pysds011

Release 0.0.3

Michele Pagot

Jan 19, 2021

CONTENTS:

CHAPTER

ONE

OVERVIEW

Simple python driver for SDS011 PM sensor from Nova.

- Free software: MIT license

1.1 Installation

```
pip install pysds011
```

You can also install the in-development version with:

```
pip install https://github.com/michelepagot/pysds011/archive/master.zip
```

1.2 Usage

Package has a class interface

```
log = logging.getLogger(__name__)
ser = serial.Serial('/dev/ttyUSB0', 9600)
ser.open()
sd = driver.SDS011(ser, log)
sd.cmd_set_sleep(0)
sd.cmd_set_mode(sd.MODE_QUERY)
sd.cmd_firmware_ver()
```

Package is also provided with a reference cli application:

```
pysds011.exe --port COM42 version
>> 21.2.2223
```

1.3 Documentation

This internal package documentation is available at https://pysds011.readthedocs.io/ Some other interesting reading are: * SDS011 datasheet http://cl.ly/ekot * This project is inspired by https://gist.github.com/kadamski/92653913a53baf9dd1a8

1.4 Development

This repo just decide to adopt git flow strategy . . .

CHAPTER

TWO

INSTALLATION

At the command line:

```
pip install pysds011
```

CHAPTER

THREE

USAGE

3.1 Use the package in a python script

To use pysds011 in a project, start by importing this package:

```
import pysds011
```

Provide an UART channel, user is in charge to also `open/close` it:

```
import serial

ser = serial.Serial('/dev/ttyUSB0', 9600)
ser.open()
```

Provide a logger:

```
import logging

log = logging.getLogger(__name__)
```

Now create a driver instance, injecting serial and logging:

```
sd = driver.SDS011(ser, log)
```

Usually the first step is to wake up the sensor:

```
sd.cmd_set_sleep(0)
```

Then start to interact with it:

```
sd.cmd_set_mode(sd.MODE_QUERY)
fw_ver = sd.cmd_firmware_ver()
dust_data = sd.cmd_query_data()
```

3.2 Use the command line tool

This package is provided with a command line tool to be able to immidiately start playing with your sensor Command line is named `pysds011`

First stop should be the embedded help. Here just an outdated version of it:

```
pysds011.exe --help

Usage: pysds011 [OPTIONS] COMMAND [ARGS]...
  pysds011 cli app entry point

Options:
  --port TEXT          UART port to communicate with dust sensor.
  -v, --verbosity LVL  Either CRITICAL, ERROR, WARNING, INFO or DEBUG
  --help               Show this message and exit.

Commands:
  dust         Get dust value
  fw-version   Get SDS011 FW version
  help         Get specific help of a command
  sleep        Set sleep MODE 1:sleep 0:wakeup
```

And each command has its own help:

```
pysds011.exe help dust

Usage: pysds011 help [OPTIONS]

Get dust value

Options:
  --warmup INTEGER  Time in sec to warm up the sensor
  --format TEXT     result format (PRETTY|JSON|PM2.5|PM10)
  --help            Show this message and exit.
```

Nova SDS011 sensor is connected to your machine through UART, so to read the actual dust value, you need to provide a **port** value:

```
pysds011.exe --port COM4 dust

    PM 2.5: 25.9 g/m^3  PM 10: 62.4 g/m^3 CRC=OK
```

Dust value can be presented in **multiple format**:

- PRETTY (default)
- JSON:

  ```
  pysds011.exe --port COM4 dust --format JSON
  {'pm25': 28.4, 'pm10': 118.6, 'pretty': 'PM 2.5: 28.4 g/m^3  PM 10: 118.6 g/m^3␣
  ↪CRC=OK'}
  ```

- Single PM:

  ```
  pysds011.exe --port COM4 dust --format PM2.5
  26.0
  ```

(continues on next page)

(continued from previous page)

```
pysds011.exe --port COM4 dust --format PM10
99.0
```

Read the dust sensor FW version:

```
pysds011.exe --port COM4 fw-version

FW version Y: 18, M: 11, D: 16, ID: 0xe748, CRC=OK
```

CHAPTER

FOUR

API DOCUMENTATION

pysds011	
pysds011.driver	
pysds011.cli	Module that contains the command line app.

4.1 Driver API

Low level layer that is in charge to manage communication with the sensor. This is the most reusable part of this package

class pysds011.driver.**SDS011**(*ser*, *log*)

Main driver class

cmd_firmware_ver(*id=b'\xff\xff'*)

Get FW version

Returns version description dictionary

Return type dict

cmd_get_mode(*id=b'\xff\xff'*)

Get active reporting mode

Parameters **id** (*2 bytes, optional*) – sensor id to request mode, defaults to b'ÿÿ' that is 'all'

Returns mode if it is ok, None if error

Return type int

cmd_get_sleep()

Get active sleep mode

Returns True if it is sleeping

Return type bool

cmd_get_working_period(*id=b'\xff\xff'*)

Get current working period

Returns working period in minutes: work 30 seconds and sleep n*60-30 seconds

Return type int

cmd_set_id(*id*, *new_id*)

Set a device ID to a specific sensor

Parameters

- **id** (*2 bytes*) – ID of sensor that need a new ID (FF FF is in theory allowed too, but be carefull)
- **new_id** (*2 bytes*) – new ID to be assigned

Returns operation result

Return type bool

cmd_set_mode(*mode=1*, *id=b'\xff\xff'*)

Set data reporting mode. The setting is still effective after power off

Parameters

- **mode** (*int, optional*) – 0report active mode 1Report query mode, defaults to 1
- **id** (*2 bytes, optional*) – sensor id to request mode, defaults to b'ÿÿ' that is 'all'

Returns True is set is ok

Return type bool

cmd_set_sleep(*sleep=1*, *id=b'\xff\xff'*)

Set sleep mode

Parameters

- **sleep** (*int, optional*) – 1:enable sleep mode, 0:wakeup, defaults to 1
- **id** (*2 bytes, optional*) – sensor id to request mode, defaults to b'ÿÿ' that is 'all'

Returns True is set is ok

Return type bool

cmd_set_working_period(*period=0*, *id=b'\xff\xff'*)

Set working period The setting is still effective after power off, factory default is continuous measurement. The sensor works periodically and reports the latest data.

Parameters **period** (*int*) – 0continuous(default), 1-30 minute work 30 seconds and sleep n*60-30 seconds

Returns result

Return type bool

4.2 CLI app API

Module that contains the command line app.

CHAPTER

FIVE

AUTHORS

- Michele Pagot - https://michelepagot.github.io/

CHAPTER

SIX

CHANGELOG

6.1 0.0.3 (2021-1-8)

- cli get subcommands and produce meaningful results
- robustness about error handling
- Testing and documentation

6.2 0.0.2 (2021-1-4)

- Improved cli (first functional).

6.3 0.0.1 (2020-12-19)

- First release on PyPI.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

C

M

P

S