## Table of Contents

**Overview** 3
- Parts

**Assemble the Controller** 6
- I2C Address Jumpers
- Case Build

**Code the Controller** 7
- Text Editor
- CircuitPython Installation
- Project Files
- Libraries and code.py

**Rack Setup** 11
- VCV Rack and Plugins Required
- MIDI Device Select
- Use the Controller with Rack
- Acknowledgements
Drum patterns and colored lights! Use your grid controller with your software synth and keep the LEDs in sync with the GUI. The NeoTrellis MIDI Feedback Controller sends MIDI messages containing note information in order to program a trigger sequencer -- but here's the cool part: when the controller receives MIDI note messages back from the software it interprets them as NeoPixel commands!

Inspired by the excellent, Arduino-based design by PatchworkBoy (), this simpler, CircuitPython-based approach will get you started quickly controlling four 16-step drum trigger sequences in a VCV Rack. You can then modify it to fit your needs.
Parts

4 x Adafruit NeoTrellis RGB Driver PCB for 4x4 Keypad  
https://www.adafruit.com/product/3954

4 x Silicone Elastomer 4x4 Button Keypad for 3mm LEDs  
https://www.adafruit.com/product/1611

Adafruit 8x8 Trellis Feather M4 Acrylic Enclosure + Hardware Kit  
A super-specifically-laser-cut enclosure that turns your 8x8 'Trellis into a handheld light/music maker! Perfect for your next cool interface, MIDI instrument,...  
https://www.adafruit.com/product/4372
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

4-pin JST PH to JST SH Cable - STEMMA to QT / Qwiic
Are you a maker in the midst of a STEMMA dilemma? This 200mm long 4-wire...
https://www.adafruit.com/product/4424

1 x USB A to USB C Cable
3'/1m
https://www.adafruit.com/product/4474

Alternative Controller

Adafruit 8x8 NeoTrellis Feather M4 Kit Pack
We've upgraded our popular UNTZtrument with a total make-over, and it's been reborn as the NeoTrellis 8x8 Kit pack. This open-source 8x8 Grid Controller Kit with...
https://www.adafruit.com/product/1929
Assemble the Controller

Follow this [page](#) for the assembly of the NeoTrellis tiles.

NOTE: there is no need to solder wires to pads on the board if you use the Feather RP2040 as you can use a [STEMMA to STEMMA QT cable](#) instead.
I2C Address Jumpers
Solder the I2C address jumpers as shown here. This gives you addresses 0x2E, 0x2F, 0x30, and 0x31.

Case Build

This guide page shows how to assemble the acrylic case for your NeoTrellis 8x8.

NOTE: You can skip the steps involving switches and battery as this will be used as a plugged-in MIDI controller over USB.

Code the Controller

Text Editor

Adafruit recommends using the Mu editor for using your CircuitPython code with the Feather. You can get more info in this guide.
Alternatively, you can use any text editor that saves text files.

## CircuitPython Installation

First make sure you are running the latest version of Adafruit CircuitPython for your board.

## Project Files

Download the Project Bundle linked below and copy them onto your CIRCUITPY drive.

### Libraries and code.py

You'll need to download the .zip file in the GitHub link below and decompress the file to get all of the library files.

Copy the lib directory as well as the code.py file to your Feather’s CIRCUITPY drive.
# holds whether pad is currently lit or not, when it was pressed before being ack'd
# tuple of (pad lit?, pad_press_time_or_zero_if_ackd)
pad_states = [(False,0)] * 64

midi_usb = adafruit_midi.MIDI( midi_in=usb_midi.ports[0],
                           midi_out=usb_midi.ports[1] )

i2c = busio.I2C(board.SCL, board.SDA)
trelli = [  # adjust these to match your jumper settings if needed
    [NeoTrellis(i2c, False, addr=0x2E), NeoTrellis(i2c, False, addr=0x2F)],
    [NeoTrellis(i2c, False, addr=0x30), NeoTrellis(i2c, False, addr=0x31)]
]
trellis = MultiTrellis(trelli)

OFF = 0x000000
RED = 0x100000
YELLOW = 0x100c00
GREEN = 0x000c00
CYAN = 0x000303
BLUE = 0x000010
PURPLE = 0x130010

colors = [OFF, RED, YELLOW, GREEN, CYAN, BLUE, PURPLE]

color_table = [  # you can make custom color sections for clarity
    1, 1, 1, 1, 5, 5, 5, 5,
    1, 1, 1, 1, 5, 5, 5, 5,
    1, 1, 1, 1, 5, 5, 5, 5,
    1, 1, 1, 1, 5, 5, 5, 5,
    4, 4, 4, 4, 6, 6, 6, 6,
    4, 4, 4, 4, 6, 6, 6, 6,
    4, 4, 4, 4, 6, 6, 6, 6,
    4, 4, 4, 4, 6, 6, 6, 6
]

# convert xy (0-7,0-7) to 0-63
def xy_to_pos(x,y):
    return x+(y*8)

# convert 0-63 to x,y
def pos_to_xy(pos):
    return (pos%8, pos//8)

# callback when pads are pressed
def handle_pad(x, y, edge):
    pos = xy_to_pos(x,y)
    note_val = pos + note_base
    if edge == NeoTrellis.EDGE_RISING:
        (pad_on, pad_time) = pad_states[pos]  # get pad state & press time
        pad_time = time.monotonic() - pad_time  # toggle state
        pad_states[pos] = (pad_on, pad_time)  # and save it w/ new press time
        if pad_on:
            noteon = NoteOn(note_val, note_vel, channel=pad_midi_channel)
            midi_usb.send(noteon)
        else:
            noteoff = NoteOff(note_val, note_vel, channel=pad_midi_channel)
            midi_usb.send(noteoff)

# called periodically in main loop to receive MIDI msgs
# saves to pad_state with LED on/off and 0 to indicate pad press acknowledgement
def midi_receive():
    msg_in = midi_usb.receive()
    if msg_in is None:
        return
    msg_in.channel == led_midi_channel:
Next, you set up VCV Rack to work with the NeoTrellis 8x8 MIDI Feedback Controller.
Rack Setup

VCV Rack is a virtual modular synthesizer that is free and open source. It can be used to patch together modules for sound design and music creation all by itself, and it can also be used with MIDI controllers.

You'll use it with the excellent Stoermelder MIDI-CAT and Trowasoft trigSeq modules so that the NeoTrellis 8x8 can both send and receive MIDI messages. This is a simplified version of the excellent system created by PatchworkBoy [here](#).

MIDI messages sent from the NeoTrellis to Rack will be used to change the sequencer trigger patterns of three drums and a bass synth.

In the other direction, MIDI messages from Rack to the NeoTrellis will be used to keep the NeoPixel lighting in sync with the on-screen GUI.

VCV Rack and Plugins Required

- [VCVRack2 + Fundamentals](#)
- Audible Instruments (Mutable Instruments) [Plaits](#)
- Stoermelder PACK-ONE Dev Builds
- Trowasoft-VCV

Download and install [VCV Rack 2](#) for your operating system. This will include the Fundamental modules as well.
Then, from the Rack Library, install **Audible Instruments Macro Oscillator 2** (software version of the esteemed Mutable Instruments Plaits module).

Download **Stoermelder PACK-ONE**. Then, put the .vcvplugin file in the /Documents/Rack/plugins directory.

Finally, download the **Trowasoft-VCV** dev build. Extract the zip, then place the folder in the /Documents/Rack/plugins directory.

MIDI Device Select
When you first open the VCV Rack patch file you may find that the MIDI-CAT module has (No device) listed for both the outgoing and incoming MIDI devices.

Click these each and set them to the name of the board in your NeoTrellis 8x8 -- either Feather RP2040 or Feather M4, depending.

Use the Controller with Rack

Once the controller is selected in the MIDI-CAT module, hit Reset and then Play in the Pulses module to start the trigger sequencers.
Now, you can add triggers to any of the four drum sequencers with the NeoTrellis and the VCV Rack GUI will match. Same goes for the other direction! Click any triggers in the Trowasoft trigSeq modules and the NeoTrellis will adjust the NeoPixels to match.

Acknowledgements

This project would not have been possible without the excellent work done by PatchworkBoy in creating the TrowasoftControl code, workflow, and example patch seen here.