



Abstract

Oregano Systems provides two Application Programming Interfaces (APIs) for the syn1588[®] product line:

- The shared memory API (sharedmem_API) for controlling, monitoring and configuring the syn1588[®] PTP Stack.
- The syn1588[®] API (syn1588_API) for accessing syn1588[®] hardware like the syn1588[®] PCIe NIC hardware. It allows to control hardware features like generating frequencies, capturing events, and configuring the timestamps. For more information about the syn1588[®] API take a look at the respective documentation (doc folder of the syn1588_API package) or contact [Oregano Systems](#).

This application note describes the shared memory API for the syn1588[®] PTP Stack allowing access to many parameters defined in IEEE 1588-2008 as well as other relevant data. This API can be used to configure, control, and monitor the syn1588[®] PTP Stack.

Please note that the API is designed and specialized in working with the syn1588[®] PTP Stack only. Other applications than the syn1588[®] PTP Stack are not supported by this API.

Introduction

The shared memory API allows the user to gather status information about the syn1588[®] PTP Stack as well as current values of all its configuration parameters. For example, the API could be used by any 3rd party application to read the current UTC offset and synchronization status, configure the current UTC offset, time traceability and clock class depending on the GPS input to mention but a few of the possibilities. Figure 1 depicts two likely use cases of the shared memory communication mechanism.

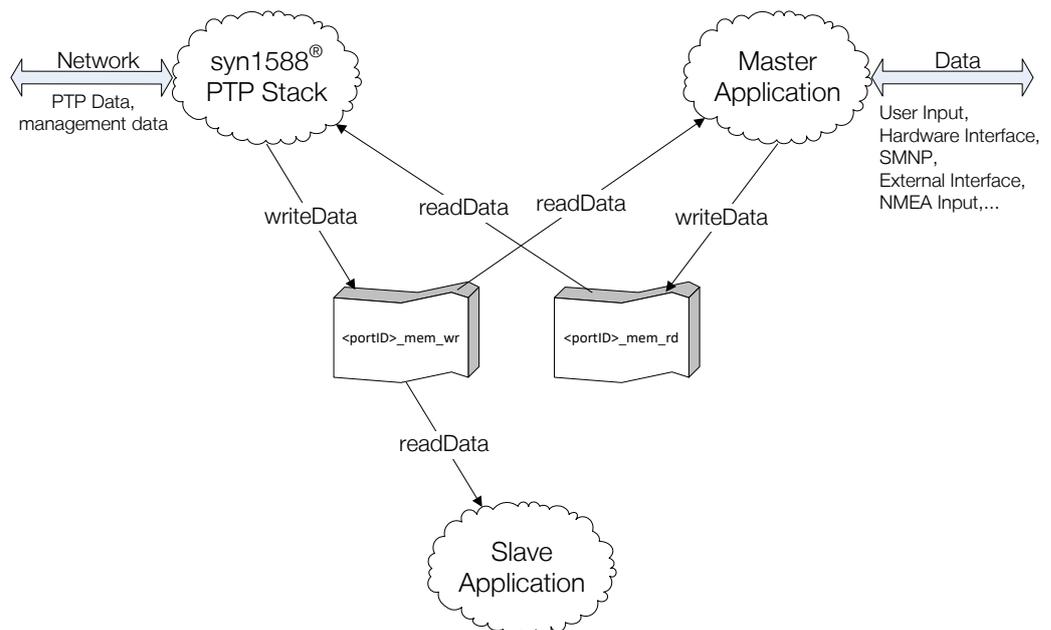


Figure 1 Shared Memory Communication

A master application reads data from the shared memory `<portID>_mem_wr` for monitoring purposes. After processing the information it will write back modified data to `<portID>_mem_rd` for (re-)configuration purposes. This updated data will in turn be read and applied by the syn1588[®] PTP Stack. A slave application, on the other hand, only reads data and processes it.

Data Format

The data format used for the shared memory communication is specified in the source file “src/include/datasets.h”. It contains structures of the datasets described in the IEEE 1588-2008 standard chapter 8 as well as other relevant data like profile specific parameters. Additionally, there is a status field to flag the current status of the shared memory (memory read? Memory updated?) as well as an ID field to flag which application accessed the shared memory.

Oregano Systems provides an API for simplified access the shared memory. The API is available in C++ for Windows as well as for Linux facilitating application development for controlling, configuring, and monitoring of the syn1588[®] PTP Stack. The include file “src/include/datasets.h” is written in a way that it can be used for C as well as C++ programs.

Please be aware that Oregano Systems reserves the right to update the data format at any time, e.g. with newer revisions of the syn1588[®] PTP Stack. Make sure that only compatible software versions are used for shared memory communication (e.g. only use identical versions of ptp and eSync).

Status

The following flags are used to identify the memory's status:

Flag	Status	Enum ContextStat_e
0x000	Empty	EEmpty
0x001	New data available (written by the PTP Stack)	ENewDataWr
0x004	New data for the PTP Stack available	ENewDataRd
0x010	Alternate time offset data is valid	EAltTimeOff
0x020	Power profile (C37.238) data is valid	EPowerData
0x040	SMPTE sync metadata is valid	ESmpteData
0x080	Telecom 2 data (G.8257.1) data is valid	ETelecom2Data
0x100	Local time zone data (C37.238 2014) is valid	ELocalTZ
0x200	Path Trace data is valid	EPathTrace
0x400	802.1 as data is valid	E802_1as
0x10000	Message counters are valid	EMessageCounter
0x20000	Unicast connections are valid	EUnicastConnections

ID

The following IDs are used to identify access to the shared memory of different applications:

ID	Application	Enum ContextId_e
0x00	Res	ERes
0x01	syn1588 [®] ptp1 (Version 1 PTP Stack)	EPtpv1
0x02	syn1588 [®] PTP Stack	EPtpv2
0x03	syn1588 [®] eSync	EEsync
0x04	syn1588 [®] lSync	ELsync
0x05-0x0F	Reserved	
0x10	Meinberg	EMbg
0x11-0xFF	Third party applications	

syn1588[®] PTP Stack Operation

This section describes how the syn1588[®] PTP Stack handles the shared memory internally. The syn1588[®] PTP Stack will attempt to access the shared memory when one of the following events occurs:

- Heartbeat timeout (once every second)
- PTP management “Set” message is received

During every access, the syn1588[®] PTP Stack performs the following actions: It updates the <portID>_mem_wr memory with the latest data and checks, if new data is available in <portID>_mem_rd applying updates, if necessary. The syn1588[®] PTP Stack always writes the whole data structure to the memory. When using a boundary clock (BC), all ports of the BC write data to the respective shared memory (<portID>_mem_rd). However, most datasets are only read and applied by the port with port number 1. Whenever the syn1588[®] PTP Stack decides to apply data from the syn1588[®] PTP Stack, it applies only the following data (limitations in brackets):

- Default Dataset: all dynamic members (BC: processed by port number 1 only)
 - Clock Quality (clock class, clock accuracy and variance)
 - Priority 1
 - Priority 2
 - Domain number
 - Slave only flag

- Time properties: all members (master/passive state only, BC: processed only port number 1)
- Port Dataset: all configurable members (BC: processed by all ports):
 - Announce Interval (master/passive state and unicast slave)
 - Announce receipt timeout
 - Sync Interval (master/passive state and unicast slave)
 - Delay mechanism
 - min PDelay Request Interval
 - Delay Request Interval (master/passive state and unicast slave)
- Alternate Time Offset Properties: all members, three instances (master/passive state only, BC: processed by port number 1 only)
- Power Profile Data (power profile activated, master/passive state only, BC: processed by port number 1 only)
- Local time zone (power 2 profile activated, master/passive state only, BC: processed by port number 1 only)
- SMPTE sync metadata (SMPTE profile activated, master/passive state only, BC: processed by port number 1 only)

The Current Dataset and the Parent Dataset are not updated because they will always be managed by the syn1588[®] PTP Stack. The same applies to the non-standard related Extended Parent Dataset which holds some additional information about the current master, e.g. the network protocol address or the clock parameters of the best master in the network in passive state.

Prerequisites

To start using the shared memory API, the following steps have to be executed:

- Add the library to your build. Please refer to the manual of your build environment to get more information of how to add an external project to your project.
- Include the header file “sharedmem_api.h” to use the functions which allows you to access the shared memory.

Please see the program sm_example for an example of how to use the API.

Reading Data

If an application other than the syn1588[®] PTP Stack wants to access the shared memory, it shall do the following steps (C++ API syntax):

- Open the desired shared memory identified by <portID> using


```
PTP_SharedMemoryIfc::open(AccessMode_e mode, PortId_t &id)
```

- in
- ```
PTP_SharedMemoryIfc::ESlave mode
```
- Read the data structure `Context_t` using

```
PTP_SharedMemoryIfc::readPtpContext(Context_t *c)
```

Please see the program `sm_example` for an example on how to read data.

## Writing Data

If an application wants to configure the syn1588<sup>®</sup> PTP Stack by writing to the shared memory, it shall do the following things (C++ API syntax):

- Open the desired shared memory identified by `<portID>` using

```
PTP_SharedMemoryIfc::open(AccessMode_e mode, PortId_t &id)
```

in

```
PTP_SharedMemoryIfc::EMaster mode
```
- Read the data structure `Context_t` using

```
PTP_SharedMemoryIfc::readPtpContext(Context_t *c)
```
- Change all desired parameters. It shall indicate the source (`ContextId_e`) and status shall indicate new data available for reading (`ContextStat_e::ENewDataRd`)
- Write the data structure `Context_t` using

```
PTP_SharedMemoryIfc::writePtpContext(const Context_t *c)
```

Please see the program `sm_example` for an example of how to write data.

## Compiling

Oregano Systems provides required C++ header- and source-files for the shared memory library and an example program (`sm_example`).

- Linux: There are two different projects: the shared memory library and the shared memory application. To compile the shared memory library go to the directory `proj/gcc/sharedMemlib` and enter

```
make -f Makefile
```

To compile the example program go to the directory `proj/gcc/sharedMemapp` and enter

```
make -f Makefile
```

- Windows: Use the Microsoft Visual Studio projects to build the example program.

To run the example program make sure that the syn1588<sup>®</sup> PTP Stack is running. Then change to the directory `bin` and enter

```
./sm_example <clkId> <port>
```

for Linux or

```
sm_example.exe <clkId> <port>
```

for Windows. <clockId> (PTP clockId of the appropriate syn1588® device) and <port> (PTP port of the selected syn1588® device, usually 1) specify which shared memory will be opened.

## Remarks

- An application using `PTP_SharedMemoryIfc::EMaster` mode **must** perform a `writePtpContext` after a `readPtpContext` call. Otherwise, the shared memory may not be unlocked correctly.
- Please be aware that an application intending to configure the syn1588® PTP Stack should check for new data periodically, using a similar interval as the heartbeat timeout (one second) of the syn1588® PTP Stack
- The shared memory will always be opened, even if there is no syn1588® PTP Stack running. Master or Slave applications may check the id and status field to make sure that a syn1588® PTP Stack is running.
- The syn1588® PTP Stack does not use any kind of range checking when data is read and applied from the shared memory. A master application is responsible for guaranteeing data integrity. If parameters are not within the range according to the appropriate specification, the syn1588® PTP Stack's behavior may be undefined.
- Please note that the shared memory API is NOT designed to analyze or monitor other functionalities than the syn1588® PTP Stack, like eSync or lSync.

## Runtime Configurable Parameters

Following parameters are dynamically configurable during runtime of the syn1588® PTP Stack. Their respective values can be changed at any given point in time, the syn1588® PTP Stack, however will apply changes caused by altering any one parameter whenever it re-accesses the content of the shared memory; by default this is done once every second. This update rate should not be modified because it affects other time triggered mechanisms within the main loop of the syn1588® PTP Stack

- default dataset
- port dataset:
  - announce Interval
  - announce reception timeout
  - sync Interval
  - (p)dlyReqInterval
  - dlyMechanism
- time properties dataset

- alternate time offset properties dataset
- power profile dataset
- power locale timezone dataset
- smpte sync metadata dataset
- telecom2 profile dataset
- 802.1AS dataset

Furthermore a port may be disabled and re-enabled during runtime.

The statistics counters may be reset at any given time.

---

|                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>Oregano Systems</b><br/>A Meinberg Company</p> <p>Franzosengraben 8<br/>A-1030 Vienna<br/>Austria<br/><a href="http://oregano.at">http://oregano.at</a><br/><a href="mailto:contact@oregano.at">contact@oregano.at</a></p> | <p>Copyright © 2020<br/>Oregano Systems – Design &amp; Consulting GmbH<br/>ALL RIGHTS RESERVED.</p> <p>Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.</p> <p>Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.</p> <p>Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.</p> <p>All trademarks used in this document are the property of their respective owners.</p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|