AVSensors, Multi Chip Technology, Digital Series

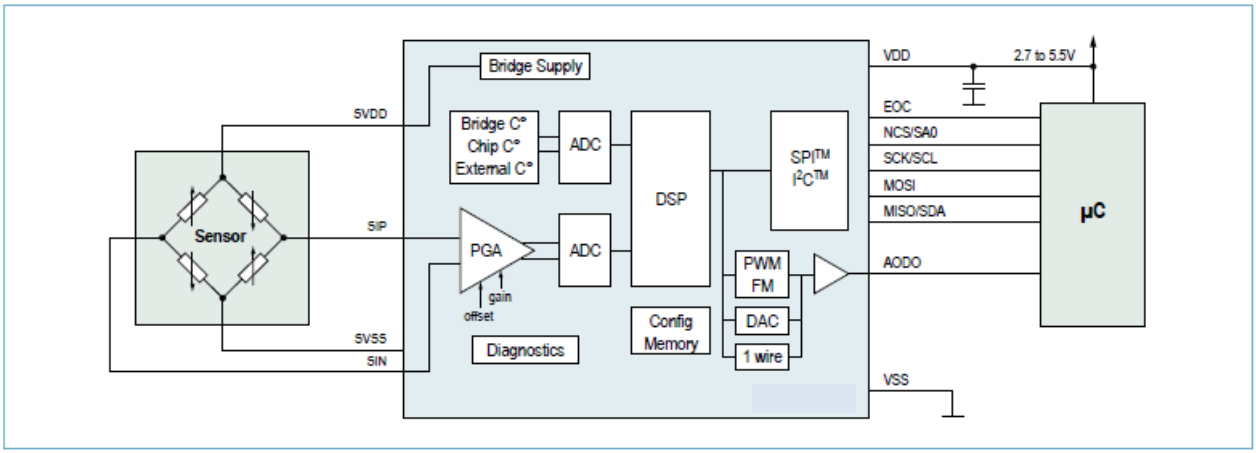
One of AVSensors' key advantages is its use of multi-chip technology, which provides exceptional flexibility in design. By integrating multiple chips into a single sensor package, AVSensors delivers a broad range of digital pressure sensors that balance resolution, performance, and cost to meet the varied requirements of OEM applications.

The 14-bit model delivers 14-bit pressure and 11-bit temperature outputs, operates on either 3.3 V or 5.0 V, and features second-order correction for offset, sensitivity over temperature, and pressure nonlinearity—providing a strong combination of accuracy and affordability for cost-sensitive designs.

The 16-bit version offers full 16-bit resolution for both pressure and temperature with the same dual-voltage flexibility, making it ideal for applications that demand higher precision without a significant cost increase.

AVSensors' 24-bit sensor operates exclusively on 3.3 V and incorporates a 26-bit internal DSP running a second-order correction algorithm, delivering exceptional resolution and stability. While priced higher, it offers outstanding value for mission-critical applications where maximum measurement fidelity and long-term stability justify the investment.

Block Diagram (16 Bit)



This 16 bit mixed-signal ASIC engineered to handle the entire signal chain for resistive bridge pressure sensors, eliminating much of the external circuitry engineers would otherwise need to design. It integrates dual 16-bit ADCs for simultaneous pressure and temperature acquisition, a programmable analog front end with adjustable gain and offset cancellation, and a DSP capable of up to third-order polynomial correction to linearize sensor output. Supply rails from 2.7 V to 5.5 V make it flexible for both 3.3 V and 5.0 V systems, while the architecture's ratiometric design ensures excellent PSRR and measurement stability. Output formats are versatile, offering calibrated data via I^2C or SPI, along with optional ratiometric analog, PWM, or FM outputs. Engineers benefit from the ability to configure bridge excitation, low-pass filter characteristics, and error diagnostics, allowing the ASIC to support a wide range of sensor spans and operating conditions with minimal external components and streamlined system integration

I²C Timing Diagram

Parameter	Condition	Symbol	Min	Typ	Max	Unit
SDA output low voltage	I_{SDA} = 3 mA	$V_{SDA,OL}$	0		0.4	V
Low-to-High transition threshold	pins SA0, SCL	$V_{SDA,LH}$	0.5	0.6	0.7	*VDD
High-to-Low transition threshold	pins SA0, SCL	$V_{SDA,HL}$	0.3	0.4	0.5	*VDD
I²C clock frequency		fSCL	0		400	kHz
Bus free time between a START and STOP condition		tBUSF	1300			ns
Clock low time		tLO	1300			ns
Clock high time		tHI	600			ns
START condition hold time		tSH	100			ns
Data setup time		tSU	100			ns
Data hold time		tH	0			ns
Setup time for repeated START condition		tRSH	600			ns
Setup time for STOP condition		tPSU	600			ns
Rise time of SDA and SCL signals		tR			300	ns
Fall time of SDA and SCL signals		tF			300	ns

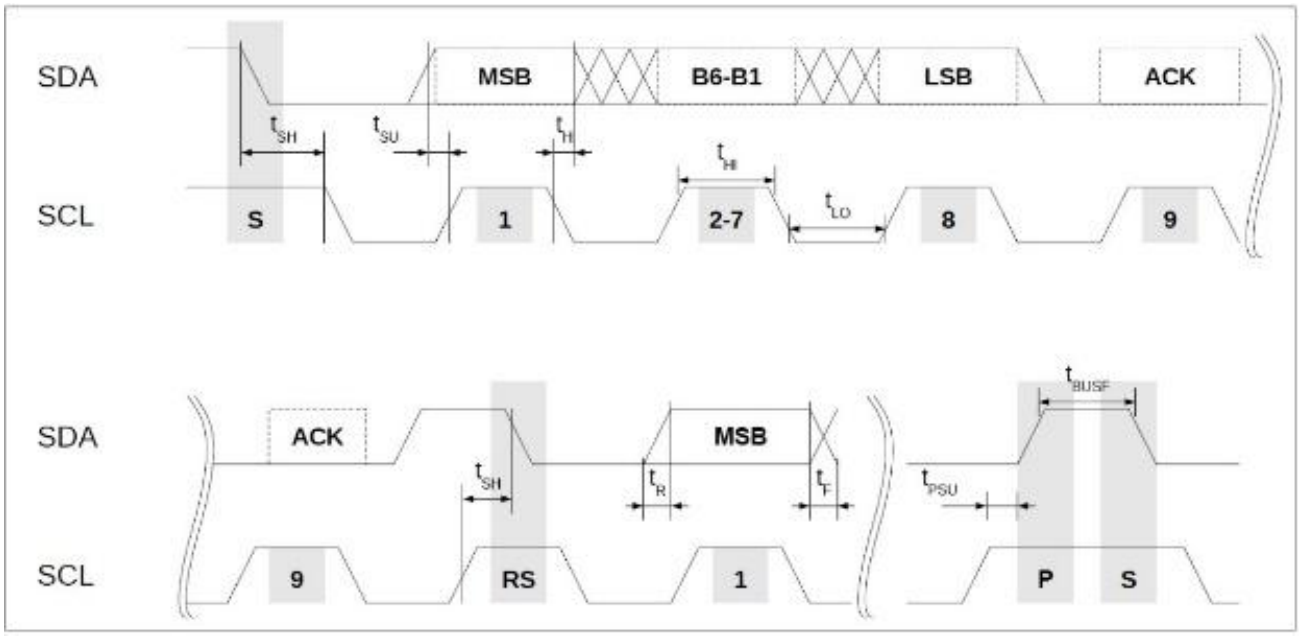


Figure 1. I²C Interface Timing Diagram

I²C Interface

The AVSensors pressure sensor features an I2C slave interface. This interface provides direct access to registers of the memory of the pressure sensor. An external I2C 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.

I²C Command Format

The AVSensors pressure sensor uses a standard 7-bit I2C 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 I2C EEPROMs, for LSB = 1 the protocol is extended by a CRC protection. Thus, each device occupies two I2C 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.

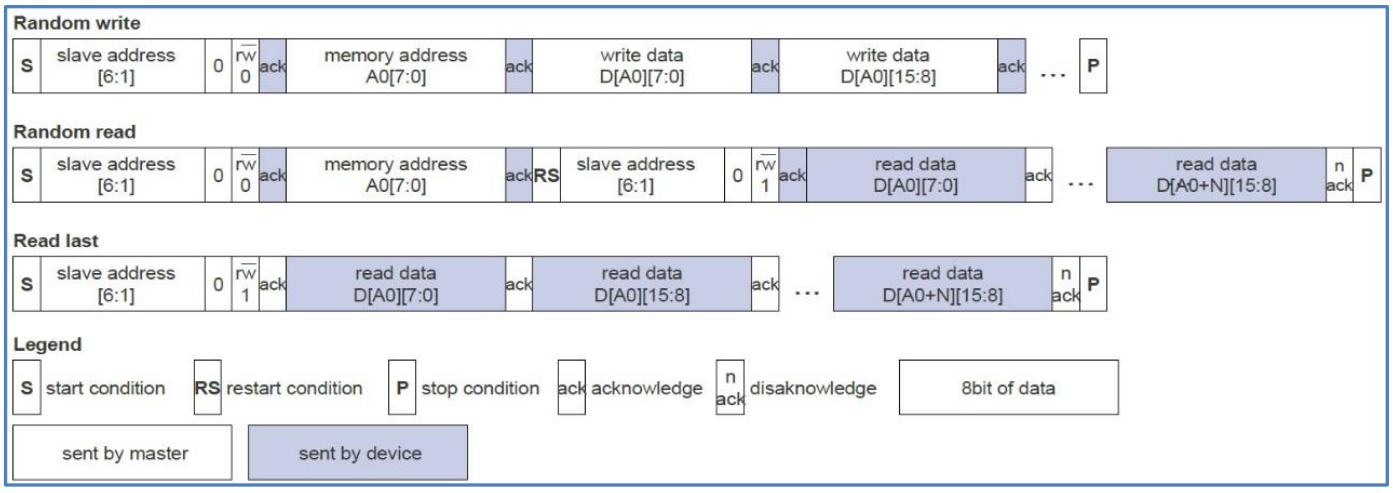


Figure 2: I²C Read / Write Commands - Standard EEPROM Compatible Protocol

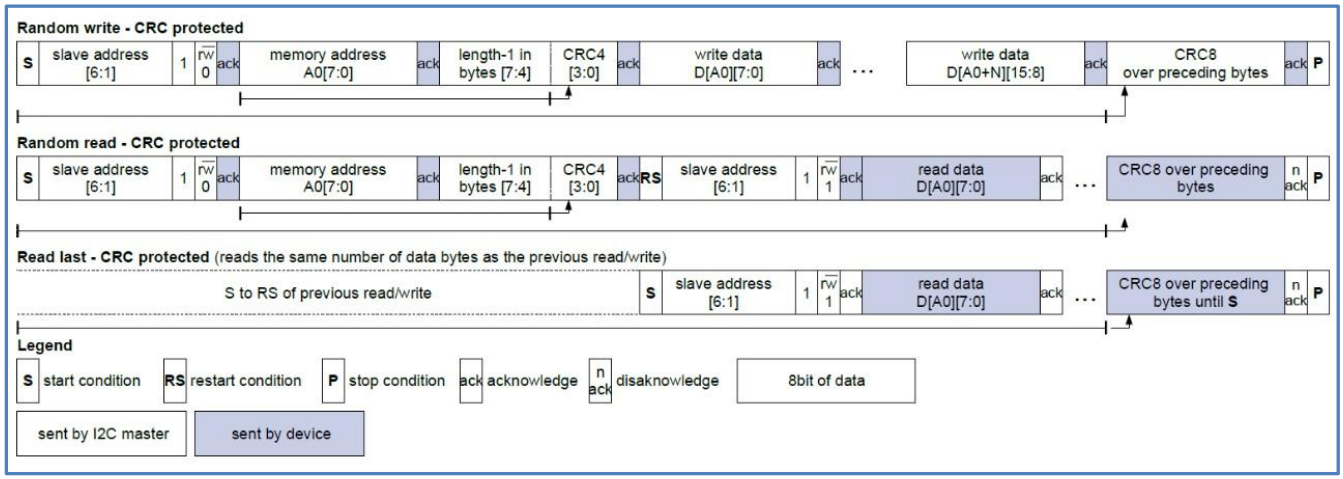


Figure 3: I²C Read / Write Commands - CRC Protected Protocol

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.

I²C 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.

Random Read

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.

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)

Random Write

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.

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				

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.