SignalRanger^{TM} DDCI⇆ Interface

***D**evelopment to **D**eployment **C**ode **I**nstrumentation*⇆

The ***DDCI***⇆ interface allows a controlling application running on a PC to communicate with and to control an embedded device. It is the cornerstone of the ***SignalRanger^{TM}*** architecture

This DDCI interface is useful during two stages in the application life-cycle:

1. During development: The interface provides real-time debugging tools at the application level, a feature not achievable using standard debugging and emulation techniques.
2. During and after deployment of the application: The ***DDCI***⇆ interface eliminates the need for any further hardware and software DSP development; the same hardware and interface can be used to provide user-control and communications with the device.

The essential features of the ***DDCI***⇆ interface are:

- The same physical USB interface, host libraries and functions (and Ethernet for SR-Mk3) are used to support the interface at all stages of the application life-cycle.
- The operation of the ***DDCI***⇆ interface requires no DSP code addition or adaptation and only requires minimal CPU time or memory overhead.
- The physical USB interface (and Ethernet for SR-Mk3) can be connected and disconnected in-operation without any disruption of the DSP code running on the *SignalRanger* platform

Advantages

Condense the two stages of the application life-cycle into a single shorter step. In-effect the application is generally deployed "as is" right after the debugging phase.

Provides extensive real-time visibility into the operation of the embedded code. In fact, in many DSP developments, it is necessary to run simulations of the signal processing algorithm to validate its operation. The real-time instrumentation provided by ***DDCI***⇆ facilitates the analysis of the high level behaviour of the signal-processing code and can do so with real-life conditions and data. Often the simulation step can be bypassed completely; this greater visibility during development directly translates to a more reliable embedded code.

The following real-time functions are supported directly by the interface:

- RAM & Flash read and write
- Peripheral read and write
- Force code execution
- CPU reset
- In-service firmware upgrade management
- Automatic target device recognition and management.

Allows the use of the extensive LabVIEW libraries for powerful real-time signal-processing, analysis and display capabilities in the debugging and in the deployed-application phases.

Symbolic access The functions of the ***DDCI***⇆ interface can be used while the embedded code is running. All these functions support *symbolic access*, where the name of DSP variables and functions can be used to access them, instead of their absolute addresses. The advantage of this feature is that the embedded DSP code can be modified and relinked without having to update the associated user-access application running on a PC. If the variable and function names remain the same, the access will stay operational across DSP code updates.

DDCI interface architecture

The ***DDCI***⇆ interface is composed of 5 mains parts:

Signed Driver for Windows XP, Windows Vista and Win 7 (x86 and x64 platforms): This driver allows the connection of any number of boards to the PC. The driver is installed as part of the *SignalRanger* installation procedure.

Communication Kernel: This kernel resides in DSP memory, along with the user's application specific code. It enhances PC to DSP communication.

LabVIEW Libraries: These libraries support wide-ranging communication, programming and control functions with the resident kernel. They can be used to build an application-specific LabVIEW executable to control the embedded *SignalRanger* device.

C/C++ libraries: For developers who prefer to work in a C/C++ environment, we provide libraries in the form of DLLs. These libraries have functionality similar to the LabVIEW libraries.

Mini-Debugger: The Mini-Debugger is a general-purpose interface application that supports programming and real-time symbolic debugging of generic DSP embedded code. It includes features such as real-time graphical data plotting, symbolic read/write access to variables, dynamic execution, Flash programming… etc. At its core, the mini-debugger uses the same interface libraries that a developer uses to design a stand-alone DSP application. This insures a seamless transition from the development/debugging environment to the deployed application.

DDCI⇆ Partner's

The *DDCI* have been developed by **Soft-dB** in collaboration with **Convergence Instruments** (www.convergenceinstruments.com). The *DDCI* interface is part of a larger methodology for the development of embedded systems and applications. It is now used with a multitude of embedded devices and is the cornerstone of the *SignalRanger* architecture.