



TANGO
Device
Server

Retriggerable Single Pulse Generation User's Guide

RetrigPulseGeneration Class

Revision: release_1_0_7 - Author: vince_soleil
Implemented in C++ - CVS repository: tango-ds

Introduction:

Generate a single pulse on 8 counters each time a trigger is received. The delay and the width of each pulse is configurable.

Class Inheritance:

- Tango::Device_4Impl
 - RetrigPulseGeneration

Properties:

Device Properties		
Property name	Property type	Description
TriggerSource	Tango::DEV_STRING	The name of the terminal for the input trigger.
BoardName	Tango::DEV_STRING	The name of the board.
AutoStart	Tango::DEV_BOOLEAN	If true, the generation will start at device startup and when command "Init" is executed.
Clock	Tango::DEV_STRING	The clock used. Possible values are "INTERNAL" or "PFIxx", with xx the pin number.
ExternalFrequency	Tango::DEV_DOUBLE	In case of using an external clock, the frequency of this clock in Hz.
IdleStateCounter0	Tango::DEV_STRING	cCan be LOW or HIGH.
IdleStateCounter1	Tango::DEV_STRING	
IdleStateCounter2	Tango::DEV_STRING	
IdleStateCounter3	Tango::DEV_STRING	
IdleStateCounter4	Tango::DEV_STRING	
IdleStateCounter5	Tango::DEV_STRING	
IdleStateCounter6	Tango::DEV_STRING	
IdleStateCounter7	Tango::DEV_STRING	

Device Properties Default Values:

Property Name	Default Values
TriggerSource	No default value
BoardName	No default value
AutoStart	No default value
Clock	No default value
ExternalFrequency	No default value
IdleStateCounter0	No default value
IdleStateCounter1	No default value
IdleStateCounter2	No default value
IdleStateCounter3	No default value
IdleStateCounter4	No default value
IdleStateCounter5	No default value
IdleStateCounter6	No default value
IdleStateCounter7	No default value

There is no Class properties.

States:

States	
Names	Descriptions
UNKNOWN	
STANDBY	
RUNNING	

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
counter0Enable: counter 0 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter0: Delay in milliseconds for counter 0.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter0: The pulse width in milliseconds of pulses generated by counter 0.	DEV_DOUBLE	READ_WRITE	No
counter1Enable: counter 1 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter1: Delay in milliseconds for counter 1.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter1: The pulse width in milliseconds of pulses generated by counter 1.	DEV_DOUBLE	READ_WRITE	No
counter2Enable: counter 2 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter2: Delay in milliseconds for counter 2.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter2: The pulse width in milliseconds of pulses generated by counter 2.	DEV_DOUBLE	READ_WRITE	No
counter3Enable: counter 3 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter3: Delay in milliseconds for counter 3.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter3: The pulse width in milliseconds of pulses generated by counter 3.	DEV_DOUBLE	READ_WRITE	No
counter4Enable: counter 4 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter4: Delay in milliseconds for counter 4.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter4: The pulse width in milliseconds of pulses generated by counter 4.	DEV_DOUBLE	READ_WRITE	No
counter5Enable: counter 5 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter5: Delay in milliseconds for counter 5.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter5: The pulse width in milliseconds of pulses generated by counter 5.	DEV_DOUBLE	READ_WRITE	No
counter6Enable: counter 6 is disable if 0, enable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter6: Delay in milliseconds for counter 6.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter6: The pulse width in milliseconds of pulses generated by counter 6.	DEV_DOUBLE	READ_WRITE	No
counter7Enable: counter 7 is enable if true, disable otherwise.	DEV_BOOLEAN	READ_WRITE	No
delayCounter7: Delay in milliseconds for counter 7.	DEV_DOUBLE	READ_WRITE	No
pulseWidthCounter7: The pulse width in milliseconds of pulses generated by counter 7.	DEV_DOUBLE	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
Start	DEV_VOID	DEV_VOID
Stop	DEV_VOID	DEV_VOID
Abort	DEV_VOID	DEV_VOID
ResetBoard	DEV_VOID	DEV_VOID

Device Commands for Expert Level Only		
Command name	Argument In	Argument Out
GetDriverVersion	DEV_VOID	DEV_STRING
GetBoardsInChassis	DEV_VOID	DEV_STRING
GetBoardType	DEV_STRING	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::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

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::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

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::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

4 - Start

- **Description:** Start generations of pulses.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::STANDBY

5 - Stop

- **Description:** Stop generations of pulses.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::RUNNING

6 - Abort

- **Description:** Abort generations.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::RUNNING

7 - ResetBoard

- **Description:** Reset the physical board. After this command, the board is in an unknown state (no more configured). The device must be initialized.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

8 - GetDriverVersion (for expert only)

- **Description:** Get the current version of the driver installed for NI counter boards.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_STRING : The driver version
- **Command allowed for:**
 - Tango::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

9 - GetBoardsInChassis (for expert only)

- **Description:** Get a string containing all the boards names (from NI) present in the chassis where this device is running. This names are defined in MAX.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_STRING : All the names of NI boards.
- **Command allowed for:**
 - Tango::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

10 - GetBoardType (for expert only)

- **Description:** Returns the type (ex: PXI-6602) of a specified board.
- **Argin:**
DEV_STRING : The board name.
- **Argout:**
DEV_STRING : The board type.
- **Command allowed for:**
 - Tango::UNKNOWN
 - Tango::STANDBY
 - Tango::RUNNING

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