



**TANGO
Device
Server**

BMC_8 User's Guide

BMC_8 Class

**Revision: release_1_0_0 - Author: coquet
Implemented in C++**

Introduction:

handles the power supply LT1 Brucker BMC 8 controller, splits data for Port devices port 1 = LT1 dipole ports 2 to 8 : LT1 Qpoles Uses ProfibusProxy to communicate with ProfibusServer

Class Inheritance:

- Tango::Device_3Impl
 - BMC_8

Properties:

Device Properties		
Property name	Property type	Description
ProfibusServerName	Tango::DEV_STRING	ango name of the Profibus DeviceServer Serveur ProfibusServer avec gestion d'abonnement Default : tango/profibus/1
BoardNumber	Tango::DEV_LONG	number of the Profibus DP Hilsher card (from 0 to 3) default : 0
DPAddress	Tango::DEV_LONG	Pour l'esclave portant les alimentations principales DP Address of the power supply from 1 to 125 DO NOT USE 0 , 126,127 reserved for the system! Default : 3
Group	Tango::DEV_LONG	Sync/Freeze Group groupe de sync/freeze parametre de 1 to 8. Default : 1
InputOffset	Tango::DEV_LONG	Memory offset of input data (seen by the master) offset mémoire des entrées de l'esclave tel que défini dans le configurateur Profibus
InputLength	Tango::DEV_LONG	length in bytes of input data taille en octets de la totalité des entrées de l'esclave tel que défini dans le configurateur Profibus Ces données doivent être consécutives.
OutputOffset	Tango::DEV_LONG	Memory offset of output data (seen by the master) offset mémoire des sorties de l'esclave tel que défini dans le configurateur Profibus
OutputLength	Tango::DEV_LONG	length in bytes of output data taille en octets de la totalité des sorties de l'esclave tel que défini dans le configurateur Profibus Ces données doivent être consécutives.

Device Properties Default Values:

Property Name	Default Values
ProfibusServerName	No default value
BoardNumber	No default value
DPAddress	No default value
Group	No default value
InputOffset	No default value
InputLength	No default value
OutputOffset	No default value
OutputLength	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
powerSupplyStatus: bit field of the BMC8 global status	DEV_SHORT	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
GetInputs	DEV_LONG	DEVVAR_LONGARRAY
SetCurrent	DEVVAR_LONGARRAY	DEV_VOID
On	DEV_LONG	DEV_VOID
Off	DEV_LONG	DEV_VOID
Reset	DEV_LONG	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 - GetInputs

- **Description:**
- **Argin:**
DEV_LONG : port number of the BMC8 1 to 8
- **Argout:**
DEVVAR_LONGARRAY : data of the port : current, voltage, setpoint, status
- **Command allowed for:**

5 - SetCurrent

- **Description:** writes the raw DAC value for current setpoint to the specified port Argin[0]= port number from 1 to 8
Argin[1]= value to write
- **Argin:**
DEVVAR_LONGARRAY : [0] port number, [1] DAC value to write
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - On

- **Description:** turns the power of the port number on
- **Argin:**
DEV_LONG : the port number
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - Off

- **Description:** turns the power of the port number off
- **Argin:**
DEV_LONG : the port number
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - Reset

- **Description:** resets the faults and alarms on the port number
- **Argin:**
DEV_LONG : the port number
- **Argout:**
DEV_VOID :
- **Command allowed for:**

ESRF - Software Engineering Group



TANGO
Device
Server

BMC_8

User's Guide

BMC_8 Class

Revision: release_1_0_0 - Author: coquet
Implemented in C++

Introduction:

handles the power supply LT1 Brucker BMC 8 controller, splits data for Port devices port 1 = LT1 dipole ports 2 to 8 : LT1 Qpoles Uses ProfibusProxy to communicate with ProfibusServer

Class Inheritance:

- Tango::Device_3Impl
 - BMC_8

Properties:

Device Properties		
Property name	Property type	Description
ProfibusServerName	Tango::DEV_STRING	tango name of the Profibus DeviceServer Serveur ProfibusServer avec gestion d'abonnement Default : tango/profibus/1
BoardNumber	Tango::DEV_LONG	number of the Profibus DP Hilsher card (from 0 to 3) default : 0
DPAddress	Tango::DEV_LONG	Pour l'esclave portant les alimentations principales DP Address of the power supply from 1 to 125 DO NOT USE 0 , 126,127 reserved for the system! Default : 3
Group	Tango::DEV_LONG	Sync/Freeze Group groupe de sync/freeze parametre de 1 to 8. Default : 1
InputOffset	Tango::DEV_LONG	Memory offset of input data (seen by the master) offset mémoire des entrées de l'esclave tel que défini dans le configurateur Profibus
InputLength	Tango::DEV_LONG	length in bytes of input data taille en octets de la totalité des entrées de l'esclave tel que défini dans le configurateur Profibus Ces données doivent être consécutives.
OutputOffset	Tango::DEV_LONG	Memory offset of output data (seen by the master) offset mémoire des sorties de l'esclave tel que défini dans le configurateur Profibus
OutputLength	Tango::DEV_LONG	length in bytes of output data taille en octets de la totalité des sorties de l'esclave tel que défini dans le configurateur Profibus Ces données doivent être consécutives.

Device Properties Default Values:

Property Name	Default Values
ProfibusServerName	No default value
BoardNumber	No default value
DPAddress	No default value
Group	No default value
InputOffset	No default value
InputLength	No default value
OutputOffset	No default value
OutputLength	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
powerSupplyStatus: bit field of the BMC8 global status	DEV_SHORT	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
GetInputs	DEV_LONG	DEVVAR_LONGARRAY
SetCurrent	DEVVAR_LONGARRAY	DEV_VOID
On	DEV_LONG	DEV_VOID
Off	DEV_LONG	DEV_VOID
Reset	DEV_LONG	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 - GetInputs

- **Description:**
- **Argin:**
DEV_LONG : port number of the BMC8 1 to 8
- **Argout:**
DEVVAR_LONGARRAY : data of the port : current, voltage, setpoint, status
- **Command allowed for:**

5 - SetCurrent

- **Description:** writes the raw DAC value for current setpoint to the specified port Argin[0]= port number from 1 to 8
Argin[1]= value to write
- **Argin:**
DEVVAR_LONGARRAY : [0] port number, [1] DAC value to write
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - On

- **Description:** turns the power of the port number on
- **Argin:**
DEV_LONG : the port number
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - Off

- **Description:** turns the power of the port number off
- **Argin:**
DEV_LONG : the port number
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - Reset

- **Description:** resets the faults and alarms on the port number
- **Argin:**
DEV_LONG : the port number
- **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

BMC_8

Device Commands Description

BMC_8 Class

Revision: release_1_0_0 - 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 - GetInputs

- **Description:**
- **Argin:**
DEV_LONG : port number of the BMC8 1 to 8
- **Argout:**
DEVVAR_LONGARRAY : data of the port : current, voltage, setpoint, status
- **Command allowed for:**

5 - SetCurrent

- **Description:** writes the raw DAC value for current setpoint to the specified port Argin[0]= port number from 1 to 8 Argin[1]= value to write
- **Argin:**
DEVVAR_LONGARRAY : [0] port number, [1] DAC value to write
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - On

- **Description:** turns the power of the port number on
- **Argin:**
DEV_LONG : the port number

- **Argout:**
DEV_VOID :

- **Command allowed for:**

7 - Off

- **Description:** turns the power of the port number off

- **Argin:**
DEV_LONG : the port number

- **Argout:**
DEV_VOID :

- **Command allowed for:**

8 - Reset

- **Description:** resets the faults and alarms on the port number

- **Argin:**
DEV_LONG : the port number

- **Argout:**
DEV_VOID :

- **Command allowed for:**

ESRF - Software Engineering Group