Green.Smart.Wireless.
enocean®

CO_2 sensor

Design concept for solar-powered CO_2 sensor

1 Introduction

The concentration of carbon dioxide (CO_2) in a room is an indicator for indoor air quality (IAQ). CO_2 sensors are therefore used to measure indoor air quality in a building to perform demand-based ventilation. Sensors for this purpose operate in the range between 500 and 5000 ppm (ppm = parts per million, e.g. 10000 ppm = 1%).

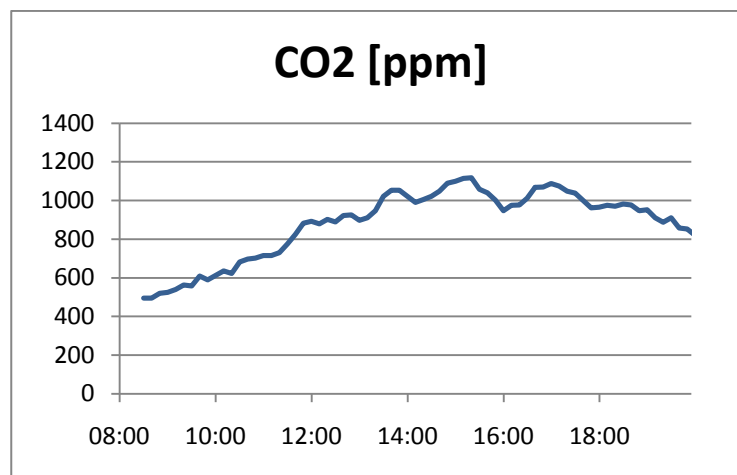The figure below shows a typical CO_2 concentration in an office during the day.

Figure 1 - example of CO2 level in an office

As shown in the table below a level of 800 to 1200 ppm inside a building is seen as "well ventilated". Achieving lower levels will result in additional energy usage e.g. by the van or heating/cooling. Achieving only higher levels might result e.g. in reduced concentration or well-being of the people in the room.

CO_2 concentration	Physiological effects
400 ppm	Outdoor – average level in nature (fresh air)
500 – 700 ppm	Indoor – over ventilated
800 – 1200 ppm	Indoor – well ventilated
1500 – 5000 ppm	Indoor – under ventilated
5000 ppm (0.5%)	MAK-level: max. concentration for an 8 hour working day
1 – 3 %	Reduced concentration, drowsiness, headaches
4 – 5 %	expiratory concentration, increased heart and lung activity
10 - 20%	Flame extinguishes, vomiting, unconsciousness
>20%	Death in a few seconds

1.1 Purpose

In order to build an energy-harvesting self-powered sensor the main challenge is energy consumption. Most of today's available CO$_2$ sensors require by far too much energy to be operated in such a system. However this document demonstrates a self-powered CO$_2$ sensor using an energy optimized sensor from Gas Sensing Solutions (see **[11.]**) in combination with a smart measuring concept based on EnOcean's STM 300 radio module.
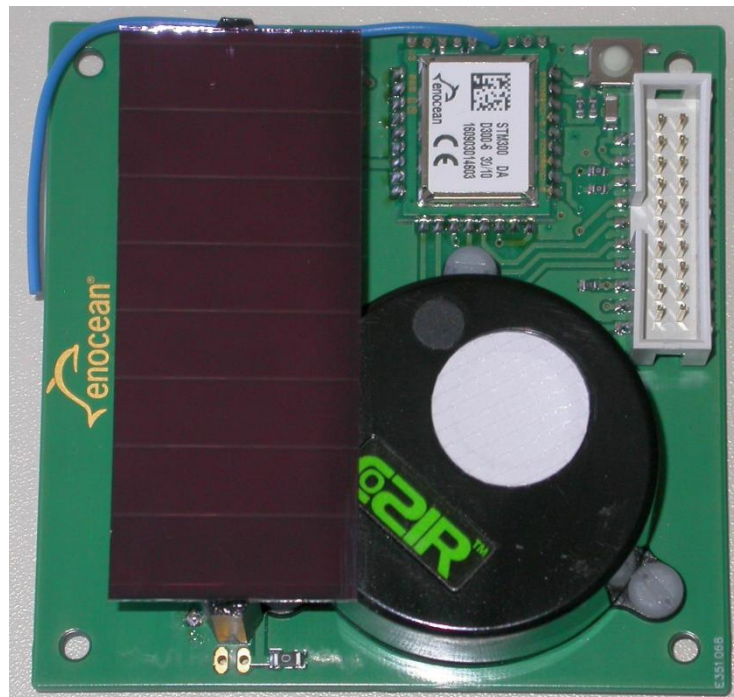

Figure 2 - Solar powered CO2 sensor

This application note describes the principles how to built such a self-powered CO$_2$ sensor and is intended as basis for own developments. It does not reflect a ready-to-use reference design. The respective software is also available for download from the EnOcean website.

NOTE: The software AN313SW which is part of this application note is provided on an "AS-IS" basis.

1.2 Table of content

1.3 References

Further details can be found in the following documentation

[1.]   DolphinAPI user manual, EO3000I_API.chm, 2.2.1.0

[2.]   STM 300 user manual, V1.25, December 3, 2010

[3.]   EnOcean Equipment Profiles (EEP) 2.1, 20. Jun. 2011

[4.]   COZIR data sheet, GSS COZIR Data Sheet.pdf (Rev B10/10)

[5.]   COZIR software user`s guide, COZIR Software User's Guide AL12 RevA.pdf

[6.]   MAX1595 step-up/step-down charge pump data sheet, Rev. 2

Useful web sites:

[7.]   EnOcean website **http://www.enocean.com**

[8.]   EnOcean Alliance website **http://www.enocean-alliance.org**

[9.]   Wikipedia website **http://en.wikipedia.org/wiki/Nondispersive_infrared_sensor**

[10.]   CO$_2$ Sensor website **http://www.cozir.com/**

[11.]   Gas Sensing Solutions website **http://www.gassensing.co.uk/**

[12.]   Maxim website **http://www.maxim-ic.com/**

2   Demonstration

The demonstrated system measures CO$_2$ concentration, ambient temperature (chip tem-perature) and storage charge level every 10 min and transmits the data via radio. In case of a poor charge level (<20%) the measurement and transmission in ceased until a higher charge level is reached. Also an automatic long term calibration mechanism is implemented taking care of long term effects like dust or dirt in the sensor.

Using an existing EnOcean equipment profile (EEP A5-09-04) the humidity level represents the charge level. Using DolphinView as receiver the system can be easily monitored with the EEP view window:

NOTE: The current implementation does not comply with the EEP A5-09-04 specifi-cation and needs modifications for an end product.
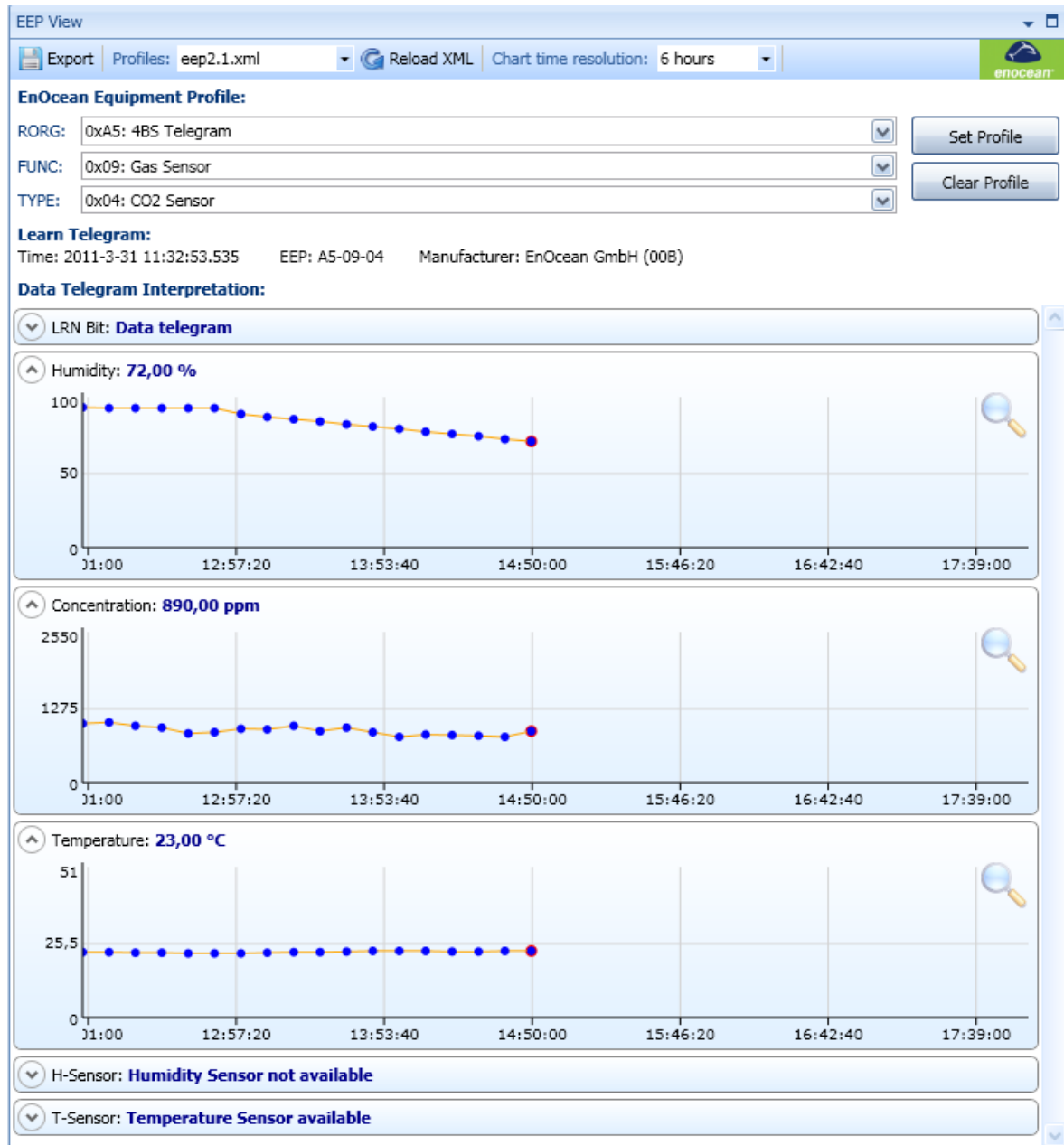
Figure 3 – DolphinView / EEP View

2.1 System overview

The block diagram below shows the functional blocks of the self-powered CO$_2$ sensor system. The energy provided by the solar cell is stored in a long term storage capacitor (super cap) supplying the system with its operating voltage. The power supply of the embedded CO$_2$ sensor is controlled from the STM 300 via the WXIDIO pin allowing switching off the embedded sensor during sleep times. The embedded sensor needs a constant 3.3 V operating voltage generated by a step-up/step-down charge pump (DC/DC converter).
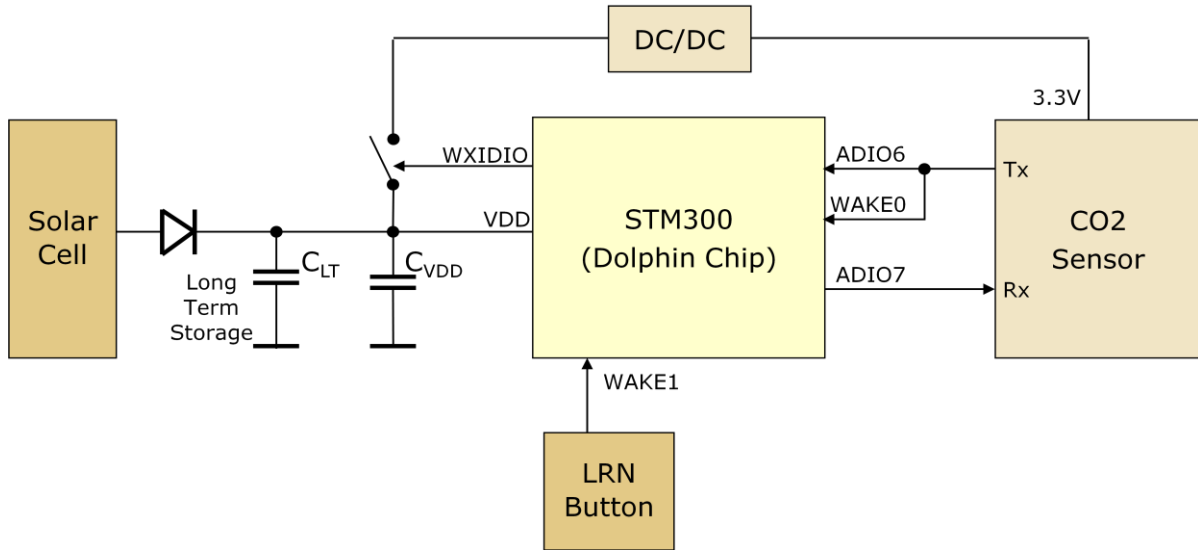
Figure 4 – Block diagram CO2 sensor

As can be seen in the **Schematics** chapter special care is required on the interface between powered and not powered components, in particular between embedded sensor and STM 300 (e.g. ADIO6, ADIO7).

2.2 Schematics

The demonstration is based on a solar cell similar (bigger size) to the ECS 300/310 with the following parameters:

Size	67.0 x 28.0 x 1.1 mm
Number of cells	8
open circuit voltage @200 lux, 25°C	4 V
short circuit current @200 lux, 25°C	25 uA
operating voltage @200 lux, 25°C	3 V
operating current @200 lux, 25°C	19 uA

For the storage of the energy provided by the solar cell 2 x 0.22 F super caps (NEC TOKIN FC0H224ZTBR24) are used. For the demonstration the super caps were selected with a high capacitance combined with the capability to provide some mA required in order to provide the current needed by the STM 300 and embedded CO$_2$ sensor.

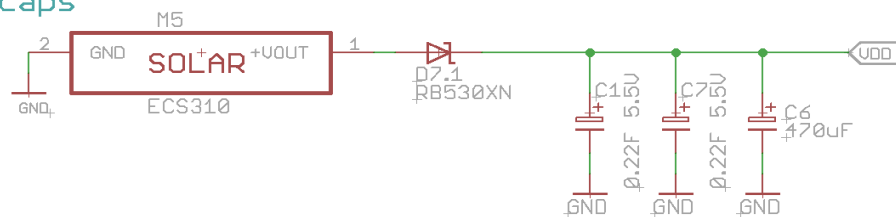**Figure 5 - Solar cell and storage capacitors**

Figure 6 – STM 300

As the embedded CO$_2$ sensor requires a fixed 3.3 V (VDD2) supply voltage a DC/DC converter is used to derive this voltage from VDD. The MAX1595 charge-pump regulator generates 3.3 V from a 1.8 V to 5.5 V input. The solar cell generated VDD (operating) can vary from 1.8 V up to 5.5 V during operation. Two MOSFETs enable to turn off the sensor supply for energy consumption reasons. Using the WXIDIO pin for the control, which is also supplied during deep sleep, allows supply the sensor even if the STM 300 is in deep sleep mode. The other pins ADIO0-7, SCSEDIO0, SCLKDIO1, WSDADIO2, RSDADIO3 are not powered in deep sleep mode and can't be used for this purpose.

Figure 7 - sensor supply (DC/DC)

As already mentioned before, special care is needed on the electrical interface between powered and not powered parts of the system. For instance it's not allowed to apply any voltage on the ADIO6 pin (diode D1) during deep sleep mode (see **[2.]** 2.3.1 GPIO supply voltage).
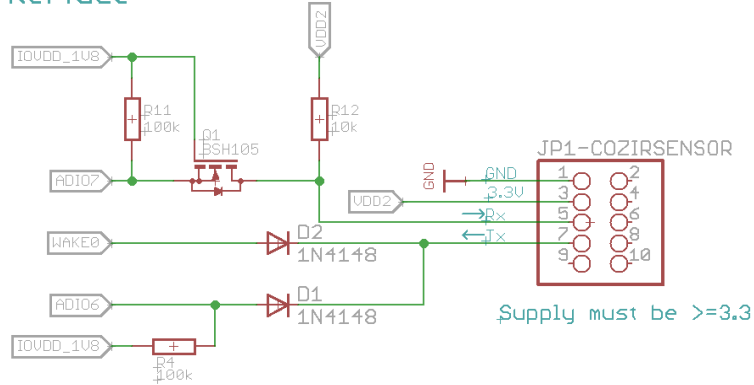
Figure 8 - Sensor interface

To program the STM 300 in system a connector to the EOP 300 programmer is foreseen. The EOP 300 needs to be operated with 3.3 V. The R7 and R8 0 Ohm resistors are optional for debugging purposes to gain access to serial port.
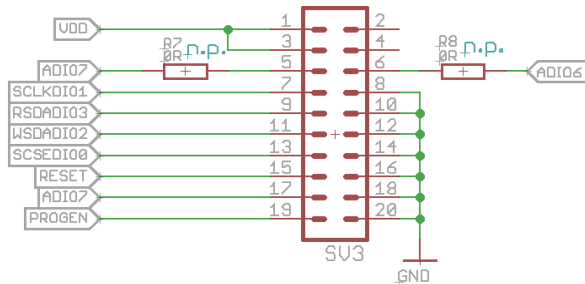

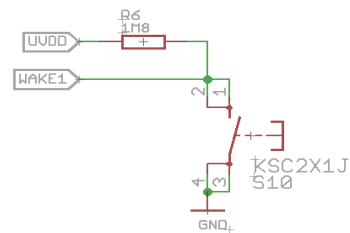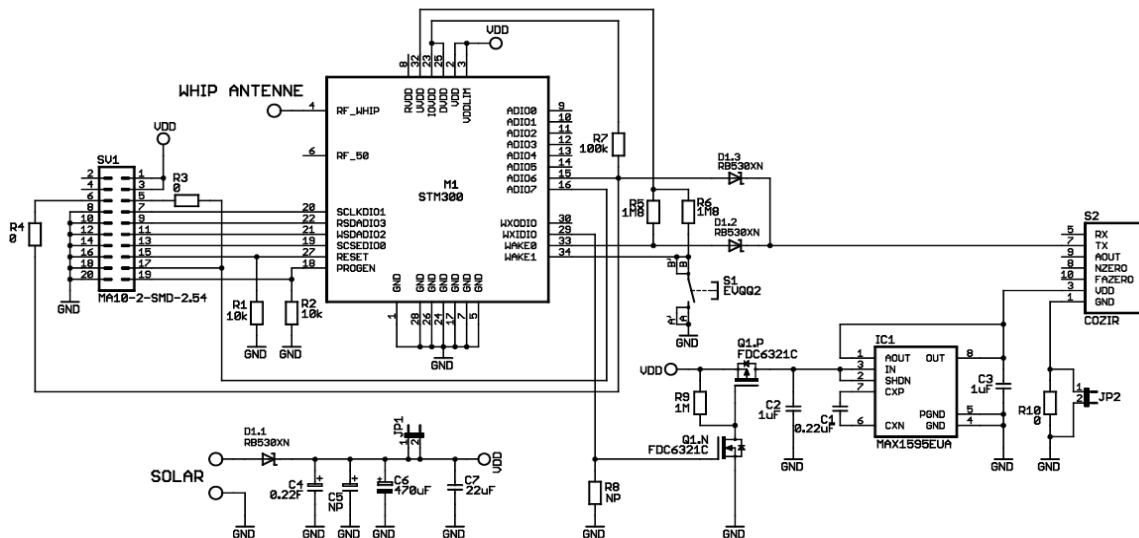**Figure 9 - Programming interface**

Figure 10 - LRN button

Figure 11 – DEMO Implementation example

2.3 Embedded CO$_2$ sensor

The COZIR™ ultra low power carbon dioxide sensor uses non-dispersive infrared (NDIR also see **[9.]**) configuration, utilizing unique mid-infrared light emitting diode source and photo diode detector technology.

The sensor is optimized for battery powered application and has the following key features:

- Low power consumption (3.5 mW in continuous operation)
- Fast power-up time
- Battery operated (3.3 V)
- Auto calibration
- Standard digital output (serial interface)
- Optional humidity and temperature measurement available

The COZIR carbon dioxide sensor is available in two ranges:

- **COZIR ambient** with a 0 to 2,000 ppm range suitable for applications such as heating, ventilation and air conditioning (HVAC), indoor air quality (IAQ), education and horticulture.

- **COZIR wide range** with a 0% to 100% range suitable for process control applications such as diving, industrial, safety and automotive.

2.4 CO2 measurement

In a self powered application there is not enough energy available to permanently supply the embedded sensor. Therefore a dynamic power concept is required to reduce the energy consumption.

After powering up of the embedded sensor it takes about 2 s till the sensor delivers the first measurement result on the serial interface. During this 2 s the sensor has actually per-formed 4 internal measurements used to calculate the first result. The STM 300 enters deep sleep mode for this 2 s in order to reduce current consumption. After the measurement result was received the sensor power supply is switched of and the result is send via radio. Then the STM 300 enters deep sleep mode for e.g. 10 minutes.
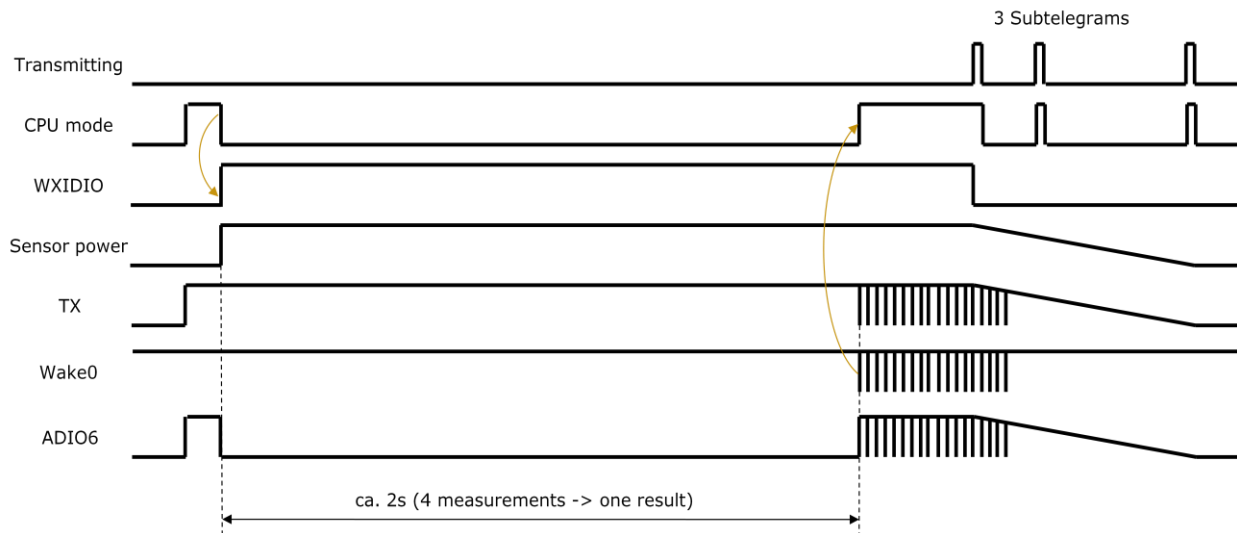
Figure 12 - one measurement cycle

One measurement cycle:

1. STM 300 module wakes up from deep sleep mode after the preset time (Watchdog Timer Reset)

2. STM 300 turns on sensor power (set WXIDIO to high)

3. STM 300 enables wakeup on sensor TX line and sets watchdog timer for sensor timeout (3 s in case no answer-> wakeup from sensor)

4. STM 300 enters deep sleep mode to wait for sensor result (or timeout)

5. Sensor performs CO$_2$ measurement (actually 4 single measurements are made internally). After about 2 s the result is output on the serial port TX line:
   " Z 00879 z 00875\r\n"

6. The first falling edge on the TX line generates a falling edge on the WAKE0 pin waking up the STM 300 module

7. The STM 300 module starts from reset, enables the uart and receives the remaining characters on the serial line (first <space> is lost) and turns of the sensor power off as soon as it has received the necessary first characters **"Z 00879"**

8. STM 300 converts the value and sends the information via radio

9. STM 300 sets the watchdog timer to 10 minutes and enters deep sleep mode with WAKE0 wakeups disabled

2.5 Chip temperature sensor

The demonstration utilizes the possibility to use the internal chip temperature sensor for measuring ambient temperature. This works for an application where the chip is on the same temperature than ambient without any power dissipation to falsify the result.

NOTE: **The STM300 module needs to be calibrated in order to use the internal chip temperature sensor.**

2.6 Energy monitoring - charge level

To monitor the energy consumption a charge level of the capacitors is calculated and transferred via radio. The charge level is evaluated by measuring the voltage VDD on the storage capacitor. The usable voltage from 2.4 V up to 4.5 V is scaled to 0% up to 100%.
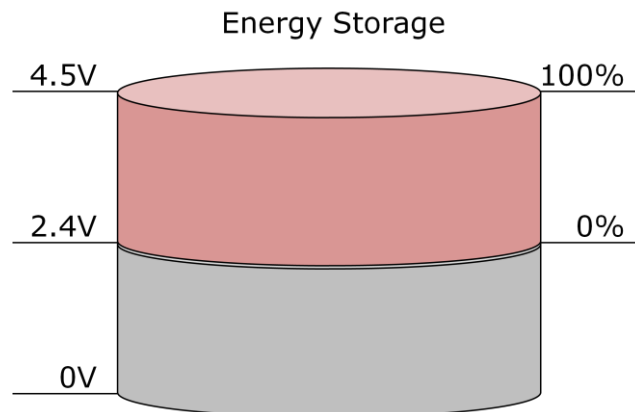

Figure 13 - Charge level

Apart from transmitting the charge level for diagnostic and energy evaluation purposes the value is also used to operate the system in a save operating area (down to 20%). Switching on the embedded sensor causes voltage dips during periods with higher current consumption of up to 0.6 V. These voltage dips might cause under voltage resets.

2.7 Radio telegram

The CO_2 sensor sends the following telegram similar to the EEP: A5-09-04

As first data field the charge level (see **Energy monitoring - charge level**) is send instead of the rel. humidity.

NOTE: The current implementation does not comply to the EEP A5-09-04 specification and needs modifications for an end product.

Offset	Size	Description	Valid range	Scale	Unit
0	8	Charge level (linear 0.5%)	0..200	0..100	%
8	8	CO2 concentration (linear 10ppm)	0..255	0..2550	ppm
16	8	Temperature (linear 0.2°C)	0..255	0..+51.0	°C
28	1	LRN bit	1 data		
29	1	Humidity valid	0 not valid		
30	1	Temperature valid	1 valid		

The figure below shows the actual EEP A5-09-04 profile as reference:

RORG	A5	**4BS Telegram**
FUNC	09	Gas Sensor
TYPE	04	CO2 Sensor

Offset	Size	Bitrange	Data	ShortCut	Description	Valid Range	Scale	Unit
0	8	DB3.7...DB3.0	Humidity	HUM	Rel. Humidity (linear), 0.5 % = 1 bit	0...200	0...100	%
8	8	DB2.7...DB2.0	Concentration	Conc	Concentration (linear), 10 ppm = 1 bit	0...255	0...2550	ppm
16	8	DB1.7...DB1.0	Temperature	TMP	Temperature (linear), 0.2 °C = 1 bit	0...255	0...+51.0	°C
24	4	DB0.7...DB0.4	Not Used (= 0)					
28	1	DB0.3	LRN Bit	LRNB	LRN Bit	Enum: 0: Teach-in telegram 1: Data telegram		
29	1	DB0.2	H-Sensor	HSN	..	Enum: 0: Humidity Sensor not available 1: Humidity Sensor available		
30	1	DB0.1	T-Sensor	TSN	..	Enum: 0: Temperature Sensor not available 1: Temperature Sensor available		
31	1	DB0.0	Not Used (= 0)					

Figure 14 - EEP A5-09-04

2.8 Embedded CO₂ sensor recalibration

The embedded CO₂ sensor requires a recalibration from time to time due to effects e.g. caused by dust or dirt in the optics. The method for recalibration assumes that during a longer timeframe of e.g. 2-3 weeks the sensor will at least measure once a "fresh air" CO₂ level.

In applications where the embedded sensor is permanently supplied it automatically performs this recalibration. In the self-powered system the STM 300 needs to take care of this functionality.

For the recalibration the STM 300 needs to monitor the lowest measured CO₂ level (in a given timeframe) and the time. This data is stored in RAM0 during deep sleep periods.

```
// ----------------------------------------------------------------
// Structure of RAM0 area
typedef struct
{
  uint8 u8State;                    // used to store system state during deep sleep
  uint8 u8ChargeLevel;              // used to store charge level during sleep
  uint16 u16MinCO2Level;            // used for dynamic sensor calibration
  uint16 u16WakeupCnt;              // used for dynamic sensor calibration
} tRam0;
```

To monitor the time simply the number of wakeups is counted. After a defined time period the recalibration would be done by sending a "F" command (see **[5.]**) to the embedded sensor.

2.8.1 Using the "F" command recalibration

The recalibration uses the difference between the lowest measured CO_2 level during the timeframe (assumed to be "fresh air" level) to a known "fresh air" level of e.g. 450 ppm.

For instance if the lowest CO_2 reading was 460 ppm than the STM 300 needs to send a "F" command **"F 460 450\r\n"** to the sensor.

The following steps need to be handled

■ Ensure that the sensor is ready to receive commands (wait at least 100 ms after power on)

■ Send the actual F command with the calibration values

■ Wait till F command is executed and calibration data is stored inside the embedded sensors EEROM memory. This can be evaluated by waiting for the sensor to reply to the F command.
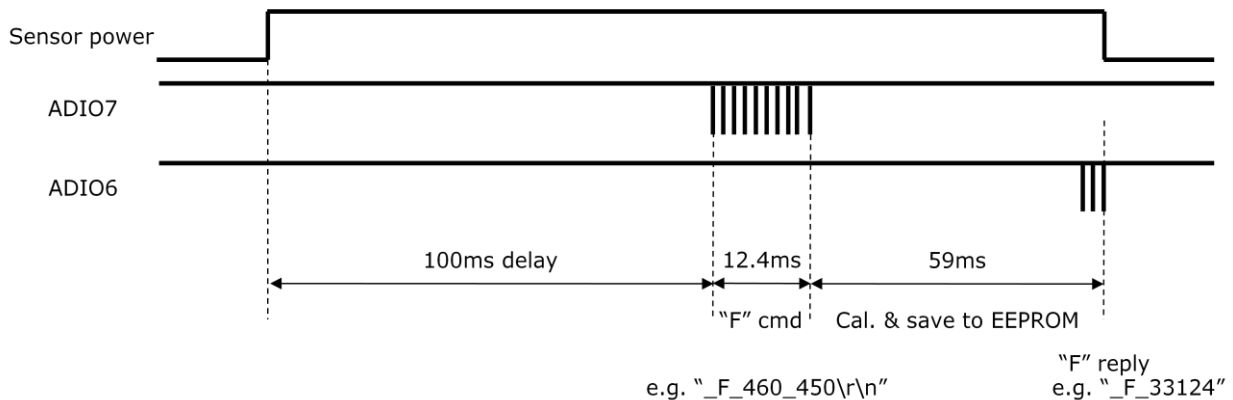

Figure 15- F command recalibration

NOTE: The actual implementation does not wait for the reply and simply waits 100 ms and then turns the sensor supply off.

2.9 Some energy considerations

Evaluating energy consumption has shown that the used solar cell provides enough energy to operate the sensor. To maintain a steady state condition (amount of energy provided by the solar cell = amount of energy used by the application) about 180 lux (waking up every 10 minutes) are needed. This are about 16.6 uA * 600 s = 10000 uAs during a 10 minutes interval.

Figure 15 shows how the energy is consumed between the different functional blocks. The most energy is related to the CO_2 measurement: about 60% is used by the embedded sensor for the measurement (powered for 2 s). An additional 40% are caused by the DC/DC converter supply current and the converting efficiency.
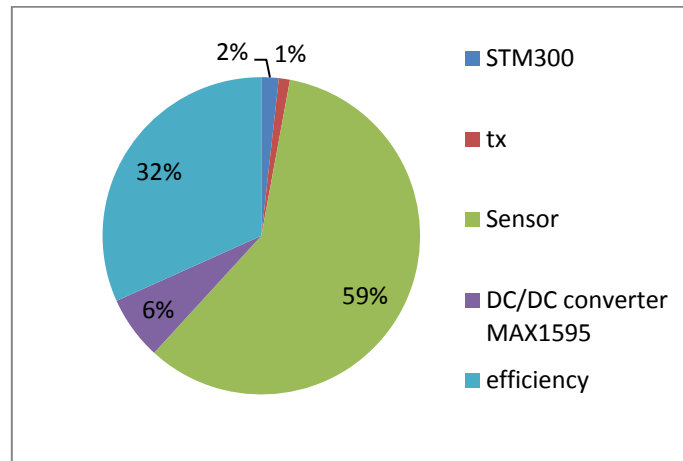
Figure 15 - Energy consumption

3 Outlook – Limitations

The demonstration proves the feasibility of a solar-powered wireless CO$_2$ sensor, but is not optimized for energy management, e.g. will not survive the night (no recharging). To convert this demonstration to a final product still some effort needs to be spent. Some of the things that need to be considered in more detail are

- required measurement rate

- maybe dynamic adopting measurement rate to available energy (charge level) or day/night time (reduce measurement rate during night time, or when room is dark)

- size of solar cell and capacitors

- need for long term and short term storage including charge management

- synchronization to uart communication after wakeup (see **2.4**). This is currently performed by enabling the uart (uart_init()) at fixed time after the wakeup. It has not been evaluated/tested if this will work under all circumstances. An optional approach could be to monitor the first bit changes on the TX line and to enable the uart on detection of a specific "bit" pattern.