

✓

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

<>

Python Library to control TPLink Switch (HS100 / HS110)

📌 75 commits

🌿 1 branch

📦 7 releases

👥 10 contributors

Branch: master ▾

New pull request

Find file

Clone or download ▾

🏠 rytilahti committed on GitHub add lb100 tests, thanks to kdschlosser (issue #58) (#82)		Latest commit 8f83ca8 6 days ago
📁 pyHS100	add lb100 tests, thanks to kdschlosser (issue #58) (#82)	6 days ago
📄 .gitchangelog.rc	Add changelog & add .gitchangelog.rc (#28)	5 months ago
📄 .gitignore	Add check to ensure devices with lat/lon with `_j` suffix are support...	4 months ago
📄 .hound.yml	set up hound-ci	9 months ago
📄 .travis.yml	Refactor and drop py2 support (#49)	5 months ago
📄 CHANGELOG	add future requirement (#47)	5 months ago
📄 HOWTO_RELEASE	Release 0.2.4 preparations (#43)	5 months ago
📄 LICENSE	Update LICENSE	10 months ago
📄 README.md	Release 0.2.4 preparations (#43)	5 months ago
📄 requirements.txt	Add new client tool (#42)	5 months ago
📄 setup.py	Refactor and drop py2 support (#49)	5 months ago
📄 tox.ini	make tox run pytest-cov, add coveralls (#81)	19 days ago

📖 README.md

pyHS100

Python Library to control TPLink smart plugs (HS100, HS105, HS110, HS200) and TPLink smart bulbs (LB1xx).

Usage

The package is shipped with a console tool named pyhs100, please refer to `pyhs100 --help` for detailed usage. **Note: The tool does not currently support bulb-specific commands, please feel free to prepare a pull request!**

Discovering devices

```
$ pyhs100 discover

Discovering devices for 5 seconds
Found device: {'ip': '192.168.250.186',
               'port': 9999,
               'sys_info': {'emeter': {'get_realtime': {'current': 0.013309,
```

```
<snip>
```

Querying the state

```
$ pyhs100 --ip 192.168.250.186

== My Smart Plug - HS110(EU) ==
Device state: OFF
LED state:      False
Time:          1970-01-01 01:52:35
On since:      2017-03-19 17:09:16.408657
Hardware:      1.0
Software:      1.0.8 Build 151101 Rel.24452
MAC (rssi):    50:C7:BF:XX:XX:XX (-61)
Location:      {'longitude': XXXX, 'latitude': XXXX}
== Emeter ==
Current state: {'power': 0, 'total': 0.001, 'current': 0.013552, 'voltage': 223.394238}
```

Library usage

For all available API functions run `help(SmartPlug)` or `help(SmartBulb)` .

```
from pyHS100 import SmartPlug, SmartBulb
from pprint import pformat as pf

plug = SmartPlug("192.168.250.186")
print("Alias, type and supported features: %s" % (plug.identify(),))
print("Hardware: %s" % pf(plug.hw_info))
print("Full sysinfo: %s" % pf(plug.get_sysinfo())) # this prints lots of information about the device
```

Time information

```
print("Current time: %s" % plug.time)
print("Timezone: %s" % plug.timezone)
```

Getting and setting the name

```
print("Alias: %s" % plug.alias)
plug.alias = "My New Smartplug"
```

State & switching

```
print("Current state: %s" % plug.state)
plug.turn_off()
plug.turn_on()
```

or

```
plug.state = "ON"
plug.state = "OFF"
```

Getting emeter status (on HS110)

```
print("Current consumption: %s" % plug.get_emeter_realtime())
```

```
print("Per day: %s" % plug.get_emeter_daily(year=2016, month=12))
print("Per month: %s" % plug.get_emeter_monthly(year=2016))
```

Switching the led

```
print("Current LED state: %s" % plug.led)
plug.led = False # turn off led
print("New LED state: %s" % plug.led)
```