Quick Start Guide

QG000120

AquaSensor

Color and Turbidity in Liquids

AS7261 Application

v3-00 • 2019-Sep-11

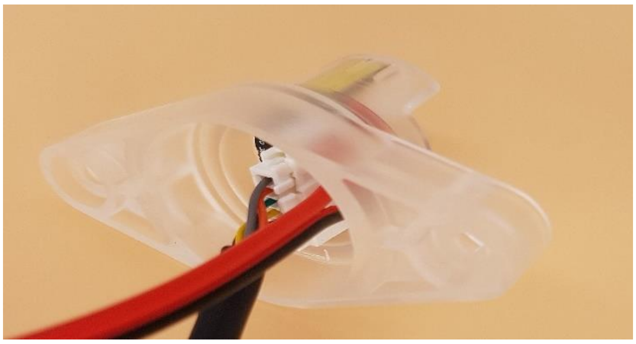
Content Guide

1 Basics of AquaSensor

The AS726x AquaSensor Demo Kit is an application specific sensor demo module for liquid measurements, including measurement of color, bio film or turbidity. Three interconnected boards make up the sensor and signal electronics. The main board includes two alternative LEDs and serves as the mechanical base for the sensor and primary LED board. By default, the sensor board contains the AS7261 multi-spectral XYZ sensor chip. Alternative sensor options are available for qualified project. The primary LED board is equipped also with two additional alternative LEDs identical to those found on the main. The 3-board solution is configured in a U-shape within a plastic housing, which can supports careful front-side immersion only.

> Note that the electronic in the housing is not water proof. Please assure that the electronic components do not come in contact with water or other liquids under test.

Figure 1:
Assembled AquaSensor



The sensor module connects to a power supply for the alternative LEDs as well as to the the FTDI cable and dongle which enables use of the PC user interface. The LED driver and sensor functions are available by using the standard AS7261 Demo Kit iSPI Software. For more details, see the software manual or the chapters for software in this document.

The sensor module was developed to measure liquids under alternative conditions in the visible and near IR wavelengths using the LED light sources in either direct or right angle optical paths between the ‘LED’ and ‘sensor’ through the plastic carrier and the liquid. All components of the module represent a demonstration and reference design to show typical results for such measurements. It is recommend using this reference as a first step in a feasibility project to check the application and verify external effects. Optimal results will be achieved by adjusting control parameters to achieve the highest accuracy and dynamic range for each specific application and its specific conditions. The key areas of optimization include LED choices, spectral filter selections (using different ams multi-spectral sensors, conversion setup and corrections for floating parameters. This document describes some typical aspects to help illustrate these.

2 Out of Box - AquaSensor

The main components in AquaSensor System pot of the Box are (see Figure 2):

1. Power supply
2. Main sensor board with embedded LED
3. Carrier case
4. FTDI-to-USB connecting cable
5. USB stick with software and documents:

• AS726x reference design.pdf	Reference-design
• AS726x.Spectral.Sensing.iSPI.Application.User.Guide.3V03.pdf	User Manual
• AS726x-iSPI.Evaluation.Kit.User.Guide.2V04.pdf	User Manual
• AN_AS726x.Design.Considerations.1V01.pdf	Considerations
• AS7261_CommandSet.xls	FW command set
• AS7261_DS000493_1-00.pdf	Data sheet AS7261
• CDM21216_Setup.exe	USB Driver setup
• AS726x_Spectral_Sensing_iSPI.exe	Test software

Please check if you have received all components before installation. Be careful by handling the blank electronic boards and consider all relevant ESD conditions. Electrical charges can destroy the electronics

Figure 2:
Components in AquaSensor Evaluation Kit

3 AquaSensor Test System

AquaSensor test system is for demonstration of spectral liquid measurements by using a color/spectral sensors. The system includes all components that are necessary to directly measure optical properties of liquids in an open container. A plastic housing surrounds the front of the AquaSensor assembly and is safe for use in most liquids. For accurate measurements, the sensor housing with the two horns on the left and right should be immersed in the liquid up to the end of each horn. Be careful and do not dip the open back of the housing into the liquids or damage will result.

Figure 4 shows the main components of the AquaSensor test system.

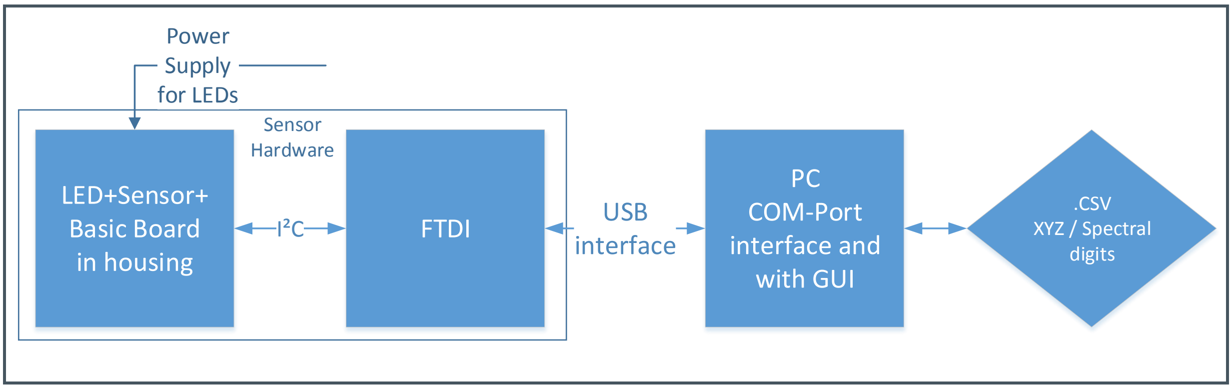
The plastic housing protects the hardware and creates the alternative optical paths from LED to sensor inclusive of the diffuser function in front of the sensor. The sensor hardware consists of three interconnected boards, which need an FTDI adapter (I^2C to PC via USB-connector and Com-Port Driver), and an additional power supply for the four alternative LED light sources.

The test software and GUI is the AS726x standard test software, which implements the sensor control via virtual I^2C register programming.

> Note: The AquaSensor hardware without FTDI adapter cannot be controlled by a µC in UART mode and AT commands because the hardware contains components (4xLEDs, 2x temperature sensors) that are directly connected to the sensor chip but are not supported by the standard AS726x firmware.

Figure 3 shows a block diagram of the electronically components and their interfaces.

Figure 3:
Block Diagram Showing the Connection Interfaces in AquaSensor



The main electronically components of the sensor system are the sensor AS7261, 2 x 2 LED combinations, an additional temperature sensor as shown and an adapter to connect the sensor to PC (see Figure 4). The LEDs can be used in two alternative optical paths – in transmission 0° and indirect in 90° reflection.

The sensor and LEDs are placed in the housing to measure the illuminated liquid around and through the plastic horns, once in transmission or alternatively indirectly illuminated under 90°. Therefore the LEDs (each a combination of VIS and NIR) are placed on the mainboard in 90° to the sensor or the LED board opposite the sensor in 0°.

Figure 4:
Concept and Main Components in AquaSensor

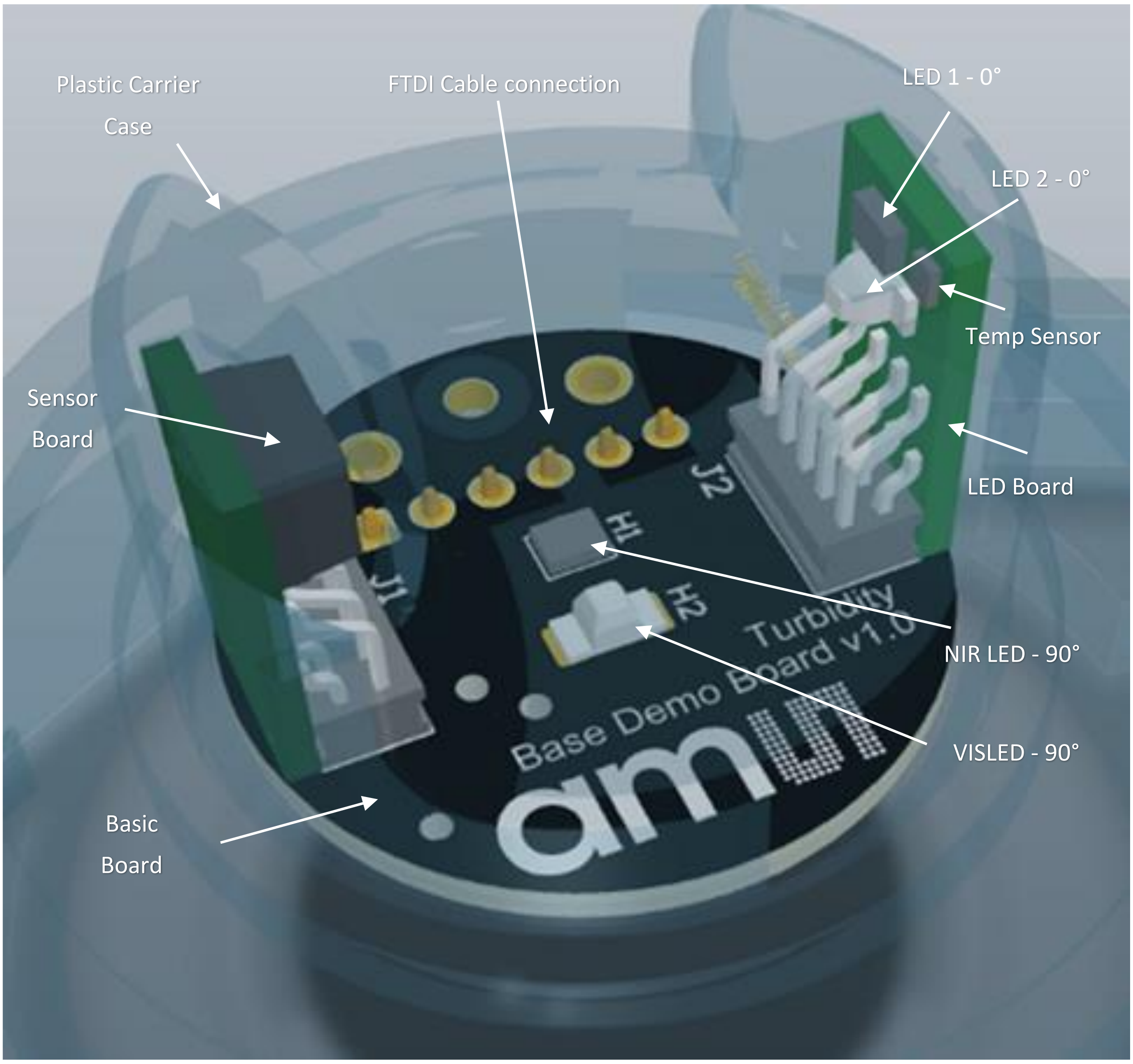
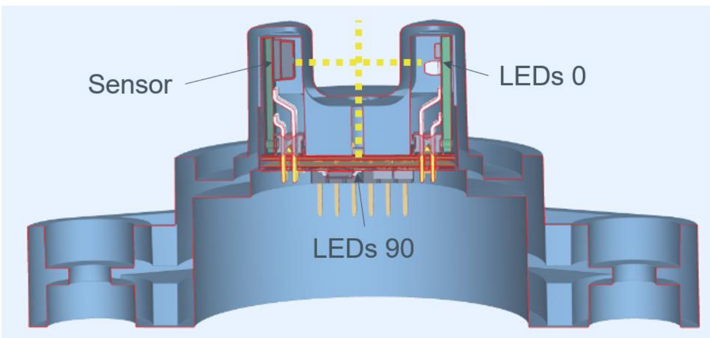


Figure 4 shows the main components of the AquaSensor after installing and mounting of all system components.

The LEDs on main and LED board are placed in angles of 0° and 90° to the sensor. Figure 5 shows the optical axes 0°/90° in the standard AquaSensor module, which hast to adjust by the application/customer specific requirements (limited/fixed optical axis from LED to sensor). For example, color or turbidity can be measured with high concentration better in transmission, whereas with lower concentration the 90° reflection achieves better results. Use also a dynamic gain (optimized setup for conversion) by variation gain and integration time (in the following named TINT) and LED driver currents. Use always such a setup to get high sensor signals and Full Scale Range (in the following named FSR). More details are available in the GUI manual or in chapter 7

> Note, the AquaSensor GUI can control the LEDs separately because it is based on an internal I²C programming. In custom specific designs, it is not possible to control the LED by AS726x usual AT commands because the AS726x pre-designed firmware was not prepared to address and control any bus participants. In such a case, use also I²C programming for extended bus configurations beyond the network structure specified in the AS726x.

Figure 5:
Optical Axes 0° and 90°



The sensor system as three interconnected board unit or module placed in plastic carrier case to measure under application specific conditions.

> Note, the plastic carrier was developed and manufactured as a demo tool for various alternative applications in liquid. The optical axes represent a compromise between accuracy and flexibility as well as the selected LED/sensor combination, which must be adapted to the requirements of the application. The housing material is a prototype material chosen as diffuser without accuracy requirements for measurements. Therefore, the material, form and all other conditions, whose affects the measurements, must be adapted by the application specific requirements.

The I²C and UART communication interface on the board makes the system compactable with PC and µC applications. A special device driver on PC is necessary as well as the iSPI test software, which was adapted. The details regarding the Sensor system in software and it functionalities can be read from the data sheet and other documentation available in the USB sticks.

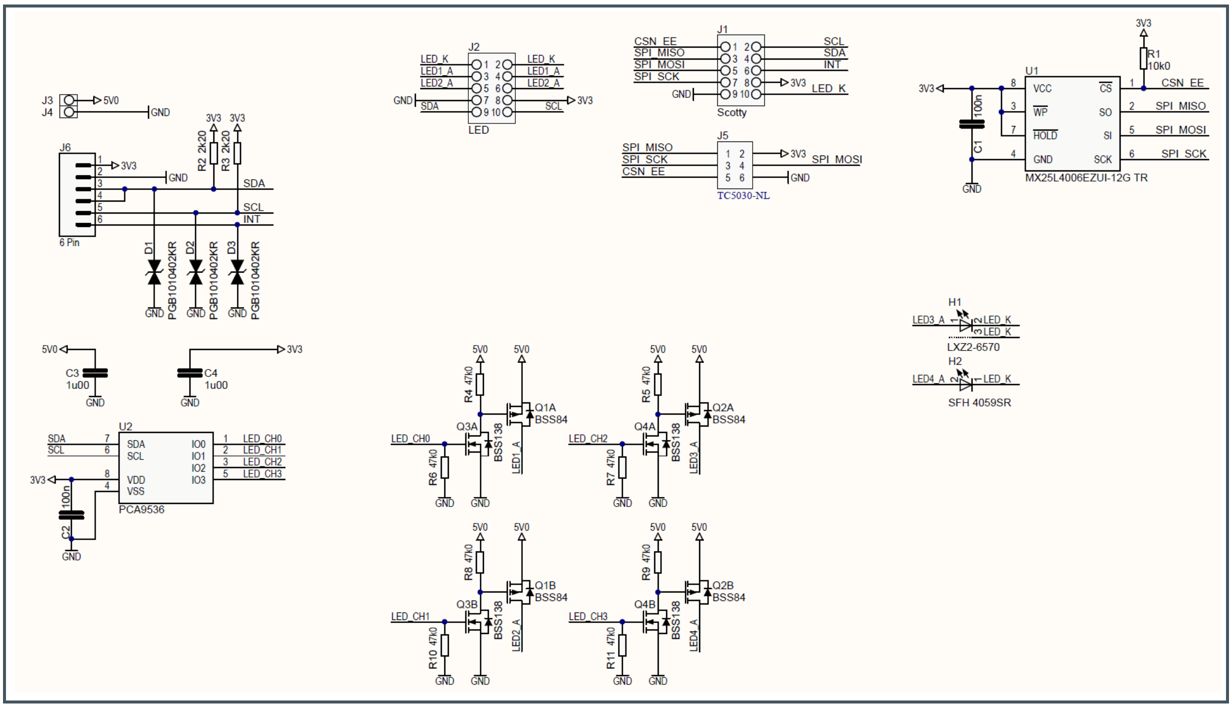
Note, there are differences in hardware and software between the standard Evaluation Kit AS726x and AquaSensor.

3.1 Mainboard

The Mainboard is used to connect the LED / Sensor board as well as the sensor unit in the housing. Further, it contains two alternative LEDs on board (VIS/NIR 90) which useable as indirect light source.

> Note, the mainboard contains more components like suggested in the AS7261 reference design, e.g. I²C splitter PCA9536 to realize the 1:4 LED function.

Figure 6:
Schematic AquaSensor Mainboard

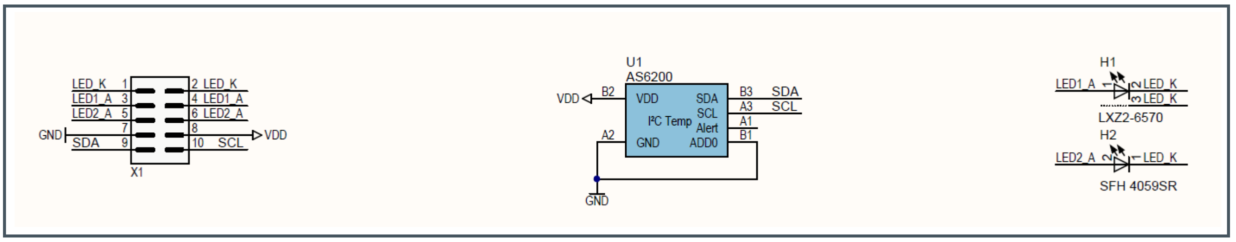


3.2 LED Board

The LED board contains the two alternative LEDs for the VIS and NIR range plus an external temperature sensor to control possible temperature shifts of the LEDs. Figure 7 shows the schematic and types of the mounted LEDs.

> Note, the APP team of **ams** can support you to select or change the actual used LEDs by another LED based on special requirements.

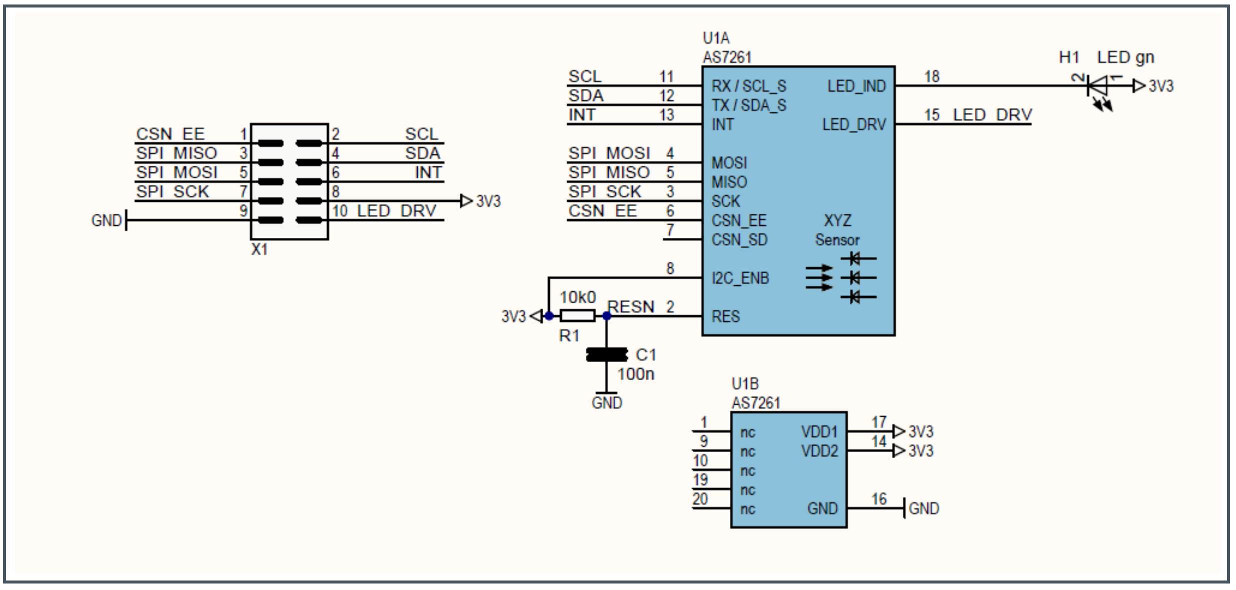
Figure 7:
Schematic AquaSensor LED Board



3.3 Sensor Board

Figure 8 shows the schematic of the sensor board. The AS7261 sensor (part of the AS726x family, it can be changed by alternatives of this family) is used as standard for the demo because the sensor combines XYZ filters (based on CIE1931 standard), a clear filter (CMOS sensitivity), a width banded NIR channel and an internal temperature sensor to control any ADC shifts. The integrated spectral optical filters consist of stable Nano-optic interference filters, non-depend on temperatures and without changings over lifetime.

Figure 8:
Schematic AquaSensor Mainboard

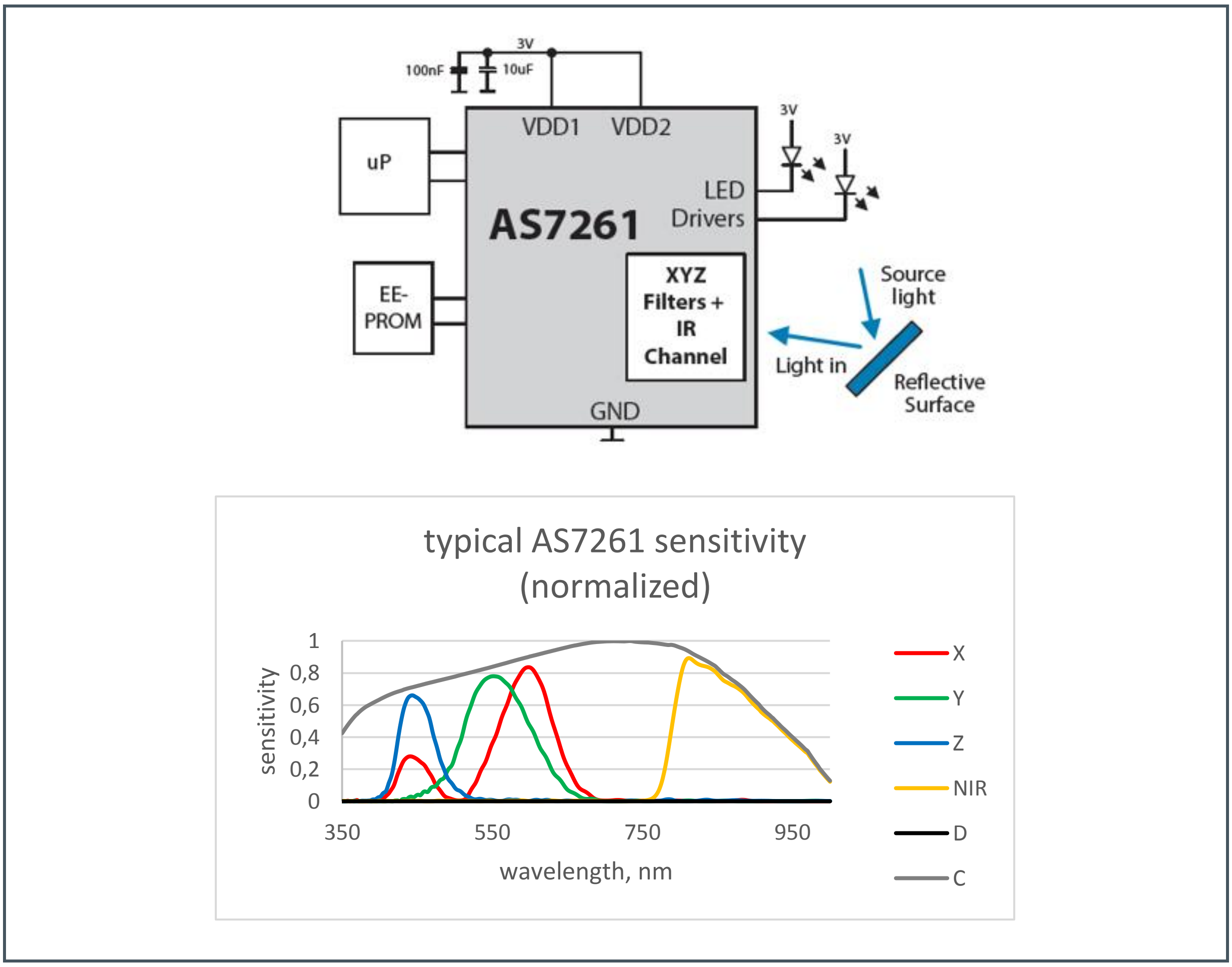


The AS726x sensor family based on a special system architecture that includes six-separated optical receiver with signal amplifier and converter, an 8051 core, various interfaces, a temperature sensor and other functions as LEDs drivers and for LED indication. The Integrated LED driver consist of two VIS LEDs and two NIR LEDs with a programmable current[1] from 12.5 mA to 100 mA.

Depend on the special sensor type the optical receivers characterized by the type-special spectral filters and can realize so alternative applications. Listened here, the AquaSensor includes the XYZ extended with Clear and NIR sensor AS7261. This sensor can be changed very easy to another AS726x, e.g. by the 6 banded spectral sensor AS7263. Therefore, a spectral NIR measurement will be possible by (only) adaption of LED and sensor.

The following figure shows block diagram of the AS7261 sensor and its filter diagram.

Figure 9:
Block/Filter Diagram of AS7261 Used in AquaSensor as Standard



---

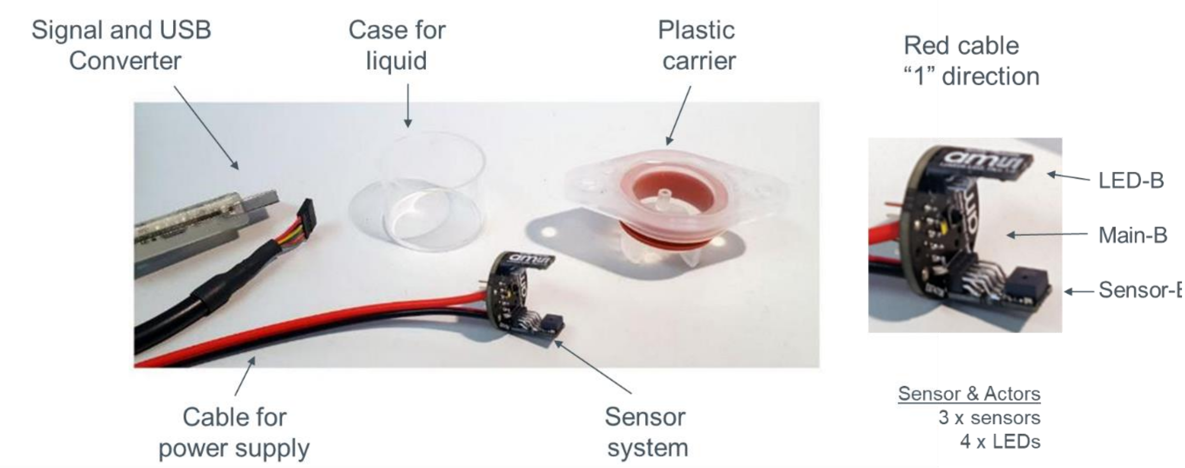
[1] PCB design of LED and Main Board was not prepared to optimal heat dissipation. Be careful with higher driver currents as results of which a LED can be damaged.

4 How to Build and Assemble the Setup

4.1 Mechanical Setup

- Make the Connection between the power supply plug and Main Sensor board. This supply power to LED driver on main sensor board
- Connect the FTDI Connecting Cable to white connecting slot in main board. USB end of FTDI cable is connected to PC and is main interface to sensor and its power supply. AquaSensor and its software is designed I^2C communication interface.

Figure 10:
Not Mounted Mechanical Components



4.2 Software Installation

4.2.1 Items Included in the USB Memory Stick:

- AS72xxx Flash Update folder contains the details regarding various methods to flash firmware to device
- AS726x Documentation Set includes Design consideration, datasheets, application note and user guide for iSPI software
- AS726x FTDI USB-MPSSE Cable Driver contains FTDI driver file which should be installed once
- AS726x Spectral Sensing iSPI GUI includes the software and related files
- AS7261 Firmware contain the binary files for the sensor and I^2C and UART command set details
- AS726x-iSPI Evaluation Kit readme.txt

4.2.2 Installation Procedures

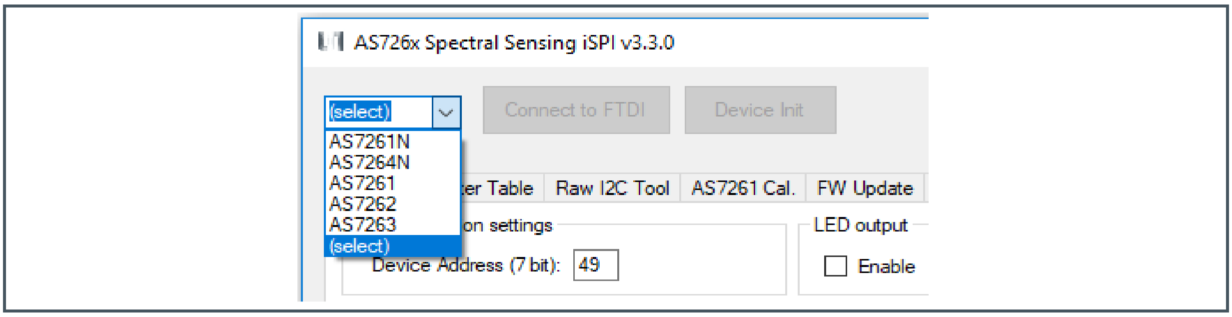
- The evaluation kit requires one time installation of FTDI CDM Driver for the USB-MPSSE cable if it is still not installed on the computer. The installation file can found in the USB Memory Stick.
- Connect the FTDI Serial Cable to AS726x AquaSensor board and USB end into PC.
- The AS726x Spectral Sensing – iSPI software does not require separate installation. Simply copy the following files to any folder on the host system and start the .exe file to start the GUI. The .NET Framework is the programming framework used to develop this application software. Thus, this software application requires 4.5 or above version of .NET Framework to be installed. It is recommend install the most up-to-date version of .NET, assuming your OS supports it.

> Note: The details regarding different functionalities of GUI Software and other documentation could read from the above mention documents in USB.

5 Software Operation

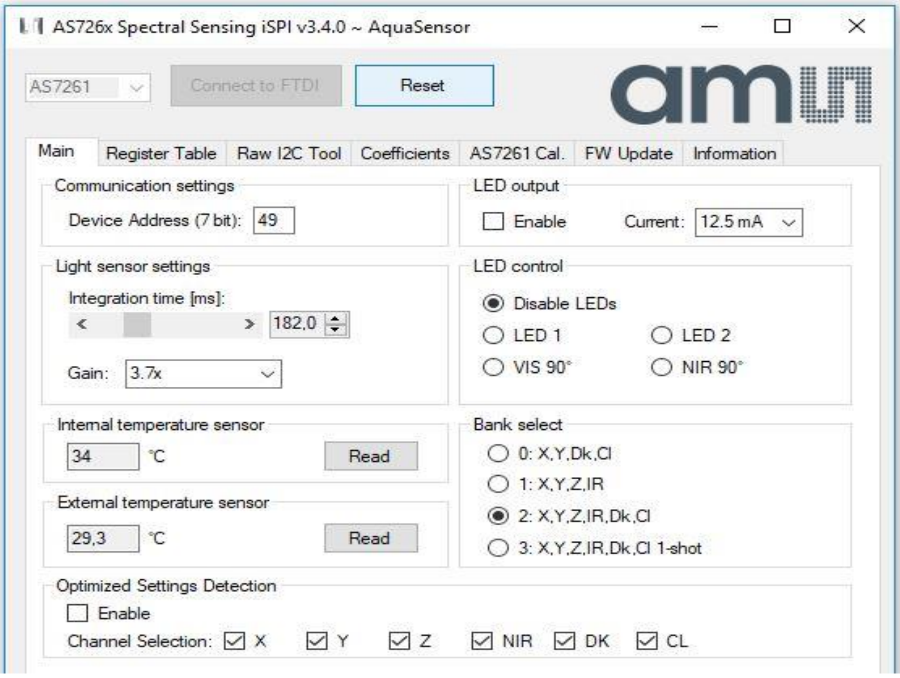
After launching the application, select the board from the top left combo box as AS7261 (as standard or select a customer specific version) as shown in Figure 11. Press the button “Connect to FTDI” and the button “Device Int” to initialize the sensor test board. An error message will printed out in case of an issue (for more details please check the software manual).

Figure 11:
Combo Box Selection for AquaSensor – AS7261



The window shown in Figure 12 and referred as Main Page will be launched when communication with the sensor has been successfully started and a hardware code named 'Device HW' is printed in the footer. Please go through the user guide and Application note in USB sticks to know the GUI software.

Figure 12:
Main Page for AquaSensor – AS7261



In the following, the specific functions of AquaSensor will described.

A hardware code is shown in the foot on the right side after selection AS7261 (or alternatives), pressing the button 'Connection to FTDI' and 'Initialize'. If not, then check the driver installation, USB connection and power supply. See in the manuals 'AS7261 test kit' for more details.

In general, the following steps are necessary to make stable and accurate measurements:

Steps of Feasibility

1. Select Sensor – Luminary combination
2. Stable SETUP
   ➔ e.g. fixed and shielded test system
3. Optimized parameter setup
   ➔ e.g. dynamic gain
4. Eliminate interferences
   ➔ e.g. temperatures, interfering light sources, drifts
5. Match sensor system into application
   ➔ reference and calibration

The optimized parameter setup have to select after selection the LED/Sensor combination and completion of the sensor module with running software.

In order to switch on the LED it is important to check, if enable and LED currents is set. The current can be set as 12.5mA, 25mA, 50mA, and 100mA. User can select the LEDs. Four LEDs on board can enabled separately. Two 0° LEDs and two 90° LEDs, named and defined as following:

- LED1 - mounted ion LED Board and normal White Light LED
- LED2 - mounted on LED Board and NIR LED
- VIS 90° - mounted ion LED Board and normal White Light LED
- NIR 90° - mounted ion LED Board and NIR LED

The LEDs switch on as flash (on/off) in case of LED control is not enabled but a measurement will start (Flash mode).

The parameter setup for conversion is possible by changing GAIN (amplifying) and TINT (integration time). An increasing of GAIN and TINT results higher digital counts as RAW values and sensor results; the higher the counts the better the accuracy. Note, FSR needs a minimum of 180ms to reserve 16bits for counting the RAW values. On the other side, prevent saturation, which is depend on the TINT (reserved number of bits) and the luminous intensity on the sensor. Following table shows the connections between TINT, number if bits and max. counts (saturation).

Figure 13:
Connections between TINT, Number of Bits and Max. Counts

TINT (ms)	Resolution (bit)	Max. Counts
2.8444	10	1024
5.6889	11	2048
11.3778	12	4096
22.7555	13	8192
45.5111	14	16384
91.0222	15	32768
182.0444	16	65536
364.0888	16	65536

Raw values represents the counts from the ADC depending on the setup used (Gain, Integration time, LED-current). Basic sensor values are divided by codes, which are determined by these setup parameters. Therefore, Basic sensor counts are not depend on setup parameters or on their changing in case of using dynamic gain or Optimized Settings Detection. Note, there are effects from temperature or other interferences, which must be considered in case of dynamic gain (gain error, temperature curve, TINT linearity, synchronous operation of channels).

Basic sensor values are corrected also by multiplying raw values with temperature coefficient, which is divided by the register scaling values of Gain and Integration time.

The file ‘temperature_compensation_sheet.csv’ file contains the temperature coefficient values from 0°C to 100°C. While calculating the basic values the corresponding internal temperature coefficient value is take from the .csv file. User can define these values, by default it is one in .csv file. Calibrated values are calculated from the raw values based on the factory calibration. Corrected value and NTU values calculated based on raw value, the init_file values and its formulas.

Figure 14:
Example of INIT_File

```
// Minimum and maximum of counts [0..65536]
MinCounts=10000
MaxCounts=60000

// Maximum of gain (register value [0,1,2,3])
MaxGain=3

// Scaling of gain depending on the register values
Number of values = 4
// Default values: 0 = 1, 1 = 3.7, 2 = 16, 3 = 64
GainScaleList=1,3.7,16,64

// Minimum and maximum of integration time (register value [1..255])
MinIntegrationTime=65
MaxIntegrationTime=255

// Maximum of difference between x and x_1931 as well as y and y_1931 (only for AS7261)
MaxXyDifference=0.1
```

The init_file.txt file is present in the same folder where the .exe file and is important to have it in the folder. The initialization file is used for locating and using the defined values of certain parameter data that is used in GUI. The init_file contains the MinCounts, MaxCounts, GainScaleList, MinIntegrationTime, MaxIntegrationTime, White and black reference values for the six channels and Parameters for NTU calculation. The user can define the minimum and maximum counts for the optimization (dynamic range) of raw data in MinCounts and MaxCounts respectively. An important remark is that the scaling of gain and integration time depends on the register scaling values in firmware. The “Optimized Settings Detection” works based on these values in init_file.txt.

Save White Reference and Save Black Reference buttons user can save the data for Black and white reference respectively only temporally. With closing the GUI, these values will be erased.

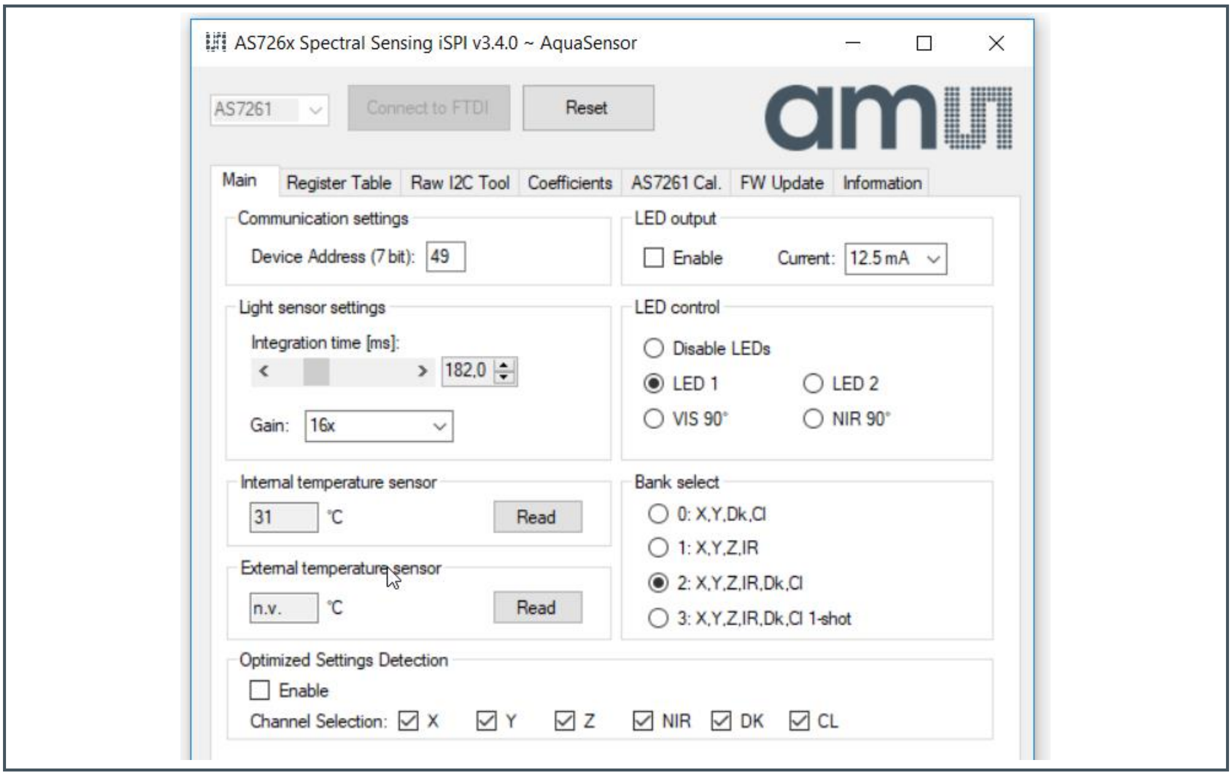
For details of these calculations or unknown procedures, please contact the **ams** customer support.

6 First Test

The following based on a connected AquaSensor and running test software with the setup shown in Figure 15.

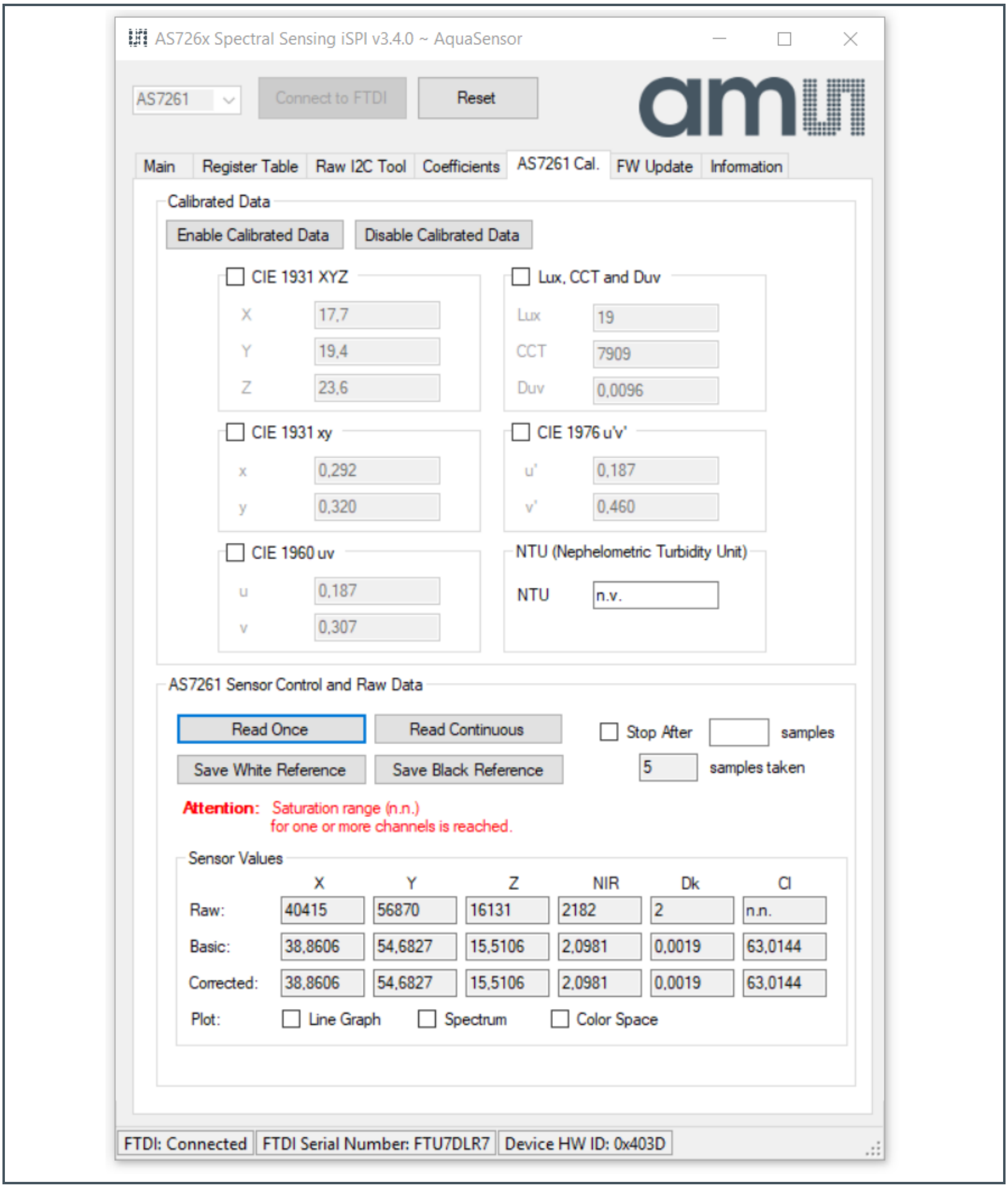
6.1 First Shoot

Figure 15:
Main Window First Step – Select TINT = 182ms, Gain = 16, LED control = LED1



Go to the Windows ‘AS7261 Cal’ and press Button ‘ReadOnce’. You should get nearly the following results (see sensor results Figure 16).

Figure 16:
Read and Compare Sensor Results[1]



(1) The results in the “Calibrated Data’ are only applicable if the sensor system was calibrated based on CIE1931 conditions and the application is CIE1931 True Color Measurement. The software cannot check whether these conditions are valid. It is recommended to use this result block ‘Calibrated data” only if the conditions are valid and to use the results.

Note, the LED was ‘selected’ but not ‘enabled’. Therefore, the LED is switched on only shortly during the measurement (Flash mode).

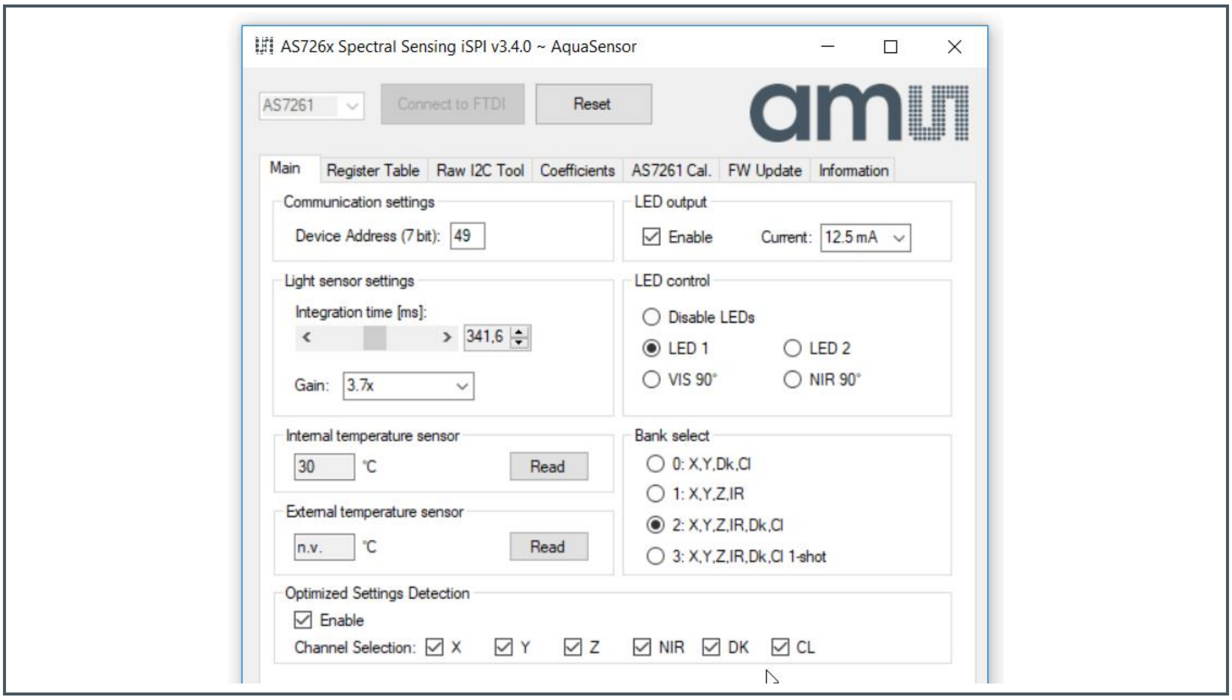
Based on the parameter setup, the raw counts for XYZ are in the upper regions of FSR, Clear and Dark[2] filters have a low response because a white LED is used and Dark should be low. The Clear channel is in saturation and out of range. Reasons for (smaller) deviations from the sensor results in the figure are standard deviations of the LEDs (bin) and sensors (non-calibrated values RAW values and different LED types), other conditions in environments (interference light) or others.

In kind of a high deviation from a standard AquaSensor in the process of a first step and the results shown here, please contact **ams** support.

6.2 Optimized

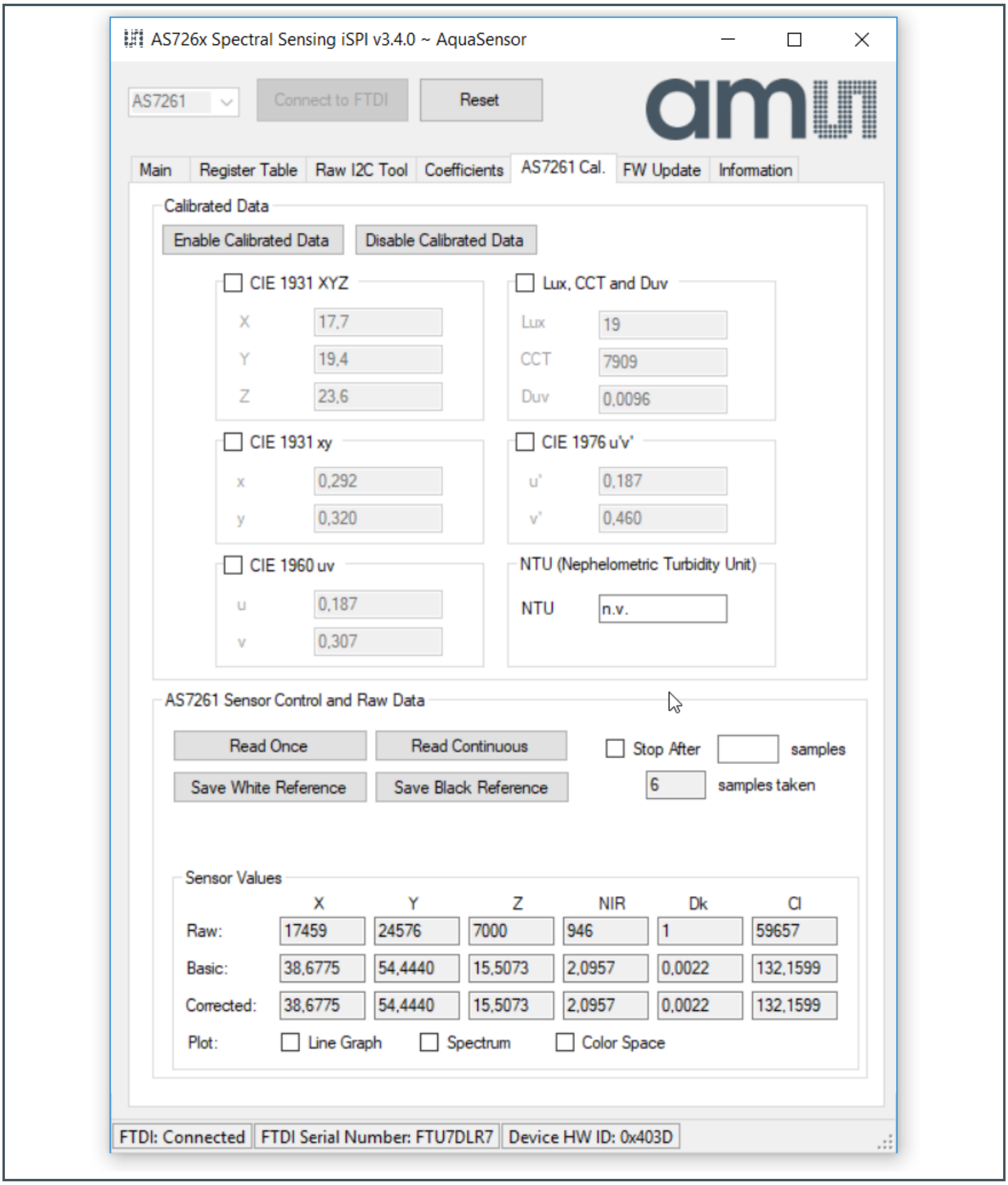
Now, go back to the Main window and switch on ‘Optimized Settings detection’. The software enables the automatic algorithm for ‘dynamic gain’ (see also chapter 7.2). Make a ‘Read once’ and compare the settings in the ‘Main windows’ and the sensor results in the ‘AS7261 Cal’. Your results should be similar. See the reduced GAIN as the software answer for saturation in Clear, compared with the other setup.

Figure 17:
Enable Optimized Settings Detection (with the optimized setup here)



[2] Read the Sensor data sheet for details

Figure 18:
Typical Sensor Results



Measure different probes by using optical grey filters or liquids with different turbidity concentrations, e.g. from pure water to dark and/or 4000NTU. Use the dynamic gain (optimized settings detection, see chapter 7.2), open a protocol file to log the data's and use the basic counts in case of dynamic gain to be not depend on the used setup for gain, integration time and LED currents. The measured sensor results can use for calibration after closing the protocol file. Make a black/white calibration to correct the existing variations of the LEDs (see chapter 7). These corrected values are the basic to match the sensor results into the application, e.g. by curve fitting to get NTU (turbidity concentrations) from the measured sensor counts. See the following example.

Figure 19:
Example 1 (measured basic counts of different sensors for alternative concentrations)

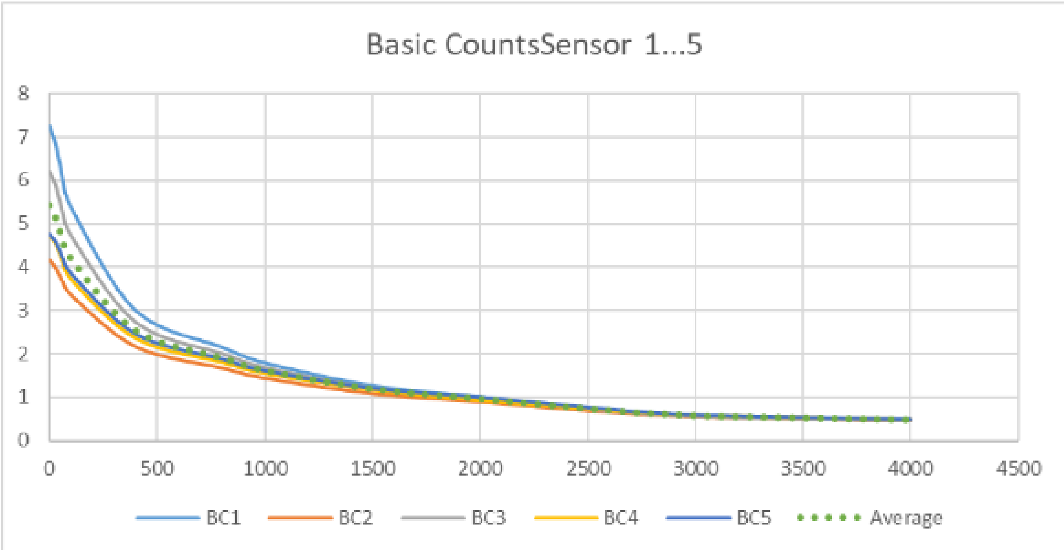
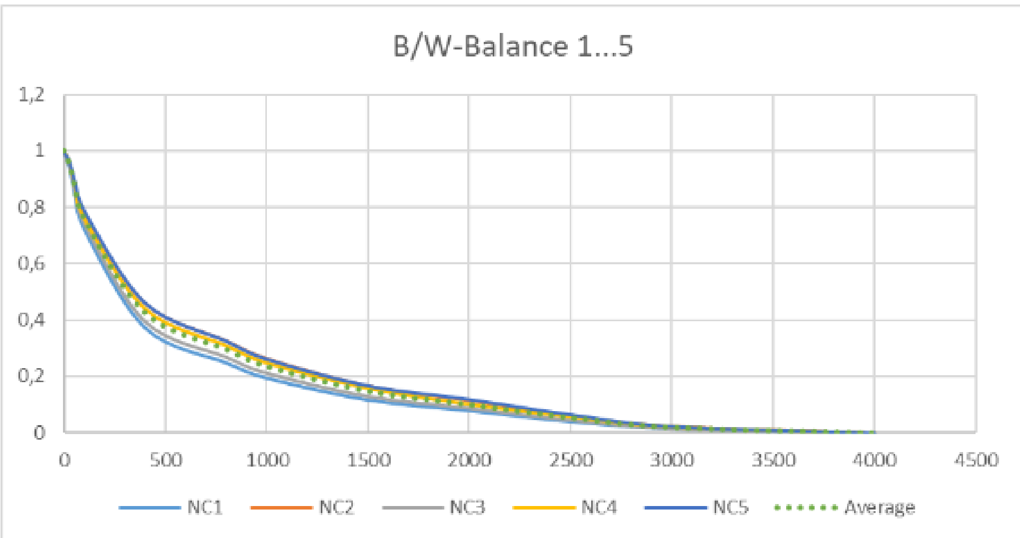


Figure 20:
Example 2 (basic counts after Black/White balance)



7 Example from Basic Counts Up to NTU

This chapter describes corrections and how to implement white-black calibration process (can be named also as 1-0 Normalization or black-white balance) after closing the measurement protocol to get sensor corrected results and sensors uniformity.

These corrected values are the basic to match the sensor results into the application (here for turbidity NTU). This process is named curve fitting and includes creation and optimization of an equation, which emulates the measured sensor results after the black/white correction as function 'Sensor Counts and NTU'. Then turbidity will be measured from the best-fitted equation.

Here with an example, it will be explained the whole procedure about white-black calibration and Curve Fitting, based on three sensor samples, with matching the sensor results from 0 NTU (Pure Water) to 4000 NTU turbidity.

7.1 Basic Counts

After taking the measurement, put all values from all sensor into one excel file as shown in the below.

Figure 21:
#Sensor_1

NTU	X	Y	Z	NIR	DK	CL	TINT [ms]	Gain	LED Current [mA]
0	12798	22824	4899	636	1	60049	607.6	1x	12.5 mA
100	13121	22492	4650	682	1	59477	196	3.7x	12.5 mA
200	13783	22431	4501	719	1	59386	232.4	3.7x	12.5 mA
300	14369	22588	4376	763	1	59866	277.2	3.7x	12.5 mA
500	15193	22243	4180	827	1	59681	364	3.7x	12.5 mA
800	16967	22154	4130	932	1	59947	548.8	3.7x	12.5 mA
1000	17890	21994	4152	992	1	59895	700	3.7x	12.5 mA
1500	19428	21941	4405	1083	1	59768	249.2	16x	12.5 mA
2000	20695	22258	4868	1118	1	59984	347.2	16x	12.5 mA
2500	21395	22513	5319	1104	1	59785	445.2	16x	12.5 mA
3000	21624	22726	5622	1080	1	59819	515.2	16x	12.5 mA
4000	22299	23556	6357	952	1	59361	184.8	64x	12.5 mA

The following graph shows the starting point which is 0 NTU (White Balance) and ending point 4000 NTU (Black Balance) with the identical coordinates for all sensors.

Figure 22:
#Sensor_2

NTU	X	Y	Z	NIR	DK	CL	TINT [ms]	Gain	LED Current [mA]
0	11108	23304	5513	764	1	59924	624.4	1x	12.5 mA
100	11960	23262	5282	814	1	59873	204.4	3.7x	12.5 mA
200	12556	23123	5056	859	1	59891	240.8	3.7x	12.5 mA
300	13658	23306	4982	852	1	59842	285.6	3.7x	12.5 mA
500	14420	22898	4685	915	1	59530	364	3.7x	12.5 mA
800	16637	23077	4602	1021	1	59782	551.6	3.7x	12.5 mA
1000	17604	23026	4579	1131	1	59925	694.4	3.7x	12.5 mA
1500	19251	22924	4726	1182	1	59459	238	16x	12.5 mA
2000	20597	23260	5130	1229	1	59846	333.2	16x	12.5 mA
2500	21484	23646	5603	1192	1	59878	420	16x	12.5 mA
3000	21862	23907	5901	1179	1	59979	492.8	16x	12.5 mA
4000	22510	24686	6629	967	0	59568	179.2	64x	12.5 mA

Figure 23:
#Sensor_3

NTU	X	Y	Z	NIR	DK	CL	TINT [ms]	Gain	LED Current [mA]
0	19031	24357	6573	663	1	59850	635.6	1x	12.5 mA
100	18810	24290	6040	700	1	59612	204.4	3.7x	12.5 mA
200	19194	24381	5830	720	1	59801	243.6	3.7x	12.5 mA
300	19305	24249	5436	757	1	59743	285.6	3.7x	12.5 mA
500	19758	24051	5044	840	1	60020	375.2	3.7x	12.5 mA
800	20411	23482	4733	915	1	59801	560	3.7x	12.5 mA
1000	20541	23044	4577	969	1	59828	708.4	3.7x	12.5 mA
1500	21323	22654	4732	1079	1	59655	252	16x	12.5 mA
2000	21756	22487	5097	1103	1	59584	350	16x	12.5 mA
2500	22202	22740	5598	1099	1	59792	456.4	16x	12.5 mA
3000	22275	22899	5863	1081	1	59882	518	16x	12.5 mA
4000	22605	23517	6568	936	0	59111	182	64x	12.5 mA

The basic counts are calculated by following formula:

Equation 1:

$$BasicCounts = \frac{Raw\ Counts}{Gain\ *\ TINT\ *\ LED\ current}$$

Basic counts are not depending on the parameter setup and should be used as sensor results in all following calculations.

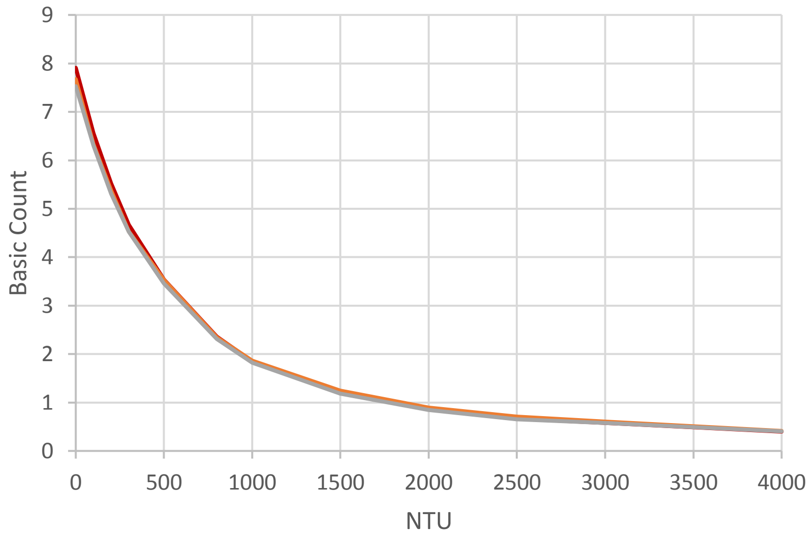
> Note, Basic counts should be corrected if the sensor system has any drift over temperature, time or by using different gains (see chapter 7.2).

In the table below are basic count from all three sensors based on clear channel for this example. The channel selection must be done application specific and depending on the used lamination and application to get maximum of counts and high accuracy. An optimum is depending on the used luminary, target and planed accuracy. Select the channel by a maximum of sensor filter luminary overlapping, number of affected filters and slopes and interferences. Target is to get a maximum of counts in case of minimal changing of sensor/luminary/target overlapping.

Figure 24:
Table Measured NTU and Counts of All Sensors

NTU	Sensor_1	Sensor_2	Sensor_3
0	7.90638578	7.677642537	7.53304
100	6.561169333	6.333421484	6.305813
200	5.525050007	5.377660052	5.30786
300	4.66955267	4.530395942	4.522901
500	3.545054945	3.536085536	3.458768
800	2.361791821	2.343335359	2.308919
1000	1.85003861	1.86589239	1.826056
1500	1.199197432	1.249138655	1.183631
2000	0.863824885	0.89804922	0.8512
2500	0.671439802	0.712833333	0.655039
3000	0.580541537	0.608553166	0.578012
4000	0.401521916	0.415513393	0.405982

Figure 25 :
Diagram Measured NTU and Counts for Sensor 1 … Sensor 3 (3 Sensors shown here nearly about each other)



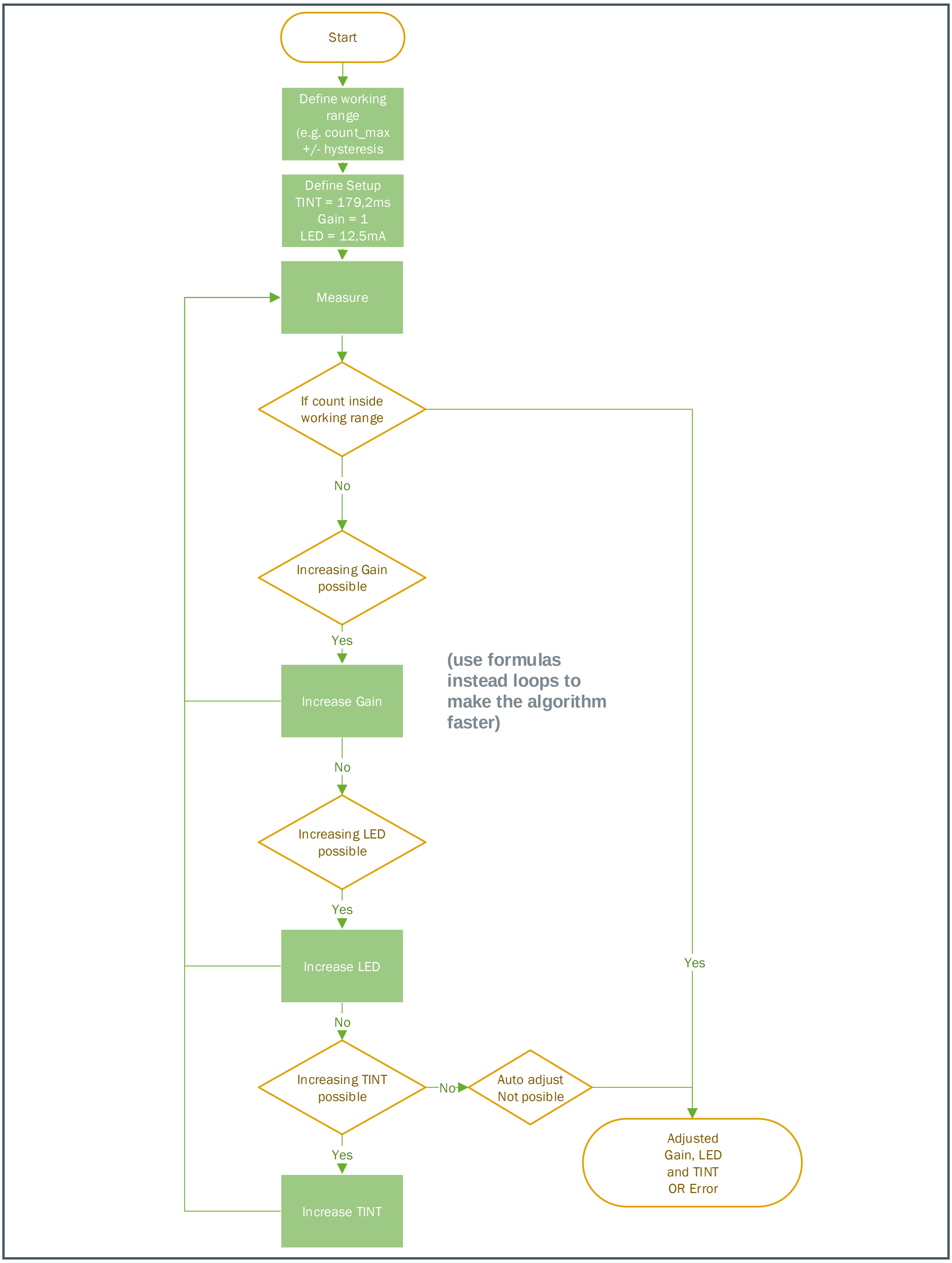
7.2 Dynamic Adjustment and Corrections of Basic Counts

For applications with a high dynamic range, an automatic adjustment of the conversion parameters is recommended in order to ensure that the meters always operate in the optimum working range. The parameters are gain, integration time and LED driver current. Each of these parameters has its own factors, causes different amplifications but also affects the system results. An increasing of LED currents means a higher temperature of the LED and neighbored elements. Higher gain results higher noise but increasing integration time means higher measurement time. Therefore, increasing parameters gain or integration time must be selected application specific by its effects and disadvantages. Besides, since the parameter setting and result are not always exactly linear or vary with temperature, a signal correction must be done in general when using dynamic adjustment.

Automatic adjustment needs a setup to start the process and a target, which represents the workings range. This range will be specified by a hysteresis to prevent an oscillation system.

The dynamic amplification works starting with a setup, its measurement and the comparison of the results with the optimal working range. Once this has been achieved with the setup, the dynamic process is successfully completed. Otherwise, the parameters are incremented until the optimum working range before saturation is reached. It is recommended to use the parameters in the order of their gain levels to quickly get the sensor out of DARK. Then the working range is adjusted FINE. Since gain works with factor 4, LED current works with factor 2 and the integration time contains very small steps, this order of the parameters is also recommended. Figure 26 shows the design flow of an automatic conversion which customer must program in firmware.

Figure 26:
Design Flow of Automatic Conversion



The result of the automatic adjustment are an optimized setup for the selected parameters to get counts in the specified dynamic range. Next step is the correction of the measured counts based on the used parameters before they will used in the calibration process. The necessary correction factors must be determined in advance in laboratory tests. Here it is recommended to carry out these tests on many sensors, on as many production batches as possible, and then to work with mean values. A better more accurate method is the device correction, where all parameters for the correction are measured directly on the device. This guarantees the highest accuracy.

All tests must be specified and done application specific by its requirements. Therefore, in the following only two short examples are shown for temperature and gain correction. Other tests are similar.

Making temperature test and correction: Use a stable light source in a climate chamber with the sensor. Set gain and integration time (min 179,2ms) to get counts >5k. Start with a min temperature and log temperature and counts. Measure over the full temperature range up to max temperature. Use these results of counts depending on temperature changing as look up table correction factor / °C (e.g. 0°C – 1.0; 10°C – 1.0206; 20°C – 1.0412; ….) to correct temperature drifts with the formula

```
counts_new (of temp e.g. 20°) = counts_new * correction factor (of 20°C)
```

Making GainError test and correction: Use a stable light source, an integration time (max as possible to get evaluable counts and gain = 1. Start the measurements and increase gain systematically from 1 to 3.7 to 16 to 64. The Gain_error is the factor for each gain change, which is the result from the formula:

```
gain_factor(gain_new) = (gain_new / gain_old) / (counts_new / counts_old)
```

Use this formula to correct the counts based on the calculated gain_factor.

```
counts_new = counts_new * gain_factor(gain_new)
```

The following figures show a simple example for the correction of two alternative measured counts.

Figure 27 :
Example of Temperature CSV and Gain Correction List

degree	0	1	7	8	9	10	…	81	82	83	84	85
factor	1,000	1,002	1,014	1,016	1,019	1,021	…	1,167	1,169	1,171	1,173	1,175

Gain	Gain Factor	
1	1	The red marked numbers are used in the example in the next figure
3,7	0,9973456	
16	0,9788965	The gain factors were calculated from test results by varying the gains under
64	1,0559696	fixed conditions and comparison with the increasing of the counts

Figure 28 :
Example Count Correction as Excel Sheet

	A	B	C	D
1		**Value 1**	**Value 2**	
2	input RAW values - counts	13009	42224	Value 1/2 represents alternative measurements/counts
3	input used Gain	1	3,7	of a constant light source, by using different gains
4	input used TINT in ms	182	182	identical TINT
5	input temp in°C	7	84	but under different temperatures
8				
9	Gain	Gain Faktor		This table is the result of a gain test
10	1	1		
11	3,7	0,9973456		
12	16	0,9788965		
13	64	1,0559696		
14				
15	**Results**	**Value 1**	**Value 2**	
16	**RAW Counts**	13009	42224	=C2
17	**Basic Counts**	71,5	62,7	=C2/($C$3*$C$4)
18	**temp_corrected**	72,4	73,4	=VLOOKUP($C$5;'csv temp'!$A$1:$G$102;2;TRUE)*C17
19	gain_corrected	72,4	73,2	=C18*VLOOKUP($C$3;$A$10:$B$13;2;TRUE)
20	Diff			
21	1,2%	72,4	73,2	the originally differently measured counts are almost completely identical after correction and calculation as basic counts

7.3 Black-White-Balance

Normalization or balancing must be done to eliminate technology and manufacturing deviations in sensing results and to get sensor uniformity. For this example, the table of measured sensor values shows different results of sensor counts between 0 NTU and 4000 NTU. Sensor and LED variations, temperature drifts, ageing can be the reason for the differences in the sensor results.

For this example, the method Black-White-Balance was selected to match the sensor results into a defined dynamic range. This range will be the basic for the matching process that means to find a correlation between sensor results COUNTS and application specific units for turbidity NTU.

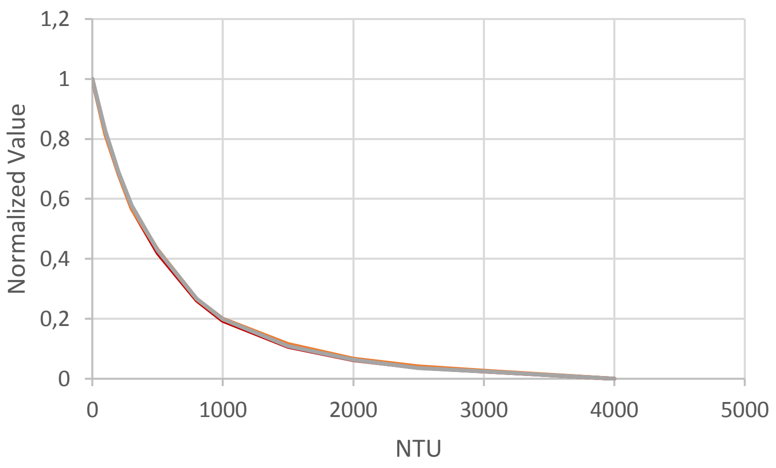
After normalization, the table will be as shown below.

Figure 29:
Sensor Results After Black-White-Balance

NTU	Sensor_1	Sensor_2	Sensor_3
0	1	1	1
100	0.820754051	0.814899869	0.827807
200	0.682694341	0.683290886	0.687784
300	0.568701955	0.566622056	0.577646

NTU	Sensor_1	Sensor_2	Sensor_3
500	0.418866096	0.429704854	0.428337
800	0.261199929	0.265462363	0.267002
1000	0.193010389	0.19971815	0.199251
1500	0.106287806	0.114790752	0.109112
2000	0.061600447	0.066445503	0.062469
2500	0.035965727	0.040941153	0.034945
3000	0.023853813	0.026581705	0.024138
4000	0	0	0

Figure 30:
Diagram Sensor Results After Black-White-Balance (3 Sensors shown here about each other)



The graph shows the starting point, which is zero NTU (White Balance), and the ending point with 4000 NTU (Black Balance), identical for all sensors but a little different between 0 and 4000 NTU.

> Note, these differences will affect the sensor system accuracy after matching and should minimized so much as possible.

The device specific Black-White-Balance should correct the sensor variations based on the mounted sensor system. Therefore, it is recommend making the Black-White-Balance during the series end test of the sensor modules. In this case, the measured values for Black and White must be stored on the sensor to readout them by Sensors Firmware if these values are necessary for the calculation of turbidity (see Chapter 7.4).

The counts and/or calculated values for the sensors are the basic for the matching and correlation counts into NTU.

7.4 Curve Fitting

Curve fitting will find out a formula as correlation counts and NTU based on the normalized values of the Black-White-Balance. MCU needs this formula with the device specific parameters to make a calculation NTU values according to sensor-normalized value. The process curve fitting is to get the best-fit line or curve for the series of data points (see Figure 30). This curve fitting will produce an equation that can be used to find the points anywhere along the curve.

The curve fitting will affected by the formula type and its parameters. For the used example, the following formula was fitted as optimal solution.

Equation 2:

$$y = an * x\hat{}n + \ldots. + a2 * x\hat{}2 + a1 * x + a0$$

y = calculated NTU
x = counts
a3…a0 = Spec. Sensor parameter from Golden Device, type, lot, or device specific

The type of formula describes the principal correlation between counts and NTU. The parameters determine the accuracy of the correlation. The better the formula and especially parameters map the correlation the better will be the result and sensor accuracy. Therefore, a using of sensor device specific formula and parameters consider the device specific deviations and promise the highest accuracy. On the other side, it needs the highest effort in matching and device calibration. A type or lot fitting cannot consider the exemplary deviations and deliver an averaged accuracy by a lower effort.

7.4.1 Curve Fitting for Golden Device

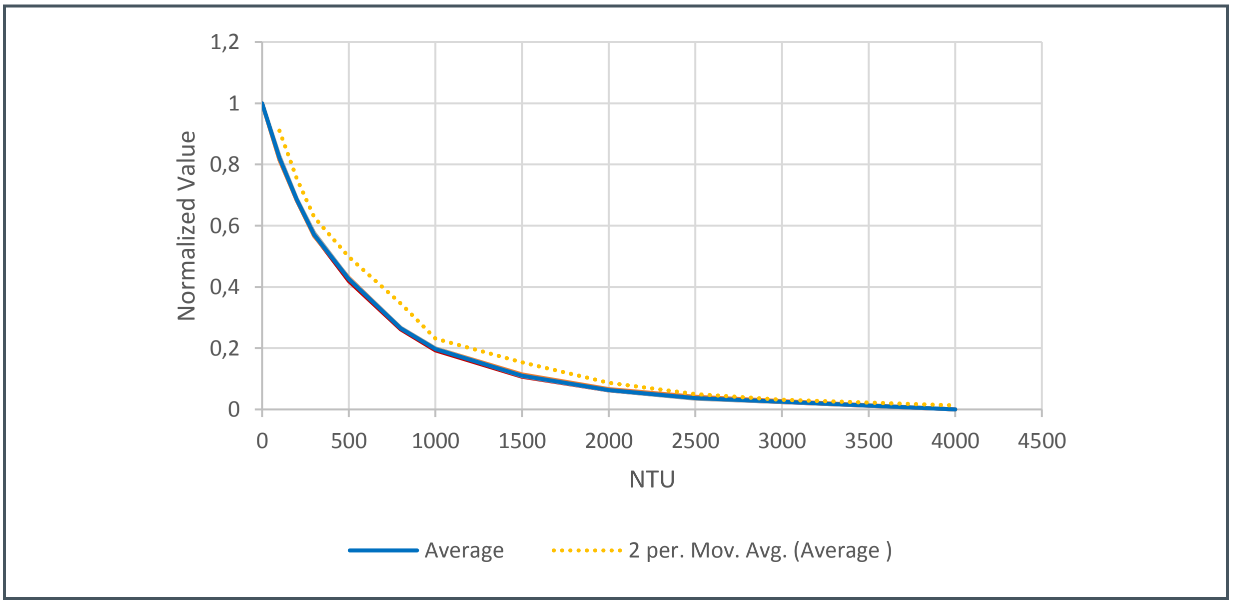
In the Golden device calibration, the balanced curves of a Golden Device (in the example, the averaged values from all sensors were used) is used to realize a correlation as formula which can be used in the sensor's firmware.

Figure 31:
Averaging of Sensor Results

NTU	Sensor_1	Sensor_2	Sensor_3
0	1	1	1
100	0.820754051	0.814899869	0.827807
200	0.682694341	0.683290886	0.687784
300	0.568701955	0.566622056	0.577646
500	0.418866096	0.429704854	0.428337
800	0.261199929	0.265462363	0.267002
1000	0.193010389	0.19971815	0.199251
1500	0.106287806	0.114790752	0.109112

NTU	Sensor_1	Sensor_2	Sensor_3
2000	0.061600447	0.066445503	0.062469
2500	0.035965727	0.040941153	0.034945
3000	0.023853813	0.026581705	0.024138
4000	0	0	0

Figure 32 :
Averaging of Sensor Results – Diagram (3 sensors shown here about each other with average and fitted curve)



In the process of curve fitting,

- the inputs are the counts on the x-axis.
- the results will be the matched NTU on the y-axe.

MATLAB was used as CAD tool for the process of curve fitting (like TRENDLINE function of MS EXCEL in diagrams, see the yellow broken line in Figure 32).

First tests show a higher accuracy in case of splitting the curve into three partitions with alternative a0…a3 parameters for the identical function of x-axe [0:500], [500-1500], [1500-4000]. These partitions result 3 alternative parameters sets for a0…a3 and must be realized in sensor's firmware as IF THEN commands.

The tests with the MatLab script result the following for the 3 sensor diagrams. As 0 NTU to 4000 NTU, there are so many points, a cubic equation is suitable for these. System will be unstable, if order of the equation will be increased.

Therefore, the equation for our example here is:

Equation 3:

$$y = a3 * x^3 + a2 * x^2 + a1 * x + a0$$

Below is the MatLab script where we used NTU and Normalization values from the example:

```
% Input data

NTU = [0 100 200 300 500 800 1000 1500 2000 2500 3000 4000];

normalized =[1  0.821153754 0.684589804 0.570990128 0.425636141 0.264554678
0.197326558 0.110063581 0.063504876 0.037284066 0.024857677 0];

%% 0 to 500NTU

x1 = normalized(1:5);

y1 = NTU(1:5);

Output(1,1:4) = polyfit(x1,y1,3);

%% 500 to 1500NTU

x2 = normalized(5:9);

y2 = NTU(5:9);

Output(2,1:4) = polyfit(x2,y2,3);

%% 1500 to 4000NTU

x3 = normalized(9:12);

y3 = NTU(9:12);

Output(3,1:4) = polyfit(x3,y3,3);
```

After execute the script, three equations have been produced as shown below.

Figure 33:
MatLab Produced Parameters

NTU	a3	a2	a1	a0
0 - 500	-1488.89	4097.258	-4318.98	1710.256
500 - 1500	-56896.6	53942.54	-18440.5	2963.21
1500 - 4000	8628314	-536391	-32227.1	4000

The following formulas can be used now to describe the correlation between x-counts and y-NTU.

```
IF x <= 0.425636 THEN y = -1488.89*x^3 + 4097.258*x^2 - 4318.98*x + 1710.256

IF x >= 0.425636 AND
   x <=0.110064 THEN y =-56896.6*x^3 + 53942.54*x^2 - 18440.5*x + 2963.21

IF x > 0.110064 THEN y = 8628314*x^3 - 536391*x^2 - 32227.1*x + 4000
```

In the sensor system and firmware (e.g. AS7261 firmware on flash), formula and parameters can be used directly.

Next step is the verification of the matching for all sensors by using the formulas of the 3 partitions.

Figure 34:
Golden Device

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.35911	0.35911
0.8182	100	103.8473	3.847333
0.681268	200	198.7379	1.26206
0.567806	300	306.3227	6.322699
0.421704	500	505.9002	5.900208
0.260834	800	836.0522	36.05216
0.194446	1000	998.7565	1.243497
0.10871	1500	1522.93	22.9303
0.062691	2000	2005.14	5.140247
0.036702	2500	2521.242	21.24199
0.024433	3000	3018.222	18.22151
0	4000	4000	0

Figure 35:
#Sensor_1

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.35911	0.35911
0.820754	100	102.3062	2.30617
0.682694	200	197.5854	2.414579
0.568702	300	305.3333	5.333262
0.418866	500	510.6223	10.62232
0.2612	800	835.1427	35.14269
0.19301	1000	1004.422	4.421568
0.106288	1500	1544.284	44.28363

Normalized Values	NTU	NTU Check	NTU Diff
0.0616	2000	2018.657	18.65721
0.035966	2500	2548.505	48.50461
0.023854	3000	3043.164	43.16442
0	4000	4000	0

Figure 36:
#Sensor_2

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.35911	0.35911
0.8149	100	105.8467	5.846743
0.683291	200	197.1047	2.895273
0.566622	300	307.6349	7.634897
0.429705	500	492.7785	7.221452
0.265462	800	824.6109	24.61085
0.199718	1000	978.6772	21.32281
0.114791	1500	1471.144	28.85574
0.066446	2000	1959.386	40.61417
0.040941	2500	2373.615	126.3847
0.026582	3000	2926.402	73.59757
0	4000	4000	0

Figure 37:
#Sensor_3

Normalized Values	NTU	NTU Check	NTU Diff
1	0	0.35911	0.35911
0.827807	100	98.07948	1.920522
0.687784	200	193.5078	6.49219
0.577646	300	295.5878	4.412185
0.428337	500	495.0012	4.998782
0.267002	800	820.8333	20.83332
0.199251	1000	980.4105	19.58945
0.109112	1500	1519.425	19.42547
0.062469	2000	2007.887	7.886994
0.034945	2500	2586.996	86.9955
0.024138	3000	3030.947	30.947
0	4000	4000	0

NTU Diff shows in all tables the difference between SHOULD and ACTUAL NTU by using the fitted curve to the Golden Device (here average of sensor 1…3).

7.4.2 Device Calibration

Device calibration considers a correlation function for each Black-White-Balanced curve. That means, more measurements over the fully dynamic range of turbidity must be done during the module end test. The results of these measurements are the calculated curve fitting with parameters from the test system, which must be stored on the sensor device. Such a process needs much more effort but guarantees highest accuracy. Below is the MatLab script.

```
% Input data

NTU = [0 100 200 300 500 800 1000 1500 2000 2500 3000 4000];

normalized_Sensor_1 =[1 0.820754051 0.682694341 0.568701955 0.418866096
0.261199929 0.193010389 0.106287806 0.061600447 0.035965727 0.023853813 0];

normalized_Sensor_2 =[1 0.814899869 0.683290886 0.566622056 0.429704854
0.265462363 0.19971815  0.114790752 0.066445503 0.040941153 0.026581705 0];

normalized_Sensor_3 =[1 0.827807341 0.687784185 0.577646372 0.428337472
0.267001743 0.199251134 0.109112184 0.062468678 0.034945319 0.024137513 0];

%% 0.2 to 500NTU

x1_Sensor_1 = normalized_Sensor_1(1:5);

y1_Sensor_1 = NTU(1:5);

Output_Sensor_1(1,1:4) = polyfit(x1_Sensor_1,y1_Sensor_1,3);

x1_Sensor_2 = normalized_Sensor_2(1:5);

y1_Sensor_2 = NTU(1:5);

Output_Sensor_2(1,1:4) = polyfit(x1_Sensor_2,y1_Sensor_2,3);

x1_Sensor_3 = normalized_Sensor_3(1:5);

y1_Sensor_3 = NTU(1:5);

Output_Sensor_3(1,1:4) = polyfit(x1_Sensor_3,y1_Sensor_3,3);

%% 500 to 1500NTU

x2_Sensor_1 = normalized_Sensor_1(5:9);

y2_Sensor_1 = NTU(5:9);
```

```
Output_Sensor_1(2,1:4) = polyfit(x2_Sensor_1,y2_Sensor_1,3);

x2_Sensor_2 = normalized_Sensor_2(5:9);

y2_Sensor_2 = NTU(5:9);

Output_Sensor_2(2,1:4) = polyfit(x2_Sensor_2,y2_Sensor_2,3);

x2_Sensor_3 = normalized_Sensor_3(5:9);

y2_Sensor_3 = NTU(5:9);

Output_Sensor_3(2,1:4) = polyfit(x2_Sensor_3,y2_Sensor_3,3);

%% 1500 to 4000NTU

x3_Sensor_1 = normalized_Sensor_1(9:12);

y3_Sensor_1 = NTU(9:12);

Output_Sensor_1(3,1:4) = polyfit(x3_Sensor_1,y3_Sensor_1,3);

x3_Sensor_2 = normalized_Sensor_2(9:12);

y3_Sensor_2 = NTU(9:12);

Output_Sensor_2(3,1:4) = polyfit(x3_Sensor_2,y3_Sensor_2,3);

x3_Sensor_3 = normalized_Sensor_3(9:12);

y3_Sensor_3 = NTU(9:12);

Output_Sensor_3(3,1:4) = polyfit(x3_Sensor_3,y3_Sensor_3,3);
```

After execute the script, parameters for three equations have been produced as shown below.

Figure 38:
Parameter for #Sensor_1

NTU	a3	a2	a1	a0
0 - 500	-1323.24	3687.404	-3982.38	1617.91
500 - 1500	-63310.7	58105.49	-19119	2965.753
1500 - 4000	9076520	-525149	-34559.8	4000

Figure 39:
Verification for #Sensor_1

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.30293	0.302933
0.820754	100	101.7227	1.722724
0.682694	200	196.7222	3.277821
0.568702	300	302.3258	2.325775
0.418866	500	499.5323	0.467745
0.2612	800	807.9283	7.928325
0.19301	1000	984.9833	15.01666
0.106288	1500	1514.045	14.04499
0.0616	2000	2000	2.59E-07
0.035966	2500	2500	9.04E-06
0.023854	3000	3000	1.45E-05
0	4000	4000	0

Figure 40:
Parameter for #Sensor_2

NTU	a3	a2	a1	a0
0 - 500	-1754.51	4749.629	-4833.98	1838.272
500 - 1500	-51711.7	50813.63	-18081.5	2989.899
1500 - 4000	4715348	-250013	-34305.9	4000

Figure 41:
Verification for #Sensor_2

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.58963	0.589628
0.820754	100	103.6669	3.666918
0.682694	200	193.0724	6.927648
0.568702	300	304.9704	4.970436
0.418866	500	498.8799	1.120078
0.2612	800	803.4024	3.402401
0.19301	1000	993.5627	6.437257
0.106288	1500	1505.655	5.654902
0.0616	2000	2000	2.1E-06
0.035966	2500	2500	3.71E-07
0.023854	3000	3000	9.58E-06
0	4000	4000	0

Figure 42:
Parameter for #Sensor_3

NTU	a3	a2	a1	a0
0 - 500	-1429.62	3945.723	-4206.06	1689.792
500 - 1500	-55820.2	52901.74	-18096.1	2931.335
1500 - 4000	13947966	-962403	-26325.6	4000

Figure 43:
Verification for #Sensor_3

Normalized Values	NTU	NTU Check	NTU Diff
1	0	-0.1625	0.162504
0.820754	100	100.8755	0.875452
0.682694	200	198.3115	1.688538
0.568702	300	301.214	1.214038
0.418866	500	499.7616	0.238447
0.2612	800	808.4852	8.485187
0.19301	1000	984.3481	15.65189
0.106288	1500	1514.134	14.13409
0.0616	2000	2000	7.52E-06
0.035966	2500	2500	1.07E-05
0.023854	3000	3000	2.05E-05
0	4000	4000	0

An insertion of these three cubic equations into the firmware which is vary device to device, will get the actual NTU value from three sensors according to their normalized values which are calculated from Basic count.

7.5 Conclusion

- ☑ To measure the turbidity, it is important to calculate Basic counts.
- ☑ To get uniformity for all sensors, Black-White-Balance is necessary.
- ☑ Golden device calibration can be used for one time calibration. In this case, average reading or one standard device has to be selected as a golden device. All other sensors result will be dependent on the Golden device.
- ☑ To get more accuracy, Device calibration is important. In this case, we need to calibrate every device separately

In case of the results of the sensor system are not in the required range then:

- Check the selection of the LED – sensor combination – via simulation, optimize sensor / luminary overlapping to get a high response for each target (here means the concentrations of substances in any liquid) – compare measurements and simulations to find reason for non-accuracy
- Optimize the optical path (limit the output of the LED – FOV FieldOfView) by an optical hole in front of the LED (affects the ration between transmitted and reflected light)
- In case of lower counts (results of ADC) increase gain/TINT or LED driver currents, use optimized gain and make sure max. FSR is set (16-bit counter is reserved and max. used)
- Check for drifts those can be corrected (non-linearity's, temperatures)
- Check for other disturbances by tests where real conditions can be switched on step-by-step (first measurements in dark without liquid by using known filters to see a clear filter response, then "open" the system step-step by changing the conditions)

The tips above are based on many feasibilities and are not a complete action list. Therefore, do not hesitate to contact us to optimize your application tests.

Headquarters
ams 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

Copyright ams AG. Trademarks Registered. All rights reserved. The material herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner.

ams 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 AG rendering of technical or other services.