MIDI Laser Harp with Time of Flight Distance Sensors

Created by Liz Clark


Last updated on 2023-08-29 03:47:23 PM EDT
# Table of Contents

## Overview
- Prerequisite Guides
- Parts
- OR

## Circuit Diagram

## Acrylic and 3D Printing

## CircuitPython on Feather M4 Express
- Set up CircuitPython Quick Start!
- Further Information

## Coding the Laser Harp
- Upload the Code and Libraries to the Feather M4 Express
- Additional Examples
- How the CircuitPython Code Works

## Assembly
- Acrylic and 3D Printed Parts
- Mount the Lasers and ToF Sensors

## STEMMA Wiring
- Prep the STEMMA QT Cables
- Plug in the STEMMA QT Cables
- Extend the SCL and SDA Wires

## TCA9548A Wiring

## Wiring the Lasers
- Power Connections
- Ground Connections
- Heat Shrink

## Laser Switch

## FeatherWing Doubler
- Headers
- Laser Power and Ground
- TCA9548A Connections
- STEMMA Power and Ground
- Mount the FeatherWing Doubler and TCA9548A

## Usage
Overview

A **laser harp** is an electronic instrument that lets you "pluck" laser beams like a harp to play notes. In this guide, you'll learn how to build your own version of a laser harp using a Feather M4 Express or Feather RP2040, a Music Maker FeatherWing and code written in CircuitPython. VL53L4CD time of flight sensors are used to detect when you're playing a note instead of simply detecting the break-beam. This means we can do cool stuff like use the distance data from the time of flight sensors to convert into different MIDI values such as modulation, sustain and velocity.

We use a Music Maker wing which has audio output from MIDI input and a **wide variety of musical instruments** it can generate sound effects for, see page 32 of the VS1053 datasheet for a full list! Of course you can also generate MIDI-over-USB signals that can be used to control your favorite software synth.

It's hard to mount sensors on the other side of the lasers so we 'cheat' and use high quality ToF distance sensors with precise distance measurements and narrow detection paths. The lasers are only there help you to visualize the detection path of the time of flight sensors so feel free to change the laser colors or positions as you please. The sensors and lasers are mounted so that they are sitting at the same angle.
The Music Maker FeatherWing is used as a MIDI synth. You can change the instrument sound in the CircuitPython code.

The TCA9548A I2C multiplexer allows you to use eight of the VL53L4CD time of flight sensors at the same time even though they all have the same I2C address.

**Prerequisite Guides**

Please look over the following Adafruit Learning System guides for information on the components used.

- Adafruit Music Maker FeatherWing
- Adafruit VL53L4CD Time of Flight Distance Sensor
- Working with Multiple Same Address I2C Devices
- Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout
Although the guide shows a Feather M4 Express being used, a Feather RP2040 is also an excellent choice!

Parts

Adafruit Feather RP2040
A new chip means a new Feather, and the Raspberry Pi RP2040 is no exception. When we saw this chip we thought "this chip is going to be awesome when we give it the Feather..."
https://www.adafruit.com/product/4884

OR

Adafruit Feather M4 Express - Featuring ATSAMD51
It's what you've been waiting for, the Feather M4 Express featuring ATSAMD51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,...
https://www.adafruit.com/product/3857

Adafruit Music Maker FeatherWing - MP3 OGG WAV MIDI Synth Player
Bend all audio files to your will with the Adafruit Music Maker FeatherWing! It's a fun-size version of our Music...
https://www.adafruit.com/product/3357
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FeatherWing Doubler</td>
<td>FeatherWing Doubler - Prototyping Add-on For All Feather Boards</td>
<td><a href="https://www.adafruit.com/product/2890">https://www.adafruit.com/product/2890</a></td>
</tr>
<tr>
<td>Fixed Laser Mounting Stand</td>
<td>Illustration of Fixed Laser Mounting Stand</td>
<td><a href="https://www.adafruit.com/product/1094">https://www.adafruit.com/product/1094</a></td>
</tr>
<tr>
<td>TCA9548A I2C Multiplexer</td>
<td>You just found the perfect I2C sensor, and you want to wire up two or three or more of them to your Arduino when you realize &quot;Uh oh, this chip has a fixed I2C address, and from...</td>
<td><a href="https://www.adafruit.com/product/2717">https://www.adafruit.com/product/2717</a></td>
</tr>
<tr>
<td>Adafruit VL53L4CD Time of Flight Distance Sensor - ~1 to 1300mm</td>
<td>The Adafruit VL53L4CD Time of Flight Sensor is another great Time of Flight distance sensor from ST in the VL5 series of chips, this one is great for...</td>
<td><a href="https://www.adafruit.com/product/5396">https://www.adafruit.com/product/5396</a></td>
</tr>
<tr>
<td>Laser Diode - 5mW 650nm Red</td>
<td>You have some frickin' sharks at your lair, and you're thinking &quot;it would be really great if I could attach some lasers to them&quot; and then you see these little laser...</td>
<td><a href="https://www.adafruit.com/product/1054">https://www.adafruit.com/product/1054</a></td>
</tr>
<tr>
<td>Description</td>
<td>Quantity</td>
<td>URL</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>100mm Long STEMMA QT cable</td>
<td>8 x</td>
<td><a href="https://www.adafruit.com/product/4210">https://www.adafruit.com/product/4210</a></td>
</tr>
<tr>
<td>100mm Long STEMMA QT cable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 x STEMMA QT / Qwiic JST SH 4-pin Cable - 100mm Long STEMMA QT cable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 x Black Nylon Machine Screw and Stand-off Set – M2.5 Thread M2.5 screws</td>
<td></td>
<td><a href="https://www.adafruit.com/product/3299">https://www.adafruit.com/product/3299</a></td>
</tr>
<tr>
<td>1 x Silicone Cover Stranded-Core Wire - 50ft 30AWG Black Black wire</td>
<td></td>
<td><a href="https://www.adafruit.com/product/3164">https://www.adafruit.com/product/3164</a></td>
</tr>
<tr>
<td>1 x Silicone Cover Stranded-Core Wire - 50ft 30AWG Red Red wire</td>
<td></td>
<td><a href="https://www.adafruit.com/product/3165">https://www.adafruit.com/product/3165</a></td>
</tr>
<tr>
<td>1 x Silicone Cover Stranded-Core Wire - 50ft 30AWG Yellow Yellow wire</td>
<td></td>
<td><a href="https://www.adafruit.com/product/3167">https://www.adafruit.com/product/3167</a></td>
</tr>
<tr>
<td>1 x Silicone Cover Stranded-Core Wire - 50ft 30AWG Blue Blue wire</td>
<td></td>
<td><a href="https://www.adafruit.com/product/3166">https://www.adafruit.com/product/3166</a></td>
</tr>
<tr>
<td>1 x Acrylic sheet 18&quot; x 24&quot; sheet of 1/8&quot; thick acrylic</td>
<td></td>
<td><a href="https://www.inventables.com/technologies/transparent-yellow-acrylic-sheet">https://www.inventables.com/technologies/transparent-yellow-acrylic-sheet</a></td>
</tr>
</tbody>
</table>
Circuit Diagram

TCA9548A

- SC7 to sensor 1 SCL
- SD7 to sensor 1 SDA
- SC6 to sensor 2 SCL
- SD6 to sensor 2 SDA
- SC5 to sensor 3 SCL
- SD5 to sensor 3 SDA
- SC4 to sensor 4 SCL
- SD4 to sensor 4 SDA
- SC3 to sensor 5 SCL
- SD3 to sensor 5 SDA
- SC2 to sensor 6 SCL
- SD2 to sensor 6 SDA
- SC1 to sensor 7 SCL
- SD1 to sensor 7 SDA
- SC0 to sensor 8 SCL
- SD0 to sensor 8 SDA
- SDA to board SDA
- SCL to board SCL
- GND to board GND
- VIN to board 3.3V
Feather RP2040

- GND to STEMMA QT GND
- GND to lasers' GND
- 3.3V to STEMMA QT VIN
- 3.3V to switch
- Plug the Feather RP2040 and Music Maker FeatherWing into a FeatherWing Doubler

STEMMA and Laser VIN and GND Connections

- Connect the VL53L4CDs' VIN and GND connections together with STEMMA QT cables
- Wire the lasers' GND connections (black wires) together
- Wire the lasers' VIN connections (red wires) together and connect them to the switch

Acrylic and 3D Printing

The laser harp may be constructed with a piece of acrylic and some 3D printed parts, described below. The acrylic stand is shaped in an arc with mounting holes for the electronics and 3D printed parts.
The 3D printed parts consist of brackets for the laser mounts and time of flight sensors, stands to prop up the piece of acrylic and fasteners to hold the acrylic stand together if it is being cut from two pieces. All parts print with no supports.

The STL, DXF and SVG files can be downloaded directly here or from Thingiverse.

- **Thingiverse download**
- **laserHarpAssemblyFiles.zip**

The acrylic can be cut using manual tools, a CNC or laser cutter. If you're using manual tools, you can cut or print out the files on paper to use as a guide for cutting and drilling.

Files are available to cut the acrylic as one large piece or in half using two smaller pieces of acrylic stock.
One of the 3D printed stands has mounting holes for the FeatherWing Doubler.

You'll print eight brackets, one for each of the lasers and time of flight sensors. The bracket allows the components to attach to the acrylic and for the time of flight sensor to be level with the laser.

CircuitPython on Feather M4 Express

CircuitPython is a derivative of MicroPython designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.
The following instructions will show you how to install CircuitPython. If you've already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)

Download the latest version of CircuitPython for this board via CircuitPython.org

Click the link above and download the latest UF2 file.

Download and save it to your desktop (or wherever is handy).
Plug your Feather M4 into your computer using a known-good USB cable.

A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

Double-click the Reset button next to the USB connector on your board, and you will see the NeoPixel RGB LED turn green. If it turns red, check the USB cable, try another USB port, etc. Note: The little red LED next to the USB connector will pulse red. That's ok!

If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!

You will see a new disk drive appear called FEATHERBOOT.

Drag the adafruit_circuitpython_etc.uf2 file to FEATHERBOOT.
The LED will flash. Then, the FEATHERBOOT drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :)

Further Information

For more detailed info on installing CircuitPython, check out Installing CircuitPython.

Coding the Laser Harp

Although the documentation for this project shows a Feather M4 Express, a Feather RP2040 will work too.

Once you've finished setting up your Feather M4 Express with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download as a zipped folder.

```python
# SPDX-FileCopyrightText: 2022 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

import board
import busio
import simpleio
import adafruit_vl53l4cd
import adafruit_tca9548a
import adafruit_midi
from adafruit_midi.note_off import NoteOff
from adafruit_midi.note_on import NoteOn
from adafruit_midi.program_change import ProgramChange
from adafruit_midi.control_change import ControlChange

# Create I2C bus as normal
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a microcontroller

# Create the TCA9548A object and give it the I2C bus
tca = adafruit_tca9548a.TCA9548A(i2c)
```
# setup time of flight sensors to use TCA9548A inputs

tof_0 = adafruit_vl53l4cd.VL53L4CD(tca[0])
tof_1 = adafruit_vl53l4cd.VL53L4CD(tca[1])
tof_2 = adafruit_vl53l4cd.VL53L4CD(tca[2])
tof_3 = adafruit_vl53l4cd.VL53L4CD(tca[3])
tof_4 = adafruit_vl53l4cd.VL53L4CD(tca[4])
tof_5 = adafruit_vl53l4cd.VL53L4CD(tca[5])
tof_6 = adafruit_vl53l4cd.VL53L4CD(tca[6])
tof_7 = adafruit_vl53l4cd.VL53L4CD(tca[7])

# array of tof sensors
flights = [tof_0, tof_1, tof_2, tof_3, tof_4, tof_5, tof_6, tof_7]

# setup each tof sensor
for flight in flights:
    flight.inter_measurement = 0
    flight.timing_budget = 50
    flight.start_ranging()

# midi uart setup for music maker featherwing
uart = busio.UART(board.TX, board.RX, baudrate=31250)

midi_in_channel = 1
midi_out_channel = 1

# midi setup
# UART is setup as the input
midi = adafruit_midi.MIDI(
    midi_in=uart,
    midi_out=uart,
    in_channel=(midi_in_channel - 1),
    out_channel=(midi_out_channel - 1),
    debug=False,
)

# height cutoff for tof sensors
# adjust depending on the height of your ceiling/performance area
flight_height = 150

# state of each tof sensor
# tracks if you have hit the laser range
pluck_0 = False
pluck_1 = False
pluck_2 = False
pluck_3 = False
pluck_4 = False
pluck_5 = False
pluck_6 = False
pluck_7 = False

# array of tof sensor states
plucks = [pluck_0, pluck_1, pluck_2, pluck_3, pluck_4, pluck_5, pluck_6, pluck_7]

# midi notes for each tof sensor
notes = [48, 52, 55, 59, 60, 64, 67, 71]

# midi instrument voice
midi.send(ProgramChange(80))

while True:
    # iterate through the 8 tof sensors
    for f in range(8):
        while not flights[f].data_ready:
            pass
        # reset tof sensors
        flights[f].clear_interrupt()
        # if the reading from a tof is not 0...
        if flights[f].distance != 0.0:
            # map range of tof sensor distance to midi parameters
            # modulation
mod = round(simpleio.map_range(flights[f].distance, 0, 100, 120, 0))
# sustain
sus = round(simpleio.map_range(flights[f].distance, 0, 100, 127, 0))
# velocity
vel = round(simpleio.map_range(flights[f].distance, 0, 150, 120, 0))
modulation = int(mod)
sustain = int(sus)
# create sustain and modulation CC message
pedal = ControlChange(71, sustain)
modWheel = ControlChange(1, modulation)
# send the sustain and modulation messages
midi.send([modWheel, pedal])
# if tof registers a height lower than the set max height...
if int(flights[f].distance) < flight_height and not plucks[f]:
    # set state tracker
    plucks[f] = True
    # convert tof distance to a velocity value
    velocity = int(vel)
    # send midi note with velocity and sustain message
    midi.send([NoteOn(notes[f], velocity), pedal])
# if tof registers a height = to or greater than set max height
# aka you remove your hand from above the sensor...
if int(flights[f].distance) > flight_height and plucks[f]:
    # reset state
    plucks[f] = False
    # send midi note off
    midi.send(NoteOff(notes[f], velocity))

Upload the Code and Libraries to the Feather M4 Express

After downloading the Project Bundle, plug your Feather M4 Express into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the Feather M4 Express' CIRCUITPY drive.

- lib folder
- code.py

Your Feather M4 Express CIRCUITPY drive should look like this after copying the lib folder and the code.py file.
Additional Examples

In addition to the main code.py file, there are two more CircuitPython code files that you can use for different features.

The **usb_midi_code.py** file has code to allow the laser harp to be used as a USB MIDI controller. This way you can control your favorite software synth or DAW.

The **laser_harp_two_voice.py** file has extra features that build on the original code.py file. It can send out two different synth instruments depending on the height that the time of flight sensors detect. You can also control pitch bend or volume by raising or lowering your hand while playing a note.

To use either of these files instead of the original code.py file, remove the code.py file from your CIRCUITPY drive and rename your chosen .py file as code.py.

How the CircuitPython Code Works

The code begins by setting up I2C to use `board.SCL` and `board.SDA`. The TCA9548A I2C multiplexer is setup as `tca`.

```python
# Create I2C bus as normal
i2c = board.I2C()  # uses board.SCL and board.SDA

# Create the TCA9548A object and give it the I2C bus
# tca = adafruit_tca9548a.TCA9548A(i2c)
```
VL53L4CD Setup with the TCA9548A

The time of flight sensors all use the same I2C address, \texttt{0x29}. When they are setup using the \texttt{adafruit_vl53l4cd} library, their I2C pins are set as different channels on the TCA9548A.

The time of flight sensors are inserted into an array called \texttt{flights}. A \texttt{for} statement is used to setup each time of flight sensor with \texttt{inter\_measurement} and \texttt{timing\_budget} values, along with the function \texttt{start\_ranging()} to begin reading data.

```python
# setup time of flight sensors to use TCA9548A inputs
tof_0 = adafruit_vl53l4cd.VL53L4CD(tca[0])
tof_1 = adafruit_vl53l4cd.VL53L4CD(tca[1])
tof_2 = adafruit_vl53l4cd.VL53L4CD(tca[2])
tof_3 = adafruit_vl53l4cd.VL53L4CD(tca[3])
tof_4 = adafruit_vl53l4cd.VL53L4CD(tca[4])
tof_5 = adafruit_vl53l4cd.VL53L4CD(tca[5])
tof_6 = adafruit_vl53l4cd.VL53L4CD(tca[6])
tof_7 = adafruit_vl53l4cd.VL53L4CD(tca[7])

# array of tof sensors
flights = [tof_0, tof_1, tof_2, tof_3, tof_4, tof_5, tof_6, tof_7]

# setup each tof sensor
for flight in flights:
    flight.inter_measurement = 0
    flight.timing_budget = 50
    flight.start_ranging()
```

MIDI Over UART Setup

MIDI is setup to use UART. The Music Maker FeatherWing takes in MIDI over UART to be used as a synth.

```python
# midi uart setup for music maker featherwing
uart = busio.UART(board.TX, board.RX, baudrate=31250)

midi_in_channel = 1
midi_out_channel = 1

# midi setup
# UART is setup as the input
midi = adafruit_midi.MIDI(
    midi_in=uart,
    midi_out=uart,
    in_channel=(midi_in_channel - 1),
    out_channel=(midi_out_channel - 1),
    debug=False,
)
```
Variables and States

There are a few variables that you may want to change depending on your preferences.

- **flight_height** is used as a maximum height value for the time of flight sensors. Adjust this to increase or decrease the expected range for the laser harp
- **notes** is the array of MIDI notes assigned to each time of flight sensor. Change these numbers to play different notes
- The **midi.send(ProgramChange(80))** message changes the instrument sound being used by the Music Maker FeatherWing. You can change the number to set a different instrument sound. Check out this reference page for a list of possible sounds and their numbers.()

```python
# height cutoff for tof sensors
# adjust depending on the height of your ceiling/performance area
flight_height = 150

# state of each tof sensor
# tracks if you have hit the laser range
pluck_0 = False
pluck_1 = False
pluck_2 = False
pluck_3 = False
pluck_4 = False
pluck_5 = False
pluck_6 = False
pluck_7 = False

# array of tof sensor states
plucks = [pluck_0, pluck_1, pluck_2, pluck_3, pluck_4, pluck_5, pluck_6, pluck_7]

# midi notes for each tof sensor
notes = [48, 52, 55, 59, 60, 64, 67, 71]

# midi instrument voice
midi.send(ProgramChange(80))
```

The Loop

In the loop, the time of flight sensors are iterated through and their values are read. The values are mapped to different MIDI parameters: modulation, sustain and velocity. Modulation and sustain are sent as a MIDI message together.

```python
while True:
    # iterate through the 8 tof sensors
    for f in range(8):
        while not flights[f].data_ready:
            pass
    # reset tof sensors
    flights[f].clear_interrupt()
```
Playing the Laser Harp

If the time of flight sensor detects a height that is lower than the maximum threshold set as `flight_height`, the assigned MIDI note is sent with a `NoteOn` message. If the time of flight sensor detects a height that is equal to or lower than `flight_height`, a `NoteOff` message is sent.

```python
# if tof registers a height lower than the set max height... 
if int(flights[f].distance) < flight_height and not plucks[f]:
    # set state tracker 
    plucks[f] = True
    # convert tof distance to a velocity value 
    velocity = int(vel)
    # send midi note with velocity and sustain message 
    midi.send((NoteOn(notes[f], velocity), pedal))
# if tof registers a height = to or greater than set max height 
# aka you remove your hand from above the sensor... 
if int(flights[f].distance) > flight_height and plucks[f]:
    # reset state 
    plucks[f] = False
    # send midi note off 
    midi.send(NoteOff(notes[f], velocity))
```

```python
# if the reading from a tof is not 0...
if flights[f].distance != 0.0:
    # map range of tof sensor distance to midi parameters
    # modulation
    mod = round(simpleio.map_range(flights[f].distance, 0, 100, 120, 0))
    # sustain
    sus = round(simpleio.map_range(flights[f].distance, 0, 100, 127, 0))
    # velocity
    vel = round(simpleio.map_range(flights[f].distance, 0, 150, 120, 0))
    modulation = int(mod)
    sustain = int(sus)
    # create sustain and modulation CC message
    pedal = ControlChange(71, sustain)
    modWheel = ControlChange(1, modulation)
    # send the sustain and modulation messages
    midi.send([modWheel, pedal])
```
Assembly

If you've cut the acrylic using smaller stock, connect the two pieces with the 3D printed connectors and M2.5 screws.

Acrylic and 3D Printed Parts
Attach the 3D printed stands to the back of the acrylic using M2.5 screws.

That completes the main acrylic assembly!
Mount the Lasers and ToF Sensors

Line up the laser mount's holes with the holes on the acrylic. Run the laser's wires through the bottom-left hole in the mount and the acrylic. This hole is larger on the acrylic.

Align the 3D printed mount's holes with the acrylic. Run the laser's wires through the larger hole of the 3D printed mount.
Attach the mount and laser to the acrylic using three M2.5 screws and nuts.

Attach the VL53L4CD board to the top of the 3D printed mount using four M2.5 screws and nuts.
Repeat this process for the remaining seven lasers and time of flight sensors.

STEMMA Wiring

STEMMA QT cables are used to connect the VL53L4CD time of flight sensors. The cables need to be cut so that the SCL and SDA wires can be soldered to the TCA9548A breakout board, while keeping the power and ground wires connected to the QT connectors.

Prep the STEMMA QT Cables

Cut the SCL and SDA (blue and yellow) wires close to one end of the QT connector. Slide those wires through the heatshrink tubing on the cable. Splice and tin the SCL and SDA wires.
Repeat this process for a total of seven cables.

For the eighth cable, cut the SCL and SDA (blue and yellow) wires close to one end of the QT connector. Then, at the opposite QT connector, cut the power and ground (red and black) wires.

You should have one cable that's power and ground only connected to a QT connector and another cable that's SCL and SDA only connected to a QT connector. Splice and tin the wires.

Plug in the STEMMA QT Cables

Plug the STEMMA QT cable into the first VL53L4CD board on the left. The connector with all four wires still connected should be plugged into the board's right-hand port. Plug the other end of the cable, with only power and ground connections, into the left-hand port of the next VL53L4CD.

Continue plugging in the remaining six cut cables in this pattern: connector with all four wires plugged into the right-hand port, connector with power and ground only into the left-hand port.
Plug the SCL and SDA only STEMMA QT cable into the open right-hand port in the last VL53L4CD board on the right.

Plug the power and ground only STEMMA QT cable into the open left-hand port in the first VL53L4CD board on the left.
Extend the SCL and SDA Wires

Cut, splice and tin 16 pieces of wire. These wires will be used to extend the SCL and SDA wires from the STEMMA QT cables. As a result, you can color coordinate them by cutting eight blue wires and eight yellow wires.

The wires should be the following approximate lengths:

- Four pieces (two blue, two yellow) measuring 32 mm
- Four pieces (two blue, two yellow) measuring 177.8 mm
- Four pieces (two blue, two yellow) measuring 285.75 mm
- Four pieces (two blue, two yellow) measuring 304.8 mm

Cut, splice and tin all of the wires' ends.

Solder the cut wires to the STEMMA QT wires.

The 304.8 mm wires are sized for the first and last VL53L4CD’s.
The 285.75 mm wires are sized for the second and seventh VL53L4CD’s.
The 177.8 mm wires are sized for the third and sixth VL53L4CD’s.
The 32 mm wires are sized for the fourth and fifth (two in the middle) VL53L4CD’s.
Add heat shrink to all of the soldered wire connections.

TCA9548A Wiring
Solder the first two VL53L4CD’s I2C connections to channel 0 and 1 on the TCA9548A.

Time of flight 1 SDA to SD0
Time of flight 1 SCL to SC0
Time of flight 2 SDA to SD1
Time of flight 2 SCL to SC1
Solder the third and fourth VL53L4CD's I2C connections to channel 2 and 3 on the TCA9548A.

Time of flight 3 SDA to SD2
Time of flight 3 SCL to SC2
Time of flight 4 SDA to SD3
Time of flight 4 SCL to SC3
Solder the fifth and sixth VL53L4CD's I2C connections to channel 4 and 5 on the TCA9548A.

Time of flight 5 SDA to SD4
Time of flight 5 SCL to SC4
Time of flight 6 SDA to SD5
Time of flight 6 SCL to SC5
Solder the last two VL53L4CD's I2C connections to channel 6 and 7 on the TCA9548A.

Time of flight 7 SDA to SD6
Time of flight 7 SCL to SC6
Time of flight 8 SDA to SD7
Time of flight 8 SCL to SC7
Wiring the Lasers

Power Connections

Slip a piece of heat shrink over the first laser's red power wire. Solder the first and second lasers' red power wires together.

Slip a piece of heat shrink over the third laser's red power wire. Solder the third and fourth lasers' red power wires together.
Cut, splice and tin two pieces of wire that are approximately 190.5 mm in length.

Solder one of the ends of the 190.5 mm lengths of wire to the solder point between the first and second laser. Then, solder the other end to the solder point between the third and fourth laser. This connects the first four laser's power connections together.
Slip a piece of heat shrink over the seventh laser's red power wire. Solder the seventh and eighth laser's red power wires together. Solder one of the ends of the second 190.5 mm length of wire to the solder point.

Slip a piece of heat shrink over the fifth laser's red power wire. Solder the fifth and sixth laser's red power wires together. Solder the other end of the 190.5 mm length of wire to the solder point. This connects the last four laser's power connections together.
Cut, splice and tin a piece of wire that is approximately 242 mm in length.

Solder one of the ends of the wire to the solder point between the third and fourth laser. Then, solder the other end to the solder point between the fifth and sixth laser. This connects all eight of the laser's power connections together.
Cut, splice and tin a piece of wire that is approximately 58 mm in length. Solder one of the ends of the wire to the solder point between the seventh and eighth laser. This wire will eventually connect to the on/off switch for the lasers.
Slip a piece of heat shrink over the first, third, fifth and seventh laser's black ground wires. Solder the following laser's black ground wires together:

- First and second
- Third and fourth
- Fifth and sixth
- Seventh and eighth
Cut, splice and tin two pieces of wire that are approximately 190.5 mm in length.
Solder one of the ends of the 190.5 mm lengths of wire to the ground solder point between the seventh and eighth laser. Then, solder the other end to the ground solder point between the fifth and sixth laser. This connects the last four laser's ground connections together.
Solder one of the ends of the second 190.5 mm length of wire to the ground solder point between the third and fourth laser. Then, solder the other end to the ground solder point between the first and second laser. This connects the first four laser's ground connections together.
Cut, splice and tin a piece of wire that is approximately 242 mm in length.

Solder one of the ends of the wire to the ground solder point between the third and fourth laser. Then, solder the other end to the ground solder point between the fifth and sixth laser. This connects all eight of the laser's ground connections together.
Cut, splice and tin a piece of wire that is approximately 152 mm in length. Solder one of the ends of the wire to the ground solder point between the seventh and eighth laser. This wire will eventually connect to the FeatherWing Doubler.
Heat Shrink

Apply the heat shrink to the four power and four ground solder connections.
Laser Switch

This switch lets you have control over whether or not the lasers are turned on, which can be handy if you're doing testing.

Mount the switch in the acrylic sheet using the mounting hole.

Solder the wire from the lasers' power connection to the middle connection on the switch.
Cut, splice and tin a piece of wire that is approximately 89 mm in length.

Solder the 89 mm long piece of wire to the top connection on the switch. The other end of the wire will eventually connect to the FeatherWing Doubler's 3.3V signal.

FeatherWing Doubler

Headers

Solder pin headers to the Feather M4 Express and the Music Maker FeatherWing. Additionally, solder the MIDI jumper closed on the back of the Music Maker FeatherWing. This puts the Music Maker FeatherWing into MIDI mode as described in the product guide ( ).
Solder socket headers to the FeatherWing Doubler. You can use the Feather M4 Express and Music Maker FeatherWing as a soldering jig to keep the headers aligned.
Laser Power and Ground

Place the FeatherWing Doubler on the 3D printed acrylic support. It will eventually be mounted there.

Solder the lasers' ground and power from the switch to the FeatherWing Doubler.

TCA9548A Connections

Cut, splice and tin two pieces of wire that are approximately 210 mm in length.

Solder one wire to 3.3V and one wire to ground on the FeatherWing Doubler.
Cut, splice and tin two pieces of wire that are approximately 191 mm in length.

Solder one wire to SDA on the FeatherWing Doubler and one wire to SCL on the FeatherWing Doubler.

Solder these wires to the TCA9548A:
- Doubler 3.3V to VIN
- Doubler GND to GND
- Doubler SCL to SCL
- Doubler SDA to SDA
STEMMA Power and Ground

Cut, splice and tin two pieces of wire that are approximately 165 mm in length.

Slip a piece of heat shrink onto the STEMMA ground and power wires. Solder one of the 165 mm long pieces of wire to STEMMA ground. Solder the second 165 mm piece of wire to STEMMA power.

Solder the STEMMA power wires to the FeatherWing Doubler.

STEMMA ground to GND
STEMMA power to 3.3V
Apply the heat shrink to the STEMMA power wires.

Mount the FeatherWing Doubler and TCA9548A

Mount the FeatherWing Doubler to the 3D printed stand using M2.5 screws and stand-offs.
Mount the TCA9548A to the center mounting holes on the acrylic using M2.5 screws and stand-offs.

And that completes the assembly for the laser harp!

**Usage**
Power up the laser harp by plugging a USB cable into the Feather M4 Express. Plug a 1/8" stereo audio cable into the Music Maker FeatherWing's headphone jack. Plug the other end of the audio cable into an amplifier or speaker.

Flip on the lasers and pluck away on the laser harp! You should hear sounds coming out of your connected speaker every time you pass your hand over a time of flight sensor.

You can adjust the CircuitPython code to change the notes, MIDI instrument and maximum height for the time of flight sensors.
Since the Music Maker FeatherWing uses 1/8" audio out, the laser harp can be added to tabletop synth setups easily.

Do not look directly into the lasers! They're low risk, but they could give you an eye ache if you stare right at em!

You will need to use a fog machine to have the actual laser beams be visible - they're decorative and only help you know where to put your hands so feel free to swap out different laser colors or designs if desired.