pypdm Documentation

Release 1.0

Olivier Hériveaux

Oct 08, 2019

Contents:

1 Getting started! 1
- 1.1 Connecting to PDM devices . . . 1
- 1.2 Laser in continuous operation . . . 2
- 1.3 Configuring the laser source . . . 2
- 1.4 Laser in pulsed operation . . . 3
- 1.5 Safety . . . 3

2 Python reference API 5

3 License 9

4 Indices and tables 11

Python Module Index 13

Index 15

CHAPTER 1

Getting started!

PyPDM is a small and simple Python3 library for controlling Alphanov's PDM laser sources. Currently supported PDM protocol version is 3.4. Daisy-chain configuration for multiple devices is supported, so it is possible to use many PDMs with only one serial link.

1.1 Connecting to PDM devices

The following example shows how to connect to a PDM device using the *pypdm.PDM* class. The first argument is the PDM device address, which may be different depending on your PDM configuration. The second argument is the serial device path (may be '/dev/ttyUSB0' on linux).

When connecting, the library will query the device version and raise a *pypdm.ProtocolVersionNotSupported* exception if not correct.

```
import pypdm

pdm = pypdm.PDM(1, 'COM0')
```

If you use multiple laser sources in daisy-chain configuration, you can instantiate one *pypdm.PDM* object for each device like in the following example:

```
import pypdm

pdm1 = pypdm.PDM(1, 'COM0')
pdm2 = pypdm.PDM(2, pdm1) # Use same serial as pdm1
pdm3 = pypdm.PDM(3, pdm1) # Use same serial as pdm1
```

Daisy-chain configuration must be used for PDM2+ or PDM4+ devices.

Alternatively, you can use the following equivalent code for daisy-chain configuration:

```
import pypdm
```

(continues on next page)

(continued from previous page)

```
link = pypdm.Link('COM0')
pdm1 = pypdm.PDM(1, link)
pdm2 = pypdm.PDM(2, link)
pdm3 = pypdm.PDM(3, link)
```

1.2 Laser in continuous operation

The following example turns on a laser source in continuous mode. Current is set to 30 mA.

```
import pypdm

pdm = pypdm

pdm = pypdm.PDM(1, 'COM0')
pdm.offset_current = 30
pdm.activation = True
# Apply new settings to the device
pdm.apply()
```

1.3 Configuring the laser source

All configuration parameters of PDM devices can be modified using *pypdm.PDM* member properties. When a setting is changed, call *pypdm.PDM.apply()* to make it effective. Here is the list of properties which can be read or changed (see reference API for more details):

- *pypdm.PDM.sync_source*
- *pypdm.PDM.delay_line_type*
- *pypdm.PDM.frequency*
- *pypdm.PDM.pulse_width*
- *pypdm.PDM.delay*
- *pypdm.PDM.offset_current*
- *pypdm.PDM.current_percentage*
- *pypdm.PDM.current*
- *pypdm.PDM.temperature*
- *pypdm.PDM.maximum_current*
- *pypdm.PDM.current_source*
- *pypdm.PDM.interlock_status*
- *pypdm.PDM.activation*
- *pypdm.PDM.mode*

1.4 Laser in pulsed operation

The following example turns on a laser source for pulsed operation. Pulse power can be specified using the current_percentage or current properties.

```
import pypdm

pdm = pypdm

pdm = pypdm.PDM(1, 'COM0')
pdm.offset_current = 0
pdm.current_source = pypdm.CurrentSource.NUMERIC
pdm.current_percentage = 20
pdm.activation = True
# Apply new settings to the device
pdm.apply()
```

1.5 Safety

When a PDM object is deleted, the library may try to switch off the laser source for safety. However, you shall not rely on this behavior and always beware of dangers when using laser equipments! Please always wear laser safety glasses or use any appropriate safety equipment to prevent any harmful accident.

CHAPTER 2

Python reference API

class pypdm.**PDM**(*address*, *link*)

Class to command one Alphanov's PDM laser sources.

MAX_DELAY = 15000

MAX_PULSE_WIDTH = 1275000

__del__()

For safety, disable laser when the object is deleted.

__init__(*address*, *link*)

Parameters

- **address** – PDM device address.
- **link** – Specify a string for the serial to be used ('/dev/ttyUSBx' or 'COMx'), a *Link* or *PDM* instance for daisy-chained configurations.

activation

True when laser is enabled, False when laser is off. Call *apply()* to make any change effective.

apply()

Apply all the instructions which are in volatile memory. This makes all settings changes effectives.

current

Current, in mA. Please note this property is in milli-amperes and is not a percentage of the maximum current, as the official PDM documentation may state. For the percentage, see current_percentage property. Call *apply()* to make any change effective.

Getter Return diode current configuration.

Setter Set the new current. Raise a ValueError if current is out of bounds.

current_percentage

Current, in percentage of maximum. This is an alternative way of changing the *current* property. Call *apply()* to make any change effective.

current_source
Current source. Set to *CurrentSource.NUMERIC* to control the laser diode pulse current from software through the *current* or *current_percentage* attributes.

Type *CurrentSource*

delay
Delay, in ps. int. Maximum value is defined in MAX_DELAY.

delay_line_type
Delay line type, *DelayLineType* instance.

frequency
Frequency, in Hz. int. Read-only.

interlock_status
True if interlock is detected, False otherwise.

maximum_current
Maximum current, in mA. The getter of this property queries the PDM device once then cache the value for next accesses.

mode
PDM mode, *Mode* instance.

offset_current
Offset current, in mA. float.

pulse_width
Pulse width, in ps. int. Maximum value is defined in MAX_PULSE_WIDTH.

read_address()
Query laser source address.

read_protocol_version()

Returns Protocol version string, for instance '3.4'.

sync_source
Synchronization source, *SyncSource* instance.

temperature
Temperature, in degrees.

class pypdm.**Link**(*dev*)
Base PDM communication implementation. An instance of *Link* uses a serial port and can be shared by multiple *PDM* instances if the devices are daisy-chained.

__init__(*dev*)
Open serial device.

Parameters **dev** – Serial device path. For instance '/dev/ttyUSB0' on linux, 'COM0' on Windows.

class pypdm.**SyncSource**
Possible PDM synchronization source.

EXTERNAL_LVDS = 1

EXTERNAL_TTL_LVTTL = 0

INTERNAL = 2

class pypdm.**DelayLineType**
Possible delay line types.

INTERNAL = 1

NONE = 0

class pypdm.**CurrentSource**
Possible current sources.

ANALOG = 0

NUMERIC = 1

class pypdm.**Mode**
Possible PDM mode.

CONTINUOUS = 1

PULSED = 0

class pypdm.**ChecksumError**
Thrown if a communication checksum error is detected.

class pypdm.**ProtocolError**
Thrown if an unexpected response from the device is received.

class pypdm.**ProtocolVersionNotSupported**(*version*)
Thrown when a PDM protocol version is not (yet) supported by the library.

class pypdm.**ConnectionFailure**

class pypdm.**StatusError**(*status*)
Thrown when a PDM device did not respond with 'OK' status to the last command.

CHAPTER 3

License

PyPDM is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pypdm, 5

Index

Symbols

__del__() (*pypdm.PDM method*), 5
__init__() (*pypdm.Link method*), 6
__init__() (*pypdm.PDM method*), 5

A

activation (*pypdm.PDM attribute*), 5
ANALOG (*pypdm.CurrentSource attribute*), 7
apply() (*pypdm.PDM method*), 5

C

ChecksumError (*class in pypdm*), 7
ConnectionFailure (*class in pypdm*), 7
CONTINUOUS (*pypdm.Mode attribute*), 7
current (*pypdm.PDM attribute*), 5
current_percentage (*pypdm.PDM attribute*), 5
current_source (*pypdm.PDM attribute*), 5
CurrentSource (*class in pypdm*), 7

D

delay (*pypdm.PDM attribute*), 6
delay_line_type (*pypdm.PDM attribute*), 6
DelayLineType (*class in pypdm*), 6

E

EXTERNAL_LVDS (*pypdm.SyncSource attribute*), 6
EXTERNAL_TTL_LVTTL (*pypdm.SyncSource attribute*), 6

F

frequency (*pypdm.PDM attribute*), 6

I

interlock_status (*pypdm.PDM attribute*), 6
INTERNAL (*pypdm.DelayLineType attribute*), 6
INTERNAL (*pypdm.SyncSource attribute*), 6

L

Link (*class in pypdm*), 6

M

MAX_DELAY (*pypdm.PDM attribute*), 5
MAX_PULSE_WIDTH (*pypdm.PDM attribute*), 5
maximum_current (*pypdm.PDM attribute*), 6
Mode (*class in pypdm*), 7
mode (*pypdm.PDM attribute*), 6

N

NONE (*pypdm.DelayLineType attribute*), 7
NUMERIC (*pypdm.CurrentSource attribute*), 7

O

offset_current (*pypdm.PDM attribute*), 6

P

PDM (*class in pypdm*), 5
ProtocolError (*class in pypdm*), 7
ProtocolVersionNotSupported (*class in pypdm*), 7
pulse_width (*pypdm.PDM attribute*), 6
PULSED (*pypdm.Mode attribute*), 7
pypdm (*module*), 5

R

read_address() (*pypdm.PDM method*), 6
read_protocol_version() (*pypdm.PDM method*), 6

S

StatusError (*class in pypdm*), 7
sync_source (*pypdm.PDM attribute*), 6
SyncSource (*class in pypdm*), 6

T

temperature (*pypdm.PDM attribute*), 6