



DL Controller Series

Single-Axis Motion Controller for Delay Line Stages



 **Newport®**

**Command Interface
Manual**

V1.0.x

©2018 by Newport Corporation, Irvine, CA. All rights reserved.

Original instructions.

No part of this document may be reproduced or copied without the prior written approval of Newport Corporation. This document is provided for information only, and product specifications are subject to change without notice. Any change will be reflected in future publishings.

Table of Contents

| | | |
|------------|--|-----------|
| 1.0 | Introduction | 1 |
| 1.1 | Purpose..... | 1 |
| 1.2 | Requirements | 1 |
| 1.3 | Overview..... | 2 |
| 2.0 | DLS.Net Installation..... | 3 |
| 2.1 | 32 bit (x86) Windows Platform..... | 3 |
| 2.2 | 64 bit (x64) Windows Platform..... | 3 |
| 2.3 | Communication Port Verification..... | 4 |
| 3.0 | Use DLS .NET Assembly from Visual Studio C# | 5 |
| 3.1 | Add Reference to Newport DLS .NET Assembly # | 5 |
| 3.2 | C# Code Sources..... | 5 |
| 4.0 | Use DLS .NET Assembly from LabVIEW | 7 |
| 4.1 | Add Reference to Newport DLS .NET Assembly # | 7 |
| 4.2 | LabVIEW Code Sources..... | 7 |
| 5.0 | Use DLS .NET Assembly from IronPython | 8 |
| 5.1 | Add Reference to Newport DLS .NET Assembly # | 8 |
| 5.2 | IronPython Code Sources..... | 8 |
| 6.0 | Use DLS .NET Assembly from Matlab..... | 10 |
| 6.1 | Add Reference to Newport DLS .NET Assembly # | 10 |
| 6.2 | Matlab Code Sources | 10 |
| 7.0 | Error Code List..... | 11 |
| 8.0 | Command Interface..... | 12 |
| 8.1 | Constructor..... | 12 |
| 8.2 | Functions..... | 12 |
| 8.2.1 | General..... | 12 |
| 8.2.1.1 | OpenInstrument..... | 12 |
| 8.2.1.2 | CloseInstrument | 12 |
| 8.2.2 | Commands | 13 |
| 8.2.2.1 | AC_Get | 13 |
| 8.2.2.2 | AC_Set..... | 13 |
| 8.2.2.3 | AF_Get..... | 13 |

| | | |
|----------|---------------|----|
| 8.2.2.4 | AF_Set | 13 |
| 8.2.2.5 | DBL_Get..... | 14 |
| 8.2.2.6 | DBL_Set | 14 |
| 8.2.2.7 | DBH_Get | 14 |
| 8.2.2.8 | DBH_Set..... | 14 |
| 8.2.2.9 | DCA..... | 15 |
| 8.2.2.10 | DCC | 15 |
| 8.2.2.11 | DCD_Get | 15 |
| 8.2.2.12 | DCD_Set..... | 15 |
| 8.2.2.13 | DCM_Get..... | 16 |
| 8.2.2.14 | DCM_Set | 16 |
| 8.2.2.15 | DCN_Get | 16 |
| 8.2.2.16 | DCN_Set..... | 16 |
| 8.2.2.17 | DCS_Get..... | 17 |
| 8.2.2.18 | DCS_Set..... | 17 |
| 8.2.2.19 | DCT | 17 |
| 8.2.2.20 | DCV_Get | 17 |
| 8.2.2.21 | DV_Get..... | 18 |
| 8.2.2.22 | DV_Set..... | 18 |
| 8.2.2.23 | ENF_Get | 18 |
| 8.2.2.24 | ENF_Set..... | 18 |
| 8.2.2.25 | ENP_Get | 19 |
| 8.2.2.26 | ENP_Set..... | 19 |
| 8.2.2.27 | EQF_Get | 19 |
| 8.2.2.28 | EQF_Set..... | 19 |
| 8.2.2.29 | EQP_Get | 20 |
| 8.2.2.30 | EQP_Set..... | 20 |
| 8.2.2.31 | EQR_Get..... | 20 |
| 8.2.2.32 | EQR_Set | 20 |
| 8.2.2.33 | FD_Get..... | 21 |
| 8.2.2.34 | FD_Set | 21 |
| 8.2.2.35 | FE_Get | 21 |
| 8.2.2.36 | FE_Set..... | 21 |
| 8.2.2.37 | FF_Get | 22 |
| 8.2.2.38 | FF_Set..... | 22 |
| 8.2.2.39 | FL_Get | 22 |
| 8.2.2.40 | FL_Set..... | 22 |
| 8.2.2.41 | FMC_Get | 23 |
| 8.2.2.42 | FMC_Set..... | 23 |
| 8.2.2.43 | FML_Get..... | 23 |
| 8.2.2.44 | FML_Set | 23 |
| 8.2.2.45 | FMP_Get..... | 24 |
| 8.2.2.46 | FMP_Set | 24 |
| 8.2.2.47 | FMS_Get..... | 24 |

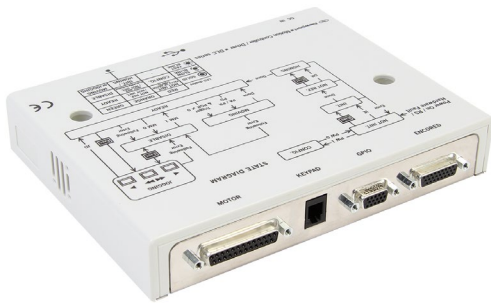
| | | |
|----------|---------------|----|
| 8.2.2.48 | FMS_Set | 24 |
| 8.2.2.49 | FSM_Get..... | 25 |
| 8.2.2.50 | FSM_Set | 25 |
| 8.2.2.51 | FSR | 25 |
| 8.2.2.52 | GIC_Get..... | 25 |
| 8.2.2.53 | GIC_Set..... | 26 |
| 8.2.2.54 | GIM_Get..... | 26 |
| 8.2.2.55 | GIM_Set..... | 26 |
| 8.2.2.56 | GIT_Get..... | 26 |
| 8.2.2.57 | GIT_Set..... | 27 |
| 8.2.2.58 | GOF_Get..... | 27 |
| 8.2.2.59 | GOF_Set | 27 |
| 8.2.2.60 | GOP_Get..... | 27 |
| 8.2.2.61 | GOP_Set | 28 |
| 8.2.2.62 | GOM_Get | 28 |
| 8.2.2.63 | GOM_Set | 28 |
| 8.2.2.64 | GO_Get..... | 28 |
| 8.2.2.65 | GOT_Set..... | 29 |
| 8.2.2.66 | GOW_Get | 29 |
| 8.2.2.67 | GOW_Set..... | 29 |
| 8.2.2.68 | GPE_Get | 29 |
| 8.2.2.69 | GPE_Set..... | 30 |
| 8.2.2.70 | GPI_Get..... | 30 |
| 8.2.2.71 | GPI_Set..... | 30 |
| 8.2.2.72 | GPL_Get | 30 |
| 8.2.2.73 | GPL_Set..... | 31 |
| 8.2.2.74 | GPS_Get | 31 |
| 8.2.2.75 | GPS_Set..... | 31 |
| 8.2.2.76 | HO_Get..... | 31 |
| 8.2.2.77 | HO_Set..... | 32 |
| 8.2.2.78 | HT_Get | 32 |
| 8.2.2.79 | HT_Set..... | 32 |
| 8.2.2.80 | ID_Get..... | 32 |
| 8.2.2.81 | ID_Set..... | 33 |
| 8.2.2.82 | IE 33 | |
| 8.2.2.83 | ITA_Get | 33 |
| 8.2.2.84 | ITA_Set..... | 33 |
| 8.2.2.85 | ITD_Get | 34 |
| 8.2.2.86 | ITD_Set..... | 34 |
| 8.2.2.87 | JA_Get | 34 |
| 8.2.2.88 | JA_Set..... | 34 |
| 8.2.2.89 | JD35 | |
| 8.2.2.90 | JM_Get..... | 35 |
| 8.2.2.91 | JM_Set | 35 |

| | | |
|-----------|--------------|----|
| 8.2.2.92 | JR_Get..... | 35 |
| 8.2.2.93 | JR_Set..... | 36 |
| 8.2.2.94 | JV_Get..... | 36 |
| 8.2.2.95 | JV_Set..... | 36 |
| 8.2.2.96 | KD_Get..... | 36 |
| 8.2.2.97 | KD_Set..... | 37 |
| 8.2.2.98 | KGD_Get..... | 37 |
| 8.2.2.99 | KGD_Set..... | 37 |
| 8.2.2.100 | KGF_Get..... | 37 |
| 8.2.2.101 | KGF_Set..... | 38 |
| 8.2.2.102 | KGI_Get..... | 38 |
| 8.2.2.103 | KGI_Set..... | 38 |
| 8.2.2.104 | KGP_Get..... | 38 |
| 8.2.2.105 | KGP_Set..... | 39 |
| 8.2.2.106 | KI_Get..... | 39 |
| 8.2.2.107 | KI_Set..... | 39 |
| 8.2.2.108 | KP_Get..... | 39 |
| 8.2.2.109 | KP_Set..... | 40 |
| 8.2.2.110 | KS_Get..... | 40 |
| 8.2.2.111 | KS_Set..... | 40 |
| 8.2.2.112 | LT_Get..... | 40 |
| 8.2.2.113 | LT_Set..... | 41 |
| 8.2.2.114 | MDA_Get..... | 41 |
| 8.2.2.115 | MDA_Set..... | 41 |
| 8.2.2.116 | MDC_Get..... | 41 |
| 8.2.2.117 | MDC_Set..... | 42 |
| 8.2.2.118 | MDM_Get..... | 42 |
| 8.2.2.119 | MDM_Set..... | 42 |
| 8.2.2.120 | MDP_Get..... | 42 |
| 8.2.2.121 | MDP_Set..... | 43 |
| 8.2.2.122 | MDT_Get..... | 43 |
| 8.2.2.123 | MDT_Set..... | 43 |
| 8.2.2.124 | MDV_Get..... | 43 |
| 8.2.2.125 | MDV_Set..... | 44 |
| 8.2.2.126 | MM_Get..... | 44 |
| 8.2.2.127 | MM_Set..... | 44 |
| 8.2.2.128 | MP_Get..... | 44 |
| 8.2.2.129 | MP_Set..... | 45 |
| 8.2.2.130 | MT_Get..... | 45 |
| 8.2.2.131 | MT_Set..... | 45 |
| 8.2.2.132 | NFF_Get..... | 45 |
| 8.2.2.133 | NFF_Set..... | 46 |
| 8.2.2.134 | NFG_Get..... | 46 |
| 8.2.2.135 | NFG_Set..... | 46 |

| | | |
|-----------|--------------|----|
| 8.2.2.136 | NFW_Get..... | 47 |
| 8.2.2.137 | NFW_Set..... | 47 |
| 8.2.2.138 | OH_Get..... | 47 |
| 8.2.2.139 | OH_Set..... | 47 |
| 8.2.2.140 | OR..... | 48 |
| 8.2.2.141 | OT_Get..... | 48 |
| 8.2.2.142 | OT_Set..... | 48 |
| 8.2.2.143 | PA_Get..... | 48 |
| 8.2.2.144 | PA_Set..... | 49 |
| 8.2.2.145 | PD..... | 49 |
| 8.2.2.146 | PG_Get..... | 49 |
| 8.2.2.147 | PG_Set..... | 49 |
| 8.2.2.148 | PI_Get..... | 50 |
| 8.2.2.149 | PI_Set..... | 50 |
| 8.2.2.150 | PR_Get..... | 50 |
| 8.2.2.151 | PR_Set..... | 50 |
| 8.2.2.152 | PTA..... | 51 |
| 8.2.2.153 | PTT..... | 51 |
| 8.2.2.154 | PW_Get..... | 51 |
| 8.2.2.155 | PW_Set..... | 51 |
| 8.2.2.156 | QCF_Get..... | 52 |
| 8.2.2.157 | QCF_Set..... | 52 |
| 8.2.2.158 | QCL_Get..... | 52 |
| 8.2.2.159 | QCL_Set..... | 52 |
| 8.2.2.160 | QCR_Get..... | 53 |
| 8.2.2.161 | QCR_Set..... | 53 |
| 8.2.2.162 | QIL_Get..... | 53 |
| 8.2.2.163 | QIL_Set..... | 53 |
| 8.2.2.164 | QIR_Get..... | 54 |
| 8.2.2.165 | QIL_Set..... | 54 |
| 8.2.2.166 | QIT_Get..... | 54 |
| 8.2.2.167 | QIT_Set..... | 54 |
| 8.2.2.168 | RAA..... | 55 |
| 8.2.2.169 | RAB..... | 55 |
| 8.2.2.170 | RF_Get..... | 55 |
| 8.2.2.171 | RF_Set..... | 55 |
| 8.2.2.172 | RS..... | 56 |
| 8.2.2.173 | SC_Get..... | 56 |
| 8.2.2.174 | SC_Set..... | 56 |
| 8.2.2.175 | SL_Get..... | 56 |
| 8.2.2.176 | SL_Set..... | 57 |
| 8.2.2.177 | SN_Get..... | 57 |
| 8.2.2.178 | SN_Set..... | 57 |
| 8.2.2.179 | SR_Get..... | 57 |

| | | |
|-----------|-------------|----|
| 8.2.2.180 | SR_Set..... | 58 |
| 8.2.2.181 | ST..... | 58 |
| 8.2.2.182 | TB..... | 58 |
| 8.2.2.183 | TE..... | 58 |
| 8.2.2.184 | TH..... | 59 |
| 8.2.2.185 | TP..... | 59 |
| 8.2.2.186 | TS..... | 59 |
| 8.2.2.187 | VA_Get..... | 59 |
| 8.2.2.188 | VA_Set..... | 60 |
| 8.2.2.189 | VAM..... | 60 |
| 8.2.2.190 | VE..... | 60 |
| 8.2.2.191 | ZT..... | 60 |

| | |
|---------------------------|-----------|
| Service Form | 61 |
|---------------------------|-----------|



Single-Axis Motion Controller for Delay Line Stages DL Controller

1.0 Introduction

1.1 Purpose

The purpose of this document is first to explain how to install the “DLS.Net” assembly designed and developed by Newport and then provide the method syntax of each command to communicate with the DL Controller device through Visual Studio C#, Labview, Iron Python and Matlab.

All Necessary files can be obtained from the provided CD or from <ftp://download.newport.com/MotionControl/Current/MotionControllers/DL>

1.2 Requirements

.Net Framework is a programming infrastructure created by Microsoft for building, deploying, and running applications and services that use .NET technologies such as custom desktop applications.

The Windows PC computer requires having at least the .NET Framework installed and you need to install either 32 bit (x86) or 64 bit (x64) .NET assembly depending on the Windows version you are using.

When developing your application, refer to the programming environment documentation to make the installed .NET assembly visible.

1.3 Overview

To communicate with the DL controller you will need to:

- Use the **OpenInstrument** method to connect to the controller.
- Communicate with the controller using any of its **commands**.

These commands are exposed as function calls of the Command Interface DLL.

NOTE

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

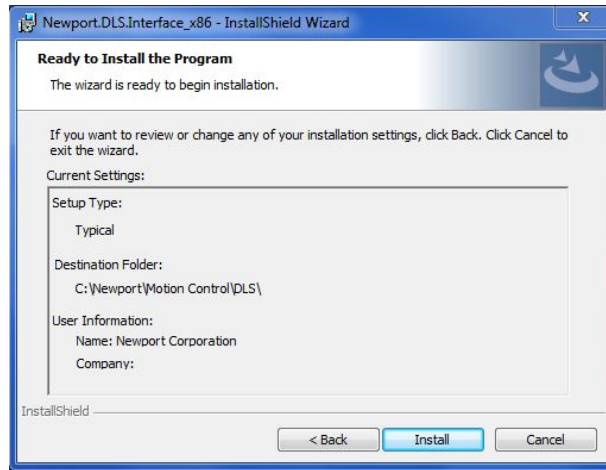
For each command function, refer to the DL Controller programmer’s manual.

-
- Once your application terminates it needs to disconnect from the controller using the **CloseInstrument** method. If it doesn’t close the communication channel and runs many connections to the controller, it can run out of free channels and gets an error.

2.0 DLS.Net Installation

2.1 32 bit (x86) Windows Platform

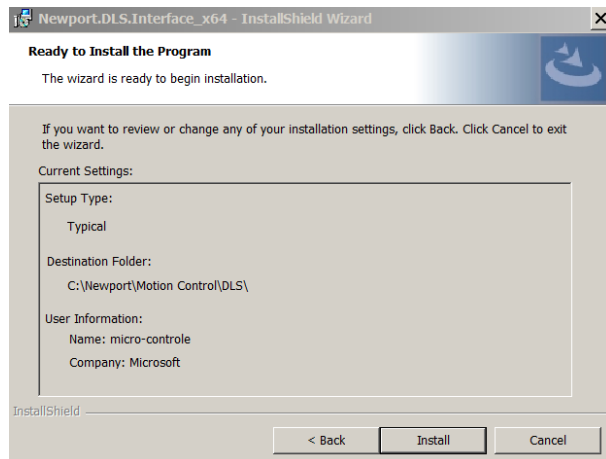
Run the “Newport.DLS.CommandInterface_x86.exe” from your computer.



The .NET assembly “Newport.DLS.CommandInterface.dll” is installed in the GAC for x86 platforms in “C:\Windows\Microsoft.NET\assembly\GAC_32\Newport.DLS.CommandInterface\” folder and is ready for use.

2.2 64 bit (x64) Windows Platform

Run the “Newport.DLS.CommandInterface_x64.exe” from your computer.



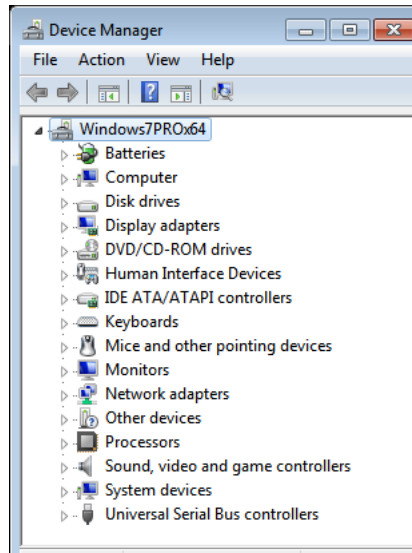
The .NET assembly “Newport.DLS.CommandInterface.dll” is installed in the GAC for x64 platforms in “C:\Windows\Microsoft.NET\assembly\GAC_64\Newport.DLS.CommandInterface\” folder and is ready for use.

2.3 Communication Port Verification

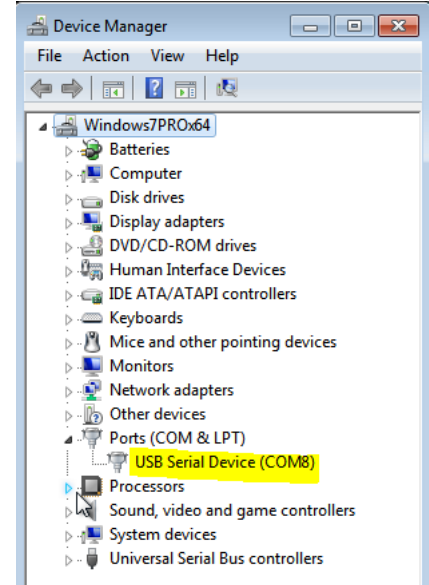
When the DL controller is connected to the computer, Windows assigns a serial (COM) port for communication. To verify which port has been assigned, open Windows “Device Manager” from the Control Panel and look for “USB serial device” in the “Ports (COM & LPT)” chapter.

In the example below (windows 10), COM8 has been reserved for Newport DL Controller.

Before connecting DL controller to the PC.



After connecting DL controller to the PC.

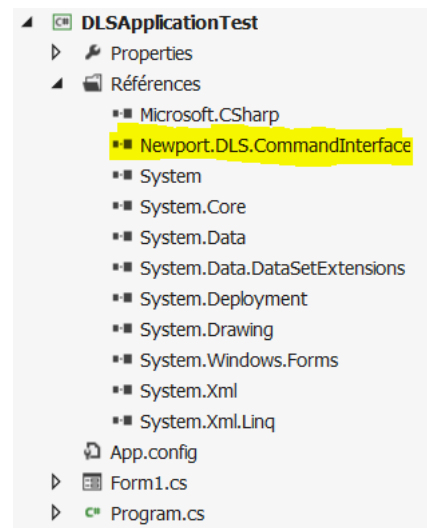


3.0 Use DLS .NET Assembly from Visual Studio C#

Refer to Microsoft for more information on how to load and use a .NET assembly depending on your Visual Studio version.

3.1 Add Reference to Newport DLS .NET Assembly

In your project add Newport.DLS.CommandInterface.dll in References from Windows GAC:



3.2 C# Code Sources

C# Header

```
using CommandInterfaceDLS; // Newport.DLS.CommandInterface .NET Assembly
access
```

Add a Variable to Declare a “DLS” Object

```
CommandInterfaceDLS.DLS m_dlsInterface = null;
```

Create an Instance of “DLS” Object

```
m_dlsInterface = new CommandInterfaceDLS.DLS();
if (m_dlsInterface != null)
    ...
```

Open DLS Connection

```
if (m_dlsInterface != null)
    int returnValue = m_dlsInterface.OpenInstrument(COMPort);
```

Call “DLS” Functions

```
if (m_dlsInterface != null)
{
string DLSVersion = string.Empty;
string errorString = string.Empty;
int queryResult = m_dlsInterface.VE(out DLSVersion, out errorString);
if (!IsQuerySuccess(queryResult))
.
.
```

Close DLS Connection

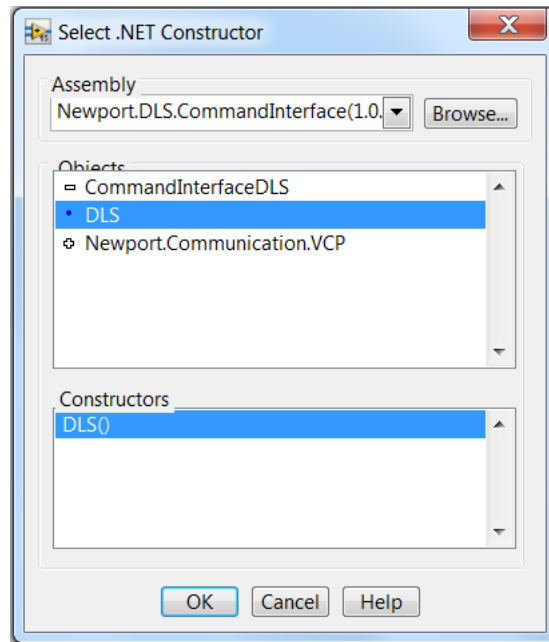
```
if (m_dlsInterface != null)
m_dlsInterface.CloseInstrument();
```

4.0 Use DLS .NET Assembly from LabVIEW

Refer to [LabVIEW](#) for more information on how to load and use a .NET assembly depending on your LabVIEW version.

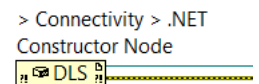
4.1 Add Reference to Newport DLS .NET Assembly

Select **CommandInterfaceDLS** and **DLS** constructor from a **.Net Constructor Node** (refer to Connectivity panel):

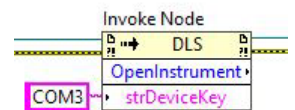


4.2 LabVIEW Code Sources

The instance of “DLS” object is created after configuration of **.Net Constructor Node**:

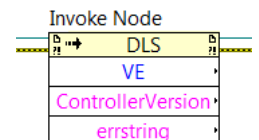


Open DLS connection (Use a **.Net Invoke Node** to select the DLS method “OpenInstrument”):

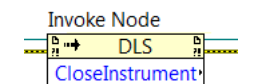


Set the assigned COM Port (i.e. COM3)

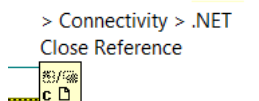
Call “DLS” functions (Use a **.Net Invoke Node** to select a DLS method):



Close DLS connection (Use a **.Net Invoke Node** to select the DLS method “CloseInstrument”):



Close .NET Reference:



5.0 Use DLS .NET Assembly from IronPython

Refer to [IronPython](#) for more information on how to load and use a .NET assembly depending on your IronPython version.

5.1 Add Reference to Newport DLS .NET Assembly

Add Newport.DLS.CommandInterface.dll in References of your script depending on Windows:

- **For x86 (32bits):**

```
import sys
```

```
sys.path.append(r'C:\Windows\Microsoft.NET\assembly\GAC_32\Newport.DLS.CommandInterface\v4.0_1.0.0.1__90ac4f829985d2bf')
```

- **For x64 (64bits):**

```
import sys
```

```
sys.path.append(r'C:\Windows\Microsoft.NET\assembly\GAC_64\Newport.DLS.CommandInterface\v4.0_1.0.0.1__90ac4f829985d2bf')
```

5.2 IronPython Code Sources

Add library path (here Windows 32bits)

```
import sys
```

```
sys.path.append(r'C:\Windows\Microsoft.NET\assembly\GAC_32\Newport.DLS.CommandInterface\v4.0_1.0.0.1__90ac4f829985d2bf')
```

Add reference to assembly and import names from namespace. The CLR module provide functions for interacting with the underlying .NET runtime

```
import clr
```

```
clr.AddReference("Newport.DLS.CommandInterface.dll")
```

```
from CommandInterfaceDLS import *
```

Open DLS Connection (here COM Port 3 Has been assigned by Windows)

```
import System
```

```
instrument="COM3"
```

```
print 'Instrument Key=>', instrument
```

Create an Instance DLS interface

```
myDLS = DLS()
```

Open a socket

```
result = myDLS.OpenInstrument(instrument)
```

```
if result == 0:
```

```
print 'Open port COM8 => Successful'
```

```
else:
```

```
print 'Open port COM8 => failure', result
```


Call DLS Functions (here get DLS firmware version: VE)

```
result = myDLS.VE()  
print 'version => ', result
```

Close DLS Connection

```
myDLS.CloseInstrument()
```

6.0 Use DLS .NET Assembly from Matlab

Refer to [Matlab](#) for more information on how to load and use a .NET assembly depending on your Matlab version.

6.1 Add Reference to Newport DLS .NET Assembly

% Make the assembly visible from Matlab

```
asmInfo = NET.addAssembly('Newport.DLS.CommandInterface');
```

6.2 Matlab Code Sources

Create an Instance

% Make the instantiation

```
mydls = CommandInterfaceDLS.DLS();
```

Open DLS Connection

% Connect to the DLS controller

% Note: in this example COM Port 3 has been assigned to DLS

```
code = mydls.OpenInstrument('COM3');
```

Call DLS Functions

% Use API's

```
[code Version] = mydls.VE;
```

Close DLS Connection

% Disconnect from the DLS controller

```
code = mydls.CloseInstrument;
```

7.0 Error Code List

The DLS Command Interface assembly returns the codes below in case an error of managing the serial communication has occurred:

- 100: The specified port on the current instance of the serial port is already open or closed.
- 101: The communication timeout has expired.
- 102: Either the port name does not begin with "COM", or the file type of the port is not supported.

8.0 Command Interface

8.1 Constructor

DLS()

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

8.2 Functions

8.2.1 General

8.2.1.1 OpenInstrument

Syntax

int OpenInstrument(string strDeviceKey)

string strDeviceKey: the device key is a serial COM port

return: 0 = successful or -1 = failure

Description

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

8.2.1.2 CloseInstrument

Syntax

int CloseInstrument()

return: 0 = successful or -1 = failure

Description

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

8.2.2 Commands

8.2.2.1 AC Get

Syntax

int AC_Get(out double Acceleration, out string errstring)

Acceleration: Acceleration

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous AC Get command which is used to Get acceleration. Refer to the Controller's manual to get the command description.

8.2.2.2 AC Set

Syntax

int AC_Set(double Acceleration, out string errstring)

Acceleration: Acceleration

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous AC Set command which is used to Set acceleration. Refer to the Controller's manual to get the command description.

8.2.2.3 AF Get

Syntax

int AF_Get(out double AccelerationFeedForward, out string errstring)

AccelerationFeedForward: AccelerationFeedForward

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous AF Get command which is used to Get acceleration feed forward. Refer to the Controller's manual to get the command description.

8.2.2.4 AF Set

Syntax

int AF_Set(double AccelerationFeedForward, out string errstring)

AccelerationFeedForward: AccelerationFeedForward

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous AF Set command which is used to Set acceleration feed forward. Refer to the Controller's manual to get the command description.

8.2.2.5 **DBL_Get**

Syntax

int DBL_Get(out double DeadbandLowLimit, out string errstring)

DeadbandLowLimit: Dead band Low Limit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to get the dead band low limit of the PID control loop.

8.2.2.6 **DBL_Set**

Syntax

int DBL_Set(double DeadbandLowLimit, out string errstring)

DeadbandLowLimit: Dead band Low Limit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to set dead band low limit of the PID control loop which can be saved in the controller's non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for deadband. This value is not saved in the controller's memory and will be lost after reboot.

To disable the deadband functionality set the low limit to 0.

8.2.2.7 **DBH_Get**

Syntax

int DBH_Get(out double DeadbandHighLimit, out string errstring)

DeadbandHighLimit: Dead band High Limit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to get the dead band high limit of the PID control loop.

8.2.2.8 **DBH_Set**

Syntax

int DBL_Set(double DeadbandHighLimit, out string errstring)

DeadbandHighLimit: Dead band High Limit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to set dead band high limit of the PID control loop which can be saved in the controller's non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

8.2.2.9 **DCA**

Syntax

int DCA_Get(int GatheringDataLineNumber, out double ReturnedLineNumber, out string errstring)

GatheringDataLineNumber: The asked gathering data line number

ReturnedLineNumber: The returned gathering data line number

errString: 0 in success and -1 on failure

return: 0 in success and -1 on failure

Description

This function is used to get the gathered data line GatheringDataLineNumber. Refer to the Controller's manual to get the command description.

8.2.2.10 **DCC**

Syntax

int DCC(out int CurrentDataNumber, out string errstring)

CurrentDataNumber: CurrentDataNumber

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCC Get command which is used to Get the current number of gathered data lines. Refer to the Controller's manual to get the command description.

8.2.2.11 **DCD Get**

Syntax

int DCD_Get(out int FrequencyDivisorValue, out string errstring)

FrequencyDivisorValue: FrequencyDivisorValue

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCD Get command which is used to Get frequency divisor for the gathering. Refer to the Controller's manual to get the command description.

8.2.2.12 **DCD Set**

Syntax

int DCD_Set(int FrequencyDivisorValue, out string errstring)

FrequencyDivisorValue: FrequencyDivisorValue

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCD Set command which is used to Set frequency divisor for the gathering. Refer to the Controller's manual to get the command description.

8.2.2.13 **DCM_Get**

Syntax

int DCM_Get(out int TriggerMode, out string errstring)

TriggerMode: TriggerMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCM Get command which is used to Get the trigger mode for the gathering. Refer to the Controller's manual to get the command description.

8.2.2.14 **DCM_Set**

Syntax

int DCM_Set(int TriggerMode, out string errstring)

TriggerMode: TriggerMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCM Set command which is used to Set the trigger mode for the gathering. Refer to the Controller's manual to get the command description.

8.2.2.15 **DCN_Get**

Syntax

int DCN_Get(out int DataNumber, out string errstring)

DataNumber: DataNumber

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCN Get command which is used to Get number of data points to be gathered. Refer to the Controller's manual to get the command description.

8.2.2.16 **DCN_Set**

Syntax

int DCN_Set(int DataNumber, out string errstring)

DataNumber: DataNumber

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCN Set command which is used to Set number of data points to be gathered. Refer to the Controller's manual to get the command description.

8.2.2.17 **DCS_Get**

Syntax

int DCS_Get(out int EnableGathering, out string errstring)

EnableGathering: EnableGathering

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCS Get command which is used to Enable/Disable gathering or get gathering status. Refer to the Controller's manual to get the command description.

8.2.2.18 **DCS_Set**

Syntax

int DCS_Set (int EnableGathering, out string errstring)

EnableGathering: EnableGathering

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCS Set command which is used to Enable/Disable gathering or get gathering status. Refer to the Controller's manual to get the command description.

8.2.2.19 **DCT**

Syntax

int DCT(out double GatheredLines, out string errstring)

GatheredLines: GatheredLines

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCT Get command which is used to Get all gathered lines. Refer to the Controller's manual to get the command description.

8.2.2.20 **DCV_Get**

Syntax

int DCV_Get(out int GatheredData, out string errstring)

GatheredData: GatheredData

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DCV Get command which is used to Get the data to be gathered with a 7-bits decimal value. Refer to the Controller's manual to get the command description.

8.2.2.21 **DV Get**

Syntax

int DV_Get(out double Voltage, out string errstring)

Voltage: Voltage

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DV Get command which is used to Get driver voltage. Refer to the Controller's manual to get the command description.

8.2.2.22 **DV Set**

Syntax

int DV_Set(double Voltage, out string errstring)

Voltage: Voltage

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous DV Set command which is used to Set driver voltage. Refer to the Controller's manual to get the command description.

8.2.2.23 **ENF Get**

Syntax

int ENF_Get(out int FilterFrequency, out string errstring)

FilterFrequency: FilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ENF Get command which is used to Get the Encoder position filter frequency. Refer to the Controller's manual to get the command description.

8.2.2.24 **ENF Set**

Syntax

int ENF_Set(int FilterFrequency, out string errstring)

FilterFrequency: FilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ENF Set command which is used to Set the Encoder position filter frequency. Refer to the Controller's manual to get the command description.

8.2.2.25 ENP_Get

Syntax

int ENP_Get(out double EncoderPitch, out string errstring)

EncoderPitch: EncoderPitch

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ENP Get command which is used to Get the encoder pitch. Refer to the Controller's manual to get the command description.

8.2.2.26 ENP_Set

Syntax

int ENP_Set(double EncoderPitch, out string errstring)

EncoderPitch: EncoderPitch

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ENP Set command which is used to Set the encoder pitch. Refer to the Controller's manual to get the command description.

8.2.2.27 EQF_Get

Syntax

int EQF_Get(out int PositionFilterFrequency, out string errstring)

PositionFilterFrequency: PositionFilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQF Get command which is used to Get the position filter frequency. Refer to the Controller's manual to get the command description.

8.2.2.28 EQF_Set

Syntax

int EQF_Set(int PositionFilterFrequency, out string errstring)

PositionFilterFrequency: PositionFilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQF Set command which is used to Set the position filter frequency. Refer to the Controller's manual to get the command description.

8.2.2.29 EQP_Get

Syntax

int EQP_Get(out int AquadBOutputPolarity, out string errstring)

AquadBOutputPolarity: AquadBOutputPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQP Get command which is used to Get the AquadB output polarity. Refer to the Controller's manual to get the command description.

8.2.2.30 EQP_Set

Syntax

int EQP_Set(int AquadBOutputPolarity, out string errstring)

AquadBOutputPolarity: AquadBOutputPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQP Set command which is used to Set the AquadB output polarity. Refer to the Controller's manual to get the command description.

8.2.2.31 EQR_Get

Syntax

int EQR_Get(out int PositionInterpolation, out string errstring)

PositionInterpolation: PositionInterpolation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQR Get command which is used to Get the position interpolation. Refer to the Controller's manual to get the command description.

8.2.2.32 EQR_Set

Syntax

int EQR_Set(int PositionInterpolation, out string errstring)

PositionInterpolation: PositionInterpolation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous EQR Set command which is used to Set the position interpolation. Refer to the Controller's manual to get the command description.

8.2.2.33 **FD_Get**

Syntax

int FD_Get(out double LowPassFilter, out string errstring)

LowPassFilter: LowPassFilter

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FD Get command which is used to Get low pass filter for Kd. Refer to the Controller's manual to get the command description.

8.2.2.34 **FD_Set**

Syntax

int FD_Set(double LowPassFilter, out string errstring)

LowPassFilter: LowPassFilter

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FD Set command which is used to Set low pass filter for Kd. Refer to the Controller's manual to get the command description.

8.2.2.35 **FE_Get**

Syntax

int FE_Get(out double FollowingError, out string errstring)

FollowingError: FollowingError

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FE Get command which is used to Get following error limit. Refer to the Controller's manual to get the command description.

8.2.2.36 **FE_Set**

Syntax

int FE_Set(double FollowingError, out string errstring)

FollowingError: FollowingError

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FE Set command which is used to Set following error limit. Refer to the Controller's manual to get the command description.

8.2.2.37 FF_Get**Syntax**

int FF_Get(out double FrictionCompensation, out string errstring)

FrictionCompensation: FrictionCompensation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FF Get command which is used to Get friction compensation. Refer to the Controller's manual to get the command description.

8.2.2.38 FF_Set**Syntax**

int FF_Set(double FrictionCompensation, out string errstring)

FrictionCompensation: FrictionCompensation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FF Set command which is used to Set friction compensation. Refer to the Controller's manual to get the command description.

8.2.2.39 FL_Get**Syntax**

int FL_Get(out double PIDCutOfFrequency, out string errstring)

PIDCutOfFrequency: PIDCutOfFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FL Get command which is used to Get PID cut of frequency. Refer to the Controller's manual to get the command description.

8.2.2.40 FL_Set**Syntax**

int FL_Set(double PIDCutOfFrequency, out string errstring)

PIDCutOfFrequency: PIDCutOfFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FL Set command which is used to Set PID cut of frequency. Refer to the Controller's manual to get the command description.

8.2.2.41 **FMC_Get**

Syntax

int FMC_Get(out double CarriageMasse, out string errstring)

CarriageMasse: CarriageMasse

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMC Get command which is used to Get carriage masse. Refer to the Controller's manual to get the command description.

8.2.2.42 **FMC_Set**

Syntax

int FMC_Set(double CarriageMasse, out string errstring)

CarriageMasse: CarriageMasse

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMC Set command which is used to Set carriage masse. Refer to the Controller's manual to get the command description.

8.2.2.43 **FML_Get**

Syntax

int FML_Get(out double ForceLimite, out string errstring)

ForceLimite: ForceLimite

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FML Get command which is used to Get force limite. Refer to the Controller's manual to get the command description.

8.2.2.44 **FML_Set**

Syntax

int FML_Set(double ForceLimite, out string errstring)

ForceLimite: ForceLimite

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FML Set command which is used to Set force limite. Refer to the Controller's manual to get the command description.

8.2.2.45 FMP_Get**Syntax**

int FMP_Get(out double PayloadMasse, out string errstring)

PayloadMasse: PayloadMasse

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMP Get command which is used to Get PayloadMasse. Refer to the Controller's manual to get the command description.

8.2.2.46 FMP_Set**Syntax**

int FMP_Set(double PayloadMasse, out string errstring)

PayloadMasse: PayloadMasse

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMP Set command which is used to Set PayloadMasse. Refer to the Controller's manual to get the command description.

8.2.2.47 FMS_Get**Syntax**

int FMS_Get(out double ScalingForce, out string errstring)

ScalingForce: ScalingForce

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMS Get command which is used to Get scaling force. Refer to the Controller's manual to get the command description.

8.2.2.48 FMS_Set**Syntax**

int FMS_Set(double ScalingForce, out string errstring)

ScalingForce: ScalingForce

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FMS Set command which is used to Set scaling force. Refer to the Controller's manual to get the command description.

8.2.2.49 **FSM_Get**

Syntax

int FSM_Get(out int Password, out string errstring)

Password: Password

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FSM Get command which is used to Send the password to allow factory settings or serial number modifications. Refer to the Controller's manual to get the command description.

8.2.2.50 **FSM_Set**

Syntax

int FSM_Set(string Password, out string errstring)

Password: Password

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FSM Set command which is used to Send the password to allow factory settings or serial number modifications. Refer to the Controller's manual to get the command description.

8.2.2.51 **FSR**

Syntax

int FSR(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous FSR Set command which is used to Restore all parameters to factory settings. Refer to the Controller's manual to get the command description.

8.2.2.52 **GIC_Get**

Syntax

int GIC_Get(out int DataCaptureTriggerPolarity, out string errstring)

DataCaptureTriggerPolarity: DataCaptureTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIC Get command which is used to Get the polarity of input trigger 2 for start motion trigger . Refer to the Controller's manual to get the command description.

8.2.2.53 GIC_Set

Syntax

int GIC_Set(int DataCaptureTriggerPolarity, out string errstring)

DataCaptureTriggerPolarity: DataCaptureTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIC Set command which is used to Set the polarity of input trigger 2 for start motion trigger . Refer to the Controller's manual to get the command description.

8.2.2.54 GIM_Get

Syntax

int GIM_Get(out int StartMotionTriggerPolarity, out string errstring)

StartMotionTriggerPolarity: StartMotionTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIM Get command which is used to Get the polarity of input trigger 1 for data capture. Refer to the Controller's manual to get the command description.

8.2.2.55 GIM_Set

Syntax

int GIM_Set(int StartMotionTriggerPolarity, out string errstring)

StartMotionTriggerPolarity: StartMotionTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIM Set command which is used to Set the polarity of input trigger 1 for data capture. Refer to the Controller's manual to get the command description.

8.2.2.56 GIT_Get

Syntax

int GIT_Get(out int DataCaptureTriggerType, out string errstring)

DataCaptureTriggerType: DataCaptureTriggerType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIT Get command which is used to Get Set the type of input trigger 2 (0: data capture / 1: PGR direction / 2: goto reference) . Refer to the Controller's manual to get the command description.

8.2.2.57 GIT Set

Syntax

int GIT_Set(int DataCaptureTriggerType, out string errstring)

DataCaptureTriggerType: DataCaptureTriggerType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GIT Set command which is used to Set the type of input trigger 2 (0: data capture / 1: PGR direction / 2: goto reference) . Refer to the Controller's manual to get the command description.

8.2.2.58 GOF Get

Syntax

int GOF_Get(out int PCOOutputTriggerPolarity , out string errstring)

PCOOutputTriggerPolarity : PCOOutputTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOF Get command which is used to Get the position filter frequency for the PCO output. Refer to the Controller's manual to get the command description.

8.2.2.59 GOF Set

Syntax

int GOF_Set(int PCOOutputTriggerPolarity , out string errstring)

PCOOutputTriggerPolarity : PCOOutputTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOF Set command which is used to Set the position filter frequency for the PCO output. Refer to the Controller's manual to get the command description.

8.2.2.60 GOP Get

Syntax

int GOP_Get(out int PositionFilterFrequency, out string errstring)

PositionFilterFrequency: PositionFilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOP Get command which is used to Get the polarity of output trigger 2 (PCO). Refer to the Controller's manual to get the command description.

8.2.2.61 GOP Set

Syntax

int GOP_Set(int PositionFilterFrequency, out string errstring)

PositionFilterFrequency: PositionFilterFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOP Set command which is used to Set the polarity of output trigger 2 (PCO). Refer to the Controller's manual to get the command description.

8.2.2.62 GOM Get

Syntax

int GOM_Get(out int MotionOutputTriggerPolarity, out string errstring)

MotionOutputTriggerPolarity: MotionOutputTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOM Get command which is used to Get the polarity of output trigger 1 for motion trigger. Refer to the Controller's manual to get the command description.

8.2.2.63 GOM Set

Syntax

int GOM_Set(int MotionOutputTriggerPolarity, out string errstring)

MotionOutputTriggerPolarity: MotionOutputTriggerPolarity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOM Set command which is used to Set the polarity of output trigger 1 for motion trigger. Refer to the Controller's manual to get the command description.

8.2.2.64 GO Get

Syntax

int GOT_Get(out int MotionOutputTriggerType, out string errstring)

MotionOutputTriggerType: MotionOutputTriggerType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOT Get command which is used to Get the type of output trigger 1 for motion trigger. Refer to the Controller's manual to get the command description.

8.2.2.65 GOT Set

Syntax

int GOT_Set(int MotionOutputTriggerType, out string errstring)

MotionOutputTriggerType: MotionOutputTriggerType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOT Set command which is used to Set the type of output trigger 1 for motion trigger. Refer to the Controller's manual to get the command description.

8.2.2.66 GOW Get

Syntax

int GOW_Get(out double PCOOutputTriggerPulseWidth, out string errstring)

PCOOutputTriggerPulseWidth: PCOOutputTriggerPulseWidth

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOW Get command which is used to Get the pulse width for PCO output trigger . Refer to the Controller's manual to get the command description.

8.2.2.67 GOW Set

Syntax

int GOW_Set(double PCOOutputTriggerPulseWidth, out string errstring)

PCOOutputTriggerPulseWidth: PCOOutputTriggerPulseWidth

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GOW Set command which is used to Set the pulse width for PCO output trigger . Refer to the Controller's manual to get the command description.

8.2.2.68 GPE Get

Syntax

int GPE_Get(out int EnablePCO, out string errstring)

EnablePCO: EnablePCO

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPE Get command which is used to Enable/Disable PCO function. Refer to the Controller's manual to get the command description.

8.2.2.69 **GPE Set**

Syntax

int GPE_Set(int EnablePCO, out string errstring)

EnablePCO: EnablePCO

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPE Set command which is used to Enable/Disable PCO function. Refer to the Controller's manual to get the command description.

8.2.2.70 **GPI Get**

Syntax

int GPI_Get(out double InitialPosition, out string errstring)

InitialPosition: InitialPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPI Get command which is used to Get the Initial position for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.71 **GPI Set**

Syntax

int GPI_Set(double InitialPosition, out string errstring)

InitialPosition: InitialPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPI Set command which is used to Set the Initial position for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.72 **GPL Get**

Syntax

int GPL_Get(out double LastPosition, out string errstring)

LastPosition: LastPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPL Get command which is used to Get the last position for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.73 GPL Set

Syntax

int GPL_Set(double LastPosition, out string errstring)

LastPosition: LastPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPL Set command which is used to Set the last position for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.74 GPS Get

Syntax

int GPS_Get(out double StepDistance, out string errstring)

StepDistance: StepDistance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPS Get command which is used to Get the step distance for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.75 GPS Set

Syntax

int GPS_Set(double StepDistance, out string errstring)

StepDistance: StepDistance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous GPS Set command which is used to Set the step distance for PCO trigger. Refer to the Controller's manual to get the command description.

8.2.2.76 HO Get

Syntax

int HO_Get(out double HomeOffset, out string errstring)

HomeOffset: HomeOffset

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous HO Get command which is used to Get HOME search offset. Refer to the Controller's manual to get the command description.

8.2.2.77 HO Set**Syntax**

int HO_Set(double HomeOffset, out string errstring)

HomeOffset: HomeOffset

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous HO Set command which is used to Set HOME search offset. Refer to the Controller's manual to get the command description.

8.2.2.78 HT Get**Syntax**

int HT_Get(out int HomeType, out string errstring)

HomeType: HomeType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous HT Get command which is used to Get HOME search type. Refer to the Controller's manual to get the command description.

8.2.2.79 HT Set**Syntax**

int HT_Set(int HomeType, out string errstring)

HomeType: HomeType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous HT Set command which is used to Set HOME search type. Refer to the Controller's manual to get the command description.

8.2.2.80 ID Get**Syntax**

int ID_Get(out string StageIdentifier, out string errstring)

StageIdentifier: StageIdentifier

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ID Get command which is used to Get stage identifier. Refer to the Controller's manual to get the command description.

8.2.2.81 ID_Set**Syntax**

int ID_Set(string StageIdentifier, out string errstring)

StageIdentifier: StageIdentifier

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ID Set command which is used to Set stage identifier. Refer to the Controller's manual to get the command description.

8.2.2.82 IE**Syntax**

int IE(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous IE Set command which is used to Start the execution of the Initialization sequence as defined by the IT command. Refer to the Controller's manual to get the command description. return: 0 in success and -1 on failure

8.2.2.83 ITA_Get**Syntax**

int ITA_Get(out double InitializationAccelerationLevel, out string errstring)

InitializationAccelerationLevel: InitializationAccelerationLevel

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ITA Get command which is used to Get initialization acceleration level. Refer to the Controller's manual to get the command description.

8.2.2.84 ITA_Set**Syntax**

int ITA_Set(double InitializationAccelerationLevel, out string errstring)

InitializationAccelerationLevel: InitializationAccelerationLevel

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ITA Set command which is used to Set initialization acceleration level. Refer to the Controller's manual to get the command description.

8.2.2.85 ITD_Get**Syntax**

int ITD_Get(out int InitializationCycleDuration, out string errstring)

InitializationCycleDuration: InitializationCycleDuration

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ITD Get command which is used to Get initialization cycle duration. Refer to the Controller's manual to get the command description.

8.2.2.86 ITD_Set**Syntax**

int ITD_Set(int InitializationCycleDuration, out string errstring)

InitializationCycleDuration: InitializationCycleDuration

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ITD Set command which is used to Set initialization cycle duration. Refer to the Controller's manual to get the command description.

8.2.2.87 JA_Get**Syntax**

int JA_Get(out double AccelerationTime, out string errstring)

AccelerationTime: AccelerationTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JA Get command which is used to Get acceleration in jogging mode with a remote keypad. Refer to the Controller's manual to get the command description.

8.2.2.88 JA_Set**Syntax**

int JA_Set(double AccelerationTime, out string errstring)

AccelerationTime: AccelerationTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JA Set command which is used to Set acceleration in jogging mode with a remote keypad. Refer to the Controller's manual to get the command description.

8.2.2.89 JD**Syntax**

int JD(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JD Set command which is used to Leave JOGGING state. Refer to the Controller's manual to get the command description.

8.2.2.90 JM Get**Syntax**

int JM_Get(out int KeypadState, out string errstring)

KeypadState: KeypadState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JM Get command which is used to Enable/Disable Keypad. Refer to the Controller's manual to get the command description.

8.2.2.91 JM Set**Syntax**

int JM_Set(int KeypadState, out string errstring)

KeypadState: KeypadState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JM Set command which is used to Enable/Disable Keypad. Refer to the Controller's manual to get the command description.

8.2.2.92 JR Get**Syntax**

int JR_Get(out double JerkTime, out string errstring)

JerkTime: JerkTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JR Get command which is used to Get jerk time. Refer to the Controller's manual to get the command description.

8.2.2.93 JR Set**Syntax**

int JR_Set(double JerkTime, out string errstring)

JerkTime: JerkTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JR Set command which is used to Set jerk time. Refer to the Controller's manual to get the command description.

8.2.2.94 JV Get**Syntax**

int JV_Get(out double VelocityTime, out string errstring)

VelocityTime: VelocityTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JV Get command which is used to Get Velocity in jogging mode with a remote keypad . Refer to the Controller's manual to get the command description.

8.2.2.95 JV Set**Syntax**

int JV_Set(double VelocityTime, out string errstring)

VelocityTime: VelocityTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous JV Set command which is used to Set Velocity in jogging mode with a remote keypad . Refer to the Controller's manual to get the command description.

8.2.2.96 KD Get**Syntax**

int KD_Get(out double DerivativeGain, out string errstring)

DerivativeGain: DerivativeGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KD Get command which is used to Get derivative gain. Refer to the Controller's manual to get the command description.

8.2.2.97 KD Set**Syntax**

int KD_Set(double DerivativeGain, out string errstring)

DerivativeGain: DerivativeGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KD Set command which is used to Set derivative gain. Refer to the Controller's manual to get the command description.

8.2.2.98 KGD Get**Syntax**

int KGD_Get(out double KdGain, out string errstring)

KdGain: KdGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGD Get command which is used to Get Kd PID gain. Refer to the Controller's manual to get the command description.

8.2.2.99 KGD Set**Syntax**

int KGD_Set(double KdGain, out string errstring)

KdGain: KdGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGD Set command which is used to Set Kd PID gain. Refer to the Controller's manual to get the command description.

8.2.2.100 KGF Get**Syntax**

int KGF_Get(out double KFormGain, out string errstring)

KFormGain: KFormGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGF Get command which is used to Set /Get Kform gain. Refer to the Controller's manual to get the command description.

8.2.2.101 **KGF Set**

Syntax

int KGF_Set(double KFormGain, out string errstring)

KFormGain: KFormGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGF Set command which is used to Set /Get Kform gain. Refer to the Controller's manual to get the command description.

8.2.2.102 **KGI Get**

Syntax

int KGI_Get(out double KiGain, out string errstring)

KiGain: KiGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGI Get command which is used to Get Ki gain. Refer to the Controller's manual to get the command description.

8.2.2.103 **KGI Set**

Syntax

int KGI_Set(double KiGain, out string errstring)

KiGain: KiGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGI Set command which is used to Set Ki gain. Refer to the Controller's manual to get the command description.

8.2.2.104 **KGP Get**

Syntax

int KGP_Get(out double KpGain, out string errstring)

KpGain: KpGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGP Get command which is used to Get Kp gain. Refer to the Controller's manual to get the command description.

8.2.2.105 **KGP Set**

Syntax

int KGP_Set(double KpGain, out string errstring)

KpGain: KpGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KGP Set command which is used to Set Kp gain. Refer to the Controller's manual to get the command description.

8.2.2.106 **KI Get**

Syntax

int KI_Get(out double IntegralGain, out string errstring)

IntegralGain: IntegralGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KI Get command which is used to Get integral gain. Refer to the Controller's manual to get the command description.

8.2.2.107 **KI Set**

Syntax

int KI_Set(double IntegralGain, out string errstring)

IntegralGain: IntegralGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KI Set command which is used to Set integral gain. Refer to the Controller's manual to get the command description.

8.2.2.108 **KP Get**

Syntax

int KP_Get(out double ProportionalGain, out string errstring)

ProportionalGain: ProportionalGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KP Get command which is used to Get proportional gain. Refer to the Controller's manual to get the command description.

8.2.2.109 **KP_Set**

Syntax

int KP_Set(double ProportionalGain, out string errstring)

ProportionalGain: ProportionalGain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KP Set command which is used to Set proportional gain. Refer to the Controller's manual to get the command description.

8.2.2.110 **KS_Get**

Syntax

int KS_Get(out double IntegralSaturation, out string errstring)

IntegralSaturation: IntegralSaturation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KS Get command which is used to Get the integral saturation level of the PID control loop . Refer to the Controller's manual to get the command description.

8.2.2.111 **KS_Set**

Syntax

int KS_Set(double IntegralSaturation, out string errstring)

IntegralSaturation: IntegralSaturation

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous KS Set command which is used to Set the integral saturation level of the PID control loop . Refer to the Controller's manual to get the command description.

8.2.2.112 **LT_Get**

Syntax

int LT_Get(out int HomeType, out string errstring)

HomeType: HomeType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous LT Get command which is used to Get the limits type of the encoder plug. Refer to the Controller's manual to get the command description.

8.2.2.113 LT_Set

Syntax

int LT_Set(int HomeType, out string errstring)

HomeType: HomeType

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous LT Set command which is used to Set the limits type of the encoder plug. Refer to the Controller's manual to get the command description.

8.2.2.114 MDA_Get

Syntax

int MDA_Get(out double MeanPeriod, out string errstring)

MeanPeriod: MeanPeriod

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDA Get command which is used to Get the Mean Period. Refer to the Controller's manual to get the command description.

8.2.2.115 MDA_Set

Syntax

int MDA_Set(double MeanPeriod, out string errstring)

MeanPeriod: MeanPeriod

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDA Set command which is used to Set the Mean Period. Refer to the Controller's manual to get the command description.

8.2.2.116 MDC_Get

Syntax

int MDC_Get(out double CheckingTime, out string errstring)

CheckingTime: CheckingTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDC Get command which is used to Get the Checking Time. Refer to the Controller's manual to get the command description.

8.2.2.117 MDC Set**Syntax**

int MDC_Set(double CheckingTime, out string errstring)

CheckingTime: CheckingTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDC Set command which is used to Set the Checking Time. Refer to the Controller's manual to get the command description.

8.2.2.118 MDM Get**Syntax**

int MDM_Get(out int MotionDoneMode, out string errstring)

MotionDoneMode: MotionDoneMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDM Get command which is used to Get the Motion Done Mode . Refer to the Controller's manual to get the command description.

8.2.2.119 MDM Set**Syntax**

int MDM_Set(int MotionDoneMode, out string errstring)

MotionDoneMode: MotionDoneMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDM Set command which is used to Set the Motion Done Mode . Refer to the Controller's manual to get the command description.

8.2.2.120 MDP Get**Syntax**

int MDP_Get(out double PositionThreshold, out string errstring)

PositionThreshold: PositionThreshold

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDP Get command which is used to Get the Position Threshold. Refer to the Controller's manual to get the command description.

8.2.2.121 MDP Set

Syntax

int MDP_Set(double PositionThreshold, out string errstring)

PositionThreshold: PositionThreshold

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDP Set command which is used to Set the Position Threshold. Refer to the Controller's manual to get the command description.

8.2.2.122 MDT Get

Syntax

int MDT_Get(out double Timeout, out string errstring)

Timeout: Timeout

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDT Get command which is used to Get the Timeout. Refer to the Controller's manual to get the command description.

8.2.2.123 MDT Set

Syntax

int MDT_Set(double Timeout, out string errstring)

Timeout: Timeout

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDT Set command which is used to Set the Timeout. Refer to the Controller's manual to get the command description.

8.2.2.124 MDV Get

Syntax

int MDV_Get(out double VelocityThreshold, out string errstring)

VelocityThreshold: VelocityThreshold

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDV Get command which is used to Get the Velocity Threshold. Refer to the Controller's manual to get the command description.

8.2.2.125 MDV_Set

Syntax

int MDV_Set(double VelocityThreshold, out string errstring)

VelocityThreshold: VelocityThreshold

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MDV Set command which is used to Set the Velocity Threshold. Refer to the Controller's manual to get the command description.

8.2.2.126 MM_Get

Syntax

int MM_Set(double DisableState, out string errstring)

DisableState: DisableState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MM Set command which is used to Enter/Leave DISABLE state. Refer to the Controller's manual to get the command description.

8.2.2.127 MM_Set

Syntax

int MM_Set(double DisableState, out string errstring)

DisableState: DisableState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MM Set command which is used to Enter/Leave DISABLE state. Refer to the Controller's manual to get the command description.

8.2.2.128 MP_Get

Syntax

int MP_Get(out double MagnetPeriod, out string errstring)

MagnetPeriod: MagnetPeriod

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MP Get command which is used to Get the magnet period. Refer to the Controller's manual to get the command description.

8.2.2.129 MP_Set**Syntax**

int MP_Set(double MagnetPeriod, out string errstring)

MagnetPeriod: MagnetPeriod

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MP Set command which is used to Set the magnet period. Refer to the Controller's manual to get the command description.

8.2.2.130 MT_Get**Syntax**

int MT_Get(out double Timeout, out string errstring)

Timeout: Timeout

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MT Get command which is used to Get the timeout value of the PD commands. Refer to the Controller's manual to get the command description.

8.2.2.131 MT_Set**Syntax**

int MT_Set(double Timeout, out string errstring)

Timeout: Timeout

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous MT Set command which is used to Set the timeout value of the PD commands. Refer to the Controller's manual to get the command description.

8.2.2.132 NFF_Get**Syntax**

int NFF_Get(out double NotchFilterCenterFreq, out string errstring)

NotchFilterCenterFreq: Notch Filter Center Frequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to get the notch filter center frequency value of the PID control loop.

8.2.2.133 NFF_Set

Syntax

int NFF_Set(double NotchFilterCenterFreq, out string errstring)

NotchFilterCenterFreq: Notch Filter Center Frequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to set notch filter frequency parameter of the PID control loop which can be saved in the controller's non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for notch filter. This value is not saved in the controller's memory and will be lost after reboot.

8.2.2.134 NFG_Get

Syntax

int NFG_Get(out double NotchFilterGain, out string errstring)

NotchFilterGain: Notch Filter Gain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to get the notch filter gain value of the PID control loop.

8.2.2.135 NFG_Set

Syntax

int NFG_Set(double NotchFilterGain, out string errstring)

NotchFilterGain: Notch Filter Gain

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to set notch filter frequency gain parameter of the PID control loop which can be saved in the controller's non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for notch filter. This value is not saved in the controller's memory and will be lost after reboot.

To disable the notch functionality set the gain to 1.

8.2.2.136 NFW_Get

Syntax

int NFW_Get(out double NotchFilterFreqWidth, out string errstring)

NotchFilterFreqWidth: Notch Filter Frequency Width

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to get the notch filter frequency width value of the PID control loop.

8.2.2.137 NFW_Set

Syntax

int NFW_Set(double NotchFilterFreqWidth, out string errstring)

NotchFilterFreqWidth: Notch Filter Frequency Width

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to set notch filter frequency width parameter of the PID control loop which can be saved in the controller's nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for notch filter. This value is not saved in the controller's memory and will be lost after reboot.

8.2.2.138 OH_Get

Syntax

int OH_Get(out double HomeVelocity, out string errstring)

HomeVelocity: HomeVelocity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous OH Get command which is used to Get HOME search velocity. Refer to the Controller's manual to get the command description.

8.2.2.139 OH_Set

Syntax

int OH_Set(double HomeVelocity, out string errstring)

HomeVelocity: HomeVelocity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous OH Set command which is used to Set HOME search velocity. Refer to the Controller's manual to get the command description.

8.2.2.140 OR**Syntax**

int OR(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous OR Set command which is used to Execute HOME search. Refer to the Controller's manual to get the command description.

8.2.2.141 OT_Get**Syntax**

int OT_Get(out double HomeTimeOut, out string errstring)

HomeTimeOut: HomeTimeOut

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous OT Get command which is used to Get HOME search time-out. Refer to the Controller's manual to get the command description.

8.2.2.142 OT_Set**Syntax**

int OT_Set(double HomeTimeOut, out string errstring)

HomeTimeOut: HomeTimeOut

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous OT Set command which is used to Set HOME search time-out. Refer to the Controller's manual to get the command description.

8.2.2.143 PA_Get**Syntax**

int PA_Get(out double TargetPosition, out string errstring)

TargetPosition: TargetPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PA Get command which is used to Move absolute. Refer to the Controller's manual to get the command description.

8.2.2.144 PA_Set**Syntax**

int PA_Set(double TargetPosition, out string errstring)

TargetPosition: TargetPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PA Set command which is used to Move absolute. Refer to the Controller's manual to get the command description.

8.2.2.145 PD**Syntax**

int PD(double Displacement, out string DisplacementStatus, out string errstring)

Displacement: The relative move value. DisplacementStatus: The controller returns PD1 if the motion is done or PD0 in case an error has occurred.

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to initiate a relative move. When received, the positioner will move, with the predefined acceleration and velocity, to a new target position away from the current target position. Refer to the Controller's manual to get the command description.

8.2.2.146 PG_Get**Syntax**

int PG_Get(out double Displacement, out string errstring)

Displacement: Displacement

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PG Get command which is used to Get triggered move distance. Refer to the Controller's manual to get the command description.

8.2.2.147 PG_Set**Syntax**

int PG_Set(double Displacement, out string errstring)

Displacement: Displacement

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PG Set command which is used to Set triggered move distance. Refer to the Controller's manual to get the command description.

8.2.2.148 PI_Get**Syntax**

int PI_Get(out double PIDIntegerTime, out string errstring)

PIDIntegerTime: PIDIntegerTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PI Get command which is used to Get PID Integration time. Refer to the Controller's manual to get the command description.

8.2.2.149 PI_Set**Syntax**

int PI_Set(double PIDIntegerTime, out string errstring)

PIDIntegerTime: PIDIntegerTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PI Set command which is used to Set PID Integration time. Refer to the Controller's manual to get the command description.

8.2.2.150 PR_Get**Syntax**

int PR_Get(out double Step, out string errstring)

Step: Step

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PR Get command which is used to Move relative. Refer to the Controller's manual to get the command description.

8.2.2.151 PR_Set**Syntax**

int PR_Set(double Step, out string errstring)

Step: Step

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PR Set command which is used to Move relative. Refer to the Controller's manual to get the command description.

8.2.2.152 PTA

Syntax

int PTA(out double AccelerationDistance, out string errstring)

AccelerationDistance: AccelerationDistance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PTA Get command which is used to Get acceleration distance. Refer to the Controller's manual to get the command description.

8.2.2.153 PTT

Syntax

int PTT(double Displacement, out string Timeout, out string errstring)

Displacement: The relative move value

Timeout: The needed time, in seconds, to execute a relative move

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PTA Get command which is used to Get acceleration distance. Refer to the Controller's manual to get the command description.

8.2.2.154 PW Get

Syntax

int PW_Get(out int ConfigurationMode, out string errstring)

ConfigurationMode: ConfigurationMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PW Get command which is used to Enter/Leave CONFIGURATION state. Refer to the Controller's manual to get the command description.

8.2.2.155 PW Set

Syntax

int PW_Set(int ConfigurationMode, out string errstring)

ConfigurationMode: ConfigurationMode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous PW Set command which is used to Enter/Leave CONFIGURATION state. Refer to the Controller's manual to get the command description.

8.2.2.156 QCF_Get

Syntax

int QCF_Get(out double CurrentLoopCutOffFrequency, out string errstring)

CurrentLoopCutOffFrequency: CurrentLoopCutOffFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCF Get command which is used to Get the current loop Cutoff frequency. Refer to the Controller's manual to get the command description.

8.2.2.157 QCF_Set

Syntax

int QCF_Set(double CurrentLoopCutOffFrequency, out string errstring)

CurrentLoopCutOffFrequency: CurrentLoopCutOffFrequency

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCF Set command which is used to Set the current loop Cutoff frequency. Refer to the Controller's manual to get the command description.

8.2.2.158 QCL_Get

Syntax

int QCL_Get(out double motorInductance, out string errstring)

motorInductance: motorInductance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCL Get command which is used to Get the motors Inductance. Refer to the Controller's manual to get the command description.

8.2.2.159 QCL_Set

Syntax

int QCL_Set(double motorInductance, out string errstring)

motorInductance: motorInductance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCL Set command which is used to Set the motors Inductance. Refer to the Controller's manual to get the command description.

8.2.2.160 QCR_Get

Syntax

int QCR_Get(out double motorResistance, out string errstring)

motorResistance: motorResistance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCR Get command which is used to Get motors resistance. Refer to the Controller's manual to get the command description.

8.2.2.161 QCR_Set

Syntax

int QCR_Set(double motorResistance, out string errstring)

motorResistance: motorResistance

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QCR Set command which is used to Set motors resistance. Refer to the Controller's manual to get the command description.

8.2.2.162 QIL_Get

Syntax

int QIL_Get(out double MotorPeakCurrentLimits, out string errstring)

MotorPeakCurrentLimits: MotorPeakCurrentLimits

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIL Get command which is used to Get motors peak current limits. Refer to the Controller's manual to get the command description.

8.2.2.163 QIL_Set

Syntax

int QIL_Set(double MotorPeakCurrentLimits, out string errstring)

MotorPeakCurrentLimits: MotorPeakCurrentLimits

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIL Set command which is used to Set motors peak current limits. Refer to the Controller's manual to get the command description.

8.2.2.164 QIR_Get

Syntax

int QIR_Get(out double MotorMsCurrentLimits, out string errstring)

MotorMsCurrentLimits: MotorMsCurrentLimits

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIR Get command which is used to Get motors ms current limits. Refer to the Controller's manual to get the command description.

8.2.2.165 QIL_Set

Syntax

int QIR_Set(double MotorMsCurrentLimits, out string errstring)

MotorMsCurrentLimits: MotorMsCurrentLimits

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIR Set command which is used to Set motors ms current limits. Refer to the Controller's manual to get the command description.

8.2.2.166 QIT_Get

Syntax

int QIT_Get(out double MotorMsCurrentAveragingTime, out string errstring)

MotorMsCurrentAveragingTime: MotorMsCurrentAveragingTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIT Get command which is used to Get motors ms current averaging time. Refer to the Controller's manual to get the command description.

8.2.2.167 QIT_Set

Syntax

int QIT_Set(double MotorMsCurrentAveragingTime, out string errstring)

MotorMsCurrentAveragingTime: MotorMsCurrentAveragingTime

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous QIT Set command which is used to Set motors ms current averaging time. Refer to the Controller's manual to get the command description.

8.2.2.168 RAA**Syntax**

int RAA(out double AnalogInputValue, out string errstring)

AnalogInputValue: AnalogInputValue

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous RAA Get command which is used to Get analog input value. Refer to the Controller's manual to get the command description.

8.2.2.169 RAB**Syntax**

int RAB(out double AnalogInputValue, out string errstring)

AnalogInputValue: AnalogInputValue

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous RAB Get command which is used to Get analog input value. Refer to the Controller's manual to get the command description.

8.2.2.170 RF Get**Syntax**

int RF_Get(out double ReferencePosition, out string errstring)

ReferencePosition: ReferencePosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous RF Get command which is used to Get the reference position. Refer to the Controller's manual to get the command description.

8.2.2.171 RF Set**Syntax**

int RF_Set(double ReferencePosition, out string errstring)

ReferencePosition: ReferencePosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous RF Set command which is used to Set the reference position. Refer to the Controller's manual to get the command description.

8.2.2.172 RS**Syntax**

int RS(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous RS Set command which is used to Reset controller. Refer to the Controller's manual to get the command description.

8.2.2.173 SC_Get**Syntax**

int SC_Get(out int ControlLoopState, out string errstring)

ControlLoopState: ControlLoopState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SC Get command which is used to Get control loop state. Refer to the Controller's manual to get the command description.

8.2.2.174 SC_Set**Syntax**

int SC_Set(int ControlLoopState, out string errstring)

ControlLoopState: ControlLoopState

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SC Set command which is used to Set control loop state. Refer to the Controller's manual to get the command description.

8.2.2.175 SL_Get**Syntax**

int SL_Get(out double LeftLimit, out string errstring)

LeftLimit: LeftLimit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SL Get command which is used to Get negative software limit. Refer to the Controller's manual to get the command description.

8.2.2.176 SL_Set**Syntax**

int SL_Set(double LeftLimit, out string errstring)

LeftLimit: LeftLimit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SL Set command which is used to Set negative software limit. Refer to the Controller's manual to get the command description.

8.2.2.177 SN_Get**Syntax**

int SN_Get(out int SerialNumber, out string errstring)

SerialNumber: SerialNumber

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SN Get command which is used to Get serial number. Refer to the Controller's manual to get the command description.

8.2.2.178 SN_Set**Syntax**

int SN_Set(int SerialNumber, out string errstring)

SerialNumber: SerialNumber

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SN Set command which is used to Set serial number. Refer to the Controller's manual to get the command description.

8.2.2.179 SR_Get**Syntax**

int SR_Get(out double RightLimit, out string errstring)

RightLimit: RightLimit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SR Get command which is used to Get positive software limit. Refer to the Controller's manual to get the command description.

8.2.2.180 SR_Set**Syntax**

int SR_Set(double RightLimit, out string errstring)

RightLimit: RightLimit

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous SR Set command which is used to Set positive software limit. Refer to the Controller's manual to get the command description.

8.2.2.181 ST**Syntax**

int ST(out string errstring)

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ST Set command which is used to Stop motion. Refer to the Controller's manual to get the command description.

8.2.2.182 TB**Syntax**

int TB(string ErrorCode, out string ErrorMessage, out string errstring)

ErrorCode: The error code returned by the TE command ErrorMessage: The meaning of the error code returned by the controller

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous TE Get command which is used to Get last command error. Refer to the Controller's manual to get the command description.

8.2.2.183 TE**Syntax**

int TE(out string LastCommandError, out string errstring)

LastCommandError: LastCommandError

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous TE Get command which is used to Get last command error. Refer to the Controller's manual to get the command description.

8.2.2.184 TH**Syntax**

int TH(out double SetPointPosition, out string errstring)

SetPointPosition: SetPointPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous TH Get command which is used to Get set-point position. Refer to the Controller's manual to get the command description.

8.2.2.185 TP**Syntax**

int TP(out double CurrentPosition, out string errstring)

CurrentPosition: CurrentPosition

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous TP Get command which is used to Get current position. Refer to the Controller's manual to get the command description.

8.2.2.186 TS**Syntax**

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

ErrorCode: ErrorCode

StatusCode: StatusCode

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous TS Get command which is used to Get positioner error and controller state. Refer to the Controller's manual to get the command description.

8.2.2.187 VA_Get**Syntax**

int VA_Get(out double Velocity, out string errstring)

Velocity: Velocity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous VA Get command which is used to Get velocity. Refer to the Controller's manual to get the command description.

8.2.2.188 VA_Set**Syntax**

int VA_Set(double Velocity, out string errstring)

Velocity: Velocity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous VA Set command which is used to Set velocity. Refer to the Controller's manual to get the command description.

8.2.2.189 VAM**Syntax**

int VAM(out double MaximumVelocity, out string errstring)

MaximumVelocity: MaximumVelocity

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous VAM Get command which is used to Get maximum velocity. Refer to the Controller's manual to get the command description.

8.2.2.190 VE**Syntax**

int VE(out string ControllerVersion, out string errstring)

ControllerVersion: ControllerVersion

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous VE Get command which is used to Get controller revision information. Refer to the Controller's manual to get the command description.

8.2.2.191 ZT**Syntax**

int ZT(out List<string> Parameters, out string errstring)

Parameters: Parameters

errString: The failure reason

return: 0 in success and -1 on failure

Description

This function is used to process synchronous ZT Get command which is used to Get all axis parameters. Refer to the Controller's manual to get the command description.



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

