

TLE5014

GMR-Based Angle Sensor

User's Manual

About this document

Scope and purpose

This document covers the TLE5014 with the different interfaces SENT, SPC, PWM and SICI. It provides further information what should be considered 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 TLE5014 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
TLE5014P16	014P	SP001231814	PG-TDSO-16	PWM Interface, single die
TLE5014S16	014S	SP001231818	PG-TDSO-16	SENT Interface, single die
TLE5014C16	014C	SP001231806	PG-TDSO-16	SPC Interface, single die
TLE5014S16D	014SD	SP001410046	PG-TDSO-16	SENT Interface, dual die
TLE5014C16D	014CD	SP001410042	PG-TDSO-16	SPC Interface, dual die
TLE5014P16D	014PD	SP001673472	PG-TDSO-16	PWM Interface, dual die

Table of Contents

1	Application Circuits	4
2	Interfaces overview	6
2.1	PWM Interface Configuration	6
2.2	SPC Interface Configuration	7
2.3	SENT Interface Configuration	11
3	EEPROM Page Map	13
4	EEPROM Register Description	17
4.1	Page 0x006	17
4.2	Page 0x009	19
4.3	Page 0x00A	22
4.4	Page 0x00B - 0x00E	24
4.5	Page 0x00F	24
5	Working Register Description	26
5.1	STAT Register 0x00: Status	26
5.2	AVAL Register 0x02: Angle Value	27
5.3	ASPD Register 0x03: Angle Speed	27
5.4	RAW_X Register 0x06: X raw value	28
5.5	RAW_Y Register 0x07: Y raw value	28
5.6	TEMP Register 0x0A: Temperature raw value	29
5.7	ANG_BASE register 0x14: Angle base and rotation direction	29
6	SICI	31
6.1	Basic Functionality	31
6.2	SICI Data Transfer	31
6.2.1	Bit Encoding	31
6.2.2	Communication Protocol	34
6.2.3	CRC generation	36
6.2.4	SICI Application Example: Reading working register	36
6.2.5	SICI Application Example: EEPROM read/write	37
6.3	SICI Programming in Bus Mode	38
7	EEPROM Programming	39
7.1	Writing and Reading of EEPROM Pages	39
7.2	Verification of EEPROM	40
7.3	Locking of EEPROM	40
8	Initial Calibration	42
8.1	Zero Angle Calibration	42
8.2	Look-up Table Calibration	42
8.3	Customer ID	43
9	Examples	44
9.1	Angle Read Out Value	44
9.2	Angle Base Calculation	45
9.3	Look-up Table Calculation	46
9.4	Temperature calculation from Short Serial Message	48
9.5	Reading registers with SICI	48



9.6	Reading EEPROM with SICI	49
9.7	Writing EEPROM with SICI	50
10	Revision History	52

Application Circuits

1 Application Circuits

The application circuits in this chapter show the various communication possibilities of the TLE5014.

Figure 1 shows a basic application circuit of a TLE5014 with PWM interface. The pull-down resistor is required to ensure that the sensor starts with a low level after power-on or reset (PWM starts with a rising edge). If the sensor is configured in such a way, that the PWM starts with a falling edge, the pull-down has to be replaced by a pull-up.

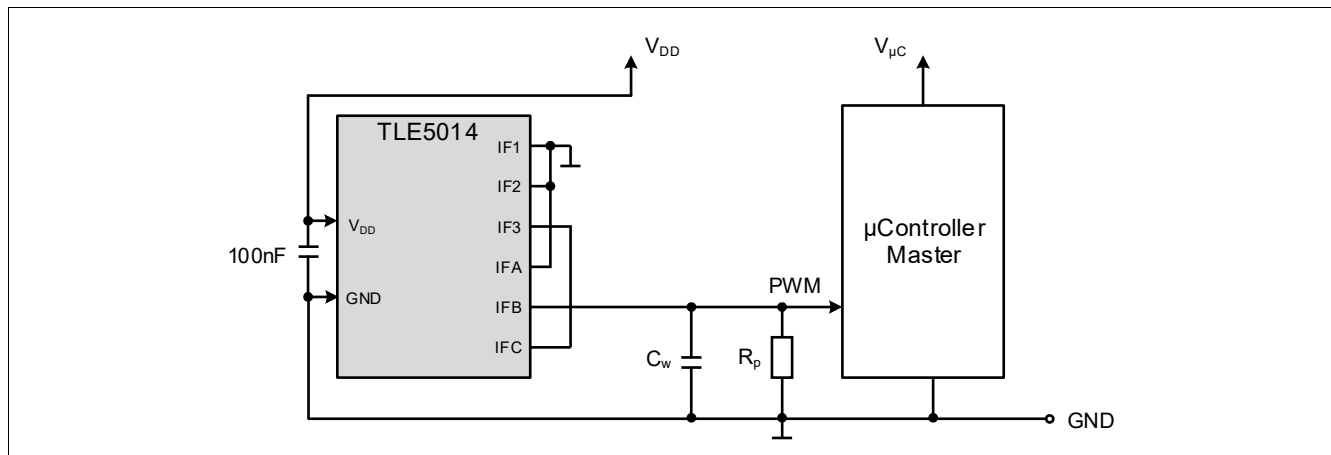


Figure 1 Application circuit for TLE5014 with PWM, sensor configured to start with a rising edge of PWM

Figure 2 shows a basic application circuit of the TLE5014 with SPC mode.

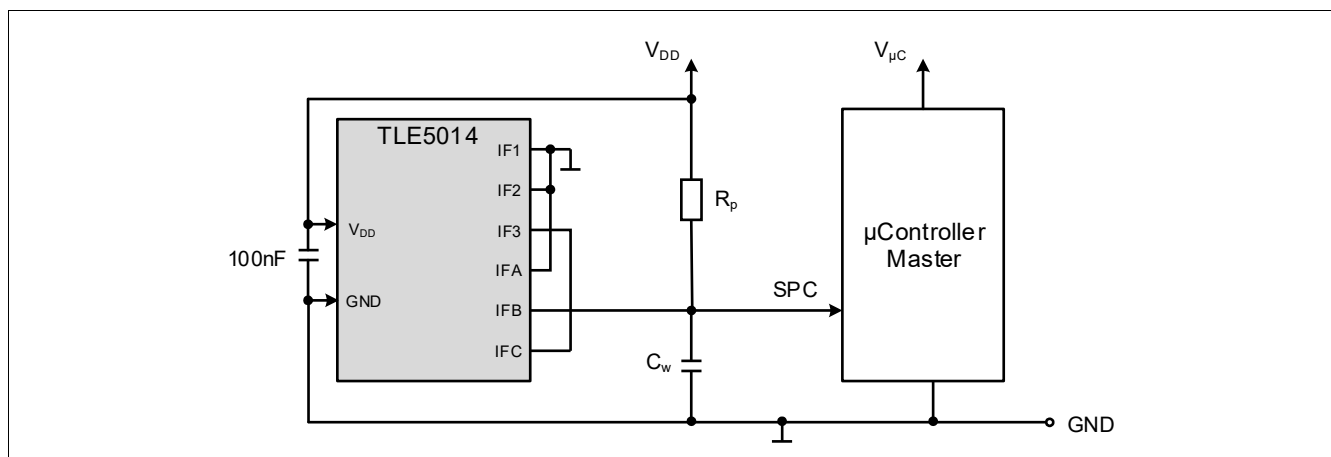


Figure 2 Application circuit for TLE5014 with SPC, SPC address is configured to be “0” (IF1 and IF2 connected to GND)

The circuit shows the minimum external circuit which is needed to operate the SPC interface. The generation of the trigger pulse (switch to pull the line low) is not shown. Depending on EMC requirements, additional circuitry may be required. The value of C_w and R_p are specified in the datasheet.

The TLE5014 can also be used with a SENT interface (**Figure 3**). The circuit shown is according to the SAE J2716-2010 SENT specification.

Application Circuits

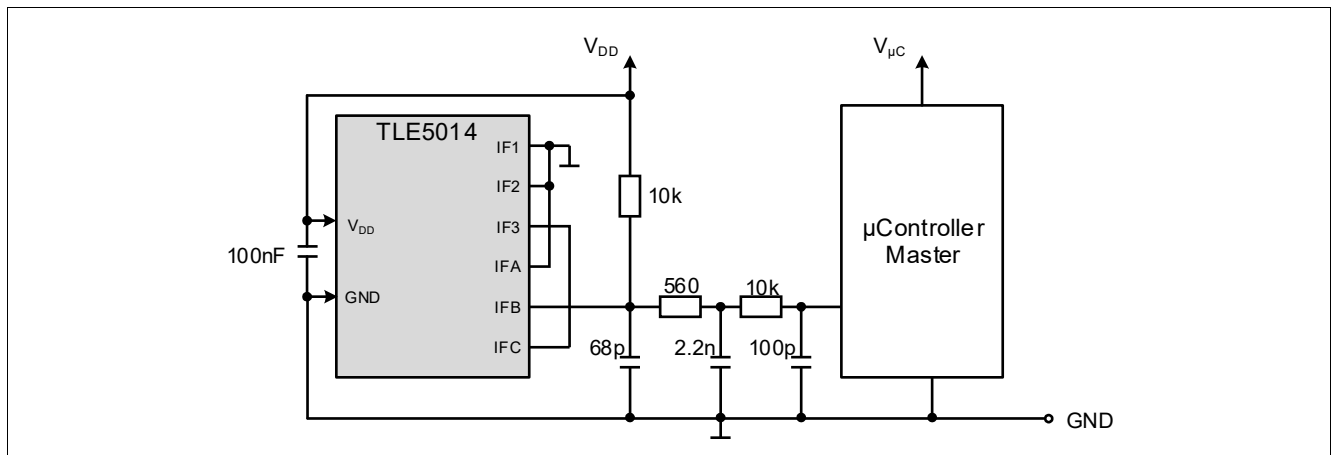


Figure 3 Application circuit for TLE5014 with SENT

Interfaces overview

2 Interfaces overview

Depending on the application, the TLE5014 is available with different interfaces:

- PWM (Pulse Width Modulation)
- SPC (Short PWM Code)
- SENT (Single Edge Nibble Transmission)
- SICI (Single-wire Interface for Calibration and Inspection; for programming purpose only)

PWM

The PWM is an unidirectional interface. Only one line is needed in which the angle value is transmitted. The angle value corresponds to the duty cycle of the signal data. A reserved duty cycle range is implemented which is used for indication of internal detected device errors. The data duty cycle range and the error duty cycle range can be programmed via EEPROM setting.

SPC

The SPC is an interface based on the SENT protocol. The ECU (master μ C) sends a trigger pulse which triggers the TLE5014 to transmit the 12-bit angle value. If desired, the temperature can also be transmitted on a slow channel (short serial message). The SPC also sends a CRC nibble and an end-pulse to terminate the communication. One line is needed for the transmission and the pins IF1, IF2 and IFC are used to set the slave number. Up to four slaves can be connected to one ECU; the ECU trigger pulse length will trigger the respective sensor.

SENT

SENT is a standardized interface described in SAE J2716-2010.

SICI

A PWM based, bidirectional interface which is used for programming purpose only.

Further details on the SENT, SPC and PWM interface can be found in the data sheet of TLE5014.

2.1 PWM Interface Configuration

The following configuration of the PWM interface is possible. Further details are in the EEPROM register description in [Chapter 4](#).

Table 2 PWM Interface configuration

Parameter	EEPROM Register	Description
PWM data duty cycle range minimum value	0x0060	This value determines the low value of the duty cycle for angle transmission. It corresponds to an angle value of 0°
PWM data duty cycle range maximum value	0x0062	This value determines the high value of the duty cycle for angle transmission. It corresponds to an angle value of 359.91°
PWM diagnostic duty cycle low	0x0064	This value determines the duty cycle which indicates a sensor failure detected by the internal safety mechanism

Interfaces overview

Table 2 PWM Interface configuration (cont'd)

Parameter	EEPROM Register	Description
PWM diagnostic duty cycle high	0x0066	This value determines the duty cycle which indicates a sensor reset or a sensor start-up error (BIST error)
PWM predivider for PWM period	0x0068	This value is used for PWM frequency adjustment according to Equation (4.1)
PWM period	0x006A	This value is used for PWM frequency adjustment according to Equation (4.1)
PWM starting edge	0x00A2	This value determines the starting edge of the PWM protocol. It can be rising or falling. In case a rising edge is selected, the PWM protocol will start with a HIGH level. In case a falling edge is selected, it will start with a LOW

Care has to be taken when programming the low and high diagnostic duty cycle that there is no overlap with the programmed data duty cycle range. In such a case, diagnostic information (e.g. sensor failure) could not be distinguished from a valid transmitted angle value. It is recommended to add a safety margin between the diagnostic and the data duty cycle range.

2.2 SPC Interface Configuration

The following table shows the different options for SPC interface configuration. The details for the EEPROM setting are shown in [Chapter 4.3](#)

Table 3 SPC Interface configuration

Parameter	EEPROM Register	Description
SPC low time	0x00A2	This value can be 5 UT or 3 UT. With 5 UT, the SPC nibble is compliant to the SENT specification
Temperature information in SPC protocol	0x00A2	The SPC frame can contain two additional nibbles for transmitting the temperature information
Rolling counter	0x00A2	A 4-bit rolling counter (rolling counter nibble) can be activated which counts the transmitted frames. This is necessary for functional safety applications to increase the diagnostic coverage
Rolling counter in CRC	0x00A2	The rolling counter information can be coded in the CRC of the SPC frame instead of being transmitted as extra nibble. In this case the rolling counter information has 2 bit only. This feature reduces overall frame length of the SPC protocol
SPC trigger	0x00A2	Two options are available for the total trigger length. It can be constant with a length of 90 UT or variable. The variable trigger length is $t_{low} + 12$ UT and depends on the trigger low time t_{low} . Both settings can be used in a bus configuration as the address of the sensor is defined by the low time t_{low} only. The advantage of the variable trigger length is that the total frame length can be reduced

Interfaces overview

Table 3 SPC Interface configuration (cont'd)

Parameter	EEPROM Register	Description
SPC short serial message	0x00A2	A short serial message (slow message) can be activated, which transmits one bit of a 16 bit information in each SPC frame and thus needs 16 SPC frames for the whole data transmission. This transmitted information is the programmed angle base and rotation direction, the chip temperature and 32 bit of the customer ID.
SPC unit time (UT)	0x00A2	The SPC unit time (UT) can be selected in the range of 1.0 μ s to 3.0 μ s.

SPC address setting in bus mode

The address of the sensor, so that it responds to a corresponding SPC trigger pulse, has to be set in the EEPROM. All parts delivered from Infineon have the default address of ID = 0 in the EEPROM. Please refer to description of EEPROM register 0x00A4 in [Description EEPROM Address 0x00A4 CRC_CFG](#) for correct setting. The address of the sensor written to the EEPROM must be identical to the address of the sensor defined via hard-wiring of the pins IF1, IF2 and IFC. [Table 4](#) shows how the sensor address has to be set using the pins IF1, IF2, IFC.

Table 4 SPC Address Configuration

Address	IF1	IF2
0	GND	GND
1	IFC	GND
2	GND	IFC
3	IFC	IFC

[Figure 4](#) shows a SPC frame with constant trigger length of 90UT, a status nibble, 3 data nibbles, a rolling counter nibble and a CRC nibble. The low time of the trigger nibble determines the address of the sensor.

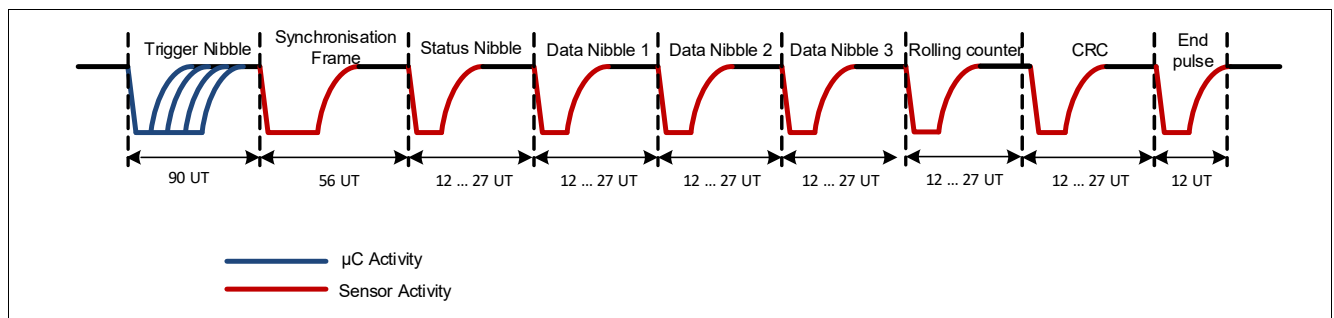


Figure 4 SPC frame with constant trigger length and rolling counter enabled as additional nibble

Interfaces overview

SPC status nibble

The 4-bit status nibble of the SPC protocol has different content depending on SPC protocol configuration:

- No short serial message:

Table 5 SPC Status Nibble

Bit	Description
0 (LSB)	Sensor ID (LSB)
1	Sensor ID (MSB)
2	Internal sensor failure
3 (MSB)	Sensor reset or start-up (BIST) failure

- Short serial message (SSM) enabled:

Table 6 SPC Status Nibble

Bit	Description
0 (LSB)	Data bit of SSM
1	Start bit of SSM
2	Internal sensor failure
3 (MSB)	Sensor reset or start-up (BIST) failure

In this configuration, the sensor ID is coded in the CRC and not available in the status nibble. There is also the option, that the rolling counter is not a dedicated nibble but the information is also coded in the CRC. Therefore, four different versions of the CRC calculation exist:

- No short serial message (SSM) and rolling counter as nibble
 Input data for the CRC calculation are:
 STATUS & DATA1 & DATA2 & DATA3 & ROLLING_CNT
- Short serial message (SSM) enabled and rolling counter as nibble
 the CRC calculation has to include a special 4-bit "ID-nibble" and is performed with the following input data:
 STATUS & DATA1 & DATA2 & DATA3 & ROLLING_CNT & ID_NIBBLE
 ID_NIBBLE = (0 0 ID(MSB) ID(LSB))
- Short serial message (SSM) is disabled and rolling counter is included in CRC (no rolling counter as nibble)
 The CRC calculation has to include a special 4-bit "RC-nibble" and is performed with the following input data:
 STATUS & DATA1 & DATA2 & DATA3 & RC_NIBBLE
 RC_NIBBLE = (RC(MSB) RC(LSB) 0 0)
- Short serial message (SSM) is enabled and rolling counter is included in CRC (no rolling counter as nibble)
 The CRC calculation has to include a special 4-bit "RC_ID-nibble" and is performed with the following input data:
 STATUS & DATA1 & DATA2 & DATA3 & RC_ID_NIBBLE
 RC_ID_NIBBLE = (RC(MSB) RC(LSB) ID(MSB) ID(LSB))

Interfaces overview

Table 7 Examples for ID Nibble, RC_Nibble and ID_RC_Nibble calculation

ID	RC	ID_Nibble	RC_Nibble	RC_ID_Nibble
00 _B	–	0000 _B	–	–
01 _B	–	0001 _B	–	–
10 _B	–	0010 _B	–	–
11 _B	–	0011 _B	–	–
–	00 _B	–	0000 _B	–
–	01 _B	–	0100 _B	–
–	10 _B	–	1000	–
–	11 _B	–	1100 _B	–
00 _B	00 _B	–	–	0000 _B
01 _B	01 _B	–	–	0101 _B
10 _B	10 _B	–	–	1010 _B
11 _B	11 _B	–	–	1111 _B

Synchronization of the Rolling Counter

The first transmitted SPC frame has a rolling counter value of 0. The next frame transmits a rolling counter value which is incremented by 1. In case the rolling counter is coded in the CRC, it is a 2-bit value. In this configuration, a synchronization of the counter value (sensor value and counter value in the micro controller) is required. The following two options are possible:

- Calculate the CRC with the received sensor data and a RC_NIBBLE of RC_NIBBLE = 0000_B. Repeat this until the calculated CRC matches with the transmitted CRC. This frame has the rolling counter value of 0. The next transmitted frame has the incremented counter value of 1 and so on. As the rolling counter has 2 bit in case it is coded in the CRC, latest the fourth transmitted frame needs to have a matched CRC. If this is not the case an error occurred.
- For the first transmitted frame, calculate the four possible CRCs with the RC_NIBBLE values 0000_B to 0011_B. The CRC which is matching the transmitted CRC has the correct rolling counter value. Use this value as starting value which is incremented with the next received frame.

CRC calculation for SPC protocol

The checksum nibble is a 4-bit CRC of the data nibbles including the status nibble. The CRC is calculated using a polynomial $x^4+x^3+x^2+1$ with a seed value of 0101_B. The remainder after the last data nibble is transmitted as CRC. The CRC calculation method is based on “recommended” implementation in the SENT standard 2010. For this “recommended” implementation, the CRC is calculated based on the input data which is then augmented with four extra zero bits and an additional CRC calculation step.

Code example:

```
//Table CRC, Poly=0xD
uint8_t cr4Table[16] = {0, 13, 7, 10, 14, 3, 9, 4, 1, 12, 6, 11, 15, 2, 8, 5};

//Example:
//Status Nibble = 0x6
//Data Nibble = 0x7, 0x4, 0x8, 0x7, 0x4, 0x8
uint8_t nibble[7] = {6, 7, 4, 8, 7, 4, 8};
uint8_t nb_nibble = 0x7;
```

Interfaces overview

```
//Seed value = 0x5
uint8_t crc = 0x5;

for(uint8_t i=0;i<nb_nibble;i++){
    crc = nibble[i] ^ cr4Table[crc];
}

//Final XOR
crc = 0x0 ^ cr4Table[crc];
```

2.3 SENT Interface Configuration

The following table shows the different options for SENT interface configuration.

Table 8 SENT Interface configuration

Parameter	EEPROM Register	Description
SENT low time	0x00A2	This value can be 5 UT or 3 UT. With 5 UT, the nibble is compliant to the SENT specification
SENT protocol type	0x00A2	Two different SENT implementations can be configured: Standard SENT: Status nibble, 3 x 4-bit data nibbles, CRC nibble Single Secure SENT: Status nibble, 3 x 4-bit data nibbles, 2x4-bit rolling counter; Inverted data nibble, CRC nibble
Sent short serial message	0x00A2	A short serial message (slow message) can be activated which transmits one bit of additional information in each SPC frame. This transmitted information is the programmed angle base and rotation direction, the chip temperature and 32 bit of the customer ID
SENT error indication	0x00A2	In case of an internal chip error, the error can be indicated only in the status nibble by setting an error bit or in the status nibble and also in the SENT data range. In this case an error code of "4091" is sent. For this option, the data range has not the full 12-bit resolution but only 1 ... 4088 LSB
Sent unit time (UT)	0x00A2	The SENT unit time (UT) can be selected in the range of 1.5 µs to 3.0 us

SENT Status Nibble

The 4-bit status nibble of the SENT protocol contains the following information:

Table 9 SENT Status Nibble

Bit	Description
0 (LSB)	error indication or start-up (BIST) error or sensor reset
1	reserved
2	short serial message bit (data bit)
3 (MSB)	short serial message bit (start indication)

Interfaces overview

CRC calculation for SENT protocol

The checksum nibble is a 4-bit CRC of the data nibbles and does not include the status nibble (different to SPC CRC implementation). The CRC is calculated using a polynomial $x^4+x^3+x^2+1$ with a seed value of 0101_B . The remainder after the last data nibble is transmitted as CRC. The CRC calculation method is based on “recommended” implementation in the SENT standard 2010. For this “recommended” implementation, the CRC is calculated based on the input data (without STATUS nibble) which is then augmented with four extra zero bits and an additional CRC calculation step.

Code example:

```
//Table CRC, Poly=0xD
uint8_t cr4Table[16] = {0, 13, 7, 10, 14, 3, 9, 4, 1, 12, 6, 11, 15, 2, 8, 5};

//Example:
//Data Nibble = 0x7, 0x4, 0x8, 0x7, 0x4, 0x8
uint8_t nibble[6] = {7, 4, 8, 7, 4, 8};
uint8_t nb_nibble = 6;
//Seed value = 0x5
uint8_t crc = 5;

for(uint8_t i=0;i<nb_nibble;i++){
    crc = nibble[i] ^ cr4Table[crc];
}

//Final XOR
crc = 0 ^ cr4Table[crc];
```

3 EEPROM Page Map

The TLE5014 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 10](#)). The algorithm for the CRC calculation is described in [Chapter 6.2.3](#).

The following [Table 10](#) 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 4](#).

The startup value of each configuration register of the TLE5014 is stored in a corresponding EEPROM register, which can be modified by the user. [Table 10](#) 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 TLE5014 datasheet.

Table 10 EEPROM page map

Page	EEPROM Register	Name	Description	Default value of Reg.
0x006	0x0060	OUT_LIM_MIN	PWM data duty cycle range minimum value	
	0x0062	OUT_LIM_MAX	PWM data duty cycle range maximum value	
	0x0064	OUT_STAT_L	PWM diagnostic duty cycle low	
	0x0066	OUT_STAT_H	PWM diagnostic duty cycle high	
	0x0068	TMR_PREDIV	PWM predivider for PWM period	
	0x006A	TMR_PERIOD	PWM period	
	0x006C	–	Reserved	
	0x006E	CRC_EEP	CRC has to be calculated with following values: page 0x002 & 0x003 & 0x004 & 0x005 & register 0x0060 & 0x0062 & 0x0064 & 0x0066 & 0x0068 & 0x006A & 0x006C & first byte of 0x006E (bit [15:8] & 0xFF00)	

EEPROM Page Map

Table 10 EEPROM page map (cont'd)

Page	EEPROM Register	Name	Description	Default value of Reg.
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	MOD2	Lock function for register PWI_MD_USR	
	0x009C	IF123_CFG	Pin configuration of pin IF1, IF2, IF3	
	0x009E	IFABC_CFG	Pin configuration of pin IFA, IFB, IFC	
0x00A	0x00A0	ANG_BASE	Angle base & rotation direction	
	0x00A2	PWI_MD_USR	Interface configuration	
	0x00A4	CRC_CFG	Configuration CRC and SPC address; 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°	

EEPROM Page Map

Table 10 **EEPROM page map** (cont'd)

Page	EEPROM Register	Name	Description	Default value of Reg.
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°	
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	[7:0] 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, will be transmitted via short serial message	
	0x00F4	CUST_ID_1	16-bit of customer ID, will be transmitted via short serial message	
	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	

EEPROM Page Map

Page 0x006	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0060	-	-	OUT_LIM_MIN														
0x0062	-	-	OUT_LIM_MAX														
0x0064	-	-	OUT_STAT_L														
0x0066	-	-	OUT_STAT_H														
0x0068	-	-	-	-	-	-	-	-	TMR_PREDIV								
0x006A	-	-	TMR_PERIOD														
0x006C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x006E	-	-	-	-	-	-	-	-	CRC_EEP								
Page 0x009																	
0x0090	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x0092	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x0094	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x0096	ACSTAT																
0x0098	MOD1																
0x009A	MOD2																
0x009C	IF123_CFG											-	-	-	-	-	-
0x009E	IFABC_CFG											-	-	-	-	-	-
Page 0x00A																	
0x00A0	ANG_BASE														ANG_DIR	-	
0x00A2	PWL_MOD																
0x00A4	-	-	-	-	-	AS_NR	S_NR_EEP	CRC_CFG									
0x00A6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x00A8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x00AA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x00AC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0x00AE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Page 0x00B																	
0x00B0	LUT_0																
0x00B2	LUT_1																
0x00B4	LUT_2																
0x00B6	LUT_3																
0x00B8	LUT_4																
0x00BA	LUT_5																
0x00BC	LUT_6																
0x00BE	LUT_7																
Page 0x00C																	
0x00C0	LUT_8																
0x00C2	LUT_9																
0x00C4	LUT_10																
0x00C6	LUT_11																
0x00C8	LUT_12																
0x00CA	LUT_13																
0x00CC	LUT_14																
0x00CE	LUT_15																
Page 0x00D																	
0x00D0	LUT_16																
0x00D2	LUT_17																
0x00D4	LUT_18																
0x00D6	LUT_19																
0x00D8	LUT_20																
0x00DA	LUT_21																
0x00DC	LUT_22																
0x00DE	LUT_23																
Page 0x00E																	
0x00E0	LUT_24																
0x00E2	LUT_25																
0x00E4	LUT_26																
0x00E6	LUT_27																
0x00E8	LUT_28																
0x00EA	LUT_29																
0x00EC	LUT_30																
0x00EE	LUT_31																
Page 0x00F																	
0x00F0	-	-	-	-	-	-	-	-	CRC_LUT								
0x00F2	CUST_ID_0																
0x00F4	CUST_ID_1																
0x00F6	CUST_ID_2																
0x00F8	CUST_ID_3																
0x00FA	CUST_ID_4																
0x00FC	CUST_ID_5																
0x00FE	CUST_ID_6																

Figure 5 TLE5014 EEPROM bitmap

EEPROM Register Description

4 EEPROM Register Description

This chapter describes the EEPROM registers which can be accessed via SICI in the EEPROM programming mode. Further registers needed for EEPROM programming are described in [Chapter 7.1](#).

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.

4.1 Page 0x006

The page 0x06 is used for the configuration of the PWM protocol. It is only relevant in case PWM is selected as interface in the register PWI_MD_USR (0x00A2). The following tables [EEPROM Register Address 0x0060 OUT_LIM_MIN](#) to [EEPROM Register Address 0x006E CRC_EEP](#) show the details.

EEPROM Register Address 0x0060 OUT_LIM_MIN

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

Field	Bits	Type	Description
OUT_LIM_MIN	13:0		Minimum value of data duty cycle range; 0% ... 100% mapped to 0 ... 16383

EEPROM Register Address 0x0062 OUT_LIM_MAX

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

Field	Bits	Type	Description
OUT_LIM_MAX	13:0		Maximum value of data duty cycle range; 0% ... 100% mapped to 0 ... 16383

EEPROM Register Address 0x0064 OUT_STAT_L

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

EEPROM Register Description

Field	Bits	Type	Description
OUT_STAT_L	13:0		Low value of diagnostic duty cycle; 0% ... 100% mapped to 0 ... 16383

EEPROM Register Address 0x0066 OUT_STAT_H

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

Field	Bits	Type	Description
OUT_STAT_H	13:0		High value of diagnostic duty cycle; 0% ... 100% mapped to 0 ... 16383

EEPROM Register Address 0x0068 TMR_PREDIV

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

Field	Bits	Type	Description
TMR_PREDIV	7:0		For calculation of PWM frequency according to Equation (4.1)

EEPROM Register Address 0x006A TMR_PERIOD

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

Field	Bits	Type	Description
TMR_PERIOD	13:0		For calculation of PWM frequency according to Equation (4.1)

EEPROM Register Address 0x006E CRC_EEP

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

EEPROM Register Description

Field	Bits	Type	Description
CRC_EEP	7:0		CRC has to be calculated with following values: page 0x002 & 0x003 & 0x004 & 0x005 & register 0x0060 & 0x0062 & 0x0064 & 0x0066 & 0x0068 & 0x006A & 0x006C & first byte of 0x006E (bit [15:8])

The PWM frequency of the PWM output can be programmed with the two parameters TMR_PREDIV and TMR_PERIOD according to the following **Equation (4.1)**:

(4.1)

$$f[\text{Hz}] = \frac{40000000}{(1 + \text{TMR_PREDIV})(\text{TMR_PERIOD})}$$

Example:

For TMR_PREDIV = 12, TMR_Period = 15384, the PWM frequency calculates to $f = 200$ Hz.

4.2 Page 0x009

Care has to 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.

Description EEPROM Register Address 0x0098 MOD1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		RST_3_5	CLK_SEL	FIR_MD			LUT_MD				IFMD_ABC				IFMD_123

Field	Bits	Type	Description
RST_3_5	13		0: Reset comparator set to < 3.0 V 1: Reset comparator set to < 4.2 V
CLK_SEL	12		For test purpose only
FIR_MD	11		Internal update rate of angle calculation: 0: 25.6 μs 1: 51.2 μs
LUT_MD	9:8		Look-up table (LUT) configuration: 00: LUT disabled 01: LUT range = 360° (angle steps = 11.25°) 10: LUT range = 180° (angle steps = 5.625°) 11: LUT range = 90° (angle steps = 2.813°)

EEPROM Register Description

Field	Bits	Type	Description
IFMD_ABC	5:4		For test purpose only
IFMD_123	0		For test purpose only

Description EEPROM Register Address 0x009A MOD2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD2															

Field	Bits	Type	Description
MOD2	15:0		Lock register for 0x00A2. The bits in this registers are connected with a logical AND function with the corresponding bit in 0x00A2. This means that a “0” in a bit of register 0x009A prohibits to set a “1” in the corresponding bit of 0x00A2. Note: The bit will show as “1” in 0x00A2 but has no effect

Description EEPROM Register Address 0x009C IF123_CFG

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

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
IF3_OD	11		IF3 open drain enable: 0 _B Disable (Push-Pull) 1 _B Enable (Open Drain)

EEPROM Register Description

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

Description EEPROM Register Address 0x009E IFABC_CFG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFB_SLP				IFC_OD	IFC_CFG	IFB_OD	IFB_CFG	IFA_OD	IFA_CFG						

Field	Bits	Type	Description
IFB_SLP	15:14		Slope control of output signal: 00: _B 2 μ s 01: _B 4 μ s 10: _B 6 μ s 11: _B Tristate
IFC_OD	11		IFC open drain enable: 0: _B Disable (Push-Pull) 1: _B Enable (Open Drain)
IFC_CFG	10		For test purpose only
IFB_OD	9		IFB open drain enable: 0: _B Disable (Push-Pull) 1: _B Enable (Open Drain with edges control)
IFB_CFG	8		For test purpose only

EEPROM Register Description

Field	Bits	Type	Description
IFA_OD	7		IFA open drain enable: 0: _B Disable (Push-Pull) 1: _B Enable (Open Drain)
IFA_CFG	6		For test purpose only

4.3 Page 0x00A

Care has 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.

Description EEPROM Address 0x00A0 ANG_BASE

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

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
Reserved	0		0: _B Reserved bit, shall not be changed

Description EEPROM Address 0x00A2 PWI_MD_USR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWI_MODE		PWM_EDGE	SENT/SPC_LT_5_3	SPC_TEMP	SPC_RCNIB	SPC_RCCRC	SPC_TRIG	SENT_PROMO	SENT_ERRSIG	SENT_PAPU	SENT/SPC_PWI_SSM	SENT/SPC_UT			

EEPROM Register Description

Field	Bits	Type	Description
PWI_MODE	15:13		Interface selection (Only available for Product Type: TLE5014F16D): 100 _B PWM 010 _B SENT 001 _B SPC
PWM_EDGE	12		PWM starting edge: 0 _B Rising edge 1 _B Falling edge
SENT/SPC_LT_5_3	11		SENT/SPC low time: 0 _B 5 UT 1 _B 3 UT
SPC_TEMP	10		Temperature information in SPC protocol: 0 _B No temperature nibble 1 _B Two temperature nibbles
SPC_RCNI	9		Rolling counter: 0 _B Rolling counter nibble enabled 1 _B No rolling counter nibble
SPC_RCCRC	8		Rolling counter in CRC: 0 _B No rolling counter in CRC 1 _B Rolling counter in CRC
SPC_TRIG	7		SPC bus mode: 0 _B 90 UT constant trigger length 1 _B Variable trigger length
SENT_PROMO	6		SENT protocol type: 0 _B Single secure sensor (with rolling counter and inverted data nibble) 1 _B Standard SENT (without rolling counter and inverted data nibble)
SENT_ERR_SIG	5		SENT error indication: 0 _B Error code “4091” in data range 1 _B Error indication in status nibble only
SENT_PAPU	4		Pause pulse activation: 0 _B No pause pulse 1 _B Pause pulse activated

EEPROM Register Description

Field	Bits	Type	Description
SENT/SPC_PWI_SSM	3		SENT/SPC short serial message: 0: _B No short serial message 1: _B Short serial message activated
SENT/SPC_UT	2:0		SENT/SPC unit time setting: 000: _B 3 μ s 001: _B 2.5 μ s 010: _B 2.0 μ s 011: _B 1.5 μ s 100: _B 1.0 μ s

Description EEPROM Address 0x00A4 CRC_CFG

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

Field	Bits	Type	Description
AS_NR	10		0: _B Address configuration for SICI as defined in bits S_NR_EEP 1: _B Address configuration for SICI as defined by hard wiring of pins <i>Note: This bit shall be set to "1", otherwise no programming in bus mode is possible. SPC protocol is always using the values from bits S_NR_EEP</i>
S_NR_EEP	9:8		Address coding of SPC slave: 00: _B ID = 0 01: _B ID = 1 10: _B ID = 2 11: _B ID = 3
CRC_CFG	7:0		Configuration CRC and SPC address; CRC has to be calculated with following values: registers 0x0096 & 0x0098 & 0x009A & 0x009C & 0x009E & 0x00A0 & 0x00A2 & first byte of 0x00A4 (bit [15:8])

4.4 Page 0x00B - 0x00E

Look-up table, see [Table 10](#).

4.5 Page 0x00F

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

Description EEPROM Address 0x00F0 CRC_LUT

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

CRC_LUT

Field	Bits	Type	Description
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.

Working Register Description

5 Working Register Description

This chapter describes the working registers which can be read with the SICI interface. These registers are updated with the time t_{update} , which is approximately 25 μs .

5.1 STAT Register 0x00: Status

This register provides detailed information on sensor status and internal sensor errors.

Register Address 0x00: STAT (Status)

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_PLA USI	S_RST
u	u	u	u			u	u	u	u	u	u	u	u	u	u

Field	Bits	Type	Description
RD_ST	15	u	Read bit that indicates that the bits in the STAT-line are updated since the last readout
RES	14	u	Reserved
S_IF	13	u	Bit is set when a interface issue happened
S_VEC	12	u	Bit indicates when the vector length of the actual angle_vector exceeds the defined limits
RES	11		Reserved
RES	10		Reserved
S_EEP	9	u	Bit indicates whenever the EEPROM initiates an error
S_CRC	8	u	Bit is set whenever one of the bit error detection functions is alarming
S_PROG	7	u	Bit indicates a DSP error
S_BIST	6	u	Bit is set whenever one of the startup-BIST is failing. (BIST: Built in self test)
S_ADCT	5	u	Bit is set when the temperature is out of limits or the temperature delta check is failing
S_ADC	4	u	Bit is set when the ADC_raw_max or ADC_raw_delta check fails or the filter-scheduling check is alarming

Working Register Description

Field	Bits	Type	Description
S_VR	3	u	Bit is a combination of several voltage checks
S_WD	2	u	Bit is set when the last reset happened due to a watchdog event
S_PLAUSI	1	u	Bit is set when the redundant angle calculation is failing the limits
S_RST	0	u	Reset occurred

5.2 AVAL Register 0x02: Angle Value

The current angle value can be read via SICI interface in register AVAL at address 0x02. It is updated approx. every 25 μ s, which is the time needed for sampling and calculating a new angle value. It is a signed 15-bit value (see [Description Register Address 0x02 AVAL](#)).

Description Register Address 0x02 AVAL

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 bit that indicates that the bits in the AVAL register are updated since last readout by interface
AVAL	14:0		Signed 15-bit internal angle value: $0^\circ = 0x0000$ $-180^\circ = 0x4000$ $179.99^\circ = 0x3FFF$

5.3 ASPD Register 0x03: Angle Speed

This register provides the actual angle speed.

Register Address 0x03: Angle speed

Read command: 0xC031

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

Working Register Description

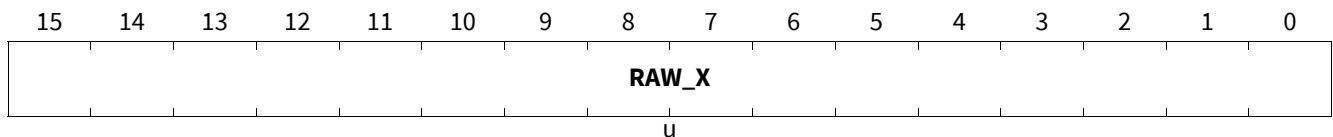
Field	Bits	Type	Description
RD_ASPD	15	u	Read angle speed Read bit that indicates that the bits in the ASPD-register are updated since last readout by interface
RES	14:12		Reserved
ASPD	11:0	u	Angle speed Calculated angle speed signed 12-bit value $\text{speed}[^{\circ}/\text{s}] = \frac{\frac{360^{\circ}}{65536}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

5.4 RAW_X Register 0x06: X raw value

This register provides the x-row value (cosine value).

Register Address 0x06: X raw value

Read command: 0xC061



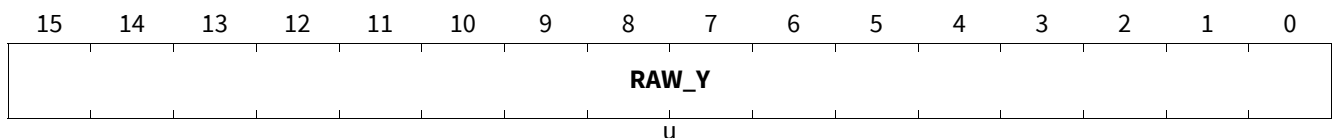
Field	Bits	Type	Description
RAW_X	15:0	u	Unsigned 16-bit value of X-row

5.5 RAW_Y Register 0x07: Y raw value

This register provides the y-row value (sine value).

Register Address 0x07: Y raw value

Read command: 0x8071



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

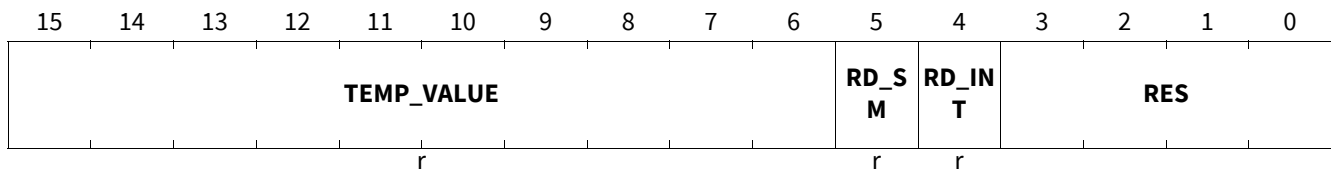
Working Register Description

5.6 TEMP Register 0x0A: Temperature raw value

This register provides the actual raw temperature value. To calculate the actual temperature, the temperature offset has to be determined. This is done by reading the register once at a known temperature (e.g. 25°C). This value is then TEMPER_25 and the offset corrected temperature can be calculated using the formula below.

Register Address 0x0A: temperature raw value

Read command: 0xC0A1



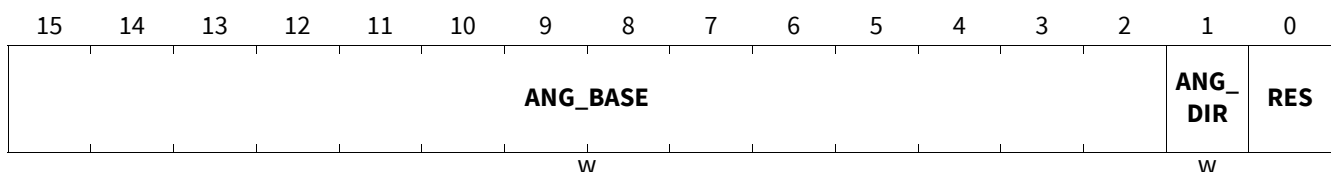
Field	Bits	Type	Description
TEMP_VALUE	15:6	r	Temperature value 10-bit temperature value, not offset corrected $T [^{\circ}\text{C}] = (\text{TEMP_VALUE} - \text{TEMPER_25}) / 1.3815 [\text{LSB}/^{\circ}\text{C}] + 25^{\circ}\text{C}$ TEMPER_25: TEMP_VALUE @25°C In case the offset correction is not done at 25°C but at a different temperature, the formula has to be modified accordingly.
RD_SM	5	r	Read state machine Bit is set when new value is available since state machine has read last one
RD_INT	4	r	Read interface Bit is set when new value is available since interface has read last one
RES	3:0		Reserved

5.7 ANG_BASE register 0x14: Angle base and rotation direction

This register contains the angle base. The output angle is referring to this angle base value.

Register Address 0x14: angle base and rotation direction

Read command: 0xC141



Working Register Description

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

6 SICI

6.1 Basic Functionality

The SICI interface (SICI: Single-Wire Interface for Calibration and Inspection) is a one-wire programming interface that allows read and write access to the working registers and EEPROM registers of the sensor. It is bidirectional for every transferred bit, meaning that for every single bit sent by the master one bit is immediately sent back from the slave. The speed of the interface is automatically determined from the length of the first bit sent by the master.

The one-wire interface enables a robust data transmission at a speed of up to 240 kbit/s. It is a voltage interface based on a bit-wise pulse-width-modulation (PWM) principle and requires an open-drain driver on micro controller side. Regardless of the interface configuration of the TLE5014, the SICI interface is always available on the IFB pin during the startup time.

In order to use the SICI interface, sensor and micro controller have to use an open drain output, so they can actively pull a “low” level. The communication line has to be connected to V_{DD} by a pull-up resistor of approx. 1.5-2 k Ω . Higher values for the pull-up resistor may lead to problems to achieve the necessary speed of the communication.

6.2 SICI Data Transfer

The communication via the SICI interface is based on transmitting a single bit to the sensor and immediately receiving a bit. These bits form a 16-bit word. It makes the interface bit-synchronous, robust and very flexible in timing. Bidirectional data transmission can thus be realized using only a single wire without a trimmed oscillator on sensor or master side. To enhance the robustness of the interface, the sensor repeats the 16-bit command word sent by the master in order to have a confirmation of the correct understanding of the message, and transmits a safety word after each communication. Additionally, a time-out feature is implemented: If during a communication the sensor does not receive a bit from the master within a certain time span (see [Table 11](#), Interface reset time), the interface is reset and a new command can be sent.

6.2.1 Bit Encoding

For the SICI interface, bits are encoded as the difference in duration of consecutive “low” and “high” levels. To transmit a bit to the sensor, the master sends one single low/high PWM signal with a period T , then pulls the line to “low” again for a short time T_3 . The logic value of the bit is then encoded as the difference between “low” and “high” time:

- to transmit a “0” to the sensor, the master pulls the line “low” for a short time T_1 , then releases it to “high” for a long time $T - T_1$ (typically $T_1 = 0.3 \cdot T$)
- to transmit a “1” to the sensor, the master pulls the line “low” for a long time T_2 , then releases it to “high” for a short time $T - T_2$ (typically $T_2 = 0.7 \cdot T$)

The sensor recognizes the total bit-time interval T as the duration between two consecutive falling edges from the master. This encoding scheme is illustrated in [Figure 6](#).

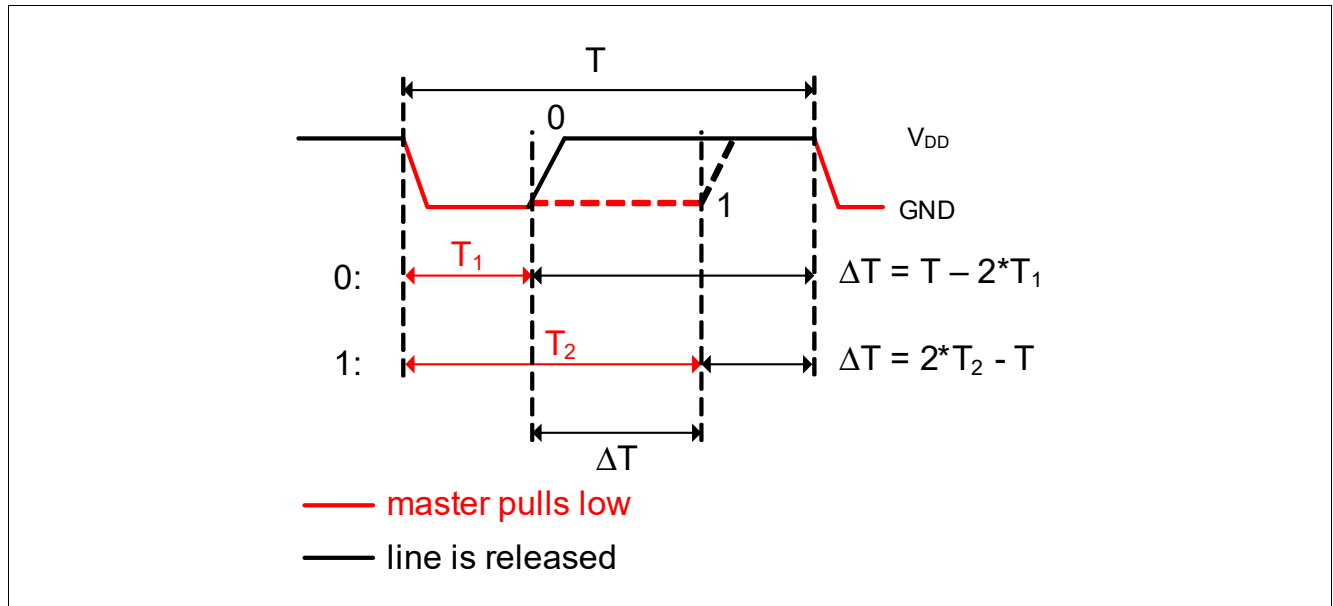


Figure 6 Timing description of master bit

After receiving a bit from the master, the sensor answers by transmitting one bit:

- the sensor transmits a “1” by pulling the SICI-line “low” for the time difference ΔT between “high” and “low” level given by the received PWM signal (i.e. if a “0” was transmitted by the master, $\Delta T = T - 2 \cdot T_1$, whereas if a “1” was transmitted by the master, $\Delta T = 2 \cdot T_2 - T$)
- the sensor transmits a “0” by keeping the line “high”

To read the bit transmitted by the sensor, the master has to check the level of the line after T_3 (master low pulse) but before time ΔT has expired (recommended 80% of ΔT , see [Figure 7](#)).

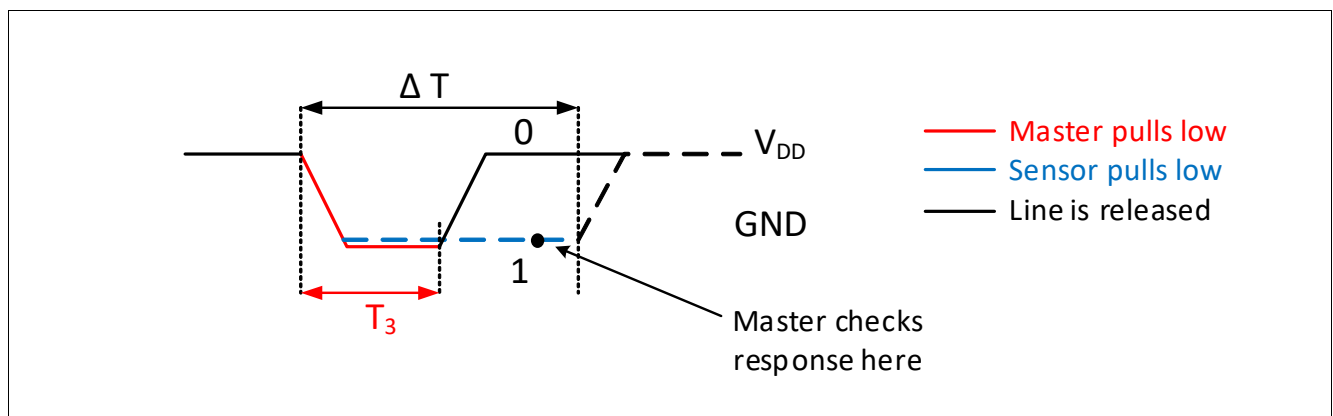


Figure 7 Timing description of response bit from sensor

After receiving the response bit from the sensor, the master can pause for a time T_4 before transmitting the next bit. T_4 has to be shorter than the time-out limit (see [Table 11](#), interface reset time), otherwise the interface is reset. [Figure 8](#) shows an example communication between master and sensor.

SICI

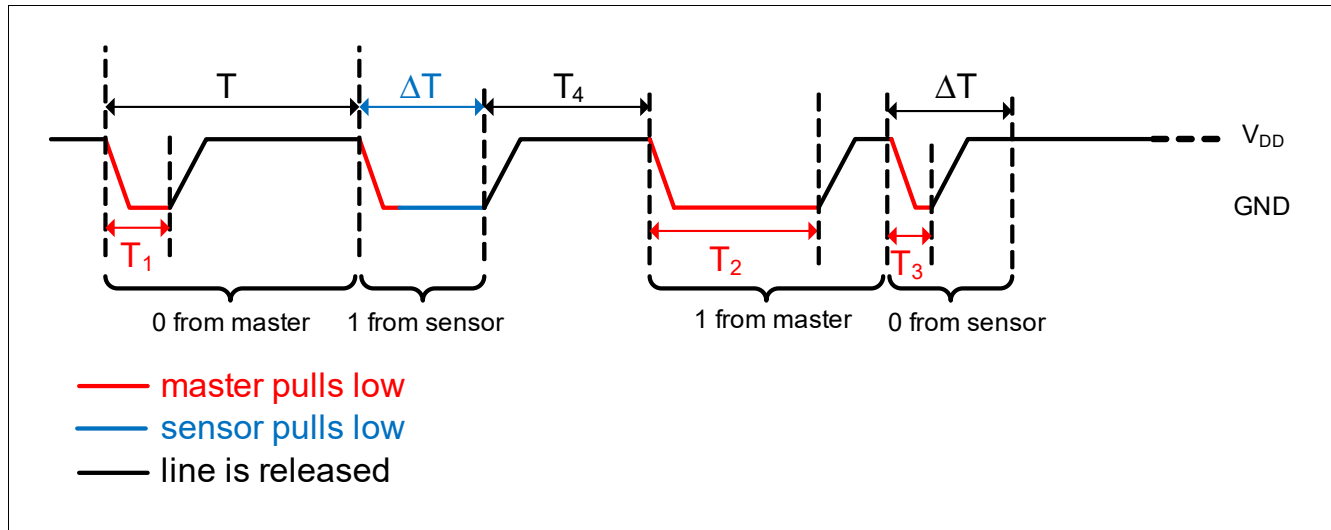


Figure 8 SICI communication example

The transmission rate of the interface is determined by the width T of the PWM signal sent by the master. A maximum transmission speed of 240 kbit/s can be reached for $T = 2 \mu\text{s}$. Lowering the transmission rate generally increases the robustness of the communication in distorted environments and/or with high capacitive loads on the line. The optimum communication speed thus depends on the application circuitry. The timing specification of the SICI interface is given in [Table 11](#).

Table 11 SICI interface timing and level specification

Parameter	Symbol	Values			Unit	Note or Test Condition
		Min.	Typ.	Max.		
Master PWM period	T	2 ¹⁾	–	980	μs	Determines interface transmission rate ²⁾
Master low time to transmit “0”	T_1	$0.2 * T$	–	$0.33 * T$	μs	²⁾
Master low time to transmit “1”	T_2	$0.66 * T$	–	$0.8 * T$	μs	²⁾
Time difference between master low and high level	ΔT	$0.33 * T$	–	$0.6 * T$	μs	$\Delta T = T - 2 * T_1$; for “0” $\Delta T = 2 * T_2 - T$; for “1” ²⁾
Master low pulse after PWM bit	T_3	0.4	–	$0.25 * T$	μs	²⁾³⁾
Master pause time after sensor bit response	T_4	$0.75 * T$	–	810	μs	²⁾³⁾
Interface reset time	T_{Res}	–	820	1640	μs	²⁾⁴⁾
Input signal low level	$V_{\text{low,in}}$	–	–	$0.3 * V_{\text{DD}}$	V	²⁾
Input signal high level	$V_{\text{high,in}}$	$0.7 * V_{\text{DD}}$	–	–	V	²⁾
Output signal low level	$V_{\text{low,in}}$	–	–	0.9	V	²⁾
SICI line pull-up resistor	R_{pu}	1	–	–	k Ω	
Wait time between transmitted 16-bit words	t_{delay}	1	–	–	μs	²⁾

1) Achievable transmission rate (minimum master PWM period) depends on parasitic capacities in external circuitry.

2) Verified by design/characterization.

3) Rise/fall times due to parasitic capacitances on the line have to be added.

4) Max. time for reset calculated for worst case interruption of SICI transmission

SICI

To enter the programming mode of TLE5014 some dedicated commands have to be sent within the start-up time of the sensor (see [Chapter 6.2.4](#)). Thus, not all possible settings of the SICI timing parameters (e.g. T , T_1 , T_2 , ...) can be used with TLE5014. An example setting of the timing parameters for SICI interface, which allows to send all necessary SICI commands within the start-up time, is shown in [Table 12](#).

Table 12 Example of SICI timing parameters

Parameter	Symbol	Value	Unit	Remarks
Master PWM period	T	21.1	μs	
Master low time to transmit "0"	T_1	6.6	μs	
Master low time to transmit "1"	T_2	14.5	μs	
Time difference between master low and high level	ΔT	7.9	μs	$\Delta T = T - 2 * T_1$; for "0" $\Delta T = 2 * T_2 - T$; for "1"
Master low pulse after PWM bit	T_3	3.5	μs	
Master pause time after sensor bit response	T_4	25	μs	

6.2.2 Communication Protocol

The master transmits a 16-bit command word to read or write registers, then sends or receives a number of data words and then receives a safety word from the sensor. Each transmission starts with the most-significant-bit (MSB). The register map is shown in [Chapter 3](#). [Table 13](#) shows the structure of the command word.

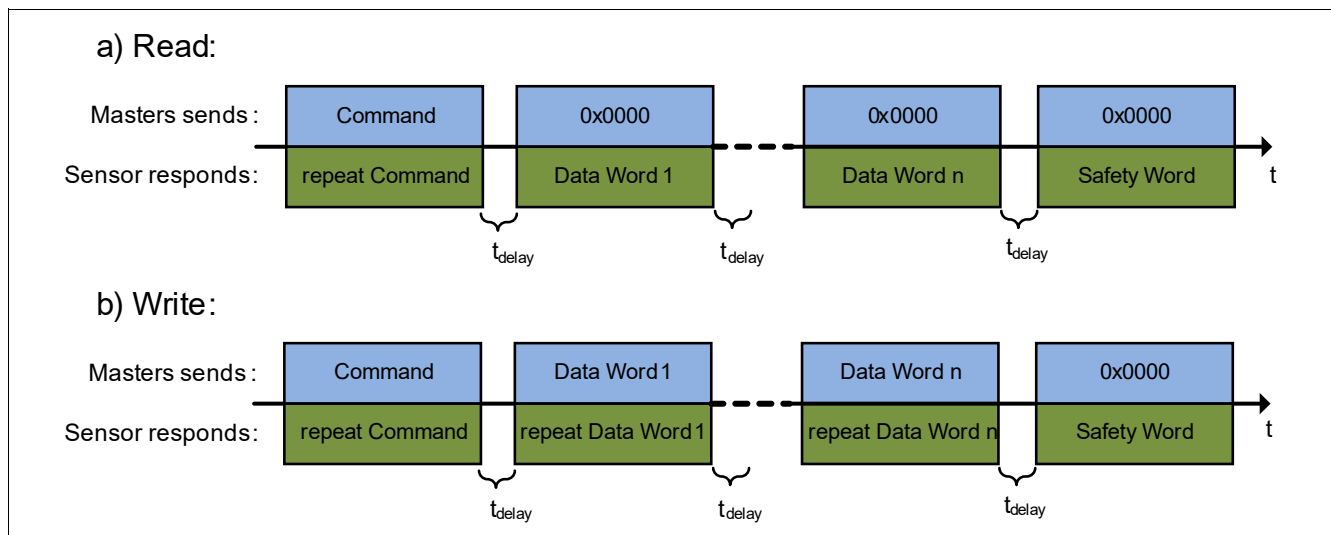


Figure 9 SICI data transfer for read command (a) and write command (b)

SICI

Table 13 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 including parity has to be odd
CMD	[13]	Command Mode 0: ADDR field is interpreted as register address to be read/written
ACCESS	[12..11]	Access mode to registers READ: 00 _B : Direct readout 01 _B : Update all registers and then readout 1x _B : Readout only updated registers WRITE: 00 _B : Direct write 10 _B : Reset active bits (ones in written data) 11 _B : Set active bits (ones in written data)
ADDR	[10..4]	7 bit address
LEN	[3..0]	4 bit length; number of data words to read/write

If a read command is transmitted, the master has to send a 16-bit sequence of 0's for every requested word to be read. The sensor then responds to each “0” from the master with a bit from the register word to be read, starting with the MSB. If a write command is transmitted, the sensor repeats every single bit of the data words. After the transmission of the data words, the master has to send another 16-bit sequence of 0's, and the sensor responds to each “0” with one bit of the safety word, starting with the MSB. Between each transmitted word, there has to be a delay t_{delay} to allow the sensor to process the data. **Figure 9** illustrates the sequence of command, data and safety words for read/write access.

A safety word is sent at the end of each communication. The structure of the safety word is shown in **Table 14**. An internally detected sensor fault is indicated in the safety word.

It is recommended to read the status register STAT (**Chapter 5.1**) in this case to get further information of the error. Reading of the STAT register causes a reset of the error bit in the safety word. If the status register is not read, the error indication will persist in the safety word, even when the error is no longer existing

Table 14 Structure of the Safety Word

Name	Bits	Description
STAT	Chip and Interface Status	
	[15]	Indication of Chip-Reset or watchdog was activated 0: Reset occurred or watchdog was activated 1: No reset
	[14]	System Error (e.g. Overvoltage; Undervoltage; GND- off; CRC) 0: Error occurred 1: No error
	[13]	Interface Access Error (access to wrong address; wrong lock, wrong parity) 0: Error occurred 1: No error
	[12]	Valid Angle Value (no ADC error, no internal plausibility check error) 0: Angle value invalid 1: Angle value valid
RESP	[11..8]	Sensor Number Response Indicator The sensor number bit is pulled low and the other bits are high.
CRC	[7..0]	Cyclic Redundancy Check (CRC) includes command word, data words, STAT & RESP

6.2.3 CRC generation

The TLE5014 uses a cyclic redundancy check (CRC) to confirm the correct processing of read and write commands via the safety word, and to monitor the integrity of configuration registers and EEPROM pages. The CRC is generated 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: $X^8+X^4+X^3+X^2+1$; for the CRC generation the fast-CRC generation circuit is used (see [Figure 10](#))
- The remainder of the fast CRC circuit is initially set to '11111111_B'
- Remainder is inverted before transmission

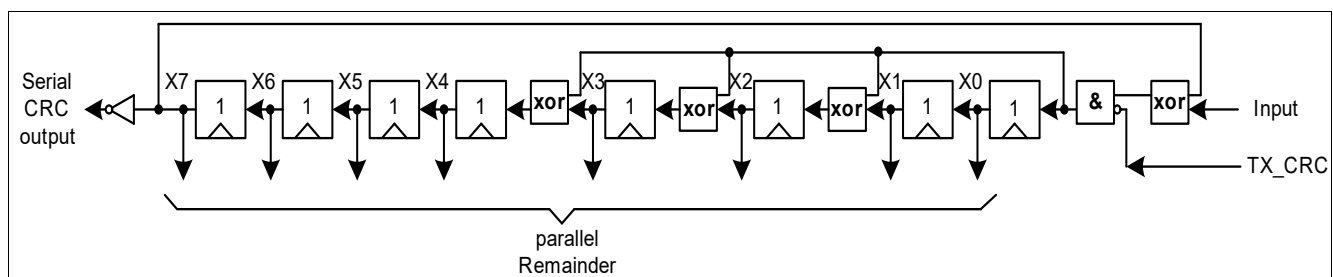


Figure 10 Fast CRC polynomial division circuit

6.2.4 SICI Application Example: Reading working register

Sensor and micro controller must use an open-drain output, so they can actively pull only a 0 Volt level; the 5 Volt level is achieved by an external pull-up resistor.

SICI

In the application, the chip will switch into SENT/SPC/PWM mode (depending on pre programmed configuration) after startup. Therefore the SICI can only be entered during startup and internal settings have to be modified to keep the SICI in case it shall be used for programming purpose.

The following sequence has to be performed to read a register (angle value register AVAL 0x02 in this example):

Power on the sensor, wait at least 10 μ s to allow the internal voltages to stabilize:

- Set the KEEP_IF bit to ensure the SICI is not disabled after startup:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0x4000 (keep SICI after startup)
 - Read Safety word
- Wait until startup is finished; refer to the TLE5014 data sheet for power-on time specification
- The 15-bit angle value can be read to get data for the angle base or look-up table programming:
 - READ-CMD to 0x02: 0x8021
 - Read DATA (send 0x0000)
 - Read Safety word

6.2.5 SICI Application Example: EEPROM read/write

Sensor and micro controller must use an open-drain output, so they can actively pull only a 0 V level; the 5 V level is achieved by an external pull-up resistor.

In the application, the chip will switch into SENT/SPC/PWM mode (depending on pre programmed configuration) after startup. Therefore the SICI can only be entered during startup and internal settings have to be modified to keep the SICI in case it shall be used for programming purpose.

The following sequence has to be performed to allow an EEPROM read/write via SICI:

- Power on the sensor, wait at least 10 μ s to allow the internal voltages to stabilize
- Write unlock word 0x4711 to 0x77 to enable access to the registers:
 - WRITE-CMD to 0x77: 0x0771
 - DATA: 0x4711 (unlock)
 - Read Safety word
- Set the KEEP_IF bit to ensure the SICI is not disabled after startup:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0x4000 (keep SICI after startup)
 - Read Safety word
- Wait until startup is finished; refer to the TLE5014 data sheet for power-on time specification
- For reading or writing EEPROM content, the internal controller has to be switched off:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0xC000 (switch off DSP)
 - Read Safety word
- Now the EEPROM-reading / writing can be entered (see [Chapter 7](#))
- After writing/reading the EEPROM, send a write command to the ACSTAT register (address 0x01) with data word 0x0001 to trigger a chip reset:
 - WRITE-CMD to 0x01: 0x4011
 - DATA: 0x0001 (chip reset)

SICI

6.3 SICI Programming in Bus Mode

In case several sensors (up to 4) are connected in a bus mode with the SPC interface, the EEPROM read/write via SICI has to be performed in the bus mode configuration. For this purpose, to each individual sensor an address has to be assigned which is done by the external wiring of the pins IF1, IF2 and IFC on the PCB as described in the data sheet and in [Table 4](#) of this document.

To address the individual sensors with SICI, one of the following ID_WORD has to be sent according to the hard-wired sensor ID:

ID=0: ID_WORD = 0x05AC

ID=1: ID_WORD = 0x05A9

ID=2: ID_WORD = 0x05A6

ID=3: ID_WORD = 0x05A3

The following sequence has to be performed to allow an EEPROM read/write via SICI in bus mode. The commands in blue have to be added to the command sequence described in [Chapter 6.2.4](#) for reading/writing in bus mode and contain the ID_WORD for sensor addressing: Power on the sensor, wait at least 10 μ s to allow the internal voltages to stabilize:

- Write unlock word 0x4711 to 0x77 to enable access to the registers:
 - WRITE-CMD to 0x77: 0x0771
 - DATA: 0x4711 (unlock)
 - Read Safety word
- Set the KEEP_IF bit to ensure the SICI is not disabled after startup:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0x4000 (keep SICI after startup)
 - Read Safety word
- Write sensor ID to SICI-DISABLE register at 0x7A:
 - WRITE-CMD to 0x7A: 0x47A1
 - DATA: ID_WORD
 - Read Safety word
- Wait until startup is finished; refer to the TLE5014 data sheet for power-on time specification
- For the programming, the internal controller has to be switched off:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0xC000
 - Read Safety word
- Now the EEPROM-reading / writing can be entered (see [Chapter 7](#))
- After writing/reading the EEPROM, send a write command to the ACSTAT register (address 0x01) with data word 0x0001 to trigger a chip reset:
 - WRITE-CMD to 0x01: 0x4011
 - DATA: 0x0001 (chip reset)

7 EEPROM Programming

The EEPROM of the TLE5014 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 onto a range of registers, which can then be read or written via the SICI interface.

7.1 Writing and Reading of EEPROM Pages

To read or write data from the EEPROM, the desired page has first to be mapped to working registers EEP_0 to EEP_7 (address 0x60 to 0x67) by writing the desired page number (0..15) to the EEP_PAGE register (address 0x68, see [Table 13](#)). The content of the EEPROM page can then be read from registers EEP_0 to EEP_7.

Before EEPROM content can be read or written, the steps describes in [Chapter 6.2.5](#) have to be performed.

Programming the EEPROM is done by sending the following command sequence (example for EEPROM page 0x009):

- Map the desired EEPROM page to the registers EEP_0 to EEP_7 (0x60 to 0x67) by writing the page number into the EEP_PAGE register 0x68 (example for EEPROM page 0x006):
 - WRITE-CMD to 0x68: 0x4681
 - DATA: 0x0009 (maps EEPROM page 0x009 to working registers 0x60 ... 0x67)
- Read the content of the desired EEPROM page by sending a read command to registers EEP_0 to EEP_7, then change the desired bits and write the content back to registers EEP_0 to EEP_7:
 - READ-CMD to 0x60: 0xC601 (reads working register 0x60 = EEP_0)
 - DATA: content of working register 0x60 (corresponds to EEPROM register 0x0090)
 - ...
 - READ-CMD to 0x67: 0x8671 (reads working register 0x67 = EEP_7)
 - DATA: content of working register 0x67 (corresponds to EEPROM register 0x009E)
- To change EEPROM content, first overwrite register EEP_0 to EEP_7 (0x60 to 0x67) with the desired content:
 - WRITE-CMD to 0x60: 0x0601
 - DATA: 0xFFFF (example for writing 0xFFFF in EEPROM register 0x0090)
 - ...
- To program the data from registers EEP_0 to EEP_7 back into the EEPROM page, the following commands have to be sent:
 - WRITE-CMD to 0x68: 0x4681
 - DATA: 0x03FF
 - WRITE-CMD to 0x67: 0x4671 (program with automatic verification)
 - DATA: 0x0009
- After writing/reading the EEPROM, send a write command to the ACSTAT register (address 0x01) with data word 0x0001 to trigger a chip reset:
 - WRITE-CMD to 0x01: 0x4011
 - DATA: 0x0001 (chip reset)

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 2.3. The addresses of the checksum registers and the register range covered by each checksum are listed in [Table 10](#).

EEPROM Programming

Changing the configuration of the TLE5014 without updating the corresponding checksum results in a CRC error, which is indicated.

EEPROM Timing

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

Table 15 EEPROM timing specification

Parameter	Symbol	Values			Unit	Note or Test Condition
		Min.	Typ.	Max.		
Read time per word	$t_{\text{read_ac}}$	–	–	150	ns	1)
Write time ²⁾	t_{write}	4.0	5.0	6.0	ms	2)
Erase time ²⁾	t_{erase}	4.0	5.0	6.0	ms	2)
Program time ²⁾	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.

7.2 Verification of EEPROM

To verify if the programming 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.

EEPROM writing verification, address 0x69

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		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

7.3 Locking of EEPROM

After programming the EEPROM, the content can be locked. In this case, no second programming is possible. The following commands show how to lock the EEPROM:

EEPROM Programming

- Power on the sensor, wait at least 10 μ s to allow the internal voltages to stabilize
- Write unlock word 0x4711 to 0x77 to enable access to the registers:
 - WRITE-CMD to 0x77: 0x0771
 - DATA: 0x4711 (unlock)
- Set the KEEP_IF bit to ensure the SICI is not disabled after startup:
 - WRITE-CMD to 0x6C: 0x06C1
 - DATA: 0x4000 (keep SICI after startup)
- Write 0x03F0 to "EEP_Page register" (Adr. 0x68):
 - WRITE-CMD to 0x68: 0x4681
 - DATA: 0x03F0
- Write 0x7FFF to "EEP_2 register (Adr. 0x62):
 - WRITE-CMD to 0x62: 0x4621
 - DATA: 0x7FFF
- Write 0x7FFF to "EEP_3 register (Adr. 0x63):
 - WRITE-CMD to 0x63: 0x0631
 - DATA: 0x7FFF
- Write 0x03FF to "EEP_Page register (Adr. 0x68):
 - WRITE-CMD to 0x68: 0x4681
 - DATA: 0x03FF
- Write 0x000C to EEP_7 register (Adr. 0x67):
 - WRITE-CMD to 0x67: 0x4671
 - DATA: 0x000C
- Send a write command to the ACSTAT register (address 0x01) with data word 0x0001 to trigger a chip reset:
 - WRITE-CMD to 0x01: 0x4011
 - DATA: 0x0001 (chip reset)

8 Initial Calibration

This chapter describes which initial calibration has to be performed with the TLE5014.

8.1 Zero Angle Calibration

The zero angle (or angle base) is stored in the bits [15:3] of address 0x00A0 of the EEPROM as a 14-bit value (see [Description EEPROM Address 0x00A0 ANG_BASE](#)). The sensor output angle is in reference to this angle base. In the application, the angle base has to be programmed by the customer.

The following procedure for angle base calibration is recommended (external reference required):

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
4. Calculation of ANG_BASE parameter:
 - If angle values were read out with SICI:
shift right by 1 bit resp. round angle values to 14 bit
 - If angle values were read out with SPC/SENT/PWM:
shift left angle values by 2 bits, i.e. multiply by 4 (i.e. scale to 14bit)
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 ([Description EEPROM Address 0x00A0 ANG_BASE](#))

8.2 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[LSB₁₆]) according to the equation below:.

$$\text{LUT}[\text{LSB}_{16}] = 2 \cdot \text{POSITION}[\text{LSB}_{16}] - (16 \cdot \text{readout}[\text{LSB}_{12}])$$

POSITION[LSB₁₆]: 16bit reference angle

readout[LSB₁₂]: 12bit angle value at reference position

3. If difference is negative or $\geq 2^{16}$: 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 ([Table 10](#))
5. The 8-bit CRC calculated over the written LUT values has to be written into the address 0x00F0, bits [7:0] (see [Description EEPROM Address 0x00F0 CRC_LUT](#))

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.

Initial Calibration

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.

An example how to use the look-up table can be found in [Chapter 9.3](#).

8.3 Customer ID

The chip provides a storage of 112 bit for customer data. This data can be written to the EEPROM page 0x00F (see [Table 10](#)). The first 32 bits (address 0x00F2 and 0x00F4) are also transmitted during operation via the short serial message feature of the SENT and SPC protocol in case it is enabled. Please note that the customer data are not CRC protected.

Examples

9 Examples

The following chapter provides some practical examples.

9.1 Angle Read Out Value

Angle value read-out with SICI, register 0x02

Please refer to [Chapter 6.2.4](#) on how to read registers with SICI.

In the register 0x02, the angle value is stored as a 15-bit value in bit [14:0]. 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:

$$Value = -b_{MSB} * 2^{N-1} + \sum_{i=0}^{N-2} b_i * 2^i \quad (9.1)$$

Example:

For a register value 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:

$$Angle[^\circ] = \frac{360^\circ}{2^{15}} ANG_VAL[digits] = \frac{360^\circ}{32768} * (-12909) = -141.82^\circ \quad (9.2)$$

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 $360^\circ/2^{15} = 360^\circ/32768$. A register value of $(100\ 1101\ 1001\ 0011)_B$ equals to 19859, thus giving an angle of:

$$Angle[^\circ] = \frac{360^\circ}{32768} \cdot readout[LSB_{15}] = \frac{19859 \cdot 360^\circ}{32768} = 218.18^\circ \quad (9.3)$$

This results is the same angle as $218.18^\circ = -141.82^\circ$

Angle value read-out with SENT/SPC

With the SENT/SPC interface, the angle value comes as a 12-bit value (0 ... 4095).

1 LSB equals an angle of $360^\circ * 1/2^{12} = 0.0879^\circ$

A read-out of "0" corresponds to an angle of 0° , an read-out of "4095" equals an angle of 359.9121° .

The angle is calculated as follows:

$$(9.4)$$

Examples

$$\text{Angle}[^{\circ}] = \frac{360^{\circ}}{4096} \cdot \text{readout}[\text{LSB}_{12}]$$

SENT with error code 4091 in data range

In case the sensor is configured to SENT interface with the option of “4091” as error code in the data range, the calculation has to be modified as in this case not the complete 12 bit are available for sensor output. The angle values have to be scaled differently to 12 bit in this case. The data range in this case is 1... 4088 LSB. The notation used for this case is [LSB'12] in contrast to [LSB12], which means the full 12-bit resolution (0... 4095)

In this case, 1 LSB corresponds to $360^{\circ}/4088 = 0.0881^{\circ}$.

A read-out of “1” corresponds to an angle of 0° , an read-out of “4088” equals an angle of 359.9119°

The angle is calculated as follows:

(9.5)

$$\text{Angle}[^{\circ}] = \frac{360^{\circ}}{4088} \cdot (\text{readout}[\text{LSB}'_{12}] - 1)$$

9.2 Angle Base Calculation

The angle base is a 14-bit value which is stored in EEPROM address 0x00A0. This value defines the “zero-angle” and all output values are in reference to this value. The angle base is stored in bit [15:2], bit [1] is used to define the rotation direction of the output angle. Bit[0] of this register is not used.

To determine the zero-angle, the application is positioned to the desired angle and the angle value is read. This can be done either by reading the angle via the 12-bit protocol (SENT / SPC / PWM) or by directly reading the angle value register 0x02 via SICI and obtaining a 15-bit angle value. To avoid any quantization error, the direct read of the 15-bit the angle value is the preferred method.

For programming the 16-bit EEPROM register 0x00A4, the 14-bit angle base together with the 1-bit rotation direction information has to be combined to a 16-bit word which then can be written to EEPROM address 0x00A4. The 14-bit angle base must be in bit [15:2], the rotation information in bit [1], bit [0] is unused and has to be “0”. This procedure for construction the 16-bit word is described in the following.

First, the angle value has to be scaled to 14 bit (multiplied by 4 in case the 12-bit value is used) and shifted two bits to the left (multiplied by 4).

This value has bit [1] set to “0” (counter-clockwise, ccw). In case the rotation direction should be set clockwise (cw), bit [1] has to be modified and set to “1”. This is performed by adding “2” to the calculated value.

Example:

12-bit angle value is 512 LSB (= 45°) This should be the new angle base.

Table 16 Example for Angle Base Calculation

Value decimal	Value HEX	Description
512	0x0200	12-bit angle value from sensor, should be new angle base
2048	0x0800	Angle value is scaled to 14 bit (multiplied by 4)
8192	0x2000	Angle value is shifted two bits to the left (multiplied by 4)
8192	0x2000	Value to be burned to EEPROM address 0x00A4 in case rotation direction is ccw

Examples

Table 16 Example for Angle Base Calculation (cont'd)

Value decimal	Value HEX	Description
8194	0x2002	Add 2 for setting bit [1] (rotation direction clockwise)
8194	0x2002	Value to be burned to EEPROM address 0x00A4 in case rotation direction is cw

In case the sensor is configured to SENT interface with the option of “4091” as error code in the data range, the calculation has to be modified as in this case not the complete 12 bit are available for sensor output. The angle values have to be scaled differently to 16 bit in this case. The data range in this case is 1... 4088 LSB. The notation used for this case is [LSB'12] in contrast to [LSB12], which means the full 12-bit resolution (0... 4095)

In this case, the calculation of the angle base is different:

Determine the angle value according to [Equation \(9.5\)](#) and scale it to 14 bit by multiplying with $2^{14}/360$. Round this value and shift it by 2 bits (multiply by 4).

Example:

Angle value is 436 LSB. This should be the new angle base.

Table 17 Example for Angle Base Calculation with “4091” Error Code enabled

Value decimal	Value HEX	Description
436	0x01B4	Angle value from sensor, should be new angle base
38.307°	--	Angle value in [°] according to Equation (9.5)
1743	0x06CF	Angle value is scaled to 14 bit and rounded (multiplied by $2^{14}/360$ and rounded)
6972	0x1B3C	Value is shifted by two bits to the left (multiplied by 4)
6972	0x1B3C	Value to be burned to 0x00A4 in case rotation direction is ccw
6974	0x1B3E	Add 2 for setting bit [1] (rotation direction clockwise)
6974	0x1B3E	Value to be burned to 0x00A4 in case rotation direction is cw

So the total formula for constructing the 16-bit word containing angle base and rotation direction (rotation direction ccw) is in this case:

(9.6)

$$\text{Anglebase}[\text{LSB}_{16}] = 4 \cdot \text{round}\left(\frac{2^{14}}{4088} \cdot (\text{readout}[\text{LSB}'_{12}] - 1)\right)$$

For changing the rotation direction to clockwise (cw), add 2 to Anglebase[LSB16].

9.3 Look-up Table Calculation

The look-up table consists of 32 values, each 16 bit, which are stored in the sensor EEPROM and used for output linearization. In this way, non-linearities which come for example from external magnetic circuit assembly tolerances, can be compensated. The 32 values for the look-up table have to be generated with an external reference encoder which allows to precisely position the sensor to defined angular positions. At this defined position, the sensor output signal is read and the correction value for the look-up table is calculated. This value has to be stored at the corresponding address (see [Table 10](#)).

Examples

The formula for calculating the look-up (LUT) correction values is as follows:

(9.7)

$$\text{LUT}[\text{LSB}_{16}] = 2 \cdot \text{POSITION}[\text{LSB}_{16}] - (16 \cdot \text{readout}[\text{LSB}_{12}])$$

With:

- $\text{LUT}[\text{LSB}_{16}]$: Calculated 16-bit value which has to be stored in the corresponding address of the LUT
- $\text{POSITION}[\text{LSB}_{16}]$: angular position ($0^\circ, 11.25^\circ, 22.5^\circ, \dots, 348.75^\circ$) as 16-bit value
- $\text{readout}[\text{LSB}_{12}]$: Sensor readout at this position as a 12-bit value

In case the calculated value is negative or $\geq 2^{16}$ (= 65536), add/subtract 2^{16} so that the result is in the range of 0, ... 65535.

The reference positions for the calculation of the look-up values are $0^\circ, 11.25^\circ, 22.5^\circ, \dots, 348.75^\circ$.

In **Table 18** some examples are given for calculation of the look-up table (LUT) correction values.

Table 18 Examples for Look-up Calculation

Position [°]	Position [LSB16]	Sensor out [LSB12]	Sensor out [°]	Sensor out [LSB16]	LUT value [LSB16]
0	0	4	0.352	64	65472
0	0	4093	359.736	65488	48
11.25	2048	130	11.426	2080	2016
22.50	4096	250	21.973	4000	4192
348.75	63488	3960	348.047	63360	63616

In case the sensor is configured to SENT interface with the option of “4091” as error code in the data range, the calculation has to be modified as in this case not the complete 12 bit are available for sensor output. The angle values have to be scaled differently to 16 bit in this case. The data range in this case is 1... 4088 LSB. The notation used is [LSB'12] in contrast to [LSB12], which means the full 12-bit resolution (0... 4095).

(9.8)

$$\text{LUT}[\text{LSB}_{16}] = 2 \cdot \text{POSITION}[\text{LSB}_{16}] - \left(\frac{2^{16}}{4088} \cdot (\text{readout}[\text{LSB}'_{12}] - 1) \right)$$

Table 19 Examples for Look-up Calculation in case SENT option “4091” enabled

Position [°]	Position [LSB16]	Sensor out [LSB'12]	Sensor out [°]	Sensor out [LSB16]	LUT value [LSB16]
0	0	4	0.264	48	65488
0	0	4088	359.912	65520	16
11.25	2048	130	11.360	2068	2028
22.50	4096	250	21.927	3992	4200
348.75	63488	3960	348.64	63468	63508

Examples

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.

9.4 Temperature calculation from Short Serial Message

The temperature information which is transmitted with the short serial message is an 8-bit value (0... 255). It has to be considered as a two-complement ranging from T[LSB] = -128 LSB ... +127 LSB.

To obtain the temperature value in °C the following calculation has to be performed:

(9.9)

$$T[^{\circ}\text{C}] = \frac{2 \cdot T[\text{LSB}] + 34.54}{1.3815}$$

Table 20 Examples for temperature calculation

Read-out 8-bit value [LSB]	T [LSB] (two-complement)	temperature [°C]
251	-5	17.76
252	-4	19.21
254	-2	22.11
0	0	25.00
1	1	26.45
127	127	208.86
128	-128	-160.30
255	-1	23.55

9.5 Reading registers with SICI

The following example in [Table 21](#) shows the commands which have to be executed to read the angle value at register 0x02.

Table 21 Reading of angle value (AVAL register, 0x02)

Type	Command	Data	Comment
write	0x06C1	0x4000	Write command to address 0x6C sensor will stay with SICI interface after power-on time and not switch to preconfigured interface SENT/SPC/PWM; this first command has to be sent within the start-up time of the sensor.
wait until power-on of sensor is finished			
read	0x8021	AVAL	Reads angle value from register 0x02. This value is updated with t_{update} , which is approx. 25 µs

Examples

Table 21 Reading of angle value (AVAL register, 0x02) (cont'd)

Type	Command	Data	Comment
sensor stays in this mode with SICI interface enabled. To return to normal interface, sensor reset is necessary (can be done by power-down or sending the reset command 0x0001 to address 0x01:			
write	0x4011	0x0001	Reset command, sensor restarts with default interface

9.6 Reading EEPROM with SICI

The following example in [Table 22](#) shows the commands which have to be executed to read the EEPROM page 0x006. In this example, it is assumed, that the sensor is connected in a bus mode configuration and has the ID = 1. First the EEPROM page 0x006 is mapped to registers 0x60 to 0x67. Then these registers are read.

Further details for bus mode programming can be found in [Chapter 6.3](#).

Table 22 Reading of EEPROM page 0x006)

Type	Command	Data	Comment
write	0x0771	0x4711	Write command to address 0x77 unlocks registers
write	0x06C1	0x4000	Write command to address 0x6C with DATA 0x4000 sensor will stay with SICI interface after power-on time and not switch to preconfigured interface SENT/SPC/PWM; The first two commands have to be sent within the start-up time of the sensor.

wait until power-on of sensor is finished

write	0x47A1	0x05A9	Write ID-WORD for ID=1 to address 0x7A; only sensor with ID=1 is addressed; only necessary in case of programming in a bus mode configuration
write	0x06C1	0xC000	Write command to address 0x6C with DATA 0xC000 internal DSP is stopped
write	0x4681	0x0006	Maps EEPROM page 0x006 to registers 0x60 ... 0x67 (WRITE command to address 0x68)
read	0xC601	DATA	Reads DATA from register 0x60, corresponds to content of EEPROM address 0x0060
read	0x8611	DATA	Reads DATA from register 0x61, corresponds to content of EEPROM address 0x0062
read	0x8621	DATA	Reads DATA from register 0x62, corresponds to content of EEPROM address 0x0064
read	0xC631	DATA	Reads DATA from register 0x63, corresponds to content of EEPROM address 0x0066
read	0x8641	DATA	Reads DATA from register 0x64, corresponds to content of EEPROM address 0x0068
read	0xC651	DATA	Reads DATA from register 0x65, corresponds to content of EEPROM address 0x006A
read	0xC661	DATA	Reads DATA from register 0x66, corresponds to content of EEPROM address 0x006C

Examples

Table 22 Reading of EEPROM page 0x006) (cont'd)

Type	Command	Data	Comment
read	0x8671	DATA	Ready DATA from register 0x67, corresponds to content of EEPROM address 0x006E
Sensor stays in this mode with SICI interface enabled. To return to normal interface, sensor reset is necessary (can be done by power-down or sending the reset command 0x0001 to address 0x01:			
write	0x4011	0x0001	Reset command, sensor restarts with default interface

9.7 Writing EEPROM with SICI

The following example in [Table 23](#) shows the commands which have to be executed to write the EEPROM page 0x006. In this example, it is assumed, that the sensor is connected in a bus mode configuration and has the ID = 1. First the EEPROM page 0x006 is mapped to registers 0x60 to 0x67. Then these registers are read. After that the content of the registers 0x60 and 0x67 can be changed. Care has to be taken that reserved bits and bits which shall not be changed are written back without modification. Following that the EEPROM is burned. Further details for bus mode programming can be found in [Chapter 6.3](#).

Table 23 Writing of EEPROM page 0x006)

Type	Command	Data	Comment
write	0x0771	0x4711	Write command to address 0x77 unlocks registers
write	0x06C1	0x4000	Write command to address 0x6C with DATA 0x4000 sensor will stay with SICI interface after power-on time and not switch to preconfigured interface SENT/SPC/PWM; The first two commands have to be sent within the start-up time of the sensor.
wait until power-on of sensor is finished			
write	0x47A1	0x05A9	Write ID-WORD for ID=1 to address 0x7A; only sensor with ID=1 is addressed; only necessary in case of programming in a bus mode configuration
write	0x06C1	0xC000	Write command to address 0x6C with DATA 0xC000 internal DSP is stopped
write	0x4681	0x0006	Maps EEPROM page 0x006 to registers 0x60 ... 0x67 (WRITE command to address 0x68)
read	0xC601	DATA	Reads DATA from register 0x60, corresponds to content of EEPROM address 0x0060
	...	...	...
read	0x8671	DATA	Ready DATA from register 0x67, corresponds to content of EEPROM address 0x006E
write	0x0601	newDATA	Writes newDATA to register 0x60, corresponds to intended new content of EEPROM address 0x0060
write	...	...	...

Examples

Table 23 Writing of EEPROM page 0x006) (cont'd)

Type	Command	Data	Comment
write	0x4671	newDATA	Writes newDATA to register 0x67, corresponds to intended new content of EEPROM address 0x006E
write	0x4681	0x03FF	Prepare EEPROM programing
write	0x4671	0x0009	Program EEPROM with automatic verification
Sensor stays in this mode with SICI interface enabled. To return to normal interface, sensor reset is necessary (can be done by power-down or sending the reset command 0x0001 to address 0x01:			
write	0x4011	0x0001	Reset command, sensor restarts with default interface

10 Revision History

Revision	Date	Changes
1.0	2019-03-12	Initial release

Trademarks of Infineon Technologies AG

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

Edition 2019-03-12

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2019 Infineon Technologies AG.

All Rights Reserved.

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

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. 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) with respect to any and all information given in this application note.

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.