TANGO
Device
Server

Event Counting
User's Guide

EventCounting Class

Revision: release_1_3_2 - Author: julien_malik
Implemented in C++ - CVS repository: tango-ds

Introduction:

Perform event counting on 7 input counters.
There is 2 modes:
- Time mode: The counters count during a certain time parametrable.
- Pulse mode: The counters count until counter1 has reached a parametrable value.

Properties:

Device Properties

Property name	Property type	Description
BoardName	Tango::DEV_STRING	The name of the board as specified in Measurement & Automation eXplorer (MAX).

Device Properties Default Values:

Property Name	Default Values
BoardName	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
FAULT	An error has occured on the board.
RUNNING	The counting is running
STANDBY	The counting is not running.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
integrationTime: (cf. description of attribute mode) - in time mode: the time to count in seconds. Use an hardware clock. - in pulse mode: the value reached by counter 1 that will stop all counters (only integer values).	DEV_DOUBLE	READ_WRITE	No
countingCompleted: indicate whether the counting operation is finished (1) or not (0).	DEV_SHORT	READ	No
mode: The chosen mode: - 1 : time mode, the counters will count for the time indicated by the attribute integrationTime (in secs) - 0: pulse mode, the counters will count until counter 1 has reached the value indicated by the attribute integrationTime (integer)	DEV_SHORT	READ_WRITE	No
counter1: The counter 1 current value.	DEV_DOUBLE	READ	No
counter2: The counter 2 current value.	DEV_DOUBLE	READ	No
counter3: The counter 3 current value.	DEV_DOUBLE	READ	No
counter4: The counter 4 current value.	DEV_DOUBLE	READ	No
counter5: The counter 5 current value.	DEV_DOUBLE	READ	No
counter6: The counter 6 current value.	DEV_DOUBLE	READ	No
counter7: The counter 7 current value.	DEV_DOUBLE	READ	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
Abort	DEV_VOID	DEV_VOID
ResetBoard	DEV_VOID	DEV_VOID
ResetCounter	DEV_SHORT	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::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

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::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

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::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

4 - Start

- **Description:** Start counting.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::STANDBY

5 - Abort

- **Description:** Stop the current counting action.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

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

6 - ResetBoard

- **Description:** Reset the physical board. The device needs to be init after.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::STANDBY

7 - ResetCounter

- **Description:** Reset the current value of a counter (put it to 0). NB: To reset all counters, use Init command.

- **Argin:**
  DEV_SHORT : The counter to reset ,1 to 7).

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::STANDBY

8 - GetDriverVersion (for expert only)

- **Description:** Get the current driver version used to control the counting board.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : The driver version.

- **Command allowed for:**
  - Tango::FAULT
  - Tango::STANDBY

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::FAULT
  - Tango::STANDBY

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::FAULT
  - Tango::STANDBY