SEN6x – Temperature Acceleration and Compensation Instructions

Deriving the Optimal Parameter Set for Offset Compensation and Acceleration of Temperature & Relative Humidity

This application note provides instructions on how to compensate the temperature reading of the SEN6x sensor family for design-in effects using its integrated STAR-Engine. The goal is to measure the ambient temperature despite the additional thermal mass and waste heat from electronic components that alter the conditions experienced by the temperature sensor. The SEN6x is fully compensated for the effects of the module itself, but will require unique tuning to the device it is built into.
In the following, we will go over the different effects that can be seen on the temperature signal and what has to be done to counter them. A detailed recipe leads through the necessary steps to determine the parameters for your device using the SEN6x sensor family.
The relative humidity output will be adjusted automatically based on the compensated temperature and must not be compensated separately.

Contents

1 Intro

Here we provide you with the necessary background knowledge on why temperature compensation is needed. We explain the different design-in effects that can be seen on the temperature signal. We also go over what the STAR-Engine is able to provide to compensate for these effects. If you are only interested in the necessary steps to determine the parameters, you can directly jump to **Recipe to Determine the Parameters for the STAR-Engine).**

Note: While the internal temperature compensation algorithm (STAR-Engine) of the SEN6x family can be tuned to deal with additional thermal effects due to the design-in, an externally induced influence is always subject to uncertainty. It will not be the same in every device due to variation in the electronics and the path of the heat flux (manufacturing tolerances). This variation is relative to the magnitude of the over-temperature, the larger the self-heating, the larger the variance from device to device. Therefore, in a first approach, the physical sensor integration must be optimized according to the *SEN6x – Mechanical Design and Assembly Guidelines* [1] before continuing with this guide. An over-temperature of less than 5 °C is recommended to proceed with temperature compensation.

1.1 Acceleration – Effect of Thermal Mass

The thermal mass (and the thermal resistance) of the device introduces a low-pass filter onto the raw temperature signal (T_{RAW}) from the sensor inside of the device. Thermal mass refers to the ability of a material to absorb and store heat energy. The device will therefore not follow changes in temperature of the environment (T_{AMB}) immediately, but it will take time until it is warmed-up or cooled-down to the new temperature. The temperature that the sensor is locally measuring, is therefore changing more slowly than the temperature of the environment. This means, simplified, that the bigger the thermal mass, the slower the temperature changes will be that one sees, as shown in **Figure 1**.

The acceleration function of the STAR-Engine inside the SEN6x sensor family compensates for the delay induced. The acceleration algorithm predicts the ambient conditions based on the delayed measurement signal. Any prediction is subject to some uncertainty and should not be chosen too aggressively. Otherwise, small fluctuations will be interpreted as an ambient condition change, leading to an oscillating output, over- or undershooting of the signal, or increased noise.

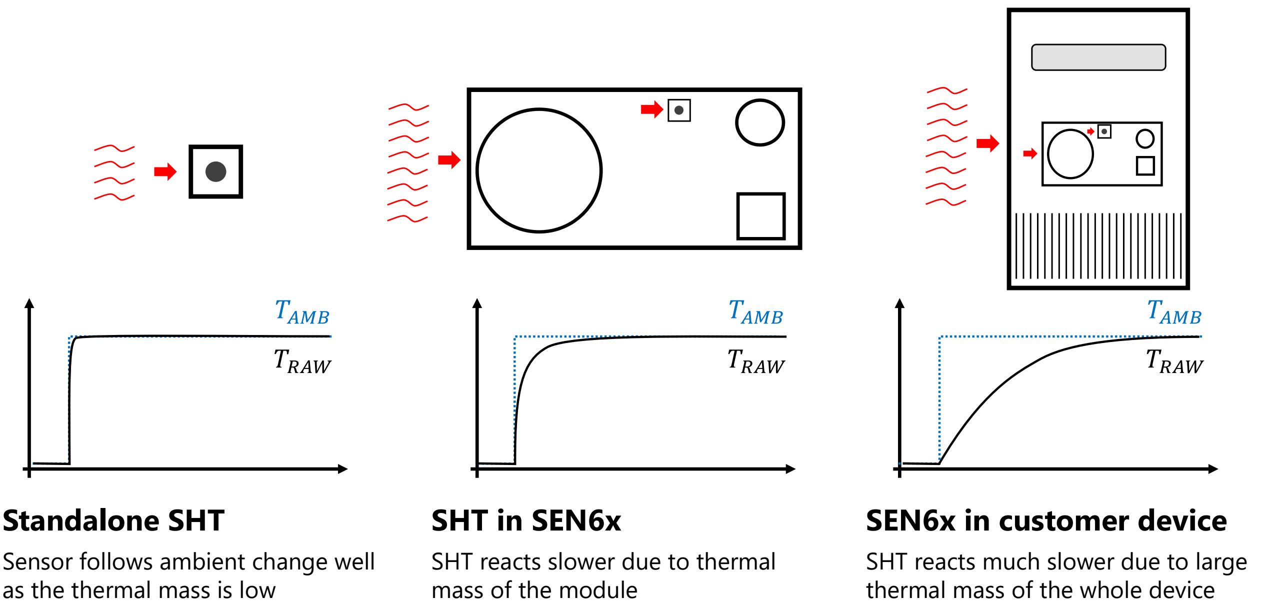

Figure 1. Effect of thermal mass on temperature signal.

1.2 Offset – Effect of Waste Heat

Every device consumes electric power and dissipates it as heat. Due to this heating effect, the sensor measures an elevated temperature (T_{RAW}) within the device compared to the ambient environment (T_{AMB}), as seen in **Figure 2**.

The offset compensation function of the STAR-Engine estimates the current offset from the local conditions to the environment and subtracts this difference from the measured signal, so the compensated output is as close as possible to the ambient conditions.

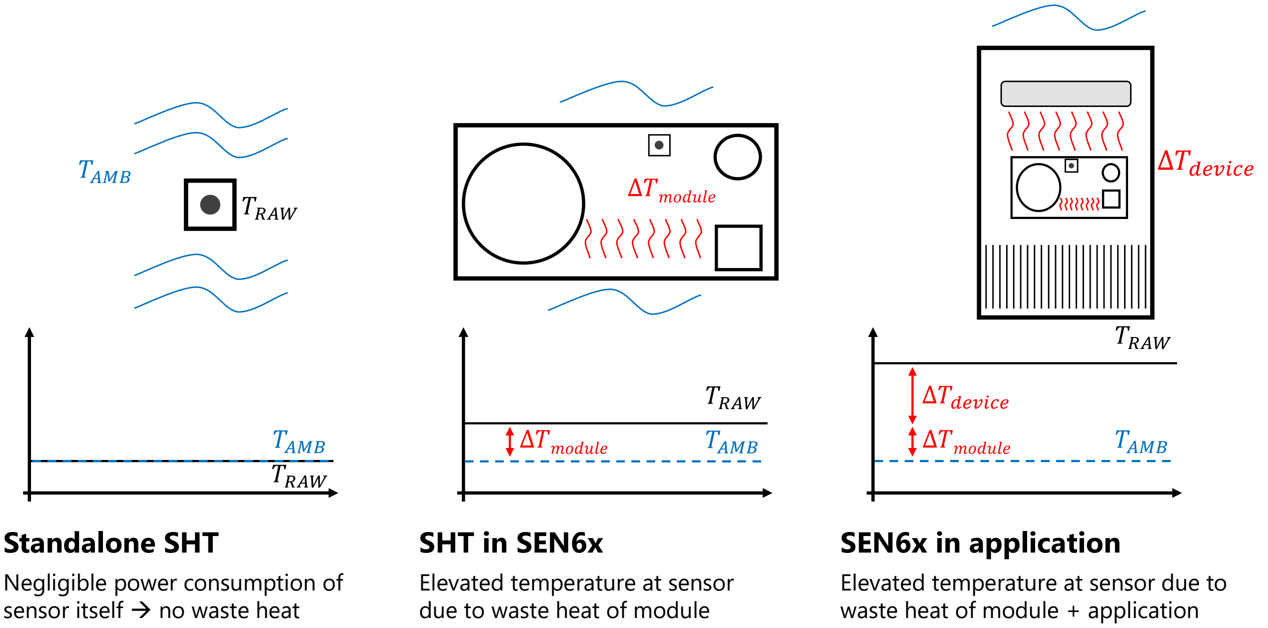

Figure 2. Effect of waste heat on temperature signal.

1.3 Time Constant – Effect of Thermal Resistance

A state change of the customer device alters the heat generation or dissipation of the device. This only gradually changes the offset at the sensor location due to the thermal resistance of the materials (and the thermal mass). The temperature signal (T_{RAW}) will therefore not change immediately, as seen in **Figure 3**.

The time constant parameter (τ_{63}) in the STAR-Engine of the SEN6x sensor family enables compensation for this delayed response.

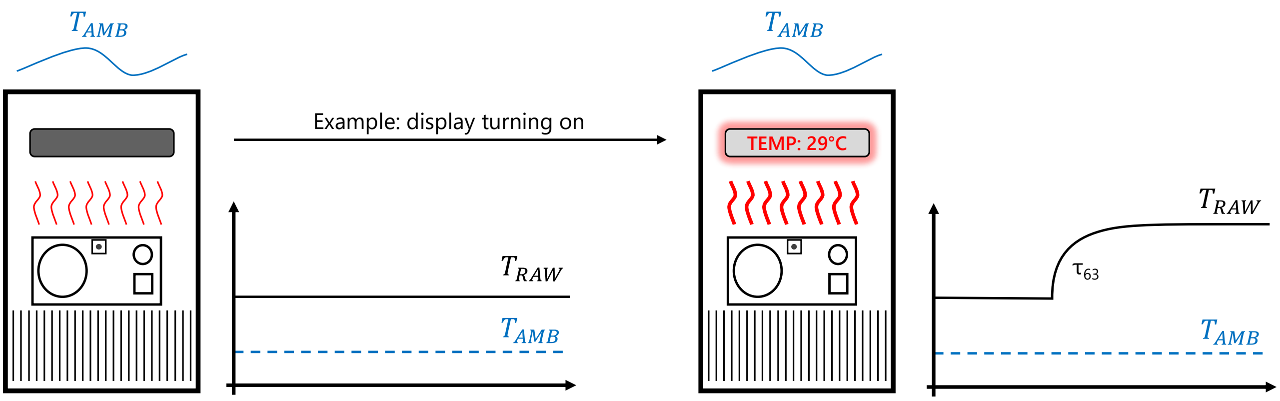

Figure 3. Effect of thermal resistance on the temperature signal.

1.4 Slope – Effect of Ambient Temperature on Waste Heat Generation

The offset induced by the heat generation or dissipation of a device can have an ambient temperature dependence, as seen in **Figure 4**. This means that the offset at one temperature can be different from the offset induced at another temperature.

The STAR-Engine in the SEN6x sensor family therefore offers a slope parameter that allows for a linear compensation of the ambient temperature dependence.
Examples:

- A fan that has more friction at lower temperatures due to lubricant viscosity
- Semiconductor components with a temperature-dependent efficiency

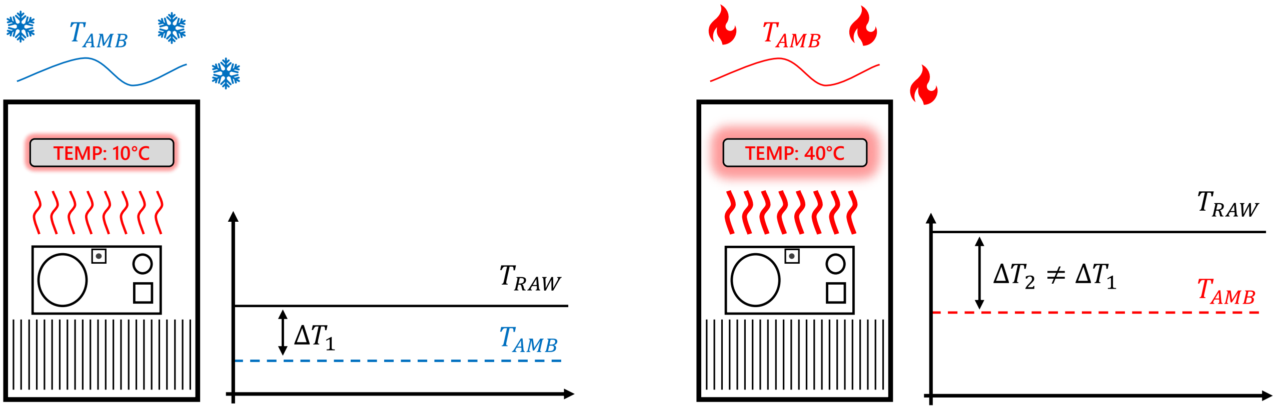

Figure 4. Effect of the ambient temperature on the waste heat generation.

1.5 Relative Humidity Compensation

Relative humidity does not need further compensation, as it can be calculated based on the raw relative humidity, the raw temperature signal, and the compensated temperature signal, with the formula provided in the *Humidity at a glance* [2] application note. This is done automatically by the STAR-Engine in the SEN6x sensor family and does not need further configuration. The relative humidity signal provided by the SEN6x sensor family is therefore already fully compensated once the temperature is compensated.
The concept behind this is that the absolute humidity is not changing, and relative humidity is a function of absolute humidity and temperature, as shown in **Figure 5**.

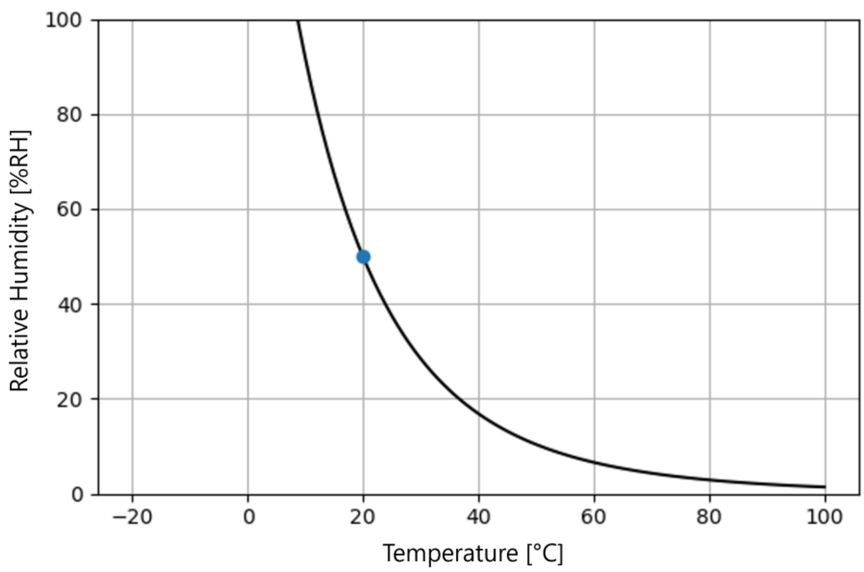

Figure 5. Change of relative humidity over temperature for a fixed absolute humidity.

2 Recipe to Determine the Parameters for the STAR-Engine

Below, we will detail how the parameters for the STAR-Engine can be determined for your device. These parameters can then be sent to the SEN6x sensor, as detailed in **Applying Parameters** to the STAR-Engine, and the STAR-Engine will take care of the compensation and acceleration of the temperature and humidity signal in your application.

Note: Time constants for offsets are acceleration dependent, so always determine and fix the acceleration parameters first!

The **Setup** section details the equipment, hardware, and software required for the experiments.
In the **Parameter identification (basic)** section, we guide you through the process of compensating the self-heating of a single operation mode, considering the startup behavior, and using a preset for the acceleration. With this the major effects of the integration are compensated for.
In the **Parameter identification (advanced)** section, we customize the acceleration settings according to a step response of your system, optimizing the response time for your application. The offset can be adjusted to achieve a good performance over a large temperature range, and the startup behavior is considered. Furthermore, there is the option to compensate for multiple operation modes with different heat dissipation.

2.1 Setup

2.1.1 Environment

For the experiments, we recommend using two climate chambers or rooms, with no forced convection, and two different (stable) temperatures (except for **Parameter identification (basic)**, where only one chamber/room is required). We recommend choosing a temperature difference between the two chambers of about 10 °C. It is fundamental that the temperature is constant over time. Fluctuations and drifts will reduce data quality and decrease the accuracy of the compensation. Ensure the environment does not have active air conditioning.

Note: Cellar rooms often have stable temperature conditions.

2.1.2 Software

It is recommended to use the latest ControlCenter & DataViewer [3] software (version >= 1.47.0), available on Sensirion's website, to record the data from the sensors. This allows you to tune the parameters and simulate the engine output directly in the software, without the need to repeat the experiments with every new parameter set to see the effect.
Disable sleep settings on the computer running the ControlCenter software during the experiments, to ensure that the data is recorded without interruptions.
In ControlCenter, make sure to click on the gear icon next to SEN6x sensor module name and select the "*Enable raw values plots*" checkbox, as seen in **Figure 6**, to see the raw temperature signal from the module additionally to the compensated signal.

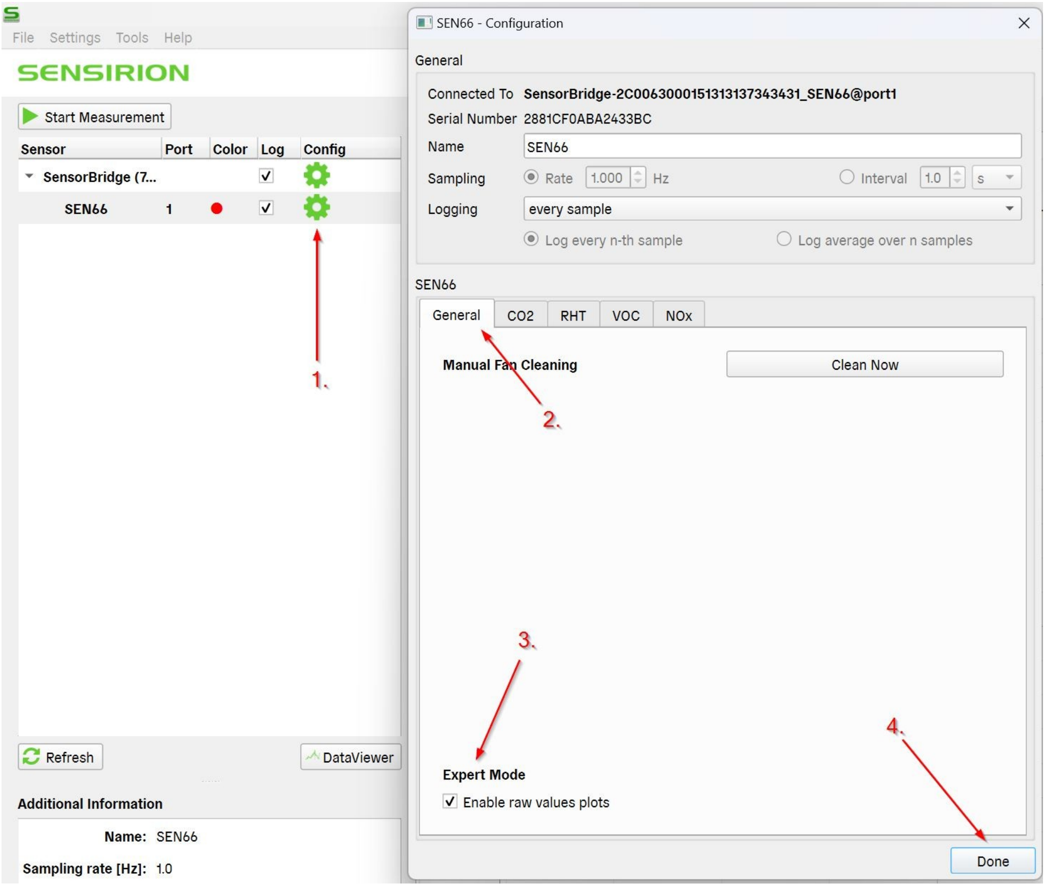

Figure 6. *Enable raw values plots* for SEN6x

2.1.3 Device Preparation

Make sure that the SEN6x is installed in the final device, and the device is oriented as in the final application, to have the same conditions as in the field. Also, make sure that the device is operating as in the final application. Any change to the operation mode, such as turning on lights, changing the brightness of screens, or turning on or off any components within the device, will alter the power consumption of your device and the waste heat generated. It is therefore vital that the test conditions are equivalent to the operation mode in the field. Especially for smaller devices, it is recommended to have them sitting on a surface with similar thermal properties as in the final application. For example, an indoor air quality monitor is recommended to be placed on a wooden plate, simulating a desk or sideboard.

To read the data from the SEN6x sensor inside the device with ControlCenter, this sensor must be directly connected via a SEK-SensorBridge to your measurement computer. To guarantee normal operation of the device and its electronics, a second SEN6x must be connected to the device electronics by using an additional long cable, and is ideally placed outside of the climate chamber (if one is used), but at least 10cm away from the device, with the outlet facing away to avoid any thermal influence. Not having this second SEN6x sensor might alter the operating conditions and will impact on the accuracy of the compensated signal. **Figure 7** below illustrates the setup and how the device is prepared for the measurement.

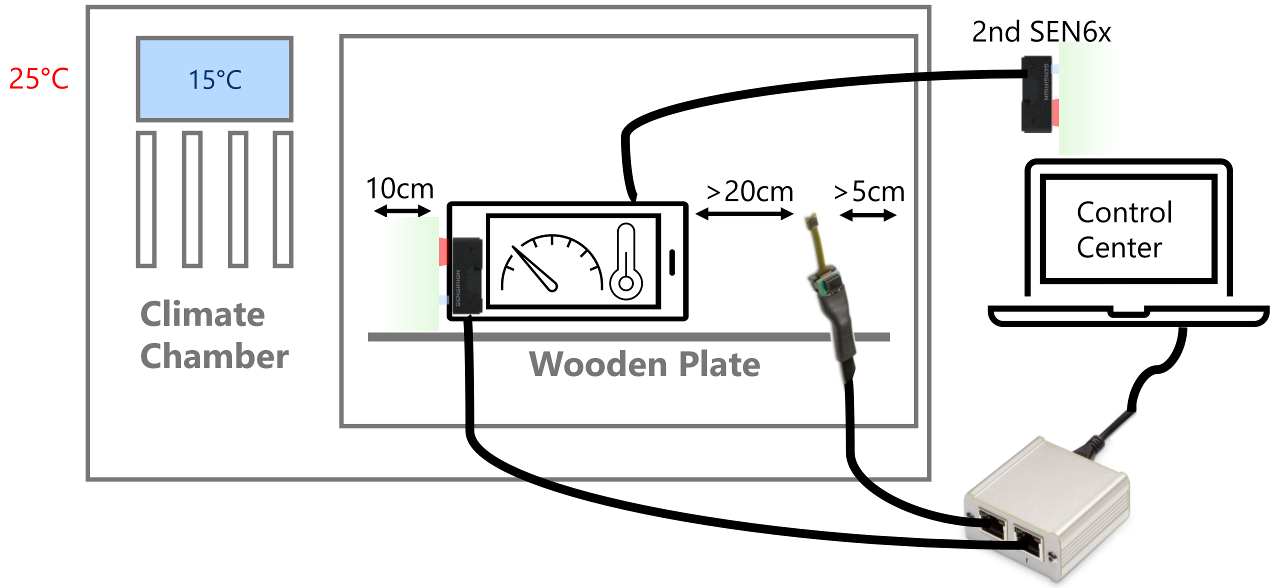

Figure 7. Climate chamber with setup: SHT4x reference & device with SEN6x connected to ControlCenter, 2nd SEN6x, connected to device electronics to simulate normal operating conditions. Electronics that produce heat and will disturb the experiment are placed outside of the climate chamber if possible.

2.1.4 Temperature Reference

We recommend using an evaluation kit of Sensirion's SHT4x temperature and humidity sensors, such as SEK-SHT45 as a temperature reference. The evaluation kit can be connected to ControlCenter through Sensirion's SEK-SensorBridge.
The reference sensors should be placed in a way that they are not affected by the experimental setup: They should not touch any surfaces, should not be placed above the device (due to rising heat), nor in the outlet of the sensor (where warm air may flow). Additionally, maintain at least 5 cm distance from any walls, and position them at least 20 cm away from heat sources (Sensor-Bridge itself is also a heat source) as seen in **Figure 7**. This ensures that reference sensors measure the environmental conditions and are not influenced by other factors. Using multiple references at different locations improves accuracy and makes it easier to detect disturbances of the setup. Any compensation can only be as accurate as the reference measurement.

2.2 Parameter identification (basic)

In the following, we will guide you through the process of compensating the self-heating of a single operation mode, considering the startup behavior and using a preset for the acceleration. With this, the major effects of integration are compensated with a minimal amount of time and equipment effort.

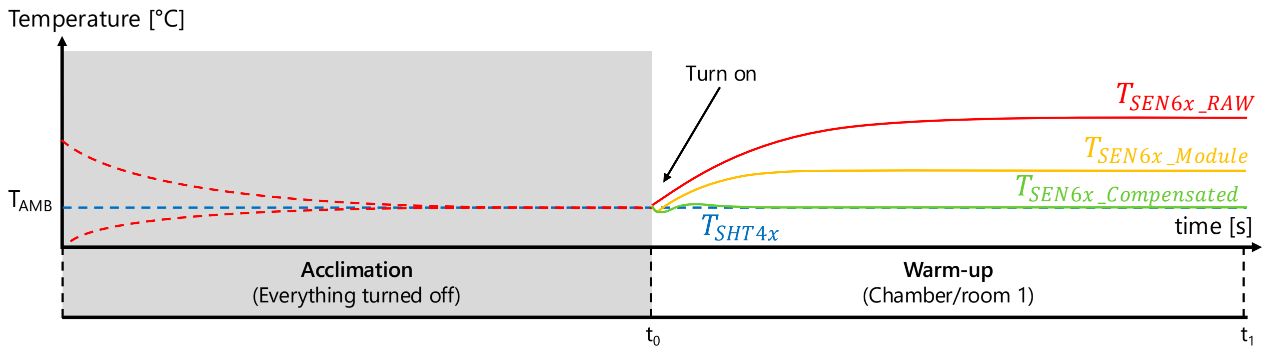

Figure 8. Overview of experiment for basic parameter identification.

Acclimation: In **Figure 8,** we see on the left the acclimation phase (grey background): The device (dashed red line) gradually approaches the stable ambient temperature T_{SHT4x} (dashed blue line) until thermal equilibrium is reached.

Warm-up: At t_0, we turn the device on and start the data recording to monitor the warm-up (white background). The SHT4x reference sensor measures the ambient temperature T_{SHT4x} (dashed blue line). The SEN6x provides the raw temperature T_{SEN6x_RAW} (red line), which is the uncompensated output, showing the raw temperature at the internal location of the sensor and the module temperature T_{SEN6x_Module} (yellow line) that is compensated for the modules' internal heating.

Finally, the compensated module temperature $T_{SEN6x\_Compensated}$ (green line) shows what can be achieved after plugging in the parameters that will be determined in the following.

2.2.1 Experiment

1. Prepare the setup as described in **Section 2.1**.
2. Turn off all electronics, including the SEN6x and the Sensirion SEK-SensorBridge, to prevent self-heating. Allow the system to acclimate to ambient temperature for at least 3 hours. Devices with high thermal mass may require more time.
3. Start the experiment quickly to capture accurate startup behavior:
    a. Power on the device in its normal operating mode.
    b. Plug in the Sensirion SEK-SensorBridge
    c. Start the Sensirion ControlCenter
    d. Tick the box to record raw values, as shown in **Figure 6**
    e. Immediately start the measurement (any delay will introduce a startup error)
4. Check in ControlCenter if the *Temperature Raw Signal* of the SEN6x and the *Temperature* of the SHT4x reference do not deviate more than 0.5 °C from each other right after the start of the measurement (at t_0). If the deviation is larger, the acclimation time was too short. In this case, stop the recording, turn off the device, unplug the SEK-SensorBridge, and wait another 2h, then repeat step 3 & 4.
5. The device will now heat up due to the internal electronics. Here, we wait until the equilibrium (stable *Temperature Raw Signal* of the SEN6x) is reached (may take several hours!).

2.2.2 Parameter Extraction

Select acceleration parameters based on device type from **Table 1.** Larger thermal mass allows for more aggressive acceleration. Over-aggressive settings may cause overshoot or noise.

Acceleration Level	T1	T2	K	P
Light (IAQM)	100	300	20	20
Middle	100	600	50	20
Strong (Air Purifier)	250	550	150	20

Table 1. Acceleration Parameter Preset.

Next, we open the two .edf files (one from SEN6x and one from the SHT4x reference) with the recordings from our experiment in DataViewer. This can be done either by navigating to *File>Open Data Log Folder* in ControlCenter and opening the selected files, or by clicking on the *DataViewer* button for the current measurement as seen in **Figure 9**.

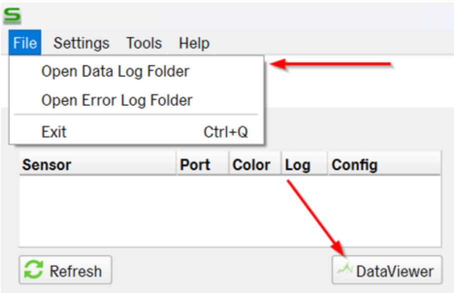

Figure 9. Opening EDF files either through the log folder or directly through the DataViewer button for the current measurement

As shown in **Figure 10**, open the Tab *Algorithms* on the left, and add a new temperature algorithm. Enter the four acceleration parameters T_1, T_2, K, P and press *Save*. The software post-processes the data with the acceleration engine. This new curve can now be used to determine the offset (C_{offset}) and warm-up time constant (τ_{63}).

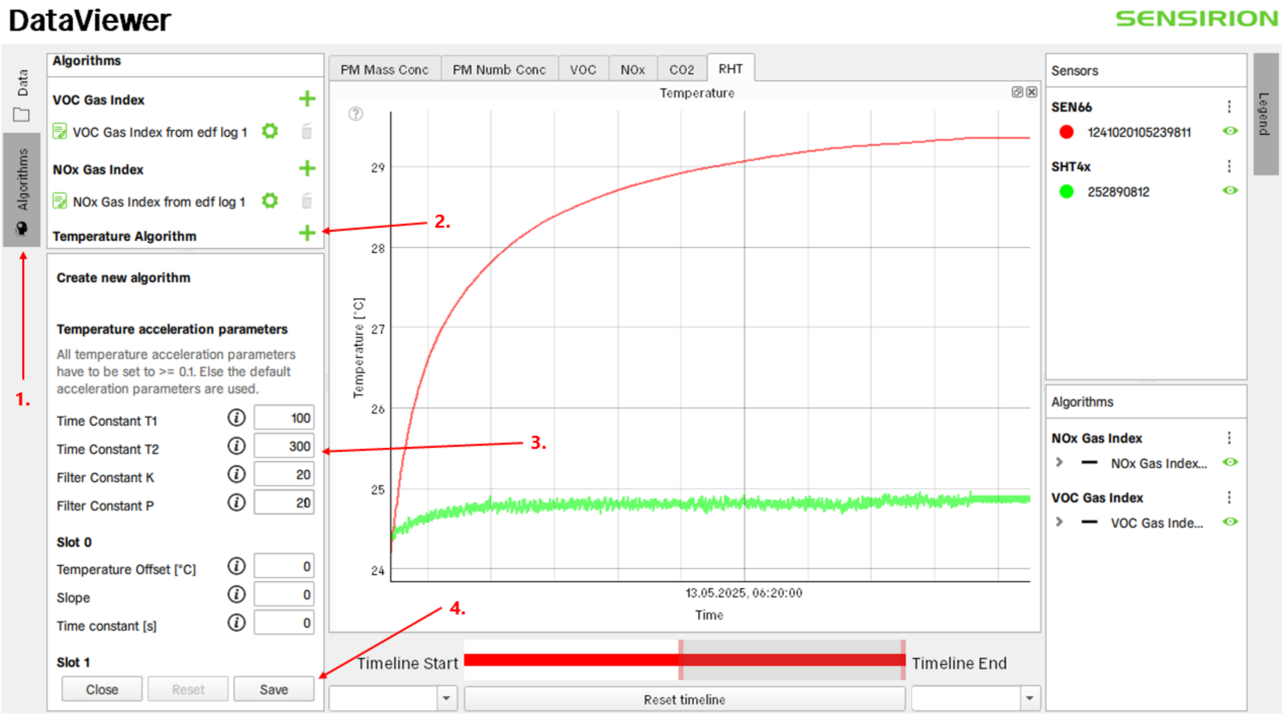

Figure 10. Applying an acceleration preset in DataViewer.

Note: Use the **accelerated signal** (post-processed) to derive the offset and warm-up time constant as they change with the acceleration and therefore must be determined after applying the parameters.

$\boldsymbol{C_{offset}}$**:** Usually, the device will have an over-temperature compared to ambient due to self-heating. In that case, C_{offset} should be negative to correct the sensor value down to the ambient temperature.
To get the steady state offset correction factor from your data, take the last few minutes of the experiment and take the average over time. The offset correction factor C_{offset} is calculated as follows:

$C_{offset} = T_{SHT4x} - T_{SEN6x\_Module}$

$\boldsymbol{\tau_{63}}$**:** Next, we correct the device performance during the warm-up period. We calculate the time constant with which the offset correction factor is applied.

First, we calculate the temperature level that corresponds to 63% self-heating. Use the steady state values from the end of the experiment:

$$T_{63} = T_{SHT4x} + (T_{SEN6x\_Module} - T_{SHT4x}) * 0.63$$

In the next step, we note how much time it takes from the start-up of the device to the time at which the T_{SEN6x_Module} curve (yellow curve) crosses the 63% level T_{63} as shown in **Figure 11.** Visualization of τ_{63} determination.. The extracted time duration τ_{63} (in seconds) is the desired time constant of the warm-up, and this can be directly put as parameter into the compensation engine.

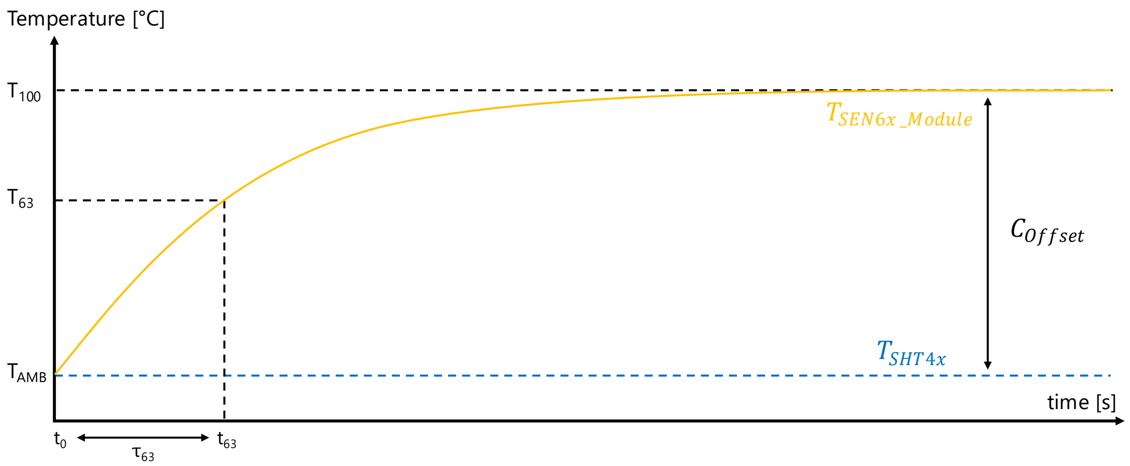

Figure 11. Visualization of τ_{63} determination.

An alternative method to derive the time constant τ_{63} is to apply the least-squares method to fit an exponential function to the data set and extract the time constant from there.

You can now enter the coefficient C_{offset} and the time constant τ_{63} in DataViewer Slot 0 (Slope remains 0), and then click *Save*. The post-processed signal gets updated again, and the SEN6x temperature should now align with the reference. If there remains a misalignment at the steady state (end of your experiment) with the reference, you may adjust the C_{offset} slightly to get a perfect alignment. With slight adjustment of τ_{63}, you can fine-tune the startup behavior.

Now you have the offset compensation factor C_{offset}, the time constant for the warmup τ_{63}, and the four acceleration parameters suited for your device size. Please proceed with **Section 2.4. (Parameter Verification)** to check for the stability of the parameter set in your application. **Section 3 (Applying Parameters to the STAR-Engine)** helps you with the application in ControlCenter and with applying the settings with I^{2}C in your firmware.

2.3 Parameter identification (advanced)

In the following, we will guide you through the process of compensating the self-heating, considering the startup behavior, and using a customized parameter set for the best possible signal acceleration. With this, the effects of the integration are compensated, and you make use of the full potential of the STAR-Engine. If you are considering compensating for more than one operation mode, please additionally have a look at **Section 2.3.3**.

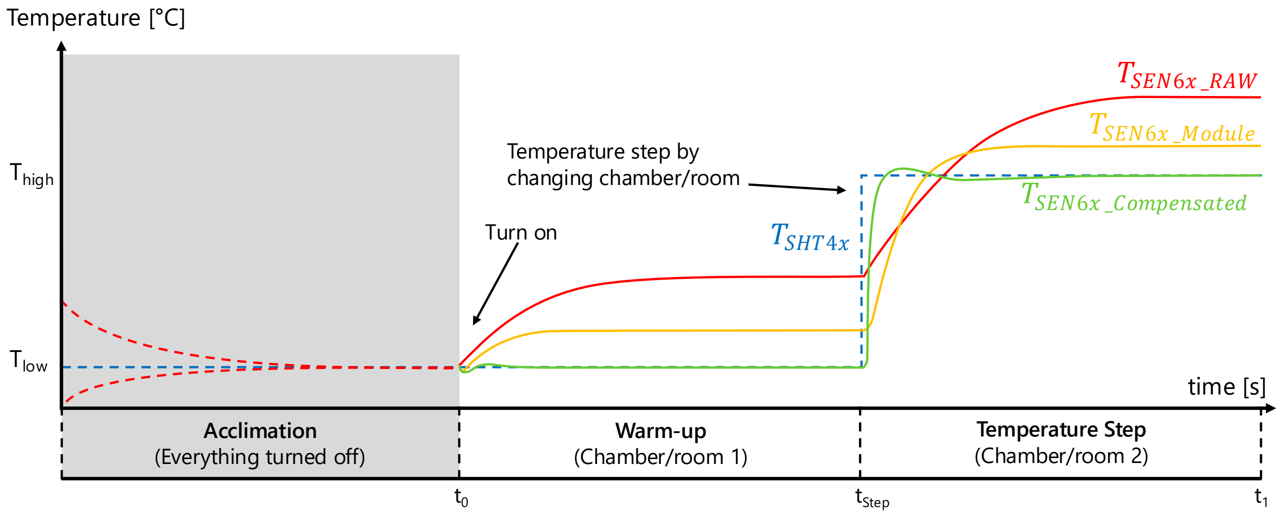

Figure 12. Overview of experiment for advanced parameter identification.

Acclimation: We see on the left the acclimation phase (grey background): The temperature of the device (dashed red line) approaching the constant ambient temperature T_{SHT4x} (dashed blue line), until equilibrium is reached.

Warm-up: At t_0, we turn the device on and start the data recording to monitor the warm-up (white background). The SHT4x reference sensor measures the ambient temperature T_{SHT4x} (dashed blue line). The SEN6x provides the raw temperature T_{SEN6x_RAW} (red line), which is the uncompensated output, showing the raw temperature at the internal location of the sensor and the module temperature T_{SEN6x_Module} (yellow line) that is compensated for the modules' internal heating.

Finally, the compensated module temperature $T_{SEN6x\_Compensated}$ (green line) shows what can be achieved after plugging in the parameters that will be determined in the following.

Temperature-Step: The data from the step response of the system allows us to derive customized acceleration parameters. Additionally, we get the offset at a second temperature level, which allows us to calculate the slope parameter, which is then used for a linear compensation of the ambient temperature dependence.

2.3.1 Experiment

1. Prepare the setup as described in **Section 2.1**.
2. Turn off all electronics, including the SEN6x and the Sensirion SEK-SensorBridge, to prevent self-heating. Allow the system to acclimate to ambient temperature for at least 3 hours. Devices with high thermal mass may require more time.
3. Start the experiment quickly to capture accurate startup behavior:
    a. Power on the device in its normal operating mode.
    b. Plug in the Sensirion SEK-SensorBridge
    c. Start the Sensirion ControlCenter
    d. Tick the box to record raw values, as shown in **Figure 6**
    e. Immediately start the measurement (any delay will introduce a startup error)

4. Check in ControlCenter if the Temperature Raw Signal of the SEN6x and the Temperature of the SHT4x reference do not deviate more than 0.5 °C from each other right after the start of the measurement (at t0). If the deviation is larger, the acclimation time was too short. In this case, stop the recording, turn off the device, unplug the SEK-SensorBridge, and wait another 2h, then repeat step 3 & 4.
5. The device will now heat up due to the internal electronics. Here, we wait until the equilibrium (stable *Temperature Raw Signal* of the SEN6x) is reached (may take several hours!).
6. Transfer the whole setup into the second chamber with a temperature approximately 10 °C warmer than the first one. Here, it is important that the data acquisition is not interrupted, and the whole change happens quickly.
7. Leave the setup to acclimate to the new temperature until the *Temperature Raw Signal* of the SEN6x does not change anymore (may take several hours again).

2.3.2 Parameter Extraction

Select acceleration parameters based on device type from **Table 1.** Larger thermal mass allows for more aggressive acceleration. Over-aggressive settings may cause overshoots or noise.

Next, we open the two .edf files (one from SEN6x and one from the SHT4x reference) with the recordings from our experiment in DataViewer. This can be done either by navigating to *File>Open Data Log Folder* in ControlCenter and opening the selected files, or by clicking on the *DataViewer* button for the current measurement as seen in **Figure 9**.

As shown in **Figure 10**, open the Tab *Algorithms* on the left, and add a new temperature algorithm. Enter the four acceleration parameters T_1, T_2, K, P and press *Save*. The software post-processes the data with the acceleration engine. This new curve can now be used to determine the offset (C_{offset}), warm-up time constant (τ_{63}) and the slope parameter ($Slope$).

Note: Use the accelerated signal (post-processed) to derive the warm-up time constant, the offsets and the slope parameter as they change with the acceleration and therefore must be determined after applying the parameters.

$\boldsymbol{C_{Offset\_0°C}}$: To get the steady state offset correction factor from your data, take the last few minutes of the warm-up phase (see **Figure 12**), and take the average over time. The offset correction factor is calculated as follows:

$$Offset_{low} = T_{SHT4x\_low} - T_{SEN6x\_Module\_low}$$

Usually, the device will have an over-temperature compared to ambient due to self-heating. In that case, $Offset_{low}$ should be negative, to correct the sensor value down to the ambient temperature.

Afterwards, we look at the data after the temperature step, the method used is the same as in the previous step:

$$Offset_{high} = T_{SHT4x\_high} - T_{SEN6x\_Module\_high}$$

We now have two offsets at different temperatures. From this, we can calculate the slope parameter ($Slope$), and the offset at 0 °C ($C_{Offset\_0°C}$).

$$Slope = \frac{Offset_{high} - Offset_{low}}{T_{SEN6x\_Module\_high} - T_{SEN6x\_Module\_low}}$$

$$C_{Offset\_0°C} = Offset_{low} - T_{SEN6x\_Module\_low} \cdot Slope$$

If any slope factor other than 0 is used, the offset correction factor at 0 °C ($C_{Offset\_0°C}$) must be calculated and used. With the $Slope$ and $C_{Offset\_0°C}$, the SEN6x engine calculates an individual offset for each ambient temperature.

$\boldsymbol{\tau_{63}}$**:** We correct the device performance during the warm-up period. We calculate the time constant with which the offset correction factor is applied.

First, we calculate the temperature level that corresponds to 63% self-heating. Use the steady state values from the end of the experiment:

$$T_{63} = T_{SHT4x\_low} + (T_{SEN6x\_Module\_low} - T_{SHT4x\_low}) * 0.63$$

In the next step, we note how much time it takes from the start-up of the device to the time at which the T_{SEN6x_Module} curve (yellow curve) crosses the 63% level T_{63} as shown in **Figure 11.** Visualization of τ_{63} determination.The extracted time duration τ_{63} (in seconds) is the desired time constant of the warm-up, and this can be directly put as parameter into the compensation engine.

An alternative method to derive the time constant τ_{63} is to apply the least-squares method to fit an exponential function to the data set and extract the time constant from there.

You can enter the coefficient $C_{Offset\_0°C}$, $Slope$, and the time constant τ_{63} in DataViewer Slot 0, and then click *Save*. The post-processed signal gets updated, and the SEN6x temperature should align with the reference. If there remains a misalignment at the steady state with the reference, you may adjust the $C_{offset\_0°C}$ or $Slope$ slightly to get a perfect alignment. With adjustment of τ_{63} you can fine-tune the startup behavior.

Now you have the offset compensation factor $C_{Offset\_0°C}$, $Slope$, the time constant for the warm-up τ_{63}, and the four acceleration parameters T_1, T_2, K, P suited for your device size. In the next section, we describe the process of compensating for multiple operation modes. If you have only one operation mode, directly proceed with **Section 2.4 (Parameter Verification)** to check for the stability of the parameter set in your application. **Section 3 (Applying Parameters to the STAR-Engine)** helps you with the application in ControlCenter and with applying the settings with I^{2}C in your firmware.

2.3.3 Multi-Mode Compensation

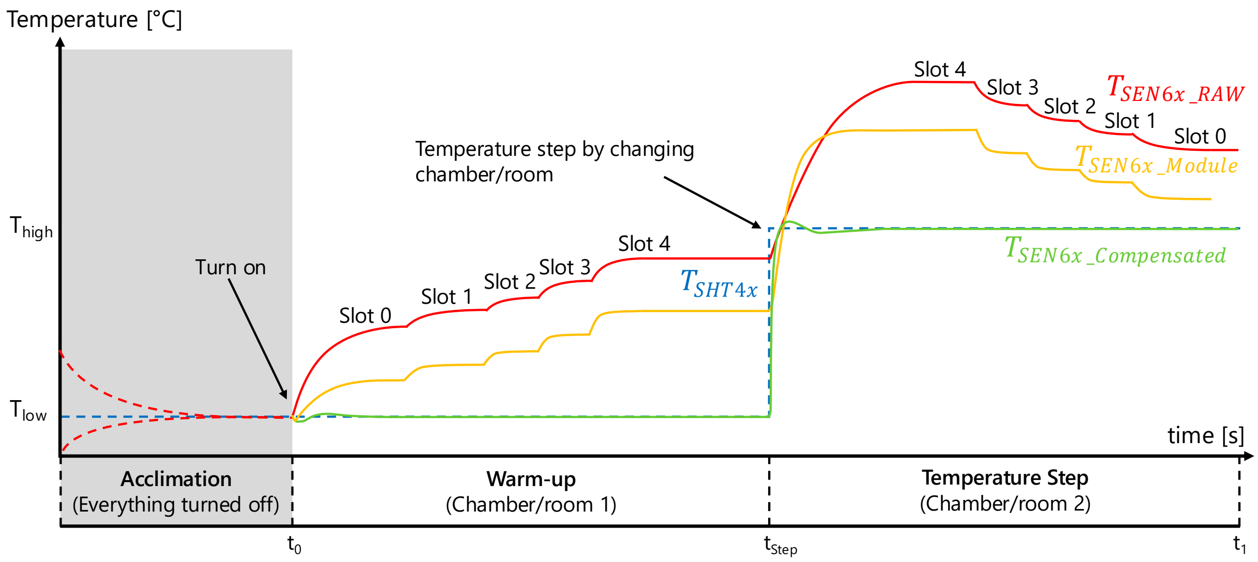

Figure 13. Experiment for parameter derivation.

If your device has multiple operation modes with significant difference in energy consumption or heat dissipation, such as a low power mode, display on/off feature, Wi-Fi module turning on/off, cooling fans, etc., the SEN6x Engine can dynamically compensate for up to 5 individual heat sources.
Each slot is defined by an offset, slope parameter (optional), and a time constant for the warm-up & cool-down.
The experiment and the parameter derivation are similar to those described in the **Sections 2.3.1** & **2.3.2** above. Additionally, make sure to note down the timestamp when a mode is switched on or off.

After the acclimation period in the first environment, switch the device through the different operation modes (always wait until a steady state is reached). The same procedure is done in reverse order in the second environment, to get the offset at the higher temperature level.

With the data, you can extract the offset and the time constant for each slot at two temperature levels (see **Figure 13**). Applying the same method as described in **Section 2.3.2**, the slope and offset correction factor at 0 °C can be calculated. It is important to note that for multiple slots, the temperature difference (ΔT) between two consecutive slots must be considered. For example, with three slopes: the first offset uses temperature values from the ambient air and the first environment; the next offset uses values from the first and second environments; and so on.

To apply the compensation for a specific mode, at the startup, the corresponding slot has to be activated (sending the I²C command with appropriate slot, offset, slope, and time constant). The slot will be added to the already active ones.

2.4 Parameter Verification

This section describes how to check if the parameter set derived from the experiment is stable in real-world application. We use the STAR-Engine simulation for this. If you would like to skip this step and test directly with your device, use the I^2C commands in **Section 3.2**.

The recorded sensor data (SEN6x (red, solid line) & SHT4x reference (green, solid line)) from the previous measurement with ControlCenter, can be opened with DataViewer. DataViewer allows post-processing acceleration and offset parameters on a given dataset.

First, the acceleration parameters must be set (see process described in **Figure 10**), after which the slots with the different offsets can be applied. Post-processing allows us to compare different engine parameter settings (red, dotted lines) to each other, and against the uncompensated temperature values. An example of post-processed compensation on a recorded step response is displayed in **Figure 14**.

The button *Update* calculates the STAR- Engine output based on the original data and your parameters. If you are not yet satisfied with the current engine performance, you can alter/tune the parameters and press the *Update* button again until the performance matches your application.

Note: Deviating too far from the previously determined parameters (**Table 1**) may lead to an unstable system. Here, a more thorough testing (**Section 2.4.1.**) on real-world data sets is recommended to ensure stable performance in all situations.

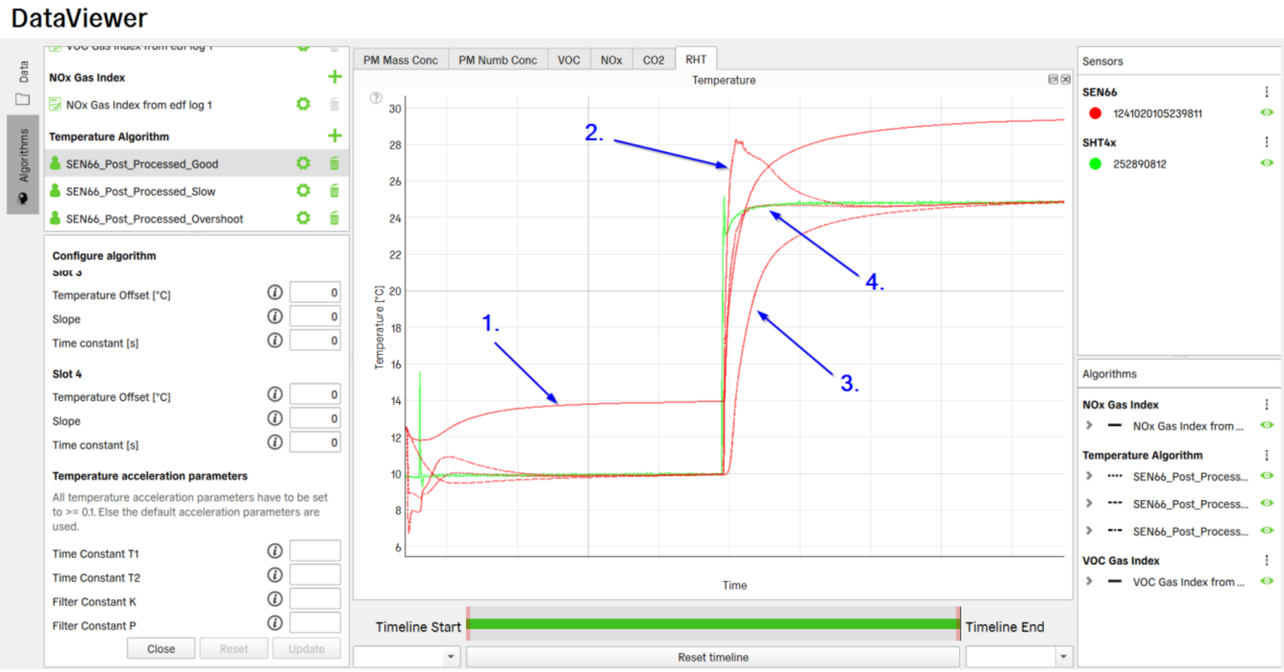

Figure 14. Sensirion DataViewer, STAR-Engine post-processing: 1. Showing the uncompensated T_{Module} 2. Showing an over-accelerated signal with overshoot, 3. Showing a slow signal with not enough acceleration and 4. Showing a well-tuned signal.

2.4.1 Stability & Real-World Application

To verify the stability of your parameter set, we recommend using real-world data recordings to test the performance, especially if you deviate from the recommendations in **Table 1** for the acceleration parameters.

There are the following corner cases we recommend for testing:

1. **Warm-start:** The device runs continuously for several hours, then gets a short power outage and restarts. The device is already warm due to the previous self-heating, but with the restart, this information is lost, and the compensation starts with the assumption that the device is cold and starts

to warm up. Here some over/undershoots are expected. Check if the amount works in your application. See **Section 2.4.2** for more information.

2. **Measurement noise:** Fluctuations in the raw data will be amplified due to the acceleration, so we recommend testing situations with high noise (for example user interaction: breathing, touching, carrying around).
3. **Faster step**: If the device experiences a temperature step that is faster than the one taken for tuning, we expect that the compensated output will show increased over/undershoots or oscillations. Be extra careful to verify that there are no faster temperature changes (additional unaccounted electronics, forced convection etc.) in your real-world datasets, and if so that the applied acceleration will not produce too large artifacts.

2.4.2 Warm-start

Figure 15 below shows the module's temperature raw signal and compensated output for an experiment with two modules and a reference sensor (SHT4x, grey dashed region) on a wooden table. The modules are powered up consecutively with a delay of approximately 10 minutes. As the modules warm up due to self-heating (cold-start), the temperature reading from the internal temperature sensor (T_SHT, dashed line) increases, while the compensated output (solid line) shows an accurate reading of ambient temperature. After approximately 45 minutes, the module described by the orange line was turned off and on again, simulating a power cycle (warm-start). Since the module was already warm, in the very first moments of operation, the temperature reading will show values larger than ambient temperature until the built-in algorithm detects the warm-start and quickly compensates for the current conditions. After 7 minutes, the sensor is within specifications.

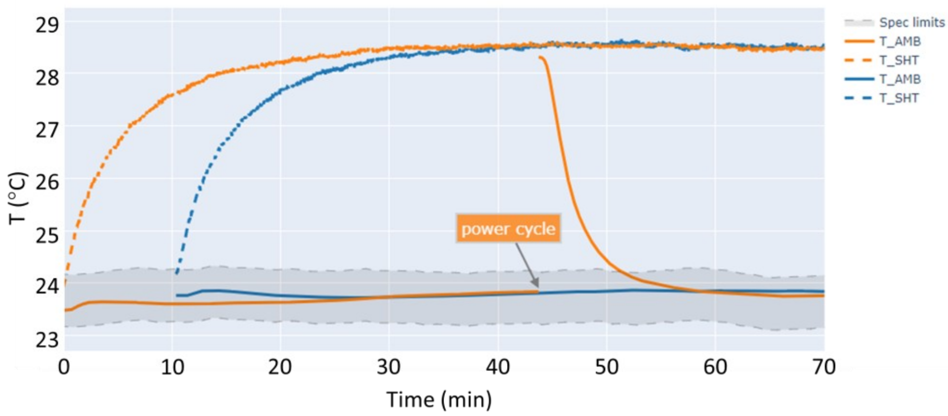

Figure 15. Warm-start.

If it is known that the device is subject to a warm-start, this effect can be compensated by applying a dedicated dynamic offset, negating the behavior seen above. Right after a warm-start, an offset with a τ_{63} = 0 is applied using one of the slots of the *Set Temperature Offset Parameters* command, seen in **Table 3**. This will offset the initial overshoot observed in the plot above back to zero, right after the start of the power cycle.
However, the engine immediately starts to compensate for the warm-start, so we have to gradually 'turn off' this warm-start offset. Turning it off can be done by setting the same Slot to an offset of 0 and selecting τ_{63} so that it matches the decay observed in the plot. After the engine has recovered from the warm-start, this Slot will not be offsetting anything anymore (or more precisely: offsetting by 0 °C) so the rest of the measurement will be correct again.

3 Applying Parameters to the STAR-Engine

For development purposes, we recommend you use ControlCenter, as you can easily apply different parameter sets to the recorded data in DataViewer. For the final integration into your device, and for testing your firmware, you will use the I^{2}C commands.

3.1 ControlCenter

Note: The acceleration parameters can only be set in idle mode, not in the active measurement mode. The parameters are saved in volatile memory, which resets after every power cycle.

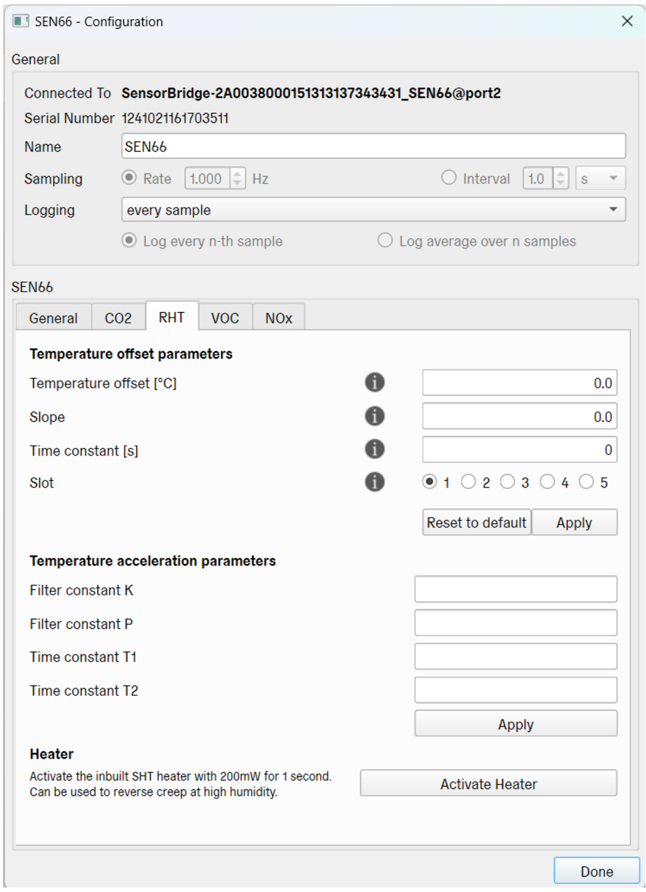

Figure 16. Setting the STAR-Engine parameters in ControlCenter.

3.2 I²C Commands

Note: In the I²C commands, there are scale factors (listed below) that must be applied to the parameters! The parameters are saved in volatile memory, which resets after every power cycle.

The I²C command 0x6100 is available to set the temperature acceleration parameters. A default module acceleration is already applied to the compensated output of the module. If the signal acceleration needs to be tuned for the application, the default parameters are overwritten with your custom parameters.

<table>
<tr><th colspan="4">Set Temperature Acceleration Parameters</th></tr>
<tr><td>Command ID</td><td colspan="3">0x6100</td></tr>
<tr><td>Available in</td><td colspan="3">Idle mode</td></tr>
<tr><td>Execution Time</td><td colspan="3">20 ms</td></tr>
<tr><td>Max. RX Data With CRC</td><td colspan="3">0 Bytes</td></tr>
<tr><td rowspan="13">TX Data</td><td colspan="2">Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>MSB</td><td rowspan="3">K: <code>uint16</code>
Filter constant K scaled with factor 10 (K = value / 10).</td></tr>
<tr><td>1</td><td>LSB</td></tr>
<tr><td>2</td><td>CRC</td></tr>
<tr><td>3</td><td>MSB</td><td rowspan="3">P: <code>uint16</code>
Filter constant P scaled with factor 10 (P = value / 10).</td></tr>
<tr><td>4</td><td>LSB</td></tr>
<tr><td>5</td><td>CRC</td></tr>
<tr><td>6</td><td>MSB</td><td rowspan="3">T1: <code>uint16</code>
Time constant T1 scaled with factor 10 (T1 [s] = value / 10).</td></tr>
<tr><td>7</td><td>LSB</td></tr>
<tr><td>8</td><td>CRC</td></tr>
<tr><td>9</td><td>MSB</td><td rowspan="3">T2: <code>uint16</code>
Time constant T2 scaled with factor 10 (T2 [s] = value / 10).</td></tr>
<tr><td>10</td><td>LSB</td></tr>
<tr><td>11</td><td>CRC</td></tr>
<tr><td>RX Data</td><td colspan="3">None</td></tr>
</table>

Table 2. Set temperature acceleration parameters I²C command description

The I²C command 0x60B2 is available to set the temperature offset and slope parameters. The module already has a dynamic offset compensation for the internal self-heating (this is calibrated individually for every sensor). The offset command does not overwrite the existing values, but will be added to the offsets of the module.

<table>
<tr><th colspan="4">Set Temperature Offset Parameters</th></tr>
<tr><td>Command ID</td><td colspan="3">0x60B2</td></tr>
<tr><td>Available in</td><td colspan="3">Idle and measurement mode</td></tr>
<tr><td>Execution Time</td><td colspan="3">20 ms</td></tr>
<tr><td>Max. RX Data With CRC</td><td colspan="3">0 Bytes</td></tr>
<tr><td rowspan="13">TX Data</td><td colspan="2">Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>MSB</td><td rowspan="3">Offset: <code>int16</code>
Constant temperature offset scaled with factor 200 (T [°C] = value / 200).</td></tr>
<tr><td>1</td><td>LSB</td></tr>
<tr><td>2</td><td>CRC</td></tr>
<tr><td>3</td><td>MSB</td><td rowspan="3">Slope: <code>int16</code>
Normalized temperature offset slope scaled with factor 10000 (applied factor = value / 10000).</td></tr>
<tr><td>4</td><td>LSB</td></tr>
<tr><td>5</td><td>CRC</td></tr>
<tr><td>6</td><td>MSB</td><td rowspan="3">Time Constant: <code>uint16</code>
The time constant determines how fast the new slope and offset will be applied. After the specified value in seconds, 63% of the new slope and offset are applied. A time constant of zero means the new values will be applied immediately (within the next measure interval of 1 second).</td></tr>
<tr><td>7</td><td>LSB</td></tr>
<tr><td>8</td><td>CRC</td></tr>
<tr><td>9</td><td>MSB</td><td rowspan="3">Slot: <code>uint16</code>
The temperature offset slot to be modified. Valid values are 0 .. 4. If the value is outside this range, the parameters will not be applied.
Note: A total of five slots are available. Each slot represents one temperature offset. Usually slot 0 is used to compensate for the base self-heating and the other slots allow to compensate for additional heating of components like screens, Wi-Fi modules, etc. that can be switched on and off independently.</td></tr>
<tr><td>10</td><td>LSB</td></tr>
<tr><td>11</td><td>CRC</td></tr>
<tr><td>RX Data</td><td colspan="3">None</td></tr>
</table>

Table 3. Set temperature offset parameters I²C command description

4 Bibliography

[1] Sensirion AG, "SEN6x – Mechanical design and assembly guidelines," May 2025. [Online]. Available: https://sensirion.com/resource/application_note/SEN6x_mechanical_design_assembly_guidelines.

[2] Sensirion AG, "Humidity at a glance," May 2025. [Online]. Available: https://sensirion.com/resource/application_note/sht/glance.

[3] Sensirion, "ControlCenter," Sensirion, [Online]. Available: https://sensirion.com/products/sensor-evaluation/control-center.

5 Revision History

Date	Version	Pages	Changes
August 2025	0.8	all	Initial version
December 2025	1.0	all	Changed illustrations, Renamed Variables, Changed to D1

Important Notices

Warning, Personal Injury

Do not use this product as safety or emergency stop devices or in any other application where failure of the product could result in personal injury (including death). Do not use this product for applications other than its intended and authorized use. Before installing, handling, using or servicing this product, please consult the data sheet and application notes. Failure to comply with these instructions could result in death or serious injury.

If the Buyer purchases or uses SENSIRION products for any unintended or unauthorized application, Buyer shall defend, indemnify and hold harmless SENSIRION and its officers, employees, subsidiaries, affiliates and distributors against all claims, costs, damages and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if SENSIRION is allegedly negligent with respect to the design or the manufacture of the product.

ESD Precautions

The inherent design of this component causes it to be sensitive to electrostatic discharge (ESD). To prevent ESD-induced damage and/or degradation, take customary and statutory ESD precautions when handling this product. See application note "ESD, Latchup and EMC" for more information.

Warranty

SENSIRION solely warrants to the original purchaser of this product for a period of 12 months (one year) from the date of delivery that this product is of the quality, material and workmanship defined in SENSIRION's published specifications of the product. Within such period, if proven to be defective, SENSIRION shall as sole and exclusive remedy, in SENSIRION's discretion, repair this product or send a replacement product, free of charge to the Buyer, provided that:

- notice in writing describing the defects shall be given to SENSIRION within fourteen (14) days after their appearance;
- such defects shall be found, to SENSIRION's reasonable satisfaction, to have arisen from SENSIRION's faulty material or workmanship;
- the defective product shall be returned to SENSIRION's factory at the Buyer's expense; and
- the warranty period for any repaired or replaced product shall be limited to the unexpired portion of the original period.

The Buyer shall at its own expense arrange for any dismantling and reassembly that is necessary to repair or replace the defective product. This warranty does not apply to any product which has not been installed or used within the specifications recommended by SENSIRION. EXCEPT FOR THE WARRANTIES EXPRESSLY SET FORTH HEREIN, SENSIRION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE PRODUCT. ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE EXPRESSLY EXCLUDED AND DECLINED.

SENSIRION is only liable for defects of this product arising under the conditions of operation provided for in the data sheet and proper use of the goods. SENSIRION explicitly disclaims all warranties, express or implied, if the goods are operated or stored not in accordance with the technical specifications.

SENSIRION does not assume any liability arising out of any application or use of any product or circuit and specifically disclaims any and all liability, including without limitation indirect, consequential, and incidental damages, and loss of profit. No obligation or liability shall arise or grow out of SENSIRION's rendering of technical advice, consulting, or implementation instructions or guidelines. All operating parameters, including without limitation recommended parameters, must be validated for each Buyer's applications by Buyer's technical experts. Recommended parameters can and do vary in different applications.

SENSIRION reserves the right, without further notice, (i) to change the product specifications and/or the information in this document and (ii) to improve reliability, functions and design of this product.

Headquarters and Subsidiaries

Sensirion AG
Laubisruetistr. 50
CH-8712 Staefa ZH
Switzerland
phone: +41 44 306 40 00
fax: +41 44 306 40 30
info@sensirion.com
www.sensirion.com

Sensirion Inc., USA
phone: +1 312 690 5858
info-us@sensirion.com
www.sensirion.com

Sensirion Korea Co. Ltd.
phone: +82 31 337 7700~3
info-kr@sensirion.com
www.sensirion.com/kr

Sensirion Japan Co. Ltd.
phone: +81 45 270 4506
info-jp@sensirion.com
www.sensirion.com/jp

Sensirion China Co. Ltd.
phone: +86 755 8252 1501
info-cn@sensirion.com
www.sensirion.com/cn

Sensirion Taiwan Co. Ltd
phone: +886 2 2218-6779
info@sensirion.com
www.sensirion.com

To find your local representative, please visit
www.sensirion.com/distributors