SKA sub-rack and cabinet API

for TANGO Devices

DRAFT

09-08-2019

# 

[Sub-rack and cabinet operating system 3](#_Toc16257140)

[Sub-rack API 3](#_Toc16257141)

[Abstraction of the observables 6](#_Toc16257142)

[Monitor functions 6](#_Toc16257143)

[Control functions 8](#_Toc16257144)

[Cabinet API 10](#_Toc16257145)

[Abstraction of the observables 13](#_Toc16257146)

[Monitor functions 13](#_Toc16257147)

[Control functions 15](#_Toc16257148)

# 

# Sub-rack and cabinet operating system

The operating system for all CPUs of the sub-rack and and cabinet management boards will be LINUX based:

* LINUX kernel: 4.9
* File system: Debian

All hardware registers will be mapped at file system level by a specific kernel driver.

# Sub-rack API

The sub-rack management board shall perform the following activities:

* 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

Monitor Capabilities

The sub-rack management board allows to monitor the following categories of parameters:

* FpgaFwVersionReg
* UserReg
* MCUReg
* UserLedReg
* HKeepRegs
* EthRegs
* Status register

Each category is composed by a set of registers mapped in the CPLD and accessible by a kernel driver.

|  |  |
| --- | --- |
| **Parameter category** | **Description** |
| FpgaFwVersionReg | Information about CPLD firmware version |
| UserReg | User registers |
| MCUReg | Registers containing physical quantities acquired by MCU (voltages and temperatures) |
| UserLedReg | Registers to manage user leds on board |
| HKeepRegs | Registers reporting information about backplane board |
| EthRegs | Registers containing information about the Ethernet interface configuration of the board |
| Status register | Register grouping all status and alarm flags of TPMs and sur-rack management board |

For each category the following table lists the registers and their description.

|  |  |  |
| --- | --- | --- |
| **Register Name** | **Category** | **Register Description** |
| regfile\_fw\_build\_low | FpgaFwVersionReg | Build Timestamp Low Part |
| fw\_build\_high | FpgaFwVersionReg | Build Timestamp High Part |
| regfile\_fw\_version | FpgaFwVersionReg | Firmware Version |
| regfile\_user\_reg0 | UserReg | User Register 0 |
| regfile\_user\_reg1 | UserReg | User Register 1 |
| regfile\_user\_reg2 | UserReg | User Register 2 |
| regfile\_user\_reg3 | UserReg | User Register 3 |
| McuFWBuildVersion | MCUReg | MCU Firmware Version |
| McuFWBuildTime | MCUReg | MCU Firmware Build time |
| McuFWBuildDate | MCUReg | MCU Firmware Build date |
| GPReg0 | MCUReg | MCU General Purpose User Register 0 |
| GPReg1 | MCUReg | MCU General Purpose User Register 1 |
| GPReg2 | MCUReg | MCU General Purpose User Register 2 |
| GPReg3 | MCUReg | MCU General Purpose User Register 3 |
| VoltageSOC | MCUReg | SOC Voltage Value |
| VoltageARM | MCUReg | ARM Voltage Value |
| VoltageDDR | MCUReg | DDR Voltage Value |
| Voltage2V5 | MCUReg | 2V5 Voltage Value |
| Voltage1V0 | MCUReg | 1V0 Voltage Value |
| Voltage1V1 | MCUReg | 1V1 Voltage Value |
| VoltageVCORE | MCUReg | VCORE Voltage Value |
| Voltage1V5 | MCUReg | 1V5 Voltage Value |
| Voltage3V3 | MCUReg | 3V3 Voltage Value |
| Voltage5V | MCUReg | 5V Voltage Value |
| Voltage3V | MCUReg | 3V Voltage Value |
| Voltage2V8 | MCUReg | 2V8 Voltage Value |
| BuckRegTemp | MCUReg | Buck Regulator Temp |
| MCUTemp | MCUReg | MCU Internal Temp |
| regfile\_led\_tpm\_k | UserLedReg | TPM LED K |
| regfile\_led\_tpm\_a | UserLedReg | TPM LED A |
| regfile\_led\_user\_k | UserLedReg | USER LED K |
| regfile\_led\_user\_a | UserLedReg | USER LED A |
| regfile\_present\_backplane | HKeepRegs | Backplane Present |
| regfile\_present\_tpm | HKeepRegs | TPMs Present |
| regfile\_hkt | HKeepRegs | House Keeping Temperature |
| regfile\_hkt\_temp2 | HKeepRegs | Temperature Alarm 2 |
| regfile\_hkt\_temp1 | HKeepRegs | Temperature Alarm 1 |
| regfile\_hkv | HKeepRegs | House Keeping Voltages |
| regfile\_hkv\_buck2\_pgood | HKeepRegs | Buck 2 Power Good |
| regfile\_hkv\_vi15\_pgood | HKeepRegs | 1V5 Power Good |
| regfile\_hkv\_buck1\_rsto | HKeepRegs | Buck 1 Reset Output (Active Low) |
| regfile\_hkv\_buck1\_pgood | HKeepRegs | Buck 1 Power Good |
| regfile\_hkv\_vin\_pgood | HKeepRegs | Step Down Power Good |
| regfile\_hkv\_buck1\_irq | HKeepRegs | Buck 1 IRQ |
| regfile\_hkv\_pwrin\_alert | HKeepRegs | Hot Swap Controller Pwrin Alert (Active Low) |
| regfile\_ethernet\_mac0\_low | EthRegs | MAC 0 Low Part |
| regfile\_ethernet\_mac0\_high | EthRegs | MAC 0 High Part |
| regfile\_ethernet\_mac1\_low | EthRegs | MAC 1 Low Part |
| regfile\_ethernet\_mac1\_high | EthRegs | MAC 1 High Part |
| regfile\_ethernet\_ip | EthRegs | IP |
| regfile\_ethernet\_mask | EthRegs | Mask |
| regfile\_ethernet\_gateway | EthRegs | Gateway |
| regfile\_UCP\_ucp\_rx\_dst\_port\_l | EthRegs | UCP RX destination UDP port Low |
| regfile\_UCP\_ucp\_rx\_dst\_port\_h | EthRegs | UCP RX destination UDP port High |
| regfile\_UCP\_ucplastpsn | EthRegs | UCP Last Psn |
| TBD | Status register | TBD |

### Abstraction of the observables

On the basis of the already developed library (ref. XXXXX, PYFABIL, Python FPGA Board Interacting Layer) for the management of TPM board, a Python package for the sub-rack will be defined.

This package will provide one Python class implementing methods for a structured access to the parameters listed in the previous paragraph.

This package could use, if necessary, the already defined PYFABIL package for some functions accessing the TPMs.

### Monitor functions

In this section a list of methods, for the class above defined, is presented. Theses functions are the API made available to implement commands for the related TANGO device through a appropriate wrapping.

For sub-rack TANGO device definition please refer to “SKA-TEL-LFAA-0600052-02\_SoftwareArchitectureDocument”.

As proposed by A. Magro a draft list of functions is reported below:

*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.

*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

*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

Control capabilities

The sub-rack management board allows to control the following items:

* TPM Power control
* Voltage and Temperature range settings
* Fan speed control
* Power supply control

|  |  |
| --- | --- |
| **Control category** | **Description** |
| TPM Power control | Manage power on and power off of TPM boards in the sub-rack |
| Voltage and Temperature range settings | Manage the threshold of each sensor |
| Fan speed control | Manage the fan speed for each fan |
| Power supply control | Manage the sub-rack power off |

### Control functions

In this section a list of methods for control operations is presented. Theses functions are the API made available to implement commands for the related TANGO device through a appropriate wrapping.

For sub-rack TANGO device definition please refer to “SKA-TEL-LFAA-0600052-02\_SoftwareArchitectureDocument”.

As proposed by A. Magro a draft list of functions is reported below:

*power\_off()*

Power off subrack

*power\_off\_tpm(tpm\_id)*

Power off specified TPM

*power\_on\_tpm(tpm\_id)*

Power on specified TPM

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

Set fan speed

*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).

# Cabinet API

Cabinet management system shall perform the following operations:

* manage the cabinet power distribution
* provide the network connection to the LFAA network and the cabinet network
* distribute synchronization to the sub-racks
* manage the cabinet cooling
* perform cabinet level diagnostic and interface to the lower level diagnostic.

The cabinet management board allows for the following operations:

* check the AC power quality
* startup the network interface to the LFAA network
* monitor the cabinet air temperature, the sub-rack air temperature and set the internal air exchanger fan speed accordingly.
* power up the stations and the related Ethernet switch
* manage the input AC supply transients and safe power-off
* check the presence of the input 10 MHz and PPS, input DPLL lock status
* measure the phase of PPS and execute a phase adjustment if needed
* check the DPLL status on the sub-racks
* check the PLL status on a TPMs (TBC)

Monitor capabilities

The cabinet management board allows to monitor the following categories of parameters:

* FpgaFwVersionReg
* UserReg
* MCUReg
* UserLedReg
* HKeepRegs
* EthRegs
* Status register

Each category is composed by a set of registers mapped in the CPLD and accessible by a kernel driver.

|  |  |
| --- | --- |
| **Parameter category** | **Description** |
| FpgaFwVersionReg | Information about CPLD firmware version |
| UserReg | User registers |
| MCUReg | Registers containing physical quantities acquired by MCU (voltages and temperatures) |
| UserLedReg | Registers to manage user leds on board |
| HKeepRegs | Registers reporting information about backplane board |
| EthRegs | Registers containing information about the Ethernet interface configuration of the board |
| Status register | Register grouping all status and alarm flags of TPMs and sur-rack management board |

For each category the following table lists the registers and their description.

|  |  |  |
| --- | --- | --- |
| **Register Name** | **Category** | **Register Description** |
| regfile\_fw\_build\_low | FpgaFwVersionReg | Build Timestamp Low Part |
| fw\_build\_high | FpgaFwVersionReg | Build Timestamp High Part |
| regfile\_fw\_version | FpgaFwVersionReg | Firmware Version |
| regfile\_user\_reg0 | UserReg | User Register 0 |
| regfile\_user\_reg1 | UserReg | User Register 1 |
| regfile\_user\_reg2 | UserReg | User Register 2 |
| regfile\_user\_reg3 | UserReg | User Register 3 |
| McuFWBuildVersion | MCUReg | MCU Firmware Version |
| McuFWBuildTime | MCUReg | MCU Firmware Build time |
| McuFWBuildDate | MCUReg | MCU Firmware Build date |
| GPReg0 | MCUReg | MCU General Purpose User Register 0 |
| GPReg1 | MCUReg | MCU General Purpose User Register 1 |
| GPReg2 | MCUReg | MCU General Purpose User Register 2 |
| GPReg3 | MCUReg | MCU General Purpose User Register 3 |
| VoltageSOC | MCUReg | SOC Voltage Value |
| VoltageARM | MCUReg | ARM Voltage Value |
| VoltageDDR | MCUReg | DDR Voltage Value |
| Voltage2V5 | MCUReg | 2V5 Voltage Value |
| Voltage1V0 | MCUReg | 1V0 Voltage Value |
| Voltage1V1 | MCUReg | 1V1 Voltage Value |
| VoltageVCORE | MCUReg | VCORE Voltage Value |
| Voltage1V5 | MCUReg | 1V5 Voltage Value |
| Voltage3V3 | MCUReg | 3V3 Voltage Value |
| Voltage5V | MCUReg | 5V Voltage Value |
| Voltage3V | MCUReg | 3V Voltage Value |
| Voltage2V8 | MCUReg | 2V8 Voltage Value |
| BuckRegTemp | MCUReg | Buck Regulator Temp |
| MCUTemp | MCUReg | MCU Internal Temp |
| regfile\_led\_tpm\_k | UserLedReg | TPM LED K |
| regfile\_led\_tpm\_a | UserLedReg | TPM LED A |
| regfile\_led\_user\_k | UserLedReg | USER LED K |
| regfile\_led\_user\_a | UserLedReg | USER LED A |
| regfile\_present\_backplane | HKeepRegs | Backplane Present |
| regfile\_present\_tpm | HKeepRegs | TPMs Present |
| regfile\_hkt | HKeepRegs | House Keeping Temperature |
| regfile\_hkt\_temp2 | HKeepRegs | Temperature Alarm 2 |
| regfile\_hkt\_temp1 | HKeepRegs | Temperature Alarm 1 |
| regfile\_hkv | HKeepRegs | House Keeping Voltages |
| regfile\_hkv\_buck2\_pgood | HKeepRegs | Buck 2 Power Good |
| regfile\_hkv\_vi15\_pgood | HKeepRegs | 1V5 Power Good |
| regfile\_hkv\_buck1\_rsto | HKeepRegs | Buck 1 Reset Output (Active Low) |
| regfile\_hkv\_buck1\_pgood | HKeepRegs | Buck 1 Power Good |
| regfile\_hkv\_vin\_pgood | HKeepRegs | Step Down Power Good |
| regfile\_hkv\_buck1\_irq | HKeepRegs | Buck 1 IRQ |
| regfile\_hkv\_pwrin\_alert | HKeepRegs | Hot Swap Controller Pwrin Alert (Active Low) |
| regfile\_ethernet\_mac0\_low | EthRegs | MAC 0 Low Part |
| regfile\_ethernet\_mac0\_high | EthRegs | MAC 0 High Part |
| regfile\_ethernet\_mac1\_low | EthRegs | MAC 1 Low Part |
| regfile\_ethernet\_mac1\_high | EthRegs | MAC 1 High Part |
| regfile\_ethernet\_ip | EthRegs | IP |
| regfile\_ethernet\_mask | EthRegs | Mask |
| regfile\_ethernet\_gateway | EthRegs | Gateway |
| regfile\_UCP\_ucp\_rx\_dst\_port\_l | EthRegs | UCP RX destination UDP port Low |
| regfile\_UCP\_ucp\_rx\_dst\_port\_h | EthRegs | UCP RX destination UDP port High |
| regfile\_UCP\_ucplastpsn | EthRegs | UCP Last Psn |
| TBD | Status register | TBD |

### Abstraction of the observables

On the basis of the already developed library (ref. XXXXX, PYFABIL, Python FPGA Board Interacting Layer) for the management of TPM board, a Python package for the sub-rack will be defined.

This package will provide one Python class implementing methods for a structured access to the parameters listed in the previous paragraph.

This package could use, if necessary, the already defined PYFABIL package for some functions accessing the TPMs.

### Monitor functions

In this section a list of methods, for the class above defined, is presented. Theses functions are the API made available to implement commands for the related TANGO device through a appropriate wrapping.

For cabinet TANGO device definition please refer to “SKA-TEL-LFAA-0600052-02\_SoftwareArchitectureDocument”.

As proposed by A. Magro, and modified by Sanitas, a draft list of functions is reported below:

*get\_cabinet\_temperature()*

Get the temperature of the cabinet management board

*get\_cabinet\_fan\_speed()*

Get the current speed of the internal exchanged 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\_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

Control capabilities

The cabinet management board allows to control the following items:

* Sub-racks and switches power control
* Voltage and Temperature range settings
* Fan speed control

|  |  |
| --- | --- |
| **Control category** | **Description** |
| TPM Power control | Manage power on and power off of TPM boards in the sub-rack |
| Voltage and Temperature range settings | Manage the threshold of each sensor |
| Fan speed control | Manage the fan speed for each fan |

### 

### Control functions

In this section a list of methods for control operations is presented. Theses functions are the API made available to implement commands for the related TANGO device through a appropriate wrapping.

For cabinet TANGO device definition please refer to “SKA-TEL-LFAA-0600052-02\_SoftwareArchitectureDocument”.

As proposed by A. Magro. and modified by Sanitas. a draft list of functions is reported below:

*power\_off()*

Power off the CMB

*reboot()*

Reboot the CMB

*power\_on\_switch (switch\_id)*

Power on 40G Ethernet switch

*power\_off\_switch (switch\_id)*

Power off 40G Ethernet switch

*power\_on\_subrack(subrack\_id, acdc\_id)*

Power on the specified sub-rack

*power\_off\_subrack(subrack\_id)*

Power off the specified sub-rack

*set\_cabinet\_fan\_speed(speed)*

Set the speed of the internal exchanged fan

*Set\_synchronization\_phase\_adjustment (TBD)*

Perform a phase adjustment between PPS and 10 MHz signals provided to sub-racks