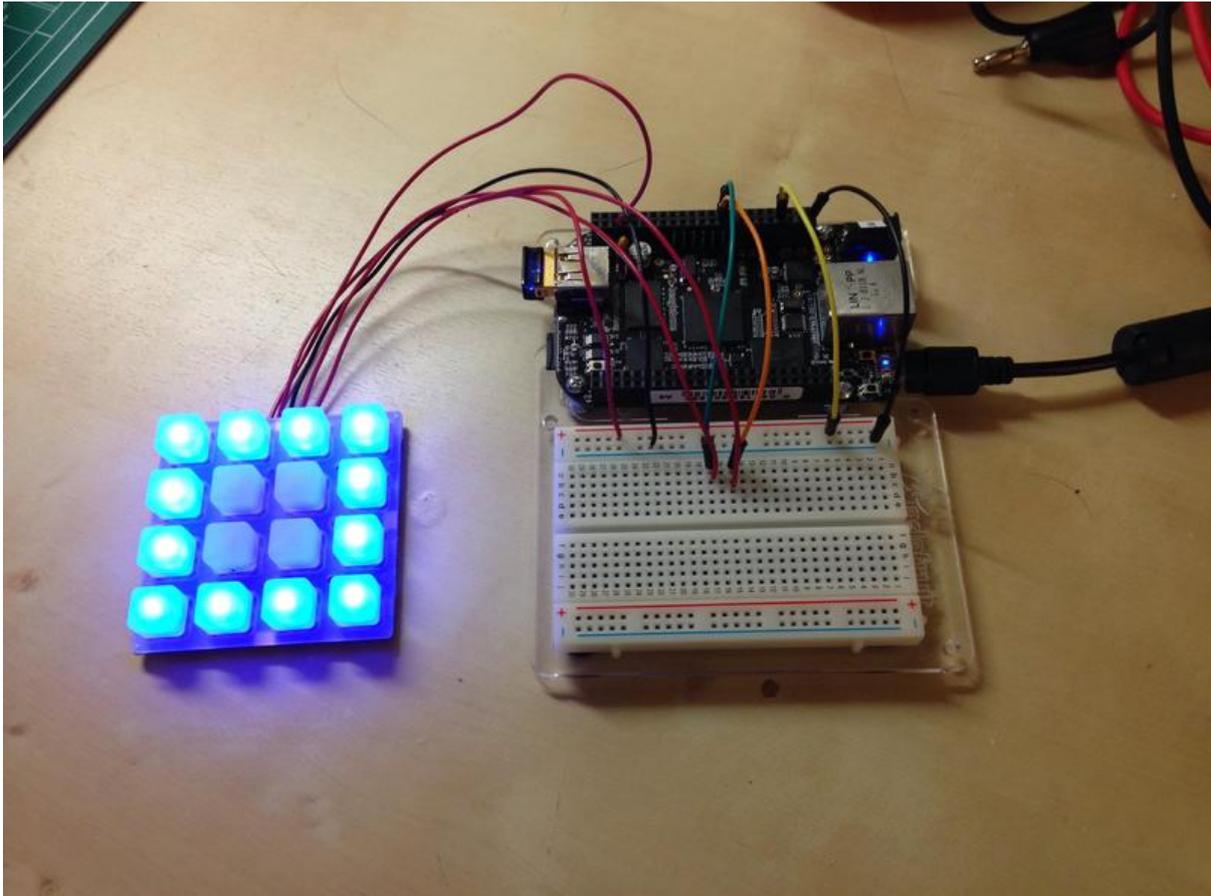




# Trellis Python Library

Created by Tony DiCola



<https://learn.adafruit.com/trellis-python-library>

Last updated on 2024-03-08 01:55:03 PM EST

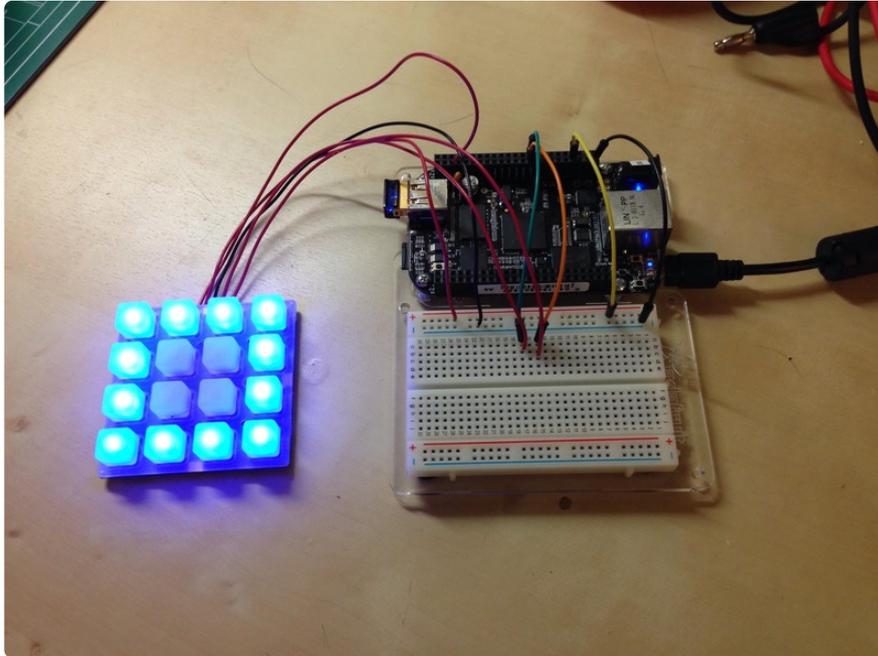
# Table of Contents

|                  |   |
|------------------|---|
| Overview         | 3 |
| Beaglebone Black | 3 |
| • Installation   |   |
| • Usage          |   |
| Raspberry Pi     | 5 |
| • Installation   |   |
| •                |   |
| • Usage          |   |
| Usage            | 6 |

---

# Overview

The examples in this guide are no longer supported. Please check out the [Introducing Adafruit Trellis guide for CircuitPython and Python usage: https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview](https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview)



[Adafruit Trellis \(http://adafru.it/1616\)](http://adafru.it/1616) is a great system for adding backlit keys to your own projects. Trellis' elastomer keypad acts as both an input and fully addressable LED display. You can even tile multiple Trellises together to form large, captivating displays.

Now there's a Python library to control Trellis from any computer with Python and I2C, like the Beaglebone Black or Raspberry Pi. Follow this guide for information on installing and using the [Trellis python library \(https://adafru.it/d92\)](https://adafru.it/d92).

---

# Beaglebone Black

The examples in this guide are no longer supported. Please check out the [Introducing Adafruit Trellis guide for CircuitPython and Python usage: https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview](https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview)

## Installation

To install the Trellis python library on the Beaglebone Black follow these steps:

- Make sure [python 2.7 \(https://adafru.it/d20\)](https://adafru.it/d20) is installed on your Beaglebone Black. If you're using the Angstrom, Ubuntu, or Debian operating system python should already be installed. However if you aren't sure, run the command 'python --version' (without quotes) at a terminal and look for an output like 'Python 2.7.3'.
- Install the Adafruit\_BBIO python library. There's a [great guide right here \(https://adafru.it/d93\)](https://adafru.it/d93) to walk through the installation.
- Download the [Trellis python \(https://adafru.it/d92\)](https://adafru.it/d92) library to the Beaglebone Black and unzip the archive. You can execute these commands to do both:

```
wget https://github.com/tdicola/Adafruit_Trellis_Python/archive/master.zip
unzip master.zip
```

- Install the library by navigating to the folder and executing setup.py:

```
cd Adafruit_Trellis_Python-master
sudo python setup.py install
```

## Usage

The Beaglebone Black by default has 2 I2C buses which means you can have up to 16 Trellises attached at once (8 per I2C bus). [This is a good blog post \(https://adafru.it/d94\)](https://adafru.it/d94) that explains where the I2C pins are located, but as a summary you can use:

- **P9 pin 17** and **18**: Channel 0 SCK and SDA
- **P9 pin 19** and **20**: Channel 1 SCK and SDA

Connect Trellis power and ground to the Beaglebone **5V power** and **ground** pins respectively (**P9 pin 7** or **8** for power, and **P9 pin 1** or **2** for ground).

If you're using the Trellis **INT** wire as a key press interrupt, connect it to any digital GPIO pin on the Beaglebone Black. The Trellis python library doesn't use the interrupt pin directly, so you'll want to monitor the pin yourself in your code.

Finally to check the Trellis is detected and confirm which I2C bus it's on, run these commands to probe each I2C bus:

```
sudo i2cdetect -y -r 0
sudo i2cdetect -y -r 1
```

Trellis should be detected at addresses starting with 70, like 70, 71, 72, etc.

---

## Raspberry Pi

The examples in this guide are no longer supported. Please check out the [Introducing Adafruit Trellis guide for CircuitPython and Python usage: https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview](https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview)

## Installation

To install the Trellis python library on the Raspberry Pi follow these steps:

- Make sure [python 2.7 \(https://adafru.it/d20\)](https://adafru.it/d20) is installed on your Raspberry Pi. If you're using the Raspbian or Occidentalis operating system python should already be installed. However if you aren't sure, run the command 'python --version' (without quotes) at a terminal and look for an output like 'Python 2.7.3'.
- If it isn't already enabled, [follow this guide \(https://adafru.it/aTl\)](https://adafru.it/aTl) for information on setting up I2C on your Raspberry Pi.
- Download the Adafruit Raspberry Pi Python code repository to your Raspberry Pi by [following this guide \(https://adafru.it/d95\)](https://adafru.it/d95).
- Download the [Trellis python \(https://adafru.it/d92\)](https://adafru.it/d92) library to the Raspberry Pi and unzip the archive. You can execute these commands to do both:

```
wget https://github.com/tdicola/Adafruit_Trellis_Python/archive/master.zip
unzip master.zip
```

- Install the library by navigating to the folder and executing setup.py:

```
cd Adafruit_Trellis_Python-master
sudo python setup.py install
```

- **Note:** For any python script you create to use Trellis, copy the `Adafruit_I2C.py` file from the **Adafruit Raspberry Pi Python code download** into the same folder as your python script. Don't skip this step--`Adafruit_I2C.py` needs to be in the same directory as your script or else the Trellis python library will fail to load!

## Usage

The Raspberry Pi has one I2C bus on it's main GPIO header. [This page \(https://adafru.it/cbG\)](https://adafru.it/cbG) has a great description of all the GPIO header pins, but as a summary you can use:

- **P1 pin 5** for SCL
- **P1 pin 3** for SDA

Connect Trellis power and ground to the Pi 5V power and ground pins respectively (**P1 pin 2** or **4** for power, and **P1 pin 6** for ground).

If you're using the Trellis **INT** wire as a key press interrupt, connect it to any digital GPIO pin on the Raspberry Pi. The Trellis python library doesn't use the interrupt pin directly, so you'll want to monitor the pin yourself in your code.

Finally to check the Trellis is detected and confirm which I2C bus it's on (different revisions of the Raspberry Pi have different I2C bus numbers), run these commands to probe each I2C bus:

```
sudo i2cdetect -y -r 0
sudo i2cdetect -y -r 1
```

Trellis should be devices detected at addresses starting with 70, like 70, 71, 72, etc.

---

## Usage

The examples in this guide are no longer supported. Please check out the [Introducing Adafruit Trellis guide for CircuitPython and Python usage: https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview](https://learn.adafruit.com/adafruit-trellis-diy-open-source-led-keypad/overview)

The Trellis python library has exactly the same API as the Arduino library so it should be familiar and straightforward to use with existing Trellis code. For an example of how to use the library, look at the **TrellisTest.py** file in the examples folder of the Trellis library download. This file is a direct port to python of the TrellisTest sketch from the Arduino library.

The only major changes in the Trellis python library are the following:

- The **Adafruit\_Trellis** and **Adafruit\_TrellisSet** class **begin()** functions take both an I2C address and bus number, instead of just an address.
- The **Adafruit\_TrellisSet** class can support any number of Trellises instead of being limited to only 8. Note that each I2C bus is still limited to only 8 Trellises, however by using multiple buses you can control any number of Trellises.

When you run your python script that uses the Trellis library, make sure to run as root using the sudo command or else you might not have access to the I2C hardware.

If you discover issues or want to contribute, please check out [the library's github page \(https://adafru.it/d92\)](https://adafru.it/d92).