Welcome to the OMERS Word Clock!

View all relevant code and design files at: https://github.com/jdolgin-OV/word-clock

## About Word Clocks

A word clock is a device that tells the time by lighting various letters or words to explain the time using common phrases, such as “it is a quarter to two” rather than a digital clock displaying “1:45”.

There are many different iterations of Word Clocks, each with different letter positions, hidden features, and granularity of time being displayed. For the OMERS Word Clock, time is displayed with a resolution of 5 minutes, starting with O’CLOCK, then continuing to FIVE PAST, TEN PAST, …, TEN TO, FIVE TO. The clock measures out to be 42.5” x 42.5” and is the largest DIY Word Clock I have seen. It measures 18cm larger than the QLOCKTWO Large, which retails for 9,000 – 15,000 USD.

The OMERS Word Clock is a 12x12 matrix of letters, with each letter backlit by 4 LEDS, using 576 LEDs in total.

| I | T | L | I | S | A | S | I | P | C | O | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | C | Q | U | A | R | T | E | R | D | M | C |
| T | W | E | N | T | Y | B | F | I | V | E | L |
| H | A | L | F | C | R | Y | T | E | N | R | O |
| M | I | N | U | T | E | S | E | T | O | S | C |
| P | A | S | T | J | D | F | O | U | R | T | K |
| S | E | V | E | N | O | T | W | E | L | V | E |
| N | I | N | E | F | I | V | E | C | T | W | O |
| E | I | G | H | T | F | E | L | E | V | E | N |
| S | I | X | T | H | R | E | E | O | N | E | T |
| T | E | N | D | O | T | O | C | L | O | C | K |
| I | N | C | T | O | R | S | F | O | L | O | N |

The layout of letters and words is customizable, allowing for hidden messages to be included. I have highlighted some of the hidden words above. “OMERS CLOCK” is lit upon start up, as well with the initials “JD” and “CR”, referencing the clocks 2 creators Josh Dolgin and Casey Rovinelli. “IP” is used to display the clocks local IP address once the device connects to a local network.

One thing to keep in mind is the hierarchy of the words, so that the time is legible when read top to bottom. Meaning that IT must precede IS, since the clock needs to read IT IS. In order of appearance from top right to bottom left:

1- IT
2- IS
3- Minute denominations (FIVE, TEN, QUARTER, TWENTY, HALF), these can appear in any order, with the exception that TWENTY must appear before 5 to read TWENTY-FIVE.
4- MINUTES
5- TO/PAST
6- Hour denominations (ONE, TWO, …)
7- IN
8- Time zones

Here is some of the digital to word time conversions:

12:00 -> IT IS TWELVE OCLOCK
12:05 -> IT IS FIVE MINUTES PAST TWELVE
12:10 -> IT IS TEN MINUTES PAST TWELVE
12:15 -> IT IS A QUARTER PAST TWELVE
12:20 -> IT IS TWENTY MINUTES PAST TWELVE
12:25 -> IT IS TWENTYFIVE MINUTES PAST TWELVE
12:30 -> IT IS HALF PAST TWELVE
12:35 -> IT IS TWENTYFIVE MINUTES TO ONE
12:40 -> IT IS TWENTY MINUTES TO ONE
12:45 -> IT IS A QUARTER TO ONE
12:50 -> IT IS TEN MINUTES TO ONE
12:55 -> IT IS FIVE MINUTES TO ONE
1:00 -> IT IS ONE OCLOCK

Once the layout is decided, the size of the clock and letter grid will need to be chosen, as this has a huge effect on overall cost. When using an LED strip at 60 LED’s / m, each LED takes up roughly 0.657 in or 1.667 cm. Since each letter will take up a square area, using 1 LED / letter means the grid would be 20x20 cm, 2 LED’s / letter would be 40x40 cm, 3 LED’s / letter would be 60x60 cm, etc. The OMERS CLOCK uses 4 LED’s / letter, resulting in a 80x80cm grid of letters.

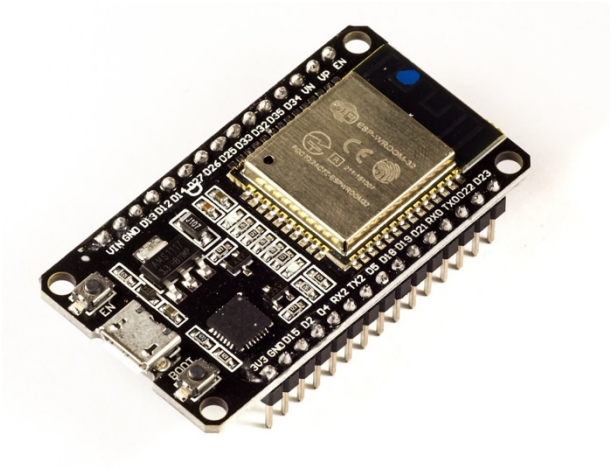
## Cost Breakdown

| Component | Price (CAD) | Quantity |
|---|---|---|
| ESP32 Module | $ 18 | 1 |
| Capacitive touch buttons | $ 6 | 4 |
| DS3231 RTC | $ 20 | 1 |
| 5 m 60 LED/m WS2812 LEDS | $ 60 | 2 |
| 18 AWG hookup wire | $ 19 | 1 |
| 5V 30 A Power Supply | $ 40 | 1 |
| Frosted window film | $30 | 1 |
| 1 Kg Black PLA | $ 30 | 2 |
| CNC machined parts | $ 1000 | 1 |

## Hardware Components

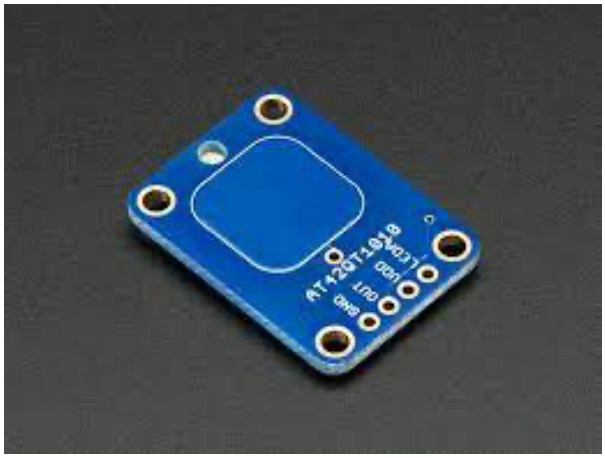
The OMERS Word Clock uses 4 main hardware components, each with a specific design purpose.

The ESP32 is a WIFI / Bluetooth enabled microcontroller that can be programmed using an Arduino C Environment. This board was selected such that the Clock modes could be configured over WIFI, allowing for more versatility than just manual inputs. Additionally, with a Clock Frequency of up to 240 MHz, and 4 MB of flash memory, the ESP32 is perfectly suited for the OMERS Word Clock. As the ESP32 has grown in popularity, various additional libraries have been made available that leverage the WIFI and speed capabilities of the device.



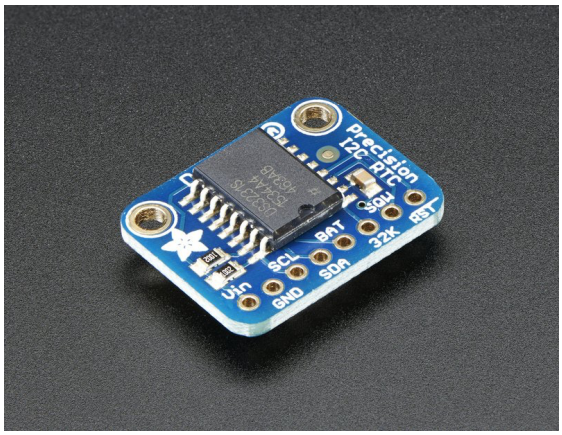
ESP32 Microcontroller

The OMERS Word Clock uses 4 of the Adafruit at42qt1010 momentary capacitive touch button breakouts to interface with the device without using WIFI. One will be used to change the modes of the device, one to change the time zone of the clock, and the remaining buttons are used to set hour and minute of the clock.



Momentary Capacitive Touch Button

The Adafruit DS3231 Real Time Clock (RTC) module is designed to precisely keep track of time. It uses a coin cell battery so that it can continue to keep track of the time even when the clock is powered off.



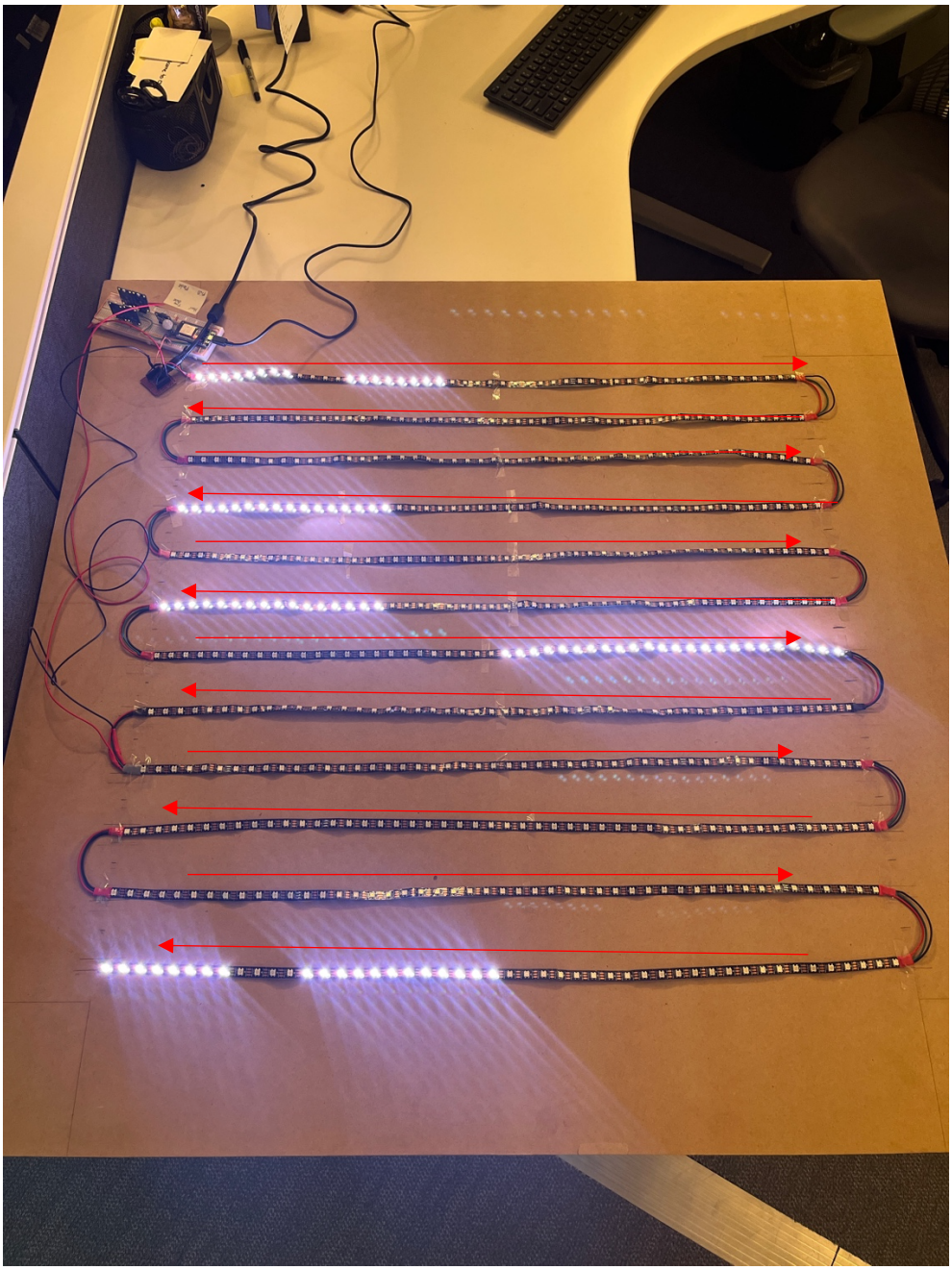
DS3231 RTC Module

The OMERS Word Clock uses 576 WS2812 individually addressable LED strips to light up the letters of the Word Clock. Individually addressable means that each LED can be set to its own color and brightness using software, as opposed to other LED strips where all the LEDS are the same color. This allows for various animations to be displayed.


WS2812 LED Strip

## Assemble the electronics

Cut the 2 reels of LEDs into strips of 48 LEDS. With the data line starting in the top left, solder the LEDS in a serpentine pattern. Make sure that the connecting wires are at least long enough for the strips to fit the 3D printed grid. For 4 LED's / letter, the strips need to be offset 6.667 cm.  Additionally, connect an extra power and ground wire approximately every 5 m of strip, to inject additional power and prevent any color degradation.



Soldered LED wires, red arrows show data path

Connections are as follows:
LED Strip:
- GND -> GND
- DIN -> D14
- VCC -> 5V

Capacitive touch 1:
- GND -> GND
- OUT -> D25
- VCC -> 3.3V

Capacitive touch 2:
- GND -> GND
- OUT -> D26
- VCC -> 3.3V

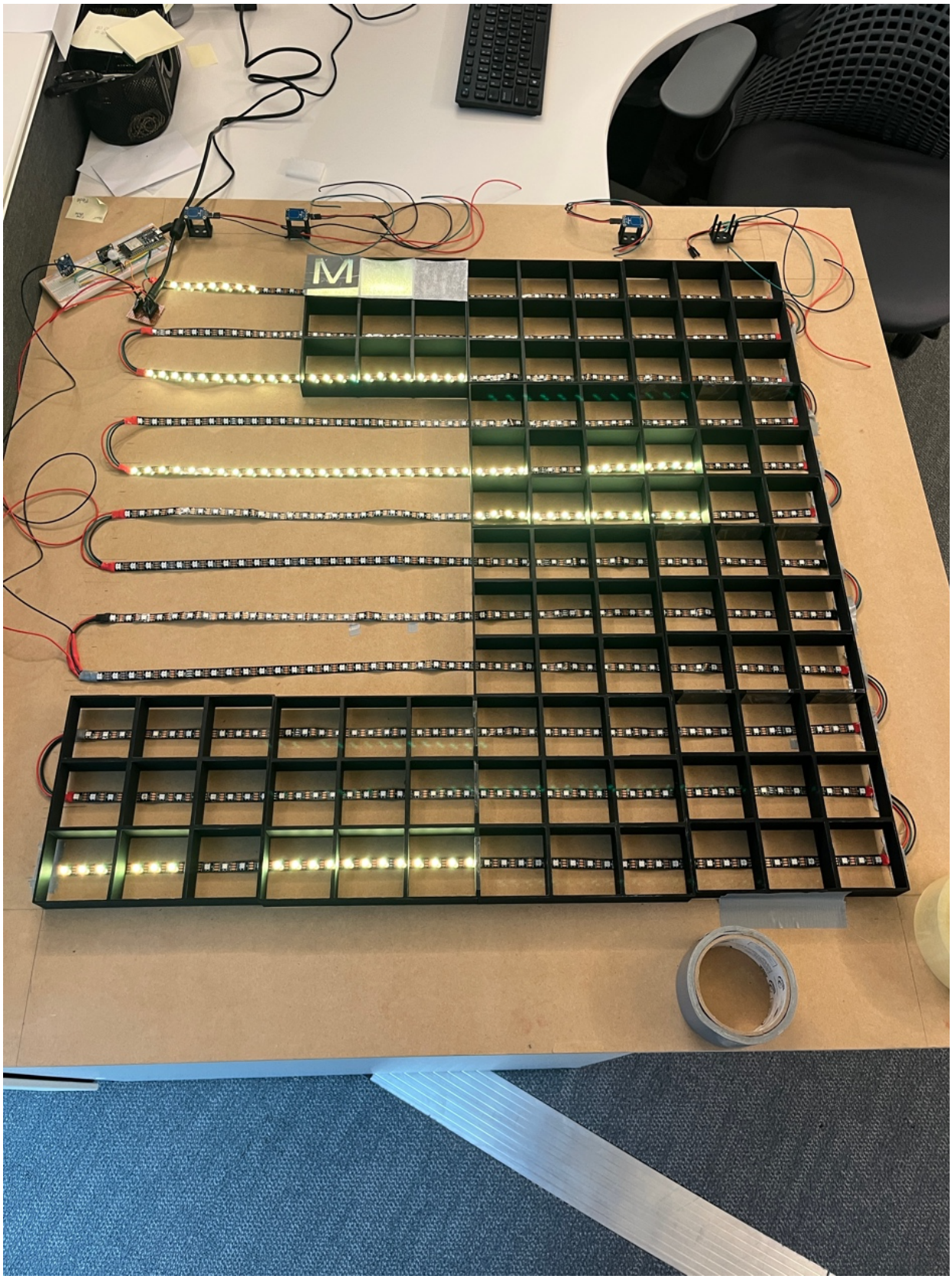
Capacitive touch 3:
- GND -> GND
- OUT -> D27
- VCC -> 3.3V

Capacitive touch 4:
- GND -> GND
- OUT -> D33
- VCC -> 3.3V

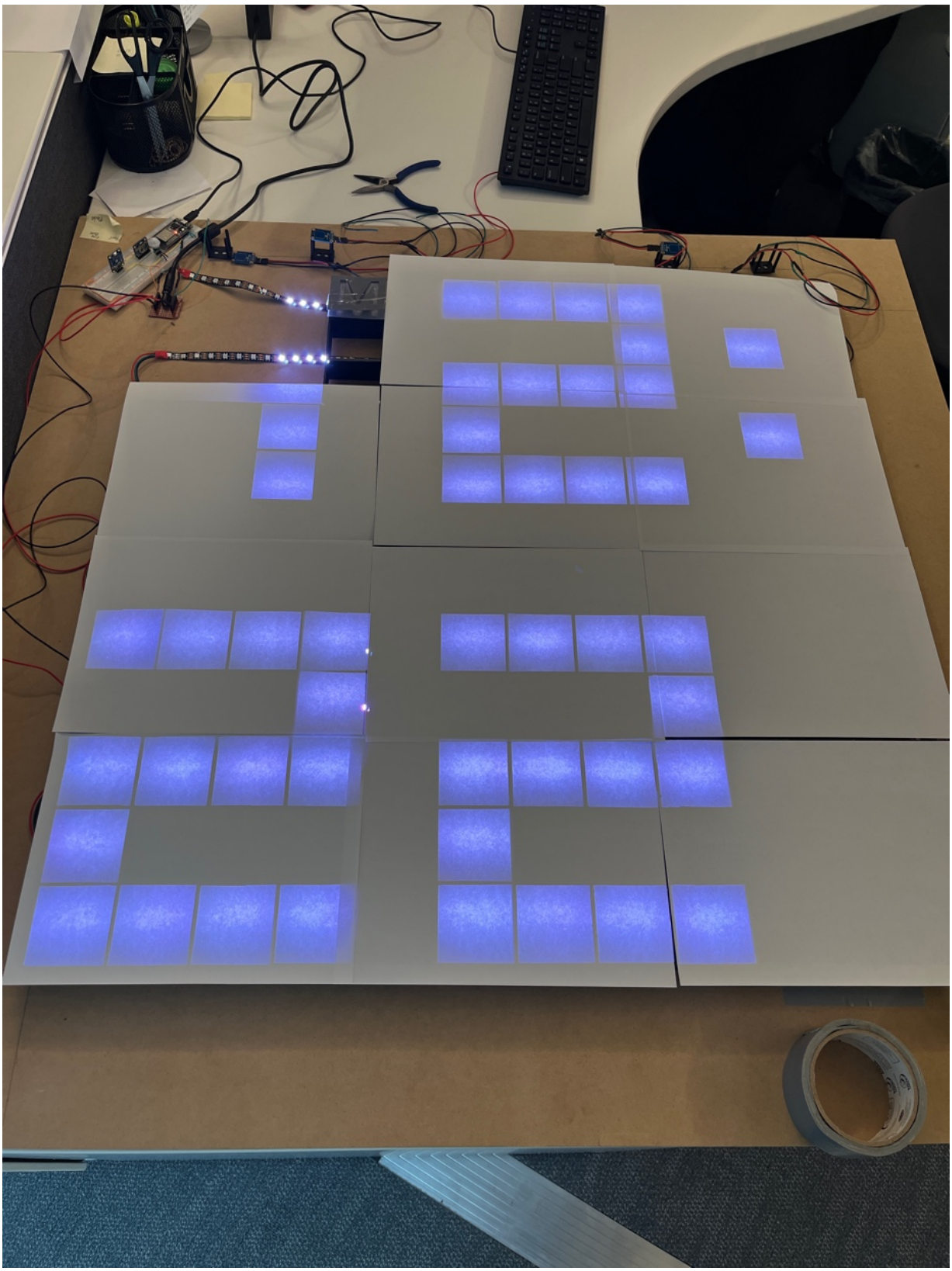
ESP32:
- VIN -> 5V
- GND -> GND

RTC:
- VIN -> 5V
- GND -> GND
- SCL -> D22
- SDA -> D21

After all the soldering is complete, it is time to begin 3D printing the grid pieces and the electronics housings. The .stl files can be found in the github repository. Print 1 of the electronics housings, and 16 of the 3x3 grids. One grid fits comfortably on a Prusa MK3 3D printer with a build volume of 25x22x22 cm.



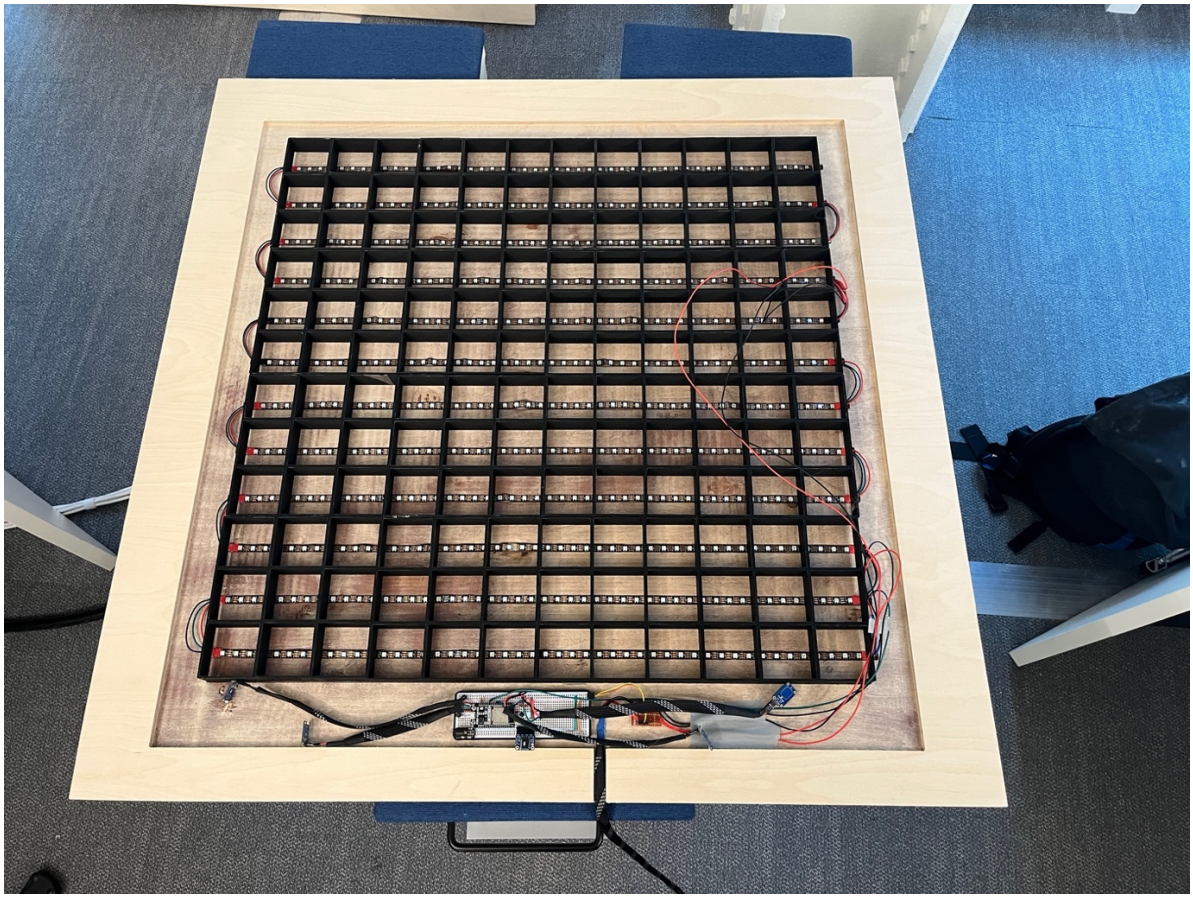
Starting to place the 3D printed grids



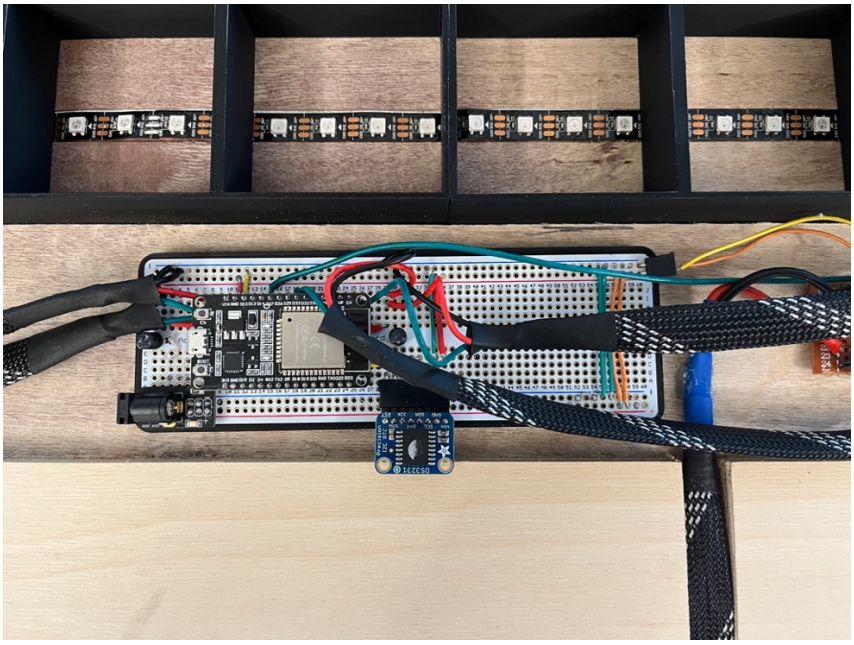
Using paper as a temporary diffuser


Window diffuser placed on top

At this point in the process, it was decided to switch the MDF base for a fully CNC routed body.



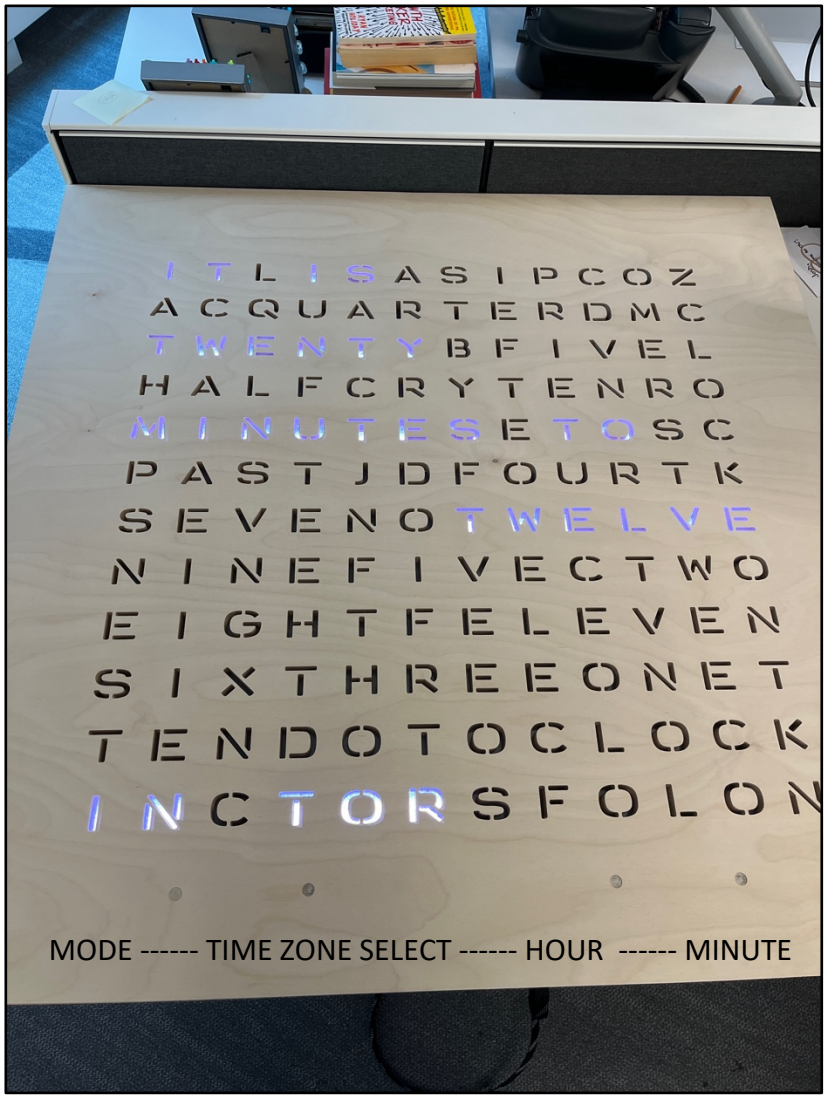
CNC routed base plate with channels for the LED



Soldered electronics

Fully assembled board

**USER GUIDE**




**Manual Interface**

The Clock has 4 buttons located underneath the recessed holes at the top of the board. These buttons detect capacitance stored in your finger and thus require a somewhat hard press. For best results press with your thumb onto the circle and try to have as much skin come into contact with the button as possible. If thumbs don't work, try using your palm. The buttons are, in order:

MODE ------ TIME ZONE SELECT ------ HOUR INCREMENT ------ MINUTE INCREMENT

- The mode button switches to the next mode.

- The time zone select button switches the time zone from Toronto to San Francisco to London, and only works in word clock or digital clock time mode.

- The minute increment button increments the current minute of the clock, and only works in word clock or digital clock mode.

- The hour increment button increments the current hour of the clock, and only works in word clock or digital clock mode.

**Turn Off/On**

- To manually turn the clock off, either unplug it or press both the time zone and hour increment button at the same time. Pressing both buttons can be used to turn the clock off or on.
- To remotely turn the clock off/on, press the power button on the web app.

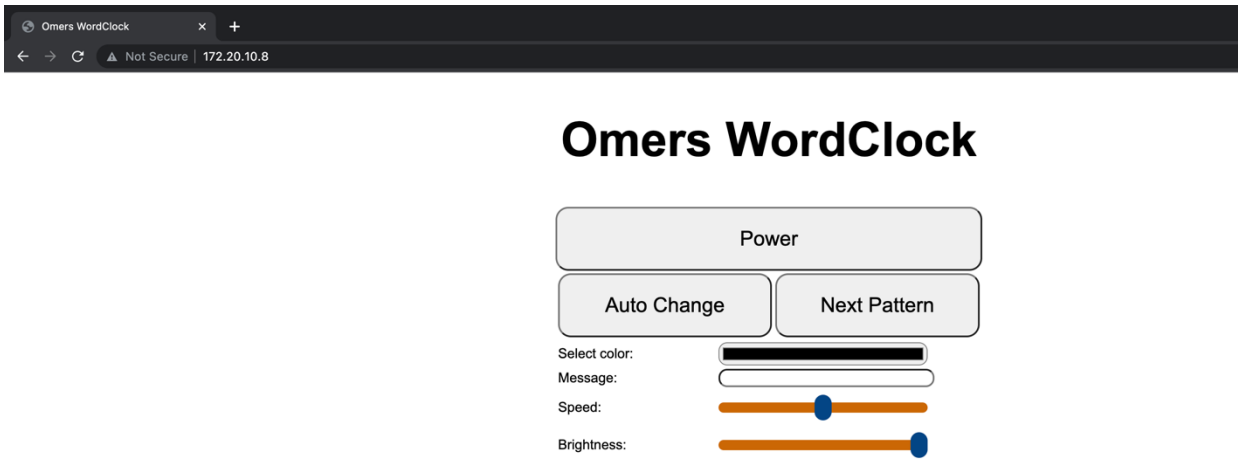
**Modes**

- The modes of the OMERS word clock are Word Clock with solid color -> Word Clock with fading rainbow color -> digital clock with solid color -> rainbow animation -> various LED animations.

**Startup**

- On start-up, the Word Clock shows the OMERS blue and orange circle logo and tries to connect to both the primary and secondary networks 5 times
- Once a network connection is established or the device couldn't connect to a network, the device starts by showing OMERS CLOCK, JD, CR for 5 seconds
- After that, the device will show the local IP address, that allows a user connected to the same network as the device to control the Word Clock.
- The device will show the IP using a word sequence, for instance IP-ONE-SEVEN-TWO-DOT-TWO-ZERO-DOT-ONE-ZERO-DOT-EIGHT means the local IP address is 172.20.10.8

Enter the IP address into the search bar of a browser of your choosing, and you will be met with a page like this:




- The Power button turns the device on and off, the device can be turned on by pressing the button again or by pressing the time zone and hour increment button at the same time on the device
- The auto change button tells the device to automatically change the mode every 30 seconds, pressing again deactivates this feature
- The next pattern button increments the mode of the device, the same as the physical button
- The message field changes the scrolling message
- The speed field changes the speed at which the message scrolls, dragging to the right speeds up the message and the left slows it down
- Brightness changes the brightness in all display modes

**Software Setup:**

- Download the Arduino IDE https://www.arduino.cc/en/software
- In Arduino, go File -> Preferences and paste this link into the “Additional Board Manager URL’s”: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
- Click OK
- Under Tools -> Boards Manager search for ESP32 and install the latest version
- Under Tools -> Board -> ESP32 Arduino Select “DOIT ESP32 DEVKIT V1”

For each of the following libraries, install them by going sketch -> include library -> manage libraries, then search for and install the latest version:

- FastLED, Wifi

For each of the following libraries, download the ZIP library from the github link and install it by going sketch -> include library -> add .ZIP library, then select the folder from the dropdown.

- Adafruit_GFX.h https://learn.adafruit.com/adafruit-gfx-graphics-library?gclid=CjwKCAjwiJqWBhBdEiwAtESPaNnGOo8SvMXSlS327Ipo8hFoow_a29YUybYNr93avlGfNRe0t7xqWhoCr6IQAvD_BwE
- FastLED_NeoMAtrix.h https://github.com/marcmerlin/FastLED_NeoMatrix
- AsyncTCP.h https://github.com/me-no-dev/AsyncTCP
- ESPAsyncWebServer.h https://github.com/me-no-dev/ESPAsyncWebServer

Download all 5 .ino/.h files from: https://github.com/jdolgin-OV/word-clock

**To upload new code:**

- Plug in the OMERS CLOCK to turn it on and provide sufficient power to the circuit. **VERY IMPORTANT**
- Plug your computer via USB-C to the provided port on the back of the clock.
- Plug in the ESP32, under tools -> port it should auto select something including "usbserial"
- In Arduino, click the right arrow icon to compile and upload
- In the bottom right corner is a progress bar of compilation and upload
- Unplug the USB-C cable

**Change the WIFI:**

- In lines 10-15 of wifi_setup.h, there is 2 sets of SSID and Passwords, update them accordingly and re-upload the code
- If the device is not connecting to WIFI, trying to comment out line 303 of wifi_setup.h "WIFI.mode(WIFI_STA);" and try again.
- If connecting to an iPhone hotspot, turn on "Maximize compatibility" to allow 2.4 GHz connections.

**Change Default Values:**

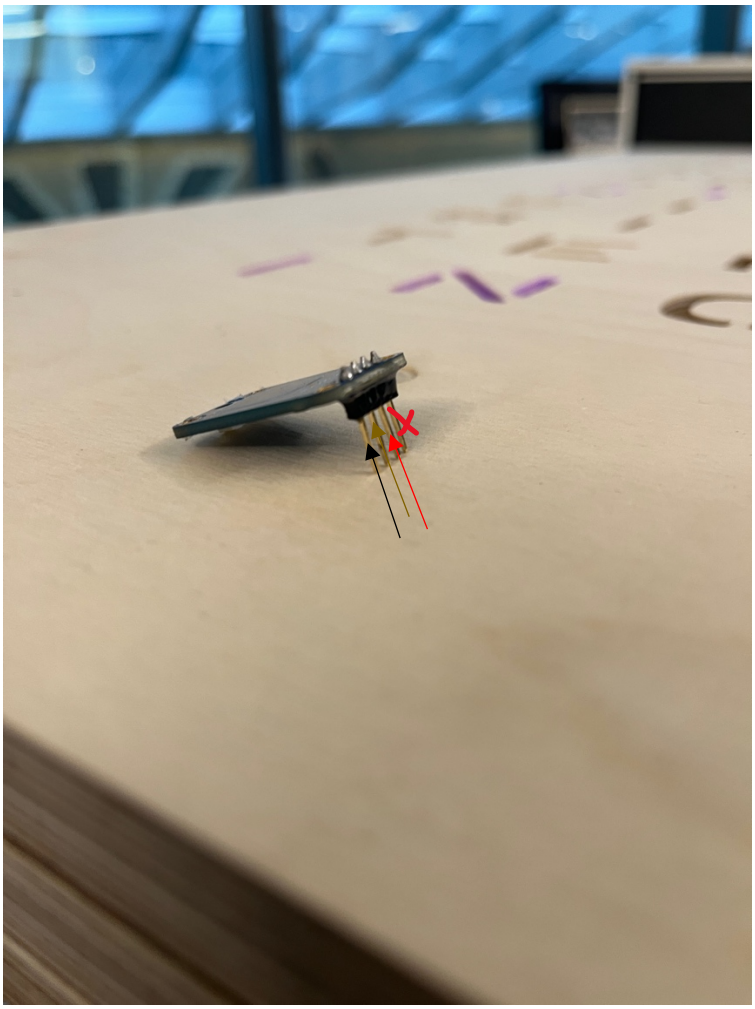
- In lines 158 – 173 of wifi_setup.h, there is all the default values for the clock
- brightness is a value from 0-255, where a larger number is brighter
- red, green, blue are values between 0-255
- text is any string
- scroll_speed is a value between 0 – 500, where a larger number is faster
- clock_mode is the starting mode of the clock
- auto_change if set true will automatically scroll through patterns every 30s

**Disassembly**

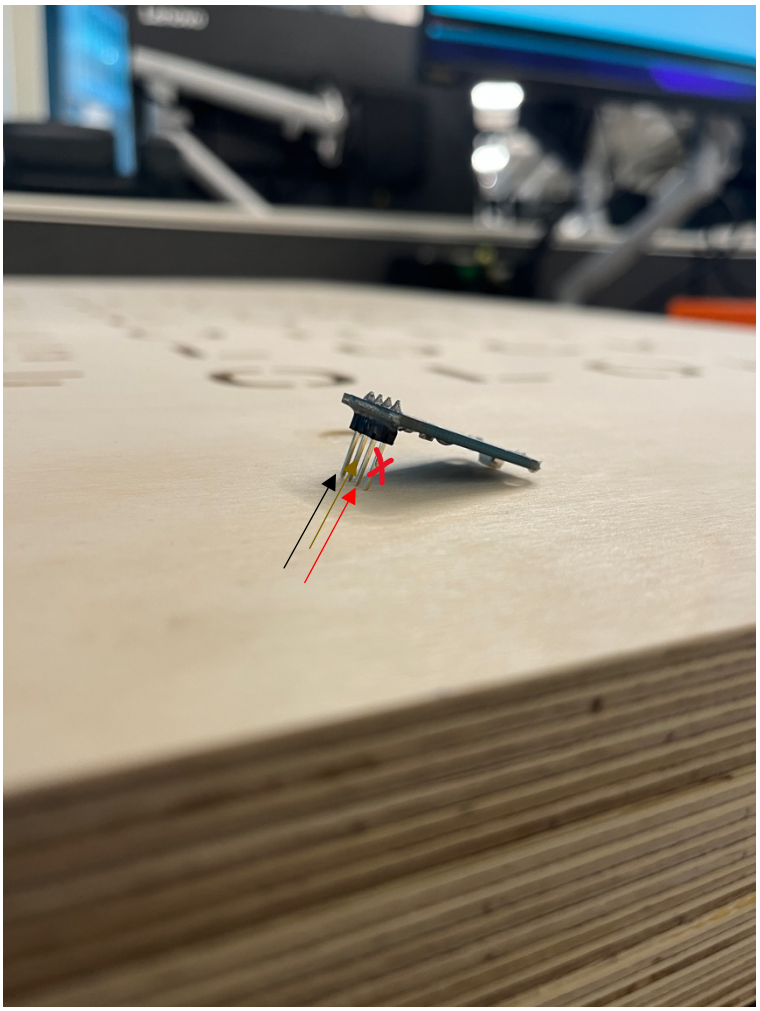
- found on the back of the clock are 8 countersunk bolts, in each 4 corner and along each 4 side.
- use an allen key to loosen and remove all the bolts, order doesn't matter.
- slightly hinge open the face plate, lifting from the side with the buttons.
- lift the face up a couple inches and unplug all 4 buttons by pulling the braided cables down. **MAKE NOTE OF THE CABLE POSITIONS**
- you are now free to remove the face plate.

**Assembly**

- place the face plate on top of the device, align the buttons with the electronics.
- hinge open the face plate and prop it open using a couple books or roll of tape.
- plug in the buttons, the same way as when they were disassembled.
- for the left buttons, the mode and time zone select, plug the cables in with the ground (BLACK) wire into the lead closest to the bottom, and the power (RED) lead into the third lead from the bottom. On the left buttons you should see black wire, brown wire, red wire, exposed pin.

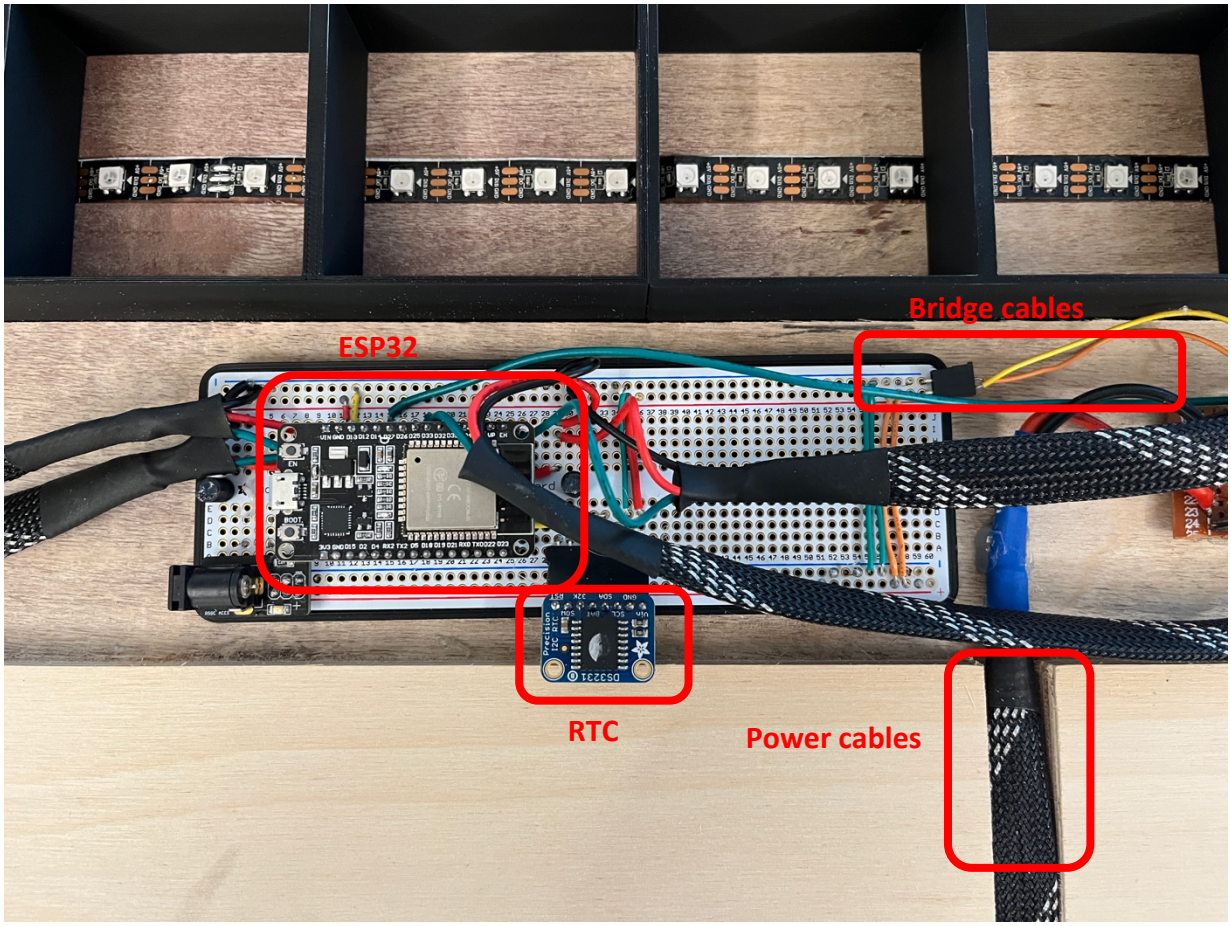


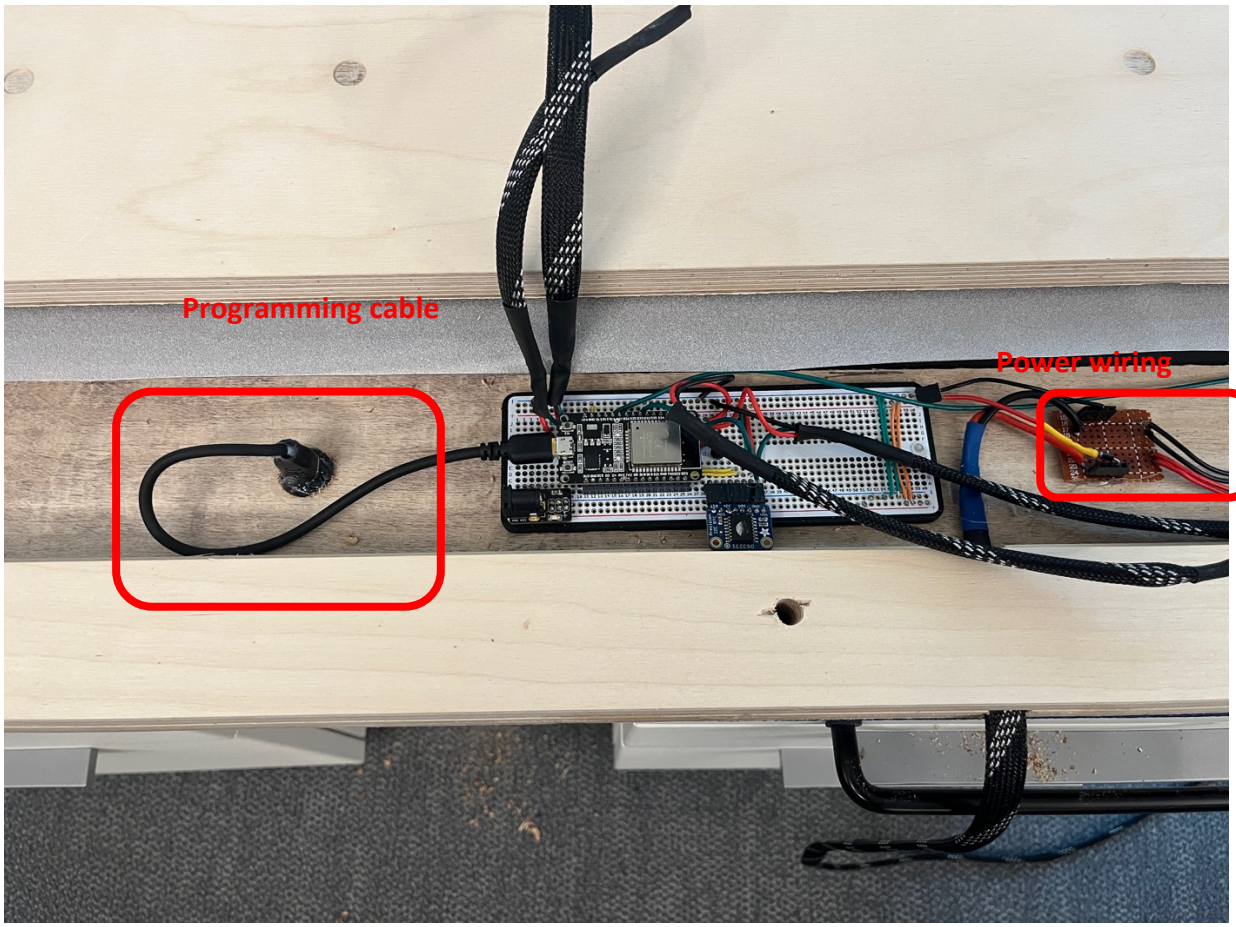
- For the right buttons, minute and hour increment, plug the cables in with the GND (BLACK) lead into the pin furthest from you, and the voltage (RED) lead into the pin second from the bottom. On the right buttons you should see exposed pin, red wire, brown wire, black wire.



- For reference, + and – icons are drawn next to the pins that the red and black leads should be connected to, respectively
- Put the face plate in place, and ensure no wires are being sandwiched between the face plate and border plate
- Attach the face plate using the bolts, loosely tightening in the corners, then loosely tightening the ones in the center of the sides, before going around and tightening all the bolts, but not overtightening them.

**Hardware**






π

If you experience any issues or have question, reach me at jdolgin@uwaterloo.ca or joshuadolgin@gmail.com.