



**TANGO**  
Device  
Server

# **Bergoz DCCT controller User's Guide**

## **DCCTBergoz Class**

**Revision: release\_1\_0\_0 - Author: siraut  
Implemented in C++**

### **Introduction:**

Controls a Bergoz DCCT module

### **Class Inheritance:**

- Tango::Device\_3Impl
  - DCCTBergoz

### **Properties:**

## Device Properties

Property name	Property type	Description
<b>DIODeviceName</b>	Tango::DEV_STRING	Name of the underlying DIO device [DIO_7432 or compatible - no default value]
<b>DIODeviceRangeCtrlPins</b>	Array of string	Two pins must be specified in order to define the range (2A/200mA) : "RANGE/10" and "RANGE/100" Format of this property is : RANGE/10:PxY RANGE/100:PzW [No default value]
<b>DIODeviceTestCtrlPin</b>	Tango::DEV_STRING	Specifies the pin that is used to write the test mode (PxY). [No default value]
<b>DIODeviceRedLionCtrlLine</b>	Tango::DEV_STRING	DIO device's pin (PxY) associated with the DCCT range [no default value]
<b>FullScaleRange</b>	Tango::DEV_STRING	The DCCT range. Should be one of the following strings: "2A" or "200mA". No default value.
<b>TestMode</b>	Tango::DEV_STRING	Test mode can be active (set this property to ACTIVE) or inactive (set this property to INACTIVE)

### Device Properties Default Values:

Property Name	Default Values
DIODeviceName	No default value
DIODeviceRangeCtrlPins	No default value
DIODeviceTestCtrlPin	No default value
DIODeviceRedLionCtrlLine	No default value
FullScaleRange	No default value
TestMode	No default value

**There is no Class properties.**

### States:

States	
Names	Descriptions
<b>FAULT</b>	Can't control Bergoz DCCT module [may be due to a com. pb with DIO device]
<b>RUNNING</b>	Device is up and running [connected to DIO device]

## Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
<b>Range:</b> The DDCT full scale range in A. Can only be change using the dedicated property.	DEV_DOUBLE	READ	No
<b>TestModeInactive:</b> Test mode can be active (TestModeInactive = 0) or inactive (TestModeInactive = 1) . Can only be change using the dedicated property.	DEV_BOOLEAN	READ	No

## Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	Argument In	Argument Out
<b>Init</b>	DEV_VOID	DEV_VOID
<b>State</b>	DEV_VOID	DEV_STATE
<b>Status</b>	DEV_VOID	CONST_DEV_STRING
<b>GetDependencies</b>	DEV_VOID	DEVVAR_STRINGARRAY

### 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::FAULT
  - Tango::RUNNING

## 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::FAULT
  - Tango::RUNNING

## 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::FAULT
  - Tango::RUNNING

## 4 - GetDependencies

- **Description:**
- **Argin:**  
**DEV\_VOID** :
- **Argout:**  
**DEVVAR\_STRINGARRAY** : The list of devices on which this device depends
- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

---

**ESRF - Software Engineering Group**



**TANGO**  
Device  
Server

# **Bergoz DCCT controller User's Guide**

## **DCCTBergoz Class**

**Revision: release\_1\_0\_0 - Author: siraut  
Implemented in C++**

### **Introduction:**

Controls a Bergoz DCCT module

### **Class Inheritance:**

- Tango::Device\_3Impl
  - DCCTBergoz

### **Properties:**

## Device Properties

Property name	Property type	Description
<b>DIODeviceName</b>	Tango::DEV_STRING	Name of the underlying DIO device [DIO_7432 or compatible - no default value]
<b>DIODeviceRangeCtrlPins</b>	Array of string	Two pins must be specified in order to define the range (2A/200mA) : "RANGE/10" and "RANGE/100" Format of this property is : RANGE/10:PxY RANGE/100:PzW [No default value]
<b>DIODeviceTestCtrlPin</b>	Tango::DEV_STRING	Specifies the pin that is used to write the test mode (PxY). [No default value]
<b>DIODeviceRedLionCtrlLine</b>	Tango::DEV_STRING	DIO device's pin (PxY) associated with the DCCT range [no default value]
<b>FullScaleRange</b>	Tango::DEV_STRING	The DCCT range. Should be one of the following strings: "2A" or "200mA". No default value.
<b>TestMode</b>	Tango::DEV_STRING	Test mode can be active (set this property to ACTIVE) or inactive (set this property to INACTIVE)

### Device Properties Default Values:

Property Name	Default Values
DIODeviceName	No default value
DIODeviceRangeCtrlPins	No default value
DIODeviceTestCtrlPin	No default value
DIODeviceRedLionCtrlLine	No default value
FullScaleRange	No default value
TestMode	No default value

**There is no Class properties.**

### States:

States	
Names	Descriptions
<b>FAULT</b>	Can't control Bergoz DCCT module [may be due to a com. pb with DIO device]
<b>RUNNING</b>	Device is up and running [connected to DIO device]

## Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
<b>Range:</b> The DDCT full scale range in A. Can only be change using the dedicated property.	DEV_DOUBLE	READ	No
<b>TestModeInactive:</b> Test mode can be active (TestModeInactive = 0) or inactive (TestModeInactive = 1) . Can only be change using the dedicated property.	DEV_BOOLEAN	READ	No

## Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	Argument In	Argument Out
<b>Init</b>	DEV_VOID	DEV_VOID
<b>State</b>	DEV_VOID	DEV_STATE
<b>Status</b>	DEV_VOID	CONST_DEV_STRING
<b>GetDependencies</b>	DEV_VOID	DEVVAR_STRINGARRAY

### 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::FAULT
  - Tango::RUNNING

## 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::FAULT
  - Tango::RUNNING

## 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::FAULT
  - Tango::RUNNING

## 4 - GetDependencies

- **Description:**
- **Argin:**  
**DEV\_VOID** :
- **Argout:**  
**DEVVAR\_STRINGARRAY** : The list of devices on which this device depends
- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

---

**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

---

# Bergoz DCCT controller Device Commands Description DCCTBergoz Class

Revision: release\_1\_0\_0 - Author: siraut

## 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::FAULT
  - Tango::RUNNING

## 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::FAULT
- Tango::RUNNING

### 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::FAULT
- Tango::RUNNING

### 4 - GetDependencies

- **Description:**

- **Argin:**

**DEV\_VOID** :

- **Argout:**

**DEVVAR\_STRINGARRAY** : The list of devices on which this device depends

- **Command allowed for:**

- Tango::FAULT
- Tango::RUNNING

---

**ESRF - Software Engineering Group**