AdafruitminiQR Library Documentation

Release 1.0

ladyada

Jun 24, 2019

Contents

1 Dependencies **3**

2 Usage Example **5**

3 Contributing **7**

4 Building locally **9**

4.1 Zip release files 9

4.2 Sphinx documentation 9

5 Table of Contents **11**

5.1 Simple test 11

5.2 `adafruit_miniqr` 12

5.2.1 Implementation Notes 12

6 Indices and tables **15**

Python Module Index **17**

Index **19**

A non-hardware dependant miniature QR generator library. All native Python!

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

```
import adafruit_miniqr

qr = adafruit_miniqr.QRCode()
qr.add_data(b'https://www.adafruit.com')
qr.make()
print(qr.matrix)
```

CHAPTER 3

Contributing

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

CHAPTER 4

Building locally

4.1 Zip release files

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-miniqr --library_
↪location .
```

4.2 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/miniqr_simpletest.py

```python
import sys
import adafruit_miniqr

# For drawing filled rectangles to the console:
out = sys.stdout
WHITE = "\x1b[1;47m  \x1b[40m"
BLACK = "  "

def prettyprint_QR(matrix):
    # white 4-pixel border at top
    for _ in range(4):
        for _ in range(matrix.width+8):
            out.write(WHITE)
        print()
    for y in range(matrix.height):
        out.write(WHITE*4)   # 4-pixel border to left
        for x in range(matrix.width):
            if matrix[x, y]:
                out.write(BLACK)
            else:
                out.write(WHITE)
        out.write(WHITE*4)   # 4-pixel bporder to right
        print()
    # white 4-pixel border at bottom
    for _ in range(4):
        for _ in range(matrix.width+8):
            out.write(WHITE)
```

(continues on next page)

(continued from previous page)

```
        print()

qr = adafruit_miniqr.QRCode(qr_type=3, error_correct=adafruit_miniqr.L)
qr.add_data(b'https://www.adafruit.com')
qr.make()
print(qr.matrix)
prettyprint_QR(qr.matrix)
```

5.2 adafruit_miniqr

A non-hardware dependant miniature QR generator library. All native Python!

- Author(s): ladyada

5.2.1 Implementation Notes

Hardware:

- Any!

Software and Dependencies:

- Python 3

class adafruit_miniqr.**QRBitBuffer**

Storage class for a length of individual bits

get (*index*)

The bit value at a location

get_length_bits ()

Size of bit buffer

put (*num*, *length*)

Add a number of bits from a single integer value

put_bit (*bit*)

Insert one bit at the end of the bit buffer

class adafruit_miniqr.**QRBitMatrix** (*width*, *height*)

A bit-packed storage class for matrices

class adafruit_miniqr.**QRCode** (***, *qr_type=None*, *error_correct=1*)

The generator class for QR code matrices

add_data (*data*)

Add more data to the QR code, must be bytestring stype

make (***, *test=False*, *mask_pattern=0*)

Perform the actual generation of the QR matrix. To keep things small and speedy we don’t generate all 8 mask patterns and pick the best. Instead, please pass in a desired mask_pattern, the default mask is 0.

class adafruit_miniqr.**QRPolynomial** (*num*, *shift*)

Structure for creating and manipulating error code polynomials

get (*index*)

The exponent at the index location

get_length()
Length of the poly

multiply(*e*)
Multiply two polynomials, returns a new one

class adafruit_miniqr.**QRUtil**
A selection of bit manipulation tools for QR generation and BCH encoding

static get_BCH_digit(*data*)
Count digits in data

static get_BCH_type_info(*data*)
Encode with G15 BCH mask

static get_BCH_type_number(*data*)
Encode with G18 BCH mask

static get_error_correct_polynomial(*ecc_length*)
Generate a ecc polynomial

static get_mask(*mask*, *i*, *j*)
Perform matching calculation on two vals for given pattern mask

static get_pattern_position(*qr_type*)
The mask pattern position array for this QR type

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_miniqr, 12

Index

A

adafruit_miniqr (*module*), 12
add_data() (*adafruit_miniqr.QRCode method*), 12

G

get() (*adafruit_miniqr.QRBitBuffer method*), 12
get() (*adafruit_miniqr.QRPolynomial method*), 12
get_BCH_digit() (*adafruit_miniqr.QRUtil static method*), 13
get_BCH_type_info() (*adafruit_miniqr.QRUtil static method*), 13
get_BCH_type_number() (*adafruit_miniqr.QRUtil static method*), 13
get_error_correct_polynomial() (*adafruit_miniqr.QRUtil static method*), 13
get_length() (*adafruit_miniqr.QRPolynomial method*), 12
get_length_bits() (*adafruit_miniqr.QRBitBuffer method*), 12
get_mask() (*adafruit_miniqr.QRUtil static method*), 13
get_pattern_position() (*adafruit_miniqr.QRUtil static method*), 13

M

make() (*adafruit_miniqr.QRCode method*), 12
multiply() (*adafruit_miniqr.QRPolynomial method*), 13

P

put() (*adafruit_miniqr.QRBitBuffer method*), 12
put_bit() (*adafruit_miniqr.QRBitBuffer method*), 12

Q

QRBitBuffer (*class in adafruit_miniqr*), 12
QRBitMatrix (*class in adafruit_miniqr*), 12
QRCode (*class in adafruit_miniqr*), 12
QRPolynomial (*class in adafruit_miniqr*), 12
QRUtil (*class in adafruit_miniqr*), 13