
AdafruitespATcontrol Library Documentation

Release 1.0

ladyada

Mar 17, 2020

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_espatcontrol.adafruit_espatcontrol	12
5.2.1	Implementation Notes	12
6	Indices and tables	17
	Python Module Index	19
	Index	21

Use the ESP AT command sent to communicate with the Interwebs. Its slow, but works to get data into CircuitPython

Command set: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf

Examples: https://www.espressif.com/sites/default/files/documentation/4b-esp8266_at_command_examples_en.pdf

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](#).

1.1 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-esp-atcontrol
```

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

```
sudo pip3 install adafruit-circuitpython-esp-atcontrol
```

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-esp-atcontrol
```


CHAPTER 2

Usage Example

See examples folder for full demos

CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 4

Documentation

For information on building library documentation, please check out [this guide](#).

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/esp_atcontrol_simpletest.py

```
1 import time
2 import board
3 import busio
4 from digitalio import DigitalInOut
5 from digitalio import Direction
6 from adafruit_esp8266 import adafruit_esp8266
7
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
17 # With a Particle Argon
18 RX = board.ESP_TX
19 TX = board.ESP_RX
20 resetpin = DigitalInOut(board.ESP_WIFI_EN)
21 rtspin = DigitalInOut(board.ESP_CTS)
22 uart = busio.UART(TX, RX, timeout=0.1)
23 esp_boot = DigitalInOut(board.ESP_BOOT_MODE)
24 esp_boot.direction = Direction.OUTPUT
25 esp_boot.value = True
26
27
```

(continues on next page)

(continued from previous page)

```
28 print("ESP AT commands")
29 esp = adafruit_espatcontrol.ESP_ATcontrol(
30     uart, 115200, reset_pin=resetpin, rts_pin=rtspin, debug=False
31 )
32 print("Resetting ESP module")
33 esp.hard_reset()
34
35 first_pass = True
36 while True:
37     try:
38         if first_pass:
39             print("Scanning for AP's")
40             for ap in esp.scan_APs():
41                 print(ap)
42             print("Checking connection...")
43             # secrets dictionary must contain 'ssid' and 'password' at a minimum
44             print("Connecting...")
45             esp.connect(secrets)
46             print("Connected to AT software version ", esp.version)
47             print("IP address ", esp.local_ip)
48             first_pass = False
49             print("Pinging 8.8.8.8...", end="")
50             print(esp.ping("8.8.8.8"))
51             time.sleep(10)
52         except (ValueError, RuntimeError, adafruit_espatcontrol.OKError) as e:
53             print("Failed to get data, retrying\n", e)
54             print("Resetting ESP module")
55             esp.hard_reset()
56             continue
```

5.2 adafruit_espatcontrol.adafruit_espatcontrol

Use the ESP AT command sent to communicate with the Interwebs. Its slow, but works to get data into CircuitPython

Command set: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf

Examples: https://www.espressif.com/sites/default/files/documentation/4b-esp8266_at_command_examples_en.pdf

- Author(s): ladyada

5.2.1 Implementation Notes

Hardware:

- Adafruit ESP8266 Huzzah Breakout (Product ID: 2471)

Software and Dependencies:

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


```
class adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol (uart, de-
fault_baudrate,
*,
run_baudrate=None,
rts_pin=None,
re-
set_pin=None,
debug=False)
```

A wrapper for AT commands to a connected ESP8266 or ESP32 module to do some very basic internetting. The ESP module must be pre-programmed with AT command firmware, you can use esptool or our CircuitPython miniesptool to upload firmware

at_response (*at_cmd*, *timeout=5*, *retries=3*)

Send an AT command, check that we got an OK response, and then cut out the reply lines to return. We can set a variable timeout (how long we'll wait for response) and how many times to retry before giving up

baudrate

The baudrate of our UART connection

begin ()

Initialize the module by syncing, resetting if necessary, setting up the desired baudrate, turning on single-socket mode, and configuring SSL support. Required before using the module but we dont do in `__init__` because this can throw an exception.

cipmux

The IP socket multiplexing setting. 0 for one socket, 1 for multi-socket

connect (*secrets*)

Repeatedly try to connect to an access point with the details in the passed in 'secrets' dictionary. Be sure 'ssid' and 'password' are defined in the secrets dict! If 'timezone' is set, we'll also configure SNTP

echo (*echo*)

Set AT command echo on or off

factory_reset ()

Perform a hard reset, then send factory restore settings request

get_version ()

Request the AT firmware version string and parse out the version number

hard_reset ()

Perform a hardware reset by toggling the reset pin, if it was defined in the initialization of this object

hw_flow (*flag*)

Turn on HW flow control (if available) on to allow data, or off to stop

is_connected

Initialize module if not done yet, and check if we're connected to an access point, returns True or False

join_AP (*ssid*, *password*)

Try to join an access point by name and password, will return immediately if we're already connected and won't try to reconnect

local_ip

Our local IP address as a dotted-quad string

mode

What mode we're in, can be MODE_STATION, MODE_SOFTAP or MODE_SOFTAPSTATION

nslookup (*host*)

Return a dotted-quad IP address strings that matches the hostname

ping (*host*)

Ping the IP or hostname given, returns ms time or None on failure

remote_AP

The name of the access point we're connected to, as a string

scan_APs (*retries=3*)

Ask the module to scan for access points and return a list of lists with name, RSSI, MAC addresses, etc

sntp_config (*enable, timezone=None, server=None*)

Configure the built in ESP SNTP client with a UTC-offset number (timezone) and server as IP or hostname.

sntp_time

Return a string with time/date information using SNTP, may return 1970 'bad data' on the first few minutes, without warning!

socket_connect (*conntype, remote, remote_port, *, keepalive=10, retries=1*)

Open a socket. conntype can be TYPE_TCP, TYPE_UDP, or TYPE_SSL. Remote can be an IP address or DNS (we'll do the lookup for you. Remote port is integer port on other side. We can't set the local port

socket_disconnect ()

Close any open socket, if there is one

socket_receive (*timeout=5*)

Check for incoming data over the open socket, returns bytes

socket_send (*buffer, timeout=1*)

Send data over the already-opened socket, buffer must be bytes

soft_reset ()

Perform a software reset by AT command. Returns True if we successfully performed, false if failed to reset

status

The IP connection status number (see AT+CIPSTATUS datasheet for meaning)

sync ()

Check if we have AT command sync by sending plain ATs

version

The cached version string retrieved via the AT+GMR command

exception `adafruit_esp8266control.adafruit_esp8266control.OKError`

The exception thrown when we didn't get acknowledgement to an AT command

A 'socket' compatible interface thru the ESP AT command set

```
adafruit_esp8266control.adafruit_esp8266control_socket.getaddrinfo(host, port,  
                                                                    family=0,  
                                                                    socktype=0,  
                                                                    proto=0,  
                                                                    flags=0)
```

Given a hostname and a port name, return a 'socket.getaddrinfo' compatible list of tuples. Honestly, we ignore anything but host & port

```
adafruit_esp8266control.adafruit_esp8266control_socket.set_interface(iface)
```

Helper to set the global internet interface

```
class adafruit_esp8266control.adafruit_esp8266control_socket.socket(family=2,  
                                                                    type=1,  
                                                                    proto=0,  
                                                                    fileno=None)
```

A simplified implementation of the Python 'socket' class, for connecting through an interface to a remote device

close ()

Close the socket, after reading whatever remains

connect (address, conntype=None)

Connect the socket to the 'address' (which should be dotted quad IP). 'conntype' is an extra that may indicate SSL or not, depending on the underlying interface

readline ()

Attempt to return as many bytes as we can up to but not including ' '

recv (num=0)

Read up to 'num' bytes from the socket, this may be buffered internally! If 'num' isnt specified, return everything in the buffer.

send (data)

Send some data to the socket

settimeout (value)

Set the read timeout for sockets, if value is 0 it will block

CHAPTER 6

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

a

adafruit_espatcontrol.adafruit_espatcontrol,
12

adafruit_espatcontrol.adafruit_espatcontrol_socket,
14

A

adafruit_espatcontrol.adafruit_espatcontrol (module), 12

adafruit_espatcontrol.adafruit_espatcontrol_socket (module), 14

at_response() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

B

baudrate (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol attribute), 13

begin() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

C

cipmux (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol attribute), 13

close() (adafruit_espatcontrol.adafruit_espatcontrol_socket.socket method), 15

connect() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

connect() (adafruit_espatcontrol.adafruit_espatcontrol_socket.socket method), 15

E

echo() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

ESP_ATcontrol (class in adafruit_espatcontrol.adafruit_espatcontrol), 12

F

factory_reset() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

G

get_version() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

H

getaddrinfo() (in module adafruit_espatcontrol.adafruit_espatcontrol_socket), 14

hard_reset() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

hw_flow() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

J

join_AP() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

L

local_ip (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol attribute), 13

M

mode (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol attribute), 13

N

nslookup() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 13

O

OKError, 14

P

ping() (adafruit_espatcontrol.adafruit_espatcontrol.ESP_ATcontrol method), 14

R

readline() (adafruit_espatcontrol.adafruit_espatcontrol_socket.socket method), 15

`recv()` (*adafruit_espcontrol.adafruit_espcontrol_socket.socket*
method), 15
`remote_AP` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
attribute), 14

S

`scan_APs()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`send()` (*adafruit_espcontrol.adafruit_espcontrol_socket.socket*
method), 15
`set_interface()` (*in module*
adafruit_espcontrol.adafruit_espcontrol_socket),
14
`settimeout()` (*adafruit_espcontrol.adafruit_espcontrol_socket.socket*
method), 15
`sntp_config()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`sntp_time` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
attribute), 14
`socket` (*class in adafruit_espcontrol.adafruit_espcontrol_socket*),
14
`socket_connect()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`socket_disconnect()`
(*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`socket_receive()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`socket_send()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`soft_reset()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14
`status` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
attribute), 14
`sync()` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
method), 14

V

`version` (*adafruit_espcontrol.adafruit_espcontrol.ESP_ATcontrol*
attribute), 14