# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td><strong>Pinouts</strong></td>
<td>5</td>
</tr>
<tr>
<td>• Power Pins</td>
<td></td>
</tr>
<tr>
<td>• Data Pins</td>
<td></td>
</tr>
<tr>
<td>• MIDI Pins</td>
<td></td>
</tr>
<tr>
<td><strong>Arduino MIDI Example</strong></td>
<td>7</td>
</tr>
<tr>
<td>• Demo</td>
<td></td>
</tr>
<tr>
<td>• Libraries</td>
<td></td>
</tr>
<tr>
<td>• Upload Code</td>
<td></td>
</tr>
<tr>
<td><strong>CircuitPython MIDI Example</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>Downloads</strong></td>
<td>11</td>
</tr>
<tr>
<td>• Files:</td>
<td></td>
</tr>
<tr>
<td>• Schematic and Fab Print</td>
<td></td>
</tr>
<tr>
<td>• 3D Model</td>
<td></td>
</tr>
</tbody>
</table>
Overview

Turn your Feather into a song-bird with this musically-enabled FeatherWing that adds MIDI input and output jacks to just about any Feather. You get both input and output DIN-5 MIDI jacks, a 3V optical isolator so you can interface with MIDI on 3.3V logic/power microcontrollers, and two blinky indicator LEDs underneath the jacks to help you know when data is sent and received.

For those who have moved from DIN-5 jacks to "TRS MIDI A" 3.5mm jacks (), we provide spots to solder in 3.5mm stereo jacks (optional and not included by default)
We use the hardware serial pins RX and TX to send/receive data - you'll need to set these to 31250 baud in your programming language and then send/receive MIDI packet data. For example, here's how you would do it in Arduino thanks to this full-featured library with helpers. Here's an example on using a UART for MIDI transport in CircuitPython.

Because we use the UART, this works with all Feathers except for those with USB-Serial converters that use the UART pins. Right now that means the ESP8266 Huzzah Feather, 328p Feather and nRF52 Feather don't work because they use the hardware UART for programming. Any other Feathers with native USB will work just fine.
Comes as a kit with an assembled PCB that has all the surface-mount components attached, loose headers, and two loose DIN-5 jacks. Some basic soldering is required to assemble the header and DIN jacks in place. If you like to use 3.5mm jacks, you'll want to pick up some through-hole 3.5mm stereo jacks and solder those in, instead.

Pinouts
Power Pins

This FeatherWing is powered from the Feather 3V power pin, and uses the common GND pin.

Data Pins
We use the hardware serial pins RX and TX to send/receive data - you'll need to set these to 31250 baud in your programming language and then send/receive MIDI packet data. Because we use the UART, this works with all Feathers except for those with USB-Serial converters that use the UART pins. Right now that means the ESP8266 Huzzah Feather, 328p Feather and nRF52 Feather don't work because they use the hardware UART for programming. Any other Feathers with native USB will work just fine.

**MIDI Pins**

![MIDI Pins Schematic](image)

Solder your MIDI or 3.5mm stereo jacks to these pins.

**Arduino MIDI Example**

**Demo**

This example plays notes on MIDI channel 1 with MIDI NoteOn and NoteOff to specify note values and velocity.

Plug a DIN-5 MIDI cable or 3.5mm TRS cable into the MIDI FeatherWing MIDI out port on one end and your favorite MIDI-capable synthesizer on the other.

In the example shown in the video above, a 3.5mm TRS MIDI cable is being used to have the Feather send notes to a KORG NTS-1 synthesizer module.

To use the MIDI FeatherWing with Arduino, first, get set up with Arduino IDE as detailed here.
Libraries

To add libraries, in the Arduino IDE click Sketch > Include Library > Manage Libraries...

In the Library Manager window, search for the MIDI Library by Francois Best, Iathoub and install the latest version.

Note: this is the same library that used to be known as the Forty Seven Effects MIDI library. Loads of great details to be had here!

Upload Code

Copy the code example below then paste it into a new Arduino document. Save the Arduino file, then compile and upload it to your Feather board. Be sure to pick the USB Stack > Tiny USB option in the Tools menu before compiling and uploading.

Once uploaded, it will begin sending MIDI to your synth!
CircuitPython MIDI Example

Here's a quick example to get you started with MIDI in CircuitPython.

Install the latest CircuitPython () on your Feather board, then download the Library bundle ()

Drag a copy of the adafruit_midi library onto your CIRCUITPY drive's lib folder

You can check out lots more details on this library here ()

This example will play notes on MIDI channel 1 with MIDI NoteOn and NoteOff as well as send modulation changes with the ControlChange command.

Plug a DIN-5 MIDI cable or 3.5mm TRS cable into the MIDI FeatherWing MIDI out port on one end and your favorite MIDI-capable synthesizer on the other.
In the example shown in the video above, a classic DIN-5 MIDI cable is being used to have the Feather send notes to a Doepfer MIDI-to-CV/Gate/CC converter, which in turn drives notes, gate, and modulation on a Winterbloom Castor & Pollux Eurorack synthesizer module.

Copy the code below and paste it into Mu, then save it to your CIRCUITPY drive as code.py.

Once the file is saved to the Feather, it will begin sending MIDI notes to your synthesizer, drum machine, or other rad groove machine!

```
# SPDX-FileCopyrightText: 2021 John Park for Adafruit Industries
# SPDX-License-Identifier: MIT
# midi_UARToutdemo.py - demonstrates sending MIDI notes

import time
import board
import busio
import adafruit_midi
from adafruit_midi.control_change import ControlChange
from adafruit_midi.note_off import NoteOff
from adafruit_midi.note_on import NoteOn

uart = busio.UART(board.TX, board.RX, baudrate=31250, timeout=0.001)  # init UART
midi_in_channel = 2
midi_out_channel = 1
midi = adafruit_midi.MIDI(
    midi_in=uart,
    midi_out=uart,
    in_channel=(midi_in_channel - 1),
    out_channel=(midi_out_channel - 1),
    debug=False,
)

note_hold = 0.85
rest = note_hold / 5

print("MIDI Out demo")
print("Default output channel: ", midi.out_channel + 1)

while True:
    # midi.send(ControlChange(64, 0))  # sustain CC
    midi.send(ControlChange(1, 0))  # modulation CC
    midi.send(NoteOn(48, 20))  # play note
    time.sleep(note_hold)  # hold note
    midi.send(NoteOff(48, 0))  # release note
    time.sleep(rest)  # rest

    midi.send(NoteOn(55, 40))
    time.sleep(note_hold)
    midi.send(NoteOff(55, 0))
    time.sleep(rest)

    midi.send(NoteOn(51, 60))
    time.sleep(note_hold)
    midi.send(NoteOff(51, 0))
    time.sleep(rest)

    midi.send(NoteOn(58, 80))
    time.sleep(note_hold)
    midi.send(NoteOff(58, 0))
```
time.sleep(rest)

# midi.send(ControlChange(64, 32))
midi.send(ControlChange(1, 127))

midi.send(NoteOn(48, 20))  # play note
time.sleep(note_hold)  # hold note
midi.send(NoteOff(48, 0))  # release note
time.sleep(rest)  # rest

midi.send(NoteOn(55, 40))
time.sleep(note_hold)
midi.send(NoteOff(55, 0))
time.sleep(rest)

midi.send(NoteOn(51, 60))
time.sleep(note_hold)
midi.send(NoteOff(51, 0))
time.sleep(rest)

midi.send(NoteOn(50, 80))
time.sleep(note_hold)
midi.send(NoteOff(50, 0))
time.sleep(rest)

Downloads

Files:

- [EagleCAD files on GitHub](#)
- [Fritzing object in Adafruit Fritzing Library](#)
- [Fritzing object (3.5mm jacks) in Adafruit Fritzing Library](#)
- [3D models on GitHub](#)
Schematic and Fab Print
3D Model