Adafruit IS31FL3731 Library Documentation

Release 1.0

Radomir Dopieralski

Aug 27, 2019

Contents

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

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

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 3

Contributing

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

CHAPTER 4

Building locally

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-is31fl3731 --
↪library_location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/is31fl3731_simpletest.py

```
import board
import busio
import adafruit_is31fl3731

i2c = busio.I2C(board.SCL, board.SDA)

# initialize display using Feather CharlieWing LED 15 x 7
display = adafruit_is31fl3731.CharlieWing(i2c)

# uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.Matrix(i2c)

# uncomment next line if you are using Adafruit 16x8 Charlieplexed Bonnet
#display = adafruit_is31fl3731.CharlieBonnet(i2c)

# draw a box on the display
# first draw the top and bottom edges
for x in range(display.width):
    display.pixel(x, 0, 50)
    display.pixel(x, display.height - 1, 50)
# now draw the left and right edges
for y in range(display.height):
    display.pixel(0, y, 50)
    display.pixel(display.width - 1, y, 50)
```

5.2 Other Examples

Listing 2: examples/is31fl3731_blink_example.py

```
import busio
import board
import adafruit_is31fl3731

i2c = busio.I2C(board.SCL, board.SDA)

# array pattern in bits; top row-> bottom row, 8 bits in each row
an_arrow = bytearray((0x08, 0x0c, 0xfe, 0xff, 0xfe, 0x0c, 0x08, 0x00, 0x00))

# initial display using Feather CharlieWing LED 15 x 7
display = adafruit_is31fl3731.CharlieWing(i2c)
# uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.Matrix(i2c)
# uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.CharlieBonnet(i2c)

# first load the frame with the arrows; moves the an_arrow to the right in each
# frame
display.sleep(True) # turn display off while updating blink bits
display.fill(0)
for y in range(display.height):
    row = an_arrow[y]
    for x in range(8):
        bit = 1 << (7-x) & row
        if bit:
            display.pixel(x + 4, y, 50, blink=True)

display.blink(1000) # ranges from 270 to 2159; smaller the number to faster blink
display.sleep(False) # turn display on
```

Listing 3: examples/is31fl3731_frame_example.py

```
import time
import board
import busio
import adafruit_is31fl3731

i2c = busio.I2C(board.SCL, board.SDA)

# arrow pattern in bits; top row-> bottom row, 8 bits in each row
arrow = bytearray((0x08, 0x0c, 0xfe, 0xff, 0xfe, 0x0c, 0x08, 0x00, 0x00))

# initial display using Feather CharlieWing LED 15 x 7
display = adafruit_is31fl3731.CharlieWing(i2c)
# uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.Matrix(i2c)
# uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.CharlieBonnet(i2c)


# first load the frame with the arrows; moves the arrow to the right in each
# frame
```

(continues on next page)

(continued from previous page)

```
display.sleep(True) # turn display off while frames are updated
for frame in range(8):
    display.frame(frame, show=False)
    display.fill(0)
    for y in range(display.height):
        row = arrow[y]
        for x in range(8):
            bit = 1 << (7-x) & row
            # display the pixel into selected frame with varying intensity
            if bit:
                display.pixel(x + frame, y, frame**2 + 1)
display.sleep(False)
# now tell the display to show the frame one at time
while True:
    for frame in range(8):
        display.frame(frame)
        time.sleep(.1)
```

Listing 4: examples/is31fl3731_text_example.py

```
import board
import busio
import adafruit_framebuf
import adafruit_is31fl3731


i2c = busio.I2C(board.SCL, board.SDA)

# initial display using Feather CharlieWing LED 15 x 7
#display = adafruit_is31fl3731.CharlieWing(i2c)
# uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.Matrix(i2c)
# uncomment line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
display = adafruit_is31fl3731.CharlieBonnet(i2c)

text_to_show = "Adafruit!!"

# Create a framebuffer for our display
buf = bytearray(32)  # 2 bytes tall x 16 wide = 32 bytes (9 bits is 2 bytes)
fb = adafruit_framebuf.FrameBuffer(buf, display.width, display.height, adafruit_
↪framebuf.MVLSB)


frame = 0 # start with frame 0
while True:
    for i in range(len(text_to_show) * 9):
        fb.fill(0)
        fb.text(text_to_show, -i + display.width, 0, color=1)

        # to improve the display flicker we can use two frame
        # fill the next frame with scrolling text, then
        # show it.
        display.frame(frame, show=False)
        # turn all LEDs off
        display.fill(0)
        for x in range(display.width):
```

(continues on next page)

(continued from previous page)

```
            # using the FrameBuffer text result
            bite = buf[x]
            for y in range(display.height):
                bit = 1 << y & bite
                # if bit > 0 then set the pixel brightness
                if bit:
                    display.pixel(x, y, 50)

        # now that the frame is filled, show it.
        display.frame(frame, show=True)
        frame = 0 if frame else 1
```

Listing 5: examples/is31fl3731_wave_example.py

```
import board
import busio
import adafruit_is31fl3731

i2c = busio.I2C(board.SCL, board.SDA)

sweep = [1, 2, 3, 4, 6, 8, 10, 15, 20, 30, 40, 60, 60, 40, 30, 20, 15, 10, 8, 6, 4, 3,
↪ 2, 1]

frame = 0

# initialize display using Feather CharlieWing LED 15 x 7
display = adafruit_is31fl3731.CharlieWing(i2c)
# uncomment next line if you are using Adafruit 16x9 Charlieplexed PWM LED Matrix
#display = adafruit_is31fl3731.Matrix(i2c)
# uncomment next line if you are using Adafruit 16x8 Charlieplexed Bonnet
#display = adafruit_is31fl3731.CharlieBonnet(i2c)

while True:
    for incr in range(24):
        # to reduce update flicker, use two frames
        # make a frame active, don't show it yet
        display.frame(frame, show=False)
        # fill the display with the next frame
        for x in range(display.width):
            for y in range(display.height):
                display.pixel(x, y, sweep[(x+y+incr)%24])
        # show the next frame
        display.frame(frame, show=True)
        if frame:
            frame = 0
        else:
            frame = 1
```

5.3 adafruit_is31fl3731

CircuitPython driver for the IS31FL3731 charlieplex IC.

- Author(s): Tony DiCola

5.3.1 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*)
Calulate 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*)
Calulate 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.

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*)
Calulate 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 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index