NeoMatrix 8x8 Word Clock

Created by Andy Doro

https://learn.adafruit.com/neomatrix-8x8-word-clock

Last updated on 2023-08-29 02:57:46 PM EDT
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Parts List</td>
<td>4</td>
</tr>
<tr>
<td>• Parts</td>
<td></td>
</tr>
<tr>
<td>• Tools</td>
<td></td>
</tr>
<tr>
<td>Circuit Assembly</td>
<td>6</td>
</tr>
<tr>
<td>• Overview</td>
<td></td>
</tr>
<tr>
<td>Uploading Code</td>
<td>12</td>
</tr>
<tr>
<td>• Understanding the Code</td>
<td></td>
</tr>
<tr>
<td>• Startup Sequence</td>
<td></td>
</tr>
<tr>
<td>• Setting the Time</td>
<td></td>
</tr>
<tr>
<td>• Daylight Saving Time</td>
<td></td>
</tr>
<tr>
<td>• Brightness Adjustment</td>
<td></td>
</tr>
<tr>
<td>• Color Shifting Speed</td>
<td></td>
</tr>
<tr>
<td>• Monochrome Mode</td>
<td></td>
</tr>
<tr>
<td>Enclosure Assembly</td>
<td>16</td>
</tr>
</tbody>
</table>
Overview

Are you fascinated with the passage of time? Do you want a stylish, modern and functional timepiece to add to your clock collection? The word clock is a one-of-a-kind time telling device, using a grid of letters to spell out the time. While you could spend thousands of dollars on other versions of this idea, this project is an inexpensive and quick way to build one for yourself.

The word clock uses the Adafruit NeoPixel NeoMatrix 8x8 to create a colorful word clock! As such, it features an original 8x8 layout of letters in order to form all of the different time phrases. You can power it over USB so it makes for a great desk time-keeper. This clock also uses the DS1307 Real Time Clock breakout kit so it'll keep time even while unplugged! The DS1307 has an accuracy of +/- 2 seconds per day, and the clock tells the time with a precision of five minutes. The microcontroller board we're using is the Pro Trinket 5V but you can swap it with any Arduino compatible or microcontroller that can use I2C and NeoPixels.
Parts List

Parts

- Trinket Pro 5V
- DS1307 Real Time Clock breakout board
- NeoPixel NeoMatrix 8x8 (http://adafru.it/1487)
- Wordclock laser-cut acrylic enclosure
- 4-40 black nylon screws (x14)
- 4-40 black nylon nuts (x14)
- 2-56 black SS machine screws (http://adafru.it/1487)
- 2-56 Black SS Hex Nut (x4)
- Wires, silicone cover are easiest to use (http://adafru.it/1311)
  just about any ~22-26 AWG wires will do (http://adafru.it/1311)
- MicroUSB cable (http://adafru.it/2008) (for uploading the code and powering the clock)
- 5V 1A USB port power supply (http://adafru.it/501) (if you don't want to just power the clock from your computer)
Adafruit NeoPixel NeoMatrix 8x8 - 64 RGB LED Pixel Matrix

Put on your sunglasses before wiring up this LED matrix - 64 eye-blistering RGB LEDs adorn the NeoMatrix for a blast of configurable color. Arranged in an 8x8 matrix, each pixel is...

https://www.adafruit.com/product/1487

Adafruit DS1307 Real Time Clock Assembled Breakout Board

This is a great battery-backed real time clock (RTC) that allows your microcontroller project to keep track of time even if it is reprogrammed, or if the power is lost. Perfect for...

https://www.adafruit.com/product/3296
Adafruit Pro Trinket - 3V 12MHz

Deprecation Warning: The Pro Trinket bit-bang USB technique it uses doesn't work as well as it did in 2014, many modern computers won't work well. So while we still...

Tools

- a computer which can program the Trinket Pro 5V
- a soldering iron (http://adafru.it/1204)
- solder (http://adafru.it/2473)
- wire strippers (http://adafru.it/527)
- diagonal cutters
- small flathead screwdriver (http://adafru.it/424) (2.4mm)

Circuit Assembly

Overview

This circuit stays a bit more compact by having the DS1307 RTC breakout soldered directly onto the Pro Trinket 5V. This makes the reset button on the Pro Trinket 5V more difficult to access, but you can still enter bootloader mode by using tweezers to push the button or just by unplugging and plugging the Pro Trinket 5V into the USB port. When the Pro Trinket 5V first powers up, it will be in bootloader mode.

You don't have to do this, it simply makes the circuit assembly more compact by using A2 and A3 as power pins (The DS1307 is powered by turning A2 low and A3 high in the Arduino code.) You can also just freewire the RTC breakout.
Start by assembling the DS1307 Real Time Clock breakout board by following this learn guide, if you don't have the newer pre-assembled version. You only need to solder in the male headers for GND, 5V, SDA and SCL. You can leave off SQW since it isn't used and the header won't fit nicely on top of the Pro Trinket. If you do solder it in, you can clip the bottom lead off.
Once the DS1307 breakout is assembled with headers, you can solder it on top of the Trinket Pro 5V so that the DS1307 GND lines up with the Pro Trinket A2, 5V with A3, SDA with A4 and SCL with A5. Make sure the boards are lined up correctly! SDA and SCL need to be connected to A4 and A5, respectively.
Solder the DS1307 RTC board onto the Pro Trinket 5V and then clip the excess leads.

NeoPixel NeoMatrix 8x8

Connect the NeoMatrix GND to the Trinket Pro GND, 5V to 5V and DIN to Pin 8. Cut the wires 5-8 inches or 13-20 centimeters long. Solder the wires into the back of the NeoMatrix so that the wires won't be visible from the front.
Here are the connections into the Pro Trinket 5V:
And this is what the finished circuit should look like!
Uploading Code

Make sure to use [Arduino IDE 1.6.4 or higher](https://www.arduino.cc/en/software) and follow [this tutorial](https://learn.adafruit.com/installing-adafruit-libraries) to install the Adafruit boards.

Download [the code](https://github.com/Adafruit/Adafruit_NeoPixel/tree/master/examples) from github by clicking Download ZIP. Uncompress the file and copy the folder to your Arduino sketchbook folder.

You will also need to install some Adafruit Arduino libraries. Follow [this tutorial](https://learn.adafruit.com/installing-adafruit-libraries) if you are unfamiliar with how to do this. You need to install the following Adafruit Arduino libraries:

- RTClib
- DST_RTC
- Adafruit_GFX
Once that is done, open the sketch and select Tools → Board → Pro Trinket 5V/16MHz (USB). See if the code compiles. If the libraries aren't installed correctly you will see errors.

Put the Pro Trinket into bootloader mode either by unplugging and replugging the Pro Trinket into the computer with your MicroUSB cable or by hitting the reset button. The reset button can be difficult or impossible to access if you've soldered the RTC on top or if you've already installed the circuit into the enclosure! So I find plugging the board into USB to work best.

When the red LED on the Pro Trinket is pulsing, the board is in bootloader mode. Once you're in bootloader mode, upload the code! If everything was done correctly, it should start telling you the time!
Understanding the Code

Startup Sequence

When the clock starts up, all of the individual words will light up sequentially during the startup sequence. This sequence is defined by the `flashWords()` function in the `setup()` function.

Setting the Time

The first time the code is run, the time on the RTC module will be set to the time that the code was compiled on your computer.

```c
if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
    // DST? If we're in it, let's subtract an hour from the RTC time to keep our DST calculation correct. This gives us // Standard Time which our DST check will add an hour back to if we're in DST.
    DateTime standardTime = RTC.now();
    if (dst_rtc.checkDST(standardTime) == true) { // check whether we're in DST right now. If we are, subtract an hour.
        standardTime = standardTime.unixtime() - 3600;
    }
    RTC.adjust(standardTime);
}
```

If you need to reset the time, this can be done by commenting out the if statement here but leaving in the line:

```c
RTC.adjust(DateTime(__DATE__, __TIME__));
```

Since this clock only displays the time within five minutes, some find it helpful to add 2.5 minutes (or 150 seconds) to the actual time to give a closer account of the time.

```c
// get the time
theTime = dst_rtc.calculateTime(RTC.now()); // takes into account DST
// add 2.5 minutes to get better estimates
theTime = theTime.unixtime() + 150;
```
Daylight Saving Time

Do you live in a territory that observes daylight saving time (DST)? If you do, you usually have to reprogram your clocks twice a year! This clock includes some code so that the adjustments are made automatically. The code follows the current rules for DST in the USA and Canada. If you live somewhere that follows different DST rules you may be able to modify the code to suit your rules—just look in the DST_RTC library functions. Wikipedia has a great reference on daylight saving time rules.

If you live in a territory that doesn't observe daylight saving time, just alter the following line by changing the 1 to 0.

```
#define OBSERVE_DST 1
```

The daylight saving time code works by keeping the real time clock on "standard time" and checking to see if the current date falls within daylight saving time. If the date falls within daylight saving time, an hour is added to the displayed time to convert from standard time to daylight saving time.

Brightness Adjustment

The clock is programmed to change brightness based on the time of day, operating at a lower brightness at nighttime. You can change these settings in the code.

DAYBRIGHTNESS is the brightness level during the day, NIGHTBRIGHTNESS is the brightness level during the night. Any number between 0-255 is acceptable, but at 0 you won't see anything! 255 is maximum brightness.

MORNINGCUTOFF and NIGHTCUTOFF set which hours the clock will operate at DAYBRIGHTNESS and NIGHTBRIGHTNESS. These hours are for a 24 hour clock, so 22 would be 10pm. Between midnight and 1am, the hour is 0.

```
// brightness based on time of day- could try warmer colors at night?
#define DAYBRIGHTNESS 80
#define NIGHTBRIGHTNESS 40

// cutoff times for day / night brightness. feel free to modify.
#define MORNINGCUTOFF 7 // when does day brightness begin? 7am
#define NIGHTCUTOFF 22 // when does night brightness begin? 10pm
```
If you don't want the clock to change brightness throughout the day, you can also just comment out the adjustBrightness() function in the main loop.

Color Shifting Speed

The speed at which the colors shift is controlled with a delay in milliseconds defined as SHIFTDELAY. To speed up the color shifting, decrease the number. To slow down the color shifting, increase the number.

```c
#define SHIFTDELAY 100   // controls color shifting speed
```

Monochrome Mode

Maybe the constant color shifting makes you queasy. You can simply run the clock to light the display in a single color.

First, we define that color- in this case white:

```c
// if you want to just run the clock monochrome
#define WHITE 200, 255, 255
```

255, 255, 255 would be pure white, but with RGB LEDs this can look a little purple so you can try adjusting the numbers.

Now in the colorFunctions tab, edit the two lines which set the Pixel color. Comment out the longer line and uncomment out the shorter line.

```c
//matrix.setPixelColor(i, Wheel(((i * 256 / matrix.numPixels()) + j) &amp; 255));
matrix.setPixelColor(i, WHITE);
```

Enclosure Assembly

Now that your circuit is complete, it's time to assemble the laser cut enclosure. You'll need to find a laser-cutting shop, hacker space or other friend with a laser cutter to cut out the pieces. You can find the files to cut in this github repository (), use 1/8" clear and black acrylic - or get creative and do something else!
Start by attaching the neopixel matrix to the acrylic plate that will hold it in place within the enclosure.

There are four screws that hold it in place.
Now take the back panel and attach the stainless steel machine screws which will hold the Pro Trinket in place.
Attach the Pro Trinket to the back plate, making sure the screws are tightened down firmly.
Connect the neopixel matrix to the back plate with the side panel, being careful to use the one with the hole for the micro USB.

Now you can add the other side panel and the top and bottom pieces, attaching each with the black nylon screws as you go.
Once all the clear acrylic pieces are put together, you are ready to add the pixel guard and diffuser.
Put the pixel guard in place on top of the neopixel grid. This will help contain the light from each pixel, making each letter on your clock crisper and easier to read.
If you would like to add a diffuser, now is the time. Diffusers are used to spread out the light from the neopixels and make the text on the faceplate easier to read.

You can make a diffuser from a plain sheet of paper, or any other material that will even out the bright light from the neopixels. Just trace the outline of the neopixel matrix and cut it out.

Place the diffuser on top of the neopixel matrix.
Now you are ready to attach the faceplate.
Before putting the faceplate in place, pull the protective paper cover off the faceplate. Any letter pieces should get pulled out along with the paper.
Use tweezers to poke out any bits of letters that don't fall out when the paper is pulled off.
Now you can screw down the faceplate onto the enclosure, sealing the diffuser inside.

Use the final four screws to affix the faceplate to the enclosure.
Your wordclock is assembled! Revel in your accomplishment.