



TANGO
Device
Server

Class to control two BROOKS 5850E debimeters User's Guide

MassFlowController Class

Revision: release_1_0_3 - Author: vince_soleil
Implemented in C++ - CVS repository: tango-ds

Introduction:

DIFABS (Soleil) controls two debimeters Independently or in MasterSlave mode sending tensions between 0 - 5 Volts.

Class Inheritance:

- Tango::Device_4Impl
 - MassFlowController

Properties:

Device Properties		
Property name	Property type	Description
SSAOProxy	Tango::DEV_STRING	The proxy on the 6208V CPCI card
SSAIProxy	Tango::DEV_STRING	Proxy on the SingleShotAI device, used to read the voltage generated with the SSAO board.
FlowMax	Tango::DEV_DOUBLE	Flow max used to convert percentage in l/min
MasterNumber	Tango::DEV_SHORT	To define what debimter (0 or 1) is the master

Device Properties Default Values:

Property Name	Default Values
SSAOProxy	No default value
SSAIProxy	No default value
FlowMax	No default value
MasterNumber	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
gasFlow0Percent: Flow in % of the debimeter 0	DEV_DOUBLE	READ_WRITE	No
gasFlow0LMin: Gas flow of the debimeter 0 in L/Min	DEV_DOUBLE	READ_WRITE	No
gasFlow1Percent: Flow in % of the debimeter 1	DEV_DOUBLE	READ_WRITE	No
gasFlow1LMin: Gas flow of the debimeter 1 in L/Min	DEV_DOUBLE	READ_WRITE	No
actualMode: Show what is the actual mode : Independent or MasterSlave	DEV_STRING	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
IndependentMode	DEV_VOID	DEV_VOID
MasterSlaveMode	DEV_VOID	DEV_VOID
Off	DEV_USHORT	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 - IndependentMode

- **Description:** The two debimeters are controlled independently
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**

5 - MasterSlaveMode

- **Description:** The two debimeters are in Master Slave mode
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**

6 - Off

- **Description:** Switch OFF the specified debimeter
- **Argin:**
DEV_USHORT : debimeter number to switch OFF (0 or 1)
- **Argout:**
DEV_VOID : no argout

- **Command allowed for:**

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
--	--