

---

# **Adafruit IS31FL3731 Library Documentation**

***Release 1.0***

**Radomir Dopieralski**

**Dec 06, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	Other Examples . . . . .	14
6.3	adafruit_is31fl3731 . . . . .	16
6.3.1	Implementation Notes . . . . .	17
<b>7</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



CircuitPython driver for the IS31FL3731 charlieplex IC.

This driver supports the following hardware:

- [Adafruit 16x9 Charlieplexed PWM LED Matrix Driver - IS31FL3731](#)
- [Adafruit 15x7 CharliePlex LED Matrix Display FeatherWings](#)
- [Adafruit 16x8 CharliePlex LED Matrix Bonnets](#)



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)

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





## CHAPTER 2

---

### Installing from PyPI

---

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#). To install for current user:

```
pip3 install adafruit-circuitpython-is31fl3731
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-is31fl3731
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-is31fl3731
```



## CHAPTER 3

---

### Usage Example

---

Matrix:

```
import adafruit_is31fl3731
import board
import busio
with busio.I2C(board.SCL, board.SDA) as i2c:
    display = adafruit_is31fl3731.Matrix(i2c)
    display.fill(127)
```

Charlie Wing:

```
import adafruit_is31fl3731
import board
import busio
with busio.I2C(board.SCL, board.SDA) as i2c:
    display = adafruit_is31fl3731.CharlieWing(i2c)
    display.fill(127)

    # Turn off pixel 4,4, change its brightness and turn it back on
    display.pixel(4, 4, 0)    # Turn off.
    display.pixel(4, 4, 50)   # Low brightness (50)
    display.pixel(4, 4, 192)  # Higher brightness (192)
```



## CHAPTER 4

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



## CHAPTER 5

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).





### 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/is31fl3731\_simpletest.py

```
1 import board
2 import busio
3 import adafruit_is31fl3731
4
5 i2c = busio.I2C(board.SCL, board.SDA)
6
7 # initialize display using Feather CharlieWing LED 15 x 7
8 display = adafruit_is31fl3731.CharlieWing(i2c)
9
10 # uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
11 #display = adafruit_is31fl3731.Matrix(i2c)
12
13 # uncomment next line if you are using Adafruit 16x8 Charlieplexed Bonnet
14 #display = adafruit_is31fl3731.CharlieBonnet(i2c)
15
16 # draw a box on the display
17 # first draw the top and bottom edges
18 for x in range(display.width):
19     display.pixel(x, 0, 50)
20     display.pixel(x, display.height - 1, 50)
21 # now draw the left and right edges
22 for y in range(display.height):
23     display.pixel(0, y, 50)
24     display.pixel(display.width - 1, y, 50)
```

## 6.2 Other Examples

Listing 2: examples/is31fl3731\_blink\_example.py

```

1 import busio
2 import board
3 import adafruit_is31fl3731
4
5 i2c = busio.I2C(board.SCL, board.SDA)
6
7 # array pattern in bits; top row-> bottom row, 8 bits in each row
8 an_arrow = bytearray((0x08, 0x0c, 0xfe, 0xff, 0xfe, 0x0c, 0x08, 0x00, 0x00))
9
10 # initial display using Feather CharlieWing LED 15 x 7
11 display = adafruit_is31fl3731.CharlieWing(i2c)
12 # uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
13 #display = adafruit_is31fl3731.Matrix(i2c)
14 # uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
15 #display = adafruit_is31fl3731.CharlieBonnet(i2c)
16
17 # first load the frame with the arrows; moves the an_arrow to the right in each
18 # frame
19 display.sleep(True) # turn display off while updating blink bits
20 display.fill(0)
21 for y in range(display.height):
22     row = an_arrow[y]
23     for x in range(8):
24         bit = 1 << (7-x) & row
25         if bit:
26             display.pixel(x + 4, y, 50, blink=True)
27
28 display.blink(1000) # ranges from 270 to 2159; smaller the number to faster blink
29 display.sleep(False) # turn display on

```

Listing 3: examples/is31fl3731\_frame\_example.py

```

1 import time
2 import board
3 import busio
4 import adafruit_is31fl3731
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7
8 # arrow pattern in bits; top row-> bottom row, 8 bits in each row
9 arrow = bytearray((0x08, 0x0c, 0xfe, 0xff, 0xfe, 0x0c, 0x08, 0x00, 0x00))
10
11 # initial display using Feather CharlieWing LED 15 x 7
12 display = adafruit_is31fl3731.CharlieWing(i2c)
13 # uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
14 #display = adafruit_is31fl3731.Matrix(i2c)
15 # uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
16 #display = adafruit_is31fl3731.CharlieBonnet(i2c)
17
18
19 # first load the frame with the arrows; moves the arrow to the right in each
20 # frame

```

(continues on next page)

(continued from previous page)

```

21 display.sleep(True) # turn display off while frames are updated
22 for frame in range(8):
23     display.frame(frame, show=False)
24     display.fill(0)
25     for y in range(display.height):
26         row = arrow[y]
27         for x in range(8):
28             bit = 1 << (7-x) & row
29             # display the pixel into selected frame with varying intensity
30             if bit:
31                 display.pixel(x + frame, y, frame*2 + 1)
32 display.sleep(False)
33 # now tell the display to show the frame one at time
34 while True:
35     for frame in range(8):
36         display.frame(frame)
37         time.sleep(.1)

```

Listing 4: examples/is31fl3731\_text\_example.py

```

1  import board
2  import busio
3  import adafruit_framebuf
4  import adafruit_is31fl3731
5
6
7  i2c = busio.I2C(board.SCL, board.SDA)
8
9  # initial display using Feather CharlieWing LED 15 x 7
10 #display = adafruit_is31fl3731.CharlieWing(i2c)
11 # uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
12 #display = adafruit_is31fl3731.Matrix(i2c)
13 # uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
14 display = adafruit_is31fl3731.CharlieBonnet(i2c)
15
16 text_to_show = "Adafruit!!"
17
18 # Create a framebuffer for our display
19 buf = bytearray(32) # 2 bytes tall x 16 wide = 32 bytes (9 bits is 2 bytes)
20 fb = adafruit_framebuf.FrameBuffer(buf, display.width, display.height, adafruit_
    ↳ framebuf.MVLSB)
21
22
23 frame = 0 # start with frame 0
24 while True:
25     for i in range(len(text_to_show) * 9):
26         fb.fill(0)
27         fb.text(text_to_show, -i + display.width, 0, color=1)
28
29         # to improve the display flicker we can use two frame
30         # fill the next frame with scrolling text, then
31         # show it.
32         display.frame(frame, show=False)
33         # turn all LEDs off
34         display.fill(0)
35         for x in range(display.width):

```

(continues on next page)

(continued from previous page)

```

36         # using the FrameBuffer text result
37         bite = buf[x]
38         for y in range(display.height):
39             bit = 1 << y & bite
40             # if bit > 0 then set the pixel brightness
41             if bit:
42                 display.pixel(x, y, 50)
43
44         # now that the frame is filled, show it.
45         display.frame(frame, show=True)
46         frame = 0 if frame else 1

```

Listing 5: examples/is31fl3731\_wave\_example.py

```

1  import board
2  import busio
3  import adafruit_is31fl3731
4
5  i2c = busio.I2C(board.SCL, board.SDA)
6
7  sweep = [1, 2, 3, 4, 6, 8, 10, 15, 20, 30, 40, 60, 60, 40, 30, 20, 15, 10, 8, 6, 4, 3,
8  ↪ 2, 1]
9
10 frame = 0
11
12 # initialize display using Feather CharlieWing LED 15 x 7
13 display = adafruit_is31fl3731.CharlieWing(i2c)
14 # uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
15 #display = adafruit_is31fl3731.Matrix(i2c)
16 # uncomment next line if you are using Adafruit 16x8 Charlieplexed Bonnet
17 #display = adafruit_is31fl3731.CharlieBonnet(i2c)
18
19 while True:
20     for incr in range(24):
21         # to reduce update flicker, use two frames
22         # make a frame active, don't show it yet
23         display.frame(frame, show=False)
24         # fill the display with the next frame
25         for x in range(display.width):
26             for y in range(display.height):
27                 display.pixel(x, y, sweep[(x+y+incr)%24])
28         # show the next frame
29         display.frame(frame, show=True)
30         if frame:
31             frame = 0
32         else:
33             frame = 1

```

## 6.3 adafruit\_is31fl3731

CircuitPython driver for the IS31FL3731 charlieplex IC.

- Author(s): Tony DiCola, Melissa LeBlanc-Williams

### 6.3.1 Implementation Notes

#### Hardware:

- Adafruit 16x9 Charlieplexed PWM LED Matrix Driver - IS31FL3731
- Adafruit 15x7 CharliePlex LED Matrix Display FeatherWings

#### Software and Dependencies:

- Adafruit CircuitPython firmware (2.2.0+) for the ESP8622 and M0-based boards: <https://github.com/adafruit/circuitpython/releases>

**class** adafruit\_is31fl3731.**CharlieBonnet** (*i2c*, *address=116*)  
Supports the Charlieplexed bonnet

**static pixel\_addr** (*x*, *y*)  
Calculate the offset into the device array for x,y pixel

**class** adafruit\_is31fl3731.**CharlieWing** (*i2c*, *address=116*)  
Supports the Charlieplexed feather wing

**static pixel\_addr** (*x*, *y*)  
Calculate the offset into the device array for x,y pixel

**class** adafruit\_is31fl3731.**Matrix** (*i2c*, *address=116*)  
The Matrix class support the main function for driving the 16x9 matrix Display

#### Parameters

- **i2c\_device** (*i2c\_device*) – the connected i2c bus i2c\_device
- **address** – the device address; defaults to 0x74

**audio\_play** (*sample\_rate*, *audio\_gain=0*, *agc\_enable=False*, *agc\_fast=False*)  
Controls the audio play feature

**audio\_sync** (*value=None*)  
Set the audio sync feature register

**autoplay** (*delay=0*, *loops=0*, *frames=0*)  
Start autoplay

#### Parameters

- **delay** – in ms
- **loops** – number of loops - 0->7
- **frames** – number of frames: 0->7

**blink** (*rate=None*)  
Updates the blink register

**fade** (*fade\_in=None*, *fade\_out=None*, *pause=0*)  
Start and stop the fade feature. If both fade\_in and fade\_out are None (the default), the breath feature is used for fading. if fade\_in is None, then fade\_in = fade\_out. If fade\_out is None, then fade\_out = fade\_in

#### Parameters

- **fade\_in** – positive number; 0->100
- **fade-out** – positive number; 0->100
- **pause** – breath register 2 pause value

**fill** (*color=None, blink=None, frame=None*)

Fill the display with a brightness level

**Parameters**

- **color** – brightness 0->255
- **blink** – True if blinking is required
- **frame** – which frame to fill 0->7

**frame** (*frame=None, show=True*)

Set the current frame

**Parameters**

- **frame** – frame number; 0-7 or None. If None function returns current frame
- **show** – True to show the frame; False to not show.

**image** (*img, blink=None, frame=None*)

Set buffer to value of Python Imaging Library image. The image should be in 8-bit mode (L) and a size equal to the display size.

**Parameters**

- **img** – Python Imaging Library image
- **blink** – True to blink
- **frame** – the frame to set the image

**pixel** (*x, y, color=None, blink=None, frame=None*)

Blink or brightness for x-, y-pixel

**Parameters**

- **x** – horizontal pixel position
- **y** – vertical pixel position
- **color** – brightness value 0->255
- **blink** – True to blink
- **frame** – the frame to set the pixel

**static pixel\_addr** (*x, y*)

Calculate the offset into the device array for x,y pixel

**reset** ()

Kill the display for 10MS

**sleep** (*value*)

Set the Software Shutdown Register bit

**Parameters** **value** – True to set software shutdown bit; False unset

## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





### a

adafruit\_is31fl3731, [16](#)



## A

`adafruit_is31fl3731` (module), 16  
`audio_play()` (*adafruit\_is31fl3731.Matrix* method), 17  
`audio_sync()` (*adafruit\_is31fl3731.Matrix* method), 17  
`autoplay()` (*adafruit\_is31fl3731.Matrix* method), 17

## B

`blink()` (*adafruit\_is31fl3731.Matrix* method), 17

## C

`CharlieBonnet` (class in *adafruit\_is31fl3731*), 17  
`CharlieWing` (class in *adafruit\_is31fl3731*), 17

## F

`fade()` (*adafruit\_is31fl3731.Matrix* method), 17  
`fill()` (*adafruit\_is31fl3731.Matrix* method), 17  
`frame()` (*adafruit\_is31fl3731.Matrix* method), 18

## I

`image()` (*adafruit\_is31fl3731.Matrix* method), 18

## M

`Matrix` (class in *adafruit\_is31fl3731*), 17

## P

`pixel()` (*adafruit\_is31fl3731.Matrix* method), 18  
`pixel_addr()` (*adafruit\_is31fl3731.CharlieBonnet* static method), 17  
`pixel_addr()` (*adafruit\_is31fl3731.CharlieWing* static method), 17  
`pixel_addr()` (*adafruit\_is31fl3731.Matrix* static method), 18

## R

`reset()` (*adafruit\_is31fl3731.Matrix* method), 18

## S

`sleep()` (*adafruit\_is31fl3731.Matrix* method), 18