
pyseneye Documentation

Release 0.0.1

Author

Feb 26, 2019

Contents:

1	pyseneye package	1
1.1	Submodules	1
1.2	pyseneye.sud module	1
1.3	Module contents	5
2	Indices and tables	7
	Python Module Index	9

1.1 Submodules

1.2 pyseneye.sud module

pyseneye.sud implements the HID interface for the Seneye USB devices.

```
class pyseneye.sud.Action
    Bases: enum.Enum

    Actions that can be passed to SUDevice.action().

    ENTER_INTERACTIVE_MODE = 1
    LEAVE_INTERACTIVE_MODE = 2
    LIGHT_READING = 3
    SENSOR_READING = 0

class pyseneye.sud.ActionDefinition(cmd_str, rdefs)
    Bases: object

    Definition for action and expected responses.

    cmd_str
        Command string to write to the device.

        Returns command string
        Return type str

    read_definitions
        Read definition for expected response.

        Returns the read definitions
        Return type ReadDefinition[]
```

```
class pyseneye.sud.BaseResponse (raw_data, read_def)
    Bases: abc.ABC

    Abstract class for the SUD responses.

    validation_bytes
        Bytes that are used to validate the message is correct.

        Returns bytes used for validation

        Return type array('B', [2])

class pyseneye.sud.DeviceType
    Bases: enum.Enum

    Different type of sensor devices.

    HOME = 0
    POND = 1
    REEF = 3

class pyseneye.sud.EnterInteractiveResponse (raw_data, read_def)
    Bases: pyseneye.sud.Response

    Received when entering interactive mode. Contains device metadata.

    device_type
        Get the device type.

        Returns the device type

        Return type DeviceType

    version
        Firmware version of the device.

        Returns the version

        Return type str

class pyseneye.sud.ReadDefinition (parse_str, validator, return_values, return_type)
    Bases: object

    Definition of expected response, including validation and parsing.

    parse_str
        Parse string, as struct format string.

        Returns format string

        Return type str

    return_type
        Subclass of BaseResponse, the expected response object.

        Returns expected response object

        Return type BaseResponse subclass

    return_values
        Comma separate list of expected return value names.

        Returns comma separated list

        Return type str
```

validator

Bytes that are used for validation of expected read.

Returns validation bytes

Return type array('B', [2])

class pyseneye.sud.**Response**(*raw_data*, *read_def*)

Bases: [pyseneye.sud.BaseResponse](#)

Response object, includes ACK status.

ack

Acknowledgment result.

Returns True was process successfully, False if not

Return type bool

class pyseneye.sud.**SUDevice**

Bases: object

Encapsulates a Seneye USB Device and it's capabilities.

action (*cmd*, *timeout=10000*)

Perform action on device.

The available actions are specified by the Action Enum. These actions can include a single write to the device and potentially multiple reads.

Raises

- **usb.core.USBError** – If having issues connecting to the USB
- **TimeoutError** – If read operation times out

Parameters

- **cmd** ([Action](#)) – Action to action
- **timeout** (*int*) – timeout in milliseconds

close ()

Close connection to USB device and clean up instance.

class pyseneye.sud.**SensorReadingResponse**(*raw_data*, *read_def*)

Bases: [pyseneye.sud.BaseResponse](#)

Response which contains all sensor data.

flags

Raw flags information. Not usable yet.

Returns the raw flags bytes

Return type array('B', [2])

is_kelvin

Is light reading on kelvin line: <https://tinyurl.com/yy2wtaz5>.

Returns True if on kelvin line, False if not

Return type bool

is_light_reading

Is the sensor reading a light reading.

Returns True if a light reading, False if a sensor reading.

Return type bool

kelvin

Kelvin value of the light reading.

Returns the kelvin value

Return type int

kelvin_x

X co-ordinate on the CIE colourspace <https://tinyurl.com/yy2wtaz5>.

Limited to colors that are near the kelvin line. Check with `is_kelvin`.

Returns X co-ordinate

Return type int

kelvin_y

Y co-ordinate on the CIE colourspace <https://tinyurl.com/yy2wtaz5>.

Limited to colors that are near the kelvin line. Check with `is_kelvin`.

Returns Y co-ordinate

Return type int

lux

LUX value for light reading.

Returns LUX value

Return type int

nh3

NH3 reading from the device.

Returns the NH3 value

Return type float

par

PAR value for light reading.

Returns PAR value

Return type int

ph

PH reading from the device.

Returns the PH value

Return type float

pur

PUR value for light reading.

Returns PUR value

Return type int

temperature

Temperature reading from the device.

Returns the temperature

Return type float

timestamp

Time the reading was taken at.

(only available for sensor readings)

Returns Unix epoch time

Return type float

1.3 Module contents

pyseneye is to integrate with the Seneye range sensors.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pyseneye`, 5

`pyseneye.sud`, 1

A

ack (*pyseneye.sud.Response* attribute), 3
Action (*class in pyseneye.sud*), 1
action() (*pyseneye.sud.SUDevice* method), 3
ActionDefinition (*class in pyseneye.sud*), 1

B

BaseResponse (*class in pyseneye.sud*), 1

C

close() (*pyseneye.sud.SUDevice* method), 3
cmd_str (*pyseneye.sud.ActionDefinition* attribute), 1

D

device_type (*pyseneye.sud.EnterInteractiveResponse* attribute), 2
DeviceType (*class in pyseneye.sud*), 2

E

ENTER_INTERACTIVE_MODE (*pyseneye.sud.Action* attribute), 1
EnterInteractiveResponse (*class in pyseneye.sud*), 2

F

flags (*pyseneye.sud.SensorReadingResponse* attribute), 3

H

HOME (*pyseneye.sud.DeviceType* attribute), 2

I

is_kelvin (*pyseneye.sud.SensorReadingResponse* attribute), 3
is_light_reading (*pyseneye.sud.SensorReadingResponse* attribute), 3

K

kelvin (*pyseneye.sud.SensorReadingResponse* attribute), 4
kelvin_x (*pyseneye.sud.SensorReadingResponse* attribute), 4
kelvin_y (*pyseneye.sud.SensorReadingResponse* attribute), 4

L

LEAVE_INTERACTIVE_MODE (*pyseneye.sud.Action* attribute), 1
LIGHT_READING (*pyseneye.sud.Action* attribute), 1
lux (*pyseneye.sud.SensorReadingResponse* attribute), 4

N

nh3 (*pyseneye.sud.SensorReadingResponse* attribute), 4

P

par (*pyseneye.sud.SensorReadingResponse* attribute), 4
parse_str (*pyseneye.sud.ReadDefinition* attribute), 2
ph (*pyseneye.sud.SensorReadingResponse* attribute), 4
POND (*pyseneye.sud.DeviceType* attribute), 2
pur (*pyseneye.sud.SensorReadingResponse* attribute), 4
pyseneye (*module*), 5
pyseneye.sud (*module*), 1

R

read_definitions (*pyseneye.sud.ActionDefinition* attribute), 1
ReadDefinition (*class in pyseneye.sud*), 2
REEF (*pyseneye.sud.DeviceType* attribute), 2
Response (*class in pyseneye.sud*), 3
return_type (*pyseneye.sud.ReadDefinition* attribute), 2
return_values (*pyseneye.sud.ReadDefinition* attribute), 2

S

SENSOR_READING (*pyseneye.sud.Action* attribute), 1

SensorReadingResponse (*class in pyseneye.sud*), 3
SUDevice (*class in pyseneye.sud*), 3

T

temperature (*pyseneye.sud.SensorReadingResponse*
attribute), 4
timestamp (*pyseneye.sud.SensorReadingResponse* *at-*
tribute), 4

V

validation_bytes (*pyseneye.sud.BaseResponse* *at-*
tribute), 2
validator (*pyseneye.sud.ReadDefinition* *attribute*), 2
version (*pyseneye.sud.EnterInteractiveResponse* *at-*
tribute), 2