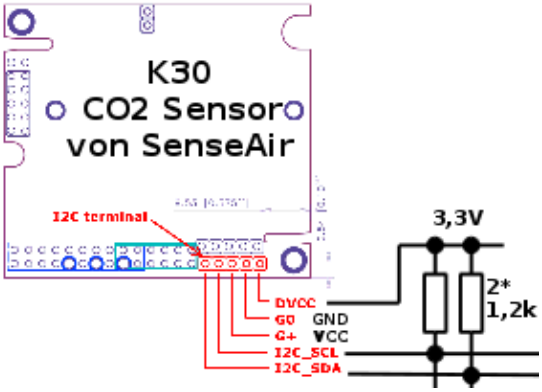
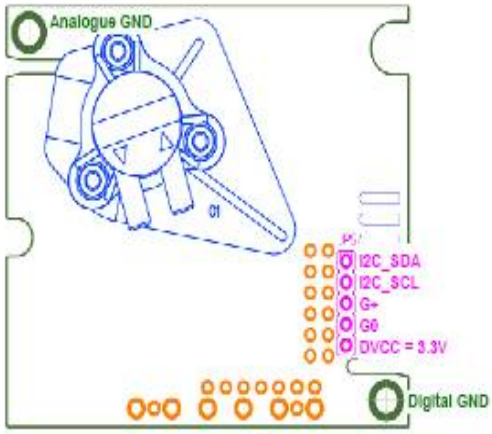
CO2-Sensor

Von Senseair

http://cdn.shopify.com/s/files/1/0019/5952/files/I2C_comm_guide_2_1031.pdf?1254748002

K30 (Messbereich 1%, Spezialanfertigung mit 2% oder 4%):	K33:
	

Kommunikation über I2C

Anschluss am I2C-Port des Controllers mit 1.2K- Pullup nach +3.3V

Beim K33 muss der Wert noch mit 10 multipliziert werden.

Testprogramm für CO2-Sensor:

Auslesen des Sensors und Daten über serielle Schnittstelle an einen PC schicken

```
'CO2-Messung
'Sensor an I2C
'PC.5 = SCL
'PC.4 = SDA

$crystal = 8000000
$regfile = "m8def.dat"
$hwstack = 50

$baud = 9600
'Enable Urxc
'Enable Interrupts
'On Urxc Rscommand


Config Portd.5 = Output                 'LED

'I2C
Config Sda = Portc.4
Config Scl = Portc.5

'CO2-Sensor
```

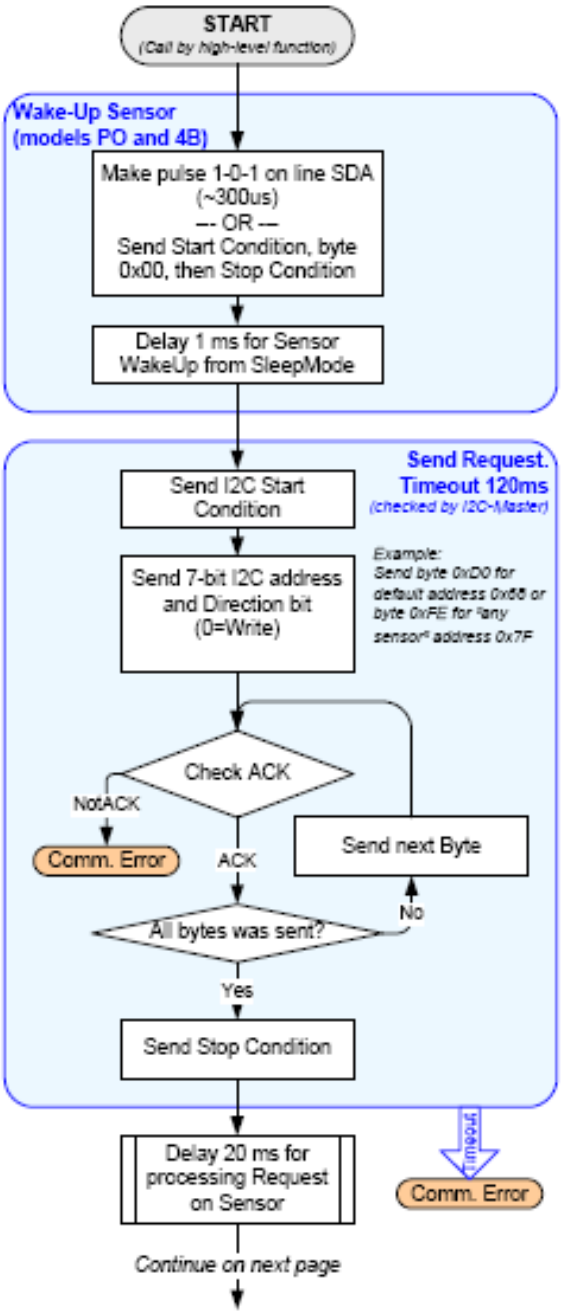
```
Dim Sensorstatus As Byte
Dim Sensorvalue As Word
Dim Sensorvalue_h As Byte At Sensorvalue Overlay
Dim Sensorvalue_l As Byte At Sensorvalue + 1 Overlay
Dim Sensorchecksum As Byte

Dim Secs As Long
'-------------------------------------------------------------------------------
Main:
Do
    Secs = Secs + 1

    Gosub Measureco2

    Print Secs;
    Print Chr(9);
    Print Sensorvalue;
    Print Chr(9);
    If Sensorstatus = 1 Then Print "OK" Else Print "Error"

    Waitms 1000

Loop
'-------------------------------------------------------------------------------
Measureco2:
'Senseair K30-Sensor über I2C ansprechen
'auf Fehler prüfen (Sensor nicht bereit) durch Auslesen des Statusbytes
'falls nicht bereit: Sensorstatus =0, falls Ok: 1

    I2cstart
    I2cwbyte 0
    I2cstop

    Waitms 1


    I2cstart
    I2cwbyte &HD0                   'Adresse (&h68 1 Bit nach links, Bit0=R/W=0 (schreiben)
    I2cwbyte &H22                   'Read RAM   2 Bytes
    I2cwbyte &H00                   'Adresse H
    I2cwbyte &H08                   'Adresse L
    I2cwbyte &H2A                   'Checksum
    I2cstop

    Waitms 20

    I2cstart
    I2cwbyte &HD1                     'Lesen an Adresse &h68
    I2crbyte Sensorstatus , Ack
    I2crbyte Sensorvalue_l , Ack
    I2crbyte Sensorvalue_h , Ack
    I2crbyte Sensorchecksum , Nack
    I2cstop

    Sensorstatus = Sensorstatus And 1
Return
```

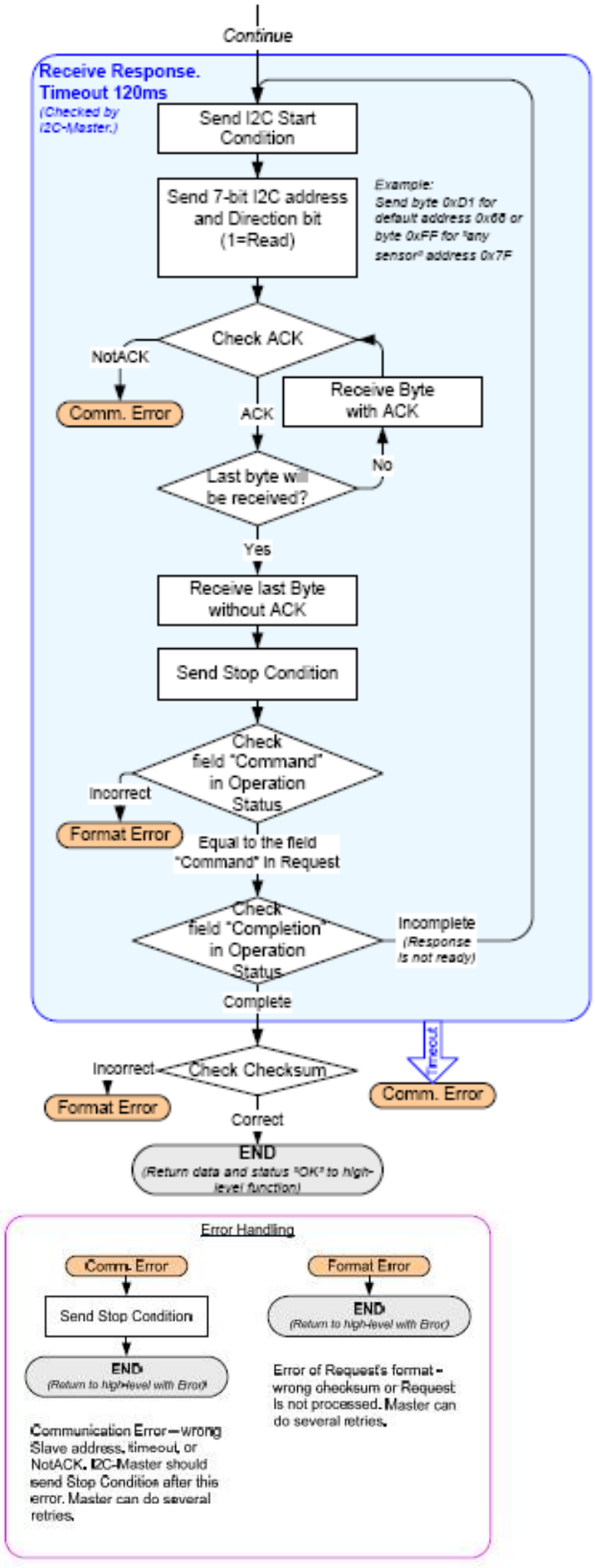
Anhang

Unterlagen von Senseair zur Programmierung

7. Flowchart of master operation and error handling

Figure 12




Continue
Receive Response.
Timeout 120ms
(Checked by I2C-Master.)
Send I2C Start Condition
Send 7-bit I2C address and Direction bit (1=Read)
Example:
Send byte 0xD1 for default address 0x68 or byte 0xFF for "any sensor" address 0x7F
Check ACK
NotACK
Comm. Error
ACK
Receive Byte with ACK
No
Last byte will be received?
Yes
Receive last Byte without ACK
Send Stop Condition
Check field "Command" in Operation Status
Incorrect
Format Error
Equal to the field "Command" in Request
Check field "Completion" in Operation Status
Incomplete
(Response is not ready)
Complete
Timeout
Incorrect
Check Checksum
Format Error
Comm. Error
Correct
END
(Return data and status "OK" to high-level function)
Error Handling
Comm. Error
Send Stop Condition
END
(Return to high-level with Error)
Communication Error – wrong Slave address, timeout, or NotACK. I2C-Master should send Stop Condition after this error. Master can do several retries.
Format Error
END
(Return to high-level with Error)
Error of Request's format – wrong checksum or Request is not processed. Master can do several retries.

10.1. Example: Reading of CO2 value from sensor

To read the current CO2 concentration from the sensor we need to read memory locations 0x08 (hi byte) and 0x09 (low byte).
To do this we need to send a sequence of two I2C frames: first we send an I2C write frame containing the sensor address, command number and how many bytes to read, RAM address to read from, and a checksum. Then we send an I2C read frame to read the status, data and checksum. See chapter 2 for details.
In our case we want to read 2 bytes starting from address 0x08. This will give us data from address 0x08 and 0x09, which contains current CO2 reading. The sensor address is 0x68 (default factory setting, configurable in EEPROM).
So, the first frame should look like:

Start | 0xD0 | 0x22 | 0x00 | 0x08 | 0x2A | Stop

A. a. 0xD0 is Sensor address and read/write bit. 0x68 shifted one bit to left and R/W bit is 0 (Write).
B. b. 0x22 is command number 2 (ReadRAM), and 2 bytes to read
C. c. Checksum 0x2A is calculated as sum of byte 2, 3 and 4.

II.
The next frame will read the actual data:

Start | 0xD1 | <4 bytes read from sensor> | Stop

A. d. The 1:st byte from the sensor will contain operation status, where bit 0 tells us if the read command was successfully executed.
B. e. The 2:nd and 3:rd byte will contain CO2 value hi byte and CO2 value low byte.
C. f. The 4:th byte contains checksum

D.

E.

F. 10.2. Example: Start background and zero calibration with I2C commands

III.
In K30 and K50 meters it is possible to start zero and background calibrations with I2C commands.
Background calibration for K30 meters look like this:

Start | 0xD0 | 0x12 | 0x00 | 0x67 | 0x7C | 0x06| 0xFB | Stop

A. a. 0xD0 is Sensor address and read/write bit. 0x68 shifted one bit to left and R/W bit is 0 (Write).
B. b. 0x12 is command number 1 (WriteRAM), and 2 bytes to write
C. c. 0x7C06 is background calibration command
D. d. Checksum 0xFB is calculated as sum of byte 2-6.

IV.
Zero calibration for K30:

Start | 0xD0 | 0x12 | 0x00 | 0x67 | 0x7C | 0x07| 0xFC | Stop

1. e. 0xD0 is Sensor address and read/write bit. 0x68 shifted one bit to left and R/W bit is 0 (Write).
2. f. 0x12 is command number 1 (WriteRAM), and 2 bytes to write
3. g. 0x7C07 is background calibration command
4. h. Checksum 0xFC is calculated as sum of byte 2-6.

In K50 meters with memory map 8 or lower the zero and background commands are identical to K30 meters. In K50 meters with memory maps higher than 8 the address to write to is moved from 0x67, 0x68 to 0x32, 0x33.

Table 11 Measured value and status

Variable	Address	Format	Notes
Space CO_2	0x08 in RAM	2 bytes signed integer, Most significant byte at lower address	Negative readings are possible, in particular at nitrogen / zero gas flow test.
Error Status	0x1E in RAM	1 Byte, bit field	Error discovered during self test is indicated by setting one of the bits in the Error Status byte to 1. Normal operation without errors is indicated by Error Status = 0. Details about error status are product specific and may be found in product technical documentation