

**TSEP**

Technical  
Software  
Engineering  
Plazotta

# **Produktbeschreibung TSEP IEEE 1588 Stack (PTP Stack)**

## Allgemein:

Die Zeitsynchronisation über IEEE 1588 ist bereits seit 2008 als IEEE Norm festgeschrieben und wird bereits in verschiedensten Bereichen verwendet.

Bisher war die Nutzung dieses Standards immer mit exotischer Hardware verbunden, also mit Implementierungen von Netzwerkadaptern in verschiedenen FPGAs oder Embedded Controller. Mit der Einführung der Intel Netzwerkchip Familien Intel I21x und Intel I35x ist dieser Standard nun für den Consumer Markt verfügbar. Somit ist die Grundlage für neue Projekte auf Basis von Consumer Hardware gelegt.

Bereits 2016 wurde bei TSEP entschieden eine eigene IEEE 1588 Implementierung zu realisieren, 2017 konnte dann der erste lauffähige Prototyp gezeigt werden. Inzwischen ist der TSEP IEEE 1588 Stack auf mehreren Plattformen verfügbar und wird von Kunden eingesetzt.

Die grundsätzliche Frage, die sich bei jedem IEEE 1588 Projekt stellt ist, mit welcher Genauigkeit die Zeitsynchronisation erfolgen muss. Die erreichbare Genauigkeit hängt in der Regel von der verwendeten Hardware, der Topologie und dem verwendeten Regelalgorithmus ab. Moderne IEEE 1588 Implementierungen haben die Möglichkeit, verschiedene Regelalgorithmen zu definieren und diese einfach auszutauschen.

Auch TSEP hat den Regelalgorithmus als eigenständiges Modul mit definierten Schnittstellen festgelegt. Somit kann der Nutzer einfach einen eigenen Algorithmus definieren und diesen ins System einbringen und testen.

Wird IEEE 1588 zum Beispiel bei der Synchronisation von WLAN-Lautsprechern verwendet, ist das menschliche Gehör die Maßeinheit für die Genauigkeit. Das menschliche Gehör kann Laufzeitdifferenzen ab  $10\ \mu\text{s}$  erkennen. Somit muss hier die erreichte Genauigkeit für die Synchronisation der WLAN-Lautsprecher unter  $10\ \mu\text{s}$  liegen. Betrachtet man aber messtechnische Aufgaben, sind andere Genauigkeiten gefordert. Innerhalb der Mess-technik werden Messungen in der Regel mit Hilfe von Triggern ausgelöst. Diese Trigger sind Signaländerungen (steigende oder fallende Flanken, Überschreitung von Pegelwerten etc.). Diese Signale werden über Kabel von der Quelle zum Messgerät übertragen. Somit ist die Laufzeiten innerhalb des Trigger Kabels die maßgebende Zielgröße für die Genauigkeit. Geht man von Kabellängen von ca. 5 Metern aus, was eher großzügig bemessen ist, kann man von einer Laufzeit von 25 ns ausgehen (Laufzeit von 5 ns per Meter). Somit wäre die Genauigkeit bei messtechnischen Problemen in dieser Größenordnung einzuordnen. Jedoch hat sich im Bereich Messtechnik mit der Vorstellung von 5G Technologien im Mobilfunk diese Größenordnung deutlich nach unten verschoben. Für diese Technologien wären Genauigkeiten im Sub-Nano-Bereich wünschenswert.

## Technische Realisierung:

Der TSEP IEEE 1588 Stack wurde komplett in C++ entwickelt und entspricht modernen Coding Richtlinien. Um unabhängig von einer speziellen Entwicklungsumgebung zu sein, wurde CMake für die Code Verwaltung gewählt. Des Weiteren wurde ein konsequent objektorientierter Ansatz verfolgt.

Der TSEP IEEE 1588 Stack kann sowohl als Ordinary Clock („Client“), als auch als Master Clock fungieren.

Eine Master Clock verteilt seine eigene Uhrzeit mittels sogenannter Sync Messages über das angeschlossene Ethernet Netz, die Ordinary Clock empfängt diese Messages und beginnt daraufhin mit der Master Clock zu kommunizieren um sich zu synchronisieren.

Der TSEP IEEE 1588 Stack unterstützt hierbei sowohl das One Step Verfahren, bei diesem wird der aktuelle Zeitstempel direkt in die Sync Message integriert, was jedoch entsprechende Hardware Unterstützung voraussetzt. Für alle Netzwerk Chips ohne diese Unterstützung wird auch das Two Step Verfahren unterstützt, hier wird der Zeitstempel in einer zweiten Nachricht geschickt.

Um eine genaue Zeitsynchronisation auch in Netzen mit Geräten (z.B. Switches oder Router), die nicht IEEE 1588 unterstützen zu gewährleisten, ist der TSEP IEEE 1588 Stack als sogenannte End-to-End (E2E) Lösung implementiert.

Der TSEP IEEE 1588 Stack kann nach der Synchronisierung auch Hardware Trigger auslösen. Diese erfordern jedoch die entsprechende Hardwareunterstützung. Dadurch besteht die Möglichkeit Messungen oder ähnliches synchron an mehreren Geräten auszulösen oder aber auch in Intervallen zu triggern, z.B. für ein PPS Signal. Zusätzlich kann der TSEP IEEE 1588 Stack auch die unterliegenden Betriebssysteme mit der IEEE 1588 Zeit versorgen.

Der TSEP IEEE 1588 Stack unterstützt alle im IEEE 1588 Standard definierten Management Messages. Über diese Messages können Informationen abgefragt oder Einstellungen einer Clock manipuliert werden.

Der TSEP IEEE 1588 Stack kann als Standalone System betrieben werden und ist somit direkt auf dem gewünschten Gerät ausführbar. Voraussetzung hierfür sind jedoch unterstützte Netzwerkkarten (z.B. I21x von Intel). Unterstützt werden sowohl die Betriebssysteme von Windows (7 und 10) und Linux (Ubuntu 16.04). Zusätzlich kann der TSEP IEEE 1588 Stack auch unter dem Echtzeit Betriebssystem RTX von IntervalZero betrieben werden. Die Kombination von TSEP IEEE 1588 Stack und RTX64 von IntervalZero ist ideal für alle zeitkritischen Anwendungsfälle. Alle Operationen werden in solch einem System zur „richtigen Zeit“ und in Echtzeit ausgeführt! Da Intel keine Windows Treiber für die IEEE 1588 Unterstützung der Intel Netzwerk Chips I21x und I35x anbietet, stellt TSEP eigene Treiber bereit.

Der gesamte Source Code ist für alle Plattformen identisch (common source) und muss nur entsprechend kompiliert werden.

## **Regelalgorithmus / Servo:**

Bei IEEE 1588 wird versucht mehrere freilaufende Uhren zu synchronisieren. Jeder dieser Uhren ist in der Regel als Zähler implementiert, der mit einer vorgegeben Frequenz seinen Zähler inkrementiert. Aufgrund der Frequenz und des Zählerstandes kann nun jederzeit die aktuelle Zeit abgeleitet werden. Da es technisch nicht möglich ist, von mehreren Oszillatoren identische Frequenzen erzeugen zu lassen, muss die Frequenz nachgeregelt werden. Da es technisch deutlich einfacher ist, den Zählerzyklus zu manipulieren, wird dieser verändert. Diese Anpassung muss über einen Regelalgorithmus erfolgen, da die Anpassungen verschiedenen Störungen unterliegen. Zusätzlich müssen aber auch Störungen, die im Transportweg auftreten können, berücksichtigt werden. Da jede IEEE 1588 Implementierung auf ihrer eigenen Hardware und Hardware-Topologie basiert, kann es „den allgemeinen Regelalgorithmus“ nicht geben. Aus diesen Grund wurde beim TSEP IEEE 1588 Stack der Regelalgorithmus in einer eigenen ständigen Bibliothek (.dll oder .so) ausgelagert. Somit kann jeder Kunde seinen eigenen Algorithmus implementieren unabhängig vom TSEP IEEE 1588 Stack. Zusätzlich liefert TSEP auch einen Standard Regelalgorithmus mit, der sich in der Regel nur darauf konzentriert, aus der ermittelten Zeitdifferenz zwischen Master und Slave (auch MeanPathDelay genannt) den Fehler

in der Frequenz der eigenen Uhr zu ermitteln. Diese Art von Algorithmus ist unabhängig von der verwendeten Hardware-Topologie und liefert durchaus brauchbare Ergebnisse. Um die Toleranzen, die im System stecken, zu ermitteln und diese bei der Berechnung des Fehlers der eigenen Frequenz miteinzubeziehen kann der Kunde seine eigenen Algorithmen implementieren. Diese Algorithmen sind nur dann sinnvoll, wenn die verwendete Hardware und die zu erwartende Topologie bekannt ist. Aufgrund der verwendeten Hardware lassen sich dann die Fehlermodelle erstellen und verwenden. Für diese Art von Regelalgorithmen sind insbesondere Kalman-Filter geeignet, die speziell auf das entsprechende Problem modelliert werden können.

Jeder Regelalgorithmus enthält mindestens zwei Zustände. Im ersten Zustand ist der Offset zwischen Master und Slave (MeanPathDelay) so groß, dass der Algorithmus diese Lücke in akzeptabler Regelzeit nicht schließen kann. In diesem Zustand wird die vom Master erhaltene Zeit ohne Korrektur direkt als eigene Slave-Zeit übernommen in der Hoffnung, dass der im nächsten Synchronisationsintervall ermittelte MeanPathDelay Wert deutlich kleiner ist. Dieser Zustand wird solange beibehalten, bis ein akzeptabler MeanPathDelay erreicht ist. Dieser Zustand ist der Default-Zustand nach dem Starten der Uhr oder wenn aufgrund von Problemen die Synchronisation verloren geht. Im zweiten Zustand greift dann der eigentliche Regelalgorithmus, der versucht die Korrekturwerte der eigenen Uhr zu ermitteln und die eigene Uhrzeit möglichst genau der Master-Uhrzeit anzunähern.

Dieser TSEP Standard Regelalgorithmus versucht im ersten Schritt, ungültige oder falsche MeanPathDelays nicht mit in die Regelung zu übernehmen. Bei Gigabit Ethernet nach IEEE-Norm 802.3 ist die Datenübertragung nicht deterministisch, jeder Teilnehmer kann jederzeit auf das Netzwerk zugreifen. Die Zugriffe werden über Paketkollisionen organisiert. Hierbei kann es dazu kommen, dass Pakete erst deutlich später übertragen werden, als eigentlich angenommen wird. Dieser Delay taucht nicht in den übertragenen Datenpaketen auf. Um die Regelalgorithmen vor diesen falschen und somit störenden Daten zu schützen, versucht der Regelalgorithmus diese falschen Datenpakete zu detektieren und nicht mit in den Regelalgorithmus einzubinden.

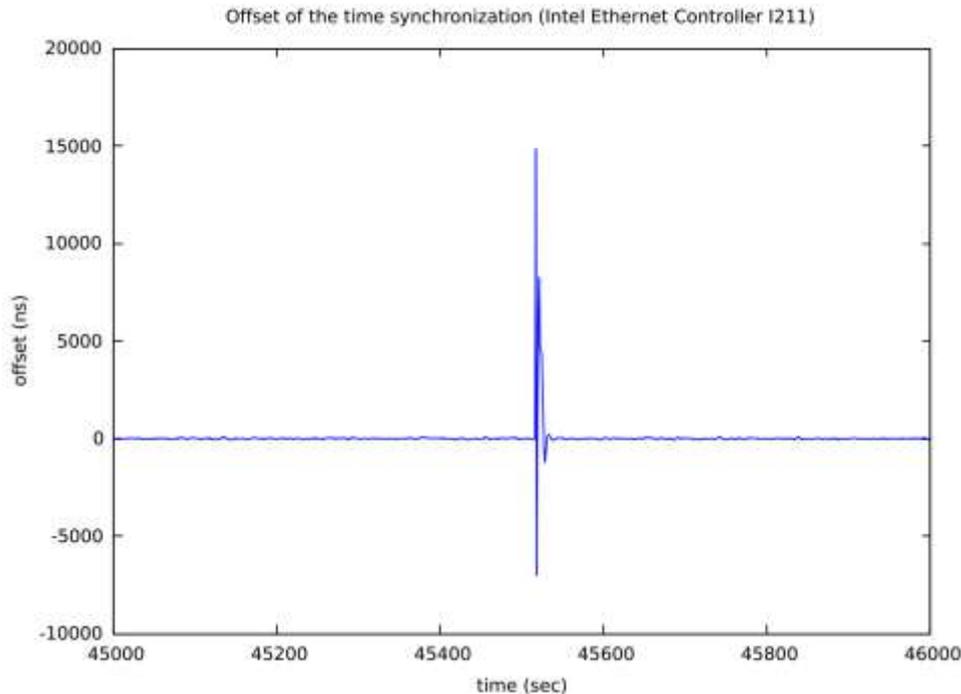


Abbildung 1: Falscher MeanPathDelay durch Netzwerkverzögerungen

Das obenstehende Diagramm zeigt ein solches falsches Paket, das nachfolgend in den Regelalgorithmus einbezogen wurde und dann ausgeglichen werden musste. Diese falschen Daten sind an den deutlich erhöhten MeanPathDelay zu erkennen. Um diese falschen MeanPathDelays zu detektieren, wird die Standardabweichung des MeanPathDelay berechnet:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

$\sigma$  Standardabweichung  
 $\mu$  Erwartungswert  
 $n$  Anzahl der Meßwerte

Abbildung 2: Berechnung der Standardabweichung MeanPathDelay

Überschreitet nun ein neuer MeanPathDelay die berechnete Standardabweichung deutlich, wird dieser nicht für die weitere Verarbeitung verwendet.

Im nächsten Schritt wird versucht, aus den berechneten MeanPathDelay, die Korrektur der eigenen Uhr zu berechnen. Hierzu wird die Abweichung des Slaves vom Master, die über den MeanpathDelay Algorithmus bestimmt wurde, umgelegt auf die Frequenz des eigenen Zählers (Uhr). Hierzu wird im ersten Schritt der Fehler der eigenen Uhr pro Zählerschritt berechnet:

$$\Delta F \text{ [sec]} = \frac{\Delta tm \text{ [sec]}}{\Delta ts \text{ [sec]} * (1 / tps)}$$

$\Delta F$ : Fehler der Uhr pro Tick  
 $\Delta tm$ : Offset Master/Slave = MeanPathDelay  
 $\Delta ts$ : Zeitdauer seit letzter Berechnung  
 $tps$ : Anzahl Tick in einer Sekunde

Abbildung 3: Berechnung des Fehlers der internen Uhr

Bei solchen einfachen Regelalgorithmen kommt es immer wieder zum Aufschwingen des Systems, da der Regelalgorithmus in Abhängigkeit vom ermittelten Fehler den Korrekturwert entsprechend anwendet. Um ein solches Aufschwingen zu vermeiden, betrachtet der TSEP IEEE 1588 Regelalgorithmus die 1. Ableitung des MeanPathDelays.

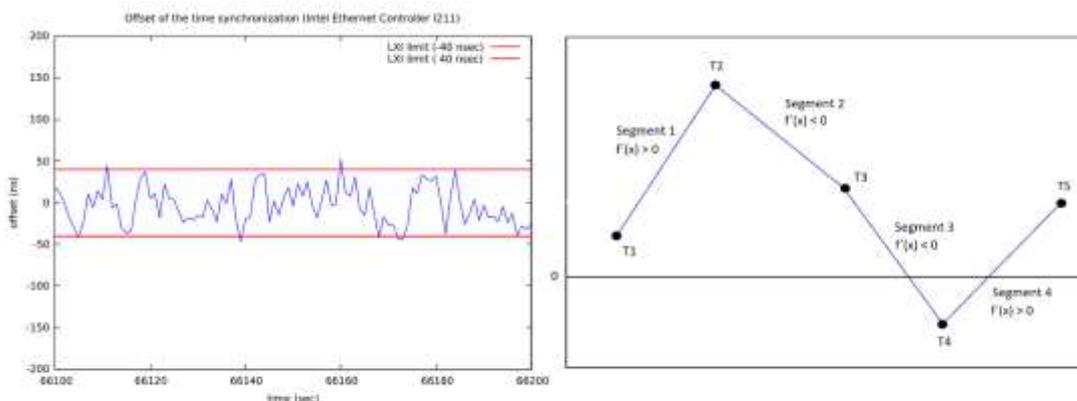


Abbildung 4: Erweiterter Regelalgorithmus

Wie in dem obigen Diagramm zu sehen ist kann der Regelalgorithmus bereits am Ende von Segment 2, also bei T3, erkennen, dass die entsprechende Regelung greift, und die Anpassung entsprechend korrigieren.

Mit dem TSEP Standard Algorithmen lassen sich unabhängig von Fehlermodellen und Hardware-Topologien einfache und funktionierende Systeme aufbauen. Natürlich sind hierbei Abstriche in der Genauigkeit zu machen. Es lassen sich jedoch einfach System aufbauen die eine Genauigkeit von +/- 80ns im mittel erreichen lassen. Bei entsprechender Topologie sogar +/- 40 ns.

## Anwendungsbeispiele / Vorteile:

Da in der heutigen Zeit Messaufgaben immer komplexer werden und die Anzahl der interagierenden Messgeräte steigt, können die bestehenden zentralistischen Messkonzepte nur noch schwer aufrechterhalten werden. Zusätzlich sind die Anforderungen an die Laufzeiten in Messsystemen deutlich gestiegen. War früher ein Trigger Kabel mit 10 Meter Länge und einer Laufzeit von 40ns akzeptabel, sind heute diese Verzögerungen nicht mehr tragbar. Da aber die Messaufgaben immer komplexer werden, verlängern sich auch die Wege zwischen den einzelnen Komponenten. Letztendlich ein Teufelskreis. Mit Hilfe von IEEE 1588 kann dieser Teufelskreis durchbrochen werden. Trigger der IEEE 1588 Komponente können, wenn diese mit einer hochgenauen Zeit synchronisiert wird, selbstständig Messaufgaben ohne einen externen Trigger ausführen. Doch nicht nur in der Messtechnik, auch in der Automations- und Produktionstechnik, kann ein zeitgesteuertes Zusammenspiel von verschiedenen Komponenten durchaus interessant sein.

Da nun seit einigen Jahren die notwendige Hardware als Consumer Chip von Intel verfügbar ist, kann diese Technologie auch ohne Bedenken eingesetzt werden. Auf der Softwareseite ist der TSEP IEEE 1588 Stack eine günstige, plattformübergreifende und performante Lösung für ihre Systeme.

## Lizenzmodelle:

Grundsätzlich benötigt der TSEP IEEE 1588 Stack auf jedem Gerät eine Lizenz. Bei vorkompilierten Versionen wird für jeden Kunden individuell die Kundenlizenznummer in die Binärdateien geschrieben, somit kann nachvollzogen werden welcher Kunde welches Gerät mit dem TSEP IEEE 588 Stack ausgestattet hat.

TSEP hat für den TSEP IEEE 1588 Stack zwei Lizenzmodelle.

Das erste Modell ist für Kunden gedacht die eher eine geringe Stückzahl an Geräten mit IEEE 1588 verkaufen wollen. Für diese Kunden stellt TSEP eine vorkompilierte Version des TSEP IEEE 1588 Stack bereit, die über eine Stückzahlregelung verkauft werden müssen. TSEP nutzt keinerlei Dongle oder ähnliches für die Lizenzverifikation bei diesem Lizenzmodell. Die Staffellungen für dieses Lizenzmodell sind 100/500/1000 Lizenzen.

Das zweite Lizenzmodell richtet sich an Kunden die keine Stückzahlbindung an den TSEP IEEE 1588 Stack haben wollen. Bei diesem Modell erhält der Kunde den kompletten Source Code des Stacks und der Treiber. Der Kunde kann dann unbefristet beliebig viele Geräte mit dem TSEP IEEE 1588 Stack ausstatten. Zusätzlich ist bei diesem Modell ein mehrjähriger Supportvertrag enthalten, der dem Kunden Unterstützung und Neuerungen am TSEP IEEE 1588 Stack zusichert.