NEXCOM

NexROBO DLL API

Reference Manual

Manual Rev.: V1.0

Revision Date: 2016/12/5

Revise note:

Ver.	Description
V1.0	1. 2016/12/5: English version released.

1. NexROBO Library Overview

1.1. Basic Specification

1. NexROBO library support Microsoft® 7 (32 bit) with RTX 2012 update 3.
2. NexROBO library only support the robot which provided by NEXCOM.

1.2. API Overview

T : Type (of function call)

C : Callback only

X :not for Callback

B : both

(**T**: Type ➔ C: Callback only, X: Not for callback, B:Both)

Function Name	Description	T
Initialization Functions		
NexR_LinktoRobotKernel	Start NexRobotKernel	X
NexR_CloseRobotKernel	Close NexRobotKernel	X
NexR_GetRobotKernelStatus	Get robot status	X
NexR_GetMasteringData	Get robot mastering data	X
NexR_SetMasteringData	Set robot mastering data	X
NexR_ReInitializeDriverParameters	Reinitialize driver parameters	X
Robot Servo Basic Operation Functions		
NexR_RobotServoON	Start to servo on all robot axis	B
NexR_RobotServoOFF	Start to servo off all robot axis	B
NexR_ReSetDriveError	Reset all robot axis servo error	X
Robot Basic Operation Functions		
NexR_MovePtp	Robot PTP move	X
NexR_Robot_MoveLine	Robot LINE move	X
NexR_ChangePTPVel	Change Moving velocity of PTP and LINE	X
NexR_EmgStop	Emergency stop the Robot PTP movement	X
NexR_GetCmdDone	Check robot PTP or LINE movement is done or not	B
NexR_GetRobotPosition	Get robot position information	B
User Mode Operation Functions		
NexR_ResetSafetyAlarm	Reset user safety alarm	X
NexR_ChangeOpMode	Change operation mode	X
Digital I/O Operation Functions		
NexR_GetDIOCount	Get DIO number of current EtherCAT configuration	B
NexR_GetDIOSize	Get each DIO information	B
NexR_SetDIO	Set DO	B

1.3. Functions for Initialization

1.3.1. NexR_LinktoRobotKernel

C/C++ Syntax:

int NexR_LinktoRobotKernel();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for initializing and start the NexRobotKernel.

Attention! The function is not allowed to be used in Callback function.

Reference:

NexR_CloseRobotKernel ();

1.3.2. NexR_CloseRobotKernel

C/C++ Syntax:

int NexR_CloseRobotKernel ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for close the NexRobotKernel.

Attention! The function is not allowed to be used in Callback function.

1.3.3. NexR_GetRobotKernelStatus

C/C++ Syntax:

void NexR_GetRobotKernelStatus (ref int ret);

Parameters:

ref int ret:

Status of robot kernel.

Returned Values:

<No Parameters>

Usage:

Call the function for getting the status of robot kernel.

Attention! The function is not allowed to be used in Callback function.

1.3.4. NexR_GetMasteringData

C/C++ Syntax:

int NexR_GetMasteringData (double[] mastering_data);

Parameters:

double[] mastering_data:

Robot's Mastering Data.

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for getting the mastering data of current robot.

Attention! The function is not allowed to be used in Callback function.

Reference:

NexR_SetMasteringData ();

1.3.5. NexR_SetMasteringData

C/C++ Syntax:

int NexR_SetMasteringData (double[] mastering_data);

Parameters:

double[] mastering_data:

Robot's Mastering Data.

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for setting the mastering data of current robot.

Attention! The function is not allowed to be used in Callback function.

Reference:

NexR_GetMasteringData();

1.3.6. NexR_ReInitializeDriverParameters

C/C++ Syntax:

int NexR_ReInitializeDriverParameters();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

If robot moving makes some noise of motors, call the function for setting the initial parameters of each driver. But be sure drives is servo off.

Attention! The function is not allowed to be used in Callback function.

Attention! Only support 6R robot currently.

Reference:

NexR_RobotServoOFF();

1.4. Functions for Robot Servo Basic Operation

1.4.1. NexR_RobotServoON

C/C++ Syntax:

int NexR_RobotServoON ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for activating the servo on procedure of all robot axes, it will take some time to do it.

Reference:

NexR_RobotServoOFF();

1.4.2. NexR_RobotServoOFF

C/C++ Syntax:

int NexR_RobotServoOFF();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for activating the servo off procedure of all robot axes.

Reference:

NexR_RobotServoOFF();

1.4.3. NexR_ReSetDriveError

C/C++ Syntax:

int NexR_ReSetDriveError ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

When at least one of robot axes is in “Fault” state, please check the servo failure state and make all fault servos corrected, then use this function to clear/reset the “Fault” state afterward.

Attention! The function is not allowed to be used in Callback function.

1.5. Functions for Robot Basic Operation

1.5.1. NexR_MovePtp

C/C++ Syntax:

void NexR_MovePtp (ushort type, double[] target_data, double[] max_vel, double[] acc);

Parameters:

ushort type:

Set input parameter format of PTP(1: Axis angle; 2: TCP).

double[] target_data:

Set target data (deg) of each robot axis angle.

double[] max_vel:

Set maximum velocity (deg/s) of each robot axis

double[] acc:

Set acceleration (deg/s^2) of each robot axis

Returned Values:

<No Parameters>

Usage:

Call the function for moving all robot axes to desire destination using the fastest speed.

Attention! The function is not allowed to be used in Callback function.

Reference:

NexR_EmgStop();

NexR_GetCmdDone();

1.5.2. NexR_Robot_MoveLine

C/C++ Syntax:

int NexR_Robot_MoveLine (byte u8CmdType, double[] target_data, double max_vel, double acc);

Parameters:

byte u8CmdType:

Set input parameter format of line (1: Axis angle; 2: TCP).

double[] target_data:

Set Target data (deg) of each robot axis angle, if input parameter format of line is axis angle; Set Target data (mm) of robot TCP, if input parameter format of line is TCP.

double max_vel:

Set maximum velocity (mm/s) of robot TCP.

double acc:

Set acceleration (mm/s^2) of each robot TCP

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for moving robot TCP to desire destination using the fastest speed.

Attention! The function is not allowed to be used in Callback function.

Reference:

NexR_EmgStop ();

NexR_GetCmdDone ();

1.5.3. NexR_ChangePTPVel

C/C++ Syntax:

int NexR_ChangePTPVel (byte speed_ratio)

Parameters:

byte speed_ratio:

Moving speed ratio of PTP or LINE (0~100%).

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for change moving speed ratio of PTP or LINE.

Reference:

NexR_EmgStop ();

NexR_GetCmdDone ();

1.5.4. NexR_EmgStop

C/C++ Syntax:

int NexR_EmgStop ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for stopping the moving robot using the default maximum deceleration.

Reference:

NexR_MovePtp ();

NexR_GetCmdDone ();

1.5.5. NexR_GetCmdDone

C/C++ Syntax:

int NexR_GetCmdDone ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for checking the PTP movement of robot is done or not.

Reference:

NexR_MovePtp ();

NexR_EmgStop ();

1.5.6. NexR_GetRobotPosition

C/C++ Syntax:

void NexR_GetRobotPosition (ref NexRobotPos Robot_Pos);

Parameters:

ref NexRobotPos Robot_Pos:

Robot's position informatioin.

Type	Name	SizeConst
int[]	Axis_actpos	7
int[]	Axis_actvel	7
double[]	Axis_Data	7
double[]	Axis_Vel	7
double[]	Tcp	6
ushort[]	Servo_Status	7

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for checking the PTP movement of robot is done or not.

Reference:

NexR_MovePtp ();

NexR_EmgStop ();

1.6. Functions for User Mode Operation

1.6.1. NexR_ResetSafetyAlarm

C/C++ Syntax:

int NexR_ResetSafetyAlarm ();

Parameters:

<No Parameters>

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for resetting the safety alarm. After the safety alarm has been successful reset, the function “NexR_MovePtp” and “NexR_Robot_MoveLine” can be used again.

Reference:

NexR_MovePtp ();

NexR_Robot_MoveLine ();

1.6.2. NexR_ChangeOpMode

C/C++ Syntax:

int NexR_ChangeOpMode (byte mode)

Parameters:

byte mode:

0 → CSP mode

1 → CSV mode

2 → CST mode

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for change Operation mode.

1.7. Functions for Digital I/O Operation

1.7.1. NexR_GetDIOCount

C/C++ Syntax:

int NexR_GetDIOCount (ref uint DIO_Cnt);

Parameters:

ref uint DIO_Cnt:

Count of using digital input and output.

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for getting Digital IO module numbers.

Reference:

NexR_GetDIOSize ();

NexR_SetDIO ();

1.7.2. NexR_GetDIOSize

C/C++ Syntax:

BOOL NexR_GetDIOSize (uint DIO_Index, ref uint inputinbyte, ref uint outputinbyte);

Parameters:

uint DIO_Index:

The index of digital input and output.

ref uint inputinbyte:

The size length in byte of digital input.

ref uint outputinbyte:

The size length in byte of digital output.

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for getting DIO slave info such as input and output size, input and output status.

Reference:

NexR_GetDIOCount ();

NexR_SetDIO ();

1.7.3. NexR_SetDIO

C/C++ Syntax:

int NexR_SetDIO (uint DO_Index, uint[] output_data);

Parameters:

uint DO_Index:

Index of DO slave.

uint[] output_data:

Output data of DO slave.

Returned Values:

Error code is returned.

Zero is returned if function call is successful, while nonzero is returned when create line buffer failed.

Usage:

Call the function for getting DO slave output data.

Reference:

NexR_GetDIOCount ();

NexR_GetDIOSize ();