Application note

Differential Pressure Sensor
Type D6F-PH (Rev 1.0)

Introduction

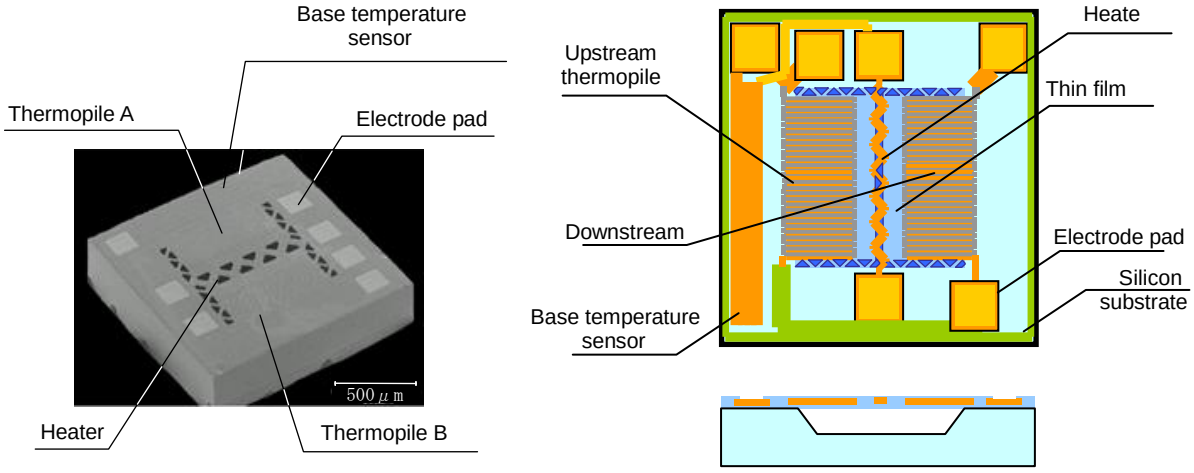
This document provides application information for the thermal flow sensor.

This is preliminary and may be changed without any notices.

Contents

1. Principle of operation

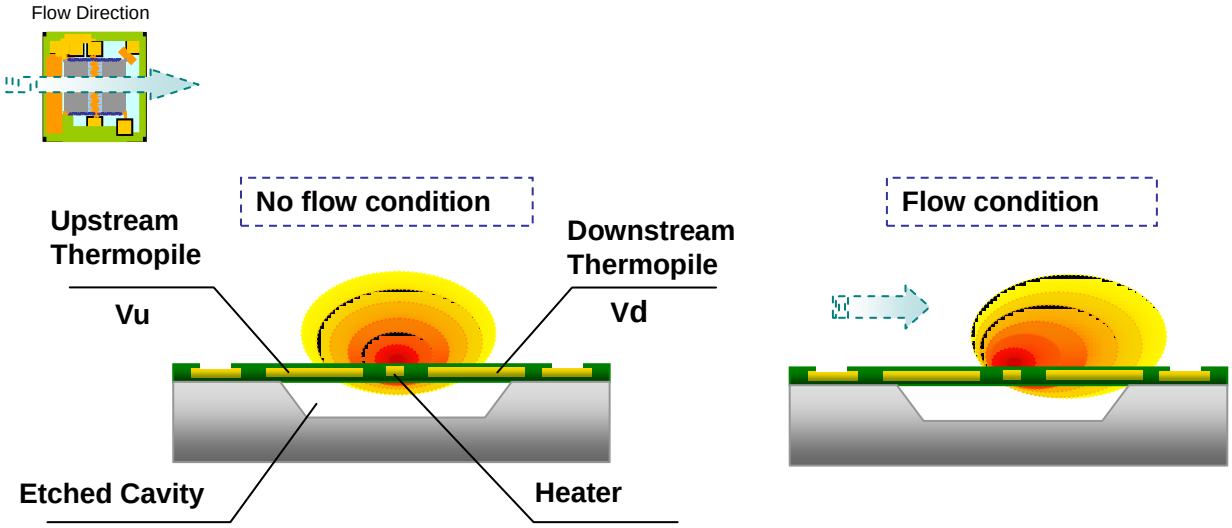
1.1 Flow sensor chip structure



1.2 Thermal mass flow sensing

Temperature difference between two thermopiles is approximately proportional to square root of the mass flow across the sensor chip. It's mass flow sensing and output sensitivity depends on gas composition.

When flow direction is perpendicular to the heater and the thermopiles.



Vd=Vu
Symmetrical temperature distribution

Vout = Voff + (Vd - Vu) × gain

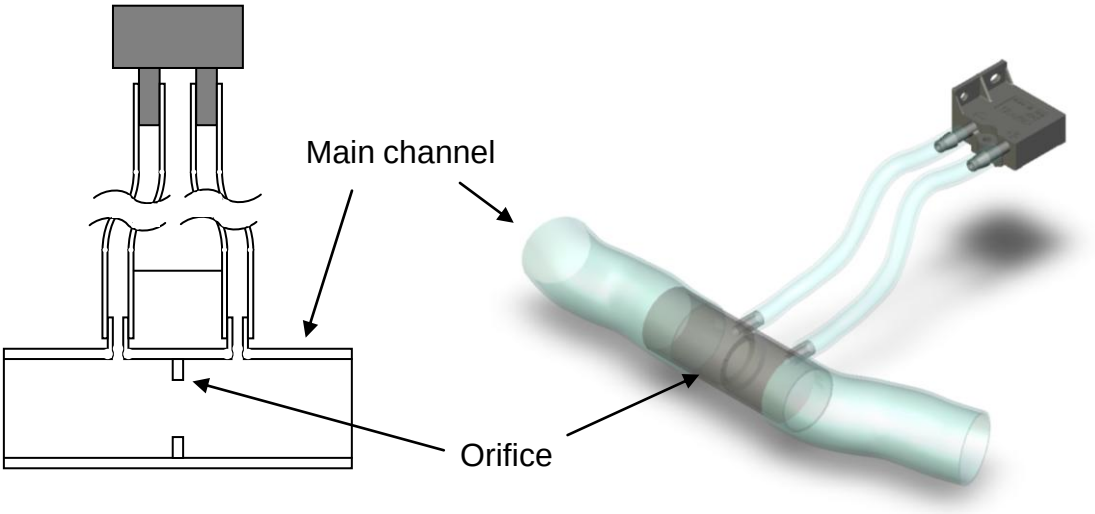
Vout : Output voltage, Voff : Offset voltage

Vd-Vu ∝ √Flow rate

Vd≠Vu (Vd > Vu)
Temperature of down stream becomes higher than that of upstream

1.3 Flow sensor mechanical construction

Case of measuring the differential pressure generated by the orifice of the main channel.

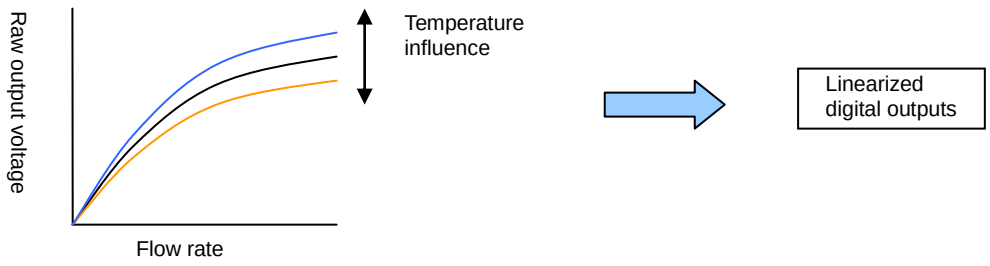


1.4 Linearization compensation and temperature compensation

Raw output voltage from flow sensor chip is approximately proportional to the square root of gas flow rate. Raw output voltage has influenced by ambient temperature.

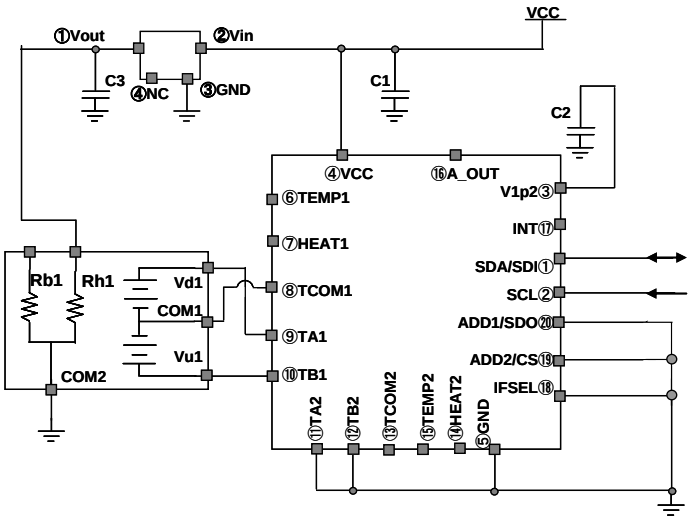
The flow sensor has sophisticated compensation functions.

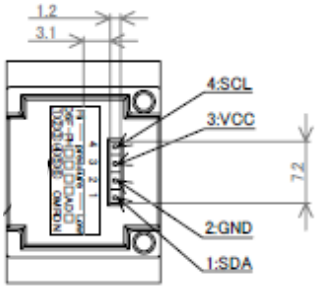
1) Linearization
2) Temperature compensation

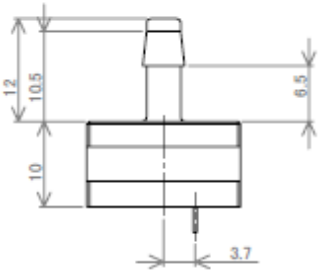


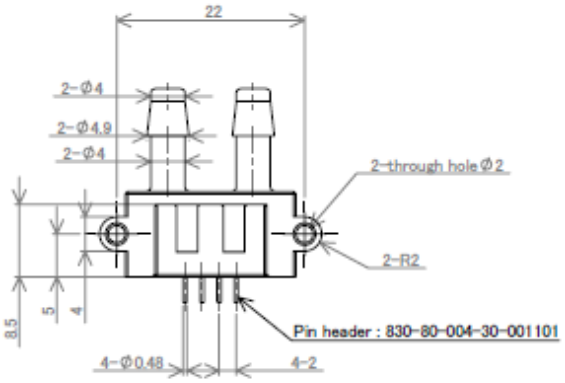
2 Hardware schematic diagram

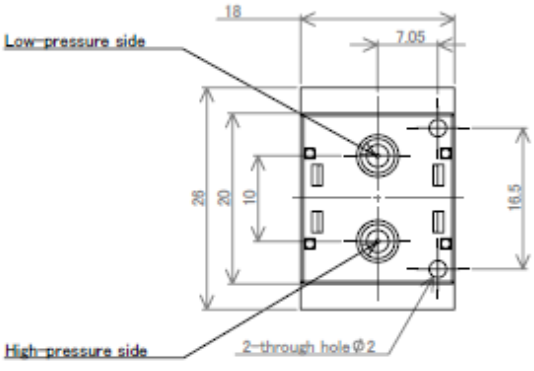
2.1 Schematic

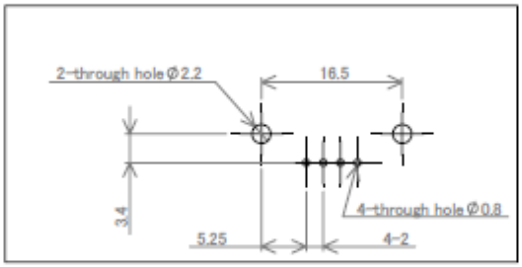












Recommendation size for pin header installation (tolerances : ±0.1)

2.2 ASIC block schematic

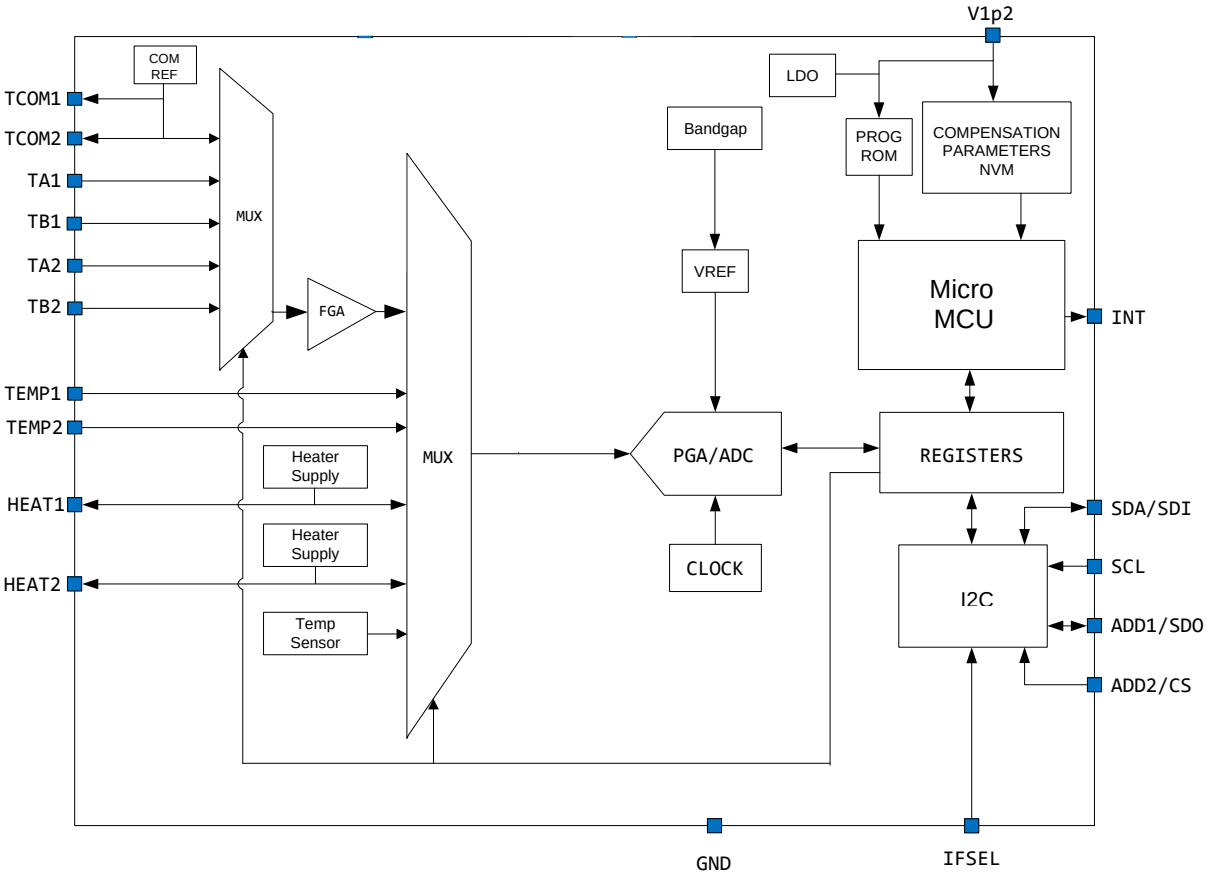


Table 1. Pin description

Pin Name	Description
SDA	I2C Data
SCL	I2C Clock
V1P2	Optional bypass capacitor connection (10 nF capacitor)
VCC	Supply Voltage
GND	Ground
TEMP1	sensor thermistor
HEAT1	sensor heater supply
TCOM1	sensor thermopiles common
TA1	sensor thermopile A
TB1	sensor thermopile B

3 Register map

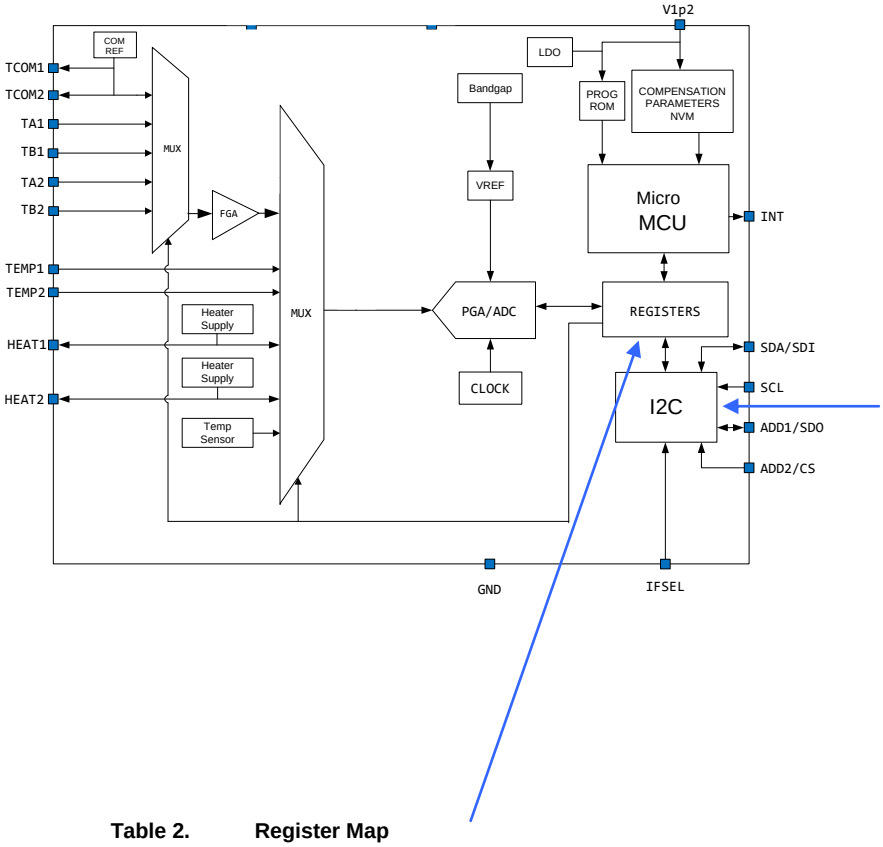


Table 1. I2C Configuration Register

Configuration Address	Function
00h	Access Address1
01h	Access Address2
02h	Serial Control
03h	Write Buffer 0
04h	Write Buffer 1
05h	Write Buffer 2
06h	Write Buffer 3
07h	Read Buffer 0
08h	Read Buffer 1
09h	Read Buffer 2
0Ah	Read Buffer 3

Table 2. Register Map

Address	Register Name	Register Description
D040h	SENS_CTRL	Sensor Control
D046h	FLAGS	Flags
D051h	COMP_DATA1_H	Compensated Data
D052h	COMP_DATA1_L	
D061h	TMP_H	Internal Temperature Data
D062h	TMP_L	

4 I2C interface

4.1 Over view

Table 3. Overview of the Communication function

Item	explanation
Communication system	I2C
SCL clock frequency	Max 400kHz (TBD)
Output format	Binary data (MSB first) *MSB: Most Significant Bit
Slave address	1101100b or 0x6C

* Please note that slave address is 7 bits.

4.2 Interface Configuration Registers

The memory and register access is controlled by writing to the interface configuration registers.

Table 4. Serial Interface Configuration Registers

Configuration Address	Function	Note
00h	Access Address 1 (MSB)	MSB of the first address to write to
01h	Access Address 2 (LSB)	LSB of the first address to write to
02h	Serial Control	Miscellaneous Control
03h	Write Buffer 0	Data to be written at Address
04h	Write Buffer 1	Data to be written at Address + 1
05h	Write Buffer 2	Data to be written at Address + 2
06h	Write Buffer 3	Data to be written at Address + 3
07h	Read Buffer 0	Data read from Address
08h	Read Buffer 1	Data read from Address + 1
09h	Read Buffer 2	Data read from Address + 2
0Ah	Read Buffer 3	Data read from Address + 3

4.3 Access Address Register (00h – 01h)

The access address registers are used to access internal register blocks including sensor register map and ADC register map and internal memory. It specifies the data transfer start address with auto increment for multiple byte data transfer.

Table 5. Access Address Register

Address	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
00h	A15	A14	A13	A12	A11	A10	A9	A8
01h	A7	A6	A5	A4	A3	A2	A1	A0

Table 6. Serial Control Register (02h)

The serial control register contains various bits to modify the behavior of the serial access.

Address	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
02h	D_byte_cnt3	D_byte_cnt3	D_byte_cnt3	D_byte_cnt3	Req	R_WZ	Acc_ctl2	Acc_ctl2

- Acc_ctl2 [2] – Access control:
  00 = 16 bit address (A15-A0) access (ROMS and registers)
  01 = 8 bit address (A7-A0) access and used to access MCU internal 256 byte dual port RAM
  10 = reserved
  11 = reserved
- R_WZ [1] – Read/ Write Select
  0 = write access
  1 = read access
- Req [1]
  0 = the previous request is done
  1 = new request. After serial bus bridge controller finishes a request, it will clear req to 0. For write requests the bridge controller moves the data in write data buffers to the location pointed by access address. For read requests the bridge controller stores the read data into read data buffer.
- D_byte_cnt3 [3]
  Transfer data byte count. It only supports 1, 2, 3, 4 data byte transfer.

4.4 Write Buffer Registers (03h – 06h)

There are four write data buffer registers at address: 03h – 06h. To perform a write, host can either use a single command or perform separate writes to the following addresses.

Host can write to sensor register map in single byte transfer: Host can burst write data start from address = 00h with the following data byte: A[15:8], A[7:0], 18h, data[0].

Or host can do four serial bus writes and write one data byte into serial bus register with the following steps:
Write A[15:8] to address = 00h
Write A[7:0] to address = 01h
Write data[0] to address = 03h
Write 18h to address = 02h (1 byte, new request, write access)
(Optional) Read Serial Control register (02h). If req=0 (02h[3]), controller is finished with write.

4.5 Read Buffer Registers (07h – 0Ah)

There are four read data buffer registers at address: 07h – 0Ah. To perform a read, host can either use a single streaming command or perform separate commands to the following addresses.

After read request is done by internal serial bus bridge controller, req bit is clear to 0 and read data is stored in rd_buf1 – rd_buf4 (address = 07h – 0Ah).
For single byte read request host can burst write A[15:8], A[7:0], 1Ch at start address = 00h. Host needs to read command register until req bit is cleared to 0, then read “read data buffer” for read data at address = 07h.

Host can perform a single byte read by individually programming the following registers.
Write A[15:8] to address = 00h
Write A[7:0] to address = 01h
Write 1Ch to address = 02h (1 byte, new request, read access)
(Optional) Read address = 02h. If req=0 (02h[3]), controller is finished with read
Read data[0] from address = 07h

4.6 I2C Command examples

I2C Command Examples

・I2C Command：I2C write

Start Address of Configuration resister

These data will be written in Configuration resister “00h” & “01h”

START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	40h	A

Serial Ctrl	ACK	Write Data	ACK	STOP
18h	A	06h	NA	P

These data will be written in Configuration resister “02h” & “03h”

・I2C Command：I2C read

Start Address of Configuration resister (Read Buffer 0)

START	Slave Address	ACK	Access Address	ACK
S	D8h (6Ch (7b)+ 0)	A	07h	A

These data will be read in Configuration resister “07h” & “08h” (Read Buffer 0 / Read Buffer 1)

Re-Start	Slave Address	ACK	Read Data H	ACK	Read Data L	ACK	STOP
RS	D9h (6Ch (7b)+ 1)	A	xxh	A	xxh	NA	P

5 Register table

In this section, each register's function is indicated by colors as follows.

Write access	MCU	HOST & MCU	NONE

5.1 Sensor Control (D040h)

Table 7. SENS_CTRL

ADDR	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
D040h						MS	DV_PWR [1]	DV_PWR [0]
Default	0	0	0	0	0	0	0	0

- DV_PWR[2] – Main Device Power Mode
  00 = Standby – All blocks are powered down.
  10 = MCU on – Used when only MCU is required. Basic analog and memories are powered on and MCU clock is running.
  NOTE: This register should not be changed during a measurement.
- MS[1] – MCU Start – Begin execution of MCU mode based on the state of DV_PWR.
  - 0 = Stop
    Sequences are stopped and blocks are turned off.
  - 1 = Start
    The MCU clock is started and the MCU mode is executed.
    If main device power mode is standby, all blocks are shutdown and sequence is stopped.

5.2 Flags (D046h)

Table 8. FLAGS

ADDR	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
D046h				OS2	OS1	HV2	HV1	SV
Default				0	0	0	0	0

- SV[1] – Supply Voltage (VDD) Flag
  - 0 = Supply voltage is within specification.
  - 1 = Supply voltage is outside of specification.
- HV1[1], - Heater Voltage
  - 0 = Heater voltage is within specification.
  - 1 = Heater voltage is outside of specification.
- OS1[1],– Open Sensor
  - 0 = Sensor is connected.
  - 1 = Sensor is not connected.
- HV2 and OS2 are researved bit. These bits shall be 0.

***** Please read it twice when reading the flag.

5.3 Data Registers (D04Fh-D068h)

Table 9. 16-Bit Data Registers Map

Address	Register Name	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0	Description
D051h	COMP_DATA1_H	DATA<15:8>								Compensated Data
D052h	COMP_DATA1_L	DATA<7:0>								
D061h	TMP_H	DATA<15:8>								Internal, Temperature Data
D062h	TMP_L	DATA<7:0>								
D065h	REF_FLOW1_H	DATA<15:8>								Sensor Reference Flow Data
D066h	REF_FLOW1_L	DATA<7:0>								
D067h	THRESH_FLOW1_H	DATA<15:8>								Sensor Threshold Flow Data
D068h	THRESH_FLOW1_L	DATA<7:0>								

Please refer to section6. Understanding output data.

6 Understanding output data

The measured data are sent to the dedicated registers. These registers contain, respectively, the most significant part and the least significant part of the sensing signals acting on each sensor.
For example, the compensated flow data is given by the concatenation COMP_DATA1_H & COMP_DATA1_L in unsigned 16bit number. And the raw flow data is given by the concatenation UCFM1_H & UCFM1_L and it is expressed as a 2's complement of signed 16bit number.

6.1 Data alignment

Sensing data are represented as 16-bit numbers. The data are split and sent to consecutive two byte of Flow Sensor Registers in "Big Endian" format.
"Big Endian" means that the high-order byte of the number is stored in register at the lowest address, and the low-order byte at the highest address.

6.2 Register content

・COMP_DATA1_H & COMP_DATA1_L [D051h – D052h] : Compensated Data (unsigned)
The values stored in these registers represent the measured flow data with the selected compensation. Compensation mode applied to the measured flow value can be switched by [COMP] setting in Compensation Control register (D048h).

・TMP_H & TMP_L [D061h – D062h] : Temperature Data (signed)
The values stored in these registers represent the temperature data measured by the internal temperature sensor of the ASIC.
The following formula can be applied to convert register data into temperature value.

$$Tv\ [degC] = (Rv - 10214) / 37.39$$

Where, Tv = Converted temperature value in degC format, and Rv = Register content stored in the Temperature Data register.

Note: Temperature data is for referential use ONLY. And its accuracy is not specified in the device specifications.

6.3 Example of sensing data

The below tables provide a few basic examples of the data that is read in Flow Sensor Registers when the device is subject to a given flow and temperature. The values listed in the tables are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,).

Table 10. Temperature Data registers content vs. Temperature value

Register address		Register values		Converted temperature values
TMP_H	TMP_L			
D061h	D062h	HEX	DEC	
2Bh	8Dh	2B8Dh	11149	25.0 degC
2Eh	FFh	2EFFh	12031	48.6 degC
26h	BBh	26BBh	9915	-8.0 degC

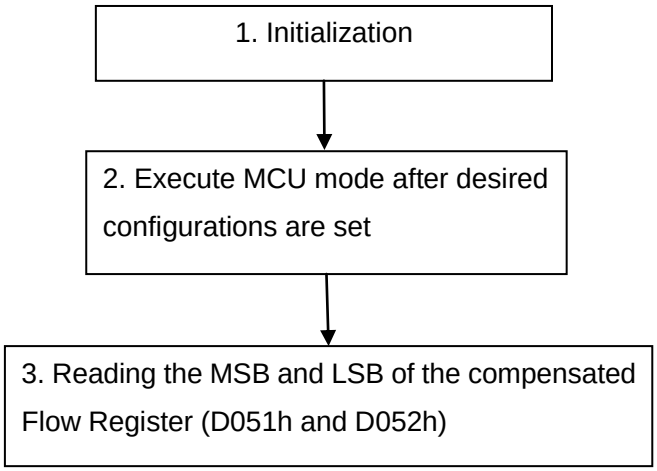
7 Measurement Mode

7.1 Measurement operation mode

Measurement operation mode can be selected by MCU mode setting in MCU Operation State register as described below.

Mode	Description	Settings	Conditions
31	Compensated flow data	Resolution : 12bit	

8 MCU mode 31 flow chart



9 MCU mode 31 I2C Instruction

1. Initialization after power up [Must be done]

I2C command: Write 00h to the Control Register (Bh) to load NVM trim values, but keep MCU in non-reset state.

START	Slave Address	ACK	Access Address	Write Data	ACK	STOP
S	D8h (6Ch (7b)+ 0)	A	0Bh	00h	N	P

2. Execute MCU mode after desired configurations are set

Writing 06h to the Sensor Control Register (D040h) will execute the MCU mode outlined in Section 6 with the configured settings for the Compensation, etc. Reading the Sensor control register after writing a 06h will show the MUX selection chosen by the MCU. MS bit will be set = 0 after execution.

[Caution]: Do not read or write to the Device while MCU is executing. It would be safe to read/write only after 30ms.

I2C command: Write 06h(MS=1 & MCU_on) to the SENS_CTRL Register (D040h).

START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	40h	A

Serial Ctrl	ACK	Write Data	ACK	STOP
18h	A	06h	NA	P

5. Reading the MSB and LSB of the compensated Flow Register (D051h and D052h)

I2C command: Write 2Ch (2bytes read) to the Serial Control Register (2h) to read from compensated Flow Register (D051h and D052h).

START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	51h	A

Serial Ctrl	ACK	STOP
2Ch	NA	P

I2C command: Read the compensated Flow Registers 2bytes through the Read buffer 0 (7h) and Read buffer 1 (8h).

START	Slave Address	ACK	Access Address	ACK
S	D8h (6Ch (7b)+ 0)	A	07h	A

Re-Start	Slave Address	ACK	Read Data H	ACK	Read Data L	ACK	STOP
RS	D9h (6Ch (7b)+ 1)	A	xxh	A	xxh	NA	P

10 I^2C bus characteristics

The bus is intended for communication between different Ics. It consists of two lines: a bidirectional data signal (SDA) and a clock signal (SCL). Both the SDA and SCL lines must be connected to a positive supply voltage via a pull-up resistor. The following protocol has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high.
- Changes in the data line, while the clock line is high, are interpreted as control signals.

Accordingly, the following bus conditions have been defined:

10.1 Bus not busy

Both data and clock lines remain high.

10.2 Start data transfer

A change in the state of the data line, from high to low, while the clock is high, defines the start condition.

10.3 Stop data transfer

A change in the state of the data line, from low to high, while the clock is high, defines the stop condition.

10.4 Data valid

The state of the data line represents valid data when after a start condition. The data line is stable for the duration of the high period of the clock signal. The data on the line may be changed during the low period of the clock signal. There is one clock pulse per bit of data. Each data transfer is initiated with a start condition and terminated with a stop condition. The number of data bytes transferred between the start and stop conditions is not limited. The information is transmitted byte-wide and each receiver acknowledges with a ninth bit.

By definition a device that gives out a message is called “transmitter,” the receiving device that gets the message is called “receiver.” The device that controls the message is called “master.” The devices that are controlled by the master are called “slaves.”

10.5 Acknowledge

Each byte of eight bits is followed by one acknowledge bit. This acknowledge bit is a low level put on the bus by the receiver whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed is obliged to generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter.

The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is a stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master receiver must signal an end of data to the slave transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this case the transmitter must leave the data line high to enable the master to generate the STOP condition.

Table 11. **Serial bus data transfer sequence**

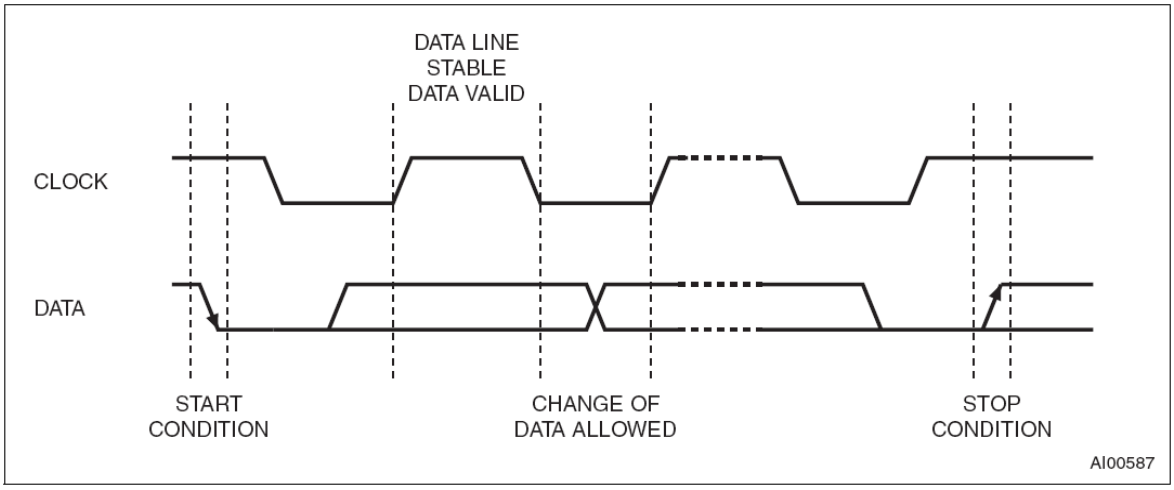
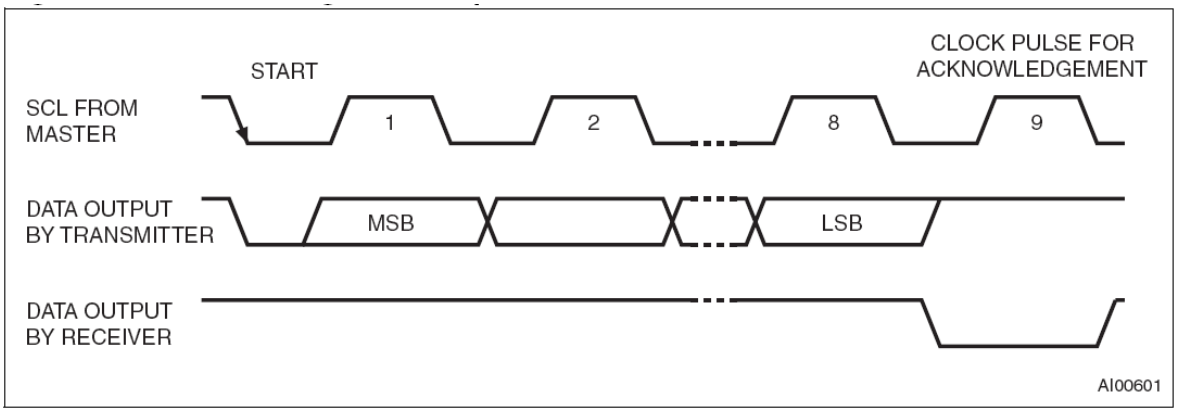


Table 12. **Acknowledgement sequence**



10.6 I²C Read Mode

In this mode the master reads the Digital Flow Sensor slave after setting the slave address (see Figure 8). Following the WRITE mode control bit (R/W = 0) and the acknowledge bit, the word address ‘An’ is written to the on-chip address pointer. Next the START condition and slave address are repeated followed by the READ mode control bit (R/W = 1). At this point the master transmitter becomes the master receiver. The data byte which was addressed will be transmitted and the master receiver will send an acknowledge it to the slave transmitter. The address pointer is only incremented on reception of an acknowledge clock. The Digital Flow Sensor slave transmitter will now place the data byte at address An+1 on the bus, the master receiver reads and acknowledges the new byte and the address pointer is incremented to An+2.

Table 13. **Slave address location**

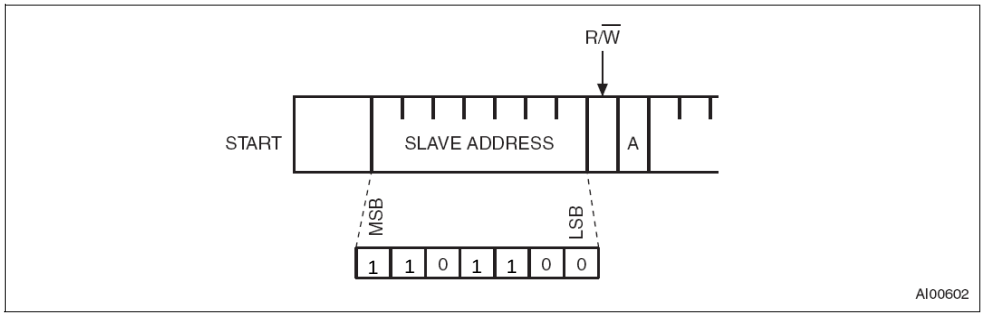


Table 14. **Read mode sequence**

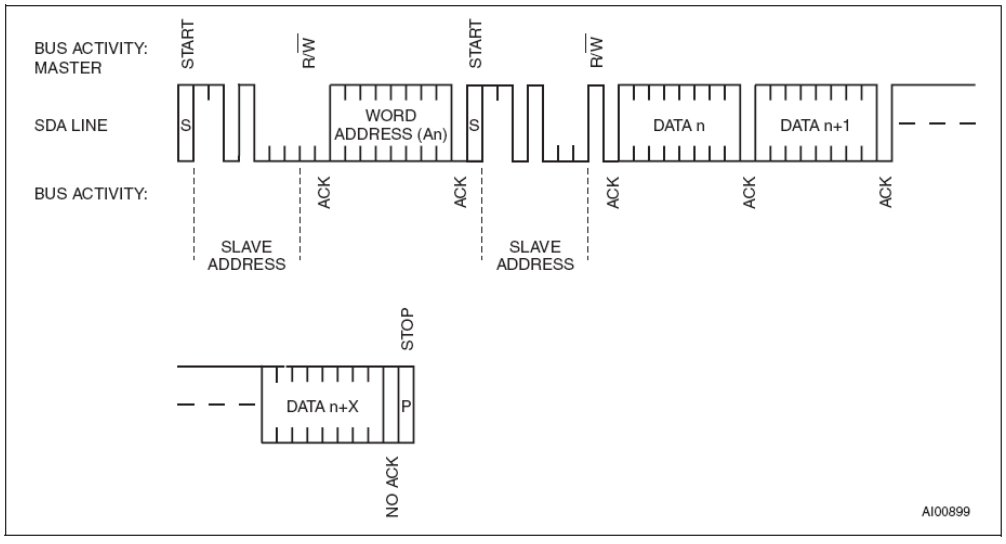
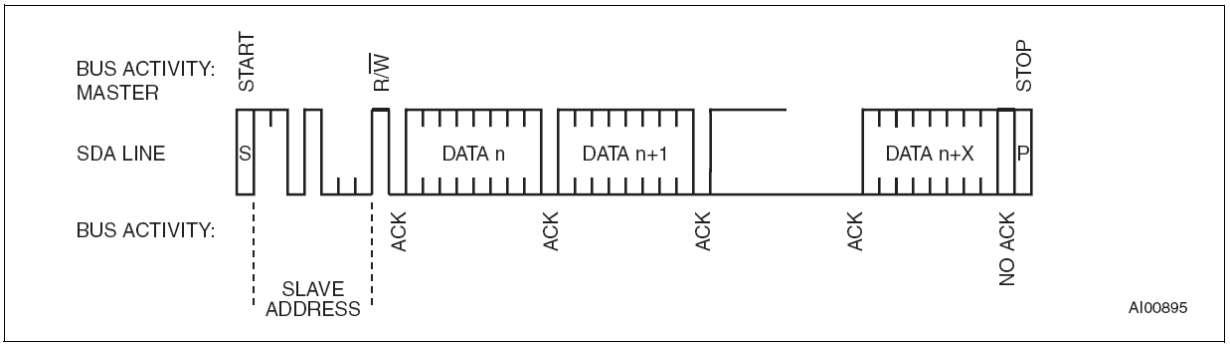


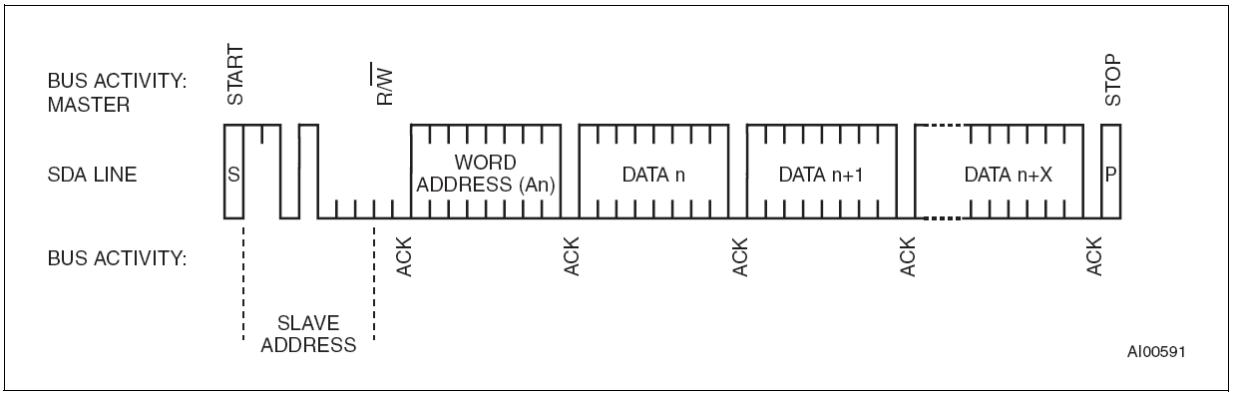
Table 15. **Alternate read mode sequence**



10.7 I²C Write Mode

In this mode the master transmitter transmits to the Digital Flow Sensor slave receiver. Bus protocol is shown in Figure 14. Following the START condition and slave address, a logic 0 (R/W = 0) is placed on the bus and indicates to the addressed device that word address “An” will follow and is to be written to the on-chip address pointer. The data word to be written to the memory is strobed in next and the internal address pointer is incremented to the next address location on the reception of an acknowledge clock. The Digital Flow Sensor slave receiver will send an acknowledge clock to the master transmitter after it has received the slave address and again after it has received the word address and each data byte.

Table 16. **Write mode sequence**



11 Revision History

Version	Date	Notes
1.0	1 March 2013	New issue