Adafruit MAX7219 Library Documentation

Release 1.0

Michael McWethy

Dec 29, 2018

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.bcddigits.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.

CHAPTER 2

Usage Example

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.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/max7219_simpletest.py

```
import time
from board import TX, RX, A1
import busio
import digitalio
from adafruit_max7219 import matrices

mosi = TX
clk = RX
cs = digitalio.DigitalInOut(A1)

spi = busio.SPI(clk, MOSI=mosi)

matrix = matrices.Matrix8x8(spi, cs)
while True:
    print("Cycle start")
    # all lit up
    matrix.fill(True)
    matrix.show()
    time.sleep(0.5)

    # all off
    matrix.fill(False)
    matrix.show()
    time.sleep(0.5)

    # one column of leds lit
    for i in range(8):
```

(continues on next page)

(continued from previous page)

```
        matrix.pixel(1, i, 1)
    matrix.show()
    time.sleep(0.5)
    # now scroll the column to the right
    for j in range(8):
        matrix.scroll(1, 0)
        matrix.show()
        time.sleep(0.5)

    # show a string one character at a time
    adafruit = 'Adafruit'
    for char in adafruit:
        matrix.fill(0)
        matrix.text(char, 0, 0)
        matrix.show()
        time.sleep(1.0)

    # scroll the last character off the display
    for i in range(8):
        matrix.scroll(-1, 0)
        matrix.show()
        time.sleep(0.5)

    # scroll a string across the display
    for pixel_position in range(len(adafruit) * 8):
        matrix.fill(0)
        matrix.text(adafruit, -pixel_position, 0)
        matrix.show()
        time.sleep(0.25)
```

Listing 2: examples/showbcddigits.py

```
import time
import random
from board import TX, RX, A1
import busio
import digitalio
from adafruit_max7219 import bcddigits

mosi = TX
clk = RX
cs = digitalio.DigitalInOut(A1)

spi = busio.SPI(clk, MOSI=mosi)

leds = bcddigits.BCDDigits(spi, cs, nDigits=8)
while True:
    # clear display and dim 0
    leds.brightness(0)
    leds.clear_all()

    # place 8-digit number on display
    value = 12345678
    leds.show_str(0, '{:8}'.format(value))
    leds.show()

```

(continues on next page)

(continued from previous page)

```
    # increase the brightness slowly
    for i in range(16):
        leds.brightness(i)
        time.sleep(0.5)

    leds.brightness(3)

    # show "-HELP-90" on display
    leds.show_str(6, '90')  # show 90 starting at position 6
    leds.set_digit(0, 10)   # show - at position 0
    leds.set_digit(1, 12)   # show H at position 1
    leds.set_digit(2, 11)   # show E at position 2
    leds.set_digit(3, 13)   # show L at position 3
    leds.set_digit(4, 14)   # show P at position 4
    leds.set_digit(5, 10)   # show - at position 5

    leds.show()
    time.sleep(1.0)

    leds.clear_all()
    leds.brightness(5)

    # set the two dots and two 4-digit numbers
    leds.show_dot(2, 1)
    leds.show_dot(6, 1)
    leds.show_str(0, ' 72.5')
    leds.show_str(4, '-10.8')

    leds.show()
    time.sleep(1.0)

    leds.brightness(10)
    leds.clear_all()
    # show a 4 character numeric string
    leds.show_str(0, '   0')
    leds.show()
    time.sleep(1.0)

    leds.clear_all()
    # show 0->8
    for digit in range(8):
        leds.set_digit(digit, digit)

    leds.show()
    time.sleep(1.0)

    # show random 8-digit numbers via show_str
    for _ in range(10):
        number = random.uniform(-1.0, 1.0)
        number *= 10000.0
        number_string = '{:9.3f}'.format(number)
        leds.clear_all()
        leds.show_str(0, number_string)
        leds.show()
        time.sleep(1.0)

    # show the help string
```

(continues on next page)

(continued from previous page)

```
    leds.clear_all()
    leds.show_help(2)
    leds.show()

    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
- 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

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.bcddigits.BCDDigits

class adafruit_max7219.bcddigits.**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, bcddigits)

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

Python Module Index

a

adafruit_max7219.bcddigits, 15
adafruit_max7219.matrices, 15
adafruit_max7219.max7219, 14

Index

A

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

B

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

C

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

F

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

I

init_display() (adafruit_max7219.bcddigits.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.bcddigits.BCDDigits method), 16
set_digits() (adafruit_max7219.bcddigits.BCDDigits method), 16
show() (adafruit_max7219.max7219.MAX7219 method), 15
show_dot() (adafruit_max7219.bcddigits.BCDDigits method), 16
show_help() (adafruit_max7219.bcddigits.BCDDigits method), 16
show_str() (adafruit_max7219.bcddigits.BCDDigits method), 16

T

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

W

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