TLE5014SP16(D)

GMR-based Angle Sensor

User Manual

About this document

Scope and purpose

This document covers the TLE5014SP using the SSC protocol. It provides further information for what should be considered when using this sensor in an application and how to program it.

Intended audience

This document is aimed at experienced hardware and software engineers using the TLE5014SP iGMR angle sensor.

This documents contains information for the following devices:

Table 1 Derivatives covered by this user manual

Product Type	Marking	Ordering Code	Package	Comment
TLE5014SP16 E0001	014SP01	SP004232096	PG-TDSO-16	SSC Interface, single die
TLE5014SP16 E0002	014SP02	SP004531446	PG-TDSO-16	SSC Interface, single die
TLE5014SP16D E0002	014SPD02	SP004531452	PG-TDSO-16	SSC interface, dual die

Table of contents

1 Application Circuits

This chapter describes the different application circuits of the TLE5014SP.

For every use case, a decoupling capacitor C_D of 100nF is recommended. The capacitor shall be located as close as possible to the sensor supply pin. The load capacitor C_L shall not exceed the maximum specified value defined in the Datasheet. The DATA line is actively driven to high or low but the output driver is switched off once reaching the high state. Therefore, a pull-up resistor R_{PU} is recommended to maintain a stable high level.

In case of a high speed communication, an additional serial resistor in the range of 140Ω can be implemented in the DATA, SCK and CSQ line to avoid reflections and enhance communication reliability. In this case the user is responsible to verify that the intended communication speed can be reached in his specific setup.

1.1 Single Sensor

Figure 1 shows an application circuit with a single TLE5014SP. The TLE5014SP is connected to the microcontroller via a 3-pin SSC. Pins IFA and IFB are inputs and should be tied to a fixed define voltage. It is recommended to connect IFA to GND and IFB to V_{DD} via a serial resistor (R_{P1} e.g. 2.2kΩ). IFC is an output and shall be left opened.

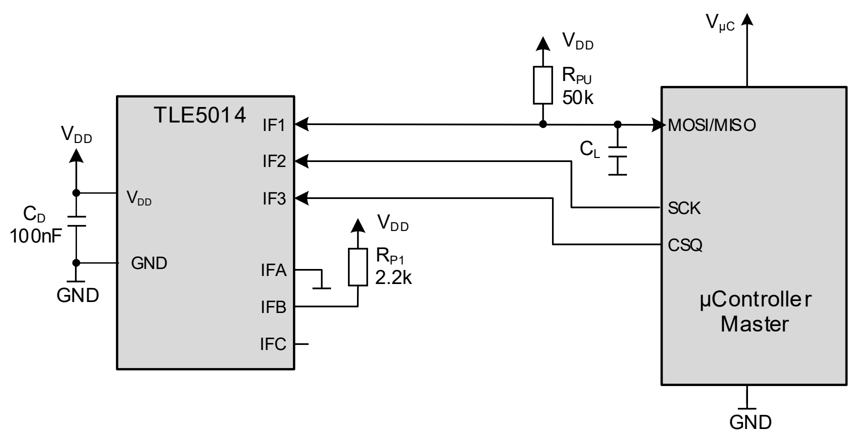

Figure 1 Application circuit for a single TLE5014SP

1.2 Bus Sensor Configuration

For redundancy purpose, two TLE5014SP can be used in a bus configuration. Two variants are possible, with a common DATA/SCK lines or with a common CSQ.

1.2.1 Common DATA/SCK lines

Figure 2 shows an application circuit where two TLE5014SP are connected in bus mode configuration. The two TLE5014SP share the same DATA and SCK lines. The microcontroller is responsible to assert only one CSQ at a time. Similar to the use case for a single sensor IFA and IFB shall be tied to a fixed defined voltage. IFC shall be left opened.

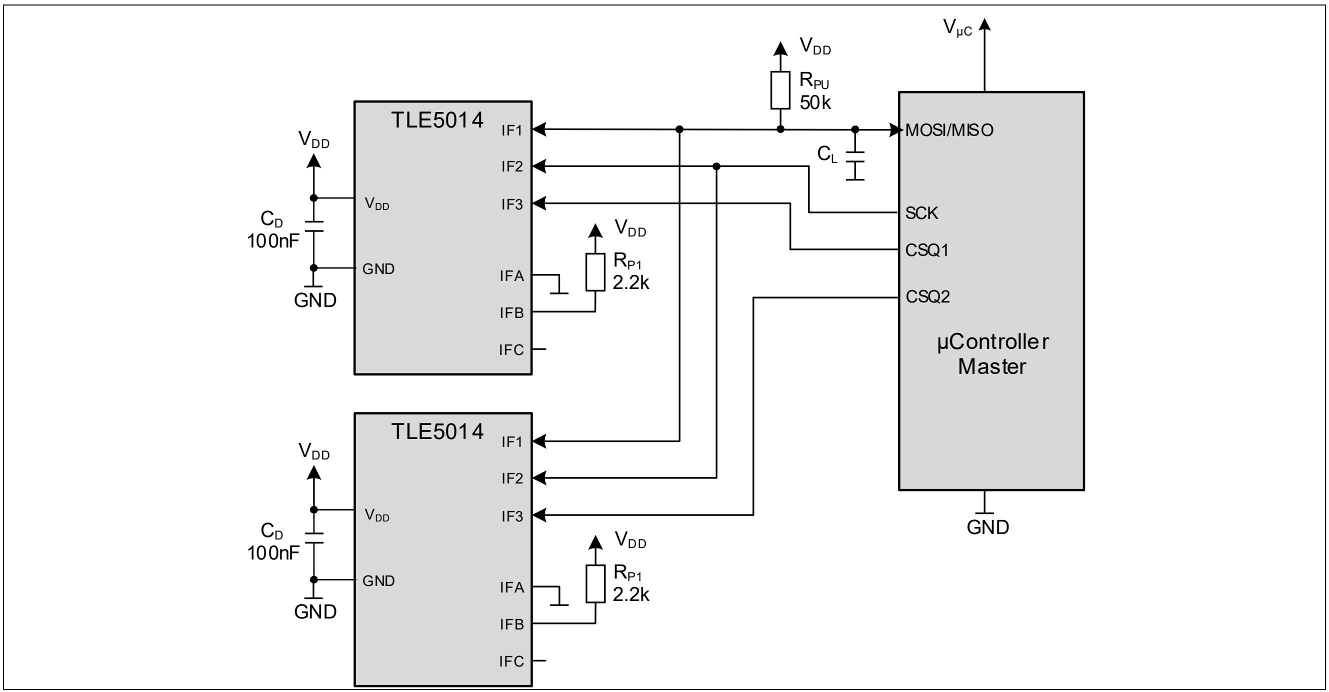

Figure 2 Application circuit for common DATA/SCK lines

1.2.2 Common CSQ/SCK lines

Figure 3 shows an application circuit where two TLE5014SP are connected in bus mode configuration. In this configuration, each TLE5014SP has its own DATA line. The CSQ and SCK are common for both sensors. This configuration can be used to read the data at the same point in time. Similar to the use case for a single sensor IFA and IFB shall be tied to a fixed define voltage. IFC shall be left opened.

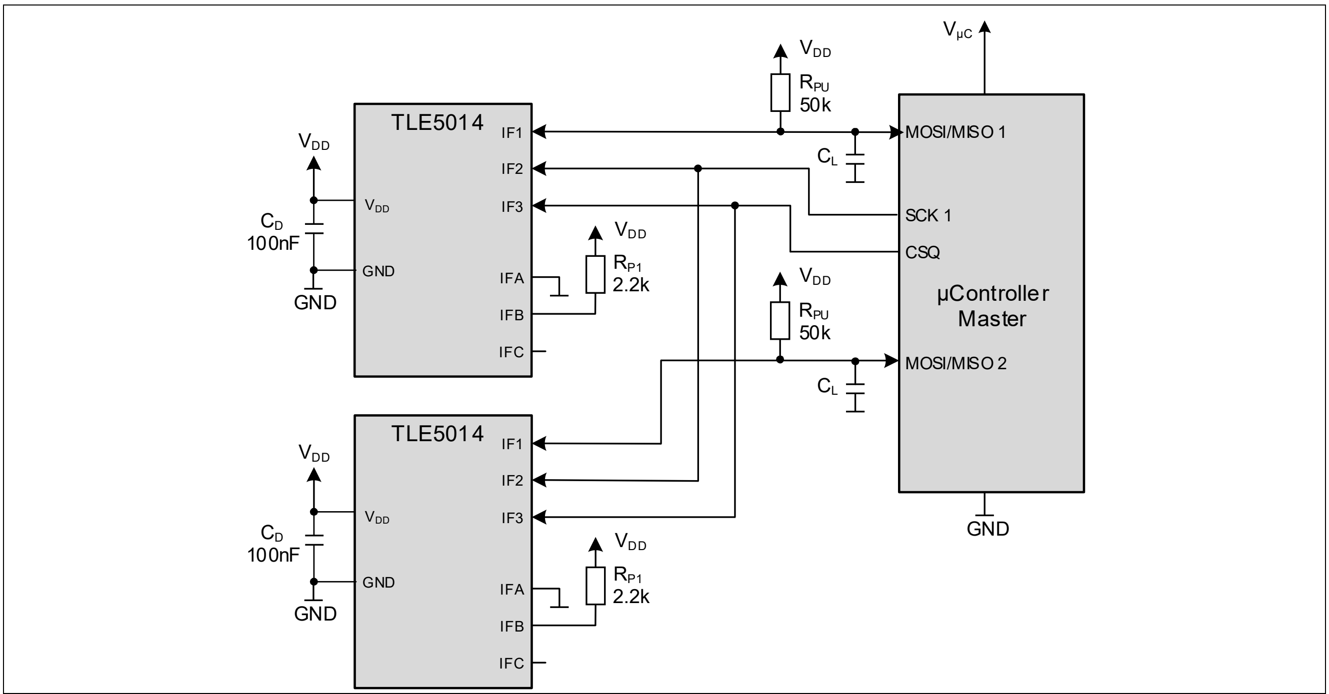

Figure 3 Application circuit for common CSQ/SCK lines

2 Synchronous Serial Communication (SSC) interface

The SSC interface is a half-duplex communication protocol. The communication is always initiated by the microcontroller by sending a command to the TLE5014SP. The command can be either a Read access (**Figure 6**) or a Write access (**Figure 7**). According to the command, the microcontroller can either send a data word to the TLE5014SP (Write access) or receive a data word from the TLE5014SP (Read access). At the end of the communication the TLE5014SP sends a safety word.

The 3-pin SSC Interface is composed of:

- DATA: Bidirectional data line. Data bits are sent synchronously with the clock line.
- SCK: Unidirectional clock line. Generated by the microcontroller, TLE5014SP is always a slave.
- CSQ: Chip select, active low. Asserted by the microcontroller to select a slave.

2.1 Data transmission

The data communication via SSC interface has the following characteristic:

- The SSC Interface is word-aligned. All functions are activated after each transmitted word.
- The microcontroller selects a TLE5014SP by asserting the CSQ to low. A "high" condition on the negated Chip Select pin (CSQ) of the selected TLE5014SP interrupts the transfer immediately. The CRC calculator is automatically reset.
- Data is put on the data line with the rising edge on SCK and read with the falling edge on SCK. Similar to a SPI configuration with CPOL=0 and CPHA=1.
- After changing the data direction, a delay (t_{wr_delay}) has to be considered before continuing the data transfer. This is necessary for internal register access.
- After sending the Safety Word, the transfer ends. To start another data transfer, the CSQ has to be deselected once for t_{CSoff}.
- The SSC is set to Push-Pull by default. The Push-Pull driver is only active if the TLE5014SP has to send data, otherwise the Push-Pull is disabled for receiving data from the microcontroller.

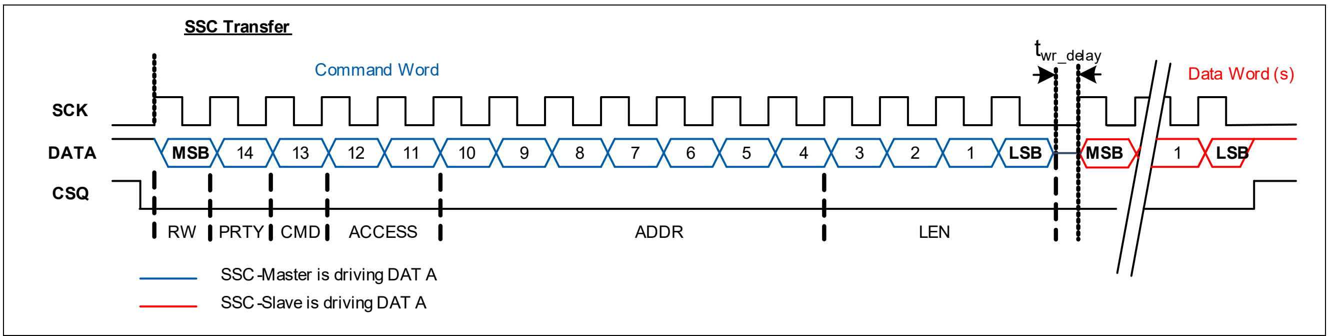

Figure 4 SSC data transmission

2.1.1 Bit Numbering

The SSC communication is using the convention: Most Significant Bit (MSB) first. **Figure 4** shows the Command Word and the beginning of the Data Word to demonstrate the bit numbering.

2.1.2 Update of update-registers

At a rising edge of CSQ without a preceding data transfer (no SCK pulse), the content of all registers which have an update buffer is saved into the buffer. The content of the update buffer can be read by sending a Read

command for the desired register and setting the ACCESS bits of the Command Word to 11_B.
This feature allows the user to take a snapshot of all necessary system parameters at the same time.

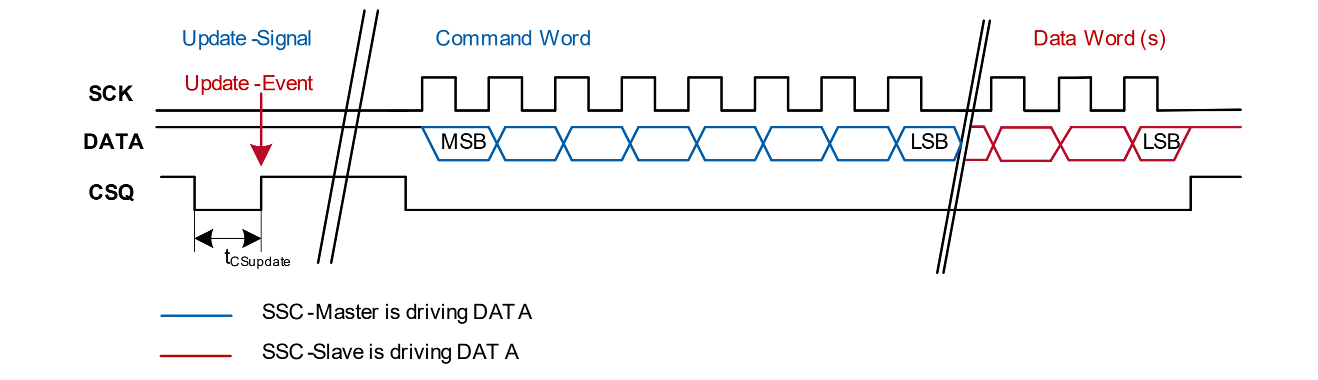

Figure 5 Update of update-registers

The types of function used in the **Chapter 3** are listed here.

Table 2 Bit types

Abbreviation	Function	Description
R	Read	Read-only registers
W	Write	Read and write registers
U	Update	Update buffer for this bit is present. If an update is issued and the Update-Register Access bits (ACCESS in Command Word) are set, the immediate values are stored in this update buffer simultaneously.

2.2 Data transfer

The SSC data transfer is word aligned. The following transfer words are possible:

- Command word (to access and change operating modes of the TLE5014SP)
- Data words (any data transferred in any direction)
- Safety word (confirms the data transfer and provide status information)

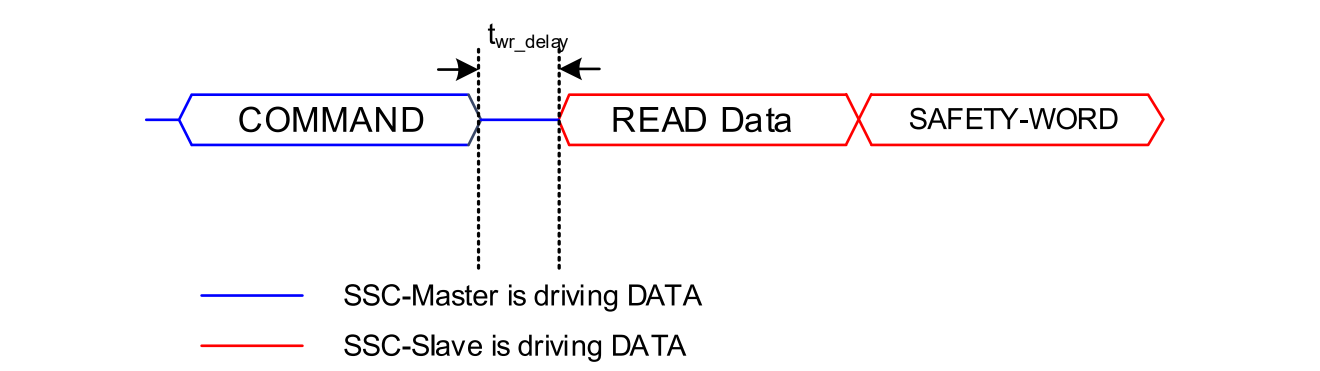

Figure 6 SSC data transfer (data read example)

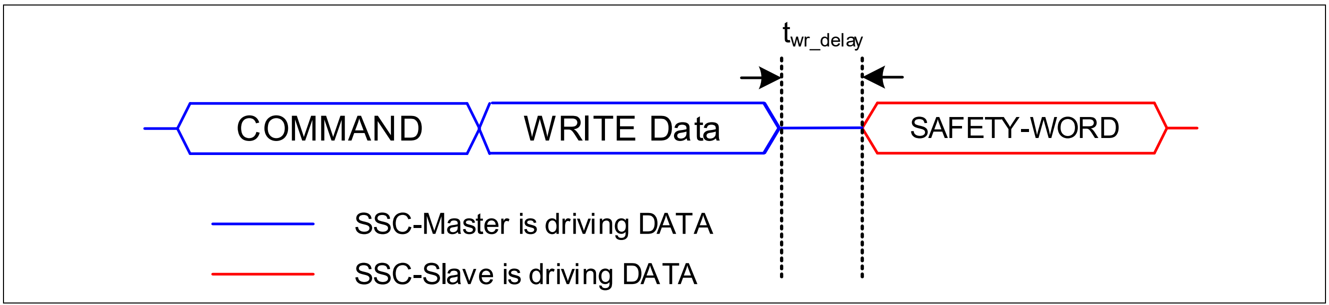

Figure 7 SSC data transfer (data write example)

2.2.1 Command Word

The TLE5014SP is controlled by a command word. It is sent first at every data transmission.The structure of the command word is shown in **Table 3**. **Figure 8** shows an example how to read the angle value in register 0x02, (AVAL register).

Table 3 Structure of the command word

Name	Bits	Description
RW	[15]	Read - Write 0: Write 1: Read
PRTY	[14]	Command parity Odd parity of all Command-Word-bits. Number of “1”s has to be odd
CMD	[13]	Set to 0_B
ACCESS	[12:11]	Access mode to registers 00_B: Direct access 11_B: Update register; read-access

Table 3 Structure of the command word (cont'd)

Name	Bits	Description
ADDR	[10:4]	7-bit Address
LEN	[3:0]	Set to 1_B

Example: In order to read the angle value AVAL, the following command word has to be sent. The angle value is stored in register 0x02. (AVAL register, see **Chapter 3**.)

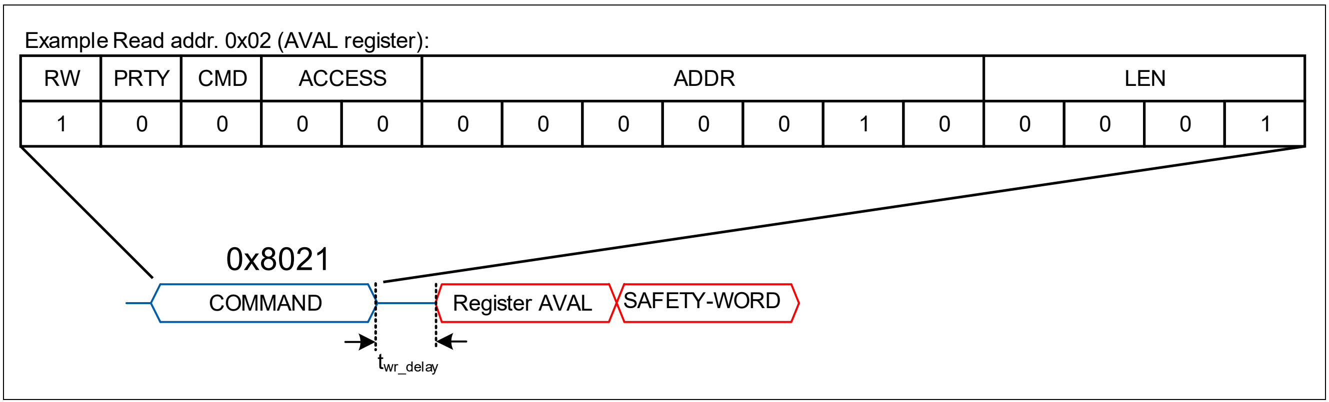

Figure 8 command word to read AVAL register

2.2.2 Safety word

The safety word contains following bits:

Table 4 Structure of the safety word

Name	Bits	Description
STAT	Chip and Interface Status.	
	[15]	Indication of chip reset (undervoltage, watchdog) (resets after readout via SSC) 0: Reset occurred 1: No reset
	[14]	System Error (e.g. Overvoltage; Undervoltage; V_{DD}-, GND- off; ROM) 0: Error occurred 1: No error
	[13]	Interface Access Error (access to wrong address; wrong lock, wrong parity, wrong access) 0: Error occurred 1: No error
	[12]	Angle Value error (ADC , vector length (S_VEC) or redundant angle calculation error(S_PLAUSI)) 0: Angle value invalid 1: Angle value valid
RESP	[11:8]	Sensor Number Response Indicator The sensor no. bit is pulled low and the other bits are high
CRC	[7:0]	Cyclic Redundancy Check (CRC) includes Command Word, Data-words, STAT and RESP

2.2.3 Bus mode

Up to four TLE5014SP can be connected on the same bus (common DATA/SCK lines). For safety purpose, the safety word includes a sensor number indicator. The sensor number indicator (RESP) can be checked by the controller to ensure that the correct sensor has been addressed. The sensor number indicator can be changed by programming the S_NR_EEP field. For further details, see **Chapter 4**.

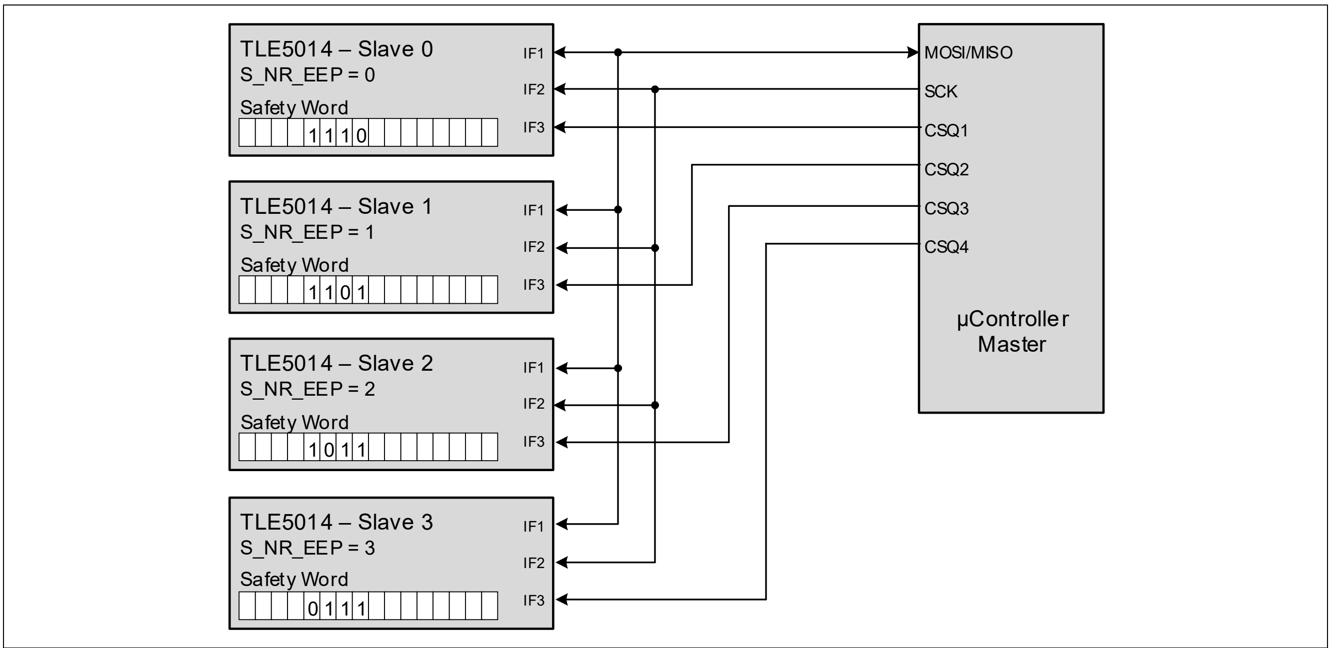

Figure 9 Sensor Number Response Indicator in bus mode

2.2.4 Cyclic Redundancy Check (CRC)

- The CRC is according to the J1850 Bus-Specification.
- Every new transfer resets the CRC generation.
- Every Byte of a transfer will be taken into account to generate the CRC (also the sent command(s)).
- Generator-Polynomial: $X^8+X^4+X^3+X^2+1$, but for the CRC generation the fast-CRC generation circuit is used (see **Figure 10**).
- The remainder of the fast CRC circuit is initial set to 11111111_B.
- Remainder is inverted before transmission.

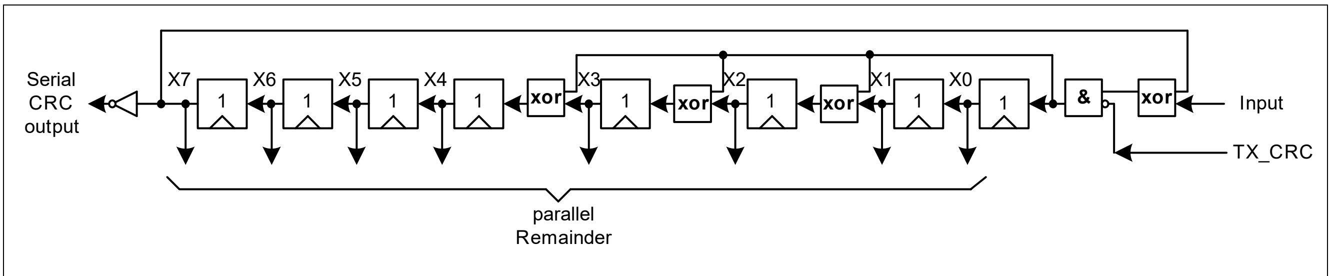

Figure 10 Fast CRC polynomial division circuit

Two code examples to compute the CRC are provided. The first implementation is based on a two loops implentation. This implementation is recommended if memory space is critical in the application. The second implementation replaces the inner loop by a look-up-table. It requires more memory space but the computation time is lower.

2.2.4.1 Code example - two loop implementation

```
//Example
//CMD word = 0x8021, Data received = 0x995C, Safety Word = 0xFE4F
//Final CRC should be equal to 0x4F

//CRC computation is using every byte.
uint8_t inputByte[5] = {0x80,0x21,0x99,0x5C,0xFE};

//Seed value = 0xFF
uint8_t crc = 0xFF;

for(uint8_t i=0;i<5;i++)
{
        crc ^= inputByte[i];
        for(uint8_t u=0;u<8;u++)
        {
                if((crc&0x80)!=0)
                {
                        crc <<= 1;
                        crc ^= 0x1D;
                }
                else
                        crc <<= 1;
        }
}

//Final inversion
crc = ~crc;
```

2.2.4.2 Code example - LUT implementation

```
//Example
//CMD word = 0x8021, Data received = 0x995C, Safety Word = 0xFE4F
//Final CRC should be equal to 0x4F

//CRC computation is using every byte.
uint8_t inputByte[5] = {0x80,0x21,0x99,0x5C,0xFE};

//Seed value = 0xFF
uint8_t crc = 0xFF;

for(uint8_t i=0;i<5;i++)
{
        crc = LUT[crc ^ inputByte[i]];
}

//Final inversion
crc = ~crc;

uint8_t LUT[256] =

0x00, 0x1D, 0x3A, 0x27, 0x74, 0x69, 0x4E, 0x53, 0xE8, 0xF5, 0xD2, 0xCF, 0x9C, 0x81,
0xA6, 0xBB, 0xCD, 0xD0, 0xF7, 0xEA, 0xB9, 0xA4, 0x83, 0x9E, 0x25, 0x38, 0x1F, 0x02,
0x51, 0x4C, 0x6B, 0x76, 0x87, 0x9A, 0xBD, 0xA0, 0xF3, 0xEE, 0xC9, 0xD4, 0x6F, 0x72,
0x55, 0x48, 0x1B, 0x06, 0x21, 0x3C, 0x4A, 0x57, 0x70, 0x6D, 0x3E, 0x23, 0x04, 0x19,
0xA2, 0xBF, 0x98, 0x85, 0xD6, 0xCB, 0xEC, 0xF1, 0x13, 0x0E, 0x29, 0x34, 0x67, 0x7A,
0x5D, 0x40, 0xFB, 0xE6, 0xC1, 0xDC, 0x8F, 0x92, 0xB5, 0xA8, 0xDE, 0xC3, 0xE4, 0xF9,
0xAA, 0xB7, 0x90, 0x8D, 0x36, 0x2B, 0x0C, 0x11, 0x42, 0x5F, 0x78, 0x65, 0x94, 0x89,
0xAE, 0xB3, 0xE0, 0xFD, 0xDA, 0xC7, 0x7C, 0x61, 0x46, 0x5B, 0x08, 0x15, 0x32, 0x2F,
0x59, 0x44, 0x63, 0x7E, 0x2D, 0x30, 0x17, 0x0A, 0xB1, 0xAC, 0x8B, 0x96, 0xC5, 0xD8,
0xFF, 0xE2, 0x26, 0x3B, 0x1C, 0x01, 0x52, 0x4F, 0x68, 0x75, 0xCE, 0xD3, 0xF4, 0xE9,
0xBA, 0xA7, 0x80, 0x9D, 0xEB, 0xF6, 0xD1, 0xCC, 0x9F, 0x82, 0xA5, 0xB8, 0x03, 0x1E,
0x39, 0x24, 0x77, 0x6A, 0x4D, 0x50, 0xA1, 0xBC, 0x9B, 0x86, 0xD5, 0xC8, 0xEF, 0xF2,
0x49, 0x54, 0x73, 0x6E, 0x3D, 0x20, 0x07, 0x1A, 0x6C, 0x71, 0x56, 0x4B, 0x18, 0x05,
0x22, 0x3F, 0x84, 0x99, 0xBE, 0xA3, 0xF0, 0xED, 0xCA, 0xD7, 0x35, 0x28, 0x0F, 0x12,
0x41, 0x5C, 0x7B, 0x66, 0xDD, 0xC0, 0xE7, 0xFA, 0xA9, 0xB4, 0x93, 0x8E, 0xF8, 0xE5,
0xC2, 0xDF, 0x8C, 0x91, 0xB6, 0xAB, 0x10, 0x0D, 0x2A, 0x37, 0x64, 0x79, 0x5E, 0x43,
0xB2, 0xAF, 0x88, 0x95, 0xC6, 0xDB, 0xFC, 0xE1, 0x5A, 0x47, 0x60, 0x7D, 0x2E, 0x33,
0x14, 0x09, 0x7F, 0x62, 0x45, 0x58, 0x0B, 0x16, 0x31, 0x2C, 0x97, 0x8A, 0xAD, 0xB0,
0xE3, 0xFe, 0xD9, 0xC4
};
```

2.2.5 Error handling

An internally detected sensor fault is indicated in the safety word.

It is recommended to read the status register STAT (**Chapter 3**) in this case to get further information of the error. Reading of the STATUS register causes a reset of the error bit in the safety word.

2.3 SSC timing definition

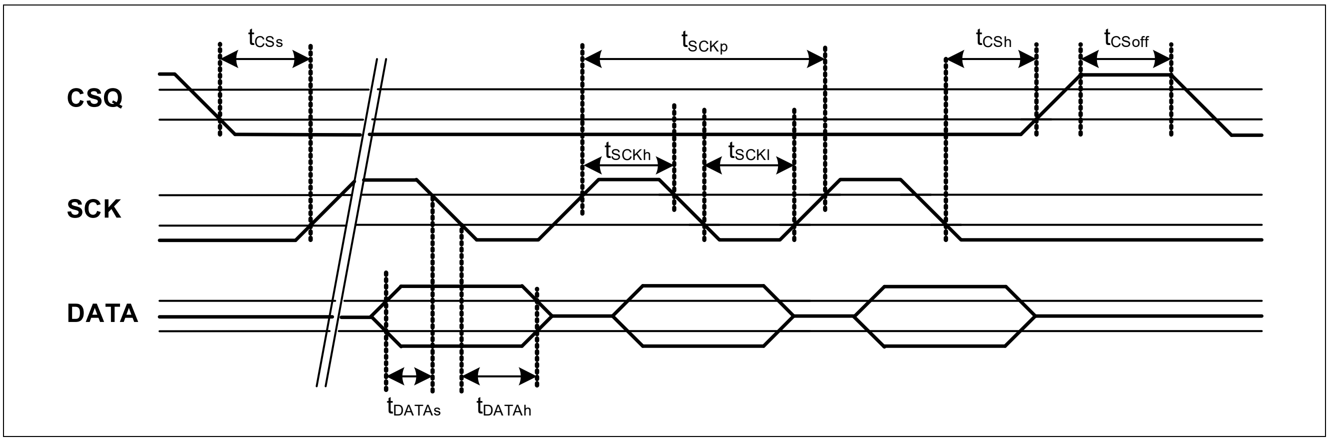

Figure 11 SSC timing

SSC inactive time (CS_{off})

The SSC inactive time defines the delay time after a transfer before the TLE5014SP can be selected again.

Table 5 SSC push-pull timing specification

Parameter	Symbol	Values			Unit	Note or Test Condition
		Min.	Typ.	Max.		
SSC baud rate	f_{SSC}	–	–	8.0	Mbit/s	C_L< 50pF
CSQ setup time	t_{CSs}	105	–	–	ns	
CSQ hold time	t_{CSh}	105	–	–	ns	
CSQ off	t_{CSoff}	600	–	–	ns	SSC inactive time
SCK period	t_{SCKp}	125	–	–	ns	
SCK high	t_{SCKh}	40	–	–	ns	
SCK low	t_{SCKl}	30	–	–	ns	
DATA setup time	t_{DATAs}	40	–	–	ns	
DATA hold time	t_{DATAh}	40	–	–	ns	
Write read delay	t_{wr_delay}	130	–	–	ns	
Update time	$t_{CSupdate}$	1	–	–	µs	See **Figure 5**
CSQ update hold time	t_{CSudh}	600	–	–	ns	See **Figure 5**

3 Register Description

The TLE5014SP includes several registers that can be accessed via SSC. These registers are used to retrieve data such as the angle value or configure the TLE5014SP. This chapter describes the registers which can be read via the SSC interface

Table 6 EEPROM page map

Register Short Name	Register Long Name	Offset Address
STAT	Status register	0x00
AVAL	Angle value register	0x02
ASPD	Angle speed register	0x03
RAW X	ADC X-raw register	0x06
RAW Y	ADC Y-raw register	0x07
ANG_BASE/AND_DIR	Angle base & Angle direction register	0x14
EEP_0 ... EEP_7	EEPROM interface register	0x60 ... 0x67
EEP_PAGE	EEPROM page interface register	0x68
EEP_STATUS	EEPROM status register	0x69

3.1 Register Description

3.1.1 STAT

The STAT register provides detailed information on sensor status and internal sensor errors.

Register Address: 0x00
Read Command: 0xC001

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_ST	**RES**	**S_IF**	**S_VEC**	**RES**	**RES**	**S_EEP**	**S_CRC**	**S_PROG**	**S_BIST**	**S_ADCT**	**S_ADC**	**S_VR**	**S_WD**	**S_PLAUSI**	**S_RST**
u		u	u			u	u	u	u	u	u	u	u	u	u

Field	Bits	Type	Description
RD_ST	15	u	**Read status** Read bit that indicates that the bits in the STAT-line are updated since the last readout
RES	14		**Reserved**
S_IF	13	u	**Interface status** Bit is set when a interface issue happened (e.g. interface command parity, wrong register access, timeout)
S_VEC	12	u	**Vector length status** Bit indicates when the vector length of the actual angle_vector exceeds the defined limits
RES	11		**Reserved**
RES	10		**Reserved**

Field	Bits	Type	Description
S_EEP	9	u	**EEPROM status** Bit indicates whenever the EEPROM initiates an error
S_CRC	8	u	**CRC status** Bit is set whenever one of the bit error detection functions is alarming
S_PROG	7	u	**State machine status** Bit indicates a state machine (iSM) error
S_BIST	6	u	**BIST status** Bit is set whenever one of the startup-BIST is failing. (BIST: Built in self test)
S_ADCT	5	u	**Temperature status** Bit is set when the temperature is out of limits or the temperature delta check is failing
S_ADC	4	u	**ADC status** Bit is set when the ADC_raw_max or ADC_raw_delta check fails or the filter-scheduling check is alarming
S_VR	3	u	**Voltage status** Bit is a combination of several voltage checks
S_WD	2	u	**Watchdog status** Bit is set when the last reset happened due to a watchdog event
S_PLAUSI	1	u	**Plausibility check status** Bit is set when the redundant angle calculation is failing the limits
S_RST	0	u	**Reset status** Reset occurred

3.1.2 AVAL

The AVAL register contains the current angle value. The angle information is represented as a signed 15-bit value. If the microcontroller is reading more often than t_{update}, the flag RD_AV can be used to determine if the current angle value has been updated since the last readout.

Register Address: 0x02
Read command: 0x8021

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_AV	**AVAL**														

Field	Bits	Type	Description
RD_AV	15		**Read angle value** Read bit that indicates that the bits in the AVAL register are updated since last readout by interface

Field	Bits	Type	Description
AVAL	14:0		**Angle value** Signed 15-bit internal angle value: 0° = 0x0000 -180° = 0x4000 179.99° = 0x3FFF

3.1.3 ASPD

The ASPD register contains the current angle speed value. It is coded as a 12-bit value. The flag RD_ASP can be used to determine if the current angle speed value has been update since the last readout.

Register Address: 0x03
Read command: 0xC031

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_ASPD	**RES**			**ASPD**											
u				u											

Field	Bits	Type	Description
RD_ASPD	15	u	**Read angle speed** Read bit that indicates that the bits in the ASPD-register are updated since last readout by interface
RES	14:12		**Reserved**
ASPD	11:0	u	**Angle speed** Calculated angle speed signed 12-bit value $\text{speed}[°/\text{s}] = \frac{\frac{360°}{65536} \times \text{ASPD[digits]}}{t_{\text{update[s]}}}$

3.1.4 RAW X

The RAW X register contains the 16 bit unsigned raw value of the ADC (cosine component). Reading RAW X register will update the RAW Y value.

Register Address: 0x06
Read command: 0xC061

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_X															
u															

Field	Bits	Type	Description
RAW_X	15:0	u	**X-raw value** Unsigned 16 bit value of X-raw.

3.1.5 RAW Y

The RAW Y register contains the 16 bit unsigned raw value of the ADC (sine component).

Register Address: 0x07
Read command: 0x8071

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_Y															
u															

Field	Bits	Type	Description
RAW_Y	15:0	u	**Y-raw value** Unsigned 16 bit value of Y-raw.

3.1.6 ANG_BASE/AND_DIR

The ANG_BASE/ANG_DIR register contains the offset value to set the 0° angle position. The ANG_DIR bit inverts angle and angle speed values.

Register Address: 0x14
Read command: 0xC141

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANG_BASE														**ANG_DIR**	**RES**
w														w	

Field	Bits	Type	Description
ANG_BASE	15:2	w	**Angle base** signed 14 bit zero angle value. Output angle is referring to this angle base 0x0000 = 0°, 0x2000 = -180°, 0x1FFF = +179,978°, 0x001 = 0,02197°
ANG_DIR	1	w	**Direction of the angle rotation** 0 = clockwise 1 = counter clockwise
RES	0		**Reserved**

3.1.7 EEP_0 ... EEP_7

The EEP_0, EEP_1 ... EEP_7 registers are used to interact with the EEPROM. EEP_x registers are mapped between address 0x60 and 0x67. For more details refer to the chapter **EEPROM Programming**.

Register Address: EEP_0:0x60, EEP_1:0x61 ... EEP_7:0x67
Read command: EEP_0:0xC601, EEP_1:0x8611 ... EEP_7:0x8671

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEP_x															
w															

Field	Bits	Type	Description
EEP_x	15:0	w	**Content of the page** EEP_0: address 0x0 ... EEP_1: address 0x1 EEP_7: address 0x7

3.1.8 EEP_PAGE

The EEP_PAGE register is used to interact with the EEPROM. Writing the field EEP_PAGE will map the corresponding page into the registers EEP_0 ... EEP_7. For more details refer to the chapter **EEPROM Programming**.

Register Address: 0x68

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES												**EEP_PAGE**			
												w			

Field	Bits	Type	Description
RES	15:4		**Reserved**
EEP_PAGE	3:0	w	**EEP_PAGE** EEPROM page number

3.1.9 EEP_STATUS

The EEP_STATUS register indicates the result after EEPROM programming. It is recommended to read this register after EEPROM programming to ensure that no error occured. For more details refer to the chapter **EEPROM Programming**.

Register Address: 0x69
Read command: 0xC691

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES												**VER_RESULT**		**RES**	

Field	Bits	Type	Description
RES	15:4		**Reserved**
VER_RESULT	3:2		**Verification Result** Indicates result of verification: $00:_B$ No verification error occurred $01:_B$ One single bit verification error occurred $10:_B$ More than one single bit verification error occurred $11:_B$ At least one multi-bit verification error occurred
RES	1:0		**Reserved**

4 EEPROM Page Map

The TLE5014SP has an EEPROM as non-volatile memory where sensor configuration data can be stored. The memory is organized in pages, each page contains 8 registers, each register has 16 bit. Some pages and registers can be accessed by the user and the content can be modified thus changing chip configuration. Care has to be taken when modifying content of the EEPROM as the device behavior can be changed and no or wrong output data can be generated. There is no automatic restore functionality where the pre-programmed status can be recovered.

The EEPROM content is secured by a CRC. In case any content of the EEPROM is changed, the CRC has to be recalculated and written to the corresponding register, otherwise an error is indicated. The CRC has always be calculated out of several pages and registers, which are indicated in the description of the corresponding CRC (**Table 7**). The algorithm for the CRC calculation is described in **Chapter 6.4**.

The following **Table 7** shows the pages and EEPROM registers of each page and their function. A detailed description of the EEPROM registers and the possible settings can be found in **Chapter 5**.

The startup value of each configuration register of the TLE5014SP is stored in a corresponding EEPROM register, which can be modified by the user. **Table 7** lists the allocation of the user-configurable EEPROM pages.

Care shall be taken when the EEPROM configuration is changed, as not all possible configurations are released by Infineon. This means, that they can be used for testing purpose but not necessarily for production. All released configurations can be found in the TLE5014SP datasheet.

Table 7 EEPROM page map

Page	EEPROM Register	Name	Description
0x009	0x0090	–	Reserved
	0x0092	–	Reserved
	0x0094	–	Reserved
	0x0096	**ACSTAT**	Activation of internal safety mechanism
	0x0098	**MOD1**	3 V/5 V, pad configuration, LUT options
	0x009A	–	Reserved
	0x009C	**IF123_CFG**	Pin configuration of pin IF1, IF2, IF3
	0x009E	–	Reserved

Table 7 EEPROM page map (cont'd)

Page	EEPROM Register	Name	Description
0x00A	0x00A0	**ANG_BASE/ANG_DIR**	Angle base & rotation direction
	0x00A2	–	Reserved
	0x00A4	**CRC_CFG**	Configuration CRC. CRC has to be calculated with following values: registers 0x0096 & 0x0098 & 0x009A & 0x009C & 0x009E & 0x00A0 & 0x00A2 & first byte of 0x00A4 (bit [15:8] & 0xFF00)
	0x00A6	–	Reserved
	0x00A8	–	Reserved
	0x00AA	–	Reserved
	0x00AC	–	Reserved
	0x00AE	–	Reserved
0x00B	0x00B0	LUT_0	Look-up table value for 0°
	0x00B2	LUT_1	Look-up table value for 11.25°
	0x00B4	LUT_2	Look-up table value for 22.50°
	0x00B6	LUT_3	Look-up table value for 33.75°
	0x00B8	LUT_4	Look-up table value for 45.00°
	0x00BA	LUT_5	Look-up table value for 56.25°
	0x00BC	LUT_6	Look-up table value for 67.50°
	0x00BE	LUT_7	Look-up table value for 78.75°
0x00C	0x00C0	LUT_8	Look-up table value for 90.00°
	0x00C2	LUT_9	Look-up table value for 101.25°
	0x00C4	LUT_10	Look-up table value for 112.50°
	0x00C6	LUT_11	Look-up table value for 123.75°
	0x00C8	LUT_12	Look-up table value for 135.00°
	0x00CA	LUT_13	Look-up table value for 146.25°
	0x00CC	LUT_14	Look-up table value for 157.50°
	0x00CE	LUT_15	Look-up table value for 168.75°
0x00D	0x00D0	LUT_16	Look-up table value for 180.00°
	0x00D2	LUT_17	Look-up table value for 191.25°
	0x00D4	LUT_18	Look-up table value for 202.50°
	0x00D6	LUT_19	Look-up table value for 213.75°
	0x00D8	LUT_20	Look-up table value for 225.00°
	0x00DA	LUT_21	Look-up table value for 236.25°
	0x00DC	LUT_22	Look-up table value for 247.50°
	0x00DE	LUT_23	Look-up table value for 258.75°

Table 7 EEPROM page map (cont'd)

Page	EEPROM Register	Name	Description
0x00E	0x00E0	LUT_24	Look-up table value for 270.00°
	0x00E2	LUT_25	Look-up table value for 281.25°
	0x00E4	LUT_26	Look-up table value for 292.50°
	0x00E6	LUT_27	Look-up table value for 303.75°
	0x00E8	LUT_28	Look-up table value for 315.00°
	0x00EA	LUT_29	Look-up table value for 326.25°
	0x00EC	LUT_30	Look-up table value for 337.50°
	0x00EE	LUT_31	Look-up table value for 348.75°
0x00F	0x00F0	**CRC_LUT**	8-bit CRC for Look-up table; CRC has to be calculated with following values: page 0x00B & 0x00C & 0x00D & 0x00E & first byte of register 0x00F0 (bits [15:8] & 0xFF00)
	0x00F2	CUST_ID_0	16-bit of customer ID
	0x00F4	CUST_ID_1	16-bit of customer ID
	0x00F6	CUST_ID_2	16-bit of customer ID
	0x00F8	CUST_ID_3	16-bit of customer ID
	0x00FA	CUST_ID_4	16-bit of customer ID
	0x00FC	CUST_ID_5	16-bit of customer ID
	0x00FE	CUST_ID_6	16-bit of customer ID

5 EEPROM Register Description

This chapter describes the EEPROM registers which can be accessed via SSC. Further registers needed for EEPROM programming are described in **Chapter 6**.

Care shall be taken that there is no unintended modification of bits. When modifying registers, reserved bits have to be read and written back with the same value.

5.1 Page 0x009

5.1.1 ACSTAT

EEPROM Address: 0x0096

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES															**AS_RST**

Field	Bits	Type	Description
RES	15:1		**Reserved** Reserved bit, shall not be changed
AS_RST	0		**System reset** Write to execute a system reset

5.1.2 MOD1

EEPROM Address: 0x0098

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		**RST_3_5**	**RES**	**FIR_MD**	**RES**	**LUT_MD**		**RES**							
					Reserv										

Field	Bits	Type	Description
RES	15:14		**Reserved**
RST_3_5	13		**Comparator reset 3V or 5V** 0_B: Reset comparator set to < 3.0 V 1_B: Reset comparator set to < 4.2 V
RES	12		**Reserved**
FIR_MD	11		**FIR mode** Internal update rate of angle calculation: 0_B: 25.6 µs 1_B: 51.2 µs

Field	Bits	Type	Description
RES	10	Reserved	
LUT_MD	9:8		**LUT mode** Look-up table (LUT) configuration: $00:_B$ LUT disabled $01:_B$ LUT range = 360° (angle steps = 11.25°) $10:_B$ LUT range = 180° (angle steps = 5.625°) $11:_B$ LUT range = 90° (angle steps = 2.813°)
RES	7:0		**Reserved**

5.1.3 IF123_CFG

EEPROM Address: 0x009C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD_OV_OFF	**PAD_SPIKE**	**RES**		**IF3_OD**	**IF3_CFG**	**IF2_OD**	**IF2_CFG**	**IF1_OD**	**IF1_CFG**	**RES**					

Field	Bits	Type	Description
PAD_OV_OFF	15		Enable/disable overvoltage functionality. Output deactivated in case of overvoltage on V_{DD}: $0:_B$ Overvoltage functionality enabled $1:_B$ Overvoltage functionality disabled
PAD_SPIKE	14		Activate spike filter for input stage: $0:_B$ Spike filter disabled $1:_B$ Spike filter enabled
RES	13:12		**Reserved**
IF3_OD	11		IF3 open drain enable: $0:_B$ Disable (Push-Pull) $1:_B$ Enable (Open Drain)
IF3_CFG	10		For test purpose only
IF2_OD	9		IF2 open drain enable: $0:_B$ Disable (Push-Pull) $1:_B$ Enable (Open Drain)
IF2_CFG	8		For test purpose only

Field	Bits	Type	Description
IF1_OD	7		IF1 open drain enable: $0:_B$ Disable (Push-Pull) $1:_B$ Enable (Open Drain)
IF1_CFG	6		For test purpose only
RES	5:0		**Reserved**

5.2 Page 0x00A

5.2.1 ANG_BASE/ANG_DIR

EEPROM Address: 0x00A0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANG_BASE														**ANG_DIR**	**RES**

Field	Bits	Type	Description
ANG_BASE	15:2		Signed 14 bit value: 0° = 0x0000 -180.000° = 0x2000 +179.978° = 0x1FFF +0.02197° = 0x001
ANG_DIR	1		Direction of magnet rotation: $0:_B$ ccw $1:_B$ cw
RES	0		**Reserved** $0:_B$ Reserved bit, shall not be changed

5.2.2 CRC_CFG

EEPROM Address: 0x00A4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						**S_NR_EEP**		**CRC_CFG**							

Field	Bits	Type	Description
RES	15:10		**Reserved**
S_NR_EEP	9:8		Address coding of SSC slave: $00:_B$ ID = 0 $01:_B$ ID = 1 $10:_B$ ID = 2 $11:_B$ ID = 3
CRC_CFG	7:0		Configuration CRC. CRC has to be calculated with following values: registers 0x0096 & 0x0098 & 0x009A & 0x009C & 0x009E & 0x00A0 & 0x00A2 & first byte of 0x00A4 (bit [15:8])

5.3 Page 0x00B - 0x00E

Look-up table, see **Table 7**.

5.4 Page 0x00F

In EEPROM register 0x00F0 the 8-bit CRC for the look-up table is stored.

5.4.1 CRC_LUT

EEPROM Address: 0x00F0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								**CRC_LUT**							

Field	Bits	Type	Description
RES	15:8		**Reserved**
CRC_LUT	7:0		8-bit CRC has to be calculated with following values: Page 0x00B & 0x00C & 0x00D & 0x00E & first byte of 0x00F0 (bits [15:8])

In the EEPROM registers 0x00F2 to 0x00FE the customer ID is stored. In total it is 16 x 7 = 112 bit.
The first 32 bits (EEPROM address 0x00F2 and 0x00F4) are transmitted in the short serial message.

6 EEPROM Programming

The EEPROM of the TLE5014SP consists of user accessible and locked pages with eight 16-bit data words each. To access a page of the EEPROM, the page is mapped into a range of registers, which can then be read or written via the SSC interface.

6.1 EEPROM Unlock

To prevent unintented access during run time the EEPROM is locked. In order to access it, the following sequence needs to be perfomed.

Table 8 Sequence to unlock the EEPROM

Command	Data	Comment
0x0771	0x4711	Write unlock word
0x06C1	0xC000	Disable the internal controller
-	-	EEPROM can now be accessed
0x4011	0x0001	After configuration, trigger a chip reset

```
//Function prototype
void writeSSC(uint16_t command, uint16_t dataToWrite, uint16_t safetyWord);

uint16_t _safetyWord = 0;

//Write unlock word
writeSSC(0x0771,0x4711,_safetyWord);

//Disable internal controller
writeSSC(0x06C1,0xC000,_safetyWord);

//EEPROM can now be accessed

//After configuration, trigger a chip reset
writeSSC(0x4011,0x0001,_safetyWord);
```

6.2 EEPROM Access

To read or write data from the EEPROM, the desired page has first to be mapped into the working registers EEP_0 to EEP_7 (address 0x60 to 0x67) by writing the desired page number (0x9 ... 0xF) to the EEP_PAGE register (address 0x68). The content of the EEPROM page can then be access from registers EEP_0 to EEP_7. **Figure 12** shows the register interface to access the EEPROM content.

Before EEPROM content can be read or written, the sequence describes in **Chapter 6.1** has to be performed.

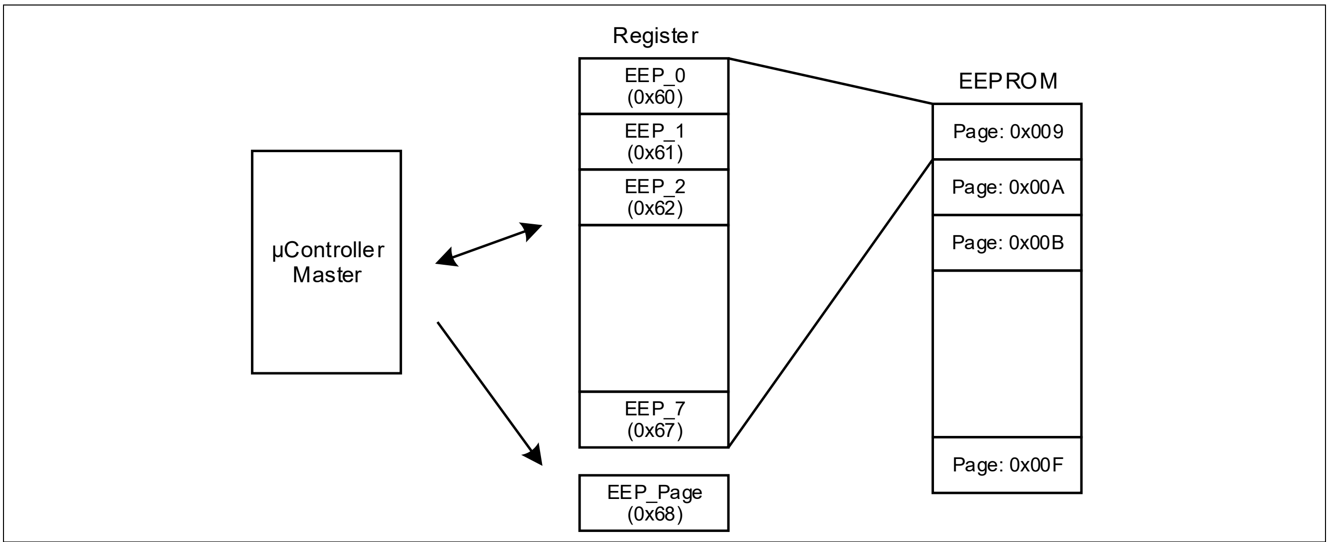

Figure 12 Register interface for EEPROM access

6.2.1 EEPROM Read

The following sequence shows how to read an EEPROM register. The sequence is using ANG_BASE/ANG_DIR located at address 0x00A0 as an example.

Table 9 Sequence to read EEPROM page

Command	Data	Comment
0x4681	0x000A	Write page number into EEP_PAGE register. (e.g. page 0x00A)
0xC601	-	Read register EEP_0 ... EEP_7. (e.g. register EEP_0. Correspond to EEPROM register 0x00A0)

```
//Function prototype
void writeSSC(uint16_t command, uint16_t dataToWrite, uint16_t safetyWord);
void readSSC(uint16_t command, uint16_t dataToRead, uint16_t safetyWord);

uint16_t _safetyWord = 0;

//Unlock EEPROM

//Map page into EEP_0 ... EEP_7 registers
writeSSC(0x4681,0x000A,_safetyWord);

uint16_t EEP_0, EEP_1 ... EEP_7 = 0;

//Read content
readSSC(0xC601, EEP_0, _safetyWord);
readSSC(0x8611, EEP_1, _safetyWord);
readSSC(0x8621, EEP_2, _safetyWord);
readSSC(0xC631, EEP_3, _safetyWord);
readSSC(0x8641, EEP_4, _safetyWord);
readSSC(0xC651, EEP_5, _safetyWord);
readSSC(0xC661, EEP_6, _safetyWord);
readSSC(0x8671, EEP_7, _safetyWord);
```

6.2.2 EEPROM Write

The following sequence shows how to modify a field located in an EEPROM register. The sequence is using the field ANG_DIR from the register ANG_BASE/ANG_DIR located at address 0x00A0 as an example.

Table 10 Sequence to write EEPROM page

Command	Data	Comment
0x4681	0x000A	Write page number into EEP_PAGE register. (e.g. page 0x00A)
0xC601	data	Read register EEP_0 ... EEP_7. (e.g. register EEP_0. Correspond to EEPROM register 0x00A0)
		Change the desired field using mask. Reserved bit shall not be changed. (e.g. data = data | 0x0002 correspond to changing ANG_DIR from ccw to cw)
0x0601	data	Write back EEP_0 ... EEP_7. (e.g. register EEP_0 with modified field)
0x4681	0x03FF	Change to page 0x3FF
0x4671	0x0009	Program with automatic verification
		Compute the new CRC. For more details, see **Chapter 6.4**
0x4011	0x0001	After configuration, trigger a chip reset

```
//Function prototype
void writeSSC(uint16_t command, uint16_t dataToWrite, uint16_t safetyWord);
void readSSC(uint16_t command, uint16_t dataToRead, uint16_t safetyWord);

uint16_t _safetyWord = 0;

//Unlock EEPROM

//Map page into EEP_0 ... EEP_7 registers
writeSSC(0x4681,0x000A,_safetyWord);

uint16_t EEP_0;
//Read address 0x00A0
readSSC(0xC601, EEP_0, _safetyWord);

//Force bit[1] to '1'
EEP_0 = EEP_0 | 0x0002;

//Write the value to EEP_0
writeSSC(0x0601,EEP_0,_safetyWord);

//Program with automatic verification
writeSSC(0x4681,0x03FF,_safetyWord);
writeSSC(0x4671,0x0009,_safetyWord);

//After configuration, trigger a chip reset
writeSSC(0x4011,0x0001,_safetyWord);
```

6.3 EEPROM Timing

Table 11 gives the timing specification for reading and programming the EEPROM.

Table 11 EEPROM timing specification

Parameter	Symbol	Values			Unit	Note or Test Condition
		Min.	Typ.	Max.		
Read time per word	t_{read_ac}	–	–	150	ns	1)
Write time2)	t_{write}	4.0	5.0	6.0	ms	2)
Erase time2)	t_{erase}	4.0	5.0	6.0	ms	2)
Program time2)	$t_{program}$	8.0	10.0	12.0	ms	2)
Startup time after reset	$t_{startup}$	46	57	68	µs	2)

1) Verified by design/characterization.
2) Times are identical, independent of the number of words that are written, erased, or programmed at a time.

6.4 EEPROM CRC

The EEPROM data is protected by checksums. When changes are made to one or more EEPROM registers, the respective checksum has to be recalculated and written into the corresponding register. Therefore, the complete register range which is covered by the checksum has to be read, and the checksum calculation has to be performed as described in **Chapter 6.4**. The addresses of the checksum registers and the register range covered by each checksum are listed in **Table 7**.

Changing the configuration of the TLE5014SP without updating the corresponding checksum results in a CRC error, which is indicated.

6.4.1 CRC generation

The TLE5014SP uses a cyclic redundancy check (CRC) to monitor the integrity of configuration registers and EEPROM pages. Similar to the one used in the Safety word of the SSC protocol, the CRC is generated according to the J1850 Bus-Specification:

- Generator-Polynomial: X8+X4+X3+X2+1; for the CRC generation the fast-CRC generation circuit is used (see **Figure 13**)
- The remainder of the fast CRC circuit is initially set to '11111111$_B$'
- Remainder is inverted before transmission

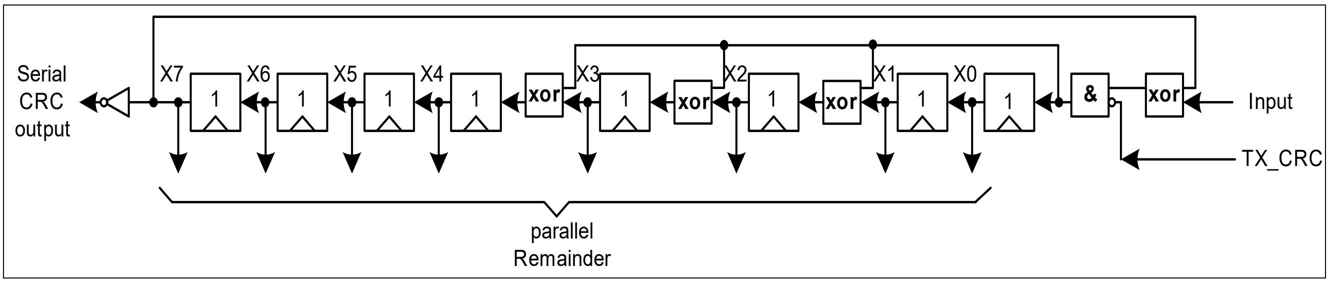

Figure 13 Fast CRC polynomial division circuit

6.5 EEPROM Verification & Lock

To verify if the programing of the EEPROM page was successful, the respective EEPROM page should be read again and verified if the content matches with the intended content.
In addition, there is a register at address 0x69, which indicates the status after the EEPROM burning.

6.5.1 EEPROM Lock

After programming the EEPROM, the content can be locked. In this case, no second programming is possible. In order to lock the EEPROM, the following sequence needs to be performed.

Table 12 Sequence to lock the EEPROM

Command	Data	Comment
0x4681	0x03F0	Change to page 0x03F0
0x4621	0x7FFF	Write 0x7FFF to EEP_2
0x0631	0x7FFF	Write 0x7FFF to EEP_3
0x4681	0x03FF	Change to page 0x3FF
0x4671	0x000C	Write 0x000C to EEP_7
0x4011	0x0001	After configuration, trigger a chip reset

```
//Function prototype
void writeSSC(uint16_t command, uint16_t dataToWrite, uint16_t safetyWord);

uint16_t _safetyWord = 0;

//Unlock EEPROM

//Sequence to lock the EEPROM
//Write 0x7FFF at 0x3F04 and 0x3F06
writeSSC(0x4681,0x03F0,_safetyWord);
writeSSC(0x4621,0x7FFF,_safetyWord);
writeSSC(0x0631,0x7FFF,_safetyWord);

//Write 0x000C at 0x3FFE
writeSSC(0x4681,0x03FF,_safetyWord);
writeSSC(0x4671,0x000C,_safetyWord);

//Trigger a chip reset
writeSSC(0x4011,0x0001,_safetyWord);
```

7 Initial Calibration

This chapter describes which initial calibration has to be performed with the TLE5014SP.

7.1 Angle Base Calibration

The angle base or zero-angle is a 14-bit value which is stored in EEPROM address 0x00A0. This value defines the zero angle in an application and has to be programmed by the customer. The sensor output angle is in reference to this angle base.

Two methods are possible to define the angle base value. The first one is using several reference points to consider the angle offset error over the full range. It is the recommended method. The second one is using only one point. It can be used when an external reference is not available.

7.1.1 Several reference points (recommended)

This method requires an external reference.

1. Measure the angle value of several reference points (e.g. ≥ 16) over a full rotation. A fixed distance between the reference points is not necessary.
2. Calculate the angle difference between each reference point (ideal position) and the corresponding measured angle value.
3. Calculate the arithmetical mean value of these angle differences to get the overall angle offset error.

$$\langle \Delta \mathrm{offset} \rangle = \frac{1}{n} \sum_{i=1}^{n} \Delta i$$

4. Scale the value on a 14 bit representation (shift right by 1). This value represents ANG_BASE.
5. Shift left ANG_BASE by 2 bits and OR it with desired ANG_DIR setting.
6. Write resulting word into ANG_BASE line in EEPROM.

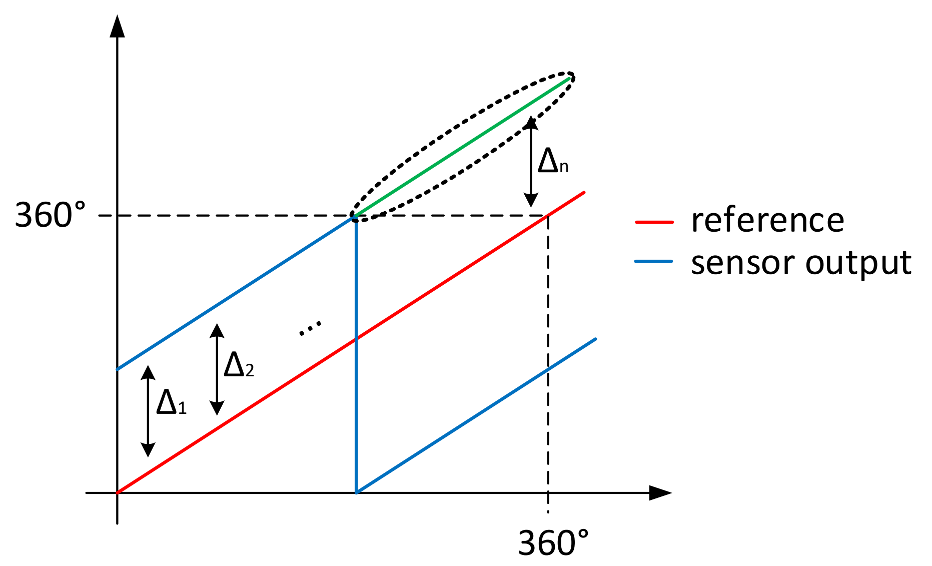

Figure 14 Application circuit for a single TLE5014SP

7.1.1.1 Example

This example shows the angle base calibration with several reference points. For clarity purpose only 4 points are used. In a real application, it is recommended to compute the average with more than 16 reference points.

Table 13 Example for Angle Base Calculation

Value decimal	Value HEX	Description
4096	0x1000	15-bit angle value from sensor, should be new angle base
2048	0x0800	Angle value is scaled to 14 bit (shift right by 2)
8192	0x2002	Angle value is shifted by 2 to the left and OR with ANG_DIR setting
8192	0x2002	Value to be burned to EEPROM address 0x00A0.

7.1.2 Single reference point

1. Position the application to the desired angle.
2. Read the angle value via the SSC interface.
3. Scale the value on a 14 bit representation (shift right by 1). This value represents ANG_BASE.
4. Shift left ANG_BASE by 2 bits and OR it with desired ANG_DIR setting.
5. Write resulting word into ANG_BASE line in EEPROM.

7.1.2.1 Example

This example shows the different steps to configure the angle base. The zero-angle of the application is at 45°. The application is running clockwise (ANG_DIR = 1).

Table 14 Example for Angle Base Calculation

Value decimal	Value HEX	Description
4096	0x1000	15-bit angle value from sensor, should be new angle base
2048	0x0800	Angle value is scaled to 14 bit (shift right by 2)
8192	0x2002	Angle value is shifted by 2 to the left and OR with ANG_DIR setting
8192	0x2002	Value to be burned to EEPROM address 0x00A0.

7.2 Look-up Table Calibration

The sensor has a 32 point look-up table (LUT) implemented. With the help of the look-up table it is possible to do an additional end-of-line calibration and reduce non-linearities resulting from a non perfect magnetic circuit (e.g. assembly tolerances as tilt and/or eccentricity). For this, an external reference is required and the sensor output at defined angle positions has to be measured.

The following procedure is recommended for look-up- table calibration:

1. Measure the angle values on the 32 equidistant distributed reference points over 360° (0°, 11.25°, 22.5°,…)
2. Calculate 16bit look-up table value ($LUT[LSB_{16}]$)according to the equation below:.

$$\mathrm{LUT[LSB16]} = 2 \times \mathrm{Position[LSB16]} - 2 \times \mathrm{Readout[LSB15]}$$

$Position[LSB_{16}]$: 16bit reference angle (0°, 11.25°, 22.5° ... 348.75°)
$Readout[LSB_{15}]$: 15bit angle value at reference position

3. If difference is negative or ≥ 2^{16}: add/subtract 2^{16} so that the result is within 0 and 2^{16}-1. This is equivalent to cutting of the overflow bits resp. is implicitly done, if the calculation is done in pure 16 bit
4. Write each of the 16-bit LUT values into the corresponding reference point's index in the LUT
5. The 8-bit CRC calculated over the written LUT values has to be written into the address 0x00F0, bits [7:0]

In case the look-up table is used for sensor output linearization, the angle base (angle offset) has to be corrected first by programming the angle base register 0x00A0. After defining the correction values for the look-up table, the angle base shall not be changed as this will introduce additional angle error.

In case the look-up table is not used, make sure that the default values are written in the corresponding addresses. The default value is the ideal angle value at the defined position, e.g. address 0x00B0: 0°, 0x00B2: 11.25°, …

The chip is delivered from Infineon with the default values already written to the look-up table.

7.2.1 Example

The table below shows some examples for calculation of the LUT correction values.

Table 15 Examples for Look-up Calculation

Position [°]	Position [LSB16]	Sensor out [LSB15]	Sensor out [°]	Sensor out [LSB16]	LUT value [LSB16]
0	0	32	0.352	64	65472
0	0	32744	359.736	65488	48
11.25	2048	1040	11.426	2080	2016
22.50	4096	2000	21.973	4000	4192
348.75	63488	31680	348.047	63360	63616

7.3 Customer ID

The chip provides a storage of 112 bit for customer data. This data can be written to the EEPROM page 0x00F. Please note that the customer data are not CRC protected.

8 Examples

The following chapter provides some practical examples.

8.1 Angle Read Out Value

The angle value is stored as a 15-bit value. It is described as a signed register, the angle value is stored as Two's complement. A Two's complement number is generated by the following formula.

$$Value = -b_{MSB} * 2^{N-1} + \sum_{i=0}^{N-2} b_i * 2^i \tag{8.1}$$

Example:
For a register of $(100\ 1101\ 1001\ 0011)_B$, the Two's complement is calculated as:
$-(100\ 0000\ 0000\ 0000)_B + (1101\ 1001\ 0011)_B = -16384 + 3475 = -12909$
The corresponding angle calculates to:

$$\text{Angle}[°] = \frac{360°}{2^{15}} \times \text{AngVal[digits]} = \frac{360°}{2^{15}} \times -(12909) = -(141, 82°) \tag{8.2}$$

It is also possible to calculate the angle in an alternative way not using Two's complement.
With the 15-bit resolution, 1 LSB corresponds to $360°/2^{15} = 360°/32768$. A register value of $(100\ 1101\ 1001\ 0011)_B$ equals to 19859, thus giving an angle of:

$$\text{Angle}[°] = \frac{360°}{2^{15}} \times \text{Readout[LSB15]} = \frac{19859° \times 360°}{2^{15}} = 218, 18° \tag{8.3}$$

This results is the same angle as 218.18° = -141.82°

8.2 Common SSC commands

The following table summarizes commands commonly used to interact with the TLE5014SP.

Table 16 Commands commonly used

Type	Command	Data	Comment
Read	0x8021	AVAL	Read angle value
Read	0xC031	ASPD	Read speed value
Read	0xC001	STAT	Read status register
Write	0x0771	0x4711	Unlock EEPROM
Write	0x06C1	0xC000	Disable internal controller
Write	0x4011	0x0001	Trigger a chip reset
Write	0x4681	[0x0009 ... 0x000F]	Map page x into EEP_0 ... EEP_7 registers

Table 16 Commands commonly used

Type	Command	Data	Comment
Read	0xC601	EEP_0	Read EEP_0
Read	0x8611	EEP_1	Read EEP_1
Read	0x8621	EEP_2	Read EEP_2
Read	0xC631	EEP_3	Read EEP_3
Read	0x8641	EEP_4	Read EEP_4
Read	0xC651	EEP_5	Read EEP_5
Read	0xC661	EEP_6	Read EEP_6
Read	0x8671	EEP_7	Read EEP_7
Write	0x0601	EEP_0	Write EEP_0
Write	0x4611	EEP_1	Write EEP_1
Write	0x4621	EEP_2	Write EEP_2
Write	0x0631	EEP_3	Write EEP_3
Write	0x4641	EEP_4	Write EEP_4
Write	0x0651	EEP_5	Write EEP_5
Write	0x0661	EEP_6	Write EEP_6
Write	0x4671	EEP_7	Write EEP_7

9 Revision history

Revision	Date	Changes
1.0	2019-08-13	Initial creation.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-08-13

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.