
AdafruitATECC Library Documentation

Release 1.0

Brent Rubell

Jan 14, 2020

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	License	13
7	Table of Contents	15
7.1	Simple test	15
7.2	API	16
8	Indices and tables	17
	Python Module Index	19
	Index	21

Driver for Microchip's ATECCx08 cryptographic co-processors with secure hardware-based key storage.

Note: This library was developed and tested with an ATECC608A, but should work for ATECC508 modules as well.

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)

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-atecc
```

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

```
sudo pip3 install adafruit-circuitpython-atecc
```

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-atecc
```


CHAPTER 3

Usage Example

Examples of using this module are in examples folder.

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](#).

CHAPTER 6

License

This library was written by Arduino SA. We've converted it to work with Adafruit CircuitPython and made changes for it to work with CircuitPython devices and single-board linux computers running CircuitPython libraries. We've added examples to demonstrate using the nonce, random, monotonic counter and SHA256 security functions within the library.

This open source code is licensed under the LGPL License (see LICENSE for details).

7.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/atecc_simpletest.py

```
1 import board
2 import busio
3 from adafruit_atecc.adafruit_atecc import ATECC, _WAKE_CLK_FREQ
4
5 # Initialize the i2c bus
6 i2c = busio.I2C(board.SCL, board.SDA,
7                 frequency=_WAKE_CLK_FREQ)
8
9 # Initialize a new atecc object
10 atecc = ATECC(i2c)
11
12 print("ATECC Serial: ", atecc.serial_number)
13
14 # Generate a random number with a maximum value of 1024
15 print("Random Value: ", atecc.random(rnd_max=1024))
16
17 # Print out the value from one of the ATECC's counters
18 # You should see this counter increase on every time the code.py runs.
19 print("ATECC Counter #1 Value: ", atecc.counter(1, increment_counter=True))
20
21 # Initialize the SHA256 calculation engine
22 atecc.sha_start()
23
24 # Append bytes to the SHA digest
25 print("Appending to the digest...")
26 atecc.sha_update(b"Nobody inspects")
27 print("Appending to the digest...")
```

(continues on next page)

(continued from previous page)

```
28 atecc.sha_update(b" the spammish repetition")
29
30 # Return the digest of the data passed to sha_update
31 message = atecc.sha_digest()
32 print("SHA Digest: ", message)
```

7.2 API

CHAPTER 8

Indices and tables

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

a

[adafruit_atecc](#), 16

A

adafruit_atecc (*module*), 16