

XPS-RL

Universal High-Performance Motion Controller/Driver





Software Drivers Manual

V1.0.x

©2017 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

2.0	How	to Install .NET Drivers for XPS Controller?	2
2.	1 Rec	uirements	2
2.2	2 x86	Platform	2
2	3 x64	Platform	3
3.0	How	to Test .NET Drivers for XPS Controller?	5
4.0	How	to Access .Net C# Project	5
5.0	How	to Use XPS .NET Assembly from a Visual Studio C#	Project?6
5.	1 Ad	Reference to .NET Assembly	6
5.2	2 C#	Code Sources	6
	5.2	.1 C# Header	6
	5.2	.2 Add a Variable to Declare an "XPS" Object	6
	5.2	.3 Create an Instance of "XPS" Object	6
	5.2	.4 Open XPS Connection	7
	5.2	.5 Call "XPS" Functions	7
	5.2	.6 Close XPS Connection	7
6.0	How	to use XPS .NET Assembly from a LabVIEW projec	t?8
6.	1 Ad	Reference to .NET Assembly	8
6.2	2 Lab	VIEW Code Sources	8
7.0	How	to Use XPS .NET Assembly Under IronPython?	
7.	1 Ad	Reference to .NET Assembly	10
7.2	2 Iron	Python Code Source	10
	7.2	.1 IronPython Header	10
	7.2	.2 Create an Instance	
	7.2	.3 Open XPS Connection	
	7.2	.4 Call XPS Functions	11
		5 Class VDS Connection	11

8.0 E	low to	Use XPS .NET Assembly Under Matlab?	12
8.1	Add Re	ference to .NET Assembly	12
8.2	Matlab	Code Source	12
	8.2.1	Create an Instance	12
	8.2.2	Open XPS Connection	12
	8.2.3	Call XPS Functions	12
	8.2.4	Close XPS Connection	12
Servic	e Form		13



Universal High-Performance Motion Controller/Driver XPS-RL

1.0 What Are .Net drivers for XPS Controller?

.Net drivers support the creation of a user application that operates on a PC host computer and communicates with XPS-RL motion controllers. These drivers implements a rich set of controller operations and conceals from the application the complexity of low-level communication and synchronization with the controller.

The aim of this document is to explain customers how to integrate the XPS-RL .Net drivers into their programming language such as C#, Labview, IronPython and Matlab.

A separate Labview library (one vi per command) is available on our Newport website (XPS-RL web page).

2.0 How to Install .NET Drivers for XPS Controller?

2.1 Requirements

The PC host computer requires at least the .NET Framework 4.5.2 installed on it.

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

2.2 x86 Platform

First connect to the XPS-RL through the web site:



Once connected, go to the Documentation menu then Drivers submenu and download the Newport.XPS.CommandInterface x86.exe.



Once downloaded to the host PC, run the **Newport.XPS.CommandInterface_x86** executable file.

🙀 Newport.XPS.CommandInterface_x86 - InstallShield Wizard	X
Ready to Install the Program The wizard is ready to begin installation.	5
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.	
Current Settings:	
Setup Type:	
Typical	
Destination Folder:	
C:\Users\myname\Documents\Wewport\MotionControl\XPS-RL\	
User Information:	
Name: micro-controle	
Company: Microsoft	
InstallShield	
< Back Install Cancel	

Once installed, the .Net assembly "Newport.XPS.CommandInterface.dll" V1.0.0.0 is located in GAC for x86 platforms:

C:\Windows\Microsoft.NET\assembly\GAC_32\Newport.XPS.CommandInterface\v4.0_ 1.0.0.0_9a267756cf640dcf

The sample application "XPSApplicationTest.exe" is located under:

C:\Users\myname\Documents\Newport\MotionControl\XPS-RL

2.3 x64 Platform

First connect to the XPS-RL through the web site:



Once connected, go to the Documentation menu then Drivers submenu and download the Newport.XPS.CommandInterface x64.exe

Documentation » Dr	rivers
□ Display as listing	
Newport.XPS.CommandInter	Newport.XPS.CommandInter

face x86.exe

Once downloaded to the host PC, run the **Newport.XPS.CommandInterface_x64** executable file.

face x64.exe

🙀 Newport.XPS.CommandInterface_x64 - InstallShield Wizard	×
Ready to Install the Program The wizard is ready to begin installation.	٩
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.	
Current Settings:	
Setup Type:	
Typical	
Destination Folder:	
C: \Users \mynane \Documents \Wewport \MotionControl \XPS-RL \	
User Information:	
Name: micro-controle	
Company: Microsoft	
InstallShield	
< Back Install Cancel	

Once installed, the .Net assembly "Newport.XPS.CommandInterface.dll" V1.0.0.0 is located in GAC for x64 platforms:

 $\label{eq:c:windows} wicrosoft. NET assembly GAC_64 \ Newport. XPS. CommandInterface \ v4.0_{1.0.0.0_9a267756cf640dcf}$

The sample application "XPSApplicationTest.exe" is located under:

C:\Users\myname\Documents\Newport\MotionControl\XPS-RL

3.0 How to Test .NET Drivers for XPS Controller?

Execute the XPS sample application "XPSApplicationTest.exe " from program files folder. This application uses **Newport.XPS.CommandInterface** assembly from GAC to test communication with XPS-RL controller.

Several instances of this applicati	on can be running in parallel.
-------------------------------------	--------------------------------

🔡 XPS Application - XPS-RL Unified No	t Released	<u>- 🗆 ×</u>
TCP IP IP address 192.168.33.229 IP Port 5001	Group / Positionner Name Status Group 1.Pos 12 Ready state from motion	Position -10.00
Connect Disconnect	Initialize Home Move to	-10
Connected to XPS	Kill	

4.0 How to Access .Net C# Project

The C# project is available under User folder to show how to create a C# project for XPS-RL controller.

C:*Users**myname**Documents**Newport**MotionControl**XPS-RL**CSharpProject*\

CSharpProject					
🕞 ◯ マ 📔 ▼ Libraries ▼ Documents ▼ Newport	🚱 ◯ ∽ 🖟 × Libraries × Documents × Newport × MotionControl × XPS-RL × CSharpProject ×				
File Edit View Tools Help					
Organize ▼ Share with ▼ Burn New folder					
☆ Favorites ↓ Downloads	Documents library CSharpProject				
E Desktop	Name -	Date modified	Туре	Size	
Deskop	CSharpCodeSources.zip	05/01/2016 12:30	ZIP File	2 077 KB	
🧮 Desktop					
🥽 Libraries					
Documents					

This project is the one that has been used to create the provided XPS sample application.

5.0 How to Use XPS .NET Assembly from a Visual Studio C# Project?

Please refer to Microsoft Visual Studio web site to get more information to help you in your development (https://www.visualstudio.com/).

5.1 Add Reference to .NET Assembly

Add Newport.XPS.CommandInterface.dll in References of your project:

x64:

 $\label{eq:c:windows} assembly \ GAC_64 \ Newport. XPS. Command Interface \ 1.0.0.0 \ 9a267756cf \ 640dcf$

x86:

C:\Windows\assembly\GAC_32\Newport.XPS.CommandInterface\1.0.0.0_9a267756cf 640dcf



5.2 C# Code Sources

5.2.1 C# Header

using CommandInterfaceXPS; // Newport.XPS.CommandInterface .NET Assembly access

5.2.2 Add a Variable to Declare an "XPS" Object

CommandInterfaceXPS.XPS m_xpsInterface = null;

5.2.3 Create an Instance of "XPS" Object

m_xpsInterface = new CommandInterfaceXPS.XPS();

if (m_xpsInterface != null)

...

5.2.4 Open XPS Connection

if (m_xpsInterface != null)

int returnValue = m_xpsInterface.OpenInstrument(m_IPAddress, m_IPPort, DEFAULT_TIMEOUT);

5.2.5 Call "XPS" Functions

if (m_xpsInterface != null)

{

string XPSVersion = string.Empty;

string errorString = string.Empty;

int result = m_xpsInterface.FirmwareVersionGet(out XPSVersion, out
errorString);

if (result == CommandInterfaceXPS.XPS.FAILURE)

...

5.2.6 Close XPS Connection

if (m_xpsInterface != null)

m_xpsInterface.CloseInstrument();

6.0 How to use XPS .NET Assembly from a LabVIEW project?

Please refer to National Instruments web site to get more information to help you in your development (http://www.ni.com/labview/).

6.1 Add Reference to .NET Assembly

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

Select .NET Constructor
Assembly
Newport.XPS.CommandInterface(1.0.0.0)
Objects
CommandInterfaceXPS XPS Newport.Communication.TCPIP Newport.XPS
Constructors XPS()
OK Cancel Help

6.2 LabVIEW Code Sources

The instance of "XPS" object is created after configuration of .Net Constructor Node:



Open XPS connection (Use a .Net Invoke Node to select the XPS method "OpenInstrument"):

	h → XPS	
	OpenInstrument •	
192.168.33.229 ~~	 address 	
5001	 port 	
10000	 timeout 	

Call "XPS" functions (Use a .Net Invoke Node to select a XPS method):

 2 ·· → X	PS N	F
FirmwareVe	ersionGet	
Versi	on 🔹	
errstr	ing 🔹	

Close XPS connection (Use a **.Net Invoke Node** to select the XPS method "CloseInstrument"):



Close .NET Reference:



7.0 How to Use XPS .NET Assembly Under IronPython?

Please refer to IronPython web site to get more information to help you in your development (http://ironpython.net/).

7.1 Add Reference to .NET Assembly

Add Newport.XPS.CommandInterface.dll in References of your script:

x64:

import sys

sys.path.append(r'C:\Windows\Microsoft.NET\assembly\GAC_64\Newport.XPS.Comma ndInterface\v4.0 1.0.0.0 9a267756cf640dcf')

x86:

import sys

 $sys.path.append(r'C:\Windows\Microsoft.NET\assembly\GAC_32\Newport.XPS.CommandInterface\v4.0_1.0.0.0_9a267756cf640dcf')$

7.2 IronPython Code Source

7.2.1 IronPython Header

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

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

7.2.2 Create an Instance

Create XPS interface
myXPS = XPS()

7.2.3 Open XPS Connection

def XPS_Open (address, port):
Create XPS interface
myXPS = XPS()

Open a socket timeout = 1000 result = myXPS.OpenInstrument(address, port, timeout) if result == 0 : print 'Open ', address, ":", port, " => Successful" else: print 'Open ', address, ":", port, " => failure ", result

return myXPS

7.2.4 Call XPS Functions

def XPS_GetControllerVersion (myXPS, flag):
result, version, errString = myXPS.FirmwareVersionGet()
if flag == 1:
 if result == 0 :
 print 'XPS firmware version => ', version
 else:
 print 'FirmwareVersionGet Error => ',errString
return result, version

def XPS_GetControllerState (myXPS, flag):
result, state, errString = myXPS.ControllerStatusGet()
if flag == 1:
 if result == 0 :
 print 'XPS controller state => ', state
 else:
 print 'ControllerStatusGet Error => ',errString

return result, state

7.2.5 Close XPS Connection

def XPS_Close(myXPS):
myXPS.CloseInstrument()

8.0 How to Use XPS .NET Assembly Under Matlab?

Please refer to MathWorks web site to get more information to help you in your development (http://www.mathworks.com/products/matlab/).

8.1 Add Reference to .NET Assembly

% Make the assembly visible from Matlab asmInfo = NET.addAssembly('Newport.XPS.CommandInterface')

8.2 Matlab Code Source

8.2.1 Create an Instance

% Make the instantiation myxps=CommandInterfaceXPS.XPS();

8.2.2 Open XPS Connection

% Connect to the XPS controller code=myxps.OpenInstrument('192.168.254.254',5001,1000);

8.2.3 Call XPS Functions

% Use API's [code Version]=myxps.FirmwareVersionGet [code]=myxps.GroupKill('Group1') [code]=myxps.GroupInitialize('Group1') [code]=myxps.GroupHomeSearch('Group1')

8.2.4 Close XPS Connection

% Disconnect from the XPS controller code=myxps.CloseInstrument;

Service Form

Your Local Representative

Tel.:	
Fax:_	

Name:	Return authorization #:	
Company:	(Please obtain prior to return of item)	
Address:	Date:	
Country:	Phone Number:	
P.O. Number:	Fax Number:	
Item(s) Being Returned:		
Model#:	Serial #:	
Description:		
Reasons of return of goods (please list any specific	c problems):	

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

