Classification	XPAC Development FAQ					No.	5-006-00
Author	Sean	Version	1.0.0	Date	2011/9/20	Page	1/3

How to use microsecond timer with real-time performance

Applied to:

Platform	*OS version*	*XPAC SDK version*	*XPACNetSDK version*
XP-8000-CE6	**V1.3.2.2 or later**	**V2.0.0.7 or later**	**V1.1.1.x or later**

In the X5-05 How-to document, we discussed how to use the Multimedia Timer to realize real-time operations. The Multimedia Timer can be used to adjust the thread priority to meet real-time requirements, but the timer units are only in milliseconds.
XPAC series controllers provide a Backplane timer that supports high resolution timers where the units are in microseconds. In addition, the thread priority of this timer also can be adjusted in order to meet the real-time requirements.

Backplane API functions:

(1) pac_SetBPTimerOut
(2) pac_SetBPTimer
(3) pac_KillBPTimer
(4) pac_SetBPTimerInterruptPriority

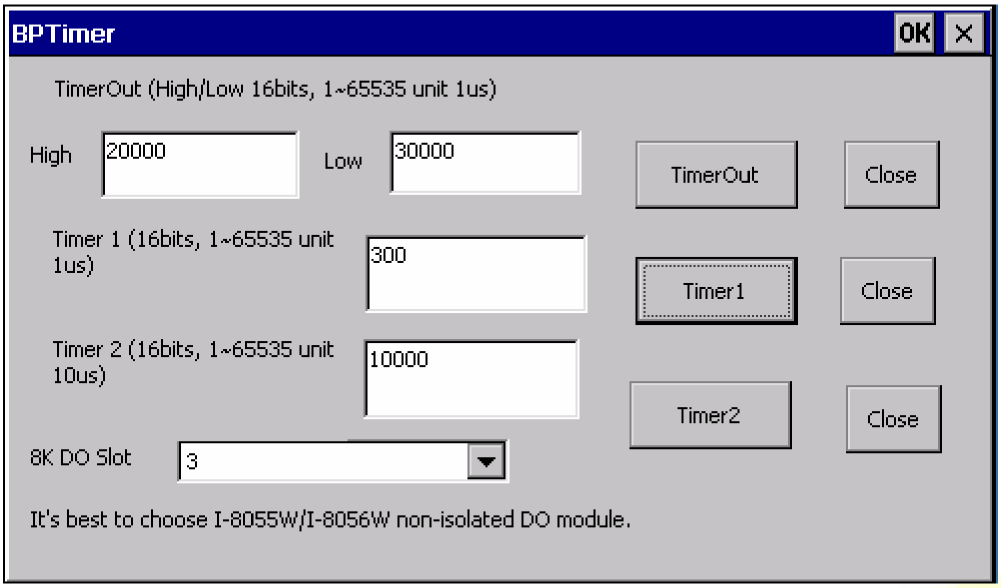
Refer to the XPAC Standard API manual at ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/document/sdk/
Or ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/document/sdk/

What is the input range of the priority for the Backplane Timer?
The input range can be from 8 to 256.

How to use Backplane Timer

1. VC++ demo

Use any 8K series DO module to output a square wave for testing in real time. Change the timer priority to enhance the real-time capabilities. Use an oscilloscope to monitor the square wave.



BPTimer

TimerOut (High/Low 16bits, 1~65535 unit 1us)

High 20000 Low 30000

Timer 1 (16bits, 1~65535 unit 1us) 300

Timer 2 (16bits, 1~65535 unit 10us) 10000

8K DO Slot 3

It's best to choose I-8055W/I-8056W non-isolated DO module.

2. C#/VB.NET demo

Writing the managed code using C# or VB operating in the .NET compact framework isn't recommended for real-time work. The features of the JIT compiler and the garbage collector of .NET compact framework will interfere with deterministic system behavior.

However, ICPDAS provides the XPacNET.dll file that uses the P/Invoke functionality to allow managed code to call the unmanaged native dynamic-link library entry points of the XPacSDK.dll file.

Refer to http://msdn.microsoft.com/en-us/library/ms836789.aspx for more details regarding the Real-time behavior of the .NET Compact Framework.

XPacNET.dll file must be put in the same folder with the executable file.

Location of demo download

C#:
http://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/c%23/standard/bptimer/

VC++:
http://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vc2008/standard/bptimer/

VB.net:
http://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vb.net/standard/bptimer/

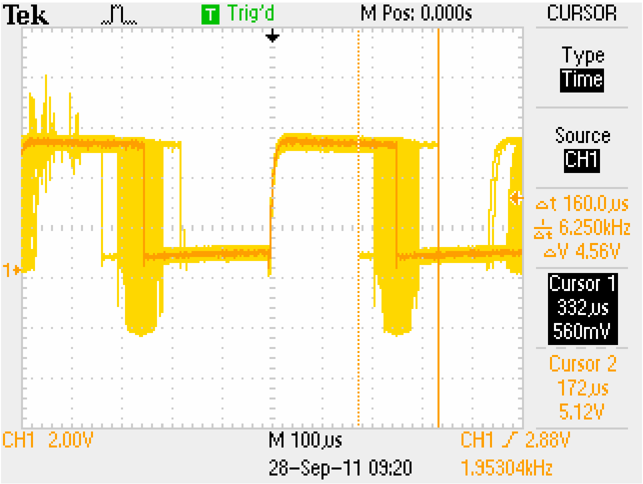
The Actual Test

1. Test using Backplane Timer1 calling the pac_SetBPTimer(1,…) function.
2. Run the remote display utility (CERDisp) and a specific program to set the CPU loading to 100% and use the persistence feature of the oscilloscope. This feature records and overwrites ALL traces for a total of many thousands of traces for 4 hours.

The Windows CE Remote Display (CERDisp) utility displays a Windows CE device screen on a remote desktop via an Ethernet network.

C# BP Timer1

pac_SetBPTimer(1, 150, ..) The period of the timer is 300µs (50% duty cycle)

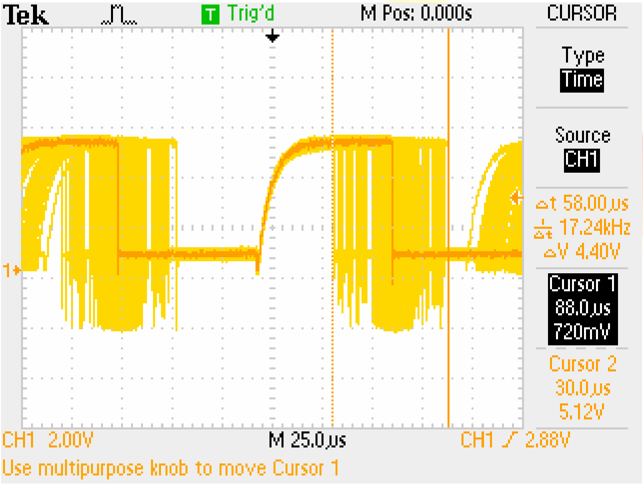


Jitter is result about 160µs.

According to the actual test, the time interval must be larger than or equal to 150µs in order to proceed with development using C# language.

VC BP Timer1

pac_SetBPTimer(1, 70, ..) The period of the timer is 140µs (50% duty cycle)



Jitter is result about 70µs.

According to the actual test, the time interval must be larger than or equal to 70µs in order to proceed with development using VC language.