


TANGO Device Server

SaiController
User's Guide

SaiController Class

Revision: release_2_2_6 - Author: xavela
Implemented in C++

Introduction:

This device interfaces the SAI daq boards (2005;2010;2204;2205) the main functions are : - the daq board inititialization - the daq board configuration - the acquisition on N channels To use retrigger mode, you must set the triggerNumber attribute with a positive value. once the user is ready to start acquisition, he must push on start command

Class Inheritance:

- Tango::Device_3Impl
  - SaiController

Properties:

Device Properties		
Property name	**Property type**	**Description**
InputRange	Tango::DEV_STRING	The signal input range of the ADLINK board. Possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25.
BoardType	Tango::DEV_STRING	ADLink ADC board type SAI_2005. default value is SAI_2005
BoardNum	Tango::DEV_SHORT	ADLink ADC board identifier in the CPCI crate.
Timeout	Tango::DEV_DOUBLE	value out of which an exception is sent to inform the user that no acquired data is coming
DTRIGPolarity	Tango::DEV_STRING	2 possibilites: - RISING_EDGE - FALLING_EDGE
ChannelList	Array of short	list of channels to activate
GroundReference	Tango::DEV_STRING	differential or single_ended

Device Properties Default Values:

Property Name	**Default Values**
InputRange	U_10
BoardType	SAI_2005
BoardNum	0
Timeout	1000
DTRIGPolarity	RISING_EDGE
ChannelList	No default value
GroundReference	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
FAULT	- DAQ hardware driver failure
RUNNING	DAQ is running
STANDBY	Device waits for the start acquisition

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
frequency: DAQ board sampling frequency	DEV_DOUBLE	READ_WRITE	No
sampleNumber: sample number = Integration time * sample frequency	DEV_LONG	READ	No
integrationTime: buffer integration time	DEV_DOUBLE	READ_WRITE	No
triggerNumber: set to 0 at init, means that extern trigger mode disabled. if triggerNumber > 0, means extern trigger mode enabled. read part is the current trigger	DEV_LONG	READ_WRITE	No
timeoutCounter: this counter is incremented when no buffer generated after starting acquisition	DEV_LONG	READ	Yes

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
Start	DEV_VOID	DEV_VOID
Stop	DEV_VOID	DEV_VOID
On	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 - Start

- **Description:** start acquiring according to 2 modes: - if the triggerNumber attribute = 0, the daq board carries out an acquisition over N samples (sampleNumber attribute) where N = integrationTime x samplingFrequency. - if the triggerNumber attribut > 0, the daq board carries out X acquisitions (triggerNumber value). each one works like the same principle as above.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

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

5 - Stop

- **Description:** stop any running acquiisitions

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
- Tango::FAULT
- Tango::RUNNING
- Tango::STANDBY

6 - On

- **Description:** acquisition in continuous mode.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
- Tango::FAULT
- Tango::RUNNING
- Tango::STANDBY




TANGO Device Server

SaiController
User's Guide

SaiController Class

Revision: release_2_2_6 - Author: xavela
Implemented in C++

Introduction:

This device interfaces the SAI daq boards (2005;2010;2204;2205) the main functions are : - the daq board inititialization - the daq board configuration - the acquisition on N channels To use retrigger mode, you must set the triggerNumber attribute with a positive value. once the user is ready to start acquisition, he must push on start command

Class Inheritance:

- Tango::Device_3Impl
  - SaiController

Properties:

Device Properties		
Property name	**Property type**	**Description**
InputRange	Tango::DEV_STRING	The signal input range of the ADLINK board. Possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25.
BoardType	Tango::DEV_STRING	ADLink ADC board type SAI_2005. default value is SAI_2005
BoardNum	Tango::DEV_SHORT	ADLink ADC board identifier in the CPCI crate.
Timeout	Tango::DEV_DOUBLE	value out of which an exception is sent to inform the user that no acquired data is coming
DTRIGPolarity	Tango::DEV_STRING	2 possibilites: - RISING_EDGE - FALLING_EDGE
ChannelList	Array of short	list of channels to activate
GroundReference	Tango::DEV_STRING	differential or single_ended

Device Properties Default Values:

Property Name	**Default Values**
InputRange	U_10
BoardType	SAI_2005
BoardNum	0
Timeout	1000
DTRIGPolarity	RISING_EDGE
ChannelList	No default value
GroundReference	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
FAULT	- DAQ hardware driver failure
RUNNING	DAQ is running
STANDBY	Device waits for the start acquisition

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
frequency: DAQ board sampling frequency	DEV_DOUBLE	READ_WRITE	No
sampleNumber: sample number = Integration time * sample frequency	DEV_LONG	READ	No
integrationTime: buffer integration time	DEV_DOUBLE	READ_WRITE	No
triggerNumber: set to 0 at init, means that extern trigger mode disabled. if triggerNumber > 0, means extern trigger mode enabled. read part is the current trigger	DEV_LONG	READ_WRITE	No
timeoutCounter: this counter is incremented when no buffer generated after starting acquisition	DEV_LONG	READ	Yes

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
Start	DEV_VOID	DEV_VOID
Stop	DEV_VOID	DEV_VOID
On	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 - Start

- **Description:** start acquiring according to 2 modes: - if the triggerNumber attribute = 0, the daq board carries out an acquisition over N samples (sampleNumber attribute) where N = integrationTime x samplingFrequency. - if the triggerNumber attribut > 0, the daq board carries out X acquisitions (triggerNumber value). each one works like the same principle as above.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

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

5 - Stop

- **Description:** stop any running acquiisitions

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
- Tango::FAULT
- Tango::RUNNING
- Tango::STANDBY

6 - On

- **Description:** acquisition in continuous mode.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
- Tango::FAULT
- Tango::RUNNING
- Tango::STANDBY

Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client.
Link to Non-frame version.




TANGO Device Server

SaiController
Device Commands Description
SaiController Class

Revision: release_2_2_6 - Author: xavela

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 - Start

- **Description:** start acquiring according to 2 modes: - if the triggerNumber attribute = 0, the daq board carries out an acquisition over N samples (sampleNumber attribute) where N = integrationTime x samplingFrequency. - if the triggerNumber attribut > 0, the daq board carries out X acquisitions (triggerNumber value). each one works like the same principle as above.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

5 - Stop

- **Description:** stop any running acquiisitions

- **Argin:**

DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

6 - On

- **Description:** acquisition in continuous mode.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING
  - Tango::STANDBY

ESRF - Software Engineering Group