
Adafruit MAX7219 Library Documentation

Release 1.0

Michael McWethy

Feb 04, 2019

Contents

1	Dependencies	3
2	Usage Example	5
2.1	adafruit_max7219.Matrix8x8 Example	5
2.2	adafruit_max7219.BCDDigits Example	6
3	Contributing	7
4	Building locally	9
4.1	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_max7219.max7219 - MAX7219 LED Matrix/Digit Display Driver	14
5.3	See Also	14
5.3.1	Implementation Notes	14
5.4	adafruit_max7219.matrices.Matrix8x8	15
5.5	adafruit_max7219.bcdigits.BCDDigits	16
6	Indices and tables	17
	Python Module Index	19

CircuitPython driver for the MAX7219 LED matrix driver chip.

See [here](#) for the equivalent MicroPython driver.

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](#).

2.1 adafruit_max7219.Matrix8x8 Example

```
from adafruit_max7219 import matrices
from board import TX, RX, A2
import busio
import digitalio
import time

clk = RX
din = TX
cs = digitalio.DigitalInOut(A2)

spi = busio.SPI(clk, MOSI=din)
display = matrices.Matrix8x8(spi, cs)
while True:
    display.brightness(3)

    display.fill(1)
    display.pixel(3, 3)
    display.pixel(3, 4)
    display.pixel(4, 3)
    display.pixel(4, 4)
    display.show()
    time.sleep(3.0)

    display.clear_all()
    s = 'Hello, World!'
    for c in range(len(s)*8):
        display.fill(0)
        display.text(s, -c, 0)
        display.show()
        time.sleep(0.25)
```

2.2 adafruit_max7219.BCDDigits Example

```
from adafruit_max7219 import bcddigits
from board import TX, RX, A2
import bitbangio
import digitalio

clk = RX
din = TX
cs = digitalio.DigitalInOut(A2)

spi = bitbangio.SPI(clk, MOSI=din)
display = bcddigits.BCDDigits(spi, cs, nDigits=8)
display.clear_all()
display.show_str(0, '{:9.2f}'.format(-1234.56))
display.show()
```

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-max7219 --
↳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.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/max7219_simpletest.py

```
1 import time
2 from board import TX, RX, A1
3 import busio
4 import digitalio
5 from adafruit_max7219 import matrices
6
7 mosi = TX
8 clk = RX
9 cs = digitalio.DigitalInOut(A1)
10
11 spi = busio.SPI(clk, MOSI=mosi)
12
13 matrix = matrices.Matrix8x8(spi, cs)
14 while True:
15     print("Cycle start")
16     # all lit up
17     matrix.fill(True)
18     matrix.show()
19     time.sleep(0.5)
20
21     # all off
22     matrix.fill(False)
23     matrix.show()
24     time.sleep(0.5)
25
26     # one column of leds lit
27     for i in range(8):
```

(continues on next page)

(continued from previous page)

```

28     matrix.pixel(1, i, 1)
29 matrix.show()
30 time.sleep(0.5)
31 # now scroll the column to the right
32 for j in range(8):
33     matrix.scroll(1, 0)
34     matrix.show()
35     time.sleep(0.5)
36
37 # show a string one character at a time
38 adafruit = 'Adafruit'
39 for char in adafruit:
40     matrix.fill(0)
41     matrix.text(char, 0, 0)
42     matrix.show()
43     time.sleep(1.0)
44
45 # scroll the last character off the display
46 for i in range(8):
47     matrix.scroll(-1, 0)
48     matrix.show()
49     time.sleep(0.5)
50
51 # scroll a string across the display
52 for pixel_position in range(len(adafruit) * 8):
53     matrix.fill(0)
54     matrix.text(adafruit, -pixel_position, 0)
55     matrix.show()
56     time.sleep(0.25)

```

Listing 2: examples/showbcdigits.py

```

1  import time
2  import random
3  from board import TX, RX, A1
4  import busio
5  import digitalio
6  from adafruit_max7219 import bcdigits
7
8  mosi = TX
9  clk = RX
10 cs = digitalio.DigitalInOut(A1)
11
12 spi = busio.SPI(clk, MOSI=mosi)
13
14 leds = bcdigits.BCDDigits(spi, cs, nDigits=8)
15 while True:
16     # clear display and dim 0
17     leds.brightness(0)
18     leds.clear_all()
19
20     # place 8-digit number on display
21     value = 12345678
22     leds.show_str(0, '{:8}'.format(value))
23     leds.show()
24

```

(continues on next page)

(continued from previous page)

```

25  # increase the brightness slowly
26  for i in range(16):
27      leds.brightness(i)
28      time.sleep(0.5)
29
30  leds.brightness(3)
31
32  # show "-HELP-90" on display
33  leds.show_str(6, '90') # show 90 starting at position 6
34  leds.set_digit(0, 10) # show - at position 0
35  leds.set_digit(1, 12) # show H at position 1
36  leds.set_digit(2, 11) # show E at position 2
37  leds.set_digit(3, 13) # show L at position 3
38  leds.set_digit(4, 14) # show P at position 4
39  leds.set_digit(5, 10) # show - at position 5
40
41  leds.show()
42  time.sleep(1.0)
43
44  leds.clear_all()
45  leds.brightness(5)
46
47  # set the two dots and two 4-digit numbers
48  leds.show_dot(2, 1)
49  leds.show_dot(6, 1)
50  leds.show_str(0, ' 72.5')
51  leds.show_str(4, '-10.8')
52
53  leds.show()
54  time.sleep(1.0)
55
56  leds.brightness(10)
57  leds.clear_all()
58  # show a 4 character numeric string
59  leds.show_str(0, ' 0')
60  leds.show()
61  time.sleep(1.0)
62
63  leds.clear_all()
64  # show 0->8
65  for digit in range(8):
66      leds.set_digit(digit, digit)
67
68  leds.show()
69  time.sleep(1.0)
70
71  # show random 8-digit numbers via show_str
72  for _ in range(10):
73      number = random.uniform(-1.0, 1.0)
74      number *= 10000.0
75      number_string = '{:9.3f}'.format(number)
76      leds.clear_all()
77      leds.show_str(0, number_string)
78      leds.show()
79      time.sleep(1.0)
80
81  # show the help string

```

(continues on next page)

(continued from previous page)

```
82     leds.clear_all()
83     leds.show_help(2)
84     leds.show()
85
86     time.sleep(1.0)
```

5.2 adafruit_max7219.max7219 - MAX7219 LED Matrix/Digit Display Driver

CircuitPython library to support MAX7219 LED Matrix/Digit Display Driver. This library supports the use of the MAX7219-based display in CircuitPython, either an 8x8 matrix or a 8 digit 7-segment numeric display.

5.3 See Also

- `matrices.Maxtrix8x8` is a class support an 8x8 led matrix display
- `bcdigits.BCDDigits` is a class that support the 8 digit 7-segment display

Beware that most CircuitPython compatible hardware are 3.3v logic level! Make sure that the input pin is 5v tolerant.

- Author(s): Michael McWethy

5.3.1 Implementation Notes

Hardware:

- Adafruit [MAX7219CNG LED Matrix/Digit Display Driver - MAX7219](#) (Product ID: 453)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

Notes: #. Datasheet: <https://cdn-shop.adafruit.com/datasheets/MAX7219.pdf>

class `adafruit_max7219.max7219.MAX7219` (*width, height, spi, cs, *, baudrate=8000000, polarity=0, phase=0*)
MAX2719 - driver for displays based on max719 chip_select

Parameters

- **width** (*int*) – the number of pixels wide
- **height** (*int*) – the number of pixels high
- **spi** (*object*) – an spi busio or spi bitbangio object
- **chip_select** (*DigitalInOut*) – digital in/out to use as chip select signal
- **baudrate** – for SPIDevice baudrate (default 8000000)
- **polarity** – for SPIDevice polarity (default 0)
- **phase** – for SPIDevice phase (default 0)

brightness (*value*)

Controls the brightness of the display.

Parameters **value** (*int*) – 0->15 dimmest to brightest

fill (*bit_value*)

Fill the display buffer.

Parameters **bit_value** (*int*) – value > 0 set the buffer bit, else clears the buffer bit

init_display ()

Must be implemented by derived class (matrices, bcddigits)

pixel (*xpos, ypos, bit_value=None*)

Set one buffer bit

Parameters

- **xpos** – x position to set bit
- **ypos** – y position to set bit
- **bit_value** (*int*) – value > 0 sets the buffer bit, else clears the buffer bit

scroll (*delta_x, delta_y*)

Srcolls the display using delta_x,delta_y.

show ()

Updates the display.

write_cmd (*cmd, data*)

Writes a command to spi device.

5.4 adafruit_max7219.matrices.Matrix8x8

class adafruit_max7219.matrices.**Matrix8x8** (*spi, cs*)

Driver for a 8x8 LED matrix based on the MAX7219 chip.

Parameters

- **spi** (*object*) – an spi busio or spi bitbangio object
- **cs** (*DigitalInOut*) – digital in/out to use as chip select signal

clear_all ()

Clears all matrix leds.

init_display ()

Must be implemented by derived class (matrices, bcddigits)

text (*strg, xpos, ypos, bit_value=1*)

Draw text in the 8x8 matrix.

Parameters

- **xpos** (*int*) – x position of LED in matrix
- **ypos** (*int*) – y position of LED in matrix
- **strg** (*string*) – string to place in to display
- **bit_value** – > 1 sets the text, otherwise resets

5.5 adafruit_max7219.bcdigits.BCDDigits

class adafruit_max7219.bcdigits.BCDDigits (*spi*, *cs*, *nDigits=1*)

Basic support for display on a 7-Segment BCD display controlled by a Max7219 chip using SPI.

Parameters

- **spi** (*object*) – an spi busio or spi bitbangio object
- **cs** (*DigitalInOut*) – digital in/out to use as chip select signal
- **nDigits** (*int*) – number of led 7-segment digits; default 1; max 8

clear_all ()

Clear all digits and decimal points.

init_display ()

Must be implemented by derived class (matrices, bcdigits)

set_digit (*dpos*, *value*)

Display one digit.

Parameters

- **dpos** (*int*) – the digit position; zero-based
- **value** (*int*) – integer ranging from 0->15

set_digits (*start*, *values*)

Display digits from a list.

Parameters

- **s** (*int*) – digit to start display zero-based
- **ds** (*list*) – list of integer values ranging from 0->15

show_dot (*dpos*, *bit_value=None*)

The decimal point for a digit.

Parameters

- **dpos** (*int*) – the digit to set the decimal point zero-based
- **value** (*int*) – value > zero lights the decimal point, else unlights the point

show_help (*start*)

Display the word HELP in the display.

Parameters **start** (*int*) – start position to show HELP

show_str (*start*, *strg*)

Displays a numeric str in the display. Shows digits 0-9, -, and ..

Parameters

- **start** (*int*) – start position to show the numeric string
- **str** (*string*) – the numeric string

CHAPTER 6

Indices and tables

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

a

`adafruit_max7219.bcdigits`, [15](#)
`adafruit_max7219.matrices`, [15](#)
`adafruit_max7219.max7219`, [14](#)

A

adafruit_max7219.bcdigits (module), 15
adafruit_max7219.matrices (module), 15
adafruit_max7219.max7219 (module), 14

B

BCDDigits (class in adafruit_max7219.bcdigits), 16
brightness() (adafruit_max7219.max7219.MAX7219 method), 14

C

clear_all() (adafruit_max7219.bcdigits.BCDDigits method), 16
clear_all() (adafruit_max7219.matrices.Matrix8x8 method), 15

F

fill() (adafruit_max7219.max7219.MAX7219 method), 15

I

init_display() (adafruit_max7219.bcdigits.BCDDigits method), 16
init_display() (adafruit_max7219.matrices.Matrix8x8 method), 15
init_display() (adafruit_max7219.max7219.MAX7219 method), 15

M

Matrix8x8 (class in adafruit_max7219.matrices), 15
MAX7219 (class in adafruit_max7219.max7219), 14

P

pixel() (adafruit_max7219.max7219.MAX7219 method), 15

S

scroll() (adafruit_max7219.max7219.MAX7219 method), 15

set_digit() (adafruit_max7219.bcdigits.BCDDigits method), 16

set_digits() (adafruit_max7219.bcdigits.BCDDigits method), 16

show() (adafruit_max7219.max7219.MAX7219 method), 15

show_dot() (adafruit_max7219.bcdigits.BCDDigits method), 16

show_help() (adafruit_max7219.bcdigits.BCDDigits method), 16

show_str() (adafruit_max7219.bcdigits.BCDDigits method), 16

T

text() (adafruit_max7219.matrices.Matrix8x8 method), 15

W

write_cmd() (adafruit_max7219.max7219.MAX7219 method), 15