rgb photonics

Software Development Kit

RgbDriverKit
Quick Start Guide

Version: 2.0.0
Date: November 30th, 2017

Donaupark 13
93309 Kelheim
Germany

Telephone: +49 9441 1750 33 – 0
Website: http://www.rgb-photonics.com
E-Mail: sales@rgb-photonics.com

Contents

1 Overview

This manual explains how to control one of our lasers or spectrometers from your own software using the Windows .NET library called “RgbDriverKit.dll”. This library provides a modern object-oriented Application Programming Interface (API). It supports all device features including recalibration. The Software Development Kit (SDK) includes this library as well as sample code and a documentation help file.

2 Using the RgbDriverKit.dll library

The SDK includes a software library called RgbDriverKit.dll that can be used to control devices and perform measurements from your own custom software. This is a Dynamic Link Library (DLL) for Windows using the .NET framework version 3.5 or 4.0. (Please note that .NET DLLs and traditional C DLLs are completely different kinds of software libraries, despite having a similar name.)

This chapter contains an overview for a quick start using this library. A detailed programming reference can be found in the help file “RgbDriverKit documentation.chm“.

2.1 Prerequisites

The system requirements for this library are:

- Windows 10, 8, 7, Vista or XP (32 or 64 bit)
- .NET Framework Version 3.5 or 4.0 (Already included with most versions of Windows. Otherwise available on the CD or to be downloaded from Microsoft.)

You can use this library with any programming language that can connect to .NET DLLs. This includes C#, Visual Basic .NET, C++/CLI, Delphi, LabVIEW, Matlab and Mathematica. You should have a basic understanding of the concepts of object-oriented programming in order to understand the structure and functionality of the library. The SDK includes sample code that demonstrates how to access the DLL from LabView, Matlab, C#, C++ and using COM.

2.2 Getting started

In order to get familiar with the library quickly, we suggest to:

1. Read the next chapter 2.3 to understand the structure of the SDK library.
2. Read the basic steps required in order to take a spectrum (chapter 2.5).
3. Study the sample projects included in the SDK.
4. Add the SDK library to your own project (chapter 2.4).
5. Implement the basic functionality of taking a spectrum in your project.
6. Implement advanced features as required.

2.3 Classes Overview

The SDK library contains classes for controlling a spectrometer or laser as well as abstract base classes for various types of measurement and control devices.

For example, the main class that you need in order to communicate with a spectrometer that includes a USB Type-C connector is:

`Qseries`
which is a descendant of the `CalibratedSpectrometer` class,
which is a descendant of the `Spectrometer` class,
which is a descendant of the `Device` class.

The `Device` class includes basic features used by all kinds of measurement and control devices. `Spectrometer` is a base class for a simple spectrometer. `CalibratedSpectrometer` is a base class for a spectrometer that supports pre-processing of the spectra. `Qseries` implements all the features specified in the abstract base classes in order to control a Qmini spectrometer.

The library also includes the `SimulatedSpectrometer` class, that may be used instead of `Qseries` to test your software in case you don't have a real device available, just like choosing "Simulated Spectrometer" in Waves.

You can find the full documentation for all relevant classes in the SDK documentation contained in the `RgbDriverKit documentation.chm` help file.

(Note: For spectrometers with Mini-USB connectors please use the `RgbSpectrometer` class instead.)

2.4 Adding the library to your project

Please copy the `RgbDriverKit.dll` file into your project folder. The device driver for Windows must also be present.

Visual Studio

If you are using one of Microsoft's languages, please add the library in Visual Studio to your project references (in the Solution Explorer window). Then import the `RgbDriverKit` library to your source code files (for example in C# or C++: "`using RgbDriverKit;`"). For enhanced IntelliSense support in Visual Studio, you may also want to copy the `RgbDriverKit.xml` file into the same folder as the library.

LabVIEW

In order to use a .NET DLL in LabVIEW, you need to:

- Use the .NET palette (Connectivity -> .NET) to find all of the functions available.
- Use a constructor node in order to instantiate a class within the .NET assembly.
- Use property nodes and invoke nodes to access properties and methods of the class by wiring in the class reference from the constructor node to the property or invoke node.

Please also see the LabVIEW sample code in the SDK.

Other languages

For more information on how to connect to the library from other programming languages, please see if there is a suitable sample code included in the SDK. Otherwise you may try a Google search for something like "use .NET DLL in [your language]".

2.5 Taking a spectrum

To control a spectrometer (with a USB Type-C connector) you first need to instantiate a `Qseries` object. You could use the class constructor, of course, but in most cases it'll be easier to search for spectrometers by calling the static `SearchDevices()` method and then choose from one of the returned objects, which are actual instances of the `Qseries` class.

Then, the steps required for taking a simple spectrum are:

1. Open the connection to the device using the `Open()` method.
2. If you need the wavelengths, get an array containing the wavelengths of each pixel with `GetWavelengths()`.
3. Set the exposure time using the `ExposureTime` property.
4. Start the exposure with the `StartExposure()` method.
5. Wait (or do something useful) while the `Status` property is `SpectrometerStatus.TakingSpectrum`.
6. Read the spectrum with the `GetData()` method.

The full documentation for these methods and properties can be found in the `RgbDriverKit documentation.chm` help file.

All methods and some properties may throw an exception as described in the SDK documentation help file. Always make sure that these exceptions are properly handled by placing these methods inside a `try{} catch{}` block.

2.6 Deploying your project

If you would like to deploy your project to other users, please:

- Include the DLL file `RgbDriverKit.dll`.
- Make sure that the .NET framework 3.5 or 4.0 is installed.
- Make sure that the Windows device driver is installed.

The device driver is normally installed during the installation of the application software Waves or Ltune. If you would like to use the device without installing Waves or Ltune, you can find the files for the device driver in the "WinUSB driver" folder in the SDK.

3 SDK Contents

The latest version of the SDK is available for download at:
http://www.rgb-photonics.com/downloads

3.1 Software

The SDK includes RgbDriverKit.dll, the main Windows .NET DLL. If the regular application software (Waves or Ltune) is not installed, you may need the device drivers for Windows, which can be found in the folder "WinUSB device driver".

3.2 Sample Code

The SDK includes the following sample codes:

- SimpleSpectrometer
- MultipleSpectrometers
- LabView
- Matlab
- C++ Wrapper
- COM Wrapper

3.3 Further Documentation

The SDK also includes the following documentation files:

- RgbDriverKit documentation.chm
- SDK release notes.txt
- API changes.txt

4 Technical Support

You can contact our engineers by sending an e-mail to support@rgb-photonics.com.

Our sales department can be reached by e-mail, phone, fax or regular mail:

RGB Photonics GmbH
Donaupark 13
93309 Kelheim
Phone: +49 9441 175033 0
Fax: +49 9441 175033 92
E-Mail: sales@rgb-photonics.com

You can find additional resources on our website
http://www.rgb-photonics.com
by clicking on “Support” on the menu bar.