



TSEP

Technical Software Engineering Plazotta

Innovation made measurable.

Poseidon

Beste Performance bei maximaler Modularität

In der heutigen Testautomation wird eine schnelle Command- und Queryausführung von Messgeräten gefordert. Poseidon von TSEP optimiert den zeitaufwendigen Prozess der Kommunikation des Messgerätes mit dem Client des Prüfsystems. Das innovative Design zielt auf die Optimierung der Befehlsausführung auf allen systematischen Ebenen ab. Die Performance von Poseidon basieren auf der baumartigen Datenstruktur der SCPI-Befehle und der Priorisierung der SCPI-Befehle während der Laufzeit auf Basis einer Zugriffsstatistik. Parallel dazu bietet Poseidon ein einfach zu integrierendes, flexibles und zuverlässiges Framework. Die Generierung und Integration von eigenen SCPI-Befehlen und Parsern kann mit einem XML-Generator durchgeführt werden.

Highlights



Performante Kommandoabarbeitung und schneller Daten Transfer



Standard Schnittstellen: TCP/IP, RS232 und HiSlip



Multi-Instanz Konzept für Parser und Data Channels



Generierung von Code Skelet und Dokumentation mithilfe von XML



Innovative und effiziente SCPI Parameter Architektur



Cross-Plattform kompatibel (Windows und Linux)

Kombination aus Effizienz und einem Multi-Instanz-Konzept

Mit Poseidon können Test- und Messgeräte einfach, effektiv und kostengünstig mit einem SCPI-konformen Parser ausgestattet werden. Poseidon enthält einen SCPI 488-konformen Parser, der die entsprechenden Befehle übersetzt und an die Geräte-Firmware weiterleitet.

Poseidon ist speziell für die schnelle Ausführung von Befehlen und Abfragen sowie die Datenübertragung konzipiert. Auf Intel I5-Rechnern sind Antwortzeiten (z. B. *IDN?) von weniger als 400µs mit TCP/IP und Localhost-Loopback kein Problem. Bei High-End-Geräten mit I7 und entsprechender CPU-Taktung können diese Zeiten noch einmal minimiert werden. Im Labor wurden *IDN?-Antwortzeiten von weniger als 100µs gemessen. TSEP verbessert kontinuierlich die Performance von Poseidon, da der Faktor Performance ein wichtiges Schlüsselmerkmal ist.

Die Modularität von Poseidon spiegelt sich in der Architektur wider. Die drei Schlüsselkomponenten sind die Kanäle, die Parser und ein Core Modul. Der Kommunikationskanal implementiert die Hardware Kommunikationsschnittstelle. Das Multi-Instanz-Kanal-Konzept kann mehrere aktive Verbindungen gleichzeitig abarbeiten. Kunden können auch unabhängige Kanäle definieren und implementieren. Der Parser definiert einzelne SCPI-Befehle, deren Syntax und Verarbeitung. Poseidon unterstützt mehrere Parser zur gleichen Zeit. Dies ermöglicht dem Kunden, seine SCPI-Kommandostruktur modular aufzubauen und innerhalb anderer Gerätevarianten wiederzuverwenden. Alle Kanäle übertragen ihre Daten an den Kern von Poseidon und werden dann vorverarbeitet und an den entsprechenden Parser weitergeleitet. Die Ergebnisse des Parsers werden dann über den Core an die Geräte-firmware kommuniziert.

Poseidon Struktur

Allgemein

Poseidon basiert auf einem objektorientierten Ansatz, der konsequent umgesetzt wurde. Alle notwendigen Schnittstellen werden über Vererbung an die entsprechende Gerätefirmware weitergegeben. Da SCPI Commands und Querys immer auf Gerätedaten verweisen, wurde eine Herangehensweise für den Zugriff auf diese Daten definiert und implementiert. Dadurch können gerätespezifische Daten einfach an die entsprechenden Befehle weitergeleitet werden.

Core

Der Kern von Poseidon regelt und steuert die Kommunikation zwischen dem Client und der Firmware. Der Kern kann nicht nur synchrone Befehle verarbeiten, sondern es werden auch asynchrone Befehle unterstützt. Dabei laufen diese Befehle in einem separaten Thread und sind somit in sich abgeschlossen und unabhängig voneinander ausführbar. Die Implementierung dieser asynchronen Befehle unterscheidet sich nicht von den synchronen Befehlen.

Der Core unterstützt außerdem die folgenden Features:

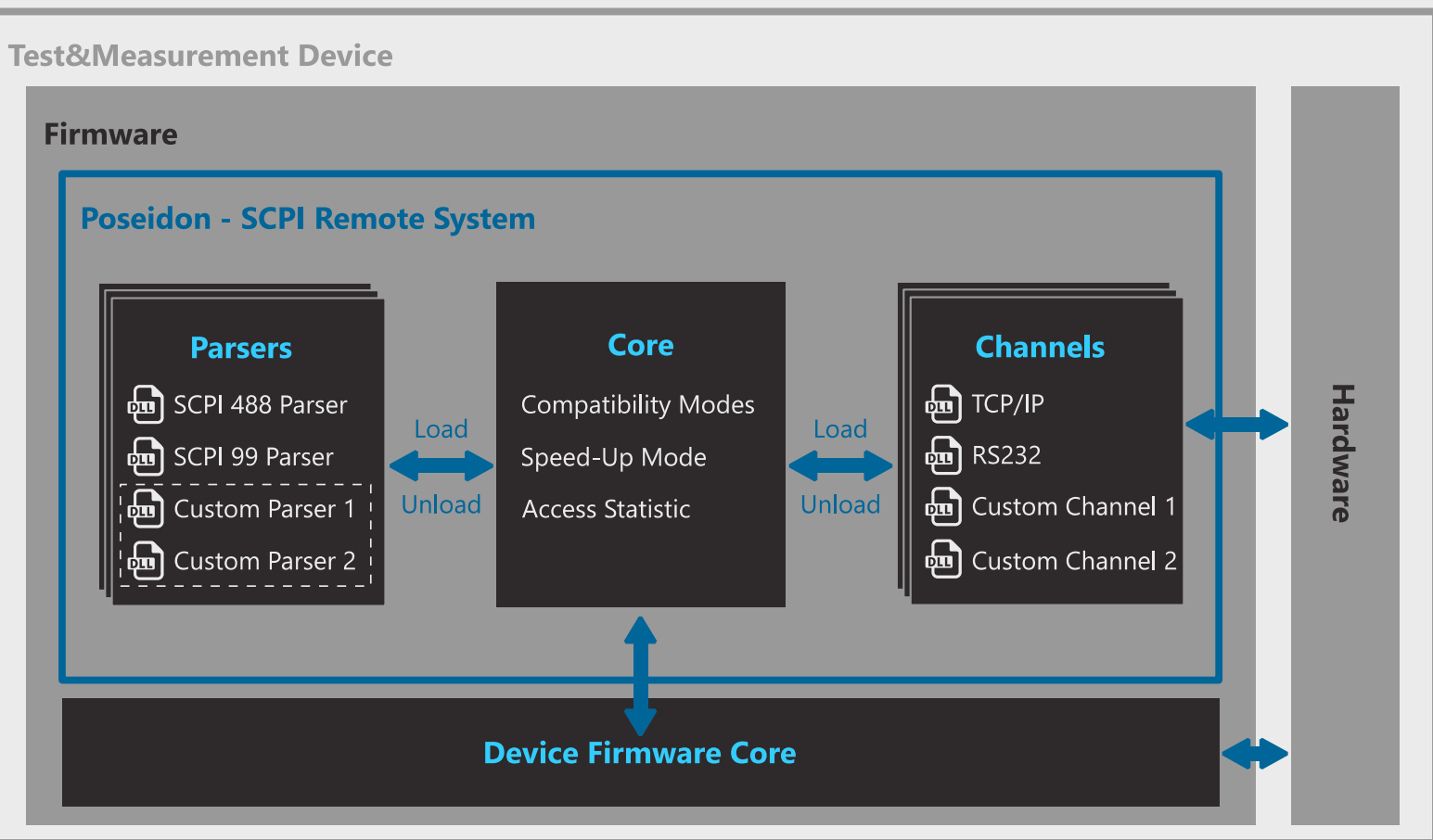
- SCPI-Statusregister;
- Kompatibilitätsmodi zur Benutzung spezieller Funktionen von anderen Geräteherstellern;
- Speed-Up-Modus zur verbesserte Befehlsverarbeitung, mit reduzierter Syntaxprüfung.

Channels

Poseidon definiert Channels als Kommunikation zwischen dem Parser und der Kommunikationshardware. Diese Channels sind die Implementierung des Datenzugriffs von und zur Hardware. Derzeit unterstützt Poseidon die Standardschnittstellen TCP/IP, HiSlip und RS232. Es ist jedoch jederzeit möglich, einen eigenen Channel zu erstellen. Die Erstellung eines eigenen Channels kann mit Hilfe einer detaillierten Beschreibung erfolgen.

In Poseidon können sowohl Single-Instanz Channels als auch Multi-Instanz Channels definiert und implementiert werden. Single-Instanz Channels sind typischerweise Channels, die nur eine Verbindung pro Hardwareressource zulassen. Als Beispiel kann die RS232-Schnittstelle genannt werden. Selbstverständlich können mehrere RS232-Hardwaremodule unterstützt werden. Dazu müssen die einzelnen Single-Instanz-Kanäle mehrfach (d.h. pro Hardware) in Poseidon registriert werden. Multi-Instanz Channels können mehrere aktive Verbindungen gleichzeitig handhaben. Als Beispiel für solche Channels können die TCP/IP- und die HiSlip-Channels aufgelistet werden. Diese Channels können auch mehrfach in Poseidon registriert werden und z. B. über verschiedene TCP/IP-Ports kommunizieren. Die Channels können zur Laufzeit geladen oder aus Poseidon entfernt werden.

Abb.1: Poseidon Einbettung in die Test&Mess-Gerätetopologie



Parser

Die eigentliche Verarbeitung der SCPI-Kommandos erfolgt im entsprechenden Parser. Innerhalb von Poseidon kann eine unbegrenzte Anzahl von Parsern definiert werden. Dadurch können Funktionen getrennt und in anderen Geräten wiederverwendet werden. TSEP stellt mehrere Parser mit Teilen des SCPI 488-2 Befehlssatzes zur Verfügung. Die Befehlssyntax basiert auf dem SCPI 488-2 Standard, enthält aber auch spezifische Erweiterungen, die separat ein- oder ausgeschaltet werden können. Das Parsen von Befehlen in Poseidon ist sehr effektiv und effizient, um eine optimale Verarbeitungszeit zu erreichen. Zusätzlich hat TSEP einige Optimierungsverfahren in die Verarbeitung integriert. So werden die Befehle nach der Anzahl der Aufrufe sortiert, wodurch häufig benutzte Befehle schneller abgearbeitet werden können. Darüber hinaus wurde die SCPI-Befehlsverarbeitung für Multicore- und Multi-Threaded-CPU konzipiert, so dass eine parallele Verarbeitung möglich ist, was erneut die Verarbeitungszeit reduziert. Poseidon unterstützt auch die Verarbeitung von überlappenden Befehlen, d.h. Befehle müssen nicht beendet werden, bevor der nächste Befehl ausgeführt werden kann. Diese Befehle können praktisch parallel ausgeführt werden. Die Implementierung dieser Befehle unterscheidet sich nur geringfügig von den "nicht überlappenden" Befehlen, lediglich die Möglichkeit, auf einen Abbruch während der Ausführung zu reagieren, muss programmtechnisch hinzugefügt werden. Die Parser erlauben auch die Verwendung von Alias-Befehlen. Diese Alias-Befehle verwenden eine andere Syntax, aber mit den gleichen Funktionalitäten und Parametern. Solche Konstruktionen sind notwendig, um ältere Befehle (Kompatibilitätsmodus) aus bestehender Messsoftware zu unterstützen.

Effiziente und innovative SCPI Parameter Architektur

Für Poseidon wurde ein vollständig objektorientierter Ansatz zur Verarbeitung der SCPI Command und Query Parameter gewählt. Diese Parameter werden innerhalb von Poseidon immer über objektorientierte Klassen abgebildet. Dieser Ansatz vereinfacht die Verarbeitung und Erweiterung von Parametern dramatisch. Poseidon stellt die notwendigen Basisparameter wie numerisch, float oder string zur Verfügung. Darüber hinaus sind einige zusätzliche Parameter definiert. So sind z. B. Arrays von Parametern möglich, was die Arbeit z. B. für Konfigurationsbefehle (Antennenparameter, Kalibrierdaten, etc.) erleichtert. Die Verarbeitung von Sequence Containern ist ebenfalls möglich.

XML Generator

Die SCPI-Befehle werden innerhalb von Poseidons über XML-basierte Definitionen erstellt und verwaltet. Von diesem Generator werden nicht nur die notwendigen Quellcodeteile erstellt, sondern auch die notwendige Dokumentation generiert. Innerhalb der XML-Dateien werden die Befehlssyntaxe, Attribute und Beschreibungen definiert. Somit werden alle befehlspezifischen Definitionen zentral gespeichert. Mit Hilfe des Generators werden die notwendigen Sourcecode-Skelette für die Einbettung in die Firmware erzeugt. Der Kunde muss nur die eigentliche Funktionalität implementieren, ohne sich um die Abarbeitung der Befehle zu kümmern. Alle SCPI-Kommandos sind als C++-Klassen implementiert. Alle notwendigen Header und das spezifische Parser Framework werden automatisch erzeugt. Die in XML definierten Beschreibungen sind ebenfalls in den Skeletten des Quellcodes enthalten. Auch die Client-Seite der SCPI-Kommunikation kann mit Hilfe des Generators erstellt werden. Analog zur Server-Seite (Test- und Messgerät) werden auch die Skelette des Quellcodes generiert. Der Generator kann auch zur Erstellung der Dokumentation der in den Parsern enthaltenen Befehle verwendet werden. Der Generator erzeugt die entsprechende Dokumentation auf Basis der XML-Definitionsdatei und einer Word-Vorlage. TSEP stellt eine Vorlage zur Verfügung, die zur Erstellung eines eigenen Dokuments verwendet werden kann. Der Kunde kann diese Vorlage jedoch jederzeit ändern und an seine Wünsche und sein CI anpassen.

Die mitgelieferten Parameter reichen im Regelfall nicht aus, um alle Kundenbedürfnisse abzudecken, deshalb können die Parameter jederzeit vom Kunden erweitert und zur Laufzeit in das Poseidon geladen werden. Es gibt mehrere Beispiele für kundeneigene Parameter, die den Einsatz von Kunden zeigen.

Bei der Generierung des Sourcecode-Skeletts integriert der Generator alle verfügbaren Parameter in den Befehl, was dem Entwickler bei der Implementierung hilft. Innerhalb des Befehls kann auf die Parameter über einen Casting-Mechanismus einfach zugegriffen werden. Der Zugriff auf die Parameterdaten erfolgt über Get/Set-Mechanismus, die mit einer entsprechenden Signatur versehen sind.

Variable Deployment Modelle

Poseidon wurde komplett in C++ erstellt und basiert auf dem C++11 Standard. Zur Generierung der Projektdaten (Visual Studio Solution/Projektdateien, Makefiles oder Eclipse-Projekte) wurde CMake verwendet, um den unterschiedlichen Entwicklungs-umgebungen Rechnung zu tragen. Dadurch ist es einfach, neue Entwicklungsumgebungen einzuführen oder Änderungen an bestehenden Projekten vorzunehmen.

Poseidon ist sowohl unter allen Windows Betriebssystemen (z. B. 7 und 10) als auch unter Linux Betriebssystemen (z. B. Ubuntu 16.04 und Ubuntu 20.4) verfügbar. Der Quellcode ist für beide Plattformen identisch (Common Source), was die Wartung und Integration vereinfacht.

Zukünftige Features

HiSlip 2.0 Channel

Als aktives Mitglied der IVI Stiftung überwacht TSEP die Entwicklung des neuen Sicherheitsstandards HiSlip 2.0. Da der HiSlip 2.0 Standard eingeführt wurde, wird TSEP einen entsprechenden Channel für Poseidon zur Verfügung stellen.

Secure TCP/IP Channel

Sicherheit für TCP/IP ist heutzutage ein wichtiges Thema. TSEP befasst sich mit diesem Thema und wird einen sicheren TCP/IP-Kanal für das Poseidon entwickeln.

VISA Library

TSEP wird eine proprietäre VISA-Library für die Kommunikation mit beliebigen Messgeräten anbieten. Um die Nutzung dieses Clients zu vereinfachen und zu standardisieren, wird TSEP den Client entsprechend den VISA Standards erweitern.

Python Plugins

Integrieren Sie neue benutzerdefinierte SCPI-Befehle und Kanäle mit einem einfachen und schnellen Python Plugin-System und das ohne die Firmware neu zu erstellen. Die Plugins werden in Python erstellt, so dass nur eine geringe Einarbeitungszeit und ein geringer Kenntnisstand erforderlich sind. Dadurch wird die Zeit bis zur Einsatzbereitschaft drastisch reduziert.

Bestellinformationen

Software Komponenten

Bestellref.	Beschreibung
PS-B	Binär Lizenz
PS-SC	Source Code Lizenz

Dienstleistungen

Bestellref.	Beschreibung
PS-SUP-1	Support für 1 Jahr
PS-SUP-3	Support für 3 Jahre
PS-UPD-1	Updates für 1 Jahr
PS-UPD-3	Updates für 3 Jahre
PS-DL	Dienstleistungen



TSEP

Technical Software
Engineering Plazotta

For more information visit www.tsep.com or contact us.

Technical Software Engineering Plazotta

Hopfenstr. 30
85283 Wolnzach
Deutschland

Tel: +49 8442 96240 0
Fax: +49 8442 96240 99
E-Mail: info@tsep.com