Laser Harp info

Helpful links:

1) https://learn.adafruit.com/adafruit-feather-m4-express-atsamd51/overview
2) https://nickfever.com/music/midi-cc-list
3) https://learn.adafruit.com/adafruit-music-maker-featherwing
4) https://learn.adafruit.com/midi-laser-harp-time-of-flight-sensors/overview

Important Warnings: DO NOT TOUCH the TOF sensors. They are very sensitive, and a touch from your finger can mess the sensor up temporarily or break them. The blue and yellow wires are not secured super well in their manufactured connectors, so be careful while handling. There is a reset button on each of the Adafruit circuit boards, and if this is pressed, circuitpython will have to be reinstalled on the boards, and your code WILL be lost. I have saved a copy on my computer, and I advise you do the same as you edit. It is unlikely, since the button must be pressed twice within a given amount of time, but it can be accidentally pressed. Avoid touching EITHER button, since they both will wipe the board.

For assembly, see link 4. Power is sent from TOF sensor 0 through TOF 7. SDA and SCL (blue and yellow wires) information is sent from each sensor to the blue accumulator board. These are condensed into one stream of data which is sent to the doubler, the larger board underneath the two processing boards. The laser switch, the toggle switch embedded in the 3-D printed manifold, is between the 3.3V pin on the M4 express and the lasers positive terminals. It is useful during setup, but it matters less during actual use. The toggle switch connected only to the doubler is used to toggle between modes. If the switch is on, it plays a modulating mode. When your hand is brought back and forth, you will increase and decrease the volume of the note. When the switch is off, a two-step mode is turned on. When you have your hand low, it'll play one note, while if your hand is higher, it will play the same note one octave higher. The lasers are not responsible for anything in this setup besides the visual.

These boards are designed to be easy to program, and while it is true, they have their quirks. They run on Circutpython, a division of python which is designed to run Adafruit's circuit boards. Some of the usual python modules will no longer be accessible, so sometimes you must do things the 'brute force' way with your code.

As seen in the code, this process works by cycling through each TOF sensor, checking to see how far the distance they are reading is, and consequently playing a note, or turning a note off. The M4 express (link 1) has all the useful code on it, while the music maker featherwing (link 3) synthesizes the notes. For the featherwing to process the midi notes, the midi jumper on the underside MUST be soldered. To access the code, download the MU python editor. For code to compile and run, simply save the sketch you're working on, and it will automatically play, so long as the name of your file is code.py. The M4 express is designed to run only on files with that name. If you do some digging in link 1, you will see there are 3 other names, but the process is still the same. The board will only run one file, and it'll run the first file it finds with the name code.py. (If you have other python apps on your PC, they will not work if you have a file name of code.py on your PC, heads up) In the code, more than once you will see an instance of "midi.send(ControlChange…)". These control changes can be used to alter the notes, and that is how I modulate the notes while the mode switch is on. More info in link 2.

Warning - the code downloaded off the Adafruit laser harp tutorial isn't perfect, and in fact, the sensors give rather noisy data. To produce sound, I had to alter the code, but it would chirp from all different notes. Therefore, most of my time coding was spent finding a way to successfully clean up the data. While it isn't perfect yet, it is pretty good.

Common code errors: **No I2C address at 0x29**… This means that one of the sensors is not communicating. Find the line in the code, comment out the sensor, and remove the sensor name from the array of sensors directly below. Scroll down to the for loops and adjust the range to whatever quantity of sensors is still ranging. **No SDA/SCL pull up**… unplug the board, wait, and replug. If it still doesn't work, if you've removed the boards from the doubler and reinstalled them, you may have switched them. Switch them back and try again. **Code stops running…** No matter what failed, if it was a one-off, you can unplug the board, and re-plug to restart the code.

Areas for improvement: 1) Lessening the 'lag' between when your hand interrupts the laser, and when the note is played. This delay is largely due to the debouncer. If there was another, faster way to debounce the data, that'd be great, as right now I'm using both the values in a list, and the std deviation from that list to ensure the values that we want are playing. 2) While in two-note mode, the notes will not transfer between high and low if your hand moves from one to the other while the note is 'on'. I tried this but was unable to make the switchover happen.