DT0004
Design tip

How to interpret the LPS331AP pressure and temperature readings

By Tom Bocchino
and Boon-Nam Poh

Main components	
LPS331AP	260-1260 mbar absolute barometer with digital output

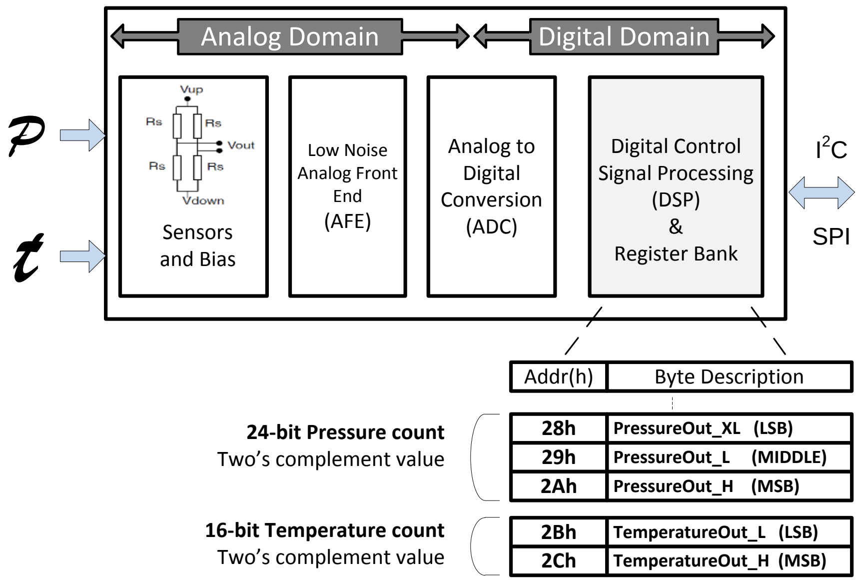
Purpose and benefits

- Review of two's complement notation for ease of design
- How to interpret pressure and temperature values in the LPS331AP

Description

The LPS331AP is a MEMS sensor which measures absolute pressure (*p*) and temperature (*t*) and stores the values in two's complement registers which can be read via the I^2C or SPI host interface. This design tip explains how to interpret the two's complement register values.

Figure 1. LPS331AP Analog to Digital Data Flow



The LPS331AP stores pressure as a 24-bit word and temperature as a 16-bit word. Both values are stored as two's complement integers as illustrated in Figure 1.

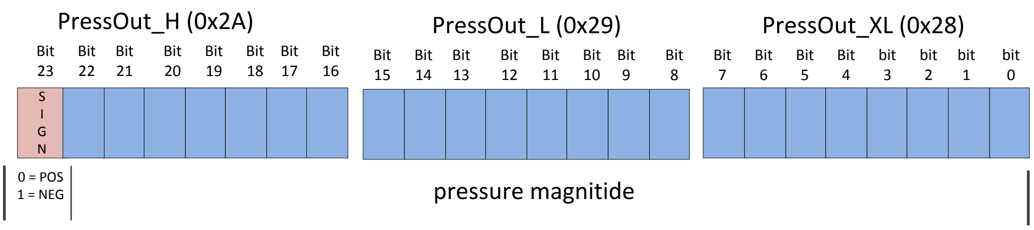
Why we use two's complement

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 Figure 2. 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



How to obtain pressure values in mbar

The LPS331AP 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 example.

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

PressOut_H (0x2A)

Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
0	0	1	1	1	1	1	1
3				F			

PressOut_L (0x29)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1	1	1	1	0	1	0	1
F				5			

PressOut_XL (0x28)

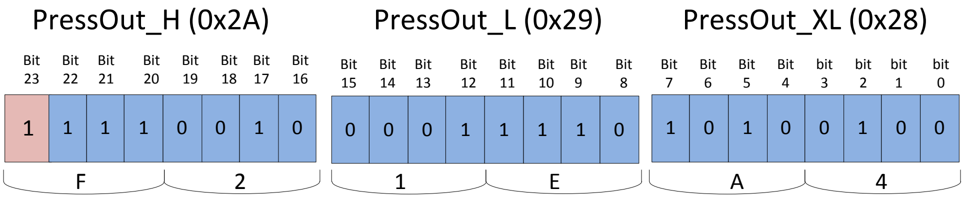
Bit 7	Bit 6	Bit 5	Bit 4	bit 3	bit 2	bit 1	bit 0
1	0	0	0	1	1	0	1
8				D			

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.

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)

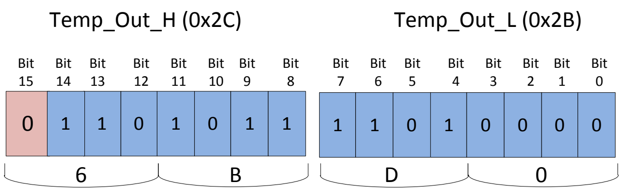
$$\text{Pressure in millibar} = \frac{-\ 909660 \text{ counts}}{4096 \text{ counts/mbar}} = -\ 222.08 \text{ mbar}$$

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.

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



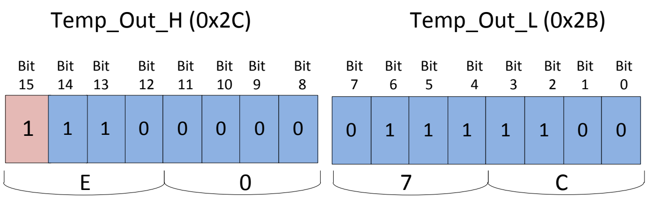
Temperature Counts = 2Ch & 2Bh = 6BD0h = 27600(dec)

$$\text{Temperature in °C} = \left[\text{Offset} + \frac{\#\text{ counts}}{\text{scale}}\right] = \left[42.5 + \frac{27600 \text{ counts}}{480 \text{ counts / °C}}\right] = 100\ °\text{C}$$

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.

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




```
1110 0000 0111 1100              ; need to take the 2's complement

0001 1111 1000  0011             ; invert
+                  1             ; add 1
-----------------------
0001 1111 1000  0100   =  - 1F84h =  - 8068(dec)
```

$$\text{Temperature in °C} = \left[\text{Offset} + \frac{\#\ \text{counts}}{\text{scale}}\right] = \left[42.5 + \frac{-8068\ \text{counts}}{480\ \text{counts / °C}}\right] = +25.7\ \text{°C}$$

As shown in Temperature example 2, 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. LPS331AP 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 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.

Support material

Related design support material	
STEVAL-MKI109V2	eMotion Main Board which accepts several MEMS modules, including pressure sensor plug-in module
STEVAL-MKI120V1	LPS331 Pressure Module for eMotion Board
Documentation	
Datasheet:	LPS331AP, MEMS pressure sensor: 260-1260 mbar absolute digital output barometer
Data brief:	STEVAL-MKI109V2, eMotion: ST MEMS adapters motherboard based on the STM32F103RE compatible with all ST MEMS adapters V2
User Manual:	UM1049, Unico GUI: software guide
Data brief:	STEVAL-MKI120V1, LPS331AP adapter board for a standard DIL24 socket
www.st.com/sensors	

Revision history

Date	Version	Changes
17-Jul-2012	1	Initial release