iolink

Release 0.0.5

Maxim-Trinamic Software Team

Apr 01, 2021

CONTENTS:

IO-Link is a standardized point-to-point connection down at the edge layer of the factory automation pyramid.

The **iolink** Python package allows access to IO-Link devices from within Python, by providing a common abstraction layer to different IO-Link adapters.

Note that for now, only the iqLink® device is supported and only under Windows.

The following example prints the product name of a connected device, by reading out the standard ISDU parameter 0x12.

```
import iolink

# create a port instance
with iolink.get_port(interface='iqLink') as port:
    # change state of the connected device to "Operate"
    port.change_device_state_to('Operate')
    # read standard ISDU
    isdu_0x12_data = port.read_device_isdu(0x12, 0)
    # convert the received bytes object that's supposed
    # to be an ASCII string to a standard Python 3 string
    # and print the result
    print(f'Product Name: {isdu_0x12_data.decode("utf8")}')
```

CHAPTER

ONE

INSTALLATION

1.1 How to install iolink itself

Install **iolink** from PyPI using `pip`:

```
$ python -m pip install iolink
```

This makes sure that the `pip` you are using belongs to your Python distribution. But you may also do it with:

```
$ pip install iolink
```

1.2 Dependencies that are required

1.2.1 When using an iqLink®

- Download the iqDLL (iqcomm.dll) from the IQ2 website.
- Make the iqDLL available to `iolink` by copying the iqcomm.dll file:
  - to the same directory where your main Python file resides, or by
  - copying the file to a known location in your system and adding this directory to the PATH environment variable.

CHAPTER

TWO

API DOCUMENTATION

iolink.**get_port**(*interface*)

Factory of specific instances of the abstract Port class.

Parameters **interface** (*str*) – ID of your IO-Link master device - currently only *iqLink* is supported.

class iolink.port.**PortABC**

Abstract base class that represents one Masters IO-Link port.

abstract change_device_state_to(*target_state: str*)

Sends a request to the device to change the state.

Parameters **target_state** (*str*) – allowed strings are 'Inactive', 'PreOperate' or 'Operate'.

abstract get_device_pd_input_and_status() → Tuple[bytes, int]

Gets the input process data from a device and the state information.

abstract power_off()

Switches off the IO-Link power line of the port.

abstract power_on()

Switches on the IO-Link power line of the port.

abstract read_device_isdu(*index: int*, *subindex: int*)

Reads content of a parameter from the device.

abstract set_device_pd_output(*data: bytes*)

Sets the output process data for a device.

abstract write_device_isdu(*index: int*, *subindex: int*, *data*)

Writes content of a parameter from the device.

Make sure the size of the data matches the size of the devices parameter.

CHAPTER

THREE

TRADEMARKS

IO-Link is a registered trademark of Profibus User Organization (PNO)

INDEX