# High Precision Digital Thermometer
# M117 M117B

# Datasheet

(V4.3)



2022/08

# Table of Contents

## 1. Overview

M117 is MinYuan fourth generation of digital high-precision temperature sensor with a typical temperature measurement accuracy of 0.1 ℃. The temperature sensing principle is based on the characteristic relationship between the semiconductor PN-junction temperature and voltage, through small signal amplification, analog to digital conversion and digital calibration, the 16-bit digital temperature data can be easily accessed. It has strength of high accuracy and consistency, long lifetime, low power consumption, programmable configuration flexibility. The thermometer integrates an EEPROM as a storage for ROM ID of the sensors, temperature alarm threshold, temperature calibration correction values and user-defined information such as sensor node information. Each sensor IC has a unique 64-bit serial number as ROM ID, and is fully calibrated before ship out. The IC has I2C interface for the host processor to access.

High-precision temperature measurement region can be customized upon requirement. For example, for wearable thermometer and cold chain logistics range, the in-fab calibration procedure is different. Therefore the product series is divided into classes of 0.1℃/0.5°C for different working conditions and precision requirement at different price, as shown below.

| Part Number | Typical Accuracy @ Range(℃) | Address bit | Package |
|---|---|---|---|
| M117 | ±0.1℃@28℃ to 43℃ | 0x44 | DFN6L |
| M117B01 | ±0.1℃@28℃ to 43℃ | 0x45 | DFN6L |
| M117Z | ±0.1℃@0℃ to 50℃ | 0x44 | DFN6L |
| M117W | ±0.1℃@20℃ to 70℃ | 0x44 | DFN6L |
| M117P | ±0.1℃@-20℃ to +30℃ | 0x44 | DFN6L |
| M117B | ±0.5℃@0℃ to 50℃ | 0x45 | DFN6L |
| M117B05 | ±0.5℃@0℃ to 50℃ | 0x44 | DFN6L |

## 2. Features

- Highest accuracy: ±0.1°C, (or ±0.5°C, accuracy in the product series)
- -70°C~+150°C working temperature range
- The standard to the series has a default 16-bit ADC output, with a maximum resolution of

0.004°C

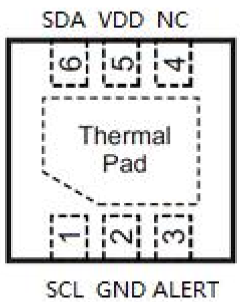
- Configurable conversion time: 10.5ms/5.5ms/4ms
- Configurable single shot mode and periodic data acquisition mode
- User configurable alarm threshold
- 32-bit non-volatile storage space for user data
- Wide supply voltage range 1.8V-5.5V
- Typical standby power consumption is 0.1μA@3.3V, maximum peak power consumption is 0.45mA@3.3V and average temperature measurement current is 5.2μA (@3.3V, 1 sample per second)
- Every thermometer has a unique ROM ID for easy network addressing

## 3. Applications

- Body and animal temperature measurement
- Electronic thermometer
- Medical Electronics
- Cold chain logistics
- Heat meter gas meter water meter
- Board level temperature monitoring
- Environmental temperature monitoring
- Smart home
- Consumer electronics
- Temperature measuring instrument

## 4. Package pin description

DFN6，2 X 2 X 0.75mm




（Top View）

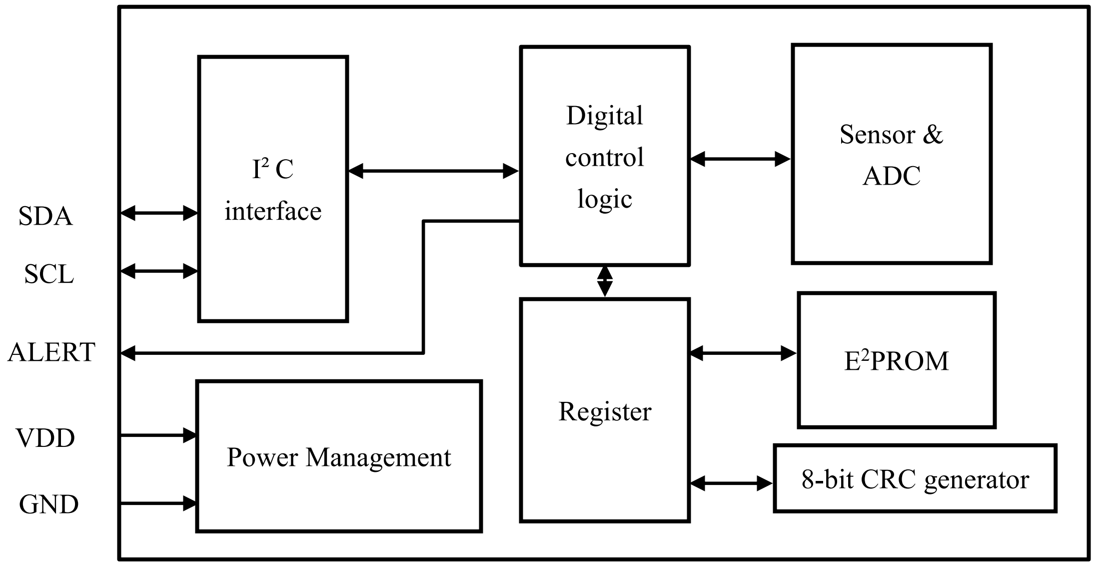

Physical Photo

| No. | Pin Name | I/O | Description |
|---|---|---|---|
| 1 | SCL | IO | $I^2C$ data line |
| 2 | GND | — | Ground |
| 3 | ALERT | O | Alarm |
| 4 | NC | — | Floated |

| | | | |
|---|---|---|---|
| 5 | VDD | — | Voltage |
| 6 | SDA | IO | $I^2C$ data line |
| Thermal pad | NC | — | Floating or grounded(1) |

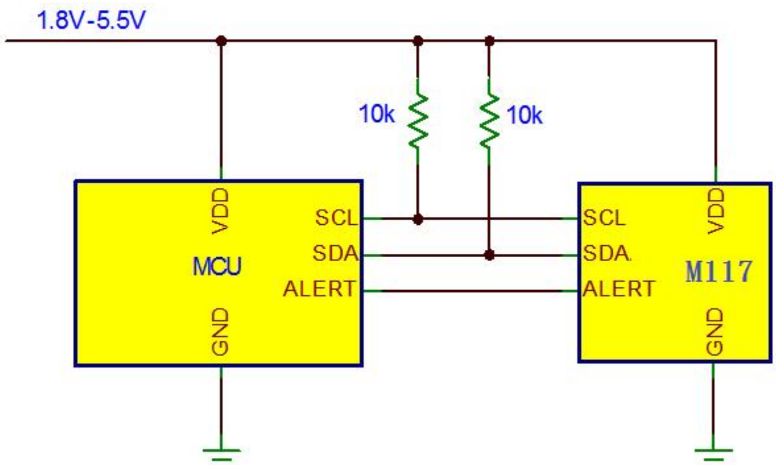
Note1: When designing the circuit, it is recommended that the thermal pad is floating or grounded.
(1) If the IC is pasted on PCB, it is recommended to connect the thermal pad to ground;
(2) If the thermal pad is pasted with metal sheets for temperature detection, it is recommended to not connect the thermal pad to ground.

## 5. System Diagram




The figure above shows the schematic diagram of the temperature sensor. The 64 bit ROM stores the unique ID serial number of the device. The register contains two bytes of temperature register, which stores the digital output from the sensor. In addition, the register of alarm trigger threshold is provided by the register and the extended register. The configuration register allows the user to set the repeatability of temperature digital conversion and the frequency of continuous measurement. The status register can be used to query whether an alarm is triggered. Data can be stored in non-volatile memory, and data will not be lost when the chip is powered down.

## 6. Typical Application




Note: For M117, M117Z, M117W, M117P and M117B05, I$^2$C address are fixed to 0x44.
For M117B and M117B01, I2C address is fixed to 0x45.

## 7. Temperature measurement performance

| Parameter | Symbol | Condition | Min | Typ | Max |
|---|---|---|---|---|---|
| Range | — | — | -70℃ | — | 150℃ |
| Accuracy tolerance | $t_{ERR}$ | M117/M117B01 | — | — | ±0.1℃@+28℃ to +43℃,<br>±0.5℃@-10℃to+28℃&+43°C to +60℃ |
| | | M117Z | — | — | ±0.1℃@0℃ to +50℃,<br>±0.2℃@-10℃ to 0℃&+50℃ to +60℃,<br>±0.5℃@-30℃to -10℃ &+60℃ to +70℃ |
| | | M117W | — | — | ±0.1℃@+20℃ to +70℃,<br>±0.2℃@0℃ to +20℃&+70℃ to+75℃,<br>±0.5@-20℃ to +0℃&+75℃ to+90℃ |
| | | M117P | — | — | ±0.1℃@-20℃ to +30℃,<br>± 0.2 ℃ @-30 ℃ to -20 ℃ &+30℃to+40℃,<br>±0.5℃@-70℃to -30℃ &+40℃to +70℃ |
| | | M117B/M117B05 | — | — | ±0.5℃@0℃ to +50℃ |

| | | | | | |
|---|---|---|---|---|---|
| | | M117/M117Z/M117W/M117P/M117B/M117B01/M117B05 | — | ±1.5℃@-55°C to +125°C | — |
| | | | — | ±2℃@-70°C to +150°C | — |
| Repeatability | | Low repeatability setting | — | 0.07°C | — |
| | | Medium repeatability setting | — | 0.05°C | Note 1 |
| | | High repeatability setting | — | 0.03°C | — |
| Resolution | | — | — | 0.004°C | — |
| Long-term drift | | — | — | — | 0.03°C/Y |

*Note 1: different output accuracy can be achieved by configuring different filter bandwidth, see table 9.2-2.*

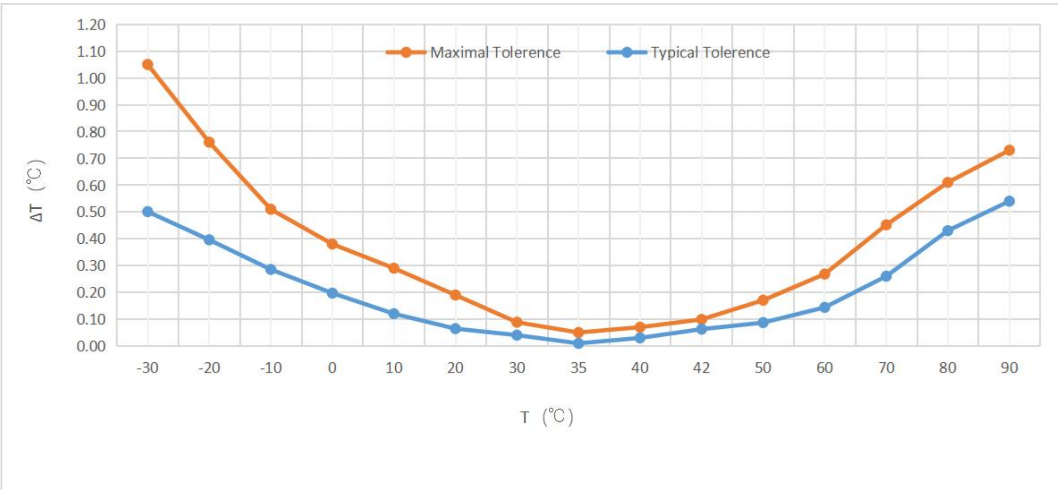



Figure 7-1 M117 accuracy curve

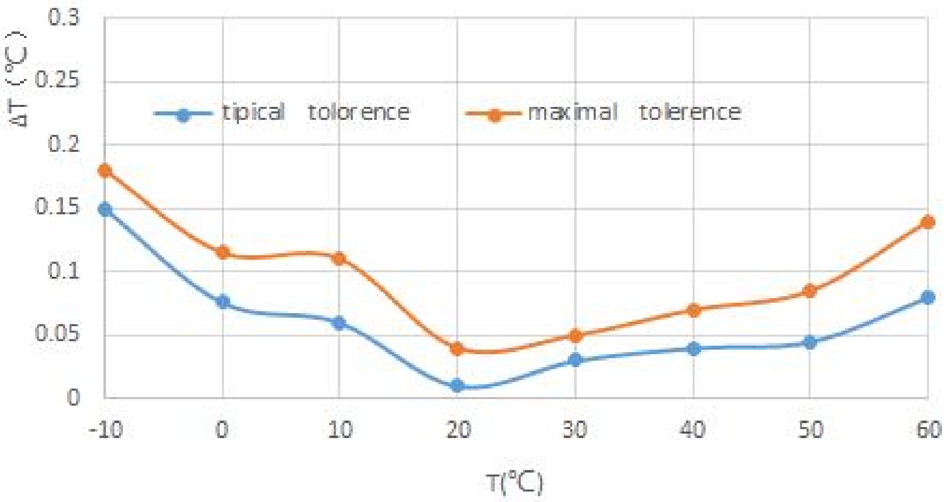



Figure 7-2 M117Z accuracy curve

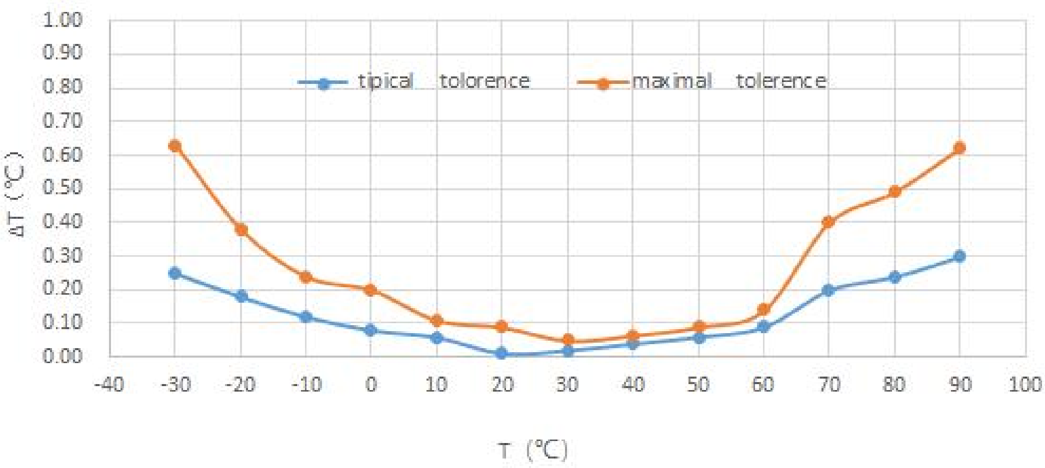



Figure 7-3 M117W accuracy curve

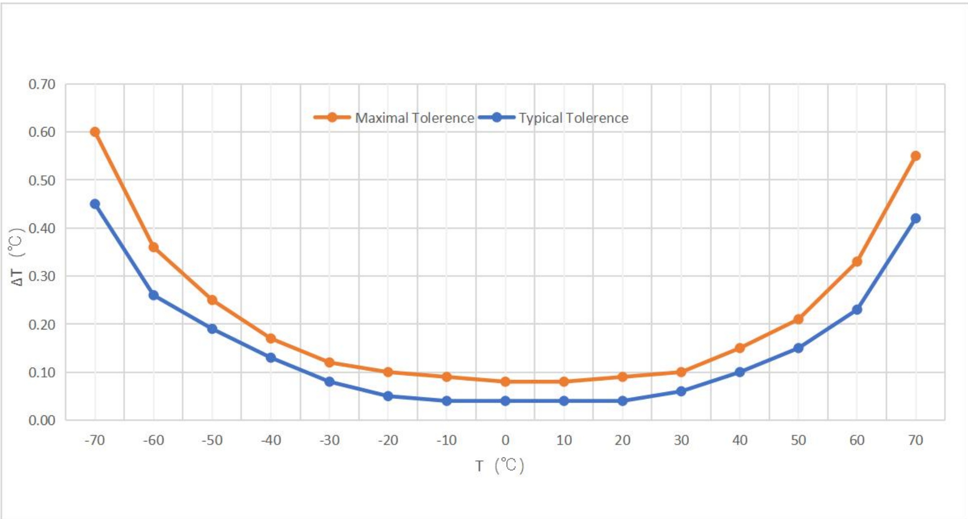



Figure 7-4 M117P accuracy curve

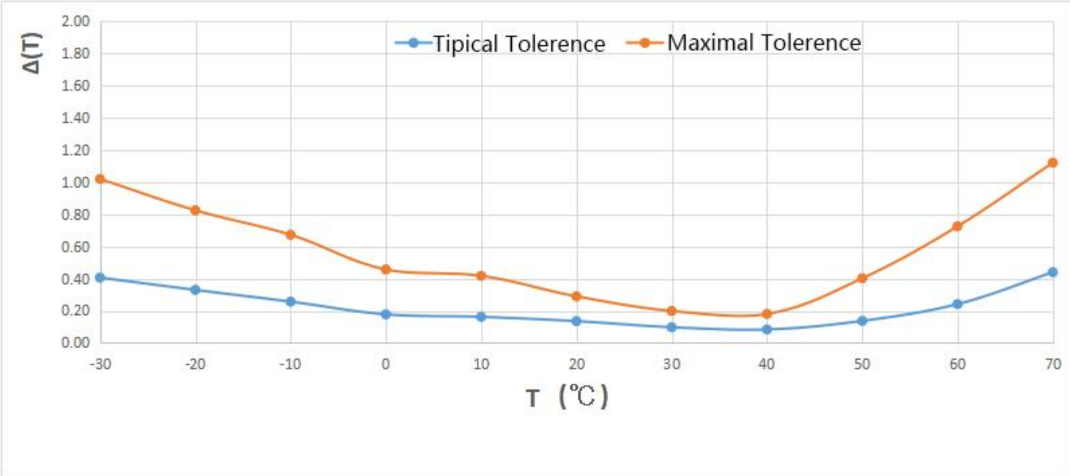



Figure 7-5 M117B accuracy curve

# 8. Electrical Characteristics

## 8.1. Absolute maximum rating

*These are only limit parameters and are not applicable to functional operation of the device in environments where this limit condition or higher. Long-term exposure to this extreme environment can affect device reliability.*

| Parameter | Value | Unit |
|---|---|---|
| Supply voltage | -0.3 to 6 | V |
| Pin-to-ground voltage | -0.3 to 6 | V |
| Input current | ±100 | mA |
| Working temperature | -70 to 150 | ℃ |
| Storage temperature | -70 to 150 | ℃ |
| Welding temperature | Refer to IPC/JEDEC J-STD-020 Spec. | |
| ESD HBM (Human body discharge mode) | ±8 | kV |

## 8.2. DC Electrical Characteristics

| Parameter | Symbol | Condition | Min | Typ | Max | Unit | Note |
|---|---|---|---|---|---|---|---|
| Voltage | | | | | | | |
| Supply voltage | VDD | | 1.8 | 3.3 | 5.5 | V | |
| Supply voltage swing | VDD slew | — | — | — | 20 | V /ms | The voltage change between |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| rate change | | | | | | | VDDmin and VDDmax on DD line should be slower than the maximum voltage swing rate, and a faster voltage swing rate may cause reset. |
| Supply current | IDD | Standby | — | 0.1 | 1 | uA | The current when no measurement |
| | | Periodic measurement mode | — | 55 | — | uA | No sleep in periodic measurement mode |
| | | Peak | — | 447 | — | uA | |
| | | Average | — | 5.2 | — | uA | Single mode, high repeatability, one measurement per second |
| Waiting time after powered-up | $t_{PU}$ | — | — | 2 | — | ms | |
| Digital input/output | | | | | | | |
| Input logic low | VIL | SCL, DQ/SDA | — | — | 0.3*VDD | V | |
| Input logic high | VIH | SCL, DQ/SDA | 0.7*VDD | — | — | V | |
| Output logic low | VOL | IOL = -3 mA | — | — | 0.4 | V | |
| Input leakage current | IIN | — | -0.1 | — | 0.1 | uA | |
| Alarm output drive strength | IOH | — | — | 1.5* VDD | — | mA | |
| Pull up resistor | Rup | | 1 | 10 | 100 | kΩ | |

## 8.3. AC Electrical Characteristics - Non-Volatile Memory

-55℃ to +125℃; $V_{DD}$=1.8V to 5.5V

| Parameter | Symbol | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Non-volatile write period | $t_{WR}$ | — | — | — | 40 | ms |
| EEPROM endurance | $N_{EEWR}$ | -55°C to +55°C | 50000 | — | — | times |
| EEPROM data retention | $t_{EEDR}$ | -55°C to +55°C | — | 10 | — | year |

## 9. Operation - Measuring Temperature

The thermometer is powered up in a low-power idle state. To initiate temperature measurement and analog-to-digital conversion, the host must issue the Convert T command. After the conversion, the generated temperature data is stored in the 2-byte temperature register. The conversion time is related to the repeatability setting. The higher the repeatability, the longer the conversion time. After power on, the default configuration is low repeatability. Refer to status register and configuration register for repeatability setting and time characteristic table for conversion time.

### 9.1. Temperature output and conversion formula

The digital output of temperature is 16bit signed binary complement, the resolution (LSB) is 1 / 256 ℃, and S is the sign bit. Data is stored in Temp_lsb and Temp_msb.

Table 9.1 Temperature register format

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **bit7** | **bit6** | **bit5** | **bit4** | **bit3** | **bit2** | **bit1** | **bit0** |
| **LS Byte** | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-8}$ |
| | **bit15** | **bit14** | **bit13** | **bit12** | **bit11** | **bit10** | **bit9** | **bit8** |
| **MS Byte** | **S** | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ |

The conversion relationship with Celsius is:

$$T\left[℃\right] = 40 + \frac{S_T}{256}$$

For example, 40 ℃ corresponds to register value 0x 00 00
150 ℃ corresponds to register value 0x 6E 00
-70 ℃ corresponds to register value 0x 92 00

## 9.2. Configuration register and status register

The configuration register contains Alarm Function, I2C clock stretch, cycle measurement frequency, and repeatability setting bits. The contents of the configuration register are described in the following table.

Table 9.2- 1 Configuration（Cfg）

| Bit | Content | Default Value |
|---|---|---|
| 7 | 1：Alarm Function ON<br>0：Alarm Function OFF | ‘0’ |
| 6 | Reserved | ‘0’ |
| 5 | Clock stretch function<br>0：Off<br>1：On | ‘0’ |
| 4:2 | MPS, Periodic measurement frequency configuration<br>000: single measurement<br>001: 0.5 times per second<br>010: once per second<br>011: 2 times per second<br>100: 4 times per second<br>101: 10 times per second | ‘000’ (Note 1) |
| 1:0 | Repeatability setting<br>00: low repeatability<br>01: medium repeatability<br>10: High repeatability | ‘00’ (Note 2) |

Note 1: The default periodic measurement configuration is single measurement, and the measurement frequency can be configured as required. The fastest speed is about 133 samples/second under low repeatability, 111 samples/second under medium repeatability, and 70 samples/second under high repeatability.
Another way to achieve the periodic measurement is to use the single measurement in a cyclical control function, the steps are shown as follows:
i. Calculate the duration of each measurement under the required measurement frequency
ii. Use duration of each measurement minus the combination of conversion time and one-wire command time, to get the delay value
iii. Put the delay value at the end of the cyclical function to achieve the required measurement frequency.
Note 2：Repeatability and conversion time are a direct compromise relationship. The higher the repeatability, the longer the conversion time; The lower the repeatability, the shorter the conversion time. See table 9.2-2 below.

Table 9.2- 2 Conversion time and repeatability settings

| Repeatability | | Precision | Conversion time $t_{CONV}$ |
|---|---|---|---|
| 0 | 0 | low | 4ms |
| 0 | 1 | medium | 5.5ms |
| 1 | 0 | high | 10.5ms |

The status register contains the status of the measurement command, the alarm status, and the status of the last command execution and the last write sequence. See table 9.2-3 for the content description of status register.

Table 9.2- 3 Status Register (Status)

| Bit | Description | | | Default |
|---|---|---|---|---|
| 7:6 | Reserved | | | '00' |
| 5 | $I^2C$ write data CRC | 0: Correct | 1: Wrong | '0' |
| 4 | $I^2C$ Command Status | 0: Wrong | 1: Correct | '0' |
| 3 | System reset detection | 0: Not detected | 1: Detected | '0' |
| 2 | Temperature alarm tracking | 0: Temperature alarm not triggered | 1: temperature alarm triggered | '0' |
| 1 | Reserved | | | '0' |
| 0 | Cycle measurement command status | 0: No measurement | 1: Measuring temperature | '0' |

## 9.3. Alarm

The alarm mode allows monitoring of ambient temperature conditions through programmable thresholds. When the threshold is reached, the output level of the dedicated alert pin will change. In addition, the status register bit has a special bit to indicate the alarm status. Use ALERT pin to control a switch. Or it can be connected to the interrupt pin of the microcontroller. After the sensor gives an alarm, the microcontroller can wake up from sleep mode and perform certain operations.

The alarm mode is activated as long as the sensor is measuring. Deactivate the alarm mode by setting the minimum setting to a value greater than or equal to the maximum setting (Tla_Set ⩾ Tha_Set).

The alarm threshold can be set by corresponding instructions (see I2C command set in section 11.1). Different thresholds are shown in the figure below.

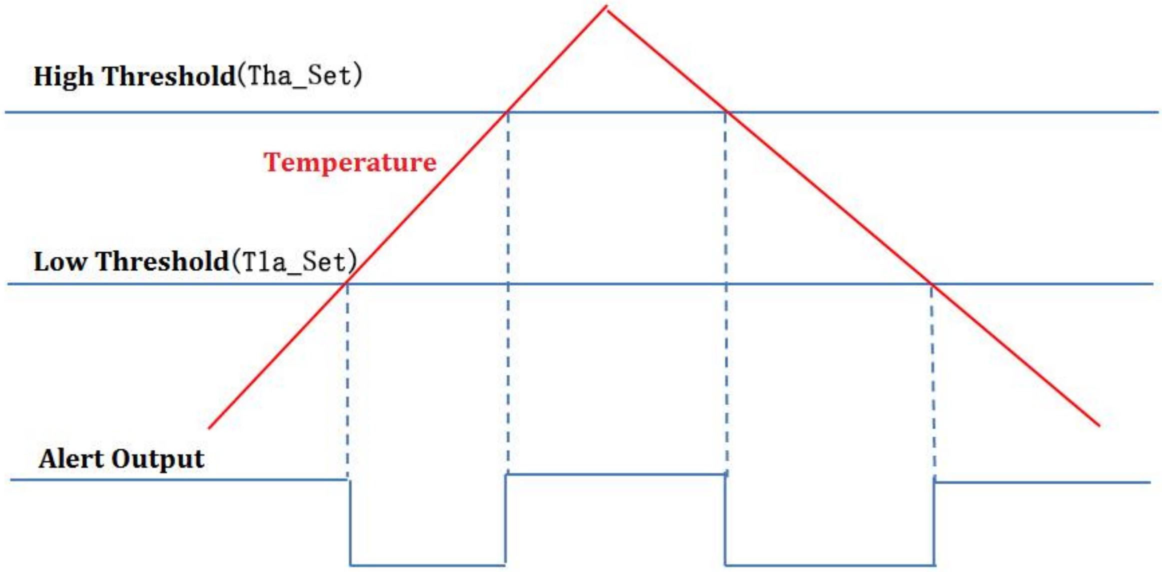



Figure 9.3- 1 Different thresholds for alarm mode

The alarm threshold is stored in a simplified format, that is, only the highest effective 9 bits are stored, and compared with the most effective 9 bits of the 16 bit standard output to determine whether the alarm conditions have been met. See the figure below. Therefore, the alarm threshold has a resolution different from the measured value. The resolution of temperature alarm threshold is Δ t ≈ 0.5° C. Note that data is always measured and stored in 16 bit format. The simplified data format is only used to determine whether the alarm conditions are met.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Highest 9 bits of temperature measured | | | | | | | | | X | X | X | X | X | X | X |

Fig. 9.3-2 Relevant data bits of alarm threshold

The requirement of alarm threshold limit setting is described hereunder:

Tha_Set>Tha_Clear>Tla_Clear>Tla_Set⩾40 or Tla_Set<Tla_Clear<Tha_Clear<Tha_Set<40

Here are three typical examples of how to calculate the threshold limit.

**Example #1，alarm threshold higher than 40℃**

1) Select the alarm threshold value of temperature:
High limit trigger threshold: Tha_Set = 60° C,
High limit clear trigger threshold: Tha_Clear = 55℃,
Low limit clear trigger threshold: Tla_Clear = 45℃,
Low limit trigger threshold: Tla_Set = 40°C

2) Use the following formula

$$S_T = (T - 40) \times 256$$

3) Convert the Tha_Set to a 16 bits binary value:

Tha_Set = 0001'0100'0000'0000

4) Delete the lower 7 bits of Tha_Set, and add 7 "0" before it

Tha_Set = 00000000001'0100'0~~000'0000~~ = 0000'0000'0010'1000

5) Thus, Tha_Set can be set as:

Tha_Set_msb=0000'0000, Tha_Set_lsb=0010'1000

6) Similarly, all other threshold limits can be set as:

Tha_Clear_msb = 0000'0000, Tha_Clear_lsb = 0001'1110
Tla_Clear_msb = 0000'0000, Tla_Clear_lsb = 0000'1010
Tla_Set_msb = 0000'0000, Tla_Set_lsb = 0000'0000

**Example #2，alarm threshold lower than 40**℃

1) Select the alarm threshold value of temperature:
High limit trigger threshold: Tha_Set = 39° C,
High limit clear trigger threshold: Tha_Clear = 34℃,
Low limit clear trigger threshold: Tla_Clear = 30℃,
Low limit trigger threshold: Tla_Set = 25°C

2) Use the following formula:

$$S_T = (T - 40) \times 256$$

3) Convert the Tha_Set to a 16 bit binary value:

Tha_Set = 1111'1111'0000'0000

4) Delete the lower 7 bits of Tha_Set, and add 7 "0" before it

Tha_Set = 00000001111'1111'0~~000'0000~~ = 0000'0001'1111'1110

5) Thus, Tha_Set can be set as:

Tha_Set_msb = 0000'0001, Tha_Set_lsb = 1111'1110

6) Similarly, all other threshold limits can be set as:

Tha_Clear_msb = 0000'0001, Tha_Clear_lsb = 1111'0100
Tla_Clear_msb = 0000'0001, Tla_Clear_lsb = 1110'1100
Tla_Set_msb = 0000'0001, Tla_Set_lsb = 1110'0010

**Example #3，disable alarm function**

1) Select the alarm threshold value of temperature:
High limit trigger threshold: Tha_Set = 38.5° C,
High limit clear trigger threshold: Tha_Clear = 34℃,
Low limit clear trigger threshold: Tla_Clear = 30℃,
Low limit trigger threshold: Tla_Set = 40°C,

2) Use the following formula:

$$S_T = (T - 40) \times 256$$

3) Convert the Tha_Set to a 16 bit binary value:

Tha_Set = 1111'1110'1000'0000

4) Delete the lower 7 bits of Tha_Set, and add 7 "0" before it

Tha_Set = 00000001111'1110'1~~000'0000~~ = 0000'0001'1111'1101

5) Thus, Tha_Set can be set as:

Tha_Set_msb = 0000'0001, Tha_Set_lsb = 1111'1101

6) Similarly, all other threshold limits can be set as:

Tha_Clear_msb = 0000'0001, Tha_Clear_lsb = 1111'0100
Tla_Clear_msb = 0000'0001, Tla_Clear_lsb = 1110'1100
Tla_Set_msb = 0000'0000, Tla_Set_lsb = 0000'0000

7) Since Tla_Set > Tha_Set，the alarm function is disabled.

## 10. Memory

The memory organization of the sensor is shown in table 10.1-1. The memory consists of a SRAM register and an extended non-volatile EEPROM register, which is used to store high and low alarm trigger values (TH and TL), configuration register and 2-byte user programmable EEPROM. It should be noted that if the alarm function of the sensor is not used, the TH and TL registers can be used as general storage. All stored instructions are described in detail in the sensor function instructions section. Register byte 0 Temp_lsb and byte 1 Temp_msb is a read-only temperature measurement. Bytes 2 and 3 are reserved registers. Byte 4, 5 Tha_Set_lsb, Tla_Set_lsb is a register for setting alarm threshold. Byte 6 CFG contains configuration register data, and byte 7 status is the status register of the chip, which is explained in detail in Chapter 9.2. Byte 8 is a read-only register and a cyclic redundancy check code generated by bytes 0 to 7. The sensor generates this code through the rules described in the cyclic redundancy check code generation section. Data can be written to bytes 4, 5, 6 with the write scratchpad [0x4e] instruction. Data transfer must start at the lowest bit of byte 2. To verify data integrity, read the registers (via the scratchpad [0xbe] instruction) after the data write operation. When reading the register, the data transmission of single bus starts from the lowest bit of byte 0.

To permanently write data from the register to EEPROM, the host must issue the copy scratchpad [0x48] command. The data in EEPROM register will be kept when power is off, and will be loaded to the corresponding register position automatically when power is on. The data can also be reloaded at any time through the Recall-EE [0xb8] instruction. Please note that the copy scratchpad [0x48] is operated by page (16 bytes). You need to make sure that all the 16 temporary contents needs to be confirmed correct and then copy uniformly.

Table 10.1 Thermometer memory storage mapping

<table>
<tr><th>Name</th><th>Logic Addr</th><th>Read</th><th>Write</th><th>E²PROM</th><th>Copy</th><th>Recall</th><th>Reset Value</th></tr>
<tr><td>Temp_lsb</td><td>00</td><td rowspan="10">Read one byte (0xD2--)</td><td>NA</td><td>NA</td><td>NA</td><td>NA</td><td>H’ 01</td></tr>
<tr><td>Temp_msb</td><td>01</td><td>NA</td><td>NA</td><td>NA</td><td>NA</td><td>H’ F1</td></tr>
<tr><td>Reserved</td><td>02</td><td rowspan="4">Write one byte (0x52--)</td><td>NA</td><td>NA</td><td>NA</td><td>H’ 00</td></tr>
<tr><td>Reserved</td><td>03</td><td>NA</td><td>NA</td><td>NA</td><td>H’ 80</td></tr>
<tr><td>Tha_Set_lsb</td><td>04</td><td>0</td><td rowspan="6">Copy Scratchpad (0xCC48)</td><td rowspan="3">Recall EE (0xCCB8)</td><td>H’ 00</td></tr>
<tr><td>Tla_Set_lsb</td><td>05</td><td>1</td><td>H’ 00</td></tr>
<tr><td>Cfg</td><td>06</td><td></td><td>2</td><td>H’ 02</td></tr>
<tr><td>Status</td><td>07</td><td>NA</td><td>NA</td><td rowspan="3">Recall Page0</td><td>XX</td></tr>
<tr><td>Crc_src</td><td>NA</td><td>NA</td><td>NA</td><td>XX</td></tr>
<tr><td>Tha_Clear_lsb</td><td>08</td><td></td><td>4</td><td>H ‘00</td></tr>
</table>

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Tla_Clear_lsb | 09 | | Write one | 5 | | Res | H '00 |
| Tha_Set_msb | 0A | | byte | 6 | | (0xCC | H '00 |
| Tla_Set_msb | 0B | | (0x52--) | 7 | | BB) | H '00 |
| Tha_Clear_msb | 0C | | | 8 | | | H '00 |
| Tla_Clear_msb | 0D | | | 9 | | | H '00 |
| Reserved | 0E~13 | | | 10-15 | | | H '00 |
| Crc_scr_ext | NA | | NA | NA | NA | NA | XX |

If larm function is disabled, the MSB of Cfg register must be set to 0. When the MSB of Cfg register is 0, registers of Tha_Set_lsb, Tla_Set_lsb, Tha_Clear_lsb, Tla_Clear_lsb, Tha_Set_msb, Tla_Set_msb, Tha_Clear_msb and Tla_Clear_msb can be used as storage registers for users. These registers can be read/written by using command Read_scratchpad(0xbe), Read_scr_ext(0xdd), Write_scratchpad(0x4e) and Write scr_ext(0x77).

# 11. I²C bus

## 11.1. I²C command

| Command | Function | Code |
|---|---|---|
| Measurement | Convert Temperature | 0xCC44 |
| Read measurement results | — | — |
| Set the upper threshold of alarm threshold | WRITE_ALERT_HI_SET | 0x611D |
| Set alarm clear threshold | WRITE_ALERT_HI_UNSET | 0x6116 |
| Set alarm threshold lower threshold | WRITE_ALERT_LO_SET | 0x6100 |
| Set alarm clear threshold lower threshold | WRITE_ALERT_LO_UNSET | 0x610B |
| Read the upper threshold of alarm threshold | READ_ALERT_HI_SET | 0xE11F |
| Read the upper threshold of alarm clear | READ_ALERT_HI_UNSET | 0xE114 |
| Read the lower threshold of alarm | READ_ALERT_LO_SET | 0xE102 |

| | | |
|---|---|---|
| Read alarm threshold lower threshold clear | READ_ALERT_LO_UNSET | 0xE109 |
| Write configuration register | CONFIG | 0x5206 |
| Read status and configuration registers | READ_STATUS | 0xF32D |
| Clear status register | CLEAR_STATUS | 0x3041 |
| Stop periodic measurement | BREAK | 0x3093 |
| Software reset | SOFT_RST | 0x30A2 |
| Save Page0 to EEPROM | COPY_PAGE0 | 0xCC48 |
| Restore EE area | RECALL_EE | 0xCCB8 |
| Restore the residual area of Page0 | RECALL_PAGE0_RES | 0xCCB6 |

## 11.2. Operation and communication

The sensor supports I2C fast mode (up to 400 kHz). Clock stretch can be enabled and disabled by the corresponding user command. After sending a command to the sensor, a minimum waiting time of 1ms is required before the sensor receives another command. All sensor commands and data are mapped to a 16 bit address space. In addition, data and commands include CRC verification, which improves communication reliability. The data sent and received by the sensor always follows the 8-bit CRC checksum. In the write operation, the host must send the checksum to the slave, and the sensor can only accept the data after receiving the correct checksum. During the read operation, the main device reads and processes the checksum.

| Parameter | Value |
|---|---|
| Name | CRC-8 |
| Width | 8-bit |
| Data Protect | W/R |
| Polynomial | 0x31 ($x^8 + x^5 + x^4 + 1$) |
| Initial Value | 0xFF |
| Whether each byte of the test data reversed by bit | No |
| After calculation and before exclusive or output, whether the whole data is reversed by bit | No |
| The calculation result is exclusive or to this parameter | 0x00 |
| Example | CRC (0xBEEF) = 0x92 |

Note: ROM ID last two bytes are 0x00 without CRC. CRC can be generated by Maxim CRC-8 polynomial above in MCU software.

## 11.3. Power on and communication starts

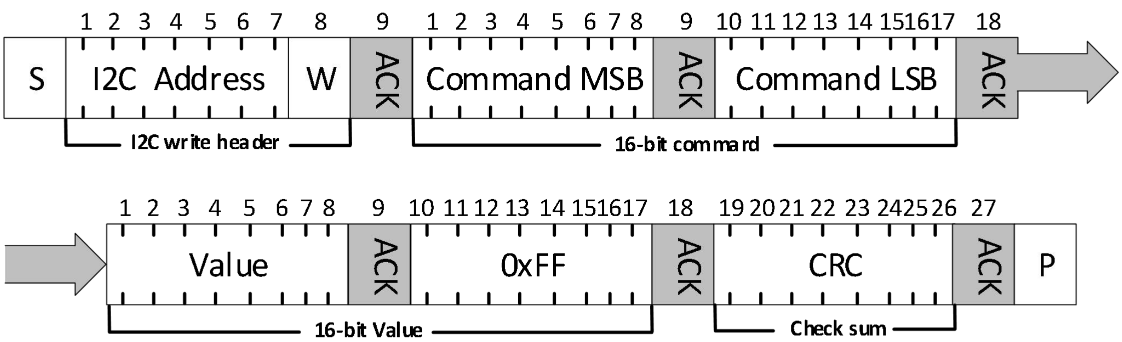
After power on, the sensor needs time for $T_{PU}$ to enter idle state. Once idle, commands can be received from the main device (microcontroller). Each transmission sequence starts with a start condition (s) and ends with a stop condition (P), as described in the I2C bus specification. Whenever the sensor is powered up but no measurement or communication is performed, it automatically goes into an idle state to save energy. The idle state cannot be controlled by the user.

## 11.4. Start measurement

The measurement communication sequence includes start condition, I2C writing head (7-Bit I2C device address plus 0 as the writing bit) and 16 bit measurement command. The sensor indicates the correct reception of each byte. It pulls the SDA pin down (ACK bit) after the falling edge of the 8th SCL clock to indicate reception. The complete measurement cycle is described in section 12.8. By confirming the measurement command, the sensor begins to measure the temperature. In addition, measurement repeatability and single / continuous measurement mode are set by corresponding control bits of configuration register.

## 11.5. Set configuration register

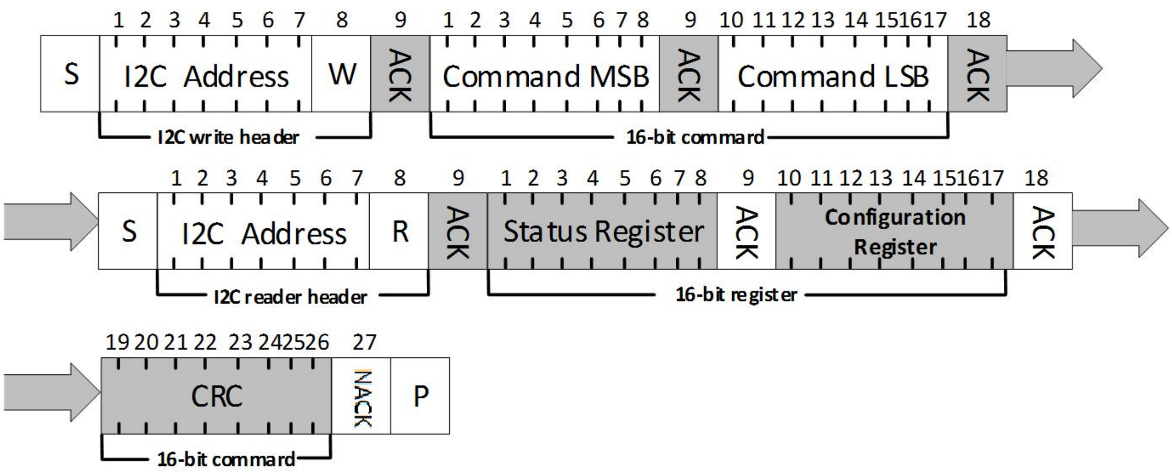
| Command | Hex Code |
|---|---|
| Set configuration register | 0x5206 |




## 11.6. Read status register and configuration register

The commands to read the status register and configuration register are shown in the following table.

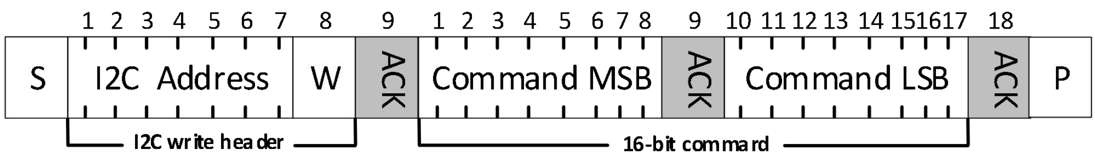
| Command | Hex Code |
|---|---|
| Read status and configuration register | 0xF32D |




## 11.7. Reset status register

The flag (bits 3,2,1) in the status register can be cleared (set to zero) by sending the command shown in the following table.

| Command | Hex Code |
|---|---|
| Reset status register | 0x3041 |

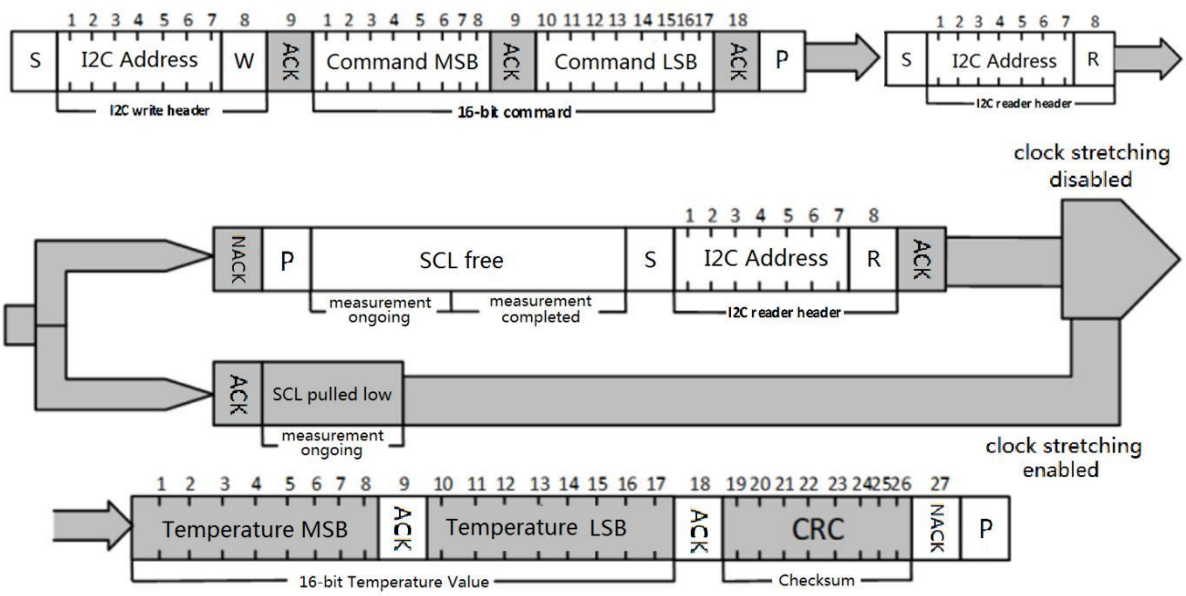



*Note: the white block is controlled by the microcontroller, and the sensor response is gray block.*

## 11.8. Single temperature measurement

In the mode, an issued measurement command triggers a temperature data acquisition. Each temperature data is a signed 16 bit binary number. During transmission, each data value always follows the CRC checksum. The 16 bit command is shown in the following table. Repeatability (low, medium and high) and clock stretch (enabled or disabled) can be achieved by modifying the configuration register. The repeatability setting affects the measurement duration, thus affecting the total energy consumption of the sensor. It is explained in Section 9.2.

| Command | Hex Code |
|---|---|
| Temperature measurement | 0xCC44 |




*Note: the first "SCL free" block indicates the temperature conversion time(See Table 9.2-2) , the white block is controlled by the microcontroller, and the sensor response is gray block.*

## 11.9. Reading data in single temperature measurement mode

After the sensor completes the measurement, the host can read the measurement results by sending the start condition and then sending the I2C reading head. After the reading temperature command is sent, the sensor will confirm the reception of the reading header and send two bytes of data (temperature), followed by a byte of CRC checksum. Each byte must be acknowledged by the host and have an ACK condition for the sensor to continue sending data. If the sensor does not receive an ACK from the host after any data bytes, it will not continue to send data.

After receiving the check sum of the temperature data, the non-acknowledge (NACK) and stop conditions shall be sent (see 12.8). If you are not interested in the subsequent data, the I2C master can suspend the read transmission after any data bytes in a Non ACK condition. For example, do not read the second byte or CRC byte of the measurement result to save time.

**Clock stretch mode off**

When the clock stretch is turned off, after the measurement command is issued, if the temperature measurement is not finished, the sensor will respond to the non-response (NACK) reading header.

**Clock stretch mode on**

The clock stretch mode is enabled by setting bit 5 of the configuration register to 1. When the clock stretch is turned on, after the measurement command is issued, the sensor responds to the read header through the ACK, and then pulls down the SCL line until the measurement is completed. Once the measurement is completed, the sensor will release the SCL line and send the measurement result.

### 11.10. Continuous temperature measurement mode

The sensor continuous measurement mode is enabled by setting bits 2, 3 and 4 of the configuration register. They differ in repeatability (low, medium, and high) and data acquisition frequency (0.5, 1, 2, 4, and 10 measurements per second). Clock stretch is off in this mode. Data acquisition frequency and repeatability settings affect the measurement duration and current consumption of the sensor. This is explained in Section 9.2.

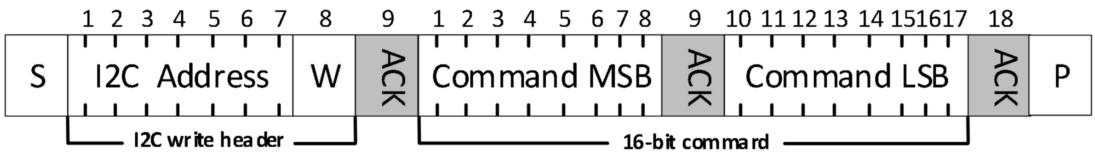
### 11.11. Read data in continuous measurement mode

In the continuous measurement mode, the host can read the measurement results by sending the start condition and then sending the I2C reading head.

### 11.12. Stop continuous measurement mode

The stop command shown in the following table to stop the continuous measurement mode. After receiving the stop command, the sensor will stop the measurement in progress but will not enter the single measurement mode. After sending the measurement command again, it will enter the continuous measurement mode again. If you want to switch to the single measurement mode, you need to set the control bits related to the configuration register.

| Command | Hex Code |
|---|---|
| Stop continuous measurement | 0x3093 |




*Note: the white block is controlled by the microcontroller, and the sensor response is gray block.*

### 11.13. Reset

The system reset signal of the sensor can be generated Internally by issuing a command of soft reset. In addition, during power on, the chip internal generation system is reset initially. During reset, the sensor does not process commands.

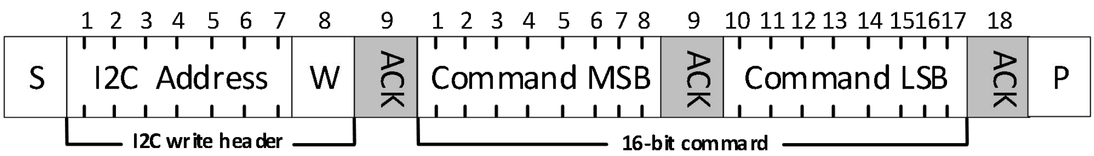
### 11.13.1. Reset interface

If the communication with the device is abnormal, the following signal sequence will reset the serial interface: switch the SCL nine or more times when the SDA remains high. This method must follow the transfer start sequence before the next command. This sequence only resets the interface and the status register retains its contents.

### 11.13.2. Soft reset/Re-initialization

The sensor provides a soft reset mechanism, which forces the system to enter a clearly defined state under the condition of continuous power supply. When the system is idle, the soft reset command can be sent to the sensor. This triggers the sensor to reset its system controller and reload calibration data from memory. In order to start the soft reset process, the commands shown in the following table should be sent.

| Command | Hex Code |
|---|---|
| Software reset | 0x30A2 |

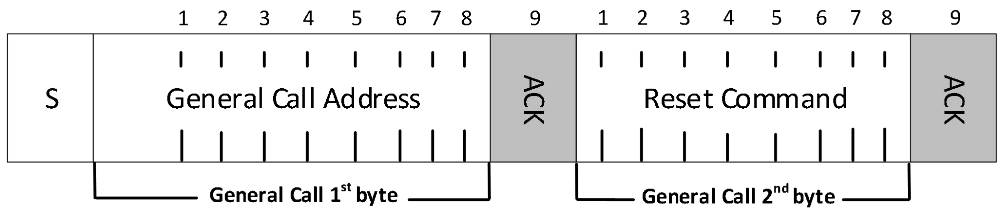



*Note: the white block is controlled by the microcontroller, and the sensor response is gray block.*

### 11.13.3. General reset

According to the I2C bus specification, the "general call" mode can also be used to generate a sensor reset. A reset generated in this way is not device specific. All devices on the same I2C bus that support universal call mode will perform a reset. In addition, this command is only valid if the sensor is able to process the I2C command. The appropriate command consists of two bytes, as shown in the following table.

| Command | Hex Code |
|---|---|
| Address Byte | 0x00 |
| The second byte | 0x06 |
| The universe reset call command | 0x0006 |




*Note: the white block is controlled by the microcontroller, and the sensor response is gray block.*

### 11.13.4. Hard reset

The hard reset is realized by turning off the power supply voltage of VDD pin and then turning it on again. To prevent the sensor from being powered by the ESD diode, the external voltages of pins 1 (SCL) and pin 6 (SDA) need to be removed.

## 11.14. Read and write alarm threshold command

| Command | Hex Code |
|---|---|
| Read alarm set upper threshold | 0xE11F |
| Read alarm clear upper threshold | 0xE114 |
| Read alarm lower threshold | 0xE102 |
| Read alarm clear lower threshold | 0xE109 |

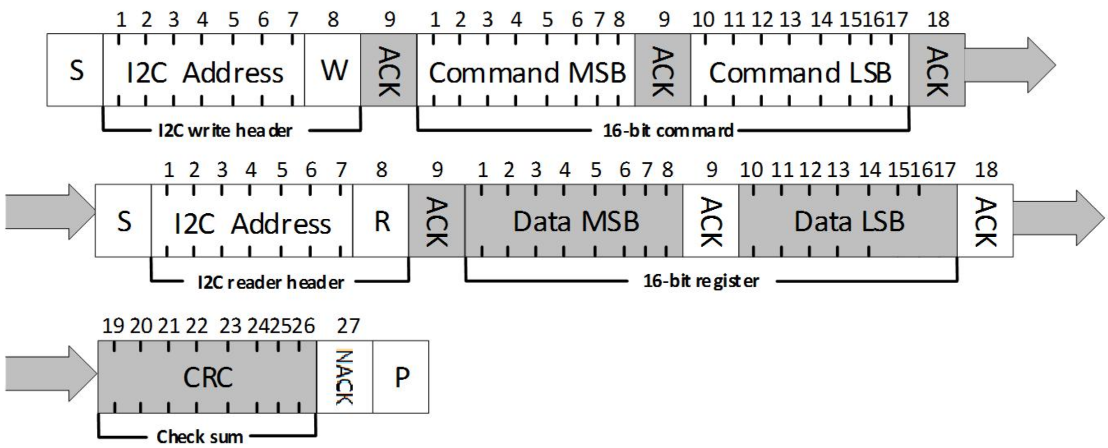



Fig. 11.14-1 Read alarm threshold command（*Note: the white block is controlled by the*

*microcontroller, and the sensor response is gray block.*)

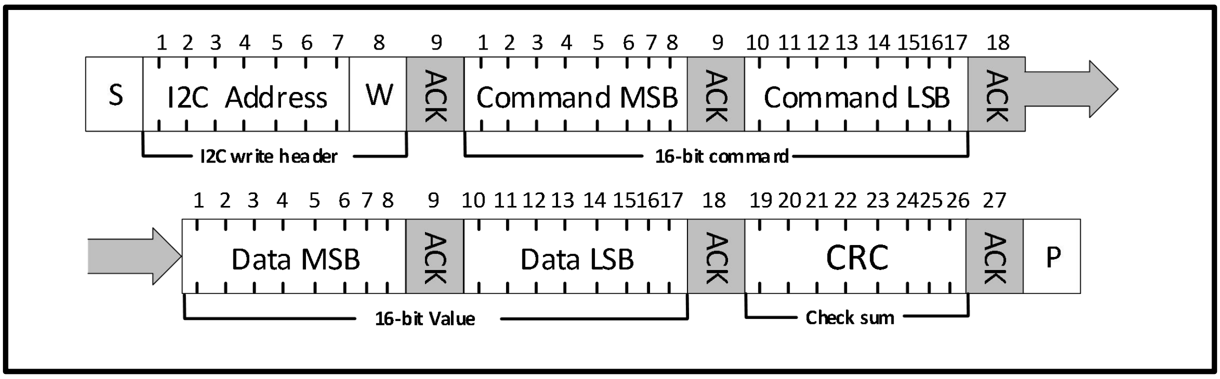



Fig. 11.14-2 Write alarm threshold limit command（*Note: the white block is controlled by the microcontroller, and the sensor response is gray block.*)

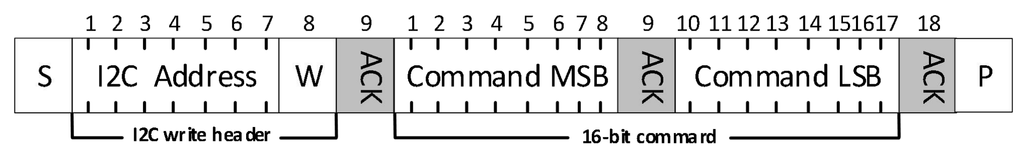
## 11.15. Register store and recovery commands

All data written to the register (working configuration and alarm threshold) is temporary. If you want these settings to stay permanently, that is, they can still be saved after power failure, you need to use the copy command to store these data in EEPROM.

When hard reset or soft reset, the system automatically loads EEPROM data into the register. In addition, data can be recovered from EEPROM to register with recovery instruction.

Table 11.15 Register store and recovery commands

| Command | HEX Code |
|---|---|
| Copy to $E^2$PROM | 0xCC48 |
| Restore EE area | 0xCCB8 |
| Restore the rest area other than EE | 0xCCB6 |




## 11.16. I²C sequence characteristics

Table 11.16 I²C bus timing characteristics (1)

| Parameter | Symbol | Standard mode | | Fast mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| SCL frequency | $f_{SCL}$ | 0 | 100 | 0 | 400 | kHz |
| SCL low level time | $t_{LOW}$ | 4.7(7) | — | 1.3(7) | — | μs |

| | | | | | | |
|---|---|---|---|---|---|---|
| SCL high level time | $t_{HIGH}$ | 4.0 | — | 0.6 | — | μs |
| Time interval from SCL pull down to SDA data change | $t_{HD;STA}$ | 4.0<br>30(6) | — | 0.6<br>30(6) | — | μs |
| Time interval from SDA data stabilization to SCL pull up | $t_{HD;DAT}$ | 5.0<br>0(2) | —<br>3.45(3) | —<br>0(2) | —<br>0.9(3) | μs<br>μs |
| High level holding time of SDA and SCL at the beginning of restart | $t_{SU;DAT}$ | 250 | — | 100(4) | — | ns |
| High level holding time of SDA and SCL at the beginning of restart | $t_{SU;STA}$ | 4.7 | — | 0.6 | — | μs |
| Time interval from SCL pull to SDA pull during stop | $t_{SU;STO}$ | 4.0 | — | 0.6 | — | μs |
| Interval between start and stop | $t_{BUF}$ | 4.7 | — | 1.3 | — | μs |
| Time required for rising edge of SCL / SDA | $t_R$ | — | 1000 | $20+0.1C_b$(5) | — | ns |
| Time required for falling edge of SCL / SDA | $t_F$ | 2.5 | 300 | $20+0.1C_b$(5) | — | ns |

*Note 1: all values are based on VIHmin and VILmax.*
*Note 2: the device must increase the holding time of at least 300ns internally to SDA.*
*Note 3: the maximum $t_{HD;DAT}$ must be met only when the SCL low level cycle (TLOW) is not stretched.*
*Note 4: fast device mode I2C device can work in standard mode, but the requirements of $t_{SU;DAT}$= 250ns must be met.*
*Note 5: CB = total capacitance of I2C bus.*
*Note 6 : For application with sleep-mode enable, to guarantee the device can be waked-up successfully, the minimum value of $t_{HD;STA}$ is reverse proportional to VDD level. The minimum value of $t_{HD;STA}$ is 30us when VDD=5V, 50us when VDD=3V, and 150us when VDD=1.8V, respectively. The details are described in FAQ.*
*Note 7: For application with sleep-mode enable, if the master uses hardware-based I2C and the parameter $t_{HD;STA}$ cannot be changed, to guarantee the device can be waked-up successfully, the minimum value of the first $t_{LOW}$ (the time from SCL first pull-low to SCL first pull-high after I2C start) is reverse proportional to VDD level. The minimum value of $t_{LOW}$ is 30us when VDD=5V, 50us when VDD=3V, and 150us when VDD=1.8V, respectively. The details are described in FAQ.*

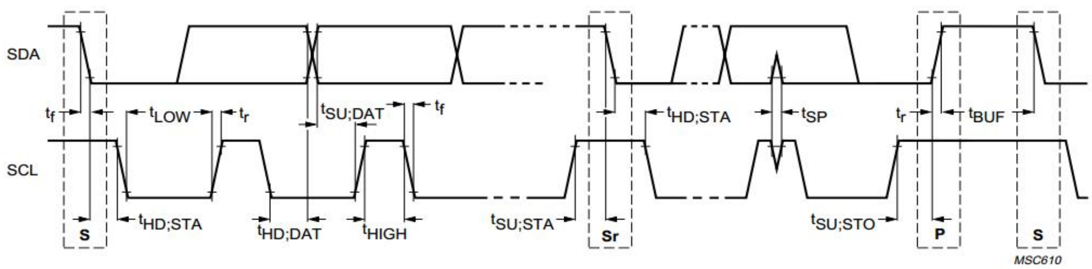
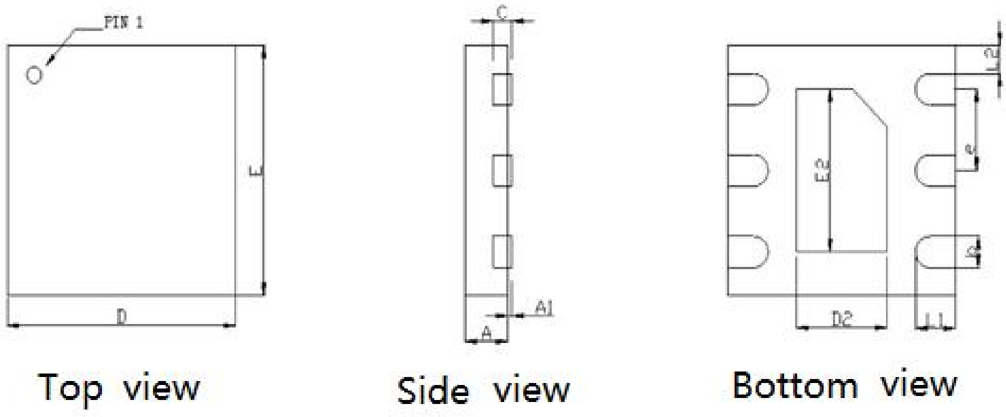



Fig. 11.16 I$^2$C sequence

## 12. Package and PCB pad pin size diagram

DFN6L(2X2X0.75mm)




| Symbol | Millimeters | | | Inches | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | 0.70 | 0.75 | 0.80 | 0.028 | 0.030 | 0.031 |
| A1 | 0.00 | 0.02 | 0.02 | 0.000 | 0.001 | 0.002 |
| b | 0.22 | 0.25 | 0.28 | 0.009 | 0.010 | 0.011 |
| C | 0.20 | | | 0.008 | | |
| D | 1.90 | 2.00 | 2.10 | 0.076 | 0.080 | 0.084 |
| D2 | 0.77 | 0.80 | 0.83 | 0.030 | 0.032 | 0.033 |
| E | 1.90 | 2.00 | 2.10 | 0.076 | 0.080 | 0.084 |
| E2 | 1.27 | 1.30 | 1.33 | 0.050 | 0.052 | 0.053 |
| e | 0.650 BSC | | | 0.026 | | |
| L1 | 0.30 | 0.35 | 0.40 | 0.012 | 0.014 | 0.016 |
| L2 | 0.100 | 0.150 | 0.200 | 0.004 | 0.006 | 0.008 |

Fig. 12.1 Package size diagram DFN6L(2X2X0.75mm)

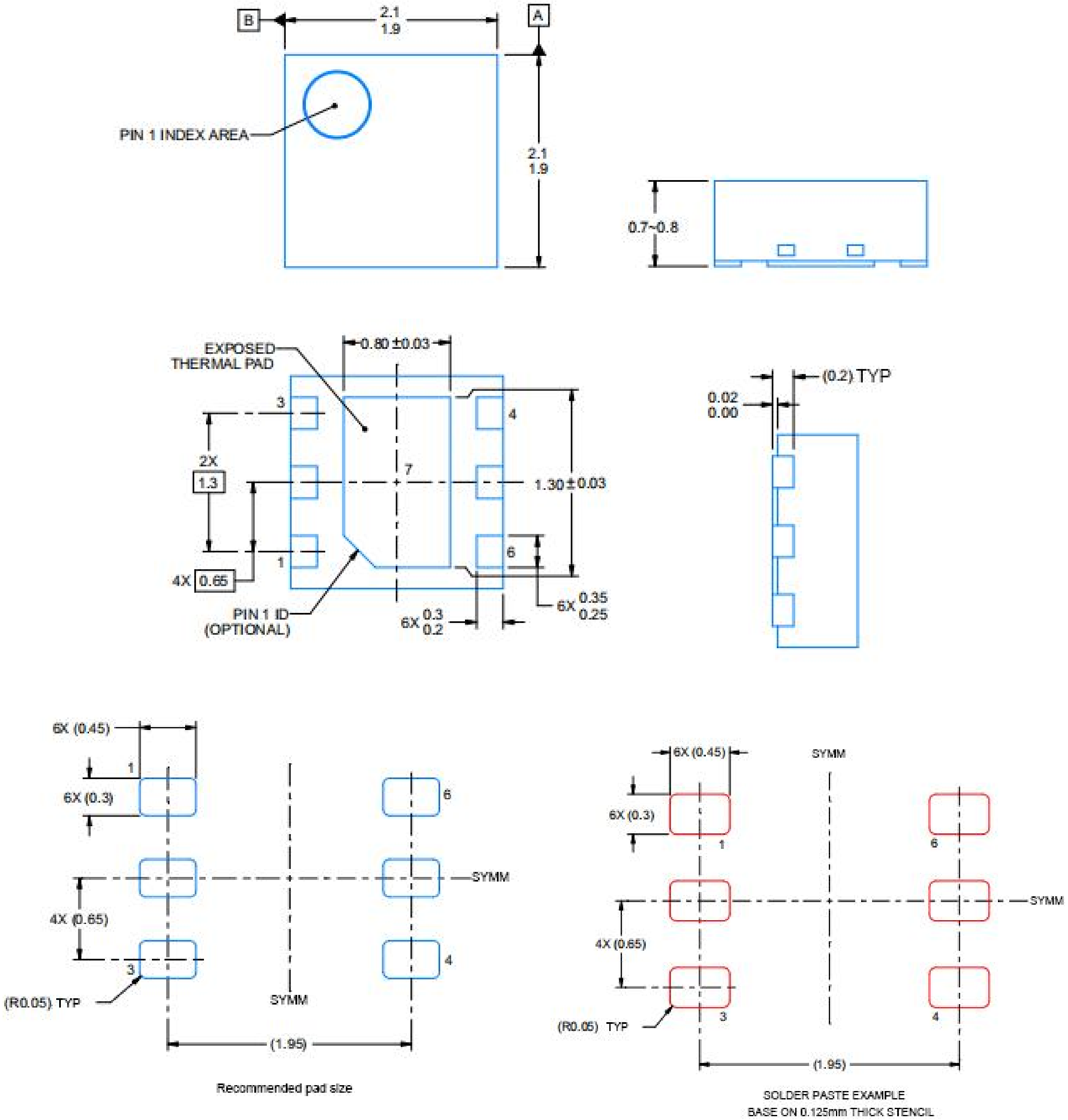



Fig. 12.2 PCB pad pin size diagram