TANGO
Device
Server

AdcMotor
User's Guide

AdcMotor Class

Revision: AdcMotor-Release_1_4_0 - Author: bourtemb
Implemented in C++ - CVS repository: ESRF

Introduction:

This class allows to control a motor. The position can be read directly from the motor counter or from an adc entry connected to the motor.

Class Inheritance:

- Tango::Device_4Impl
  - AdcMotor

Class Description:

Properties:

Device Properties		
Property name	**Property type**	**Description**
Motor_name	Tango::DEV_STRING	name of the associated motor object.
Adc_name	Tango::DEV_STRING	Name of the associated adc device.
Trust	Tango::DEV_STRING	MOTOR --> trust on the motor counter. SENSOR --> trust on the adc value read.
Open_pos	Tango::DEV_DOUBLE	Open position. When the motor is at that position, the state will be Tango::OPEN.
Close_pos	Tango::DEV_DOUBLE	Close position. When the motor is at that position, the state will be Tango::CLOSE.
Tolerance	Tango::DEV_DOUBLE	Tolerance on the position measured leading to a Tango::ALARM state. This tolerance is also used to determine if we are in Tango::OPEN or Tango::CLOSE states. Set by default to 10 motor steps.
Calibrated	Tango::DEV_SHORT	When this property is different from 0, the motor is considered as calibrated and a certain number of attributes cannot be changed anymore. (e.g. close_pos and open_pos) The goal is to avoid undesired change when the calibration process has been performed.
Nbretries	Tango::DEV_SHORT	Number of retries to perform after a movement if the position given by the sensor is too different from the wanted position. (Only if trust=SENSOR).
VSensor_adc_name	Tango::DEV_STRING	adc device name for the VSensor Attribute.
Alarm_motor_sensor	Tango::DEV_BOOLEAN	if TRUST = SENSOR alarm_motor_sensor = true => the device is on alarm if there is Too much difference between the value given by the adc and the position given by the motor. alarm_motor_sensor = false => there is no alarm if there is too much difference between the value given by the adc and the position given by the motor.
Time_before_retry	Tango::DEV_LONG	Time in seconds after the last movement to wait before the next retry when trust = SENSOR.
Min_vsensor_authorized	Tango::DEV_DOUBLE	If the VSensor attribute (voltage given to the potentiometer) is under that value, any movement will be forbidden if trust = sensor.
Max_vsensor_authorized	Tango::DEV_DOUBLE	If the VSensor attribute (voltage given to the potentiometer) is above or equal to that value, any movement will be forbidden if trust = sensor.
Poszero	Tango::DEV_DOUBLE	0V read on the sensor (potentiometer) will correspond to that position.
Posmax	Tango::DEV_DOUBLE	The maximum possible value of the potentiometer (<=> VSensor value) will correspond to that position.

Device Properties Default Values:

Property Name	Default Values
Motor_name	No default value
Adc_name	No default value
Trust	No default value
Open_pos	No default value
Close_pos	No default value
Tolerance	No default value
Calibrated	No default value
Nbretries	No default value
VSensor_adc_name	No default value
Alarm_motor_sensor	No default value
Time_before_retry	0
Min_vsensor_authorized	9.0
Max_vsensor_authorized	10.0
Poszero	0
Posmax	10

There is no Class properties.

States:

States

Names	Descriptions
ON	Indicate that the motor is on but stopped.
OFF	Indicate that the motor is off.
MOVING	The motor is moving.
OPEN	The motor is at the position open_pos defined in property. |Position - Open_pos| <= Tolerance
CLOSE	The motor is at the position close_pos defined in property. |Position - Close_pos| <= Tolerance
ALARM	The motor indicates an alarm state for example has reached a limit switch. The device will be in alarm state also if |PositionWanted - Position| > Tolerance
FAULT	The potentiometer is not well supplied. Vsensor < min_vsensor_authorized or Vsensor >= max_vsensor_authorized or the motor is in FAULT state

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
Position: if Trust = Motor, this attribute returns the position read on the motor counter (=Motor_position) if Trust = Sensor, this attribute returns the position read via the Adc (=Sensor_position)	DEV_DOUBLE	READ_WRITE	No
Open_pos: Open position. The state will be Tango::OPEN when the motor will be at that position.	DEV_DOUBLE	READ_WRITE	Yes
Close_pos: close position. The state will be Tango::CLOSE when the motor will be at that position.	DEV_DOUBLE	READ_WRITE	Yes
Tolerance: Tolerance on the position measured leading to a Tango::ALARM state. This tolerance is also used to determine if we are in Tango::OPEN or Tango::CLOSE states. |Open_pos - Position| <= Tolerance => device in Tango::OPEN state |Close_pos - Position| <= Tolerance => device in Tango::CLOSE state |Wanted_pos - Position| <= Tolerance => device in Tango::ON state |Wanted_pos - Position| > Tolerance => device in Tango::ALARM state and if nbretries > 0 the server will try to move to the wanted position again.	DEV_DOUBLE	READ_WRITE	Yes
Sensor_position: Position read via the potentiometer. If VSensor_adc_name property is not defined, this attribute will return directly the value read on the Value attribute of the Adc_name device. if VSensor_adc_name property is defined, this attribute returns the adc_pos value calculated as follow: adc_pos = (posmax - poszero) * adc_val / VSensor + poszero where adc_val is the value of the Value attribute of the Adc_name device and VSensor is the valur of the VSensor attribute. posmin and poszero are device properties.	DEV_DOUBLE	READ	No
Motor_position: Position read on the motor counter (Position attribute of the Motor_name device)	DEV_DOUBLE	READ	No
Trust: MOTOR -> trust on motor counter => Position=Motor_position SENSOR -> trust on adc read value => Position=Sensor_position	DEV_STRING	READ_WRITE	Yes
VSensor: This is the voltage given to the potentiometer used to read the position. This value is used in the calcul of the Sensor_position. If VSensor_adc_name is not defined, this attribute will always be invalid.	DEV_DOUBLE	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
On	DEV_VOID	DEV_VOID
Off	DEV_VOID	DEV_VOID
GoHome	DEV_VOID	DEV_VOID
Open	DEV_VOID	DEV_VOID
Close	DEV_VOID	DEV_VOID
Abort	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::ON
○ Tango::OFF
○ Tango::MOVING
○ Tango::OPEN
○ Tango::CLOSE
○ Tango::ALARM
○ Tango::FAULT

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
  - Tango::OFF
  - Tango::MOVING
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

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
  - Tango::OFF
  - Tango::MOVING
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

4 - On

- **Description:** Enable the power of the motor.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::OFF
  - Tango::MOVING
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

5 - Off

- **Description:** Disable the power of the motor.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::OFF
  - Tango::MOVING
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

6 - GoHome

- **Description:** Request the motor to go to its home position defined as Home_pos in the properties.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

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

- Tango::OPEN
- Tango::CLOSE
- Tango::ALARM
- Tango::FAULT

7 - Open

- **Description:** Ask the motor to go to the open position defined in property (and expert attribute).

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

8 - Close

- **Description:** Ask the motor to go to the close position defined in property (and expert attribute).

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT

9 - Abort

- **Description:** Abort the current movement.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**
  - Tango::ON
  - Tango::OFF
  - Tango::MOVING
  - Tango::OPEN
  - Tango::CLOSE
  - Tango::ALARM
  - Tango::FAULT