User Manual – AS5048-AB-v1.1

AS5048

14-bit Rotary Position Sensor with Digital Angle (Interface) and PWM Output

Table of Contents

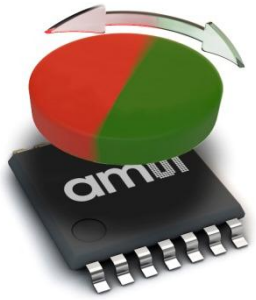
Revision History

Revision	Date	Owner	Description
1.0	01.10.2009		Initial revision
1.1	18.10.2013	azen	Updated to new template
1.2	14.1.2013	rph	Minor corrections in section 4.1

1 General Description

The AS5048 is an easy to use 360° angle position sensor with a 14-bit high resolution output. To measure the angle, only a simple two-pole magnet, rotating over the center of the chip, is required. The magnet may be placed above or below the IC. This is shown in Figure 1.

Figure 1:
Magnetic Position Sensor AS5048 + Magnet



2 The AS5048 adapter board

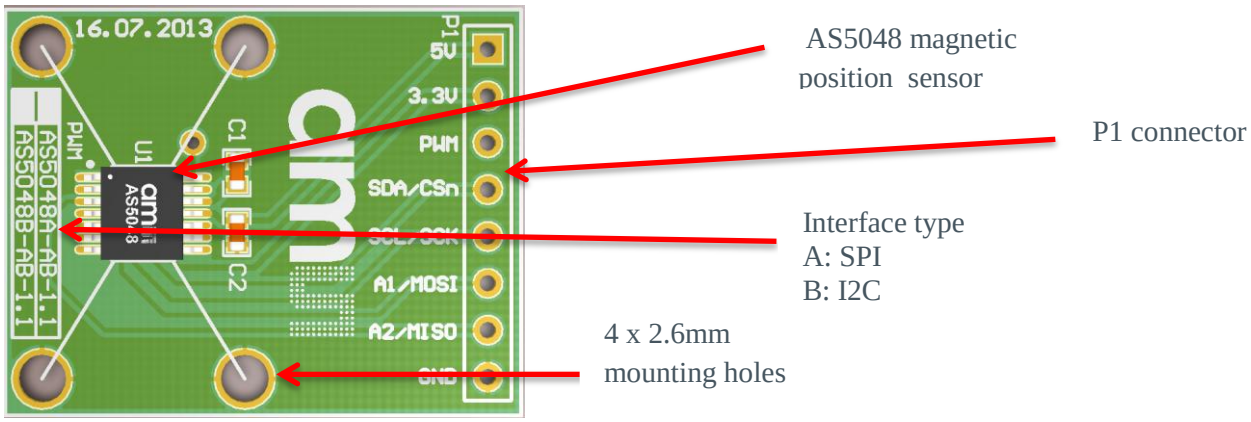
The AS5048 adapter board is a simple circuit allowing test and evaluation of the AS5048 magnetic position sensor quickly without building a test fixture or PCB.

2.1 Board description

The AS5048 Adapterboard is a simple circuit allowing test and evaluation of the AS5048 rotary encoder quickly without building a test fixture or PCB.

The PCB can be attached to a microcontroller or to the AS5048- Demoboard as external device.

Figure 2:
AS5048 Adapterboard



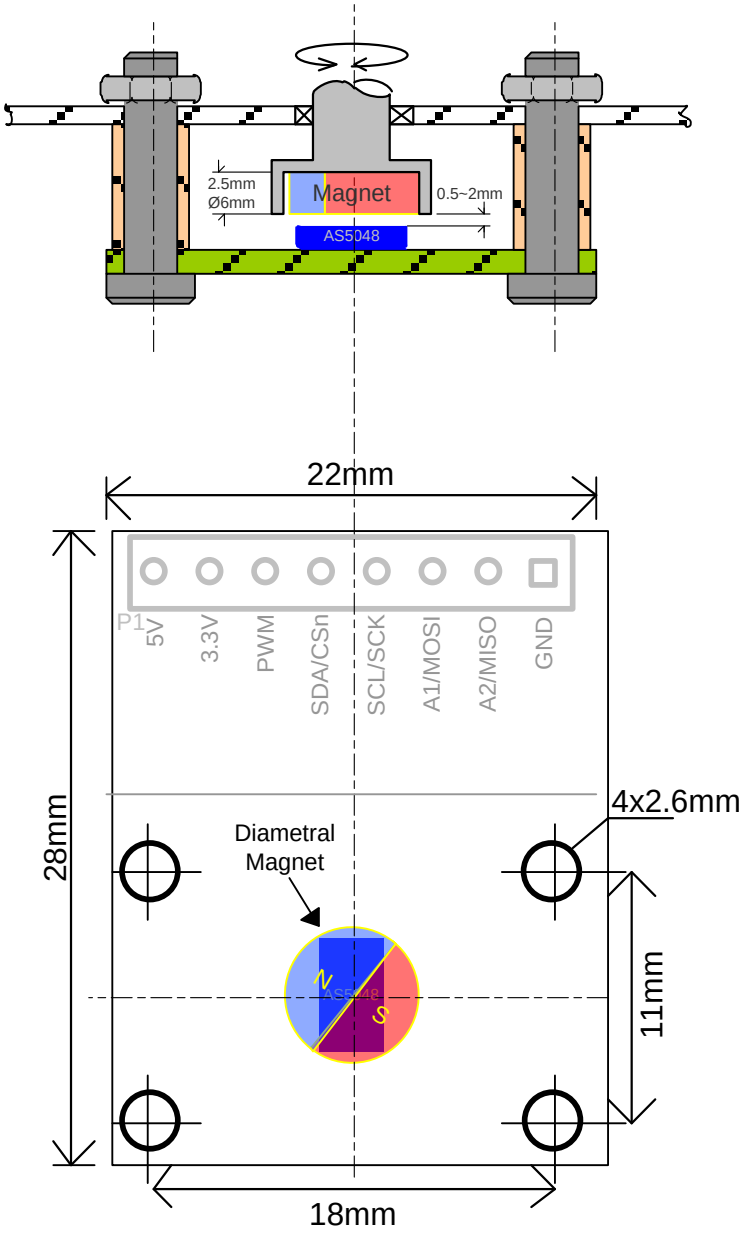
2.2 Mounting the AS5048 adapter board

A diametric magnet must be placed over on under the AS5048 position sensor, and should be centered on the middle of the package with a tolerance of 0.5mm.

The airgap between the magnet and the encoder casing should be maintained in the range 0.5mm~2mm. The magnet holder must not be ferromagnetic. Materials as brass, copper, aluminum, stainless steel are the best choices to make this part.

Figure 3:

AS5048 – AB - mounting and dimension



3 AS5048 adapter board and pinout

Figure 4:
AS5048 adapter board connectors and encoder pinout

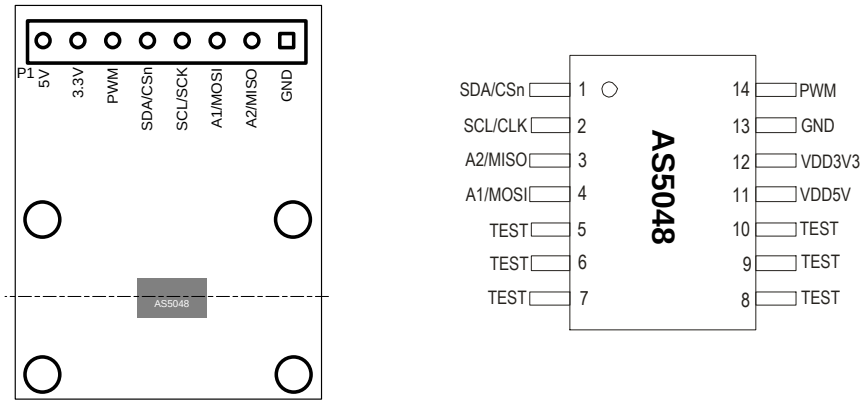


Table 1:
Pin description

Pin# Board	Pin# AS5048	Symbol Board	Description
P1 - 1	13	GND	Supply ground
P1 - 2	3	A2/MISO	SPI master in/slave out; shared with I2C address selection pin 2
P1 - 3	4	A1/MOSI	SPI master out/slave in; shared with I2C address selection pin 1
P1 - 4	2	SCL/SCK	SPI clock input; shared with I2C clock input
P1 - 5	1	SDA/CSn	SPI chip select-active low; shared with I2C data pin
P1 - 6	14	PWM	Pulse width modulation output
P1 - 7	12	3.3V	3V-Regulator output; internally regulated from VDD. Connect to VDD for 3V supply voltage
P1 - 8	11	5V	Supply voltage

4 Operation cases

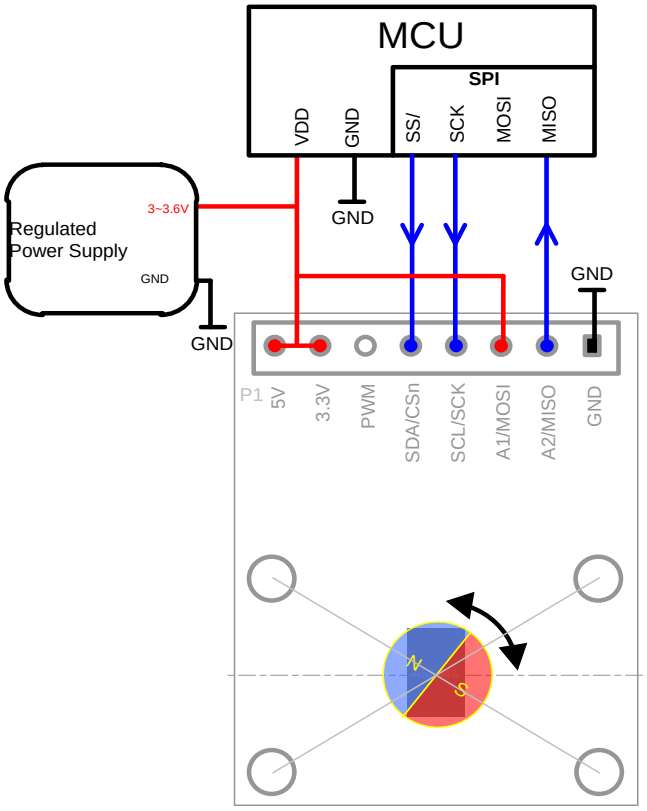
The most complete and accurate solution for a MCU to read the angle of a magnet is the SPI interface.

4.1 One Device SPI mode, unidirectional – 3 wire

The AS5048-AB can be directly connected to an industry standard SPI port of a microcontroller. The minimum connection requirement for unidirectional communication (angle + alarm values reading) between the microcontroller and the AS5048 are MISO, SCK, SS/.

The angle will be read at each 16-bit SPI transfer. See AS5048 datasheet register table, register 3FFFh.

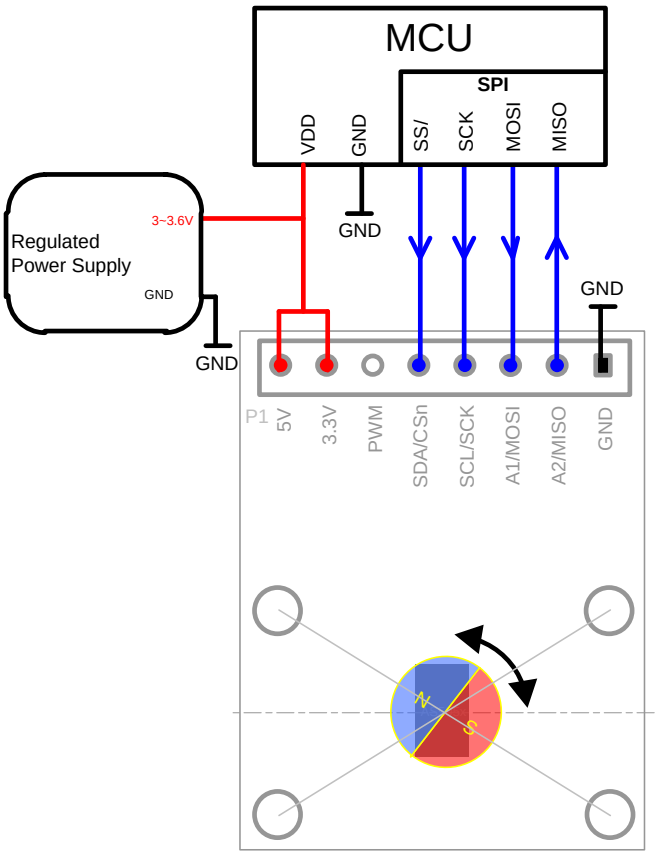
Figure 5:
Using the SPI Interface unidirectional with a microcontroller



4.2 One device SPI mode, bidirectional – 4 wire

If other registers than only angle values have to be read, or in order to write registers into the AS5048, the signal MOSI is necessary.

Figure 6:
Using the SPI Interface bidirectional with a microcontroller



4.3 Multi devices SPI Daisy chain mode

The AS5048 can be daisy chained, using 4 wires only for SPI communication.

In this configuration with n x encoders, the sequence will be processed as follow:

- MCU sets SS/ = 0
- MCU shifts n x 16-bit (e.g. READ command FFFFh) through the chain
- MCU sets SS/=1

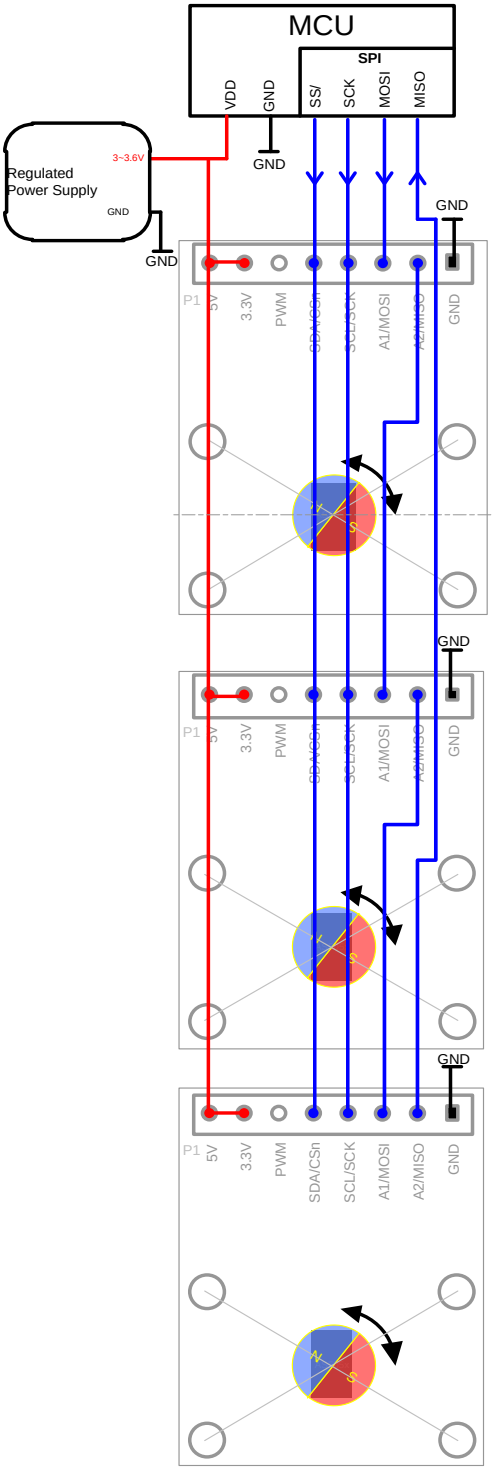
At that point all the n x encoders have received the READ command FFFFh.

- MCU sets SS/=0
- MCU shifts n x 16-bit (e.g. NOP command 0000h)
- MCU sets SS/=1

At that point the n x 16-bit received on MISO are the n x angle values.

Figure 7:

Multi Devices in Daisy chain mode



5 Firmware coding

The following source code fits the 4-Wire application
The function **void spiReadData()** reads/writes 4 values from the AS5048
- Send command READ AGC / Receive value unknown
- Send command READ MAG / Receive value AGC
- Send command READ Angle / Receive value MAG
- Send command NOP (no operation) / Receive value ANGLE

If a READ ANGLE only is necessary in a loop, the procedure can be reduced to one line:
- Send command READ Angle / Receive value Angle

The function **static u8 spiCalcEvenParity(ushort value)** is optional, it calculates the parity bit of the 16-bit SPI stream.

```
/*!
 ***************************************************************************
 *  Reads out chip data via SPI interface
 *
 *  This function is used to read out cordic value from chips supporting SPI
 *  interface.
 ***************************************************************************
 */
#define SPI_CMD_READ 0x4000     /*!< flag indicating read attempt when using SPI interface */
#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 */

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 precedent
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
precedent 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 precedent 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 precedent 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 */
```

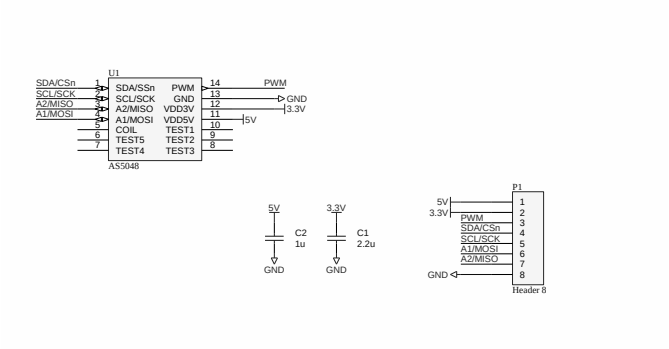
```
        dat  = SPI_CMD_READ | SPI_REG_CLRERR;
        dat |= spiCalcEvenParity(dat)<<15;
        spiTransfer((u8*)&dat, sizeof(u16));
    }
    else
    {
        agc = agcreg & 0xff                         // AGC value (0..255)
        value = dat & (16384 - 31 - 1);             // Angle value (0.. 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;
    }
}
/*!
 ****************************************************************************
 *  Calculate even parity of a 16 bit unsigned integer
 *
 *  This function is used by the SPI interface to calculate the even parity
 *  of the data which will be sent via SPI to the encoder.
 *
 *  \param[in] value : 16 bit unsigned integer whose parity shall be calculated
 *
 *  \return : Even parity
 *
 ****************************************************************************
 */
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;
}
/*!
 ****************************************************************************
 *  Calculate even parity of a 16 bit unsigned integer
 *
 *  This function is used by the SPI interface to calculate the even parity
 *  of the data which will be sent via SPI to the encoder.
 *
 *  \param[in] value : 16 bit unsigned integer whose parity shall be calculated
 *
 *  \return : Even parity
 *
 ****************************************************************************
 */
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;
}
```

6 AS5048-AB-Hardware

Following the schematic and layout of the Adapterboard can be found.

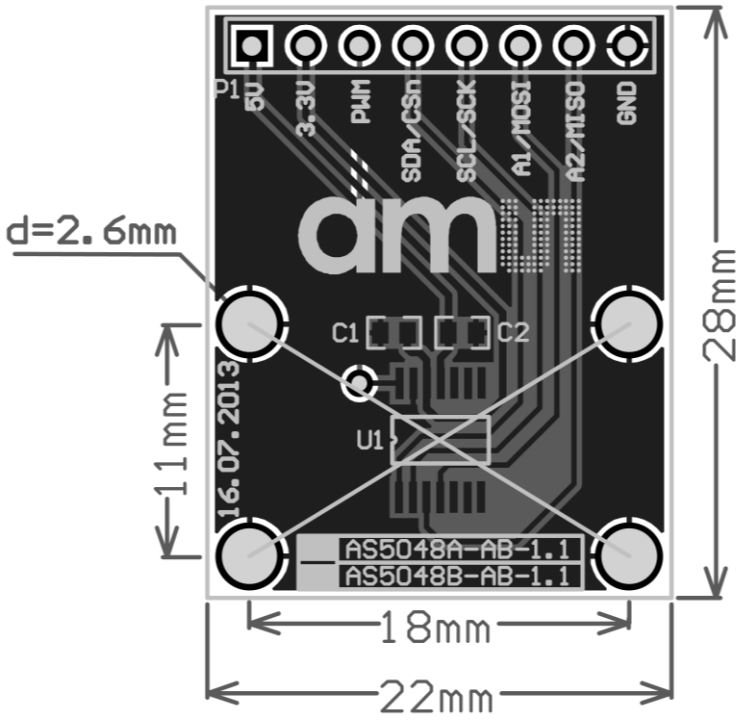
6.1 AS5048-AB-1.1 Schematics

Figure 8:
AS5048-AB-1.1 adapterboard schematics



6.2 AS5048 – AB – 1.1 PCB layout

Figure 9:
AS5048-AB-1.1 adapter board layout



7 Copyright

8 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. 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 commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by ams AG for each application. This Product is provided by ams “AS IS” and any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

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.

9 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