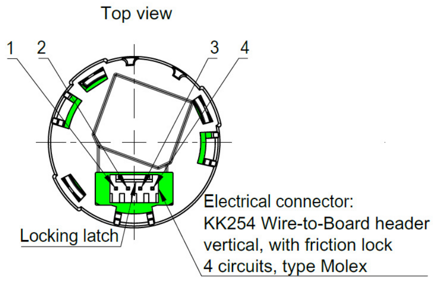
Application Note:

# $NO_2$ - Industrial Sensor / Type I-56D

## .: ELECTRICAL INTERFACE :.

**Pin 1: 0 V**
**Pin 2: SDA**
**Pin 3: SCL**
**Pin 4: Between +3.3V to +5.0V**




## .: GENERAL INFORMATION :.

I2C_Start, I2C_Stop, I2C_Sendbyte and I2C_Readbyte are functions defined by the programming language. For more information see datasheet of I2C-device and programming library of your I2C-Master device.

All measurements, tests and data read outs refer to a supply voltage of 3.3V and the configurations mentioned below (set by the configuration byte). The change of the configuration byte may change the sensor properties and is not recommended by ITG.

## .: READ OUT SENSOR DATA :.

| | |
|---|---|
| **Address (write to device):** | **0xDC** |
| **Address (read from device):** | **0xDD** |
| **ADC type (see datasheet):** | **MCP3427** |

| No. | Step | Description | Example (pseudo code) | Result |
|---|---|---|---|---|
| **#1** | Configure | Send configuration byte (0x98) after *bus connection* of sensor or *power on reset* | I2C_Start()<br>I2C_Sendbyte(0xDC)<br>I2C_Sendbyte(0x98)<br>I2C_Stop() | ADC is configured |
| **#2** | Read Out | Read sensor signal as 3 Byte repetition | I2C_Start()<br>I2C_Sendbyte(0xDD)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(NACK)<br>I2C_Stop() | 3 Byte repetition until NACK send by master:<br>Upper Data Byte (8 Bit); Lower Data Byte (8 Bit); Configuration Byte (8 Bit) |
| **#3** | Control Option | Compare (Bit 0-6) of configuration byte from read out (#2) to configuration byte 0x98 (#1) | - | Read out is performed correct |

**Important note:** The read out value (#3, Upper Data Byte and Lower Data Byte) is an INT(16) number and may not be treated as INT(32). Calculation example for INT(16): FF1A = -230.




**IT Dr. Gambert GmbH** .: Hinter dem Chor 21 .: 23966 Wismar .: Germany
Phone: +49 (0)3841 220 050 .: Fax: +49 (0)3841 220 052 2 .: E-Mail: sales@itg-wismar.de **www.itg-wismar.de**

## .: READ OUT TEMPERATURE :.

**Address (write to device):** **0x36**
**Address (read from device):** **0x37**
**Temperature sensor type (see datasheet):** **MCP9808**

| No. | Step | Description | Example (pseudo code) | Result |
|---|---|---|---|---|
| #1 | Select and Read Out | Read out 2 Bytes from temperature device [Bit 4 – 7 (Upper Data Byte) = flag bits] | I2C_Start()<br>I2C_Sendbyte(0x36)<br>I2C_Sendbyte(0x05)<br>I2C_Start()<br>I2C_Sendbyte(0x37)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(NACK)<br>I2C_Stop() | Temperature device is activated and 2 Bytes received from temperature register<br><br>**Bit 7 (Upper Data Byte):** Critical temperature (not used)<br>**Bit 6 (Upper Data Byte):** Upper limit (not used)<br>**Bit 5 (Upper Data Byte):** Lower limit (not used)<br>**Bit 4 (Upper Data Byte):** Algebraic sign<br>**Bit 0 - 3 (Upper Data byte) +**<br>**Bit 0 - 7 (Lower Data Byte):** ambient temperature bits<br><br>**Clear flag bits:** Upper Data Byte = Upper Data Byte & 0x1F<br><br>**Temperature $T_A \geq 0$ °C [Bit 4 (Upper Data Byte) = 0]**<br>$T_A$ = (Upper Data Byte x $2^4$ + Lower Data Byte x $2^{-4}$)<br>**Temperature $T_A < 0$ °C [Bit 4 (Upper Data Byte) = 1]**<br>$T_A$ = 256 - (Upper Data Byte x $2^4$ + Lower Data Byte x $2^{-4}$) |

## .: READ OUT EEPROM DATA :.

**Address (write to device):** **0xA6**
**Address (read from device):** **0xA7**
**EEPROM type (see datasheet):** **24LC32A**

**Example for read out the first two Bytes (address: 0x00)**

| No. | Step | Description | Example (pseudo code) | Result |
|---|---|---|---|---|
| #1 | Select and Read Out | Send selected EEPROM register address as High Byte and Low Byte to EEPROM device and read out EEPROM content until NACK send by master | I2C_Start()<br>I2C_Sendbyte(0xA6)<br>I2C_Sendbyte(0x00)<br>I2C_Sendbyte(0x00)<br>I2C_Start()<br>I2C_Sendbyte(0xA7)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(NACK)<br>I2C_Stop() | EEPROM register address is selected and get back EEPROM content (Bytes) until NACK send by master |

The EEPROM content (e. g. serial number, manufacturing data, sensitivity, ambient measurement conditions, temperature and humidity coefficients and **data types**) depends on individual agreements with customer (see EEPROM document).




**IT Dr. Gambert GmbH** .: Hinter dem Chor 21 .: 23966 Wismar .: Germany
Phone: +49 (0)3841 220 050 .: Fax: +49 (0)3841 220 052 2 .: E-Mail: sales@itg-wismar.de **www.itg-wismar.de**

## .: READ OUT BATTERY VOLTAGE :.

**Address (write to device):** **0xDC**
**Address (read from device):** **0xDD**
**ADC type (see datasheet):** **MCP3427**

| No. | Step | Description | Example (pseudo code) | Result |
|---|---|---|---|---|
| #1 | Configure | Send configuration byte (0xA8) after *bus connection* of sensor or *power on reset* | I2C_Start()<br>I2C_Sendbyte(0xDC)<br>I2C_Sendbyte(0xA8)<br>I2C_Stop() | ADC is configured |
| #2 | Read Out | Read sensor signal as 3 Byte repetition until (Byte 3) bitwise AND 0x80 == 0 | I2C_Start()<br>I2C_Sendbyte(0xDD)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(ACK)<br>I2C_Readbyte(NACK)<br>I2C_Stop() | 3 Byte repetition until NACK send by master:<br>Upper Data Byte (8 Bit); Lower Data Byte (8 Bit);<br>Configuration Byte<br>(8 Bit)<br>verify<br>(Configuration Byte) bitwise AND 0x80 == 0 |
| #3 | Calculate | Use Upper Data Byte (HByte) and Lower Data Byte (LByte) to calculate battery voltage | - | Battery voltage [µV] |

## .: COMPENSATION, CALIBRATION AND MEASUREMENT :.

**Compensation of electronic offset :** Each sensor electronic generates - by principle - a very low but individual offset. This has to be taken into account especially when measuring very low concentrations. ITG offers a very simple solution to eliminate the influence of the sensor electronics and to obtain accurate readings:

The offset of each individual sensor electronic is determined by ITG and stored in the individual sensor EEPROM as a constant (see EEPROM documentation eeprom_content_I-56D). **There is a flag value which indicates if the values are programmed or not** (see EEPROM documentation eeprom_content_I-56D). **Please contact ITG if the constants are not programmed.**

**Just add to each value which your receive from the ADC the offset constant from the EEPROM (constant may be positive or negative, take algebraic sign into account).** No other actions are necessary.

Corrected signal [Digits] = Signal read out [Digits] + EEPROM constant [Digits]

**Compensation of ambient conditions:** The influence and compensation of ambient condition is explained in the separate document Compensation_of_ambient_conditions_for_tracegas_sensors

**How to measure and calibrate (to keep the example simple the electronic offset compensation mentioned above is not applied in the calculation below):**

**1.** Supply sensor with zero gas (contains no $NO_2$ , e.g. $N_2$ or scrubbed air) and record sensor signal as baseline (temperature, humidity, pressure and flow constant). Convert Upper Data Byte and Lower Data Byte of senor signal to decimal number (0 to 32768).
**2.** Supply sensor with calibration gas (e. g. 1ppm $NO_2$ bal. $N_2$) and record sensor signal as calibration signal (no change in temperature, humidity, pressure and flow). Convert Upper Data Byte and Lower Data Byte of senor signal to decimal number (0 to 32768).
**3.** Calibration signal minus baseline (=span) as decimal number equals calibration gas concentration (e. g. 1ppm $NO_2$ bal. $N_2$) The sensitivity of the sensor can be calculated.
**4.** Measure other test gases and Read Out sensor output. Change of decimal number is linear to change of $NO_2$ concentration over full scale if measurement and calibration conditions are the same (repeat calibration if measurement conditions differ from calibration conditions).

**Example**:

| | | | |
|---|---|---|---|
| **1-3.** Calibration | Baseline (100 Vol.%$N_2$): | Sensor output: 200 Digits | |
| | Calibration gas (1ppm NO bal. $N_2$): | Sensor output: 5200 Digits | |
| | Span and sensitivity (20ppm NO): | Span = 5200 -200 Digits = 5000 Digits | Sensitivity = 5000 Digits / 1ppm |
| **4.** Measurement | Test gas (3ppm NO bal. $N_2$): | Span: 15000 Digits | 5000 Digits / 1ppm * 3ppm = 15000 Digits |
| Measurement | Test gas (0.1ppm NO bal. $N_2$): | Span: 500 Digits | 5000 Digits / 1ppm * 0.1 ppm= 500 Digits |



**IT Dr. Gambert GmbH** .: Hinter dem Chor 21 .: 23966 Wismar .: Germany
Phone: +49 (0)3841 220 050 .: Fax: +49 (0)3841 220 052 2 .: E-Mail: sales@itg-wismar.de **www.itg-wismar.de**