

PyVantagePro Documentation

Release 0.3.3dev

Salem Harrache & Lionel Darras

Jan 11, 2018

build unknown

1 About

This project was conducted by the laboratory [ISTerre Grenoble](#) (FR) as part of the development of an observation network [OMIV](#) (Observatoire Multi-disciplinaires des Instabilités de Versants)

2 Description

PyVantagePro is a python project which aims to make the communication with weather stations Davis VantagePro2 easier and funnier... i.e. more pythonic.

The main feature of this project is to get data automatically. In order to do so, it uses the basic methods `get_archives()` (to get archive data) and `get_current_data()` (to get real-time data).

About configuration, it only uses `gettime()` and `settime()` because we are assuming that stations are already configured.

Note: PyVantagePro uses the [PyLink](#) lib, offers a universal communication interface with File-Like API.

Examples:

```
>>> from pyvantagepro import VantagePro2
>>>
>>> device = VantagePro2.from_url('tcp:host-ip:port')
>>> device.gettime()
2012-06-13 16:44:56
>>> data = device.get_current_data()
>>> data['TempIn']
87.3
>>> data.raw
4C 4F 4F ... 0D E6 3B
>>> data.filter(('TempIn', 'TempOut', 'SunRise', 'SunSet')).to_csv()
TempIn,TempOut,SunRise,SunSet
87.3,71.5,03:50,19:25
```

3 Features

- Collecting real-time data as a python dictionary
- Collecting archives as a list of dictionaries
- Collecting data in a CSV file
- Updating station time
- Getting some information about the station, such as date and firmware version.
- Various types of connections are supported
- Comes with a command-line script
- Compatible with Python 2.6+ and 3.x

4 Installation

You can install, upgrade, uninstall PyVantagePro with these commands:

```
$ pip install pyvantagepro
$ pip install --upgrade pyvantagepro
$ pip uninstall pyvantagepro
```

Or if you don't have pip:

```
$ easy_install pyvantagepro
```

Or you can get the [source code from github](#).

5 Command-line usage

PyVantagePro has a command-line script that interacts with the station.:

```
$ pyvantagepro -h

usage: pyvantagepro [-h] [--version]
                  {gettime, settime, getinfo, getarchives, getdata, update} ...

Extract data from VantagePro 2 station

optional arguments:
  -h, --help            show this help message and exit
  --version             Print PyVPDriver's version number and exit.

The pyvantagepro commands:
  {gettime, settime, getinfo, getarchives, getdata, update}
  gettime              Print the current date on the station.
  settime              Set the datetime argument on the station.
  getinfo              Print VantagePro information.
  getarchives          Extract archives data from the station between
                      `start` date and `stop` date. By default the
                      entire contents of the data archive will be
                      downloaded.
```

getdata	Extract real-time data from the station.
update	Update csv database records with getting automatically new records.

5.1 Gettime

The *gettime* command gives, as its name suggests, the current datetime of the station (with the timezone offset).

Usage:

```
pyvantagepro gettime [-h] [--timeout TIMEOUT] [--debug] url

positional arguments:
  url                    Specify URL for connection link.
                        E.g. tcp:iphost:port or serial:/dev/ttyUSB0:19200:8N1

optional arguments:
  -h, --help            show this help message and exit
  --timeout TIMEOUT    Connection link timeout
  --debug              Display log
```

Example:

```
$ pyvantagepro gettime tcp:192.168.0.18:1111
2012-06-12 15:14:23 - Localtime
```

5.2 Settime

Allows us to update the station date and time(no timezone for now).

Usage:

```
pyvantagepro settime [-h] [--timeout TIMEOUT] [--debug] url datetime

Set the given datetime argument on the station

positional arguments:
  url                    Specify URL for connection link.
                        E.g. tcp:iphost:port or serial:/dev/ttyUSB0:19200:8N1
  datetime              The chosen datetime value. (like : "2012-06-12 17:31")

optional arguments:
  -h, --help            show this help message and exit
  --timeout TIMEOUT    Connection link timeout
  --debug              Display log
```

Example:

```
$ pyvantagepro settime tcp:192.168.0.18:1111 "2012-06-12 17:32"
Old value : 2012-06-12 16:24:15 - Localtime
New value : 2012-06-12 17:32:02 - Localtime
```

5.3 Getinfo

Gives some information about the station, such as date and firmware version.

Usage:

```
pyvantagepro getinfo [-h] [--timeout TIMEOUT] [--debug] url

Print VantagePro 2 information.

positional arguments:
  url                Specifiy URL for connection link.
                    E.g. tcp:iphost:port or serial:/dev/ttyUSB0:19200:8N1

optional arguments:
  -h, --help        show this help message and exit
  --timeout TIMEOUT Connection link timeout
  --debug           Display log
```

Example:

```
$ pyvantagepro getinfo tcp:192.168.0.18:1111 --timeout 2
Firmware date : 2009-11-27
Firmware version : 1.90
Diagnostics : {'max_received': 21211, 'crc_errors': 0,
'total_missed': 0, 'total_received': 21211, 'resyn': 0}
```

5.4 Getarchives

Downloads the archive records from the station between two dates. By default all records are downloaded. If no stop date is specified, the download will stop at the last record available in the station memory.

Usage:

```
pyvantagepro getarchives [-h] [--timeout TIMEOUT] [--debug]
                        [--output OUTPUT] [--start START]
                        [--stop STOP] [--delim DELIM]
                        url

Extract the archive records from the station between `start` date
and `stop` date. By default the entire contents of the data archive
will be downloaded.

positional arguments:
  url                Specifiy URL for connection link.
                    E.g. tcp:iphost:port or serial:/dev/ttyUSB0:19200:8N1

optional arguments:
  -h, --help        Show this help message and exit
  --timeout TIMEOUT Connection link timeout
  --debug           Display log
  --output OUTPUT   Filename where output is written
  --start START     The beginning date record. (like : "2012-06-12 17:36")
  --stop STOP       The stopping date record. (like : "2012-06-12 17:36")
  --delim DELIM     CSV char delimiter
```

Example:

```
$ pyvantagepro getarchives tcp:192.168.0.18:1111 \  
--start "2012-06-12 16:19" --stop "2012-06-12 16:21" \  
--output archive.csv  
Archives download: 100% |#####|  
1 record was found
```

If you want to get all records, you can use this command without specifying any date

```
$ pyvantagepro getarchives tcp:192.168.0.18:1111 --output archive.csv  
Archives download: 100% |#####|  
2145 records were found
```

5.5 Update

This command is useful for maintaining a database which is updated regularly (e.g. with a crontab). The database is a simple CSV file that contains all archive records. You can use 'getarchives' with a specific date range and manually update your data. However, the update command automatically analyzes the CSV file, retrieves the datetime of the last record, then downloads the data from the station and add it to the end of the file.

Finally we have a CSV file with all the data which is updated automatically.

Usage:

```
pyvantagepro update [-h] [--timeout TIMEOUT] [--debug] [--delim DELIM]  
                    url db  
  
Update CSV database records by getting automatically new archive records.  
  
positional arguments:  
  url          Specifiy URL for connection link.  
              E.g. tcp:iphost:port or serial:/dev/ttyUSB0:19200:8N1  
  db          The CSV database  
  
optional arguments:  
  -h, --help    Show this help message and exit  
  --timeout TIMEOUT Connection link timeout  
  --debug       Display log  
  --delim DELIM CSV char delimiter
```

Example:

If the file does not exist, it will be created automatically:

```
$ pyvantagepro update tcp:192.168.0.18:1111 ./database.csv --timeout 2  
Archives download: 100% |#####|  
2145 new records
```

again...

```
$ pyvantagepro update tcp:192.168.0.18:1111 ./database.csv --timeout 2  
Archives download: 100% |#####|  
No new records were found
```

5.6 Debug mode

You can use debug option if you want to print log and see the flowing data:

```
$ pyvantagepro settime tcp:192.168.0.18:1111 "2012-06-12 16:24" --debug
2012-06-12 17:24:45,311 INFO: new <TCPLink tcp:127.0.0.1:1111> was initialized
2012-06-12 17:24:45,311 INFO: try wake up console
2012-06-12 17:24:45,311 INFO: Write : <u'\n'>
2012-06-12 17:24:45,412 INFO: Read : <0A 0D>
2012-06-12 17:24:45,412 INFO: Check ACK: OK ('\n\r')
2012-06-12 17:24:45,413 INFO: try send : VER
2012-06-12 17:24:45,413 INFO: Write : <u'VER\n'>
2012-06-12 17:24:45,514 INFO: Read : <0A 0D 4F 4B 0A 0D>
2012-06-12 17:24:45,514 INFO: Check ACK: OK ('\n\rOK\n\r')
2012-06-12 17:24:45,515 INFO: Read : <41 70 72 20 31 30 20 32 30 30 36 0A 0D>
2012-06-12 17:24:45,521 INFO: try wake up console
2012-06-12 17:24:45,521 INFO: Write : <u'\n'>
```

6 API reference

class `pyvantagepro.VantagePro2` (*link*)

Communicates with the station by sending commands, reads the binary data and parsing it into usable scalar values.

Parameters `link` – A *PyLink* connection.

wake_up ()

Wakeup the station console.

send (*data*, *wait_ack=None*, *timeout=None*)

Sends data to station.

Parameters

- **data** – Can be a byte array or an ASCII command. If this is the case for an ascii command, a <LF> will be added.
- **wait_ack** – If *wait_ack* is not None, the function must check that acknowledgement is the one expected.
- **timeout** – Define this timeout when reading ACK from link.

read_from_eeprom (*hex_address*, *size*)

Reads from EEPROM the *size* number of bytes starting at the *hex_address*. Results are given as hex strings.

archive_period

Returns number of minutes in the archive period.

diagnostics

Return the Console Diagnostics report. (RXCHECK command)

firmware_date

Return the firmware date code

firmware_version

Returns the firmware version as string

classmethod **from_url** (*url*, *timeout=10*)

Get device from url.

Parameters

- **url** – A *PyLink* connection URL.

- **timeout** – Set a read timeout value.

get_archives (*start_date=None, stop_date=None*)

Get archive records until *start_date* and *stop_date* as ListDict.

Parameters

- **start_date** – The beginning datetime record.
- **stop_date** – The stopping datetime record.

get_current_data ()

Returns the real-time data as a *Dict*.

gettime ()

Returns the current datetime of the console.

settime (*dtime*)

Set the given *dtime* on the station.

timezone

Returns timezone offset as string.

class `pyvantagepro.utils.Dict` (**args, **kws*)

A dict with some additional methods.

filter (*keys*)

Create a dict with only the following *keys*.

```
>>> mydict = Dict({"name": "foo", "firstname": "bar", "age": 1})
>>> mydict.filter(['age', 'name'])
{'age': 1, 'name': 'foo'}
```

to_csv (*delimiter='u', ', header=True*)

Serialize list of dictionaries to csv.

class `pyvantagepro.utils.ListDict`

List of dicts with some additional methods.

filter (*keys*)

Create a list of dictionaries with only the following *keys*.

```
>>> mylist = ListDict([{"name": "foo", "age": 31},
...                    {"name": "bar", "age": 24}])
>>> mylist.filter(['name'])
[{'name': 'foo'}, {'name': 'bar'}]
```

sorted_by (*keyword, reverse=False*)

Returns list sorted by *keyword*.

to_csv (*delimiter='u', ', header=True*)

Serialize list of dictionaries to csv.

exception `pyvantagepro.device.NoDeviceException`

Can not access weather station.

exception `pyvantagepro.device.BadAckException`

No valid acknowledgement.

exception `pyvantagepro.device.BadCRCException`

No valid checksum.

exception `pyvantagepro.device.BadDataException`
No valid data.

7 Feedback & Contribute

Your feedback is more than welcome. Write email to the [PyVantagePro mailing list](#).

There are several ways to contribute to the project:

1. Post bugs and feature [requests on github](#).
2. Fork [the repository](#) on Github to start making your changes.
3. Write a test which shows that the bug was fixed or that the feature works as expected.
4. Send a pull request and bug the maintainer until it gets merged and published. :) Make sure to add yourself to [AUTHORS](#).

8 Changelog

8.1 Version 0.3.2

Released on 2014-02-02.

- Added python3.3 and pypy support
- Use detox for parallel multienv tests
- Corrected WindAvgDir and WindHiDir
- Added one byte shift if wake-up is not working

8.2 Version 0.3.1

Released on 2012-06-28.

- remove duplicate records
- sort records by Datetime field

8.3 Version 0.3

Released on 2012-06-26.

- Use ordereddict to order data fields (Datetime field first)
- Fix a bug related to timeout
- Set timeout to 10 sec by default

8.4 Version 0.2

Released on 2012-06-20.

- Remove blist from requirements
- Minor bug fixes

8.5 Version 0.1

Released on 2012-06-14.

- First properly tagged release.
- Support VantagePro2 revB only.
- Parsing binary data into dict and list of dict.
- Command-line script.

Index

A

archive_period (pyvantagepro.VantagePro2 attribute), 6

B

BadAckException, 7

BadCRCEXception, 7

BadDataException, 7

D

diagnostics (pyvantagepro.VantagePro2 attribute), 6

Dict (class in pyvantagepro.utils), 7

F

filter() (pyvantagepro.utils.Dict method), 7

filter() (pyvantagepro.utils.ListDict method), 7

firmware_date (pyvantagepro.VantagePro2 attribute), 6

firmware_version (pyvantagepro.VantagePro2 attribute),
6

from_url() (pyvantagepro.VantagePro2 class method), 6

G

get_archives() (pyvantagepro.VantagePro2 method), 7

get_current_data() (pyvantagepro.VantagePro2 method),
7

gettime() (pyvantagepro.VantagePro2 method), 7

L

ListDict (class in pyvantagepro.utils), 7

N

NoDeviceException, 7

P

pyvantagepro (module), 1

R

read_from_eeprom() (pyvantagepro.VantagePro2
method), 6

S

send() (pyvantagepro.VantagePro2 method), 6

settime() (pyvantagepro.VantagePro2 method), 7

sorted_by() (pyvantagepro.utils.ListDict method), 7

T

timezone (pyvantagepro.VantagePro2 attribute), 7

to_csv() (pyvantagepro.utils.Dict method), 7

to_csv() (pyvantagepro.utils.ListDict method), 7

V

VantagePro2 (class in pyvantagepro), 6

W

wake_up() (pyvantagepro.VantagePro2 method), 6