



[articles](#) [quick answers](#) [discussions](#) [features](#)  
[community](#) [help](#)

Articles / Programming Languages / C#

Watch



[.NET2.0](#) [WinXP](#) [Win2003](#) [VS.NET2003](#) [VS2005](#) [C#2.0](#) [C#](#) [Windows](#)  
[.NETCF](#) [.NET](#) [Visual-Studio](#)

# Using XNA to Access an Xbox 360 Joystick



**Joel Ivory Johnson**

Rate me: ★★★★★ 4.76/5 (19 votes)

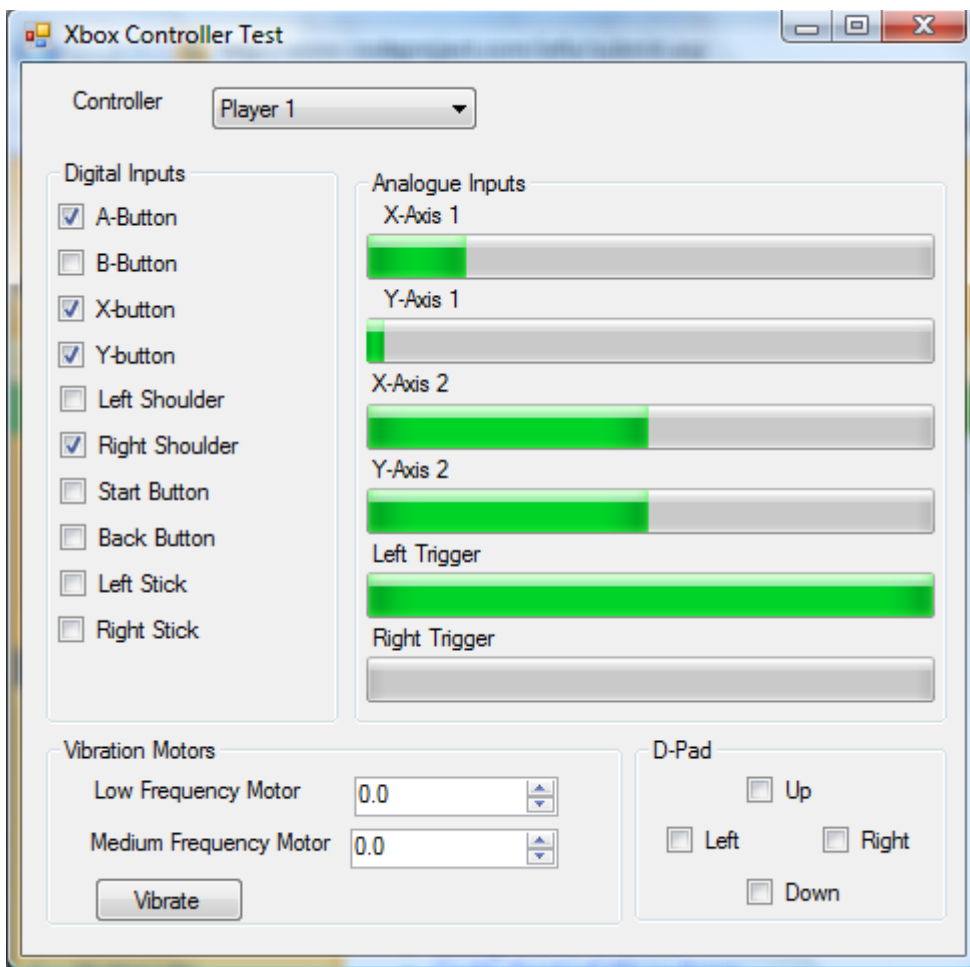
12 Nov 2012 [CPOL](#) 5 min read

196.7K 6.8K 37 33

This article demonstrates how to use the Microsoft XNA Framework to read the state of an Xbox 360 controller

**Download source files - 13.74 KB**

**Download demo project - 216.95 KB**



## Introduction

With the release of XNA, Microsoft has made available a platform for writing games on the PC and for the Xbox 360. XNA provides more simplified access to hardware than managed DirectX did. The Xbox 360 controllers are also fully supported by the XNA Framework. In this article, I'll be using Visual Studio 2005 and the XNA Framework to create a demonstration form that will display the full state of Xbox 360 controllers connected to the system.

**Update 11 November 2012** - I've posted another article that makes use of XInput instead of Xna to access the same functionality. XInput gives access to additional information (such as information on the battery) but the version of the code in the other articles requires Windows 8. You can find the article [here](#).

## Prerequisites

This code example uses assemblies from the XNA Framework and you will need to install the XNA redistributables to run the example program. You can download it from [here](#). In creating your own XNA project, remember to add an assembly reference to `Microsoft.Xna.Framework` and `Microsoft.Xna.Framework.Input`.

## XNA

The XNA Framework is based on a subset of the .NET Compact Framework and was designed to run on Windows XP SP2. At the time of this writing, Windows 2003 and Vista are not officially supported, although the Framework does operate on these systems. The Xbox 360 has an XNA game loader which provides a runtime environment for XNA based games.

Since XNA is based on a subset of the .NET Compact Framework, there is no support for the Windows Forms namespace or the network namespace. However, if you are targeting the Windows platform only, you can use `System.Windows.Forms` and `System.Net` and other namespaces as needed.

## Types of Inputs

The inputs on the controller are either analog or digital. The digital inputs will only have one of two states to indicate whether or not they are being pressed: `ButtonState.Pressed` or `ButtonState.Released`. The digital inputs on the controller are the buttons labelled A, B, X, Y, LB, RB, Back, Start, the buttons under both of the thumb sticks, and the D-Pad.

The state of the analog inputs is represented with a floating point number. The triggers on the controller will cause values between 0.0 (the trigger is not pressed) and 1.0 (the trigger is pressed all the way down) to be returned. The two thumb sticks return values between -1.0 and +1.0 for the x and y axis, where 0 is the center point for the axis.

The button in the center of the Xbox 360 controller is not accessible. It's only used by the Xbox operating system.

## Accessing the Controller

The state of the controller is accessed through the `Microsoft.Xna.Framework.Input.GamePad` class. This class only has three methods (beyond those inherited from `System.Object`). Those methods are `GetCapabilities`, `GetState`, and `SetVibration`. For this demonstration I will only use `GetState` and `SetVibration`. These methods take for their first argument a `PlayerIndex` value. This is used to specify the controller with which we will interact.

The `GetState` method returns a `GamePadState` struct. `GamePadState` contains the state of the controller in several sub-structs named `Buttons`, `DPad`, `IsConnected`, `ThumbSticks`, and `Triggers`. The `Button` and `DPad` structs have members named after the 10 buttons on the controller and the four directions on the `DPad` that will each be set to `ButtonState.Pressed` and `ButtonState.Released` (there is also an enumeration in the `System.Windows.Forms` namespace, you may need to disambiguate between the two if you are using the `System.Windows.Forms` with `Microsoft.Xna.Framework.Input`).

C#



```
//Setting or clearing a checkbox depending on the state of a button on the controller.
```

```
this.gamePadState = GamePad.GetState(this.playerIndex);  
this.buttonA.Checked = (this.gamePadState.Buttons.A == Input.ButtonState.Pressed);
```

The **Triggers** struct contains the members **Left** and **Right** that have floating point to represent how far down the trigger is pressed.

The **ThumbSticks** member has two members **Left** and **Right**. Each one of these is a **Vector2**, or a group of floating point X and Y values indicating the position of the joystick. A **Vector** with the values (X:0,Y:0) mean the joystick is in the center position, (X:-1,Y:0) means the joystick is to the far left position, (X:0, Y:1) means the controller is being pressed up, and so on. Note that you will never see the X and Y values in their most extreme state at the same time.

## The Two Vibration Motors

The Xbox 360 controllers have two motors for vibration. The left motor causes a slow vibration. The right motor causes a quick vibration. The motors can be turned on with the **SetVibration** method. This method takes a **PlayerIndex** and two floating point numbers between 0.0f and 1.0f to indicate how strong each motor should vibrate.

C#



```
//Turning the slow vibration motor on to 50% maximum  
GamePad.SetVibration(playerIndex, 0.5f, 0.0f);
```

## About the Program

The example program continually polls the Xbox 360 controller on a timed interval and updates the onscreen state. **Checkboxes** are used for the digital inputs. The analog inputs are represented with progress bars. The axis for the thumb sticks will be empty in the most negative position, half full in the center position, and full in the most positive position.

You can also turn the motors on using the numeric up/down inputs at the bottom of the UI. I've only allowed the motors to be on for a limited time and also turn them off when the program is terminating.

## Conclusion

You'll find that the XNA Framework greatly simplifies interaction with the Xbox 360 controllers. In my next article, I'll show how to render graphics using XNA and using an XNA input device to control onscreen objects.

## History

- 20th December, 2006 - Article created

## License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

Written By

**Joel Ivory Johnson**

Software Developer

 United States

I attended Southern Polytechnic State University and earned a Bachelors of Science in Computer Science and later returned to earn a Masters of Science in Software Engineering. I've largely developed solutions that are based on a mix of Microsoft technologies with open source technologies mixed in. I've got an interest in astronomy and you'll see that interest overflow into some of my code project articles from time to time.

[J2i.net](#)

Twitter:[@j2inet](#)

Instagram: [j2inet](#)



Watch

## Comments and Discussions

Add a Comment or Question

?
















Email Alerts





























































Search Comments

🔍

SpacingRelaxedLayoutNormalPer page25Update

FirstPrevNext

 <b>Not working</b> 	 <b>Soraiko</b>	<b>11-Sep-16 18:19</b>
 <a href="#">XNA is Discontinued, Use XInput</a> 	 Joel Ivory Johnson	20-Jan-17 2:33
 <b>My vote of 5</b> 	 <b>Member 11269090</b>	<b>12-May-15 16:52</b>
 <b>Silverlight</b> 	 <b>Mike C. Adams</b>	<b>25-Sep-12 20:37</b>
 <a href="#">Re: Silverlight</a> 	 Joel Ivory Johnson	27-Oct-12 7:39

 Re: Silverlight 	 Joel Ivory Johnson	12-Nov-12 0:50
 <b>Xbox Control</b> 	 CP-Ayyub	4-Aug-12 20:49
 Re: Xbox Control 	 Joel Ivory Johnson	12-Nov-12 0:42
 <b>Button state</b> 	 tom.95	7-Mar-12 21:56
 Re: Button state 	 Joel Ivory Johnson	27-Oct-12 7:40
 <b>WARNING: READ IF YOU ARE USING (OR PLAN TO RUN THIS ON) 64BIT MACHINE!!!! [modified]</b> 	 FocusedWolf	7-Jul-09 22:19
 Re: WARNING: READ IF YOU ARE USING (OR PLAN TO RUN THIS ON) 64BIT MACHINE!!!! 	 Joel Ivory Johnson	7-Jul-09 22:57
 Re: WARNING: READ IF YOU ARE USING (OR PLAN TO RUN THIS ON) 64BIT MACHINE!!!! 	 Joel Ivory Johnson	13-Jul-09 5:55
 Re: WARNING: READ IF YOU ARE USING (OR PLAN TO RUN THIS ON) 64BIT MACHINE!!!! [modified] 	 tom.95	7-Mar-12 19:47
 <b>How do I add VB code to it?</b> 	 cmilianxp	30-May-09 2:09
 <b>Error when trying to run sample program</b> 	 michaelry	9-Apr-09 4:16
 Re: Error when trying to run sample program 	 Joel Ivory Johnson	9-Apr-09 4:21
 Re: Error when trying to run sample program [modified] 	 michaelry	9-Apr-09 4:29
 <b>Could not load file or assembly 'Microsoft.Xna.Framework, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6d5c3888ef60e27d' or one of its dependencies.</b> 	 Joel Ivory Johnson	26-Nov-07 0:01
 Re: Could not load file or assembly 'Microsoft.Xna.Framework, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6d5c3888ef60e27d' or one of its dependencies. 	 Joel Ivory Johnson	26-Nov-07 3:58
 <b>Controller connect</b> 	 kaylee31	13-May-07 0:22
 Re: Controller connect 	 Joel Ivory Johnson	13-May-07 1:30
 <b>XNA with PS2 Controller</b> 	 Tony1986	21-Mar-07 7:30
 Re: XNA with PS2 Controller 	 Joel Ivory Johnson	21-Mar-07 19:50
 Re: XNA with PS2 Controller 	 Tony1986	22-Mar-07 0:45

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+ Left/Right to switch pages.