OPENFIELD™
TECHNOLOGY
MICRO INSTRUMENTS FOR HARSH ENVIRONMENTS

# Detailed electrical specifications for the

# Pressure and Temperature sensor with I²C communication.

This document details the hardware interface and communication scheme to program and retrieve data from a Pressure and Temperature sensor with standard I²C communication.

## Contents

## 1 Power supply

| | Min | Typical | Max |
|---|---|---|---|
| V power | 3.0V | 3.6V | 6.5V |
| I current consumption | 0.3mA (sleep) | 5mA (monitor) | 6mA (startup) |

*Table 1 : power supply specifications.*

## 2 Communication hardware

| | Min | Typical | Max |
|---|---|---|---|
| Input low voltage | 0V | 0.6V | 1V |
| Input high voltage | 2.3V | 3V | 3.3V |
| Output low voltage | 0V | 0.4V | 0.8V |
| Output high voltage | 2.5V | 3V | 3.3V |
| I2C Clock | 10kHz | 100kHz | 800kHz |

*Table 2 : communication specifications.*

### I²C communication

The sensor uses a 2-wire I²C communication scheme, and is adressed as a slave. An additional white wire is used for firmware upgrades.

| Signal | Name | Wire color | Comments |
|---|---|---|---|
| Vcc | Power | **Red** | |
| GND | Ground | **Black** | |
| SCL | I²C clock | **Brown** | |
| SDA | I²C data | **Grey** | |
| RTX | Download | White | Not part of the I²C communication. |

*Table 3 : I²C wires.*




| Detailed electrical specifications. | | |
|---|---|---|
| **#ES15007-AA** | **Pressure and Temperature sensor I²C communication** | |
| Date : 20/07/2015 | Order : - | Version: AB |
| Owner : E. Tavernier | | |


## 3 Communication scheme

Each sensor acts as an I²C slave and is identified on the I²C bus by its 7-bit address. The default address upon shipment is 0x10. This address can be modified using the appropriate configuration register.

Data can be read from, and written to, the sensor through registers. Registers are divided in two types : configuration registers and data registers. The following table gives a summary of the available registers and their address. Configuration registers are identified by a light gray background.

| Register name | Address | Length (bits) | Type | Default value | Comments |
|---|---|---|---|---|---|
| Serial number | 0x01 | 16 | R | 0xXXXX | |
| Status | 0x02 | 8 | R/W | 0x01 | Write to perform a RESET, toggle Analog output, ... |
| I2C address | 0x03 | 8 | R/W | 0x32 | 7 bits value. Check I²C specifications for reserved values. |
| Reserved | 0x04 | 16 | R/W | 0x8000 | Not used. |
| Statistics time | 0x05 | 16 | R/W | 1000 | Maximum number of milliseconds over which the bubble count, holdup, and levels are computed. |
| Reserved | From 0x06 to 0x0A | 16 | R | 0x0000 | Do not use to stay compatible with further versions. |
| User registers | From 0x0B to 0x0F | 16 | R/W | 0x0000 | Arbitrary data can be written in these registers. Value is persistent over power cycling. |
| | | | | | |
| Reserved | From 0x010 to 0x15 | 16 | R | 0x0000 | Do not use to stay compatible with further versions. |
| Pressure low bytes | 0x16 | 16 | R | 0x8000 | 16-bit unsigned integer |
| Pressure high bytes | 0x17 | 16 | R | 0x00 | 16-bit unsigned integer |
| Temperature low bytes | 0x18 | 16 | R | 0x00 | 16-bit unsigned integer |
| Temperature high bytes | 0x19 | 16 | R | 0xFFFF | 16-bit unsigned integer |

*Table 4 : Configuration and data registers for Pressure and Temperature sensor.*

Pressure and temperature measurement are stored in two registers each, so that each value is represented by 32 bits. To retrieve the measured value, apply the following formula :

$$Pressure(Psi) = \frac{Pressure_{LowBytes} + 65536 \times Pressure_{HighBytes}}{2^{16}}$$




| Detailed electrical specifications. | | |
|---|---|---|
| **#ES15007-AA** | **Pressure and Temperature sensor I²C communication** | |
| Date : 20/07/2015 | Order : - | Version: AB |
| Owner : E. Tavernier | | |


$$Temperature(°C) = \frac{Temperature_{LowBytes} + 65536 \times Temperature_{HighBytes}}{2^{23}}$$

### Retrieving data

To read a data register, the device is first addressed as a slave receiver, given the register address to read, then addressed as a slave transmitter and read until no more data is available. The steps are outlined below :

- Send slave address and with Write bit
- Slave ACKs the request
- Send register address (0x16 for the low bytes of the pressure measurement)
- Slave ACKs the request
- Send Start Repeat with slave address and Read bit
- Slave ACKs the request
- Slave sends first data byte
- Master ACKs the data
- Slave sends second data byte
- Master NACKs to signify that enough data has been retrieved
- Master STOPs the transmission.

This scheme is illustrated fig 1 where a sensor whose address is 0x10 is polled for the current pressure measurement. The shaded parts are sent by the master, while the clear parts are sent by the slave.

| Start | Slave address (7 bits) | W | Ack | Register address | Ack | Start repeat | Slave address (7 bits) | R | Ack | Pressure low LSB | Ack | Pressure low MSB | NAck | Stop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | 0x10 | 0 | A | 0x10 | A | Sr | 0x10 | 1 | A | 0xXX | A | 0xXX | N | P |

*Fig 1 : Sequence illustrating a read of the pressure measurement low bytes.*

If the register address is invalid, the slave sends 0x00 and NACKs the transmission.

### Writing configuration data

To write to a configuration register, the device is addressed as a slave receiver, given the register address to write to, then the values to write, MSB first. The steps are outlined below :

- Send slave address and with Write bit
- Slave ACKs the request
- Send register address
- Slave ACKs the request
- Send most significant byte
- Slave ACKs the data
- Slave least significant byte
- Slave ACKs the data
- Master STOPs the transmission.

This scheme is illustrated fig 2 where a sensor whose address is 0x10 is sent a Reset command. The shaded parts are sent by the master, while the clear parts are sent by the slave.

| Start | Slave address | W | Ack | Register address | Ack | Reset command | Ack | Stop |
|---|---|---|---|---|---|---|---|---|
| S | 0x10 | 0 | A | 0x02 | A | 0x80 | A | P |

*Fig 2 : Sequence illustrating a reset command.*

Unlike a register read, there is no automatic incrementation of the register address, in order to reduce the risk of unintentional register write. If a data-write is attempted on a register which is either read-only, or whose length is shorter than the data sent (trying to write 2 bytes into an 8-bit register), the slave will NACK the transmission and discard the superfluous data.




| Detailed electrical specifications. | | |
|---|---|---|
| #ES15007-AA | Pressure and Temperature sensor I²C communication | |
| Date : 20/07/2015 | Order : - | Version: AB |
| Owner : E. Tavernier | | |


### Status register

The status register is used to send commands to the sensor and read its configuration. It is illustated fig 3.

| Bit | 7 | 6 .. 1 | 0 |
|---|---|---|---|
| Action | Reset (write only) | Reserved for future use | Reserved |
| Default value | 0 | 000000 | 1 |

*Fig 3 : Status register.*

### Reset

Upon setting bit 7 of the status register (write 0x80 in register 0x02), the microcontroler performs a RESET. This can also be triggered using the General Call address (0x00) followed by the Software Reset command (0x06).
The result is similar to a power cycling.

### Address update

In order to change the I2C address of the sensor, it is advised to connect it alone to the master, perform the write command to the address register (0x03), and check that the new address is enforced.

### Reserved addresses

The I2C specification reserves the following address values :

| Slave address | R/W bit | Description |
|---|---|---|
| 0000 000 | 0 | General call address |
| 0000 000 | 1 | START byte |
| 0000 001 | X | CBUS address |
| 0000 010 | X | Reserved for different bus format |
| 0000 011 | X | Reserved for future purposes |
| 0000 1XX | X | Hs-mode master code |
| 1111 1XX | 1 | Device ID (not implemented) |
| 1111 0XX | X | 10-bit slave adressing (not implemented) |

*Table 5 : Reserved addresses as specified in the I²C implementation.*

X = don't care, 1 = HIGH, 0 = LOW.

However, apart from the general call, the sensor address can be defined as other reserved values as long as it does not interfere with other features of the local bus.