

# Data Acquisition: I/O for Embedded Systems

## Overview

Embedded design can describe the design of virtually any electronically controlled device or machine, from very high-volume consumer electronics devices to particle accelerators used in large physics labs. The requirements for an mp3 player and a particle accelerator are obviously different and, thus, rely on different architectures. Most systems fall somewhere between these two extremes and incorporate commercially available hardware and software to help balance component costs with development costs.

This is where a company specializing in high-quality I/O, such as National Instruments, can provide value to the embedded designer. This paper examines how most embedded computing platforms offer USB and/or PCI bus support, making it easy to take advantage of NI commercial off-the-shelf (COTS) hardware benefits for I/O in embedded systems.

[National Instruments OEM Solutions](#)

[National Instruments USB OEM Data Acquisition Products](#)

## Embedded Design

Embedded designers are concerned with choosing the right processors, OSs, development environments, and I/O for their applications. Their objective is to meet their application specifications while striking a balance with system size, cost, and time to market. They incur costs, which are very much volume-driven, in development, time to market, and system components, which includes hardware and software costs.

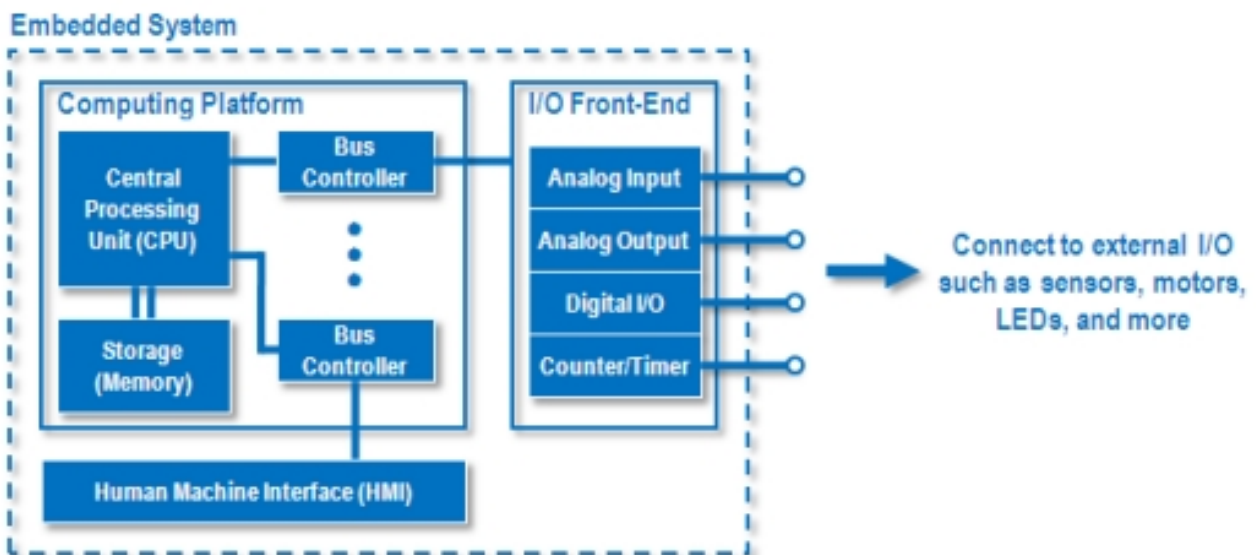


Figure 1. Components of a Typical Embedded System

There is not a universal systematic approach for developing an embedded system. At a high level, the application specifications dictate system software and I/O requirements, which in turn influence the selection of processor, peripheral hardware, and OS. Although the software and hardware component considerations are common across all applications, the importance attributed to each component varies widely. For example, tight I/O timing might not be important for a greenhouse temperature monitoring system but is imperative for an engine controller. And I/O quality might not be important for a toy robot but is crucial for machine condition monitoring. Nonetheless, key elements such

as OS compatibility, software development, embedded platform, bus availability, and cost should be universally considered. The flexibility of a COTS OEM I/O supplier can considerably simplify the design process by optimizing these key elements and reduce time to market.

## Custom versus COTS I/O

The custom design of an I/O front end requires a large engineering investment and could be attractive if designers are producing embedded systems at very high quantities. However, most companies do not have the required in-house expertise to produce world-class I/O, and they do not produce products at high mass-market volumes. NI OEM data acquisition (DAQ) devices include COTS products that deliver high-performance and high-quality I/O to embedded designers in a variety of form factors at attractive OEM prices.

NI OEM I/O products are compatible with a wide variety of computing platforms, buses, development environments, and OSs to help embedded designers easily adapt to specific design requirements. The identical form factor for each product line and common software API across all NI I/O also make it easy to increase channel count, add isolation, and more. This is a stark contrast to designing custom I/O front ends, which have longer time to market, are costly to develop and certify, are often compatible with only one system, and could cause future maintenance and spare parts issues.



*Figure 2. COTS NI PCI and USB OEM I/O Products*

For demanding machine and industrial control applications, NI provides preintegrated, off-the-shelf programmable automation controllers (PACs) for embedded designers. This new class of industrial controllers empowers embedded designers to skip most of the hardware development and focus on the application itself. Capabilities include advanced control, communication, data logging, and signal processing with rugged hardware performing logic, motion, process control, and vision. Learn more at [ni.com/pac](http://ni.com/pac).

## Software I/O Considerations

The selection of the appropriate software development environment and OS to meet specific application I/O demands is based on a number of factors, including processor, bus, and human-computer interface support. Computing performance and driver support also influence software selection, while the basic requirements for cost, time to market, and size still apply.

Most embedded designers choose text-based programming languages such as C and C++ for their versatility and widespread adoption. However, this approach requires in-depth knowledge on the many caveats and unique considerations related to the custom integration of OSs, drivers, I/O, and human machine interfaces (HMIs) along with extensive development time. For highly complex projects requiring short time to market, graphical system design can

help abstract the various software layers and hardware components so embedded designers can focus on their applications. At the center of graphical system design is NI LabVIEW, a graphical programming language for both conventional and real-time OSs, along with field-programmable gate arrays (FPGAs). Graphical system design simplifies the integration of hardware and software for embedded systems.

## Operating System

Each OS has specific strengths tailored for various applications. For example, QNX is a rugged, real-time OS used for mission-critical applications. It has an architecture where every driver, application, protocol stack, and file system runs outside of the kernel, in a memory-protected space. Thus any component can fail and be automatically restarted without affecting other components of the kernel.

Windows XP Embedded on the other hand is essentially a componentized version of Windows XP. Designers can use it to assemble an OS image that meets their footprint requirements while maintaining OS dependencies. Standard OSs such as Windows XP and Mac OS X are not deterministic, meaning that instructions cannot be guaranteed to execute within a specified amount of time.

Although a real-time OS is not always necessary, designers need to carefully consider timing to ensure proper execution of software and I/O.

## Software Development

Software development is especially time-consuming for custom I/O because designers need to write an exclusively compatible driver. Any changes in the hardware or software environment, such as channel-count expansion, forces a redesign of the custom software. NI OEM products remove these impediments by providing drivers that are compatible with a wide range of OSs and software development environments. For OEM I/O products, NI offers three levels of driver software to meet individual application requirements at no additional cost: NI-DAQmx, NI-DAQmx Base, and the NI Measurement Hardware Driver Development Kit (MHDDK). Each has a different footprint, feature, development environment, and OS support, empowering developers with more flexibility to meet their software requirements.

Embedded designers can take advantage of NI OEM products with LabVIEW and most text-based programming languages such as ANSI C/C++, C#, Visual Basic .NET, and Visual Basic 6.0. NI also provides an extensive support network, including thousands of example programs, to help reduce development time.

## National Instruments Driver Software

NI offers three different software drivers to accommodate the programming language, OS, and processor requirements of an embedded system.

### NI-DAQmx

NI-DAQmx is a high-performance, feature-rich multithreaded driver for Windows and LabVIEW Real-Time. It provides measurement services including the Measurement &#38; Automation Explorer (MAX) and the DAQ Assistant for text-based and LabVIEW code generation. With NI-DAQmx, a componentized and extensible driver software package, users can choose between five run-time options when deploying an NI-DAQmx application onto an embedded system. These range from approximately 167 MB for deployment on embedded platforms to more than 600 MB for the development package.

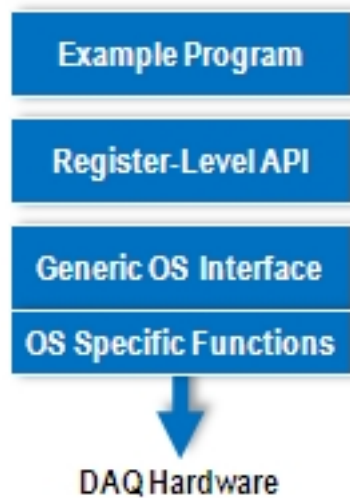
NI-DAQmx supports LabVIEW, ANSI C/C++, C#, Visual Basic .NET, and Visual Basic 6.0. For NI M Series plug-in I/O boards on PCI, designers can use LabVIEW graphical code to develop and deploy applications to all NI real-time hardware targets including standard PC platforms.

### NI-DAQmx Base

NI-DAQmx Base extends a subset of NI-DAQmx functionality to Mac OS X, Linux®, and Windows Mobile for Pocket PC users. National Instruments built NI-DAQmx Base with the MHDDK, providing an agile driver with a small footprint. NI-DAQmx Base supports ANSI C and LabVIEW, and typical deployment sizes are between approximately 75 and 100 MB.

## MHDDK

The MHDDK offers the highest level of customization for a wide range of OSs with a very small footprint. It features development tools and a register-level programming interface for system designers to develop custom drivers with NI measurement hardware. Although the MHDDK requires more advanced embedded development skills, it delivers more control and is often necessary due to hardware performance or OS constraints.



*Figure 3. Software Layers for the NI Measurement Hardware DDK*

The following OS-specific bus interface components are available online with the MHDDK: Windows XP/2000/Me/9x/NT, WDM (Win32/Win64), Linux, LabVIEW Real-Time, Windows CE/Pocket PC (PCMCIA only), RTX, QNX Neutrino, and TenAsys INtime. Embedded designers can easily work with other OSs by using the provided basic framework to create their own OS-specific interface components.

The MHDDK, which supports C and C++ development languages, can be deployed with less than 10 MB of storage space.

## Hardware I/O Considerations

Hardware I/O considerations for embedded applications include bus availability such as PCI and USB, OS, overall system performance, and development environment. The overarching requirements for cost, time to market, and size still apply and often dictate the hardware platform selection. Hardware platforms are essentially variations of a computer system designed for specific applications.

## Embedded Platforms

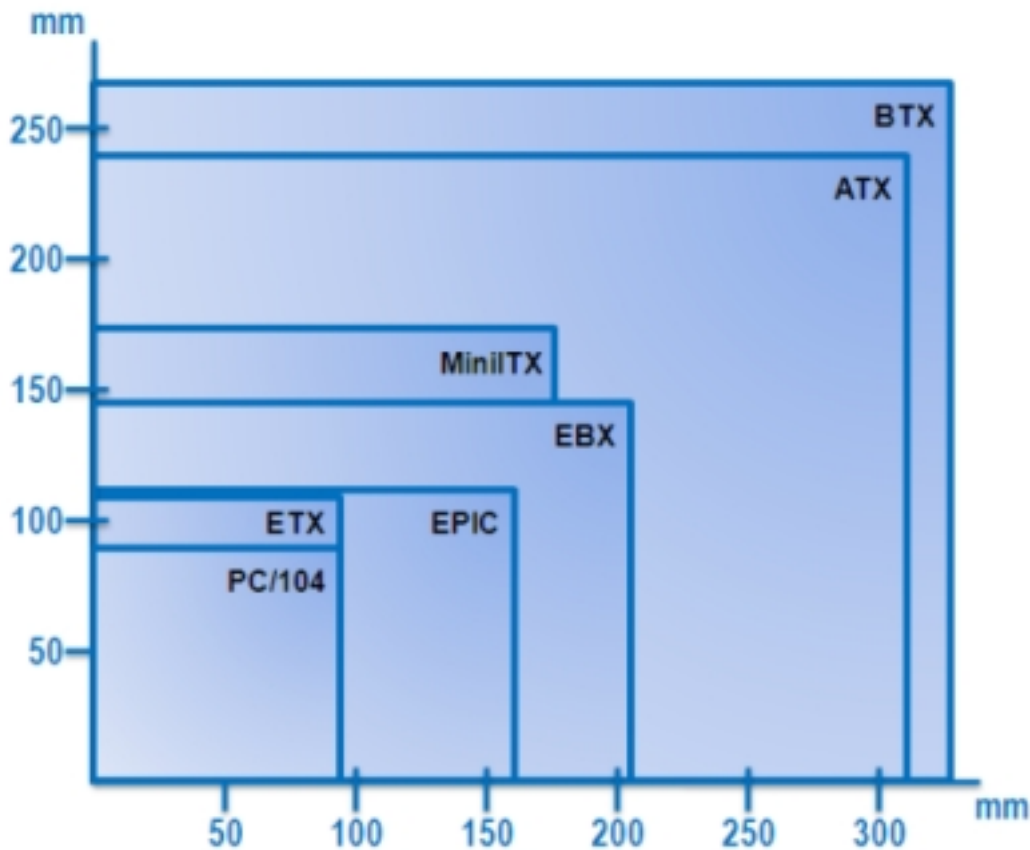


Figure 4. Embedded Platform Size Comparison Chart

The two distinguishing characteristics of an embedded platform are its size and adoption in the small form factor market. The most popular platforms are typically lower in cost and offer the most flexibility in processor, bus, and memory selection. Size dictates how much effort it takes to squeeze in all the various components of an embedded device, and a bigger space allows for easier component layouts and better heat dissipation. For the same performance, a smaller form factor could dictate a higher price.

The small and very popular PC/104 standard provides both an I/O expansion standard and a single-board computer (SBC) form factor. Modern PC/104 devices all have PCI bus connectors and a USB controller for I/O expansion.

Slightly bigger than the PC/104 is the ETX/COM standard, which can require extensive development effort to create the mating board necessary to interface with I/O. Other platforms include EBX, which is lower in cost than PC/104 due to its larger size; ATX and BTX, which are common technologies for motherboards on desktop computers; and finally MiniITX, which has a low-power-consumption architecture. These are popular in commercial applications. Finally, EPIC and EBX form factors target small and rugged embedded systems in industrial, military, and medical applications.

Processor selection is closely tied to power consumption, heat dissipation, and software support. In fact, a processor platform's success is highly dependent on the tools provided for its software development, and each embedded platform supports a select set of processors. One universal feature is bus connectivity, and embedded designers can take advantage of this for low-cost high-performance I/O.

## COTS I/O

As mentioned earlier, custom I/O has a longer time to market and could be exclusively compatible with one system, making any design change very difficult. In addition to an increased software development time, other future hardware issues such as maintenance, certification, scalability, and component replacement could plague the forthcoming system.

COTS I/O from SBC and embedded platform suppliers can alleviate some of the difficulties associated with custom I/O, but these suppliers are typically not specialized in signal analysis, processing, high-performance, and high-accuracy measurements.

Most embedded computing platforms provide USB and/or PCI bus support, making it very easy to take advantage of world-class NI COTS hardware for I/O in embedded systems. The choice of bus technology is critical for meeting the hardware I/O requirements of an embedded system.

### Bus Considerations

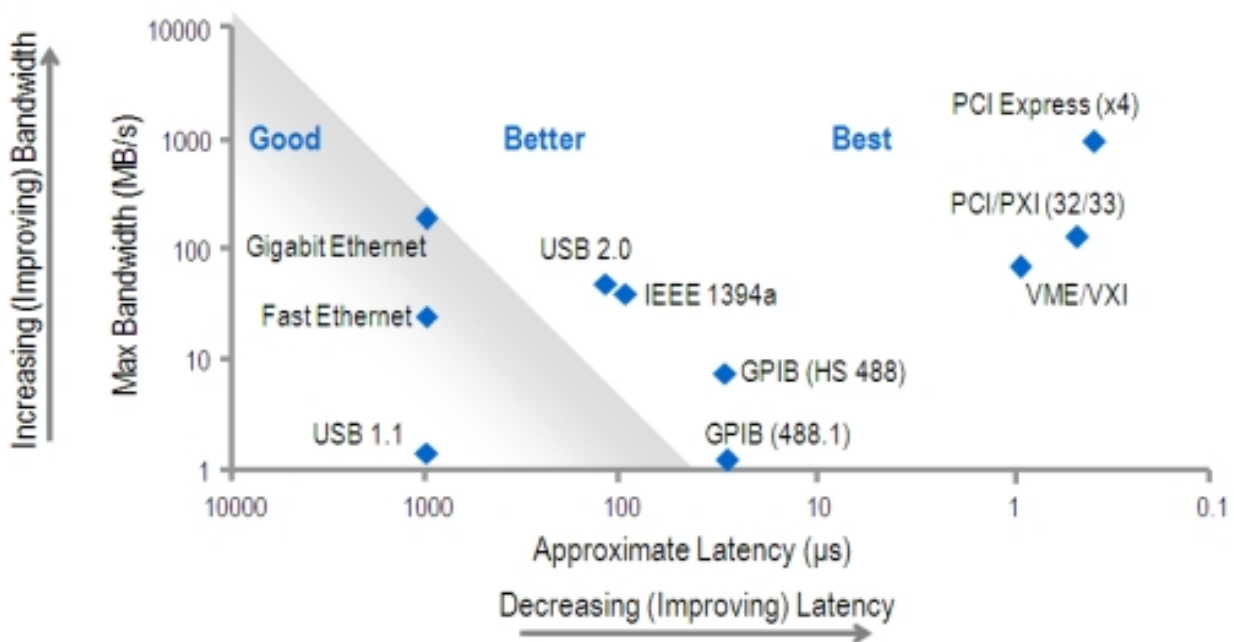


Figure 5. Bus Comparison Chart

The size and cost of buses vary greatly, as does bandwidth and latency performance specifications. Bandwidth is defined as the amount of data that can be transferred within a prescribed amount of time, and latency (also known as responsiveness) is defined as the time delay between the initiation of a request for data and the beginning of actual data transfer. PCI is a very popular bus in embedded systems due to its high bandwidth, low latency, and maturity in the PC industry. It is ideal for designing highly responsive systems with fast I/O that command DMA channels.

USB is growing in popularity in embedded systems; in fact, most modern PC/104 SBCs offer direct USB connectivity. To maximize USB performance and achieve multiple high-speed data streams for high-speed and multifunction applications, National Instruments created NI signal streaming. This technology adds device-side intelligence, custom data management hardware, and streamlined data control, which improve single-point performance by up to 1,600 percent for analog input and up to 250 percent for analog output. NI signal streaming effectively brings USB performance closer to PCI performance while reducing the overall footprint of the system. A PCI connector uses more than three times printed circuit board (PCB) real estate than a standard type-A USB receptacle. USB is also cheaper and easier to implement than PCI for custom-built computing platforms.

### NI OEM I/O Offerings

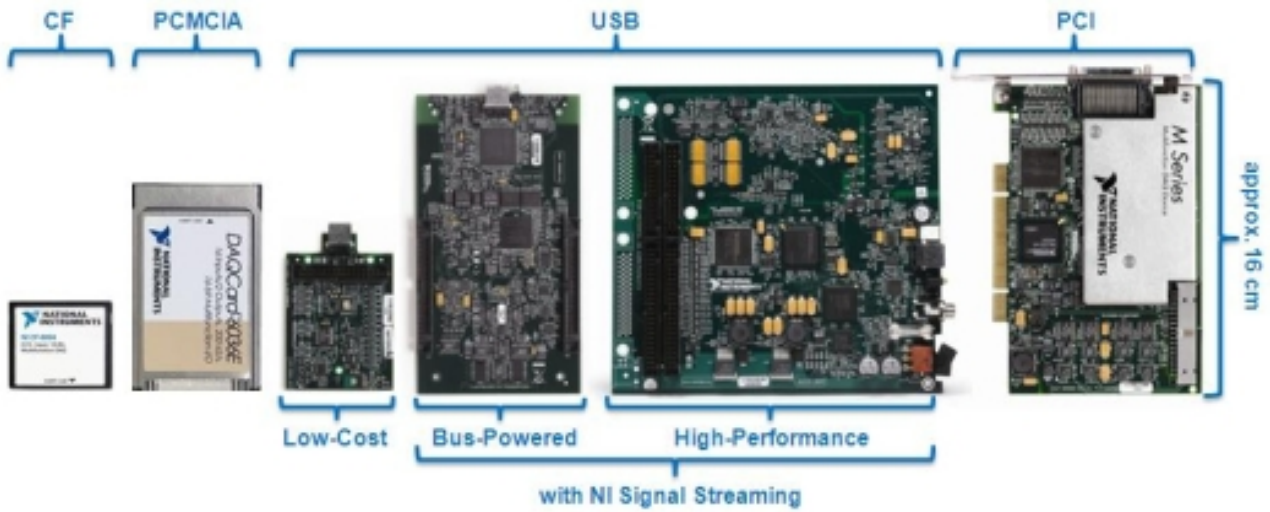


Figure 6. NI OEM I/O Offerings for Embedded Designs

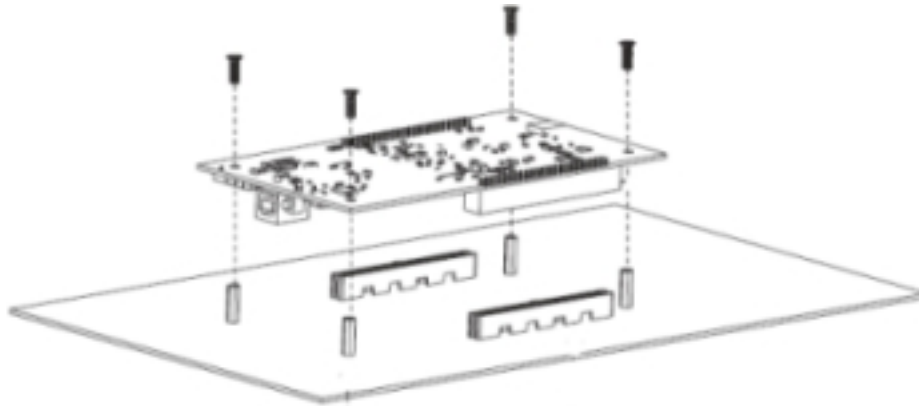
Low-cost USB and CompactFlash devices are ideal in systems where size and cost are more valuable than performance. They have small channel counts and low resolution but provide all the NI I/O software facilities previously discussed. PCMCIA and bus-powered USB devices are designed for higher-performance applications where power consumption needs to be optimized and a small form factor is still necessary. Finally, high-performance USB and PCI devices are ideal for high-speed, high-throughput, and high-accuracy applications, where external power is readily available and size is not an issue. Table 1 illustrates the maximum performance for each product category.

Product	Power	Connector type	Analog Input			Analog Output			Digital I/O (Clocked)	Counter/Timer
			Channel	Resolution	Rate	Channel	Resolution	Rate		
USB-600x Low-Cost	Bus	34-pin IDC	8 SE/4 DI	14-bit	48 kS/s	2	12-bit	Static	12 (0)	2
USB-621x Bus-Powered	Bus	50-pin IDC	32 SE/16 DI	16-bit	400 kS/s	2	16-bit	250 kS/s	32 (0)	2
USB-62xx High-Performance	External	34 and 50-pin IDC	80 SE/40 DI	18-bit	1.25 MS/s	4	16-bit	2.86 MS/s	48 (32)	2
PCI-62xx	Bus	68-pin VHDCI	80 SE/40 DI	18-bit	1.25 MS/s	4	16-bit	2.86 MS/s	48 (32)	2

Table 1. Highest Specifications of NI OEM I/O, Divided by Segment

### Connectivity

Upgrading a system has never been easier. In most cases, embedded designers can simply replace the NI OEM device and, with minimal changes in software, can double the channel count, increase the sampling rate, or even add isolation to their I/O. NI OEM USB products are designed for embedded connectivity – 34- and 50-pin IDC connectors make it easy for designers to mate OEM USB DAQ devices with another PCB or connect an inexpensive and widely available ribbon cable. NI OEM plug-in boards such as PCI and PCMCIA devices also provide widely available connectors, such as the 68-pin VHDCI or 68-pin SCSI.



*Figure 7. Mounting Procedure for the NI USB-621x OEM*

## Conclusion

Using COTS measurement products for adding I/O on embedded systems can help save development time and cost and increase system performance. National Instruments delivers high-end I/O with flexibility in connectivity, measurement performance, bus, driver software, and OS compatibility – all with the ease of use of COTS products at OEM prices. For mid- to high-volume OEM customers, NI also offers product customization such as firmware modifications, custom connectivity, and hardware modification, as well as top support and direct communication with the National Instruments R&#38;D team.

Nathan Yang

Data Acquisition Product Marketing Engineer

[National Instruments OEM Solutions](#)

[USB OEM Data Acquisition Products](#)

[National Instruments Data Acquisition Software](#)

[Register-Level Programming for OEM I/O Customers](#)

[NI Signal Streaming: Sustaining High-Speed Data Streams over USB](#)

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.