I²C-protocol for models MPR-1 and MTF-1

Version 3.0

WIKA

Part of your business

Content

1. I²C general

1.1 Description

I²C is a simple, serial 8-Bit data bus that was developed for short distance communication between PCBs.
It is not a field-bus, so long wires from the sensor to the master electronics are not recommended. The maximum wire length is to be defined based on the surrounding during usage and the according disturbances.

The pressure sensor module works as a slave. The I²C master communicates with the sensor using its I²C-address. The address is set as "0" as a standard but can be adapted customer specifically and can also be changed by the customer.

The I²C bus requires in its most simple variation only two lines additional to the power supply:

- SCL (Serial Clock)
- SDA (Serial Data)

1.2 Configuration

Pull-Up resistors (Rp) are needed at the SDA and SCL line.
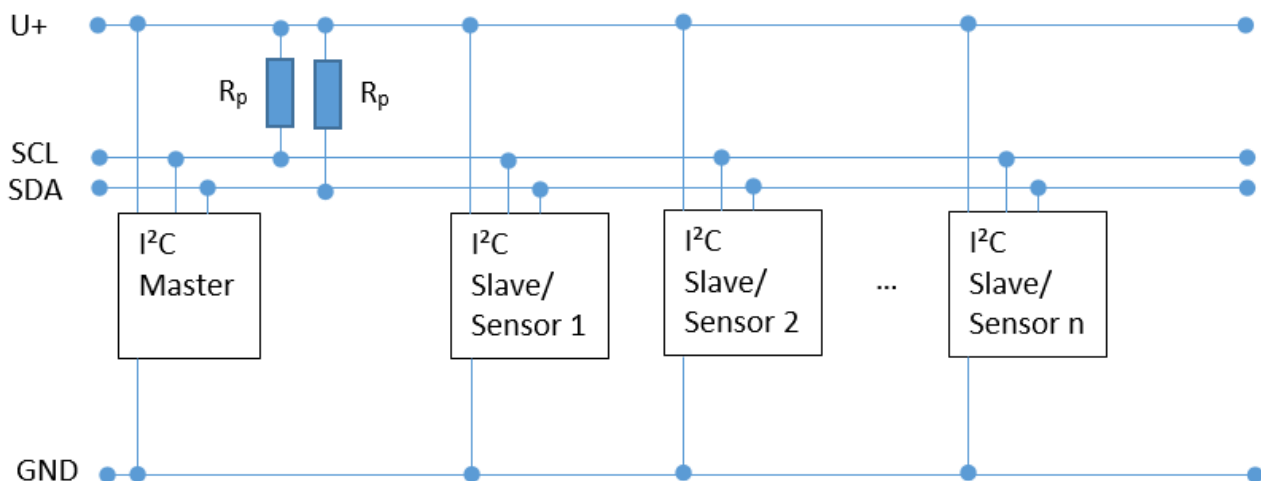
An additional End of Conversion (EOC) Pin provides the possibility to detect the end of the pressure measurement.

2. I²C-protocol of the pressure sensor module

The I²C signal of the pressure sensor module provides the possibility to read out both pressure and temperature values.
Using the status bit a diagnosis function is provided.

2.1   Parameters

	Min	Max	Unit
V_{IL} (Input low level voltage)	-0,3	0,9	V
V_{IH} (Input high level voltage) for 2,3V ≤ U+ ≤ 3V	U+ -0,1	U+ +0,5	V
V_{IH} (Input high level voltage) for U+ > 3V	2,9	3,5	V
V_{OL} (Output low level voltage)		0,45	V
Clock Frequency	0,01	3,4	MHz
Pull-up-resistors each at SDA and SCL	1,5	10	kΩ

2.2   Current-EOC-time graph

Default settings with Command Request (Oversampling = 1)
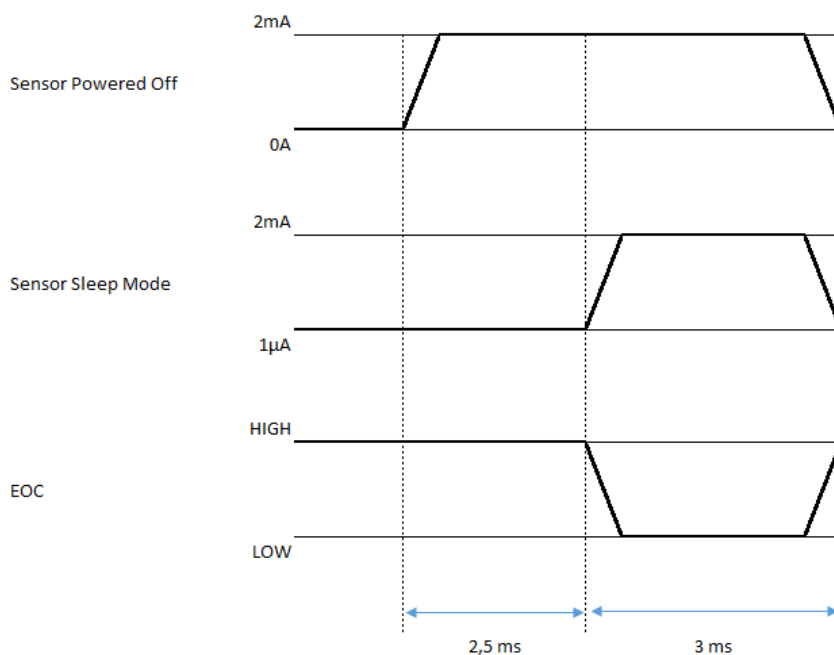
Time until the pressure and temperature value is available after the request by the I²C master:

- using Sleep-Mode: ~3ms
- complete power-off: ~5,5ms

I²C-protocol of the pressure sensor module

Optional settings with Command Request (Oversampling = 4)

Oversampling = 4 is required for pressure sensor module model MTF-1 in case the optional version with accuracy of 0,25% of span is used. See datasheet PE83.01.
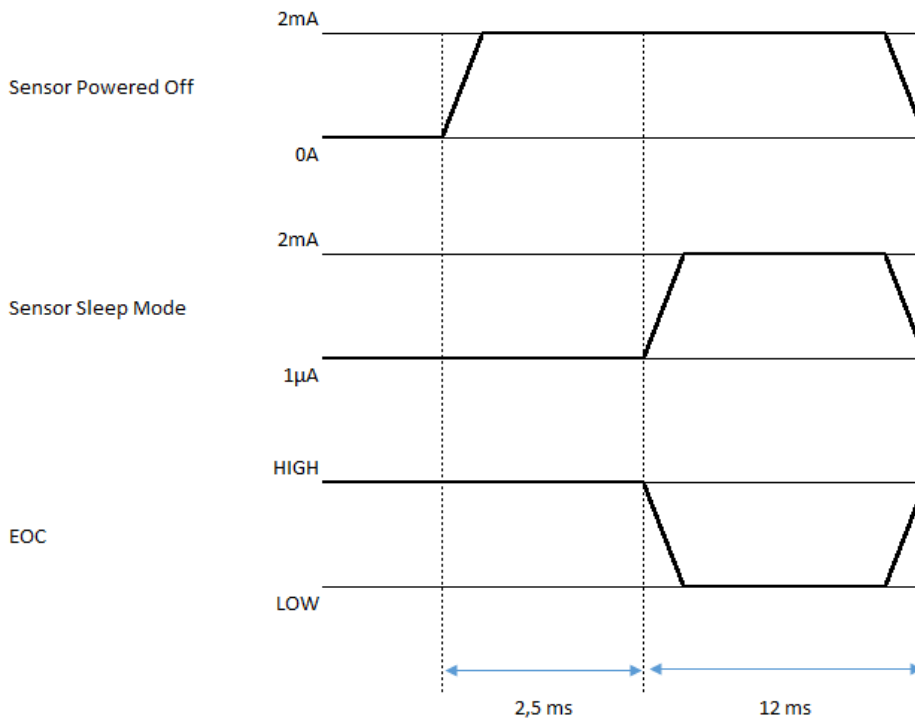

Time until the pressure and temperature value is available after the request by the I²C master:

- using Sleep-Mode: ~12ms
- complete power-off: ~14,5ms

Communication Services

2.3   I²C address

The pressure sensor module works as I²C-slave and has to be addressed and controlled by an I²C master.

Preset Standard-address: 0
Possible addresses that can be set: 0…3, 8…127
                                            (4…7 are reserved and must not be used)

3.   Communication Services

3.1   Reading pressure and temperature value

Command Request (Oversampling = 1):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	0xAA (1 Byte)
Master	**Master**

optional Command Request (Oversampling = 4):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	0xAD (1 Byte)
Master	**Master**

Option 1:
Wait time: ≥ 3ms with Oversampling 1
Wait time: ≥ 12ms with Oversampling 4

Option 2: (recommended)
Use EOC-pin, see **"3.6 Using the EOC-pin"**

Command Response:

Slave-Address << 1 + 1 (Read-Bit) (1 Byte)	Status (1 Byte)	PD-pressure Bits <23:16> (1 Byte)	PD- pressure Bits <15:8> (1 Byte)	PD- pressure Bits <7:0> (1 Byte)
Master	**Slave**	**Slave**	**Slave**	**Slave**

PD-temperature Bits <23:16> (1 Byte)	PD- temperature Bits <15:8> (1 Byte)	PD- temperature Bits <7:0> (1 Byte)
Slave	**Slave**	**Slave**

Slave-Address: set 7-Bit Slave-Address

PD-pressure: PD - pressure >> 6   → Interpretation of PD-pressure: MBA…MBE = 50.000…250.000 digits

PD-temperature: PD - temperature >> 6   → Interpretation of PD-temperature: -45…+110°C = 0…262.143 digits

The master can stop the data transmission after 4 byte in case the temperature sensor is not required.

Calculation of the pressure value:
([unit] = bar, MPa or psi)

Sensitivity: $\qquad S = \frac{MBE[digits] - MBA[digits]}{MBE\,[unit] - MBA[unit]}$

Pressure: $\qquad p = \frac{p\,[digits] - MBA\,[digits]}{S} + MBA[unit]$

Example:
A Sensor with pressure range 0…25 bar is used. Digital pressure value: 125000 digits.

$$S = \frac{MBE[digits] - MBA[digits]}{MBE\,[unit] - MBA[unit]} = \frac{250000\,digits - 50000\,digits}{25\,bar - 0\,bar} = 8000\,digits/bar$$

$$p = \frac{p\,[digits] - MBA\,[digits]}{S} + MBA[unit] = \frac{125000\,digits - 50000\,digits}{8000\,digits/bar} + 0\,bar = 9{,}375\,bar$$

Calculation of the temperature value:

Sensitivity: $\qquad S = \frac{MBE[digits] - MBA[digits]}{MBE\,[unit] - MBA[unit]} = \frac{262143\,digits - 0\,digits}{110°C - (-45°C)} \approx 1691\ \text{digits/°C}$

Temperature: $\qquad t = \frac{t\,[digits] - 0}{S} + (-45°C) = \frac{t\,[digits]}{1691{,}25\,digits/°C} - 45°C$

Example:
A Sensor delivers hex-value 112500 as temperature.

$$t = \frac{t\,[digits]}{1691\,digits/°C} - 45°C = \frac{112500\,digits}{1691\,digits/°C} - 45°C = 21{,}5°C$$

3.2 Interpretation status-byte

Status:

Bit	7	6	5	4	3	2	1	0
Meaning	0	1	Busy?	Mode1	Mode0	Memory error?	0	ALU Saturation

- **Busy indication (bit 5):** 1 if the device is busy, which indicates that the data for the last command is not available yet. No new commands are processed if the device is busy.
- **Memory integrity/error flag (bit 2):** 0 if integrity test passed, 1 if test failed. This bit indicates whether the checksum-based integrity check passed or failed. The memory error status bit is calculated only during the power-up sequence, so a newly written CRC will only be used for memory verification and status update after a subsequent device power-on reset (POR) or reset by means of the RES pin.
- **ALU saturation (bit 0):** If the last command was a measurement request, this bit is 0 if any intermediate value and the final SSC result are in a valid range and no SSC-calculation internal saturation occurred, respectively. If the last command was a measurement request, this bit is 1 if an SSC-calculation internal saturation occurred. This bit is also 0 for any non-measurement command.

3.3   Writing the Slave-address

Because during a write mode always 16 Bit are written into the MTP but the I²C address is only 7 Bit long the content of the remaining 9 Bit has to be read.

Reading the existing Slave-address:

Command Request (Read MTP):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	MTP-Address **0x02** (1 Byte)
Master	**Master**

Command Response (Read MTP):

Slave-Address << 1 + 1 (Read-Bit) (1 Byte)	Status (1 Byte)	MTP-Data Bits <15:8> (1 Byte)	MTP-Data Bits <7:0> (1 Byte)
Master	**Slave**	**Slave**	**Slave**

Set new I²C-Address:
MTP-Data <6:0> = new address that will be written

Command Request (Write MTP):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	MTP-Address **0x42** (1 Byte)	MTP-Data Bits <15:8> (1 Byte)	MTP-Data Bits <7:0> (1 Byte)
Master	**Master**	**Master**	**Master**

After writing the slave address the MTP check sum should be generated newly and written
(see "**3.4. Calculate and write MTP-checksum**")
If this is not done the status byte will show the checksum error "Memory integrity/error flag (bit 2)"
(see "**3.2 Interpretation of the status-byte**").
To finalize writing the slave address a Power On Reset has to be done.

Recommended:
To finalize writing the slave address a reset and new power-up sequence must be triggered by an IC-reset signal (high low) at the RES pin.

Alternatively a Power On Reset has to be done. Due to internal capacitance values this can take up to 3 minutes.

Attention: an I²C address from 4 to 7 must not be used . If such an address is set a communication with the sensor is no longer possible!

3.4 Calculate and write MTP-checksum

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	0x90 (1 Byte)
Master	**Master**

3.5 Reading general sensor data

General data of the sensor module can be read from the MTP to ensure a unique identification.

Based on this data the master electronics can confirm that the right sensor module with the correct technical specification is used.

Information	unit	MTP-address	interpretation
Start of measuring range pressure	float32	0x25 und 0x26	0x25 Low Word 0x26 High Word
End of measuring range pressure	float32	0x27 und 0x28	0x27 Low Word 0x28 High Word
Measurement unit	uint16	0x29	0 = bar 5 = MPa 11 = psi Bit 8: 0 = gauge, 1 = absolute
Serial number	dtSerial	0x2A bis 0x34	WIKA serialnumber (11 Bytes) in ASCII (only Low Byte per MTP-address is used)
Partnumber	uint32	0x35 und 0x36	0x35 Low Word 0x36 High Word WIKA device partnumber

Example reading start of measurement range:

Command Request (Read MTP):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	MTP-Address **0x25** (1 Byte)
Master	**Master**

Command Response (Read MTP):

Slave-Address << 1 + 1 (Read-Bit) (1 Byte)	Status (1 Byte)	MTP-Data Bits <15:8> (1 Byte)	MTP-Data Bits <7:0> (1 Byte)
Master	**Slave**	**Slave**	**Slave**

Command Request (Read MTP):

Slave-Address << 1 + 0 (Write-Bit) (1 Byte)	MTP-Address **0x26** (1 Byte)
Master	**Master**

Command Response (Read MTP):

Slave-Address << 1 + 1 (Read-Bit) (1 Byte)	Status (1 Byte)	MTP-Data Bits <15:8> (1 Byte)	MTP-Data Bits <7:0> (1 Byte)
Master	**Slave**	**Slave**	**Slave**

Interpretation example start of measurement range pressure, end of measurement range pressure, unit:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	1508	0000	0000	A319	FB28	6A6D	C439	02AB	5B82	0186	9DDA	AFFD	0148	0B26	0602	0000
1x	0007	3700	F21A	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
2x	0000	C1A0	0000	42C8	0000	0000	0000	0000	40C0	0000	0031	0041	0030	0030	0053	004E
3x	0056	0048	0033	0033	0035	EC3B	00D9	0000	0000	30A7						

Tabs: Communication | MTP | Cycle Communication | Compensation

Start of measurement range pressure (0x25, 0x26) → Hexadecimal 0000 0000 → float 0
End of measurement range pressure (0x27, 0x28) → Hexadecimal 40C0 0000 → float 6
unit (0x29) → 0000 → 0 → „bar gauge"
Pressure measurement range of the sensor is: 0 … 6 bar gauge

Interpretation serial number:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	1508	0000	0000	A319	FB28	6A6D	C439	02AB	5B82	0186	9DDA	AFFD	0148	0B26	0602	0000
1x	0007	3700	F21A	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
2x	0000	C1A0	0000	42C8	0000	0000	0000	0000	40C0	0000	0031	0041	0030	0030	0053	004E
3x	0056	0048	0033	0033	0035	EC3B	00D9	0000	0000	30A7						

(Tabs: Communication | MTP | Cycle Communication | Compensation)

0x2A until 0x34 → Hex: 31 41 30 30 53 4E 56 48 33 33 35 → ASCII: 1A00SNVH335
Serial number is 1A00SNVH335.

Interpretation partnumber:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	1508	0000	0000	A319	FB28	6A6D	C439	02AB	5B82	0186	9DDA	AFFD	0148	0B26	0602	0000
1x	0007	3700	F21A	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
2x	0000	C1A0	0000	42C8	0000	0000	0000	0000	40C0	0000	0031	0041	0030	0030	0053	004E
3x	0056	0048	0033	0033	0035	EC3B	00D9	0000	0000	30A7						

(Tabs: Communication | MTP | Cycle Communication | Compensation)

0x35 und 0x36 → Hex: 00D9EC3B → Decimal: 14281787
Partnumber is 14281787.

3.6   Using the EOC-pins

Measurement duration is defined as the time from the high/low transition on the EOC pin at the beginning of the measurement until the next low/high transition on the EOC signal at the end of a single measurement.

4. Abbreviations

MBA = Start of measuring range ("**M**ess**B**ereichs**A**nfang")

MBE = End of measuring range ("**M**ess**B**ereichs**E**nde")

MTP = multiple-time programmable memory

ALU = arithmetic logic unit

SSC = sensor signal conditioner

5. Change log

Document Version	Reason for change	Date
1.0	Initial release	01.08.2018
1.01	- Changed „Busy indication": Note regarding cyclic mode eliminated cyclic mode not available! - Actual mode (bits 4:3) eliminated from "Status Bit" used internally only	29.08.2018
2.0	- Adding point 3.5. "Reading general sensor data" - correction digit value temperature on page 3 - editorial changes	17.02.2020
3.0	Adding MTF-1, incl. oversampling = 4	11.05.2021

WIKA subsidiaries worldwide can be found online at www.wika.com.
WIKA-Niederlassungen weltweit finden Sie online unter www.wika.de.

WIKA

WIKA Alexander Wiegand SE & Co. KG
Alexander-Wiegand-Strasse 30
63911 Klingenberg • Germany
Tel. +49 9372 132-0
Fax +49 9372 132-406
info@wika.de
www.wika.de