32x32 Square Pixel Art Animation Display

Created by Ruiz Brothers

https://learn.adafruit.com/32x32-square-pixel-display

Last updated on 2023-08-29 04:37:41 PM EDT
# Table of Contents

## Overview
- Retro Pixel Display
- RP2040 & RGB Matrices
- Black LED Acrylic
- Prerequisite Guides
- Parts from Adafruit

## CAD Files
- CAD Files
- CAD Assembly
- Acrylic PDF Template

## Install CircuitPython
- CircuitPython Quickstart
- Safe Mode
- Flash Resetting UF2

## Code
- Download Project Bundle
- Upload Code, Libraries and Assets
- Modify Code
- Adjusted Code

## Sprite Sheets
- Animated Sprite Sheets
- Sprite Sheet Bitmap Memes
- Creating Custom Sprite Sheets

## Feather Headers
- Solder Feather RP2040 Headers
- Solder RGB Matrix FeatherWing Headers
- Solder DC Jack to RGB Matrix FeatherWing
- Solder IDC Header to RGB Matrix FeatherWing
- Solder Power Terminal to RGB Matrix FeatherWing
- Solder Doubler FeatherWing Headers
- Assembled Feathers

## Assembly
- Hardware for Doubler FeatherWing
- Install Hardware to Doubler FeatherWing
- Secure FeatherWing Doubler to Frame
- Install Feet
- Install Acrylic
- Install Grid
- Install Grid to Frame
- Install Power Cable
- Install IDC Cable
- Final Assembly
Overview

Retro Pixel Display
This is a square pixel display that uses an 32x32 RGB matrix display. It’s powered by the Feather RP2040 and RGB Matrix FeatherWing. This runs CircuitPython and the Matrix Portal Library to play animated sprite sheets on the display.

RP2040 & RGB Matrices
The Feather RP2040 and RGB Matrix FeatherWing are mounted to a 3D printed frame on the back of the display. Paired with the FeatherWing Doubler, it’s easy to swap out the Feathers for future projects.

Black LED Acrylic
A sheet of black LED acrylic fits over the display and diffuses the LEDs. The 3D printed grid fits over the PCB with each LED enclosed in a square. This separates each LED and blocks the light from leaking into each square.

Prerequisite Guides
Take a moment to walk through the following guides:

- [Introducing Feather RP2040](#)
• RGB Matrix FeatherWing ()
• RGB Matrices with CircuitPython ()

Parts from Adafruit
List of parts required for this build.

32x32 RGB Matrix - 6mm Pitch ()
Feather RP2040 ()
FeatherWing Doubler ()
RGB Matrix (M4+M0+RP2040)
FeatherWing ()
Black LED Acrylic 12x12in ()
5V 10A Power Supply ()
M2.5 Hardware Kit ()

32x32 RGB LED Matrix Panel - 6mm pitch
Bring a little bit of Times Square into your
home with this sweet 32 x 32 square RGB
LED matrix panel. These panels are
normally used to make video walls, here
in New York we see them...
https://www.adafruit.com/product/1484

Adafruit RGB Matrix Featherwing Kit
Ahoy! It's time to create a dazzling light up
project with our new RGB Matrix
FeatherWing. Now you can quickly and
easily create...
https://www.adafruit.com/product/3036
Adafruit Feather RP2040
A new chip means a new Feather, and the Raspberry Pi RP2040 is no exception. When we saw this chip we thought "this chip is going to be awesome when we give it the Feather..."
https://www.adafruit.com/product/4884

Black LED Diffusion Acrylic Panel 12" x 12"
- 0.1" / 2.6mm thick
A nice whoppin' slab of some lovely black acrylic to add some extra diffusion to your LED Matrix project. This material is 2.6mm (0.1") thick and is made of special cast...
https://www.adafruit.com/product/4594

FeatherWing Doubler - Prototyping Add-on For All Feather Boards
This is the FeatherWing Doubler - a prototyping add-on and more for all Feather boards. This is similar to our
https://www.adafruit.com/product/2890

5V 10A switching power supply
This is a beefy switching supply, for when you need a lot of power! It can supply 5V DC up to 10 Amps, running from 110V or 220V power (the plug it comes with is for US/Canada/Japan...
https://www.adafruit.com/product/658
CAD Files

STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material. Original design source may be downloaded using the links below.

frame.stl
grid.stl
dcover.stl
feet.stl

Parts require minimum build volume: 210mm x 210mm

CAD Assembly

The black LED acrylic is fitted into the cover. The 32x32 RGB Matrix PCB is placed inside the frame with the grid fitted on top. The frame and grid are press fitted into the cover.

The Doubler FeatherWing is secured to the frame using M2.5 hardware standoffs and screws.

Feet are optionally secured to the bottom of the frame using M3 hexnuts and screws.

Grab just the STL files for 3D printing.

Download STL Files

Includes a STEP and Fusion 360 Archive.

Download CAD Source

Opens in the web browser to preview 3D models. More download options available in the preview page.
Acrylic PDF Template
Print the template on a sheet of paper. Ensure the scale is 100% when printing to get the correct dimension.

Install CircuitPython

CircuitPython () is a derivative of MicroPython () designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

Download the latest version of CircuitPython for this board via circuitpython.org
Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.
You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :

Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.
Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

Auto-reload is off.
Running in safe mode! Not running saved code.
CircuitPython is in safe mode because you pressed the reset button during boot. Press again to exit safe mode.
Press any key to enter the REPL. Use CTRL-D to reload.

You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

Flash Resetting UF2

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep
clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

Code

Download Project Bundle

Once you've installed the latest version of CircuitPython onto your board, you'll need to grab the code, libraries and any assets included with the project.

Click the download button below to get the code, libraries and assets all in one!

```python
# SPDX-FileCopyrightText: 2020 John Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import os
import board
import displayio
from digitalio import DigitalInOut, Pull
from adafruit_matrixportal.matrix import Matrix
from adafruit_debouncer import Debouncer

SPRITESHEET_FOLDER = "/bmps"
DEFAULT_FRAME_DURATION = 0.1  # 100ms
AUTO_ADVANCE_LOOPS = 3
FRAME_DURATION_OVERRIDES = {
    "three_rings1-sheet.bmp": 0.15,
    "hop1-sheet.bmp": 0.05,
    "firework1-sheet.bmp": 0.03,
}

# --- Display setup ---
matrix = Matrix(bit_depth=4)
sprite_group = displayio.Group()
matrix.display.show(sprite_group)

# --- Button setup ---
pin_down = DigitalInOut(board.BUTTON_DOWN)
```

© Adafruit Industries
pin_down.switch_to_input(pull=Pull.UP)
button_down = Debouncer(pin_down)
pin_up = DigitalInOut(board.BUTTON_UP)
pin_up.switch_to_input(pull=Pull.UP)
button_up = Debouncer(pin_up)

auto_advance = True

file_list = sorted([f
    for f in os.listdir(SPRITESHEET_FOLDER)
    if (f.endswith(".bmp") and not f.startswith("."))
])

if len(file_list) == 0:
    raise RuntimeError("No images found")

current_image = None
current_frame = 0
current_loop = 0
frame_count = 0
frame_duration = DEFAULT_FRAME_DURATION

def load_image():
    """
    Load an image as a sprite
    """
    # pylint: disable=global-statement
    global current_frame, current_loop, frame_count, frame_duration
    while sprite_group:
        sprite_group.pop()

    filename = SPRITESHEET_FOLDER + "/" + file_list[current_image]
    
    # CircuitPython 6 & 7 compatible
    bitmap = displayio.OnDiskBitmap(open(filename, "rb"))
    sprite = displayio.TileGrid(
        bitmap,
        pixel_shader=getattr(bitmap, 'pixel_shader', displayio.ColorConverter()),
        tile_width=bitmap.width,
        tile_height=matrix.display.height,
    )
    
    # CircuitPython 7+ compatible
    # bitmap = displayio.OnDiskBitmap(filename)
    # sprite = displayio.TileGrid(
    #     bitmap,
    #     pixel_shader=bitmap.pixel_shader,
    #     tile_width=bitmap.width,
    #     tile_height=matrix.display.height,
    # )

    sprite_group.append(sprite)
    current_frame = 0
    current_loop = 0
    frame_count = int(bitmap.height / matrix.display.height)
    frame_duration = DEFAULT_FRAME_DURATION
    if file_list[current_image] in FRAME_DURATION_OVERRIDES:
        frame_duration = FRAME_DURATION_OVERRIDES[file_list[current_image]]

def advance_image():
    """
    Advance to the next image in the list and loop back at the end
    """
# pylint: disable=global-statement
global current_image
if current_image is not None:
    current_image += 1
if current_image is None or current_image >= len(file_list):
    current_image = 0
load_image()

def advance_frame():
    """Advance to the next frame and loop back at the end""
    # pylint: disable=global-statement
    global current_frame, current_loop
    current_frame = current_frame + 1
    if current_frame >= frame_count:
        current_frame = 0
        current_loop = current_loop + 1
    sprite_group[0][0] = current_frame
advance_image()

while True:
    if auto_advance and current_loop >= AUTO_ADVANCE_LOOPS:
        advance_image()
    button_down.update()
    button_up.update()
    if button_up.fell:
        auto_advance = not auto_advance
    if button_down.fell:
        advance_image()
    advance_frame()
    time.sleep(frame_duration)

Upload Code, Libraries and Assets

Unzip the project bundle and upload the files to the CIRCUITPY drive.

Your CIRCUITPY drive should look like this after you've uploaded the code, libraries and assets.

Modify Code

To make the bitmaps display properly on the 32x32 RGB matrix display, look for the following code and update.
You'll also need to update the pins for the Up and Down buttons – These aren't used in this project but you'll need to update them in order to run the code.

Adjusted Code

```python
# --- Display setup ---
matrix = Matrix(bit_depth=6, width=32)
sprite_group = displayio.Group()
matrix.display.show(sprite_group)

# --- Button setup ---
pin_down = DigitalInOut(board.D4)
pin_down.switch_to_input(pull=Pull.UP)
button_down = Debouncer(pin_down)
pin_up = DigitalInOut(board.A0)
pin_up.switch_to_input(pull=Pull.UP)
button_up = Debouncer(pin_up)
```

Sprite Sheets

Animated Sprite Sheets

The sprite sheets are a series of images that are merged together in a single bitmap. Each section in the image is a single frame of the animation that gets played from top to bottom.

It's really easy to play new animations by just tossing new bitmaps onto the drive. The code automatically cycles through all of the images stored in the bitmaps folder.

Sprite Sheet Bitmap Memes

Download the bitmap images below for your testing. These are pre-made sprite sheets of the infamous party parrot and nyan cat.

```
led_matrices_nyancat-vertical.bmp
```
Creating Custom Sprite Sheets

Check out the following guide for lessons on how to create sprite sheet bitmap images.

Guide: Sprite Sheet Animation

Feather Headers

Solder Feather RP2040 Headers
Install a 12-pin and 16-pin strip of male header pins to the Feather RP2040.

Using a breadboard can help keep headers straight while soldering. Solder all of the pins to the RP2040.
Solder RGB Matrix FeatherWing Headers

Install a 12-pin and 16-pin strip of headers to the RGB Matrix FeatherWing.

Using a breadboard once again to help while soldering.
Solder DC Jack to RGB Matrix FeatherWing
Install the DC jack onto the RGB Matrix FeatherWing. Solder the DC jack from the bottom side of the PCB. Ensure to use a sufficient amount of solder.
Solder IDC Header to RGB Matrix FeatherWing

Use the notch marking on the silkscreen to determine the correct orientation of the IDC header. Install and insert the IDC header to the top of the PCB. Solder all of the pins.
Solder Power Terminal to RGB Matrix FeatherWing

Install and insert the screw-block terminal to the top side of the RGB Matrix FeatherWing. Solder the two pins from the bottom of the PCB.
Solder Doubler FeatherWing Headers
Use two sets of 12-pin and 16-pin female headers to the Doubler FeatherWing. Start by installing the headers on the top side of the PCB. While holding headers in place, flip the PCB over. Carefully solder all of the pins to the PCB.

Assembled Feathers
The Feather RP2040 and RGB Matrix FeatherWing are ready to install on to the Doubler FeatherWing.
Assembly

**Hardware for Doubler FeatherWing**
Use the following hardware for securing the Doubler FeatherWing.

- 4x M2.5 x 4mm screws
- 4x M2.5 x 6mm screws
- 4x M2.5 x 6mm FF standoffs

**Install Hardware to Doubler FeatherWing**
Insert an M2.5 x 4mm screws through one of mounting holes (near the corner) on the Doubler. While holding screw in place, fasten an M2.5 x 6mm FF standoff onto the screw.

Repeat process for a total of four standoffs.
Secure FeatherWing Doubler to Frame

Get the Doubler ready to secure onto the frame. Use 4x M2.5 x6mm screws to secure the Doubler to the frame. Place the Doubler over the frame with the standoffs lined up with the mounting holes. Insert and fasten 4x M2.5 x 6mm screws to secure the Doubler to the frame.

Get the Feathers ready to install on the Doubler FeatherWing.
Install Feet

Use the following hardware to secure the feet to the cover.

4x M3 x 6mm screws
4x M3 hex nuts

Insert the M3 hex nuts into the recesses in the inside cover. Firmly press the nuts to install them.

Insert the M3 screws through the holes in the feet. Place the feet over the cover and line up the mounting holes.

Insert and fasten the M3 screws to secure the feet to the cover. Repeat this process for the second foot.
Install Acrylic
Place the cut sheet of acrylic into the cover with the matte side facing down, reflective side facing up.

Install Grid
Place the grid over the PCB of the 32x32 RGB matrix. Make sure the grid is flush with the PCB and the LEDs are fitted through the squares.
Install Grid to Frame

While holding the grid and PCB together, place the frame over with the Doubler oriented correctly. Press the the two together so the frame is fitted over the grid and PCB.
Install Power Cable

Insert the red voltage wire into the positive screw-block on the FeatherWing. Insert the black ground wire into the negative screw-block. Use a screw driver to secure the connectors to the screw-block terminal on the FeatherWing.

Connect the power cable to the power connectors on the back of the RGB matrices PCB.
Install IDC Cable
Connect the IDC cable to the port on the RGB Matrix FeatherWing. Connect the other end of the cable to the HUB75 input on the back of the RGB Matrix.

One of the side clips on the IDC cable may need to be cut in order to fit through the header and DC jack on the RGB Matrix FeatherWing.

Final Assembly
And thats it! To power the display, plug in the 5V 10A power supply into the DC Jack on the RGB Matrix FeatherWing.