
Adafruit MAX7219 Library Documentation

Release 1.0

Michael McWethy

Oct 25, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
3.1	adafruit_max7219.Matrix8x8 Example	7
3.2	adafruit_max7219.BCDDigits Example	8
4	Documentation	9
5	Contributing	11
6	Documentation	13
7	Table of Contents	15
7.1	Simple test	15
7.2	adafruit_max7219.max7219 - MAX7219 LED Matrix/Digit Display Driver	18
7.3	See Also	18
7.3.1	Implementation Notes	18
7.4	adafruit_max7219.matrices.Matrix8x8	19
7.5	adafruit_max7219.bcdigits.BCDDigits	20
8	Indices and tables	23
	Python Module Index	25
	Index	27

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

Documentation

API documentation for this library can be found on [Read the Docs](#).

CHAPTER 5

Contributing

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

CHAPTER 6

Documentation

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

7.1 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)

```

7.2 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.

7.3 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

7.3.1 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: int, height: int, spi: busio.SPI, cs: digitalio.DigitalInOut, *, baudrate: int = 8000000, polarity: int = 0, phase: int = 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** (*SPI*) – an spi busio or spi bitbangio object

- **chip_select** (*DigitalInOut*) – digital in/out to use as chip select signal
- **baudrate** (*int*) – for SPIDevice baudrate (default 8000000)
- **polarity** (*int*) – for SPIDevice polarity (default 0)
- **phase** (*int*) – for SPIDevice phase (default 0)

brightness (*value: int*) → None
Controls the brightness of the display.

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

fill (*bit_value: int*) → None
Fill the display buffer.

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

init_display () → None
Must be implemented by derived class (*matrices*, *bcddigits*)

pixel (*xpos: int, ypos: int, bit_value: int = None*) → None
Set one buffer bit

Parameters

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

scroll (*delta_x: int, delta_y: int*) → None
Srcolls the display using *delta_x*, *delta_y*.

Parameters

- **delta_x** (*int*) – positions to scroll in the x direction
- **delta_y** (*int*) – positions to scroll in the y direction

show () → None
Updates the display.

write_cmd (*cmd: int, data: int*) → None
Writes a command to spi device.

Parameters

- **cmd** (*int*) – register address to write data to
- **data** (*int*) – data to be written to commanded register

7.4 adafruit_max7219.matrices.Matrix8x8

class `adafruit_max7219.matrices.Matrix8x8` (*spi: busio.SPI, cs: digitalio.DigitalInOut*)
Driver for a 8x8 LED matrix based on the MAX7219 chip.

Parameters

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

clear_all () → None
Clears all matrix leds.

init_display () → None
Must be implemented by derived class (matrices, bcddigits)

text (strg: str, xpos: int, ypos: int, bit_value: int = 1) → None
Draw text in the 8x8 matrix.

Parameters

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

7.5 adafruit_max7219.bcddigits.BCDDigits

class adafruit_max7219.bcddigits.BCDDigits (spi: busio.SPI, cs: digitalio.DigitalInOut, nDigits: int = 1)

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

Parameters

- **spi** (SPI) – 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 () → None
Clear all digits and decimal points.

init_display () → None
Must be implemented by derived class (matrices, bcddigits)

set_digit (dpos: int, value: int) → None
Display one digit.

Parameters

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

set_digits (start: int, values: List[int]) → None
Display digits from a list.

Parameters

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

show_dot (dpos: int, bit_value: int = None) → None
The decimal point for a digit.

Parameters

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

show_help (*start: int*) → None

Display the word HELP in the display.

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

show_str (*start: int, strg: str*) → None

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

Parameters

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

CHAPTER 8

Indices and tables

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

a

adafruit_max7219.bcdigits, 20
adafruit_max7219.matrices, 19
adafruit_max7219.max7219, 18

A

adafruit_max7219.bcdigits (*module*), 20
adafruit_max7219.matrices (*module*), 19
adafruit_max7219.max7219 (*module*), 18

B

BCDDigits (*class in adafruit_max7219.bcdigits*), 20
brightness() (*adafruit_max7219.max7219.MAX7219*
method), 19

C

clear_all() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20
clear_all() (*adafruit_max7219.matrices.Matrix8x8*
method), 19

F

fill() (*adafruit_max7219.max7219.MAX7219*
method), 19

I

init_display() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20
init_display() (*adafruit_max7219.matrices.Matrix8x8*
method), 20
init_display() (*adafruit_max7219.max7219.MAX7219*
method), 19

M

Matrix8x8 (*class in adafruit_max7219.matrices*), 19
MAX7219 (*class in adafruit_max7219.max7219*), 18

P

pixel() (*adafruit_max7219.max7219.MAX7219*
method), 19

S

scroll() (*adafruit_max7219.max7219.MAX7219*
method), 19

set_digit() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20

set_digits() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20

show() (*adafruit_max7219.max7219.MAX7219*
method), 19

show_dot() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20

show_help() (*adafruit_max7219.bcdigits.BCDDigits*
method), 20

show_str() (*adafruit_max7219.bcdigits.BCDDigits*
method), 21

T

text() (*adafruit_max7219.matrices.Matrix8x8*
method), 20

W

write_cmd() (*adafruit_max7219.max7219.MAX7219*
method), 19