
STHS34PF80: low-power, high-sensitivity infrared (IR) sensor for presence and motion detection

Introduction

This document provides usage information and application hints related to ST's [STHS34PF80](#) infrared motion and presence detection sensor.

The STHS34PF80 is an uncooled, factory-calibrated IR sensor with operating wavelength between 5 μm and 20 μm , which is equipped with an I²C / 3-wire SPI interface.

The STHS34PF80 sensor has been designed to measure the amount of infrared radiation emitted from an object within its field of view. The information is digitally processed by the ASIC, which can be programmed to monitor motion, presence, and an overtemperature condition. Thanks to its exceptional infrared sensitivity of 2000 LSB/°C, the STHS34PF80 can detect human presence at a distance up to 4 meters without the need of an optical lens. The field of view guaranteed by the package is 80°.

The STHS34PF80 is housed in a small 3.2 x 4.2 x 1.455 (max) mm 10-lead organic LGA package compatible with SMD mounting, and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

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

1 Pin description

Figure 1. Pin configuration (package bottom view)

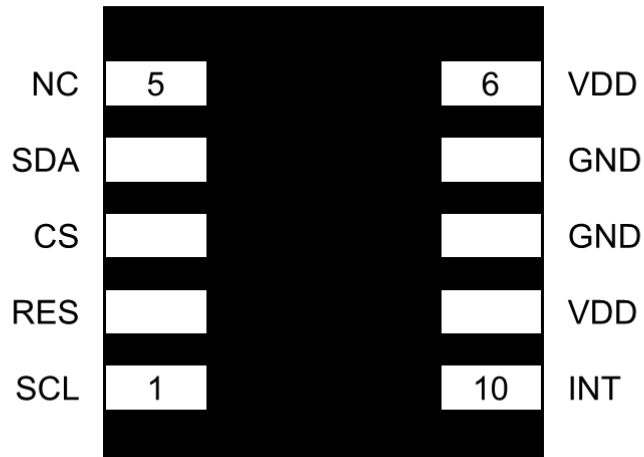


Table 1. Pin list, functions, and internal status

Pin number	Name	Function	Internal pin status
1	SCL	I ² C serial clock (SCL)	Default: input without pull-up
	SPC	SPI serial port clock (SPC)	
2	RES	Connect to GND	
3	CS	I ² C / SPI interface selection (1: I ² C enabled; 0: SPI enabled)	Default: input with pull-up
4	SDA	I ² C serial data (SDA)	Default: input without pull-up
	SDI/SDO	3-wire SPI serial data input/output (SDI/SDO)	
5	NC	Do not connect	
6	VDD	Power supply	
7	GND	0 V supply	
8	GND	0 V supply	
9	VDD	Power supply	
10	INT	Interrupt signal	Default: input without pull-up

2 Registers

Table 2. Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPF1	0Ch	-	-	LPF_P_M2	LPF_P_M1	LPF_P_M0	LPF_M2	LPF_M1	LPF_M0
LPF2	0Dh	-	-	LPF_P2	LPF_P11	LPF_P0	LPF_A_T2	LPF_A_T1	LPF_A_T0
WHO_AM_I ⁽¹⁾	0Fh	1	1	0	1	0	0	1	1
AVG_TRIM	10h	0	0	AVG_T1	AVG_T0	0	AVG_TMOS2	AVG_TMOS1	AVG_TMOS0
CTRL0	17h	1	GAIN2	GAIN1	GAIN0	0	0	0	1
SENS_DATA	1Dh	SENS7	SENS6	SENS5	SENS4	SENS3	SENS2	SENS1	SENS0
CTRL1	20h	0	-	-	BDU	ODR3	ODR2	ODR1	ODR0
CTRL2	21h	BOOT	-	-	FUNC_CFG_ACCESS	-	0	0	ONE_SHOT
CTRL3	22h	INT_H_L	PP_OD	INT_MSK2	INT_MSK1	INT_MSK0	INT_LATCHED	IEN1	IEN0
STATUS ⁽¹⁾	23h	-	-	-	-	-	DRDY	-	-
FUNC_STATUS ⁽¹⁾	25h	-	-	-	-	-	PRES_FLAG	MOT_FLAG	TAMB_SHOCK_FLAG
TOBJECT_L ⁽¹⁾	26h	TOBJECT7	TOBJECT6	TOBJECT5	TOBJECT4	TOBJECT3	TOBJECT2	TOBJECT1	TOBJECT0
TOBJECT_H ⁽¹⁾	27h	TOBJECT15	TOBJECT14	TOBJECT13	TOBJECT12	TOBJECT11	TOBJECT10	TOBJECT9	TOBJECT8
TAMBIENT_L ⁽¹⁾	28h	TAMBIENT7	TAMBIENT6	TAMBIENT5	TAMBIENT4	TAMBIENT3	TAMBIENT2	TAMBIENT1	TAMBIENT0
TAMBIENT_H ⁽¹⁾	29h	TAMBIENT15	TAMBIENT14	TAMBIENT13	TAMBIENT12	TAMBIENT11	TAMBIENT10	TAMBIENT9	TAMBIENT8
TOBJ_COMP_L ⁽¹⁾	38h	TOBJ_COMP7	TOBJ_COMP6	TOBJ_COMP5	TOBJ_COMP4	TOBJ_COMP3	TOBJ_COMP2	TOBJ_COMP1	TOBJ_COMP0
TOBJ_COMP_H ⁽¹⁾	39h	TOBJ_COMP15	TOBJ_COMP14	TOBJ_COMP13	TOBJ_COMP12	TOBJ_COMP11	TOBJ_COMP10	TOBJ_COMP9	TOBJ_COMP8
TPRESENCE_L ⁽¹⁾	3Ah	TPRESENCE7	TPRESENCE6	TPRESENCE5	TPRESENCE4	TPRESENCE3	TPRESENCE2	TPRESENCE1	TPRESENCE0
TPRESENCE_H ⁽¹⁾	3Bh	TPRESENCE15	TPRESENCE14	TPRESENCE13	TPRESENCE12	TPRESENCE11	TPRESENCE10	TPRESENCE9	TPRESENCE8
TMOTION_L ⁽¹⁾	3Ch	TMOTION7	TMOTION6	TMOTION5	TMOTION4	TMOTION3	TMOTION2	TMOTION1	TMOTION0
TMOTION_H ⁽¹⁾	3Dh	TMOTION15	TMOTION14	TMOTION13	TMOTION12	TMOTION11	TMOTION10	TMOTION9	TMOTION8
TAMB_SHOCK_L ⁽¹⁾	3Eh	TAMB_SHOCK7	TAMB_SHOCK6	TAMB_SHOCK5	TAMB_SHOCK4	TAMB_SHOCK3	TAMB_SHOCK2	TAMB_SHOCK1	TAMB_SHOCK0
TAMB_SHOCK_H ⁽¹⁾	3Fh	TAMB_SHOCK15	TAMB_SHOCK14	TAMB_SHOCK13	TAMB_SHOCK12	TAMB_SHOCK11	TAMB_SHOCK10	TAMB_SHOCK9	TAMB_SHOCK8

1. Read-only register



2.1 Embedded functions registers

The embedded functions registers are accessible when the FUNC_CFG_ACCESS bit in the CTRL2 (21h) register is set to 1. These registers can be used to configure the embedded algorithm for compensating ambient temperature variations in the object temperature (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)) and the embedded smart digital algorithms (see [Section 7 Embedded smart digital algorithms](#)).

The FUNC_CFG_ADDR (08h), FUNC_CFG_DATA (09h), and PAGE_RW (11h) registers are used to configure the procedures for reading/writing all the other embedded functions registers (see [Section 2.1.1 Write procedure for embedded functions registers](#) and [Section 2.1.2 Read procedure for embedded functions registers](#)).

Table 3. Embedded functions registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNC_CFG_ADDR	08h	FUNC_CFG_ADDR7	FUNC_CFG_ADDR6	FUNC_CFG_ADDR5	FUNC_CFG_ADDR4	FUNC_CFG_ADDR3	FUNC_CFG_ADDR2	FUNC_CFG_ADDR1	FUNC_CFG_ADDR0
FUNC_CFG_DATA	09h	FUNC_CFG_DATA7	FUNC_CFG_DATA6	FUNC_CFG_DATA5	FUNC_CFG_DATA4	FUNC_CFG_DATA3	FUNC_CFG_DATA2	FUNC_CFG_DATA1	FUNC_CFG_DATA0
PAGE_RW	11h	0	FUNC_CFG_WRITE	FUNC_CFG_READ	0	-	-	-	-
PRESENCE_THS_L	20h	PRESENCE_THS7	PRESENCE_THS6	PRESENCE_THS5	PRESENCE_THS4	PRESENCE_THS3	PRESENCE_THS2	PRESENCE_THS1	PRESENCE_THS0
PRESENCE_THS_H	21h	-	PRESENCE_THS14	PRESENCE_THS13	PRESENCE_THS12	PRESENCE_THS11	PRESENCE_THS10	PRESENCE_THS9	PRESENCE_THS8
MOTION_THS_L	22h	MOTION_THS7	MOTION_THS6	MOTION_THS5	MOTION_THS4	MOTION_THS3	MOTION_THS2	MOTION_THS1	MOTION_THS0
MOTION_THS_H	23h	-	MOTION_THS14	MOTION_THS13	MOTION_THS12	MOTION_THS11	MOTION_THS10	MOTION_THS9	MOTION_THS8
TAMB_SHOCK_THS_L	24h	TAMB_SHOCK_THS7	TAMB_SHOCK_THS6	TAMB_SHOCK_THS5	TAMB_SHOCK_THS4	TAMB_SHOCK_THS3	TAMB_SHOCK_THS2	TAMB_SHOCK_THS1	TAMB_SHOCK_THS0
TAMB_SHOCK_THS_H	25h	-	TAMB_SHOCK_THS14	TAMB_SHOCK_THS13	TAMB_SHOCK_THS12	TAMB_SHOCK_THS11	TAMB_SHOCK_THS10	TAMB_SHOCK_THS9	TAMB_SHOCK_THS8
HYST_MOTION	26h	HYST_MOTION7	HYST_MOTION6	HYST_MOTION5	HYST_MOTION4	HYST_MOTION3	HYST_MOTION2	HYST_MOTION1	HYST_MOTION0
HYST_PRESENCE	27h	HYST_PRESENCE7	HYST_PRESENCE6	HYST_PRESENCE5	HYST_PRESENCE4	HYST_PRESENCE3	HYST_PRESENCE2	HYST_PRESENCE1	HYST_PRESENCE0
ALGO_CONFIG	28h	-	-	-	0	INT_PULSED	COMP_TYPE	SEL_ABS	0
HYST_TAMB_SHOCK	29h	HYST_TAMB_SHOCK7	HYST_TAMB_SHOCK6	HYST_TAMB_SHOCK5	HYST_TAMB_SHOCK4	HYST_TAMB_SHOCK3	HYST_TAMB_SHOCK2	HYST_TAMB_SHOCK1	HYST_TAMB_SHOCK0
RESET_ALGO	2Ah	-	-	-	-	-	-	-	ALGO_ENABLE_RESET



2.1.1 Write procedure for embedded functions registers

Write procedure example: write value YYh in the embedded functions register at address XXh.

1. Enter power-down mode (see [Section 3.1 Power-down mode](#)), if not already in it
2. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
3. Write bit FUNC_CFG_WRITE = 1 in PAGE_RW (11h) // Select write operation mode
4. Write XXh in FUNC_CFG_ADDR (08h) // Set address XXh of the embedded functions register to be written
5. Write YYh in FUNC_CFG_DATA (09h) // Set value YYh to be written
6. Write bit FUNC_CFG_WRITE = 0 in PAGE_RW (11h) // Disable write operation mode
7. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
8. Enter continuous mode (see [Section 3.3 Continuous mode](#)) or one-shot mode (see [Section 3.2 One-shot mode](#))

Note: *When changing one or more of the parameters of the smart digital algorithms through the embedded functions registers, it is necessary to perform the procedure detailed in [Section 7.4 Resetting the algorithms for the changes to take effect](#).*

2.1.2 Read procedure for embedded functions registers

Read procedure example: read the value of the embedded functions register at address XXh.

1. Enter power-down mode (see [Section 3.1 Power-down mode](#)), if not already in it
2. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
3. Write bit FUNC_CFG_READ = 1 in PAGE_RW (11h) // Select read operation mode
4. Write XXh in FUNC_CFG_ADDR (08h) // Set address XXh of the embedded functions register to be read
5. Read the value of FUNC_CFG_DATA (09h) // Get register value
6. Write bit FUNC_CFG_READ = 0 in PAGE_RW (11h) // Disable read operation mode
7. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
8. Enter continuous mode (see [Section 3.3 Continuous mode](#)) or one-shot mode (see [Section 3.2 One-shot mode](#))

Note: *If a read or write procedure involves multiple registers, only steps 4 and 5 have to be performed repeatedly. Moreover, if a write procedure involves multiple consecutive registers, only step 5 has to be performed repeatedly.*

3 Operating modes

The STHS34PF80 features three operating modes:

- Power-down mode
- One-shot mode
- Continuous mode

When configured in one-shot mode, the device can read environmental data at the very moment the controlling MCU requires it. When configured in continuous mode, the device keeps reading data at predefined frequencies (fixed output data rates, ODRs).

In both one-shot mode and continuous mode, the STHS34PF80 allows performing prior averaging (filtering) of the values of ambient temperature and object temperature to obtain smoother outputs. The AVG_T[1:0] and AVG_TMOS[2:0] bit ranges of the AVG_TRIM (10h) register select the number of averaged samples for the ambient temperature and the object temperature filtering, respectively, as can be seen in the following tables. [Table 5](#) also shows the typical RMS noise of the object temperature based on the selected number of averaged samples. The data found in the output registers are already averaged.

Table 4. Averaging selection for ambient temperature

AVG_T[1:0]	Number of averaged samples for ambient temperature
00	8 (default)
01	4
10	2
11	1

Table 5. Averaging selection for object temperature and noise

AVG_TMOS [2:0]	Number of averaged samples for object temperature	RMS noise (LSB _{rms})
000	2	90
001	8	50
010	32	25
011	128 (default)	20
100	256	15
101	512	12
110	1024	11
111	2048	10

The device offers a wide VDD voltage range from 1.7 V to 3.6 V. 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, until VDD is set. After VDD is set, the lines connected to the IO pins have to be configured according to their default status described in [Table 1. Pin list, functions, and internal status](#).

After the power-supply has been applied and the boot procedure has been completed (see [Section 5 Boot and reboot](#)), 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 a resting 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 STHS34PF80 can reach its lowest power consumption achievable while power is supplied. While in power-down mode, the I²C / 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.

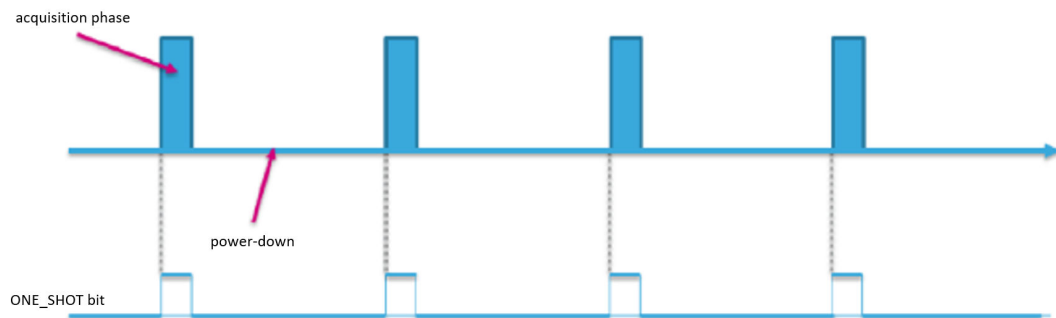
The device is in power-down mode when the ODR[3:0] bit range of the CTRL1 (20h) register is set to 0000. The correct procedure to enter power-down mode from continuous mode in order to avoid reading wrong output data from the registers when re-entering continuous mode is the following:

1. Read the FUNC_STATUS (25h) register // This is done to reset the DRDY bit of the STATUS (23h) register to 0
2. Wait that the DRDY bit of the STATUS (23h) register is set to 1
3. Set the ODR[3:0] bits of the CTRL1 (20h) register // This is to be carried out as soon as possible after the DRDY bit of the STATUS (23h) register is set to 1
4. Read the FUNC_STATUS (25h) register // This is done to reset the DRDY bit of the STATUS (23h) register to 0 before re-entering continuous mode

3.2 One-shot mode

One-shot mode is used to execute single data acquisitions at a desired rate. One-shot mode must be executed while the device is in power-down mode by setting the ONE_SHOT bit (default value 0) in the CTRL2 (21h) register to 1. When this happens, a single acquisition of a set of data (ambient temperature and object temperature data) 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). See [Section 4 Reading output data](#) for how to interpret output data.

Figure 2. One-shot mode



One-shot mode requires the ODR[3:0] bit range of the CTRL1 (20h) register to be set to 0000. Power consumption in this mode depends on the frequency at which the ONE_SHOT bit is set to 1 by the microcontroller/application processor.

The typical time needed for the generation of the new data (which the maximum data rate achievable in one-shot mode depends on) is given in [Table 6. Typical conversion time and maximum data rate in one-shot mode](#), and strictly depends on the value selected for the average parameter through the AVG_TMOS[2:0] bit range of the AVG_TRIM (10h) register. These values apply if the AVG_T[1:0] bit range of the AVG_TRIM register is set to 00 (default value), giving 8 as the number of averaged samples for ambient temperature.

Table 6. Typical conversion time and maximum data rate in one-shot mode

AVG_TMOS[2:0]	Number of averaged samples for object temperature	Typical data conversion time [ms]	Maximum data rate [Hz]
000	2	5.5	164
001	8	6.3	145
010	32	9.1	99
011	128 (default)	20.7	44
100	256	36.1	25
101	512	67.0	14
110	1024	128.6	7
111	2048	252.0	3.5

One-shot mode is not compatible with the embedded smart digital algorithms for presence detection, motion detection, and ambient temperature shock detection. However, it is compatible with the embedded compensation algorithm (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)).

3.3 Continuous mode

Continuous mode is designed to keep sampling data at a specific predefined selectable output data rate (ODR). The output registers are updated with new readings every given period according to the selected ODR frequency (see [Table 7. ODR configuration](#)). Continuous mode ODR selection is made through the ODR[3:0] bit range of the CTRL1 (20h) register. When the ODR[3:0] bit range is set to a value different than 0000 (power-down mode), the device enters continuous mode and immediately starts sampling data and putting them in the output registers at the selected frequency. See [Section 4 Reading output data](#) for how to read and interpret output data.

Table 7. ODR configuration

ODR [3:0]	ODR frequency [Hz]	Sampling period [ms]
0000	Power-down mode	-
0001	0.25	4000
0010	0.5	2000
0011	1	1000
0100	2	500
0101	4	250
0110	8	126
0111	15	66.67
1xxx	30	33.33

When entering continuous mode from power-down mode, a reset of the embedded smart digital algorithms must be performed. The complete procedure is the following:

1. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
2. Write bit FUNC_CFG_WRITE = 1 in PAGE_RW (11h) // Enable write operation
3. Write 2Ah in FUNC_CFG_ADDR (08h) // Set register address (RESET_ALGO register)
4. Write 01h in FUNC_CFG_DATA (09h) // Write register value (set ALGO_ENABLE_RESET bit)
5. Write bit FUNC_CFG_WRITE = 0 in PAGE_RW (11h) // Disable write operation
6. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
7. Write bits ODR[3:0] in CTRL1 (20h) with the desired value // Start acquiring data at the desired ODR value

It is not possible to change the ODR value seamlessly. In order to change the ODR, it is necessary first to enter in power-down mode (see [Section 3.1 Power-down mode](#)) and then to perform the procedure detailed above for entering continuous mode with the new ODR value.

The reset procedure (and therefore the ODR change, too) should be carried out only when the field of view of the sensor is empty, otherwise the outputs of the algorithms could be impacted. The procedure for resetting the algorithms is described separately in [Section 7.4 Resetting the algorithms](#).

In continuous mode, device power consumption depends on the averaging selection for object temperature (AVG_TMOS[2:0] bit range) and on the selected ODR, as shown in [Table 8. Current consumption at different ODRs and AVG_TMOS setting \(@ V_{DD} = 1.8 V, T = 25 °C\)](#). In continuous mode, for each AVG_TMOS[2:0] value, a maximum ODR value can be set (see the table below, where a blank space means incompatibility). If the user sets an ODR value not compatible with the chosen number of averaging, the highest ODR compatible with that number is automatically set instead. The values of the following table apply if the AVG_T[1:0] bit range of the AVG_TRIM register is set to 00 (default value), giving 8 as the number of averaged samples for ambient temperature.

Table 8. Current consumption at different ODRs and AVG_TMOS setting (@ V_{DD} = 1.8 V, T = 25 °C)

AVG_TMOS [2:0]	One-shot mode	Continuous mode – current consumption (µA) vs. ODR							
	Current consumption (µA) @ 1Hz	0.25 Hz	0.5 Hz	1 Hz	2 Hz	4 Hz	8 Hz	15 Hz	30 Hz
000 (2)	3.23	3.2	3.52	4.39	6.58	10.54	18.32	33.45	64.50
001 (8)	3.74	3.27	3.82	4.9	7.23	11.4	20.62	38.3	74.94
010 (32)	5	3.6	4.48	6.26	9.58	17.05	30.75	59	115.65
011 (128)	10	4.89	7.07	11.44	19.65	37.25	71.85		
100 (256)	16.89	6.55	10.55	18.02	33.1	65.5			
101 (512)	31.16	10.05	17.45	32.25	59.90				
110 (1024)	56.34	16.97	31.3	57.60					
111 (2048)	113	30.86	58.97						

4 Reading output data

4.1 Startup sequence

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 is completed (that is, after 2.5 ms from power-on), the sensor automatically enters power-down mode. Refer to [Section 5 Boot and reboot](#) for more information on the boot procedure.

To turn on the sensor and gather object temperature and ambient temperature data through the I²C / SPI interface, it is necessary to select one of the operating modes through the ODR[3:0] bit range in the CTRL1 (20h) register (continuous mode), or to set the ONE_SHOT bit to 1 in the CTRL2 (21h) register (one-shot mode).

The following general-purpose sequence can be used to configure the STHS34PF80 device:

1. Write 02h in AVG_TRIM (10h) // AVG_T = 8, AVG_TMOS = 32
2. Write 07h in CTRL1 (20h) // ODR = 15 Hz

4.2 Using the status register

The device is provided with a STATUS (23h) register, which can be polled to check when a new set of data is available. A set of data is defined as the combination of the object temperature output, the ambient temperature output, the outputs from the three smart digital algorithms (see [Section 4.5.4 Embedded smart digital algorithms data](#) for configuring them) and, if the embedded compensation algorithm is enabled (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)), also the compensated object temperature output.

The data-ready signals are represented by the DRDY bit of the STATUS register. This bit is set to 1 when a new set of data is available, and it is cleared when reading the FUNC_STATUS (25h) register. The FUNC_STATUS register contains the flags for presence detection, motion detection, and ambient temperature shock detection of the smart digital algorithms.

For object temperature data, the read of the output registers should be performed as follows:

1. Read the STATUS (23h) register.
2. If DRDY = 0, then go to 1.
3. Read FUNC_STATUS (25h) to reset the DRDY flag.
4. Read TOBJECT_L (26h).
5. Read TOBJECT_H (27h).
6. Data processing.
7. Go to 1.

The procedure to obtain ambient temperature data is equivalent.

If the device is configured in one-shot mode instead of continuous mode, the routine is pending at step 1 after one execution, since the device performs a single measurement, sets the DRDY 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.

4.3 Using the data-ready signal

The device can be configured to have a hardware signal to determine when a new set of measurement data is available to be read and trigger synchronous actions (for example, read output registers as soon as data are available).

The value of the DRDY bit of the STATUS (23h) register can be driven to the INT pin by setting the IEN[1:0] bit range of the CTRL3 (22h) register to 01. The DRDY signal acting on the INT pin is in pulsed mode by default. It is set to latched mode by setting the INT_LATCHED bit of the CTRL3 (22h) register to 1. In latched mode, the INT pin status is reset after reading the FUNC_STATUS (25h) register.

Note: The DRDY bit of the STATUS register is always in latched mode, and it is reset after reading the FUNC_STATUS (25h) register.

Refer to [Section 6 Interrupt generation](#) for more information on the interrupt generator block.

4.4 Using the block data update (BDU) feature

If reading the output data is not synchronized with the data-ready signal, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL1 (20h) register.

This feature avoids reading register values (most significant and least significant bytes of the output data) related to different samples. In particular, when the BDU is activated, the output data registers of ambient temperature (28h-29h) and of object temperature (26h-27h) always contain the most recent output data produced by the device but, in case the read of a given pair is initiated (for example, object temperature read is initiated by reading TOBJECT_L), the refresh for that pair is blocked until both registers are read.

BDU applies to both object temperature data and ambient temperature data but manages them separately. When the BDU feature is enabled, object and ambient temperature data are separately refreshed depending on when TOBJECT_H and TAMBIENT_H are read.

BDU does not apply to embedded smart digital algorithm data (see [Section 7 Embedded smart digital algorithms](#)) and compensated object temperature data (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)). In this case, reading the registers should be synchronous with the DRDY flag in order to avoid reading register values related to different samples.

Note: To guarantee the correct behavior of the BDU feature, TOBJECT_H (27h) and TAMBIENT_H (29h) must be the last address read when reading object temperature and ambient temperature.

4.5 Understanding output data

4.5.1 Object temperature data

The measured object temperature data are sent to the TOBJECT_H (27h) and TOBJECT_L (26h) registers. These registers contain, respectively, the most significant byte and the least significant byte of the object temperature value.

The complete output data is given by the concatenation of TOBJECT_H and TOBJECT_L. This object temperature value is represented as a 16-bit signed two's complement binary number (each digit is referred to as LSB).

The object temperature output data (in LSB) is proportional to the transmittance of the optical system in use with the sensor in the range of the operating wavelengths (between 5 μm and 20 μm). Each object temperature output sample can be divided by the actual value of sensitivity for object temperature data in order to obtain the corresponding value in °C (independent from the value of transmittance), as in the following formula:

$$TOBJECT[°C] = \frac{TOBJECT(\tau)[LSB]}{SENSITIVITY(\tau)\left[\frac{LSB}{°C}\right]} = \frac{TOBJECT(\tau)[LSB]}{\tau \times SENSITIVITY_{DEFAULT}\left[\frac{LSB}{°C}\right]}$$

where τ is the transmittance of the optical system in the range of the operating wavelengths (it can range from 0 to 1) and $SENSITIVITY_{DEFAULT}$ is the default value of sensitivity for object temperature data (it is usually around 2000 LSB/°C). The default value of sensitivity is calibrated for each unit with a blackbody covering the full field of view of the sensor without the use of optical elements (that is, $\tau = 1$). It can be obtained with the following formula:

$$SENSITIVITY_{DEFAULT}\left[\frac{LSB}{°C}\right] = SENS_DATA_{DEFAULT} \times 16 + 2048$$

where $SENS_DATA_{DEFAULT}$ is the default value of register 1Dh (stored as signed two's complement data). If the register content was previously overwritten, its default value can be restored with the boot/reboot procedure (see [Section 5 Boot and reboot](#)). Note that the object temperature data does not actually represent the temperature of the object(s) inside the field of view, as explained below.

If no optical element is used (that is, $\tau = 1$), the actual value of sensitivity is equal to the default value and the object temperature output data is not impacted. If an optical system with transmittance lower than 100% is mounted on the sensor (that is, $\tau < 1$), the actual value of sensitivity is lower than the default value and the object temperature output data is impacted. In the second case, if the embedded compensation algorithm was enabled, the $SENS_DATA$ (1Dh) register should be changed in order for the algorithm to work correctly. Refer to [Section 4.6 Compensating ambient temperature variations in the object temperature](#) for details on how to change this register. Note that changing the $SENS_DATA$ (1Dh) register does not change the actual value of sensitivity, which is a physical quantity independent from the value of this register, and does not have an impact on the measured object temperature data (TOBJECT_H and TOBJECT_L output data).

If gain reduction is enabled (see [Section 4.7 Reducing gain to avoid saturation](#)), each object temperature sample should also be multiplied by the gain reduction factor of 8 in order to obtain the corresponding value in °C.

Object temperature data represents the amount of infrared radiation emitted from the objects inside the field of view. If an ideal blackbody (that is, its emissivity is 100%) covers the entire field of view of the sensor, the object temperature data is proportional, using a constant σ , to the difference between the 4th power of its temperature T_{BB} and the 4th power of the ambient temperature of the environment in thermal coupling with the sensor, following the Steffan-Boltzmann law:

$$TOBJECT(\tau)[LSB] = \tau \times \sigma \left[\frac{LSB}{K^4} \right] \times (T_{BB}^4[K^4] - T_{AMBIENT}^4[K^4])$$

The actual value of the sensitivity for object temperature is linked to the constant of proportionality σ by the following formula:

$$SENSITIVITY(\tau) \left[\frac{LSB}{K} \right] = \frac{dTOBJECT(\tau)}{dT_{BB}} = 4 \times \tau \times \sigma \left[\frac{LSB}{K^4} \right] \times T_{BB}^3[K^3]$$

As already mentioned, the default value of the sensitivity for object temperature is calibrated for each unit with a blackbody at $T_{BB} = 24\text{ °C}$ (297.15 K) covering the full field of view of the sensor without the use of optical elements (that is, $\tau = 1$).

4.5.2 Ambient temperature data

The measured ambient temperature data are sent to the `TAMBIENT_H` (29h) and `TAMBIENT_L` (28h) registers. These registers contain, respectively, the most significant byte and the least significant byte of the ambient temperature value.

The complete output data is given by the concatenation of `TAMBIENT_H` and `TAMBIENT_L`. This ambient temperature value is represented as a 16-bit signed two's complement binary number (each digit is referred to as LSB).

Each ambient temperature output sample can be divided by the value of the sensitivity for ambient temperature data (that is, 100 LSB/°C) in order to obtain the corresponding value in °C, as in the following formula:

$$TAMBIENT [°C] = \frac{TAMBIENT[LSB]}{100 \left[\frac{LSB}{°C} \right]}$$

Ambient temperature data represent the actual temperature of the environment in thermal coupling with the device. These data can be used to compensate changes in the object temperature output data caused by a change in temperature of the environment in thermal coupling with the sensor (see [Section 4.6 Compensating ambient temperature variations in the object temperature](#)).

4.5.3 Example of temperature data

The following procedure is a simple example of how to get the object temperature data in LSB and transform it into °C:

1. Get raw data from the sensor:
 - a. `TOBJECT_L` (26h): 84h
 - b. `TOBJECT_H` (27h): 3Eh
2. Do registers concatenation:
 - a. `TOBJECT_L + (TOBJECT_H << 8)`: 3E84h
3. Calculate the signed decimal value (from signed two's complement 16-digit binary format):
 - a. `TOBJECT [LSB] = +16004`
4. Apply the sensitivity (in this case, the sensitivity for object temperature is assumed to be 2000 LSB/°C):
 - a. `TOBJECT [°C] = +16004 / 2000 ≈ +8`

The procedure to obtain ambient temperature data is equivalent.

4.5.4 Embedded smart digital algorithms data

The output data of the embedded smart digital algorithms (that is, presence detection, motion detection, and the ambient shock detection output data) are sent to the TPRESSENCE_H (3Bh) and TPRESSENCE_L (3Ah) registers, the TMOTION_H (3Dh) and TMOTION_L (3Ch) registers, and the TAMB_SHOCK_H (3Fh) and TAMB_SHOCK_L (3Eh) registers. Each pair contains, respectively, the most significant byte and the least significant byte of each output value. These data do not have an intrinsic physical meaning, but they are just used by the embedded smart digital algorithms to assert the respective flags.

The complete output data are given by the concatenation of each register pair. The values of the outputs of the embedded smart digital algorithms are represented as a 16-bit signed two's complement binary number.

The embedded smart digital algorithms are not compatible with one-shot mode. See [Section 7 Embedded smart digital algorithms](#) for more information on the smart digital algorithms.

4.6 Compensating ambient temperature variations in the object temperature

If during measurement the ambient temperature data changes significantly, the object temperature data might be affected, even if no change in the infrared radiation of the object(s) inside the field of view occurs. For this reason, a compensation algorithm based on ambient temperature data can be performed to improve the accuracy of the object temperature data. This procedure is effective only for compensating temperature changes that occur in the environment in thermal coupling with the sensor (for example, changes in the temperature of the PCB on which the sensor is mounted).

A simpler (linear) compensation algorithm is embedded in the STHS34PF80 device (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)), while a more advanced (nonlinear) compensation algorithm is offered by ST in a software library for motion and presence detection, which can be run on the microcontroller/application processor. The software library can also run a linear compensation algorithm, which is equivalent to the embedded one, in case the gain reduction factor is different from the default one and so the linear embedded compensation algorithm cannot be enabled (see [Section 4.7 Reducing gain to avoid saturation](#)). The software library is integrated in the X-CUBE-MEMS1 software package as the InfraredPD middleware library. Visit [X-CUBE-MEMS1](#) for more details.

In the case of small ambient temperature variations (for example, $\pm 10^{\circ}\text{C}$ around 24°C), the object temperature output can be effectively compensated with the linear algorithm, which is based on a linear approximation of the Stefan-Boltzmann law. The formula of this algorithm is the following:

$$TOBJ_COMP[LSB] = TOBJECT[LSB] + SENSITIVITY\left[\frac{LSB}{^{\circ}C}\right] \times (TAMBIENT - TAMBIENT_{ZERO})[^{\circ}C] = TOBJECT[LSB] + \frac{SENSITIVITY}{100}[1] \times (TAMBIENT - TAMBIENT_{ZERO})[LSB]$$

where TOBJECT is the current value of object temperature, SENSITIVITY is the value of the sensitivity for object temperature data, TAMBIENT is the current value of the ambient temperature, and TAMBIENT_{ZERO} is the value of ambient temperature at the instant the compensation algorithm is initialized (or reinitialized).

Since the embedded compensation algorithm uses the content of the SENS_DATA (1Dh) register to obtain the SENSITIVITY parameter of the formula above and to calculate the compensated object temperature signal, if an optical system with transmittance lower than 100% in the range of the operating wavelengths (between 5 μm and 20 μm) is used with the sensor, the value of this register should be updated accordingly in order for the algorithm to work correctly. The register should be updated according to the following formula:

$$SENS_DATA = \frac{\tau \times SENSITIVITY_{DEFAULT}\left[\frac{LSB}{^{\circ}C}\right] - 2048}{16}$$

where τ is the transmittance of the optical system in the range of the operating wavelengths and SENSITIVITY_{DEFAULT} is the default value of sensitivity for object temperature data. Refer to [Section 4.5.1 Object temperature data](#) for details on how to obtain the default value. If the value of SENS_DATA obtained from the formula is not an integer, it must be rounded; if rounded to the nearest integer, the round-off error of the actual sensitivity value due to internal computations can be ± 50 LSB/ $^{\circ}\text{C}$ at most. Note that the value of the SENS_DATA (1Dh) register is stored as signed two's complement data and its default value is restored after performing a boot/reboot procedure (see [Section 5 Boot and reboot](#)), so it should be updated every time this procedure is executed, if necessary.

If the value of the transmittance of the optical system is unknown, it can be estimated by performing two object temperature data measurements with the same conditions inside the field of view and having the same ambient temperature data output, one without the optical system mounted on the sensor ($\tau = 1$) and one with the optical system mounted on the sensor ($\tau_{UNK} < 1$, to be estimated), as in the following formula:

$$\tau_{UNK} = \frac{TOBJECT(\tau = \tau_{UNK}) [LSB]}{TOBJECT [^{\circ}C] \times SENSITIVITY_{DEFAULT} \left[\frac{LSB}{^{\circ}C} \right]} = \frac{TOBJECT(\tau = \tau_{UNK}) [LSB]}{TOBJECT(\tau = 1) [LSB]}$$

Note that this estimate is less accurate the closer the object temperature data is to saturation and the higher the emissivity of the optical system. This is due to the fact that an optical system with high emissivity introduces an undesirable contribution of radiation in the object temperature signal that cannot be compensated. Refer to [AN5983](#) for more information on lens and cover mounting guidelines for the STHS34PF80 sensor.

In the case of larger ambient temperature variations, the object temperature output should be compensated with the nonlinear algorithm, which is based on the nonapproximated Stefan-Boltzmann law. Note that the nonlinear algorithm, too, cannot compensate temperature changes that occur in an environment not in thermal coupling with the sensor (for example, effects caused by heaters or coolers distant from the sensor). These environmental temperature changes might affect the output of the embedded presence detection algorithm (see [Section 7.1 Presence detection](#)). The software library mentioned above is designed to deal with these conditions and to return a more accurate presence state flag.

4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature

In order to enable the embedded algorithm for compensating ambient temperature variations in the object temperature, a procedure that consists of setting the COMP_TYPE bit of the ALGO_CONFIG (28h) embedded functions register must be implemented. The complete procedure is the following:

1. Enter power-down mode (see [Section 3.1 Power-down mode](#)), if not already in it
2. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
3. Write bit FUNC_CFG_READ = 1 in PAGE_RW (11h) // Enable read operation
4. Write 28h in FUNC_CFG_ADDR (08h) // Set register address (ALGO_CONFIG register)
5. Read the value XXh from FUNC_CFG_DATA (09h) // Read register value XX
6. Perform bitwise OR operation between the value XXh and the value 04h // Set COMP_TYPE bit
7. Write bit FUNC_CFG_READ = 0 in PAGE_RW (11h) // Disable read operation
8. Write bit FUNC_CFG_WRITE = 1 in PAGE_RW (11h) // Enable write operation
9. Write 28h in FUNC_CFG_ADDR (08h) // Set register address (ALGO_CONFIG register)
10. Write the value obtained at step 6 in FUNC_CFG_DATA (09h) // Write register value
11. Write 2Ah in FUNC_CFG_ADDR (08h) // Set register address (RESET_ALGO register)
12. Write 01h in FUNC_CFG_DATA (09h) // Write register value (set ALGO_ENABLE_RESET bit to reset embedded algorithms)
13. Write bit FUNC_CFG_WRITE = 0 in PAGE_RW (11h) // Disable write operation
14. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
15. Write bits ODR[3:0] in CTRL1 (20h) with the desired value or trigger a one-shot acquisition (see [Section 3.2 One-shot mode](#))

This procedure includes the reset of the algorithms, which must be performed after changing the COMP_TYPE bit. The embedded compensation algorithm can be reinitialized at any time by performing the procedure for resetting the algorithms, without having to change the COMP_TYPE bit. Note that by performing the reset procedure, the embedded smart digital algorithms are also reinitialized. The procedure for resetting the algorithms is described separately in [Section 7.4 Resetting the algorithms](#).

Once the procedure has been completed, the compensated object temperature data are available in the `TOBJ_COMP_L` (38h) and `TOBJ_COMP_H` (39h) registers. These registers contain, respectively, the most significant byte and the least significant byte of the compensated object temperature value. The complete output data is given by the concatenation of `TOBJ_COMP_H` and `TOBJ_COMP_L`, and the value is represented as a 16-bit signed two's complement binary number. Note that registers 26h and 27h always contain the raw object temperature data, even if the compensation is active, and that register 38h and 39h are aligned with registers 26h and 27h if the compensation is not active.

As for the raw object temperature output data, the compensated object temperature output data (in LSB) is proportional to the transmittance of the optical system in use with the sensor in the range of the operating wavelengths (between 5 μm and 20 μm). Each compensated object temperature output sample can be divided by the actual value of sensitivity for object temperature data in order to obtain the corresponding value in $^{\circ}\text{C}$. Note that the compensated object temperature data does not actually represent the temperature of the object(s) inside the field of view.

While the embedded algorithm for compensating ambient temperature variations in the object temperature is active, the embedded smart digital algorithms are applied to the compensated object temperature data, and not the raw one. The registers of the algorithms remain the same.

In order to disable the embedded compensation algorithm, the same procedure seen above must be applied, except that the operation at step 6 must be substituted by a bitwise AND operation between the value `XXh` and the value `FBh`.

If gain reduction is enabled (see [Section 4.7 Reducing gain to avoid saturation](#)), the embedded compensation algorithm cannot be enabled.

4.7 Reducing gain to avoid saturation

The object temperature signal is sensitive to temperature differences between the temperature of the environment in thermal coupling with the sensor (`TAMBIENT` data output) and the temperature of the room in which the object in the field of view of the sensor is located. This delta of temperature could impact the operating temperature range of the sensor, and the object temperature signal could reach saturation.

The sensitivity of the device for the object temperature data can be reduced from the default value in order to increase the object temperature dynamic and avoid the possibility of falling into the condition of saturation. In order to decrease the sensitivity, the gain of the sensor can be reduced by a factor of 8. In the reduced gain mode (also called "wide mode"), saturation happens only if the ambient temperature of the environment in thermal coupling with the sensor is 90 $^{\circ}\text{C}$ higher or is 50 $^{\circ}\text{C}$ lower than the ambient temperature of the room over an operating range from -40 $^{\circ}\text{C}$ to 85 $^{\circ}\text{C}$. More details on the maximum ambient temperature differences and on the operating temperature ranges at the different gain values are available in the datasheet.

The procedure to divide the gain by a factor of 8 consists of setting the `GAIN[2:0]` bit range of the `CTRL0` (17h) register to `000b`, which corresponds to the `CTRL0` register value of `81h`. The default value of the bit range is `111b`, which corresponds to the `CTRL0` register value of `F1h`; the default value is associated with a gain reduction factor of 1 (that is, a nonreduced gain).

The gain reduction does not have an impact on the signal-to-noise ratio (SNR) of the object temperature data. The original sensitivity can be recovered later using software by multiplying the data for the gain reduction factor of 8, if necessary.

The gain reduction is not applied to the ambient temperature data.

When the gain reduction is enabled, the embedded compensation algorithm cannot be enabled (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)), and the embedded smart digital algorithms cannot be used (see [Section 7 Embedded smart digital algorithms](#)). In this case, the algorithms for compensating ambient temperature variations in the object temperature, for motion detection and for presence detection can be performed through a software library offered by ST, which can be run on the microcontroller/application processor. The software library is integrated in the `X-CUBE-MEMS1` software package as the `InfraredPD` middleware library. Visit [X-CUBE-MEMS1](#) for more details.

5 Boot and reboot

After the device is powered up, it performs a 2.5 ms boot procedure to download the calibration coefficients from the embedded flash to the internal registers and it loads the default content of the AVG_TRIM (10h), the CTRL0 (17h), and the SENS_DATA (1Dh) registers. During the boot time, the registers of the device are not accessible. After the boot is completed, the device is automatically configured in power-down mode.

After power-up, the boot procedure can be restarted in order to reload the default values of the above-mentioned registers by setting the BOOT bit of the CTRL2 (21h) register to 1. No toggle of the device power lines is required. During the reboot time, the registers of the device are not accessible. After the reboot is completed, the device is automatically configured in the previous operating mode, and the BOOT bit is self-cleared to 0.

6 Interrupt generation

The STHS34PF80 has a built-in configurable interrupt generator block, which allows generating interrupt events based on temperature data samples and on the output flags of the embedded smart digital algorithms.

The INT output pin allows selecting push-pull or open-drain configuration by acting on the PP_OD bit of the CTRL3 (22h) register. The default configuration is push-pull (PP_OD set to 0).

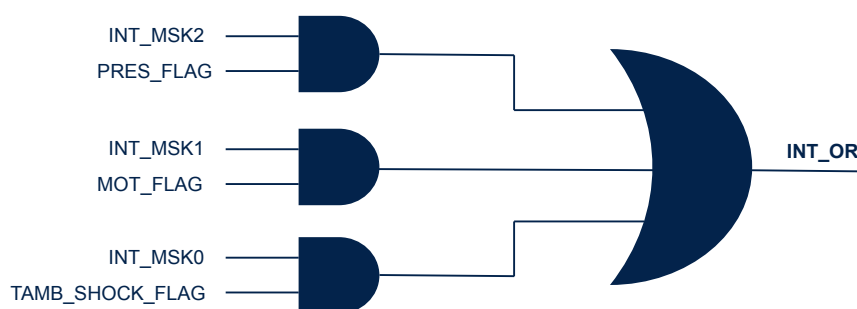
The interrupt electrical signal logic on the INT pin is selected by acting on the INT_H_L bit of the CTRL3 register. The default configuration is active high (INT_H_L set to 0), meaning that the INT pin is at voltage level high when the interrupt is asserted. By setting the INT_H_L bit to 1, the interrupt configuration changes to active low. This choice does not apply to the status registers logic, which is always active high.

Interrupt events are generated when either the data-ready signal from the STATUS (23h) register is asserted or when one of the enabled flags from the embedded smart algorithms from the FUNC_STATUS (25h) register is asserted. The IEN[1:0] bit range of the CTRL3 register allows configuring the signal routed to the INT pin with the following logic:

- When set to 00 (default value), the INT pin is left at high impedance.
- When set to 01, the DRDY signal is routed to the INT pin (see [Section 4.3 Using the data-ready signal](#)).
- When set to 10, the INT_OR signal is routed to the INT pin.

The INT_OR signal allows diverting one or more of the output flags of the embedded smart digital algorithms to the INT pin. The INT_OR condition is the result of the logical operation represented in [Figure 3. INT_OR](#), where INT_MSK0, INT_MSK1 and INT_MSK2 are bits of the CTRL3 register and work as masks for the flags of the algorithms for ambient temperature shock detection, motion detection, and presence detection, respectively. See [Section 7 Embedded smart digital algorithms](#) for more details on the embedded smart digital algorithms.

Figure 3. INT_OR



The INT_OR signal diverted to the INT pin can be pulsed or not depending on the value of the INT_PULSED bit of the ALGO_CONFIG (28h) embedded functions register. If the bit is set to 0 (default value), then the INT_OR signal is equivalent at every ODR cycle to the INT_OR condition. If the bit is set to 1, then the INT_OR signal is pulsed; in this case, the signal is set again at each ODR cycle if the INT_OR condition is still true. Note that one additional spurious interrupt pulse might be observed even if the INT_OR condition has already turned false. When the INT_OR signal is diverted to the INT pin, the INT_LATCHED bit of the CTRL3 (22h) register must be set to 0. In one-shot mode and in reduced gain mode (wide mode), the INT_OR signal must not be routed to the INT pin, since the embedded smart digital algorithms are not compatible with these modes.

If the interrupt service routine (ISR) of the application processor is set to be triggered on both the rising and falling edges of the INT pin and the INT_OR signal is configured to be routed to the INT pin and the INT_PULSED bit is set to 0, the user can recognize when a change in the INT_OR condition occurs (and act accordingly). For example, in order to be able to recognize the changes of the presence detection algorithm flag (from absence to presence, and vice versa), the following procedure must be performed:

1. Write 22h in the CTRL3 (22h) register // Enable routing the INT_OR signal to the INT pin and set the INT_MSK2 bit (enabling the presence detection flag)
2. Set the ISR of the application processor in edge-triggered mode, on both rising and falling edges of the INT pin
3. Enter continuous mode (see [Section 3.3 Continuous mode](#)), if not already in it

7 Embedded smart digital algorithms

The STHS34PF80 embeds smart digital algorithms to support the following three detection modes:

- Presence detection
- Motion detection
- Ambient temperature shock detection

Each of these algorithms makes use of a different low-pass filter (LPF_P, LPF_M and LPF_A_T, respectively). Moreover, the presence and motion detection algorithms make use of another common low-pass filter (LPF_P_M). These filters are used to generate intermediate signals (TPRESENCE, TMOTION, and TAMB_SHOCK, respectively), which can be used to fine-tune the algorithms themselves. The value of the cutoff frequency of each of these filters can be configured through their respective bit ranges, found in the LPF1 (0Ch) and LPF2 (0Dh) registers, as explained in the following table.

Table 9. Configuration of low-pass filters

LPF_P_M[2:0] / LPF_M[2:0] / LPF_P[2:0] / LPF_A_T[2:0]	Cutoff frequency
000	ODR/9
001	ODR/20
010	ODR/50
011	ODR/100
100	ODR/200
101	ODR/400
110	ODR/800

The smart digital algorithms themselves can be configured through the embedded functions registers (see [Section 2.1 Embedded functions registers](#) for how to access them), as explained in the following sections. When changing one or more of these registers, or when changing registers LPF1 and/or LPF2, the device needs to be set in power-down mode beforehand, and the algorithms must be reset through the procedure described in [Section 7.4 Resetting the algorithms](#) for the changes to take effect. The reset procedure can also be used to reinitialize the algorithm, but it should be carried out only when the field of view of the sensor is empty, otherwise the outputs of the algorithms could be impacted. The field of view must also be empty whenever going from power-down mode to continuous mode, both at startup and after performing the procedure detailed in [Section 3.3 Continuous mode](#) in order to change the ODR. Note that by performing the reset procedure, the embedded compensation algorithm is also reinitialized (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)). All these procedures reset both the signals and the flags of the embedded smart digital algorithms to 0.

The flags for presence detection, motion detection, and ambient temperature shock detection can be checked from the PRES_FLAG, MOT_FLAG and TAMB_SCHOCK_FLAG bits of the FUNC_STATUS (25h) register, respectively. At each ODR cycle, these flags are set to 1 when the respective condition is true and are reset to 0 when the condition is not true anymore.

If the embedded algorithm for compensating ambient temperature variations in the object temperature is enabled, the embedded smart digital algorithms apply to the compensated object temperature data instead of the raw object temperature data.

The outputs (signals and flags) of the embedded smart digital algorithms are all updated together with the readings of the object and ambient temperature at the data rate set from the ODR[3:0] bit range of the CTRL1 (20h) register. The embedded smart digital algorithms are not compatible with one-shot mode, as the cutoff frequencies of the filters cannot be guaranteed.

If gain reduction is enabled (see [Section 4.7 Reducing gain to avoid saturation](#)), the embedded smart digital algorithms cannot be used.

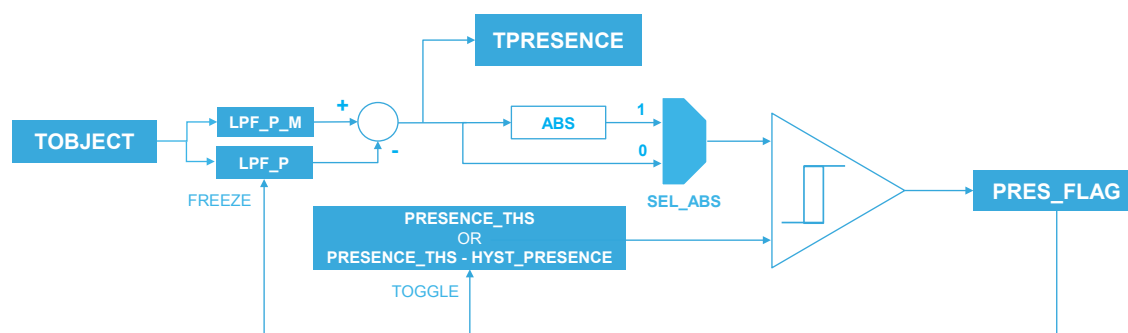
7.1 Presence detection

The presence detection algorithm is performed on the difference between the output value of the low-pass filter for motion and presence detection (LPF_P_M) and of the low-pass filter for presence detection (LPF_P), which both have as input the (either raw or compensated) object temperature data. This difference is the TPRESSENCE data (saved in registers 3Ah and 3Bh). The filter for motion and presence detection is used to remove high-frequency noise and is configured through the LPF_P_M[2:0] bit range of the LPF1 (0Ch) register, while the filter for presence detection is used to remove background low-frequency noise and is configured through the LPF_P[2:0] bit range of the LPF2 (0Dh) register. It follows that the LPF_P_M filter should have a higher cutoff frequency than the LPF_P filter.

The presence detection flag data (PRES_FLAG) is set when the TPRESSENCE data exceeds the threshold value defined in the PRESENCE_THS (20h) embedded functions register (default value: 200 LSB), as described in [Figure 4. Block diagram of presence detection algorithm](#). If bit SEL_ABS of the ALGO_CONFIG (28h) embedded functions register is set, the absolute value of the TPRESSENCE data is used instead. When the PRES_FLAG is asserted, the LPF_P output remains at its last value; the LPF_P filter starts processing again the input data when the PRES_FLAG is deasserted.

A hysteresis logic function on the threshold is implemented. The hysteresis value can be configured through the HYST_PRESENCE (27h) embedded functions register (default value: 50 LSB). When TPRESSENCE becomes greater than PRESENCE_THS, PRES_FLAG goes to 1; PRES_FLAG returns to 0 when TPRESSENCE becomes lower than PRESENCE_THS - HYST_PRESENCE.

Figure 4. Block diagram of presence detection algorithm



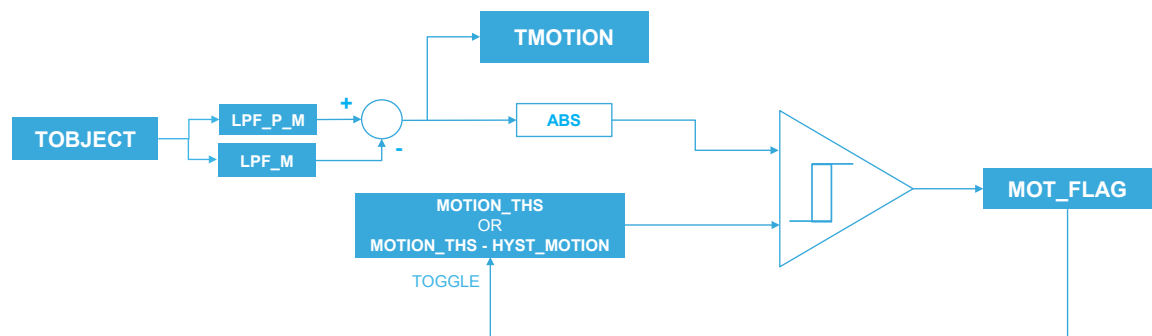
A more advanced presence detection algorithm is offered by ST in a software library, which can be run on the microcontroller/application processor. This software library also includes the software algorithm for compensating ambient temperature variations in the object temperature described in [Section 4.6 Compensating ambient temperature variations in the object temperature](#) and an algorithm for motion detection similar to the embedded one (see [Section 7.2 Motion detection](#)). The presence detection algorithm of the software library improves upon the embedded algorithm and can also be used in case gain reduction is enabled and so the embedded presence detection algorithm cannot be used (see [Section 4.7 Reducing gain to avoid saturation](#)). Moreover, unlike the embedded algorithm for presence detection, if a reset of the algorithms is performed, the presence state is not lost (that is, the presence flag is not reset to 0 if it had been set to 1). The presence detection algorithm of the software library improves upon the embedded algorithm particularly in the cases where environmental temperature changes not related to a modification in the presence state and which cannot be compensated with the compensation algorithms cause the TOBJECT/TOBJ_COMP signal to drift. Examples include the effect of heaters and coolers. Since the LPF_P filter is frozen while the presence condition is true (that is, the PRES_FLAG bit set to 1), in these intervals the drifts related to slow environmental temperature changes also affect the TPRESSENCE signal, and in turn affect the presence state. The software library is designed to deal with these conditions and to return a more accurate presence state flag. The software library is integrated in the X-CUBE-MEMS1 software package as the InfraredPD middleware library. Visit [X-CUBE-MEMS1](#) for more details.

7.2 Motion detection

The motion detection algorithm is performed on the difference between the output value of the low-pass filter for motion and presence detection (LPF_P_M) and of the low-pass filter for motion detection (LPF_M), which both have as input the (either raw or compensated) object temperature data. This difference is the TMOTION data (saved in registers 3Ch and 3Dh). The filter for motion and presence detection is used to remove high-frequency noise and is configured through the LPF_P_M[2:0] bit range of the LPF1 (0Ch) register, while the filter for motion detection is used to remove background low-frequency noise and is configured through the LPF_M[2:0] bit range of the same register. It follows that the LPF_P_M filter should have a higher cutoff frequency than the LPF_P filter. The motion detection flag data (MOT_FLAG) is set when the TMOTION data exceeds the threshold value defined in the MOTION_THS (22h) embedded functions register (default value: 200 LSB), as described in [Figure 5. Block diagram of motion detection algorithm](#).

A hysteresis logic function on the threshold is implemented. The hysteresis value can be configured through the HYST_MOTION (26h) embedded functions register (default value: 50 LSB). When TMOTION becomes greater than MOTION_THS, MOTION_FLAG goes to 1; MOTION_FLAG returns to 0 when TMOTION becomes lower than MOTION_THS – HYST_MOTION.

Figure 5. Block diagram of motion detection algorithm



A similar motion detection algorithm is offered by ST in a software library, which can be run on the microcontroller/application processor. This software library also includes the software algorithm for compensating ambient temperature variations in the object temperature described in [Section 4.6 Compensating ambient temperature variations in the object temperature](#) and an advanced presence detection algorithm (see [Section 7.1 Presence detection](#)). The motion detection algorithm of the software library can be used in case gain reduction is enabled and so the embedded motion detection algorithm cannot be used (see [Section 4.7 Reducing gain to avoid saturation](#)). The software library is integrated in the X-CUBE-MEMS1 software package as the InfraredPD middleware library. Visit [X-CUBE-MEMS1](#) for more details.

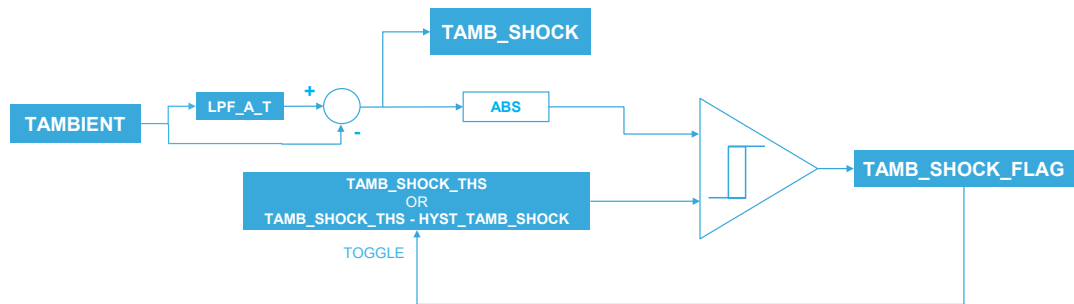
7.3 Ambient temperature shock detection

Ambient temperature shock detection is useful for all applications in which the environment where the device is located is subject to sudden temperature changes. This detection algorithm is performed on the difference between the output value of the low-pass filter for ambient temperature shock detection (LPF_A_T), which has as input the ambient temperature data, and the ambient temperature data itself. This difference is the TAMB_SHOCK data (saved in registers 3Eh and 3Fh). The filter is used to remove background low-frequency noise and is configured through the LPF_A_T[2:0] bit range of the LPF2 (0Dh) register.

The ambient temperature shock detection flag data (TAMB_SHOCK_FLAG) is set when the TAMB_SHOCK data exceeds the threshold value defined in the TAMB_SHOCK_THS (24h) embedded functions register (default value: 10 LSB), as described in Figure 6. Block diagram of ambient temperature shock detection algorithm.

A hysteresis logic function on the threshold is implemented: the hysteresis value can be configured through the HYST_TAMB_SHOCK (29h) embedded functions register (default value: 2 LSB). When TAMB_SHOCK becomes greater than TAMB_SHOCK_THS, TAMB_SHOCK_FLAG goes to 1; TAMB_SHOCK_FLAG return to 0 when TAMB_SHOCK becomes lower than TAMB_SHOCK_THS - HYST_TAMB_SHOCK.

Figure 6. Block diagram of ambient temperature shock detection algorithm



7.4 Resetting the algorithms

When changing one or more of the parameters of the smart digital algorithms (through the embedded functions registers) and/or the cutoff frequencies of the filters (through registers LPF1 and LPF2) and also when enabling/disabling the embedded compensation algorithm (see [Section 4.6.1 Embedded algorithm for compensating ambient temperature variations in the object temperature](#)), it is necessary to perform the following procedure to reset the algorithms:

1. Enter power-down mode (see [Section 3.1 Power-down mode](#)), if not already in it
2. Change the configuration of the algorithms, if necessary
3. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
4. Write bit FUNC_CFG_WRITE = 1 in PAGE_RW (11h) // Enable write operation
5. Write 2Ah in FUNC_CFG_ADDR (08h) // Set register address (RESET_ALGO register)
6. Write 01h in FUNC_CFG_DATA (09h) // Write register value (set ALGO_ENABLE_RESET bit)
7. Write bit FUNC_CFG_WRITE = 0 in PAGE_RW (11h) // Disable write operation
8. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
9. Write bits ODR[3:0] in CTRL1 (20h) with the desired value or trigger a one-shot acquisition (see [Section 3.2 One-shot mode](#))

When changing one or more of the parameters of the smart digital algorithms through the embedded functions registers, it is possible to access the embedded functions registers at once to both set the new value of the parameter(s) and enable the ALGO_ENABLE_RESET bit. For example, in order to set the threshold for the presence detection algorithm to 500 LSB (0x1F4 in hexadecimal), follow this procedure:

1. Enter power-down mode (see [Section 3.1 Power-down mode](#)), if not already in it
2. Write bit FUNC_CFG_ACCESS = 1 in CTRL2 (21h) // Enable access to the embedded functions registers
3. Write bit FUNC_CFG_WRITE = 1 in PAGE_RW (11h) // Enable write operation
4. Write 20h in FUNC_CFG_ADDR (08h) // Set register address (PRESENCE_THS_L register)
5. Write F4h in FUNC_CFG_DATA (09h) // Write register value (PRESENCE_THS_L register)
6. Write 01h in FUNC_CFG_DATA (09h) // Write register value (PRESENCE_THS_H register)
7. Write 2Ah in FUNC_CFG_ADDR (08h) // Set register address (RESET_ALGO register)
8. Write 01h in FUNC_CFG_DATA (09h) // Write register value (set ALGO_ENABLE_RESET bit)
9. Write bit FUNC_CFG_WRITE = 0 in PAGE_RW (11h) // Disable write operation
10. Write bit FUNC_CFG_ACCESS = 0 in CTRL2 (21h) // Disable access to the embedded functions registers
11. Write bits ODR[3:0] in CTRL1 (20h) with the desired value or trigger a one-shot acquisition (see [Section 3.2 One-shot mode](#))

The procedure for resetting the algorithms can be used to reinitialize at any time both the embedded compensation algorithm and the embedded smart digital algorithms, even if no change of their parameters is necessary.

As previously stated, the reset procedure should be ideally carried out only when there is no human body and/or heat source inside the field of view of the sensor (which should correspond to a presence flag reporting absence), otherwise the outputs of the embedded smart digital algorithms could be impacted, particularly the output of the presence detection algorithm. For example, if the reset procedure were to be performed while a person is still inside the field of view of the sensor, the exit of the person from the field of view after the reset could decrease the TPRESNCE signal to a value lower than the negative presence threshold, which could, in turn, incorrectly trigger the presence flag if the SEL_ABS bit of the ALGO_CONFIG embedded functions register was previously set to 1. If it is not possible for the field of view to be empty before performing the reset procedure, an alternative option could be to use the information coming from the output of the embedded motion detection algorithm to assess movement in the field of view just after the reset in order to discard the presence detection output until the field of view has become empty (at which point the reset procedure can be reapplied).

Revision history

Table 10. Document revision history

Date	Version	Changes
31-May-2023	1	Initial release
05-Jul-2023	2	Minor textual updates

Contents

1	Pin description	2
2	Registers	3
2.1	Embedded functions registers	4
2.1.1	Write procedure for embedded functions registers	5
2.1.2	Read procedure for embedded functions registers	5
3	Operating modes	6
3.1	Power-down mode	7
3.2	One-shot mode	7
3.3	Continuous mode	8
4	Reading output data	10
4.1	Startup sequence	10
4.2	Using the status register	10
4.3	Using the data-ready signal	10
4.4	Using the block data update (BDU) feature	11
4.5	Understanding output data	11
4.5.1	Object temperature data	11
4.5.2	Ambient temperature data	12
4.5.3	Example of temperature data	12
4.5.4	Embedded smart digital algorithms data	13
4.6	Compensating ambient temperature variations in the object temperature	13
4.6.1	Embedded algorithm for compensating ambient temperature variations in the object temperature	14
4.7	Reducing gain to avoid saturation	15
5	Boot and reboot	16
6	Interrupt generation	17
7	Embedded smart digital algorithms	18
7.1	Presence detection	19
7.2	Motion detection	20
7.3	Ambient temperature shock detection	21
7.4	Resetting the algorithms	22
	Revision history	23
	List of tables	25
	List of figures	26

List of tables

Table 1.	Pin list, functions, and internal status	2
Table 2.	Registers	3
Table 3.	Embedded functions registers	4
Table 4.	Averaging selection for ambient temperature	6
Table 5.	Averaging selection for object temperature and noise	6
Table 6.	Typical conversion time and maximum data rate in one-shot mode	8
Table 7.	ODR configuration	8
Table 8.	Current consumption at different ODRs and AVG_TMOS setting (@ $V_{DD} = 1.8\text{ V}$, $T = 25\text{ °C}$)	9
Table 9.	Configuration of low-pass filters	18
Table 10.	Document revision history	23

List of figures

Figure 1.	Pin configuration (package bottom view)	2
Figure 2.	One-shot mode.	7
Figure 3.	INT_OR.	17
Figure 4.	Block diagram of presence detection algorithm	19
Figure 5.	Block diagram of motion detection algorithm.	20
Figure 6.	Block diagram of ambient temperature shock detection algorithm	21

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.

© 2023 STMicroelectronics – All rights reserved