Package Index > **bleep** > **0.4.3**

bleep 0.4.3

Bluetooth Low Energy (BLE) Library for Python

Download
bleep-0.4.3.tar.gz

# bleep

A BLE abstraction layer for Python inspired by [bleat](**https://github.com/thegecko/bleat**). Currently only supports Linux, with experimental support for Mac OS X.

## Current Support

- Discovering devices
- Reading advertising data
- Connecting to devices
- Discovering services, characteristics and descriptors
- Read from characteristics

## Installation

### Linux

First, install my fork of pygattlib and its dependencies:

```
`bash sudo apt-get install libboost-python-dev libboost-thread-dev
libbluetooth-dev libglib2.0-dev python-dev `
```

You should also make sure that your version of libbluetooth is at least 4.101:

```
`bash apt-cache policy libbluetooth-dev | grep Installed `
```

Then, clone the repository, and install the python package.

```
`bash git clone https://github.com/matthewelse/pygattlib.git cd
pygattlib sudo python setup.py install `
```

This will build the dynamic library, and install the python package.

You can then install bleep easily:

```
`bash sudo pip install bleep `
```

If you want to develop bleep, instead of the last line, run:

```
` sudo python setup.py develop `
```

This will cause any changes you make to bleep to be reflected when you import the library.

> NOTE: You may need to run all BLE code with *sudo*, even when using the Python interactive shell.

### Mac OS X

Installation on Mac OS X is very simple:

```
`bash git clone https://github.com/matthewelse/bleep.git cd bleep sudo
python setup.py install `
```

Likewise, if you would like to develop bleep, run this instead of the last line:

```
`bash sudo python setup.py develop `
```

## Examples

### tree.py

You can run tree.py to see all of the services, characteristics and descriptors attached to a device with a specific mac address. In order to find the device’s mac address, you could use *hcitool lescan*, or use *BLEDevice.discoverDevices()*.

```
` usage: tree.py [-h] mac `
```

## Usage

### Include bleep

```
`python >>> from bleep import BLEDevice `
```

### Scan for devices

```
`python >>> devices = BLEDevice.discoverDevices() >>> devices [Device
Name:  (5A:79:8E:91:83:1C), Device Name:  (C1:20:68:1B:00:26), Device
Name: BLE Keyboard (C9:E8:56:3B:4D:B1), Device Name:
(4C:25:F5:C2:E6:61), Device Name:  (60:03:08:B2:47:F1), Device Name:
(C1:62:3A:1D:00:14)] `
```

This will return a list of Device objects, however you won’t be connected to any of them, so pick one you like, and connect to it:

```
`python >>> device = devices[2] >>> device.connect() `
```

You can then access the device’s services:

```
`python >>> device.services [Generic Access, Generic Attribute, Device
Information, Battery Service, Human Interface Device] `
```

each service’s characteristics

```
`python >>> service = device.services[4] >>> service Human Interface
Device >>> service.characteristics [HID Information, Report Map,
Protocol Mode, HID Control Point, Report, Report] `
```

and each characteristic’s descriptors

```
`python >>> char = service.characteristics[4] >>> char Report >>>
char.descriptors [Client Characteristic Configuration, Report Reference]
`
```

### Useful Functionality

BLEDevice.discoverDevices supports parameters which allow you to specify which BLE device to connect to (ignored on OSes other than Linux), how long to sample for, as well as a function which returns a boolean value, allowing you to cherry-pick your devices.

```
`python def discoverDevices(device='hci0', timeout=5, filter=lambda x:
True) `
```

File	Type	Py Version	Uploaded on	Size
bleep-0.4.3.tar.gz (md5)	Source		2015-09-11	18KB

Author: Matthew Else
Home Page: https://github.com/matthewelse/bleep
Keywords: ble
License: Apache-2.0
Package Index Owner: melse
DOAP record: bleep-0.4.3.xml