Application Note: AS5048 SPI Sensor Readout

AS5048A

14-bit Rotary Position Sensor with Digital Interface

Table of Contents

Revision History

Revision	Date	Owner	Description
1.0	03.28.2013	rph	Initial Version
1.1	26.06.2013	rph	Updated Averaging Description

1. General Description

This Application Note explains how to communicate with the AS5048 14-bit Rotary Position Sensor. There are two versions of the AS5048 available:

Table 1:

AS5048 versions

Ordering Code	Digital Angle Interface
AS5048A	SPI
AS5048B	I2C

Find more information on our webpage:
http://ams.com/eng/Products/Magnetic-Position-Sensors

2. Reading the AS5048A using SPI

The AS5048A comes with SPI interface. Following example shows how to read the AS5048A.

Example:

```
#define SPI_CMD_READ 0x4000 // flag indicating read attempt
#define SPI_CMD_WRITE 0x8000 // flag indicating write attempt
#define SPI_REG_AGC 0x3ffd // agc register when using SPI
#define SPI_REG_MAG 0x3ffe // magnitude register when using SPI
#define SPI_REG_DATA 0x3fff // data register when using SPI
#define SPI_REG_CLRERR 0x1 // clear error register when using SPI
#define SPI_REG_ZEROPOS_HI 0x0016 // zero position register high byte
#define SPI_REG_ ZEROPOS_LO 0x0017 // zero position register low byte

void spiReadData()
{
      u16 dat; // 16-bit data buffer for SPI communication
      u16 magreg;
      ushort angle, agcreg;
      ubyte agc;
      ushort value;
      bit alarmHi, alarmLo;
```

```
/* Send READ AGC command. Received data is thrown away: this data comes
from the previous command (unknown)*/

dat = SPI_CMD_READ | SPI_REG_AGC;
dat |= spiCalcEvenParity(dat) << 15;
spiTransfer((u8*)&dat, sizeof(u16));

/* Send READ MAG command. Received data is the AGC value: this data comes
from the previous command (unknown)*/

dat = SPI_CMD_READ | SPI_REG_MAG;
dat |= spiCalcEvenParity(dat) << 15;
spiTransfer((u8*)&dat, sizeof(u16));
magreg = dat;

/* Send READ ANGLE command. Received data is the MAG value, from the
previous command */

dat = SPI_CMD_READ | SPI_REG_DATA;
dat |= spiCalcEvenParity(dat) << 15;
spiTransfer((u8*)&dat, sizeof(u16));
agcreg = dat;

/* Send NOP command. Received data is the ANGLE value, from the previous
command */

dat = 0x0000; // NOP command.
spiTransfer((u8*)&dat, sizeof(u16));
angle = dat >> 2;
}
if ((dat & 0x4000) || (agcreg & 0x4000) || (magreg & 0x4000))
{
/* error flag set - need to reset it */
dat = SPI_CMD_READ | SPI_REG_CLRERR;
dat |= spiCalcEvenParity(dat)<<15;
spiTransfer((u8*)&dat, sizeof(u16));
}
```

```
else
{
agc = agcreg & 0xff // AGC value (0..255)
value = dat & (16384 - 31 - 1); // Angle value (0.. 16384 steps)
angle = (value * 360) / 16384; // Angle value in degree (0..359.9°)
magnitude = magreg & (16384 - 31 - 1);
alarmLo = (agcreg >> 10) & 0x1;
alarmHi = (agcreg >> 11) & 0x1;
}
```

3. Sensor Output Averageing

The code example below shows an easy averaging algorithm to reduce noise of the angular output for angular values within one rotation and with no zero crossing of the output data.

Note: As 360° is the same as 0° this would cause an averaging error. To perform averaging of output data with more than one rotation refer to “Mean of circular quantities”.

Example:

```
#define average 100
for (count = average; count; count--)
{
            data = sensor_read(); //reading,computing sensor output
            position += data;
}
position = position / average;
```

4. Parity calculation

Following example shows how the parity of a 16bit unsigned integer can be calculated.

Example:

```
static u8 spiCalcEvenParity(ushort value)
{
      u8 cnt = 0;
      u8 i;
      for (i = 0; i < 16; i++)
      {
            if (value & 0x1) cnt++;
            value >>= 1;
      }
      return cnt & 0x1;
}
```

5. AS5048A SPI Register Map

The available registers for SPI communication of the AS5048A are listed in Figure 1.

Figure 1:
SPI Register Map

	Address hex	Name	Access Type	Bit Nr.	Symbol	Default	Description
Control and Error Registers	x0000	SPI NOP	R	13 : 0	NOP	0	No operation dummy information
	x0001	Clear Error Flag	R	13 : 3	not used	n.a.	Error Register. All errors are cleared by access
				2	Parity Error	0	
				1	Command Invalid		
				0	Framing Error		
	x0003	Programming Control	R/W	13 : 7	not used	0	Programming control register. Programming must be enabled before burning the fuse(s). After programming is a verification mandatory. See programming procedure.
				6	Verify		
				5	not used		
				4			
				3	Burn		
				2	reserved		
				1			
				0	Programming Enable		
Programmable Customer settings	x0016	OTP Register Zero Position Hi	R/W + Program	13 : 8	not used	0	Zero Position value high byte
				7	Zero Position <13>	0	
				:	:	:	
				0	Zero Position <6>	0	
	x0017	OTP Register Zero Position Low 6 LSBs	R/W + Program	13 : 6	not used	0	Zero Position remaining 6 lower LSB's
				5	Zero Position <5>	0	
				:	:	:	
				0	Zero Position <0>	0	
Readout Registers	x3FFD	Diagnostics + Automatic Gain Control (AGC)	R	13 : 12	not used	n.a.	Diagnostics flags
				11	Comp High	0	
				10	Comp Low	0	
				9	COF	0	
				8	OCF	1	
				7	AGC value<7>	1	Automatic Gain Control value. 0 decimal represents high magnetic field 255 decimal represents low magnetic field
				:	:	:	
				0	AGC value<0>	0	
	x3FFE	Magnitude	R	13	Magnitude<13>	0	Magnitude information after ATAN calculation
				:	:	:	
				0	Magnitude<0>	0	
	x3FFF	Angle	R	13	Angle <13>	0	Angle information after ATAN calculation and zero position adder
				:	:	:	
				0	Angle<0>	0	

6. Ordering Information

Table 2:
Ordering Information

Ordering Code	Description	comments
AS5048A-EK-AB-STM1.0	AS5048A Eval-Kit for stepper motor	SPI interface

Copyright

Disclaimer

Devices sold by ams AG are covered by the warranty and patent indemnification provisions appearing in its Term of Sale. ams AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. ams AG reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with ams AG for current information.

This product is intended for use in normal commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or lifesustaining equipment are specifically not recommended without additional processing by ams AG for each application. For shipments of less than 100 parts the manufacturing flow might show deviations from the standard production flow, such as test flow or test location.

The information furnished here by ams AG is believed to be correct and accurate. However, 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 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.

Contact Information

Headquarters

ams AG
Tobelbader Strasse 30
8141 Unterpremstaetten
Austria
T. +43 (0) 3136 500 0
For Sales Offices, Distributors and Representatives, please visit:
http://www.ams.com/contact