NES Emulator for RP2040 DVI Boards

Created by Phillip Burgess

https://learn.adafruit.com/nes-emulator-for-rp2040-dvi-boards

Last updated on 2023-08-29 04:57:28 PM EDT
# Table of Contents

**Overview** 3
- Hardware Selection

**Controllers** 4
- USB Controllers
- NES and SNES Gamepads
- Wii Classic Controller

**Using Feather RP2040 DVI** 6
- Input Options
- Software

**Using Pico + DV Demo Base** 9
- Input Options
- Software

**Using Pico + Breakouts** 12
- Input Options
- Breadboard Method
- Custom PCB Method
- Software
Overview

Nearly 40 years on, the original Nintendo Entertainment System (or Famicom in Japan) still holds a certain mystique, with genre-defining games like Super Mario Bros. and The Legend of Zelda. The console remains a popular target for emulation, which is what we'll do here on the bonkers-affordable RP2040 microcontroller paired with an HDMI TV or monitor for display, and with several controller options (including, with a little soldering work, original NES controllers for That Authentic Feel).

The software for this guide is Frank Hoedemakers’ pico-infonesPlus — a veritable rock soup of contributions, starting from Jay Kumogata’s InfoNES, ported to RP2040 by Shuichi Takano (building on Luke Wren’s PicoDVI library), Frank’s work on the SD card menu, and some controller code from Adafruit. That’s a lot of pieces…but we have ready-made .UF2 files to make installation easy on various hardware combos!

Hardware Selection

There are a few ways this can be built, please read through the guide to see what your options are. Parts links are provided as we go. Think about how you might use this…and whether and how you may want to re-use elements for other DIY projects later.

One path uses the Adafruit Feather RP2040 DVI board and SD card FeatherWing. Other paths use the Raspberry Pi Pico development board and either a Pimoroni Pico DV Demo Base or a Adafruit DVI and Micro SD breakout boards. In any
case, there may be some additional bits and bobs needed — soldering iron and
related paraphernalia, USB cables depending on the path taken, and so forth. These
are discussed later, so again, please read through.

There are also different controller options: Sony Dualshock 4 or DualSense USB
controllers, NES or SNES gamepads (with some soldering), or Wii Classic Controller,
each with some pros and cons which we’ll explain.

All the different builds require a microSD memory card for holding game ROM files.
NES games are comparatively tiny, so if you have an old card lying around that’s “too
small for anything useful,” this is its moment to shine.

An HDMI cable and TV or monitor are also assumed.

Controllers

Already, a fork in the road: choosing a controller. How do you want this to feel in your
hand? What might you have around already, or how much fuss to invest?

USB Controllers

pico-infonesPlus can work with Sony DUALSHOCK 4 or Dualsense controllers
(via USB cable). These are not inexpensive…but maybe you already have
one on hand.

A downside to USB-based controllers with
this project is that they do have a bit of
lag; not ideal for fast-twitch shooters, but
may suffice if RPGs and casino games are
more your style.

In addition to the controller itself, you’ll need a USB OTG cable to adapt the
controller’s USB-A plug to the receptacle on the microcontroller board. This controller
option might be easiest with the Pico DV Demo Base, since that board has its own
USB connector for power, leaving the Pico RP2040 board’s USB micro-B port free. For
other boards, you’d need a USB OTG splitter cable with both a USB-A port and a
second port (USB micro-B or USB-C depending on the board) for feeding power.
NES and SNES Gamepads

If you’re comfortable with a little bit of wiring and soldering, it’s possible to use original NES and Super Nintendo (and compatible) controllers, and they’re super responsive with the emulator, no perceptible lag.

To use these controllers unmodified, you’ll need to source the corresponding sockets, which can be found on eBay. Check where you’re buying from and understand parts may take a few weeks to arrive if overseas.

A different route is to forego the game port, cut the end off a controller and hard-wire it directly into the circuit. You lose the ability to ever use that controller with an actual console or other emulator in the future.

Note: although Super Nintendo controllers are supported, the emulator itself can only handle original NES games, of the non-Super variety.

Wii Classic Controller

This controller was an adjunct to the Nintendo Wii “Nunchuk,” but is possible to use it on its own with this breakout adapter and a bit of wiring or a STEMMA QT cable. This too is a highly responsive controller.

Support for the Wii Classic Controller is currently only enabled on the Feather RP2040 DVI, as that board has the STEMMA connector built-in. It could be enabled on other boards if building from source and an I2C-capable pair of pins are free (this may be challenging, as nearly every pin is already assigned to something).
Using Feather RP2040 DVI

The first hardware option, and smallest overall, uses the Feather RP2040 DVI and Ada logger FeatherWing. This is also the only option supporting the Wii Classic Controller by default.

The Adalogger FeatherWing provides the microSD card slot used for holding games. This board also supports a battery-backed realtime clock, not used by this project; battery neither included nor required here.

It’s common in some Feather projects that the 'Wing stacks on top...but here, in order to access the Feather’s STEMMA port and boot and reset buttons, a different approach should be taken, your choice:

- Stack the Feather board on top of the 'Wing using header row pins (Feather) and sockets (Wing)...low-profile headers/pins are available if you want a slimmer package. Soldering the two permanently (using just header row pins, no sockets) is the slimmest of all options, but blocks access to the battery socket and limits how this might get re-used in future projects.
- Use a FeatherWing Doubler or Tripler to mount the boards side-by-side (see image below). A little extra bulk and cost, but has better re-use potential later.

These are required parts for this variant:

1 x **Feather RP2040 DVI**
Adafruit Feather RP2040 with DVI Output Port

1 x **Adalogger FeatherWing**
Adalogger FeatherWing - RTC + SD Add-on For All Feather Boards

And these parts are optional, picking and choosing which suit your build plans:

1 x **FeatherWing Doubler**
FeatherWing Doubler - Prototyping Add-on For All Feather Boards

1 x **FeatherWing Tripler**
FeatherWing Tripler Mini Kit - Prototyping Add-on For Feathers
Input Options

To use a Wii Classic Controller:


To use one of the supported Sony USB controllers:

- A suitable USB-OTG splitter cable is needed; this is something we don’t stock but can be found online. Look for USB-C plug for the Feather, USB-A socket for the controller, and then whatever connector you would prefer for connecting power (USB micro-B and USB-C are common).

To use NES or SNES compatible gamepads:

- This is not plug-and-play and will require a little soldering and creativity. You might need to “blue wire” to a few header edge pins on the Feather board, or
add a FeatherWing Proto () to the stack (or FeatherWing Tripler () if mounting boards side-by-side).

- Wire up a NES or SNES controller port (these can be found on eBay)...or, if you don’t mind sacrificing a junky controller, cut the plug off and wire directly between the circuits.

- Connections are:

<table>
<thead>
<tr>
<th>NES/SNES Controller</th>
<th>Feather DVI Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>3.3V</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Clock</td>
<td>5</td>
</tr>
<tr>
<td>Data</td>
<td>6</td>
</tr>
<tr>
<td>Latch</td>
<td>9</td>
</tr>
</tbody>
</table>

The ports are shown here from the front; wire order is mirrored if looking at the back.

Software

Here is a pre-compiled .UF2 file for the Feather RP2040 DVI. Install as you would most .UF2 files: hold down the BOOT button while tapping reset or when connecting USB, and the board appears to your computer as a small flash drive called RPi-RP2. Just drag the file over and allow it some time to complete the transfer.
If the RPi-RP2 drive does not appear on your computer, try a different USB cable. Even in this modern age some devices come bundled with a budget “charge only” cable, but this needs a proper “charge and sync” cable.

### Using Pico + DV Demo Base

With two USB ports, this might be the cleanest option for using one of the supported Sony USB controllers. It also uses the super economical Pico RP2040 board for the brains...maybe you already have one!

1 x Raspberry Pi Pico H
Raspberry Pi Pico H - Pico with Headers Soldered
https://www.adafruit.com/product/5525

There are several versions of the Pico board; with or without headers, with or without wireless, any variant can work here. Wireless isn’t used in this project, but if that’s what you’ve got on hand, the board’s still compatible (only difference is you won’t see an LED “heartbeat” on this board). If your Pico board doesn’t have headers installed, you’ll need to add these, which involves a bit of soldering (also, see notes later about headers and gamepads).

1 x Pico DV Demo Base
Pimoroni Pico DV Demo Base - RP2040 DVI Multimedia Board
https://www.adafruit.com/product/5674

### Input Options

Wii Classic Controller is not supported with this hardware combo.

To use one of the supported Sony USB controllers:

- A common USB-OTG cable is needed, this is even something we stock (). USB micro-B plug, USB-A socket. Connect this to the USB port on the Pico RP2040 board.
- Power the circuit through the Pico DV Demo Base’s USB micro-B socket.

To use NES or SNES compatible gamepads:

- This is not plug-and-play and will require a little soldering and creativity.
• If your Pico board already has headers installed, you might need to “blue wire” to a few pins there. Or...
• If your Pico does not yet have headers, you’ll need to add these...but consider using extra-long male headers () so you have pins facing both up and down, and then use female jumper wires () to interface to the controller.
• Wire up a NES or SNES controller port (these can be found on eBay)...or, if you don’t mind sacrificing a junky controller, cut the plug off and wire directly between the circuits.

• Connections are:

<table>
<thead>
<tr>
<th>NES/SNES Controller</th>
<th>Pico RP2040 Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>3V3</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Clock</td>
<td>GP6</td>
</tr>
<tr>
<td>Data</td>
<td>GP7</td>
</tr>
<tr>
<td>Latch</td>
<td>GP8</td>
</tr>
</tbody>
</table>

The ports are shown here from the front; wire order is mirrored if looking at the back.
As with the Feather on the prior page, there’s no One Hard Rule to follow for this build. Here are a couple variants: one with a USB-OTG cable, another wired to an NES controller port (notice that long pin headers were soldered to the Pico RP2040 board in the latter case — it fits in a socket and has pins going “up” for jumper wires ()).

With a Pico H board, USB-OTG cable and USB controller, this is the only zero-soldering option. Every other configuration requires a little bit here and there.

Software

Here is a pre-compiled .UF2 file for the Pico DV Demo Base. Install as you would most .UF2 files: hold down the BOOT button while tapping RUN or when connecting USB, and the board appears to your computer as a small flash drive called RPi-RP2. Just drag the file over and allow it some time to complete the transfer.

```
piconesPlusPimoroniDV.uf2
```

If the RPi-RP2 drive does not appear on your computer, try a different USB cable. Even in this modern age some devices come bundled with a budget “charge only” cable, but this needs a proper “charge and sync” cable.
Using Pico + Breakouts

This combination uses breadboardable components if you’d just like to mess around a bit before recycling everything into another project. There’s also an option to use most of the same components in a custom printed circuit board — this is the best and cleanest route for using a NES or SNES controller and if you want something permanent and purpose-built for the task.

Here are some of the parts used in this version of the project. You might already have some of these around! Additionally, common soldering bits-and-bobs (iron, solder, flush cutters, etc.) are needed.

1 x Pico RP2040 [Raspberry Pi Pico RP2040](https://www.adafruit.com/product/4864)  
There are several versions of the Pico board; with or without headers, with or without wireless, any variant can work here. Wireless isn’t used in this project, but if that’s what you’ve got on hand, the board’s still compatible (only difference is you won’t see an LED “heartbeat” on this board).

This is one part that can’t move over directly; the breadboard build requires headers, while the PCB build relies on the castellated pads and must sit flat. Fortunately Pico boards are super affordable like popcorn, and there are tons of other cool projects to make if you end up with a spare.

1 x Micro SD Breakout Board [Adafruit Micro SD SPI or SDIO Card Breakout Board](https://www.adafruit.com/product/4682)  
Other microSD adapter boards should work fine for breadboard use. Some might have 5V input instead of 3.3V; power these from the Pico’s VBUS pin rather than 3V3.
For the custom PCB, confirm your microSD adapter has a compatible pinout, or use the Adafruit breakout board specifically.

1 x **DVI Breakout Board**
Adafruit DVI Breakout Board

1 x **HDMI Cable**
HDMI Cable - 1 meter

Nothing special about this HDMI cable. If you have something around already, use that!

Additional parts specific to different builds or controllers are explained as we go...

**Input Options**

Wii Classic Controller is not supported with this hardware combo.

To use one of the supported Sony USB controllers:

- A suitable USB-OTG splitter cable is needed; this is something we don’t stock but can be found online. Look for USB micro-B plug for the Pico board, USB-A socket for the controller, and then whatever connector you would prefer for connecting power (USB micro-B and USB-C are common).

To use NES or SNES compatible gamepads:

- Wire up a NES or SNES controller port (these can be found on eBay)...or, if you don’t mind sacrificing a junky controller, cut the plug off and wire directly between the circuits.

The USB-OTG cable can be omitted if using an NES/SNES gamepad. Conversely, the NES/SNES port (and connections described below) can be omitted if using a USB controller.

**Breadboard Method**

Breadboarding’s great for temporary projects. If you play around with this and decide you really like it and want something permanent, almost everything can be moved over to the custom PCB explained later.
Note: for the breadboard build, the Pico board must have pin headers. The PCB build must have no headers (but the other boards do use headers in either case).

Here’s some additional parts used in the breadboard build. If you’ve been doing electronics for a while there’s a good chance you have most of this around already:

1 x **Breadboard**
Full Sized Breadboard

[View on Adafruit](https://www.adafruit.com/product/239)

1 x **Long Jumper Wires**
Male/Male Jumper Wires - 6" (150mm)

[View on Adafruit](https://www.adafruit.com/product/1957)

1 x **Short Jumper Wires**
Male/Male Jumper Wires - 3" (75mm)

[View on Adafruit](https://www.adafruit.com/product/1956)

1 x **Tactile Button**
Tactile Button switch (6mm)

[View on Adafruit](https://www.adafruit.com/product/367)

A visual breadboard representation:

![Breadboard Diagram](image)

Notice the microSD clock SO and SI pins cross on their way over; these are not in the same order on both boards.

The NES or SNES controller port is not directly breadboard-friendly; you’ll need to solder on some jumper wires to make this work. The ports are shown here from the front; wire order is mirrored if looking at the back.

A schematic view if that’s easier to follow:
Or a pin-by-pin list of connections:

<table>
<thead>
<tr>
<th>Pico RP2040</th>
<th>microSD Breakout</th>
<th>NES/SNES Controller</th>
<th>DVI Breakout</th>
</tr>
</thead>
<tbody>
<tr>
<td>3V3</td>
<td>3V</td>
<td>VCC</td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>GP2</td>
<td>CLK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP3</td>
<td>SI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP4</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP5</td>
<td>CS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP6</td>
<td></td>
<td>Clock</td>
<td></td>
</tr>
<tr>
<td>GP7</td>
<td></td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>GP8</td>
<td>Latch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP12</td>
<td>0+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP13</td>
<td>0–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP14</td>
<td>TXC+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP15</td>
<td>TC–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP16</td>
<td>TX2+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP17</td>
<td>TX2–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP18</td>
<td>1+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP19</td>
<td>1–</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Any variety of momentary button/switch can be connected between Pico RUN and any available GND pin to provide a reset button.
Custom PCB Method

Connecting everything on a purpose-made circuit board makes this project robust. An NES or SNES controller can be used with a solid connection to a physical port. Mounting holes facilitate installing in a case if you like.

This shows both the NES and SNES ports installed. You don’t need to do this; one or other is fine, it’s up to you! Unfortunately there is no two-player support, but for single player you can use whichever gamepad type you prefer.

1 x SPDT Switch
Breadboard-friendly SPDT Slide Switch

The custom circuit board can be ordered from services such as PCBWay, using files provided below. At the time this was written, a set of 5 boards cost $22 USD plus postage. Most other small-run PCB services should be able to work with these files (for a little more, OSH Park has a very good reputation with hobbyists, and might arrive sooner for US addresses). Be patient and allow ample time for panelization, production and shipping.

To get the PCB files, click the following link:

Download PicoNES PCB ZIP File

The PCB production service — PCBWay, OSH Park or other — will ask for a ZIP file. Upload the still-compressed ZIP file and they should be able to decode the board size and details and provide a quote.
If you previously built the breadboard version, the microSD and DVI breakout boards will move right over. After soldering in place, the long pins can be trimmed on the underside of the board. A fresh, headerless Pico board must be used as this sits flat against the PCB...tack down two pins at opposite corners and the rest should proceed smoothly.

Software

Here is a pre-compiled .UF2 file for the Pico/breakout combo. Install as you would most .UF2 files: hold down the BOOT button when connecting USB, and the board appears to your computer as a small flash drive called RPi-RP2. Just drag the file over and allow it some time to complete the transfer.

piconesPlusAdaFruitDVISD.uf2

If the RPi-RP2 drive does not appear on your computer, try a different USB cable. Even in this modern age some devices come bundled with a budget “charge only” cable, but this needs a proper “charge and sync” cable.