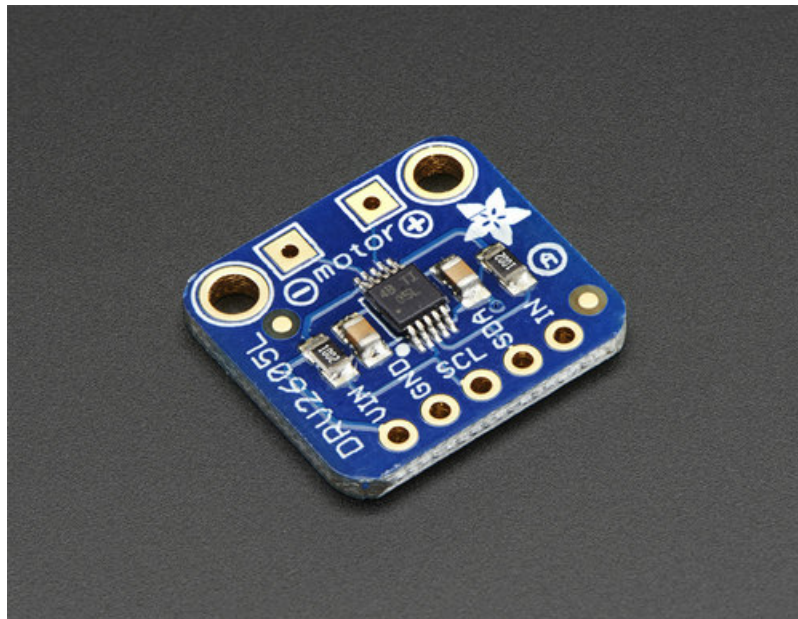


Adafruit DRV2605 Haptic Controller Breakout

Created by lady ada

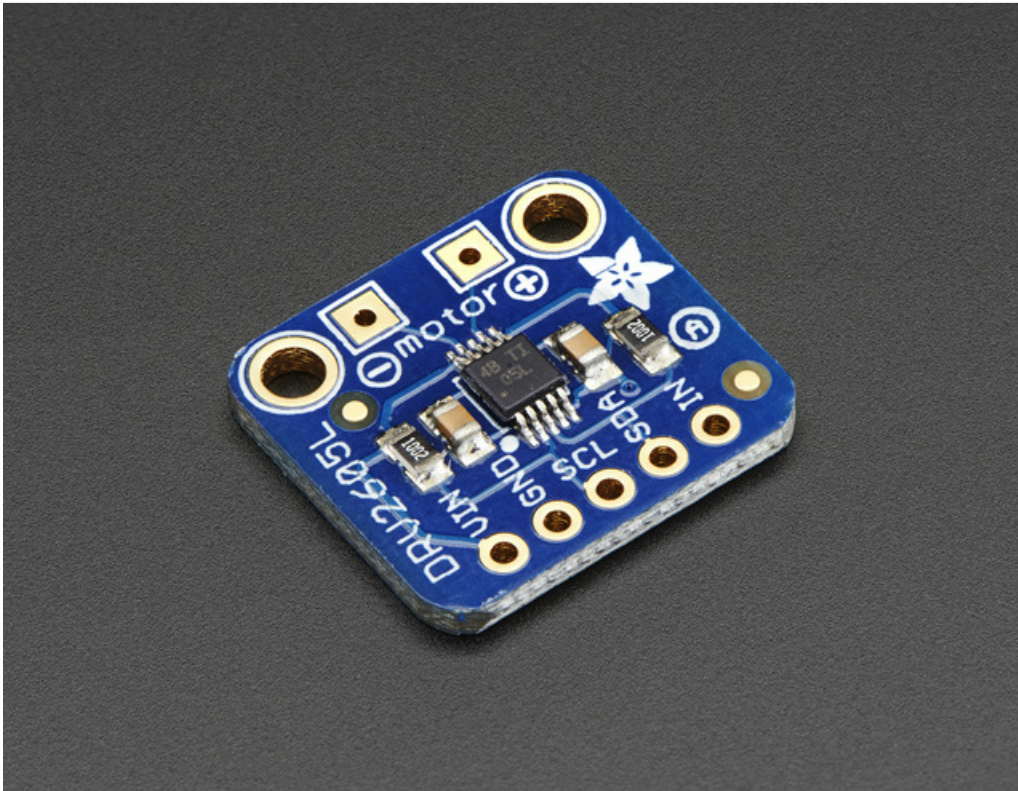


Last updated on 2018-02-27 02:38:17 AM UTC

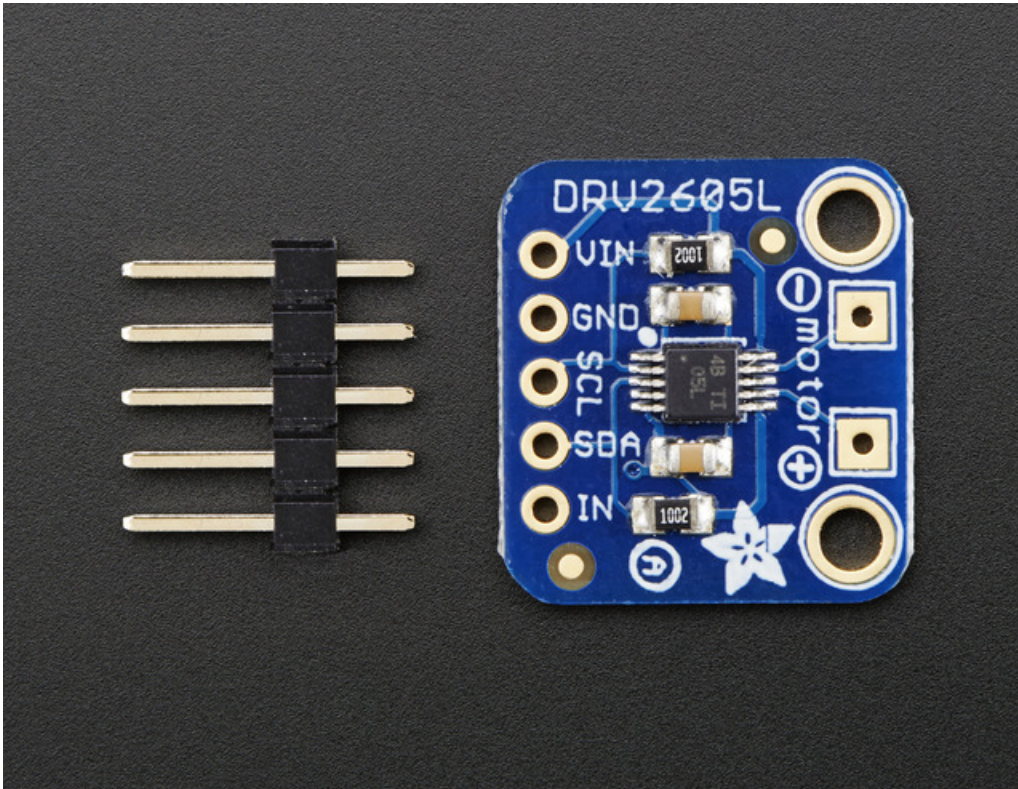
Guide Contents

Guide Contents	2
Overview	3
Pinouts	6
Power Pins	6
I2C Pins	6
Other!	6
Assembly	8
Prepare the header strip:	8
Add the breakout board:	9
And Solder!	10
Attach Motor	11
Arduino Code	13
Wiring for Arduino	13
Install Adafruit_DRV2605 Library	13
Load Demo Sketch	14
Multiple Waveforms	16
Audio	16
CircuitPython Code	17
Library Installation	17
Usage	18
Downloads	22
Datasheets	22
Schematic	22
Fabrication print	22

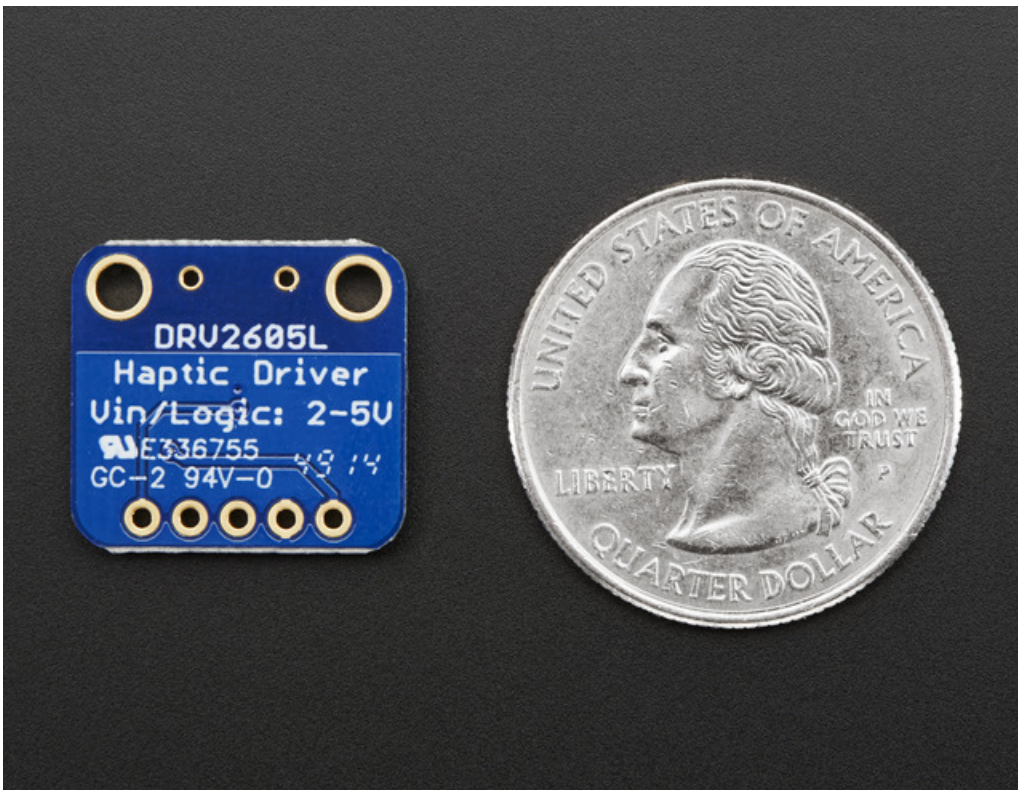
Overview



The DRV2605 from TI is a fancy little motor driver. Rather than controlling a stepper motor or DC motor, its 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.



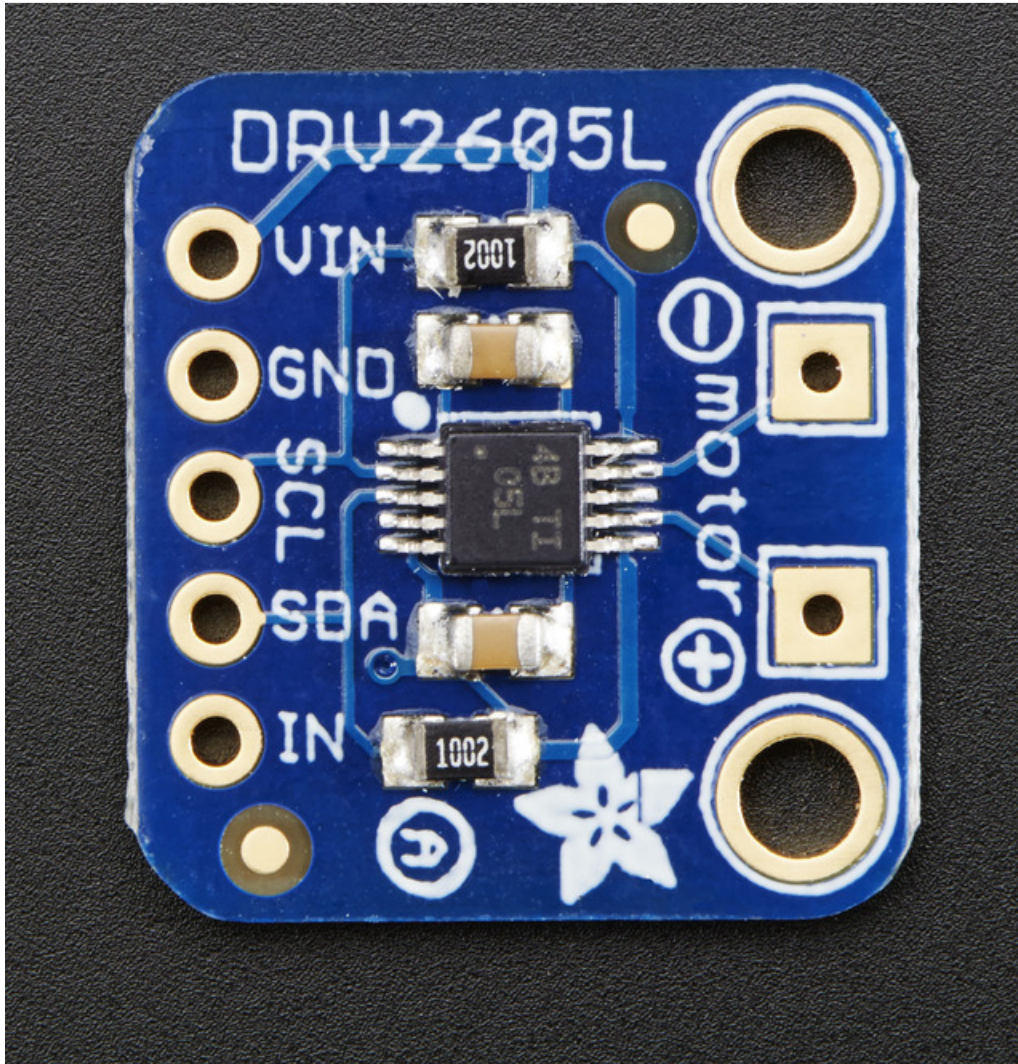
This chip is controlled over I2C - after initialization, a 'string' of multiple effects can be strung together in the chips memory and then triggered to actuate in a row. The built in effects are much much nicer than just 'on' and 'off' and will make your haptic project way nicer feeling.



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 put this nice chip onto a breakout board. it works with both 3V and 5V power/logic, we have code specifically for Arduino but porting it to any I2C-capable processor should be quite simple. Check it out and get buzzing!

Pinouts



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 Pins

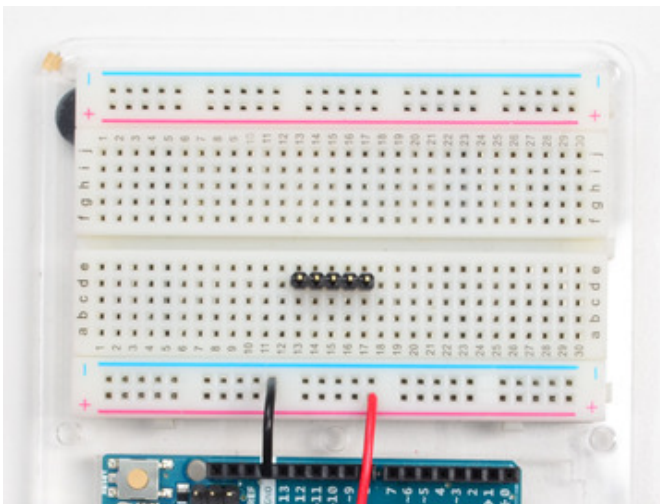
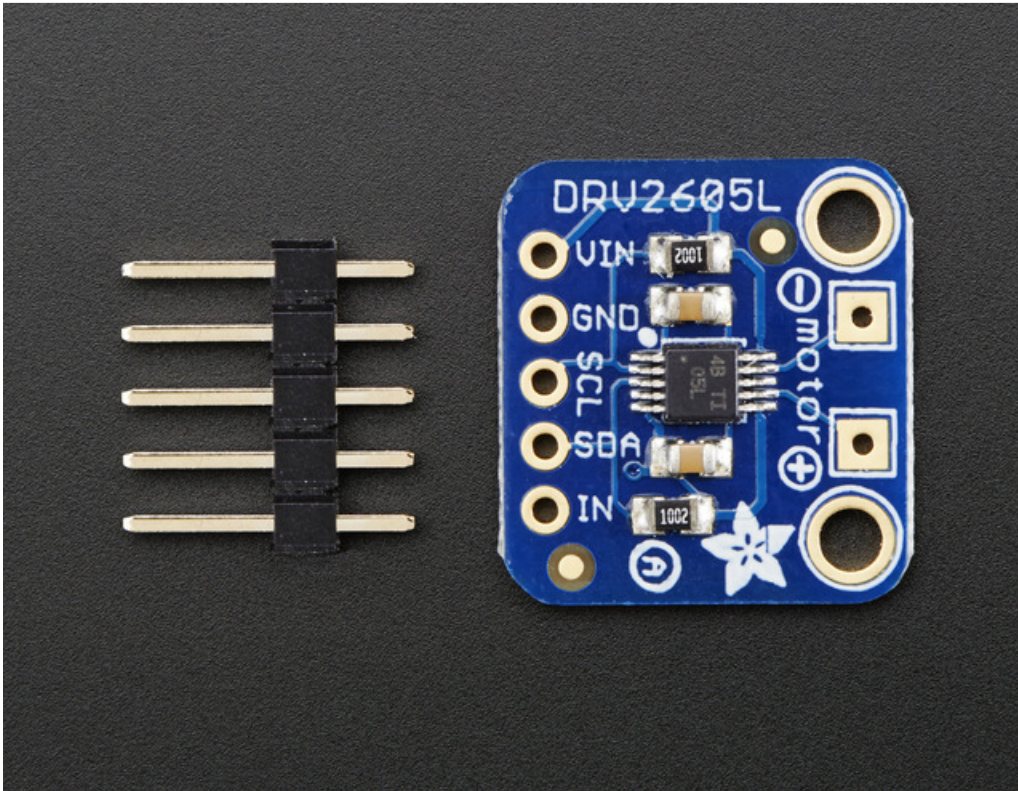
- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.

Other!

- **IN/TRIG** - 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

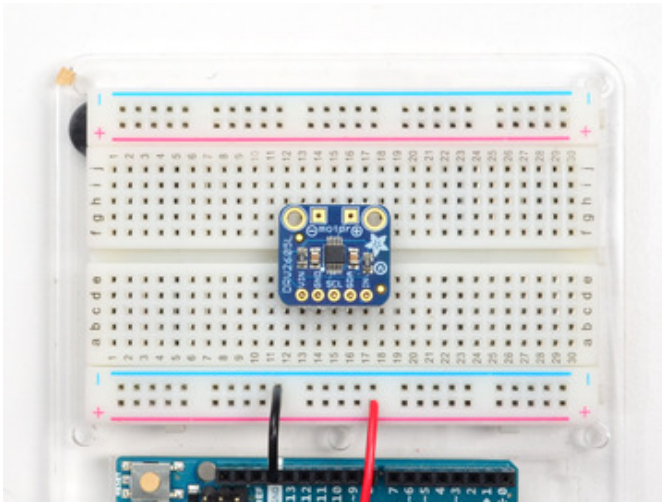
command.

Assembly

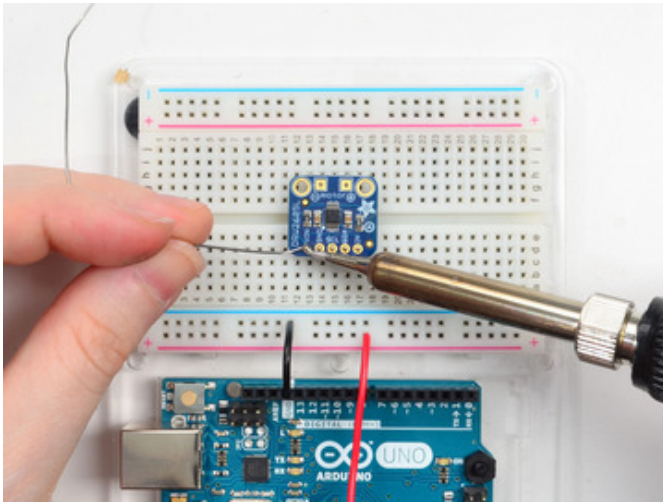


Prepare the header strip:

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

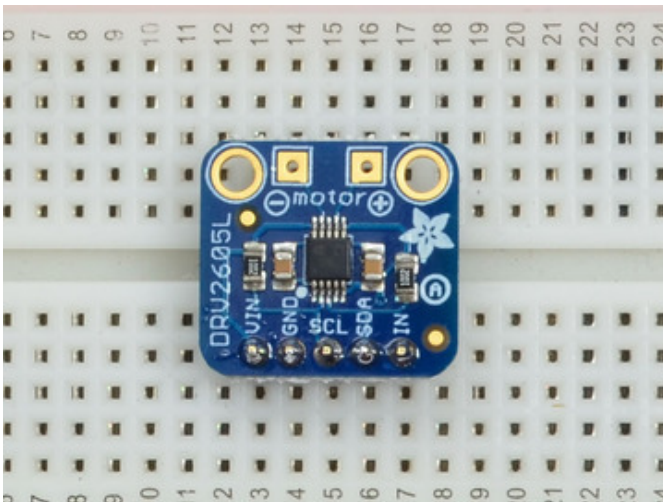
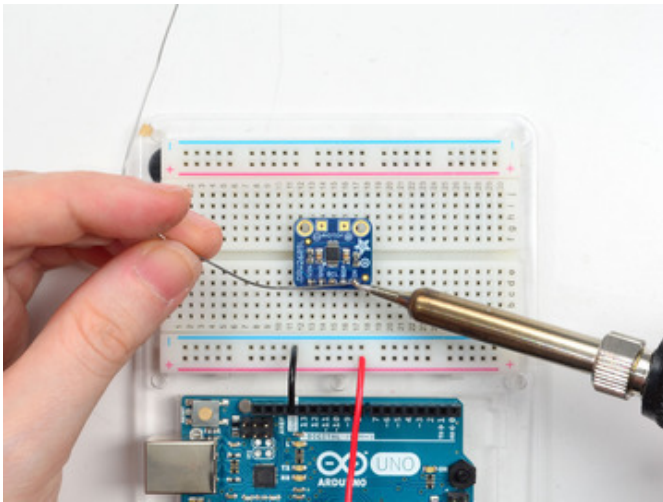


And Solder!

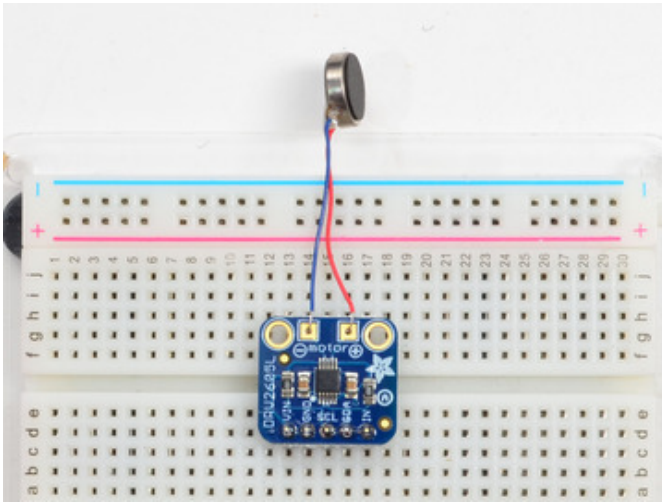
Be sure to solder all pins for reliable electrical contact.

Solder the longer power/data strip first

(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](https://adafruit.it/aTk) (<https://adafruit.it/aTk>)).

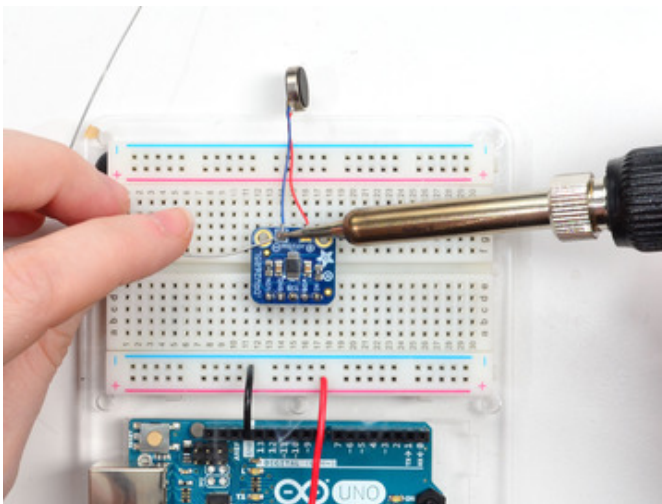


You're done! Check your solder joints visually and continue onto the next steps

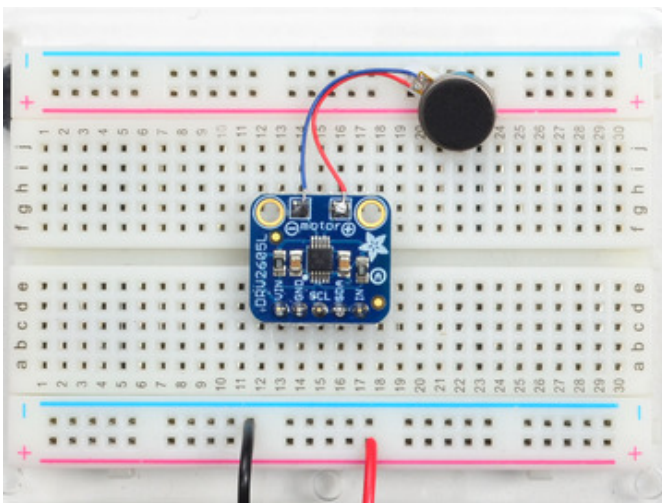


Attach Motor

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



Solder in place

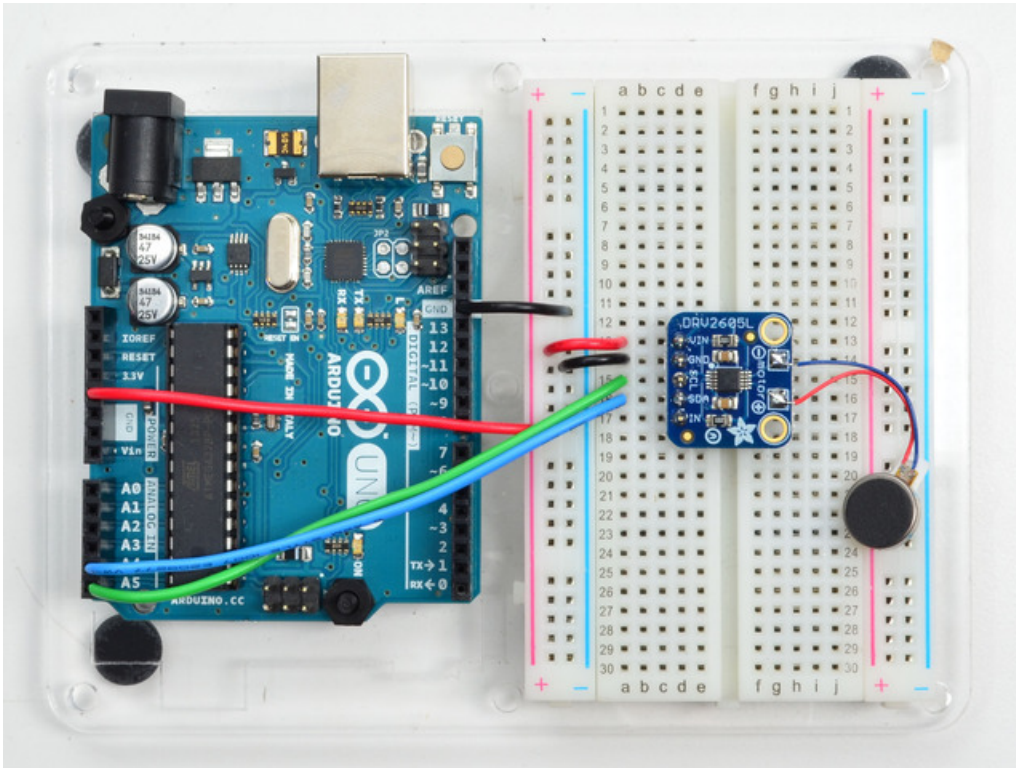


Check your work and continue!

Arduino Code

Wiring for Arduino

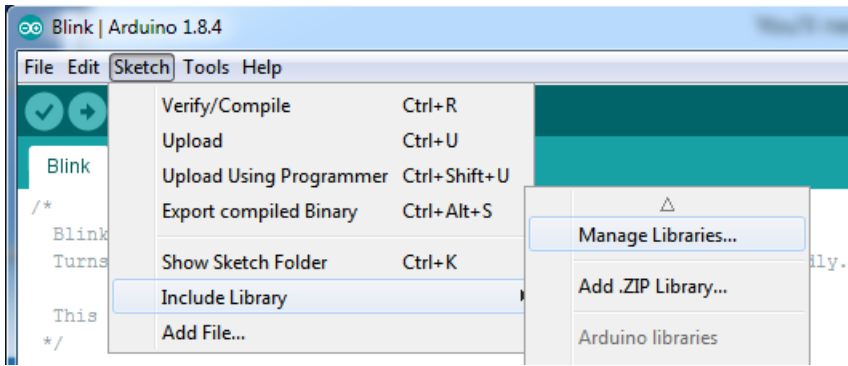
You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C capability, then port the code - its pretty simple stuff!



- Connect **Vin** to the power supply, 3-5V is fine. 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
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. 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. 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**

Install Adafruit_DRV2605 Library

To begin controlling 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...**



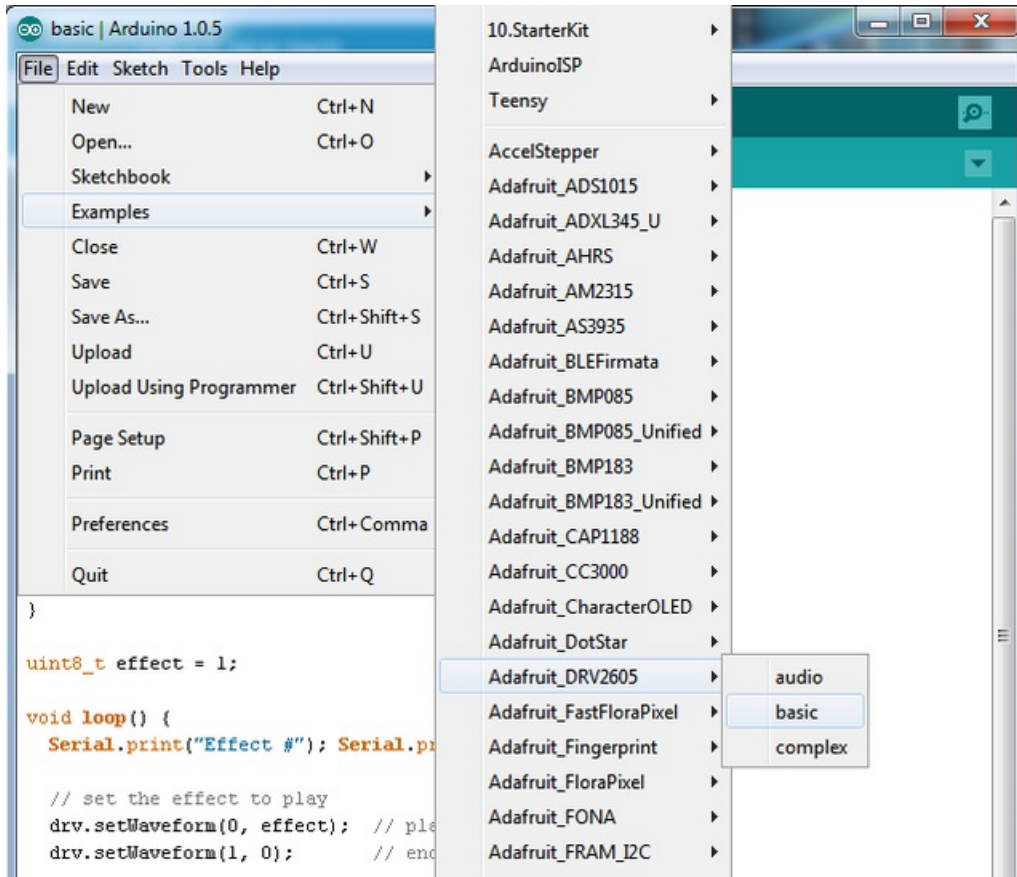
Then search for DRV2605 and find the **Adafruit DRV2605 Library** and click **Install**



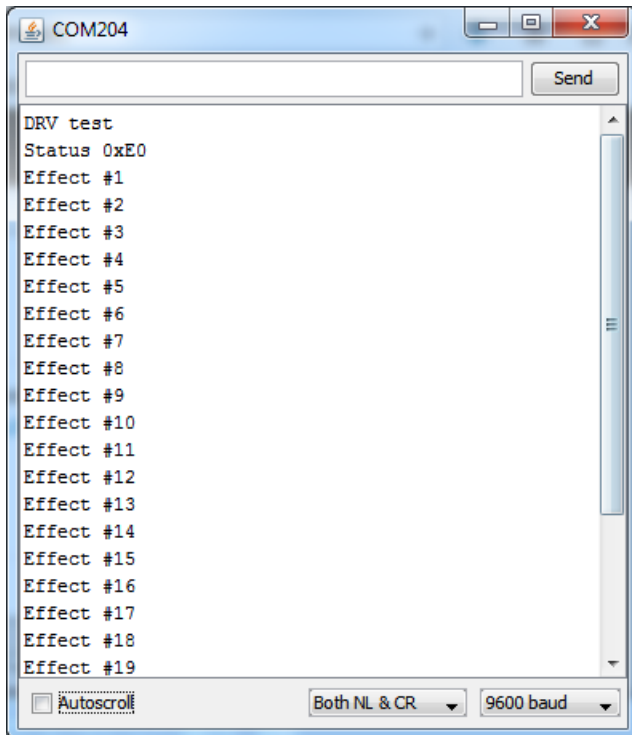
We also have a great tutorial on Arduino library installation at:
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>

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 117 built in vibration effects in order. [The full list with names is available in the DRV2605 datasheet](#)

Here's a screenshot for quick reference

11.2 Waveform Library Effects List

EFFECT ID NO.	WAVEFORM NAME	EFFECT ID NO>	WAVEFORM NAME	EFFECT ID NO.	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%

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 its the end.

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

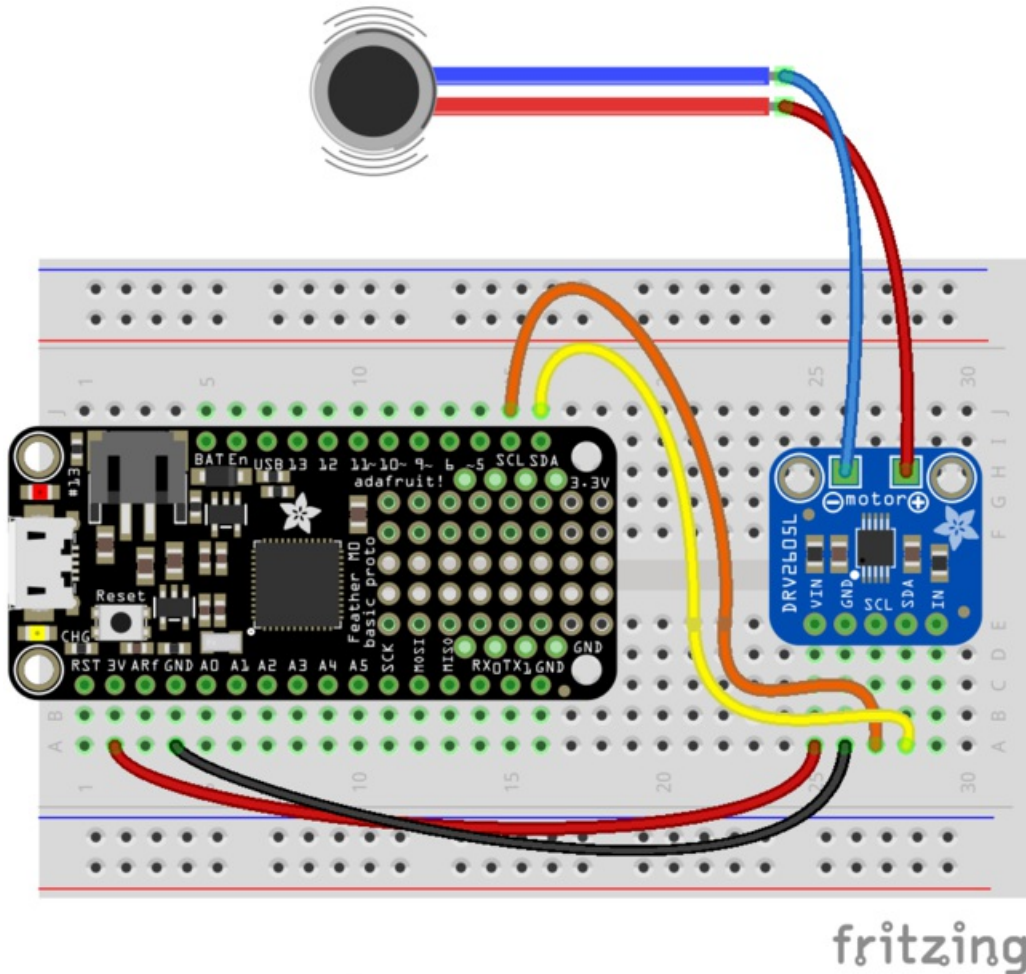
Audio

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!

CircuitPython Code

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

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
- Board GND to controller GND
- Board SCL to controller SCL
- Board SDA to controller SDA
- Controller Motor - to motor negative / blue wire.
- Controller Motor + to motor positive / red wire.

Library Installation

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`

You can also download the `adafruit_drv2605.mpy` from [its releases page on Github](#).

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.

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)
```

Remember if you're using a board that doesn't support hardware I2C (like the ESP8266) you need to use the **bitbangio** module instead:

```
import board
import bitbangio
import adafruit_drv2605
i2c = bitbangio.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, but for example a strong click with effect ID #1 would be:

```
drv.set_waveform(1)
```

Simply call the `set_waveform` function and pass the effect ID number of the desired effect (a value from 1 to 123). For reference here's a quick snapshot of the effects from [table 11.2 of the datasheet](#):

11.2 Waveform Library Effects List

EFFECT ID NO.	WAVEFORM NAME	EFFECT ID NO.	WAVEFORM NAME	EFFECT ID NO.	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%

Now to play the effect 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 sharp click effect.

Try selecting a different effect and playing it, like the strong buzz with effect ID 47:

```
drv.set_waveform(47)
drv.play()
```

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

There are a few other handy features to be aware of with the DRV2605 module. One is that you can stop playback of effects with the `stop` function:

```
drv.stop()
```

You can also combine multiple effects to play back at once and create interesting compound effects. Read the datasheet for more details but there are 7 slots which can each be independently loaded with an effect to play back

when the `play` function is called. Simply provide the slot number as an optional keyword argument to `set_waveform` (the default if not specified is slot 0).

For example try running:

```
drv.set_waveform(84)           # Effect 84 in slot 0
drv.set_waveform(1, slot=1)   # Effect 1 in slot 1
drv.set_waveform(0, slot=2)   # End effects after slot 1 (set slot 2 to 0)
drv.play()
```

This will build a compound effect from two waveforms:

- Effect #84 (ramp up, medium smooth) in slot 0, the default slot.
- Effect #1 (strong click) in slot 1

Notice how slot #2 is set to a value of zero. This tells the chip to stop adding compound effects after slot 1. You can add more effects in other slots up to a value of slot #6.

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!

Here's a complete example of using the board to play the 117 first 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. Remember to switch to the `bitbangio` module if necessary for your board!

```

# Simple demo of the DRV2605 haptic feedback motor driver.
# Will play all 117 effects in order for about a half second each.
# Author: Tony DiCola
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 (1-117).
# 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 = 1
while True:
    print('Playing effect #{0}'.format(effect))
    drv.set_waveform(effect) # Select the effect on slot 0.
    # Optionally you can assign effects to up to 7 different slots to combine
    # them in interesting ways. Use the slot keyword and specify a slot 0 to 6
    # (0 is the default).
    #drv.set_waveform(effect, slot=1)
    drv.play() # Play the effect.
    time.sleep(0.5)
    # Increment effect ID and wrap back around to 1.
    effect += 1
    if effect > 117:
        effect = 1

```

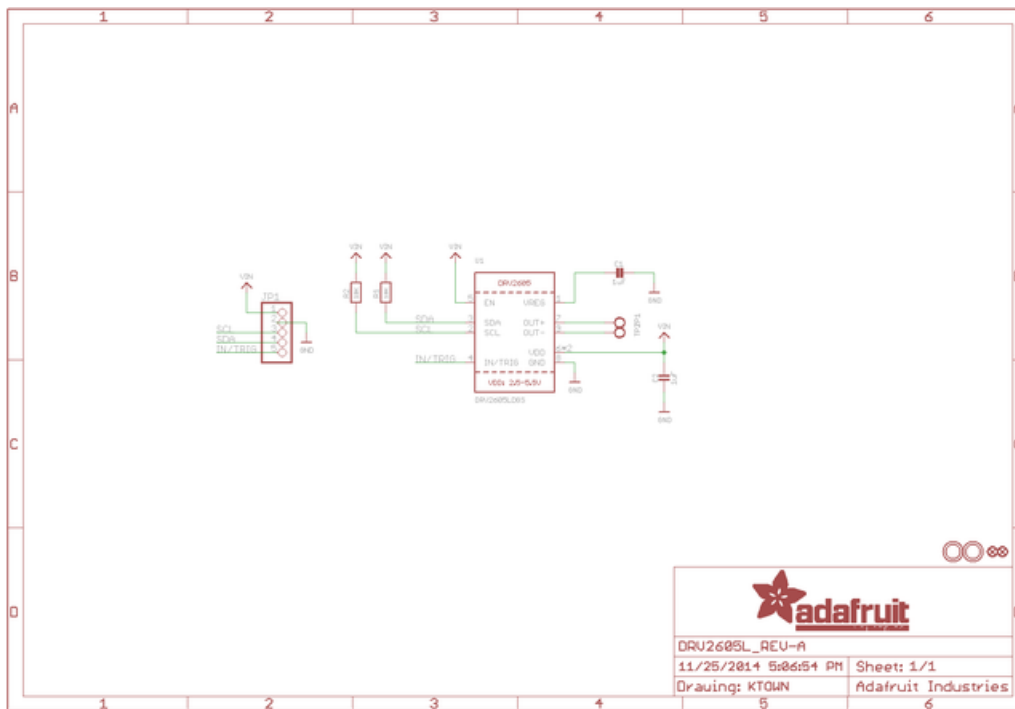
Downloads

Datasheets

- [DRV2605 Datasheet](#)
- [EagleCAD PCB files on GitHub](#)
- [Fritzing object in the Adafruit Fritzing Library](#)

Schematic

Click to embiggen



Fabrication print

Dimensions in Inches

