TPM HARDWARE TEST USER MANUAL

Document Number..... NA
Document Type ..... NA
Revision .....A
Author ..... R. Chiello
Date 2021-05-06
Document Classification..... FOR PROJECT USE ONLY
Status.....Draft

Name	Designation	Affiliation	Signature	
Authored by:				
R.Chiello		University of Oxford		
			Date:	
Owned by:				
			Date:	
Approved by:				
			Date:	
Released by:				
			Date:	

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
A	2017-01-01	-	First draft release for internal review

DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor	MS Word	Word 2019	TPM_Hardware_test_user_manual.docx
Block diagrams			
Other			

ORGANISATION DETAILS

Name	SKA Organisation
Registered Address	Jodrell Bank Observatory Lower Withington Macclesfield Cheshire SK11 9DL United Kingdom Registered in England & Wales Company Number: 07881918
Fax.	+44 (0)161 306 9600
Website	www.skatelescope.org

TABLE OF CONTENTS

LIST OF FIGURES

No table of figures entries found.

LIST OF TABLES

No table of figures entries found.

LIST OF ABBREVIATIONS

TPM.....Tile Processing Module

AAVS.....Aperture Array Verification System

SKA .....Square Kilometre Array

SKAO .....SKA Project Office

GbE .....Gigabit Ethernet

1 Introduction

1.1 Purpose of the document

This document explains how to use the TPM hardware test suite included in the aavs-system software.

1.2 Scope of the document

This document refers specifically to the test suite included in the aavs-system software hosted on the SKA GitLab https://gitlab.com/ska-telescope/aavs-system/

2 References

2.1 Applicable documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.

[AD1] Applicable Document 1
[AD2] Applicable Document 2

2.2 Reference documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, **this document** shall take precedence.

[RD1] Reference Document 1
[RD2] Reference Document 2

3 Prerequisites

The following hardware setup should be available:

- A server running the aavs-system software.
- One or more TPMs, either 1.2 or 1.6 version, only homogeneous systems are supported at the moment.
- The server should be able to communicate with all TPMs using 1 GbE control network with MTU of 2000 bytes.
- A 40G network should connect the TPMs and suitable NIC on the server, the NIC should have IP address 10.0.X.Y, where X is in the range [3,254] and Y is in the range [1,254], netmask 255.255.0.0, MTU equal or larger than 9000 bytes. The IP address associated to the TPM 40G interfaces is 10.0.X.Y, where X is 1 or 2, depending on the FPGA, and Y is the last octet of the TPM IP address associated to its 1GbE interface, netmask is configured as 255.0.0.0
- 10 MHz reference clock and PPS should be connected to all TPMs.

The test suite runs on top of the aavs-system software, the following requirements should be satisfied in order to run the tests:

- aavs-system correctly installed, the distribution includes a deploy.sh script which takes care of all relevant dependencies
- YAML station configuration file in aavs-system/config directory reflecting the used hardware setup. To create a new station configuration file the user should refer to the existing default_config.yml as a template to create his own configuration file.

4 Configuring the test environment

The test setup is configured by a specific YAML configuration file which should be located in aavs-system/python/pyaavs/tests/config. The following parameter should be specified:

```
# DAQ Ethernet Interface
daq_eth_if: "eth2"

# Single TPM tests will be run on the TPM identified by index within the
station
single_tpm_test_station_idx: 0
```

`daq_eth_if` is the identifier of the NIC connected to the TPM 40GbE network.
`single_tpm_test_station_idx` identifies the TPM used for tests running on a single TPM, the TPM is identified by an index in the TPM lists defined in the station configuration file.

The user should refer to the existing test_config.yml as a template to create his own test configuration file, only the above two parameters can be customised by the user.

The test suite include a checking function to detect issues in the network configuration, specifically the following checks will be performed:

- When running tests that require station beam acquisition, the IP address associated to the NIC designated by `daq_eth_if` will be checked against the csp_ingest destination IP specified in the station configuration file. The NIC MTU will be checked to be >= 9000.
- When running test that require LMC data acquisition, the IP address associated to the NIC designated by `daq_eth_if` will be checked against the LMC destination IP specified in the station configuration file. The NIC MTU will be checked to be >= 2000.

- When running test that require integrated LMC data acquisition, the IP address associated to the NIC designated by `daq_eth_if` will be checked against the LMC integrated data destination IP specified in the station configuration file. The NIC MTU will be checked to be >= 2000.

5 Running the tests

The tests can be run either in interactive or non-interactive mode. In non-interactive mode a single python script executes all the available test and logs to file the test results. In interactive mode the user can select the test case to be run and configure the relevant test parameters. To start the tests execution in non-interactive mode the following command should be executed from command line in the aavs-system/python/pyaavs/tests folder:

```
python3 test_wrapper.py --config=../config/<station_config_file>.yml
                        --test_config=config/<test_config_file>.yml
                        --init
```

This command configures and initialises the TPMs using the bitfile specified in the station configuration file and executes all the available test cases using the default test parameters. Test results are logged into test_log/test_wrapper.log, a filtered version of the log file excluding debug messages and standard library messages is available in test_log/test_wrapper_filtered.log

Alternatively, executing

```
python3 test_wrapper.py --config=../config/<station_config_file>.yml
                        --test_config=config/<test_config_file>.yml
                        -i
```

starts the test script in interactive mode. In this case the user can select the test case to execute and perform basic operations on the defined station using a simple CLI based on menu, for instance station initialisation. Default test parameters can be modified via menu interface. Test results are logged into test specific log files stored in the test_log folder, for instance running the DAQ test creates a log file named test_daq.loq in test_log folder.

6 Available test cases

Executing:

```
python3 test_wrapper.py --config=../config/<station_config_file>.yml
                        --test_config=config/<test_config_file>.yml
                        -p
```

prints the available test cases along with a short description for each test case. Available tests and corresponding description are also shown in interactive mode in the main menu page. By default, running the script in non-interactive mode execute all the available tests.

7 Test output

Each test case is self-checking. After all the selected test cases have been executed, a summary is printed and logged to file showing the result for each test as shown below:

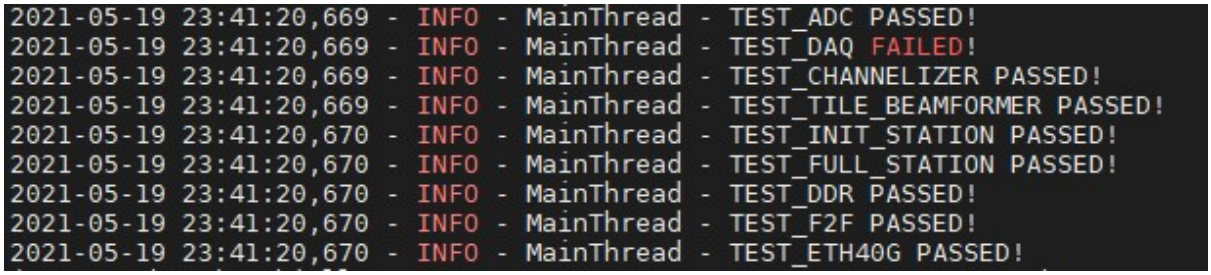

Figure 1. Sample test summary.

For specific information about test errors the user can refer to the test log file.

8 Extending the test suite with additional test cases

Each test case is defined as a Python class. In order to support the automatic integration of additional test cases into the test suite, the following requirements should be met:

- The python file name should begin with "test_", for instance test_f2f.py
- __init__ method should be defined as follows:

  ```
  def __init__(self, station_config, logger)
  ```

  where `station_config` is the station configuration dictionary created from the station configuration file using the relevant function defined in station.py and `logger` is a python logger which should be used by the test executor to log test messages. Both parameters are automatically passed from test_wrapper.py to the test object instance.

- `execute` method which executes the test on the TPMs. Optional arguments can be specified in the `execute` method definition, for instance test duration or number of test loops that should be executed. The `execute` method should return 0 when test is successful or a positive value when the test fails. All defined parameters should have a default value which will be used as default parameters in non-interactive and interactive mode.

Considering a new test case implemented in test_new_case.py and fulfilling the requirements above, to integrate it into the test suite, it is necessary to modify test_wrapper.py adding the corresponding key into the python dictionary self._test defined in the __init__ of the class TestWrapper. In this example, the new key should be "new_case", obtained by removing the prefix "test_" from the corresponding test_new_case.py file name. The associated value should be a description of the test case.