Product Document

Published by
ams OSRAM Group

User Guide

UG001037

AS7331 EVK Logger

AS7331 Evaluation Kit

GUI Version Data Logger

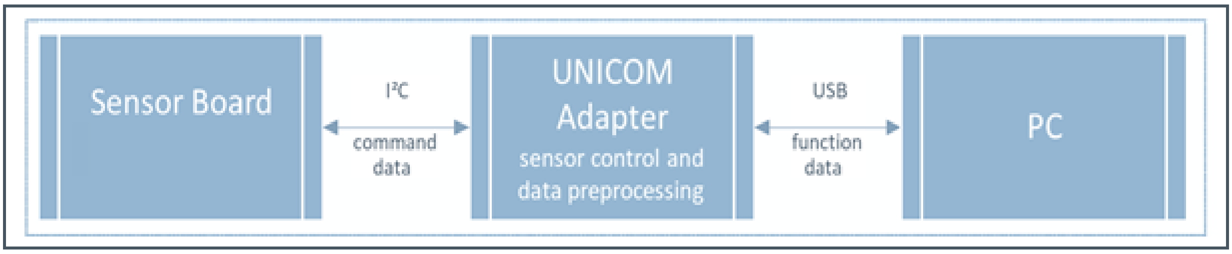
v3-00 • 2022-Nov-15

Content Guide

1 Introduction

The AS7331 EVK Logger allows, the testing of the UVA-UVC sensitized photodiodes and the ADC conversion of the photocurrents, as well as a digital I²C interface and temperature sensors on-chip. The IC sensor is described in the datasheet [1].

Figure 1:
System Solution for the AS7331 EVK Logger



The AS7331 EVK Logger consists of a sensor board with a plugged optical adapter above the sensor and LEDs[1] and an attached interface board. The interface board is connected to the PC via USB, which supplies all the connected hardware with a power supply. The adapter realizes the optical requirements for the used interference filters on the chip, as well as the optical path when using LEDs on the board.

The EVK can only be tested successfully under the conditions for which it is prepared. The optical coupling plays an essential role, as well as the light situation in front of the sensor. The optical coupling in front of the sensor decides what and under which conditions the sensor sees something. It also considers whether the conversion of the optical signal into digital results is possible with the dynamics of the sensor. Further details are given in chapters 6 and 8.2.

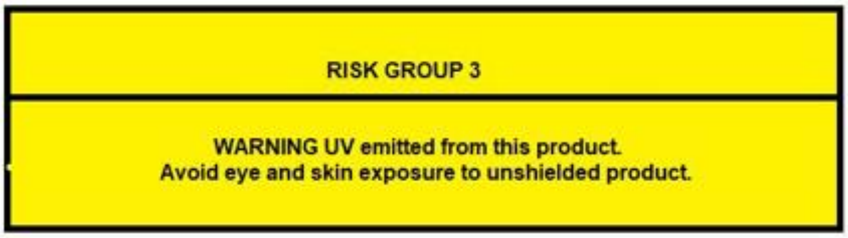
[1] Generally unassembled and must be soldered customer-specifically.

2 Safety Notifications

CAUTION

- This product emits UV light, when UV-LEDs are mounted on the board.
- Avoid exposure to the skin and eyes, when UV-LEDs are mounted on the board.

UV – RISC Group 3



There are delivery forms where an EVK on the sensor board can be equipped with harmful UV LEDs of risk group 3. The UV LEDs are partially covered by optical adapters when mounted. The LEDs can be switched on in the EVK software or through firmware and register programming.

CAUTION

For the switched-on state of the UV LEDs, the following safety warnings for the eye and skin apply:

1. UV radiation hazard.
2. Use only with shielding in place.
3. Protect the eyes and skin from exposure to UV light.

3 What Is In the Box

Figure 2 shows a typical scope of delivery for the AS7331 EVK Logger with a sensor, and interface hardware in a plastic housing, with a mounted adapter and USB stick. If the hardware is provided as individual modules, please assemble them completely (see chapter 4).

The USB stick contains all the documents for the sensor and hardware, application notes, firmware, the GUI setup, possible libraries, and other details for customers.

Figure 2
Kit Content of the AS7331 EVK Logger

Item 1: Sensor Hardware and Interface in Plastic Housing with Front Adapter	Item 2: USB Stick
	

Item No.	Item	Comment
1	990601135	AS7331 EVK Logger(1)
2	USB Stick	Documents, software, firmware, and drivers (2)

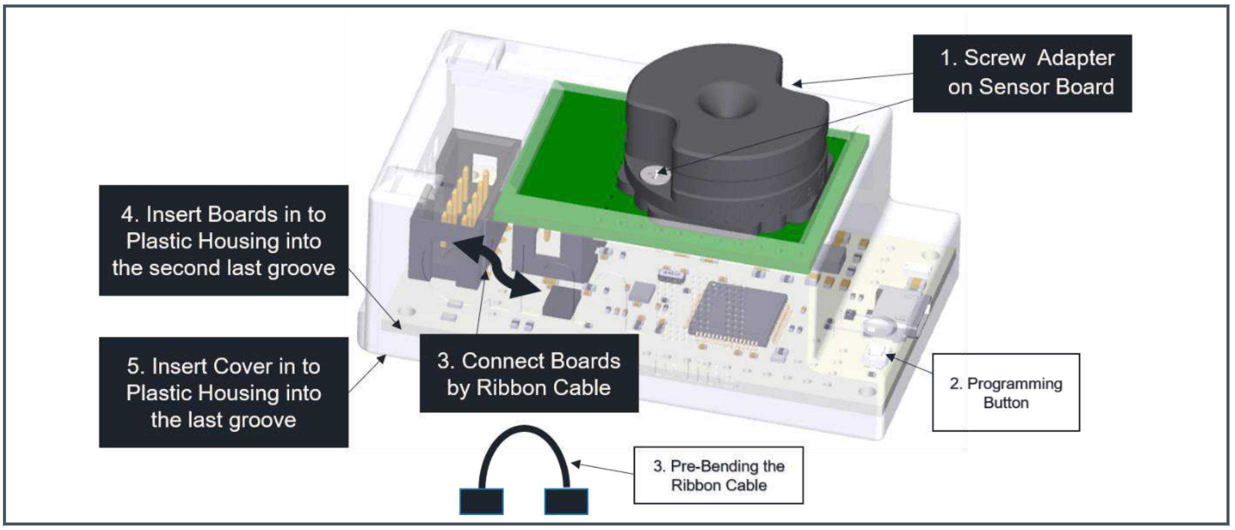
(1) This can be delivered as individual components (Sensor Board, Interface Board, ribbon cable, optical adapter, housing base, and cover). The screws, nuts, and cable can be omitted. Please use a standard USB cable in this case. The sensor and Interface board can also be delivered as single variants with a ribbon cable and an alternative or without an optical adapter.

(2) The version of the documents may be old. Therefore, we always recommend downloading the latest USB stick as a ZIP variant from the ams website.

4 Device Assembly

If the hardware is provided as individual modules, please assemble them completely before commissioning.

Figure 3:
Hardware Assembly



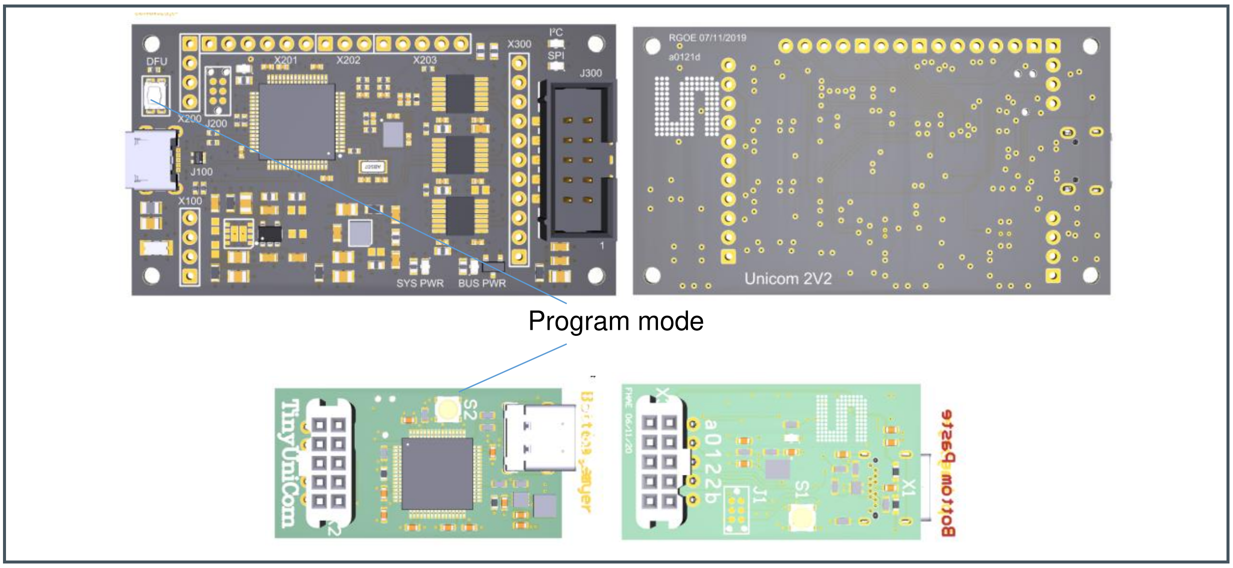
1. Firstly, screw the adapter onto the sensor board. Please note the different holes for the screws and the guide of the adapter on the sensor board. Afterward, tighten the screws.
2. Instructions for programming the interface board (if necessary)[2] are stated in this user manual (refer to chapter 7). Uploading the firmware can only be done via the hardware outside the plastic packaging, since the card can be put into programming mode by pressing a button on the card during the USB connection.
3. Next, connect the sensor board to the Unicom interface board with the ribbon cable. Then, bend the cable slightly forward like a sling, to relieve pressure at the sockets when sliding the connected boards into the case.
4. Afterward, insert the Unicom (USB connector at the front) and the sensor board into the penultimate groove of the plastic housing.
5. Finally, connect the completed EVK via USB to the PC and start the software. More details are given in this EVK manual (refer to chapter 8).

[2] Usually, the Unicom contains the necessary firmware upon delivery. If that is not the case, or a new firmware version is to be flashed, please refer to chapter 7.

5 Hardware

There is hardware for the AS7331 sensor and the Unicom Interface board[3]. Both boards are connected by a cable. The AS7331 sensor board contains the AS7331 sensor chip and digital temperature sensors, next to the free solder joints for the LEDs that serve as the board lights for direct reflective illumination. Figure 7 shows the schematic of the sensor board with the individual components and the interface.

Figure 4:
Front and Bottom Views with the S1 Push Button on the Unicom (Standard and Tiny version)



The Unicom controls the sensor and logs sensor data before transferring it as data packets. The board is derived from the STM32F413 microcontroller, which provides a generic interface such as USB to I²C, SPI, UART, etc. The I²C bus interface supports Standard-mode (Sm), Fast-mode (Fm), and Fast-mode Plus (Fm+). The firmware is configured, coded, and uploaded for a Fast-mode (which supports up to a 400 kHz clock frequency) connection with the AS7331 sensor board (part of the AS7331 EVK). Usually, the Unicom contains the necessary firmware upon delivery. If the Unicom requires new firmware, carefully remove the board from the housing. To set the Unicom to program mode, use the S1 Push Button (see Figure 4, program mode). The process is described in chapter 7.

Regardless of the Unicom version, the board connection between the sensor and the Unicom board, is the connection between the Unicom and the PC via USB. An intact and connected AS7331 EVK Logger is displayed as a virtual COM port in the Device Manager or the GUI after USB connection and software installation.

[3] The Unicom is available in different forms as a Standard board with extended interfaces and a Tiny version characterized by a lower size and reduced interfaces.

Figure 5:
Block Schematic of the Sensor Board

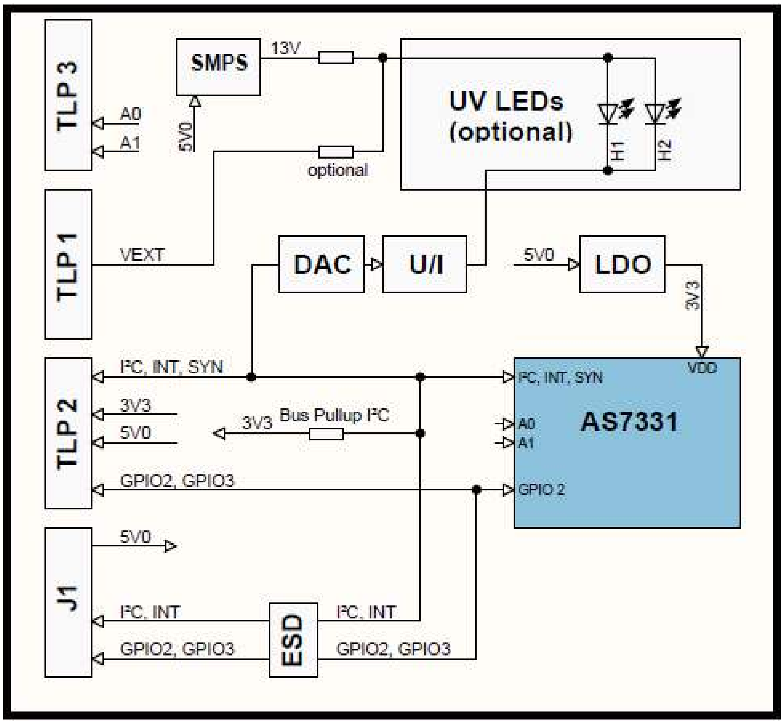


Figure 6:
Block Schematic of the Unicom Interface Board

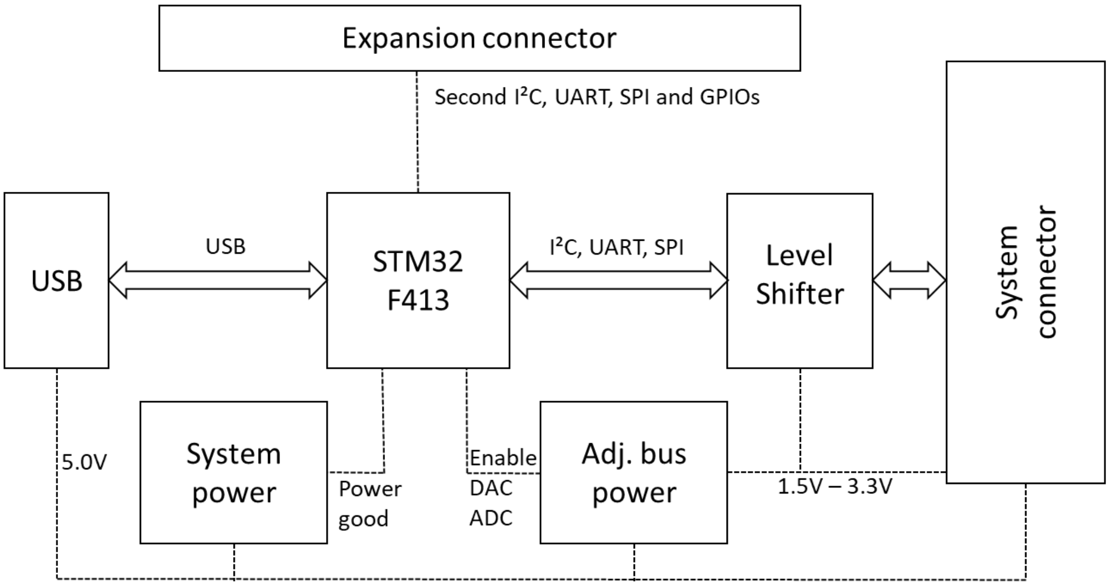


Figure 7:
Schematic and Interface of the AS7331 Sensor Board

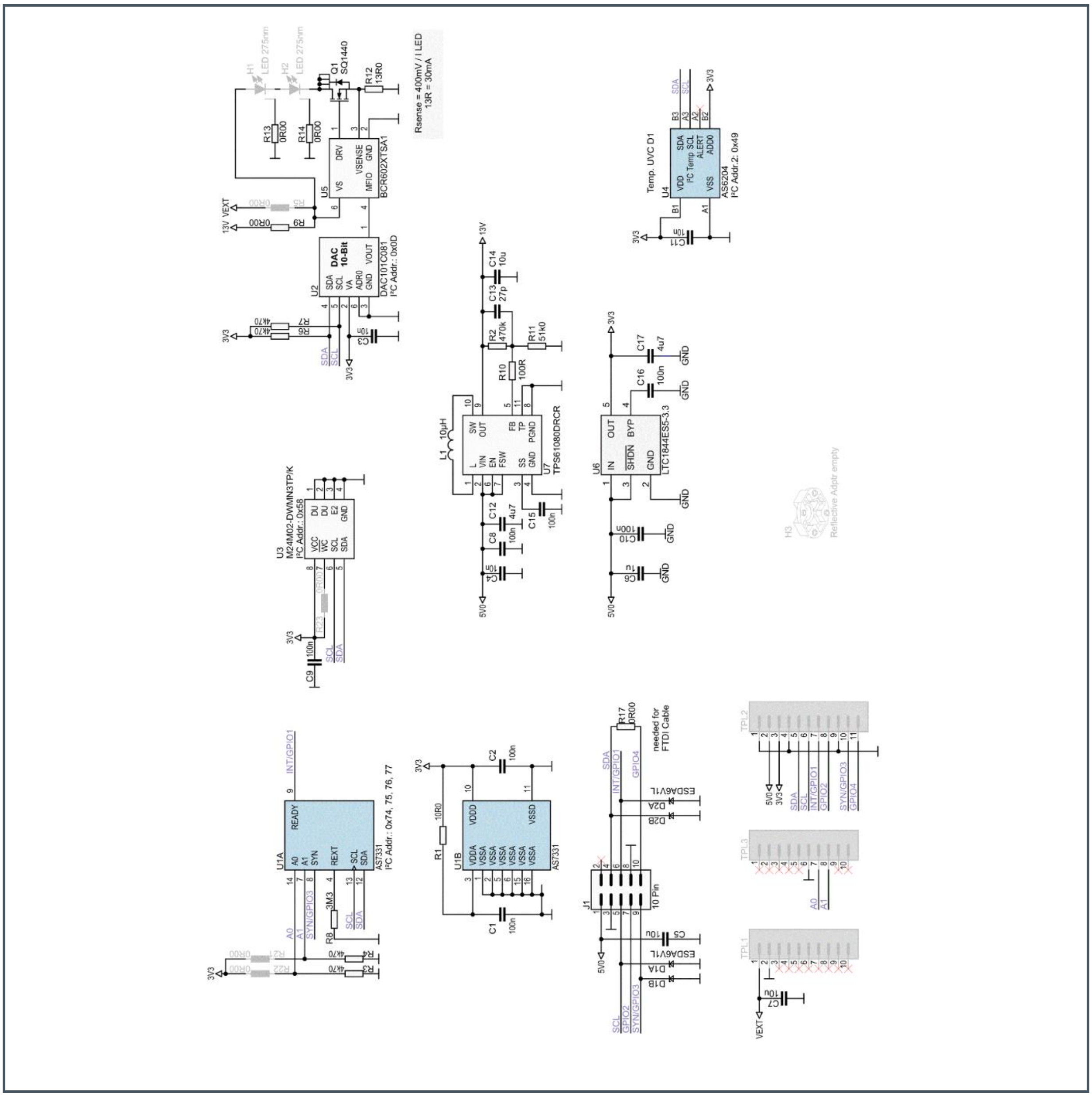


Figure 8:
Schematic and Interface of the Unicom Interface Board I

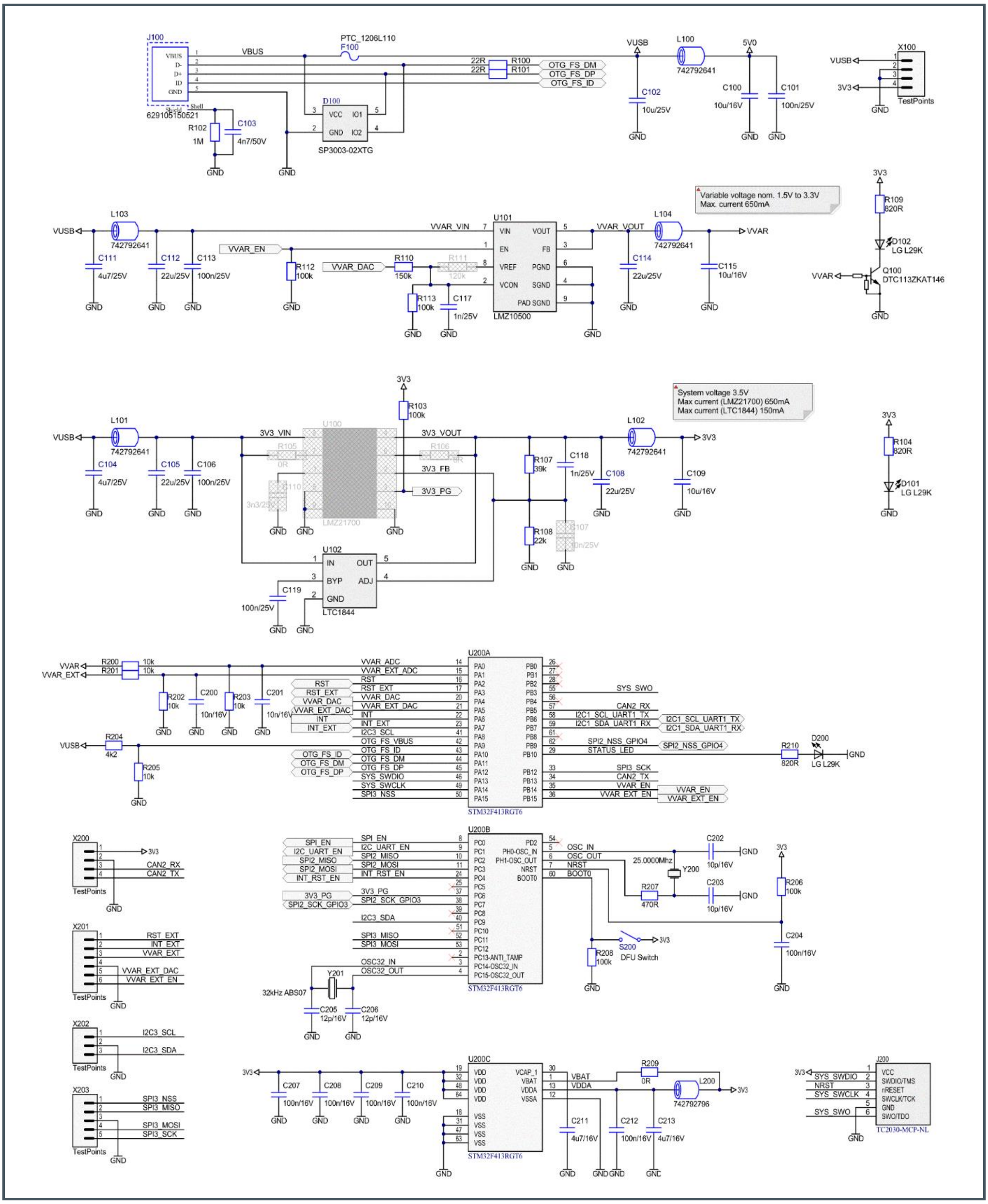
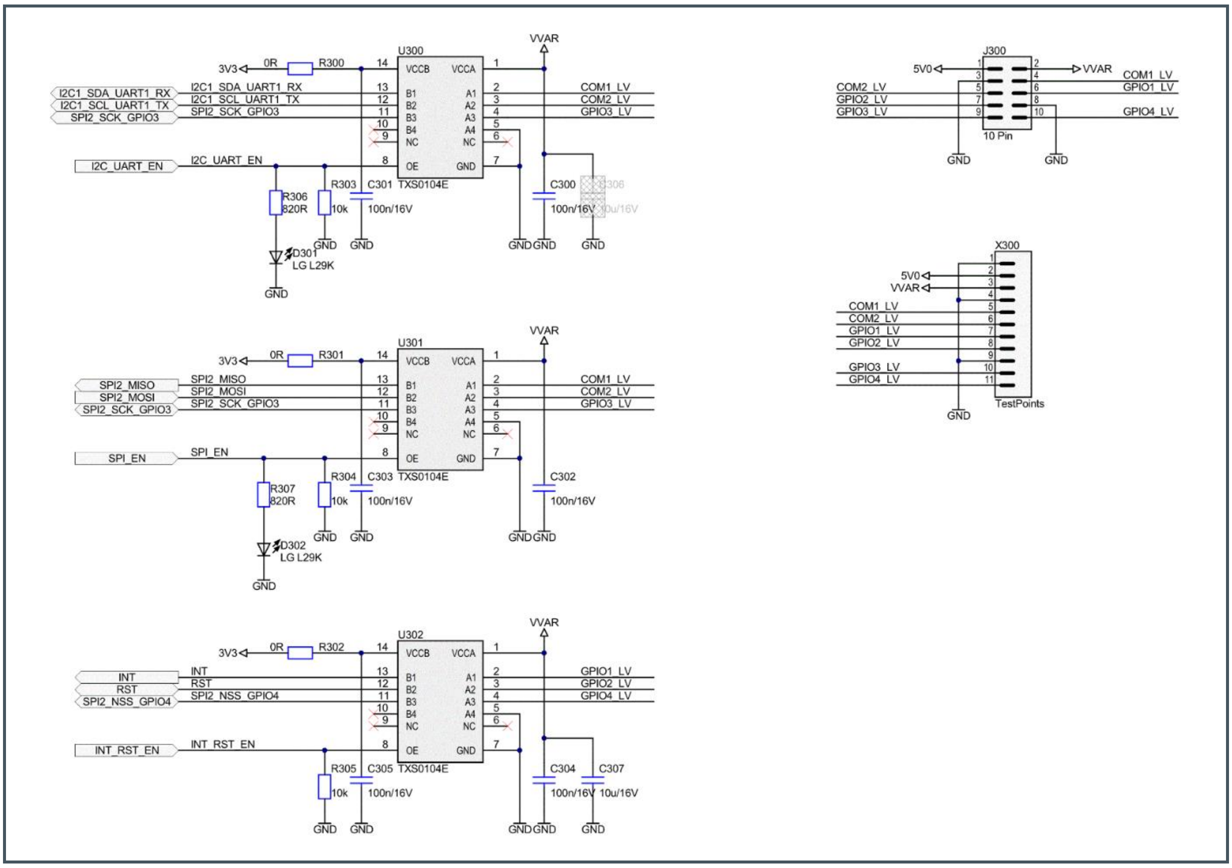


Figure 9:
Schematic and Interface of the Unicom Interface Board II



6 Optics

In the case of reflection, the optical adapter realizes a 0°/45° pinhole optics (45° LEDs, 0° sensor view). For applications without the LED light source, the adapter realizes the necessary limitation of the incidence angle of ±10°, which is important for the fulfillment of the filter function according to the datasheet [1]. Opening the angle of incidence beyond the allowed 10° will widen and distort the bandpass filters of the sensor channels, such that the sensor effect will no longer meet the specified values.

Attention

The adapter was carried out of a material that is not fully UV-tight. Therefore, the supplied adapter does not convert the required angle of incidence exactly and is only a design example. For full implementation and to achieve higher accuracy, the design must be made with a UV-proof material, such as metal.

A UV-compatible diffuser in front of the sensor and adapter should be used in case of a non-diffuse application.

In most applications, the optical adapter must be adapted to the application, such that the adapter included in the kit will only be regarded as an example, in principle. Due to its limited angle of ±10°, the EVK is very sensitive to angular changes in the direction of the light source if it is not homogeneous in its radiation. If necessary, a solution must be found by using diffusers.

Figure 10:
Optical Adapter on the Board Realizes an Angle of Incidence of ±10° and 0°/45° Optics

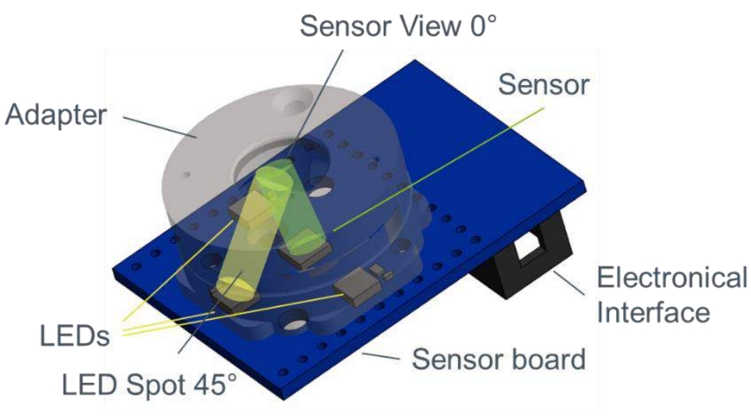
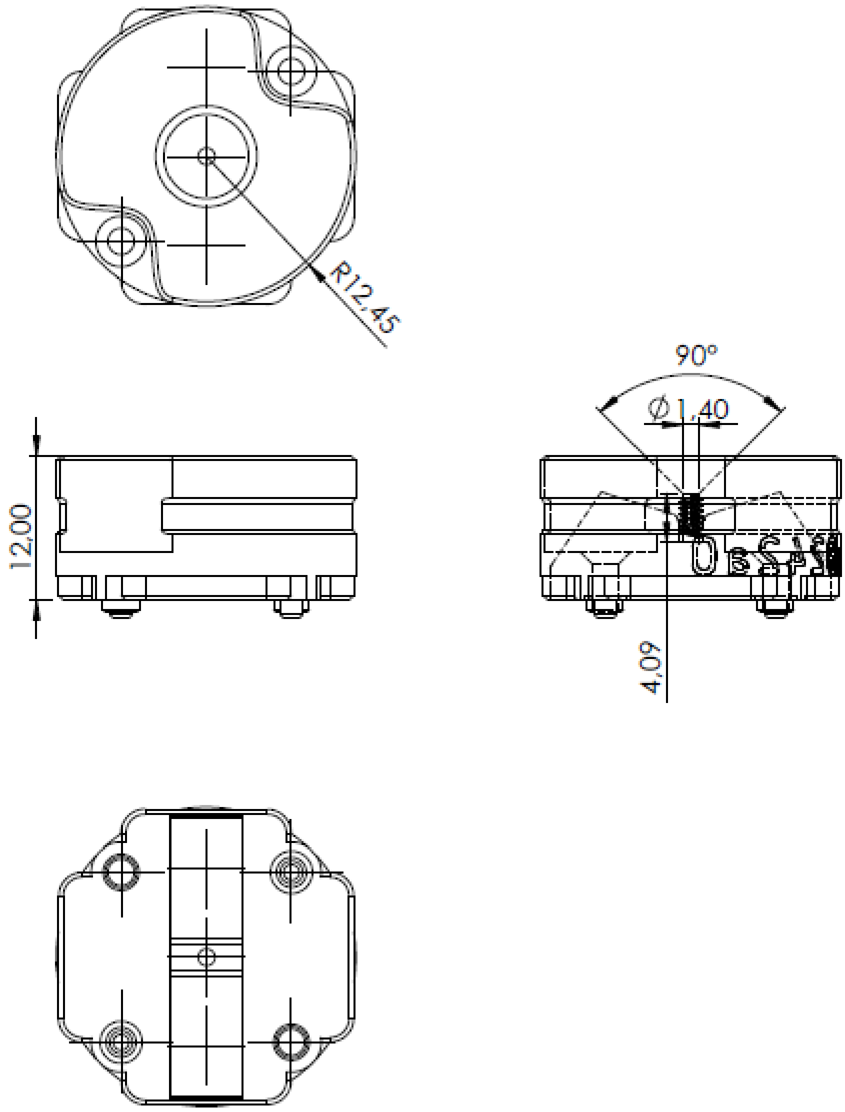


Figure 11:
CAD Models of the Used Adapter are Deliverable



(1) Dimensions in mm.

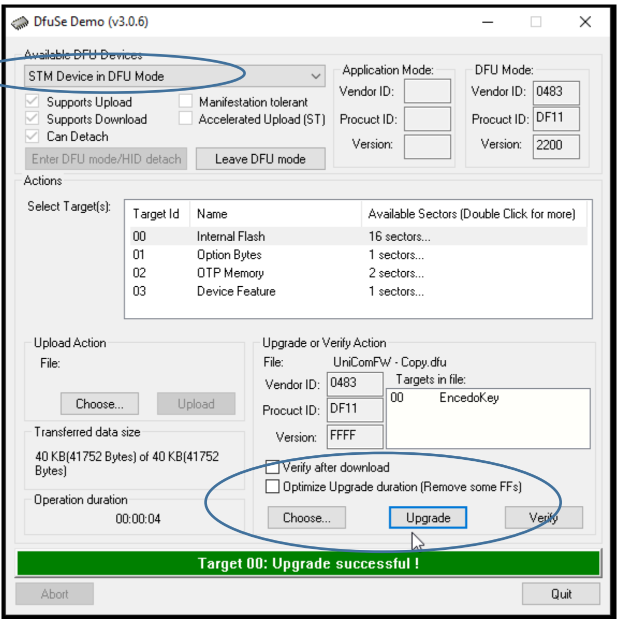
7 Firmware

The Unicom interface board includes the required firmware for the AS7331 EVK Logger. Firmware version v0.1.0 or higher is required. In case an upgrade or upload is necessary, please follow these steps with the DfuSe Demo:

1. Install DfuSeDemo.
2. Press and hold the DFU button to set the Unicom board to program mode (see Figure 4) while connecting the Unicom board (without the sensor board) directly to a laptop (no hub) via USB. Once connected, release the button.
3. Start "DfuSeDemo". If the Unicom board is in the correct mode, you should see "STM Device in DFU Mode"[4] as an available DFU-Device (see Figure 12).
4. Click on the "Choose" button and select the desired DFU file. Click on the "Upgrade" button to start the upload process. Within seconds, you should see a message from the software: "Upgrade successful".

More details for the software are described in the DfuSeDemo manual "*UM4012.pdf*", which is installed in the Windows Start Directory "STM32 directory".

Figure 12:
DfuSe Demo Software – Main Window After a Successful Firmware Upgrade



[4] Check and change the STM driver "STM32 Bootloader", with "STM device in DFU mode" in the Device Manager if: (A) no available DFU device is shown in the DFU multi-select window, (B) "STM32 Bootloader" is activated as a current USB device in the Device Manager, and (C) it is possible to uninstall the driver under the selected properties to set the new driver.

Alternatively, you can also use “STM32CubeProgrammer”. In this case, perform the following steps:

1. Install STM32CubeProgrammer.
2. Press and hold the DFU button to set the Unicom board to program mode (see Figure 4) while connecting the Unicom board (without the sensor board) directly to a laptop (no hub) via USB. Once connected, release the button.
3. Start “STM32CubeProgrammer”. If the Unicom board is in the correct mode, you can connect the USB device (number 3 marked in orange in Figure 13).
4. Select “Erase and Programming” by clicking on its button (see Figure 13 - Number 4).
5. Delete EEPROM by clicking on the “Full Chip Erase” button (see Figure 14 - Number 5).
6. Select the latest firmware update for the AS7331 *.hex and start the Programming (see Figure 14 - Number 6 and Number 7).
7. Within seconds, you should see a message from the software: “Upgrade successful”, “Connection is lost”, and other messages about the upload process. Confirm the messages, and disconnect the USB after all the confirmations and the closing of the Programmer. Then, the hardware can be reconnected, and the EVK will be ready to be upgraded for the GUI.

Figure 13:
STM32CubeProgrammer – Main Window After Starting

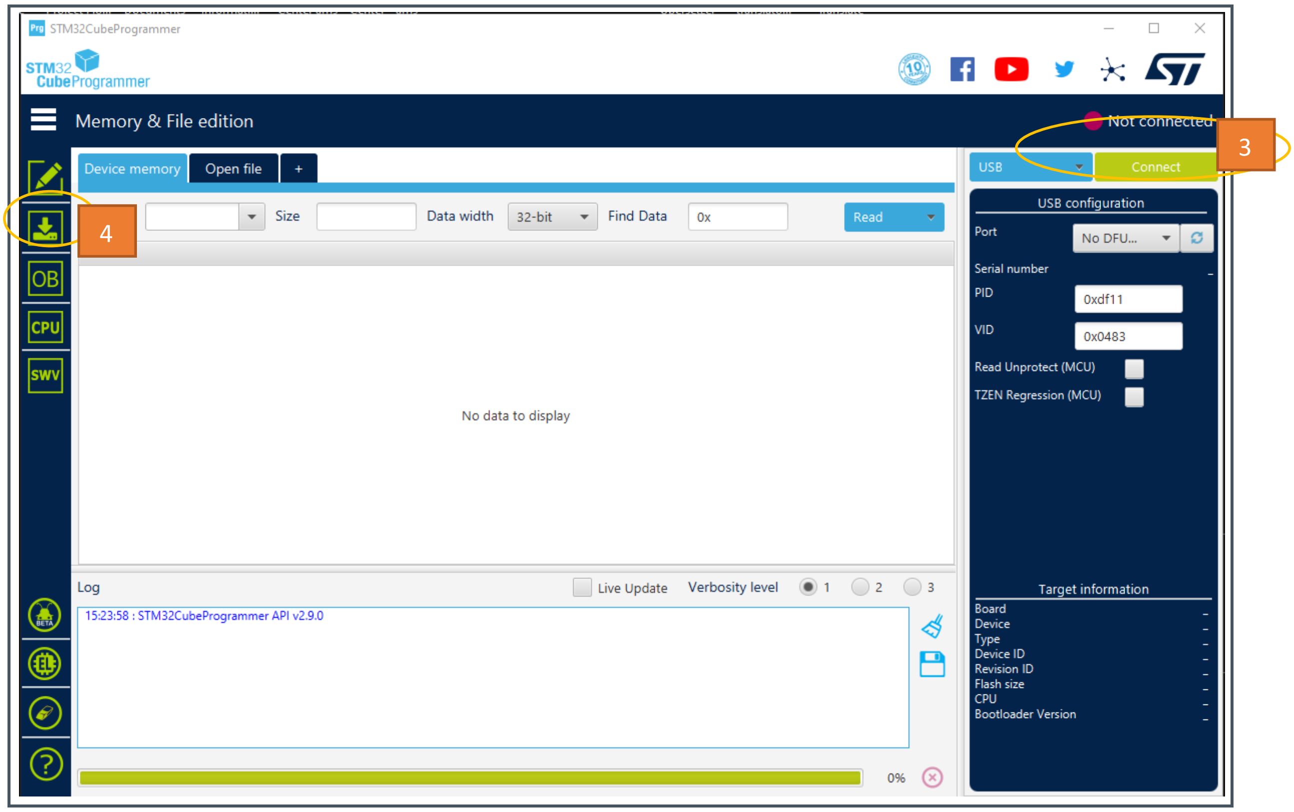
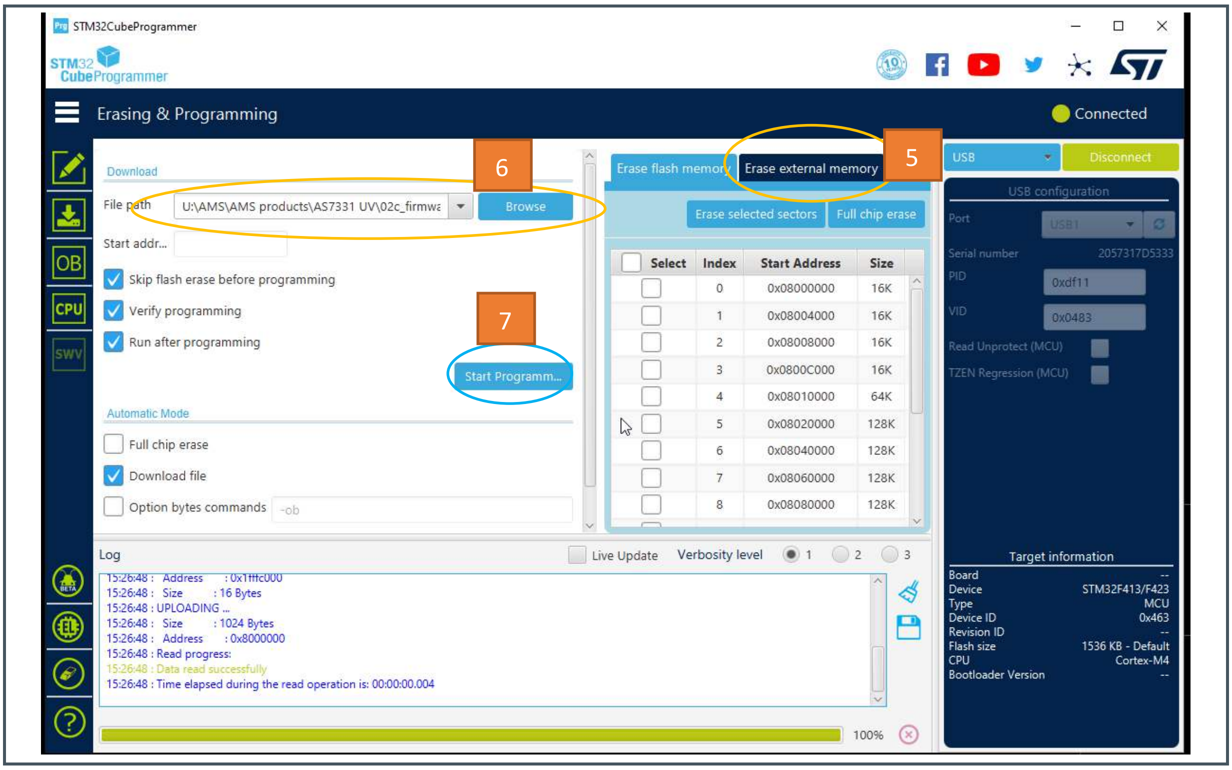


Figure 14:
STM32CubeProgrammer – Window for Erasing and Programming



8 Software

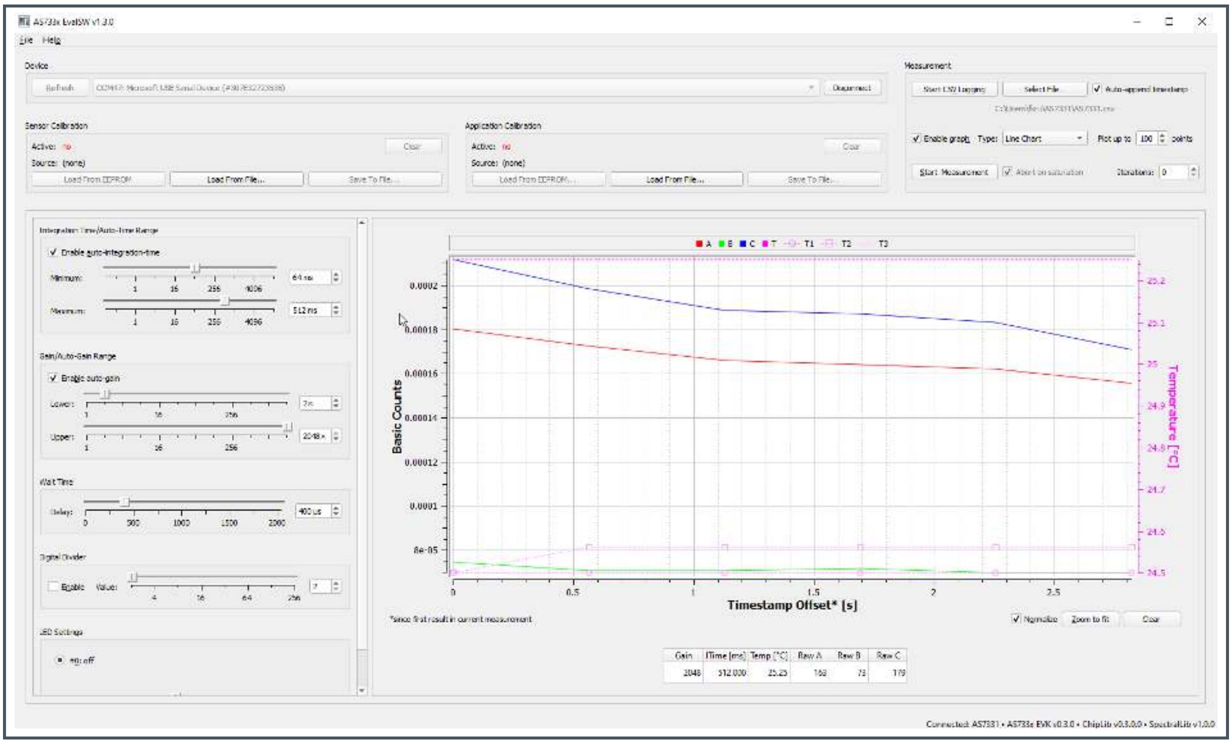
ams OSRAM provides a GUI application for the AS7331 EVK Logger called “AS7331 Evaluation Software”. This software is available as a Windows installer and a Linux (amd64) AppImage.

To install the application on a Windows system[5], run the installer, “*AS7331_EvalSW-1.0.0-win64.exe*”, from the USB stick with administrator privileges, and follow the instructions. This will create the application in your start menu, where it can be launched.

For Linux systems, the included AppImage should be compatible with most modern distributions. Simply run the “*.AppImage*” file.

Launching the application will give you a user interface such as the one shown in Figure 15. This window allows the user to connect to an attached device, configure, run a measurement, and output the results on a chart[6] and/or to a CSV log file.

Figure 15:
Screenshot of the Application’s Main Window

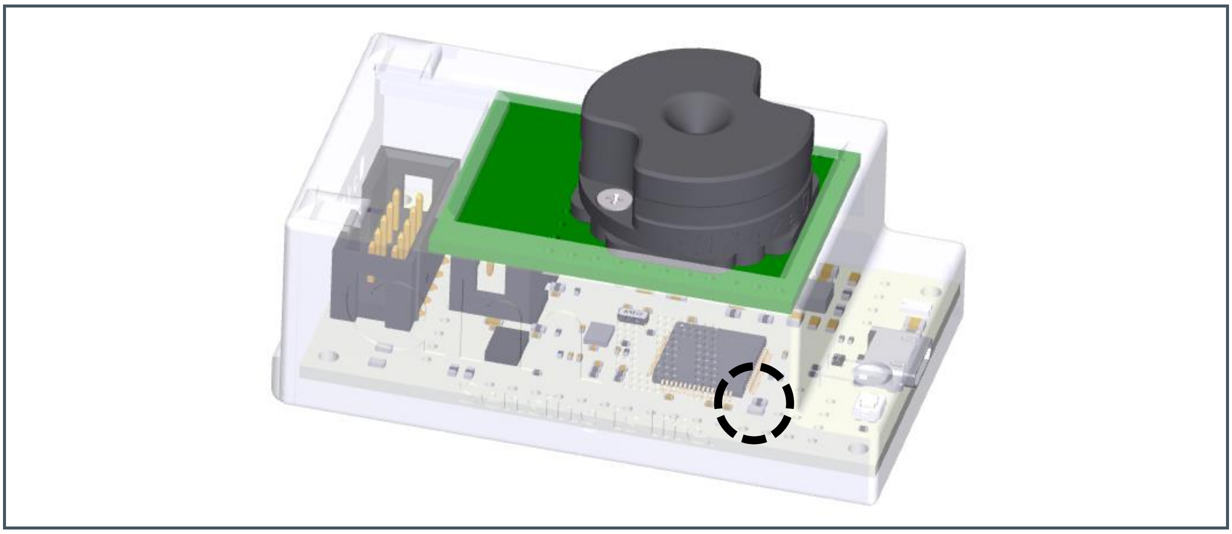


[5] Windows 10 and higher
[6] Higher GUI versions than 1.0 consider line and bar charts, which can be selected in a multiple-choice window next to the chart activation.

8.1 Connecting to a Device

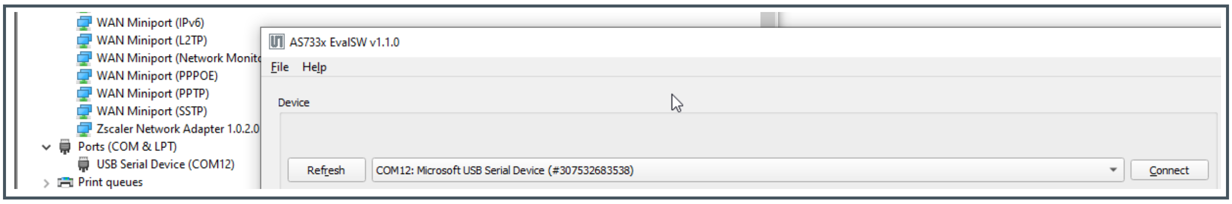
To connect an AS7331 EVK Logger device, connect it via USB to your computer running the application. A green LED on the Unicom board switches on after connecting the EVK to the PC via USB.

Figure 16:
LED (marked in the figure) Should Switch on after USB Connection of the EVK to a PC



In the case of a connection error to the kit, check the status of the active COM ports (with and without the USB connection of the kit to the PC) in the device manager and check the cable and USB connector in the case of an error.

Figure 17:
COM Port in the Device Manager Displays the List of Available Devices



When it starts, the device should automatically appear in the drop-down list at the top left. Otherwise, you may need to refresh the list of available devices using the “Refresh” button to the left-hand side of the box.

Once you have selected a device, press the “Connect” button. This should establish a connection to that device, and the button should turn into a “Disconnect” button - which allows you to end the connection if you wish. If the connection fails for any reason, an error dialog will be displayed.

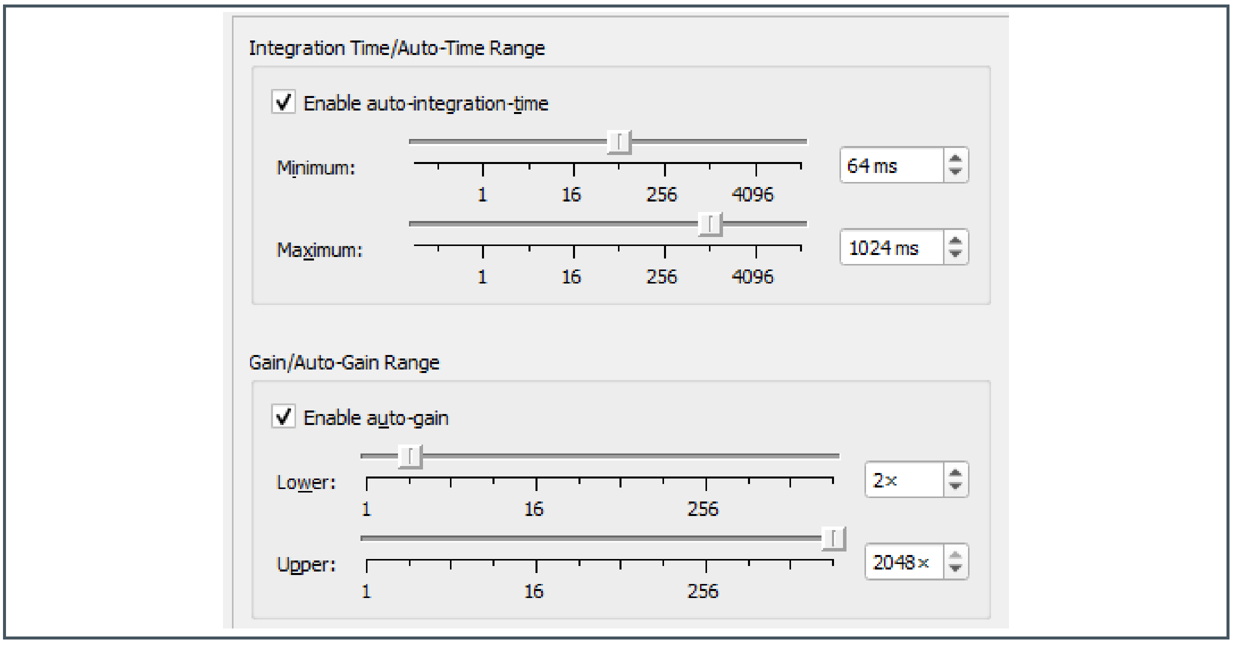
When the connection is established, the status on the right-hand side of the lower status bar will be updated, and it will be possible to configure and perform measurements or terminate the connection by disconnecting the hardware.

Figure 18:
Connection in the Main Tab of the GUI of the Plugged EVK or Disconnect/Refresh

8.2 Configuring a Measurement

To configure measurements, a connected device is needed. On the left side of the diagram are several parameters that can be changed to set up measurements. Most parameters are displayed as horizontal sliders and synchronized incrementable/decrementable numeric fields. In addition to the mouse, the sliders can be moved with the left/right arrow keys when selected, and the values of the numeric fields can be changed with the up/down arrow keys. Entering numbers is not supported.

Figure 19:
Suggested ADC-Parameter for the First Tests – Use Auto-Setup for TINT and Gain



Configuration files for mapping sensor results into application-specific units can be set as input parameters by definition: “Sensor Calibration” and ‘Application Calibration’ (see chapter 8.3). The settable parameters are (also see Figure 15):

- **Integration Time**: This determines the conversion time for the ADC, i.e., the time the sensor is exposed, in milliseconds. Internally, this adjusts the CREG1:TIME and CREG3:CCLK registers, specifying a multiple and the duration of the internal clock periods, respectively. Optionally, an auto-integration-time mode can be enabled via the “Enable auto-integration-time” checkbox, and the lower and upper limits are set. The sliders follow an exponential scale. If the lower and upper limits are set to the same value, the auto-gain mechanism is disabled. The same can be achieved by disabling the checkbox and adjusting the single slider/numerical input. The actual integration time value applicable to a set of measured values will be logged, along with the data, if CSV logging is enabled.
- **Gain/Auto-Gain Range**: This specifies either a fixed gain or lower and upper thresholds of an auto-gain mechanism in the device’s firmware. Internally, this mechanism adjusts the CREG1:GAIN register, aiming for an optimal operating point. The gain value and the user-specified thresholds follow an exponential scale. If the lower and upper thresholds are set to the same value, the auto-gain mechanism is disabled. The same can be achieved by disabling the “**Enable auto-gain**” checkbox and adjusting the single slider/numerical input. The actual gain value applicable to a set of measured values will be logged along with the data, if CSV logging is enabled.
- **Wait Time**: This determines the wait time between measurements in microseconds. Internally, this operates on the BREAK register. A value of zero will still incur a wait of three clocks.
- **Digital Divider**[7]: This enables an optional digital division of measured counts values by a value on an exponential scale. The checkbox enables or disables the divider, and the slider and numerical field adjust its value.
- **LED Settings**: This allows for each of the up to eight external LEDs to be configured. The exact number varies with the type of sensor board used and must be specified by EEPROM programming. A group of radio buttons at the top of the group box allows each of the LEDs to be selected in turn. The checkbox and slider/numerical field below allow for it to be toggled and its brightness to be changed on a scale of 0 to 1000. Not all available LEDs may support a brightness setting; please refer to the documentation of the sensor board. To view at a glance which LEDs are enabled, the radio buttons will show the state, and the radio button labels of the enabled LEDs will be bold with a white background. In contrast to the other settings, changing LED settings will take effect immediately.

CAUTION

In general, EEPROM programming for LED use is already done by the manufacturer, except when equipped with harmful UV LEDs and these LEDs are not covered with an optical adapter. In this case, the harmful LEDs are visible to the human eye and the EEPROM must be configured for safety reasons before first-time use. In this case, please follow the instructions from the USB Stick: “*\Init_EEPROM*”.

[7] Must also be considered in calculations for the sensor results – use them as a factor for multiplication with the ADC counts.

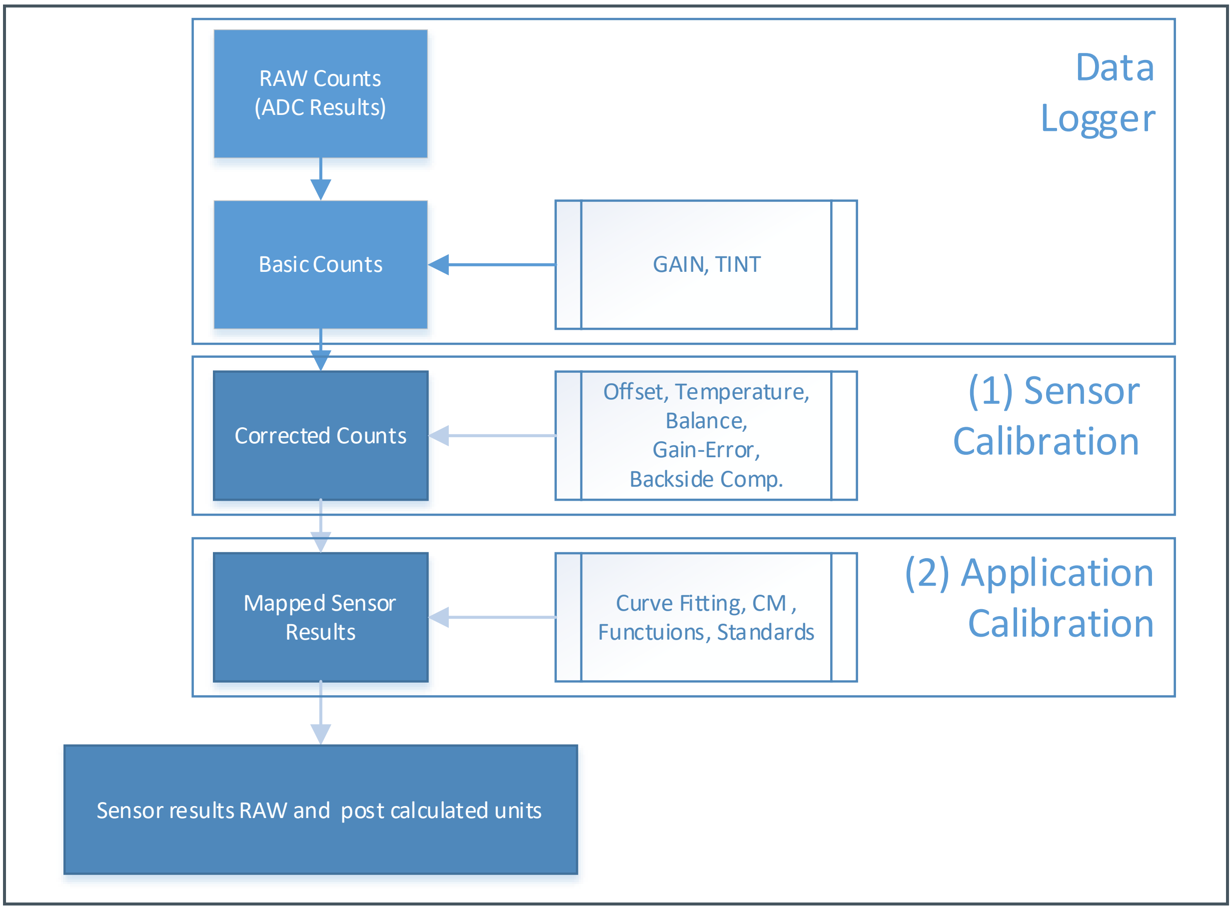
8.3 Calibration Data

The direct sensor results are RAW counts from the ADC or Basic_counts, which represent normalized form digits or counts based on TINT = 1 ms, and gain = 1. Such normalized values are not dependent on any or dynamic sensor setup (e.g. ADC).

Basic_counts are comparably unrestricted but are affected by optical adapters and other application-specific setups. These include the light source, the target to be measured, and any disturbance variables (internal and external) that must be taken into account.

The AS7331 GUI reads the parameters for initialization to the correct sensor-specific ADC results (Sensor Calibration in Figure 20) or to map sensor results into an application and its units (Application calibration). Figure 20 shows the data flow in the GUI with the calibration processes.

Figure 20:
Data Flow of the AS7331 GUI with Data Logger, Sensor, and Application-Specific Calibration



The parameters for the calibrations can be read from the EEPROM or the files on the hard disk. A written EEPROM can be read out with standard functions in the GUI and saved to the disk. Custom writing of calibration data to the EEPROM is not provided in the GUI, to protect the sensor-specific correction data. However, the files on the disk can be used for customer-specific purposes, and they are fixed in their syntax but can be named and saved customer-specifically.

8.3.1 Sensor Calibration File

The following keywords can be used in the Sensor Calibration file. The order of the counts from left to right are generally [UVA, UVB, UVC]:

- **`//:`** Comment line in the calibration files.
- **`ref_temp [1x1]:`** The temperature in which the calibration data is measured, in millidegrees (e.g. 21000 for 21°C).
- **`ref_gain [1x1]:`** The gain in which the calibration data is measured (e.g. 5 for gain 2^5=32). This parameter is not used in calculations, but it is saved in the calibration file to enable possible advanced calibration algorithms in the future.
- **`ref_time [1x1]:`** The integration time in which the calibration data is measured. It is in microseconds (e.g. 100000 for 0.1 seconds).
- **`ref_divider_enable [1x1]:`** Enable or disable divider for calibration.
- **`ref_divider_value [1x1]:`** The divider with which the calibration data is measured.
- **`poly_coeff_for_temp [1x2]:`** The two coefficients [a,b] represent the parameters for a linear function temperature = temperature sensor output * a + b, whereby temperature sensor output is in millidegrees (e.g. around 21000 for the room temperature (21°C)). The default values are [a=1, b=0], if the temperature sensor output is already calibrated.
- **`poly_coeff_for_temp_counts [1x18]:`** The coefficients are parameters of a set of fifth-order polynomial functions for temperature compensation in three channels with respect to the reference temperature (`ref_temp`):
  $[C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}\ ,\ C_{20}, C_{21}, C_{22}, C_{23}, C_{24}, C_{25}\ ,\ C_{30}, C_{31}, C_{32}, C_{33}, C_{34}, C_{35}\ ]$
- For channel i : $C(T) = C_{i0} + C_{i1}T + C_{i2}T^2 + \cdots + C_{i5}T^5$, where T= temperature in millidegree Celsius , and i is the channel number ($i = 1{,}2{,}3$ for UVA,UVB,UVC respectively).
  The default values for no temperature compensation are: $[1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0]$

  - The determination of the temperature coefficients are done in two steps:

1. Measure the counts in different temperatures and calculate the following:

   $C(T) = (\,measured\ counts\ in\ T\,)/(\,measured\ counts\ in\ T_0\,)$, where T_0 is the `ref_temp.`
   (The light source should be kept constant during the temperature change of the sensor.)

2. Afterward, fit the polynomial coefficients C_0 ... C_5 to the measured values of $C(T)$

- **`poly_coeff_for_temp_offset [1x18]:`** The description "`poly_coeff_for_temp_counts [1x18]`" will be seen but for "offset". Note that the offset counts are the dark counts and the coefficients should be determined after switching off the light source. The `ref_temp` should be the same for both offset and counts.
- **`offset_Basic_counts [1x3]:`** The values are offset counts (dark counts, normalized to gain = 1 and Tint = 1ms). Assume the measured offset counts in gain G and integration time T_{int} are [N_1 , N_2, N_3], then the `offset_Basic_counts` are: [$N_1\ /GT_{int}, N_2\ /GT_{int}, N_3\ /GT_{int}$]. For the best performance, the offset measurement should be done in `ref_temp.`
- Note that the offset is valid for a fixed divider value, and by changing the shifter value, the offset values in calibration data have to be updated.

- `balancing_coeffs [1x9]:` These coefficients are used to match the counts of different devices to a golden device. This improves the inter-module repeatability: [a_{11}, a_{12}, a_{13} , a_{21}, a_{22}, a_{23} , a_{31}, a_{32}, a_{33}]. Assuming the counts are [N_1, N_2, N_3] , the balanced counts are calculated as follows:

  Balanced counts = $[N_1, N_2, N_3] * \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

  - **A simple balancing concept:** The counts of different devices are measured in the `ref_temp` under a constant light source, which illuminates all three channels. The ratio between the counts of different devices to the golden device can be put in the diagonal elements while keeping the other elements of the matrix zero.
  - **Default values for no balancing:** [1 0 0, 0 1 0, 0 0 1]

- `gain_corr [1x36]:` These sets of coefficients are used to fine-tune the gain ratios. The coefficients are stored as follows: [$a_{1,0}, a_{1,1}, \dots a_{1,11}$, $a_{2,0}, a_{2,1}, \dots a_{2,11}$, $a_{3,0}, a_{3,1}, \dots a_{3,11}$]. Which relate to the gains 0,1, ….11 according to the following pattern:

  - Gain 0 (2^0): [$a_{1,0}$, $a_{2,0}$, $a_{3,0}$]
  - Gain 1 (2^1): [$a_{1,1}$, $a_{2,1}$, $a_{3,1}$]

  ….
  - Gain 11 (2^11): [$a_{1,11}$, $a_{2,11}$, $a_{3,11}$]

  Assuming the counts are [N_1, N_2, N_3], after normalizing the counts to gain (G) and integration time (T_{int}), you will have[$N_1/GT_{int}, N_2/GT_{int}, N_3/GT_{int}$]. Then, by considering the gain correction factors, the final normalized counts will be:

  $$[N_1 \frac{1}{GT_{int}} * a_{1,i} \quad , \quad N_2 \frac{1}{GT_{int}} * a_{2,i} \quad , \quad N_3 \frac{1}{GT_{int}} * a_{3,i}]$$

  Where i is the gain coefficient index according to the above relations. The default values for having no gain coefficient correction is by setting all the values to one: [1, 1, 1 …1].

8.3.2 Application Calibration File

The following keywords can be used in the Application Calibration file. The counts order from left to right are generally [UVA, UVB, UVC]:

- `//:` Comment line in the calibration files.
- `Units:` Definition of the application-specific units that are used by the GUI for outputs. It has no effects on any of the calculations or definitions.
- `ref_temp:` It is the same as that in the sensor calibration data.
- `ref_gain:` It is the same as that in the sensor calibration data.
- `ref_time:` It is the same as that in the sensor calibration data.
- `calib_matrix [1x9]:` This matrix can be used to remove inter-channel spectral crosstalk if necessary: [$a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}$]. Assuming the counts are [N_1, N_2, N_3], the counts without crosstalk are calculated as follows:

  Counts without crosstalk = $[N_1, N_2, N_3] * \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

  The default values for “without crosstalk”: [1 0 0, 0 1 0, 0 0 1]
- `poly_coeff_per_channel [1x9]:` $[r_{10} \ \ r_{11} \ \ r_{12} \ \ r_{20} \ \ r_{21} \ \ r_{22} \ \ r_{30} \ \ r_{31} \ \ r_{32}]$

For each channel, a second-order polynomial is considered to convert the counts to the desired parameter. The desired parameter could be the light intensity for example (W/m^2).
Example: $[c_1,c_2,c_3]$ = normalized_counts for all the sensor channels.
Then, the coefficients are used to calculate the following absolute values by the polynomial functions:

$$I_1 = r_{10} + r_{11}c_1 + r_{12}c_1^2$$

$$I_2 = r_{20} + r_{21}c_2 + r_{22}c_2^2$$

$$I_3 = r_{30} + r_{31}c_3 + r_{32}c_3^2$$

Where, $[I_1,I_2,I_3]$ are the absolute sensor results and can be e.g. intensities_for_each_channel in (W/m^2).

- **`channel_combination_coeff [1x3]:`** $[q_1,q_2,q_3]$
  These coefficients are used to get to one value from the three channels to do further processing. Where,
  [q1,q2,q3] = channel combination coefficients
  [c1,c2,c3] = normalized counts

  $$x = c1q1 + c2q2 + c3q3$$

  By setting some of the coefficients to zero, channels can be neglected.
- **`poly_coeff_combined [1x3]:`** $[t_1,t_2,t_3]$
  These coefficients are applied to the single value obtained in the above step to get to the final desired parameter (for example the final intensity). Using this and the previous parameters it is possible to give an absolute intensity over the whole wavelength range of the three channels in W/m^2.

  $$I_{tot} = t_1 + t_2x + t_3x^2$$

  Where x is obtained in the above step.

Figure 21 shows an example of the absolute sensor results, where the calibration data were used to calculate:

- Intensity in W/m^2 for channel A (B, C) from RAW and Basic_counts A (B, C).
- Total intensity in W/m^2 as the sum of all the Intensities in W/m^2 from channels A, B, C.
- Thresholds are not used; therefore, the Application Output is “0”.

Figure 21:
Sensor Output Table with the Calibration Results

Gain	ITime [ms]	Temp [°C]	Raw A	Raw B	Raw C	Intensity A [W/sqm]	Intensity B [W/sqm]	Intensity C [W/sqm]	Intensity Total [W/sqm]	App Output [W/sqm]
2048	1024.000	27.45	43128	24523	57829	0.000000	0.000000	0.041249	0.041249	0.000000

8.4 Performing Measurements

After setting up all the useful parameters, measurements can be performed using the controls in the upper right-hand side of the main window. Gain and integration time are the most important parameters to affect the counts or digits as sensor results. If these parameters are set to low, the sensor will measure noise, and if they are too high, the sensor will go into saturation. These two conditions are wrong and will not give the proper and acceptable sensor results. Chapter 9.1 shows more details about the first test to get optimized sensor results. Pressing the "Start" button begins the measurement. This turns the button into a "Stop" button - so that a continuous measurement process can be stopped.

> **NOTE:** It may happen that a certain combination of parameter settings, which are subject to complex conditions at the device level, is not permissible. In this case, starting a measurement will fail with an error message indicating this. If no signal can be seen in the diagram after starting measurements, the setup for the ADC may be wrong, and the sensor measures only noise. Then, increase gain and integration time or select automatic for gain and integration time. At this point, it is important to set the limits for both parameters correctly.

To perform a certain number of measurements, a value greater than 0 is selected in the “n =” input field to the right of the “Start” button, and the measurement is started. After the specified number of measurements has been performed, the process is automatically stopped. To perform an indefinite number of measurements in CONT means to set the value “n =” to zero.

By default, the graphical output is enabled via the “Enable graph” checkbox. This will display the measurements as a graph and line diagram, for the sensor outputs’ UV and temperature. The graph will display up to the last n measurements, specified via the “up to” input box on the right side of the checkbox. This limitation is done for performance reasons, as a large number of points plotted live during an ongoing measurement process, with a short integration time, can lead to performance problems. Depending on the system, performance can also suffer from short integration times. As a result, a warning message is displayed when starting a measurement, with graph plotting enabled in such cases. In case of very short integration times, it is recommended to deactivate the graph and only output to a CSV log. The graph can be switched on and off, as desired, during a running measurement process. Starting a new measurement process deletes the graph (but not the log file).

The measurement data can be logged to a CSV file. This is disabled by default and can be enabled via the “Log CSV to” checkbox. By default, a file is preset to be located in a subdirectory named after the device in your home directory and has a name that includes both the device type and a wall clock timestamp. The file can be changed via the "Select File..." button. This will open a file selection dialog box that defaults to the same directory and has a file name matching the default name but containing a current timestamp. In the case of write errors to the log file during a running measurement, an error box is displayed, and the measurement stops. The checkbox “Abort at saturation” causes a running measurement to abort if saturation is reached for one of the channels of the sensor. In this case, an error box appears, and the measurement stops. Regardless of this, saturation will be highlighted on the chart (if enabled; see chapter 8.5), notified on the status bar, and logged to the CSV file (if enabled; see chapter 8.5). The checkbox will become locked in a checked state if auto-integration-time or auto-gain are enabled.

8.5 Graphical Output

If enabled, a graph of an ongoing measurement is rendered in real-time in the right-hand area of the main window (see Figure 15).

The graph shows four series:

- Three, for the three sensor channels, labeled A, B, and C, relating to the left y-axis, with a unit of counts.
- One for the internal temperature sensor, labeled T, relating to the right y-axis, with a unit of °C.

Depending on the device, there may be up to eight additional series for **external temperature sensors**, labeled **T0** through **T7**, corresponding to the respective sensor index, and relating to the right y-axis with a unit of °C. These have their points marked by a distinctive symbol per series, in addition to a color gradient across the set of series.

> **NOTE:** Due to technical limitations, if there are more than four external temperature sensors available, values will only be received and plotted for the first four values. Regardless, the legend will show all the available sensors.

Intervals during which **saturation** has occurred for any of the channels will be **highlighted in red**.

The time axis reflects the timestamps of the measurements as obtained from the device. For comprehensibility, the scale shows them as an offset to the first measurement in the currently-running (or last-run) measurement process in seconds.

Hovering on the chart will show the channel and time values at the crosshair’s location. Drag, using the left mouse button, and zoom in on a rectangular region. Right-clicking or pressing the “Zoom to fit” button will reset the zoom to show the entire series.

The “Normalize” checkbox will toggle normalized Basic_counts obtained from the raw counts by normalizing to:

- An integration time of 1ms.
- A gain of 1.
- A divider value of 1 (i.e., no divider).

This does not affect the data logged to a CSV file.

A table with the current effective gain, integration time, and the last measured results as RAW counts for all the channels, is shown below the chart (see Figure 21). Additionally, the absolute values, in units, are added to this table if the calibration files are read in the GUI.

Attention

Unit is only shown in the table in the absolute results if it is specified in the calibration files.

8.6 CSV Output

If enabled, measurement data can be logged to a CSV file. Logging can be toggled via the “**Start CSV Logging**” button (which will then turn into “**Stop CSV Logging**”).

A default file with a path and name, depending on the device type, will be initially set. This can be changed via the “**Select File…**” button, which will open a file selection dialog. The current log file path is displayed below the buttons.

When selecting a new file, a preexisting file of the same name will be overwritten, (you will be prompted for confirmation). Running multiple measurement processes with the same file set will append to that file. Changing measurement settings intermittently, will ask for confirmation of whether you want to log data obtained with different settings to the same file.

To the right of the file selection button, there is a checkbox labeled “**Auto-append timestamp**” (checked by default). With this checked, the selected file will automatically get a textual timestamp appended to its name, as soon as a measurement is started with logging enabled. Stopping and starting logging without selecting a new file will generate and use a new timestamp.

When logging is active, and any data has been written to the current file, a “**Logging**” marker will be shown on the right-hand side of the status bar.

Figure 22:
Example of CSV Log File with the Timestamp, Sensor Results, and Setup

	A	B	C	D	E	F	G	H	I	R	S	T	U
1	Timestam	A	B	C	Temperat	Integratio	Gain	Divider	LED 0	ExtTemp 1	ExtTemp 2	ExtTemp 3	Error
2	0	3258	2688	6623	22	256	2048	1		21.562	21.812	21.562	0
3	286689	2582	2316	5545	22	256	2048	1		21.562	21.812	21.562	0
4	573409	2594	2352	5651	22	256	2048	1		21.562	21.875	21.562	0
5	860009	2445	2272	5476	22	256	2048	1		21.562	21.875	21.562	0
6	1146641	2624	2380	5729	22	256	2048	1		21.562	21.875	21.562	0

The CSV includes a header row and consists of the following columns, in this order:

- (device) Timestamp (in microseconds)
- Raw counts
- Internal temperature (in °C)
- Integration time (in milliseconds)
- (effective) Gain value
- Divider value (‘1’ if disabled).
- Error value (see below).
- State of external LEDs 0 through 7, with an empty value, if not present. Otherwise, the brightness value or 0 if disabled.

The error field logs any error that may have occurred. These values correspond to error values as used internally with the host and chip libraries; please refer to the corresponding documentation for the meaning of the values. Value 34 indicates the saturation of any channel. The occurrence of an error aborts the measurement; only value 34 may be logged continuously along with data.

8.7 Further UI Elements

The application also features a menu bar containing two menus:

- The “File” menu only contains an option to quit the application.
- The “Help” menu contains an “About…” option - showing a dialog with information about the application version, used open-source software, etc. Another option, shows a dialog of application log messages, otherwise printed to the console if run via a terminal.

At the bottom of the main window, a status bar gives notifications about events as they happen on the left-hand side. It shows persistent status information on the right-hand side, including the connection and measuring states, whether saturation has occurred during the current or last measurement process, and internal version numbers.

9 Application Notes

9.1 First Tests

The basics and principles of the ADC are listed in the datasheet [1]. Gain and TINT are the essential parameters (besides Shift) to influence the Counts as the sensor result. The counts correspond to a digital measure of the amount of light received by the sensor with the set gain and integration time. It is recommended to normalize the counts, with a gain of 1 and a 1ms integration time - to get the results independent of the parameter setup (= Basic_counts = RAW_Counts/(Gain*TINT)).

The integration time also determines the size of the counter. At 64 ms, 16-bit is reached, i.e. 65335 counts can be reached before digital saturation is reached (see Figure 23).

Figure 23:
Relationship Between Integration Time, Resolution, and Maximum Counts (1 MHz Frequency)

TINT (ms)	Resolution (bit)	Max. Counts
1	10	1024
2	11	2048
4	12	4096
8	13	8192
16	14	16384
32	15	32768
64	16	65536
128	16	65536
256	16	65536
512	16	65536
1024	16	65536
2048	16	65536
4096	16	65536

A measurement aims to obtain optimal counts in which stable results with high accuracy can be achieved. The sensor should not measure noise or be in saturation. The integration time (TINT) is mostly predefined as a large part of the measurement time by the application time. With the gain, the higher value of the counts can also be adjusted. The first step in the measurement is to find optimal counts using the parameters for the application and their limits. This can be done manually step-by-step, or it can be optimized automatically by the GUI using auto-Gain and TINT.

The following figures (Figure 24 and Figure 25) show two examples for an optimized setup to get a maximum number of counts; the first is by using automatic settings and the second by manually setting parameters. The counts for the maximum should be at least 60% to 80% of the maximum achievable "Max Counts". If the results are only in the lower range of the counts (one or two digits), the gain or integration time must be increased. If saturation is indicated for the sensor, the gain or integration time must be reduced.

Figure 24:
Optimized Setting Using Automatic Gain and TINT

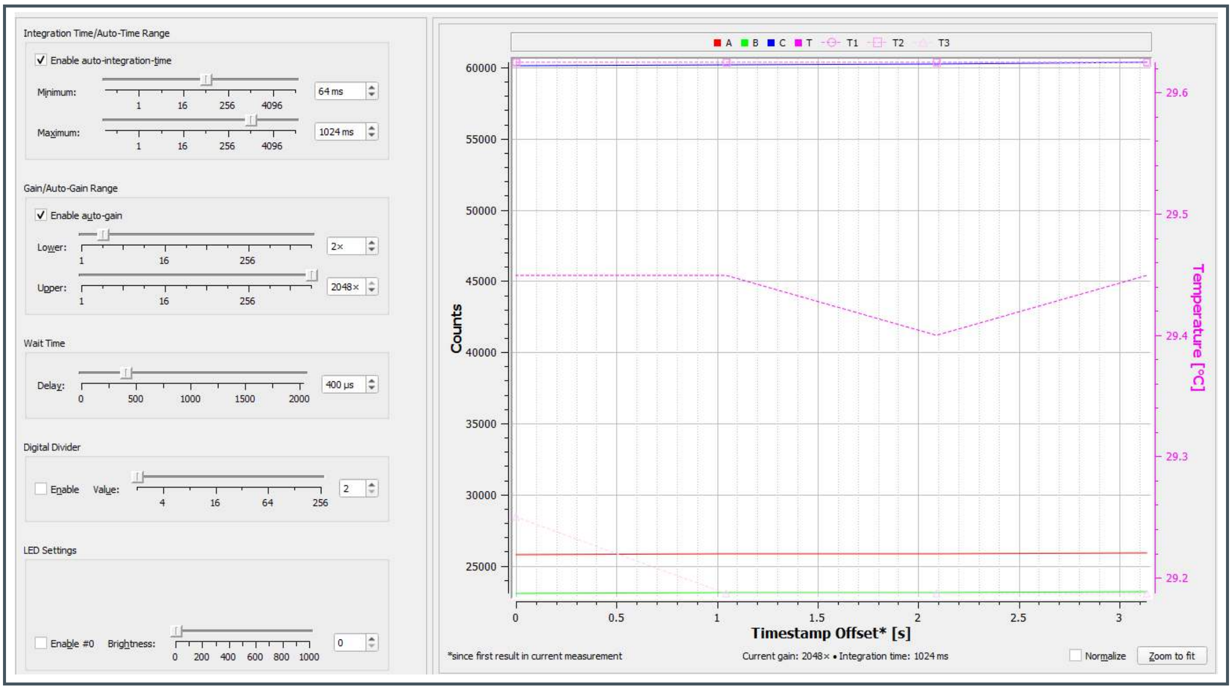
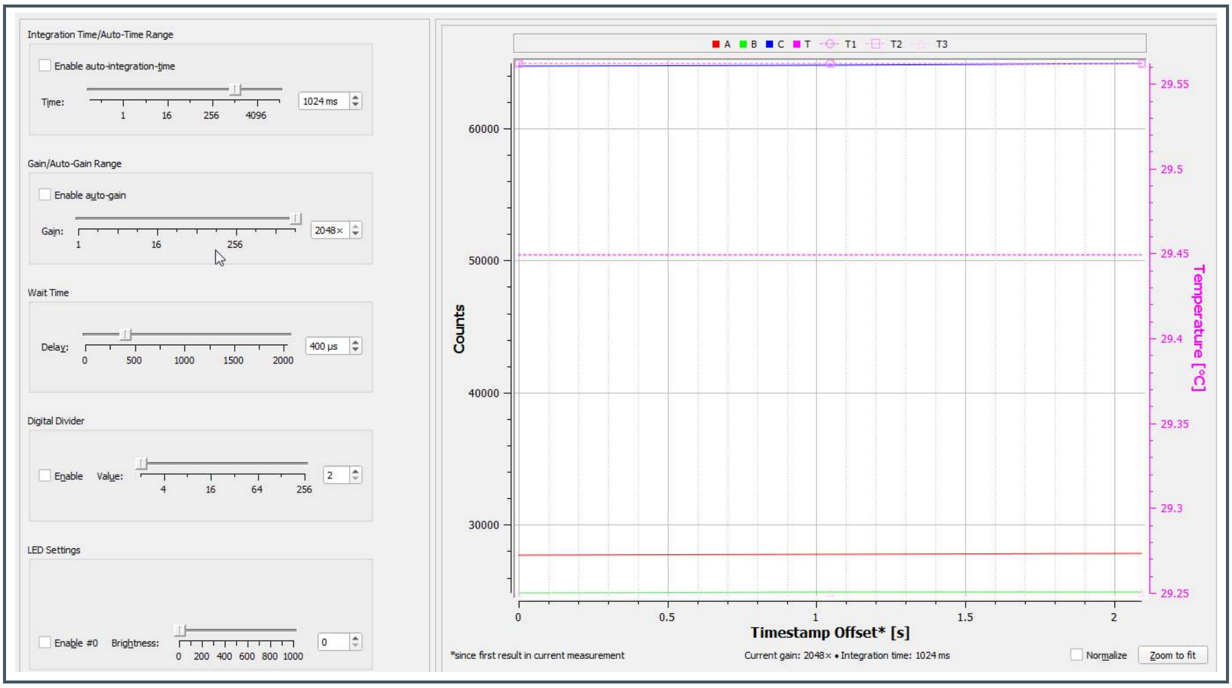


Figure 25:
Optimized Setting Using Manually Set Gain and TINT



9.2 Sensor Signal Correction

Before starting the measurements, check if the sensor complies with all the operating conditions and which disturbances may influence the measurements.

In particular, the angle of incidence of less than 10° deviation from the vertical must be observed, in order to perform the measurements with the specified filter functions. Otherwise, the results of the sensor measurements will be erroneous.

Avoid saturating the sensor and check the sensor settings (gain, integration time, and shifter) to ensure it works in an optimized working range close to FSR (Full Scale Range). Work with a dynamic sensor setting in case of strongly changed sensor inputs. If the setup variables change in the measurement series, the sensor results must be converted to a form independent of the sensor setup before calibration. Use “Basic-Counts” as an independent sensor result for the following corrections.

Disturbances influence the sensor results statically or dynamically. A verification and optimization process must correct or eliminate all these negative effects to obtain optimum results.

The following list includes some examples, which can affect the accuracy, more or less depending on the application:

- Basic noise (e.g. dark current - note, dark currents depend on temperatures)
- Gain error (none linearity)
- Temperature and ageing effects from LEDs
- Overcrossing inside the sensor system
- Ambient light

Such deviations and disturbances are measured within tests and can be corrected customer or application-specifically. This can be done using external software tools like MATLAB or Excel, with the GUI results from the protocol files, or using the correction function that was implemented in the GUI. These corrections are based on the AS7331 libraries, which use the files for corrections and calibrations (see chapter 8.3). If these files are prepared and used in combination with the GUI, then the sensor outputs can also be as application-specific units.

The GUI and hardware can be used to prepare the sensor and application calibration. The following chapter (chapter 9.3) gives an example of how to prepare the corrections and mapping; to measure a UV LED and map the sensor results into the application-specific unit for irradiation in W/sqm.

CAUTION

Accuracy, sensitivity, dynamic, and other application parameters always depend on the physical setup of the test, the calibration data, and external disruptions. The GUI only and always calculates and links the measured counts from the sensor with the specified data from the calibrations files.

9.3 Application Example: UV LED – From Counts to W/sqm

A simple task for the UV sensor and software is to measure active UV LEDs and calculate the irradiance in W/sqm, based on the calibration. Depending on the required accuracy, the sensors have to be calibrated individually or as a batch in series. This procedure is briefly explained in this chapter using the LED measurement example.

The light values of an LED are measured in different current steps [0%, 10%, 20% ...100%] using a spectrometer and sensor. The conditions should be the same for the measurements by the sensor and the spectrometer. The following figures (Figure 26 to Figure 28) show the results as diagrams and numerical values for both devices in a table. The sensor used was the AS7331 EVK, complete in the housing with a standard adapter. The spectrometer is a commercially available instrument with a sensitivity from 250 nm.

Figure 26:
Spectrometer Results of UV-LED using 0%, 10%, 20%...100% LED Current

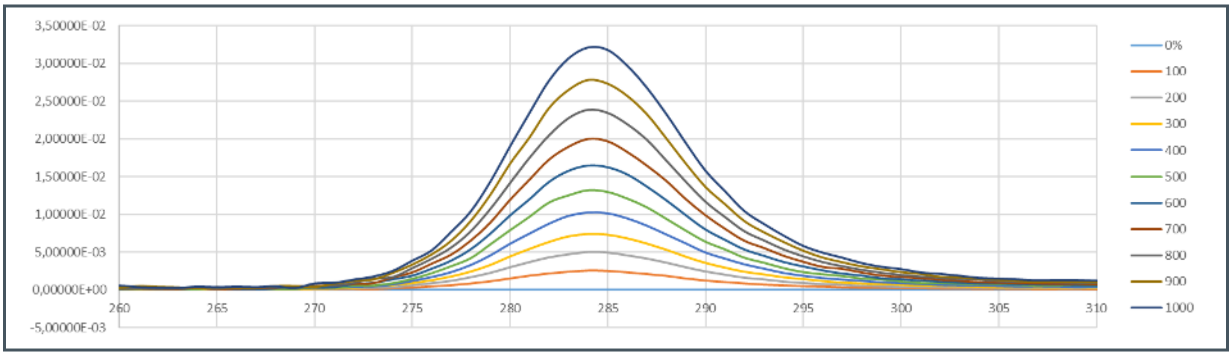
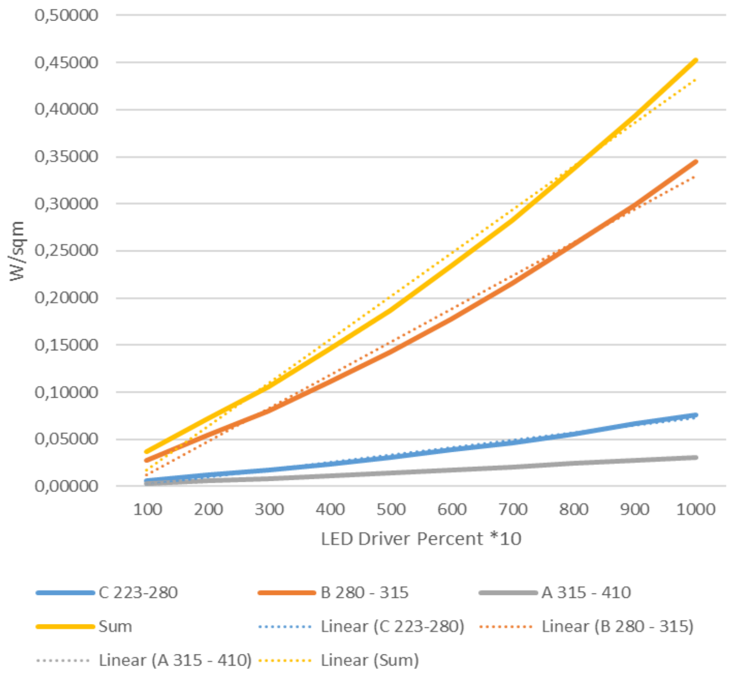


Figure 27:
Spectral Sum of A/B/C Ranges Based on the Spectrometer Results from Figure 26

m/W/sqm	10	20	30	40	50	60	70	80	90	100
C 223-280	0.00569	0.01197	0.01739	0.02395	0.03052	0.03871	0.04624	0.05573	0.06668	0.07607
B 280 - 315	0.02777	0.05395	0.07987	0.11042	0.14232	0.17793	0.21506	0.25707	0.29913	0.34511
A 315 - 410	0.00287	0.00574	0.00843	0.01159	0.01436	0.01765	0.02049	0.02436	0.02745	0.03102
Sum	0.03632	0.07165	0.10569	0.14595	0.18720	0.23429	0.28179	0.33715	0.39326	0.45220

Figure 28:
Diagram of the Spectrometer From Figure 27



The LED irradiance in UVA/B/C measured by the spectrometer is more or less linear (Figure 28) and will be used as the target to map the sensor results into the unit W/sqm per filter response for UVA-UVC. Temperature deviations or other disruptions were not optimized and considered to explain this example. It should be done in the case of any device development, depending on the required series stability, repeatability, and accuracy. The sensor results were measured with the set of Gain = 2048 and TINT = 1024 ms under identical conditions. The results are shown in the following figures (Figure 29 to Figure 31).

Figure 29:
Sensor Results as Raw_Counts for Channel A/B/C

LED Driver in %	Average of A in CBC	Average of B in CBC	Average of C in CBC
0 = Dark	13.8	11.2	17.2
100	18.1	33	20
200	23	53.9	22.1
300	27.3	74.6	24.9
400	32.9	98.8	28.6
500	37.2	123.4	31
600	42.2	152.1	34.6
700	47.7	182.4	38.5
800	53.8	216.4	43.2
900	58.6	251.6	46.8

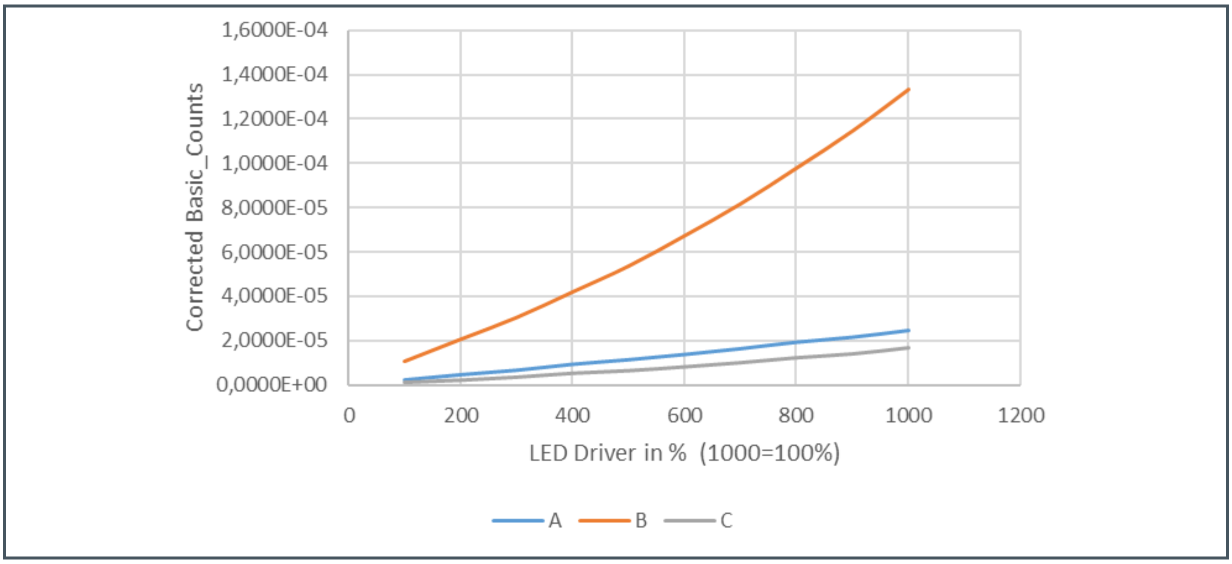
LED Driver in %	Average of A in CBC	Average of B in CBC	Average of C in CBC
1000	64.9	290.9	52.5

The sensor counts must be corrected with Offset = Dark measurement. Basic_counts are used to compensate for Gain and TINT settings (Basic_counts = Raw_Counts/(TINT * Gain)). The following figures show the results for the calculation of corrected Basic_counts (CBC).

Figure 30:
Sensor Results as Corrected Basic_counts for Channel A/B/C

LED Driver in %	Average of A in CBC	Average of B in CBC	Average of C in CBC
0	6.58035E-06	5.34058E-06	8.2016E-06
100	2.0504E-06	1.0395E-05	1.3351E-06
200	4.3869E-06	2.0361E-05	2.3365E-06
300	6.4373E-06	3.0231E-05	3.6716E-06
400	9.1076E-06	4.1771E-05	5.4359E-06
500	1.1158E-05	5.3501E-05	6.5804E-06
600	1.3542E-05	6.7186E-05	8.2970E-06
700	1.6165E-05	8.1635E-05	1.0157E-05
800	1.9073E-05	9.7847E-05	1.2398E-05
900	2.1362E-05	1.1463E-04	1.4114E-05
1000	2.4366E-05	1.3337E-04	1.6832E-05

Figure 31:
Sensor Results as Corrected Basic_counts for Channel A/B/C



At this point, all other corrections on the Basic_counts level can be done if necessary. This refers to:

- Disruptions based on Ambient Light by “Backlight Compensation”.
- Temperature compensation by referencing or correction function, just in case the environment temperature changes during the process or the LEDs show deviations over the processing time.
- Gain correction in the case of using “Automatic Gain Correction”.
- Balancing of individual deviations using correction values of the Golden Devices.

The following formula describes the calculation of the Corrected Basic_counts of the sensor, which must be done for each channel. Such Basic_counts are the basis for mapping the sensor results, using the target values of the spectrometer into the unit of the application.

Equation 1:

$$Corrected\ Basic\_Counts(A,B,C) = \frac{SFac_1 \times \ldots \times SFac_n \times (Raw\_Counts - Offset\_Counts)}{(TINT \times Gain)}$$

Where:

Basic_Counts	= ADC Raw_Counts normalized to gain = 1 and TINT = 1 ms
$SFac_{1\ldots n}$	= Scaling_Factors for Temperature, Gain Correction, Balancing, etc.
Raw_Counts	= Results of the ADC on the Sensor
Offset_Counts	= Measurement in Setup e.g. Dark with switched off LEDs
TINT, Gain	= Used ADC parameters in tests

This mapping will depend on the application. Often, more than one different method can be used, e.g. in linearity, stability, and accuracy. The reasons for this are diverse and can be seen as an example in Figure 32.

The LED is active between wavelengths of 260 nm and 310 nm, with a peak of 284 nm. The filter characteristics of the sensor are C [223 nm – 280 nm], B [280 nm – 315 nm], and A [315 nm – 410 nm]. The following details are observed by looking at the overlaps between the sensor filters and the LED spectral curve:

- All the channels are affected by LED changes, but the proportion per channel is different. The ratios of the channels to each other can be used to identify spectral interferences.
- In the wavelengths of filters B and partly C, the highest overlaps are to be expected, i.e. channel B especially is to be used for mapping and brings the largest proportion when summed.
- Channel C is only slightly affected by the overlap, but will also change slightly with the LED.

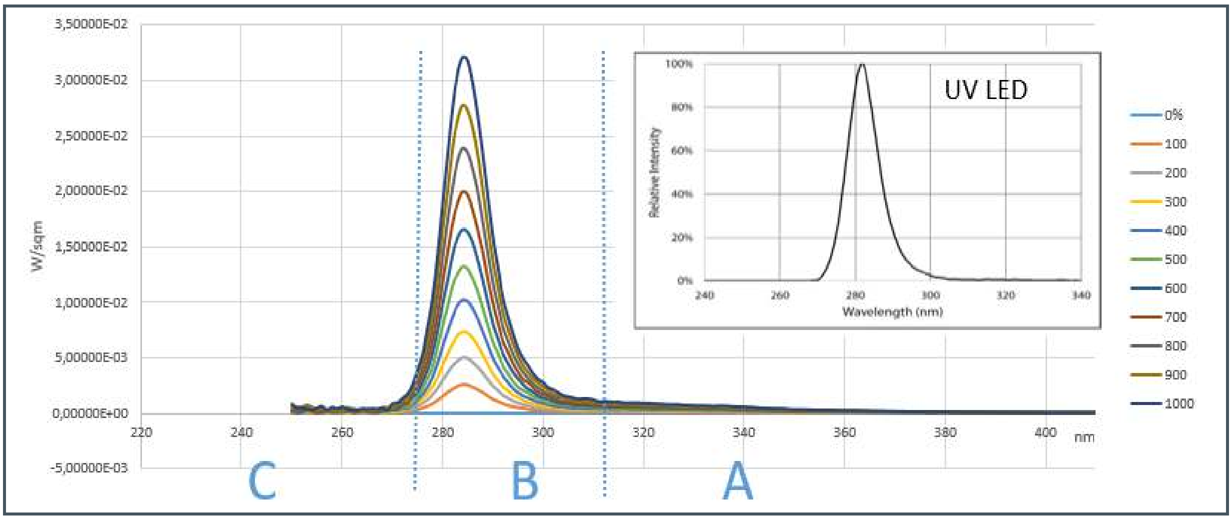
For these points, sensor-specific characteristics must be taken into account, which can be seen in the datasheet of the sensor [1]. For example, the diode area of sensor channel C is doubled and as large as that of channels A + B, i.e. the counts will therefore be larger by a factor of 2. The filters of the sensor differ in series by a maximum of 1% * lambda, i.e. there can be shifts in the ratios of the channels among each other.

The target values (measured by the spectrometer) for the spectral irradiance of the UV LED are known and defined for all measured LED driver currents. These values can be used to map the Sensor corrected Basic_counts per channel, per sum of channels, or as a combination of single channels.

Figure 33 shows the mapping table. Column 1 contains the LED driver levels in percent (%), column 2 to column 4 shows the measured sensor channels, and the other columns show the references and target values of the spectrometer. The contents of the table in Figure 33 are shown in Figure 34 to Figure 37 as a curve fitting.

For example, the "fitted function" in Figure 34 corresponds as the closest linear function to the mapped linear function, resulting from the "reference value" of the spectrometer for UVA and the "corrected basis values" of the sensor channel A. This fitted function can be used for calculation, based on the corrected Basic_counts for channel A (represented as “x” in the function), the absolute UV irradiance in W/sqm (represented “y” in the function), for range A [315 nm – 410 nm].

Figure 32:
LED Curve and Sensor Filter Overlapping for C, B (Left), and A (Right)



The same procedure can be done for B and C, to get their irradiance in W/sqm. The sum of all three shared ranges can be used as UV irradiance in W/sqm from 223 nm up to 410 nm.

Figure 33:
Mapping Target Values from the Spectrometer to the Sensor Channels

LED Driver in %	A in CBC	B in CBC	C in CBC	Ref A in W/sqm	Ref B in W/sqm	Ref C in W/sqm
10	2.0504E-06	1.0395E-05	1.3351E-06	2.8680E-03	2.7771E-02	5.6854E-03
20	4.3869E-06	2.0361E-05	2.3365E-06	5.7381E-03	5.3948E-02	1.1966E-02
30	6.4373E-06	3.0231E-05	3.6716E-06	8.4340E-03	7.9865E-02	1.7392E-02
40	9.1076E-06	4.1771E-05	5.4359E-06	1.1587E-02	1.1042E-01	2.3948E-02
50	1.1158E-05	5.3501E-05	6.5804E-06	1.4364E-02	1.4232E-01	3.0520E-02
60	1.3542E-05	6.7186E-05	8.2970E-06	1.7648E-02	1.7793E-01	3.8713E-02
70	1.6165E-05	8.1635E-05	1.0157E-05	2.0489E-02	2.1506E-01	4.6241E-02
80	1.9073E-05	9.7847E-05	1.2398E-05	2.4357E-02	2.5707E-01	5.5726E-02
90	2.1362E-05	1.1463E-04	1.4114E-05	2.7450E-02	2.9913E-01	6.6680E-02

LED Driver in %	A in CBC	B in CBC	C in CBC	Ref A in W/sqm	Ref B in W/sqm	Ref C in W/sqm
100	2.4366E-05	1.3337E-04	1.6832E-05	3.1021E-02	3.4511E-01	7.6067E-02

Figure 34:
Curve Fitting Target Values Ref A from the Spectrometer to the Sensor Channel A

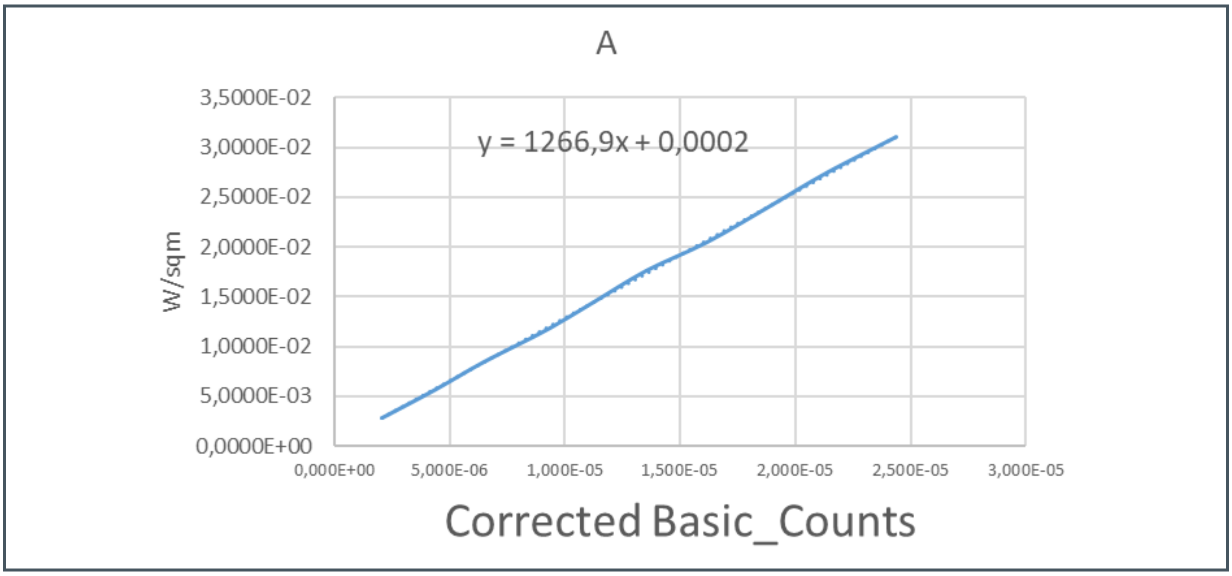


Figure 35:
Curve Fitting Target Values Ref B from the Spectrometer to the Sensor Channel B

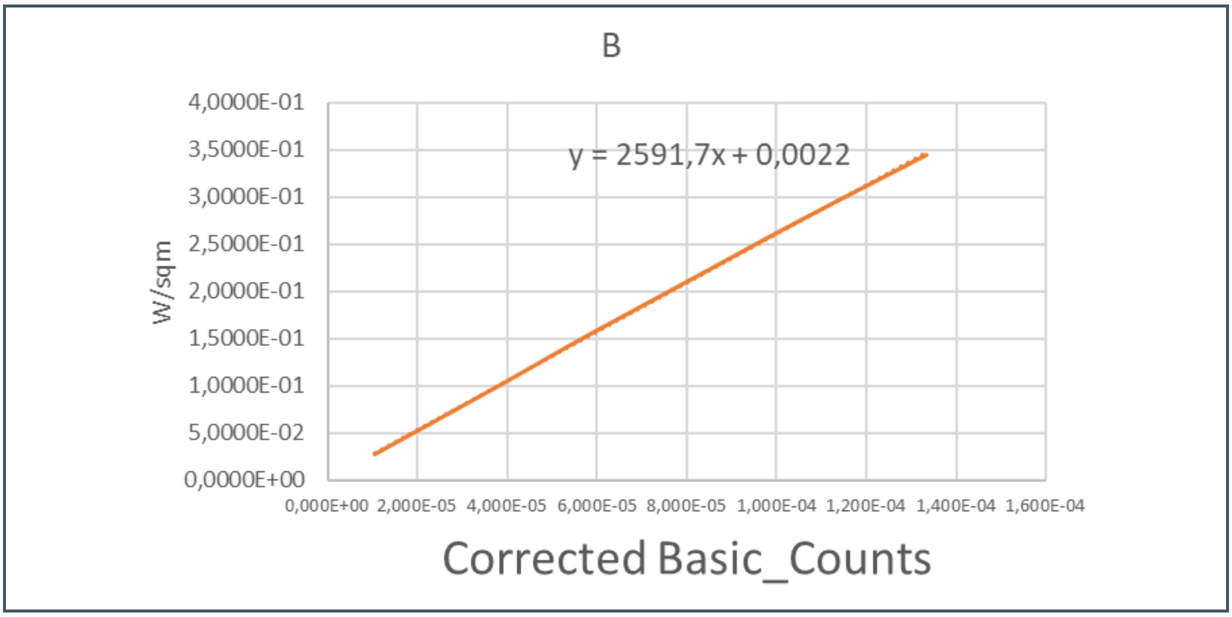
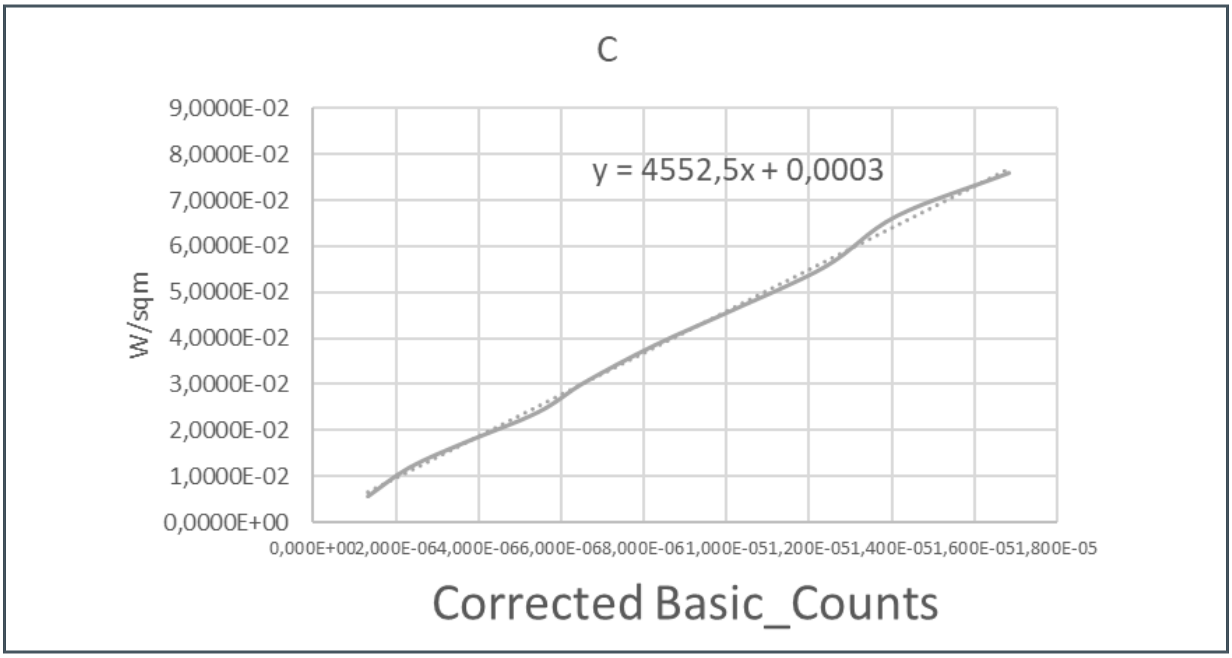


Figure 36:
Curve Fitting Target Values Ref C from the Spectrometer to the Sensor Channel C



Such fitted functions exist from the function type (linear, logarithmic, quadratic, etc.), parameters, and, if useful, shared dynamic ranges. The formulas, as well as the parameters and ranges, can be inserted directly into the sensor device software to convert the sensor results' Basic_counts, after corrections into the application result irradiance. Shared ranges and their parameters can also be part of a device-specific series calibration and must be saved and read, in this case from the software, as device-specific technical input parameters.

Figure 37 shows the results for the three channels in W/sqm, using fitted functions from Figure 34, their deviations (absolute error) from the column "reference" for each selected W/sqm per step, and the overall average. Such results must be verified within typical laboratory and field tests.

Figure 37:
Deviation Comparison Using Different Fitted Functions

Verification Fitting	A in CBC	B in CBC	C in CBC	Dev A	Dev B	Dev C
100	2.7977E-03	2.9141E-02	6.3782E-03	2.5%	4.9%	12.2%
200	5.7578E-03	5.4969E-02	1.0937E-02	0.3%	1.9%	8.6%
300	8.3554E-03	8.0551E-02	1.7015E-02	0.9%	0.9%	2.2%
400	1.1738E-02	1.1046E-01	2.5047E-02	1.3%	0.0%	4.6%
500	1.4336E-02	1.4086E-01	3.0257E-02	0.2%	1.0%	0.9%
600	1.7357E-02	1.7633E-01	3.8072E-02	1.6%	0.9%	1.7%
700	2.0679E-02	2.1377E-01	4.6538E-02	0.9%	0.6%	0.6%
800	2.4364E-02	2.5579E-01	5.6741E-02	0.0%	0.5%	1.8%

Verification Fitting	A in CBC	B in CBC	C in CBC	Dev A	Dev B	Dev C
900	2.7264E-02	2.9929E-01	6.4556E-02	0.7%	0.1%	3.2%
1000	3.1070E-02	3.4786E-01	7.6929E-02	0.2%	0.8%	1.1%
			Average	0.9%	1.2%	3.7%

The results are characterized by a consistently good accuracy, which could be optimized in the lower ranges. More measurements and possibly a separate fitting curve with divided ranges for a lower "driver power" of the LED can still increase the accuracy here. The causes could be the non-linearity of the LED driver and the sensor, the signal-to-noise ratio of the sensor signals, and other things.

If corrections and calibrations are to be used in the GUI, then the data from the process above must be set into the GUI's calibration file. Figure 38 and Figure 39 show the adapted calibration files for the example described in this chapter.

Figure 38:
Example of the Sensor Calibration File

```
// correction file, example

ref_temp,21000

ref_gain,12

ref_time,1024000

ref_divider_value,0

ref_divider_enable,0

poly_coeff_for_temp,1,0

poly_coeff_for_temp_counts,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0

poly_coeff_for_temp_offset,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0

// offset order ABC in µs

offset_Basic_counts, 6.58035E-09, 5.34058E-09, 8.2016E-09

balancing_coeffs,1,0,0,0,1,0,0,0,1

gain_corr,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1
```

Figure 39:
Example of an Application Calibration File

```
// calibration file, Example

description,desinfec_box

unit, W/sqcm

// CM not used

calib_matrix,1,0,0,0,1,0,0,0,1

// parameter for curve fitting in µs

poly_coeff_per_channel, 0.0002, 1266.9, 0, 0.0022,2591.7, 0, 0.0003,
4552.5, 0

poly_coeff_combined, 0.0023, 2811.3, 0

// not used

channel_combination_coeff,0,1,1

threshold,1,2

application_coeff,0,0,0,0,0,0,0,0,0
```

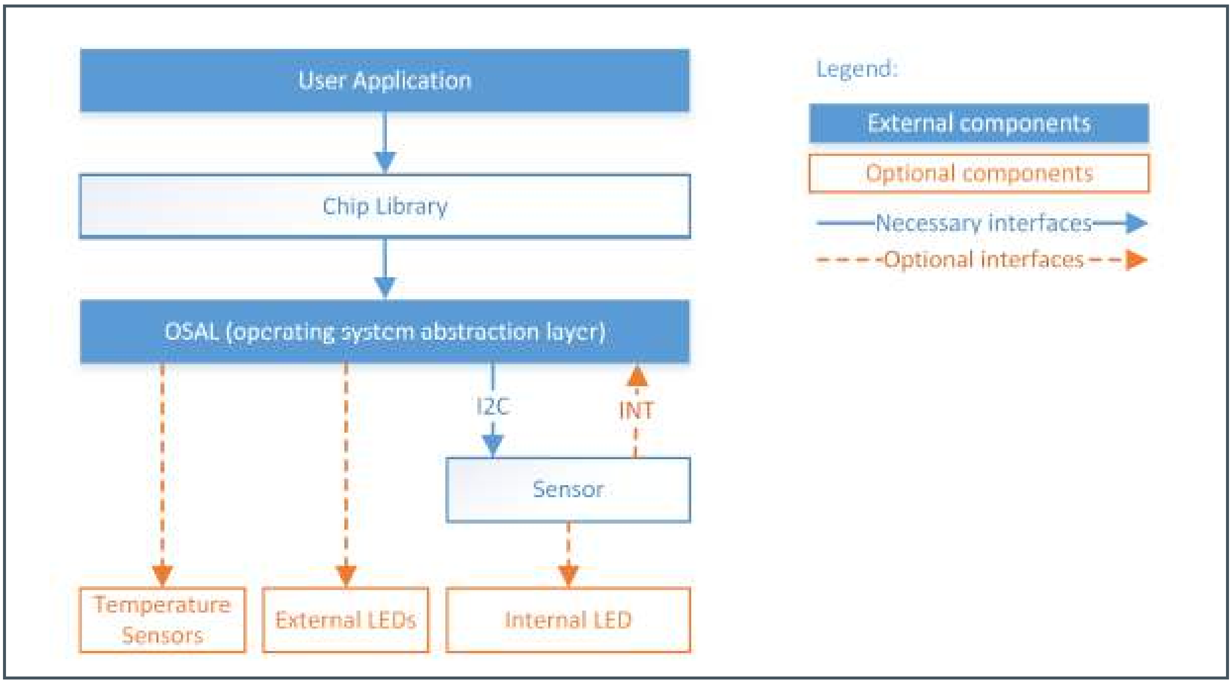
Afterward, the sensor and the application calibration files can be used in the GUI to measure not in Raw counts but W/sqm for the activated channels and the sum of all the channels.

9.4 Using User-Specific Hardware and Software

ams OSRAM offers pre-developed software libraries and modules to use the AS733x-based EVKs in user-specific hardware and software. The chip library (ChipLib) is the central component that serves as the sensor driver. It is a generic API and allows for usage in several different fields of application. It is programmed in standard C, can directly be called by the user application, and can be used without adaption. Hardware and platform dependencies must be specified separately in the OSAL.

The OSAL interface is kept simple to keep integration efforts for the customers to a minimum. The simplest implementation would be a measurement routine without an interrupt and external peripherals. More information on the API functions can be found in the ChipLib documentation.

Figure 40:
Structure Overview Using AS733x ChipLib



9.5 AS733x SDK

The AS733x Software Development Kit SDK is a collection of software components written in C programming language, allowing a user to quickly prototype solutions using the AS733x EVK with a sensor and interface board. The SDK uses the ChipLib. (This is also an example of how to use it.) The SDK includes the following components:

- **ChipLib:** The AS733x Chip Library implements a driver for the AS7331 and AS73211 devices. It handles communication with the device and is used to configure the device and perform measurements. This software component is provided as a source code.
- **Spectral_library:** The AS733x Spectral Library includes exemplary functions for the correction and mapping of the sensor results into application-specific units.
- **Utilities:** This directory contains a helper code that is used by the other software components, such as error code definitions. It is provided as a source code.
  Sample_code and sample Calibration Files: This directory contains sample code using the software components provided in the AS733x Software Development Kit. An example of printing measurement data using the AS733x Chip Library is provided. The sample code is intended to be executed directly on a Windows computer. All examples are provided as source code and as executables.
  All the software components run on the computer, while the microcontroller on the board is only used as a bridge for I²C communication and to signal interrupts from the sensor.
- **Bin:** This directory contains tools that are needed for the making procedure.

- **OSAL (Operating System Abstraction Layer):** The software components of the AS733x Software Development Kit are designed to be used on different platforms. As the interface to access I²C buses, GPIO pins, and timer ticks differ per platform, each software component using such resources contains an OSAL, which contains the platform-specific code to access these resources. When porting a software component to a custom platform, only the corresponding OSAL needs to be adopted. All other source codes of the software component can be left unmodified. For the sample code provided, the AS733x Software Development Kit contains an OSAL library, which implements the OSAL for the AS733x Chip Library. The OSAL library, which is available as a binary only, uses the board as a bridge for I²C communication, and to signal interrupts from the sensor.

10 Additional Documents

The following list includes a selection of available documents with more technical details for the AS7331 Sensor and its test kits. 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, *AS7331 Spectral UVA/B/C Sensor* (DS001047), Datasheet.

11 Revision Information

Changes from previous version to current revision v3-00	Page
Figure 3, Figure 5, Figure 7, Figure 8, Figure 11, Figure 16 changed for B3 samples	6, 8, 9, 10, 13, 18
Figure 9 added	11
XYZ function replaced by A,B,C sensor channels	12, 26

- 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.

12 Legal Information

Demo Kits, Evaluation Kits and Reference Designs are provided to recipient on an “as is” basis for demonstration and evaluation purposes only and are not considered to be finished end-products intended and fit for general consumer use, commercial applications and applications with special requirements such as but not limited to medical equipment or automotive applications. Demo Kits, Evaluation Kits and Reference Designs have not been tested for compliance with electromagnetic compatibility (EMC) standards and directives, unless otherwise specified. Demo Kits, Evaluation Kits and Reference Designs shall be used by qualified personnel only.

ams-OSRAM AG reserves the right to change functionality and price of Demo Kits, Evaluation Kits and Reference Designs at any time and without notice.

Any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed. Any claims and demands and any direct, indirect, incidental, special, exemplary or consequential damages arising from the inadequacy of the provided Demo Kits, Evaluation Kits and Reference Designs or incurred losses of any kind (e.g. loss of use, data or profits or business interruption however caused) as a consequence of their use are excluded.

ams-OSRAM AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of ams-OSRAM AG rendering of technical or other services.

RoHS Compliant & ams Green Statement

RoHS Compliant: The term RoHS compliant means that ams-OSRAM AG products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories plus additional 4 substance categories (per amendment EU 2015/863), including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ams Green (RoHS compliant and no Sb/Br/Cl): ams Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material) and do not contain Chlorine (Cl not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents ams-OSRAM AG knowledge and belief as of the date that it is provided. ams-OSRAM AG bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. ams-OSRAM AG has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. ams-OSRAM AG and ams-OSRAM AG suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Headquarters

ams-OSRAM AG

Tobelbader Strasse 30

8141 Premstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Please visit our website at www.ams.com

Buy our products or get free samples online at www.ams.com/Products

Technical Support is available at www.ams.com/Technical-Support

Provide feedback about this document at www.ams.com/Document-Feedback

For sales offices, distributors and representatives go to www.ams.com/Contact

For further information and requests, e-mail us at ams_sales@ams.com