LW57 系列 压 力 传 感 器

硅陶瓷系列
数字输出
模拟放大输出

应 用 领 域

▢ 呼吸机、麻醉机	▢ 肺活量计	▢ 医院室内气压控制
▢ 雾化器	▢ 风道静压	▢ VAV 调节系统
▢ HVAC 滤清器堵塞检测	▢ HVAC（暖通空调）变送器	

产 品 概 述

LW57 高精度硅陶瓷系列为压阻硅压力传感器，可提供指定满量程压力范围和温度范围读取压力的数字输出。LW57 系列通过使用板载专用集成电路（ASIC）针对传感器偏移、 灵敏度、温度效应和非线性进行了充分校准和温度补偿。经校准的压力输出值会在 1 kHz 左右更新。LW57 系列在 0°C 到 60°C 的温度范围内进行校准。该传感器可在 3.3 Vdc 或 5.0 Vdc 的单电源条件下工作。这些传感器测量绝压、差压和表压。绝压型号的传感器具备内部真空参照以及与绝压成比例的输出值。差压型号的传感器允许向感应模片的任意一侧加压。表压型号的传感器以大气压力为参考，提供与大气压力变化成比例的输出值。LW57 压力传感器适用于无腐蚀性、非离子气体（例如空气和其他干燥气体）。提供的选件可延伸这些传感器的性能，使其适用于无腐蚀性、非离子的液体。所有产品均遵循 ISO 9001 标准设计和制造。

产 品 特 点

- ▢ 多样化的封装：LW57 系列压力传感器为硅压阻精密压力传感器，采用模块化设计，具有双管朝上封装，可满足客户不同安装环境的需要
- ▢ 工作电压较低，能耗极小，供电电压可以为 3.3V 或 5V 输入
- ▢ 业界领先的长期稳定性：通过压力敏感芯片的优选和封装工艺的技术处理，作为微压力传感器与业内其他传感器相比表现出色，具有优异的长期稳定性
- ▢ 内部诊断功能可增强系统的可靠性
- ▢ 还提供了模拟输出选项
- ▢ 绝压、差压和表压类型
- ▢ 达到 0.25% FSS BFSL（满刻度量程最佳直线）的极高精确度
- ▢ 总误差带为 1% 的满刻度量程最大值
- ▢ 所有这些产品都同样具备业界领先的性能规格
- ▢ 与 I2C 或 SPI 兼容的 16 位数字输出，压力输出 16 位，可以同时监控温度，温度输出 16 位
- ▢ 模拟数字放大输出提供 10% 到 90% 输出或 5% 到 95% 输出
- ▢ 在 0℃ 到 60℃ 的温度范围内进行精密 ASIC 调节和温度补偿
- ▢ 符合 RoHS 标准
- ▢ 压力口特点：直径 3.175MM 倒钩状压力口可以稳固的和 2.38MM 内径的压力管牢固连接测试压力
- ▢ 客户定制：精度、总偏差和温度补偿范围以及信号输出方式等可根据客户需求定制，非标准产品请联系我们

标 准 压 力 范 围 量 程 （PSI， PA）

1 PSI	表压、差压	模拟放大和数字输出
2 PSI	表压、差压	模拟放大和数字输出
5 PSI	表压、差压	模拟放大和数字输出
15 PSI	表压、差压、绝压	模拟放大和数字输出
30 PSI	表压、差压、绝压	模拟放大和数字输出

100 pa	表压、差压	模拟放大和数字输出
1 Kpa	表压、差压	模拟放大和数字输出
10 Kpa	表压、差压	模拟放大和数字输出
30 Kpa	表压、差压、绝压	模拟放大和数字输出
100 Kpa	表压、差压、绝压	模拟放大和数字输出
200 Kpa	表压、差压、绝压	模拟放大和数字输出

额 定 值

参数	最小值	最大值	单位
电源电压 (Vsupply)	-0.3	5.25	VDC
任意引脚上的电压	-0.3	Vsupply +0.3	V
数字接口时钟频率： I2C SPI	100 50	400 800	kHz
ESD 敏感度 (人体模式)	2		Kv
储存温度	-40	125	℃
焊接时间和温度 铅焊料温度 (SIP、DIP) 回流峰值温度 (SMT)	最多 5 秒， 在 250℃ 时 最多 15 秒， 在 250℃ 时		

性 能 参 数

参数	最小值	典型值	最大值	单位	注释
电源电压 (Vsupply) (根据所选料号) 3.3 5.0	 3.0 4.75	 3.3 5.0	 3.6 5.25	 V	 2
电源电流 3.3 Vdc 电源 5.0 Vdc 电源		 3 4.5		 mA mA	
补偿温度范围	0		60	℃	3
工作温度范围	-40		125	℃	4
启动时间（从加电到数据准备就绪）		2.8	7.3	ms	
响应时间		0.46		ms	
SPI 和 I2C 低电平			0.2	Vsupply	
SPI 和 I2C 高电平	0.8			Vsupply	
SDA/MISO, SCL/SCLK, SS上拉电阻	1			Kohm	
精度			±0.25	% FSS	5、7
位置灵敏度			±0.15	% FSS	
综合偏差 (TEB)	-1%		1%	% FSS	6
过载压力		3		倍	9
爆破压力		5		倍	10
输出分辨率	16			位	

环 境 规 格

参数	特性	注释
湿度：仅干燥气体	0% 到 95% RH，非冷凝	
振动	MIL-STD-202F，方法 214，条件 F (20.7g 随机)	
冲击	MIL-STD-202F，方法 213B，条件 F	
寿命	100 万次循环	8
回流焊	J-STD-020D.1 MSL 湿度灵敏度级别 1	

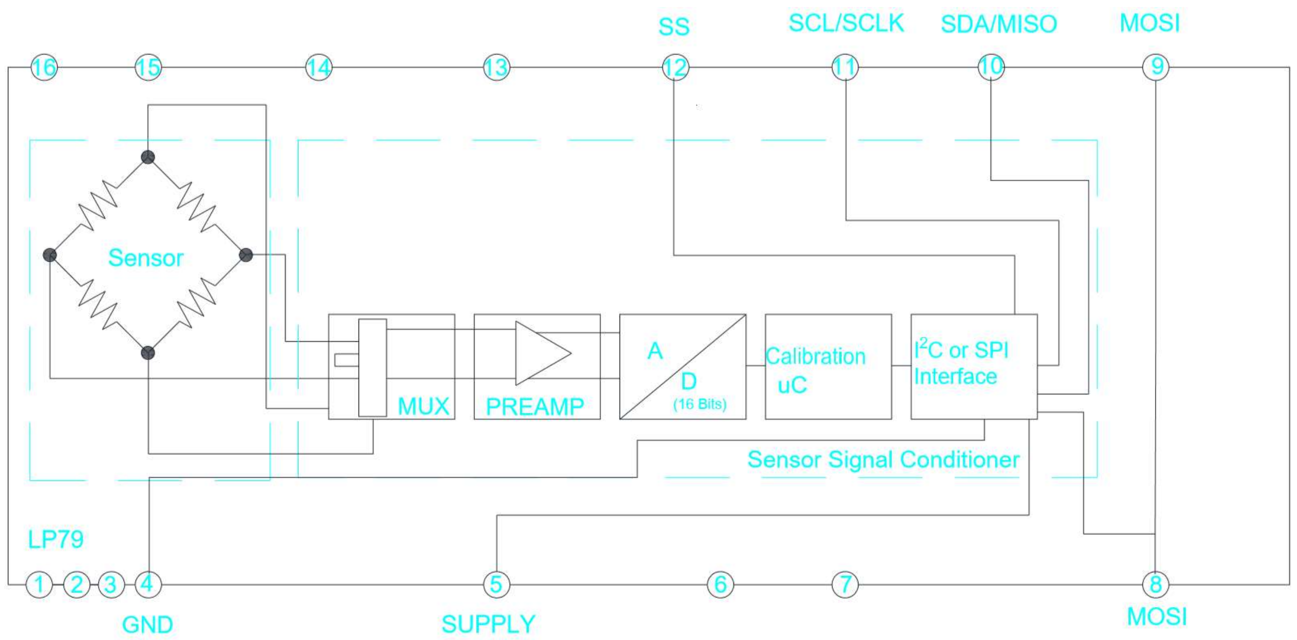
被测介质接触材料

盖子	PPS	PPS
基材	氧化铝陶瓷	氧化铝陶瓷
粘合剂	环氧树脂、硅树脂	环氧树脂、硅树脂
电子组件	陶瓷、玻璃、焊料、硅	硅、玻璃、金、焊料

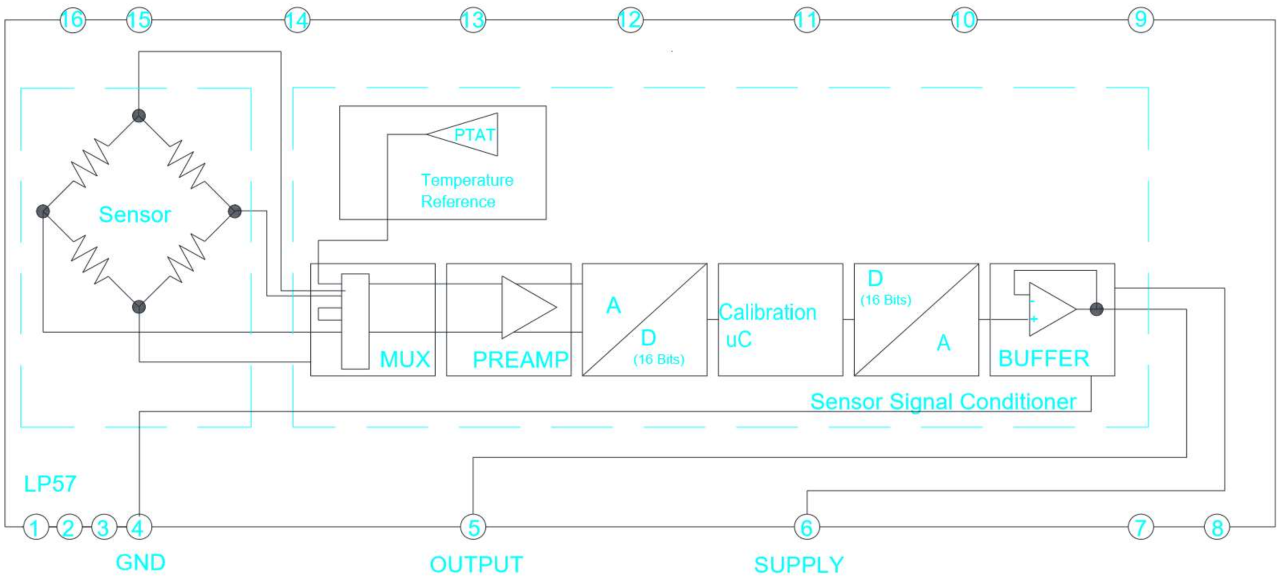
注释:
1、 额定值是设备在不损坏的前提下所能承受的最大极限。
2、 该传感器不受反向极性保护。将错误的引脚与电源连接或者接地可能会导致电气故障。
3、 补偿温度范围是指传感器可以在特定的性能限制下产生与压力成比例的输出的温度范围。
4、 工作温度范围是指传感器可以产生与压力成比例的输出的温度范围，但不一定在特定性能限制范之内。
5、 精度：相对适用于在 25°C 时的压力范围内所测输出的最佳直线 (BFSL) 的最大输出偏差。包括所有因压力非线性、压力滞后和不重复性造成的误差。
6、 综合偏差：相对整个补偿温度和压力范围内理想传递函数的最大偏差。包括所有因偏置、满刻度量程、压力非线性、压力滞后、可重复性、偏置热效应、量程热效应和热滞后造成的误差。
7、 满刻度量程 (FSS) 是指在压力范围最大限制值 (Pmax.) 和最小限制值 (Pmin.) 处测得的输出信号之间的代数差。
8、 寿命可能因传感器使用的特定应用而有所变化。
9、 过压：可安全施加到产品的最大压力，使产品在压力返回到工作压力范围时规格保持不变。施加过高的压力可能会对产品造成永久损坏。除非另有规定，否则这适用于工作温度范围内任何温度下的所有可用压力口。
10、爆破压力：可施加到产品的任意压力口而不造成压力媒介脱离的最大压力。在经受任何超过爆破压力的压力之后，产品将不能正常工作。
11、客户定制请联系零壹智控业务人员。

注意：
产品损坏
- 确保液体介质仅用于压力口 A；压力口 B 与液体不相容。
- 确保液体介质不含颗粒。所有 LW57传感器均为终端密封设备，颗粒会在传感器内积聚，造成设备损坏或影响传感器输出。
- 建议将传感器的压力口 A 朝下放置，这样系统中的颗粒就不容易进入并停留在压力传感器内。
- 确保液体介质在干燥时不会产生残留物。传感器内的堆积物可能会影响传感器输出。清洗终端密封的传感器十分困难，而且无法有效地去除残留物。
- 确保液体介质与接液材料相容。不相容的液体介质只会降低传感器的性能，并可能导致传感器故障。
- 不遵循这些说明可能会导致产品损坏。

等效电路



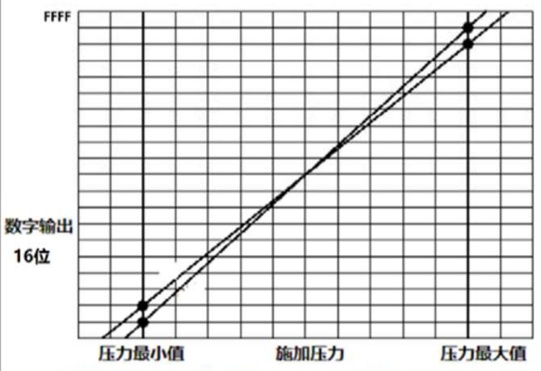
LW57 数字输出等效电路



LW57 模拟放大输出等效电路

压力和温度输出对应公式

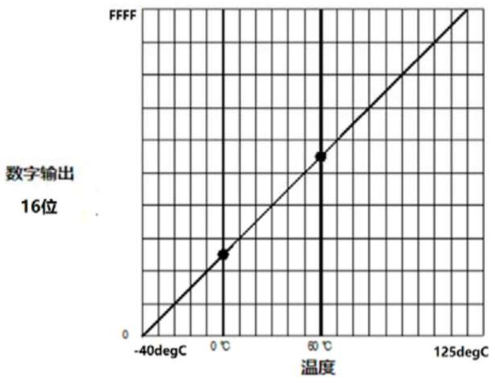
压力转换方程



A型：压力数字输出=80% *65535/(压力最大值-压力最小值)*（施加压力-压力最小值）-26214

B型：将上述公式80%改为90%，10%改为5%

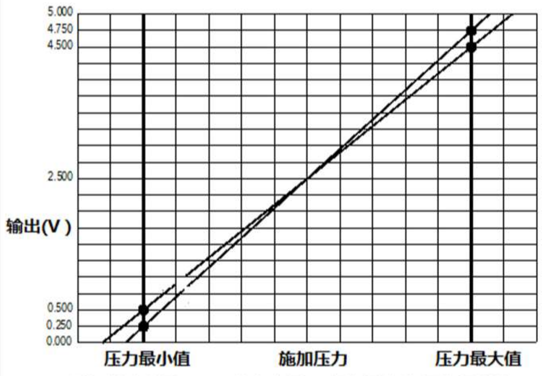
温度转换方程



温度数字输出=（待测温度-（-40））*65535 /(125-(-40))

LW57 数字输出

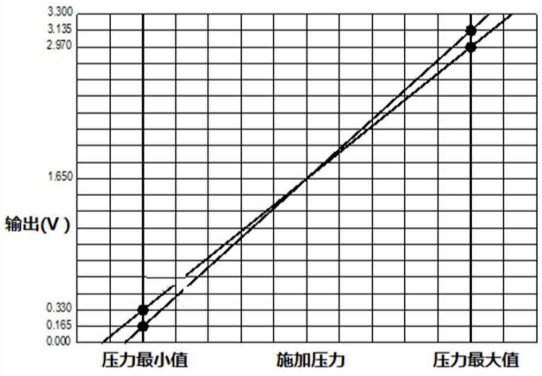
压力转换方程，电压输入5V



A型：输出电压=80%*输入电压/（压力最大值-压力最小值）*（施加压力-压力最小值）+0.500

B型：将上述公式80%改为90%，0.500改为0.250

压力转换方程，电压输入 3.3V



A型：输出电压=80%*输入电压（压力最大值-压力最小值）*（施加压力-压力最小值）+0.330

B型：将上述公式80%改为90%，0.330改为0.165

LW57 模拟放大输出

压力类型	说明
差压	输出与施加到各压力口的压力差成比例
表压	输出与施加到压力和大气（环境）压力之间的差值成比例
绝压	输出与施加压力和真空压力之间的差值成比例

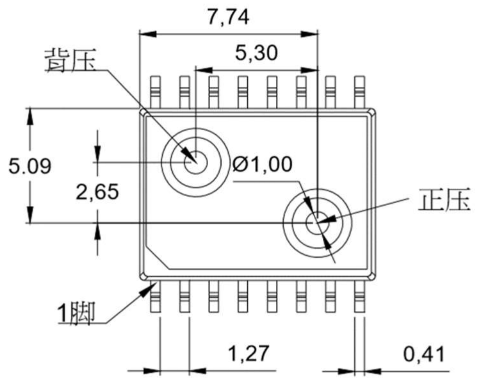
输出百分比

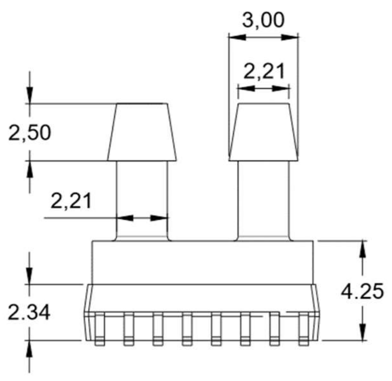
输出百分比（%）	数字计数（十进制）	模拟放大（5V）	模拟放大（3.3V）
0	-32768	0	0
5	-29491	0.25	0.165
10	-26214	0.5	0.33
50	0	2.5	1.65
90	26214	4.5	2.97
95	29491	4.75	3.135
100	32768	5	3.3

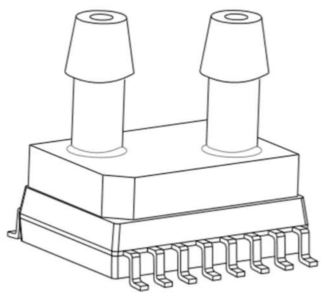
脚位定义

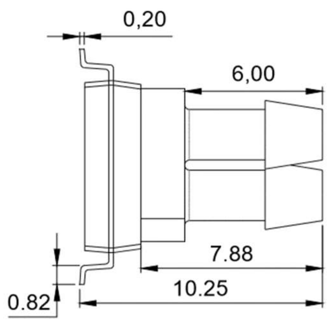
输出类型/脚位	Pin 1-3	Pin 4	Pin 5	Pin 6	Pin 8-9	Pin 10	Pin 11	Pin 12	Pin 13-16
模拟输出	Blank	Vss	Vdd	Sig	Blank	Blank	Blank	Blank	Blank
数字输出 (SPI)	Blank	Vss	Vdd	Blank	MOSI	MISO	SCLK	SS	Blank
数字输出 (I2C)	Blank	Vss	Vdd	Blank	Blank	SDA	SCL	Blank	Blank

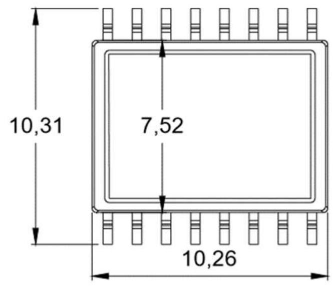
尺 寸（毫米）

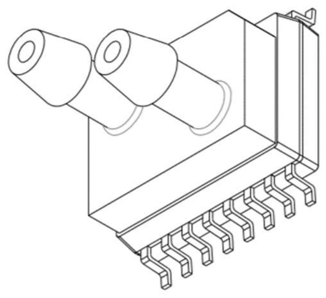












备注:
1、差压时，当压力口 A 的压力大于压力口 B 时，输出大于输入电压或者 0；
当压力口 A 的压力等于压力口 B 时，输出小于输入电压或者 0 。
2、表压时，当压力口 A 的压力大于压力口 B 时，输出大于输入电压或者 0 。
3、压力口 A 在调试的时候始终加的是正压。

订 购 指 南

LW57 A-DS 3 A I XXXX X S X

Model
LW57

Output Type
A = Analog
D = Digital

Package Style
DS = Dual Tube

Supply Voltage
3 = 3.3 Vdc
5 = 5.0 Vdc

Output Type
A = 10% to 90%
B = 5% to 95%

Pressure Type
I = I2C (Addr. 0x28H)
J = I2C (Addr. 0x36H)
K = I2C (Addr. 0x46H)
S = SPI (Not Available for 'L' Pin Style)
0 = I2C (Addr. Ox48H)
……
9 = I2C (Addr. Ox51H)

Pressure Range
002 M = 2 Mpa
004 W = 4 InH2O
005 K = 5 Kpa
010 P = 10 Psi
020 B = 20 Bar
100 A = 100 Pascal

Pressure Type
D = Differential
G = Gauge
A = Absolute

Pin Style
S = SMT

Humidity Protection
C = Gel Coating
Blank

零壹智控 INTELLIGENT

零壹智能控制技术（深圳）有限公司
www.linkwon.com.cn
Email：sales1@bill-well.com
深圳市 南山区 南光路 17号 现代城华庭 4栋25A
Tel：0755-2664 7945 / 189 2389 8109 Fax：0755-2641 9680

I2C 应用说明和相关例程

1：Electrinal table

Item	Description	Condition	Symbol	Min	Typ	Max	Unit
1	SDA output low voltage	I_{SDA}=3 mA	$V_{SDA,OL}$	0		0.4	V
2	Low-to-High transition threshold	pins SA0, SCL	$V_{SDA,LH}$	0.5	0.6	0.7	*VDD
3	High-to-Low transition threshold	pins SA0, SCL	$V_{SDA,HL}$	0.3	0.4	0.5	*VDD
4	I^{2}C clock frequency		fSCL	0		400	kHz
5	Bus free time between a START and STOP condition		tBUSF	1300			ns
6	Clock low time		tLO	1300			ns
7	Clock high time		tHI	600			ns
8	START condition hold time		tSH	100			ns
9	Data setup time		tSU	100			ns
10	Data hold time		tH	0			ns
11	Setup time for repeated START condition		tRSH	600			ns
12	Setup time for STOP condition		tPSU	600			ns
13	Rise time for SDA and SCL signals		tR			300	ns
14	Fall time for SDA and SCL signals		tF			300	ns

2: I^{2}C Interface

The SQMEAS pressure sensor features an I^{2}C slave interface. This interface provides direct access to registers of the memory of the pressure sensor. An external I^{2}C master (e.g. a microcontroller) can read from and write to memory addresses (registers) of the device using the following commands:

•**Random write:** Sets a memory address and writes data to consecutive memory addresses of the device starting at the set memory address.

•**Random read:** Sets a memory address and reads data from consecutive memory addresses of the device starting at the set memory address.

•**Read last:** Reads data from the device starting at the last memory address set by the master. This facilitates repeated reading of the same memory addresses without transmitting a memory address first.

All reads/writes must start at word aligned addresses (i.e. LSB of memory address equals 0) and read/write an even number of bytes.

3. I^{2}C Command Format

The SQMEAS pressure sensor uses a standard 7-bit I^{2}C slave address field. The LSB of the slave address specifies the frame type used to perform read and write operations.

For LSB = 0 the protocol is compatible to standard I^{2}C EEPROMs, for LSB = 1 the protocol is extended by a CRC protection. Thus, each device occupies two I^{2}C addresses: even addresses are for standard EEPROM compatible protocols and odd addresses are for CRC protected protocols. Unprotected and CRC protected frames can be interleaved.The two different frame types - standard EEPROM (without CRC) or CRC protected - are shown in the next two figures.

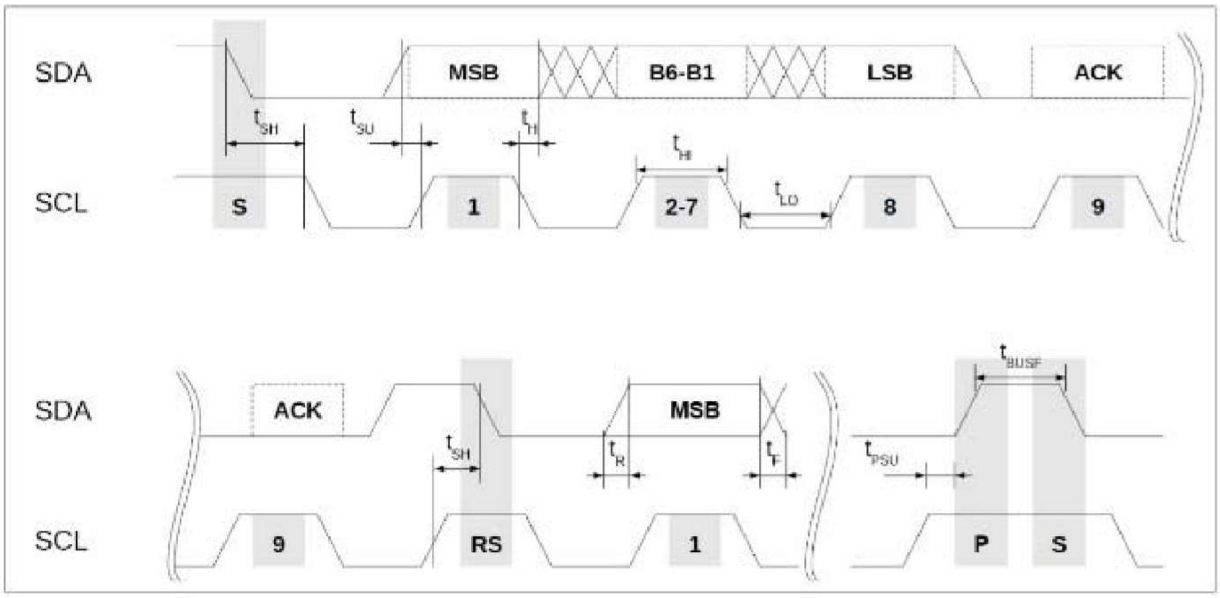


Figure 1: I^{2}C Interface Timing Diagram

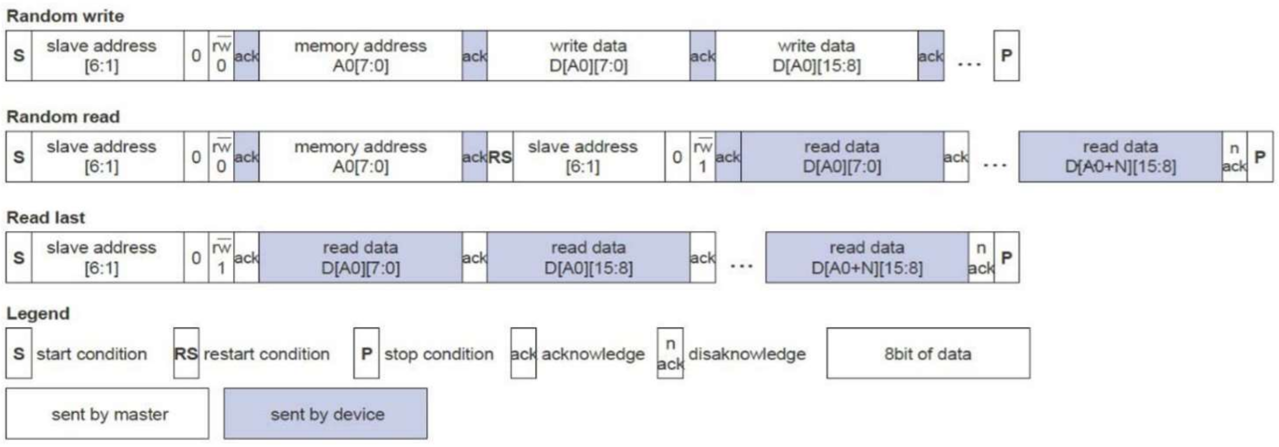

Random write
S | slave address [6:1] | 0 | rw 0 | ack | memory address A0[7:0] | ack | write data D[A0][7:0] | ack | write data D[A0][15:8] | ack | … | P
Random read
S | slave address [6:1] | 0 | rw 0 | ack | memory address A0[7:0] | ack | RS | slave address [6:1] | 0 | rw 1 | ack | read data D[A0][7:0] | ack | … | read data D[A0+N][15:8] | n ack | P
Read last
S | slave address [6:1] | 0 | rw 1 | ack | read data D[A0][7:0] | ack | read data D[A0][15:8] | ack | … | read data D[A0+N][15:8] | n ack | P
Legend
S start condition
RS restart condition
P stop condition
ack acknowledge
n ack disaknowledge
8bit of data
sent by master
sent by device

Figure 2: I^{2}C Read / Write Commands - Standard EEPROM compatible protocol

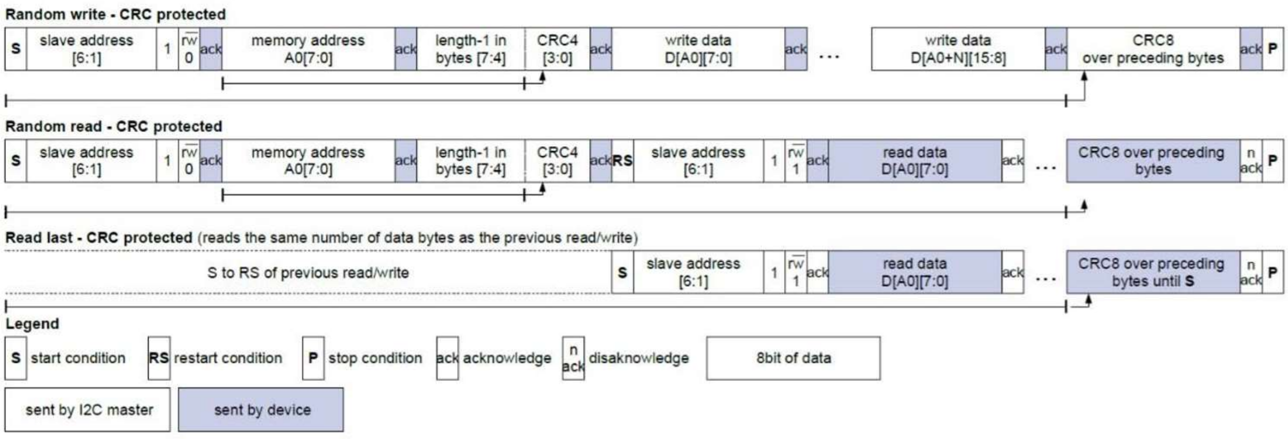

Random write - CRC protected
S | slave address [6:1] | 1 | rw 0 | ack | memory address A0[7:0] | ack | length-1 in bytes [7:4] | CRC4 [3:0] | ack | write data D[A0][7:0] | ack | … | write data D[A0+N][15:8] | ack | CRC8 over preceding bytes | ack | P
Random read - CRC protected
S | slave address [6:1] | 1 | rw 0 | ack | memory address A0[7:0] | ack | length-1 in bytes [7:4] | CRC4 [3:0] | ack | RS | slave address [6:1] | 1 | rw 1 | ack | read data D[A0][7:0] | ack | … | CRC8 over preceding bytes | n ack | P
Read last - CRC protected (reads the same number of data bytes as the previous read/write)
S to RS of previous read/write | S | slave address [6:1] | 1 | rw 1 | ack | read data D[A0][7:0] | ack | … | CRC8 over preceding bytes until S | n ack | P
Legend
S start condition
RS restart condition
P stop condition
ack acknowledge
n ack disaknowledge
8bit of data
sent by I2C master
sent by device

Figure 3: I^{2}C Read / Write Commands - CRC protected protocol

The memory address field sets the byte address of the first memory location to be read from or written to. Only 16-bit-word aligned reads/writes are supported, i.e. the LSB of memory address has to be always zero. The read/write data is transferred MSB first, low byte before high byte.

The length field (bits[7:4]) required for CRC protected frames specifies the number of data bytes to be transferred decremented by one, i.e. a value of 0001b corresponds to two bytes. All frames must transfer an even number of bytes. The maximum length for CRC protected read/write frames is 16/4 bytes. For unprotected frames the length is unlimited.

The CRC4 and CRC8 for redundancy check are computed in the same bit and byte order as the transmission over the bus. The polynomials employed are:

CRC4: polynomial 0x03; initialization value: 0x0F

CRC8: polynomial 0xD5; initialization value: 0xFF

If a CRC error occurs, then the event bit "com_crc_error" in the STATUS register will be set.

4. I^{2}C Command Examples

For all examples below the 7-bit device slave address used is 0x6C for unprotected commands, and 0x6D for CRC protected commands, respectively. These addresses are the default addresses and are used unless otherwise stated in the part number specific data sheet.

The command sequence following describes an unprotected Read command (without CRC) of 3 subsequent 16-bit words starting at memory address 0x2E to read the corrected IC temperature, corrected pressure signal, and (synchronized) status bits of the sensor.

Random Read

Byte#	0	1	2	3	4	5	6	7	8
SBM (sent by master)	0xD8	0x2E	0xD9						
SBM comment	slave address 6C + LSB = 0 for *Write*	memory address	slave address 6C + LSB = 1 for *Read*						
SBS (sent by sensor)				0xF2	0x7D	0xEA	0x82	0x1E	0x00
SBS comment				DSP_T (Lo-Byte) ad. 0x2E	DSP_T (Hi-Byte)	DSP_S (Lo-Byte) ad. 0x30	DSP_S (Hi-Byte)	sync'ed Status (b7 - b0) ad. 0x32	sync'ed Status (b15 - b8)

Random read with CRC protection

The following sequence describes the CRC protected version of reading 3 subsequent 16-bit words starting at memory address 0x2E to read the corrected IC temperature, corrected pressure signal, and (synchronized) status bits of the sensor.

Byte#	0	1	2	3	4	5	6	7	8	9	10
SBM (sent by master)	0xDA	0x2E	0x5B	0xDB							
SBM comment	slave address 6D + LSB = 0 for *Write*	memory address	3: length = 4Byte B: CRC4	slave address 6D + LSB = 1 for *Read*							
SBS (sent by sensor)					0xF2	0x7D	0xEA	0x82	0x1E	0x00	0x65
SBS comment					DSP_T (Lo-Byte) ad. 0x2E	DSP_T (Hi-Byte)	DSP_S (Lo-Byte) ad. 0x30	DSP_S (Hi-Byte)	sync'ed Status (b7 - b0) ad. 0x32	sync'ed Status (b15 - b8)	CRC8 (calc'd)

The following sequence writes one 16-bit word to address 0x22 (without CRC protection). This will copy 0x6C32 into the command register CMD to move the component to Sleep Mode.

Random Write:

Byte#	0	1	2	3
SBM (sent by master)	0xD8	0x22	0x32	0x6C
SBM comment	slave address 6C + LSB = 0 for *Write*	memory address	Lo-Byte written to CMD[7:0]	Hi-Byte written to CMD[15:8]
SBS (sent by sensor)				
SBS comment				

5. Register Descriptions

Register Read or Write are performed via the digital communication interface. After power-up of the IC all registers except STATUS and CMD are write protected.

Command register:

0X22	**CMD**			
bits	**name**	**default**	**rw**	**description**
15:0	cmd	0	w	Writing to this register controls the state of the BAP device. 0x6C32: SLEEP Mode Initiate the power state SLEEP, powering down the ASIC 0xB169: RESET Performs a reset. After reset the power-up sequence will be executed, i.e. the registers are loaded with data from the configuration memory, also a CRC check is performed.

Temperature register:

0X2E	DSP_T			
bits	**name**	**default**	**rw**	**description**
15:0	**Dsp_T**		**r**	Corrected temperature measurement value of the sensor. Whenever this register is updated with a new measurement the STATUS.dsp_t_up event bit is set.

Pressure register:

0X30	DSP_P			
bits	**name**	**default**	**rw**	**description**
15:0	**Dsp_P**		**r**	corrected pressure measurement value of the sensor. Whenever this register is updated with a new measurement the STATUS.dsp_s_up event bit is set.

The registers DSP_T and DSP_S contain invalid data after power-up until the first temperature and pressure values have been measured by the device and transferred to these registers. In case a NVM CRC error occurred, the DSP_T and DSP_S registers would never be updated. ***Thus, after power up it is necessary to wait until the STATUS.dsp_s_up and dsp_t_up bits have been set at least once before using the temperature or pressure data.***
It is not sufficient to wait just for a fixed time delay.

Status register - synchronized:

0X32	Status_sync				
bits	**name**	**default**	**rw**	**type**	**description**
0	idle	0	rw	status	STATUS.idle
1	- reserved -	0	rw	event	reserved
2	- reserved -	0	rw	event	reserved
3	dsp_s_up	0	rw	event	when DSP_S is read STATUS.dsp_s_up is copied here
4	dsp_t_up	0	rw	event	when DSP_T is read STATUS.dsp_t_up is copied here
5	- reserved -	0	rw	status	reserved
6	- reserved -	0	rw	status	reserved
7	bs_fail	0	rw	event	STATUS.bs_fail
8	bc_fail	0	rw	event	STATUS.bc_fail
9	- reserved -	0	rw	event	reserved
10	dsp_sat	0	rw	status	STATUS.dsp_sat
11	com_crc_error	0	rw	event	STATUS.com_crc_error
12	- reserved -	0	rw	status	reserved
13	- reserved -	0	rw	status	reserved
14	dsp_s_missed	0	rw	event	STATUS.dsp_s_missed
15	dsp_t_missed	0	rw	event	STATUS.dsp_t_missed

The bits STATUS_SYNC[15:5,0] are identical to the bits STATUS[15:5,0].

The bits STATUS_SYNC[4:3] are copied from the STATUS register when the corresponding DSP registers are read. First reading the DSP registers and then STATUS_SYNC ensures that both values are consistent to each other.

The synchronized status STATUS_SYNC register can be used to continuously poll the pressure, temperature and status of the device with a single read command by reading three 16 bit words starting at address 0x2E. By evaluating STATUS_SYNC.dsp_t_up and STATUS_SYNC.dsp_s_up it can be determined if the values in DSP_T and DSP_S acquired during the same read contain recently updated temperature or pressure values.

Status register:

0X36	**Status**				
bits	**name**	**default**	**rw**	**type**[1]	**description**
0	idle	0	rw	status	0: chip in busy state 1: chip in idle state
1	- reserved -	0	rw	event	reserved
2	- reserved -	0	rw	event	reserved
3	dsp_s_up	0	rw	event	1: DSP_S register has been updated. Cleared when DSP_S is read
4	dsp_t_up	0	rw	event	1: DSP_T register has been updated. Cleared when DSP_T is read.
5	- reserved -	0	rw	status	reserved
6	- reserved -	0	rw	status	reserved
7	bs_fail	0	rw	event	1: bridge supply failure occurred
8	bc_fail	0	rw	event	1: sensor bridge check failure occurred
9	- reserved -	0	rw	event	reserved
10	dsp_sat	0	rw	status	1: a DSP computation leading to the current DSP_T or DSP_S values was saturated to prevent overflow
11	com_crc_error	0	rw	event	1: communication CRC error
12	- reserved -	0	rw	status	reserved
13	- reserved -	0	rw	status	reserved
14	dsp_s_missed	0	rw	event	1: dsp_s_up was 1 when DSP_S updated
15	dsp_t_missed	0	rw	event	1: dsp_t_up was 1 when DSP_T updated

1)"Event" type flags remain set until cleared by writing '1' to the respective bit position in STATUS register (not STATUS_SYNC). Writing 0xFFFF to the STATUS register will clear all event bits. "Status" type flag represents a condition of a hardware module of the IC and persists until the condition has disappeared.

6. I2C Examples

```
/*******************************************************************************

* Function Name  : SKATER_I2C_MultRead

* Description    : 多字节写读取指定地址的数据（7 位地址）.

* Input        : - pBuffer：数据存储区.

*              - I2C_SLAVE_ADDRESS : 设备地址。

*              - RegAddr:起始寄存器地址.

*              - NumByteToRead：连续读取的字节数目.
* Output       : None

* Return       : None

*******************************************************************************/

void I2C_MutiRead(uint8_t* data, uint8_t dev, uint8_t reg,uint8_t NumByteToRead)

{

  uint8_t count=0;
          IIC_Start();//产生一个起始条件

          IIC_Send_Byte(dev);      //发送写命令

          IIC_Wait_Ack();//应答

          IIC_Send_Byte(reg);  //发送地址

 IIC_Wait_Ack();            //应答

          IIC_Start();//产生一个起始条件

          IIC_Send_Byte(dev+1);  //进入接收模式

          IIC_Wait_Ack();//应答


  for(count=0;count<NumByteToRead;count++){


                    if(count!=NumByteToRead-1)data[count]=IIC_Read_Byte(1);  //带 ACK 的读数据

                             else  data[count]=IIC_Read_Byte(0);      //最后一个字节 NACK

          }

  IIC_Stop();//产生一个停止条件

  return ;


}
```

```
float GetPa(void)

{

        uint8_t data[6]={0};

        int16_t PaD=0,Temp=0;

        float Per[6]={0},Psum=0,Pmax=-1000,Pmin=1000;//Psum 总和， Pmax：最大压力值， Pmin：最小压
力值

        Pa_flag=0;

        for(uint8_t i=0;i<4;i++)

        {

//              HAL_I2C_Mem_Read(&hi2c1,0xD8, 0x2E, I2C_MEMADD_SIZE_8BIT,data, 6, 1000);

                I2C_MutiRead(data, 0xD8,0x2E,6);//读取传感器数据放到 data 里，data：数据 buff、0xD8：
从地址，0x2E：温度寄存器地址，6：连续读 6 个；

                Temp=(data[1]<<8)+data[0];//温度：data[1]：数据高 8 位，data[0]：数据低 8 位

                PaD=(data[3]<<8)+data[2];                    //压力大小，高地址+低地址

                Per[i]=(-500.0+(PaD+26214)*1000.0/52428)*1.0;  //6895

                if(Per[i]>Pmax)

                        Pmax=Per[i];

                if(Per[i]<Pmin)

                        Pmin=Per[i];

                Psum+=Per[i];

        }

        Pa=(Psum-Pmax-Pmin)/2-FlashParameter.Offset;


        Tempr=-40-2+(Temp+32768)*165/65535;                          //2 度偏差

        Pa_flag=1;

        if(FlashParameter.Range==1)                                //40-50Pa

        {

                if(fabs(Pa)>=50)

                        Alarm=1;

                else if(fabs(Pa)<=40)
```

```
                        Alarm=0;
            }
            else                //25-30Pa
            {
                  if(fabs(Pa)>=30)
                        Alarm=1;
                  else if(fabs(Pa)<=25)
                        Alarm=0;
            }
            return Pa;
}
```