TANGO Device Server

Acquire_Waveform_Lecroy_Prj
User's Guide

AcquireWaveformLecroy Class

Revision: release_1_1_13 - Author: xavela
Implemented in C++ - CVS repository: tango-ds

Introduction:

This class allows the acquisition of a waveform (the description and the data), from a specific channel and from any Lecroy scope series. The description is a structure which contains in particular : the length of the array 1 (the raw data) the length of the array 2 (2nd part of the calculated waveform data, if any) the timebase the total length of the acquired data (lgth array1 + lgth array2) .... its represents the context of the acquisition. NOTE : for all Lecroy scope, the port number is the same : 1861

Class Description:

This class allows the acquisition of a waveform (the description and the data), from a specific channel and from any Lecroy scope series. The description is a structure which contains in particular : the length of the array 1 (the raw data) the length of the array 2 (2nd part of the calculated waveform data, if any) the timebase the total length of the acquired data (lgth array1 + lgth array2) .... its represents the context of the acquisition. NOTE : for all Lecroy scope, the port number is the same : 1861

Properties:

Device Properties		
Property name	**Property type**	**Description**
IPaddress	Tango::DEV_STRING	The IP address of the Lecroy scope to build a connection with.
ChannelName	Tango::DEV_STRING	The name of channel on which the acquisition will be done. This name must be composed with two caracters : (a letter followed by a number) C for a physical channel F for a calculated waveform M for a memorised waveform ... Example : C2 (is the channel 2) Default : C1 (channel 1

Device Properties Default Values:

Property Name	**Default Values**
IPaddress	No default value
ChannelName	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
OPEN	The socket TCP IP is successfully opened between the Lecroy scope and this DServer.
CLOSE	The communication between the Lecroy scope and the DServer is closed.
FAULT	The communication between the Lecroy scope and the DServer is not done.
ALARM	An error occured during a Write or Read command.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
waveArray1: Length of the first array. This second array contains data of simples waveforms.	DEV_LONG	READ	Yes
waveArray2: Length of the second array. This second array contains data of complex waveforms (as FFT, Extrema ....).	DEV_LONG	READ	Yes
waveArrayCount: Length of the two arrays. If the length of wavearray1 is different of this length that's means data are present in the array wavearray2. Which is used for complex data as math operations on a waveform.	DEV_LONG	READ	No
nominalBits: ADC resolution. For simple data this ADC is an 8 bits resolution else it's can be 10 up to 12 bits	DEV_SHORT	READ	Yes
horizontalInterval: Sampling interval for time domain waveforms Needed to scale the waveform data stored in the rawWaveformData attribute.	DEV_DOUBLE	READ	Yes
horizontalOffset: Trigger offset for the first sweep of the trigger, seconds between the trigger and the first data point. Needed to scale the waveform data stored in the rawWaveformData attribute.	DEV_DOUBLE	READ	Yes
verticalGain: The vertical gain. Used to scale the waveform data stored in the rawWaveformData attribute.	DEV_DOUBLE	READ	Yes
verticalOffset: Needed to scale the waveform data stored in the rawWaveformData attribute.	DEV_DOUBLE	READ	Yes
triggerTime	DEV_STRING	READ	No

Spectrum Attributes			
Attribute name	**Data Type**	**X Data Length**	**Expert**
rawWaveformData: Gets the raw waveform data. Maximum size set to 150000 data by default.	DEV_SHORT	150000	No
verticalScaledData: Gets the scaled waveform data. Maximum size set to 150000 data by default.	DEV_DOUBLE	150000	No

Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	**Argument In**	**Argument Out**
Init	DEV_VOID	DEV_VOID
State	DEV_VOID	DEV_STATE
Status	DEV_VOID	CONST_DEV_STRING

1 - Init

- **Description:** This commands re-initialise a device keeping the same network connection.
  After an Init command executed on a device, it is not necessary for client to re-connect to the device.
  This command first calls the device *delete_device()* method and then execute its *init_device()* method.
  For C++ device server, all the memory allocated in the *nit_device()* method must be freed in the *delete_device()* method.
  The language device desctructor automatically calls the *delete_device()* method.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_VOID : none.

- **Command allowed for:**
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::FAULT
  - Tango::ALARM

2 - State

- **Description:** This command gets the device state (stored in its *device_state* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_STATE : State Code

- **Command allowed for:**
  - Tango::OPEN
  - Tango::CLOSE

- Tango::FAULT
- Tango::ALARM

3 - Status

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  CONST_DEV_STRING : Status description

- **Command allowed for:**
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::FAULT
  - Tango::ALARM