

---

# AdafruitminiQR Library Documentation

*Release 1.0*

**ladyada**

**Nov 13, 2019**



---

## 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>
<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_minqr .....	14
6.2.1	Implementation Notes .....	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



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

---

### 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-miniqr
```

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

```
sudo pip3 install adafruit-circuitpython-miniqr
```

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-miniqr
```



## CHAPTER 3

---

### Usage Example

---

```
import adafruit_miniqr

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



## 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/miniqr\_simpletest.py

```
1 import sys
2 import adafruit_miniqr
3
4 # For drawing filled rectangles to the console:
5 out = sys.stdout
6 WHITE = "\x1b[1;47m  \x1b[40m"
7 BLACK = "  "
8
9 def prettyprint_QR(matrix):
10     # white 4-pixel border at top
11     for _ in range(4):
12         for _ in range(matrix.width+8):
13             out.write(WHITE)
14         print()
15     for y in range(matrix.height):
16         out.write(WHITE*4) # 4-pixel border to left
17         for x in range(matrix.width):
18             if matrix[x, y]:
19                 out.write(BLACK)
20             else:
21                 out.write(WHITE)
22         out.write(WHITE*4) # 4-pixel bborder to right
23         print()
24     # white 4-pixel border at bottom
25     for _ in range(4):
26         for _ in range(matrix.width+8):
27             out.write(WHITE)
```

(continues on next page)

(continued from previous page)

```
28     print()
29
30 qr = adafruit_miniqr.QRCode(qr_type=3, error_correct=adafruit_miniqr.L)
31 qr.add_data(b'https://www.adafruit.com')
32 qr.make()
33 print(qr.matrix)
34 prettyprint_QR(qr.matrix)
```

## 6.2 adafruit\_miniqr

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

- Author(s): ladyada

### 6.2.1 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_minigr.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 7

---

### Indices and tables

---

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



**a**

adafruit\_minqr, 14





## A

adafruit\_minqr (*module*), 14  
add\_data() (*adafruit\_minqr.QRCode method*), 14

## G

get() (*adafruit\_minqr.QRBitBuffer method*), 14  
get() (*adafruit\_minqr.QRPolynomial method*), 14  
get\_BCH\_digit() (*adafruit\_minqr.QRUtil static method*), 15  
get\_BCH\_type\_info() (*adafruit\_minqr.QRUtil static method*), 15  
get\_BCH\_type\_number() (*adafruit\_minqr.QRUtil static method*), 15  
get\_error\_correct\_polynomial() (*adafruit\_minqr.QRUtil static method*), 15  
get\_length() (*adafruit\_minqr.QRPolynomial method*), 14  
get\_length\_bits() (*adafruit\_minqr.QRBitBuffer method*), 14  
get\_mask() (*adafruit\_minqr.QRUtil static method*), 15  
get\_pattern\_position() (*adafruit\_minqr.QRUtil static method*), 15

## M

make() (*adafruit\_minqr.QRCode method*), 14  
multiply() (*adafruit\_minqr.QRPolynomial method*), 15

## P

put() (*adafruit\_minqr.QRBitBuffer method*), 14  
put\_bit() (*adafruit\_minqr.QRBitBuffer method*), 14

## Q

QRBitBuffer (*class in adafruit\_minqr*), 14  
QRBitMatrix (*class in adafruit\_minqr*), 14  
QRCode (*class in adafruit\_minqr*), 14

QRPolynomial (*class in adafruit\_minqr*), 14  
QRUtil (*class in adafruit\_minqr*), 15