Digital Output Pressure Transmitter

(Model： WPCK89)

Manual

Version： 1.0
Date： 2024-01-26

Zhengzhou Winsen Electronic Technology Co., Ltd

Statement

This manual copyright belongs to Zhengzhou Winsen Electronics Technology Co., LTD. Without the written permission, any part of this manual shall not be copied, translated, stored in database or retrieval system, also can't spread through electronic, copying, record ways.

Thanks for purchasing our product. In order to let customers use it better and reduce the faults caused by misuse, please read the manual carefully and operate it correctly in accordance with the instructions. If users disobey the terms or remove, disassemble, change the components inside of the sensor, we shall not be responsible for the loss.

The specific such as color, appearance, sizes …etc., please in kind prevail.

We are devoting ourselves to products development and technical innovation, so we reserve the right to improve the products without notice. Please confirm it is the valid version before using this manual. At the same time, users' comments on optimized using way are welcome.

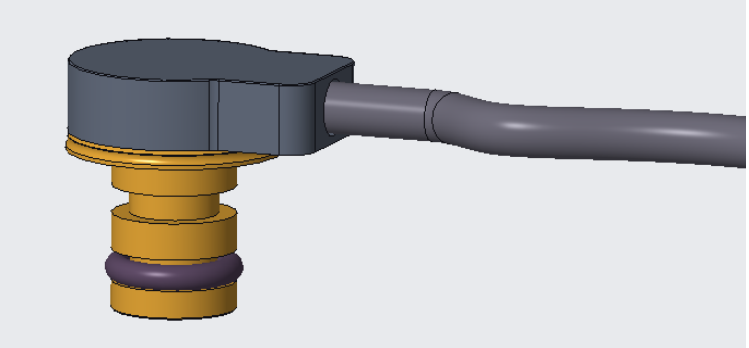
Please keep the manual properly, in order to get help if you have questions during the usage in the future.

Zhengzhou Winsen Electronics Technology CO., LTD

WPCK89 Digital Output Pressure Transmitter

1. Product Descriptio：

The WPCK89 digital output pressure transmitter is a new generation of intelligent digital products, characterized by high precision, high reliability, convenient use and installation. The product has a wide voltage range, low power consumption, and multi-level protection design, with strong anti-interference ability. The standard IIC bus has a maximum speed of 400K, a collection rate of 50 times/second, a power consumption of less than 2mA during operation, and a standby current of 1uA. The standard IIC communication protocol makes this transmitter particularly suitable for computer control systems.



The pressure seat of WPCK89 digital output pressure transmitter is processed with a single piece integrated structure of 316L stainless steel or 17-4PH stainless steel, with high overload performance. Its pressure interface has no welds, silicone oil or other organic substances, and good sealing performance. The sensitive components of the sensor are firmly sintered on the pressure seat using glass micro melting technology, greatly improving the high temperature and vibration resistance of the sensor, ensuring the long-term stability, reliability, and durability of the transmitter in harsh industrial environments. In the standard purification production process, all parameters are strictly controlled, and the entire product has undergone strict testing and aging screening of components, semi-finished products, and finished products, with stable and reliable performance.

2. Feature：

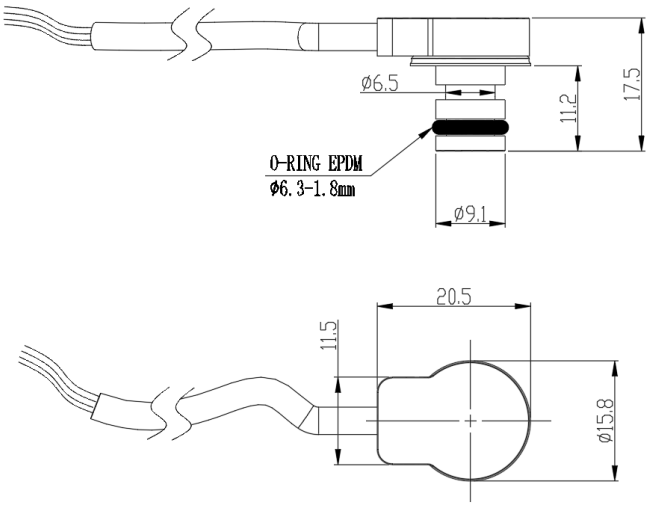
- Stainless steel single piece integrated structure without O-ring
- High Pressure
- Adapt to harsh environments
- Digital signal output
- High reliability and stability
- Wide detection range
- Strong overload capacity
- Wide working temperature range

3. Performance parameter：

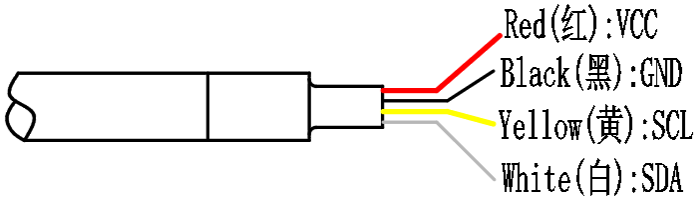
Item	Minimum	Typical	Maxium	Unit
Working Voltage	2.0	3.0	3.6	Vdc
Working Current	1		3	mA
Standby Current	0.1		1	uA
Combined accuracy	-0.5		+0.5	%FS/℃
Pressure accuracy(Including nonlinearity, hysteresis and repeatability)	-0.25		+0.25	%Span
A/D Resolution		24		Bit
Long-term stability (1year)	-0.25		+0.25	%Span
Insulation resistance（250Vdc）	50			MΩ
Overload pressure	1.5×			Rated
Breakdown pressure	3×			Rated
Pressure cycling (Zero-full range)	10			Million
Working temperature	0		55	℃
Storage temperature	-20		85	℃
Compensation temperature	0		45	℃
Temperature measure error	-2		2	℃
Temperature measure resolution		0.1		℃
Start-up time（10% to 90%）			2.5	mS@4MHz(No sleep mode)
Response time（10% to 90%）	0.1		2	mS@4MHz(Sleep mode)
Vibration	±20g		GB/T 2423.48-2018	
Impact	50g		GB/T 2423.5-2019	
Detection range options		0 ～ 100		psig
		0 ～ 150		psig
		0 ～ 250		psig
		0 ～ 500		psig

PS：Please contact the manufacturer if any special request.

4. Overall Dimentions



5. Wiring Defination



直接引线：I^2C 输出

Direct leads: I2C output

PS: The actual color is subject to the factory object, special needs, please contact the manufacturer.

IIC Series Digital Pressure/Temperature Transmitter

Communication Protocol

1. Scope of application：

The IIC series digital pressure/temperature transmitter is a new generation of intelligent digital products, characterized by high precision, high reliability, convenient use and installation. High precision and low-power design of the product; The standard IIC communication protocol makes this transmitter particularly suitable for computer control systems.

2. Performance Parameter

a. Standard IIC bus, 100-400K rate; The default address is 0;

b. Collection rate: 50 times/second, 24 bit AD output data, simultaneously outputting main measurement and auxiliary temperature data;

3. IIC Communication Protocol

The I2C interface is a simple 8-bit protocol that uses a serial data line (SDA) and a serial clock line (SCL), where each device connected to the bus can be addressed by software with a unique address. For detailed specifications of the I2C protocol, please refer to the I2C bus specifications.

3.1 Interface external connection

The bidirectional bus is implemented by the devices (master and slave) using an open-drain output stage and a pull-up resistor connected to the positive supply voltage. The recommended pull-up resistor value depends on the system setup (circuit or cable capacitance and bus clock frequency). In most cases, 2 to 4.7kΩ is a reasonable choice. The capacitive loading on the SDA and SCL lines must be the same. It is important to avoid asymmetric capacitive loading.

3.2 IIC Protocol Defination

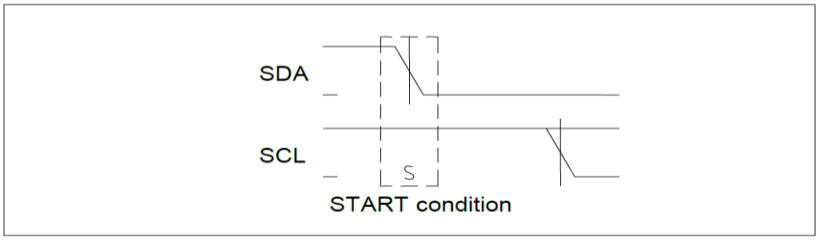
- Idle period

  During the inactive period of the bus, SDA and SCL are pulled up to the power supply voltage VDDA.

- Starting Conditions

When SCL is at a high potential and SDA jumps from a high level to a low level, it means that a starting condition is generated. Any instruction transmitted by the host must be transmitted after a starting condition is generated. The host can generate starting conditions at any time.

I²C Transmission Start Condition

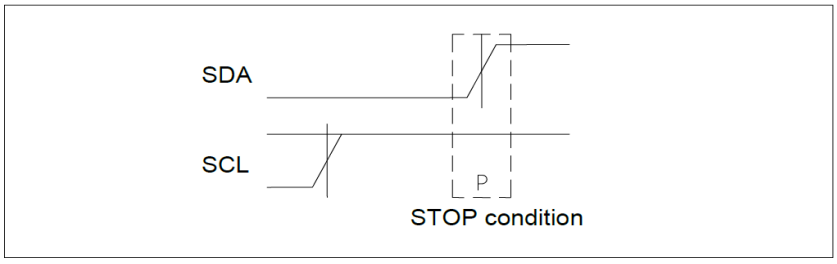


A HIGH to LOW transition on the SDA line while SCL is HIGH

- Stop Conditions

SCL is at a high potential while SDA jumps from a low level to a high level, indicating a stopping condition. After the stop condition is generated, the command transmission is completed, and the transmission begins to execute internal command codes.

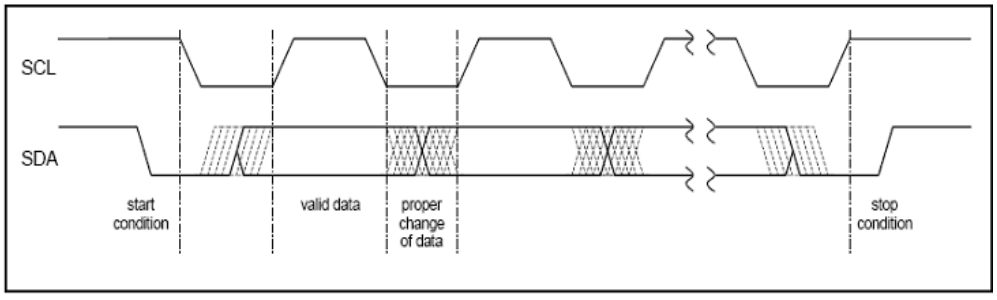
I²C Transmission Stop Condition



A LOW to HIGH transition on the SDA line while SCL is HIGH

- Valid Data

The data is transmitted in bytes (8 bits), and the high bits are transmitted first before the low bits. After each byte transmission is completed, the receiving device will return an acknowledgment bit. After the effective start condition, if SCL is at a high potential and SDA is at a stable voltage level, the transmitted data is valid. When SCL is at a low potential, the voltage of SDA can only change.



IIC Communication protocol rules

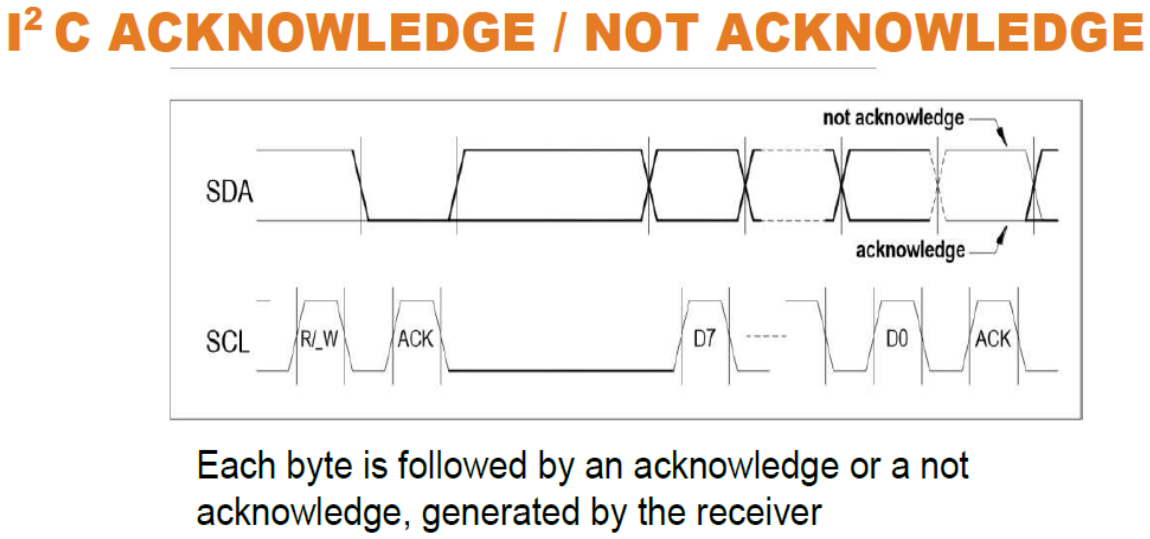
- Confirmed/unconfirmed

The master must wait for the slave to send back a confirmation pulse after transmitting one byte of data, by lowering the SDA voltage during the confirmation clock period. If the slave does not send a confirmation signal, the slave device will be paused. At this point, the host can generate a stop condition to stop communication and then resend the previous command.。

When the host acts as a receiver, a stop bit is added at the end of the transmitted instruction instead of an

acknowledgment signal.

When the transmission acts as a slave and receives the previous command and is processing this command program, the interface will be disabled.



- Addressing

Each slave device on the I2C bus has a specific address. When a starting condition is generated, the host will send an address byte that contains a 7-bit address and a read/write (R/W) control bit. Here, "0" indicates that the data is sent (written) from the host to the slave device; And "1" is exactly the opposite, indicating "reading" data.

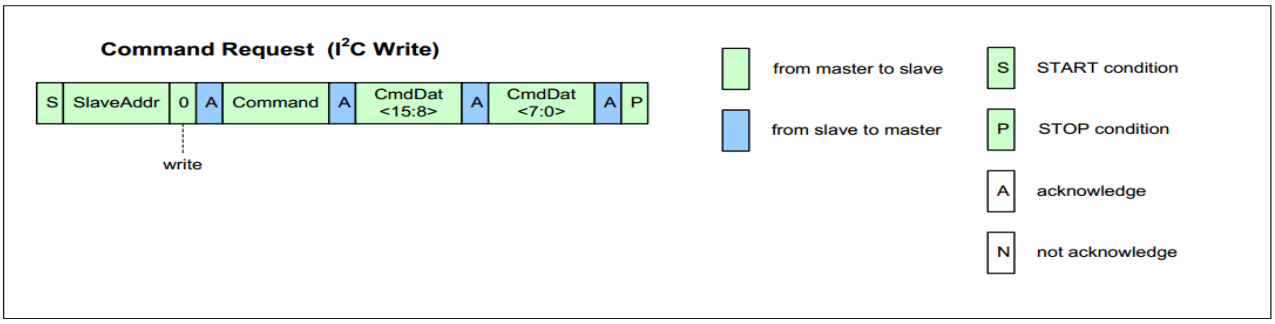
After receiving the instruction, the device at the corresponding address will reply with an "confirm" signal, while other slave devices connected to the I2C bus will not respond to this instruction.

The initial address of the transmitter is 0x00 (7 bits), which can be modified through dedicated commands.

3.3 IIC Data read and write operations

- Write operation

Writing data is transmitted from the host to the slave device, usually with one instruction byte and two bytes of data after an address byte (note: some instructions do not require these two bytes of data). The internal controller receives this instruction and processes the relevant program.

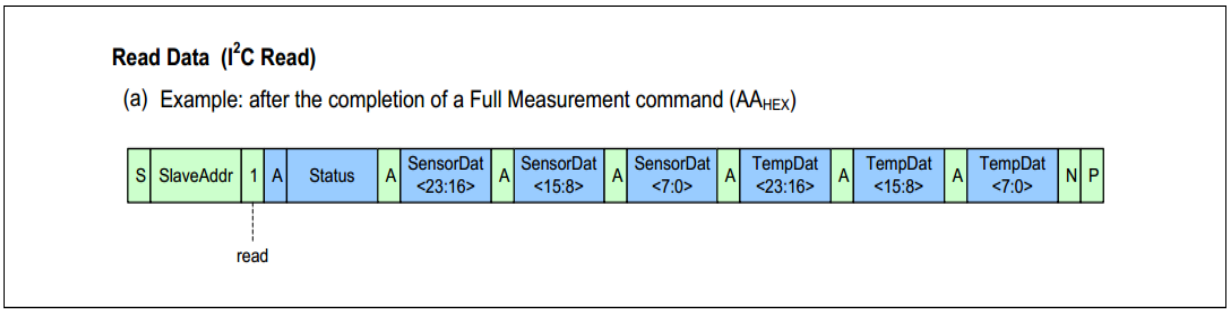


IIC Write Operation

- Write Operation

After the host sends an address data request containing read and write directions to the slave device, the slave device responds to this instruction and sends back the data through the activated serial port register. During this process, the host must generate a serial clock SCL, an acknowledgment signal (except for the last data exception), and a stop condition.

After each transmitter signal measurement and calibration is completed, the final output data processed by CMC calculation is transmitted to the serial port register. At this time, only a read instruction is needed to obtain the data from the slave device.



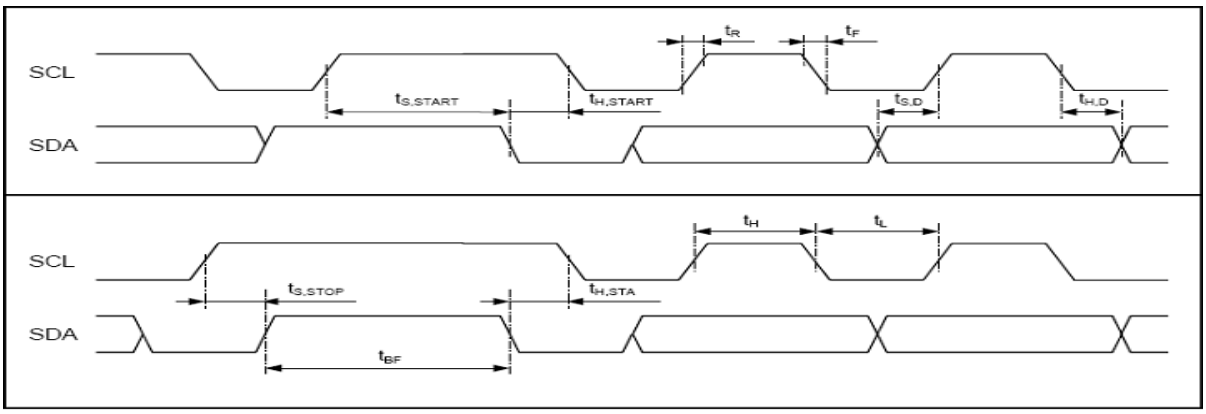
IIC Write operation

Status Byte

Bit	7	6	5	4	3	2	1	0
Meaning	0	Powered?	Busy?					ALU Saturation?

ALU Saturation(bit 0)： 1 indicates data overflow error, 0 indicates data correctness.

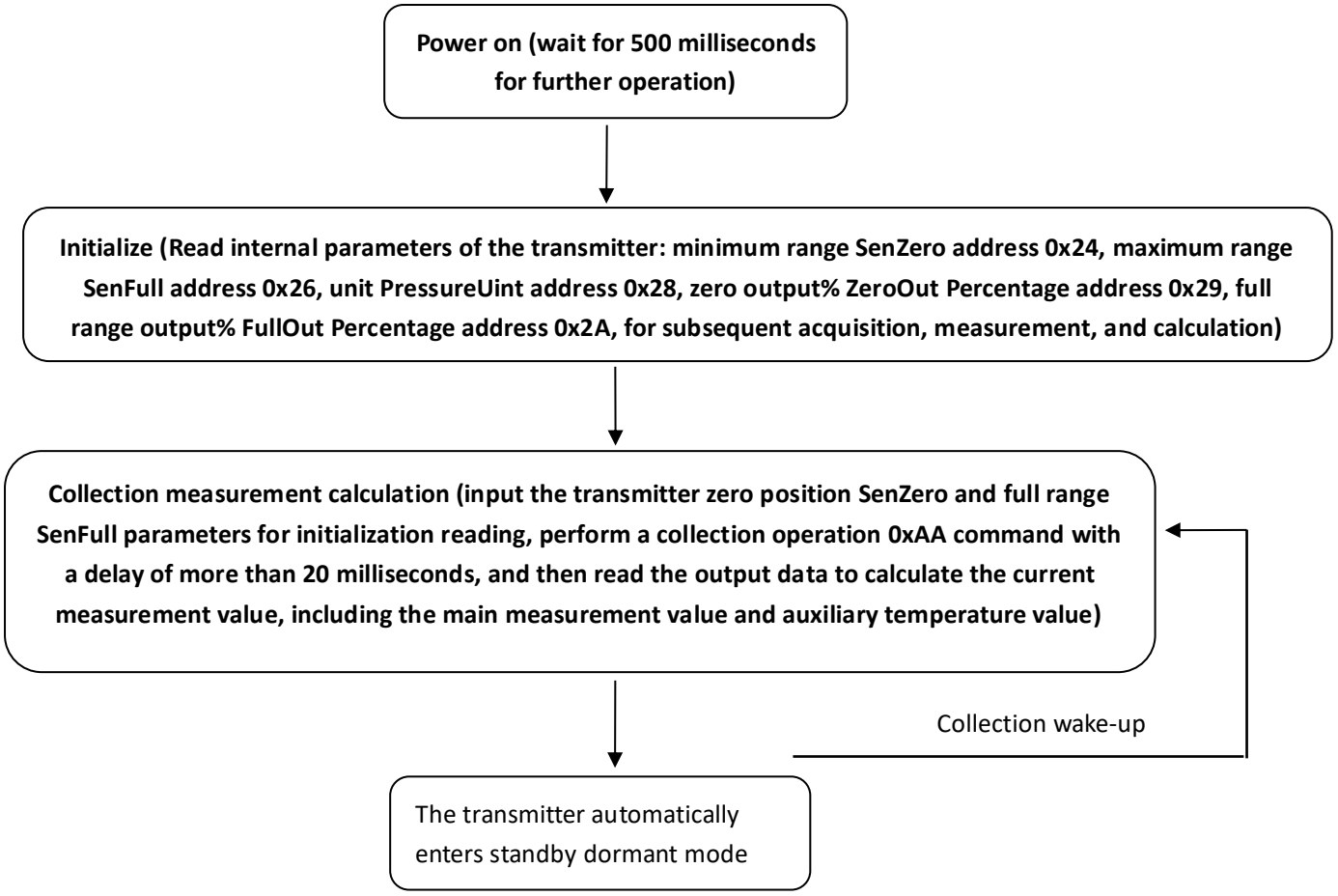
3.4 IIC Timing Parameters



IIC IIC Communication Timing

Item	Parameters	Symbol	Min	TYP	Max	Unit	Conditions
1	SCL Clock Frequency	f_{SCL}		100	400	kHz	
2	Bus idle time Between start and stop conditions	$t_{I2C,BF}$	1.3			μs	
3	The duration of the start condition	$t_{I2C,H,Start}$	0.6			μs	
4	The creation time of the rebuild start condition	$t_{I2C,S,Start}$	0.6			μs	
5	SCL/SDA Low voltage period	$t_{I2C,L}$	1.3			μs	
6	SCL/SDA High voltage period	$t_{I2C,H}$	0.6			μs	
7	Data duration	$t_{I2C,H,D}$	0			μs	
8	Data setup time	$t_{I2C,S,D}$	0.1			μs	
9	SCL/SDA Upward jump time	$t_{I2C,R}$			0.3	μs	
10	SCL/SDA Downward jump time	$t_{I2C,F}$			0.3	μs	
11	Stop condition establishment time	$t_{I2C,S,Stop}$	0.6			μs	
12	Noise interference SDA	$t_{I2C,NI}$			50	ns	Noise peak is weakened

4. Operation process and examples



After the transmitter is powered on, it will sleep immediately; the host sends a collection command to wake up the transmitter to measure and collect data, and then automatically enters the sleep state; the cycle continues.

The complete operating procedure for the transmitter has been written by our company, please refer to the attached file Sensor for details_ I2c_ Moni. c, Sensor_ I2c_ Moni. h. This program is a correctly running program, and users only need to port the I2C interface part based on their own MCU. The transmitter operation part cannot be modified. The I2C interface of the example file is simulated through a regular IO port. If the MCU used has a hardware I2C interface, the hardware interface should be used first; Unless there is no hardware interface, simulate the I2C interface. The main function is as follows:

```
void SenMeasure(void)
{
  static SENDATA SensorValue;
  static double SensorPressure,SensorTemp;

  Sensor_Init(0);
  while(1)
  {
      SensorValue = GetPresValue(0,SenZero,SenFull);// Pressure and temperature collection
      SensorPressure = SensorValue.Pressure;
```

```
    SensorTemp = SensorValue.Temp;
    delay_ms(10);
  }
}
```

1. Initialize the transmitter before powering onSensor_Init(0)，The default address of the transmitter is 0. See below：

```
void Sensor_Init(unsigned char I2CAddress)
{
  unsigned char sdbuf[16],rdbuf[16];
  rdbuf[0]=0;
  SendStr(I2CAddress,rdbuf,1);// After the transmitter is initially powered on, the I2C bus communication method confirms the command.
  delay_ms(3);

  sdbuf[0] = 0x0;
  sdbuf[1] = 0x0;
  sdbuf[2] = 0x0;
  SendStr(I2CAddress,sdbuf,3);
  delay_ms(1);
  RcvStr(I2CAddress,rdbuf,3);
  delay_ms(1);

  if((rdbuf[0]==0x40)||(rdbuf[0]==0x44)||(rdbuf[0]==0x60))
      {
      SenNumberL = (unsigned long) (ReadSenData(I2CAddress,0x20,3));//Low bit of Serial number: 0x9ABCDEF1
      SenNumberH = (unsigned long) (ReadSenData(I2CAddress,0x22,3));//High bit of serial number: 0x12345678
      SenZero = ReadSenData(I2CAddress,0x24,4);// Transmitter Zero position value: 10
      SenFull = ReadSenData(I2CAddress,0x26,4);// Transmitter full range value: 1000
      PressureUint = (unsigned short int) ReadSenData(I2CAddress,0x28,1);// Transmitter pressure unit
      ZeroOutPercent = (unsigned short int) ReadSenData(I2CAddress,0x29,1);//Transmitter zero position AD output %
      FullOutPercent = (unsigned short int) ReadSenData(I2CAddress,0x2A,1);//Transmitter full AD output %
      }
  }
```

Initialization includes reading the serial number, transmitter zero position, and full range; Transmitter unit; Output%. These are the main parameters of the transmitter, which must be used for subsequent measurements. Users cannot modify them and can only read them. This step is also the key to verifying the correct user migration. Only when the I2C interface is operated correctly can the parameters be read correctly. Zero position, full range consistent with nameplate. Zero position, full range% is generally 10, 90.

2. Measure and collect GetPresValue (0, SenZero, SenFull), address 0, and input the zero and full range values for initialization reading. Perform a measurement and collection, and calculate the current measurement value - main measurement value and auxiliary temperature value. The returned structure contains two values, SensorPressure and SensorTemp.

Zhengzhou Winsen Electronics Technology Co., Ltd
Add: No.299, Jinsuo Road, National Hi-Tech Zone, Zhengzhou 450001 China
Tel: +86-371-67169097/67169670
Fax: +86-371-60932988
E-mail: sales@winsensor.com
Website: www.winsen-sensor.com