Python

Release 1.0.5

Apr 22, 2017

Contents

1 API Documentation **3**
1.1 modbusreader 3
1.2 modbusreader.structutils 5

2 Indices and tables **7**

Python Module Index **9**

Contents:

CHAPTER 1

API Documentation

modbusreader

class `modbusreader.`**`ModbusReader`**(*host*, *port*, *unit*, *modbus_device_definition*, *float_low_byte_first=False*)

Bases: `object`

ModbusReader is an automated modbus client which reads all discretes and registers of a modbus server over TCP

`__init__`(*host*, *port*, *unit*, *modbus_device_definition*, *float_low_byte_first=False*)

Initializes a new instance

Parameters

- **`host`** (*`str`*) – host of modbus server
- **`port`** (*`int`*) – port of modbus server
- **`unit`** (*`int`*) – unit id
- **`modbus_device_definition`** (*`dict or str`*) – modbus device definition python dictionary or file name based on config *https://github.com/smueller18/modbus-readermodbusreader/modbus_definition.config.json*
- **`float_low_byte_first`** (*`bool`*) – Because modbus float datatype consists of two integer bytes, there are 2 possibilities for the determination of the float value. Set to True if float interpretation order is Low Byte and then High Byte. Otherwise interpretation order is High Byte and then Low Byte.

Raises

- **`ValidationError`** – if the modbus device definition dictionary or file is invalid
- **`SchemaError`** – if the modbus device definition config itself is invalid

`__weakref__`

list of weak references to the object (if defined)

static **group_modbus_device_definition**(*modbus_device_definition*)
Groups modbus addresses. This method is needed, if there are gaps of non existent modbus addresses.

Parameters **modbus_device_definition** (*dict*) – modbus device definition dictionary

Returns grouped modbus device definition dictionary

Type dict

read_all_values()
read discretes and registers

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises **IOError** – is raised if reading discretes or registers over TCP connection fails

read_discrete_inputs()
read discrete inputs

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises **IOError** – is raised if reading discretes over TCP connection fails

read_discrete_outputs()
read discrete outputs

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises **IOError** – is raised if reading discretes over TCP connection fails

read_discretes(*discrete_type*)
read either discrete inputs or outputs

Parameters **discrete_type** (*str*) – type of discrete. either 'input' or 'output'

Returns discrete values: { sensor_id: sensor_value, ... }

Type dict

Raises

- **AttributeError** – is raised if discrete_type doesn't match required types
- **IOError** – is raised if reading discretes over TCP connection fails

read_input_registers()
read input registers

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises **IOError** – is raised if reading registers over TCP connection fails

read_output_registers()
read output registers

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises **IOError** – is raised if reading registers over TCP connection fails

read_registers (*register_type*)
read either input or output registers

Parameters **register_type** (*str*) – type of register. either 'input' or 'output'

Returns discrete output values as follows: { sensor_id: sensor_value, ... }

Type dict

Raises

- **AttributeError** – If register_type doesn't match required types
- **IOError** – If reading registers over TCP connection fails

modbusreader.structutils

structutils.py: extends the function of the struct package

modbusreader.structutils.**bytes_to_datatype** (*byte_list*, *data_type*)
Unpacks a bytes object to the given data type

Parameters

- **byte_list** (*bytes*) – bytes object
- **data_type** (*str*) – human readable data type. One of: int16, int32, uint32, float, byte, boolean

Returns unpacked value

Type int, float, byte, boolean

Raises **ValueError** – If size of data type is not equal to the size of the bytes object.

modbusreader.structutils.**calcsize** (*data_type*)
Return size in bytes of the struct described by the given data type

Parameters **data_type** (*str*) – human readable data type. One of: int16, int32, uint32, float, byte, boolean

Returns size in bytes of the struct described by the given data type

Type int

modbusreader.structutils.**get_format** (*data_type*)
Get struct format type from human readable data type

Parameters **data_type** (*str*) – human readable data type. One of: int16, int32, uint32, float, byte, boolean

Returns struct format type

Type str

modbusreader.structutils.**int16list_to_bytes** (*int16_list*)
Packs all given integer values into bytes object

Parameters **int16_list** (*list of int*) – list containing unsigned 16 bit integers

Returns packed integer values

Type bytes

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

m

modbusreader, 3
modbusreader.structutils, 5

Index

Symbols

__init__() (modbusreader.ModbusReader method), 3
__weakref__ (modbusreader.ModbusReader attribute), 3

B

bytes_to_datatype() (in module modbusreader.structutils), 5

C

calcsize() (in module modbusreader.structutils), 5

G

get_format() (in module modbusreader.structutils), 5
group_modbus_device_definition() (modbusreader.ModbusReader static method), 3

I

int16list_to_bytes() (in module modbusreader.structutils), 5

M

ModbusReader (class in modbusreader), 3
modbusreader (module), 3
modbusreader.structutils (module), 5

R

read_all_values() (modbusreader.ModbusReader method), 4
read_discrete_inputs() (modbusreader.ModbusReader method), 4
read_discrete_outputs() (modbusreader.ModbusReader method), 4
read_discretes() (modbusreader.ModbusReader method), 4
read_input_registers() (modbusreader.ModbusReader method), 4
read_output_registers() (modbusreader.ModbusReader method), 4
read_registers() (modbusreader.ModbusReader method), 4