MIDI Controlled Robot Lyre with CircuitPython

Created by Liz Clark


Last updated on 2023-11-13 11:20:44 AM EST
# Table of Contents

**Overview**  
- A Note on Robotic Instruments  
- Parts  

**Circuit Diagram**  
- Servos  

**3D Printing**  

**Install CircuitPython**  
- CircuitPython Quickstart  
- Safe Mode  
- Flash Resetting UF2  

**Code the Robot Lyre**  
- Upload the Code and Libraries to the Feather RP2040  
- How the CircuitPython Code Works  
- Servos Over I2C  
- Arrays of Servos and Notes  
- Angles  
- The Loop  

**Wiring**  

**Assembly**  
- Cutting Acrylic and Aluminum  

**Prep and Attach the Servo Horns**  

**Mount the Servos**  

**Mount the Electronics**  

**Attach the Servos to the Acrylic**  

**Acrylic Risers**  

**Attach the Aluminum Extrusion to the Acrylic**  

**Plug in the Servos**  

**Hot Glue!**  

**Enclose the Servos**  

**Use**  
- Using with a DAW  
- Using with a MIDI Controller
What they don't tell you about starting a band is that the hardest part is organizing the people. So, why not skip that step and build a bunch of MIDI controlled robots to play music with?

In this project, you'll mount servo motors over the strings of a lyre. The servos will pluck those strings when they receive MIDI NoteOn messages via a Feather RP2040 running CircuitPython. You can control the lyre with an eternal MIDI controller or by sending MIDI out from your favorite DAW program.

The entire build sits on a plate of acrylic with some 3D printed parts. The servos have 3D printed horn extenders to pleasantly pluck the strings.
The servos are driven by the PCA9685 servo driver breakout. The driver board has an external 5V power input and connects to the Feather RP2040 over I2C.

A Note on Robotic Instruments

All acoustic instruments are slightly different and, as a result, if you build this project, you may need to do some fine tuning to the mechanical aspects of the design for best results. This lyre (https://www.adafruit.com/product/4884) was used for the build because it seems to be a common shape for budget lyres and is from a known budget music brand (Donner).

The parts that were the most finicky to dial in were the servo horns and the acrylic pegs that lift the acrylic base. It's important to achieve a proper balance between these two measurements for the best timbre from the lyre, with less than a millimeter often making a difference. All of the design files are included and should work as-is, but the source files are also available if you find you need to adjust things.

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
Adafruit 16-Channel 12-bit PWM/Servo Driver - I2C interface
You want to make a cool robot, maybe a hexapod walker, or maybe just a piece of art with a lot of moving parts. Or maybe you want to drive a lot of LEDs with precise PWM output. Then...
https://www.adafruit.com/product/815

Micro Servo - MG90D High Torque Metal Gear
Add more power to your robot with this metal-geared MG90D servo. The tiny little servo can rotate approximately 90 degrees (45 in each direction) and works just like the standard...
https://www.adafruit.com/product/1143

5V 10A switching power supply
This is a beefy switching supply, for when you need a lot of power! It can supply 5V DC up to 10 Amps, running from 110V or 220V power (the plug it comes with is for US/Canada/Japan...
https://www.adafruit.com/product/658

Slotted Aluminum Extrusion - 20mm x 20mm - 610mm long
Oh boy, now you have no excuse for not building that robot! Nothing is easier or faster than building with extrusion and we've added 20mm square aluminum extrusion and components...
https://www.adafruit.com/product/1221
<table>
<thead>
<tr>
<th>Item</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x STEMMA QT cable</td>
<td><a href="https://www.adafruit.com/product/4399">https://www.adafruit.com/product/4399</a></td>
</tr>
<tr>
<td>50 mm</td>
<td></td>
</tr>
<tr>
<td>1 x USB C to USB A Cable</td>
<td><a href="https://www.adafruit.com/product/5044">https://www.adafruit.com/product/5044</a></td>
</tr>
<tr>
<td>2 meters long</td>
<td></td>
</tr>
<tr>
<td>1 x Pack of M4 screws</td>
<td><a href="https://www.adafruit.com/product/1159">https://www.adafruit.com/product/1159</a></td>
</tr>
<tr>
<td>10 mm long</td>
<td></td>
</tr>
<tr>
<td>1 x Pack of M4 T-Nuts</td>
<td><a href="https://www.adafruit.com/product/1157">https://www.adafruit.com/product/1157</a></td>
</tr>
<tr>
<td>for 20x20 extrusion</td>
<td></td>
</tr>
<tr>
<td>1 x M2 screws</td>
<td><a href="https://www.amazon.com/dp/B07W5HBRMP">https://www.amazon.com/dp/B07W5HBRMP</a></td>
</tr>
<tr>
<td>M2 hardware</td>
<td></td>
</tr>
<tr>
<td>1 x 16 String Lyre Harp</td>
<td><a href="https://www.amazon.com/gp/product/B08L3FJZD7">https://www.amazon.com/gp/product/B08L3FJZD7</a></td>
</tr>
<tr>
<td>Donner 16 string lyre harp</td>
<td></td>
</tr>
</tbody>
</table>

### Circuit Diagram

- Feather 3.3V to PCA9685 VCC (red wire)
- Feather GND to PCA9685 GND (black wire)
- Feather SCL to PCA9685 SCL (yellow wire)
- Feather SDA to PCA9685 SDA (blue wire)

Servos

The servos have header plugs that can plug into the headers on the PCA9685. You will plug in a servo to each of the 16 (0-15) inputs on the PCA9685. Only one servo is shown in the Fritzing diagram to keep the diagram uncluttered.

3D Printing

The project may be assembled with a few 3D printed parts, described below. The parts can print with no supports. You'll need to print multiples of some parts:

Servo Rack

- 1x servoRack_base.stl
- 1x servoRack_top.stl
- 1x servoRack_topBrace.stl

Aluminum Extrusion

- 3x extrusionMount.stl
• 1x electronicsMount.stl

Acrylic - 3 mm thick

• 1x acrylicBrace.stl
• 4x acrylicPegs.stl

Servo Horns

• 16x servoHorn.stl

The STL files can be downloaded directly here or from Printables. Also included is a .DXF file for the acrylic base. More details on that file can be found on the Assembly page() in this guide.

Printables Download

finalRobotLyreCADparts.zip

Servo Rack
Three parts hold and mount the servos so that they sit on a curve directly above the lyre strings.
Aluminum Extrusion Mounts
The Feather and servo driver are mounted to a piece of aluminum extrusion with a 3D printed plate. The piece of extrusion slides onto three mounts that are secured to an acrylic plate. The extrusion also counter balances the weight of the servos.

Acrylic Pegs
Four pegs are mounted to the acrylic plate to raise it to the correct height for the servo motors.

Servo Picks
The servo horns are used to pluck the lyre strings with these 3D printed picks. You'll need to print one for each servo.

Install CircuitPython
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.
CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

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

Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.
A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.

You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

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

Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-
only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.

Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.

CircuitPython is in safe mode because you pressed the reset button during boot. Press again to exit safe mode.

Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.
Flash Resetting UF2

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

Code the Robot Lyre

Once you've finished setting up your Feather RP2040 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
import time
import board
from adafruit_motor import servo
from adafruit_pca9685 import PCA9685
import usb_midi
import adafruit_midi
from adafruit_midi.note_on          import NoteOn

# MIDI input setup
midi = adafruit_midi.MIDI(midi_in=usb_midi.ports[0], in_channel=0)

# i2c PCA9685 setup
i2c = board.STEMMA_I2C()
pca = PCA9685(i2c)
pca.frequency = 50

# create the servo objects and add them to the servos array
servos = [servo.Servo(pca.channels[i]) for i in range(16)]

# array of midi notes, high to low
midi_notes = [83, 81, 79, 77, 76, 74, 72, 71, 69, 67, 65, 64, 62, 60, 59, 57]
angle0 = 20
angle1 = 70

# set servos to the same angle on boot
# easier to adjust angles of the horns if needed
print("setting servos")
for i in range(16):
    s = servos[i]
    s.angle = angle1
    time.sleep(0.05)
print("servos set")
```
while True:
    # msg holds MIDI messages
    msg = midi.receive()

    for i in range(16):
        # iterate through servos array & midi notes array
        servo = servos[i]
        note_played = midi_notes[i]

        # if a noteon msg comes in that matches a note in the midi notes array..
        if isinstance(msg, NoteOn) and msg.note == note_played:
            # print(servo)
            # print(note_played)
            # servo moves
            # angle alternates between angle0 and angle1
            if servo.angle <= angle0:
                servo.angle = angle1
            else:
                servo.angle = angle0
            # print(servo.angle)

Upload the Code and Libraries to the Feather RP2040

After downloading the Project Bundle, plug your Feather RP2040 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 RP2040's CIRCUITPY drive.

- lib folder
- code.py

Your Feather RP2040 CIRCUITPY drive should look like this after copying the lib folder and the code.py file:
How the CircuitPython Code Works

The code begins by creating a `midi` object over USB as a MIDI input. This means that the Feather will take in, or receive, MIDI messages from another source. The source could be your computer or an external MIDI device.

```python
# MIDI input setup
midi = adafruit_midi.MIDI(midi_in=usb_midi.ports[0], in_channel=0)
```

Servos Over I²C

The STEMMA QT port on the Feather is setup as the default `i2c` object. The PCA9685 servo driver is instantiated over I²C. Then, the servo objects are created and added to the `servos` array with a Python `list comprehension`.

```python
# i2c PCA9685 setup
i2c = board.STEMMA_I2C()
pca = PCA9685(i2c)
pca.frequency = 50
servos = [servo.Servo(pca.channels[i]) for i in range(16)]
```

Arrays of Servos and Notes

An array is created for the MIDI note numbers. This array will correspond with the `servos` array. Both will be iterated through in the loop. If you want to change which MIDI notes are assigned to each servo, you can edit the `midi_notes` array. By default, it lists a C major scale.

```python
# array of midi notes, high to low
midi_notes = [83, 81, 79, 77, 76, 74, 72, 71, 69, 67, 65, 64, 62, 60, 59, 57]
```

Angles

The servos will move back and forth between two set angles. These angles are defined with `angle0` and `angle1`. Before the loop, all of the servos are set to `angle1`. This makes it easier to adjust the servo horns if needed.

```python
angle0 = 20
angle1 = 70

# set servos to the same angle on boot
# easier to adjust angles of the horns if needed
print("setting servos")
```
The Loop

In the loop, the Feather listens for incoming MIDI messages with `midi.receive()`. If a `NoteOn` message comes in containing a note number that is defined in the `midi_notes` array, then the corresponding servo will move to either `angle0` or `angle1` depending on the current angle of the servo.

```python
while True:
    # msg holds MIDI messages
    msg = midi.receive()

    for i in range(16):
        # iterate through servos array & midi notes array
        servo = servos[i]
        note_played = midi_notes[i]

        # if a noteon msg comes in that matches a note in the midi notes array..
        if isinstance(msg, NoteOn) and msg.note == note_played:
            # servo moves
            # angle alternates between angle0 and angle1
            if servo.angle <= angle0:
                servo.angle = angle1
            else:
                servo.angle = angle0
```
The PCA9685 requires a little bit of soldering to prepare it for the robot lyre. You'll solder a STEMMA QT cable, 4x3 headers and a terminal block to the PCA9685. You'll also prepare the DC power input cable.

Cut and splice one end of a STEMMA QT cable. Then, solder the wires to the PCA9685 inputs:

- STEMMA red wire to VCC
- STEMMA black wire to GND
- STEMMA blue wire to SDA
- STEMMA yellow wire to SCL

This will allow you to plug the PCA9685 into the Feather STEMMA QT port.
Solder the four sets of 4x3 headers to the PCA9685. These headers will be used to plug the servo motors into the PCA9685. You can use a breadboard to secure the headers while you solder.

Solder the included terminal block to the power input on the PCA9685.

Cut off the alligator clips on the DC to alligator clip cable. Then, splice each of those wires. These wires can be inserted into the terminal block to power the PCA9685 with a DC power supply.
Assembly

Cutting Acrylic and Aluminum

The robot lyre is made of multiple parts, some of them moving, but the core of the build rests (literally) on a piece of acrylic and a piece of aluminum extrusion.

There are many ways to get this acrylic shape with the various mounting holes. You could use a CNC, laser cutter or you could cut it using your preferred power tool. The DXF file is available on Printables or directly below.

finalRobotLyreCADparts.zip
Printables Download
The piece of aluminum extrusion serves two purposes: to mount the electronics and to counter balance the servo motors. You'll want to cut your piece of extrusion to 13.5 inches (34.29 cm). Aluminum can be cut easily with a miter saw.

Prep and Attach the Servo Horns

The lyre strings are plucked by 3D printed picks that are attached to the stock servo horn with an M2 screw.
Use a pin vise to tap the third hole from the bottom of the horn so that it will accept an M2 screw.

Slot the pick over the horn and secure it with an M2 screw.

Attach the horn pick to the servo with its point facing away from the purple label.

Repeat this process for all 16 servos.
Mount the Servos

The base of the servo cage has staggered mounting holes for each servo motor. You'll attach the servos to the cage with an M2 screw.

Begin by mounting the first servo on the left. This servo is located in the back row of the servo cage. Attach it with an M2 screw.
Next, mount the second servo in the next mounting hole on the servo cage with an M2 screw. This is the first servo in the front row of the cage.

Continue mounting the servos until all 16 are mounted on the cage.

Mount the Electronics

Attach M2.5 nuts and stand-offs to the board mounting holes on the electronics mount. Then, attach three M4 screws with t-nuts in the three center holes.
Slide the mount onto the aluminum extrusion using the t-nuts. Secure the mount by tightening the M4 screws.

Attach the Feather and servo driver to the M2.5 stand-offs with M2.5 screws. Plug the servo driver STEMMA cable into the Feather STEMMA QT port.

Attach the Servos to the Acrylic

The servos are mounted to the edge of the acrylic using M3 screws and nuts.
Attach the side of the servo cage to the acrylic using M3 screws and nuts.

Attach the opposite side with M3 screws and nuts.

Complete the attachment by mounting the nine mounting holes on the servo cage to the acrylic with M3 screws and nuts.
A 3D printed brace is attached to the center of the acrylic to try and prevent bowing. The riser pieces that raise the acrylic above the lyre strings mount to the corners of the brace.

Attach the brace to the three center mounting holes with M3 screws and nuts. You'll notice that they are arranged in a triangle.
Attach each of the risers to the four corners of the brace with long M3 screws. The riser holes are sized so that the screws will tap the riser.

**Attach the Aluminum Extrusion to the Acrylic**

Take the three extrusion mounts and attach them to the mounting holes on the acrylic using long M3 screws. The holes of the mounts are sized so that the screws will tap them.
Insert M4 screws into the holes on the back of the extrusion mounts. Attach t-nuts to the M4 screws.

Slide the aluminum extrusion onto the mounts so that the bottom side and t-nuts slot into the extrusion. The electronics should be facing up with the DC jack facing away from the servo motors.

---

**Plug in the Servos**

Plug the servo furthest to the left into port 15 on the servo driver. Make sure that the yellow wire is plugging into the PWM pin.
Continue plugging in the servos in order from ports 15-0.

The build is now in a state where you can test it with the lyre.
Align the servos so that they are directly above the lyre strings. Once you are happy with the placement, use some masking tape to temporarily hold the apparatus in place.

Before deciding on a final placement, you may want to do a test run playing the lyre to make sure that the servos are striking the strings evenly.

Then, take some hot glue and tack the acrylic stand-offs in place.
Enclose the Servos

The final step in securing the servos is installing the top of the servo cage. Begin by press fitting the main piece behind the front row of servos.

Secure the main piece to the bottom base with an M3 screw and nut.

Attach the first servo with an M2 screw. There are mounting holes for the entire front row of servos that align with the servos' top mounting hole.
Attach the rest of the front row of servos with M2 screws to the main piece.

Press fit the back piece behind the back row of servos.

Attach the back piece to the bottom piece with M3 screws and nuts.
Secure the back row of servos to the back piece with M2 screws. The screw will thread through the back piece, servo mounting hole and the mounting hole on the main piece.

Secure the remaining servos in the back row with M2 screws.

Use
After you're finishing assembling, plug a 5V DC power supply into the power input on the PCA9685. Then, connect the Feather RP2040 via USB to either your computer or USB MIDI host. There are two ways to control the robot lyre: with a DAW or with a hardware MIDI controller.

Using with a DAW

A digital audio workstation (DAW) is any computer program that lets you work with audio on your computer. Some examples include Logic, Reason and Ableton Live.

You can set the Feather RP2040 as a MIDI output to send MIDI data to it from a DAW. This is usually done with a MIDI track or with a MIDI plugin. The Feather will usually show up as a device called CircuitPython Audio.

Using with a MIDI Controller

You can control the robot lyre directly with another hardware MIDI controller. This can be accomplished with the use of a USB MIDI host. A USB MIDI host lets two USB MIDI devices talk to each other directly, with one sending MIDI messages and the other receiving those MIDI messages.

There are hardware USB MIDI host solutions (including DIY ones, like this Raspberry Pi project) and software solutions like MIDiberry on Windows and MIDI Studio on macOS.
After configuring your MIDI controller as the MIDI input and the robot lyre as the MIDI output on your MIDI host, you can play melodies and sequences with your controller to directly pluck the robot lyre.