
NSP32 Python API for desktop Documentation

nanoLambda, Inc.

Jan 09, 2019

CONTENTS

1	NanoLambdaNSP32 Module	1
1.1	Classes	1
2	Indices and tables	7
	Python Module Index	9
	Index	11

NANOLAMBDANSP32 MODULE

1.1 Classes

<i>CmdCodeEnum</i>	command code enumeration
<i>NSP32</i> (sendDataDelegate, ...)	NSP32 main class
<i>ReturnPacket</i> (cmdCode, userCode, ...)	return packet
<i>SpectrumInfo</i> (packetBytes)	spectrum info
<i>WavelengthInfo</i> (packetBytes)	wavelength info
<i>XYZInfo</i> (packetBytes)	XYZ info

1.1.1 CmdCodeEnum

class NanoLambdaNSP32.**CmdCodeEnum**

Bases: `enum.IntEnum`

command code enumeration

Attributes Summary

<i>AcqSpectrum</i>	spectrum acquisition
<i>AcqXYZ</i>	XYZ acquisition
<i>GetSensorId</i>	get sensor id
<i>GetSpectrum</i>	get spectrum data
<i>GetWavelength</i>	get wavelength
<i>GetXYZ</i>	get XYZ data
<i>Hello</i>	hello
<i>Prefix0</i>	prefix 0
<i>Prefix1</i>	prefix 1
<i>Standby</i>	standby
<i>Unknown</i>	unknown

Attributes Documentation

AcqSpectrum = 38
spectrum acquisition

AcqXYZ = 42
XYZ acquisition

GetSensorId = 6
get sensor id

GetSpectrum = 40
get spectrum data

GetWavelength = 36
get wavelength

GetXYZ = 44
get XYZ data

Hello = 1
hello

Prefix0 = 3
prefix 0

Prefix1 = 187
prefix 1

Standby = 4
standby

Unknown = 0
unknown

1.1.2 NSP32

class NanoLambdaNSP32.**NSP32** (*sendDataDelegate, returnPacketReceivedDelegate*)

Bases: object

NSP32 main class

Methods Summary

<i>AcqSpectrum</i> (userCode, integrationTime, ...)	start spectrum acquisition
<i>AcqXYZ</i> (userCode, integrationTime, ...)	start XYZ acquisition
<i>GetSensorId</i> (userCode)	get sensor id
<i>GetWavelength</i> (userCode)	get wavelength
<i>Hello</i> (userCode)	say hello to NSP32
<i>OnReturnByteReceived</i> (rcv)	byte received handler (call this function when receiving a single byte from data channel)
<i>OnReturnBytesReceived</i> (data)	bytes received handler (call this function when receiving multiple bytes from data channel)
<i>Standby</i> (userCode)	standby NSP32

Methods Documentation

AcqSpectrum (*userCode, integrationTime, frameAvgNum, enableAE*)

start spectrum acquisition

We will let NSP32 actively send out the “GetSpectrum” return packet once the acquisition is done, so there is no GetSpectrum() function in this API.

Args: userCode(int): user code

integrationTime(int): integration time

frameAvgNum(int): frame average num

enableAE(bool): True to enable AE; False to disable AE

AcqXYZ (*userCode*, *integrationTime*, *frameAvgNum*, *enableAE*)
start XYZ acquisition

We will let NSP32 actively send out the “GetXYZ” return packet once the acquisition is done, so there is no GetXYZ() function in this API.

Args: *userCode*(int): user code
integrationTime(int): integration time
frameAvgNum(int): frame average num
enableAE(bool): True to enable AE; False to disable AE

GetSensorId (*userCode*)
get sensor id

Args: *userCode*(int): user code

GetWavelength (*userCode*)
get wavelength

Args: *userCode*(int): user code

Hello (*userCode*)
say hello to NSP32

Args: *userCode*(int): user code

OnReturnByteReceived (*rcv*)
byte received handler (call this function when receiving a single byte from data channel)

Args: *rcv*(int): single byte received

OnReturnBytesReceived (*data*)
bytes received handler (call this function when receiving multiple bytes from data channel)

Args: *data*(list): bytes received

Standby (*userCode*)
standby NSP32

Args: *userCode*(int): user code

1.1.3 ReturnPacket

class NanoLambdaNSP32.**ReturnPacket** (*cmdCode*, *userCode*, *isPacketValid*, *packetBytes*)

Bases: object

return packet

Attributes Summary

<i>CmdCode</i>	CmdCodeEnum: command function code
<i>IsPacketValid</i>	bool: check if the packet is valid (True for valid; False for invalid)
<i>PacketBytes</i>	bytearray: packet data bytes
<i>UserCode</i>	int: command user code

Methods Summary

<i>ExtractSensorIdStr()</i>	extract sensor id string from the return packet
<i>ExtractSpectrumInfo()</i>	extract spectrum info from the return packet
<i>ExtractWavelengthInfo()</i>	extract wavelength info from the return packet
<i>ExtractXYZInfo()</i>	extract XYZ info from the return packet

Attributes Documentation

CmdCode

CmdCodeEnum: command function code

IsPacketValid

bool: check if the packet is valid (True for valid; False for invalid)

PacketBytes

bytearray: packet data bytes

UserCode

int: command user code

Methods Documentation

ExtractSensorIdStr ()

extract sensor id string from the return packet

Returns: str: sensor id string (return None if the packet type mismatches)

ExtractSpectrumInfo ()

extract spectrum info from the return packet

Returns: SpectrumInfo: spectrum info (return None if the packet type mismatches)

ExtractWavelengthInfo ()

extract wavelength info from the return packet

Returns: WavelengthInfo: wavelength info (return None if the packet type mismatches)

ExtractXYZInfo ()

extract XYZ info from the return packet

Returns: XYZInfo: XYZ info (return None if the packet type mismatches)

1.1.4 SpectrumInfo

class NanoLambdaNSP32.**SpectrumInfo** (*packetBytes*)

Bases: object

spectrum info

Attributes Summary

<i>IntegrationTime</i>	int: integration time
<i>IsSaturated</i>	bool: saturation flag (True for saturated; False for not saturated)
<i>NumOfPoints</i>	int: num of points
<i>Spectrum</i>	tuple: spectrum data
<i>X</i>	float: X
<i>Y</i>	float: Y
<i>Z</i>	float: Z

Attributes Documentation

IntegrationTime

int: integration time

IsSaturated

bool: saturation flag (True for saturated; False for not saturated)

NumOfPoints

int: num of points

Spectrum

tuple: spectrum data

X

float: X

Y

float: Y

Z

float: Z

1.1.5 WavelengthInfo

class NanoLambdaNSP32.**WavelengthInfo** (*packetBytes*)

Bases: object

wavelength info

Attributes Summary

<i>NumOfPoints</i>	int: num of points
<i>Wavelength</i>	tuple: wavelength data

Attributes Documentation

NumOfPoints

int: num of points

Wavelength

tuple: wavelength data

1.1.6 XYZInfo

class NanoLambdaNSP32.**XYZInfo** (*packetBytes*)

Bases: object

XYZ info

Attributes Summary

<i>IntegrationTime</i>	int: integration time
<i>IsSaturated</i>	bool: saturation flag (True for saturated; False for not saturated)
<i>X</i>	float: X
<i>Y</i>	float: Y

Continued on next page

Table 8 – continued from previous page

Z	float: Z
---	----------

Attributes Documentation

IntegrationTime

int: integration time

IsSaturated

bool: saturation flag (True for saturated; False for not saturated)

X

float: X

Y

float: Y

Z

float: Z

INDICES AND TABLES

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

PYTHON MODULE INDEX

n

NanoLambdaNSP32, [1](#)

A

AcqSpectrum (*NanoLambdaNSP32.CmdCodeEnum attribute*), 1
 AcqSpectrum() (*NanoLambdaNSP32.NSP32 method*), 2
 AcqXYZ (*NanoLambdaNSP32.CmdCodeEnum attribute*), 1
 AcqXYZ() (*NanoLambdaNSP32.NSP32 method*), 2

C

CmdCode (*NanoLambdaNSP32.ReturnPacket attribute*), 4
 CmdCodeEnum (*class in NanoLambdaNSP32*), 1

E

ExtractSensorIdStr() (*NanoLambdaNSP32.ReturnPacket method*), 4
 ExtractSpectrumInfo() (*NanoLambdaNSP32.ReturnPacket method*), 4
 ExtractWavelengthInfo() (*NanoLambdaNSP32.ReturnPacket method*), 4
 ExtractXYZInfo() (*NanoLambdaNSP32.ReturnPacket method*), 4

G

GetSensorId (*NanoLambdaNSP32.CmdCodeEnum attribute*), 1
 GetSensorId() (*NanoLambdaNSP32.NSP32 method*), 3
 GetSpectrum (*NanoLambdaNSP32.CmdCodeEnum attribute*), 1
 GetWavelength (*NanoLambdaNSP32.CmdCodeEnum attribute*), 2
 GetWavelength() (*NanoLambdaNSP32.NSP32 method*), 3
 GetXYZ (*NanoLambdaNSP32.CmdCodeEnum attribute*), 2

H

Hello (*NanoLambdaNSP32.CmdCodeEnum attribute*), 2
 Hello() (*NanoLambdaNSP32.NSP32 method*), 3

I

IntegrationTime (*NanoLambdaNSP32.SpectrumInfo attribute*), 5
 IntegrationTime (*NanoLambdaNSP32.XYZInfo attribute*), 6
 IsPacketValid (*NanoLambdaNSP32.ReturnPacket attribute*), 4
 IsSaturated (*NanoLambdaNSP32.SpectrumInfo attribute*), 5
 IsSaturated (*NanoLambdaNSP32.XYZInfo attribute*), 6

N

NanoLambdaNSP32 (*module*), 1
 NSP32 (*class in NanoLambdaNSP32*), 2
 NumOfPoints (*NanoLambdaNSP32.SpectrumInfo attribute*), 5
 NumOfPoints (*NanoLambdaNSP32.WavelengthInfo attribute*), 5

O

OnReturnByteReceived() (*NanoLambdaNSP32.NSP32 method*), 3
 OnReturnBytesReceived() (*NanoLambdaNSP32.NSP32 method*), 3

P

PacketBytes (*NanoLambdaNSP32.ReturnPacket attribute*), 4
 Prefix0 (*NanoLambdaNSP32.CmdCodeEnum attribute*), 2
 Prefix1 (*NanoLambdaNSP32.CmdCodeEnum attribute*), 2

R

ReturnPacket (*class in NanoLambdaNSP32*), 3

S

Spectrum (*NanoLambdaNSP32.SpectrumInfo attribute*), 5
 SpectrumInfo (*class in NanoLambdaNSP32*), 4

Standby (*NanoLambdaNSP32.CmdCodeEnum* attribute), [2](#)

Standby () (*NanoLambdaNSP32.NSP32* method), [3](#)

U

Unknown (*NanoLambdaNSP32.CmdCodeEnum* attribute), [2](#)

UserCode (*NanoLambdaNSP32.ReturnPacket* attribute), [4](#)

W

Wavelength (*NanoLambdaNSP32.WavelengthInfo* attribute), [5](#)

WavelengthInfo (class in *NanoLambdaNSP32*), [5](#)

X

X (*NanoLambdaNSP32.SpectrumInfo* attribute), [5](#)

X (*NanoLambdaNSP32.XYZInfo* attribute), [6](#)

XYZInfo (class in *NanoLambdaNSP32*), [5](#)

Y

Y (*NanoLambdaNSP32.SpectrumInfo* attribute), [5](#)

Y (*NanoLambdaNSP32.XYZInfo* attribute), [6](#)

Z

Z (*NanoLambdaNSP32.SpectrumInfo* attribute), [5](#)

Z (*NanoLambdaNSP32.XYZInfo* attribute), [6](#)