



---

## TranSpec++

### Programming Library for TranSpec<sup>®</sup> - Spectrometer

---

For developing your own spectroscopy and photometry Windows applications using our TranSpec spectrometers, we provide our powerful and easy-to-use programming library **TranSpec++**.

With TranSpec++ the entire spectra data acquisition, like scanning the diode array, raw data averaging, dark current correction and the spectra normalization is fully encapsulated in just a few simple function calls. At the same time TranSpec++ gives you full access to all measured spectra, including the raw data.

- Runtime licensed Dynamic Link Library (DLL) providing standard C calls  
Compatible with common C/C++ compilers, Visual Basic and VBA (Excel), LabView, etc.
- Extensive parameter checks and measurement status verification  
You hardly can do anything wrong when working with TranSpec++
- Detailed user's manual as compiled HTML file and printed PDF document
- Demo software as Windows console application, including C/C++ source code
- See next page for a programming example!

Technical specifications on next page ►



---

## TranSpec++ Programming Library • Technical Specifications

August 2008, related to version 5.0, without guarantee, subject to changes.

### Minimum Hardware and Software Requirements

- PC/Laptop with at least Pentium-4
- Windows XP or Windows Vista
- C/C++ development system (MS Visual Studio recommended), Delphi, Visual Basic or VBA, LabView
- TranSpec Spectrometer

### Programming Example

The following shows a small programming example, in order to demonstrate the capabilities and easy usage of the TranSpec++ library. As a simple exercise, we program the fully automatic measurement of a dark current corrected emission spectrum:

```
// Step 1: open and initialize spectrometer
TRANSPEC_SPECHARDWARE sSpecHardwareInfo;
TranSpec_OpenSpectrometer( TRANSPEC_LITE, &sSpecHardwareInfo );

// Step 2: setup measurement parameter:
TRANSPEC_MEASPARAM sMeasPara;
sMeasPara.dIntegrationTime = 20.0;           // 20 ms integration time
sMeasPara.bEnableAverage = 1;             // averaging on
sMeasPara.lNumberAverage = 10;          // 10 times averaging
TranSpec_SetMeasPara( &sMeasPara );       // notify settings to spectrometer

// Step 3: perform measurement of an averaged Dark Current
TranSpec_RunMeasDarkCurrent();           // start measurement

TRANSPEC_SPECSTATUS sSpecStatus;
TranSpec_GetSpecStatus( &sSpecStatus );   // wait until measurement is done
while ( sSpecStatus.bRunDarkCurrent )
    TranSpec_GetSpecStatus( &sSpecStatus ); // get next TranSpec status information

// Step 4: perform measurement of an averaged and Dark Current corrected Emission spectrum
TranSpec_RunMeasSpectrum ();             // start measurement
TranSpec_GetSpecStatus( &sSpecStatus );   // wait until measurement is done
while ( sSpecStatus.bRunSpectrum )
    TranSpec_GetSpecStatus( &sSpecStatus );

// Step 5: the measurement is done, now retrieve the spectrum data
TRANSPEC_SPECDATA sSpecData;
TranSpec_GetSpectrum( EMISSION, &sSpecData );

// Done! Aside from other special measurement and parameter information
// the structure <sSpecData> contains the normalized Emission spectrum
```

---

**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.