Product Document

Published by
ams OSRAM Group

Application Note

AS5x47y

Zero Position Programming in Motor Control Applications

AS5047D

AS5047P

AS5147

AS5147P

AS5247

Content Guide

1 General Description

This application note describes different approaches of zero position programming of the absolute encoder AS5x47y. The alignment of the zero angle in combination with a brushless EC-Motor (electric commutated) system is discussed in detail. Especially the Multipoint Calibration mode in direct combination with the MCU (motor control unit), generates good results.

The suggested methods could be used for evaluation purpose, as well as for mass production of EC-Motor systems.

1.1 Why zero angle alignment has to be done?

The AS5x47y angular encoder gives an absolute angular position, based on the angel position of a magnet above (or below) the sensor. In terms of angle orientation, the magnet mounts randomly on the shaft of the motor. Therefore, zero position programming is required.

As a second reason an EC-Motor is as well an absolute system within one pole pair segment. To be able to gain the best torque and efficiency performance, it is necessary to align the two absolute systems (motor and sensor) once perfectly.

2 How to program Zero Position

In this section detailed information of Zero Position registers and implemented function are given.

2.1 Relevant register structure

Within the sensor IC two registers (ZPOSM 0x0016, ZPOSL 0x0017) are prepared to store the 14bit Zero Position value. This angular value defines the new zero output position. The register could be written several times. Programming itself (date are permanently stored within the sensor IC) is possible just once. Default register content for both registers is 0x0000.

Figure 1 shows the register apportionment of the ANGLECOM register. To assign the values to the ZPOSM and ZPOSL registers, the 14bit value has to be separated into two groups.

The LSB part (bits [5:0]) has to be written into ZPOSL, the MSB part (bits [13:6]) has to be written into ZPOSM.

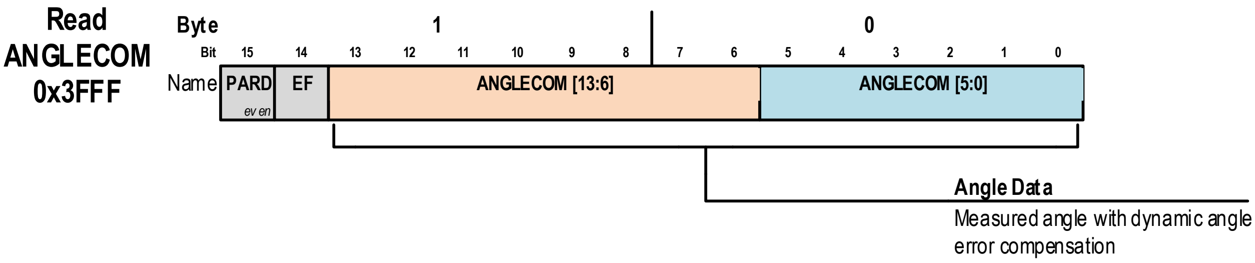
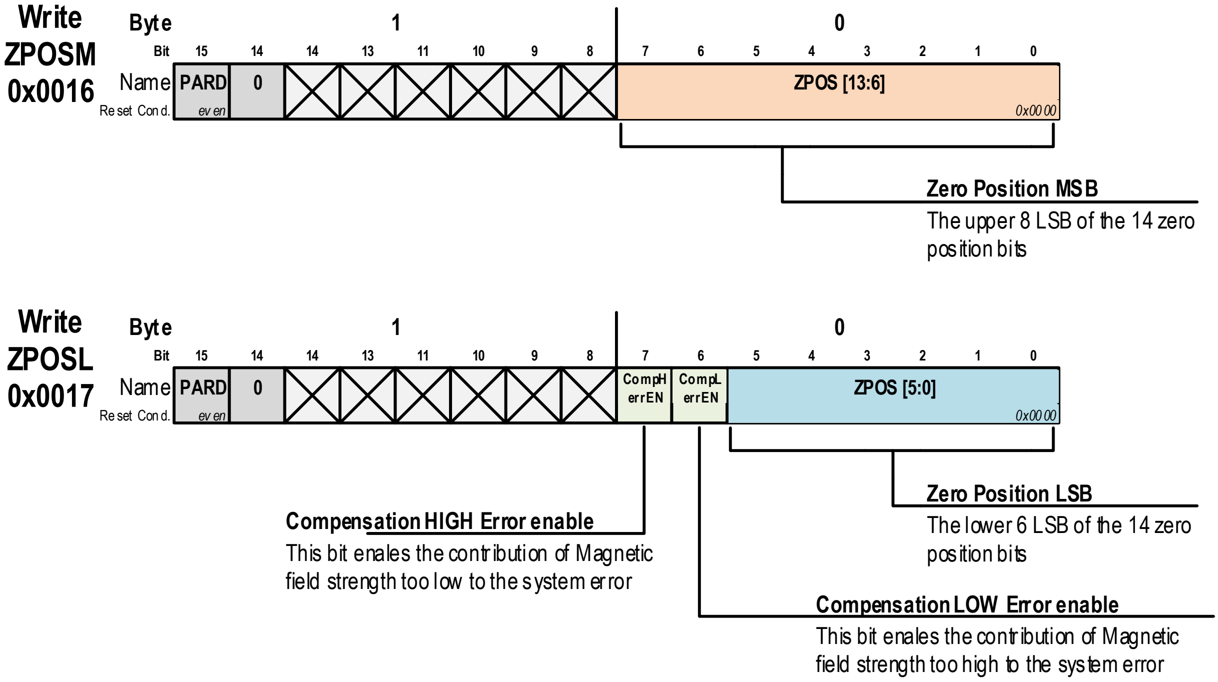


Figure 1: Angle output register ANGLECOM with 14bit Ange Data + Error Flag bit + Paridy bit.

Figure 2 shows in detail the register arrangement of the two Zero Position registers. The coloring is directly related to Figure 1.



Zero Position MSB
The upper 8 LSB of the 14 zero position bits

Zero Position LSB
The lower 6 LSB of the 14 zero position bits

Compensation HIGH Error enable
This bit enales the contribution of Magnetic field strength too low to the system error

Compensation LOW Error enable
This bit enales the contribution of Magnetic field strength too high to the system error

Figure 2: Register arrangement of ZPOSM and ZPOSL register.

2.2 ZPOS programming example

Figure 3 shows the output angel in default configuration. Register content of ZPOSM and ZPOSL is 0x0000 (default condition).

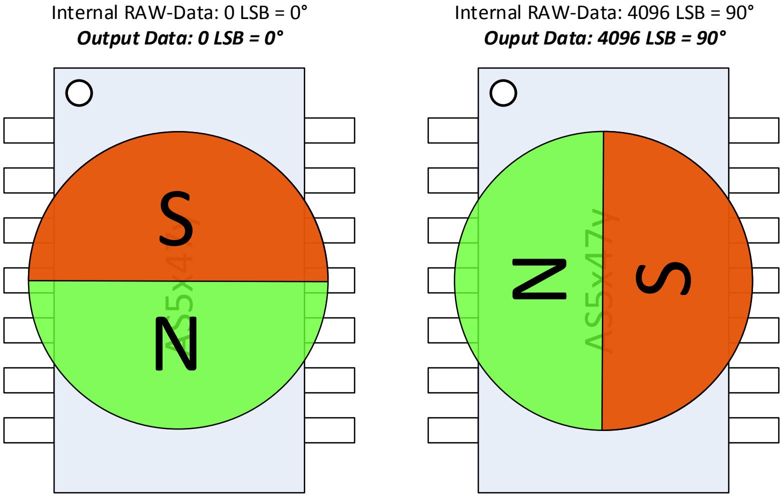

Internal RAW-Data: 0 LSB = 0°
Output Data: 0 LSB = 0°
Internal RAW-Data: 4096 LSB = 90°
Ouput Data: 4096 LSB = 90°
S
N
AS5x47y
N
S
AS5x47y

Figure 3: Angle output in default configuration.

In the following example, the Zero Position of the output angle is shifted about 30° in CCW (counter clockwise) direction.

Figure 4 shows the output angle with a shift of the Zero Position about 30° in CCW direction (-1365 LSB).

$$\textit{RAW-Data: 0x3FFF @ new Zero Pos} = 15019\text{ LSB} = \text{0x3AAB} = \underbrace{11\ 1010\ 10}_{\text{0x00EA}}\underbrace{10\ 1011}_{\text{0x002B}}$$

ZPOSM: 0x0016 – 0x00EA *ZPOSL*: 0x0017 – 0x002B

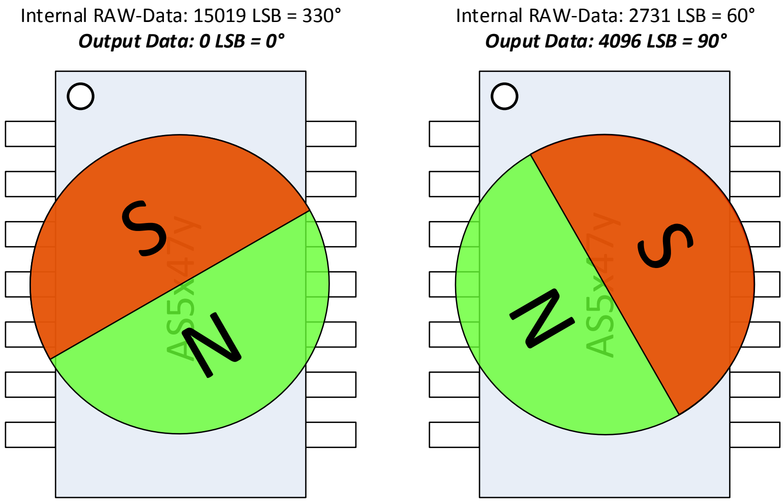

Internal RAW-Data: 15019 LSB = 330°
Output Data: 0 LSB = 0°
Internal RAW-Data: 2731 LSB = 60°
Ouput Data: 4096 LSB = 90°
S
N
AS5x47y
N
S
AS5x47y

Figure 4: Angle output shifted about 30°.

3 Zero Angle Setting in Motor Control applications

This chapter focuses on applications that use an AS5x47y as feedback sensor for closed loop motor control. The following suggested methods are valid for most EC-Motors (electric commutated motors like BLDC and PMSM).

Figure 5 shows a typical setup configuration for a closed loop motor application. The MCU uses the absolute angle information of the *UVW* outputs, to define the starting phase of the motor. In addition, the MCU takes high-resolution incremental pulses of the ABI interface. This is required for smooth vector commutation.

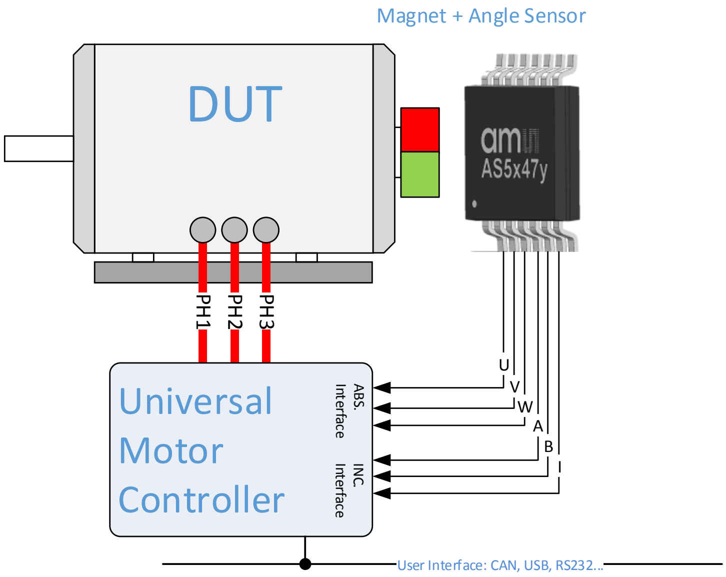


Figure 5: Basic application Setup, Motor + Angular Sensor + MCU

3.1 Single Point Calibration

The Single Point Calibration is the easiest and fastest way to set the zero angle position in combination with a MCU unit.

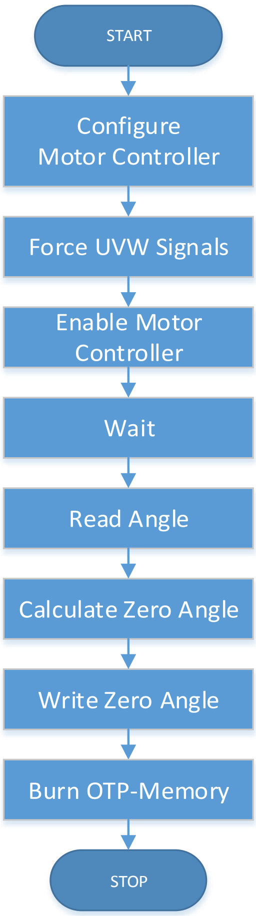


Figure 6: State diagram of the Single Point Calibration process.

1. ***Configure MCU***: Configure motor controller as in the final application. That means e.g. setting of pole pares, current direction, current limit, voltage limit, used feedback signals (UVW = Hall). Use at least nominal current (better max. current) of the motor as limit

1. ***Force UVW signals:*** Force hall inputs of the MCU to a defined state. Instead of the encoder, an external source should force the UVW inputs to: U = 1, V = 0, W = 1.

   This is the same pattern, as the sensor would generate at angular position 0° (please check Table 1).

   If full access to the MCU is given, no additional stimulation of UVW inputs is required. The MCU should force the phase current pattern related to hall input pattern <101>.

2.

3. ***Enable MCU:*** Enable the outputs of the MCU. Current will flow through the motor windings with respect to the hall input setting. A static magnetic field vector, generated by the stator windings, occurs. The magnetic field vector of the rotor will align immediately with the stator field vector. That means, the rotor will settle to a particular position regarding the stator field.

4. ***Wait:*** Wait as long as the rotor stands still. This depends mainly on the centrifugal mass of the rotor and takes usually about 1 second or less.

5. ***Read Angle:*** Read the actual position of the 0x3FFF register. This angular value gives the exact angle of the stator vector, with the defined hall configuration (U=1, V=0, W=1). To reduce the noise level on the 14bit angular output, several samples should be taken and averaged (e.g. 1000 readings). That reduces the noise to a minimum and generate a highly accurate value for this particular position – $refAngle\_av_0$. Also small oscillations of the rotor are cancelled out due to the averaging (wait time could be reduced).

6. ***Calculate Zero Angle:*** By the principle of an EC-Motor, it defines that the stator field slips forward referred to the rotor field. The switching point in block or trapezoidal commutation is 120°e (electric angle) before the rotor field. That means that in normal operation this hall (UVW) configuration has to be switched on 120°e before.

$$Angle_{ZeroLSB} = mod\left(refAngle\_av_0 - \frac{2^{14}}{Nr.\ of\ PP \cdot 3}\right) 2^{14}$$

7. ***Write Zero Angle:*** Write the calculated 14bit zero angle value into the two zero angle registers (as described in chapter 2.1).

8. ***Burn OTP-Memory:*** To store the zero angle permanently in the OTP registers (one-time programmable), the burn procedure has to be started. For detailed description please check the section *“Burn and Verification of the OTP Memory”* in the AS5x47y datasheet.

3.2 Multipoint Calibration

Instead of one calibration point, this method takes several calibration segments within one rotation. Therefore, it is possible to find a better calibration point as with the single point approach.

Also this method requires a MCU.

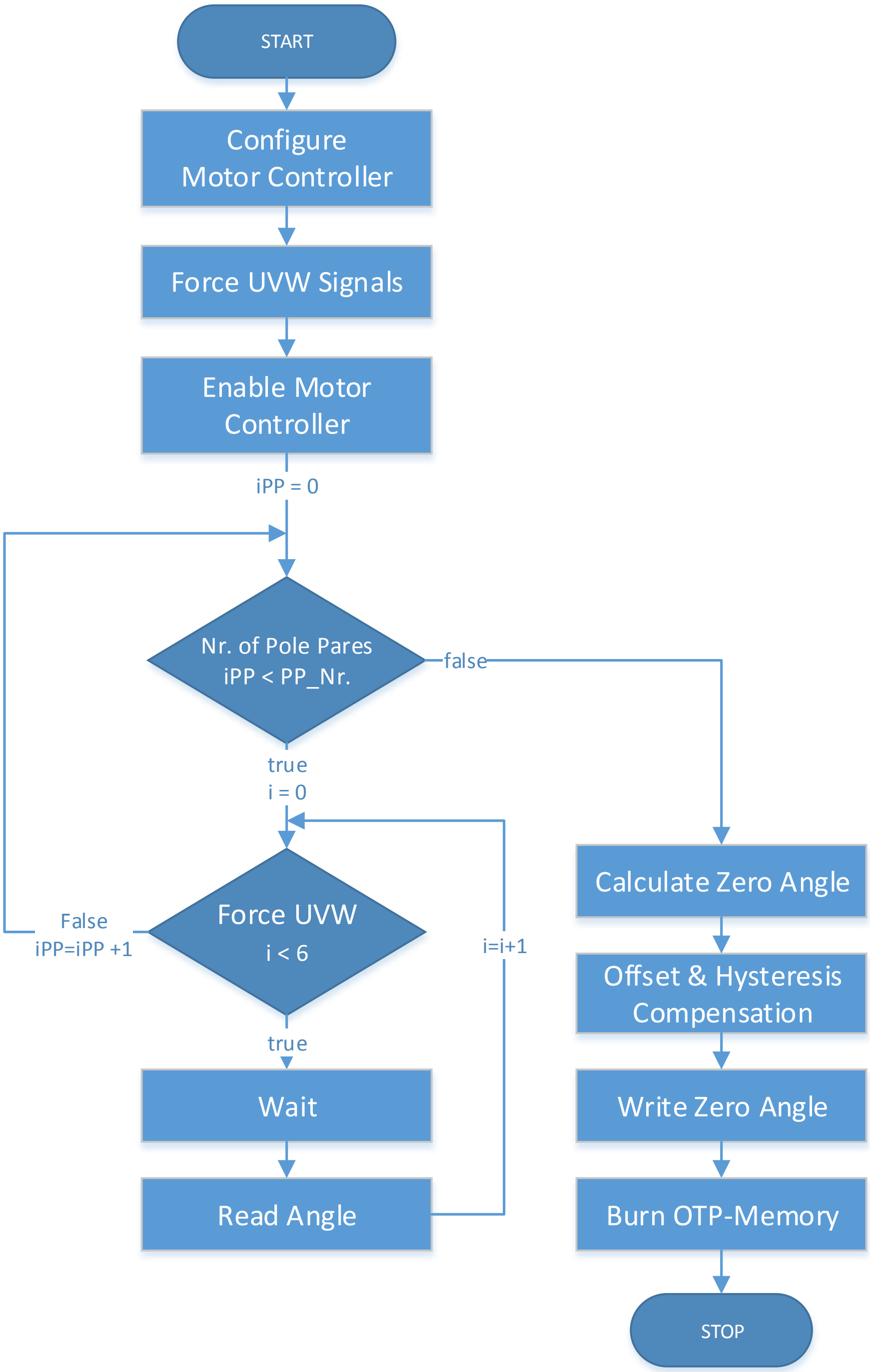


Figure 7: State diagram of Multipoint Calibration process.

2. ***Configure MCU***: Configure motor control unit as in the final application. That means e.g. setting of pole pares, current direction, current limit, voltage limit, used feedback signals (UVW = Hall, ABI). Use at least nominal current (better max. current) of the motor as limit.

3. ***Force UVW signals:*** Force hall inputs of the MCU to a defined state. Therefore, disconnected the *UVW* pins of the sensor (pins 8-10) from the MCU. Instead of the encoder, an external source should force the UVW inputs to: U = 1, V = 0, W = 1.

   This is the same pattern, as the sensor would generate at angular position 0° (please check Table 1).

   If full access to the MCU is given, no additional stimulation of UVW inputs is required. The MCU should force the phase current pattern related to hall input pattern <101>.

4. ***Enable MCU:*** Enable the outputs of the MCU. Current will flow through the motor windings with respect to the hall input setting. A static magnetic field vector, generated by the stator windings, occurs. The magnetic field vector of the permanent magnets on the rotor will align immediately with the stator field vector. That means, rotor will settle to a particular position regarding the stator field.

5. ***Pole Pare Loop:*** The number of Pole Pairs of the motor define the number of loop segments within a 360° rotation.

6. ***UVW Loop:*** Step through all six segments within the pole pair. Read angle position at every segment. This will end up in $i_x = Nr\_of\_PP \cdot 6$ reference points.
7. Table 1 shows UVW settings.

8. ***Wait:*** Wait as long as the rotor stands still. This depends mainly on the centrifugal mass of the rotor and takes usually about 1 second or less.

9. ***Read Angle:*** After rotor settling, read the position information of the sensor IC. This position value gives the exact angle value of the stator vector, with the appropriate hall configuration. To reduce the noise level on the 14bit angular output, read several samples and take the average (e.g. 1000 readings). Also small oscillations of the rotor are cancelled out due to the averaging (wait time could be reduced). Repeat reading of the angle at every settling position. The average values of the measured angles should be stored in an array $segAngle_{av}\,[0 \ldots i_x]$ for further calculations

10. ***Calculate Zero Angle:*** It is defined by the principle of an EC-Motor, that the stator field always slips forward referred to the rotor field. The switching point in block or trapezoidal commutation is 120°e (electric angle) before the rotor field. In normal operation the MCU takes care (based on the rotor angle feedback), that the next stator current pattern is switched on 120°e before the rotor field vector.

$$\boldsymbol{zeroAngle_{LSB} = mod\left(segAngle_{av_0} - \frac{2^{14}}{Nr.\ of\ PP\ \cdot 3}\right) 2^{14}} \qquad \text{3-1}$$

11. ***Offset & Hysteresis Compensation:*** The aim of the offset shift calculation is to minimize the peak INL error of the sensor system. The error distributes symmetrically over the full 360° rotation.

    For offset calculation, a reference angle table $refAngle_{LSB}\,[0 \dots i_x]$ is required - Equation 3-2.

$$\boldsymbol{refAngle_{LSB}[0 \dots i_x] = mod\left(segAngle_{av_0} + \frac{2^{14}}{Nr.\ of\ PP \cdot 6} \cdot i\right) 2^{14}} \qquad 3\text{-}2$$

The difference between the measured segment angles and the calculated reference angles, give an error angle for each segment - Equation 3-3.

$$\boldsymbol{errAngle_{LSB}[0 \dots i_x] = segAngle_{av}[0 \dots i_x] - refAngle_{LSB}[0 \dots i_x]} \qquad 3\text{-}3$$

To calculate the angle correction value, detect the difference between maximum and minimum value of the error array. This correction will guarantee a symmetric error distribution over the full 360° rotation.

$$\boldsymbol{corAngle_{LSB} = \max(errAngle_{LSB}[\dots]) - \frac{\max\ (errAngle_{LSB}[\dots]) - \min(errAngle_{LSB}[\dots])}{2}} \qquad 3\text{-}4$$

The implemented hysteresis function, which is necessary to avoid incorrect direction changes on incremental/hall outputs, should also be taken into account to calculate the optimal zero angle value.

Convert the configured hysteresis setting into a 14bit angular value, to be compatible to the further the calculation.

$$\boldsymbol{compHyst_{LSB} = \frac{HystSetting_{14bit}}{2}} \qquad 3\text{-}5$$

Add the offset correction and hysteresis compensation value to the previously calculated zero angle value.

$$\boldsymbol{zeroAngleCorr_{LSB} = zeroAngle_{LSB} + corAngle_{LSB} + compHyst_{LSB}} \qquad 3\text{-}6$$

12. ***Write Zero Angle:*** Write the calculated 14bit zero angle value into the two zero angle registers (as described in chapter 2.1).

13. ***Burn OTP-Memory:*** To store the zero angle permanently in the OTP registers (one-time programmable), the burn procedure has to be started. For detailed description please check the section *"Burn and Verification of the OTP Memory"* in the AS5x47y datasheet.

3.3 Setup description

The hardware setup for Zero Position alignment is similar to the normal operation setup.

To enable read and write access to the sensor, SPI-connection is mandatory. Additionally it is required, to be able to force constant currents trough the motor windings. Ideally the MCU (motor controller unit) itself should handle this task.

An appropriate way is, to set the hall inputs of the controller to constant states. Motor controller will set the corresponding current pattern.

If full access to the MCU is available (embedded system), motor currents can be defined directly out of the MCU unit.

3.3.1 UVW Signals forced by Programmer Unit

Use this procedure if no direct access to the MCU is available (remote systems). The hall inputs of the MCU has to be stimulated, as the sensor would do it during one rotation.

The *UVW* outputs of the AS5x47y are the counterpart of discrete hall elements, which are still very common in low cost EC-Motor systems. Most of general-purpose motor controllers have digital Hall-Inputs available. Therefore, connect *UVW* outputs of the sensor directly to the motor controller.

In programming mode, disconnect *UVW* outputs of the angular sensor from the MCU. Connect stimulation signals instead. The user has to be able to force the same pattern as *UVW* is creating during a complete 360° rotation (shown in Figure 8).

Config Nr.	Abs. Angle Point	Abs. Angle Point	Pattern U V W
0	0°	0 LSB_{14}	1 0 1
1	60°	2731 LSB_{14}	1 0 0
2	120°	5461 LSB_{14}	1 1 0
3	180°	8192 LSB_{14}	0 1 0
4	240°	10923 LSB_{14}	0 1 1
5	300°	13659 LSB_{14}	0 0 1

Table 1: Related output angle to UVW pattern.

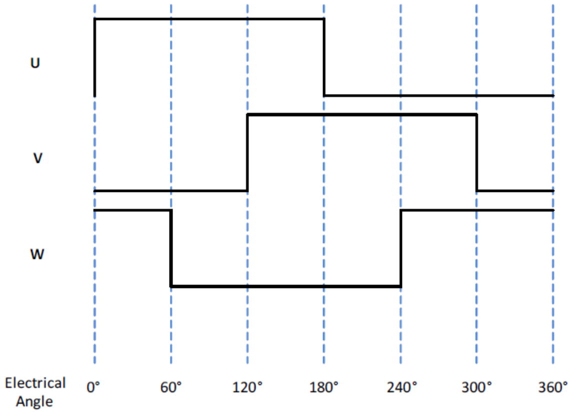


Figure 8: UVW - Output patter over 360°.

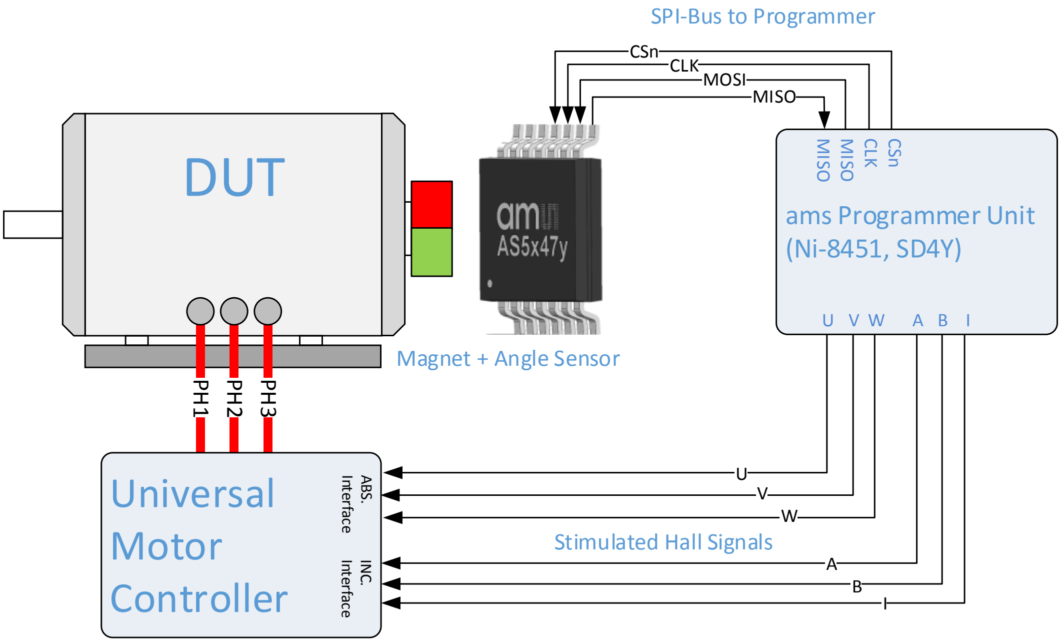


Figure 9: Zero Position Programming setup + Stimulation Signals (UVW, ABI).

3.3.2 Embedded programming Mode in MCU

If full access to the MCU is given, no additional stimulation signals out of the programmer box are required. It is required to generate the same current pattern, as it would occur during block commutation. Therefore, the controller itself requires an emulation of the UVW-Inputs.

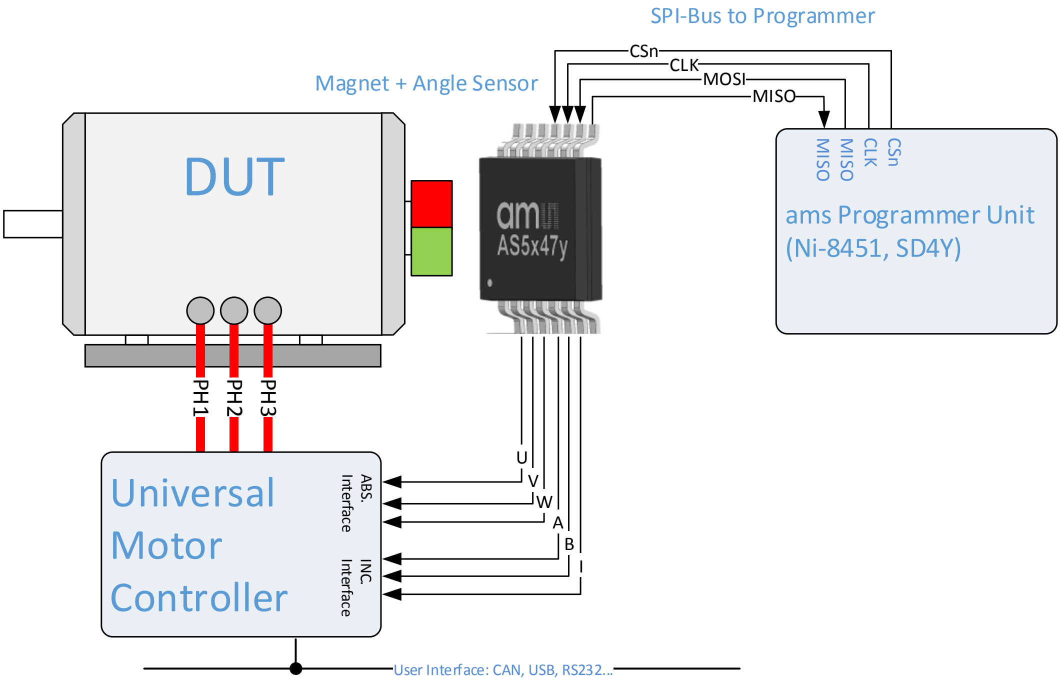


Figure 10: Zero Position Programming setup, full access given to the MCU - no simulation required.

3.4 Example: Zero Position Programming with a 4-PP Motor

This example represents the complete Multipoint Calibration procedure according to section 0.

A 4 pole pair (8 pole) PMSM-Motor was used. Therefore 24 angle segment positions are given. A general purpose MCU forced the current patterns. As no direct access to the MCU was available, the approach of UVW stimulation by the programmer unit was elected.

Table 2 shows the measured angle date related to the current pattern given by the MCU (*segAngle/LSB*). All values in LSB are reference to 14bit angle resolution. Column *refAngle* and *errAngle* were calculated out of *segAngle* data, based on the formulas given in section 0.

av Index	Ralated Angle/°	segAngle/LSB	refAngle/LSB	errAngle/LSB
0	0	3761.1	3761.1	0.0
1	15	4443.2	4443.8	-0.6
2	30	5134.3	5126.5	7.8
3	45	5810.7	5809.1	1.6
4	60	6501.1	6491.8	9.3
5	75	7183.6	7174.5	9.1
6	90	7869.2	7857.1	12.0
7	105	8553.7	8539.8	13.9
8	120	9246.3	9222.5	23.8
9	135	9922.8	9905.1	17.7
10	150	10610.7	10587.8	22.9
11	165	11289.8	11270.5	19.3
12	180	11971.0	11953.1	17.9
13	195	12650.9	12635.8	15.1
14	210	13337.8	13318.5	19.3
15	225	14009.2	14001.1	8.1
16	240	14693.7	14683.8	9.9
17	255	15371.0	15366.5	4.5
18	270	16051.8	16049.1	2.7
19	285	348.1	347.8	0.3
20	300	1037.2	1030.5	6.7
21	315	1711.7	1713.1	-1.4
22	330	2399.1	2395.8	3.3
23	345	3079.0	3078.5	0.6

Table 2: Measured angle values on segment positions + error calculation.

Calculate zero angle position:

4 pole pair motor is used –> $Nr.\ of\ PP = 4$

$$zeroAngle_{LSB} = mod\left(segAngle_{av_0} - \frac{2^{14}}{Nr.\ of\ PP \cdot 3}\right) 2^{14} = mod\left(3761.1 - \frac{2^{14}}{4 \cdot 3}\right) 2^{14} = 2395.8\ LSB_{14}$$

$$1° = 45.5\ LSB_{14}$$

Calculate offset correction value:

$$corAngle_{LSB} = \max(errAngle_{LSB}[\ldots]) - \frac{\max\ (errAngle_{LSB}[\ldots]) - \min(errAngle_{LSB}[\ldots])}{2} = 23.8 - \frac{23.8-(-1.4)}{2}$$
$$= 11.2\ LSB$$

In Figure 11 shows the basic error angel and the compensated error angle. The curves are representing the INL (integral non-linearity) of the sensor (ideally 0 LSB), measured with the motor itself as reverence.

The red curve is shifted about the offset correction value $corAngle$, which gives a symmetric error distribution around 0. Reduced peak error is the result.

The $corAngle$ value is a good indicator of overall system quality. Defects in magnet, sensor, motor and MCU could be detected, by an appropriate limit check of the $corAngle$ value.

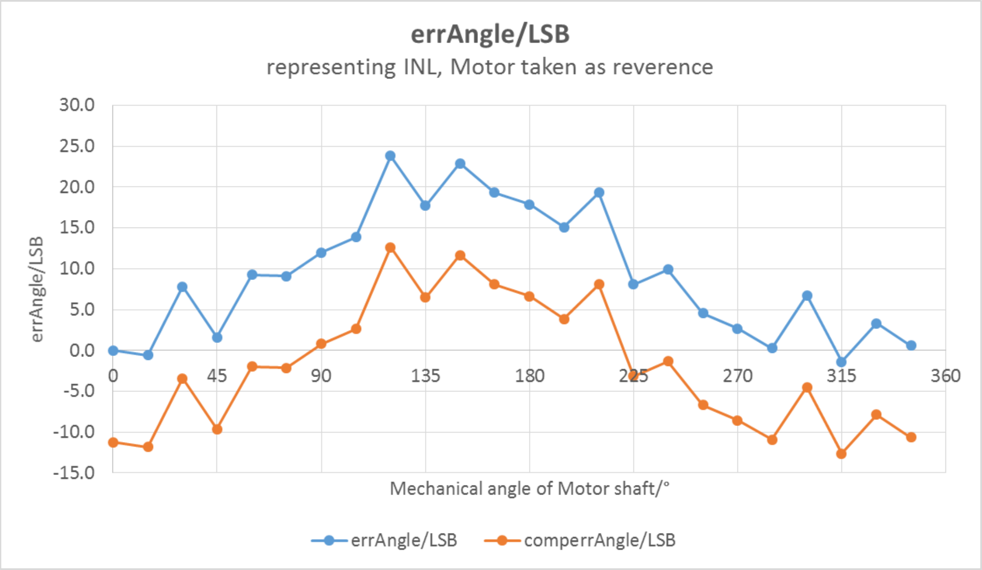


Figure 11: Plot of errAngle date with and without offset shift.

Calculate hysteresis correction:

In this example an AS5147P was used.

Hysteresis setting 01 = 2 LSB11 - related to 11 bit

= 16 LSB14 - related to 14 bit

$$compHyst_{LSB} = \frac{HystSetting_{14bit}}{2} = \frac{16}{2} = 8\ LSB_{14}$$

Calculate corrected zero angle position value:

$$zeroAngleCorr_{LSB} = zeroAngle_{LSB} + corAngle_{LSB} + compHyst_{LSB} = 2395.8 + 11.2 + 8 = \underline{2415\ LSB_{14}}$$

Writing new zero angle position value to ZPOSM and ZPOSL register:

New zeroAngle value: 2415 LSB = 0x096F = $\underbrace{00\ 1001\ 01}_{\text{0x0025}}\underbrace{10\ 1111}_{\text{0x002F}}$

ZPOSM: 0x0016 – 0x0025 *ZPOSL*: 0x0017 – 0x002F

Before the zero angel were stored permanently in the OTP registers, perform a test run of the motor. The whole system in combination with angle sensor + MCU + motor has to be tested.

A further possibility it to check the back EMF + *UVW* signals.

If everything is running as expected, start the burning and verification procedure. The determined zero angle position value permanently into the sensor.

4 Summary

This application note gives detailed information about our recommended Zero Position Programming procedure, in the field of EC-Motor commutation. Customers can use it for evaluation purpose, as well as in mass production (EOL test).

The Multipoint Calibration approach offers many benefits. Systematic error peaks will be reduced. In addition, the hysteresis is taken into account, to guarantee same motor performance in both rotation directions. Furthermore defects in the motor/sensor/driver system can be detected in an easy way, by checking the limits of the calculated parameters during Multipoint Calibration.

5 Ordering & Contact Information

Ordering Code	Description
Ordering Code	AS5x47y Select a Kit Type Select a Kit Variant

Buy our products or get free samples online at:

www.ams.com/ICdirect

Technical Support is available at:

www.ams.com/Technical-Support

Provide feedback about this document at:

www.ams.com/Document-Feedback

For further information and requests, e-mail us at:

ams_sales@ams.com

For sales offices, distributors and representatives, please visit:

www.ams.com/contact

Headquarters

ams AG
Tobelbaderstrasse 30
8141 Unterpremstaetten
Austria, Europe

Tel: +43 (0) 3136 500 0
Website: www.ams.com

6 Copyrights & Disclaimer

Information in this document is believed to be accurate and reliable. However, ams AG does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Applications that are described herein are for illustrative purposes only. ams AG makes no representation or warranty that such applications will be appropriate for the specified use without further testing or modification. ams AG takes no responsibility for the design, operation and testing of the applications and end-products as well as assistance with the applications or end-product designs when using ams AG products. ams AG is not liable for the suitability and fit of ams AG products in applications and end-products planned.

ams AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data or applications described herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.

ams AG reserves the right to change information in this document at any time and without notice.

7 Revision Information

Initial version 1-00

Changes from 1-00 (Jun-15) to current revision 1-01 (Jun-16)	Page
Wording	

Note: Page numbers for the previous version may differ from page numbers in the current revision.