



TANGO
Device
Server

MicroControleurRFBooster

User's Guide

RFMicroAmpli Class

Revision: release_1_0_2 - Author: elattaoui
Implemented in C++

Introduction:

This class ask the booster microcontroler to send all the values of the modules and the defaults if there are. The speed propertie is the maximum of the RS232 bus. The RSProxy : the DServer Serial to connect to.

Class Inheritance:

- Tango::Device_3Impl
 - RFMicroAmpli

Properties:

Device Properties		
Property name	Property type	Description
RSPProxy	Tango::DEV_STRING	The name of the DServer Serial which open the RS232 communication.
Timeout	Tango::DEV_SHORT	timeout (in ms) to read data on the serial line
Parity	Tango::DEV_STRING	parity could be - none - even - odd
Charlength	Tango::DEV_SHORT	Sets the new charlength. 0 = 8 bits 1 = 7 bits 2 = 6 bits 3 = 5 bits
Stopbits	Tango::DEV_SHORT	Number of stop bit(s) 0 = 1 bit 2 = 2 bits 1 = 1.5 bits (seems not supported)
Baudrate	Tango::DEV_ULONG	Speed of the transmission. At this moment the communication speed with the microcontroller is set to 57600.
Newline	Tango::DEV_SHORT	End of message Character used in particular by the DevSerReadLine command Default = 13
Nb_max_Tour	Tango::DEV_SHORT	Nombre maximum d'amplis presents
Nb_max_Dissipateur	Tango::DEV_SHORT	Nombre max de barres presentes par ampli
Nb_max_Module	Tango::DEV_SHORT	Nombre max de modules par barre

Device Properties Default Values:

Property Name	Default Values
RSPProxy	No default value
Timeout	No default value
Parity	No default value
Charlength	No default value
Stopbits	No default value
Baudrate	No default value
Newline	No default value
Nb_max_Tour	No default value
Nb_max_Dissipateur	No default value
Nb_max_Module	No default value

There is no Class properties.

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
GetModuleValue	DEVVAR_SHORTARRAY	DEVVAR_DOUBLEARRAY
ReadHardwareValues	DEV_VOID	DEV_VOID
ReadHardwareDefault	DEV_VOID	DEV_STRING
GetDefaultList	DEV_VOID	DEVVAR_STRINGARRAY
ClearDefaultList	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:**

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:**

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:**

4 - GetModuleValue

- **Description:** This command returns the report of the module(Courant 1&2 or Incidente & Reflechie Power) specified by its number and the number of its barre (which the module is one).
- **Argin:**
DEVVAR_SHORTARRAY : Tour,Dissipateur,Niveau and Module numbers
- **Argout:**
DEVVAR_DOUBLEARRAY : module values (2 values)
- **Command allowed for:**

5 - ReadHardwareValues

- **Description:** This command reads the hardware (MicroController) values and completes the *amplifier_values* array which is a Tango::DevVarCharArray.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**
- **Polled every 3000 ms**

6 - ReadHardwareDefault

- **Description:** This command reads the hardware (MicroController) default and completes the default_value array which is a Tango::DevVarCharArray.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_STRING : Module default if any, else No default
- **Command allowed for:**
- **Polled every 1000 ms**

7 - GetDefaultList

- **Description:** List of all defaults that occur : when read the i₁/₂Controller erase the default, so this command build a default list.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEVVAR_STRINGARRAY : Default list
- **Command allowed for:**

8 - ClearDefaultList

- **Description:** Clear the default list.
 - **Argin:**
DEV_VOID : no argin
 - **Argout:**
DEV_VOID : no argout
 - **Command allowed for:**
-

ESRF - Software Engineering Group



TANGO
Device
Server

MicroControleurRFBooster User's Guide

RFMicroAmpli Class

**Revision: release_1_0_2 - Author: elattaoui
Implemented in C++**

Introduction:

This class ask the booster microcontroler to send all the values of the modules and the defaults if there are. The speed propertie is the maximum of the RS232 bus. The RSProxy : the DServer Serial to connect to.

Class Inheritance:

- Tango::Device_3Impl
 - RFMicroAmpli

Properties:

Device Properties		
Property name	Property type	Description
RSPProxy	Tango::DEV_STRING	The name of the DServer Serial which open the RS232 communication.
Timeout	Tango::DEV_SHORT	timeout (in ms) to read data on the serial line
Parity	Tango::DEV_STRING	parity could be - none - even - odd
Charlength	Tango::DEV_SHORT	Sets the new charlength. 0 = 8 bits 1 = 7 bits 2 = 6 bits 3 = 5 bits
Stopbits	Tango::DEV_SHORT	Number of stop bit(s) 0 = 1 bit 2 = 2 bits 1 = 1.5 bits (seems not supported)
Baudrate	Tango::DEV_ULONG	Speed of the transmission. At this moment the communication speed with the microcontroller is set to 57600.
Newline	Tango::DEV_SHORT	End of message Character used in particular by the DevSerReadLine command Default = 13
Nb_max_Tour	Tango::DEV_SHORT	Nombre maximum d'amplis presents
Nb_max_Dissipateur	Tango::DEV_SHORT	Nombre max de barres presentes par ampli
Nb_max_Module	Tango::DEV_SHORT	Nombre max de modules par barre

Device Properties Default Values:

Property Name	Default Values
RSPProxy	No default value
Timeout	No default value
Parity	No default value
Charlength	No default value
Stopbits	No default value
Baudrate	No default value
Newline	No default value
Nb_max_Tour	No default value
Nb_max_Dissipateur	No default value
Nb_max_Module	No default value

There is no Class properties.

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
GetModuleValue	DEVVAR_SHORTARRAY	DEVVAR_DOUBLEARRAY
ReadHardwareValues	DEV_VOID	DEV_VOID
ReadHardwareDefault	DEV_VOID	DEV_STRING
GetDefaultList	DEV_VOID	DEVVAR_STRINGARRAY
ClearDefaultList	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:**

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:**

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:**

4 - GetModuleValue

- **Description:** This command returns the report of the module(Courant 1&2 or Incidente & Reflechie Power) specified by its number and the number of its barre (which the module is one).
- **Argin:**
DEVVAR_SHORTARRAY : Tour,Dissipateur,Niveau and Module numbers
- **Argout:**
DEVVAR_DOUBLEARRAY : module values (2 values)
- **Command allowed for:**

5 - ReadHardwareValues

- **Description:** This command reads the hardware (MicroController) values and completes the *amplifier_values* array which is a Tango::DevVarCharArray.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**
- **Polled every 3000 ms**

6 - ReadHardwareDefault

- **Description:** This command reads the hardware (MicroController) default and completes the default_value array which is a Tango::DevVarCharArray.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_STRING : Module default if any, else No default
- **Command allowed for:**
- **Polled every 1000 ms**

7 - GetDefaultList

- **Description:** List of all defaults that occur : when read the i₁/₂Controller erase the default, so this command build a default list.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEVVAR_STRINGARRAY : Default list
- **Command allowed for:**

8 - ClearDefaultList

- **Description:** Clear the default list.
 - **Argin:**
DEV_VOID : no argin
 - **Argout:**
DEV_VOID : no argout
 - **Command allowed for:**
-

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

MicroControleurRFBooster Device Commands Description RFMicroAmpli Class

Revision: release_1_0_2 - Author: elattaoui

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:**

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:**

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:**

4 - GetModuleValue

- **Description:** This command returns the report of the module(Courant 1&2 or Incidente & Reflechie Power) specified by its number and the number of its barre (which the module is one).
- **Argin:**
DEVVAR_SHORTARRAY : Tour,Dissipateur,Niveau and Module numbers
- **Argout:**
DEVVAR_DOUBLEARRAY : module values (2 values)
- **Command allowed for:**

5 - ReadHardwareValues

- **Description:** This command reads the hardware (MicroController) values and completes the *amplifier_values* array which is a Tango::DevVarCharArray.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**
- **Polled every 3000 ms**

6 - ReadHardwareDefault

- **Description:** This command reads the hardware (MicroController) default and completes the *default_value* array which is a Tango::DevVarCharArray.

- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_STRING : Module default if any, else No default
- **Command allowed for:**
- **Polled every 1000 ms**

7 - GetDefaultList

- **Description:** List of all defaults that occur : when read the i₁/₂Controller erase the default, so this command build a default list.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEVVAR_STRINGARRAY : Default list
- **Command allowed for:**

8 - ClearDefaultList

- **Description:** Clear the default list.
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**

ESRF - Software Engineering Group