

PEM-ProLib++

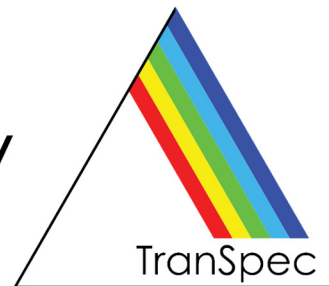
Programming Library for Plasma Emission Measurement

For developing your own plasma emission measurement applications using our TranSpec spectrometers, we provide our powerful and very easy-to-use programming library **PEM-ProLib++**.

With PEM-ProLib++ the entire spectra data acquisition, like scanning the diode array, raw data averaging, possible dark current correction and the emission spectra normalization is fully encapsulated in just a few simple function calls. PEM-ProLib+ gives you full access to all measured spectra (including the raw data) and easily allow to report emission trace values to an external analog out module.

- Runtime licensed Dynamic Link Library (DLL) providing standard C calls
Compatible with common C/C++ compiler, Visual Basic and VBA (Excel), LabVIEW
- Extensive parameter checking and measurement status verification during runtime
You hardly can do anything wrong when working with PEM-ProLib++
- Supports special external I/O module with 8-channel TTL and 4-channel analog out
- Detailed user manual as PDF file and optionally as printed document
- Demo software as Windows console application, including C/C++ source code
- See next page for a short programming example!

Technical specifications on next page ►



PEM-ProLib++ Programming Library • Technical Specifications

May 2025, related to version 2.0, without guarantee, subject to changes.

Minimum Hardware and Software Requirements

- Standard PC/Laptop with Windows 10 or Windows 11
- C/C++ development system (MS Visual Studio recommended), Delphi, Visual Basic or VBA (Excel), LabVIEW
- TranSpec Process Spectrometer
- PEM-ProVis Professional is recommended, but not required

Programming Example

The following short programming example may demonstrate how easy to use the PEM-ProLib++ is. As an example, we code the fully automated measurement of a 10 times averaged emission spectrum with subsequent analog output of an emission trace value:

```
// Step 1: open and initialize spectrometer
PEMPRO_SPECHARDWARE sSpecHardwareInfo;
PEMPPro_OpenSpectrometer( PEMPRO_TRANSPEC_19Z, &sSpecHardwareInfo );

// Step 2: setup measurement parameter:
PEMPRO_MEASPARA sMeasPara;
sMeasPara.dIntegrationTime = 20.0;           // 20 ms integration time
sMeasPara.bEnableAverage = 1;               // averaging on
sMeasPara.iNumberAverage = 10;             // 10 scans for averaging
PEMPPro_SetMeasPara( &sMeasPara );         // notify settings to spectrometer

// Step 3: open and initialize external digital/analog module
PEMPPro_USB3110_OpenDevice( PEMPRO_DIGITAL-8OUT0IN, PEMPRO_ANALOG_UNIPOLAR );

// Step 4: perform measurement of an averaged emission spectrum
PEMPPro_RunMeasSpectrum();                 // start measurement

PEMPRO_SPECSTATUS sSpecStatus;
PEMPPro_GetSpecStatus( &sSpecStatus );     // wait until measurement is done
while ( sSpecStatus.bRunSpectrum )
    PEMPro_GetSpecStatus( &sSpecStatus );

// Step 5: retrieve measured emission spectrum
PEMPRO_SPECDATA sSpecData;
PEMPPro_GetSpectrumData( PEMPRO_SPECTRUM_EMISSION, &sSpecData );

// Step 6: compute and report emission trace value at 254 nm, for instance:
double dTraceValue;
PEMPPro_GetTraceValue( 254.0 , &dTraceValue );

// normalize trace value to +10 V and report at analog channel 0
....
PEMPPro_USB3110_SetAnalogOut( 0 , dTraceValueAsVolt );
```

Note **TranSpec** is a registered German trademark of Dipl.-Ing. (FH) Th. Fuchs, Engineer's Office for Applied Spectroscopy. All other mentioned product names are or possibly might be trademarks or registered trademarks of their owners.