SKA Subrack
Quick Guide

Written by

Cristian
Albanese

DOCUMENT CHANGE RECORD

Issue	Date	Paragraphs affected	Change information
1.0	31/03/2021	All	New document
1.1	21/05/2021	3.1	Added paragraph

TABLE OF CONTENTS

1. SCOPE

This document is aimed at providing indications for the initial usage of the main features of the SKA subracks by means of the on-board scripts.

SKA LFAA subrack provides the management operation to the LFAA TPM including:

- Power distribution, managed
- Clock and PPS distribution
- Gigabit network interface
- Initialization and detailed subrack level monitoring and safety capabilities
- Air forced cooling

The document deals with the following topics:

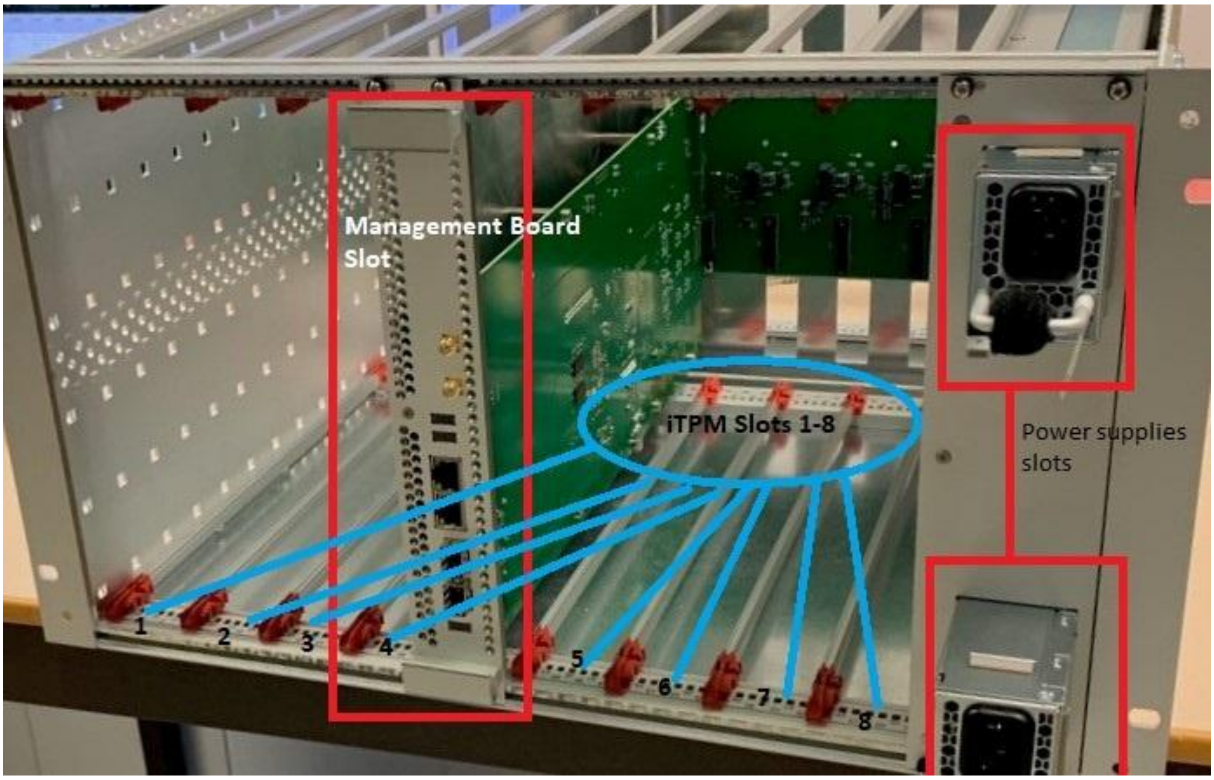
- Mechanical overview
- Cabling description
- Led description
- Access and control using delivered SW
- Mechanical installation notes

Note: the document is a quick guide and is not exhaustive: only the commands and instructions for the main operations are provided.

2. SUBRACK MECHANICAL OVERVIEW

The SKA LFAA subrack is a standard 19 inches, 6U size, the subrack is composed by the following elements:

1) Mechanical chassis, aluminum with EMI gasket
2) backplane board
3) 4x 50W fans, controlled by the backplane, fixed on the rear plate
4) 1 supply adapter with TWO standard removable power supplier modules
5) 1 slot for the Subrack Management Board, placed in the middle of the array
6) 8x slots for the LFAA iTPM boards, 4 for each side of the Management Board



To avoid mechanical damages during shipping, the subrack is delivered with power suppliers and boards not inserted, so the first operation before powering the subrack on, after its mechanical integrity check, is the Management Board insertion and fix, in the dedicated slot, the central slot, like in figure above.

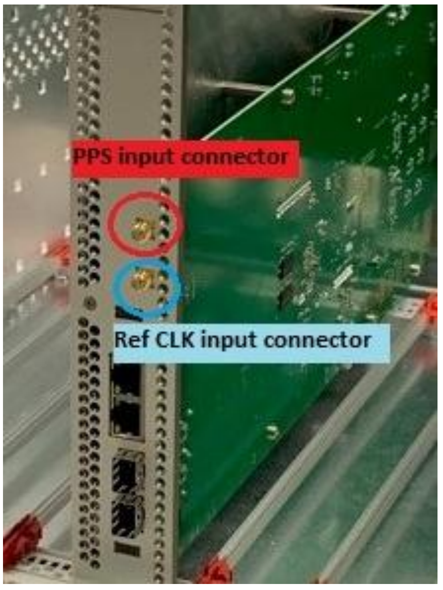
2.1 Cabling Description

2.1.1 Network Connection

The subrack access and use is possible only via Ethernet connection. All the control functionalities are developed by the Management Board, that is equipped with a CPU unit where a Linux distribution is running.
The available ETH connections to the subrack are 4, 2 via SFP fiber modules and 2 via copper Ethernet cable.
All the ETH connections are managed by two embedded switches available on board.

2.1.2 Reference Clock (10 MHz) and PPS connection

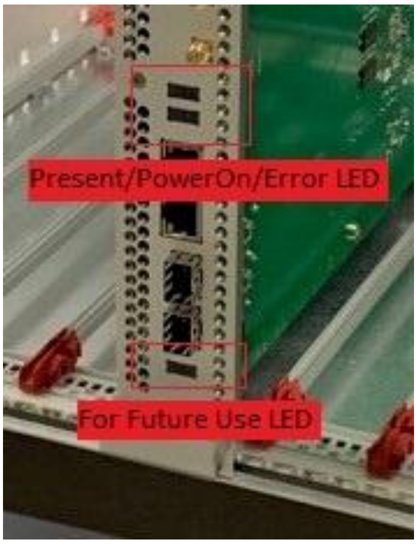
The Management Board distributes to iTPM boards the reference clock and the PPS signal, through the Backplane connectors.
These signals could be provided to the Management Board by two SMA connectors available on the front panel as shown in the figure below.



2.2 Led Description

On the Management Board front panel there are some LEDs, used to show some subrack states.
The 8 LEDs just below the PPS and Ref CLK connectors are used as present/powered/error for each iTPM slot.

Each LED (one for each iTPM slot) can be in the following state:
- blinking Red: ERROR
- Red: iTPM present in slot
- Green: iTPM present and powered on



The 4 LEDs on the bottom of the panel are controlled by CPLD on the Management Board and are reserved for future use.

3. MANAGEMENT COMMANDS OF SKA SUBRACK

For the subrack management board a set of Python API has been defined. It can be used by a proper TANGO device driver or by means of Python control scripts. Together with the API, some control scripts are also defined and stored on-board. These scripts can be used for the subrack remote control. Some indications for accessing the subrack and using the scripts are provided in the following sections.

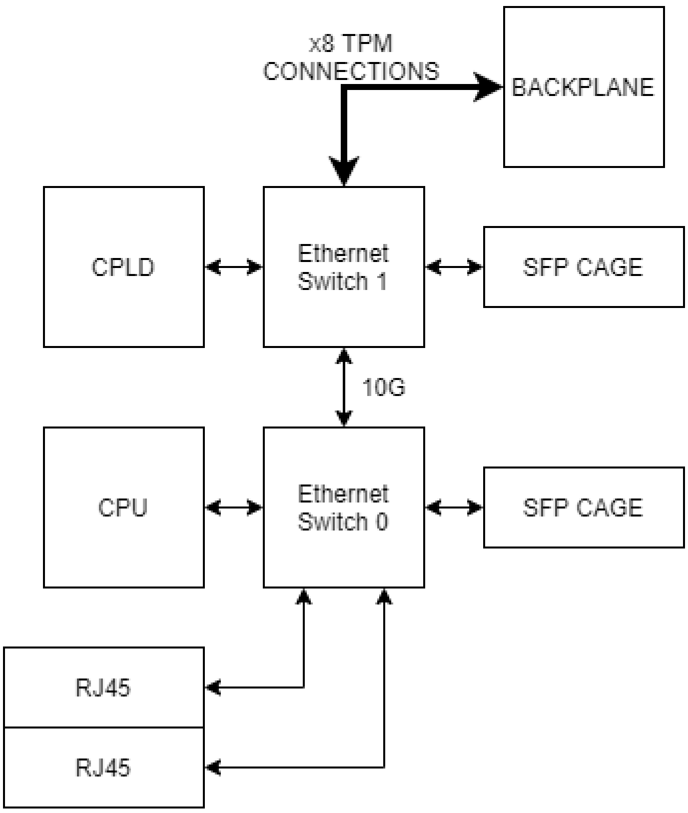
The subrack mounts a CPU running Linux DEBIAN 9, kernel 4.14.98 with a command line interface.

The subrack main functions are: monitoring some physical parameters (temperatures, currents, voltages and absorbed power of the TPM boards), controlling the subrack power status and adjusting the fan speed.

Note: The following commands and instructions can vary based on the delivered SW and API version. In particular this document issue is fully compliant with the SW and API versions currently available.

3.1 Ethernet configuration

The overall subrack network configuration is depicted in the following figure.



The main subrack functions are performed by the CPU while the CPLD performs lower level functions that are not directly accessible by the user and are out of the scope of this document.

Two IP addresses are associated to the CPU network interface: one IP belongs to the same subnet of the TPM boards, its value is 10.0.10.x where x depends on the specific Subrack Management Board. The other IP address can be configured as a DHCP IP address. It is not mandatory to have an IP address assigned by a DHCP server, however this can be useful to provide the CPU the ability to access the Internet. The CPU can be accessed via SSH using either the static IP or the DHCP configured IP.

The CPLD has an IP address in the 10.0.10.y subnet as well, however the CPLD cannot be accessed via SSH.

It has to be noted that at power-on the system boot waits for a CPU IP address assigned by a DHCP server until the end of a timeout, that currently lasts maximum 5 minutes. The CPU cannot be accessed before boot completion, while the CPLD will reply to a ping request even during the boot time. Therefore, receiving a ping reply from the CPLD IP address does not mean that the CPU has booted and ready to accept connections.

The IP address values configured on a specific subrack can be found in the corresponding test report: SKA_Subrack-Kit-XYZ_Test_Report.pdf.

The following table summarizes the current IP addresses configuration:

IP Address	Description	Value
CPU eth0	DHCP address	configurable
CPU eth0:1	Fixed address	10.0.10.x
CPLD	Fixed address	10.0.10.y

3.2 Defined User Id

Two user ids are implemented on the subrack: one user type account and a root account that has administrator privileges:

user: mnguser
password: SkaUser

user: root
password: skapassword

On the board there is a folder where are stored the scripts to control the subrack: here below a description on how to use it and some example of the main functions are provided.

Note: connecting as mnguser id is strongly suggested.

Connect to the subrack CPU with SSH using the following command (note: each subrack CPU has a predefined IP address on subnet 10.0.10.x, as shown in the previous section):

ssh mnguser@<subrack.ip>

once the connection is established a prompt like this should appear:

```
cristian.albanese@skahost:~$ ssh mnguser@10.0.10.96
mnguser@10.0.10.96's password:
Linux ska-management 4.14.98-0002-g297b88f2-dirty #43 SMP PREEMPT Fri Nov 13 10:16:28 CET 2020 armv7l
   _____ _  __     /\
  / ____| |/ /    /  \
 | (___ | ' /    / /\ \
  \___ \|  <    / /__\ \
  ____) | . \  / ______ \
 |_____/|_|\_\/_/      \_\

 SKA Management 1.1

------------------------------------------------------------------------

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 14 14:42:38 2021 from 10.0.10.1
mnguser@ska-management:~$
```

In the folder “/home/mnguser/SubrackMngApi-dev” there are API source and useful scripts.

3.3 API and Scripts folder structure

The structure of the folder “/home/mnguser/SubrackMngApi-dev” is the following :

```
mnguser@ska-management:~/SubrackMngAPI-dev$ ls -al
total 68
drwxr-xr-x 11 mnguser mnguser 4096 Apr  7 18:05 .
drwxr-xr-x  5 mnguser mnguser 4096 Apr  7 19:04 ..
drwxr-xr-x  4 mnguser mnguser 4096 Apr  7 17:23 build
-rw-r--r--  1 mnguser mnguser  683 Apr  6 15:33 Changelog
drwxr-xr-x  4 mnguser mnguser 4096 Apr  6 15:33 cpld_mng_api
-rw-r--r--  1 mnguser mnguser 1703 Apr  7 17:20 create_venv.py
drwxr-xr-x  2 mnguser mnguser 4096 Apr  7 17:23 dist
drwxr-xr-x  3 mnguser mnguser 4096 Apr  6 15:33 doc
drwxr-xr-x  2 mnguser mnguser 4096 Apr  7 17:42 .eggs
-rw-r--r--  1 mnguser mnguser 1402 Apr  6 15:33 .gitignore
-rw-r--r--  1 mnguser mnguser  131 Apr  7 18:05 install_packages.sh
-rw-r--r--  1 mnguser mnguser    0 Apr  6 15:33 LICENSE
-rw-r--r--  1 mnguser mnguser    0 Apr  6 15:33 NEWS
-rw-r--r--  1 mnguser mnguser  139 Apr  6 15:33 README.md
-rw-r--r--  1 mnguser mnguser  561 Apr  7 18:04 setup.py
drwxr-xr-x  3 mnguser mnguser 4096 Apr  6 15:33 subrack_mng_api
drwxr-xr-x  2 mnguser mnguser 4096 Apr  7 17:23 subrack_mng_api.egg-info
drwxr-xr-x  6 mnguser mnguser 4096 Apr  7 18:05 venv_py3
drwxr-xr-x  2 mnguser mnguser 4096 Apr  6 15:33 web_server
mnguser@ska-management:~/SubrackMngAPI-dev$
```

the API and scripts are developed to run under python3, and a virtual env python has been defined. All necessary libraries and packages are installed.

Here we have:

- **cpld_mng_api**: some low-level scripts used to access the CPLD internal registers, used by the high-level API, so the user should not use it.
- **web_server**: scripts that implements a web server, currently they are in develop and debug phase.
- **build**: compiled scripts used when the libraries are created for the installation in virtualenv.
- **venv_py3**: this is the virtualenv folder.
- **subrack_mng_api**: high level API and useful scripts. The script that has to be used to control and monitor the subrack is ‘subrack_monitor.py’: it can be used either to provide in formatted output (Table) the main status of the subrack, or to control and provide main functionalities of the subrack. The script has to be run within the ipython shell.

In the next sections monitor and control operations are described.

3.4 Virtual Environment initialization

A virtual env for python scripts was created: before using the scripts, it is necessary to enable it.

The following command enables the virtual env:

```
mnguser@ska-management:~/SubrackMngAPI-dev$ source venv_py3/bin/activate
(venv_py3) mnguser@ska-management:~/SubrackMngAPI-dev$
```

3.5 Subrack status monitor

Once the virtual env is enabled it is possible to monitor the subrack status by running the following commands:

```
(venv_py3) mnguser@ska-management:~/SubrackMngAPI-dev$ cd subrack_mng_api
(venv_py3) mnguser@ska-management:~/SubrackMngAPI-dev/subrack_mng_api$ sudo python subrack_monitor.py -s
[sudo] password for mnguser:
```

The following output will be shown:

```
+TPM Present & ON---+---------+---------+---------+---------+---------+---------+
| Board 1 | Board 2 | Board 3 | Board 4 | Board 5 | Board 6 | Board 7 | Board 8 |
+---------+---------+---------+---------+---------+---------+---------+---------+
| 0       | 1       | 0       | 0       | 0       | 0       | 0       | 0       |
| 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |
+---------+---------+---------+---------+---------+---------+---------+---------+
+Board Powers (Voltage, Current, Power)-+---------+---------+---------+---------+
| Board 1 | Board 2 | Board 3 | Board 4 | Board 5 | Board 6 | Board 7 | Board 8 | MEAS     |
+---------+---------+---------+---------+---------+---------+---------+---------+----------+
|  0.00   |  0.01   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   | Vin (V)  |
|  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   | Iin (A)  |
|  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   | Pin (W)  |
|         |         |         |         |         |         |         |         |          |
|  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   | Imax (A) |
|  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   |  0.00   | Pmax (W) |
+---------+---------+---------+---------+---------+---------+---------+---------+----------+
+RPM Fan Speed (Rpm) & PWM (%)--+
| Fan 1 | Fan 2 | Fan 3 | Fan 4 |
+-------+-------+-------+-------+
| 0     | 0     | 0     | 0     |
| 20%   | 20%   | 20%   | 20%   |
+-------+-------+-------+-------+
+Subrack Temperatures (Deg)---------+-------+
| Mng1  | Mng2  | Bckp1   | Bckp2   | CPU   |
+-------+-------+---------+---------+-------+
| 32.50 | 32.63 | 4095.94 | 4095.94 | 36.11 |
+-------+-------+---------+---------+-------+
+PSU Data+----------+----------+----------+
| PSU 1 | PSU 2    | TOT      | MEAS     |
+-------+----------+----------+----------+
| 12.03 | 128.00   |          | Vout (V) |
|  3.25 | 255.88   |          | Iout (A) |
| 39.09 | 32751.49 | 32790.58 | Pout (W) |
|       |          |          |          |
| 39.09 | 32751.49 | 32790.58 | Pmax (W) |
+-------+----------+----------+----------+
```

iTPM Present Flags

Itpm Powered OnFlag

First power on
or Backplane board turned off
Strange value can be showed

In the first two rows the iTPM boards inserted in the subrack slots together with their power status (row 1 TPM board present, row 2 TPM board power status) are reported.

The second table shows the current, voltages and absorbed power values together with the maximum reached values for currents and power.

The third table deals with the 4 fans of the subrack. For each of them the fan speed and the PWM percentage are reported.
In nominal function mode the fan speed is automatically adjusted by the MCU. If needed, a manual adjusting mode is available.

The last two sections of the table report the temperatures provided by sensors of the management board, backplane board and the CPU, and the information related to the two subrack power suppliers (voltage, current and absorbed power).

Some sensors are mounted on the backplane board, and as default at the subrack power-on backplane board is turned off, so these sensors can’t be reached, and in the monitor table undefined values (0 or maximum range type variable used to represent it) can be shown, this will be fixed fix in a future API.

4. SUBRACK CONTROL

The basic operations for using an iTPM board are the board powering on/off and the fan speed adjusting in order to avoid the board overheating.

In order to perform the above operations, the **subrack_monitor.py** script is used by means of ipython shell in order to access all the available APIs.

After the virtual environment is enabled (see section "Virtual Environment initialization"), the following command must be executed:

```
(venv_py3) mnguser@ska-management:~/SubrackMngAPI-dev/subrack_mng_api$ sudo ipython -i subrack_monitor.py
[sudo] password for mnguser:
```

and the ipython prompt will be shown:

```
root@ska-management:~/mng_cpu_script_pro_tango/subrack_mng_api# ipython2 -i -- subrack_monitor.p
y
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
pll res = 0x33

In [1]:
```

Since the defined device is "subrack", writing subrack followed by TAB all available API functions related to the subrack are listed.

```
In [1]: subrack.PowerOnTPM
          subrack.GetPSVout                 subrack.GetTPMPresent       subrack.PllInitialize
          subrack.GetSubrackTemperatures    subrack.GetTPMSupplyFault   subrack.powermon_cfgd
          subrack.GetTPMCurrent             subrack.GetTPMVoltage       subrack.PowerOffTPM
          subrack.GetTPMOnOffVect           subrack.Mng                 subrack.PowerOnTPM
          subrack.GetTPMPower               subrack.mode                subrack.read_tpm_singlewire
```

4.1 TPM board power-on

For powering-on a TPM board the following command is used:

subrack.PowerOnTPM(index)
index= TPM board slot index, 1..8

for example, to power-on a TPM board hosted in the slot 2:

If we try to power-on an iTPM not present in the selected slot an error message will be returned:

```
In [4]: subrack.PowerOnTPM(3)
2021-04-14 17:58:14,312 - ERROR - ERROR: TPM not present in selected slot
---------------------------------------------------------------------------
SubrackExecFault                          Traceback (most recent call last)
<ipython-input-4-a74c94b4cbf6> in <module>()
----> 1 subrack.PowerOnTPM(3)

/usr/local/lib/python2.7/dist-packages/subrack_mng_api-1.1-py2.7.egg/subrack_mng_api/subrack_management_board.pyc in PowerOnTPM(self,
, force)
    339                 self.Bkpln.pwr_on_tpm(tpm_slot_id)
    340             else:
--> 341                 raise SubrackExecFault("ERROR: TPM not present in selected slot")
    342         else:
    343             if self.Bkpln.get_bkpln_is_onoff()==0:

SubrackExecFault: ERROR: TPM not present in selected slot
```

4.2 TPM board power-off

For powering-off an iTPM board the following command is used:

subrack.PowerOffTPM(index)
index=TPM board slot index, 1..8

for example, to power-off a TPM board hosted in the slot 2:

```
In [5]: subrack.PowerOffTPM(2)
2021-04-14 17:59:03,920 - INFO - End I2C OP
2021-04-14 17:59:04,375 - INFO - End I2C OP

In [6]:
```

4.3 Fan speed setting

Two function modes are available for the fan speed adjusting: automatic and manual.
In automatic mode the MCU adjusts the fan speed depending on the number of TPM boards powered on. For some applications the current adjusting algorithm could not be enough and a manual operation can be required in order to properly set the fan speed.

The four fans are managed in couples: fans 1-2 and fans 3-4. Fans 1-2 cover subrack slots from 1 to 4, while fans 3-4 cover subrack slots from 5 to 8.
Acting on fan 1 adjusts the speed of both fans 1 and 2.
Acting on fan 3 adjusts the speed of both fans 3 and 4.

The command for setting the fan mode from automatic to manual for fan 1 and 2 is the following:

```
In [6]: subrack.SetFanMode(auto_mode=0,fan_id_blk=1)

In [7]:
```

The command for setting the fan mode from automatic to manual for fan 3 and 4 is the following:

```
In [7]: subrack.SetFanMode(auto_mode=0,fan_id_blk=3)

In [8]:
```

In order to change the mode from manual to automatic the command is the same but the auto_mode value shall be 1:

```
In [13]: subrack.SetFanMode(auto_mode=1,fan_id_blk=3)

In [14]:
```

The command for setting the fan speed of fan 1 and 2, for example at 20%, is the following:

```
In [9]: subrack.SetFanSpeed(fan_id=1,speed_pwm_perc=20)
```

Similarly, to set fan 3 and 4 at 80% the command is:

```
In [10]: subrack.SetFanSpeed(fan_id=3,speed_pwm_perc=80)
```

The commands to get the fan mode are the following:

```
In [12]: subrack.GetFanMode(fan_id=1)
Out[12]: 0
```

The returned mode is the same for fan 1 and 2 since the sensors are always managed in couple; similarly, for fan 3 and 4 getting the mode of fan 3 returns the mode of both 3 and 4.

The command to get the fan speed is:

```
In [14]: subrack.GetFanSpeed(fan_id=3)
Out[14]: (4643, 20.0)
```

this returns the speed in rpm (first value) and PWM percentage (second value).

5. MECHANICAL AND ELECTRICAL INSTALLATION NOTES

In this chapter some instructions and suggestions regarding the correct subrack installation are summarized.

5.1 Subrack unpackaging and first steps

After packaging removal, follow the following steps to proceed with installation:

1. Check the mechanical integrity
2. Fix the empty subrack in the cabinet with four screws, ensure that space for cooling in the rear and for the cabling is available
3. Insert the Power supply modules (one is enough up to 800W), check the lock mechanism
4. Insert the Subrack Management Board in the central slot. During the board insertion, check the correct connector alignment without forcing. The metallic plate is opposite to the Supply Module side.
5. Insert the iTPM board and fix the screws after insertion
6. Verify that the empty slots are closed by the metallic plate and fixed with their screws to allow the adequate air flow through the expected path
7. Plug the PPS and 10 MHz reference cable to the management board
8. Plug at least one of the Ethernet connection to the management board (RJ45 Gigabit cable, Cat5e or Gigabit optical SFP) to the local management network
9. Plug the AC supply cable and wait for some seconds before try to connect to the system
10. Based on the current Ethernet configuration (see par 3.1) at power-on the system boot waits for a CPU IP assigned by an DHCP server until the end of the configured timeout, currently maximum 5 minutes. The CPU cannot be accessed until completion of the Linux boot.

5.2 Installation recommendations

The subrack shall be installed in an air-conditioned cabinet with adequate rear space. Air cooling specification expect an air temperature between 18°C and 22°C circulating from the front to the rear of the subrack.
Noise reduction systems are also suggested to reduce the external noise level and allow human presence.
The high speed QSFP cable interface has been designed and verified for short range (1m-2.5m) passive cable to be directly connected to a close network switch. Active fibre cable has been verified too with a maximum of 3W of absorbed power.
In the current software release, only one SFP is configured to work at powerup, it is the SFP connector nearest the RJ45 ETH connector. The other SFP will be configured to work at system powerup in next SW release.

END OF DOCUMENT