



TANGO
Device
Server

Ministat125 User's Guide

Ministat125 Class

**Revision: release_1_2_3 - Author: pierrejoseph
Implemented in C++**

Introduction:

device which controls the ministat 125 HUBER thermal regulator. it is possible to set the preset temperature and to get the measured temperature. moreover, 2 kinds of regulation is possible. Extern (extern Temperature probe) or Intern.

Class Inheritance:

- Tango::Device_3Impl
 - Ministat125

Properties:

Device Properties		
Property name	Property type	Description
SerialProxyName	Tango::DEV_STRING	name of the Serial device proxy
LowSetPointLimit	Tango::DEV_DOUBLE	Low Set Point Limit
HighSetPointLimit	Tango::DEV_DOUBLE	High Set Point Limit
HardwarePollingTime	Tango::DEV_ULONG	one thread is specialised to send command to the Serialline. The elapsed time between to commant sent is specified by the HardwarePollingTime (ms) property

Device Properties Default Values:

Property Name	Default Values
SerialProxyName	No default value
LowSetPointLimit	No default value
HighSetPointLimit	No default value
HardwarePollingTime	4000

There is no Class properties.

States:

States	
Names	Descriptions
RUNNING	the regulation is running
FAULT	Communication with serial device is broken
STANDBY	the device is ON and is waiting for the start regulation

Attributes:

Scalar Attributes

Attribute name	Data Type	R/W Type	Expert
isRemote: true means Remote Control activated false means Local Control activated default value : Remote	DEV_BOOLEAN	WRITE	Yes
externProbeUsed: true means Regulation with external probe false means Regulation with internal probe default value : Intern	DEV_BOOLEAN	READ_WRITE	No
internalTemperature: intrenal probe measurement	DEV_DOUBLE	READ	No
externalTemperature: external probe measurement	DEV_DOUBLE	READ	No
setPointTemperature: set point temperature whatever the regulation mode	DEV_DOUBLE	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

Device Commands for Expert Level Only

Command name	Argument In	Argument Out
StartRegulation	DEV_VOID	DEV_VOID
StopRegulation	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::RUNNING
- 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::RUNNING
- 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::RUNNING
- Tango::FAULT
- Tango::STANDBY

4 - StartRegulation (for expert only)

- **Description:** start the thermal regulation

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

- **Command allowed for:**

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

5 - StopRegulation (for expert only)

- **Description:** stop the thermal regulation
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::RUNNING
 - Tango::FAULT
 - Tango::STANDBY

ESRF - Software Engineering Group



TANGO
Device
Server

Ministat125 User's Guide

Ministat125 Class

**Revision: release_1_2_3 - Author: pierrejoseph
Implemented in C++**

Introduction:

device which controls the ministat 125 HUBER thermal regulator. it is possible to set the preset temperature and to get the measured temperature. moreover, 2 kinds of regulation is possible. Extern (extern Temperature probe) or Intern.

Class Inheritance:

- Tango::Device_3Impl
 - Ministat125

Properties:

Device Properties		
Property name	Property type	Description
SerialProxyName	Tango::DEV_STRING	name of the Serial device proxy
LowSetPointLimit	Tango::DEV_DOUBLE	Low Set Point Limit
HighSetPointLimit	Tango::DEV_DOUBLE	High Set Point Limit
HardwarePollingTime	Tango::DEV_ULONG	one thread is specialised to send command to the Serialline. The elapsed time between to commant sent is specified by the HardwarePollingTime (ms) property

Device Properties Default Values:

Property Name	Default Values
SerialProxyName	No default value
LowSetPointLimit	No default value
HighSetPointLimit	No default value
HardwarePollingTime	4000

There is no Class properties.

States:

States	
Names	Descriptions
RUNNING	the regulation is running
FAULT	Communication with serial device is broken
STANDBY	the device is ON and is waiting for the start regulation

Attributes:

Scalar Attributes

Attribute name	Data Type	R/W Type	Expert
isRemote: true means Remote Control activated false means Local Control activated default value : Remote	DEV_BOOLEAN	WRITE	Yes
externProbeUsed: true means Regulation with external probe false means Regulation with internal probe default value : Intern	DEV_BOOLEAN	READ_WRITE	No
internalTemperature: intrenal probe measurement	DEV_DOUBLE	READ	No
externalTemperature: external probe measurement	DEV_DOUBLE	READ	No
setPointTemperature: set point temperature whatever the regulation mode	DEV_DOUBLE	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

Device Commands for Expert Level Only

Command name	Argument In	Argument Out
StartRegulation	DEV_VOID	DEV_VOID
StopRegulation	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::RUNNING
- 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::RUNNING
- 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::RUNNING
- Tango::FAULT
- Tango::STANDBY

4 - StartRegulation (for expert only)

- **Description:** start the thermal regulation

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

- **Command allowed for:**

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

5 - StopRegulation (for expert only)

- **Description:** stop the thermal regulation
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**
 - Tango::RUNNING
 - Tango::FAULT
 - Tango::STANDBY

ESRF - Software Engineering Group

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

Ministat125

Device Commands Description

Ministat125 Class

Revision: release_1_2_3 - Author: pierrejoseph

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 *init_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::RUNNING
 - 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::RUNNING
- 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::RUNNING
- Tango::FAULT
- Tango::STANDBY

4 - StartRegulation (for expert only)

- **Description:** start the thermal regulation

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

- **Command allowed for:**

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

5 - StopRegulation (for expert only)

- **Description:** stop the thermal regulation

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

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

ESRF - Software Engineering Group