

CONEX-AGP

**Agilis-P Controller
with Encoder Feedback**



©2026 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	Overview	1
2.0	Command Interface.....	2
2.1	Constructor	2
2.2	Functions	2
2.2.1	General Functions	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	Controller Command Functions.....	3
2.2.2.1	DB_Get.....	3
2.2.2.2	DB_Set.....	3
2.2.2.3	HT_Get.....	4
2.2.2.4	HT_Set.....	4
2.2.2.5	ID_Get.....	4
2.2.2.6	ID_Set.....	5
2.2.2.7	IF_Get.....	5
2.2.2.8	IF_Set	5
2.2.2.9	KI_Get.....	6
2.2.2.10	KI_Set.....	6
2.2.2.11	KP_Get	6
2.2.2.12	KP_Set.....	7
2.2.2.13	LF_Get.....	7
2.2.2.14	LF_Set	7
2.2.2.15	MM_Get	8
2.2.2.16	MM_Set.....	8
2.2.2.17	OR	8
2.2.2.18	PA_Get	9
2.2.2.19	PA_Set.....	9
2.2.2.20	PR_Get.....	9
2.2.2.21	PR_Set.....	10
2.2.2.22	PW_Get.....	10
2.2.2.23	PW_Set.....	11
2.2.2.24	RS.....	11
2.2.2.25	RS485	11

2.2.2.26	SA_Get	12
2.2.2.27	SA_Set.....	12
2.2.2.28	SL_Get.....	12
2.2.2.29	SL_Set	13
2.2.2.30	SR_Get.....	13
2.2.2.31	SR_Set.....	13
2.2.2.32	ST	14
2.2.2.33	SU_Get	14
2.2.2.34	SU_Set.....	14
2.2.2.35	TB.....	15
2.2.2.36	TE.....	15
2.2.2.37	TH	15
2.2.2.38	TP.....	16
2.2.2.39	TS	16
2.2.2.40	VE	16
2.2.2.41	ZT.....	17
Service Form		19



Agilis-P Controller with Encoder Feedback CONEX-AGP

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-AGP device.

1.2 Overview

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

These commands work in both synchronous and asynchronous mode in the NStruct environment or not. The communication is based on the NStruct server.

NOTE

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

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

2.0 Command Interface

2.1 Constructor

ConexAGP()

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

2.2 Functions

2.2.1 General Functions

2.2.1.1 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.

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 devices

Description

This function is used to get the list of the connected devices

2.2.1.4 WriteToInstrument

Syntax

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

componentID: Instrument ID

command: Instrument command

response: Response of the command

stage: Instrument Stage

Return: Communication error code

Description

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

2.2.2 Controller Command Functions

2.2.2.1 DB_Get

Syntax

int DB_Get(int componentID, int controllerAddress, out double outDeadband, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outDeadband: outDeadband

errString: The failure reason

Return: 0 in success and -1 on failure

Description

This function is used to process synchronous DB Get command which is used to Get corrector deadband.

2.2.2.2 DB_Set

Syntax

int DB_Set(int componentID, int controllerAddress, double inDeadband, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inDeadband: inDeadband.

errString: The failure reason

Return: 0 in success and -1 on failure

Description

This function is used to process synchronous DB Set command which is used to Set corrector deadband.

2.2.2.3 HT_Get

Syntax

int HT_Get(int componentID, int controllerAddress, out int outHomeTypeValue, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outHomeTypeValue: outHomeTypeValue

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.

2.2.2.4 HT_Set

Syntax

int HT_Set(int componentID, int controllerAddress, int inHomeTypeValue, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inHomeTypeValue: inHomeTypeValue.

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.

2.2.2.5 ID_Get

Syntax

int ID_Get(int componentID, int controllerAddress, out string outStageIdentifier, out string errString)

componentID: Instrument ID

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.

2.2.2.6 ID_Set

Syntax

int ID_Set(int componentID, int controllerAddress, string inStageIdentifier, out string errString)

componentID: Instrument ID

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.

2.2.2.7 IF_Get

Syntax

int IF_Get(int componentID, int controllerAddress, out double outInterpolationFactor, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outInterpolationFactor: outInterpolationFactor

errString: The failure reason

Return: 0 in success and -1 on failure

Description

This function is used to process synchronous IF Get command which is used to Get interpolation factor.

2.2.2.8 IF_Set

Syntax

int IF_Set(int componentID, int controllerAddress, double inInterpolationFactor, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inInterpolationFactor: inInterpolationFactor.

errString: The failure reason

Return: 0 in success and -1 on failure

Description

This function is used to process synchronous IF Set command which is used to Set interpolation factor.

2.2.2.9 KI_Get

Syntax

int KI_Get(int componentID, int controllerAddress, out double outIntegralGain, out string errString)

componentID: Instrument ID

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.

2.2.2.10 KI_Set

Syntax

int KI_Set(int componentID, int controllerAddress, double inIntegralGain, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inIntegralGain: inIntegralGain.

errString: The failure reason

This function is used to process synchronous KI Set command which is used to Set integral gain.

Description

Return: 0 in success and -1 on failure

2.2.2.11 KP_Get

Syntax

int KP_Get(int componentID, int controllerAddress, out double outProportionalGain, out string errString)

componentID: Instrument ID

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.

2.2.2.12 KP_Set

Syntax

int KP_Set(int componentID, int controllerAddress, double inProportionalGain, out string errString)

componentID: Instrument ID

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.

2.2.2.13 LF_Get

Syntax

int LF_Get(int componentID, int controllerAddress, out double outFrequency, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outFrequency: outFrequency

errString: The failure reason

Return: 0 in success and -1 on failure

Description

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

2.2.2.14 LF_Set

Syntax

int LF_Set(int componentID, int controllerAddress, double inFrequency, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inFrequency: inFrequency.

errString: The failure reason

Return: 0 in success and -1 on failure

Description

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

2.2.2.15 MM_Get

Syntax

int MM_Get(int componentID, int controllerAddress, out string outState, out string errString)

componentID: Instrument ID

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.

2.2.2.16 MM_Set

Syntax

int MM_Set(int componentID, int controllerAddress, int inState, out string errString)

componentID: Instrument ID

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.

2.2.2.17 OR

Syntax

int OR(int componentID, 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 .

2.2.2.18 PA_Get

Syntax

int PA_Get(int componentID, int controllerAddress, out double outTarget, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outTarget: outTarget

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.

2.2.2.19 PA_Set

Syntax

int PA_Set(int componentID, int controllerAddress, double inTarget, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inTarget: inTarget.

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.

2.2.2.20 PR_Get

Syntax

int PR_Get(int componentID, int controllerAddress, out double outTarget, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outTarget: outTarget

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.

2.2.2.21 PR_Set

Syntax

int PR_Set(int componentID, int controllerAddress, double inTarget, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inTarget: inTarget.

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.

2.2.2.22 PW_Get

Syntax

int PW_Get(int componentID, int controllerAddress, out int outState, out string errString)

componentID: Instrument ID

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.

2.2.2.23 PW_Set

Syntax

int PW_Set(int componentID, int controllerAddress, int inState, out string errString)

componentID: Instrument ID

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.

2.2.2.24 RS

Syntax

int RS(int componentID, 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.

2.2.2.25 RS485

Syntax

int RS485(int componentID, 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.

2.2.2.26 SA_Get

Syntax

int SA_Get(int componentID, int controllerAddress, out int outAddress, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outAddress : outAddress

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.

2.2.2.27 SA_Set

Syntax

int SA_Set(int componentID, int controllerAddress, int inAddress, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inAddress : inAddress.

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.

2.2.2.28 SL_Get

Syntax

int SL_Get(int componentID, int controllerAddress, out double outLimit, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outLimit : outLimit

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.

2.2.2.29 SL_Set

Syntax

int SL_Set(int componentID, int controllerAddress, double inLimit, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inLimit : inLimit.

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.

2.2.2.30 SR_Get

Syntax

int SR_Get(int componentID, int controllerAddress, out double outLimit, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outLimit : outLimit

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.

2.2.2.31 SR_Set

Syntax

int SR_Set(int componentID, int controllerAddress, double inLimit, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inLimit : inLimit.

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.

2.2.2.32 ST

Syntax

int ST(int componentID, 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.

2.2.2.33 SU_Get

Syntax

int SU_Get(int componentID, int controllerAddress, out double outIncrementValue, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outIncrementValue : outIncrementValue

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.

2.2.2.34 SU_Set

Syntax

int SU_Set(int componentID, int controllerAddress, double inIncrementValue, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inIncrementValue : inIncrementValue.

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.

2.2.2.35 TB

Syntax

int TB(int componentID, int controllerAddress, string inErrorCode, out string outErrorCode, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

inErrorCode : inErrorCode.

outErrorCode : outErrorCode

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.

2.2.2.36 TE

Syntax

int TE(int componentID, int controllerAddress, out string outLastCommandError, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outLastCommandError : outLastCommandError

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.

2.2.2.37 TH

Syntax

int TH(int componentID, int controllerAddress, out double outPosition, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outPosition : outPosition

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.

2.2.2.38 TP**Syntax**

int TP(int componentID, int controllerAddress, out double outPosition, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outPosition : outPosition

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.

2.2.2.39 TS**Syntax**

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

componentID: Instrument ID

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.

2.2.2.40 VE**Syntax**

int VE(int componentID, int controllerAddress, out string outInformation, out string errString)

componentID: Instrument ID

controllerAddress: Address of Controller

outInformation : outInformation

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.

2.2.2.41 ZT

Syntax

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

componentID: Instrument ID

controllerAddress: Address of Controller

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 controller parameters.

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): _____

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



Visit MKS | Newport Online at:

www.newport.com

North America & Asia

Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA

Sales

Tel.: +1 (949)-863-3144
e-mail: sales@newport.com

Technical Support

Tel.: +1 (949)-863-3144
e-mail: tech@newport.com

Service, RMAs & Returns

Tel.: +1 (949)-863-3144
e-mail: service@newport.com

Europe

MICRO-CONTROLE Spectra-Physics S.A.S
7 rue des Plantes
45340 Beaune-la-Rolande
France

Sales Europe (EMEA)

Tel.: +49 (0) 6151-708-0
e-mail: germany@newport.com

Sales France

Tel.: +33 (0)1 60 91 68 68
e-mail: france@newport.com

Sales UK

Tel.: +44 (0)1235 432 710
e-mail: uk@newport.com

Technical Support

e-mail: tech_europe@newport.com

Service & Returns

Tel.: +33 (0)2 38 40 51 55
DST-BEA-RMA-service@newport.com