

---

# **AdafruitAzureIoT Library Documentation**

*Release 1.0*

**Brent Rubell**

**Jan 14, 2020**



---

## Contents

---

<b>1</b>	<b>Installing from PyPI</b>	<b>3</b>
<b>2</b>	<b>Dependencies</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_azureiot .....	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>



Access to [Microsoft Azure IoT](#) from a CircuitPython device. This library can perform device messaging services (cloud-to-device, device-to-cloud), device services, and job services.



# CHAPTER 1

---

## 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-azureiot
```

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

```
sudo pip3 install adafruit-circuitpython-azureiot
```

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-azureiot
```





## CHAPTER 2

---

### 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 3

---

### Usage Example

---

Create an instance of an Azure IoT Hub (you'll need your SAS Token).

```
my_hub = IOT_HUB(wifi, 'Azure_IOT_Hub_Name', 'Azure_IOT_Hub_SAS-Token', 'Azure_Device_↪Identifier')
```

Send a device-to-cloud message

```
my_hub.send_device_message('Hello Azure IoT!')
```

Enumerate all devices on an Azure IOT Hub

```
hub_devices = my_hub.get_devices()
```

Get information about the current device on an Azure IoT Hub

```
device_info = my_hub.get_device()
```

Get information about the current device's device twin

```
twin_info = my_hub.get_device_twin()
```

Update the current device's device twin properties

```
my_hub.update_device_twin(device_properties)
```



## 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/azureiot\_simpletest.py

```

1  from random import randint
2  import board
3  import busio
4  from digitalio import DigitalInOut
5  from adafruit_esp32spi import adafruit_esp32spi, adafruit_esp32spi_wifimanager
6  import neopixel
7  from adafruit_azureiot import IOT_Hub
8
9  # Get wifi details and more from a secrets.py file
10 try:
11     from secrets import secrets
12 except ImportError:
13     print("WiFi secrets are kept in secrets.py, please add them there!")
14     raise
15
16 # ESP32 Setup
17 try:
18     esp32_cs = DigitalInOut(board.ESP_CS)
19     esp32_ready = DigitalInOut(board.ESP_BUSY)
20     esp32_reset = DigitalInOut(board.ESP_RESET)
21 except AttributeError:
22     esp32_cs = DigitalInOut(board.D9)
23     esp32_ready = DigitalInOut(board.D10)
24     esp32_reset = DigitalInOut(board.D5)
25 spi = busio.SPI(board.SCK, board.MOSI, board.MISO)
26 esp = adafruit_esp32spi.ESP_SPIcontrol(spi, esp32_cs, esp32_ready, esp32_reset)
27 status_light = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.2) # Uncomment for

```

↳ Most Boards

(continues on next page)

(continued from previous page)

```

28 """Uncomment below for ItsyBitsy M4"""
29 #status_light = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.
    ↳2)
30 wifi = adafruit_esp32spi_wifimanager.ESP32SPI_WiFiManager(esp, secrets, status_light)
31
32 # Create an instance of the Azure IoT Hub
33 hub = IOT_Hub(wifi, secrets['azure_iot_hub'], secrets['azure_iot_sas'], secrets[
    ↳'device_id'])
34
35 # Send a Device-to-Cloud message
36 print('Sending Data to Azure IoT Hub...')
37 data = randint(0, 100)
38 hub.send_device_message(str(data))
39 print('Data Sent!')
40
41 # Receive a Cloud-to-Device message
42 # NOTE: HTTP Cloud-to-Device messages are HEAVILY throttled over HTTP.
43 # Microsoft suggests a polling interval of the below code for every 25 minutes.
44 print('Receiving a message from an Azure IoT Hub...')
45 message = hub.get_hub_message()
46 if message is None:
47     print('IoT Hub Message Queue is empty!')
48 else:
49     print(message)

```

## 6.2 adafruit\_azureiot

Microsoft Azure IoT for CircuitPython

- Author(s): Brent Rubell

### 6.2.1 Implementation Notes

#### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BusDevice](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)
- Adafruit's ESP32SPI library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_ESP32SPI](https://github.com/adafruit/Adafruit_CircuitPython_ESP32SPI)

**class** `adafruit_azureiot.IOT_Hub` (*wifi\_manager, iot\_hub\_name, sas\_token, device\_id*)  
 Provides access to a Microsoft Azure IoT Hub. <https://docs.microsoft.com/en-us/rest/api/iothub/>

#### **device\_id**

Returns the current device identifier

#### **get\_device()**

Gets device information from the identity registry of an IoT Hub.

#### **get\_device\_twin()**

Returns the device's device twin information in JSON format.

#### **get\_devices()**

Enumerate devices from the identity registry of the IoT Hub.

**get\_hub\_message** ()

Returns a message from a Microsoft Azure IoT Hub (Cloud-to-Device). Returns None if the message queue is empty. NOTE: HTTP Cloud-to-Device messages are throttled. Poll every 25+ minutes.

**replace\_device\_twin** (*properties*)

Replaces tags and desired properties of a device twin. :param str properties: Device Twin Properties.

**send\_device\_message** (*message*)

Sends a device-to-cloud message. :param string message: Message to send to Azure IoT.

**update\_device\_twin** (*properties*)

Updates tags and desired properties of the device's device twin. :param str properties: Device Twin Properties (<https://docs.microsoft.com/en-us/rest/api/iothub/service/updatetwin#twinproperties>)



## CHAPTER 7

---

### Indices and tables

---

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



**a**

`adafruit_azureiot`, 14





## A

adafruit\_azureiot (*module*), 14

## D

device\_id (*adafruit\_azureiot.IOT\_Hub attribute*), 14

## G

get\_device() (*adafruit\_azureiot.IOT\_Hub method*),  
14

get\_device\_twin() (*adafruit\_azureiot.IOT\_Hub  
method*), 14

get\_devices() (*adafruit\_azureiot.IOT\_Hub  
method*), 14

get\_hub\_message() (*adafruit\_azureiot.IOT\_Hub  
method*), 14

## I

IOT\_Hub (*class in adafruit\_azureiot*), 14

## R

replace\_device\_twin()  
(*adafruit\_azureiot.IOT\_Hub method*), 15

## S

send\_device\_message()  
(*adafruit\_azureiot.IOT\_Hub method*), 15

## U

update\_device\_twin()  
(*adafruit\_azureiot.IOT\_Hub method*), 15