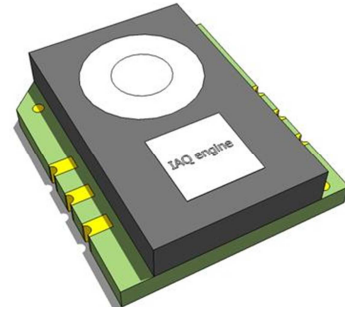AppliedSensor

# Manual iAQ-engine

Indoor Air Quality sensor

- Digital and analog I/O
- SMD type package



---

## Product summary

iAQ-engine is used for measuring VOC levels that can be read out as a prediction via the $I^2C$ bus. The iAQ-engine can also control equipment directly by an analog output with a 0-5 V DAC.

The sensor itself is protected by a plastic cap and a filter membrane. The sensor module can be soldered directly to a host circuit board with selective soldering.
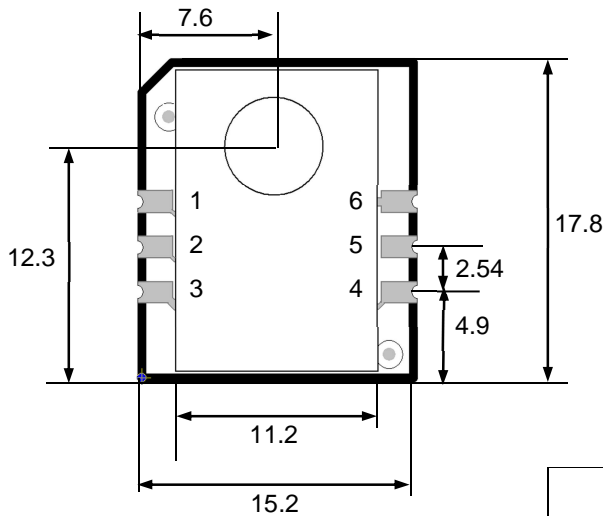
---

## Dimensions



**Figure 1**: iAQ-engine sensor
(dimensions in mm, Top View)

| Pin | Name | Comment |
|-----|------|---------|
| 1 | PRED | Prediction I/O |
| 2 | SCL | Serial clock |
| 3 | GND | Ground |
| 4 | SDA | Serial data |
| 5 | NC | Not connected |
| 6 | VCC | +5V |

| | |
|---|---|
| Dimensions (approximate values) | PCB 15.24 x 17.78 mm |
| | HEIGHT PCB 1.7 mm |
| | HOOD 11.2 x 17.78 mm |
| | TOTAL HEIGHT 4.3 mm |
| Sensor position (approximate values) | 7.6 x 12.3 mm |
| | Radius 3,5 mm |
| Weight | Approximately 1g |
| IP-Class | 00 |
| Connector | Card edge (cut via) |

# 1 Electrical specifications

## 1.1 Power supply

| Voltage | 5.0 ± 0.25V, max. 20mV ripple |
|---|---|
| Power consumption | 225mW @ 5.0VDC |

**Note**: Module features a decoupling capacitor.

## 1.2 Communication

| Output signal options | $I^2C$ |
|---|---|
| | DAC (0-5)V |
| First functional reading after start up | 15 minutes |

➔ For more communication details see chapter 4

# 2 Environmental

| Temperature range operation | 0 to 50°C |
|---|---|
| Temperature range storage | -25 to 50°C |
| Humidity range | 5 to 95 %r.h., non-condensing |

# 3 Sensor Features

| Sensing technology | MEMS metal oxide sensor |
|---|---|
| Sensing range | I²C:   450 – 65535 ppm $CO_2$ equivalents (relative) |
| | DAC: 450 – 2000 ppm $CO_2$ equivalents (relative) |
| Module | Automatic baseline correction |

# 4  I²C Interface

## 4.1    Interface description

### 4.1.1    Physical interface
The physical interface is two-wire open drain SCL (clock) and SDA (data).

| Pull-up resistors | External pull-up resistor required |
|---|---|
| Clock speed | 100kHz |
| Clock stretching | Bus master clock stretching support is required |

### 4.1.2    Clock stretching
Clock stretching pauses a transaction by holding the clock line low. The transaction cannot continue until the line is released to high again. Although the module could send the bytes of data at a fast rate, it could happen that the module is busy at the request time. It can then hold the clock line low after reception and acknowledgement of a byte to force the master into a wait state until the iAQ-engine module is ready for the next byte transfer in a type of handshake procedure. (See official I²C specification and user manual UM10204, http://www.nxp.com/documents/user_manual/UM10204.pdf)

### 4.1.3    Address
Standard 7 bit I$^2$C address for iAQ-engine is **decimal 90** or **hexadecimal 0x5A**. The addressing byte includes the read/write bit at the lowest significant bit. The communication with the iAQ-engine starts with **0xB5** for reading data.

| | | Address | | | | | R/ W |
|---|---|---|---|---|---|---|---|
| **Bit** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **data** | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

**Table 1**: Addressing byte for the iAQ-engine

## 4.2    Interface protocol
The standard I²C specification is used for the iAQ-engine interface protocol. The I²C bus master should request 7 bytes. These seven bytes include information about the indoor air quality value, the iAQ-engine status and the resistance of the sensor. If there is a need just for the indoor air quality value and the status, the master should request three bytes from the iAQ-engine. All bytes are reported back as shown in the following table. A graphical description for a standard I²C communication with the iAQ-engine module is shown in figure 2 – figure 5.

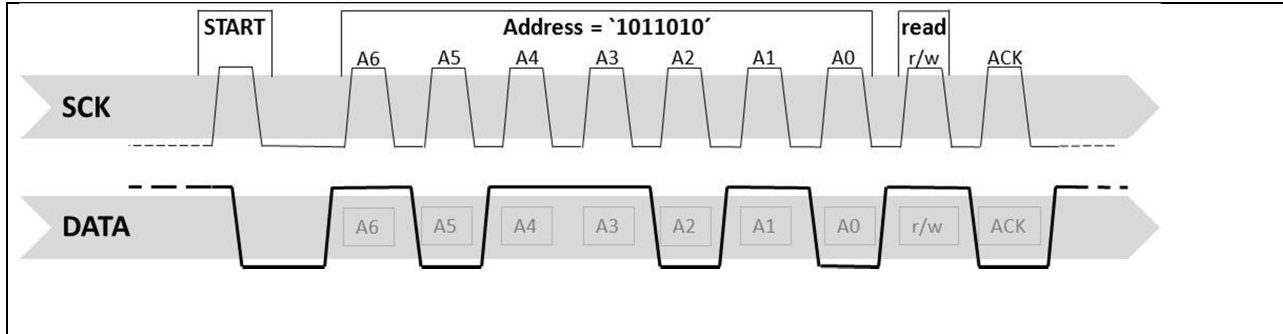| Byte | Name | Data type | Typical/example value | Explanation / notes |
|---|---|---|---|---|
| 0-1 | pred | uint16 | 450 | Prediction [ppm] |
| 2 | status | uint8 | 0 | 0x00: OK    (data valid)<br>0x01: BUSY  (re-read multi byte data!)<br>0x80: ERROR (if constant:replace sensor) |
| 3-6 | resistance | int32 | 256431 | Sensor resistance [Ohm] |

**Table 3**: Read data from the iAQ-engine

**Figure 2:** The first byte is send by the master, containing address (0x5A) and read/write bit. The slave sends an acknowledgement (ACK) by pulling the data line to low.
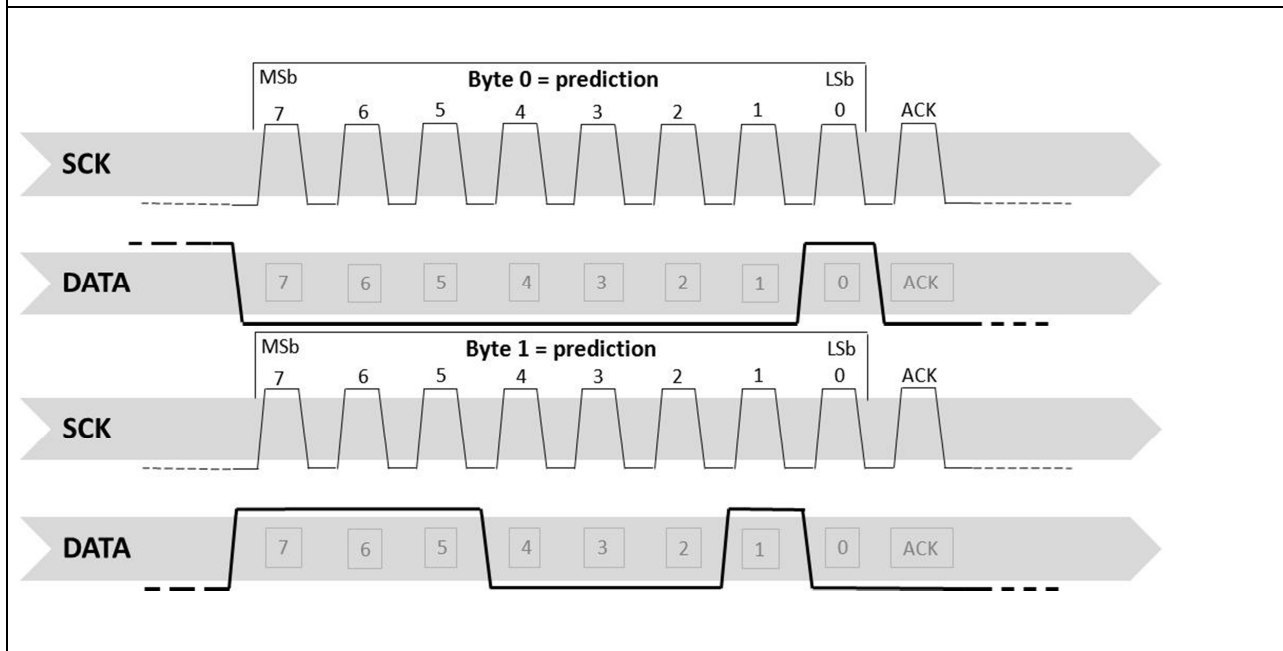


**Figure 3:** The slave will answer by sending bytes with MSB first. Byte0 and byte1 contain the prediction value. All bytes are Acknowledged by the master.
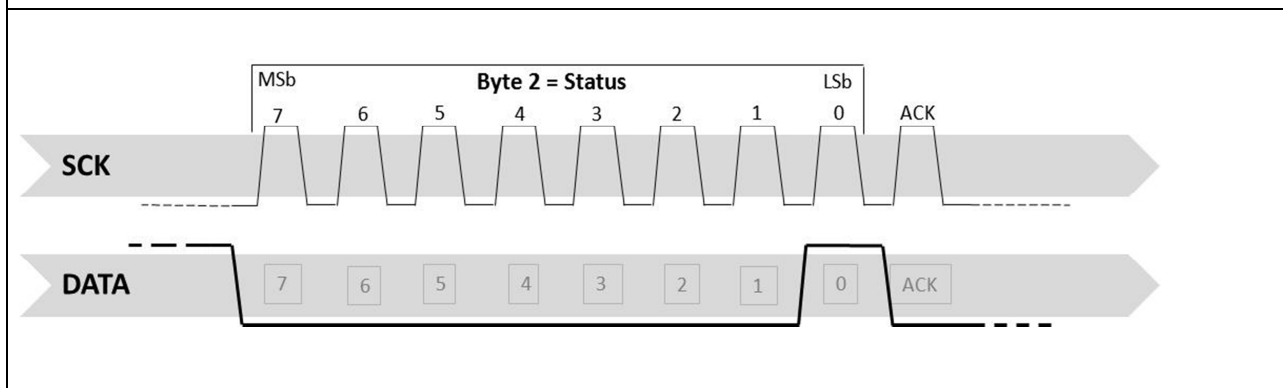


**Figure 4:** The third byte contains the information of the iAQ-engine module state, in this case status = 1. The master answeres with an Acknowledge.
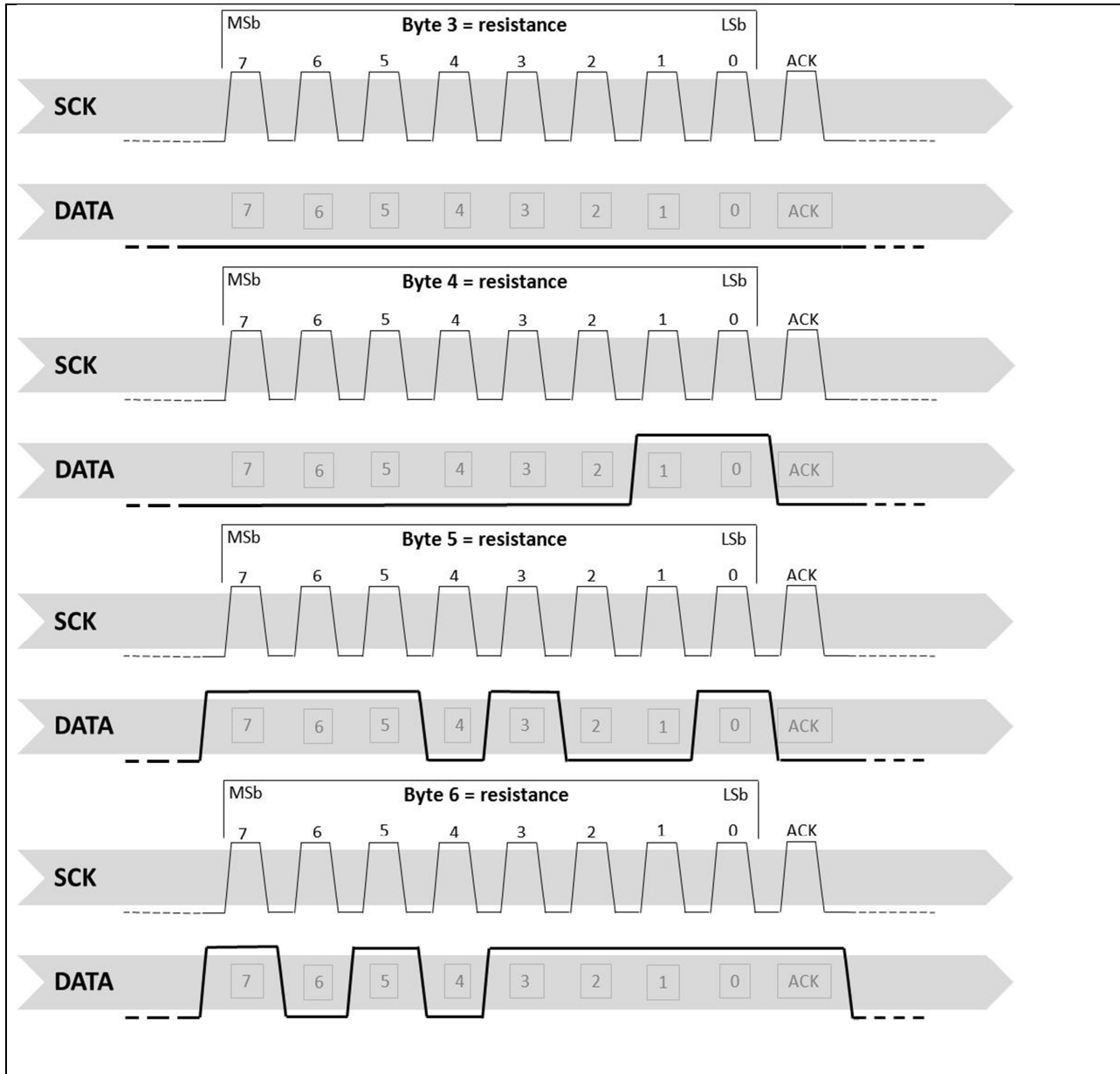
**Figure 5:** The last four bytes contain the resistance value. For the calculation of the resistance only byte4, byte5 and byte 6 are relevant, because byte3 is zero .After the last requested byte, the master sends a not Acknowledge.

## 4.2.1 Prediction

| Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
|-------|-------|-------|-------|-------|-------|-------|

The first two bytes contain the prediction value, which gives the information about the indoor air quality. The value is a $CO_2$ equivalent and the calculation is shown in the following example.

**Equation 1 :**

### 4.2.2      Status Flag

| Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
|-------|-------|-------|-------|-------|-------|-------|

The third byte indicates status of the module.
- 0x00:  OK
- 0x01:  BUSY
- 0x80:  ERROR

If status is OK the data is valid.  If the status is BUSY, the data integrity is not guaranteed for variables of size > 8 bits, because the module may be updating a part of the variable.
If the status is ERROR constantly (or very frequently) this indicates that the module is reading non-realistic values, and the sensor element is probably defective.

### 4.2.3      Resistance

| Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
|-------|-------|-------|-------|-------|-------|-------|

The next four bytes contain the sensor resistance in Ohm. The fourth byte of the int32 variable is 0.

**Equation 2:**

## 4.3   Typical applications
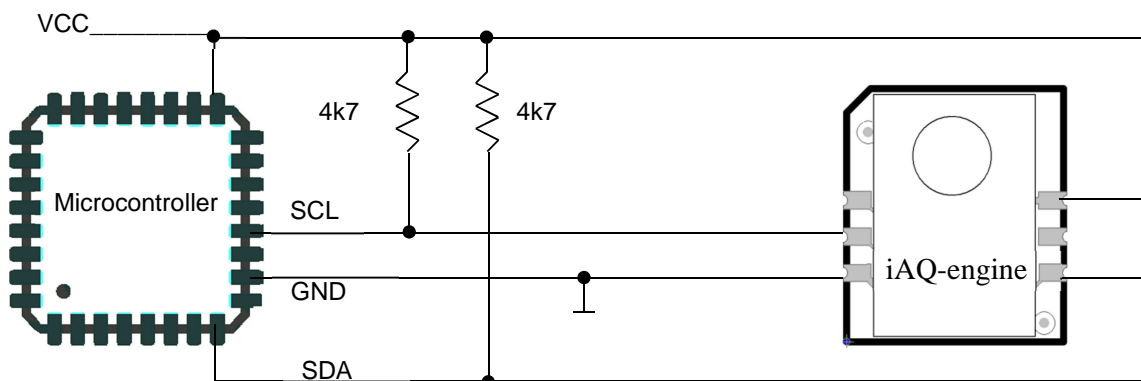


**Figure 6:** Simple microcontroller application

## 4.4    Recommended footprint
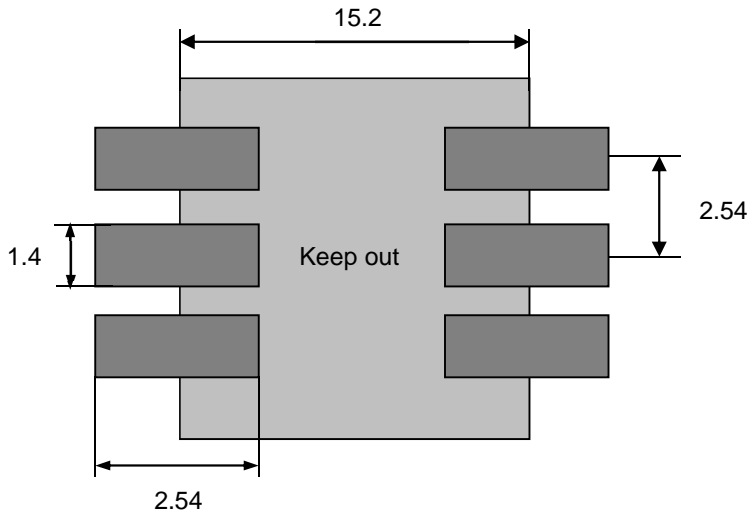


**Figure 7:** Recommended footprint (standard)

## 4.5    Ordering information

| Order code | Comment |
|---|---|
| 60-0100 | iAQ-engine |