Application Note AN-15
I2C Protocol

This Application Note applies to the following
APSP pressure sensors:

89DL 102A 28P0

Version 1.0

1-Mar-2022

Address: 0x28

The Sensor Module includes an I²C digital, two-wire interface with a bidirectional data line (SDA) and a clock line (SCL). The two lines are open drain and connected to the supply voltage via two pull-up resistors (Rp). In a system with master-slave configuration, the Sensor Module is the slave.

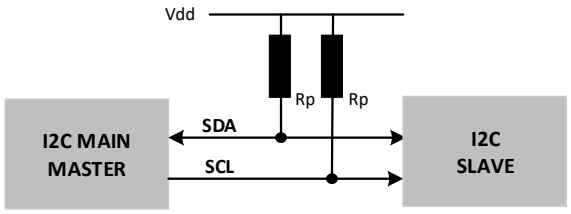

Figure 1: *I2C master-slave configuration*

The recommended pull-up resistor (Rp) values depend on the system implementation, but a value between 2.2kΩ and 10kΩ can be used for prototyping.
The capacitive load on both SDA and SCL should be the same, hence the signal lengths should be similar to avoid asymmetry. It is recommended to use shielded cabled for wire lengths above 10cm and I²C buffers should be used if signal paths are longer than 30cm.

- Read Example(Full Measurement):

Byte#		0									1										2								
Send By Master		0x50									0xAA										0x51								
	S	0	1	0	1	0	0	0	0	A	1	0	1	0	1	0	1	0	A	P	S	0	1	0	1	0	0	0	1	A
		Address(0x28)							W		Command											Address(0x28)							R	

Byte#	0		1		2		3		4		5		6		
Receive From Slave		A	[23:16]	A	[15:8]	A	[7:0]	A	[23:16]	A	[15:8]	A	[7:0]	A	P
	Status		Pressure data						Temperature data						

- S : Start bit
- P : Stop bit
- A : ACK
- W: i2c write mode
- R: i2c read mode
- General Status byte

ALU : arithmetic logic unit

Bit #	7	6	5	4	3	2	1	0
Meaning	0	Power?	Busy?	Mode		Memory Error?	Connection Check Fault?	Math Saturation?

```
// Using Arduino “Wire.h” library to access sensor
int num = 7; // to read 7 bytes from response
define ADDR =0x28;  //as default slave address
byte Writebyte[1] = {0xAA}; // write command
byte Readbyte[7]; // read buffer

void i2c_command(int num) {
    Wire.beginTransmission(ADDR); // transmit to device #0x28
    Wire.write(Writebyte, 1);       // sends five bytes
    Wire.endTransmission();    // stop transmitting
    delay(100);
    Wire.requestFrom(ADDR, num);
for (int i = 0; i < num; i++) {
        Readbyte[i] = 0;
        if (Wire.available()) { // slave may send less than requested
           Readbyte[i] = Wire.read(); // receive a byte as character
        }
       else {
         Readbyte[i] = 0;
       }
    }
}
```

Limitations

The I²C bus is susceptible to noise and can lock up, especially if there are glitches on SCL or the Master does not acknowledge the first byte sent from the Slave.

The following guidelines are best practices for the I²C bus and to avoid lock up:

- Minimize signal length between sensor and microcontroller (< 30cm). Signal lengths over 10cm should be shielded
- Every data read from a Slave should be acknowledged by an ACK from the Master
- It should be possible to hard-reset the sensor should the I²C bus lock up

Angst+Pfister
Sensors and Power

We are here for you. Addresses and Contacts.

Headquarter Switzerland:

Angst+Pfister Sensors and Power AG
Thurgauerstrasse 66
CH-8050 Zurich
Phone +41 44 877 35 00
sensorsandpower@angst-pfister.com

Office Germany:

Angst+Pfister Sensors and Power Deutschland GmbH
Edisonstraße 16
D-85716 Unterschleißheim
Phone +49 89 374 288 87 00
sensorsandpower.de@angst-pfister.com

Scan here and get an overview of personal contacts!

sensorsandpower.angst-pfister.com