TANGO
Device
Server

PEM100
User's Guide

PEM100 Class

Revision: release_1_0_2 - Author: jean_coquet
Implemented in C++

Introduction:

Device used to control the PEM100 controller associated with a photoelastic modulator Communication part managed by Serial Device with the following RS232 parameters: BaudRate = 2400 Newline = 10 CharLength = 8 StopBits = 0 (means 1 bit) SerialLine = COMx (where x is the number of the serial port) TimeOut = 2000 Parity = none

Class Inheritance:

- Tango::Device_3Impl
  - PEM100

Properties:

Device Properties		
Property name	**Property type**	**Description**
SerialLineProxyName	Tango::DEV_STRING	SerialLine Proxy Name

Device Properties Default Values:

Property Name	Default Values
SerialLineProxyName	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
ON	normal operation
FAULT	communication problem

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
retardation: Phase Shift (memorized)	DEV_DOUBLE	READ_WRITE	No
frequency: displays the 1F or 2 F frequency according to activated 1F or 2F	DEV_DOUBLE	READ	No
waveLength: wavelength (memorized)	DEV_DOUBLE	READ_WRITE	No
voltage: voltage control	DEV_DOUBLE	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
ChangeHarmonic	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::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 - ChangeHarmonic

- **Description:** change current harmonic monitoring
- **Argin:**
  DEV_VOID : nothing
- **Argout:**
  DEV_VOID : nothing
- **Command allowed for:**
  - Tango::ON
  - Tango::FAULT

PEM100
User's Guide

PEM100 Class

Revision: release_1_0_2 - Author: jean_coquet
Implemented in C++

Introduction:

Device used to control the PEM100 controller associated with a photoelastic modulator Communication part managed by Serial Device with the following RS232 parameters: BaudRate = 2400 Newline = 10 CharLength = 8 StopBits = 0 (means 1 bit) SerialLine = COMx (where x is the number of the serial port) TimeOut = 2000 Parity = none

Class Inheritance:

- Tango::Device_3Impl
  - PEM100

Properties:

Device Properties		
Property name	**Property type**	**Description**
SerialLineProxyName	Tango::DEV_STRING	SerialLine Proxy Name

Device Properties Default Values:

Property Name	Default Values
SerialLineProxyName	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
ON	normal operation
FAULT	communication problem

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
retardation: Phase Shift (memorized)	DEV_DOUBLE	READ_WRITE	No
frequency: displays the 1F or 2 F frequency according to activated 1F or 2F	DEV_DOUBLE	READ	No
waveLength: wavelength (memorized)	DEV_DOUBLE	READ_WRITE	No
voltage: voltage control	DEV_DOUBLE	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
ChangeHarmonic	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::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 - ChangeHarmonic

- **Description:** change current harmonic monitoring

- **Argin:**
  DEV_VOID : nothing

- **Argout:**
  DEV_VOID : nothing

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

Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client.
Link to Non-frame version.

TANGO Device Server

PEM100
Device Commands Description
PEM100 Class

Revision: release_1_0_2 - Author: jean_coquet

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 - ChangeHarmonic

- **Description:** change current harmonic monitoring

- **Argin:**
  DEV_VOID : nothing

- **Argout:**
  DEV_VOID : nothing

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

ESRF - Software Engineering Group