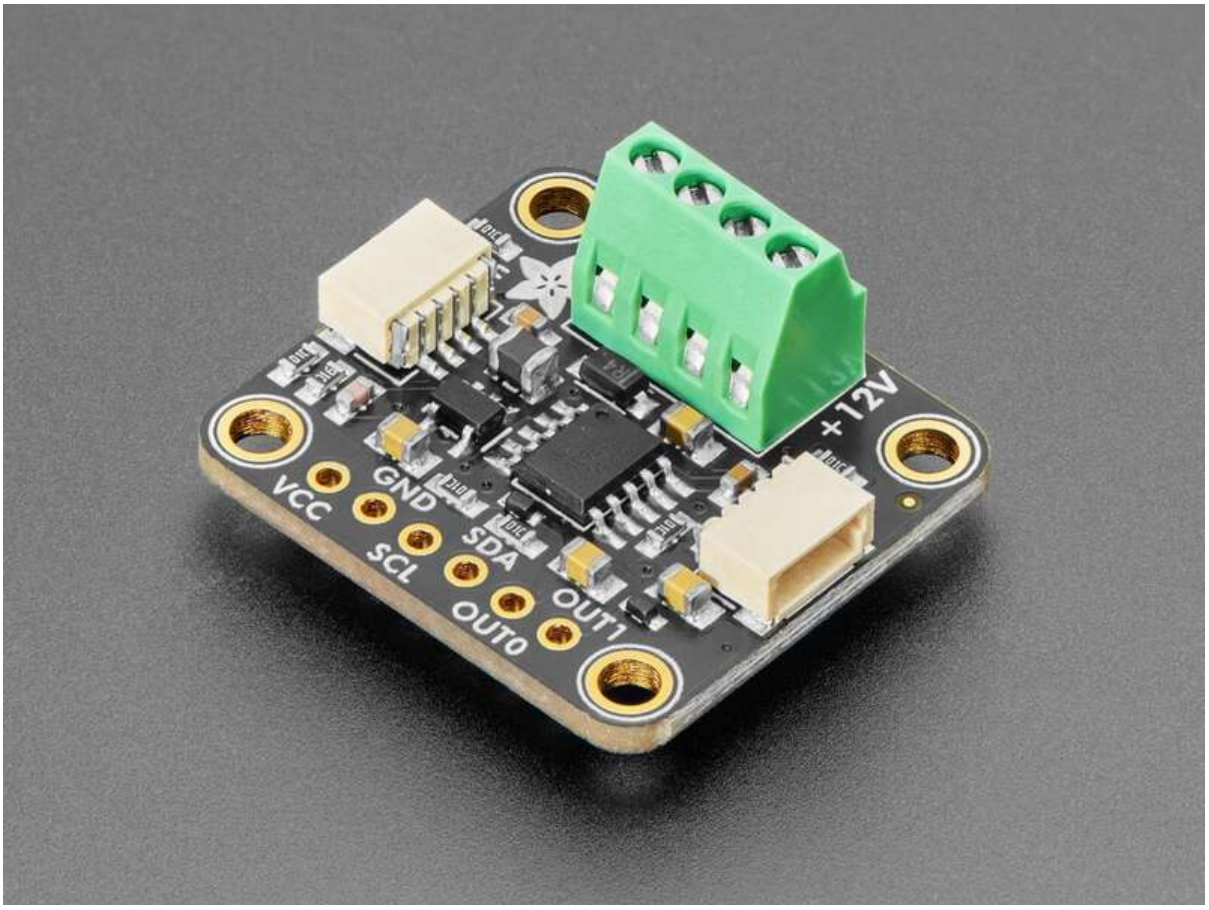
Adafruit GP8403 I2C DAC with 12V Booster

Created by Liz Clark



https://learn.adafruit.com/adafruit-gp8403-i2c-dac-with-12v-booster

Last updated on 2026-08-27 11:11:47 PM UTC

Table of Contents

Overview 3

Pinouts 6

- Power Pins
- I2C Logic Pins
- DAC Outputs
- Address Pins
- Power LED and LED Jumper

CircuitPython and Python 9

- CircuitPython Microcontroller Wiring
- Python Computer Wiring
- Python Installation of GP8403 Library
- CircuitPython Usage
- Python Usage
- Example Code

Python Docs 13

Arduino 14

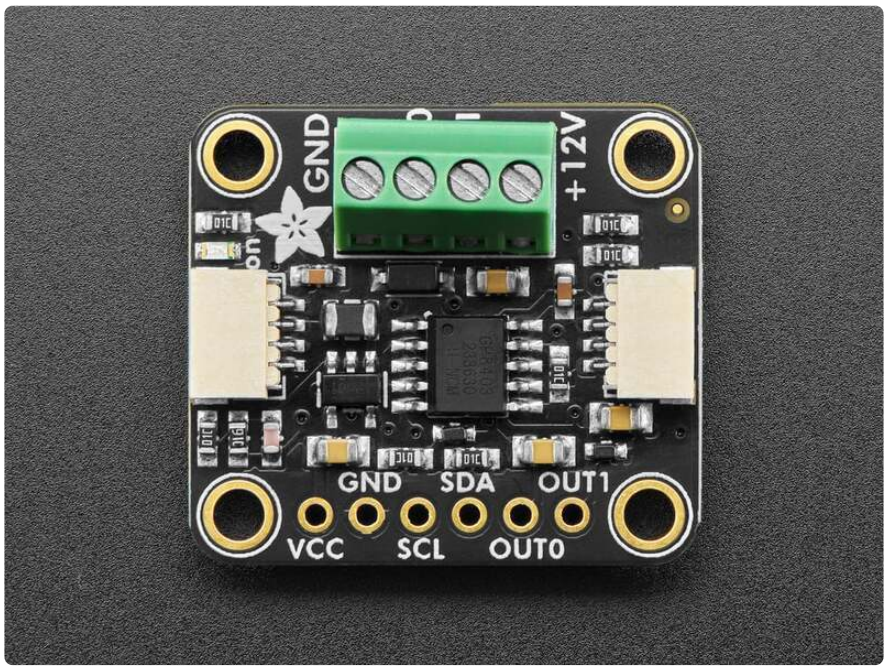
- Wiring
- Library Installation
- Example Code

Arduino Docs 18

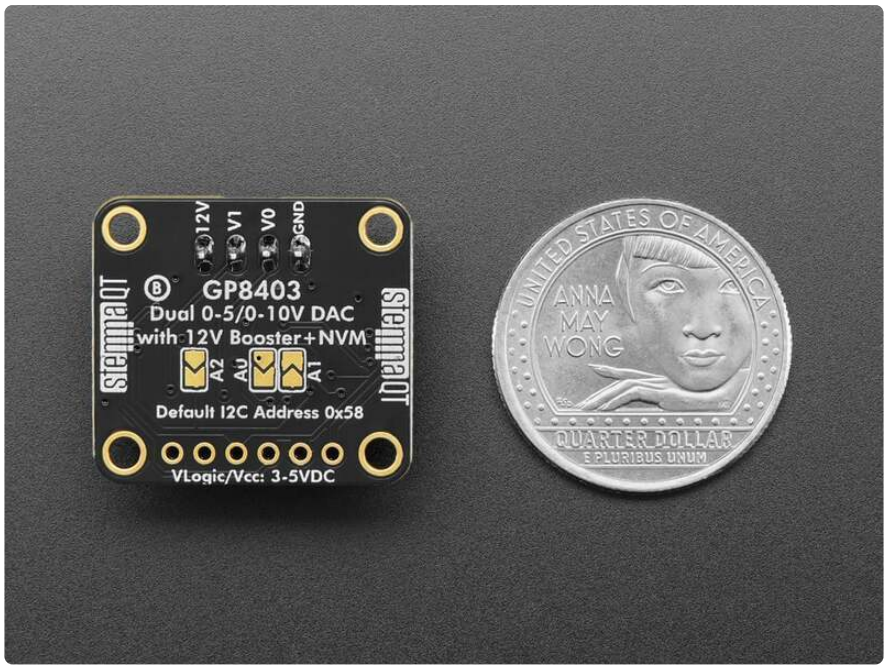
Downloads 18

- Files
- Schematic and Fab Print

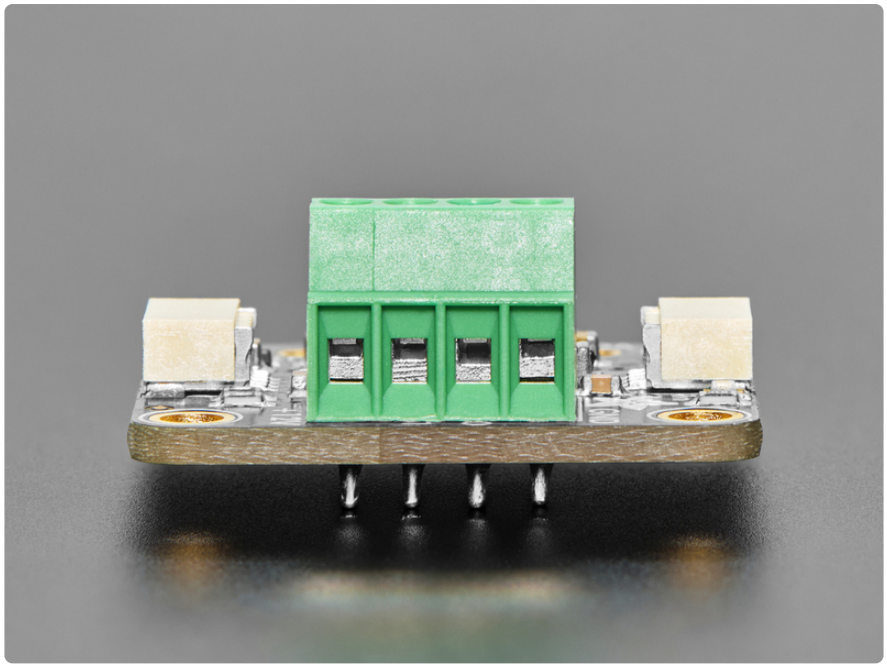
Overview



We stock a bunch of DAC boards here at the 'fruit. These are chips that will take I2C or SPI and pop out an analog voltage. We have single channels like the MCP4725 (http://adafru.it/935) and even quad channel like MCP4728 (http://adafru.it/4470), but one thing most DACs have in common is they max out at whatever the input power is for the chip, which is usually 3V or 5V. The **Adafruit GP8403 Breakout Board** is a funky exception - it has the ability to output 0-10V analog voltages, so it's useful for devices that have wide voltage input ranges - like audio or op-amps. Note that the DAC outputs have 10uF caps on them (as indicated by the datasheet), so they aren't going to be terribly fast - best for bias setting maybe, instead of trying to generate fast sine waves!



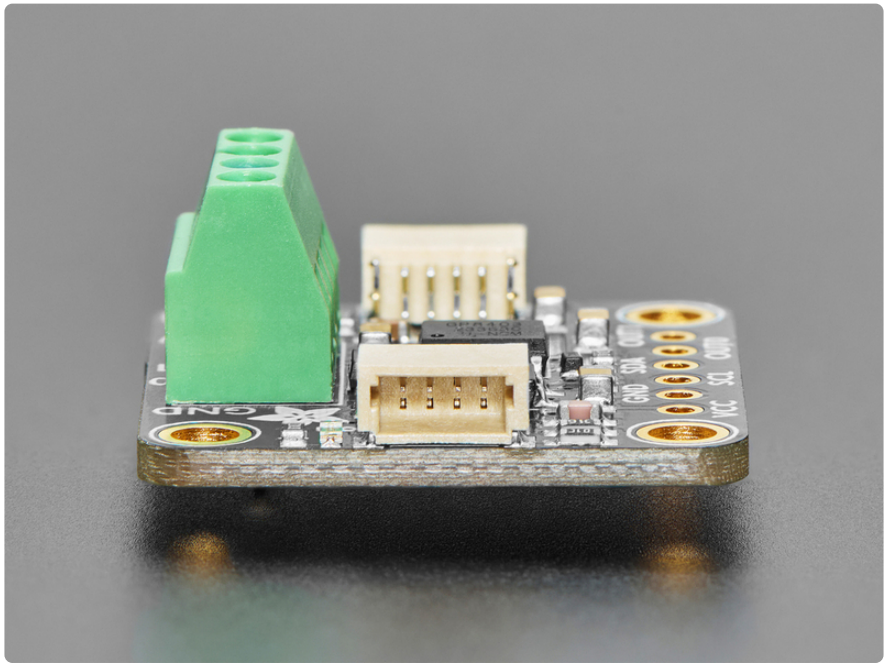
The GP8402 chip has a built-in I2C-to-analog signal converter that linearly converts 12-bit digital values from 0x000 to 0xFFF into two independent analog voltage outputs of 0–5V or 0–10V, with an output voltage error of 0.2%. This chip is great for when you need wide / high voltage range outputs thanks to its built-in booster, we also provide the 12V output from the small on-board DC/DC converter. You can grab up to 100mA if you need to power something externally.



On the back are three A0, A1, and A2 address selection jumpers, so you can connect up to 8 of these DACs on one I2C bus, for up to 16 DAC outputs. Also included is a 6-

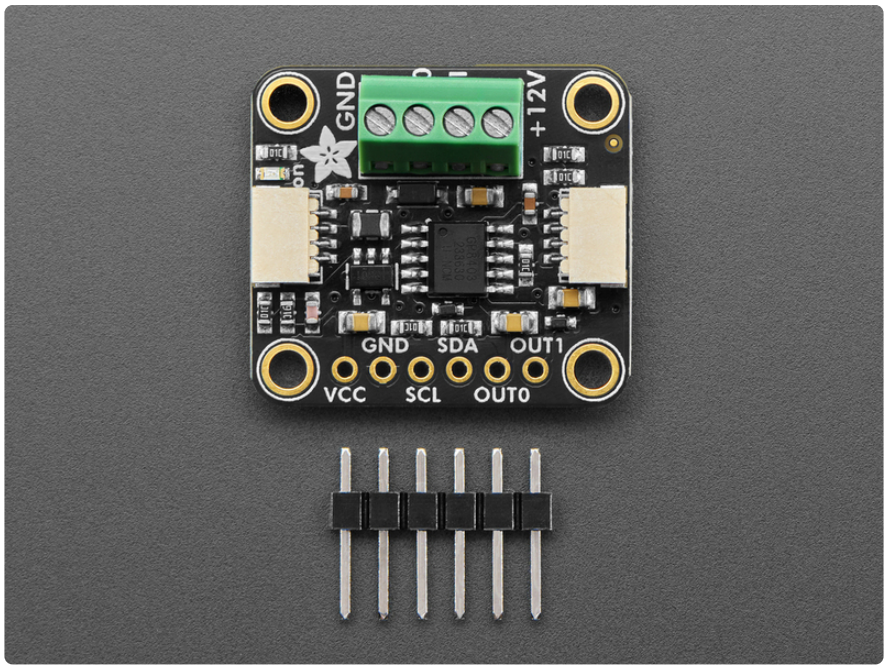
pin header, for use in a breadboard. Works with both 3.3V or 5V logic, which is handled separately from the boosted DAC outputs.

This chip does have a non-volatile-memory capability, so it will boot up into a prior-set voltage. However, it uses a funky bit-banging technique on the I2C lines so while we did get it to work on an ATmega328, you should definitely verify it on your platform since some devices will not let you 'take over' the I2C pins for bit-banging!



We have easy-to-use Arduino and Python/CircuitPython libraries and examples (https://adafru.it/1aI3) that can be used with just about any microcontroller or microcomputer with I2C host - from the classic Arduino UNO-compatible to Raspberry Pi, Pico, ESP32 and beyond.

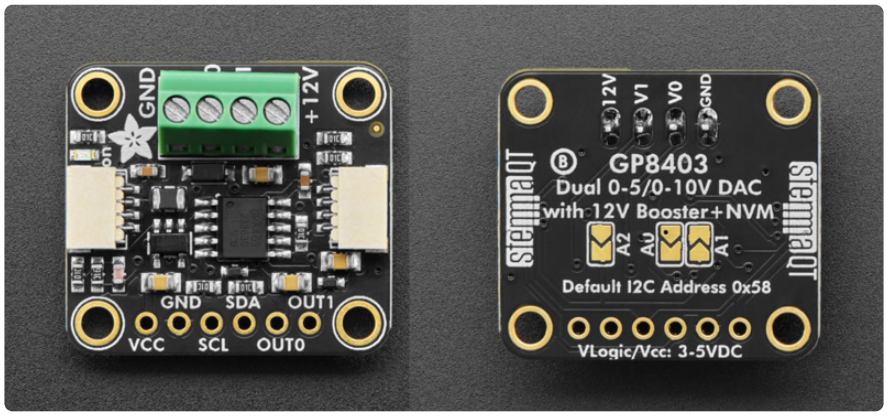
Comes with a bit of 0.1" standard header in case you want to use it with a breadboard or perfboard. Four mounting holes for easy attachment. There's a 4-pin 2.5mm terminal block already soldered on the PCB.



To get you going fast, we spun up a custom-made PCB in the **STEMMA QT** form factor (https://adafru.it/LBQ), making it easy to interface with. The STEMMA QT connectors (https://adafru.it/JqB) on either side are compatible with the SparkFun Qwiic (https://adafru.it/Fpw) I2C connectors. This allows you to make solderless connections between your development board and the GP8403 or to chain it with a wide range of other sensors and accessories using a **compatible cable** (https://adafru.it/JnB).

QT Cable is not included, but we have a variety in the shop (https://adafru.it/17VE).

Pinouts



The default I2C address is **0x58**.

Power Pins

- **VCC** - this is the power pin. Since the DAC chip uses 3-5 VDC to power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V.
- **+12V** - this is the 12V output from the small on-board DC/DC converter. You can grab up to 100mA if you need to power something externally.
- **GND** - common ground for power and logic.

I2C Logic Pins

- **SCL** - I2C clock pin, connect to your microcontroller's I2C clock line. This pin can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller's I2C data line. This pin can use 3-5V logic, and there's a **10K pullup** on this pin.
- STEMMA QT **(https://adafru.it/Ft4) -** These connectors allow you to connectors to dev boards with **STEMMA QT** connectors or to other things with various associated accessories (https://adafru.it/Ft6)

DAC Outputs

- **OUT0/V0** and **OUT1/V1** - The analog voltage outputs from the DAC. They are available along the bottom edge of the board (**OUT0** and **OUT1**) or via the terminal block (**V0** and **V1**). You can get an output of 0-5V or 0-10V, with an output voltage error of 0.2%. Note that the DAC outputs have 10uF caps on them (as indicated by the datasheet), so they aren't going to be terribly fast.

Address Pins

On the back of the board are **three address jumpers**, labeled **A0**, **A1**, and **A2**. These jumpers allow you to chain up to 8 of these boards on the same pair of I2C clock and data pins. To do so, you solder the jumpers "closed" by connecting the two pads.

The default I2C address is **0x58**. The other address options can be calculated by “adding” the **A0/A1/A2** to the base of **0x58**.

A0 sets the lowest bit with a value of **1**, **A1** sets the next bit with a value of **2** and **A2** sets the next bit with a value of **4.** The final address is **0x58** + **A2** + **A1** + **A0** which would be **0x5F**.

So for example if **A2** is soldered closed and **A0** is soldered closed, the address is **0x58** + **4** + **1** = **0x5D**.

If only **A0** is soldered closed, the address is **0x58** + **1** = **0x59**

If only **A1** is soldered closed, the address is **0x58** + **2** = **0x5A**

If only **A2** is soldered closed, the address is **0x58** + **4** = **0x5C**

The table below shows all possible addresses, and whether the jumper(s) should be high (closed) or low (open).

ADDR	A0	A1	A2
0x58	L	L	L
0x59	H	L	L
0x5A	L	H	L
0x5B	H	H	L
0x5C	L	L	H
0x5D	H	L	H
0x5E	L	H	H
0x5F	H	H	H

Power LED and LED Jumper

- **Power LED -** In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled **on**. It is a green LED.
- **LED jumper** - This jumper, labeled **LED** on the board silk, is located on the back of the board, on the right. Cut the trace on this jumper to cut power to the "on" LED.

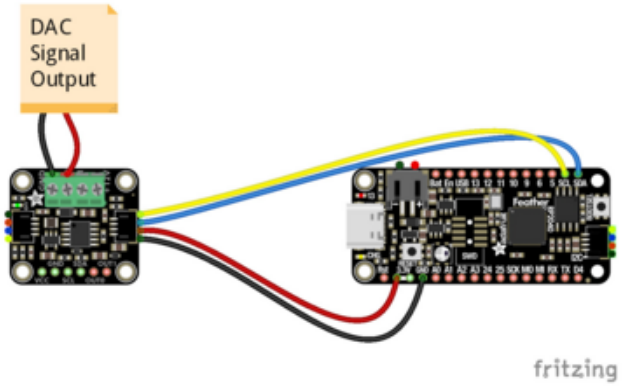
CircuitPython and Python

It's easy to use the **GP8403** with Python or CircuitPython, and the Adafruit_CircuitPython_GP8403 (https://adafru.it/1al4) module. This module allows you to easily write Python code to control the DAC.

You can use this driver with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

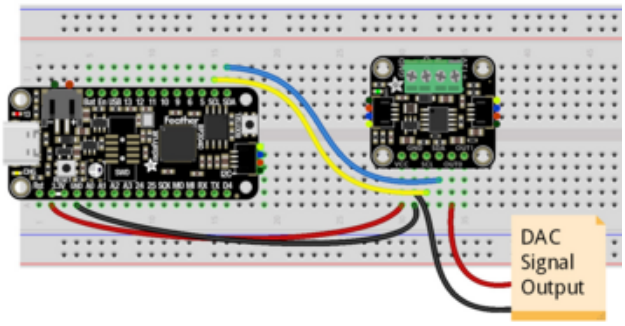
First wire up the breakout to your board exactly as follows. The following is the breakout wired to a Feather RP2040 using the STEMMA connector:



Board STEMMA 3V to **breakout VIN (red wire)**
Board STEMMA GND to **breakout GND (black wire)**
Board STEMMA SCL to **breakout SCL (yellow wire)**
Board STEMMA SDA to **breakout SDA (blue wire)**

The signal will be output from the DAC **V0 terminal block** and **GND**.

The following is the breakout wired to a Feather RP2040 using a solderless breadboard:



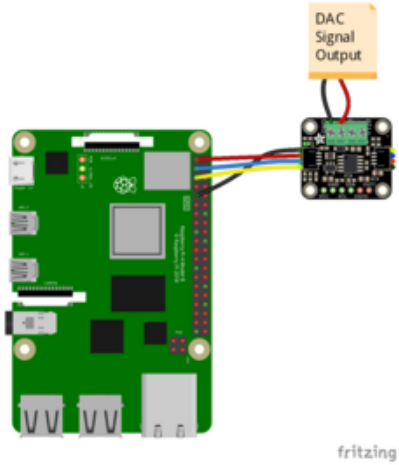
Board 3V to **breakout VIN (red wire)**
Board GND to **breakout GND (black wire)**
Board SCL to **breakout SCL (yellow wire)**
Board SDA to **breakout SDA (blue wire)**

The signal will be output from the DAC **OUT0** pin and **GND**.

Python Computer Wiring

Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported](https://adafru.it/BSN) (https://adafru.it/BSN).

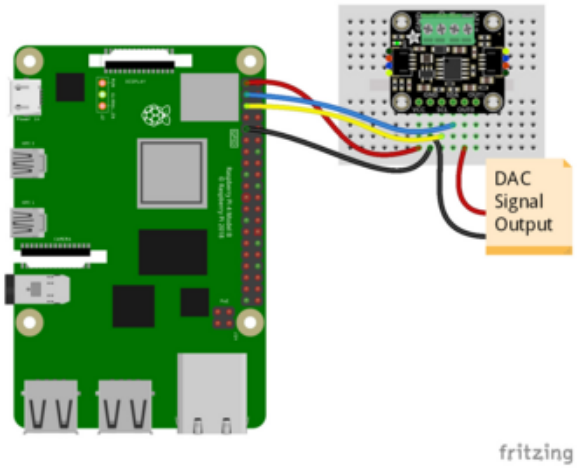
Here's the Raspberry Pi wired with I2C using the STEMMA connector:



Pi 3V to **breakout VIN (red wire)**
Pi GND to **breakout GND (black wire)**
Pi SCL to **breakout SCL (yellow wire)**
Pi SDA to **breakout SDA (blue wire)**

The signal will be output from the DAC **V0 terminal block** and **GND.**

Here's the Raspberry Pi wired with I2C using a solderless breadboard:



Pi 3V to **breakout VIN (red wire)**
Pi GND to **breakout GND (black wire)**
Pi SCL to **breakout SCL (yellow wire)**
Pi SDA to **breakout SDA (blue wire)**

The signal will be output from the DAC **V0** pin and **GND**.

Python Installation of GP8403 Library

You'll need to install the **Adafruit_Blinka** library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `pip3 install adafruit-circuitpython-gp8403`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython Usage

To use with CircuitPython, you need to first install the **Adafruit_CircuitPython_GP8403** library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the **code.py** file in a zip file. Extract the contents of the zip file, and copy the **entire lib folder** and the **code.py** file to your **CIRCUITPY** drive.

Your **CIRCUITPY/lib** folder should contain the following folders and file:

- **adafruit_bus_device/**
- **adafruit_register/**

- adafruit_gp8403.mpy

Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing **code.py** with whatever you named the file:

```
python3 code.py
```

Example Code

If running CircuitPython: Once everything is saved to the **CIRCUITPY** drive, connect to the serial console (https://adafru.it/Bec) to see the data printed out!

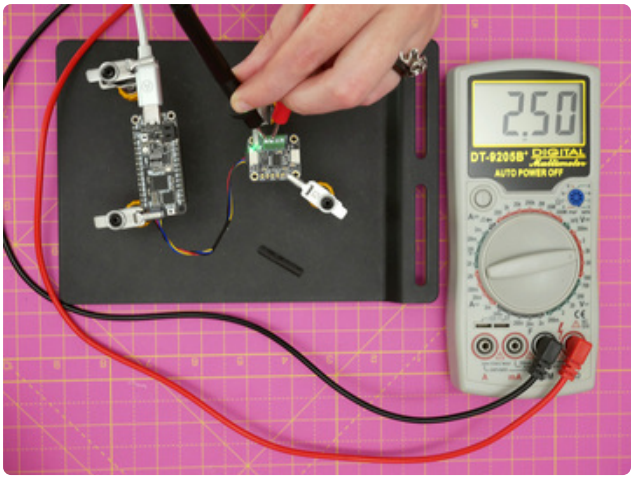
If running Python: The console output will appear wherever you are running Python.

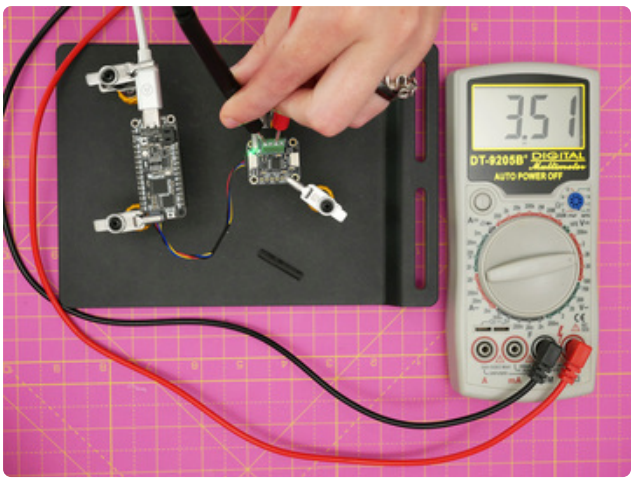
```
# SPDX-FileCopyrightText: Copyright (c) 2026 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time

import board

import adafruit_gp8403

i2c = board.I2C()
# default 0-5V range:
dac = adafruit_gp8403.GP8403(i2c)
# for 0-10V range:
# dac = adafruit_gp8403.GP8403(i2c, output_range=adafruit_gp8403.Range.RANGE_10V)

dac.channel_0.voltage = 2.5  # set channel 0 to 2.5V
dac.channel_1.voltage = 3.5  # set channel 0 to 3.5V
```

```
while True:
    # can access voltages as a tuple too (dac.voltages):
    print(f"Channel 0: {dac.voltages[0]:.01f}V")
    print(f"Channel 1: {dac.voltages[1]:.01f}V")
    time.sleep(1)
```

https://github.com/adafruit/Adafruit_CircuitPython_GP8403/blob/main/examples/gp8403_simpletest.py

```
CircuitPython REPL

Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Channel 0: 2.5V
Channel 1: 3.5V
Channel 0: 2.5V
Channel 1: 3.5V
Channel 0: 2.5V
Channel 1: 3.5V
```





The GP8403 is initialized over I2C. Then, it sets output 0 to 2.5V and output 1 to 3.5V, +/-0.2%. If you connect the DAC outputs to a multimeter or scope, you will be able to confirm the voltage readings.

Python Docs

Python Docs (https://adafru.it/1aI0)

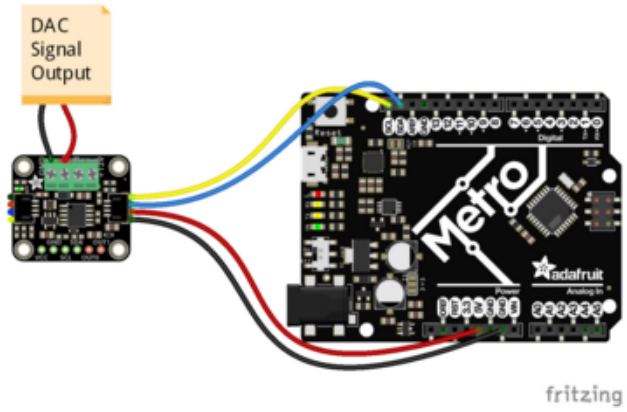
Arduino

Using the GP8403 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller, installing the Adafruit_GP8403 (https://adafru.it/1aI5) library, and running the provided example code.

Wiring

Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the breakout VIN.

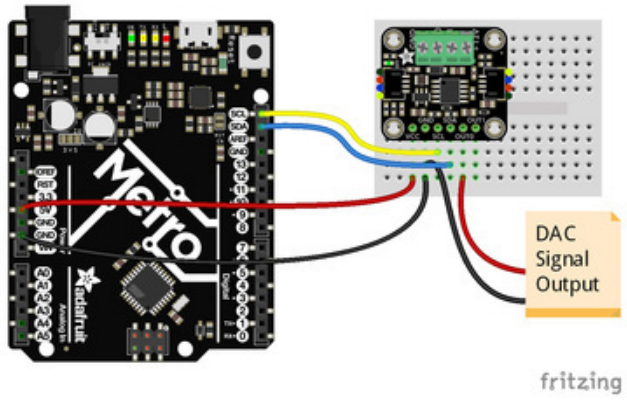
Here is an Adafruit Metro wired up to the breakout using the STEMMA QT connector:



Board 5V to **breakout VIN (red wire)**
Board GND to **breakout GND (black wire)**
Board SCL to **breakout SCL (yellow wire)**
Board SDA to **breakout SDA (blue wire)**

The signal will be output from the DAC **V0** terminal block with **GND**.

Here is an Adafruit Metro wired up using a solderless breadboard:

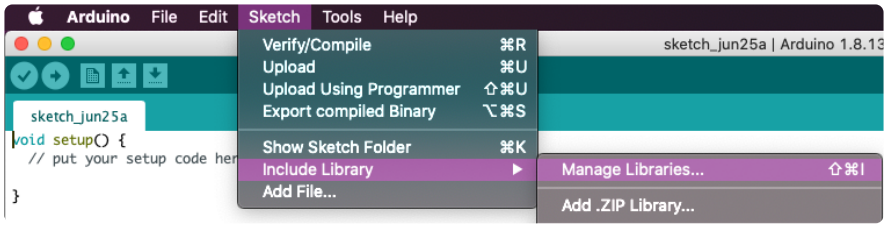


Board 5V to **breakout VIN (red wire)**
Board GND to **breakout GND (black wire)**
Board SCL to **breakout SCL (yellow wire)**
Board SDA to **breakout SDA (blue wire)**

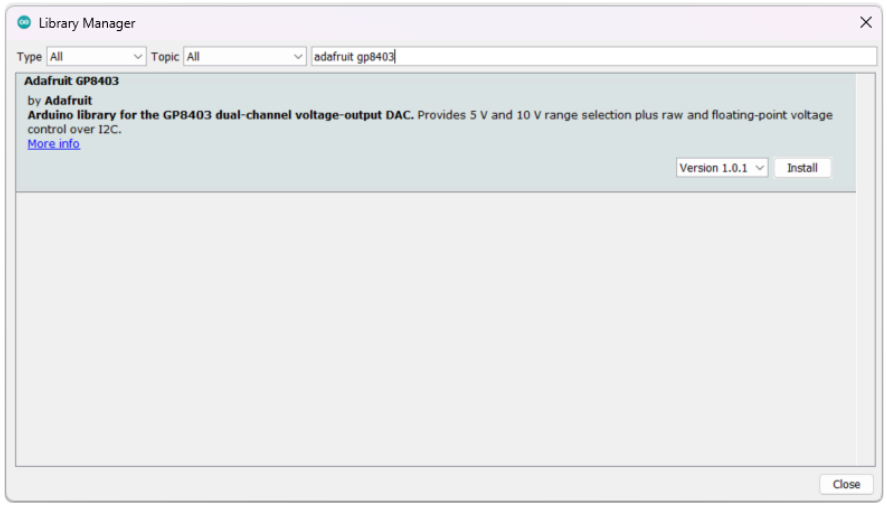
The signal will be output from the DAC **OUT0** pin and **GND**.

Library Installation

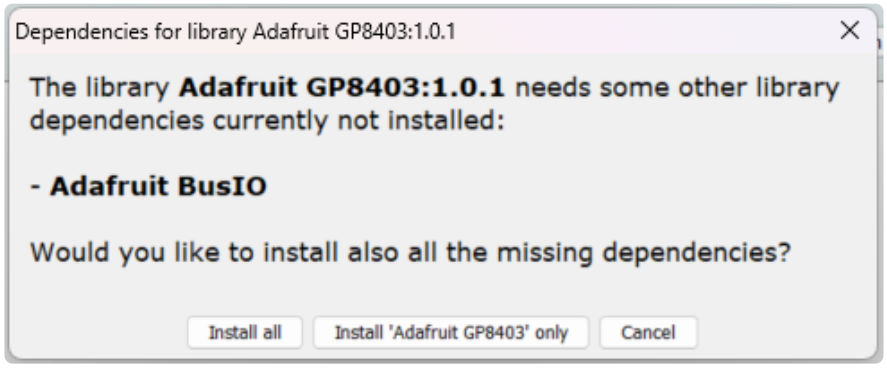
You can install the **Adafruit_GP8403** library for Arduino using the Library Manager in the Arduino IDE.



Click the **Manage Libraries ...** menu item, search for **Adafruit_GP8403**, and select the **Adafruit GP8403** library:



If asked about dependencies, click "Install all".



If the "Dependencies" window does not come up, then you already have the dependencies installed.

Text emphasized with a yellow exclamation:	If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

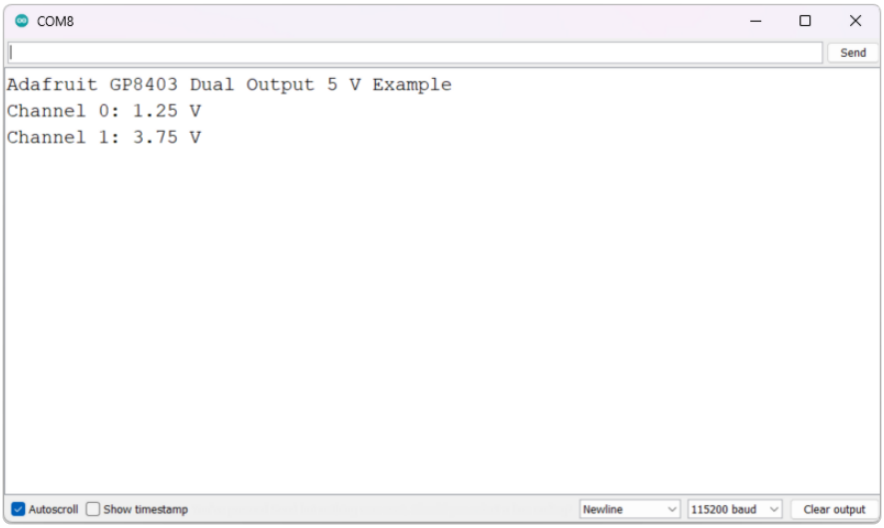
Example Code

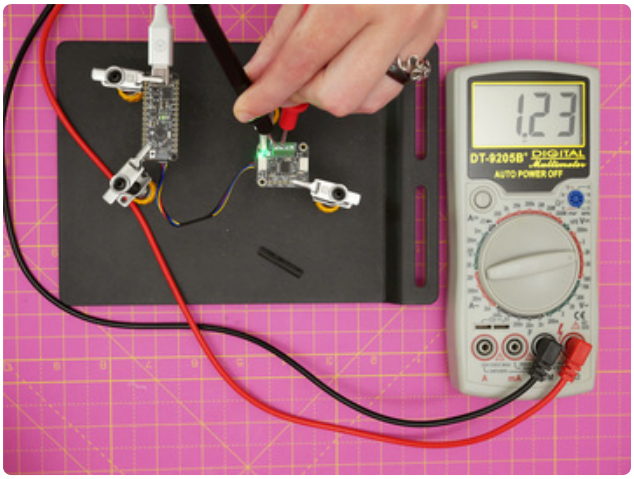
```
/*
 * Adafruit GP8403 Dual Output 5 V Example
 *
 * Selects the 0-5 V range and sets both output channels.
 */

#include <Adafruit_GP8403.h>

Adafruit_GP8403 gp8403;

void setup() {
  Serial.begin(115200);
  while (!Serial) {
    delay(10);
  }
  Serial.println("Adafruit GP8403 Dual Output 5 V Example");

  if (!gp8403.begin()) {
    Serial.println("Could not find GP8403");
    while (1) {
      delay(10);
    }
  }

  if (!gp8403.setOutputRange(GP8403_RANGE_5V)) {
    Serial.println("Could not select the 5 V output range");
    while (1) {
      delay(10);
    }
  }

  if (!gp8403.setVoltages(1.25, 3.75)) {
    Serial.println("Could not set both output voltages");
    while (1) {
      delay(10);
    }
  }

```

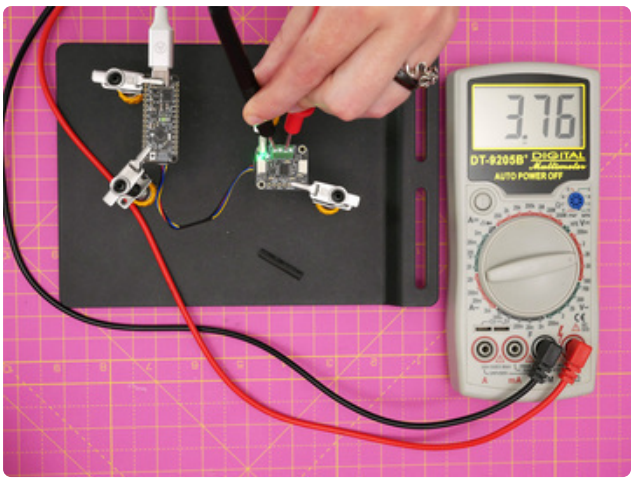
```
39    // getVoltage() returns cached values, not hardware readback.
40    Serial.print("Channel 0: ");
41    Serial.print(gp8403.getVoltage(0), 2);
42    Serial.println(" V");
43    Serial.print("Channel 1: ");
44    Serial.print(gp8403.getVoltage(1), 2);
45    Serial.println(" V");
46  }
47
48  void loop() { delay(1000); }
```

https://github.com/adafruit/Adafruit_GP8403/blob/main/examples/dual_output_5v/dual_output_5v.ino

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You'll see the GP8403 recognized over I2C.







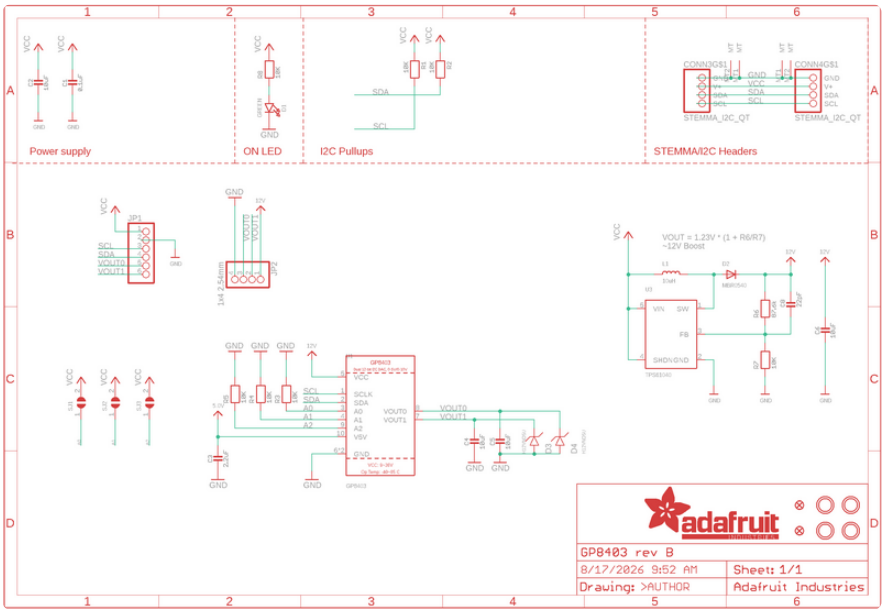
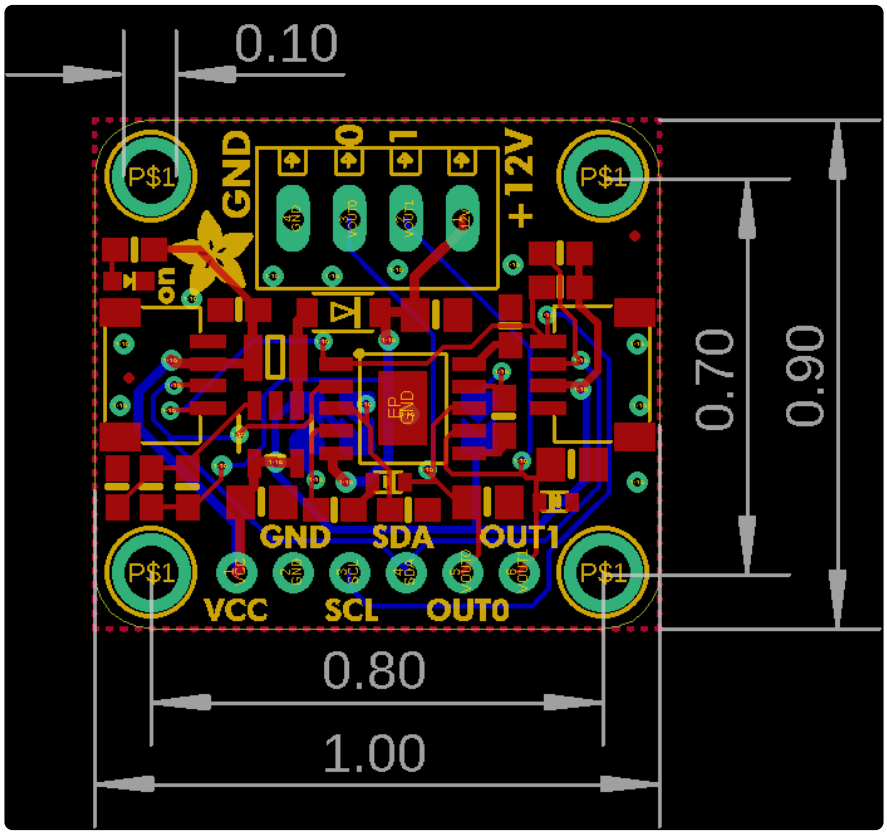
If you connect the DAC output to a multimeter or scope, you will be able to measure 1.25V on output 0 and 3.75V on output 1 +/-0.2%.

Arduino Docs

Arduino Docs (https://adafru.it/1aHZ)

Downloads

Files

- GP8403 Datasheet (https://adafru.it/1al6)
- EagleCAD PCB Files on GitHub (https://adafru.it/1al7)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/1al8)

Schematic and Fab Print