



TANGO
Device
Server

RFModule User's Guide

RFModule Class

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

Introduction:

Ce device represente un module RadioFrequence . Ce module fait partie de l_amplificateur RF de l_anneau ou du booster.

Class Inheritance:

- Tango::Device_3Impl
 - RFModule

Properties:

Device Properties		
Property name	Property type	Description
Numero_Tour	Tango::DEV_SHORT	Numero de la tour scrutée par le microcontrôleur RF. Pour le Booster -> 1
Numero_Dissipateur	Tango::DEV_SHORT	Numero de la barre de l'ampli qui herberge le module
Numero_Niveau	Tango::DEV_SHORT	Numero du niveau dans la barre : 0 -> HAUT 1 -> BAS
Numero_Module	Tango::DEV_SHORT	Numero du module au sein de la barre qui l'herberge
AmpliRF_Proxy	Tango::DEV_STRING	the proxy on which the module will get data.

Device Properties Default Values:

Property Name	Default Values
Numero_Tour	No default value
Numero_Dissipateur	No default value
Numero_Niveau	No default value
Numero_Module	No default value
AmpliRF_Proxy	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
valeur_1	DEV_DOUBLE	READ	No
valeur_2	DEV_DOUBLE	READ	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
AckDefault	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 - AckDefault

- **Description:** Les defauts sont non persistants : Une fois lu depuis le i_c/2controleur RF, le defaut disparaît. Ici, le defaut est maintenu jusqu'a ce qu il soit acquitte !!!
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**

ESRF - Software Engineering Group



TANGO
Device
Server

RFModule User's Guide

RFModule Class

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

Introduction:

Ce device represente un module RadioFrequence . Ce module fait partie de l_amplificateur RF de l_anneau ou du booster.

Class Inheritance:

- Tango::Device_3Impl
 - RFModule

Properties:

Device Properties		
Property name	Property type	Description
Numero_Tour	Tango::DEV_SHORT	Numero de la tour scrutée par le microcontrôleur RF. Pour le Booster -> 1
Numero_Dissipateur	Tango::DEV_SHORT	Numero de la barre de l'ampli qui herberge le module
Numero_Niveau	Tango::DEV_SHORT	Numero du niveau dans la barre : 0 -> HAUT 1 -> BAS
Numero_Module	Tango::DEV_SHORT	Numero du module au sein de la barre qui l'herberge
AmpliRF_Proxy	Tango::DEV_STRING	the proxy on which the module will get data.

Device Properties Default Values:

Property Name	Default Values
Numero_Tour	No default value
Numero_Dissipateur	No default value
Numero_Niveau	No default value
Numero_Module	No default value
AmpliRF_Proxy	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
valeur_1	DEV_DOUBLE	READ	No
valeur_2	DEV_DOUBLE	READ	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
AckDefault	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 - AckDefault

- **Description:** Les defauts sont non persistants : Une fois lu depuis le i_c/2controleur RF, le defaut disparaît. Ici, le defaut est maintenu jusqu'a ce qu il soit acquitte !!!
- **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

RFModule

Device Commands Description

RFModule 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 - AckDefault

- **Description:** Les defaults sont non persistants : Une fois lu depuis le i_c/2controleur RF, le default disparaît. Ici, le default est maintenu jusqu'a ce qu'il soit acquitte !!!
- **Argin:**
DEV_VOID : no argin
- **Argout:**
DEV_VOID : no argout
- **Command allowed for:**

ESRF - Software Engineering Group