



TANGO Device Server

REP1806
User's Guide

REP1806 Class

Revision: keyword substitution - Author: sebleport
Implemented in C++ - CVS repository: tango-ds

Introduction:

This device is used to control the REP1806 device which include 2 parts: - temperature controller (3504 process controller) - Thyristor power unit B 70.9040) it is possible to : - configure the regulation parameters - apply preset temperature - monitor sample temperature - monitor oven temperature - limit heating power To set regulation parameters, 2 modes are available: - manual mode: the user independently set each regulation parameters (P, I, D, cutback) - auto mode: a predefined list is contained in a property and the parameters are loaded at init device the communication is managed by a device Serial line called "ds_Serial". the communication parameters are: baudrate = 9600; parity = NONE; serialline = COMx; timeour = 2000 ms; charlength = 8; stopbit = 0 (means 1 stop bit).

Class Inheritance:

- Tango::Device_4Impl
  - REP1806

Class Description:

This device is used to control the REP1806 device which include 2 parts: - temperature controller (3504 process controller) - Thyristor power unit B 70.9040) it is possible to : - configure the regulation parameters - apply preset temperature - monitor sample temperature - monitor oven temperature - limit heating power To set regulation parameters, 2 modes are available: - manual mode: the user independently set each regulation parameters (P, I, D, cutback) - auto mode: a predefined list is contained in a property and the parameters are loaded at init device the communication is managed by a device Serial line called "ds_Serial". the communication parameters are: baudrate = 9600; parity = NONE; serialline = COMx; timeour = 2000 ms; charlength = 8; stopbit = 0 (means 1 stop bit).

Properties:

Device Properties		
Property name	**Property type**	**Description**
PIDTable	Array of string	list of PID parameters according to the: - oven model - sample environnement (air/vaccum) - temperature range each line is consisted of 8 fields separated by commas like: oven_model,sample_env,temp_min,temp_max,P,I,D,cutback ex: H10,air,0.75,1.2,31,5.2,10 H10,vaccum,75,100,3.2,13,2.1,27 ... H205,air,250,NotLimitMax,7.5,8,1.3,65
SafetyTemperature	Tango::DEV_DOUBLE	limit the preset temperature
SerialProxyName	Tango::DEV_STRING	name of the serial line device proxy

Device Properties Default Values:

Property Name	**Default Values**
PIDTable	No default value
SafetyTemperature	100
SerialProxyName	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
autoPIDSetting: to choose the setting PID parameters mode. true means auto, false means manual	DEV_BOOLEAN	WRITE	No
rampTime: ramp time (memorized)	DEV_DOUBLE	READ_WRITE	No
sampleTemperature: sample temperature to monitor	DEV_DOUBLE	READ	No
temperature: oven temperature (memorized)	DEV_DOUBLE	READ_WRITE	No
proportionalRegulCoeff: proportional Regulation Coefficient	DEV_DOUBLE	READ_WRITE	No
IntegralRegulCoeff: Integral Regulation Coefficient (memorized)	DEV_BOOLEAN	READ_WRITE	No
DifferentialRegulCoeff: Differential Regulation Coeff (memorized)	DEV_DOUBLE	READ_WRITE	No
cutBack: cut Back (memorized)	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
PowerOff	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:**

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:**

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:**

4 - PowerOff

- **Description:** power off

- **Argin:**
  DEV_VOID : void

- **Argout:**
  DEV_VOID : void

- **Command allowed for:**

TANGO is an open source project hosted by : 	Core and Tools : CVS repository on tango-cs project Device Servers : CVS repository on tango-ds project