



TANGO
Device
Server

SY2527Crate User's Guide

SY2527Crate Class

Revision: release_1_2_3 - Author: vince_soleil
Implemented in C++ - CVS repository: tango-ds

Introduction:

This device controls the high voltage power supply boards fixed in the SY2527 crate. the hardware is manufactured by the "CAEN "company. The used communication protocole is TCP/IP. On device can be seen as 1 HV power supply (PS) system consisted of several PS boards. Each Board is consisted of 12 Channels (PS board types : A1733; A1833A)

Class Inheritance:

- Tango::Device_4Impl
 - SY2527Crate

Class Description:

This device is the SY2527 tango interface. it controls the high voltage power supply boards fixed in the SY2527 crate. it uses a TCP/IP communication

Properties:

Device Properties		
Property name	Property type	Description
Host	Tango::DEV_STRING	SY2527 IP adress
HvPowerSupplyName	Tango::DEV_STRING	an HV power supply name is consisted of severals boards, each one locked in different slots
SlotList	Array of short	this list contains the used slots in the HV Power Supply
DefaultVoltagesList	Array of string	each element of the list is a default voltage configuration. One configuration is consists of 3 fields according to the following pattern: SlotNumer; ChannelNumber;DefaultValue; don't forget the ' ; ' character at the end of string
RampVoltage	Tango::DEV_DOUBLE	ramp up & down voltaages are specified by assigning this property

Device Properties Default Values:

Property Name	Default Values
Host	No default value
HvPowerSupplyName	No default value
SlotList	No default value
DefaultVoltagesList	No default value
RampVoltage	No default value

There is no Class properties.

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
GetVoltageOnChannel	DEVVAR_USHORTARRAY	DEV_DOUBLE
GetMeasVoltageOnChannel	DEVVAR_USHORTARRAY	DEV_DOUBLE
SetVoltageOnChannel	DEVVAR_DOUBLEARRAY	DEV_VOID
SetVoltageOnAllChannels	DEV_DOUBLE	DEV_VOID
PowerOnOverAllChannels	DEV_VOID	DEV_VOID
PowerOffOverAllChannels	DEV_VOID	DEV_VOID
PowerOnChannel	DEVVAR_USHORTARRAY	DEV_VOID
PowerOffChannel	DEVVAR_USHORTARRAY	DEV_VOID
GetChannelStatus	DEVVAR_USHORTARRAY	DEV_USHORT
SetDefaultVoltageOnAllChannels	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 - GetVoltageOnChannel

- **Description:**
- **Argin:**
DEVVAR_USHORTARRAY : arg1 : slot number, arg2 : channel number
- **Argout:**
DEV_DOUBLE : preset voltage
- **Command allowed for:**

5 - GetMeasVoltageOnChannel

- **Description:**
- **Argin:**
DEVVAR_USHORTARRAY : arg1 : slot number, arg2 : channel number
- **Argout:**
DEV_DOUBLE : measure voltage
- **Command allowed for:**

6 - SetVoltageOnChannel

- **Description:**
- **Argin:**
DEVVAR_DOUBLEARRAY : arg1 : slot number, arg2 : channel number, arg3: preset voltage
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**

7 - SetVoltageOnAllChannels

- **Description:**
- **Argin:**
DEV_DOUBLE : preset voltage to apply on all channels
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**

8 - PowerOnOverAllChannels

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

9 - PowerOffOverAllChannels

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :

- **Command allowed for:**

10 - PowerOnChannel

- **Description:**
- **Argin:**
DEVVAR_USHORTARRAY : arg 1: slot number, arg 2: channel number
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**

11 - PowerOffChannel

- **Description:**
- **Argin:**
DEVVAR_USHORTARRAY : arg 1: slot number, arg 2: channel number
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**

12 - GetChannelStatus

- **Description:**
- **Argin:**
DEVVAR_USHORTARRAY : arg1 : slot number, arg2 : channel number
- **Argout:**
DEV_USHORT : status byte
- **Command allowed for:**

13 - SetDefaultVoltageOnAllChannels

- **Description:** set Default Voltage On All Channels defined in properties
- **Argin:**
DEV_VOID : nothing

- **Argout:**
DEV_VOID : nothing
- **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
---	--