



TANGO
Device
Server

BruckerBTPS User's Guide

BruckerBTPS Class

**Revision: release_1_0_5 - Author: coquet
Implemented in C++**

Introduction:

Brucker Power Supply, used to control high magnetic fields at ODE Beamline 40/45 A, 120 V, 2 Tesla magnet control via RS232.

Class Inheritance:

- Tango::Device_3Impl
 - BruckerBTPS

Properties:

Device Properties		
Property name	Property type	Description
SerialProxy	Tango::DEV_STRING	serial line device proxy used to communicate with the PS
MaxCurrent	Tango::DEV_DOUBLE	maximum current for the power supply used to convert from user unit Ampere to DAC value
CurrentOffset	Tango::DEV_DOUBLE	offset if needed in Ampere
Accuracy	Tango::DEV_DOUBLE	deadband for ramping. when the current reaches the setpoint +/- accuracy, the state goes back to STANDBY
SignedADCValue	Tango::DEV_BOOLEAN	les registres DAC et ADC sont signés dans ce cas Signé -40A = 8000 0A = 0 = FFFF 40A = 7FFF dans le cas non signé -40A = 0 0A = 7FFF 40A = FFFF

Device Properties Default Values:

Property Name	Default Values
SerialProxy	No default value
MaxCurrent	No default value
CurrentOffset	No default value
Accuracy	0.0048828125
SignedADCValue	false

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
current: current in the PS	DEV_DOUBLE	READ_WRITE	No
stepSize: size of the steps for ramping in Amperes	DEV_DOUBLE	READ_WRITE	Yes
stepTime: time for each ramp step in millisecond	DEV_DOUBLE	READ_WRITE	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
Reset	DEV_VOID	DEV_VOID
On	DEV_VOID	DEV_VOID
Off	DEV_VOID	DEV_VOID
Stop	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 - Reset

- **Description:** resets faults on the PS
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - On

- **Description:** turns the PS ON
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - Off

- **Description:** turns the PS OFF
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - Stop

- **Description:** stops the ramp
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

ESRF - Software Engineering Group



TANGO
Device
Server

BruckerBTPS User's Guide

BruckerBTPS Class

Revision: release_1_0_5 - Author: coquet
Implemented in C++

Introduction:

Brucker Power Supply, used to control high magnetic fields at ODE Beamline 40/45 A, 120 V, 2 Tesla magnet control via RS232.

Class Inheritance:

- Tango::Device_3Impl
 - BruckerBTPS

Properties:

Device Properties		
Property name	Property type	Description
SerialProxy	Tango::DEV_STRING	serial line device proxy used to communicate with the PS
MaxCurrent	Tango::DEV_DOUBLE	maximum current for the power supply used to convert from user unit Ampere to DAC value
CurrentOffset	Tango::DEV_DOUBLE	offset if needed in Ampere
Accuracy	Tango::DEV_DOUBLE	deadband for ramping. when the current reaches the setpoint +/- accuracy, the state goes back to STANDBY
SignedADCValue	Tango::DEV_BOOLEAN	les registres DAC et ADC sont signés dans ce cas Signé -40A = 8000 0A = 0 = FFFF 40A = 7FFF dans le cas non signé -40A = 0 0A = 7FFF 40A = FFFF

Device Properties Default Values:

Property Name	Default Values
SerialProxy	No default value
MaxCurrent	No default value
CurrentOffset	No default value
Accuracy	0.0048828125
SignedADCValue	false

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
current: current in the PS	DEV_DOUBLE	READ_WRITE	No
stepSize: size of the steps for ramping in Amperes	DEV_DOUBLE	READ_WRITE	Yes
stepTime: time for each ramp step in millisecond	DEV_DOUBLE	READ_WRITE	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
Reset	DEV_VOID	DEV_VOID
On	DEV_VOID	DEV_VOID
Off	DEV_VOID	DEV_VOID
Stop	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 - Reset

- **Description:** resets faults on the PS
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - On

- **Description:** turns the PS ON
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - Off

- **Description:** turns the PS OFF
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - Stop

- **Description:** stops the ramp
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **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

BruckerBTPS

Device Commands Description

BruckerBTPS Class

Revision: release_1_0_5 - Author: coquet

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 - Reset

- **Description:** resets faults on the PS
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - On

- **Description:** turns the PS ON
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - Off

- **Description:** turns the PS OFF
- **Argin:**
DEV_VOID :
- **Argout:**

DEV_VOID :

- **Command allowed for:**

7 - Stop

- **Description:** stops the ramp
 - **Argin:**
DEV_VOID :
 - **Argout:**
DEV_VOID :
 - **Command allowed for:**
-

ESRF - Software Engineering Group