Infra Red Extraction mirror User’s Guide

IRMirror Class

Revision: IRMirror-Release_2_5 - Author: taurel
Implemented in C++ - CVS repository: ESRF

Introduction:

This class drives a mirror installed at the second dipole of cell 22. This mirror is used to deviate the IR part of the photon beam. The mirror is installed on a motor and has several thermoprobes which allows its positionning. This class is also used for the so-called "Fuse" device. This is exactly like the IR mirror but with only its "Slotted mode". A pre-processor variable called FUSE allows the Fuse device server generation.

Class Inheritance:

- Tango::Device_4Impl
  - IRMirror

Class Description:

This class drives a mirror installed at the second dipole of cell 22. This mirror is used to deviate the IR part of the photon beam. The mirror is installed on a motor and has several thermoprobes which allows its positionning. This class is also used for the so-called "Fuse" device. This is exactly like the IR mirror but with only its "Slotted mode". A pre-processor variable called FUSE allows the Fuse device server generation.

Properties:

Device Properties

Property name	Property type	Description
HM_LargeDownStep	Tango::DEV_LONG	The large step to move the mirror down in Half Mode (unit = um
HM_Delay	Tango::DEV_LONG	Delay between each control loop in Half Mode (unit = mS
HM_LowestPos	Tango::DEV_LONG	Lowest Mirror position in half mode (unit = um
HM_LowerLimitTemp	Tango::DEV_DOUBLE	Lower Temperature limit in Half Mode (unit = deg celcius
HM_UpperLimitTemp	Tango::DEV_DOUBLE	Upper limit temperature in Half Mode (unit = deg celcius
HM_FaultTemp	Tango::DEV_DOUBLE	Temperature threshold to switch the mirror to fault (unit = deg celcius
HM_StepUp	Tango::DEV_LONG	Step (up) size in Half Mode (unit = um
HM_StepDown	Tango::DEV_LONG	Step (down) size in Half Mode (unit = um
SM_BeamCenter	Tango::DEV_LONG	Theoritical beam center position (unit = um
SM_MinTemp	Tango::DEV_DOUBLE	Temperature threshold to start the mirror position control (unit = deg celcius
SM_HystTemp	Tango::DEV_DOUBLE	Temperature hysteresis in Slotted Mode (unit = deg celcius
SM_Step	Tango::DEV_LONG	Step size in Slotted Mode (unit = um
SM_Delay	Tango::DEV_SHORT	Control loop delay in Slotted mode (unit = mS
SM_FollowRange	Tango::DEV_LONG	Threashold to switch the mirror in FAULT in Slotted Mode (unit = um
T1_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T1 measurement
T2_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T2 measurement
T3_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T3 measurement
T4_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T4 measurement
T5_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T5 measurement
T6_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T6 measurement

T7_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T7 measurement
T8_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T8 measurement
T9_CorrFactor	Tango::DEV_DOUBLE	Correction factor for T9 measurement
Incoherency_low	Tango::DEV_DOUBLE	Low value for temperature incoherency
Incoherency_high	Tango::DEV_DOUBLE	High value for temperature incoherency
SR_CurrentLimit	Tango::DEV_DOUBLE	Storage ring current limit to move the mirror through the beam in Slotted Mode (unit = mA
SR_CurrentDevName	Tango::DEV_STRING	Storage ring beam current device name
HM_SmallDownStep	Tango::DEV_LONG	The small step to move the mirror down in Half Mode (unit = um
HM_InterlockTemp	Tango::DEV_DOUBLE	Temperarute hard-wired interlock in half mode
SM_UpperInterlockTemp	Tango::DEV_DOUBLE	Upper temperature hard-wired interlock in slotted mode
SM_LowerInterlockTemp	Tango::DEV_DOUBLE	Lower temperature hard wired interlock in slotted mode
DHDBDevName	Tango::DEV_STRING	Tango HDB DHDB device name
SR_CheckCurrent	Tango::DEV_STRING	Check if there is some beam in SR. Should be "yes" or "no"
HDB_enabled	Tango::DEV_STRING	Set to "yes" to send value history in HDB when an error is detected
Mirror_mode	Tango::DEV_STRING	Mode of the mirror. This property corresponds to the command which is executed when the Insert command is called. Possible values are (case insensitive) HalfMode or SlottedMode.

Device Properties Default Values:

Property Name	**Default Values**
HM_LargeDownStep	No default value
HM_Delay	No default value
HM_LowestPos	No default value
HM_LowerLimitTemp	No default value
HM_UpperLimitTemp	No default value
HM_FaultTemp	No default value
HM_StepUp	No default value
HM_StepDown	No default value
SM_BeamCenter	No default value
SM_MinTemp	No default value
SM_HystTemp	No default value
SM_Step	No default value

SM_Delay	No default value
SM_FollowRange	No default value
T1_CorrFactor	No default value
T2_CorrFactor	No default value
T3_CorrFactor	No default value
T4_CorrFactor	No default value
T5_CorrFactor	No default value
T6_CorrFactor	No default value
T7_CorrFactor	No default value
T8_CorrFactor	No default value
T9_CorrFactor	No default value
Incoherency_low	No default value
Incoherency_high	No default value
SR_CurrentLimit	No default value
SR_CurrentDevName	No default value
HM_SmallDownStep	No default value
HM_InterlockTemp	No default value
SM_UpperInterlockTemp	No default value
SM_LowerInterlockTemp	No default value
DHDBDevName	No default value
SR_CheckCurrent	No default value
HDB_enabled	No default value
Mirror_mode	SlottedMode

There is no Class properties.

States:

States	
Names	**Descriptions**
OFF	The mirror is at its home position
MOVING	The mirror is moving
ON	The mirror position is permently monitored and adjusted
FAULT	The mirror is faulty
STANDBY	This state allows clien to manually move the mirror

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
T1: IR mirror first temperature used to control the mirror in Half Mode. Fuse class: This attribute does not exist	DEV_DOUBLE	READ	Yes
T2: R mirror second temperature used to control the mirror in Half Mode Fuse class: This attribute does not exist	DEV_DOUBLE	READ	Yes
T3: IR mirror third temperature used to control the mirror in Half Mode Fuse class: This attribute does not exist	DEV_DOUBLE	READ	Yes
T4: IR mirror upper side first temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
T5: IR mirror upper side second temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
T6: IR mirror upper side third temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
T7: IR mirror lower side first temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
T8: IR mirror lower side second temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
T9: IR mirror lower side third temperature used to control the mirror in Slotted Mode	DEV_DOUBLE	READ	Yes
THM: Average IR mirror temperature in Half mode Fuse class: This attribute does not exist	DEV_DOUBLE	READ	No
TU: Average upper side IR mirror temperature in Slotted Mode	DEV_DOUBLE	READ	No
TL: Average lower side IR mirror temperature in Slotted Mode	DEV_DOUBLE	READ	No
TDIFF: Temperature difference between upper and lower side of the IR mirror in Slotted Mode	DEV_DOUBLE	READ	No
Position: Mirror position	DEV_LONG	READ_WRITE	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
Reset	DEV_VOID	DEV_VOID
Home	DEV_VOID	DEV_VOID
HalfMode	DEV_VOID	DEV_VOID
SlottedMode	DEV_VOID	DEV_VOID
GetMode	DEV_VOID	DEV_SHORT
DevReadValues	DEV_VOID	DEVVAR_DOUBLEARRAY
Insert	DEV_VOID	DEV_VOID
Extract	DEV_VOID	DEV_VOID

Device Commands for Expert Level Only		
Command name	**Argument In**	**Argument Out**
Manual	DEV_VOID	DEV_VOID
InterlockLevel	DEV_VOID	DEVVAR_STRINGARRAY

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::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - 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::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - 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::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY

4 - Reset

- **Description:** Reset the mirror when it is faulty

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

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

5 - Home

- **Description:** Send the mirror to its home position

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::OFF
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY

6 - HalfMode

- **Description:** Set the mirror in Half mode. In this mode, only half of the mirror is used to deviate the IR. The mirror is moved down close to the photon beam. This command does not exist when compiled for the Fuse class

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::OFF

7 - SlottedMode

- **Description:** In this mode, the mirror is inserted into the beam. There is a hole in the mirror center and the X ray beam must pass through this hole. Only the IR which are not as collimated as the X ray will be deviated by the mirror. The mirror must not be inserted or extracted if there is a X ray beam. The Xray beam burns the mirror except if it passes through the mirror hole. The mirror position is permanently controlled to keep the X ray beam in the mirror hole.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

○ Tango::OFF

8 - Manual (for expert only)

- **Description:** Manually move the mirror. There is no automatic control of mirror position.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

○ Tango::OFF

9 - GetMode

- **Description:** Return the mirror mode

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_SHORT : Mirror mode

- **Command allowed for:**

○ Tango::OFF
○ Tango::MOVING
○ Tango::ON
○ Tango::STANDBY

10 - InterlockLevel (for expert only)

- **Description:** Return the hard wired interlocks level

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_STRINGARRAY : HW interlocks level

- **Command allowed for:**
  - Tango::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY

11 - DevReadValues

- **Description:** This command has been implemented only for compatibility with the TACO HDB system. It is necessary until with have a Tango HDB. It allows the Taco poller to retrieve data which need to be archived

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_DOUBLEARRAY : Four attributes values ,Mirror position, HM average temp, SM upper average temp and SM lower average temp

- **Command allowed for:**
  - Tango::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY

12 - Insert

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY

13 - Extract

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::OFF
  - Tango::MOVING
  - Tango::ON
  - Tango::FAULT
  - Tango::STANDBY