



TANGO
Device
Server

XbpmBeamLine User's Guide

XbpmBeamLine Class

Revision: release_1_2_2 - Author: vince_soleil
Implemented in C++ - CVS repository: tango-ds

Introduction:

it calculates the vertical and horizontal positions of the beam: $X = f(I_a, I_b, I_c, I_d, G, \text{offsets}, \text{factors} \dots)$ where: I_a = generated current on blade A I_b = generated current on blade B I_c = generated current on blade C I_d = generated current on blade D G = locum4 amplifier gain and offset & factors ...

Class Inheritance:

- Tango::Device_4Impl
 - XbpmBeamLine

Class Description:

it calculates the vertical and horizontal positions of the beam: $X = f(I_a, I_b, I_c, I_d, G, \text{offsets}, \text{factors} \dots)$ where: I_a = generated current on blade A I_b = generated current on blade B I_c = generated current on blade C I_d = generated current on blade D G = locum4 amplifier

Properties:

Device Properties		
Property name	Property type	Description
Locum4ProxyName	Tango::DEV_STRING	Locum4 Proxy Name
Geometry	Tango::DEV_USHORT	you have to choose between 2 different modes, either insertion or bipolar mode. 2 possible values - 1 : insertion - 2 : bipolar
LowVoltageThreshold	Tango::DEV_DOUBLE	if the input voltages of the ADC decrease this threshold, the corresponding average current attributes become ALARM (ex : if Vmes < low_threshold)
HighVoltageThreshold	Tango::DEV_DOUBLE	if the input voltages of the ADC exceed this threshold, the corresponding average current attributes become ALARM (ex : if Vmes > threshold)
CurrentOffset0	Tango::DEV_DOUBLE	current offset on channel 1
CurrentOffset1	Tango::DEV_DOUBLE	current offset on channel 2
CurrentOffset2	Tango::DEV_DOUBLE	current offset on channel 3
CurrentOffset3	Tango::DEV_DOUBLE	current offset on channel 4
VoltageOffset0	Tango::DEV_DOUBLE	voltage offset on channel 1
VoltageOffset1	Tango::DEV_DOUBLE	Voltage Offset on channel 2
VoltageOffset2	Tango::DEV_DOUBLE	Voltage Offset on channel3
VoltageOffset3	Tango::DEV_DOUBLE	Voltage Offset on channel 4
VerticalPositionFactor	Tango::DEV_DOUBLE	Vertical Position Factor
VerticalPositionOffset	Tango::DEV_DOUBLE	Vertical Position Offset
HorizontalPositionFactor	Tango::DEV_DOUBLE	Horizontal Position Factor
HorizontalPositionOffset	Tango::DEV_DOUBLE	Horizontal Position Offset
IntensityThreshold	Tango::DEV_DOUBLE	current out of which the Measured Intensity has a meaning (in micro-A)
SaiControllerProxyName	Tango::DEV_STRING	saiControllerProxyName
StartAtInit	Tango::DEV_BOOLEAN	Launch command at Init. phase. [default = false]

Device Properties Default Values:

Property Name	Default Values
Locum4ProxyName	No default value
Geometry	1
LowVoltageThreshold	0.9
HighVoltageThreshold	9.9
CurrentOffset0	0
CurrentOffset1	0
CurrentOffset2	0
CurrentOffset3	0
VoltageOffset0	0
VoltageOffset1	0
VoltageOffset2	0
VoltageOffset3	0
VerticalPositionFactor	1
VerticalPositionOffset	0
HorizontalPositionFactor	1
HorizontalPositionOffset	0
IntensityThreshold	No default value
SaiControllerProxyName	No default value
StartAtInit	false

There is no Class properties.

States:

States	
Names	Descriptions
FAULT	causes may be: - DAQ hardware driver failure - a fatal error occurred
RUNNING	DAQ is running
STANDBY	device is up and ready to acquire data

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
quadrant1: average current calculated over N samples on channel 1	DEV_DOUBLE	READ	No
quadrant2: average current calculated over N samples on channel 2	DEV_DOUBLE	READ	No
quadrant3: average current calculated over N samples on channel 3	DEV_DOUBLE	READ	No
quadrant4: average current calculated over N samples on channel 4	DEV_DOUBLE	READ	No
intensity: tsum of the 4 quadrants	DEV_DOUBLE	READ	No
standardDeviationIntensity1: standard Deviation Intensity on channel 1	DEV_DOUBLE	READ	Yes
standardDeviationIntensity2: standard Deviation Intensity on channel 2	DEV_DOUBLE	READ	Yes
standardDeviationIntensity3: standard Deviation Intensity on channel 3	DEV_DOUBLE	READ	Yes
standardDeviationIntensity4: standard Deviation Intensity on channel 4	DEV_DOUBLE	READ	Yes
horizontalPosition: horizontal position calculated as: $X = Xfactor * ((I1 + I4) - (I2 + I3)) / (sum) + Xoffset$ where: - $I1 = (V1 + V1offset) * gain + I1offset$ - $I2 = (V2 + V2offset) * gain + I2offset$ - $I3 = (V3 + V3offset) * gain + I3offset$ - $I4 = (V4 + V4offset) * gain + I4offset$	DEV_DOUBLE	READ	No
verticalPosition: vertical position calculated as: $Z = Zfactor * ((I1 + I2) - (I3 + I4)) / (sum) + Zoffset$ where: - $I1 = (V1 + V1offset) * gain + I1offset$ - $I2 = (V2 + V2offset) * gain + I2offset$ - $I3 = (V3 + V3offset) * gain + I3offset$ - $I4 = (V4 + V4offset) * gain + I4offset$	DEV_DOUBLE	READ	No
gain: the gain change according to the Locum4 input range: Input Range gain (micro_A) (A/V) 1000 100 100 10 10 1 1 0.1 0.1 0.01 0.01 0.001 0.001 0.0001 0.0001 0.00001	DEV_DOUBLE	READ	Yes
measurementUnit: displays the current measurement unit	DEV_STRING	READ	No
enableAutoRange: true: enable auto range false: disable auto range	DEV_BOOLEAN	WRITE	No

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
quadrant1Spectrum: samples acquired on channel corresponding to quadrant1	DEV_DOUBLE	1000000	No
quadrant2Spectrum: samples acquired on channel corresponding to quadrant2.	DEV_DOUBLE	1000000	No
quadrant3Spectrum: samples acquired on channel corresponding to quadrant 3	DEV_DOUBLE	1000000	No
quadrant4Spectrum: samples acquired on channel corresponding to quadrant4	DEV_DOUBLE	1000000	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
SetUnit	DEV_USHORT	DEV_VOID
Start	DEV_VOID	DEV_VOID
Stop	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::FAULT
- Tango::RUNNING
- Tango::STANDBY

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::FAULT
- Tango::RUNNING
- Tango::STANDBY

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::FAULT
- Tango::RUNNING
- Tango::STANDBY

4 - SetUnit

- **Description:** set beam intensity measurement unit

- **Argin:**

DEV_USHORT : 1: nA or 2 : microA or 3 : mA

- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::RUNNING
 - Tango::STANDBY

5 - Start

- **Description:** Start acquisitions in continuous mode
- **Argin:**
DEV_VOID : nothing
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::RUNNING
 - Tango::STANDBY

6 - Stop

- **Description:** Stop acquisitions
- **Argin:**
DEV_VOID : nothing
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::RUNNING
 - Tango::STANDBY

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