



TANGO
Device
Server

multibunch feedback

User's Guide

MultiBunchFeed Class

Revision: V2_0 - Author: chaize
Implemented in C++ - CVS repository: ESRF

Introduction:

Implement the interface of the multibunch feedback device. The device is a client of the LiberaAccess device server and define a certain number of attributes from the FPGA parameters

Class Inheritance:

- Tango::Device_4Impl
 - MultiBunchFeed

Properties:

Device Properties		
Property name	Property type	Description
LiberaDevice	Tango::DEV_STRING	Name of the TANGO LiberaAccess device on which the FPGA program runs
Parameter_offset	Tango::DEV_LONG	offset of the first parameter in the libera
Acquisition_offset	Tango::DEV_LONG	Offset of the acquisition data
FirCoefNumber	Tango::DEV_SHORT	number of Fir coefficients. usually 7 for vertical and 8 for horizontal
Config_file_base_path	Tango::DEV_STRING	This is 'root' file base path. This string will be concatenated with parameter of loadConfigFile command, to obtain an absolute filename.

Device Properties Default Values:

Property Name	Default Values
LiberaDevice	sr/d-mfdbk/libera1
Parameter_offset	2048
Acquisition_offset	4096
FirCoefNumber	7
Config_file_base_path	/operation/operator/s700/tune/libera_config

There is no Class properties.

Attributes:

Scalar Attributes

Attribute name	Data Type	R/W Type	Expert
DacOutput	DEV_USHORT	READ_WRITE	Yes
DacDelay	DEV_USHORT	READ_WRITE	No
Gain	DEV_USHORT	READ_WRITE	No
Polarity	DEV_USHORT	READ_WRITE	No
Phase	DEV_DOUBLE	READ_WRITE	No
TriggerSource	DEV_USHORT	READ_WRITE	Yes
Channel	DEV_USHORT	READ_WRITE	Yes
SignalToArchive	DEV_USHORT	READ_WRITE	Yes
BetatronFreq	DEV_DOUBLE	READ_WRITE	Yes
HOMexLevel	DEV_USHORT	READ_WRITE	Yes
HOMmode	DEV_USHORT	READ_WRITE	Yes
HOMside	DEV_USHORT	READ_WRITE	Yes
NoiseModulation	DEV_BOOLEAN	READ_WRITE	No
Tune	DEV_DOUBLE	READ	No
FFTAveraging	DEV_USHORT	READ_WRITE	No
FreqMax	DEV_DOUBLE	READ	Yes
FreqMin	DEV_DOUBLE	READ	Yes
Scaling: define if the spectrum is displayed in frequency unit or in tune unit	DEV_USHORT	READ_WRITE	No
SingleBunch	DEV_BOOLEAN	READ_WRITE	No
MaxIndex	DEV_SHORT	READ	Yes
TrackSpan	DEV_DOUBLE	READ_WRITE	Yes
RefCurveFreqMin	DEV_DOUBLE	READ	Yes
RefCurveFreqMax	DEV_DOUBLE	READ	Yes
ConfigFileName	DEV_STRING	READ	Yes
Shaker	DEV_BOOLEAN	READ_WRITE	No
SelectOutput: Select if the output is ADC or FIR	DEV_USHORT	READ_WRITE	No
TimeDomain: Define if the output is in frequency or in time. In frequency, the signal to archive is 10khz or 88 khz in time, the signal to archive is adc or DAC	DEV_BOOLEAN	READ_WRITE	Yes
FrequencySpan: Define if the full span is 10 Khz or 88 Khz	DEV_USHORT	READ_WRITE	No

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
Coefficients	DEV_DOUBLE	8	Yes
Idata	DEV_DOUBLE	4096	No
Qdata	DEV_DOUBLE	4096	No
FFT	DEV_DOUBLE	4096	No
ReferenceFFT	DEV_DOUBLE	4096	Yes

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
On	DEV_VOID	DEV_VOID
Off	DEV_VOID	DEV_VOID
Reset	DEV_VOID	DEV_VOID
SoftTrig	DEV_VOID	DEV_VOID

Device Commands for Expert Level Only		
Command name	Argument In	Argument Out
SoftArm	DEV_VOID	DEV_VOID
LoadConfigurationFile	DEV_STRING	DEV_VOID
SaveConfigurationFile	DEV_STRING	DEV_VOID
GetConfigurationFilePath	DEV_VOID	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:**

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:**

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:**

4 - On

- **Description:** switch the feedback ON
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - Off

- **Description:** Switch the feedback OFF
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - Reset

- **Description:** reset the faults
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - SoftArm (for expert only)

- **Description:** software arm
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :

- **Command allowed for:**

8 - SoftTrig

- **Description:** does a software trigger of the acquisition
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

9 - LoadConfigurationFile (for expert only)

- **Description:** This command will load configuration file and write attributes and command accordingly.
- **Argin:**
DEV_STRING : This is the path concatenated to 'Config_file_base_path' property.
- **Argout:**
DEV_VOID :
- **Command allowed for:**

10 - SaveConfigurationFile (for expert only)

- **Description:** this command saves actual Server parameters
- **Argin:**
DEV_STRING : Name of the configuration file
- **Argout:**
DEV_VOID :
- **Command allowed for:**

11 - GetConfigurationFilePath (for expert only)

- **Description:** This command returns the absolute path of configuration path. It can be found by reading device property.
- **Argin:**

DEV_VOID :

- **Argout:**
DEV_STRING :

- **Command allowed for:**

<p>TANGO is an open source project hosted by : SOURCEFORGE.NET®</p>	<p>Core and Tools : CVS repository on tango-cs project Device Servers : CVS repository on tango-ds project</p>
---	--