Adafruit DRV2605L Haptic Controller Breakout
Created by lady ada
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.(https://adafru.it/dDc)

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)).
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, 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 (https://adafruit.it/eh0). 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:

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 (https://adafru.it/eh2)

Here's a screenshot for quick reference
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!
Python & CircuitPython

It's easy to use the DRV2605 controller with Python or CircuitPython, and the Adafruit CircuitPython DRV2605 (https://adafruit.it/C4f) 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 (https://adafruit.it/BSN).

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
- 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.

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 (https://adafruit.it/BSN).

Here's the Raspberry Pi wired with I2C:

- Pi 3V3 to sensor VIN
- Pi GND to sensor GND
- Pi SCL to sensor SCL
- Pi SDA to sensor SDA

CircuitPython Installation of DRV2605 Library

You'll need to install the Adafruit CircuitPython DRV2605 (https://adafruit.it/C4f) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafruit.it/Amd) 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 (https://adafruit.it/zdx). Our introduction guide has a great page on how to install the library bundle (https://adafruit.it/ABU) 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 (https://adafruit.it/Awz) 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 (https://adafruit.it/BSN)!

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:

```python
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 (https://adafruit.it/eh2) for a list of all the effects:
Each effect needs to be loaded into one of eight slots in the Waveform Sequencer:

<table>
<thead>
<tr>
<th>Effect ID</th>
<th>Waveform Name</th>
<th>Effect ID</th>
<th>Waveform Name</th>
<th>Effect ID</th>
<th>Waveform Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Staccato</td>
<td>2</td>
<td>Harmonica</td>
<td>3</td>
<td>Buzzing</td>
</tr>
<tr>
<td>2</td>
<td>Scream</td>
<td>3</td>
<td>Buzzing</td>
<td>4</td>
<td>Bobbing</td>
</tr>
<tr>
<td>3</td>
<td>Grunt</td>
<td>4</td>
<td>Bobbing</td>
<td>5</td>
<td>Flapping</td>
</tr>
<tr>
<td>4</td>
<td>Buzzing</td>
<td>5</td>
<td>Flapping</td>
<td>6</td>
<td>Fluttering</td>
</tr>
<tr>
<td>5</td>
<td>Fluttering</td>
<td>6</td>
<td>Fluttering</td>
<td>7</td>
<td>Fluttering</td>
</tr>
<tr>
<td>6</td>
<td>Fluttering</td>
<td>7</td>
<td>Fluttering</td>
<td>8</td>
<td>Fluttering</td>
</tr>
<tr>
<td>7</td>
<td>Fluttering</td>
<td>8</td>
<td>Fluttering</td>
<td>9</td>
<td>Fluttering</td>
</tr>
<tr>
<td>8</td>
<td>Fluttering</td>
<td>9</td>
<td>Fluttering</td>
<td>10</td>
<td>Fluttering</td>
</tr>
</tbody>
</table>

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:

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

To play the sequence of effects call the **play** function:

```python
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 effect 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:

```python
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 combine multiple effects to play back at once 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:

```python
drv.sequence[0] = adafruit_drv2605.Effect(1)
drv.sequence[1] = adafruit_drv2605.Pause(0.5)
drv.sequence[3] = 0 # stops the sequence here
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:

```python
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:

```python
drv.use_LRM()
```

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

drv.use_ERM()
Downloads

Datasheets

- EagleCAD PCB files on GitHub (https://adafruit.it/oEt)
- Fritzing object in the Adafruit Fritzing Library (https://adafruit.it/aP3)

Schematic

Click to embiggen

Fabrication print

Dimensions in Inches