

# CONEX-CC

## *Single-Axis DC Motion with Controller/Driver*



**Newport®**  
Experience | Solutions

## Command Interface Manual

V2.0.x

*For Motion, Think Newport™*



# Table of Contents

<b>1.0</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Purpose .....	1
1.2	Overview .....	1
<b>2.0</b>	<b>Command Interface.....</b>	<b>2</b>
2.1	Constructor .....	2
2.2	Functions .....	2
2.2.1	General .....	2
2.2.1.1	OpenInstrument .....	2
2.2.1.2	CloseInstrument .....	2
2.2.1.3	GetDevices .....	2
2.2.1.4	WriteToInstrument .....	3
2.2.2	Commands.....	3
2.2.2.1	AC_Get .....	3
2.2.2.2	AC_Set .....	3
2.2.2.3	BA_Get .....	4
2.2.2.4	BA_Set .....	4
2.2.2.5	BH_Get .....	4
2.2.2.6	BH_Set .....	5
2.2.2.7	DV_Get .....	5
2.2.2.8	DV_Set .....	5
2.2.2.9	FD_Get .....	6
2.2.2.10	FD_Set .....	6
2.2.2.11	FE_Get .....	6
2.2.2.12	FE_Set .....	7
2.2.2.13	FF_Get .....	7
2.2.2.14	FF_Set .....	7
2.2.2.15	HT_Get .....	8
2.2.2.16	HT_Set .....	8
2.2.2.17	ID_Get .....	8
2.2.2.18	ID_Set .....	9
2.2.2.19	JR_Get .....	9
2.2.2.20	JR_Set .....	9
2.2.2.21	KD_Get .....	10
2.2.2.22	KD_Set .....	10
2.2.2.23	KI_Get .....	10
2.2.2.24	KI_Set .....	11
2.2.2.25	KP_Get .....	11

2.2.2.26	KP_Set.....	11
2.2.2.27	KV_Get.....	12
2.2.2.28	KV_Set.....	12
2.2.2.29	MM_Get.....	12
2.2.2.30	MM_Set.....	13
2.2.2.31	OH_Get.....	13
2.2.2.32	OH_Set.....	13
2.2.2.33	OR.....	14
2.2.2.34	OT_Get.....	14
2.2.2.35	OT_Set.....	14
2.2.2.36	PA_Get.....	15
2.2.2.37	PA_Set.....	15
2.2.2.38	PR_Get.....	15
2.2.2.39	PR_Set.....	16
2.2.2.40	PT_Get.....	16
2.2.2.41	PT_Set.....	16
2.2.2.42	PW_Get.....	17
2.2.2.43	PW_Set.....	17
2.2.2.44	QIL_Get.....	17
2.2.2.45	QIL_Set.....	18
2.2.2.46	QIR_Get.....	18
2.2.2.47	QIR_Set.....	18
2.2.2.48	QIT_Get.....	19
2.2.2.49	QIT_Set.....	19
2.2.2.50	RS.....	19
2.2.2.51	RS485.....	20
2.2.2.52	SA_Get.....	20
2.2.2.53	SA_Set.....	20
2.2.2.54	SC_Get.....	21
2.2.2.55	SC_Set.....	21
2.2.2.56	SE.....	21
2.2.2.57	SL_Get.....	22
2.2.2.58	SL_Set.....	22
2.2.2.59	SR_Get.....	22
2.2.2.60	SR_Set.....	23
2.2.2.61	ST.....	23
2.2.2.62	SU_Get.....	23
2.2.2.63	SU_Set.....	24
2.2.2.64	TB.....	24
2.2.2.65	TE.....	24
2.2.2.66	TH.....	25
2.2.2.67	TK_Get.....	25
2.2.2.68	TK_Set.....	25
2.2.2.69	TP.....	26

2.2.2.70

TS .....

26

2.2.2.71

VA\_Get .....

26

2.2.2.72

VA\_Set.....

27

2.2.2.73

VE .....

27

2.2.2.74

ZT .....

27

3.0

Python Example.....

28

Service Form .....

31



# CONEX-CC

## Single-Axis DC Motor Controller/Driver

### 1.0 Introduction

---

#### 1.1 Purpose

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

#### 1.2 Overview

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

---

#### NOTE

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

For each command function, refer to the CONEX-CC programmer’s manual.

---

## 2.0 Command Interface

---

### 2.1 Constructor

ConexCC()

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

### 2.2 Functions

#### 2.2.1 General

##### 2.2.1.1 OpenInstrument

###### Syntax

int OpenInstrument(string strDeviceKey)

string strDeviceKey: device key

return: 0 = successful or -1 = failure

###### Description

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

##### 2.2.1.2 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.

##### 2.2.1.3 GetDevices

###### Syntax

string[] GetDevices()

return: list of connected devices available to communicate

###### Description

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



#### 2.2.1.4 WriteToInstrument

##### Syntax

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

command: Instrument command

response: Response of the command

stage: Instrument Stage

return:

##### Description

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

#### 2.2.2 **Commands**

##### 2.2.2.1 AC\_Get

##### Syntax

int AC\_Get(int controllerAddress, out double outAcceleration, out string errString)

controllerAddress: Address of Controller

outAcceleration: outAcceleration

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.

##### 2.2.2.2 AC\_Set

##### Syntax

int AC\_Set(int controllerAddress, double inAcceleration, out string errString)

controllerAddress: Address of Controller

inAcceleration: inAcceleration.

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.

### 2.2.2.3 BA\_Get

#### Syntax

int BA\_Get(int controllerAddress, out double outBacklash, out string errString)

controllerAddress: Address of Controller

outBacklash: outBacklash

errString: The failure reason

return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous BA Get command which is used to Get backlash compensation.

### 2.2.2.4 BA\_Set

#### Syntax

int BA\_Set(int controllerAddress, double inBacklash, out string errString)

controllerAddress: Address of Controller

inBacklash: inBacklash.

errString: The failure reason

return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous BA Set command which is used to Set backlash compensation.

### 2.2.2.5 BH\_Get

#### Syntax

int BH\_Get(int controllerAddress, out double outHysteresis, out string errString)

controllerAddress: Address of Controller

outHysteresis: outHysteresis

errString: The failure reason

return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous BH Get command which is used to Get hysteresis compensation.

#### 2.2.2.6 BH\_Set

##### **Syntax**

int BH\_Set(int controllerAddress, double inHysteresis, out string errString)

controllerAddress: Address of Controller

inHysteresis: inHysteresis.

errString: The failure reason

return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous BH Set command which is used to Set hysteresis compensation.

#### 2.2.2.7 DV\_Get

##### **Syntax**

int DV\_Get(int controllerAddress, out double outDriverVoltage, out string errString)

controllerAddress: Address of Controller

outDriverVoltage: outDriverVoltage

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.

#### 2.2.2.8 DV\_Set

##### **Syntax**

int DV\_Set(int controllerAddress, double inDriverVoltage, out string errString)

controllerAddress: Address of Controller

inDriverVoltage: inDriverVoltage.

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.

### 2.2.2.9 FD\_Get

#### Syntax

int FD\_Get(int controllerAddress, out double outLowPassFilterKd, out string errString)

controllerAddress: Address of Controller

outLowPassFilterKd: outLowPassFilterKd

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.

### 2.2.2.10 FD\_Set

#### Syntax

int FD\_Set(int controllerAddress, double inLowPassFilterKd, out string errString)

controllerAddress: Address of Controller

inLowPassFilterKd: inLowPassFilterKd.

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.

### 2.2.2.11 FE\_Get

#### Syntax

int FE\_Get(int controllerAddress, out double outFollowingError, out string errString)

controllerAddress: Address of Controller

outFollowingError: outFollowingError

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.

**2.2.2.12 FE\_Set****Syntax**

int FE\_Set(int controllerAddress, double inFollowingError, out string errString)

controllerAddress: Address of Controller

inFollowingError: inFollowingError.

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.

**2.2.2.13 FF\_Get****Syntax**

int FF\_Get(int controllerAddress, out double outFrictionCompensation, out string errString)

controllerAddress: Address of Controller

outFrictionCompensation: outFrictionCompensation

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.

**2.2.2.14 FF\_Set****Syntax**

int FF\_Set(int controllerAddress, double inFrictionCompensation, out string errString)

controllerAddress: Address of Controller

inFrictionCompensation: inFrictionCompensation.

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.

#### 2.2.2.15 HT\_Get

##### Syntax

int HT\_Get(int controllerAddress, out int outHomeType, out string errString)

controllerAddress: Address of Controller

outHomeType: outHomeType

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.

#### 2.2.2.16 HT\_Set

##### Syntax

int HT\_Set(int controllerAddress, int inHomeType, out string errString)

controllerAddress: Address of Controller

inHomeType: inHomeType.

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.

#### 2.2.2.17 ID\_Get

##### Syntax

int ID\_Get(int controllerAddress, out string outStageIdentifier, out string errString)

controllerAddress: Address of Controller

outStageIdentifier: outStageIdentifier

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.

**2.2.2.18 ID\_Set****Syntax**

int ID\_Set(int controllerAddress, string inStageIdentifier, out string errString)

controllerAddress: Address of Controller

inStageIdentifier: inStageIdentifier.

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.

**2.2.2.19 JR\_Get****Syntax**

int JR\_Get(int controllerAddress, out double outJerkTime, out string errString)

controllerAddress: Address of Controller

outJerkTime: outJerkTime

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.

**2.2.2.20 JR\_Set****Syntax**

int JR\_Set(int controllerAddress, double inJerkTime, out string errString)

controllerAddress: Address of Controller

inJerkTime: inJerkTime.

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.

**2.2.2.21 KD\_Get****Syntax**

int KD\_Get(int controllerAddress, out double outDerivativeGain, out string errString)

controllerAddress: Address of Controller

outDerivativeGain: outDerivativeGain

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.

**2.2.2.22 KD\_Set****Syntax**

int KD\_Set(int controllerAddress, double inDerivativeGain, out string errString)

controllerAddress: Address of Controller

inDerivativeGain: inDerivativeGain.

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.

**2.2.2.23 KI\_Get****Syntax**

int KI\_Get(int controllerAddress, out double outIntegralGain, out string errString)

controllerAddress: Address of Controller

outIntegralGain: outIntegralGain

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.



**2.2.2.24 KI\_Set****Syntax**

int KI\_Set(int controllerAddress, double inIntegralGain, out string errString)

controllerAddress: Address of Controller

inIntegralGain: inIntegralGain.

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.

**2.2.2.25 KP\_Get****Syntax**

int KP\_Get(int controllerAddress, out double outProportionalGain, out string errString)

controllerAddress: Address of Controller

outProportionalGain: outProportionalGain

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.

**2.2.2.26 KP\_Set****Syntax**

int KP\_Set(int controllerAddress, double inProportionalGain, out string errString)

controllerAddress: Address of Controller

inProportionalGain: inProportionalGain.

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.

**2.2.2.27 KV\_Get****Syntax**

int KV\_Get(int controllerAddress, out double outVelocityFeedForward, out string errString)

controllerAddress: Address of Controller

outVelocityFeedForward: outVelocityFeedForward

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous KV Get command which is used to Get velocity feed forward.

**2.2.2.28 KV\_Set****Syntax**

int KV\_Set(int controllerAddress, double inVelocityFeedForward, out string errString)

controllerAddress: Address of Controller

inVelocityFeedForward: inVelocityFeedForward.

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous KV Set command which is used to Set velocity feed forward.

**2.2.2.29 MM\_Get****Syntax**

int MM\_Get(int controllerAddress, out string outState, out string errString)

controllerAddress: Address of Controller

outState: outState

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous MM Get command which is used to Enter/Leave DISABLE state.

**2.2.2.30 MM\_Set****Syntax**

int MM\_Set(int controllerAddress, int inState, out string errString)

controllerAddress: Address of Controller

inState: inState.

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.

**2.2.2.31 OH\_Get****Syntax**

int OH\_Get(int controllerAddress, out double outHomeVelocity, out string errString)

controllerAddress: Address of Controller

outHomeVelocity: outHomeVelocity

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.

**2.2.2.32 OH\_Set****Syntax**

int OH\_Set(int controllerAddress, double inHomeVelocity, out string errString)

controllerAddress: Address of Controller

inHomeVelocity: inHomeVelocity.

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.

### 2.2.2.33 OR

#### **Syntax**

int OR(int controllerAddress, out string errString)

clientID: Instrument ID

controllerAddress: controllerAddress identifying the Address of Controller

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.

### 2.2.2.34 OT\_Get

#### **Syntax**

int OT\_Get(int controllerAddress, out double outHomeTimeOut, out string errString)

controllerAddress: Address of Controller

outHomeTimeOut: outHomeTimeOut

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.

### 2.2.2.35 OT\_Set

#### **Syntax**

int OT\_Set(int controllerAddress, double inHomeTimeOut, out string errString)

controllerAddress: Address of Controller

inHomeTimeOut: inHomeTimeOut.

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.

**2.2.2.36 PA\_Get****Syntax**

int PA\_Get(int controllerAddress, out double outTargetPosition, out string errString)

controllerAddress: Address of Controller

outTargetPosition: outTargetPosition

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.

**2.2.2.37 PA\_Set****Syntax**

int PA\_Set(int controllerAddress, double inTargetPosition, out string errString)

controllerAddress: Address of Controller

inTargetPosition: inTargetPosition.

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.

**2.2.2.38 PR\_Get****Syntax**

int PR\_Get(int controllerAddress, out double outStep, out string errString)

controllerAddress: Address of Controller

outStep: outStep

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.

**2.2.2.39 PR\_Set****Syntax**

int PR\_Set(int controllerAddress, double inStep, out string errString)

controllerAddress: Address of Controller

inStep: inStep.

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.

**2.2.2.40 PT\_Get****Syntax**

int PT\_Get(int controllerAddress, out double outMotionTime, out string errString)

controllerAddress: Address of Controller

outMotionTime: outMotionTime

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous PT Get command which is used to Get motion time for a relative move.

**2.2.2.41 PT\_Set****Syntax**

int PT\_Set(int controllerAddress, double inMotionTime, out string errString)

controllerAddress: Address of Controller

inMotionTime: inMotionTime.

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous PT Set command which is used to Get motion time for a relative move.

#### 2.2.2.42 PW\_Get

##### Syntax

int PW\_Get(int controllerAddress, out int outState, out string errString)

controllerAddress: Address of Controller

outState: outState

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.

#### 2.2.2.43 PW\_Set

##### Syntax

int PW\_Set(int controllerAddress, int inState, out string errString)

controllerAddress: Address of Controller

inState: inState.

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.

---

##### **NOTE**

**The PW command is limited to 100 writes. Unit failure due to excessive use of the PW command is not covered by warranty.**

**The PW command is used to change the configuration parameters that are stored in memory, and not parameters that are needed to be changed on the fly.**

---

#### 2.2.2.44 QIL\_Get

##### Syntax

int QIL\_Get(int controllerAddress, out double outMotorPeakLimit, out string errString)

controllerAddress: Address of Controller

outMotorPeakLimit: outMotorPeakLimit

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 motor's peak current limits.

**2.2.2.45 QIL\_Set****Syntax**

int QIL\_Set(int controllerAddress, double inMotorPeakLimit, out string errString)

controllerAddress: Address of Controller

inMotorPeakLimit: inMotorPeakLimit.

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 motor's peak current limits.

**2.2.2.46 QIR\_Get****Syntax**

int QIR\_Get(int controllerAddress, out double outMotorMsLimit, out string errString)

controllerAddress: Address of Controller

outMotorMsLimit: outMotorMsLimit

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 motor's ms current limits.

**2.2.2.47 QIR\_Set****Syntax**

int QIR\_Set(int controllerAddress, double inMotorMsLimit, out string errString)

controllerAddress: Address of Controller

inMotorMsLimit: inMotorMsLimit.

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 motor's ms current limits.



**2.2.2.48 QIT\_Get****Syntax**

int QIT\_Get(int controllerAddress, out double outMotorAveragingTime, out string errString)

controllerAddress: Address of Controller

outMotorAveragingTime: outMotorAveragingTime

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 motor's ms current averaging time.

**2.2.2.49 QIT\_Set****Syntax**

int QIT\_Set(int controllerAddress, double inMotorAveragingTime, out string errString)

controllerAddress: Address of Controller

inMotorAveragingTime: inMotorAveragingTime.

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 motor's ms current averaging time.

**2.2.2.50 RS****Syntax**

int RS(int controllerAddress, out string errString)

clientID: Instrument ID

controllerAddress: controllerAddress identifying the Address of Controller

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.

#### 2.2.2.51 RS485

##### **Syntax**

int RS485(int controllerAddress, out string errString)

clientID: Instrument ID

controllerAddress: controllerAddress identifying the Address of Controller

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's address to 1.

#### 2.2.2.52 SA\_Get

##### **Syntax**

int SA\_Get(int controllerAddress, out int outRS485Address, out string errString)

controllerAddress: Address of Controller

outRS485Address: outRS485Address

errString: The failure reason

return: 0 in success and -1 on failure

##### **Description**

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

#### 2.2.2.53 SA\_Set

##### **Syntax**

int SA\_Set(int controllerAddress, int inRS485Address, out string errString)

controllerAddress: Address of Controller

inRS485Address: inRS485Address.

errString: The failure reason

return: 0 in success and -1 on failure

##### **Description**

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

**2.2.2.54 SC\_Get****Syntax**

int SC\_Get(int controllerAddress, out int outControlLoopState, out string errString)

controllerAddress: Address of Controller

outControlLoopState: outControlLoopState

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.

**2.2.2.55 SC\_Set****Syntax**

int SC\_Set(int controllerAddress, int inControlLoopState, out string errString)

controllerAddress: Address of Controller

inControlLoopState: inControlLoopState.

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.

**2.2.2.56 SE****Syntax**

int SE(int controllerAddress, double inTargetPosition, out string errString)

controllerAddress: Address of Controller

inTargetPosition: inTargetPosition.

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SE Set command which is used to Configure/Execute simultaneous started move.

### 2.2.2.57 SL\_Get

#### **Syntax**

int SL\_Get(int controllerAddress, out double outNegativeLimit, out string errString)

controllerAddress: Address of Controller

outNegativeLimit: outNegativeLimit

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.

### 2.2.2.58 SL\_Set

#### **Syntax**

int SL\_Set(int controllerAddress, double inNegativeLimit, out string errString)

controllerAddress: Address of Controller

inNegativeLimit: inNegativeLimit.

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.

### 2.2.2.59 SR\_Get

#### **Syntax**

int SR\_Get(int controllerAddress, out double outPositiveLimit, out string errString)

controllerAddress: Address of Controller

outPositiveLimit: outPositiveLimit

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.

**2.2.2.60 SR\_Set****Syntax**

int SR\_Set(int controllerAddress, double inPositiveLimit, out string errString)

controllerAddress: Address of Controller

inPositiveLimit: inPositiveLimit.

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.

**2.2.2.61 ST****Syntax**

int ST(int controllerAddress, out string errString)

clientID: Instrument ID

controllerAddress: controllerAddress identifying the Address of Controller

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.

**2.2.2.62 SU\_Get****Syntax**

int SU\_Get(int controllerAddress, out double outEncoderIncrement, out string errString)

controllerAddress: Address of Controller

outEncoderIncrement: outEncoderIncrement

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SU Get command which is used to Get encoder increment value.

**2.2.2.63 SU\_Set****Syntax**

int SU\_Set(int controllerAddress, double inEncoderIncrement, out string errString)

controllerAddress: Address of Controller

inEncoderIncrement: inEncoderIncrement.

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SU Set command which is used to Set encoder increment value.

**2.2.2.64 TB****Syntax**

int TB(int controllerAddress, string inError, out string outError, out string errString)

controllerAddress: Address of Controller

inError: inError.

outError: outError

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

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

**2.2.2.65 TE****Syntax**

int TE(int controllerAddress, out string outError, out string errString)

controllerAddress: Address of Controller

outError: outError

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.

**2.2.2.66** TH**Syntax**

int TH(int controllerAddress, out double outSetPointPosition, out string errString)

controllerAddress: Address of Controller

outSetPointPosition: outSetPointPosition

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.

**2.2.2.67** TK\_Get**Syntax**

int TK\_Get(int controllerAddress, out string outState, out string errString)

controllerAddress: Address of Controller

outState: outState

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TK Get command which is used to Enter/Leave ReadyT state.

**2.2.2.68** TK\_Set**Syntax**

int TK\_Set(int controllerAddress, int inState, out string errString)

controllerAddress: Address of Controller

inState: inState.

errString: The failure reason

return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TK Set command which is used to Enter/Leave ReadyT state.

**2.2.2.69 TP****Syntax**

int TP(int controllerAddress, out double outCurrentPosition, out string errString)

controllerAddress: Address of Controller

outCurrentPosition: outCurrentPosition

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.

**2.2.2.70 TS****Syntax**

int TS(int controllerAddress, out string errorCode, out string controllerState, out string errString)

controllerAddress: Address of Controller

errorCode: errorCode

controllerState: controllerState

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.

**2.2.2.71 VA\_Get****Syntax**

int VA\_Get(int controllerAddress, out double outVelocity, out string errString)

controllerAddress: Address of Controller

outVelocity: outVelocity

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.



**2.2.2.72** VA\_Set**Syntax**

int VA\_Set(int controllerAddress, double inVelocity, out string errString)

controllerAddress: Address of Controller

inVelocity: inVelocity.

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.

**2.2.2.73** VE**Syntax**

int VE(int controllerAddress, out string outControllerVersion, out string errString)

controllerAddress: Address of Controller

outControllerVersion: outControllerVersion

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.

**2.2.2.74** ZT**Syntax**

int ZT(int controllerAddress, out List<string> AxisParameters, out string errString)

controllerAddress: Address of Controller

AxisParameters: AxisParameters

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.

### 3.0 Python Example

```

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

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

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

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

import System
=====

# Instrument Initialization
# The key should have double slashes since
# (one of them is escape character)
instrument="COM25"
print 'Instrument Key=>', instrument

# create a device instance and open communication with the instrument
CC = ConexCC()
ret = CC.OpenInstrument(instrumentKey)
print 'OpenInstrument => ', ret

# Get positive software limit
result, response, errString = CC.SR_Get(1)
if result == 0 :
    print 'positive software limit=>', response
else:
    print 'Error=>',errString

```

```
# Get negative software limit
result, response, errString = CC.SL_Get(1)
if result == 0 :
    print 'negative software limit=>', response
else:
    print 'Error=>',errString

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

# Get current position
result, response, errString = CC.TP(1)
if result == 0 :
    print 'position=>', response
else:
    print 'Error=>',errString

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



Service Form

Your Local Representative

Tel.:  
Fax:

Name:  
Company:  
Address:  
Country:  
P.O. Number:  
Item(s) Being Returned:  
Model#:

Return authorization #:  
*(Please obtain prior to return of item)*  
  
Date:  
Phone Number:  
Fax Number:  
  
Serial #:

Description:

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



**Newport®**

Experience | Solutions

Visit Newport Online at:  
**[www.newport.com](http://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](mailto:sales@newport.com)

**Technical Support**

Tel.: (800) 222-6440  
e-mail: [tech@newport.com](mailto:tech@newport.com)

**Service, RMAs & Returns**

Tel.: (800) 222-6440  
e-mail: [rma.service@newport.com](mailto:rma.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-fr.com](mailto:france@newport-fr.com)

**Technical Support**

e-mail: [tech\\_europe@newport.com](mailto:tech_europe@newport.com)

**Service & Returns**

Tel.: +33 (0)2.38.40.51.55