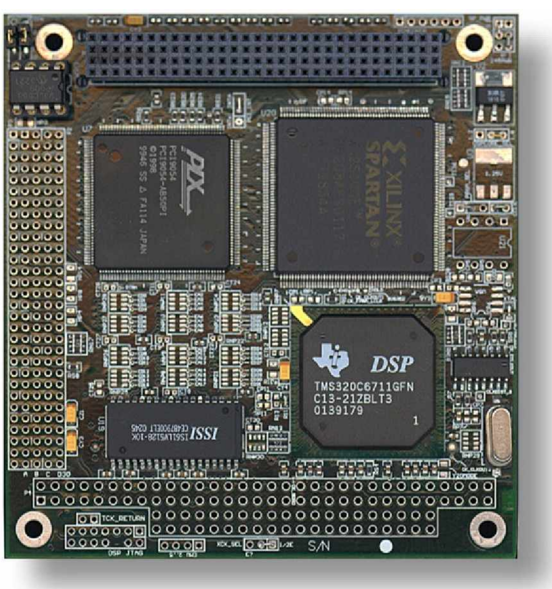
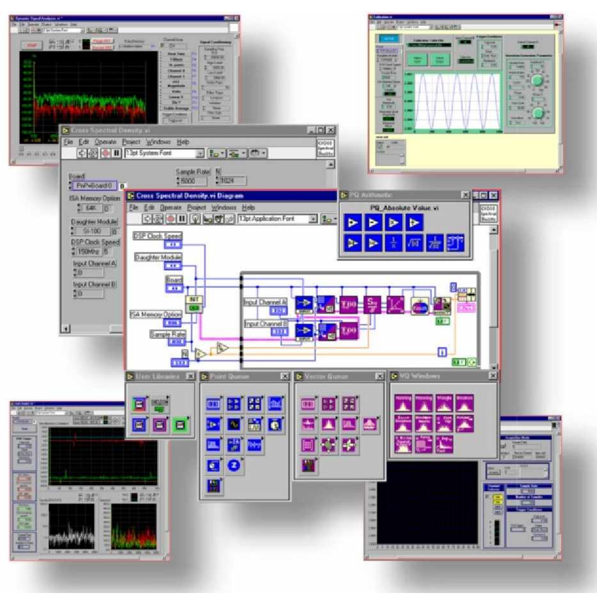
Easy Real-Time DSP Programming with LabVIEW and Visual Basic

For many measurement and control applications a PC nowadays can be a cost-efficient solution. PCs offer an excellent price-per-computation-horsepower ratio, there is a broad range of I/O interface boards available on the market, and they are supported by nice graphical software development packages like National Instruments' LabVIEW. However, a single CPU can only perform so much work, even if it's a multi-GHz Pentium. A comfortable, but resource-demanding operating system like Windows will not make the CPU's job any easier. Therefore, when a single host CPU must administrate various operating system tasks and applications, and run a complex Graphical User Environment, while at the same time also acquiring, controlling, analyzing and plotting real time data, throughput rates must invariably suffer. Furthermore, stringent real-time requirements of demanding control applications are likely impossible to satisfy.

One solution is to pass real-time work to a coprocessor, but even a power packed DSP does little good if a much slower host must baby sit each function; any speed gains disappear when the host CPU and the DSP must poll each other's status before proceeding to the next function. An alternative developed by Sheldon Instruments, involves a combination of hardware and software: It loads all acquisition, control, and analysis functions on to a separate DSP carrier card that can run *without interference* while the host attends to other tasks. Better yet, one can even program DSP applications with two familiar products: either National Instruments' LabVIEW or Microsoft's Visual Basic.

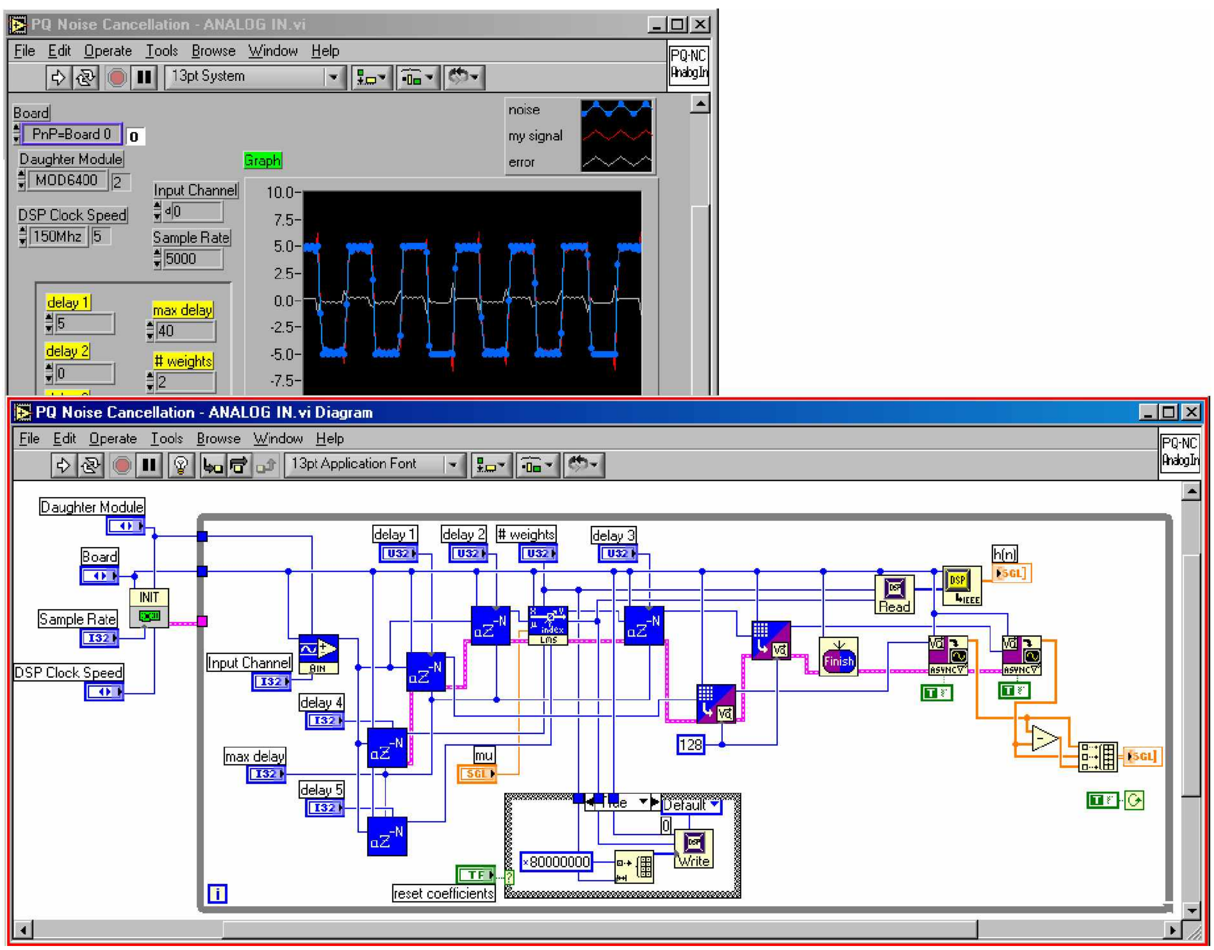




QuVIEW and QuBASE: An Evolution In User Friendly DSP Programming

The product line offering this is called QuX. It consists of QuVIEW and QuBASE, and its goal is simple: to allow a novice and expert alike to create the most sophisticated real-time DSP applications without spending a fortune or suffering the downtime associated with learning capricious, non-mainstream tools.

Since its introduction in 1991, QuX has evolved from a simple set of 'point by point' processing functions, targeted mainly for real time motion control, to a full fledged multi-tasking, real-time OS platform that now also includes the ability to easily program multiple threads of 'point by point' and 'block' processing routines into a DSP carrier card fitted with a multifunction I/O daughter module. The tool offers a very comprehensive set of arithmetic, classic time and frequency domain signal processing, vector, matrix, filter and I/O functions (see table 1), more than sufficient for even the most demanding measurement and control applications. The latest enhancement is a set of z-transform based functions, useful in a real-time control and digital filter applications, that allows to implement a polynomial directly in the z domain.



QuVIEW and QuBASE DSP Library Function List:

Arithmetic:
Addition, Subtraction, Multiplication, Division, Reciprocal, Square Root, Inverse Square Root, Negate, Scale aX+b, Complex Addition, Complex Subtraction, Complex Multiplication, Complex Division, Summation, Recursive Summation, Pi-mation.

Calculus:
Integration, Derivative, Slope.

Comparison:
Equal?, Not Equal ?, Greater Than?, Greater Than or Equal?, Less Than?, Less Than or Equal?, Maximum & Minimum, Trigger.

Constants:
Constant, 1D/2D Constant, Latch Constant

Counters, PWMs, Quadrature Encoders, & Timers:
EventCounter, Pulse In and Out.

Data Acquisition:
DSP Init (Start Acquisition), Finish Acquisition, Analog Input Channel, Analog Input Channel List, Analog Output Channel.

Digital Filters:
FIR Filter with FIR Coefficient Design, Cascade-IIR/IIR Filter with IIR Coefficient Design, LMS Adaptive Filter.

Digital I/O:
Digital Input/Output, Logical Gates (And/Nand/Or/Nor/Exor/Nexor/Not), Shift Right/Left, Bitwise Gates (And/Nand/Or/Nor/Exor/Nexor/Not).

Display:
Scope & Display, Packed Scope & Display, Waveform Scope & Playback, Get Vector from DSP, Put Vector to DSP.

Matrices:
Addition, Subtraction, Cross Product, Dot Product, Outer Product, Determinant, Inverse, Transpose, Solve Linear Equations.

Process Control:
PID Design, PID Controller, PID with Reset, Ramp, 4th order State Space Controller, Deadbeat Controller.

Signal Processing:
Complex/Real FFT, Inverse Complex/Real FFT, Power/Cross-Power Spectral Density, Convolution, Correlation, Maximax Shock Response with Smallwood Coefficient Design, Decimation.

Special Functions:
Index, Peak Detect, Average, Custom User Functions.

Structures:
Conditional Execute, Case.

Trig & Log:
X to Y, Sine, Cosine, Natural Log, Exponent base e, Log Base 10.

Vector Manipulation:
Resize, Subset, Mirror, Bit Reversal.

Waveform Generation:
Waveform Design, Waveform Synthesizer, Waveform/File Playback.

Windows:
Blackman, Blackman-Harris, Exact Blackman, Flat Top, Force, Hanning, Hamming, Triangle.

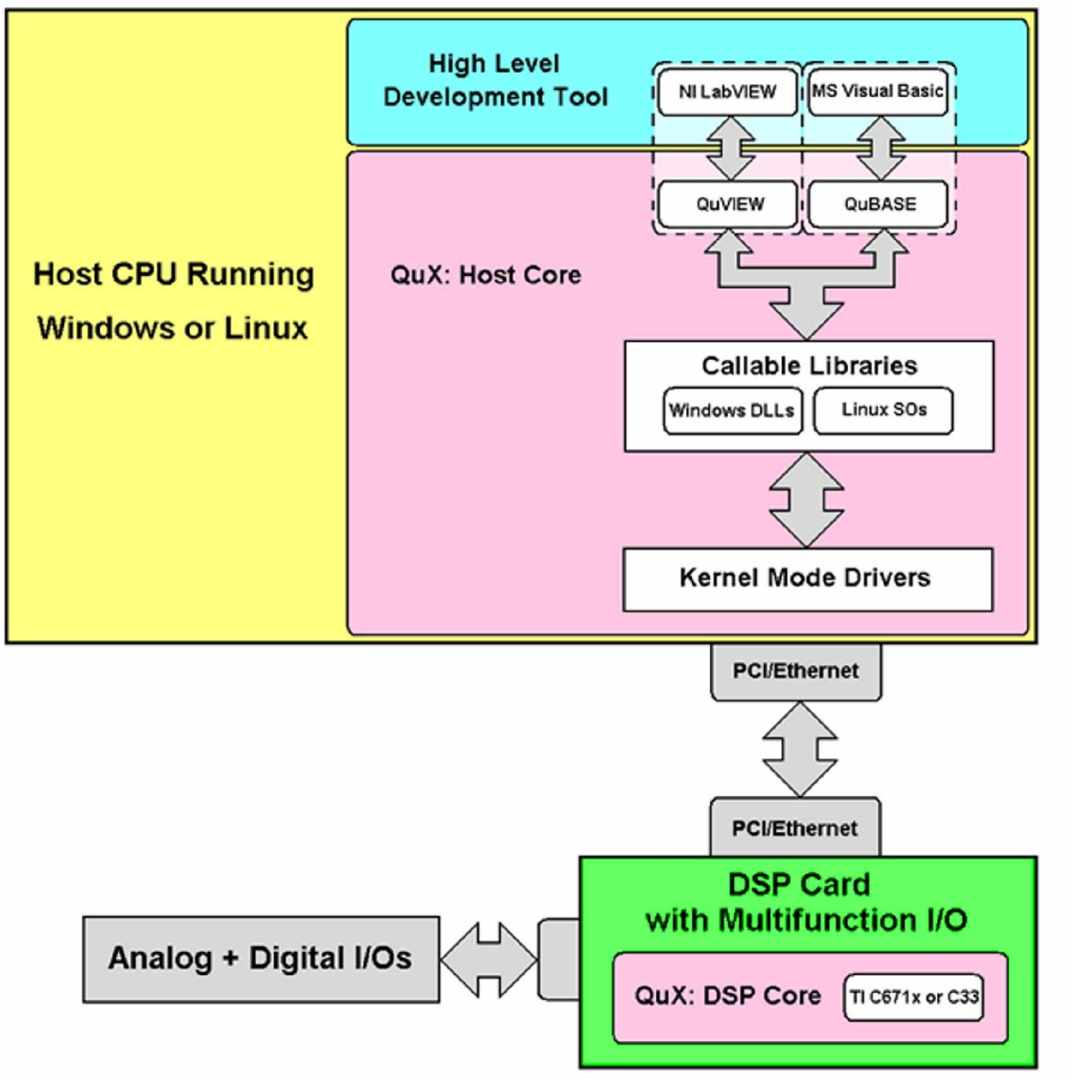
Z Transforms:
Feedback Node, Backward Z Delay, Forward Z Delay, Feedback Delay, Summing Node, Multiplying Node.

Structures:
Case Statement, Conditional Execute, While Loop.

Perfect Job Sharing: DSP and PC Software Components

QuX's core is comprised of two basic functional components: a host component and a DSP component. The interactions between these two components are synchronized with a dynamic scheme involving a shared memory pool.

The host component consists of:
1) low level kernel mode drivers to communicate with the DSP hardware,
2) dynamically callable libraries (DLLs for Windows, or shared objects – 'SO' - for Linux) that serve as a bridge between the low level drivers and the user interface, and
3) the user interface or high level development tool which serve as 'wrappers' or 'shells' around the callable libraries.



QuX offers two choices for the PC programming environment and user interface: National Instrument's LabVIEW for the 'icon' programming approach, and Visual Basic for the traditional text based programming approach. The QuX software for LabVIEW is called QuVIEW, the version for Visual Basic QuBASE. QuX is fully compatible to run on either Win9x/NT/2000/XP or Linux.

The DSP component consists of a single DSP program (COFF file), which itself comprises the finely tuned real-time OS, the complete function library (both point by point and block processing functions), along with all of the appropriate communications code to support data transfers with the host system.

No C Compiler, No Assembler, No DSP Experience Required

Suffice it to say that the QuX core has been meticulously optimized by Sheldon Instruments and is supplied as an ideal, compact turnkey development package: no vendor specific C compiler, assembler, or linker necessary; nor is it expedient to have had any DSP experience whatsoever. Simply put, QuVIEW and QuBASE based programs are extremely fast and efficient: depending upon the application, LabVIEW and Visual Basic may both be accelerated by as much as 100 times, even more if multiple DSPs are used!

Points, Vectors and Structures: You Make the Choice

Two distinct methods of data processing along with program flow control and structuring are implemented within the DSP, and hence the libraries are divided in such a way so as to make this distinction more obvious.

The first data processing method, categorized as "Point Queue", is better suited for real-time and continuous data processing in which data is processed on a 'point by point' (sample by sample) basis and a result is returned before the next point is to be processed. The most obvious type of application benefited by this paradigm would be motion control.

The second method, categorized as "Vector Queue", is better suited for processing blocks of data points in which data is processed after the entire block has been received. Classic dynamic signal analysis applications are an obvious fit.

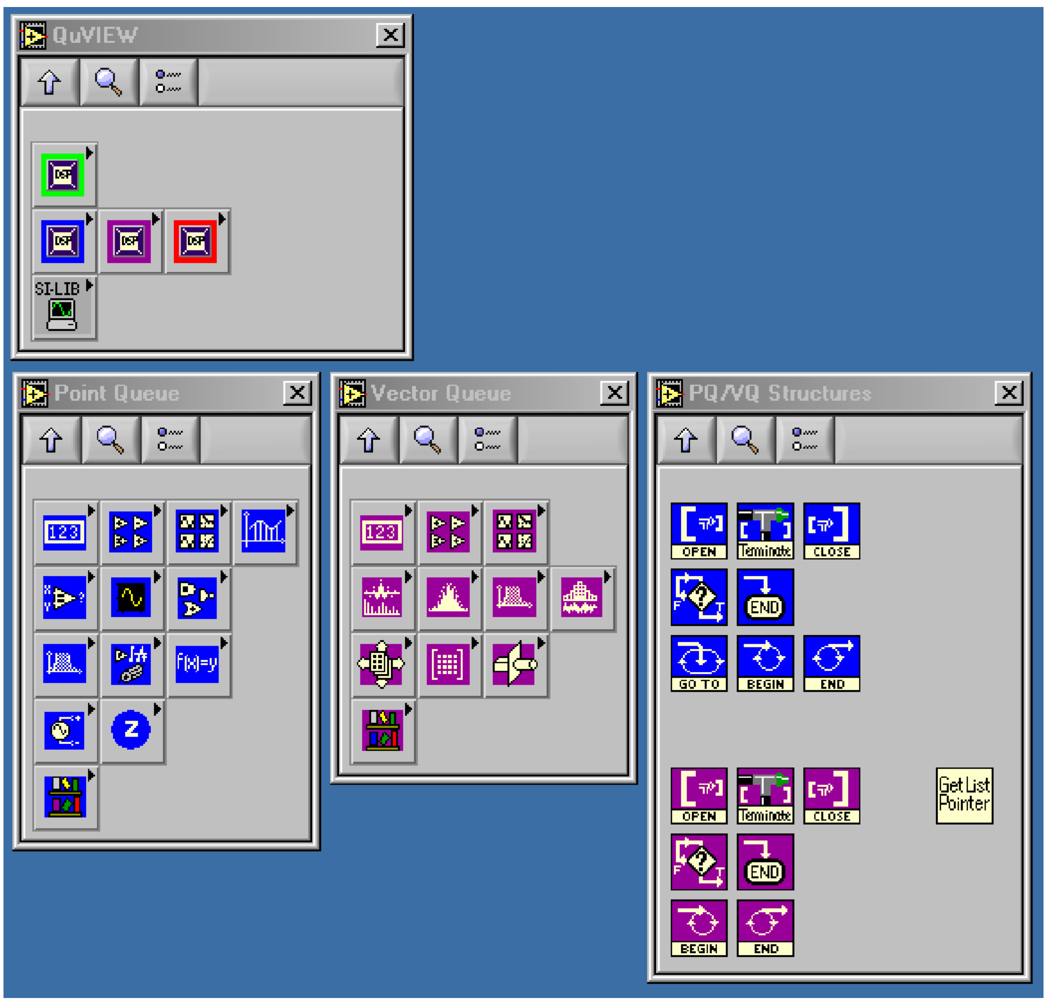
Lastly, program flow can be controlled with universal "Conditional" structures which are implemented with traditional 'Case Statement', 'Conditional Execute/Branch', and 'While Loop'.

To illustrate what can be accomplished with QuVIEW and QuBASE, consider a typical application example:

- Sample an arbitrary number of inputs
- Perform real time digital filters, arithmetic and calculus functions
- Process control with PIDs or State Space Algorithms
- Perform arbitrary waveform generations and synthesis
- Continuous stream to hard disk at rates up to 800khz
- Simultaneous display to screen of time and frequency data

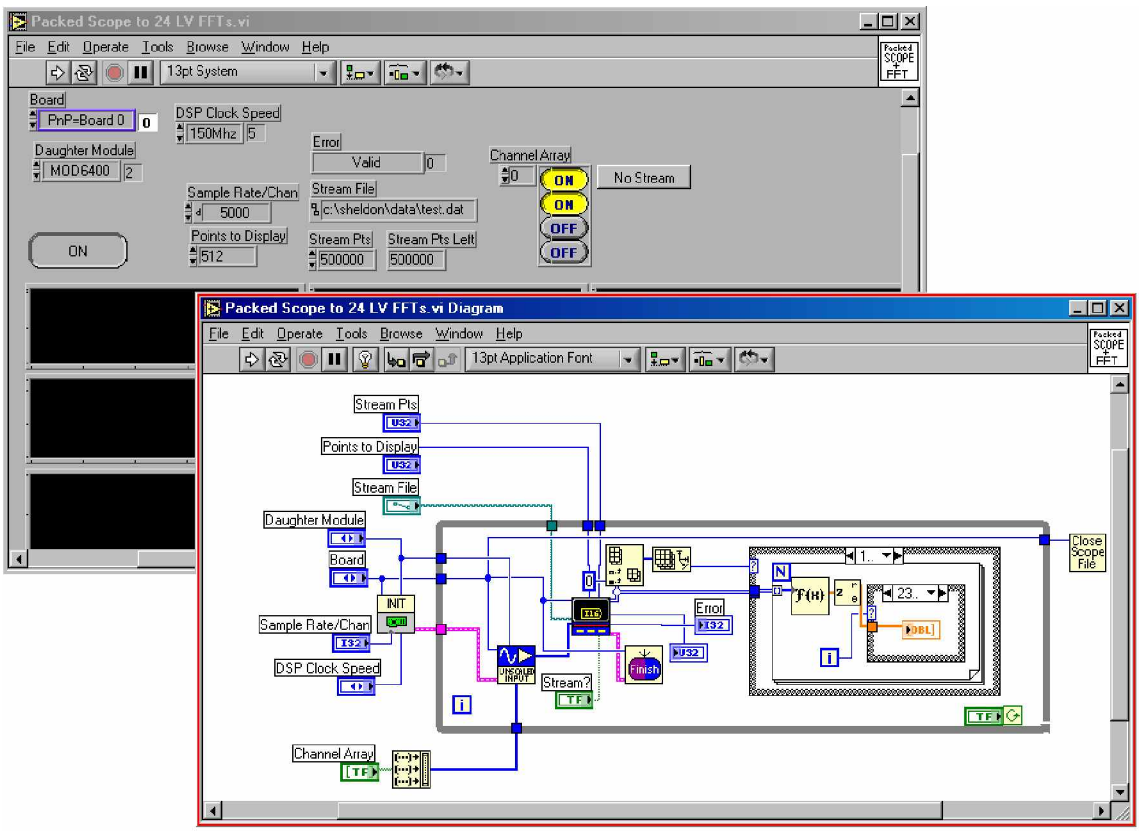
For data that needs to be exchanged between the host and DSP for display or updating parameters in real time, transfers seamlessly take place in the background via DMA.

On the other hand, because QuX runs as its own autonomous kernel on the DSP, it is not imperative that the host application be running at all. In fact, one can even shut down LabVIEW or Visual Basic and open an unrelated application while the acquisition/control app runs on the DSP. For true embedded applications, the SI-DSP carrier cards can be fitted with flash memory for stand-alone operation.



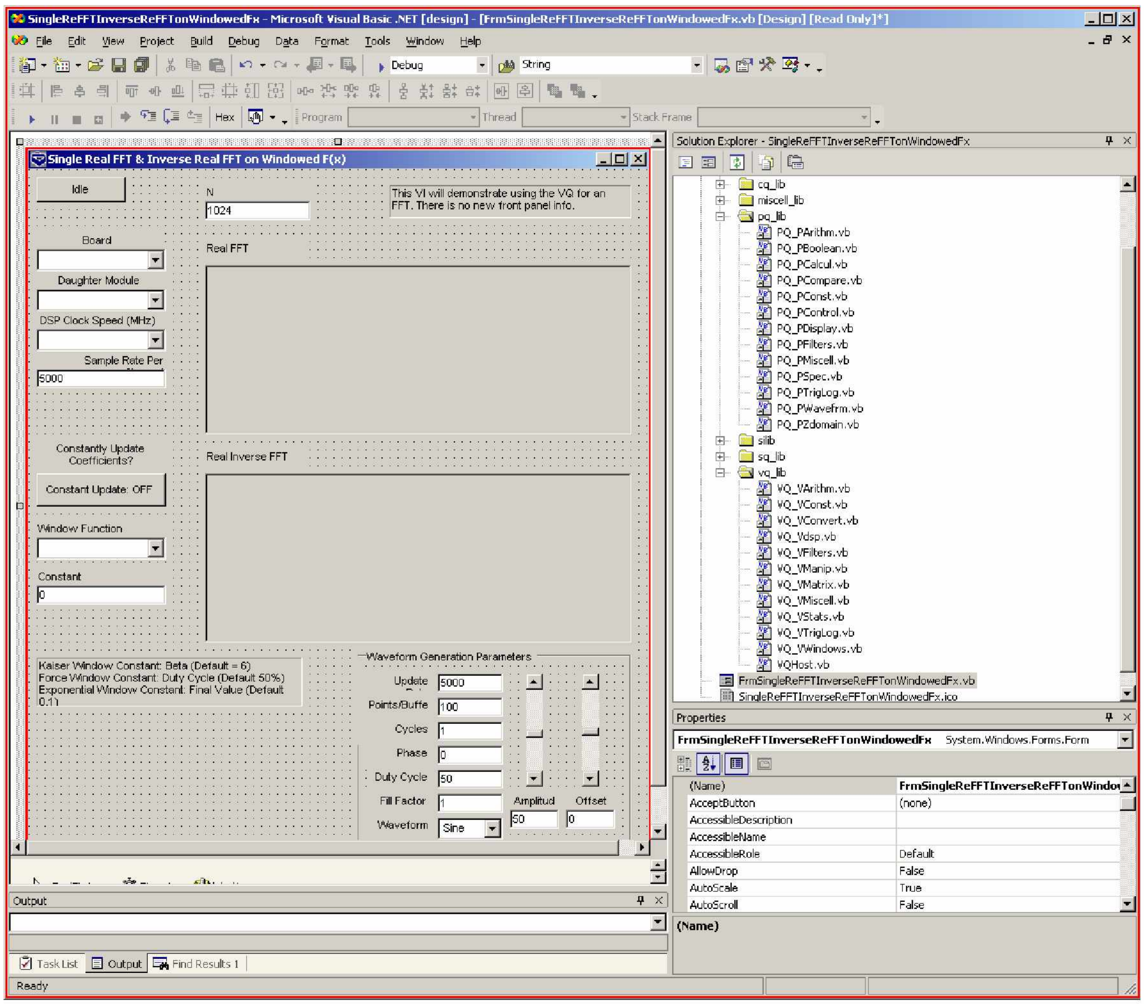
QuVIEW: The Real-Time Supercharger for LabVIEW

To allow the DSP carrier card to accelerate LabVIEW applications, Sheldon Instruments has rewritten roughly 75% of that program's math and analysis icons, as well as added others. The "Point Queue" and "Vector Queue" function libraries are nicely laid out as separate icon groups, just like those included by default. Users can mix a diagram with icons that use either the host CPU (native LabVIEW icons) or the DSP (QuVIEW icons).



QuBASE: The Real-Time Accelerator for Visual Basic

For budget conscious developers, Visual Basic's text based programming paradigm and the ability to create succinct 'executables' can be especially appealing, and yet with QuBASE there is no compromise in performance. The "Point Queue" and "Vector Queue" function libraries for QuBASE include exactly the same functionality as those for LabVIEW, and are equally neatly structured as simple inline function calls, clearly distinguished with unique prefixes for each function type. Users can mix library calls that execute on the host CPU or on the DSP (QuBASE calls).

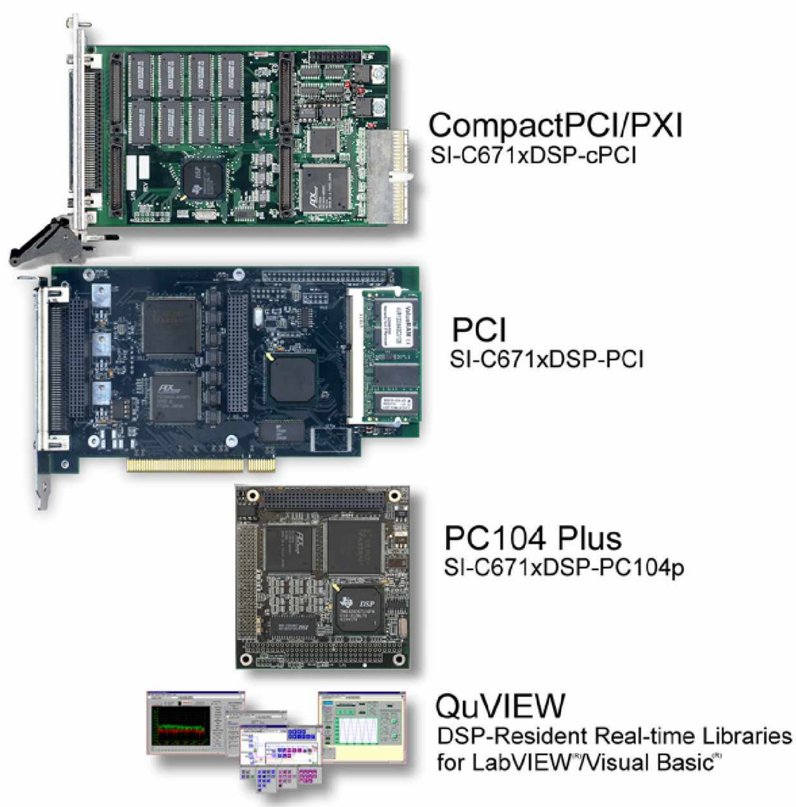


SI-DDK: The C and Assembly Language Purists Have Not Been Neglected

When purchased as a DSP evaluation board or for those developers that need a solid foundation for which to develop a custom application from scratch in C or assembly, Sheldon Instruments also provides all of the necessary code templates for both the DSP and the host sides, in a software suite named SI-DDK. Sample DSP source code to accompany Texas Instruments' development environment (Code Composer) illustrates various communication modes that make full use of the DSP's onchip resources. For the host side, Sheldon Instruments also includes the industry's most stable Windows and Linux kernel mode drivers - with the actual source code, a comprehensive API library, along with COFF file loader utilities.

The Hardware Behind The Software

Currently, QuX works in conjunction with all Sheldon Instruments' hardware which include a full array of SI-DSP carrier cards, along with a family of SI-MOD multifunction I/O modules. The SI-DSP carrier cards are based on TI's TMS320VC33 and TMS320VC3206713 floating point DSPs offering up to 1800 MFLOPS peak performance at 300 MHz clock rate, and are available in several popular PCI implementations, such as the standard desktop PCI, CompactPCI, PC104Plus, and PMC form factors. The latest product line addition will integrate remote access via an Ethernet connection.



The SI-MOD multifunction I/O cards conveniently fit on to any one of the DSP carrier cards to form a single integrated card solution. Their myriad of features include:

- 4 to 64 analog input channels with gain amps of 1-1000
- up to 16 analog outputs
- 16 bit resolution
- sampling ranging from 250 khz to an additive 1Mhz
- 36 digital IOs
- two sets of PWM outputs, pulse/frequency measurement inputs, and quadrature encoders
- flexible onboard clock management which includes 3 direct digital synthesizers and 4 event counters

Finally, No Budget Worries

QuX (QuVIEW and QuBASE) and the SI-DDK are automatically included for free with all purchases of Sheldon Instruments hardware, without any restrictive royalties or licenses. For pricing that starts at approximately US-$ 875, you can be up and running in no time!

For more information, please contact Sheldon Instruments, www.sheldoninst.com or call (619) 282-6700.

Sheldon Instruments
10393 San Diego Mission Rd. - Suite 202
San Diego, CA. 92108-2176
Tel: (619) 282-6700
Fax: (619) 282-6710
E-mail: info@sheldoninst.com
Web: www.sheldoninst.com