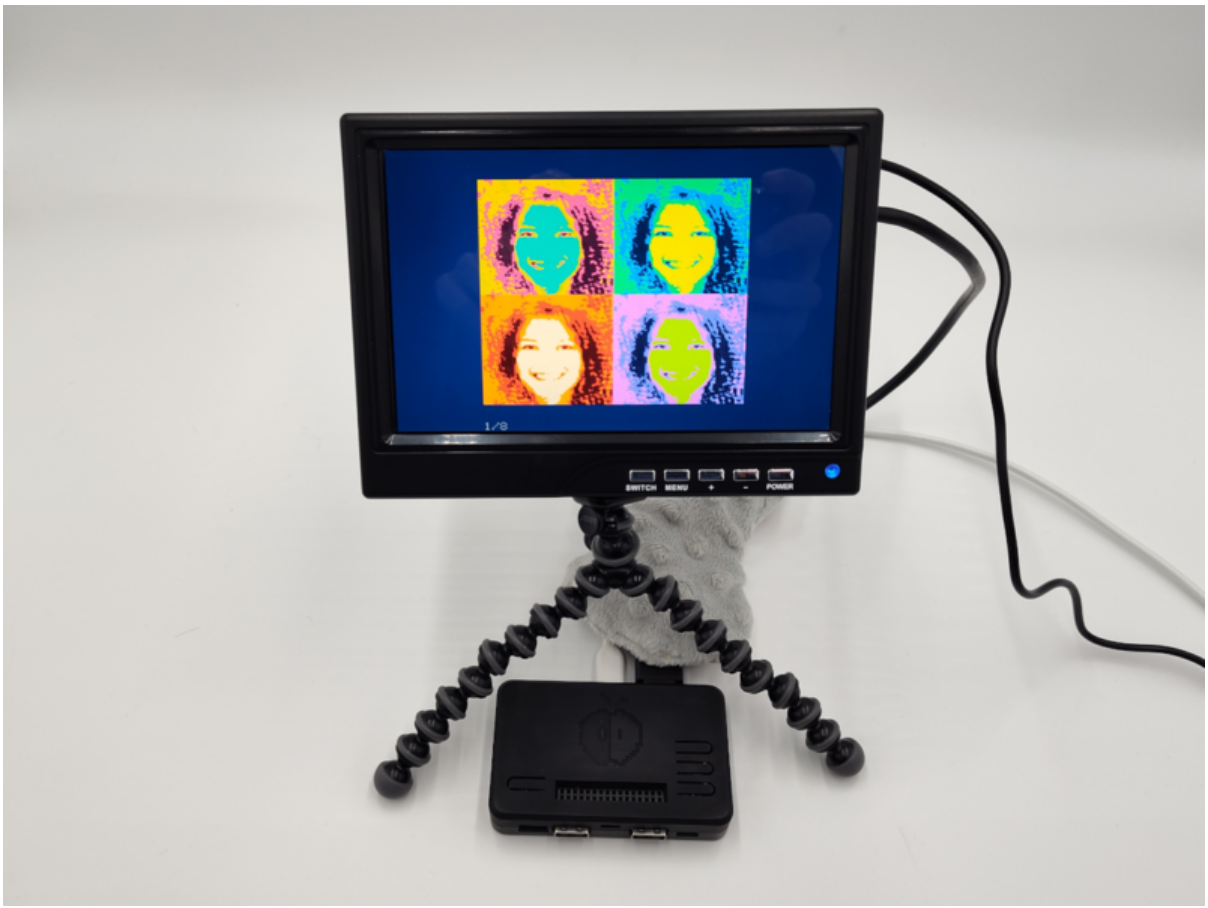
Pico-Faces On Fruit Jam

Created by Tim C



https://learn.adafruit.com/pico-faces-on-fruit-jam

Last updated on 2026-09-15 04:45:41 PM UTC

Table of Contents

Overview 3

- Parts

Code 4

- Building

Install Firmware 6

- Download
- Bootloader

Use 8

- Button 1
- Button 2
- Button 3

Overview

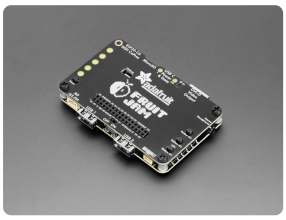
> In the future, every microcontroller will be able to run models with 1.5 million parameters.

The Pico-Faces () project is an image generation model that can run on a microcontroller. The model implements a latent flow diffusion transformer (DiT). Similar to techniques used by some larger GenAI image models, but with some tricks and efficiencies that pack it down to 5000 times fewer parameters than typical larger image gen models. The model runs on the RP2350 and generates 128x128 pixel images of faces.

The original project supports two output modes: 1) USB serial link with image data transferred in it and rendered on a host PC with help from the Python Imaging Library. 2) VGA output using the Pimoroni Pico VGA Demo Base (). I have added support for the Fruit Jam and HSTX/DVI video output.

In the ethos of "Art is what you can get away with", I also added a fun 2x2 color grid effect inspired by Andy Warhol's artwork.

Parts



Featured

Adafruit Fruit Jam - Mini RP2350 Computer

We were catching up on a recent hackaday hackchat with eben upton and learned some fun facts:...

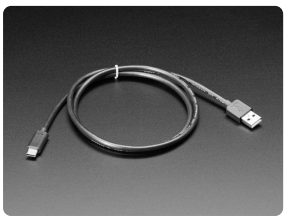
https://www.adafruit.com/product/6200

Featured

Snap-on Enclosure for Adafruit Fruit Jam

Here is a cool and minimal enclosure for your Fruit Jam to keep it safe during use and transport. This case has been...

https://www.adafruit.com/product/6425



Featured

USB Type A to Type C Cable - approx 1 meter / 3 ft long

As technology changes and adapts, so does Adafruit. This USB Type A to Type C cable will help you with the transition to USB C, even if you're still...

https://www.adafruit.com/product/4474

Featured

HDMI Cable - 1 meter

Connect two HDMI devices together with this basic HDMI cable. It has nice molded grips for easy installation, and is 1 meter long (about 3 feet). This is a HDMI 1.3...

https://www.adafruit.com/product/608

Featured

7" Display 1280x800 (720p) IPS + Speakers - HDMI/VGA/NTSC/PAL

Yes, this is an adorable small HDMI television with incredibly high resolution and built in 3W stereo speakers! We tried to get the smallest possible HDMI/VGA display with...

https://www.adafruit.com/product/1667

Code

Text emphasized with a blue info:

It's possible to run the Pico-Faces demo on the Fruit Jam without diving into the code and building from source. The UF2 files linked on the Install Firmware page are ready to go if you just want to try it out. This page contains links and build instructions in case you want to inspect the code, or modify the behavior.

There a two branches with code relevant to this project:

- The fruit_jam branch (), which contains the bare minimum to run Pico-Faces on the Fruit Jam and control it over USB serial. It does not display a face by default on boot up. To render a face, connect via USB serial to the device and send a capital "G" then press enter. This would be the best branch to build from if you want to take the bare Pico-Faces image generation and make something different from it on the Fruit Jam.

- The color_grid_mode branch (), which contains the full 2x2 color grid demo. This one does render the first face on boot up automatically. It also provides controls using the built-in buttons to change rendering modes and color style, so it can be used without a host computer or USB serial. Start from this branch if you want to customize the demo behavior or make custom color styles.

Building

Building the Pico-Faces project is a two step process. First configure the build, and then run it to make the firmware files. There are a few requirements to set up before the first build.

Clone a specified branch of the project using the following command. Replace `color_grid_mode` with `fruit_jam` if you want the bare Fruit Jam and DVI support without the 2x2 grid demo with button controls.

```
git clone -b color_grid_mode https://github.com/adafruit/pico-faces.git
```

Requirements

Clone the pico-sdk repo () and fetch the tinyusb () submodule inside of it using the following commands.

```
git clone https://github.com/raspberrypi/pico-sdk.git

cd pico-sdk
git submodule update --init lib/tinyusb
cd ../
```

Configure & Build

These commands will let the environment know where to find pico-sdk, configure a build, and then make the build.

Update the value of `PICO_SDK_PATH` to point to the pico-sdk repo that was cloned in the previous step if it isn't in home directory at `~/pico-sdk/`.

```
export PICO_SDK_PATH=~/pico-sdk/

# building on x86_64
cmake -S firmware -B build/fruitjam -G "Unix Makefiles" -DPICO_BOARD=adafruit_fruit_jam -DRF_D

# building on ARM like RPi
# cmake -S firmware -B build/fruitjam -DCMAKE_BUILD_TYPE=Release -DRF_DVI=1 -DRF_VGA=0 -DPICO_
```

```
cmake --build build/fruitjam -j4
```

When the build is complete, the firmware file will be output as `build/fruitjam/pico_faces.uf2`. That file is ready to be copied to the RP2350 bootloader drive of the Fruit Jam.

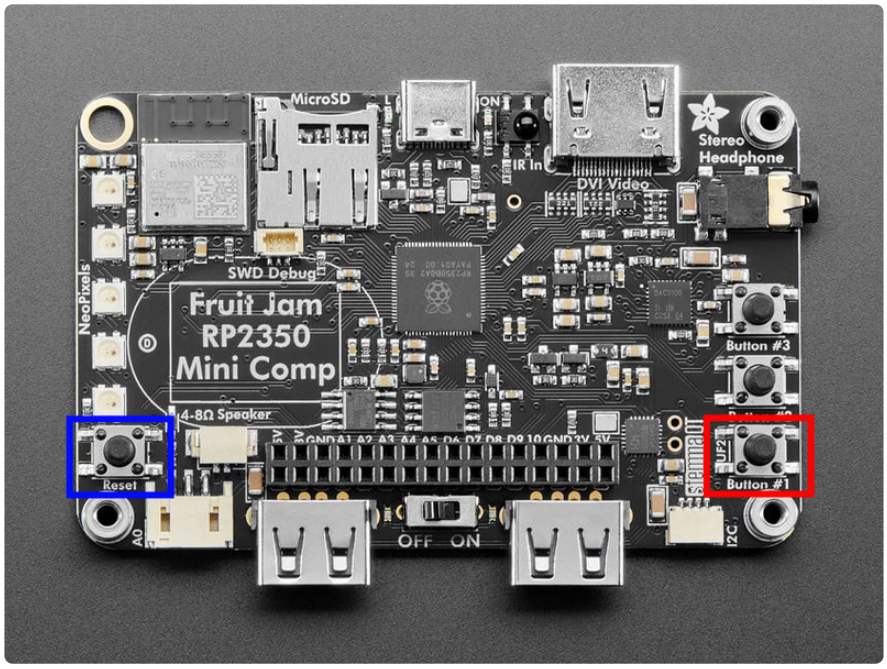
Install Firmware

Installing the Pico-Faces Fruit Jam demo is a breeze thanks to the RP2350's UF2 bootloader.

Download

Click the button below to download the Pico-Faces Fruit Jam demo firmware UF2 file.

Bootloader



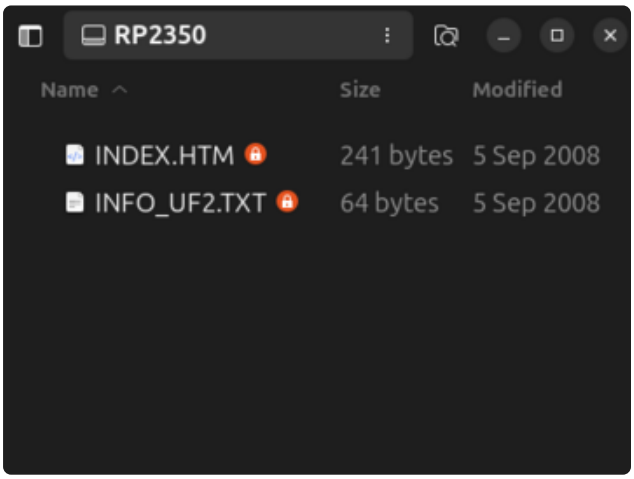
Connect your Fruit Jam board to your computer via a known good USB data+power cable. Go to Explorer or Finder (depending if you use Windows or Mac respectively).

To enter the Fruit Jam 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 red or blue above). **Continue to hold the BOOT/BOOTSEL button until the RP2350 drive appears in File Explorer or Finder!**

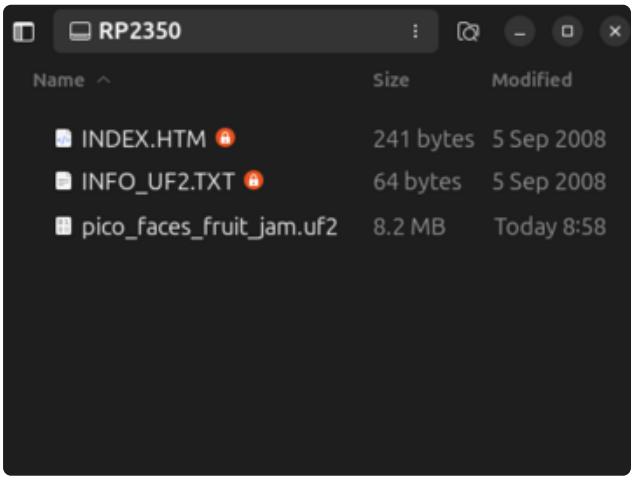
If the drive does not appear, release all the buttons, and then repeat the process above.

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.**

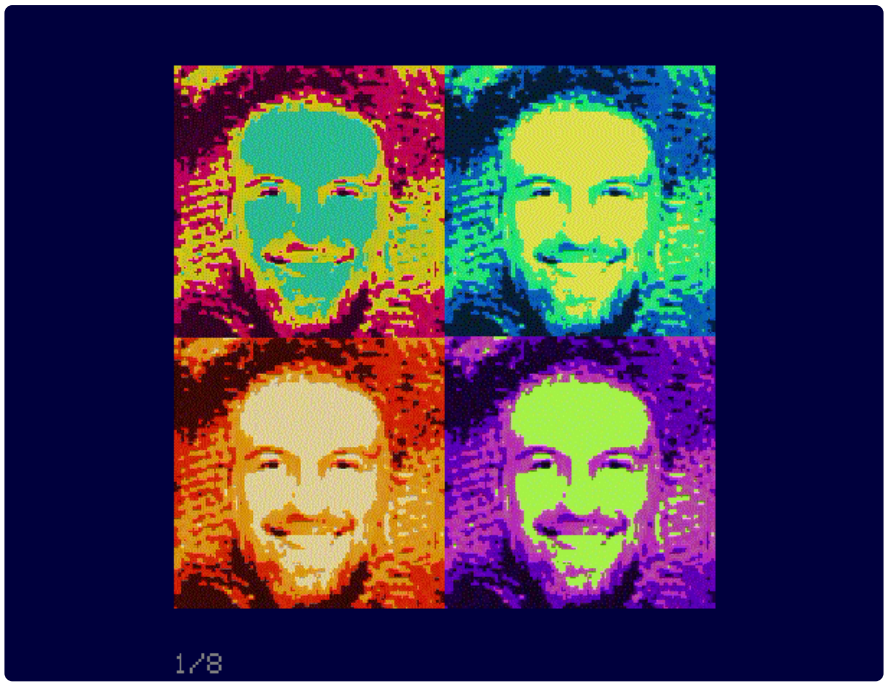


You will see a new disk drive appear called **RP2350**.



Drag or copy/paste the **pico_faces_fruit_jam.uf2** file to **RP2350.**

The firmware file is relatively large, it will take 30-60 seconds to copy to the bootloader drive. After copying is complete the Fruit Jam will reset automatically. When it starts up it will show a face grid on the connected display.

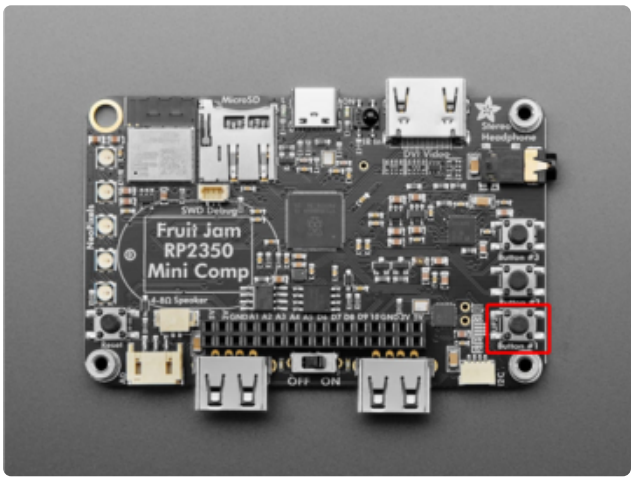


Use

Connect the Fruit Jam to a DVI compatible display and then connect a USB C cable to power it. The demo project will start automatically.

The demo can be controlled by the built-in buttons. If you want more fine grained control over specific configurations, the USB serial control () implemented by the original pico-faces project is also supported.

Button 1

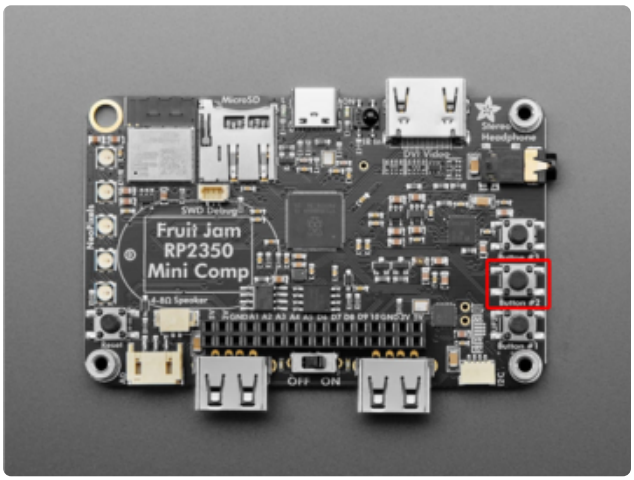


This button controls refreshing.

A single short press will trigger it to generate a new face and refresh the display to show it.

Pressing and holding button #1 will toggle between automatic and manual refresh modes. The demo starts in manual mode. When changed to automatic mode, it will generate a new face every 15 seconds without any button pressing.

Button 2

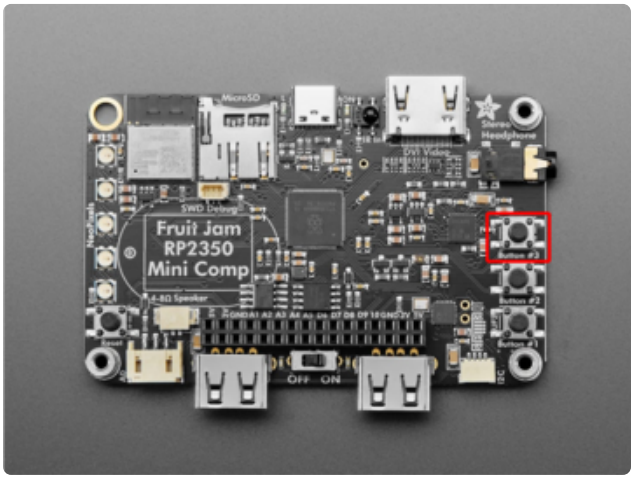


This button controls the color style and toggles the color grid mode on and off.

Single short presses on button #2 will cycle between the different available color styles in the color grid mode.

Pressing and holding the button will toggle between color grid mode and plain output mode. In plain output mode the face is rendered as it was output by the model.

Button 3



This button gives basic control over the model configuration. There are 8 configuration presets available to cycle between. The current preset is shown as a fraction in the lower left corner.

Single short presses on button #3 advances forward to the next config preset.

Pressing and holding button #3 will go backwards to the previous config preset.

The 8 config presets are shown in the table below.

K is how many Euler steps the model will use. More steps takes longer to render, and makes a more refined and detailed image.
Class is what type of face the model will be guided towards. The possibilities are the 4 permutations of female/male and neutral/smiling, as well as null class for unguided model output. The first two presets select one of the classes at random.
w is the guidance strength. Higher **w** will be more strongly guided towards the specified class. When the model is unguided, the guidance strength will have no effect.

Name	K	Class	w
1 `soft K2 w4 rnd`	2	Random	4
2 `strong K8 w8 rnd`	8	Random	8
3 `unconditional`	4	Null	0

4 `sketch K1`	1	Null	0
5 `female neutral`	4	Female, neutral	6
6 `female smiling`	4	Female, smiling	6
7 `male neutral`	4	Male, neutral	6
8 `male smiling`	4	Male, smiling	6