GZP6848D

Pressure Sensor

Digital Output（IIC)

Datasheet

Version：V1.0

Issued Data：2025.08.28

Table of Contents

Document Revision History

Revision	Description	Date
V1.0	Initial release	2025.08.28

1 Product Description

The GZP6848D pressure sensor is a custom MEMS pressure sensor designed particularly for a wide applications with various pressure range. It is composed of a silicon piezoresistive pressure sensing chip and a signal conditioning integrated circuit. The initial signal from the sensing chip is amplified, temperature compensated, calibrated and finally converted to a digital signal(I2C) that is corresponding to the applied pressure.

1.1 Features

- Available range 0 to 1400kPa
- Absolute pressure type
- DIP package
- Power supply voltage: 2.5V ~ 5.5V
- IIC Interface
- Optional gel filling for waterproof application

1.2 Applications

- Laser cutting machine
- Electric welding machine
- Industrial equipment
- Pneumatic control system
- Vacuum system

2 Function Description

This product is manufactured using advanced micro-electromechanical principles. Its core technologies are a MEMS pressure sensor chip based on the silicon piezoresistive effect and a high-performance signal conditioning AISC chip. The silicon micro-piezoresistive MEMS pressure sensor chip forms a Wheatstone bridge through four strain-sensitive resistors. The output signal is amplified, temperature-compensated, and linearized by the ASIC chip. The linearization and temperature compensation of the transfer function are implemented by the digital processing circuit in the ASIC. Through the polynomial compensation algorithm and multi-point pressure calibration technology at multiple temperatures, high-precision pressure measurement is achieved over the entire operating temperature range.

2.1 Definition

The pin configuration of the pressure sensor is shown in Figure 2.

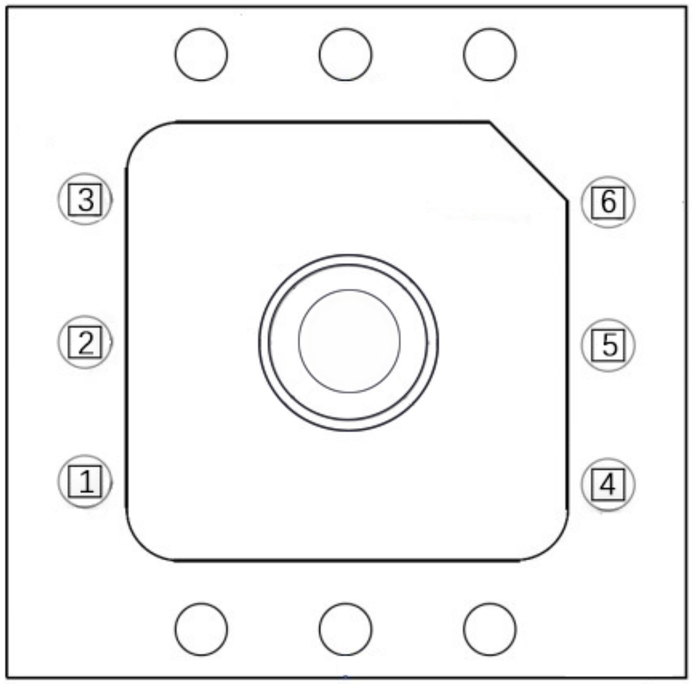


Fig.2 Pin configuration diagram(Top View)

The corresponding relationship of the pressure sensor pins is shown in Table 1.

Tab.1 Pin Definition

PIN No.	Description	Remark
1	NC	Floating pin
2	SCL	Signal output pin
3	SDA	Data output pin
4	VDD	Power input positive
5	NC	Floating pin
6	GND	Power input negative

2.2 Block Diagram

The functional block diagram of the pressure sensor is shown in Figure 1.

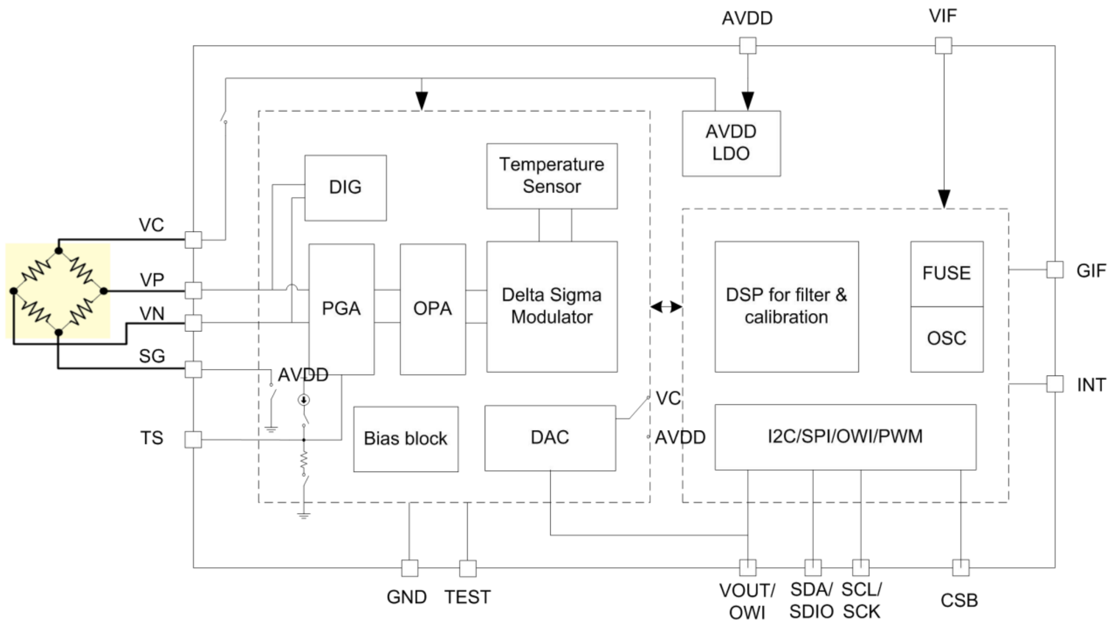


Fig.1 Block Diagram

2.3 Accuracy

GZP6848D pressure sensor is affected by supply voltage, input pressure, ambient temperature, and aging. The value calculated using the transfer function is the sensor's specified value, also known as the theoretical value. The sensor's error is the difference between the actual output value and the specified output value at a specified input pressure.

Overall Accuracy

The overall error includes more accuracy sources based on the product accuracy :

Pressure drift: The output deviation between the actual output voltage at zero point and full scale and the specified output voltage within the specified pressure range.

Temperature effect: The output deviation of zero point and full scale at different temperatures within the temperature range.

The overall accuracy is expressed by error band, and the data are shown in Figure 3 and Table 2 shown.

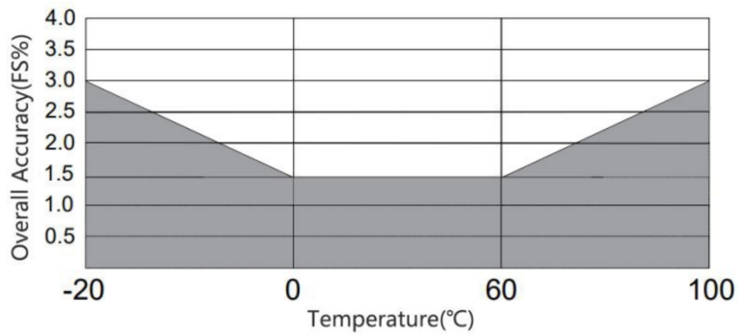


Fig.3 Overall Accuracy vs. temperature.

Tab.2 Overall Accuracy

Temperature (℃)	**Overall Accuracy(Full Span)**
-20~100	±3.0%
0~60	±1.5%

* Different pressure ranges have different Overall Accuracy. Please consult customer service for more details.

3 Technical Indicators

The following indicators of the sensor are measured with (5±0.25)V DC and 25℃.

3.1 Maximum Ratings

The maximum rated parameters of the sensor are shown in Table 3.

Tab.3 The maximum rated parameters

Parameter	**Min.**	**Typical Value**	**Max.**	**Unit**	**Remark**
Supply Voltage	-0.3		6.5	V	
ESD Protection		±2		kV	HBM
Overload Pressure	2X (Range≤350kPa)			Rate	
	1.5X (Range≥350kPa)				
Bursting Pressure	3X (Range≤350kPa)			Rate	
	2X (Range≥350kPa)				
Working Temperature	-30		105	℃	
Storage Temperature	-40		125	℃	

1. Different pressure range may have different overload pressure and burst pressure, please consult Sencoch for more details.

2. Long exposure at the specified limits may cause degradation to the device.

3.2 Performance Indicators

The sensor performance indicators are shown in Table 4.

Tab.4 Sensor performance indicators

* The different pressure range may have different accuracy, please consult Sencoch for more details.

Parameter	Value	Unit	Remark
Pressure Range	0～1400	kPa	Customizable
Power Supply	2.7~5.5	V	
Standby Current	100	nA	
Accuracy	±1	%Span	
Pressure Resolution	24	Bits	
Built-in Temperature Accuracy	±2	℃	@0~60℃
Temperature Resolution	16	Bits	LSB = (1/256) °C
Compensation Temperature	0～60	℃	Customizable
Pull-up Resistors	4.7	K ohm	
Clock Frequency	400	KHz	Max.
Response Time	2.5ms	ms	OSR_P=512X

4 Application Circuit

The recommended application circuit of the sensor is shown in Figure 4.

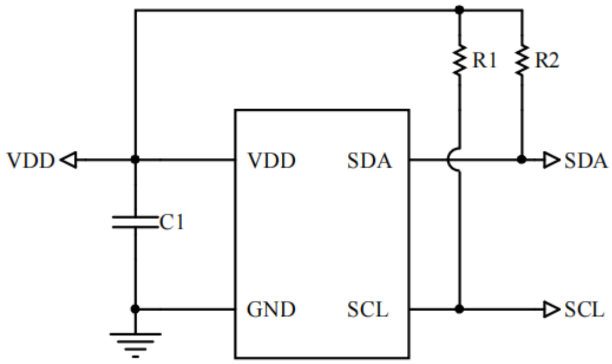


Fig.4 Recommended sensor application circuit diagram

Notice :

- The recommended value of C1 is 100nF, and the recommended values of R1 and R2 are 4.7k
- Please confirm the electrical definition before assembly
- Do not have any electrical connection to the NC pin, otherwise it may cause product failure.
- Provide anti-static protection during welding
- Overload voltage (6.5Vdc) may burn out the circuit chip
- This product has no reverse polarity protection, please pay attention to the power polarity during assembly

5 I²C Communication Protocol

The I²C bus uses SCL and SDA as signal lines. Both lines are connected to VDD through pull-up resistors (typical value 4.7K) and remain high when not communicating. The I²C device address is 0x6D.

The I²C communication protocol has specific start (S) and stop (P) conditions. While SCL is high, a falling edge on SDA signals the start of data transmission. The I²C master device sequentially transmits the slave device's address (7 bits) and the read/write control bit. When the slave device recognizes this address, it generates an acknowledge signal and pulls SDA low in the ninth cycle. After receiving the slave device's acknowledgement, the master device continues to transmit the 8-bit register address and, upon receiving the acknowledgement, continues to send or read data. A rising edge on SDA while SCL is high signals the end of I²C communication . In addition to the start and stop signals, data transmitted by SDA must remain stable while SCL is high. The value transmitted by SDA can change while SCL is low. All data transmission in I²C communication is in 8-bit units, and an acknowledge signal is required after every 8 bits of data transmission to ensure continued transmission.

The I²C timing diagrams are shown in Figures 5 and 6.

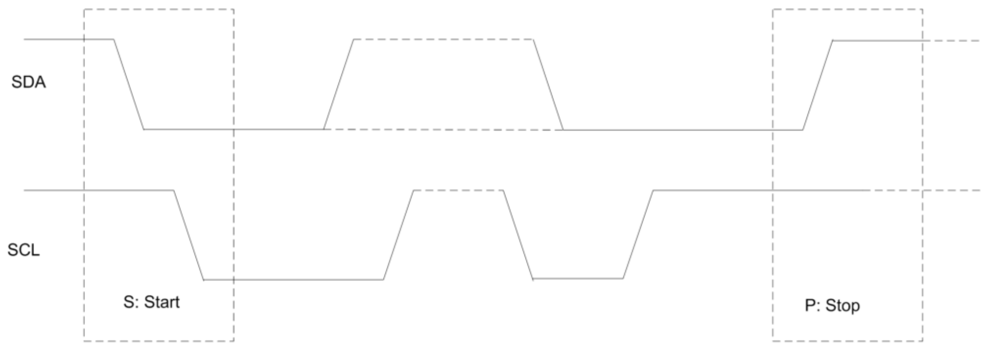


Fig.5. I^{2}C Timing Diagram 1

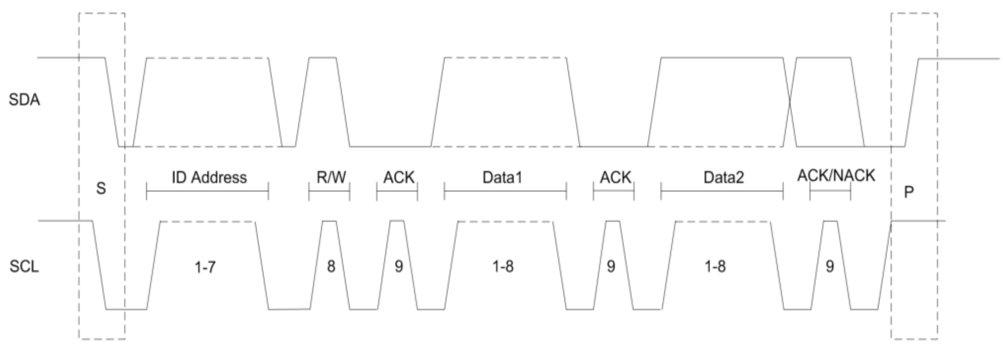


Fig.6 I^{2}C Timing Diagram 2

6 Register Description

The register description is shown in Table 5.

Tab.5 Register description

Add.	Desc.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default
0x06	DATA_MSB	R	Pressure Data out<23:16>								0x00
0x07	DATA_CSB	R	Pressure Data out<15:8>								0x00
0x08	DATA_LSB	R	Pressure Data out<7:0>								0x00
0x09	TEMP_MSB	R	Temp Data out<15:8>								0x00
0x0A	TEMP_LSB	R	Temp Data out<7:0>								0x00
0x30	CMD	RW	Sleep_time<7:4>				Sco	Measurement_ctrl<2:0>			0x00
0xA5	Sys_config	RW	Aout_config<7:4>				LDO_config	Unipolar	Data_out_control	Diag_on	OTP
0xA6	P_config	RW		Input Swap	Gain_P<5:3>			OSR_P<2:0>			OTP

Reg0x06-Reg0x08 :

Pressure ADC data Register

Reg0x09 & Reg0x0A :

Temperature ADC data Register

Reg0x30 : Measurement Command Register

Sleep_time<7:4>: 0000:0ms; 0001:62.5ms; 0010:125ms ;... 1111: 1s, only active during sleep mode conversion.

Sco: 1: Start of conversion, automatically come back to 0 after conversion ends (except sleep mode conversion).

Measurement_control<2:0>:

010: indicate the combined conversion (namely a temperature conversion immediately followed by a pressure-signal conversion).

011: indicate a sleep mode conversion (periodically perform a combined conversion with an interval time of ‘sleep_time’),

Reg0xA5(configured at factory)

Aout_config<7:4>: Analog output setting(recommending reserve default value)

LDO_config: 0: set with 1.8V;
1: set with 3.6V.

Polarity: 0: ADC output in bipolar format(signed binary),
1: ADC output in unipolar format. (Unsigned binary, Only take effect when ‘raw_data_on’ = 1)

Raw_data_on: 0: output calibrated data(as default value),
1: output ADC raw data.

Diag_on: 1, Enable diagnosis function(default).

Reg0xA6(configured at factory)

Input Swap: Swap VINP and VINN inside the ASIC

Gain_<5:3>: 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<2:0>: 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.

7 Read and Calculation

Read Operation

1. Send instructions 0x0A(combined conversion mode command) to 0x30 register to start conversion.
2. Delay 20ms for reading
3. Read 0x06, 0x07, 0x08 register address data to form a 24-bit AD value (pressure data AD value);
4. Read 0x09, 0x0A register address data to form a 16-bit AD value (temperature data AD value)

Pressure Calculation

The total pressure output value which include 0x06, 0x07 and 0x08 registers are 24 bits. The highest position is the signed bit, and the value is "0", it represents "positive" pressure; The symbol digit value is "1", it represents "negative" pressure.

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 convert into decimal value;

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

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

For Pressure conversion formula are as follows:

The highest bit is "0", which means positive pressure, then Pressure = Pressure_ ADC/k;

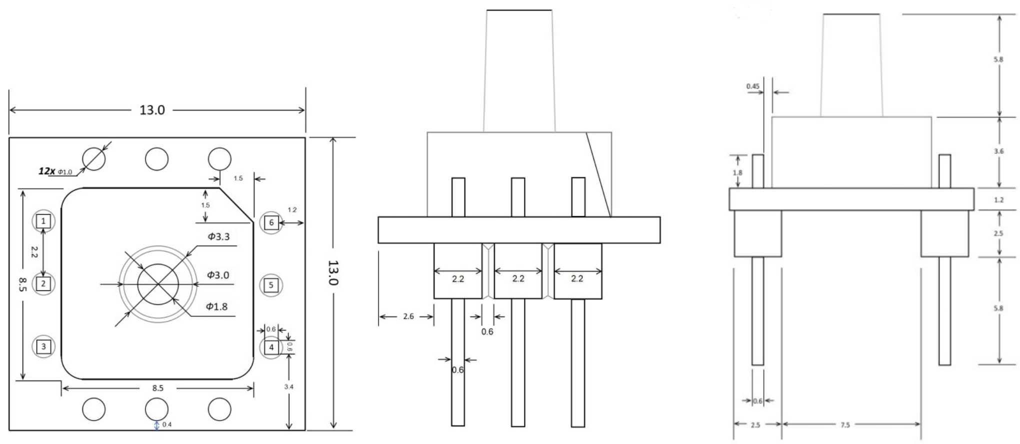
The highest bit is "1", which means negative pressure, then Pressure=(Pressure_ADC-2^24)/k;

Tab.6 Range calibration parameters

Pressure Range(Absolute Value-KPa)	K	Sensor Pressure Range Example
1000≤Pressure Range	4	0~1000kPa; 0~1400kPa;
500<Pressure Range＜1000	8	0~700kPa;
260<Pressure Range≤500	16	0~500kPa; 0~300kPa...
130<Pressure Range≤260	32	0~200kPa;
65<Pressure Range≤130	64	0~100kPa;

8 Structure Specification (unit:mm)

Refer to Figure 7 for the sensor's dimensions (error is ± 0.1mm if not specified).



Top View　　Side View　　Front View

Fig.7 Product dimensions

The recommended pad dimensions are shown in Figure 8.

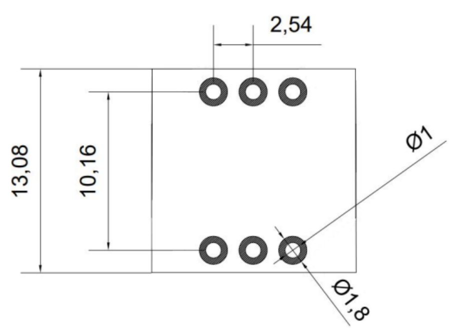


Fig.8 Recommended footprint

9 Order Guide

GZP 6848 D G1 00014BAA F01 WX

Tab.7 Order Guide

GZP	Pressure Sensor Series
6848	Product Series
D	Output type A: Analog output D: IIC output
G1 (Optional)	Protection Type G1: Gel filling
00014BAA	Pressure range: 00014 means the minimum pressure (00) and the maximum pressure (014) Pressure unit: KP: KPa MP: MPa PS: PSI BA: Bar Pressure Type: A：Absolute Pressure P：Gauge Pressure Therefore, 00014BAA represents the measurement pressure of 0bar to 14bar absolute pressure
F01	Packaging Method: B01: Reel&Tape F01: Tube
WX	Company interior code

10 Model Example

Tab.8 Model example

Pressure Range	**Model**	**Remark**
0～1400kPa	GZP6848DG1 00014 BA A F01WX	Gel filling

1. Above model example is for order information only, contact Sencoch for production and stock status.
2. For more customized ranges and special parameter part numbers, please consult Sencoch or agents.

11 Instruction for Use

11.1 Soldering

Since this product has a small structure with low heat capacity, please minimize the influence of heat from the outside. Otherwise, it may be damaged due to thermal deformation and cause changes in characteristics. Please use non-corrosive rosin type flux. In addition, since the product is exposed to the outside, please be careful not to allow flux to penetrate into the inside.

(1) Manual soldering

- Use a soldering iron with a head temperature between 260 and 300°C (30 W) and perform the work within 5 seconds.
- Please note that the output may change when soldering with a load applied to the terminals.
- Please keep the soldering iron tip clean.

(2) Reflow soldering (SMD terminal type)

- The recommended reflow oven temperature setting conditions are shown:

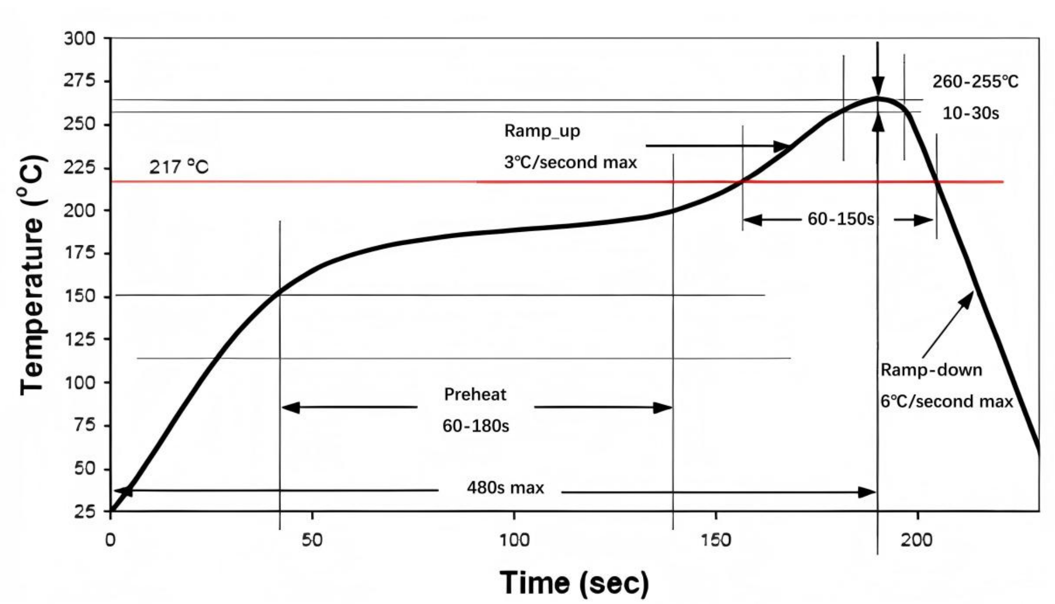


Fig.9 Remelting temperature setting conditions

(3) The warping of the printed circuit board relative to the entire sensor should be kept below 0.05mm. Please manage this.

(4) After installing the sensor, be careful not to generate stress on the solder joint when cutting and bending the substrate.

(5) Since the sensor terminals are exposed, contact with metal pieces or other objects may cause abnormal output. Be careful not to touch the terminals with metal pieces or your hands.

(6) When applying coating to prevent insulation degradation of the substrate after soldering, be careful not to allow chemicals to adhere to the sensor.

11.2 Cleaning Requirements

(1) Since the product is open type, cleaning fluid is not allowed to enter interior.

(2) Please avoid using ultrasonic cleaning as it may cause product failure.

11.3 Storage and Transportation

(1) This product is not drip-proof, so do not use it in places where it may be splashed with water.

(2) Do not use in an environment where condensation occurs. In addition, if moisture attached to the sensor chip freezes, it may cause fluctuations in sensor output or damage.

(3) Due to the structure of the pressure sensor chip, the output will fluctuate when it is exposed to light. Especially when applying pressure through a transparent cover, etc., please avoid light from reaching the sensor chip.

(4) Normally packaged pressure sensors can be transported by ordinary transportation vehicles. Please note: The product must be protected from moisture, shock, sunburn and pressure during transportation.

11.4 Other Precautions

(1) If the installation method is incorrect, it may cause an accident, so please be careful.

(2) Avoid using the product in a manner that applies high-frequency vibrations.

(3) The only pressure medium that can be used directly is dry, non-corrosive gas(Gel fill series is resist-moisture). Other media, especially corrosive media or media containing foreign matter, may cause malfunction and damage. Therefore, please avoid using it in the above environment.

(4) A pressure sensor chip is located inside the pressure inlet. Inserting a needle or other foreign object into the pressure inlet can damage the chip and clog the inlet, so please avoid such an operation.

(5) Regarding the operating pressure, please use it within the rated pressure range. Using it outside the range may cause damage.

(6) Since static electricity may cause damage, please be careful to ground charged objects on the table and operators when using it to allow the surrounding static electricity to discharge safely.

If you have any questions, please feel free to ask.

12 Packaging Information

Tube Packing

Quantity per tube: 38 PCS

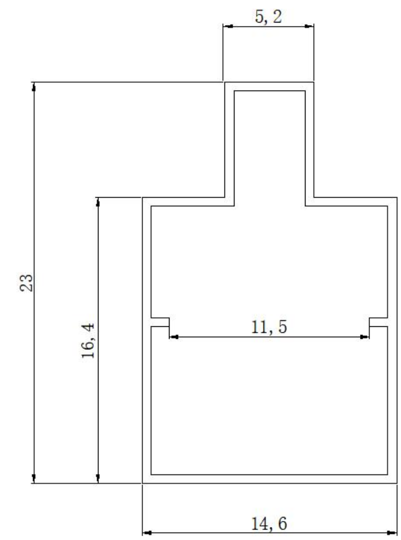
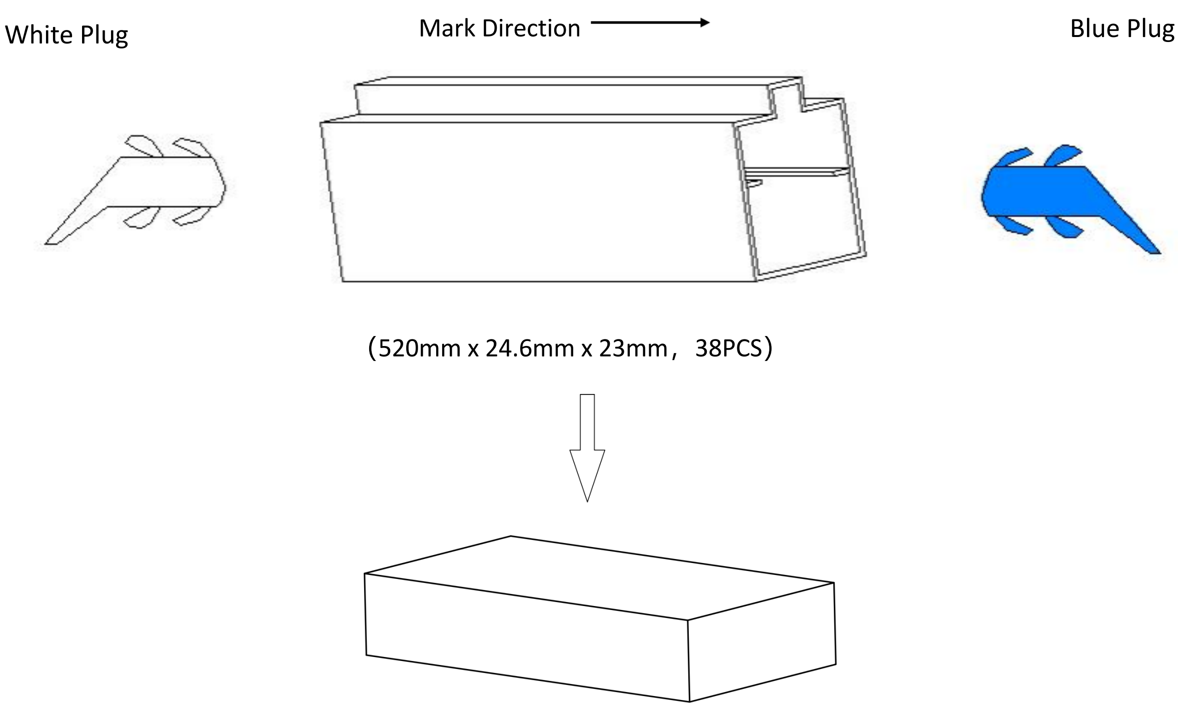


Fig.10 Section schematic diagram



530mm x 145mm x 53mm， 760PCS

Fig.11 Outer Packing

Safety Precautions

This product is made of semiconductor components for general electronic equipment (communication equipment, measuring equipment, working machinery, etc.). Products using these semiconductor components may malfunction and fail due to external interference and surges, so please confirm the performance and quality under actual use. To be on the safe side, please perform safety design on the device (setting of protection circuits such as fuses and circuit breakers, multiple devices, etc.) so that life, body, property, etc. will not be harmed in the event of a malfunction. To prevent injuries and accidents, please be sure to comply with the following matters:

·The driving current and voltage should be used below the rated values.

Please wire according to the electrical definition . In particular, reverse connection of the power supply may cause accidents due to circuit damage such as heat, smoke, and fire, so please be careful.

·Be careful when fixing the product and connecting the pressure inlet .

Warranty and Disclaimer

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. Sencoch Technology reserves the right to make changes without further notice to any product herein. Sencoch Technology makes no warranty, representation or guarantee regarding the suitability of its product for any particular purpose, nor does Sencoch Technology 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. Sencoch Technology does not convey any license under its patent rights nor the rights of others.

IIC Example Code (Attachment: IIC code example)

```
#include <reg52.h>
#include <math.h>
#define DELAY_TIME 600
#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;

//----delay time_us----
void DELAY(uint t)
{
	while (t != 0)
		t--;
}

//----IIC START CONDITION----
void I2C_Start(void)
{
	SDA = 1;	//SDA output high
	 DELAY(DELAY_TIME);
	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
	DELAY(DELAY_TIME);
	 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;
	 DELAY(DELAY_TIME);
	SCL = 1;
	DELAY(DELAY_TIME);
	SCL = 0;
	DELAY(DELAY_TIME);
}

//----IIC SEND DATA "1"----
```

```
void SEND_1(void)
{
        SDA = 1;
        DELAY(DELAY_TIME);
        SCL = 1;
        DELAY(DELAY_TIME);
        SCL = 0;
        DELAY(DELAY_TIME);
}

//----Check SLAVE's Acknowledge----
bit Check_Acknowledge(void)
{
        SDA = 1;
        DELAY(DELAY_TIME);
        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 WriteI2CByte(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;
                DELAY(DELAY_TIME);
                SCL = 1;
                DELAY(DELAY_TIME);
                F0 = SDA;
                DELAY(DELAY_TIME);
                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----
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(0x6D << 1 + 0);          //IIC WRITE operation, SLAVER address bit: 0x6D
    acktemp = Check_Acknowledge();           //check the SLAVER
    WriteI2CByte(addr);               //address
    acktemp = Check_Acknowledge();
    WriteI2CByte(thedata);          //thedata
    acktemp = Check_Acknowledge();
    I2C_Stop();  //IIC STOP
}

//----Read One Byte of Data,Data from SLAVER to the MASTER----
uchar Read_One_Byte(uchar addr)
{
    bit acktemp = 1;
    uchar mydata;

    I2C_Start();
    WriteI2CByte(0x6D << 1 + 0);          //IIC WRITE operation, SLAVER address bit: 0x6D
    acktemp = Check_Acknowledge();
    WriteI2CByte(addr);
    acktemp = Check_Acknowledge();
    I2C_Start();
    WriteI2CByte(0x6D << 1 + 1);          //IIC READ operation, SLAVER address bit: 0x6D
    acktemp = Check_Acknowledge();
    mydata = ReadI2CByte();
    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++)
            ;
}

//----The main function----
void main(void)
{
     uchar pressure_H, pressure_M, pressure_L, temperature_H, temperature_L;
     //temporary variables of pressure and temperature
     long int pressure_adc, temperature_adc;
     //The value of pressure and temperature converted by the sensor's ADC
     long float pressure, temperature;
     //The calibrated value of pressure and temperature
     Delay_xms(1000);        //delay 1000ms

    while (1)
    {
```

```
        Write_One_Byte(0x30, 0x0A);
          //indicate a combined conversion (once temperature conversion immediately followed by once sensor
          signal conversion)
          //more measurement method, check Register 0x30
        while ((Read_One_Byte(0x30) & 0x08) > 0);
          //Judge whether Data collection is over
        Delay_xma(20);

        pressure_H = Read_One_Byte(0x06);
        pressure_M = Read_One_Byte(0x07);
        pressure_L = Read_One_Byte(0x08);
          // Read ADC output Data of Pressure
        pressure_adc = pressure_H * 65536 + pressure_M * 256 + pressure_L;
          //Compute the value of pressure converted by ADC

          if (pressure_adc > 8388608)
              pressure = (pressure_adc - 16777216) / 4;        //unit is Pa
        else
              pressure = pressure_adc / 4;    //unit is Pa
          //The conversion formula of calibrated pressure, its unit is Pa

         temperature_H = Read_One_Byte(0x09);
        temperature_L = Read_One_Byte(0x0A);
        //Read ADC output data of temperature
          temperature_adc = temperature_H * 256 + temperature_L;
          //Compute the value of temperature converted by ADC
                temperature = (temperature_adc – 65536) / 256;          //unit is ℃
          else
                temperature = temperature_adc / 256;          //unit is ℃
          //The conversion formula of calibrated temperature, its unit is Centigrade

        Delay_xms(100);   //delay 100ms
    }
}
```