# SPIDER-80X API

## APPLICATION PROGRAMMING INTERFACE

DEVELOP CUSTOM APPLICATIONS

PROVIDES A HIGH LEVEL INTERFACE ACCESSIBLE IN VISUAL C++, C#, AND BASIC

USE ANY PROGRAMMING LANGUAGE SUPPORTING DYNAMIC LINKED LIBRARIES

CRYSTAL instruments

www.crystalinstruments.com

# SPIDER-80X API

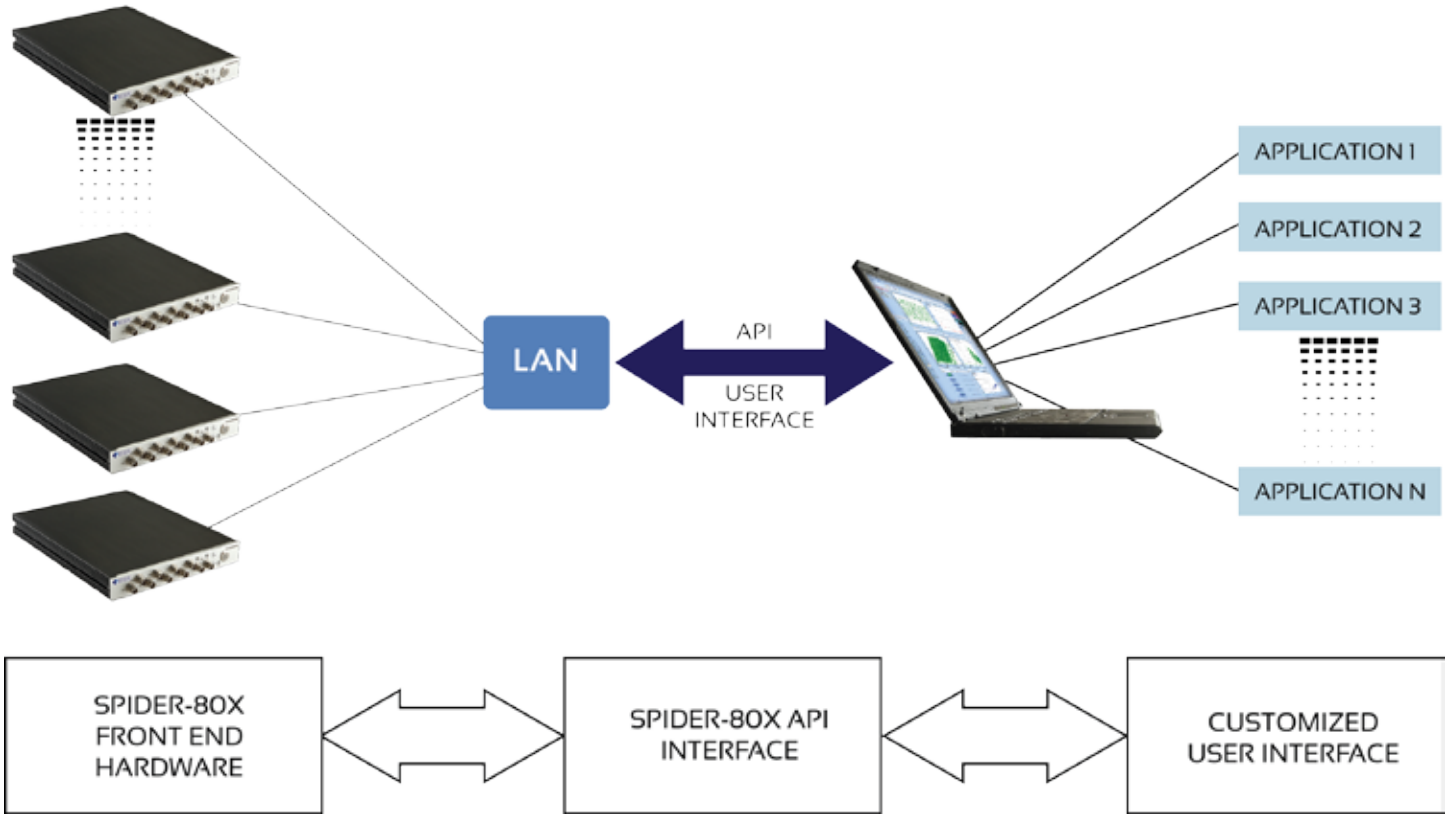
## SPIDER-80X API

Crystal Instruments (CI) Application Programming Interface (API) for the Spider-80X provides scientists and engineers a powerful tool to develop custom applications that interface directly with CI's industry leading hardware. Since 1996, Crystal Instruments has been dedicated to developing advanced DSP technologies and signal processing solutions. The Spider-80X is a highly modular, distributed, scalable dynamic measurement system. It is ideal for a wide range of industries including automotive, aviation, aerospace, electronics, and military use. The Spider-80X excels in applications requiring easy, quick, and accurate data recording and real-time signal processing.

With the Spider-80X API, users can focus on the interface of their applications and leave the hardware design to Crystal Instruments. While the Spider system is running, users have access to the real-time signal data in both the time and frequency domain. Additionally, with the long-time data recording function the Spider-80X records up to 4 GB of continuous time data to its own internal memory.

# API BASICS

The Spider-80X API is implemented in the Microsoft Visual Studio development environment and .NET framework. It provides a high level interface accessible in Visual C++, C#, and Basic. In addition, it can be used by any programming language that supports Dynamic-Linked Libraries (DLLs). Command can configure the Spider front end, control data acquisition, check the status of the processor, and retrieve DSP data.







## RUN API IN SIMULATION MODE

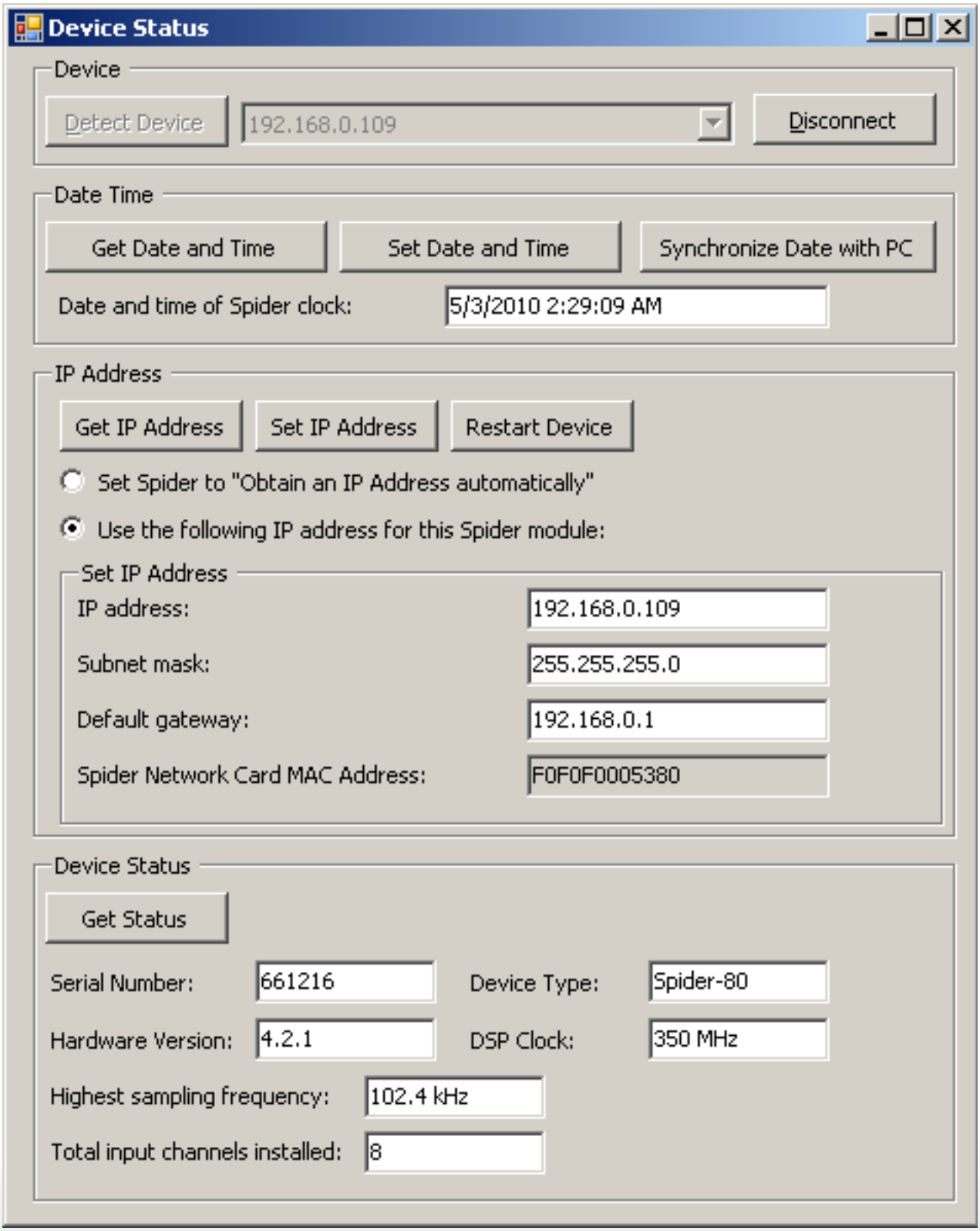
The Spider-80X API includes a Simulation Mode. This useful tool allows the user to develop applications even if the hardware is not connected. This tool also makes training and development much more efficient, especially when hardware is not continuously available.

## INCLUDED IN THE API PACKAGE

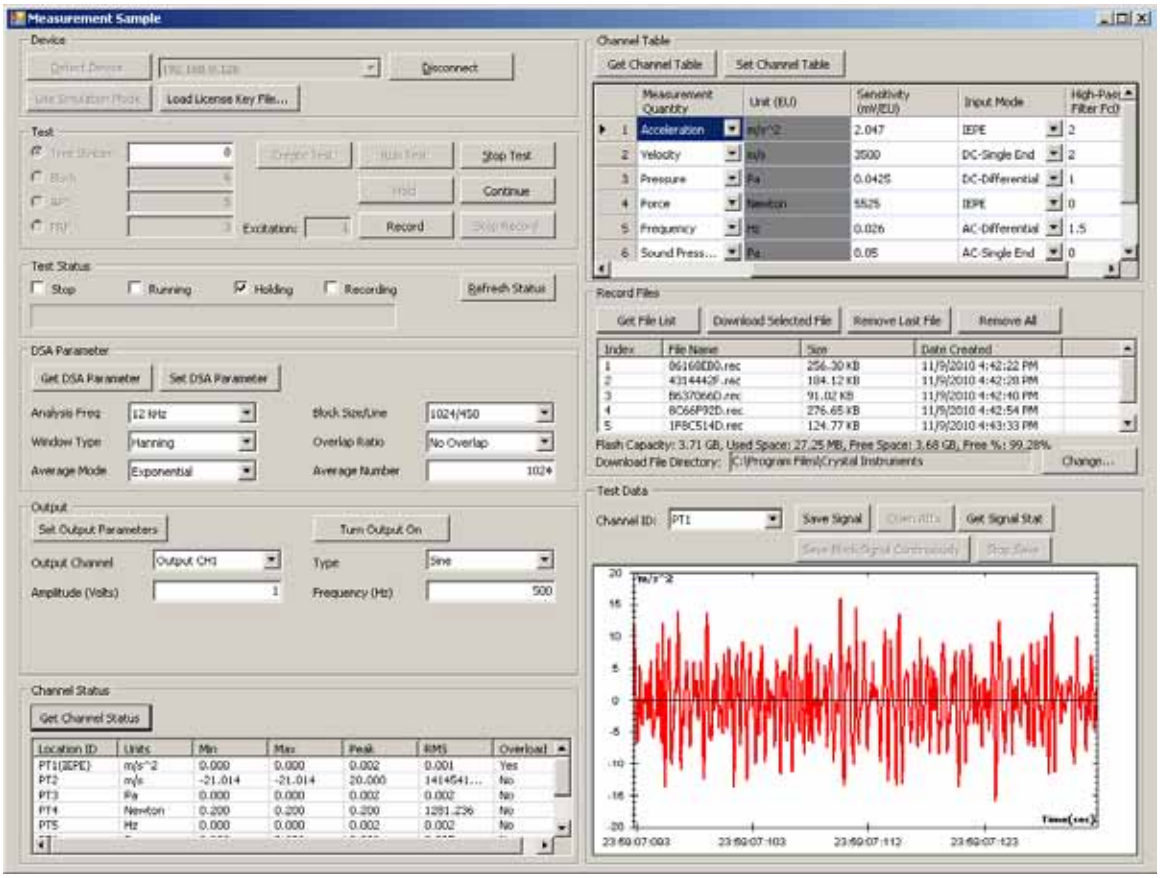
- One or More Calibrated Spider-80X Systems
- Spider-80X API Libraries
- Three API Samples
- Source Code
- Spider-80X API Product Brochure
- Spider-80X API User's Guide

With these tools offered by Crystal Instruments, users are able to easily build customized solutions. The Spider-80X system provides high quality data capturing and real-time processing. The Spider-80X API libraries provide useful function calls to control the powerful hardware. The API samples help the user verify the connection and demonstrate how to set up front-end parameters and triggers. Source code helps programmers understand how the API works internally and reduces workload.

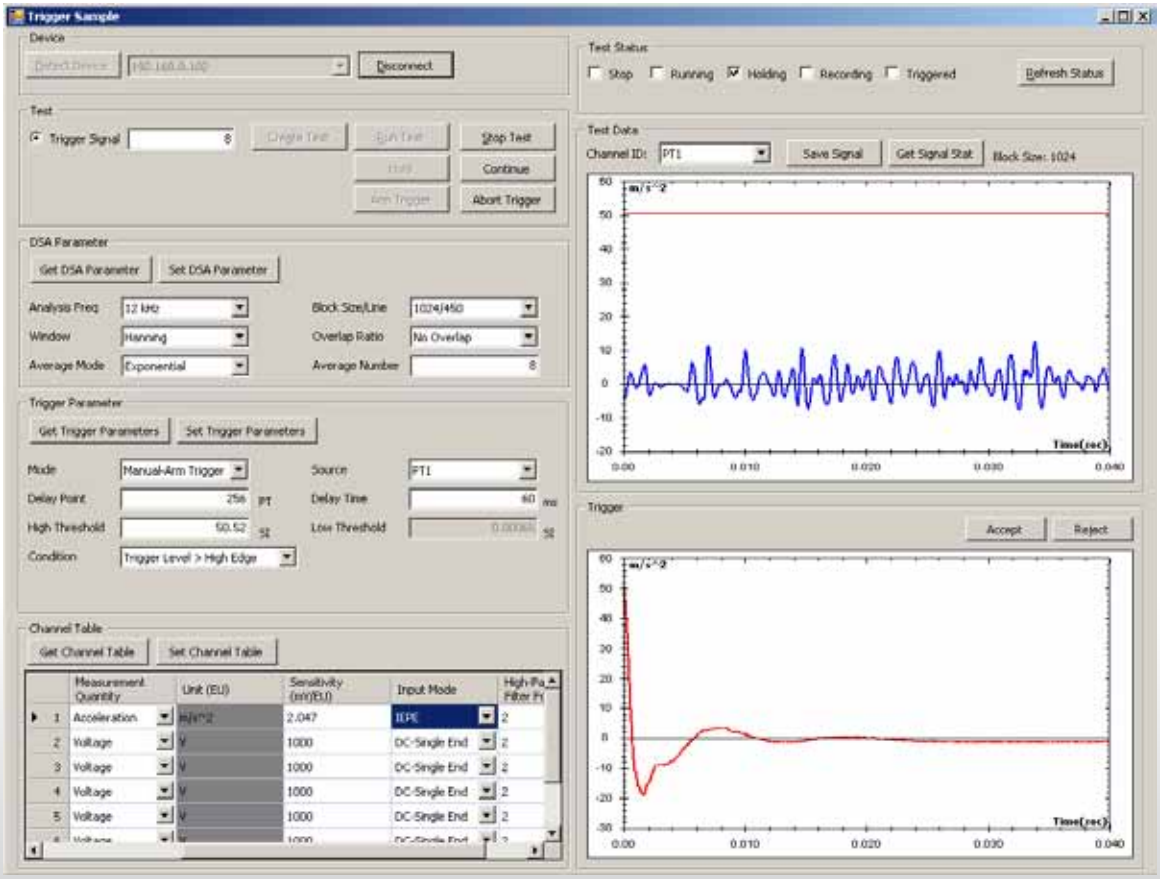
## GET STATUS


Device Status
Device
Detect Device
192.168.0.109
Disconnect
Date Time
Get Date and Time
Set Date and Time
Synchronize Date with PC
Date and time of Spider clock:
5/3/2010 2:29:09 AM
IP Address
Get IP Address
Set IP Address
Restart Device
Set Spider to "Obtain an IP Address automatically"
Use the following IP address for this Spider module:
Set IP Address
IP address:
192.168.0.109
Subnet mask:
255.255.255.0
Default gateway:
192.168.0.1
Spider Network Card MAC Address:
F0F0F0005380
Device Status
Get Status
Serial Number:
661216
Device Type:
Spider-80
Hardware Version:
4.2.1
DSP Clock:
350 MHz
Highest sampling frequency:
102.4 kHz
Total input channels installed:
8

## MEASUREMENT SAMPLE




## TRIGGER SAMPLE



# API SPECIFICATIONS

| SPIDER-80X API METHODS | |
|---|---|
| Connection Methods | |
| GetDeviceList | Get all available Spiders' information |
| Connect | Connect to a device |
| Disconnect | Disconnect from device |
| GetLastError | Get last error info |
| Command Methods | |
| SendCommand | Send commands |
| Test Methods | |
| CreateTest | Create new test |
| CreateTriggerTest | Create trigger test |
| CreateFRFTest | Create FRF test |
| GetTestStatus | Get current test status |
| GetChannelTable | Get channel table parameters |
| SetChannelTable | Set parameters for channel table |
| SetDSAParamter | Set parameters for DSA |
| GetDSAParameter | Get DSA Parameters |
| SetOutputParameter | Set parameters for output |
| GetOutputParameter | Get output parameters |
| GetChannelStatus | Get channel status |
| GetTestStatus | Get test status |
| GetSpiderTime | Get hardware system time |
| SetSpiderTime | Set hardware system time |
| GetSpiderConfig | Get hardware parameter (IP address) |
| SetSpiderConfig | Set hardware parameter (IP address) |
| CheckLicenseKey | Check the status of license key file |
| LoadLKFile | Load license key file |

| SPIDER-80X API METHODS continued | |
|---|---|
| Trigger Methods | |
| SetTriggerParameter | Set parameters for trigger |
| GetTriggerParameter | Get trigger parameters |
| TriggerArm | Trigger Arm |
| TriggerNext | Trigger Next |
| TriggerAbort | Trigger Abort |
| TriggerAccept | Trigger Accept |
| Signal Methods | |
| GetSignalStatus | Get status parameters of all signals |
| GetSignalData | Get signal data |
| GetSignalFrameCount | Get signal frame counts from atfx files |
| ReadSignal | Read signal properties and values |
| Record | |
| StartRecord | Start recording |
| StopRecord | Stop recording |
| SaveSignal | Save a frame of data |
| GetFileList | Get the list of files |
| DownloadFile | Download data file |
| RemoveLastFile | Delete the last data file |
| RemoveAll | Delete all data files on hardware |
| StartSaveSignal | Save block signal continuously |
| StopSaveSignal | Stop saving block signal continuously |
| Simulation Mode | |
| SetSimulationMode | Enter simulation mode without Spider |
| GetSimulationMode | Get simulation mode status |

| CALLBACK AND EVENTS |
| --- |
| event DeviceDSPMessageHandler DeviceDSPMessageReceived;// Called after DSP instruction message received |
| event DeviceNotifyMessageHandler DeviceNotifyMessageReceived;// Called after device status received |
| event TestRunStatHandle TestRunStatChanged;// Called after status changed |
| event DeviceDataIsReadyHandler DeviceDataIsReady;// Called after data collected |
| event DeviceConnectedHandler DeviceConnected;// Called after connected |
| event DeviceDisconnectedHandler DeviceDisconnected;// Called after disconnected |
| event DeviceReadyHandler DeviceReady;// Called after test created |
| event DeviceStoppedHandler DeviceStopped;// Called after device stopped |
| event TriggerArmedHandler TestTriggerArmedChanged;// Called after trigger armed |
| event TriggerDisarmedHandler TestTriggerDisarmedChanged;// Called after trigger disarmed |
| event TriggerFiredHandler TestTriggerFiredChanged;// Called after trigger fired |

## FULL RANGE OF SUPPORT

The most challenging part of any development project is the beginning. Crystal Instruments is here to help. We can work with users and provide assistance in defining the fundamental requirements, such as defining parameters, commands, and control settings. Crystal Instruments can even deliver an alpha version of the user's application, which includes the basic interfaces and commands needed to interact with the Spider-80X hardware.

Crystal Instruments provides a one-year hardware warranty and comprehensive tech support for each purchase. When it is time to recalibrate the Spider-80X hardware, specialized software is available to enable the user to calibrate the system, or the system can be shipped back to Crystal Instruments for calibration.

## DATA CAPTURING AND PROCESSING

The Spider-80X API can control the hardware to function as both a data recorder and dynamic signal analyzer at the same time. All time stream signals can be simultaneously recorded and displayed.

Acquisition mode controls how the data is acquired block-by-block and processes the data with signal analyzer functions. These time blocks are either gap free, with gaps, or overlapped depending on the acquisition mode selection. Real-time processing has a 46 kHz spectral bandwidth with all inputs enabled (102.4 kHz sampling rate). The sampling rate can be set in 54 increments.

Analysis functions include time capture, APS, FRF, and correlation with many windowing options available. Output source waveforms include sine, triangle, square, white noise, DC, chirp, and swept sine. Averaging can be applied to the frequency data with linear, exponential, overlap, or peak hold options. Triggering includes free-run, continuous after trigger, single shot by user, auto/manual-arm trigger.

Front end setup includes the following steps:

- Type of test (Time Stream, Block, APS, FRF)
- DSA parameters (Analysis Frequency, Block Size/Line, Window Type, Overlap Ratio, Average Mode, and Average Number)
- Output Channels (Channel Selection, Type, Amplitude, Frequency)
- Input Channels (Location ID, Measurement Quantity, Engineering Unit, Sensitivity, Input Mode, High-Pass Filter)
- Trigger Parameters (Mode, Source, Delay Point, Delay Time, High Threshold, Low Threshold, Condition)

**CRYSTAL INSTRUMENTS - HEADQUARTERS**
2370 OWEN STREET
SANTA CLARA, CA 95054, USA
PHONE: +1-408-986-8880
FAX: +1-408-834-7818
EMAIL: SALES@GO-CI.COM
WWW.CRYSTALINSTRUMENTS.COM

**CRYSTAL INSTRUMENTS INDIA PRIVATE LIMITED**
#3-6-259, BLOCK A, THIRD FLOOR, SPAN CHAMBERS
HIMAYATH NAGAR, HYDERABAD-500029
ANDHRA PRADESH, INDIA
PHONE: +91-040-23223378
MOBILE: +91-80999-70666
FAX: +91-040-23222275
EMAIL: INDIASALES@GO-CI.COM
WWW.GO-CI.IN