Integrated Semester Project

Matthew Bachelder
ITIS 5358 - Physical Computing

Table of Contents

Back to Table of Contents

Section 1: Planning and Design

Original Plan

The original plan for my Integrated Semester Project was as follows: A weekly calendar (Sun-Sat) for habit tracking, with a 3D printed case, a laser cut calendar, and a circuit for time/date display and task tracking. For the most part, the project itself turned out just like that. Specific issues that came up between planning and implementation will be discussed in their respective sections later in this document.

Tinkercad (circuit design)

Because I didn't have a set-in-stone plan for the circuit design, I experimented a little with Tinkercad. The very first thing I did for my project was create the Tinkercad circuit. I did this because I needed to know what parts I would need for the physical circuit, as well as how my 3D model and laser-cut design were going to look. After looking through the options, I came upon the PCF8574 8-port I2C Port Expander chip, which was exactly what I needed for both my undergraduate and graduate projects.

Using this chip, I was able to have the full functionality of my original design using only 2 analog input pins on my Arduino. This left me plenty of space for both digital and analog inputs, as well as digital outputs such as the LCD that I was planning to use. In the circuit, I use one PCF8574 with the address of 0x20 to determine which light should be on for the day of the week. I then use a second PCF8574 with the address of 0x21 to determine which tasks should be lit.

The selection of tasks is done with two key inputs: a potentiometer and a button. The potentiometer is mapped from 0-1023 to 0-4 to select a specific task, with the selected task being output to the serial monitor. The selected task can then either be turned on or off using the button, which also changes which task light is on using the 0x21 PCF8574.

Lastly, the LCD screen was programmed to display the current date and time. The date is displayed in MM/DD format, with the time in HH:MM AM/PM format. Both date and time are configured to include leading zeroes if they don't fit the format (e.g., 01/07 for January 7th). After finishing up code for all of these functionalities, I was ready to plan and design my 3D model and SVG.

3D Printing Plan

The original plan for the 3D printing of this project was to create a full case that surrounds the electronics, hides them in the back, and supports the calendar. In the back, there would be snaps or rails to hold the Arduino and require breadboards

Back to Table of Contents

in place, as well as a place to hold the battery. In the front, the idea was to have a hole where the calendar would go and a hole where the LCD and Arduino LED array would go. This entire print would have been standalone and may have included a place for hanging on a screw.

Laser Cutting Plan

The original plan for the laser cutting of this project was to create a week calendar, which starts on Sunday on the left and ends on Saturday on the right. Each day would also have a small hole to let an LED through to show what day of the week it currently is. Below the 7 days would be the task area, where each day has a small square to write down tasks, with each area having 5 holes for 5 LEDs (42 total LEDs: 7 for days, 35 for tasks).

Section 2: Creation and Testing

Laser Cutting Implementation

The first thing I actually implemented after creating the Tinkercad circuit was the laser-cut calendar. I designed this to use 3 in x 3 in sticky notes as the piece to hold the tasks for each day. Another thing that ended up affecting the plan for my laser-cut calendar was the LEDs that I would use. After deciding that my circuit would only have a total of 12 LEDs (7 for days, 5 for tasks), I had to make the task LED holes only along the left side.

After creating the SVG, it barely fit within the 24 in x 18 in space for the Epilog laser cutters. I tested out the laser cut on cardboard twice, and both times it was weirdly cut off at about 18 inches, which I believe was a printer issue with dimensions. Just in case, I reduced the used space in the SVG, and it fit within 23 in x 5 in, which fits in both laser cutters. I ran a third test on cardboard to test swapped dimensions (24 for vertical, 18 for horizontal, which was swapped originally) to see if it would still cut off at 18 inches.

3D Printing Implementation

Before I created the model for this project, I had to know what my laser-cut calendar was going to look like. I designed that before I modeled anything for 3D printing. After creating the SVG, and seeing that it would come out to be about 23 inches long, I knew that my 3D print would have to be either printed in pieces or not encase the entire calendar. I decided to pivot on my 3D design to print stands to hold the calendar upright, making it a desktop calendar display. This would not encase any of the electronics, but only hold up the calendar.

Back to Table of Contents

I printed out a stand successfully and tried it out with some test pieces of wood, and it worked great. My print was ready before my laser cut, so when my laser cut was finally ready, I tried the stand with my calendar, and it worked perfectly. The fit was a little tight, but that helped keep the calendar in.

Arduino (circuit)

Before setting up the circuit with the calendar, I assembled the circuit according to my Tinkercad design to test it out before integrating everything. A major difference I had to figure out before I combined the circuitry with the physical pieces was connecting to WiFi to receive Unix time from an NTP server using UDP. To solve this portion of the problem, I reviewed the Arduino documentation for Real Time Clock (RTC) information, which provides solutions for retrieving RTC data from an NTP server after connecting to WiFi.

After following the schematic from Tinkercad, implementing the beginner code from Tinkercad, along with the changes for the Real Time Clock, the circuit was working properly. I did plenty of testing before combining the circuitry with the physical pieces, and it worked very well. I did put more serial monitor writes in the code for my Arduino than I had on my Tinkercad design to assist with debugging. I also used the LED matrix on the Arduino Uno R4 to display what task (numbered 1-5) was selected to toggle on or off.

Section 3: Conclusion and Outcome

To combine the circuitry and the wood, I put tape on the back side of the wood and scored the holes for the LEDs. When cutting the wood originally, I tested an LED in the LED hole, and it was fairly loose, which didn't hold the LED at all. My dad suggested I use scored tape to act as a shim to hold the LEDs in, which worked perfectly. I also taped the 800-port breadboard that contains both PCF8574 chips and all of the logic for the lights to the back of the wood. The logic for the potentiometer, button, and LCD was all put on a separate breadboard that connects to the Arduino.

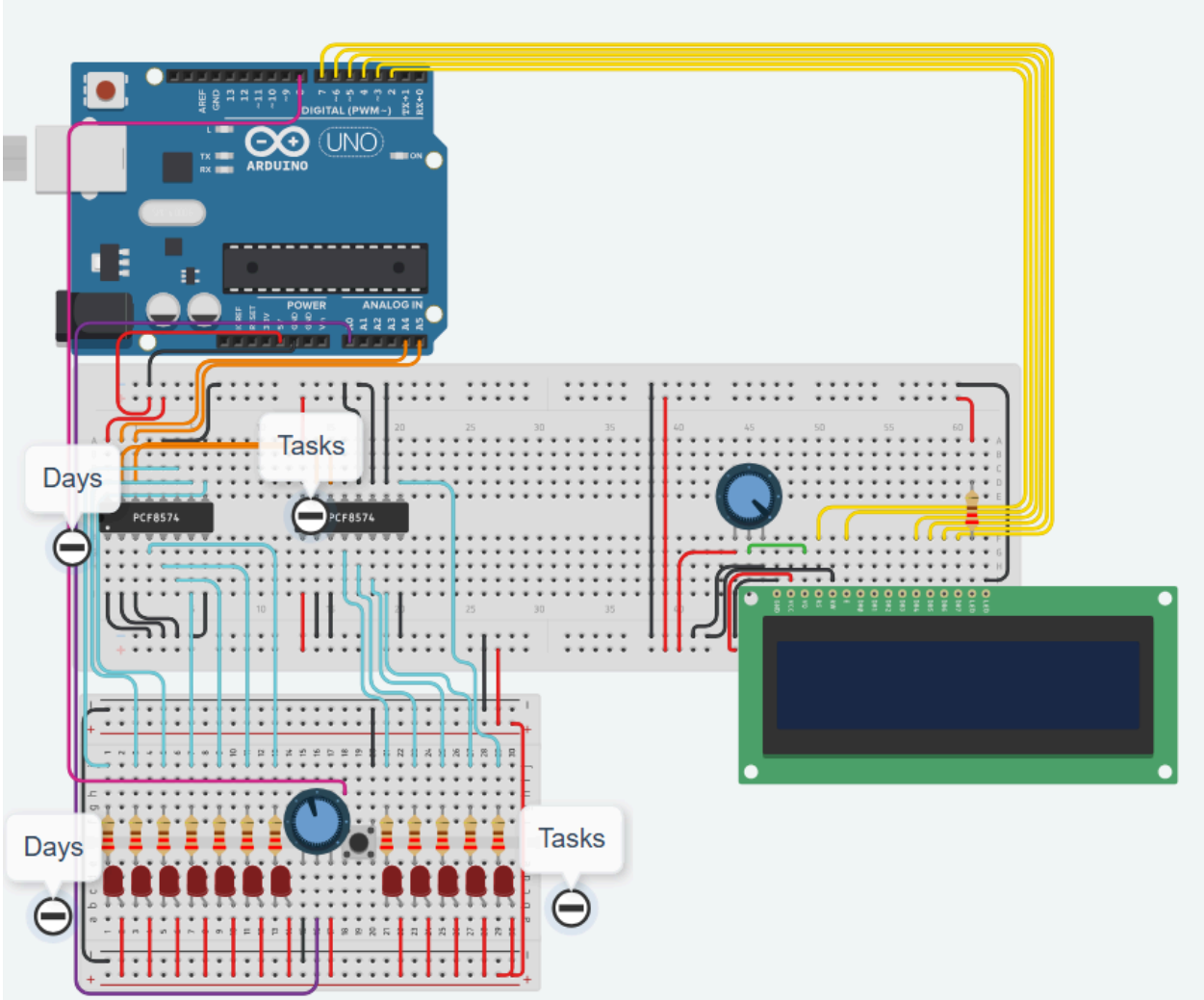
I presented my project on Monday, 12/1, during the normal class time, and I was the first person to present. My presentation went well, and I had one student come up to me to talk about my project before class, and a couple after. Overall, I was very happy with the final implementation of my project.

There are plenty of improvements that I could make to my project. For example, I could reduce the size of the printed calendar. I could also come up with a better way to display actual habits or tasks over sticky notes. A big improvement for code is creating 7 boolean arrays to hold the individual day's tasks completion, so that I could scroll through the week and see what I was able to accomplish or not.

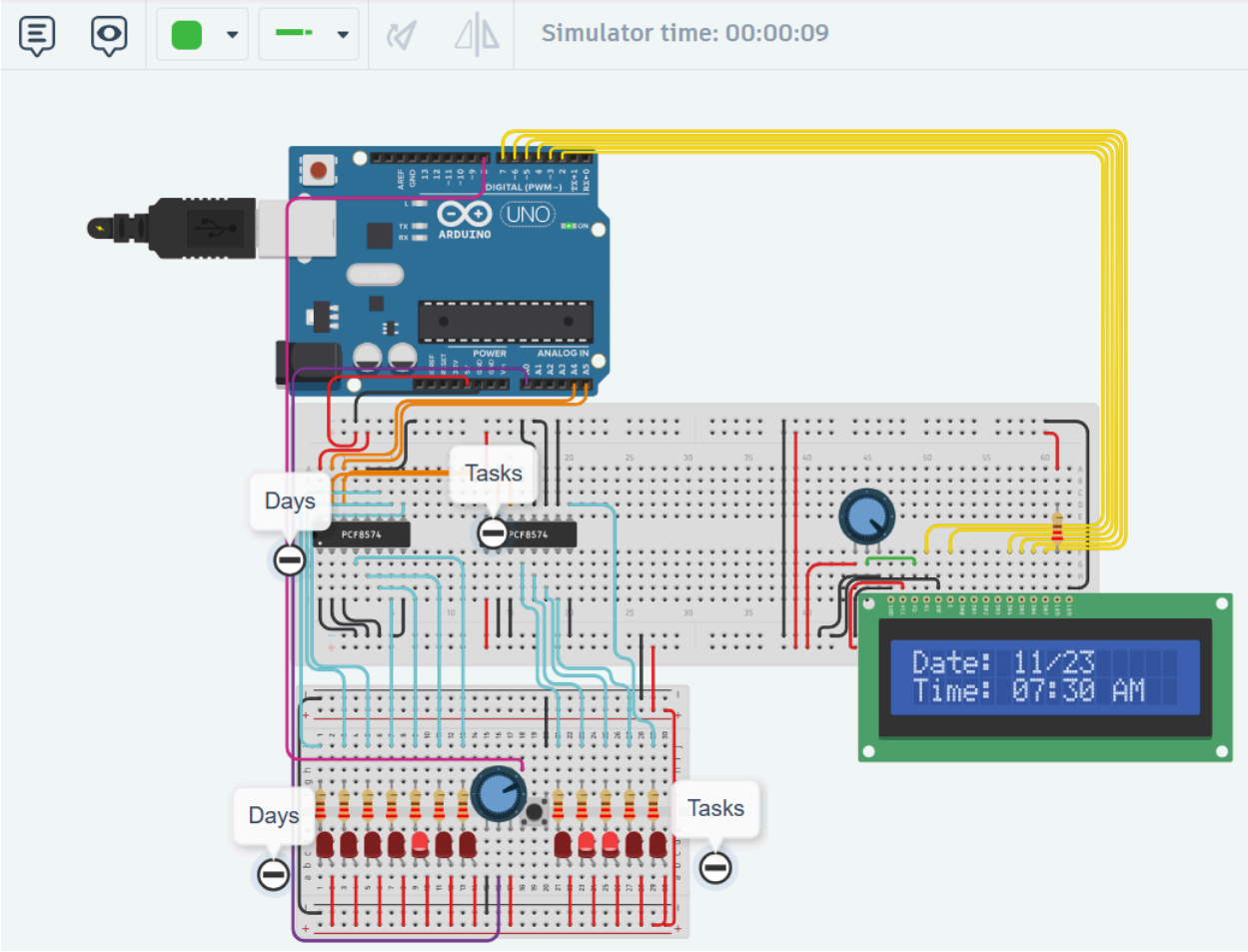
Back to Table of Contents

Section 4: Pictures and Screenshots

Tinkercad Design Screenshots



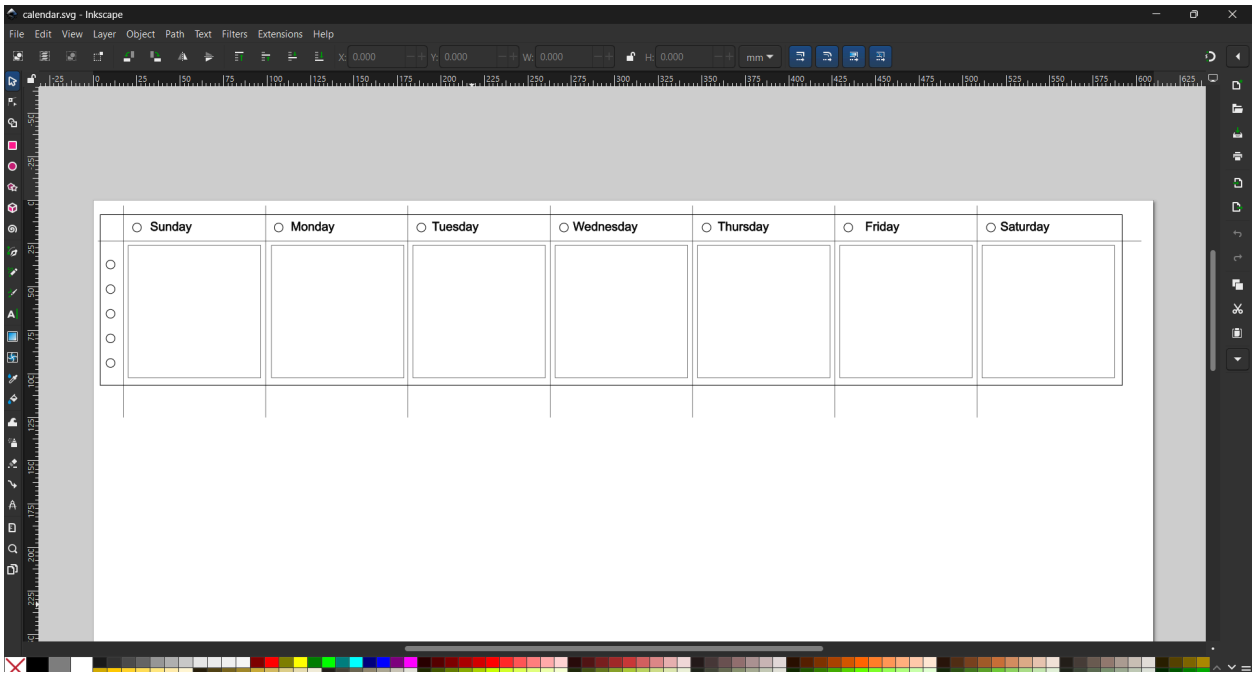
Back to Table of Contents

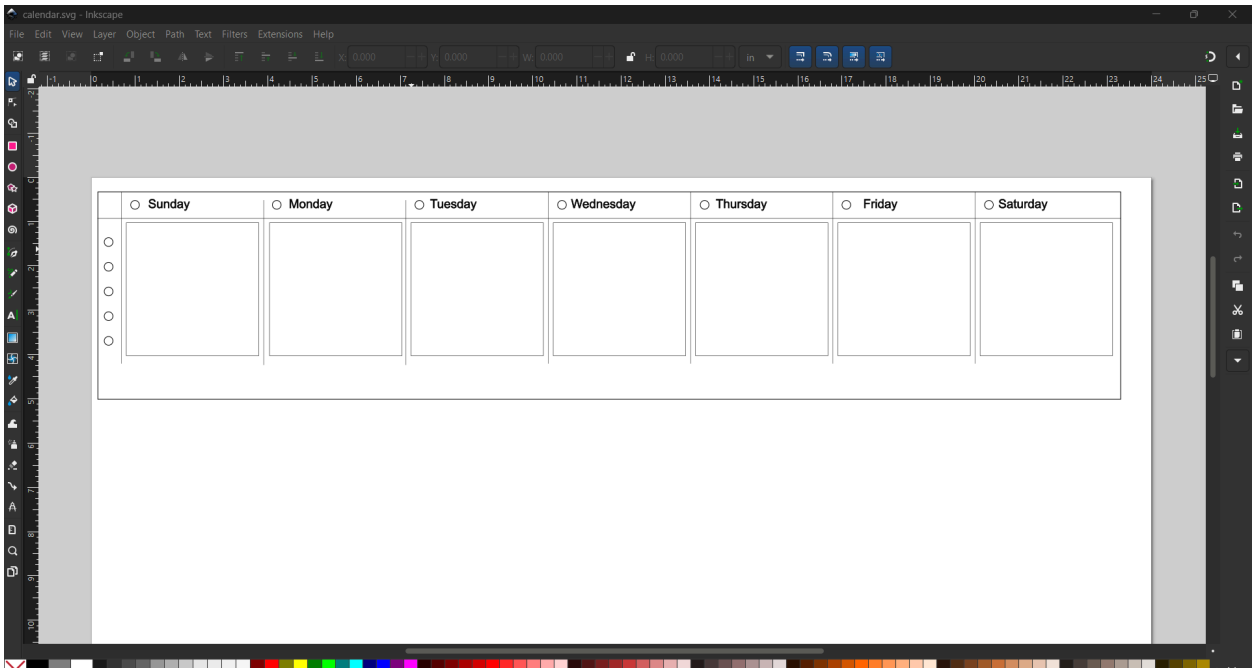


3D Printing Pictures and Screenshots

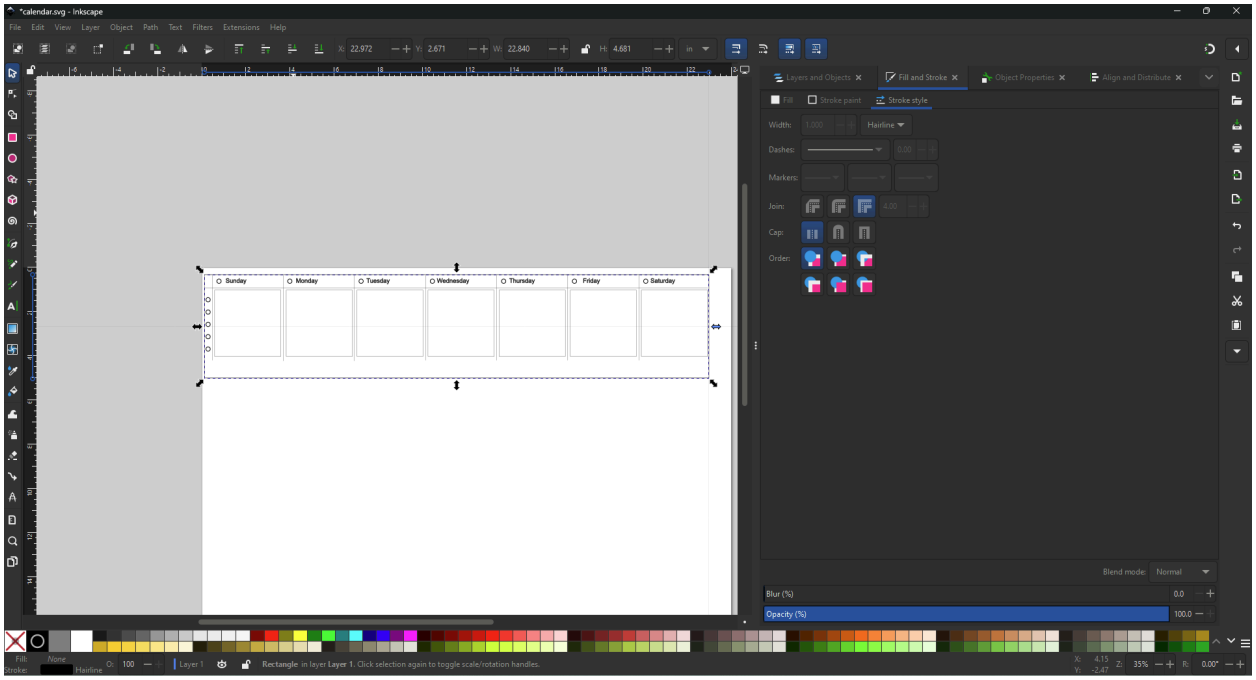
Back to Table of Contents

Laser Cutting Pictures and Screenshots





Back to Table of Contents



Arduino Circuit Pictures and Screenshots

Back to Table of Contents