



Simple Vertical Wordclock

Created by Dano Wall



<https://learn.adafruit.com/vertical-wordclock>

Last updated on 2024-06-03 02:49:55 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Vertical Clock• Materials	
Circuit Diagram	6
<ul style="list-style-type: none">• Solder Time• Flexible Power Source• Daylight Savings Time• Toggle Switch	
CircuitPython	11
<ul style="list-style-type: none">• Getting familiar with CircuitPython• Preparing your Board• Setting the Time• Vertical Wordclock Code• Making Changes	
Laser Cutting	16
Build the Clock	18
<ul style="list-style-type: none">• NeoPixel Channel• Clock Foundation• Bring it Together• Add Acrylic	

Overview

Vertical Clock

This simple vertical version on the ever-popular wordclock spells out the time by illuminating rows of words. Reading from top to bottom, any five minute increment of time can be spelled out by illuminating some combination of these rows. (Credit to Twitter user [@3d0g](https://adafru.it/Fi9) (<https://adafru.it/Fi9>) for this concept.)

This project uses a laser engraved piece of acrylic paired with a cardboard stand and is lit by a strip of [NeoPixels](https://adafru.it/dAR) (<https://adafru.it/dAR>) powered by a [Feather M4 Express](http://adafru.it/3857) (<http://adafru.it/3857>) board with a [Precision Real Time Clock FeatherWing](http://adafru.it/3028) (<http://adafru.it/3028>) keeping time even if the clock is disconnected from power.



[Adafruit NeoPixel LED Side Light Strip - Black 60 LED](https://www.adafruit.com/product/3636)

Fancy new side light LED strips are a great alternative for folks who have loved and used Adafruit LED strips for a few years but want gorgeous, glowy light emitting at...

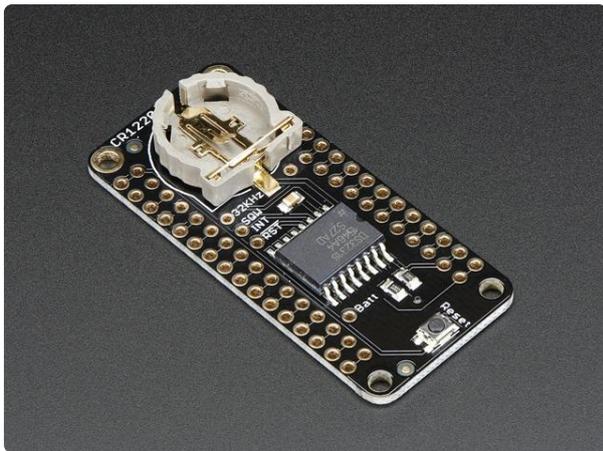
<https://www.adafruit.com/product/3636>



Female DC Power adapter - 2.1mm jack to screw terminal block

If you need to connect a DC power wall wart to a board that doesn't have a DC jack - this adapter will come in very handy! There is a 2.1mm DC jack on one end, and a screw terminal...

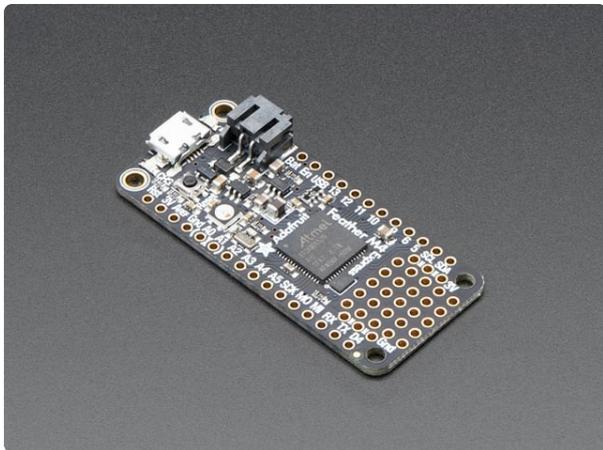
<https://www.adafruit.com/product/368>



DS3231 Precision RTC FeatherWing - RTC Add-on For Feather Boards

A Feather board without ambition is a Feather board without FeatherWings! This is the DS3231 Precision RTC FeatherWing: it adds an extremely accurate I2C-integrated...

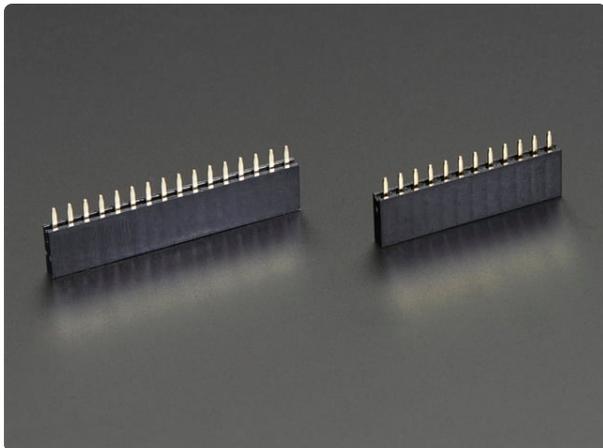
<https://www.adafruit.com/product/3028>



Adafruit Feather M4 Express - Featuring ATSAM51

It's what you've been waiting for, the Feather M4 Express featuring ATSAM51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,...

<https://www.adafruit.com/product/3857>



Header Kit for Feather - 12-pin and 16-pin Female Header Set

These two Female Headers alone are, well, lonely. But pair them with any of our <https://www.adafruit.com/product/2886>



5V 2A (2000mA) switching power supply - UL Listed

This is an FCC/CE certified and UL listed power supply. Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 2000mA. 110 or 240 input, so it works...

<https://www.adafruit.com/product/276>



CR1220 12mm Diameter - 3V Lithium Coin Cell Battery

These are the highest quality & capacity batteries, the same as shipped with the iCufflinks, iNecklace, Datalogging and GPS Shields, GPS HAT, etc. One battery per order...

<https://www.adafruit.com/product/380>



USB cable - USB A to Micro-B

This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...

<https://www.adafruit.com/product/592>



Breadboard-friendly SPDT Slide Switch

These nice switches are perfect for use with breadboard and perfboard projects. They have 0.1" spacing and snap in nicely into a solderless breadboard. They're easy to switch...

<https://www.adafruit.com/product/805>



2-pin JST SM Plug + Receptacle Cable Set

These 2-wire cables are 16cm long and come as a set. One side has a 2-pin JST SM type connector plug on the end. The other side has a matching 2-pin JST SM type...

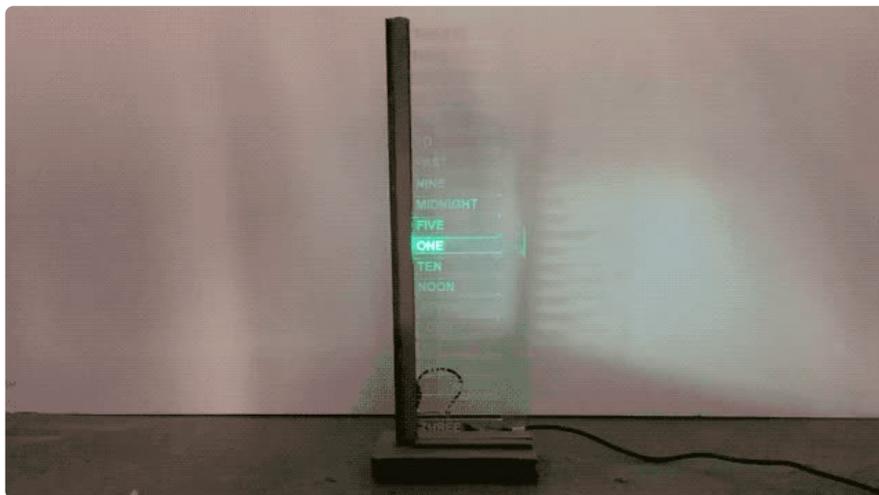
<https://www.adafruit.com/product/2880>

Materials

In addition to the electronics, for this project you will need:

- Scrap corrugated cardboard
- Scissors and/or box cutter
- [Hot glue gun \(https://adafru.it/Djk\)](https://adafru.it/Djk)
- [Hot glue sticks \(https://adafru.it/Dlg\)](https://adafru.it/Dlg)
- Soldering iron

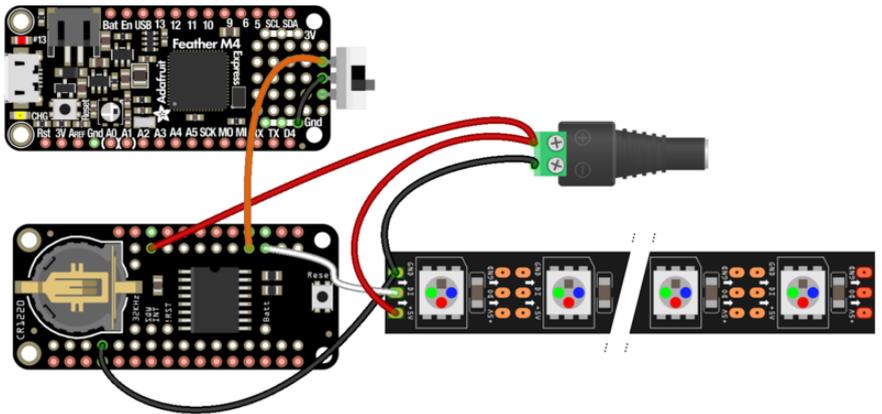
Read on to learn how to build your own!



Circuit Diagram

The circuit for this project requires a few soldered connections. Follow the diagram below to see how the circuit is constructed.

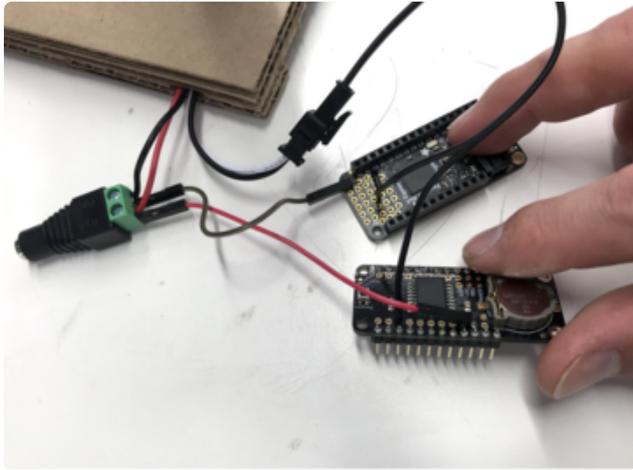
This circuit includes an optional switch that will allow the user to toggle the clock daylight savings time (+ one hour).



fritzing

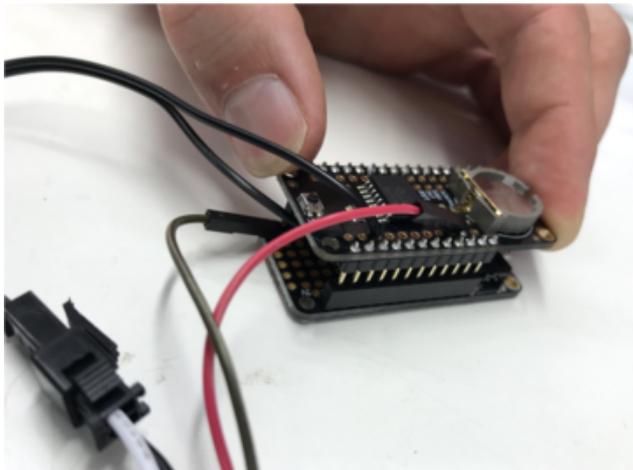
Start building this circuit by connecting the power and GND lines to the terminal blocks on the DC power adapter.





Solder Time

The Feather needs [Female headers \(http://adafru.it/2886\)](http://adafru.it/2886) soldered on. Solder male headers (included with the 'Wing) onto the RTC FeatherWing.



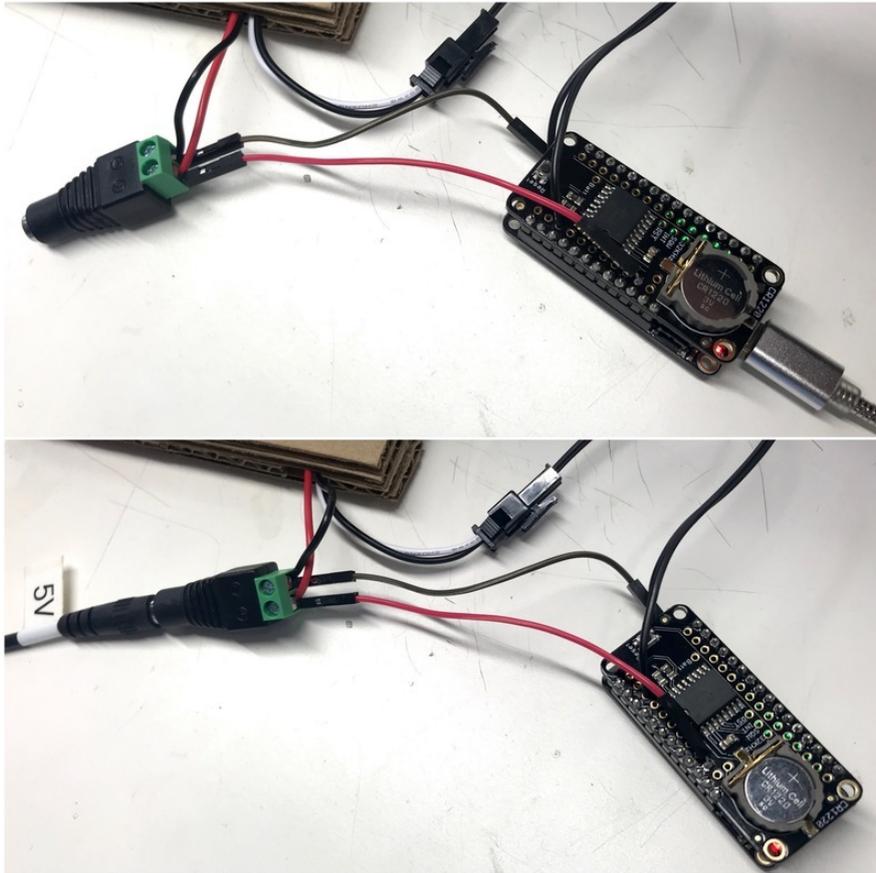
Solder the jumper cables from the terminal block to the Feather M4 and Featherwing.

Plug RTC FeatherWing into Feather M4 Express.

Flexible Power Source

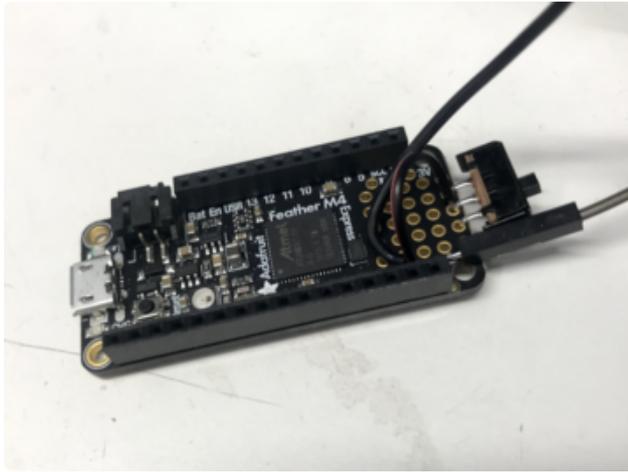
This circuit arrangement will allow you to conveniently power your clock via a micro USB cable or a 5V DC power supply interchangeably.

The small coin cell battery in the RTC FeatherWing will keep time for you even if the clock is unplugged, so you never have to worry about setting the time.



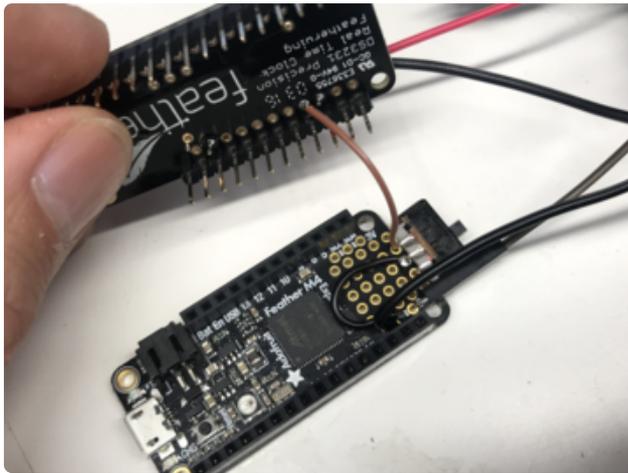
Daylight Savings Time

To account for the spring forward and fall back cycle we go through every 6 months, a small switch can be added to the circuit to allow the user to toggle the time forward an hour and back when needed.



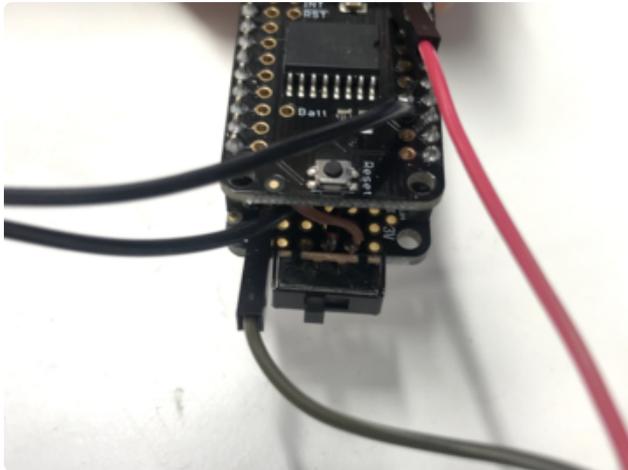
Toggle Switch

Bend the legs of the slide switch 90 degrees.

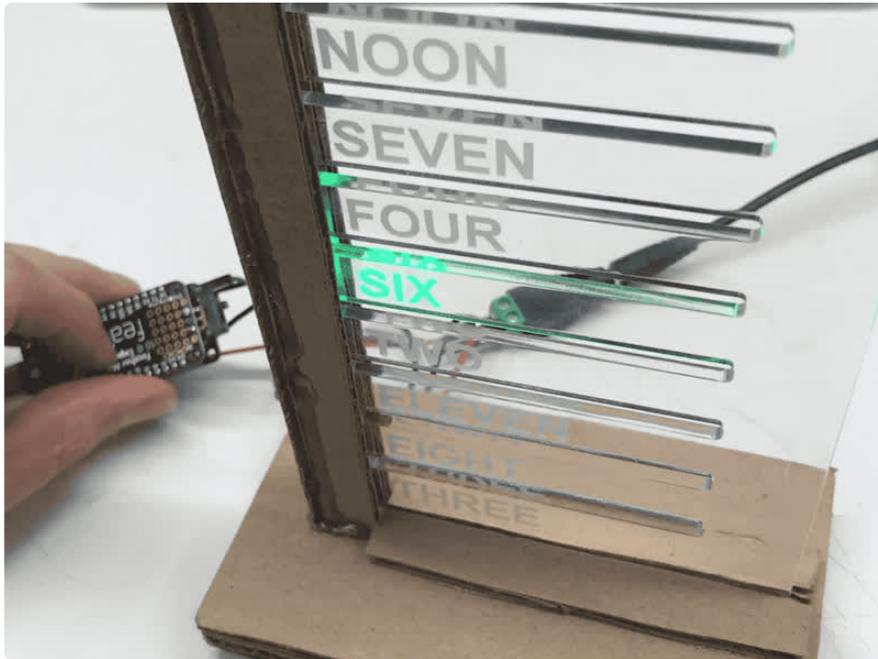


Solder one wire from the middle leg of the switch to the GND line.

Connect the left leg of the switch to pin #6 on the RTC FeatherWing.



Now, flipping this switch to the left will activate Daylight Savings mode, adjusting the clock one hour forward.



CircuitPython

Getting familiar with CircuitPython

[CircuitPython \(https://adafru.it/CgS\)](https://adafru.it/CgS) is a programming language based on Python, one of the fastest growing programming languages in the world. It is specifically designed to simplify experimenting and learning to code on low-cost microcontroller boards.

CircuitPython is easiest to use within the Mu Editor. If you haven't previously used Mu, [this guide will get you started \(https://adafru.it/ANO\)](https://adafru.it/ANO).



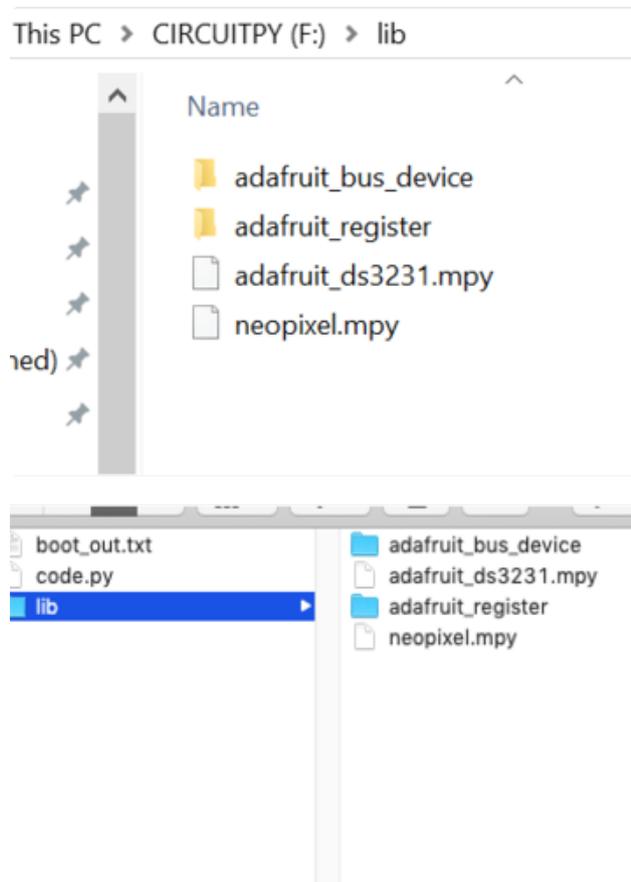
Preparing your Board

1) Download the [latest 4.0 library pack \(https://adafru.it/zB-\)](https://adafru.it/zB-),

2) Unzip it the library pack and drag the `adafruit_bus_device`, `adafruit_ds3231`, `adafruit_register`, and `neopixel` libraries over into the `/lib` folder on **CIRCUITPY**.

If there is no `lib` directory on your CIRCUITPY drive, create one to put the file into.

More info on installing libraries can be found [here \(https://adafru.it/C9M\)](https://adafru.it/C9M).



Make sure you have the following libraries installed:

`adafruit_bus_device`
`adafruit_ds3231`
`adafruit_register`
`neopixel`

Setting the Time

First things first, let's tell the RTC FeatherWing what time it is!

Copy and paste the code below into Mu. In line 22, fill in the correct year, month, day, hour, and minute. Save this as `code.py` to your **CIRCUITPY** device.

(More details on using the DS3231 Precision RTC can be found in [this guide \(https://adafru.it/Fia\)](https://adafru.it/Fia))

```
# SPDX-FileCopyrightText: 2019 Limor Fried for Adafruit Industries
# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT
```

```

# Write the time for the Adafruit DS3231 real-time clock.
# Limor Fried/Anne Barela for Adafruit Industries

import time
import board
import busio as io
import digitalio
import adafruit_ds3231

i2c = io.I2C(board.SCL, board.SDA)

# Create the RTC instance:
rtc = adafruit_ds3231.DS3231(i2c)

LED13 = digitalio.DigitalInOut(board.D13)
LED13.direction = digitalio.Direction.OUTPUT

# pylint: disable-msg=using-constant-test
if True:
    #                               year, mon, date, hour, min, sec, wday, yday, isdst
    t = time.struct_time((2019, 7, 10, 17, 00, 0, 0, -1, -1))
    # you must set year, mon, date, hour, min, sec and weekday
    # yearday is not supported
    # isdst can be set but we don't do anything with it at this time
    print("Setting time to:", t)      # uncomment for debugging
    rtc.datetime = t
    print("Done!")
    LED13.value = True
# pylint: enable-msg=using-constant-test

```

Once the time has been set we're ready to move on to uploading the Wordclock code.

Vertical Wordclock Code

Make sure you've connected the Feather M4 Express to your computer (mac/PC/Linux) via a known good USB A to micro-B cable. Your board should show up as a flash disk drive named **CIRCUITPY** (If you see a disk name **FEATHERBOOT**, try pressing the reset button again. If the only drive name you get is named **FEATHERBOOT**, CircuitPython may not be loaded on the board. You can load CircuitPython as [per this guide \(https://adafru.it/AFI\)](https://adafru.it/AFI)).

Once your board is connected, copy the code below from the window below to a file on the Feather **CIRCUITPY** drive as **code.py**. You can select **Download** in the upper left to save the code onto your computer.

```

# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Vertical Word Clock using the Adafruit Feather M4 and
# the Adafruit DS3231 real-time clock FeatherWing

import time
import board
import busio as io

```

```

import digitalio
import adafruit_ds3231
import neopixel

i2c = io.I2C(board.SCL, board.SDA)

# Create the RTC instance:
rtc = adafruit_ds3231.DS3231(i2c)

# Set up Feather M4 onboard LED for output
LED13 = digitalio.DigitalInOut(board.D13)
LED13.direction = digitalio.Direction.OUTPUT

# Set digital 6 as an input for slide switch
Slide_Switch = digitalio.DigitalInOut(board.D6)
Slide_Switch.switch_to_input(pull=digitalio.Pull.UP)

pixel_pin = board.D5
num_pixels = 21
pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=1.0,
                           auto_write=False)
pixels.fill((0, 0, 0)) # Blanking Display
COLOR = (0, 200, 0)   # Green for time later in code

# Bitmap values for each value. These can be OR'ed together
THREE = 1
EIGHT = 1 << 1
ELEVEN = 1 << 2
TWO = 1 << 3
SIX = 1 << 4
FOUR = 1 << 5
SEVEN = 1 << 6
NOON = 1 << 7
TEN = 1 << 8
ONE = 1 << 9
FIVE = 1 << 10
MIDNIGHT = 1 << 11
NINE = 1 << 12
PAST = 1 << 13
TO = 1 << 14
FIVEMIN = 1 << 15
QUARTER = 1 << 16
TENMIN = 1 << 17
HALF = 1 << 18
TWENTY = 1 << 19

# Pass in hour and minute, return LED bitmask
# pylint: disable=too-many-branches
# pylint: disable=too-many-statements
def writetime(the_hr, the_min):
    value = 0 # Start with zero, which is no words
    if (the_hr == 24) and (the_min == 0): # Special cases: Midnight and Noon
        return MIDNIGHT
    if (the_hr == 12) and (the_min == 0):
        return NOON
    # set minute
    if (the_min > 3) and (the_min < 8):
        value = value | FIVEMIN
    if (the_min > 7) and (the_min < 13):
        value = value | TENMIN
    if (the_min > 12) and (the_min < 18):
        value = value | QUARTER
    if (the_min > 17) and (the_min < 23):
        value = value | TWENTY
    if (the_min > 22) and (the_min < 28):
        value = value | TWENTY | FIVEMIN
    if (the_min > 27) and (the_min < 33):
        value = value | HALF
    if (the_min > 32) and (the_min < 38):

```

```

    value = value | TWENTY | FIVEMIN
if (the_min > 37) and (the_min < 43):
    value = value | TWENTY
if (the_min > 42) and (the_min < 48):
    value = value | QUARTER
if (the_min > 47) and (the_min <= 53):
    value = value | TENMIN
if the_min >= 54:
    value = value | FIVEMIN
# before or after
if (the_min > 3) and (the_min <= 32):
    value = value | PAST
if the_min >= 33:
    the_hr = the_hr + 1 # for the T0 case
    value = value | T0
# set hour
if the_hr > 12:
    the_hr = the_hr - 12 # Convert 24 hour format to 12 hour
if the_hr == 1:
    value = value | ONE
if the_hr == 2:
    value = value | TWO
if the_hr == 3:
    value = value | THREE
if the_hr == 4:
    value = value | FOUR
if the_hr == 5:
    value = value | FIVE
if the_hr == 6:
    value = value | SIX
if the_hr == 7:
    value = value | SEVEN
if the_hr == 8:
    value = value | EIGHT
if the_hr == 9:
    value = value | NINE
if the_hr == 10:
    value = value | TEN
if the_hr == 11:
    value = value | ELEVEN
if the_hr == 0:
    value = value | MIDNIGHT
if the_hr == 12:
    value = value | NOON
return value
# end def
# pylint: enable=too-many-branches
# pylint: enable=too-many-statements

# Main loop
LEDstate = 0
Write_Now = True

while True:
    t = rtc.datetime
    # print("The date is {} {}/{}{}".format(days[int(t.tm_wday)],
    #         t.tm_mday, t.tm_mon, t.tm_year))
    # print("The time is {}:{:02}:{:02}".format(t.tm_hour, t.tm_min, t.tm_sec))
    hour = t.tm_hour
    if not Slide_Switch.value: # Slide switch activate = Daylight savings
        # print("Switch detected for daylight savings")
        if hour == 24:
            hour = 1
        else:
            hour += 1
        Write_Now = True # Trigger a write
    minute = t.tm_min
    second = t.tm_sec
    if second == 59 or Write_Now:

```

```

# print("The time is {}:{:02}".format(t.tm_hour, t.tm_min))
pixels.fill((0, 0, 0)) # blank all pixels for change
the_time = writetime(hour, minute)
for i in range(0, 21): # Check all bits
    if the_time & 1 << i: # If the bit is true
        pixels[i+1] = COLOR # set pixel on (shift up 2 for buried one)
pixels.show()
if LEDstate == 0: # Flash the D13 LED every other second for activity
    LED13.value = True
    LEDstate = 1
else:
    LED13.value = False
    LEDstate = 0
Write_Now = False
time.sleep(1) # wait a second

```

Open the code up in the Mu editor. Press the **Save** button and your code should automatically be saved to the **CIRCUITPY** disk drive (which appears when the Feather M4 Express is plugged into your computer) as **code.py**.



Make sure the file saved to CIRCUITPY is named "code.py", this will allow it to run automatically when your Feather is powered on.

Making Changes

If you'd like to adjust the color of your clock, this can be changed in **line 29** of the code. Simply change the numbers in `COLOR = (0, 200, 0)` (green) - make the clock glow blue `(0, 0, 200)` or red `(200, 0, 0)`, or any color in between! A great color picker is at https://www.w3schools.com/colors/colors_picker.asp (<https://adafru.it/BPY>).

Laser Cutting

The face of this clock is made from a piece of acrylic that's engraved and cut so as to contain the light from each NeoPixel so that each word glows distinctly.

Vertical_Wordclock.ai

<https://adafru.it/Fib>

This file has the words spaced to align with the spacing between each NeoPixel, 0.656" between centers. The words are engraved, the outlines cut.

TWENTY

HALF

TEN

QUARTER

FIVE

TO

PAST

NINE

MIDNIGHT

FIVE

ONE

TEN

NOON

SEVEN

FOUR

SIX

TWO

ELEVEN

EIGHT

THREE



Build the Clock

The frame for your clock can be as simple or as fancy as you wish. Here, a minimalist and inexpensive option is shown, creating a thin channel in which to hold the NeoPixel strip and a small cardboard base to keep it standing.



NeoPixel Channel

Cut three long, thin strips of cardboard.

Glue them together to create a channel that will hold the NeoPixel strip.

Lay the NeoPixel strip into the channel to test its fit.



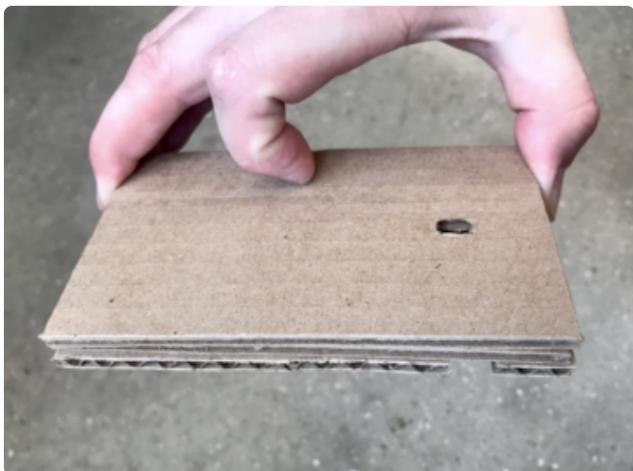


Clock Foundation

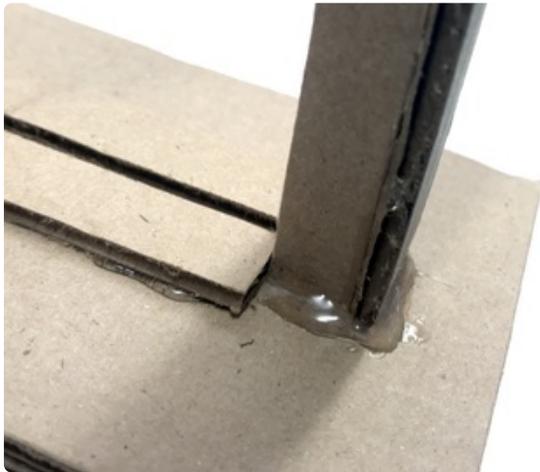
Cut three rectangular sections of cardboard.

Make a small square hole through two of them and cut a longer channel out of the third.

Glue the three pieces together, putting the channel on the bottom

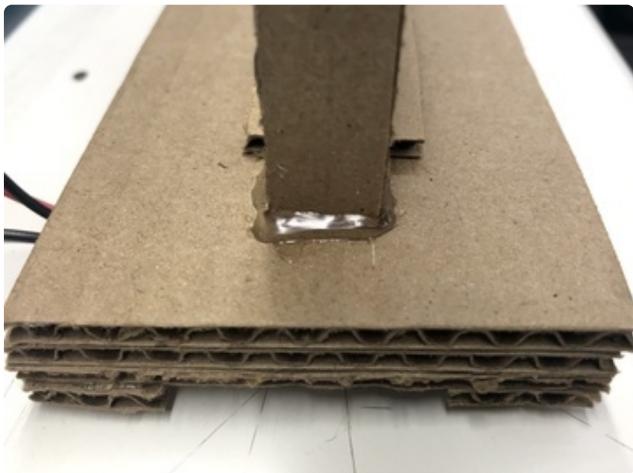


Be careful using hot glue as the dispenser and fresh glue can cause burns.



Bring it Together

Glue two strips of cardboard on the base to hold the bottom of the acrylic in place.



Use hot glue to affix the NeoPixel channel to the base, centering it over the hole cut for the the strip.

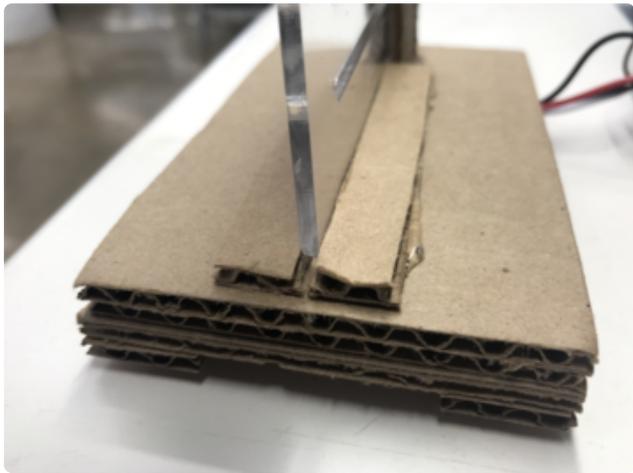
Thread the NeoPixel strip through the hole in the base, pulling it all the way through so the last pixel is at the top of the channel. Keep in mind the bottom pixel will be hidden in the base of the clock; it is ignored in the code, this is ok.



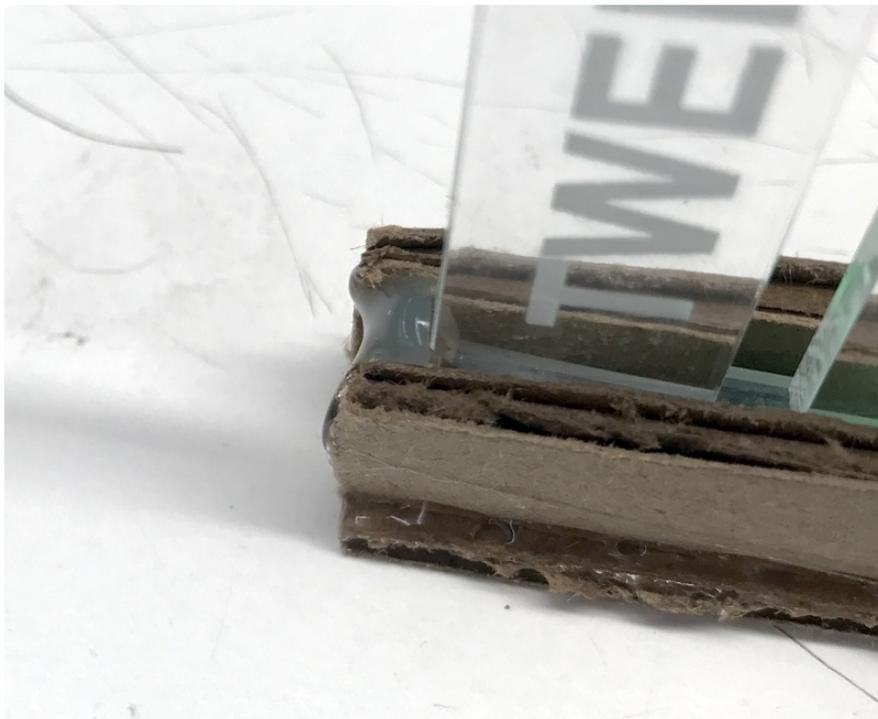
Add two more strips of cardboard to the bottom of the base, pinning the end of the NeoPixel strip so that it comes out the back.



Add Acrylic
Slot acrylic piece into place.



As a final touch, once the acrylic piece is in place, add a dab of hot glue to the top to keep the acrylic held tightly against the the NeoPixel strip.



Power on your clock and you should see the time pop up!

