

## LINEARIZATION AND TWO-POINT PROGRAMMING ON THE A31100 AND A31102

By Jacob Bader  
Allegro MicroSystems

### ABSTRACT

Many magnetic systems that operate using linear sensing produce a nonlinear output. This nonlinear output is undesirable, but it can be corrected using a strategy known as *linearization*. In order to maximize linearization efficiency, it is necessary to employ a first-order compensation strategy, here referred to as *two-point programming*, to correct for offset and sensitivity. The Allegro A31100 and A31102 programmable linear Hall ICs provide a high level of programmability, including linearization features, that make each ideal for many different applications. This application note highlights the procedure for calculating the magnetic field compensation parameters through two-point programming and calculating the linearization coefficients using available software tools from Allegro MicroSystems, with the intention of providing an easy-to-use guide to programming these sensor ICs.

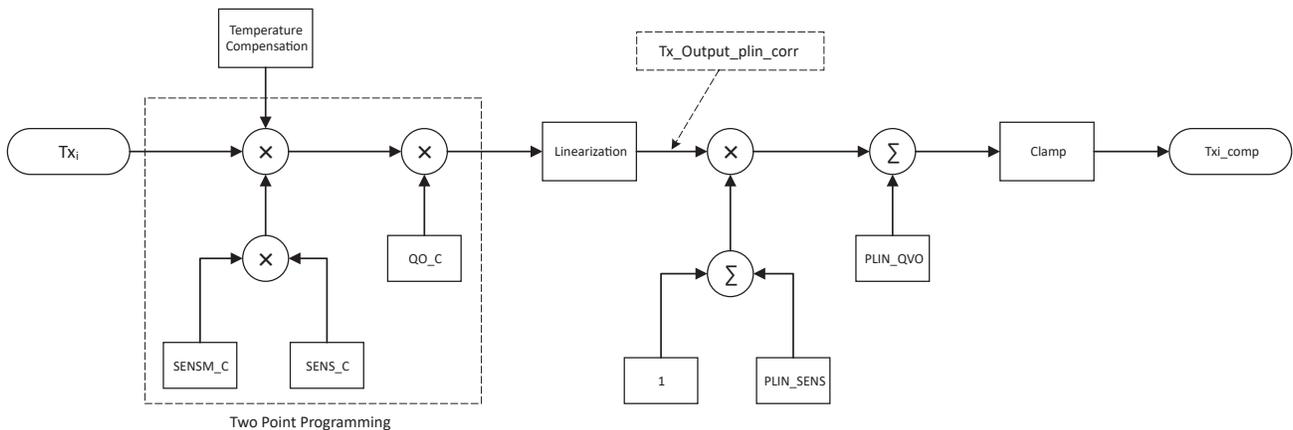


Figure 1: Signal Path of the A31100/A31102

### INTRODUCTION

Magnetic sensing in harsh environments requires high accuracy while overcoming limitations both from the sensor IC and magnetic system. Limitations from the sensor IC, including errors caused from temperature drift and noise, can be overcome through digital filtering and factory and/or customer sensor calibration. The magnetic system limitations can be caused by field strength variation and field nonlinearity. The A31100 and A31102 (referred to collectively as A31100/2) programmable linear Hall ICs contain a high level of customer-reprogrammable options to allow the sensor to be used in a wide variety of input fields and applications. One important feature of the A31100/2 is the ability to produce a linear output from many nonideal input field signals.

#### Advantages of Linearization

Linearizing and scaling of the A31100/2 outputs allow nonideal fields to be used in applications. While the Hall-effect sensor IC can output a proportional digital signal to the input magnetic signal, it is often desirable to have a linear output to more easily differentiate between different field inputs. Nonlinear magnetic fields can be common among magnets, and on-chip linearization of the nonlinear field may help to reduce cost and system complexity. These features also will help minimize detection errors that come from the inherent errors of the magnetic field and reduce uncertainty of linear position.

## Linearization and Compensation Process

In order to obtain the best results out of the linearization it is first necessary to have a basic understanding of the signal path where the linearization occurs. Figure 2 shows a simplified block diagram depicting the linearization process. The beginning of the relevant signal path labeled “Magnetic Input” is the filtered output of the ADC. Before any linearization occurs, the signal goes through pre-linearization adjustment which consists of sensitivity and offset adjustment and then linearization. The next step in the signal path is the post-linearization adjustment followed by optional output clamping. After this, the signal is ready to be output by the sensor.

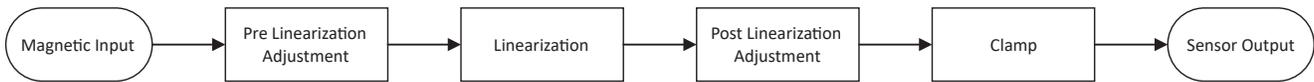


Figure 2: Basic A31100/2 Signal Path

## Linearization with the A31100 and A31102

The linearization block used in the A31100 and A31102 differs slightly from prior Allegro linearization-capable parts. Linearization works by storing a set of linearization coefficients in the sensor EEPROM that together act as the endpoints of a piecewise linear transfer function, such as seen in Figure 3. The transfer function is made up of 16 line segments that are created by the 17 coefficients. These 17 coefficients represent a scaled deviation, or error, from the desired output code for a corresponding input code. The error for an input code that falls between the defined points is linearly interpolated using the two closest coefficients.

The resolution of the linearization function can be controlled using the lin\_scalar register. The lin\_scalar parameter defaults to logic value 0, which sets the range of the coefficients to 25% of the full output range; this gives the coefficients the best resolution. If an application has large deviations making it very nonlinear, then lin\_scalar can be set to logic value 1. In this configuration, the range of the coefficients is 50% of the full output range. This means that the coefficients have less resolution but will be able to correct for larger linearization errors.

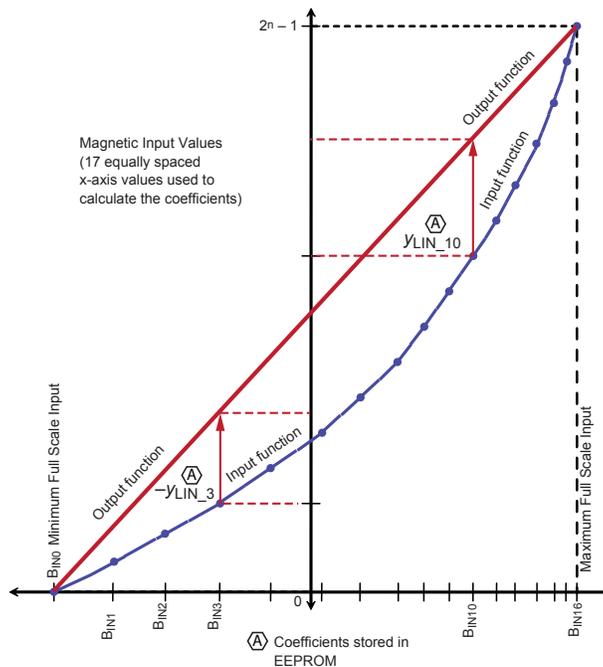


Figure 3: Visualizing A31100/2 Linearization

## Pre-Linearization Adjustment

The A31100 and A31102 and other Allegro Hall-effect sensors are factory-trimmed to provide the same initial sensitivity and offset to help overcome die-to-die sensitivity differences from fabrication. To use the sensor's full-scale output for the desired field range, it is often necessary to adjust the sensitivity and/or offset when the magnetic field range is lower than the factory-trimmed target. This is accomplished on the A31100/2 using the customer-programmable EEPROM parameters, SENS\_C, SENSM\_C, and QO\_C. SENS\_C and SENSM\_C are coarse and fine adjustments to the sensitivity, respectively. The parameter QO\_C adjusts the offset from the default factory-trimmed value. The compensation algorithm is shown visually in Figure 4. It is important to apply the sensitivity and offset correction before data points are collected for the linearization algorithm to allow the scaling to perform properly. The parameters can be calculated by means of two-point programming. This is where two points of the magnetic range are used to calculate the sensitivity scaling and offset adjustment to apply a near full-scale digital output to what is not a full-scale magnetic range of the Hall plate and ADC. For more information on factory default sensitivity and offset, consult the A31100/2 datasheets.

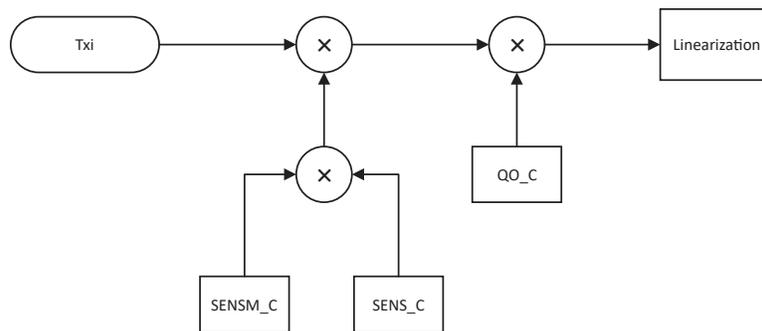


Figure 4: Pre-Linearization Compensation

## Post-Linearization Adjustment

An additional gain and offset trim stage is available in the linearization block. This can be used to attenuate or gain the signal. Using post-linearization offset adjustment with signal attenuation may be useful to extend the input to the linearization block to all 17 linearization points when the final output range is not full scale. Figure 5 shows a visualization of the post-linearization process. The output of the post-linearization block is then passed onto the clamping block.

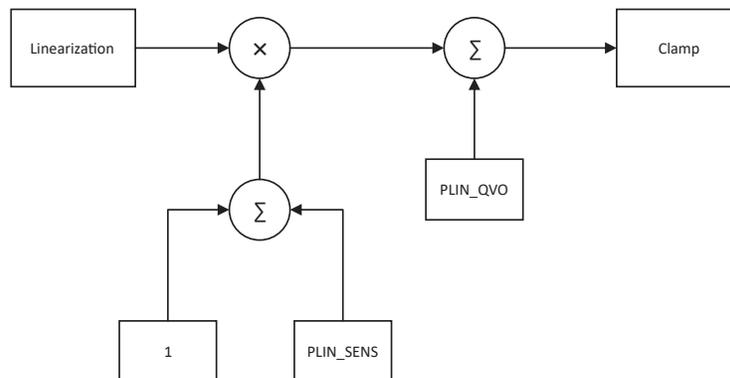


Figure 5: Post-Linearization Compensation

Together, the sensitivity and offset trim parameters allow small magnetic range signals to be used with the full available output range of the A31100/2. Though the linearization algorithm does have some ability to scale the magnetic range, it should not be used for such. Doing so may cause an increase in sensing errors due to the large range of the inputs. Using the customer sensitivity and offset parameters is the ideal way to produce a full-scale output in an application that does not align with the default magnetic range of the device.

## Calculating Linearization Coefficients

To calculate the linearization coefficients to be used by the A31100/2, 17 magnetic input points are recommended to be collected. Alternatively, it is feasible to use more or less than 17 points and then apply curve fitting techniques to approximate the coefficients. These inputs must be monotonically increasing. In some applications, it may be desired to invert the output versus input transfer function; an example of this is an application with redundancy where one output has a positive slope and the other output has a negative slope. The POL\_C register can be set to logic value 1 to invert the input of the internal linearization block. Alternatively, LINT\_OUT\_INV can be used to invert the output of the linearization block, but in most cases using POL\_C for inversion is preferred. Higher accuracy will be achieved from the linearization algorithm if the input points are evenly spread across the entire sensing range of the sensor IC, which is defined by the gain and offset parameters chosen. They may be collected after the trim components are applied by the two-point programming algorithm, or the points may be collected and adjusted externally to the part as shown later in this document.

To program the A31100/2, Allegro MicroSystems provides a programming solution compatible with the available ASEK-20 Samples Programmer, and the ASEK-21 Module Programmer in the form of a programming GUI, as well as available commands for use in a variety of programming languages and hardware interface tools. These hardware and software tools allow simple implementation and programming of the new additions to the Allegro linear family. Example code for use of these tools is included throughout this application note.

The calculation of the linearization coefficients can be done with the available programming function included in the software tools called *CalculateLinearizationCoefficients*. This command takes the 17 input magnetic data points and converts them for use with the equally spaced output points, using the full range of the output of the part (0-65535) as a means of scaling. The function returns an integer array of the 17 coefficients to be programmed to the device EEPROM. Additionally, the function can determine if the input and output of linearization must be inverted, as well as the value of the lin\_scalar register.

## LINEARIZATION PROCEDURE

The following is a step-by-step description of how to program the A31100/2 to provide a linear output for the desired input magnetic. The process can be completed with DLLs and software commands for the customer's own end-of-line solution or through the software programming GUI available from Allegro. This procedure assumes the measurements are made at 25°C to ignore the effects of the temperature compensation. See A31100 and A31102 datasheets for more information. This application note includes directions for those customers who prefer to integrate device programming into their production line, as well as directions for those who would like to use Allegro's ASEK-20 programmers and software.

### Customer Programming Solution

In many cases, it is desirable to program—quickly and efficiently—many A31100/2 sensor ICs using a custom end-of-line test solution. Mass programming of devices is possible through available DLLs and software functions and will be discussed in this section. The steps of the process involve a combination of sensor memory reads/writes and external calculations to determine parameter values.

In order to make the linearization as efficient and least time-consuming as possible in an end-of-line test, the process has been adjusted to only use a single mechanical sweep to collect one set of data. Collecting a single set of data points means that a majority of the following calculations will occur backwards over the signal path which might be confusing at first glance.

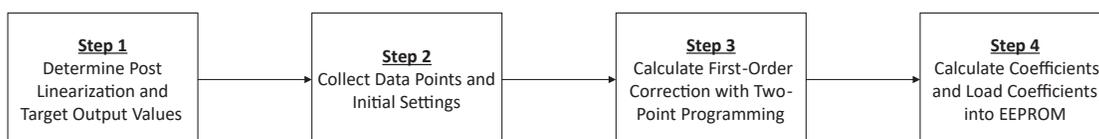


Figure 6: Linearization Steps Visual Guide

## Step 1: Determine Post-Linearization and Target Output Values

To use all 17 linearization coefficients, the input to the linearization block must span nearly full scale. But a full-scale output for a given magnetic motion is not desirable for all applications. By making use of the pre-linearization gain and offset adjust, the input signal to the linearization block can be maximized. The signal is attenuated by using the post linearization gain/offset adjust to match the desired output range. This allows the linearization algorithm to provide higher accuracy and a scaled output based on Equation 1:

$$\text{Linearized\_Tx\_Output} = (1 + \text{PLIN\_SENS}) \times \text{Linearization\_Output(Tx)} + \text{PLIN\_QVO}$$

Equation 1: Output after Linearization and Post-Linearization is Applied

In Equation 1, PLIN\_SENS and PLIN\_QVO are programmable EEPROM parameters. These post-linearization parameters are final adjustments on the sensitivity and offset of the sensor. Together with SENS\_M\_C and SENS\_C (the gain parameters) and QO\_C (the offset parameter), the default sensor IC sensitivity and offset can be adjusted to provide a compensated output to achieve the requirements of the application within the device specifications.

To determine the values to write to EEPROM for the post-linearization compensation, the desired outputs that signify the minimum and maximum field applied are used along with a user-defined range for the post-linearization (typically a maximum of 98% of full-scale output is chosen to allow padding for rounding and signed-to-unsigned conversion errors that may cause undesired clipping). If the customer does not desire to use post-linearization, these values can be set to zero, creating a gain of 1 and offset of 0. It is recommended to use the post-linearization feature to provide the highest accuracy from the algorithm.

$$\text{PLIN\_SENSE} = [ (\text{TARGET\_MAX} - \text{TARGET\_MIN}) / (\text{POST\_LINEARIZATION\_RANGE} \times 65536) ] - 1$$

Equation 2: Post-Linearization Sensitivity

$$\text{PLIN\_QVO} = \text{TARGET\_MIN} + [ (\text{TARGET\_MAX} - \text{TARGET\_MIN}) / 2 ] - 32768$$

Equation 3: Post-Linearization Offset

To incorporate the post-linearization into the calculations, the TARGET\_MAX and TARGET\_MIN unsigned values should be adjusted to account for the post-linearization in the calculations. Equation 4 and Equation 5 show how the desired target values can be adjusted to accommodate for post-linearization sensitivity and offset trim.

$$\text{TARGET\_MAX\_PLIN} = [ (\text{TARGET\_MAX} - 32768 - \text{PLIN\_QVO}) / (1 + \text{PLIN\_SENS}) ] + 32768$$

Equation 4

$$\text{TARGET\_MIN\_PLIN} = [ (\text{TARGET\_MIN} - 32768 - \text{PLIN\_QVO}) / (1 + \text{PLIN\_SENS}) ] + 32768$$

Equation 5

These new targets are used to increase the output signal, with the post-linearization block providing the attenuation necessary to make the final output match the desired values, improving the result of the linearization for small magnetic ranges. An example use case of post-linearization correction is in an application where the effective output range is less than the full-scale range.

## Step 2: Collect Data Points and Initial Settings

To perform the trim procedure, initial data points should be recorded. To use these points to perform linearization on the output of the device, the linearization table should be disabled so that any linearization coefficients previously loaded will not affect the reading of the magnetic value. To do so, set the parameter lint\_e = 0 in the lin8\_c register (Address [Bit]: 0x12 [12]) to bypass the linearization table.

Before collecting the data points, all the trim parameters shown in Table 1 should be set to their default code (value may vary for zero code). If this cannot be done before data points are collected, the trim procedure can still be run by factoring out the initial parameter values, as will be shown later.

The A31100/2 uses 17 data points to linearize the target field. The target points must be collected and stored in either continually increasing or continually decreasing order (i.e. Tx<sub>1</sub> < Tx<sub>2</sub> < ... < Tx<sub>17</sub> or Tx<sub>1</sub> > Tx<sub>2</sub> > ... > Tx<sub>17</sub>). The field should be within the operating range of the part which corresponds to greater than 0 LSB output and less than 65535 LSB output. Using data outside of this range may reduce the accuracy of the trim procedure.

If the pre-linearization gain and offset parameters were not set to zero before the data points were collected, the initial settings of the part should be read to calculate the first-order correction.

This step and all preceding steps must be done for each sensor IC within the dual-die device to achieve the higher accuracy from the individual trimming of sensor ICs.

Example code to read the setting registers and then convert the codes to values:

```
pol_c = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x03, 15, 15);
sensm_c = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x03, 14, 12);
sens_c = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x03, 10, 0);
qo_c = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x06, 17, 0);
plin_sens = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x13, 11, 0);
plin_qvo = A31100.ReadPartialRegister (Allegro.ASEK.MemoryAccessType.primary,
0x13, 23, 12);
```

The code representation of the parameters read from EEPROM can be converted to usable values with the following equations (in the form  $VALUE = f(code)$ ):

$$POL\_C = pol\_c$$

Equation 6

$$SENSM\_C = 2^{sensm\_c}$$

Equation 7

$$\text{if } sens\_c < 1024_{10} : SENS\_C = sens\_c \div 2^{11} + 1$$

Equation 8a

$$\text{if } sens\_c \geq 1024_{10} : SENS\_C = sens\_c \div 2^{11}$$

Equation 8b

$$\text{if } plin\_sens < 2048_{10} : PLIN\_SENSE = plin\_sens \div 2^{11}$$

Equation 9a

$$\text{if } plin\_sens \geq 2048_{10} : PLIN\_SENSE = plin\_sens \div 2^{11} - 2$$

Equation 9b

$$\text{if } qo\_c < 131072_{10} : QO\_C = qo\_c \times 2$$

Equation 10a

$$\text{if } qo\_c \geq 131072_{10} : QO\_C = 2(2^{18} - qo\_c)$$

Equation 10b

$$PLIN\_QVO = plin\_qvo$$

Equation 11

### Step 3: Calculate First-Order Correction with Two-Point Programming

To use linearization with the desired field at the highest resolution and accuracy, the initial sensitivity and offset values should be calculated to achieve full-scale output of the magnetic signal. This is done through two-point programming which takes place in this step.

First, the data points should be corrected for the post-linearization sensitivity and offset. The result of Equation 12 is  $Tx\_output\_plin\_corr$ ; this represents the value for the magnetic reading before post-linearization correction is applied in the signal path. An important note is that the  $Tx\_Output$  in Equation 12 is unsigned even though the results,  $Tx\_output\_plin\_corr$ , will be signed.

After correcting for the post-linearization sensitivity and offset, it is then necessary to correct for the pre-linearization gain and offset. The function of Equation 13 is to perform this correction using the result of the calculation done in Equation 12. The result of Equation 12 is  $Tx_i$ , which is the uncorrected digital signal from the Hall element which serves as the starting point for the signal path relevant in this application note.

$$Tx\_output\_plin\_corr = (Tx\_Output - 32768 - PLIN\_QVO) / (1 + PLIN\_SENS)$$

Equation 12

$$Tx_i = (Tx\_output\_plin\_corr - QO\_C) / (SENS\_C \times SENSM\_C)$$

Equation 13

$$\text{if } pol\_c = 1: \quad Tx_i = -1 \times Tx_i$$

Equation 14

Note that if there are no compensation parameters loaded to the device when the initial data points are collected, these corrections will result in the same data. This calculation can be skipped if these compensation parameters in Table 1 are set to the default zero (default code is 0; default value can vary, i.e. the sensitivity parameters are default value of 1 for a code of 0). It may be desired to set these parameters to zero before reading the sensor output since invalid parameters may cause the sensor output to clamp if the gain is too high for the applied field, resulting in erroneous data. Additionally, the  $PLIN\_SENS$  and  $PLIN\_QVO$  calculated in step 1 should only be used in Equation 18, and the  $PLIN\_SENS$  and  $PLIN\_QVO$  which were read from EEPROM must be used in Equation 12.

In order to begin performing first-order correction, it is necessary to follow the above steps to compute  $Tx_i$  for  $TARGET\_MIN$  and  $TARGET\_MAX$  using Equation 12 and Equation 13. These two values will be referred to as  $Tx_1$  and  $Tx_{17}$ , which correspond to  $TARGET\_MIN$  and  $TARGET\_MAX$  respectively. The calculation begins with the gain component of the correction in Equation 15.

$$Gain = (TARGET\_MAX\_PLIN - TARGET\_MIN\_PLIN) / (Tx_{17} - Tx_1)$$

Equation 15

Gain in this case is the product of the parameters  $SENS\_C$  and  $SENSM\_C$ . The  $SENS\_C$  parameter has a limit of  $0.5 \leq SENS\_C < 1.5$  (the value of  $SENS\_C$  can equal 0.5, but it cannot equal 1.5; the largest accepted value is 1.4995). To incorporate more gain, the parameter  $SENSM\_C$  is used as a multiplier with the following available options: 1x, 2x, 4x, 8x, 16x, and 32x. This will give an effective gain range of  $0.5 \leq Gain < 48$  for the device. The gain factor can be divided by the possible  $SENSM\_C$  values until the  $SENS\_C$  value is less than 1.5. This will give the values to write to EEPROM. If no available  $SENSM\_C$  value can be used, then the field delta used in the setup will need to be increased or the maximum and minimum target output values decreased.

$$SENS\_C = Gain / SENSM\_C$$

Equation 16

Next, the offset parameter is calculated:

$$QO\_C = (TARGET\_MIN\_PLIN - 32768) - GAIN \times Tx_1$$

Equation 17

These parameters are now used to adjust the measured points to apply a desired output following the block diagram in Figure 1 and Equation 18:

$$Tx_{i\_comp} = \{ (1 + PLIN\_SENSE) \times [ Gain \times Tx_i + QO\_C ] \} + PLIN\_QVO + 32768$$

Equation 18

where  $Tx_{i\_comp}$  is the compensated  $i^{th}$  data point, calculated from  $Tx_i$ , the  $i^{th}$  corrected measured output data point. The set of data points will be referred to as *Tx-Compensated*.

The polarity bit can be easily calculated by the slope of the gain factor. If the gain factor is negative, the absolute value of gain should be used for the SENS\_C and SENSM\_C calculations, and pol\_c should be set to 1 to invert the slope. The polarity bit has no effect on the QO\_C parameter. If it is desired to change the polarity but keep the offset and sensitivity calculations the same, the user must manually invert the polarity of the QO\_C value in EEPROM.

These values can all be written into EEPROM following the example code below:

```
pol_c = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x03, 15, 15);
sensm_c = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x03, 14, 12);
sens_c = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x03, 10, 0);
qo_c = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x06, 17, 0);
plin_sens = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x13, 11, 0);
plin_qvo = A31100.WritePartialRegister(Allegro.ASEK.MemoryAccessType.primary, 0x13, 23, 12);
```

## Step 4: Calculate Coefficients and Load Coefficients into EEPROM

With the data points corrected for the new first-order compensation, the linearization coefficients can be calculated. This can be done by simply passing the 17 corrected data points from step 2 (Tx-Compensated) into the DLL function CalculateCoefficients as shown below. This will return the coefficients in the array LinearizationCoefficients.

**A31100.CalculateLinearizationCoefficients(inputArray, startValue, stopValue, linScalar, invertInput, invertOutput);**

Once the linearization coefficients have been calculated, the coefficient table can be updated in the sensor EEPROM. Two successive coefficients are stored in each register with the 17th coefficient contained in the register with the linearization enable bit as shown in Table 1.

**Table 1: Linearization EEPROM Register Addresses**

Parameter	Memory Register Address	Bit(s)
lint00, lint02, lint04, ..., lint16	0x0A – 0x12	0:12
lint01, lint03, lint05, ..., lint15	0x0A – 0x11	13:25
lint_e	0x12	13

To load the coefficients to the device, the proceeding example code may be used.

```
gui_coeffs = [...];
gui_coeffs(gui_coeffs<0) = gui_coeffs(gui_coeffs<0)+2^13; %converts to signed values

%packs the lin values into the eeprom structure
for ii = 1:8
    lin_eep_vals(ii) = gui_coeffs(2*ii)*2^13+gui_coeffs((2*ii)-1);
end
lin_eep_vals(9) = gui_coeffs(end)+2^13; %this is the 17th point. The 2^13 sets the lin enable bit
address_offset = hex2dec('2a')-1;

%writes coeffs to shadow
for ii = 1:length(lin_eep_vals)
    A31100.WriteMemory(Allegro.ASEK.MemoryAccessType.shadow,address_offset+ii,lin_eep_vals(ii))
end
```

## Linearization with the Allegro A31102 Programmer

When evaluating the A31100/2 for use in a linear sensing application, Allegro can provide a software programming GUI that can make implementation of the steps mentioned in the previous section easier. The calculation of trim parameters and writing to EEPROM are handled automatically using the same process. For more information on the GUI and how to use it to evaluate the linearization of the sensor, contact Allegro MicroSystems.

## LINEARIZATION EXAMPLE

With each die in the dual die A31100/2 functioning as an independent sensor, the application of the linearization steps can be focused on one single die (called T1 here). Applying the same steps and target values to each die is not required if the application calls for different output from each die. Each die should calculate the linearization coefficients and compensation parameters from its own collected data points. While factory trimming aims to provide sensors that will produce the same output for a given input strength, varying drift of the Hall sensing elements can cause variation in the output. To provide the highest correlation between the final outputs of the two sensors, the 17 output points for each die should be collected from the same target positions.

This example will go through the steps to compensate and linearize readings from an imaginary system. The first step will be to determine the target output values and calculate the post linearization settings. The second step will be to collect initial data to characterize the system. This data will be used in step 3 to calculate the optimal settings for the application. Once the relevant settings have been calculated, step 4 will consist of calculating the correct linearization coefficients. Step 5, the last step, the results will be loaded into the device's EEPROM and the results of the complete process can be visualized. It should be noted that all calculations use 16-bit integers where applicable.

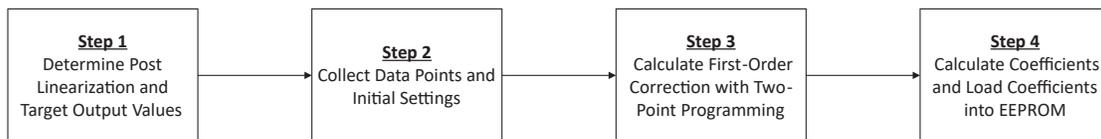


Figure 7: Linearization Steps Visual Guide

### Step 1: Determine Post-Linearization and Target Output Values

For this example, target maximum and minimum are selected to be 2% and 98% of the full-scale output. This means that the theoretical maximum output code should be 64225 and the minimum should be 1310, which means that TARGET\_MIN will be 1310 and TARGET\_MAX will be 64225.

Using Equation 2 and Equation 3, PLIN\_SENS and PLIN\_QVO can be calculated.

$$PLIN\_SENS = (64225 - 32768) / (0.98 \times 65536) - 1 = -0.0204$$

$$PLIN\_QVO = 1310 + (64225 - 1310) / 2 - 32768 = 0$$

PLIN\_SENS is calculated to be -0.0204 and PLIN\_QVO is 0. This means that after linearization is completed, the output code will be scaled by a factor of 0.9796 (1 - 0.0204) keeping the output code between 1310 and 64225 as desired. Equation 4 and Equation 5 are useful for verifying that the post-linearization sensitivity and offset was computed correctly.

$$TARGET\_MAX\_PLIN = (64225 - 32768 - 0) / (1 - 0.0204) + 32768 = 64880$$

$$TARGET\_MIN\_PLIN = (1310 - 32768 - 0) / (1 - 0.0204) + 32768 = 655$$

## Step 2: Collect Data Points and Initial Settings

Begin by reading the initial parameters from the sensor to populate Table 2. This data is necessary so that any gain or offset compensation done by the sensor can be factored out in order that the linearization coefficients can be correctly determined. As mentioned earlier, the reading of these values can be accomplished using calls to the provided DLLs, the Allegro MicroSystems GUI, or other custom scripts. Also mentioned earlier, the codes from EEPROM must be converted into values which will be used in later calculations.

**Table 2: Initial settings in the sensor EEPROM**

Parameter	Initial Code	Initial Value
pol_c	0	0
sensm_c	0	1×
sens_c	0	1.0000
qo_c	0	0
plin_sens	0	0
plin_qvo	0	0

Once the initial parameters have been collected, the sensor readings should be captured over the range of the application. The data collected in the example application is seen in Table 3.

**Table 3: Collected magnetic data points in both signed and unsigned representation**

Point	Initial Sensor Reading	Initial Sensor Reading Signed
1	12739	-20029
2	16451	-16317
3	20130	-12638
4	23302	-9466
5	25831	-6937
6	28502	-4266
7	32697	-71
8	35685	2917
9	39287	6519
10	42688	9920
11	46114	13346
12	49173	16405
13	52300	19532
14	56734	23966
15	59604	26836
16	62038	29270
17	64296	31528

### Step 3: Calculate First-Order Correction (Two-Point Programming)

The first step in two-point-programming is to factor out the post-linearization sensitivity and offset. This calculation is accomplished using Equation 12. The calculation should be done for point 1 and 17.

$$Tx_{1\_output\_plin\_corr} = (12739 - 32768 - PLIN\_QVO) / (1 + 0) = -20029$$

$$Tx_{17\_output\_plin\_corr} = (64296 - 32768 - PLIN\_QVO) / (1 + 0) = 31528$$

Once Equation 12 has been used to factor out the post-linearization parameters, Equation 15 should be used on the results of Equation 12 for both point 1 and 17.

$$Tx_1 = (-20029 - 0) / (1 \times 1) = -20029$$

$$Tx_{17} = (31528 - 0) / (1 \times 1) = 31528$$

Now that the initial parameters of the sensor have been factored out, what is left is the necessary information to determine the values for SENS\_M\_C, SENS\_C, and QO\_C. The first step to do this is determine the gain which represents the product of SENS\_C and SENS\_M\_C using Equation 15.

$$Gain = (64880 - 656) / [(31528 - (-20029))] = 1.2457$$

The gain value may need to be then further divided down so that it may be turned into values which can be written into EEPROM. The maximum value for SENS\_C is 1.5 and if the calculated gain is larger than that, then SENS\_M\_C, the coarse sensitivity adjustment, can be adjusted to maintain a SENS\_C below 1.5. In this case, the gain was 1.2457, which is between the minimum and maximum values of SENS\_C, which means SENS\_M\_C should be set to a value of 1.

The remaining parameter, QO\_C, can be calculated by using Equation 17 as shown below.

$$QO_C = (656 - 32768) - 1.2457 \times (-20029) = -7162$$

Combining the calculated parameters, the following table can be generated which will provide the values that must be written to EEPROM.

**Table 4: Calculated Compensation Parameters from the Two-Point Programming**

Parameter	Final Value	Final Code
pol_c	0	0
sensm_c	1	0
sens_c	1.2457	502
qo_c	-7162	258563
plin_sens	-0.0204	4054
plin_qvo	0	0

Now the measured values can be corrected using Equation 18 which applies the corrections calculated from two-point programming. Below is an example using the calculation of  $Tx_{1\_comp}$  which is the value the sensor would output given an input equal to  $Tx_1$ . The calculation should be repeated to determine the Tx compensated value for each data point. Table 5 shows the initial data points, Tx Corrected data, and Tx Compensated data. To review, Tx Corrected represents the value at  $Tx_1$  on the Signal Path Diagram in Figure 1 and Tx Compensated is final output of the sensor after the values calculated from two-point programming are calculated. Additionally in Table 5, under Tx Corrected is the magnetic reading with all the sensitivity and offset parameters factored out.

$$Tx_{1\_comp} = \{ (1 - 0.0204) \times [ 1.2457 \times (-20029) - 7162 ] \} + 32768 = 1311$$

Table 5: Initial Sensor Data, Backwards Calculated Tx Value, and Data After Two-Point Programming

Point	Initial Sensor Reading	Tx Corrected	Tx Compensated
1	12739	12739	1311
2	16451	16451	5841
3	20130	20130	10330
4	23302	23302	14201
5	25831	25831	17287
6	28502	28502	20546
7	32697	32697	25665
8	35685	35685	29312
9	39287	39287	33707
10	42688	42688	37857
11	46114	46114	42038
12	49173	49173	45771
13	52300	52300	49587
14	56734	56734	54998
15	59604	59604	58500
16	62038	62038	61470
17	64296	64296	64225

By plotting the three data sets in Figure 8, the figure below can be produced. The figure highlights the effects of two-point programming versus the initial data collected from the sensor.

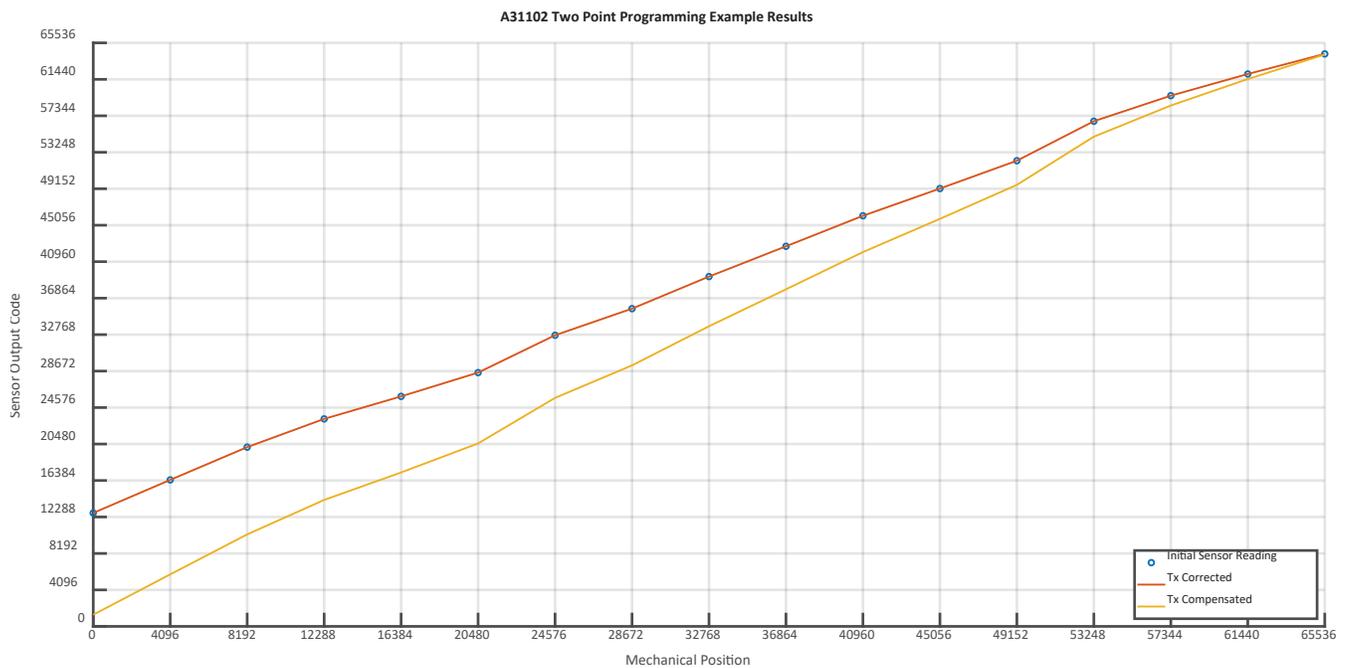


Figure 8: Two-Point Programming Example Results

## Step 4: Calculate Coefficients and Load Coefficients into EEPROM

Now that two-point programming has been completed and the newly calculated parameters are loaded into EEPROM, the process of coefficient calculation can be started. To begin, it is suggested that Tx Compensated is calculated with post-linearization sensitivity and offset removed. This new data set is the input to the linearization block.

Using the provided Allegro MicroSystems programming GUI (shown in Figure 9), the coefficients can be easily calculated. The calculation can also be done using a call to the CalculateLinearizationCoefficients function as mentioned earlier in the application note.

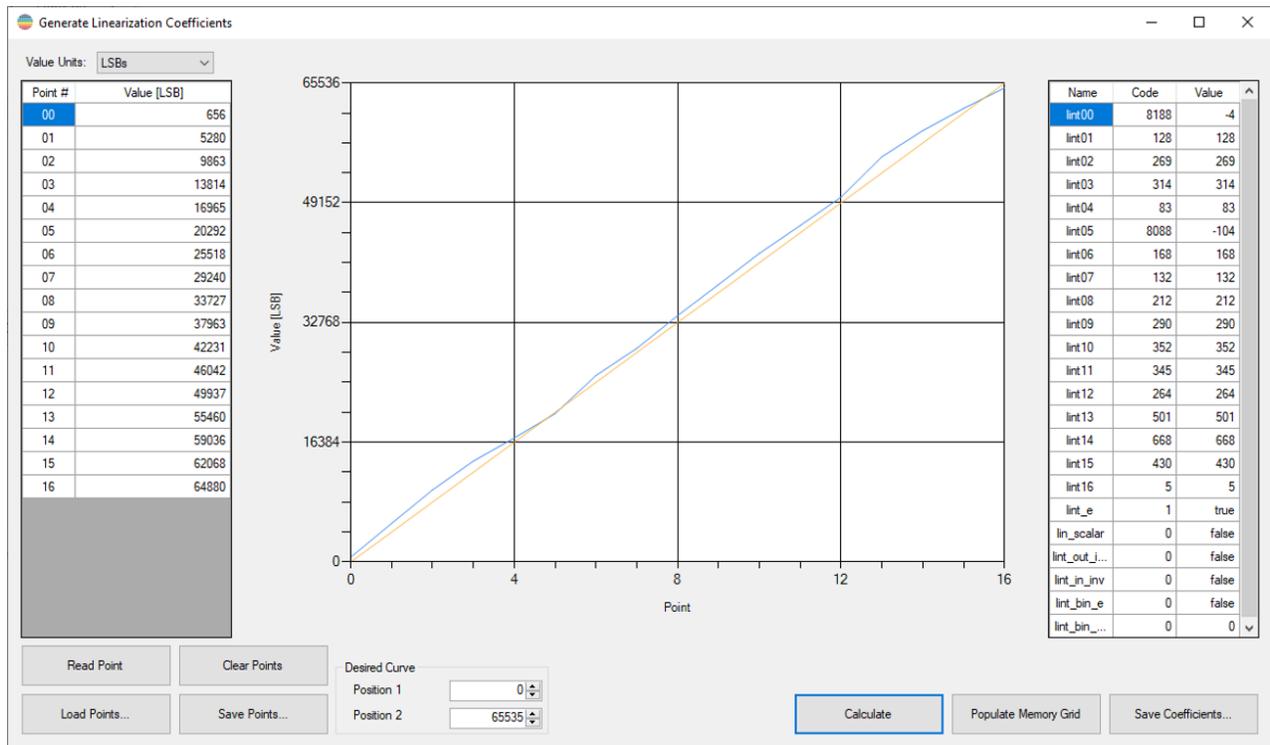


Figure 9: Calculating Linearization Coefficients with the Allegro GUI

After writing the calculated coefficients to EEPROM and ensuring linearization is enabled, a final set of data can be collected. Using this final set of data points, the effectiveness of the linearization can be visualized. Below in Figure 10, the linearized output of the device is seen in blue and the non-linearized output is shown in red. Additionally, the output of the device was characterized by taking the difference between the output of the sensor and the ideal output of the device. The uncorrected output error—represented by the pink dashed line—shows a much higher error than the solid pink line which represents the corrected output error. The average error of the uncorrected output is 913 codes and the average error of the corrected error was 29 codes. This decrease in error is proof that the linearization was successful.

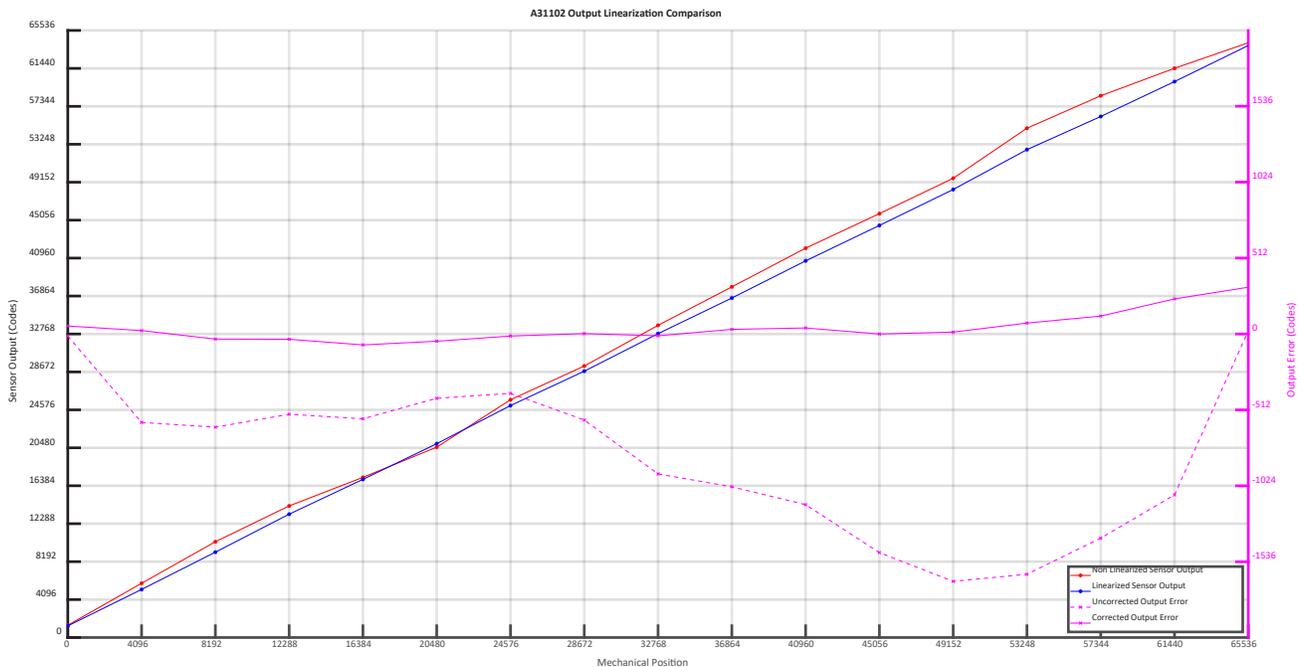


Figure 10: A31102 Output Linearization Comparison

## LIMITATIONS OF LINEARIZATION

While linearization can help overcome many non-ideal magnetic input signals, certain cases occur where linearization will not function properly. For example, if the range of the magnetic signal used is sufficiently small, noise and gain limitations could cause even discrete magnetic inputs to be misinterpreted. The A31100/2 has the capability of increasing the magnetic signal with gain stages of up to 48 $\times$ . This can allow the part to function on field ranges as low as  $\pm 40$  G; however, using a gain this high will also increase the noise from the ADC and other components since they also pass through the gain block. This can lead to difficulty detecting differences in small step sizes.

When calculating the coefficients, the software uses functions that use the output magnetic data as input points. This creates an inverse function for linearizing the output. Because of this inverse relationship, flat portions of magnetic data can cause a problem for the algorithm. Since the inverse of the slope of a flat line is infinite, the output algorithm may be undefined, creating the potential for errors.

In the selection of the sensed magnetic system, it becomes imperative to have distinct data points to collect, in a similar fashion that a small magnetic range can be hard to detect as the noise hinders the linearization.

## CONCLUSION

Using the A31100 and A31102 linearization and two-point programming can help overcome many inherent errors in a magnetic sensing system while reducing cost and complexity. Each of the A31100 and A31102 follow the procedure highlighted in this document to scale the magnetic range to the full-scale output of the sensor and linearize the output based on user-collected data from the physical system.

This can be accomplished by only two sensor read/write cycles and minimal processing. While not every input signal can be linearized by the A31102 and A31100, many fields can be, resulting in higher accuracy in linear sensing applications which, combined with the other sensor features, make the A31100 and A31102 ideal for harsh sensing environments.

## PARAMETERS AND TERMINOLOGY

Terms and parameters used in this application note are defined here. For additional information, refer to the A31100 and A31102 datasheets.

**Sensitivity:** The change in digital output versus the change in magnetic input. At the sensor IC level, sensitivity is  $\Delta\text{LSBs}/\Delta B$ , where B is the applied magnet flux density perpendicular to the Hall sensing element, and LSBs represent the digital magnetic output. At the system level, the input may be defined in units of distance, rotation, or other, depending on the application.

**Offset:** The output at zero input. At the sensor IC level, the offset is the output when the applied magnetic flux density is zero gauss. This point is also referred to as the quiescent output, QO. At the system level, the offset is the output at zero input.

**Linear Error:** How closely the output resembles a straight-line linear function. Determined by Equation 19.

$$\text{Linear Error} (\% \text{FSO}) = \max(\text{SensorOutput} - \text{LinearRegression}_i) / 65536 \times 100$$

Equation 19

**Linearization\_Output:** The output of the linearization block which is the input to the post-linearization adjustment block.

**Linearized\_Tx\_Output:** Fully compensated output after compensation, linearization, and post-linearization correction.

**Gain:**  $\text{SENSM\_C} \times \text{SENS\_C}$ .

**Tx:** Output of the ADC and filter block which is the input to the pre-linearization compensation block.

**Txi\_compensated:** Final output for the sensor at the  $i^{\text{th}}$  data point without linearization, meaning that the output includes pre-linearization offset and gain as well as post-linearization offset and sensitivity. When the linearization table is disabled, the trim parameters will scale the output based on Equation 20:

$$\text{Txi\_compensated} = (1 + \text{PLIN\_SENS}) \times [\text{Gain} \times (\text{Txi} - 32768) + \text{QO\_C}] + \text{PLIN\_QVO} + 32768$$

Equation 20

**Txi\_corrected:** The output of the sensor at the  $i^{\text{th}}$  data point when sensitivity, offset, post linearization sensitivity, and post linearization offset are disabled. Equivalent to Tx.

**TARGET\_MIN:** The minimum desired output for Tx (range of 0-65536) at the maximum physical position.

**TARGET\_MAX:** The maximum desired output for Tx (range of 0-65536) at the minimum physical position.

**Tx1:** Sensor output for the minimum applied field from Tx, corrected for initial sensitivity and offset settings.

**Tx17:** Sensor output for the maximum applied field from Tx, corrected for initial sensitivity and offset settings.

**Table 6: First order and post-linearization compensation parameters**

Parameter	Description	Memory Register Location (address[bits])
pol_c	Determines the sensitivity polarity. Default polarity is increasing with output with increasing applied south magnetic flux density.	pol_c (Register sens_trim_c 0x3 [15])
sensm_c	Coarse Sensitivity multiplier	sensm_c (Register sens_trim_c 0x3 [14:12])
sens_c	Fine Sensitivity multiplier	sens_c (Register sens_trim_c 0x3 [10:0])
qo_c	Fine quiescent output adjustment	qo_c (Register qo_trim_c 0x6 [17:0])
plin_sens	Customer post-linearization sensitivity adjustment	plin_sens (post_lin_c 0x13 [11:0])
plin_qvo	Customer post-linearization offset adjustment	plin_qvo (post_lin_c 0x13 [23:12])

*Revision History*

Number	Date	Description	Responsibility
-	February 4, 2021	Initial release	J. Bader

**Copyright 2021, Allegro MicroSystems.**

The information contained in this document does not constitute any representation, warranty, assurance, guaranty, or inducement by Allegro to the customer with respect to the subject matter of this document. The information being provided does not guarantee that a process based on this information will be reliable, or that Allegro has explored all of the possible failure modes. It is the customer's responsibility to do sufficient qualification testing of the final product to insure that it is reliable and meets all design requirements.

Copies of this document are considered uncontrolled documents.