trio-serial

Release 0.1.0

Jörn Heissler

May 15, 2021

CONTENTS:

This project is an adaption of the pyserial project for trio.

CHAPTER

ONE

INSTALLATION

```
$ pip install trio-serial
```

CHAPTER

TWO

EXAMPLE

```
from trio import run
from trio_serial import SerialStream

async def main():
    async with SerialStream("/dev/ttyUSB0") as ser:
        for i in range(10):
            buf = await ser.receive_some()
            await ser.send_all(buf)
            await ser.send_all(buf)

run(main)
```

CHAPTER

THREE

API REFERENCE

class trio_serial.abstract.**AbstractSerialStream**(*port*, ***, *exclusive=False*, *baudrate=115200*, *bytesize=8*, *parity=<Parity.NONE: 1>*, *stopbits=<StopBits.ONE: 1>*, *xonxoff=False*, *rtscts=False*)

Operating system independant public interface of `SerialStream`.

Create new SerialStream object.

Parameters

- **port** (*str*) – Name of port. Format depends on implementation. This could be "/dev/ttyUSB0" on Linux or "COM7" on Windows.
- **exclusive** (*bool*) – Lock port for exclusive use
- **baudrate** (*int*) – Initial Port speed
- **bytesize** (*int*) – Number of bits per byte
- **parity** (`Parity`) – Parity
- **stopbits** (`StopBits`) – Number of stop bits
- **xonxoff** (*bool*) – Software Flow Control with XON/XOFF
- **rtscts** (*bool*) – Hardware Flow Control with RTS/CTS

Return type None

abstractmethod await aclose()

Cleanly close the port.

Do nothing if already closed.

Return type None

abstractmethod await aopen()

Open the port and configure it with the initial state from `__init__()`.

Return type None

abstractmethod await discard_input()

Discard any unread input.

Return type None

abstractmethod await discard_output()

Discard any unwritten output.

Return type None

abstractmethod await get_cts()
Retrieve current *Clear To Send* state.

Returns Current CTS state

Return type bool

abstractmethod await get_hangup()
Retrieve current *Hangup on Close* state.

Returns Current *Hangup on Close* state

Return type bool

abstractmethod await get_rts()
Retrieve current *Ready To Send* state.

Returns Current RTS state

Return type bool

property port: str
Get the port name.

Returns port name or device

await receive_some(*max_bytes=None*)
Receive between 1 and `max_bytes` bytes from the serial port.

Parameters **max_bytes** (*Optional[int]*) – Maximum number of bytes to receive.

Return type bytes

await send_all(*data*)
Send `data` to the serial port. :param data: Data to send

Parameters **data** (*ByteString*) –

Return type None

abstractmethod await send_break(*duration=0.25*)
Transmit a continuous stream of zero-valued bits for a specific duration.

Params: duration: Number of seconds

Parameters **duration** (*float*) –

Return type None

abstractmethod await set_hangup(*value*)
Set *Hangup on Close* state.

Parameters **value** (*bool*) – New *Hangup on Close* state

Return type None

abstractmethod await set_rts(*value*)
Set *Ready To Send* state.

Parameters **value** (*bool*) – New *Ready To Send* state

Return type None

await wait_send_all_might_not_block()
Wait until sending might not block (it still might block).

Return type None

class trio_serial.abstract.**Parity**(*value*)

Enumeration of parity types.

EVEN = 2

Even parity

MARK = 4

Parity bit always 1

NONE = 1

No parity

ODD = 3

Odd parity

SPACE = 5

Parity bit always 0

class trio_serial.abstract.**StopBits**(*value*)

Enumeration of stop bit lengths.

ONE = 1

One bit

ONE_POINT_FIVE = 2

One and a half bits

TWO = 3

Two bits

CHAPTER

FOUR

CONTRIBUTING AND SUPPORT

Please create a new Issue or Pull request on GitHub to contribute changes or ask questions.

CHAPTER

FIVE

DEVELOPMENT AND TESTING

To test the package locally, run: .. code-block:: console

    $ /path/to/pip install -U .

where pip could be in a virtual environment.

CHAPTER

SIX

CHANGELOG

6.1 0.3.0 - 2021-05-15

- Fix cleanup issue when running in sync context. Remove `close` method. (#4)

6.2 0.2.1 - 2021-04-09

- Do not handle modem bits (e.g. rts) in constructor / aopen. Use *set_rts()*/*get_rts()* instead. (#2)
- Add new methods to control "hangup on close":
  - *get_hangup()*
  - *set_hangup()*

6.3 0.1.2 - 2021-02-07

- More relaxed dependencies

6.4 0.1.1 - 2021-01-31

- Add new methods:
  - *port()*
  - *discard_input()*
  - *discard_output()*
  - *send_break()*

6.5 0.1.0 - 2020-12-21

- Initial release.

CHAPTER

SEVEN

LICENSE

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search

INDEX