
ILPS22QS: dual full-scale absolute digital output barometer with embedded Qvar electrostatic sensor

Introduction

This document provides usage information and application hints related to ST's [ILPS22QS](#) device.

The ILPS22QS is an ultra-compact piezoresistive absolute pressure sensor which functions as a digital output barometer with a digital I²C / MIPI I3CSM / 3-wire SPI serial interface standard output. The device supports two selectable pressure ranges, 260 ~ 1260 hPa (mode 1) and 260 ~ 4060 hPa (mode 2), and it is capable of measuring pressure values with output data rates up to 200 Hz.

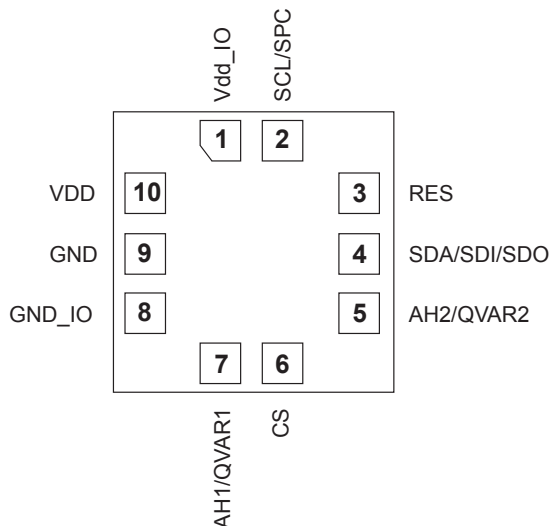
The ILPS22QS has an integrated 128-level first-in first-out (FIFO) buffer allowing the user to store data in order to limit intervention by the host processor.

The ILPS22QS embeds an analog hub sensing functionality which is able to connect an analog input and convert it to a digital signal for embedded processing. In addition, an embedded Qvar (electric charge variation detection) channel can be used for human presence and motion detection, touch detection, and user interface (UI) applications.

The ILPS22QS is available in a full-mold, holed LGA package (HLGA) and is guaranteed to operate over an extended temperature range from -40 °C to +105 °C. The package is holed to allow external pressure to reach the sensing element.

This document does not modify the content of the official datasheet, refer to the datasheet for parameter specifications.

1 Pin description

Figure 1. Pin connections

Table 1. Pin list, functions, and internal status

Pin number	Name	Function	Internal pin status
1	Vdd_IO	Power supply for I/O pins	
2	SCL SPC	I ² C / MIPI I3C SM serial clock (SCL) SPI serial port clock (SPC)	Default: input without pull-up.
3	RES	Connect to GND	
4	SDA SDI SDO	I ² C / MIPI I3C SM serial data (SDA) 3-wire SPI serial data input (SDI) 3-wire SPI serial data output (SDO)	Default: input without pull-up. Pull-up is enabled if bit SDA_PU_EN = 1, register IF_CTRL (@0Eh).
5	AH2/QVAR2	Connect to GND if the analog hub and Qvar are disabled. AH input 2 (or Qvar electrode 2) is connected if the analog hub (or Qvar functionality) is enabled.	Default: input without pull-up.
6	CS	SPI enable, I ² C and MIPI I3C SM / SPI mode selection (1: SPI idle mode / I ² C and MIPI I3C SM communication: enabled; 0: SPI communication mode / I ² C and MIPI I3C SM disabled)	Default: input with pull-up. Pull-up is disabled if bit CS_PU_DIS = 1, register IF_CTRL (@0Eh).
7	AH1/QVAR1	Connect to GND if the analog hub and Qvar are disabled. AH input 1 (or Qvar electrode 1) is connected if the analog hub (or Qvar functionality) is enabled.	Default: input without pull-up.
8	GND_IO	0 V supply	
9	GND	0 V supply	
10	VDD	Power supply	

Note: Internal pull-up value is from 30 kΩ to 50 kΩ, depending on Vdd_IO.

2 Registers

Table 2. Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTERRUPT_CFG	0Bh	AUTOREFP	RESET_ARP	AUTOZERO	RESET_AZ	-	LIR	PLE	PHE
THS_P_L	0Ch	THS7	THS6	THS5	THS4	THS3	THS2	THS1	THS0
THS_P_H	0Dh	-	THS14	THS13	THS12	THS11	THS10	THS9	THS8
IF_CTRL	0Eh	0	I2C_I3C_DIS	EN_SPI_READ	SDA_PU_EN	0	0	CS_PU_DIS	-
WHO_AM_I	0Fh	1	0	1	1	0	1	0	0
CTRL_REG1	10h	0	ODR3	ODR2	ODR1	ODR0	AVG2	AVG1	AVG0
CTRL_REG2	11h	BOOT	FS_MODE	LPPF_CFG	EN_LPPF	BDU	SWRESET	-	ONE_SHOT
CTRL_REG3	12h	AH_QVAR_EN	0	AH_QVAR_P_AUTO_EN	0	0	0	0	IF_ADD_INC
FIFO_CTRL	14h	0	0	0	AH_QVAR_P_FIFO_EN	STOP_ON_WTM	TRIG_MODES	F_MODE1	F_MODE0
FIFO_WTM	15h	0	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0
REF_P_L	16h	REFP7	REFP6	REFP5	REFP4	REFP3	REFP2	REFP1	REFP0
REF_P_H	17h	REFP15	REFP14	REFP13	REFP12	REFP11	REFP10	REFP9	REFP8
I3C_IF_CTRL	19h	1	0	ASF_ON	0	0	0	I3C_Bus_Avb_Sel1	I3C_Bus_Avb_Sel0
RPDS_L	1Ah	RPDS7	RPDS6	RPDS5	RPDS4	RPDS3	RPDS2	RPDS1	RPDS0
RPDS_H	1Bh	RPDS15	RPDS14	RPDS13	RPDS12	RPDS11	RPDS10	RPDS9	RPDS8
INT_SOURCE	24h	BOOT_ON	0	0	0	0	IA	PL	PH
FIFO_STATUS1	25h	FSS7	FSS6	FSS5	FSS4	FSS3	FSS2	FSS1	FSS0
FIFO_STATUS2	26h	FIFO_WTM_IA	FIFO_OVR_IA	FIFO_FULL_IA	-	-	-	-	-
STATUS	27h	-	-	T_OR	P_OR	-	-	T_DA	P_DA
PRESS_OUT_XL	28h	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
PRESS_OUT_L	29h	POUT15	POUT14	POUT13	POUT12	POUT11	POUT10	POUT9	POUT8
PRESS_OUT_H	2Ah	POUT23	POUT22	POUT21	POUT20	POUT19	POUT18	POUT17	POUT16
TEMP_OUT_L	2Bh	TOUT7	TOUT6	TOUT5	TOUT4	TOUT3	TOUT2	TOUT1	TOUT0
TEMP_OUT_H	2Ch	TOUT15	TOUT14	TOUT13	TOUT12	TOUT11	TOUT10	TOUT9	TOUT8
FIFO_DATA_OUT_PRESS_XL	78h	FIFO_P7	FIFO_P6	FIFO_P5	FIFO_P4	FIFO_P3	FIFO_P2	FIFO_P1	FIFO_P0
FIFO_DATA_OUT_PRESS_L	79h	FIFO_P15	FIFO_P14	FIFO_P13	FIFO_P12	FIFO_P11	FIFO_P10	FIFO_P9	FIFO_P8
FIFO_DATA_OUT_PRESS_H	7Ah	FIFO_P23	FIFO_P22	FIFO_P21	FIFO_P20	FIFO_P19	FIFO_P18	FIFO_P17	FIFO_P16



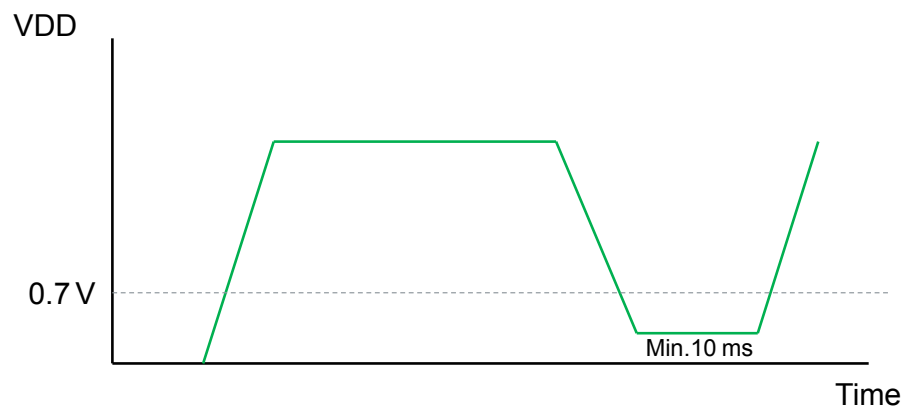
3 Operating modes

The ILPS22QS features three operating modes:

- Power-down mode;
- One-shot mode;
- Continuous mode.

Basically, the ILPS22QS can read environmental data at the very moment the controlling MCU requires it, when configured in one-shot mode, or can keep reading data at predefined frequencies (fixed output data rates, ODRs), when configured in continuous mode. The device offers a wide VDD voltage range from 1.7 V to 3.6 V and a Vdd_IO range from 1.08 V to 3.6 V. The Vdd_IO pin can be powered at the same time as the VDD pin or before. In order to avoid potential conflicts, during the power-on sequence it is recommended to set the lines connected to the device IO pins to high-impedance state on the host side. Furthermore, to guarantee proper device power-off and the next power-on reset sequence to be successful, it is recommended to drive the VDD line to GND (less than 0.7 V) and keep it at this level for at least 10 ms as illustrated in the figure below. Power timings and profiles need to be taken into account when managing the device power supplies (VDD and Vdd_IO).

Figure 2. VDD power-on/off sequence



- VDD Rising / Falling time : 10 μ s ~ 100 ms
- VDD must be lower than 0.7 V for at least 10 ms during power-off sequence for correct POR

After the power supply is applied, the ILPS22QS requires a boot procedure of 10 ms (maximum) to load the trimming parameters. After the boot is completed, the device is configured in power-down mode, ready to communicate with the host for register configurations and data measurements.

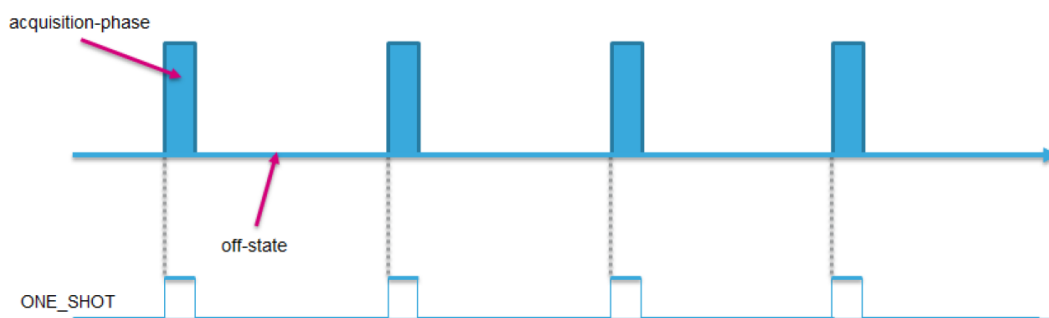
3.1 Power-down mode

Power-down mode is used to put the device in rest condition. When the device is in power-down mode, no data acquisition happens, almost all internal blocks of the device are switched off to minimize current drainage. In power-down mode the ILPS22QS can reach its lowest power consumption achievable while power supplied. While in power-down mode, I²C / MIPI I3CSM / SPI communication serial interfaces are kept active to allow communication with the device and setting configurations. The content of the configuration registers is preserved and the output data registers are not updated, keeping the last data sampled in memory before going into power-down mode. The device is in power-down mode when the ODR[3:0] bits of CTRL_REG1 (@10h) register are set to 0000.

3.2 One-shot mode

One-shot mode is used to execute single data acquisitions at a desired pace. After the acquisition has been completed, the device automatically sets itself to power-down mode. One-shot mode has to be executed while the device is in power-down mode by setting the ONESHOT bit (default value 0) in CTRL_REG2(@11h) to 1. When this happens, a single data acquisition is executed and read data are made available in the output registers. Once the acquisition is completed and the output registers updated, the device automatically enters again power-down mode and the ONE_SHOT bit is self-cleared (to 0).

Figure 3. One-shot mode



One-shot mode requires the bit range ODR[3:0], CTRL_REG1(@10h) register to be set to 0000. This mode depends on the frequency at which the ONESHOT bit is set to 1 by the microcontroller/application processor. The typical time needed for the generation of the new data and the maximum ODR frequency achievable in one-shot mode is given in the following table and strictly depends on the value selected for the Average parameter through bit range AVG[2:0], CTRL_REG1(@10h) register.

Table 3. Typical conversion time and maximum ODR in one-shot mode

AVG[2:0]	Averaging of pressure and temperature	Typical data conversion time [ms]	Maximum ODR [Hz]
111	512	33.4	25
101	128	9.4	75
100	64	5.4	100
011	32	3.4	200
010	16	2.4	300
001	8	1.5	400
000	4	1.2	500

3.3 Continuous mode

Continuous mode is designed to keep reading data at a specific predefined selectable output data rate (ODR). Output registers are updated with fresher readings every given period according to the selected ODR frequency. Continuous mode ODR selection is made through bit range ODR[3:0], CTRL_REG1(@10h) register. When ODR[3:0] bits are set to a value different than 0000 (power-down mode), the device enters continuous mode and immediately starts to sample pressure and temperature data and put them in the output registers at the selected frequency (Table 4).

Table 4. ODR selection

ODR[3:0]	ODR [Hz]
0000	Power-down mode
0001	1
0010	4
0011	10
0100	25
0101	50
0110	75
0111	100
1000	200

3.4 Pressure sensor noise and current consumption

In continuous and one-shot mode a set of configuration options is available. The right trade-off among resolution, output data rate and power consumption has to be identified in order to make the sensor suitable for the specific design requirements.

The power consumption of the ILPS22QS mainly depends on the data rate and on the selected resolution. In continuous mode, the user can select the desired ODR and the oversampling frequency for pressure measurements in the CTRL_REG1(@10h) register. The ODR[3:0] bits are used for ODR selection, while the AVG[2:0] bits are used to configure the resolution. In one-shot mode the output data rate depends on the frequency at which the ONESHOT bit is set to 1 by the microcontroller/application processor.

Table 5 summarizes the pressure sensor typical consumption for different configurations of the ODR and the resolution in continuous mode. Table 6 indicates the corresponding current consumption values obtained when using the pressure sensor in one-shot mode.

Note: The values indicated in the two tables below refer to the condition of the analog hub / Qvar disabled (by writing the register 5Fh to 00h when the device is powered on) and with the AH1/QVAR1 and the AH2/QVAR2 pins connected to GND.

Table 5. Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (AH / Qvar disabled)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	32.8	126.8	314.4	783.8	-	-	-	-
128	10.0	35.6	86.7	214.3	427.0	639.8	-	-
64	6.3	20.4	48.7	119.4	237.2	355.0	472.8	-
32	4.4	12.8	29.8	71.9	142.2	212.6	282.9	564.4
16	3.5	9.0	20.2	48.2	94.8	141.4	188.0	374.0
8	2.7	6.0	12.6	29.1	56.5	84.2	111.5	221.7
4	2.5	5.0	10.2	23.2	44.7	66.2	87.8	174.0

Table 6. Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (AH / Qvar disabled)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	32.2	126.2	313.8	783.2	-	-	-	-
128	9.4	35.0	86.1	213.7	426.4	639.2	-	-
64	5.6	19.7	48.0	118.7	236.5	354.3	472.1	-
32	3.7	12.1	29.1	71.2	141.5	211.9	282.2	563.7
16	2.7	8.2	19.4	47.4	94.0	140.6	187.2	373.2
8	2.0	5.3	11.9	28.4	55.8	83.5	110.8	221.0
4	1.7	4.2	9.4	22.4	43.9	65.4	87.0	173.2

The pressure sensor RMS noise depends on the selected bandwidth and on the selected resolution. The user can select the overall device bandwidth by configuring the LPF1 filter, as described in Section 4.1 Digital low-pass filters. The bit range AVG[2:0], CTRL_REG1(@10h) register is used to configure the resolution.

Table 7 indicates the pressure sensor RMS noise for different configuration of bandwidth and resolution. This table is valid for continuous and one-shot mode.

Table 7. Typical RMS noise [Pa] at VDD = 1.8 V, T = 25 °C

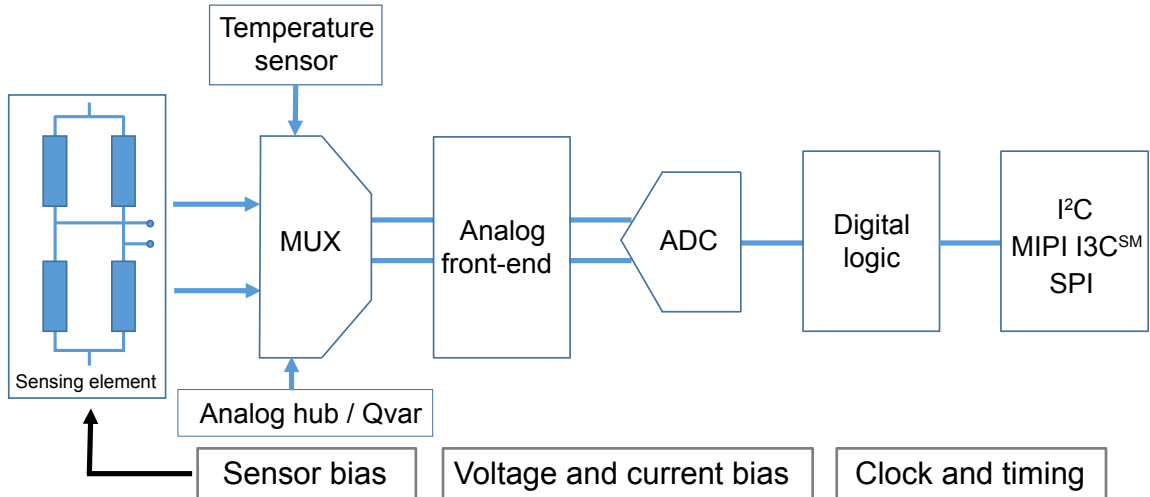
AVG	FS = 1260 hPa			FS = 4060 hPa		
	Pressure noise ($P_{a_{rms}}$)			Pressure noise ($P_{a_{rms}}$)		
	ODR/2 ⁽¹⁾	ODR/4	ODR/9	ODR/2 ⁽¹⁾	ODR/4	ODR/9
512	0.59	0.44	0.34	1.16	0.85	0.64
128	0.90	0.67	0.48	2.15	1.48	1.11
64	1.20	0.86	0.64	2.98	2.05	1.55
32	1.64	1.16	0.87	4.14	2.89	2.10
16	2.30	1.63	1.18	5.86	4.11	2.97
8	3.12	2.18	1.60	8.11	5.73	4.23
4	4.26	2.87	2.20	11.3	7.82	5.81

1. LPF1 filter is disabled.

4 Sampling chain

The ILPS22QS is a piezoresistive absolute pressure sensor that works as a digital output barometer. The device comprises a sensing element and an IC interface that communicates through selectable serial protocols SPI (3-wire), I²C or MIPI I3CSM, from the sensing element to the application.

Figure 4. ILPS22QS architecture block diagram



The sensing element, which detects absolute pressure, consists of a suspended membrane manufactured using a dedicated process developed by STMicroelectronics.

The following sections focus on the "Digital logic" block, describing the ILPS22QS filtering chain and the entire data path.

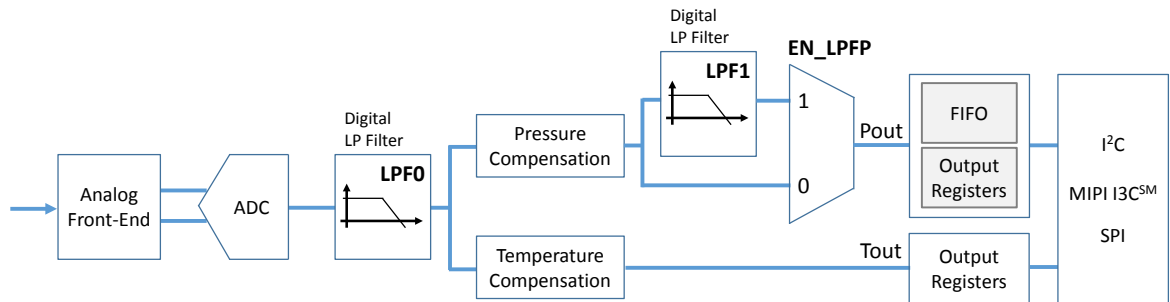
Note:

The analog hub (AH) / Qvar feature is enabled by default: it implies higher power consumption during operating and power-down modes to support AH / Qvar functions. If the AH / Qvar feature is not intended to be used, in order to disable it (and save power consumption), the register at address 5Fh must be written to 00h whenever the application is powered on. Furthermore, AH1/QVAR1 (pin 7) and AH2/QVAR2 (pin 5) must be connected to GND on the PCB.

4.1 Digital low-pass filters

The ILPS22QS filtering chain for pressure and temperature sensors embeds two digital low-pass filters. The first one is the average filter (LPF0), which is applied to both temperature and pressure data. A second low-pass filter (LPF1) is also present. It can be applied optionally on the pressure readout path (temperature data are not filtered at this stage) when the device is in continuous mode. LPF1 affects data also when one-shot is used. Note that LPF1 applies to pressure data only.

Figure 5. ILPS22QS digital LP filters block diagram



The LPF1 digital filter can be enabled by configuring the bit EN_LPFP, CTRL_REG2(@11h) register, and the overall device bandwidth can be configured acting on the bit LPFP_CFG, CTRL_REG2(@11h) register, as shown in Table 9. Setting EN_LPFP = 1 enables the filter and diverts its output to the pressure output registers and FIFO buffer. Setting EN_LPFP = 0 resets the filter, too; the LPF1 filter is also reset if the data rate (bit range ODR[3:0], CTRL_REG1(@10h)) or the filter bandwidth (bit LPFP_CFG, CTRL_REG2(@11h)) or the Average parameter (bit range AVG[2:0], CTRL_REG1(@10h)) is updated.

Table 8. Related registers and bit ranges of the LP filters

Register	Address	Bit	Bit range mask
CTRL_REG2	11h	EN_LPFP	00010000b = 10h
CTRL_REG2	11h	LPFP_CFG	00100000b = 20h

Table 9. Settings of the LP filters

EN_LPFP	LPFP_CFG	LPF1 filter status	Overall device bandwidth	Max overall settling time ⁽¹⁾ (samples to be discarded)
0	x	Disabled, filter reset	ODR/2 (LPF0 only)	0 (first sample correct)
1	0	Enabled	ODR/4	1
1	1	Enabled	ODR/9	6

1. Settling time @ 99% of the final value.

Table 9. Settings of the LP filters indicates the number of samples that should be discarded when the filter is enabled and/or after it's reset, before the filter reaches settling condition and output data can be considered meaningful.

4.2 Data path

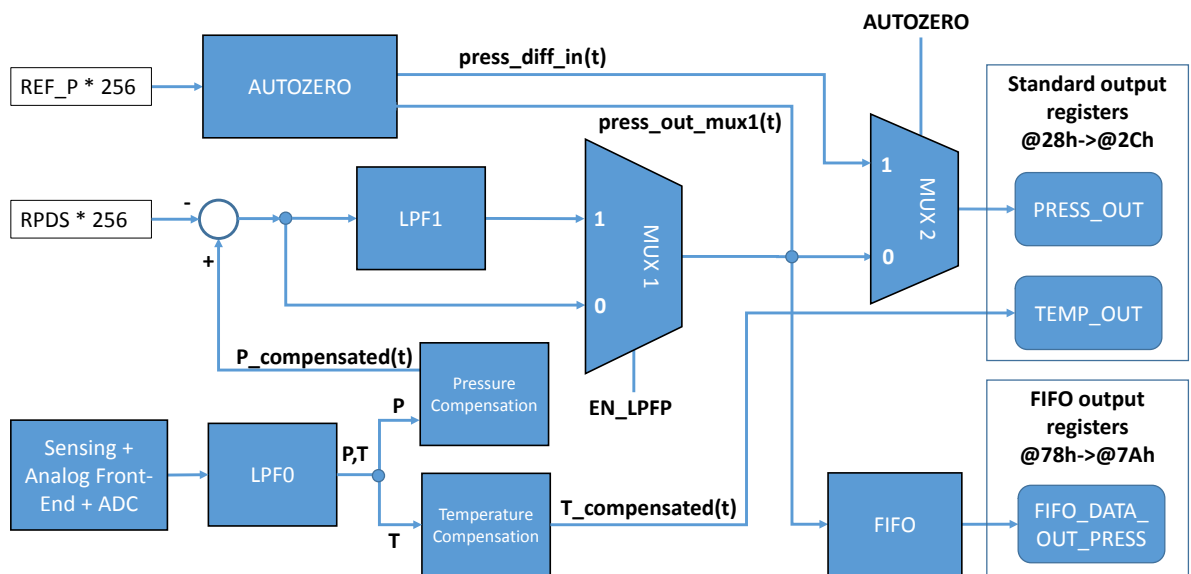
The following block diagram (Figure 6. Data path for output registers (standard and FIFO)) highlights the data path for pressure and temperature data from the sensing element and the analog-to-digital conversion to the standard output registers and FIFO under different operating/setting conditions for filters, FIFO and features like comparison with thresholds and references.

Details for the specific blocks can be found in the related paragraphs.

Important details are:

- Temperature output registers section always receive $T_compensated(t)$ signal data;
- FIFO buffer is always filled with signal $press_out_mux1(t)$;
- Standard output register pressure section always receives $press_out_mux1(t)$, except when the AUTOZERO feature is engaged;

Figure 6. Data path for output registers (standard and FIFO)



5 Reading output data

Once the device has been power supplied, it automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure completes, that is, after 10 milliseconds (maximum) from power-on, the sensor automatically enters power-down mode.

To turn on the sensor and gather pressure and temperature data through the primary I²C / MIPI I3CSM / SPI interface, it is necessary to select one of the operating modes through the ODR[3:0] bits in CTRL_REG1(@10h) register (continuous mode) or to set the ONESHOT bit to 1 in CTRL_REG2(@11h) (one-shot mode).

Data generated on board can be acquired by the controlling MCU by reads of the output registers.

Reading can be done:

- Asynchronously, polling the output registers at the proper pace;
- Synchronously, leveraging on the data-ready signal;
- Delayed, leveraging on the FIFO buffer, which allows improved power consumption (see [Section 9 First-in, first-out \(FIFO\) buffer](#)).

The basic startup sequence and the options available for reading data are described in the following paragraphs.

As a general rule, it is always recommended to read pressure and temperature samples from standard output (PRESS_OUT_x and TEMP_OUT_x) registers and from FIFO output registers (FIFO_DATA_OUT_PRESS_x) as well, starting from the lower address to the higher one, avoiding a different reading order.

5.1 Multi-read / write automatic address increment and automatic wraparound features

5.1.1 Automatic address increment feature

Available serial interfaces protocols allow executing single-read/write and multi-read/write register operations.

A single read/write operation is quite simple and requires specifying the register address that is going to be read/written.

A multi-read/write operation allows easily and effectively executing a repeated read/write operation during a single bus transaction. Once the user knows the starting register and the desired number N of repetitions, according to two different behaviors:

- Read N times the same register at the address provided;
- Read a sequence of N registers starting from the address provided.

The IF_ADD_INC bit in CTRL_REG3(@12h) allows switching between the two behaviors by automatically incrementing the read/write address.

Setting IF_ADD_INC = 1 (default) enables automatic address increments in multi-read and multi-write register operations. This allows executing faster and more effective bus transactions to read/write adjacent registers.

For example, executing a multi-read operation of N=5 bytes starting from register PRESS_OUT_XL(@28h) results in reading with a single transaction all standard output registers: from PRESS_OUT_XL(@28h) to TEMP_OUT_H(@2Ch), without the need for explicitly managing each single address involved.

5.1.2 Automatic address wraparound feature

Automatic address wraparound is another helpful feature available in the ILPS22QS, related to automatic address increment and designed to facilitate burst readings of standard output and FIFO output samples. It is always enabled when `IF_ADD_INC = 1`, `CTRL_REG3(@12h)`.

It allows effectively reading multiple times the output register ranges (both standard or FIFO) by executing one single multi-read operation.

It is available for both output register ranges:

- `STANDARD_OUTPUT (PRESS_OUT_x, TEMP_OUT_x)(@28h-@2Ch)`, 5 bytes long;
- `FIFO_OUTPUT(@78h-@7Ah)`, 3 bytes long.

Reading them starting from the initial register range to the final one results in the auto-increment feature providing, as the next address being read, the initial address range once the last address range has been read.

For example, a multi-read of 10 bytes ($M=2$, $2*5$ bytes), starting at `@28h` results in reading two times the standard output range (from `@28h` to `@2Ch`), as for the following sequence:

```
@28h->@29h->@2Ah->@2Bh->@2Ch->@28h->@29h->@2Ah->@2Bh->@2Ch
```

This feature is effective in reading/emptying the FIFO buffer data.

5.2 Startup sequence

To turn on the device and gather pressure/temperature data, it is necessary to select one of the operating modes. The following general-purpose sequence can be used to configure the ILPS22QS device:

1. Write `CTRL_REG1(@10h) = 27h` // `ODR = 25 Hz, AVG = 512`

5.3 Using the status register

The device is provided with a `STATUS(@27h)` register that should be polled to check when a new set of data (a pressure sample and a temperature sample) is available.

The `P_DA` bit is set to 1 whenever a new sample is available in the pressure output registers; the `T_DA` bit is set to 1 whenever a new sample is available in the temperature output registers.

The `P_DA` bit is cleared when the corresponding pressure sample is read (its most significant byte: `PRESS_OUT_H(@2Ah)`).

The `T_DA` bit is cleared when the corresponding temperature sample is read (its most significant byte: `TEMP_OUT_H(@2Ch)`).

The `STATUS(@27h)` register also includes the overrun flags: `P_OR` bit for pressure samples and `T_OR` bit for temperature samples. They are individually set to 1 when the corresponding sample is generated and the corresponding `DA` bit is already at 1, meaning the previous sample has been overwritten unread by the later generated new one, hence its value has been lost. The overrun bits are automatically cleared when all the data present inside the device have been read and new data have not been produced in the meantime.

The data-ready signals are represented by the `P_DA` and `T_DA` bits, `STATUS(@27h)` register.

Pressure and temperature data are synchronously generated; hence, bits `P_DA` and `T_DA` are synchronously rising at 1 (unless one of the two isn't already at 1), but not synchronously resetting at 0. It depends on when the respective data are read.

Reading the output registers before $1/ODR$ time period has expired allows acquiring the data and resetting `P_DA` and `T_DA` before an overwrite happens.

For the pressure sensor (for temperature sensor it is similar), the read operation of the output registers should be performed as follows:

1. Read `STATUS(@27h)`.
2. If `P_DA = 0`, then go to 1.
3. Read `PRESS_OUT_XL(@28h)`.
4. Read `PRESS_OUT_L(@29h)`.
5. Read `PRESS_OUT_H(@2Ah)`.
6. Data processing
7. Go to 1.

If the device is configured in one-shot mode instead of continuous mode, the routine hangs at step 1 after one execution, since the device performs a single measurement, sets the P_DA/T_DA bit high and returns to power-down mode. Note that the ONE_SHOT bit is self-cleared when the device returns to power-down mode. It is possible to trigger another one-shot reading by setting the ONE_SHOT bit to 1 again.

5.4 Using the block data update (BDU) feature

If reading the data is particularly slow and cannot be synchronized (or it is not required to be) with the P_DA/T_DA event bit in the STATUS(@27h) register, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL_REG2(@11h) register.

This feature avoids reading values (XL, L, and H parts of output data) related to different samples. In particular, when the BDU is activated, the output data registers always contain the most recent output data produced by the device, but, in case the read of a given part (for example, pressure, starting from PRESS_OUT_XL(@28h)) is initiated, the refresh of the remaining bytes for that part (pressure), remains blocked until all XL, L and H parts of the data are read.

The same happens for the temperature section. If TEMP_OUT_L(@2Bh) is read, TEMP_OUT_H(@2Ch) content does not update until read.

BDU applies to both pressure data and temperature data, but manages them separately. When the BDU feature is enabled, pressure and temperature data are separately refreshed depending on when PRESS_OUT_H(@2Ah) / TEMP_OUT_H(@2Ch) is read.

Note: To guarantee the correct behavior of the BDU feature, PRESS_OUT_H(@2Ah) / TEMP_OUT_H(@2Ch), must be the last address read.

The BDU feature also acts on the FIFO_STATUSx(@25h, @26h) registers. When the BDU bit is set to 1, it is mandatory to read FIFO_STATUS1 first and then FIFO_STATUS2.

5.5 Understanding output data

5.5.1 Pressure data

The measured pressure data are sent to the PRESS_OUT_XL(@28h), PRESS_OUT_L(@29h), PRESS_OUT_H(@2Ah) registers. When AUTOZERO = 0, these registers contain, respectively, the least significant byte, the middle significant byte and the most significant byte of the pressure data.

The complete pressure data is given by the concatenation PRESS_OUT_H & PRESS_OUT_L & PRESS_OUT_XL and it is expressed as a binary number.

Pressure data is represented as 24-digit signed 2's complement binary numbers, (each digit referred to as LSB).

To translate the digital representation to its corresponding real number with its SI unit (Pa for pressure), a sensitivity parameter must be applied.

Each pressure sample must be divided by the proper sensitivity parameter P_{sens} (refer to the datasheet) in order to obtain the corresponding value in hPa. The sensitivity value to be applied depends on the selected full-scale range:

- Mode 1 (full scale up to 1260 hPa): $P_{\text{sens}} = 4096$ [LSB/hPa]
- Mode 2 (full scale up to 4060 hPa): $P_{\text{sens}} = 2048$ [LSB/hPa]

5.5.2 Example of pressure data

Hereafter is a simple example of how to get the pressure LSB data and transform it into hPa.

1. Get raw data from the sensor:
 - PRESS_OUT_XL(@28h): 1Ah
 - PRESS_OUT_L(@29h): 84h
 - PRESS_OUT_H(@2Ah): 3Eh
2. Do registers concatenation:
 - PRESS_OUT_H & PRESS_OUT_L & PRESS_OUT_XL: 3E841Ah
3. Calculate signed decimal value (from signed 2's complement 24-digit binary format):
 - P[LSB]: +4097050d
4. Apply P_{sens} sensitivity (mode 1, when FS_MODE = 0):
 - P[hPa] = +4097050 / 4096 = +1000.2563

5.5.3 Temperature data

The measured temperature data are sent to the TEMP_OUT_L(@2Bh), TEMP_OUT_H(@2Ch) registers. These registers contain, respectively, the least significant byte and the most significant byte of the temperature data.

The complete temperature data is given by the concatenation of TEMP_OUT_H & TEMP_OUT_L registers and it is expressed as binary signed number by two's complement representation.

Temperature data is represented as 16-bit signed binary number, each digit referred to as LSB.

To translate the digital representation to its corresponding real number with its SI unit (°C, Celsius degrees, for temperature), a sensitivity parameter for temperature must be applied.

Each temperature sample must be divided by the proper sensitivity parameter (refer to the datasheet) in order to obtain the corresponding value in °C:

$$T_{\text{sens}} = 100 \text{ [LSB/°C]}$$

5.5.4 Example of temperature data

Hereafter is a simple example of how to get the temperature LSB data and transform it into °C.

1. Get raw data from the sensor:
 - TEMP_OUT_L(@2Bh): 7Bh
 - TEMP_OUT_H(@2Ch): FEh
2. Do register concatenation:
 - TEMP_OUT_H & TEMP_OUT_L: FE7Bh
3. Calculate signed decimal value (from 16-bit signed represented by two's complement format):
 - T[LSB]: -389d
4. Apply T_{sens} sensitivity:
 - $T[°C] = -389 / 100 = -3.89$

6 Reboot and software reset

After the device is powered up, the ILPS22QS performs a 10 ms (maximum) boot procedure to load the trimming parameters. After the boot is completed, the device is automatically configured in power-down mode.

During the boot time the registers are not accessible. Anyway, in order to check when the boot procedure is completed, the user can read the BOOT_ON bit of register INT_SOURCE (@24h). If this bit is equal to 1, the boot is running, when it goes to 0, the boot is ended.

After power-up, when the BOOT bit of the CTRL_REG2(@11h) register is set to 1, the trimming parameters are reloaded and the registers RPDS_L(@1Ah) and RPDS_H(@1Bh) are reset to 0.

No toggle of the device power lines is required. After the reboot is completed, the device enters in power-down mode (regardless of the selected operating mode) and the BOOT bit is self-cleared to 0. The BOOT_ON bit described above can be used to check when the reboot procedure has ended.

Note: The reboot reloads also the default value of the register 5Fh, so after the reboot is completed it is necessary to set the register 5Fh to 00h if the analog hub / Qvar feature needs to be disabled.

If the reset to the default value of the device registers is required, it can be performed by setting the SWRESET bit of the CTRL_REG2(@11h) register to 1. When this bit is set to 1, the following registers are reset to their default value:

- INTERRUPT_CFG(@0Bh)
- THS_P_L(@0Ch)
- THS_P_H(@0Dh)
- IF_CTRL(@0Eh)
- CTRL_REG1(@10h)
- CTRL_REG2(@11h)
- CTRL_REG3(@12h)
- FIFO_CTRL(@14h)
- FIFO_WTM(@15h)
- INT_SOURCE(@24h)
- FIFO_STATUS1(@25h)
- FIFO_STATUS2(@26h)
- STATUS(@27h)

The software reset procedure can take a few tens of μ s; the status of the reset is signaled by the status of the SWRESET bit of the CTRL_REG2(@11h) register. Once the reset is completed, this bit is automatically set low.

In order to avoid conflicts, the reboot and the software reset must not be executed at the same time (do not set to 1 at the same time both the BOOT bit and SWRESET bit of the CTRL_REG2(@11h) register).

The flow must be performed serially as shown in the example below:

1. Set the BOOT bit of the CTRL_REG2(@11h) register to 1.
2. Wait 10 ms (or wait until the BOOT_ON bit of the INT_SOURCE (@24h) register returns to 0).
3. Set the SWRESET bit of the CTRL_REG2(@11h) register to 1.
4. Wait 50 μ s (or wait until the SWRESET bit of the CTRL_REG2(@11h) register returns to 0).

7 Offset compensation (OPC - one-point calibration)

If, after the soldering of the component, a residual pressure offset is still present, it can be removed with a one-point calibration (OPC), leveraging on the RPDS registers which can store the residual offset value to be removed.

The calibration offset value is expected to be stored as a signed 16-bit value expressed as 2's complement in the RPDS_L(@1Ah) and RPDS_H(@1Bh) registers.

The default value for RPDS is 0 (zero).

The content of the RPDS registers is always automatically subtracted from the compensated pressure output and provided to the standard output pressure registers PRESS_OUT_x(@28h, 29h and 2Ah) and FIFO. It is provided as the difference between the measured pressure and the content of the RPDS registers (@1Ah, @1Bh) multiplied by 256, when AUTOZERO = 0 (see [Section 4.2 Data path](#)).

8 Interrupt modes

The ILPS22QS has a built-in configurable interrupt generator block, which allows generating interrupt events based on pressure samples and comparisons to customizable references and thresholds.

Interrupt event signals can be selected and enabled; they are made available at a dedicated status register (INT_SOURCE(@24h)) to monitor them.

Additionally, other signals are generated by the FIFO buffer subsystem which can be used as event signals as well and monitored through the FIFO_STATUS1(@25h) and FIFO_STATUS2(@26h) registers (see [Section 9.2.2 Monitoring the FIFO buffer status](#)).

The interrupt generator block control is managed through the INTERRUPT_CFG(@0Bh) register.

The interrupt events related to pressure and temperature sampling are the following:

- Data-ready: new data available
- Threshold-based

The events generated by the FIFO buffer are the following:

- FIFO watermark
- FIFO full
- FIFO overrun

8.1 Interrupt events related to pressure and temperature sampling

8.1.1 Data-ready

If data generation is enabled, it is possible to identify when a new pressure or new temperature data value has been generated and is ready to be read by monitoring the STATUS(@27) register bits.

Every time a new pressure data value is ready, the bit P_DA, STATUS(@27h) register, is set to 1.

Every time a new temperature data value is generated, the bit T_DA in STATUS(@27h) register is set to 1.

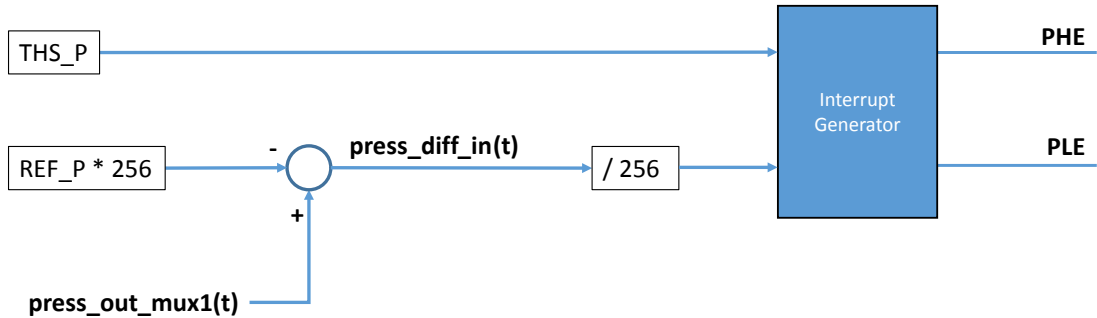
Pressure and temperature data are synchronously generated, see [Section 5.3 Using the status register](#).

The P_DA bit in STATUS(@27h) register is always latched: it is reset when the PRESS_OUT_H(@2Ah) register is read.

8.1.2 Threshold-based differential interrupt

The interrupt generator in ILPS22QS allows generating an interrupt event based on a *user-defined pressure threshold value*, THS_P, which can be stored in two dedicated registers THS_P_L(@0Ch) and THS_P_H(@0Dh). The interrupt generator generates interrupt signals based on a comparison executed on the differential signal *press_diff_in* and the *user-defined pressure threshold* stored in THS_P, as illustrated in [Figure 7. Interrupt generator](#).

Figure 7. Interrupt generator



Signal *press_diff_in* is the difference between compensated pressure samples taken at MUX1 output (*press_out_mux1*, see [Figure 6](#)) and an instantaneous sample of the same signal taken at enabling time and stored in dedicated registers REF_P_L (@16h) and REF_P_H (@17h).

The instantaneous sample is automatically stored in read-only REF_P registers, REF_P_L(@16h) and REF_P_H(@17h), when one of the two differential modes (AUTOZERO mode or AUTOREFP mode, they are described later) is engaged.

Only the compensated pressure signal 16 most significant bits are stored in REF_P and used to obtain differential signal *press_diff_in*.

$$press_diff_in(t) = press_out_mux1(t) - REF_P * 256$$

$$\text{with } REF_P = press_out_mux1(t=Differential_Mode_Engaging_Time) / 256.$$

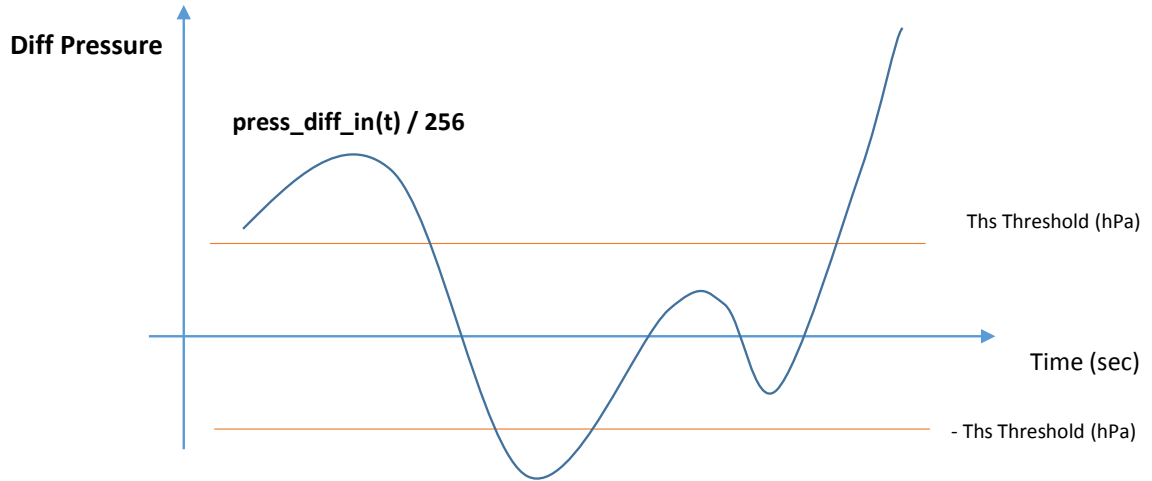
Note: *Differential Mode Engaging Time* is the instant when AUTOZERO mode or AUTOREFP mode is engaged.

Basically, thresholds (THS_P) and references (REF_P), in combination with *press_out_mux1(t)*, are used for comparisons: it is important to note that THS_P is a differential pressure threshold, not an absolute pressure threshold. Thresholds and references have to be considered as the 2 most significant bytes with respect to the standard pressure samples which are 24-bit (or 3 bytes) long. For this reason, the *press_diff_in(t)* data is divided by 256 before being processed by the interrupt generator logic.

When enabled, the device executes a comparison between the *press_diff_in* samples (divided by 256) and THS_P at each new sample arrival (1/ODR period). In this way, the user-defined pressure threshold (THS_P), identifies three areas in which each *press_diff_in* sample (divided by 256) can fall:

- above +THS_P
- below -THS_P
- between -THS_P and +THS_P

The interrupt generator allows selecting the generation of two different interrupt signals (PLE and PHE) which rise when the *press_diff_in(t)/256* signal falls below -THS_P and/or above +THS_P areas (see [Figure 8](#)).

Figure 8. Differential interrupt input signal and threshold relationship


The desired interrupt threshold quantity used for pressure interrupt generation, expressed as hPa, has to be converted to binary as a 15-bit unsigned right-justified notation and must be stored in THS_P_H(@0Dh) for MSByte and THS_P_L(@0Ch) for LSByte.

The binary unsigned integer to be stored in THS_P can be computed starting from its physical value, expressed as hPa as follows:

- $\text{THS_P [LSB]} = \text{abs}(\text{interrupt pressure threshold [hPa]}) * 4096 \text{ [LSB/hPa]} / 256 = \text{abs}(\text{interrupt pressure threshold [hPa]}) * 16$ in mode 1 (full scale up to 1260 hPa, FS_MODE = 0)

or

- $\text{THS_P [LSB]} = \text{abs}(\text{interrupt pressure threshold [hPa]}) * 2048 \text{ [LSB/hPa]} / 256 = \text{abs}(\text{interrupt pressure threshold [hPa]}) * 8$ in mode 2 (full scale up to 4060 hPa, FS_MODE = 1)

Example:

Suppose mode 1 (up to 1260 hPa) was selected and the desired interrupt pressure threshold = 10 hPa;

$\text{THS_P} = 10\text{d} * 16 = 160\text{d} = 00\text{A0h}$

values to store in THS_P are:

THS_P_H = 00h; THS_P_L = A0h

Differential interrupt can be used by selecting between two different modes:

- AUTOREFP
- AUTOZERO

Both of them result in:

- storage in REF_P of current sample of *press_out_mux1* at engaging time. As for above description.
- enabling interrupt generation for above positive threshold and/or below negative threshold.

The output available in the standard output registers is different according to the two different modes (see Section 4.2 Data path):

- AUTOZERO: standard output registers report the differential *press_diff_in(t)* signal.
- AUTOREFP: standard output registers keep reporting the usual output (as for all other configurations *press_out_mux1(t)* signal).

To enable the above interrupt modes, the PHE bit or PLE bit (or both of them) in INTERRUPT_CFG(@0Bh) have to be set to 1 to enable, respectively, the interrupt generation on the positive or negative events. The desired threshold values must be stored in the THS_P_L(@0Ch) and THS_P_H(@0Dh) registers.

Finally, according to the AUTOZERO or AUTOREFP choice, bit AUTOREFP and AUTOZERO need to be set to 1 (see other details in Section 8.1.2.1 AUTOZERO mode and Section 8.1.2.2 AUTOREFP mode).

8.1.2.1
AUTOZERO mode

AUTOZERO mode when engaged:

- instantly triggers the storage of current pressure measurements in dedicated registers (REF_P);
- reports to standard output registers, pressure part, (@28h->@2Ah) the current pressure decreased by the REF_P*256 stored sample (*press_diff_in(t)* signal);
- detects when pressure variations with respect to REF_P are over the user-defined differential threshold (\pm THS_P) and generates separate interrupt signals (PLE, PHE) accordingly.

At the instant it is engaged ($t = t_{\text{AUTOZERO}}$), the current measured pressure sample is stored in the REF_P(@16h, @17h) registers and used as the pressure reference. From the engaging time on, the output pressure registers PRESS_OUT(@28h, @29h and @2Ah) are filled with:

$$\text{PRESS_OUT}(t) = \text{press_out_mux1}(t) - \text{REF_P} * 256$$

where:

$$\text{REF_P} = \text{press_out_mux1}(t = t_{\text{AUTOZERO}}) / 256$$

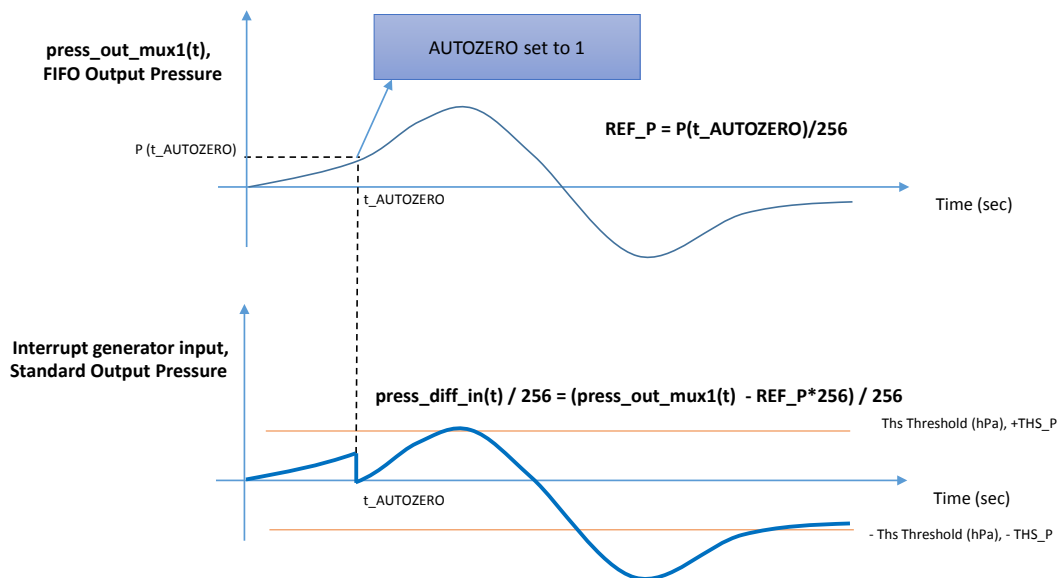
The signal diverted to the interrupt generator and used as input for the interrupt generation is *press_diff_in(t)/256*, with *press_diff_in(t)* defined as follows:

$$\text{press_diff_in}(t) = \text{press_out_mux1}(t) - \text{REF_P} * 256$$

Hence, what is available in the standard output registers (PRESS_OUT) for pressure is a differential pressure, the same used for interrupt generation.

In case the differential pressure value reported in the standard output registers (PRESS_OUT[LSB]) has to be converted to the corresponding value in hPa, it can be done by dividing the PRESS_OUT[LSB] by a factor equal to 4096 [LSB/hPa] in mode 1 (full scale up to 1260 hPa, FS_MODE = 0) or equal to 2048 [LSB/hPa] in mode 2 (full scale up to 4060 hPa, FS_MODE = 1).

Figure 9. Differential interrupt AUTOZERO mode, outputs and thresholds



To enable the AUTOZERO differential interrupt feature, bits AUTOZERO, PLE and/or PHE, need to be set to 1. After the first conversion, the AUTOZERO bit is automatically set back to 0, but the Autozero mode remains engaged. To disable the engaged Autozero feature and return back to normal mode, an explicit action is required: the bit RESET_AZ, INTERRUPT_CFG(@0Bh) register has to be set to 1. The RESET_AZ bit, autonomously sets back to 0, too.

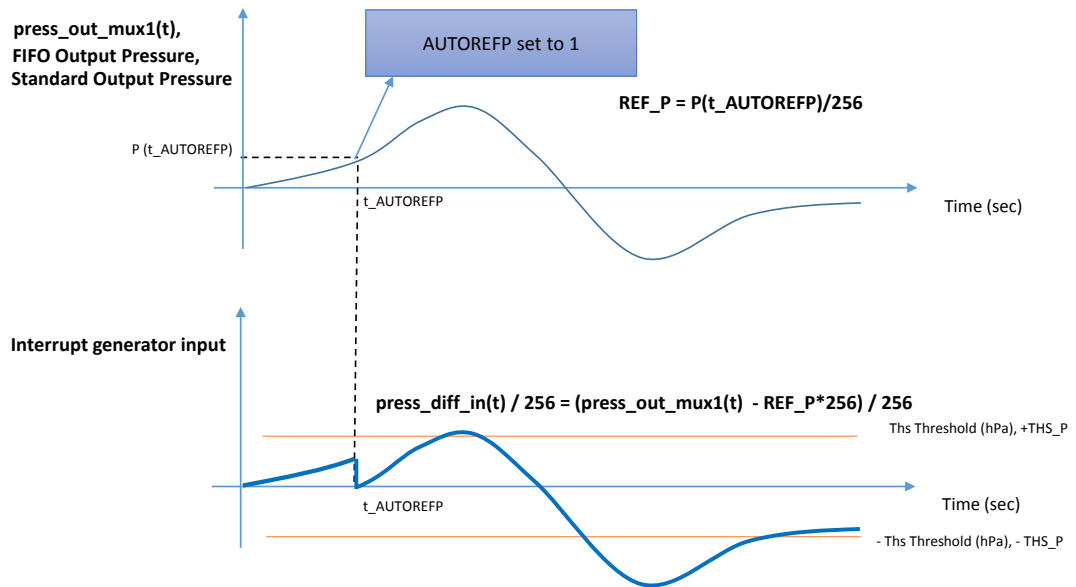
The Autozero feature disengagement effects are: resets REF_P to default value 0 (zero), content of standard output registers switches back to default signal: $\text{PRESS_OUT}(t) = \text{press_out_mux1}(t)$.

8.1.2.2 AUTOREFP mode

The AUTOREFP mode, when engaged:

- instantly triggers the storage of the current pressure measurements in dedicated registers (REF_P);
- reports to standard output registers, pressure part, (@28h->@2Ah) the *press_out_mux1(t)* signal;
- detects when pressure variations with respect to REF_P are over the user-defined threshold (\pm THS_P) and generates separate interrupt signals (PLE, PHE) accordingly.

Figure 10. Differential interrupt AUTOREFP mode, outputs and thresholds



At the instant it is engaged ($t = t_{\text{AUTOREFP}}$), the current measured pressure sample is stored in the REF_P(@16h, @17h) registers and used as the pressure reference. From the engaging time on, the output pressure registers PRESS_OUT(@28h, @29h and @2Ah) keep being filled, as usual, with:

$$\text{PRESS_OUT}(t) = \text{press_out_mux1}(t)$$

The signal diverted to the interrupt generator and used as input for the interrupt generation is *press_diff_in(t) / 256*, with *press_diff_in(t)* defined as follows:

$$\text{press_diff_in}(t) = \text{press_out_mux1}(t) - \text{REF_P} * 256$$

Where:

$$\text{REF_P} = \text{press_out_mux1}(t = t_{\text{AUTOREFP}}) / 256$$

Hence, what is available in the standard output registers (PRESS_OUT) for pressure is the usual pressure signal: this is different with respect to what happens for AUTOZERO mode.

To enable the AUTOREFP differential interrupt feature, bits AUTOREFP, PLE and/or PHE, need to be set to 1.

After the first conversion, the AUTOREFP bit is automatically set back to 0, but the AUTOREFP mode remains engaged. To disable the engaged AUTOREFP feature and return back to normal mode, an explicit action is required: the bit RESET_ARP, INTERRUPT_CFG(@0Bh) register has to be set to 1. The RESET_ARP bit, autonomously sets back to 0, too.

The AUTOREFP feature disengagement effect is: resets REF_P to default value 0 (zero).

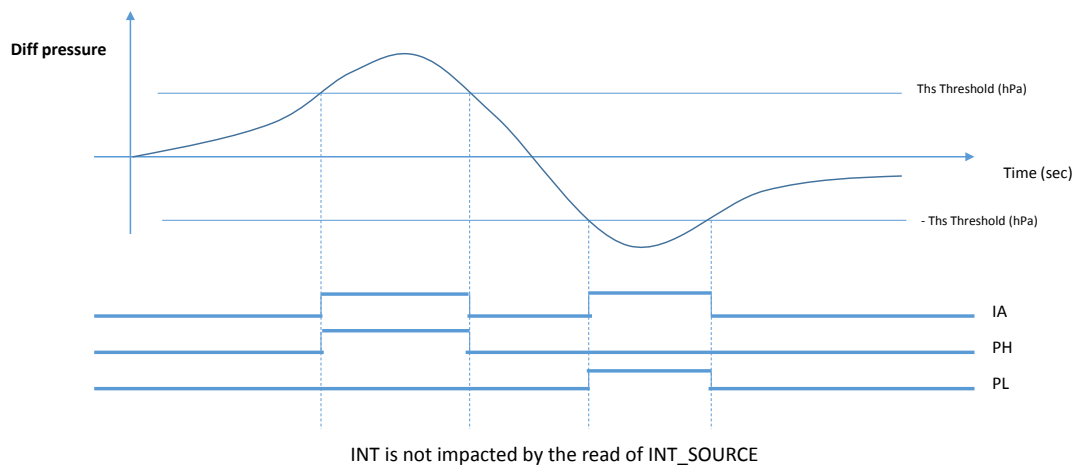
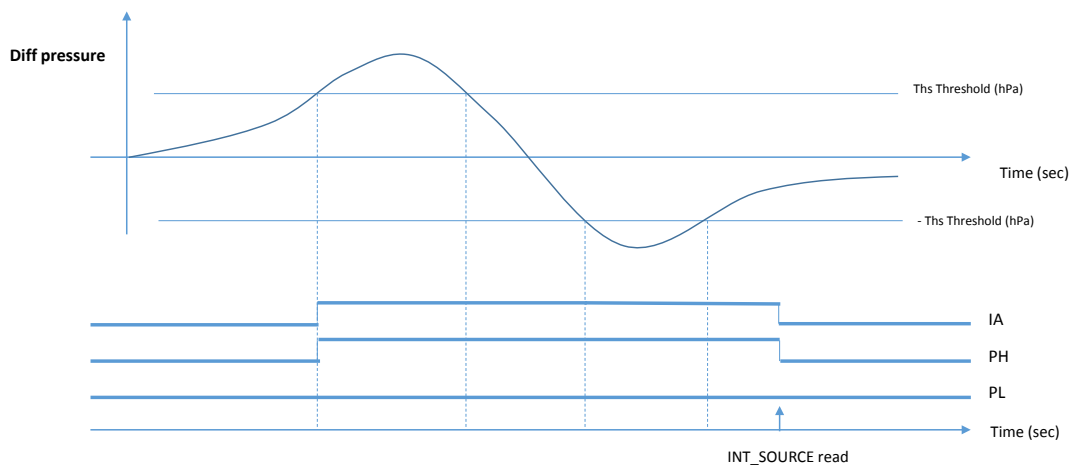
8.1.2.3 Interrupt latching

Threshold-based differential interrupts offer the option to latch the content of the INT_SOURCE(@24h) register. This option is controlled through bit LIR, INTERRUPT_CFG(@0Bh).

Latching has the effect to keep the content of the register bits “frozen”, even if the condition which triggered their rise is no longer valid, until the INT_SOURCE register has been read.

If latched, when the IA bit rises, the latch feature keeps it at 1 and “freezes” the PL and PH content, too, until the INT_SOURCE register is read.

When latching is not enabled, the IA value and PL and PH values as well keep updating every 1/ODR time period according to *press_diff_int(t)* and THS_P.

Figure 11. Latch disabled (LIR = 0): interrupt behavior

Figure 12. Latch enabled (LIR = 1): interrupt behavior


8.1.3 Interrupt events related to FIFO status

With the ILPS22QS pressure sensor, when FIFO is running, it generates, according to its selected operating mode, a set of event signals which allow monitoring its status and which are available in the FIFO_STATUS2(@26h) register (see [Section 9 First-in, first-out \(FIFO\) buffer](#)).

8.1.4 Interrupt events for FIFO triggered modes

It is possible to leverage on differential interrupt events related to pressure sample values (see [Section 8.1.2 Threshold-based differential interrupt](#)) to trigger transitions in FIFO triggered modes.

For the following FIFO buffer modes:

- Continuous (dynamic-stream)-to-FIFO mode
- Bypass-to-continuous (dynamic-stream) mode
- Bypass-to-FIFO mode

the bit IA, INT_SOURCE(@24h) register, with: IA = (PL or PH), INT_SOURCE(@24h) is used as an internal trigger event signal to drive the switch from one FIFO behavior to the following one according to the selected mode (see [Section 9.3.2 Triggered FIFO modes](#)).

9 First-in, first-out (FIFO) buffer

The ILPS22QS provides an embedded 128-slot deep, first-in, first-out (FIFO) buffer to store pressure data. The FIFO buffer allows limiting intervention by the host processor and facilitates post-processing data for event or pattern detection and analysis. FIFO usage allows remarkable power saving for the system: the host CPU can wake up only when notified by FIFO events, burst the significant data out from the FIFO, return to sleep. The FIFO buffer can work according to six different selectable modes that guarantee a high-level of flexibility during application development:

- Bypass mode
- FIFO mode
- Continuous (dynamic-stream) mode
- Bypass-to-FIFO mode
- Bypass-to-continuous (dynamic-stream) mode
- Continuous (dynamic-stream)-to-FIFO mode

The device interrupt generator monitors the FIFO operating parameters which are used for generating FIFO-related interrupt signals that are available in dedicated registers.

9.1 FIFO description

The FIFO buffer is able to store up to 128 "data sample sets". Each "data sample set" (dss, from now on) is sized at 3 bytes corresponding to one pressure sample (24-bit) and is the logic atomic information for FIFO.

When FIFO is enabled, the samples from the pressure sensor are also diverted to feed the FIFO buffer at the currently selected output data rate (ODR). The data path for feeding standard output registers and FIFO buffer is common. The dss feeding FIFO are always the data coming out from the LPF blocks (see [Section 4.2 Data path](#)). While FIFO is fed, the current dss data remain available in the standard output registers.

Usually dss available in the standard output registers `PRESS_OUT_XL(@28h)`, `PRESS_OUT_L(@29h)`, `PRESS_OUT_H(@2Ah)` are the same data sets filling the FIFO buffer (but when interrupt `AUTOZERO` is enabled, `AUTOZERO=1`, in this case what is available in the standard outputs is different from what is sent to FIFO).

The number of FIFO slots can be limited according to user needs by enabling the "Stop on Watermark level" feature which allows setting a custom level of fullness lower than the allowed maximum of 128 dss.

FIFO behaves as a resizable circular buffer: each new dss is placed in the first available empty FIFO slot until the buffer reaches fullness level. Once the fullness level has been reached, the oldest value is overwritten or FIFO stops filling according to the selected operating mode. Reading a dss from `FIFO_OUTPUT` registers removes it from the FIFO buffer. Each time FIFO is fed by a new dss or a dss is read, FIFO and its status registers are updated accordingly.

9.2 FIFO settings and control

Upon device power-up, FIFO is not enabled: the pressure sensor data are not stored in the FIFO, but stored in the standard output pressure registers only. The FIFO can be configured and controlled by setting two registers:

- FIFO_CTRL(@14h): for FIFO mode and FIFO trigger mode selection and enabling FIFO depth limit (watermark)
- FIFO_WTM(@15h): for setting the FIFO watermark level

The FIFO buffer status can be monitored by:

- FIFO_STATUS1(@25h): for reading the FIFO stored data level during operation
- FIFO_STATUS2(@26h): for reading the FIFO status during operation

Table 10. List of registers involved in managing FIFO

FIFO control registers			
Register	@address	Bit range	Bit range mask
FIFO_CTRL	14h	F_MODE[1:0]	0000011b = 03h
FIFO_CTRL	14h	TRIG_MODES	00000100b = 04h
FIFO_CTRL	14h	STOP_ON_WTM	00001000b = 08h
FIFO_WTM	15h	WTM[6:0]	01111111b = 7Fh

Table 11. FIFO modes and FIFO trigger mode selection

TRIG_MODES	F_MODE1	F_MODE0	FIFO mode selection
x	0	0	Bypass mode: turn off and reset FIFO
0	0	1	FIFO mode
0	1	x	Continuous (dynamic-stream) mode
1	0	1	Bypass-to-FIFO mode
1	1	0	Bypass-to-continuous (dynamic-stream) mode
1	1	1	Continuous (dynamic-stream)-to-FIFO mode

9.2.1 Limiting FIFO depth: stop-on-watermark level

The full FIFO buffer can store up to max. depth = 128 depth levels of dss, or slots.

The FIFO depth can be logically sized, limited by the stop-on-watermark feature. When enabled, it logically resizes FIFO, or the number of its available slots, reducing its depth. It works by defining a watermark level and enabling it.

Defining the required FIFO depth limit is performed by setting bit range WTM[6:0], FIFO_WTM(@15h) register, its maximum value is 127d = 7Fh.

When the stop-on-watermark feature is enabled, FIFO depth level is then equal to the value stored in bit range WTM[6:0]. The watermark limit is enabled by setting bit STOP_ON_WTM = 1, FIFO_CTRL(@14h) register.

The watermark level needs to be set and enabled before enabling FIFO; it cannot be changed while FIFO is already running.

Table 12. Watermark settings

Register	@address	Bit range	Bit range mask
FIFO_CTRL	14h	STOP_ON_WTM	00001000b = 08h
FIFO_WTM	15h	WTM[6:0]	01111111b = 7Fh

9.2.2 Monitoring the FIFO buffer status

The FIFO buffer status can be monitored by reading the dedicated registers. FIFO_STATUSx(@25h, @26h) are read-only registers that allow monitoring the current FIFO status.

Note: The BDU feature also acts on the FIFO_STATUS1 and FIFO_STATUS2 registers. When the BDU bit is set to 1, it is mandatory to read FIFO_STATUS1 first and then FIFO_STATUS2.

The FSS[7:0] bit range, FIFO_STATUS1(@25h) register provides information about the current number of data set samples (dss) stored in the FIFO buffer, hence:

- FSS is equal to 00000001b when 1 data set is stored in the FIFO.
- FSS is equal to 10000000b when 128 data sets are stored in the FIFO.

The FIFO_WTM_IA bit, FIFO_STATUS2(@26h) register is a watermark monitor and is asserted while the current FIFO buffer filling level is equal to or higher than the level defined by WTM[6:0], FIFO_WTM(@15h). The FIFO current filling level, is available in FSS[7:0], FIFO_STATUS1(@25h). The monitor works even if the stop-on-watermark feature is not enabled (STOP_ON_WTM = 0).

The FIFO_WTM_IA bit value reflects the comparison result between FSS[7:0] and WTM[6:0], FIFO_WTM(@15h) bit range value. A comparison is enabled and the monitor bit is working when WTM[6:0] is set to a value between 1d and 127d. If bit range WTM[6:0] = 0, then the watermark monitor is de-asserted: FIFO_WTM_IA = 0, FIFO_STATUS2(@26h).

FIFO_WTM_IA = 1, FIFO_STATUS2(@26), if and while the number of stored sample sets in the FIFO buffer FSS[7:0], FIFO_STATUS1@25) is greater than or equal to the watermark level selected by bit range WTM[6:0], FIFO_CTRL(@14h).

The FIFO_FULL_IA bit, FIFO_STATUS2(@26h), goes to 1 if and while FIFO is completely filled and no dss in FIFO have been overwritten, which means it remains at level 1 for a duration of 1/ODR.

The FIFO_OVR_IA bit, FIFO_STATUS2(@26h), goes to 1 if the FIFO buffer is full and at least one dss in the FIFO has been overwritten by the last generated dss.

While FIFO is running, the FIFO_STATUSx registers bits keep updating with the current status: they are reset when FIFO is reset.

The three status bits in FIFO_STATUS2(@26h) are helpful to trigger actions related to the specific buffer conditions and FIFO operating modes. See [Section 8 Interrupt modes](#) for details.

Table 13. FIFO buffer status monitoring registers and bit ranges

Register	@address	Bit range	Bit range mask
FIFO_STATUS1	25h	FSS[7:0]	11111111b = FFh
FIFO_STATUS2	26h	FIFO_WTM_IA	10000000b = 80h
FIFO_STATUS2	26h	FIFO_OVR_IA	01000000b = 40h
FIFO_STATUS2	26h	FIFO_FULL_IA	00100000b = 20h

9.3 FIFO buffer modes

Three main FIFO buffer behaviors are available:

- Bypass mode: the FIFO buffer is disabled and arriving dss are not filling it, hence bypassing the buffer.
- FIFO mode: arriving dss are filling all the available buffer slots until its selected size is completely filled (hence full or to stop-on-watermark level is reached) and no more empty slots are available at which point the buffer stops filling and keeps stored dss in it until an action is taken. Once filled, newer dss are lost (not stored in the FIFO buffer).
- Continuous (or dynamic-stream) mode: when arriving dss are filling all the available buffer slots until its selected size is completely filled (hence full or watermark level is reached) and no more empty slots are available at which point new arriving dss start to replace older stored data. Once filled, older dss are lost (replaced in the FIFO buffer by newer dss).

FIFO buffer behaviors can be singularly selected (non-triggered modes), default, or as a sequence of two behaviors with the transition from the first to the second one triggered by an event signal (triggered modes). Triggered mode combinations can be selected by setting bit: TRIG_MODES = 1, FIFO_CTRL(@14h) and by bit range F_MODE[1:0], FIFO_CTRL(@14h), see Table 11. FIFO modes and FIFO trigger mode selection. The event signal is the interrupt event generated when the threshold-based differential interrupt generator is enabled: IA, INT_SOURCE(@24h) register.

9.3.1 Non-triggered FIFO modes

9.3.1.1 Bypass mode

In bypass mode, (TRIG_MODES & F_MODE[1:0] = x00, TRIGMODES = X, FIFO_CTRL(@14h)). When bypass mode is selected, the full FIFO buffer content is cleared and all is cleared and reset. The FIFO buffer is not operational and it remains empty. If the device is operational, the pressure values are sent to the PRESS_OUT_x registers (standard output) only. Switching to bypass mode must be used in order to stop and to reset the FIFO buffer and its counters. This is the mandatory step to switch between different FIFO operating modes.

Figure 13. FIFO bypass mode

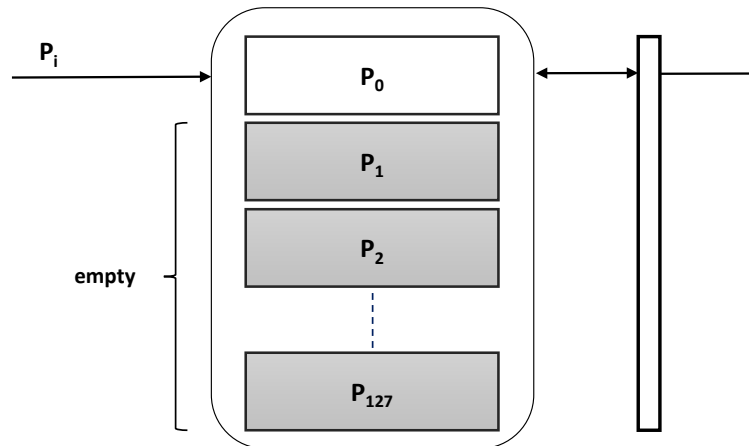


Table 14. Settings for FIFO buffer bypass mode

Register	@address	Bit range	Value	Mask	Mode
FIFO_CTRL	14h	TRIG_MODES & F_MODE[1:0]	X00	00000111b = 07h	Bypass mode

9.3.1.2 FIFO mode

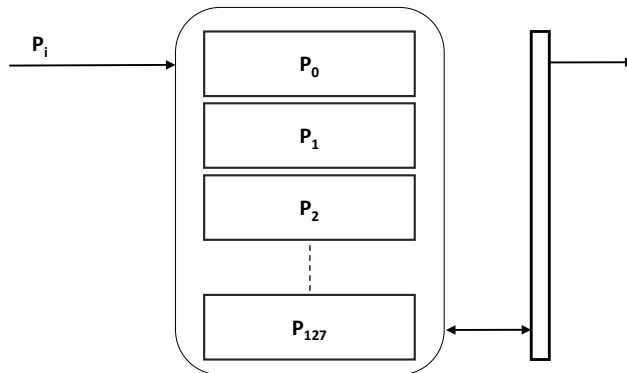
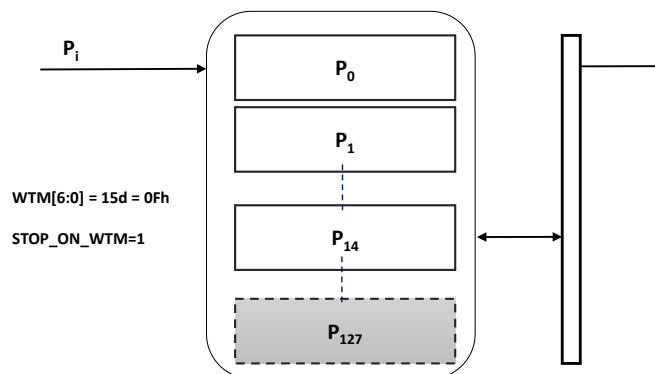
In FIFO mode, (TRIG_MODES & F_MODE[1:0] = 001, FIFO_CTRL(@14h)), the acquired dss generated at the current ODR pressure data are stored in the FIFO buffer.

The FIFO buffer keeps filling until the FIFO is full (128 slots) or the selected and enabled watermark level is reached (if STOP_ON_WTM=1, FIFO_CTRL(@14h)).

When one of the two above conditions is true, the FIFO filling process stops and data in FIFO is no longer updated and remains stored, unchanged until another action is taken. At this point, the FIFO content can be read. FIFO cannot be filled again until the buffer reset operation is executed. This is done by switching it to “bypass mode”, F_MODE[1:0] = 00, FIFO_CTRL(@14h). To use FIFO mode again once it has stopped, the FIFO reset step is mandatory.

After this reset command, it is possible to re-engage FIFO mode by setting F_MODE[1:0] = 01, FIFO_CTRL(@14h).

In FIFO mode, if the stop-on-watermark feature isn't enabled (STOP_ON_WTM=0, FIFO_CTRL(@14h)), the FIFO buffer filling status can be monitored by checking FIFO_STATUS2(@26h), bit FIFO_FULL_IA. On the other hand, if the stop-on-watermark feature is enabled (STOP_ON_WTM = 1, FIFO_CTRL(@14h)), limited filling can be monitored by FIFO_STATUS2(@26h), bit FIFO_WTM_IA status.

Figure 14. FIFO mode with disabled stop-on-watermark feature

Figure 15. FIFO mode with enabled stop-on-watermark feature

Table 15. Settings for FIFO buffer FIFO mode

Register	@address	Bit range	Value	Mask	Mode
FIFO_CTRL	14h	TRIG_MODES & F_MODE[1:0]	001	00000111b = 07h	FIFO mode

9.3.1.3 Continuous (or dynamic-stream) mode

When continuous (or dynamic-stream) mode is enabled through the following settings: (TRIG_MODES & F_MODE[1:0] = 01x, FIFO_CTRL(@14h)), the FIFO buffer starts running and the newer dss feed the available FIFO buffer slots. Once the FIFO is full (or the watermark level is reached if the stop-on-watermark feature has been configured and enabled), each new arriving dss overwrites the older dss stored in the buffer: hence this older dss is unread.

While the fullness status and zero slots are available, the overwriting cycle continues until an action is taken on FIFO which can be:

- a (multi-)read operation
- a FIFO reset

To avoid losing data sets, a read operation should be performed at a pace faster than the ODR which results in free FIFO slots. The reading speed of the host processor is important in order to free slots faster than new data sets are made available.

To stop continuous mode configuration, bypass mode must be selected which results in stopping and resetting FIFO.

While continuous mode is active, the FIFO collects data continuously and the FIFO_STATUS1 and FIFO_STATUS2 registers keep updating according to the number of stored dss.

When the next FIFO write operation renders the FIFO completely full, the FIFO_FULL_IA bit, FIFO_STATUS2(@26h) register goes to 1 and remains at 1 for a duration of 1/ODR.

The next and successive arriving dss (if no slot has not been released in the meantime) overwrites the oldest dss and generates a FIFO overrun condition. The FIFO_OVR_IA bit, FIFO_STATUS2(@26h) register rises to indicate when at least one FIFO dss has been overwritten to store the new data.

Data can be retrieved after the FIFO_FULL_IA event by reading the FIFO_DATA_OUT_PRESS_x (from 78h to 7Ah) registers for the number of times specified by the content of the FIFO_STATUS1 register. Otherwise, leveraging on the FIFO_WTM_IA bit, FIFO_STATUS2(@26h) register, data can alternatively be retrieved when a threshold level, (WTM[6:0] in FIFO_WTM(@15h) register, is reached.

If the stop-on-watermark feature is enabled, bit STOP_ON_WTM = 1, FIFO_CTRL(@14h) register, the FIFO buffer size is limited to a customizable depth (number of slots or dss), equal to the value stored in the bit range WTM[6:0], FIFO_WTM(@15h) register. In this case, bit FIFO_WTM_IA, FIFO_STATUS2(@26h) register rises to 1, when the number of samples in FIFO reaches or surpasses the WTM[6:0] value.

If, after FIFO_WTM_IA has risen, a read operation happens which reduces the amount of stored dss to a level lower than the one defined by WTM[6:0], the FIFO_WTM_IA de-asserts.

When empty slots are no longer available, the WTM[6:0] level is exceeded and an overwrite happens. As soon as this condition occurs, the FIFO_OVR_IA bit, FIFO_STATUS2(@26h) register rises to 1.

9.3.2 Triggered FIFO modes

Triggered modes are a combination, a sequence of two, of the previously mentioned FIFO buffer behaviors, with the switch between the first and the second one triggered by one of the interrupt event conditions made available by the interrupt settings. Once the second behavior has been set, this condition remains until an action is taken on FIFO. Three triggered modes are available.

The triggered modes are dependent on the interrupt active signal edge, bit IA, INT_SOURCE(@24h).

The IA signal is the output of the interrupt generator. The interrupt generator can be configured to toggle the IA signal according to different conditions and settings (see Section 8 Interrupt modes).

9.3.2.1 Bypass-to-FIFO mode

In bypass-to-FIFO mode (TRIG_MODES & F_MODE[1:0] = 101, FIFO_CTRL(@14h)), the FIFO works in “bypass mode” until an interrupt trigger event is generated, then it switches to FIFO mode. The trigger event is set through INTERRUPT_CFG(@0Bh) register.

When the interrupt is triggered, bit IA, INT_SOURCE(@24h) rises to 1 and the FIFO switches from bypass to FIFO mode.

When the interrupt is de-asserted, the IA bit, INT_SOURCE(@24h) is equal to 0, the FIFO doesn't automatically switch back to bypass mode: an action on it has to be taken according to the FIFO mode description.

Figure 16. Bypass-to-FIFO mode

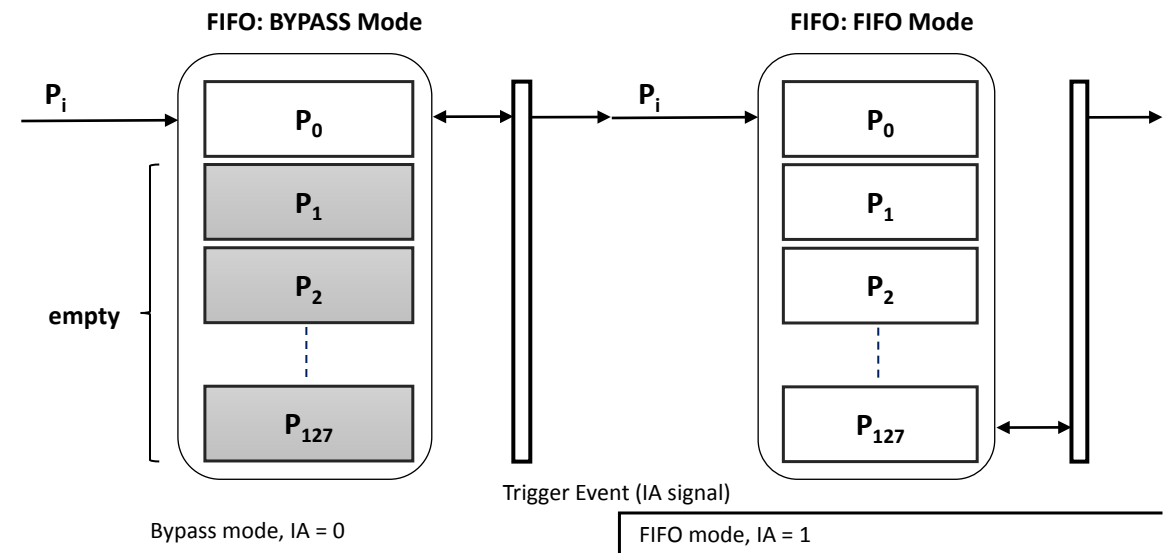


Table 16. Settings for FIFO buffer bypass-to-FIFO mode

Register	@address	Bit range	Value	Mask	Mode
FIFO_CTRL	14h	TRIG_MODES & F_MODE[1:0]	101	00000111b = 07h	Bypass-to-FIFO mode

9.3.2.2 **Bypass-to-continuous (dynamic-stream) mode**

When bypass-to-continuous (dynamic-stream) mode (TRIG_MODES & F_MODE[1:0] = 110, FIFO_CTRL(@14h)) is selected, the FIFO is set to bypass mode and stays in this condition until a trigger event is generated. When this happens FIFO automatically switches to continuous (dynamic-stream) mode. The trigger event is set through the INTERRUPT_CFG(@0Bh) register.

If the interrupt is triggered, the IA bit, INT_SOURCE(@24h) is equal to 1, and the FIFO switches from bypass to continuous (dynamic-stream) mode and stays in it until an action on FIFO is taken. FIFO doesn't automatically switch back to bypass mode if the interrupt is de-asserted (IA bit, INT_SOURCE(@24h) goes back to 0). An action on FIFO has to be taken according to the continuous mode description.

Bypass-to-continuous can be used in order to start the acquisition when the configured interrupt is generated.

Table 17. Settings for FIFO buffer bypass-to-continuous mode

Register	@address	Bit range	Value	Mask	Mode
FIFO_CTRL	14h	TRIG_MODES & F_MODE[1:0]	110	00000111b = 07h	Bypass-to-continuous mode

9.3.2.3 **Continuous (dynamic-stream)-to-FIFO mode**

In continuous (dynamic-stream)-to-FIFO mode (TRIG_MODES & F_MODE[1:0] = 111, FIFO_CTRL(@14h)), the FIFO buffer works in continuous (dynamic-stream) behavior until a trigger event is generated (IA signal), then it switches to FIFO mode behavior. The trigger event is set through INTERRUPT_CFG(@0Bh). If the interrupt is triggered, the bit IA = 1, INT_SOURCE(@24h) is asserted, and the FIFO switches from continuous (dynamic-stream) to FIFO mode behavior. When the interrupt is de-asserted, the IA = 0, INT_SOURCE(@24h), the FIFO doesn't automatically switch back to continuous (dynamic-stream) behavior: an action to it has to be taken according to the continuous (dynamic-stream) behavior description.

Table 18. Settings for FIFO buffer continuous-to-FIFO mode

Register	@address	Bit range	Value	Mask	Mode
FIFO_CTRL	14h	TRIG_MODES & F_MODE[1:0]	111	00000111b = 07h	Continuous-to-FIFO mode

9.4 Retrieving data from FIFO

The action of consuming data from FIFO is performed by reading FIFO output registers. Data sample sets in FIFO are readable on a one-by-one basis, through the FIFO_DATA_OUT_PRESS_x registers, from address @78h to address @7Ah.

Every time a dss is read from the FIFO, the remaining oldest entry is placed in the FIFO_DATA_OUT_PRESS_x registers and status registers are updated accordingly. The FIFO_STATUS1 register content reports the number of remaining dss currently stored in FIFO.

Data can be retrieved from FIFO using every read byte combination in order to increase application flexibility (for example: 128 reads of 3 bytes, 1 multiple read of 384 bytes, and so forth). It is recommended to read all FIFO slots in a multiple byte read of 384 bytes (3 output registers by 128 slots).

FIFO output registers (FIFO_DATA_OUT_PRESS_x) support the automatic wraparound feature. The automatic wraparound feature allows executing register multi-reads without the need to manage the addresses, but it specifies the starting address and the number of bytes to be read. It defines the process of multiple reads where the address is automatically updated by the interface (I²C / MIPI I3CSM / SPI) and it rolls back to 78h when register 7Ah is reached (see [Section 5.1 Multi-read / write automatic address increment and automatic wraparound features](#)).

A single complete dss can be read by a 3-byte multi-read operation executed against the FIFO_DATA_OUT_PRESS_x registers (starting @78h). It returns the 3 bytes of the current oldest dss stored in FIFO.

Similarly, M (M=1,2, ..., 128) data sets can be extracted from FIFO with a single bus transaction through an M*3-byte multi-read operation executed against the FIFO_DATA_OUT_PRESS_x registers starting from FIFO_DATA_OUT_PRESS_XL(@78h). It is performed leveraging on the automatic address increment and automatic wraparound features, which are enabled by setting bit IF_ADD_INC=1, CTRL_REG3(@12h).

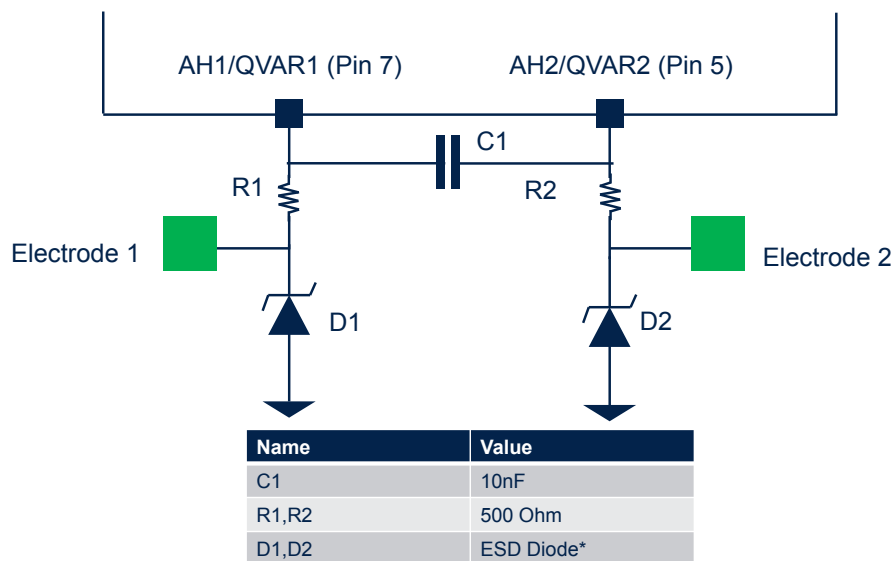
To read all the FIFO levels (128 data sample sets) stored in a full FIFO by a multiple-read operation, 384 bytes (3 output registers by 128 levels) have to be read, starting at FIFO_DATA_OUT_PRESS_XL (@78h).

10 Analog hub and Qvar functionality

The ILPS22QS embeds an analog hub (AH) chain which provides the capability to connect an external analog input and convert it to a digital signal for processing.

The same analog chain can be used to implement the Qvar (electric charge variation detection) functionality by connecting external electrodes and an electronic signal conditioning circuit, as illustrated in Figure 17. Additional details about Qvar functionality are available in the application note AN5755 on www.st.com.

Figure 17. Qvar external connections to pin 5, pin 7



*ST ESDALCL5-1BM2 is referenced as an ST catalog product but other similar features of ESD diodes also can be used.

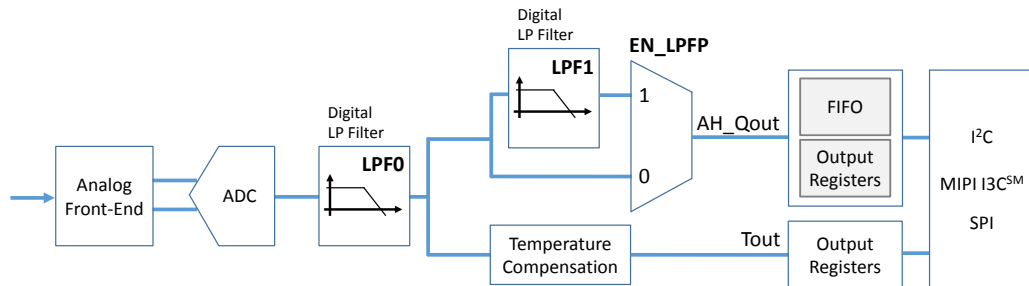
Note: If the analog hub / Qvar feature is not intended to be used, in order to disable it (and save power consumption), the register at address 5Fh must be written to 00h whenever the application is powered on. Furthermore, AH1/QVAR1 (pin 7) and AH2/QVAR2 (pin 5) must be connected to GND on the PCB.

Note: The input impedance between AH1/QVAR1 (pin 7) and AH2/QVAR2 (pin 5) of ILPS22QS depends on the AVG and ODR configuration (see ILPS22QS datasheet for more details).

In the ILPS22QS, the pressure sensor and the analog hub / Qvar share the same sensing chain (described in Section 4 Sampling chain), so the pressure sensor and the analog hub / Qvar output values cannot be read in parallel at the same time.

The analog hub / Qvar sensing channel can be activated by setting the AH_QVAR_EN bit of the CTRL_REG3(@12h) register to 1. With this configuration, the ILPS22QS filtering chain corresponds to the one shown in Figure 18. Basically, when the AH / Qvar is enabled, the operating parameters (like ODR, AVG, ...), the operating modes (one-shot mode or continuous mode), the interrupt features as well as the FIFO functionalities described in the other sections of this application note apply to the AH / Qvar data instead of the pressure sensor data. Also the output registers PRESS_OUT_x (@28h-@2Ah) and the FIFO output registers FIFO_OUTPUT (@78h-@7Ah) contain the AH / Qvar output data and the AH / Qvar data-ready signal is represented by the P_DA bit, STATUS(@27h) register.

Note: When the analog hub / Qvar sensing chain is activated, the RPDS registers, used in one-point calibration (OPC) of pressure to store the residual offset value to be removed, are automatically bypassed and their value is not applied to the AH / Qvar data and to the interrupt logic. No need to reset them to 0 when the AH / Qvar functionality is enabled, they can keep storing the value needed for pressure data calibration.

Figure 18. ILPS22QS filtering chain with AH / Qvar enabled


When the AH_QVAR_EN bit is configured to 0, the sampling chain processes the pressure sensor data and corresponds to the chain shown in Figure 5.

10.1 Analog hub / Qvar startup sequence and output data

To start the generation of analog hub / Qvar output data, it is necessary to select one of the operating modes and enable the AH / Qvar feature. The following general-purpose sequence can be used:

1. Write CTRL_REG3(@12h) = 80h // Enable AH / Qvar functionality
2. Write CTRL_REG1(@10h) = 27h // ODR = 25 Hz, AVG = 512

Similarly to the pressure sensor case, the read operation of the AH / Qvar output registers is performed as follows:

1. Read STATUS(@27h).
2. If P_DA = 0, then go to 1.
3. Read PRESS_OUT_XL(@28h).
4. Read PRESS_OUT_L(@29h).
5. Read PRESS_OUT_H(@2Ah).
6. Data processing
7. Go to 1.

If the device is configured in one-shot mode instead of continuous mode, the routine hangs at step 1 after one execution, since the device performs a single measurement, sets the P_DA bit high and returns to power-down mode. Note that the ONE_SHOT bit is self-cleared when the device returns to power-down mode. It is possible to trigger another one-shot read by setting the ONE_SHOT bit to 1 again.

The complete AH / Qvar data is given by the concatenation PRESS_OUT_H & PRESS_OUT_L & PRESS_OUT_XL and it is expressed as a binary number. AH / Qvar data is represented as 24-digit signed 2's complement binary numbers, (each digit referred to as LSB).

For the ILPS22QS device, in order to convert the AH / Qvar digital representation to its corresponding value in [mV], the value of Qvar_Gain of this sensor (438000 [LSB/mV], typical) is used in the following formula:

$$AH_Q_{out}[mV] = AH_Q_{out} [LSB] / Qvar_Gain$$

10.2 Analog hub / Qvar and pressure hardware interleaved mode

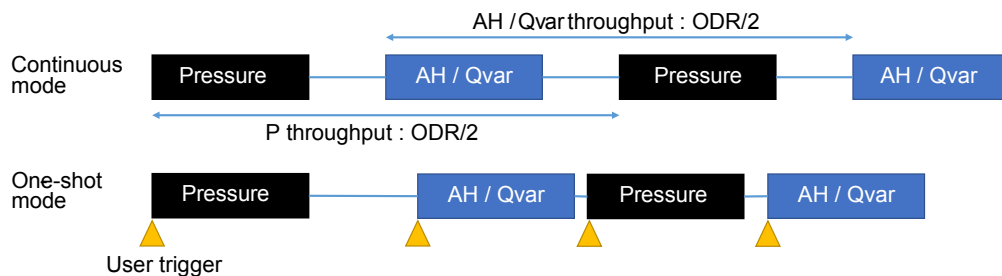
The ILPS22QS device supports the automatic generation of pressure and analog hub / Qvar data in interleaved mode. This functionality is enabled when the AH_QVAR_P_AUTO_EN bit in the CTRL_REG3 (12h) register is set to 1. In this configuration, the device alternately generates and makes the pressure data and the AH / Qvar data available in the output registers PRESS_OUT_x (@28h-@2Ah).

Note: Before enabling the analog hub / Qvar and pressure hardware interleaved mode, the device must be configured in power-down mode and the AH_QVAR_EN bit in the CTRL_REG3 (12h) register must be set to 0.

When this mode is enabled, the LPF1 filter, the FIFO buffer, and the AUTOZERO / AUTOREFP mode are applied to the pressure sensor chain and can be used as described in the previous sections of this application note.

This mode can be enabled in both continuous mode and in one-shot mode. In continuous mode, the actual output data rate is halved compared to the value set in the bit range ODR[3:0] of the CTRL_REG1 (10h) register. For the one-shot mode the (typical) data conversion time and the limitations (maximum ODR vs. AVG[2:0] parameter) described in Section 3.2 are still valid.

Figure 19. Analog hub / Qvar and pressure hardware interleaved mode



When the analog hub / Qvar and pressure hardware interleaved mode is enabled, a tag is applied to the output data available in the output registers PRESS_OUT_x (@28h-@2Ah). The least significant bit is set to 0 for pressure data and to 1 for AH / Qvar data, as described in the following figure.

Figure 20. Analog hub / Qvar and pressure data tag



10.2.1 Analog hub / Qvar and pressure hardware interleaved mode in FIFO

In the ILPS22QS device, the analog hub / Qvar and pressure hardware interleaved mode can be used in combination with the FIFO features to store the analog hub / Qvar data and the pressure data in the FIFO buffer.

This functionality is enabled when the both the AH_QVAR_P_AUTO_EN bit in the CTRL_REG3 (12h) register and the AH_QVAR_P_FIFO_EN bit in the FIFO_CTRL (14h) register are set to 1.

In this configuration, the device alternately generates and stores the pressure data and the AH / Qvar data in the FIFO buffer. A tag is applied to the output data stored in FIFO. The least significant bit is set to 0 for pressure data and to 1 for AH / Qvar data, as shown in Figure 20. As described in Section 9 First-in, first-out (FIFO) buffer, the data stored in FIFO can be retrieved by reading the FIFO_DATA_OUT_PRESS_x (from 78h to 7Ah) registers.

Note: Before enabling the analog hub / Qvar and pressure hardware interleaved mode, the device must be configured in power-down mode and the AH_QVAR_EN bit in the CTRL_REG3 (12h) register must be set to 0.

10.3 Analog hub / Qvar current consumption

As described in [Section 3.4 Pressure sensor noise and current consumption](#), the power consumption of the ILPS22QS mainly depends on the data rate (ODR parameter) and on the selected resolution (AVG parameter). This is valid also when the analog hub / Qvar functionality is enabled.

When the AH_QVAR_EN bit of the CTRL_REG3(@12h) register is set to 1, only the analog hub / Qvar sensing channel is activated (hardware interleaved mode is disabled). [Table 19](#) summarizes the ILPS22QS sensor typical consumption of the AH / Qvar channel for different configurations of the ODR and the resolution in continuous mode. [Table 20](#) indicates the corresponding current consumption values obtained when using the ILPS22QS sensor in one-shot mode.

Note: The values indicated in the two tables below refer to the condition of the analog hub / Qvar feature enabled (the register 5Fh was not set to 00h) and analog hub / Qvar chain enabled (AH_QVAR_EN bit is set to 1 and AH_QVAR_P_AUTO_EN bit is set to 0 in the CTRL_REG3 (12h) register).

Table 19. Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (AH / Qvar enabled)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	22.6	76.4	184.5	458.6	-	-	-	-
128	9.3	24.1	53.9	128.4	253.4	379.0	-	-
64	7.2	15.4	32.1	73.6	143.2	213.1	283.1	-
32	6.1	11.1	21.3	46.5	88.6	131.0	173.4	343.4
16	5.5	8.9	15.7	32.8	61.4	90.0	118.7	233.4
8	5.0	6.9	10.7	20.0	35.8	51.4	67.1	130.2
4	4.9	6.4	9.3	16.6	28.9	41.2	53.5	102.7

Table 20. Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (AH / Qvar enabled)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	21.9	75.7	183.8	457.9	-	-	-	-
128	8.6	23.4	53.2	127.7	252.7	378.3	-	-
64	6.5	14.7	31.4	72.9	142.5	212.4	282.4	-
32	5.4	10.4	20.6	45.8	87.9	130.3	172.7	342.7
16	4.8	8.2	15.0	32.1	60.7	89.3	118.0	232.7
8	4.3	6.2	10.0	19.3	35.1	50.7	66.4	129.5
4	4.2	5.7	8.6	15.9	28.2	40.5	52.8	102.0

If the analog hub / Qvar and pressure hardware interleaved mode is enabled, by setting the AH_QVAR_P_AUTO_EN bit to 1 in the CTRL_REG3 (12h), the pressure sensor chain and the AH / Qvar chain are alternately enabled / disabled and the resulting current consumption is the one indicated in [Table 21](#) (continuous mode) and [Table 22](#) (one-shot mode).

Note: The values indicated in the two tables below refer to the condition of the analog hub / Qvar feature enabled (the register 5Fh was not set to 00h) and the analog hub / Qvar and pressure hardware interleaved mode enabled (AH_QVAR_EN bit is set to 0 and AH_QVAR_P_AUTO_EN bit is set to 1 in the CTRL_REG3 (12h) register).

Table 21. Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (interleaved mode)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	27.7	101.6	249.5	621.2	-	-	-	-
128	9.7	29.9	70.3	171.4	340.2	509.4	-	-
64	6.8	17.9	40.4	96.5	190.2	284.1	378.	-
32	5.3	12.0	25.6	59.2	115.4	171.8	228.2	453.9
16	4.5	9.0	18.0	40.5	78.1	115.7	153.4	303.7
8	3.9	6.5	11.7	24.6	46.2	67.8	89.3	176.0
4	3.7	5.7	9.8	19.9	36.8	53.7	70.7	138.4

Table 22. Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (interleaved mode)

AVG	ODR = 1 Hz	ODR = 4 Hz	ODR = 10 Hz	ODR = 25 Hz	ODR = 50 Hz	ODR = 75 Hz	ODR = 100 Hz	ODR = 200 Hz
512	27.1	101.0	248.8	620.6	-	-	-	-
128	9.0	29.2	69.7	170.7	339.6	508.8	-	-
64	6.1	17.2	39.7	95.8	189.5	283.4	377.3	-
32	4.6	11.3	24.9	58.5	114.7	171.1	227.5	453.2
16	3.8	8.2	17.2	39.8	77.4	115.0	152.6	303.0
8	3.2	5.8	11.0	23.9	45.5	67.1	88.6	175.3
4	3.0	5.0	9.0	19.2	36.1	53.0	69.9	137.6

Revision history

Table 23. Document revision history

Date	Version	Changes
16-Mar-2022	1	Initial release

Contents

1	Pin description	2
2	Registers	3
3	Operating modes	4
3.1	Power-down mode	4
3.2	One-shot mode	5
3.3	Continuous mode	6
3.4	Pressure sensor noise and current consumption	7
4	Sampling chain	9
4.1	Digital low-pass filters	10
4.2	Data path	11
5	Reading output data	12
5.1	Multi-read / write automatic address increment and automatic wraparound features	12
5.1.1	Automatic address increment feature	12
5.1.2	Automatic address wraparound feature	13
5.2	Startup sequence	13
5.3	Using the status register	13
5.4	Using the block data update (BDU) feature	14
5.5	Understanding output data	15
5.5.1	Pressure data	15
5.5.2	Example of pressure data	15
5.5.3	Temperature data	16
5.5.4	Example of temperature data	16
6	Reboot and software reset	17
7	Offset compensation (OPC - one-point calibration)	18
8	Interrupt modes	19
8.1	Interrupt events related to pressure and temperature sampling	19
8.1.1	Data-ready	19
8.1.2	Threshold-based differential interrupt	20
8.1.3	Interrupt events related to FIFO status	25
8.1.4	Interrupt events for FIFO triggered modes	25
9	First-in, first-out (FIFO) buffer	26
9.1	FIFO description	26
9.2	FIFO settings and control	27
9.2.1	Limiting FIFO depth: stop-on-watermark level	27

9.2.2	Monitoring the FIFO buffer status	28
9.3	FIFO buffer modes	29
9.3.1	Non-triggered FIFO modes	29
9.3.2	Triggered FIFO modes	32
9.4	Retrieving data from FIFO	34
10	Analog hub and Qvar functionality	35
10.1	Analog hub / Qvar startup sequence and output data	36
10.2	Analog hub / Qvar and pressure hardware interleaved mode	37
10.2.1	Analog hub / Qvar and pressure hardware interleaved mode in FIFO	37
10.3	Analog hub / Qvar current consumption	38
	Revision history	40
	List of tables	43
	List of figures	44

List of tables

Table 1.	Pin list, functions, and internal status	2
Table 2.	Registers	3
Table 3.	Typical conversion time and maximum ODR in one-shot mode	5
Table 4.	ODR selection	6
Table 5.	Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (AH / Qvar disabled)	7
Table 6.	Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (AH / Qvar disabled)	7
Table 7.	Typical RMS noise [Pa] at VDD = 1.8 V, T = 25 °C	8
Table 8.	Related registers and bit ranges of the LP filters	10
Table 9.	Settings of the LP filters	10
Table 10.	List of registers involved in managing FIFO	27
Table 11.	FIFO modes and FIFO trigger mode selection	27
Table 12.	Watermark settings	27
Table 13.	FIFO buffer status monitoring registers and bit ranges	28
Table 14.	Settings for FIFO buffer bypass mode	29
Table 15.	Settings for FIFO buffer FIFO mode	30
Table 16.	Settings for FIFO buffer bypass-to-FIFO mode	32
Table 17.	Settings for FIFO buffer bypass-to-continuous mode	33
Table 18.	Settings for FIFO buffer continuous-to-FIFO mode	33
Table 19.	Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (AH / Qvar enabled)	38
Table 20.	Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (AH / Qvar enabled)	38
Table 21.	Typical current consumption [μ A] in continuous mode at VDD = 1.8 V, T = 25 °C (interleaved mode)	39
Table 22.	Typical current consumption [μ A] in one-shot mode at VDD = 1.8 V, T = 25 °C (interleaved mode)	39
Table 23.	Document revision history	40

List of figures

Figure 1.	Pin connections	2
Figure 2.	VDD power-on/off sequence.	4
Figure 3.	One-shot mode.	5
Figure 4.	ILPS22QS architecture block diagram	9
Figure 5.	ILPS22QS digital LP filters block diagram	10
Figure 6.	Data path for output registers (standard and FIFO)	11
Figure 7.	Interrupt generator	20
Figure 8.	Differential interrupt input signal and threshold relationship.	21
Figure 9.	Differential interrupt AUTOZERO mode, outputs and thresholds	22
Figure 10.	Differential interrupt AUTOREFP mode, outputs and thresholds	23
Figure 11.	Latch disabled (LIR = 0): interrupt behavior	24
Figure 12.	Latch enabled (LIR = 1): interrupt behavior.	24
Figure 13.	FIFO bypass mode	29
Figure 14.	FIFO mode with disabled stop-on-watermark feature	30
Figure 15.	FIFO mode with enabled stop-on-watermark feature	30
Figure 16.	Bypass-to-FIFO mode	32
Figure 17.	Qvar external connections to pin 5, pin 7	35
Figure 18.	ILPS22QS filtering chain with AH / Qvar enabled	36
Figure 19.	Analog hub / Qvar and pressure hardware interleaved mode	37
Figure 20.	Analog hub / Qvar and pressure data tag	37

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2022 STMicroelectronics – All rights reserved