

---

**AdafruitBLE** *AppleNotificationCenterLibraryDoc*  
**Release 1.0**

**Scott Shawcroft**

**Oct 25, 2021**



---

## 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>Documentation</b>	<b>9</b>
<b>5</b>	<b>Contributing</b>	<b>11</b>
<b>6</b>	<b>Documentation</b>	<b>13</b>
<b>7</b>	<b>Table of Contents</b>	<b>15</b>
7.1	Simple test .....	15
7.2	adafruit_ble_apple_notification_center .....	16
<b>8</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



BLE library for the Apple Notification Center



# 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

---

---

**Note:** This library is not available on PyPI yet. Install documentation is included as a standard element. Stay tuned for PyPI availability!

---

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-ble-apple-notification-center
```

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

```
sudo pip3 install adafruit-circuitpython-ble-apple-notification-center
```

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-ble-apple-notification-center
```



## CHAPTER 3

---

Usage Example

---



## 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/ble\_apple\_notification\_center\_simpletest.py

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """
5 This example solicits that apple devices that provide notifications connect to it,
6 ↳initiates
7 ↳pairing, and prints existing notifications.
8 """
9
10 import adafruit_ble
11 from adafruit_ble.advertising.standard import SolicitServicesAdvertisement
12 import adafruit_ble_apple_notification_center as anc
13
14 # PyLint can't find BLERadio for some reason so special case it here.
15 radio = adafruit_ble.BLERadio() # pylint: disable=no-member
16 a = SolicitServicesAdvertisement()
17 a.solicited_services.append(anc.AppleNotificationCenterService)
18 radio.start_advertising(a)
19
20 print("Waiting for connection")
21
22 while not radio.connected:
23     pass
24
25 print("Connected")
26
27 for connection in radio.connections:
```

(continues on next page)

(continued from previous page)

```

27     if ancs.AppleNotificationCenterService not in connection:
28         continue
29
30     if not connection.paired:
31         connection.pair()
32         print("Paired")
33
34     ans = connection[ancs.AppleNotificationCenterService]
35     # Wait for the notifications to load.
36     while len(ans.active_notifications) == 0:
37         pass
38     for notification_id in ans.active_notifications:
39         notification = ans.active_notifications[notification_id]
40         print(notification.app_id, notification.title)

```

## 7.2 adafruit\_ble\_apple\_notification\_center

BLE library for the Apple Notification Center

- Author(s): Scott Shawcroft

### Software and Dependencies:

- Adafruit CircuitPython (5.0.0-beta.2+) firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit’s BLE library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BLE](https://github.com/adafruit/Adafruit_CircuitPython_BLE)

**class** `adafruit_ble_apple_notification_center.AppleNotificationCenterService` (*service=None*)  
Notification service.

**Documented by Apple here:** <https://developer.apple.com/library/archive/documentation/CoreBluetooth/Reference/AppleNotificationCenterServiceSpecification/Specification/Specification.html>

#### **active\_notifications**

A dictionary of active notifications keyed by id.

#### **wait\_for\_new\_notifications** (*timeout=None*)

Waits for new notifications and yields them. Returns on timeout, update, disconnect or clear.

**class** `adafruit_ble_apple_notification_center.Notification` (*notification\_id*,  
*event\_flags*, *category\_id*, *category\_count*,  
*control\_point*, *data\_source*)

One notification that appears in the iOS notification center.

#### **app\_id**

String id of the app that generated the notification. It is not the name of the app. For example, Slack is “com.tinyspeck.chatlyio” and Twitter is “com.atebits.Tweetie2”.

#### **category\_count = None**

Number of other notifications with the same category.

#### **id = None**

Integer id of the notification.

**message**

Message body of the notification. Varies per app.

**message\_size**

Total length of the message string.

**negative\_action = None**

True if the notification has a negative action to respond with. For example, this could be declining a phone call.

**negative\_action\_label**

Human readable label of the negative action.

**positive\_action = None**

True if the notification has a positive action to respond with. For example, this could be answering a phone call.

**positive\_action\_label**

Human readable label of the positive action.

**preexisting = None**

True if the notification existed before we connected to the iOS device.

**removed = None**

True when the notification has been cleared on the iOS device.

**subtitle**

Subtitle of the notification. Varies per app.

**title**

Title of the notification. Varies per app.

**update** (*event\_flags, category\_id, category\_count*)

Update the notification and clear the attribute cache.



## CHAPTER 8

---

### Indices and tables

---

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





**a**

adafruit\_ble\_apple\_notification\_center,

16



## A

active\_notifications

(adafruit\_ble\_apple\_notification\_center.AppleNotificationCenterService attribute), 16

adafruit\_ble\_apple\_notification\_center

(module), 16

app\_id(adafruit\_ble\_apple\_notification\_center.Notification

attribute), 16

AppleNotificationCenterService (class in

adafruit\_ble\_apple\_notification\_center), 16

## C

category\_count(adafruit\_ble\_apple\_notification\_center.Notification attribute), 16

## I

id(adafruit\_ble\_apple\_notification\_center.Notification

attribute), 16

## M

message(adafruit\_ble\_apple\_notification\_center.Notification

attribute), 16

message\_size(adafruit\_ble\_apple\_notification\_center.Notification

attribute), 17

## N

negative\_action(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

negative\_action\_label

(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

Notification (class in

adafruit\_ble\_apple\_notification\_center),

16

## P

positive\_action(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

positive\_action\_label

(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

preexisting(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

## R

removed(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

## S

subtitle(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

## T

title(adafruit\_ble\_apple\_notification\_center.Notification attribute), 17

## U

update() (adafruit\_ble\_apple\_notification\_center.Notification method), 17

## W

wait\_for\_new\_notifications() (adafruit\_ble\_apple\_notification\_center.AppleNotificationCenterService method), 16