GZP6847DC

Pressure Sensor

Digital Output(IIC)

Datasheet

Version： V1.3

Issued Data： 2025.09.19

Table of Contents

Document Revision History

Revision	Description	Date
V1.0	Initial release	2024.12.23
V1.1	Change performance data and IIC code	2025.01.21
V1.2	Update drawings, templates and code	2025.07.04
V1.3	Modify the routine	2025.09.19

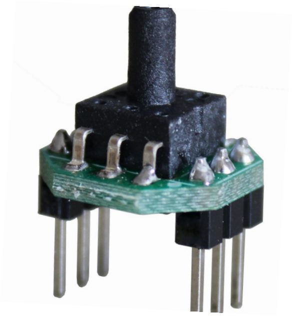
The company reserves the right to make changes in the specifications contained herein without further notice.

1. Product Description

The GZP6847DC pressure sensor is a state-of-the-art MEMS pressure sensor designed particularly for a wide pressure measurement application in medical electronics, automotive electronics, and sports fitness equipment. It is composed of a silicon piezoresistive pressure sensing chip and a signal conditioning integrated circuit chip. 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

- Multiple range from -100…0 to 1…1500kPa
- Gauge pressure
- DIP6 package
- 2.7~5.5V power supply
- IIC output
- Suitable for non-corrosive gases



1.2. Application Areas

- Medical fields such as respirators, oxygen generators, monitors, hemostasis devices, nebulizers, etc.
- Automotive electronics fields such as power steering, braking assistance, etc.
- Sports and fitness equipment fields such as massagers, massage chairs, air mattresses, etc.
- Domestic appliances fields such as water heaters, ozone water machines, beer machines, coffee machines, air pumps, electric breast pumps, vacuum cleaners, etc.
- Vacuum pumps, pressure gauges, pneumatic tools, etc.

2. Functional Description

This product is made with advanced micro-electromechanical principles. The crucial technology is the MEMS pressure sensor chip based on the silicon piezoresistive effect and the high-performance signal conditioning AISC chip. The silicon micro-piezoresistive MEMS pressure sensor chip is connected to a Wheatstone bridge composed of four strain-sensitive resistors. The output signal is amplified, temperature compensated and linearized by the ASIC chip. The linearity and temperature compensation of the transfer function are realized by the digital processing circuit in the ASIC. The high-precision pressure measurement in the full operating temperature range is achieved through the polynomial compensation algorithm and multi-point pressure calibration technology at multiple temperatures.

2.1 Pin Definition

The pin configuration of the pressure sensor is shown in Figure 1.

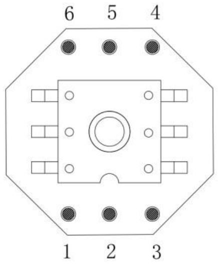


Fig.1 Pin Diagram

Tab.1 Pin Definition

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

2.2 Block Diagram

The sensor functional block diagram is shown in Figure 2.

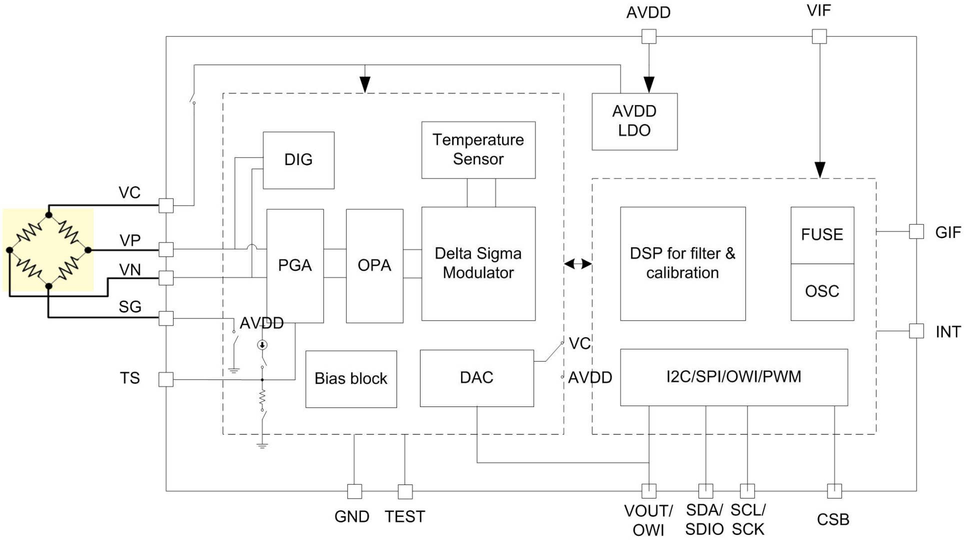


Fig.2 Block Diagram

2.3 Accuracy

GZP6847DC 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 s cale 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.

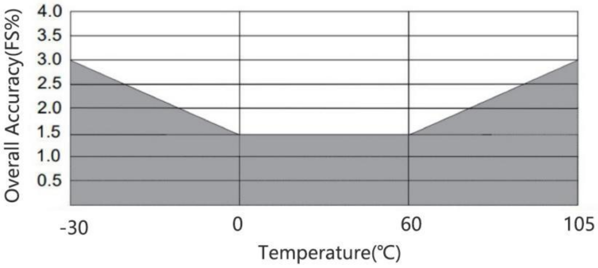


Fig.3 Relationship between overall accuracy and temperature

Tab.2 Overall Accuracy

Temperature (°C)	Overall Accuracy(Full Span)
-30~105	±3.0%
0~60	±1.5%

*Different pressure ranges have different overall errors. Please consult customer service for more details.

3. Technical Indicators

Measured at a power supply of (5±0.25) V DC and a temperature of 25°C.

3.1 Maximum Rated Parameters

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

Tab.3 The maximum rated parameters

Parameter	Min.	Typ.	Max.	Unit	Remark
Supply Voltage	-0.3		6.5	V	
ESD Protection		±2		kV	HBM
Overload Pressure	4X (Range≤60kPa)			Rated	
	2.5X (60kPa≤Range≤200kPa)			Rated	
	1.5X (Range≥200kPa)			Rated	
Bursting Pressure	5X (Range≤60kPa)			Rated	
	3X (60kPa≤Range≤200kPa)			Rated	
	2X (Range≥200kPa)			Rated	
Working Temperature	-30		105	°C	
Storage Temperature	-40		125	°C	

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

Parameter	Value	Unit	Remark
Pressure Range	-100…0 ~ 1…1500	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	°C	@0~60°C
Temperature Resolution	16	Bits	LSB = (1/256) °C
Compensation Temperature	0 ~ 60	°C	Customizable
Pull-up Resistors	2.2	K ohm	
Clock Frequency	400	KHz	Max.
Response Time	2.5	ms	OSR_P=512X

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

4. Application Circuit

The recommended application circuit is shown in Figure 4.

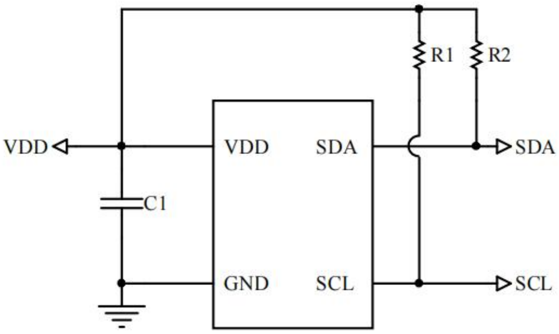


Fig.4 Application circuit

Notice :

1. The recommended value of C1 is 100nF, and the recommended values of R1 and R2 are 4.7k.
2. Please confirm the electrical definition before assembly.
3. Do not have any electrical connection to the NC pin, otherwise it may cause product failure.
4. Provide anti-static protection during welding.

5. Overload voltage (6.5Vdc) may burn out the circuit chip.

6. 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 0x58.

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.

- ■ I²C Timing Diagram

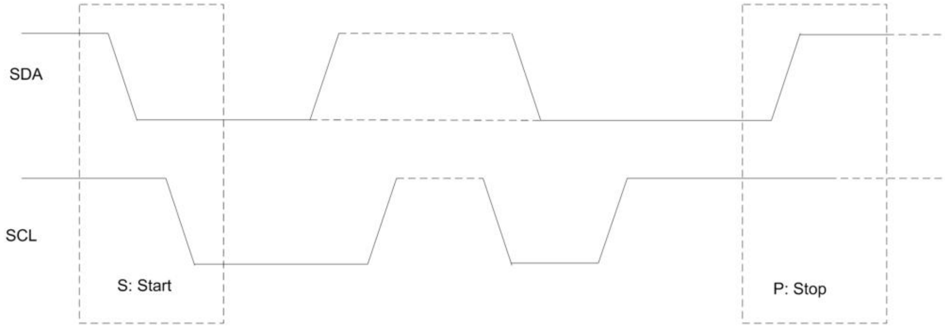


Fig.5 I²C timing diagram 1

- I^2C Protocol

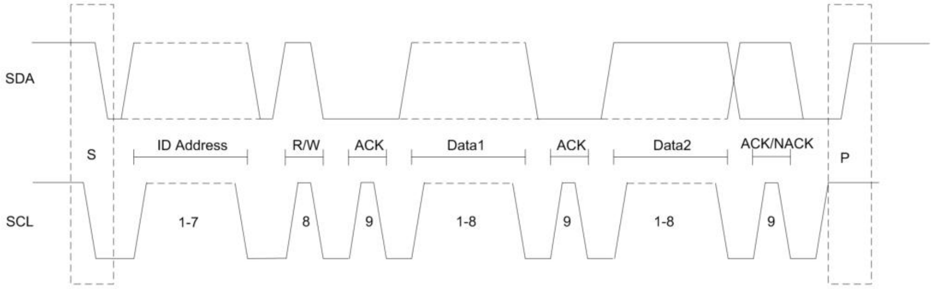


Fig.6 I^2C timing diagram 2

6. Register Description

The register description is shown in Table 5.

Tab.5 Register Description

Add.	Description	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x00	ID	R	ID<7:0>							
0x01	Chip_Control	R/W	/		data_Ready	/	data_out	measurement_ctrl	Active<1:0>	
0x02	CFG_OSR	R/W	OSR_T<7:5>			OSR_P<4:2>			MODE[1:0]	
0x03	CFG_MEAS	R/W	/		T_SB[5:3]			PT_R[2:0]		
0x04	P_data	R	Data out<23:16>							
0x05	P_data	R	Data out<15:8>							
0x06	P_data	R	Data out<7:0>							
0x07	T_data	R	Temp out<15:8>							
0x08	T_data	R	Temp out<7:0>							
0x24	CFG_OPER	R/W	reserved<7:1>						DAC_EN	

Reg0x00 I^2C device address, the default address is 0x58.

Reg0x01 (factory pre-configured)

Chip Control Register

active<1:0>: 00, the chip is powered off; 01, the chip is powered on;

measurement_ctrl: 0, pressure measurement; 1, temperature measurement;

data_out: 0, output calibration data; 1, output original data;

data_ready: 0, data conversion is not completed; 1, data conversion is completed.

Reg0x02 (factory pre-configured)

MODE[1:0]: 00: Sleep mode, 01: Normal mode, 10: One shot mode

OSR_P[4:2]: (pressure oversampling):

000: over sampling x 256

001: over sampling x 512

010: over sampling x 1024

011: over sampling x 2048

100: over sampling x 4096

101: over sampling x 8192

110: over sampling x 16384

111: over sampling x 32768

OSR_T[7:5] (temperature oversampling):

000: over sampling x 256

001: over sampling x 512

010: over sampling x 1024

011: over sampling x 2048

100: over sampling x 4096

101: over sampling x 8192

110: over sampling x 16384

111: over sampling x 32768

Reg0x03 (factory pre-configured)

PT_R[2:0]: 000: 64/1, 001: 32/1, 010: 16/1, 011: 8/1, 100: 4/1, 101: 1/1, Others: 128/1 (pressure/temperature measurement ratio in normal mode)

T_SB[5:3]: 000: 0ms, 001: 62.5ms, 010: 125ms, 011: 250ms, 100: 500ms, 101: 750ms, 110: 1000ms, 111: 2000ms (standby time setting in normal mode)

Reg0x04-Reg0x06

Pressure Data Register

Reg0x07-Reg0x08

Temperature Data Register

Reg0x24

DAC_EN: 0: disable DAC, 1: enable DAC

7. Working Mode Description

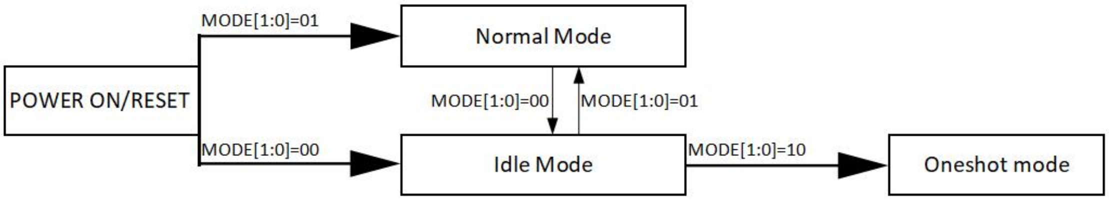


Fig.7 Working Mode

Normal Mode:

When powered on, the sensor automatically enters Normal Mode. If switching from another mode to Normal Mode, it can be enabled by writing 01b to the MODE register (0x02[1:0]). The pressure and temperature sensor signals output measurement data cyclically at a predetermined frequency(Normally the standby time was pre-configured with 0mS).

One Shot Mode:

It can be enabled by writing 10b to the MODE register (0x02[1:0]). The user can specify whether to measure the temperature or pressure signal by clearing or setting the measurement_ctrl bit (0x01[2]). After completing a single measurement, the sensor enters Idle Mode to await the next command.

Idle Mode: The sensor keep the low-consumption sleep situation till it is activated.

In normal mode, read 5 bytes continuously from 0x04 to 0x08 after power-up (ASIC will automatically refresh the data. The first 3 bytes are the pressure data, later 2 bytes are temperature data.

Pressure Calculation

Sum = (0x04 value * 2^16 + 0x05 value * 2^8 + 0x06 value),

If sum < 8388608, P=sum / 2^21*(PMAX - PMIN)   (Unit: Pa)

If sum ≥8388608，P=(sum-2^24)/2^21*(PMAX - PMIN)   (Unit: Pa)

※PMAX is the upper value of pressure range, PMIN is the lower value of pressure range.

Tab.6 Range calibration parameters

Pressure range	Calibration parameters	
	PMIN	PMAX
0-40kPa	0	40000
0-100kPa	0	100000
-100-700kPa	-10000	700000

8. Structure Specification (unit: mm)

Refer to Figure 8 for sensor dimensions (Tolerance ±0.5mm).

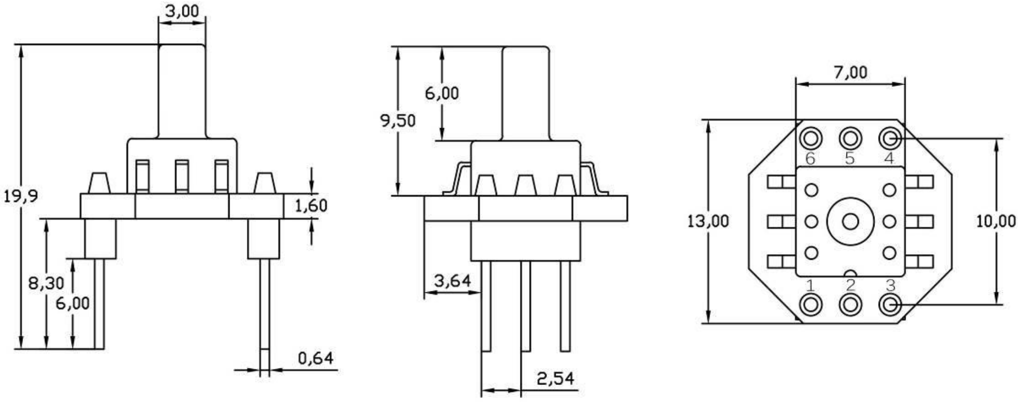


Fig.8 Product Dimensions

Recommended Footprint Layout refer to Figure 9.

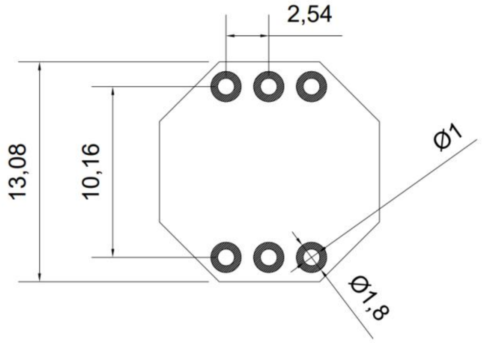


Fig.9 Recommended footprint

9. Order Guide

GZP 6847 DC - 040KPP F01 WX

Tab.7 Order Guide

GZP	Pressure Sensor Series
6847	Product Series
D	Output type A: Analog output D: IIC output
C	IIC Communication Content Format
040KPP	Pressure range: 040 means the measured pressure value is 0~40 (including 0~40, -40~0, -40~40) Pressure unit: KP: KPa MP: MPa PS: PSI BA: Bar Pressure type: P: positive pressure (e.g. 0~40) N: negative pressure (e.g. -40~0) W: negative pressure to positive pressure (e.g.-40~40) Therefore, 040KPP means the measured pressure from 0KPA to 40KPA
F01	Packaging Method: B01: Reel&Tape ; F01: Tube
WX	Company interior code

10. Model Example

Tab.8 Model example

Pressure Range	Model
0～10kPa	GZP6847DC010KPP F01 WX
0～20kPa	GZP6847DC020KPP F01 WX
0～40kPa	GZP6847DC040KPP F01WX
0～100kPa	GZP6847DC101KPP F01WX
0～400kPa	GZP6847DC401KPP F01WX
0～700kPa	GZP6847DC701KPP F01WX
0～1000kPa	GZP6847DC001MPP F01WX
-100～0kPa	GZP6847DC101KPN F01WX
-20～0kPa	GZP6847DC020KPN F01 WX
-100～1000kPa	GZP6847DC001MPW F01WX

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. Instructions 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

· Please use a soldering iron with a head temperature of 260 to 300°C (30 W) and perform the work within 5 seconds.

· When soldering with a load applied to the terminals, the output may change, so be careful. Please keep the soldering iron tip clean.

2) Reflow soldering (SMD terminal type)

- The recommended reflow oven temperature setting conditions are shown:

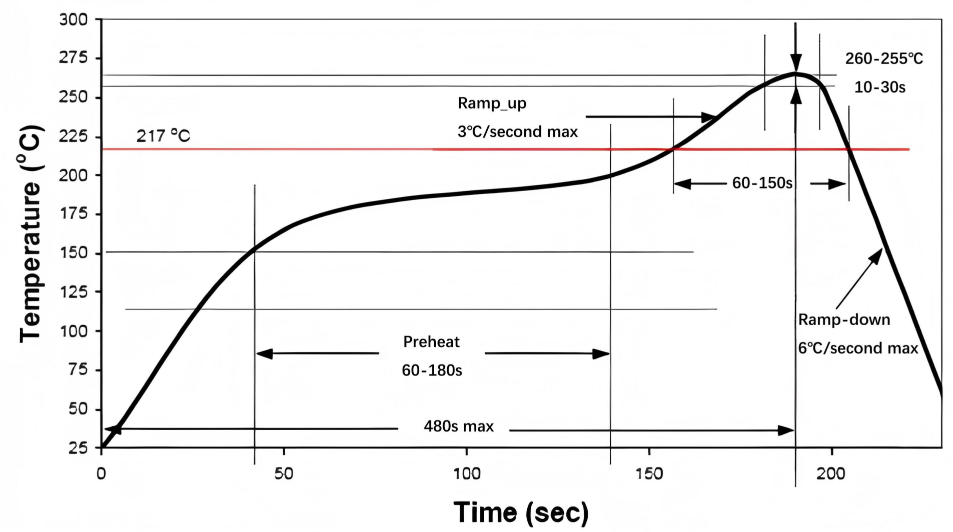


Fig.10 Remelting temperature setting conditions

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, when cutting and bending the substrate, be careful not to cause stress on the soldered parts.

5) Since the sensor terminals are exposed, if metal pieces touch the terminals, abnormal output will occur. Be careful not to touch them with metal pieces or hands.

6) After soldering, when coating is applied to prevent insulation degradation of the substrate, be careful not to allow chemicals to adhere to the sensor.

11.2 Cleaning Requirements

1) Since the product is an open type, be careful not to allow cleaning fluid to enter the 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 a location 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 changes in sensor output or damage.

3) The chip of the pressure sensor is structurally exposed to light, and the output will change. Especially when applying pressure through a transparent cover, please avoid light from reaching the chip of the sensor.

4) Normally packaged pressure sensors can be transported by ordinary transportation tools. Please note: The product should be protected from moisture, impact, 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, such as ultrasonic waves.

3) The only pressure media that can be used directly are air and non-corrosive gases. Other media, especially when used in corrosive media or media containing foreign matter, may cause malfunctions and damage, so please avoid using it in the above environment .

4) There is a pressure sensor chip inside the pressure inlet. If a needle or other foreign object is inserted into the pressure inlet, the chip will be damaged and the inlet will be blocked, so please avoid such operations. In addition, please avoid blocking the atmosphere inlet when using it.

5) Please use the pressure within the rated pressure range. Using outside the range may cause damage.

6) Since static electricity may cause damage, please pay attention to the following matters when using.

Please ground the charged objects and workers on the table to discharge the static electricity around safely.

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

12.Packing Information

Tube Packing

Product Series		Tube (PCS)	Box (PCS)	Box (PCS)	Remark
GZP6847DC	DIP package	38	760	7600	Vacuum packaging

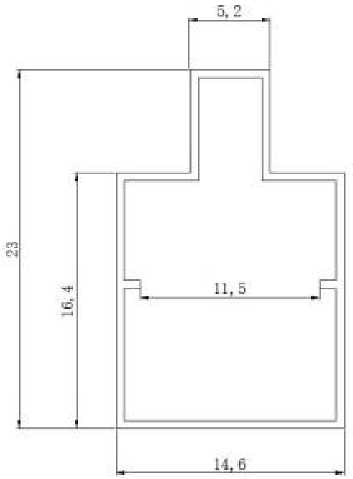
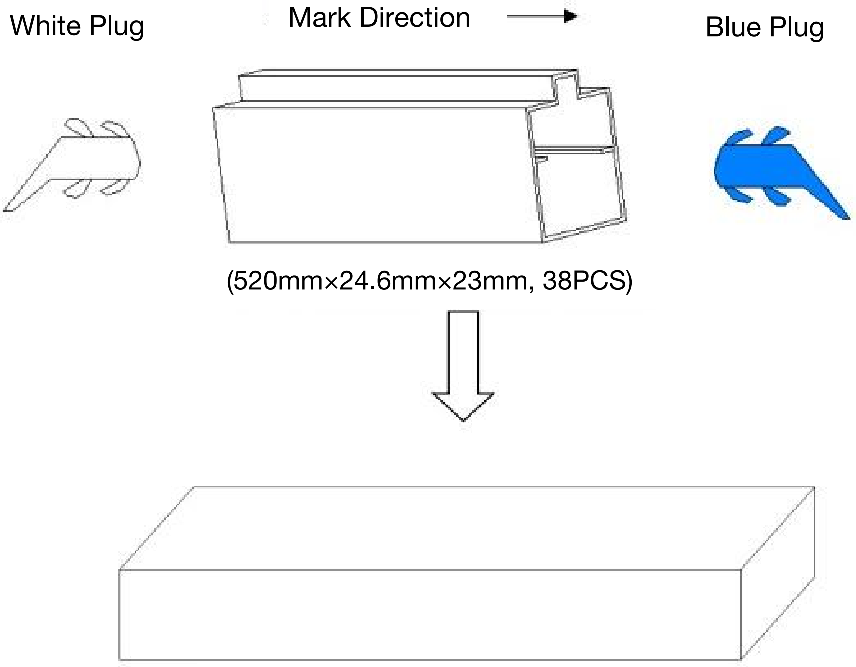


Fig.11 Section Schematic Diagram（Unit:mm）



(530mm×145mm×53mm, 760PCS)

Fig.12 Section Schematic Diagram（Unit:mm）

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

```
#include <reg52.h>
#include <math.h>
#define DELAY_TIME 600    //Time-Delay Parameter
#define TRUE 1
#define FALSE 0
#define uchar unsigned char
#define uint unsigned int

float SPAN = 40;    //SPAN is the span of the sensor

sbit SCL = P1 ^ 7;    //IIC clock line
sbit SDA = P1 ^ 6;    //IIC clock line

//Time-Delay Function
void DELAY(uint t)
{
       while (t != 0)
              t--;
}

void I2C_Start(void)    //IIC Start signal
{
       SDA = 1;    //SDA output high
       DELAY(DELAY_TIME);
       SCL = 1;    //SCL output high
       DELAY(DELAY_TIME);
       SDA = 0;    //SDA output low
       DELAY(DELAY_TIME);
       SCL = 0;    //SCL output low
       DELAY(DELAY_TIME);
}

void I2C_Stop(void)    //IIC Stop signal
{
       SDA = 0;
       DELAY(DELAY_TIME);
       SCL = 1;
       DELAY(DELAY_TIME);
       SDA = 1;
       DELAY(DELAY_TIME);
       SCL = 0;
       DELAY(DELAY_TIME);
}

void SEND_0(void)    //IIC send data "0"
{
       SDA = 0;
       DELAY(DELAY_TIME);
       SCL = 1;
       DELAY(DELAY_TIME);
       SCL = 0;
```

```
    DELAY(DELAY_TIME);
}

void SEND_1(void)    //IIC send data "1"
{
    SDA = 1;
     DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}

bit Check_Acknowledge(void)    //Read ACK signal
{
     char F0 = 0;
    SDA = 1;
     DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME / 2);
    F0 = SDA;
    DELAY(DELAY_TIME / 2);
    SCL = 0;
    DELAY(DELAY_TIME);
    if (F0)
      return FALSE;
    return TRUE;
}

void WriteI2CByte(uchar b) reentrant    //Write One Byte of Data
{
    char i;
    for (i = 0; i < 8; i++)
        if ((b << i) & 0x80)    //Send high bits first
            SEND_1();
        else
            SEND_0();
}

uchar ReadI2CByte(void) reentrant    //Receive one byte
{
    char b = 0, i, F0 = 0;
    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)
        {
            b = b << 1;    //Receive high bits first
            b = b | 0x01;
        }
        else
            b = b << 1;
    }
    return b;
}

//Write a register’s address and a command byte to the sensor
//"addr": register's address, "thedata": the command byte
void Write_One_Byte(uchar addr, uchar thedata)
{
    bit acktemp = 1;
    I2C_Start();    //IIC START Signal
    WriteI2CByte(0x58 << 1 + 0);    //The SLAVER address is 0x58
    // The lowest bit of address is 0 means writing
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    WriteI2CByte(addr);      //Send the register’s address
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    WriteI2CByte(thedata);    //Write command to the sensor
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    I2C_Stop();    //IIC STOP Signal
}

//Read one byte of data from the sensor
uchar Read_One_Byte(uchar addr)
{
    bit acktemp = 1;
    uchar mydata;
    I2C_Start();    //IIC START Signal
    WriteI2CByte(0x58 << 1 + 0);    //The SLAVER address is 0x58
    // The lowest bit of address is 0 means writing
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    WriteI2CByte(addr);      //Send the register’s address
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    I2C_Start();    //IIC START Signal   发
    WriteI2CByte(0x58 << 1 + 1);    //The SLAVER address is 0x58
    // The lowest bit of address is 1 means Reading
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    mydata = ReadI2CByte();     //Read the above register’s data
    acktemp = Check_Acknowledge();    //check the SLAVER’s ACK
    I2C_Stop();    //IIC STOP Signal
    return mydata;
}

//Ms Time-Delay Function
void Delay_xms(uint x)
{
    uint i, j;
    for (i = 0; i < x; i++)
        for (j = 0; j < 112; j++)
            ;
}
```

```
void main(void)   //The main function
{
    uchar pressure_H, pressure_M, pressure_L,temperature_H,temperature_L;
    //Temporary variables used to save bytes of pressure and temperature from the sensor
    long int pressure_ad,temperature_ad;
    //Temporary variables used to save AD values of pressure and temperature from the sensor
    float pressure,temperature,Shift_N;
    //pressure: actual pressure
    //temperature: actual temperature
    uchar byte1,byte2;
    int EOFF;

    Delay_xms(1000);
    while (1)
    {
      Write_One_Byte(0x01, 0x01);
      //Send 0x01 to the register whose address is 0x01 to start a data collection

      Delay_xms(20);

      pressure_H = Read_One_Byte(0x04);   //Read bytes of pressure from the sensor
      pressure_M = Read_One_Byte(0x05);
      pressure_L = Read_One_Byte(0x06);
      pressure_ad = pressure_H * 65536 + pressure_M * 256 + pressure_L;
      //compute the AD pressure of the sensor
      temperature_H = Read_One_Byte(0x07);   //Read bytes of temperature from the sensor
      temperature_L = Read_One_Byte(0x08);
      temperature_ad = temperature_H * 256 + temperature_L;
      //compute the AD temperature of the sensor

      //compute the actual pressure of the sensor
      //pressure’s unit is Pa
      if (pressure_ad >= 8388608)
        pressure = (float) (pressure_ad - 16777216) / 2^21 *(PMAX-PMIN) ;
      else
        pressure = (float) pressure_ad / 2^21 * (PMAX-PMIN) ;

      //compute the actual temperature of the sensor
      //temperature’s unit is Centigrade
      if(temperature_ad > 32768)
        temperature_ad -= 65536;
      byte1 = Read_One_Byte(0x20);   //Read temperature parameter from the sensor
      byte2 = Read_One_Byte(0x21);   //Read temperature parameter from the sensor
      if (byte1 == 0x0C)   //According byte1 to evaluate the variable EOFF
        EOFF = 4096;
      else if (byte1 == 0x8C)
        EOFF = -4096;
      else if(byte1 == 0x0D)
        EOFF = 8192;
      else if(byte1 == 0x8D)
        EOFF = -8192;
      else if(byte1 == 0x0E)
        EOFF = 16384;
        else if(byte1 == 0x8E)
            EOFF = -16384;
        Shift_N = byte2 / 10;    //compute the variable Shift_N
        temperature = (temperature_ad - EOFF) / 2 ^ Shift_N + 25;
        //the actual temperature of the sensor

        printf("Actual pressure is %f Pa\r\n",pressure);
        printf("Actual temperature is %f Centigrade\r\n\r\n",temperature);

        Delay_xms(1000);
    }
}
```