

# **tt-master Interface**

## **High level control of the SAI MASS/DIMM software**

Specification of an OpenTPL based interface to control the DIMM software

Version 1.4

Document ID:

`dev-dimm-tt-master:spec-en`

Revision 213 as of June 7, 2019 08:59:51



## Revision history:

| Date       | Version | Changes   |
|------------|---------|---|
| 22.03.2019 | 1.0     | Initial version (mv)  |
| 23.03.2019 | 1.1     | Initial draft release (mr)  |
| 01.04.2019 | 1.2     | Added *.SERVICE.FAIL_STATE, AMEBA.CURRENT.TIMESTAMP and AMEBA.CONDITION, update values of AMEBA.STATE, removed AMEBA.CURRENT.STELLAR_CLASS (mv)   |
| 23.05.2019 | 1.3     | Added AMEBA.SUN_ALT, AMEBA.START_TIME AMEBA.FINISH_TIME and DIMM.FLUX, DIMM.FLUX_RMS, DIMM.AIRMASS, DIMM.STREHL_LEFT and DIMM.STREHL_RIGHT(mv,mr) |
| 07.06.2019 | 1.4     | Added DIMM.FLUX_LEFT, DIMM.FLUX_RIGHT, DIMM.FLUX_RMS_LEFT and DIMM.FLUX_RMS_RIGHT, removed DIMM.FLUX and DIMM.FLUX_RMS (mv)                       |



tt-master Interface — High level control of the SAI MASS/DIMM software

Version 1.4

Document ID: dev-dimm-tt-master:spec-en

Revision 213 as of June 7, 2019 08:59:51 by Michael Ruder

Copyright © tau-tec GmbH, Hintere Grabenstr. 30, 72070 Tübingen, Germany, all rights reserved.

Any storage, duplication, reproduction, translation or forwarding, even in excerpts, is only permissible with our written authorisation.

dev-dimm-tt-master\_spec-en.lyx 213 2019-06-07 08:59:51Z ruder

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Preface</b>                         | <b>5</b>  |
| <b>2</b> | <b>Basics</b>                          | <b>5</b>  |
| 2.1      | Conventions in this document . . . . . | 5         |
| <b>3</b> | <b>List of modules</b>                 | <b>5</b>  |
| 3.1      | Common module variables . . . . .      | 6         |
| 3.2      | Command execution times . . . . .      | 6         |
| 3.3      | Permissions . . . . .                  | 6         |
| <b>4</b> | <b>The generic SERVICE submodule</b>   | <b>7</b>  |
| <b>5</b> | <b>The AMEBA module</b>                | <b>7</b>  |
| 5.1      | The AMEBA.MANUAL submodule . . . . .   | 8         |
| 5.2      | The AMEBA.CURRENT submodule . . . . .  | 9         |
| <b>6</b> | <b>The DIMM module</b>                 | <b>9</b>  |
| <b>7</b> | <b>The DOME module</b>                 | <b>10</b> |
| <b>8</b> | <b>The METEO module</b>                | <b>10</b> |
| <b>9</b> | <b>The SCOPE module</b>                | <b>11</b> |
| 9.1      | The SCOPE.STATUS submodule . . . . .   | 11        |



---

# 1 Preface

This document describes a high level interface to control the SAI MASS/DIMM software. For the communication the OpenTPL protocol is used (refer to [2] for protocol specifications).

## 2 Basics

### 2.1 Conventions in this document

This manual uses the **typewriter font** whenever OpenTPL protocol text is shown. Text that is enclosed in `<...>` stands as a placeholder. So `VARIABLE=<value>[,<value>[,...]]` means that some real values and not the string `<value>` is expected by the server. Sometimes parameters are optional. This is shown as in the previous example by putting the text in `[...]`. If these brackets are used in the OpenTPL protocol itself, they will be typeset as `<...>` and `[...]`.

In the sections that describe the OpenTSI modules and variables, tables will be used with the following columns:

**Name** the OpenTPL module/variable name. For the full OpenTPL path, the prefix from the section caption needs to be added. For instance, if a variable `VAR` is discussed in the section describing the `HARDWARE.MONITOR` submodule, the OpenTPL path of this variable is `HARDWARE.MONITOR.VAR`.

**T** The data type of the OpenTPL variable. Valid types can be found in [2].

**R** The standard read level of the OpenTPL variable. While it is not recommended to do so, it is of course possible to adapt the individual levels of the different variables to specific needs. Therefore, clients should not be designed in a way that they expect the here documented levels, but either check the real levels or correctly cope with a `DENIED` error. Refer to [2] for an explanation of the read (and write) levels.

**W** The standard write level of the OpenTPL variable. The same comments as for the read level hold here as well.

**Description** The detailed explanation of the functionality behind the variable.

## 3 List of modules

The following modules are specified by the current version of the tt-master interface:

| Name  | Function   |
|-------|--|
| AMEBA | Status and control of ameba master daemon (see section 5).     |
| DIMM  | Status and control of dimm measurement daemon (see section 6). |
| DOVE  | Status and control of dove daemon (see section 7).             |
| METEO | Status and control of meteo weather daemon (see section 8).    |
| SCOPE | Status and control of scope telescope daemon (see section 9).  |

### 3.1 Common module variables

Every module must have at least the following variable, which will not be listed in the detailed module description sections:

| Name    | T | R | W  | Description  |
|---------|---|---|----|--|
| VERSION | i |   | -1 | <p>Interface version (I), interface age (A) and revision number (R), coded as 0xIIIIAARR (<math>\neq 0</math> if the module exists, <math>= 0</math> if it does not):</p> <ul style="list-style-type: none"><li>• The interface version will be increased whenever variables are added or removed. The latest tt-master interface version is 0x0010 which corresponds to version 1.0. This number must be the same for all available modules on a single server.</li><li>• The interface age allows clients to check if the discovered interface version can be used even if the interface version is different from the expected version. The age is computed as maximum supported interface version minus minimum supported version.<br/><i>Example:</i> interface version is 3 but all variables of version 2 are still in place to stay compatible with old clients. Therefore the age equals <math>3 - 2 = 1</math>.</li><li>• The revision number indicates the code revision. In general: the higher, the better, fewer bugs etc.</li></ul> |

### 3.2 Command execution times

When reading from variables, values will be returned immediately (e.g. the typical callback run time is in the order of 1 ms). However, writing to variables may take longer times to finish, as the completion of the triggered action is often waited for. Therefore, for every writable variable where SET will not return immediately, this is explicitly stated in the following chapters.

### 3.3 Permissions

tt-master allows access to all variables for any user (regardless of its RLEVEL). Writing to variables has however been restricted. The following table gives an overview of the functional groups accessible for a certain WLEVEL.

| WLEVEL | Access rights  |
|--------|--|
| 20     | Access to AMEBA.MANUAL.*, AMEBA.MODE and SCOPE.STATUS.CLEAR to allow setting up monitoring and clearing the scope status when problems are reported. |
| 10     | Access to *.SERVICE.CONTROL to allow starting and stopping the programs.   |
| 0      | Admin user, allowed to access the management interface of the OpenTPL server (SERVER module).  |

---

## 4 The generic SERVICE submodule

This submodule contains variables to read and manually manage the running state of each program, therefore it exists in all modules.

| Name       | T | R  | W  | Description   |
|------------|---|----|----|---|
| STATE      | i |    | -1 | Status of the program. This state resembles the <b>is-active</b> property of systemctl. Possible values are:<br><br><b>0</b> not running<br><br><b>1</b> running  |
| FAIL_STATE | i |    | -1 | Failure status of the program. This state resembles the <b>is-failed</b> property of systemctl. Possible values are:<br><br><b>0</b> no failure reported<br><br><b>1</b> failure reported   |
| CONTROL    | i | -1 | 10 | Manage the status of the program by writing a value. Possible values are:<br><br><b>0</b> stop<br><br><b>1</b> start program<br><br><b>2</b> restart program<br><br>SET returns after the requested operation has been carried out. |

## 5 The AMEBA module

This modules gives access to the overall software status and allow controlling the monitoring mode.

| Name    | T | R | W | Description   |
|---------|---|---|---|---|
| SERVICE | M |   |   | Read service status and manual stop/start/restart service (see section 4) |
| MANUAL  | M |   |   | Target selection for manual observation mode (see section 5.1)            |
| CURRENT | M |   |   | Currently used target (see section 5.2)                                   |

| Name        | T | R | W  | Description   |
|-------------|---|---|----|---|
| MODE        | i |   | 20 | Control the target state of the DIMM monitoring. Possible values are:<br><br><b>0</b> off<br><br><b>1</b> auto — uses the star list with automatic selection of the used target for monitoring.<br><br><b>2</b> manual — forces the observation target to be the configured target in <code>AMEBA.MANUAL</code> (see section 5.1).<br><br>SET returns after the observation mode was changed. |
| STATE       | i |   | -1 | Current working state. Possible values are:<br><br><b>0</b> inactive<br><br><b>1</b> waiting<br><br><b>2</b> slewing<br><br><b>3</b> tracking<br><br><b>4</b> focusing<br><br><b>5</b> monitoring   |
| SUN_ALT     | f |   | -1 | Current sun altitude at the site [°]  |
| CONDITION   | i |   | -1 | Current status of the ambient conditions as a bit-mask. Monitoring will start if all conditions bits are set. The following bits are possible:<br><br><b>Bit 0 (value: 1)</b> Humidity is OK.<br><br><b>Bit 1 (value: 2)</b> Wind is OK.<br><br><b>Bit 2 (value: 4)</b> Sky is clear.   |
| START_TIME  | f |   | -1 | Time when system will power up, dome will open and automatic monitoring will start. [UTC seconds since 01.01.1970 00:00:00]   |
| FINISH_TIME | f |   | -1 | Time when automatic monitoring will end, dome will close and system will park and power off. [UTC seconds since 01.01.1970 00:00:00]  |

### 5.1 The AMEBA.MANUAL submodule

This module contains variables to set up a target for the manual monitoring mode. After the object has been set up, `AMEBA.MODE` needs to be set to 2 to apply the changes, even if the system is running in manual mode already.

| Name | T | R | W  | Description        |
|------|---|---|----|--------------------|
| NAME | s |   | 20 | Name of the target |



| Name              | T | R | W  | Description   |
|-------------------|---|---|----|---|
| RA                | f |   | 20 | Right ascension (equinox of J2000.0) [h]  |
| DEC               | f |   | 20 | Declination (equinox of J2000.0) [°]  |
| BRIGHTNESS        | f |   | 20 | Apparent visual brightness in the V-band in magnitudes  |
| COLOR             | f |   | 20 | Object color in difference between the B- and V-band in magnitudes  |
| STELLAR_CLASS     | s |   | 20 | Yerkes spectral classification of the object, e.g. 'G5III'  |
| STELLAR_CLASSFILE | s |   | -1 | Selected spectral file, based on the value given in STELLAR_CLASS. This depends on the files in the /opt/dimm/dimm_tool/data/spectra directory. |

## 5.2 The AMEBA.CURRENT submodule

This module contains variables to get information about the currently used monitoring target.

| Name              | T | R | W  | Description   |
|-------------------|---|---|----|---|
| NAME              | s |   | -1 | Name of the target  |
| RA                | f |   | -1 | Right ascension (equinox of J2000.0) [h]  |
| DEC               | f |   | -1 | Declination (equinox of J2000.0) [°]  |
| BRIGHTNESS        | f |   | -1 | Apparent visual brightness in the V-band in magnitudes  |
| COLOR             | f |   | -1 | Object color in difference between the B- and V-band in magnitudes  |
| STELLAR_CLASSFILE | s |   | -1 | Spectral file of the object. This depends on the files in the /opt/dimm/dimm_tool/data/spectra directory. |
| START_TIME        | f |   | -1 | Time when the scope was requested to track the object [UTC seconds since 01.01.1970 00:00:00]             |

## 6 The DIMM module

This module contains variables to provide access to the results of the monitoring.

| Name           | T | R | W  | Description  |
|----------------|---|---|----|--|
| SERVICE        | M |   |    | Read service status and manual stop/start/restart service (see section 4)          |
| SEEING         | f |   | -1 | Last seeing estimation [arcseconds]  |
| SEEING_LOWFREQ | f |   | -1 | Low-frequency part of the last seeing estimation [arcseconds]                      |
| FLUX_LEFT      | f |   | -1 | Integral flux for the left aperture image of the last seeing estimation [ADU]      |
| FLUX_RIGHT     | f |   | -1 | Integral flux for the right aperture image of the last seeing estimation [ADU]     |
| FLUX_RMS_LEFT  | f |   | -1 | Integral flux RMS for the left aperture image of the last seeing estimation [ADU]  |
| FLUX_RMS_RIGHT | f |   | -1 | Integral flux RMS for the right aperture image of the last seeing estimation [ADU] |
| AIRMASS        | f |   | -1 | Mean airmass of the last seeing estimation   |
| STREHL_LEFT    | f |   | -1 | Strehl ratio for the left aperture image of the last seeing estimation             |

| Name         | T | R | W  | Description  |
|--------------|---|---|----|--|
| STREHL_RIGHT | f |   | -1 | Strehl ratio for the right aperture image of the last seeing estimation    |
| TIMESTAMP    | f |   | -1 | Time of the last seeing estimation [UTC seconds since 01.01.1970 00:00:00] |

## 7 The DOME module

This module contains variables to provide information about the dome.

| Name           | T | R | W  | Description   |
|----------------|---|---|----|---|
| SERVICE        | M |   |    | Read service status and manual stop/start/restart service (see section 4)   |
| POSITION       | f |   | -1 | Position of the dome. Possible values are:<br><br><b>0.0</b> fully closed<br><br><b>0.0 ... 1.0</b> partially opened<br><br><b>1.0</b> fully opened             |
| POSITION_SIDEA | f |   | -1 | Position of the first dome part. Possible values are:<br><br><b>0.0</b> fully closed<br><br><b>0.0 ... 1.0</b> partially opened<br><br><b>1.0</b> fully opened  |
| POSITION_SIDEB | f |   | -1 | Position of the second dome part. Possible values are:<br><br><b>0.0</b> fully closed<br><br><b>0.0 ... 1.0</b> partially opened<br><br><b>1.0</b> fully opened |
| TEMPERATURE    | f |   | -1 | Dome temperature [°C]   |
| POWER_STATE    | i |   | -1 | Current power state of the dome. Possible values are:<br><br><b>0</b> parked<br><br><b>1</b> powered up   |

## 8 The METEO module

This module contains variables to provide information about the weather station monitor (currently only the **SERVICE** submodule).

| Name    | T | R | W | Description   |
|---------|---|---|---|---|
| SERVICE | M |   |   | Read service status and manual stop/start/restart service (see section 4) |

## 9 The SCOPE module

This module contains variables to provide information about the current scope state and its pointing.

| Name         | T | R | W  | Description  |
|--------------|---|---|----|--|
| SERVICE      | M |   |    | Read service status and manual stop/start/restart service (see section 4)  |
| STATUS       | M |   |    | Telescope status and error reset (see section 9.1)   |
| RA           | f |   | -1 | Current telescope right ascension (equinox of J2000.0) [h]   |
| DEC          | f |   | -1 | Current telescope declination (equinox of J2000.0) [°]   |
| AZ           | f |   | -1 | Current telescope azimuth [°]  |
| ALT          | f |   | -1 | Current telescope altitude [°]   |
| FOCUS        | f |   | -1 | Current focus position in arbitrary units  |
| MOTION_STATE | i |   | -1 | Current motion state of the telescope. Possible values are:<br><br>-2 error<br><br>-1 parked<br><br>0 stopped<br><br>1 slewing<br><br>2 tracking |
| POWER_STATE  | i |   | -1 | Current power state of the telescope. Possible values are:<br><br>0 parked<br><br>1 powered up   |

### 9.1 The *SCOPE.STATUS* submodule

This submodule contain variables to check and reset reported events from the telescope, that may be indicating abnormal behavior or problems.

| Name  | T | R  | W  | Description   |
|-------|---|----|----|---|
| LIST  | s |    | -1 | <p>A comma separated list of all saved error events in the format <code>&lt;error&gt;:&lt;status&gt;:&lt;device&gt;[:&lt;comment&gt;][,...]</code>.</p> <p><b>&lt;error&gt;</b> A hardware dependent error name</p> <p><b>&lt;status&gt;</b> Bit coded severity of the error (bits not listed below are reserved and need to be zero):</p> <p><b>Bit 0</b> <i>PANIC</i>: Cabinet does not work anymore; the error is so severe that the whole telescope cannot operate anymore.</p> <p><b>Bit 1</b> <i>ERROR</i>: Cabinet has errors and is not working.</p> <p><b>Bit 2</b> <i>WARNING</i>: Cabinet is still working but performance may be influenced.</p> <p><b>Bit 3</b> <i>INFO</i>: Cabinet is working; some information on the status have been saved.</p> <p><b>Bit 4</b> <i>DEBUG</i>: Cabinet is working; some debug information is available.</p> <p><b>&lt;device&gt;</b> The OpenTCI (refer to [1]) module where the error occurred (e.g. HA) or UNKNOWN if the origin cannot be matched to a module.</p> <p><b>&lt;comment&gt;</b> Additional information on the error (optional)</p> |
| CLEAR | i | -1 | 20 | <p>Acknowledges all events on write. If the condition that caused the event persists, the event will reoccur or not get cleared at all. Otherwise, the affected module will retry its action or perform a reinitialization.</p>   |

## References

- [1] Michael Ruder, Mario Velten, and Daniel Plasa. *OpenTCI, Open Telescope Control Interface — An open specification of an OpenTPL based interface to provide generic, low-level control of telescopes*. tau-tec GmbH, Hintere Grabenstr. 30, 72070 Tübingen, Germany, 2012. `tclm-tci:spec-en`.
- [2] Michael Ruder, Mario Velten, and Daniel Plasa. *OpenTPL, Open Transfer Protocol Language — A protocol for client-server based exchange of data and commands over a TCP/IP network connection*. tau-tec GmbH, Hintere Grabenstr. 30, 72070 Tübingen, Germany, 2012. `opentpl:spec-en`.