

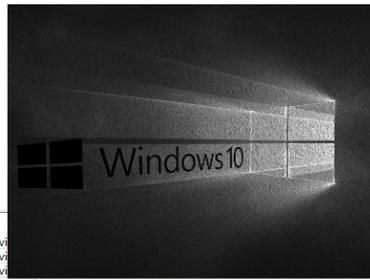


TSEP

Technical
Software
Engineering
Plazotta

Product Description Remote System (SCPI Parser)

HiSlip
SCPI 488-2

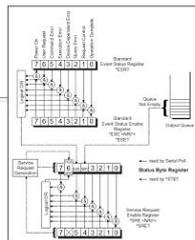


Visa Resource
String



Line	Duration	Status	Operation
4	119 ms	VI_SUCCESS	viRead(sessionid= 4, buf= "", cnt= 1024, retCnt= 25)
5	46 µs	VI_SUCCESS	viWrite(sessionid= 4, buf= "TSEP 98765T,112233,V1.10\n", cnt= 9, retCnt= 9)
6	60 µs	VI_SUCCESS	viRead(sessionid= 4, buf= "SYST:ERR?", cnt= 9, retCnt= 9)
7	7 ms	VI_SUCCESS	viWrite(sessionid= 4, buf= "0, No error\n", cnt= 1024, retCnt= 13)
8	45 µs	VI_SUCCESS	viRead(sessionid= 4, buf= "", cnt= 1024, retCnt= 0)
9	9 ms	VI_SUCCESS	viWrite(sessionid= 4, buf= "ESE 2\n", cnt= 7, retCnt= 7)
10	1 s	VI_ERROR_TMO	viRead(sessionid= 4, buf= "", cnt= 1024, retCnt= 0)
11	45 µs	VI_SUCCESS	viWrite(sessionid= 4, buf= "IDN?", cnt= 6, retCnt= 6)
12	2 s	VI_ERROR_TMO	viRead(sessionid= 4, buf= "TSEP 98765T,112233,V1.10\n", cnt= 1024, retCnt= 25)
13	44 µs	VI_SUCCESS	viWrite(sessionid= 4, buf= "SYST:ERR?", cnt= 9, retCnt= 9)
14	6 ms	VI_SUCCESS	viRead(sessionid= 4, buf= "0, No error\n", cnt= 1024, retCnt= 13)
15	54 µs	VI_SUCCESS	viWrite(sessionid= 4, buf= "IDN?", cnt= 6, retCnt= 6)
16	586 µs	VI_SUCCESS	viRead(sessionid= 4, buf= "0, No error\n", cnt= 1024, retCnt= 13)

*IDN?



The TSEP Remote System provides a SCPI compatible parser for test and measurement devices. Using the TSEP Remote System, SCPI-488 compliant commands can be processed over different communication channels.

General:

With the TSEP Remote System, measuring instruments can be easily and effectively equipped with a SCPI compliant parser. The TSEP Remote System contains a SCPI 488 compliant parser, which translates the corresponding commands and passes them on to the device firmware.

The TSEP Remote System differs between channels and parsers. The communication channel is referred to as a channel and implements the hardware-technical interface for the parser. Customers can always define and implement independent channels. The channels TCP/IP, RS232 and for LXI members the HiSlip channel are supplied as standard. TCP / IP and HiSlip support both IPV4 and IPV6.

The parser defines the individual SCPI commands, the syntax and their processing. The TSEP Remote System can support multiple parsers at the same time. This allows the customer to modularly build up their SCPI commands and to reuse them with other device variants. By default, the TSEP Remote System contains multiple parsers that contain some of the 488-2 compliant SCPI commands.

The TSEP Remote System is available on both Windows and Linux. The source code is identical for both platforms (Common Source), which simplifies maintenance and integration into the device software.

The TSEP Remote System is especially designed for fast command execution. On Intel I5 computers, response times (eg. *IDN?) of less than 400 μ s with TCP/IP and localhost loopback 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 100 μ s have been measured. TSEP works on a continuous improvement of the performance of the TSEP Remote System since the factor performance is an important key feature.

For the documentation of the SCPI commands, TSEP has created its own tool, which generates free-formatted Word documents from the XML definitions of the commands. The customer gets this tool in the source code and can customize it for his CI and purposes. The tool has been created in C#, so only a low training time and a low skill level are necessary.

TSEP Remote System / Basics:



The TSEP Remote System 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) in order to take account of the different development environments. This makes it easy to introduce new development environments or to make changes to existing projects.

The TSEP Remote System 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 de-fined and implemented. This allows device-specific data to be simply forwarded to the corresponding commands.

The TSEP Remote System can not only process synchronous commands, but also asynchronous commands are supported. 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 synchronically commands.

The TSEP Remote System also supports the following features:

- Short and long form of a command
- Concatenated commands
- Suffixes in commands
- Alias Commands for compatibility modes
- Check the syntax of parameters directly in the parser
- User-specific command parameters
- Built-in error handler
- SCPI status register
- Overlapped Commands
- Compatibility modes to emanate special features from other instrument manufacturers
- Speed-Up mode, improved command processing, with reduced syntax check

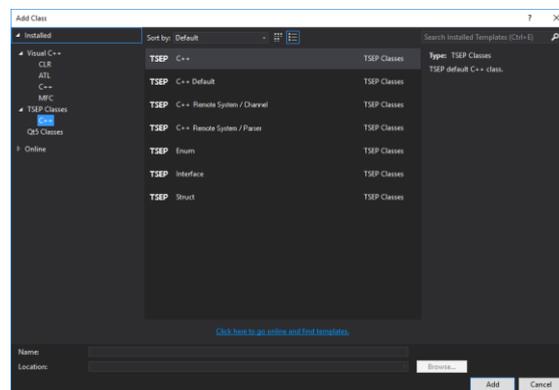
A separate SDK (software development kit) is available for the development of your own commands or channels, providing the necessary tools and software.

TSEP Remote System / Channels:



The TSEP Remote System defines the channels for communication between the parser and the actual hardware. These channels implement access to the data from and to the communication hardware. Currently the TSEP Remote System supports the standard interfaces TCP/IP, HiSlip (only for LXI members) and RS232.

However, it is always possible for a customer to create his own channel. To create your own channel, a generic class wizard is available for Visual Studio.



A detailed description is available for Linux development systems. In the TSEP Remote System, single-instance channels can be defined and implemented as well as multi-instance channels.

Single-instance channels are typically channels of communication 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 single-instance channels have to be registered several times (ie. for each hardware) in the TSEP Remote System.

Multi-instance 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 the TSEP Remote System and, for example, communicated via different TCP/IP ports.

All registered channels transfer their data to the TSEP Remote System and are then processed by the corresponding parser. The results are then forwarded to the responsible channel.

Channels can be loaded at runtime or removed from the TSEP Remote System.

TSEP Remote System / Parser:



The actual processing of the SCPI commands is carried out in the corresponding parser. Within the TSEP Remote System, any 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 ex-tensions which can be separately switched on or off.

Parsing commands in the TSEP Remote System 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 the frequent commands can also be processed more quickly. In addition, SCPI command processing has been de-signed for multi-core and multi-threaded CPUs, which means that parallel processing is feasible, which in turn contributes to the optimization of the processing time.

The TSEP Remote System also sup-ports the processing of overlapped commands, these are commands which do not wait for their termination. 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 can be added programmatically.

The parsers also allow the use of Alias commands. Alias commands are commands with a different syntax, but with the same functionalities and parameters. Such constructions are necessary to support older commands (compatibility mode) and existing measurement software.

Analog to the channels, the user can independently create his own parser. The user is supported by the XML generator, which generates the necessary source code skeletons for all defined SCPI commands.

TSEP Remote System / XML Command Definition:

```
<ScpiParser>
  <ScpiParserData
    ParserName="ScpiParser488"
    ParserLongName="Scpi Parser for IEEE 488.2 Commands"
    ParserDescription="The parser contains the Scpi commands as defined in the IEEE488.2." />
  <ScpiCommands>
    <ScpiSubsystem
      SubSystemName="Common Commands"
      SubSystemDescription="Description of the Common Commands" >
      <ScpiCommand
        ClassName="CLSCCommand"
        ClassLongName="Clear Status Command"
        ClassDescription="Clears the Operation status register, Questionable status register"
        String="CLS"
        Overlapping="false"
        Type="Command" >
        <Examples>
          <Example Name="CLS" Description="Clears all status data." />
        </Examples>
      </ScpiCommand>
    </ScpiSubsystem>
  </ScpiCommands>
</ScpiParser>
```

The SCPI commands are created and managed within the TSEP Remote System 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 for embedding 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..

```
/**
 * @class CLSCCommand
 * @brief The class defines the parser for Clear Status Command
 * Clears the Operation status register, Questionable status register and the Error/Event queue.
 */
class TSEP_SCPI_PARSER_API CLSCCommand : public IScpiCommand
{
public:
  /**
   * @brief Constructor
   * The constructor adds all defined command parameters to the command
   */
  CLSCCommand(void)
  : IScpiCommand(eScpiCommandType::Command, "CLS", "", "", "") {}

  /**
   * @brief Destructor
   */
  virtual ~CLSCCommand(void) {}

  /**
   * @brief Executes the defined SCPI command
   * This method is called from the parser if the defined SCPI command sequence is detected.
   * All relevant data for the command execution is transferred to and from the method via the
   * cScpiCmdData structure.
   * If the command succeeds the method returns true, otherwise false is returned.
   * @param cScpiCmdData Contains all command relevant data and retrieves all result data.
   */
  TSEP_Boolean ExecuteCommand(CScpiCommandData& cScpiCmdData);
};
```

In addition, the client side of the SCPI communication can also 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 on the basis of the XML definition file and a Word template. TSEP provides a template which 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.

TSEP Remote System / SCPI Parameter:



For the TSEP Remote System a completely object-oriented approach for processing the SCPI command parameters has been chosen. Parameters of SCPI commands or queries are always mapped within the TSEP Remote System via object-oriented classes. This approach dramatically simplifies the processing and extension of parameters.

The TSEP Remote System provides the necessary basic parameters, such as numeric, float or string, as a basic implementation. In addition, a whole series of additional standard parameters are defined. For example, arrays of parameters are possible, which makes the work easier, especially for some configuration commands (antenna parameters, calibration data, etc.).

The processing of enumeration parameters is, of course, also possible in the TSEP Remote System. There are a number of examples that the customer can adapt and modify at any time.

Since the parameters supplied with the TSEP Remote System are not sufficient to cover all customer requirements, the parameters can also be extended by the customer at any time and loaded at runtime into the TSEP Remote System.

The access to the data of the command parameter is made using Get/Set methods, which are equipped with the corresponding signature.

Within commands, the parameters can be easily accessed using a parameter cast. When generating the source code skeleton, the generator can integrate all available parameters into the command, which helps the developer in the implementation.

Price:

All prices are in Euro and excl. VAT. With each license you can use as many devices as you want with the TSEP Remote System. The license does not refer to an operating system.

The binary license contains all the necessary sources to create your own parsers and channels. The TSEP Remote System is delivered in a binary version for the operating systems Windows 7 and 10 and Linux Ubuntu 14.04 / 16.04. Other operating system on request.

The source code license contains all the sources of the complete TSEP Remote System.

Binary License:

- **TSEP Remote System** **9.999,-- €**
- **TSEP Remote System Support (Telephone + Mail)** **per year 2.400,-- €**

Source Code License:

- **TSEP Remote System + 1 Jahr Support + Updates** **24.999,-- €**
- **TSEP Remote System + 3 Jahr Support + Updates** **35.499,-- €**
- **TSEP Remote System Support (Telephone + Mail)** **per year 2.400,-- €**
- **TSEP Remote System Support + Updates** **per year 4.800,-- €**