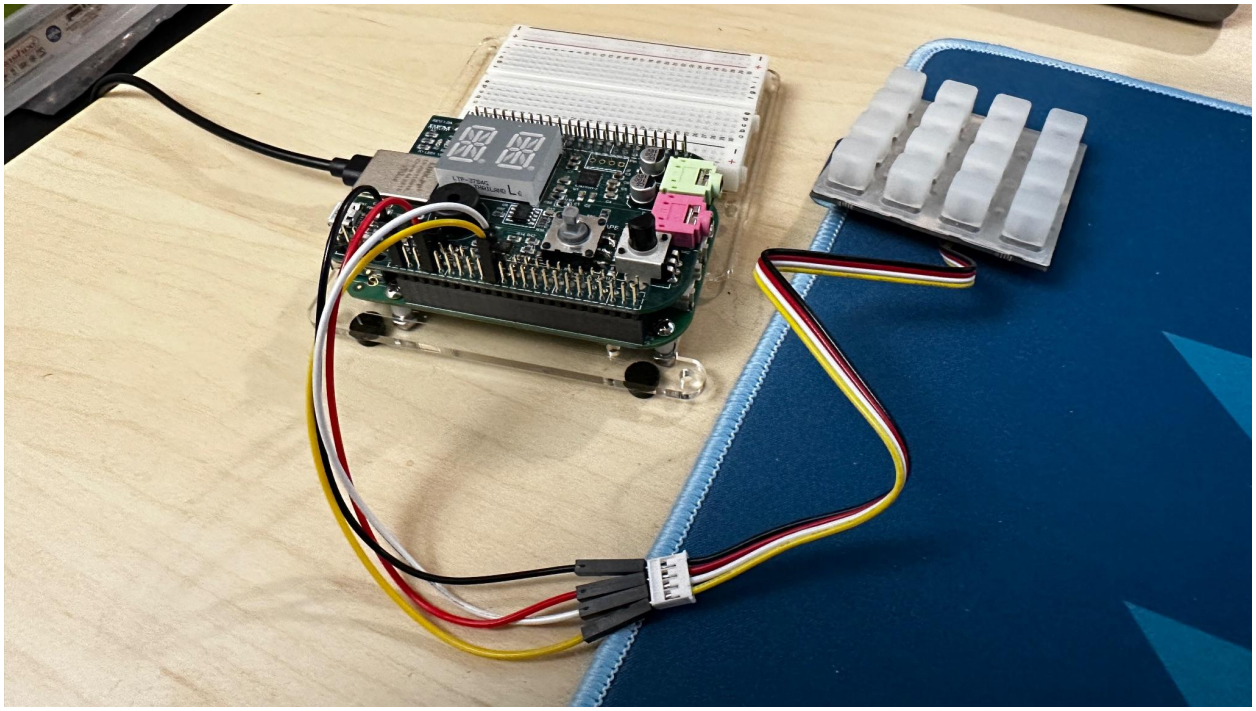
How to connect and control Adafruit's NeoTrellis 4x4 Button with integrated LED matrix (with silicone keypad)



By "Pattern Matrix" (Spring 2023)

Written on April 10, 2023

Compatible devices: **BeagleBone Green, BeagleBone Red**

Guide has been tested on: **BeagleBone (target) Debian 11.4**, **PC OS (host) Debian 11.5**

Introduction

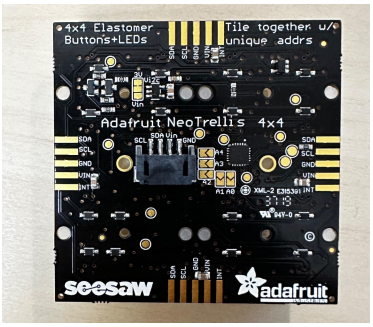
The purpose of this how-to guide is to explain the connection between the BeagleBone device and the 4x4 matrix. This guide is built based on the previous guide (Spring 2022), and the guide will be provided in the reference section.

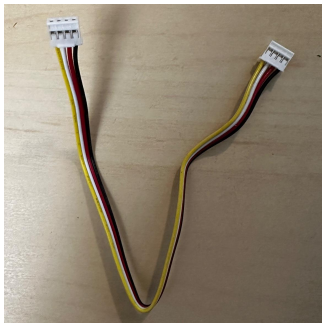
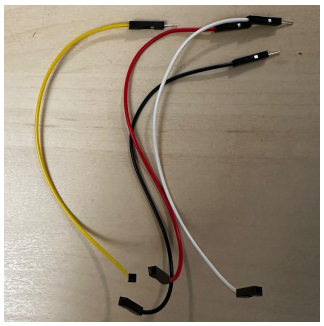
Table of Contents

1. Wire connection from 4x4 matrix to BeagleBone Green/Red

1.1 Components needed

1. Adafruit 4x4 matrix
2. 4 pin female to female connector cable
3. 4x female to male extension jumper wires



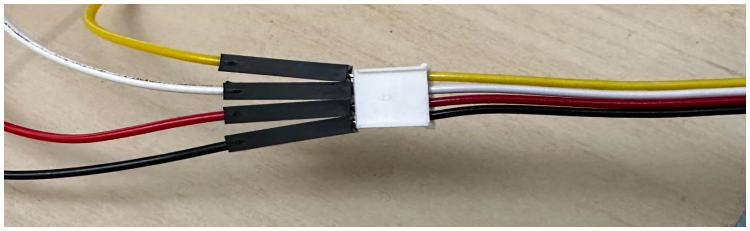



1.2 How to connect

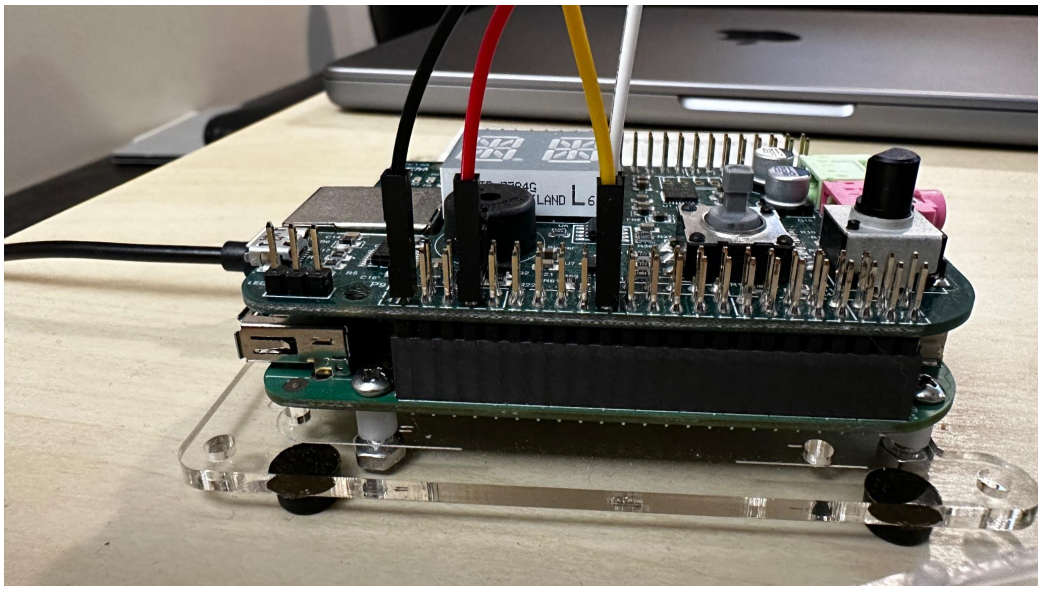
The 4x4 matrix is using 4 pins on the board, which are:

- **GND** (ground)
- **VIN** (power)
- **SDA** (data)
- **SCL** (clock)

1. Assemble the jumper wires to one side of the 4-pin cable, as shown in the picture. (Tip: connect the middle 2 cables first to ensure the metal pins go all the way down to connect properly)



2. **GND cable** (**black**) will connect to **P9_1**, **VIN cable** (**red**) will connect to **P9_7**, **SDA cable** (**yellow**) will connect to **P9_19**, and **SCL cable** (**white**) will connect to **P9_20**. In the end, the connection should be like the picture shown.
(Tip: This connection is using I2C2 bus. If wishing to use I2C1 bus instead, connect **SDA** cable to **P9_17** and **SCL** cable to **P9_18**.)



1.3 Checking if the connection is successful

In order to make sure the connection is working, we need to configure SDA and SCL pins to i2c mode first, then we can double-check it on I2C2 bus on the terminal:

(bbg)$ config-pin P9_19 i2c

(bbg)$ config-pin P9_20 i2c

(bbg)$ i2cdetect -y -r 2

```
debian@BeagleBone:~$ i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 2e --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- UU -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

(Tip: we are only looking to see if **0x2e** is connected, we can ignore the UU in the chart (those are indicating other connections))

2. How to control 4x4 matrix

2.1 support Files

There are 2 modules needed to control the 4x4 matrix:

1. seeSaw.c

In the seeSaw module, we are initializing the i2c bus so users don't have to manually configure the pins every time, and define seesaw write/read functions for the trellis module to use. You will need this module in order for the 4x4 trellis matrix to work.

2. neoTrellis.c

This will be the main module you will use to control the 4x4 trellis matrix. The interface looks like this:

```
#ifndef _TRELLIS_H_
#define _TRELLIS_H_

#include <stdbool.h>

typedef struct color
{
    unsigned char red;
    unsigned char blue;
    unsigned char green;
}color;

void NeoTrellis_LEDs_Init(void);
void NeoTrellis_LEDs_SetPixel_to_Color(unsigned int index, color color);
void NeoTrellis_LEDs_SetAllLEDs_to(color color);
void NeoTrellis_LEDs_TurnAllLEDs_off(void);
void NeoTrellis_LEDs_TurnLED_off(unsigned int index);
void NeoTrellis_LEDs_Destroy(void);
void NeoTrellis_LEDs_UpdateTrellisBuff (void);
int NeoTrellis_Keys_getPushedButtonIndex (void);

#endif
```

The functions are pretty straightforward, and they perform as the names suggest.
(Tip: make sure to call **NeoTrellis_LEDS_Destroy()** when exiting the program to ensure no memory leaks!!)

2.2 Basic control of LED matrix

To control RGB color, we can use the color struct to control the RGB value. For example, to make the color yellow, we can have the RGB value as such:

```
static color YELLOW;
YELLOW.red = 255;
YELLOW.green = 255;
YELLOW.blue = 0;
```

To set a certain pixel to a certain color, we can use the function **NeoTrellis_LEDs_Set_Pixel_to_Color(index, color)** with the arguments of the index and the color. Make sure to call **NeoTrellos_LEDs_UpdateTrellisBuff()** to let the board know the pixel color has been updated.

```
NeoTrellis_LEDs_SetPixel_to_Color(0, BLUE);
NeoTrellis_LEDs_UpdateTrellisBuff();
```

3. Troubleshooting

If you are using the previous student's support files, you might run into the issue below:

```
debian@BeagleBone:/mnt/remote/myApps$ ./neotrellis_example
???     ?
        ??????!?$?'?*?-?debian@BeagleBone:/mnt/remote/myApps$
```

To fix this issue, we will need to initialize the trellis matrix. I have fixed it in the current files provided.

4. References

Previous student guide on 4x4 matrix
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2022-student-howtos/NeoTrellis_4x4.pdf
NeoTrellis product details
https://www.adafruit.com/product/3954
I2C guide
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files/I2CGuide.pdf