ENS161
Operating Modes
Application note

ENS161 Operating Modes Application Note

Version: 1.0
Release Date: 2024-12-12
Document Status: Production

Table of Contents

1 General Description

The ENS161 is a multi-gas sensor with I^2C and SPI interface using a LGA package. Composed of a metal oxide layer (MOX) on a sensing chip and a heater underneath, the sensor's MOX changes its electrical resistance depending on the ambient gas composition, from which the air quality can be derived. The principle is very straightforward: oxidizing gases elevate resistance, while reducing gases lower resistance.

The ENS161 offers effortless integration, reliable performance and low-power modes, with minimal system requirements. Supported with improved and intelligent algorithms, the sensor automatically turns raw data into total Volatile Organic Compound (TVOC), equivalent CO_2 (eCO_2) and additional air quality indices, without any overhead needed by the end user.

The ENS161 incorporates energy-saving features designed to enhance efficiency and minimize power consumption. On one side, the internal MCU handles all data conversion and computation that means saving the host CPU from heavy processing loads; on the other side, the sensor operates in three different power modes: Standard, Low Power, and Ultra-Low Power. This flexibility enables balancing energy consumption with performance, based on the specific requirements of the application.

This Application Note provides a comprehensive overview of the ENS161 sensor's various operating modes, focusing on performance, consumption, and practical implementation. Through detailed explanations and case studies, it highlights the advantages, limitations, and best practices for transitioning between modes, managing environmental adaptability, and optimizing sensor usage in diverse scenarios.

2 Sensing Modality

The ENS161 is a broadband gas sensor, based on metal oxide (MOX) technology with four sensor elements supporting isothermal and low-power operating modes plus an unrivaled wealth of fully processed output signals.

The broadband sensitivity refers to the sensor’s ability to detect the overall gas concentration across different chemical compositions instead of a single gas, since the main purpose of MOX technology is monitoring indoor air quality. From building materials to everyday activities like cooking, our indoor spaces are inundated with volatile organic compounds (VOCs), an ever-present yet invisible threat, often ignored or underestimated.

The MOX technology uses tin dioxide (SnO_2), a granular semiconductor material, applied to an electrode structure and being sensitive to gases when heated to temperatures from 200 to 400 °C. The MOX sensor operates by absorbing oxygen from the air onto its semiconductor particles, capturing electrons and creating a depletion zone in the material. By adjusting this zone based on the presence of gases, it is possible to measure gas concentration and gather crucial data on air quality and environmental conditions. This method, utilizing semiconductor materials to detect changes in the air's chemical composition, provides the most sensitive and fastest response compared to other technologies, revolutionizing environmental monitoring.

The independent hotplate control allows the detection of a wide range of VOCs including ethanol, toluene, hydrogen and oxidizing gases with superior sensitivity. The ENS161 supports intelligent algorithms to process raw sensor measurements on-chip. These algorithms calculate TVOC- and CO_2-equivalents, various air quality indices (AQIs) and perform humidity and temperature compensation, as well as baseline management, all on chip.

These real-time outputs allow for proactive adjustments to ventilation settings, ensuring a consistently healthy indoor environment based on your occupancy needs. While not immediately life-threatening, gases in closed spaces can have negative consequences over time, potentially leading to various health issues. Embracing efficient ventilation technologies creates comfortable indoor environments, safeguarding the planet.

Raw sensor measurements can be read for further customization. The LGA-packaged device includes SPI and I^2C slave interfaces to communicate with a main host processor.

The ENS161 is a proven and maintenance-free technology, designed for high volume and reliability.

In the chapters that follow, considerations and recommendations are made to integrate ENS161 into an electronic system with ease and to fully leverage the benefits. Beginning with electrical, followed by mechanical integration, insightful recommendations will be provided to ensure a clear and systematic process.

3 Overview of available operating modes

This section provides an in-depth guide to the various operating modes of the ENS161 sensor, offering insights into how each mode functions, the relative power consumption and performance, and the register setup required for smooth operation.

3.1 IDLE Mode (0x01)

Description: The default mode where the ENS161 is powered but not actively sensing gas. This mode is designed for communication and configuration tasks.

Operation:

- The sensor is not running its gas-sensing functionality, allowing the host system to send and receive commands.
- The **INTn** pin can be used for signalling when new data is available or when configuration is complete.
- The sensor transitions to this mode from other modes (e.g. DEEP SLEEP or an active sensing mode) to prepare for new commands.

Usage: This mode is often used before switching to an active sensing mode (STANDARD, LOW POWER, or ULTRA LOW POWER) and represents a mandatory transition point when moving from one operating mode to another.

Registers:

- Set by writing 0x01 to the **OPMODE** register (address 0x10).
- Transitions from or to active modes require passing through IDLE mode.

Current Consumption: typical value is around 2mA.

3.2 DEEP SLEEP Mode (0x00)

Description: The most power-efficient mode, where the ENS161 consumes the least energy.

Operation:

- The sensor is in a standby state, with minimal functionality. It cannot perform gas sensing in this mode.
- The sensor remains responsive only to commands that can wake it up (e.g., transitioning to IDLE mode).

Usage: Ideal for energy-saving scenarios where the system doesn't need constant gas monitoring but must wake the sensor occasionally.

Registers: Set by writing 0x00 to the **OPMODE** register (address 0x10).

Current Consumption: Extremely low, around 10 µA, but no gas sensing can be performed.

3.3 STANDARD Mode (0x02)

Description: The main active gas-sensing mode, providing reliable performance for detecting air quality.

Operation:

- The sensor takes one gas sample per second, providing continuous air quality data output.
- New gas data is made available in the **DATA_XXX** registers. The host system can either poll these registers or use the **INTn** pin (if enabled via the **CONFIG** register) to get notified when new data is ready.
- This mode ensures high sensitivity to air quality, but at the cost of higher power consumption compared to other modes.

Usage: Best suited for real-time applications where precise and regular gas measurements are required.

Registers: Set by writing 0x02 to the **OPMODE** register. Gas data can be accessed via registers such as

- **DATA_ETVOC** (address 0x22) for Total Volatile Organic Compounds (TVOC).
- **DATA_ECO2** (address 0x24) for Equivalent CO_2.
- **DATA_AQI_UBA** (address 0x21) for Air Quality Index according to UBA (German Federal
- Environmental Agency).
- **DATA AQI_S** (address 0x26) for Air Quality Index according to ScioSense

Status can be checked through the **DEVICE_STATUS** register (address 0x20).

Current Consumption: 12mA obviously higher than other modes, but providing continuous, reliable measurements.

3.4 LOW POWER Mode (0x03)

Description: A power-saving mode that still allows gas sensing, but at a slower sampling rate.

Operation:

- The sensor takes one sample per minute or more specifically the heater underlying the hotplate is off for 57s and on for 3s, balancing energy efficiency with the need for occasional gas measurements.
- Data is still provided through the **DATA_XXX** registers, but due to the slower sampling rate, responsiveness to rapid changes in gas concentration is lower.
- This mode is useful in scenarios where air quality changes slowly or the application can tolerate less frequent measurements.

Usage: Ideal for battery-powered or energy-constrained applications where power consumption needs to be minimized without fully sacrificing gas sensing.

Registers: Set by writing 0x03 to the **OPMODE** register.

- Data can be retrieved from the same registers used in STANDARD mode.

Current Consumption: The average current consumption is around 700 µA, obviously lower than STANDARD mode but higher than ULTRA LOW POWER mode.

3.5 ULTRA LOW POWER Mode (0x04)

Description: The most energy-efficient gas-sensing mode, designed to save power by significantly reducing the sampling rate.

Operation:

- The sensor takes one sample every 5 minutes, meaning that the heater is off for 299s and on for 1s, which greatly reduces energy consumption. However, short gas events might not be detected, as the sampling rate is too low to capture rapid changes.
- This mode is suitable for long-term monitoring in environments where gas concentration changes gradually or infrequently.

Usage: This mode is best for applications that require minimal power consumption and can afford to miss short-term changes in air quality.

Registers: Set by writing 0x04 to the **OPMODE** register. Data retrieval is the same as in other active modes but occurs less frequently.

Current Consumption: The lowest among active modes, with an average current consumption around 150 µA, but with potential trade-offs in data accuracy and sensitivity to short-duration events.

3.6 Essential Overview: Modes, Registers and Data handling

Key Registers:

OPMODE register (Address 0x10) controls the operating mode. The mode values are:

0x00: DEEP SLEEP

0x01: IDLE

0x02: STANDARD

0x03: LOW POWER

0x04: ULTRA LOW POWER

Mode Transitions:

- Before switching between active modes (STANDARD, LOW POWER, ULTRA LOW POWER), the sensor must first transition to **IDLE mode**. This allows the sensor to reset and avoid issues with warm-up times or incomplete transitions.
- It is not recommended to frequently switch between active modes, as this can cause algorithms to restart or prolong the sensor's warm-up phase, potentially affecting performance.

CONFIG Register (Address 0x11):

- The **CONFIG** register configures the behaviour of the interrupt pin (**INTn**), which can notify the host system when new gas data is available. For example, setting the register to 0x23 enables an active-low interrupt when new data is ready in the **DATA_XXX** registers.

Interrupt and Data Polling: Data can be accessed either by polling the DEVICE_STATUS register to check for new data or by enabling the **INTn** pin for automatic notifications when data is ready.

DEVICE_STATUS Register (Address 0x20) works in the following way:

- Bit 1: **NEWDAT** – Indicates if new data is available in the **DATA_XXX** registers.
- Bit 0: **NEWGPR** – Indicates if new data is available in the **GPR_READ** registers (used for general-purpose data transfer).

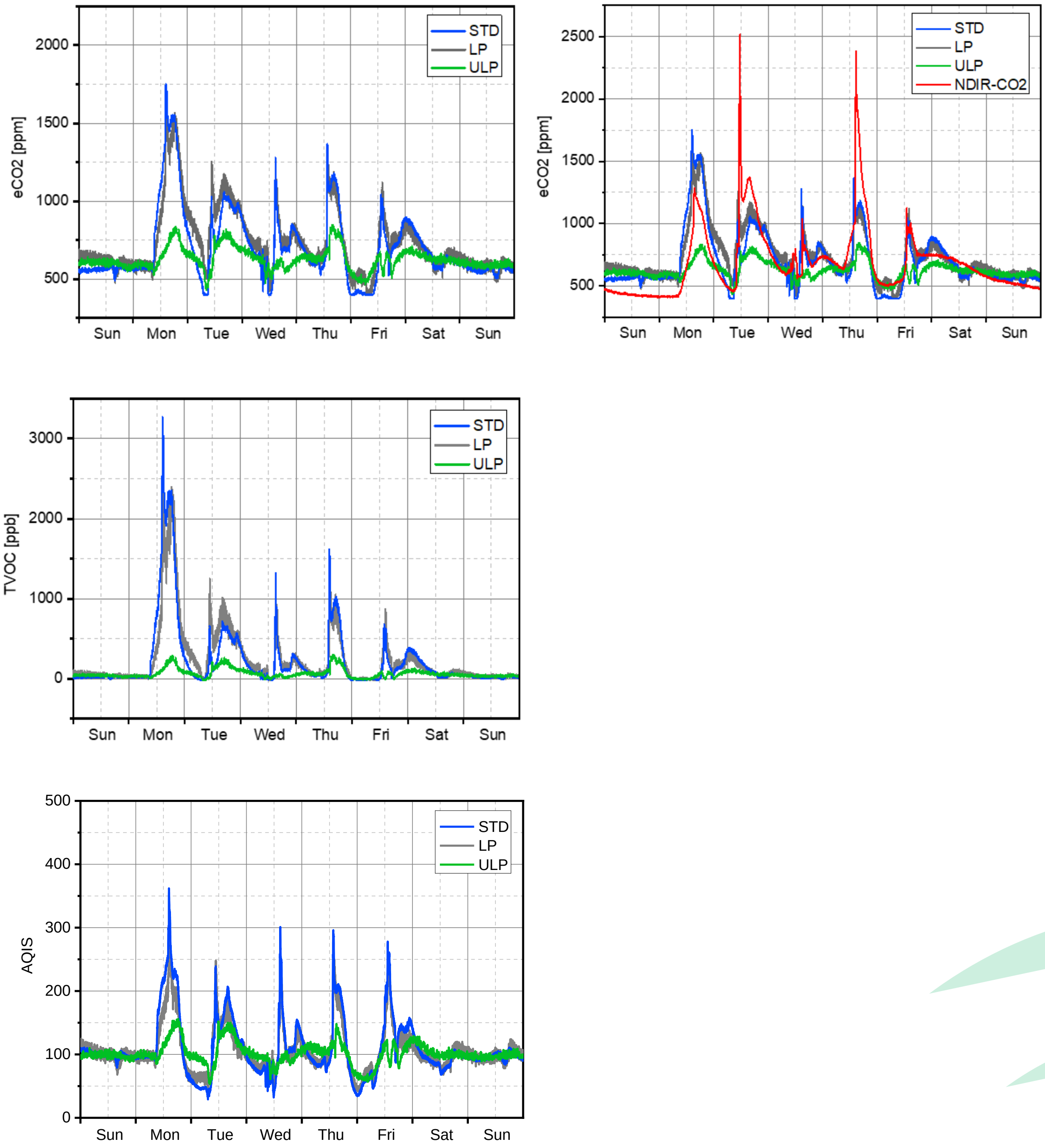
4 Operating Mode in the Real World

This section provides key insights into the sensor's various output metrics, comparing these indices across the three operating modes in a typical indoor scenario, such as a meeting room.

4.1 Performance of Three Main Operating Modes

ENS161 offers three operating modes based on the energy consumption requirements of the target application and how many measurements over time the application needs.

Figure 1: Comparison of different output waveforms in STD, LP and ULP Mode



The presented waveforms aim to evaluate the performance of the ENS161 sensor in its various operating modes and to illustrate both the advantages and limitations of ScioSense’s MOX technology. We have chosen to display the three operating modes side by side for each of the sensor's outputs: TVOC, eCO_2 (with and without NDIR reference sensor), and AQI-S. The test environment was a meeting room typically occupied for intervals throughout the week—an ideal setting to assess air quality, which attendees often notice intuitively. Before describing the waveforms in detail, we provide a key for understanding the TVOC, eCO_2, and AQI-S metrics.

A standout feature of the ENS161 sensor is its range of post-processed outputs, especially TVOC, eCO_2, and AQI-S.

- AQI-S is a new and widely recognized air quality index on a 0–500 scale, enabling simple, real-time data interpretation.
- The TVOC value represents the sum of all VOCs in the ambient air (there are approximately 5,000 to 10,000 different VOCs in existence), which are typically 2-5 times more concentrated indoors than outdoors due to multiple sources. As VOCs are a major contributor to indoor air pollution, tracking TVOC levels is essential for maintaining good air quality.
- The eCO_2 output provides a standardized measurement in ppm that aligns with CO_2 standards: as a compromise to directly measuring CO_2, ENS161 provides equivalent CO_2 (eCO_2) readings. These should not be mistaken for actual CO_2 measurements. eCO_2 is calculated using the "Reverse Metabolic Rule," which adapts the "Pettenkofer law" to align eCO_2 signals with CO_2. This correlation works as the eCO_2 value is derived from measured VOCs plus hydrogen, as an indicator of CO_2 levels exhaled by humans. The term “equivalent” specifies that eCO_2 is an indirect measurement—estimated from TVOC and hydrogen content rather than directly measuring CO_2. The ENS161’s ability to detect odours and bio-effluents untraceable by CO_2 sensors offers a substantial advantage, especially in indoor applications where the primary CO_2 source is human activity.

This versatility empowers users across smart homes, commercial buildings, and industrial environments to quickly adjust ventilation or activate purifiers, ensuring healthier indoor air quality. Always ensure that the number of measurements is sufficient to detect significant changes in the application where it's deployed.

Focusing now on the waveforms to examine performance:

1. the Standard mode is highly responsive to any changes, making it very effective, although it may consume too much power in battery-operated applications like thermostats or small appliances.
2. Low Power mode offers a compelling alternative; it reduces power consumption by a factor of about 17 while delivering reliable, comparable performance to Standard mode. ScioSense’s daily measurements have consistently shown Low Power mode to yield satisfactory and dependable results.
3. Lastly, the ULP mode should be considered carefully for applications where event sensitivity and information retention are less critical. The general trend in ULP mode aligns with the other two modes, but it is less responsive to variations. It is best suited for applications with gradual air quality changes, where the output signal can be amplified if needed.

4.2 Use Cases

Here’s a breakdown of use cases for deploying the MOX sensor in Standard, Low Power, and Ultra-Low Power modes:

1. Standard Mode:
Best for environments requiring high accuracy and quick response times, where power consumption is less of a concern.

- **Smart Home Hubs:** Regularly measure air quality and promptly alert users to sharp increases in VOCs or eCO_2, ensuring quick response to maintain a healthy environment.
- **Commercial Building Ventilation Systems:** Continuously monitor air quality in real-time, adjusting HVAC operations as soon as indoor pollutants increase, maximizing occupants' comfort.
- **Laboratories and Clean Rooms:** Detect even minor fluctuations in air quality to maintain stringent indoor standards for health and safety.

2. Low Power Mode:
Ideal for applications needing continuous but less frequent measurements, where power savings are a priority without sacrificing performance.

- **Battery-Operated Thermostats and Small Appliances:** Maintain air quality measurements with minimal power consumption, making this mode especially useful for long battery life.
- **Smart Office Systems:** Provide reliable air quality data throughout the day to adjust ventilation without draining power, aligning with energy-efficient building strategies.
- **Public Transportation and Aircraft Cabins:** Efficiently monitor and respond to air quality changes, ensuring passenger comfort while preserving battery life in transportation equipment.

3. Ultra-Low Power Mode (ULP):
Most suitable for applications where slow, gradual air quality monitoring suffices, with strict power limitations for long-term operation.

- **IoT Air Quality Sensors for Smart Cities:** Use ULP mode to capture baseline trends in urban environments, providing long-term data with low maintenance.
- **Remote or Hard-to-Reach Locations:** Deploy in settings where sensors need to run autonomously for long periods (e.g., inside ductwork or ceiling-mounted sensors), tracking gradual air quality changes without requiring constant access or recharging.

5 Operating Mode - Software Guidelines

In this section, we'll demonstrate how straightforward it is to configure the ENS161 and generate the waveforms covered in the previous section. Only minimal programming knowledge is required from a software perspective, as the code and libraries for the Arduino IDE are publicly accessible at: https://github.com/sciosense. Here are the steps to set up the sensor and conduct measurements, allowing you to obtain the desired waveforms.

1. Include the libraries

```
#include <ScioSense_ENS16x.h>
#include "ens16x_i2c_interface.h"
using namespace ScioSense;
```

2. Choose the operating mode

```
ens161.startStandardMeasure();
OR
ens161.startLowPowerMeasure();
OR
ens161.startUltraLowPowerMeasure();
```

3. Setup the sensor

```
ens161.setInterruptPin(2);
   ens161.writeConfiguration
(
   ENS161::Configuration::InterruptEnable;
   ENS161::Configuration::NewGeneralPurposeData;
   ENS161::Configuration::NewData;
);
ens161.startStandardMeasure();
```

4. Retrieve the measurements

```
//Raw Resistance of hotplate3
Serial.print("\tRS3:");Serial.println(ens161.getRs3());
//Indoor Air Quality according to German Federal Office for Environment
Serial.print("AQIUBA:");Serial.print((uint8_t)ens161.getAirQualityInd
ex_UBA());
//Indoor Air Quality according to ScioSense
Serial.print("\tAQIScioSense:");Serial.print(ens161.getAirQualityInde
x_ScioSense());
//Total VOC
Serial.print("\tTVOC:");Serial.print(ens161.getTvoc());
//Equivalent CO2
Serial.print("\tECO2:");Serial.println(ens161.getEco2());
```

No additional initialization is needed. Integrated MCU eliminates concerns about the timing, wake-up operations or compensation processes. Data processing is entirely handed on-chip, without requiring post-processing steps or the use of additional libraries.

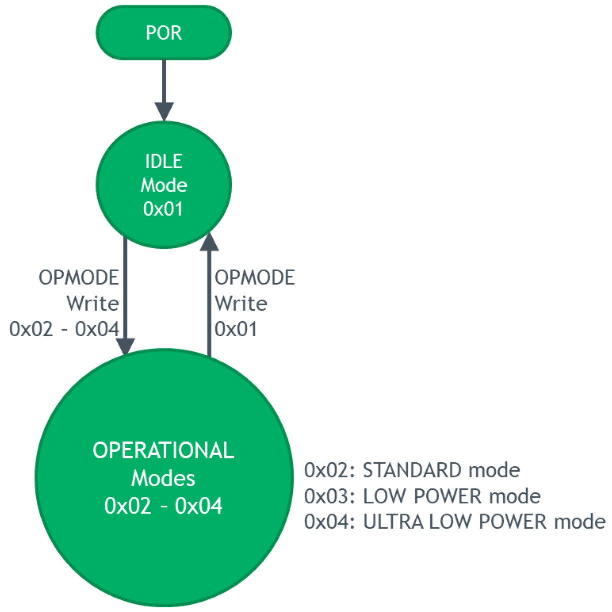
For an objective assessment of total energy consumption, it is also important to consider the significant advantage of the integrated MCU, which would further reduce the consumption outlined in the section 3 of this document.

6 Common questions and concerns

6.1 Error handling: Transition between modes

For transitioning between different operating modes, the sensor must be set into IDLE MODE first, as depicted in Figure 2. This intermediate IDLE step serves as a configuration phase before entering an active sensing mode. Failing to include this IDLE step may lead to inconsistencies or errors in the sensor's behaviour. Therefore, it is recommended to adhere to this protocol to ensure smooth and reliable operation.

Figure 2: ENS161 Operating Modes Diagram



When changing between operating modes, it’s essential to verify the device’s state by checking the **DEVICE_STATUS register (address 0x20)**, with a focus on the **VALIDITY FLAG** (Please see section 16.2.7 of the Datasheet).This flag confirms that the MCU has correctly interpreted and executed the command to place the device in IDLE mode. Ensuring this status is crucial to confirm the sensor is prepared before switching between operating modes.

The standard procedure is:

1. First, put the sensor into IDLE MODE.
2. Confirm that the transition was successful by reading the DEVICE_STATUS byte (0x20) to check the VALIDITY FLAG.
3. Setting the preferred Operating Mode.

Alternatively, if the precise status check isn’t feasible or necessary, you can wait for a certain duration after sending the IDLE command to give the device time to settle into IDLE mode. The minimum recommended wait times, based on the initial Operating Mode, are as follows:

- **From STD Mode to IDLE Mode**: Wait at least **10 ms**

- **From LP Mode to IDLE Mode**: Wait at least **2 seconds**
- **From ULP Mode to IDLE Mode**: Wait at least **25 seconds**

After waiting for these minimum durations, you can proceed to change the Operating Mode. However, please note these times are approximate. They offer a way to toggle between modes if you are not checking the DEVICE_STATUS byte. In summary:

- **Precise Approach**: Always read address 0x20 and check the VALIDITY FLAG to ensure the sensor is in IDLE mode.
- **Approximate Approach**: Wait the suggested durations above based on the initial mode, which should generally allow the processor to place the sensor in IDLE mode. Consider these time intervals as a guide for the **Software RESET** process. This time allows the processor to complete any current commands and be ready for new instructions.

ENS161 offers tremendous flexibility in terms of energy consumption. However, it's important to be aware that frequently toggling between active operating modes is not advisable.

The following code is provided to switch between different operating modes, specifically from Standard mode to Low Power mode. It can, of course, be applied to any other operating mode change as needed.

1. Define Registers (according to ENS161 Datasheet)

```
#define PART_ID_REG 0x00
#define STATUS_REG 0x20
#define OPERATION_MODE_REG 0x10
```

2. Define Operating Modes

```
#define MODE_IDLE 0x01
#define MODE_STANDARD 0x02
#define MODE_LOW_POWER 0x03
#define MODE_ULTRA_LOW_POWER 0x04
```

3. Set to STANDARD mode

```
writeRegister(OPERATION_MODE_REG, MODE_STANDARD);
```

4. Set to IDLE mode and read the status register

```
writeRegister(OPERATION_MODE_REG, MODE_IDLE);
status = readRegister(STATUS_REG);
```

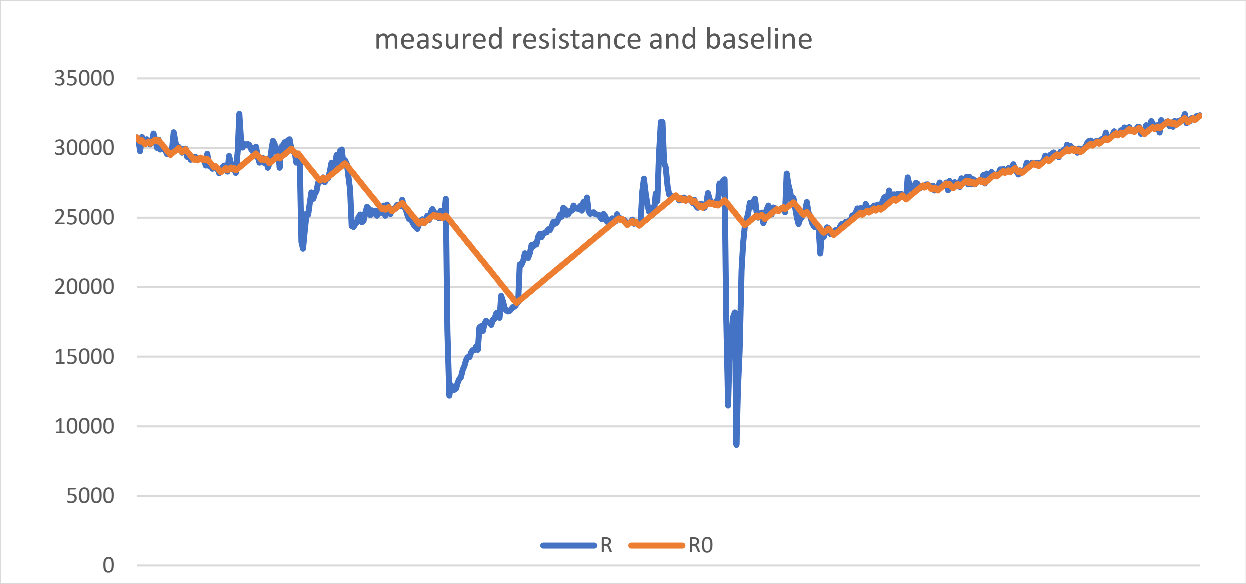
5. Set to LOW_POWER mode and read the status register

```
writeRegister(OPERATION_MODE_REG, MODE_LOW_POWER);
```

6.2 Baseline explanation

The ENS161, being metal oxide (MOX) sensor, relies on a "Baseline", which refers to an approximation to the sensor's highest resistance value seen during a certain timeframe of operation. An example for a moving baseline (R_0) is given in Figure 3.

Figure 3: Raw resistance versus Baseline



This Baseline serves as a reference point for detecting changes in the environment caused by target gases like VOCs. Maintaining this Baseline is essential to compensate for the drift that almost every MOX sensor experiences over time.
The integrated ASIC of the ENS161 includes an MCU that automatically calculates and adjusts the Baseline, requiring no effort or input from the customer. However, understanding the Baseline's role in detecting environmental changes and addressing drift can provide valuable insights into optimizing the ENS161's performance and gaining familiarity with MOX technology. While the specific algorithm used for Baseline management is proprietary, a description of its basic operating principles is provided to explain its underlying logic.

Principle of Baseline Calculation
In order to get familiar with the algorithm, it is important to understand the variables involved:

1. **Sensor Resistance Data (R):**
   This is the raw resistance value measured by the metal oxide sensor.
2. **Baseline Resistance (R_0):**
   The baseline resistance is a moving-average type reference value that the algorithm adjusts based on the sensor's measured resistance.
3. **Signal Calculation:**
   The signal is computed using the formula $\left(\left(\frac{R_0}{R}\right) - 1\right) * factor$), where $factor$ is a scaling constant. This signal could potentially be negative.
4. **Displayed Signal:**
   The displayed signal is always non-negative. If the computed signal is less than zero, it is set to zero.
5. **Threshold Application:**
   A threshold is applied to the displayed signal to determine if the signal should be adjusted further.

Therefore:

- given that R is the 'raw hotplate resistance' and R_0 is the 'reference value´;
- assuming for the first iteration that the value of R_0 is equal to the measured value R;
- choosing a positive or a negative addendum, respectively pos_add or neg_add;

the simple formula from which the more complex algorithm was developed is the following:

$$if\ R_{(i+1)} > R_{0i}\ \ then\ \ R_{0(i+1)} = R_{0i} + pos\_add$$

If the measured value is greater than the baseline value:
Add a constant value to the baseline to ensure that the baseline increases when the measured resistance increases, again closely following the sensor's resistance changes.

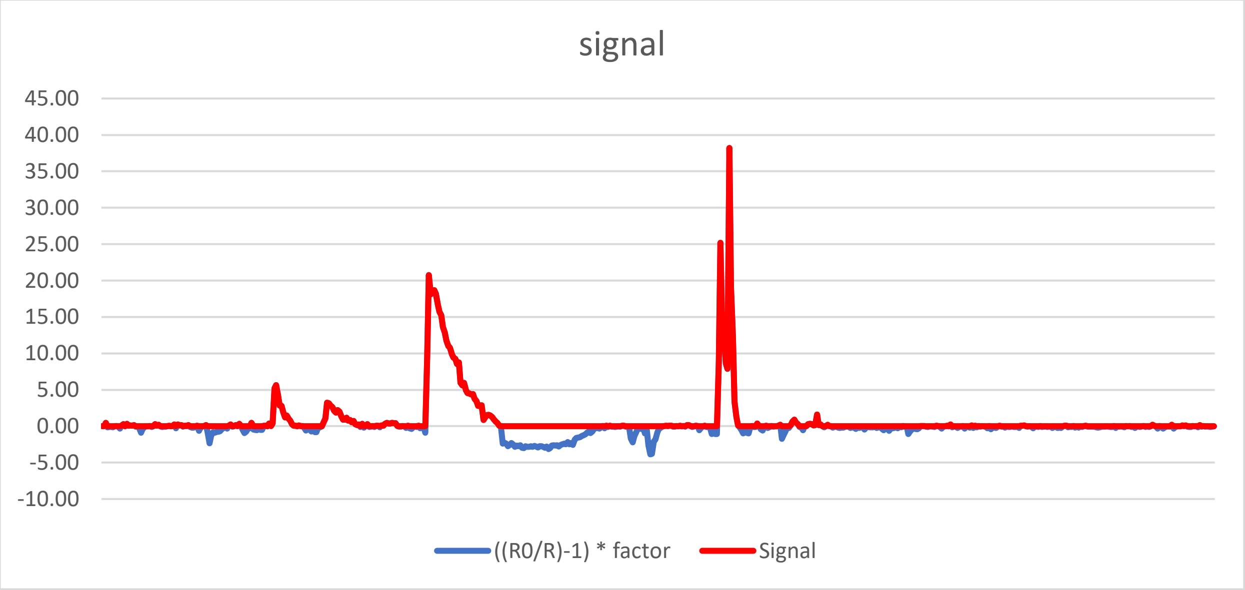
$$else\ R_{0(i+1)} = R_{0i} - neg\_add$$

The measured value is smaller than the baseline value:
Subtract a constant value from the baseline to ensure that the baseline decreases when the measured resistance decreases, maintaining a close tracking of the sensor's resistance changes.

Once again, Figure 3 provides an overview of the results of this additive and subtractive strategy, ensuring that the baseline remains consistently close to the raw resistance values. It shows how the sensor's measured resistance changes over time and how the baseline resistance (R_0) follows these changes.

Given the relationship between the measured resistance (R) and the baseline resistance (R_0), the adjustments made to keep R_0 aligned with the sensor's behaviour will result in a signal that reflects the deviation of the sensor's resistance from its expected baseline. Figure 4 shows the mentioned signal, which provides a meaningful representation of environmental changes, such as the presence and concentration of target gases like VOCs.

Figure 4: Signal derived by the comparison Raw resistance versus Baseline



Additionally, the baseline adjustment can be further refined by applying a factor and a positive threshold. These parameters enhance the algorithm's sensitivity and responsiveness, ensuring that the baseline accurately reflects the sensor's operating environment, as implemented in our actual algorithm.

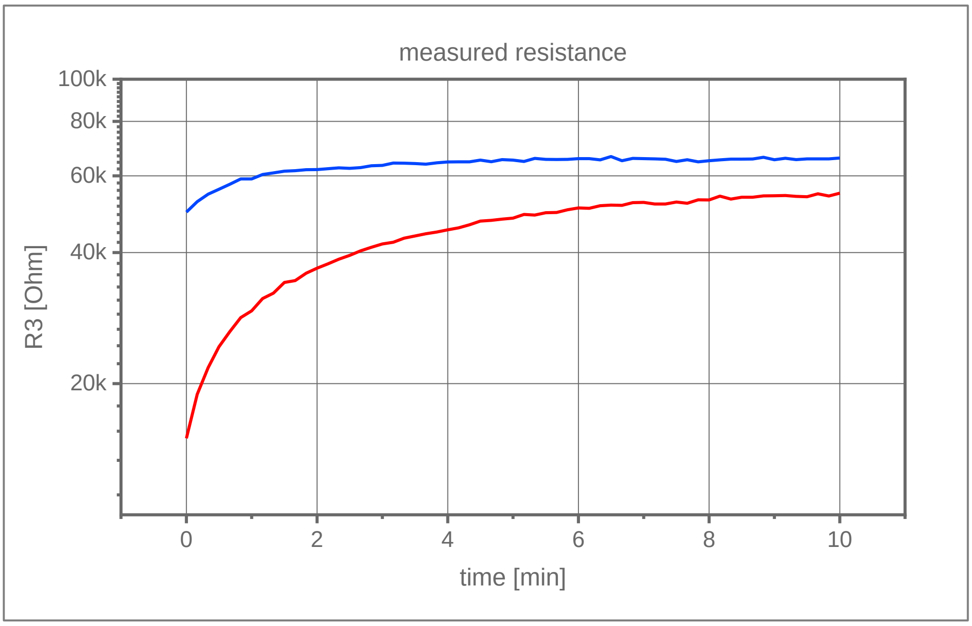
Algorithm Efficiency: The efficiency of the algorithm can be evaluated by examining how quickly and accurately the baseline resistance adapts to changes in the measured resistance.
Sudden spikes or drops in the displayed signal might indicate significant events or anomalies detected by the sensor.

Non-Linear Baseline Correction and Algorithm Insights
The baseline correction approach outlined here employs a simple method of adding or subtracting a constant value to compensate for long-term drifts, such as those caused by seasonal changes. However, advanced non-linear techniques, including the use of weighting factors and short-term projections, offer powerful customization for specific applications, as implemented with ENS161 sensors.
The baseline algorithm ensures that the signal remains non-negative and incorporates thresholds to maintain responsiveness to significant environmental changes. Visual representations of the adjustments further highlight the algorithm’s ability to adapt and optimize real-time sensor data monitoring.

6.3 Initial behaviour considerations

After starting an operation mode, the sensitive material is consuming absorbed VOCs and gets in equilibrium with the ambient air which is reflected as drift of sensor resistance.
Sensors powered off for hours and days (red curve) show a prolonged drift while sensors powered off for seconds or minutes (blue curve) are back to their initial resistance within seconds or a minute.
Drift of sensor resistance does not mean the sensor is not·sensitive to changes in the ambient gas concentration, but the overlaying drift might compensate small variations, especially during the first seconds after setting an operation mode where drift is the highest.
Therefore, it is recommended to allow the sensor to stabilize for a few minutes after starting an operation mode, especially in applications requiring precise measurements immediately after power-up.

Figure 5: Raw resistance at warm-up in two different scenarios: in blue ENS161 was previously powered off for days and in red for minutes



7 Copyrights & Disclaimer

Copyright **Sciosense B.V., High Tech Campus 10, 5656 AE Eindhoven, The Netherlands**. 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.

Devices sold by Sciosense B.V. are covered by the warranty and patent indemnification provisions appearing in its General Terms of Trade. Sciosense B.V. makes no warranty, express, statutory, implied, or by description regarding the information set forth herein. Sciosense B.V. reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with Sciosense B.V. for current information. This product is intended for use in commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by Sciosense B.V. for each application. This product is provided by Sciosense B.V. “AS IS” and any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

This product is not intended for use as safety device and/or in applications where failure of the product could result in personal injury. Sciosense B.V. 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 Sciosense B.V. rendering of technical or other services.

8 Revision information

Table 1: Revision history

Revision	Date	Comment	Page
1.0	2024-12-12	First version	All

Note(s) and/or Footnote(s):

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

Address: Sciosense B.V.
High Tech Campus 10
5656 AE Eindhoven
The Netherlands

Contact: www.sciosense.com
info@sciosense.com