This product, formerly sold by ams AG, and before that optionally by either Applied Sensors GmbH, acam-messelectronic GmbH or Cambridge CMOS Sensors, is now owned and sold by

ScioSense

The technical content of this document under ams / Applied Sensors / acam-messelectronic / Cambridge CMOS Sensors is still valid.

Contact information

Headquarters:

Sciosense B.V.

High Tech Campus 10

5656 AE Eindhoven

The Netherlands

info@sciosense.com

www.sciosense.com

Application Note

AN000615

PCap04

Linearization

v1-00 • 2019-May-15

Content Guide

1 Introduction

Most types of capacitive sensors show a non-linear behavior. This means that the physical unit Z and the sensor's capacitance are not simply linearly proportional. Furthermore, the relation will include a temperature dependent term. The physical unit itself may be pressure, humidity, position or anything else. This application note describes the linearization firmware "PCap04 Linearize Firmware". This firmware is provided by **ams** for free and can be used to linearize sensors and to compensate them over temperature as measured inside the chip. The 32-bit DSP uses the resistance ratio to calculate the temperature and, based on this, takes a capacitance ratio to do all the further calculations. The final results Z for the sensor and the temperature ϑ are provided in PCap04 Results Registers 0 and 1.

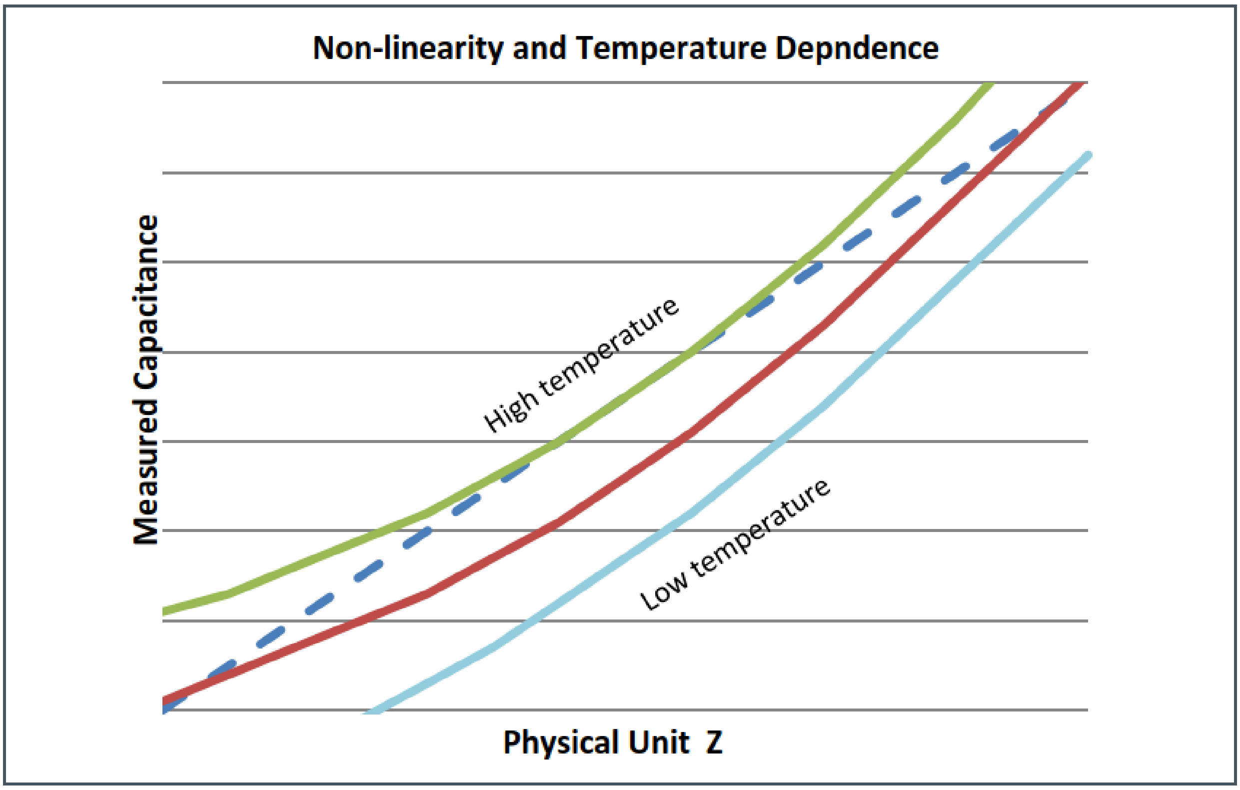
The linearization coefficients can be determined and stored individually in the EEPROM. For a simplified process, the firmware offers the option of a simple 2-point calibration where the same initial linearization coefficients are used across a complete batch of sensors. Individual sensors are then finely calibrated only at two points.

Chapter 3 will cover how this can be done with the evaluation kit.

1.1 Non-Linearity and Temperature Dependence

The characteristics of the non-linearity as well as the behavior over temperature are defined by the mechanical, electrical and chemical properties of the sensor itself.

Figure 1 :
Non-Linearity and Temperature Dependence

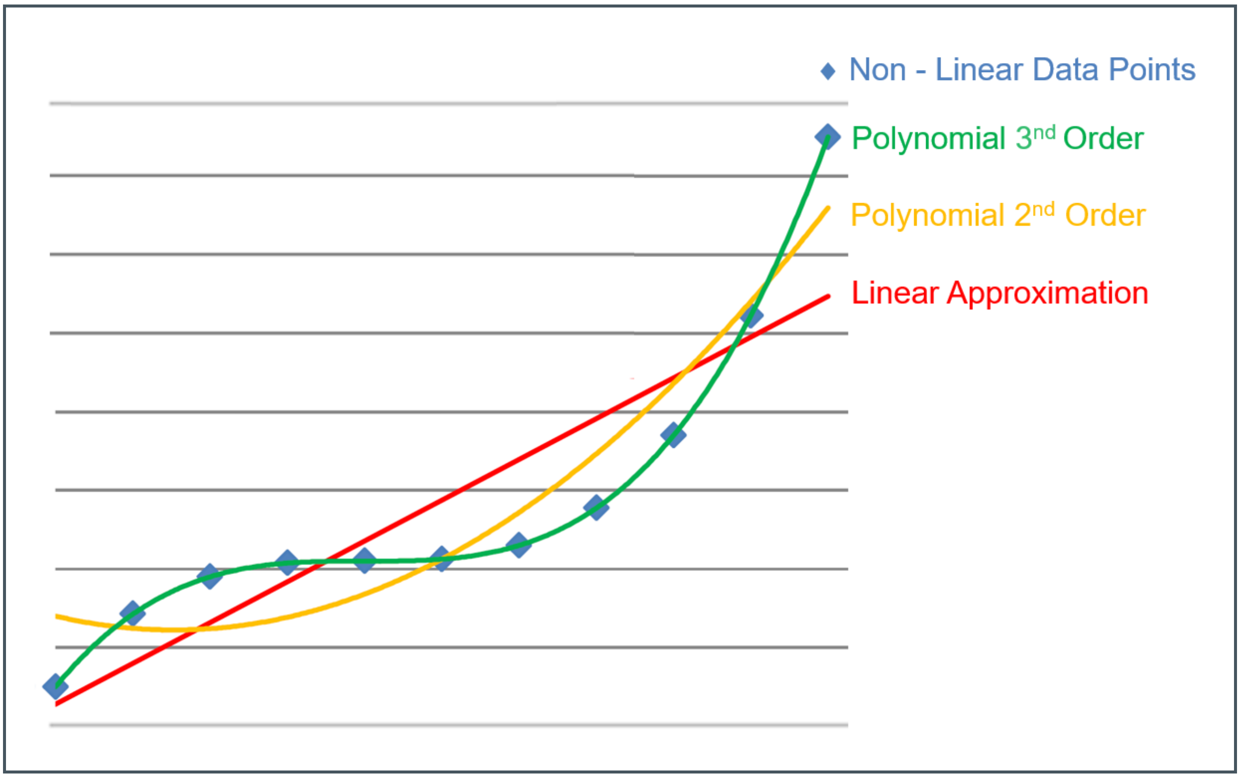


In any case it is necessary to characterize a sensor by collecting data at different temperatures for various reference values of the unit of interest. On the basis of those data, the measured values can be corrected by means of a correction table or a complex mathematical calculation for linearization.

1.2 Polynomial Approximation

An elegant way to approximate a non-linear function is the polynomial approach. The higher the order of the polynomial the better it will match the function. Of course, the mathematical effort will also increase with this method. The following graph illustrates the "best-fit" – curves ranging from a straight line to a 3rd order polynomial:

Figure 2 :
Polynomial Approximation



The blue dots indicate a non-linear response of a sensor. The red line shows the linear approximation, the yellow line an approximation by a polynomial of 2nd order and the green line an approximation by a polynomial of 3rd order. Obviously, the quality of approximation gets better the higher the order of the polynomial – especially if the non-linear curve bends several times and/or has an inflection point.

The PCap04 linearization firmware implements a 3rd-order polynomial approach for the linearization of the capacitance as well as for the resistance-to-temperature conversion.

Equation 1:

$$Z = k_3C^3 + k_2C^2 + k_1C + k_0$$

Equation 2:

$$\vartheta = tc_3R^3 + tc_2R^2 + tc_1R + tc_0$$

k_x	Coefficients of the capacitance polynomial
tc_x	Coefficients of the temperature polynomial
C	Capacitance ratio
R	Resistance ratio
Z	Output quantity
ϑ	Temperature

Additionally, the temperature information is used to correct the capacitance information. This correction is done by replacing the linearization coefficients by polynomials of second degree with temperature.

Equation 3:

$$k_3 = cc_{32}\vartheta^2 + cc_{31}\vartheta + cc_{30}$$

Equation 4:

$$k_2 = cc_{22}\vartheta^2 + cc_{21}\vartheta + cc_{20}$$

Equation 5:

$$k_1 = cc_{12}\vartheta^2 + cc_{11}\vartheta + cc_{10}$$

Equation 6:

$$k_0 = cc_{02}\vartheta^2 + cc_{01}\vartheta + cc_{00}$$

k_x	Coefficients of linearization polynomial
cc_x	Coefficients of temperature compensated polynomial
ϑ	Temperature

The 12 coefficients cc_x fully describe the characteristics of the sensor. The key point is to determine the coefficients accurately to describe the non-linear characteristic of the sensor in the best possible way. The choice of the right calibrations points is therefore important.

Note that in the firmware the function is expressed as a function of the temperature and capacitance. Replacing k_x with a_x with the substitution from (1), (3), (4), (5) and (6) and regrouping the formula gives (7) with output quantity of Z.:

Equation 7:

$$Z = a_2\vartheta^2 + a_1\vartheta + a_0$$

1.3 Determination of Coefficients

The cc_x coefficients for linearization are determined by means of least squares method. A set of measurement data needs to be collected to characterize the sensor. The physical parameter should be measured at minimum four values and three temperatures, in total minimum 12 points, to have enough data for a polynomial of third degree. Having more points will give better approximation. Critical points might be weighted by adding them twice.

Figure 3:
Coefficients Example

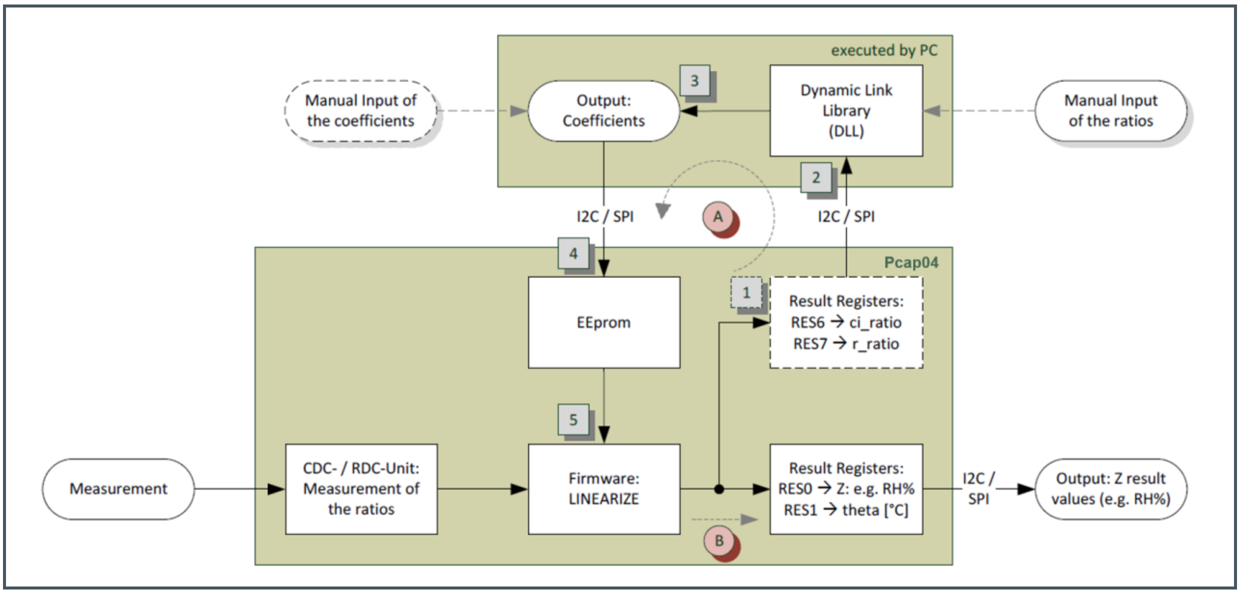
ϑ	Z	Cr/Cs
20 °C	10%	0.85123
20 °C	20%	0.86443
20 °C	30%	0.87743
40 °C	10%	0.8411

Having such a data set, it is possible to determine the linearization coefficients by means of e.g. the "least squares" method, LINEST or RGP function in Excel.

Figure 4 shows the major action items during a calibration run:

A.1 Collect raw data of capacitance ratio (ci_ratio) at various measure points by means of the linearization firmware.
A.2 Transfer these data to the DLL, which can be found in the default installation folder: c:\Program Files (x86)\ams\PCap04\data\linearize_r01.dll.
A.3 In the DLL the coefficients will be calculated.
Follow chapter 3 to accomplish these tasks with the evaluation software
A.4 Get back the coefficients cc_x and write the coefficients into the EEPROM.
B Make again measurements at various points and read back the Z output for verification.

Figure 4:
Calibration Run



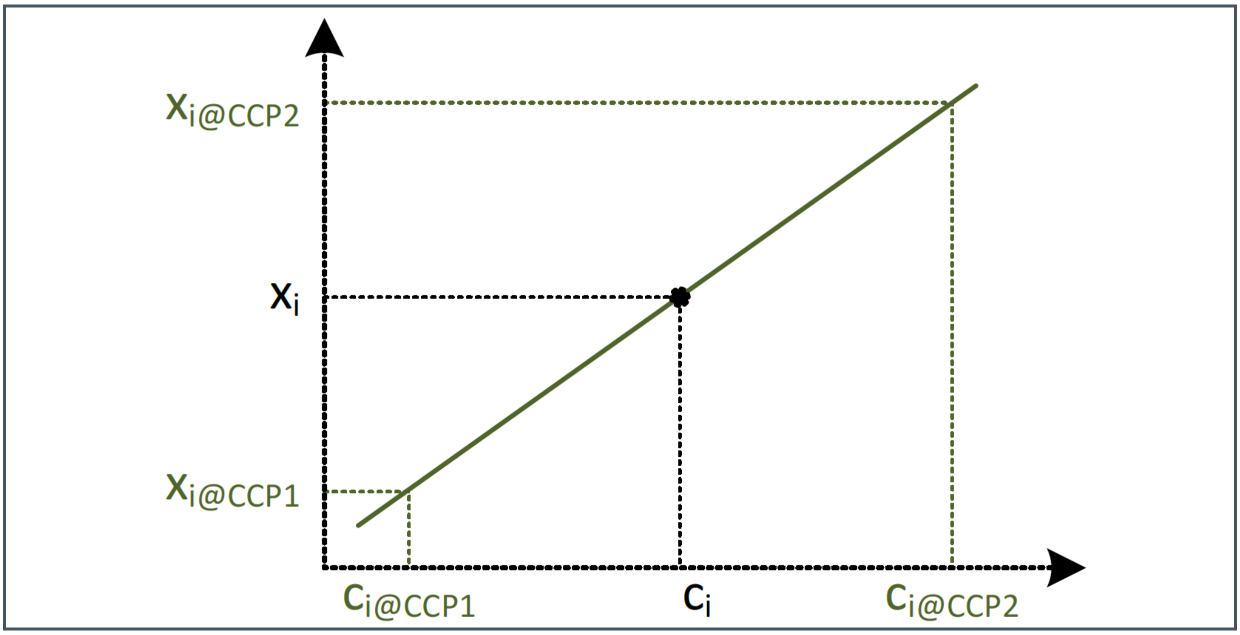
1.4 Sensor Characterization

This procedure can be used for a full calibration of each single transducer to achieve the best precision.

1.4.1 1-2 Point Calibration

In many applications, a full calibration of every single sensor will be too expensive. In case the sensors show more or less the same characteristics over a production lot, there is a chance that 2-point calibration or even 1-point calibration is sufficient to achieve a good level of precision. In such case, the full measurement data set is collected only for a small number of samples. The coefficients from this sample lot are then used for all other sensors of the lot. The individual sensor itself is calibrated only at two points, ideally taken at two different temperatures. From those two calibration points, the offset and slope are calculated and used to correct the initial capacitance ratio. For convenience, the firmware is programmed in a way that the user enters the theoretical ratios at the two calibration points and then the really measured ratios.

Figure 5:
2-Point Calibrated Intermediate Result



The 2-point calibrated intermediate result x_i is calculated as:

Equation 8:

$$x_i = \frac{x_{i@CCP2} - x_{i@CCP1}}{c_{i@CCP2} - c_{i@CCP1}} * (c_i - c_{i@CCP1}) + x_{i@CCP1}$$

$x_{i@CCP}$ Theoretical capacitance ratios at calibration points
$c_{i@CCP}$ Measured capacitance ratios at calibration points
c_i Actual capacitance ratio
x_i 2-point corrected capacitance ratio

This intermediate x_i result is then fed into the linearization polynomial for calculation of the final Z result. In case of a 1-point calibration the coefficients $x_{i@ccp1}$ and $c_{i@ccp1}$ simply need to be set to 0.

2 Linearize Firmware

2.1 Functionality

The Firmware is capable of performing these tasks:

- **Capacitance**
  - Output of original inverted capacitance ratio ci_ratio (c_ref/c_sense)
  - Output of x_i: 2-point corrected capacitance ratio
  - Output of Z: Linearized and temperature corrected result
  - Selectable sensor port
  - Single result, no support for combo sensors at the moment
  - Polynomial of 3rd order for the capacitance linearization
  - Coefficients are temperature corrected by a polynomial of 2nd order
  - => Total 12 coefficients
  - 2-point calibration => another 4 calibration values
  - Programmable limits for minimum/maximum of Z
- **Temperature**
  - Output of original inverted resistance ratio ri_ratio (r_ref/r_sense)
  - Output of 2-point corrected resistance ratio yi
  - Output of final temperature ϑ in °C
  - Selectable temperature sensor input (R0..R2 in PARA8)
  - Single result
  - Linearization by polynomial 3rd order
  - => 4 coefficients + 4 calibration values from 2-point calibration
  - Programmable limits for minimum/maximum of ϑ
- **Alarm Outputs**
  - 2 Alarm Outputs
  - Selectable alarm source (Z-result/theta, 1 bit in PARA8)
  - On/off threshold each
  - Selectable polarity
- **PDM**
  - Pulse0:= Capacitance is fixed
  - Pulse1:= Temperature is fixed
  - Each output is scalable via “scale” and “offset”
  - Limits can be set in LSB
- **Filter**
  - Selectable median 5 filter for capacitance
  - Selectable median 5 filter for temperature
  - Both to be activated in PARA8

2.2 Implemented Functions

2.2.1 Variables and Coefficients

$C_i = \frac{c_{ref}}{c_{sense}}$	Inverse capacitance ratio
$r_i = \frac{r_{ref}}{r_{measure}}$	Inverse resistance ratio
Z	Linearized and temperature compensated final result
$cc_{k,l}$	Coefficients for the capacitance polynomial, non-inverse
tc_k	Coefficients for the temperature polynomial, non-inverse
$x_{i@CCpn}$	Expected values for capacitance ratios with 2-point calibration
x_i	Inverse 2-point corrected capacitance ratio
y_i	Inverse 2-point corrected resistance ratio
ϑ	Linearized temperature result
$cci_{k,l}$	Coefficients for the inverse capacitance polynomial
tci_k	Coefficients for the inverse temperature polynomial
$y_{i@TCPn}$	Expected values for resistance ratios with 2-point calibration

2.2.2 Temperature Linearization

$y_i = \frac{y_{i@TCP2} - y_{i@TCP1}}{r_{i@TCP2} - r_{i@TCP1}} * (r_i - r_{i@TCP1}) + y_{i@TCP1}$	Inverse Linearized Resistance Ratio (2-Point Calibration)
$y_i = TQ(\vartheta) = \sum_{k=0}^{3} tci_k \vartheta^k$	Inverse Temperature Polynomial
$\vartheta = TP(y) = TQ^{-1} = \sum_{k=0}^{3} tc_k * \frac{1}{{y_i}^k}$	Temperature Polynomial

2.2.3 Sensor Linearization

$x_i = \frac{x_{i@TCP2} - x_{i@TCP1}}{c_{i@TCP2} - c_{i@TCP1}} * (c_i - c_{i@TCP1}) + x_{i@TCP1}$	Inverse Linearized Capacitance Ratio (2-Point Calibration)
$x_i = CQ(z, \vartheta) = \sum_{k=0}^{3} \sum_{l=0}^{2} cc_{k,l} z^k \vartheta^l$	Inverse Capacitance Polynomial
$Z = CP(x_i, \vartheta) = CQ^{-1} = \sum_{k=0}^{3} \sum_{l=0}^{2} cc_{k,l} \frac{1}{x_i{}^k} \vartheta^l$	Capacitance Polynomial

3 Linearization with Evaluation Kit Software

3.1 Introduction

The following exercise should give you an understanding how to use the PCap04 Evaluation Kit to linearize a sensor. The goal in this approach is to linearize a simple capacitive potentiometer. After linearization, the sensor and PCap04 together should build up an absolute position sensor. The linearized output will represent the absolute angle position in degree. How to set up the Eval Kit and evaluation software is not part of this document. This topic is covered in the user guide manual "PCap04_UG000358_1-00". Be sure you are familiar with this user guide before continuing the exercise.

3.2 Hardware Test Setup

Be sure your eval kit is set up correctly and ready to run. Replace the discrete capacitance between PC0 and PC1 with a capacitive potentiometer. The proposed value for this exercise would be in the range of 1 pF to 31 pF. Keep the wires to the sensor short and thin to reduce the influence of noise.

Figure 6:
Sensor Setup

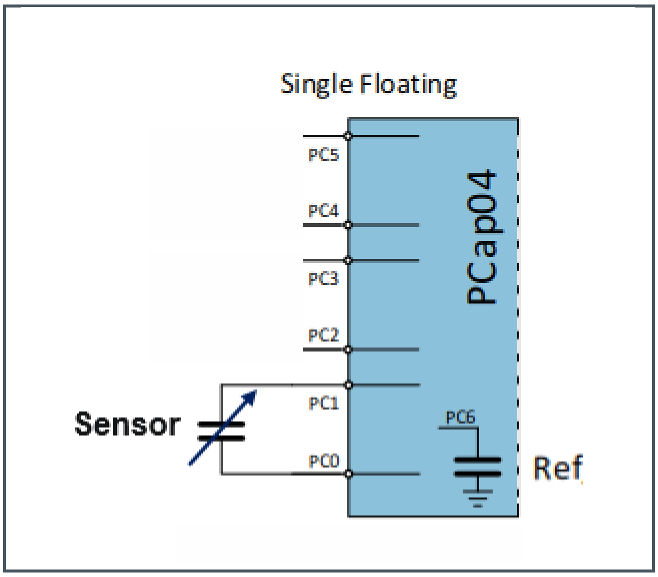


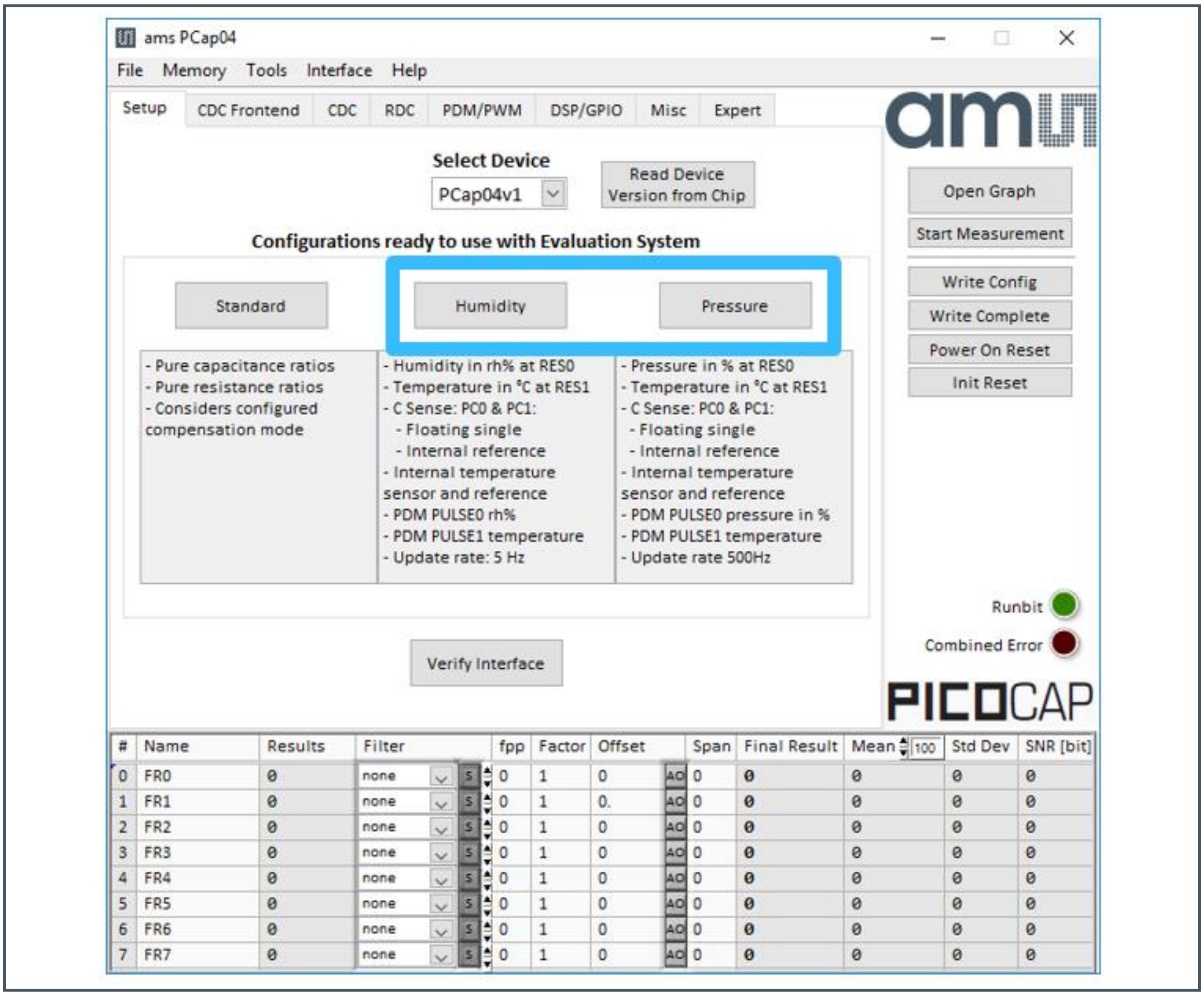
Figure 6 shows how to connect the capacitive potentiometer.

3.3 Software Linearization Tool

3.3.1 Linearized Firmware

Your hardware setup should now be ready. Power up your kit and start PCap04 evaluation software. The PCap04 evaluation software includes tools to linearize your sensor. In order to do this you have to load a firmware with linearized functionally on PCap04. The fastest way to do so, is to use one of the quick load buttons “Humidity” or “Pressure” on the main screen.

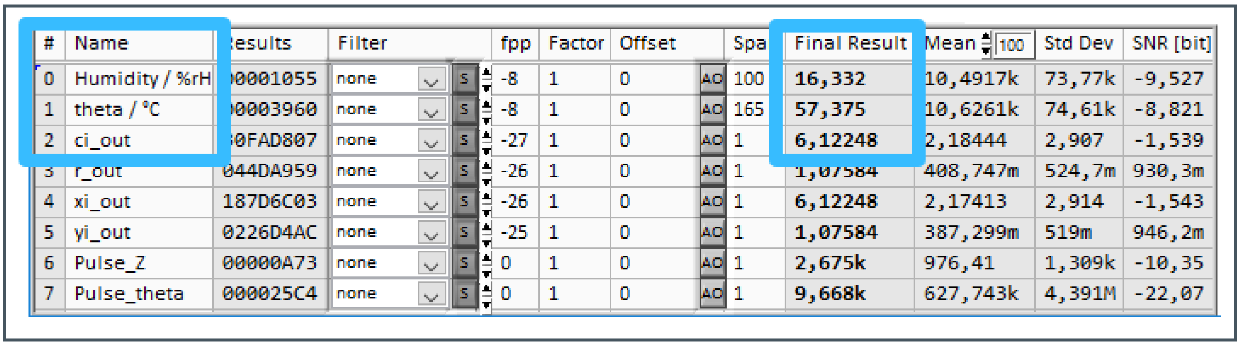
Figure 7 :
Setup Screen



In this case, we have selected the Humidity button. You may notice that immediately some namings and settings have changed.

Figure 8 shows the name and final result of the current measurement. Here it is configured for humidity sensors.

Figure 8 :
Setup Screen (Name and Final Result Columns)



#	Name	esults	Filter		fpp	Factor	Offset		Spa	Final Result	Mean 100	Std Dev	SNR [bit]
0	Humidity / %rH	)0001055	none	S	-8	1	0	AO	100	16,332	10,4917k	73,77k	-9,527
1	theta / °C	)0003960	none	S	-8	1	0	AO	165	57,375	10,6261k	74,61k	-8,821
2	ci_out	30FAD807	none	S	-27	1	0	AO	1	6,12248	2,18444	2,907	-1,539
3	r_out	044DA959	none	S	-26	1	0	AO	1	1,07584	408,747m	524,7m	930,3m
4	xi_out	187D6C03	none	S	-26	1	0	AO	1	6,12248	2,17413	2,914	-1,543
5	yi_out	0226D4AC	none	S	-25	1	0	AO	1	1,07584	387,299m	519m	946,2m
6	Pulse_Z	00000A73	none	S	0	1	0	AO	1	2,675k	976,41	1,309k	-10,35
7	Pulse_theta	000025C4	none	S	0	1	0	AO	1	9,668k	627,743k	4,391M	-22,07

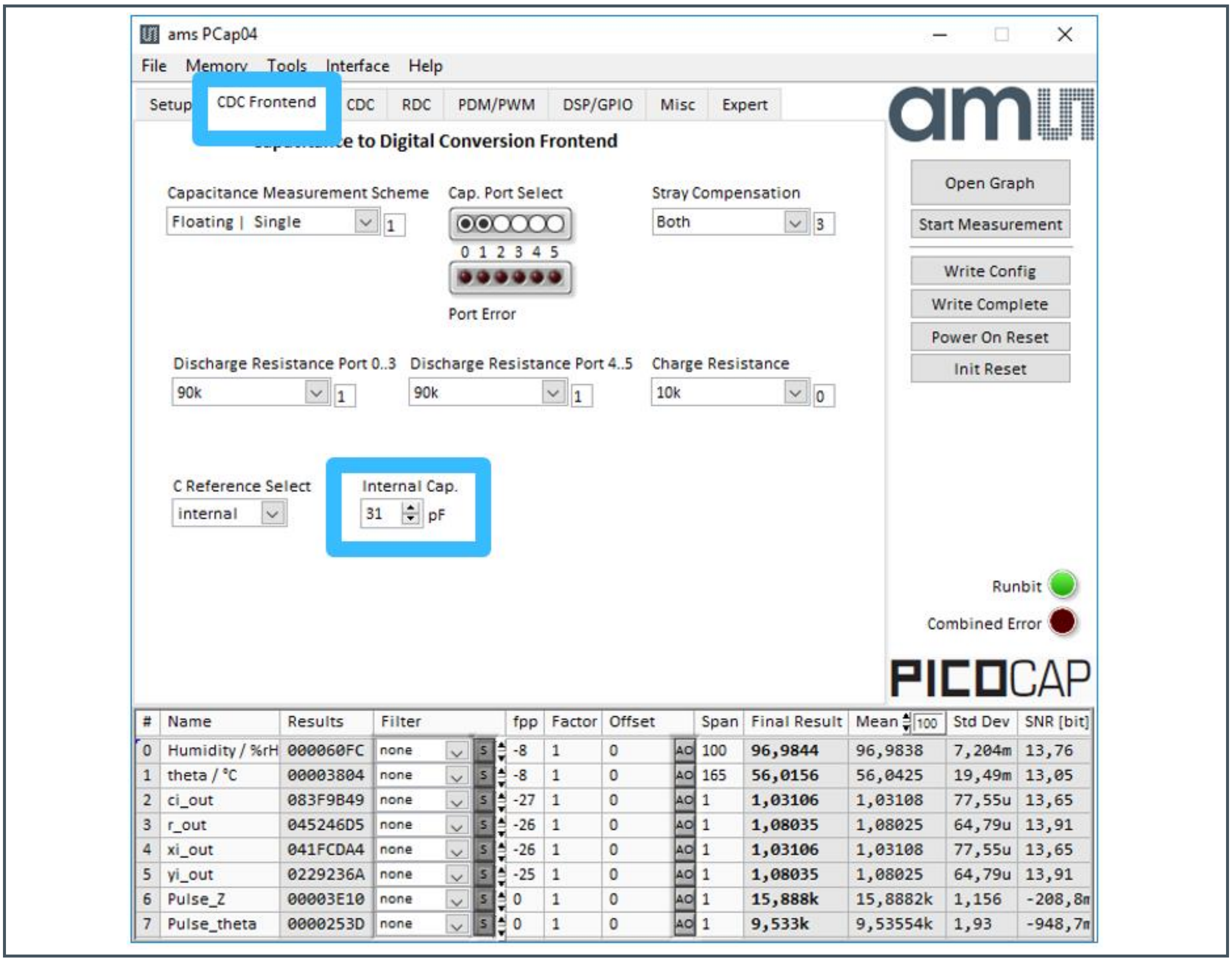
#0 “Humidity / %rH”	is the name for the linearized final result of the sensor.
#1 “theta/ °C”	is the name for final result of the temperature.
#2 “ci_out”	is the ratio between sensor and reference capacitance.

Our goal is to have a final linearized result in degree. Therefore, we change the name in column #0 to “Angle [°]”.Click the cell to modify it. For now leave the others as is.

3.3.2 Sensor Configuration

Now we have to bring sensor and reference into a proper ratio. Therefore, select the “CDC Frontend” tab.

Figure 9 :
CDC Frontend Tab



The settings in this window should be as shown in picture above. Adjust the internal reference capacitor “Internal Cap.” to the nominal value of the sensor. In this case 31 pF. After this, you are ready to start measurements by pushing the “Start Measurements” button.

Try now to make a 360° turn on the potentiometer. On column 3 under “Final Result” values should now be changing during the rotation. At the end of the turn, the value should be nearly the same as at the beginning. This is due to the typical design of the capacitive potentiometer.

Figure 10:
Capacitive Potentiometer

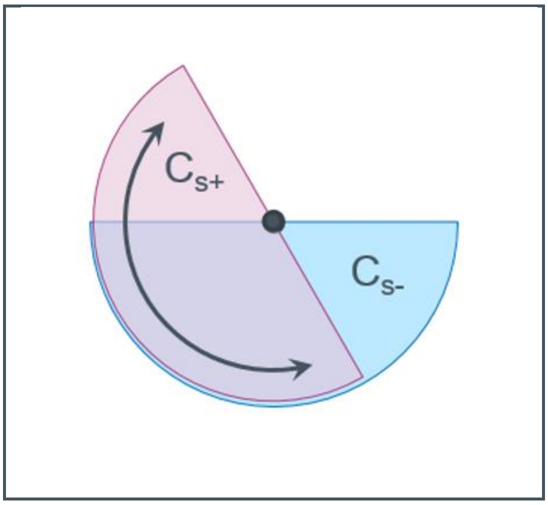
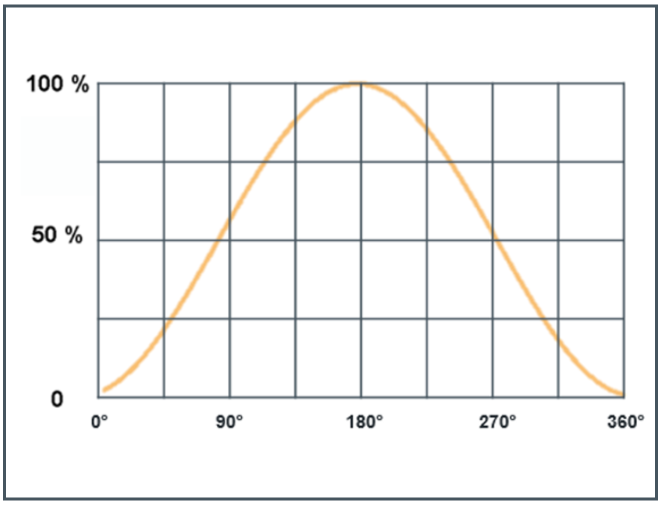


Figure 11:
Non-Linear Output

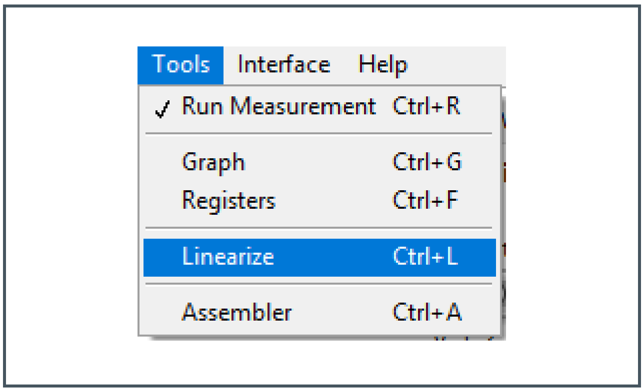


The sensor shows a non-linear output over the complete turn. After linearization, we expect to see a linear output. However, it makes sense only to linearize a turn from 0° to 180° as the values from 180° to 360° are just inverted.

3.3.3 Linearization Tool

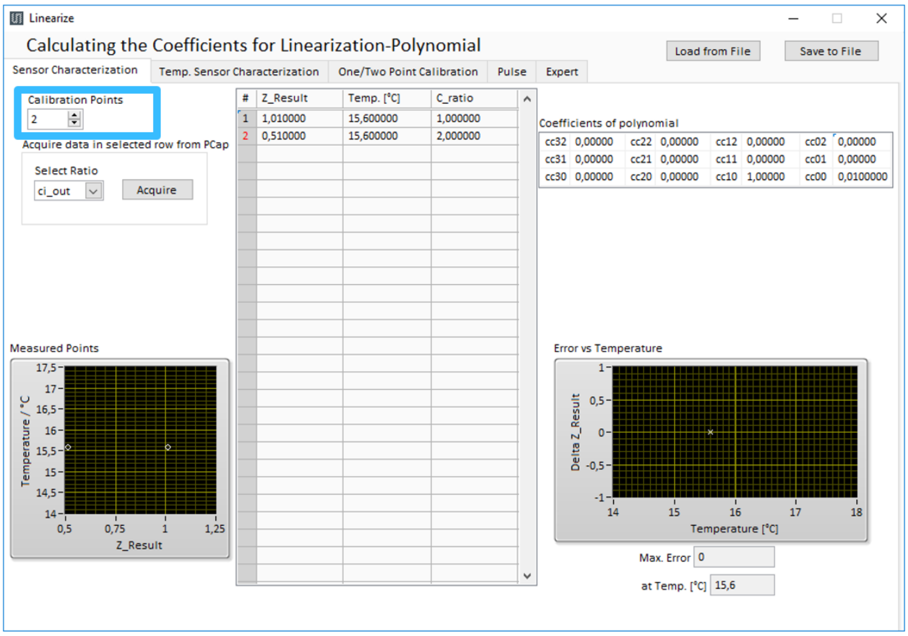
Now we can use the linearization tool. Go to “Linearize” under “Tools”.

Figure 12:
Tools Menu



The linearized window should appear as can be seen below.

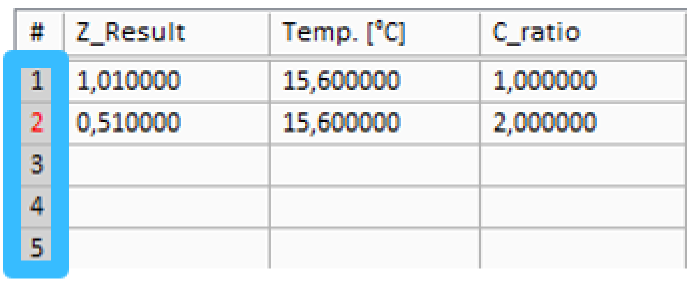
Figure 13 :
Linearize Screen



First, we need to know how many points we would like to calibrate. This is a question of precision and time. More points can give you a more linear result in the end, but it will take longer to acquire. Therefore, it depends on the requirements of the application. For our position sensor application, we chose 5 calibration points. We have to set this number in "Calibration Points".

If done so, you will recognize that the table has grown to five rows:

Figure 14:
Calibration Points



#	Z_Result	Temp. [°C]	C_ratio
1	1,010000	15,600000	1,000000
2	0,510000	15,600000	2,000000
3			
4			
5			

Z_Result	Linearized absolute value
Temp.[°C]	Actual temperature during measurements
C_ratio	Measured ratio between sensor and reference capacitance

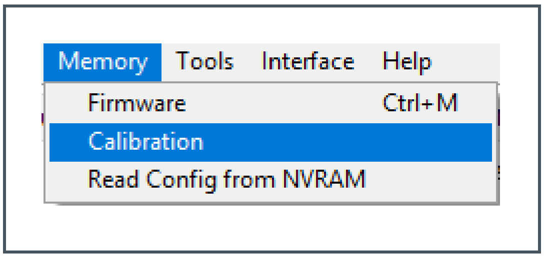
Now you can start the linearization process by turning your potentiometer to the lowest value. This position will be our start position for 0° angle. Write this value in the first row under “Z_Result”. Next will be the actual ambient temperature value. This value comes under “Temp[°C]”. After this, you have to push the “Acquire” button. This will add the measured ratio to the first calibration point of the table. Repeat this procedure for all five calibration points. Here we have acquired 0, 45, 90, 135 and 180°. You can select any prefered angel. Just make sure the potentiometer is in the same position as you enter in Z_Result. Do not forget to push “Acquire” for each!

Figure 15:
Linearization Data

#	Z_Result	Temp. [°C]	C_ratio
1	0,000000	24,000000	1,078350
2	45,000000	24,000000	1,309955
3	90,000000	24,000000	1,971716
4	135,000000	24,000000	4,182966
5	180,000000	24,000000	8,064481

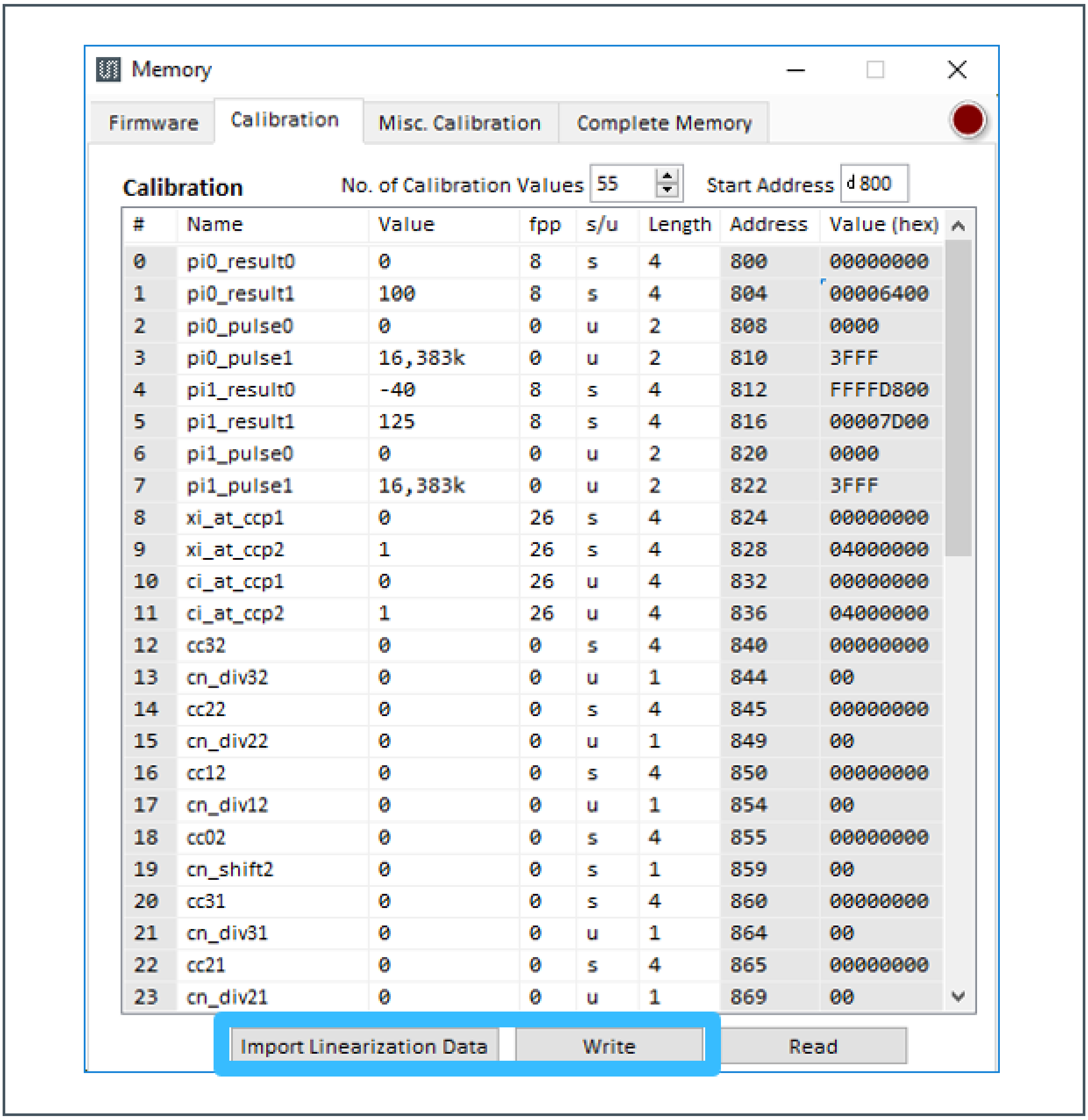
Now go to “Calibration” under “Memory”

Figure 16:
Memory Menu



Here you have to import and write the linearization data in the memory. Therefore, push the “import Linearization Data” button to import values in the calibration table. The next step is to write those values into the memory of the PCap04 by pushing the “Write” button.

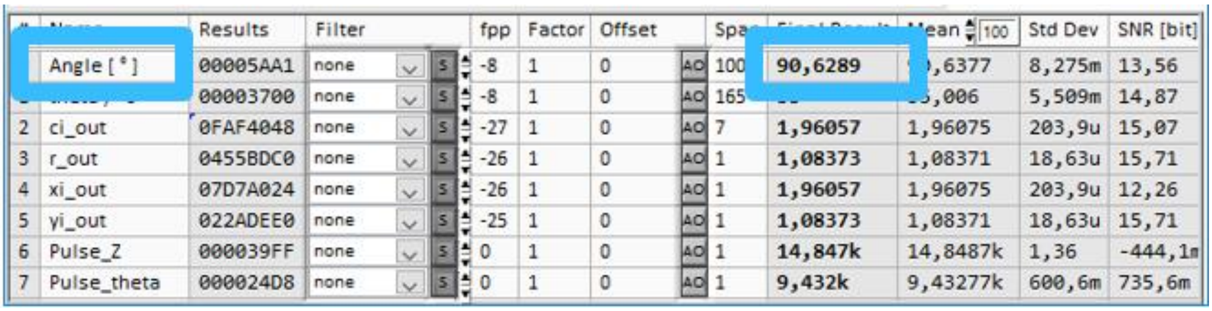
Figure 17 :
Calibration Menu



#	Name	Value	fpp	s/u	Length	Address	Value (hex)
0	pi0_result0	0	8	s	4	800	00000000
1	pi0_result1	100	8	s	4	804	00006400
2	pi0_pulse0	0	0	u	2	808	0000
3	pi0_pulse1	16,383k	0	u	2	810	3FFF
4	pi1_result0	-40	8	s	4	812	FFFFD800
5	pi1_result1	125	8	s	4	816	00007D00
6	pi1_pulse0	0	0	u	2	820	0000
7	pi1_pulse1	16,383k	0	u	2	822	3FFF
8	xi_at_ccp1	0	26	s	4	824	00000000
9	xi_at_ccp2	1	26	s	4	828	04000000
10	ci_at_ccp1	0	26	u	4	832	00000000
11	ci_at_ccp2	1	26	u	4	836	04000000
12	cc32	0	0	s	4	840	00000000
13	cn_div32	0	0	u	1	844	00
14	cc22	0	0	s	4	845	00000000
15	cn_div22	0	0	u	1	849	00
16	cc12	0	0	s	4	850	00000000
17	cn_div12	0	0	u	1	854	00
18	cc02	0	0	s	4	855	00000000
19	cn_shift2	0	0	s	1	859	00
20	cc31	0	0	s	4	860	00000000
21	cn_div31	0	0	u	1	864	00
22	cc21	0	0	s	4	865	00000000
23	cn_div21	0	0	u	1	869	00

With this, you have finished the linearization and you will have the absolute and linearized result.

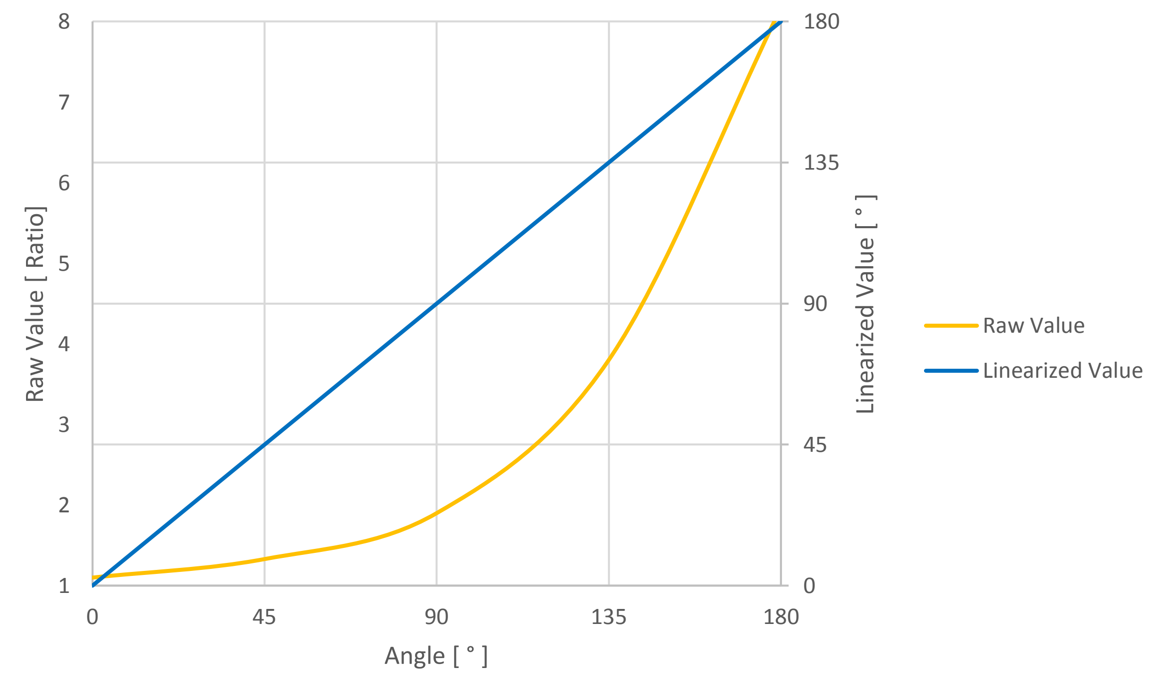
Figure 18 :
Absolute and Linearized Result



#	Name	Results	Filter	fpp	Factor	Offset	Span	Final Result	Mean 100	Std Dev	SNR [bit]
	Angle [°]	00005AA1	none	-8	1	0	100	90,6289	[illegible],6377	8,275m	13,56
[illegible]	[illegible]	00003700	none	-8	1	0	165	[illegible]	[illegible],006	5,509m	14,87
2	ci_out	0FAF4048	none	-27	1	0	7	1,96057	1,96075	203,9u	15,07
3	r_out	0455BDC0	none	-26	1	0	1	1,08373	1,08371	18,63u	15,71
4	xi_out	07D7A024	none	-26	1	0	1	1,96057	1,96075	203,9u	12,26
5	yi_out	022ADEE0	none	-25	1	0	1	1,08373	1,08371	18,63u	15,71
6	Pulse_Z	000039FF	none	0	1	0	1	14,847k	14,8487k	1,36	-444,1m
7	Pulse_theta	000024D8	none	0	1	0	1	9,432k	9,43277k	600,6m	735,6m

The chart in Figure 19 shows a comparison between the raw value (ci_out) vs the linearized value.

Figure 19:
Raw Value vs. Linearized Value



If you compare now to the yellow raw plot with the blue linearized plot, you can see that it clearly shows a linear behavior. In addition, the values directly converted to their reference unit.

4 Summary / Results

As shown in the position sensor example, that linearization is mandatory in situations where you require corrected absolute values. PCap04 linearized firmware and evaluation software together offer a complete solution for accomplishing this. No additional micro controller is required for these kind of calculations.

5 Revision Information

Changes from previous version to current revision v1-00	Page
Initial Version	

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

6 Legal Information

RoHS Compliant & ams Green Statement

RoHS Compliant: The term RoHS compliant means that ams AG products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ams Green (RoHS compliant and no Sb/Br): ams Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents ams AG knowledge and belief as of the date that it is provided. ams AG bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. ams AG has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. ams AG and ams AG suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Headquarters

ams AG

Tobelbader Strasse 30

8141 Premstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Please visit our website at www.ams.com

Buy our products or get free samples online at www.ams.com/Products

Technical Support is available at www.ams.com/Technical-Support

Provide feedback about this document at www.ams.com/Document-Feedback

For sales offices, distributors and representatives go to www.ams.com/Contact

For further information and requests, e-mail us at ams_sales@ams.com