Product Document

Published by
ams OSRAM Group

User Guide

UG000400

AS7341 11-Channel Spectral Sensor

AS7341 Evaluation Kit

v6-00 • 2022-Sep-08

Content Guide

1 Introduction

This Evaluation Kit (EVK) is a platform to evaluate the 11-channel spectral sensor applications for the AS7341 and to demonstrate different use cases. It realizes basic functions to get the sensor's ADC counts based on alternative setups and includes special functions - to illustrate application-specific tasks (Ambient Light Sensing (ALS), Reflection Mode, Flicker Detection and Spectral, and Color Mask Compare). These functions may need a hardware adaptation (e.g. an LED on the board for reflection) or an additional optional mechanical interface, which must be ordered separately. In general, there are two standard evaluation kits available, which are different in use case and optical adapters.

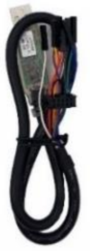
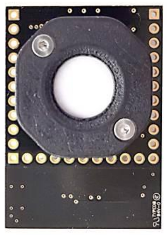
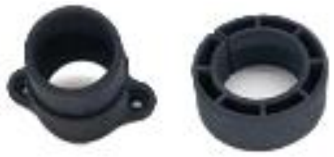
- The standard ALS kit is for measuring contactless ambient lighting or light sources - it includes the hardware version without onboard LEDs but with a diffuser in front of the sensor with a maximum field of view. It represents the standard kit (see Figure 1) for light measurement/detection or liquid measurements during transmission.
- The extended kit (Reflection mode) is for contact measurement of colored surfaces (see Figure 2) and consists of the sensor hardware but with a pre-assembled LED and a special adapter in front of the sensor with 0° (Sensor)/45° (LEDs) geometry.

This user guide describes the features and functions for both variants, all supported sensor types, and sensor hardware connected to a PC with an I²C-FTDI Interface. The software and driver are limited to a PC with MS Windows.

1.1 Kit Content

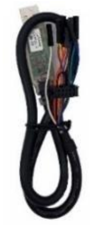
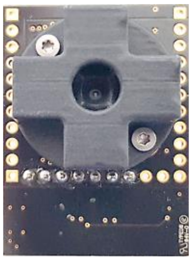
The Evaluation Kit contains the following items, shown in Figure 1.

Figure 1
Kit Content of EVK Variant ALS

Item 1	Item 2	Item 3	Item 4
FTDI	**EVK with Diffuser in Front Adapter**	**USB Stick**	**Special Backside Adapter**
			

Item No.:	Item	Comment
1	FTDI - USB Cable	USB – I²C Cable with 10-pole IDC Connector, 3.3 V.
2	a0013a0_CSS EVAL KIT AS7341	Evaluation Kit with pre-assembled front adapter and a diffuser.
3	USB Data Stick	Documents, software, firmware, and drivers.
4	Special Backside Adapter EVAL Linos 16 EVAL Linos 16-25	3D-Printing part, used to adapt the 16 mm holder into: ● 16 mm Linos-Nano-Bank (above) ● 25 mm Linos-Nano-Bank (below)

Figure 2 :
Kit Content of EVK Variant Reflection

Item 1 FTDI	Item 2 EVAL KIT with LED and 0°/45° Optical Front Adapter	Item 3 USB Stick	Item 4 Special Backside Adapter
			

Item No.:	Item	Comment
1	FTDI - USB Cable	USB – I²C Cable with 10 pole IDC Connector, 3.3 V.
2	a0013a0_CSS EVAL KIT AS7341 Reflection	Evaluation Kit with a pre-assembled LED and 0°/45° front adapter (without a diffuser[1]).
3	USB Data Stick	Documents, software, firmware, and drivers.
4	Special Backside Adapter EVAL Linos 16 EVAL Linos 16-25	3D-Printing part, used to adapt the 16 mm holder into: ● 16 mm Linos-Nano-Bank (above) ● 25 mm Linos-Nano-Bank (below)

[1] Customers should add a diffuser in front of the sensor in the case of a nondiffusible application.

1.2 Ordering Information

Ordering Code	Description
AS7341 EVAL KIT	AS7341 11-Channel Spectral Sensor Evaluation Kit.
AS7341 EVK REFLECTION	AS7341 11-Channel Spectral Sensor Evaluation Kit for reflection with pre-assembled LEDs and a special 0°/45° adapter.

Attention

Please order the optional parts separately.

Information

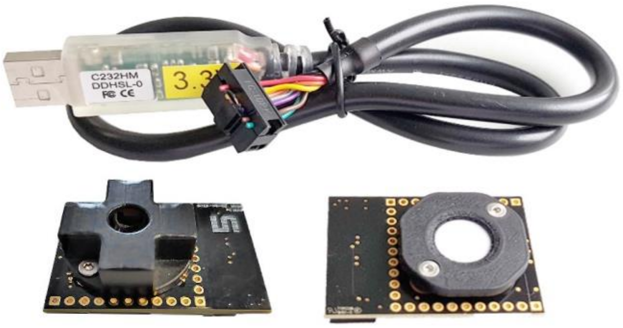
It is also possible to print or modify the customized adapters. See the document path of the USB Stick for the 3D models.

2 Getting Started

The Evaluation Kit consists of the FTDI cable (I²C to USB converter) and the sensor board with pre-mounted adapters. Adapters on the front side are necessary to meet the optical requirements of the sensor filter specification or to adapt the sensor to application-specific requirements (FOV field of view or 0°/45° geometry). Adapters on the rear side mount the EVK in mechanical test systems.

The sensor board has an I²C interface. The 3.3 V FTDI-adapter (with a USB interface and a COM-Port driver) converts the sensor signals into the PC GUI. Figure 3 shows the FTDI cable (above) and the sensor board for Reflection Mode (below left), and the standard version e.g. for ALS Light Detection (right).

Figure 3:
Evaluation Kit with Sensor Hardware, Optical Adapters (Front), and FTDI Adapter[1]



[1] The accessories may differ from the picture.

To get started with the Evaluation Kit, perform the following steps:

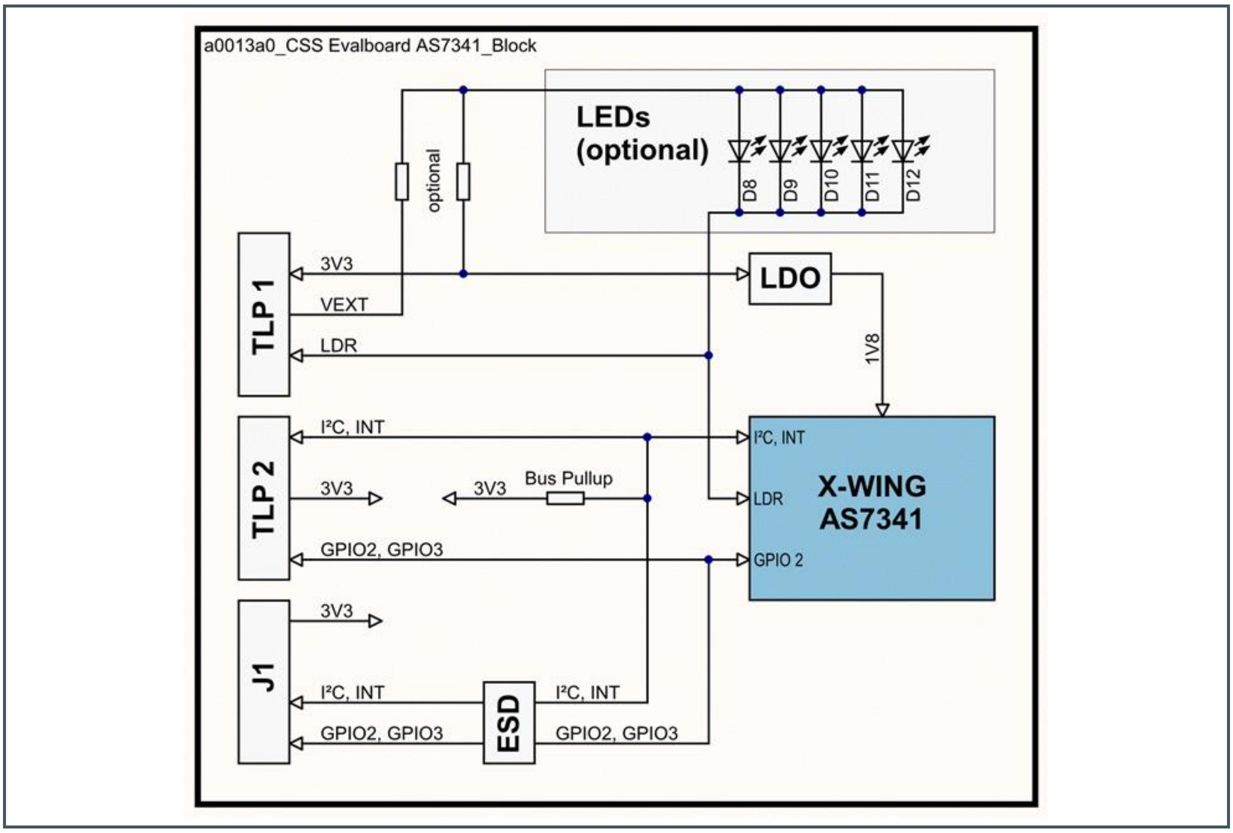
1. Select the correct Evaluation Kit for your application.
2. Install the GUI (MSI) and all the drivers from the USB stick. Please read the details in chapter 6.1.
3. Plug the FTDI cable into the socket of the sensor board. Then, connect the kit to the PC.
4. If there is an issue during the installation process, please read the details in chapter 6.1.

3 Hardware Description

3.1 Hardware Architecture

The evaluation kit includes an LDO to provide the 1.8 V supply voltage for the sensor chip, ESD protection diodes for the I²C bus and GPIO lines, and a placeholder for optional LEDs. The LEDs[1] can be powered - either from the FTDI adapter or externally via the TLP1 connector. This is achieved with a 10-pole IDC socket for connecting the FTDI adapter cable (J1) and 1/10 inch rows of holes for mounting on a 1/10 inch hole grid plate or by directly contacting signals (TLP1 to TLP2).

Figure 4:
Block Diagram of AS7341 Board



[1] Must be soldered by the customer or use AS7341 EVK Reflection. The function of the LED current, light intensity, and/or maximal driver current depends on the used LEDs and the connected power supply. Check the LED datasheet before using this LED setting. An LED current can destroy the LEDs if it is too high.

3.2 Power Supply

Generally, the FTDI converter produces the 3.3 V power supply for the sensor board. The converter is part of the original package of the EVK. Customers can use onboard alternative pre-designed power supplies. For more details, see Figure 9.

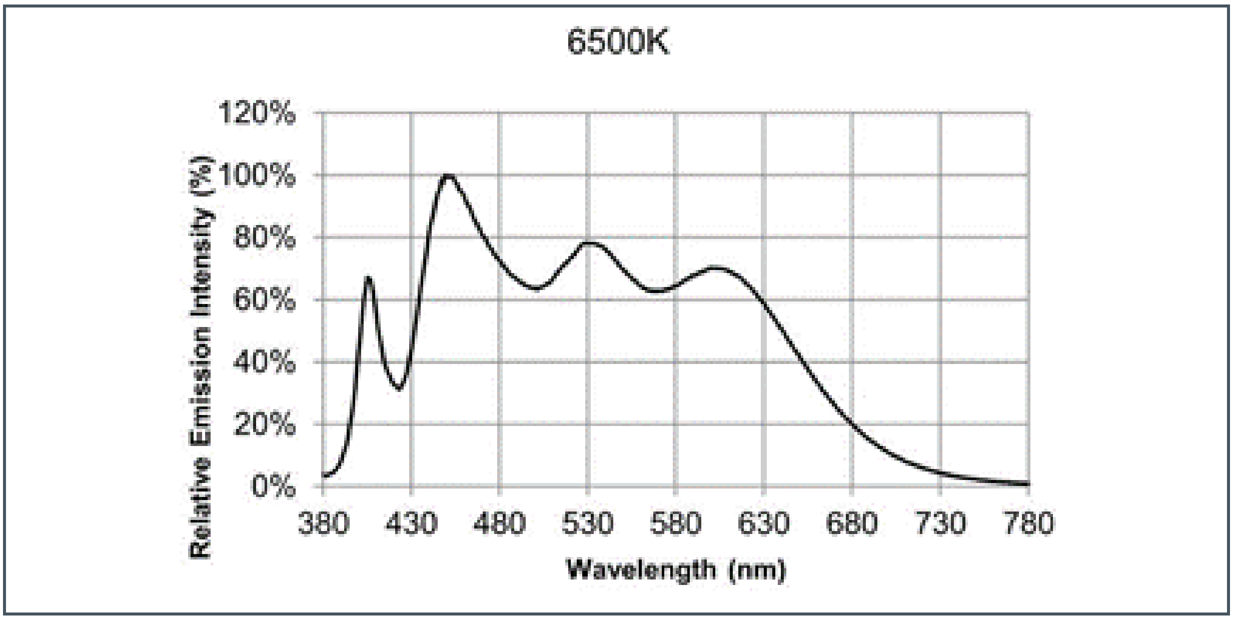
Information

Ensure to use the original adapters and converters before connecting the hardware to the USB.

3.3 Onboard LEDs

The LEDs on this evaluation board are broadband white light LEDs. Figure 5 below shows the typical spectral distribution.

Figure 5:
Typical Spectral Distribution of the Onboard LED



If the power supply of the evaluation board is 3.3 V, it will not be possible to get a current bigger than 10 mA through both LEDs. This limit is given by a 3.3 V power supply and both 10 Ω series resistors.

Figure 6:
Forward Current vs. Forward Voltage

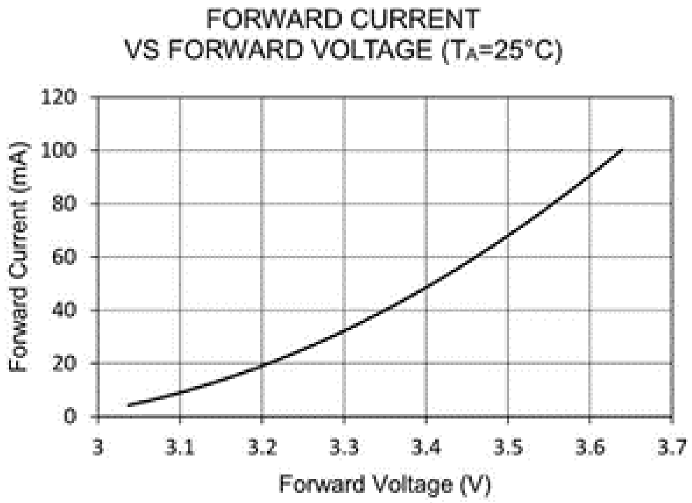
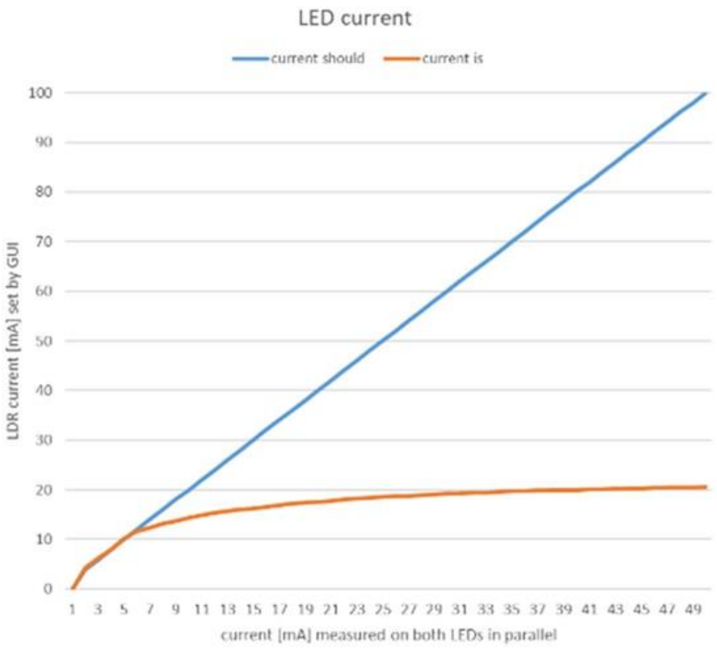


Figure 7 describes the nonlinearity between the GUI set current and the real current through the LEDs by the 3.3 V power supply.

Figure 7:
Nonlinearity Between the GUI Set Current and a Real Measured Current



It is possible to connect the evaluation board to a bigger power consumption voltage. Pin1 on TLP1 is for external LED voltage distribution. (Pin2 TLP1 is GND). R11 is to be disassembled and fitted as R13.

3.4 Connector Pinout Description

Figure 8:
Connector Pinout Description

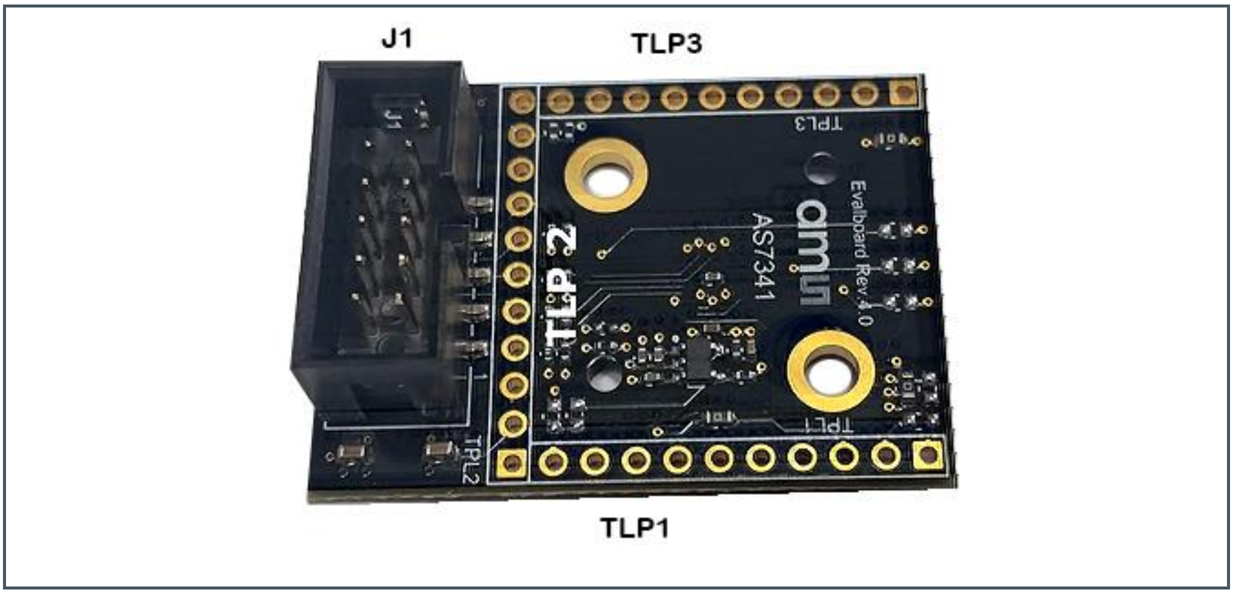


Figure 9:
Overview of Connectors and Interfaces

Designator	Comment
J1	10-pole IDC Socket, connect to the PC via the FTDI adapter (3.3 V).
TLP 1	VEXT, LDR, GND
TLP 2	3V3 (5V0), I²C, INT, GPIO
TLP 3	GND

Figure 10:
Connectors

Pin Number	Net Name	Function
TLP1 1	VEXT	Supply the optional fitted LEDs from an external source.
TLP1 2	GND	Ground
TLP1 3	LDR	Constant current sink from the sensor chip.
TLP1 4…9	NC	Not connected.
TLP1 10	1V8	External power supply - before using this pin, disassemble the LDO.
TLP 2 1	GND	Ground
TLP 2 2	NC	Not connected.
TLP 2 3	3V3	3V3 power input if there is no power on J1 (FTDI adapter) or 3V3 power output, if the power comes over J1.
TLP 2 4	GND	Ground
TLP 2 5	SDA	I²C data signal.
TLP 2 6	SCL	I²C clock signal.
TLP 2 7	INT	Sensor chip interrupt signal.
TLP 2 8	GPIO 2	GPIO signal, bridged to the FTDI adapter.
TLP 2 9	GND	Ground
TLP 2 10	GPIO 3	GPIO signal, bridged to the FTDI adapter and sensor chip.
TLP 2 11	SDA2	Normally bridged to the SDA, only needed for the FTDI adapter.
TLP 3 1..5, 7..10	NC	Not connected.
TLP 3 6	GND	Ground
J1 1	NC	Not connected.
J1 2	3V3	3V3 power input from the FTDI adapter.
J1 3	GND	Ground
J1 4	SDA	I²C data signal.
J1 5	SCL	I²C clock signal.
J1 6	INT	Sensor chip Interrupt signal.
J1 7	GPIO 2	GPIO signal, bridged to TPL 2.
J1 8	GND	Ground
J1 9	GPIO 3	GPIO signal, bridged to the sensor chip.
J1 10	SDA2	Normally bridged to the SDA, only needed for the FTDI adapter.

4 Optical Diffuser

For non-diffuse applications[2], as light detection from a light source, a translucent diffuser in front of the AS7341 EVK is required - which uses scattering centers to spread incoming directed light in pseudorandom directions. These scattering centers can be tiny surface structures on the top (e.g. grounded glass) or small white particles inside (e.g. opal glass) the diffuser. This property divides diffusers into two main groups: surface diffusers and volume diffusers. Selecting a diffuser depends on the use case, the irradiance, the angular distribution of the light, and the needed flexibility of the setup in a fixed or mobile application.

A surface diffuser with high transmission and a radiation pattern as wide as possible should be used for low light measurement applications.

Narrowing the radiation pattern for higher transmission efficiency is only possible for detecting large and homogenous light sources or if the conditions for calibration are the same as measurement geometry and stability. Calibration can compensate for stable conditions, chromatic effects, deviations, and others. In applications under changing conditions, regarding size, direction, and/or orientation of the light source, a volume diffuser with nearly Lambertian and achromatic characteristics is the best choice to create a sensor system unaffected by the direction of incoming light.

In addition to the technical aspects, its material thickness, surface (robustness), availability, and price, also play a vital role in the selection of the diffuser. Compromises may be necessary.

Figure 11 lists the recommended diffuser parameters and/or the parameters of the Kimoto diffuser in the EVK (=recommended parameters for a similar diffuser).

Figure 11:
Recommended Diffuser Parameters(1)

Parameter	Value	Value
Diffuser Material	Kimoto 100 PBU	Kimoto OptSaver L-57
Diffuser Thickness	125 Microns	100 Microns
Transmission	66%	60%
Haze	89.5%	93.1%
Half-Angle	35.5°	57°

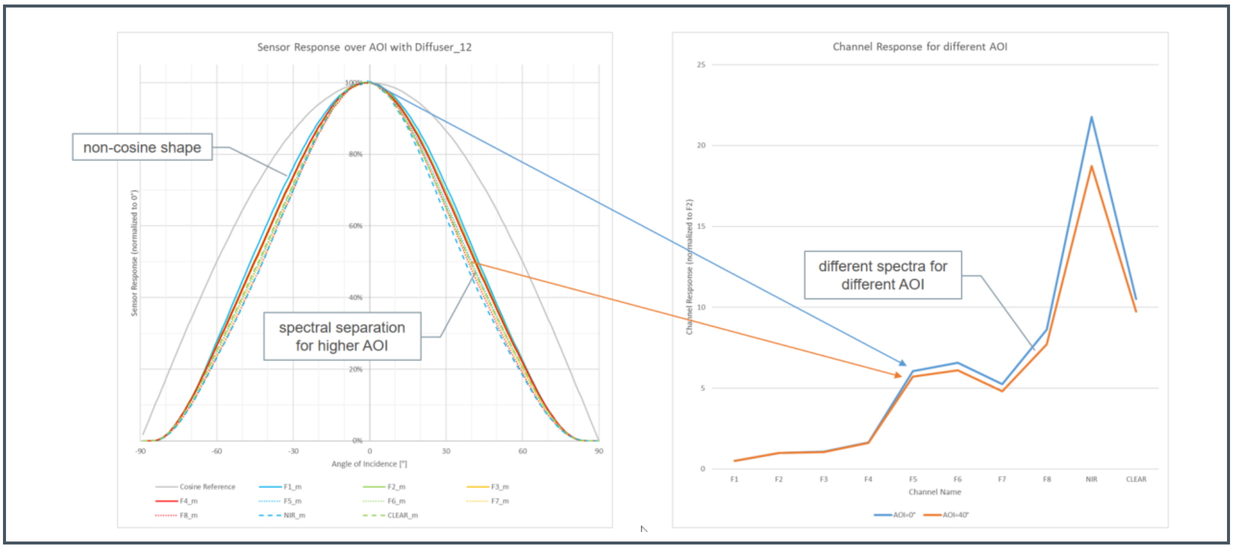
(1) The EVKs can either contain the 100 PBU or be equipped with the L-57. Information about this can be obtained from ams OSRAM support.

[2] e.g. ALS (Ambient Light Sensing) in contrast to reflections which are mostly diffused.

The AS7341 EVKs currently includes a Kimoto 100 PBU diffuser directly above the sensor. This diffuser has good technical parameters (see Figure 11), taking into account the price and availability. It is located in two simple plastic shells and screwed onto the evaluation kit (see Figure 14).

The PBU-100 diffuser is somewhat dependent on the incident angle of the light. Figure 12 shows the spectral change for different irradiation angles of the PBU-100, which can lead to deviations in dynamic applications depending on the application.

Figure 12:
Angular Dependence of the Kimoto 100-PBU for Spectral Scanning



For such cases, alternative diffusers can be used, which differ, for example, in angular dependence and possibly other parameters. One such diffuser is the “Kimoto OptSaver L-57”, which has better angular consistency but similar technical parameters.

Figure 13:
Angular Dependence of the Kimoto OptSaver L-57 for Spectral Scanning

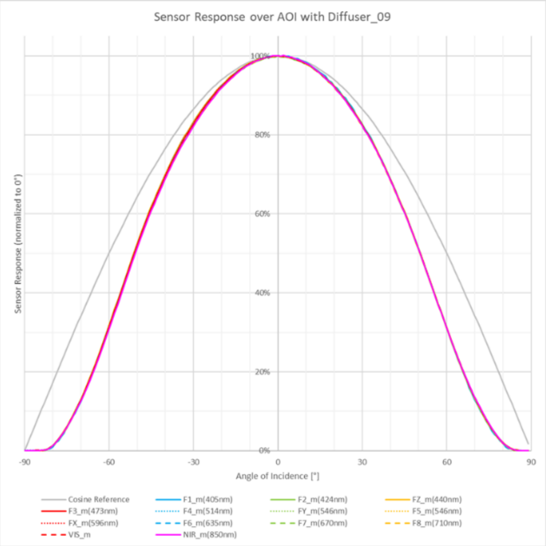
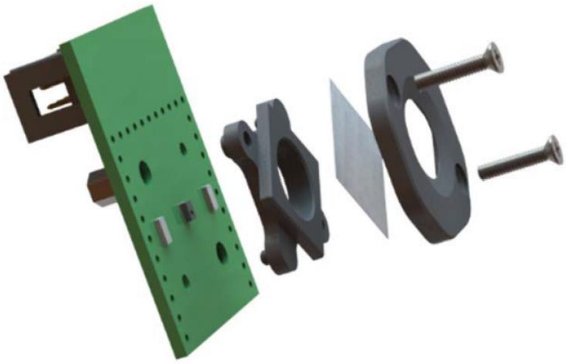


Figure 14:
Installation of the Diffusor Holder ALS



Information

The diffuser specification depends on the customer’s application. Therefore, check the technical parameters of the standard diffuser for technical details before you start any tests. Changing the diffuser is possible. Be careful not to touch the diffusers with your fingers in case of any mounting activities. The surface of the diffusers is sensitive, and any touch, mechanical stress, or dirt can dramatically change the optical behavior. Changing the diffuser and re-assembling typically changes the calibration parameters and requires recalibration for optimal results.

For more details, please order [2].

5 Calibration

The spectral characteristic of a sensor varies in the specified range over the series.

Other disruptions can be:

- Variances, deviations, and limitations in technologies, e.g. the sensor, and especially the LED shifts, and differences over time and temperature. Please see the appropriate datasheet for more information;
- Variances in the diffuser material, which will cause a variant in absorption/emission/transmission;
- Minor variations of the sensor's filter peak sensitivity and the shape of the channels, depending on thickness tolerances in the filter deposition (layers thickness of high and low refracting index);
- Variation of the spectral shape due to tolerances of the aperture (device packaging and field of view, FOV, of the system).

The sum of all errors add up and lead to inaccuracies and considerable differences from device to device. Alternatively, devices should have high accuracy, comparability, and repeatability over a long period - independent of changing conditions. Therefore, a calibration is necessary - which considers all existing conditions and deviations. A high accuracy measurement needs calibration to increase the spectral performance, repeatability, and comparability of the sensor.

A spectral sensor calibration often consists of two phases:

1. In the first step, the raw sensor data is a corrected sensor data – specifically done to compensate for known offsets or variations and drifts during measurement. These can be, for example, corrections of temperature fluctuations, optical and/or electronic offsets (Equation 1 and Figure 15 show an offset compensation), and constant deviations based on the sensor setups (Equation 2 and Figure 16 show the calculation with a correction vector). The result of this step represents the basis for the actual sensor system calibration, which considers the compensation of the device fluctuations.

Equation 1:

$$\begin{pmatrix} ChR\ F1(t) \\ ChR\ F2(t) \\ ChR\ F3(t) \\ \vdots \\ ChR\ Fn(t) \end{pmatrix} = \begin{pmatrix} Ch\ F1(t) \\ Ch\ F2(t) \\ Ch\ F3(t) \\ \vdots \\ Ch\ Fn(t) \end{pmatrix} + \begin{pmatrix} Off\ F1 \\ Off\ F2 \\ Off\ F3 \\ \vdots \\ Off\ Fn \end{pmatrix}$$

Figure 15:
Offset Compensation in Init_file.txt

```
// Offset decreases basic values, valid for a golden device, replace it with the following values:

Offset=0.003101;0.004636;0.005484;0.006407;0.007179;0.007330;0.009379;0.012889
```

2. In the second step, for the best performance, and as an additional step, a device-specific calibration of the sensor system is recommended - which considers all optical effects and other deviations in the specific complete system.

Other calibration methods are possible, such as using a golden device as a calibration standard for all sensors of a series (LOT or product specific Type calibration). This method requires less effort but is usually less accurate because the exemplary deviation per device is not considered.

There is no general solution for the best calibrations. A calibration must always be adapted to the target application and its required accuracy under specified conditions. Effort and accuracy must be balanced - to achieve the required device specification. Chapter 5.1 below highlights a few examples.

5.1 Calibration Examples

The calibration of the system depends on the application, but it will increase accuracy and the uniformity of the sensor in the environment. The uniformity of the sensor is an identical sensor in the application with compensation for all production or system and ambient deviations.

For example, the sensor measures the physical quantity via the references of the calibration, and the sensor results directly represent the concentration of a substance in percent (e.g. NTU as a typical unit for turbidity measurement). This requires a formula or reference where the sensor values[3] can convert directly into the concentration in percent. The response of the spectral sensors consists of multiple numerical (spectrally selective) values (without the flicker channel but using e.g. n=10 … CLEAR and NIR in calibration) or eight numerical values (only using n = 8 filters in VIS) of the sensor channels. Due to manufacturing and based on all deviations in the system, the sensitivity of each channel can vary spectrally or absolutely.

A simple method to compensate for these deviations is a simple correction of each channel (Ch) with a correction factor (Cor) per filter. The software compares selected target reference values with the measured sensor value for all channels. Then, the correction factors become the deviation from the sensor to the target. A reference device (spectrometer, Golden Device, or others) can measure the reference values. This is the method of one-point calibration (e.g. white or spectral balance), which is used in the Evaluation Kit as a correction factor in the “Init_file” (see chapter 6.2).

Equation 2:

$$\begin{pmatrix} ChR\ F1(t) \\ ChR\ F2(t) \\ ChR\ F3(t) \\ \vdots \\ ChR\ Fn(t) \end{pmatrix} = \begin{pmatrix} Cor\ F1 \\ CorF2 \\ CorF3 \\ \vdots \\ Cor\ Fn \end{pmatrix} * \begin{pmatrix} Ch\ F1(t) \\ Ch\ F2(t) \\ Ch\ F3(t) \\ \vdots \\ Ch\ Fn(t) \end{pmatrix}$$

[3] Sensor values are the sensor results from the measurement process, in counts, calculated (corrected) counts, or digits.

Figure 16:
Correction Factors in Init_file.txt

```
// Correction factor of Raw values:

CorrectionFactor=0.966685671,0.982922517,1.016964706,1.027684041,1.037
```

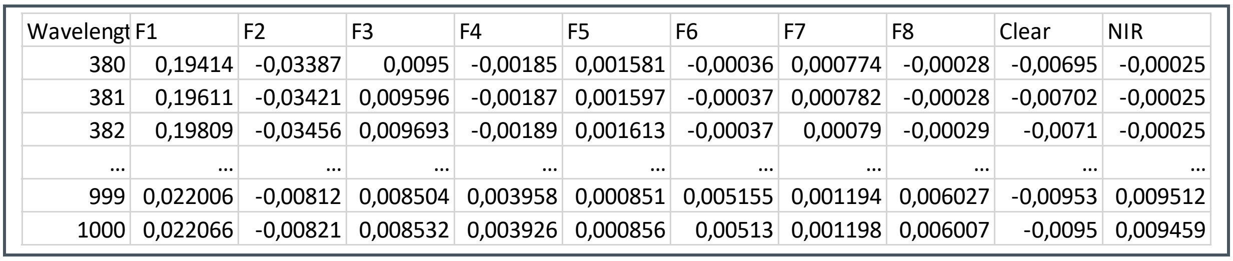
For a spectral reconstruction, a correction matrix is required, which converts the sensor signals with all the filters (e.g. 8 x VIS and NIR + CLEAR => 10) into a reconstructed spectrum according to the step width (n) of the targets.

Equation 3:

$$\begin{pmatrix} S380nm \\ \vdots \\ S1000nm \end{pmatrix} = \mathrm{CM}\begin{pmatrix} Cor\ 380nm\ F1 \ldots .\ Cor\ 380nm\ Fn \\ \vdots \\ Corn\ 1000nm\ F1\ \ldots Cor\ 1000nm\ Fn \end{pmatrix} * \begin{pmatrix} Ch\ F1(t) \\ \vdots \\ Ch\ Fn(t) \end{pmatrix}$$

The accuracy of transformation and spectral reconstruction depends on the quality of the coefficients of the Correction Matrix (CM) - which the calibration must calculate. This method can be used in the GUI by defining a calibration matrix (e.g. CM for ALS Light Detection – Light Detection Calibration Matrix File = CM_L1_v1_0_0.csv, see chapter 6.2).

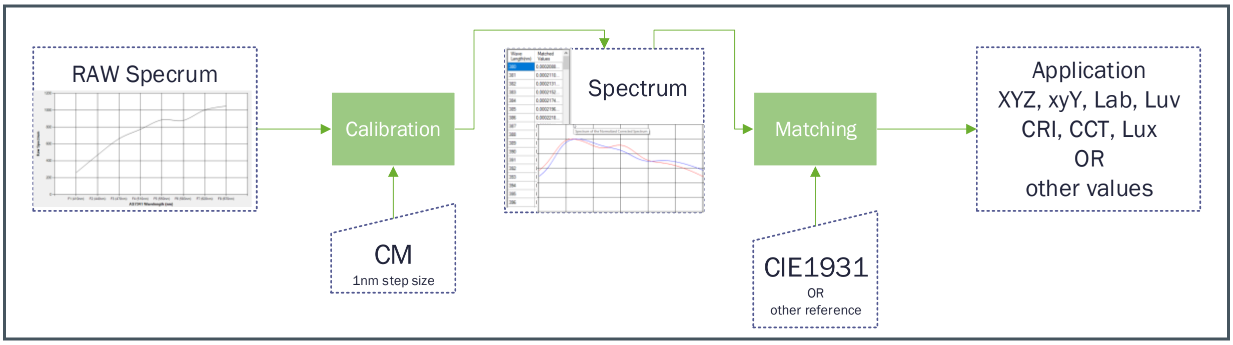
Figure 17:
Calibration Matrix for Light Detection



Wavelengt	F1	F2	F3	F4	F5	F6	F7	F8	Clear	NIR
380	0,19414	-0,03387	0,0095	-0,00185	0,001581	-0,00036	0,000774	-0,00028	-0,00695	-0,00025
381	0,19611	-0,03421	0,009596	-0,00187	0,001597	-0,00037	0,000782	-0,00028	-0,00702	-0,00025
382	0,19809	-0,03456	0,009693	-0,00189	0,001613	-0,00037	0,00079	-0,00029	-0,0071	-0,00025
...	...	...	...	...	...	...	...	...	...	...
999	0,022006	-0,00812	0,008504	0,003958	0,000851	0,005155	0,001194	0,006027	-0,00953	0,009512
1000	0,022066	-0,00821	0,008532	0,003926	0,000856	0,00513	0,001198	0,006007	-0,0095	0,009459

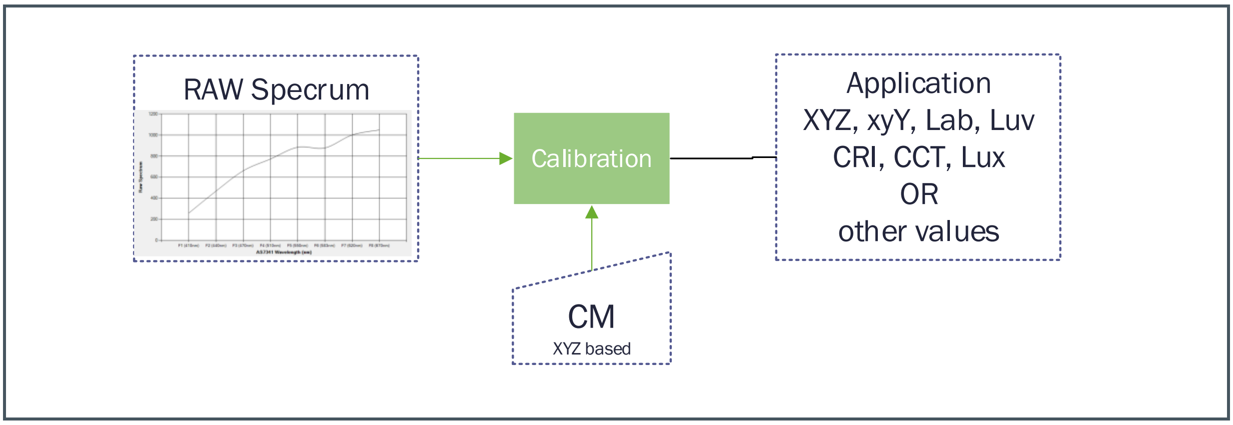
The reconstructed sensor spectrum must be transformed into the values of the CIE1931 color space or another application-specific unit. This can easily be done by overlapping (multiplying) the spectrum and the standard CIE1931 XYZ observer function. The result of this overlapping is the specific XYZ values for the reconstructed spectrum. This allows you to directly calculate xyz, xyY, CCT, and other typical colorimetric results of CIE1931.

Figure 18:
Example: Transfer of a Reconstructed Spectrum Directly Into the CIE1931 Color Space



Alternatively, the correction matrix can directly contain the calibration data (here rows) for X, Y, and Z. Then, calibration is performed directly via the CIE1931 XYZ target transformation. Such calibrations can be done with global and/or local targets ([6]).

Figure 19:
Example: Transfer of a Reconstructed Spectrum Directly Into the CIE1931 Color Space



For example, if the sensor measures the concentration of a substance or liquid, the filter function(s) or matrix must provide the corresponding data in the software for conversion (e.g. percentage for substances or NTU for turbidity). These data must be available before matching the sample measurements during development (spectral fingerprint) or as reference data from commercially available standards. Here, the result is often one or more mathematical functions or lookup tables that convert the digits measured by the sensor directly into a physical unit (sensor matching or sensor data conversion), whereby the tolerances are also compensated by sensor-specific correction variables (sensor uniformity).

The GUI for the Evaluation Kit supports the calibration methods: Scaling, Global Matrixing, and Local Matrixing[4]. The control takes place via the input files for calibration. The GUI uses the correction

[4] Limited algorithm with twelve targets as the demo.

matrix (lines) to recognize the calibration method, step size, and other parameters specified by the user and considers this in further calculations and software outputs (see also chapter 6.2.2 and [6]).

The GUI setup installs some files for calibration as examples. These installed calibration data are NOT the result of a generally accepted device calibration with the highest accuracy for each device. Using an evaluation kit under conditions similar to those in the calibration, can only achieve a maximum accuracy like a type or batch calibration. Type or batch calibration is always performed by calibrating a golden device (=typical sensor), which represents a model for several sensors (batch, type) for calibration. Its calibration data are valid for all sensors of this type/batch without paying attention to exemplary scattering and differences.

A device calibration considers all device-specific realities and deviations and requires an individual calibration process. It is always the target to achieve the highest accuracy. Therefore, use the pre-installed database for type calibration and other spectral-sensitive functions (Spectral Mask Compare, color recognition) only to see the form and example of these data and, if necessary, replace them with your application-specific database for device calibration.

Initialization files ("*Init_file.txt*") control the database for the calibration from alternative files for light detection and reflection mode. For the standard, the correction vectors or other parameters can be defined by default as "1" - no correction of the raw values occurs. In this case, use such parameters to adapt to the sensor functions. The file for the lighting of the correction matrix contains a universal standard calibration for the transformation of the raw sensor values in reconstructed (+corrected) spectra for standard light sources. The values are calculated based on the design or measured data of typical filter curves and do not consider any device-specific scattering and deviations in series. The standard calibration can be used for limited demonstrations but replaced with alternative methods for higher, accurate measurement and control.

Alternative methods include target calibrations for applications with constant targets and conditions. Another method involves a device calibration on the spectral filters, acting as a golden device calibration, with a subsequent adjustment of the sensors in series to the golden device. The methods are manifold and strongly dependent on the application, requirements, and system conditions. The result is affected negatively, for example, by step overlaps between light sources and filters, strong infrared components of the light sources, or spectral components where the sensor has only low or no sensitivity.

Figure 20 shows the reconstructed spectrum of an identical standard light source, INC A, compared to the spectrometer and sensor results - where the raw sensor results were calibrated using the general calibration matrix and/or a (golden) device matrix. In this example, the result of the device calibration is better by a factor of five (see also [6]).

Figure 20 :
Results of Alternative Correction Matrices (General on the left, Device Calibration on the right)

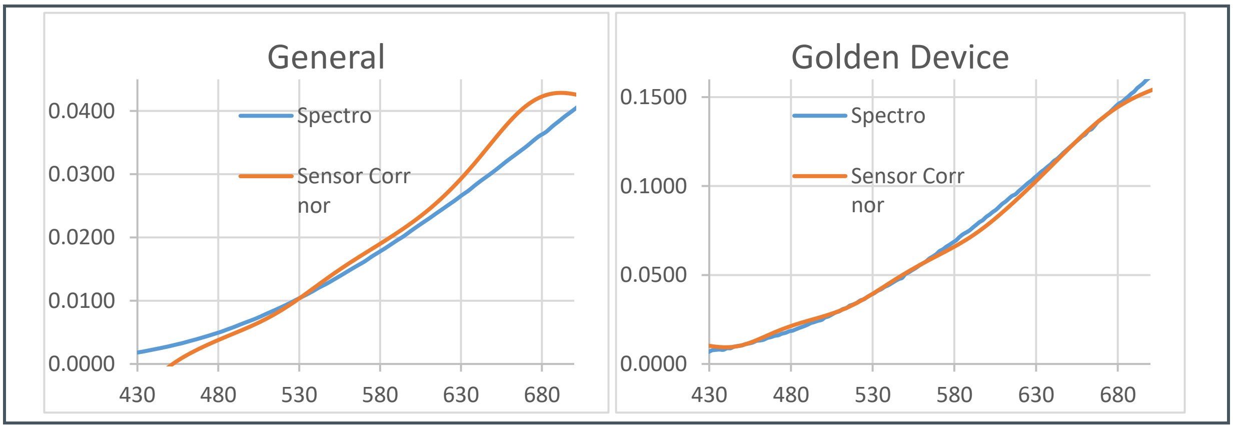


Figure 21 and Figure 22 show alternative processes of used calibration methods, which customers can use by definition in the initial files.

Figure 21:
From Spectral RAW Values to Recalculated Spectrum to CIE1931 XYZ Results

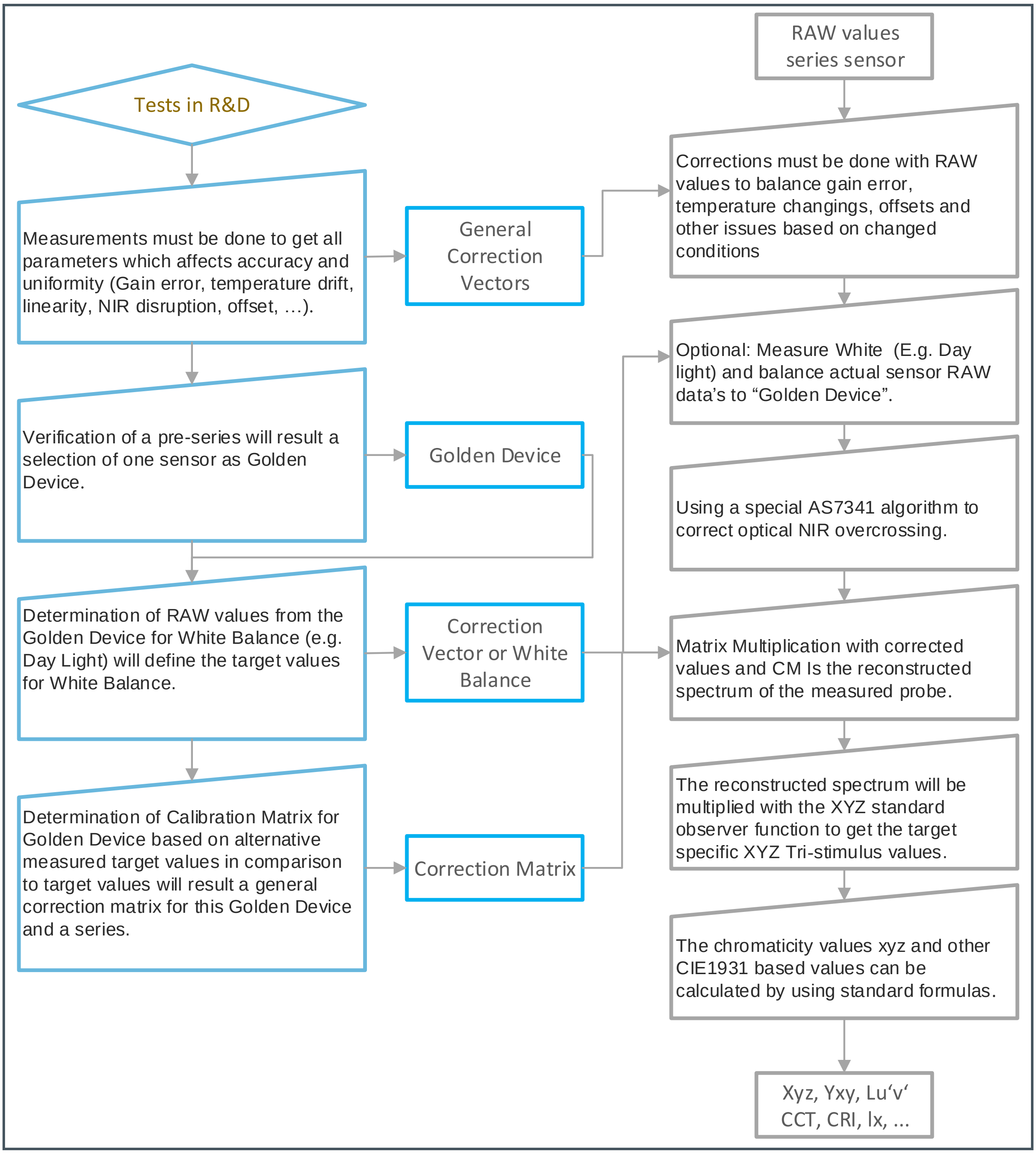
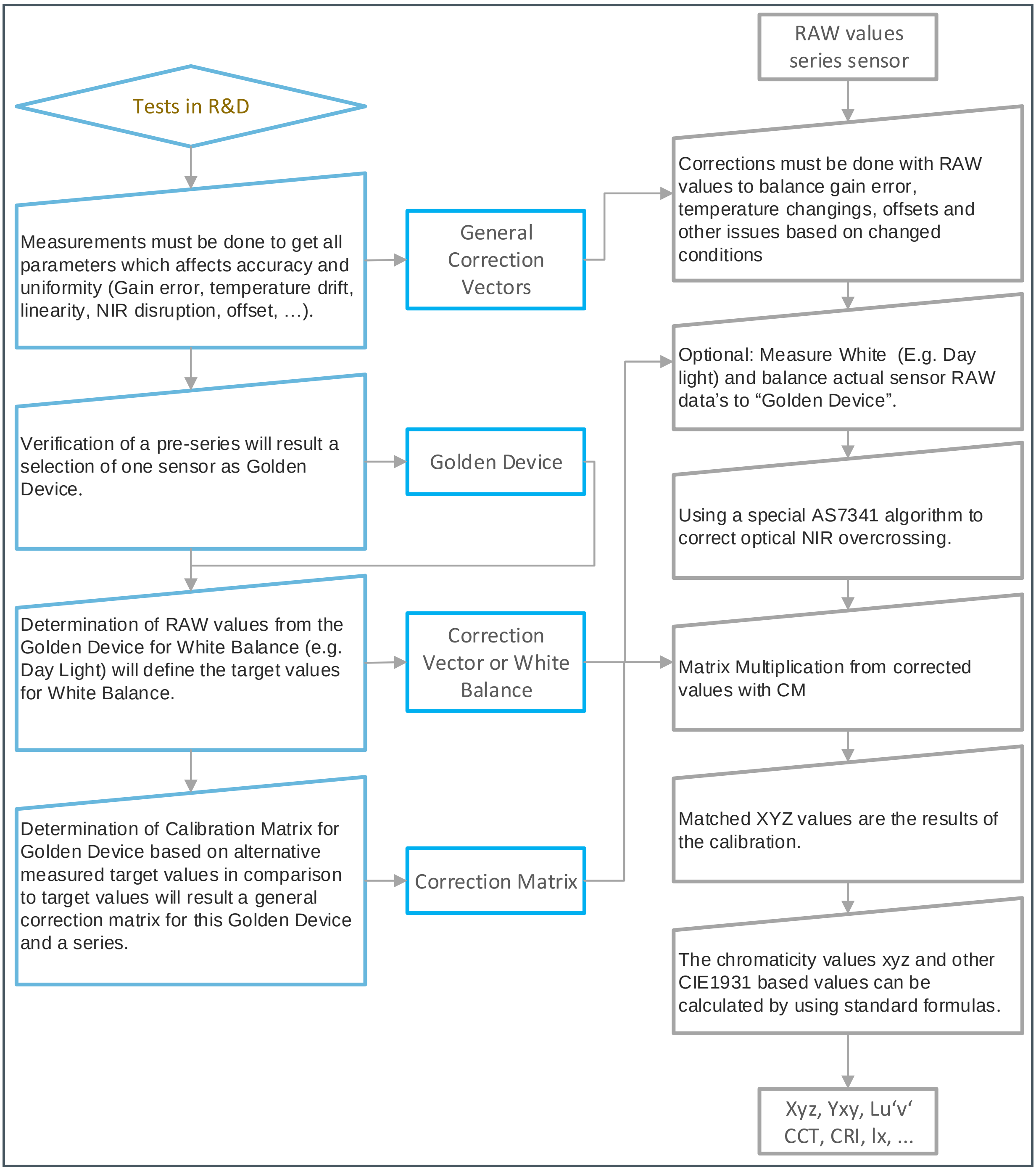

RAW values series sensor
Tests in R&D
Corrections must be done with RAW values to balance gain error, temperature changings, offsets and other issues based on changed conditions
Measurements must be done to get all parameters which affects accuracy and uniformity (Gain error, temperature drift, linearity, NIR disruption, offset, …).
General Correction Vectors
Optional: Measure White (E.g. Day light) and balance actual sensor RAW data's to "Golden Device".
Verification of a pre-series will result a selection of one sensor as Golden Device.
Golden Device
Using a special AS7341 algorithm to correct optical NIR overcrossing.
Determination of RAW values from the Golden Device for White Balance (e.g. Day Light) will define the target values for White Balance.
Correction Vector or White Balance
Matrix Multiplication with corrected values and CM Is the reconstructed spectrum of the measured probe.
The reconstructed spectrum will be multiplied with the XYZ standard observer function to get the target specific XYZ Tri-stimulus values.
Determination of Calibration Matrix for Golden Device based on alternative measured target values in comparison to target values will result a general correction matrix for this Golden Device and a series.
Correction Matrix
The chromaticity values xyz and other CIE1931 based values can be calculated by using standard formulas.
Xyz, Yxy, Lu'v'
CCT, CRI, lx, …

Figure 22:
From Spectral RAW Values to CIE1931 XYZ

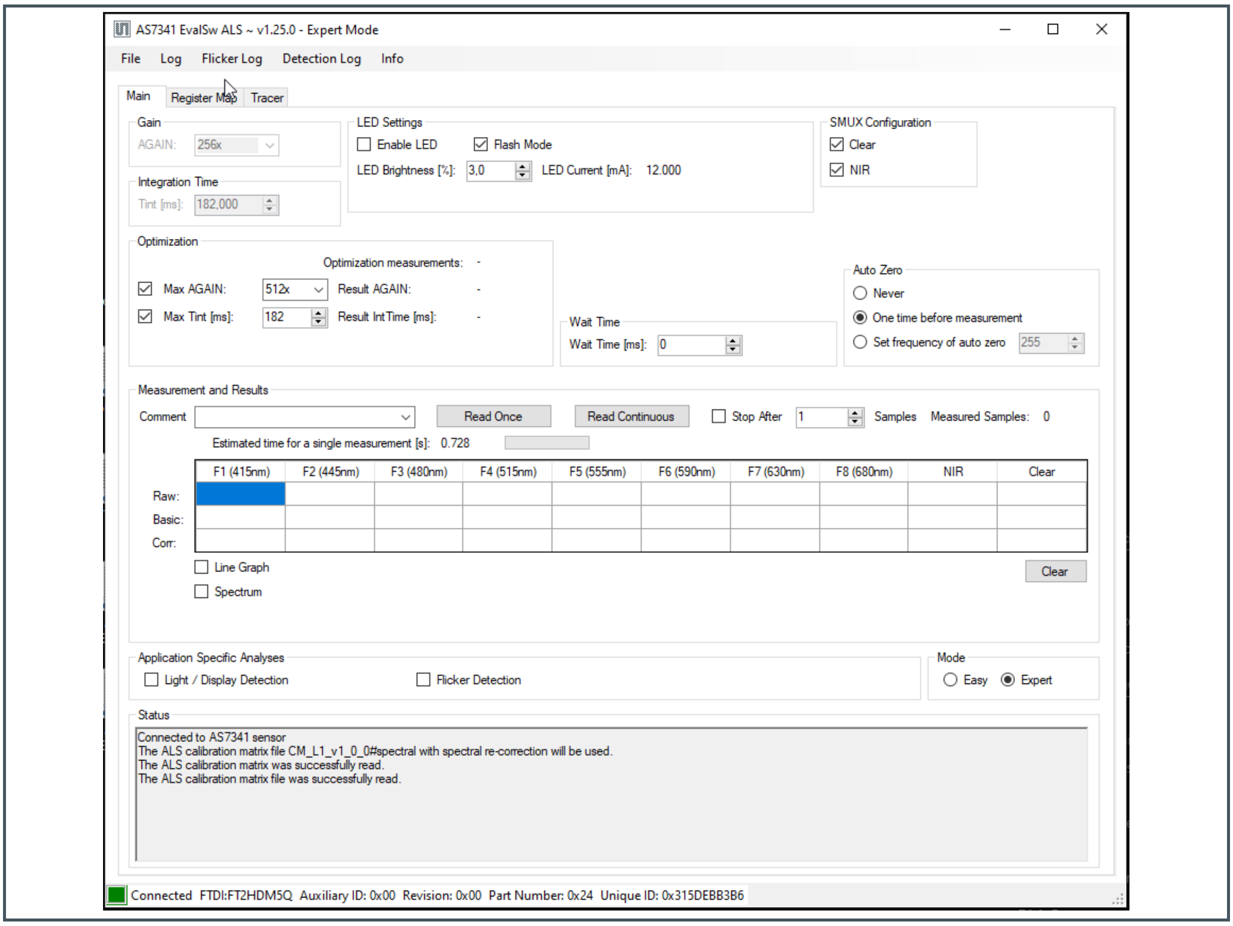

Tests in R&D
Measurements must be done to get all parameters which affects accuracy and uniformity (Gain error, temperature drift, linearity, NIR disruption, offset, …).
General Correction Vectors
Verification of a pre-series will result a selection of one sensor as Golden Device.
Golden Device
Determination of RAW values from the Golden Device for White Balance (e.g. Day Light) will define the target values for White Balance.
Correction Vector or White Balance
Determination of Calibration Matrix for Golden Device based on alternative measured target values in comparison to target values will result a general correction matrix for this Golden Device and a series.
Correction Matrix
RAW values series sensor
Corrections must be done with RAW values to balance gain error, temperature changings, offsets and other issues based on changed conditions
Optional: Measure White (E.g. Day light) and balance actual sensor RAW data's to "Golden Device".
Using a special AS7341 algorithm to correct optical NIR overcrossing.
Matrix Multiplication from corrected values with CM
Matched XYZ values are the results of the calibration.
The chromaticity values xyz and other CIE1931 based values can be calculated by using standard formulas.
Xyz, Yxy, Lu'v'
CCT, CRI, lx, ...

6 Software

6.1 Software Installation

The GUI is compatible with Windows 10 - where .NET Framework 4.5.2 or a later version is pre-installed. It is mandatory to install the 32-bit Visual C++ redistributable for visual studio 2015 runtime. The "*vc_redist.x86.exe*" setup is included with the evaluation kit. Before connecting the FTDI adapter to the computer, the specified FTDI driver must be installed[5]. Start the "*CDM21216_Setup.exe*" and follow the step-by-step instructions. Then, connect the EVK to the computer via the FTDI adapter.

Figure 23:
Typical Main GUI Window with Expert Mode and Status Bar with Connected Sensor Hardware



Gain
AGAIN: 256x

Integration Time
Tint [ms]: 182,000

LED Settings
□ Enable LED ☑ Flash Mode
LED Brightness [%]: 3,0 LED Current [mA]: 12.000

SMUX Configuration
☑ Clear
☑ NIR

Optimization
Optimization measurements: -
☑ Max AGAIN: 512x Result AGAIN: -
☑ Max Tint [ms]: 182 Result IntTime [ms]: -

Wait Time
Wait Time [ms]: 0

Auto Zero
○ Never
● One time before measurement
○ Set frequency of auto zero 255

Measurement and Results
Comment
□ Stop After 1 Samples Measured Samples: 0
Estimated time for a single measurement [s]: 0.728

	F1 (415nm)	F2 (445nm)	F3 (480nm)	F4 (515nm)	F5 (555nm)	F6 (590nm)	F7 (630nm)	F8 (680nm)	NIR	Clear
Raw:										
Basic:										
Corr:										

□ Line Graph
□ Spectrum

Application Specific Analyses
□ Light / Display Detection □ Flicker Detection

Mode
○ Easy ● Expert

Status
Connected to AS7341 sensor
The ALS calibration matrix file CM_L1_v1_0_0#spectral with spectral re-correction will be used.
The ALS calibration matrix was successfully read.
The ALS calibration matrix file was successfully read.

🟩 Connected FTDI:FT2HDM5Q Auxiliary ID: 0x00 Revision: 0x00 Part Number: 0x24 Unique ID: 0x315DEBB3B6

[5] Newer Windows 10 installations do not need an extra FTDI driver installation because they are useable with the standard Windows drivers. On the contrary, sometimes an old FTDI driver interferes with the system and has to be uninstalled with special software to use the Windows driver - https://ftdichip.com/utilities/#cdm-uninstaller

Start the software installation from the USB stick “*.msi” and then start the GUI directly from the program directory: “*C:\Program Files (x86)\ams-OSRAM\AS734x Demo*”. Afterward, it should show the main window and the successful connection to the EVK in the status bar and footer line (see Figure 23).

The GUI requires some initialization files, which will be installed during the setup in the user directory: “*C:\Users\user\AppData\Roaming\ams-OSRAM\AS734x Demo*”.

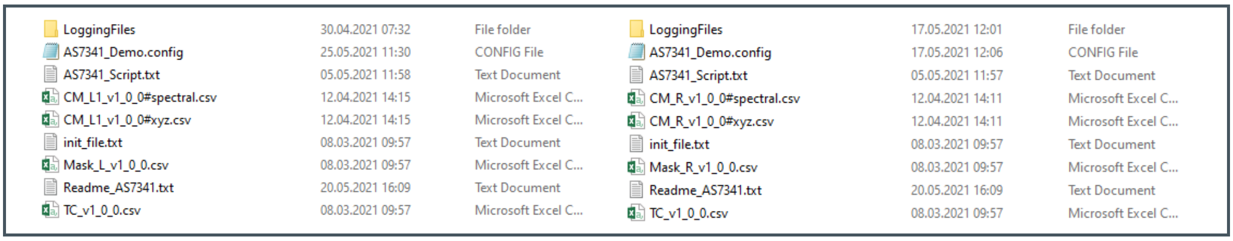
During installation, the standard initialization files are generated in the UserData folder (*C:\Users\xxx\AppData\Roaming\ams-OSRAM\AS734x Demo*). The GUI continuously checks if these standard files still exist in this area. If not, the GUI will install the missing files again.

Consider saving the previously used initialization files if needed. For more details, see chapter 6.2.

Attention

A de-installation process will also delete the standard initialization files. Therefore, create a backup or copy of such files if you want to use them after de-installation or use alternative file names for the initialization files.

Figure 24:
Typical User Directory with Initialization Files (ALS on the left, Reflection Mode on the right)



LoggingFiles	30.04.2021 07:32	File folder	LoggingFiles	17.05.2021 12:01	File folder
AS7341_Demo.config	25.05.2021 11:30	CONFIG File	AS7341_Demo.config	17.05.2021 12:06	CONFIG File
AS7341_Script.txt	05.05.2021 11:58	Text Document	AS7341_Script.txt	05.05.2021 11:57	Text Document
CM_L1_v1_0_0#spectral.csv	12.04.2021 14:15	Microsoft Excel C...	CM_R_v1_0_0#spectral.csv	12.04.2021 14:11	Microsoft Excel C...
CM_L1_v1_0_0#xyz.csv	12.04.2021 14:15	Microsoft Excel C...	CM_R_v1_0_0#xyz.csv	12.04.2021 14:11	Microsoft Excel C...
init_file.txt	08.03.2021 09:57	Text Document	init_file.txt	08.03.2021 09:57	Text Document
Mask_L_v1_0_0.csv	08.03.2021 09:57	Microsoft Excel C...	Mask_R_v1_0_0.csv	08.03.2021 09:57	Microsoft Excel C...
Readme_AS7341.txt	20.05.2021 16:09	Text Document	Readme_AS7341.txt	20.05.2021 16:09	Text Document
TC_v1_0_0.csv	08.03.2021 09:57	Microsoft Excel C...	TC_v1_0_0.csv	08.03.2021 09:57	Microsoft Excel C...

6.2 Initialization Files

The initialization files (see Figure 24) supply important parameters for the GUI. Customers can adapt these files but note that changes of any specified parameters in these files can dramatically affect the sensor results and accuracy. It is recommended to create a backup file of the initialization files before making any changes.

The initialization files[6] are saved on the hard disk in the directory “*C:\Users\xxx\AppData\Roaming\ams-OSRAM\AS734x Demo*[7]” after the installation of the GUI. In the

[6] Change the AppData directory properties only if the directory or subdirectories are not visible or read-only.
[7] Replace XXX with a username.

case of an AS734x demo with an onboard EEPROM (e.g. AS7341 Pen), the parameters from all the initialization files are located in the EEPROM and are loaded from there.

- "*Init_file.txt*" - necessary for all the GUI functions. It includes values to control functions and limitations, e.g. sensor signal correction, light detection.
- "*CM_x1_v1_0_0.csv*" – Correction matrices for x=L (Light measurement) and x=R (Reflection mode).
- "*Mask_x_v1_0_0.csv*" – Database for x=L, Spectral Mask Compare (Light measurement), and x=R, Color Recognition (Reflection mode).

Comments are not considered by the GUI and can be done as any comment in all initialization files with the characters '//' at the beginning of a line.

All initialization files must be checked/adapted/changed before the GUI starts running. The GUI will not consider adapted parameters in these files by the actively running GUI. It will not change the initialization files or correction matrix files but adapt or create mask files in the case this function is active, and a new mask is created and saved. Check your adaptations of these files if the GUI prints out a syntax error. The syntax and format of parameters are fixed and depend on the MS Windows region on the computer used.

6.2.1 Init_file.txt

"Init_file.txt" can include the following commands and parameters.

- **xxxxxxFile: xxxxxx.csv:** These are the standard or user-defined files for calibration (see chapter 6.2.2), spectral compare, and other functions, which are used by the GUI as input files (in the case of no EEPROM hardware version). Otherwise, the GUI will generate a file selection dialog box to select these files after choosing the application Light or Reflection mode.
- **ChannelOrder:** Defines the order of the filters as they are used in subsequent blocks (e.g. Offset, Correction and CM).
- **Offset:** Use this to correct constant issues in the sensor setup for tests. The specified values for the single channels F1 to F8, Clear, and NIR in the Init_file, will be added/subtracted from Basic_Count after gain correction. The result will be the corrected count. Therefore, offset must be specified in the form of gain-corrected Basic_Counts (see Figure 39). Use the offset correction, for example, to compensate for ambient light, optical overcrossing, or other constant deviations. The offset in the Init_file refers only to all functions in the basic window. The offset for ALS and Reflection Mode is defined separately in the calibration data files. If not mentioned in the calibration files, offset will be used from the Init_file or can be measured and set by CTRL+O keyboard combination[8]. When offset values are not defined in any of the files, the default value, zero, is considered for each channel. The order of the filters is defined in the "ChannelOrder".

[8] In this case, the latest measured sensor results will be used as the offset. It will temporarily overwrite all other defined offset values until the GUI is stopped, and the process asks to save the new offset in the init_file or EEPROM.

- **CorrectionFactor:** Is the scale sensor response by multiplication of the single Correction Factors (=balancing). The specified factors for the single channels F1 to F8, Clear, and NIR in the Init_file, will be multiplied (see Figure 39). Use this correction factor for customized calibration (see chapter 5), e.g. to compensate for the effects of diffusers or other balance operations. The correction factor in the Init_file only refers to the functions in the basic window. The correction factors for ALS and Reflection Mode are defined separately in the calibration data files. If not mentioned in these files, it will choose from the Init_file. When the correction factors are not defined in any of the files, the default value “1” is taken for each channel. The order of the filters is defined in “ChannelOrder”.
- **CorrectionGain:** Use this to correct the GainError in all the GUI functions. The given numbers in the installed initialization file are from the datasheet. Customers should verify them and make an individual gain correction in the case of the highest accuracy requirements. When the correction parameters for gain are not defined in the Init_file, the default value “1” is taken for each channel.
- **MaxAutogain:** Use this to define a maximum for gain in the automatic setup optimization mode. The specified value will be used as an initialization value but can be changed in the GUI.
- **MaxAutoTINT:** Use this to define a maximum for TINT in the automatic setup optimization mode. The specified value will be used as an initialization value but can be changed in the GUI.
- **NIR_Correction:** Activate the Dynamic NIR compensation algorithm for light detection and Spectral Mask Compare – must be used for Light Detection but not Reflection mode. If not declared in the file, the default value, “ON”, is considered from the main window.
- **Corr_lx:** Conversion factor from Y (based on the calculated XYZ from the corrected spectrum) to Y(lx). This factor only affects the output of the results for Y(lx) in light detection. If not stated in the initialization file, the default value is “683”.
- **Limit_mask:** Will limit the wavelengths for the output diagram and Spectral Mask Compare. The lower and upper limits of the wavelength are considered. By default, the value “381” is taken as the lower limit, and “781” is taken as the upper limit.
- **Limit_Delta_xy:** XYZ masks with a higher “Delta_xy” than defined here will not be recognized.
- **Limit_Compare:** Spectral masks with a higher deviation in percent than defined here will not be recognized.
- **Delta_uv/LowerLimit_u/UpperLimit_u/UpperLimit_v/minCCT/maxCCT:** Specifies the parameters, which are the conditions and limitations for CCT calculations in light detection. These values directly affect the CCT calculation. When not mentioned in the Init_file, the following default values are taken:
  Deltauv = 0.2000, LowerUlimit = 0.1807, UpperUlimit = 0.3988, LowerVlimit = 0.3624, UpperVlimit = 0.5408, MinCCT = 2000, MaxCCT = 7000

Figure 25:
Example of INIT_FILE.TXT

```
LightDetectionCalibrationMatrixFile= CM_L1_v1_0_0.csv

LightDetectionMaskFile= Mask_L_v1_0_0.csv

ReflectionCalibrationMatrixFile= CM_R_v1_0_0#xyz.csv

ReflectionMaskFile= Mask_R_v1_0_0.csv
```

```
Offset= 0.00283;0.00427;0.00514;0.00595;0.00660;0.00673;0.00857;0.01193;0.02851;0.002491

CorrectionFactor= 1.0000;1.0000;1.0000;1.0000;1.0000;1.0000;1.0000;1.0000;1.0000;1.0000

CorrectionGain= 1.0388;1.0289;1.0288;1.0327;1.0037;1.0;0.9955;0.98640;0.9873;0.97350;0.9461

NIR_Correction= On

corr_lx= 683

LowerLimit_u= 0.18070000

LowerLimit_v= 0.39880000

UpperLimit_u= 0.36240000

UpperLimit_v= 0.54300000

minCCT= 1000

maxCCT= 8000

Delta_uv= 0.20000000

limit_mask= 380;780

limit_Compare= 50

Limit_Delta_xy= 1.00

MaxAutoGain= 10

MaxAutoTInt= 500
```

6.2.2 Calibration_matrix_file

The calibration matrices are CSV files, which include correction and calibration data for application-specific analysis – ALS Light Detection and/or Reflection mode:

- **LEDCurrent:** Only for Reflection mode. These values will be used for actual measurements.
- **LEDxxx:** Toggle LED on/off (xxx=[Left, Top, Bottom]).
- **ChannelOrder:** Defines the order of the filters as they are used in subsequent blocks (e.g. Offset, Correction and CM).
- **Offset:** See chapter 6.2.1.
- **WaitTime:** This is defined in the GUI as the waiting time between two measurements in ms and possible steps in 10 ms increments. The function does not use the WTIME (LONG) function of the sensor chip.
- **CorrectionGain:** See chapter 6.2.1.
- **CorrectionFactor:** See chapter 6.2.1.

- **ReferenceWhiteBalance:** If a white balance is made in the GUI with the command “Ctrl+B”, the GUI uses the XYZ comparison values to calculate the white balance. The order of the filters is defined in the “ChannelOrder”.
- **CorrectionWhiteBalance:** If available, these values are generally applied as the white balance and correction vector for XYZ.
- **ConversionXYZ2RGB:** Here, a general 3x3 matrix is specified for the conversion of the measured XYZ into RGB colors for the screen print in Reflection Mode.
- **Calibration matrix:** Will be used for the calibration process by matrix multiplication. A special syntax in the calibration file is used to define the calibration matrix (see Figure 26 or later in this chapter). The order of the filters is defined in the “ChannelOrder”.
- **NIR_Correction:** See chapter 6.2.1.

Information

Here, defined parameters replace the specified values from “Init_file.txt” (see chapter 6.2.2).

The calibration matrices are part of the software setup and represent general solutions for the demonstration of the Evaluation Kits. They are based on the specified AS734x filter definitions, components on the evaluation kits, and the existing conditions during calibration. They do not consider any real existing series deviations. Therefore, deviations in the sensor results compared with the spectrometers are possible[9].

In spectral matrices, the maximum and minimum of the used wavelengths and step sizes determine the dynamic (min and max) and spectral steps of the reconstructed spectrum. In general, the calibration matrix determines the dimension of the corrected vector values after matriculation according to the mathematical rules for matrix multiplication.

Figure 26 shows a typical correction matrix file using a global target (created here for the XYZ target calibration with the offset) correction, LED current settings, and prepared correction vector.

The Calibration matrices for a global target can be based on:

- Simple balance algorithms using the diagonal axis of CM[10:10][10];
  Rows = 10 x step sizes in nm; Columns = 10 filters of AS7341;
- Direct XYZ transformation using a CM[3:8]
  Rows = X,Y,Z; Columns = 8 VIS filters of AS7341;
- Spectral Reconstruction using a CM[621:10]
  Rows = 621 steps between 380 nm and 1000 nm; Columns = 10 filters of the AS7341;

[9] Calibration, effort, and accuracy, always depend on customer-specific requirements. Use alternative calibration methods to increase accuracy in general. A device-to-target calibration, where sensor and reference values are calculated to get the device-specific calibration matrix, will achieve the highest accuracy. A good compromise between effort and accuracy is to use the general calculation matrix with a peak adjustment to a Golden device in advance.

[10] Identical by using Correction Factor= in “Init_file.txt”.

Depending on the CM matrix, the algorithm for calibration converts the Basic_Counts (sensor results from F1 to F8 + NIR/Clear) directly into the corrected wavelength, XYZ coordinates, or to an interpolated spectrum.

Figure 26:
XYZ Calibration Matrix File – Example of a Target Calibration in Reflection Mode

```
LEDCurrent=9

Offset=0.002444;0.007075;0.003559;0.010774;0.012211;0.014322;0.010946;0.008662

CorrectionFactor=1;1;1;1;1;1;1;1;1;1

NIR_Correction=ON

WaitTime=500

Wavelength; F1;F2;F3;F4;F5;F6;F7;F8

X;-370.02735;65.107441;59.96563;-4.2633151;9.487805;21.903770;-12.466643;53.561871

Y;-383.18132;53.29510;83.532073;8.874448;16.704869;15.15450;-19.77170;52.909989

Z;-480.1549;148.72844;91.762775;-0.041612;-2.8765691;12.630754;-33.4291;67.516507
```

Figure 27:
Spectral Calibration Matrix File – Example of a Target Calibration in Reflection Mode

```
Offset=0.003101;0.004636;0.005484;0.006407;0.007179;0.007330;0.009379;0.012889

CorrectionGain= 1.0388;1.0289;1.0288;1.0327;1.00370;1.0;0.9955;0.9864;0.98730;0.9735;0.9461

NIR_Correction= OFF

CorrectionWhiteBalance=0.426443643;0.9408319733;0.9352714227

LedCurrent= 38

ConversionXYZ2RGB= 10.03558;-6.165361;-0.542154;1.369359;1.005419;0.57871;1.154142;-
0.590247;2.149525

Wavelength;F1;F2;F3;F4;F5;F6;F7;F8

400;4.036772643;-0.818815514;-0.144452398;0.509614933;-0.718516813;0.381627226;-
0.471626464;0.397308931

410;4.243968918;-0.432167107;-0.091670116;0.192294829;-0.344006358;0.146885619;-
0.236176536;0.125157765
```

```
700;-1.962472313;0.901288867;0.102799117;-0.071629683;-0.036856937;0.223529246;-
0.255486732;1.490185886
```

In the case of a local calibration (the use of a locally delimited target), the GUI needs all the reference and sensor data from the global calibration. Therefore, the calibration file must contain the blocks “TargetBase” and “SensorDataTarget”. Figure 28 shows an example for both in shortened form.

Figure 28:
XYZ Calibration Matrix Data for Local Calibration in Reflection Mode

```
//TargetBase=D65;2°Observer. Spektrometer valuesTarget;Dark Skin;Light Skin;Blue
Sky;Foliage;Blue Flower;Bluish Green;Orange Yellow;Yellow Green;Purple;Moderate
Red;PurplishBlue;Orange;Blue;Green;Red;Yellow;Magenta;Cyan;Black; Neutral 3.5;Neutral
5;Neutral 6.5;Neutral 8;White

X;11.55;38;18.02;10.72;24.65;30.3;45.84;34.38;8.82;28.72;14.08;38.75;8.92;14.84;20.59;57.63
;30.6;14.99;3.22;8.75;18.64;35.26;55.5;88.08
Y;10.57;35.03;19.12;13.4;22.96;41.93;43.05;45.2;6.67;19.27;12.16;30.91;6.92;23.73;12.59;61.
48;19.97;20.51;3.39;9.27;19.66;37.3;58.79;93.18
Z;7.55;25.66;35.01;7.18;43.02;44.77;8.07;11.62;14.92;13.74;40.48;6.35;30.4;10.47;5.22;9.62;
31.66;41.37;3.82;10.28;21.02;40.39;64.36;96.71

//SensorDatas are measured with Golden Device but data's must be replaced by the sensor
specific results (=corrected Basic_Counts) to get high accuracy accuracy

SensorDataTarget;Dark Skin;Light Skin;Blue Sky;Foliage;Blue Flower;Bluish Green;Orange
Yellow;Yellow Green;Purple;Moderate Red;Purplish
Blue;Orange;Blue;Green;Red;Yellow;Magenta;Cyan;Black;Neutral 3.5;Neutral 5;Neutral
6.5;Neutral 8;White

415;0.0164012;0.0491729;0.0413218;0.0148017;0.0606691;0.0534005;0.0405359;0.0321033;0.02490
98;0.0391705;0.0470008;0.0337166;0.0286606;0.0177893;0.0277276;0.0471173;0.0641632;0.036352
7;0.0068549;0.0166559;0.0316264;0.0599908;0.0945920;0.1302734

680;0.1388945;0.5013828;0.1365428;0.1012702;0.3405770;0.1599904;0.5254613;0.2791456;0.12645
54;0.4711969;0.1433661;0.4689053;0.0589107;0.0886307;0.4879801;0.5809885;0.5533715;0.067969
3;0.0323798;0.0782787;0.1533260;0.2947422;0.4815212;0.7270016
```

6.2.3 Mask_files

A spectrum in ALS Light Detection can be used to detect the type of luminary or to calculate CIE1931 xyz standard values, xyz, uv, Y (lux), CCT, and CRI. In reflection mode, the comparison result can be a recognized and/or identified color in the CIE1931 color space. The aim is to identify the measured quantities in a set of defined masks in all applications. Such masks are defined in the file “*Mask_x_v1_0_0_.csv*”. These masks were predefined in a format, which must also be used in the calibration file (XYZ or spectrum). The dimensions of the mask file depend on the corrected sensor values and Calibration Matrix (CM). Changes in the CM make it necessary to adapt the mask file.

The process for spectral mask compare involves a simple algorithm based on the sum of square deviations by comparing each wavelength of the two normalized spectra - the reconstructed spectrum from the last measurement - with a masking spectrum from the mask compare database. Each mask will be checked in its dimensions (wavelengths or Tristimulus values) for deviations from the actual sensor values. The recognized result is the mask with the smallest deviation given in the mask file, or the process will stop without recognition if all the calculated deviations are greater than “limit_Compare” from the Init_file. It is possible to create or adapt the mask file in the GUI. See chapter 6.10. for more details.

The GUI uses the calibration and mask files from the standard installation or selected by the user with a pop-up selection window (exclude EEPROM versions like Pen – see chapter 8). Users can teach or add spectral masks if ALS or Reflection mode is selected. The new spectral mask data are saved into the previously selected mask’s CSV file.

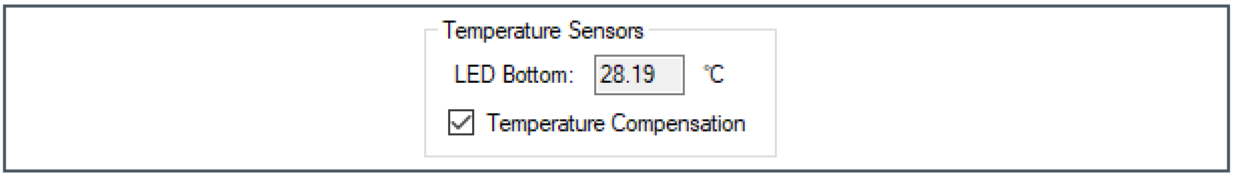
Figure 29:
XYZ Mask Compare File – Example of Target Color Checker in Reflection Mode

```
Title;Dark Skin;Light Skin;Blue Sky;Foliage;Blue Flower;Bluish Green;Orange
Yellow;Purple;Yellow Green;Moderate Red;Purplish
Blue;Orange;Blue;Green;Red;Yellow;Magenta;Cyan;Black;Neutral 3.5;Neutral 5;Neutral
6.5;Neutral 8;White

X;13.10766995;40.23476661;19.76357155;12.32057009;28.11456750;34.46215054;47.36491495;9.414
09355;35.31011303;30.49006183;15.73690724;40.67648446;9.93201455;14.72641231;20.43601873;55
.67261275;30.33960376;15.44604387;5.71390518;13.70273791;28.03028335;52.71169307;83.5900360
6;89.00220431

Y;12.11358191;37.23689769;20.85253670;15.25623321;26.42880375;46.55675887;44.18996120;7.205
15710;46.36512629;20.83666417;13.74792625;33.04739064;7.80960310;24.24115586;12.89418579;59
.87306657;20.12365170;20.57636046;5.99827841;14.49131624;29.52774904;55.81342223;88.2181490
0;94.13973102

Z;8.65162881;27.67085941;37.52982863;8.67690824;48.40070988;49.77952809;9.24912269;15.00693
966;12.89679721;15.17793351;42.53656128;8.88358254;32.10168805;10.73278583;6.29334046;10.21
586535;31.77621005;40.00016743;6.52437328;15.95813300;31.84615015;60.80244365;95.55411388;9
7.91562250
```

6.2.4 Temperature Compensation File

If the sensor board supports temperature correction with an onboard temperature sensor, and the temperature per channel is to be corrected, the option can be actively selected for correction in the GUI.

Figure 30:
Activated Temperature Compensation in the Main Tab of the GUI



Temperature Sensors

LED Bottom: 28.19 °C

☑ Temperature Compensation

In this case, it opens a selection window for selecting the temperature compensation and the corresponding file. The file, in CSV format, contains in column 1, the temperature in °C, and the correction values for all the sensor channels in the following columns. The temperatures must be entered in an orderly sequence “from low to high” in range per line for one temperature. Figure 31 shows an example.

Figure 31:
Example of a Temperature Compensation File

```
Temp [deg C];F1;F2;F3;F4;F5;F6;F7;F8;Clear;NIR

0;1.046363857;1.042574276;1.03818407;1.034293382;1.022931904;1.027424675;1.021267811;1.0168
07684;1.031872986;1.074089106

1;1.044304016;1.040652851;1.036367219;1.032703761;1.021741875;1.026016411;1.020013718;1.015
803313;1.030282011;1.071082263

2;1.042244176;1.038731426;1.034550367;1.03111414;1.020551845;1.024608147;1.018759625;1.0147
98942;1.028691036;1.06807542

3;1.040184335;1.036810002;1.032733516;1.029524519;1.019361816;1.023199884;1.017505532;1.013
794571;1.027100061;1.065068577

4;1.038124495;1.034888577;1.030916664;1.027934898;1.018171786;1.02179162;1.016251439;1.0127
90199;1.025509085;1.062061734
```

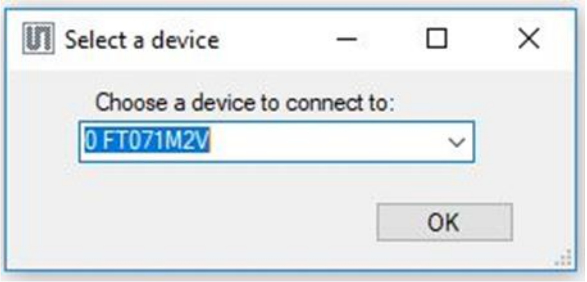
6.3 EVM Graphical User Interface

Connect the sensor board hardware to the system via an FTDI cable and double-click the software icon to open the GUI. If there are more than two FTDI cables connected to the computer, a pop-up window will appear, as shown below in Figure 32.

Please select the correct cable used for the sensor board and click OK. If there is only one FTDI cable, the software will automatically select the cable connected.

When the connection is good, the bottom section in the GUI will display the positive status of the FTDI connection, FTDI cable series number, auxiliary sensor ID, revision, and part number.

Figure 32:
Window for FTDI Cable Selection if Multiple FTDI Cables are Connected


Select a device

Choose a device to connect to:

0 FT071M2V

Check the part number, including any 0x-code, if any issues occur. A 0x zero code indicates no connection to the sensor hardware. In the event of a problem, check the USB driver installation and connections, start the software again, or use the scan function in the menu file.

6.4 Tab File

The GUI will automatically open when launching the software. If no device is connected, an error message will pop up.

When the GUI starts showing a red indicator at the bottom section of the FTDI connection, connect a device, navigate to the “File” tab in the top corner of the GUI, and click “Scan and Connect”. Afterward, the GUI will relaunch with the device connected. Use the “Disconnect” button to terminate the connection. Then, click the “Exit” button to end the GUI application.

6.5 Hot Keys

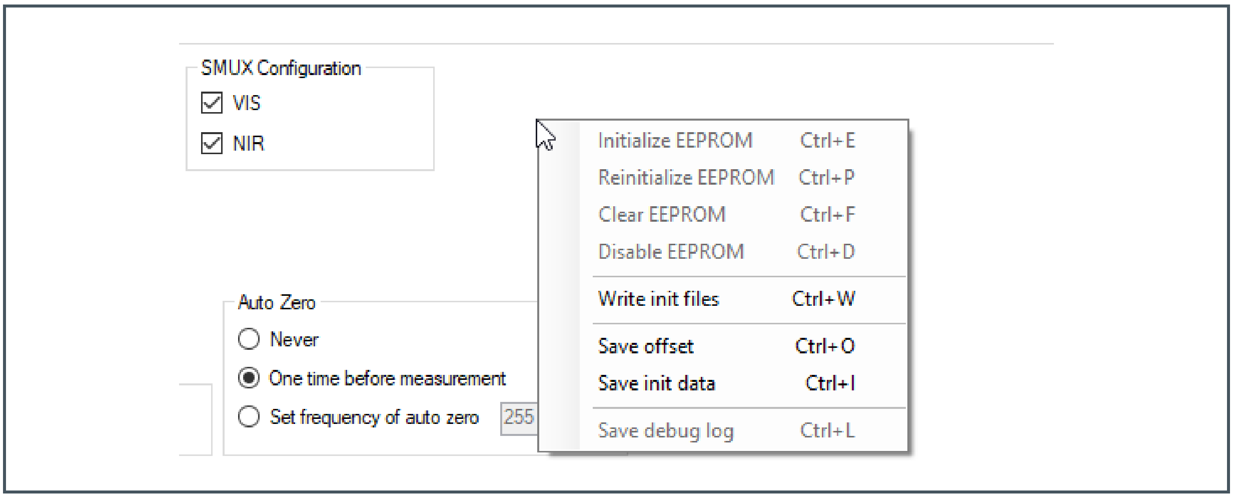
The following hotkeys, as “CTRL-functions”, are inserted in the GUI. Alternatively, the ctrl functions are implemented as a context menu, which can be opened by clicking the right mouse button in the main tab.

Figure 33:
Evaluation Kit Commands and Demonstrators

Command	Description
Ctrl + A	Start detection logging (only active when using ALS or reflection mode).
Ctrl + B	Use XYZ reflection from the last shot as the white balance (only active when using reflection mode).
Ctrl + C	Stop Script Processing (only active when using trace mode and an activated process).
Ctrl + D	Disable EEPROM. GUI uses data directly from the directory (EEPROM commands are only active for EEPROM-compatible devices).
Ctrl + E	Initialize or re-write the EEPROM with the files from the directory (EEPROM commands are only active for EEPROM-compatible devices).
Ctrl + F	Clear the EEPROM (EEPROM commands are only active for EEPROM-compatible devices).
Ctrl + I	Write initialization data from the memory to the EEPROM or initialization file (EEPROM commands are only active for EEPROM-compatible devices).
Ctrl + L	Save Log[11] file during the active session (only active when using ALS or reflection mode).
Ctrl + M	Read a new calibration file (only active when using ALS or reflection mode).
Ctrl + O	Use Basic_Counts with gain correction from the last shot as offset values.
Ctrl + R	Read mask file (only active when using ALS or reflection mode).
Ctrl + S	Save detection logging (only active when using ALS or reflection mode).
Ctrl + W	Readout initialization files from the EEPROM to the directory (EEPROM commands are only active for EEPROM-compatible devices),
Ctrl + Z	Make a screenshot of a diagram.
Ctrl + left mouse button	Zoom-in on diagrams.

[11] Set in the file AS7341_Demo.config in the user directory the option <add key='LogFileEnable' value='True' /> to activate log files.

Figure 34:
Use the Context Menu "Ctrl-Function" (right-click the mouse button in the Main window to open it)



SMUX Configuration
☑ VIS
☑ NIR

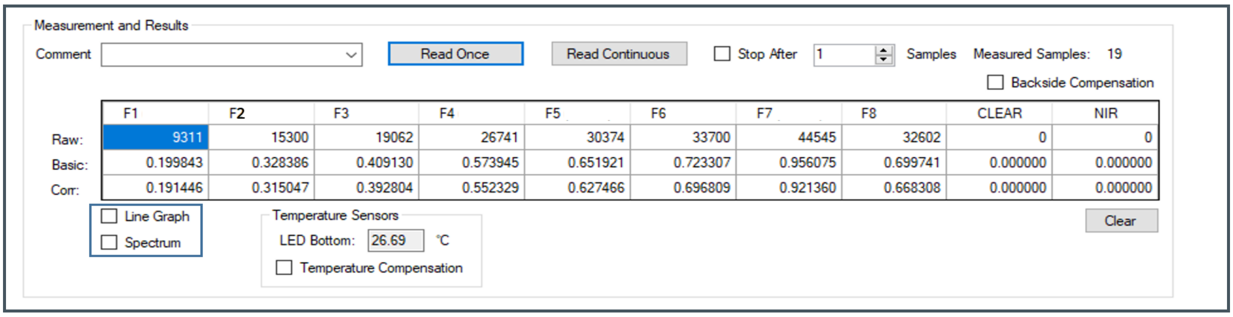
Auto Zero
○ Never
● One time before measurement
○ Set frequency of auto zero 255

Initialize EEPROM	Ctrl+E
Reinitialize EEPROM	Ctrl+P
Clear EEPROM	Ctrl+F
Disable EEPROM	Ctrl+D
Write init files	Ctrl+W
Save offset	Ctrl+O
Save init data	Ctrl+I
Save debug log	Ctrl+L

6.6 Sensor Board Test

To check the function of the EVK, start the measurement by pressing the "Read Once" button (see Figure 25). This executes one measurement step using "Easy Mode" (see chapter 6.6). It should show measured values in the table based on the sensor's location to a luminary in front of the sensor, and its spectrum in another window. Change the mode to "Expert Mode", adjust the Integration Time and Gain, to change the digits based on application requirements. More details about sensor functions and the parameters are listed in the sensor's datasheet or later in this manual. Select one of the plot options, "Line Graph" or "Spectrum" (see Figure 35 below the table), to see the results (Corrected counts) as graphical output. The table (Figure 35) shows the "RAW", "Basic", and "Corrected" values. Check the given results based on logic and reason. Usually, when in Easy and Expert mode, by default, the optimization for Gain and Integration Time is switched-on. In Expert mode, optimization can be turned off to increase the parameters for conversion in the case of zero digits, or to decrease in the case of saturation.

Figure 35:
First Sensor Board Test – Output of the First Results in the Measurement Window



Measurement and Results

Comment [ ] Read Once Read Continuous □ Stop After 1 Samples Measured Samples: 19

□ Backside Compensation

	F1	F2	F3	F4	F5	F6	F7	F8	CLEAR	NIR
Raw:	9311	15300	19062	26741	30374	33700	44545	32602	0	0
Basic:	0.199843	0.328386	0.409130	0.573945	0.651921	0.723307	0.956075	0.699741	0.000000	0.000000
Corr:	0.191446	0.315047	0.392804	0.552329	0.627466	0.696809	0.921360	0.668308	0.000000	0.000000

□ Line Graph
□ Spectrum

Temperature Sensors
LED Bottom: 26.69 °C
□ Temperature Compensation

Clear

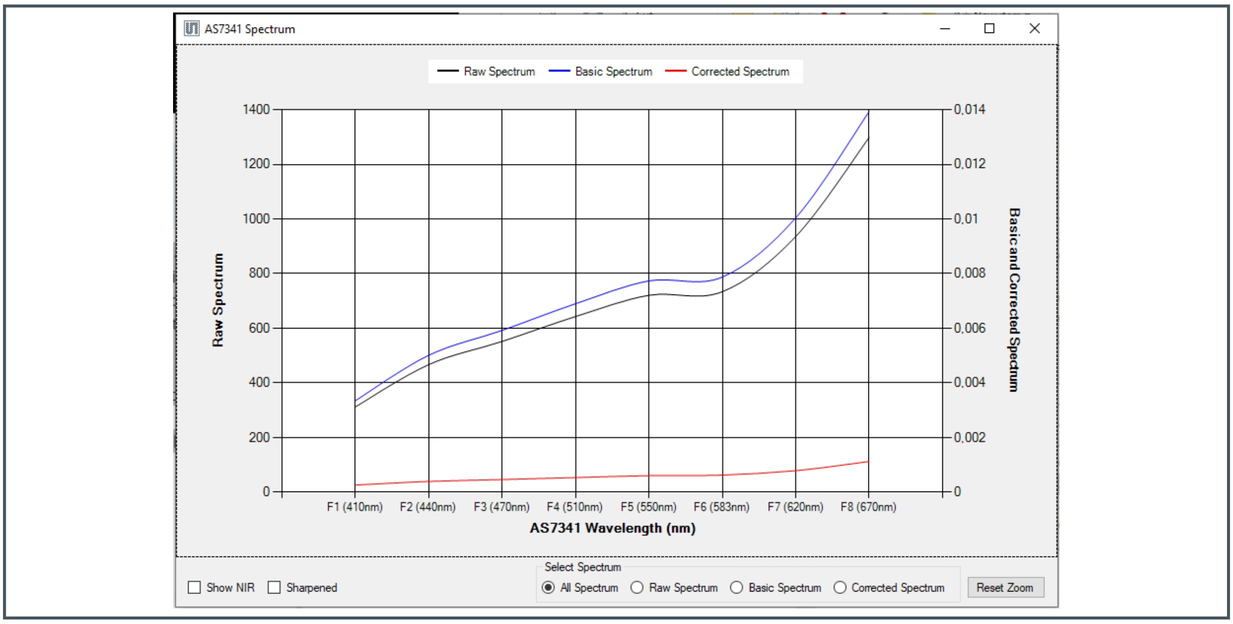
An increase/decrease should change the counts and values in the table or diagrams. Look for the printed values and check them with the specified commands and parameters in the Init_file - to ensure that the GUI works well with all the initialization files. Always check the stability of a test setup before the actual measurements begin. Set the target to its minimum and maximum - to check the required dynamic range. Ensure the parameter setup for gain and integration time can realize the full dynamic range. Verify any potential drifts like temperature effects (if temperature correction is supported) and interferences like ambient lighting (which affects measurements). For accuracy, verify and delete or compensate for all the effects before starting measurements.

6.7 Offset Compensation

Offset is an effect that should be corrected at the beginning after completing the setup and the definition of all the sensor parameters to obtain the best possible sensor results. In Reflection mode, such an offset is caused, for example, by direct irradiation of the LEDs via the optics onto the sensor. This offset is measured directly with the “set and switched on” LEDs as the sensor result without a target (reflection) in front of the sensor; for example, a setup where the sensor looks into a dark room with activated LEDs).

Figure 36 shows a measurement result as a RAW value (black), Basic_Counts (blue), and Corrected Counts (red, with a preset standard offset for the kit in correction), where the sensor measures into a dark room with the activated LEDs. Although no result is expected from a target in reflection, the sensor naturally shows digital counts as a result, which are caused by internal reflections directly from the LED to the sensor. This error, as the offset, is constant with and without a target, independent of the color the sensor later measures. Since the red line is not on the zero line directly, the preset offset must be corrected or not, depending on its absolute size and required accuracy.

Figure 36:
Sensor Results Without the Precise Offset Correction (see the red line)

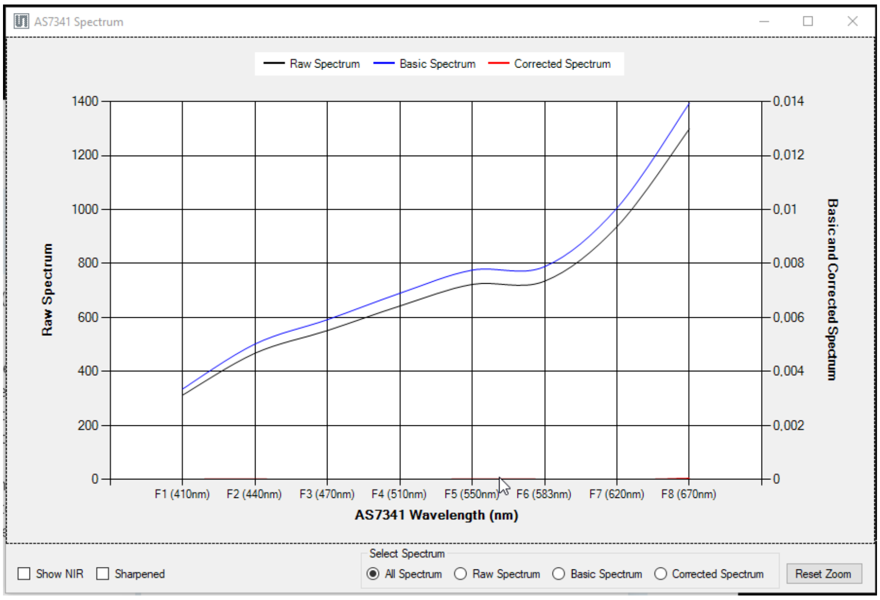


□ Show NIR □ Sharpened

Select Spectrum: ● All Spectrum ○ Raw Spectrum ○ Basic Spectrum ○ Corrected Spectrum

In general, this correction can be done by measuring under identical conditions without offset and entering the result values as an offset definition in the INIT file. Alternatively, the last measurement can be set as the offset with “CTRL+O”. Its sensor values are then stored in the GUI as the offset and always subtracted from the current measurements as an offset. Only at the end of the session will the GUI prompt to save this offset in the Init_file before it is closed.

Figure 37:
Sensor Results with the Correct Offset Correction (Corrected values are on the Zero Curve)



□ Show NIR □ Sharpened

Select Spectrum
● All Spectrum ○ Raw Spectrum ○ Basic Spectrum ○ Corrected Spectrum

6.8 Operation Mode

The GUI has two modes of operations – Easy and Expert mode. By default, Easy mode is selected. Users can switch modes by selecting the radio button of Expert mode.

In Easy mode, most of the parameter settings are automatically set. The user has limited permission to make modifications to the parameter settings in Easy mode. The parameters like Gain, Integration Time, LED settings, SMUX configuration, etc. cannot be changed in Easy mode. However, the optimization of Gain and Integration Time are active during “Easy mode” measurements. Also, Easy mode has limitations and restrictions in the application-specific analysis windows.

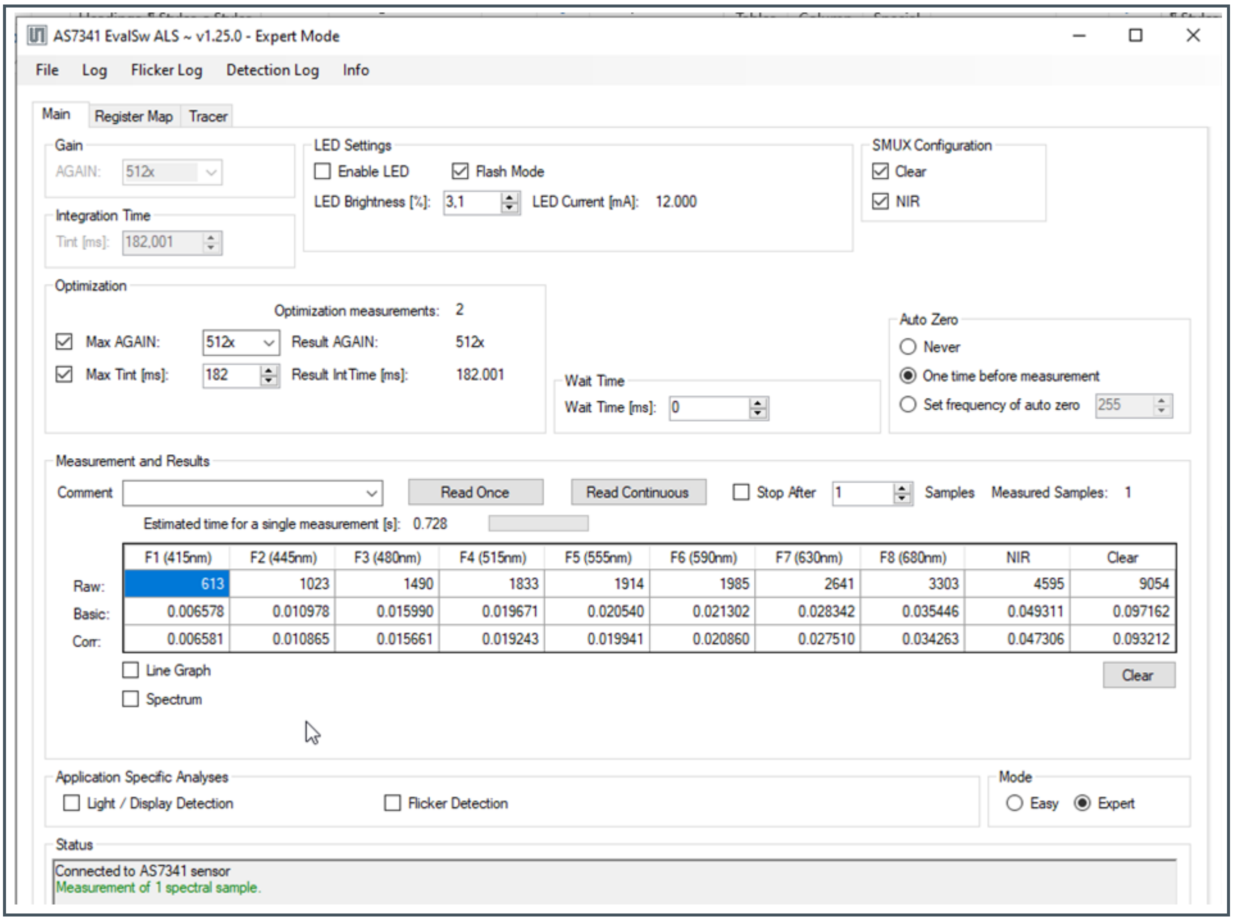
Expert mode overcomes all restrictions in Easy mode. Users can control all the parameter settings in Expert mode. In this mode, the optimization of max AGAIN and Max TINT to control the Gain and Integration time is automatically turned off. Switch them on manually to use it in Automatic mode. For operating the GUI in Expert mode, by default, users can use the “*xxx Demo Version.bat*” batch file with the below command and execute the file in “*C:\Program Files (x86)\ams-OSRAM\AS734x Demo*” or create a shortcut icon of this batch file on the desktop. Also, the software version can be edited with this batch file.

```
start "" ".\AS7341_Demo_Software.exe" -Expert -Version=X.XX.00
```

6.9 Main Window

The main window in Expert mode contains the user interface with control buttons, fields, selection boxes, and output values for the identified connected device. The main window permits the configuration of the sensor device and initializes the default setting for the devices. It allows the user to modify and configure the Integration time setting, Gain setting, LED setting, “AutoZero”, “SMUX Configuration”, “Dynamic Conversion” (Optimization), and other application-specific sensor functions and demos.

Figure 38:
Main Tab of the GUI



AS7341 EvalSw ALS ~ v1.25.0 - Expert Mode

Gain
AGAIN: 512x

Integration Time
Tint [ms]: 182,001

LED Settings
□ Enable LED ☑ Flash Mode
LED Brightness [%]: 3,1 LED Current [mA]: 12.000

SMUX Configuration
☑ Clear
☑ NIR

Optimization
Optimization measurements: 2
☑ Max AGAIN: 512x Result AGAIN: 512x
☑ Max Tint [ms]: 182 Result Int Time [ms]: 182.001

Wait Time
Wait Time [ms]: 0

Auto Zero
○ Never
● One time before measurement
○ Set frequency of auto zero 255

Measurement and Results
Comment
□ Stop After 1 Samples Measured Samples: 1
Estimated time for a single measurement [s]: 0.728

	F1 (415nm)	F2 (445nm)	F3 (480nm)	F4 (515nm)	F5 (555nm)	F6 (590nm)	F7 (630nm)	F8 (680nm)	NIR	Clear
Raw:	613	1023	1490	1833	1914	1985	2641	3303	4595	9054
Basic:	0.006578	0.010978	0.015990	0.019671	0.020540	0.021302	0.028342	0.035446	0.049311	0.097162
Corr:	0.006581	0.010865	0.015661	0.019243	0.019941	0.020860	0.027510	0.034263	0.047306	0.093212

□ Line Graph
□ Spectrum

Application Specific Analyses
□ Light / Display Detection □ Flicker Detection

Mode
○ Easy ● Expert

Status
Connected to AS7341 sensor
Measurement of 1 spectral sample.

Information

The higher the counts (before saturation), the better the accuracy. Changing the Gain or TINT will affect the counts. Both parameters will have different effects like FSR, noise, linearities, time, and others.

Integration Time (TINT): Integration time is one parameter that affects the sensor result (= digital counts or digits). The Integration Time directly affects the saturation (FSR 16-bit = 2^16 is reached for the first time at 182 ms = 2.78 µs * 2^16). The Integration time is set using ATIME (0x81) and ASTEP (0xCA, 0xCB) registers, and it is displayed in milliseconds. It is calculated using Equation 4 below.

Equation 4:

$$tint = (ATIME + 1) \times (ASTEP + 1) \times 2.78\ \mu s$$

The Integration Time (TINT) parameter, like ATIME and ASTEP, can be set by clicking the up or down arrow button. ATIME sets the number of integration steps from 0 to 255. Set the Integration time per step, in increments of 2.78 µs. ASTEP sets the Integration time from 1 to 65534 steps. The default configuration in the GUI for these two registers are ASTEP = 0 and ATIME = 65534, which results in an Integration time of 182 ms. The Sensor specification does not allow both settings – ATIME and ASTEP – to be set to zero.

Gain is the second parameter to affect the sensor result (= digital counts). The Gain control, AGAIN, allows the user access to the Gain settings in the 0xAA Register (4:0 bits). The Gain amplifies the signal of the six integrated ADCs to increase sensitivity by switching to a higher Gain value. The Gain options include eleven alternative Gain stages between 0.5x and 512x. Select these options from the list box when the down arrow is pressed.

Enable Optimization: An algorithm of the GUI switches on, analyzing the sensor output and parameter setting to find an optimal ADC parameter, set for Gain and TINT at given maximum values. The goal of optimization is to achieve a maximum number of digits at high accuracy. Usually, measurements in noise and saturation should be prevented by using this mode. By default, this is ON during Easy mode and OFF during EXPERT mode. Optimization of Gain is enabled by checking Max AGAIN and denoting the maximum Gain value to be considered in the list box besides the checkbox. Similarly, optimization of Integration time is enabled by checking Max TINT and stating the maximum value to consider in the list.

Wait Time: It is defined in the GUI as the waiting time between two measurements in ms and possible steps in 10 ms increments. The function does not use the WTIME (LONG) function of the sensor chip.

LED Setting (LED Current): If LEDs are mounted on the sensor board for Reflection mode, enable the switch on the LED and set the LED currents. The current can be set using the up-down control, which has a range of 1% up to 100% of the supported LED current[12]. Activate Flash mode if the LED is only switched on during the measurement process.

Auto Zero: This sets the options and frequency at which the device performs auto zero of the spectral engines to compensate for changes in the device temperature.

SMUX Configuration: The device integrates a multiplexer (SMUX). With the SMUX, it is possible to map all available photodiodes to one of the six available light-to-frequency converters (ADCx). After the power-up of the device, the SMUX needs to be configured before a spectral measurement begins. Here in the GUI, the SMUX is pre-configured to work with all the spectral channels - excluding **VIS** and **NIR**, which can be selected separately.

Measurement Setting: Select “Read Once” or “Read Continuous” to measure systematically or in Continuous Mode (an alternative with a specified number of steps), and/or to stop a Continuous Mode after n steps. The ADC results are printed after each measurement as numeric values representing RAW_Counts, calculated Basic_Counts, or Corrected_Counts.

Raw_Counts: Raw_Counts represent the counts from the ADC depending on the setup used (Offset, Gain, Integration Time, LED current, etc.).

Basic_Counts: The Basic_Counts are calculated based on the RAW measurement values and the corresponding Gain and Integration time at that time - to get sensor results not dependent on the parameter setup (Gain, TINT).

Equation 5:

$$Basic\_Counts = (Raw\_Counts)/((Gain * TINT)\ )$$

Corrected_Counts: These are the results of calculations based on specified parameters in init_list, e.g. Gain_Correction or Correction_Factor and Offset.

Equation 6:

$$Corrected\_Counts = Basic\_Counts * Gain\_Correction * Correction\_Factor - Offset$$

Figure 39 shows the data flow used and the order in making corrections.

Select “Line Graph” or “Spectrum” (see Figure 23) if the sensor should be presented as diagrams for RAW values. These diagram windows include some functions to adapt the graphical output in wavelength and data. Such diagrams can be saved on a hard disk as a bitmap file or any other format by selecting the keyboard combination “Ctrl + Z”.

STATUS is a user interface and shows the GUI messages and/or failure reports.

[12] LED current, light intensity, and/or maximal driver current depend on the used LEDs and the connected power supply. Check the LED datasheet before using this LED setting. A very high LED current can destroy the LEDs.

Figure 39:
Data Flow Used in the GUI

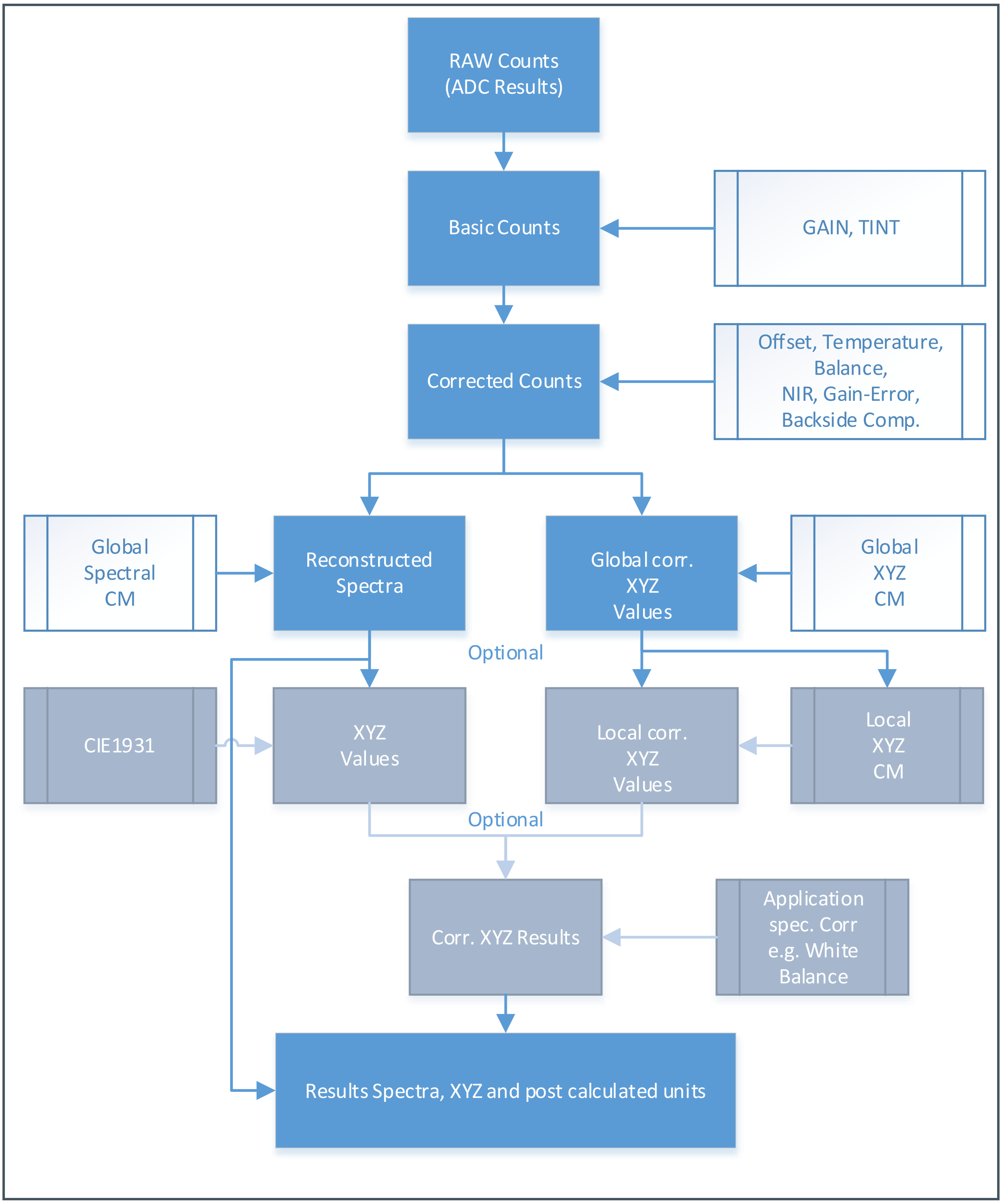
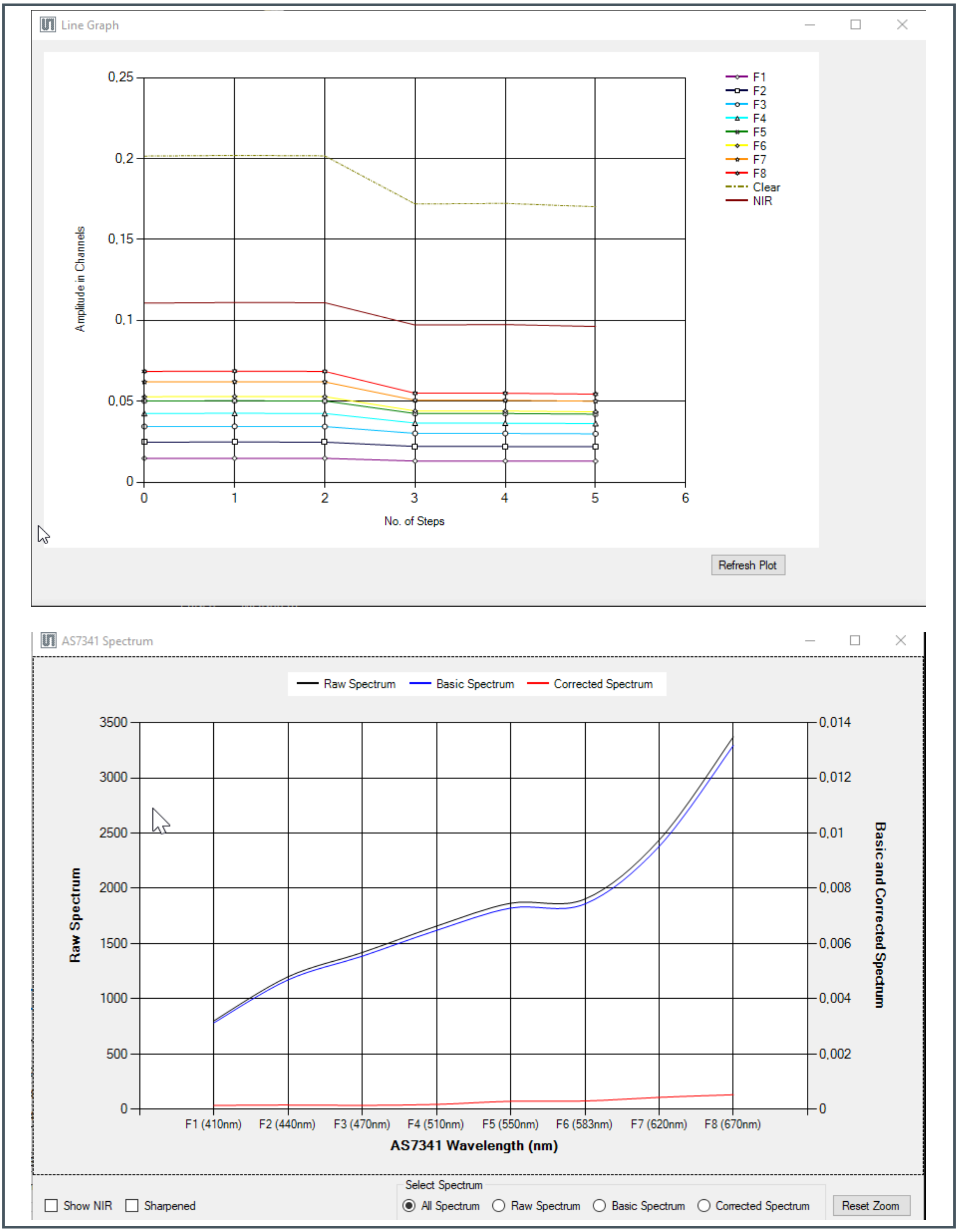


Figure 40:
Sensor Results in a Line Diagram and Spectrum (Offset Reflection)(1)

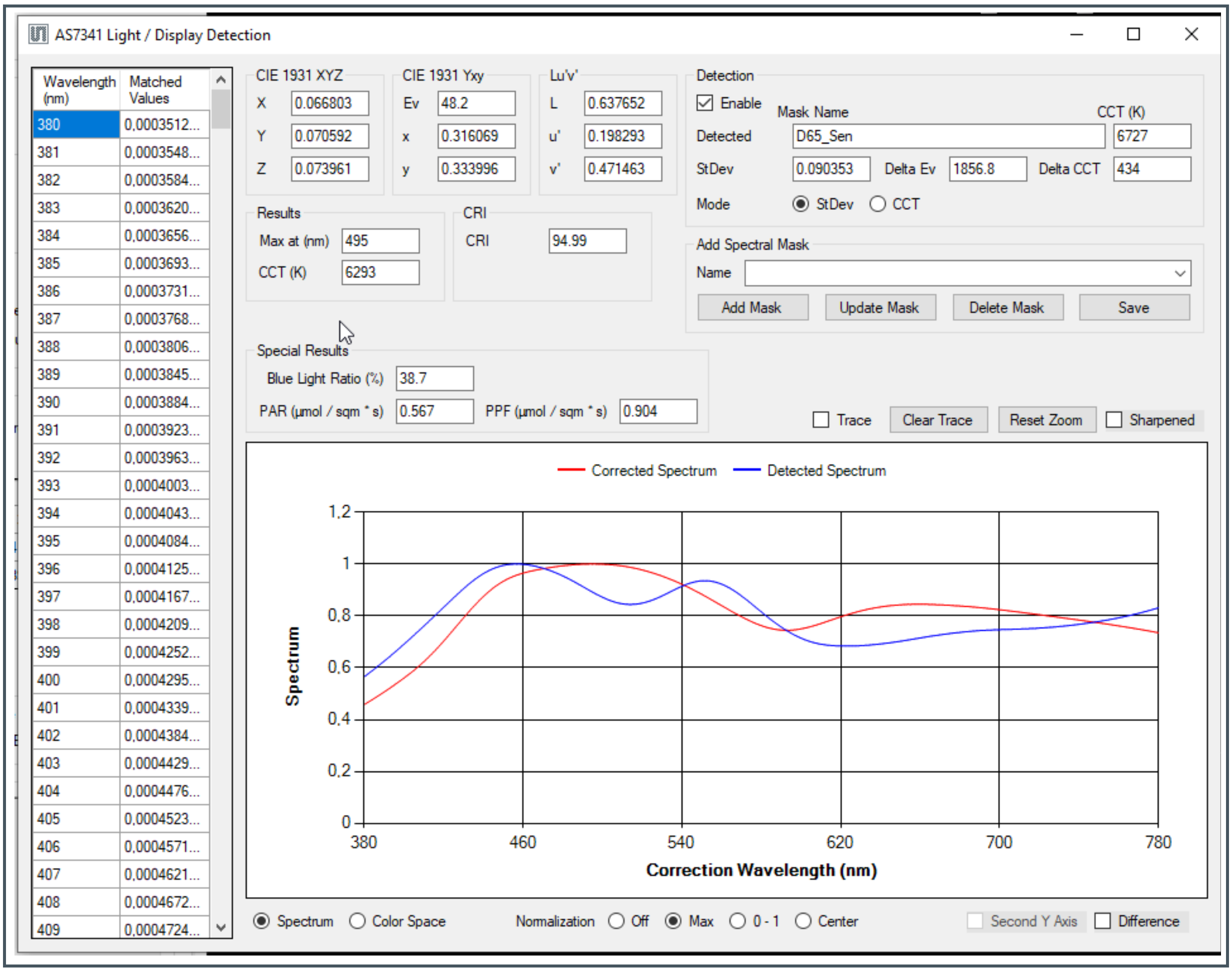


[1] Press Ctrl+Mouse to activate a magnifying glass in the diagram and enlarge a section in the spectra.

6.10 ALS Light Detection

ALS (Ambient Light Sensing) or Light Detection identifies the type of light, max (peak), CCT, lx (Ev illumination intensity), CRI, and other spectral and color results based on a reconstructed spectrum of the sensor results (e.g. Blue Light Ratio). Clicking the “Light/Display Detection” checkbox in the main window and pressing “Read Once” opens a new pop-up window, and shows the results as a vector, graphics (a spectrum as a list, diagram, or xy-oriented color point in the CIE1931 color space), calculated CIE1931 results[13], and the detected light source from the spectral mask compare - if this function is enabled as detection. A calibration file is required to correct the raw sensor values and transfer them into a light spectrum. This spectrum is used to calculate the photometric CIE1931 color quantities.

Figure 41:
Pop-Up Window of Light Detection by Spectral Light Comparison with the Enabled Mask Detection



AS7341 Light / Display Detection

Wavelength (nm)	Matched Values
380	0,0003512...
381	0,0003548...
382	0,0003584...
383	0,0003620...
384	0,0003656...
385	0,0003693...
386	0,0003731...
387	0,0003768...
388	0,0003806...
389	0,0003845...
390	0,0003884...
391	0,0003923...
392	0,0003963...
393	0,0004003...
394	0,0004043...
395	0,0004084...
396	0,0004125...
397	0,0004167...
398	0,0004209...
399	0,0004252...
400	0,0004295...
401	0,0004339...
402	0,0004384...
403	0,0004429...
404	0,0004476...
405	0,0004523...
406	0,0004571...
407	0,0004621...
408	0,0004672...
409	0,0004724...

CIE 1931 XYZ
X 0.066803
Y 0.070592
Z 0.073961

CIE 1931 Yxy
Ev 48.2
x 0.316069
y 0.333996

Lu'v'
L 0.637652
u' 0.198293
v' 0.471463

Results
Max at (nm) 495
CCT (K) 6293

CRI
CRI 94.99

Special Results
Blue Light Ratio (%) 38.7
PAR (µmol / sqm * s) 0.567
PPF (µmol / sqm * s) 0.904

Detection
☑ Enable
Mask Name / CCT (K)
Detected D65_Sen 6727
StDev 0.090353 Delta Ev 1856.8 Delta CCT 434
Mode ● StDev ○ CCT

Add Spectral Mask
Name

● Spectrum ○ Color Space Normalization ○ Off ● Max ○ 0 - 1 ○ Center □ Second Y Axis □ Difference

[13] The INIT, calibration, and correction files prepared for the GUI are not valid for all applications. Therefore, incorrect or deviating results may be possible. In practice, this means that the user has to check, adjust, and prepare for themselves the INIT and correction or calibration files for their application. Furthermore, all faults must be eliminated or considered in the setup. This usually shows formally incorrect results in red in the status window to indicate irregularities.

The function can only be used with a calibrated sensor for Light Detection, where the calibration matrix has been specially adapted to this application, and where spectral masks based on this calibration exists. Otherwise, the results may range from inexact to illogical.

After opening the Light Detection, please start, at minimum, one measurement or more to activate and actualize all the data and results in Light Detection.

The left-hand side of the pop-up window shows the sensor results as a vector between wavelengths and reconstructed sensor results. The calibration file affects the wavelength dynamic (from 380 nm up to 1000 nm in the standard init files) and the step sizes (step size of 1nm in the standard init files). Data from the matrix in the reconstructed spectrum can be copied directly to another window using the copy/paste commands.

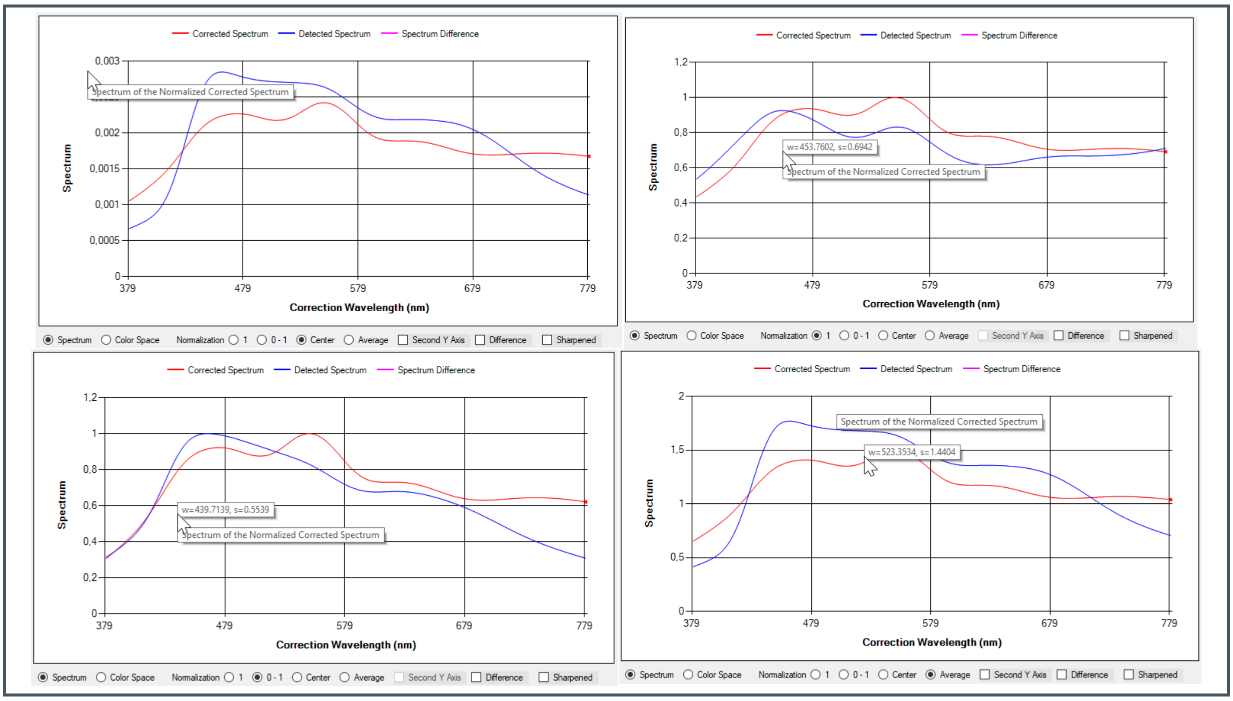
At the top of the pop-up windows are the results of all CIE1931 calculations as CIE1931 Tristimulus values XYZ, xy coordinates, Y (Ev in Lux[14]), uv, CCT[15], Peak, and CRI[15]. All these results, calculated standard values, and accuracy depend on the application, its conditions, deviations, calibration, parameters from init_file(s), and other effects.

For more details, see chapter 6.2 or the application notes for sensor calibration.

If Light Detection is enabled, then the GUI finds the nearest spectrum as masks, which is similar to the measured light, and its reconstructed spectrum. The comparison works based on alternative normalization procedures (Max, Min, 0-1, Center oriented or averaged), which will affect the results dramatically. Normalization must be adapted to the application, application-specific, and selected to get the best results.

[14] Only valid for a calibrated sensor and depends on the specified parameter, corr_lux, in the int_file (see chapter 6.2.1).
[15] See the status window in the basic tab if the CRI value is printed in red.

Figure 42:
Normalization Method That Affects the Mask Compare Process and Result



● Spectrum ○ Color Space Normalization ○ 1 ○ 0 - 1 ● Center ○ Average □ Second Y Axis □ Difference □ Sharpened

● Spectrum ○ Color Space Normalization ● 1 ○ 0 - 1 ○ Center ○ Average □ Second Y Axis □ Difference □ Sharpened

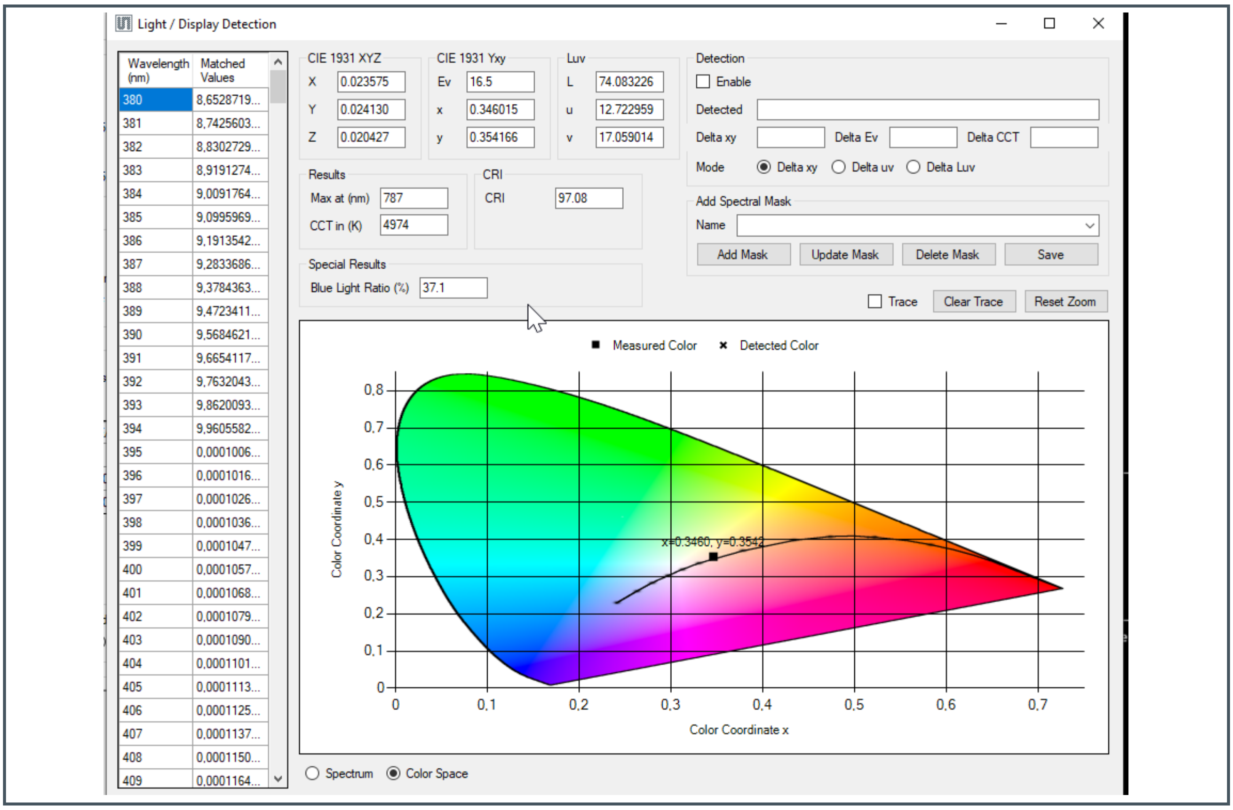
● Spectrum ○ Color Space Normalization ○ 1 ● 0 - 1 ○ Center ○ Average □ Second Y Axis □ Difference □ Sharpened

● Spectrum ○ Color Space Normalization ○ 1 ○ 0 - 1 ○ Center ● Average □ Second Y Axis □ Difference □ Sharpened

The spectral mask comparison uses the specified format from the calibration file (spectral data, XYZ) as a calculation basis for checking the differences between measured and normalized sensor results and masks. This means that spectral error squares for spectrum, Delta E (or Dxy or Dab), or other values (selectable) and their standard deviation (or specific routines) are used to find out the nearest mask.

Figure 41 shows the results from the sensor and its derived color and photometric quantities in numerical form and as a spectral diagram. Figure 43 shows the same result in the CIE1931 color space.

Figure 43:
Diagram of CIE1931 Color Space and the Printed Results for Light Detection



Light / Display Detection

Wavelength (nm)	Matched Values
380	8,6528719...
381	8,7425603...
382	8,8302729...
383	8,9191274...
384	9,0091764...
385	9,0995969...
386	9,1913542...
387	9,2833686...
388	9,3784363...
389	9,4723411...
390	9,5684621...
391	9,6654117...
392	9,7632043...
393	9,8620093...
394	9,9605582...
395	0,0001006...
396	0,0001016...
397	0,0001026...
398	0,0001036...
399	0,0001047...
400	0,0001057...
401	0,0001068...
402	0,0001079...
403	0,0001090...
404	0,0001101...
405	0,0001113...
406	0,0001125...
407	0,0001137...
408	0,0001150...
409	0,0001164...

CIE 1931 XYZ
X 0.023575
Y 0.024130
Z 0.020427

CIE 1931 Yxy
Ev 16.5
x 0.346015
y 0.354166

Luv
L 74.083226
u 12.722959
v 17.059014

Detection
☐ Enable
Detected
Delta xy
Delta Ev
Delta CCT
Mode ● Delta xy ○ Delta uv ○ Delta Luv

Results
Max at (nm) 787
CCT in (K) 4974

CRI
CRI 97.08

Special Results
Blue Light Ratio (%) 37.1

Add Spectral Mask
Name

☐ Trace

○ Spectrum ● Color Space

All the diagrams can be saved on a hard disk as a bitmap file by selecting the keyboard combination Ctrl + Z. Use Ctrl and the left mouse button in combination to activate a temporary magnifying glass in the diagram.

The activated “Trace” button allows several measuring points in the diagram until deselecting Trace or pressing the “Clear Trace” button.

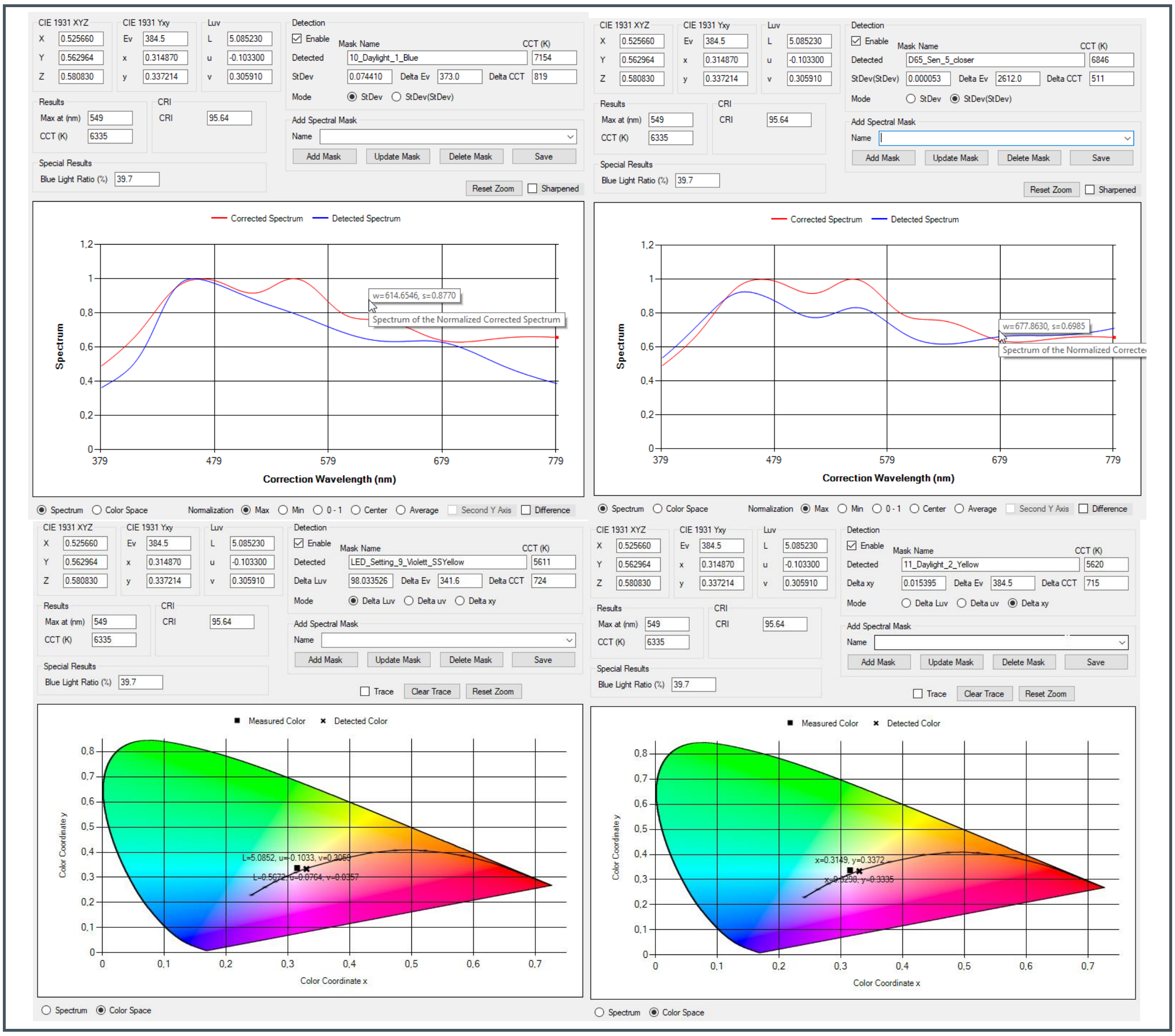
Not selecting Trace means only the most recent color coordinates are displayed as one color point in the diagram.

In general, the measured and recognized results will have deviations. Non-overlapping curves indicate such deviations. The reasons are different conditions in the setup for measurements, spectral differences of light sources (measured and specified), and interferences by mixed lightings, tolerances, and drifts over time or shifts over temperature, different fields of views, and others.

The more similar the types and samples of the light sources are between measurement and mask, and the smaller the deviations of the measurement conditions are, or the more adapted the algorithm is in the application, the better the result of the comparison will be.

The following example in Figure 44 shows examples of different results for the same measurement but using different algorithms for evaluation.

Figure 44:
Example of Detection in Different Modes



CIE 1931 XYZ: X 0.525660, Y 0.562964, Z 0.580830
CIE 1931 Yxy: Ev 384.5, x 0.314870, y 0.337214
Luv: L 5.085230, u -0.103300, v 0.305910
Results: Max at (nm) 549, CCT (K) 6335
CRI: CRI 95.64
Special Results: Blue Light Ratio (%) 39.7

Detection (top left): ☑ Enable
Detected: Mask Name 10_Daylight_1_Blue, CCT (K) 7154
StDev 0.074410, Delta Ev 373.0, Delta CCT 819
Mode: ● StDev ○ StDev(StDev)
□ Sharpened

Detection (top right): ☑ Enable
Detected: Mask Name D65_Sen_5_closer, CCT (K) 6846
StDev(StDev) 0.000053, Delta Ev 2612.0, Delta CCT 511
Mode: ○ StDev ● StDev(StDev)
□ Sharpened

● Spectrum ○ Color Space Normalization ● Max ○ Min ○ 0 - 1 ○ Center ○ Average □ Second Y Axis □ Difference

Detection (bottom left): ☑ Enable
Detected: Mask Name LED_Setting_9_Violett_SSYellow, CCT (K) 5611
Delta Luv 98.033526, Delta Ev 341.6, Delta CCT 724
Mode: ● Delta Luv ○ Delta uv ○ Delta xy
□ Trace

Detection (bottom right): ☑ Enable
Detected: Mask Name 11_Daylight_2_Yellow, CCT (K) 5620
Delta xy 0.015395, Delta Ev 384.5, Delta CCT 715
Mode: ○ Delta Luv ○ Delta uv ● Delta xy
□ Trace

○ Spectrum ● Color Space

In the image on the left-hand side from the example above (Figure 44), the mask with the smallest standard deviation over the entire spectrum in the selected wavelengths is displayed as the result, i.e. the mask with the name “10_Daylight_1_Blue” is spectrally closest to the measured and reconstructed light spectra. In the image on the right-hand side from the example above (Figure 44), the mode, StDev (Standard Deviation), is selected, i.e. the mask with the smallest standard deviations between the CIE1931 wavelengths and their standard deviation for the separated X, Y, and Z is selected as the result. This means the most similar spectrum, but with the smallest differences between X, Y, and Z (affects the color temperature calculation), is considered and detected.

The evaluation in the CIE1931 color space is shown in Figure 44 as examples and considers the modes Delta Luv and Delta xy. Thus, the mask is found as the result of the detection with the smallest distance in the Luv or xy color space, while spectral details are not considered. Therefore, a result of an “LED adjustment” is possible even though the original light source was natural light.

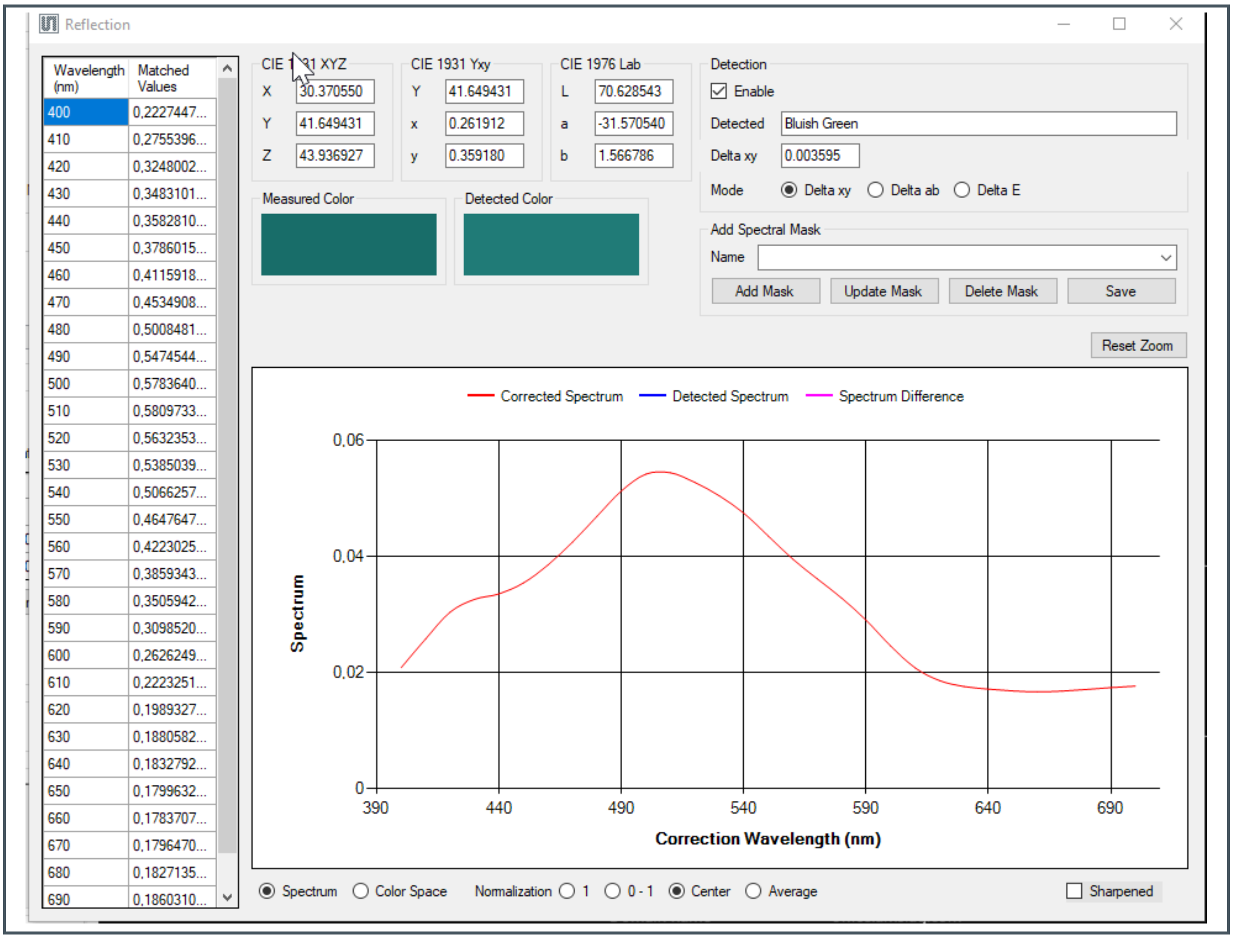
If the GUI does not identify a spectral mask within the given accuracy from the Init_file, the customer can specify a new spectral mask by pressing “Add Mask’”. This will add the spectrum of the measured light source (not normalized) as a new mask by a newly defined name. This new mask will be temporarily active and is active until the end of the program.

The mask will only be inserted into the mask file using the Save button. The GUI will not check any duplicates in the mask file during all the activities (read, add, save). Therefore, be careful and check the names and masks before any changes. The Light Detection function will not recognize a light source in the case of a non-existing mask file but works for the CIE1931 calculations.

6.11 Reflection Mode

Reflection mode is prepared for measuring reflected light from colored surfaces in contact measurement based on the Evaluation Kit Reflection Mode (see Figure 2). After setting up all the parameters to get optimized counts, the function can be opened by selecting “Reflection mode” in the section application-specific analyzes “Reflection Mode” (Chapter 6.11).

Figure 45:
Pop-Up Window of Reflection Mode with Spectral Diagram and Detection (Spectrally calibrated)



Wavelength (nm)	Matched Values
400	0,2227447...
410	0,2755396...
420	0,3248002...
430	0,3483101...
440	0,3582810...
450	0,3786015...
460	0,4115918...
470	0,4534908...
480	0,5008481...
490	0,5474544...
500	0,5783640...
510	0,5809733...
520	0,5632353...
530	0,5385039...
540	0,5066257...
550	0,4647647...
560	0,4223025...
570	0,3859343...
580	0,3505942...
590	0,3098520...
600	0,2626249...
610	0,2223251...
620	0,1989327...
630	0,1880582...
640	0,1832792...
650	0,1799632...
660	0,1783707...
670	0,1796470...
680	0,1827135...
690	0,1860310...

CIE 1931 XYZ
X 30.370550
Y 41.649431
Z 43.936927

CIE 1931 Yxy
Y 41.649431
x 0.261912
y 0.359180

CIE 1976 Lab
L 70.628543
a -31.570540
b 1.566786

Detection
☑ Enable
Detected Bluish Green
Delta xy 0.003595
Mode ● Delta xy ○ Delta ab ○ Delta E

Measured Color

Detected Color

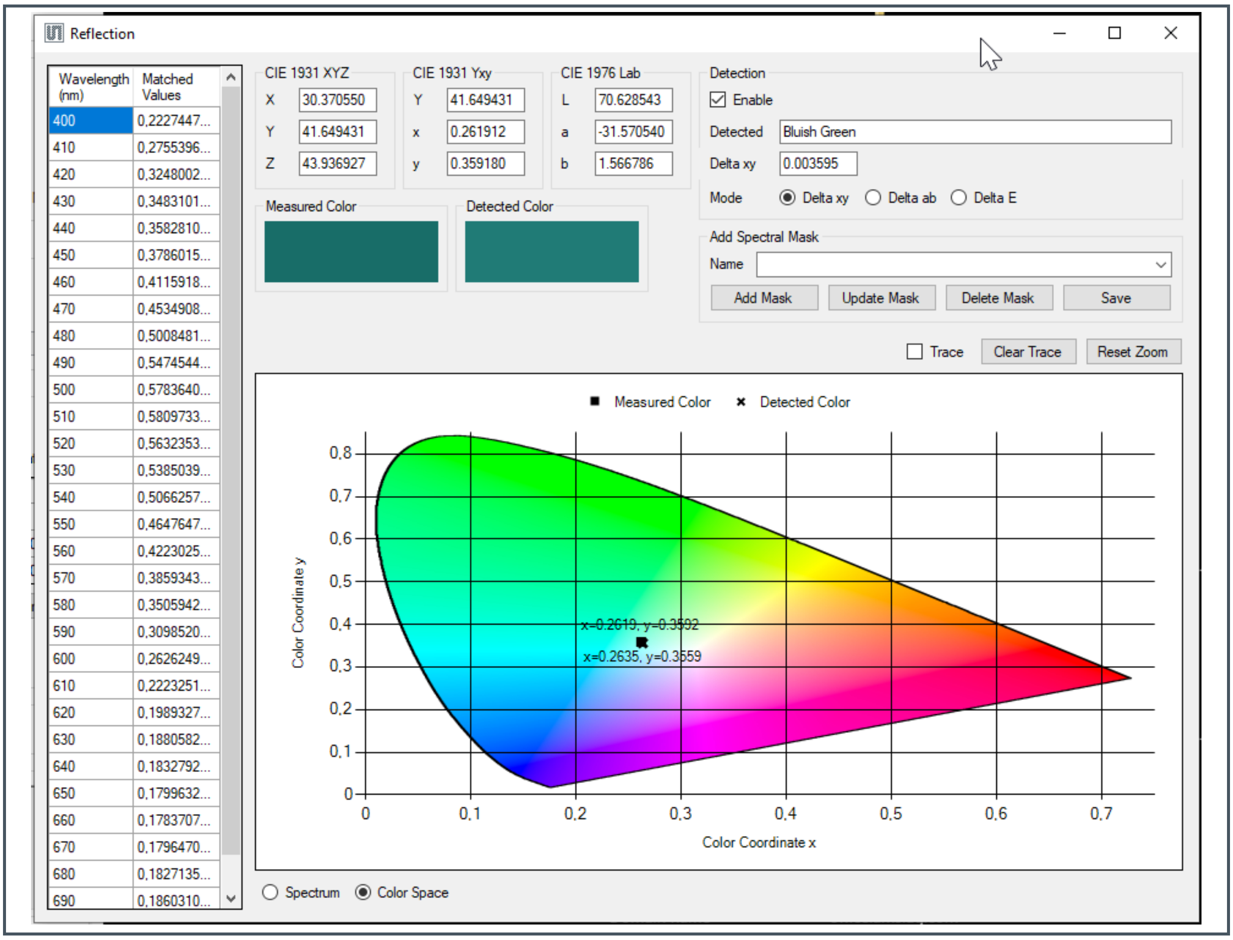
Add Spectral Mask
Name

● Spectrum ○ Color Space Normalization ○ 1 ○ 0 - 1 ● Center ○ Average □ Sharpened

A new pop-up window will open, which shows based on the calibration of the spectrum, CIE1931 calculated results: XYZ[16], xy, and Lab as a numeric result in the xy-diagram. Similar to light detection, after checking the checkbox for reflection, select the calibration files for reflection.

By default, LED settings are enabled in flash mode with a default current of 110 mA. This function only uses a wavelength in the VIS range. Therefore, XYZ Tristimulus values are the basis for the correction matrix and the mask file for color recognition. Select between the diagrams in XYZ or as a reconstructed spectrum. Detection is always based on the XYZ values.

Figure 46:
Pop-Up Window of Reflection Mode with XYZ Diagram and Detection (Spectrally Calibrated)



Wavelength (nm)	Matched Values
400	0,2227447...
410	0,2755396...
420	0,3248002...
430	0,3483101...
440	0,3582810...
450	0,3786015...
460	0,4115918...
470	0,4534908...
480	0,5008481...
490	0,5474544...
500	0,5783640...
510	0,5809733...
520	0,5632353...
530	0,5385039...
540	0,5066257...
550	0,4647647...
560	0,4223025...
570	0,3859343...
580	0,3505942...
590	0,3098520...
600	0,2626249...
610	0,2223251...
620	0,1989327...
630	0,1880582...
640	0,1832792...
650	0,1799632...
660	0,1783707...
670	0,1796470...
680	0,1827135...
690	0,1860310...

CIE 1931 XYZ
X 30.370550
Y 41.649431
Z 43.936927

CIE 1931 Yxy
Y 41.649431
x 0.261912
y 0.359180

CIE 1976 Lab
L 70.628543
a -31.570540
b 1.566786

Detection
☑ Enable
Detected Bluish Green
Delta xy 0.003595
Mode ● Delta xy ○ Delta ab ○ Delta E

Add Spectral Mask
Name

□ Trace

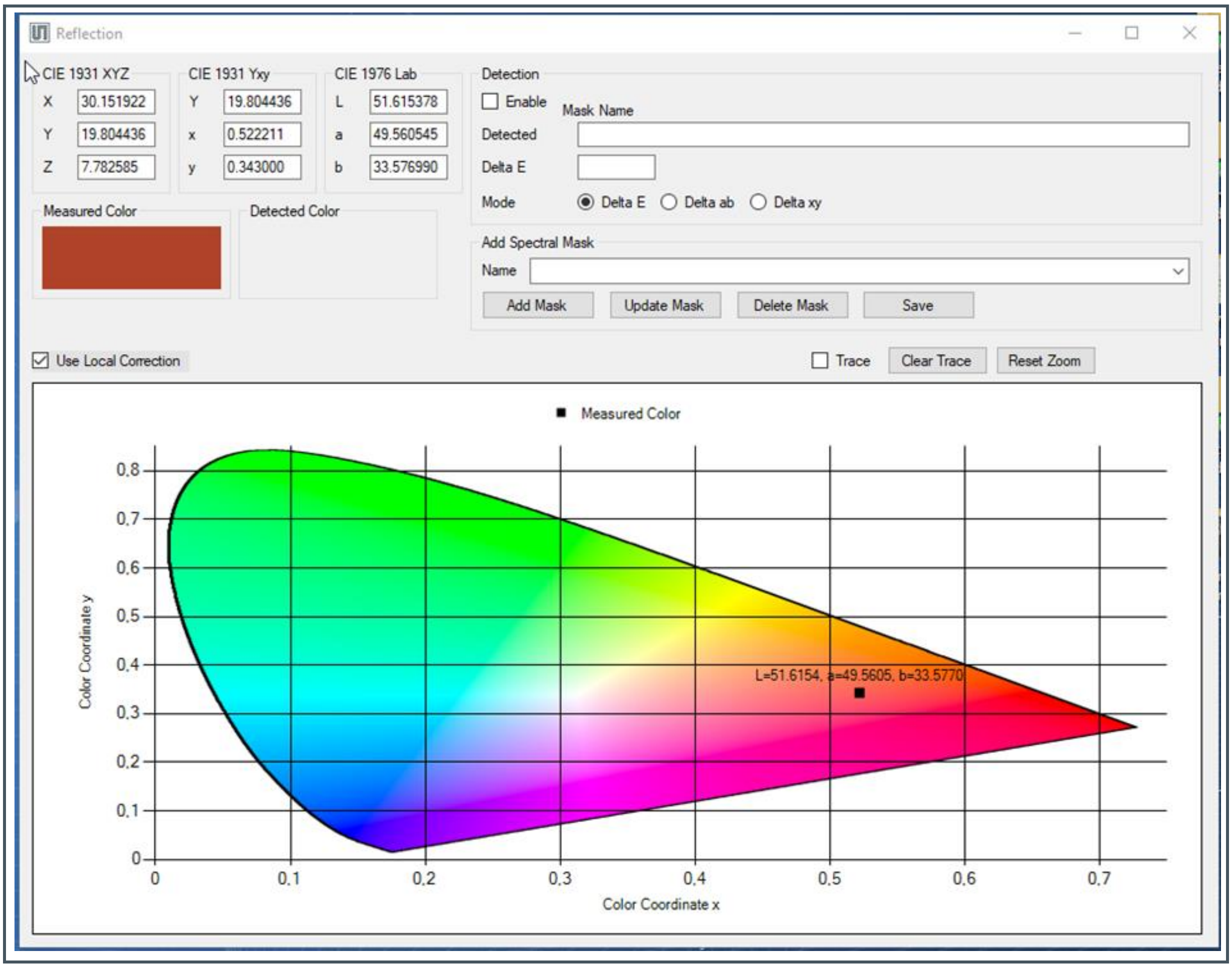
■ Measured Color ✖ Detected Color

○ Spectrum ● Color Space

After opening the Reflection mode, please start at the minimum, one measurement to activate and actualize all data and results in Reflection mode.

[16] The INIT, calibration, and correction files prepared for the GUI are not valid for all applications. Therefore, incorrect or deviating results may be possible. In practice, this means that the user has to check, adjust, and prepare the INIT and correction or calibration files for their application. Furthermore, all faults must be eliminated or considered in the setup. This usually shows formally incorrect results in red or west in the status window to indicate irregularities.

Figure 47:
Pop-Up Window of Reflection Mode with XYZ Diagram and Detection (XYZ for Local Calibration)



Reflection

CIE 1931 XYZ		CIE 1931 Yxy		CIE 1976 Lab	
X	30.151922	Y	19.804436	L	51.615378
Y	19.804436	x	0.522211	a	49.560545
Z	7.782585	y	0.343000	b	33.576990

Measured Color 🟥

Detected Color

Detection
□ Enable
Mask Name
Detected
Delta E
Mode ● Delta E ○ Delta ab ○ Delta xy

Add Spectral Mask
Name

☑ Use Local Correction

□ Trace

L=51.6154, a=49.5605, b=33.5770

When "Backside Compensation" is selected, the GUI activates two consecutive measurements in "Step" or "Continuous" mode. The first measurement is performed without the LED illumination and the second measurement with illumination. The results of the first measurement, in counts, are subtracted from the second measurement as background compensation[17].

The GUI realizes the calibration depending on the calibration data from the calibration file. If the calibration file contains data for "Local calibration", the local calibration can be switched on as a demo. In this case, the global calibration is used to determine the approximate position of the measured value in the color space in the first step. Based on this position, a local and temporary correction matrix is calculated and used as step 2 with twelve targets closest to the estimated color position. In

[17] Offset compensation can be applied for constant ambient light. Use backside compensation for dynamic ambient light, backside compensation is recommended.

most cases, such a calibration leads to an improvement in the calibration results. For further details, see [6].

If “Detection” is enabled, the GUI will show the recognized color from the mask file (point 1 of the xy-diagram), with the nearest distance (smallest Delta E or Delta ab or Delta xy) to the actual measured sensor value (point 2 in the xy-diagram). “Add Mask” inserts a new mask in the mask file. This new mask will be temporarily active until the end of the program. The mask will only be inserted into the mask file using the “Save” button. Pressing the “Update Mask” button replaces the detected existing reference mask values with the current measured values.

6.12 Tab Log

The log file stores the sensor setup and data in a CSV data Excel format based on the Windows setup[18]. Click “Start Logging” and/or “Stop Logging” to select the samples, and close the process using “Save Log” to store the CSV file and/or “Clear Log” to delete the sampled log data. The title and content in the log files are different for ALS, light, and reflection measurements.

6.13 Flicker Log

The flicker log functions are identical to the menu log functions. The only difference is that they solely store the hardware and FIFO flicker data.

6.14 Detection Log

This is for light and reflection detection logging. After enabling either light or reflection detection, select “Start” from the Detection Log Tab or “Ctrl+a” in the main tab, which will subsequently ask the user to save a CSV file. After each measurement, the user can save their light source by pressing “Save” for each detection or “Ctrl+s”.

A new pop-up window will ask the user to enter the name of the user light source, and the user can select whether the light is detected or not.

6.15 Flicker Detection

Light sources may be modulated at different frequencies, as their power supply will cause a flicker of the light source. Usually, light sources such as fluorescent tubes or incandescent bulbs flicker at

[18] It can result in issues in notations (fixed-point, semicolon, and comma) in combination with wizards to convert text into numeric values and tables. Use a standard Windows editor to replace the notations before processing with a wizard in the case of such formatting issues.

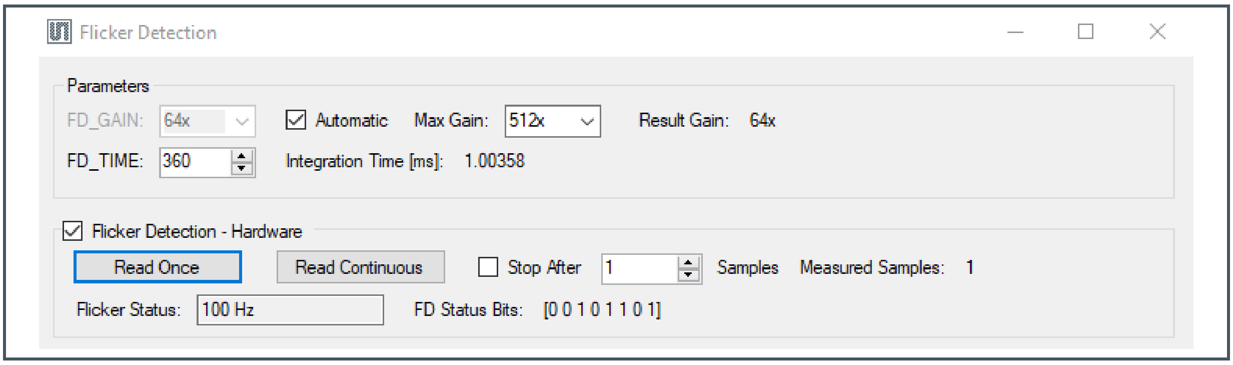
frequencies proportional to their power supply. An AC flicker caused by an alternating current power supply varies at a frequency of 50 Hz or 60 Hz.

The sensor device has an integrated ambient light flicker detection on-chip.

In the GUI, two types of flicker detection methods are implemented – hardware and FIFO methods, as shown in Figure 48 and Figure 49. This window will pop up when clicking the flicker detection in the main window, which is disabled at the same time as the ALS measurements in the main window. Selecting the "Flicker Detection – Hardware” checkbox enables the hardware flicker detection and disables the FIFO method of flicker detection and vice versa.

The flicker detection with the standard EVK is limited, with up to approximately 1800 Hz flicker frequency. This is due to the limited frequency used by the I²C to USB interface. Therefore, another solution named EVK Fast Demo is recommended in the case of fast flicker measurements (see [7]).

Figure 48:
Flicker Detection – Hardware



FD_GAIN: 64x

☑ Automatic

Max Gain: 512x

Result Gain: 64x

FD_TIME: 360

Integration Time [ms]: 1.00358

☑ Flicker Detection - Hardware

□ Stop After 1 Samples

Measured Samples: 1

Flicker Status: 100 Hz

FD Status Bits: [0 0 1 0 1 1 0 1]

By default, “Flicker Detection – Hardware” is selected. The hardware flicker detection detects the presence of 100 Hz and 120 Hz flickering. The “Read Once” button displays the status of the flicker in the “Flicker Status”. The corresponding bit in the FD status register is shown as “FD Status Bits”. Clicking the “Read Continuous” button will keep updating the FD samples. If the “Stop After” checkbox is selected, and the number of samples is mentioned, the mentioned samples will be measured upon pressing the “Read Continuous” button. The samples taken show the number of finished measurements.

Alternatively, the FIFO method of flicker detection is selected and the default integration time for flicker detection, Gain, and threshold are configured. Use the parameters, Gain, and integration time, to adapt the measurement to application-specific requirements by unchecking the Automatic checkbox.

The “FD Integration time” can be set by clicking the up-down arrow or by setting a value and pressing enter. The respective Integration time is calculated based on the formula (Fd_time +1)*2.78 and displayed in milliseconds. The “FD Gain” can be picked by choosing one of the values 0.5x, 1x, 2x, 4x, 8x, 16x, 32x, 64x, 128x, 256x, and 512x from the selection list. Similarly, FD Threshold can be selected as 1, 4, 8, and 16. FD Time and FD Gain define the sampling rate, amplitude of counts, and the maximum number of counts.

Click the “Read Once” button to update the Flicker detection graph with current samples of the FIFO data in bytes. The output of the flicker channel in counts (raw data) is plotted against the corresponding sample and the number in the primary Y-axis (left) and X-axis respectively (bottom), in red on the colored graph (Figure 49).

Clicking the “Read Continuous” button keeps updating the FD samples. If the “Stop After” checkbox is selected, and the number of samples is mentioned, the mentioned samples start to be measured after pressing the “Read Continuous” button.

The phrase samples taken show the number of finished measurements. In a single cycle, FIFO_Lvl samples give the maximum number of FIFO entries (every 2 Bytes), read out either before the overflow flag goes high, or the maximum sample taken is less than 250.

The sample levels are added back to back on consecutive measurements. Sampling Frequency is the reciprocal of the Flickering Integration time. The maximum detectable frequency is half of the sampling frequency. The maximum detectable frequency and the detected flicker frequency are shown below the graph (Figure 49).

“FD Saturation” flags show the status of the saturation flag bit like FIFO overflow, FD trigger, FDSat_Analog, and FDSat_Digital. At a lower FD_TIME, there is more chance of getting FIFO overflow.

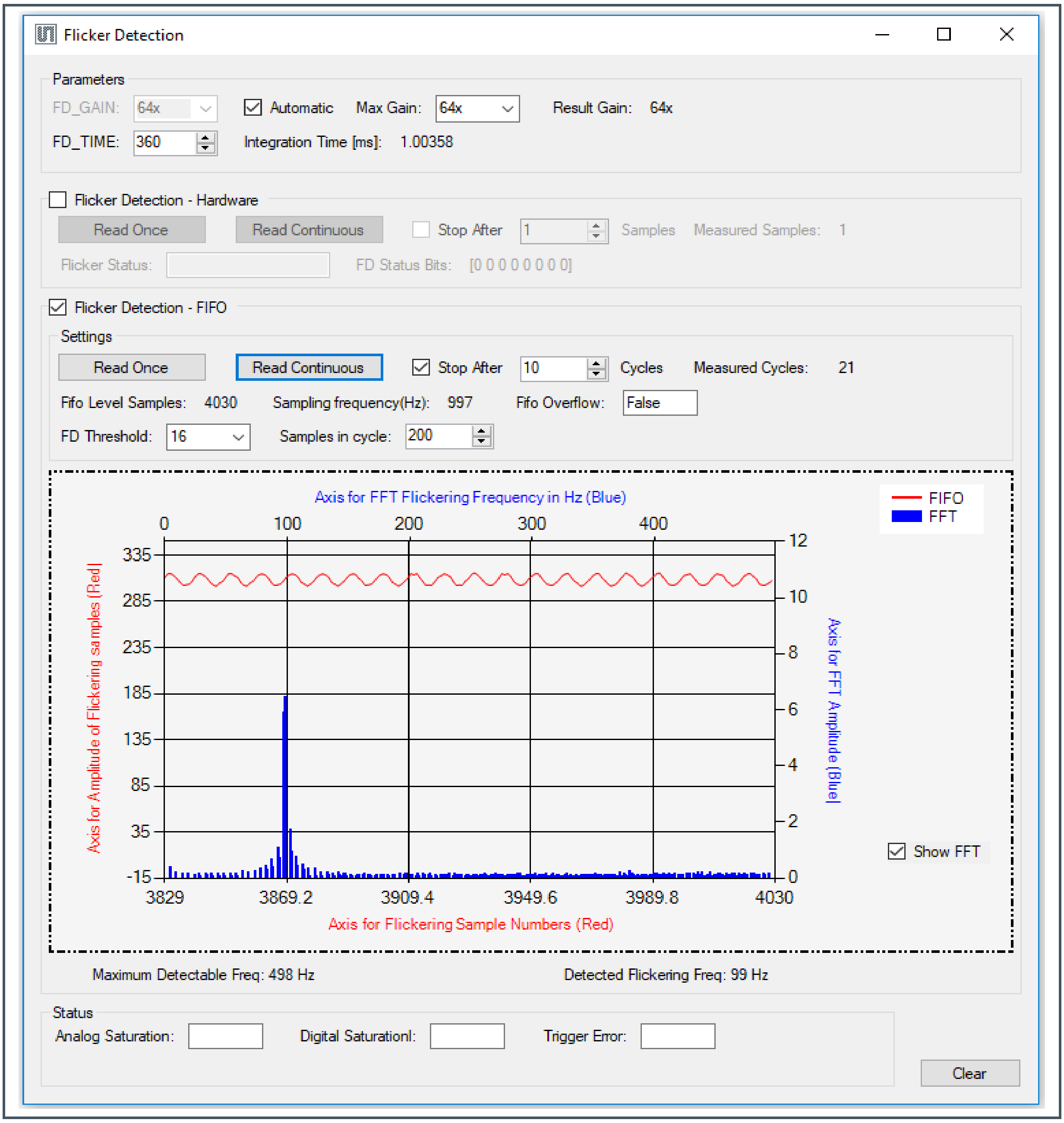
Fourier analysis converts a signal from its original domain (often used) to a representation in the frequency domain. Selecting the FFT checkbox computes the Fast Fourier Transformation and enumerates the discrete results graphically.

The FFT plot is represented in blue, with the amplitude of the FFT in the secondary Y-axis (top) and the frequency shown in the secondary X-axis (right). Figure 49 shows the results of an example. The frequency line with the highest amplitude is represented as the result of the detected flickering frequency (99 Hz in the example).

If more than one frequency is shown in a diagram with similar maximal amplitudes, then more than one flicker frequency is detected as potential results. It means the sensor results may be noisy; the counts too low, the transitions not clean enough - the flicker frequency not assignable.

The “Clear” button clears out the last measurement data and plot.

Figure 49:
Flicker Detection - FIFO



Flicker Detection

Parameters

FD_GAIN: 64x ☑ Automatic Max Gain: 64x Result Gain: 64x

FD_TIME: 360 Integration Time [ms]: 1.00358

□ Flicker Detection - Hardware

Read Once Read Continuous □ Stop After 1 Samples Measured Samples: 1

Flicker Status: FD Status Bits: [0 0 0 0 0 0 0 0]

☑ Flicker Detection - FIFO

Settings

Read Once Read Continuous ☑ Stop After 10 Cycles Measured Cycles: 21

Fifo Level Samples: 4030 Sampling frequency(Hz): 997 Fifo Overflow: False

FD Threshold: 16 Samples in cycle: 200

Axis for FFT Flickering Frequency in Hz (Blue)
0
100
200
300
400
FIFO
FFT
335
285
235
185
135
85
35
-15
12
10
8
6
4
2
0
Axis for Amplitude of Flickering samples (Red)
Axis for FFT Amplitude (Blue)
3829
3869.2
3909.4
3949.6
3989.8
4030
Axis for Flickering Sample Numbers (Red)
☑ Show FFT

Maximum Detectable Freq: 498 Hz Detected Flickering Freq: 99 Hz

Status

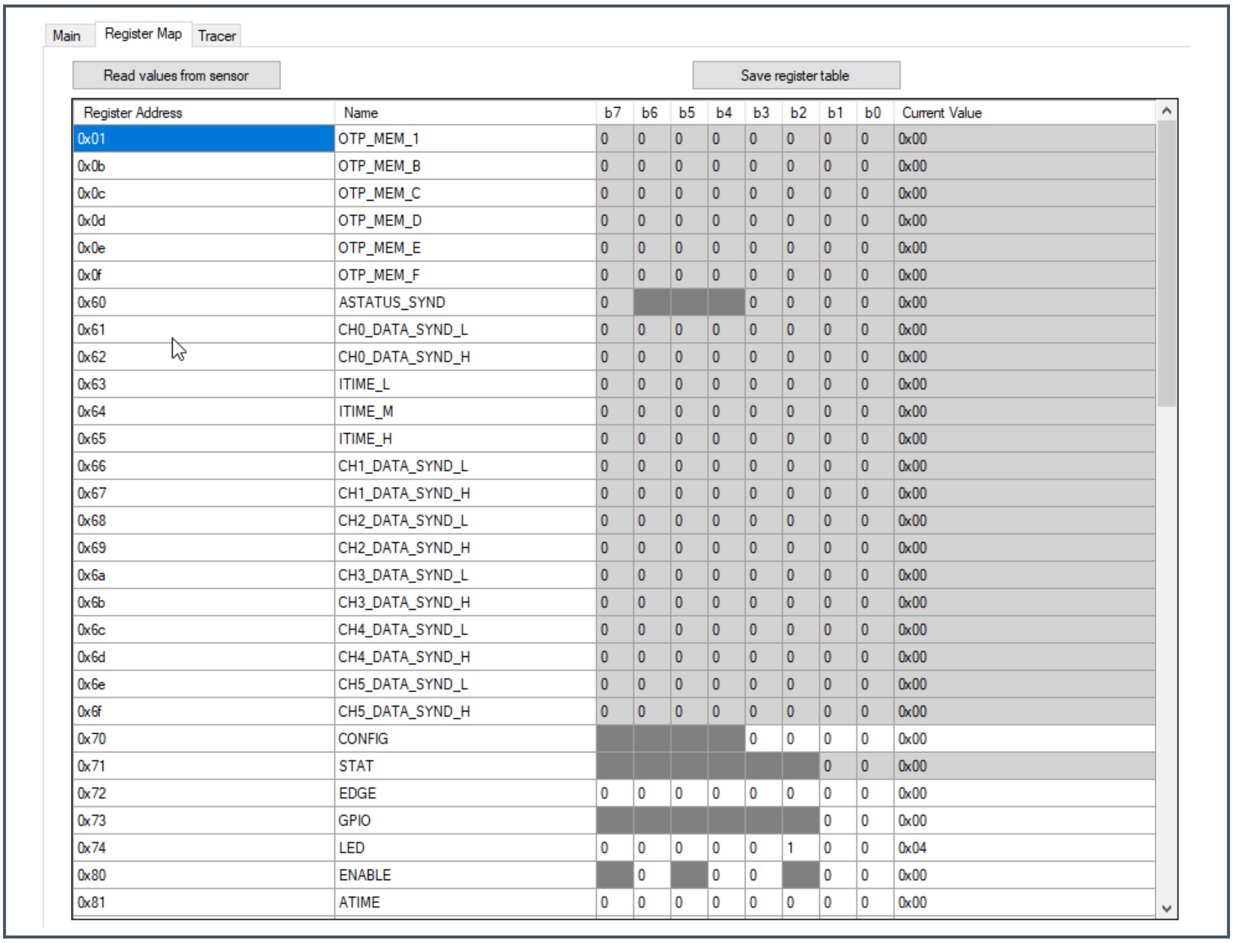
Analog Saturation: Digital Saturationl: Trigger Error:

Clear

6.16 Register Mapping

The Register table window lists and allows the writing of all I²C register values with the address, name, and information on the authorization to read or write.

Figure 50:
Register Mapping Table



Register Address	Name	b7	b6	b5	b4	b3	b2	b1	b0	Current Value
0x01	OTP_MEM_1	0	0	0	0	0	0	0	0	0x00
0x0b	OTP_MEM_B	0	0	0	0	0	0	0	0	0x00
0x0c	OTP_MEM_C	0	0	0	0	0	0	0	0	0x00
0x0d	OTP_MEM_D	0	0	0	0	0	0	0	0	0x00
0x0e	OTP_MEM_E	0	0	0	0	0	0	0	0	0x00
0x0f	OTP_MEM_F	0	0	0	0	0	0	0	0	0x00
0x60	ASTATUS_SYND	0				0	0	0	0	0x00
0x61	CH0_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x62	CH0_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x63	ITIME_L	0	0	0	0	0	0	0	0	0x00
0x64	ITIME_M	0	0	0	0	0	0	0	0	0x00
0x65	ITIME_H	0	0	0	0	0	0	0	0	0x00
0x66	CH1_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x67	CH1_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x68	CH2_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x69	CH2_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x6a	CH3_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x6b	CH3_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x6c	CH4_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x6d	CH4_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x6e	CH5_DATA_SYND_L	0	0	0	0	0	0	0	0	0x00
0x6f	CH5_DATA_SYND_H	0	0	0	0	0	0	0	0	0x00
0x70	CONFIG					0	0	0	0	0x00
0x71	STAT							0	0	0x00
0x72	EDGE	0	0	0	0	0	0	0	0	0x00
0x73	GPIO							0	0	0x00
0x74	LED	0	0	0	0	0	1	0	0	0x04
0x80	ENABLE		0		0	0		0	0	0x00
0x81	ATIME	0	0	0	0	0	0	0	0	0x00

Click the “Read values” from the sensor button to update the whole table. It is recommended to update the table when using or leaving this window. Write a value to a register by clicking the “Current Value” cell that corresponds to the register and typing a new value into the cell. The value will only be saved temporarily in the window. Use the “Save register table” to make an external copy of all the registers and the values into a CSV file.

Information

The application synchronizes the changes in the Main Window and the Register Table Page automatically.

6.17 Tracer

The Tracer[19] controls the software process by using pre-designed scripts in TXT format. Such scripts can be loaded, saved, proceeded, or cleared. “Log” is a protocol function that can be saved or cleared. Figure 52 shows the Window Tracer with an example code for a pre-designed script with “log”.

The following commands are implemented in the actual Tracer function:

Figure 51:
List of Tracer Commands with Examples

Tracer Command Syntax	Function Explanation	Usage Example
AutoGain <on, off>	Switch on/off autogain.	autoGain on autoGain off
AutoTint <on, off>	Switch on/off autoint.	autoTint on autoTint off
BackSideComp <on, off>	Switch on/off BackSideCompensation.	BackSideComp off
BeginLoop <n> …. EndLoop	The actions between BeginLoop and EndLoop will be done by the GUI n-times as specified.	BeginLoop 3 Readsamples 5 Pause 1000 EndLoop
Clearlog	Clear the measured sample log.	Clearlog
Gain <Value>	To set Gain < 0.5x, 1x, 2x, 4x, 8x, 16x, 32x, 64x, 128x, 256x, 512x >.	Gain 256
Inttime <Atime><Astep>	To set Integration time, Atime <0 - 255> and Astep <1 - 65534>.	Inttime 0 60000
Ledon <Value> <flash>	To enable the LED with a current <0-127> with/without flash mode.	Ledon 50 flash
Ledoff	To disable the LED and LED current to default value.	Ledoff
Mode <clear> <nir>	To enable or disable NIR and VIS mode <clear> <nir> : < 0, 1 > < 0, 1 >	Mode 0 0

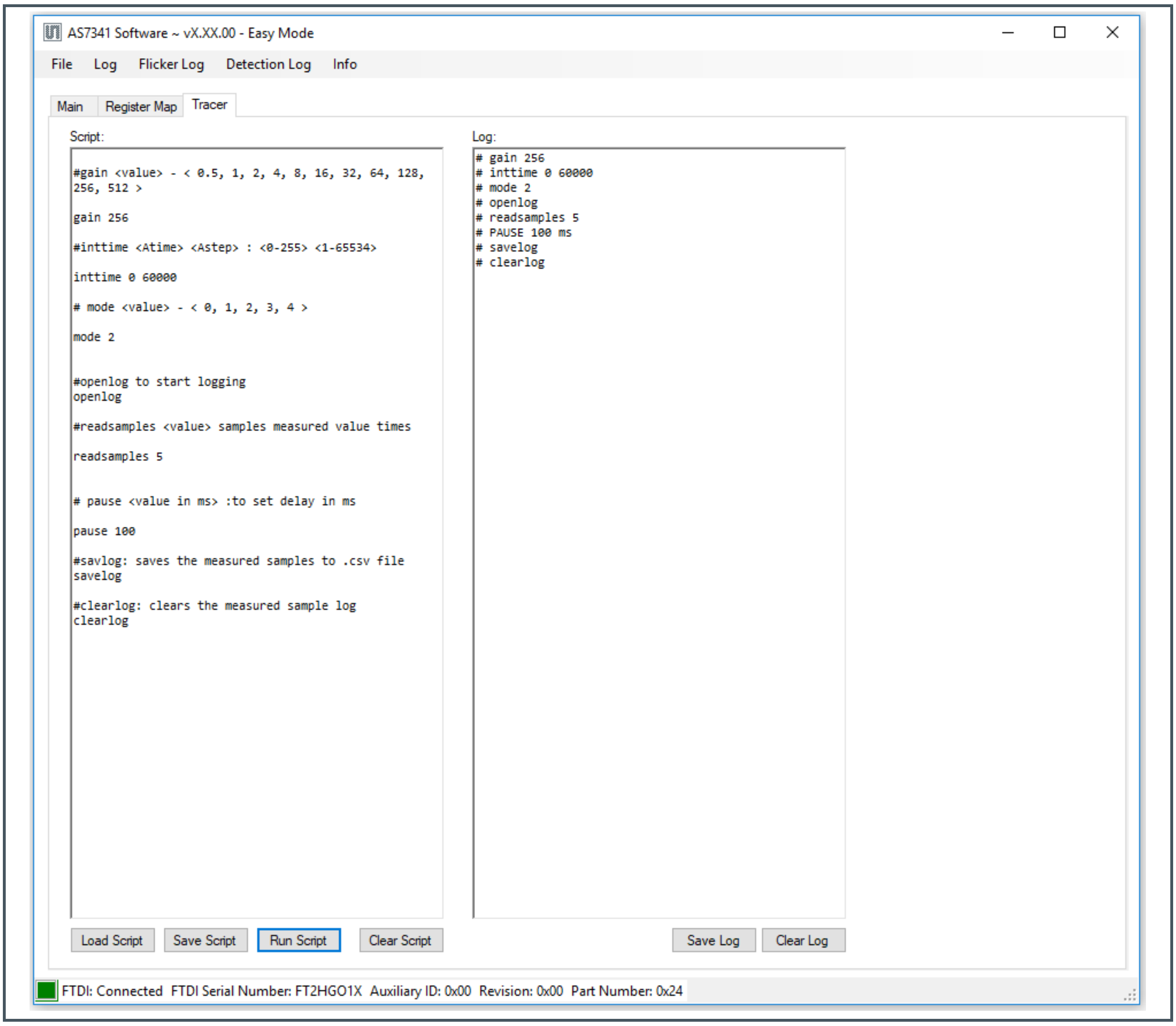
[19] See always the AS7341_script.TXT file for the newest tracer commands and examples.

Tracer Command Syntax	Function Explanation	Usage Example
OpenDialogAls	Open the ALS Function.	openDialogAls
Openlog	Open the log file.	Openlog
Pause <value in ms>	To set a delay in ms.	Pause 100
Readsamples <>	Samples measured continuously for the value times.	Readsamples 5
Readtime < time in seconds>	Read continuous measurement for given seconds.	Readtime 10
R < Register Addr>	To read out the specified register.	R 80
Savelog	Saves the measured samples to a CSV file.	Savelog
SaveOffset	Take over the last measured sensor results as a new Offset.	SaveOffset
SaveInitData	Save all the actual correction values (e.g. offset and white balance) to the disk and/or EEPROM.	SaveInitData
SetComment “txt”	TXT will be copied into the logfile as a comment for the actual measurment.	SetComment “DarlTest”
ShowMessage “txt”	Print out ‘txt” on the screen until the user confirms the message with <ok>.	showMessage ‘This is a message’
SaveWhiteBalance	Take over the last measured sensor results as White.	SaveWhiteBalance
StartLog	Start logging.	StartLog
StopLog	Stop logging.	StopLog
W < Register Addr ><Value>	To write the mentioned value to a specified register.	W 80 01
WriteLine “”	Write the “Text” on display.	writeLine "This is a line of text"
#	To comment out a line in a tracer script.	# comment
Only useful if it will be supported from the hardware.		
LedWhite60 <On/Off>	Followed by “ledon” command.	
LedWhite120 <On/Off>	Followed by “ledon” command.	
LedNir60 <On/Off>	Followed by “ledon” command.	
LedNir120 <On/Off>	Followed by “ledon” command.	
Ledleft <On/Off>	Followed by “ledon” command.	
LedTop <On/Off>	Followed by “ledon” command.	
LedBottom <On/Off>	Followed by “ledon” command.	

Information

The register address and value are always specified as hexadecimal numbers without 0x. Comments are directly behind a command or in a new line. The upper and lower case are neglected.

Figure 52:
Tracer with an Example in the Script and Log Window



Script:

```
#gain <value> - < 0.5, 1, 2, 4, 8, 16, 32, 64, 128,
256, 512 >

gain 256

#inttime <Atime> <Astep> : <0-255> <1-65534>

inttime 0 60000

# mode <value> - < 0, 1, 2, 3, 4 >

mode 2


#openlog to start logging
openlog

#readsamples <value> samples measured value times

readsamples 5


# pause <value in ms> :to set delay in ms

pause 100

#savlog: saves the measured samples to .csv file
savelog

#clearlog: clears the measured sample log
clearlog
```

Log:

```
# gain 256
# inttime 0 60000
# mode 2
# openlog
# readsamples 5
# PAUSE 100 ms
# savelog
# clearlog
```

🟩 FTDI: Connected FTDI Serial Number: FT2HGO1X Auxiliary ID: 0x00 Revision: 0x00 Part Number: 0x24

7 Error Message List

Figure 53:
Error Message List

Error Message	Explanation
Error during connecting the device: An error occurred during connecting the FTDI device.	Check the FTDI connection between the sensor and the USB port, and ensure that the FTDI cable or sensor is not faulty.
An error occurred during scanning for FTDI devices.	Check the FTDI connection between the sensor and the USB port, and ensure that the FTDI cable or sensor is not faulty.
Analog Saturation: Intensity of ambient light has exceeded the max INT level for the spectral analog circuit.	Indicates analog saturation. Reduce the light intensity, Integration time, or Gain to overcome the saturation.
Digital Saturation: The maximum counter value has been reached.	Indicates digital saturation. Reduce the light intensity, Integration time, or Gain to overcome the saturation. The maximum value range 65635 is reached.
Digital and analog saturation has been reached.	Indicates both digital and analog saturation are reached. Reduce the light intensity, Integration time, or Gain to overcome the saturation.
Status SP_TRIG Error: WTIME is too short for the selected TINT	Spectral Trigger Error: This indicates that there is a timing error. WTIME is too short for the selected ATIME. Increase WTIME to overcome this issue.
Error during reading the register value.	While reading the register value from the sensor an error has occurred.
Error during setting power on.	Error occurred. Set the PON bit in the Enable register (0x80). Disconnect the device and try again.
Error during enabling or disabling the spectral measurement.	Error whilst enabling or disabling the spectral. Enable bit in Enable register (0x80).
An error occurred during enabling or disabling external LED.	Setting the LED current produces an error.
Status Over Temperature Detected: Device temperature is too high.	Over temperature detected. The device temperature is too high.
Cannot detect the optimal gain. Please change the settings.	The optimal gain was out of range. Change the settings for optimal gain.
Unequal number of correction factor in Init_file.txt	The number entered for the correction factor should be crosschecked in the “Init_file”.

Error Message	Explanation
Error reading light detection light source matrix file:<file location > Input string was not in a correct format	Crosscheck the corresponding file data. The data entered may not be in the correct format or mismatching in the index.
File opened error	Another process opens the log file to which the data is saved. Do not open the file between measurement logging.
The Init_file was not found	Ensure that the Init_file is in the directory or with the correct name, where the software files are installed as mentioned in chapter 6.2.
Error opening the Init_file	Ensure that the Init_file is closed while operating the GUI. The changes made to the Init_file should be saved and the software should be reopened to view the effect of the changes.
Error reading the Init_file	When the parameters in the Init_file are incorrectly named, no value is provided for a parameter, commented out, or not in the correct format and dimension, the corresponding parameter is selected from the default values. The status box will display this information with a blue status message.
Error opening <filename><file location> because it is used by another process	File not found or save the changes made to the file and close it. To see the effects of the change, close and reopen the GUI. Do not open the file while the software is running.
Measured color coordinate out of range	Error in the calibration, correction, or measurement. The measured sensor results are not valid in the CIE1931 Color space. Check the setup, init_files, and calibration.
CCT out of range	During light detection, if the CCT calculated is out of range from the value defined in the “Init_file.txt”.
Calculated delta uv greater than given delta uv	During light detection, if the delta uv calculated is greater than the given Delta uv in the “Init_file.txt”.
u and v not in limit	During light detection, if the u and v calculated is not in the limit, which is in the “Init_file.txt”.
An error occurred during writing the SMUX configuration for FIFO flicker detection	There was an error while configuring the SMUX for flicker detection.
An error occurred during enabling or disabling flicker detection	An error occurred during the Fden bit in the Enable register (0x80).
No samples were logged	Take measurements for the process or save data to a log file.

Error Message	Explanation
An overflow detected. Data has been lost in continuous measurement	Warning - An overflow is detected. Data has been lost in continuous measurement.
Mask number is too big or Maximum number of mask entries reached	Whilst writing the mask files to the EEPROM, the maximum limit of entry is reached.
Mask file contains wavelength that are not in calibration matrix file	Warning of an unequal wavelength. Could use the mask compare even though they have different wavelengths. However, each will overwrite the old mask file with the new wavelength entry.
Header Item: Wavelength or title was not found	In the calibration file, the wavelength is considered as the starting point for the calibration matrix, and title is considered as the starting point for the reference file. If these are wrongly typed, the GUI generates an error.
Unexpected line detected	When calibration or reference files do not have the expected number of values or errors in the entry format. An empty matrix is considered in this case.

8 Other Devices Using the GUI

The GUI further supports special forms of the AS7341 sensors as a Demo and EVK. Unless otherwise stated, the details described in this document also apply to these devices.

The sensor variant, AS7341 PEN, implements an EEPROM to save all initialization, calibration, and mask files directly on the sensor device. PEN is the first AS7341 device with an EEPROM. Other devices will follow. In this case, all details are also valid for these devices.

All Init files for the AS7341 GUI must be loaded onto the EEPROM before being used for the first time. A direct change of the data on the EEPROM (excluding Spectral Masks) is not possible. It can only be done by a readout from the “EEPROM on disk”, making a change in the directory by clicking Edit, and rewriting to the EEPROM using the GUI.

In the AS7341 PEN, the EEPROM includes all the data files for corrections and GUI control. The maximum for spectral mask entries for light is 32, and the maximum spectral entries for reflection are 500. Therefore, when writing the mask files from the directory to the EEPROM, the mask data present in the first 32 columns in the light mask compare file and 500 columns in the reflection mask file will be copied into the EEPROM.

To teach or add more mask entries into the PEN, limit the entry made during the writing process to the EEPROM. For example, if the light mask file contains ten entries of the spectral mask data, after writing this file to the EEPROM, the EEPROM will still have the possibility to add twenty-two spectral mask data in the case of lighting. Similarly, in the case of reflection, if fifty entries were in the spectral mask for the reflection file, after loading this data to the EEPROM, the EEPROM will still have 450 entry spaces. Once the maximum number of entries is reached in the PEN, an error message is generated upon adding new entries. This will restrict any new entry to the EEPROM.

The EEPROM can be disabled to use the Initialization files from the directory. Afterwards, start the GUI in order to be able to use the EEPROM again. Figure 54 lists all the EEPROM commands.

Figure 54:
EEPROM Commands for an EEPROM-Based AS7341 EVK and Demo

Command	Description
Ctrl + D	Disable EEPROM. GUI uses data directly from the directory.
Ctrl + E	Initialize or rewrite EEPROM with the files from the directory.
Ctrl + F	EEPROM clear.
Ctrl + I	Write initialization data from memory to the EEPROM or initialization file.
Ctrl + W	Readout Initialization files from the EEPROM to the directory.

9 Additional Documents

The following list includes a selection of additional documents with more technical details for the Sensor and AS7341 Evaluation Kit. This list is not fixed and it is constantly changing. Ask us for new details.

For further information, please refer to the following documents:

1. ams-OSRAM AG, *AS7341 11-Channel Multi-Spectral Digital Sensor (DS000504)*, Datasheet.
2. ams-OSRAM AG, *AS7341 Details for Opto-Mechanical Design*, Application Note.
3. ams-OSRAM AG, *AS7341 EVAL KIT Flicker Detection (AN000605)*, Application Note.
4. ams-OSRAM AG, *SMUX Configuration (AN000666)*, Application Note.
5. ams-OSRAM AG, *Schematic a0013a_CSS Evalboard AS7341*, Reference Design.
6. ams-OSRAM AG, *AS734x Eval Kit –Spectral Balance and Calibration (QG000139)*, Quick Start Guide.
7. ams-OSRAM AG, *AS7341 Demo for Fast Measurement Using Unicom Board (AN000660)*, Application Note.

10 Revision Information

Changes from previous version to current revision v6-00	Page
all	
Removed AS7342 and implementation of the new software functions.	

- Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
- Correction of typographical errors is not explicitly mentioned.