
Adafruitradio Library Documentation

Release 1.0

Nicholas H.Tollervey

Sep 28, 2021

Contents

1	Dependencies	3
2	Usage Example	5
3	Unit Tests	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_ble_radio	14
7	Indices and tables	17
	Python Module Index	19
	Index	21

This library provides simple byte and string based inter-device communication via BLE.

It works like a walkie-talkie: configure your device to use a certain channel (numbered 0-255, default being 42) and it will broadcast on that channel and receive any messages from other devices using that channel.

CHAPTER 1

Dependencies

This library 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

All the functionality is exposed via the very simple Radio class:

```
from adafruit_ble_radio import Radio

# A radio instance listens/broadcasts on a numbered channel.
r = Radio(channel=7)

# Update radio instance settings.
r.configure(channel=9)

# Broadcast a simple string message.
r.send("Hello")

# Broadcast raw bytes.
r.send_bytes(b"Hello")

# A loop to listen for incoming string based messages...
while True:
    msg = r.receive()
    if msg:
        print(msg)

# Alternatively, to get the raw bytes and other details...
while True:
    msg = r.receive_full()
    if msg:
        msg_bytes = msg[0]
        msg_strength = msg[1]
        msg_time = msg[2]
        print("Recieved {} (strength {}, at time {})".format(
            msg_bytes,
            msg_strength,
            msg_time))
```


CHAPTER 3

Unit Tests

To run the test suite you should have `pytest` and `pytest-cov` installed (pip install `pytest` `pytest-cov`).

Run the unit tests with the following command:

```
$ pytest --cov-report term-missing --cov=adafruit_ble_radio tests/
```


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/ble_radio_simpletest.py

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """
5 This demo uses the adafruit_radio module to send and receive messages.
6 Devices are switched between broadcast and scanning using the slide switch.
7 The buttons change the message to be sent.
8 """
9 import digitalio
10 import board
11 from adafruit_ble_radio import Radio
12
13
14 slide_switch = digitalio.DigitalInOut(board.SLIDE_SWITCH)
15 slide_switch.pull = digitalio.Pull.UP
16 button_a = digitalio.DigitalInOut(board.BUTTON_A)
17 button_a.pull = digitalio.Pull.DOWN
18 button_b = digitalio.DigitalInOut(board.BUTTON_B)
19 button_b.pull = digitalio.Pull.DOWN
20
21 led = digitalio.DigitalInOut(board.D13)
22 led.switch_to_output()
23
24 msg = [
25     "hello",
26     "hi",
27     "foo",
```

(continues on next page)

```

28     "bar",
29     "baz",
30 ]
31
32 i = 0
33 r = Radio()
34
35 while True:
36     if slide_switch.value:
37         print("Sending messages...")
38         while slide_switch.value:
39             last_i = i
40             if button_a.value:
41                 i += 1
42             if button_b.value:
43                 i -= 1
44             i %= len(msg)
45             m = msg[i]
46             print("Sending {}".format(m))
47             r.send(m)
48             # Alternative
49             # r.send_bytes(b"Arbitrary bytes")
50     else:
51         print("Scanning for messages...")
52         while not slide_switch.value:
53             m = r.receive_full()
54             if m:
55                 print("Received message: {}".format(m))
56             # Alternative
57             # m = r.receive()
58             # if m:
59             #     print(m)

```

6.2 adafruit_ble_radio

Simple byte and string based inter-device communication via BLE.

- Author(s): Nicholas H.Tollervey for Adafruit Industries

Hardware:

Adafruit Feather nRF52840 Express <<https://www.adafruit.com/product/4062>> Adafruit Circuit Playground Bluefruit <<https://www.adafruit.com/product/4333>>

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>

`adafruit_ble_radio.AD_DURATION = 0.5`

Amount of time to advertise a message (in seconds).

`adafruit_ble_radio.MAX_LENGTH = 248`

Maximum length of a message (in bytes).

`class adafruit_ble_radio.Radio(**args)`

Represents a connection through which one can send or receive strings and bytes. The radio can be tuned to a specific channel upon initialisation or via the `configure` method.

configure (*channel=42*)

Set configuration values for the radio.

Parameters **channel** (*int*) – The channel (0-255) the radio is listening / broadcasting on.

receive (*timeout=1*)

Returns a message received on the channel on which the radio is listening.

Returns A string representation of the received message, or else None.

receive_full (*timeout=1*)

Returns a tuple containing three values representing a message received on the channel on which the radio is listening. If no message was received then `None` is returned.

The three values in the tuple represent:

- the bytes received.
- the RSSI (signal strength: 0 = max, -255 = min).
- a microsecond timestamp: the value returned by `time.monotonic()` when the message was received.

Parameters **timeout** (*float*) – The length of time (in seconds) the radio listens for a broadcast

Returns A tuple representation of the received message, or else None.

send (*message*)

Send a message string on the channel to which the radio is broadcasting.

Parameters **message** (*str*) – The message string to broadcast.

send_bytes (*message*)

Send bytes on the channel to which the radio is broadcasting.

Parameters **message** (*bytes*) – The bytes to broadcast.

CHAPTER 7

Indices and tables

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

a

`adafruit_ble_radio`, 14

A

AD_DURATION (*in module adafruit_ble_radio*), 14
adafruit_ble_radio (*module*), 14

C

configure() (*adafruit_ble_radio.Radio method*), 14

M

MAX_LENGTH (*in module adafruit_ble_radio*), 14

R

Radio (*class in adafruit_ble_radio*), 14
receive() (*adafruit_ble_radio.Radio method*), 15
receive_full() (*adafruit_ble_radio.Radio method*),
15

S

send() (*adafruit_ble_radio.Radio method*), 15
send_bytes() (*adafruit_ble_radio.Radio method*), 15