Programmer's Guide

JETI Software Development Kit
jeti_spectro.dll

Version 3.x

JETI Technische Instrumente GmbH
Tatzendpromenade 2
D-07745 Jena
Tel.: +49-3641-225680
Fax: +49-3641-225681
e-mail: sales@jeti.com
internet: www.jeti.com

Table of contents

1 JETI SDK Overview 3

2 Introduction 4

2.1 How to communicate 4

3 Function Reference 5

3.1 JETI_GetSpectroDLLVersion 6
3.2 JETI_GetNumSpectro 7
3.3 JETI_GetSerialSpectro 8
3.4 JETI_OpenSpectro 9
3.5 JETI_CloseSpectro 10
3.6 JETI_DarkSpec 11
3.7 JETI_LightSpec 12
3.8 JETI_ReferSpec 13
3.9 JETI_TransReflSpec 14
3.10 JETI_SpectroTint 15

4 Examples 16

Appendix A 17

5 Service 18

1 JETI SDK Overview

The JETI Software Development Kit provides a complete software solution for interfacing spectrometric and radiometric devices from JETI Technische Instrumente GmbH. No firmware command expertise is required. Instead, a simple, high-level Application Program Interface (API) is used to provide complete connectivity. The API is provided in the form of several Windows Dynamic Link Libraries (DLL). The libraries can be used by any programming language that can handle DLL's such C/C++, VisualBasic, or LabVIEW.
To get access to the functions the needed DLL files have to be copied to the Windows System Folder or to the working directory of the calling application.

The following DLLs are available:

- jeti_spectro.dll / jeti_spectro64.dll
  - provides a set of functions for simple spectrometric measurement

- jeti_spectro_ex.dll / jeti_spectro_ex64.dll
  - a set of functions like jeti_spectro.dll, but with more options to control the measurement

- jeti_radio.dll / jeti_radio64.dll
  - provides a set of functions for simple radiometric measurement, including calculation of colorimetric values (e.g. xy- and u'v'-values, CCT, CRI,…)

- jeti_radio_ex.dll / jeti_radio_ex64.dll
  - a set of functions like jeti_radio.dll, but with more options to control the measurement and calculations

- jeti_core.dll / jeti_core64.dll
  - a set of functions to fully control the device and perform custom measurement sequences

Please note that this documentation describes only the functions provided by the* jeti_spectro.dll*.
For description of the other DLL's please refer to the corresponding documents.

2 Introduction

The jeti_spectro API is provided in the form of a Windows Dynamic Link Library (DLL). The interface DLL communicates with the device via the provided device driver and the basic driver DLL jeti_core.dll.
JETI Technische Instrumente GmbH offers two versions of the DLL. The first version is for 32bit Windows operating systems (WinXP / Windows Vista / Windows 7).
The second version is for real 64 bit programs under the 64 bit versions of Windows Vista and Windows 7.
There are no differences in the functionality between the two versions.

2.1 How to communicate

To get access to the functions you have to copy the files jeti_spectro.dll and jeti_core.dll to the working directory of your application, or to the windows\system32 directory.
In general, the user initiates communication with the target device(s) by making a call to *JETI_GetNumSpectro*. This call will return the number of target devices. This number is then used as a range when calling *JETI_GetSerialSpectro* to build a list of device serial numbers.

To access a device, it must first be opened by a call to *JETI_OpenSpectro* using an index determined from the call to *JETI_GetNumSpectro*. The *JETI_OpenSpectro* function will return a handle to the device that is used in all subsequent accesses. When I/O operations are complete, the device is closed by a call to *JETI_CloseSpectro*.

In case of a fatal communication error (error code 0xFF) JETI_HardReset (from jeti_core.dll) could be used to reset the device and resume the communication. For more information see the function description of JETI_HardReset in 'JETI SDK Programmer's Guide jeti_core.dll' and the Appendix A.

3 Function Reference

Convention for calling : __stdcall

Type	Size in Bit	Minimum	Maximum
DWORD (unsigned long)	32	0	2^{32}-1
FLOAT (IEEE standard)	32	-3.40282E+38	3.40282E+38
BYTE	8	0	255

3.1 JETI_GetSpectroDLLVersion

This function returns the current version number of the jeti_spectro DLL.

3.1.1 Prototype

DWORD JETI_GetSpectroDLLVersion (WORD *wMajorVersion, WORD *wMinorVersion, WORD *wBuildNumber)

3.1.2 Parameters

Input

Name	Type	Description	Call
wMajorVersion	WORD *	address of a WORD variable that will contain the major version	By reference
wMinorVersion	WORD *	address of a WORD variable that will contain the minor version	By reference
wBuildNumber	WORD *	address of a WORD variable that will contain the build number	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.2 JETI_GetNumSpectro

This function returns the number of JETI devices connected to the PC and to the LAN.

3.2.1 Prototype

DWORD JETI_GetNumSpectro (DWORD *dwNumDevices)

3.2.2 Parameters

Input

Name	Type	Description	Call
dwNumDevices	DWORD *	address of a DWORD variable that will contain the number of devices connected	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.3 JETI_GetSerialSpectro

This function returns the serial numbers for the device specified by an index passed in dwDeviceNum. The index for the first device is 0 and the last device is the value returned by *JETI_GetNumSpectro* – 1.

3.3.1 Prototype

DWORD JETI_GetSerialSpectro (DWORD dwDeviceNum, DWORD *dwSerial1, DWORD *dwSerial2)

3.3.2 Parameters

Input

Name	Type	Description	Call
dwDeviceNum	DWORD	index of the device for which the serial numbers are desired	By value
dwSerial1	DWORD *	address of a DWORD variable that will contain the first serial number	By reference
dwSerial2	DWORD *	address of a DWORD variable that will contain the second serial number	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.4 JETI_OpenSpectro

Opens a device (using device number returned by *JETI_GetNumSpectro*) and returns a handle which will be used for subsequent accesses.

3.4.1 Prototype

DWORD JETI_OpenSpectro (DWORD dwDeviceNum, DWORD *dwDevice)

3.4.2 Parameters

Input

Name	Type	Description	Call
dwDeviceNum	DWORD	Device index. 0 for first device, 1 for second, etc.	By value
dwDevice	DWORD *	Pointer to a variable where the handle to the device will be stored	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.5 JETI_CloseSpectro

Closes an open device using the handle provided by *JETI_OpenSpectro*.

3.5.1 Prototype

DWORD JETI_CloseSpectro (DWORD dwDevice)

3.5.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device to close as returned by *JETI_OpenSpectro*	By value

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.6 JETI_DarkSpec

Performs a dark measurement and returns the counts per wavelength in the range of 380 to 780 nm with a step-width of 5 nm (81 values).
During a dark measurement the shutter (if present) is closed, and the internal lamp (if present) is off.

NOTE: The function will return ***after*** the measurement has finished. Depending on the used integration time this could take several seconds (up to 65 seconds).
E.g. if dwTint = 1500 ms, then the function will return after approx. 1.5 … 2 seconds.

3.6.1 Prototype

DWORD JETI_DarkSpec (DWORD dwDevice, DWORD dwTint, FLOAT *fDark)

3.6.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device as returned by *JETI_OpenSpectro*	By value
dwTint	DWORD	Integration time in [ms] for the measurement	By value
fDark	FLOAT *	Pointer to an array where the dark values will be stored **(the array must provide space for at least 81 values)**	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.7 JETI_LightSpec

Performs a light measurement and returns the counts per wavelength in the range of 380 to 780 nm with a step-width of 5 nm (81 values).
During a light measurement the shutter (if present) is open, and the internal lamp (if present) is on.

NOTE: The function will return ***after*** the measurement has finished. Depending on the used integration time this could take several seconds (up to 65 seconds).
E.g. if dwTint = 1500 ms, then the function will return after approx. 1.5 … 2 seconds.

3.7.1 Prototype

DWORD JETI_LightSpec (DWORD dwDevice, DWORD dwTint, FLOAT *fLight)

3.7.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device as returned by *JETI_OpenSpectro*	By value
dwTint	DWORD	Integration time in [ms] for the measurement	By value
fLight	FLOAT *	Pointer to an array where the light values will be stored **(the array must provide space for at least 81 values)**	By reference

Return Value

Type	Description	
DWORD	0x00	JETI_SUCCESS
	0x…	see Appendix A for error codes

3.8 JETI_ReferSpec

Performs a reference measurement and returns the counts per wavelength in the range of 380 to 780 nm with a step-width of 5 nm.
The reference spectra is the light spectra with dark current compensation.
If no dark measurement was performed before using the same integration time, then a dark measurement is also performed.

NOTE: The function will return ***after*** the measurement has finished. Depending on the used integration time this could take several seconds (up to 130 seconds).
E.g. if dwTint = 1500 ms, then the function will return after approx. 1.5 … 2 seconds.
If no dark measurement was performed before using an integration time of 1500 ms, then the function will return after approx. 3…4 seconds.

3.8.1 Prototype

DWORD JETI_ReferSpec (DWORD dwDevice, DWORD dwTint, FLOAT *fRefer)

3.8.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device as returned by *JETI_OpenSpectro*	By value
dwTint	DWORD	Integration time in [ms] for the measurement	By value
fRefer	FLOAT *	Pointer to an array where the reference values will be stored **(the array must provide space for at least 81 values)**	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

3.9 JETI_TransReflSpec

Performs a transmission/reflection measurement and returns the ppt (parts per thousand) per wavelength in the range of 380 to 780 nm with a step-width of 5 nm.
To obtain a valid measuring result a reference measurement has to be performed before, using the same integration time.

NOTE: The function will return ***after*** the measurement has finished. Depending on the used integration time this could take several seconds (up to 65 seconds).
E.g. if dwTint = 1500 ms, then the function will return after approx. 1.5 … 2 seconds.

3.9.1 Prototype

DWORD JETI_TransReflSpec (DWORD dwDevice, DWORD dwTint, FLOAT *fTransRefl)

3.9.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device as returned by *JETI_OpenSpectro*	By value
dwTint	DWORD	Integration time in [ms] for the measurement	By value
fTransRefl	FLOAT *	Pointer to an array where the transmission/reflection values will be stored **(the array must provide space for at least 81 values)**	By reference

Return Value

Type	Description	
DWORD	0x00 0x..	JETI_SUCCESS see Appendix A for error codes

3.10 JETI_SpectroTint

Obtains the last used integration time in [ms].

3.10.1 Prototype

DWORD JETI_SpectroTint (DWORD dwDevice, DWORD *dwTint)

3.10.2 Parameters

Input

Name	Type	Description	Call
dwDevice	DWORD	Handle to a device as returned by *JETI_OpenSpectro*	By value
dwTint	DWORD *	Pointer to a variable where the integration time will be stored	By reference

Return Value

Type	Description	
DWORD	0x00 0x…	JETI_SUCCESS see Appendix A for error codes

4 Examples

To help starting development the SDK includes several examples.

Appendix A

Error codes and their description:

Error code	*#define*	*Description*
0x00	JETI_SUCCESS	no error occured
0x02	JETI_ERROR_OPEN_PORT	could not open COM-port
0x03	JETI_ERROR_PORT_SETTING	could not set COM-port settings
0x04	JETI_ERROR_BUFFER_SIZE	could not set buffer size of COM-port
0x05	JETI_ERROR_PURGE	could not purge buffers of COM-port
0x06	JETI_ERROR_TIMEOUT_SETTING	could not set COM-port timeout
0x07	JETI_ERROR_SEND	could not send to device
0x08	JETI_TIMEOUT	communication timeout error
0x0A	JETI_ERROR_RECEIVE	could not receive from device
0x0B	JETI_ERROR_NAK	command not supported or invalid argument
0x0C	JETI_ERROR_CONVERT	could not convert received data
0x0D	JETI_ERROR_PARAMETER	invalid argument
0x0E	JETI_BUSY	device busy
0x11	JETI_CHECKSUM_ERROR	invalid checksum of received data
0x12	JETI_INVALID_STEPWIDTH	invalid step width
0x13	JETI_INVALID_NUMBER	invalid device number
0x14	JETI_NOT_CONNECTED	device not connected
0x15	JETI_INVALID_HANDLE	invalid device handle
0x16	JETI_INVALID_CALIB	invalid calibration file number
0x17	JETI_CALIB_NOT_READ	calibration data not read
0x20	JETI_MEASURE_FAIL	measurement failed (overexposure)
0x80	JETI_DLL_ERROR	internal DLL error
0xFF	JETI_FATAL_ERROR	fatal communication error
0x2710...0x2AFC		Windows sockets error codes

If a fatal communication error occurs (error code 0xFF) there are several ways to solve the problem.

1) Call JETI_HardReset (from jeti_core) to perform a device hardware reset. The effect of this function is the same as disconnecting then reconnecting the device from USB. This will work only if the device uses an FTDI USB-to-serial converter and was opened with direct access to the FTDI driver (opened with JETI_GetNumSpectro and JETI_OpenSpectro) instead of using the VCP (virtual com port) driver (JETI_OpenCOMDevice and/or JETI_SetComSearch).
Please note that all custom settings (e.g. integration time, function,...) will be set to default values and have to be repeated.

2) Closing the device with JETI_CloseSpectro will also perform a hardware reset if a fatal communication error occurred on a device with FTDI USB-to-Serial converter. After closing the device it should be possible to reopen the device with JETI_GetNumSpectro and JETI_OpenSpectro.

3) If a JETI device with FTDI USB-to-Serial converter was opened using VCP driver (e.g. by using JETI_OpenCOMDevice) or by using other connections (like RS232, bluetooth,...) a fatal communication error can only be resolved by closing the device with JETI_CloseSpectro and manually reset the device.

5 Service

In case of any questions or technical problems please contact:

JETI Technische Instrumente GmbH
Tatzendpromenade 2
D-07745 Jena
Tel. +49 3641 225 680
Fax +49 3641 225 681
e-mail: sales@jeti.com
Internet: www.jeti.com

Copyright (c) 2012 JETI Technische Instrumente GmbH. All rights reserved.

Software and operating instruction are delivered with respect to the License agreement and can be used only in accordance with this License agreement.

The hard and software as well as the operating instruction are subject to change without notice. JETI Technische Instrumente GmbH assumes no liability or responsibility for inaccuracies and errors in the operating instruction.

It is not allowed to copy this documentation or parts of it without previous written permission by JETI Technische Instrumente GmbH.

May 24, 2012