Driver API

### I.S1L.MCCS\_SPS.006: Monitoring & Control of Sub-racks

The Cabinet Management System is a three-level system corresponding to the related functions and hardware: Cabinet, Sub-rack and TPM. Each level manages the related hardware, provides the safety functionality and generates a status reports for upper level. All the elements within these levels are connected by a gigabit connection. Each board has a dedicated CPLD and micro-controller for safety functionality and implementing the start-up procedure.

The sub-rack management board and the associated backplane allow for the following operations:

* Manage power distribution and current monitoring
* Monitor voltages and temperatures
* Manage AC-DC power modules
* Initialise and configure the TPMs on the network
* Load the TPM configuration and execute the start-up sequence
* Execute the TPM stand-by/shut-off sequence
* Manage cooling
* Distribute synchronisation signals

#### Sub-rack management board commands

The list of commands which should be made available on a Sub-rack management board are defined below. MCCS will regularly poll the boards to update the value of the monitoring points (through the commands below). If certain values exceed an allowed range, MCCS can decide to shutdown down the board. The polling rate for all monitoring point specified below is not less than 1 second.

NOTE: The Sub-rack management board can be powered off remotely, however if it is off it can only be turned on through the Cabinet management board.

power\_off()

Power off subrack

power\_off\_tpm(tpm\_id)

Power off specified TPM

power\_on\_tpm(tpm\_id)

Power on specified TPM

get\_board\_temperature()

Get the temperature of the sub-rack management board

get\_backplane\_temperature()

Get the temperature of the backplane

get\_board\_current()

Get the current of the sub-rack management board

get\_tpm\_current(tpm\_id)

Get the current for the specified TPM as measured from the backplane

get\_tpm\_temperature(tpm\_id)

Get the temperature of the specified TPM as measured from the backplane

get\_tpm\_mac\_address(tpm\_id)

Get the MAC address of the specified TPM (that is, the TPM attached to the location associated with tpm\_id on the backplane. This is useful for hot-swapping TPMs, such that the MAC address of a new TPM can automatically be queried and added to the DHCP configuration on MCCS.

set\_fan\_speed(fan\_id, speed)

Set fan speed

get\_fan\_speed(fan\_id)

Get fan speed of specified fan

get\_power\_supply\_information(power\_supply\_id)

Get the input voltage level, output voltage level, internal temperature, internal fan speed and other power quality information of the specified power supply

get\_cooling\_information()

Return the input and output cold plate temperatures and speed, temperature of cabinet air and speed of the air inside the cabinet

configure\_tpms(tpm\_ids, firmware)

Program the specified TPMs with the specified firmware and perform initial configuration (such as configuring and starting the PLL, powering on the ADUs and pre-ADUs and start signal acquisition)

get\_synchronisation\_information()

Return the status of the PLL and 10 MHz signals (through lock status of PLL)

get\_network\_information()

Return the status of the network switch on the board, include switch port status and packet counters

TANGO Device

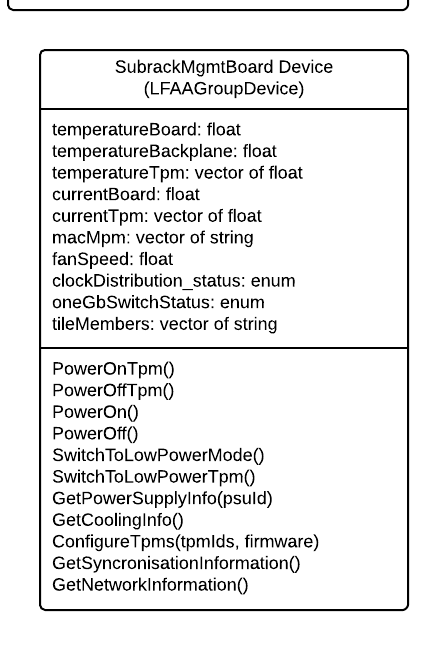


Figure 6‑1: Subrack Management Board device class diagram (inherits from LFAAGroupDevice)

The SubrackMgmtBoardDevice device is responsible for monitoring and controlling subrack boards. It is inherently an LFAAGroupDevice, as there are a number of associated member devices forming a subrack (tiles, switch). Main responsibilities are health monitoring of temperatures of the subrack, powering up and down of the unit, and switching to low power mode.

#### Element Behaviour

States and Modes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Range** | **Description and comments** |
| **adminMode**  **(read-write)** |  | Set by an outside authority (the Observatory operations via TM). |
| ONLINE | The subrack can be used for scientific observing. |
| MAINTENANCE | The subrack is not used for scientific observing but can be used for testing and commissioning. The LFAA is not aware of the higher observation goals and does not enforce this restriction; the LFAA executes commands received from TM. However, some test modes may be available only when the subrack is set to MAINTENANCE mode. |
| OFFLINE | The subrack is not used at all; when adminMode=OFFLINE, the operational state=DISABLE. |
| NOT\_FITTED | Set by operations to suppress alarm generation. |
| **opState**  **(read-only)** |  | LFAA reports the operational state for the subrack. |
| INIT | The subrack is being initialised, and the subrack device is checked for when process is complete. |
| OFF | The subrack is turned off. |
| ON | The subrack is turned on an has been initialized. |
| ALARM | The Quality Factor for at least one attribute is outside the pre-defined ALARM limits. Some or all functionality may not be available. |
| DISABLE | The subrack is administratively disabled (adminMode=OFFLINE, NOT\_FITTED, or RESERVE); only basic monitor and control functionality is available. |
| FAULT | An unrecoverable fault has been detected. The subrack is not available for use; maintainer/operator intervention is required. |
| UNKNOWN | The subrack is unresponsive, *e.g.*, due to loss of communication. |
| **healthState**  **(read-only)** | OK  DEGRADED  FAILED | The overall subrack healthState. |
| **obsState**  **(read-only)** |  | The subrack Observing State indicates status related to scan configuration and execution. |
| IDLE | The subrack is not being used for LFAA observations. |
| CONFIGURING | N/A |
| READY | The subrack enters READY when all subrack devices are ready to participate in an observation. |
| SCANNING | The subrack is currently in use for an observation. |
| PAUSED | N/A |
| ABORTED | N/A |
| FAULT | An unrecoverable error that requires operator intervention has been detected. |

Commands

|  |  |  |
| --- | --- | --- |
| **#** | **Switch Commands Required** | **Description** |
| **1** | PowerOnTpm(tpmId) | Powers on a TPM connected to the subrack |
| **2** | PowerOffTpm(tpmId) | Powers off a TPM connected to the subrack |
| **3** | PowerOn() | Powers on the subrack |
| **4** | PowerOff() | Powers off the subrack |
| **5** | SwitchToLowPowerMode() | Switches the entire subrack to low power mode |
| **6** | SwitchTpmToLowPowerMode() | Switches a connected TPM to low power mode |
| **7** | GetCoolingInformation() | Return the input and output cold plate temperatures and speed, temperature of cabinet air and speed of the air inside the cabinet |
| **8** | ConfigureTpms(tpmIds, firmware) | Program the specified TPMs with the specified firmware and perform initial configuration (such as configuring and starting the PLL, powering on the ADUs and start signal acquisition) |
| **9** | GetSynchronisationInformation() | Return the status of the PLL and 10 MHz signals (through lock status of PLL) |
| **10** | GetNetworkInformation() | Return the status of the network switch on the board, including switch port status and packet counters |

Alarms

|  |  |  |
| --- | --- | --- |
| **#** | **Alarm Condition** | **Description** |
| **1** | State = FAULT |  |