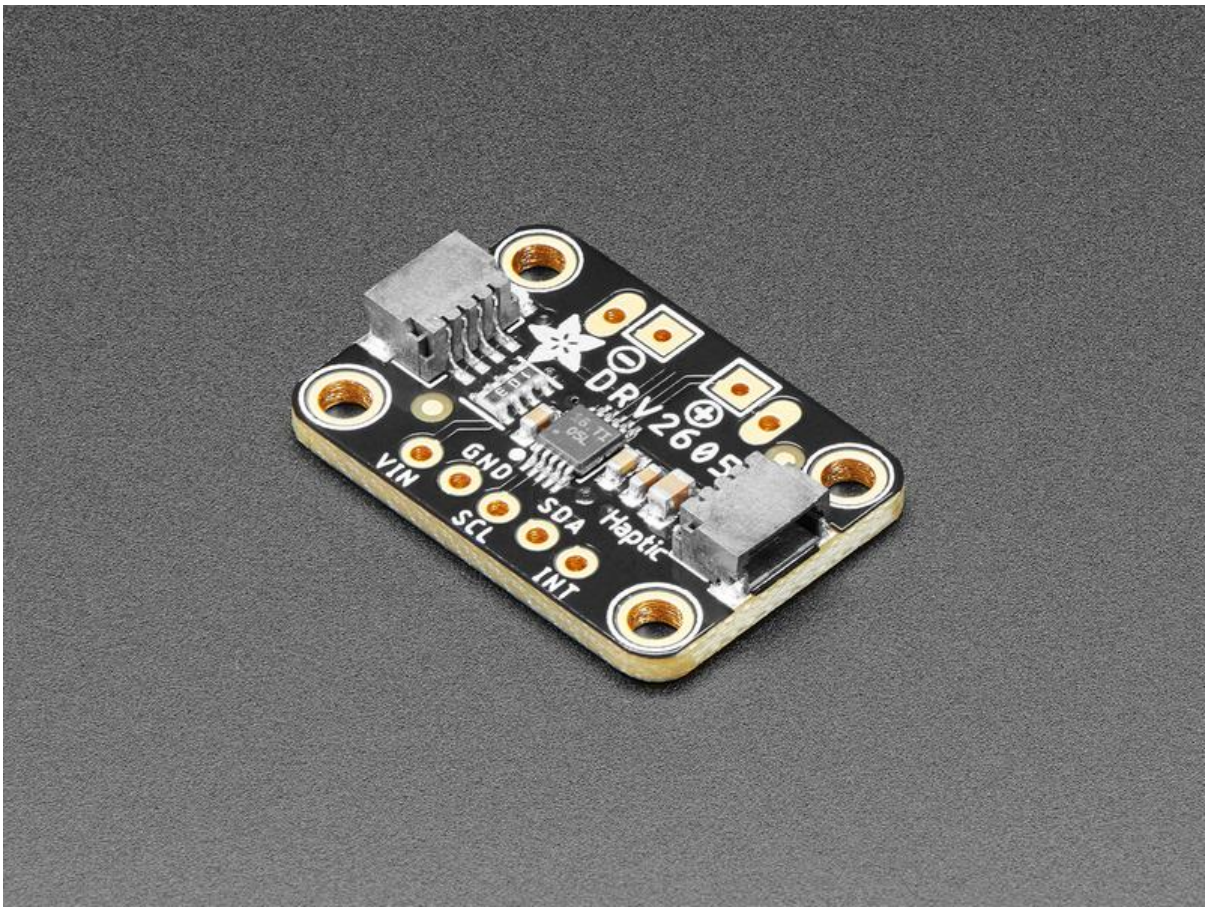
Adafruit DRV2605L Haptic Controller Breakout

Created by lady ada

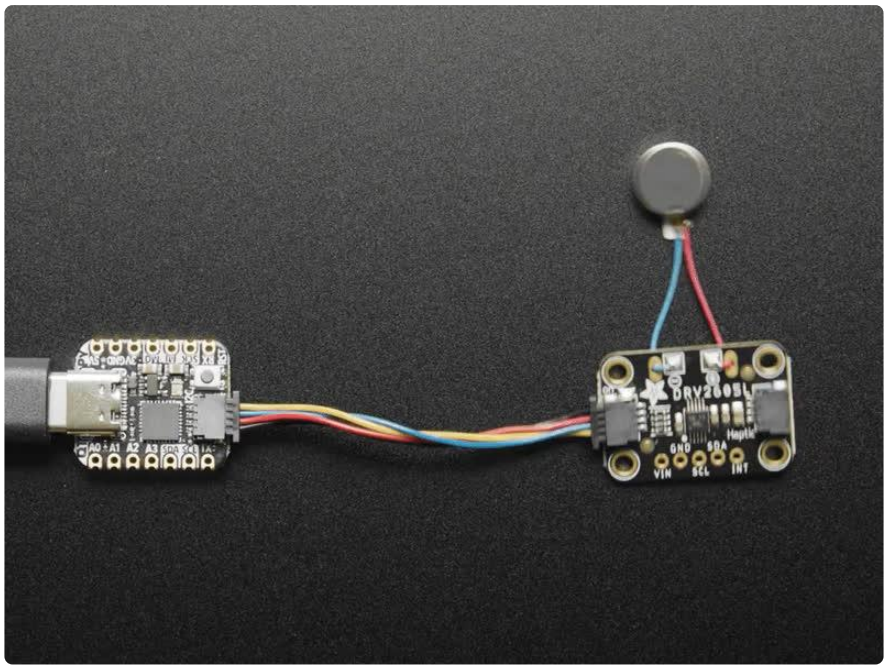


https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout

Last updated on 2022-12-01 02:21:03 PM EST

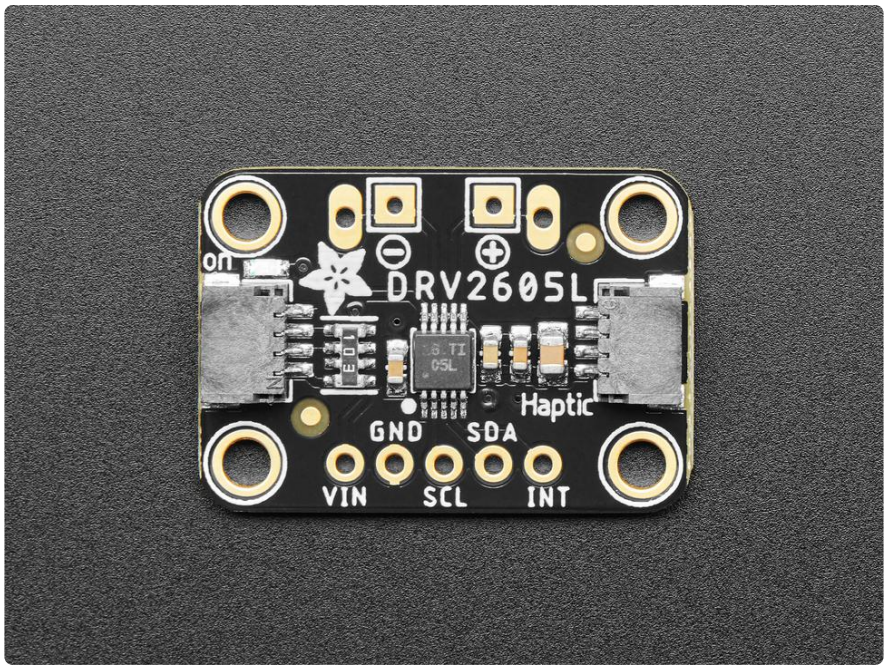
Table of Contents

Overview



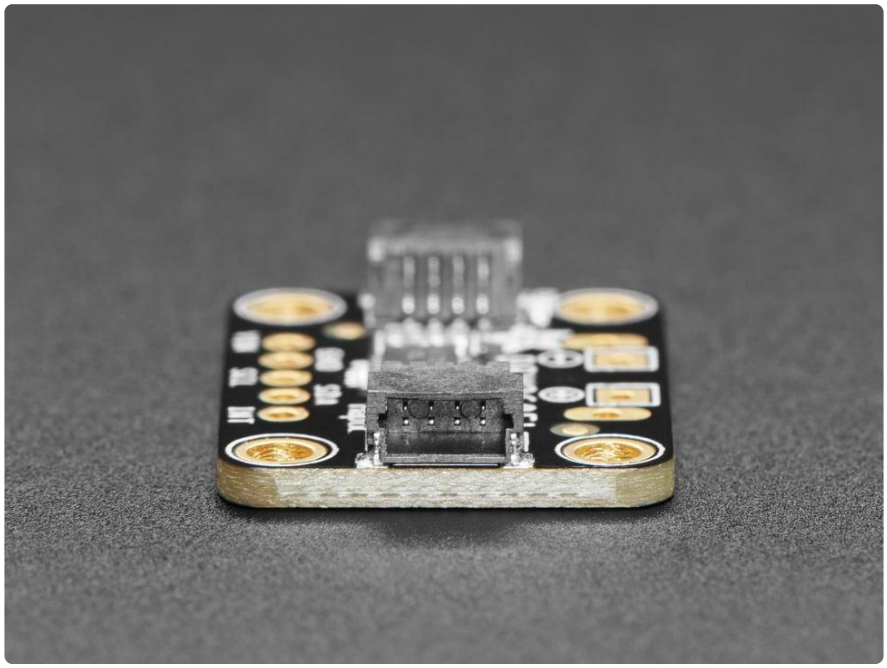
The DRV2605 from TI is a fancy little motor driver. Rather than controlling a stepper motor or DC motor, it's designed specifically for controlling haptic motors - buzzers and vibration motors. Normally one would just turn those kinds of motors on and off, but this driver has the ability to have various effects when driving a vibe motor. For example, ramping the vibration level up and down, 'click' effects, different buzzer levels, or even having the vibration follow a musical/audio input.

This chip is controlled over I2C - after initialization, a 'string' of multiple effects can be strung together in the chips memory and then triggered to actuate in a row. The built in effects are much much nicer than just 'on' and 'off' and will make your haptic project way nicer feeling.



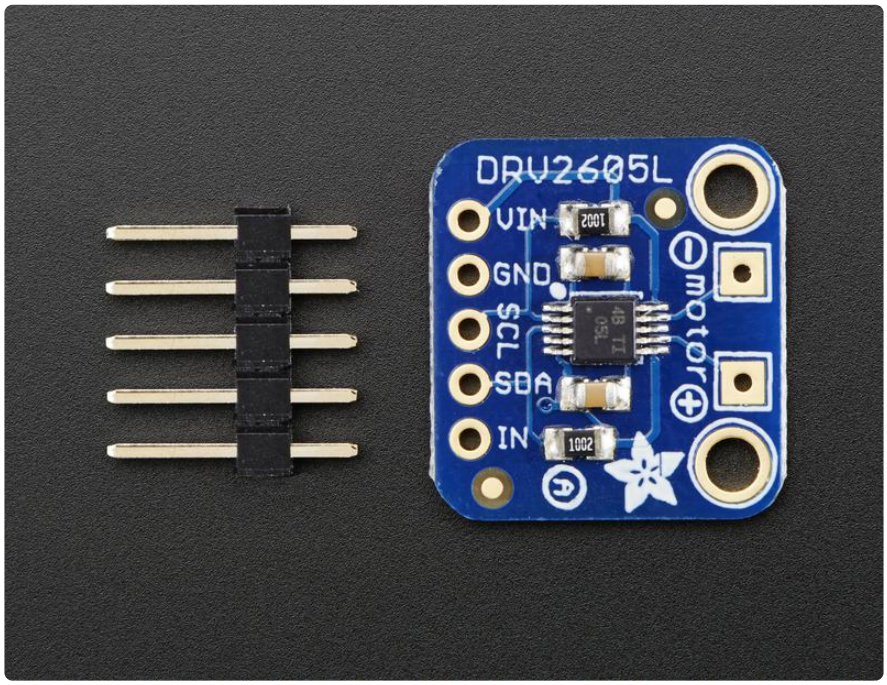
According to the product page, it can be used with both LRA (Linear Resonance Actuator) and ERM (Eccentric Rotating Mass) type motors but we have only used it with our little vibration pancake ERM. ()

We added an onboard 3.3V regulator and logic-level shifting circuitry, making it a perfect choice for interfacing with any 3V or 5V microcontroller or computer, such as Arduino or Raspberry Pi. We've got both Arduino (C/C++) () and CircuitPython (Python 3) libraries () available so you can use it with any microcontroller like Arduino, ESP32, Metro, etc, or with Raspberry Pi or other Linux computers, thanks to Blinka (our CircuitPython library support helper).

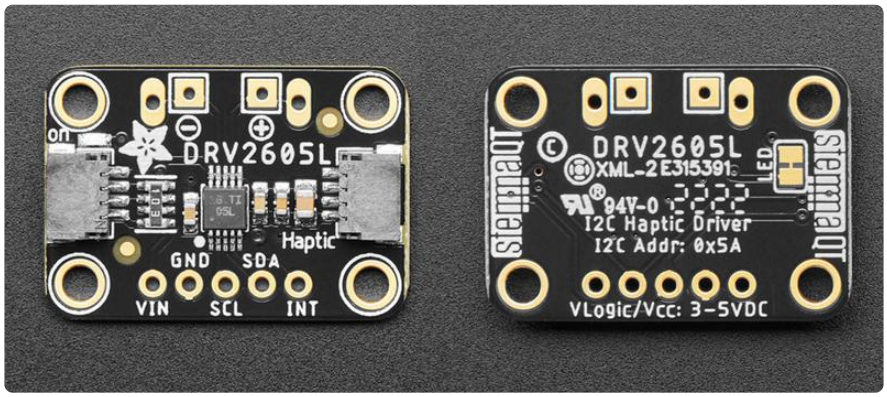


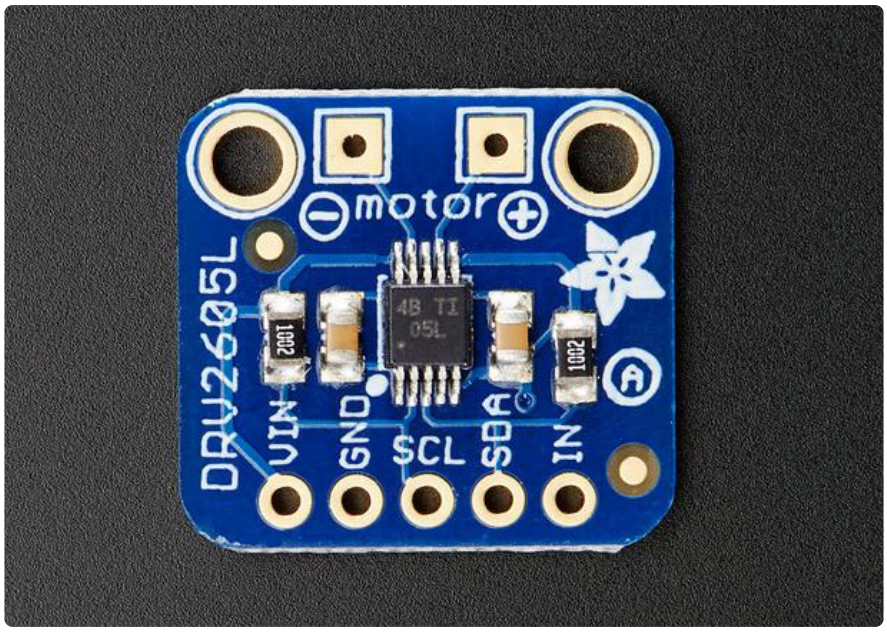
As if that weren't enough, we've also added SparkFun qwiic () compatible STEMMA QT () connectors for the I2C bus so you don't even need to solder. For a no-solder experience, just wire up to your favorite micro () using a STEMMA QT adapter cable. () The Stemma QT connectors also mean the breakout can be used with our various associated accessories. () QT Cable is not included, but we have a variety in the shop ()

There are two versions of this board - the STEMMA QT version shown above, and the original header-only version shown below. Code works the same on both!



Pinouts





The default I2C address is 0x5A.

Power Pins

The motor driver/controller on the breakout requires 3-5V power. You can use either, whichever logic level you use on your embedded processor

- VIN - 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
- GND - common ground for power and logic

I2C Logic Pins

- SCL - I2C clock pin, connect to your microcontroller's I2C clock line. This pin has a 10K pullup to VIN.
- SDA -I2C data pin, connect to your microcontroller's I2C data line. This pin has a 10K pullup to VIN.
- STEMMA QT () - These connectors allow you to connect to development boards with STEMMA QT connectors, or to other things, with various associated accessories ().

Motor Input

- Motor Positive (+) - positive input for a haptic motor
- Motor Negative (-) - negative input for a haptic motor

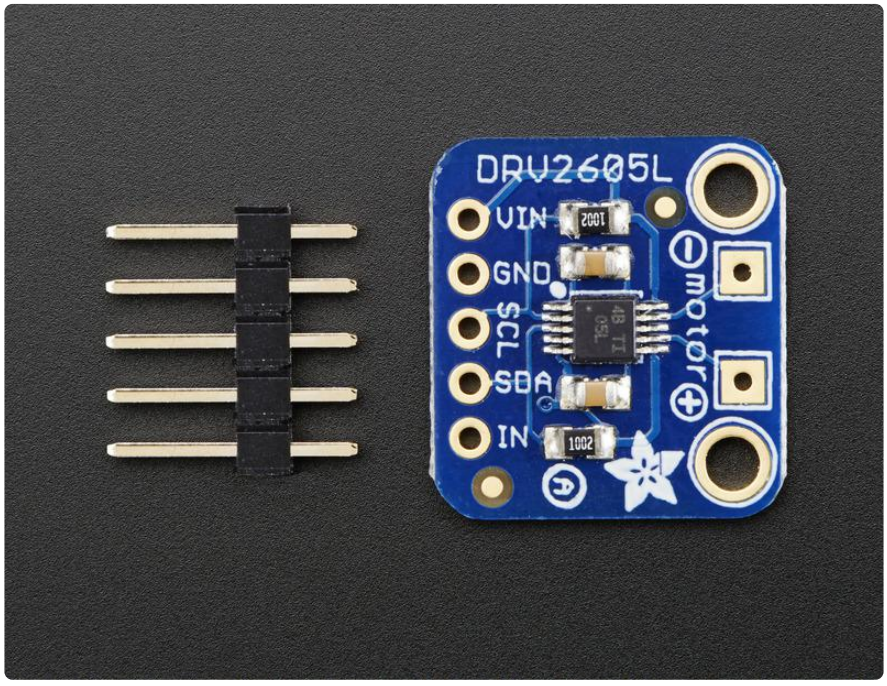
Other

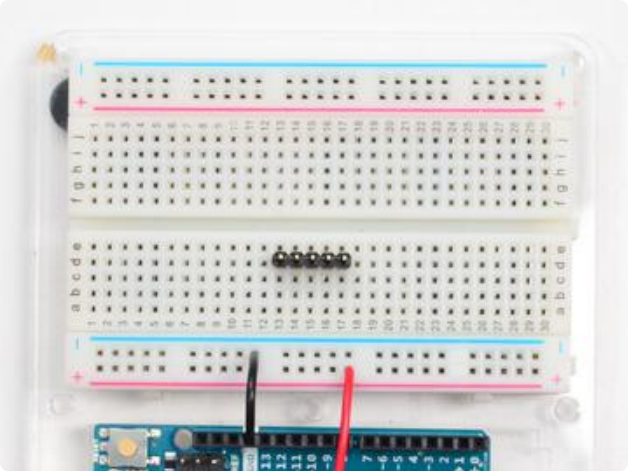
- INT - This is a general purpose pin that can be used for a couple different uses. One use is to read analog audio in to control the audio-to-haptic code. Another use is to 'trigger' the effects to go rather than sending a I2C command.

Power LED and 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 the green LED.
- LED jumper - In the upper right corner on the back of the board is a jumper for the power LED. If you wish to disable the power LED, simply cut the trace on this jumper.

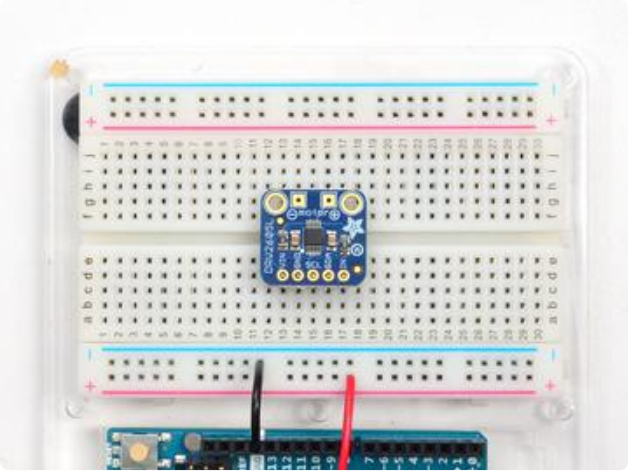
Assembly





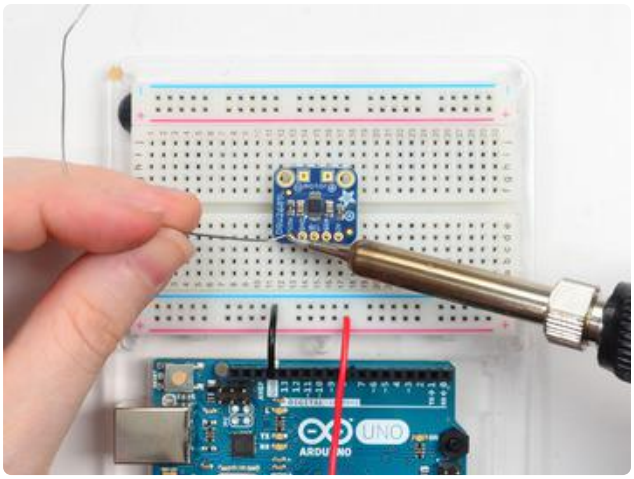
Prepare the header strip:

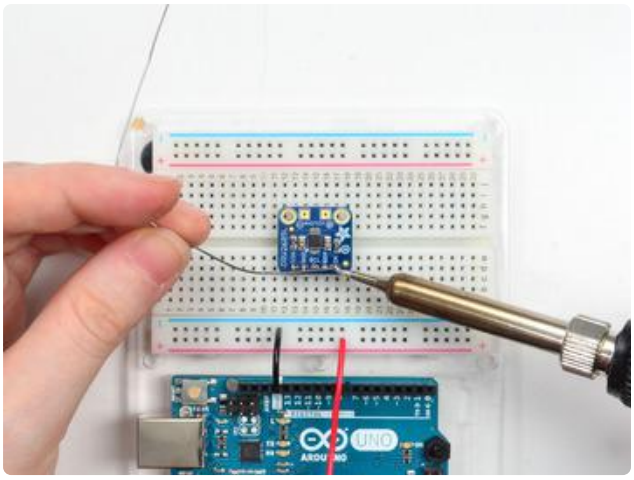
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down



Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads



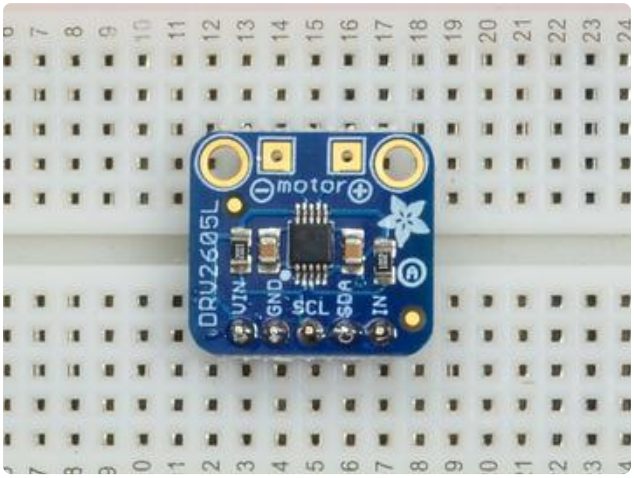


And Solder!

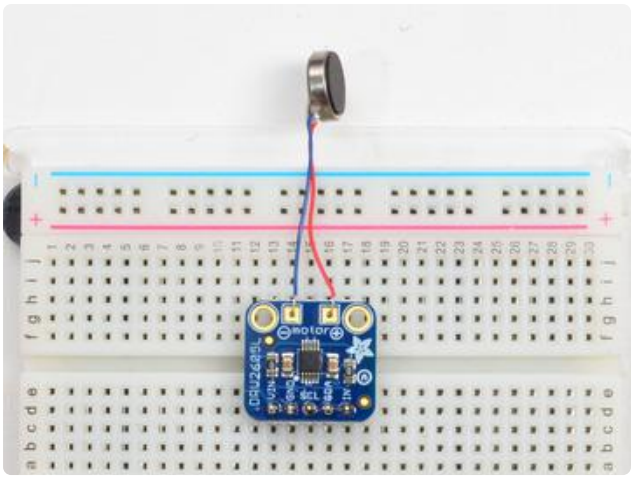
Be sure to solder all pins for reliable electrical contact.

Solder the longer power/data strip first

(For tips on soldering, be sure to check out our Guide to Excellent Soldering ()).

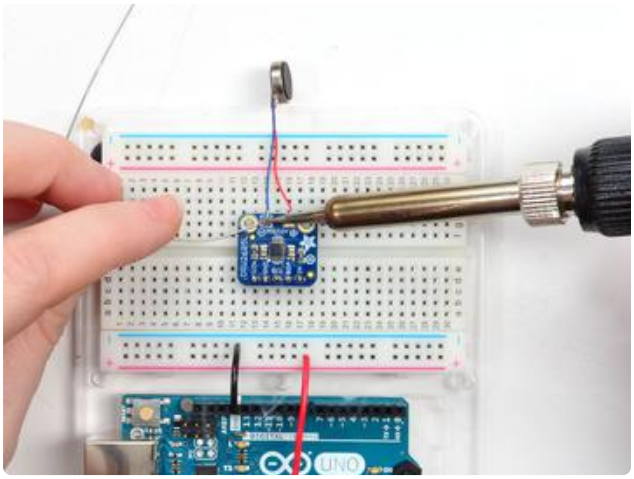


You're done! Check your solder joints visually and continue onto the next steps

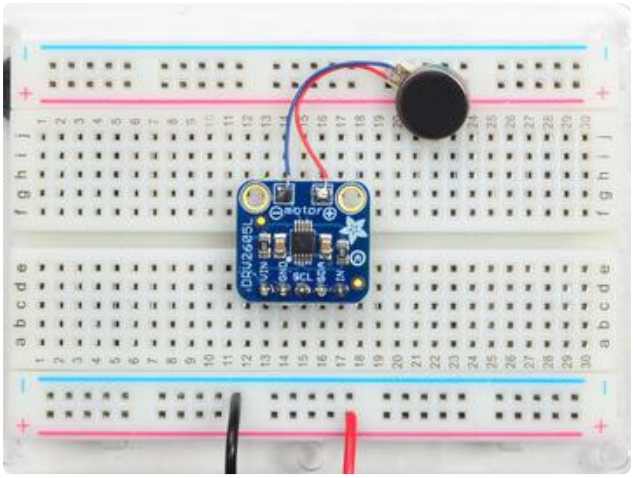


Attach Motor

We prefer to attach the little vibration motor directly to the Motor+ and Motor- pads



Solder in place

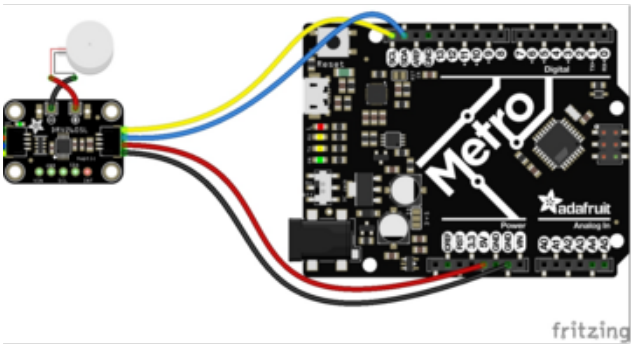


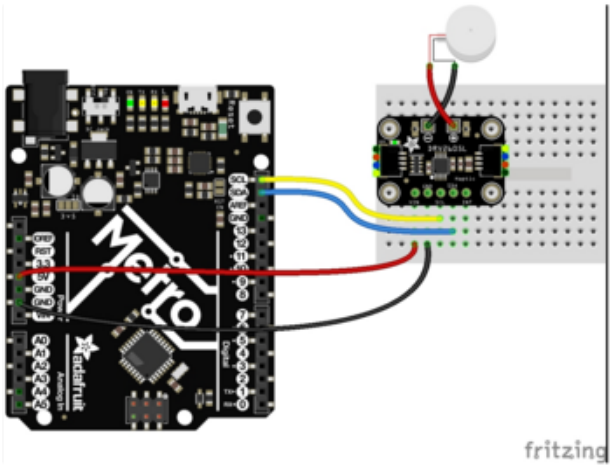
Check your work and continue!

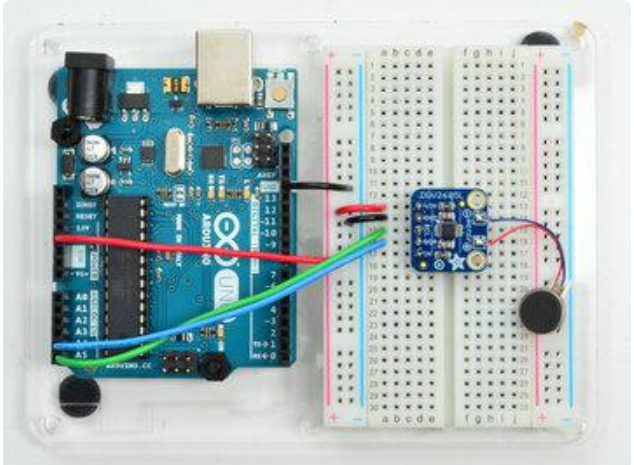
Arduino Code

Wiring for Arduino

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C capability, then port the code - it's pretty simple stuff!



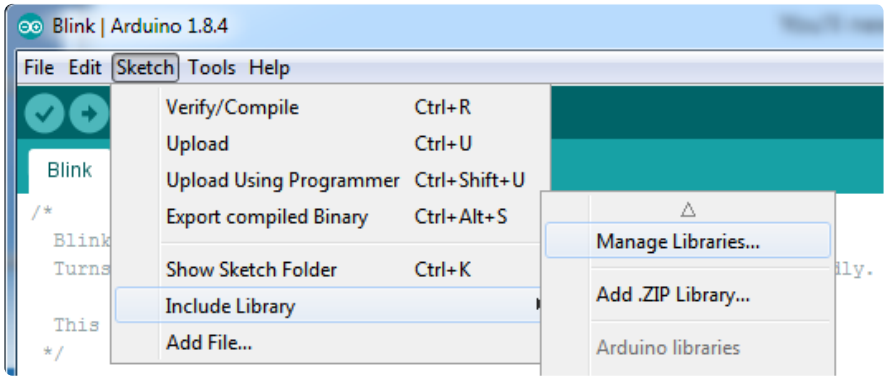




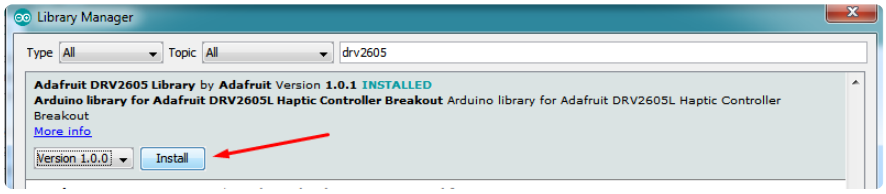
Connect Vin to the power supply, (red wire in STEMMA QT version). Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
Connect GND to common power/data ground (black wire in STEMMA QT version)
Connect the SCL pin to the I2C clock SCL pin on your Arduino (yellow wire in STEMMA QT version). On an UNO & '328 based Arduino, this is also known as A5, on a Mega it is also known as digital 21 and on a Leonardo/Micro, digital 3
Connect the SDA pin to the I2C data SDA pin on your Arduino (blue wire in STEMMA QT version). On an UNO & '328 based Arduino, this is also known as A4, on a Mega it is also known as digital 20 and on a Leonardo/Micro, digital 2
Controller Motor - to motor negative (black wire in STEMMA QT version).
Controller Motor + to motor positive (red wire in STEMMA QT version).

Install Adafruit_DRV2605 Library

To begin controling the motor chip, you will need to install the Adafruit_DRV2605 Library (). You can do that by going to the Arduino library manager under Sketch -> Include Library -> Manage Libraries...



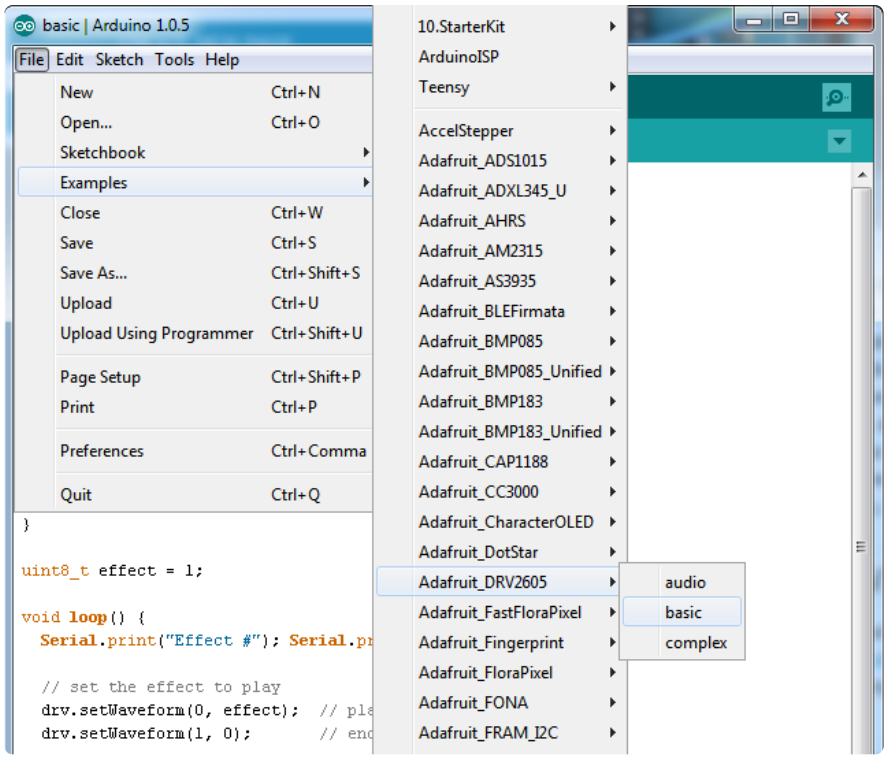
Then search for DRV2605 and find the Adafruit DRV2605 Library and click Install



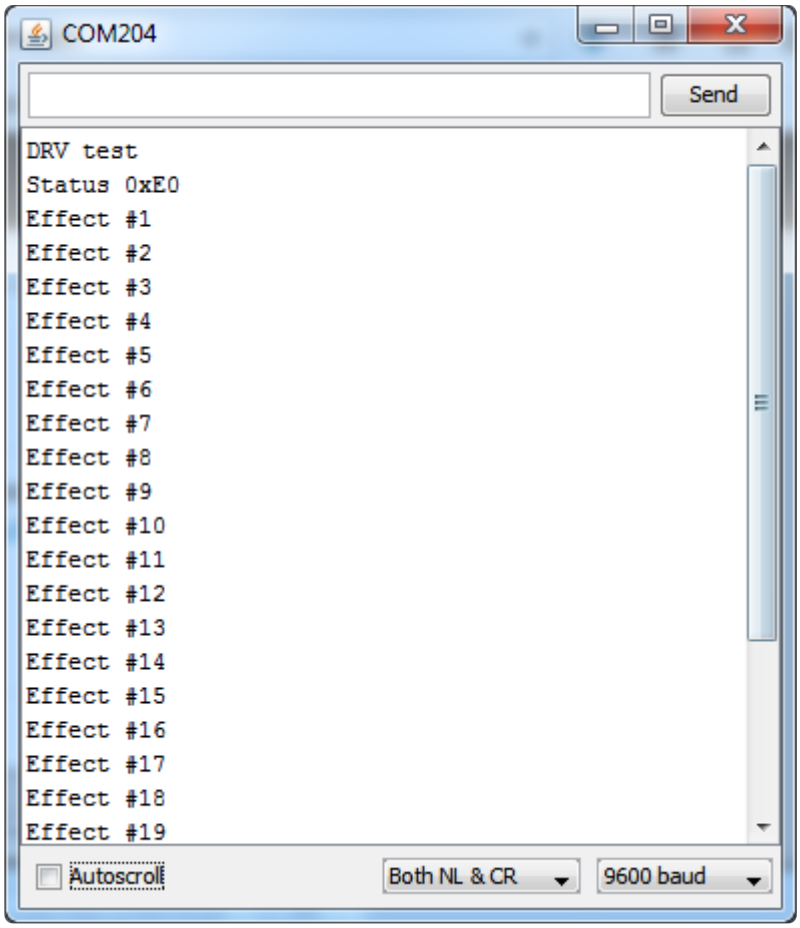
We also have a great tutorial on Arduino library installation at: http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use ()

Load Demo Sketch

Now you can open up File->Examples->Adafruit_DRV2605->basic and upload to your Arduino wired up to the breakout.



Open up the serial console and hold the vibration motor between your fingers.



The sketch will play all 123 built-in vibration effects in order. The full list with names is available in the DRV2605 datasheet ()

Here's a screenshot for quick reference

Effect ID#	Waveform Name	Effect ID#	Waveform Name	Effect ID#	Waveform Name
1	Strong click – 100%	42	Long double sharp click medium 2 – 80%	83	Transition ramp up long smooth 2 – 0 to 100%
2	Strong click – 60%	43	Long double sharp click medium 3 – 60%	84	Transition ramp up medium smooth 1 – 0 to 100%
3	Strong click – 30%	44	Long double sharp tick 1 – 100%	85	Transition ramp up medium smooth 2 – 0 to 100%
4	Sharp click – 100%	45	Long double sharp tick 2 – 80%	86	Transition ramp up short smooth 1 – 0 to 100%
5	Sharp click – 60%	46	Long double sharp tick 3 – 60%	87	Transition ramp up short smooth 2 – 0 to 100%
6	Sharp click – 30%	47	Buzz 1 – 100%	88	Transition ramp up long sharp 1 – 0 to 100%
7	Soft bump – 100%	48	Buzz 2 – 80%	89	Transition ramp up long sharp 2 – 0 to 100%
8	Soft bump – 60%	49	Buzz 3 – 60%	90	Transition ramp up medium sharp 1 – 0 to 100%
9	Soft bump – 30%	50	Buzz 4 – 40%	91	Transition ramp up medium sharp 2 – 0 to 100%
10	Double click – 100%	51	Buzz 5 – 20%	92	Transition ramp up short sharp 1 – 0 to 100%
11	Double click – 60%	52	Pulsing strong 1 – 100%	93	Transition ramp up short sharp 2 – 0 to 100%
12	Triple click – 100%	53	Pulsing strong 2 – 60%	94	Transition ramp down long smooth 1 – 50 to 0%
13	Soft fuzz – 60%	54	Pulsing medium 1 – 100%	95	Transition ramp down long smooth 2 – 50 to 0%
14	Strong buzz – 100%	55	Pulsing medium 2 – 60%	96	Transition ramp down medium smooth 1 – 50 to 0%
15	750-ms alert 100%	56	Pulsing sharp 1 – 100%	97	Transition ramp down medium smooth 2 – 50 to 0%
16	1000-ms alert 100%	57	Pulsing sharp 2 – 60%	98	Transition ramp down short smooth 1 – 50 to 0%
17	Strong click 1 – 100%	58	Transition click 1 – 100%	99	Transition ramp down short smooth 2 – 50 to 0%
18	Strong click 2 – 80%	59	Transition click 2 – 80%	100	Transition ramp down long sharp 1 – 50 to 0%
19	Strong click 3 – 60%	60	Transition click 3 – 60%	101	Transition ramp down long sharp 2 – 50 to 0%
20	Strong click 4 – 30%	61	Transition click 4 – 40%	102	Transition ramp down medium sharp 1 – 50 to 0%
21	Medium click 1 – 100%	62	Transition click 5 – 20%	103	Transition ramp down medium sharp 2 – 50 to 0%
22	Medium click 2 – 80%	63	Transition click 6 – 10%	104	Transition ramp down short sharp 1 – 50 to 0%
23	Medium click 3 – 60%	64	Transition hum 1 – 100%	105	Transition ramp down short sharp 2 – 50 to 0%
24	Sharp tick 1 – 100%	65	Transition hum 2 – 80%	106	Transition ramp up long smooth 1 – 0 to 50%
25	Sharp tick 2 – 80%	66	Transition hum 3 – 60%	107	Transition ramp up long smooth 2 – 0 to 50%
26	Sharp tick 3 – 60%	67	Transition hum 4 – 40%	108	Transition ramp up medium smooth 1 – 0 to 50%
27	Short double click strong 1 – 100%	68	Transition hum 5 – 20%	109	Transition ramp up medium smooth 2 – 0 to 50%
28	Short double click strong 2 – 80%	69	Transition hum 6 – 10%	110	Transition ramp up short smooth 1 – 0 to 50%
29	Short double click strong 3 – 60%	70	Transition ramp down long smooth 1 – 100 to 0%	111	Transition ramp up short smooth 2 – 0 to 50%
30	Short double click strong 4 – 30%	71	Transition ramp down long smooth 2 – 100 to 0%	112	Transition ramp up long sharp 1 – 0 to 50%
31	Short double click medium 1 – 100%	72	Transition ramp down medium smooth 1 – 100 to 0%	113	Transition ramp up long sharp 2 – 0 to 50%
32	Short double click medium 2 – 80%	73	Transition ramp down medium smooth 2 – 100 to 0%	114	Transition ramp up medium sharp 1 – 0 to 50%
33	Short double click medium 3 – 60%	74	Transition ramp down short smooth 1 – 100 to 0%	115	Transition ramp up medium sharp 2 – 0 to 50%
34	Short double sharp tick 1 – 100%	75	Transition ramp down short smooth 2 – 100 to 0%	116	Transition ramp up short sharp 1 – 0 to 50%
35	Short double sharp tick 2 – 80%	76	Transition ramp down long sharp 1 – 100 to 0%	117	Transition ramp up short sharp 2 – 0 to 50%
36	Short double sharp tick 3 – 60%	77	Transition ramp down long sharp 2 – 100 to 0%	118	Long buzz for programmatic stopping – 100%
37	Long double sharp click strong 1 – 100%	78	Transition ramp down medium sharp 1 – 100 to 0%	119	Smooth hum 1 (No kick or brake pulse) – 50%
38	Long double sharp click strong 2 – 80%	79	Transition ramp down medium sharp 2 – 100 to 0%	120	Smooth hum 2 (No kick or brake pulse) – 40%
39	Long double sharp click strong 3 – 60%	80	Transition ramp down short sharp 1 – 100 to 0%	121	Smooth hum 3 (No kick or brake pulse) – 30%
40	Long double sharp click strong 4 – 30%	81	Transition ramp down short sharp 2 – 100 to 0%	122	Smooth hum 4 (No kick or brake pulse) – 20%
41	Long double sharp click medium 1 – 100%	82	Transition ramp up long smooth 1 – 0 to 100%	123	Smooth hum 5 (No kick or brake pulse) – 10%

Multiple Waveforms

You can also string together multiple effects in a row, up to 7. Check out the complex example sketch, and `setWaveform` for each slot. The last slot should be set to 0 to indicate it's the end.

When you are ready to place the full waveform sequence, send the `go()` command!

Audio

You can also turn the DRV2605 into an audio-to-vibration driver. Use a 1uF capacitor in series to line level voltage audio into the IN pin, then load up the audio example sketch. If you don't feel anything, try boosting up the source audio volume, it has to be pretty loud!

Arduino Docs

Arduino Docs ()

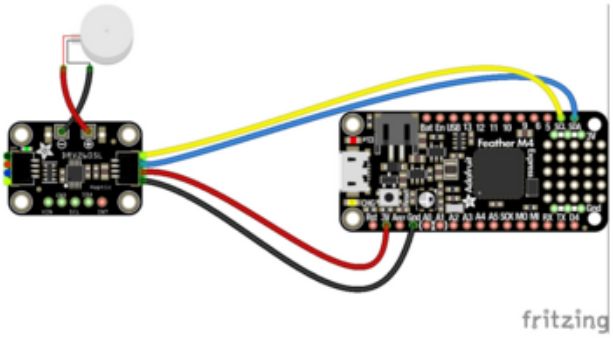
Python & CircuitPython

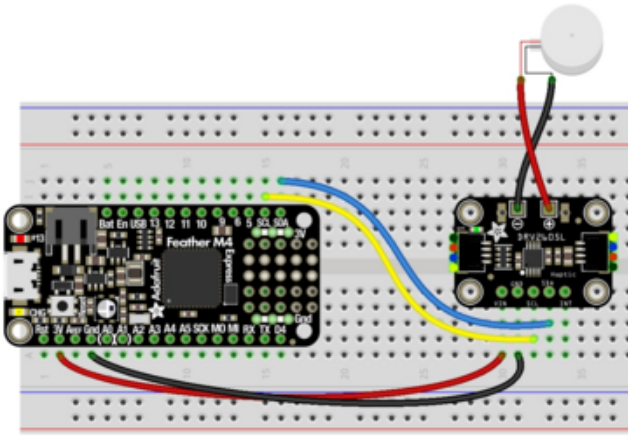
It's easy to use the DRV2605 controller with Python or CircuitPython, and the Adafruit CircuitPython DRV2605 () module. This module allows you to easily write Python code that controls the vibration of the motor.

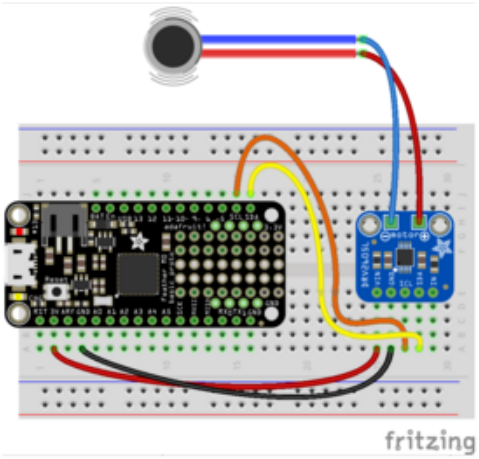
You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

CircuitPython Microcontroller Wiring

First wire up a DRV2605 to your board exactly as shown on the previous pages for Arduino using an I2C connection. Here's an example of wiring a Feather M0 to the controller with I2C:





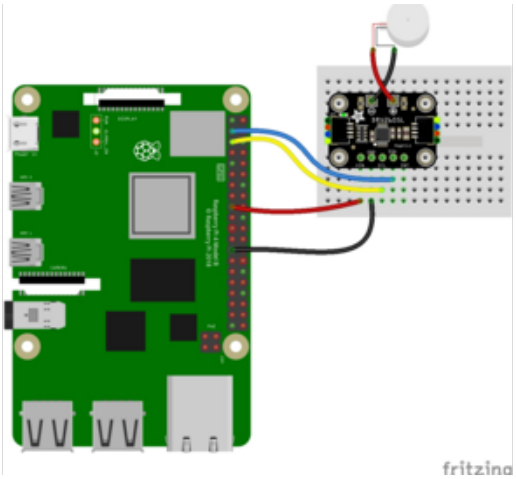


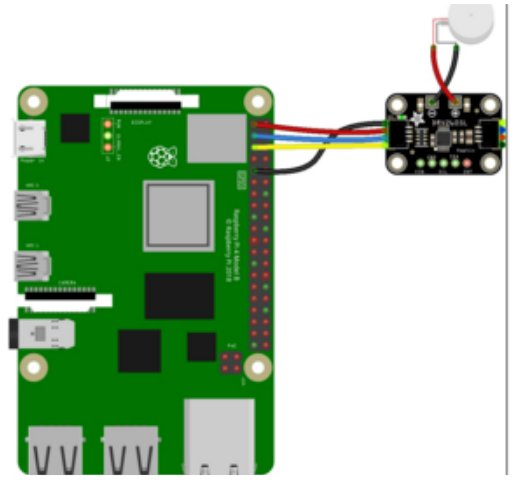
Board 3V to controller VIN (red wire in STEMMA QT version)
Board GND to controller GND (black wire in STEMMA QT version)
Board SCL to controller SCL (yellow wire in STEMMA QT version)
Board SDA to controller SDA (blue wire in STEMMA QT version)
Controller Motor - to motor negative (black wire in STEMMA QT version)
Controller Motor + to motor positive (red wire in STEMMA QT version)

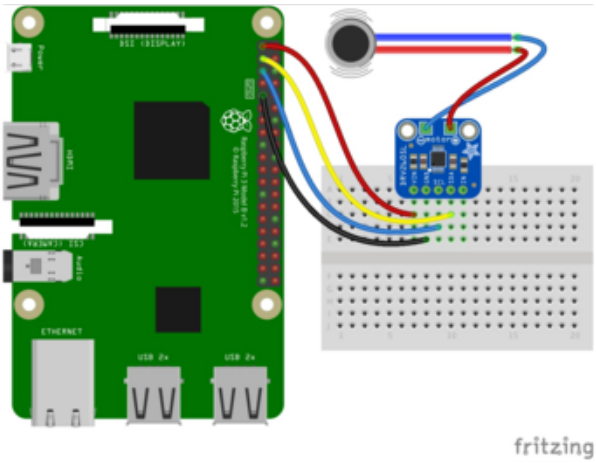
Python Computer Wiring

Since there's 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 ().

Here's the Raspberry Pi wired with I2C:







Pi 3V3 to sensor VIN (red wire in STEMMA QT version)
Pi GND to sensor GND (black wire in STEMMA QT version)
Pi SCL to sensor SCL (yellow wire in STEMMA QT version)
Pi SDA to sensor SDA (blue wire in STEMMA QT version)
Controller Motor - to motor negative (black wire in STEMMA QT version)
Controller Motor + to motor positive (red wire in STEMMA QT version)

CircuitPython Installation of DRV2605 Library

You'll need to install the Adafruit CircuitPython DRV2605 () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). Our introduction guide has a great page on how to install the library bundle () for both express and non-express boards.

Remember for non-express boards like the, you'll need to manually install the necessary libraries from the bundle:

- adafruit_drv2605.mpy
- adafruit_bus_device

Before continuing make sure your board's lib folder or root filesystem has the adafruit_drv2605.mpy, and adafruit_bus_device files and folders copied over.

Next connect to the board's serial REPL () so you are at the CircuitPython >>> prompt.

Python Installation of DRV2605 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 ()!

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

- `sudo pip3 install adafruit-circuitpython-drv2605`

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 & Python Usage

To demonstrate the usage of the board we'll initialize it and vibrate the motor using effects built-in to the DRV2605 chip. First run the following code to import the necessary modules and initialize the I2C connection with the controller:

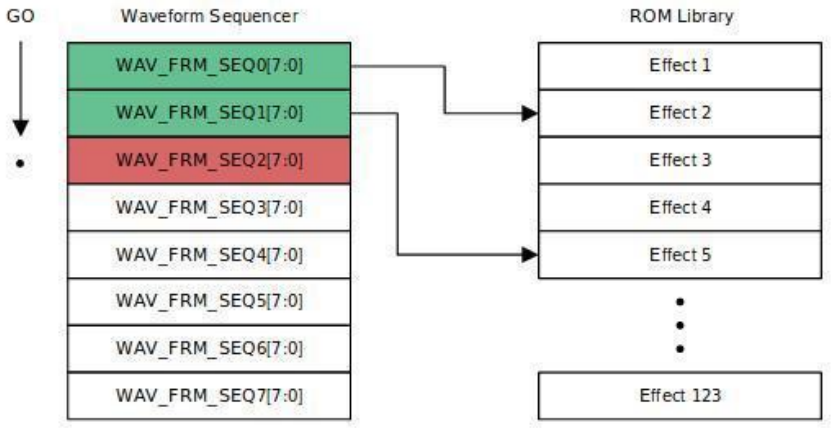
```
import board
import busio
import adafruit_drv2605
```

```
i2c = busio.I2C(board.SCL, board.SDA)
drv = adafruit_drv2605.DRV2605(i2c)
```

Now you're ready to start vibrating the motor with different built-in effects. First you need to choose an effect to play based on its ID number. See table 11.2 in the datasheet () for a list of all the effects:

Effect ID#	Waveform Name	Effect ID#	Waveform Name	Effect ID#	Waveform Name
1	Strong click – 100%	42	Long double sharp click medium 2 – 80%	83	Transition ramp up long smooth 2 – 0 to 100%
2	Strong click – 60%	43	Long double sharp click medium 3 – 60%	84	Transition ramp up medium smooth 1 – 0 to 100%
3	Strong click – 30%	44	Long double sharp tick 1 – 100%	85	Transition ramp up medium smooth 2 – 0 to 100%
4	Sharp click – 100%	45	Long double sharp tick 2 – 80%	86	Transition ramp up short smooth 1 – 0 to 100%
5	Sharp click – 60%	46	Long double sharp tick 3 – 60%	87	Transition ramp up short smooth 2 – 0 to 100%
6	Sharp click – 30%	47	Buzz 1 – 100%	88	Transition ramp up long sharp 1 – 0 to 100%
7	Soft bump – 100%	48	Buzz 2 – 80%	89	Transition ramp up long sharp 2 – 0 to 100%
8	Soft bump – 60%	49	Buzz 3 – 60%	90	Transition ramp up medium sharp 1 – 0 to 100%
9	Soft bump – 30%	50	Buzz 4 – 40%	91	Transition ramp up medium sharp 2 – 0 to 100%
10	Double click – 100%	51	Buzz 5 – 20%	92	Transition ramp up short sharp 1 – 0 to 100%
11	Double click – 60%	52	Pulsing strong 1 – 100%	93	Transition ramp up short sharp 2 – 0 to 100%
12	Triple click – 100%	53	Pulsing strong 2 – 60%	94	Transition ramp down long smooth 1 – 50 to 0%
13	Soft fuzz – 60%	54	Pulsing medium 1 – 100%	95	Transition ramp down long smooth 2 – 50 to 0%
14	Strong buzz – 100%	55	Pulsing medium 2 – 60%	96	Transition ramp down medium smooth 1 – 50 to 0%
15	750-ms alert 100%	56	Pulsing sharp 1 – 100%	97	Transition ramp down medium smooth 2 – 50 to 0%
16	1000-ms alert 100%	57	Pulsing sharp 2 – 60%	98	Transition ramp down short smooth 1 – 50 to 0%
17	Strong click 1 – 100%	58	Transition click 1 – 100%	99	Transition ramp down short smooth 2 – 50 to 0%
18	Strong click 2 – 80%	59	Transition click 2 – 80%	100	Transition ramp down long sharp 1 – 50 to 0%
19	Strong click 3 – 60%	60	Transition click 3 – 60%	101	Transition ramp down long sharp 2 – 50 to 0%
20	Strong click 4 – 30%	61	Transition click 4 – 40%	102	Transition ramp down medium sharp 1 – 50 to 0%
21	Medium click 1 – 100%	62	Transition click 5 – 20%	103	Transition ramp down medium sharp 2 – 50 to 0%
22	Medium click 2 – 80%	63	Transition click 6 – 10%	104	Transition ramp down short sharp 1 – 50 to 0%
23	Medium click 3 – 60%	64	Transition hum 1 – 100%	105	Transition ramp down short sharp 2 – 50 to 0%
24	Sharp tick 1 – 100%	65	Transition hum 2 – 80%	106	Transition ramp up long smooth 1 – 0 to 50%
25	Sharp tick 2 – 80%	66	Transition hum 3 – 60%	107	Transition ramp up long smooth 2 – 0 to 50%
26	Sharp tick 3 – 60%	67	Transition hum 4 – 40%	108	Transition ramp up medium smooth 1 – 0 to 50%
27	Short double click strong 1 – 100%	68	Transition hum 5 – 20%	109	Transition ramp up medium smooth 2 – 0 to 50%
28	Short double click strong 2 – 80%	69	Transition hum 6 – 10%	110	Transition ramp up short smooth 1 – 0 to 50%
29	Short double click strong 3 – 60%	70	Transition ramp down long smooth 1 – 100 to 0%	111	Transition ramp up short smooth 2 – 0 to 50%
30	Short double click strong 4 – 30%	71	Transition ramp down long smooth 2 – 100 to 0%	112	Transition ramp up long sharp 1 – 0 to 50%
31	Short double click medium 1 – 100%	72	Transition ramp down medium smooth 1 – 100 to 0%	113	Transition ramp up long sharp 2 – 0 to 50%
32	Short double click medium 2 – 80%	73	Transition ramp down medium smooth 2 – 100 to 0%	114	Transition ramp up medium sharp 1 – 0 to 50%
33	Short double click medium 3 – 60%	74	Transition ramp down short smooth 1 – 100 to 0%	115	Transition ramp up medium sharp 2 – 0 to 50%
34	Short double sharp tick 1 – 100%	75	Transition ramp down short smooth 2 – 100 to 0%	116	Transition ramp up short sharp 1 – 0 to 50%
35	Short double sharp tick 2 – 80%	76	Transition ramp down long sharp 1 – 100 to 0%	117	Transition ramp up short sharp 2 – 0 to 50%
36	Short double sharp tick 3 – 60%	77	Transition ramp down long sharp 2 – 100 to 0%	118	Long buzz for programmatic stopping – 100%
37	Long double sharp click strong 1 – 100%	78	Transition ramp down medium sharp 1 – 100 to 0%	119	Smooth hum 1 (No kick or brake pulse) – 50%
38	Long double sharp click strong 2 – 80%	79	Transition ramp down medium sharp 2 – 100 to 0%	120	Smooth hum 2 (No kick or brake pulse) – 40%
39	Long double sharp click strong 3 – 60%	80	Transition ramp down short sharp 1 – 100 to 0%	121	Smooth hum 3 (No kick or brake pulse) – 30%
40	Long double sharp click strong 4 – 30%	81	Transition ramp down short sharp 2 – 100 to 0%	122	Smooth hum 4 (No kick or brake pulse) – 20%
41	Long double sharp click medium 1 – 100%	82	Transition ramp up long smooth 1 – 0 to 100%	123	Smooth hum 5 (No kick or brake pulse) – 10%

Each effect needs to be loaded into one of eight slots in the Waveform Sequencer:



You access the Waveform Sequencer by using the sequence property and providing an index for the slot location. To specify the effect, use the special Effect class provided in the library, giving it the effect number.

For example, here is how to load Effect ID No. 1 "Strong Click - 100%" into slot 0 of the sequence:

```
drv.sequence[0] = adafruit_drv2605.Effect(1)
```

To play the sequence of effects call the play function:

```
drv.play()
```

You should feel the motor vibrate in a sharp click! Every time you call the `play` function the motor will play back the sequence. Right now, that's just the click effected loaded into slot 0.

Try selecting a different effect and playing it, like the "Buzz 1 - 100%", a strong buzz, which is Effect ID 47:

```
drv.sequence[0] = adafruit_drv2605.Effect(47)
drv.play()
```

Try other effects to see which ones are interesting and useful in your project!

You can select multiple effects to play back at sequentially and create interesting compound effects. When you call `play`, the sequence is played until it either reaches a 0 or plays through all the slots. There's also a special `Pause` class you can use to generate a pause. You specify the pause time in seconds and it has to occupy a slot.

Let's play the two effects from above, separated by a half second pause:

```
drv.sequence[0] = adafruit_drv2605.Effect(1)
drv.sequence[1] = adafruit_drv2605.Pause(0.5)
drv.sequence[2] = adafruit_drv2605.Effect(47)
drv.sequence[3] = adafruit_drv2605.Effect(0)
drv.play()
```

The call to `play` is non-blocking, that is, it returns immediately and the rest of the code after that keeps running. The DRV2605 is doing the work of playing back the loaded sequence. If you ever want to stop the current playback, use the `stop` function:

```
drv.stop()
```

Changing Motor Types

Finally it's uncommon but you might want to switch between using a linear resonance actuator motor or eccentric rotating mass motor. The small flat pancake motors like sold in the Adafruit shop are ERM motors and the library defaults to their usage. However you can call the `use_LRM` function to switch to configure the chip to use a LRM style motor:

```
drv.use_LRM()
```

You can also switch back to using an ERM style motor (the default) with the `use_ERM` function:

```
drv.use_ERM()
```

That's all there is to using the DRV2605 with CircuitPython! Happy motoring!

Full Example Code

Here's a complete example of using the board to play the 123 effects for a half second each. Save this as main.py on your board and watch the REPL output to see the effect ID printed as it plays on the motor.

```
# SPDX-FileCopyrightText: 2017 Tony DiCola for Adafruit Industries
# SPDX-License-Identifier: MIT

# Simple demo of the DRV2605 haptic feedback motor driver.
# Will play all 123 effects in order for about a half second each.
import time
```

```
import board
import busio

import adafruit_drv2605


# Initialize I2C bus and DRV2605 module.
i2c = busio.I2C(board.SCL, board.SDA)
drv = adafruit_drv2605.DRV2605(i2c)

# Main loop runs forever trying each effect (1-123).
# See table 11.2 in the datasheet for a list of all the effect names and IDs.
#   http://www.ti.com/lit/ds/symlink/drv2605.pdf
effect_id = 1
while True:
    print("Playing effect #{0}".format(effect_id))
    drv.sequence[0] = adafruit_drv2605.Effect(effect_id)  # Set the effect on slot
0.
    # You can assign effects to up to 8 different slots to combine
    # them in interesting ways. Index the sequence property with a
    # slot number 0 to 7.
    # Optionally, you can assign a pause to a slot. E.g.
    # drv.sequence[1] = adafruit_drv2605.Pause(0.5)  # Pause for half a second
    drv.play()  # play the effect
    time.sleep(0.5)  # for 0.5 seconds
    drv.stop()  # and then stop (if it's still running)
    # Increment effect ID and wrap back around to 1.
    effect_id += 1
    if effect_id > 123:
        effect_id = 1
```

Python Docs

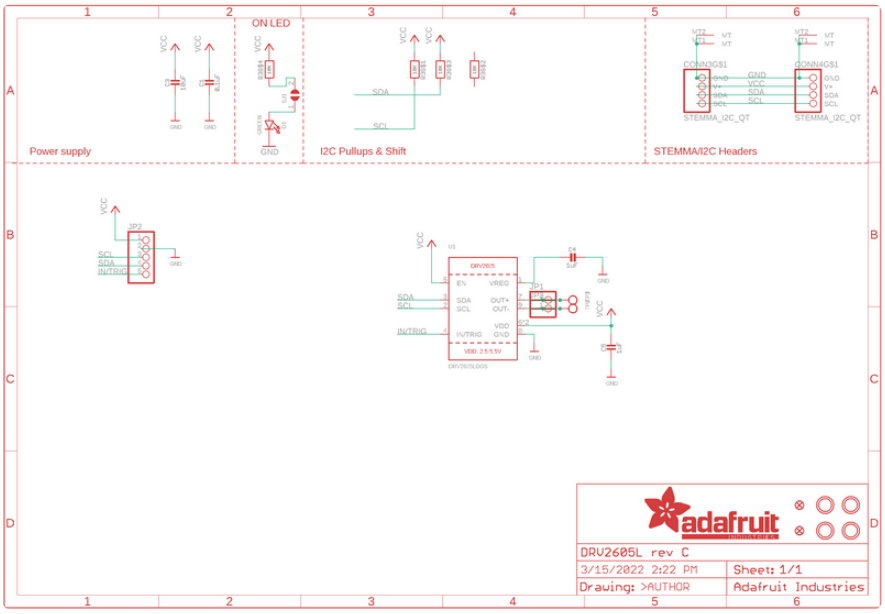
Python Docs ()

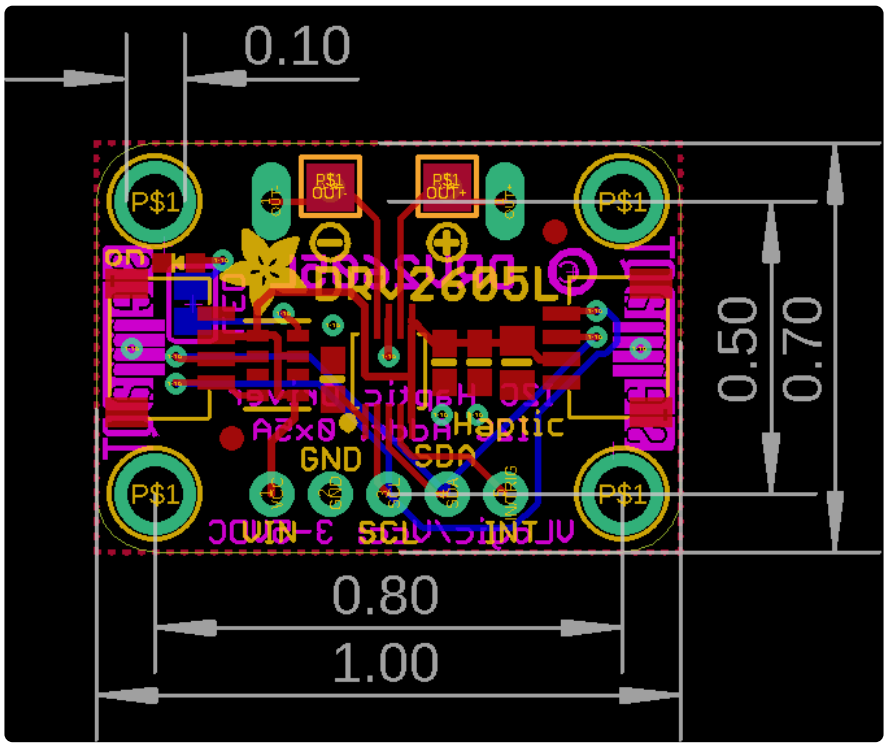
Downloads

Files

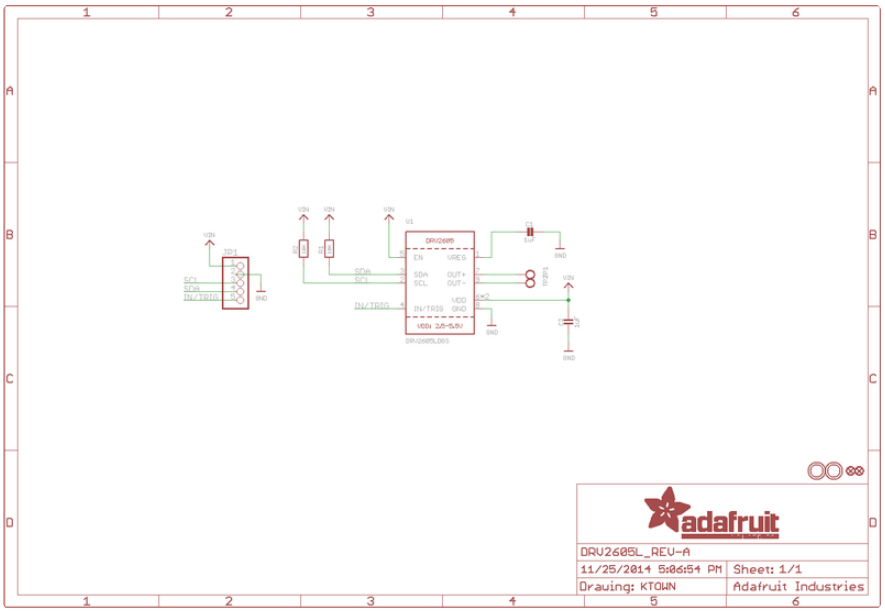
- EagleCAD PCB files on GitHub ()
- Fritzing object (original version) in the Adafruit Fritzing Library ()
- Fritzing object (STEMMA QT version) in the Adafruit Fritzing Library ()
- DRV2605L Datasheet ()
- 3D Models on GitHub ()

Schematic and Fab Print STEMMA QT Version





Schematic and Fab Print Original Version



Fabrication print

Dimensions in Inches