pymochad Documentation

Release

Matthew Treinish

Nov 15, 2017

Contents

1 pymochad **3**
1.1 Usage 3

2 PyMochad API **5**
2.1 The PyMochad Controller Class 5
2.2 The PyMochad Device Class 6

Python Module Index **7**

Contents:

CHAPTER 1

pymochad

A python library for sending commands to the mochad TCP gateway daemon for the X10 CMA15A controller:

https://sourceforge.net/projects/mochad/

Complete documentation is here: http://pymochad.readthedocs.io/en/latest/

1.1 Usage

Using PyMochad is pretty straightforward you just need to init a PyMochad object and then issue commands to it. For example:

```
from pymochad import controller

mochad = remote.PyMochad()
print(mochad.status())
```

will connect to a running mochad instance (running on your localhost) and print the device status.

You can also send a command directly to a device using a device class. For example:

```
from pymochad import controller
from pymochad import device

mochad = controller.PyMochad()
light_switch = device.Device(mochad, 'a1')
light_switch.send_cmd('on')
```

will connect to a running a mochad instance and send the *on* command to the light switch device at address *a1* on the power line interface.

For a complete API documentation see: *PyMochad API*.

CHAPTER 2

PyMochad API

2.1 The PyMochad Controller Class

This is used to interact with the X10 controller directly using the mochad socket

class `pymochad.controller.`**`PyMochad`**(*server=None*, *port=1099*)

Bases: `object`

PyMochad controller class

This class is used to create a PyMochad controller object that is used to send commands to a running PyMochad daemon.

Parameters

- **`server`** (*`str`*) – The host to connect to the pymochad socket on, it defaults to localhost
- **`port`** (*`int`*) – The port to use for remote connections. If one is not provided it will just use the default port of 1099.

`read_data`()

Read data from mochad

Return data The data returned over the mochad socket

Return type str

`reconnect`()

Reconnect when mochad server is restarted/lost connection.

`send_cmd`(*cmd*)

Send a raw command to mochad.

Parameters **`cmd`** (*`str`*) – The command to send to mochad

`status`()

Send a show device status command.

Return status The status of device including RF security devices

Return type str

2.2 The PyMochad Device Class

This is used to interact with an X10 device

class `pymochad.device.`**`Device`**(*controller*, *address*, *comm_type='pl'*)

Bases: `object`

PyMochad Device class

This class represents an X1 device connected to your controller

Parameters

- **`controller`** (`PyMochad`) – A PyMochad controller object for the device to use
- **`address`** (*`str`*) – The device address
- **`comm_type`** (*`str`*) – The communication type to use for the device. This is either pl (for power line) or rf (for radio frequency)

`get_status`()

Get the on/off status for the devices

Returns Device status

Return type str

`get_statussec`()

Get the on/off status for the X10 Security devices

Returns Device status

Return type str

`send_cmd`(*cmd*)

Send a raw command to device.

Parameters **`cmd`** (*`str`*) – The command to send to the device

Python Module Index

p

pymochad.controller, 5
pymochad.device, 6

Index

D

Device (class in pymochad.device), 6

G

get_status() (pymochad.device.Device method), 6
get_statussec() (pymochad.device.Device method), 6

P

PyMochad (class in pymochad.controller), 5
pymochad.controller (module), 5
pymochad.device (module), 6

R

read_data() (pymochad.controller.PyMochad method), 5
reconnect() (pymochad.controller.PyMochad method), 5

S

send_cmd() (pymochad.controller.PyMochad method), 5
send_cmd() (pymochad.device.Device method), 6
status() (pymochad.controller.PyMochad method), 5