
PyAbacus Documentation

Release 1.0.5

Tausand Electronica

Jul 29, 2018

CONTENTS

| | | |
|----------|-------------------------------|-----------|
| 1 | Contents | 3 |
| 1.1 | pyAbacus.core | 3 |
| 1.2 | pyAbacus.exceptions | 4 |
| 1.3 | pyAbacus.constants | 4 |
| 2 | Indices and tables | 7 |
| | Python Module Index | 9 |
| | Index | 11 |



pyAbacus was build to simplify the usage of [Tausands](#) tools.

CONTENTS

1.1 pyAbacus.core

class pyAbacus.core.**AbacusSerial** (*port, bounce_timeout=20*)
Builds a serial port from pyserial.

flush ()

getIdn ()

readSerial ()

writeSerial (*command, address, data_u16*)

class pyAbacus.core.**CountersValues** (*n_channels*)

getCountersID ()

getTimeLeft ()

getValue (*channel*)

setValueFromArray (*address, value*)

class pyAbacus.core.**Settings2Ch**

getAddressAndValue (*timer*)

getSetting (*timer*)

setSetting (*setting, value*)

setValueFromArray (*address, value*)

pyAbacus.core.**close** (*abacus_port*)

pyAbacus.core.**dataArraysToCounters** (*addresses, data*)

pyAbacus.core.**dataArraysToSettings** (*addresses, data*)

pyAbacus.core.**dataStreamToDataArrays** (*input_string*)

pyAbacus.core.**findDevices** ()

pyAbacus.core.**getAllCounters** (*abacus_port*)

pyAbacus.core.**getAllSettings** (*abacus_port*)

pyAbacus.core.**getCountersID** (*abacus_port*)

pyAbacus.core.**getIdn** (*abacus_port*)

```
pyAbacus.core.getSetting(abacus_port, setting)
pyAbacus.core.getTimeLeft(abacus_port)
pyAbacus.core.open(abacus_port)
pyAbacus.core.readSerial(abacus_port)
pyAbacus.core.setAllSettings(abacus_port, new_settings)
pyAbacus.core.setSetting(abacus_port, setting, value)
pyAbacus.core.writeSerial(abacus_port, command, address, data_u16)
```

1.2 pyAbacus.exceptions

exception pyAbacus.exceptions.AbacusError(message="")
An unexpected error occurred.

exception pyAbacus.exceptions.BaseError(message)

exception pyAbacus.exceptions.CheckSumError
An error occurred while doing check sum.

exception pyAbacus.exceptions.TimeoutError(message="")
A time out error occurred

1.3 pyAbacus.constants

pyAbacus.constants.ADDRESS_DIRECTORY_2CH = {'coincidence_window_ms': 22, 'coincidence_windo
Memory addresses

pyAbacus.constants.BAUDRATE = 115200
Default baudrate for the serial port communication

pyAbacus.constants.BOUNCE_TIMEOUT = 20
Number of times a specific transmittion is tried

pyAbacus.constants.COINCIDENCE_WINDOW_DEFAULT_VALUE = 5
Default coincidence window time value (ns).

pyAbacus.constants.COINCIDENCE_WINDOW_MAXIMUM_VALUE = 50000
Maximum coincidence window time value (ns).

pyAbacus.constants.COINCIDENCE_WINDOW_MINIMUM_VALUE = 5
Minimum coincidence window time value (ns).

pyAbacus.constants.COINCIDENCE_WINDOW_STEP_VALUE = 5
Increase ratio on the coincidence window time value (ns).

pyAbacus.constants.COUNTERS_VALUES = None
Global counters values variable

pyAbacus.constants.CURRENT_OS = 'linux'
Current operative system

pyAbacus.constants.DELAY_DEFAULT_VALUE = 100
Default delay time value (ns).

`pyAbacus.constants.DELAY_MAXIMUM_VALUE = 100`
Maximum delay time value (ns).

`pyAbacus.constants.DELAY_MINIMUM_VALUE = 0`
Minimum delay time value (ns).

`pyAbacus.constants.DELAY_STEP_VALUE = 5`
Increase ratio on the delay time value (ns).

`pyAbacus.constants.END_COMMUNICATION = 4`
End of message

`pyAbacus.constants.MAXIMUM_WRITING_TRIES = 20`
Number of tries done to write a value

`pyAbacus.constants.READ_VALUE = 14`
Reading operation signal

`pyAbacus.constants.SAMPLING_DEFAULT_VALUE = 100`
Default sampling time value (ms)

`pyAbacus.constants.SAMPLING_VALUES = [1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000]`
From (1, 2, 5) ms to 1000 s

`pyAbacus.constants.SETTINGS = None`
Global settings variable

`pyAbacus.constants.SLEEP_DEFAULT_VALUE = 25`
Default sleep time value (ns).

`pyAbacus.constants.SLEEP_MAXIMUM_VALUE = 100`
Maximum sleep time value (ns).

`pyAbacus.constants.SLEEP_MINIMUM_VALUE = 0`
Minimum sleep time value (ns).

`pyAbacus.constants.SLEEP_STEP_VALUE = 5`
Increase ratio on the sleep time value (ns).

`pyAbacus.constants.START_COMMUNICATION = 2`
Begin message signal

`pyAbacus.constants.TIMEOUT = 0.04`
Maximum time without answer from the serial port

`pyAbacus.constants.WRITE_VALUE = 15`
Writing operation signal

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

p

`pyAbacus.constants`, 4
`pyAbacus.core`, 3
`pyAbacus.exceptions`, 4

A

AbacusError, 4
 AbacusSerial (class in pyAbacus.core), 3
 ADDRESS_DIRECTORY_2CH (in module pyAbacus.constants), 4

B

BaseError, 4
 BAUDRATE (in module pyAbacus.constants), 4
 BOUNCE_TIMEOUT (in module pyAbacus.constants), 4

C

ChecksumError, 4
 close() (in module pyAbacus.core), 3
 COINCIDENCE_WINDOW_DEFAULT_VALUE (in module pyAbacus.constants), 4
 COINCIDENCE_WINDOW_MAXIMUM_VALUE (in module pyAbacus.constants), 4
 COINCIDENCE_WINDOW_MINIMUM_VALUE (in module pyAbacus.constants), 4
 COINCIDENCE_WINDOW_STEP_VALUE (in module pyAbacus.constants), 4
 COUNTERS_VALUES (in module pyAbacus.constants), 4
 CountersValues (class in pyAbacus.core), 3
 CURRENT_OS (in module pyAbacus.constants), 4

D

dataArraysToCounters() (in module pyAbacus.core), 3
 dataArraysToSettings() (in module pyAbacus.core), 3
 dataStreamToDataArrays() (in module pyAbacus.core), 3
 DELAY_DEFAULT_VALUE (in module pyAbacus.constants), 4
 DELAY_MAXIMUM_VALUE (in module pyAbacus.constants), 4
 DELAY_MINIMUM_VALUE (in module pyAbacus.constants), 5
 DELAY_STEP_VALUE (in module pyAbacus.constants), 5

E

END_COMMUNICATION (in module pyAbacus.constants), 5

F

findDevices() (in module pyAbacus.core), 3
 flush() (pyAbacus.core.AbacusSerial method), 3

G

getAddressAndValue() (pyAbacus.core.Settings2Ch method), 3
 getAllCounters() (in module pyAbacus.core), 3
 getAllSettings() (in module pyAbacus.core), 3
 getCountersID() (in module pyAbacus.core), 3
 getCountersID() (pyAbacus.core.CountersValues method), 3
 getIdn() (in module pyAbacus.core), 3
 getIdn() (pyAbacus.core.AbacusSerial method), 3
 getSetting() (in module pyAbacus.core), 4
 getSetting() (pyAbacus.core.Settings2Ch method), 3
 getTimeLeft() (in module pyAbacus.core), 4
 getTimeLeft() (pyAbacus.core.CountersValues method), 3
 getValue() (pyAbacus.core.CountersValues method), 3

M

MAXIMUM_WRITING_TRIES (in module pyAbacus.constants), 5

O

open() (in module pyAbacus.core), 4

P

pyAbacus.constants (module), 4
 pyAbacus.core (module), 3
 pyAbacus.exceptions (module), 4

R

READ_VALUE (in module pyAbacus.constants), 5
 readSerial() (in module pyAbacus.core), 4
 readSerial() (pyAbacus.core.AbacusSerial method), 3

S

SAMPLING_DEFAULT_VALUE (in module pyAbacus.constants), [5](#)
SAMPLING_VALUES (in module pyAbacus.constants), [5](#)
setAllSettings() (in module pyAbacus.core), [4](#)
setSetting() (in module pyAbacus.core), [4](#)
setSetting() (pyAbacus.core.Settings2Ch method), [3](#)
SETTINGS (in module pyAbacus.constants), [5](#)
Settings2Ch (class in pyAbacus.core), [3](#)
setValueFromArray() (pyAbacus.core.CountersValues method), [3](#)
setValueFromArray() (pyAbacus.core.Settings2Ch method), [3](#)
SLEEP_DEFAULT_VALUE (in module pyAbacus.constants), [5](#)
SLEEP_MAXIMUM_VALUE (in module pyAbacus.constants), [5](#)
SLEEP_MINIMUM_VALUE (in module pyAbacus.constants), [5](#)
SLEEP_STEP_VALUE (in module pyAbacus.constants), [5](#)
START_COMMUNICATION (in module pyAbacus.constants), [5](#)

T

TIMEOUT (in module pyAbacus.constants), [5](#)
TimeOutError, [4](#)

W

WRITE_VALUE (in module pyAbacus.constants), [5](#)
writeSerial() (in module pyAbacus.core), [4](#)
writeSerial() (pyAbacus.core.AbacusSerial method), [3](#)