

Product Overview

NSPDS7 is a calibrated differential pressure sensor series product launched by NOVOSENSE for low pressure measurement market. This series use a high performance ASIC to calibrate and compensate the MEMS sensor element. While ensuring the reliability of the product, the two chips are integrated and packaged, reduces the package size greatly, this series provide JEDEC standard SOIC-16 package with vertical porting. The pressure signal from 0.5kPa differential to 250kPa differential can be converted into I²C output signal with a customizable output range. The pressure sensor can be directly mounted on a standard printed circuit board very suitable for ventilators, sleep apnea and IOT applications.

Key Features

- Large pressure range
Pressure range 0.5kPa to 250kPa
- Operating temperature range -20°C to 70°C
- Total error band initially better than $\pm 1\%$ F.S.
- 24bit I²C digital output
- Vertical port
- RoHS & REACH-compliant

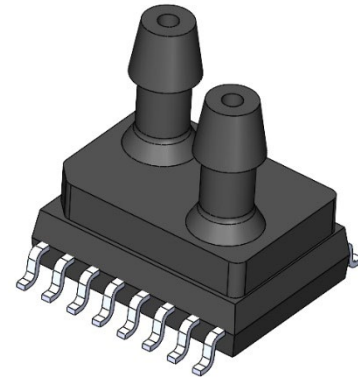
Applications

- Residual pressure for fire protection
- Ventilators
- CPAP/sleep apnea
- HVAC/NAV
- Safety cabinets
- Pressure switches

Device Information

Part Number	Package	Body Size
NSPDS7	SOIC16	10.26mm×7.52mm

Outline



INDEX

1. PIN CONFIGURATION AND FUNCTIONS	3
2. ABSOLUTE MAXIMUM RATINGS	4
3. ESD RATINGS.....	4
4. OPERATING RANGE	4
5. SPECIFICATIONS	5
5.1. ELECTRICAL CHARACTERISTIC	5
5.2. I2C TIMING DIAGRAM	5
5.3. I2C ELECTRICAL CHARACTERISTICS	6
6. FUNCTION DESCRIPTION	6
6.1. OVERVIEW	6
6.2. DIGITAL OUTPUT TRANSFER FUNCTION.....	7
6.3. REGISTER MAP	7
6.4. I2C INTERFACE	8
7. TYPICAL APPLICATION.....	10
7.1. APPLICATION CIRCUIT	10
8. PACKAGE INFORMATION.....	11
9. ORDER INFORMATION.....	12
10. SOLDERING PARAMETERS.....	13
10.1. REFLOW SOLDERING (SMD TERMINAL).....	13
10.2. MANUAL SOLDERING	13
11. PACKING INFORMATION	14
12. IDENTIFICATION CODE	14
13. REVISION HISTORY	15
14. NOTES	16
14.1. I2C ROUTINE	16

1. Pin Configuration and Functions

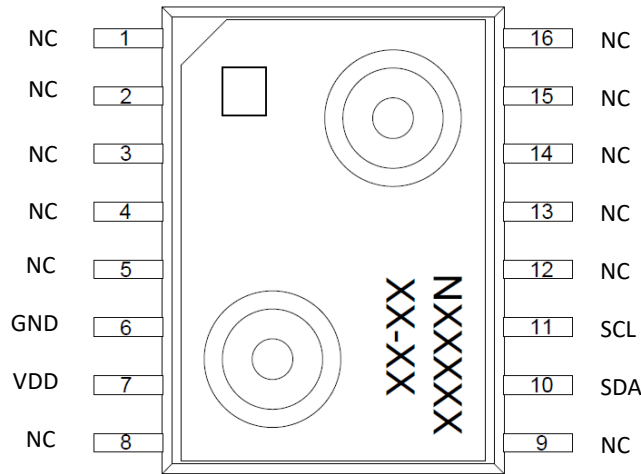


Figure 1.1 NSPDS7 series pin definition (Top view)

Table 1.1 NSPDS7 pin description

Pin NO.	Pin Name	Description
1	NC	No connect
2	NC	No connect
3	NC	No connect
4	NC	No connect
5	NC	No connect
6	GND	Ground
7	VDD	Power supply
8	NC	No connect
9	NC	No connect
10	SDA	I ² C data signal
11	SCL	I ² C clock signal
12	NC	No connect
13	NC	No connect
14	NC	No connect
15	NC	No connect
16	NC	No connect

2. Absolute Maximum Ratings

Parameters	Symbol	Min	Typ	Max	Unit	Comments
Supply voltage	VDD _{max}	-0.3		6.5	V	
Analog output current limit				25	mA	
Digital pin voltage		-0.3		VDD+0.3	V	25°C
Proof pressure	P _{proof}	30			kPa	Operating pressure range <20kPa@25°C
		300				Operating pressure range ≥20kPa@25°C
Burst pressure	P _{burst}	50			kPa	Operating pressure range <20kPa@25°C
		350				Operating pressure range ≥20kPa@25°C
Storage temperature	T _{stg}	-40		85	°C	

3. ESD Ratings

Ratings		Value	Unit
Electrostatic discharge	Human body model (HBM), per AEC-Q100-002-RevE	±2.5	kV
	Charged device model (CDM), per AEC-Q100-011-RevB	±500	V
	Latch up (LU), per JESD78F	±100	mA

4. Operating Range

Parameters	Symbol	Min	Typ	Max	Unit	Comments
Supply voltage	VDD	3	3.3	3.6	V	
		4.5	5	5.5	V	
Operating pressure range	P _{range}	0.5		250	kPa	P _{max} – P _{min}
I ² C clock frequency	F _{sclk}			400	kHz	
Operating temperature	T _{opr}	-20		70	°C	

5.3. I²C Electrical Characteristics

Parameters	Symbol	Min	Typ	Max	Unit	Comments
Clock frequency	f _{scl}			400	kHz	
SCL low pulse	t _{LOW}	1.3			μs	
SCL high pulse	t _{HIGH}	0.6			μs	
SDA setup time	t _{SUDAT}	0.1			μs	
SDA hold time	t _{HDDAT}	0.0			μs	
Setup time for a repeated start condition	t _{SUSTA}	0.6			μs	
Hold time for a start condition	t _{HDSTA}	0.6			μs	
Setup time for a stop condition	t _{SUSTO}	0.6			μs	
Time before a new transmission can start	t _{BUF}	1.3			μs	

6. Function Description

6.1. Overview

NSPDS7 uses a MEMS piezoresistive differentia pressure sensor element as a pressure sensitive component that provide an original signal output that is proportional to ambient pressure. The built-in conditioning IC drives the sensitive component and amplifies, temperature compensates, and linearizes the original signal to output a digital signal that is linear with the applied pressure.

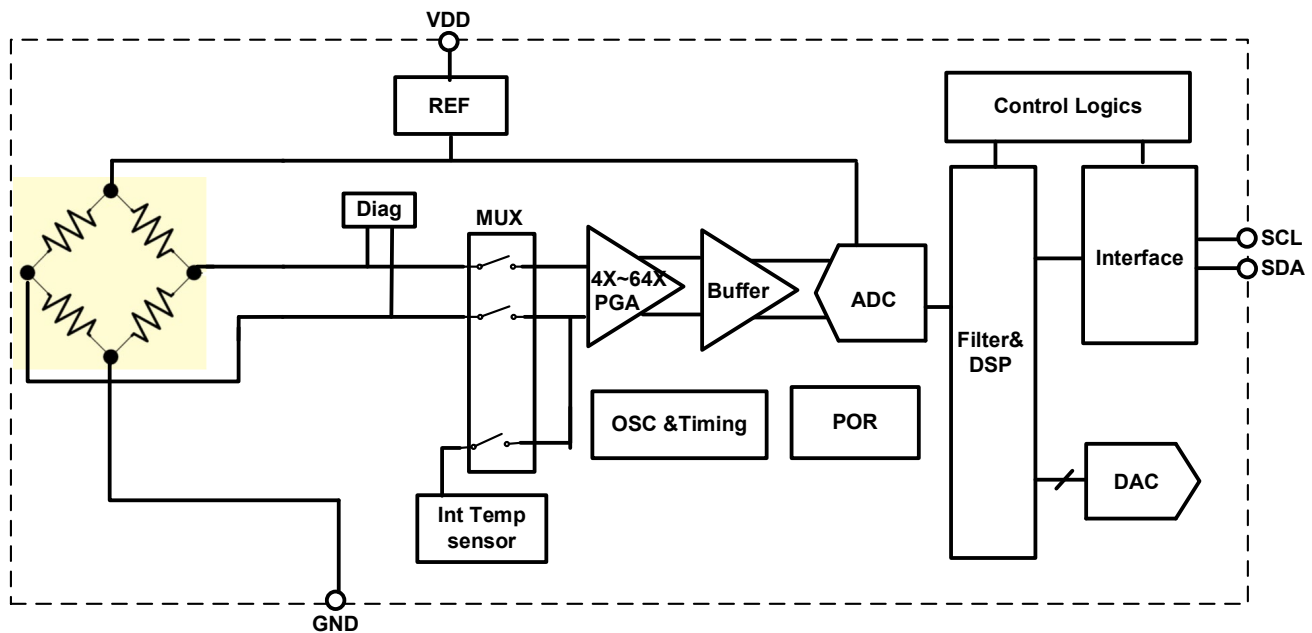


Figure 6.1 Product Function Block Diagram

6.2. Digital Output Transfer Function

$$P=A \times \text{code}/8388607+B$$

Code is the register 0x06~0x08 value;

P is the pressure value, differential pressure ,unit is kPa;

Table 6.1 Digital Output Transfer Function Coefficient

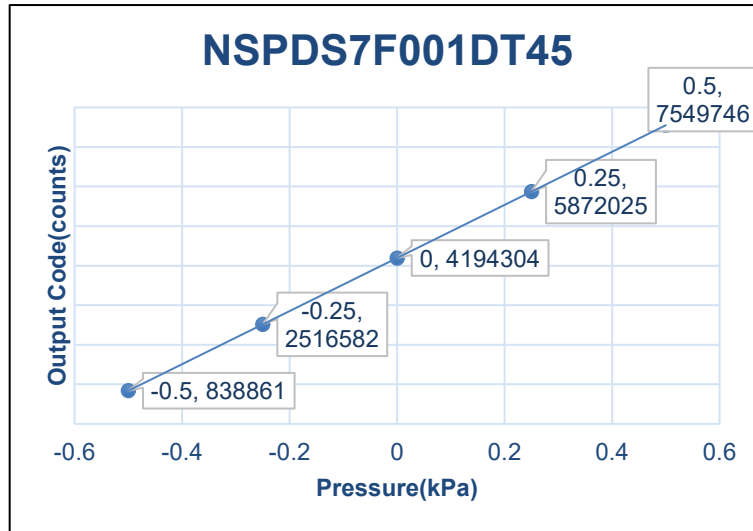


Figure 6.2 Transfer Function

6.3. Register Map

Addr	Bit Addr	Description	Default	Description
0x30	7 – 4	Reserve	4'b0000	Write with 0x0A to start a conversion, automatically come back to 0x02 after conversion ends.
	3	Sco	1'b0	
	2 – 0	Measurement_ctrl<2:0>	3'b000	
0x06	7 – 0	PDATA<23:16>	0x00	Output Pressure Data. Code = Data0x06*2^16+ Data0x07*2^8+ Data0x08;
0x07	7 – 0	PDATA<15:8>	0x00	
0x08	7 – 0	PDATA<7:0>	0x00	

For example:

If the value of the registers 0x06、0x07、0x08 are 0x3F, 0xFF, 0xFF, according to NSPDS7F001DT45 transfer function, Code = 4194303, P(Pa) = 4194303/8388607*A+B, and finally get the value of pressure about 0kPa.

6.4. I²C Interface

I²C bus uses SCL and SDA as signal lines. Both lines are connected to VDD externally via pull-up resistors so that they are pulled high when the bus is free. The I²C device address of NSPDS7 is shown below.

Table 6.2 I²C Address

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	1	1	1	1	1	0/1

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.

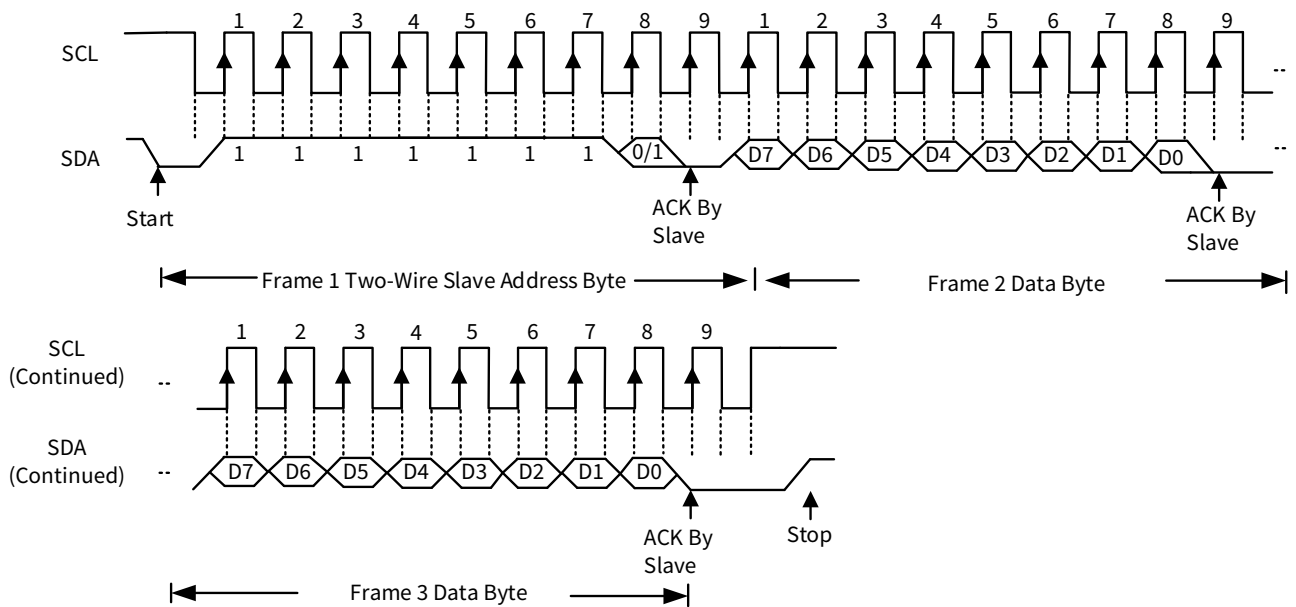


Figure 6.3 I²C Protocol

Byte Write

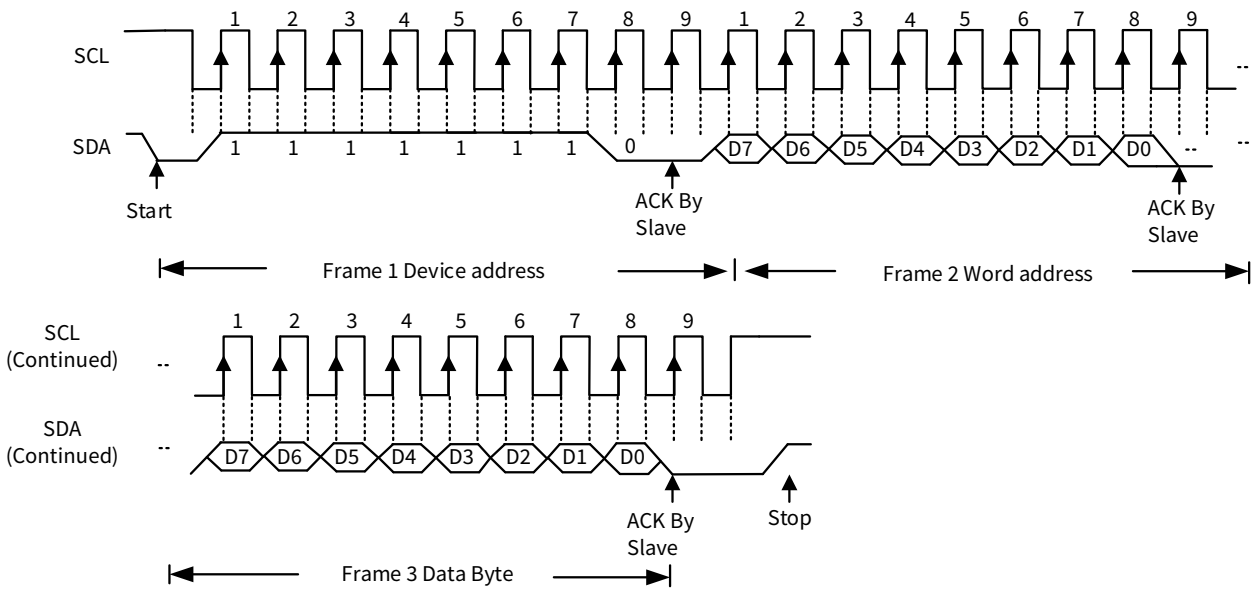


Figure 6.4 I²C Write Byte

Random Read

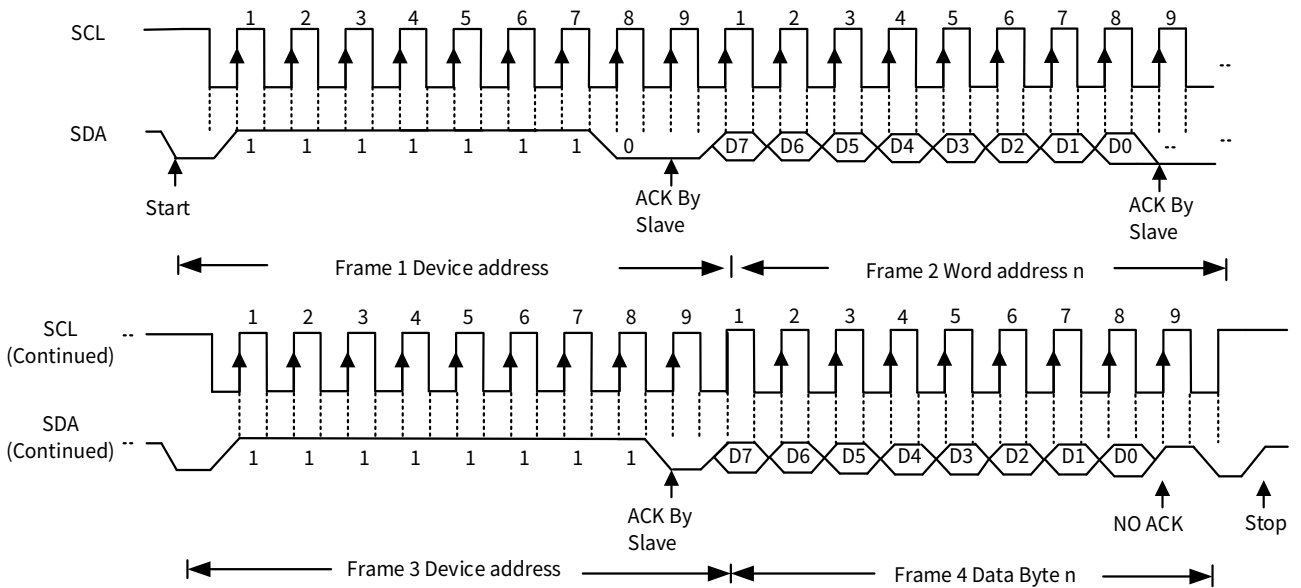


Figure 6.5 I²C Read Byte

7. Typical Application

7.1. Application Circuit

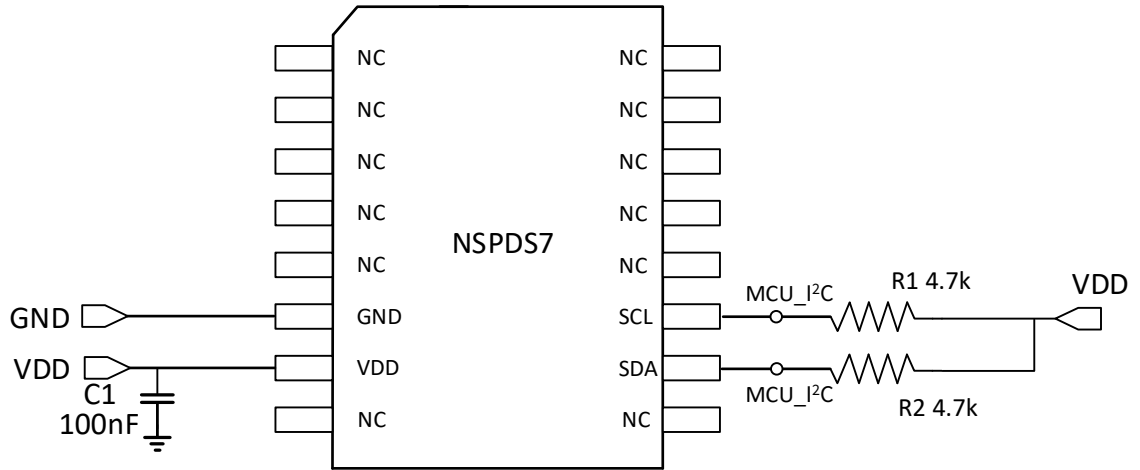


Figure 7.1 I²C Output Application Circuit

8. Package Information

8.1. Package Size

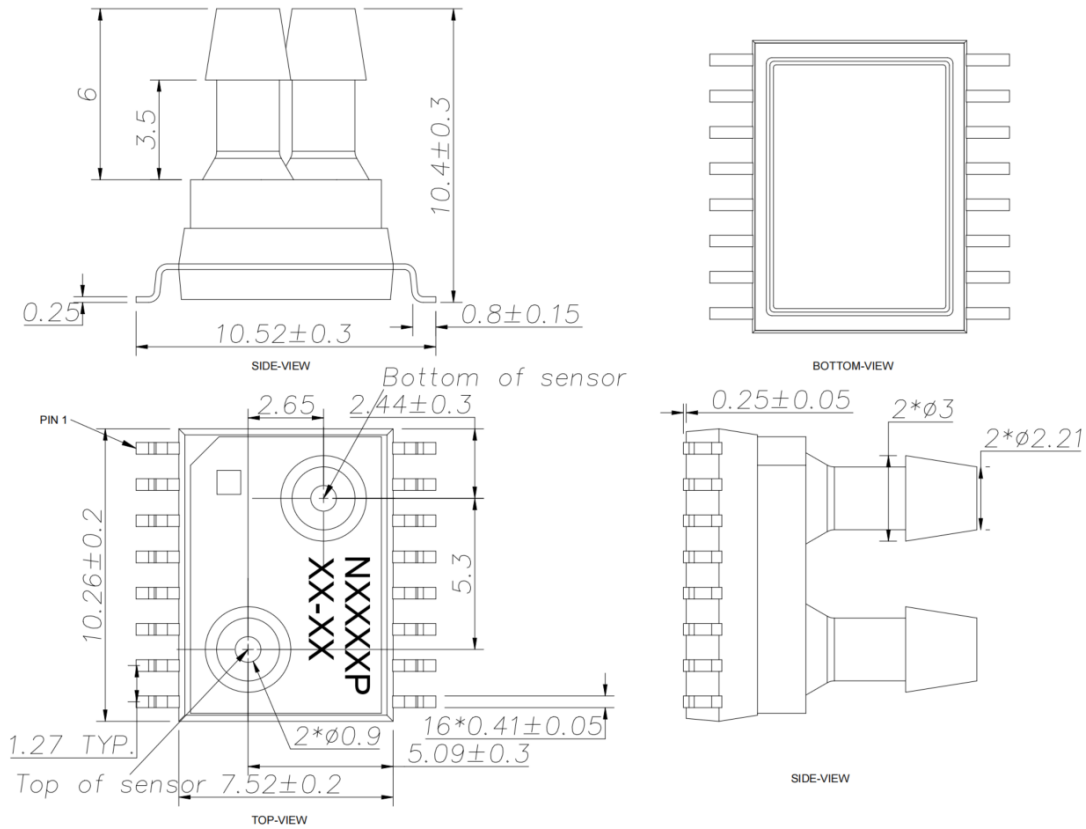


Figure 8.1 SOIC16 Package Outline mm

1. Top of sensor is tube connected to top side of sensor die. Topside pressure is positive pressure. An increase in topside pressure will result in a increase in sensor output.
2. Bottom of sensor is tube connected to bottom side of sensor die.

10. Soldering Parameters

10.1. Reflow Soldering (SMD Terminal)

Table 10.1 Soldering Parameters

Reflow Condition		Lead-free assembly
Pre Heat	Temperature Min (Ts(min))	150°C
	Temperature Max (Ts(max))	180°C
	Time (min to max) (ts)	60 – 150 secs
Average ramp up rate (Liquidus Temp (TL) to peak)		2°C/second max
TS(max)to TL – Ramp-up Rate		2°C/second max
Reflow	Temperature (TL) (Liquidus)	210°C
	Time (min to max) (tL)	60 – 120 seconds
Peak Temperature (TP)		240°C
Time within 5°C of actual peak Temperature (tp)		12 – 30 seconds
Ramp-down Rate		6°C/second max
Time 25°C to peak Temperature (TP)		230 seconds Max.
Do not exceed		240°C

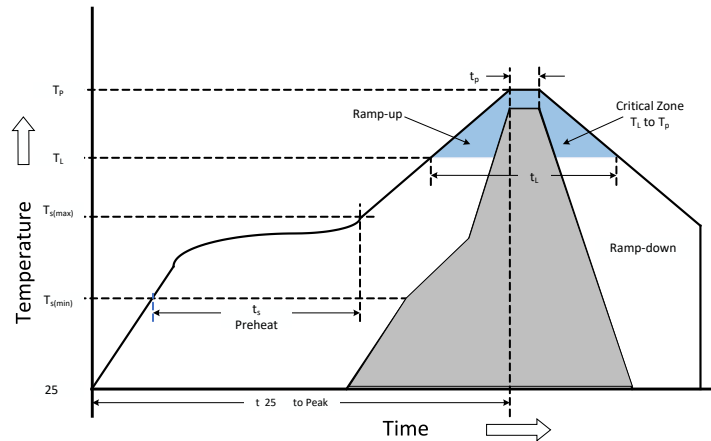
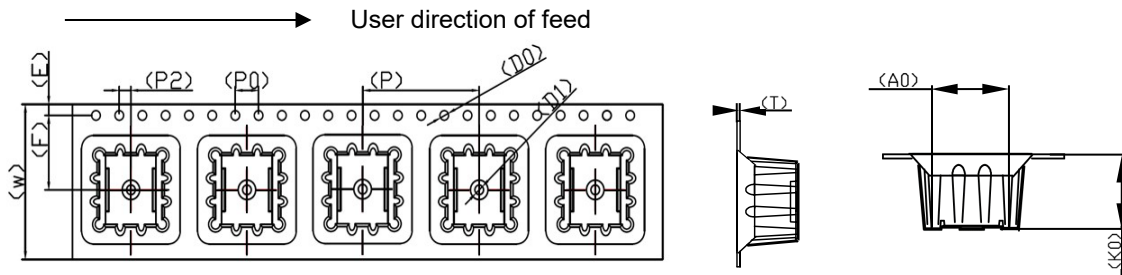


Figure 10.1 Reflow Soldering Curve

10.2. Manual Soldering

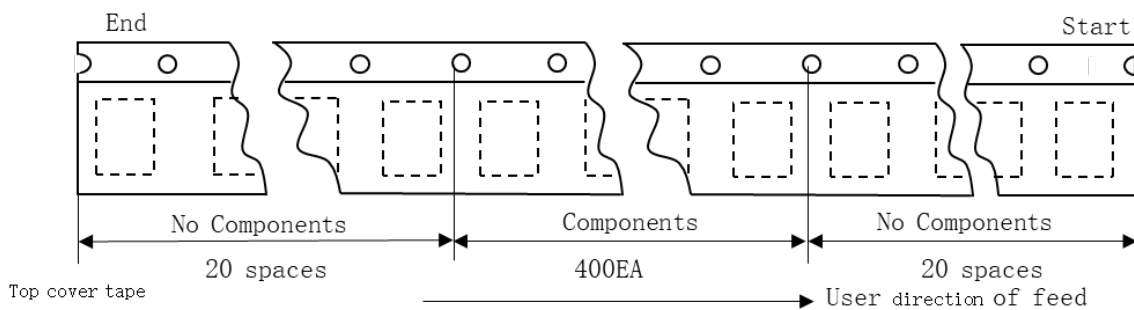
- Raise the temperature of the soldering tip between 260 °C and 300 °C and solder within 5 seconds.
- Use a flattened soldering tip when performing rework on the solder bridge.
- Complete rework in one time.

11. Packing Information

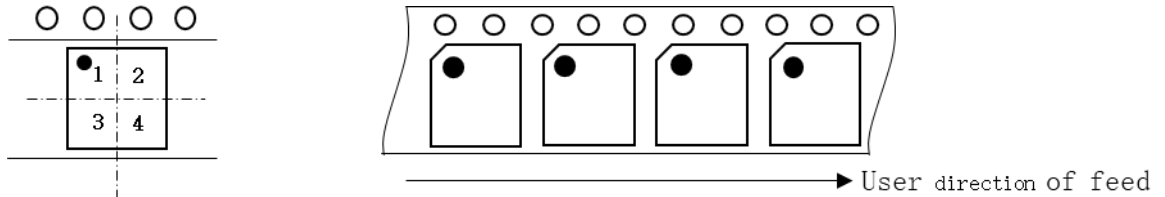


E(mm)	F(mm)	P2(mm)	D0/D1(mm)	P0(mm)	10P0(mm)	W(mm)	P(mm)	A0(mm)	B0(mm)	K0(mm)	T(mm)
1.75±0.10	11.5±0.10	2.0±0.10	1.5±0.1	4.0±0.1	40.0±0.20	24.0±0.30	20.0±0.10	10.80±0.10	10.70±0.10	9.90±0.10	0.50±0.05

There is no component at the head and tail of each tape, with a spacing of 20 spaces, as shown in the following figure.

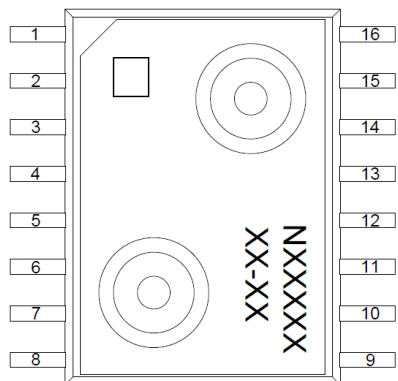


Pin1 is located at the first quadrant, as shown in the following figure.



Minimum ordering quantity (MOQ): 400EA.
Standard pack quantity (SPQ): 400EA.

12. Identification Code



Nxxxxx xx-xx: package date code.

13. Revision History

Revision	Description	Date
1.0	Formal release.	2023/09/27
1.1	Update Packing Information and Package Size.	2025/12/26

14. Notes

14.1. I²C Routine

```

#define ACK    1
#define NACK   0
uchar REG06=0,REG07=0,REG08=0;
uchar number=1;
uchar Reg30[1];
int PCode=0,Pdata=0;
float Pressure=0.0;
void IIC_Start(void)           //Start the I2C, SDA High-to-low when SCL is high
{
    IIC_SCL(1);               //SCL output high level
    SDA_OUT(1);               //SDA output high level
    Delay_us(2);              //Delay 2us
    SDA_OUT(0);               //SDA output low level
    Delay_us(2);
}

void IIC_Stop(void)           //Stop the I2C, SDA Low-to-high when SCL is high
{
    IIC_SCL(0);
    Delay_us(2);
    IIC_SCL(1);
    SDA_OUT(0);
    Delay_us(2);
    SDA_OUT(1);
    Delay_us(2);
}

void IIC_ACK(void)           //Send ACK (LOW)
{
    SDA_OUT(0);
    IIC_SCL(1);
    Delay_us(2);
    IIC_SCL(0);
}

void IIC_NACK(void)          //Send No ACK (High)
{
    SDA_OUT(1);
    IIC_SCL(1);
    Delay_us(2);
    IIC_SCL(0);
}

uchar IIC_Wait_ACK(void)     //Check ACK, if return 0, then right, if return 1, then error
{
    int ErrTime=0;
    SDA_IN();                 //SDA set as input
    IIC_SCL(1);
    Delay_us(2);
    while(Read_SDA)

```

```
{
    ErrTime++;
    if(ErrTime>200)
    {
        IIC_Stop();
        return 1;
    }
}
IIC_SCL(0);
SDA_OUT(0);
Delay_us(2);
return 0;
}

void IIC_Send(uchar IIC_Data)           //Send a byte to I2C
{
    uchar i;
    IIC_SCL(0);
    Delay_us(2);
    for(i=0;i<8;i++)
    {
        if((IIC_Data&0x80)>>7)
            SDA_OUT(1);
        else
            SDA_OUT(0);
        IIC_Data<<=1;
        IIC_SCL(1);
        Delay_us(2);
        IIC_SCL(0);
        Delay_us(2);
    }
}

uchar IIC_Receive(uchar ACK)           //Receive a byte from I2C
{
    uchar i,Receive_Data=0;
    SDA_IN();
    for(i=0;i<8;i++)
    {
        IIC_SCL(0);
        Delay_us(2);
        IIC_SCL(1);
        Receive_Data<<=1;
        if(Read_SDA==1)
            Receive_Data++;
        Delay_us(2);
    }
    IIC_SCL(0);
    Delay_us(2);
    if(ACK==0x01)
        IIC_ACK();
    else
        IIC_NACK();
    return Receive_Data;
}
```

```
}
```

```
void NSPDS7F001DT45_Write_Byte(uchar WriteAddr,uchar WriteData)
```

```
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(WriteAddr);
    IIC_Wait_ACK();
    IIC_Send(WriteData);
    IIC_Wait_ACK();
    IIC_Stop();
}
```

```
void NSPDS7F001DT45_Read_Byte(uchar ReadAddr, uchar *pBuffer)
```

```
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(ReadAddr);
    IIC_Wait_ACK();
    IIC_Start();
    IIC_Send(0xFE|0x01);
    IIC_Wait_ACK();
    pBuffer[0]=IIC_Receive(0);
    IIC_Stop();
}
```

```
void NSPDS7F001DT45_Read_3Byte(uchar ReadAddr,uchar *pBuffer)
```

```
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(ReadAddr);
    IIC_Wait_ACK();
    IIC_Start();
    IIC_Send(0xFE|0x01);
    IIC_Wait_ACK();
    pBuffer[0]=IIC_Receive(ACK);
    pBuffer[1]=IIC_Receive(ACK);
    pBuffer[2]=IIC_Receive(NACK);
    IIC_Stop();
}
```

```
void main()
```

```
{
    uchar PData[3]={0,0,0};
    while(1)
    {
        NSPDS7F001DT45_Write_Byte(0x30,0x0A);
        while(1) //Check whether the conversion ends
        {
            if(number<=50)
            {
```

```
        number++;
        delay_ms(1);
        NSPDS7F001DT45_Read_Byte(0x30,Reg30);
        if(0x02==Reg30[0])
        {
            number=1;
            break;
        }
    }
    if(number>50)
    {
        number=1;
        //User can add his own error handler function
        break;
    }
}
NSPDS7F001DT45_Read_3Byte(0x06,PData);
REG06 = PData [0];           //Register 0x06
REG07 = PData [1];           //Register 0x07
REG08 = PData [2];           //Register 0x08
PCode=(REG06*65536+REG07*256+REG08); //PCode = Data0x06*2^16+ Data0x07*2^8+Data0x08
if (PCode >8388607)
    Pdata= PCode-16777216;    //Symbol processing
else
    Pdata= PCode;
Pressure = float (1.25*Pdata/8388607-0.625); //P=A*PCode/8388607+B
//A=1.25, B= -0.625
//PNormalized=PCode/8388607
}
}
```

IMPORTANT NOTICE

The information given in this document (the “Document”) shall in no event be regarded as any warranty or authorization of, express or implied, including but not limited to accuracy, completeness, merchantability, fitness for a particular purpose or infringement of any third party’s intellectual property rights.

Users of this Document shall be solely responsible for the use of NOVOSENSE’s products and applications, and for the safety thereof. Users shall comply with all laws, regulations and requirements related to NOVOSENSE’s products and applications, although information or support related to any application may still be provided by NOVOSENSE.

This Document is provided on an “AS IS” basis, and is intended only for skilled developers designing with NOVOSENSE’s products. NOVOSENSE reserves the rights to make corrections, modifications, enhancements, improvements or other changes to the products and services provided without notice. NOVOSENSE authorizes users to use this Document exclusively for the development of relevant applications or systems designed to integrate NOVOSENSE’s products. No license to any intellectual property rights of NOVOSENSE is granted by implication or otherwise. Using this Document for any other purpose, or any unauthorized reproduction or display of this Document is strictly prohibited. In no event shall NOVOSENSE be liable for any claims, damages, costs, losses or liabilities arising out of or in connection with this Document or the use of this Document.

For further information on applications, products and technologies, please contact NOVOSENSE (www.novosns.com).

Suzhou NOVOSENSE Microelectronics Co., Ltd