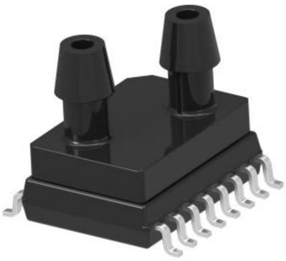


TPS MEDIUM PRESSURE DIGITAL & ANALOG SENSOR

Gage & Differential Pressure Sensors

The TPS (TE Connectivity Pressure Sensors) are medium pressure MEMS sensors offering state-of-the-art pressure transducer technology and CMOS mixed signal processing technology to produce either an analog and/or digital output fully conditioned, multi-order pressure and temperature compensated. This series provide a JEDEC standard SOIC-16 package with vertical porting option. It is available in gage and differential configurations. With the dual porting, a reference measurement is possible to minimize errors due to changes in ambient pressure.

The total error band after board mount and system level autozero is less than 1%FS. The warmup behavior and long-term stability further confirms its expected performance over the life of the part.

Combining the pressure sensor with a signal-conditioning ASIC in a single package simplifies the use of advanced silicon micro-machined pressure sensors. The pressure sensor can be mounted directly on a standard printed circuit board and a high level, calibrated pressure signal can be acquired from the digital interface. This eliminates the need for additional circuitry, such as a compensation network or microcontroller containing a custom correction algorithm.

The TPS products are shipped in tape & reel.

Features

- Pressure ranges:
  - From 5 to 30 PSI
- Pressure type:
  - ✓ gage
  - ✓ differential
  - ✓ asymmetric differential
- 24 bits ADC
- I^{2}C digital and analog output interface available
- Pressure total error band : ± 1% FS (digital & analog)
- Pressure calibrated and temperature compensated output
- Compensated temperature range: -20 to 85°C

Applications

- Patient monitoring
- Ventilators
- Gas Flow Instrumentation
- Air Flow Measurement
- Pressure Transmitters
- Pneumatic Gauges
- Pressure Switches
- Safety Cabinets

1 PERFORMANCE SPECIFICATION

1.1 Part Number & Calibrated Pressure Ranges

Dual vertical port configuration :

Part number	Alias[1]	P_{MIN} (PSI)	P_{MAX} (PSI)
20032306-00	TPS-005SG-CA1N-00-T	0	+5
20032307-00	TPS-015SG-CA1N-00-T	0	+15
20032308-00	TPS-030SG-CA1N-00-T	0	+30
20032309-00	TPS-005SD-CA1N-00-T	-5	+5
20032310-00	TPS-015SD-CA1N-00-T	-15	+15

Note :

1. Alias description is given on last datasheet page.

1.2 Absolute Maximum Ratings

All parameters are specified at VDD = 5.0 V supply voltage at 25°C, unless otherwise noted.

Characteristic	Symbol	Min	Max	Units
Compensated Temperature	T_{COMP}	-20	85	°C
Operating Temperature[a]	T_{OP}	-40	105	°C
Storage Temperature[a]	T_{STG}	-40	125	°C
Supply Voltage	V_{DD}	-0.3	6	V
Media Compatibility[a]		Clean, dry air compatible with wetted materials [b]		

Burst pressure and proof pressure by pressure range

Max Operating Pressure range Pmax (PSI)	Proof Pressure[a, c] P_{Proof} (PSI)	Burst Pressure[a, d] P_{Burst} (PSI)
Up to P_{max} = 5PSI	25	40
Up to P_{max} = 30PSI	45	75

Notes:

a) Tested on a sample basis.
b) Wetted materials include Silicon, glass, gold, aluminum, copper, silicone, epoxy, mold compound.
c) Proof pressure is defined as the maximum pressure to which the device can be taken and still perform within specifications after returning to the operating pressure range.
d) Burst pressure is the pressure at which the device suffers catastrophic failure resulting in pressure loss through the device.

1.3 ESD

Description	Condition	Symbol	Min	Max	Units
ESD HBM Protection at all Pins	JEDEC JESD47	$V_{ESD(HBM)}$	-2	2	kV

1.4 External Components

Description	Symbol	Min	Typ	Max	Units
Supply bypass capacitor	C_{VDD}		100		nF

1.5 Operating Conditions

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Supply Voltage	VDD		3.0	5.0	5.5	V
Sleep supply Current	Islp_25oC	VDD = 5.0V, T = 25°C (no conversion, DAC off)		1.8	8	µA
Standby supply Current	Isty_25oC	VDD = 5.0V, T = 25°C (no conversion, DAC off, fast_start =”1”)		156	200	µA
Supply current during analog output	Iaout	VDD = 5.0 V, T = 25°C, hvreg off, buffer on, ratiometric output		362		µA
Supply current during active conversion[1]	Iac_p Iac_Tr Iac_Tdsvdd	VDD = 5.0 V, T = 25°C, svdd = 1.8 V, fadc = 1 MHz excluding sensor current pressure resistive temperature diode temperature		2058 1857 1715		µA

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Conversion time for 1 conversion(P, T1 or T2) based on: fadc = 1 MHz fast start no CRC no reload with calculation with transfer add 10 us from sleep mode	Tconv	OSR = 0 OSR = 1 OSR = 2 OSR = 3 OSR = 4 OSR = 5 OSR = 6 OSR = 7 OSR = 8 OSR = 9	0.07 0.10 0.15 0.27 0.50 0.96 1.89 3.76 7.48 14.93	0.08 0.11 0.17 0.30 0.56 1.07 2.09 4.14 8.24 16.43	0.09 0.12 0.19 0.33 0.62 1.19 2.32 4.59 9.14 18.24	ms
Start up time	Tstart	Applying Power Supply to digital output ready		16.2		ms
Wake up time	Twaket	Wake up from sleep mode Wake up from standby		30 0		µs
Digital I/O leakage	Ileak	VDD = 5.0 V, T = 25°C	-1		1	µA

Note :

1. Analog output add 200µA

1.6 Operating Characteristics Table

All parameters are specified at V_{DD} = 5.0 V supply voltage at 25°C, unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Units
Digital Pressure Output $^{@}P_{MIN}$	$DOUT_{MIN}$		10		%Full Scale
Digital Pressure Output $^{@}$ P_{MAX}	$DOUT_{MAX}$		90		%Full Scale
Digital Full-Scale Span	DFS		80		%Full Scale
Resolution			24		Bits
Digital Output Total Error Band	DACC	-1		+1	%FS
Analog Pressure Output $^{@}P_{MIN}$	$AOUT_{MIN}$		10		%VDD
Analog Pressure Output $^{@}$ P_{MAX}	$AOUT_{MAX}$		90		%VDD
Analog Full-Scale Span	AFS		80		%VDD
Analog Output Total Error Band	AACC	-1		+1	%Full Scale
Temperature accuracy	TACC		1		°C

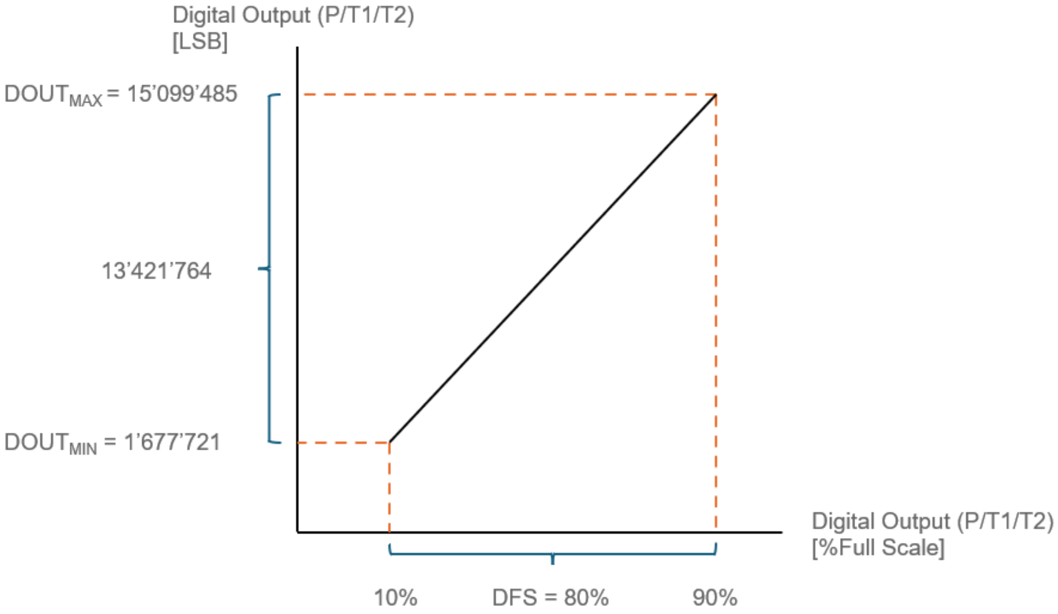
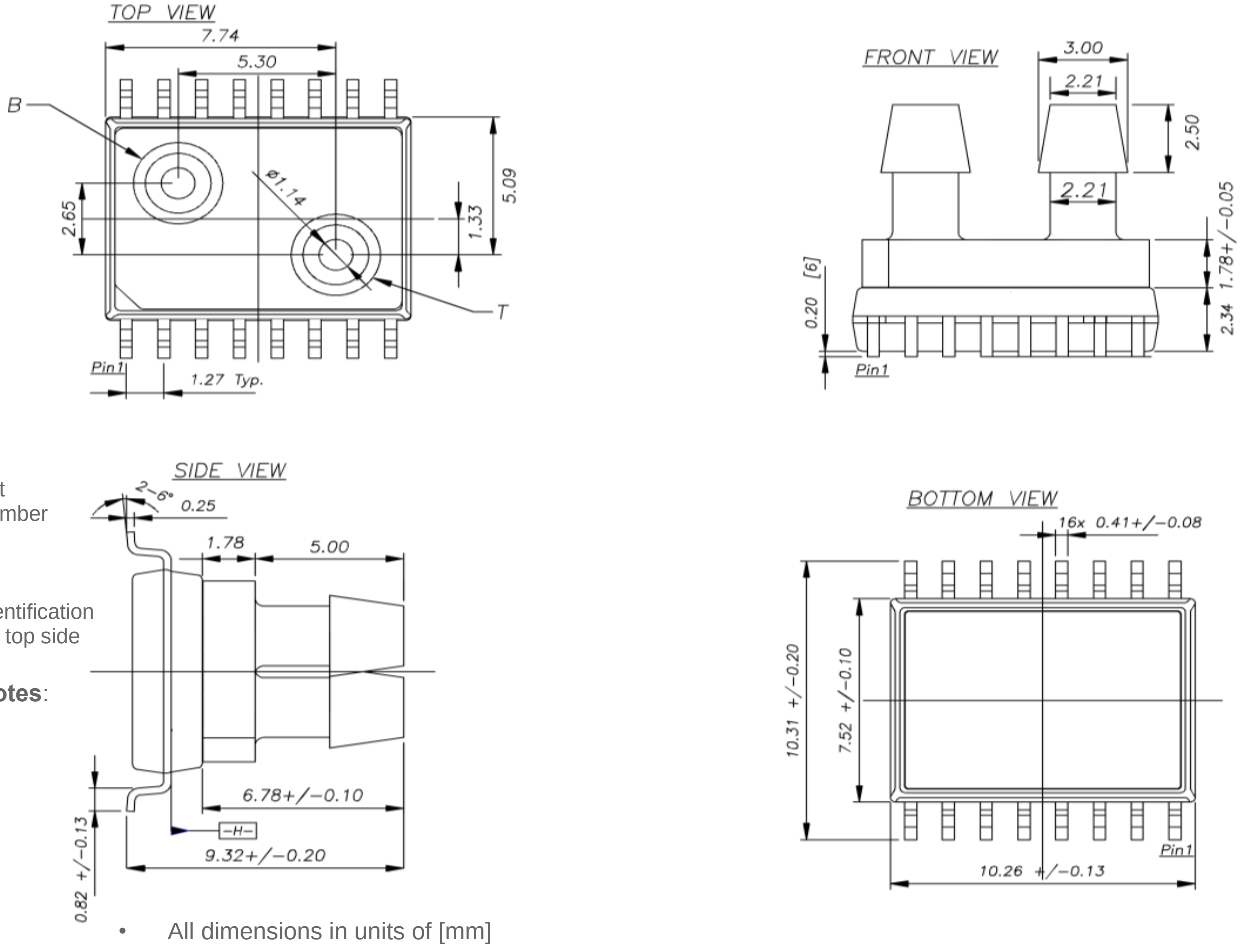


Figure 1: Digital Output / %FS Output

2 PACKAGE DIMENSIONS

SOIC-16 Dual Vertical port (C) Package Dimensions



Lot number

identification on top side

Notes:

- All dimensions in units of [mm]
- Moisture Sensitivity Level (MSL): Level 3
- Wetted materials: Silicon, glass, gold, aluminum, copper, silicone, epoxy, mold compound.
- [B] is tube connected to bottom side of sensor die.
- [T] is tube connected to top side of sensor die. Topside pressure is positive pressure. An increase in topside pressure will result in an increase in sensor output.
- Bottom plate is anodized lid.
- Robust JEDEC SOIC-16 package for automated assembly
- Manufactured according to ISO9001, ISO14001 and ISO/TS 16949 standards

2.1 Pinout functions

Dual port

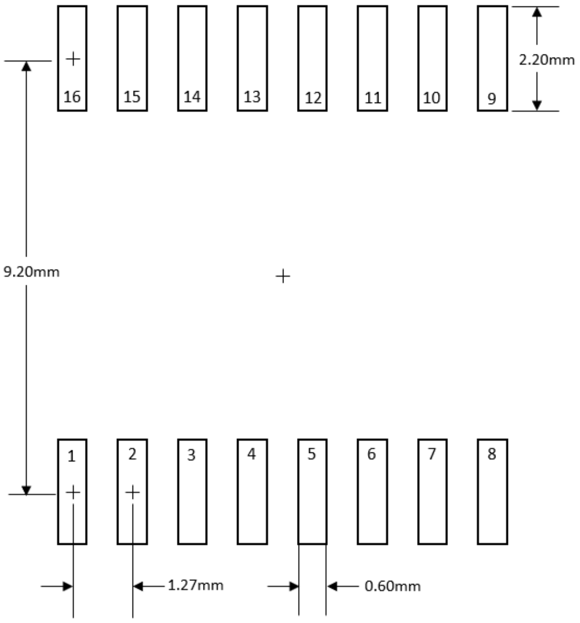
	Dual port
Pin No	**Pin Function**
1	Aout
2	-
3	-
4	-
5	-
6	-
7	SDO
8	-
9	-
10	SDA
11	SCL
12	VSS
13	-
14	-
15	-
16	VDD

Notes:

- SDO : Refer to chapter 7.

2.2 PCB design guidelines

Below suggested PCB footprint is recommended to ensure high mount assembly yields.



Following PCB finishes are compatible with SO16 package :

- Hot Air Solder Leveled (HASL)
- Organic Solderability Protectant (OSP)
- Electroless Nickel Immersion Gold (ENIG)
- Immersion Sn and Immersion Ag.

3 REFLOW PROFILE

The actual profile parameters depend upon the solder paste used. The recommendations from paste manufacturers should be followed.

Below recommendations may be used as alternative solution.

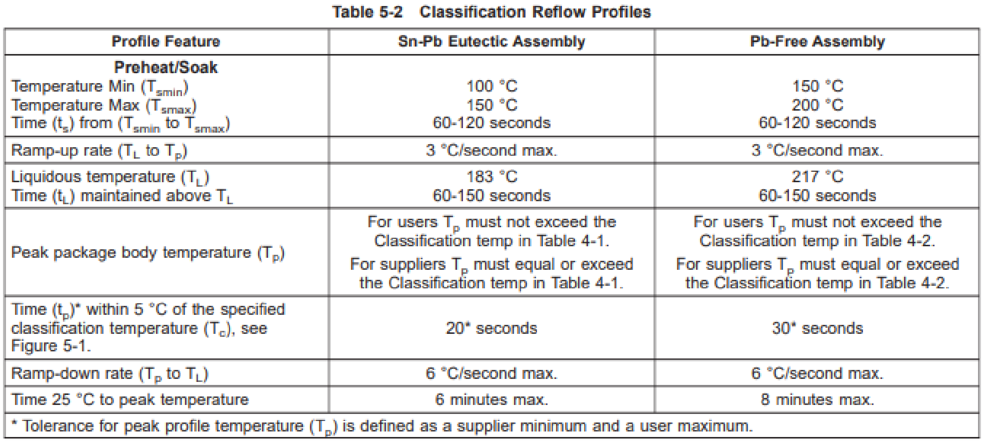
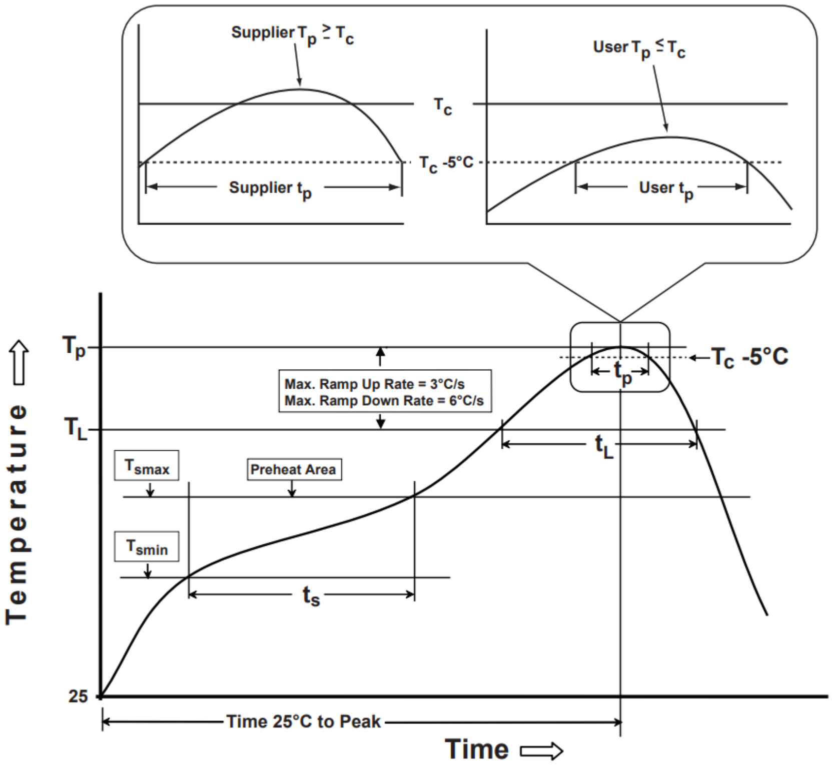


Table 5-2 Classification Reflow Profiles

Profile Feature	Sn-Pb Eutectic Assembly	Pb-Free Assembly
Preheat/Soak Temperature Min (T_{smin}) Temperature Max (T_{smax}) Time (t_s) from (T_{smin} to T_{smax})	100 °C 150 °C 60-120 seconds	150 °C 200 °C 60-120 seconds
Ramp-up rate (T_L to T_p)	3 °C/second max.	3 °C/second max.
Liquidous temperature (T_L) Time (t_L) maintained above T_L	183 °C 60-150 seconds	217 °C 60-150 seconds
Peak package body temperature (T_p)	For users T_p must not exceed the Classification temp in Table 4-1. For suppliers T_p must equal or exceed the Classification temp in Table 4-1.	For users T_p must not exceed the Classification temp in Table 4-2. For suppliers T_p must equal or exceed the Classification temp in Table 4-2.
Time (t_p)* within 5 °C of the specified classification temperature (T_c), see Figure 5-1.	20* seconds	30* seconds
Ramp-down rate (T_p to T_L)	6 °C/second max.	6 °C/second max.
Time 25 °C to peak temperature	6 minutes max.	8 minutes max.

* Tolerance for peak profile temperature (T_p) is defined as a supplier minimum and a user maximum.

1.Classification reflow profile



2.Classification profile

4 FUNCTIONAL BLOCK

4.1 Memory mapping

The TPS Medium Pressure sensor implements 2 major types of memory:

1. A few-time-programmable (FTP) non-volatile memory (NVM) of 32 words.
2. Registers implemented as Flip-Flops.

4.1.1 Non-Volatile Memory Mapping (NVM)

The TPS Medium Pressure uses a memory IP that is a few-time-programmable (FTP) nonvolatile memory (NVM). This NVM is used to keep configuration data while not powered.

The user can access to the NVM address 0 to 31 as shown in the Table 1: User memoryTable 1

NVM can be directly accessed via the interface commands *Erase/Write/Read NVM*.

4.1.2 Register

Typically, registers are physically instantiated as Flip-Flops. Access to registers is direct via the interface commands *Write/Read* register.

The following convention is used:

- RO : Read in normal mode. Write in Unlock mode
- RW : Read/Write in normal mode
- RW* : Read/Write special behavior
- R: Read only

The memory's integrity is verified directly on the register through a CRC check. This verification can be independently activated for each page using the "en_crc_px" flag.

If a CRC error is detected, the "crc_reg" flag in the "int_0" register will toggle to "1". However, this error does not halt the conversion processes.

If "reload" is set to "1", one or two additional rounds of CRC computation will occur. Those additional attempts are made before triggering the conversion process.

4.1.3 User Memory

User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	PRES_4	97	61							p_ratio[2:0]			p_filt[2:0]			p_osr[3:0]			
RW	TEMP1_4	98	62							t1_ratio[2:0]			t1_filt[2:0]			t1_osr[3:0]			
RW	TEMP2_4	99	63							t2_ratio[2:0]			t2_filt[2:0]			t2_osr[3:0]			
RW	OPER	100	64	interrupt_mode	fast_start	int_cont_mode	en_switch	sel_t	fifo_interrupt_threshold[4:0]					fifo_mode[1:0]		delay[3:0]			
RW	DAC_2	101	65		dac_offset_comp_off	en_diag_ana	dac_clip_h[2:0]			dac_clip_l[2:0]			dac_t	drv_vdd_sel[1:0]		dac_buff_mode	dac_mode[2:0]		
RW	POST_GAIN_P	102	66	pgpp[15:0]															
RW	POST_GAIN_T1	103	67	t1gpp[15:0]															
RW	POST_GAIN_T2	104	68	t2gpp[15:0]															
RW	Post_OFF_P	105	69	popp[15:0]															
RW	Post_OFF_T1	106	6A	t1opp[15:0]															
RW	Post_OFF_T2	107	6B	t2opp[15:0]															
RW	COM	108	6C								i2c_daisy_on	en_spike_filter		I2C programmable address[6:1]					
RW	LIMIT_1	109	6D	p_thresh[15:00]															
RW	LIMIT_2	110	6E	p_thresl[15:00]															
RW	LIMIT_3	111	6F	t1_thresh[15:00]															
RW	LIMIT_4	112	70	t1_thresl[15:00]															
RW	LIMIT_5	113	71	t2_thresh[15:00]															
RW	LIMIT_6	114	72	t2_thresl[15:00]															
RW	INT_EN_0	115	73		general_int_en	en_crc_com	en_crc_reg		en_adc_error	en_calc_error	en_sensor_error	en_phigh	en_plow	en_thigh	en_tlow	en_ffull	en_fempty	en_fthr	en_adc_done
RW	INT_EN_1	116	74					en_adc_udfl_t2	en_adc_udfl_t1	en_adc_udfl_p	en_adc_ovfl_t2	en_adc_ovfl_t1	en_adc_ovfl_p	en_calc_udfl_t2	en_calc_udfl_t1	en_calc_udfl_p	en_calc_ovfl_t2	en_calc_ovfl_t1	en_calc_ovfl_p
RW	INT_EN_2	117	75	en_sensor_chk[15:0]															
RW	RESERVED_1	118	76																
RW	RESERVED_2	119	77																
RW	RESERVED_3	120	78								space for customer data								
RW	RESERVED_4	121	79																
RW	RESERVED_5	122	7A																
RW	RESERVED_6	123	7B																
RW	RESERVED_7	124	7C																
RW	RESERVED_8	125	7D																
RW	EN_CRC_4	126	7E																en_crc_mem
RW	CRC_4	127	7F									crc_mem							

Table 1: User memory

Word	Name	Description
PRES_4	p_ratio[2:0]	Enable chopper for pressure measurment
PRES_4	p_filt[2:0]	Pressure filtering in autonomous mode
PRES_4	p_osr[3:0]	Pressure oversampling
TEMP1_4	t1_ratio[2:0]	Enable chopper for temperature 1 measurment
TEMP1_4	t1_filt[2:0]	Temperature 1 filtering in autonomous mode
TEMP1_4	t1_osr[3:0]	Temperature 1 oversampling
TEMP2_4	t2_ratio[2:0]	Enable chopper for temperature 2 measurment
TEMP2_4	t2_filt[2:0]	Temperature 2 filtering in autonomous mode
TEMP2_4	t2_osr[3:0]	Temperature 2 oversampling
OPER	interrupt_mode	interrupt mode behavior, interrupt_mode = 0 collecting by OR function, interrupt_mode = 1 update mode
OPER	fast_start	Fast startup mode (keep biasing on & keep the oscillator powered). 0:Bias OFF; 1: Bias ON
OPER	int_cont_mode	Define the behavior of the interrupt pin while using thresholds detections. 0: Interrupt mode; 1: Continious mode
OPER	en_switch	Enable switch operation (switch with hysteresis). 0: Switch OFF; 1: Switch ON
OPER	sel_t	Selected output temperature (T1 or T2). 0: T1 selected; 1: T2 selected
OPER	fifo_interrupt_threshold[4:0]	Triggers the interrupt if n measurements are ready in the FIFO. 0: OFF; 1:threshold=1…31: Theshold=31
OPER	fifo_mode[1:0]	FIFO operation 0: off; 1: stop at FIFO full; 2,3: overwrite at FIFO full
OPER	delay[3:0]	Delay between measurements used for the autonomous mode. 0:OFF (digital mode); Programmable between 0 to 60s
DAC_2	dac_offset_comp_off	Offset compensation of the output buffer
DAC_2	en_diag_ana	Enable diagnostic on analog ouput, might be redundant from the enable value in the register
DAC_2	dac_clip_h[2:0]	DAC upper voltage limit = 0.65 vdd + dac(2:0) x 0.05 vdd
DAC_2	dac_clip_l[2:0]	DAC lower voltage limit = dac(2:0) 0.05 x vdd
DAC_2	dac_t	Switching between temperature and pressure output to the DAC; 0 = pressure, 1 = temperature
DAC_2	drv_vdd_sel[1:0]	Program regulated vdd in case of high voltage supply
DAC_2	dac_buff_mode	Switch between analog buffer and current buffer. 0: dac buffer for analog modes, 1: dac buffer for current loop application
DAC_2	dac_mode[2:0]	Define the output behavior of the dac (ratiometric, 0-5V absolute, current loop)
POST_GAIN_P	pgpp[15:0]	Post calibration gain for pressure (value added after the digital filter)
POST_GAIN_T1	t1gpp[15:0]	Post calibration gain for Temperature 1 (value added after the digital filter)
POST_GAIN_T2	t2gpp[15:0]	Post calibration gain for Temperature 2 (value added after the digital filter)
Post_OFF_P	popp[15:0]	Post calibration offset for pressure (value added after the digital filter)
Post_OFF_T1	t1opp[15:0]	Post calibration offset for Temperature 1 (value added after the digital filter)
Post_OFF_T2	t2opp[15:0]	Post calibration offset for Temperature 2 (value added after the digital filter)
COM	i2c_daisy_on	Enable i2c daisy chain mode
COM	en_spike_filter	Enable I2C internal 50ns spike filter
COM	I2C programmable address[6:1]	Optional I2C address bits 6:1; bit 0, the LSB is "1"
LIMIT	p_thresh[15:00]	High limit for pressure detection
LIMIT	p_thresl[15:00]	Low limit for pressure detection
LIMIT	t1_thresh[15:00]	High limit for temperature 1 detection
LIMIT	t1_thresl[15:00]	Low limit for temperature 1 detection
LIMIT	t2_thresh[15:00]	High limit for temperature 2 detection
LIMIT	t2_thresl[15:00]	Low limit for temperature 2 detection
INT_EN	general_int_en	If general_int_en is set to "0", all interrupt will be masked
INT_EN	en_crc_com	Enable CRC check during communication
INT_EN	en_crc_reg	Enable CRC check on the full memory
INT_EN	en_adc_error	Global enable of ADC errors.
INT_EN	en_calc_error	Global enable of calculation errors.
INT_EN	en_sensor_error	Global enable of sensor errors.
INT_EN	en_phigh	Enable detection of pressure higher than the higher threshold
INT_EN	en_plow	Enable detection of pressure lower than the lower threshold
INT_EN	en_thigh	Enable detection of temperature higher than the higher threshold
INT_EN	en_tlow	Enable detection of temperature lower than the lower Threshold
INT_EN	en_ffull	Enable detection of FIFO containing 32 unread values
INT_EN	en_fempty	Enable detection of FIFO when all values are read back
INT_EN	en_fthr	Enable detection of FIFO containing n unread values
INT_EN	en_adc_done	Enable detection of a finished conversion
INT_EN	en_adc_udfl_t2	Enable ADC underflow check for temperature 2
INT_EN	en_adc_udfl_t1	Enable ADC underflow check for temperature 1
INT_EN	en_adc_udfl_p	Enable ADC underflow check for pressure
INT_EN	en_adc_ovfl_t2	Enable ADC overflow check for temperature 2
INT_EN	en_adc_ovfl_t1	Enable ADC overflow check for temperature 1
INT_EN	en_adc_ovfl_p	Enable ADC overflow check for pressure
INT_EN	en_calc_udfl_t2	Enable calculation underflow check for temperature 2
INT_EN	en_calc_udfl_t1	Enable calculation underflow check for temperature 1
INT_EN	en_calc_udfl_p	Enable calculation underflow check for pressure
INT_EN	en_calc_ovfl_t2	Enable calculation overflow check for temperature 2
INT_EN	en_calc_ovfl_t1	Enable calculation overflow check for temperature 1
INT_EN	en_calc_ovfl_p	Enable calculation overflow check for pressure
INT_EN	en_sensor_chk[15:0]	Enable sensor check mask
CUST	space for customer data	Customer memory space
CRC	en_crc_mem	Enable CRC on User memory
CRC	crc_memory	CRC of memory

Table 2: Memory content and default values

4.1.4 Operating Register (ALL BIT SET IN NVM BY USER)

Operating register can be accessed through the Read/Write REG commands, as well as with Read/Write-Operating commands. In the NVM via the Read/Write NVM commands.

The operating register does set various operating modes like the FIFO and the delay between the automatic measurements.

When the **delay** register is set (not off), the user needs to enter the command start automatic mode in order to enable the automatic measurement according to the setup in the configuration register. The delay means the delay between one measurement finished and the next measurement starting.

Page	User	Word name	Addr		Data [15:0]															
			Dec	Hex	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
4	RW	OPER	100	64	interrupt_mode	fast_start	int_cont_mode	en_switch	sel_t	fifo_interrupt_threshold[4:0]					fifo_mode[1:0]		delay[3:0]			

Table 3: OPER register

Word	Name	Description
OPER	interrupt_mode	interrupt mode behavior, interrupt_mode = 0 collecting by OR function, interrupt_mode = 1 update mode
OPER	fast_start	Fast startup mode (keep biasing on & keep the oscillator powered). 0:Bias OFF; 1: Bias ON
OPER	int_cont_mode	Define the behavior of the interrupt pin while using thresholds detections. 0: Interrupt mode; 1: Continious mode
OPER	en_switch	Enable switch operation (switch with hysteresis). 0: Switch OFF; 1: Switch ON
OPER	sel_t	Selected output temperature (T1 or T2). 0: T1 selected; 1: T2 selected
OPER	fifo_interrupt_threshold[4:0]	Triggers the interrupt if n measurements are ready in the FIFO. 0: OFF; 1:threshold=1…31: Theshold=31
OPER	fifo_mode[1:0]	FIFO operation 0: off; 1: stop at FIFO full; 2,3: overwrite at FIFO full
OPER	delay[3:0]	Delay between measurements used for the autonomous mode. 0:OFF (digital mode); Programmable between 0 to 60s

Table 4: Operating register definition

“sel_t” is needed to select which temperature we output on the analog interface (when used) and which temperature will be used to populate the FIFO. The main reason is to reduce the size of the FIFO. Even if temperature 2 is selected to populate the FIFO, T1 is always used to compensate the pressure.

During the automatic measurement all user commands are accepted, except the ‘conversion’, all write commands and all NVM actions.

The automatic mode can be stopped by a stop automatic mode command. In case there’s an ongoing conversion running the same time, the automatic triggering of new conversions is stopped but the already started conversions will finish regularly and the ratio counter will be reset. This always means a complete set of requested measurements (P, T1, T2) will be done.

delay[3:0]				**Delay**
bit[3]	bit[2]	bit[1]	bit[0]	
0	0	0	0	off
0	0	0	1	0 ms
0	0	1	0	1 ms
0	0	1	1	5 ms
0	1	0	0	10 ms
0	1	0	1	20 ms
0	1	1	0	50 ms
0	1	1	1	0.1 sec
1	0	0	0	0.2 sec
1	0	0	1	0.5 sec
1	0	1	0	1 sec
1	0	1	1	2 sec
1	1	0	0	5 sec
1	1	0	1	10 sec
1	1	1	0	20 sec
1	1	1	1	60 sec

Table 5: Delay

As soon as the FIFO mode is ‘ON’ the FIFO starts the operation according to the mode set. To reset the FIFO, it must be switched off.

The FIFO interrupt threshold can be chosen between 1.31 and initiates an interrupt as soon as the threshold number of samples is reached. If the threshold is off the interrupt is not activated.

4.1.5 Configuration Register

In the NVM, there are 3 distinct registers respectively for Pressure, Temperature 1 and Temperature 2. These registers can be accessed with Read/Write REG commands, as well as Read/Write Config commands. Value in the NVM is accessed through Read/Write NVM commands.

User	Word name	Register		Data [15:0]															
		Addr																	
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	PRES_4	97	61					p_resol[1:0]		p_ratio[2:0]			p_filt[2:0]			p_osr[3:0]			
RW	TEMP1_4	98	62					t1_resol[1:0]		t1_ratio[2:0]			t1_filt[2:0]			t1_osr[3:0]			
RW	TEMP2_4	99	63					t2_resol[1:0]		t2_ratio[2:0]			t2_filt[2:0]			t2_osr[3:0]			

Table 6: Configuration register

Word	Name	Description
PRES_4	p_resol[1:0]	Number of bit sent during ADC read command for pressure
PRES_4	p_ratio[2:0]	Enable chopper for pressure measurment
PRES_4	p_filt[2:0]	Pressure filtering in autonomous mode
PRES_4	p_osr[3:0]	Pressure oversampling
TEMP1_4	t1_resol[1:0]	Number of bit sent during ADC read command for T1
TEMP1_4	t1_ratio[2:0]	Enable chopper for temperature 1 measurment
TEMP1_4	t1_filt[2:0]	Temperature 1 filtering in autonomous mode
TEMP1_4	t1_osr[3:0]	Temperature 1 oversampling
TEMP2_4	t2_resol[1:0]	Number of bit sent during ADC read command for T2
TEMP2_4	t2_ratio[2:0]	Enable chopper for temperature 2 measurment
TEMP2_4	t2_filt[2:0]	Temperature 2 filtering in autonomous mode
TEMP2_4	t2_osr[3:0]	Temperature 2 oversampling

Table 7: Configuration register definition

The **ratio** is used to select different conversion ratios of temperature and pressure. The delay for automatic conversion is set in the operating register.

ratio[2:0]			**Ratio**
bit[2]	bit[1]	bit[0]	**P/T1/T2**
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	64

Table 8 ratio between pressure and temperatures in automatic mode.

In case of FIFO update or FIFO full mode, the latest temperature is always copied to always have a pair of measurements in the FIFO.

In the phase where the timer triggers an event, but no conversion is scheduled due to all ratios ≥2.

One full measurement cycle corresponds to the acquisition (in this order) of T2, T1 and P.

A ratio of “1” indicates that you perform 1 measurement over 1 cycle. A ratio of “2”, indicates that you perform 1 measurement over 2 cycles.

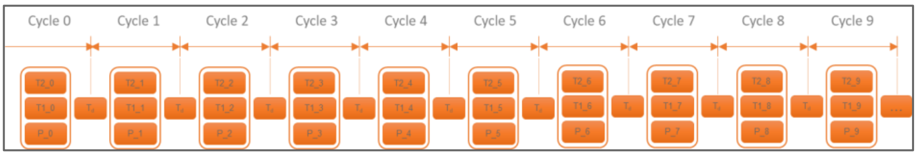


Figure 2 Measurement cycles

Figure 2 shows the full measurement cycles (ADC acquisitions) when p_ratio=1, t1_ratio=1 & t2_ratio=1. The time between two measurement cycles (T_d) is defined by delay[3:0].

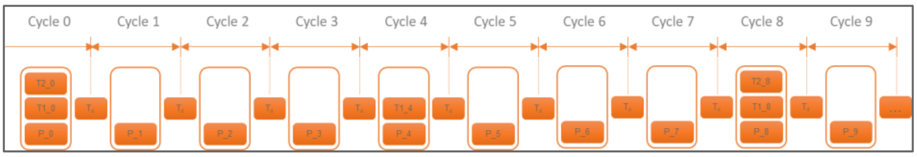


Figure 3 Measurement cycles

Figure 3 shows the full measurement cycles (ADC acquisitions) when p_ratio=1, t1_ratio=4 & t2_ratio=8.

The **filter** calculates an IIR average with the programmable coefficient k :

$$adc_{mean(n)} = adc_{mean(n-1)} - \frac{adc_{mean(n-1)}}{k} + \frac{adc}{k}$$

Which yields in a transfer function of:

$$H(z) = \frac{1}{k} * \frac{1}{1 - z^{-1} * \frac{k-1}{k}}$$

The delay between the samples is defined by the update rate.

If the filter is switched on, the already available ADC value is used as the start value adc_mean. This can be either the previous adc_mean before the filter was turned off, or the value of the last ADC conversion done.

To restart the filter, switch it off, do a new conversion and switch it on again.

The settling time for a big jump to reach 90% of the final value is 2.2 x k samples.

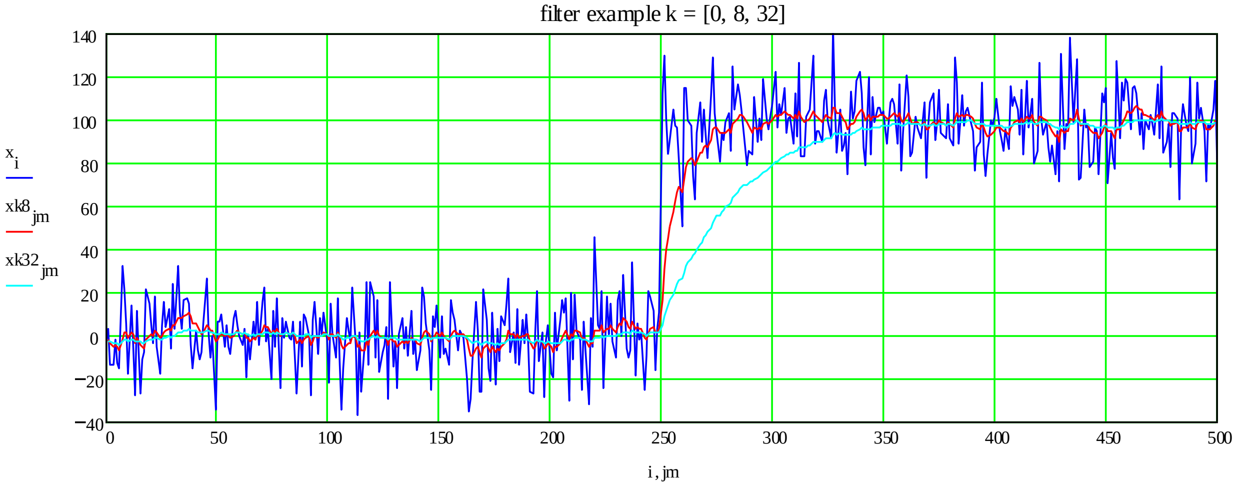


Figure 4: Filter behavior of a noise signal jumping from 0 to 100 using k=0.8 and 32

The table here below shows the reduction factor on the rms noise:

filt[2:0]			IIR Filter	noise_reduction factor
bit[2]	bit[1]	bit[0]		
0	0	0	off	1.00 +/- 0.00
0	0	1	2	1.73 +/- 0.07
0	1	0	4	2.65 +/- 0.21
0	1	1	8	3.87 +/- 0.50
1	0	0	16	5.54 +/- 1.00
1	0	1	32	8.20 +/- 1.20
1	1	0	64	11.95 +/- 2.00
1	1	1	128	15.82 +/- 2.70

Table 9 IIR filter.

The **OSR** defines the speed and noise of the ADC.

OSR[3:0]				Over sampling
bit[3]	bit[2]	bit[1]	bit[0]	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	9
1	0	1	1	9
1	1	0	0	9
1	1	0	1	9
1	1	1	0	9
1	1	1	1	9

Table 10: Over sampling ration (accuracy of the conversion)

During a conversion running or in the automatic mode, the ‘write config’ command is not accepted.

4.1.6 COM Register

The COM register, only contain the parameters for communication. The register is accessible with Read/Write REG commands. Value in the NVM is accessed through Read/Write REG commands.

User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	COM	108	6C								i2c_daisy_on	en_spike_filter		I2C programmable address[6:1]					

Table 11: COM register

Word	Name	Description
COM	i2c_daisy_on	Enable i2c daisy chain mode
COM	en_spike_filter	Enable I2C internal 50ns spike filter
COM	I2C programmable address[6:1]	Optional I2C address bits 6:1; bit 0, the LSB is "1"

Table 12: COM register definition

4.2 DAC

An on-chip 14-bit DAC is provided by the ASIC to generate analog voltage output. The 14-bit resolution spans from 0% to 100% of the DAC reference voltage, while the linear range is assured within the 5% to 95% range.

The reference voltage can be configured in two ways:

1. It can be directly linked to the supply VDD when operating in ratiometric mode.
2. It can be connected to the output of an internal regulator, which may provide either 3V or 5V. In both scenarios, VDD must exceed the targeted regulated voltage by at least 200mV.

The choice between these two options is made automatically during the configuration of dac_mode[2:0]:

dac_mode[2:0]			DAC Mode
bit[2]	bit[1]	bit[0]	
0	0	0	DAC OFF
1	0	0	Ratiometric analog output
1	0	1	Absolute 3V output
1	1	0	Absolute 5V output
1	1	1	Current loop mode

word

Table 13: dac_mode[2:0]

The output voltage is expressed as:

$$DAC\_VOUT = DAC\_CODE \cdot \frac{DAC\_VREF}{2^{14} - 1}$$

Where:

1. DAC_VREF can be either VDD, 3V or 5V depending on dac_mode[2:0]
2. DAC_CODE[13:0] represents either the pressure value when dac_t=”0” or the temperature value when dac_t=”1”. The selection between Temperature 1 or 2 depends on sel_t flag. DAC_CODE[13:0] corresponds to the MSB of the compensated data (p_comp[24:10], t1_comp[24:10] or t2_comp[24:10]) without any diagnostic or error flag insertion. In case of interrupt, diagnostic or error flag, AOUT will be forced either to 0V or DAC_VREF
3. DAC_CODE can also be raw data in case p_raw, t1_raw or t2_raw are set to 1
4. DAC_CODE is clipped according to dac_clip_h[2:0] & dac_clip_l[2:0] stored into the memory (see here below)

The user can limit the output voltage. The lower limit can be adjusted from 0% to 35% in 5% increments, while the upper limit can be adjusted from 65% to 100% in 5% increments.

dac_clip_h[2:0]			Output voltage	DAC clipping
bit[2]	bit[1]	bit[0]	clippping high	Value in hex
0	0	0	1.00×Vref	3FFF
0	0	1	0.95×Vref	3CCC
0	1	0	0.90×Vref	3999
0	1	1	0.85×Vref	3666
1	0	0	0.80×Vref	3332
1	0	1	0.75×Vref	2FFF
1	1	0	0.70×Vref	2CCC
1	1	1	0.65×Vref	2999

Table 14: dac_clip_h[2:0]

dac_clip_l[2:0]			Output voltage	DAC clipping
bit[2]	bit[1]	bit[0]	clippping low	Value in hex
0	0	0	0.0×Vref	0000
0	0	1	0.5×Vref	0333
0	1	0	0.10×Vref	0666
0	1	1	0.15×Vref	0999
1	0	0	0.20×Vref	0CCD
1	0	1	0.25×Vref	1000
1	1	0	0.30×Vref	1333
1	1	1	0.35×Vref	1666

Table 15: dac_clip_l[2:0]

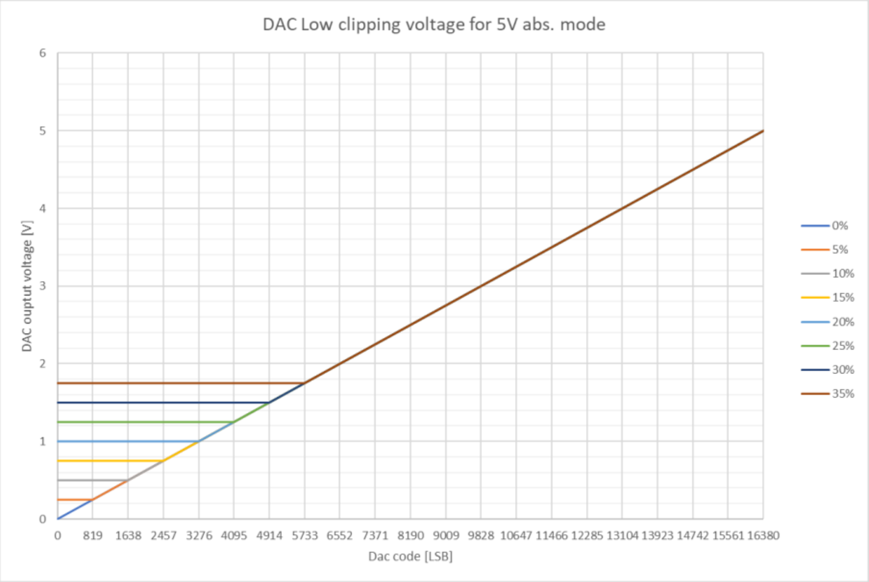


Figure 5: DAC clipping voltage on the low side

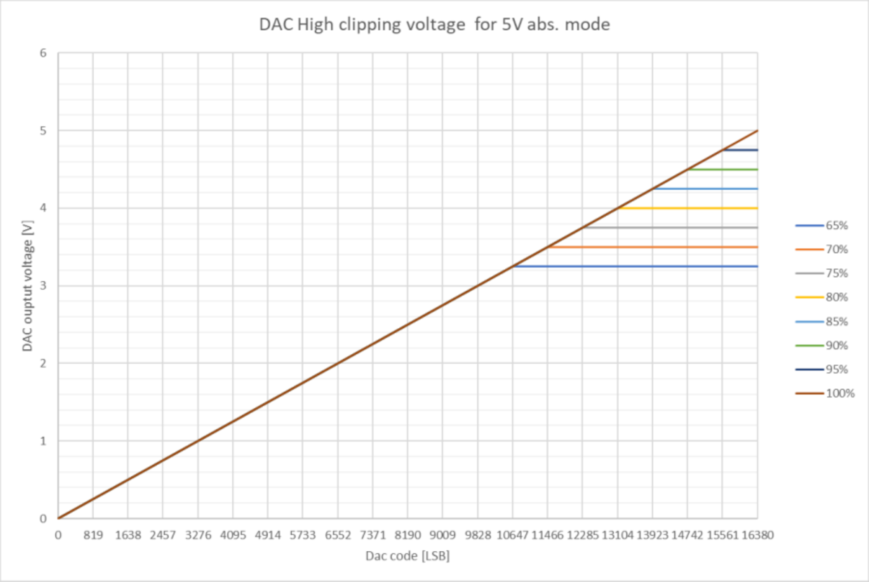


Figure 6: DAC clipping voltage on the high side

en_diag_ana signal (from the memory) can globally disable the error & interrupt signaling on AOUT. By default, en_diag_ana is “0”. If en_diag_ana is set to “1”, the value of the interrupt register can be propagated to AOUT according to the table here below.

Diagnostic or interrupt description	Flag	dac_out [15] signaling high error	dac_out [14] signaling low error
Sensor diagnostic error	sensor_error	0	1
Sensor ADC saturation	adc_error	0	1
Memory CRC	crc_reg	0	1
Sensor computation error	calc_error	1	0
Low threshold detection	plow / tlow	0	1
High threshold detection	thigh / thigh	1	0

Table 16: Error & Interrupt signaling

Sensor diagnostic error has the highest priority, and it decreases until threshold detection that has the lowest priority.

4.2.1 Ratiometric analog output mode

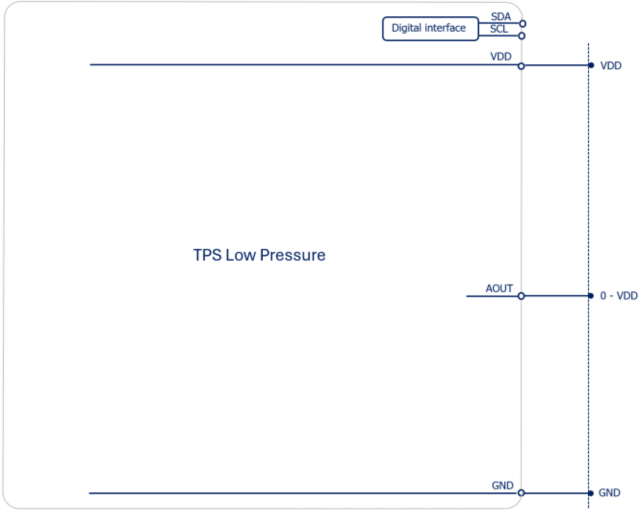


Figure 7: Ratiometric analog output

In ratiometric analog output mode, communication can occur in several ways:

1. Following the standard communication for a three-pin module, AOUT is connected to the SDA pin, as depicted in the image above (Figure 7).
2. Employing all other interfaces.

TPS Medium Pressure can be switched to “analog out” mode under the following conditions:

1. After a restart (Power down – Power up cycle), provided correct configuration in the NVM
2. Upon receiving a reset command, given correct configuration in the NVM.

4.2.2 Absolute 3V analog output mode

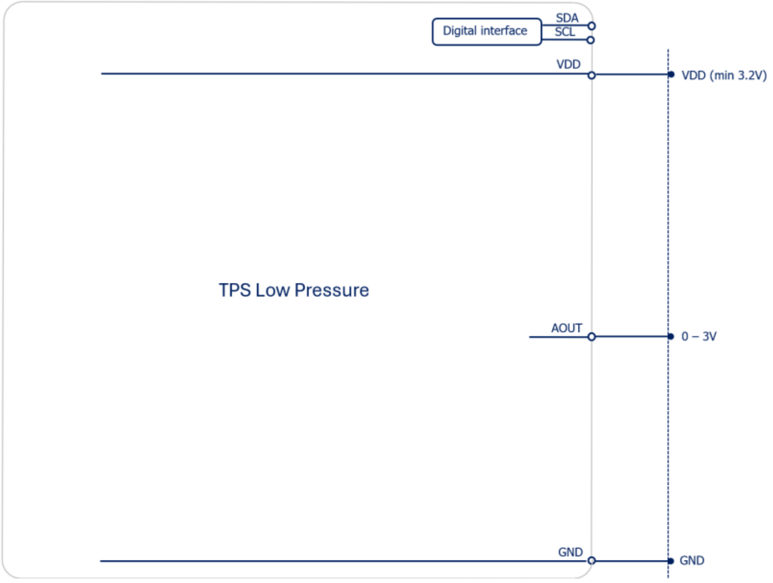


Figure 8: Absolute 3V analog output

The communication interface is set the same way as for Ratiometric analog output.

4.2.3 Absolute 5V analog output mode

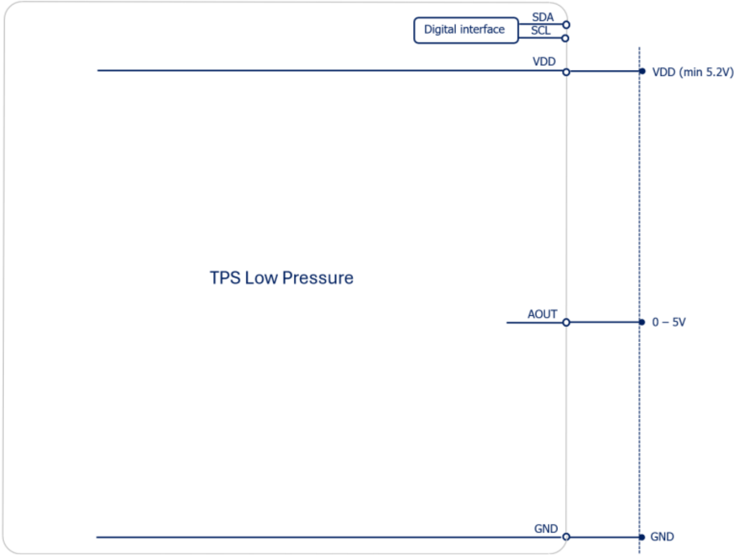


Figure 9: Absolute 5V analog output

The communication interface is set the same way as for Ratiometric analog output.

4.3 Diagnostics

TPS Medium Pressure has internal checks for:

1. Sensor bridge connections & external temperature sensor connections (sensor_error)
2. ADC overflow/underflow (adc_error)
3. Calculation engine saturation (overflow/underflow) (calc_error)
4. Pressure / temperature threshold (p/t high/low)
5. FIFO check (f*)
6. Communication check (crc_com)
7. Memory checks (crc_reg)

The effect of a diagnostic can be observed:

1. On the FIFO value stored (diagnostics are showed by 8 highest values of the communication)
2. By a read register command (direct access to the diagnostic bit)
3. On the analog output. In case of detection the analog voltage is clamped either high or low

Sensor check is performed at the beginning of each conversion. The result is available after the checking period in front of the conversion and stays stable until the next conversion is triggered.

When “interrupt_mode” = 0, the errors are collected and persistent.

Since the diagnostic block does not know which conversion is run, the digital part must take care and collect them by the equivalent of an logical OR function on each group of the sensor_chk[15:0] after each single conversion triggered like:

$$int\_2.sensor\_chk[15:13] \ = \ int\_2.sensor\_chk[15:13]\ OR\ sensor\_chk[15:13]\ (at\ the\ end\ of\ T2\ conversion)$$

$$int\_2.sensor\_chk[12:10] \ = \ int\_2.sensor\_chk[12:10]\ OR\ sensor\_chk[12:10]\ (at\ the\ end\ of\ T1\ conversion)$$

$$int\_2.sensor\_chk[\,9:\ 0] \ = \ int\_2.sensor\_chk[\,9:\ 0]\ OR\ sensor\_chk[\,9:\ 0]\ (at\ the\ end\ of\ P\ conversion)$$

With that, we can insert sensor_error diagnostic code (0xFF_FF9) into proper FIFO data slice

When “interrupt_mode” = 1, each group must be updated after the conversion.
This way an analog sensor can recover from an error without any external action needed

$$int\_2.sensor\_chk[15:13]\ sensor\_chk[15:13]\ (at\ the\ end\ of\ T2\ conversion)$$

$$int\_2.sensor\_chk[12:10] \ = sensor\_chk[12:10]\ (at\ the\ end\ of\ T1\ conversion)$$

$$int\_2.sensor\_chk[\,9:\ 0] \ = \ sensor\_chk[\,9:\ 0]\ (at\ the\ end\ of\ P\ conversion)$$

The basic behavior of the sensor_check is shown below:

NVM		User	Word name	Register		Data [15:0]															
Page	Addr			Addr																	
	dec			Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
4	19	RW	INT_EN_0	115	73	Reserved_0	general_int_en	en_crc_com	en_crc_reg	Reserved_0	en_adc_error	en_calc_error	en_sensor_error	en_phigh	en_plow	en_thigh	en_tlow	en_ffull	en_fempty	en_fthr	en_adc_done
4	20	RW	INT_EN_1	116	74	Reserved_0				en_adc_udfl_t2	en_adc_udfl_t1	en_adc_udfl_p	en_adc_ovfl_t2	en_adc_ovfl_t1	en_adc_ovfl_p	en_calc_udfl_t2	en_calc_udfl_t1	en_calc_udfl_p	en_calc_ovfl_t2	en_calc_ovfl_t1	en_calc_ovfl_p
4	21	RW	INT_EN_2	117	75	en_sensor_chk[15:0]															

Table 17: Sensor check enable registers

NVM		User	Word name	Register		Data [15:0]															
Page	Addr			Addr																	
	dec			Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
						T2			T1			P									
		RW*	INT_2	130	82	text open	sgnd2 open text short svdd	svdd open text short sgnd	text open	sgnd2 open text short svdd	svdd open text short sgnd			inp open	inn open	inp short svdd	inn short svdd	inp short sgnd	inn short sgnd	sgnd open	svdd open

Table 18: Sensor check interrupt registers

4.4 FIFO Operation

The FIFO can be enabled by setting “FIFO mode” in “OPER” register:

User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	OPER	100	64	interrupt_mode	fast_start	int_cont_mode	en_switch	sel_t	fifo_interrupt_threshold[4:0]					fifo_mode[1:0]		delay[3:0]			

Table 19: OPER register

fifo_mode[1:0]		Mode
bit[1]	bit[0]	
0	0	off
0	1	full mode
1	0	update mode
1	1	update mode

Table 20: FIFO mode

Once activated, FIFO registers have the capacity to hold a maximum of 32 sets of temperature and pressure data.

The temperature stored is determined by the "sel_t" parameter: if "sel_t" equals 0, T1 is utilized, otherwise, T2 is used.

If neither pressure nor temperature measurements are enabled (t2/t1/p_off=1), the FIFO remains empty. If at least one of them is On, the FIFO gets updated, but the stored value for Off measurements will come from the last valid prior measurement.

The FIFO operates on a first-in, first-out (FIFO) basis, meaning the first data entered is the first to be outputted, as illustrated in figure here below. Configuration of the operation is based on the relevant bit settings in the "OPER" register.

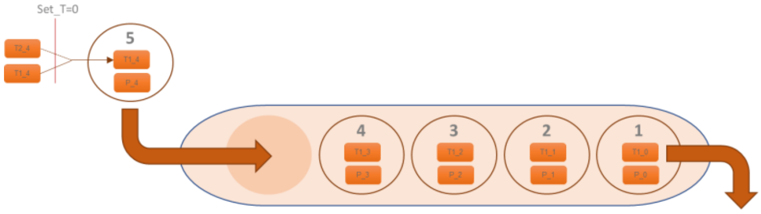


Figure 10: FIFO principle

The data stored in the FIFO depends on the “raw” flag and the configuration of the IIR filter.

If "raw" equals "1", raw data is stored in the FIFO. The 24-ADC bits are stored in the FIFO and may be replaced by an error code in case of bridge diagnostic errors, ADC overflow, or underflow, as outlined in Table 30. These diagnostic errors can be disabled using diagnostic masks.

If "raw" equals "0", compensated data is stored in the FIFO. Compensated data uses a 22-bit format. Hence, if the IIR filter is turned off, two LSBs equal to "00" will be appended to the 22-bit MSB. With filtering enabled, we aim for better resolution. Consequently, the compensated and filtered value will be stored in 24 bits. The compensated data can also be replaced by diagnostics if they are enabled in the diagnostics mask register.

In FIFO mode "OFF", both pointers (FPTRRD & FPTRWR) are aligned and set to "0", ", and error coding insertion on Read ADC data is disabled

Two FIFO modes are available in TPS Medium Pressure:

1. In **FIFO mode 1**, also known as **FIFO full mode**, when the FIFO reaches its capacity, any incoming values from the ADC are discarded, meaning they are not written into any FIFO register.

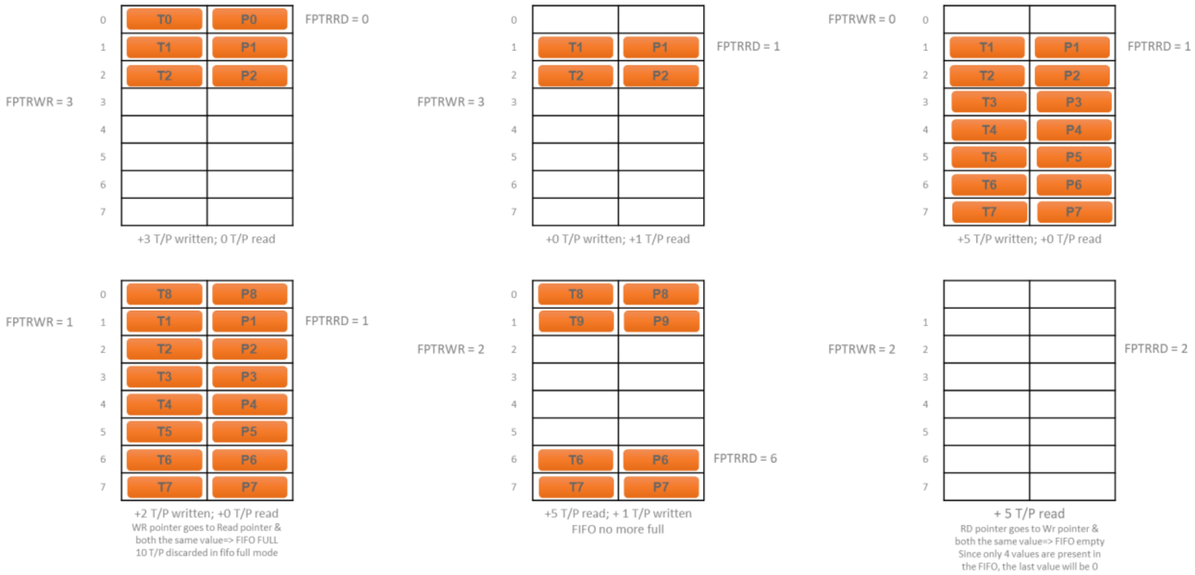
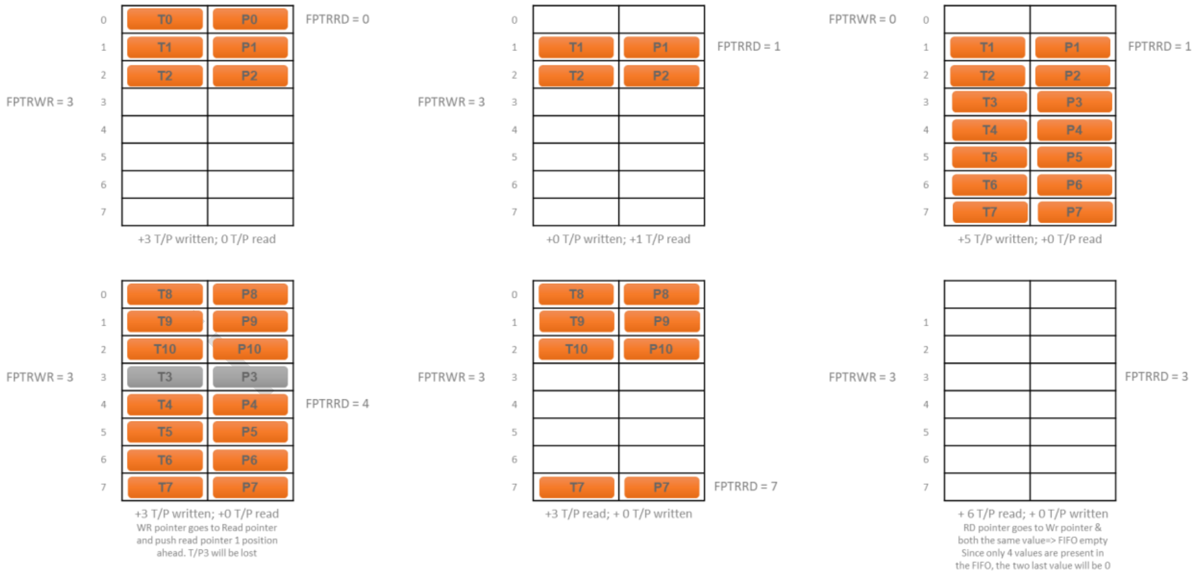

FPTRWR = 3
FPTRRD = 0
+3 T/P written; 0 T/P read
FPTRWR = 3
FPTRRD = 1
+0 T/P written; +1 T/P read
FPTRWR = 0
FPTRRD = 1
+5 T/P written; +0 T/P read
FPTRWR = 1
FPTRRD = 1
+2 T/P written; +0 T/P read
WR pointer goes to Read pointer &
both the same value=> FIFO FULL
10 T/P discarded in fifo full mode
FPTRWR = 2
FPTRRD = 6
+5 T/P read; + 1 T/P written
FIFO no more full
FPTRWR = 2
FPTRRD = 2
+ 5 T/P read
RD pointer goes to Wr pointer &
both the same value=> FIFO empty
Since only 4 values are present in
the FIFO, the last value will be 0

Figure 11. FIFO full mode operation

2. In **FIFO mode 2**, referred to as **FIFO update mode**, when the FIFO reaches its capacity, not only does the write cursor progress forward, but the read cursor also advances to ensure it remains at least one position ahead of the write cursor.


FPTRWR = 3
FPTRRD = 0
+3 T/P written; 0 T/P read
FPTRWR = 3
FPTRRD = 1
+0 T/P written; +1 T/P read
FPTRWR = 0
FPTRRD = 1
+5 T/P written; +0 T/P read
FPTRWR = 3
FPTRRD = 4
+3 T/P written; +0 T/P read
WR pointer goes to Read pointer
and push read pointer 1 position
ahead. T/P3 will be lost
FPTRWR = 3
FPTRRD = 7
+3 T/P read; + 0 T/P written
FPTRWR = 3
FPTRRD = 3
+ 6 T/P read; + 0 T/P written
RD pointer goes to Wr pointer &
both the same value=> FIFO empty
Since only 4 values are present in
the FIFO, the two last value will be 0

The depth of the FIFO can be configured using FIFO threshold. The FIFO threshold can be configured within the operating register (see Chapter: Interrupt & Interrupt mask register). It allows for setting a FIFO threshold ranging from 1 to 31. An interrupt triggered by the FIFO threshold occurs immediately after the FIFO has stored the designated number of data samples.

FIFO_interrupt_threshold[4:0]					Threshold
bit[4]	bit[3]	bit[2]	bit[1]	bit[0]	
0	0	0	0	0	OFF
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
0	0	1	0	1	5
0	0	1	1	0	6
0	0	1	1	1	7
0	1	0	0	0	8
0	1	0	0	1	9
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16
1	0	0	0	1	17
1	0	0	1	0	18
1	0	0	1	1	19
1	0	1	0	0	20
1	0	1	0	1	21
1	0	1	1	0	22
1	0	1	1	1	23
1	1	0	0	0	24
1	1	0	0	1	25
1	1	0	1	0	26
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	0	1	29
1	1	1	1	0	30
1	1	1	1	1	31

Table 21: FIFO Interrupt threshold

For example, if FIFO threshold is equal to 5, the interrupt will occur if the write cursor is 5 data samples ahead of the read cursor and 5 data samples can be read. The FIFO interrupts will be set only at the end of a conversion. The FIFO will not stop at the threshold and fills in the 32 positions until the FIFO is full.

Please refer to the Interrupt Register for other FIFO interrupts behaviors.

4.4.1 Read values from FIFO registers

To extract data from the FIFO, utilize the "read ADC" command. Upon receiving this command, the data at the designated position indicated by the read cursor is accessed and at the same time, the read pointer is incremented. If the reading is incomplete, the data is lost.

The maximum number of readings attainable varies between FIFO full mode and FIFO update mode. In FIFO full mode, waiting until the FIFO registers are filled enables the retrieval of the 32 most recent values. Conversely, in FIFO update mode under identical conditions, only 31 values can be retrieved. This intentional design choice in update mode prevents the write cursor and read cursor from addressing the same FIFO register.

4.4.2 A common application: FIFO combined with automatic mode

Once the automatic mode is activated, access to configuration registers is no longer permitted. Hence, it is crucial to verify that the P, T1, and T2 configurations are correctly configured prior to enabling automatic mode.

Example 1:

In automatic mode (delay[3:0] higher than 0), and with the following configuration,

	p	t1	t2
p/t1/t2_off	0	0	1
p/t1/t2_ratio	1	4	X

Table 22: FIFO example 1 configuration

A pressure conversion is done every cycle and a temperature conversion every 4th cycle. In this case the latest available temperature is copied to the actual position, to always have a pair of measurements in the FIFO.

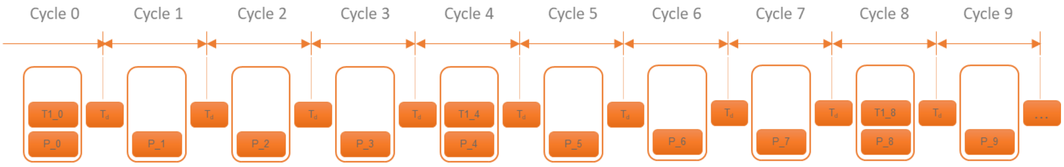
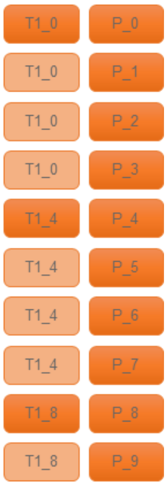


Figure 12: Measurement cycles for example 1

Each cycle time is different depending on the number of action(s) it must perform. The computation engine must be turned on at the end of each cycle where at least one conversion was performed.

Inside the FIFO:



T1_0	P_0
T1_0	P_1
T1_0	P_2
T1_0	P_3
T1_4	P_4
T1_4	P_5
T1_4	P_6
T1_4	P_7
T1_8	P_8
T1_8	P_9

Figure 13: FIFO content for example 1

Example 2:

In automatic mode (delay[3:0] higher than 0), and with the following configuration,

	p	t1	t2
p/t1/t2_off	0	0	1
p/t1/t2_ratio	2	4	X

Table 23: FIFO example 2 configuration

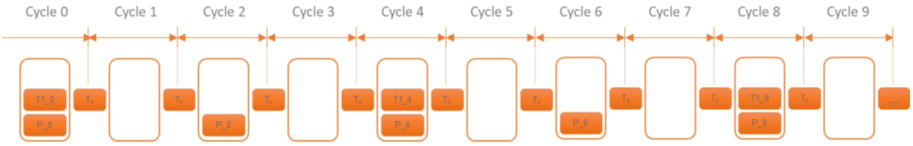
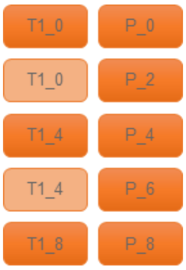


Figure 14: Measurement cycle for example 2

Remark

- For example, cycle 1 is a real short cycle since no conversion is triggered and no DSP compensation is required. The timing will be mostly defined by the two delays.

Inside the FIFO:



T1_0	P_0
T1_0	P_2
T1_4	P_4
T1_4	P_6
T1_8	P_8

Figure 15: FIFO content for example 2

4.5 Interrupt & Interrupt mask register

An interrupt register, labeled as INT_0, is in place to alert the user of various conditions.
These include completion of a conversion, FIFO status (empty/full/threshold), surpassing temperature or pressure limits, sensor diagnostics errors, ADC errors, and computation errors. Within INT_0, errors are grouped together.
If diagnostic errors occur, detailed information can be accessed in the INT_2 register, while all other errors can be found in INT_1.

User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	INT_0	128	80			crc_com	crc_reg		adc_error	calc_error	sensor_error	phigh	plow	thigh	tlow	ffull	fempty	fthr	adc_done
RW	INT_1	129	81					adc_udfl_t2	adc_udfl_t1	adc_udfl_p	adc_ovfl_t2	adc_ovfl_t1	adc_ovfl_p	calc_udfl_t2	calc_udfl_t1	calc_udfl_p	calc_ovfl_t2	calc_ovfl_t1	calc_ovfl_p
RW	INT_2	130	82	sensor_chk[15:0]															

Table 24: Interrupt & Diagnostic registers

The signaling on AOUT can be enabled by setting "1" in the corresponding field of INT_EN_0.
For grouped errors, the corresponding bit of INT_EN_0 must be enabled, and each error can be individually enabled in INT_EN_1 & INT_EN_2. In addition, "general_int_en" must be set to "1".

User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RW	INT_EN_0	115	73		general_int_en	en_crc_com	en_crc_reg		en_adc_error	en_calc_error	en_sensor_error	en_phigh	en_plow	en_thigh	en_tlow	en_ffull	en_fempty	en_fthr	en_adc_done
RW	INT_EN_1	116	74					en_adc_udfl_t2	en_adc_udfl_t1	en_adc_udfl_p	en_adc_ovfl_t2	en_adc_ovfl_t1	en_adc_ovfl_p	en_calc_udfl_t2	en_calc_udfl_t1	en_calc_udfl_p	en_calc_ovfl_t2	en_calc_ovfl_t1	en_calc_ovfl_p
RW	INT_EN_2	117	75	en_sensor_chk[15:0]															

Figure 16: Interrupt & Diagnostic enable bit in the NVM

This will allow the user to handle the interrupt, clear the corresponding flag and enable again the interrupt for further application. Prior to clear the interrupt flag in INT_0, the user must clear the corresponding error in the sub register INT_1 or INT_2.

At startup, all bits of interrupt enable registers are set to "0". The interrupt register can be read and reset by the user. The interrupt bit must be cleared by the user by writing a "1" into the according register bit himself after handling the interrupt.

Interrupt Registers INT_0:

- Can be read either with a ReadREG command or via the Read-Interrupt command
- In normal mode, writing a bit to '1' to INT_0, INT_1 & INT_2 will clear the corresponding bit. As example, write-interrupt '0x0005' will clear the FIFO empty and ADC done interrupt and leave all the others unchanged.
- In service unlock mode, writing these registers will update the entire registers value (true write)

Interrupt Enable Registers INT_EN_0:

- 0 = interrupt bit not seen/enabled at INT
- 1 = interrupt bit seen/enabled at INT

All interrupts (except sensor diagnostics) will be evaluated at the end of a complete T2/T1/P sequence (including compensation algorithm). If t2_off=1, t1_off=0 & P_off=0, t1 and p will be acquired. Only at the end of the whole sequence, temperature 1 will be evaluated against the limits and set the corresponding

if sel_t="1", the T2 limit will be evaluated. But, since t2_off="1", the T2 value used in the limit or switch comparisons will be the last valid. If no T2 conversion was carried out since reset, the comparison will be against the initial T2 value, which is 0.

The sequence looks like:

start T2 not performed -> T1 conversion -> wait end of T1 conversion, start P conversion -> wait end of P conversion, compute compensated T1/P, evaluate interrupt.

Depending on interrupt mode, we have:

1. interrupt_mode = 0. the errors are collected and persistent ('OR function of new and actual state->register')
2. interrupt_mode = 1, the errors are updated and not persistent (actual state ->register)

4.6 Outputs Calculation

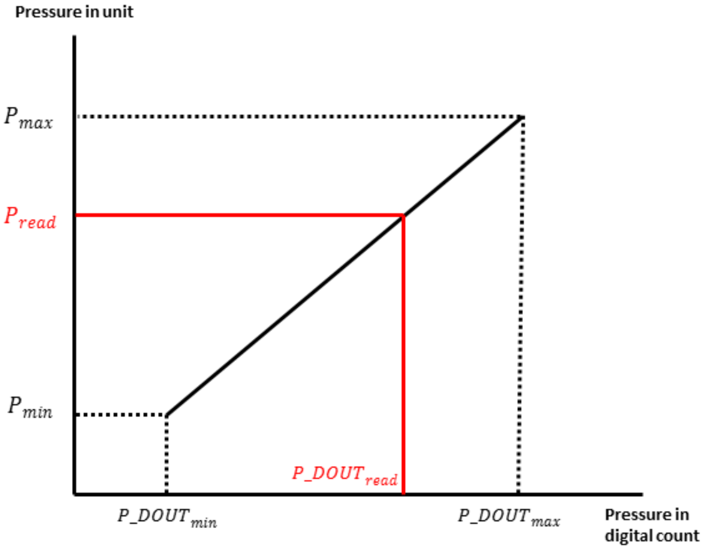
4.6.1 Pressure Calculation

$$P_{read} = \mathrm{P_{min}} + \frac{\mathrm{P\_DOUT_{read}} - \mathrm{P\_DOUT_{min}}}{\mathrm{P\_DOUT_{max}} - \mathrm{P\_DOUT_{min}}}(\mathrm{P_{max}} - \mathrm{P_{min}})$$

$\mathrm{P_{min}}$ and $\mathrm{P_{max}}$ are minimum and maximum rating pressure in specified pressure unit on the specification.

$\mathrm{P\_DOUT_{min}}$ and $\mathrm{P\_DOUT_{max}}$ are minimum and maximum digital counts on the specification.

$\mathrm{P\_DOUT_{read}}$ is digital reading from the output and $\mathrm{P_{read}}$ is the converted pressure output based on $\mathrm{P\_DOUT_{read}}$.



For 20032306-00, calculation is as below:

P_{min} = 0 PSI

P_{max} = 5 PSI

$P_{DOUTmin}$: 1'677'721

$P_{DOUTmax}$: 15'099'485

For $P_{DOUTread}$= 8'388'603, you calculate P_{read} = $0 + \frac{8388603-1677721}{15099485-1677721} * (5-0) = 2.5$ PSI

4.6.2 Temperature Calculation

$T_{DOUTmin}$: 1'677'721

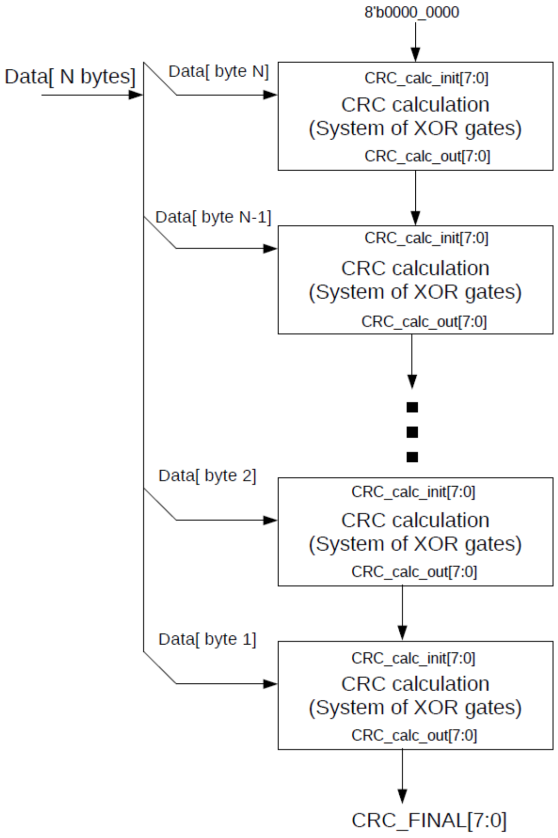
$T_{DOUTmax}$: 15'099'485

$$\mathrm{T_{read}} = -20°\mathrm{C} + \frac{\mathrm{T\_DOUT_{read}} - \mathrm{T\_DOUT_{min}}}{\mathrm{T\_DOUT_{max}} - \mathrm{T\_DOUT_{min}}}(105)$$

5 CHECKSUM CALCULATION (CRC)

CRC is implemented in the digital core to check the integrity of the memory data & for the communication integrity.

CRC is calculated for each 8-bit data, i.e., one byte of data as the minimum data size over which CRC is executed. For N-byte data, calculation will be done in N steps. This can be illustrated as below.



Its properties are displayed in table below.

Property	Value
Width	8 bits
Protected data	Read and/or write data
Polynomial	0x31 (x8 + x5 + x4 + 1)
Initialization	0x00
Reflect input	False
Reflect output	False
Final XOR	0x00
Examples	CRC (0xBEEF) = 0x13

Memory CRC is calculated for each page (if enabled for page) and takes 2x32 master clocks / page :

1. starting with the upper byte of lowest page address (i.e., 0, 32) and ending with the lower byte of the second-highest page address (i.e., 30, 62)
2. lower byte of the highest page address is the CRC result byte & upper byte of the highest page address is ignored

6 APPLICATION CIRCUIT

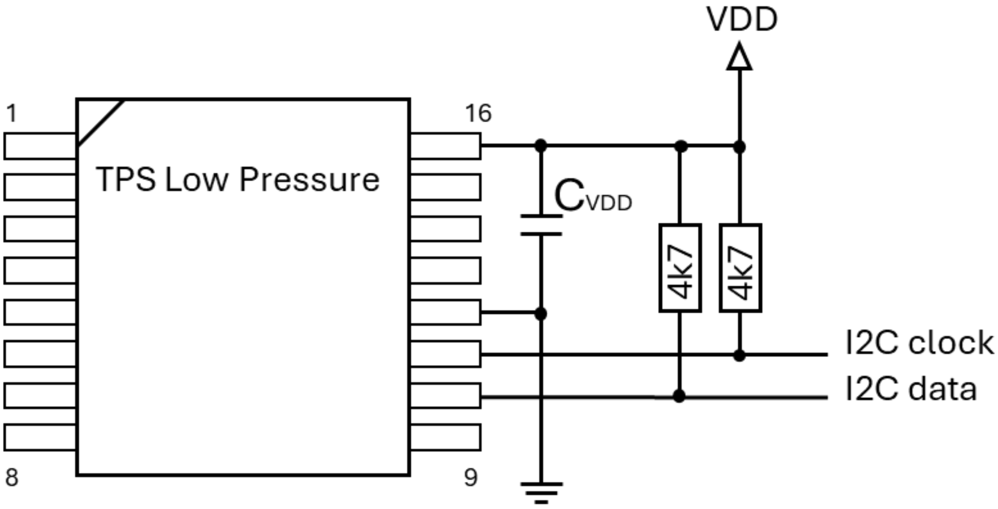


Figure 17: Application circuit Dual-port I2C

7 SERIAL DIGITAL INTERFACE

TPS Medium Pressure sensor has built the serial interface I2C.

There are commands, which trigger an internal action which may take more time than the command itself. This is mainly the case at PON, Reset, Refresh Register or if a Conversion is started. To keep a predictable behavior of the chip during a running conversion or in the automatic mode the configuration cannot be changed (Write Config is not accepted by the digital part). A new conversion cannot be started during a conversion running (Conversion command is not accepted by the digital part during this time).

The internal behavior can any time be monitored on SDO. The serial data output (SDO) is used to monitor the status of TPS Medium Pressure Sensor digital core and to send the results requested by a command. The SDO can indicate various situations, depending on the state of the TPS Medium Pressure Sensor digital core. Here are possible conditions and corresponding states of SDO:

- SDO = “1” – when device is reset and is ready.
- SDO = “0” – whenever a valid command is detected, SDO goes low for one clock cycle.
- SDO = “0” – when device is busy, erase/write NVM, conversion, reset, refresh, POR.
- SDO during power-on-reset: Refer to dedicated chapter on Power on (PON).
- SDO has a different behavior in daisy chain mode application
- SDO output indicates only busy state and command recognition.

7.1 Command Structure

Size of each command is 1 byte (8 bits) as described in Table 25 below.
ADC read command will return 24 bits result of the above requested finished conversion (P, T1, T2).
All commands are the same and are explained in the next table.

User Commands										
Commands	bit [7]	bit [6]	bit [5]	bit [4]	bit [3]	bit [2]	bit [1]	bit [0]	Data MOSI	Data MISO
Read Sensor ID	0	0	0	0	1	0	0	CRC	no data	+ 8 data bytes
Reset	0	0	0	1	0	0	0	CRC	no data	no data
Refresh	0	0	0	1	0	0	1	CRC	no data	no data
Write Operating Mode	0	0	0	1	0	1	0	CRC	+2 data bytes	no data
Read Operating Mode	0	0	0	1	0	1	1	CRC	no data	+2 data bytes
Start automatic mode	0	0	0	1	1	0	0	CRC	no data	no data
Stop automatic mode	0	0	0	1	1	0	1	CRC	no data	no data
Write Config	0	0	1	0	0	A1	A0	CRC	+2 data bytes	no data
Read Config	0	0	1	0	1	A1	A0	CRC	no data	+2 data bytes
Conversion **	0	1	0	0	1	1	1	CRC	no data	no data
Read ADC **	0	1	0	1	T2	T1	P	CRC	no data	Depends config
Write Interrupt Mask	0	1	1	0	0	A1	A0	CRC	+2 data bytes	no data
Read Interrupt Mask	0	1	1	0	1	A1	A0	CRC	no data	+2 data bytes
Write Interrupt Reg	0	1	1	1	0	A1	A0	CRC	+2 data bytes	no data
Read Interrupt Reg	0	1	1	1	1	A1	A0	CRC	no data	+2 data bytes
Write NVM	1	0	0	0	0	1	1	CRC	+1 addr byte + 2 data bytes	no data
Erase NVM	1	0	0	1	0	1	1	CRC	+1 addr byte	no data
Read NVM	1	0	1	0	0	1	1	CRC	+1 addr byte	+2 data bytes
Write Limits	1	1	0	0	A2	A1	A0	CRC	+2 data bytes	no data
Read Limits	1	1	0	1	A2	A1	A0	CRC	no data	+2 data bytes
Write REG***	1	1	1	0	0	0	0	CRC	+1 addr byte + 2 data bytes	no data
Read REG	1	1	1	0	0	0	1	CRC	+1 addr byte	+2 data bytes

Table 25: User commands

** Read ADC with all T2, T1 and P = 0 are invalid commands and will not be accepted by the controller. The command will not be acknowledged in the I2C mode

*** only page 4 can be accessed by the user.

7.2 CRC during communication

CRC 8 is implemented for all commands and can be activated by utilizing the least significant bit (LSB). This CRC-8 functionality is applicable to I2C communication protocol and maintains a consistent structure.

Is an incorrect CRC is detected, the command is not executed, and an interrupt is generated. It’s important to note that in I2C, the command and remaining data continue to be acknowledged even the CRC is wrong and the command is not executed.

Two CRC checks are planned: one for verifying the command’s integrity and another for verifying the integrity of the fata. If a CRC issue arises, either the master or the slave can stop the communication.

7.3 I2C Interface

TPS Medium Pressure sensor acts as an ordinary I2C Slave Device without Master capability. Supports both SDR Mode and HDR Mode.

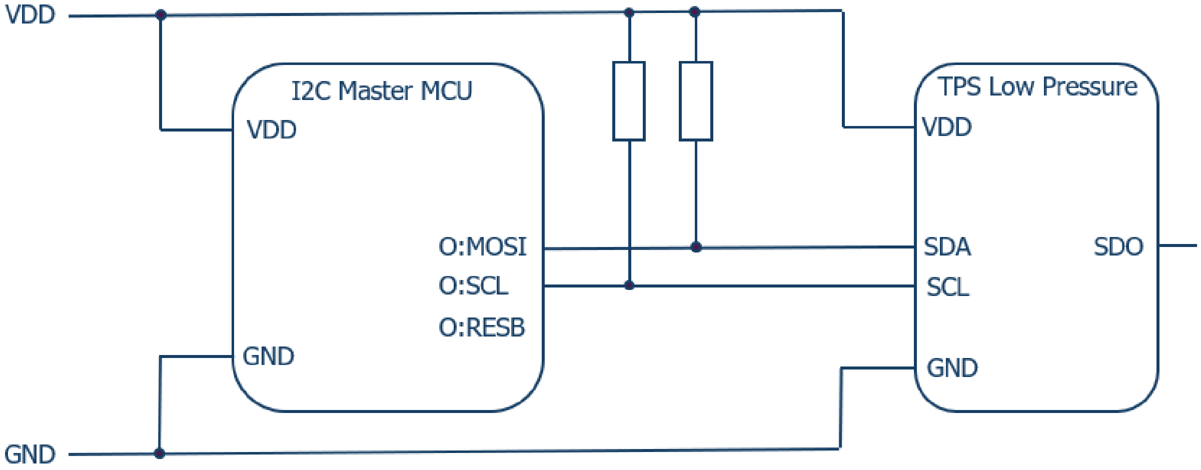


Figure 18: TPS Medium Pressure sensor I2C application circuit

TPS Medium Pressure sensor has a hard coded I2C address following the Table 26.

Dual-Port	Bin	Decimal	Hexadecimal
Standard I2C address	1110’100	116	0x74
Alternative I2C address	0100’010	34	0x22

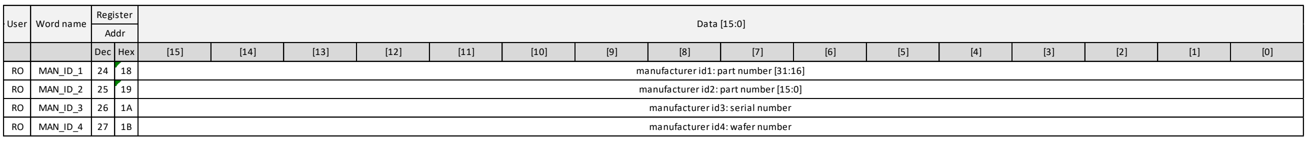
Table 26: Dual-Port I2C address

In I2C mode, I2C command write(R/W=0) with a valid address will always be acknowledged (ACK). However, invalid command won’t be accepted, as described in Table 25. The command acceptance result will be reflected on SDO output.

Detailled Commands

7.3.1 Read sensor ID

The sensor ID is stored into 4 words in the memory and can be directly accessed through this command.



User	Word name	Register Addr		Data [15:0]															
		Dec	Hex	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
RO	MAN_ID_1	24	18	manufacturer id1: part number [31:16]															
RO	MAN_ID_2	25	19	manufacturer id2: part number [15:0]															
RO	MAN_ID_3	26	1A	manufacturer id3: serial number															
RO	MAN_ID_4	27	1B	manufacturer id4: wafer number															

Figure 19: Sensor ID

The part number is coded with one 32 bits word compose of part number LSB and part number MSB (part number [31:16] and part number [15:0]).

So for exemple:

In the two first memory word of read sensor ID, the following values are read:

Read_sensor_ID[0] = 30565_{10} = $0x7765_{16}$
Read_sensor_ID[1] = 55660_{10} = $0xD96C_{16}$

To compute this, the MSB (Read_sensor_ID[0]) should be shifted Left 16 bits which is 2003107840_{10}=$0x77650000_{16}$.

Last step is to add the LSB (Read_sensor_ID[1]) which is 2003163500_{10} and then separate the 2 last digit of the result in decimal to find 20031635-00

7.3.2 Refresh

During a refresh command, the registers values are updated. The data like configuration, operation mode, limits and interrupt enable are not changed.

conditions	min	typ	max	unit
fast_start=1, no crc		84		µs
fast_start=0 adds		4		
crc per page adds		9		
reload 2× per page adds		74		

Table 27: Refresh typical timing

To allow the slave to compute the CRC correctly right before the 8th rising edge, the integrity of the master's data must be guaranteed.

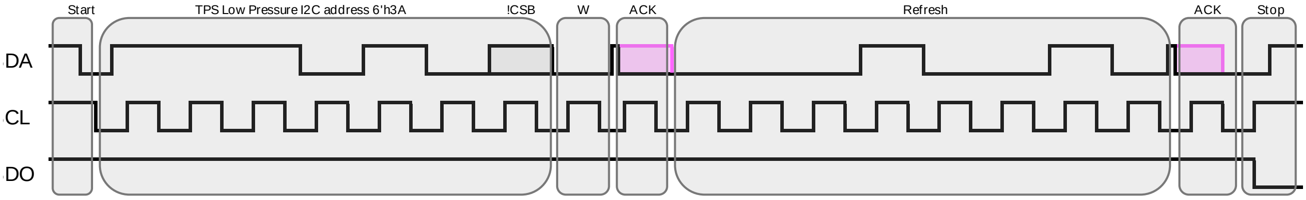


Figure 20: I2C Refresh command

7.3.3 Reset

During a reset the NVM data is transferred to the RAM (see memory mapping for more details).

conditions	min	typ	max	unit
fast_start=1, no crc		112		µs
fast_start=0 adds		4		
crc per page adds		9		
reload 2× per page adds		74		

Table 28: Reset Typical Timing

Using a reset command during an ongoing NVM erase or write operation is not allowed!

Bit 8 of the command is sensitive to outside conditions and not fully covered by the communication CRC check. If bit 8 is corrupted, the following cases can appear:

1. 1 becoming 0 => reset will be executed immediately. The master will get wrong CRC and therefore knows that something went wrong. The chip configuration will be overridden by the content of NVM.
2. 0 becoming 1 => the reset will not be executed since clocks should be given by the master to check CRC. Neither the slave nor the master will know that the reset was not executed.

To allow the slave to compute the CRC before the 8th rising edge, the data from the master must be well setup and guaranteed.

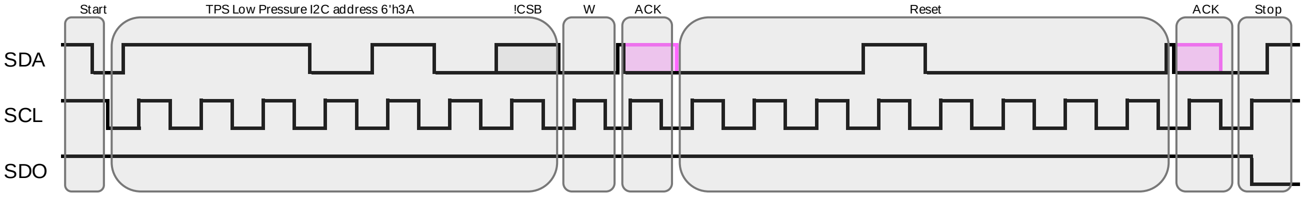


Figure 21: I2C Reset command

7.3.4 Write Config

Write config allows to configure raw/comp data, read resolution, ratio, filter and OSR of each measurement type separately.

This command is not accepted during an ongoing conversion.

Command		Register				
Wr config		Page	User	Word name	addr	
A1	A0				dec	hex
0	0	4	RW	PRES_4	97	61
0	1	4	RW	TEMP1_4	98	62
1	0	4	RW	TEMP2_4	99	63
1	1	-	-	-	-	-

Table 29: Write config command address vs register address

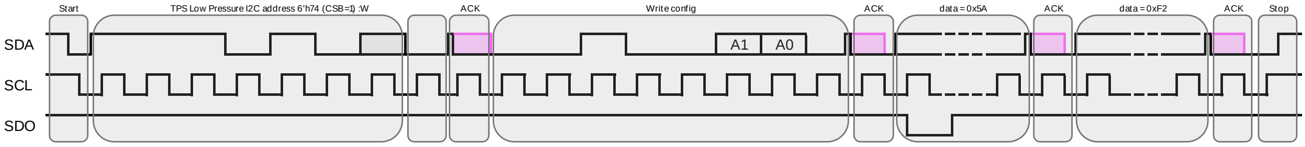


Figure 22: I2C write configuration register command

7.3.5 Read Config

Read config allows to verify the written configuration.

Command		Register				
Rd config		Page	User	Word name	addr	
A1	A0				dec	hex
0	0	4	RW	PRES_4	97	61
0	1	4	RW	TEMP1_4	98	62
1	0	4	RW	TEMP2_4	99	63
1	1	-	-	-	-	-

Table 30: read config command address vs register address

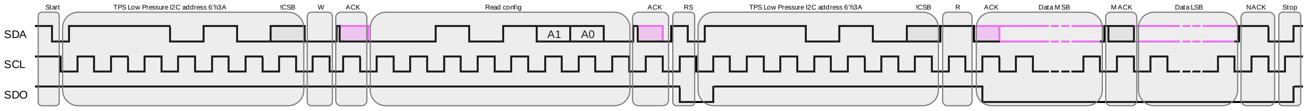


Figure 23: I2C read configuration register command

7.3.6 Start automatic mode

Start automatic mode from the command mode. To correctly start the automatic mode, the delay needs to be set beforehand to something different than 0.

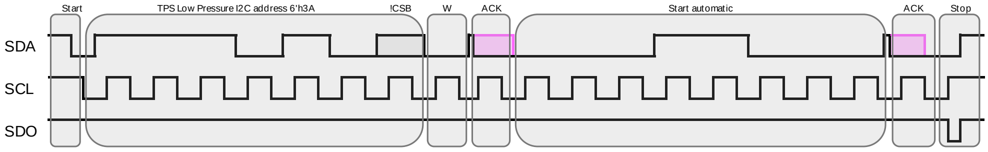


Figure 24: I2C start automatic mode command

7.3.7 Stop automatic mode

Stop automatic mode while TPS Medium Pressure is continuously operating.

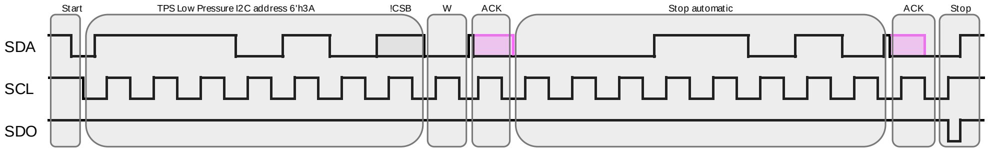


Figure 25: I2C stop automatic mode command

7.3.8 Write operating mode

With the write operating mode command, the behavior if the FIFO and the automatic mode can be controlled.

Data corresponds to the OPER register stored in the registers (address 0x64)

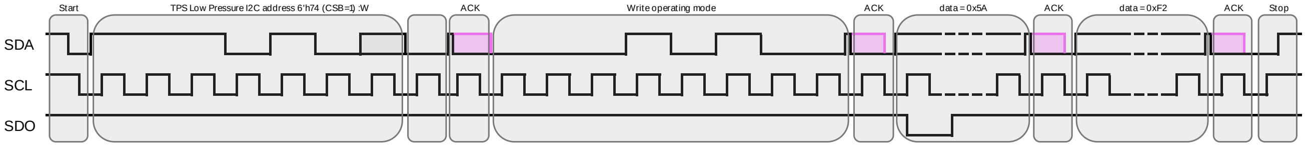


Figure 26: I2C write operating mode command

7.3.9 Write REG

This command allow writing into the register map.The address range for this command is from 96 to 127.

Burst mode is implemented for this command.

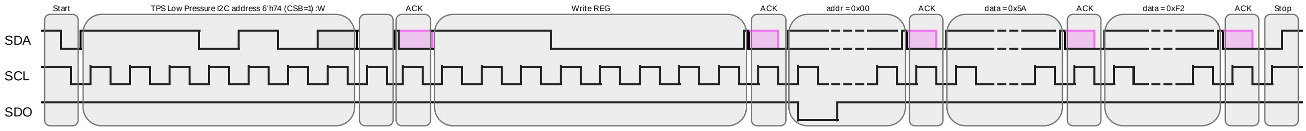


Figure 27: I2C write REG command

In case CRC and burst mode is used, the CRC is sent after every 2 data bytes.

7.3.10 Read REG

This command allows reading the register map. The address range for this command is from 96 to 127.

Burst mode is implemented for this command.

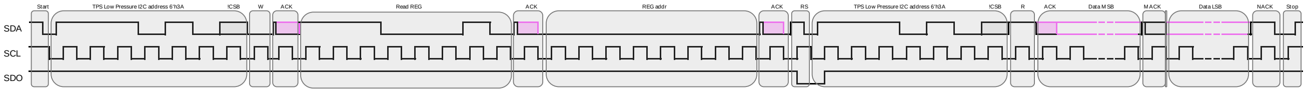


Figure 28: I2C read REG command

With CRC, the data packet will be sent in the following order and still operates in a burst mode.

- Read-out data MSB, Read-out data LSB, CRC, Read-out data MSB, Read-out data LSB, CRC …

7.3.11 Conversion

An ADC conversion is started using a conversion command. After the command is recognized by the chip SDO goes low. SDO goes high again when conversion is completed. The conversions time is depending on the OSR.

The result of the conversion is transferred to the data register after the conversion. It is possible to trigger on the rising edge of SDO to get the time when the operation is finished. This command is not accepted during an ongoing conversion.

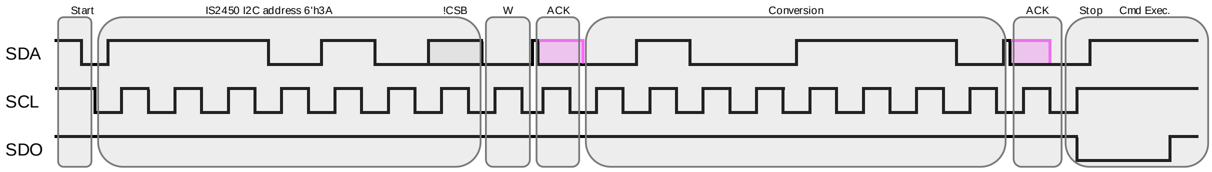


Figure 29: I2C conversion command

7.3.12 Read ADC

The ADC read command retrieves the result of a conversion command, and its behavior is influenced by several configuration bits:

1. The settings of the T2, T1, and P flags within the command itself.
2. The data read length can be configured in the operating register using t2_resol, t1_resol, and p_resol.

After power up and in FIFO "off" mode, attempting to read the ADC without any prior conversion will yield all zeros. However, after a conversion is completed, the last conversion result will be read.

When FIFO is empty, reading will yield all zeros regardless of whether FIFO update mode or FIFO full mode is activated,

The temperature conversions will be executed first, followed by the pressure conversion. As soon as the three conversions are done, the computation engine is triggered. All data are then available and can be read out by a read ADC command.

Reading data while the commanded conversion session is still ongoing will return previous values for all requested data. This precaution is taken to prevent the possibility of returning corrupt data resulting from concurrent write and read operations. The host should only read data after the conversion process has fully completed.

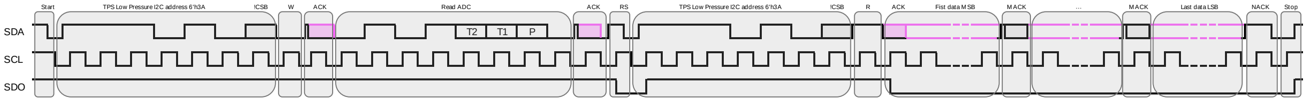


Figure 30: I2C read ADC

7.3.13 Erase NVM

The Erase NVM command deletes the 16-bit data at the selected address. After erasing the 16-bit, the data is undefined and must be programmed to 0 or 1 before a reading can be done on the specific address.

Following command, the chip does autonomously erase the data which takes additional time of 1.5 ms typical to 3 ms maximum where the chip is busy. Is CSB goes high just after the command and data, this time cannot be seen. In this case wait at least >3 ms before the next command is applied.

No burst mode implemented for this command.

Erase NVM address range goes from 0 to 31. All MSBs of the address are ignored.

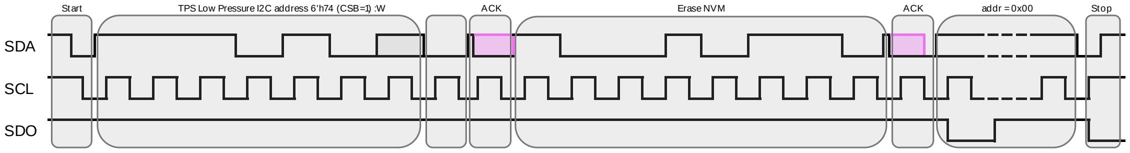


Figure 31: I2C erase NVM command

7.3.14 Write NVM

The write command for NVM consists of two parts. First part sets the chip into the Write NVM-Mode and transfers the data to be written. After that, the chip does autonomously program the data to the NVM which takes additional time of 1.5 ms typical to 3 ms maximum where the chip is busy. If CSB goes high just after the command and data, this time cannot be seen.

No burst mode implemented for this command.

Write NVM address range goes from 0 to 31. All MSBs of the address are ignored.

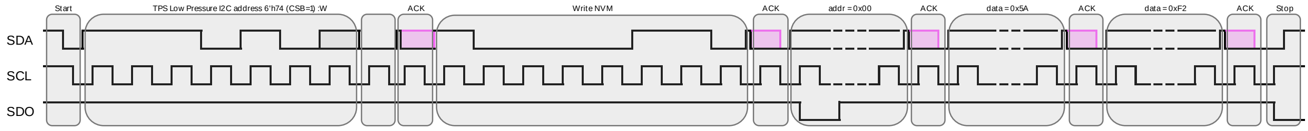


Figure 32: I2C Write NVM command

7.3.15 Read NVM

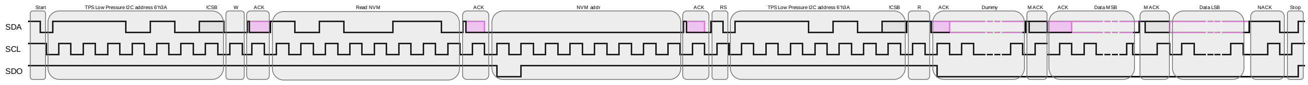
The read NVM command consists of two parts. First Part sets the chip into the Read NVM-Mode. The user must wait just after the first command byte, “0.5 µs + Tosc_on(= 11.3 to 17.6us)“ until the chip has internally transferred the data from the NVM to the read-out register. This is signaled to the user by SDO going high after the command and address has been sent. After SDO going high (data ready) a dummy byte (eight SCL clocks) are needed before the real output data is available on SDO.

No burst mode implemented for this command.

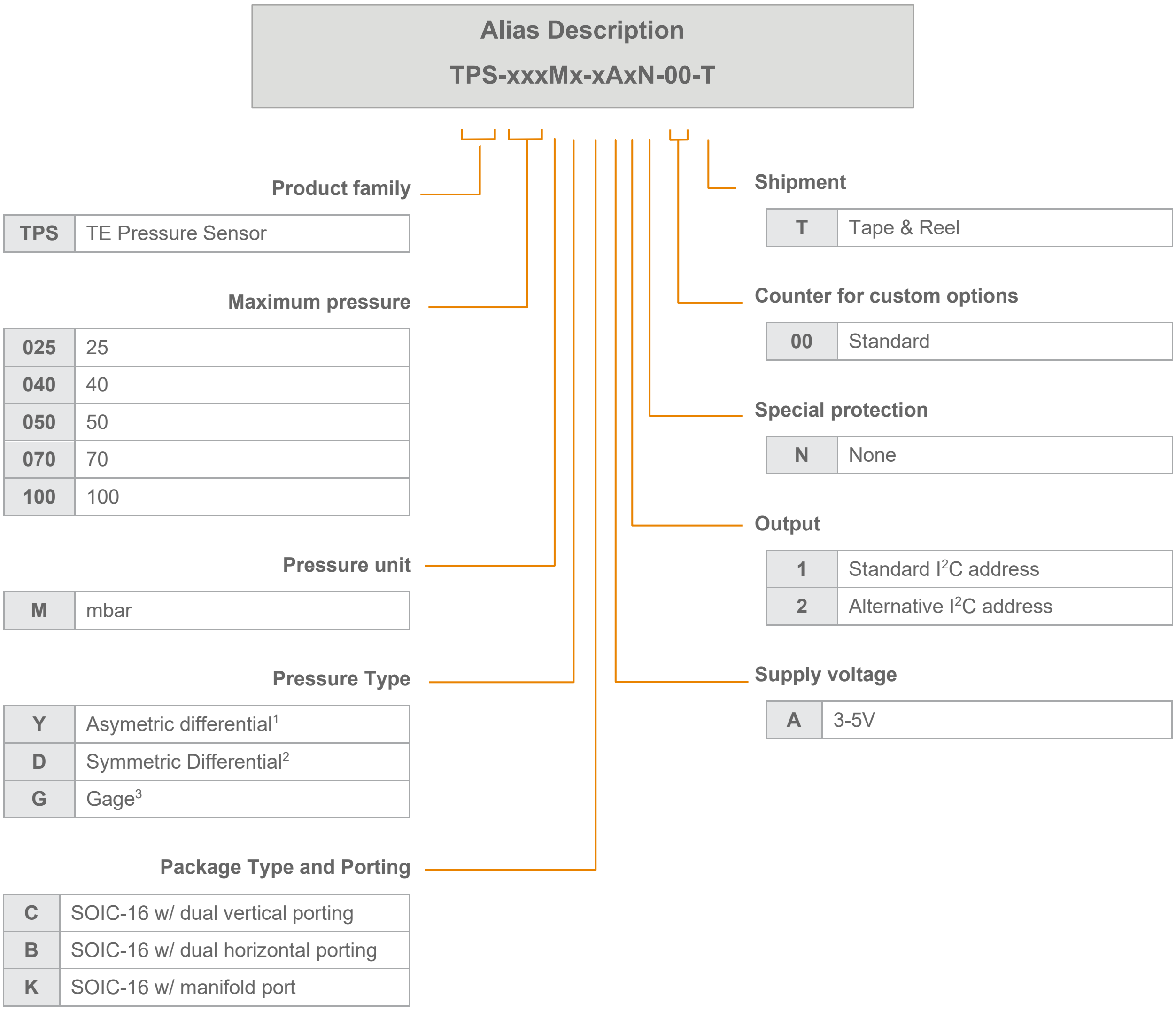
Read NVM address range goes from 0 to 31. All MSBs of the address are ignored.

Prior to any read operation, it is recommended to have the bit fast_start set to 1 in the registers to reduce the latency by removing Tosc_on from the waiting time.

Warning: Performing a read operation after an Erase is invalid as both floating gates of the differential storage have been initialized to the off state



. Alias description



Alias Description

TPS-xxxMx-xAxN-00-T

Product family

TPS	TE Pressure Sensor

Maximum pressure

025	25
040	40
050	50
070	70
100	100

Pressure unit

M	mbar

Pressure Type

Y	Asymetric differential[1]
D	Symmetric Differential[2]
G	Gage[3]

Package Type and Porting

C	SOIC-16 w/ dual vertical porting
B	SOIC-16 w/ dual horizontal porting
K	SOIC-16 w/ manifold port

Shipment

T	Tape & Reel

Counter for custom options

00	Standard

Special protection

N	None

Output

1	Standard I^{2}C address
2	Alternative I^{2}C address

Supply voltage

A	3-5V

Notes :

1. **Minimum pressure is -5 mbar as standard.**
2. **Minimum pressure is negative value of maximum pressure**
3. **Minimum pressure is 0 mbar.**