HXP

Hexapod Motion Controller

Newport®

Command Interface Manual

Intaller Pack Version #30002

Original instructions.

Table of Contents

Hexapod Motion Controller HXP

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide the method syntax of each command to communicate with the HXP device exposed in assembly Newport. HXP.CommandInterface.dll.

1.2 Overview

Command Interface DLL is a generic name that refers to a DLL used to communicate with an instrument. Typically this DLL exposes all the commands the instrument supports. These commands are exposed as function calls of the Command Interface DLL.

Newport. HXP.CommandInterface.dll is the assembly used for communicating with HXP instrument. This assembly gets installed when HXP applet is installed.

NOTE

Each function name is defined with the command code “AA”.

For each command function, refer to the HXP programmer’s manual.

1.3 Location

Newport. HXP.CommandInterface.dll is located at

C:\Program Files (x86)\Newport\MotionControl\HXP\Bin\Newport.HXP.CommandInterface.dll

1.4 Possible Uses of Newport. HXP.CommandInterface.dll

Newport. HXP.CommandInterface.dll is used by HXP applet for communicating with HXP instrument. The same DLL can be used as a reusable software component for creating Python script or for creating LabVIEW VIs.

2.0 Command Interface

2.1 Constructor

Syntax

HXP()

Decription

The constructor is used to create an instance of the HXP device.

2.2 Functions

2.2.1 OpenInstrument

Syntax

C# prototype

int OpenInstrument (string Address, int port)

Python prototype

[Return] OpenInstrument (Address, Port)

Parameter

Input parameters

None

Output parameters

(string) Address: TCP IP address controller

(int) Port: TCP IP port

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to open the communication with the controller. A socket is created and the TCP IP connection is established. Refer to the HXP Programmer's manual to get the command description.

2.2.2 CloseInstrument

Syntax

C# prototype

int CloseInstrument ()

Python prototype

[Return] CloseInstrument ()

Parameter

Input parameters

None

Output parameters

None

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to close the communication with the controller. The current socket is deleted and the TCP IP connection is closed.

Refer to HXP Programmer's manual to get the command description.

2.2.3 ControllerMotionKernelTimeLoadGet

Syntax

C# prototype

int ControllerMotionKernelTimeLoadGet(out double CPUTotalLoadRatio, out double CPUCorrectorLoadRatio, out double CPUProfilerLoadRatio, out double CPUServitudesLoadRatio, out string errstring)

Python prototype

[CPUTotalLoadRatio, CPUCorrectorLoadRatio, CPUProfilerLoadRatio, CPUServitudesLoadRatio, errstring] ControllerMotionKernelTimeLoadGet ()

Parameters

Input parameters

None

Output parameters

(double) CPUTotalLoadRatio: CPUTotalLoadRatio

(double) CPUCorrectorLoadRatio: CPUCorrectorLoadRatio

(double) CPUProfilerLoadRatio: CPUProfilerLoadRatio

(double) CPUServitudesLoadRatio: CPUServitudesLoadRatio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ControllerMotionKernelTimeLoadGet command which is used to Get controller motion kernel time load. Refer to the XPS Programmer's manual to get the command description.

2.2.4 ElapsedTimeGet

Syntax

C# prototype

int ElapsedTimeGet(out double ElapsedTime, out string errstring)

Python prototype

[ElapsedTime, errstring] ElapsedTimeGet ()

Parameters

Input parameters

None

Output parameters

(double) ElapsedTime: ElapsedTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ElapsedTimeGet command which is used to Return elapsed time from controller power on. Refer to the XPS Programmer's manual to get the command description.

2.2.5 ErrorStringGet

Syntax

C# prototype

int ErrorStringGet(Int32 ErrorCode, out string ErrorString, out string errstring)

Python prototype

[ErrorString, errstring] ErrorStringGet (ErrorCode)

Parameters

Input parameters

(Int32) ErrorCode: ErrorCode

Output parameters

(string) ErrorString: ErrorString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ErrorStringGet command which is used to Return the error string corresponding to the error code. Refer to the XPS Programmer's manual to get the command description.

2.2.6 FirmwareVersionGet

Syntax

C# prototype

int FirmwareVersionGet(out string Version, out string errstring)

Python prototype

[Version, errstring] FirmwareVersionGet ()

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous FirmwareVersionGet command which is used to Return firmware version. Refer to the XPS Programmer's manual to get the command description.

2.2.7 TCLScriptExecute

Syntax

C# prototype

int TCLScriptExecute(string TCLFileName, string TaskName, string ParametersList, out string errstring)

Python prototype

[errstring] TCLScriptExecute (TCLFileName, TaskName, ParametersList)

Parameters

Input parameters

(string) TCLFileName: TCLFileName

(string) TaskName: TaskName

(string) ParametersList: ParametersList

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TCLScriptExecute command which is used to Execute a TCL script from a TCL file. Refer to the XPS Programmer's manual to get the command description.

2.2.8 TCLScriptExecuteAndWait

Syntax

C# prototype

int TCLScriptExecuteAndWait(string TCLFileName, string TaskName, string InputParametersList, out string OutputParametersList, out string errstring)

Python prototype

[OutputParametersList, errstring] TCLScriptExecuteAndWait (TCLFileName, TaskName, InputParametersList)

Parameters

Input parameters

(string) TCLFileName: TCLFileName

(string) TaskName: TaskName

(string) InputParametersList: InputParametersList

Output parameters

(string) OutputParametersList: OutputParametersList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TCLScriptExecuteAndWait command which is used to Execute a TCL script from a TCL file and wait the end of execution to return. Refer to the XPS Programmer's manual to get the command description.

2.2.9 TCLScriptKill

Syntax

C# prototype

int TCLScriptKill(string TaskName, out string errstring)

Python prototype

[errstring] TCLScriptKill (TaskName)

Parameters

Input parameters

(string) TaskName: TaskName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TCLScriptKill command which is used to Kill TCL Task. Refer to the XPS Programmer's manual to get the command description.

2.2.10 TCLScriptKillAll

Syntax

C# prototype

int TCLScriptKillAll(out string errstring)

Python prototype

[errstring] TCLScriptKillAll ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TCLScriptKillAll command which is used to Kill all TCL Tasks. Refer to the XPS Programmer's manual to get the command description.

2.2.11 TimerGet

Syntax

C# prototype

int TimerGet(string TimerName, out Int32 FrequencyTicks, out string errstring)

Python prototype

[FrequencyTicks, errstring] TimerGet (TimerName)

Parameters

Input parameters

(string) TimerName: TimerName

Output parameters

(Int32_i) FrequencyTicks: FrequencyTicks

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TimerGet command which is used to Get a timer. Refer to the XPS Programmer's manual to get the command description.

2.2.12 TimerSet

Syntax

C# prototype

int TimerSet(string TimerName, Int32 FrequencyTicks, out string errstring)

Python prototype

[errstring] TimerSet (TimerName, FrequencyTicks)

Parameters

Input parameters

(string) TimerName: TimerName

(Int32) FrequencyTicks: FrequencyTicks

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TimerSet command which is used to Set a timer. Refer to the XPS Programmer's manual to get the command description.

2.2.13 Reboot

Syntax

C# prototype

int Reboot(out string errstring)

Python prototype

[errstring] Reboot ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous Reboot command which is used to Reboot the controller. Refer to the XPS Programmer's manual to get the command description.

2.2.14 Login

Syntax

C# prototype

int Login(string Name, string Password, out string errstring)

Python prototype

[errstring] Login (Name, Password)

Parameters

Input parameters

(string) Name: Name

(string) Password: Password

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous Login command which is used to Log in. Refer to the XPS Programmer's manual to get the command description.

2.2.15 CloseAllOtherSockets

Syntax

C# prototype

int CloseAllOtherSockets(out string errstring)

Python prototype

[errstring] CloseAllOtherSockets ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous CloseAllOtherSockets command which is used to Close all socket beside the one used to send this command. Refer to the XPS Programmer's manual to get the command description.

2.2.16 EventAdd

Syntax

C# prototype

int EventAdd(string PositionerName, string EventName, string EventParameter, string ActionName, string ActionParameter1, string ActionParameter2, string ActionParameter3, out string errstring)

Python prototype

[errstring] EventAdd (PositionerName, EventName, EventParameter, ActionName, ActionParameter1, ActionParameter2, ActionParameter3)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

(string) ActionName: ActionName

(string) ActionParameter1: ActionParameter1

(string) ActionParameter2: ActionParameter2

(string) ActionParameter3: ActionParameter3

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventAdd command which is used to ** OBSOLETE ** Add an event. Refer to the XPS Programmer's manual to get the command description.

2.2.17 EventGet

Syntax

C# prototype

int EventGet(string PositionerName, out string EventsAndActionsList, out string errstring)

Python prototype

[EventsAndActionsList, errstring] EventGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) EventsAndActionsList: EventsAndActionsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventGet command which is used to ** OBSOLETE ** Read events and actions list. Refer to the XPS Programmer's manual to get the command description.

2.2.18 EventRemove

Syntax

C# prototype

int EventRemove(string PositionerName, string EventName, string EventParameter, out string errstring)

Python prototype

[errstring] EventRemove (PositionerName, EventName, EventParameter)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventRemove command which is used to ** OBSOLETE ** Delete an event. Refer to the XPS Programmer's manual to get the command description.

2.2.19 EventWait

Syntax

C# prototype

int EventWait(string PositionerName, string EventName, string EventParameter, out string errstring)

Python prototype

[errstring] EventWait (PositionerName, EventName, EventParameter)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventWait command which is used to ** OBSOLETE ** Wait an event. Refer to the XPS Programmer's manual to get the command description.

2.2.20 EventExtendedConfigurationTriggerSet

Syntax

C# prototype

int EventExtendedConfigurationTriggerSet(string[] ExtendedEventName, string[] EventParameter1, string[] EventParameter2, string[] EventParameter3, string[] EventParameter4, out string errstring)

Python prototype

[errstring] EventExtendedConfigurationTriggerSet (ExtendedEventName, EventParameter1, EventParameter2, EventParameter3, EventParameter4)

Parameters

Input parameters

(string[]) ExtendedEventName: ExtendedEventName

(string[]) EventParameter1: EventParameter1

(string[]) EventParameter2: EventParameter2

(string[]) EventParameter3: EventParameter3

(string[]) EventParameter4: EventParameter4

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedConfigurationTriggerSet command which is used to Configure one or several events. Refer to the XPS Programmer's manual to get the command description.

2.2.21 EventExtendedConfigurationTriggerGet

Syntax

C# prototype

int EventExtendedConfigurationTriggerGet(out string EventTriggerConfiguration, out string errstring)

Python prototype

[EventTriggerConfiguration, errstring] EventExtendedConfigurationTriggerGet ()

Parameters

Input parameters

None

Output parameters

(string) EventTriggerConfiguration: EventTriggerConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedConfigurationTriggerGet command which is used to Read the event configuration. Refer to the XPS Programmer's manual to get the command description.

2.2.22 EventExtendedConfigurationActionSet

Syntax

C# prototype

int EventExtendedConfigurationActionSet(string[] ExtendedActionName, string[] ActionParameter1, string[] ActionParameter2, string[] ActionParameter3, string[] ActionParameter4, out string errstring)

Python prototype

[errstring] EventExtendedConfigurationActionSet (ExtendedActionName, ActionParameter1, ActionParameter2, ActionParameter3, ActionParameter4)

Parameters

Input parameters

(string[]) ExtendedActionName: ExtendedActionName

(string[]) ActionParameter1: ActionParameter1

(string[]) ActionParameter2: ActionParameter2

(string[]) ActionParameter3: ActionParameter3

(string[]) ActionParameter4: ActionParameter4

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedConfigurationActionSet command which is used to Configure one or several actions. Refer to the XPS Programmer's manual to get the command description.

2.2.23 EventExtendedConfigurationActionGet

Syntax

C# prototype

int EventExtendedConfigurationActionGet(out string ActionConfiguration, out string errstring)

Python prototype

[ActionConfiguration, errstring] EventExtendedConfigurationActionGet ()

Parameters

Input parameters

None

Output parameters

(string) ActionConfiguration: ActionConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedConfigurationActionGet command which is used to Read the action configuration. Refer to the XPS Programmer's manual to get the command description.

2.2.24 EventExtendedStart

Syntax

C# prototype

int EventExtendedStart(out Int32 ID, out string errstring)

Python prototype

[ID, errstring] EventExtendedStart ()

Parameters

Input parameters

None

Output parameters

(Int32_i) ID: ID

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedStart command which is used to Launch the last event and action configuration and return an ID. Refer to the XPS Programmer's manual to get the command description.

2.2.25 EventExtendedAllGet

Syntax

C# prototype

int EventExtendedAllGet(out string EventActionConfigurations, out string errstring)

Python prototype

[EventActionConfigurations, errstring] EventExtendedAllGet ()

Parameters

Input parameters

None

Output parameters

(string) EventActionConfigurations: EventActionConfigurations

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedAllGet command which is used to Read all event and action configurations. Refer to the XPS Programmer's manual to get the command description.

2.2.26 EventExtendedGet

Syntax

C# prototype

int EventExtendedGet(Int32 ID, out string EventTriggerConfiguration, out string ActionConfiguration, out string errstring)

Python prototype

[EventTriggerConfiguration, ActionConfiguration, errstring] EventExtendedGet (ID)

Parameters

Input parameters

(Int32) ID: ID

Output parameters

(string) EventTriggerConfiguration: EventTriggerConfiguration

(string) ActionConfiguration: ActionConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedGet command which is used to Read the event and action configuration defined by ID. Refer to the XPS Programmer's manual to get the command description.

2.2.27 EventExtendedRemove

Syntax

C# prototype

int EventExtendedRemove(Int32 ID, out string errstring)

Python prototype

[errstring] EventExtendedRemove (ID)

Parameters

Input parameters

(Int32) ID: ID

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedRemove command which is used to Remove the event and action configuration defined by ID. Refer to the XPS Programmer's manual to get the command description.

2.2.28 EventExtendedWait

Syntax

C# prototype

int EventExtendedWait(out string errstring)

Python prototype

[errstring] EventExtendedWait ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventExtendedWait command which is used to Wait events from the last event configuration. Refer to the XPS Programmer's manual to get the command description.

2.2.29 GatheringConfigurationGet

Syntax

C# prototype

int GatheringConfigurationGet(out string Type, out string errstring)

Python prototype

[Type, errstring] GatheringConfigurationGet ()

Parameters

Input parameters

None

Output parameters

(string) Type: Type

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringConfigurationGet command which is used to Read different mnemonique type. Refer to the XPS Programmer's manual to get the command description.

2.2.30 GatheringConfigurationSet

Syntax

C# prototype

int GatheringConfigurationSet(string[] Type, out string errstring)

Python prototype

[errstring] GatheringConfigurationSet (Type)

Parameters

Input parameters

(string[]) Type: Type

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringConfigurationSet command which is used to Configuration acquisition. Refer to the XPS Programmer's manual to get the command description.

2.2.31 GatheringCurrentNumberGet

Syntax

C# prototype

int GatheringCurrentNumberGet(out Int32 CurrentNumber, out Int32 MaximumSamplesNumber, out string errstring)

Python prototype

[CurrentNumber, MaximumSamplesNumber, errstring] GatheringCurrentNumberGet ()

Parameters

Input parameters

None

Output parameters

(Int32_i) CurrentNumber: CurrentNumber

(Int32_i) MaximumSamplesNumber: MaximumSamplesNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringCurrentNumberGet command which is used to Maximum number of samples and current number during acquisition. Refer to the XPS Programmer's manual to get the command description.

2.2.32 GatheringStopAndSave

Syntax

C# prototype

int GatheringStopAndSave(out string errstring)

Python prototype

[errstring] GatheringStopAndSave ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringStopAndSave command which is used to Stop acquisition and save data. Refer to the XPS Programmer's manual to get the command description.

2.2.33 GatheringDataAcquire

Syntax

C# prototype

int GatheringDataAcquire(out string errstring)

Python prototype

[errstring] GatheringDataAcquire ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringDataAcquire command which is used to Acquire a configured data. Refer to the XPS Programmer's manual to get the command description.

2.2.34 GatheringDataGet

Syntax

C# prototype

int GatheringDataGet(Int32 IndexPoint, out string DataBufferLine, out string errstring)

Python prototype

[DataBufferLine, errstring] GatheringDataGet (IndexPoint)

Parameters

Input parameters

(Int32) IndexPoint: IndexPoint

Output parameters

(string) DataBufferLine: DataBufferLine

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringDataGet command which is used to Get a data line from gathering buffer. Refer to the XPS Programmer's manual to get the command description.

2.2.35 GatheringDataMultipleLinesGet

Syntax

C# prototype

int GatheringDataMultipleLinesGet(Int32 IndexPoint, Int32 NumberOfLines, out string DataBufferLine, out string errstring)

Python prototype

[DataBufferLine, errstring] GatheringDataMultipleLinesGet (IndexPoint, NumberOfLines)

Parameters

Input parameters

(Int32) IndexPoint: IndexPoint

(Int32) NumberOfLines: NumberOfLines

Output parameters

(string) DataBufferLine: DataBufferLine

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringDataMultipleLinesGet command which is used to Get multiple data lines from gathering buffer. Refer to the XPS Programmer's manual to get the command description.

2.2.36 GatheringReset

Syntax

C# prototype

int GatheringReset(out string errstring)

Python prototype

[errstring] GatheringReset ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringReset command which is used to Empty the gathered data in memory to start new gathering from scratch. Refer to the XPS Programmer's manual to get the command description.

2.2.37 GatheringRun

Syntax

C# prototype

int GatheringRun(Int32 DataNumber, Int32 Divisor, out string errstring)

Python prototype

[errstring] GatheringRun (DataNumber, Divisor)

Parameters

Input parameters

(Int32) DataNumber: DataNumber

(Int32) Divisor: Divisor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringRun command which is used to Start a new gathering. Refer to the XPS Programmer's manual to get the command description.

2.2.38 GatheringStop

Syntax

C# prototype

int GatheringStop(out string errstring)

Python prototype

[errstring] GatheringStop ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringStop command which is used to Stop the data gathering . Refer to the XPS Programmer's manual to get the command description.

2.2.39 GatheringExternalConfigurationSet

Syntax

C# prototype

int GatheringExternalConfigurationSet(string[] Type, out string errstring)

Python prototype

[errstring] GatheringExternalConfigurationSet (Type)

Parameters

Input parameters

(string[]) Type: Type

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExternalConfigurationSet command which is used to Configuration acquisition. Refer to the XPS Programmer's manual to get the command description.

2.2.40 GatheringExternalConfigurationGet

Syntax

C# prototype

int GatheringExternalConfigurationGet(out string Type, out string errstring)

Python prototype

[Type, errstring] GatheringExternalConfigurationGet ()

Parameters

Input parameters

None

Output parameters

(string) Type: Type

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExternalConfigurationGet command which is used to Read different mnemonique type. Refer to the XPS Programmer's manual to get the command description.

2.2.41 GatheringExternalCurrentNumberGet

Syntax

C# prototype

int GatheringExternalCurrentNumberGet(out Int32 CurrentNumber, out Int32 MaximumSamplesNumber, out string errstring)

Python prototype

[CurrentNumber, MaximumSamplesNumber, errstring] GatheringExternalCurrentNumberGet ()

Parameters

Input parameters

None

Output parameters

(Int32_i) CurrentNumber: CurrentNumber

(Int32_i) MaximumSamplesNumber: MaximumSamplesNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExternalCurrentNumberGet command which is used to Maximum number of samples and current number during acquisition. Refer to the XPS Programmer's manual to get the command description.

2.2.42 GatheringExternalStopAndSave

Syntax

C# prototype

int GatheringExternalStopAndSave(out string errstring)

Python prototype

[errstring] GatheringExternalStopAndSave ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExternalStopAndSave command which is used to Stop acquisition and save data. Refer to the XPS Programmer's manual to get the command description.

2.2.43 GlobalArrayGet

Syntax

C# prototype

int GlobalArrayGet(Int32 Number, out string ValueString, out string errstring)

Python prototype

[ValueString, errstring] GlobalArrayGet (Number)

Parameters

Input parameters

(Int32) Number: Number

Output parameters

(string) ValueString: ValueString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GlobalArrayGet command which is used to Get global array value. Refer to the XPS Programmer's manual to get the command description.

2.2.44 GlobalArraySet

Syntax

C# prototype

int GlobalArraySet(Int32 Number, string ValueString, out string errstring)

Python prototype

[errstring] GlobalArraySet (Number, ValueString)

Parameters

Input parameters

(Int32) Number: Number

(string) ValueString: ValueString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GlobalArraySet command which is used to Set global array value. Refer to the XPS Programmer's manual to get the command description.

2.2.45 DoubleGlobalArrayGet

Syntax

C# prototype

int DoubleGlobalArrayGet(Int32 Number, out double DoubleValue, out string errstring)

Python prototype

[DoubleValue, errstring] DoubleGlobalArrayGet (Number)

Parameters

Input parameters

(Int32) Number: Number

Output parameters

(double) DoubleValue: DoubleValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous DoubleGlobalArrayGet command which is used to Get double global array value. Refer to the XPS Programmer's manual to get the command description.

2.2.46 DoubleGlobalArraySet

Syntax

C# prototype

int DoubleGlobalArraySet(Int32 Number, double DoubleValue, out string errstring)

Python prototype

[errstring] DoubleGlobalArraySet (Number, DoubleValue)

Parameters

Input parameters

(Int32) Number: Number

(double) DoubleValue: DoubleValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous DoubleGlobalArraySet command which is used to Set double global array value. Refer to the XPS Programmer's manual to get the command description.

2.2.47 GPIOAnalogGet

Syntax

C# prototype

int GPIOAnalogGet(string[] GPIOName, out double[] AnalogValue, out string errstring)

Python prototype

[AnalogValue, errstring] GPIOAnalogGet (GPIOName)

Parameters

Input parameters

(string[]) GPIOName: GPIOName

Output parameters

(double[]) AnalogValue: AnalogValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIOAnalogGet command which is used to Read analog input or analog output for one or few input. Refer to the XPS Programmer's manual to get the command description.

2.2.48 GPIOAnalogSet

Syntax

C# prototype

int GPIOAnalogSet(string[] GPIOName, double[] AnalogOutputValue, out string errstring)

Python prototype

[errstring] GPIOAnalogSet (GPIOName, AnalogOutputValue)

Parameters

Input parameters

(string[]) GPIOName: GPIOName

(double[]) AnalogOutputValue: AnalogOutputValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIOAnalogSet command which is used to Set analog output for one or few output. Refer to the XPS Programmer's manual to get the command description.

2.2.49 GPIOAnalogGainGet

Syntax

C# prototype

int GPIOAnalogGainGet(string[] GPIOName, out Int32[] AnalogInputGainValue, out string errstring)

Python prototype

[AnalogInputGainValue, errstring] GPIOAnalogGainGet (GPIOName)

Parameters

Input parameters

(string[]) GPIOName: GPIOName

Output parameters

(Int32_i[]) AnalogInputGainValue: AnalogInputGainValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIOAnalogGainGet command which is used to Read analog input gain . Refer to the XPS Programmer's manual to get the command description.

2.2.50 GPIOAnalogGainSet

Syntax

C# prototype

int GPIOAnalogGainSet(string[] GPIOName, Int32[] AnalogInputGainValue, out string errstring)

Python prototype

[errstring] GPIOAnalogGainSet (GPIOName, AnalogInputGainValue)

Parameters

Input parameters

(string[]) GPIOName: GPIOName

(Int32) AnalogInputGainValue: AnalogInputGainValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIOAnalogGainSet command which is used to Set analog input gain . Refer to the XPS Programmer's manual to get the command description.

2.2.51 GPIODigitalGet

Syntax

C# prototype

int GPIODigitalGet(string GPIOName, out UInt16 DigitalValue, out string errstring)

Python prototype

[DigitalValue, errstring] GPIODigitalGet (GPIOName)

Parameters

Input parameters

(string) GPIOName: GPIOName

Output parameters

(UInt16) DigitalValue: DigitalValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIODigitalGet command which is used to Read digital output or digital input . Refer to the XPS Programmer's manual to get the command description.

2.2.52 GPIODigitalSet

Syntax

C# prototype

int GPIODigitalSet(string GPIOName, UInt16 Mask, UInt16 DigitalOutputValue, out string errstring)

Python prototype

[errstring] GPIODigitalSet (GPIOName, Mask, DigitalOutputValue)

Parameters

Input parameters

(string) GPIOName: GPIOName

(UInt16) Mask: Mask

(UInt16) DigitalOutputValue: DigitalOutputValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GPIODigitalSet command which is used to Set Digital Output for one or few output TTL. Refer to the XPS Programmer's manual to get the command description.

2.2.53 GroupCorrectorOutputGet

Syntax

C# prototype

int GroupCorrectorOutputGet(string GroupName, out double[] CorrectorOutput, Int32 nbItems, out string errstring)

Python prototype

[CorrectorOutput, errstring] GroupCorrectorOutputGet (GroupName, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) CorrectorOutput: CorrectorOutput

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupCorrectorOutputGet command which is used to Return corrector outputs. Refer to the XPS Programmer's manual to get the command description.

2.2.54 GroupHomeSearch

Syntax

C# prototype

int GroupHomeSearch(string GroupName, out string errstring)

Python prototype

[errstring] GroupHomeSearch (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupHomeSearch command which is used to Start home search sequence. Refer to the XPS Programmer's manual to get the command description.

2.2.55 GroupHomeSearchAndRelativeMove

Syntax

C# prototype

int GroupHomeSearchAndRelativeMove(string GroupName, double[] TargetDisplacement, Int32 nbItems, out string errstring)

Python prototype

[errstring] GroupHomeSearchAndRelativeMove (GroupName, TargetDisplacement, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetDisplacement: TargetDisplacement

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupHomeSearchAndRelativeMove command which is used to Start home search sequence and execute a displacement. Refer to the XPS Programmer's manual to get the command description.

2.2.56 GroupReadyAtPosition

Syntax

C# prototype

int GroupReadyAtPosition(string GroupName, double EncoderPosition1, double EncoderPosition2, double EncoderPosition3, double EncoderPosition4, double EncoderPosition5, double EncoderPosition6, out string errstring)

Python prototype

[errstring] GroupReadyAtPosition (GroupName, EncoderPosition1, EncoderPosition2, EncoderPosition3, EncoderPosition4, EncoderPosition5, EncoderPosition6)

Parameters

Input parameters

(string) GroupName: GroupName

(double) EncoderPosition1: EncoderPosition1

(double) EncoderPosition2: EncoderPosition2

(double) EncoderPosition3: EncoderPosition3

(double) EncoderPosition4: EncoderPosition4

(double) EncoderPosition5: EncoderPosition5

(double) EncoderPosition6: EncoderPosition6

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupReadyAtPosition command which is used to Go to READY state with the users positions . Refer to the XPS Programmer's manual to get the command description.

2.2.57 GroupInitialize

Syntax

C# prototype

int GroupInitialize(string GroupName, out string errstring)

Python prototype

[errstring] GroupInitialize (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupInitialize command which is used to Start the initialization. Refer to the XPS Programmer's manual to get the command description.

2.2.58 GroupInitializeWithEncoderCalibration

Syntax

C# prototype

int GroupInitializeWithEncoderCalibration(string GroupName, out string errstring)

Python prototype

[errstring] GroupInitializeWithEncoderCalibration (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupInitializeWithEncoderCalibration command which is used to Start the initialization with encoder calibration. Refer to the XPS Programmer's manual to get the command description.

2.2.59 GroupKill

Syntax

C# prototype

int GroupKill(string GroupName, out string errstring)

Python prototype

[errstring] GroupKill (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupKill command which is used to Kill the group. Refer to the XPS Programmer's manual to get the command description.

2.2.60 GroupMoveAbort

Syntax

C# prototype

int GroupMoveAbort(string GroupName, out string errstring)

Python prototype

[errstring] GroupMoveAbort (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupMoveAbort command which is used to Abort a move. Refer to the XPS Programmer's manual to get the command description.

2.2.61 GroupMoveAbsolute

Syntax

C# prototype

int GroupMoveAbsolute(string GroupName, double[] TargetPosition, Int32 nbItems, out string errstring)

Python prototype

[errstring] GroupMoveAbsolute (GroupName, TargetPosition, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetPosition: TargetPosition

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupMoveAbsolute command which is used to Do an absolute move. Refer to the XPS Programmer's manual to get the command description.

2.2.62 GroupMoveRelative

Syntax

C# prototype

int GroupMoveRelative(string GroupName, double[] TargetDisplacement, Int32 nbItems, out string errstring)

Python prototype

[errstring] GroupMoveRelative (GroupName, TargetDisplacement, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetDisplacement: TargetDisplacement

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupMoveRelative command which is used to Do a relative move. Refer to the XPS Programmer's manual to get the command description.

2.2.63 GroupMotionDisable

Syntax

C# prototype

int GroupMotionDisable(string GroupName, out string errstring)

Python prototype

[errstring] GroupMotionDisable (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupMotionDisable command which is used to Set Motion disable on selected group. Refer to the XPS Programmer's manual to get the command description.

2.2.64 GroupMotionEnable

Syntax

C# prototype

int GroupMotionEnable(string GroupName, out string errstring)

Python prototype

[errstring] GroupMotionEnable (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupMotionEnable command which is used to Set Motion enable on selected group. Refer to the XPS Programmer's manual to get the command description.

2.2.65 GroupPositionCorrectedProfilerGet

Syntax

C# prototype

int GroupPositionCorrectedProfilerGet(string GroupName, double PositionX, double PositionY, out double CorrectedProfilerPositionX, out double CorrectedProfilerPositionY, out string errstring)

Python prototype

[CorrectedProfilerPositionX, CorrectedProfilerPositionY, errstring] GroupPositionCorrectedProfilerGet (GroupName, PositionX, PositionY)

Parameters

Input parameters

(string) GroupName: GroupName

(double) PositionX: PositionX

(double) PositionY: PositionY

Output parameters

(double) CorrectedProfilerPositionX: CorrectedProfilerPositionX

(double) CorrectedProfilerPositionY: CorrectedProfilerPositionY

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupPositionCorrectedProfilerGet command which is used to Return corrected profiler positions. Refer to the XPS Programmer's manual to get the command description.

2.2.66 GroupPositionCurrentGet

Syntax

C# prototype

int GroupPositionCurrentGet(string GroupName, out double[] CurrentEncoderPosition, Int32 nbItems, out string errstring)

Python prototype

[CurrentEncoderPosition, errstring] GroupPositionCurrentGet (GroupName, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) CurrentEncoderPosition: CurrentEncoderPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupPositionCurrentGet command which is used to Return current positions. Refer to the XPS Programmer's manual to get the command description.

2.2.67 GroupPositionSetpointGet

Syntax

C# prototype

int GroupPositionSetpointGet(string GroupName, out double[] SetPointPosition, Int32 nbItems, out string errstring)

Python prototype

[SetPointPosition, errstring] GroupPositionSetpointGet (GroupName, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) SetPointPosition: SetPointPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupPositionSetpointGet command which is used to Return setpoint positions. Refer to the XPS Programmer's manual to get the command description.

2.2.68 GroupPositionTargetGet

Syntax

C# prototype

int GroupPositionTargetGet(string GroupName, out double[] TargetPosition, Int32 nbItems, out string errstring)

Python prototype

[TargetPosition, errstring] GroupPositionTargetGet (GroupName, nbItems)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) TargetPosition: TargetPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupPositionTargetGet command which is used to Return target positions. Refer to the XPS Programmer's manual to get the command description.

2.2.69 GroupReferencingActionExecute

Syntax

C# prototype

int GroupReferencingActionExecute(string PositionerName, string ReferencingAction, string ReferencingSensor, double ReferencingParameter, out string errstring)

Python prototype

[errstring] GroupReferencingActionExecute (PositionerName, ReferencingAction, ReferencingSensor, ReferencingParameter)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ReferencingAction: ReferencingAction

(string) ReferencingSensor: ReferencingSensor

(double) ReferencingParameter: ReferencingParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupReferencingActionExecute command which is used to Execute an action in referencing mode. Refer to the XPS Programmer's manual to get the command description.

2.2.70 GroupReferencingStart

Syntax

C# prototype

int GroupReferencingStart(string GroupName, out string errstring)

Python prototype

[errstring] GroupReferencingStart (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupReferencingStart command which is used to Enter referencing mode. Refer to the XPS Programmer's manual to get the command description.

2.2.71 GroupReferencingStop

Syntax

C# prototype

int GroupReferencingStop(string GroupName, out string errstring)

Python prototype

[errstring] GroupReferencingStop (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupReferencingStop command which is used to Exit referencing mode. Refer to the XPS Programmer's manual to get the command description.

2.2.72 GroupStatusGet

Syntax

C# prototype

int GroupStatusGet(string GroupName, out Int32 Status, out string errstring)

Python prototype

[Status, errstring] GroupStatusGet (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupStatusGet command which is used to Return group status. Refer to the XPS Programmer's manual to get the command description.

2.2.73 GroupStatusStringGet

Syntax

C# prototype

int GroupStatusStringGet(Int32 GroupStatusCode, out string GroupStatusString, out string errstring)

Python prototype

[GroupStatusString, errstring] GroupStatusStringGet (GroupStatusCode)

Parameters

Input parameters

(Int32) GroupStatusCode: GroupStatusCode

Output parameters

(string) GroupStatusString: GroupStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupStatusStringGet command which is used to Return the group status string corresponding to the group status code. Refer to the XPS Programmer's manual to get the command description.

2.2.74 KillAll

Syntax

C# prototype

int KillAll(out string errstring)

Python prototype

[errstring] KillAll ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous KillAll command which is used to Put all groups in 'Not initialized' state. Refer to the XPS Programmer's manual to get the command description.

2.2.75 RestartApplication

Syntax

C# prototype

int RestartApplication(out string errstring)

Python prototype

[errstring] RestartApplication ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous RestartApplication command which is used to Restart the Controller. Refer to the XPS Programmer's manual to get the command description.

2.2.76 PositionerBacklashGet

Syntax

C# prototype

int PositionerBacklashGet(string PositionerName, out double BacklashValue, out string BacklaskStatus, out string errstring)

Python prototype

[BacklashValue, BacklaskStatus, errstring] PositionerBacklashGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) BacklashValue: BacklashValue

(string) BacklaskStatus: BacklaskStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerBacklashGet command which is used to Read backlash value and status. Refer to the XPS Programmer's manual to get the command description.

2.2.77 PositionerBacklashSet

Syntax

C# prototype

int PositionerBacklashSet(string PositionerName, double BacklashValue, out string errstring)

Python prototype

[errstring] PositionerBacklashSet (PositionerName, BacklashValue)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) BacklashValue: BacklashValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerBacklashSet command which is used to Set backlash value. Refer to the XPS Programmer's manual to get the command description.

2.2.78 PositionerBacklashEnable

Syntax

C# prototype

int PositionerBacklashEnable(string PositionerName, out string errstring)

Python prototype

[errstring] PositionerBacklashEnable (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerBacklashEnable command which is used to Enable the backlash. Refer to the XPS Programmer's manual to get the command description.

2.2.79 PositionerBacklashDisable

Syntax

C# prototype

int PositionerBacklashDisable(string PositionerName, out string errstring)

Python prototype

[errstring] PositionerBacklashDisable (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerBacklashDisable command which is used to Disable the backlash. Refer to the XPS Programmer's manual to get the command description.

2.2.80 PositionerCorrectorNotchFiltersSet

Syntax

C# prototype

int PositionerCorrectorNotchFiltersSet(string PositionerName, double NotchFrequency1, double NotchBandwith1, double NotchGain1, double NotchFrequency2, double NotchBandwith2, double NotchGain2, out string errstring)

Python prototype

[errstring] PositionerCorrectorNotchFiltersSet (PositionerName, NotchFrequency1, NotchBandwith1, NotchGain1, NotchFrequency2, NotchBandwith2, NotchGain2)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwith1: NotchBandwith1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwith2: NotchBandwith2

(double) NotchGain2: NotchGain2

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorNotchFiltersSet command which is used to Update filters parameters . Refer to the XPS Programmer's manual to get the command description.

2.2.81 PositionerCorrectorNotchFiltersGet

Syntax

C# prototype

int PositionerCorrectorNotchFiltersGet(string PositionerName, out double NotchFrequency1, out double NotchBandwith1, out double NotchGain1, out double NotchFrequency2, out double NotchBandwith2, out double NotchGain2, out string errstring)

Python prototype

[NotchFrequency1, NotchBandwith1, NotchGain1, NotchFrequency2, NotchBandwith2, NotchGain2, errstring] PositionerCorrectorNotchFiltersGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwith1: NotchBandwith1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwith2: NotchBandwith2

(double) NotchGain2: NotchGain2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorNotchFiltersGet command which is used to Read filters parameters . Refer to the XPS Programmer's manual to get the command description.

2.2.82 PositionerCorrectorPIDFFAccelerationSet

Syntax

C# prototype

int PositionerCorrectorPIDFFAccelerationSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI, double GKD, double KForm, double FeedForwardGainAcceleration, out string errstring)

Python prototype

[errstring] PositionerCorrectorPIDFFAccelerationSet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainAcceleration)

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainAcceleration: FeedForwardGainAcceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDFFAccelerationSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.83 PositionerCorrectorPIDFFAccelerationGet

Syntax

C# prototype

int PositionerCorrectorPIDFFAccelerationGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double FeedForwardGainAcceleration, out string errstring)

Python prototype

[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainAcceleration, errstring] PositionerCorrectorPIDFFAccelerationGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainAcceleration: FeedForwardGainAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDFFAccelerationGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.84 PositionerCorrectorPIDFFVelocitySet

Syntax

C# prototype

int PositionerCorrectorPIDFFVelocitySet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI, double GKD, double KForm, double FeedForwardGainVelocity, out string errstring)

Python prototype

[errstring] PositionerCorrectorPIDFFVelocitySet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainVelocity)

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainVelocity: FeedForwardGainVelocity

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDFFVelocitySet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.85 PositionerCorrectorPIDFFVelocityGet

Syntax

C# prototype

int PositionerCorrectorPIDFFVelocityGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double FeedForwardGainVelocity, out string errstring)

Python prototype

[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainVelocity, errstring] PositionerCorrectorPIDFFVelocityGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainVelocity: FeedForwardGainVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDFFVelocityGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.86 PositionerCorrectorPIDDualFFVoltageSet

Syntax

C# prototype

int PositionerCorrectorPIDDualFFVoltageSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI, double GKD, double KForm, double FeedForwardGainVelocity, double FeedForwardGainAcceleration, double Friction, out string errstring)

Python prototype

[errstring] PositionerCorrectorPIDDualFFVoltageSet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainVelocity, FeedForwardGainAcceleration, Friction)

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainVelocity: FeedForwardGainVelocity

(double) FeedForwardGainAcceleration: FeedForwardGainAcceleration

(double) Friction: Friction

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDDualFFVoltageSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.87 PositionerCorrectorPIDDualFFVoltageGet

Syntax

C# prototype

int PositionerCorrectorPIDDualFFVoltageGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double FeedForwardGainVelocity, out double FeedForwardGainAcceleration, out double Friction, out string errstring)

Python prototype

[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, FeedForwardGainVelocity, FeedForwardGainAcceleration, Friction, errstring] PositionerCorrectorPIDDualFFVoltageGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) FeedForwardGainVelocity: FeedForwardGainVelocity

(double) FeedForwardGainAcceleration: FeedForwardGainAcceleration

(double) Friction: Friction

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIDDualFFVoltageGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.88 PositionerCorrectorPIPositionSet

Syntax

C# prototype

int PositionerCorrectorPIPositionSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double IntegrationTime, out string errstring)

Python prototype

[errstring] PositionerCorrectorPIPositionSet (PositionerName, ClosedLoopStatus, KP, KI, IntegrationTime)

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) IntegrationTime: IntegrationTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIPositionSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.89 PositionerCorrectorPIPositionGet

Syntax

C# prototype

int PositionerCorrectorPIPositionGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double IntegrationTime, out string errstring)

Python prototype

[ClosedLoopStatus, KP, KI, IntegrationTime, errstring] PositionerCorrectorPIPositionGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) IntegrationTime: IntegrationTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorPIPositionGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.90 PositionerCorrectorTypeGet

Syntax

C# prototype

int PositionerCorrectorTypeGet(string PositionerName, out string CorrectorType, out string errstring)

Python prototype

[CorrectorType, errstring] PositionerCorrectorTypeGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) CorrectorType: CorrectorType

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCorrectorTypeGet command which is used to Read corrector type. Refer to the XPS Programmer's manual to get the command description.

2.2.91 PositionerCurrentVelocityAccelerationFiltersSet

Syntax

C# prototype

int PositionerCurrentVelocityAccelerationFiltersSet(string PositionerName, double CurrentVelocityCutOffFrequency, double CurrentAccelerationCutOffFrequency, out string errstring)

Python prototype

[errstring] PositionerCurrentVelocityAccelerationFiltersSet (PositionerName, CurrentVelocityCutOffFrequency, CurrentAccelerationCutOffFrequency)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) CurrentVelocityCutOffFrequency: CurrentVelocityCutOffFrequency

(double) CurrentAccelerationCutOffFrequency: CurrentAccelerationCutOffFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCurrentVelocityAccelerationFiltersSet command which is used to Set current velocity and acceleration cut off frequencies. Refer to the XPS Programmer's manual to get the command description.

2.2.92 PositionerCurrentVelocityAccelerationFiltersGet

Syntax

C# prototype

int PositionerCurrentVelocityAccelerationFiltersGet(string PositionerName, out double CurrentVelocityCutOffFrequency, out double CurrentAccelerationCutOffFrequency, out string errstring)

Python prototype

[CurrentVelocityCutOffFrequency, CurrentAccelerationCutOffFrequency, errstring] PositionerCurrentVelocityAccelerationFiltersGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CurrentVelocityCutOffFrequency: CurrentVelocityCutOffFrequency

(double) CurrentAccelerationCutOffFrequency: CurrentAccelerationCutOffFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerCurrentVelocityAccelerationFiltersGet command which is used to Get current velocity and acceleration cut off frequencies. Refer to the XPS Programmer's manual to get the command description.

2.2.93 PositionerDriverStatusGet

Syntax

C# prototype

int PositionerDriverStatusGet(string PositionerName, out Int32 DriverStatus, out string errstring)

Python prototype

[DriverStatus, errstring] PositionerDriverStatusGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) DriverStatus: DriverStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerDriverStatusGet command which is used to Read positioner driver status. Refer to the XPS Programmer's manual to get the command description.

2.2.94 PositionerDriverStatusStringGet

Syntax

C# prototype

int PositionerDriverStatusStringGet(Int32 PositionerDriverStatus, out string PositionerDriverStatusString, out string errstring)

Python prototype

[PositionerDriverStatusString, errstring] PositionerDriverStatusStringGet (PositionerDriverStatus)

Parameters

Input parameters

(Int32) PositionerDriverStatus: PositionerDriverStatus

Output parameters

(string) PositionerDriverStatusString: PositionerDriverStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerDriverStatusStringGet command which is used to Return the positioner driver status string corresponding to the positioner error code. Refer to the XPS Programmer's manual to get the command description.

2.2.95 PositionerEncoderAmplitudeValuesGet

Syntax

C# prototype

int PositionerEncoderAmplitudeValuesGet(string PositionerName, out double CalibrationSinusAmplitude, out double CurrentSinusAmplitude, out double CalibrationCosinusAmplitude, out double CurrentCosinusAmplitude, out string errstring)

Python prototype

[CalibrationSinusAmplitude, CurrentSinusAmplitude, CalibrationCosinusAmplitude, CurrentCosinusAmplitude, errstring] PositionerEncoderAmplitudeValuesGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CalibrationSinusAmplitude: CalibrationSinusAmplitude

(double) CurrentSinusAmplitude: CurrentSinusAmplitude

(double) CalibrationCosinusAmplitude: CalibrationCosinusAmplitude

(double) CurrentCosinusAmplitude: CurrentCosinusAmplitude

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerEncoderAmplitudeValuesGet command which is used to Read analog interpolated encoder amplitude values. Refer to the XPS Programmer's manual to get the command description.

2.2.96 PositionerEncoderCalibrationParametersGet

Syntax

C# prototype

int PositionerEncoderCalibrationParametersGet(string PositionerName, out double SinusOffset, out double CosinusOffset, out double DifferentialGain, out double PhaseCompensation, out string errstring)

Python prototype

[SinusOffset, CosinusOffset, DifferentialGain, PhaseCompensation, errstring] PositionerEncoderCalibrationParametersGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) SinusOffset: SinusOffset

(double) CosinusOffset: CosinusOffset

(double) DifferentialGain: DifferentialGain

(double) PhaseCompensation: PhaseCompensation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerEncoderCalibrationParametersGet command which is used to Read analog interpolated encoder calibration parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.97 PositionerErrorGet

Syntax

C# prototype

int PositionerErrorGet(string PositionerName, out Int32 ErrorCode, out string errstring)

Python prototype

[ErrorCode, errstring] PositionerErrorGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) ErrorCode: ErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerErrorGet command which is used to Read and clear positioner error code. Refer to the XPS Programmer's manual to get the command description.

2.2.98 PositionerErrorRead

Syntax

C# prototype

int PositionerErrorRead(string PositionerName, out Int32 ErrorCode, out string errstring)

Python prototype

[ErrorCode, errstring] PositionerErrorRead (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) ErrorCode: ErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerErrorRead command which is used to Read only positioner error code without clear it. Refer to the XPS Programmer's manual to get the command description.

2.2.99 PositionerErrorStringGet

Syntax

C# prototype

int PositionerErrorStringGet(Int32 PositionerErrorCode, out string PositionerErrorString, out string errstring)

Python prototype

[PositionerErrorString, errstring] PositionerErrorStringGet (PositionerErrorCode)

Parameters

Input parameters

(Int32) PositionerErrorCode: PositionerErrorCode

Output parameters

(string) PositionerErrorString: PositionerErrorString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerErrorStringGet command which is used to Return the positioner status string corresponding to the positioner error code. Refer to the XPS Programmer's manual to get the command description.

2.2.100 PositionerHardwareStatusGet

Syntax

C# prototype

int PositionerHardwareStatusGet(string PositionerName, out Int32 HardwareStatus, out string errstring)

Python prototype

[HardwareStatus, errstring] PositionerHardwareStatusGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) HardwareStatus: HardwareStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerHardwareStatusGet command which is used to Read positioner hardware status. Refer to the XPS Programmer's manual to get the command description.

2.2.101 PositionerHardwareStatusStringGet

Syntax

C# prototype

int PositionerHardwareStatusStringGet(Int32 PositionerHardwareStatus, out string PositionerHardwareStatusString, out string errstring)

Python prototype

[PositionerHardwareStatusString, errstring] PositionerHardwareStatusStringGet (PositionerHardwareStatus)

Parameters

Input parameters

(Int32) PositionerHardwareStatus: PositionerHardwareStatus

Output parameters

(string) PositionerHardwareStatusString: PositionerHardwareStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerHardwareStatusStringGet command which is used to Return the positioner hardware status string corresponding to the positioner error code. Refer to the XPS Programmer's manual to get the command description.

2.2.102 PositionerHardInterpolatorFactorGet

Syntax

C# prototype

int PositionerHardInterpolatorFactorGet(string PositionerName, out Int32 InterpolationFactor, out string errstring)

Python prototype

[InterpolationFactor, errstring] PositionerHardInterpolatorFactorGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) InterpolationFactor: InterpolationFactor

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerHardInterpolatorFactorGet command which is used to Get hard interpolator parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.103 PositionerHardInterpolatorFactorSet

Syntax

C# prototype

int PositionerHardInterpolatorFactorSet(string PositionerName, Int32 InterpolationFactor, out string errstring)

Python prototype

[errstring] PositionerHardInterpolatorFactorSet (PositionerName, InterpolationFactor)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) InterpolationFactor: InterpolationFactor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerHardInterpolatorFactorSet command which is used to Set hard interpolator parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.104 PositionerMaximumVelocityAndAccelerationGet

Syntax

C# prototype

int PositionerMaximumVelocityAndAccelerationGet(string PositionerName, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)

Python prototype

[MaximumVelocity, MaximumAcceleration, errstring] PositionerMaximumVelocityAndAccelerationGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerMaximumVelocityAndAccelerationGet command which is used to Return maximum velocity and acceleration of the positioner. Refer to the XPS Programmer's manual to get the command description.

2.2.105 PositionerMotionDoneGet

Syntax

C# prototype

int PositionerMotionDoneGet(string PositionerName, out double PositionWindow, out double VelocityWindow, out double CheckingTime, out double MeanPeriod, out double TimeOut, out string errstring)

Python prototype

[PositionWindow, VelocityWindow, CheckingTime, MeanPeriod, TimeOut, errstring] PositionerMotionDoneGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PositionWindow: PositionWindow

(double) VelocityWindow: VelocityWindow

(double) CheckingTime: CheckingTime

(double) MeanPeriod: MeanPeriod

(double) TimeOut: TimeOut

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerMotionDoneGet command which is used to Read motion done parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.106 PositionerMotionDoneSet

Syntax

C# prototype

int PositionerMotionDoneSet(string PositionerName, double PositionWindow, double VelocityWindow, double CheckingTime, double MeanPeriod, double TimeOut, out string errstring)

Python prototype

[errstring] PositionerMotionDoneSet (PositionerName, PositionWindow, VelocityWindow, CheckingTime, MeanPeriod, TimeOut)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PositionWindow: PositionWindow

(double) VelocityWindow: VelocityWindow

(double) CheckingTime: CheckingTime

(double) MeanPeriod: MeanPeriod

(double) TimeOut: TimeOut

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerMotionDoneSet command which is used to Update motion done parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.107 PositionerSGammaExactVelocityAjustedDisplacementGet

Syntax

C# prototype

int PositionerSGammaExactVelocityAjustedDisplacementGet(string PositionerName, double DesiredDisplacement, out double AdjustedDisplacement, out string errstring)

Python prototype

[AdjustedDisplacement, errstring] PositionerSGammaExactVelocityAjustedDisplacementGet (PositionerName, DesiredDisplacement)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) DesiredDisplacement: DesiredDisplacement

Output parameters

(double) AdjustedDisplacement: AdjustedDisplacement

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerSGammaExactVelocityAjustedDisplacementGet command which is used to Return adjusted displacement to get exact velocity. Refer to the XPS Programmer's manual to get the command description.

2.2.108 PositionerSGammaParametersGet

Syntax

C# prototype

int PositionerSGammaParametersGet(string PositionerName, out double Velocity, out double Acceleration, out double MinimumTjerkTime, out double MaximumTjerkTime, out string errstring)

Python prototype

[Velocity, Acceleration, MinimumTjerkTime, MaximumTjerkTime, errstring] PositionerSGammaParametersGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinimumTjerkTime: MinimumTjerkTime

(double) MaximumTjerkTime: MaximumTjerkTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerSGammaParametersGet command which is used to Read dynamic parameters for one axe of a group for a future displacement . Refer to the XPS Programmer's manual to get the command description.

2.2.109 PositionerSGammaParametersSet

Syntax

C# prototype

int PositionerSGammaParametersSet(string PositionerName, double Velocity, double Acceleration, double MinimumTjerkTime, double MaximumTjerkTime, out string errstring)

Python prototype

[errstring] PositionerSGammaParametersSet (PositionerName, Velocity, Acceleration, MinimumTjerkTime, MaximumTjerkTime)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinimumTjerkTime: MinimumTjerkTime

(double) MaximumTjerkTime: MaximumTjerkTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerSGammaParametersSet command which is used to Update dynamic parameters for one axe of a group for a future displacement. Refer to the XPS Programmer's manual to get the command description.

2.2.110 PositionerSGammaParametersDistanceGet

Syntax

C# prototype

int PositionerSGammaParametersDistanceGet(string PositionerName, double Displacement, double Velocity, double Acceleration, double MinJerkTime, double MaxJerkTime, out double DisplacementDuringAcc, out double DisplacementDuringVel, out string errstring)

Python prototype

[DisplacementDuringAcc, DisplacementDuringVel, errstring] PositionerSGammaParametersDistanceGet (PositionerName, Displacement, Velocity, Acceleration, MinJerkTime, MaxJerkTime)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Displacement: Displacement

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinJerkTime: MinJerkTime

(double) MaxJerkTime: MaxJerkTime

Output parameters

(double) DisplacementDuringAcc: DisplacementDuringAcc

(double) DisplacementDuringVel: DisplacementDuringVel

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerSGammaParametersDistanceGet command which is used to Returns distance during acceleration phase and distance during constant velocity phase . Refer to the XPS Programmer's manual to get the command description.

2.2.111 PositionerSGammaPreviousMotionTimesGet

Syntax

C# prototype

int PositionerSGammaPreviousMotionTimesGet(string PositionerName, out double SettingTime, out double SettlingTime, out string errstring)

Python prototype

[SettingTime, SettlingTime, errstring] PositionerSGammaPreviousMotionTimesGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) SettingTime: SettingTime

(double) SettlingTime: SettlingTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerSGammaPreviousMotionTimesGet command which is used to Read SettingTime and SettlingTime. Refer to the XPS Programmer's manual to get the command description.

2.2.112 PositionerStageParameterGet

Syntax

C# prototype

int PositionerStageParameterGet(string PositionerName, string ParameterName, out string ParameterValue, out string errstring)

Python prototype

[ParameterValue, errstring] PositionerStageParameterGet (PositionerName, ParameterName)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ParameterName: ParameterName

Output parameters

(string) ParameterValue: ParameterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerStageParameterGet command which is used to Return the stage parameter. Refer to the XPS Programmer's manual to get the command description.

2.2.113 PositionerStageParameterSet

Syntax

C# prototype

int PositionerStageParameterSet(string PositionerName, string ParameterName, string ParameterValue, out string errstring)

Python prototype

[errstring] PositionerStageParameterSet (PositionerName, ParameterName, ParameterValue)

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ParameterName: ParameterName

(string) ParameterValue: ParameterValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerStageParameterSet command which is used to Save the stage parameter. Refer to the XPS Programmer's manual to get the command description.

2.2.114 PositionerUserTravelLimitsGet

Syntax

C# prototype

int PositionerUserTravelLimitsGet(string PositionerName, out double UserMinimumTarget, out double UserMaximumTarget, out string errstring)

Python prototype

[UserMinimumTarget, UserMaximumTarget, errstring] PositionerUserTravelLimitsGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) UserMinimumTarget: UserMinimumTarget

(double) UserMaximumTarget: UserMaximumTarget

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerUserTravelLimitsGet command which is used to Read UserMinimumTarget and UserMaximumTarget. Refer to the XPS Programmer's manual to get the command description.

2.2.115 PositionerUserTravelLimitsSet

Syntax

C# prototype

int PositionerUserTravelLimitsSet(string PositionerName, double UserMinimumTarget, double UserMaximumTarget, out string errstring)

Python prototype

[errstring] PositionerUserTravelLimitsSet (PositionerName, UserMinimumTarget, UserMaximumTarget)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) UserMinimumTarget: UserMinimumTarget

(double) UserMaximumTarget: UserMaximumTarget

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerUserTravelLimitsSet command which is used to Update UserMinimumTarget and UserMaximumTarget. Refer to the XPS Programmer's manual to get the command description.

2.2.116 HexapodMoveAbsolute

Syntax

C# prototype

int HexapodMoveAbsolute(string GroupName, string CoordinateSystem, double X, double Y, double Z, double U, double V, double W, out string errstring)

Python prototype

[errstring] HexapodMoveAbsolute (GroupName, CoordinateSystem, X, Y, Z, U, V, W)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveAbsolute command which is used to Hexapod absolute move in a specific coordinate system. Refer to the XPS Programmer's manual to get the command description.

2.2.117 HexapodMoveIncremental

Syntax

C# prototype

int HexapodMoveIncremental(string GroupName, string CoordinateSystem, double dX, double dY, double dZ, double dU, double dV, double dW, out string errstring)

Python prototype

[errstring] HexapodMoveIncremental (GroupName, CoordinateSystem, dX, dY, dZ, dU, dV, dW)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

(double) dU: dU

(double) dV: dV

(double) dW: dW

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveIncremental command which is used to Hexapod incremental move in a specific coordinate system. Refer to the XPS Programmer's manual to get the command description.

2.2.118 HexapodCoordinatesGet

Syntax

C# prototype

int HexapodCoordinatesGet(string GroupName, string CoordinateSystemIn, string CoordinateSystemOut, double Xin, double Yin, double Zin, double Uin, double Vin, double Win, out double Xout, out double Yout, out double Zout, out double Uout, out double Vout, out double Wout, out string errstring)

Python prototype

[Xout, Yout, Zout, Uout, Vout, Wout, errstring] HexapodCoordinatesGet (GroupName, CoordinateSystemIn, CoordinateSystemOut, Xin, Yin, Zin, Uin, Vin, Win)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystemIn: CoordinateSystemIn

(string) CoordinateSystemOut: CoordinateSystemOut

(double) Xin: Xin

(double) Yin: Yin

(double) Zin: Zin

(double) Uin: Uin

(double) Vin: Vin

(double) Win: Win

Output parameters

(double) Xout: Xout

(double) Yout: Yout

(double) Zout: Zout

(double) Uout: Uout

(double) Vout: Vout

(double) Wout: Wout

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodCoordinatesGet command which is used to Get coordinates in a specific coordinate system of a point specified in another coordinate system. Refer to the XPS Programmer's manual to get the command description.

2.2.119 HexapodCoordinateSystemSet

Syntax

C# prototype

int HexapodCoordinateSystemSet(string GroupName, string CoordinateSystem, double X, double Y, double Z, double U, double V, double W, out string errstring)

Python prototype

[errstring] HexapodCoordinateSystemSet (GroupName, CoordinateSystem, X, Y, Z, U, V, W)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodCoordinateSystemSet command which is used to Modify the position of a coordinate system. Refer to the XPS Programmer's manual to get the command description.

2.2.120 HexapodCoordinateSystemGet

Syntax

C# prototype

int HexapodCoordinateSystemGet(string GroupName, string CoordinateSystem, out double X, out double Y, out double Z, out double U, out double V, out double W, out string errstring)

Python prototype

[X, Y, Z, U, V, W, errstring] HexapodCoordinateSystemGet (GroupName, CoordinateSystem)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

Output parameters

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodCoordinateSystemGet command which is used to Get the position of a coordinate system. Refer to the XPS Programmer's manual to get the command description.

2.2.121 HexapodMoveIncrementalControl

Syntax

C# prototype

int HexapodMoveIncrementalControl(string GroupName, string CoordinateSystem, string HexapodTrajectoryType, double dX, double dY, double dZ, out string errstring)

Python prototype

[errstring] HexapodMoveIncrementalControl (GroupName, CoordinateSystem, HexapodTrajectoryType, dX, dY, dZ)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveIncrementalControl command which is used to Hexapod trajectory . Refer to the XPS Programmer's manual to get the command description.

2.2.122 HexapodMoveIncrementalControlWithTargetVelocity

Syntax

C# prototype

int HexapodMoveIncrementalControlWithTargetVelocity(string GroupName, string CoordinateSystem, string HexapodTrajectoryType, double dX, double dY, double dZ, double Velocity, out string errstring)

Python prototype

[errstring] HexapodMoveIncrementalControlWithTargetVelocity (GroupName, CoordinateSystem, HexapodTrajectoryType, dX, dY, dZ, Velocity)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

(double) Velocity: Velocity

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveIncrementalControlWithTargetVelocity command which is used to Hexapod trajectory . Refer to the XPS Programmer's manual to get the command description.

2.2.123 HexapodMoveIncrementalControlPulseAndGatheringSet

Syntax

C# prototype

int HexapodMoveIncrementalControlPulseAndGatheringSet(string GroupName, Int32 Divisor, out string errstring)

Python prototype

[errstring] HexapodMoveIncrementalControlPulseAndGatheringSet (GroupName, Divisor)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Divisor: Divisor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveIncrementalControlPulseAndGatheringSet command which is used to Configure gathering with pulses : gathered data are X, Y, Z, U, V, W and pulses will be generated during only constant velocity. Refer to the XPS Programmer's manual to get the command description.

2.2.124 HexapodMoveIncrementalControlLimitGet

Syntax

C# prototype

int HexapodMoveIncrementalControlLimitGet(string GroupName, string CoordinateSystem, string HexapodTrajectoryType, double dX, double dY, double dZ, out double MaximumVelocityCarriage, out double TrajectoryPercent, out string errstring)

Python prototype

[MaximumVelocityCarriage, TrajectoryPercent, errstring] HexapodMoveIncrementalControlLimitGet (GroupName, CoordinateSystem, HexapodTrajectoryType, dX, dY, dZ)

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

Output parameters

(double) MaximumVelocityCarriage: MaximumVelocityCarriage

(double) TrajectoryPercent: TrajectoryPercent

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodMoveIncrementalControlLimitGet command which is used to Returns the maximum velocity of carriage and the percent of the trajectory executable. Refer to the XPS Programmer's manual to get the command description.

2.2.125 HexapodSGammaParametersDistanceGet

Syntax

C# prototype

int HexapodSGammaParametersDistanceGet(string PositionerName, double Displacement, double Velocity, double Acceleration, double MinJerkTime, double MaxJerkTime, out double DisplacementDuringAcc, out double DisplacementDuringVel, out string errstring)

Python prototype

[DisplacementDuringAcc, DisplacementDuringVel, errstring] HexapodSGammaParametersDistanceGet (PositionerName, Displacement, Velocity, Acceleration, MinJerkTime, MaxJerkTime)

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Displacement: Displacement

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinJerkTime: MinJerkTime

(double) MaxJerkTime: MaxJerkTime

Output parameters

(double) DisplacementDuringAcc: DisplacementDuringAcc

(double) DisplacementDuringVel: DisplacementDuringVel

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodSGammaParametersDistanceGet command which is used to Returns distance during acceleration phase and distance during constant velocity phase for a virtual SGamma profiler. Refer to the XPS Programmer's manual to get the command description.

2.2.126 SingleAxisSlaveModeEnable

Syntax

C# prototype

int SingleAxisSlaveModeEnable(string GroupName, out string errstring)

Python prototype

[errstring] SingleAxisSlaveModeEnable (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous SingleAxisSlaveModeEnable command which is used to Enable the slave mode. Refer to the XPS Programmer's manual to get the command description.

2.2.127 SingleAxisSlaveModeDisable

Syntax

C# prototype

int SingleAxisSlaveModeDisable(string GroupName, out string errstring)

Python prototype

[errstring] SingleAxisSlaveModeDisable (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous SingleAxisSlaveModeDisable command which is used to Disable the slave mode. Refer to the XPS Programmer's manual to get the command description.

2.2.128 SingleAxisSlaveParametersSet

Syntax

C# prototype

int SingleAxisSlaveParametersSet(string GroupName, string PositionerName, double Ratio, out string errstring)

Python prototype

[errstring] SingleAxisSlaveParametersSet (GroupName, PositionerName, Ratio)

Parameters

Input parameters

(string) GroupName: GroupName

(string) PositionerName: PositionerName

(double) Ratio: Ratio

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous SingleAxisSlaveParametersSet command which is used to Set slave parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.129 SingleAxisSlaveParametersGet

Syntax

C# prototype

int SingleAxisSlaveParametersGet(string GroupName, out string PositionerName, out double Ratio, out string errstring)

Python prototype

[PositionerName, Ratio, errstring] SingleAxisSlaveParametersGet (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) PositionerName: PositionerName

(double) Ratio: Ratio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous SingleAxisSlaveParametersGet command which is used to Get slave parameters. Refer to the XPS Programmer's manual to get the command description.

2.2.130 OptionalModuleExecute

Syntax

C# prototype

int OptionalModuleExecute(string ModuleFileName, string TaskName, out string errstring)

Python prototype

[errstring] OptionalModuleExecute (ModuleFileName, TaskName)

Parameters

Input parameters

(string) ModuleFileName: ModuleFileName

(string) TaskName: TaskName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous OptionalModuleExecute command which is used to Execute an optional module. Refer to the XPS Programmer's manual to get the command description.

2.2.131 OptionalModuleKill

Syntax

C# prototype

int OptionalModuleKill(string TaskName, out string errstring)

Python prototype

[errstring] OptionalModuleKill (TaskName)

Parameters

Input parameters

(string) TaskName: TaskName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous OptionalModuleKill command which is used to Kill an optional module. Refer to the XPS Programmer's manual to get the command description.

2.2.132 ControllerStatusGet

Syntax

C# prototype

int ControllerStatusGet(out Int32 ControllerStatus, out string errstring)

Python prototype

[ControllerStatus, errstring] ControllerStatusGet ()

Parameters

Input parameters

None

Output parameters

(Int32_i) ControllerStatus: ControllerStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ControllerStatusGet command which is used to Read controller current status. Refer to the XPS Programmer's manual to get the command description.

2.2.133 ControllerStatusStringGet

Syntax

C# prototype

int ControllerStatusStringGet(Int32 ControllerStatusCode, out string ControllerStatusString, out string errstring)

Python prototype

[ControllerStatusString, errstring] ControllerStatusStringGet (ControllerStatusCode)

Parameters

Input parameters

(Int32) ControllerStatusCode: ControllerStatusCode

Output parameters

(string) ControllerStatusString: ControllerStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ControllerStatusStringGet command which is used to Return the controller status string corresponding to the controller status code. Refer to the XPS Programmer's manual to get the command description.

2.2.134 EEPROMCIESet

Syntax

C# prototype

int EEPROMCIESet(Int32 CardNumber, string ReferenceString, out string errstring)

Python prototype

[errstring] EEPROMCIESet (CardNumber, ReferenceString)

Parameters

Input parameters

(Int32) CardNumber: CardNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EEPROMCIESet command which is used to Set CIE EEPROM reference string. Refer to the XPS Programmer's manual to get the command description.

2.2.135 EEPROMDACOffsetCIESet

Syntax

C# prototype

int EEPROMDACOffsetCIESet(Int32 PlugNumber, double DAC1Offset, double DAC2Offset, out string errstring)

Python prototype

[errstring] EEPROMDACOffsetCIESet (PlugNumber, DAC1Offset, DAC2Offset)

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

(double) DAC1Offset: DAC1Offset

(double) DAC2Offset: DAC2Offset

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EEPROMDACOffsetCIESet command which is used to Set CIE DAC offsets. Refer to the XPS Programmer's manual to get the command description.

2.2.136 EEPROMDriverSet

Syntax

C# prototype

int EEPROMDriverSet(Int32 PlugNumber, string ReferenceString, out string errstring)

Python prototype

[errstring] EEPROMDriverSet (PlugNumber, ReferenceString)

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EEPROMDriverSet command which is used to Set Driver EEPROM reference string. Refer to the XPS Programmer's manual to get the command description.

2.2.137 EEPROMINTSet

Syntax

C# prototype

int EEPROMINTSet(Int32 CardNumber, string ReferenceString, out string errstring)

Python prototype

[errstring] EEPROMINTSet (CardNumber, ReferenceString)

Parameters

Input parameters

(Int32) CardNumber: CardNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EEPROMINTSet command which is used to Set INT EEPROM reference string. Refer to the XPS Programmer's manual to get the command description.

2.2.138 CPUCoreAndBoardSupplyVoltagesGet

Syntax

C# prototype

int CPUCoreAndBoardSupplyVoltagesGet(out double VoltageCPUCore, out double SupplyVoltage1P5V, out double SupplyVoltage3P3V, out double SupplyVoltage5V, out double SupplyVoltage12V, out double SupplyVoltageM12V, out double SupplyVoltageM5V, out double SupplyVoltage5VSB, out string errstring)

Python prototype

[VoltageCPUCore, SupplyVoltage1P5V, SupplyVoltage3P3V, SupplyVoltage5V, SupplyVoltage12V, SupplyVoltageM12V, SupplyVoltageM5V, SupplyVoltage5VSB, errstring] CPUCoreAndBoardSupplyVoltagesGet ()

Parameters

Input parameters

None

Output parameters

(double) VoltageCPUCore: VoltageCPUCore

(double) SupplyVoltage1P5V: SupplyVoltage1P5V

(double) SupplyVoltage3P3V: SupplyVoltage3P3V

(double) SupplyVoltage5V: SupplyVoltage5V

(double) SupplyVoltage12V: SupplyVoltage12V

(double) SupplyVoltageM12V: SupplyVoltageM12V

(double) SupplyVoltageM5V: SupplyVoltageM5V

(double) SupplyVoltage5VSB: SupplyVoltage5VSB

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous CPUCoreAndBoardSupplyVoltagesGet command which is used to Get power informations. Refer to the XPS Programmer's manual to get the command description.

2.2.139 CPUTemperatureAndFanSpeedGet

Syntax

C# prototype

int CPUTemperatureAndFanSpeedGet(out double CPUTemperature, out double CPUFanSpeed, out string errstring)

Python prototype

[CPUTemperature, CPUFanSpeed, errstring] CPUTemperatureAndFanSpeedGet ()

Parameters

Input parameters

None

Output parameters

(double) CPUTemperature: CPUTemperature

(double) CPUFanSpeed: CPUFanSpeed

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous CPUTemperatureAndFanSpeedGet command which is used to Get CPU temperature and fan speed. Refer to the XPS Programmer's manual to get the command description.

2.2.140 RunTraceloggerProcessWithTimeSetting

Syntax

C# prototype

int RunTraceloggerProcessWithTimeSetting(Int32 TraceloggerTime, out string errstring)

Python prototype

[errstring] RunTraceloggerProcessWithTimeSetting (TraceloggerTime)

Parameters

Input parameters

(Int32) TraceloggerTime: TraceloggerTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous RunTraceloggerProcessWithTimeSetting command which is used to Run Tracelogger Process With Time Setting. Refer to the XPS Programmer's manual to get the command description.

2.2.141 RunTraceloggerProcessWithRollingBuffer

Syntax

C# prototype

int RunTraceloggerProcessWithRollingBuffer(out string errstring)

Python prototype

[errstring] RunTraceloggerProcessWithRollingBuffer ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous RunTraceloggerProcessWithRollingBuffer command which is used to Run Tracelogger Process With Rolling Buffer. Refer to the XPS Programmer's manual to get the command description.

2.2.142 CreateQNXEvent

Syntax

C# prototype

int CreateQNXEvent(Int32 EventNumber, string EventName, out string errstring)

Python prototype

[errstring] CreateQNXEvent (EventNumber, EventName)

Parameters

Input parameters

(Int32) EventNumber: EventNumber

(string) EventName: EventName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous CreateQNXEvent command which is used to Run Create Event. Refer to the XPS Programmer's manual to get the command description.

2.2.143 StartEventsAcqusition

Syntax

C# prototype

int StartEventsAcqusition(out string errstring)

Python prototype

[errstring] StartEventsAcqusition ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous StartEventsAcqusition command which is used to Start Events Acqusition. Refer to the XPS Programmer's manual to get the command description.

2.2.144 StopEventsAcqusition

Syntax

C# prototype

int StopEventsAcqusition(out string errstring)

Python prototype

[errstring] StopEventsAcqusition ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous StopEventsAcqusition command which is used to Stop Events Acqusition. Refer to the XPS Programmer's manual to get the command description.

2.2.145 EventTriggerSet

Syntax

C# prototype

int EventTriggerSet(out string errstring)

Python prototype

[errstring] EventTriggerSet ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventTriggerSet command which is used to Generate internal event. Refer to the XPS Programmer's manual to get the command description.

2.2.146 ActionListGet

Syntax

C# prototype

int ActionListGet(out string ActionList, out string errstring)

Python prototype

[ActionList, errstring] ActionListGet ()

Parameters

Input parameters

None

Output parameters

(string) ActionList: ActionList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ActionListGet command which is used to Action list. Refer to the XPS Programmer's manual to get the command description.

2.2.147 ActionExtendedListGet

Syntax

C# prototype

int ActionExtendedListGet(out string ActionList, out string errstring)

Python prototype

[ActionList, errstring] ActionExtendedListGet ()

Parameters

Input parameters

None

Output parameters

(string) ActionList: ActionList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ActionExtendedListGet command which is used to Action extended list. Refer to the XPS Programmer's manual to get the command description.

2.2.148 APIExtendedListGet

Syntax

C# prototype

int APIExtendedListGet(out string Method, out string errstring)

Python prototype

[Method, errstring] APIExtendedListGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous APIExtendedListGet command which is used to API method list. Refer to the XPS Programmer's manual to get the command description.

2.2.149 APIListGet

Syntax

C# prototype

int APIListGet(out string Method, out string errstring)

Python prototype

[Method, errstring] APIListGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous APIListGet command which is used to API method list without extended API. Refer to the XPS Programmer's manual to get the command description.

2.2.150 ErrorListGet

Syntax

C# prototype

int ErrorListGet(out string ErrorsList, out string errstring)

Python prototype

[ErrorsList, errstring] ErrorListGet ()

Parameters

Input parameters

None

Output parameters

(string) ErrorsList: ErrorsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ErrorListGet command which is used to Error list. Refer to the XPS Programmer's manual to get the command description.

2.2.151 EventListGet

Syntax

C# prototype

int EventListGet(out string EventList, out string errstring)

Python prototype

[EventList, errstring] EventListGet ()

Parameters

Input parameters

None

Output parameters

(string) EventList: EventList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous EventListGet command which is used to General event list. Refer to the XPS Programmer's manual to get the command description.

2.2.152 GatheringListGet

Syntax

C# prototype

int GatheringListGet(out string list, out string errstring)

Python prototype

[list, errstring] GatheringListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringListGet command which is used to Gathering type list. Refer to the XPS Programmer's manual to get the command description.

2.2.153 GatheringExtendedListGet

Syntax

C# prototype

int GatheringExtendedListGet(out string list, out string errstring)

Python prototype

[list, errstring] GatheringExtendedListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExtendedListGet command which is used to Gathering type extended list. Refer to the XPS Programmer's manual to get the command description.

2.2.154 GatheringExternalListGet

Syntax

C# prototype

int GatheringExternalListGet(out string list, out string errstring)

Python prototype

[list, errstring] GatheringExternalListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringExternalListGet command which is used to External Gathering type list. Refer to the XPS Programmer's manual to get the command description.

2.2.155 GroupStatusListGet

Syntax

C# prototype

int GroupStatusListGet(out string GroupStatusList, out string errstring)

Python prototype

[GroupStatusList, errstring] GroupStatusListGet ()

Parameters

Input parameters

None

Output parameters

(string) GroupStatusList: GroupStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GroupStatusListGet command which is used to Group status list. Refer to the XPS Programmer's manual to get the command description.

2.2.156 HardwareInternalListGet

Syntax

C# prototype

int HardwareInternalListGet(out string InternalHardwareList, out string errstring)

Python prototype

[InternalHardwareList, errstring] HardwareInternalListGet ()

Parameters

Input parameters

None

Output parameters

(string) InternalHardwareList: InternalHardwareList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HardwareInternalListGet command which is used to Internal hardware list. Refer to the XPS Programmer's manual to get the command description.

2.2.157 HardwareDriverAndStageGet

Syntax

C# prototype

int HardwareDriverAndStageGet(Int32 PlugNumber, out string DriverName, out string StageName, out string errstring)

Python prototype

[DriverName, StageName, errstring] HardwareDriverAndStageGet (PlugNumber)

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

Output parameters

(string) DriverName: DriverName

(string) StageName: StageName

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HardwareDriverAndStageGet command which is used to Smart hardware. Refer to the XPS Programmer's manual to get the command description.

2.2.158 HexapodTrajectoryListGet

Syntax

C# prototype

int HexapodTrajectoryListGet(out string list, out string errstring)

Python prototype

[list, errstring] HexapodTrajectoryListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous HexapodTrajectoryListGet command which is used to Hexapod trajectory type list. Refer to the XPS Programmer's manual to get the command description.

2.2.159 ObjectsListGet

Syntax

C# prototype

int ObjectsListGet(out string ObjectsList, out string errstring)

Python prototype

[ObjectsList, errstring] ObjectsListGet ()

Parameters

Input parameters

None

Output parameters

(string) ObjectsList: ObjectsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ObjectsListGet command which is used to Group name and positioner name. Refer to the XPS Programmer's manual to get the command description.

2.2.160 PositionerErrorListGet

Syntax

C# prototype

int PositionerErrorListGet(out string PositionerErrorList, out string errstring)

Python prototype

[PositionerErrorList, errstring] PositionerErrorListGet ()

Parameters

Input parameters

None

Output parameters

(string) PositionerErrorList: PositionerErrorList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerErrorListGet command which is used to Positioner error list. Refer to the XPS Programmer's manual to get the command description.

2.2.161 PositionerHardwareStatusListGet

Syntax

C# prototype

int PositionerHardwareStatusListGet(out string PositionerHardwareStatusList, out string errstring)

Python prototype

[PositionerHardwareStatusList, errstring] PositionerHardwareStatusListGet ()

Parameters

Input parameters

None

Output parameters

(string) PositionerHardwareStatusList: PositionerHardwareStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerHardwareStatusListGet command which is used to Positioner hardware status list. Refer to the XPS Programmer's manual to get the command description.

2.2.162 PositionerDriverStatusListGet

Syntax

C# prototype

int PositionerDriverStatusListGet(out string PositionerDriverStatusList, out string errstring)

Python prototype

[PositionerDriverStatusList, errstring] PositionerDriverStatusListGet ()

Parameters

Input parameters

None

Output parameters

(string) PositionerDriverStatusList: PositionerDriverStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PositionerDriverStatusListGet command which is used to Positioner driver status list. Refer to the XPS Programmer's manual to get the command description.

2.2.163 ReferencingActionListGet

Syntax

C# prototype

int ReferencingActionListGet(out string list, out string errstring)

Python prototype

[list, errstring] ReferencingActionListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ReferencingActionListGet command which is used to Get referencing action list. Refer to the XPS Programmer's manual to get the command description.

2.2.164 ReferencingSensorListGet

Syntax

C# prototype

int ReferencingSensorListGet(out string list, out string errstring)

Python prototype

[list, errstring] ReferencingSensorListGet ()

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ReferencingSensorListGet command which is used to Get referencing sensor list. Refer to the XPS Programmer's manual to get the command description.

2.2.165 GatheringUserDatasGet

Syntax

C# prototype

int GatheringUserDatasGet(out double UserData1, out double UserData2, out double UserData3, out double UserData4, out double UserData5, out double UserData6, out double UserData7, out double UserData8, out string errstring)

Python prototype

[UserData1, UserData2, UserData3, UserData4, UserData5, UserData6, UserData7, UserData8, errstring] GatheringUserDatasGet ()

Parameters

Input parameters

None

Output parameters

(double) UserData1: UserData1

(double) UserData2: UserData2

(double) UserData3: UserData3

(double) UserData4: UserData4

(double) UserData5: UserData5

(double) UserData6: UserData6

(double) UserData7: UserData7

(double) UserData8: UserData8

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous GatheringUserDatasGet command which is used to Return UserDatas values. Refer to the XPS Programmer's manual to get the command description.

2.2.166 ControllerMotionKernelPeriodMinMaxGet

Syntax

C# prototype

int ControllerMotionKernelPeriodMinMaxGet(out double MinimumCorrectorPeriod, out double MaximumCorrectorPeriod, out double MinimumProfilerPeriod, out double MaximumProfilerPeriod, out double MinimumServitudesPeriod, out double MaximumServitudesPeriod, out string errstring)

Python prototype

[MinimumCorrectorPeriod, MaximumCorrectorPeriod, MinimumProfilerPeriod, MaximumProfilerPeriod, MinimumServitudesPeriod, MaximumServitudesPeriod, errstring] ControllerMotionKernelPeriodMinMaxGet ()

Parameters

Input parameters

None

Output parameters

(double) MinimumCorrectorPeriod: MinimumCorrectorPeriod

(double) MaximumCorrectorPeriod: MaximumCorrectorPeriod

(double) MinimumProfilerPeriod: MinimumProfilerPeriod

(double) MaximumProfilerPeriod: MaximumProfilerPeriod

(double) MinimumServitudesPeriod: MinimumServitudesPeriod

(double) MaximumServitudesPeriod: MaximumServitudesPeriod

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ControllerMotionKernelPeriodMinMaxGet command which is used to Get controller motion kernel min/max periods. Refer to the XPS Programmer's manual to get the command description.

2.2.167 ControllerMotionKernelPeriodMinMaxReset

Syntax

C# prototype

int ControllerMotionKernelPeriodMinMaxReset(out string errstring)

Python prototype

[errstring] ControllerMotionKernelPeriodMinMaxReset ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous ControllerMotionKernelPeriodMinMaxReset command which is used to Reset controller motion kernel min/max periods. Refer to the XPS Programmer's manual to get the command description.

2.2.168 TestTCP

Syntax

C# prototype

int TestTCP(string InputString, out string ReturnString, out string errstring)

Python prototype

[ReturnString, errstring] TestTCP (InputString)

Parameters

Input parameters

(string) InputString: InputString

Output parameters

(string) ReturnString: ReturnString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous TestTCP command which is used to Test TCP/IP transfert. Refer to the XPS Programmer's manual to get the command description.

2.2.169 PrepareForUpdate

Syntax

C# prototype

int PrepareForUpdate(out string errstring)

Python prototype

[errstring] PrepareForUpdate ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchrounous PrepareForUpdate command which is used to Kill QNX processes for firmware update. Refer to the XPS Programmer's manual to get the command description.

3.0 Examples

3.1 Python Script

```
#==========================================================================
=
# Newport Proprietary and Confidential         Newport Corporation 2011
#
# No part of this file in any format, with or without modification
# shall be used, copied or distributed without the express written
# consent of Newport Corporation.
#
# Description: This is a sample Python Script to illustrate how to
execute
# "query" commands.
#==========================================================================

#==========================================================================
#Initialization Start The script within Initialization Start and
#Initialization End is needed for #properly
#Command Interface DLL for HXP #instrument. The user should copy this
code as
#is and specify correct paths here.
import sys
import time

#Command Interface DLL can be found here.
sys.path.append(r'C:\Program Files (x86)\Newport\MotionControl\HXP\Bin')

# The CLR module provide functions for interacting with the underlying
# .NET runtime
import clr

# Add reference to assembly and import names from namespace (IronPython)
clr.AddReferenceToFile("Newport.HXP.CommandInterface.dll")
from CommandInterface HXP import *

import System
#==========================================================================

# Instrument Initialization
# The key should have double slashes since
# (one of them is escape character)
address="192.168.33.3"
port=5001

# Create an HXP instance
hxp = HXP()

print '--------------- General data ---------------'
# Connect and open a socket
result = hxp. OpenInstrument(address, port, 1000)
if result == 0 :
 print 'Connection ', address, ":", port, " => Successful"
else:
 print 'Connection ', address, ":", port, " => failure ", result

# Get controller revision information
result, response, errString = hxp.FirmwareVersionGet()
if result == 0 :
 print 'Controller version: ', response
else:
 print 'Error=>',errString
```

```
# Get object list
print '---------------- Object list ----------------'
result, response, errString = hxp.ObjectsListGet()
objectList = response.split(';')
nbObjects = len(objectList)
if result == 0 :
  for i in range(nbObjects):
         print i+1, ') ', objectList[i]
else:
  print 'Error=>', result, " : ", errString

# Get GPIO analog gain
print '------------ GPIO analog gain ---------------'
from System import Array
strGpioArray = Array[str](['GPIO2.ADC1', 'GPIO2.ADC2', 'GPIO2.ADC3'])
nbItems = len(strGpioArray)
result, responseArray, errString = hxp.GPIOAnalogGainGet(strGpioArray)
if result == 0 :
  for i in range(nbItems):
         print i+1, ') Analog Gain ', strGpioArray[i], " = ",
responseArray[i]
else:
  print 'Error=>', result, " : ", errString

# Get Controller Status
print '------------ Controller Status --------------'
result, response, errString = hxp.ControllerStatusGet()
if result == 0 :
  print 'Controller Status=>', response
else:
  print 'Error=> ', result, " :  HXP error ", errString

print '--------------- End of script ---------------'

# unregister TCP IP server
hxp. CloseInstrument();
```

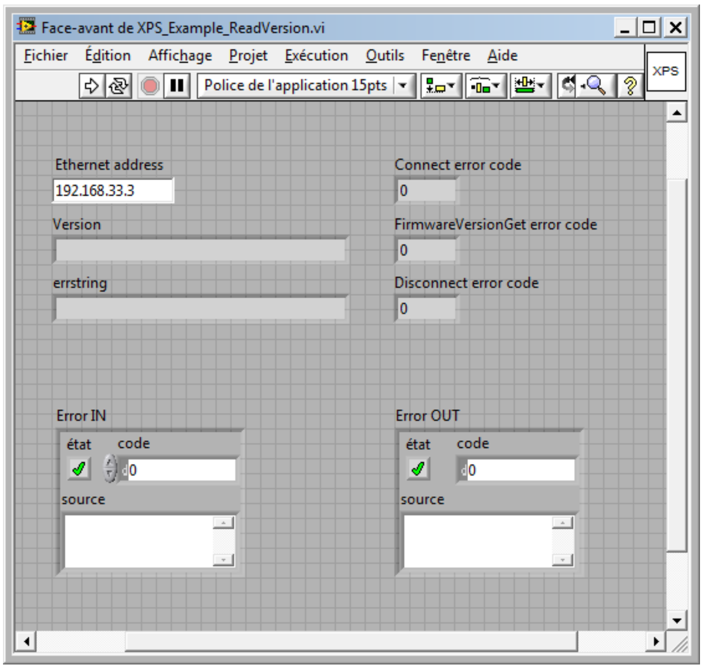
3.2 C#

```
       #region Constants
        const int DEFAULT_TO = 1000;  // timeout in milliseconds
        const int DEFAULT_PORT = 5001;
        const string DEFAULT_ADDRESS = "192.168.33.3";
        #endregion Constants

       #region Members
        /// <summary>
        /// Object of HXP Command Interface
        /// </summary>
         HXP m_hxpInterface;
        /// <summary>
        /// Controller's version
        /// </summary>
        string m_controllerVersion = string.Empty;
        #endregion Members

         /// <summary>
         /// This function allows checking the  HXP communication.
         /// Open the communication, get the controller's version and
close
         /// the communication
         /// </summary>
         /// <param name="deviceKey">The device key.</param>
         /// <param name="caption">The window caption.</param>
         public bool CheckCommunication (string deviceKey, int port)
         {
             bool startSuccessfull = false;
             int returnValue;
             string err = string.Empty;
             m_controllerVersion = string.Empty;

             try
             {
                 // Create  HXP instance
                 m_hxpInterface = new  HXP();
                 if (m_hxpInterface != null)
                 {
                     //  HXP Connection
                     returnValue = m_hxpInterface.
OpenInstrument(IPAddress, IPport, DEFAULT_TO)
                     if (returnValue == 0)
                     {
                         string str = string.Empty;

                         // Get  HXP controller's version
                         int res = m_hxpInterface.FirmwareVersionGet(out
str, out err);
                         if (res == CommandInterface. HXP.FAILURE)
                         {
                             labelControllerVersion.Text = "Failure: " +
err;
                             startSuccessfull = false;
                         }
                         else
                         {
                             m_controllerVersion = str.Trim();
                             labelControllerVersion.Text =
m_controllerVersion;
                             startSuccessfull = true;
                         }
                     }

                     //  HXP disconnection
                     m_hxpInterface.CloseInstrument();
                 }
```

```
                    }
                    catch (Exception e)
                    {
                        startSuccessfull = false;
                        MessageBox.Show("CheckCommunication error: " +
e.Message);
                    }
                    return startSuccessfull;
                }
```

3.3 LabVIEW

The .NET component named “Newport. HXP.CommandInterface” can be used with LabVIEW.



Face-avant de XPS_Example_ReadVersion.vi
Fichier Édition Affichage Projet Exécution Outils Fenêtre Aide
Police de l'application 15pts
XPS

Ethernet address: 192.168.33.3

Version:

errstring:

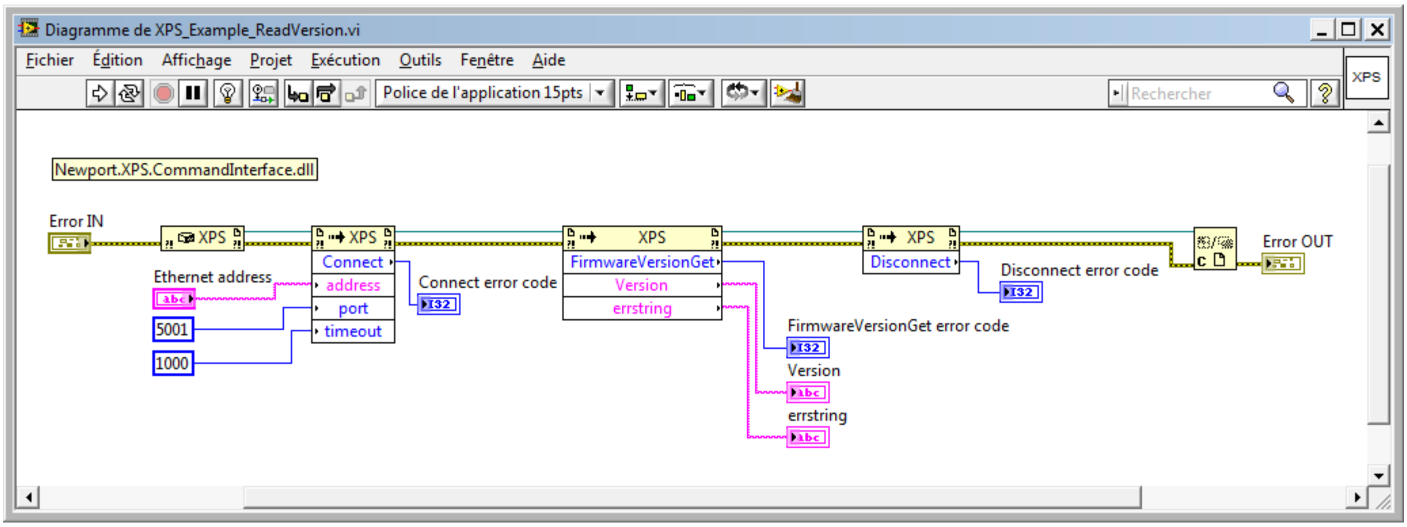
Connect error code: 0

FirmwareVersionGet error code: 0

Disconnect error code: 0

Error IN — état: ✓ code: 0 source:

Error OUT — état: ✓ code: 0 source:



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): ________________________

Newport®

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

mks

Newport® Ophir® Spectra-Physics®