

---

# Python

*Release 1.0.3*

Apr 06, 2017



---

## Contents

---

<b>1</b>	<b>API Documentation</b>	<b>3</b>
1.1	modbusreader . . . . .	3
1.2	modbusreader.structutils . . . . .	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



Contents:



## 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](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`



### m

`modbusreader`, 3

`modbusreader.structutils`, 5



## 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