CONEX-PSD

Two-Axis Position
& Power Sensing device

Command Interface
Manual

V3.0.x

Table of Contents

Two-Axis Position & Power Sensing Device CONEX-PSD

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide the method syntax of each command to communicate with the CONEX-PSD device.

1.2 Overview

The Command Interface is the wrapper class that maintains a list of CONEX-PSD instruments. It exposes methods to communicate with any CONEX-PSD device.

NOTE

Each function name is defined with the command code "AA".

For each command function, refer to the CONEX-PSD programmer's manual.

2.0 Command Interface

2.1 Constructor

ConexPSD()

The constructor is used to create an instance of the CONEX-PSD device.

2.2 Functions

2.2.1 General

♦ CloseInstrument

Syntax

int CloseInstrument()

return: 0 = successful or -1 = failure

Description

This function allows closing communication with the selected device. If the closing failed, the returned code is -1.

♦ GetDevices

Syntax

string[] GetDevices()

return: list of connected devices available to communicate

Description

This function returns the list of connected devices available to communicate.

♦ OpenInstrument

Syntax

int OpenInstrument(string strDeviceKey)

string strDeviceKey: device key

return: 0 = successful or -1 = failure

Description

This function allows opening communication with the selected device. If the opening failed, the returned code is -1.

♦ **WriteToInstrument**

Syntax

int WriteToInstrument(string command, ref string response, int stage)

command: Instrument command

response: Response of the command

stage: Instrument Stage

return:

Description

This Overridden function Queries or writes the command given by the user to the instrument.

2.2.2 Commands

♦ **GP**

Syntax

int GP(int controllerAddress, out double PositionX, out double PositionY, out double LaserPower, out string errstring)

controllerAddress: controllerAddress

PositionX: PositionX

PositionY: PositionY

LaserPower: LaserPower

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous GP Get command which is used to Get X, Y positions and laser power level.

♦ **ID_Get**

Syntax

int ID_Get(int controllerAddress, out string SensorIdentifier, out string errstring)

controllerAddress: controllerAddress

SensorIdentifier: SensorIdentifier

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous ID Get command which is used to get sensor identifier.

♦ ID_Set

Syntax

int ID_Set(int controllerAddress, string SensorIdentifier, out string errstring)

controllerAddress: controllerAddress

SensorIdentifier: SensorIdentifier

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous ID Set command which is used to Set sensor identifier.

♦ IS_Get

Syntax

int IS_Get(int controllerAddress, out double Offset, out string errstring)

controllerAddress: controllerAddress

Offset: Offset

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IS Get command which is used to Get offset on ADC input SUM.

♦ IS_Set

Syntax

int IS_Set(int controllerAddress, double Offset, out string errstring)

controllerAddress: controllerAddress

Offset: Offset

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IS Set command which is used to Set offset on ADC input SUM.

♦ IX_Get

Syntax

int IX_Get(int controllerAddress, out double OffsetADC1, out string errstring)

controllerAddress: controllerAddress

OffsetADC1: OffsetADC1

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IX Get command which is used to get offset on ADC input X.

♦ IX_Set

Syntax

int IX_Set(int controllerAddress, double OffsetADC1, out string errstring)

controllerAddress: controllerAddress

OffsetADC1: OffsetADC1

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IX Set command which is used to set offset on ADC input X.

♦ IY_Get

Syntax

int IY_Get(int controllerAddress, out double OffsetADC2, out string errstring)

controllerAddress: controllerAddress

OffsetADC2: OffsetADC2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IY Get command which is used to get offset on ADC input Y.

♦ IY_Set

Syntax

int IY_Set(int controllerAddress, double OffsetADC2, out string errstring)

controllerAddress: controllerAddress

OffsetADC2: OffsetADC2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous IY Set command which is used to set offset on ADC input Y.

♦ LF_Get

Syntax

int LF_Get(int controllerAddress, out double Frequency, out string errstring)

controllerAddress: controllerAddress

Frequency: Frequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous LF Get command which is used to get low pass filter frequency.

♦ LF_Set

Syntax

int LF_Set(int controllerAddress, double Frequency, out string errstring)

controllerAddress: controllerAddress

Frequency: Frequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous LF Set command which is used to Set low pass filter frequency.

♦ OF_Get

NOTE

This command is not used with the CONEX-PSD9.

Syntax

int OF_Get(int controllerAddress, out double Offset1, out double Offset2, out double Offset3, out double Offset4, out string errstring)

controllerAddress: controllerAddress

Offset1: Offset #1

Offset2: Offset #2

Offset3: Offset #3

Offset4: Offset #4

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous OF Get command which is used to get offsets. Refer to the CONEX-PSD Controller's manual to get the command description.

♦ OF_Set

NOTE

This command is not used with the CONEX-PSD9.

Syntax

int OF_Set(int controllerAddress, double Offset1, double Offset2, double Offset3, double Offset4, out string errstring)

controllerAddress: Controller's address

Offset1: Offset #1

Offset2: Offset #2

Offset3: Offset #3

Offset4: Offset #4

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous OF Set command which is used to set offsets. Refer to the CONEX-PSD Controller's manual to get the command description.

♦ PS_Get

Syntax

int PS_Get(int controllerAddress, out double Gain, out string errstring)

controllerAddress: controllerAddress

Gain: Gain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PS Get command which is used to Get gain on ADC input SUM.

♦ PS_Set

Syntax

int PS_Set(int controllerAddress, double Gain, out string errstring)

controllerAddress: controllerAddress

Gain: Gain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PS Set command which is used to Set gain on ADC input SUM.

♦ PX_Get

Syntax

int PX_Get(int controllerAddress, out double GainADC1, out string errstring)

controllerAddress: controllerAddress

GainADC1: GainADC1

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PX Get command which is used to Get gain on ADC input X.

♦ PX_Set

Syntax

int PX_Set(int controllerAddress, double GainADC1, out string errstring)

controllerAddress: controllerAddress

GainADC1: GainADC1

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PX Set command which is used to Set gain on ADC input X.

♦ PY_Get

Syntax

int PY_Get(int controllerAddress, out double GainADC2, out string errstring)

controllerAddress: controllerAddress

GainADC2: GainADC2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PY Get command which is used to Get gain on ADC input Y.

♦ PY_Set

Syntax

int PY_Set(int controllerAddress, double GainADC2, out string errstring)

controllerAddress: controllerAddress

GainADC2: GainADC2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PY Set command which is used to Set gain on ADC input Y.

♦ PW_Get

Syntax

int PW_Get(int controllerAddress, out int ConfigurationState, out string errstring)

controllerAddress: controllerAddress

ConfigurationState: ConfigurationState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PW Get command which is used to Enter/Leave CONFIGURATION state.

♦ PW_Set

Syntax

int PW_Set(int controllerAddress, int ConfigurationState, out string errstring)

controllerAddress: controllerAddress

ConfigurationState : ConfigurationState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous PW Set command which is used to Enter/Leave CONFIGURATION state.

♦ RA

Syntax

int RA(int controllerAddress, out double RawAnalogInput1, out double RawAnalogInput2, out string errstring)

controllerAddress: controllerAddress

RawAnalogInput1: RawAnalogInput1

RawAnalogInput2: RawAnalogInput2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous RA Get command which is used to get raw analog input values.

♦ RC

Syntax

int RC(int controllerAddress, out double CorrectedAnalogInput1, out double CorrectedAnalogInput2, out string errstring)

controllerAddress: controllerAddress

CorrectedAnalogInput1: CorrectedAnalogInput1

CorrectedAnalogInput2: CorrectedAnalogInput2

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous RC Get command which is used to Get corrected analog input values.

♦ RS

Syntax

int RS(int controllerAddress, out string errstring)

controllerAddress: controllerAddress

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous RS Set command which is used to Reset controller.

♦ RS485

Syntax

int RS485(int controllerAddress, out string errstring)

controllerAddress: controllerAddress

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous RS485 Set command which is used to Reset controller’s address to 1.

♦ SA_Get

Syntax

int SA_Get(int controllerAddress, out int Adress, out string errstring)

controllerAddress: controllerAddress

Adress: Adress

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous SA Get command which is used to get controller's RS-485 address.

♦ SA_Set

Syntax

int SA_Set(int controllerAddress, int Address, out string errstring)

controllerAddress: controllerAddress

Address : Address

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous SA Set command which is used to Set controller's RS-485 address.

♦ TB_Get

Syntax

int TB_Get(int controllerAddress, string inErrorCode, string outErrorCode, out string errstring)

controllerAddress: controllerAddress

inErrorCode: Input error code (optional)

outErrorCode: Output error description

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous TB Get command which is used to Get command error string.

♦ TE

Syntax

int TE(int controllerAddress, out string LastCommandError, out string errstring)

controllerAddress: controllerAddress

LastCommandError: LastCommandError

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous TE Get command which is used to Get last command error.

♦ TS

Syntax

int TS(int controllerAddress, out string ErrorCode, out string StatusCode, out string errstring)

controllerAddress: controllerAddress

ErrorCode: ErrorCode

StatusCode: StatusCode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous TS Get command which is used to Get positioner error and controller state.

♦ VE

Syntax

int VE(int controllerAddress, out string ControllerVersion, out string errstring)

controllerAddress: controllerAddress

ControllerVersion: ControllerVersion

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchrounous VE Get command which is used to Get controller revision information.

3.0 Python Example

```
#==================================================================
#Initialization Start
#The script within Initialization Start and Initialization End is needed for properly
#initializing Command Interface for Conex-PSD instrument.
#The user should copy this code as is and specify correct paths here.
import sys

#Command Interface DLL can be found here.
print "Adding location of Newport.CONEXPSD.CommandInterface.dll to sys.path"
sys.path.append(r'C:\Program Files (x86)\Newport\MotionControl\CONEX-PSD\Bin')

# The CLR module provide functions for interacting with the underlying
# .NET runtime
import clr
# Add reference to assembly and import names from namespace
clr.AddReferenceToFile("Newport.CONEXPSD.CommandInterface.dll")
from CommandInterface import *

import System
#==================================================================
# Instrument Initialization
# The key should have double slashes since
# (one of them is escape character)
instrument="CONEX-PSD (A6T7NSPR)"
print 'Instrument Key=>', instrument

# create a device instance
PSD = ConexPSD()

componentID = PSD. OpenInstrument(instrument);
print 'componentID=>', componentID

# Get analog output #1 value
result, X, Y, LaserPower, errString = PSD.GP_Get(1)
if result == 0 :
  print 'X =>', X
  print 'Y =>', Y
  print 'Laser power =>', LaserPower
else:
  print 'Error=>',errString
```

```
# Get controller revision information
result, response, errString = PSD.VE(1)
if result == 0 :
   print 'controller revision=>', response
else:
   print 'Error=>',errString

# Get last command error
result, response, errString = PSD.TE(1)
if result == 0 :
   print 'Last command error =>', response
else:
   print 'Error=>',errString

# Unregister device
PSD. CloseInstrument();
```

Service Form

Your Local Representative

Tel.: ___________________

Fax:___________________

Name: ____

Company:____

Address:____

Country:____

P.O. Number:____

Item(s) Being Returned:____

Model#: ____

Return authorization #: ____

(Please obtain prior to return of item)

Date: ____

Phone Number: ____

Fax Number: ____

Serial #: ____

Description: ____

Reasons of return of goods (please list any specific problems): ____

Visit Newport Online at:

www.newport.com

North America & Asia
Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA

Sales
Tel.: (800) 222-6440
e-mail: sales@newport.com

Technical Support
Tel.: (800) 222-6440
e-mail: tech@newport.com

Service, RMAs & Returns
Tel.: (800) 222-6440
e-mail: service@newport.com

Europe
MICRO-CONTROLE Spectra-Physics S.A.S
9, rue du Bois Sauvage
91055 Évry CEDEX
France

Sales
Tel.: +33 (0)1.60.91.68.68
e-mail: france@newport.com

Technical Support
e-mail: tech_europe@newport.com

Service & Returns
Tel.: +33 (0)2.38.40.51.55