Release Notes

SDR: Simplified Nov/Dec 2011

The Programs

The ZIP file contains three directories each with a different stage in the process. The first step is writing a simple loopback program (easier said than done). This program reads a small amount of data from the line in/microphone and sends it out to the line out/speaker. There is a small amount of delay between the two that is inherent in how the PC and DirectX handle sound out. The second step in the process is to write a simple DDS sine wave generator. It is a lot easier to use the sampling of the input to pace the output than to try to use the callback feature to get updates when the buffer is ready for more data. The last program marries the input capture with the DDS generation to produce an FM signal.

The Infrastructure

I went out to the Microsoft web site and downloaded the latest free version of Visual Studio 2010 Express. You can find this at www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express. You will also want to download the DirectX SDK toolkit to allow you to have better documentation and the DirectX debug tools. The SDK is DXSDK_Jun10.exe. You may need to enter that file name in the search tool on the Microsoft site. Install both products. Unzip the files in my ZIP file into the Projects directory under My Documents\Visual Studio 2010. That is the default location unless you specifically installed the work directory somewhere else.

You start the DirectX SDK tool from the SDK program group. Select DirectX Utilities/DirectX Control Panel. Select the Audio tab as shown below and increase the debug level to the maximum. This causes a lot of overhead as the system starts running a program, but it also tells you exactly what is going wrong if a DirectSound or DirectSoundCapture call fails for some reason.

Now you are ready to start Visual Studio. Start Visual Studio from the start menu and you will get an initial screen as shown on the second page following. Select Open Existing Project and browse for one of the projects from my ZIP file. You can operate the programs just as you would any other Windows project for debugging.

The Ugly Stuff

I ran into numerous problems along the way. The worst was my own doing where I divided instead of multiplying when computing buffer sizes. My brain skipped a cog and I was thinking

in terms of bits per second rather than words per second. It took a considerable amount of time until I found the SDK control panel to allow detailed debugging. I looked everywhere for books on DirectX but none exist, so you are stuck with the documentation you can find in the SDK and in Visual Studio. There is a very small amount of information on the web, but not much. The divide/multiply bug cost me at least two weeks given that I don't get to work on this in very concentrated time periods.

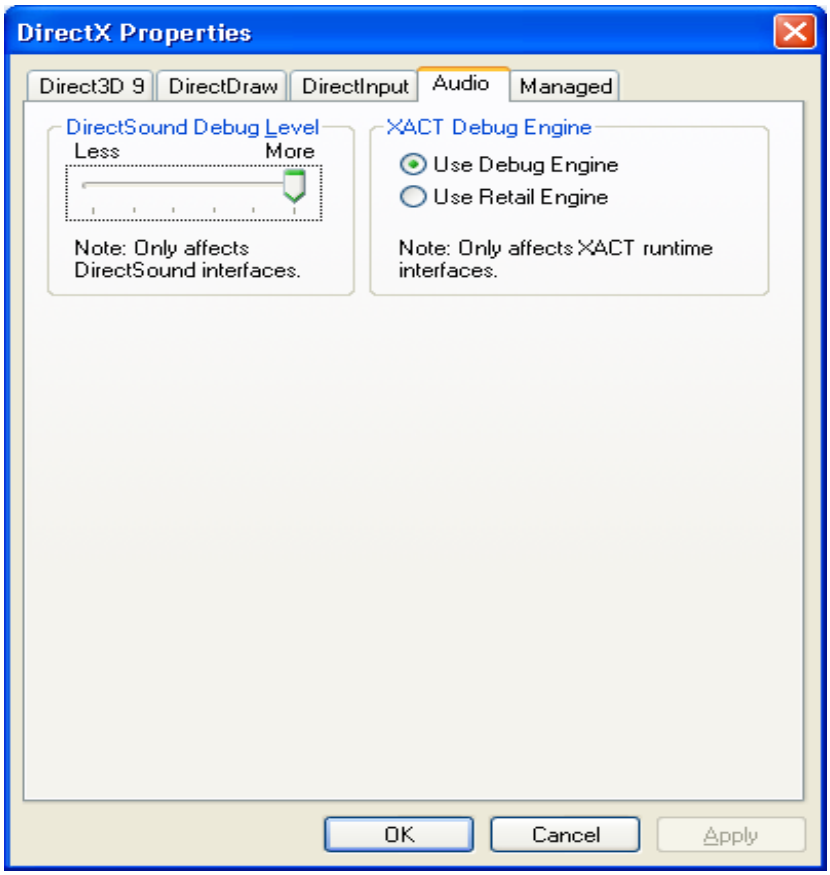
The Microsoft documentation frequently leads you to believe you can do alternate methods for input or output. Examples are using primary buffers or writing using the write cursor to keep your output position. If either of those work well, it wasn't obvious how to do it from the examples given.

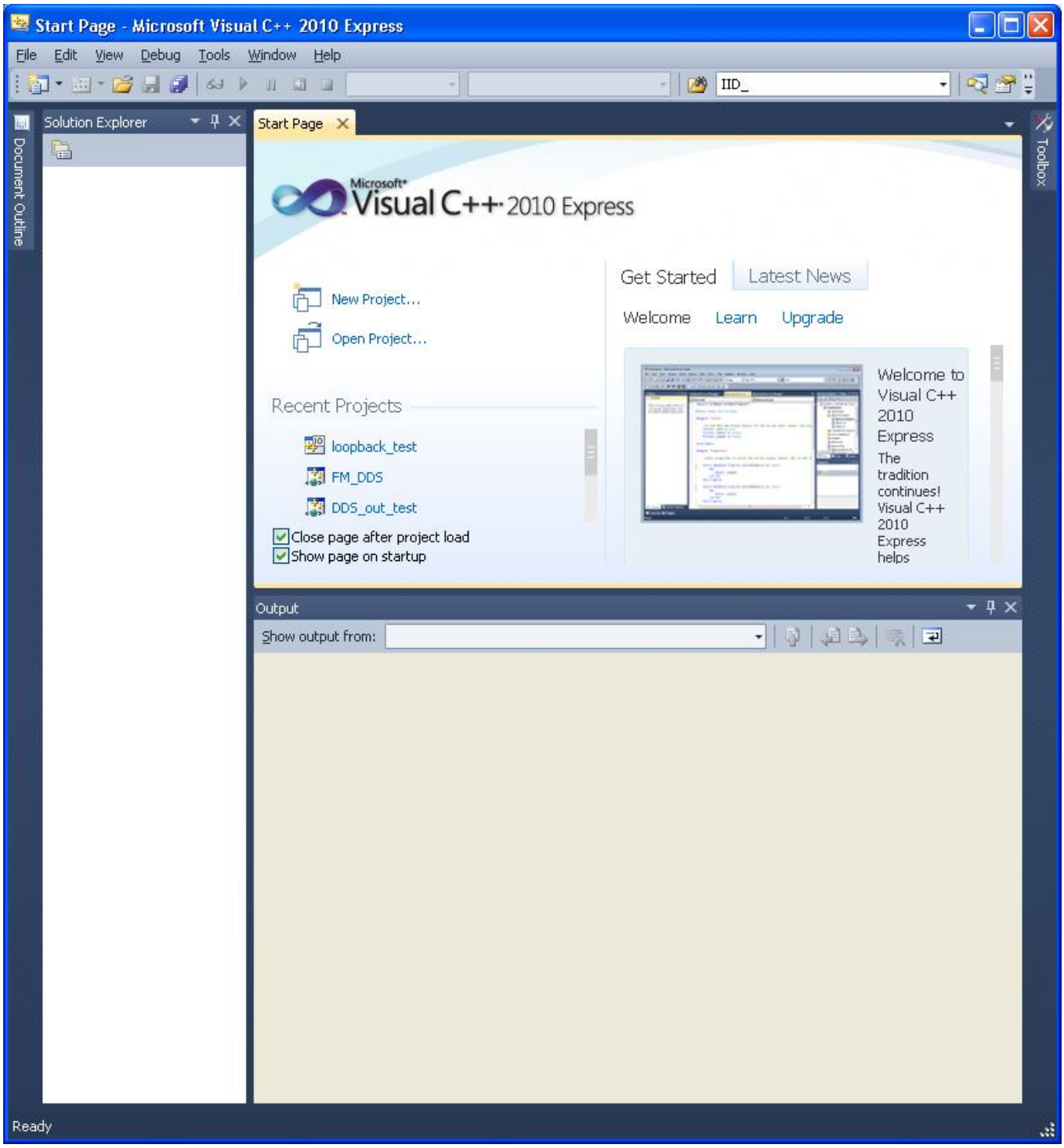
There is still a problem with keeping the input and output buffers in sync. There is a glitch that occurs occasionally as the output buffer wraps. I have looked, but cannot find the error in my logic. It causes a pop in both the DDS audio (try telling the DDS program to do 50 hz) and in the FM DDS. I found that setting the read size to 100 bytes rather than 1000 seemed to make quite a difference.

If you are a DirectSound expert, we could sure use some insight into how to make this platform more robust.

The next step is to work directly with the sound hardware on the TI card and abandon the Windows stuff until I can find more resources. I suspect that there is a better example of Windows operation by looking at PowerSDR, but I haven't had a chance to download and slog through the files to find what we need.

I have promised Larry a much bigger article for the March/April issue since I have the Christmas break to work on all of this.


DirectX Properties
Direct3D 9
DirectDraw
DirectInput
Audio
Managed
DirectSound Debug Level
Less
More
Note: Only affects DirectSound interfaces.
XACT Debug Engine
Use Debug Engine
Use Retail Engine
Note: Only affects XACT runtime interfaces.
OK
Cancel
Apply


Start Page - Microsoft Visual C++ 2010 Express
File Edit View Debug Tools Window Help
IID_
Solution Explorer
Start Page
Document Outline
Toolbox
Microsoft Visual C++ 2010 Express
New Project...
Open Project...
Recent Projects
loopback_test
FM_DDS
DDS_out_test
Close page after project load
Show page on startup
Get Started
Latest News
Welcome
Learn
Upgrade
Welcome to Visual C++ 2010 Express
The tradition continues! Visual C++ 2010 Express helps
Output
Show output from:
Ready