Application Note, V 1.1, August 2008

TLE4998S/P

User Programming Guide

Sensors

Never stop thinking.

Edition 2008-08

Published by Infineon Technologies AG,
Am Campeon 1-12,
85579 Neubiberg, Germany

Attention please!

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Information

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

Warnings

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

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision history

Date/Version: **2008-08** V 1.1

Previous Version: V 1.0

Page	Subjects (major changes since last revision)
Page 15	Threshold voltage for ‘0’ adjusted, footnote adapted in **Table 8**
Page 19	“Margin zero on” explained in test register
Page 21	Information on specific EEPROM bits, e.g. sensor lock bits
Page 22	Added missing EEPROM write block in **Figure 17**
Page 24	Explanation of EEPROM threshold measurement

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

sensors@infineon.com

TLE4998S/P-Programming Guide

1 Overview

1.1 General Information

- This document is valid for the TLE4998S and TLE4998P products and derivatives
- It is intended as add-on to the currently available TLE4998 datasheets
- It gives an overview of the internal signal processing capabilities
- It contains basic information about accessing the device using the digital interface
- It describes how to access internal registers and parameters stored in the EEPROM
- Furthermore it shows how to apply the programming voltage for the EEPROM and how to verify the programming

1.2 Block Diagram

Figure 1 shows a simplified block diagram of the TLE4998.

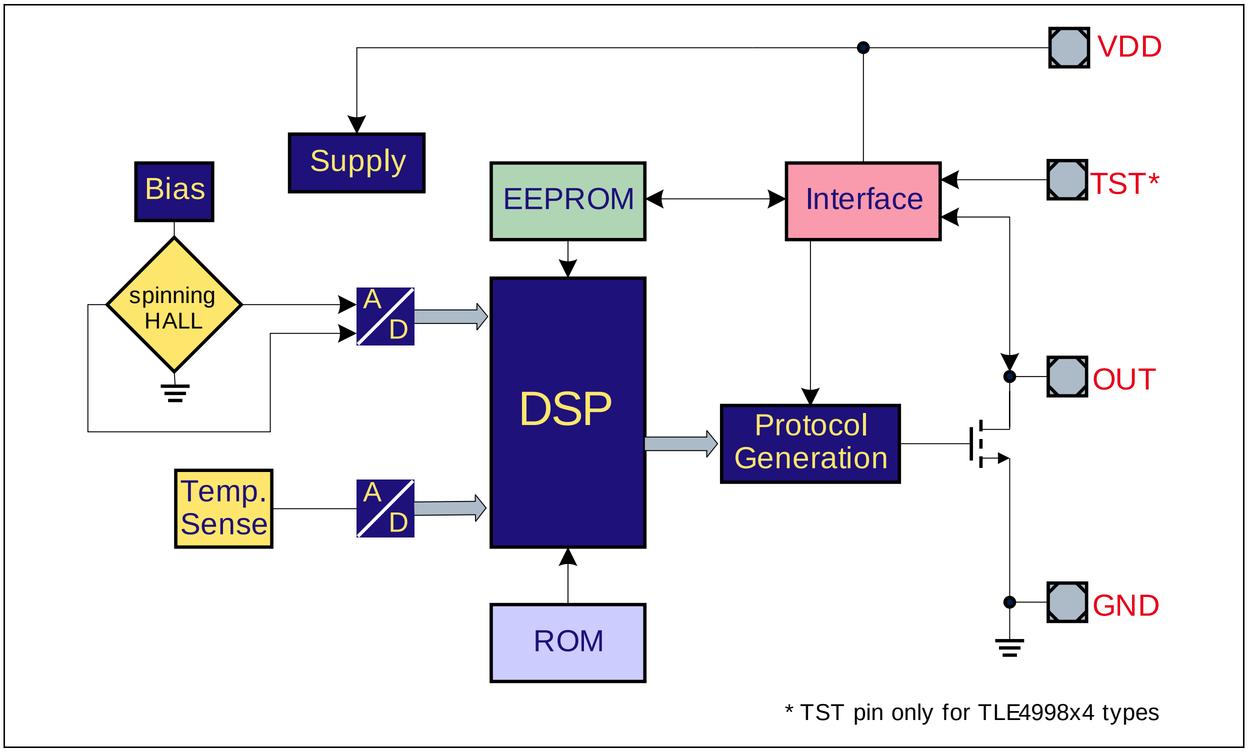


Figure 1 Block Diagram

The device can be accessed using a two-wire synchronous interface. The device supply pin (V_{dd} pin) acts as clock line and the output pin (OUT pin) is used as a bidirectional serial data line and to apply the required programming voltage.

This method allows the connection of several devices to a single power supply line while accessing the devices either separately or in parallel. Using a parallel access, multiple devices can be accessed simultaneously, consuming less time than for a serial programming. This is especially important for time-consuming operations like programming the EEPROMs in multi-device setups.

1.3 Pin Configuration

Figure 2 shows the location of the four pins of the PG-SSO-4-1 package and **Table 1** gives the corresponding pin definition and function. The same information can be found for the PG-SSO-3-10 package in **Figure 3** and **Table 2**, respectively.

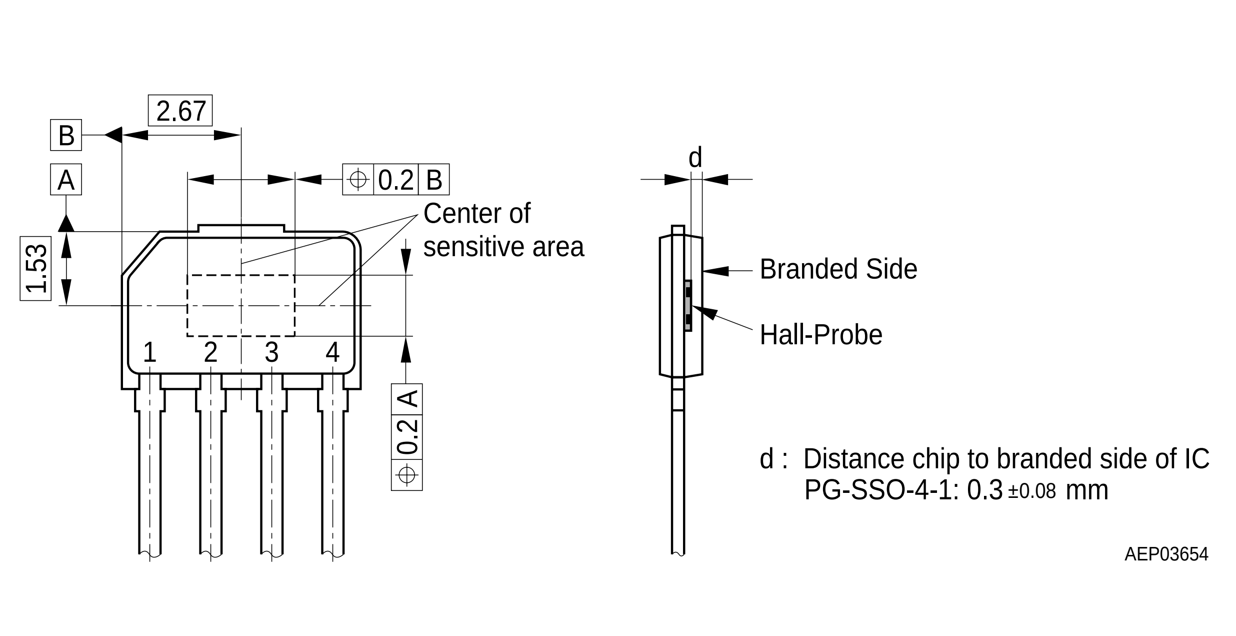


Figure 2 Pin Configuration for the PG-SSO-4-1 package

Table 1 Pin Definitions and Functions for the PG-SSO-4-1 package

Pin No.	Symbol	Function
1	*TST*	Test pin (connection to GND is recommended)
2	*VDD*	Supply voltage / programming interface (clock)
3	*GND*	Ground
4	*OUT*	Output/ programming interface (I/O data, V_{prog})

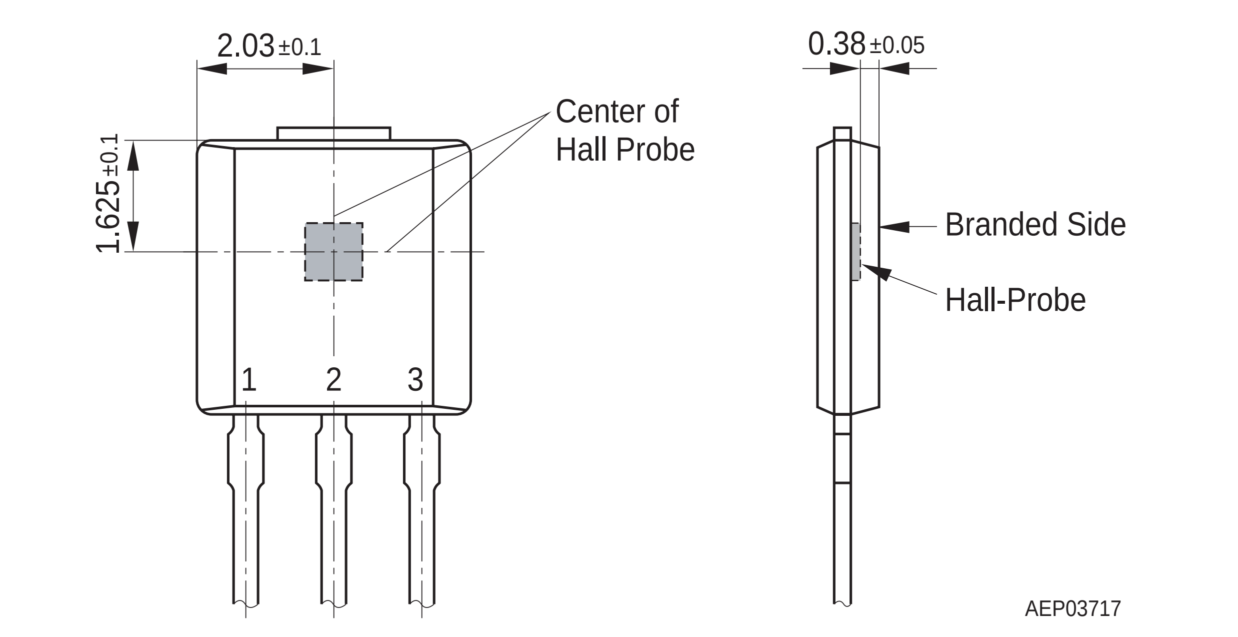


Figure 3 Pin configuration for the PG-SSO-3-10 package

Table 2 Pin Definitions and Functions for the PG-SSO-3-10 package

Pin No.	Symbol	Function
1	*VDD*	Supply voltage / programming interface (clock)
2	*GND*	Ground
3	*OUT*	Output/ programming interface (I/O data, V_{prog})

More information regarding location of branding, Hall probe etc. can be found in the corresponding datasheet.

1.4 Signal Flow

Figure 4 shows the signal flow diagram including important internal data values.

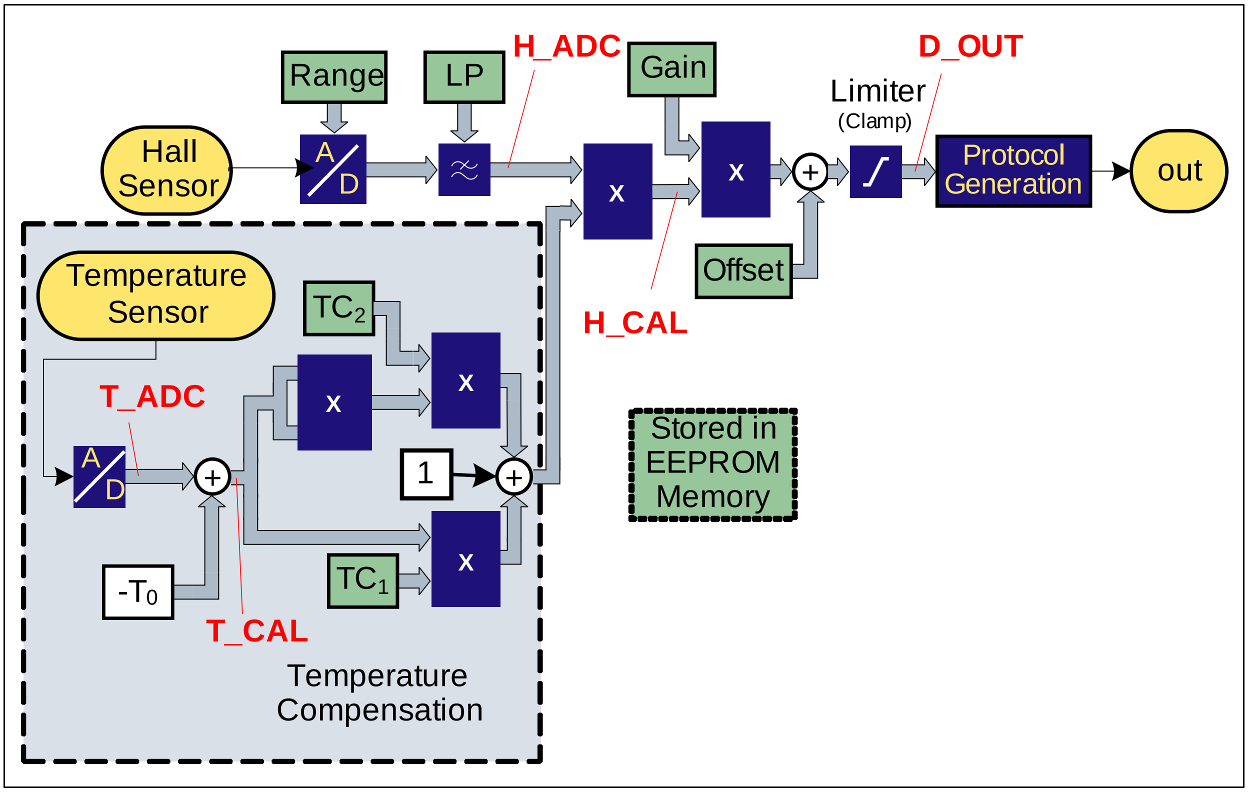


Figure 4 Block Diagram

Table 3 Internal data values

Address.	Symbol	Function
0x00	*D_OUT*	Data out value (16 bit unsigned, with clamping)[1]
0x05	*H_CAL*	Calibrated Hall value[1][2]
0x06	*T_CAL*	Calibrated temperature value, incl. reference-temp. T_0[1][2]
0x0A	*H_ADC*	Uncalibrated Hall ADC value [1][3]
0x0B	*T_ADC*	Uncalibrated temperature ADC value [1][3]

[1] requires activated interface - access possible only with unlocked devices

[2] requires special debug mode

[3] please note that this value does not include any compensation - these are just the internal “raw” ADC values

2 Interface Access Details - Part I

2.1 Functional Interface Description

All internal data is organized in a memory-like setup. Each data value or EEPROM parameter is located at a specific address. The data width is always 16 bit. The interface uses specific frames for information exchange, which can have one of the two following functions:

- *Command frames* contain a specific task (e.g. read/write data, select EEPROM programming etc.) and a corresponding address
- *Data frames* contain a 16 bit data value sent to or received from the device - these frames can only follow a proper command frame for reading or writing data

A valid frame has these properties:

- A frame consists always of 21 bits
- A bit is shifted in or out via the output line with a rising clock edge on the supply line.
- A frame always starts and ends with a '1' (frame bits)
- The LSB of a transmitted frame is shifted in first
- The LSB of a resulting frame is shifted out first
- The whole frame sent to the device, including frame bits, is protected with an even positional and an odd positional parity bit

The first frame sent must always be a valid command to activate the interface mode and has to be sent within 19ms after power up. As an additional protection, the device does not deactivate its output stage during this transmission (using 21 clock pulses) as shown in **Figure 5**. This means that the external interface driver needs to overrule the open drain output stage of the sensor during this initial transmission.

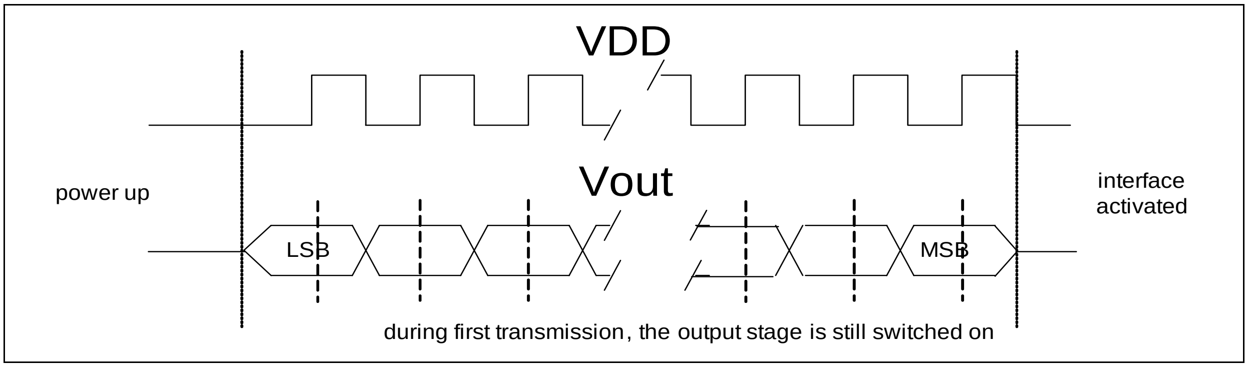

VDD
Vout
power up
interface activated
LSB
MSB
during first transmission, the output stage is still switched on

Figure 5 First frame transmission to the device

Note: Overruling V_{out} requires a strong driver, since the line must be driven to both low levels close to GND for any "0"-bit and close to VDD for any "1"-bit in order to assure a proper detected by the sensor.

Later, to avoid additional power consumption in the output stage of the device, the internal driver is deactivated as soon as the first clock pulse of a frame is detected, and put on again after completion of the transmission. This is illustrated in **Figure 6**.

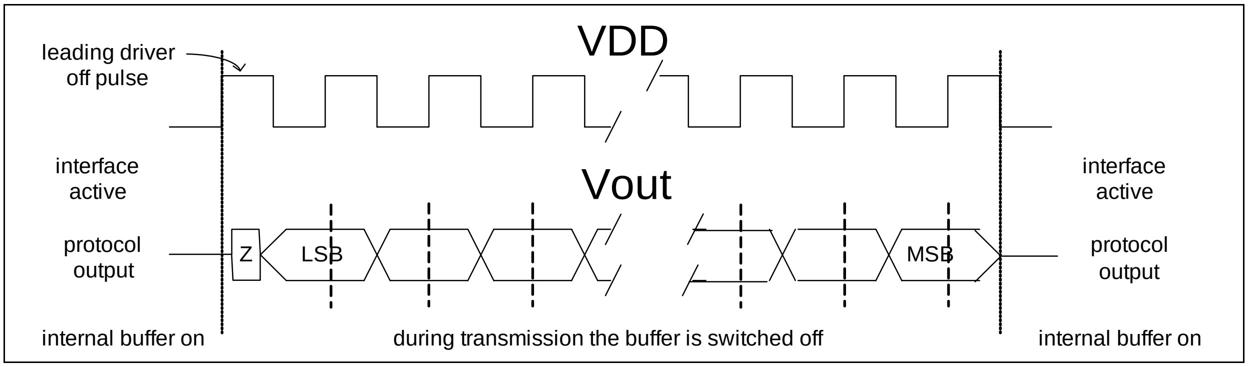


Figure 6 Further frame transmission to the device (write access)

In case of a wrong command or data frame, the interface is immediately locked and the device falls back to its normal application mode. As long as the device is in the interface mode, selected test modes stay activated, too. Special transmission modes based on these frames, used for programming of the EEPROM, will be explained in the EEPROM programming section.

The read access to the device is triggered by clock pulses on the supply line as shown in figure **Figure 7**. The exact timing of both read and write accesses are outlined in **Section 3.2**.

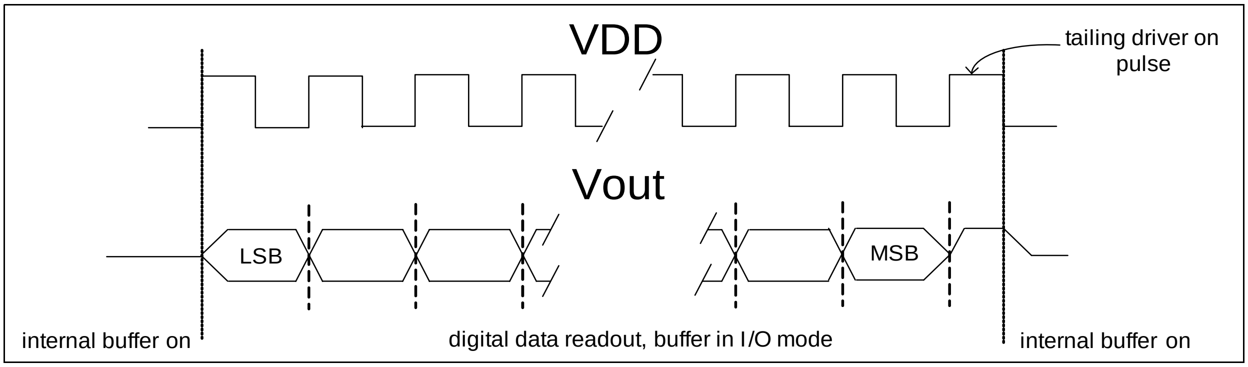


Figure 7 Frame reception from the device (read access)

2.2 Command Frame Description

As already described, the data transmission is performed by command frames. These command frames are supported by following data frames, if required. A general command frame is shown in **Figure 8**. Available commands are given in **Table 4**. Available addresses are summarized in **Section 4.1** based on address maps. The parity bits PE (bit 17) and PO (bit 18) need to be set in a way that the following conditions are met (bit 0 is the LSB, bit 20 is the MSB):

- bit0 XOR bit2 XOR bit4 XOR …. XOR bit20 = 0
- bit1 XOR bit3 XOR bit5 XOR …. XOR bit19 = 0

Please refer to **Chapter 2.4** for a source code example of a parity generator.

MSB										LSB
1	1	P O	P E	0	0	ADDR (6bit)	1	0	CMD (6bit)	1

Figure 8 Command Frame

Table 4 List of available device commands

CMD No.	Bits[1]	Function
0x0	*“000000”*	Leave interface mode [2]
0x1	*“000001”*	Single data readout from given address without increment [3]
0x2	*“000010”*	Continuous data readout from given address without increment (power-on cycle needed to leave this state)
0x3	*“000011”*	Continuous readout from given address with increment (readout finished when address *“xxx111”* is reached – block read, can not be canceled before!) [4]
0x9	*“001001”*	Single write data to given address without increment [3]
0xA	*“001010”*	Continuous write data to given address without increment (finished by sending an arbitrary sync. command frame to cancel)
0xB	*“001011”*	Continuous write data to given address with increment (block write, finished at address *“xxx111”* or by sending an arbitrary sync. command frame to cancel) [4]
0xC	*“001100”*	Enable EEPROM write mode (programs *“1”*-bits) [2] [5]
0xD	*“001101”*	Enable EEPROM erase mode (programs *“0”*-bits) [2] [5]
0xE	*“001110”*	Enable EEPROM margin mode (program level check) [2] [6]
0xF	*“001111”*	Enable EEPROM refresh (update EEPROM registers) [2]

[1] Left is MSB, right is LSB

[2] No data frame must follow

[3] Exactly one data frame must follow.

[4] One or more data frames must follow until address reaches block boundary (*“xxx111”*).

5) A program pulse must follow after the frame (output stage is kept disabled).

6) A margin voltage level must follow before the last Vdd clock pulse falling edge (this edge is used for refreshing the EEPROM registers using the margin voltage).

2.3 Data Frame Description

A general data frame sent to the device is shown in **Figure 9**. The parity bits PE (bit 17) and PO (bit 18) need to be set (in the same way as for the command frame) that the following conditions are met (bit 0 is the LSB, bit 20 is the MSB):

- bit0 XOR bit2 XOR bit4 XOR …. XOR bit20 = 0
- bit1 XOR bit3 XOR bit5 XOR …. XOR bit19 = 0

Please refer to **Chapter 2.4** for a source code example of a parity generator. **Figure 10** shows a general data frame received from the sensor. Instead of a zero bit followed by two parity bits, the least significant 3 bits of the address used for the readout are transmitted together with the data. This allows to check the plausibility of the received data.

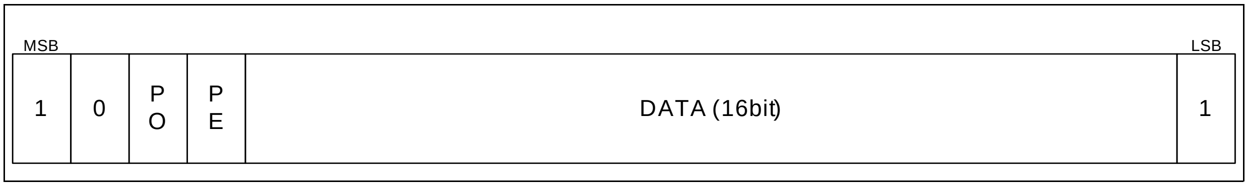


Figure 9 Data frame (write to device)

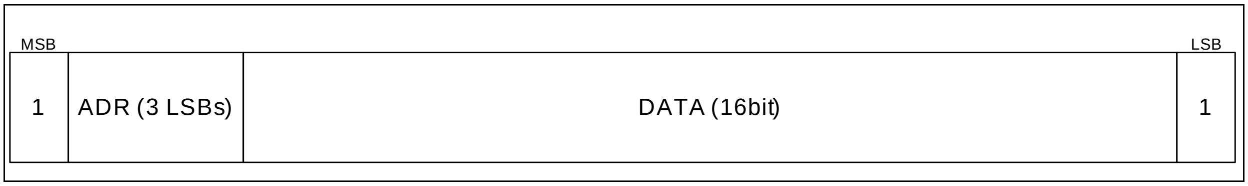


Figure 10 Data frame (read from device)

2.4 Interface Parity Calculation

An example parity generator is shown using a pseudo code. The array “framedatabits” contains the data bits to transmit including the framebits, its index corresponds to 0 ... LSB and 20 ... MSB.

This parity calculation is valid for command and data frame transmissions:

```
// count framedatabits from 0 (LSB) to 20 (MSB) - this are 21 bits
// bit 0 and 20 are always '1' (framebits)

pe = framedatabit(19);
```

```
po = 0;

for (i=1; i<17; i++)  // go through all data bits
{  // handle even/odd separately
   if ((i&1)==1) {
      if (framedatabit(i)==1) { if (pe) pe=0; else pe=1; }//toggle pe
   } else {
      if (framedatabit(i)==1) { if (po) po=0; else po=1; }//toggle po
   }
}

framedatabit(17) = pe;
framedatabit(18) = po;
```

For example, a command 0x03 using address 0x02 should be transmitted (this command triggers a block readout for addresses 0x02 to 0x07):

Table 5 Valid Frame Example

Bitcount	***20***	***19***	***18***	***17***	***16***	***15***	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Descr. of CMD-frame	*1* MSB	*1*	*P O*	*P E*	*0*	*0*	*A 5*	*A 4*	*A 3*	*A 2*	*A 1*	*A 0*	*1*	*0*	*C 5*	*C 4*	*C 3*	*C 2*	*C 1*	*C 0*	*1* LSB
Descr. of DATA-frame	*1* MSB	*0*	*P O*	*P E*	*D 15*	*D 14*	*D 13*	*D 12*	*D 11*	*D 10*	*D 9*	*D 8*	*D 7*	*D 6*	*D 5*	*D 4*	*D 3*	*D 2*	*D 1*	*D 0*	*1* LSB
Bits for PO	*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*
Bits for PE		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*		*X*	
CMD-frame	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**

3 Interface Access Details - Part II

3.1 General Operating Conditions

The following operating conditions must not be exceeded in order to ensure correct operation of the TLE4998 during programming and debugging. All parameters specified in the following sections refer to these operating conditions, unless otherwise noticed.

Table 6 Operating Range[1]

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Supply voltage	V_{DD}	4.5	5.5	V	
Supply buffer cap.	C_S	47	1000	nF	from Vdd to GND [2] [3]
Load capacitance	C_L	-	8	nF	from OUT to GND [3]
Ambient temperature	T_a	10	30	°C	at programming [4]

[1] Keeping signal levels within the limits specified in this table ensures correct setup and programming.

[2] Prevents severe supply drops causing a device reset during interface access. For high reliability use, the capacitance must be soldered to the device to avoid contact failures.

[3] Please be aware that the driving circuits of V_{dd} and V_{out} need also additionally proper driving strength for these capacitors.

[4] Interface readouts are also possible at higher and lower temperatures (altough not explicitly tested and guaranteed), but applying the programming- or margin- voltage outside this room temperature range is strictly forbidden.

3.2 Timing and Electrical Parameters for Interface Access

For accessing the interface, the supply pin and output pin must be properly accessed; the timing parameters correspond to **Figure 11**.

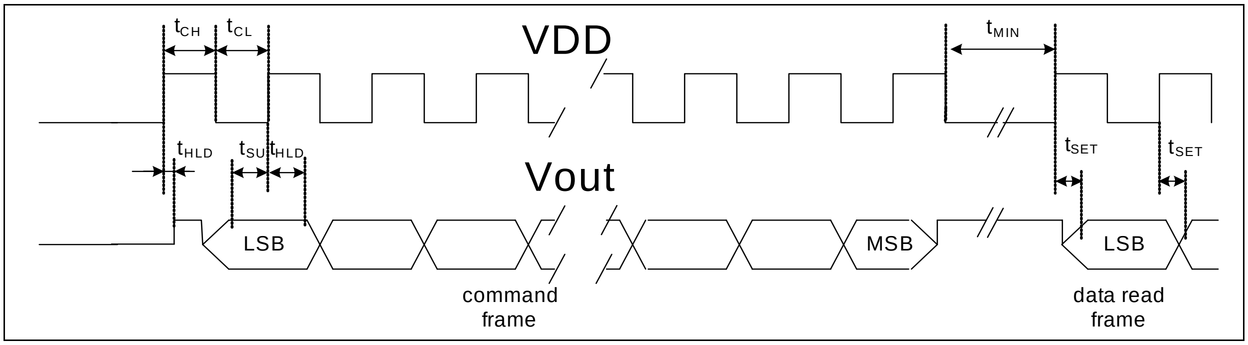


Figure 11 Frame timing

3.3 Timing and Electrical Parameters for Programming

Table 7 Electrical levels and interface timing

Parameter	Symbol	Limit Values			Unit	Notes
		min.	typ.	max.		
V_{dd} clock high level	$V_{dd,CLKHI}$	7.4	7.7	8.0	V	[1]
V_{dd} clock low level	$V_{dd,CLKLOW}$	4.8	5.0	5.2	V	this is V_{dd}[1]
OUT data out high level	$V_{O,OHIGH}$	-	-	-	V	[2]
OUT data out low level	$V_{O,OLOW}$	-	-	-	V	
OUT data in high level	$V_{O,IHIGH}$	50% V_{dd}	-	V_{dd}	V	
OUT data in low level	$V_{O,ILOW}$	-0.2	0.0	0.1	V	
OUT data input current	I_O	-5	-	5	mA	[3]
V_{dd} clock high time	t_{CH}	2.5	5.0	25	µs	exact bitrate tbd[4]
V_{dd} clock low time	t_{CL}	2.5	5.0	25	µs	
Data in setup time	t_{SU}	1.5	2.0	-	µs	to rising V_{dd}
Data in hold time	t_{HLD}	2.3	3.0	-	µs	after rising V_{dd}
Data out settling time	t_{SET}	-	1.0	1.7	µs	after rising V_{dd}
Time between frames	t_{MIN}	10.0	-	-	µs	[5]

[1] Prevent over-/ underswing during Vdd switching to avoid unexpected sensor behavior (e.g. undervoltage sensor reset).

[2] Corresponds to the open drain specification in the data sheet.

[3] Capability of external driver, especially during initial interface access (to overwrite device output).

[4] Excact bitrates depend also on several further conditions, like length of cables (inductors) and the electrical behavior of the used programming device/setup. Furthermore the desired customer margins for the timing and voltage levels may limit the bitrate even more. In case of problems try the typical recommended bitrate first and optimize the timing based on measurements using the given system. Also temperatures outside the allowed temperature range for the EEPROM programming are not considered in the above recommendation.

[5] In interface mode, EMC influences or Vdd drops during and between frames may cause internally to stop the interface mode due to safety reasons; a power cycle is needed to allow interface access again.

Additionally, for programming, **Figure 12** shows a general Vprog pulse timing:

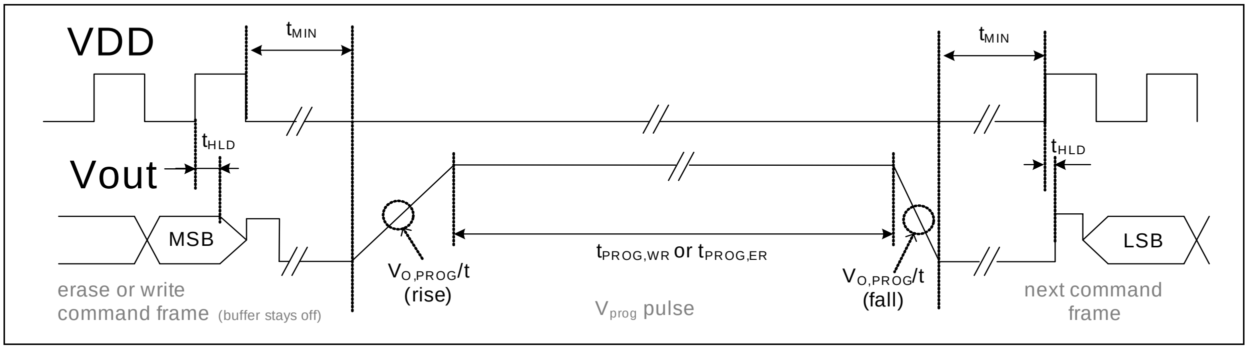


Figure 12 Program pulse timing

A margin readout needs a special behavior (**Figure 13**) at the end of a command frame:

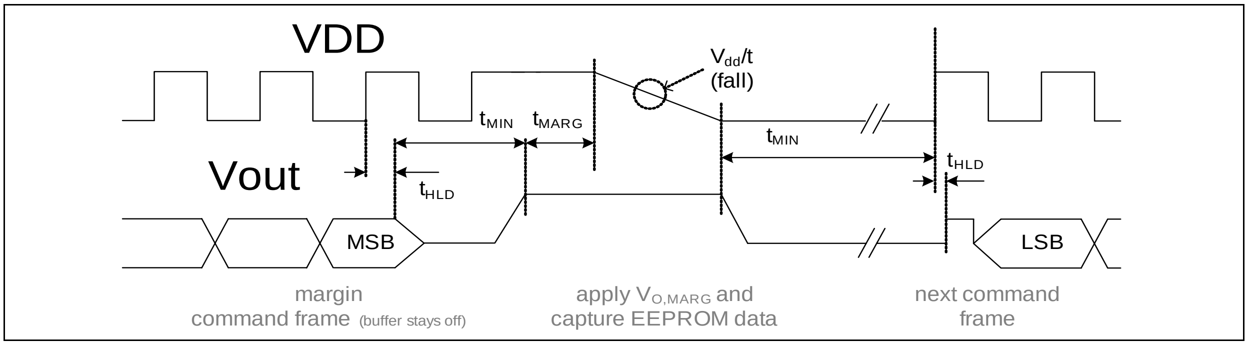


Figure 13 Margin setup timing

Table 8 Electrical levels and interface timing

Parameter	Symbol	Limit Values			Unit	Notes
		min.	typ.	max.		
OUT input current	I_O	0		5	mA	[1]
OUT margin level	$V_{O,MARG}$	-0.1		7	V	[2]
Threshold margin level	V_{TH}	2.23		4.5 0.4	V V	check ’1’[3] check ’0’[4]
Margin setup time	t_{MARG}	200			µs	
V_{dd} slope for margin	V_{dd}/t	100	200	300	mV/µs	falling edge[5]
OUT program level	$V_{O,PROG}$	19.2	19.3	19.4	V	low tolerance![2]
OUT prog. slope (rise)	$V_{O,PROG}/t$	[6]		2	V/µs	[7]
OUT prog. slope (fall)	$V_{O,PROG}/t$	-10		[7]	V/µs	[8]
OUT write time	$t_{PROG,WR}$	9.9	10.0	10.1	ms	
OUT erase time	$t_{PROG,ER}$	79.2	80.0	80.8	ms	

[1] When $V_{o,prog}$ or $V_{o,marg}$ is applied.

[2] Proper command must be applied first to switch off internal output stage of device.

[3] Level range within which programmed EEPROM bits start to flip from ones to zeros - to be checked after programming:
- a too low value could be given by too short programming pulse or a too low programming voltage
- a too high value could be given by a too long programming pulse or a too high programming voltage
To check the programmed ‘1’ threshold levels, the “Margin zero on” bit needs to be set to ‘0’ in the test register

[4] To check the programmed ‘0’ threshold levels, the “Margin zero on” bit needs to be set to ‘1’ in the test register

[5] To improve the margin accuracy, the last falling clock edge must be slowed down for the margin command.

[6] Time to reach $V_{O,PROG}$ min. must not exceed 50µs

[7] Time to reach 1V max. must not exceed 50µs

[8] Ramp up/down needs to be assured by the programming hardware - especially faster slopes, when applying the programming voltage, may damage the EEPROM cell.

Due to this specification for the programming pulse, either a linear programming ramp or an exponential ramp (using an R/C circuit) may be applied as shown in **Figure 14**.

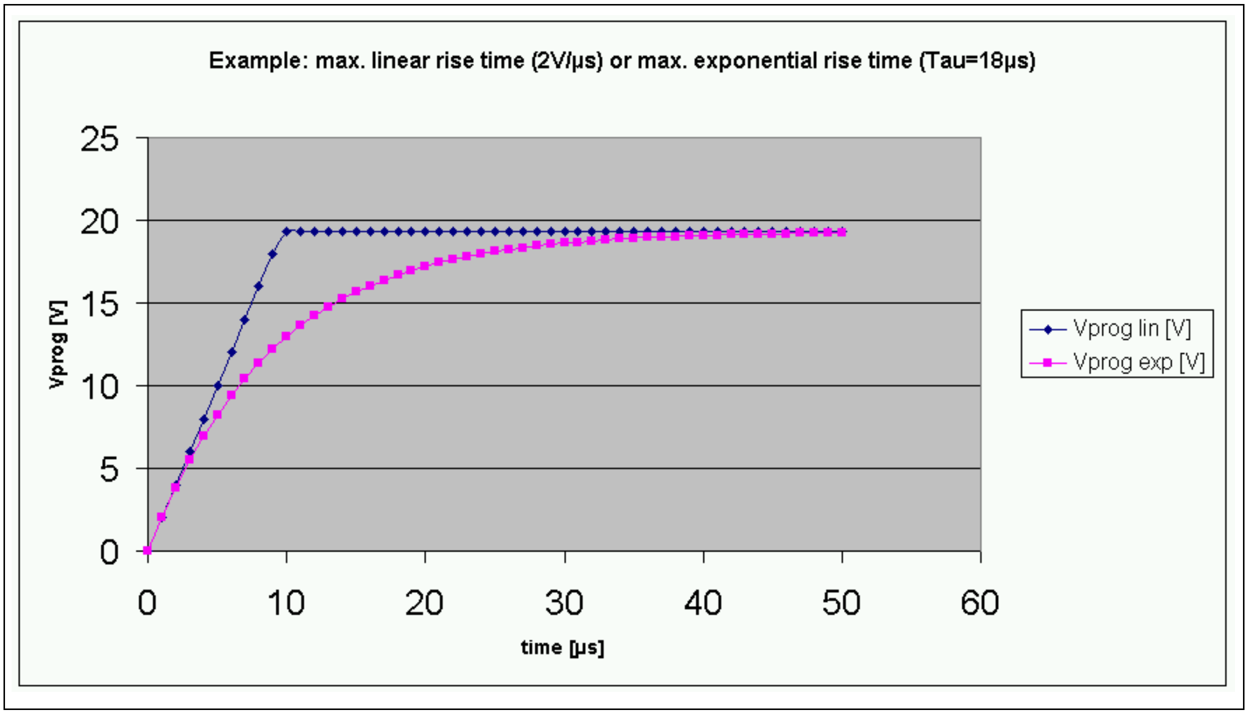


Figure 14 Example slopes for $V_{O,PROG}$

4 Interface Access Details - Part III

4.1 Complete Memory Map

Table 9 Memory map

Address.	Symbol	Function
0x00	*D_OUT*	Data out value (16 bit unsigned, with clamping)
0x05	*H_CAL*	Calibrated Hall value
0x06	*T_CAL*	Calibrated temperature value, incl. reference-temp. T_0
0x0A	*H_ADC*	Uncalibrated Hall ADC value
0x0B	*T_ADC*	Uncalibrated temperature ADC value
0x0F	*STATUS*	Chip status register
0x10...0x1A	*EEPROM*	EEPROM map [1)]
0x1B	*TEST*	Test mode register [1)]

[1)] these addresses allow write access, too

- D_OUT, H_CAL, T_CAL, H_ADC, T_ADC: These registers correspond to the signal flow diagram shown in **Chapter 1.4**
- STATUS: This register contains internal status information as well as the chip version
- EEPROM: These registers contains the EEPROM based parameter-set of the device
- TEST: This register activates several test modes necessary for accessing internals of the device

Note: To access the registers (except STATUS, H_ADC, T_ADC, D_OUT and TEST) the internal digital signal processor must be deactivated as it has priority over interface read and write commands. Please check out the TEST register content how to disable the DSP.

4.2 Register Details

4.2.1 H_ADC

This register contains a 16bit signed value. When read as unsigned value and this value is larger than 32767, it is necessary to subtract 65536 to get a signed value again:

```
0111111111111111 (unsigned dec. 32767) the (theoretical) max. pos. field
0100111000100000 (unsigned dec. 20000) the max. allowed positive field
0000000000000001 (unsigned dec.     1) is a growing positive field
0000000000000000 (unsigned dec.     0) is the zero field (without offset error)
1111111111111111 (unsigned dec. 65535) is a growing negative field (-1)
1011000111100000 (unsigned dec. 45536) the max. allowed negative field (-20000)
1000000000000000 (unsigned dec. 32768) the (theoretical) max. neg. field (-32768)
```

For valid usage of the device, the H_ADC value must be always in a range of +/- 20000 decimal, which corresponds to approximately 2/3 of the theoretical integer range. Otherwise the used magnetic flux density is too high and the ADC might be saturated.

4.2.2 T_ADC

This register contains a 15bit unsigned value. This value is not important for the usage of the IC. Just for information this value is roughly in a range of 22000 to 29000 decimal for temperatures between -50°C and 150°C.

4.2.3 H_CAL

This register contains a 16bit signed value similar to the H_ADC value and is required to calculate the output DAC value for a specific magnetic value. This value is at a given magnetic range (for a calibrated device) in the range of +/- 30000 (approx. 1.5 times of H_ADC) when the max. positive or negative field is applied.

4.2.4 T_CAL

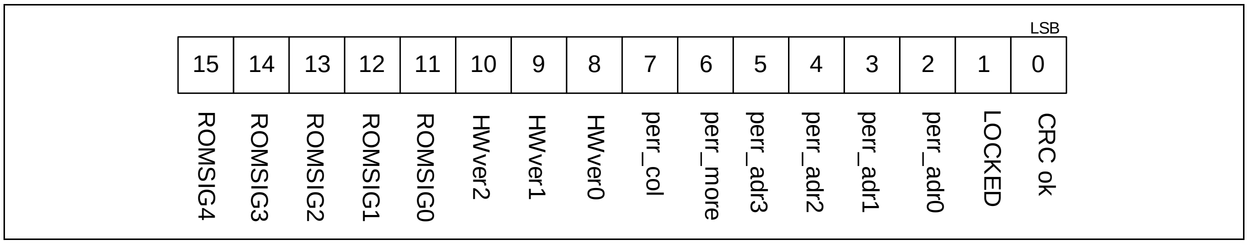
This register contains a 16 bit signed value and delivers the current junction temperature of the device. To retreive the actual temperature in °C, the register value needs to be calculated with T_J=T_CAL/16+48.

4.2.5 D_OUT

This value is the 16 bit unsigned decimal result applied to the internal protocol generation for the open drain output stage. It includes the clamping limits if programmed. The value range is from decimal 0 to 65535.

4.2.6 STATUS

The content of the status register is shown in **Figure 15**.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROMSIG4	ROMSIG3	ROMSIG2	ROMSIG1	ROMSIG0	HWver2	HWver1	HWver0	perr_col	perr_more	perr_adr3	perr_adr2	perr_adr1	perr_adr0	LOCKED	CRC ok

Figure 15 Status register

- CRC ok must be ’1’, otherwise the DSP BIST failed and the device is defective.

- LOCKED must be ’0’ as long as the lockbits are not programmed. Newly programmed lock bits effect the LOCKED bit after the next power cycle. A locked interface indicates that the chip has been locked successfully.
- perr_adr must be on address 0xF (=”1111”) otherwise it shows the first EEPROM address (=line) where the internal parity check failed.
- perr_more must be ’0’, otherwise more than one EEPROM address (=lines) has a parity error.
- perr_col must be ’0’, otherwise one or more EEPROM columns have a parity error.
- HWver contains the actual silicon revision (for the TLE4998S/P A11, this number is set to “000”).
- ROMSIG must be 0x15 (=”10101”) otherwise the DSP ROM is not valid and the device itself is defective.

To summarize, for a sensor without defects and appropriate parity programming, the status register should have the setting 0xA83D for the TLE4998S/P.

4.2.7 TEST

The content of the test register is shown in **Figure 16**.

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	DSP stop	DSP off	REF off	0	PROTOCOL off	FEC off	Margin zero on	0	0	0	0	0

Figure 16 Test register

All bits are ’0’ after reset. All bits not described or used must kept at ’0’.

- “Margin zero on” needs to be set to ‘1’ for testing the EEPROM threshold voltages of cells programmed to ‘0’. The bit has to be set to ‘0’ if the EEPROM threshold voltages of cells programmed to ‘1’ are tested
- “FEC off” switches off the error correction of the EEPROM. This bit should be set when reading the EEPROM content to ensure to retreive the real data stored in the EEPROM (address range 0x10 to 0x1A)
- “REF off” switches off the automatic (cyclic) refresh performed by the DSP to actualize the EEPROM registers from the EEPROM cells. When writing new values to the EEPROM registers this bit must be set, otherwise these values will be always overwritten by the EEPROM content
- “DSP off” switches off the signal processor immediately. This bit must be set prior to access the internal register values via the interface (H_CAL, T_CAL and EEPROM).

- “DSP stop” should be set prior (as a separate step before switching the DSP off) when reading out the calculated data H_CAL and T_CAL. This allows the DSP to finish the calculation of the current sample and all values in the RAM are consistent. Normally the cycle time of the protocol should be waited before the DSP is switched off completely (“DSP off” =1).
- “PROTOCOL off” should only be set together with “DSP off ”. It can be set if it is desired to get correct protocol output after “DSP off” and “PROTOCOL off” are cleared again. Otherwise, correct protocol output is guaranteed only after reset.

4.2.8 EEPROM

The content of the EEPROM setup registers is shown in **Table 10**. The red marked parameters set the sensor hardware, the yellow marked parameters are used by the DSP algorithms and the magenta/cyan values correspond to the parity setup for the internal forward error correction (FEC). All parameters are unsigned integer values. The white areas must not be changed.

Table 10 EEPROM registers

ADR	Description	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0x10	*Parity of each column*	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c	P_c
0x11	*IC lock (high locks), clamping high/ low value*	P_l	**LH**	**CH - register (bit 6...0)**							**CL - register (bit 6...0)**						
0x12	*gain setting*	P_l	**G - register (bit 14...0)**														
0x13	*offset setting*	P_l	**OS - register (bit 14...0)**														
0x14	*ID, PC...temp. cal. status bit, Predivider, TQ value*	P_l	**ID**		**PC**	**Prediv (bit 3...0)**				**TQ - register (bit 7...0)**							
0x15	*Bandwidth, Range, TL value, IC lock (low locks)*	P_l	**BW (2...0)**			**R (1...0)**		**TL - register (bit 8...0)**									**LL**
0x16	*precal*	P_l	**precal area - do not modify**														
0x17	*precal*	P_l	**precal area - do not modify**														
0x18	*precal*	P_l	**precal area - do not modify**														
0x19	*precal*	P_l	**precal area - do not modify**														
0x1A	*precal*	P_l	**precal area - do not modify**														

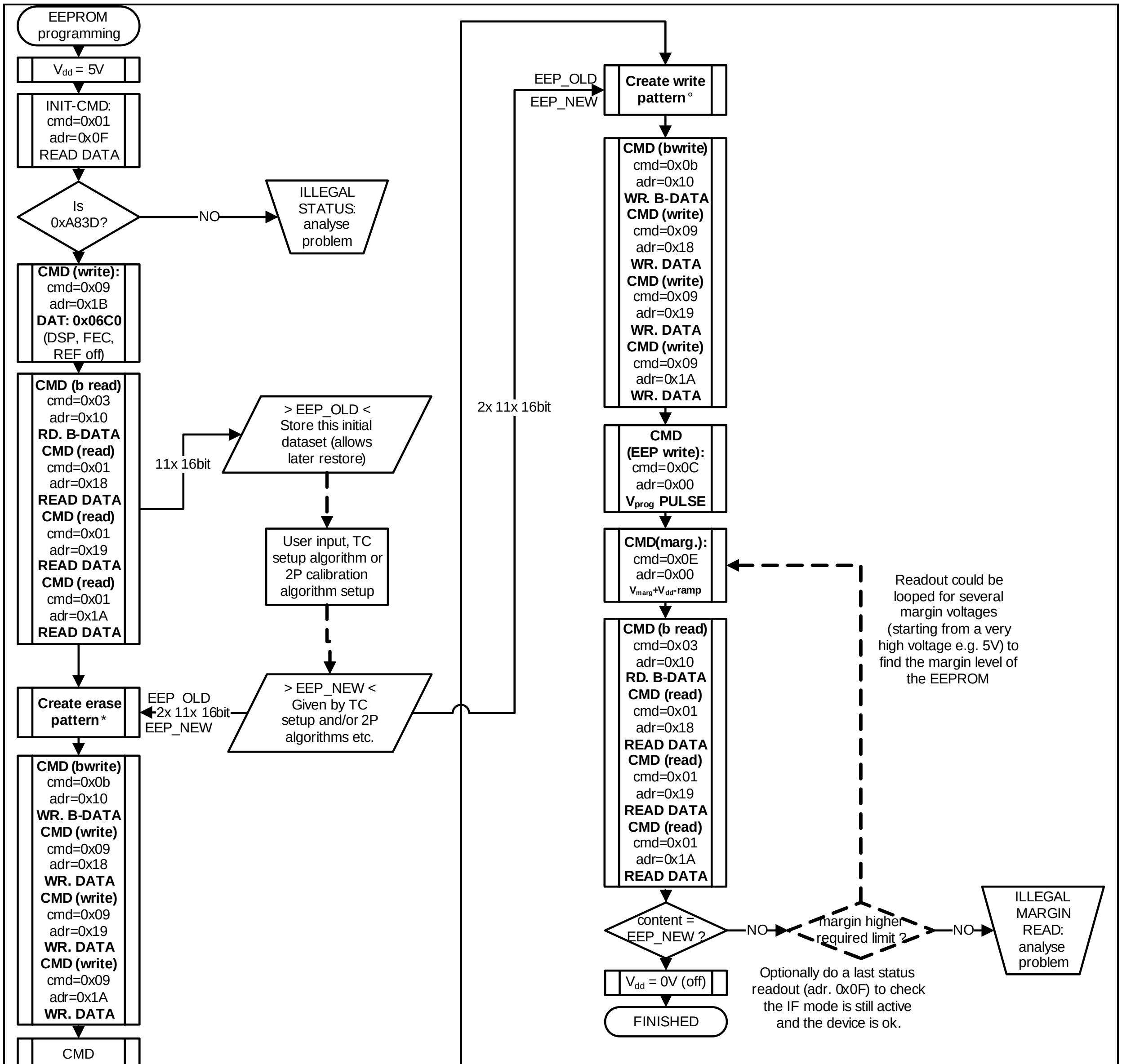
- The parity P_c of each column (including the precalibration ranges) must be even for even bit positions (bit0=LSB, bit2, bit4, ... bit14) and the parity P_c for all odd columns (bit1, bit3, ... bit13) must be odd. The parity P_c for the column at bit15 (MSB) must be even. The parity P_l of every EEPROM line (adress 0x11 ... 0x1A) needs to be calculated in a way that the sum of bits is always odd

- The two ID bits are free bits and can be used by the customer
- PC stands for precal bit. Infineon's eval software toggles this bit after the first programming. This way it is possible to see whether a device has still the precalibrated settings.
- LH and LL are lock bits (LH locked if '1', LL locked if '0'). As soon as either LH, LL or both are set to locked state, the interface mode cannot be accessed anymore. Therefore, LH and LL can be used to lock the interface and avoid that the interface is accessed and registers changed by mistake after final programming
- All other parameters (G, OS, etc.) are are defined and explained in the datasheet

Note: Don't forget to switch off the FEC during access of the EEPROM to read the data actually stored in the EEPROM cells to detect possible faults.

4.3 Basic EEPROM Access and Programming Procedure

Following steps are required to setup the EEPROM and to program new values, assuming additional external memory called EEP_NEW (new EEP values), EEP_PROG (for intermediate values) and EEP_OLD (current EEP values).


EEPROM programming
Vdd = 5V
INIT-CMD: cmd=0x01 adr=0x0F READ DATA
Is 0xA83D?
NO
ILLEGAL STATUS: analyse problem
CMD (write): cmd=0x09 adr=0x1B DAT: 0x06C0 (DSP, FEC, REF off)
CMD (b read) cmd=0x03 adr=0x10 RD. B-DATA
CMD (read) cmd=0x01 adr=0x18 READ DATA
CMD (read) cmd=0x01 adr=0x19 READ DATA
CMD (read) cmd=0x01 adr=0x1A READ DATA
11x 16bit
> EEP_OLD < Store this initial dataset (allows later restore)
User input, TC setup algorithm or 2P calibration algorithm setup
> EEP_NEW < Given by TC setup and/or 2P algorithms etc.
Create erase pattern*
EEP_OLD
2x 11x 16bit
EEP_NEW
CMD (bwrite) cmd=0x0b adr=0x10 WR. B-DATA
CMD (write) cmd=0x09 adr=0x18 WR. DATA
CMD (write) cmd=0x09 adr=0x19 WR. DATA
CMD (write) cmd=0x09 adr=0x1A WR. DATA
CMD (EEP erase): cmd=0x0D adr=0x00 Vprog PULSE
2x 11x 16bit
EEP_OLD
EEP_NEW
Create write pattern°
CMD (bwrite) cmd=0x0b adr=0x10 WR. B-DATA
CMD (write) cmd=0x09 adr=0x18 WR. DATA
CMD (write) cmd=0x09 adr=0x19 WR. DATA
CMD (write) cmd=0x09 adr=0x1A WR. DATA
CMD (EEP write): cmd=0x0C adr=0x00 Vprog PULSE
CMD(marg.): cmd=0x0E adr=0x00 Vmarg+Vdd-ramp
Readout could be looped for several margin voltages (starting from a very high voltage e.g. 5V) to find the margin level of the EEPROM
CMD (b read) cmd=0x03 adr=0x10 RD. B-DATA
CMD (read) cmd=0x01 adr=0x18 READ DATA
CMD (read) cmd=0x01 adr=0x19 READ DATA
CMD (read) cmd=0x01 adr=0x1A READ DATA
content = EEP_NEW ?
NO
margin higher required limit ?
NO
ILLEGAL MARGIN READ: analyse problem
Vdd = 0V (off)
FINISHED
Optionally do a last status readout (adr. 0x0F) to check the IF mode is still active and the device is ok.

* Erase pattern: For each line I from 0x10 to 0x1A:
EEP_PROG[i] = INVERT ((EEP_OLD[i] XOR EEP_NEW[i]) AND EEP_OLD[i])
(as precal areas must not be changed, the bits in this areas must remain ‚1')

° Write pattern: For each line I from 0x10 to 0x1A:
EEP_PROG[i] = (EEP_OLD[i] XOR EEP_NEW[i]) AND EEP_NEW[i]
(as precal areas must not be changed, the bits in this areas must remain ‚0')

Figure 17 Basic EEPROM programming flow

Description of the flowchart shown in Figure 17:

1. Switch on the device
2. Send an inital command (status line readout)
   - read the status,check that the device is valid and the EEPROM content is valid
   - if it is not correct, do not continue and check for the failure
3. Set the test register bits FECoff=1, DSPoff=1, REFoff=1 (allows EEPROM access)

4. Read out the EEPROM content to the array EEP_OLD (store also for reference purposes and for traceability of programming)
   - *Parallel task:* prepare the data necessary to program as array EEP_NEW (see application notes for calculation of TC parameters and how to do a 2-point calibration)
5. Calculate the bits to be cleared from EEP_OLD to EEP_NEW as EEP_PROG array
6. Write the EEPROM content from the EEP_PROG array to the EEPROM registers
7. Send the EEPROM erase command
   - Apply an erase programming pulse on the output pin (see electrical specification)
8. Calculate the bits to be set from EEP_OLD to EEP_NEW as EEP_PROG array
9. Write the EEPROM content from the EEP_PROG array to the EEPROM registers
10. Send the EEPROM write command
   - Apply a porgramming pulse on the output pin (see electrical specification)
11. Send a margin command
   - During the falling edge of the margin pulse on VDD, apply $V_{O,MARG}$ on the output (see electrical specification)
12. Read out the EEPROM content to the array EEP_PROG
13. Verify the EEP_PROG data against EEP_NEW to check the programming (no/all bits flipped)
   - Optionally, item 11 to 13 might be looped to find the exact V_{TH} EEPROM level.
   - If the acquired V_{TH} level is too low, do not continue and check for the failure.
14. Finally it is recommended to send a read command to check the status register again to see if the sensor is still running in the interface mode before switching off the device

Note: The EEPROM parity status bits are not meaningful at that time. Also the LOCK bit does not correspond to the settings in the EEPROM.

Detail one: How to set the TEST register:

1. Send a write command (TEST register set: CMD=0x09, ADR=0x1B)
2. Send the new data word for the TEST register

Detail two: How to read out the EEPROM content:

1. Send a block command (EEPROM data readout: CMD=0x03, ADR=0x10)
2. Read the first 8 data words of the EEPROM and store it in an array
3. Send a read command (EEPROM data readout: CMD=0x01, ADR=0x18)
4. Read the 9th data word of the EEPROM and store it in an array
5. Send a read command (EEPROM data readout: CMD=0x01, ADR=0x19)
6. Read the 10th data word of the EEPROM and store it in an array
7. Send a read command (EEPROM data readout: CMD=0x01, ADR=0x1A)
8. Read the 11th data word of the EEPROM and store it in an array

Detail three: How to set the EEPROM content:

1. Send a block command (EEPROM data writeout: CMD=0x0B, ADR=0x10)
2. Send the first 8 data words from the array to the EEPROM

3. Send a write command (EEPROM data write: CMD=0x09, ADR=0x18)
4. Send the 9th data word from the array to the EEPROM
5. Send a write command (EEPROM data write: CMD=0x09, ADR=0x19)
6. Send the 10th data word from the array to the EEPROM
7. Send a write command (EEPROM data write: CMD=0x09, ADR=0x1A)
8. Send the 11th data word from the array to the EEPROM

Detail four: How to calculate the bits to be cleared from EEP_OLD to EEP_NEW:

1. For each data word i in the arrays calculate:
2. EEP_PROG[i] = INVERT ((EEP_OLD[i] XOR EEP_NEW[i]) AND EEP_OLD[i])

Example of a calculated erase mask:

```
EEP_OLD:  0101010101010101
EEP_NEW:  0101110001010101
EEP_PROG: 1111111011111111
```

Detail five: How to calculate the bits to be set from EEP_OLD to EEP_NEW:

1. For each data word i in the arrays calculate:
2. EEP_PROG[i] = (EEP_OLD[i] XOR EEP_NEW[i]) AND EEP_NEW[i]

Example of a calculated program mask:

```
EEP_OLD:  0101010101010101
EEP_NEW:  0101110001010101
EEP_PROG: 0000100000000000
```

Detail six: How to determine the EEPROM margin voltages

The threshold voltage of EEPROM cells is dependent on the programming voltage and programming pulse length. The margin command can be used to check the threshold voltages of the programmed cells: A voltage $\mathrm{V}_{o,margin}$ is applied after the margin mode command (CMD No. OxF, see timing diagram in **Figure 13**). For EEPROM cells with a threshold voltage smaller than the applied $V_{o,margin}$, a '0' will be stored to the EEPROM registers, for those with a higher threshold voltage, a '1' will be written. By sweeping the applied $V_{o,margin}$, the effective threshold voltages of each EEPROM cell can be identified. The threshold voltages of cells programmed to ‘1’ can be found in this way.

In order to check the threshold voltages of EEPROM cells programmed to ‘0’, it is necessary to activate the “Margin zero on” bit in the test register (**Figure 16**). The smallest possible $V_{o,margin}$ is 0V, and it is therefore not possible to determine the threshold voltages below 0V.

Note: This routine can be merged with other (exemplary shown) routines. In that case only one initial frame (the very first interface access) is required after power-on.

4.4 DATA Access Example

The following steps are required to read out other internal data like the calibrated temperature and Hall value (as shown below). Of course these routines can be used for an EEPROM access as well (in that case also FEC off should be set to ’1’).

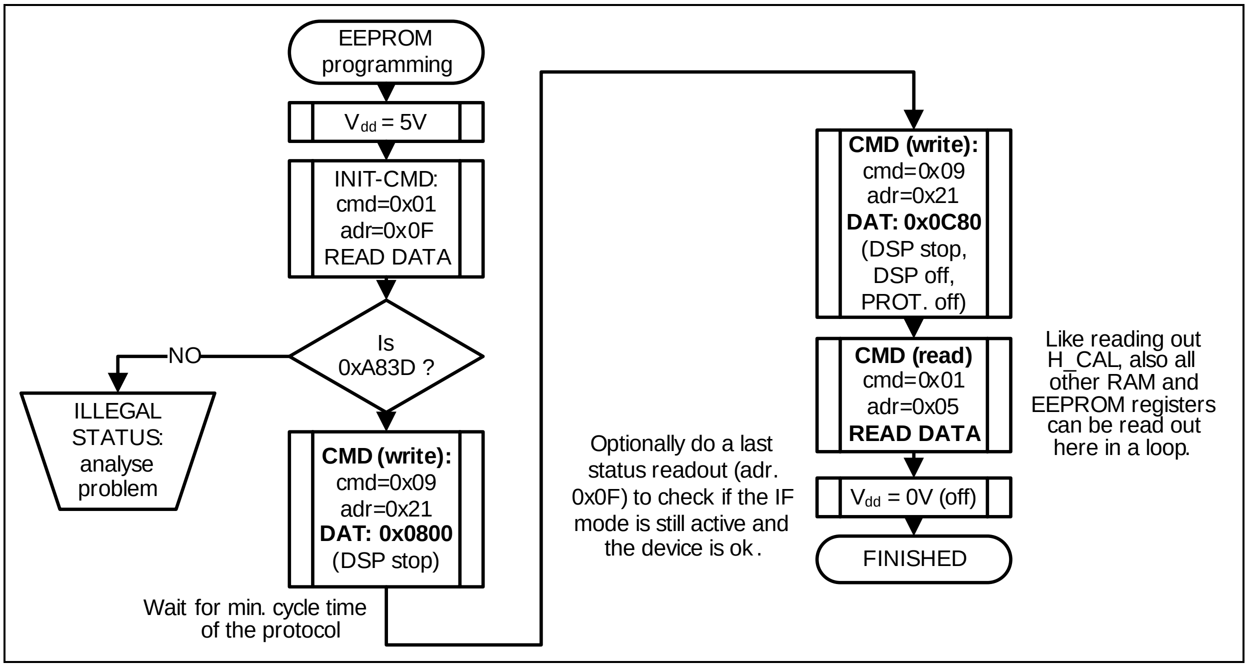

EEPROM programming
V_{dd} = 5V
INIT-CMD: cmd=0x01 adr=0x0F READ DATA
Is 0xA83D ?
NO
ILLEGAL STATUS: analyse problem
CMD (write): cmd=0x09 adr=0x21 DAT: 0x0800 (DSP stop)
Wait for min. cycle time of the protocol
Optionally do a last status readout (adr. 0x0F) to check if the IF mode is still active and the device is ok.
CMD (write): cmd=0x09 adr=0x21 DAT: 0x0C80 (DSP stop, DSP off, PROT. off)
CMD (read) cmd=0x01 adr=0x05 READ DATA
Like reading out H_CAL, also all other RAM and EEPROM registers can be read out here in a loop.
V_{dd} = 0V (off)
FINISHED

Figure 18 Basic data access flow

Description of the above flowchart:

1. Switch on the device
2. Send an inital command (status register readout)
3. Read the status data, check that the device is valid and the EEPROM content is valid
4. Set the test register: DSP stop=1 (see previous chapter)
5. Set the test register: DSP stop=1, DSP off=1, PROTOCOL off=1 (see previous chap.)
6. Send a read command (H_CAL)
   - Read the data word
   - This readout might be looped for reading out also other parameters (like T_CAL)
7. Finally it is recommended to send a read command to check the status register again to see, if the sensor is still running in the interface mode before switching off the device

Note: This routine can be merged with other (exemplary shown) routines. In that case only one initial frame (the very first interface access) is required after power-on. It is recommended to do a power cycle to get back to normal operating mode, even if it is possible by switching off “DSP stop”, “DSP off” and “PROTOCOL off”.

4.5 Temporary Overwrite of EEPROM Data

Following steps are required to readout other internal data like the calibrated temperature and Hall value (as shown below). As the error correction stays disabled, it is not necessary to use correct parity values for this temporary setup. In case the parity is always corrected (and it is desired to check the complete behavior and correct EEPROM array calculation), the “FEC off” bit could be switched off again after the temprary EEPROM write, too. Wait cycle time of protocol (min. time).

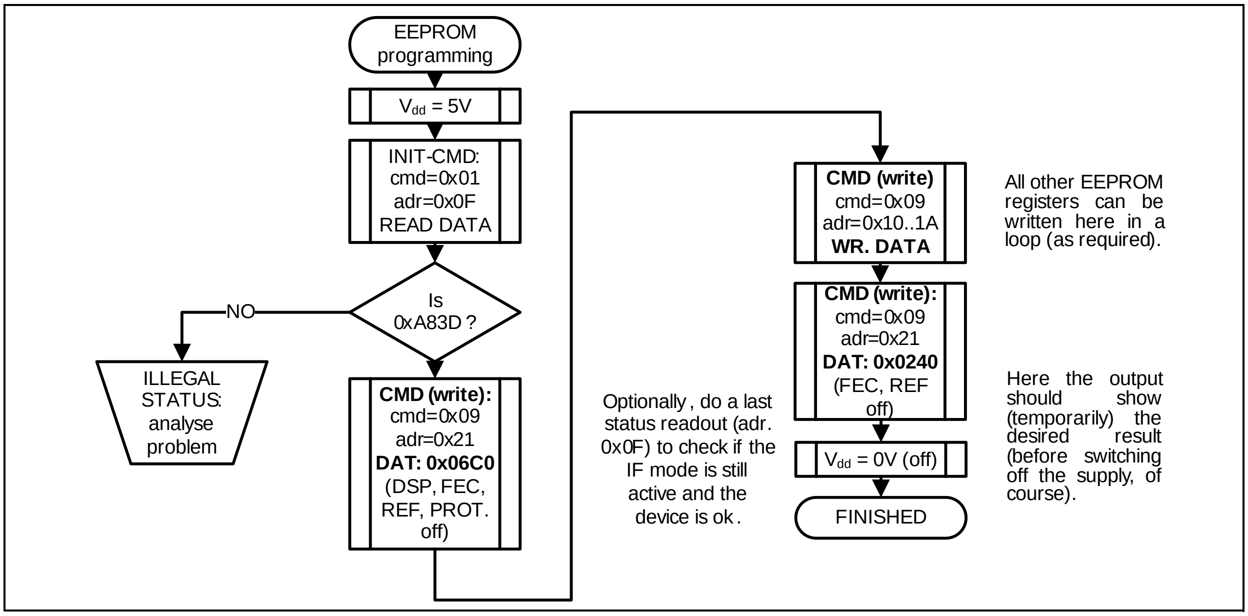

EEPROM programming
Vdd = 5V
INIT-CMD: cmd=0x01 adr=0x0F READ DATA
Is 0xA83D ?
NO
ILLEGAL STATUS: analyse problem
CMD (write): cmd=0x09 adr=0x21 DAT: 0x06C0 (DSP, FEC, REF, PROT. off)
CMD (write) cmd=0x09 adr=0x10..1A WR. DATA
CMD (write): cmd=0x09 adr=0x21 DAT: 0x0240 (FEC, REF off)
Vdd = 0V (off)
FINISHED
All other EEPROM registers can be written here in a loop (as required).
Optionally, do a last status readout (adr. 0x0F) to check if the IF mode is still active and the device is ok.
Here the output should show (temporarily) the desired result (before switching off the supply, of course).

Figure 19 Basic (temporary) EEPROM register overwrite flow

Description of the above flowchart:

1. Switch on the device.
2. Send an inital command (status register readout).
3. Read the status data, check that the device is valid and the EEPROM content is valid.
4. Set the test register: DSP off=1 FEC off=1 REF off=1 PROTOCOL off=1 (see previous chapter).
5. Send a write command (for any EEPROM register).
   The parity bits in the 16 bit data word may be kept zero, if FEC off =1.
6. Set the test register: FEC off=1 REF off=1 (see previous chapter).
   The device is now temporarily working with the new EEPROM setting.
7. Finally it is recommended to send a read command to check the status register again to see, if the sensor is still running in the interface mode before switching off the device.

Note: This routine can be merged with other (exemplary shown) routines. In that case only one initial frame (the very first interface access) is required after power-on.

5 Application Circuit for Programming

Figure 20 shows the connection of multiple sensors to a programmer.

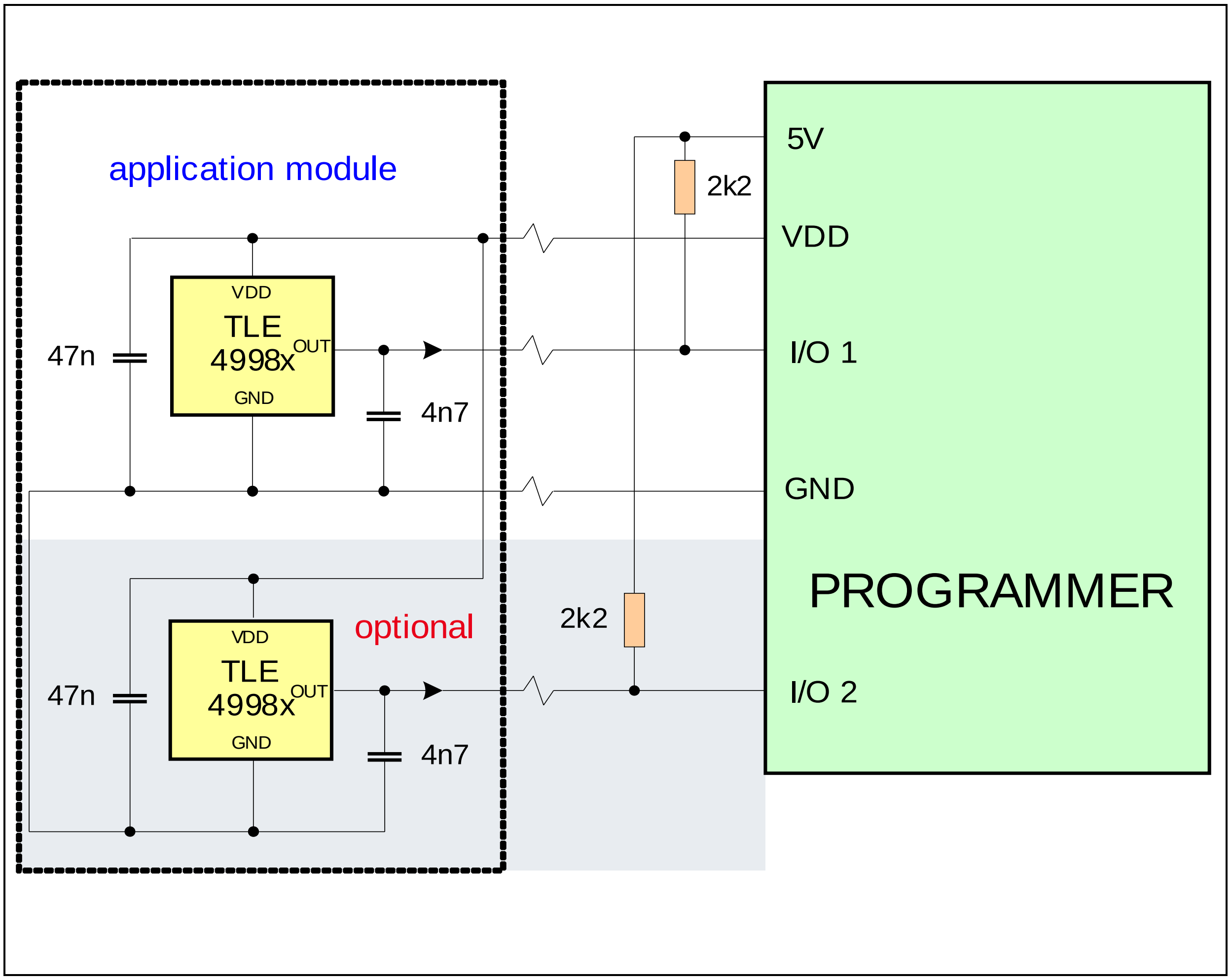


Figure 20 Application Circuit

Note: For calibration and programming, the interface has to be connected directly to the output pin.

www.infineon.com

Published by Infineon Technologies AG