



TANGO Device Server

Control of the N354 clock module for libera BPM User's Guide

BpmLiberaClock Class

Revision: - Author: anatoly.kolozhvari
Implemented in C++ - CVS repository: ESRF

Introduction:

Class Identification:

- **Contact :** at esrf.fr - anatoly.kolozhvari
- **Class Family :** BeamDiag
- **Platform :** Unix Like
- **Bus :** Ethernet

Class Inheritance:

- Tango::Device_4Impl
  - BpmLiberaClock

Properties:

Device Properties		
Property name	**Property type**	**Description**
DeviceIP	Tango::DEV_STRING	IP-address or name of the N354 module
N354Port	Tango::DEV_LONG	TCP/IP port of the N354 device to create a socket connection
RF_value	Tango::DEV_DOUBLE	Value of the radio frequency of the SR (MHz). It is used by the device server to convert absolute time (like seconds or microseconds) into internal units of the N354 (which is the RF period).
FPGA_file_name	Tango::DEV_STRING	Full file name of the FPGA image file. The file is downloaded into N354 FPGA during initialisation of the device by the server.

Device Properties Default Values:

Property Name	**Default Values**
DeviceIP	No default value
N354Port	5001
RF_value	352.2
FPGA_file_name	/opt/dserver/common/FPGA-bin/N354/n354mbpm.bit

There is no Class properties.

States:

States	
Names	**Descriptions**
ON	The device is ready, but does not generate triggers
RUNNING	The device is generating triggers
FAULT	The device has problems
ALARM	

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
KickerLoadDelay: Delay of the selection widow relatively to the kicker pulse	DEV_DOUBLE	READ_WRITE	No
KickerWindowWidth: Width of the selection window	DEV_DOUBLE	READ_WRITE	No
TriggerSource: true - kicker extraction false - kicker 1	DEV_BOOLEAN	READ_WRITE	No
SRClockDelay	DEV_USHORT	READ_WRITE	No
OutputPulseWidth: 0 : width = 1 SR period (2.82 usec) 1: 2 SR periods (5.64 usec) 2-7 4 SR periods (11.28 usec)	DEV_USHORT	READ_WRITE	No
DieTemperature: Maximum permitted temperature of the module	DEV_USHORT	READ	No
AmbTemperature: Measured temperature of the module	DEV_USHORT	READ	No
TemperatureAlert: A bit #16 taken from the device status register 0x04 causes ALARM state	DEV_BOOLEAN	READ	No
TemperaturePanic: Bit #17 taken from the status register 0x04, causes ALARM state	DEV_BOOLEAN	READ	No
SetDivider	DEV_SHORT	READ_WRITE	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
Stop	DEV_VOID	DEV_VOID
OneShot	DEV_VOID	DEV_VOID
ForcedOneShot	DEV_VOID	DEV_VOID
AsyncContinuous	DEV_VOID	DEV_VOID
SyncContinuous	DEV_VOID	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::RUNNING
  - 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::ON
  - Tango::RUNNING
  - 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::ON
  - Tango::RUNNING
  - Tango::FAULT
  - Tango::ALARM

4 - Stop

- **Description:** Stops the generation of triggers originating from the kicker pulses

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::RUNNING
  - Tango::FAULT
  - Tango::ALARM

5 - OneShot

- **Description:** Issues a single trigger originating from the kicker pulse and synchronised with RF

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::RUNNING
  - Tango::FAULT
  - Tango::ALARM

6 - ForcedOneShot

- **Description:** Issues a single trigger synchronised with RF independently of the kicker trigger

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::RUNNING
  - Tango::FAULT
  - Tango::ALARM

7 - AsyncContinuous

- **Description:** Issue a continuous acquisition independently of kickers and RF synchronisation

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::RUNNING
  - Tango::FAULT
  - Tango::ALARM

8 - SyncContinuous

- **Description:** Starts generation of triggers originating from kicker pulses and synchronised with RF

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::RUNNING
  - Tango::FAULT

- Tango::ALARM

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