



TANGO
Device
Server

ESRF BunchClock device server

User's Guide

BunchClock Class

Revision: BunchClock-Release_1_1 - Author: kolozhva
Implemented in C++ - CVS repository: ESRF

Introduction:

This class will allow to control the ESRF bunchclock.

Class Inheritance:

- Tango::Device_4Impl
 - BunchClock

Properties:

Class Properties		
Property name	Property type	Description
Icv196_timeout	Tango::DEV_LONG	Timeout in ms used when waiting for the injection and extraction signals. -1 => wait forever.

Device Properties		
Property name	Property type	Description
Icv196_timeout	Tango::DEV_LONG	Timeout in ms used when waiting for the injection and extraction signals. -1 => wait forever.
Board_pathname	Tango::DEV_STRING	VME board (ICV196) pathname
Syn_inj	Tango::DEV_BOOLEAN	Value of the Sin_inj attribute at startup. true => synchronous injection false => asynchronous injection
Syn_ext	Tango::DEV_BOOLEAN	Value of the Syn_ext attribute at init. true => synchronous extraction false => asynchronous extraction
Ext_compensation	Tango::DEV_BOOLEAN	Value of the Ext_compensation attribute at init. true => bit mod1 of the control register = 1 => extraction compensation enabled false => bit mod1 of the control register = 0 => extraction compensation disabled
Output_pulses	Tango::DEV_BOOLEAN	Value of the Output_pulses attribute at init. true => pulses enabled - Bit 6 (inh) of the control register = 1 false => pulses disabled - Bit 6 (inh) of the control register = 0
Bunch_list	Array of short	Bunch list loaded at start-up of the device server.

Class Properties Default Values:

Property Name	Default Values
Icv196_timeout	1000

Device Properties Default Values:

Property Name	Default Values
Icv196_timeout	1000
Board_pathname	/dev/icv1
Syn_inj	true
Syn_ext	true
Ext_compensation	false
Output_pulses	true
Bunch_list	0

States:

States	
Names	Descriptions
ON	The BunchClock is ON and the thread loop changing the bucket number is running.
FAULT	The thread loop changing the bucket number is stopped because there is no extraction or injection signal (=> Communication problem with the hardware?) or because the bunchs list is empty.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
Syn_inj: true => synchronous injection false => asynchronous injection	DEV_BOOLEAN	READ_WRITE	No
Syn_ext: true => synchronous extraction false => asynchronous extraction	DEV_BOOLEAN	READ_WRITE	No
Ext_compensation: Extraction compensation. True => bit mod1 of the control register = 1 => extraction compensation enabled False => bit mod1 of the control register = 0 => extraction compensation disabled	DEV_BOOLEAN	READ_WRITE	No
Output_pulses: Output pulses: True => pulses enabled - Bit 6 (inh) of the control register = 1 False =>pulses disabled - Bit 6 (inh) of the control register = 0	DEV_BOOLEAN	READ_WRITE	No

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
Bunch_list: The list of buckets to fill (from 0 to 992).	DEV_SHORT	1000	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
Reset	DEV_VOID	DEV_VOID

Device Commands for Expert Level Only		
Command name	Argument In	Argument Out
WriteWord	DEVVAR_SHORTARRAY	DEV_VOID

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::ON
 - Tango::FAULT

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::ON
- Tango::FAULT

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::ON
- Tango::FAULT

4 - Reset

- **Description:** Reset the state machine

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

- **Command allowed for:**

- Tango::ON
- Tango::FAULT

5 - WriteWord (for expert only)

- **Description:** Will write the data `argin[0]` in the register `argin[1]` of the BunchClock.

- **Argin:**

DEVVAR_SHORTARRAY : `argin[0]` = data, `argin[1]` = register

- **Argout:**

DEV_VOID :

- **Command allowed for:**

- Tango::ON

- Tango::FAULT

TANGO is an open source project hosted by :
SOURCEFORGE.NET®

Core and Tools : CVS repository on tango-cs project
Device Servers : CVS repository on tango-ds project