

PEM-ProLib++

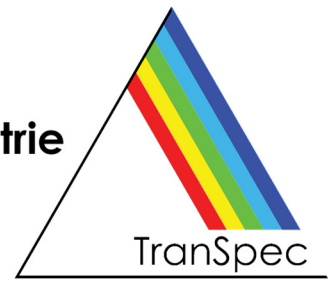
Programmierbibliothek für TranSpec[®] - Spektrometer

Zur Erstellung eigener Applikationen für die allgemeine Spektroskopie und speziell zur Messung von Plasma-Emissionen mit Hilfe unserer TranSpec Spektrometer bieten wir unsere leistungsfähige Programmierbibliothek **PEM-ProLib++** an.

In PEM-ProLib++ ist die gesamte Spektrenerfassung, d.h., das Scannen der Diodenzelle, eventuelle Mittelung der Rohdaten, Dunkelstrom-Korrektur und die anschließende Spektrennormierung vollständig in einfachen Funktionsaufrufen gekapselt, so dass Sie sich ganz auf die Erstellung Ihrer Applikation konzentrieren können. Natürlich haben Sie Zugriff auf alle gemessenen Spektren, inklusive der Rohdaten.

- Laufzeit-Lizenzierte Dynamische Link Library (DLL) mit Standard-C Aufrufen
Kompatibel zu gängigen C/C++ Compilern, Visual Basic und VBA (Excel), LabVIEW
- Ausführliche Überprüfung von Parametern und aktuellem Mess-Status
Sie können mit PEM-ProLib++ praktisch nichts falsch machen!
- Unterstützung von optionalem, externen I/O-Modul mit 8x TTL und 4x Analogausgabe
- Detailliertes Benutzerhandbuch in Englisch, gedrucktes Handbuch und als PDF
- Beispielprogramm als Windows Konsolenapplikation, mit C/C++ Quellcode
- Auf der nächsten Seite finden Sie ein kleines Programmierbeispiel

Technische Spezifikationen auf der nächsten Seite ►



PEM-ProLib++ Programmierbibliothek - Technische Daten

Januar 2022, bezogen auf Version 2.0, alle Angaben ohne Gewähr, technische Änderungen vorbehalten

Hardware- und Softwarevoraussetzung

- Standard PC/Laptop mit Windows 10
- C/C++ Entwicklungssystem (MS Visual Studio empfohlen), Delphi, Visual Basic oder VBA (Excel), LabVIEW
- TranSpec Spektrometer
- PEM-ProVis Professional wird empfohlen, ist aber nicht notwendig

Programmierbeispiel

Nachfolgend ein kleines Programmbeispiel, um die Leistungsfähigkeit und einfache Handhabung von PEM-ProLib++ zu demonstrieren. Als exemplarische Aufgabe programmieren wir die vollautomatische Messung eines 10fach gemittelten Emissions-Spektrums mit Analogausgabe eines Emissions-Trendwertes:

```
// Schritt 1: Spektrometer öffnen und initialisieren
PEMPRO_SPECHARDWARE sSpecHardwareInfo;
PEMPro_OpenSpectrometer( PEMPRO_TRANSPEC_19Z, &sSpecHardwareInfo );

// Schritt 2: Messparameter festlegen
PEMPRO_MEASPARAM sMeasPara;
sMeasPara.dIntegrationTime = 20.0;           // 20 ms Integrationszeit
sMeasPara.bEnableAverage = 1;             // Spektrenmittelung an
sMeasPara.lNumberAverage = 10;          // 10fach messen und mitteln
PEMPro_SetMeasPara( &sMeasPara );        // Messparameter an Spektrometer senden

// Schritt 3: Externes Digital/Analog-Modul öffnen und initialisieren
PEMPro_USB3110_OpenDevice( PEMPRO_DIGITAL-8OUT0IN, PEMPRO_ANALOG_UNIPOLAR );

// Schritt 4: Messung eines gemittelten Emissionsspektrums
PEMPRO_SPECSTATUS sSpecStatus;
PEMPro_RunMeasSpectrum ();                // Messung starten

PEMPro_GetSpecStatus( &sSpecStatus );     // Warten bis Messung beendet ist
while ( sSpecStatus.bRunSpectrum )
    PEMPro_GetSpecStatus( &sSpecStatus );

// Schritt 5: Die Messung ist beendet, wir rufen das Spektrum ab
PEMPRO_SPECDATA sSpecData;
PEMPro_GetSpectrumData( PEMPRO_SPECTRUM_EMISSION, &sSpecData );

// Schritt 6: Berechnung und Ausgabe des Emissions-Trendwertes bei z.B. 254 nm als analoger Spannungswert:
double dTraceValue;
PEMPro_GetTraceValue( 254.0 , &dTraceValue );

// Trendwert auf +10 Volt normieren und am Kanal 0 ausgeben
....
PEMPro_USB3110_SetAnalogOut( 0 , dTraceValueAsVolt );
```

Hinweis TranSpec ist ein in Deutschland eingetragenes Warenzeichen des Ing.-Büros für Angewandte Spektrometrie, Dipl.-Ing. (FH) Thomas Fuchs. Alle sonstigen Produktnamen sind möglicherweise Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Hersteller.