



TANGO
Device
Server

SIS3820 in Multi Channel Mode User's Guide

SIS3820MCS Class

Revision: - Author: tere29
Implemented in C++

Introduction:

Tango Class for the SIS3820 counter used in MultiChannelMode.

Class Inheritance:

- Tango::DeviceImpl
 - MCS
 - SIS3820MCS

Properties:

Device Properties		
Property name	Property type	Description
MaxNbChannels	Tango::DEV_LONG	Maximum number of channels en Multi Channel Scaler
MaxNbAcquisitions	Tango::DEV_LONG	Maximum number of acquisitions to be readout for every channel
Base	Tango::DEV_ULONG	VME base address. Default: 0x38000000 (939524096)
SimulationMode	Tango::DEV_ULONG	0 real mode, 1 simulation mode

Device Properties Default Values:

Property Name	Default Values
MaxNbChannels	No default value
MaxNbAcquisitions	No default value
Base	939524096
SimulationMode	No default value

There is no Class properties.

States:

States	
Names	Descriptions
ON	

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
NbChannels: Nb of channels to be readout	DEV_LONG	READ_WRITE	No
NbAcquisitions: Nb of acquisitions to be readout	DEV_LONG	READ_WRITE	No
Preset: MCS preset value	DEV_LONG	READ_WRITE	No
FlagMCSSet	DEV_LONG	READ	No

Image Attributes				
Attribute name	Data Type	X Data Length	Y Data Length	Expert
CountsArray: Readout of the MCS. It is one array of effective dimension NbChannel*NbAcquisitions. These has to be used for extracting the readout of a given channel. The counts for all channels in one acquisition are consecutive. For exemplo if the array values are: 1, 3, 5, 6, 2, 8, for 2 channels in 3 acquisitions, the readout of the first channel would be: 1, 5, 2 and of the second: 3, 6, 8.	DEV_LONG	32	10000	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
SetupMCS	DEV_VOID	DEV_LONG
ReadMCS	DEV_VOID	DEV_LONG

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::ON

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::ON

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::ON

4 - SetupMCS

- **Description:** Setup the counter to Multi Channel Scaler Mode
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_LONG : Completion status
- **Command allowed for:**

- Tango::ON

5 - ReadMCS

- **Description:** Read Multi Channel Scaler
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_LONG : Completion status
- **Command allowed for:**
 - Tango::ON

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