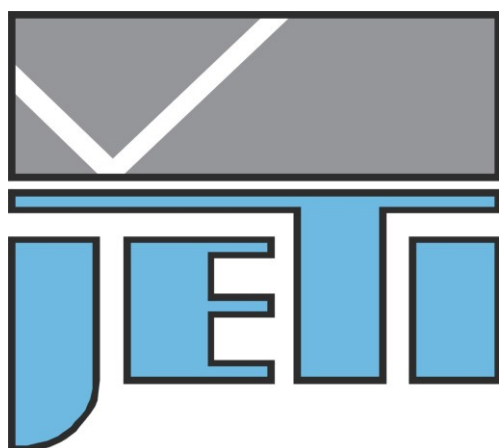


# **Programmer's Guide**

## **JETI Software Development Kit jeti\_spectro\_ex.dll**

**Version 4.8.x**



**JETI Technische Instrumente GmbH  
Göschwitzer Str. 48  
D-07745 Jena**

**Tel.: +49-3641-2329200**

**Fax: +49-3641-2329201**

**e-mail: [sales@jeti.com](mailto:sales@jeti.com)**

**internet: [www.jeti.com](http://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_GETSPECTROEXDLLVERSION .....	6
3.2	JETI_GETNUMSPECTROEX .....	7
3.3	JETI_GETSERIALSPECTROEX .....	8
3.4	JETI_OPENSPECTROEX .....	9
3.5	JETI_CLOSESPECTROEX .....	10
3.6	JETI_STARTDARKEX .....	11
3.7	JETI_DARKPIXEX .....	12
3.8	JETI_DARKWAVEEX .....	13
3.9	JETI_STARTLIGHTEX .....	14
3.10	JETI_PREPARELIGHTEX .....	15
3.11	JETI_LIGHTPIXEX .....	16
3.12	JETI_LIGHTWAVEEX .....	17
3.13	JETI_STARTREFEREX .....	18
3.14	JETI_PREPAREREFEREX .....	19
3.15	JETI_REFERPIXEX .....	20
3.16	JETI_REFERWAVEEX .....	21
3.17	JETI_STARTTRANSREFLEX .....	22
3.18	JETI_PREPARETRANSREFLEX .....	23
3.19	JETI_TRANSREFLPIXEX .....	24
3.20	JETI_TRANSREFLWAVEEX .....	25
3.21	JETI_SPECTROSTATUSEX .....	26
3.22	JETI_SPECTROBREAKEX .....	27
3.23	JETI_PIXELCOUNTEX .....	28
3.24	JETI_SPECTROTINTEX .....	29
4	EXAMPLES .....	30
4.1	C EXAMPLES .....	30
4.2	LABVIEW EXAMPLES .....	30
4.3	VISUALBASIC / VBA EXAMPLES .....	30
5	APPENDIX A .....	31
6	LICENSE AGREEMENT .....	32
7	SERVICE .....	33

# 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_ex.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 (Windows 10/11).

The second version is for real 64 bit programs under the 64 bit versions of Windows 10/11.

There are no differences in the functionality between the two versions.

### 2.1 How to communicate

In general, the user initiates communication with the target device(s) by making a call to **Fehler! Verweisquelle konnte nicht gefunden werden.** This call will return the number of target devices. This number is then used as a range when calling **Fehler! Verweisquelle konnte nicht gefunden werden.** to build a list of device serial numbers.

To access a device, it must first be opened by a call to **Fehler! Verweisquelle konnte nicht gefunden werden.** using an index determined from the call to **Fehler! Verweisquelle konnte nicht gefunden werden.** The **Fehler! Verweisquelle konnte nicht gefunden werden.** 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 **Fehler! Verweisquelle konnte nicht gefunden werden.**

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
WORD (unsigned short integer)	16	0	65535
BOOL (int)	32	$-2^{31}$	$2^{31}-1$
DWORD (unsigned long)	32	0	$2^{32}-1$
DWORD_PTR (unsigned long integer) (unsigned long long)	32 (32bit DLLs) 64 (64bit DLLs)	0 0	$2^{32}-1$ $2^{64}-1$
FLOAT (IEEE standard)	32	-3.40282E+38	3.40282E+38

### 3.1 JETI\_GetSpectroExDLLVersion

This function returns the current version number of the jeti\_spectro\_ex DLL.

#### Prototype

DWORD JETI\_GetSpectroExDLLVersion (WORD \*wMajorVersion, WORD \*wMinorVersion, WORD \*wBuildNumber)

#### 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 JETI_SUCCESS 0x... see Appendix A for error codes

## 3.2 JETI\_GetNumSpectroEx

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

### Prototype

DWORD JETI\_GetNumSpectroEx (DWORD \*dwNumDevices)

### 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 JETI_SUCCESS 0x... see Appendix A for error codes

### 3.3 JETI\_GetSerialSpectroEx

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 *Fehler! Verweisquelle konnte nicht gefunden werden.* – 1.

#### Prototype

DWORD JETI\_GetSerialSpectroEx (DWORD dwDeviceNum, char \*cBoardSerialNr, char \*cSpecSerialNr, char \*cDeviceSerialNr)

#### Parameters

##### Input

Name	Type	Description	Call
dwDeviceNum	DWORD	index of the device for which the serial numbers are desired	By value
cBoardSerialNr	char *	address of a string that will contain the electronics serial number	By reference
cSpecSerialNr	char *	address of a string that will contain the spectrometer serial number	By reference
cDeviceSerialNr	char *	address of a string that will contain the device serial number	By reference

##### Return Value

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



### 3.4 JETI\_OpenSpectroEx

Opens a device (using device number returned by *Fehler! Verweisquelle konnte nicht gefunden werden.*) and returns a handle which will be used for subsequent accesses.

#### Prototype

DWORD JETI\_OpenSpectroEx (DWORD dwDeviceNum, DWORD\_PTR \*dwDevice)

#### Parameters

##### Input

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

##### Return Value

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

### 3.5 JETI\_CloseSpectroEx

Closes an open device using the handle provided by *Fehler! Verweisquelle konnte nicht gefunden werden..*

#### Prototype

DWORD JETI\_CloseSpectroEx (DWORD\_PTR dwDevice)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device to close as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value

##### Return Value

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

### 3.6 JETI\_StartDarkEx

Starts a dark measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a dark measurement the shutter (if present) is closed, and the internal lamp (if present) is off.

**NOTE:** The function will initiate the measurement and returns immediately. To check if the measurement has finished the function *Fehler! Verweisquelle konnte nicht gefunden werden.* should be used.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_StartDarkEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement (1...65000)	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.7 JETI\_DarkPixEx

Returns the ADC-counts per pixel of a previously performed dark measurement, as initiated by

***Fehler! Verweisquelle konnte nicht gefunden werden..***

The function takes as an argument a pointer to an array where the dark values will be stored. The array must provide space for at least as many values as the used detector array has pixel.

To obtain the count of pixel of the used detector array call the function ***Fehler! Verweisquelle konnte nicht gefunden werden..***

**NOTE:** A previously initiated dark measurement must be finished, before calling this function!  
Call function ***Fehler! Verweisquelle konnte nicht gefunden werden.*** to check if the measurement has finished.

#### Prototype

DWORD JETI\_DarkPixEx (DWORD\_PTR dwDevice, INT32 \*iDark)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b><i>Fehler! Verweisquelle konnte nicht gefunden werden.</i></b>	By value
iDark	INT32 *	Pointer to an array where the dark values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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

### 3.8 JETI\_DarkWaveEx

Calculates the linear interpolated ADC-counts per wavelength of a previously performed dark measurement, as initiated by *Fehler! Verweisquelle konnte nicht gefunden werden.*

The function takes as an argument a pointer to an array where the dark values will be stored.

The array must provide space for at least the count of calculated values. The count of values can be calculated by subtract the start wavelength (dwBeg) from the end wavelength (dwEnd), divide the result by the wavelength step (fStep), and increase the result by 1:

$$count = \frac{dwEnd - dwBeg}{fStep} + 1$$

**NOTE:** A previously initiated dark measurement must be finished, before calling this function!  
Call function *Fehler! Verweisquelle konnte nicht gefunden werden.* to check if the measurement has finished.

#### Prototype

DWORD JETI\_DarkWaveEx (DWORD\_PTR dwDevice, DWORD dwBeg, DWORD dwEnd, FLOAT fStep, FLOAT \*fDark)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
dwBeg	DWORD	start wavelength in [nm]	By value
dwEnd	DWORD	end wavelength in [nm]	By value
fStep	FLOAT	wavelength step in [nm]	By value
fDark	FLOAT *	Pointer to an array where the dark values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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

### 3.9 JETI\_StartLightEx

Starts a light measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a light measurement the shutter (if present) is open, and the internal lamp (if present) is on.

To adapt the integration time automatically set the dwTint parameter to 0 (zero).

**NOTE:** The function will initiate the measurement and returns immediately. To check if the measurement has finished the function *Fehler! Verweisquelle konnte nicht gefunden werden.* should be used.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_StartLightEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement (pass 0 for adaption)	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.10 JETI\_PrepareLightEx

Prepares a light measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a light measurement the shutter (if present) is open, and the internal lamp (if present) is on.

To adapt the integration time automatically set the dwTint parameter to 0 (zero).

It will not start the measurement.

**NOTE:** The function will return *immediately*. Before any other DLL-function call the function *Fehler! Verweisquelle konnte nicht gefunden werden.* must be used to check if the measurement has started by an external trigger.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_PrepareLightEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement (pass 0 for adaption)	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.11 JETI\_LightPixEx

Returns the ADC-counts per pixel of a previously performed light measurement, as initiated by

***Fehler! Verweisquelle konnte nicht gefunden werden..***

The function takes as an argument a pointer to an array where the light values will be stored. The array must provide space for at least as many values as the used detector array has pixel.

To obtain the count of pixel of the used detector array call the function ***Fehler! Verweisquelle konnte nicht gefunden werden..***

**NOTE:** A previously initiated light measurement must be finished, before calling this function!  
Call function ***Fehler! Verweisquelle konnte nicht gefunden werden.*** to check if the measurement has finished.

#### Prototype

DWORD JETI\_LightPixEx (DWORD\_PTR dwDevice, INT32 \*iLight)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b><i>Fehler! Verweisquelle konnte nicht gefunden werden.</i></b>	By value
iLight	INT32 *	Pointer to an array where the light values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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



### 3.12 JETI\_LightWaveEx

Calculates the linear interpolated ADC-counts per wavelength of a previously performed light measurement, as initiated by **Fehler! Verweisquelle konnte nicht gefunden werden.**

The function takes as an argument a pointer to an array where the light values will be stored.

The array must provide space for at least the count of calculated values. The count of values can be calculated by subtract the start wavelength (dwBeg) from the end wavelength (dwEnd), divide the result by the wavelength step (fStep), and increase the result by 1:

$$count = \frac{dwEnd - dwBeg}{fStep} + 1$$

**NOTE:** A previously initiated light measurement must be finished, before calling this function!  
Call function **Fehler! Verweisquelle konnte nicht gefunden werden.** to check if the measurement has finished.

#### Prototype

DWORD JETI\_LightWaveEx (DWORD\_PTR dwDevice, DWORD dwBeg, DWORD dwEnd, FLOAT fStep, FLOAT \*fLight)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b>	By value
dwBeg	DWORD	start wavelength in [nm]	By value
dwEnd	DWORD	end wavelength in [nm]	By value
fStep	FLOAT	wavelength step in [nm]	By value
fLight	FLOAT *	Pointer to an array where the light values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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

### 3.13 JETI\_StartReferEx

Starts a reference measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a reference measurement the shutter (if present) is open, and the internal lamp (if present) is on. If no dark measurement was performed before using the same integration time, then a dark measurement is also performed.

To adapt the integration time automatically set the dwTint parameter to 0 (zero).

**NOTE:** The function will initiate the measurement and returns immediately. To check if the measurement has finished the function *Fehler! Verweisquelle konnte nicht gefunden werden.* should be used.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_StartReferEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement (pass 0 for adaption)	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.14 JETI\_PrepareReferEx

Prepares a reference measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a reference measurement the shutter (if present) is open, and the internal lamp (if present) is on. If no dark measurement was performed before using the same integration time, then a dark measurement is also performed.

To adapt the integration time automatically set the dwTint parameter to 0 (zero).

It will not start a measurement.

**NOTE:** The function will return *immediately*. Before any other DLL-function call the function ***Fehler! Verweisquelle konnte nicht gefunden werden.*** must be used to check if the measurement has started by an external trigger.

After the measurement has finished use function ***Fehler! Verweisquelle konnte nicht gefunden werden.*** or ***Fehler! Verweisquelle konnte nicht gefunden werden.*** to get the measuring results.

#### Prototype

DWORD JETI\_StartReferEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b><i>Fehler! Verweisquelle konnte nicht gefunden werden.</i></b>	By value
fTint	FLOAT	Integration time in [ms] for the measurement (pass 0 for adaption)	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.15 JETI\_ReferPixEx

Returns the ADC-counts per pixel of a previously performed reference measurement, as initiated by ***Fehler! Verweisquelle konnte nicht gefunden werden.***

The function takes as an argument a pointer to an array where the reference values will be stored. The array must provide space for at least as many values as the used detector array has pixel.

To obtain the count of pixel of the used detector array call the function ***Fehler! Verweisquelle konnte nicht gefunden werden.***

**NOTE:** A previously initiated reference measurement must be finished, before calling this function!  
Call function ***Fehler! Verweisquelle konnte nicht gefunden werden.*** to check if the measurement has finished.

#### Prototype

DWORD JETI\_ReferPixEx (DWORD\_PTR dwDevice, INT32 \*iRefer)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b><i>Fehler! Verweisquelle konnte nicht gefunden werden.</i></b>	By value
iRefer	INT32 *	Pointer to an array where the reference values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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

### 3.16 JETI\_ReferWaveEx

Calculates the linear interpolated ADC-counts per wavelength of a previously performed reference measurement, as initiated by **Fehler! Verweisquelle konnte nicht gefunden werden.**

The function takes as an argument a pointer to an array where the reference values will be stored.

The array must provide space for at least the count of calculated values. The count of values can be calculated by subtract the start wavelength (dwBeg) from the end wavelength (dwEnd), divide the result by the wavelength step (fStep), and increase the result by 1:

$$count = \frac{dwEnd - dwBeg}{fStep} + 1$$

**NOTE:** A previously initiated reference measurement must be finished, before calling this function!

Call function **Fehler! Verweisquelle konnte nicht gefunden werden.** to check if the measurement has finished.

#### Prototype

DWORD JETI\_ReferWaveEx (DWORD\_PTR dwDevice, DWORD dwBeg, DWORD dwEnd, FLOAT fStep, FLOAT \*fRefer)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b>	By value
dwBeg	DWORD	start wavelength in [nm]	By value
dwEnd	DWORD	end wavelength in [nm]	By value
fStep	FLOAT	wavelength step in [nm]	By value
fRefer	FLOAT *	Pointer to an array where the reference values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

##### Return Value

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

### 3.17 JETI\_StartTransRefEx

Starts a transmission/reflection measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a transmission/reflection measurement the shutter (if present) is open, and the internal lamp (if present) is on.

**NOTE:** To get reasonable measuring results, a reference measurement with the same integration time has to be performed before calling this function.

**NOTE:** The function will initiate the measurement and returns immediately. To check if the measurement has finished the function *Fehler! Verweisquelle konnte nicht gefunden werden.* should be used.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_StartReferEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.18 JETI\_PrepareTransRefEx

Prepares a transmission/reflection measurement with an integration time of dwTint. To get the average values of up to 10000 measurements set the parameter wAver to the corresponding value. To perform only one measurement without averaging set the wAver parameter to 1.

During a transmission/reflection measurement the shutter (if present) is open, and the internal lamp (if present) is on.

It will not start a measurement.

**NOTE:** To get reasonable measuring results, a reference measurement with the same integration time has to be performed before calling this function.

**NOTE:** The function will return *immediately*. Before any other DLL-function call the function *Fehler! Verweisquelle konnte nicht gefunden werden.* must be used to check if the measurement has started by an external trigger.

After the measurement has finished use function *Fehler! Verweisquelle konnte nicht gefunden werden.* or *Fehler! Verweisquelle konnte nicht gefunden werden.* to get the measuring results.

#### Prototype

DWORD JETI\_PrepareReferEx (DWORD\_PTR dwDevice, FLOAT fTint, WORD wAver)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT	Integration time in [ms] for the measurement	By value
wAver	WORD	Count of scans for averaging (if no averaging is needed, set this value to 1)	By value

##### Return Value

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

### 3.19 JETI\_TransReflPixEx

Returns the ADC-counts per pixel of a previously performed transmission/reflection measurement, as initiated by *Fehler! Verweisquelle konnte nicht gefunden werden.*

The function takes as an argument a pointer to an array where the reference values will be stored. The array must provide space for at least as many values as the used detector array has pixel.

To obtain the count of pixel of the used detector array call the function *Fehler! Verweisquelle konnte nicht gefunden werden.*

**NOTE:** A previously initiated transmission/reflection measurement must be finished, before calling this function! Call function *Fehler! Verweisquelle konnte nicht gefunden werden.* to check if the measurement has finished.

#### Prototype

DWORD JETI\_TransReflPixEx (DWORD\_PTR dwDevice, INT32 \*iTransRefl)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
iTransRefl	INT32 *	Pointer to an array where the transmission/reflection values will be stored  (THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)	By reference

##### Return Value

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



## 3.20 JETI\_TransReflWaveEx

Calculates the linear interpolated ADC-counts per wavelength of a previously performed transmission/reflection measurement, as initiated by *Fehler! Verweisquelle konnte nicht gefunden werden..*

The function takes as an argument a pointer to an array where the transmission/reflection values will be stored.

The array must provide space for at least the count of calculated values. The count of values can be calculated by subtract the start wavelength (dwBeg) from the end wavelength (dwEnd), divide the result by the wavelength step (fStep), and increase the result by 1:

$$count = \frac{dwEnd - dwBeg}{fStep} + 1$$

**NOTE:** A previously initiated transmission/reflection measurement must be finished, before calling this function! Call function *Fehler! Verweisquelle konnte nicht gefunden werden.* to check if the measurement has finished.

### Prototype

DWORD JETI\_TransReflWaveEx (DWORD\_PTR dwDevice, DWORD dwBeg, DWORD dwEnd, FLOAT fStep, FLOAT \*fRefer)

### Parameters

#### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
dwBeg	DWORD	start wavelength in [nm]	By value
dwEnd	DWORD	end wavelength in [nm]	By value
fStep	FLOAT	wavelength step in [nm]	By value
fTransRefl	FLOAT *	Pointer to an array where the transmission/reflection values will be stored  <b>(THE ARRAY MUST PROVIDE ENOUGH SPACE, SEE DESCRIPTION ABOVE)</b>	By reference

#### Return Value

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

### 3.21 JETI\_SpectroStatusEx

Determines the status of the device. This function can be used to check if an initiated measurement has finished.

bolsBusy = TRUE (1) means that the device is performing a measurement and/or is busy.

bolsBusy = FALSE (0) indicates that the device is in an idle state.

If a measurement was initiated with automatic adaption of integration time and the measurement could not be performed because of overexposure bolsBusy will be switched to FALSE (0) and the function will return an error code 0x20.

#### Prototype

DWORD JETI\_SpectroStatusEx (DWORD\_PTR dwDevice, BOOL \*bolsBusy)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
bolsBusy	BOOL *	Pointer to a variable where the status will be stored	By reference

##### Return Value

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

## 3.22 JETI\_SpectroBreakEx

This function cancels an initiated measurement.

### Prototype

DWORD JETI\_SpectroBreakEx (DWORD\_PTR dwDevice)

### Parameters

#### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value

#### Return Value

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

### 3.23 JETI\_PixelCountEx

Obtains the counts of pixel of the used detector array. This value may be useful to determine the needed size of the array for the functions *Fehler! Verweisquelle konnte nicht gefunden werden.*, *Fehler! Verweisquelle konnte nicht gefunden werden.*, *Fehler! Verweisquelle konnte nicht gefunden werden.* and *Fehler! Verweisquelle konnte nicht gefunden werden.*.

#### Prototype

DWORD JETI\_PixelCountEx (DWORD\_PTR dwDevice, DWORD \*dwPixel)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
dwPixel	DWORD *	Pointer to a variable where the pixel count will be stored	By reference

##### Return Value

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

### 3.24 JETI\_SpectroTintEx

Obtains the last used integration time in [ms].

#### Prototype

DWORD JETI\_SpectroTintEx (DWORD\_PTR dwDevice, FLOAT \*fTint)

#### Parameters

##### Input

Name	Type	Description	Call
dwDevice	DWORD_PTR	Handle to a device as returned by <i>Fehler! Verweisquelle konnte nicht gefunden werden.</i>	By value
fTint	FLOAT *	Pointer to a variable where the integration time will be stored	By reference

##### Return Value

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

## 4 Examples

To help starting development the SDK includes several examples.

### 4.1 C Examples

#### RadioSample

This sample demonstrates the basic usage of the jeti\_radio DLL.

#### SyncSample

The SyncSample demonstrates the use of special functions to synchronize the measurements integration time with the frequency of pulsed light sources and pulsed monitor back-lights.

#### TriggerSample

This sample demonstrates the handle of measurements initiated by an external trigger event.

### 4.2 LabVIEW Examples

These samples demonstrate the basic usage of the DLLs within a LabVIEW program.

### 4.3 VisualBasic / VBA Examples

These sample demonstrate the usage of the jeti\_radio DLL within a VBA macro inside an excel spreadsheet.

## 5 Appendix A

Error codes and their description:

Error code	#define	Description
0x00	JETI_SUCCESS	no error occurred
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_OVEREXPOSURE	measurement failed due to overexposure
0x22	JETI_MEASURE_FAIL	measurement failed due to other reasons
0x23	JETI_ADAPTION_FAIL	adaption failed
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 **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden.**) 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 **Fehler! Verweisquelle konnte nicht gefunden werden.** 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 **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden.**
- 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 **Fehler! Verweisquelle konnte nicht gefunden werden.** and manually reset the device.

## 6 License Agreement

This End-User License Agreement ("EULA") is a legal agreement between you and JETI Technische Instrumente GmbH ("JETI") for the JETI SDK product which may include associated software components, media, printed materials, and "online" electronic documentation ("SOFTWARE PRODUCT").

You agree to be bound by the terms of this EULA by using the SOFTWARE PRODUCT. If you do not agree, do not use it.

### 1. COPYRIGHT

All rights, title, interests in and to copyrights in the SOFTWARE PRODUCT are owned exclusively (or licensed) by JETI. The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

### 2. GRANT OF LICENSE

JETI grants Customer the following rights:

- a. Customer is granted the non-transferable, non-exclusive, and perpetual right to integrate the SOFTWARE PRODUCT in their software products, to use these products in their own company and to distribute these products under their own trade name.
- b. Customer is NOT required to pay any distribution, runtime, royalty, per-developer or per-end-user license fees.
- c. Customer may make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.

### 3. RESTRICTIONS

- a. Customer may not disassemble, decompile, reverse engineer the SOFTWARE PRODUCT.
- b. Customer may not transfer, modify, sell, lease or sublicense the SOFTWARE PRODUCT.

### 4. TERMINATION

Without prejudice to any other rights, JETI may terminate this EULA if Customer fails to comply with the terms and conditions of this EULA. In such event, Customer must destroy all copies of the SOFTWARE PRODUCT in your possession.

### 5. NO WARRANTIES

JETI expressly disclaims any warranty for the SOFTWARE PRODUCT. JETI SDK is provided "As Is" without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, noninfringement, or fitness of a particular purpose. JETI does not warrant or assume responsibility for the accuracy or completeness of any information, text, algorithms, graphics, links or other items contained within this SOFTWARE PRODUCT. JETI will not be liable for data loss, damages, loss of profits or any other kind of loss while using or misusing this library.



## 7 Service

In case of any questions or technical problems please contact:

JETI Technische Instrumente GmbH Göschwitzer Str. 48 D-07745 Jena Tel. +49 3641 23292 00 Fax +49 3641 23292 01 e-mail: <a href="mailto:sales@jeti.com">sales@jeti.com</a> Internet: <a href="http://www.jeti.com">www.jeti.com</a>
--

Copyright (c) 2024 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.

February 9, 2024