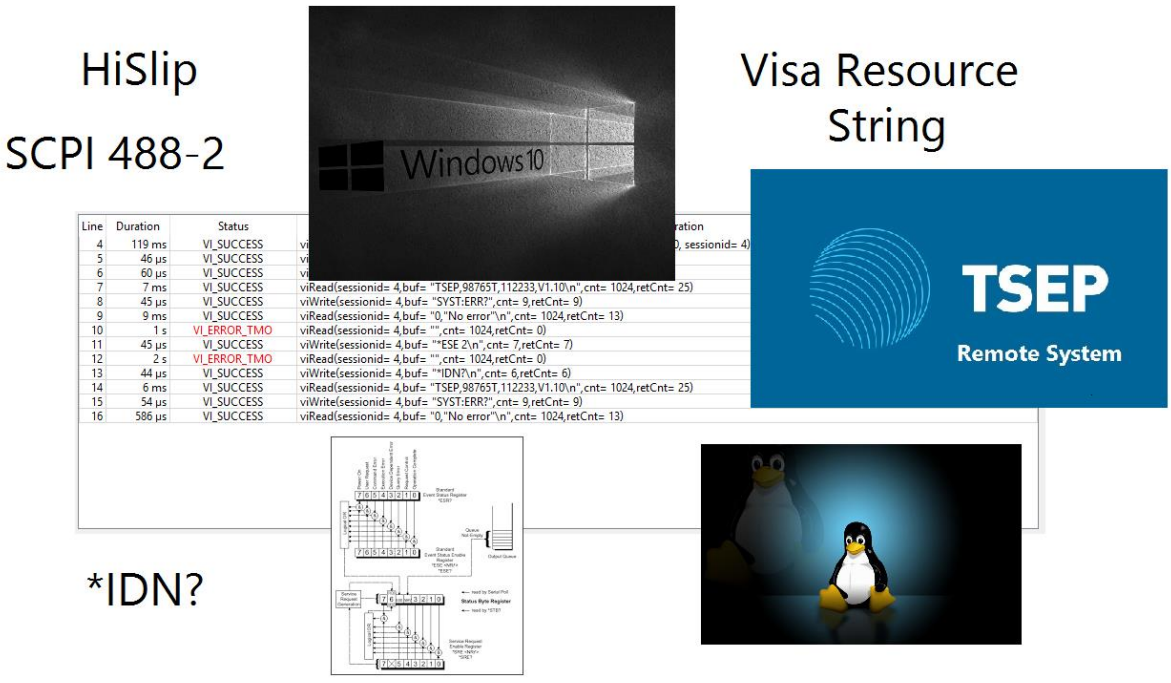
TSEP

Technical
Software
Engineering
Plazotta

Produktbeschreibung
Remote System (SCPI Parser)



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

Das TSEP Remote System stellt einen SCPI kompatiblen Parser für Messgeräte zur Verfügung. Mit Hilfe des TSEP Remote Systems können SCPI-488 konforme Befehle über unterschiedliche Kommunikationskanäle verarbeitet werden.

Einführung:

Mit dem TSEP Remote System können Messgeräte einfach und effektiv mit einem SCPI konformen Parser ausgestattet werden. Das TSEP Remote System enthält einen SCPI 488 konformen Parser, der die entsprechenden Befehle umsetzt und an die Gerätefirmware weitergibt.

Das TSEP Remote System unterscheidet hierbei in Kanäle (Channels) und Parsern. Der Kommunikationskanal wird als Kanal oder Channel bezeichnet und implementiert das hardwaretechnische Interface für den Parser. Kunden können jederzeit eigenständige Kanäle definieren und implementieren. Standardmäßig werden die Kanäle TCP/IP, RS232 und für LXI Member der HiSlip Kanal mitgeliefert. TCP/IP und HiSlip unterstützen sowohl IPV4 als auch IPV6.

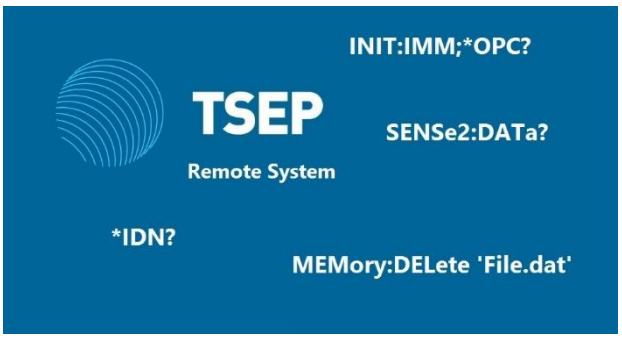
Der Parser definiert die einzelnen SCPI Kommandos, die Syntax und deren Abarbeitung. Das TSEP Remote System kann mehrere Parser gleichzeitig unterstützen. Dies erlaubt es dem Kunden modular seine SCPI Kommandos aufzubauen und bei anderen Gerätevarianten wiederzuverwenden. Standardmäßig enthält das TSEP Remote System mehrere Parser die einen Teil der 488-2 konformen SCPI Befehle enthalten.

Das TSEP Remote System ist sowohl unter Windows, als auch unter Linux verfügbar. Der Source-Code ist für beide Plattformen identisch (Common Source), was Wartung und Integration in die Gerätesoftware erleichtert.

Das TSEP Remote System ist besonders auf eine schnelle Kommunikation ausgelegt. Auf Intel I5 Rechner sind Antwortzeiten (Beispielsweise *IDN?) von unter 400µs mit TCP/IP und Localhost Loop back 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 unter 100µs bereits gemessen. TSEP arbeitet an einer kontinuierlichen Verbesserung der Performance des TSEP Remote Systems, da der Faktor Performance ein durchaus wichtiges Keyfeature ist.

Für die Dokumentation der SCPI Kommandos hat TSEP ein eigenes Tool erstellt, das aus den XML Definitionen der Kommandos, freidefinierte Word-Dokumente erstellt. Der Kunde erhält dieses Tool im Source-Code und kann es für seine CI und Zwecke anpassen. Das Tool wurde in C# erstellt, so das nur eine geringe Einarbeitung Zeit und ein niedriger Skill-Level notwendig sind.

TSEP Remote System / Basics:



Das TSEP Remote System wurde komplett in C++ erstellt und basiert auf dem C++11 Standard. Um den unterschiedlichen Entwicklungs-umgebungen Rechnung tragen zu können, wurde mit CMake für die Generierung der Projektdaten (Visual Studio Solution/Projekt Dateien, Make Dateien oder Eclipse Projekte) verwendet. Somit lassen sich einfach neue Entwicklungsumgebungen einbringen, bzw. Änderungen an bestehenden Projekten durchführen.

Das TSEP Remote System basiert auf einen objektorientierten Ansatz, der konsequent in der Implementierung umgesetzt worden ist. Alle notwenigen Interfaces werden über Vererbung an die entsprechende Gerätefirmware weitergegeben. Da SCPI Kommandos und Queries immer bezugnehmen auf Gerätedaten, wurde eine Vorgehensweise für den Zugriff auf diese Daten definiert und implementiert. Somit lassen sich einfach gerätespezifische Daten an die entsprechenden Kommandos weiterleiten.

Das TSEP Remote System kann nicht nur synchrone Kommandos verarbeiten, sondern auch asynchronen Kommandos werden unterstützt. Hierbei laufen diese Kommandos in einem eigenen Thread und sind somit eigenständig und unabhängig lauffähig. Die Implementierung dieser asynchronen Kommandos unterscheidet sich nicht zu den synchronen Kommandos.

Das TSEP Remote System unterstützt zusätzlich noch folgende Features:

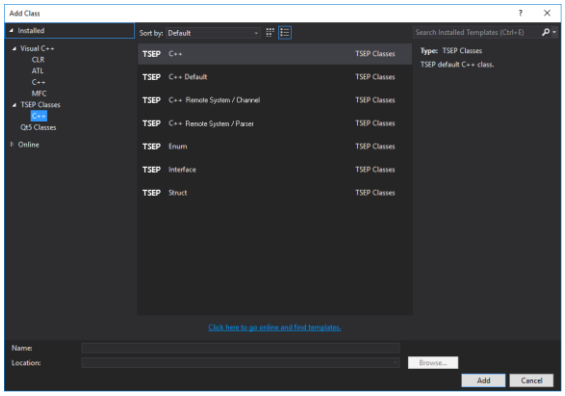
- Kurz- und Langform eines Kommandos
- Zusammengesetze Kommandos
- Suffixe in Kommandos
- Alias Kommandos für Kompatibilitätsmodi
- Überprüfung der Syntax von Parameter direkt im Parser
- Benutzerspezifische Kommandoparameter
- Integriertes Fehlerhandling
- SCPI Statusregister
- Overlapped Commands
- Kompatibilitäts-Modi, zur Emulation andere Messgerätehersteller
- Speed-Up Modus, verbesserte Kommandoverarbeitung, bei reduziertem Syntaxcheck

Für die Entwicklung eigener Kommandos oder Channels steht ein eigener SDK (Software Development Kit) zur Verfügung der die notwendigen Tools und Software bereitstellt.

TSEP Remote System / Channels:

Das TSEP Remote System definiert für die Kommunikation zwischen Parser und der eigentlichen Hardware die Channels. Diese Channels implementieren den Zugriff auf die Daten von und zu der Kommunikationshardware. Zurzeit unterstützt das TSEP Remote System die Standard Schnittstellen TCP/IP, HiSlip (nur für LXI Mitglieder) und RS232.

Es besteht aber jederzeit die Möglichkeit für einen Kunden einen eigenen Channel zu erstellen. Für die Erstellung eines eigenen Channels sind für Visual Studio ein eigener Class Wizard vorhanden.



Für die Linux Entwicklungssysteme steht eine detaillierte Beschreibung zur Verfügung.

Im TSEP Remotesystem können sowohl Single-Instanz-Channels als Multi-Instanz-Channels definiert und implementiert werden.

Single-Instanz-Channels sind typischer Weise Kommunikationskanäle die nur eine Verbindung pro Hard-ware Ressource zulassen. Als Beispiel hierfür kann die RS232 Schnittstelle erwähnt werden. Natürlich können mehrere RS232 Hardwarebausteine unterstützt werden, hierzu müssen die einzelnen Single-Instanz-Kanäle nur mehrfach (also für jede Hard-ware) im TSEP Remote System an-gemeldet werden.

Multi-Instanz-Channels können mehrere aktive Verbindungen gleichzeitig bearbeiten. Als Beispiel für derartige Channels können der TCP/IP und der HiSlip Channels aufgeführt werden. Auch diese Channels können mehrfach im TSEP Remote System angemeldet werden und zum Beispielsweise über verschiedene TCP/IP Ports zu kommunizieren.

Alle angemeldeten Channels übertragen ihre Daten in das TSEP Remote System und werden dann vom entsprechenden Parser verarbeitet. Die Ergebnisse werden dann an den zuständigen Channel weitergeleitet.

Channels können zur Laufzeit hinzugeladen werden, oder aus dem TSEP Remote System entfernt werden verwendet.

TSEP Remote System / Parser:

Die eigentliche Verarbeitung der SCPI Kommandos erfolgt im entsprechenden Parser. Innerhalb des TSEP Remote System können beliebig viele Parser definiert werden. Dadurch können Funktionalitäten separiert werden und in anderen Geräten wiederverwendet werden. So liefert TSEP mehrere Parser mit die Teile des SCPI 488-2 Kommandosatzes beinhalten. Die Kommandosyntax orientier sich an SCPI 488-2, enthält aber auch spezifische Erweiterungen die separate zu- oder abgeschaltet werden können.

Das Parsen der Kommandos ist hoch effektiv und effizient gestaltet, so dass eine optimale Verarbeitungs-zeit erreicht wird. Zusätzlich hat TSEP einige Optimierungsverfahren in die Abarbeitung eingearbeitet. So werden Kommandos nach ihrer Anzahl von Aufrufen sortiert, damit wird gewährleistet das häufige Kommandos auch schneller bearbeitet werden können. Zusätzlich wurden die SCPI Kommandoverarbeitung auf Multi-Core und Multi-Thread CPUs ausgelegt, somit sind parallele Verarbeitungen machbar, was wiederum zur Optimierung der Verarbeitungszeit beiträgt.

Das TSEP Remotesystem unterstützt auch die Verarbeitung von Overlapped Kommandos, das sind Kommandos auf deren Beendigung nicht gewartet wird, diese Kommandos können quasi parallel ablaufen. Die Implementierung dieser Kommandos unterscheidet sich nur geringfügig von den nicht „overlapped" Kommandos, es kann lediglich auf den Abbruch während des Kommandos programmatisch reagiert werden. Die Parser erlauben auch die Verwendung von Alias Kommandos.
Alias Kommandos sind Kommandos mit einer anderen Syntax, jedoch mit demselben Funktionalitäten und Parametern. In der Regel sind derartige Konstrukte notwendig um ältere Kommandos (Komptabilitätsmodus) und somit bestehende Messsoftware zu unterstützen.

Analog zu den Kanälen kann der Benutzer selbstständig eigene Parser erstellen. Unterstützt wird der Benutzer durch den XML Generator, der für sämtliche definierten SCPI Kommandos die notwendigen Source-Code Skeletons erzeugt.

TSEP Remote System / XML Definition der Kommandos:

```
<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="CLSCommand"
         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>
```

Die SCPI Kommandos werden innerhalb des TSEP Remote System über eine XML basierte Definitionen erstellt und verwaltet. Nicht nur die notwenigen Source-Code Teile werden von diesem Generator erstellt, es wird auch die notwendige Dokumentation mit generiert.

Innerhalb der XML Dateien, werden nicht nur die Kommandosyntax definiert, es werden auch Attribute und Beschreibungen hinterlegt. Somit sind alle kommandospezifischen Definitionen zentral hinterlegt.

Mit Hilfe des Generators werden die notwendigen Source Code Skeletons für die Einbettung in die Firmware generiert. Der Kunden muss sich nur noch um die Implementierung der eigentlichen Funktionalität kümmern. Alle SCPI Kommandos sind als C++ Klassen realisiert. Alle notwendigen Header und das spezifische Parser Framework werden automatisch erzeugt. Die im XML definierten Beschreibungen werden ebenfalls mit in die Source Code Skeletons übernommen.

```
/**
* @class CCLSCommand
*
* @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 CCLSCommand : public IScpiCommand
{

public:

  /**
  * @brief Constructor
  *
  * The constructor adds all defined command parameters to the command
  */
  CCLSCommand(void)
    : IScpiCommand(eScpiCommandType::Command, "*CLS", "", "", "") {}

  /**
  * @brief Destructor
  */
  virtual ~CCLSCommand(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 transfered 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.
  */
  TSEPBoolean ExecuteCommand(CScpiCommandData& cScpiCmdData);
```

Zusätzlich kann mit Hilfe des Generators auch die Clientseite der SCPI Kommunikation erstellt werden. Hierzu wird Analog zu der Serverseite (Messgerät) auch die Source Code Skeletons mit erzeugt.

Zur Erstellung der Dokumentation für die in den Parsern enthaltenen Kommandos kann auch der Generator verwendet werden. Der Generator erzeugt auf Basis der XML Definitionsdatei und eines Word Templates der entsprechenden Dokumentation. TSEP liefert eine Standardvorlage mit Hilfe derer ein Dokument erstellt werden kann. Der Kunde kann diese Vorlage aber jederzeit ändern und seinen Wünschen und seiner CI anpassen.

TSEP Remote System / SCPI Parameter:

Das TSEP Remote System hat für die Verarbeitung der SCPI Kommandoparameter einen komplett objektorientierten Ansatz gewählt. Parameter von SCPI Kommandos oder Abfragen werden innerhalb des TSEP Remote Systems immer über objektorientierte Klassen abgebildet. Dieser Ansatz erleichtert die Verarbeitung und die Erweiterung von Parameter dramatisch.

Das TSEP Remote System liefert die notwendigen Basisparameter wie Integer, Gleitkommazahl oder String natürlich als Basisimplementierung mit. Zusätzlich ist noch eine ganze Reihe von zusätzlichen Standardparametern definiert. So sind zum Beispiel auf Arrays von Parametern möglich, was ins besonders bei einigen Konfigurationskommandos (Antennenparameter, Kalibrierdaten etc.) die Arbeit erleichtert. Auch die Verarbeitung von Enumerationsparameter ist natürlich im TSEP Remote System möglich. Hierzu existiert eine Reihe von Beispielen die der Kunde jederzeit für seine Anforderungen anpassen und modifizieren kann.

Da die mit dem TSEP Remote System mitgelieferten Parameter nicht ausreichen um alle Kundenwünsche abzudecken, können die Parameter auch jederzeit vom Kunden erweitert werden und zur Laufzeit geladen werden.

Der Zugriff auf die Daten Der Kommandoparameter erfolgt über Get/Set Methoden, die mit der entsprechenden Signatur ausgestattet sind.

Innerhalb von Kommandos kann mit Hilfe eines Parameter-Casts einfach auf die entsprechenden Parameter zugegriffen werden. Der Generator kann bei der Erzeugung der Source-Code-Skeleton alle verfügbaren Parameter mit in das Kommando integrieren, was den Entwickler bei der Implementierung hilft.

Preise:

Alle nachfolgenden Preise verstehen sich excl. MwSt. Mit jeder Lizenz können beliebig viele Geräte mit dem TSEP Remote System versehen werden. Die Lizenz bezieht sich nicht auf ein Betriebssystem.

Die Binärlizenz enthält alle notwendigen Sourcen um eigene Parser und Kanäle zu erstellen. Das eigentliche TSEP Remote System liegt Standardmäßig in einer Binärversion für die Betriebssysteme Windows 7 und 10 und Linux Ubuntu 14.04/16.04 vor. Alle anderen Betriebssystem auf Anfrage.

Die Source Code Lizenz enthält alle Sourcen des Remotesystems.

Binärlizenz:

▪ **TSEP Remote System**		**9.999,-- €**
▪ **TSEP Remote System Support (Telefon + Mail)**	**pro Jahr**	**2.400,-- €**

Source Code Lizenz:

▪ **TSEP Remote System + 1 Jahr Support + Updates**		**24.999,-- €**
▪ **TSEP Remote System + 3 Jahr Support + Updates**		**35.499,-- €**
▪ **TSEP Remote System Support (Telefon + Mail)**	**pro Jahr**	**2.400,-- €**
▪ **TSEP Remote System Support + Updates**	**pro Jahr**	**4.800,-- €**