



Poseidon

Best Performance with Maximum Modularity

In today's test automation there is a demand for fast command and query execution of measurement instruments. TSEP'S Poseidon optimizes the measurement instruments time-consuming process of communication with test system clients. The innovative design aims for the optimization of the command execution on all systematic levels. Optimization capabilities of the Poseidon are based on the tree-like data structure of the SCPI-commands and the prioritization of the SCPI-commands during the run time on grounds of an access statistic. Parallely Poseidon provides an easy to integrate, flexible, and reliable framework. The generation, integration, and documentation of custom SCPI-commands and parsers can be done with an XML-generator.

Highlights



High-Performance Command Execution and Data Transfer



Default Data Channels: TCP/IP, RS232 and HiSlip (only for LXI Members)



Multi-Instance Concept for Parser and Data Channels



Generation of Code Skeletons and Documentation with XML



Innovative and Efficient SCPI Parameter Architecture



Cross-Platform Compatible (Windows and Linux)

Efficiency Combined with a Multi-Instance Concept

With Poseidon, test and measurement instruments can be easily, effectively, and economically equipped with a SCPI compliant parser. Poseidon contains a SCPI 488 compliant parser, which translates the corresponding commands and passes them on to the device firmware. Poseidon is specially designed for fast command and query execution and data transfer. On Intel I5 computers, response times (eg. *IDN?) of less than 400µs with TCP/IP and localhost loop-back are no problem. For high-end devices with I7 and appropriate CPU clocking, these times can be minimized once again. In the lab *IDN? response times of less than 100µs have been measured. TSEP continuously improves the performance of Poseidon since the factor performance is an important key feature.

The modularity of Poseidon is reflected in the architecture. The three key components are the channels, the parsers, and a core module. The communication channel implements the hardware-communication interface. The multi-instance channel concept can handle multiple active connections at the same time. Customers can also define and implement independent channels. The parser defines individual SCPI commands, their syntax, and processing. Poseidon supports multiple parsers at the same time. This allows the customer to modularly build up their SCPI command structure and to reuse them within other device variants. All channels transfer their data to the core of Poseidon and are then preprocessed and forwarded to the corresponding parser. The results from the parser are then communicated to the device firmware via the core.

Poseidon Structure

General

Poseidon is based on an object-oriented approach, which has been consistently implemented. All necessary interfaces are passed on to the corresponding device firmware via inheritance. Since SCPI commands and queries always reference device data, an approach to access this data has been defined and implemented. This allows device-specific data to be simply forwarded to the corresponding commands.

Core

The core of Poseidon manages and handles the communication between the client and the firmware. The core can not only process synchronous commands, but also supports asynchronous commands. Here, these commands run in a separate thread and are thus self-contained and independently executable. The implementation of these asynchronous commands does not differ from the synchronous commands.

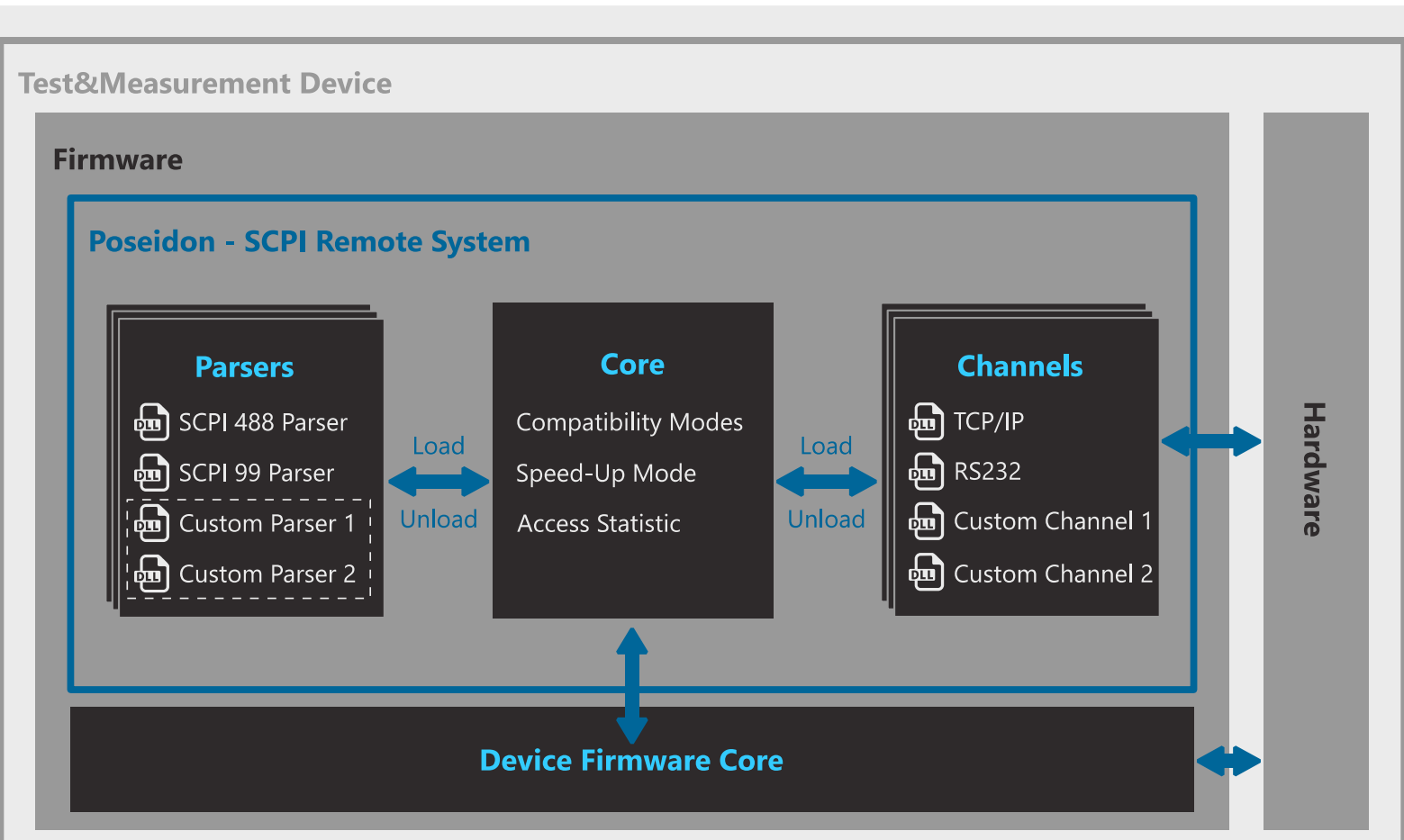
The core also supports the following features:

- SCPI status register;
- Compatibility modes to emanate special features from other instrument manufacturers;
- Speed-Up mode, improved command processing.

Channels

Poseidon defines channels as communication between the parser and the communication hardware. These channels are the implementation of the data access from and to the hardware. Currently, Poseidon supports the standard interfaces TCP/IP, HiSlip and RS232. However, it is always possible for a customer to create their own channel. The creation of a custom channel can be done with the help of a detailed description. In Poseidon, singleinstance channels can be defined and implemented as well as multiinstance channels. Singleinstance channels are typically channels that allow only one connection per hardware resource. As an example, the RS232 interface can be mentioned. Of course, several RS232 hardware modules can be supported. For this purpose, the individual singleinstance channels have to be registered several times (i.e. for each hardware) in Poseidon. Multiinstance channels can handle multiple active connections at the same time. As an example for such channels, the TCP/IP and the HiSlip channels can be listed. These channels can also be registered several times in Poseidon and, for example, communicated via different TCP/IP ports. Channels can be loaded at runtime or removed from Poseidon.

Fig. 1. Poseidon embedding in the test and measurement device topology



Parser

The actual processing of the SCPI commands is carried out in the corresponding parser. Within Poseidon, an unlimited number of parsers can be defined. This means that functionalities can be separated and reused in other devices. TSEP provides several parsers with parts of the SCPI 488-2 command set. The command syntax is based on SCPI 488-2, but also includes specific extensions which can be separately switched on or off. Parsing commands in Poseidon is highly effective and efficient, so that an optimal processing time is achieved. In addition, TSEP has incorporated some optimization procedures into the processing. Thus, commands are sorted according to their number of calls, which ensures that frequent commands can also be processed more quickly. In addition, SCPI command processing has been designed for multicore and multi-threaded CPUs, which means that parallel processing is feasible, which in turn contributes to the optimization of the processing time. Poseidon also supports the processing of overlapped commands, these commands do not need to be finished before the next command starts executing. These commands can run virtually parallel. The implementation of these commands differs only marginally from the "non-overlapped" commands, only the possibility to react to an abortion during execution has to be added programmatically. The parsers also allow the use of Alias commands. Alias commands are commands which use a different syntax, but with the same functionalities and parameters. Such constructions are necessary to support older commands (compatibility mode) from existing measurement software.

Efficient and innovative SCPI Parameter Architecture

For Poseidon it became a completely object-oriented one approach for processing the SCPI command and query parameters chosen. Within Poseidon, these parameters are always mapped using object-oriented classes. This approach dramatically simplifies the processing and expansion of parameters. Poseidon provides the necessary basic parameters such as numeric, float or string. In addition, some additional parameters are defined. So are for example arrays of parameters possible what the work for configuration commands (antenna parameters, calibration data, etc.). The processing of sequence containers is also possible.

Variable Deployment Models

Poseidon was created completely in C++ and is based on the C++11 standard. CMake was used to generate the project data (Visual Studio Solution/project files, makefiles or Eclipse projects) to take account of the different development environments. This makes it easy to introduce new development environments or to make changes to existing projects.

XML Generator

The SCPI commands are created and managed within Poseidon via XML-based definitions. Not only the necessary source code parts are created by this generator, but also the necessary documentation is generated. Within the XML files, not only the command syntax is defined, attributes and descriptions are also stored. Thus, all command-specific definitions are stored centrally. Using the generator, the necessary source code skeletons are generated to embed in the firmware. The customer only has to worry about the implementation of the actual functionality. All SCPI commands are implemented as C++ classes. All necessary headers and the specific Parser Framework are created automatically. The descriptions defined in XML are also included in the source code skeletons. Also, the client-side of the SCPI communication can be created using the generator. In analogy to the server-side (measuring device), the source code skeletons are generated as well. The generator can also be used to create the documentation for the commands contained in the parsers. The generator generates the corresponding documentation based on the XML definition file and a Word template. TSEP provides a template that can be used to create your own document. The customer can, however, change this template at any time and adapt it to their wishes and CI.

The parameters supplied are usually not sufficient to cover all customer needs, so the customer can extend the parameters at any time and load them into the Poseidon at runtime. There are several examples of custom parameters that show customer use.

When generating the source code skeleton, the generator integrates all available parameters into the command, which helps the developer with the implementation. Within the command, the parameters can be easily accessed via a casting mechanism. The parameter data is accessed via a get / set mechanism, which is provided with a corresponding signature..

Poseidon is available on all Windows (e.g. 7 and 10) and Linux (e.g. Ubuntu 16.04 and Ubuntu 20.04) systems. The source code is identical for both platforms (Common Source), which simplifies maintenance and integration

Upcoming Features

HiSlip 2.0 Channel

As an active member of the IVI Foundation, TSEP monitors the development of the new HiSlip 2.0 security standard. As the HiSlip 2.0 standard has been launched, TSEP will provide a corresponding channel for Poseidon.

Secure TCP/IP Channel

Security for TCP/IP is a prominent issue nowadays. TSEP is addressing this issue and is going to develop a Secure TCP/IP Channel for the Poseidon.

VISA Library

TSEP will offer a proprietary VISA library for communication with any measurement instrument. To simplify and standardize the use of this client, TSEP is going to extend the client according to the VISA standards.

Python Plugins

Integrate new custom SCPI commands and channels with an easy and fast Python plugin system such that no rebuild of the firmware is necessary. The plugins are created in Python, so only a low training time and a low skill level are necessary. Hence, the time until ready to use is reduced drastically.

Order Information

Software Components

Order ref.	Description
PS-B	Binary Licenses
PS-SC	Source Code Licenses

Services

Order ref.	Description
PS-SUP-1	Support for 1 year
PS-SUP-3	Support for 3 years
PS-UPD-1	Updates for 1 year
PS-UPD-3	Updates for 3 years
PS-DL	Consulting



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
E-Mail: info@tsep.com