

---

# **Adafruit MAX7219 Library Documentation**

***Release 1.0***

**Michael McWethy**

**Jan 20, 2021**



---

## 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>
3.1	adafruit_max7219.Matrix8x8 Example . . . . .	7
3.2	adafruit_max7219.BCDDigits Example . . . . .	8
<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	adafruit_max7219.max7219 - MAX7219 LED Matrix/Digit Display Driver . . . . .	16
6.3	See Also . . . . .	16
6.3.1	Implementation Notes . . . . .	16
6.4	adafruit_max7219.matrices.Matrix8x8 . . . . .	17
6.5	adafruit_max7219.bcdigits.BCDDigits . . . . .	18
<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 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](#).





## 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-max7219
```

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

```
sudo pip3 install adafruit-circuitpython-max7219
```

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-max7219
```



### 3.1 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)
```

## 3.2 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 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/max7219\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  from board import TX, RX, A1
6  import busio
7  import digitalio
8  from adafruit_max7219 import matrices
9
10 mosi = TX
11 clk = RX
12 cs = digitalio.DigitalInOut(A1)
13
14 spi = busio.SPI(clk, MOSI=mosi)
15
16 matrix = matrices.Matrix8x8(spi, cs)
17 while True:
18     print("Cycle start")
19     # all lit up
20     matrix.fill(True)
21     matrix.show()
22     time.sleep(0.5)
23
24     # all off
25     matrix.fill(False)
26     matrix.show()
27     time.sleep(0.5)
```

(continues on next page)

(continued from previous page)

```

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

```

Listing 2: examples/max7219\_showbcdigits.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import random
6  from board import TX, RX, A1
7  import busio
8  import digitalio
9  from adafruit_max7219 import bcdigits
10
11  mosi = TX
12  clk = RX
13  cs = digitalio.DigitalInOut(A1)
14
15  spi = busio.SPI(clk, MOSI=mosi)
16
17  leds = bcdigits.BCDDigits(spi, cs, nDigits=8)
18  while True:
19     # clear display and dim 0
20     leds.brightness(0)
21     leds.clear_all()

```

(continues on next page)

(continued from previous page)

```

22
23     # place 8-digit number on display
24     value = 12345678
25     leds.show_str(0, "{:8}".format(value))
26     leds.show()
27
28     # increase the brightness slowly
29     for i in range(16):
30         leds.brightness(i)
31         time.sleep(0.5)
32
33     leds.brightness(3)
34
35     # show "-HELP-90" on display
36     leds.show_str(6, "90") # show 90 starting at position 6
37     leds.set_digit(0, 10) # show - at position 0
38     leds.set_digit(1, 12) # show H at position 1
39     leds.set_digit(2, 11) # show E at position 2
40     leds.set_digit(3, 13) # show L at position 3
41     leds.set_digit(4, 14) # show P at position 4
42     leds.set_digit(5, 10) # show - at position 5
43
44     leds.show()
45     time.sleep(1.0)
46
47     leds.clear_all()
48     leds.brightness(5)
49
50     # set the two dots and two 4-digit numbers
51     leds.show_dot(2, 1)
52     leds.show_dot(6, 1)
53     leds.show_str(0, " 72.5")
54     leds.show_str(4, "-10.8")
55
56     leds.show()
57     time.sleep(1.0)
58
59     leds.brightness(10)
60     leds.clear_all()
61     # show a 4 character numeric string
62     leds.show_str(0, "  0")
63     leds.show()
64     time.sleep(1.0)
65
66     leds.clear_all()
67     # show 0->8
68     for digit in range(8):
69         leds.set_digit(digit, digit)
70
71     leds.show()
72     time.sleep(1.0)
73
74     # show random 8-digit numbers via show_str
75     for _ in range(10):
76         number = random.uniform(-1.0, 1.0)
77         number *= 10000.0
78         number_string = "{:9.3f}".format(number)

```

(continues on next page)

(continued from previous page)

```

79     leds.clear_all()
80     leds.show_str(0, number_string)
81     leds.show()
82     time.sleep(1.0)
83
84     # show the help string
85     leds.clear_all()
86     leds.show_help(2)
87     leds.show()
88
89     time.sleep(1.0)

```

## 6.2 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.

## 6.3 See Also

- `matrices.Maxtrix8x8` is a class support an 8x8 led matrix display
- `bcddigits.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

### 6.3.1 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](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*)

MAX7219 - 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.

## 6.4 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

## 6.5 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 7

---

### Indices and tables

---

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





### a

`adafruit_max7219.bcdigits`, [18](#)  
`adafruit_max7219.matrices`, [17](#)  
`adafruit_max7219.max7219`, [16](#)



## A

`adafruit_max7219.bcdigits` (module), 18  
`adafruit_max7219.matrices` (module), 17  
`adafruit_max7219.max7219` (module), 16

## B

`BCDDigits` (class in `adafruit_max7219.bcdigits`), 18  
`brightness()` (`adafruit_max7219.max7219.MAX7219` method), 17

## C

`clear_all()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`clear_all()` (`adafruit_max7219.matrices.Matrix8x8` method), 17

## F

`fill()` (`adafruit_max7219.max7219.MAX7219` method), 17

## I

`init_display()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`init_display()` (`adafruit_max7219.matrices.Matrix8x8` method), 17  
`init_display()` (`adafruit_max7219.max7219.MAX7219` method), 17

## M

`Matrix8x8` (class in `adafruit_max7219.matrices`), 17  
`MAX7219` (class in `adafruit_max7219.max7219`), 16

## P

`pixel()` (`adafruit_max7219.max7219.MAX7219` method), 17

## S

`scroll()` (`adafruit_max7219.max7219.MAX7219` method), 17

`set_digit()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`set_digits()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`show()` (`adafruit_max7219.max7219.MAX7219` method), 17  
`show_dot()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`show_help()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18  
`show_str()` (`adafruit_max7219.bcdigits.BCDDigits` method), 18

## T

`text()` (`adafruit_max7219.matrices.Matrix8x8` method), 17

## W

`write_cmd()` (`adafruit_max7219.max7219.MAX7219` method), 17