



**TANGO**  
Device  
Server

# control of cryogenic loop User's Guide

## CryoCtrl Class

Revision: - Author: chaize  
Implemented in C++ - CVS repository: ESRF

### Introduction:

this class controls a cryogenic generator connected to a wago controller. It supposes that 4 signals are connected to 4 ADC channels of the controller. the 4 signals are: Temperature supply, temperature return, pressure supply and pressure return It uses the WagoAdc classe. It calculates the extracted power with these 4 items

### Class Inheritance:

- Tango::Device\_4Impl
  - CryoCtrl

## Properties:

<b>Device Properties</b>		
<b>Property name</b>	<b>Property type</b>	<b>Description</b>
<b>T_supply_adcname</b>	Tango::DEV_STRING	name of the Wagoadc tango device used to build the Tsupply attribute
<b>T_return_adcname</b>	Tango::DEV_STRING	name of the Wagoadc tango device used to build the Treturn attribute
<b>P_supply_adcname</b>	Tango::DEV_STRING	name of the Wagoadc tango device used to build the Psupply attribute
<b>P_return_adcname</b>	Tango::DEV_STRING	name of the Wagoadc tango device used to build the Preturn attribute
<b>Power_factor</b>	Tango::DEV_DOUBLE	extracted power = deltaT * deltaP * power_factor
<b>Averaging</b>	Tango::DEV_LONG	number of point to average.
<b>DeltaPREF</b>	Tango::DEV_DOUBLE	reference temperature difference. It enters in the extracted power calculation as follow: Power = sqrt(DeltaP/DeltaPREF) * DeltaT * power_factor
<b>DeltaT0</b>	Tango::DEV_DOUBLE	Temperature difference for an extracted power of 0.0 watt

## Device Properties Default Values:

<b>Property Name</b>	<b>Default Values</b>
T_supply_adcname	id/id6-cryo/t-supply
T_return_adcname	id/id6-cryo/t-return
P_supply_adcname	id/id6-cryo/p-supply
P_return_adcname	id/id6-cryo/p-return
Power_factor	1
Averaging	1
DeltaPREF	0.49
DeltaT0	1.25

**There is no Class properties.**

## Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
<b>Tsupply</b>	DEV_DOUBLE	READ	Yes
<b>DeltaT</b> : Temperature difference between the supply and the return	DEV_DOUBLE	READ	No
<b>Psupply</b>	DEV_DOUBLE	READ	Yes
<b>DeltaP</b> : Pressure difference between the supply and the return	DEV_DOUBLE	READ	No
<b>Epower</b> : Extracted power in W: it is function of the temperature difference multiplied by the pressure difference $P = \sqrt{\Delta P / \Delta P_{REF}} * \Delta T * Power\_factor$	DEV_DOUBLE	READ	No
<b>Treturn</b>	DEV_DOUBLE	READ	No
<b>Preturn</b>	DEV_DOUBLE	READ	No

## Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	Argument In	Argument Out
<b>Init</b>	DEV_VOID	DEV_VOID
<b>State</b>	DEV_VOID	DEV_STATE
<b>Status</b>	DEV_VOID	CONST_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:**

## 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:**

TANGO is an open source project hosted by :  
**SOURCEFORGE.NET**<sup>®</sup>

Core and Tools : CVS repository on tango-cs project  
Device Servers : CVS repository on tango-ds project