Final Report

E44: Electronic LED Display

Caroline Gilger, Logan Wedel, Drew Maatman, Kevin Etta, Tuoxuan Ren

3 May 2019

Table of Contents

Abstract

Project Team:	Caroline Gilger	Faculty Advisor:	Dr. Cris Ababei
	Kevin Etta		
	Drew Maatman		
	Logan Wedel		
	Tuoxuan Ren		

Electronic displays are used in many applications ranging from billboards on the side of a highway, to scoreboards in a sports arena, to the main user interface for a computer or smartphone. Most displays used in consume electronics such as computers or smartphones have used liquid crystal displays (LCDs) as their display hardware, but LEDs are also a popular choice. More recently, Organic LED (OLED) displays have increased in popularity. LED displays are a popular choice for open source hardware projects since they require far less upfront engineering work before they can be used to display images, video or text. While LCD displays require custom integrated circuits to be able to drive them, as well as requiring much more energy, LED displays are sold in panels with standardized serial interfaces. These LED panels can be strung together in a serial fashion to be able to drive many panels from a small number of digital logic signals. This project describes the creation of a 320 by 256 pixel LED display system with 9 bit color resolution, fully integrated logic and power, and an Android App user interface.

The purpose of this project was to develop and electronic display consisting of LED panels. In order to accomplish this, a custom logic PCB was developed, hosting a 32-bit, 252MHz microcontroller, 8 serial Flash memory chips, and a complex image caching system. In addition to the logic board, a power PCB was developed to safely power the LED panels. This power PCB sports a quad-phase 1MHz buck converter, which can supply 5V at up to 45 Amps. The buck converter features many protection and ruggedization features to ensure the LED panels are protected against electrical faults. The project required an extensive microcontroller firmware to be developed to allow the microcontroller to draw images on the LED panels, control the flow of image data, and interact with users. The main user interface is an Android App which allows users to select up to eight images to be displayed in a slide-show like fashion, and the user can select the cycle time between images as well. The android app also allows the user to dim the screen and turn it on and off.

The final project yields a 320 x 256 pixel LED display, the two custom circuit boards, a consumer-off-the-shelf power supply, and a custom fabricated aluminum frame. It can display any set of eight images through the android app, and has a color density of 9 bits of color. The final product is about 2.5 feet by 2 feet in size. Both circuit boards and the COTS power supply are attached to the back of the 20 LED panels, so the project is self-contained, and creates its own WiFi network that a user can connect to using an Android phone. The display is vibrant, and has a refresh rate of 60 Hz, without any visible flickering.

Problem Definition

The Electrical Engineering and Computer Engineering (EECE) Department lacks a modern electronic display for presenting information to students, faculty, and prospective students. This project is intended to fill that need while also providing an opportunity for the department to showcase the engineering talents of Marquette students.

This project's scope is limited to creating a functional color LED display with resolution of 256x320 for the Electrical and Computer Engineering Department office. The display is capable of displaying images which are programmed to the display controller using an Android app interface. The display operates autonomously once programmed. We have prepared a paper for HardwareX, a journal dedicated to presenting open-source hardware projects.

Customer Needs

After speaking with a few of the people who would be receiving this product, a list of customer needs was developed. The customer needs developed can be seen in the table below.

Table 1. The table below describes the customer needs required for the project design.

Question	Customer Statement	Interpreted Need	Need #
Typical Uses	The current display is meant to display the phrase information to students, visitors, and professors passing through EECE office.	The display must be large enough to display information meaningfully to passersby	1
	I want something that will change frames and display multiple images and text messages.	The display must have enough memory to store multiple images	2
	Large and visible display	The display should be big enough to have a resolution which allows for the text to be easily read.	3
	A phone app to control what is displayed on the screen	An Android app will interface with the screen over Wi-Fi to upload images and data	4
Likes – current LED display	Displaying information is helpful to passersby	A functioning display is needed	5
Dislikes – current LED display	The current display does not work anymore	A functioning display is needed	

	It has a low resolution	A higher screen resolution is needed	6
	It can display one color	The display needs to display more vibrant color	7
Suggested Improvements	Wi-Fi connectivity, full color, larger size and resolution, enhanced functionality	Wi-Fi connectivity needs to be included	8

Target Specifications

Table 2. The table below shows the interpreted target specifications based on the customer needs discovered above.

Original Customer Needs	**Original Target Specifications**	**Final Specifications**	**Specification Met?**
Large display to portray meaningful information	Size must be approximately 3 x 2 feet	2.6 x 2 feet	Yes
	MCU speed must be at least 200 MHz	MCU speed is 252 MHz	Yes
Enough memory to store multiple images	8 different slides	8 different slides	Yes
High resolution so that text can be easily read	Resolution must be 320 x 240 RGB LED pixels	Resolution is 320 x 256 RGB LED pixels	Yes
Android App allows for connectivity over WiFi	Connection from app to device must be across WiFi interface	Connection from app to device must be across WiFi interface	Yes
Must display vibrant colors	Color resolution must be 12 bits (4096 colors)	Color resolution is 9 bits (512 colors)	No
Must have multiple saved “projects” to switch between	15 saved configurations	25 saved configurations	Yes

Justification of Design

Hardware

Power Board

To power the LED panels, a power system was designed to provide the required 45A using a 5V buck converter. The power board was needed to provide a tightly regulated and safe voltage for the LED panels, and had various protection features such as overvoltage, overcurrent, and over temperature protection. These features were important because if anything were to go wrong with the power supply or the LED panels, the power board would turn itself off to protect itself and the LED panels from being destroyed.

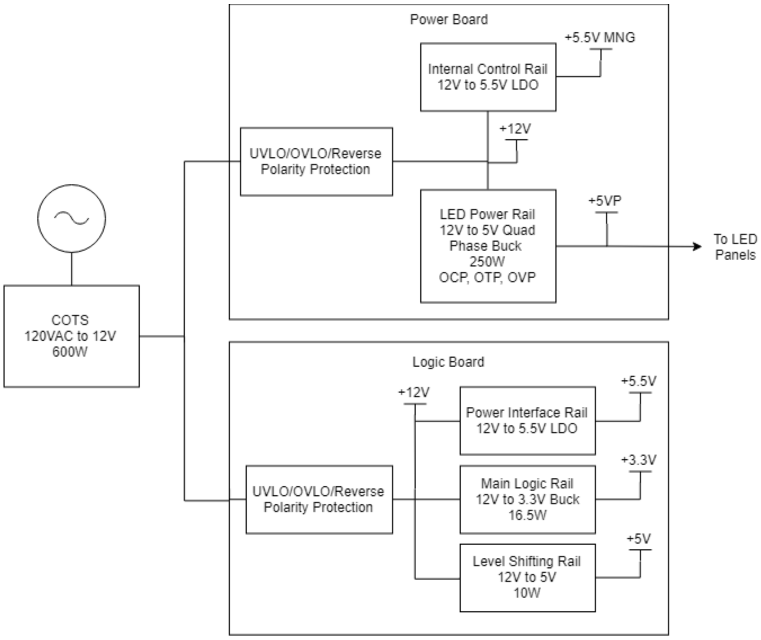


Figure 1. A diagram for the power supply breakdown for all parts of the project.

Logic Board

The hardware concept which was selected for this project was to use a PIC32MZ2048EFH144 microcontroller running at 252MHz using 2MB of flash memory, 512KB of SRAM, and a 32-bit bus width. Using a microcontroller instead of an FPGA is because the associated hardware would be much more complicated with an FPGA where a microcontroller would be less complex. The general-purpose computer such as a Raspberry Pi would not be fast enough to output a proper image with no flicker. The specific 32-bit microcontroller was chosen because it has a clock speed fast enough to provide a smooth image display without flicker. Also, it has 144 pins which is required for providing data input to the LED panels.

The external memory required was 256kB parallel SRAM for the frame buffer cache along with 8 256kB serial flash memory for storing the 8 slides for the display. The 8 256kB flash chips store the existing project slides and the 256kB SRAM stores the current slide being presented on the display. The microcontroller copies the information from a flash chip and stores it into the SRAM. After the data has been stored into the SRAM, it displays the image onto the display. Flash chips were chosen because they are onboard chips which can be located physically close to the microcontroller to reduce any delays with communication. Flash chips typically can be overwritten 100,000 cycles.

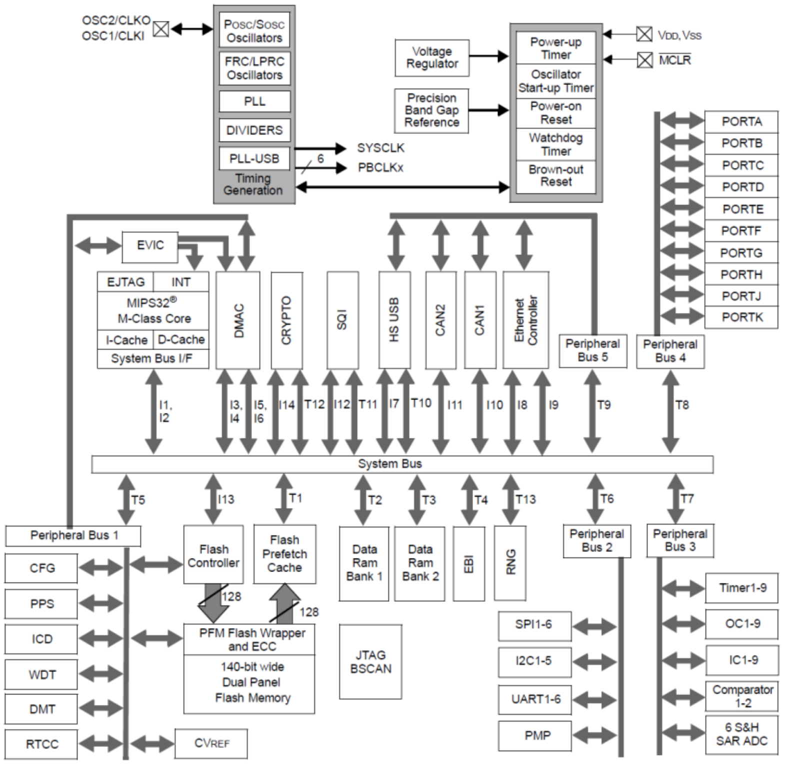


Figure 2. Block diagram of PIC32MZ microcontroller chosen

The WiFi module which was used was the ESP8266 module because it provided its own connection between itself and the Android app. Thus, it was not dependent on external WiFi such as Marquette WiFi. For quick debugging, an FT234 USB-UART bridge was designed which has a ground isolated from the rest of the circuitry.

Embedded Firmware

The firmware which was implemented on this design was written in C programming language. It was written completely from scratch including all of the pin out initializations, interrupt functions, and other very low-level operations required for the system to run. This was done to make the firmware completely customized and to keep the program as optimized as possible. The firmware had to run very efficiently in order for the panel to display a clear image with no flicker.

Software

The software concept which was selected for this project was an Android app to communicate with a WiFi module to send images and text slides to the microcontroller. The

Android app will have the capabilities to create a new project, edit an existing project, preview the existing projects, and upload projects onto the microcontroller. There will also be the capability to add multiple display board's information so that the scope of this app is extensible for the future possibility of multiple boards controlled by one Android device. While adding new images to a "project" the user will be able to choose whether to resize the image to fit the display screen or keep the aspect ratio of the image intact. The app also has the capability to turn the display on and off, along with adjusting the brightness.

Display Frame

The display frame was created to hold the display upright and keep all of the panels together in a secure way. The material used for the frame was chosen to be 80/20 aluminum. This was chosen because it was an inexpensive but reliable material to keep the panels together. The 80/20 was used along the border of the panel and as legs behind the panel to keep it upright. Additionally, acrylic squares with screw holes were used for connecting each panel together in the middle of the display. This kept the panels tightly connected together.

Characteristics of Design

Overall Block Diagram

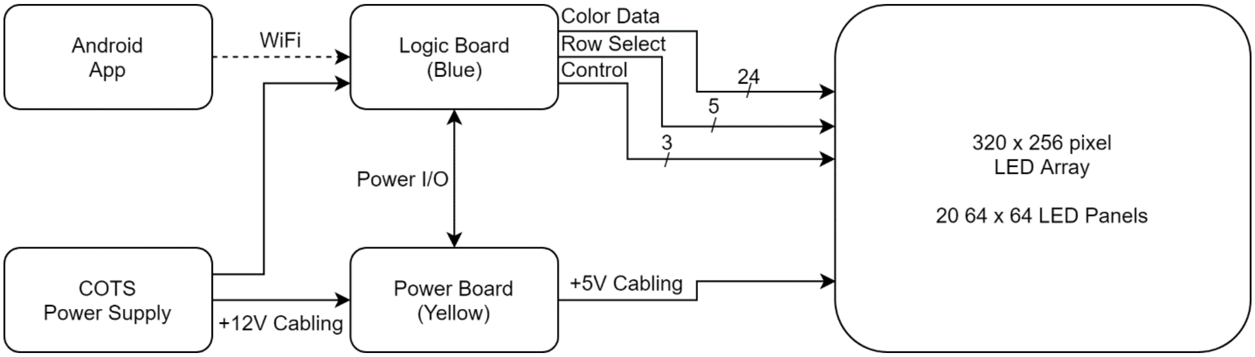


Figure 3. The image above is a block diagram of the full design concept

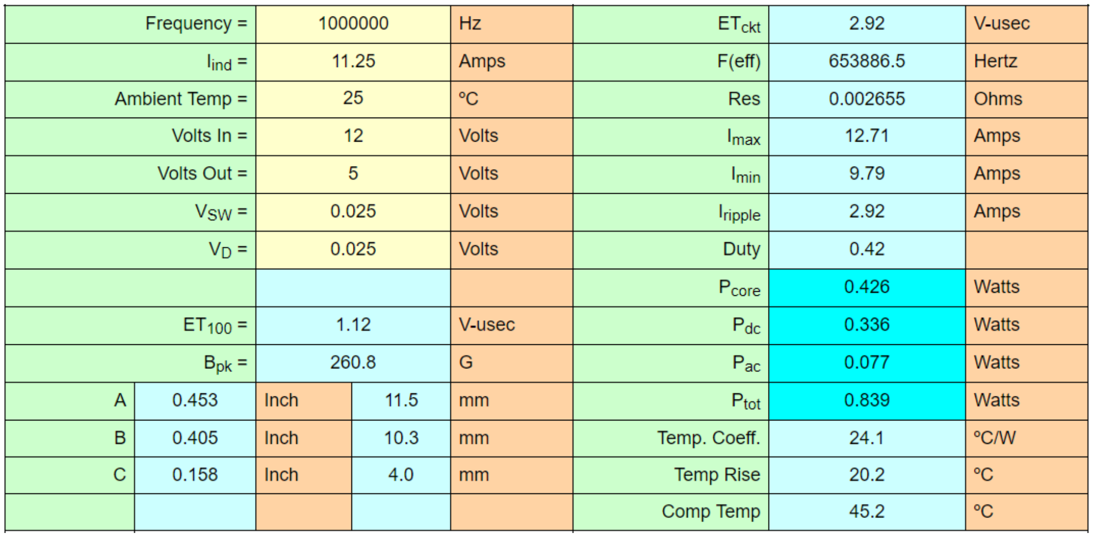
This project contains 5 major parts. It contains 20 LED panels each containing 64x64 pixels with each pixel holding three LEDs: red, green, and blue colors. Overall there are 245,000 LEDs that need to be driven. The logic board built is the main controller of the system. It holds the image data that the panels display, talks to the Android application through WiFi and can control the power board through various power I/O pins. The power board was built to safely power all the LED panels. Also, a consumer-off-the-shelf 120V to 12V power supply was purchased to plug into the wall to power everything. The final part is the Android app that lets a user customize and configure what images the board displays. Figure 2 shows an overall block diagram of the project.

Hardware

<u>Power Board</u>

A power supply circuit board was designed to regulate a very tight 5 volt power rail to be distributed to the 20 panels, since the panels require a very tight tolerance voltage supply. Each panel was found to require around 2 amps of current at a worst case scenario of displaying a fully white image at the maximum brightness. Because of this, the power supply would need to supply at least 40 amps of current at 5 volts.

To give enough headroom, the supply was designed to provide a maximum of 45 amps. In designing the power supply, a four phase buck converter topology was selected since the individual passive components within the supply would be stressed less by distributing the power processing across four phases. At this high of an output current, the inductors within the supply would need to be able to handle a peak current of 11.25 amps per phase. Vishay powdered iron inductors were selected because of their good thermal properties and higher saturation current, and the supply was designed around using this inductors. Various computer simulations were run to determine if the power supply would function as expected.



Frequency =		1000000		Hz	ET_{ckt}	2.92	V-usec
I_{ind} =		11.25		Amps	F(eff)	653886.5	Hertz
Ambient Temp =		25		°C	Res	0.002655	Ohms
Volts In =		12		Volts	I_{max}	12.71	Amps
Volts Out =		5		Volts	I_{min}	9.79	Amps
V_{SW} =		0.025		Volts	I_{ripple}	2.92	Amps
V_D =		0.025		Volts	Duty	0.42	
					P_{core}	0.426	Watts
ET_{100} =		1.12		V-usec	P_{dc}	0.336	Watts
B_{pk} =		260.8		G	P_{ac}	0.077	Watts
A	0.453	Inch	11.5	mm	P_{tot}	0.839	Watts
B	0.405	Inch	10.3	mm	Temp. Coeff.	24.1	°C/W
C	0.158	Inch	4.0	mm	Temp Rise	20.2	°C
					Comp Temp	45.2	°C

Figure 4. The image above is a screenshot from the Vishay Inductor Calculator used to calculate the inductor power loss and temperature rise within the power supply.

Figure 3 shows results of a calculation for inductor power loss and temperature rise within the power supply. The operating parameters such as switching frequency (1MHz), input voltage (12V), output voltage (5V), and phase output current (11.25A) were entered based on the design of the power supply for the LED panels. The output parameters of the calculation show that the inductor will have a 20C temperature rise (which is about the highest temperature rise that can safely be tolerated), and the power dissipation will be about 850mW. This calculation proves the 1uH Vishay inductors selected will function properly in the context of this power supply.

Besides the inductors within the power supply, the control loop of the power supply needed to be compensated to ensure that the output voltage reaching the LED panels was tightly regulated across a wide range of frequencies. Components for this function were also selected through a simulation. The controller selected for the power supply was the LTC7851, which is a four phase voltage mode control buck controller. Linear Technology, the company which designed the chip, also provides a simulation tool to accomplish most aspects of designing the power supply. The simulation tool was used to properly compensate the control loop.

Figure 4 shows the simulated magnitude response of the control loop for the power supply. The bandwidth is around 12.6kHz, which is substantially lower than 1/20th of the 1MHz switching frequency, which is desired. The bandwidth is lower because of the massive amount of output capacitance used in the power supply. The phase margin, not shown, is around 60 degrees, which is higher than the 50 degree benchmark that should be exceeded for control loop stability. This essentially proves the power supply should be stable across a wide range of frequencies and operating conditions.

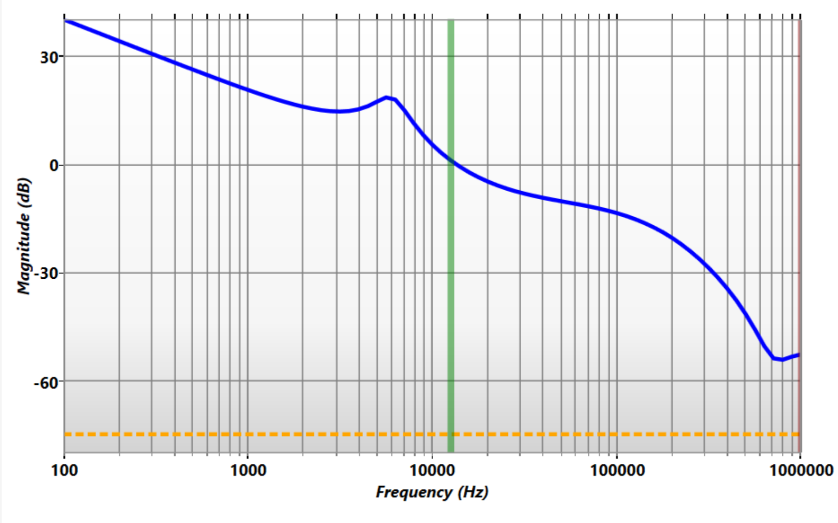


Figure 5. The image above proves that the power supply control loop will be stable.

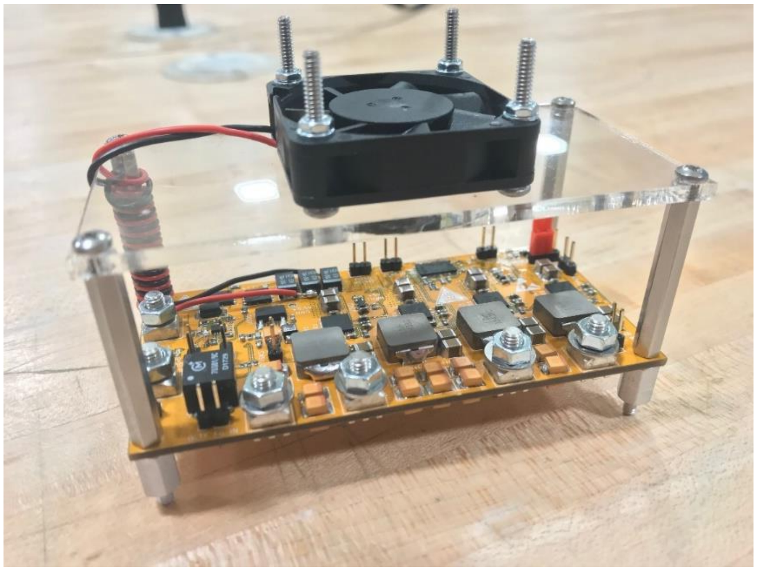


Figure 6. The image above is a picture of the power system board with the components populated.

The finished power circuit board is shown in figure 5. In addition to the inductors and control loop components, input capacitors, output capacitors, power stage components, and input power protection were designed such that this power supply can regulate 5 volts at 45 amps.

Logic Board

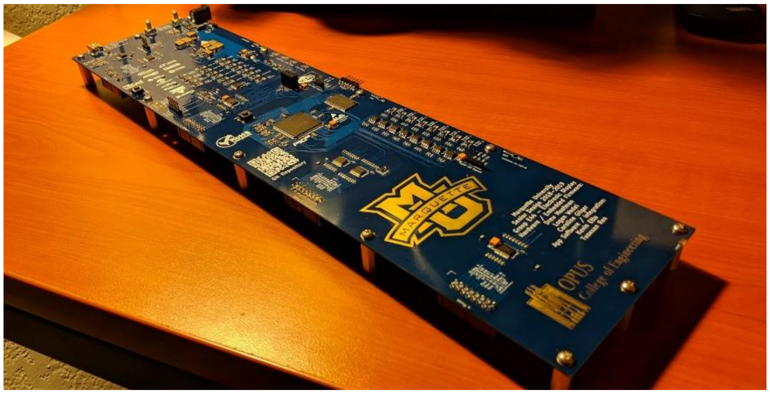


Figure 7. The image above is the completed Logic board with all parts soldered on.

The logic board serves as the center of the project. It holds the PIC32MZ EF microcontroller, external memory, 3 point of load power supplies, a USB debugging interface, the WiFi module, high speed level shifters, debugging LEDs, the microcontroller programming interface, interface circuitry for the power board, data connectors for interfacing with the LED panels, and lots of silkscreen and copper artwork. The microcontroller is at the center of the logic board, and a plethora of decoupling capacitors surround the MCU. An input protection circuit protects all downstream circuits from input voltage that is too high or low in magnitude, or reversed in polarity. Two buck converters provide 3.3V for most components along with 5V for the level shifters, which are required to allow the 3.3V MCU to control the 5V logic level LED panels. A 5.5V LDO biases interface circuits which allow the MCU to control and monitor the power board, which has a logic level of 5.5V.

In addition to driving the display, the microcontroller also has several other tasks to service. During normal operation, up to eight images are stored across eight 256kB serial Flash memory chips. This external non-volatile memory acts as long-term image storage, and stores image data across MCU resets and power cycles. It is accessed through a Serial Peripheral Interface (SPI). The MCU can read and write entire images worth of data to a Flash chip in around three seconds. When a project is programmed into the system, each image is stored in the SPI Flash, while the number of images for the project and the cycle time between images is stored into the microcontroller’s internal Flash.

The microcontroller also has an external parallel 256kB SRAM. It is able to read and write data to this SRAM through the External Bus Interface (EBI). EBI allows external memory to be attached to the MCU and it treats it just like any other virtual memory address space. This allows data in EBI SRAM to be treated like a normal byte array in C firmware. During normal operation, while an image is being shown on the screen, the next image to be displayed is moved from SPI Flash into EBI SRAM. When the image is ready to be displayed, it is moved all at once from EBI SRAM into the internal frame buffer and shown on the screen. In this memory structure, EBI SRAM acts as a data cache, since moving data from EBI SRAM into internal RAM is much faster than moving data from SPI Flash into internal RAM. Using this technique, image transitions appear seamless.

The MCU has a serial interface with a WiFi module included on the logic board. This is the main interface between the Android app (described in the next section) and the system. Through the WiFi interface, the user can turn the screen on and off, adjust the brightness of the screen, and load projects into the system. The WiFi module and the MCU communicate through a UART interface. A ring buffer in MCU firmware allows massive amounts of data to be transferred at a time from the WiFi module into the MCU.

The MCU also has many other features to help with debugging. These include a ruggedized USB debugging interface, which was implemented with a UART to USB bridge. This allows a user to connect to the microcontroller through a 115.2 Kb serial connection from a software such as PuTTY or Tera Term. This was a crucial debugging feature when developing the MCU firmware. The MCU's analog to digital converter was also used to be able to measure system level power supply voltages. The ADC can be initialized through the USB interface, and results can be printed through the serial interface. The ADC is only initialized when the user requests it; this allows for the CPU bandwidth to be maximized when the debug feature is not needed. Finally, the MCU can monitor digital signals returning from the power module. This allows the MCU to record power regulation faults and over-thermal protection events. The MCU logs these events, along with other faults, in an error handler C structure. The flags within this structure can be accessed through the USB debugging interface and are used to also drive LEDs on the logic board. This feature allows the user to see if any faults have occurred at run time.

Embedded Firmware

The firmware was written in C programming language which allowed for very simple low-level functions to be implemented efficiently on the microcontroller. A few important aspects of the firmware included the multiplexing algorithm, the UART ring buffer and the state machine algorithm. The multiplexing algorithm is interrupt driven. Thus, once an interrupt has occurred, it begins a sequence of commands to shift data into each row, drive that LED row high to display the LEDs, then increment to the next row and repeat that step. There are two rows of LEDs which data can be shifted into and LEDs can be displayed at one time per panel. Therefore, for the entire display, only eight rows are displayed at one time. This can be seen in Figure 7 which is an image of a prototype developed with only four (2x2) panels. As it is shown, there are only four rows on at one time because each panel has two rows on at a time.

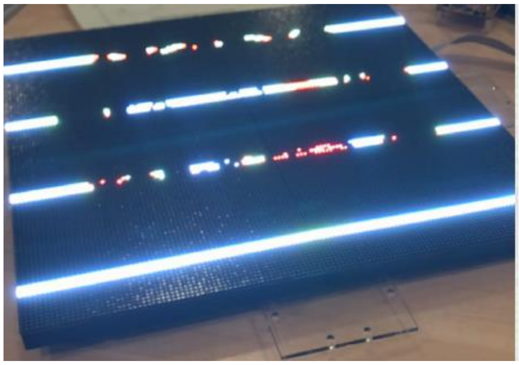


Figure 8. The 2x2 panel prototype showing how data is multiplexed across the panels.

Due to the fact that the RGB LED pixels can only show 8 distinct colors at one time, the screen is refreshed 8 times with different color components. The viewer's eyes blend these 8 frames into one image. The multiplexing algorithm can be shown in the flowchart in Figure 8.

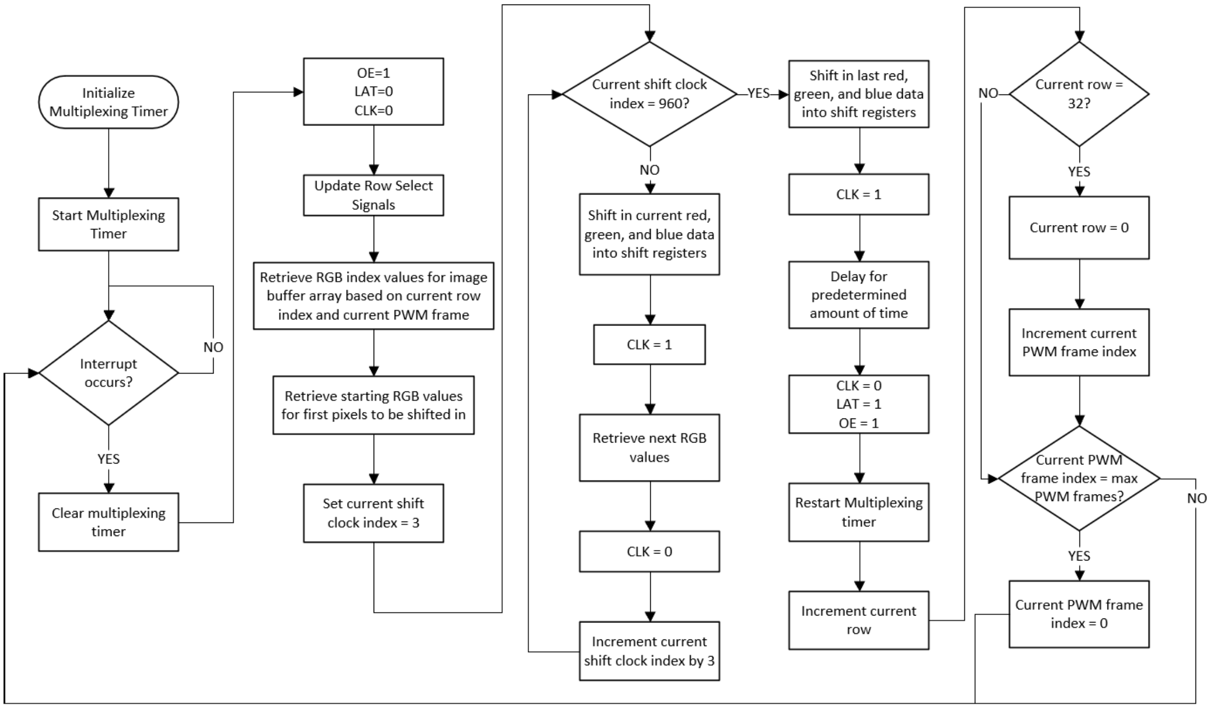

Initialize Multiplexing Timer
Start Multiplexing Timer
Interrupt occurs?
NO
YES
Clear multiplexing timer
OE=1
LAT=0
CLK=0
Update Row Select Signals
Retrieve RGB index values for image buffer array based on current row index and current PWM frame
Retrieve starting RGB values for first pixels to be shifted in
Set current shift clock index = 3
Current shift clock index = 960?
YES
NO
Shift in current red, green, and blue data into shift registers
CLK = 1
Retrieve next RGB values
CLK = 0
Increment current shift clock index by 3
Shift in last red, green, and blue data into shift registers
CLK = 1
Delay for predetermined amount of time
CLK = 0
LAT = 1
OE = 1
Restart Multiplexing timer
Increment current row
Current row = 32?
NO
YES
Current row = 0
Increment current PWM frame index
Current PWM frame index = max PWM frames?
NO
YES
Current PWM frame index = 0

Figure 9. Flow chart of multiplexing.

Another development in the firmware was the state machine algorithm. The state machine works so that it is automatically started at power up. The first state includes loading in the first flash chip while displaying the “splash screen”. Then, it will incrementally load in the next flash chip while displaying an image at the same time. Once an image has been loaded into the EBI SRAM and the timer set by the user has completed, the new image is loaded into the microcontroller internal memory to be displayed on the screen. To stop the state machine, all SPI interrupts are disabled so that it does not service this process.

Software

A custom Android App, named “MU-MatriX” was developed to allow the user to create and upload to the display system new “projects”. A “project” consists of eight images that form a presentation displayed periodically at a configured interval. The app stores each project’s name, set of images, and duration between image switch into its internal application storage. After naming a new project, a user can populate their project with images, selecting them one-by-one from their device. The images can maintain the aspect ratio of the original image or stretch to fit the size of the display board. There is also the option to rotate images in 90 degree intervals before adding them into the project. In the background, any selected image is saved off into a 320x256 bitmap with black background filling in any gaps created by the aspect ratio lock. As images are being added to a project, they can be reordered by simple up and down arrows next to their thumbnail on the "Edit Project" screen. A spinner widget provides a simple dropdown selection for the project-level property of duration between slides.

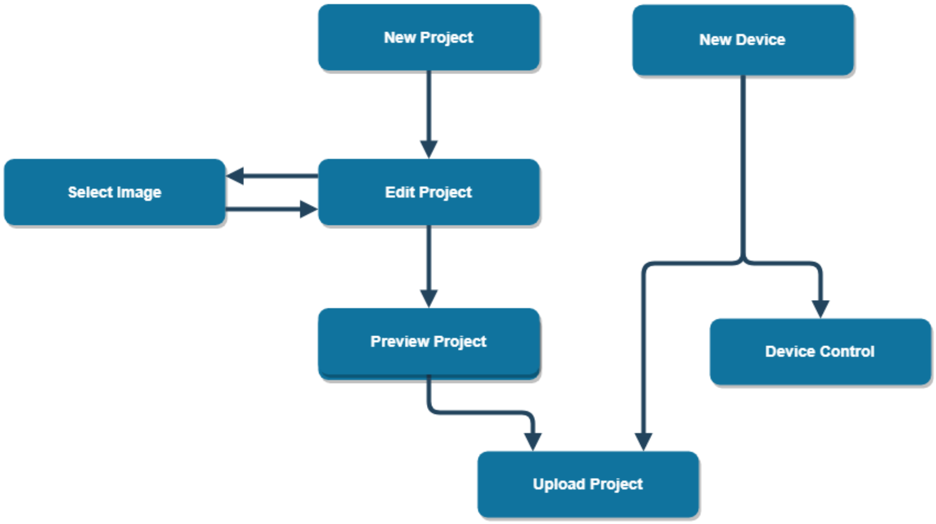


Figure 10. Overall structure of the Android Application

The default display board device is automatically stored under the user's device list on app installation. The IP address and port number used by default for the ESP8266 WiFi module (on the logic board) are simply placed under a device name "Default Display Board". The app allows more than just this one display board configuration to be saved in memory, allowing for this system to scale to multiple screens all managed under one app. A device control screen gives the app simple commands to send the Wifi module on the Logic Board: Power On, Power Off, and a screen brightness slider that sends a numeric value between 5 and 100. Simple WiFi messages holding single-line commands are sent over the network using the TCP protocol. Uploading a project to the Logic Board from the application takes a little bit more work and is given its own screen. After selecting one of the projects, one of the hardware devices, and pressing "Upload", a progress bar and message will become visible on the screen tracking the conversion and transmission of each image within that project. After all the images have been sent, a message will notify the user of the upload completion.

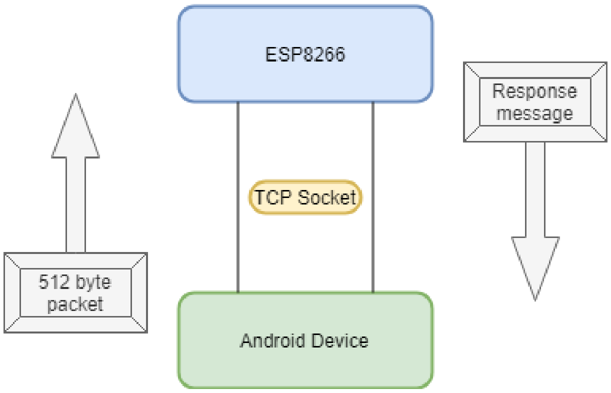
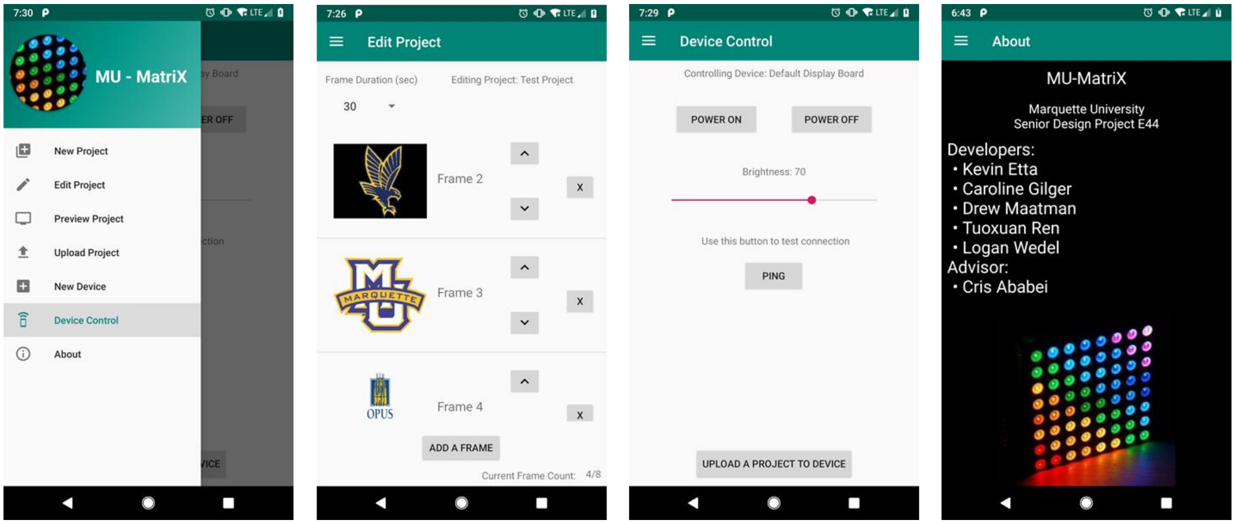


Figure 11. WiFi interface connecting the Android device to the Logic Board.

The conversion step of the project upload performs a variety of operations to turn the 320x256 bitmap into a 245,760 byte long array. This contains the hexadecimal values which represent the intensity of color for each LED in the pixel's RGB matrix. The transmission operation starts by telling the Logic board to stop multiplexing images across its screen (to allow it to focus on receiving the new data) and clear the parallel RAM chip (about to be filled with the image data). The app separates the long byte array of data into smaller chunks that it sends across a TCP connection to the ESP8266 WiFi module on a new thread. This thread sends a chunk of the image data to the ESP module, and after receiving confirmation from the microcontroller that the message arrived safely, it updates the progress bar on the main thread via a Handler. After one full image is sent, commands are sent to specify which non-volatile flash chip to store the image into. After the full project is sent, a message containing the number of images the project contains and the time duration between images is sent over WiFi again to be stored into non-volatile program flash and used during the state machine operations.



MU-MatriX

Marquette University
Senior Design Project E44

Developers:
- Kevin Etta
- Caroline Gilger
- Drew Maatman
- Tuoxuan Ren
- Logan Wedel

Advisor:
- Cris Ababei

Figure 12. WiFi interface connecting the Android device to the Logic Board.

The Android App was developed using Android Studio, meaning the code behind the application is mostly Java, xml files, and some Gradle files to define the package and build configurations. With simplicity in mind, a drawer navigation system was used with menu items each navigating to a different fragment within the main activity. A few buttons also provide navigation shortcuts that help the user follow the flow of operation. A separate activity was built to handle image selection from the image gallery and handle the image editing tools at the user's disposal. Due to the lightweight nature of the data being stored outside the images, Shared Preferences were used to store much of the project and device information that needed to be persisted. A couple of screen-shots of the app in action are shown in Figure 11.

Display Frame

An aluminum frame was fabricated using T slotted extruded 80-20 framing as the base material. Two six foot T-slotted frame bars were cut to lengths to match the four sides of the display, and mounting holes were drilled along the center of the frame to allow M3 screws to attach the frame onto the back of the outside panels of the display. Corner brackets were used to secure the frame corners together, giving the frame its rigidity. Handles were attached along the top section of the frame to allow easy handling of the project. Two support struts were added on the side frame bars to allow the display to stand upright. The frame provided substantial mechanical rigidity to the project, ensuring that LED panels would not be damaged due to mechanical stress and that the project would be able to be stood upright.

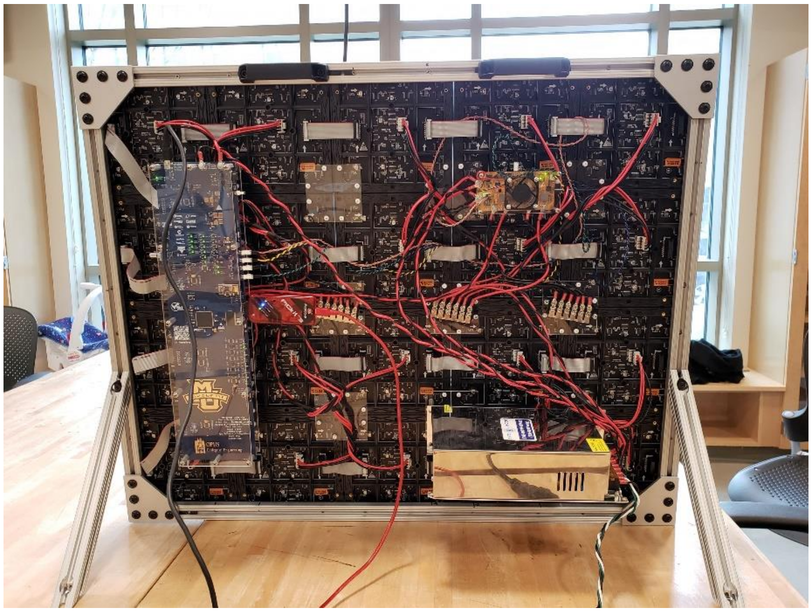


Figure 13. The final assembly of the display, logic board, power board, and power supply, all with aluminum framing.

Design Performance

Number of Images to Display

The objective of this experiment was to verify that eight images can be saved to the flash chips located on the logic board and that all images can be displayed correctly in order and repeatedly. The testing strategy was to cycle through all eight flash chips, write and then read the image data from, and drive the LED panels to display each of the images. First, all eight flash chips were written to with eight different images. Then, each image was read from the flash chips and displayed onto the panels. The images displayed on the panels were verified whether they were the desired image. This was repeated five times and all images were changed each repetition to verify the reliability of the entire process of writing and reading images into the flash chips. During this process, we monitored the error (as percentage of pixel data written or read incorrectly) during both writing to and reading from the flash chips. This test was successful with an outcome of 0% error.

Color Resolution

The objective for this experiment was to identify the optimal number of bits for the color representation used to render images. The constraints of this experiments included the amount of memory available inside the microcontroller to implement the buffer (stores the data of the image currently being displayed) as well as the time it takes to upload all eight images from the App to the board’s flash chips. The desired color representation was 12-bit color code, which would be capable of generating 4096 different colors per pixel. However, due to how the 5x4 panels (64x64 pixels each) are driven by the only 8 PWM frames (giving 3 bits per color channel) with information about pixels forming an image, we arrived at the compromise of using only a 9-bit color code, which is still capable of generating 512 different colors per pixel. This color code ensures that images have reasonable color resolution to make them appear in natural colors onto the display. While 12-bit color code would obviously be preferable, the 9-bit color code is sufficient to be able to display rich colors, yet to restrict the amount of memory required, achieve better performance (as refresh rate), and to naturally match the PWM control of the 64x64 LED panels.

Connectivity

The objective of this experiment was to verify the Android App and the logic board can communicate successfully via WiFi. The ESP8266 module, mounted on the logic board, was used for this connection. The test plan for this experiment involved sending twenty (20) images of data from the Android App to the logic board and verifying that no data was lost during the transfer. Testing to be sure that no data was lost in the transfer involved displaying the image onto the screen and verifying that the image appeared as intended. This test was successful with an outcome of 0% error.

Refresh Rate

The objective of this experiment was to verify that the microcontroller was fast enough to properly display the images without any flicker. This was performed by using an oscilloscope to measure the frequency of the clock and latch signals and seeing that the images displayed on the panels did not flicker. This procedure was repeated five times on five different images to verify that a range of different detailed images can be shown without flickering. This test was successful with an achieved refresh rate of 60Hz, which is invisible to the human eye.

Manufacturing Design

LED Panel Array

The display consists of 20 smaller LED panels arranged in a 5x4 array. The panels are attached to each other with acrylic brackets. When assembling the display, it was found that full-size display was flexible. This placed the individual subpanels at risk of damage. Additionally, the edges of the panel were exposed. Contact with a surface caused several individual LEDs to become damaged or fall out of the panels. To mitigate this damage and stiffen the large display, we constructed a frame from 80/20 extruded aluminum struts. We also added handles to lift the display, and legs to prop the display up when placed on a surface.

Power and Logic Boards

The power and logic PCBs were manufactured by JLCPCB in Shenzhen, China. Circuit schematics and PCB layouts were created using KiCad. The completed PCBs were populated by the group, each component on all boards is hand-soldered. The logic boards are attached to the display board by aluminum standoffs inserted into the acrylic brackets which connect the LED subpanels. The boards are covered by acrylic panels for protection and user safety.

Economic Analysis

Tangible Costs

<u>Product Cost</u>

Table 3. The table below is a depiction of the bill of materials for this project.

Material	**Quantity**	**Cost Per Unit**	**Calculated Cost**
LED Panels	20	$18.70	$374
Power System PCB	1	$100	$100
Power System Components	N/A	$150	$150
12V Power Supply	1	$60	$60
Logic Board PCB	1	$100	$100
Logic Board Components	N/A	$150	$150
Fabricated Aluminum Frame	1	$100	$100
Cables, Wires, & Misc.	N/A	$60	$60
XC32 Pro Compiler License	1	$30	$30
Total Cost			**$1,124.00**

The product cost is the cost of producing a one-off prototype. If the circuit boards and electronic components had been purchased in bulk, the price of each unit would drop dramatically, perhaps by greater than 50%. Electronic components and circuit boards drop dramatically in price when purchased/fabricated in bulk.

<u>Project Cost</u>

The number of hours required to complete this project during the spring semester was estimated to have been approximately 700 hours from the group members and 7 hours from the advisor. The breakdown of the development cost can be shown in the table below. A few constructs were set to estimate to this price including the 14 weeks of next semester, 10 hours per week is expected per team member, $50 per hour for each team member and $100 per hour for the advisor's time, Cris Ababei. This equates to approximately $42,000 in development cost. In addition to the development cost, the cost to build the prototype, seen in Table 6, was factored into the project cost. This makes the overall project cost $43,044.

Table 4. The table below shows the number of hours which were estimated among the team members and the overall cost of the work.

Team Member	**Hours/Week**	**Number of Weeks**	**Hourly Wage**	**Overall Cost**

Drew Maatman	10	14	$50	$7000
Tuoxuan Ren	10	14	$50	$7000
Logan Wedel	10	14	$50	$7000
Kevin Etta	10	14	$50	$7000
Caroline Gilger	10	14	$50	$7000
Cris Ababei	0.5	14	$100	$700
Total Cost				**$42,000**

Recurring Cost

The project does not require any license fees because all the platforms used to develop the hardware schematics and the Android app is on open source platforms. Therefore, the only possible recurring cost of the project will be maintenance and operation. However, maintenance on this product will be minimal and therefore, the maintenance cost is estimated to be $0 per year. To estimate the operation cost of this project, the maximum power required to display full white image onto the panel is 225W, calculated by multiplying the 5V by 45A that the panels are pulling. If it is assumed that the panel is on 24 hours all 7 days of the week, this amounts to $175 per year to keep the panels running. However, the panel will definitely not be on 24 hours per day all 7 days per week and most likely will not be powering a full white image constantly, therefore the overall yearly cost of operation will be lower than $175.

Economic Justification

Tangible Benefits

The monetary benefits for this product are minimal since its scope is only for the Electrical and Computer Engineering Department office and it is not essentially a device developed to make money for its customer. Possible non-monetary benefits from this could be students' tuition money if the students were so impressed by the LED panel that they committed to coming here based on the LED panel. On another note, if this product were to be sold to other companies as advertising modules, that would add some tangible benefits to their sales but that is not within the scope of this project. Although no quantitative values can be applied to the tangible benefits of this product, it has been proven that the LED display will return the customer with added value based on advertising and information shared to all passersby.

Risk Analysis

The table below describes the risk to the timely completion of the project and the preventative actions we will take to avoid these risks as best possible, which were determined about halfway through the completion of the project.

Table 5 – Project Risk Analysis

Project Risk	Preventative Action	Severity	Probability	Risk Calculation
Component failure / Part performance	Order extra parts that are known to have a higher failure rate	10	0.3	3.3
Design shortcomings	Schedule extra time to accommodate difficulties, incremental feature testing, form backup plans	8	0.4	3.2
Misguided/Lack of effort	Meet regularly, keep everyone on task	4	.5	2
Illness	Sharing responsibilities, strict handwashing policy	1	.8	.8
Bottlenecks in design, build or test phase	Parallelizing as many tasks as possible, spreading tasks out which are tolerant for reordering	5	.8	4
HardwareX rejects article	Read HardwareX to determine their publishing standards, contact a representative of HardwareX prior to submittal	6	.1	.6
Hardware damage due to errors in firmware	Make incremental changes and test frequently; have replacement components in case of damage	5	.4	2
Physical damage to panels from moving or frame assembly	Be extremely careful in handling hardware and frame	6	.2	1.2
Difficulties integrating WiFi functionality	Schedule extra time for Wifi integration, keep WiFi integration off the critical path	4	.5	2

Risky situations that ended up becoming issues were physical damage to LED panels during project development. This was because the LEDs soldered onto the purchased LED panels were very fragile, and not enough care was taken to ensure they were ripped off of the LED panels during project development. Extra care should have been taken. Around 10 of the 81,000 pixels were ripped off of the entire display during project development, mostly along the edges of the display. Adding the frame to the project drastically reduced the damage to the project afterward.

Another situation which arose was hardware failure of components on the power board. We suspect that this was due to poor handling of the project and/or an ESD strike on the component which failed, since the operating parameters of the power board were well understood and the circuitry was heavily ruggedized to ensure damage would not occur due to electrical stresses.

HardwareX did not accept our journal paper, citing that the scope of the project did not fit the purpose of the journal. The project mentor, Cris Ababei, stated that another journal would be sought for publication of the journal paper that was written about project development.

WiFi integration into the system was a lot more difficult than first assumed. In order to integrate WiFi functionality and Android App communication into the project, many compromises in download speed and functionality had to be made. This was due to the serial interface of the ESP8266 WiFi module as used out of the box. The ESP was meant to be used over a serial port with a user typing in AT commands in a slow manner. Trying to automate this with a 252 MHz microcontroller ended up being more challenging. By the end of project development, the fastest image download time achieved was 1.5 minutes, which seemed slower than anticipated. This was a compromise that had to be made due to project deadlines and hardware selected.

Finally, microcontroller performance ended up being a risk that was mitigated through purchasing a pro C compiler license in order to highly optimize MCU code execution at runtime. Without purchasing this higher optimization level, images could only be drawn on the screen with a visible flicker. After using the optimization features of the compiler, microcontroller code executed so quickly that the logic board could not physically support the speed of the I/O signals to drive the LED panels. After fine tuning the signal timing and dialing back optimization settings, images could be drawn on the screen without any visible flicker, and the PCB could support these high speed signals.

Project Legacy

Overall, there were many lessons learned throughout this process. The microcontroller used in this design, though one of the most powerful PIC microcontrollers on the market, was still very limited in speed bandwidth and data throughput. In the future, it would be possible to look into using an FPGA or possibly a full SoC to implement in the design. However, this would require a completely new hardware schematic and PCB along with full VHDL code to run on the FPGA. Perhaps by using this different design concept, it would be possible to provide video capabilities on the display.

Another possible addition to the design would be to implement an authentication PIN code within the Android App to be able to secure the device with a target phone. This way, only users with the proper PIN code can access the device so that strangers who download the app from the Google Play store will not put up their own images onto the display. Another possible future improvement to the app would be to allow simple text to be placed across the display. We hesitated to implement this initially because we did not want the Android app to turn into an image editing/creating software. It serves more as a tool to place ready-to-go images onto the display board.

Although there were various features or design concepts which we could have been done differently, the overall product was successful. The full design was able to provide all of the required target specifications listed at the beginning of the year. The power board was able to supply a safe and regulated 5V supply to the panels as well as a 3.3V supply to the microcontroller. The logic board was able to shift in the data to the panels fast enough so that the display did not flicker. The display has high pixel resolution and color resolution and can be fully controlled by the app. These were the most important aspects of the design which were required and they were all successfully implemented in the project.

Appendix I: Hardware Schematics

Electronic Display Logic Board

Marquette Universiy Senior Design 2018, Group E44

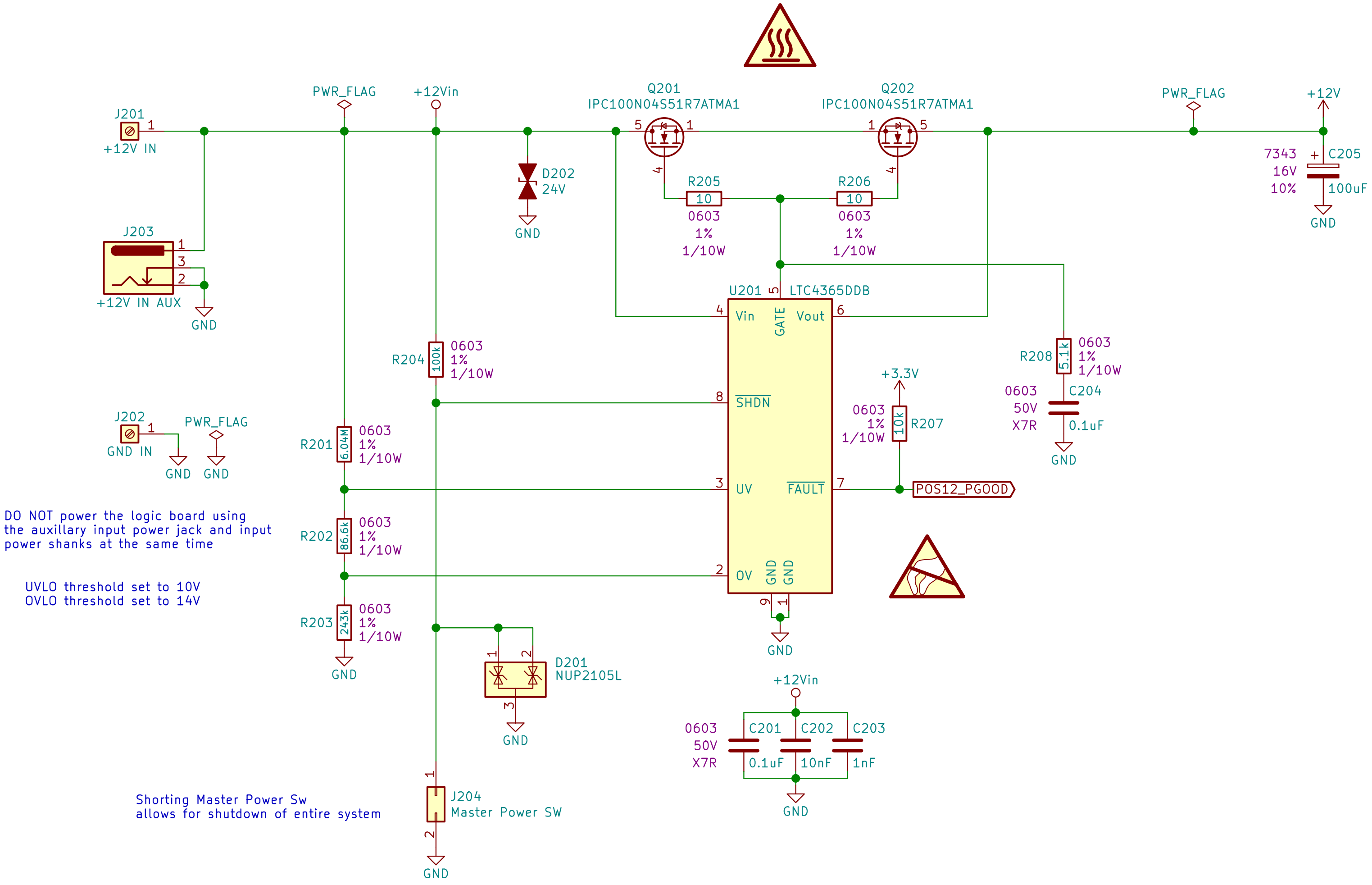
Drew Maatman, Kevin Etta, Logan Wedel, Caroline Gilger, Tuoxuan Ren

01. Table of Contents

Note: If component footprints, tolerances, and power ratings are hidden, components are:
0603 case size, 1% tolerance, 1/10W power rating

Marquette University Senior Design 2018/2019 Group E44		
Sheet: /		
File: LED_Display_Controller.sch		
Title: Electronic Display Logic Board		
Size: A	Date: 2018-12-15	Rev: A
KiCad E.D.A. kicad (5.1.0)-1		Id: 1/32

02. Power Input



DO NOT power the logic board using
the auxillary input power jack and input
power shanks at the same time

UVLO threshold set to 10V
OVLO threshold set to 14V

Shorting Master Power Sw
allows for shutdown of entire system

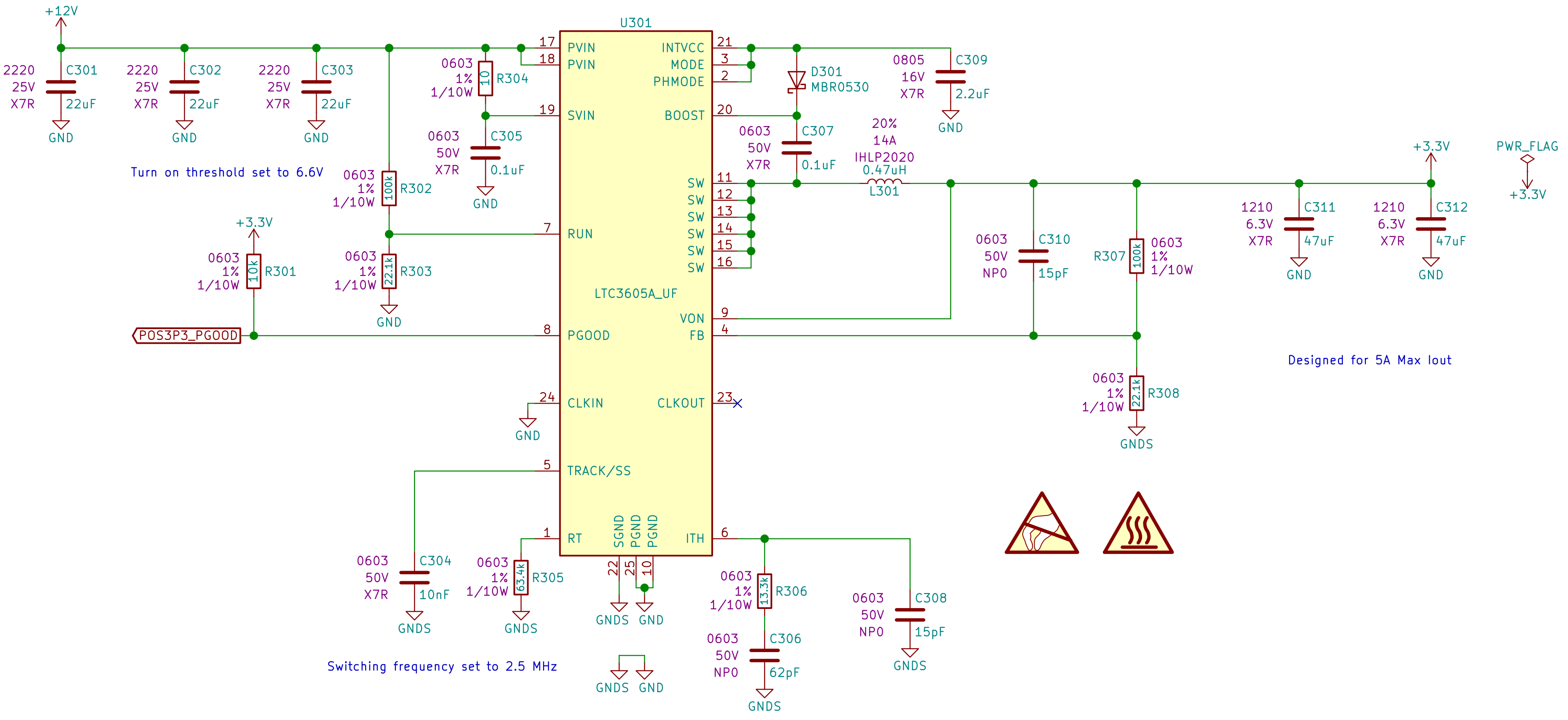
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Power Input/
File: Power_Input.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 2/32

03. +3.3V Power Supply



Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS3P3 Power Supply/

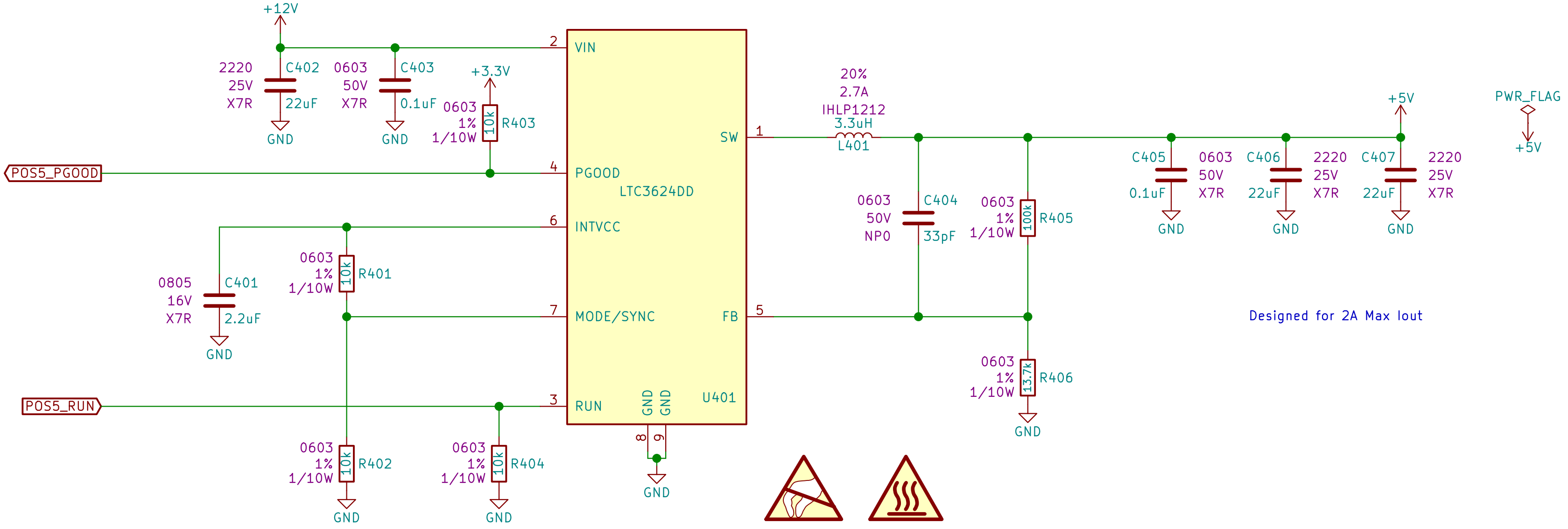
File: POS3P3_Power_Supply.sch

Title: Electronic Display Logic Board

Size: A Date: 2018-12-15 **Rev: A**

KiCad E.D.A. kicad (5.1.0)-1 Id: 3/32

04. +5V Power Supply



Designed for 2A Max Iout

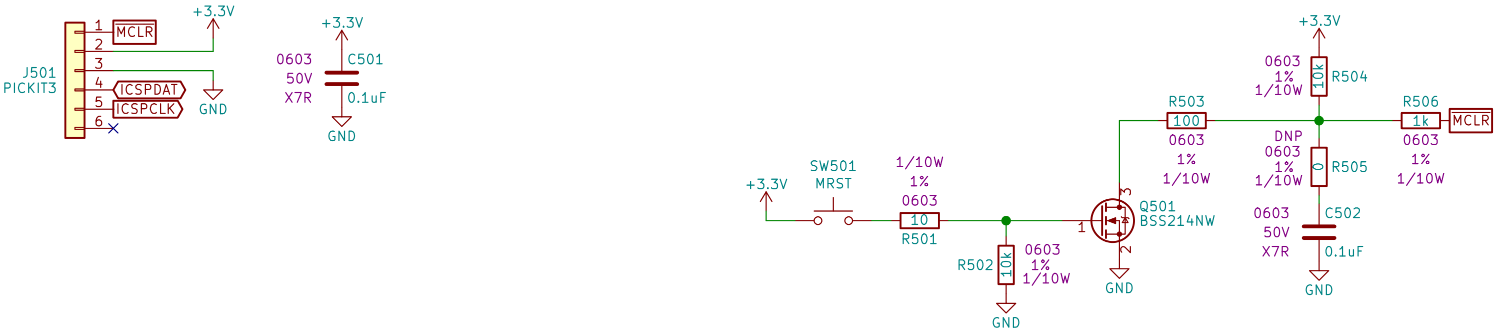
Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS5 Power Supply/
File: POS5_Power_Supply.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 4/32

05. Microcontroller Programming



Marquette University Senior Design 2018/2019 Group E44

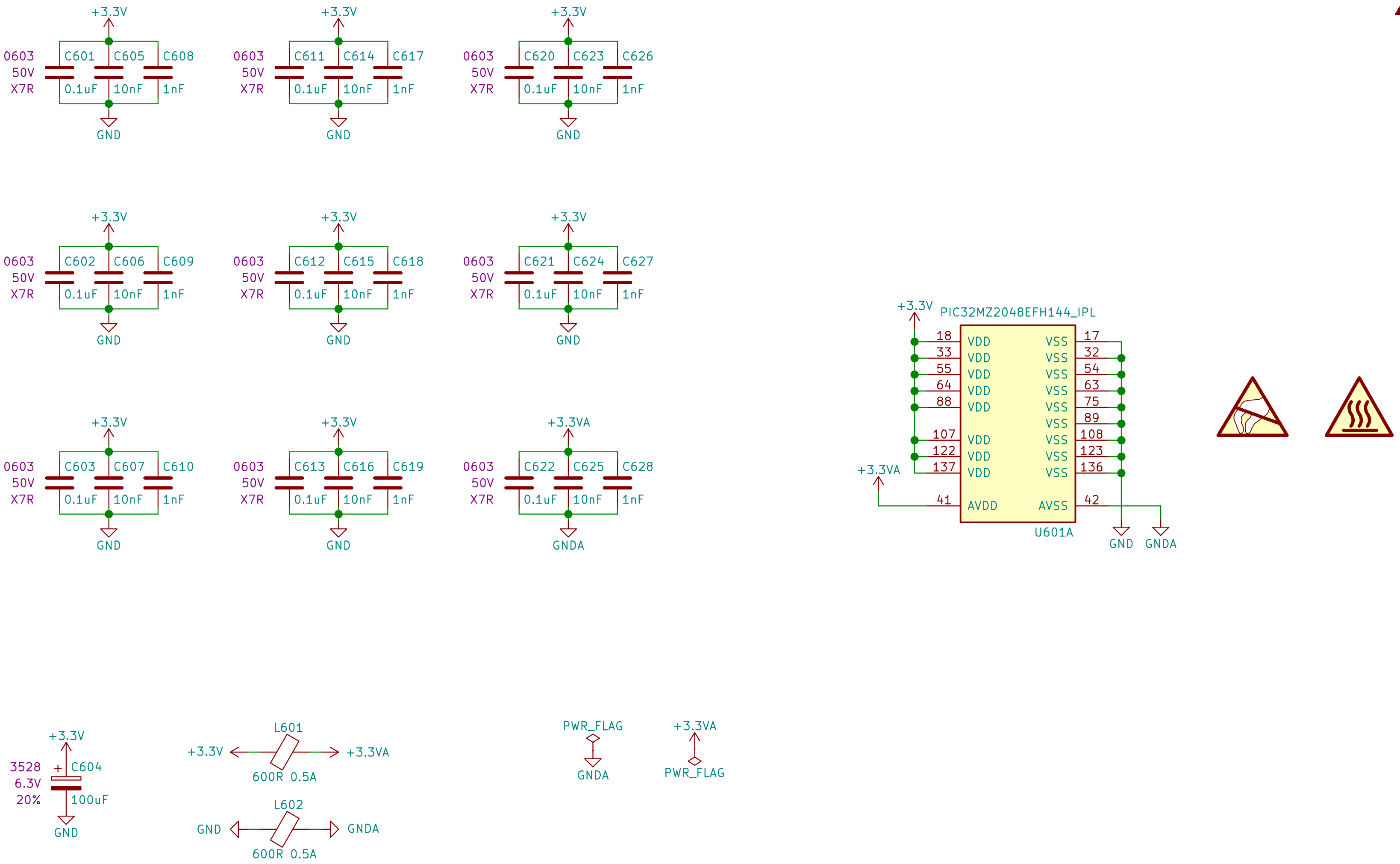
Sheet: /Microcontroller Programming/
File: Microcontroller_Programming.sch

Title: Electronic Display Logic Board

Size: A Date: 2018-12-15 **Rev: A**

KiCad E.D.A. kicad (5.1.0)-1 Id: 5/32

06. Microcontroller Power



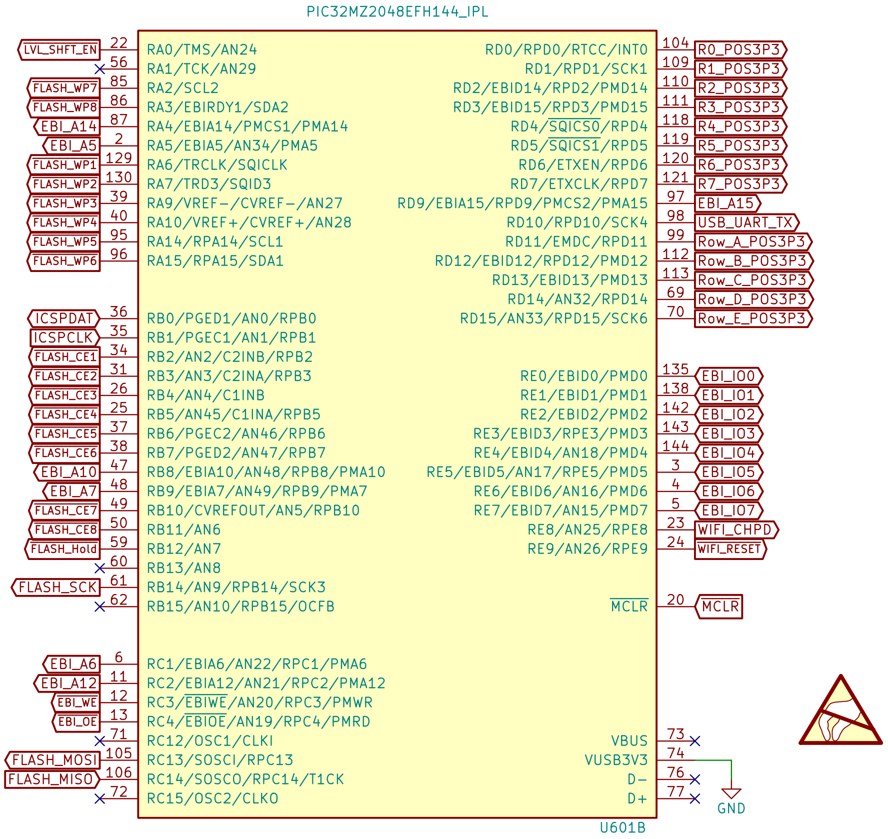
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Microcontroller Power/
File: Microcontroller_Power.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 6/32

07. Microcontroller IO Bank 1


PIC32MZ2048EFH144_IPL
LVL_SHFT_EN 22 RA0/TMS/AN24
56 RA1/TCK/AN29
FLASH_WP7 85 RA2/SCL2
FLASH_WP8 86 RA3/EBIRDY1/SDA2
EBI_A14 87 RA4/EBIA14/PMCS1/PMA14
EBI_A5 2 RA5/EBIA5/AN34/PMA5
FLASH_WP1 129 RA6/TRCLK/SQICLK
FLASH_WP2 130 RA7/TRD3/SQID3
FLASH_WP3 39 RA9/VREF−/CVREF−/AN27
FLASH_WP4 40 RA10/VREF+/CVREF+/AN28
FLASH_WP5 95 RA14/RPA14/SCL1
FLASH_WP6 96 RA15/RPA15/SDA1
ICSPDAT 36 RB0/PGED1/AN0/RPB0
ICSPCLK 35 RB1/PGEC1/AN1/RPB1
FLASH_CE1 34 RB2/AN2/C2INB/RPB2
FLASH_CE2 31 RB3/AN3/C2INA/RPB3
FLASH_CE3 26 RB4/AN4/C1INB
FLASH_CE4 25 RB5/AN45/C1INA/RPB5
FLASH_CE5 37 RB6/PGEC2/AN46/RPB6
FLASH_CE6 38 RB7/PGED2/AN47/RPB7
EBI_A10 47 RB8/EBIA10/AN48/RPB8/PMA10
EBI_A7 48 RB9/EBIA7/AN49/RPB9/PMA7
FLASH_CE7 49 RB10/CVREFOUT/AN5/RPB10
FLASH_CE8 50 RB11/AN6
FLASH_Hold 59 RB12/AN7
60 RB13/AN8
FLASH_SCK 61 RB14/AN9/RPB14/SCK3
62 RB15/AN10/RPB15/OCFB
EBI_A6 6 RC1/EBIA6/AN22/RPC1/PMA6
EBI_A12 11 RC2/EBIA12/AN21/RPC2/PMA12
EBI_WE 12 RC3/EBIWE/AN20/RPC3/PMWR
EBI_OE 13 RC4/EBIOE/AN19/RPC4/PMRD
71 RC12/OSC1/CLKI
FLASH_MOSI 105 RC13/SOSCI/RPC13
FLASH_MISO 106 RC14/SOSCO/RPC14/T1CK
72 RC15/OSC2/CLKO
RD0/RPD0/RTCC/INT0 104 R0_POS3P3
RD1/RPD1/SCK1 109 R1_POS3P3
RD2/EBID14/RPD2/PMD14 110 R2_POS3P3
RD3/EBID15/RPD3/PMD15 111 R3_POS3P3
RD4/SQICS0/RPD4 118 R4_POS3P3
RD5/SQICS1/RPD5 119 R5_POS3P3
RD6/ETXEN/RPD6 120 R6_POS3P3
RD7/ETXCLK/RPD7 121 R7_POS3P3
RD9/EBIA15/RPD9/PMCS2/PMA15 97 EBI_A15
RD10/RPD10/SCK4 98 USB_UART_TX
RD11/EMDC/RPD11 99 Row_A_POS3P3
RD12/EBID12/RPD12/PMD12 112 Row_B_POS3P3
RD13/EBID13/PMD13 113 Row_C_POS3P3
RD14/AN32/RPD14 69 Row_D_POS3P3
RD15/AN33/RPD15/SCK6 70 Row_E_POS3P3
RE0/EBID0/PMD0 135 EBI_IO0
RE1/EBID1/PMD1 138 EBI_IO1
RE2/EBID2/PMD2 142 EBI_IO2
RE3/EBID3/RPE3/PMD3 143 EBI_IO3
RE4/EBID4/AN18/PMD4 144 EBI_IO4
RE5/EBID5/AN17/RPE5/PMD5 3 EBI_IO5
RE6/EBID6/AN16/PMD6 4 EBI_IO6
RE7/EBID7/AN15/PMD7 5 EBI_IO7
RE8/AN25/RPE8 23 WIFI_CHPD
RE9/AN26/RPE9 24 WIFI_RESET
MCLR 20 MCLR
VBUS 73
VUSB3V3 74
D− 76
D+ 77
GND
U601B

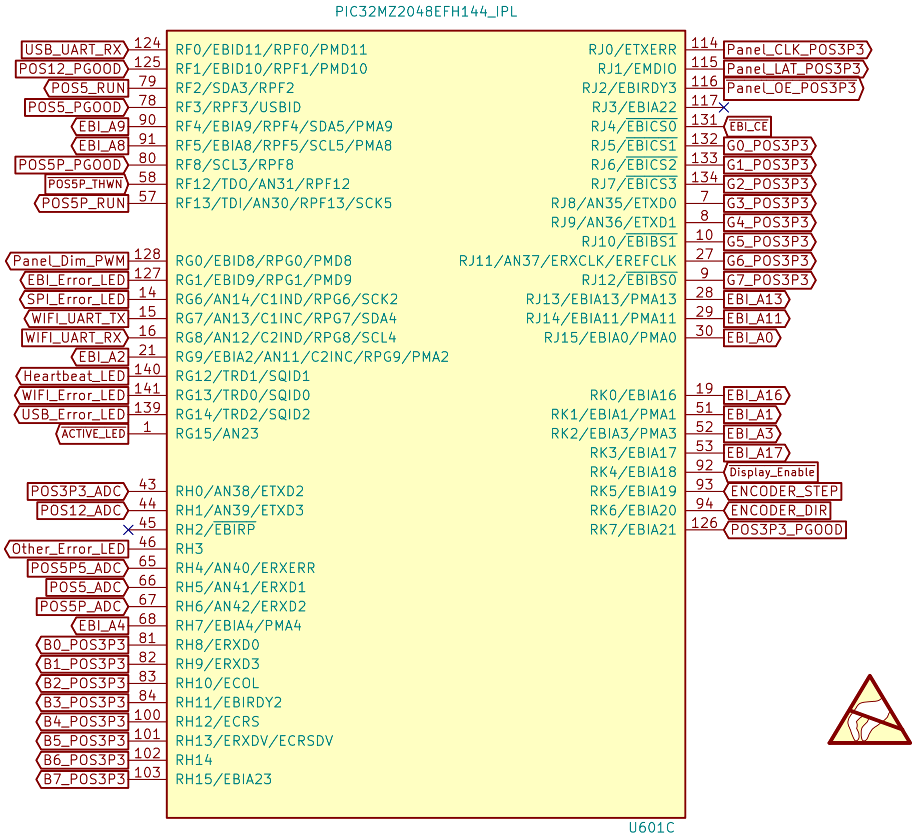
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Microcontroller 1/
File: Microcontroller_1.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 7/32

08. Microcontroller IO Bank 2


PIC32MZ2048EFH144_IPL
U601C
USB_UART_RX 124 RF0/EBID11/RPF0/PMD11
POS12_PGOOD 125 RF1/EBID10/RPF1/PMD10
POS5_RUN 79 RF2/SDA3/RPF2
POS5_PGOOD 78 RF3/RPF3/USBID
EBI_A9 90 RF4/EBIA9/RPF4/SDA5/PMA9
EBI_A8 91 RF5/EBIA8/RPF5/SCL5/PMA8
POS5P_PGOOD 80 RF8/SCL3/RPF8
POS5P_THWN 58 RF12/TDO/AN31/RPF12
POS5P_RUN 57 RF13/TDI/AN30/RPF13/SCK5
Panel_Dim_PWM 128 RG0/EBID8/RPG0/PMD8
EBI_Error_LED 127 RG1/EBID9/RPG1/PMD9
SPI_Error_LED 14 RG6/AN14/C1IND/RPG6/SCK2
WIFI_UART_TX 15 RG7/AN13/C1INC/RPG7/SDA4
WIFI_UART_RX 16 RG8/AN12/C2IND/RPG8/SCL4
EBI_A2 21 RG9/EBIA2/AN11/C2INC/RPG9/PMA2
Heartbeat_LED 140 RG12/TRD1/SQID1
WIFI_Error_LED 141 RG13/TRD0/SQID0
USB_Error_LED 139 RG14/TRD2/SQID2
ACTIVE_LED 1 RG15/AN23
POS3P3_ADC 43 RH0/AN38/ETXD2
POS12_ADC 44 RH1/AN39/ETXD3
45 RH2/EBIRP
Other_Error_LED 46 RH3
POS5P5_ADC 65 RH4/AN40/ERXERR
POS5_ADC 66 RH5/AN41/ERXD1
POS5P_ADC 67 RH6/AN42/ERXD2
EBI_A4 68 RH7/EBIA4/PMA4
B0_POS3P3 81 RH8/ERXD0
B1_POS3P3 82 RH9/ERXD3
B2_POS3P3 83 RH10/ECOL
B3_POS3P3 84 RH11/EBIRDY2
B4_POS3P3 100 RH12/ECRS
B5_POS3P3 101 RH13/ERXDV/ECRSDV
B6_POS3P3 102 RH14
B7_POS3P3 103 RH15/EBIA23
RJ0/ETXERR 114 Panel_CLK_POS3P3
RJ1/EMDIO 115 Panel_LAT_POS3P3
RJ2/EBIRDY3 116 Panel_OE_POS3P3
RJ3/EBIA22 117
RJ4/EBICS0 131 EBI_CE
RJ5/EBICS1 132 G0_POS3P3
RJ6/EBICS2 133 G1_POS3P3
RJ7/EBICS3 134 G2_POS3P3
RJ8/AN35/ETXD0 7 G3_POS3P3
RJ9/AN36/ETXD1 8 G4_POS3P3
RJ10/EBIBS1 10 G5_POS3P3
RJ11/AN37/ERXCLK/EREFCLK 27 G6_POS3P3
RJ12/EBIBS0 9 G7_POS3P3
RJ13/EBIA13/PMA13 28 EBI_A13
RJ14/EBIA11/PMA11 29 EBI_A11
RJ15/EBIA0/PMA0 30 EBI_A0
RK0/EBIA16 19 EBI_A16
RK1/EBIA1/PMA1 51 EBI_A1
RK2/EBIA3/PMA3 52 EBI_A3
RK3/EBIA17 53 EBI_A17
RK4/EBIA18 92 Display_Enable
RK5/EBIA19 93 ENCODER_STEP
RK6/EBIA20 94 ENCODER_DIR
RK7/EBIA21 126 POS3P3_PGOOD

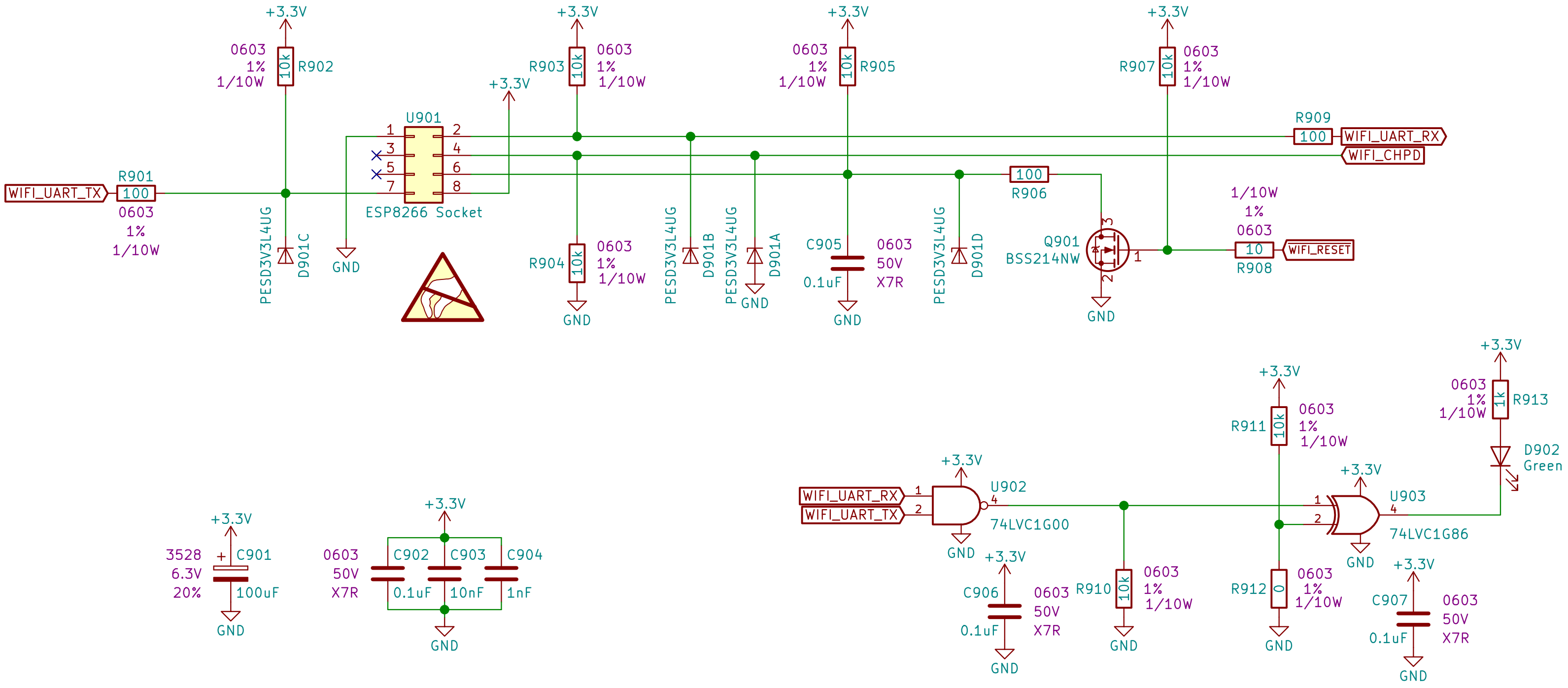
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Microcontroller 2/
File: Microcontroller_2.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 8/32

09. WiFi Module



Marquette University Senior Design 2018/2019 Group E44

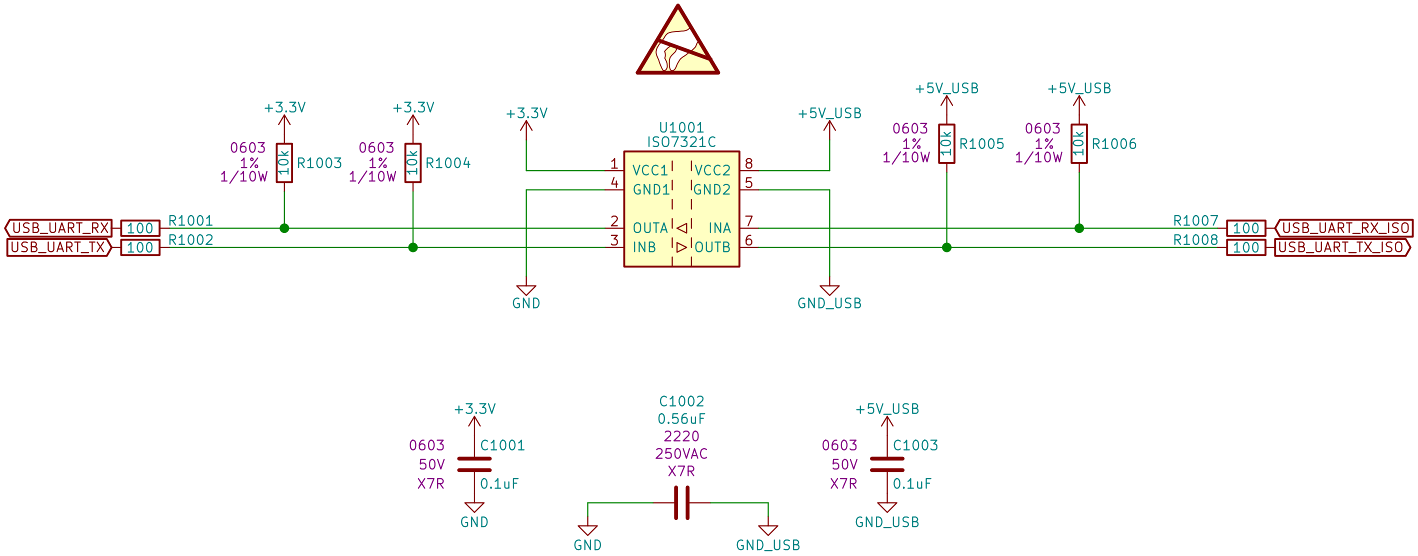
Sheet: /WiFi Module/

File: Wi_Fi_Module.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 9/32

10. USB UART Digital Isolation



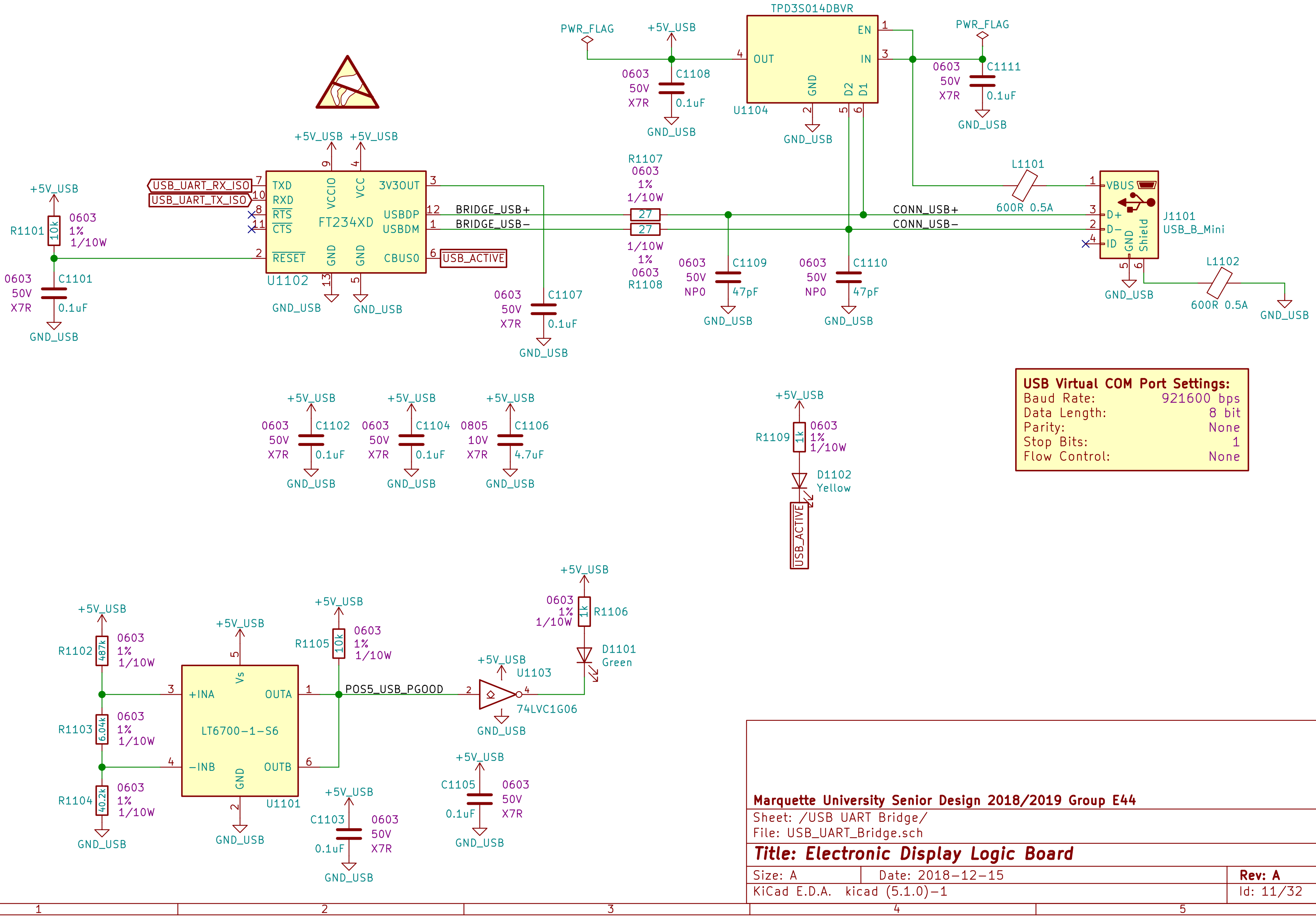
Marquette University Senior Design 2018/2019 Group E44

Sheet: /USB UART Isolation/
File: USB_UART_Isolation.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 10/32

11. USB UART Bridge



USB Virtual COM Port Settings:

Baud Rate:	921600 bps
Data Length:	8 bit
Parity:	None
Stop Bits:	1
Flow Control:	None

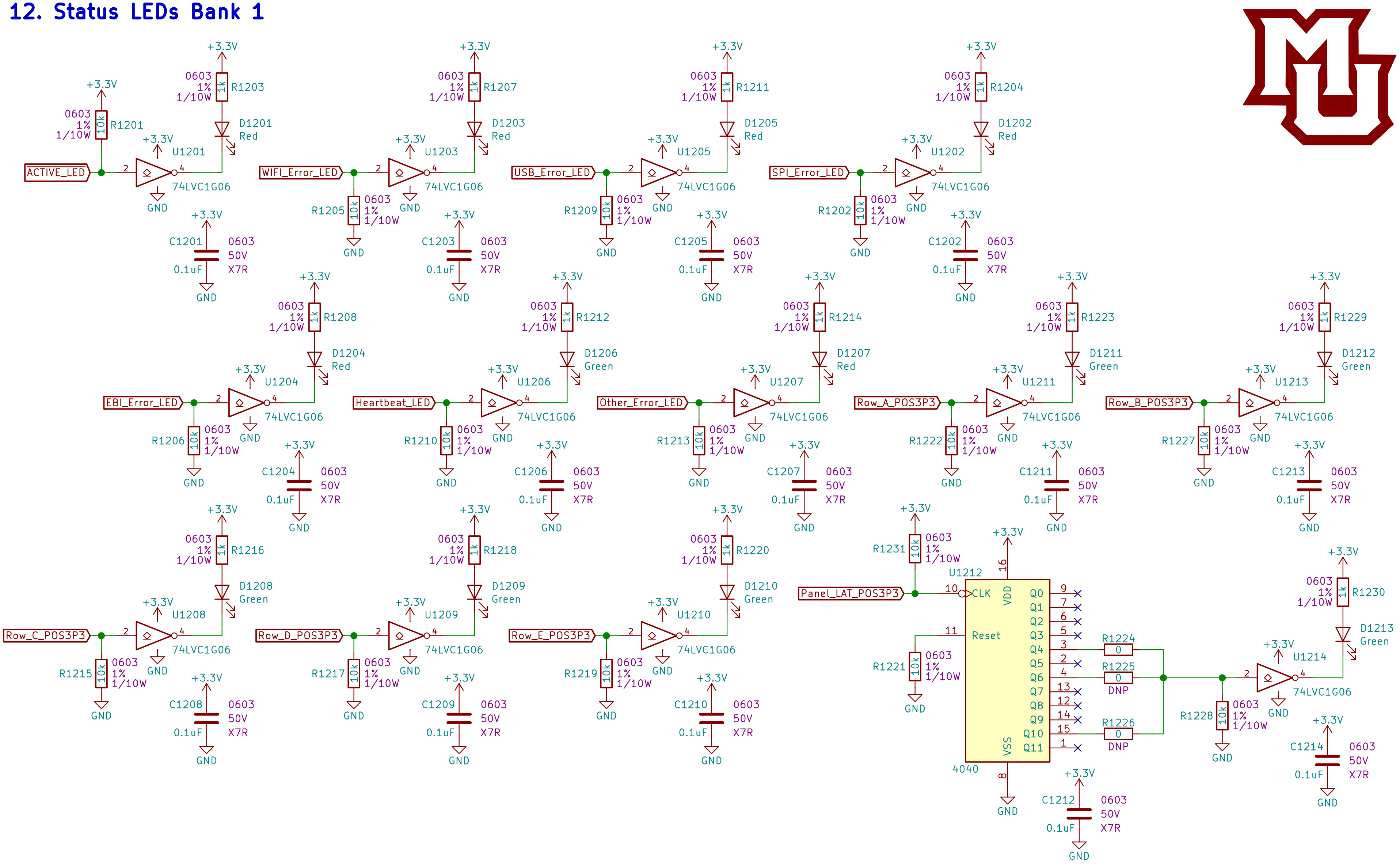
Marquette University Senior Design 2018/2019 Group E44

Sheet: /USB UART Bridge/
File: USB_UART_Bridge.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 11/32

12. Status LEDs Bank 1



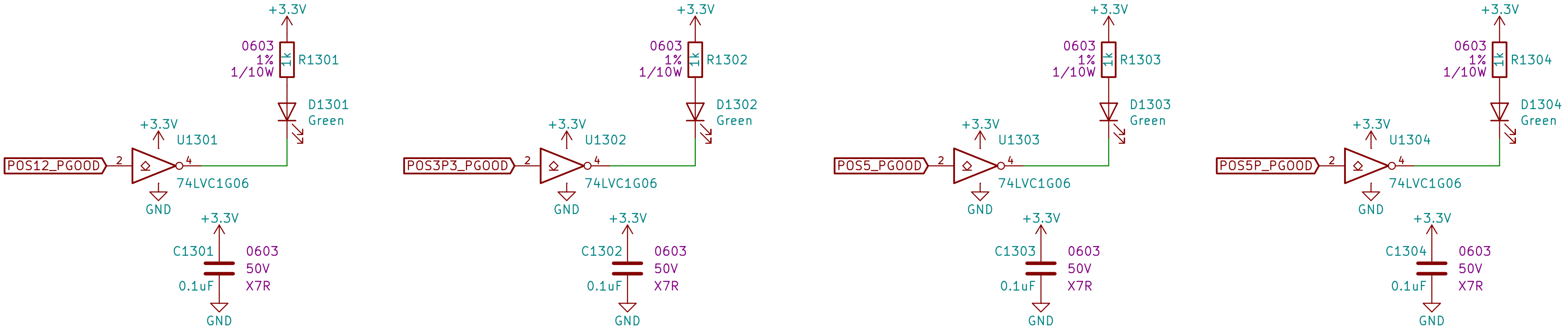
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Status LEDs 1/
File: Status_LEDs_1.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 12/32

13. Status LEDs Bank 2



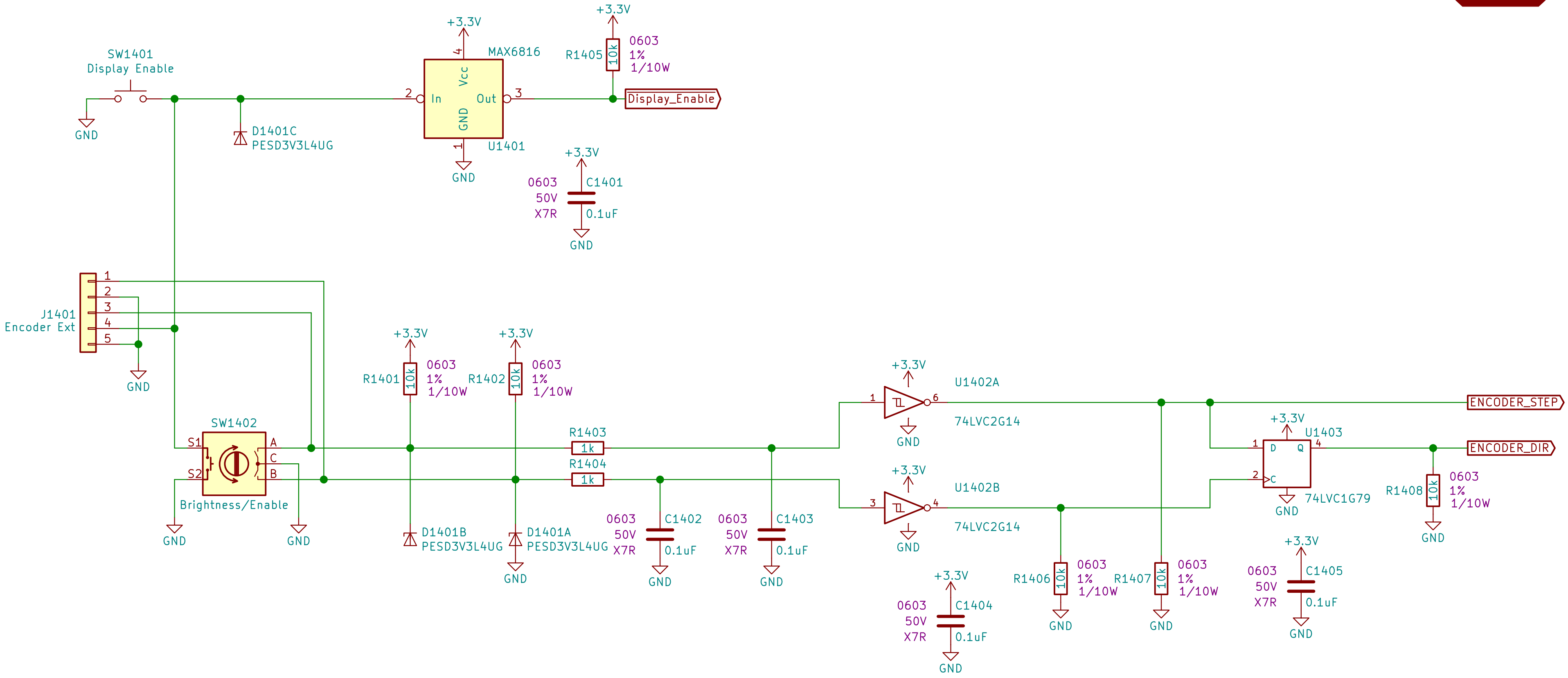
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Status LEDs 2/
File: Status_LEDs_2.sch

Title: Electronic Display Logic Board

Size: A Date: 2018-12-15 **Rev: A**
KiCad E.D.A. kicad (5.1.0)-1 Id: 13/32

14. Pushbuttons



Turning rotary encoder will adjust display master brightness
Pressing encoder switch will toggle display on/off state

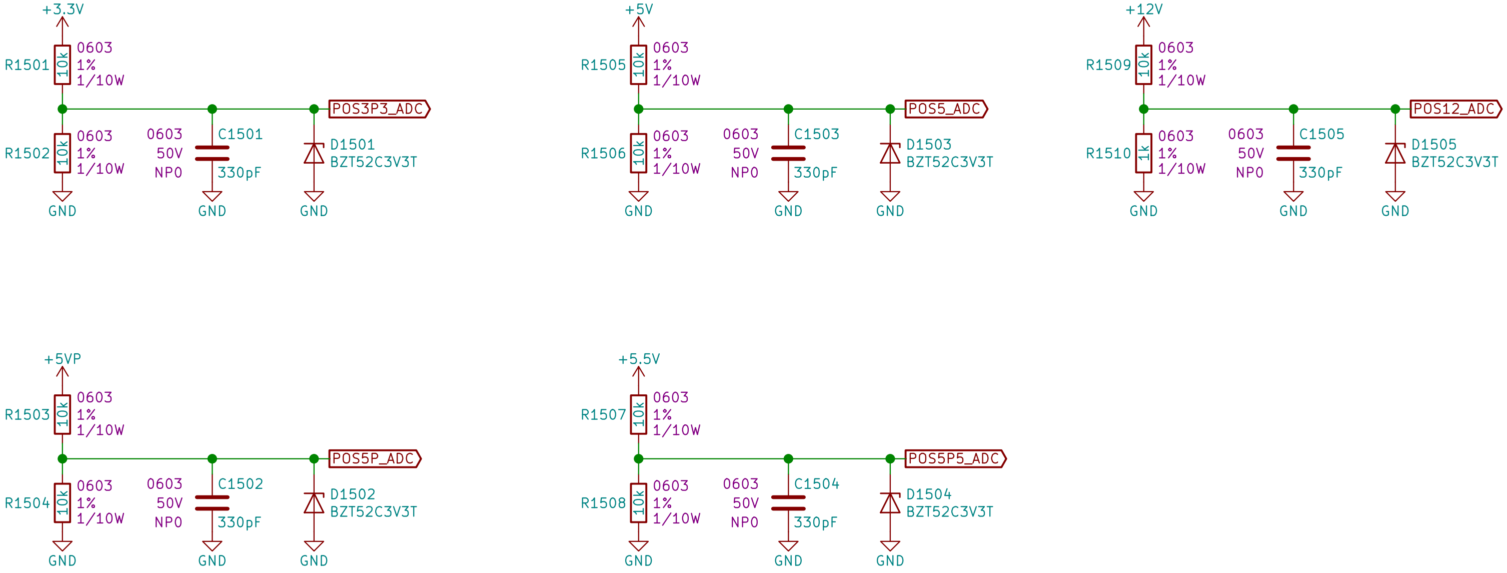
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Pushbuttons/
File: Pushbuttons.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 14/32

15. Internal Rail Monitoring



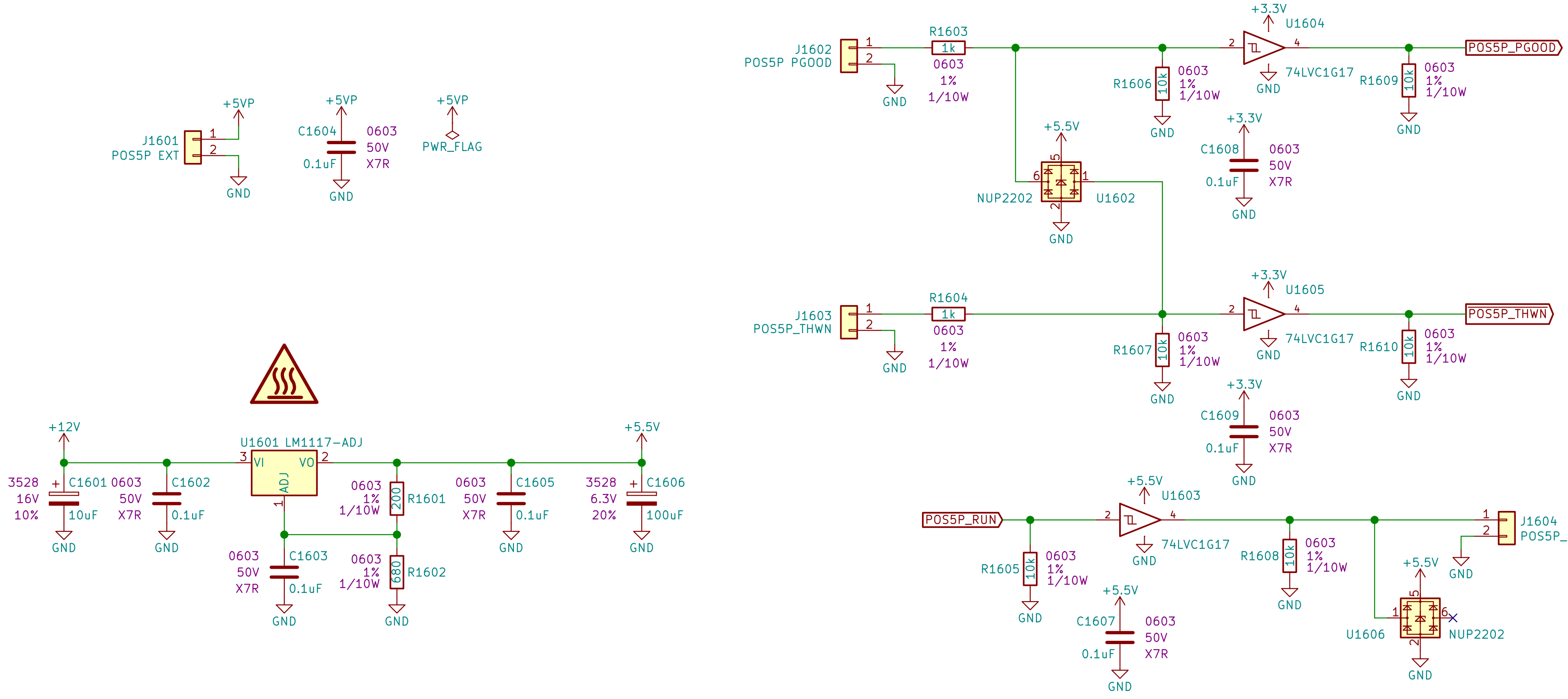
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Internal Rail Monitoring/
File: Internal_Rail_Monitoring.sch

Title: Electronic Display Logic Board

Size: A Date: 2018-12-15 **Rev: A**
KiCad E.D.A. kicad (5.1.0)-1 Id: 15/32

16. LED Power Supply Monitoring



All headers on this sheet interface with power board to power panels

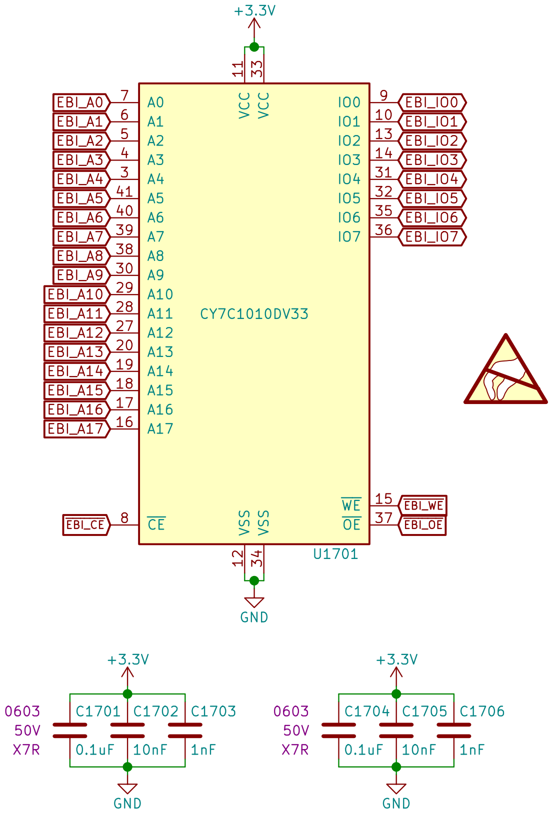
Marquette University Senior Design 2018/2019 Group E44

Sheet: /LED POS5 Monitoring/
File: LED_POS5_Monitoring.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 16/32

17. External SRAM



External SRAM stores the current slide being displayed
Each of the eight programmed slides are stored in external serial flash

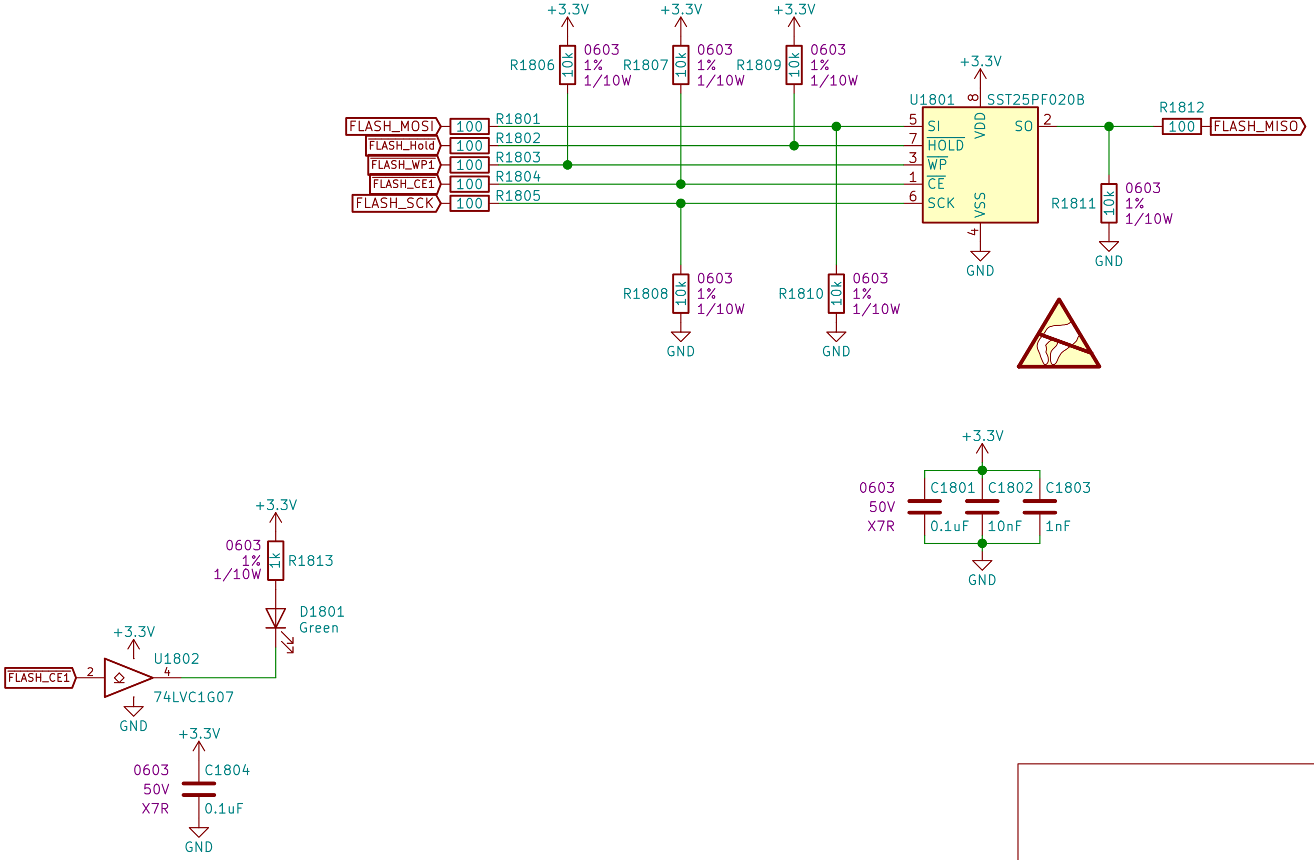
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External SRAM/
File: External_SRAM.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 17/32

18. External FLASH 1



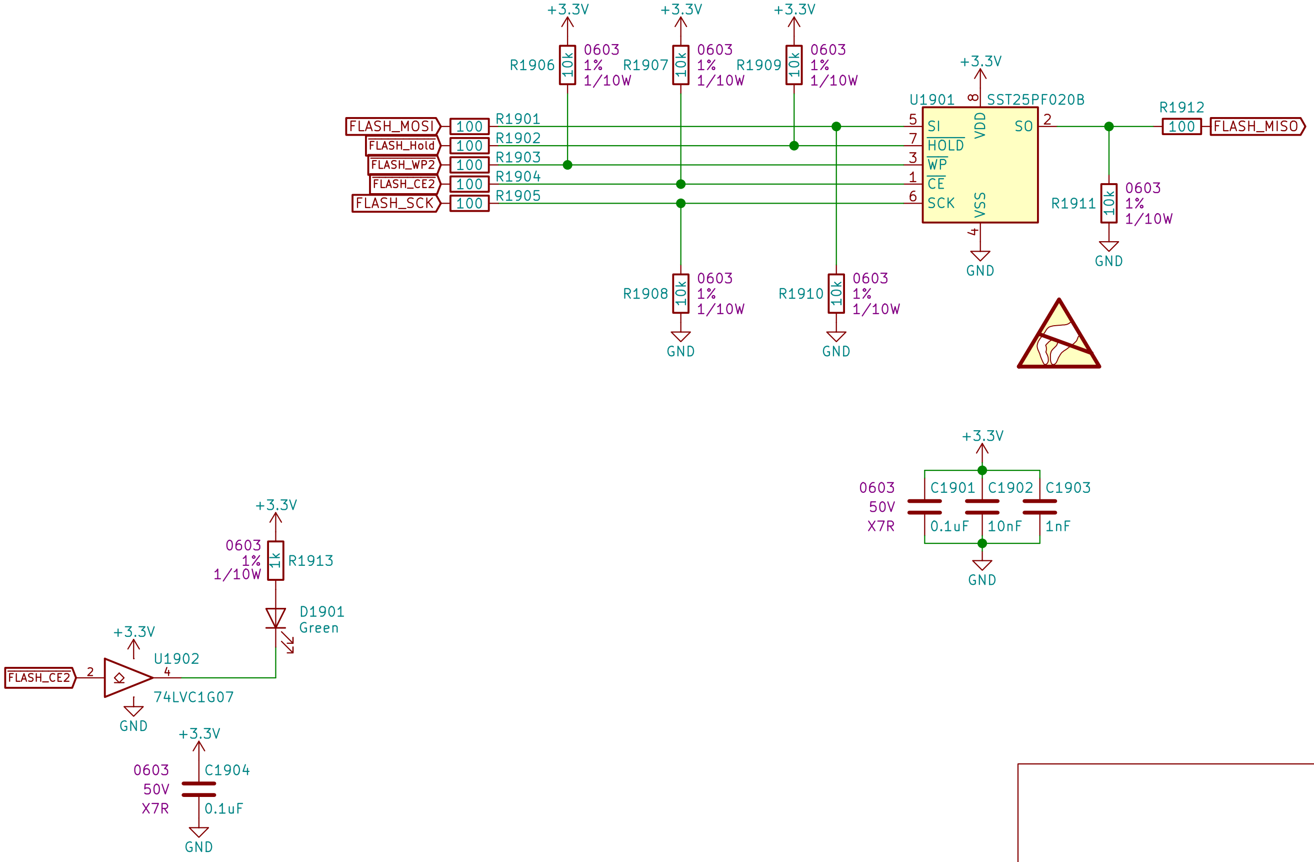
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 1/
File: External_Flash_1.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	Rev: A
KiCad E.D.A. kicad (5.1.0)-1		Id: 18/32

19. External FLASH 2



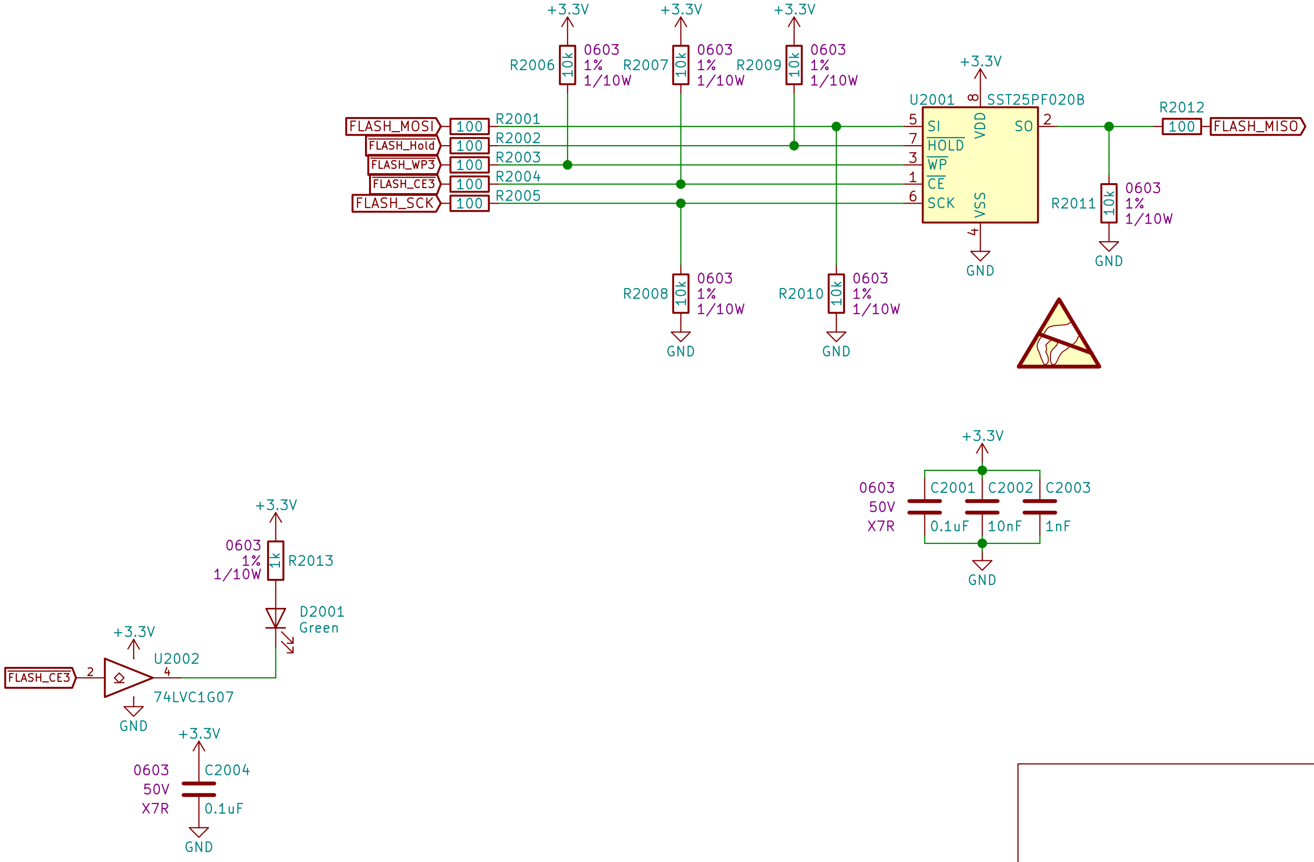
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 2/
File: External_Flash_2.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	Rev: A
KiCad E.D.A. kicad (5.1.0)-1		Id: 19/32

20. External FLASH 3



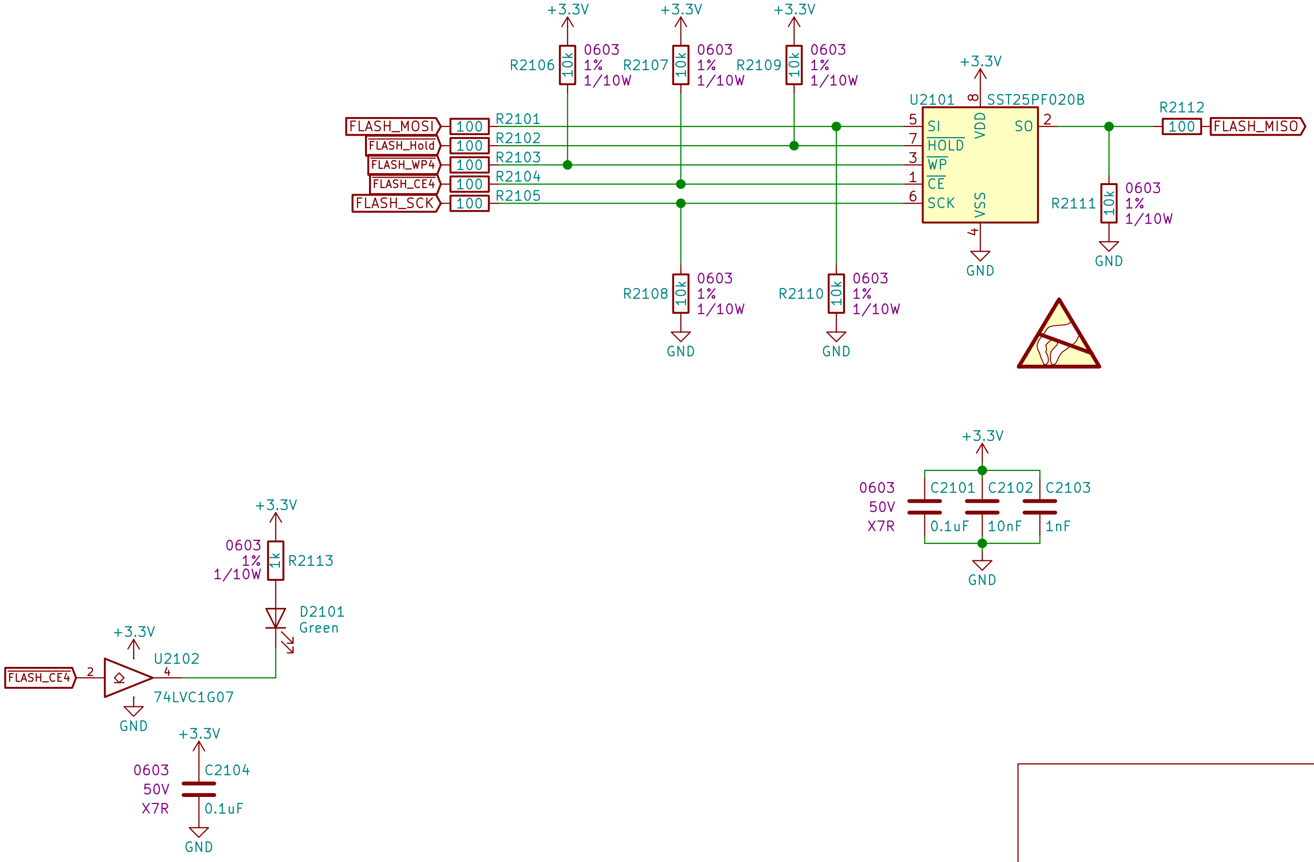
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 3/
File: External_Flash_3.sch

Title: Electronic Display Logic Board

Size: A Date: 2018-12-15 **Rev: A**
KiCad E.D.A. kicad (5.1.0)-1 Id: 20/32

21. External FLASH 4



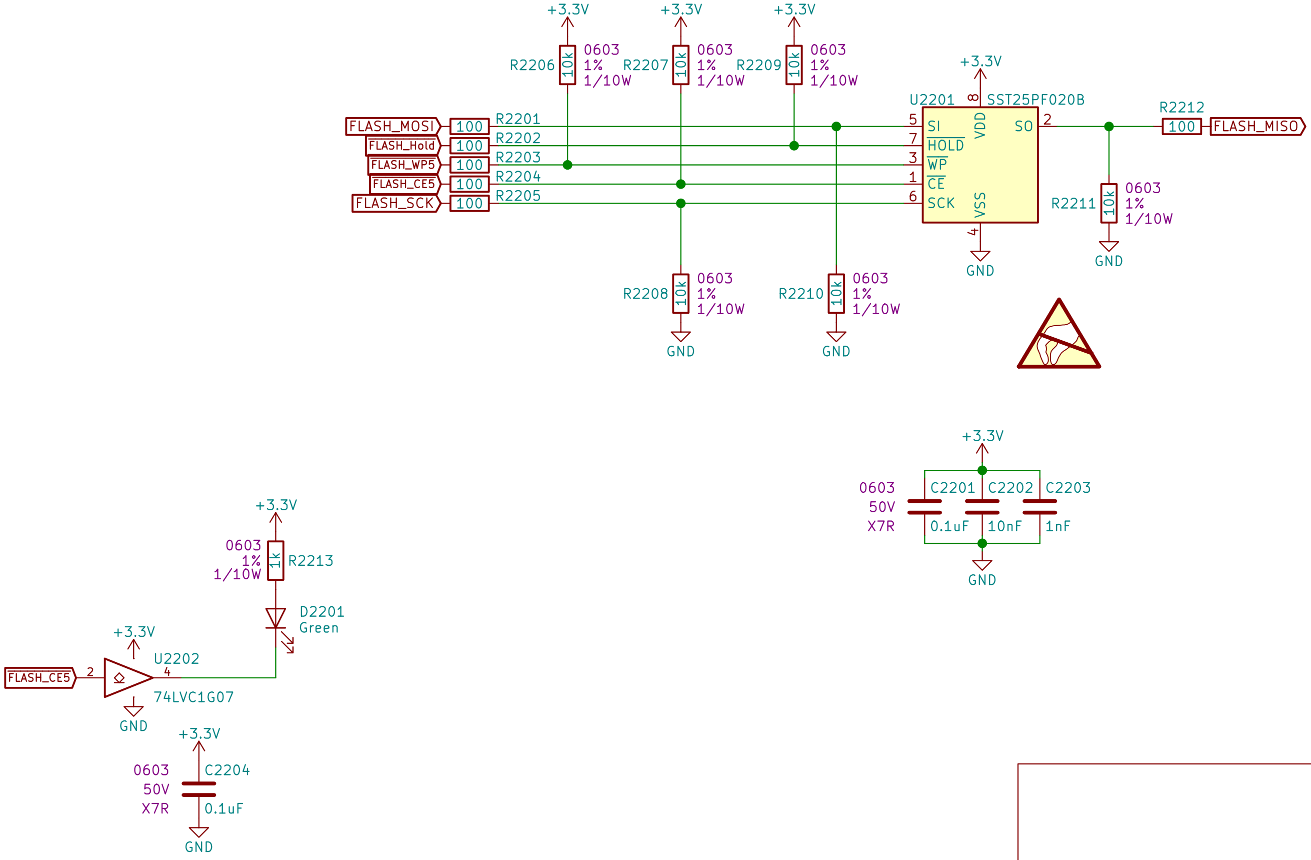
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 4/
File: External_Flash_4.sch

Title: Electronic Display Logic Board

Size: A   Date: 2018-12-15   **Rev: A**
KiCad E.D.A. kicad (5.1.0)-1   Id: 21/32

22. External FLASH 5



Marquette University Senior Design 2018/2019 Group E44

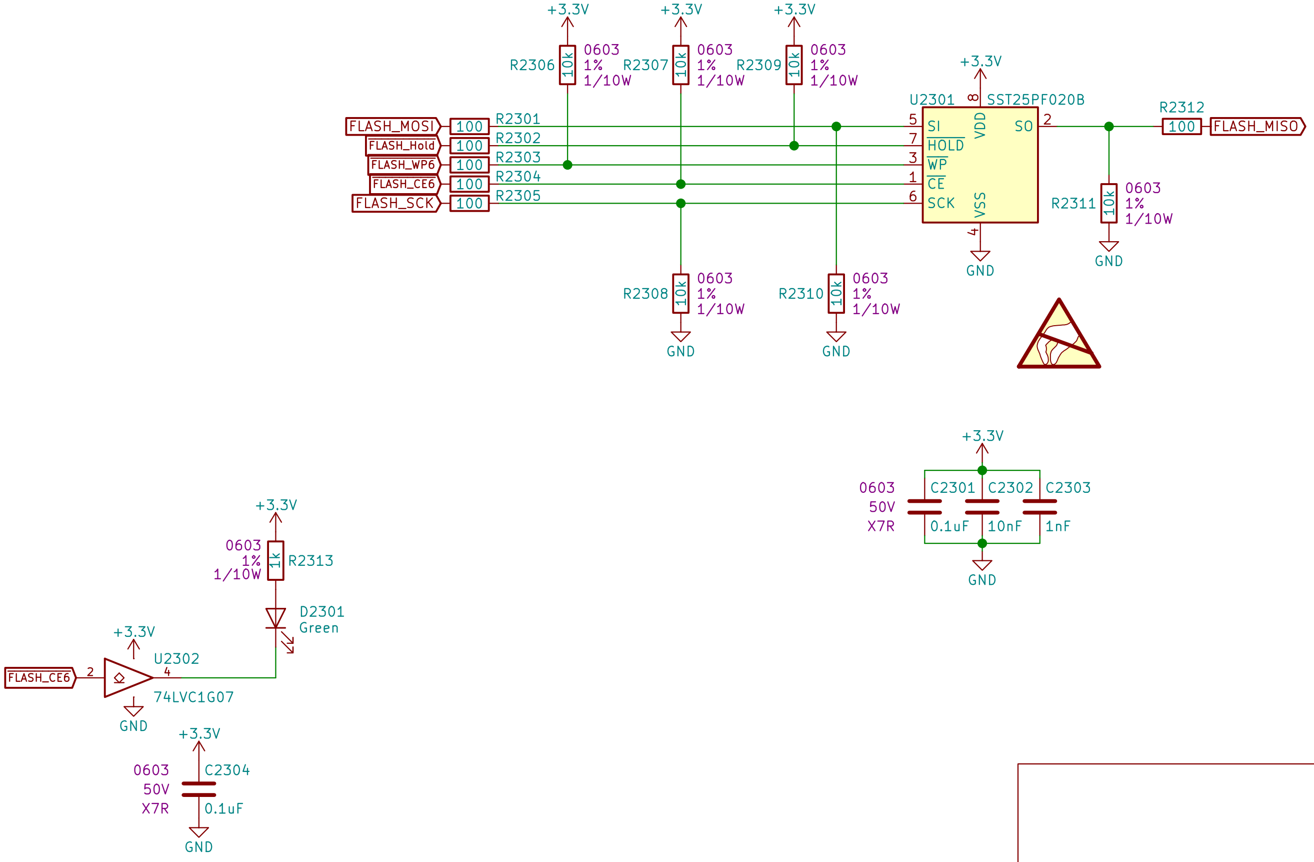
Sheet: /External Flash 5/
File: External_Flash_5.sch

Title: Electronic Display Logic Board

Size: A | Date: 2018-12-15 | **Rev: A**

KiCad E.D.A. kicad (5.1.0)-1 | Id: 22/32

23. External FLASH 6



Marquette University Senior Design 2018/2019 Group E44

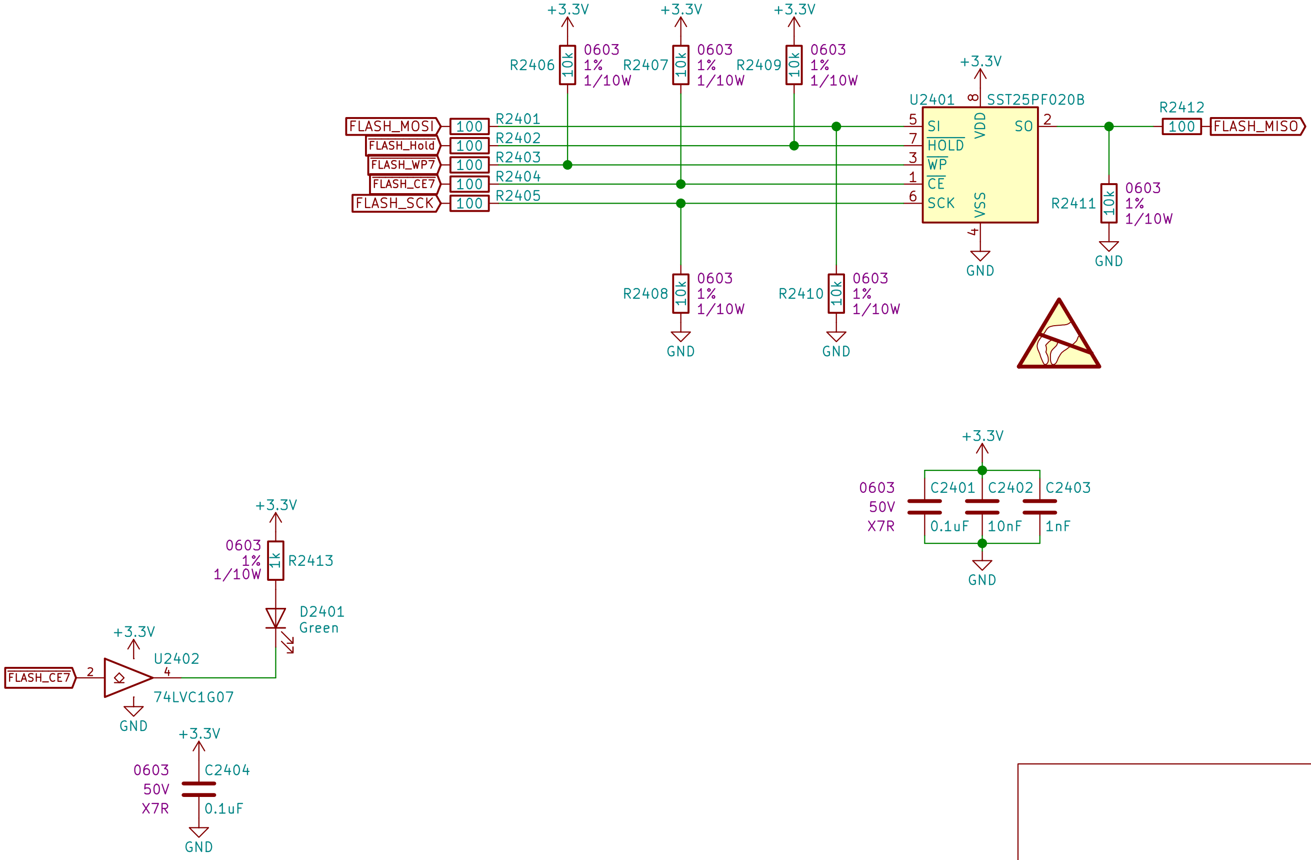
Sheet: /External Flash 6/

File: External_Flash_6.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 23/32

24. External FLASH 7



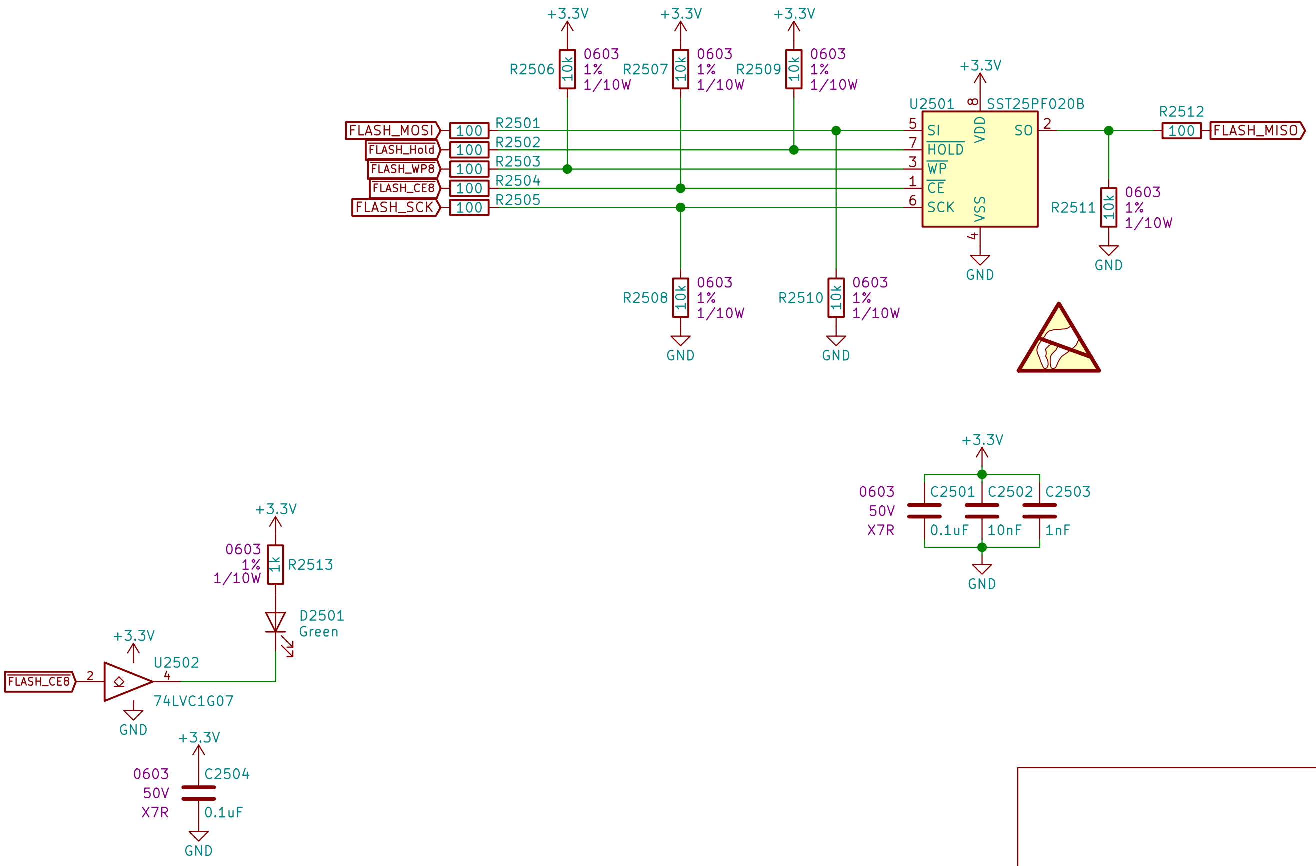
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 7/
File: External_Flash_7.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 24/32

25. External FLASH 8



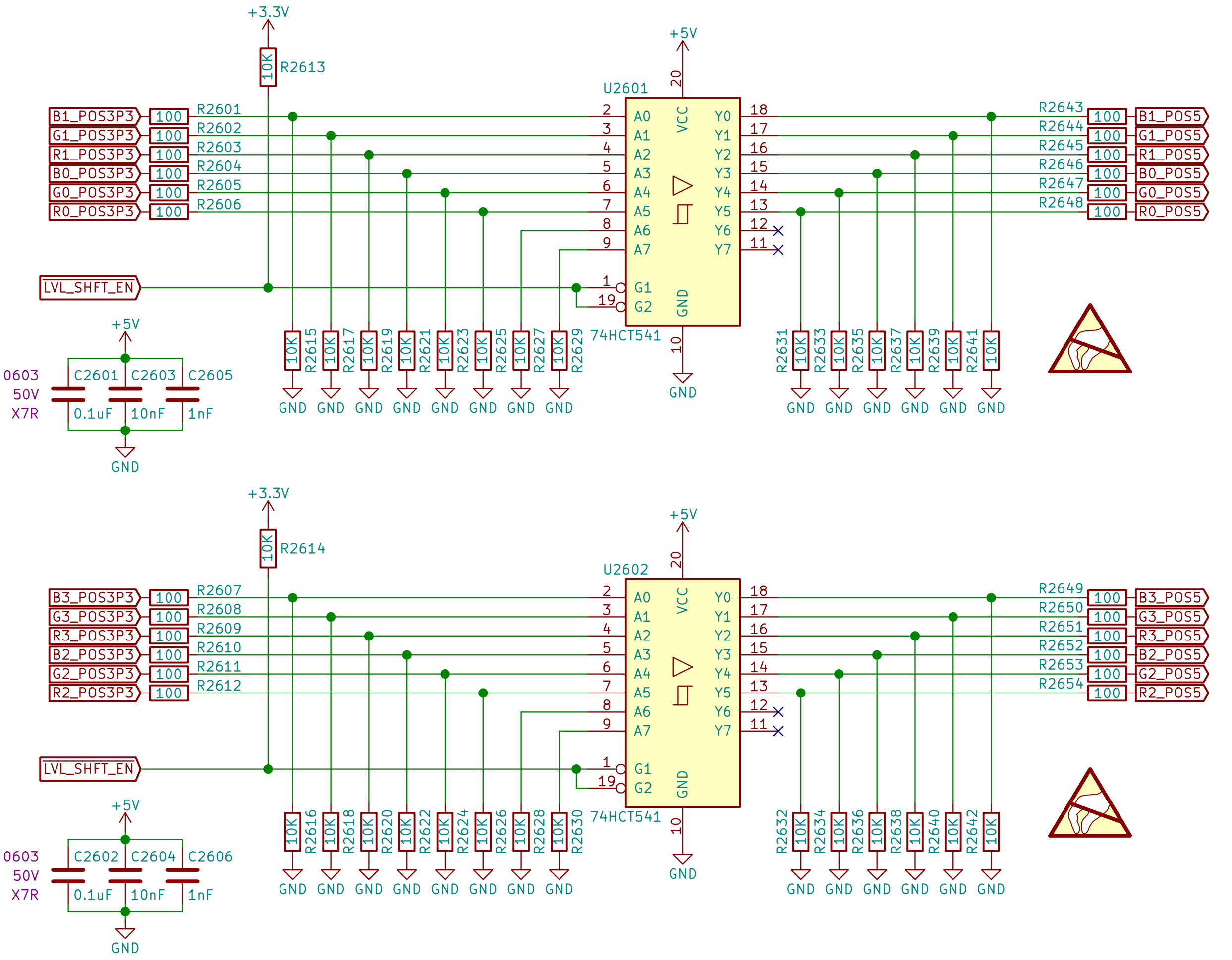
Marquette University Senior Design 2018/2019 Group E44

Sheet: /External Flash 8/
File: External_Flash_8.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 25/32

26. Panel Data Level Shifters 1



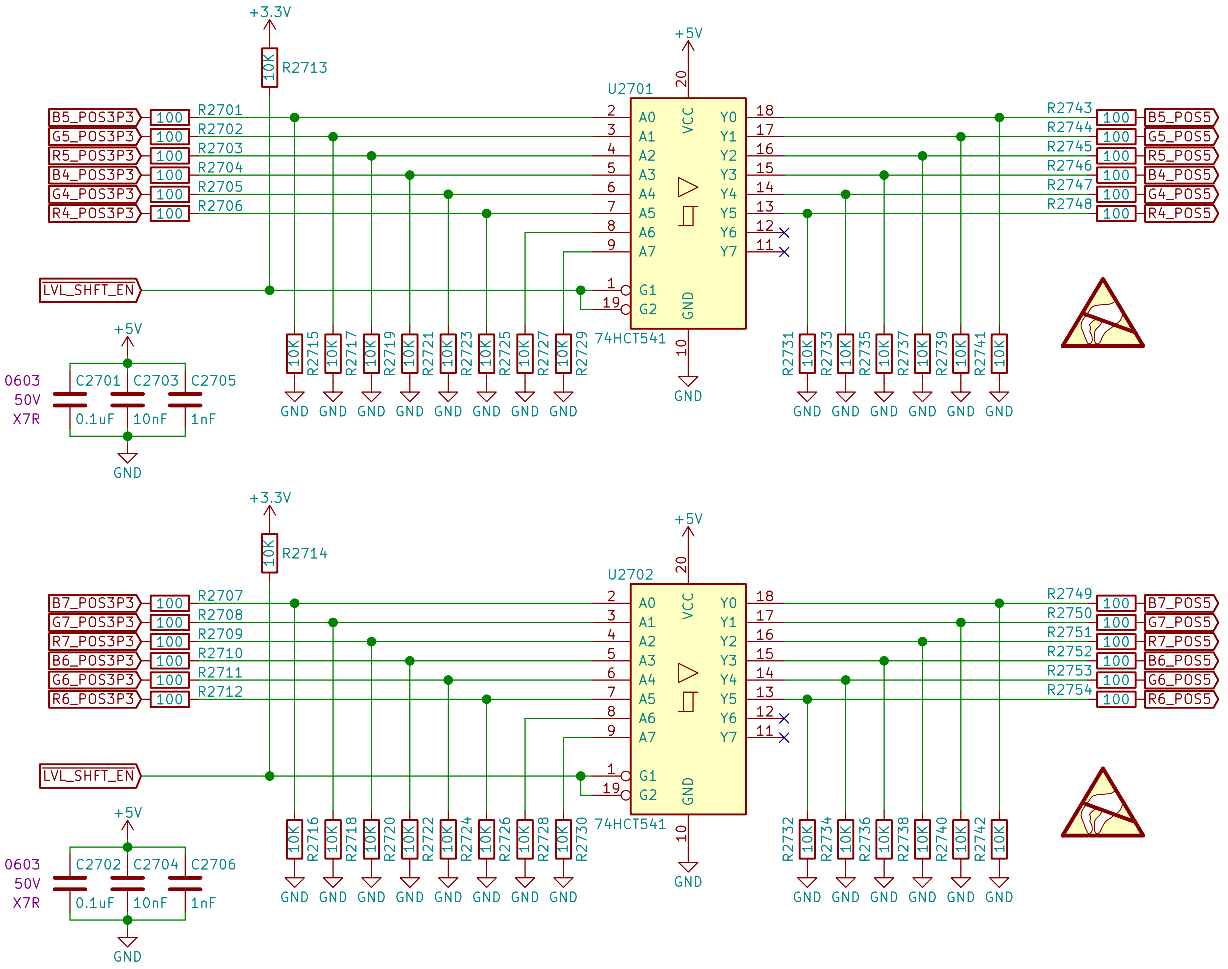
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Panel Data Level Shifters 1/
File: Panel_Data_Level_Shifters_1.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 26/32

27. Panel Data Level Shifters 2



Marquette University Senior Design 2018/2019 Group E44

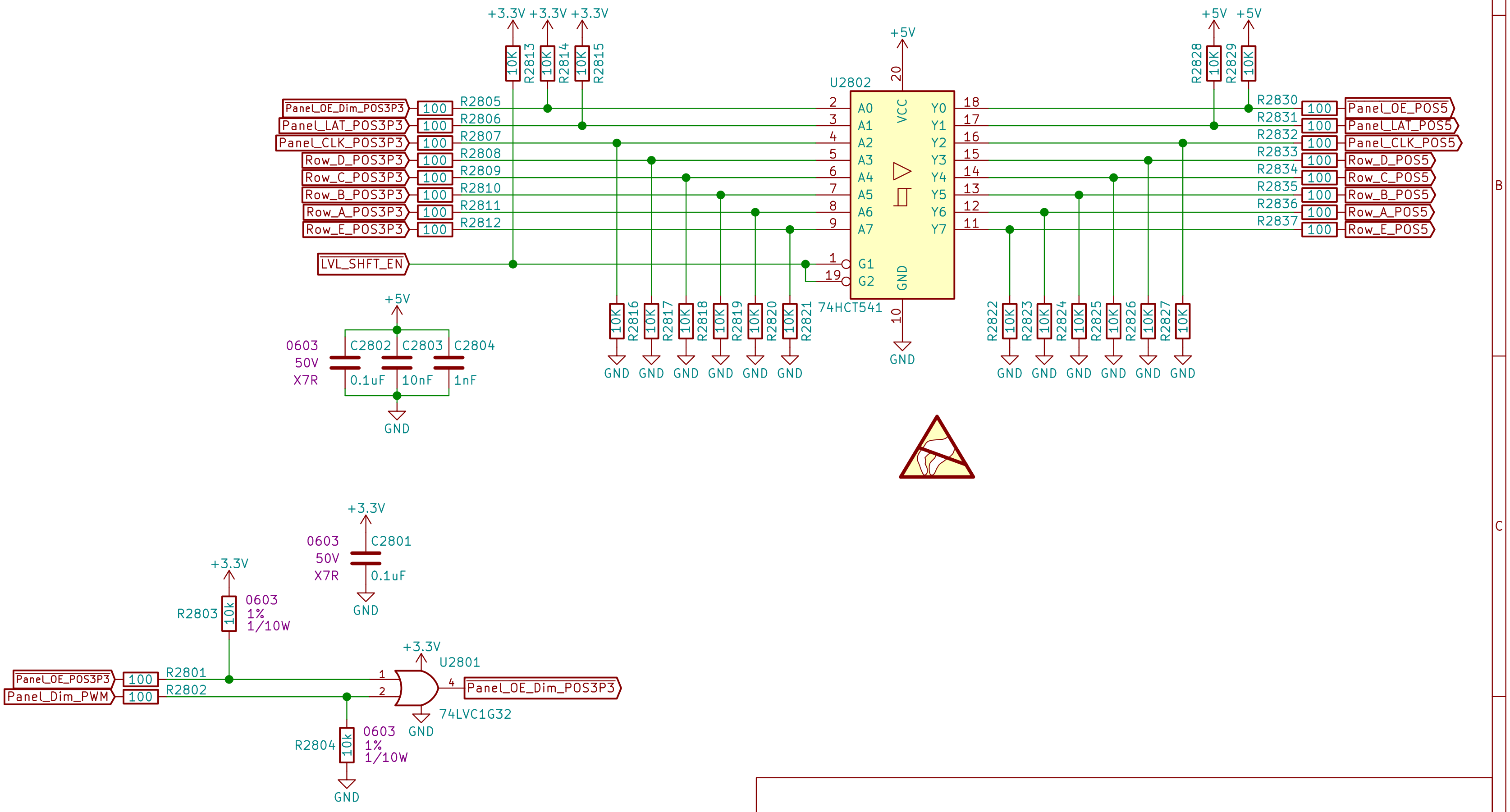
Sheet: /Panel Data Level Shifters 2/

File: Panel_Data_Level_Shifters_2.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 27/32

28. Panel Data Level Shifters 3



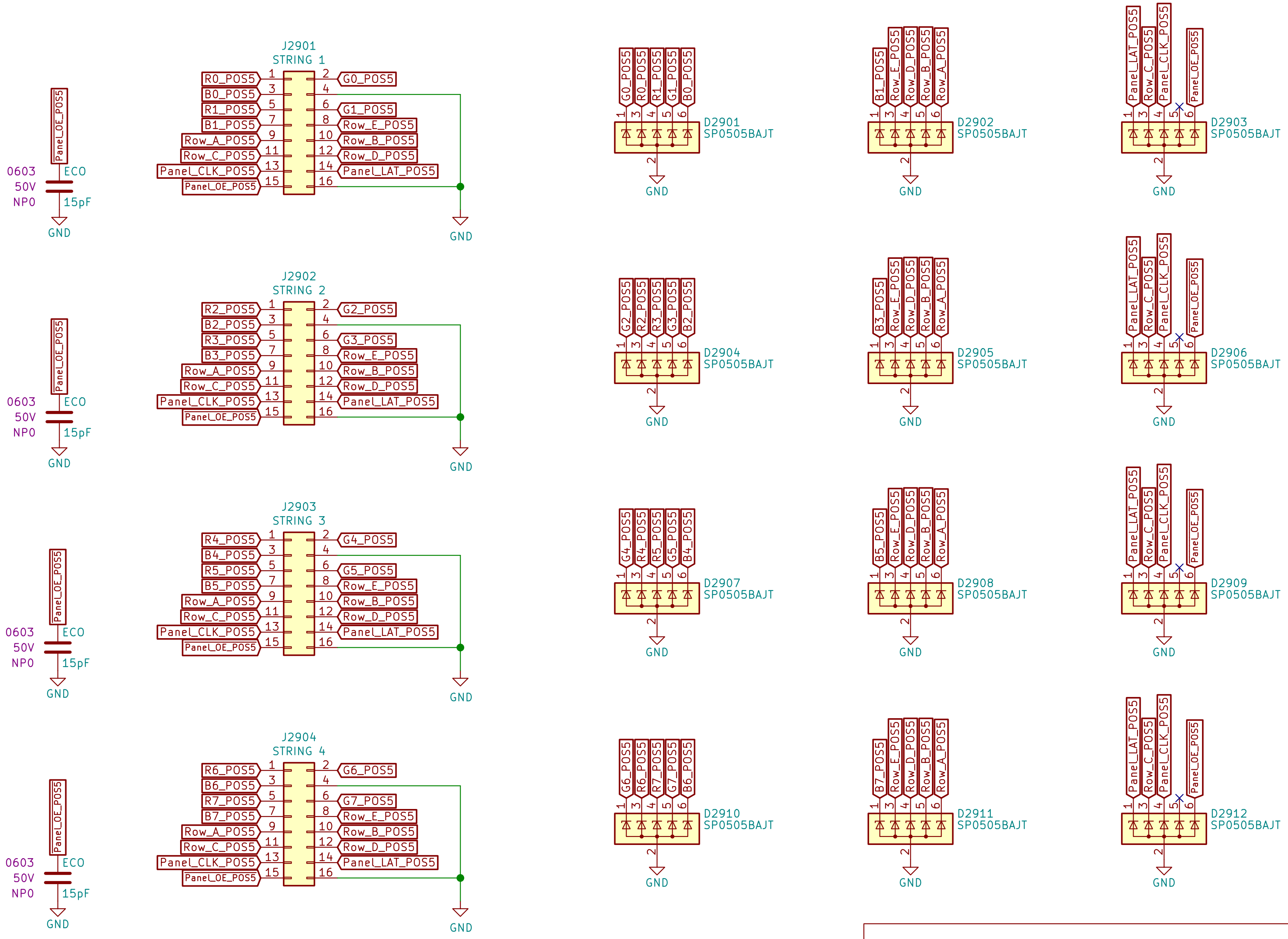
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Panel Data Level Shifters 3/
File: Panel_Data_Level_Shifters_3.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 28/32

29. Panel Data Connectors



ECO 15pF caps added to slow down $\overline{\text{OE}}$ rising edge

String 1 is the top string.
String 4 is the bottom string.
Each string is 5 panels long
Each panel is 64x64 pixels
This yields an overall 320x256 pixel display resolution

Marquette University Senior Design 2018/2019 Group E44

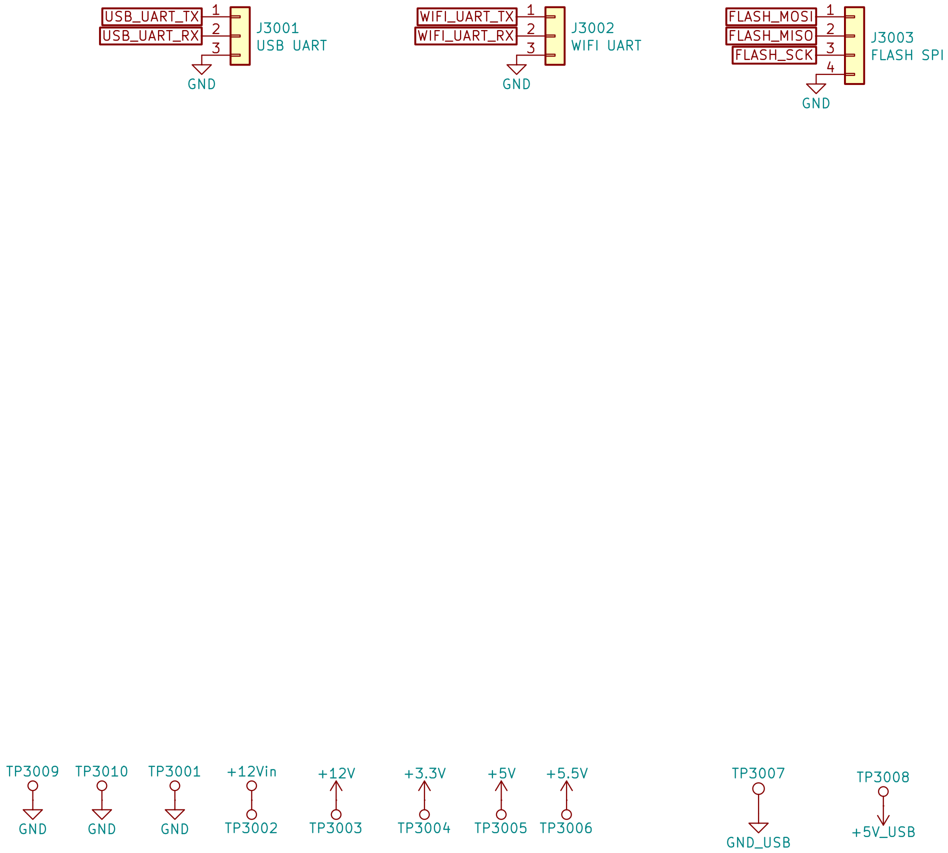
Sheet: /Panel Data Connectors/
File: Panel_Data_Connectors.sch

Title: Electronic Display Logic Board

Size: A | Date: 2018-12-15 | **Rev: A**

KiCad E.D.A. kicad (5.1.0)-1 | Id: 29/32

30. Test Points



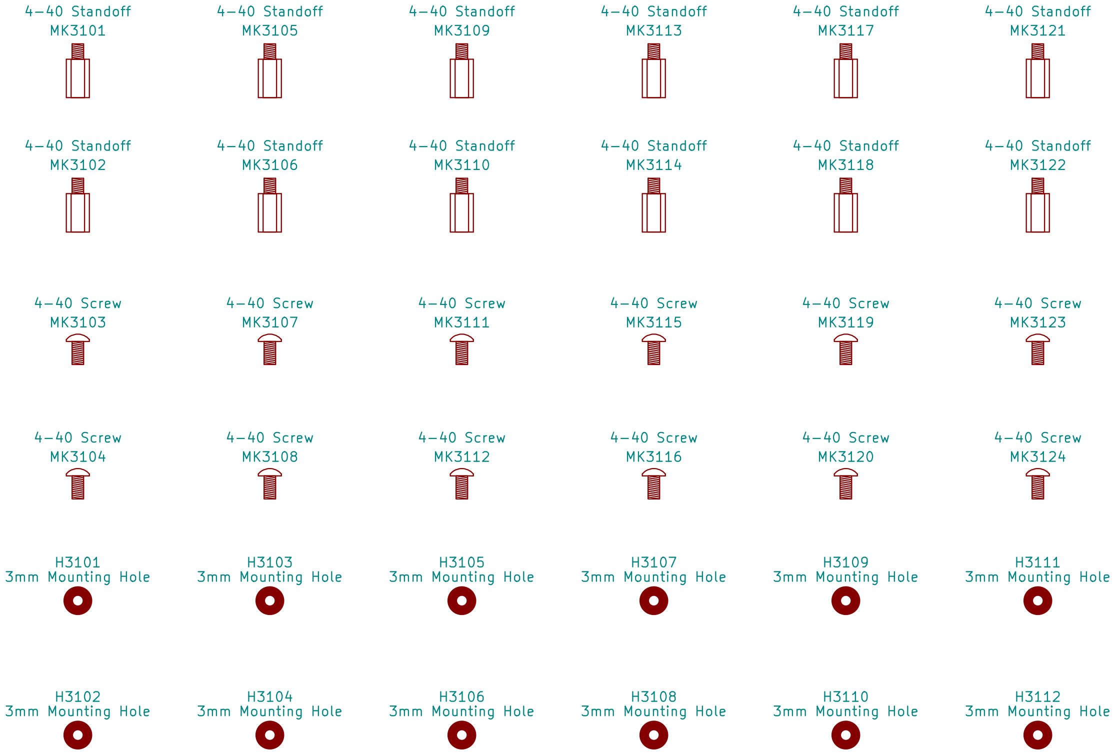
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Test Points/
File: Test_Points.sch

Title: Electronic Display Logic Board

Size: A | Date: 2018-12-15 | **Rev: A**
KiCad E.D.A. kicad (5.1.0)-1 | Id: 30/32

31. Mechanical



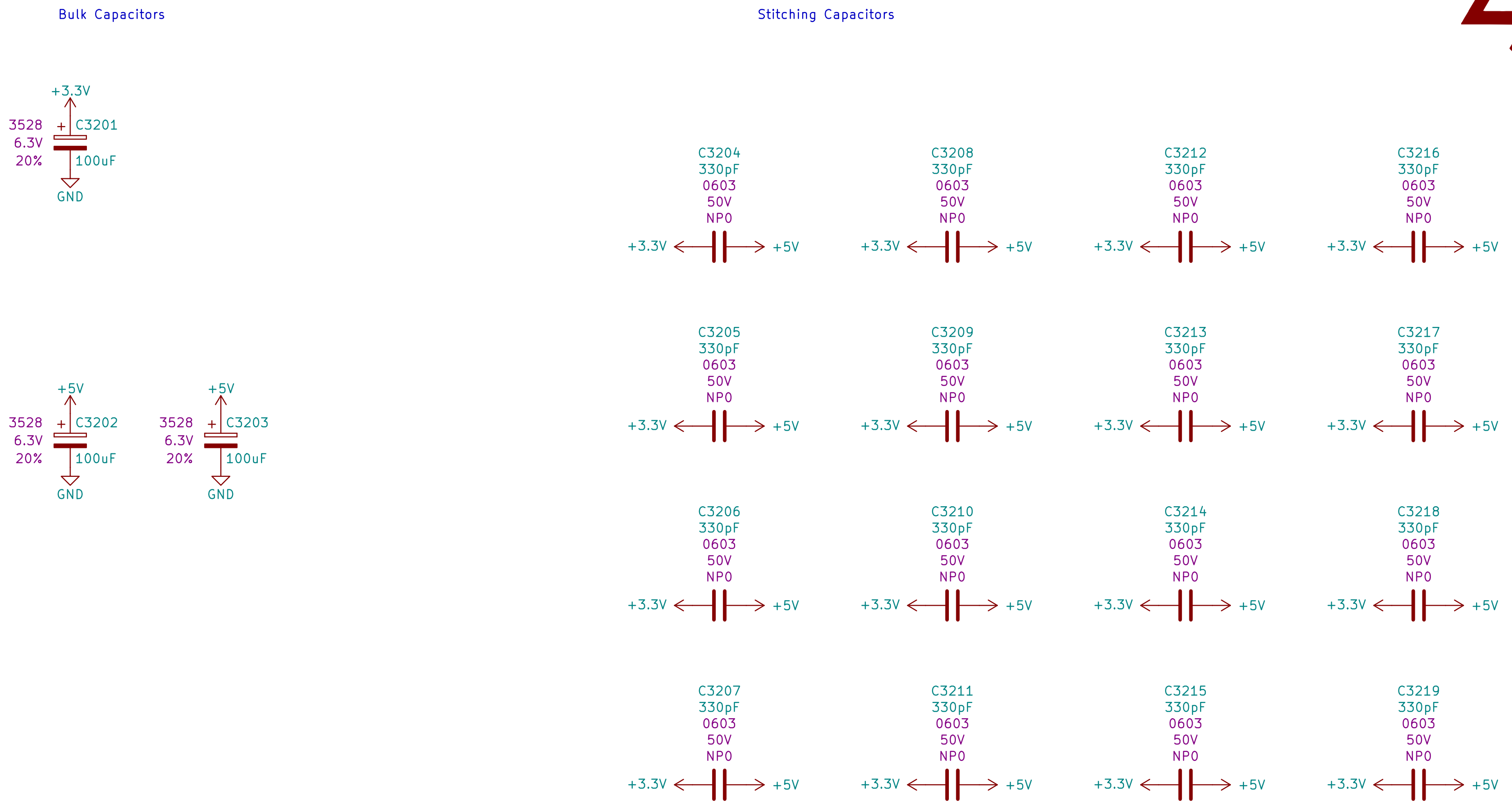
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Mechanical/
File: Mechanical.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 31/32

32. Additional Capacitance



Marquette University Senior Design 2018/2019 Group E44
Sheet: /Additional Capacitance/
File: Additional_Capacitance.sch

Title: Electronic Display Logic Board

Size: A	Date: 2018-12-15	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 32/32

01. Table of Contents:

LED Display Power Board

Marquette University Senior Design 2018/2019, Group E44

Drew Maatman, Kvein Etta, Logan Wedel, Caroline Gilger, Tuoxuan Ren

Note: If component footprints, tolerances, and power ratings are hidden, components are:
0402 case size, 1% tolerance, 1/16W power rating

Marquette University Senior Design 2018/2019 Group E44

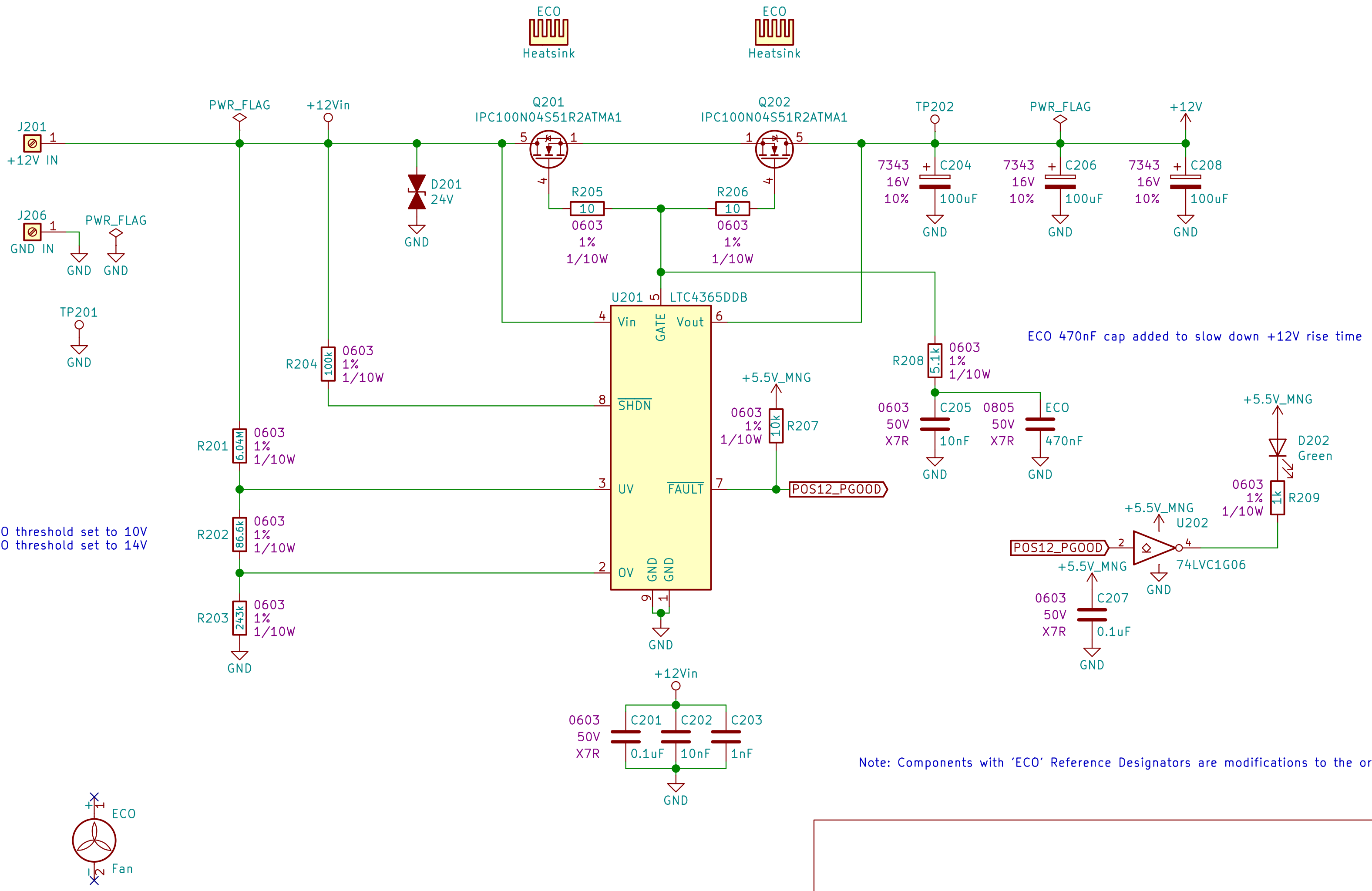
Sheet: /

File: LTC7851_Demo.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 1/14

02. Power Input



ECO heatsinks added due to insufficient PCB copper area for cooling of Q201

ECO 470nF cap added to slow down +12V rise time

UVLO threshold set to 10V
OVLO threshold set to 14V

Note: Components with 'ECO' Reference Designators are modifications to the original design

Fan added for forced air cooling.
Wired to +12V input voltage through external connector

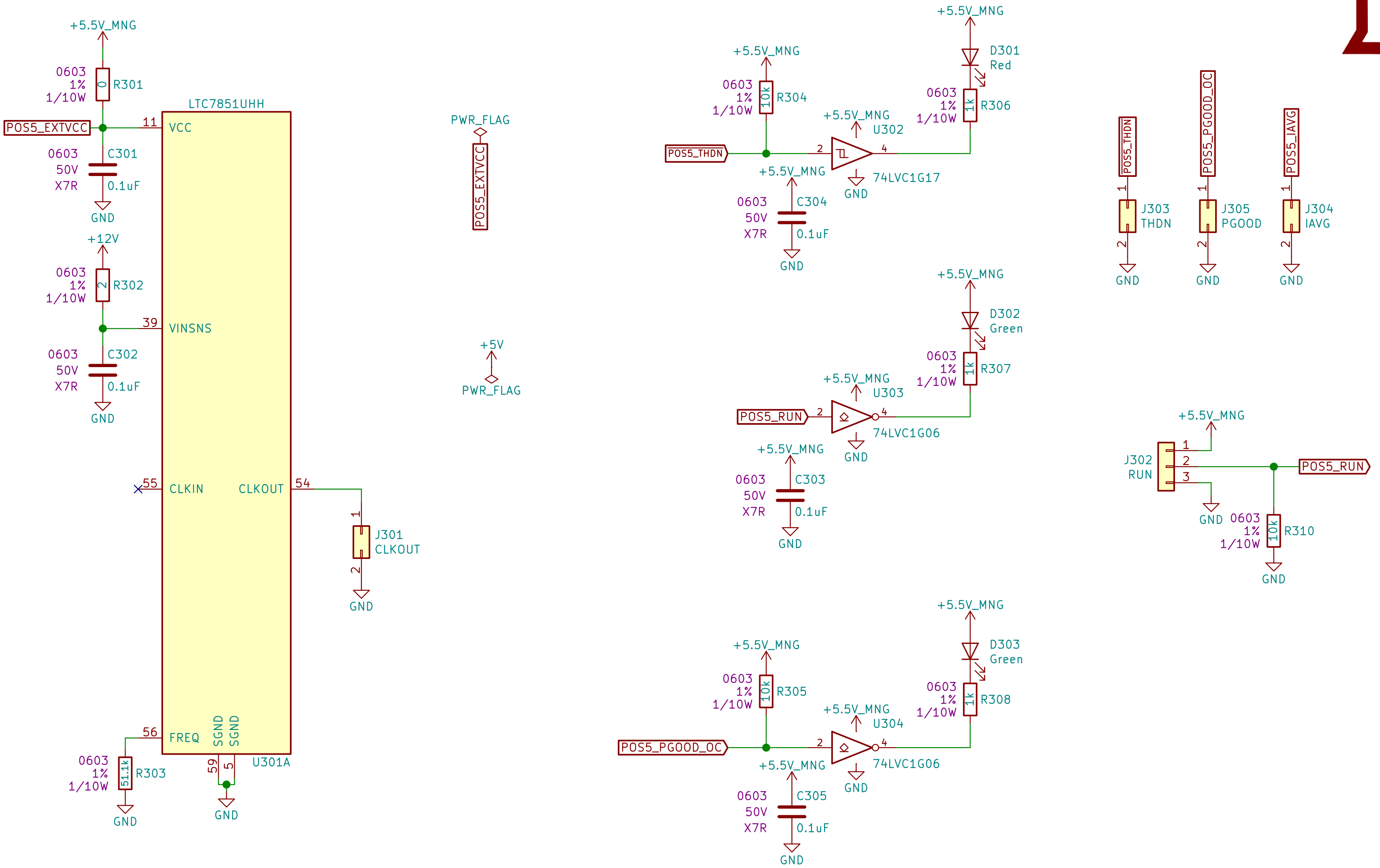
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Power Input/
File: Power_Input.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 2/14

03. +5V Control



Marquette University Senior Design 2018/2019 Group E44

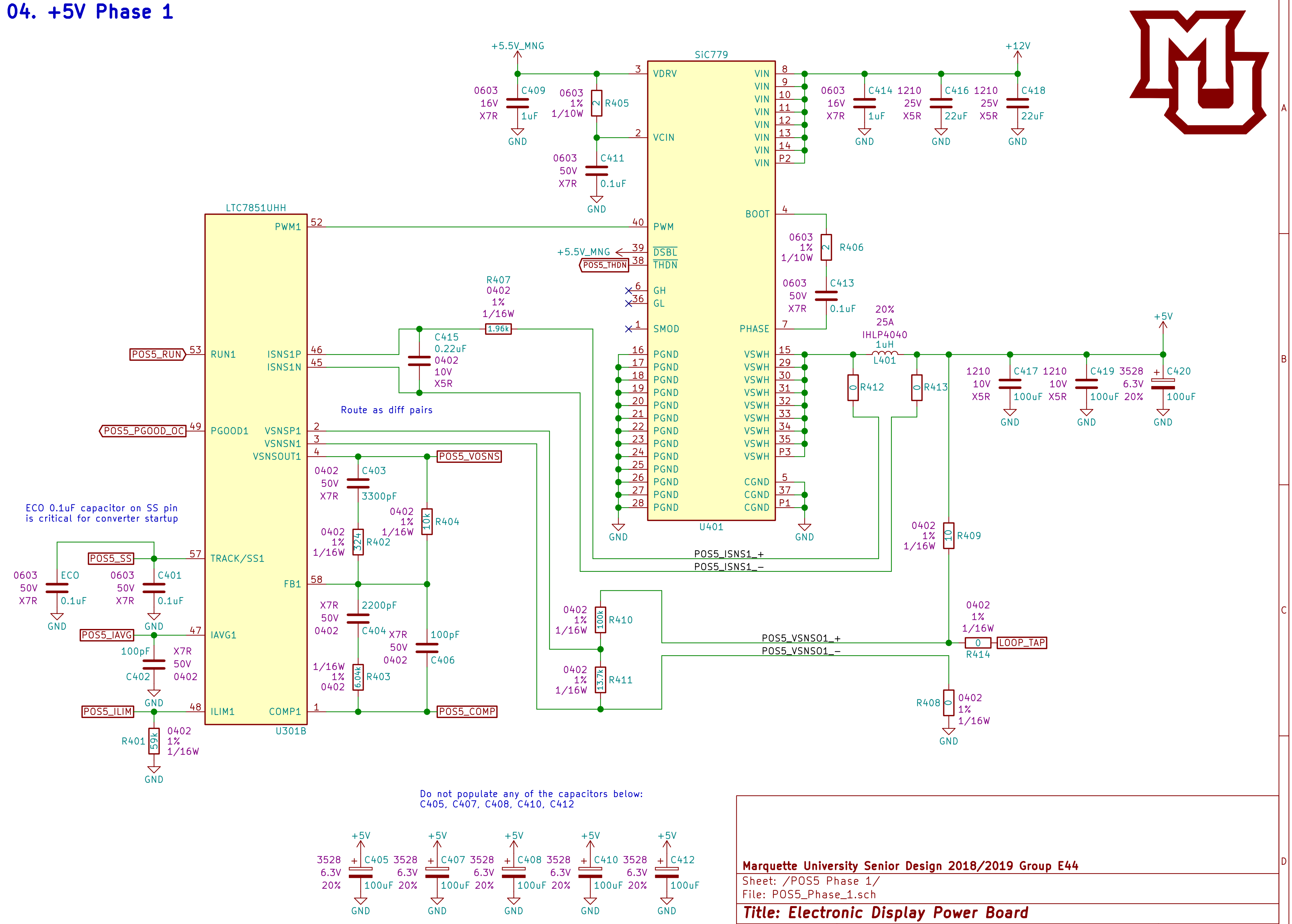
Sheet: /POS5 Control/

File: POS5_Control.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 3/14

04. +5V Phase 1



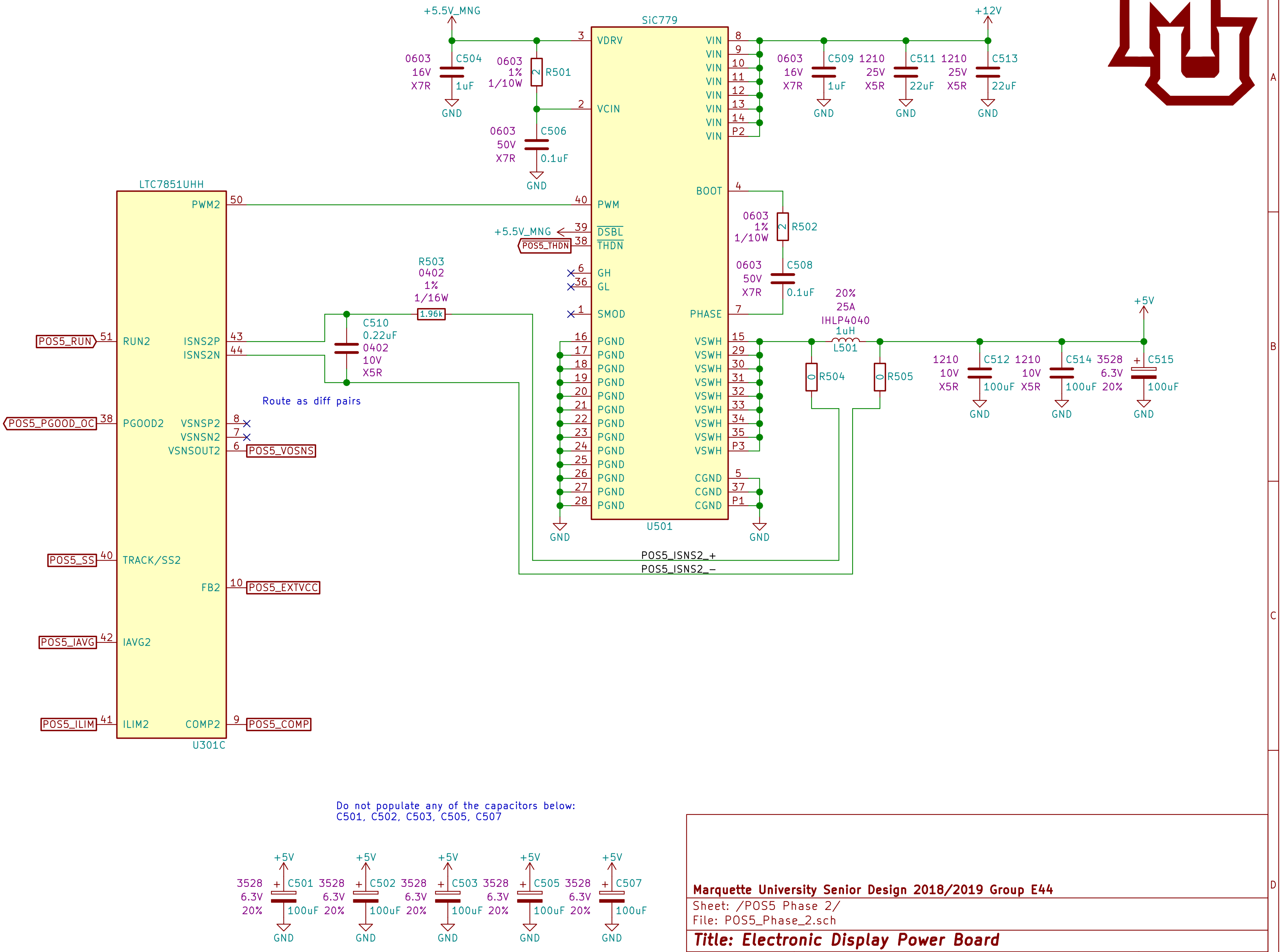
Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS5 Phase 1/
File: POS5_Phase_1.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 4/14

05. +5V Phase 2



Marquette University Senior Design 2018/2019 Group E44

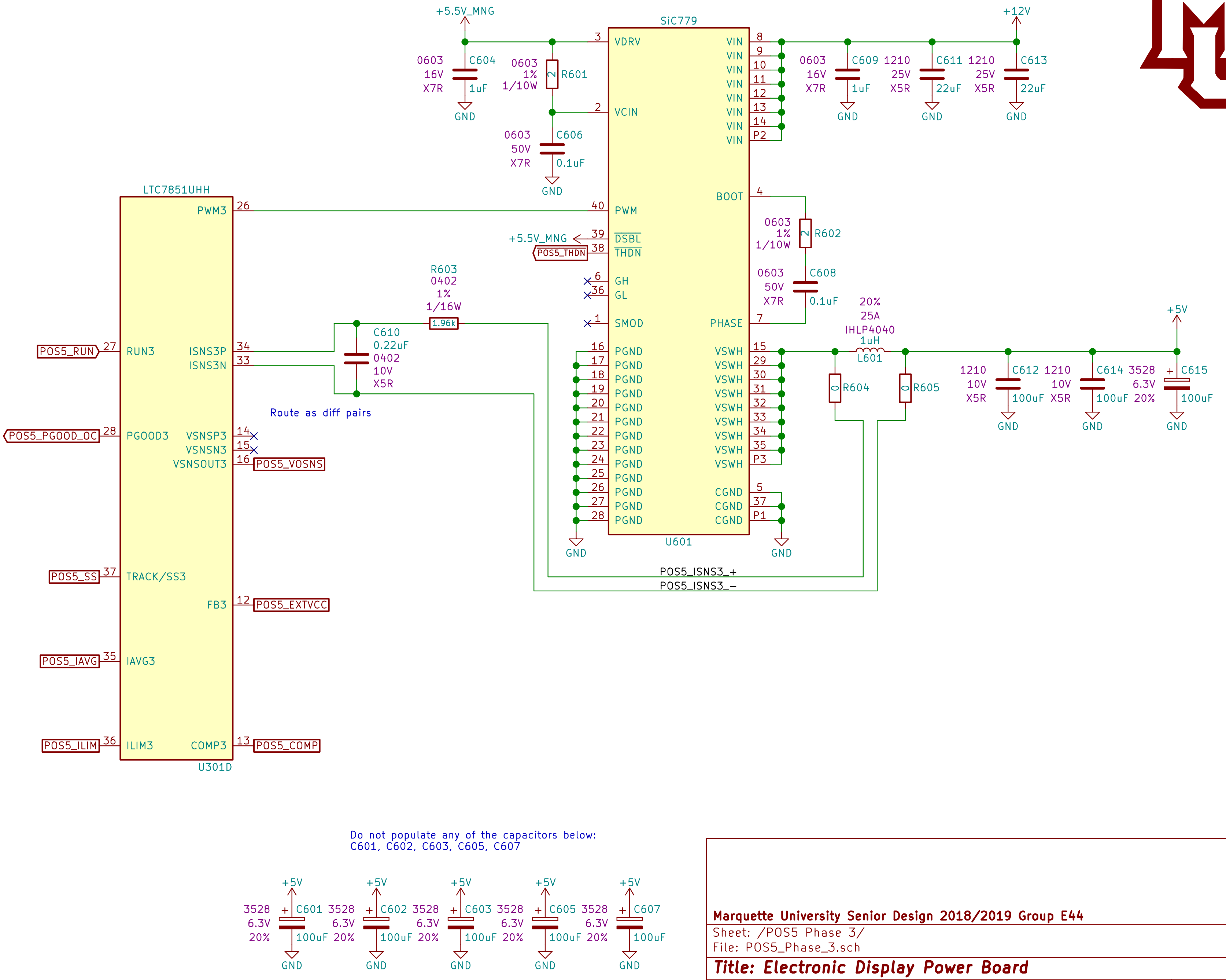
Sheet: /POS5 Phase 2/

File: POS5_Phase_2.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 5/14

06. +5V Phase 3



Route as diff pairs

Do not populate any of the capacitors below:
C601, C602, C603, C605, C607

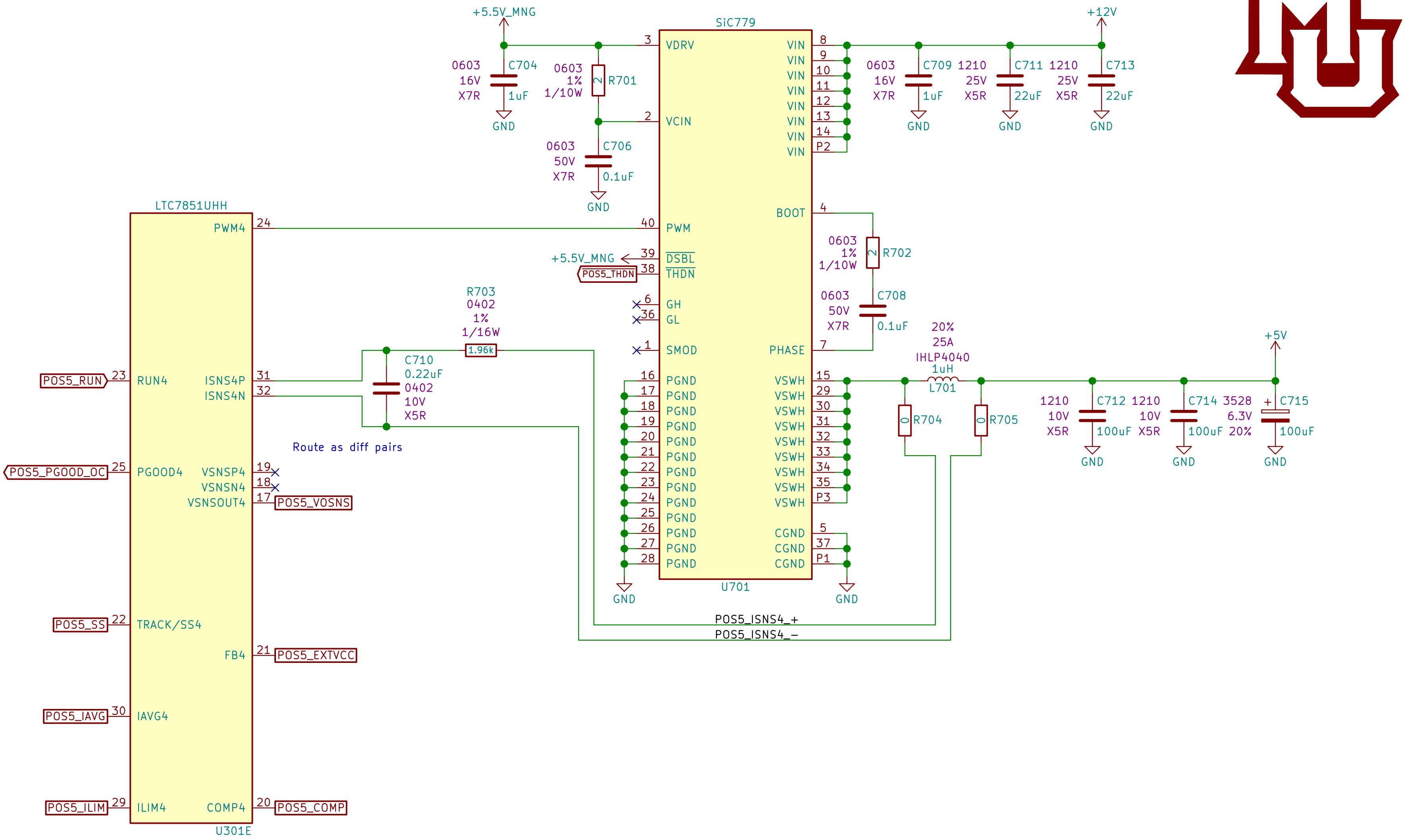
Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS5 Phase 3/
File: POS5_Phase_3.sch

Title: Electronic Display Power Board

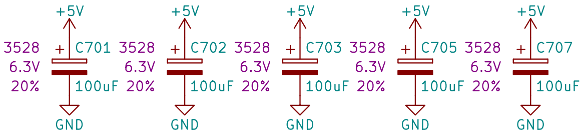
Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 6/14

07. +5V Phase 4



Route as diff pairs

Do not populate any of the capacitors below:
C701, C702, C703, C705, C707



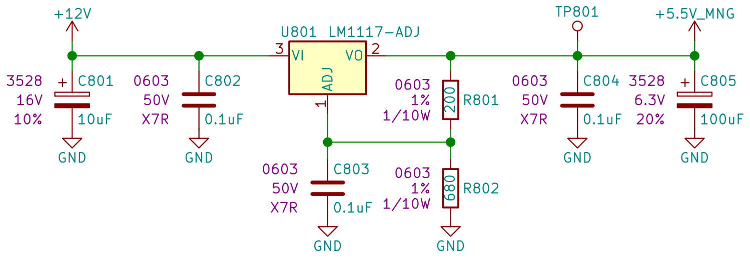
Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS5 Phase 4/
File: POS5_Phase_4.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 7/14

08. +5.5V MNG



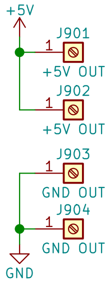
Marquette University Senior Design 2018/2019 Group E44

Sheet: /POS5P5 MNG/
File: POS5P5_MNG.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 8/14

09. Power Output



Marquette University Senior Design 2018/2019 Group E44

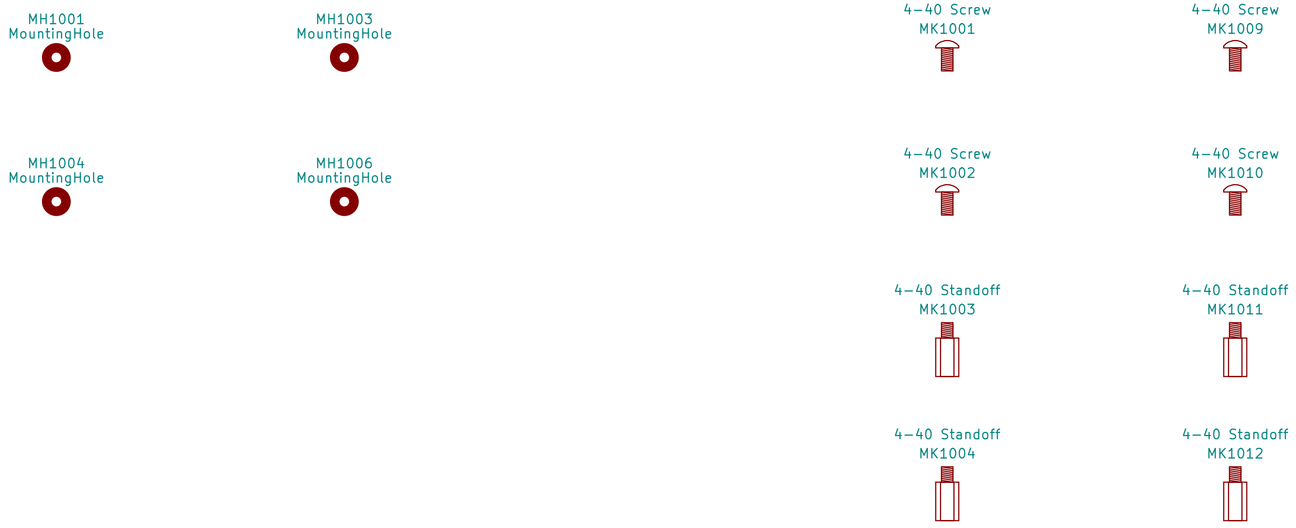
Sheet: /Power Output/

File: Power_Output.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 9/14

10. Mounting Holes and Mechanical Components



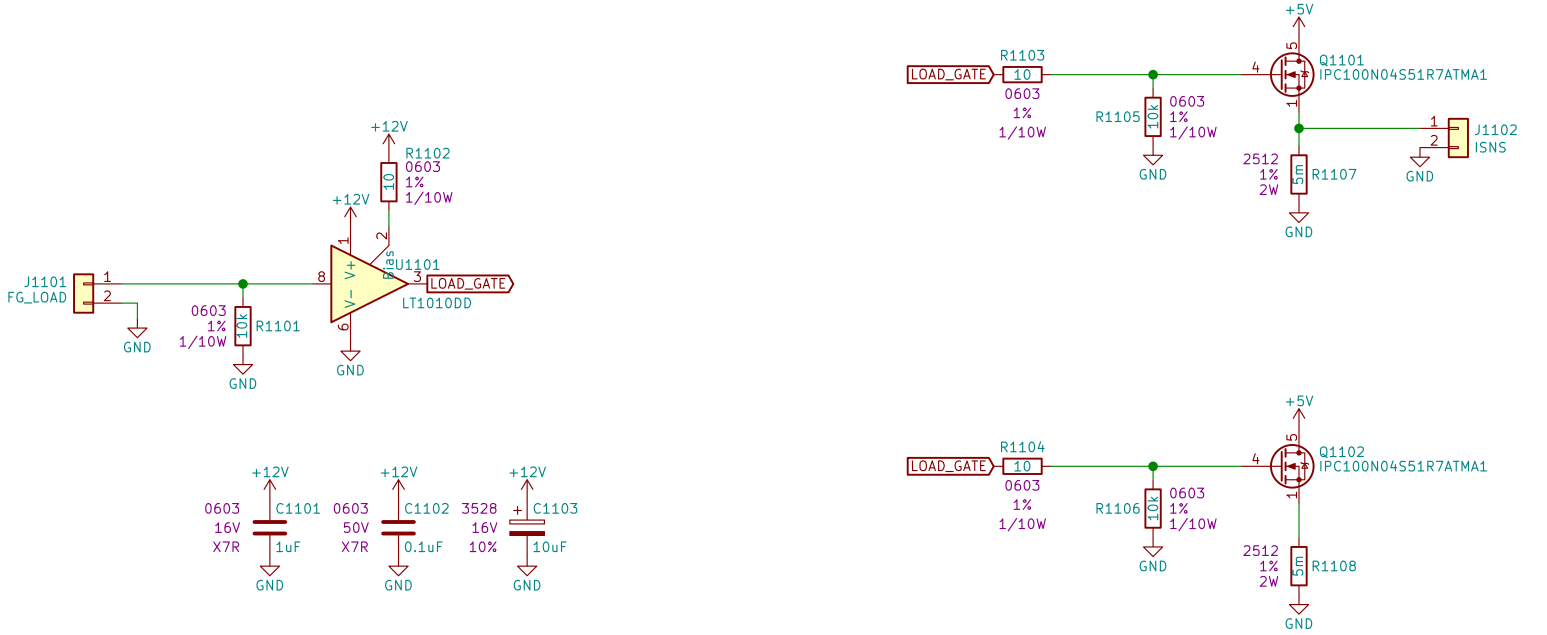
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Mechanical/
File: Mechanical.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 10/14

11. Active Load Driver and Active Load Bank 1



Do not populate any components on this sheet

Marquette University Senior Design 2018/2019 Group E44

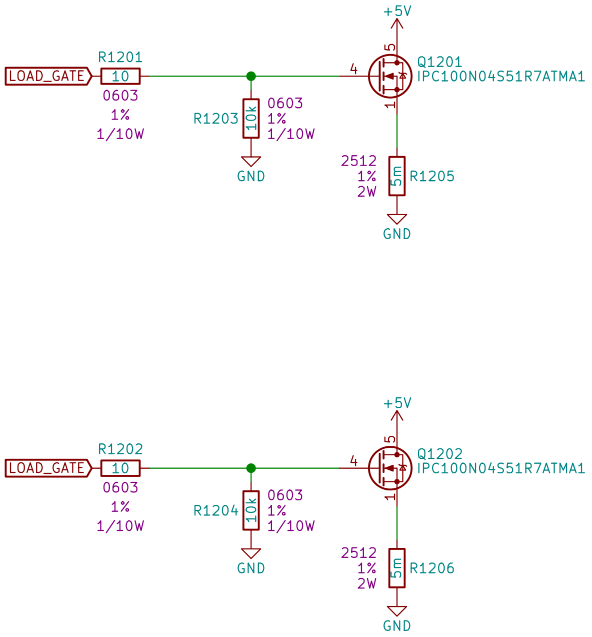
Sheet: /Active Load 1/

File: Active_Load_1.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 11/14

12. Active Load Bank 2



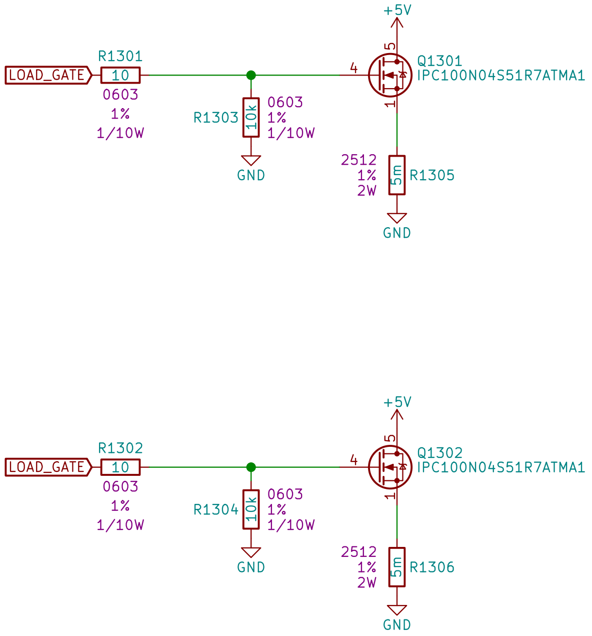
Do not populate any components on this sheet

Marquette University Senior Design 2018/2019 Group E44
Sheet: /Active Load 2/
File: Active_Load_2.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 12/14

13. Active Load Bank 3



Do not populate any components on this sheet

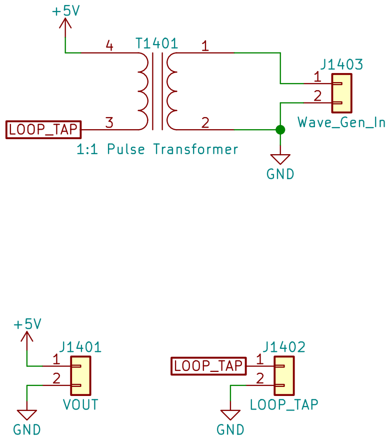
Marquette University Senior Design 2018/2019 Group E44

Sheet: /Active Load 3/
File: Active_Load_3.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 13/14

14. Open Loop Transfer Function Test Components



Do not populate any components on this sheet

Marquette University Senior Design 2018/2019 Group E44

Sheet: /Loop Response/
File: Loop_Response.sch

Title: Electronic Display Power Board

Size: A	Date: 2018-11-28	**Rev: A**
KiCad E.D.A. kicad (5.1.0)-1		Id: 14/14