CircuitPython

TrellisM4 $_{e}xtended Library Documentation$

Release 1.0

Arofarn

Jun 28, 2021

CONTENTS

Use Adafruit TrellisM4 Express board as 2 Neotrellis board. You can you use this to extend TrellisM4 with Neotrellis (seesaw) boards.

CHAPTER

ONE

DEPENDENCIES

This driver depends on:

- Adafruit CircuitPython
- Bus Device
- Adafruit Neopixel driver
- Adafruit Seesaw driver
- Adafruit Matrix Keypad library

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER

TWO

USAGE EXAMPLE

How to solder boards together

To use Trellis as 2 Neotrellis (seesaw):

```
from neotrellism4 import NeoTrellisM4
trellis_left = NeoTrellisM4()
trellis_right = NeoTrellisM4(left_part=trellis_left)
```

To use TrellisM4 tilled with Neotrellis (seesaw):

```
from board import SCL, SDA
import busio
from adafruit_neotrellis.neotrellism4 import NeoTrellisM4
from adafruit_neotrellis.neotrellis import NeoTrellis
from adafruit_neotrellis.multitrellis import MultiTrellis
I2C = busio.I2C(SCL, SDA)
trellim4_left = NeoTrellisM4()
trellim4_right = NeoTrellisM4(left_part=trellim4_left)
trelli = [
    [trellim4_left, trellim4_right],
    [NeoTrellis(I2C, False, addr=0x2F), NeoTrellis(I2C, False, addr=0x2E)]
    ]
trellis = MultiTrellis(trelli)
```

CHAPTER

THREE

CONTRIBUTING

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER

FOUR

DOCUMENTATION

For information on building library documentation, please check out this guide.

CHAPTER

FIVE

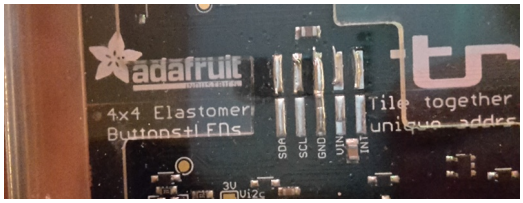
TABLE OF CONTENTS

5.1 Soldering TrellisM4 with 2 Neotrellis

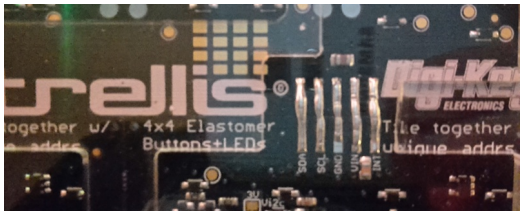
Here is how to do tilled one trellis m4 and 2 neotrellis (seesaw) into a square :

1. solder the 2 neotrellis together as described in the tutorial (including pad for I2C addresses as needed).
2. solder the tilled neotrellis to the trellis :

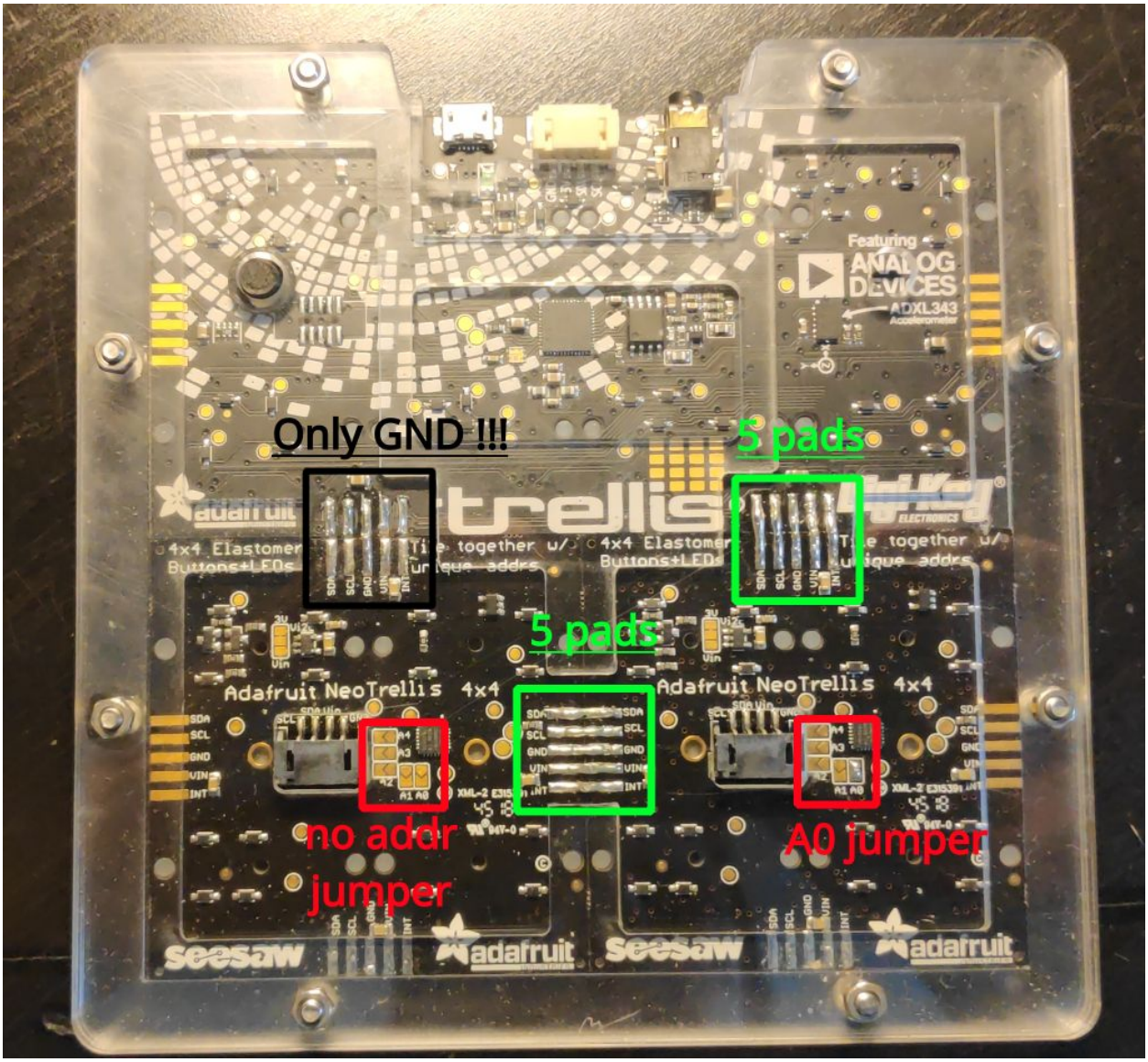
- every texts on the silkscreen in the same direction = the LED N°1 of the neotrellis and the USB port up
- among the left pads on the bottom edge of the trellis m4 : solder ONLY the center one (GND)



- solder every 5 pads on the right



Here is a picture of the full tilled board:



5.2 Simple test

Ensure your device works with this simple test.

Listing 1: examples/trellism4_extended_simpletest.py

```
# SPDX-FileCopyrightText: 2017 Scott Shawcroft, written for Adafruit Industries
# SPDX-FileCopyrightText: Copyright (c) 2021 Arofarn
#
# SPDX-License-Identifier: MIT

import time

from board import SCL, SDA
import busio
```

(continues on next page)

(continued from previous page)

```
from adafruit_neotrellis.neotrellis import NeoTrellis
from adafruit_neotrellis.multitrellis import MultiTrellis
from neotrellism4 import NeoTrellisM4

# Create the i2c object for the trellis
I2C = busio.I2C(SCL, SDA)

# Create the trellis. This is for a 2x2 array of TrellisM4 (first row) with
# 2 Neotrellis (second row).
#
# [ NeoM4_left | NeoM4_right ]
#  neotrellis0 | neotrellis1

trellim4_left = NeoTrellisM4()
trellim4_right = NeoTrellisM4(left_part=trellim4_left)
trelli = [
    [trellim4_left, trellim4_right],
    [NeoTrellis(I2C, False, addr=0x2F), NeoTrellis(I2C, False, addr=0x2E)],
]

trellis = MultiTrellis(trelli)

# some color definitions
OFF = (0, 0, 0)
RED = (127, 0, 0)
YELLOW = (127, 75, 0)
GREEN = (0, 127, 0)
CYAN = (0, 127, 127)
BLUE = (0, 0, 127)
PURPLE = (90, 0, 127)

# this will be called when button events are received
def blink(xcoord, ycoord, edge):
    """Turn the LED on when a rising edge is detected or
    turn the LED off when a falling edge is detected
    """
    if edge == NeoTrellis.EDGE_RISING:
        trellis.color(xcoord, ycoord, BLUE)
    elif edge == NeoTrellis.EDGE_FALLING:
        trellis.color(xcoord, ycoord, OFF)


for y in range(8):
    for x in range(8):
        # activate rising edge events on all keys
        print(x, y)
        trellis.activate_key(x, y, NeoTrellis.EDGE_RISING)
        # activate falling edge events on all keys
        trellis.activate_key(x, y, NeoTrellis.EDGE_FALLING)
        trellis.set_callback(x, y, blink)
        trellis.color(x, y, PURPLE)
        time.sleep(0.05)
```

(continues on next page)

(continued from previous page)

```
for y in range(8):
    for x in range(8):
        trellis.color(x, y, OFF)
        time.sleep(0.05)

while True:
    # the trellis can only be read every 17 millisecons or so
    trellis.sync()
    time.sleep(0.02)
```

5.3 trellism4_extended

CircuitPython library to extended Adafruit NeotrellisM4 board with two Neotrellis seesaw boards (or more !).

- Author(s): arofarn

5.3.1 Implementation Notes

Hardware:

- Adafruit NeoTrellis M4 Express: https://www.adafruit.com/product/3938
- Adafruit NeoTrellis RGB Driver PCB for 4x4 Keypad :https://www.adafruit.com/product/3954

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases
- Adafruit CircuitPython bus_devices library (from Adafruit_CircuitPython_Bundle): https://github.com/adafruit/Adafruit_CircuitPython_Bundle/releases

class trellism4_extended.**NeoTrellisM4**(*left_part=None*)

Driver for the Adafruit NeoTrellis.

Parameters **left_part** – None (default) or left part object

---

Note: if None (or ommitted) the class create a neotrellis.multitrellis-compatible object for the left half of the TrellisM4 board. Else the right part is created and the arguement should be the left part object.

Example:

```
from neotrellism4 import NeoTrellisM4
trellis_left = NeoTrellisM4()
trellis_right = NeoTrellisM4(left_part=trellis_left)
```

activate_key(*key*, *edge*, *enable=True*)

Activate or deactivate a key on the trellis

:param int key : key number from 0 to 16. :param int edge : specifies what edge to register an event on and can be NeoTrellis.EDGE_FALLING or NeoTrellis.EDGE_RISING. :param bool enable : should be set to True if the event is to be enabled, or False if the event is to be disabled.

property count
Return the pressed keys count

property interrupt_enabled
Only for compatibility with neotrellis module: Interrupts are disable on trellis M4 keypad

read_keypad(*num*)
Read data from the keypad

Parameters **num** (*int*) – The number of bytes to read

set_event(*key*, *edge*, *enable*)
Control which kinds of events are set

Parameters

- **key** (*int*) – the key number
- **edge** (*int*) – the type of event
- **enable** (*bool*) – True to enable the event, False to disable it

sync()
Read any events from the Trellis hardware and call associated callbacks.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

t

INDEX