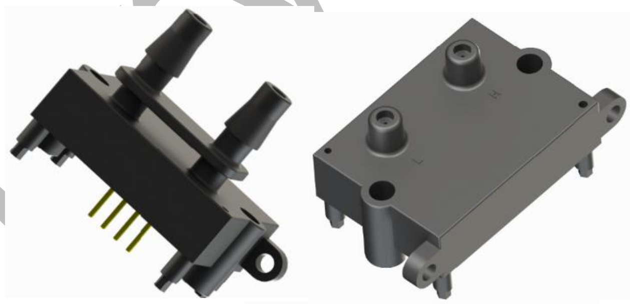
Differential Pressure Sensor

MDP200 Series

Features

- *Pressure range up to ±500Pa with high accuracy of ±3.0% m.v.*
- *Pressure based on thermal micro-flow measurement*
- *Outstanding hysteresis and repeatability*
- *Linearized and temperature compensated*
- *Digital I²C with 16bit resolution*
- *Cost effective*
- *RoHS and REACH compliant*
- *Minimum detectable pressure difference as low as 0.02 Pascal*



Applications

- *Medical CPAP and ventilator*
- *HAVC and building*
- *Burner control*
- *Filter monitoring*
- *Process control and automation*

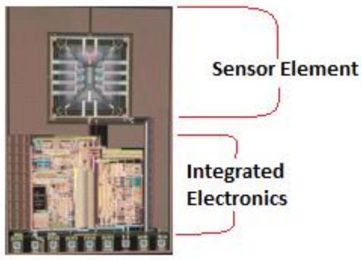
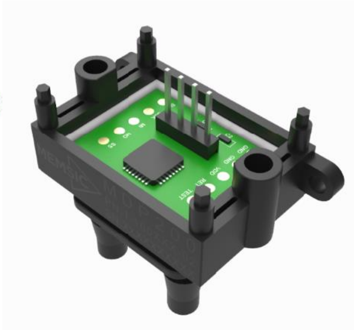


Image of flow sensor



General Description

ACEINNA's MDP200 series MEMS differential pressure sensors measure ultra-low gas pressures covering the range of up to ±500Pa (±2 inH2O). The technology is based on ACEINNA's highly successful proprietary CMOS thermal accelerometers already sold in millions. ACEINNA's thermal flow sensing element is monolithically integrated with CMOS signal processing circuitry and embedded software capable of converting gas flow rates to a digital format. The signal is linearized and temperature compensated. MDP200 series offers incredible sensitivity detecting pressure down below 0.02 Pascal near zero differential pressure. Other features include wide dynamic range, superb long-term stability, and outstanding repeatability and hysteresis.

1. Performance[1]

Parameter	Condition	Min.	Typ.	Max.	Unit
Measurement Range	Air/N2			+/-500	Pa
Zero-point Accuracy[2]				+/-0.08	Pa
Span Accuracy[2]				+/-3.0%	m.v
Total Error (0ºC~50ºC)				±3.5%	m.v
Zero-point Repeatability and Hysteresis[2]				+/-0.08	Pa
Resolution (Near Zero)/Lowest Detectable Pressure		0.02			Pa
Sampling Rate				8	ms
Span Repeatability and Hysteresis[3]				0.50%	m.v
Over Pressure				3	Bar
Span Shift due to Temperature Variation				0.05	%m.v.perºC
Zero Point Variation with Temperature	0-50ºC			0.08	Pa
Offset Stability			TBD		
Non-Linearity(BFSL)	0-100 Pa			0.30%	FS
	0-200 Pa			0.40%	FS
	0-500 Pa			0.80%	FS
Orientation Sensitivity	port@90º vs. 270º			0.02	Pa
	port@90º vs.180º			0.05	Pa
Gas Flow Through Sensor[3]	500 Pa		100		ml/min

1) All sensor specifications are valid with air as medium at 3°C temperatures with 1 standard atmospheric pressure (101325Pa), 50% RH, and a 3.3V DC power supply, unless otherwise specified. Customized versions are available, please contact factory for calibration under other conditions of pressure ranges, temperatures, and gases
2) Accuracy specifications apply over operating conditions. With 16-bit resolution, this accuracy represents the total of non-linearity, hysteresis, zero and span shift, repeatability and temperature effects.
3) MDP200 operates based on thermal mass flow principle. Gas flow is required to measure the pressure difference.

2. Environment

Parameter	Value
Operating Temperature	-20°C to +80°C
Storage Temperature	-40°C to +85°C
Relative Humidity	To 95% (Non-Condensing)
Radiated Susceptibility	3V/m
ESD	4kV contact, 8kV Air
Free Drop	Flat concrete ;1m;10 times
Shock	100g; half-sine; 6ms;3 Axes, total 18 times of shock
Vibration (5-2000 Hz)	Random vibration test at 20-2000 Hz, 3Grms; 2 axis of X and Y; no power
Calibrated gases	Air/N2
Media Compatibility	N_2, O_2, Air
Barb Strength	4, lbf (3 orthogonal directions)

3. Electrical

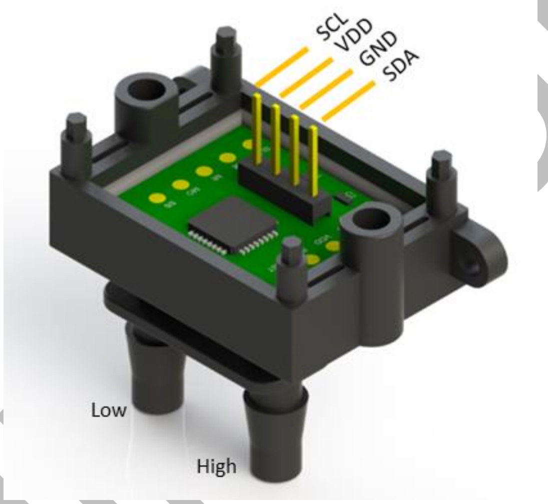
Parameter	Value
Input Voltage Range	3.0-3.6 Vdc (3.3V is recommended)
Supply Current	<8 mA
Interface	I^2C
Resolution	16 Bits (Bi-direction)
Bus Clock Frequency	100KHz Typical, 400KHz Max
I2C Default Address	0x31

4. Material

Parameter	Description
Wetted Material	FR4, Silicon Nitride, Silicon Oxide, Silicon, Gold, Epoxy, PBT (polybutylene terephthalate)
Standard Compliant	RoHS and REACH

5. I^2C Interface

Pinout Configuration

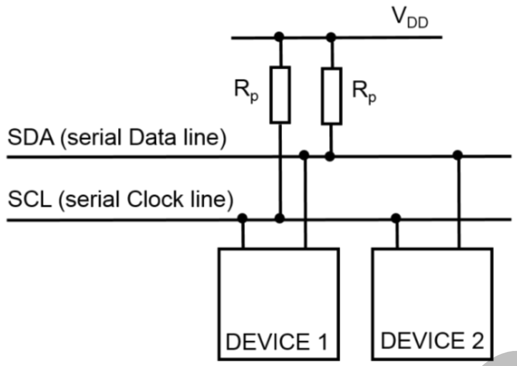


Pin	Name	Description
1	SCL	Serial Clock Line for I^2C bus
2	VDD	Power Supply
3	GND	Ground
4	SDA	Serial Data Line for I^2C bus

WARNING: Reversed or misaligned connection will cause permanent damage.

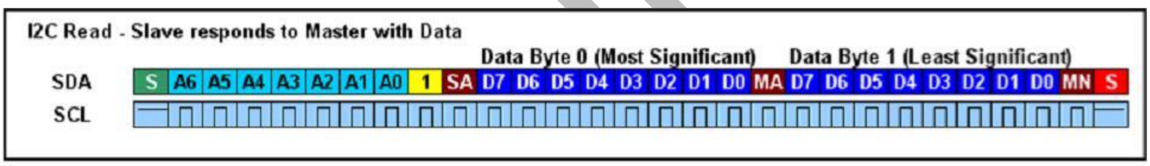
5.1 External Interface

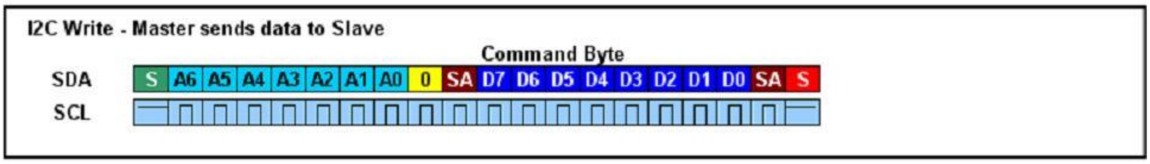
Each SCL and SDA line must be connected to V_{DD} with about 4.7k Ohm pull-up resistor as shown below. The MDP200 has a V_{DD} input range of 3.0-3.6V, but recommended VDD is 3.3V (calibration condition).



5.2 I²C Read and Write Timing

MDP200's communication interface is Phillips I²C compatible as shown below. the recommended frequency of SCL line is approximately100 kHz.





Key

Bit	Name	Description
S	Start Condition	Master pulls SDA from high to low while SCL remains high
S	Stop Condition	Master allows SDA to float from low to high while SCL remains high
A6	Address Bit	I2C Slave Address is the 7 Most Significant bits of the first transmitted byte
1	Read/Write Bit	Read = 1, Write = 0
D7	Data Bit	
SA	Slave ACK	Slave pulls SDA low
MA	Master ACK	Master pulls SDA low
MN	Master NACK	Master allows SDA to float high

5.3 Register Map

Every register includes two bytes. The register map is defined as the table below.

Register	Byte Number	Data Definition
Register 0	Byte 0	Differential Pressure High Byte
	Byte 1	Differential Pressure Low Byte
Register 1	Byte 2	CRC for Register 0
	Byte 3	NA
Register 2	Byte 4	NA
	Byte 5	NA
Register 3	Byte 6	NA
	Byte 7	NA
Register 4	Byte 8	NA
	Byte 9	NA
Register 5	Byte 10	NA
	Byte 11	NA
Register 6	Byte 12	NA
	Byte 13	NA
Register 7	Byte 14	NA
	Byte 15	NA
Register 8	Byte 16	Temperature Sensor Count High Byte(Tcount_Hi)
	Byte 17	Temperature Sensor Count Low Byte(Tcount_Lo)

5.4 Command List For Sending To Slave

Table 2 Command List for Sending To Slave

No	Command Name	Data Frame Of Command			Byte Number
1	Trigger Single Measurement	0xC1			1
2	Set Slave Address	0xF6	1-byte i2c_slave_address	1-byte random number	3
3	Set Window Averaging Time	0xFA	1-byte average time	1-byte random number	3
4	Reset MCU	0xFE			1

Note:

1) Byte number in Table 2 is the number of data bytes slave received, for every command, byte number must be correct, otherwise, command will not be implemented
2) The range of I2C slave address is from 0 to 127. If the address has received the value beyond the range, it will be set as default address 0x31
3) The range of average time is from 1 to 64, if the time received beyond the range, it will set as 1 default average time 1
4) Random number means varying data value works

Command Example

0xC1: MDP200 will measure once immediately
0xF6 0x04 0x00: I2C slave address will be changed to 0x04 immediately
0xFA 0x02 0x00: Window averaging time will be changed to 0x02 immediately
0xFE: MCU will reset immediately

5.5 Trigger measurement operation

MDP200 works in slave mode. The default 7-bit address is 0x31 with a following bit either write bit(0) or read bit(1).

To trigger a measurement, the master writes a command 0xC1 to slave (sensor), then wait at least 7ms. The master can read out N bytes data from the register address 0.

To read differential pressure data, the first byte transmitted must be 0x63 which indicates that master will read from sensor whose slave address is 0x31. Immediately with master generating pulses of N bytes, the master can read out the N bytes data.

For example, the master triggers the measurement first; and then read out the data in register 0. There are following three steps:

Step 1: Master write into sensor the trigger command 0xC1.

Slave address + write bit	Measurement trigger command
0x62	0xC1

Step 2: Wait at least 7ms.

Step 3: Master reads differential data

Slave address + read bit	N Bytes
0x63	Byte 0 – Byte N-1

Once the slave receives the address and reads bit (0x63) from master, slave will return N bytes data as shown in the register table in Section 5.4.

With the pulses of N bytes generated by master, slave will return N bytes data.

Note:

1) Slave returns a high byte first, then the low byte, and at last the CRC code
2) Trigger measurement command is recommended to be sent after data reading is completed.
3) N is number of bytes which master reads from slave, as for how many bytes master read, it is not constant and is from 1 to 18, if only differential pressure and CRC code are read, N should be at least 3, if temperature sensor output are read, N should be at least 18.

5.6 Data Format

The output data from slave's register 0 is 2 bytes signed integer data. If the data is divided by 50, the result will be the differential pressure which is from -655 to 655 Pa.

The output data from slave's register 8 is 2 bytes unsigned integer data, it is raw count output of temperature sensor. It can be converted to the real temperature (Unit: ℃) using the formula below:

Equation (1) Tcount = Tcount_Hi × 256 + Tcount_Lo

Equation (2) Temperature = 0.2926 × Tcount - 124.7073

For example, while Tcount_Hi = 0x01, Tcount_Lo = 0xFF, Temperature ≈ 24.8ºC

Note:
'0x' mean the number is hexadecimal.

5.7 Reset Command

MDP200's circuit can be reset to initial status by writing reset command 0xFE into device.

Note:
The device will not work normally until 2 seconds after writing reset command. Do not write any data into device within 2 seconds after writing reset command or there will be unpredictable error.

5.8 CRC-8 Redundant Data Transmission

MDP200 use cyclic redundancy checking (CRC) technique for error detection in I2C transmission. The master appends an 8-bit checksum to the actual data sequence. The checksum holds redundant information about the data sequence and allows the receiver to detect transmission errors. The computed checksum can be regarded as the remainder of a polynomial division, where the dividend is the binary polynomial defined by the data sequence and the divisor is a "generator polynomial". The MDP200 implements the CRC-8 standard based on the generator polynomial $x^8 + x^5 + x^4 + 1$.

The master's program must use the 2 bytes differential pressure reading from MDP200's register 0 and polynomial $x^8 + x^5 + x^4 + 1$ to calculate out the one byte result data. Then compare the result data calculated by master and the CRC code from MDP200. If they are not equal, it indicates that there is error in I2C communication. The master must discard the measurement data and trigger measurement again, then read from MDP200's register 0 until the result data and the CRC code are equal. Appendix at the end of this document is C code for reference.

Note:
CRC is only used for data transmitted from slave to master. For the details of the cyclic redundancy check, please refer to the relevant literature.

6 Altitude Correction

The MDP200 series utilizes a thermal principal to measure pressure difference to achieve high sensitivity, robustness and stability. Changes in altitude from the calibration condition (sea level) require output adjustment as shown below. Air pressure above sea level can be calculated as:

$$P = 101325(1 - 2.25577 \times 10^{-5}h)^{5.25588}$$

where

P is air pressure (Pa)

h is altitude above sea level

Example:

Air pressure at Elevation 1000 meters.

The air pressure altitude 1000meters can be calculated as

$$P = 101325(1 - 2.25577 \times 10^{-5} \times 1000)^{5.25588}$$

$$= 26436(\mathrm{Pa})$$

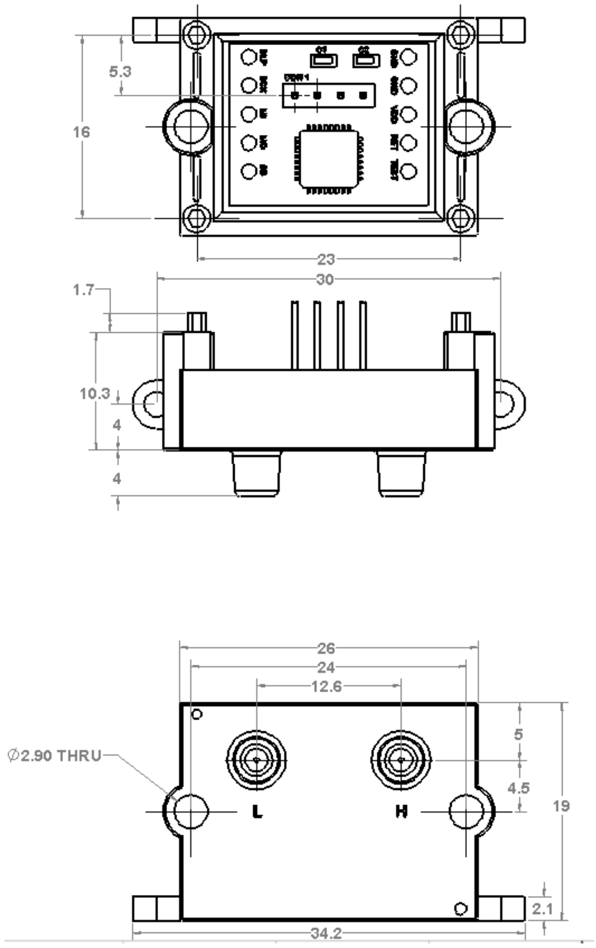
Altitude (meters)	Correction Factor
0	1.00
250	1.03
425	1.05
500	1.06
750	1.09
1000	1.13
1500	1.20
2000	1.27
3000	1.44

6. Effects on Hose Lengths

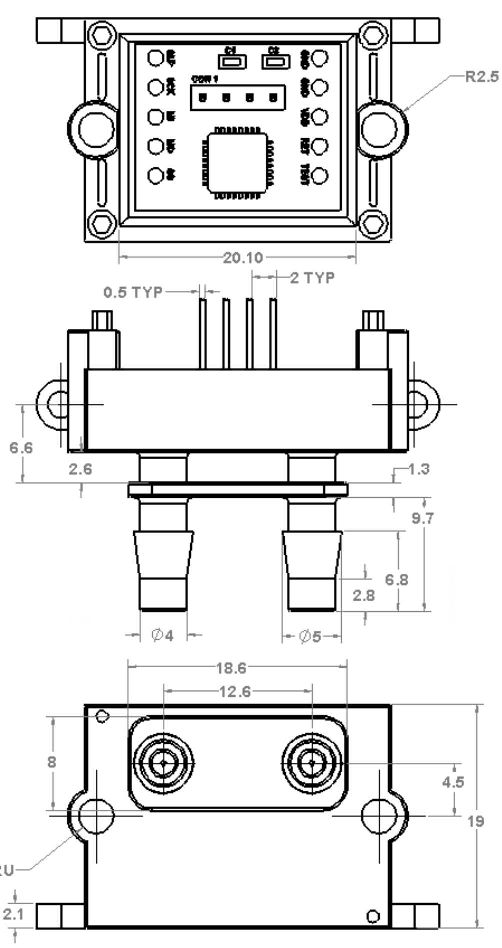
Since the MDP200 series utilizes a thermal measurement principal with air flowing through the sensor, long tubing length has an impact to the sensor output due to frictional losses. The amount of impact depends on the hose material, internal diameter and total length leading to and away from the sensor. In general, tubing length shorter than 1 meter has less 1% (m.v.) impact. Refer to the application notes on tubing length effect of MDP200 series for details.

7 Mechanical Specifications

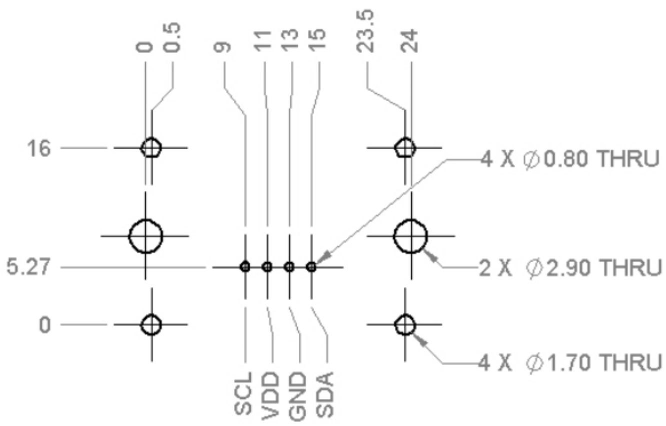
MANIFOLD MOUNT VERSION



BARB FITTING VERSION



8 Foot Print



9 Ordering Information

Options	Ranges		Calibration		Housing	
MDP200	**-500**	500Pa	**B**	Bi-Directional	T	Manifold
			U	Uni-Directional	Y	Barb

Example: MDP200-500BY = MDP200 differential pressure sensor, 500 Pascal, Bi-Directional, Barb Fitting

10 Appendix

Master's CRC calculation C code

```
#define   POLYNOMIAL  0x131 // P(x) = x^8 + x^5 + x^4 + 1 = 100110001
unsigned char CheckCrc(unsigned char data[], unsigned char nbrOfBytes)
{
  unsigned char crc = 0;
  unsigned char byteCtr;
  //calculates 8-Bit checksum with given polynomial P(x) = x^8 + x^5 + x^4 + 1 =
100110001
  for (byteCtr = 0; byteCtr < nbrOfBytes; ++byteCtr)
  {
        crc ^= (data[byteCtr]);
        for (unsigned char bit = 8; bit > 0; --bit)
        {
              if (crc & 0x80)
                    crc = (crc << 1) ^ POLYNOMIAL;
              else crc = (crc << 1);
        }
  }
  return crc;
}
```

Revision History

Date	Author	Version	Changes
June 2017	O. Silpachai	1.0	Initial Release
July 2017	J. Pern	1.1	Revised Zero-Point Accuracy and Repeatability. Improvement of Null DP test setup (H and L ports common)
Sept 2017	J. Pern	1.2	Updated I^2C Communication section and foot print section
Mar 2018	SY. Xiao	1.3	Updated I^2C Communication section; Added the CRC check