ECE-388

Final Project: Pattern Generator

Marcel Vieira

December 8, 2020

I Marcel Vieira certify that this is lab report consists only of work that is my own.

Contents

List of Figures

1 Abstract

This document describes the process taken in planning designing and constructing a pattern generator. The steps detailed includes identification and defining the problem, creating a bill of materials, creating simulations, prototyping, and drafting PCB design layouts. This document also details the problems encountered along with how they were solved or proposed solutions. Due to the project being incomplete this only details up to the creation of the PCB and the creation of a demonstrational prototype.

2 Problem Statement

Use surface mount components and a ATMEGA 328pb to generate binary patterns at either 5V or 3.3V, with a user interface that allows a user to change the pattern as well as the logic level across 8 independent channels.

3 Procedures

3.1 Materials

- Multisim
- Eagle
- Web Browser
- Atmel Studio

3.2 Test Process

In order to begin the process of creating a pattern generator the problem must be properly defined to set the guidelines for what the solution will be. To define the problem the constraints must be

laid out in an Engineering note book and clarified with the TAs and professor. They required that the ATMEGA328PB chip be used, and all components that are chosen must be surface mount. These two constraints are not enough to clearly define the problem so more must be added such as an ability to switch logic levels and a user interface that allows people to interact with the board and adjust the the output pattern. Now the problem can be defined with a problem statement, in this case it is done with the one above.

Begin brain storming possible ways to resolve the problem. The ATMEGA328PB is capable of outputting digital signals in patterns so one of the biggest issues is changing the 5V is outputs from 5V to 3.3V when desired. This could be done with a dc converter, using resistors or diodes to drop the voltage, or using MOSFETS to toggle the flow of electricity form whatever logic source is selected. The second issue is in getting the 3.3V form the 5V that will come into the board. This could be done using a voltage divider, a DC to DC converter, or a Zener diode. The third issue was how was the user interface going to be made. The interface could be made either using hard ware switches, buttons, and rotary encoders or use serial port to type in the pattern. The last problem was how was information going to be shown to the user? The information could be shown using as series of LEDs or on a screen.

The next step is to decide which brainstormed solutions will be used to solve the problem. For this project the user interface will use a serial input with a screen for the display. For the power logic level selection a Zener diode will be used to get the 3.3V with a sp3t switch to switch between the power states which will include 5V 3.3V and 0V. The voltage that comes out of the switch then will be passed to MOSFETS that will be toggled by the ATMEGA328PB chip.

Now that the plan has been completed the parts must be selected and tested in Multisim (figure 1). For this case the displays cannot be properly tested so they will be tested physically later on. The MOSFETS when input in Multisim showed that the were capable of using 5V at the gate to toggle both 5V and 3.3V. The Zener diode also showed it was a valid solution to getting 3.3V.

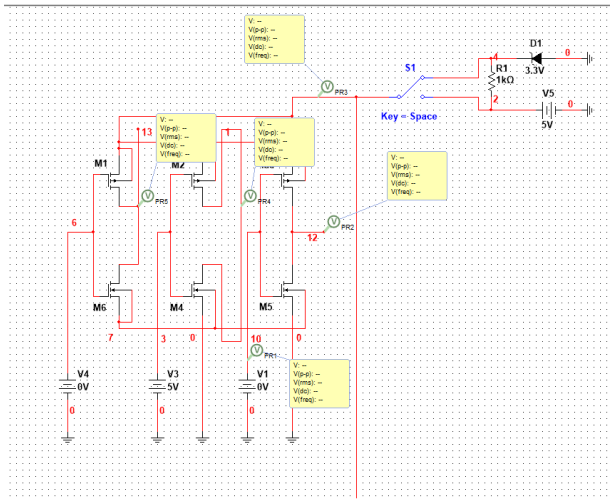


Figure 1: Pattern Gen Multisim V1

Then the parts that are required to build the circuit are searched in LCSC and Digikey. When a part was found it was inserted into Eagle (figure 2). Once all the parts were placed in Eagle a bill of materials is exported and opened in excel in order to add purchase information and remove irrelevant information (figure 3).

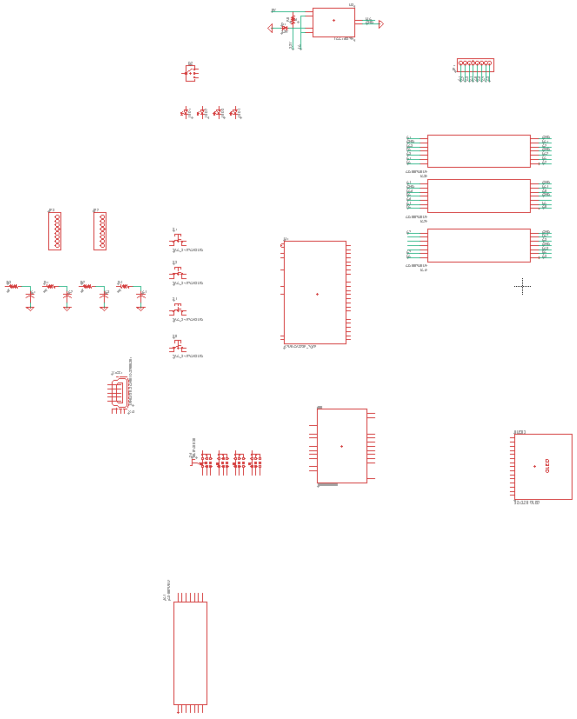


Figure 2: Pattern Gen Eagle Schematic V1

Qty	Value	Device	Package	Parts	Description	Sorce	Price for 1	Price for 5	Price for 10
4	1uF	C-USC0805	C0805	C1, C2, C3, C4	CAPACITOR, American symbol	Digikey	0.15	1.05	4.93
4		LEDCHIP-LED0805	CHIP-LED0805	LED1, LED2, LED3, LED4	LED	Digikey	0.27		7.27(Reel)
3		PINHD-1X8-3.5MM	1X08-3.5MM	JP1, JP2, JP3	PIN HEADER	Digikey	0.65	6.41	54.2
1	CUS-13	SWITCH-SPST-SMD	SWITCH_SPST_SM	S2	Single-Pole, Triple-Throw (SP3T) Switch	Digikey	0.72	6.96	58.08
1	AZ23C3V3	ZENER-DIODESOT23	SOT23	D1	Z-Diode	Digikey	0.33		7.33(Reel)
5	1k	R-US_M0805	M0805	R1, R2, R3, R4, R5	RESISTOR, American symbol	Digikey	0.1		7.1(Reel)
1	32x128 OLED	OLED128X64SIDE	OLED-UG-2832TSV	OLED1	Micro OLED	Stock Sheet			
1	EN11-HSM1BF20	EN11-HSM1BF20	S6		Rotary Encoder	Digikey	1.16	10.75	79.55
1	ATMEGA328P_TQFP	TQFP32-08	U1		Popular 328P in QFP	Digikey	1.34	13.4	112
4	CD4007UBM	SOIC127P600X175	IC1, IC2, IC3, IC4		Texas Instruments CD4007UBM Inverter, 3 18 V, 14-Pin SOI	Digikey	0.45	3.83	28.58
1	MCP2200-I/SS	SOP65P780X200-2	U2		MCP2200 Series 5.5 V 8 Channel USB to UART Serial Conver	Digikey	1.94	19.4	147
1	MINI-USB-SCHIELD-32005-301	MINI-USB-SCHIELD-3	32005-301	X1	MINI USB-B Conector	Stock Sheet			
4	TAC_SWITCHSMD	TACTILE-SWITCH-SM		S1, S3, S4, S5	Momentary Switch	Stock Sheet			
1	LCC110STR	SOIC254P952X67-8N			Solid state relay	Digikey	3.84		10.84(Reel)

Figure 3: Pattern Gem BOM 1

After the parts were selected and the circuit was successfully simulated it was time to begin construction of the prototype. The prototype allows for the circuit to be tested prior to having everything on the PCB. The use of a prototype especially one on a bread board allows for parts to be more easily isolated and replaced.The the prototype is constructed based on the circuit design used in Multisim but the prototype is broken into sections to test the functionality of all the individual smaller parts prior to testing the circuit in its entirety. The prototype consisted of a voltage regulator, a switching circuits made with 2 PMOSs and an inverter, and a set of CMOS inverters.

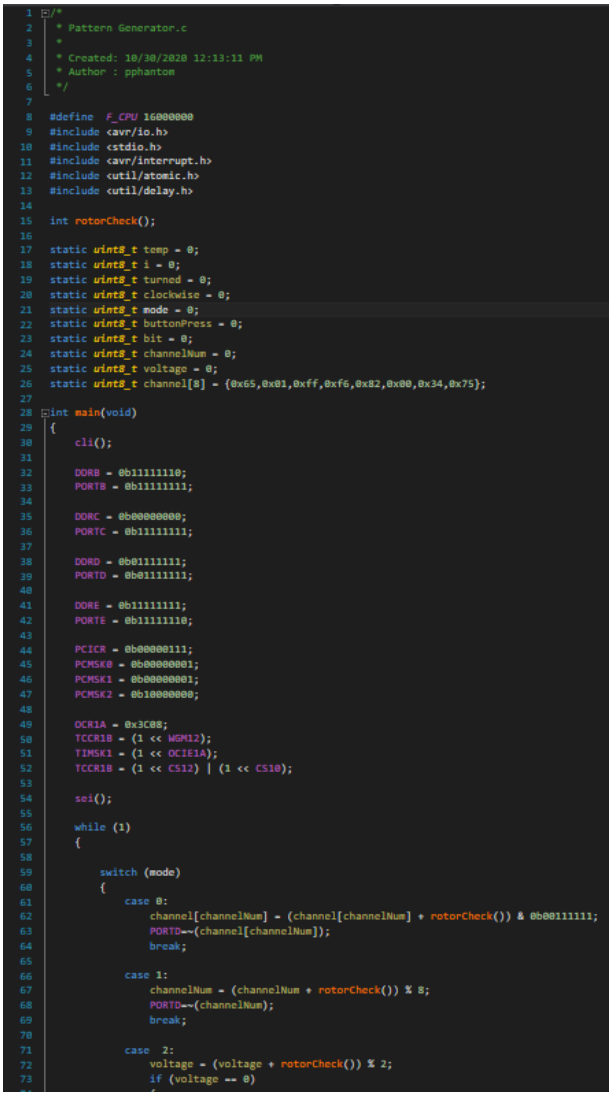
The first test conducted was connect a voltmeter to the the voltage regulator and check if the proper voltages are produced. The expected values were 5V at the input and 3.3V at the output. This ensures that the voltage does not damage the circuit and sets a base line for the voltage to expect in the inverters.

The second test is to put 5V and 3.3V to the switching circuit use the voltmeter to check if it is outputting the correct input depending on what signal is sent to it. If a high is put into the circuit it should output 3.3V, if a low is put into the circuit it should output 5V.

The final test for the first prototype was to check if the CMOS inverters could produce the expected signal. To test this a clock pulse is put into the inverter and the response as well as the input are measured and compared with an oscilloscope. The result showed be the opposite of the input with the exception of the high being what voltage is put to the sources of the PMOSs in the CMOS inverters.

Figure 4: Prototype Output

After taking notes on the first prototype the second was built. This prototype is focused on the user interface. This prototype consists of a rotary encoder with a built in button, several LEDs, and a 328pb. For this prototype code was written in Atmel Studio. The code for the prototype used various interrupts with 2 pin change interrupts being used to determine the direction the rotary encoder is turned on. There is also another pin change interrupts that is used to change the button press. This user interface was tested using debugging and visual observation to tell whether it is operational.



```c
/*
 * Pattern Generator.c
 *
 * Created: 10/30/2020 12:13:11 PM
 * Author : pphantom
 */

#define  F_CPU 16000000
#include <avr/io.h>
#include <stdio.h>
#include <avr/interrupt.h>
#include <util/atomic.h>
#include <util/delay.h>

int rotorCheck();

static uint8_t temp = 0;
static uint8_t i = 0;
static uint8_t turned = 0;
static uint8_t clockwise = 0;
static uint8_t mode = 0;
static uint8_t buttonPress = 0;
static uint8_t bit = 0;
static uint8_t channelNum = 0;
static uint8_t voltage = 0;
static uint8_t channel[8] = {0x65,0x01,0xff,0xf6,0x82,0x00,0x34,0x75};

int main(void)
{
    cli();

    DDRB = 0b11111110;
    PORTB = 0b11111111;

    DDRC = 0b00000000;
    PORTC = 0b11111111;

    DDRD = 0b01111111;
    PORTD = 0b01111111;

    DDRE = 0b11111111;
    PORTE = 0b11111110;

    PCICR = 0b00000111;
    PCMSK0 = 0b00000001;
    PCMSK1 = 0b00000001;
    PCMSK2 = 0b10000000;

    OCR1A = 0x3C08;
    TCCR1B = (1 << WGM12);
    TIMSK1 = (1 << OCIE1A);
    TCCR1B = (1 << CS12) | (1 << CS10);

    sei();

    while (1)
    {

        switch (mode)
        {
            case 0:
                channel[channelNum] = (channel[channelNum] + rotorCheck()) & 0b00111111;
                PORTD=~(channel[channelNum]);
                break;

            case 1:
                channelNum = (channelNum + rotorCheck()) % 8;
                PORTD=~(channelNum);
                break;

            case  2:
                voltage = (voltage + rotorCheck()) % 2;
                if (voltage == 0)
```

Figure 5: Pattern Gen Code V1 PG 1

```
                }
                else
                {
                    PORTD = 0b11011111;
                }
                break;
            }
        }
    }


int rotorCheck()
{
    if(turned)
    {
        turned = 0;
        if(clockwise)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }

    return 0;
}


ISR(PCINT0_vect)
{
    if(!turned)
    {
    clockwise = 1;
    }
    turned = 1;
    _delay_ms(150);
}


ISR(PCINT1_vect)
{
    if(!turned)
    {
    clockwise = 0;
    }
    turned = 1;
    _delay_ms(150);
}


ISR(PCINT2_vect)
{
    if (!buttonPress)
    {
    mode = (mode + 1) % 3;
    }
    buttonPress = ~buttonPress;
}


ISR(TIMER1_COMPA_vect)
{
    cli();

    i=0;
    temp = 0;
```

Figure 6: Pattern Gen Code V1 PG 2

The user interface should have LEDs that start off displaying the current binary signal, this is referred to as mode 1. In mode 1 the rotary encoder changes the binary value going up when turned clockwise and down when turned counter clockwise. When the button is pressed it enters mode 2 which allows the user to view the the current channel number in binary displayed on the LEDs. Mode 2 also allows for the user to switch between channels using the rotary switch. When the button is pressed again it goes into mode 3 which simulates the display of the voltage settings which is adjustable with the rotary encoder as well.

Once the final design has been decided on the the design must properly be translated into a schematic that will be uploaded to Github and checked by Tim Chase. The schematic must contain all parts that are to be used, in this instance that would be the micro USB, the CMOS chips, the serial chip, the LEDs, the OLED display, and the rotary encoder. Power must flow in the schematic from the left to right and be clearly labeled. The schematic must finally pass all ERC checks with no errors or only errors approved by Tim Chase.

Once the schematic is approved by Tim Chace the board lay out can then be worked on. The board layout must include all parts, ground planes, fiducials to show the alignment of the board, and must pass all DRC checks in the file Dirty8TS.dru. On the back of the board the board must have a name, date, version number, and the Git repository link printed on the back of the board.

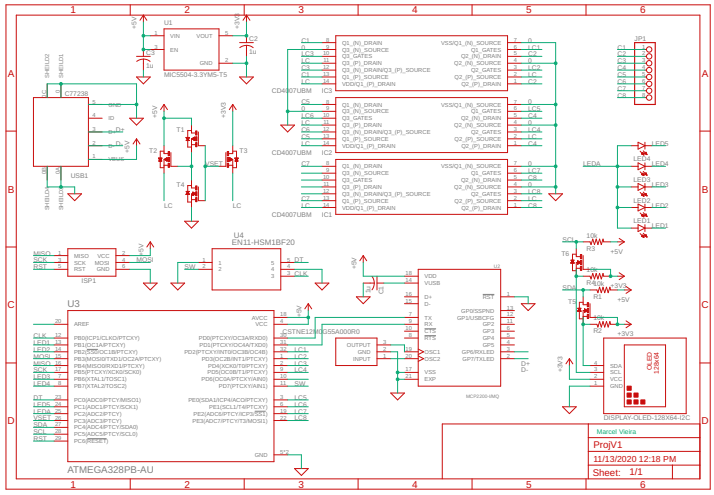


Figure 7: Schematic V5

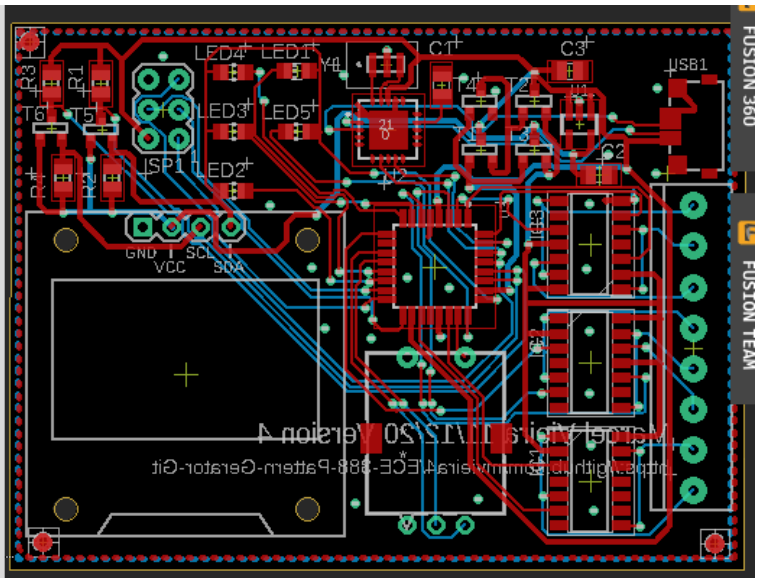


Figure 8: Layout V5

After the board received approval from Tim Chase the Gerber files were sent to JCL PCB to design create the board. Now that the boards were being manufactured an acceptance test is needed to determine whether or not the end result performs to a desirable standard. In this case the test will be how well does the device hold up to the problem statement. Does it have 8 independent channels that can be changed, does it have a UI that makes sense, and can it switch voltages? This will be tested using an a multimeter and a digital discovery which can read voltage patterns in both 5V and 3.3V.

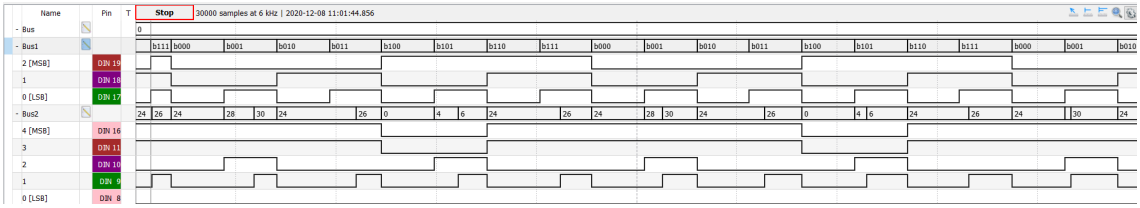


Figure 9: Schematic V5

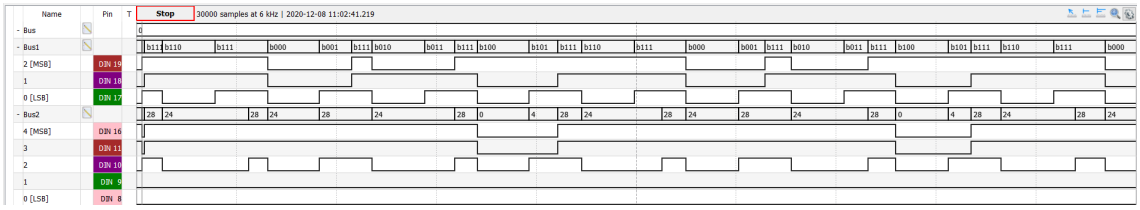


Figure 10: Layout V5

Because of complications with the manufacturer the stencil arrived in time but the PCBs did not making board population impossible. To work around the this a fully functional prototype was created that contains similar parts to show how the board was intended to work. After it is build test the prototype in place of the final board.

4 Discussion

Originally the circuit was going to use a voltage divider to drop the voltage to 3.3V, and N-type MOSFETS to toggle to logic voltages on and off but they presented several issues in the Multisim simulations. For the voltage divider applying the 3.3V to a load changed the resistance on altering the voltage and because the resistance was going to be difficult to predict the Zener diode was used. For the N-type MOSFETS they worked perfectly but each of the MOSFETS was there own chip, but after a suggestion by Tim Chace CMOS inverters were used instead which proved to be just as affective and required 3 times less chips saving space. These test processes lead to the revision of the bill of materials 3 times.

For prototyping all test ran as expected proving that all systems individually are operational. This is with the exception of the linear regulator, the MCP2200 serial chip, and the OLED display. For the display and serial chip they have not come in time to be properly tested, but the voltage regulator was tested and failed. The failure at first was due to the wrong chip being supplied without a part number which meant that the wrong data sheet was used to wire the chip. This in turn damaged the chip and because there was only one further tests could not be done. An attempt was made to test a complete prototyping but there were complications with the 328pb which made it unable to be identified by any computer. This resulted in the 328pb not being able to be programmed. This caused the loss of another test session. Fortunately Ben Vial was able to repair the board so even if the issue occurs again there is a know solution for the problem now.

In creating the first schematic a new USB and an OLED were added to the schematic that originally only contained parts. After wiring the schematic several issues presented themselves.

First the resonator that was attached to the previous schematic was the wrong size. The second issue was that there were no power labels connected. Once these corrections were made even more issues appeared. These issues were brought up by Ben Vial that let students know that the OLED display runs on 3Vs while the 328pb runs on 5V in order. To eliminate issues with communication between the 2 chips he recommended incorporate a MOSFET level shifter. After fixing this the USB had to be replaced with one that had a similar foot print to what was in the stock sheet.

Once the schematic was all set all focus was paced on completing the layout. The issues that were worked on for the layout focused more on labeling and having correct foot prints for all components. Fiducials were also added to the layout to indicate proper orientation. While replacing one of the components in the layout it disappeared but still appeared in the schematics the rest of the time working on the boards were centralized around attempting to resolve the issues. After the third attempt the entirety of the board layout was populated by the correct parts.

After the board was cleared to send to the manufacturer by Tim the Grerbers were sent out. The stencil came in on time but the supplier sent the stencil but not the PCB. Without the PCBs I could not build and test the finished product. To resolve this problem a fully functional prototype was built in its place.

Due to not having enough time to work with the OLED or serial chip the prototypes UI only used the rotary encoder and LEDs. Although it was not ideal it was still in scope for what was acceptable.

To test the prototype for the demo the first step was to use the digital discovery to show the pattern it generated.The first 3 channels in the demo were used to number and keep track of the order of each bit in the pattern.

For the 5V setting the pattern show was perfect exactly matching what was input. When it came to using the UI that worked exactly as intended. The UI allowed for the changing of channels, changing of the selected channel pattern, and changing of voltage.

For the 3V setting the pattern shown was similar but did not match what was expected. This is hypothesized to be because of the final board being intended to have a 3.3V regulator while this

prototyping uses 3V. This presents an issue because the logic range for the digital discovery is from 3V to 5V which means that because there wont be a consistent 3V.

5 Conclusion

For the bill of materials it is completed but it still may change because no solution has been found to electronically switch between the sources being used for the logic levels. There are still issues with the voltage regulator and a full prototype replicating the entire system must still be tested. The schematics and board layout have been completed and have been approved.The board design was sent to the manufacturer but the PCBs did not arrive in time. To be able to demo the project still a fully functional prototype was built and tested. Although there were issues with testing the 3V setting the 5V setting worked flawlessly.

6 Reflection

The planning and simulation had no major issues. There only three minor issues. The first was that it was unclear early on what the pattern generator was and what it was supposed to do.The second was that simulating the circuits proved difficult at first because the necessary chips were not in Multisim so they had to be built out of equivalents that had to be found in Multisim. The third issue was that no clear way to use the MOSFETS to switch between logic sources was found.

In the prototyping stage one of the major issues was a lack of parts to work with which caused a lot of time to be lost and a lot of confusion. This problem was difficult to avoid due to the current pandemic and the restrictions placed as a consequence. Another issue encountered with prototyping was time management. Due to issues with the regulator a majority of lab time was spent around trouble shooting it. Because so much time was lost the rest of the test were done quickly and pictures were not taken. This is why there are no images of the first and second prototyping stages.

The process of creating a final schematic and board lay out came with unexpected issues. One of the major issues was communication. Due to the time constraints being so tight it was difficult to make changes and get the revised quickly. The biggest issue was due to a lack of understanding of what was on the write up that detailed the requirements of the schematic and layout. To solve

this issue other students were consulted and they were able to clarify what was needed. In the future other students will be contacted sooner in the event of confusion.

Creating the final board had many issues. One of the first issues was that the PCBs did not arrive on time. This was due to not getting the board design out in time. This happened because the Github repository did not clearly show which design was the final design. This caused confusion and lead to the board getting approved later than it should have been. To improve this it would have been best to better label the final design version. Had the PCBs and stencil arrived sooner the pick and place would have been used to populate a majority of the board and the OLED and pins would have been hand soldered.

Because the PCB did not arrive a fully functional prototyping was built but this prototype had a 3V regulator instead of a 3.3V regulator. This meant that the digital discovery used to test it could not accurately predict highs. Had the final PCB been constructed this would have most likely not have been the case. This could have been resolved by using an oscilloscope or having a 3.3V regulator.

For the project in its entirety the major issues had to due with a lack of time. This was because too much time was spent in the early stages of design and the equipment that was needed to test the design were on campus. This could have been resolved with more of a realization of time sensitivity due to the pandemic.

Communication also posed and issue which drained plenty of time as well. One major solution to this was to take more time and talk to more students about the formats required for the Github. Also there was an issue were Github notifications were being sent directly to spam which could have been avoided by simply checking spam and making sure that all communications went through efficently.

7 Appendix

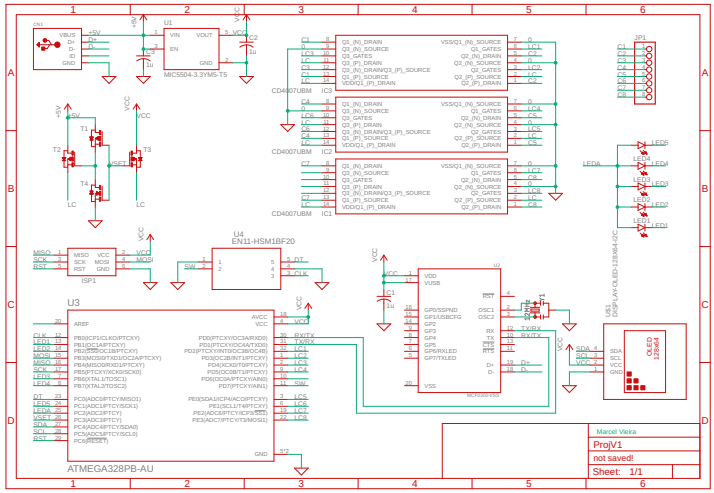


Figure 11: Schematic V1

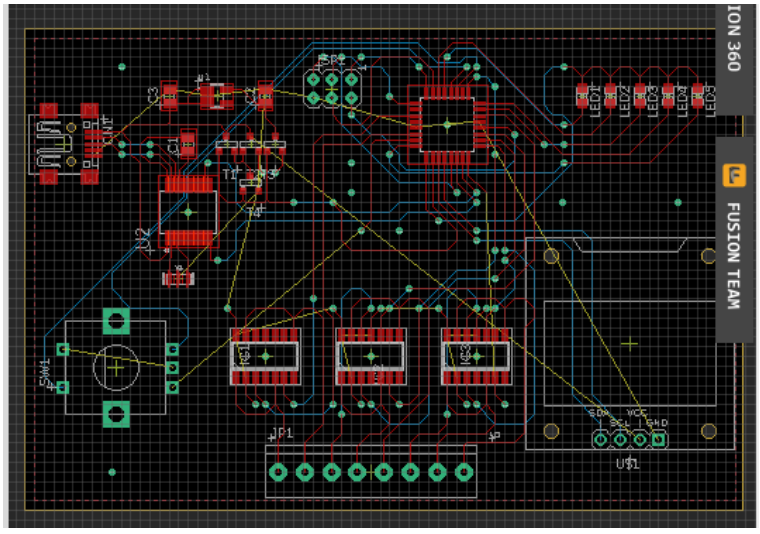
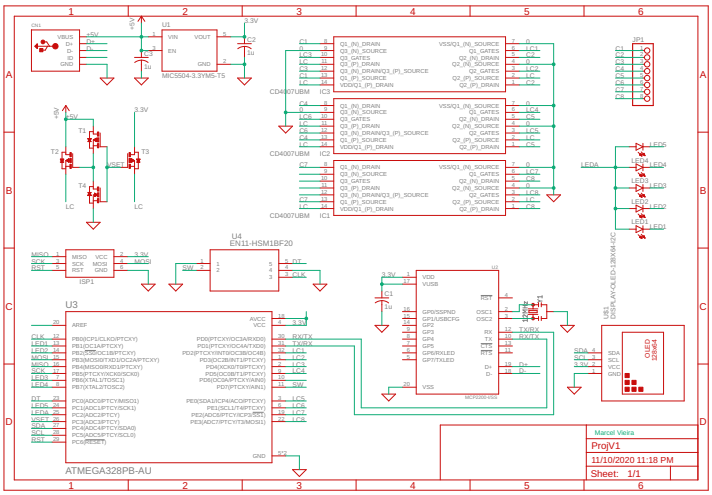
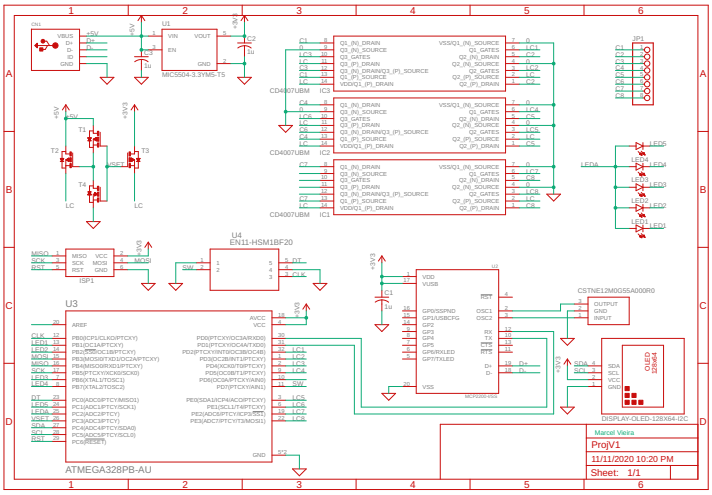


Figure 12: Layout V1



11/10/2020 11:33 PM C:\Users\pphantom\OneDrive\Umass\ECE-388 Work\Lab\ECE-388-Proj-V1\ECE-388 Pattern Gerator Git\ProjV1.sch (Sheet: 1/1)

Figure 13: Schematic V2



11/12/2020 7:11 AM C:\Users\pphantom\OneDrive\Umass\ECE-388 Work\Lab\ECE-388-Proj-V1\ECE-388 Pattern Gerator Git\ProjV1.sch (Sheet: 1/1)

Figure 14: Schematic V3

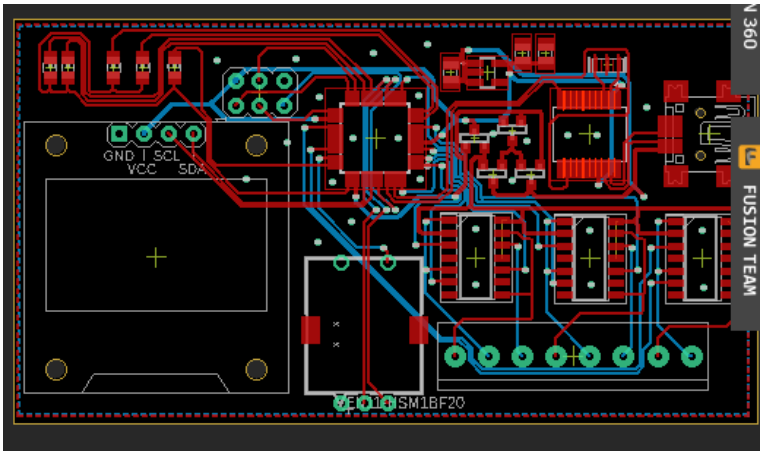


Figure 15: Layout V3

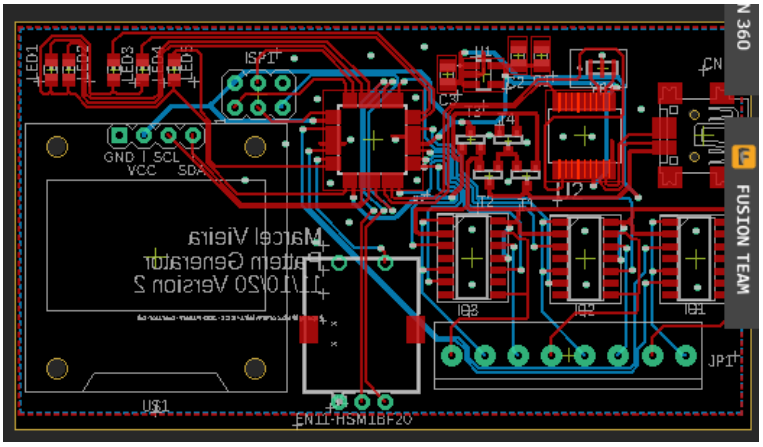


Figure 16: Layout V4