python-sakuraio Documentation

Release 0.1

SAKURA Internet Inc.

Jul 01, 2022

for Hardware

Python-sakuraio is a library for IoT PaaS of SAKURA Internet Inc. It allows users to connect to the Sakura Communication Modules, and APIs of the platform.

CHAPTER 1

Installation

Python-sakuraio can be installed from Github.com with tools like `pip`:

```
# From PyPi
$ pip install sakuraio
# From Github.com
$ pip install -e git+https://github.com/sakuraio/python-sakuraio.git#egg=sakuraio
```

1.1 Requirements

Python-sakuraio is tested on Python >= 3.4,

1.1.1 Raspberry Pi

Python-sakuraio is tested against all supported versions of Raspberry Pi and Raspbian with Raspberry Pi.

- **Raspberry Pi**: 3, Zero
- **Raspbian**: Raspbian Jessie 2017-03-02

```
$ sudo apt-get install python3 python3-pip python3-smbus
$ pip3 install -e git+https://github.com/sakuraio/python-sakuraio.git#egg=sakuraio
```

CHAPTER 2

Getting Started

Python-sakuraio provides a functions to execute command on Sakura Communication Modules.

2.1 Opening a interface

Create instance for treat a Sakura Communication Module:

```
from sakuraio.hardware.rpi import SakuraIOSMBus

sakuraio = SakuraIOSMBus()
```

2.2 Examples

Get the unique id of a Sakura Communication Module:

```
>>> from sakuraio.hardware.rpi import SakuraIOSMBus
>>> sakuraio = SakuraIOSMBus()
>>> sakuraio.get_unique_id()
"16X0000001"
```

CHAPTER 3

Hardware API

3.1 Common

CommonMixins.**get_connection_status**()
Get connection status

Returns Status. Please see the datasheet.

Return type int

CommonMixins.**get_is_online**()
Get online

Returns Weather or not the module is online.

Return type bool

CommonMixins.**get_connection_error**()
Get connection error

Returns Status. Possible values: `CONNECTION_ERROR_NONE`, `CONNECTION_ERROR_OUT_OF_SERVICE`, `CONNECTION_ERROR_CONNECTION`, `CONNECTION_ERROR_DISCONNECTED`

Return type int

CommonMixins.**get_signal_quality**()
Get signal quality

Returns Signal quality. `0`: out of service. `5`: most strong.

Return type int

CommonMixins.**get_datetime**()
Get current datetime

Returns Current datetime.

Return type datetime.datetime

CommonMixins.**echoback**()
Test echoback MCU <-> Communication Module

Parameters **values** (*list*) – List of int values to send.

Returns Values echoed. It must equals `values` param.

Return type list

3.2 Transmit

TransmitMixins.**enqueue_tx_raw**(*channel*, *type*, *data*, *offset=0*)
Enqueue channel data by raw values.

Parameters

- **channel** (*int*) – Channel number of data. Must be 0 to 127.
- **type** (*string*) – Type of data. Possible values `"i"`, `"I"`, `"l"`, `"L"`, `"f"`, `"d"` or `"b"`.
- **values** (*list*) – List of int values to enqueue.

Params int offset Time offset in ms. Default `0`. It must be less than or equal `0`.

TransmitMixins.**enqueue_tx**(*channel*, *value*, *offset=0*)
Enqueue channel data by value.

Parameters

- **channel** (*int*) – Channel number of data. Must be 0 to 127.
- **value** (*integer, float, str or bytes*) – value to enqueue.

Params int offset Time offset in ms. Default `0`. It must be less than or equal `0`.

TransmitMixins.**send_immediate_raw**(*channel*, *type*, *data*)
Send channel data immediately by raw values.

Parameters

- **channel** (*int*) – Channel number of data. Must be 0 to 127.
- **type** (*string*) – Type of data. Possible values `"i"`, `"I"`, `"l"`, `"L"`, `"f"`, `"d"` or `"b"`.
- **values** (*list*) – List of int values to send.

TransmitMixins.**send_immediate**(*channel*, *value*)
Send channel data immediately by value.

Parameters

- **channel** (*int*) – Channel number of data. Must be 0 to 127.
- **value** (*integer, float, str or bytes*) – value to enqueue.

TransmitMixins.**get_tx_queue_length**()
Get available and queued length of tramsmit queue.

Returns Size of available and queued data.

Return type dict

TransmitMixins.**clear_tx**()
Clear transmit queue.

TransmitMixins.**send**()
Send data in transmit queue.

TransmitMixins.**get_tx_status**()
Get status of send

Returns Status of send.

Return type dict

3.3 Receive

ReceiveMixins.**dequeue_rx_raw**()
Dequeue received data

Returns Dict of received data.

Return type dict

ReceiveMixins.**peek_rx_raw**()
Peek received data

Returns Dict of received data.

Return type dict

ReceiveMixins.**get_rx_queue_length**()
Get available and queued length of receive queue.

Returns Size of available and queued data.

Return type dict

ReceiveMixins.**clear_rx**()
Clear receive queue.

3.4 Operation

OperationMixins.**get_product_id**()
Get product id

Returns Product ID. Possible values: `PRODUCT_ID_SCM_LTE_BETA`, `PRODUCT_ID_SCM_LTE_01`

Return type int

OperationMixins.**get_product_name**()
Get product name

Returns Product name. Possible values: `"SCM-LTE-BETA"`, `"SCM-LTE-01"`.

Return type str

OperationMixins.**get_unique_id**()
Get unique id

Returns Unique ID. For example *"16X0000001"*.

Return type str

OperationMixins.**get_firmware_version**()
Get firmware version

Returns Firmware version. For example *"v1.1.2-170223-7e6ce64"*.

Return type str

OperationMixins.**unlock**()
Unlock critical command

OperationMixins.**update_firmware**()
Request to update firmware

OperationMixins.**get_firmware_update_status**()
Get firmware update status

Returns Status.

Return type dict

OperationMixins.**reset**()
Request software reset

3.5 File

FileMixins.**start_file_download**(*fileid*)
Start file download

Parameters **fileid** (*integer*) – FileID of start to download, must be 1 to 5.

FileMixins.**get_file_metadata**()
Get file metadata

Returns Dict of file metadata (status, filesize, timestamp, checksum).

Return type dict

FileMixins.**get_file_download_status**()
Get file download status

Returns Dict of download status and received datasize.

Return type dict

FileMixins.**cancel_file_download**()
Cancel file download

FileMixins.**get_file_data**(*rsize*)
Get file data

Parameters **rsize** (*integer*) – Max receive size, must be 1 to 255.

Returns Part of data

Return type list

Index

R

S

U