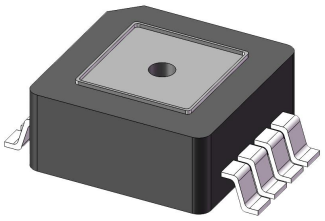
XGZP6878D PRESSURE SENSOR MODULE

FEATURES



- Wide Ranges: 0kPa ~ 100kPa···2500kPa(show in Pressure Range)
- Optional 2.5V ~ 5.5V Power Supply
- Absolute Pressure Type
- For Non-corrosive Gas or Air or Liquid
- Calibrated Digital I2C Signal(Refer to XGZP6878A for Analog signal)
- Temp. Compensated: 0℃ ~ +60℃(32℉ ~ +140℉)
- Current Consumption: 5uA(single measurement)
- Standby Current: <100nA (25°C)

APPLICATIONS

- Tank level measurement, Liquid level measurement, White goods(e.g.washer), Heating, Ventilation, Air conditioning, Fire extinguisher, Filter detection, Industrial controls.

- Medical gas control system, Hospital beds, Massage device, Oil pressure, Engine control/manifold absolute pressure (MAP), Green gas(LPG and CNG) control and measurement.

- Weather station and weather reporting device barometers, Sport and leisure equipment, Vacuum measurement, pneumatic device, Absolute pressure meter.

INTRODUCTION

XGZP6878D is a perfect silicon pressure sensor module offering a ratiometric digital data(I2C interface) for reading pressure and temparature over the specified full scale pressure span.

The XGZP6878D incorporates a silicon piezoresistive pressure sensor die and an interior Application Specific Integrated Circuit(ASIC) in a SMT package.

The XGZP6878D is fully calibrated and temperature compensated for specified span, so XGZP6878D pressure sensor module satisfy the perfect accuracry, which is designed for a wide range of application in medical care&health, home appliances, consumer electronic, industry, automotive, IoT and other pneumatic devices etc by utilizing a microcontroller or microprocessor with D/A inputs.

XGZP6878D pressure sensor module is for high volume application at an affordable cost but perfect performance.

Customized calibrations(pressure range, working voltage, output etc.) are available.

PERFORMANCE PARAMETER

Unless otherwise specified, measurements were taken with a a temperature of 25±1℃ and humidity ranging from 25％ ~ 85％%RH.

Item	Data	Unit
Available Pressure Range[1]	0 ~ 100…2500	kPa
Power Supply[2]	5.0V	Vdc
Max. Excitation Current	3	mA
Output Resolution[3]	24	Bit
SDA/SCL pull up resistor	4.7	Kohm
ESD HBM	4000	V
Accuracy[4]	±2.0	%Span
Long Term Stability[5]	±0.5	%Span
Over Pressure[6]	2X	Rated
Burst Pressure[7]	3X	Rated
Compensation Temp.[8]	0 ~ 60/32 ~ 140	℃/℉
Operating Temp.[9]	-40 ~ 100/-40 ~ 212	℃/℉
Storage Temp.	-40 ~ 125/-40 ~ 257	℃/℉
Response Time[10]	2.5	mS

1 Pressure Range(Operating pressure): The available pressure range including various span, not a specific pressure range.

2 Power supply: The default test voltage value: 5V, optional power supply voltage range:2.5 ~ 5.5V; For better accuracy, please specify the working voltage as order guide.

3 Output Resolution:

3.1. Output Resolution is defined as the max.output AD value from min. rated pressure to max. rated pressure, including:

Offset(Zero output) : it is defined as the output AD value at the minimum rated pressure;

Full Scale Output (FSO): it is defined as the AD value at the maximum or full rated pressure;

Full Scale Span (FSS): it is the algebraic difference between the output AD value at FSO and Offset.

3.2. Output value is nominal values without the count of Accuracy deviation.

4 Accuracy: The max. deviation in output from ideal transfer function at any pressure or temperature over the specified ranges, units are in percent of full scale span (%FSS), which mainly consists of: Offset and Span Shift;Linearity(Non-linearity); Repeatability; Pressure Hsteresis ; TcOffset and TcSpan.

4.1. The accuracy in table is the typical output accuracy during specified pressure range. Contact factory for higher accuracy requirement(e.g ±1%Span) if need.

4.2 Non-linearity(Linearity): the deviation of measured output from "Best Straight Line" through three points (Offset pressure, FS pressure and ½ FS pressure)at constant temperature.

4.3 Repeatability: the deviation of measured output when the same pressure is applied continuously, with pressure approaching from the same direction within the specified operating pressure range,under the same operating conditions.

4.4 Pressure Hysteresis: the deviation of measured output at any pressure within the specified range, when this pressure is applied continuously, with pressure approaching from opposite directions within the specified operating pressure range, under the same operating conditions.

4.5 TcOffset (TCO:Temp. Coefficient of Offset): the deviation of measured output with minimum rated pressure applied, over the temperature range of 0° to 60°C, relative to 25°C.

4.6 TcSpan (TCS:Temp. Coefficient of Span): the deviation of measured output over the temperature range of 0° to 60°C, relative to 25°C.

5. Long Term Stability: the sensor's output deviation when subjected to 1000 hours pressure test.

6. Over Pressure (Proof pressure): the maximum pressure which may be applied without causing durable shifts of the electrical parameters of the sensing element and remain the specification once pressure is returned to the operating pressure range.

7. Burst Pressure: the maximum pressure which may be applied without causing damage to the sensing die or leaks; The sensor should not be expected to recover function after exposure to any pressure beyond the burst pressure.

8. Compensated Temperature: the temperature range over which the sensor have an output proportional to pressure within the specified performance limits.

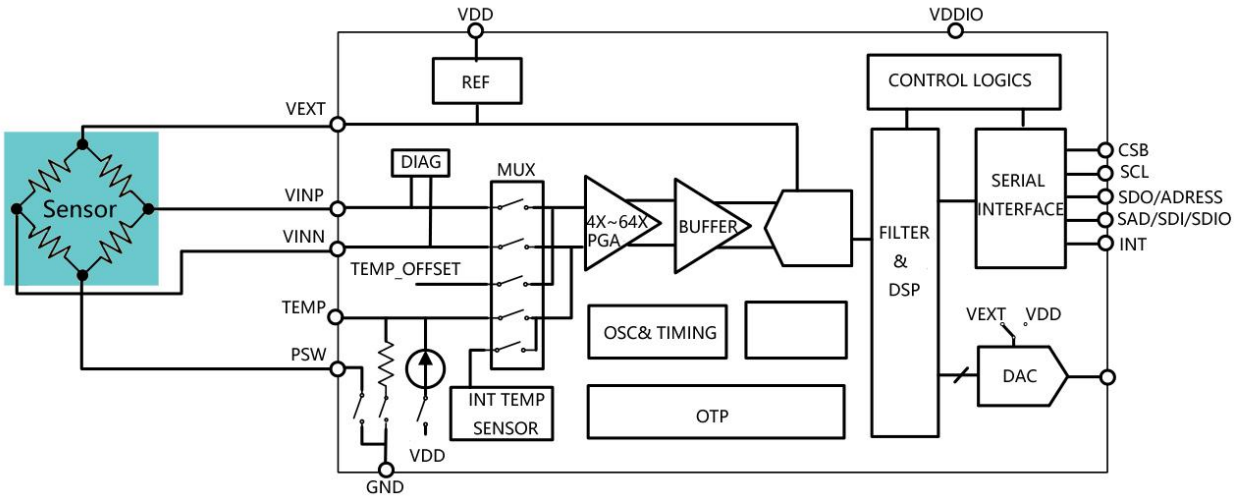
9. Operating Temperature (or Ambient Temperature): the temperature range over which the sensor have an output proportional to pressure but may not remain within the specified performance limits.

10. Response Time: it is defined as the time for the incremental change in the output from 10% to 90% of full-scale pressure base on OSR_P=1024X.

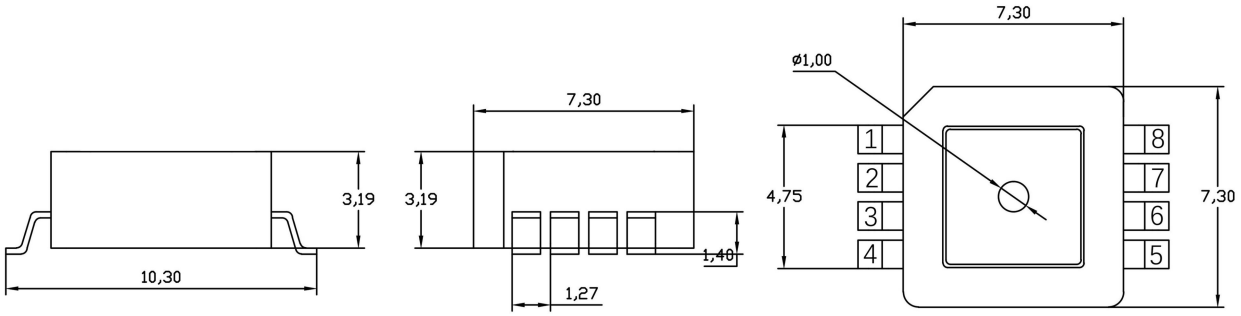
ELECTRICAL CHARACTERISTICS(ASIC)

Parameter	Min.	Typ.	Max.	Unit	Notes
LDO Output	1.62	1.8	1.98	V	'regulator_sel'= 0
	3.24	3.6	3.96	V	'regulator_sel'= 1
PSRR		60		dB	
Input Common Mode Rejection Ratio	80	110			
Interior Temp. Sensor Accuracy			±0.5	℃	@25℃
			±1	℃	-40 to 85 ℃
Temp.Output Data Resolution	8			Bit	LSB = (1/256) ℃
I2C Clock frequency			400	khz	I2C interface

BLOCK DIAGRAM



DIMENSION (Unit:mm)

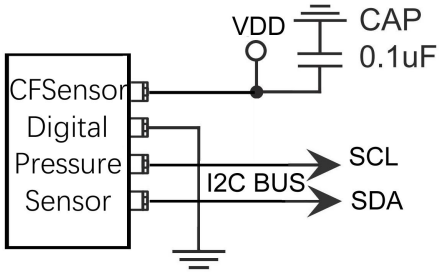


ELECTRIC CONNECTION

1	2	3	4	5	6	7	8
SDA	NC	VDD	NC	GND	NC	SCL	NC

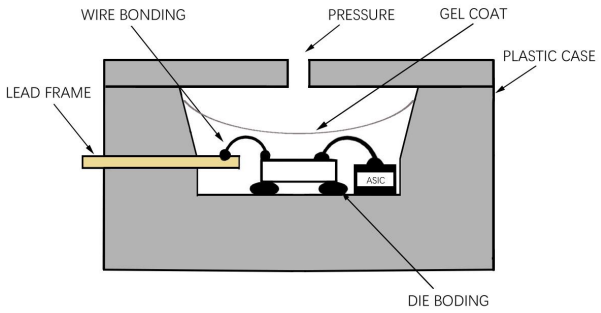
NAME	FUNCTION
NC	Do not connect to external circuitry or ground
VDD	Voltage supply
SDA	Data signal(Send& Receive)
SCL	The clock signal
GND	Ground

CIRCUIT DIAGRAM

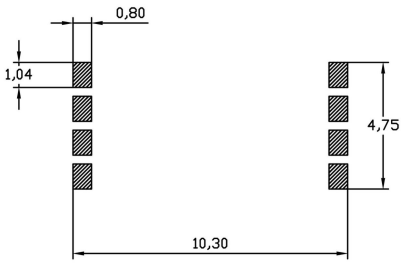


Note: Diagram state schematic connection only;
Check Pin allocation in Dimension drawing.

CRROSS SECTION



FOOTPRINT(REFERENCE)



Unit: mm

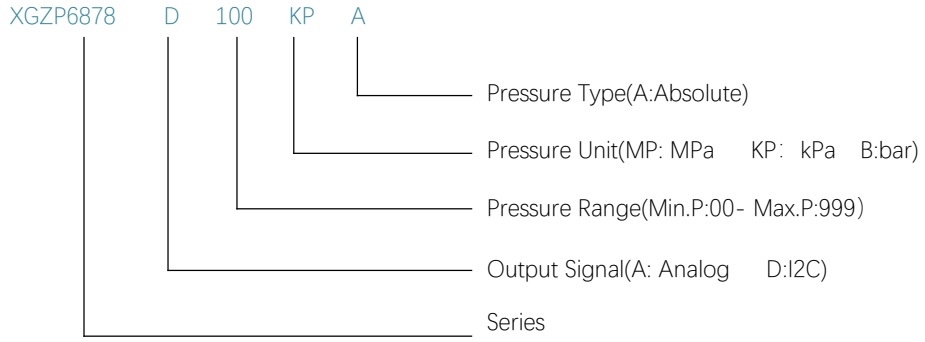
Notes:

1.Implement ESD protection during whole soldering and assembly process.

2.Overload voltage(max.6.5Vdc) or current(max.5mA) may burn the ASIC and cause the sensor fail throughly.

3.More detalis about soldering and storage etc., refer to Overall notes.

ORDER GUIDE



Note: 1. Voltage 5Vdc as default value, add 33(or 30) behind model signify 3.3V(or 3.0V) power supply, e.g. : XGZP6878D200KPA33.
2. Any custom requirement, please comment herewith Part number(e.g custom pressure range etc,.)

ROUTINE PRESSURE RANGE

Notes: 1. Unit conversion: 1000hPa=1000mbar≈750mmHg≈100kPa≈14.5PSI≈10mH_2O≈1bar=0.1MPa;
2. Available for more custom pressure range e.g. 15 ~ 115kPa,75 ~ 325kPa etc,.

Pressure Range (kPa)	Pressure Range (by other units)	Part Number
0 ~ 100kPa	0 ~ 1bar /0 ~ 14.5PSIA	XGZP6878D00100KPA
30 ~ 110kPa	4 ~ 29PSIA	XGZP6878D30110KPA
0 ~ 350kPa	0 ~ 3.5bar /0 ~ 50PSIA	XGZP6878D00350KPA
0 ~ 700kPa	0 ~ 7bar /0 to 100 psi	XGZP6878D00700KPA
0 ~ 1000kPa	0 ~ 10bar /0 to 1MPa	XGZP6878D01000KPA
0 ~ 1500kPa	0 ~ 15bar /0 to 1.5MPa	XGZP6878D01500KPA
Other higher pressure range or custom pressure span, eg 20~250kPa, consult CFSensor		

I2C INTERFACE

I2C bus uses SCL and SDA as signal lines. Both lines are connected to VDDIO externally via pull-up resistors(Typ value:4.7kΩ) so that they are pulled high when the bus is free. The I2C device address of IC is shown below.
I2C device factory setting slave address: **0X6D**

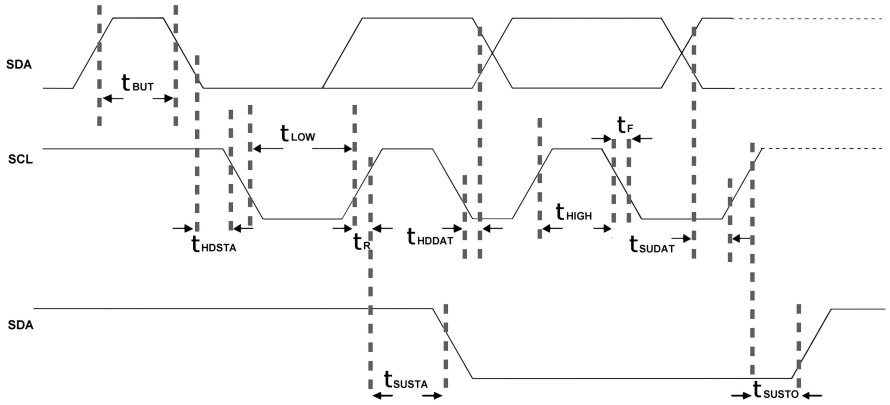
ELECTRICAL SPEC. OF I2C INTERFACE PIN

Symbol	Parameter	Condition	Min	Max	Unit
f_{scl}	Clock frequency			400	KHz
t_{LOW}	SCL low pulse		1.3		us
t_{HIGH}	SCL high pulse		0.6		us
t_{SUDAT}	SDA setup time		0.1		us
t_{HDDAT}	SDA hold time		0.0		us
t_{SUSTA}	Setup Time for a repeated start		0.6		us
t_{HDSTA}	Hold time for a start condition		0.6		us
t_{SUSTO}	Setup Time for a stop condition		0.6		us
t_{BUF}	Time before a new transmission		1.3		us

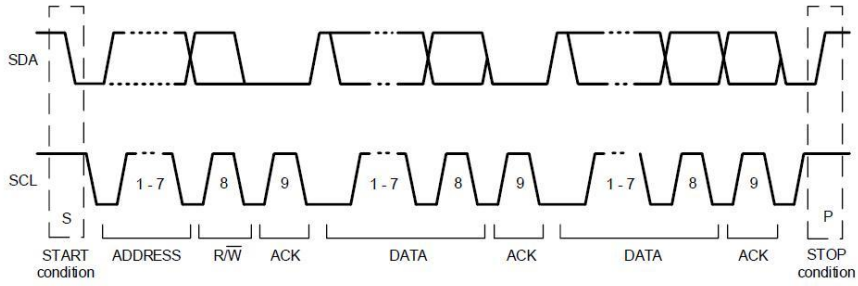
I2C TIME DIAGRAM

The I2C interface protocol has special bus signal conditions. Start (S), stop (P) and binary data conditions are shown below. At start condition, SCL is high and SDA has a falling edge. Then the slave address is sent. After the 7 address bits, the direction control bit R/W selects the read or write operation. When a slave device recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle.

At stop condition, SCL is also high, but SDA has a rising edge. Data must be held stable at SDA when SCL is high. Data can change value at SDA only when SCL is low.



I2C PROTOCAL



GENERAL REGISTER DESC.

Add.	Desc.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default
0x00	SPI _Ctrl	RW	SDO_active	LSB_first	Softreset			Softreset	LSB_first	SDO_active	0x00
0x01	Part_ID	R	PartID<7:0>								0x00
0x02	Status	R	Error_code<3:0>						1'b0	DRDY	
0x06	DATA_MSB	R	Data out<23:16>								0x00
0x07	DATA_CSB	R	Data out<15:8>								0x00
0x08	DATA_LSB	R	Data out<7:0>								0x00
0x09	TEMP_MSB	R	Temp out<15:8>								0x00
0x0A	TEMP_LSB	R	Temp out<7:0>								0x00
0x30	CMD	RW	Sleep_time<3:0>				Sco	Measurement_ctrl<2:0>			0x00
0x6C	OTP_CMD	RW	Blow Start<6:0>							margin	0x00

Reg0x00(for SPI interface only)
SDO_active: 1: 4-wire SPI, 0: 3-wire SPI
LSB_first: 1: LSB first for SPI interface, 0: MSB first for SPI interface
Soft_reset: 1: Reset all the registers (except 'margin'), automatically come back to 0 after reset complete.

Reg0x01
Part ID: OTP programmed 8 bits Part ID, corresponding to OTP register Reg0xA4. Read only from the address 0x01.

Reg0x02
DRDY: 1, indicates once conversion complete, and the output data is ready for reading.
Error_code: When diagnostic function enabled, These bits stores the error information.
Error_code[3]: VINP short to VDD
Error_code[2]: VINP short to GND
Error_code[1]: VINN short to VDD
Error_code[0]: VINN short to GND

Reg0x06-Reg0x08
Data_out: 24 bits ADC output data

Reg0x09-Reg0x0a
Temp_out: Temperature output with an LSB equals to (1/256) °C

Reg0x30
Sleep_time<3:0>: 0000:0ms, 0001:62.5ms, 0010:125ms ... 1111: 1s, only active during sleep mode conversion.
Measurement_control<1:0>: 000b, indicate a single shot temperature signal conversion. 001b, indicate a single shot sensor signal conversion. 010b: indicate a combined conversion (once temperature conversion immediately followed by once sensor signal conversion). 011b: indicate a sleep mode conversion (periodically perform once combined conversion with an interval time of 'sleep_time'), 100b: OTP programming mode, enter this mode to when programming OTP banks.
Sco: 1, Start of conversion, automatically come back to 0 after conversion ends (except sleep mode conversion).

Reg0x6C(Factory setting, no necessory to access)

OTP REGISTER DESC.

Addr	Desc.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default
0xA4	PartID	RW	PartID<7:0>								OTP
0xA5	Sys_config	RW	DAC_on	P_T_ration <1:0>	Vout_sel		Regulator_sel	Unipolar	Raw_dat a_on	DIAG_o n	OTP
0xA6	P_config	RW	1'b0	Input_sw ap	Gain_P<2:0>			OSR_P<2:0>			OTP
0xA7	T_config_1	RW	Temp_sel<1:0>		Gain_T<2:0>			OSR_T<2:0>			OTP
0xA8	T_config_2	RW	4'b0000				T_offset_trim<3:0>				OTP
0xA9	DAC_limit	RW	DAC_limit_h<3:0>				DAC_limit_l<3:0>				OTP
0xAA	Cal_OTP_1	RW	Cal_coff_1<7:0>								OTP
...	...	RW	...								OTP
0xBB	Cal_OTP_18	RW	Cal_coff_19<7:0>								OTP
0xBC	Redundancy	RW	Redundancy<7:0>								OTP

Reg0xA4

PartID: OTP programmed 8 bits Part ID, also can be read from address 0x01.

Reg0xA5

DAC_on: 1, enable analog output. When analog output enabled, ASIC continuously performs once temperature conversion after 64/32/16/1 (configured by 'P_T_ratio') times sensor signal conversions, no matter what 'CMD' (reg0x30) register settings.

P_T_ratio: set how many sensor signal conversions performed after once temperature conversion during analog output mode. 00: 64 times, 01: 32 times, 10: 16 times, 11: once.

Vout_sel: 0: set the DAC output voltage to be rail to rail, that is goes with the voltage on VDD pin, 1: set the DAC output fixed at a voltage range of 0-1.5*VEXT.

Regulator_sel: 0: set the VEXT voltage to be 1.8V, 1: set the VEXT voltage to be 3.6V.

Unipolar: 0: ADC output in bipolar format, 1: ADC output in unipolar format. (Only take effect when 'raw_data_on' = 1)

Raw_data_on: 0: output calibrated data, 1: output ADC raw data. (Only take effect in single shot sensor signal conversion and single shot temperature conversion)

Diag_on: 1, Enable diagnosis function.

Reg0xA6

Input Swap: Swap VINP and VINN inside the ASIC

Gain_P: set the gain of the sensor signal conversion channel. 000: gain=1, 001: gain=2, 010: gain=4, 011: gain=8, 100: gain=16, 101: gain=32, 110: gain=64, 111: gain=128.

OSR_P: set the over sampling ratio of the sensor signal conversion channel. 000:1024X, 001:2048X, 010:4096X, 011:8192X, 100:256X, 101:512X, 110:16384X, 111:32768X.

Reg0xA7

Temp_sel: select different temperature sensing methods. 00: external temp sensor with a resistance connected between TEMP and GND inside chip, 01: external temperature sensor with a current source output via TEMP pin, 10: external temperature sensor, 11: internal temperature sensor.

Gain_T: set the gain of the temperature conversion channel. 000: gain=1, 001: gain=2, 010: gain=4, 011: gain=8, 100: gain=16, 101: gain=32, 110: gain=64, 111: gain=128.

OSR_T: set the over sampling ratio of the temperature conversion channel. 000:1024X, 001:2048X, 010:4096X, 011:8192X, 100:256X, 101:512X, 110:16384X, 111:32768X.

Reg0xA8

T_offset_trim: set the offset voltage for external temperature conversion from 0V to VEXT.

Reg0xA9

DAC_limit_h: set an upper clipping limit for the analog output from 3/4VfsB (0000b) to Vfs(1111b)

DAC_limit_l: set a lower clipping limit for the analog output from 0(0000b) to 1/4Vfs(1111b)

Reg0xAA-Reg0xBB

Cal_coff: Coefficients used for sensor calibrating.

READ OPERATION

As the following instruction sequences for reading data:

1. Read the 0xA5 register value, put the read binary value "and" on "11111101(ie 0xFD)" then write to 0xA5.
2. Send instructions 0x0A to 0x30 register for one temperature acquisition, one pressure data acquisition.
3. Read the 0x30 register address. If Sco bit is 0,signify the acquisition end, the data can be read.
4. Read 0x06, 0x07, 0x08 register address data to form a 24-bit AD value (pressure data AD value).

Read Pressure

The total pressure output value which include 0x06, 0x07 and 0x08 registers are 24 bits. The highest position is the sign bit, The symbol digit value is "0" when it represents "positive".

Pressure_ADC value: = (Pressure 3rd Byte [23:16] x 65536+Pressure 2nd Byte [15:8] x 256 + Pressure1st Byte [7:0])

Note: 1 Pressure 3rd Byte [23:16] is the hexadecimal value read out by REG0x06 and need converte into decimal value;

2 Pressure 2nd Byte [15:8] is the hexadecimal value read out by REG0x07 and need converte into decimal value;

3 Pressure 1st Byte [7:0] is the hexadecimal value read out by REG0x08 and need converte into decimal value.

For Pressure conversion formula are as follows:

The highest bit is "0", which means positive pressure

Pressure = Pressure_ ADC /k;

Note:

1 the unit is Pa (default). If need to display other units, fill the corresponding coefficient in the conversion formula for conversion;

2 the pressure range selection is judged by the max sensor pressure range. e.g pressure sensor 0 ~ 200kPa, the K value is 32.

Pressure range(kpa)	K(value)
2000<P≤4000	2
1000<P≤2000	4
500<P≤1000	8
260<P≤500	16
131<P≤260	32
65<P≤131	64

Read Temperature

The number of temperature output values in the 0x09 and 0x0A registers are 16 bits, the highest is the symbol bit.

The symbol digit value is "1" when it represents "negative", and the symbol digit value is "0" when it represents "positive".

Supposing if the decimal values of REG0x09 and REG0x0A readout are X, Y,

For Temperature ADC value and conversion formula are: Temperature value: N=X*256+Y

If n<2^15, Temperature is positive value, temperature T =N/256; (℃).

If n>2^15, Temperature is negative value, Temperature value = (N-2^16) /256; (℃)

OVERALL NOTES

Mounting

The following steps is for transmitting the air pressure to sensor after sensor soldering on PCB.

▼ Inlet pipe can't be blocked with gel or glue etc,..

▼ Avoiding excessive external force operation

Soldering

Due to its small size, the thermal capacity of the pressure sensor is low. Therefore, take steps to minimize the effects of external heat.Damage and changes to characteristics may occur due to heat deformation. Use a non-corrosive resin type of flux. Since the pressure sensor is exposed to the atmosphere, do not allow flux to enter inside.

▼ Manual soldering

⊙Raise the temperature of the soldering tip between 260 and 300°C/500 and 572°F (30 W) and solder within 5 seconds.

⊙The sensor output may vary if the load is applied on the terminal during soldering.

⊙Keep the soldering tip clean.

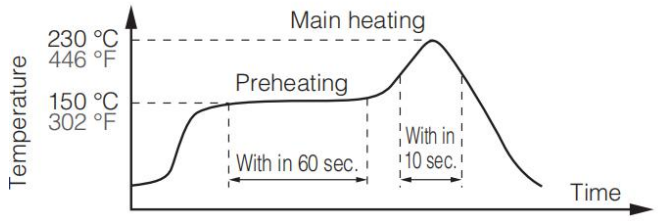
▼DIP soldering (DIP Terminal)

⊙Keep the temperature of the DIP solder tank below 260°C/500 and solder within 5 seconds.

⊙To avoid heat deformation, do not perform DIP soldering when mounting on the PCB which has a small thermal capacity.

▼Reflow soldering (SMD Terminal)

⊙The recommended reflow temperature profile conditions are given below.



⊙We recommend the screen solder printing method as the method of cream.

⊙Please refer to the recommended PC board specification diagram for the PC board foot pattern.

⊙Self alignment may not always work as expected, therefore, please carefully the position of the terminals and pattern.

⊙ The temperature of the profile is assumed to be a value measured with the printed wiring board of the terminal neighborhood.

⊙Please evaluate solderbility under the actual mounting conditions since welding and deformation of the pressure inlet port may occur due to heat stress depending on equipments or conditions.

▼ Rework soldering

⊙Complete rework at a time.

⊙Use a flattened soldering tip when performing rework on the solder bridge. Do not add the flux.

⊙Keep the soldering tip below the temperature described in the specifications.

▼ Avoid drop and rough handling as excessive force may deform the terminal and damage soldering characteristics.

▼ Keep the circuit board warpage within 0.05 mm of the full width of the sensor.

▼ After soldering, do not apply stress on the soldered part when cutting or bending the circuit board.

▼ Prevent human hands or metal pieces from contacting with the sensor terminal. Such contact may cause anomalous outlets as the terminal is exposed to the atmosphere.

▼ After soldering, prevent chemical agents from adhering to the sensor when applying coating to avoid insulation deterioration of the circuit board.

Connecting

▼ Correctly wire as in the connection diagram. Reverse connection may damage the product and degrade the performance.

▼ Do not use idle terminals(N/C) to prevent damages to the sensor.

Cleaning

▼ Since the pressure sensor is exposed to the atmosphere, do not allow cleaning fluid to enter inside.

▼ Avoid ultrasonic cleaning since this may cause breaks or disconnections in the wiring.

Environment

▼Please avoid using or storing the pressure sensor in a place exposed to corrosive gases (such as the gases given off by organic solvents, sulfurous acid gas, hydrogen sulfides, etc.) which will adversely affect the performance of the pressure sensor chip.

▼Since this pressure sensor does not have a water-proof construction, please do not use the sensor in a location where it may be sprayed with water, etc.

▼Avoid using the pressure sensors in an environment where condensation may form. Furthermore, its output may fluctuate if any moisture adhering to it freezes.

▼The pressure sensor is constructed in such a way that its output will fluctuate when it is exposed to light. Especially when pressure is to be applied by means of a transparent tube, take steps to prevent the pressure sensor chip from being exposed to light.

▼Avoid using pressure sensor where it will be susceptible to ultrasonic or other high-frequency vibration.

▼Please keep the sensors sealed using static shielding bags on storage. The PINs of sensor are plated by Ag. If the sensors expose to an atmosphere, the PINs will be black by oxidation， althought it wouldn't affect the sensor performance.

More Precautions

▼That using the wrong pressure range or mounting method may result in accidents.

▼The only direct pressure medium you can use is air(25％ ~ 85％%RH). The use of other media, in particular, corrosive gases (organic solvent based gases, sulfurous acid based gases, and hydrogen sulfide based gases, etc.) and media that contains heavy moisture or foreign substances will cause malfunction and damage. Please do not use them and check with CFSensor.

▼The pressure sensor is positioned inside the pressure inlet. Never poke wires or other foreign matter through the pressure inlet since they may damage the sensor or block the inlet. Avoid use when the atmospheric pressure inlet is blocked.

▼Use an operating pressure which is within the rated pressure range. Using a pressure beyond this range may cause damage.

▼Since static charge can damage the pressure sensor, bear in mind the following handling precautions.

⊙When storing the pressure sensor, use a conductive material to short the pins or wrap the entire sensor in aluminum foil. Plastic containers should not be used to store or transport the sensor since they readily become charged.

⊙ When using the pressure sensor, all the charged articles on the bench surface and the work personnel should be grounded so that any ambient static will be safely discharged.

▼Based on the pressure involved, give due consideration to the securing of the pressure sensor DIP type and to the securing and selection of the inlet tube.

【 SAFETY NOTES 】

Using these sensors products may malfunction due to external interference and surges, therefore, please confirm the performance and quality in actual use. Just in case, please make a safety design on the device (fuse, circuit breaker, such as the installation of protection circuits, multiple devices, etc.), so it would not harm life, body, property, etc even a malfunction occurs.

To prevent injuries and accidents, please be sure to observe the following items:

- The driving current and voltage should be used below the rated value.
- Please follow the terminal connection diagram for wiring. Especially for the reverse connection of the power supply, it will cause an accident due to circuit damage such as heat, smoke, fire, etc.
- In order to ensure safety, especially for important uses, please be sure to consider double safety circuit configuration.
- Do not apply pressure above the maximum applied pressure. In addition, please be careful not to mix foreign matter into the pressure medium. Otherwise, the sensor will be discarded, or the media will blew out and cause an accident.
- Be careful when fixing the product and connecting the pressure inlet. Otherwise, accidents may occur due to sensor scattering and the blowing out of the media.
- Because the sensor PIN is sharp, please be careful not to hurt your body when using it.

【 WARRANTY 】

The information in this sheet has been carefully reviewed and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Furthermore, this information does not convey to the purchaser of such devices any license under the patent rights to the manufacturer. CFSensor reserves the right to make changes without further notice to any product herein. CFSensor makes no warranty, representation or guarantee regarding the suitability of its product for any particular purpose, nor does CFSensor assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Typical parameters can and do vary in different applications. All operating parameters must be validated for each customer application by customer's technical experts. CFSensor does not convey any license under its patent rights nor the rights of others.

【 CONTACT 】

CFSensor

14-15F/4Bldg High-Tech Park High-Tech Area Wuhu P.R.C.241000

Tel/Fax:+86 18226771331 Email:INFO@CFSensor.com

North America II Europe II Southeast Asia II Middle East II Latin America

READ CODE(REFERNCE)

```
#include <reg52.h>
#include <math.h>
#define DELAY_TIME 60
#define TRUE 1
#define FALSE 0
#define uchar unsigned char
#define uint unsigned int
//--------------define IIC SCL,SDA port-----------------------------
sbit SCL = P1 ^ 7;
sbit SDA = P1 ^ 6;
//-----------------define Max7219 port--------------------------
sbit Max7219_pinCLK = P2 ^ 2;
sbit Max7219_pinCS = P2 ^ 1;
sbit Max7219_pinDIN = P2 ^ 0;
//-------------delay time_us-------------------------------
void DELAY(uint t)
{
      while (t != 0)
            t--;
}
//-------------IIC START CONDITION-------------------------------
void I2C_Start(void)
{
      SDA = 1;                    //SDA output high
      SCL = 1;
      DELAY(DELAY_TIME);       //SCL output high
      SDA = 0;
      DELAY(DELAY_TIME);
      SCL = 0;
      DELAY(DELAY_TIME);
}
//-------------IIC STOP CONDITION-------------------------------
void I2C_Stop(void)
{
      SDA = 0;                    //SDA OUTPUT LOW
      SCL = 1;
      DELAY(DELAY_TIME);
      SDA = 1;
      DELAY(DELAY_TIME);
      SCL = 0;                    //SCL OUTPUT LOW
```

```
    DELAY(DELAY_TIME);
}
//-------------IIC SEND DATA "0"------------------------------
void SEND_0(void)
{
    SDA = 0;
    SCL = 1;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}
//-------------IIC SEND DATA "1"------------------------------
void SEND_1(void)
{
    SDA = 1;
    SCL = 1;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}
//-------------Check SLAVE's Acknowledge ----------------------------
bit Check_Acknowledge(void)
{
    SDA = 1;
    SCL = 1;
    DELAY(DELAY_TIME / 2);
    F0 = SDA;
    DELAY(DELAY_TIME / 2);
    SCL = 0;
    DELAY(DELAY_TIME);
    if (F0 == 1)
        return FALSE;
    return TRUE;
}
//-------------Write One Byte of Data -------------------------------
void Writel2CByte(uchar b) reentrant
{
    char i;
    for (i = 0; i < 8; i++)
        if ((b << i) & 0x80)
            SEND_1();
        else
```

```
                SEND_0();
}
//-------------Read One Byte of Data ------------------------------
uchar ReadI2CByte(void) reentrant
{
    char b = 0, i;
    for (i = 0; i < 8; i++)
    {
        SDA = 1;
        SCL = 1;
        DELAY(10);
        F0 = SDA;
        DELAY(10);
        SCL = 0;
        if (F0 == 1)
        {
            b = b << 1;
            b = b | 0x01;
        }
        else
            b = b << 1;
    }
    return b;
}
//-------------write One Byte of Data,Data from MASTER to the SLAVER ------------------------------
//-------------SLAVER address bit:01101101------------------------------
void Write_One_Byte(uchar addr, uchar thedata)    //Write "thedata" to the SLAVER's address of "addr"
{
    bit acktemp = 1;
    I2C_Start();                                  //IIC START
      WriteI2CByte(0xDA);                         //IIC WRITE operation,SLAVER address bit:01101010
    acktemp = Check_Acknowledge();                //check the SLAVER
    WriteI2CByte(addr); /*address*/
    acktemp = Check_Acknowledge();
    WriteI2CByte(thedata); /*thedata*/
    acktemp = Check_Acknowledge();
    I2C_Stop();                                   //IIC STOP
}
//-------------Reaed One Byte of Data,Data from SLAVER to the MASTER ------------------------------
uchar Read_One_Byte(uchar addr)
{
    bit acktemp = 1;
```

```
    uchar mydata;

    I2C_Start();
    WriteI2CByte(0xDA);
    acktemp = Check_Acknowledge();
    WriteI2CByte(addr);
    acktemp = Check_Acknowledge();
    I2C_Start();
    WriteI2CByte(0xDB);                    //IIC READ operation
    acktemp = Check_Acknowledge();
    mydata = ReadI2CByte();
    acktemp = Check_Acknowledge();
    I2C_Stop();
    return mydata;
}
//-------------Delay_ms -------------------------------
void Delay_xms(uint x)
{
    uint i, j;
    for (i = 0; i < x; i++)
        for (j = 0; j < 112; j++)
            ;
}
//-------------Write One Byte to the Max7219-------------
void Write_Max7219_byte(uchar DATA)
{
    uchar i;
    Max7219_pinCS = 0;          //CS low effect
    for (i = 8; i >= 1; i--)
    {
        Max7219_pinCLK = 0;
        Max7219_pinDIN = DATA & 0x80;
        DATA = DATA << 1;
        Max7219_pinCLK = 1;                //when pinCLK is high send the Data
    }
}
//-------------decide which address shows the Data-------------
void Write_Max7219(uchar address,uchar dat)
{
    Max7219_pinCS = 0;
    Write_Max7219_byte(address);
      Write_Max7219_byte(dat);
```

```
    Max7219_pinCS = 1;
}
//-------------MAX_7219 Initialization-------------
void Init_MAX7219(void)
{
    Write_Max7219(0x09, 0xff);     //
    Write_Max7219(0x0a, 0x03);     //
    Write_Max7219(0x0b, 0x07);     //
    Write_Max7219(0x0c, 0x01);     //
    Write_Max7219(0x0f, 0x01);     //
}
void main(void)
{
    uchar yali1, yali2, yali3,wendu1,wendu2;
    uchar temp_a5;
    long int ad,temp;
    long float pas;
    uchar dis[8];
    Init_MAX7219();
    Delay_xms(1000);
    Write_Max7219(0x0f, 0x00);
    while (1)
    {
        temp_a5 = Read_One_Byte(0xA5);//Read ASIC Sys_config
        temp_a5 = temp_a5 & 0xFD;
        Write_One_Byte(0xA5, temp_a5); //Set ADC output calibrated Data
        Write_One_Byte(0x30, 0x0A); //indicate a combined conversion (once temperature conversion immediately
followed by once sensor signal conversion)
        while ((Read_One_Byte(0x30) & 0x08) > 0);    //Judge whether Data collection is over
// -------------READ ADC output Data of Pressure    -------------
                yali1 = Read_One_Byte(0x06);
        yali2 = Read_One_Byte(0x07);
        yali3 = Read_One_Byte(0x08);

        ad = yali1 * 65536 + yali2 * 256 + yali3;
// -------------READ ADC output Data of Temperature    -------------
                wendu1= Read_One_Byte(0x09);
                wendu2= Read_One_Byte(0x0a);
                temp=wendu1*256+wendu2;
/*Conversion, the following is the conversion formula of 40kpa*/
                if (ad > 8388608)
        {
```

```
            pas = (ad - 16777216) * 0.0078125;
        }
        else
        {
            pas = ad * 0.0078125;
        }
        if (pas < 0)
            pas = fabs(pas);
        dis[0] = (long int)pas / 10000000;
        dis[1] = (long int)pas % 10000000 / 1000000;
        dis[2] = (long int)pas % 1000000 / 100000;
        dis[3] = (long int)pas % 100000 / 10000;
        dis[4] = (long int)pas % 10000 / 1000;
        dis[5] = (long int)pas % 1000 / 100;
        dis[6] = (long int)pas % 100 / 10;
        dis[7] = (long int)pas % 10;
        Write_Max7219(8, dis[0]);
        Write_Max7219(7, dis[1]);
        Write_Max7219(6, dis[2]);
        Write_Max7219(5, dis[3]);
        Write_Max7219(4, dis[4]);
        Write_Max7219(3, dis[5]);
        Write_Max7219(2, dis[6]);
        Write_Max7219(1, dis[7]);
        Delay_xms(100);                    //delay 100ms
    }
}
```