TN1180
Technical note

How to interpret pressure and temperature readings in pressure sensors

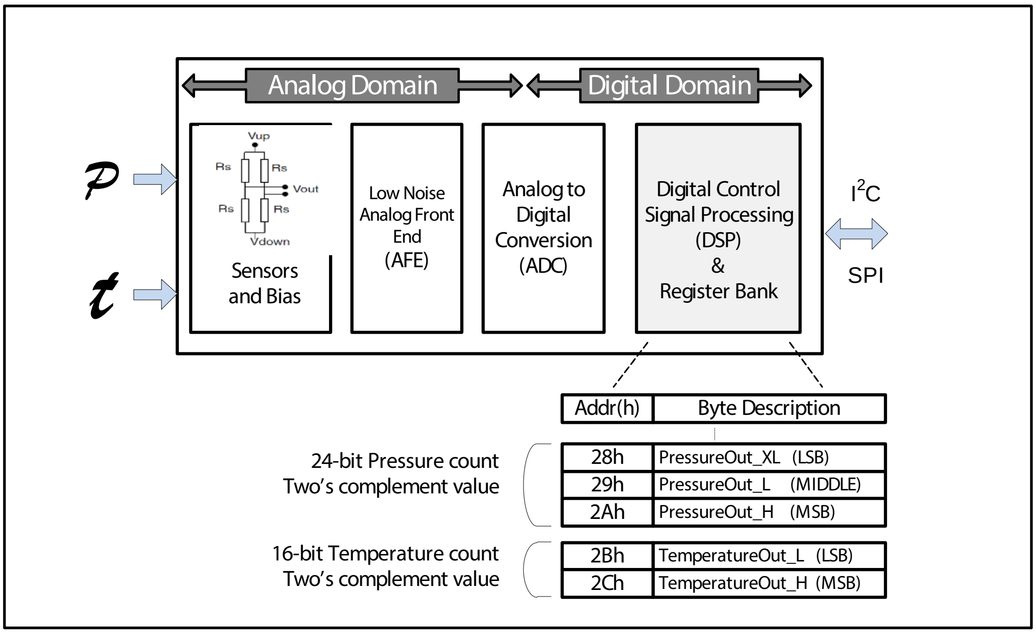
By Tom Bocchino

Introduction

The purpose of this technical note is to provide a review of two's complement notation for ease of design in pressure sensors and to guide the user in the interpretation of pressure and temperature values in the device registers. MEMS pressure sensors measure absolute pressure (*P*) and temperature (*t*) and store the values in two's complement registers which can be read via the I^2C or SPI host interface.

The pressure sensor stores pressure as a 24-bit word. Both values are stored as two's complement integers as illustrated below.

Figure 1: Pressure sensor analog-to-digital data flow



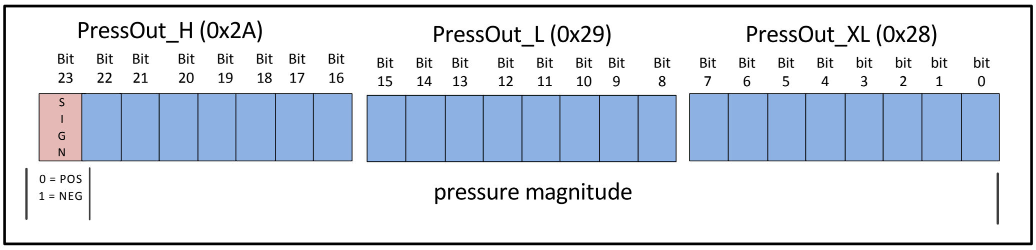
1 Why we use two's complement

Measuring analog values and storing the data in a digital domain creates an interesting dilemma. How do we represent *negative* analog values in a digital system? How do we handle the number zero?

A computing system uses two's complement to simplify the processing logic required to handle negative numbers and subtractions. A circuit designed for addition can handle negative operands by using two’s complement. This minimizes the need for additional circuitry capable of subtraction or additional switching circuitry based on the sign. Most modern computers, MCU’s, and DSP’s use two’s complement notation.

If a register is defined as two’s complement, typically the most significant bit (msb) of the most significant byte (MSB) indicates the sign as shown in the figure below. If the msb of the register is 1, the number is negative and we use two’s complement. If the bit is 0, the integer is positive and no translation is necessary.

Figure 2: Sign bit indicates whether a value is positive or negative



2 How to obtain pressure values in mbar

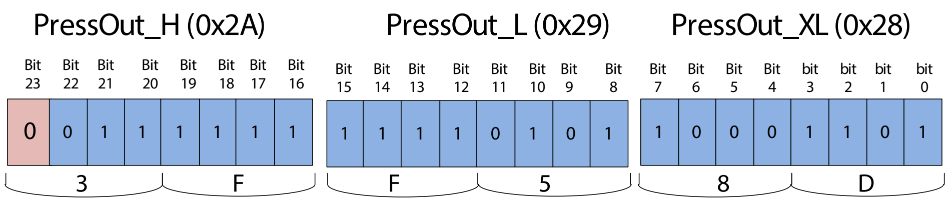
The pressure sensor stores the pressure value in raw counts in 3 registers: PressOut_H, PressOut_L, and PressOut_XL. The most significant bit of the PressOut_H register indicates the polarity. If the sign bit is zero, then the value is positive and the pressure in mbar is determined by dividing the decimal value by the scaling factor of 4096. A sign bit of 1 indicates a negative value, so we first take the two's complement of the complete word and then divide by 4096.

Keep in mind that the sign bit determines whether we should perform the two's complement operation or not. The 3 bytes are concatenated to form a 24-bit word and the complete word is represented in two's complement (not the single bytes).

When reading the pressure value, it is important to note the byte ordering. This is especially important in auto-increment mode when the address is incremented automatically. The designer should assemble the proper 24-bit endian format (pressure count = 2Ah & 29h & 28h).

The device may be configured to report a delta pressure by using the auto-zero feature in CTRL_REG2 (0x21). When using the auto-zero feature, the PressOut() registers could contain a negative value which is the sum difference between the current pressure and the reference pressure. Negative values can also occur when a device is defective. Hence it can be used for self-test during a boot-up operation and when the auto-zero feature is not used. An example of a pressure calculation when the sign bit is 0 is shown in the following figure.

Figure 3: Pressure example 1: sign bit is 0, this is a positive value

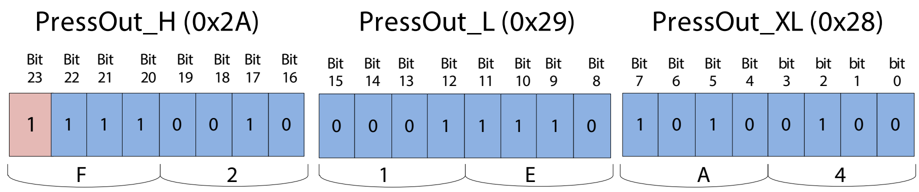


Pressure Counts = 2Ah & 29h & 28h = 3FF58Dh = 4191629(dec)

$$\text{Pressure millibar} = \frac{\#\ \text{counts}}{\text{Scaling factor}} = \frac{4191629\ \text{counts}}{4096\ \text{counts/mbar}} = 1023.3\ \text{mbar}$$

In some applications the device may be configured to report a differential pressure value in the PressOut() registers by using the auto-zero feature. In this case the PressOut() registers could contain a negative value as illustrated in the second example of a pressure calculation shown below. To obtain the pressure in mbar we first take the two’s complement of the complete word and then divide by 4096.

Figure 4: Pressure example 2: negative number, two's complement



1111 0010 0001 1110 1010 0100 ; *need to take the two's complement*

0000 1101 1110 0001 0101 1011 *; invert*

\+ 1 *; add 1*

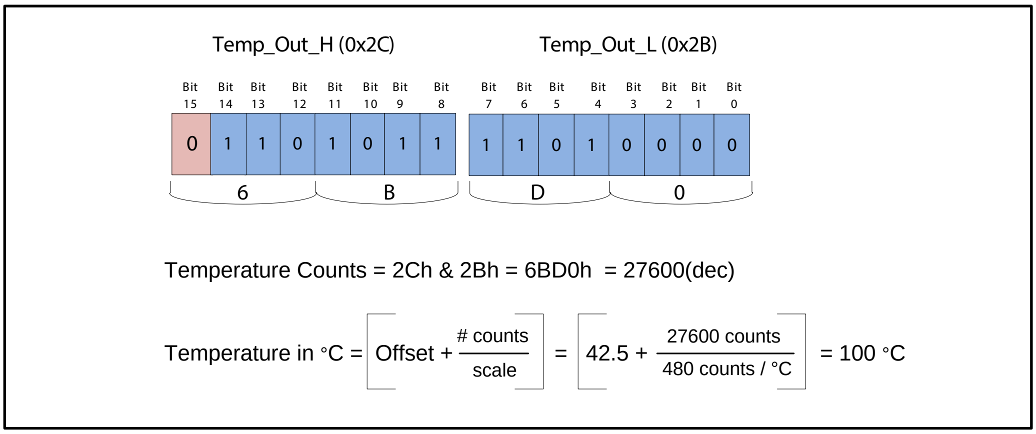
0000 1101 1110 0001 0101 1100 = - 0DE15Ch = - 909660(dec)

$$\text{Pressure in millibar} = \frac{-\,909660\text{ counts}}{4096\text{ counts/mbar}} = -\,222.08\text{ mbar}$$

3 How to obtain temperature values in °C

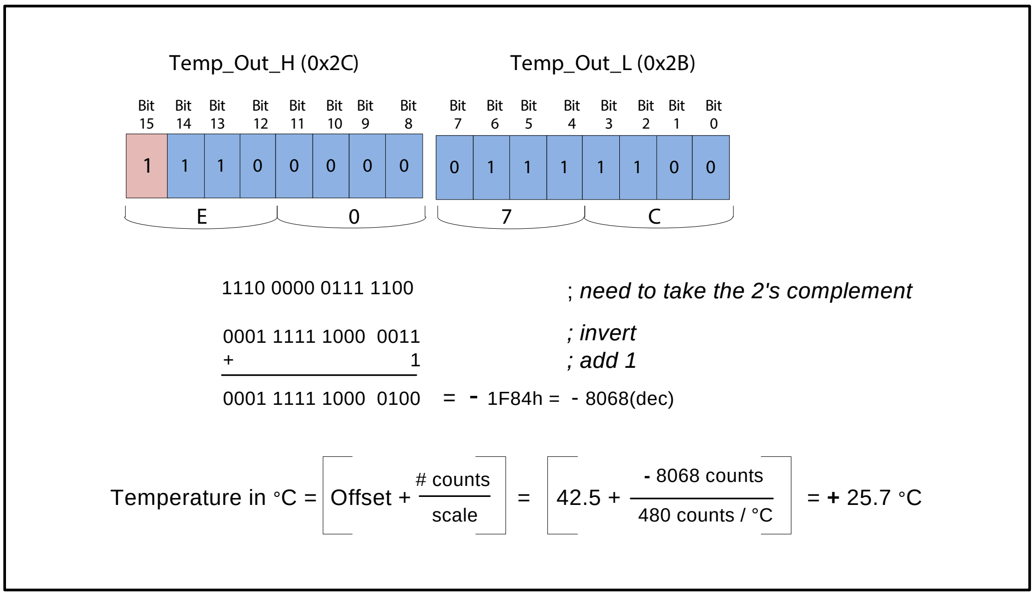
The LPS331AP also contains a temperature sensor which is used for compensation of the pressure sensor. The temperature data is also available for outside applications. The temperature raw count is stored in registers Temp_Out_H (2Ch) and Temp_Out_L (2Bh). The temperature in °C can be determined by calculating the two's complement, if necessary, and perform the scaling. An example of a temperature calculation when the sign bit is 0 is shown below.

Figure 5: Temperature example 1: sign bit is 0, this is a positive value



Likewise, a negative value is indicated when the MSB in the temperature word is equal to 1. To obtain the temperature, we first take the two's complement of the complete word and then perform the scaling and offset operation. An example of a temperature calculation when the sign bit is 1 is shown below.

Figure 6: Temperature example 2: sign bit is 1, two's complement



As shown in the figure above, a negative value in the raw count register does not always result in a negative a temperature value in degrees Celsius. This is because the offset and scaling factor must be applied to determine the final value in °C units.

When reading the temperature values it is also important to note the byte ordering. This is especially true in auto-increment mode in which case address 2Bh is read first, followed by address 2Ch. The designer should assemble the proper 16-bit endian format (temperature count = 2Ch & 2Bh).

The following registers in the device are two's complement. All other registers in the device are either unsigned or bitwise representations (not negative).

Table 1: Pressure sensor two's complement registers

Function	Bytes	Description
PRESS_OUT()	2Ah & 29h & 28h	24-bit absolute pressure data or this contains the difference in pressure between Ref_P() and Press_Out() when auto-zero mode is used
TEMP_OUT()	2Bh & 2Ch	16-bit temperature data
REF_P()	0Ah & 09h & 08h	24-bit absolute pressure in auto-zero mode

4 Support material

Table 2: Related design support material

Document type	Part number	Title
Datasheet/ data brief	LPS331AP	MEMS pressure sensor: 260-1260 mbar absolute digital output barometer
	LPS25H	MEMS pressure sensor: 260-1260 mbar absolute digital output barometer
Evaluation board	STEVAL-MKI109V2	eMotion: ST MEMS adapter motherboard based on STM32. Compatible with all ST MEMS adapters based on STM32F103.
	STEVAL-MKI120V1	LPS331AP adapter board for standard DIL24 socket
	STEVAL-MKI142V1	LPS25H adapter board for standard DIL24 socket
User manual	UM0412	Getting started with DfuSe USB device firmware upgrade STMicroelectronics extension
	UM0979	STEVAL-MKI109V1 and STEVAL-MKI109V2 - eMotion motherboards for MEMS adapter boards
	UM1049	Unico graphical user interface (GUI)
	UM1064	Software guide for Unico lite
Application note	AN4159	Hardware and software guidelines for use of the LPS331AP
	AN4450	Hardware and software guidelines for use of the LPS25H pressure sensor

5 Revision history

Table 3: Document revision history

Date	Revision	Changes
27-May-2014	1	Initial release.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com