MICRONAS

Application Note

CUR 4000

User Manual

cur SENS Technology

Edition June 16, 2021
APN000173_002EN

Copyright, Warranty, and Limitation of Liability

TDK-Micronas Trademarks

– HAL
– curSENS

Third-Party Trademarks

Contents

Contents

1. General Information

This document is intended as guidance manual on how to use the CUR 4000 sensor. Every integrated functionality of the sensor is described. The registers, their access and the necessary sensor commands are specified. Additionally all calibration and linearization procedures are explained in detail. In combination with the respective data sheet and the application notes “CUR 4000 Programming Guide” and “CUR 4000 Programming Environment” it represents the complete customer documentation of the CUR 4000 linear and differential magnetic flux sensor for current sensing applications.

1.1. Certification

TDK-Micronas GmbH fulfills the requirements of the international automotive standard IATF 16949 and is certified according to ISO 9001. This ISO standard is a worldwide accepted quality standard.

1.2. Support

We kindly ask you to register on https://service.micronas.com in order to obtain access to the workgroups for our various product families. Here you are able to get support by opening a support ticket.

TDK-Micronas GmbH – Application Engineering

Hans-Bunte-Strasse 19

D-79108 Freiburg im Breisgau

2. Functional Description

2.1. General Function

The CUR 4000 is a linear magnetic flux sensor based on Hall-effect technology. The sensor includes an array of horizontal plates based on TDK-Micronas' HAL technology. The Hall plate signals are first measured by two/three separate A/D-converters (one for each channel), filtered and temperature compensated. A linearization block can be used optionally to reduce the overall system non-linearity error, due to mechanical misalignment, material imperfections, etc. Offset compensation by spinning current minimizes the errors due to supply voltage and temperature variations as well as external package stress. Stray-field compensation (according to ISO11452-8) is done automatically (see Section 2.1.1), depending on the measurement configuration.

The sensor supports various measurement configurations:

– Linear measurement using the sum of two z-plates with offset, gain and non-linearity compensation,
– stray-field compensated, differential measurement using two z-plates with offset, gain and non-linearity compensation,
– linear measurement using six z-plates with offset and gain compensation.

Depending on the measurement configuration, different combination of Hall plates will be used for the magnetic field sensing.

In addition a low-power mode is available. An external ECU can send the device into low-power mode. The device wakes up periodically and provides measurement data in the active time. The sensor can as well wake up the external ECU in case that one magnetic field component exceeds a programmable threshold or if the calculated current information is passing a certain threshold. For more details please refer to Section 2.5.

The CUR 4000 is programmable by SPI interface due to which fast end-of-line programming is enabled. Detailed information can be found in the Application Note "CUR 4000 Programming Guide".

This document describes the two/three input channels and the data channel (starting after the MIX block).

2.1.1. Stray-Field Compensation

The CUR 4000 offers a measurement configuration with stray-field compensation (according to ISO11452-8). The differential signal of the two Hall-plates Z4 and Z1 is used to suppress homogeneous stray-fields. It requires the magnetic field generated by the current to be non-homogeneous. This can be achieved by using the conductor as is, without any shielding or flux concentrator.

Note: All other measurement configurations are not capable of suppressing stray-fields.

2.2. Signal Path

The following section shows the two different signal paths selectable with the CUR 4000. Both signal paths contain two kinds of registers: registers that are readout only (RAM registers) and programmable registers (non-volatile memory, EEPROM registers). The RAM registers contain magnetic field data at certain stages of the signal path and the non-volatile memory registers influence the sensor's signal processing. The orange marked EEPROM settings are individually configurable bits within an EEPROM register. The black marked sensor interfaces are located at the end of the signal paths. All registers are described in detail in the following sections.

The signal path with two channels (see Fig. 2–1) is relevant for linear or differential measurements with two Hall-plates (see SETUP_FRONTEND[3:0] in Table 3–7). The MIX block has two inputs (signal channels) and one output (data channel).

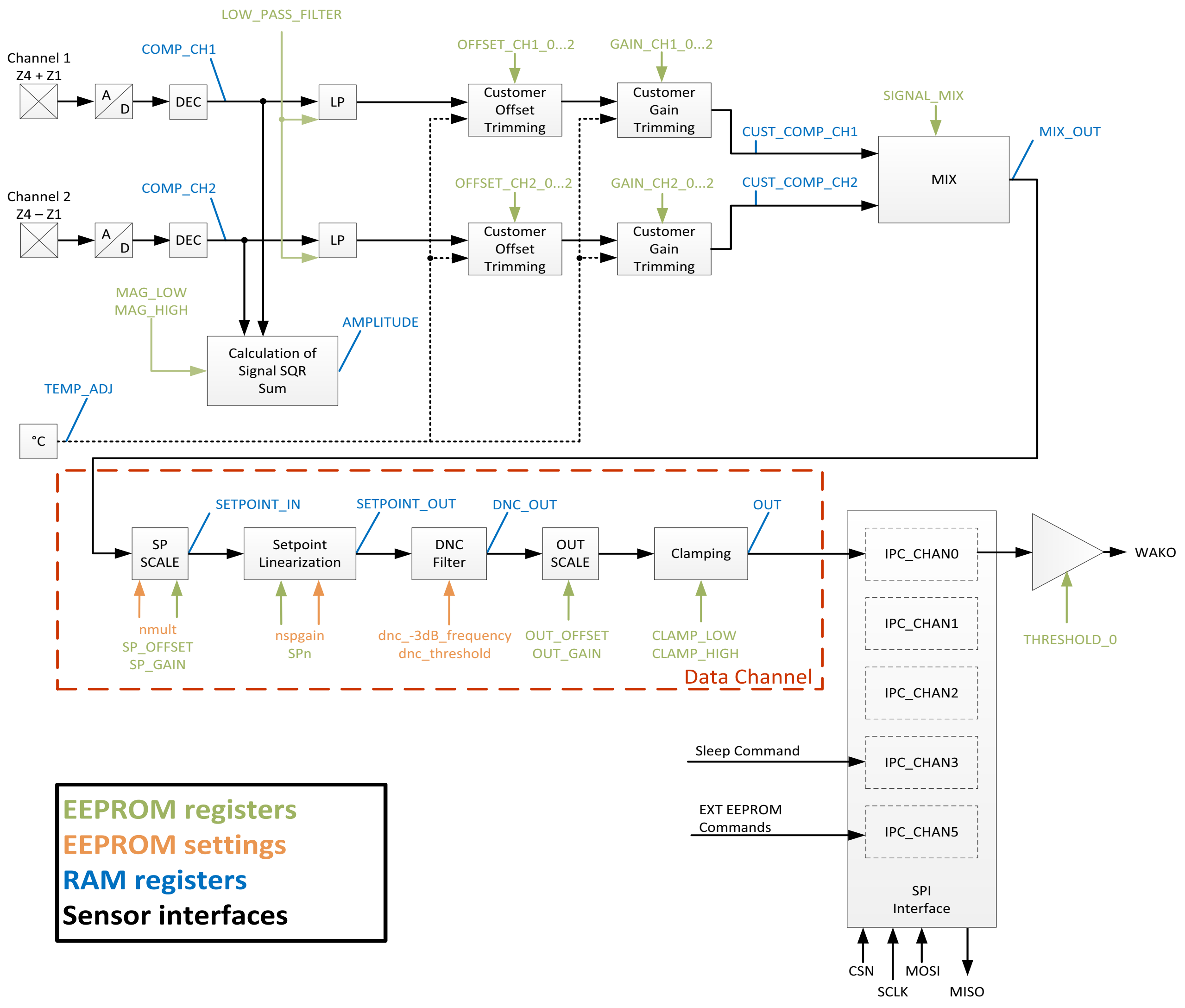

Fig. 2–1: Signal Path for 2 Z Plates

The following signal path (see Fig. 2–2) is relevant for 6Z Hall plate applications (see SETUP_FRONTEND[3:0] in Table 3–7). In this configuration three signal channels and no data channel are available.

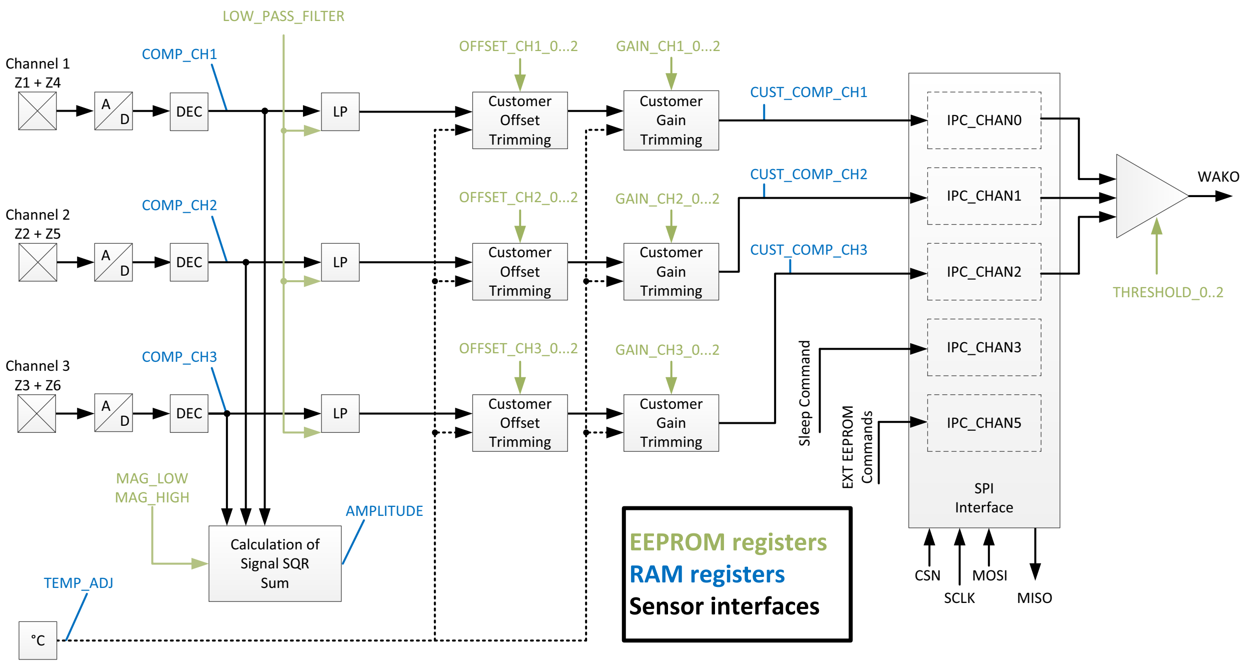

Fig. 2–2: Signal Path for 6 Z Hall Plates

The signal path and mathematical link between EEPROM parameters and RAM values for the data channel is described in more detail below.

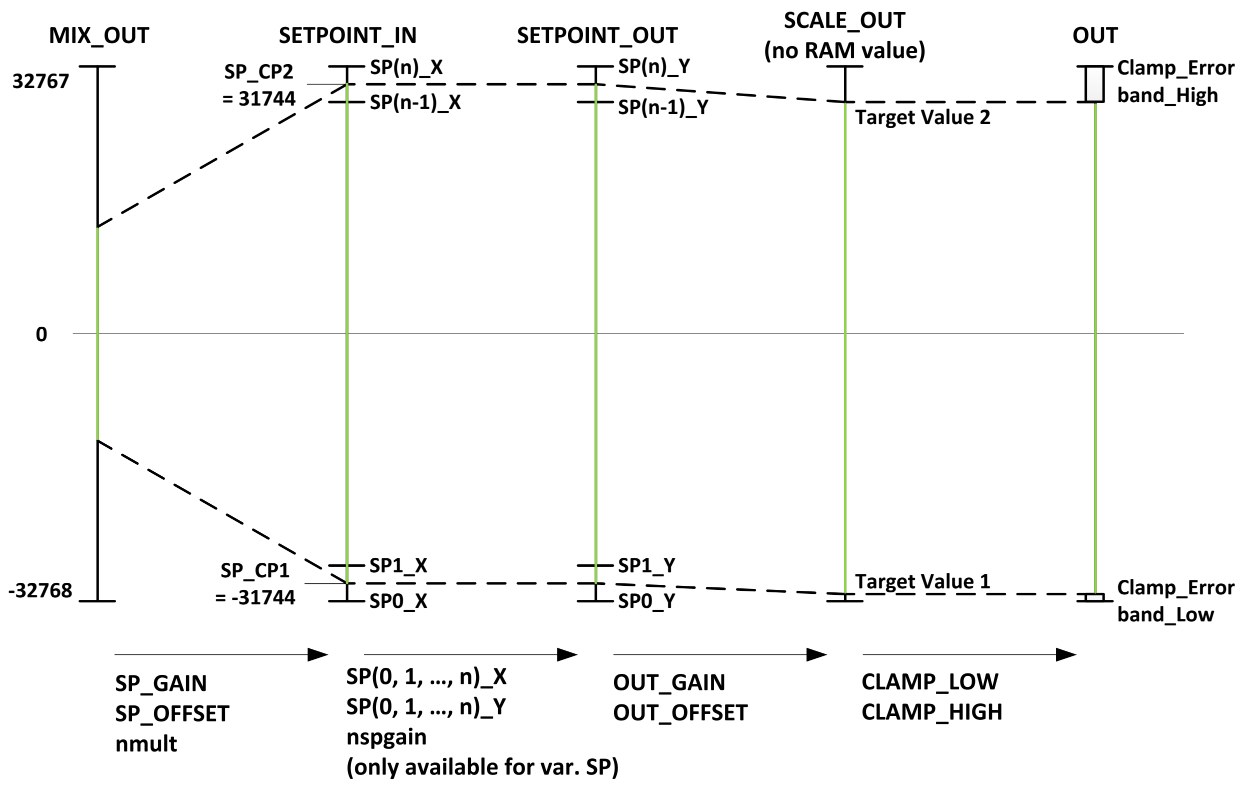

Fig. 2–3: Customer Scaling of the Magnetic Field Information along the Data Channel (Available for 2 Z Setup)

2.3. EEPROM Registers Description

The following signal path description is split up into three parts: first, the description of the EEPROM registers and settings in the following sections and subsequently, the description of the RAM registers (Section 2.4). At the end of this chapter the low-power mode (Section 2.5) is described.

Note: Terminology example:
SP_GAIN: name of the register or register value.
SP_Gain: name of the parameter.

2.3.1. MAG_LOW and MAG_HIGH

MAG_LOW and MAG_HIGH define the low and the high level for the magnetic-field range detection. The Mag_Low and Mag_High detection compares the AMPLITUDE register value with an upper and lower threshold limit defined by the registers MAG_LOW and MAG_HIGH. If either the lower or the upper limit is exceeded the sensor indicates an error and DIAG_0[4] or DIAG_0[5] is set. In application mode diagnostic bits are sent within the next output frame. The MAG_LOW limit is reached in case of a low magnetic flux. The low limit can be turned off by setting it to 0. On the other hand the MAG_HIGH limit can be used for high magnetic flux detection (over current detection). The registers MAG_LOW and MAG_HIGH have a length of 16 bits and are two's complement-coded (–32768...32767). An example of how to calculate MAG_HIGH for a given magnetic field amplitude in mT is given below (valid if no customer offset is used).

Assuming to set the over current detection threshold to 90 mT the MAG_HIGH register value is calculated as:

$$\mathrm{MAG\_HIGH} = \frac{\left(\mathrm{Mag\_High} \times 128\,\frac{1}{\mathrm{mT}}\right)^2}{32768} = \frac{\left(90\ \mathrm{mT} \times 128\,\frac{1}{\mathrm{mT}}\right)^2}{32768} = 4050 \qquad (1)$$

The accuracy of MAG_LOW and MAG_HIGH values depends mainly on the sensitivity. The variation of the sensitivity is specified in the data sheet.

Default values: MAG_LOW = 0 and MAG_HIGH = 32767.

Note: The Mag_Low and Mag_High limits have to consider the temperature drift of the setup.

2.3.2. LOW_PASS_FILTER

The Low-Pass (LP) Filter is realized as first order digital infinite impulse response (IIR) filter to reduce the sampling noise. With the LOW_PASS_FILTER register value it is possible to select different –3 dB (cutoff) frequencies for CUR 4000. The register is 16 bit organized.

The following equation and Fig. 2–4 show how a desired cutoff frequency f_c is converted into the corresponding register value.

(2)

$$\text{LOW\_PASS\_FILTER} = \left(2 - \cos\left(\frac{2 \times \pi \times f_c}{f_s}\right) - 0{,}5 \times \sqrt{14 - 16 \times \cos\left(\frac{2 \times \pi \times f_c}{f_s}\right) + 2 \times \cos\left(\frac{4 \times \pi \times f_c}{f_s}\right)}\right) \times 32768$$

f_c = cutoff frequency

f_s = sampling frequency

for example SETUP_FRONTEND[5:4] = 01: f_s = 4000 Hz

Default value of the LOW_PASS_FILTER register is 0 (low pass filter disabled). In case of LOW_PASS_FILTER = 1...32767 the filter is enabled for all channels.

Note: Only cutoff frequencies up to 0.5 * f_s are allowed.

The next equation is used to calculate the programmed cutoff frequency based on the register value.

(3)

$$f_c \approx f_s \times 0.31831 \times \arctan\left(\sqrt{2 \times \sqrt{\frac{a^2}{(a^2 - 6a + 1)^2}} - \frac{a^2 - 4a + 1}{a^2 - 6a + 1}}\right)$$

$0.171573 < a < 1$

$$a = \frac{\text{LOW\_PASS\_FILTER}}{32768}$$

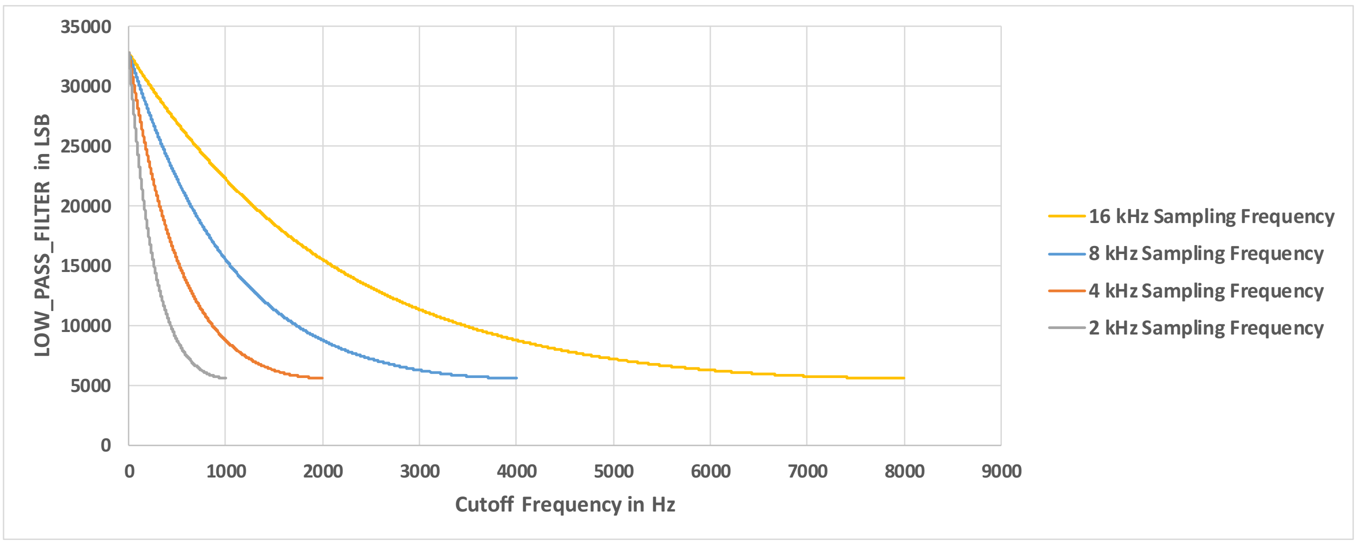

Fig. 2–4: Conversion LOW_PASS_FILTER to Cutoff Frequency

2.3.3. OFFSET_CHx_0...2

OFFSET_CH1_0...2, OFFSET_CH2_0...2 and OFFSET_CH3_0...2 support three polynomials of second order and describe the temperature compensation of the offset of channel 1, channel 2 and channel 3 (compensating a remaining offset in each of the three channels). This means a constant, linear and quadratic offset factor can be programmed individually for the three channels (temperature dependent offset). TDK-Micronas delivers pre-calibrated sensors. Nevertheless, it is possible that due to the magnetic circuit an offset in one of the three channels occurs. This can be compensated by OFFSET_CH1_0...2, OFFSET_CH2_0...2 and OFFSET_CH3_0...2. The registers have a length of 16 bits and are two's complement-coded. The temperature dependent offset is calculated with the polynomial as follows:

$$\mathrm{Offset\_CHx} = \mathrm{OFFSET\_CHx\_0} + \frac{\mathrm{TEMP\_ADJ}}{32768} \times \mathrm{OFFSET\_CHx\_1} + \left(\frac{\mathrm{TEMP\_ADJ}}{32768}\right)^2 \times \mathrm{OFFSET\_CHx\_2} \tag{4}$$

For the polynomial coefficients calculation it is necessary to know the system drift over temperature. For more details please refer to Section 4.1.

Default values:

– OFFSET_CHx_0/OFFSET_CHx_1/OFFSET_CHx_2 = 0

2.3.4. GAIN_CHx_0...2

GAIN_CH1_0...2, GAIN_CH2_0...2 and GAIN_CH3_0...2 support three polynomials of second order and describe the temperature compensation of the sensitivity of channel 1, channel 2 and channel 3 (compensating the amplitude mismatches between the channels). This means a constant, linear and quadratic gain factor can be programmed individually for the three channels (temperature dependent gain). TDK-Micronas delivers pre-calibrated sensors with compensated gain errors of channel 1, channel 2 and channel 3. Nevertheless, due to the magnetic circuit, a mismatch between the gains of the channels may occur. This can be compensated by GAIN_CH1_0...2, GAIN_CH2_0...2 and GAIN_CH3_0...2. The registers have a length of 16 bits and are two's complement-coded. The default values are:

– GAIN_CHx_0 = 8192 (100% of scaling range; 8192 as default value arises from the processing in the signal path, this is then multiplied by factor 4)

– GAIN_CHx_1/GAIN_CHx_2 = 0

$$\mathrm{Gain\_CHx} = \mathrm{GAIN\_CHx\_0} + \frac{\mathrm{TEMP\_ADJ}}{32768} \times \mathrm{GAIN\_CHx\_1} + \left(\frac{\mathrm{TEMP\_ADJ}}{32768}\right)^2 \times \mathrm{GAIN\_CHx\_2} \tag{5}$$

2.3.5. SIGNAL_MIX

The register SIGNAL_MIX selects which signal channel is used in the data channel. Channel 1, channel 2 or a combination of both can be selected. The register has a length of 16 bits and is two's complement-coded. Setting SIGNAL_MIX to 0 will select channel 1, setting it to 32767 will select channel 2.

2.3.6. nmult (EEPROM Settings)

nmult (SETUP_DATAPATH[7:5]) sets the gain exponent for the setpoint scaling block on the data channel. The factor is multiplied by SP_GAIN to achieve gain factors up to 128.

$$\mathrm{SP\_Gain} = \frac{\mathrm{SP\_GAIN}}{32768} \times 2^{\mathrm{nmult}} \tag{6}$$

2.3.7. SP_GAIN

The registers SP_GAIN scales the signal to the desired range of the subsequent Setpoint Linearization block. SP_GAIN = 32767 is the neutral setting and equates to 100% of the scaling range (–32768 equates to –100%). SP_GAIN is multiplied by two to the power of nmult to achieve gain factors up to 128 (default factor 1). The register SP_GAIN has a length of 16 bits and is two's complement-coded.

2.3.8. SP_OFFSET

The register SP_OFFSET shifts the signal to the desired range of the subsequent Setpoint Linearization block. SP_OFFSET = 0 is the neutral setting. The register has a length of 16 bits and is two's complement-coded (–32768...32767).

2.3.9. Setpoint Linearization

The Setpoint Linearization block enables the linearization of the sensor's output characteristic for the customer's application. For fixed setpoints it consists of 33 setpoints (SP0, SP1, ..., SP32). Each setpoint is defined by its fixed x position and its programmable y value. The setpoint x positions (SP(n)_X) are equally distributed between –32768...32767 LSB along the signal range. Therefore the position for each setpoint can be calculated by the following equation:

$$\begin{gathered} \mathrm{SP(n)\_X} = n \times i - 32768 \\ i = 2048;\ n \in \{0, 1, \ldots, 31\};\ \mathrm{SP32\_X} = 32767 \end{gathered} \tag{7}$$

If variable setpoints are enabled (SETUP_DATAPATH[0] = 1), both position values (x and y) of the setpoints are programmable. For more details see Section 4.4.1. The neutral x position values for the 17 programmable setpoints (except the two outer setpoints SP0_X = -32768 and SP18_X = 32767) are as follows:

Table 2–1: Neutral X Position for Variable Setpoints

1 Data Channel Register Name	1 Data Channel Setpoint Name	1 Data Channel Value
SP0	SP1_X	–29128
SP1	SP2_X	–25487
SP2	SP3_X	–21846

Table 2–1: Neutral X Position for Variable Setpoints

1 Data Channel Register Name	1 Data Channel Setpoint Name	1 Data Channel Value
SP3	SP4_X	–18205
SP4	SP5_X	–14564
SP5	SP6_X	–10923
SP6	SP7_X	–7282
SP7	SP8_X	–3641
SP8	SP9_X	0
SP9	SP10_X	3641
SP10	SP11_X	7282
SP11	SP12_X	10923
SP12	SP13_X	14564
SP13	SP14_X	18205
SP14	SP15_X	21846
SP15	SP16_X	25487
SP16	SP17_X	29128

The setpoint registers have a length of 16 bits and are two's complement-coded. Therefore the setpoint register values can vary between –32768...32767. The setpoint x values are stored absolutely and the setpoint y values are stored differentially to the corresponding x values. Thus the following has to be programmed into the setpoint registers for the SP(n)_Y register values:

$$\text{SP(n)\_Y\_Register\_Val} = \text{SP(n)\_Y} - \text{SP(n)\_X} \tag{8}$$

The setpoint y register values are initially set to 0 (neutral) by default.

The Setpoint Linearization block works in a way that the incoming signal (SETPOINT_IN value) is interpolated linearly between two adjacent setpoints (SP(n) and SP(n+1)) (see Fig. 2–5). The resulting SETPOINT_OUT register value represents the magnetic flux information after the setpoint scaling. This can also be mathematically expressed by the following equation:

$$\begin{gathered}\text{SP\_Slope\_n} = \frac{(\text{SP(n+1)\_Y} - \text{SP(n+1)\_X}) - (\text{SP(n)\_Y} - \text{SP(n)\_X})}{\text{SP(n+1)\_X} - \text{SP(n)\_X}} \\ \text{SETPOINT\_OUT} = \text{SP\_Slope\_n} \times (\text{SETPOINT\_IN} - \text{SP(n)\_X}) + \text{SP(n)\_Y} \\ \text{SP(n)\_X} \le \text{SETPOINT\_IN} < \text{SP(n+1)\_X}\end{gathered} \tag{9}$$

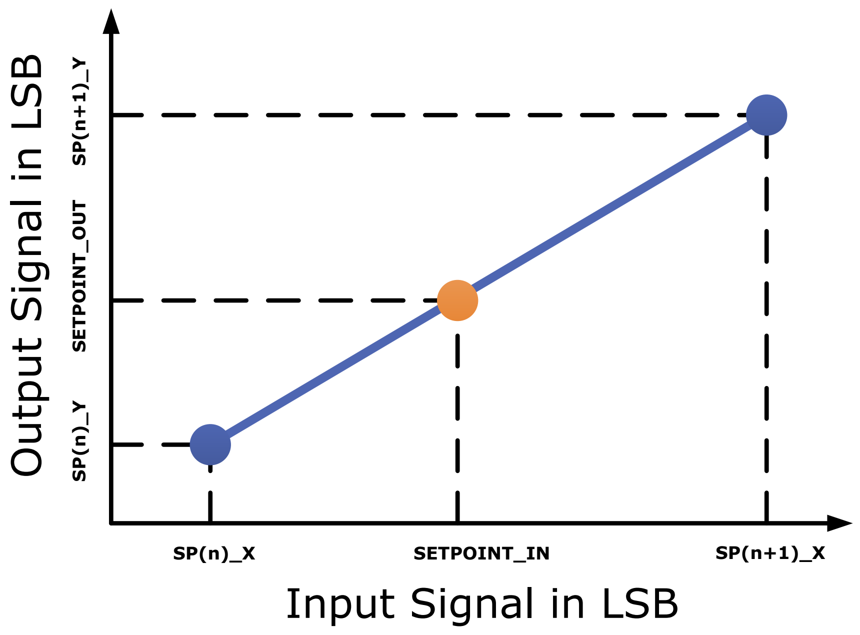

Fig. 2–5: Setpoint Scaling of the Setpoint Linearization Block

Eq. 9 can be further simplified by substitution and factorization, leading to Eq. 10:

$$\text{SETPOINT\_OUT} = \text{SP(n)\_Y} \times (1 - \alpha) + \text{SP(n+1)\_Y} \times \alpha \tag{10}$$

$$\text{with} \qquad \alpha = \frac{\text{SETPOINT\_IN - SP(n)\_X}}{\text{SP(n+1)\_X - SP(n)\_X}}$$

$$\text{for} \qquad \text{SP(n)\_X} \leq \text{SETPOINT\_IN} < \text{SP(n+1)\_X}$$

A neutral setting of the setpoints will result in an output signal being equal to the input signal (SETPOINT_OUT = SETPOINT_IN), since the input signal is being interpolated along the identity function (multiplied by 1). For fixed setpoints all setpoint y register values have to be set to 0 to achieve the neutral setting, which is the default setting for CUR 4000.

For a detailed description of the mathematical calculation for the ideal setpoints based on the measured magnetic-field information please refer to Section 4.4.

2.3.10. nspgain (EEPROM Settings)

This EEPROM setting is for variable setpoints only. The SETUP_DATAPATH[4:1] bits (= nspgain) set the gain exponent for the internal setpoint slope calculation on data channel 1/2. The EEPROM settings nspgain can be calculated as follows:

(11)

$$\text{maxSlope} = \max\left|\frac{\text{SP(n+1)\_Y} - \text{SP(n)\_Y}}{\text{SP(n+1)\_X} - \text{SP(n)\_X}}\right|$$

$$n \in \{0 \ldots 18\}$$

$$\text{nspgain} = \lceil \log 2(\text{maxSlope}) \rceil - 1$$

$$\text{nspgain} = \begin{cases} 0, & \text{nspgain} < 0 \\ \text{nspgain} & \text{else} \end{cases}$$

2.3.11. DNC Filter (EEPROM Settings)

The DNC (Dynamic Noise Cancellation) filter is a non-linear filter and can be used for further noise reduction, in addition to the first order low-pass filters after the A/D converters. It decreases the output noise significantly by adding a low-pass filter with a very low cutoff frequency for signals below a certain signal change threshold (dnc_threshold, DNC[15:8]).

The cutoff frequency dnc_–3dB_frequency of this IIR filter (Infinite Impulse Response low-pass filter of first order) can be selected by the bits DNC[7:0] of the DNC register. Both parameters have a length of eight bits.

Signals with a very low amplitude (signals classified as noise) and periodic signals with a low amplitude will be filtered whereas signals with a higher amplitude are untouched (i. e. rapid changes). The activation of the DNC filter has no impact on the resolution of the output and does not add any additional processing delay for signals.

For dnc_threshold only values from 0 to 255 are allowed. Because of the step size of 4, the maximum threshold value is 1020 LSB (= 4 x 255). The default value is 0.
Example: dnc_threshold = 5 equates to a threshold of ±20 LSB.

(12)

$$\text{dnc\_threshold} = \frac{\text{threshold value}}{4}$$

Only cutoff frequencies up to 0.5 * f_S are allowed. The default value is 0 (off). The following equation and Fig. 2–6 show how a desired cutoff frequency f_C is converted into the corresponding register value:

(13)

$$\text{dnc\_-3dB\_frequency} = \left(2 - \cos\left(\frac{2 \times \pi \times f_{c\_dnc}}{f_s}\right) - 0{,}5 \times \sqrt{14 - 16 \times \cos\left(\frac{2 \times \pi \times f_{c\_dnc}}{f_s}\right) + 2 \times \cos\left(\frac{4 \times \pi \times f_{c\_dnc}}{f_s}\right)}\right) \times \frac{32768}{128}$$

$$f_{c\_dnc} = \text{cutoff frequency dnc}$$

$$f_s = \text{sampling frequency}$$

$$\text{for example SETUP\_FRONTEND[5:4]} = 01\text{: } f_s = 4000 \text{ Hz}$$

Note: To disable the DNC filter the EEPROM setting dnc_threshold or dnc_–3dB_frequency must be programmed to 0.

The next equation is used to calculate the programmed cutoff frequency based on the register value.

$$f_{c\_dnc} \approx f_s \times 0.31831 \times \arctan\left(\sqrt{2 \times \sqrt{\frac{a^2}{(a^2 - 6a + 1)^2}} - \frac{a^2 - 4a + 1}{a^2 - 6a + 1}}\right) \quad (14)$$

$$0.171573 < a < 1$$

$$a = \frac{dnc\_\text{-}3dB\_frequency \times 128}{32768}$$

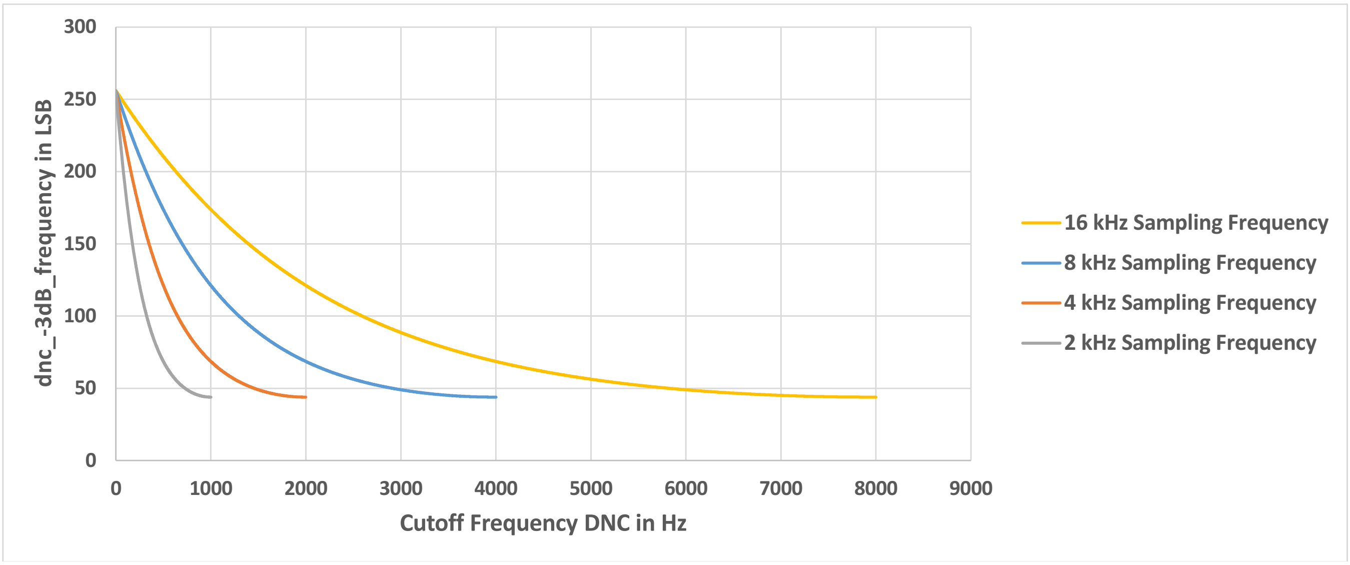

Fig. 2–6: Conversion dnc_-3dB_frequency to Cutoff Frequency

2.3.12. OUT_OFFSET and OUT_GAIN

The final customer programmable scaling blocks are used to scale the signal (data channel) according to the desired output signal ranges (see Fig. 2–3). This is realized by the EEPROM registers OUT_OFFSET and OUT_GAIN. The registers have a length of 16 bits and are two's complement-coded. Default value for OUT_OFFSET is 0 and default value for OUT_GAIN is 16384. In case of OUT_GAIN is smaller than 0 the register can also be used to invert the output signal. The mathematical functionality of these scaling blocks are described in Section 2.4.8. For a detailed description of the calculation of OUT_GAIN and OUT_OFFSET please refer to Section 4.2.3.

2.3.13. CLAMP_LOW

The clamping level CLAMP_LOW establishes the minimum value of the data channel output signal. The register has a length of 16 bits, is two’s complement-coded and has a scaling range from -32767 (0%) to 32767 (+100%).
For example the output signal shall clamp at Clamp_Low 10% full-scale:

$$\text{CLAMP\_LOW} = -32767 + \text{Clamp\_Low} \times 65536 = -32767 + 0.1 \times 65536 \approx -26214 \tag{15}$$

Note: If the result of the data channel is below the expected valid range the clmp_lo (DIAG_0[2]) bit is set to 1.

Note: Due to functional safety reasons it is not recommended to use CLAMP_LOW = -32768.

2.3.14. CLAMP_HIGH

The clamping level CLAMP_HIGH establishes the maximum value of the data channel output signal. The register has a length of 16 bits, is two’s complement-coded and has a scaling range from -32767 (0%) to 32767 (+100%).
For example the output signal shall clamp at Clamp_High 90% full-scale:

$$\text{CLAMP\_HIGH} = -32767 + \text{Clamp\_High} \times 65536 = -32767 + 0.9 \times 65536 \approx 26215 \tag{16}$$

Note: If the result of the data channel is above the expected valid range the clmp_high (DIAG_0[3]) bit is set to 1.

Note: Due to functional safety reasons it is not recommended to use CLAMP_HIGH = -32768.

2.4. RAM Registers Description

The following part describes the RAM registers of the signal path.

2.4.1. TEMP_ADJ

The TEMP_ADJ register contains the adjusted digital value of the sensor junction temperature. It has a length of 16 bits and is binary coded. From the 16 bits only the range between 0...32767 is used for the temperature information. Typically the temperature sensor is calibrated in the way that the register value is 150 LSB at –40 °C and 18000 LSB at 160 °C.

The temperature is converted into °C as follows:

$$T = \frac{160\ °C - (-40\ °C)}{18000 - 150} \times TEMP\_ADJ - 41{,}68\ °C \tag{17}$$

The temperature is converted into LSB as follows:

$$TEMP\_ADJ = \frac{89.25}{°C} \times T + 3720 \tag{18}$$

The accuracy of the temperature is approximately ±5 °C.

2.4.2. COMP_CH1, COMP_CH2 and COMP_CH3

COMP_CH1, COMP_CH2 and COMP_CH3 registers contain the magnetic field information of channel 1, channel 2 and channel 3 after temperature-drift compensation of the Hall plates. Internally the sensor converts the magnetic field information into digital values. To obtain the digital values the magnetic values are multiplied by the sensitivity of the Hall plates. Typically the sensitivity is 128 LSB/mT.

$$COMP\_CHx = B \times 128\ \frac{1}{mT} \tag{19}$$

B = component of the magnetic field (depending on measurement configuration)

Fig. 2–7: Digital Values in Channel 1 and Channel 2

These digital values are corresponding to the registers COMP_CH1 and COMP_CH2 of the signal path (see Fig. 2–1). COMP_CH3 is only available if a measurement configuration including channel 3 is selected (see Fig. 2–2).

2.4.3. AMPLITUDE

AMPLITUDE contains the sum of squares of TDK-Micronas compensated channel 1, channel 2 and channel 3.

$$AMPLITUDE = \frac{COMP\_CH1^2}{32768} + \frac{COMP\_CH2^2}{32768} + \frac{COMP\_CH3^2}{32768} \tag{20}$$

2.4.4. CUST_COMP_CHx

CUST_COMP_CH1, CUST_COMP_CH2 and CUST_COMP_CH3 contain the customer compensated magnetic field information of data channel 1, 2 and 3. These registers contain already the customer gain and offset corrected data. The calculation of the input signals CUST_COMP_CHx for the data channel are shown below (Eq. 5 and Eq. 4 are needed here):

$$\begin{gathered} \text{Lpf\_CHx} = \text{COMP\_CHx} \times (1 - k) + k \times \text{Lpf\_CHx\_old} \\ \text{CUST\_COMP\_CHx} = 4 \times (\text{Lpf\_CHx} + \text{Offset\_CHx}) \times \frac{\text{Gain\_CHx}}{32768} \\ k = \frac{\text{LOW\_PASS\_FILTER}}{32768} \\ \text{Lpf\_CHx\_old} = \text{tapped Lpf\_CHx value} \end{gathered} \tag{21}$$

2.4.5. MIX_OUT

MIX_OUT contains the digital value of the mixed signal channels. The signal mixing is described by the following formula:

$$\text{MIX\_OUT} = \left(1 - \frac{\text{SIGNAL\_MIX}}{32768}\right) \times \text{CUST\_COMP\_CH1} + \left(\frac{\text{SIGNAL\_MIX}}{32768}\right) \times \text{CUST\_COMP\_CH2} \tag{22}$$

2.4.6. SETPOINT_IN and SETPOINT_OUT

The SETPOINT_IN register contains the gain and offset corrected magnetic flux information on the data channel. The SETPOINT_OUT register contains the magnetic flux information after the setpoint adjustment on the data channel. Accordingly SETPOINT_IN is the input and SETPOINT_OUT is the result of the Setpoint Linearization block. For SETPOINT_IN the MIX_OUT is scaled with SP_GAIN, nmult and SP_OFFSET as follows:

$$\text{SETPOINT\_IN} = \frac{\text{SP\_GAIN} \times 2^{\text{nmult}}}{32768} + \text{SP\_OFFSET} \tag{23}$$

For the calculation of SETPOINT_OUT please refer to Section 2.3.9.

2.4.7. DNC_OUT

The DNC_OUT register contains the magnetic flux information of the data channel after dynamic noise cancellation.

$$\begin{gathered}
\text{Dnc\_Out\_New} = \text{SETPOINT\_OUT} \\
\text{Dnc\_Out\_T} = \text{Dnc\_Out\_New} \times (1-k) + k \times \text{Dnc\_Out\_Old} \\
k = \frac{\text{dnc\_-3dB\_frequency}}{32768} \\
\text{Dnc\_Out\_Old} = \text{tapped Dnc\_Out\_T value} \\
\text{DNC\_OUT} = \begin{cases} \text{Dnc\_Out\_T} & \text{for abs(Dnc\_Out\_New - Dnc\_Out\_Old)} \leq \text{dnc\_threshold} \\ \text{Dnc\_Out\_New} & \text{else} \end{cases}
\end{gathered} \tag{24}$$

2.4.8. OUT

The output of the OUT SCALE block represents the position information after final scaling with the constant factors OUT_OFFSET and OUT_GAIN and is calculated as follows:

$$\text{SCALE\_OUT} = 2 \times \text{DNC\_OUT} \times \frac{\text{OUT\_GAIN}}{32768} + \text{OUT\_OFFSET} \tag{25}$$

The OUT register contains the magnetic flux information of the data channel after clamping. Eq. 25 is needed for the following equation:

$$\text{OUT} = \begin{cases} \text{CLAMP\_LOW} & \text{for SCALE\_OUT} < \text{CLAMP\_LOW} \\ \text{CLAMP\_HIGH} & \text{for SCALE\_OUT} > \text{CLAMP\_HIGH} \\ \text{SCALE\_OUT} & \text{else} \end{cases} \tag{26}$$

2.4.9. IPC_CHAN0

The IPC_CHAN0 register contains the value of the OUT register, in case of the 2Z measurement configuration. It contains the value of the CUST_COMP_CH1 register, in case of the 6Z measurement configuration.

2.4.10. IPC_CHAN1

The IPC_CHAN1 register contains the value of the CUST_COMP_CH2 register, in case of the 6Z measurement configuration.

2.4.11. IPC_CHAN2

The IPC_CHAN2 register contains the value of the CUST_COMP_CH3 register, in case of the 6Z measurement configuration.

2.4.12. IPC_CHAN3

The IPC_CHAN3 register is used for the send to sleep command via SPI Interface (see Section 2.5).

Note: The above mentioned RAM registers can be read in programming mode. For normal application mode respectively in the running application only IPC_CHAN0...3 registers must be used. Only those registers are secured with CRCs and error reporting (DIAG_0...1 registers are only updated while reading the IPC_CHAN0...3 registers). Furthermore, the NEW flag is only visible in IPC_CHAN0...2 (not visible when reading COMP_CH1...3). See Table 3–6 for more details about the available data.

2.4.13. IPC_CHAN5

The IPC_CHAN5 register is used for general sensor command via SPI Interface (see Section 3.2).

2.5. Low-power Mode

The CUR 4000 is featuring two kinds of operational modes. A so called application mode and a low-power mode. In application mode the sensor is continuously capturing the magnetic field generated from a current flowing through a busbar and an ECU can poll the measured information. In low-power mode the sensor is in sleep mode, it wakes up for a brief time to capture the magnetic field generated from a current flowing through a busbar. During sleep mode the current consumption of the device is significantly reduced. The low-power mode offers different possibilities for configuration, like periodically wake-up by internal timer, wake-up due to current or magnetic field change, etc. All different low-power mode configurations are described in Section 2.5.4.

2.5.1. STANDBY_SLEEP_TIME

The periodical wake-up is user configurable. STANDBY_SLEEP_TIME defines the wake-up timer (sleep period) of CUR 4000. The register has a length of 16 bits with only 8 significant bits:

$$\text{sleep period} = (\text{STANDBY\_SLEEP\_TIME} + 1) \times 2\ \text{ms} \tag{27}$$

2.5.2. Thresholds for Low-power Mode

The THRESHOLD_0...2 registers define the thresholds for the three different wake-up sources in low-power mode. The sensor compares its measurement data in the active mode of the low-power mode with these thresholds. In case that at least one threshold is exceeded the sensor will notify the external ECU via the WAKI/O pin 6.

The table below shows the link between THRESHOLD_0...2 registers and the signal sources. The available source depends on the selected measurement configuration as well.

Table 2–2: Sources of THRESHOLD_0...2 registers

THRESHOLD_0...2	IPC Channel	Signal Source
THRESHOLD_0	IPC_CHAN0	OUT (2Z) or CUST_COMP_CH1 (6Z)
THRESHOLD_1	IPC_CHAN1	CUST_COMP_CH2 (6Z)
THRESHOLD_2	IPC_CHAN2	CUST_COMP_CH3 (6Z)

2.5.3. Send to Sleep Command

To enter the low-power mode the sensor can be sent to sleep mode by using the ‘send to sleep’ command. For this, the value 0xA55A has to be written into IPC_CHAN3 register.

Note: The command only works in application mode and not in programming mode.

2.5.4. Modes for Power Consumption Reduction

CUR 4000 supports five different modes for power consumption reduction. Those five low-power modes are split into a sleep phase with very low current consumption and an active phase in which the device performs defined measurement tasks. By setting dedicated EEPROM bits the customer or the ECU can select between the different modes (see Table 3–8).

The following table describes the different use cases:

Table 2–3: Overview of Low-power Mode Use Cases

Use Case	ECU Mode	Sensor Tasks	ECU Tasks	Configuration of SETUP_STANDBY
1	Control status of the sensor (sleep phase or active phase)	Check status of WAKI/O and start measurements after wake-up	Wake-up sensor by WAKI/O pin. Poll SPI read until NEW bit is set. Send sensor to sleep phase by ‘send to sleep’ command	ext_wakeup = 01/10/11 All other bits are set to 0

2	Always active and sensor is periodically in sleep phase	Wake-up by internal sleep period. Indicate start of active phase to ECU on WAKI/O pin	Poll SPI read for NEW bit after indication of active phase on WAKI/O pin. Send sensor to sleep phase by ‘send to sleep’ command	cnt_wakeup = 1
				wakout = 1
				All other bits are set to 0

3	Operate in low-power mode until wake-up by the sensor via WAKI/O pin	Wake-up by internal sleep period. Indicate start of active phase to ECU on WAKI/O pin	Poll SPI read for NEW bit after indication of active phase on WAKI/O pin. Send sensor to sleep phase by ‘send to sleep’ command	cnt_wakeup = 1
				wakout = 1
				All other bits are set to 0

4		Wake-up by internal sleep period. Compare measured value with a defined threshold and wake-up ECU by WAKI/O pin if threshold condition is fulfilled, else go back to sleep phase	Poll SPI read for NEW bit after wake-up by the sensor. Send sensor to sleep phase by ‘send to sleep’ command	cnt_wakeup = 1
				wakout = 1
				thrd_x = 01/10
				All other bits are set to 0

5	Operate in low-power mode until wake-up by the sensor via WAKI/O pin or actively wake-up the sensor	Wake-up by internal sleep period (like use case 3 and 4) or wake-up by external trigger on WAKI/O pin	Wake-up sensor by WAKI/O pin or wait for wake-up by the sensor. Poll SPI read for NEW bit. Send sensor to sleep phase by ‘send to sleep’ command	cnt_wakeup = 1
				ext_wakeup = 01/10/11
				wakout = 1
				thrd_x = 01/10

Note: To wake up the sensor by the ECU in Use Case 5, it is mandatory that the ECU is generating minimum two edges on the WAKI/O pin. Otherwise, it might happen that the sensor is missing the wake-up signal from the ECU.

For more detailed information, timing diagrams, and additional required external circuits for each use case, please refer to the CUR 4000 data sheet.

3. Register Access

3.1. Memory Table

3.1.1. Register Overview

Note: The CUR 4000 has 80 EEPROM registers and 16 EXT EEPROM registers (extended EEPROM registers). All registers are writable and readable but all registers excluding the EEPROM registers will be overwritten internally after power-on-reset. Each register has a range from 0x8000 to 0x7FFF and is two's complement-coded if nothing else is mentioned in the explanation column.

Table 3–1: CUR 4000 EEPROM Memory Table

Address	Register Name	Explanation	Neutral Value
0x00	SETUP_DATAPATH	Sensor datapath settings (see Table 3–6)	0x0000
0x01	OFFSET_CH1_0	Offset correction for channel 1: constant coefficient	0x0000
0x02	OFFSET_CH1_1	Offset correction for channel 1: linear coefficient	0x0000
0x03	OFFSET_CH1_2	Offset correction for channel 1: quadratic coefficient	0x0000
0x04	GAIN_CH1_0	Gain correction for channel 1: constant coefficient	0x2000
0x05	GAIN_CH1_1	Gain correction for channel 1: linear coefficient	0x0000
0x06	GAIN_CH1_2	Gain correction for channel 1: quadratic coefficient	0x0000
0x07	OFFSET_CH2_0	Offset correction for channel 2: constant coefficient	0x0000
0x08	OFFSET_CH2_1	Offset correction for channel 2: linear coefficient	0x0000
0x09	OFFSET_CH2_2	Offset correction for channel 2: quadratic coefficient	0x0000
0x0A	GAIN_CH2_0	Gain correction for channel 2: constant coefficient	0x2000
0x0B	GAIN_CH2_1	Gain correction for channel 2: linear coefficient	0x0000
0x0C	GAIN_CH2_2	Gain correction for channel 2: quadratic coefficient	0x0000
0x0D	OFFSET_CH3_0	Offset correction for channel 3: constant coefficient	0x0000
0x0E	OFFSET_CH3_1	Offset correction for channel 3: linear coefficient	0x0000
0x0F	OFFSET_CH3_2	Offset correction for channel 3: quadratic coefficient	0x0000
0x10	GAIN_CH3_0	Gain correction for channel 3: constant coefficient	0x2000
0x11	GAIN_CH3_1	Gain correction for channel 3: linear coefficient	0x0000
0x12	GAIN_CH3_2	Gain correction for channel 3: quadratic coefficient	0x0000
0x13	MAG_LOW	Magnetic compare level low Range: 0x0000 to 0x7FFF	0x0000
0x14	MAG_HIGH	Magnetic compare level high Range: 0x0000 to 0x7FFF	0x7FFF
0x15	STANDBY_SLEEP_TIME	Wake-up timer (sleep period)	0x0000
0x18	SP_OFFSET	Output offset correction for optimal performance of the data channel	0x0000
0x19	SP_GAIN	Output gain correction for optimal performance of the data channel	0x7FFF
0x1C	SIGNAL_MIX	Signal selection in 2Z measurement configuration	0x0000

Table 3–1: CUR 4000 EEPROM Memory Table, continued

Address	Register Name	Explanation	Neutral Value
0x1D	LOW_PASS_FILTER	LP-Filter Coefficient:–3 dB frequency 0: Filter off 1...32767: Filter on (for all input channels) (see Section 2.3.2)	0x0000
0x22	DNC	Dynamic noise cancellation filter DNC[15:8]: threshold (TH) = threshold value in LSB divided by 4 0: DNC off 1: ±4 LSB ... 255: ±1020 LSB DNC[7:0]: –3 dB frequency 0: DNC off 1...255: DNC on (see Section 2.3.11)	0x0 0x0
0x23-0x46	SP0...SP35	(see Table 3–2)	-
0x47	OUT_OFFSET	Output offset correction for the data channel	0x0000
0x48	OUT_GAIN	Output gain correction for the data channel	0x4000
0x49	CLAMP_LOW	Clamping level low	0x8001
0x4A	CLAMP_HIGH	Clamping level high	0x7FFF
0x4B	THRESHOLD_0	Threshold for Channel/Signal 0 (IPC_CHAN0)	0x0000
0x4C	THRESHOLD_1	Threshold for Channel/Signal 1 (IPC_CHAN1)	0x0000
0x4F	THRESHOLD_2	Threshold for Channel/Signal 2 (IPC_CHAN2)	0x0000
All unlisted register addresses are reserved (must be set to 0).			

Table 3–2: CUR 4000 EEPROM Setpoints Memory Table

Address	Register Name	Fixed Setpoints	Variable Setpoints
0x23	SP0	SP0_Y (diff.)	SP1_X
0x24	SP1	SP1_Y (diff.)	SP2_X
0x25	SP2	SP2_Y (diff.)	SP3_X
0x26	SP3	SP3_Y (diff.)	SP4_X
0x27	SP4	SP4_Y (diff.)	SP5_X
0x28	SP5	SP5_Y (diff.)	SP6_X
0x29	SP6	SP6_Y (diff.)	SP7_X
0x2A	SP7	SP7_Y (diff.)	SP8_X
0x2B	SP8	SP8_Y (diff.)	SP9_X
0x2C	SP9	SP9_Y (diff.)	SP10_X
0x2D	SP10	SP10_Y (diff.)	SP11_X
0x2E	SP11	SP11_Y (diff.)	SP12_X
0x2F	SP12	SP12_Y (diff.)	SP13_X
0x30	SP13	SP13_Y (diff.)	SP14_X
0x31	SP14	SP14_Y (diff.)	SP15_X
0x32	SP15	SP15_Y (diff.)	SP16_X
0x33	SP16	SP16_Y (diff.)	SP17_X

Table 3–2: CUR 4000 EEPROM Setpoints Memory Table, continued

Address	Register Name	Fixed Setpoints	Variable Setpoints
0x34	SP17	SP17_Y (diff.)	SP0_Y (diff.) (X = –32768)
0x35	SP18	SP18_Y (diff.)	SP1_Y (diff.)
0x36	SP19	SP19_Y (diff.)	SP2_Y (diff.)
0x37	SP20	SP20_Y (diff.)	SP3_Y (diff.)
0x38	SP21	SP21_Y (diff.)	SP4_Y (diff.)
0x39	SP22	SP22_Y (diff.)	SP5_Y (diff.)
0x3A	SP23	SP23_Y (diff.)	SP6_Y (diff.)
0x3B	SP24	SP24_Y (diff.)	SP7_Y (diff.)
0x3C	SP25	SP25_Y (diff.)	SP8_Y (diff.)
0x3D	SP26	SP26_Y (diff.)	SP9_Y (diff.)
0x3E	SP27	SP27_Y (diff.)	SP10_Y (diff.)
0x3F	SP28	SP28_Y (diff.)	SP11_Y (diff.)
0x40	SP29	SP29_Y (diff.)	SP12_Y (diff.)
0x41	SP30	SP30_Y (diff.)	SP13_Y (diff.)
0x42	SP31	SP31_Y (diff.)	SP14_Y (diff.)
0x43	SP32	SP32_Y (diff.)	SP15_Y (diff.)
0x44	SP33	unused	SP16_Y (diff.)
0x45	SP34	unused	SP17_Y (diff.)
0x46	SP35	unused	SP18_Y (diff.) (X = 32767)
SP(n)_Y (diff.) = y setpoint is stored differentially to the respective x value.			

Table 3–3: CUR 4000 EXT EEPROM Memory Table (Extended EEPROM with Separate Addresses)

Address	Register Name	Explanation	Neutral Value
0x00	CUSTOMER_ID0	Customer specific content	0x0000
0x01	CUSTOMER_ID1	Customer specific content	0x0000
0x02	CUSTOMER_ID2	Customer specific content	0x0000
0x03	CUSTOMER_ID3	Customer specific content	0x0000
0x04	CUSTOMER_ID4	Customer specific content	0x0000
0x05	CUSTOMER_ID5	Customer specific content	0x0000
0x06	CUSTOMER_ID6	Customer specific content	0x0000
0x07	CUSTOMER_ID7	Customer specific content	0x0000
0x08	CUSTOMER_ID8	Customer specific content	0x0000
0x09	CUSTOMER_ID9	Customer specific content	0x0000
0x0A	SETUP_FRONTEND	Sensor frontend settings (see Table 3–7)	0x002D
0x0B	SETUP_STANDBY	Sensor frontend settings (see Table 3–8)	0x0000
0x0E	SUPPLY_SUPERVISION	Supply supervision (see Table 3–9)	0x0000
0x0F	CUSTOMER_CHECKSUM	16 bit CCITT checksum over all customer settings (byte-inverted)	0x8F1D

Table 3–4: CUR 4000 RAM Memory Table

Address	Register Name	Explanation
0x50	TEMP_ADJ	Adjusted temperature value Range: 0x0064 to 0x2EE0
0x51	COMP_CH1	TDK-Micronas temperature compensated magnetic-field value on channel 1
0x52	COMP_CH2	TDK-Micronas temperature compensated magnetic-field value on channel 2
0x53	COMP_CH3	TDK-Micronas temperature compensated magnetic-field value on channel 3
0x56	CUST_COMP_CH1	Channel 1 signal after customer compensation
0x57	CUST_COMP_CH2	Channel 2 signal after customer compensation
0x58	CUST_COMP_CH3	Channel 3 signal after customer compensation
0x59	MIX_OUT	Calculated input of the data channel
0x5F	SETPOINT_IN	Gain and offset corrected magnetic flux on the data channel
0x60	SETPOINT_OUT	Magnetic flux after setpoint adjustment on the data channel
0x61	DNC_OUT	Magnetic flux after dynamic noise cancellation on primary data channel
0x66	AMPLITUDE	Sum of squares of TDK-Micronas compensated magnetic-field values COMP_CH1, COMP_CH2 and COMP_CH3
0x67	OUT	Data channel output after clamping
0x6E	PROG_BUFFER0	Primary communication buffer for programming
0x6F	PROG_BUFFER1	Secondary communication buffer for programming

Table 3–5: CUR 4000 Hardware Register Memory Table

Address	Register Name	Explanation
0x70	IPC_CHAN0	Inter-processor data channel 0 OUT (2Z) or CUST_COMP_CH1 (6Z) In case of an error the register value could be overwritten for reasons of functional safety.
0x71	IPC_CHAN1	Inter-processor data channel 1 CUST_COMP_CH2 (6Z)
0x72	IPC_CHAN2	Inter-processor data channel 2 CUST_COMP_CH3 (6Z)
0x73	IPC_CHAN3	Inter-processor data channel 3 Used for send to sleep command (0xA55A) via SPI
0x75	IPC_CHAN5	Inter-processor data channel 5 8051 communication register (for EEPROM memory access and programming or RAM register read)
0x7D	DIAG_0	Diagnosis 0 (see Table 3–10) Diagnosis bits, which do not directly evoke actions
0x7E	DIAG_1	Diagnosis 1 (see Table 3–11) Diagnosis bits, which directly evoke actions
0x7F	HW_ID	Hardware ID base
All unlisted register addresses are reserved.		

3.1.2. Customer Configuration Register

SETUP_DATAPATH, SETUP_FRONTEND, SETUP_STANDBY and SUPPLY_SUPERVISION registers are 16 bit registers that enable the customer to activate various functions of the sensor. The tables below describe the available combinations and resulting functions in detail.

Table 3–6: SETUP_DATAPATH Register

Bit no.	Name	Description
15:8	-	Reserved (must be set to 0)
7:5	nmult	Gain exponent for SETPOINT_IN: SP_Gain = SP_GAIN $\times$ 2^{nmult}
4:1	nspgain	Gain exponent for setpoint slope of primary data channel: Slope = SPGain $\times$ $2^{nspgain+1}$
0	variable_ setpoints	Fixed/variable setpoints: 0: fixed setpoints 1: variable setpoints

Table 3–7: SETUP_FRONTEND Register

Bit no.	Name	Description
15	customer_lock	Customer lock: 0: unlocked 1: locked
14:9	-	Reserved (must be set to 0)

Table 3–7: SETUP_FRONTEND Register

Bit no.	Name	Description
8	cluster	0: IPC_CHAN0 to IPC_CHAN2 are independent 1: IPC_CHAN0 to IPC_CHAN2 are updated after IPC_CHAN0 is read
7:6	-	Reserved (must be set to 0)
5:4	fdecsel	Select decimation frequency: 00: 2 kHz 01: 4 kHz 10: 8 kHz 11: 16 kHz
3:0	-	Reserved

Table 3–8: SETUP_STANDBY Register

Bit no.	Name	Description
15:10	-	Reserved (must be set to 0)
9	wakout	WAKI/O pin 6 as wake output: 0: disabled 1: enabled Note: used with internal counter wakeup. Wake ECU via this pin if desired.
8	cnt_wakeup	Internal wakeup by sleep period: 0: disabled 1: enabled Note: the standby phase is enabled if either the internal counter or the external wakeup or both are enabled.
7:6	ext_wakeup	External wakeup by WAKI/O pin 6: 00: disabled 01: rising edge 10: falling edge 11: both edges
5:4	threshold_2	Threshold mode 2: 00: no threshold 01: wake ECU if signal is above THRESHOLD_2 10: wake ECU if signal is below THRESHOLD_2 11: reserved
3:2	threshold_1	Threshold mode 1: 00: no threshold 01: wake ECU if signal is above THRESHOLD_1 10: wake ECU if signal is below THRESHOLD_1 11: reserved
1:0	threshold_0	Threshold mode 0: 00: no threshold 01: wake ECU if signal is above THRESHOLD_0 10: wake ECU if signal is below THRESHOLD_0 11: reserved

Table 3–9: SUPPLY_SUPERVISION Register

Bit no.	Name	Description
15:8	ov_level	Over voltage threshold Supervision of external power supply, directly defined in mV (1 LSB = 100 mV) Example: ov_level = 0b1000000 --> ov_level = 6.4 V
7:0	uv_level	Under voltage threshold Supervision of external power supply, directly defined in mV (1 LSB = 100 mV) Example: uv_level = 0b100000 --> uv_level = 3.2 V

Table 3–10: DIAG_0 Register

Bit no.	Name	Description when bit is set to 1
15	dsp_self_check	DSP self check routines (redundancy or plausibility checks) failed
14	eeprom_checksum	DSP and µC check for 16 bit checksum covering the EEPROM parameter failed
13	rom_ram_checksum	DSP checksum for ROM and RAM failed
12	temp_out_of_range	Measured chip junction temperature outside of –40 to 170 °C range
11	temp_sens_redundancy	Plausibility check against each other of two redundant temperature sensor elements failed
10	hvdd_overvoltage	Internal Hall plate supply is too high
9	overtemp	Hardware overtemperature supervision: junction temperature > 180 °C
8	-	Reserved
7	adc_stuck	One of the three A/D converters delivers a stuck signal
6	clipping_dec_filter	Overflow/underflow of a decimation filter has been detected
5	mag_hi	The magnetic field is above the programmed upper threshold MAG_HIGH register value (magnetic field too high)
4	mag_lo	The magnetic field is below the programmed lower threshold MAG_LOW register value (magnetic field too low)
3	clmp_hi	The result of the magnetic flux calculation (high) is out of the expected (valid) range
2	clmp_lo	The result of the magnetic flux calculation (low) is out of the expected (valid) range
1	hall_current	The monitoring for the correct current flowing of the Hall plates failed
0	-	Reserved

Table 3–11: DIAG_1 Register

Bit no.	Name	Description when bit is set to 1
15	-	Reserved
14	gpadc_err	General purpose ADC error
13	-	Reserved
12	vsbg_err	The bandgap voltage of the standby domain is out of range
11	xvdd_undervoltage	External supply voltage is too low (undervoltage error); only visible in application mode
10	xvdd_overvoltage	External supply voltage is too high (overvoltage error); only visible in application mode
9	avdd_range	Internal analog voltage is out of the expected (valid) range
8	dvdd_range	Internal digital voltage is out of the expected (valid) range
7:0	-	Cannot be read out. These bits immediately trigger a device reset.

3.2. Sensor Commands

The CUR 4000 supports internal programming commands (IPCs) by writing one of the following sensor commands as data bits to the register IPC_CHAN5.

Start Listen Mode (0x22A2)

The first step to communicate with the sensor is to set the sensor into listen mode by writing the data 0x22A2 to the IPC_CHAN5 register. After 4 ms the sensor switches and remains in listen mode for max. 110 ms, during which the sensor can be switched to programming mode. After max. 110 ms without receiving the start programming mode command the sensor will go into reset.

Start Programming Mode (0x2EAE)

In programming mode it is possible to change the register settings of the sensor. This means for writing data into the shadow EEPROM it is necessary to set the sensor in the programming mode. By writing the data 0x2EAE to the IPC_CHAN5 register when the sensor is in listen mode (for details see application note “CUR 4000 Programming Guide”), the sensor will be set into the programming mode. The sensor will remain in programming mode until the next power-down. After power-up the sensor is always operating in application mode.

Note: The following sensor commands only work in programming mode and not in application mode.

Read Micronas ID (0x0B)

After writing this command to IPC_CHAN5 the 16 bit MIC ID1 can be read in PROG_BUFFER0 and the 16 bit MIC ID2 can be read in PROG_BUFFER1.

Read product ID and release ID (0xE03F)

After writing this command to IPC_CHAN5 the 16 bit release ID can be read in PROG_BUFFER0 and the 16 bit product ID can be read in PROG_BUFFER1.

Download EEPROM (0x02)

After the command Download EEPROM the data is copied from the EEPROM to the corresponding shadow EEPROM registers.

Generate Reset (0x06)

The command Generate Reset triggers a full reset. It is intended for scenarios, where an external Power-on-Reset is not desired, e.g. in bus configurations.

Program EEPROM

For storing a specified part (from start to end register number, 7 bit each) of the shadow EEPROM to the permanent EEPROM the command Program EEPROM is necessary. The execution status of the command can be checked in the registers PROG_BUFFER0.

In register PROG_BUFFER0 a countdown is set. The number of registers which should be stored with the command defines the countdown. The countdown decreases with every stored register. If there is any problem during the storing of the registers and the programming is stopped, the remaining countdown can be read in the PROG_BUFFER0 register, so that it is possible to check which registers are written and which are not. In this case the Program EEPROM sequence shall be repeated.

To ensure that the programming of the EEPROM was successful, the registers IPC5 and PROG_BUFFER0 shall be read afterwards.

Note: The EXT EEPROM register numbers follow directly on the EEPROM register numbers. This means the total register number is 96. The EEPROM register numbers start at 0x00 and end at 0x4F. The EXT EEPROM register numbers start at 0x50 and end at 0x5F.

For more details of the command please refer to Table 3–12.

Read EXT EEPROM (0xD070 to 0xD07F)

Read EXT EEPROM is a command to access and read the EXT EEPROM registers (word addresses between 0x0 and 0xF). The desired address is added to the command 0xD070. This gives the final sensor command (see Table 3–12). After the command has been sent to IPC_CHAN5 register the requested 16 bit register data can be read in PROG_BUFFER0.

Write EXT EEPROM (0xF070 to 0xF07F)

Write EXT EEPROM is a command to access and write the EXT EEPROM registers (word addresses between 0x0 and 0xF). The desired address is added to the command 0xF070. This gives the final sensor command (see Table 3–12). To use the command correctly the desired data must first be written into the register PROG_BUFFER0. After writing data to the sensor a power-on-reset is necessary to reload the sensor with the programmed settings.

Table 3–12: Commands on IPC_CHAN5 Register

Command	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start Programming Mode	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0
Read Micronas ID	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Read Product and Release ID	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1
Download EEPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Generate Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Program EEPROM	0	1	end register address							start register address						
Read EXT EEPROM	1	1	0	1	0	0	0	0	0	1	1	1	4 bit address			
Write EXT EEPROM	1	1	1	1	0	0	0	0	0	1	1	1	4 bit address			

Examples for the Read EXT EEPROM, the Write EXT EEPROM and the Program EEPROM commands are given in the following flowcharts:

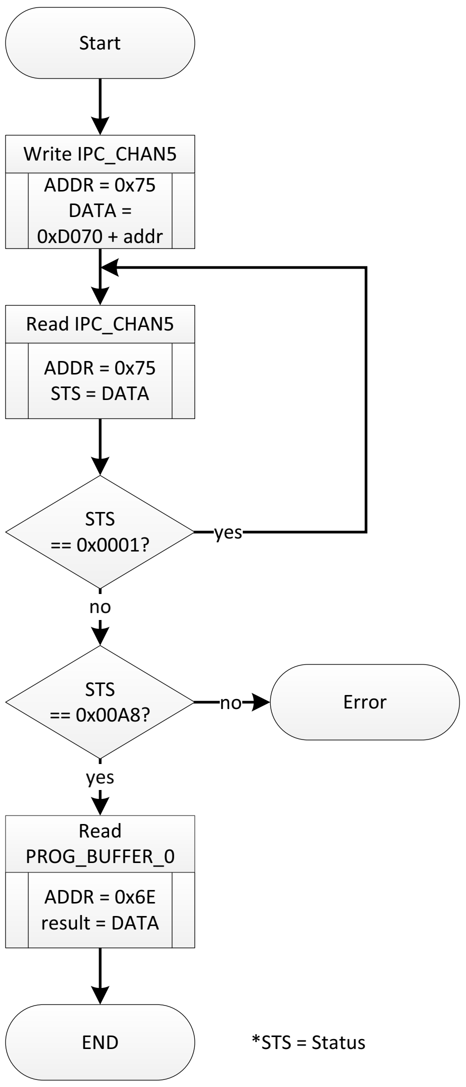

Fig. 3–12: Read Address 'addr' using the Read EXT EEPROM Command

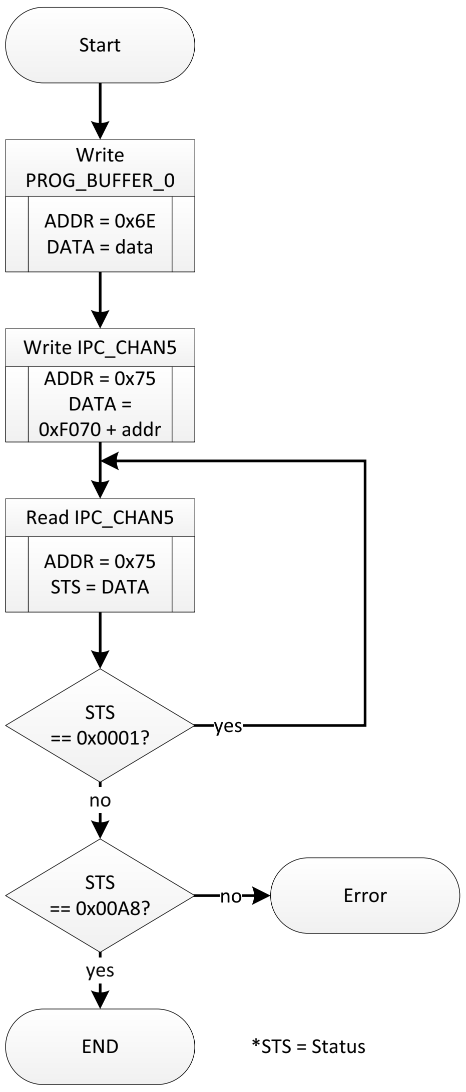

Fig. 3–13: Write Data ‘data’ to Address ‘addr’ using the Write EXT EEPROM Command

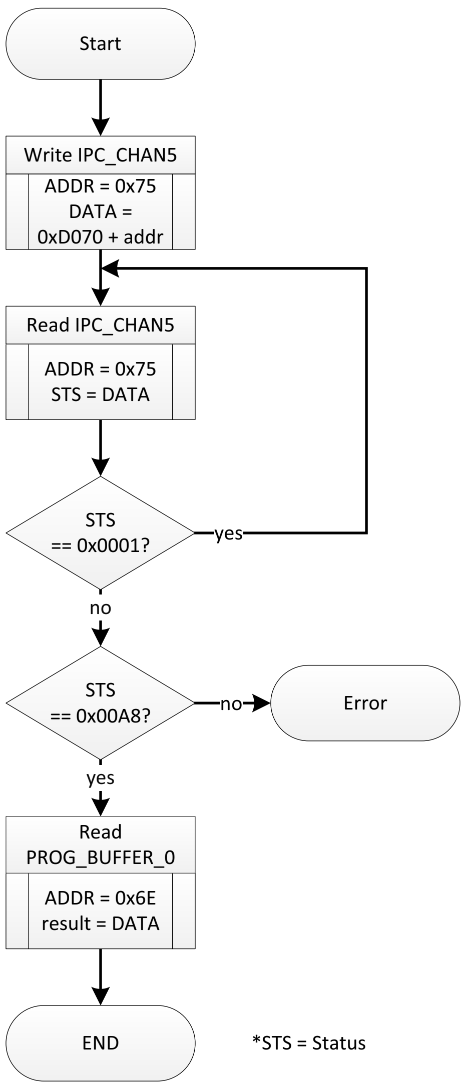

Fig. 3–14: Program EEPROM using the Program EEPROM Command

3.3. Status and Error Codes

Status and error information is given in the PROG_BUFFER0 and IPC_CHAN5 registers. Generally, IPC_CHAN5 indicates the status and error codes of commands executed by the sensor to give an universal register to be read out by the customer. The two high bytes are set to 0x00 when the command is known and accepted by the sensor. If the two high bytes are not equal to 0x00 an error has occurred and the low bytes must be ignored (see Table 3–14). The two low bytes of IPC_CHAN5 contain the information about the status of the executed command: 0xA8 is written to IPC_CHAN5 low bytes in case of a successful executed command (´success´ in Table 3–13), the standard status flag 0x01 is written to IPC_CHAN5 low bytes when the command starts and the sensor is busy, and 0x02 is written to IPC_CHAN5 low bytes when an error has occurred.

An overview of the detailed status and error codes in the two high bytes of IPC_CHAN5 is given in Table 3–14 and the content of PROG_BUFFER0 and PROG_BUFFER1 after IPC_CHAN5 programming in Table 3–13.

Table 3–13: Content of PROG_BUFFER0, PROG_BUFFER1 and IPC_CHAN5 after IPC_CHAN5 Programming

Sensor Command	PROG_BUFFER0	PROG_BUFFER1	IPC_CHAN5
Program EEPROM	0/Countdown status at error	-	success/error
Start programming mode	-	-	success
Read EXT EEPROM	16 bit result	-	success
Write EXT EEPROM	-	-	success
Read Micronas ID	16 bit result (MIC ID1)	16 bit result (MIC ID2)	success/error
Read product and release ID	16 bit result (Release ID1)	16 bit result (Product ID2)	success/error
Download EEPROM	-	-	success
Generate reset	-	-	-

Table 3–14: IPC5 Content Codes at Register IPC_CHAN5

Status code	Description
0x00A8	No error (‘success’ in Table 3–13)
0x0001	Sensor is busy executing the last command
0x00XX	Error of unspecified source
0x01XX	Command is locked
0x02XX	Invalid command
0x03XX	Register number out of range (start register number > end register number or end register number > 95)
0x04XX	Mismatch content
0x05XX	Time-out (hardware failure)

3.4. Customer Checksum Calculation

The sensor has the capability to store a 16 bit checksum over all customer settings (address 0x00 to 0x5E). The checksum is checked by the sensor during startup and normal operation. A checksum error is indicated in DIAG_0[14]. If the sensor is locked, a detected checksum error directly evokes a reset.

The customer checksum has to be calculated manually from address 0x00 to 0x5E (only EEPROM registers and EXT EEPROM registers, RAM registers are ignored). The EXT EEPROM register numbers follow directly on the EEPROM register numbers. This means the total register number is 95. The EEPROM register numbers start at 0x00 and end at 0x4F. The EXT EEPROM register numbers start at 0x50 and end at 0x5E. The CRC calculation result has to be written in register CUSTOMER_CHECKSUM.

The polynomial for the CRC calculation is $X^{16} + X^{12} + X^{5} + 1$ (CRC16_CCITTC_FALSE), with a seed value of 0xFFFF.

An example of the customer checksum calculation is shown below:

```
const uint16_t CHECKSUM_POLY = 0x1021U; // x^16+x^12+x^5+1
uint16_t crc16_CCITT (uint16_t *data, uint8_t length)
{
    uint8_t i, bit;
    uint16_t crc, temp_byte_swap;

    crc = 0xFFFFU;
    for (i = 0; i < length; i++)
    {
        temp_byte_swap = ((data[i] >> 8) & 0xFFU) + ((data[i] & 0xFFU) << 8);
        crc = crc ^ temp_byte_swap;

        for (bit = 0; bit < 16; bit++)
        {
            if (crc & 0x8000U)
            {
                crc = crc << 1;
                crc = crc ^ CHECKSUM_POLY;
            }
            else
                crc = crc << 1;
        }
    }
    temp_byte_swap = ((crc >> 8) & 0xFFU) + ((crc & 0xFFU) << 8);
    return temp_byte_swap;
}
```

Note: The EEPROM register numbers start at 0x00 and end at 0x4F. The EXT EEPROM register numbers start at 0x50 and end at 0x5E.

3.5. Locking the Sensor

For reliability in service, it is mandatory to set the customer lock bit (SETUP_FRONTEND[15]) to 1 after final adjustment and final EEPROM programming.

Note: - After the customer lock is activated (by writing and power-on-reset), it is not possible to program the sensor anymore.
- Because the lock bit is also included in the CRC, the CRC has to be calculated considering the lock bit.

To lock the sensor the customer checksum has to be calculated and stored as described in Section 3.4. Refer to this section and the following flowchart for more details.

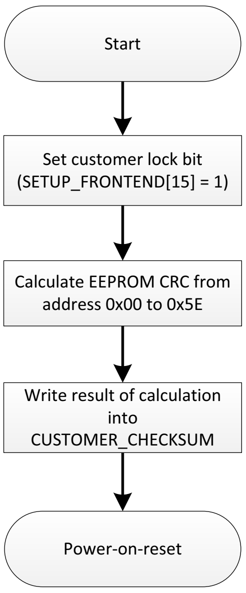
Start
Set customer lock bit (SETUP_FRONTEND[15] = 1)
Calculate EEPROM CRC from address 0x00 to 0x5E
Write result of calculation into CUSTOMER_CHECKSUM
Power-on-reset

Fig. 3–15: Locking the Sensor

The successful setting of the customer lock bit shall be checked, e.g. by reading back the customer lock bit after programming and downloading it to the shadow RAM, but before a power-on-reset. The lock mechanism becomes only active after the next power-on-reset.

Electro-static discharges (ESD) may disturb the supply voltage during programming. Please take precautions against ESD.

A programming tool including hardware and software can be provided for product evaluation and application development. It is recommended to use the TDK-Micronas' tool kit to simplify the product development phase.

The CUR 4000 allows to read the registers after locking, provided that valid commands with correct CRC are recognized by the sensor. The programming command does not have any effect on the memory of locked devices.

4. Calibration and Linearization

4.1. Customer Offset and Gain Calculation at RT

The programmable customer offset and gain can be used to improve the performance of the sensor. The IC itself is already calibrated by TDK-Micronas, but effects on module level need to be calibrated by the customer.

Note: CUR 4000 can be configured in different measurement configurations. Depending on the required measurement task one of the measurement configurations can be selected. The SETUP_FRONTEND[3:0] bits are defining the different available configurations (which Hall plates/channels are active).

4.1.1. Offset Compensation

The offset for the application can be determined by reading the COMP_CHx registers at zero primary current (Ip = 0). The target is that all signals, become zero at Ip = 0. Therefore the offset value that compensates the error can be calculated as follows:

$$\text{OFFSET\_CHx\_0} = -\text{COMP\_CHx} \tag{28}$$

This is a simple method to calculate the offset. Other methods can be for example a linear or nonlinear fit to the magnetic field components.

4.1.2. Gain Compensation

The gain for the application can be determined by reading the COMP_CHx registers at two different current (Ip_1, Ip_2). The target is to achieve the desired gain for the application.

Note: Increasing the gain has no influence on the resolution or accuracy. All data sheet values are referred to the default gain of 128 LSB/mT

With the desired gain and the measured gain, the GAIN_CHx register can be calculated as follows:

$$\text{GAIN\_CHx\_0} = 8192 \times \frac{\text{desired gain}}{\text{measured gain}} \tag{29}$$

4.2. Customer Offset and Gain Calculation over Temperature

The customer offset and gain polynomials can be used to compensate for module level temperature dependencies.

Note: This procedure is also valid for all temperature dependent parameters like OFFSET_CHx, Gain_CHx while in the following section only the OFFSET_CHx is described for shortness.

The compensation for a temperature dependent drift of the offset requires an adaptation of the higher order customer offset. The calculation of the necessary parameters (OFFSET_CHx_0...2) is described in the next section and the flow-chart in Fig. 4–17.

4.2.1. Nonlinear Temperature Compensation

For the nonlinear temperature compensation, at least three different temperatures and the corresponding signals are necessary.

Basis for the calculation of the polynomial coefficients for a nonlinear temperature drift compensation are the register values TEMP_ADJ and COMP_CH1 or COMP_CH2 (or COMP_CH3 in case channel 3 is active). The purpose is to measure the offset change over temperature. As the offset can change independently for each active channel (1, 2 and 3) it is recommended to measure all active channels.

Note that in this equation it is assumed that COMP_CH1 exhibits negative and positive values. The following calculations have to be done to determine the polynomial coefficients of the temperature dependent offset:

1. Step:
Measure the Offset value and the TEMP_ADJ value at different temperatures. The TEMP_ADJ value can also be calculated from the applied temperature using Eq. 18, but it is recommended to read the value from the IC for better offset compensation.

$$\begin{gathered}\text{Offset(T) = Offset at measured temperature T}\\ \text{TEMP\_ADJ(T) = TEMP\_ADJ at measured temperature T}\end{gathered} \tag{30}$$

2. Step:
Calculation of a polynomial of second order by using a Least-Square polynomial fit with TEMP_ADJ as independent and Offset as dependent variables. This will result in the polynomial coefficients (a0, a1 and a2) for the temperature dependent offset:

$$\text{Offset(TEMP\_ADJ)} = a0 + a1 \times \text{TEMP\_ADJ} + a2 \times \text{TEMP\_ADJ}^2 \tag{31}$$

The corresponding register values OFFSET_CHx_0, OFFSET_CHx_1 and OFFSET_CHx_2 can be obtained by using the following equation:

(32)

$$\mathrm{OFFSET\_CHx\_0} = \mathrm{a0}$$
$$\mathrm{OFFSET\_CHx\_1} = \mathrm{a1} \times 32768$$
$$\mathrm{OFFSET\_CHx\_2} = \mathrm{a2} \times 32768^2$$

4.2.2. Linear Temperature Compensation

When the temperature coefficient of the module is known and a linear temperature compensation is sufficient, a temperature dependent multiplication factor for the desired offset can be calculated as follows:

(33)

$$\mathrm{MF(T)} = \mathrm{T_C} \times (\mathrm{T} - \mathrm{RT}) + 1$$
$$\mathrm{MF(T)} = \text{multiplication factor for the desired offset}$$
$$\mathrm{T_C} = \text{temperature coefficient of the module in } \frac{\mathrm{ppm}}{^\circ\mathrm{C}}$$

The corresponding adjusted temperature sensor value can be read from the sensor at the desired temperature or the ambient temperature can be converted from °C into LSB. An example is shown in Section 2.4.1.

For at least two temperatures the adjusted temperature sensor value and the corresponding multiplication factor for the desired offset need to be calculated.

The corresponding temperature dependent offset is obtained by multiplying the desired offset at room temperature by the calculated multiplication factor MF(T). In this case Eq. 4 of Section 2 becomes a linear equation (OFFSET_CHx_2 = 0) and no fit is necessary to calculate the coefficients.

Example:

The linear temperature coefficient of the module is –0.1 %/°C and the desired offset is 3000 LSB. For this calculation two temperatures are sufficient. In this example the temperatures 25 °C and 150 °C are used.

Table 4–15: Example of a Linear Offset Coefficient Calculation

	T in °C	TEMP_ADJ in LSB	Offset in LSB
T1	25	5951	3000
T2	150	17108	2625

The coefficients can be calculated as follows:

(34)

$$\text{Offset} = \text{a1} \times \text{TEMP\_ADJ} + \text{a0}$$

$$\text{a1} = \frac{\text{Offset(T2)} - \text{Offset(T1)}}{\text{TEMP\_ADJ(T2)} - \text{TEMP\_ADJ(T1)}} = \frac{2625 - 3000}{17108 - 5951} = -0.0336$$

$$\text{a0} = \text{Offset(T1)} - \text{TEMP\_ADJ(T1)} \times \text{a1} = 3000 - 5951 \times (-0.0336) = 3200$$

$$\text{Offset} = \text{temperature dependent offset}$$

$$\text{a2} = 0$$

$$\text{a1} = \text{linear compensation coefficient}$$

$$\text{a0} = \text{constant compensation coefficient}$$

The corresponding register values OFFSET_CHx_0, OFFSET_CHx_1 and OFFSET_CHx_2 can be obtained by equation Eq. 32.

4.2.3. SP_GAIN and SP_OFFSET, OUT_GAIN and OUT_OFFSET

Depending on the type of linearization it is recommended to scale SETPOINT_IN in a way that:

– fixed setpoints: the first calibration point is scaled between setpoint 0 and setpoint 1 (SP_CP1) and the second calibration point is scaled between setpoint 31 and setpoint 32 (SP_CP2).

– variable setpoints: the first calibration point is scaled between -32768 and the x value of setpoint 1 (SP_CP1) and the second calibration point is scaled between the x value of setpoint 17 and 32767 (SP_CP2).

Note: For fixed setpoints it is recommended to set SP_CP1 = -31744 and SP_CP2 = 31744 to use the full scaling range.

The final scaling to the desired output range e.g. from Target Value 1 = -32752 to Target Value 2 = 32640 is then realized by OUT_GAIN and OUT_OFFSET (see Eq. 38). SP_Gain, SP_GAIN, SP_OFFSET, OUT_GAIN and OUT_OFFSET are calculated as follows:

(35)

$$\text{SP\_Gain} = \frac{\text{SP\_CP2} - \text{SP\_CP1}}{\text{MIX\_OUT}_{\text{pos2}} - \text{MIX\_OUT}_{\text{pos1}}}$$

$$\text{if } |\text{SP\_Gain}| > 1 \text{ then nmult} = \text{round up } (\log_2|\text{SP\_Gain}|)$$

$$\text{else nmult\_1} = 0$$

$$\text{SP\_GAIN} = 32768 \times \frac{\text{SP\_Gain}}{2^{\text{nmult}}}$$

$$\text{SP\_OFFSET} = \text{SP\_CP1} \ - \text{MIX\_OUT}_{\text{pos1}} \times \text{SP\_Gain}$$

with:

$$\text{SP\_CP1} = \text{desired setpoint value at first Calibration Point (CP1)}$$

$$\text{SP\_CP2} = \text{desired setpoint value at second Calibration Point (CP2)}$$

$$\text{MIX\_OUT}_{\text{pos1}} = \text{MIX\_OUT at CP1}$$

$$\text{MIX\_OUT}_{\text{pos2}} = \text{MIX\_OUT at CP2}$$

For the case the system has a too strong offset, it is possible that the values for the SP_CP1 and SP_CP2 have to be decreased. The following check has to be done after calculating SP_Gain:

(36)

$$-32768 < (\text{MIX\_OUT}_{\text{pos1}} \times \text{SP\_Gain}) < 32768$$

$$-32768 < (\text{MIX\_OUT}_{\text{pos2}} \times \text{SP\_Gain}) < 32768$$

If these checks in Eq. 36 are in range equation Eq. 37 and Eq. 38 can be skipped.

If one of these calculation is out of range the decreased SP_CP values will be calculated as follows:

(37)

$$\text{If } \mathrm{MIX\_OUT}_{pos1} \times \mathrm{SP\_Gain} \text{ is out of range:}$$

$$\mathrm{SP\_CP2} = \text{round down}\left(32768 \times \frac{\mathrm{MIX\_OUT}_{pos1} - \mathrm{MIX\_OUT}_{pos2}}{2 \times \mathrm{MIX\_OUT}_{pos1}}\right)$$

$$\mathrm{SP\_CP1} = \mathrm{SP\_CP2} \times -1$$

$$\text{If } \mathrm{MIX\_OUT}_{pos2} \times \mathrm{SP\_Gain} \text{ is out of range:}$$

$$\mathrm{SP\_CP2} = \text{round down}\left(32768 \times \frac{\mathrm{MIX\_OUT}_{pos2} - \mathrm{MIX\_OUT}_{pos1}}{2 \times \mathrm{MIX\_OUT}_{pos2}}\right)$$

$$\mathrm{SP\_CP1} = \mathrm{SP\_CP2} \times -1$$

With the decreased SP_CP values the parameter SP_Gain, SP_GAIN and SP_OFFSET can be calculated as described in equation 34.

(38)

$$\mathrm{OUT\_GAIN} = 16384 \times \frac{\text{Target Value 2} - \text{Target Value 1}}{\mathrm{SP\_CP2} - \mathrm{SP\_CP1}}$$

$$\mathrm{OUT\_OFFSET} = \text{Target Value 1} - \frac{\mathrm{OUT\_GAIN}}{16384} \times \mathrm{SP\_CP1}$$

4.3. Calibration with the Digital Output

The calibration settings of SP_GAIN and nmult, SP_OFFSET, OUT_GAIN and OUT_OFFSET can be calculated from the output signal of the sensor. With this method the necessary values for the calibration and linearization can be gathered faster compared to reading the MIX_OUT register at several positions. For that purpose the parameters are first programmed to defined initial values, here denoted with the subscript ini. The following values can be used:

- $\text{SP\_GAIN}_{\text{ini}} = 27853$
- $\text{SP\_OFFSET}_{\text{ini}} = 0$
- $\text{OUT\_GAIN}_{\text{ini}} = 16384$
- $\text{OUT\_OFFSET}_{\text{ini}} = 0$
- $\text{nmult}_{\text{ini}} = 0$
- The setpoints must be programmed to neutral settings (SPx = 0).

After the defined initial values are programmed, the output at the first and the second calibration point has to be measured. The first and the last measurement points are used to calculate new SP_GAIN, nmult, SP_OFFSET, OUT_GAIN and OUT_OFFSET values (see Fig. 4–20).

It is also necessary to know if the output is increasing or decreasing from the start to the end position. This can be checked by analyzing e.g. the first ten samples.

With the SETPOINT_IN value the corresponding $\text{MIX\_OUT}_{\text{calc}}$ value is calculated as follows:

$$\text{SP\_Gain} = \frac{\text{SP\_GAINx}_{\text{ini}}}{32768} \times 2^{\text{nmult}_{\text{ini}}} = \frac{27853}{32768} \times 2^{0} = 0{,}85 \tag{39}$$

$$\text{SP\_OFFSET}_{\text{ini}} = 0$$

$$\text{MIX\_OUT}_{\text{calc,i}} = \frac{\text{SETPOINT\_IN}}{\text{SP\_Gain}}$$

$$\text{MIX\_OUT}_{\text{calc,i}} = \text{i}^{\text{th}} \text{ calculated MIX\_OUT}$$

$$\text{i} \in \{1, 2\}$$

With $\text{MIX\_OUT}_{\text{calc,1}}$ and $\text{MIX\_OUT}_{\text{calc,2}}$ the new values for SP_GAIN and nmult, SP_OFFSET, OUT_GAIN and OUT_OFFSET are calculated (see Section 4.2.3).

4.4. Linearization

The CUR 4000 features 33 programmable setpoints for one channel with fixed setpoints. The setpoints can be used to linearize the output. It depends on the signal whether a setpoint linearization can further improve the linearity of the sensor's output signal. In case a setpoint linearization is desired, a readout of the SETPOINT_IN register value or the output signal of the sensor with neutral setpoint values over the full calibration range is necessary. These values can be used to calculate the ideal setpoints for linearizing the signal. The procedure flowchart in Fig. 4–19 illustrates the sequence of the necessary steps for performing a linearization.

Note: The calibration procedure must be done before the linearization.

4.4.1. Measurement of the Nonlinear Information

After a successful two-point calibration an additional setpoint linearization can be performed. The readout of the SETPOINT_IN register value (or the sensor output with neutral setpoint values) over the full calibration range provides the input to the Setpoint Linearization block of the signal path. These values represent the nonlinear signal profile of the calibration range, which needs to be linearized. The following three points have to be considered for the measurement:

1. For a proper calculation it is necessary to match the first and the last measurement point with the first and the second calibration point used during the two-point calibration.
2. The total number of measurement points can be chosen arbitrarily. More measurement points provide more information on the nonlinearity of the calculated current and enable a potentially more accurate linearization, although it will require more time.
3. The measurement intervals do not have to be equidistant. In this case the system position of the measurement point shall be saved in relation to the calibration range. In case of equidistant measurement positions a consecutive numbering of the measurement points is sufficient.

4.4.2. Linearization and Calibration with the Output of the Sensor in One Step

The calculation of SP_GAIN and nmult, SP_OFFSET, OUT_GAIN, OUT_OFFSET and the linearization parameters can be done in one step without a previous two-point calibration. For that purpose the parameters are firstly programmed to defined initial values. The following values can be used:

– $SP\_GAIN_{ini} = 27853$
– $SP\_OFFSET_{ini} = 0$
– $OUT\_GAIN_{ini} = 16384$
– $OUT\_OFFSET_{ini} = 0$
– $nmult_{ini} = 0$
– The setpoints must be programmed to neutral settings (SPx = 0).

After the defined initial values are programmed, the measurement range should be measured. The measurements can be done either by applying current at each measurement point or, in a faster way, by measuring continuously the sensor's output while the current is increased at a constant rate.

After the measurement, N measurement points are available. The first and the last measurement points are used to calculate the new SP_GAIN and nmult, SP_OFFSET, OUT_GAIN and OUT_OFFSET values as described in Section 4.3.

First with the SETPOINT_IN values the corresponding MIX_OUT values for i = 1, 2...N (N = number of measurements) can be calculated with Eq. 39.

With $MIX\_OUT_{calc,1}$ and $MIX\_OUT_{calc,N}$ the new values for SP_GAIN and nmult, SP_OFFSET, OUT_GAIN and OUT_OFFSET are calculated (see Section 4.2.3).

The new values are denoted as SP_GAIN_{new}, $nmult_{new}$, SP_OFFSET_{new}, OUT_GAIN_{new} and OUT_OFFSET_{new}.

The following target values can be used for the SP_GAIN_{new}, $nmult_{new}$ and SP_OFFSET_{new} calculation.

The target values for the SETPOINT_IN values are SP_CP1 = -31744 and SP_CP2 = 31744.

All measured values must be rescaled with the SP_GAIN_{new}, $nmult_{new}$, SP_OFFSET_{new}, OUT_GAIN_{new} and OUT_OFFSET_{new}.

The input for the linearization block is $SETPOINT\_IN_{new}$ (SP_IN_{new}) which is calculated as follows:

$$SP\_IN_{new,\,i} = MIX\_OUT_{new,\,i} \times \frac{SP\_GAIN_{new} \times 2^{nmult_{new}}}{32768} + SP\_OFFSET_{new} \qquad i \in \{1, 2, \ldots, N\} \tag{40}$$

The procedure flowchart in Fig. 4–20 illustrates the sequence of the necessary steps to perform a linearization with the sensor output.

4.4.3. Calculation of the Setpoints (for Fixed Setpoints)

After the measurement over the full calibration range N measurement points (SETPOINT_IN values) are available. The Setpoint Linearization block interpolates the incoming signal (SETPOINT_IN value) linearly between two adjacent setpoints. With the 33 setpoints there are 32 setpoint intervals.

- Interval 1: $-32768 \leq$ SETPOINT_IN ≤ -30721
- Interval 2: $-32720 \leq$ SETPOINT_IN ≤ -28673
- Interval 3: $-28672 \leq$ SETPOINT_IN ≤ -26625
- ...

For a calculation of the ideal setpoints it is necessary to have at least one measurement value in each setpoint interval. In case of a reduced number of measurements the missing values can be calculated by interpolating e.g. linearly between two measurement points.

Since a linear characteristic between the two calibration points is desired, for every measurement point a corresponding SETPOINT_OUT target value on the linear interconnection between the two calibration points can be calculated using the following equation:

$$\text{SETPOINT\_OUT}(i) = \text{SP\_CP1} + \frac{\text{Pos}(i) - \text{CP1}}{\text{CP2} - \text{CP1}} \times (\text{SP\_CP2} - \text{SP\_CP1}) \tag{41}$$

$$i \in \{1, 2, \ldots, N\}$$

$$\text{Pos}(i) = i^{\text{th}} \text{ position in A}$$

$$\text{CP1} = \text{position of first Calibration Point in A}$$

$$\text{CP2} = \text{position of second Calibration Point in A}$$

This equation considers the position of the respective current measurement point in relation to the calibration range.

In case of equidistant measurement points, Eq. 41 simplifies to:

$$\text{SETPOINT\_OUT}(i) = \text{SP\_CP1} + \frac{i - 1}{N - 1} \times (\text{SP\_CP2} - \text{SP\_CP1}) \tag{42}$$

$$i \in \{1, 2, \ldots, N\}$$

A least squares fit is used to solve a number of equations for a number of unknown variables. In this case the unknown variables are the setpoints. Each equation represents a measurement or interpolation point. The other condition is that there are at least as many equations as unknown variables.

The equations for the linearization are the following:

(43)

$$\begin{gathered}
\mathrm{SP0\_Y} \times (1-\alpha_1) + \mathrm{SP1\_Y} \times \alpha_1 = \mathrm{SETPOINT\_OUT(1)} \\
\dots \\
\mathrm{SP(n\text{-}1)\_Y} \times (1-\alpha_N) + \mathrm{SP(n)\_Y} \times \alpha_N = \mathrm{SETPOINT\_OUT(N)} \\
\text{with } \alpha_i = \frac{\mathrm{SETPOINT\_IN(i)\ \text{-}\ dX}}{\mathrm{dX}} \\
\mathrm{SP(n\text{-}1)\_X} \le \mathrm{SETPOINT\_IN(i)} < \mathrm{SP(n)\_X} \\
\mathrm{dX} = \mathrm{SP(n)\_X} - \mathrm{SP(n\text{-}1)\_X} \\
i \in \{1, 2, \dots, N\} \text{ data points} \\
\mathrm{dX} = 2048;\ n \in \{1, 2, \dots, 32\} \text{ setpoint intervals}
\end{gathered}$$

This can be written in N-by-33-matrix (respectively two N-by-17-matrices) form as:

(44)

$$\begin{gathered}
\begin{bmatrix}
(1-\alpha_1) & \alpha_1 & 0 & \dots & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots \\
0 & \dots & \dots & 0 & (1-\alpha_N) & \alpha_N
\end{bmatrix}
\times
\begin{bmatrix}
\mathrm{SP0\_Y} \\ \mathrm{SP1\_Y} \\ \dots \\ \mathrm{SP31\_Y\ (SP16\_Y)} \\ \mathrm{SP32\_Y\ (SP17\_Y)}
\end{bmatrix}
=
\begin{bmatrix}
\mathrm{SETPOINT\_OUT(1)} \\ \dots \\ \mathrm{SETPOINT\_OUT(N)}
\end{bmatrix} \\
\text{or } A \times x = b \\
\text{with } A = \text{setpoint relation matrix} \\
b = \text{list of target values} \\
x = \text{list of setpoints}
\end{gathered}$$

The number of rows in the matrix and the number of values in vector b (SETPOINT_OUT values) equal the number of data points (measured or interpolated). The minimum number of rows in the matrix is 32. It is possible to have more than one measured or calculated value per interval. In this case the system is overdetermined.

The goal is to calculate the coefficients $\bar{x}$ by solving the quadratic minimization problem (Eq. 45). This minimization problem has a unique solution given by solving the normal equations (Eq. 46).

(45)

$$\bar{x} = \min \|Ax - b\|^2$$

(46)

$$\begin{gathered}
A^T \times A \times \bar{x} = A^T \times b \\
\bar{x} = (A^T \times A)^{-1} \times A^T \times b \\
\bar{x} = \text{solution (new setpoints)}
\end{gathered}$$

Usually this method is used by linear equation solvers and is called least squares method. The new setpoint values in the vector $\bar{x}$ (solution) have to be translated into register values (described in Section 2.3.9) before they are programmed into the sensor. The final scaling to the desired output range is then realized by OUT_GAIN and OUT_OFFSET (see Eq. 38).

4.4.4. Calculation of the Setpoints (for Variable Setpoints)

If variable setpoints are enabled (SETUP_DATAPATH[0] = 1) the customer is able to set the position of the setpoints defined by the x and y values. The setpoint x values are stored absolute and the setpoint y values are stored differentially to the corresponding x values (see Eq. 8 and Table 2–1 for neutral x positions). The advantage of variable setpoints over fixed setpoints is the possibility to focus more on the nonlinear parts of the measurement range. Therefore, fewer setpoints are needed to achieve a linearization of the output. An example of a distribution of variable setpoints is shown in Fig. 4–16.

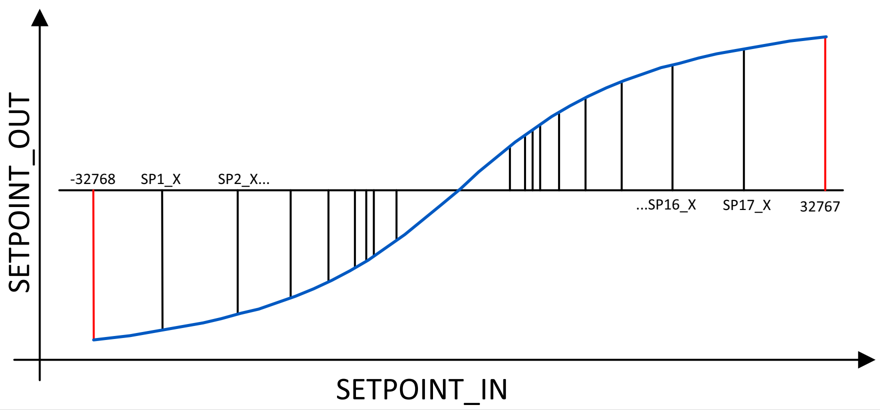

Fig. 4–16: Distribution Example of Variable Setpoints

17 setpoint x values (SP1_X to SP17_X) and 19 setpoint y values (SP0_Y to SP18_Y) are available. The first and the last setpoint y value (SP0_Y and SP18_Y) are stored differentially to the minimum (SP0_X = –32768) and the maximum (SP18_X = 32767) of full scale.

Note: It must be ensured that the setpoint x values are ascending.

4.5. Procedure Flowcharts

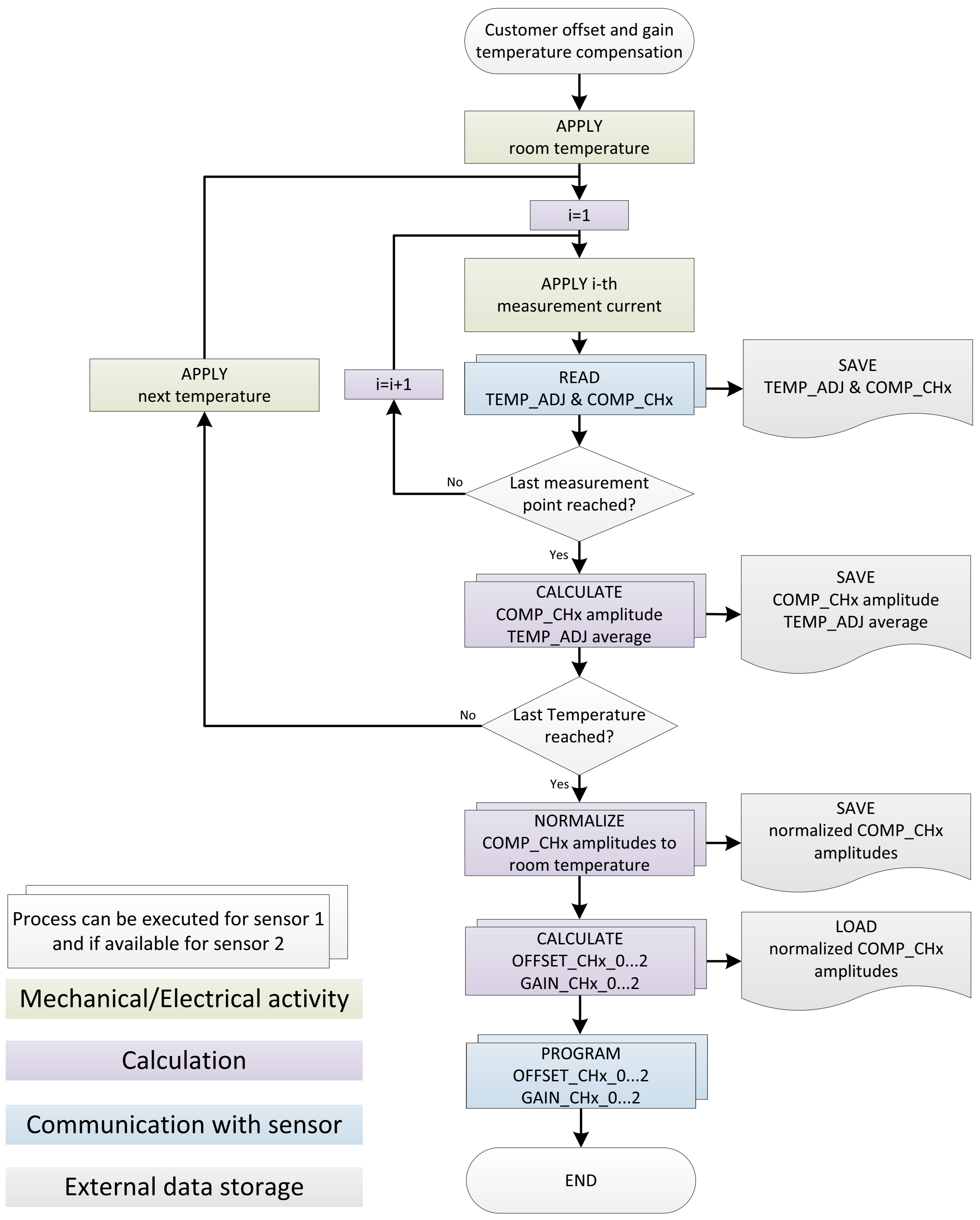
Customer offset and gain temperature compensation
APPLY room temperature
i=1
APPLY i-th measurement current
READ TEMP_ADJ & COMP_CHx
SAVE TEMP_ADJ & COMP_CHx
i=i+1
APPLY next temperature
Last measurement point reached?
No
Yes
CALCULATE COMP_CHx amplitude TEMP_ADJ average
SAVE COMP_CHx amplitude TEMP_ADJ average
Last Temperature reached?
No
Yes
NORMALIZE COMP_CHx amplitudes to room temperature
SAVE normalized COMP_CHx amplitudes
CALCULATE OFFSET_CHx_0...2 GAIN_CHx_0...2
LOAD normalized COMP_CHx amplitudes
PROGRAM OFFSET_CHx_0...2 GAIN_CHx_0...2
END
Process can be executed for sensor 1 and if available for sensor 2
Mechanical/Electrical activity
Calculation
Communication with sensor
External data storage

Fig. 4–17: Procedure Flowchart for a Customer Offset and Gain Temperature Compensation

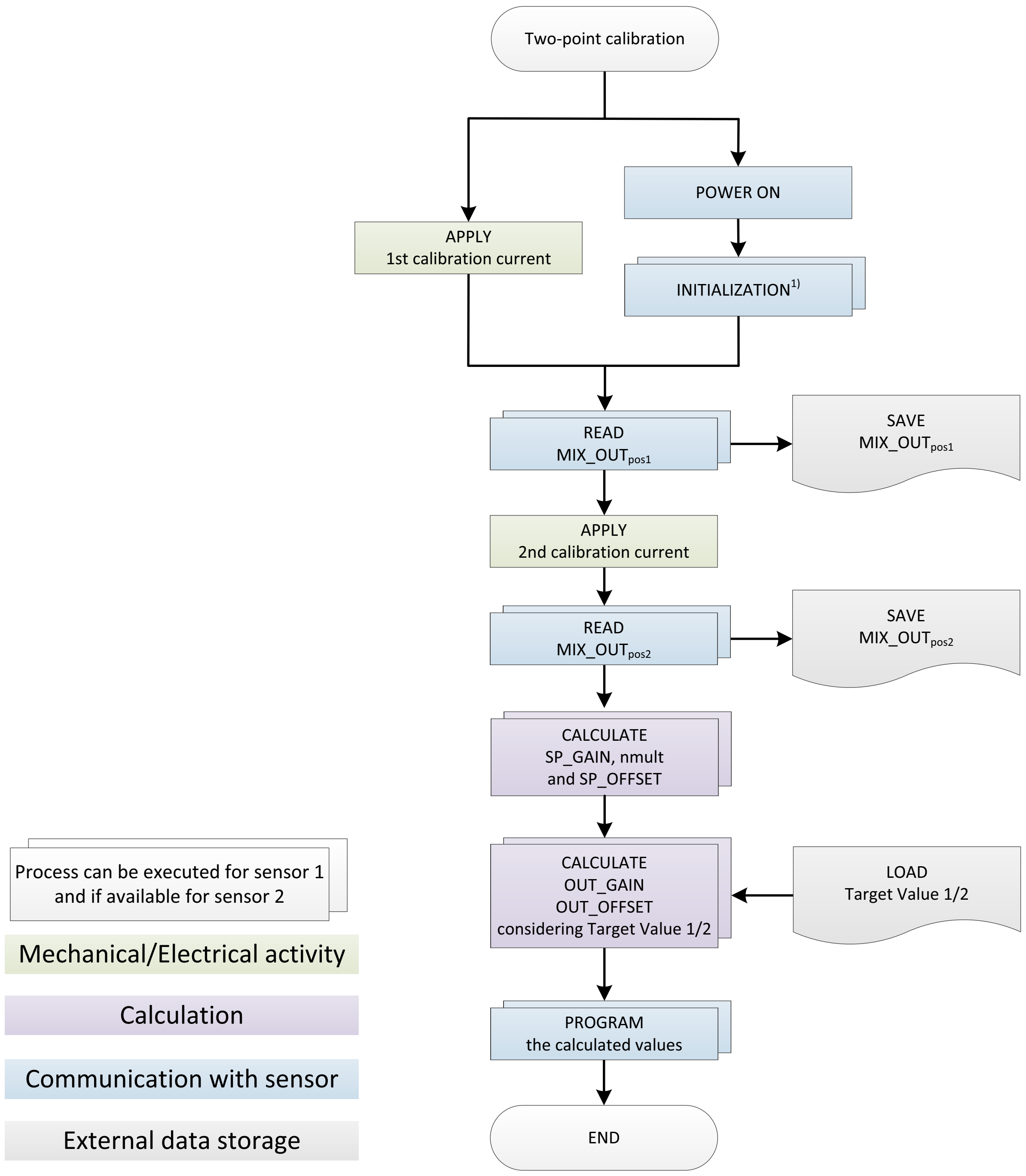

Fig. 4–18: Procedure Flowchart for a Two-Point Calibration

$^{1)}$Initialization can include several different programming steps, which can be adapted to the customer's requirements, such as: switch to programming mode, setting the set-points to neutral values (recommended), choosing proper channel gains, choosing a proper customer offset if necessary, etc. These programming steps can be performed while moving the application to the first calibration point.

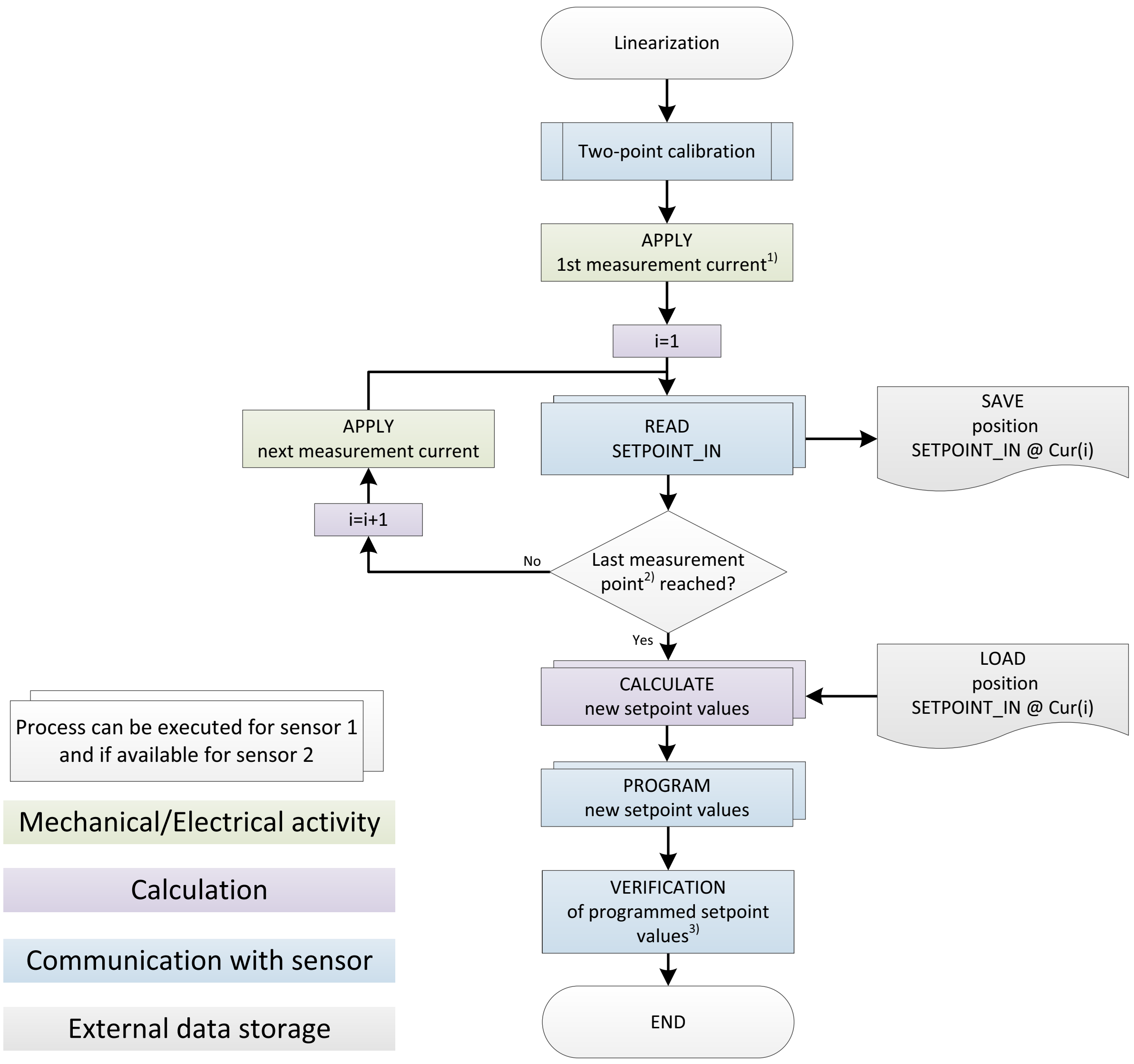

Fig. 4–19: Procedure Flowchart for a Linearization with Initial Two-Point Calibration

[1]It is recommended to set the first measurement point between the first and the second setpoint. For example: 1st calibration point.

[2]It is recommended to set the last measurement point between the second last and the last setpoint. For example: 2nd calibration point.

[3]The programmed setpoint values should be verified at the end of the linearization procedure. This can be simply realized by reading out the sensor's register values and comparing with the calculated setpoint values. To evaluate the performance of the conducted linearization, it is recommended to measure the output level over the full calibration range and compare with the target values.

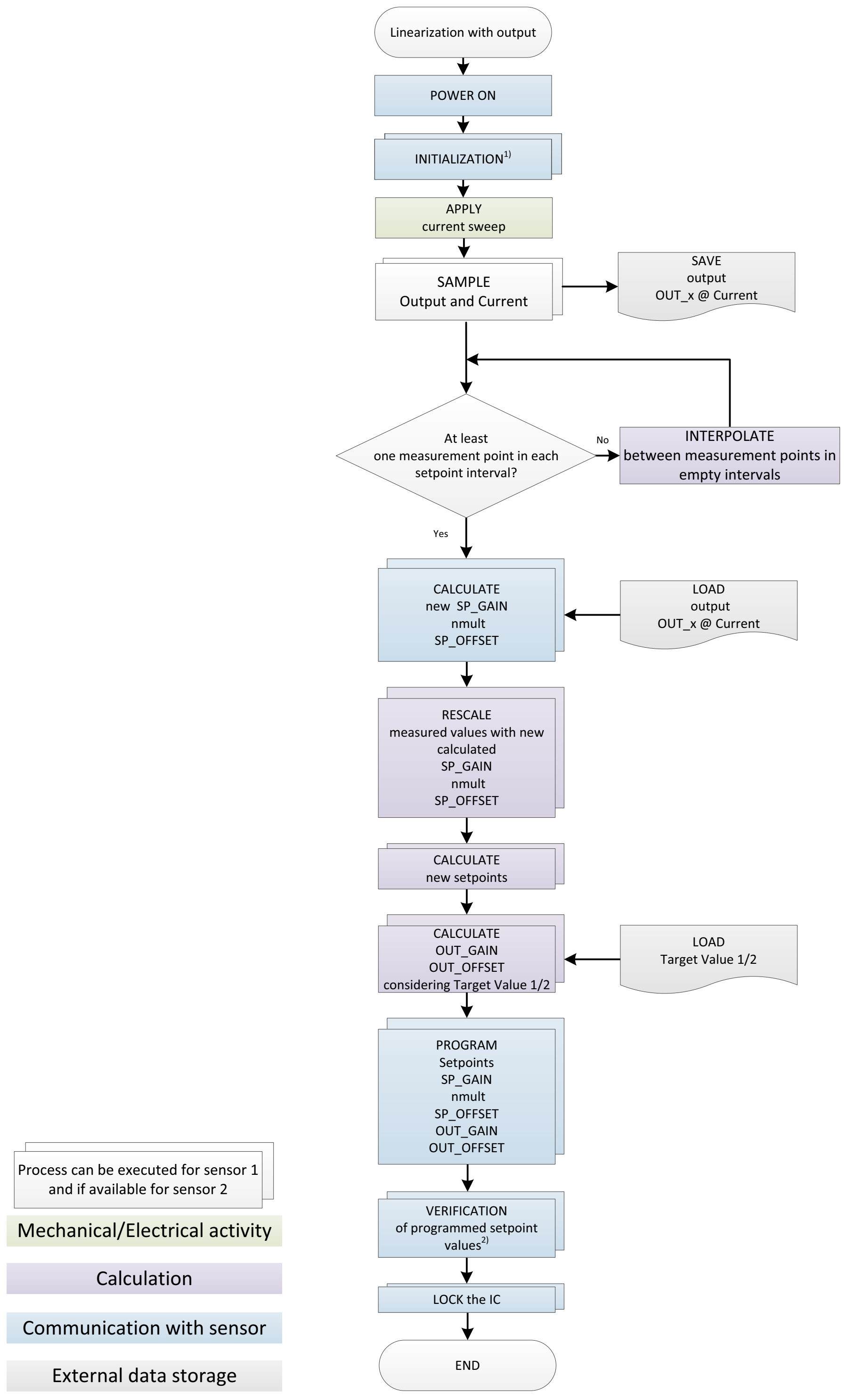
Linearization with output
POWER ON
INITIALIZATION1)
APPLY current sweep
SAMPLE Output and Current
SAVE output OUT_x @ Current
At least one measurement point in each setpoint interval?
No
INTERPOLATE between measurement points in empty intervals
Yes
CALCULATE new SP_GAIN nmult SP_OFFSET
LOAD output OUT_x @ Current
RESCALE measured values with new calculated SP_GAIN nmult SP_OFFSET
CALCULATE new setpoints
CALCULATE OUT_GAIN OUT_OFFSET considering Target Value 1/2
LOAD Target Value 1/2
PROGRAM Setpoints SP_GAIN nmult SP_OFFSET OUT_GAIN OUT_OFFSET
VERIFICATION of programmed setpoint values2)
LOCK the IC
END
Process can be executed for sensor 1 and if available for sensor 2
Mechanical/Electrical activity
Calculation
Communication with sensor
External data storage

Fig. 4–20: Procedure Flowchart for a Linearization with Sensor Output

1)Programming of the initial parameters.
2)The programmed setpoint values should be verified at the end of the linearization procedure. This can be simply realized by reading out the sensor's register values and comparing with the calculated setpoint values. To evaluate the performance of the conducted linearization, it is recommended to measure the output level over the full calibration range and compare with the target values.

5. Application Note History

1. CUR 4000 User Manual, Dec. 9, 2020; APN000173_001EN.
   First release of the application note.
2. CUR 4000 User Manual, June 16, 2021; APN000173_002EN.
   Second release of the application note.

Major Changes:

– RAM value SCALE_OUT removed

– Signal path calibration routine updated: Equation 36 and 37 added in Section 4.2.3

TDK-Micronas GmbH
Hans-Bunte-Strasse 19 · D-79108 Freiburg · P.O. Box 840 · D-79008 Freiburg, Germany
Tel. +49-761-517-0 · Fax +49-761-517-2174 · www.micronas.tdk.com