arduino Input:**

**SCK:** A clock pin that controls the bit rate of the Master and the slaves.

**MOSI:** Master output Slave input. The pin that transfers 8 bits of data from the Master (FPGA) to the input (Arduino) on each transfer.

**SS:** Slave Select. The pin that controls what Slave device that the Master talks to.

**Arduino Input Specifications:**

The Serial clock pin is used to control the timing of the transfers from the Master to the Slave, to guarantee that there are no bit slips in the transfer. If both devices had their own clock there would be a high possibility that the clocks not being synced could cause transfer errors.

MOSI is the pin used to transfer the manipulated data from the FPGA that represents the distance the current Sensor reads to the Arduino; where it will be repacked and sent to the specific LED. The SS is not necessary, for there is one Slave, but due to the Arduino being a higher level deceive, using its software would be slower than the actual hardware. To fix this possible bottleneck, additional Arduinos can be connected in parallel to increase the transfer rate using the SS.

**Arduino Output:**

**MISO:** Master input Slave output. The pin that is used to transfer data from Arduino to the Master.

**LED\_DATA:** Serial pin that outputs data to the LED in 8-bit transfers.

**Arduino Output Specifications:**

MISO is used to occasionally check if the later LED\_DATA is being transferred properly by the Master reading the Arduino Data Buffer and checking it with the data it sent earlier.

**ZedBoard Input:**

**SDA:** Serial Data Address. It is the data pin that connects to every sensor and reads their data using I2C protocol.

**MISO:** Master input Slave Output. The pin that is used to transfer data from Arduino to the Master.

**ZedBoard Input Specifications:**

SDA is a bidirectional pin that transfers data to and from the Master (FPGA) for processing and manipulations. As seen in figure 2, the pin adheres to the I2C (Inter-Integrated Circuit) protocol where at the start of each transfer, the first 7 bits are an address to which sensor or slave to talk to. Then an 8th bit for write or read. The next transfers are the data such as what register to write, and what register to read from the Slave. The SDA is the address line, and the data line is the one making it ideal for a Master device with multiple slave sensors.

**ZedBoard Output:**

**SCL:** The Master clock that is sent to the sensors to control timing and bit rate.

**SCK:** The Master clock that is sent to the Arduino to control timing and bit rate.

**MOSI:** The data pin that transfers distance data from the sensor to the Arduino.

**Address:** 7 address pins that are sent to the DEMUX and are used as selects to determine what LED to update.

**SS:** Slave Select. The pin that controls what Slave device the Master talks to.

**ZedBoard Output Specifications**

The SCL and SCK are ported extensions of the Master clock controlled by the FPGA that go to the Arduino and sensors to ensure proper timing is held at each transfer of data to and from the FPGA. Address pins bypass the Arduino because the LEDs have no addressing logic so the addressing must be done external to the LED. Due to delay of the Arduino, the address pins will be set in the DEMUX before the data arrives.

USER INTERFACE COMPONENT



Figure 3: User Interface Component

The LEDs and sensors will be assembled in a square 3D printed casing as referred to in figure 3. The LEDs will be laid out in an 8 x 8 matrix with the same number of sensors below to correspond with each LED. The LEDs and sensors will be wired together using PCB arrays and encased in a 3D printed display. The display will be 2 x 2 feet. Attached to the back of the display will be a box containing our Zedboard and Arduino which controls the entire system. On the bottom of the display will be the 5V DC power supply that runs off a 110V AC wall outlet. The actual user interfacing for the project is very simple. There are no buttons or manual activators at all. The user simply needs to put an object or body part in front of the sensors, and each corresponding LED will light up.

DIGITAL SYSTEMS

**I2C Timing Characteristics**



Figure 4: I2C Interface

Communication between the sensors and FPGA will be done through I2C. I2C is performed by devices with an agreed upon clock using two signals, SDA and SCL, as seen in figure 4. Information is packed in bytes and are followed by an acknowledge bit, issued by the receiving device. Timing characteristics can be seen in Table 2.



Table 2: I2C Interface

**Basic Power Analysis**

The device will be powered from a 5V DC power supply that runs off a 110VAC wall outlet. The Based on previous projects and intuition, the wires and internal device resistance of the sensors and LEDs, this project may require a higher voltage to be divided between multiple points in the circuit. The sensors use 20mW at 10Hz, each.

**Pin Definitions**

Sparkfun VL53L1X sensors have 6 pins, in order from left to right, are GND, 3.3V, SDA (Serial Data Line), SCL (Serial Clock Line), INT’, SHUT’. These devices have pre-built cable interfacing systems which can be used to reduce soldering if needed. NeoPixel LEDs will have VCC, Data\_In, Data\_Out, and GND. The Arduino will have LED\_Data, and address, to program colors into the LEDs.

**Controller Software**

Interfacing between the FPGA and sensors will be performed by he Arduino. Neopixel LED’s and Sparkfun sensors both have Arduino libraries. These libraries will be used to simplify communications between devices.

MECHANICAL COMPONENT

The only mechanical component of this project will be the 3D printed casing which will house the LEDs, sensors, Arduino, and Zedboard. The ultimate purpose of the casing will be to protect the electronic components from dust, static electricity, and other hazards while containing the display as a visually appealing final product. An example of this can be referred to in figure 3 of the user interface section.

CONTROL SYSTEMS COMPONENT

N/A

INTEGRATED CIRCUITS

N/A

SYSTEM TEST PLAN

Once we have received the parts for the project, we will test each sensor and LED individually to make sure each component is in full working condition. From there, we will create a working model utilizing one LED, and one sensor as a proof of concept to make sure all the parts work together as intended. After that is accomplished, we will begin building it to full scale.

DEVELOPMENT PLANS, MILESTONES, ASSUMPTIONS, RISKS, FACILITIES

The development plan, milestones, assumptions, risks and facilities were all described in detail in the Functional Specifications Document. Please refer to that document for information on those aspects of the project.

FINAL BUDGET

The Final Budget listed below in Table 3, is a more specific version of the budget originally proposed in the Functional Specifications Document. This is a more accurate list of the exact products and their pricings we need to purchase.

|  |  |  |  |
| --- | --- | --- | --- |
| Item     | Quantity    | Cost per Unit    | Total Cost    |
| Adafuit Neopixel LEDs  | 64  | $1.00  | $64.00  |
| Sparkfun VL53LIX Sensors  | 64  | $20.00  | $1280.00  |
| Arduino  | 1  | $15.00  | $15.00  |
| Zedboard - ZYB0 ZYNQ 7000  | 1  | $189.00  | $189.00  |
| 5V Power Supply  | 1  | $30  | $30.00  |
| Wire Bundles  | 2  | $11  | $22.00  |
| Miscellaneous (Extra Parts, Materials, and Shipping)  | 1  | N/A  | $100.00  |
| **TOTAL**  |    |    | **$1700.00**  |

Table 3: Final Budget

CONCLUSION

Overall, the LED Mirror project has been making great progress as everything has been on schedule and the design of the project is 100% complete. The parts for the project have been chosen and the high-level architecture design has been finalized. Other important objectives to complete will be ordering parts and testing them individually, then assembling them in order to achieve a proof of concept. We will also be working on the code to control the Zedboard and Arduino. The final project will need to be constructed before spring break in March 2020. The project will continue to meet all the expectations of the University of Portland, our faculty advisor Dr. Osterberg, and our industry advisor, Mr. Matt Clark.

GLOSSARY

**LED -** A light source that is created by a light emitting diode.

**PCB –** Aluminum circuit board material used to connect the LEDs together.

**Arduino -** An inexpensive single board computer (micro-controller) which can be programmed and has I/O pins allowing for easy interaction with other electronics.

**FPGA -** Field Programmable Gate Array can be used to simulate hardware designs.

**Zedboard –** The FPGA of the project used as the master that controls vital functions listed in the documentation above.

**SPAD -** Single-photon avalanche diode. A solid-state photodetector.

**Demux -**   a device that takes a single input line and routes it to one of several digital output lines.

**Clock Signal -** A particular type of signal that oscillates between a high and a low state to control a component.

**I2C -** Inter-Integrated Circuit used for communication between the sensors.

**SDA -** A bidirectional pin that transfers data from and to the Master(FPGA) for processing and manipulations.

**SCL –** The Master clock that is sent to the sensors to control timing and bit rate.

**SPI -** A simple 4-bit signal that is used for communication between hardware.

BIBLIOGRAPHY

<https://www.arduino.cc/en/Hacking/PinMapping168>

https://cdn.sparkfun.com/assets/8/9/9/a/6/VL53L0X\_DS.pdf