



EPFL
GRENOBLE-INP
POLITECNICO DI TORINO

INTERNATIONAL NANOTECH MASTER

A miniaturized acetone sensor for breath-analysis. Electronic design and characterization

From February 19th to August 17th
at
ams Sensors UK Limited
Deanland House - 160 Cowley Road
Cambridge (UK)

Author:
Joaquim LUQUE

ams supervisor:
Claudio ZIULANI

Grenoble-INP supervisor:
Quentin RAFHAY

*A thesis submitted in fulfillment of the requirements
for the Engineer's degree*

in the

Nanotech Master

September 2018

Abstract

The chemical analysis of the exhaled breath is an increasingly popular approach to detect many health conditions and body issues. The inexpensiveness, rapidity and non-invasiveness of this method contribute to its attractiveness. In this Master project report, we present the electronic design of a miniaturized in-breath acetone analyzer for weight-loss prediction purposes. The main feature of these electronics is the constant temperature controller implemented in the four sensors of the device. For a maximum accuracy, the sensing device is based on a combination of sensors: three metal oxides gas sensors and one flow sensor. In addition to the electronic design, different candidate materials for coating the metal oxide sensors were tested and their applicability in this project was assessed. Ultimately, the capability of the new electronics to drive a quad-sensor at constant temperature was demonstrated during a three hours gas test.

L'analyse de la composition chimique de l'air expiré pour diagnostiquer des problèmes de santé est une approche qui bénéficie d'une popularité grandissante. Cette attractivité est due aux faibles coûts, à la rapidité et au caractère non invasif associés à cette méthode. Dans ce rapport de projet de fin d'études nous présentons la conception électronique d'un détecteur d'acétone dans l'air expiré pour prédire une possible perte de poids. La principale caractéristique de ce système est le régulateur de température constante implanté dans les quatre capteurs de l'appareil. Pour une précision maximale, le dispositif de détection est basé sur une combinaison de plusieurs capteurs: trois détecteurs de gaz à oxydes métalliques et un anémomètre. En plus de la conception électronique, différents matériaux candidats pour le revêtement des capteurs à oxydes métalliques ont été testés et leur pertinence pour ce projet a été évaluée. Enfin, le bon fonctionnement du nouveau circuit électronique a été démontré lors d'un test de trois heures pendant lesquelles un multi-capteur a été opéré à température constante.

L'analisi chimica del respiro espirato è un approccio sempre più popolare per rilevare molte condizioni di salute e problemi corporei. L'economicità, la rapidità e la non invasività di questo metodo contribuiscono alla sua attrattiva. In questo rapporto del progetto Master presentiamo il design elettronico di un analizzatore di acetone miniaturizzato per scopi di previsione della perdita di peso. la caratteristica principale del design è il regolatore di temperatura costante implementato nei quattro sensori del dispositivo. Per una massima precisione, il dispositivo di rilevamento si basa su una combinazione di sensori: tre sensori di gas a ossidi metallici e un sensore di flusso. Oltre al design elettronico, sono stati testati diversi materiali candidati per il rivestimento dei sensori a ossidi metallici e la loro applicabilità in questo progetto è stata valutata. Finalmente, la capacità della nuova elettronica di operare un quad-sensore a temperatura costante è stata dimostrata durante un gas test di tre ore.

Contents

Abstract	i
1 Introduction	4
1.1 Cambridge CMOS Sensors (CCMOSS)	4
1.2 The ams corporate	4
1.3 Presentation of the Breath-Fit project	5
1.4 Outline	5
2 State of the art	6
2.1 Metal oxide sensing	6
2.1.1 Principle	6
2.1.2 Metal oxide sensors	7
2.2 Flow Sensor overview	8
2.3 Quad-sensor approach (Combo-chip)	9
3 Prior Work	10
3.1 Introduction	10
3.2 Hardware	10
3.2.1 Overview	10
3.2.2 Sensor board	11
3.2.3 Adaptor board	11
3.2.4 Main board	11
3.3 Firmware	13
3.3.1 Overview	13
3.3.2 Source code	13
3.4 Graphical User Interface	15
4 The constant temperature driving	17
4.1 Concept	17
4.2 Interests	18
4.3 Implementation	19
5 Temperature feedback loop electronics	20
5.1 Purpose and scope	20
5.2 Hardware	20
5.2.1 Overview	20
5.2.2 Sensor board	20
5.2.3 Adaptor board	21
5.2.4 Main board	22
5.2.5 Component placement and routing	24
5.2.6 Component choosing	25
5.2.7 Board manufacturing and assembly	26
5.3 Firmware	27

5.3.1	Overview	27
5.3.2	Minor changes	28
5.3.3	The feedback loop control	28
5.4	Graphical User Interface	31
5.4.1	Purpose and scope	31
5.4.2	Operation and description of the GUI	32
6	Gas tests and characterization	34
6.1	Purpose and scope	34
6.2	Sensor preparation	34
6.3	Gas Tests	35
6.3.1	Test 1: Material characterization	35
6.3.2	Test 2: Test of the new electronics	36
7	Project conclusion	41

List of Abbreviations

ADC	Analog to Digital Convertor
ams	Austrian Micro Systems
DAC	Digital to Analog Convertor
GUI	Graphical User Interface
MCU	Micro Controller Unit
MOS	Metal Oxide Semiconductor
MOx	Metal Oxide
op-amp	operational Amplifier
PWM	Pulse Width Modulation
VOC	Volatile Organic Compound

Chapter 1

Introduction

1.1 Cambridge CMOS Sensors (CCMOSS)

Before being acquired by ams in 2016, Cambridge CMOS Sensors (CCMOSS) was a spin-out from the Electrical Engineering Division of Cambridge University founded in 2008. This start-up company was specialized in micro hot-plate design and manufacturing for gas and infrared sensing. The main know-how of CCMOSS are the sensitive metal-oxide materials and designs of the sensors. As metal oxides are often sensitive to many different gases, most of the CCMOSS products targeted the market of air quality control. As one of the worldwide leaders in analog solutions and environmental sensing, ams acquired CCMOSS to extend its portfolio of products and technologies. CCMOSS was renamed ams Sensors UK after the acquisition.

1.2 The ams corporate

In 2018 ams is a company headquartered in Premstaetten (Austria), employing over 10 000 people with a revenue over 1 billion (EUR) per year. They design and manufacture integrated sensor solutions for applications requiring high accuracy and sensitivity at a lowest energy cost. ams sensors have been used in mobile phones, smart homes, industrial, automotive and medical applications. The company has more than 20 design centers worldwide and 4 production facilities in Singapore and Austria. The company has 3 main business lines:

- **Optical sensors:** with technologies such as ambient light sensing, UV sensing, proximity detection, gesture sensing, structured light and time of flight sensing that find use in mobile, health and automotive applications. Important customers for this type of products are Samsung, Apple, Oppo, Huawei, Sony...
- **Environmental and audio sensors:** with microphones, humidity, temperature, gas and pressure sensors based on capacitive, resistive and band-gap measurements. ams sensors UK belongs to this division
- **Image sensors:** most of the technology is a legacy of CMOSIS that was acquired by ams in 2015. They develop high resolution image sensors, line scanners based on CMOS sensing. Their products are used for scientific imaging, surveillance applications and camera & video applications.

1.3 Presentation of the Breath-Fit project

The Breath-Fit project started in October 2017 as a collaboration between ams, Flusso Limited (spin-off from Cambridge University specialized in flow sensors), Microsemi (US semiconductor company) and Cambridge University. The project is partly funded by the Technology programme Innovate UK. These programs are managed by UK Research and Innovation which is a semi-governmental funding agency that aims to promote partnerships between universities and companies in order to foster innovation and economic growth in the UK. The project runs for 24 months and the budget is approximately £0.5 M.

The scope of the Breath-Fit project is to develop an acetone breath analyzer for weight loss prediction. Several studies have shown the correlation between acetone concentration in breath and in blood [1]. This compound is produced in the body when carbohydrates are not available, i.e., fat-burning regime [2]. In addition, breath acetone levels are also associated to glucose-related diseases such as diabetes although this correlation has not been fully demonstrated [3]. In this project, the acetone detection relies on a quad-sensor (combo-chip) configuration: 3 MOx gas sensor for the acetone sensing and 1 flow sensor. This combination of sensors should improve the precision of the measurements. The project leverages on the developments of the ENS160, a new quad-sensor currently developed by ams.

The task repartition between the four entities is the following:

- **ams** for the development of the gas sensors (sensing materials and designs), the gas tests, the hardware and software to drive and readout the combo-chip, the algorithm and the GUI.
- **Flusso Limited** for the development of the flow sensor and the design of the combo-chip. The combo-chip shall be composed of three ams MOx gas sensors and one Flusso flow sensor.
- **Microsemi** for the device packaging and exploitation plan. They are also in charge of the dissemination and communication activities.
- **Cambridge University** for the design of the breath-analyzer housing, the characterization of the combo-chip and numerical simulations of the flow sensor.

1.4 Outline

The report will begin with a presentation of a state of the art on MOx sensors and flow sensors, then an indispensable description of the prior work, achieved before the beginning of my project. Will follow a summary of the constant temperature driving concept and its interests for the project. The main chapter will describe the new electronics together with the temperature control implementation i.e. the hardware design, the firmware writing and the development of a GUI. Lastly, various gas tests and MOx characterization results will be described, followed by the first gas test run with the new electronics.

Chapter 2

State of the art

2.1 Metal oxide sensing

2.1.1 Principle

The detection in metal oxide sensors is based on the change of the MOx resistivity after the material reacts with a target gas. The change in resistivity depends on the nature of the MOx (p- or n-type) and the nature of the gas (oxidizing or reducing). In practice, if the semiconductor is n-type, the conductivity will increase after reacting with a reducing gas and decrease after reacting with an oxidizing gas. The trend is the opposite in a p-type semiconductor, table 2.1 sums up these two behaviors. The sensitivity of the detection obviously depends on the chemical composition of the MOx but also on other parameters [4] such as :

- **The temperature:** some reactions will not occur if there is not enough thermal energy (insufficient activation energy). By controlling the temperature, it is possible to favour a reaction with particular gases. Typically, the sensitivity increases with the temperature until a given point and then starts decreasing. In addition, the temperature controls the energy levels of the MOx materials and thus the baseline resistivity.
- **The MOx grain size:** The reaction of the MOx with the gas creates a depletion or accumulation region at the MOx surface. If the particle size is smaller than this region, the conduction will be completely blocked (or enhanced) in this region. On the other hand if the particle size is much larger than this region, the conductivity in these particles will not change significantly. Consequently, the sensitivity generally increases if the particle size is small. [5]
- **Doping:** many noble metals such as Pt, Au, Pd or Ag enhance the reactions on gas surfaces. This fact can be explained by complex electronic and chemical mechanisms such as the modulation of the Schottky barrier by the dopant or the catalytic of the oxygen dissociation promoted by the metal [4].

Type	Oxidizing gas	Reducing gas
n-type	Resistance increases	Resistance decreases
p-type	Resistance decreases	Resistance increases

TABLE 2.1: MOx interaction principle [6]

2.1.2 Metal oxide sensors

SnO_2 , TiO_2 and WO_3 are among the most common metal oxides employed as chemiresistors and the typical operation range goes from 300 to 500 °C. However, the main limitation of the chemiresistors remains selectivity, i.e., the sensors respond to a broad range of gases and accurate quantification of single gases cannot be achieved [7]. Much research aims at exploring or tailoring metal oxides in order to increase the selectivity towards a particular gas. In this regard, for instance ZnO nanorods operating at 300 °C have shown to be very sensitive to ethanol [8]. In addition, carbon nanotubes on metal oxide particles [9] operating at 250 °C responded to NO_2 and CO. However, synthesis of materials as the above is often challenging and perhaps difficult to scale-up thus posing some concerns over commercial exploitation of these results. In addition, the screening of new materials is a costly and slow process although advances in the combinatorial evaluation of MOx materials have been reported and may represent a better approach. In this sense, combination of the output of multiple sensors may represent a quicker avenue for improving on the gas selectivity challenge.

Besides the MOx materials, the design of the micro-heater and the interdigitated electrodes (IDEs) is also important as they respectively heat up the MOx and measure its conductivity during operation. The heater is most of the time a metallic track (Platinum, Gold, Tungsten, Aluminum...) situated underneath or next to the metal oxide [10] [11]. Platinum and Tungsten have gained traction in the commercial environment thanks to their superior performances due to resistance to electro-migration effects and temperature reliability [12] [13]. For the sensing electrodes (IDEs), several configurations are possible [11]:

- **Two electrodes:** it is the most common arrangement, the electrodes measure the drop of potential across the MOx which is deposited between them. Most of the times the electrodes are interdigitated, comb-like shaped.
- **One electrode:** which acts also as the heater. The operation is based on the shunting of the MOx by a electrode whose resistance is perfectly known. It is commonly integrated in a wheat-stone bridge and the voltage-drop is determined [14]. In this configuration, the resistivity and geometry of the electrode are very important and should be perfectly controlled.
- **Gate electrode:** the MOx is deposited on the gate of a field effect transistor [15]. The modifications of the surface charge distribution after reaction with a specific gas acts like it would on the gate voltage of a classic MOS transistor. This configuration suits the current integration and miniaturization techniques.

The form factor of the chemiresistor gas sensors can be various. The most common are planar and cylindrical, or transistor shaped if the electrode is gated. In the planar geometry, the heater, the IDE and the MOx are vertically stacked. In the cylindrical one, the electrode is cylindrical, the MOx is deposited all around and the heater is inside the electrode. As most of the MOx sensors have integrated heaters, it is necessary to thermally insulate the sensing material. For this reason the sensor (heater + electrodes + MOx) is often on a membrane.

ams MOx sensors are planar and have an integrated micro-heater, the comb-shaped IDE is on top of it and the MOx is deposited on the IDE. In Figure 2.1 we see the top view (roughly 1mm x 1mm) and layer stack of a typical ams MOx sensor. Usually these sensors are integrated in a simple circuitry and the resistivity of the MOx is calculated with an ADC that reads the voltage across the IDE.

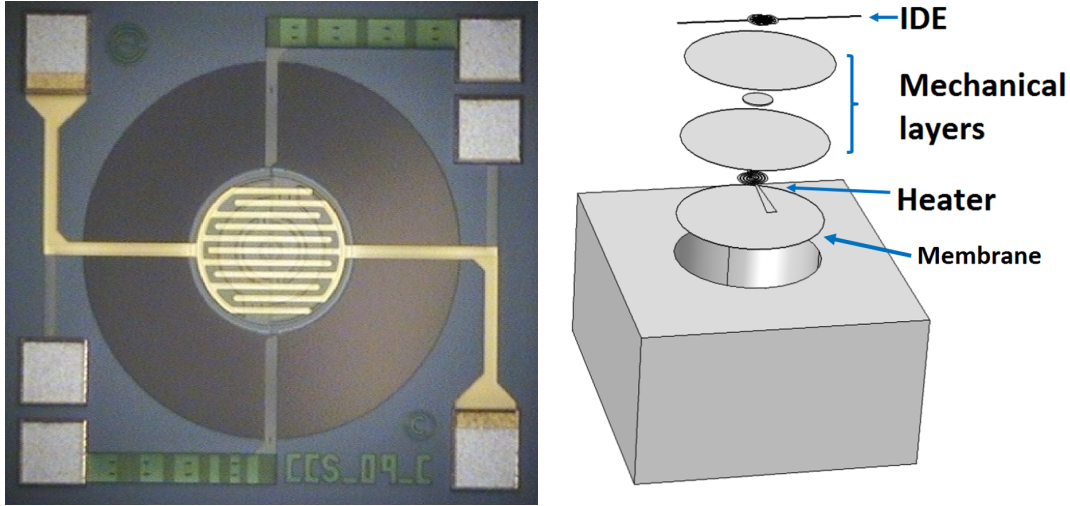


FIGURE 2.1: Bare ams MOx sensor(left), typical layer stack in planar MOx sensor(right)

2.2 Flow Sensor overview

In this project, the Flow sensor will be designed by Flusso and combined with 3 other MOx gas sensors in a single chip, the interests of this arrangement are explained in the next part.

Most of the flow sensors are based on thermal effects, the most common approach is the hot wire. Another approach can be the calorimetric flow sensor with one temperature sensor upstream, another downstream and a heating element between them. If all the variables of such a system are known, the flow velocity can be deduced from the difference: $\Delta T_{upstream} - \Delta T_{downstream}$ [16]. Another approach can be the time-of-flight flow sensors [17], they measure the transit time of a heat pulse. At least one upstream heater and one thermal sensor are needed. Non thermal flow sensors can be achieved with piezoelectric/piezoresistive cantilevers whose deflection is a function of the flow, or more generally with differential pressure sensors [17].

Thermal flow sensors are interesting as there is a known semi-empirical law, the King's law, that links the flows' velocity to the heat transfer of the hot wire. Having this kind of equation eases the calibration the flow sensor and the search of a fitting function.

The King's law states that:

$$h = a + b\sqrt{v_f} \quad (2.1)$$

h the heat transfer in $W \cdot m^{-2} \cdot K^{-1}$, v_f the velocity of the flow in $m \cdot s^{-1}$, a and b the King's law coefficients.

The first order relation between the resistance and temperature of the wire gives:

$$R_{wire} = R_{ref}(1 + \alpha(T_{wire} - T_{flow})) \quad (2.2)$$

R_{wire} in Ω , the resistance of the hot wire, T_{wire} the temperature of the hot wire, R_{ref} the wire's resistance at T_{ref} and α the temperature coefficient of the wire in K^{-1} .

The power equilibrium between the hot wire and the flow gives:

$$P_{elec} = P_{heat-transfer} \Leftrightarrow R_{wire} I_{wire}^2 = h \cdot A_{wire} (T_{wire} - T_{flow}) \quad (2.3)$$

If we inject (2.1) and (2.2) into (2.3) we have:

$$v_f = \left(\frac{1}{b} \left(\frac{R_w I^2}{A_w (T_w - T_f)} - a \right) \right)^2 \quad (2.4)$$

Depending on how the wire is driven (constant current, constant power, constant voltage or constant temperature), v_f will be calculated either by monitoring the current, the power consumption or the resistance of the filament.

The implementation of the constant temperature driving was an important part of the project and will be described in Chapter 4.

2.3 Quad-sensor approach (Combo-chip)

As mentioned, the detection in the Breath-Fit project is based on a quad-sensor, three MOx gas sensors and one flow sensor are integrated in a single chip. This configuration has two main interests:

1. MOx materials response to a lot of gases, if three relevant MOx are deposited on the each one of the gas sensors, it might be possible to isolate the acetone signal. For instance if one material is sensitive to ethanol and acetone and another only to ethanol, by differentiating both signals (assuming the sensitivity is the same for both materials and gases), we can obtain the acetone signal.
2. As mentioned, the temperature of the MOx has an important effect on the sensitivity, if the flow over the gas sensors is too important, the detection will be biased. With a flow sensor we'll be able to quantify the intensity of the flow and compensate its effect on the MOx resistance (algorithm needed).

In this combo-chip the MOx sensors will be similar to the one depicted in Figure 2.1. However, they will be smaller to keep an acceptable overall chip size (1mm²) and the pad arrangement will be reviewed.

The flow sensor of the combo-chip will be a hot wire. Two designs and driving ways were considered, the second one was kept.

1. A hot wire driven at constant current to measure $|V_f|$ with two small filaments on both sides of the hot wire to determine the direction of the flow.
2. A single hot wire driven at constant temperature without side filaments

Chapter 3

Prior Work

3.1 Introduction

Previous work on electronics was done by summer interns at ams during the last quart of 2017. The hardware and firmware to drive and readout the quad-sensor together with a GUI were prepared. This section summarizes the development of the electronics before the beginning of this internship. This phase was very important as extensive part of my internship project builds on top of the previous development work.

The purpose of the electronics is to drive and readout 4 combo-chips, so 16 sensors in total (12 MOx gas sensors and 4 flow sensor) and communicate with the GUI to plot the results. The key features the system should fulfill are listed in table 3.1.

Feature No	Feature description
F1	The system shall allow four main boards to be tested simultaneously.
F2	Each main board shall support monitoring of three MOx gas sensors, one flow sensor, one pressure-temperature sensor and one humidity-temperature sensor.
F3	The system shall be compatible with the existing gas test chamber hardware, where possible.
F4	The system shall support sensors packaged in DIL16 packages.
F5	The system shall support adjustable drive of sensors and heaters in DC drive mode independently.
F6	The system shall support both analogue and digital sensors.
F7	The system shall support acquisition frequency of at least 10 Hz.
F8	Control and acquisition system via USB.

TABLE 3.1: Key specifications of the system

3.2 Hardware

3.2.1 Overview

The hardware is the physical implementation of the application, it contains all the necessary components and a microcontroller on which the firmware will be written. For this one, many resources from the CCS EVK02 were used. Essentially the EVK02 is a development board developed by CCMOSS which supports driving and monitoring of 4 gas sensors. It supports PWM drive, as well as constant DC. It also supports pulsed mode and mobile phone connections. In the Breath-Fit project, only

DC modes are required to drive and monitor three gas sensors and a flow sensor. However, the heater and sensor voltages of each sensor need to be set independently.

For this application it was decided to split the hardware in three boards:

- **Sensor board** that connects 4 combo-chips and is placed into the chamber during the operation.
- **Adaptor board** that ensures the connection between 1 sensor board and 4 main boards.
- **Main board** that contains all the circuitry to drive and readout 1 combo-chip. 4 main boards are required to operate the 4 combo-chips of the sensor board.

Each one of these boards is described below.

3.2.2 Sensor board

This board is inserted into a chamber during the operation and contains all the sensors. It contains footprints for:

- 4 DIL16 packages, one per combo-chip
- ENS210, temperature and humidity sensor
- ENS220, not released yet but similar to the ENS210
- MS5637 pressure sensor.

At this point, the flow sensor was assumed to be a three filaments one (see part 2.3), the board was designed in line with it.

All the sensor signals are passed to the adaptor board through a 64 pins connector positioned at the edge of the sensor board. This board has the dimensions of the chamber in which it will be inserted during the test.

3.2.3 Adaptor board

This board connects 1 sensor board to 4 main boards. The analog signals coming from the sensor board are split in four. Each main board reads a single combo chip of the sensor board, the I2C signals are read by a single main board.

3.2.4 Main board

The main board contains all the circuitry to read and drive 1 combo-chip and the digital temperature, humidity and pressure sensor. The three main electronic blocs are: the heater's bloc (*3), the IDE sensing bloc (*3) and the flow sensor bloc (*1). The blocs dedicated to the gas sensor are implemented three times and the bloc for the flow sensor is implemented once. Their schematics can be seen in figures 3.1 to 3.3. There is a mini-USB port to connect a PC.

Heater's bloc

The heaters of the gas sensors are controlled by the electronics shown in figure 3.1. This circuit is designed to keep the heater at constant voltage. The demonstration is straightforward: the operational amplifier is operated in closed loop (output connected to inverting input), hence $V_+ = V_-$. V_+ is set by the external DAC which is

controlled by the MCU. We therefore have $V_{\text{heater}} = V_{\text{DAC}}$. This voltage will only change if the MCU feeds a new digital input to the DAC. The heater's resistance being very small, its power consumption is very high, the bipolar transistor ensures the current in the heater's branch is sufficient.

The second differential amplifier (gain ≈ 10) measures the voltage across the 1Ω resistance ($R_{1\Omega}$), the current in the heater's branch is calculated from this voltage. As the resistance in the branch of the heater is much smaller than the surrounding branches, it is reasonable to assume that $I_{R_{1\Omega}} = I_{\text{heater}}$.

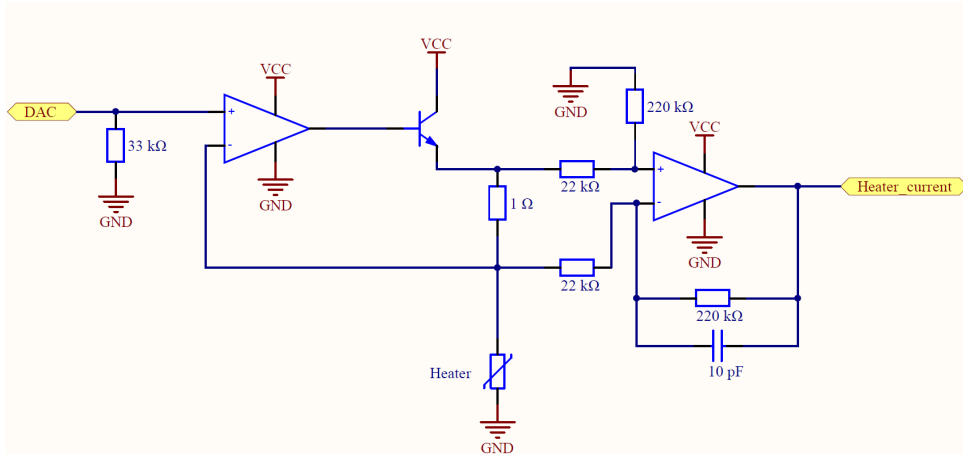


FIGURE 3.1: Heaters block of the first electronics

IDE sensing bloc

This bloc is responsible of the driving and reading of the IDE. The bloc sets the pull-up voltage of the potential divider with an external DAC and reads the voltage across the IDE with an ADC. The resistance of the MOx is deduced from this voltage. The schematic of the bloc is represented in Figure 3.2. Here, the buffers ensure the input impedance of the ADC is low enough for the voltage to be read. The low pass filter absorbs any high-frequency noise that could fake the measurement. It was chosen to set the potential across the potential divider with a DAC rather than with the board VCC for a higher precision and to be able to modify it from the GUI.

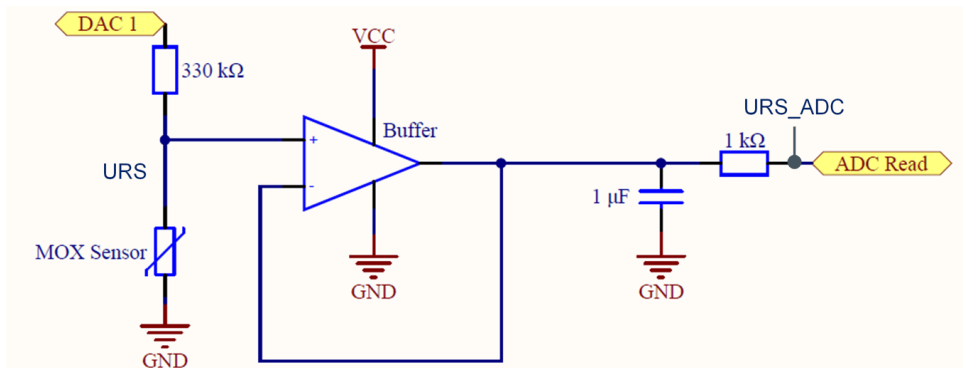


FIGURE 3.2: MOx sensors bloc

Flow sensor bloc

Initially, the flow sensor integrated in the combo-chip was supposed to consist of three individual micro-wires (one hot and two cold) in order to quantify the intensity and directionality of the flow, respectively. The side filaments are powered by two current sources. The main filament, sensitive to the flow intensity, it is constant current driven, see figure 3.3. The second amplifier (gain ≈ 1) measures the voltage across the flow sensor. The resistance of the flow sensor can be calculated since the voltage across the serial $40.2\ \Omega$ resistor is known (fixed by the DAC). Thus, after calibration, the resistance of the flow sensor is used to measure the flow velocity.

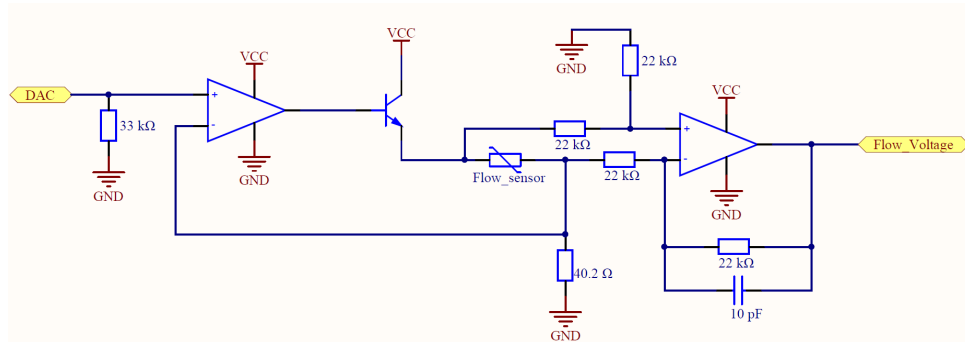


FIGURE 3.3: Flow sensor block of the first electronics

Microcontroller

For this board the MCU is an XMEGAA128A3U a 64-pins Atmel microcontroller, also used in the EVK02. It is responsible of the ADC readings, the driving of the external DACs, the I2C communication with the digital sensors and the USB communication with the GUI. It is connected to an external 12MHz oscillator to set the internal clock frequency. The writing of its firmware is an important part of the project and it is described below.

3.3 Firmware

3.3.1 Overview

The firmware was coded in C language with the Atmel Studio Framework (ASF). This means a lot of embedded software that has already been implemented (Memory drivers, USB and I2C drivers, ADC drivers, Watchdog timer, SPI services...). This embedded software provides a many low-level functions to ensure the basic operation of the MCU. Without it the MCU should be programmed from scratch, register by register.

3.3.2 Source code

The main part of the source code is a modified version of the EVK02 firmware. It consists in a multitude of functions and a main program that will be executed by the MCU when powered up. The principal functions are listed by module below:

- **Analog to Digital Converter (ADC):** these functions operate the two internal ADC of the MCU.

- *adcj_init*: sets the ADCs parameters: reference voltage, signed/unsigned mode, conversion triggering, clock rate and enables them.
- *adc_channel_event_setup*: there are 2 ADC in the MCU, each of them has 4 output channels, as there are more than 8 analog signals to read, the channel assignment must change according to the required measurement. When this function is called, the ADC channel assignment is updated.
- *adc_measure_internal*: measures one of the internal signals of the ADC (such as band-gap voltage, $V_{cc}/10$, temperature sensor...). This can be used to measure precisely the reference voltage of the ADC.
- Many functions to read in different ways one or both ADCs. Some of these functions read the result and then add it to the USB packet.
- **External DAC**: these functions operate the two external DACs of the main board
 - *DAC8554Init*: this function enables the synchronization of the data, loads the DACs registers and enables the SPI interface to the serial port.
 - *DAC8554SetChannel1* & *DAC8554SetChannel2*: these functions send the 24-bits coded data to one of the DACs. The 8 first bits of the data set which output channel will be updated (4 output channels per DAC, 8 in total) and what is the updating mode. The 16 following bits are the value of the output voltage.
- **Commands and parameters**: These functions define an extensive list of commands that can be requested by the GUI while the sensor is in its routine state (while periodically measuring and sending the sensors output). These commands can be used for example to request specific firmware parameters, or enable/disable portions of the main script.
- **ENS210**: these functions operate the digital ENS210 sensor. The communication is based on I2C protocols.
 - Many functions to initialize and write the registers of the sensor.
 - *ENS210_MeasureT*: Requests the temperature reading.
 - *ENS210_MeasureRH*: Requests the relative humidity reading.
- **Script**: Its main function is *script_do_tick*, it performs the measurements of the 4 sensors when the variable *newtick* is set to 1. The variable *newtick* is set to 1 by an independent timer every 10ms. As the measurements of the digital sensors are significantly longer than the others, they are read less often at a frequency defined by the GUI.
- **HID communication functions**: These functions write and send the HID reports that will be received by the GUI. Each HID report is composed of a fixed number of bytes.
 - *comm_putchar*: adds a 8-bits number to the report
 - *comm_puts*: adds an array of bytes to the report
 - *comm_send*: sends the report through USB.

The main program, executed by the microcontroller, does sequentially the following

1. Initializes all internal the sub-systems (such as clock, timers, ADC or USB communications)
2. Measures the board reference voltage (based on ADC internal measurement)
3. Initializes external systems (such as the external DAC or the ENS210)
4. Starts the USB communication
5. Enters the routine loop
 - (a) If $newtick = 1$ the scrip is executed
 - (b) If $newcommand = 1$ the specific command requested by the GUI is performed. $newcommand$ is automatically set to 1 when the MCU receives an HID report.
 - (c) Starts the loop again (until device is powered down or a shut-down command is received)

The flow diagram of this main program is represented in Figure 3.4

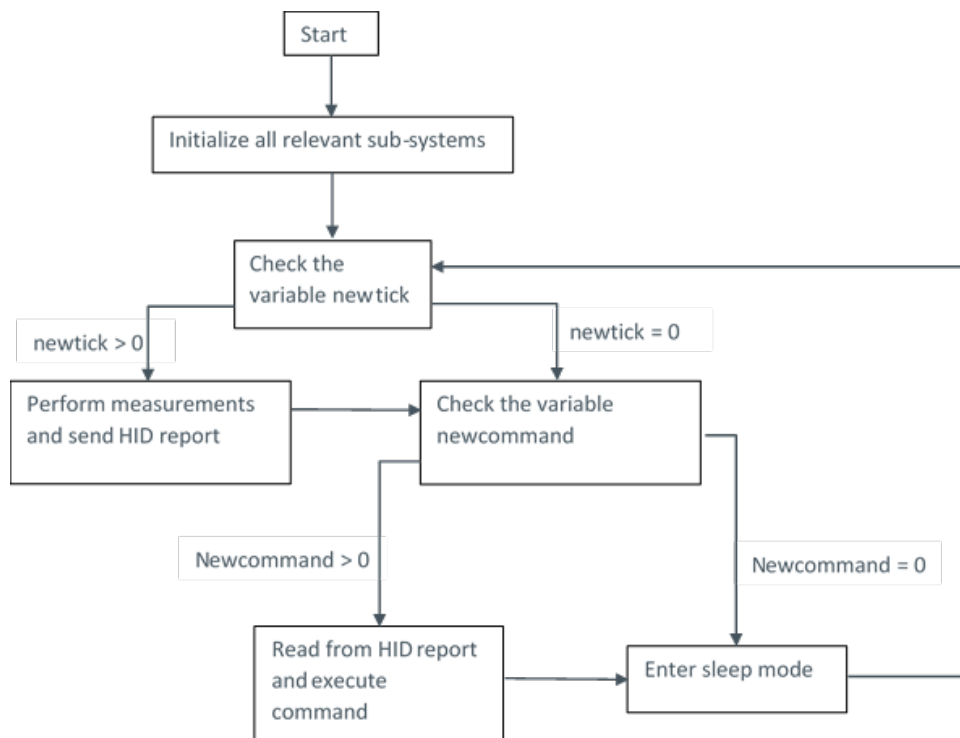


FIGURE 3.4: Flow digram of the main program

3.4 Graphical User Interface

The GUI is necessary to plot the output of the sensors and send the commands to the MCU. It was developed with Microsoft Visual Studio and it is a modified version of the public ENS Dashboard, an interface developed by ams for the company commercially available products. It is composed of two main windows, the first window displays the connected sensors and opens one of them upon double click. The second window (figure 3.5) displays all the graphs, some of them are hidden at

first but can be shown by setting the "view" options. The following charts plotted against time can be visualized:

Chart 1: Resistance of the 3 MOx

Chart 2: Resistance of the flow sensor

Chart 3: Voltage across the 3 IDE

Chart 4: Temperature and relative humidity (if the ENS210 is present)

The GUI also writes all the data in a .txt file for further analysis. These charts only display "raw data" (resistances, voltages and currents). The actual acetone concentration is estimated by calibrating the sensors in terms of response to several gases which may be present in breath.

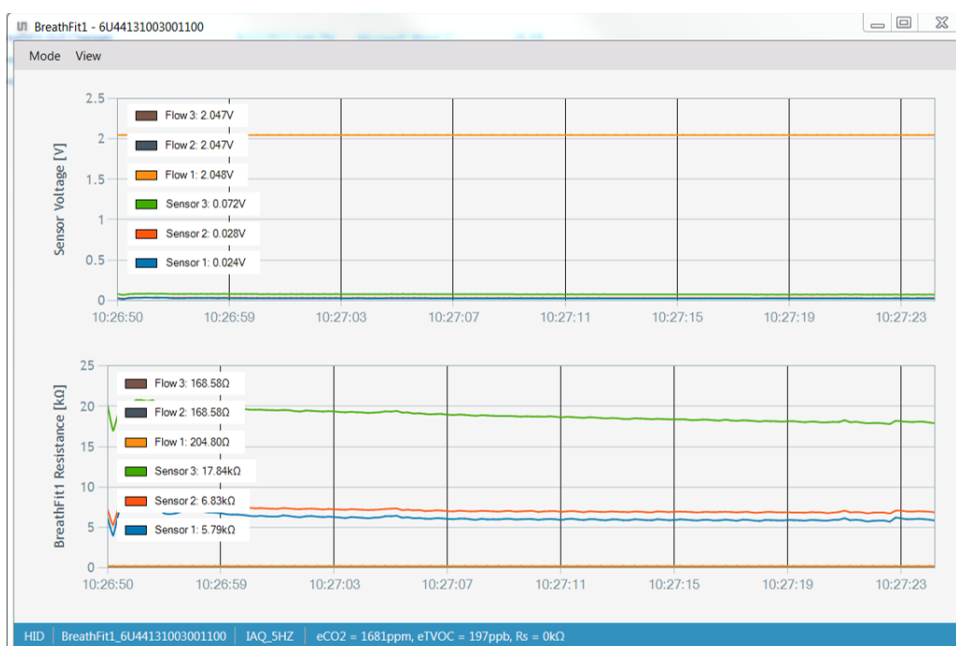


FIGURE 3.5: Charts Window 1st GUI

Chapter 4

The constant temperature driving

4.1 Concept

The constant temperature control was the main implementation of this new electronics, its operation is based on a feedback loop circuitry. The actual temperature of the component (a heater here) is measured and compared to a set-point, if there is a difference between these two values, the alimentation parameters of the heater will be re-adjusted to bring the temperature back to its set-point.

There are several ways to implement this feedback-control mechanism; either with hardware or software techniques. The advantage of the hardware techniques being that they are not limited by computational operations but by the physical characteristics of its components. On the other hand they are less tunable than software techniques and harder to calibrate.

Usually, the temperature is assessed by monitoring the resistance of the heater. This is particularly interesting as very basic electronic circuits can be used to calculate a resistance. The relation between temperature and resistance in (semi-)conductors is unique and given by the equation (2.2). Generally the temperature coefficient α is positive for the metals and negative for the ceramics.

One example of a hardware feedback loop control is a Wheatstone bridge with feedback branch (see Figure 4.1). Here $R_1 = R_2$ and the heating element is modeled by a resistance R_{heater} (in practice, most of the heaters exhibit a resistive behavior). The amplifier is in linear mode, hence it will try to keep its two input voltages equals:

$$V_+ = V_-$$

And

$$V_{out} = R_1 * I_1 + V_+ = R_2 * I_2 + V_-$$

Therefore, as $R_1 = R_2$:

$$I_1 = I_2$$

Leading to

$$R_{heater} = R_{var} \quad (4.1)$$

Therefore, the bridge will only be balanced when 4.1 is verified. To understand how the compensation mechanism works let's take the case where the temperature rises, assuming the heater is a metal with a positive temperature coefficient:

$T \uparrow \Rightarrow R_{heater} \uparrow \Rightarrow V_{heater} = R_{heater} \cdot \frac{V_{out}}{R_{heater} + R_1} \uparrow \Rightarrow V_{out} = A \cdot (V_+ - V_{heater}) \downarrow \Rightarrow P_{heater} = \frac{V_{heater}^2}{R_{heater}} \downarrow$ and the heat generated by joule effect decreases. When the temperature decreases, the exact complementary sequel of events takes place.

If the heater has a negative temperature coefficient it should be connected to the non

inverter input of the op-amp, the above deduction can be verified by changing the appropriated implications. Here R_{var} is the set-point, if it is increased the targeted temperature will increase and vice versa (assuming a metallic heater).

This kind of approach is particularly robust as it relies on basic components and simple mechanisms, however, as any hardware implementation it lacks of flexibility. For more sophisticated applications featuring many setpoints and response times, software implementations and tuning algorithms are required.

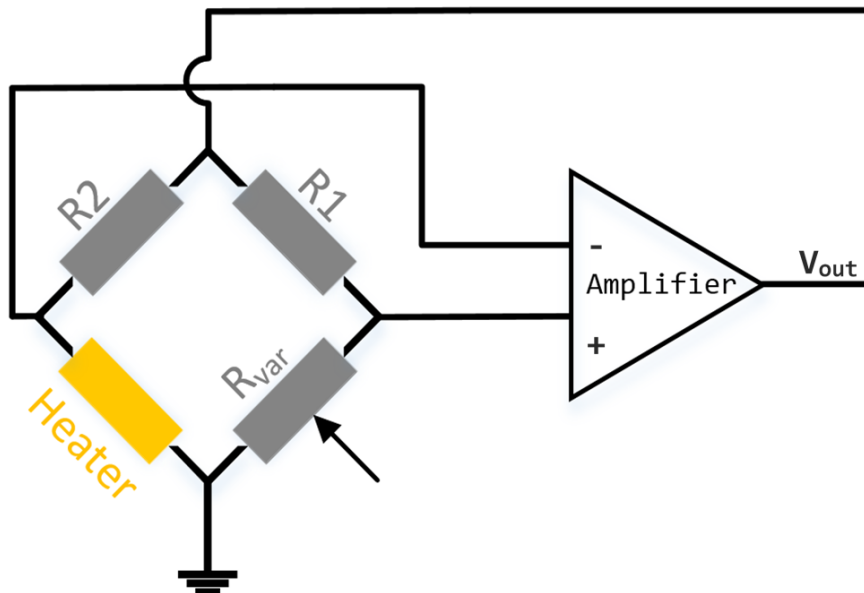


FIGURE 4.1: Constant temperature circuit with Wheatstone bridge

In the Breath-Fit project the temperature control will be software based. It will be used to control the temperature of the sensors' heaters and flow sensor. The interests of this new operation and its precise implementation are described below.

4.2 Interests

In the previous version, the flow-sensor was constant-current driven and the heaters constant-voltage driven, the interests of this new operation are summed-up in Figure 4.2. It is particularly interesting for the gas sensors as the MO_x detection is very dependent on the temperature, if the temperature of the sensing material is constant we remove an undesirable dependency.

The electronics here implemented will enable to monitor the intensity of the flow as a function of the power consumption. The stronger the flow the higher the power required to keep the flow-sensor at constant temperature, as demonstrated previously in equation (2.4).

In terms of project scope this modification will be useful to assess the interest of a dedicated flow sensor. Up to now flow-sensor and heaters were operated differently, now, by monitoring their respective power consumption we'll determine if the flow-sensor has a clear advantage with respect to the heaters in terms of flow velocity quantification. We expect the flow-sensor to be more sensitive to the flow variations as – on the contrary of the heaters – it is directly exposed to them. However if the power consumption of the heaters gives a good “enough” idea of the flow intensity,

the dedicated flow sensor will be removed and replaced by a classical MOx sensor with heater.

	Pros	Cons
Flow sensor Constant current drive (V1) ↓ Constant temperature drive (V2)	<ul style="list-style-type: none">• No thermal inertia<ul style="list-style-type: none">➢ Faster response time➢ Higher frequency response• Stability<ul style="list-style-type: none">➢ No burnout risk at low flow rates	<ul style="list-style-type: none">• Higher power consumption at high flow rates• Signal more noisy (due to feedback re-adjustment)
MOX Sensor Constant voltage drive (V1) ↓ Constant temperature drive (V2)	<ul style="list-style-type: none">• Sensor stability and accuracy<ul style="list-style-type: none">➢ Sensitivity towards a gas depends on temperature	<ul style="list-style-type: none">• Higher power consumption

FIGURE 4.2: Pros and cons of the constant temperature operation

4.3 Implementation

As mentioned before, the operation is software based. The heater is integrated in a potential divider, its resistance is deduced from the measurement of two voltages in this potential divider. The error between the set-point and the actual resistance is fed to a software-implemented PI controller which calculates a new potential across the potential divider. This potential, either higher or lower than the previous one increases or decreases the current going through the heater to re-establish the wanted temperature. This loop runs as frequently as possible to ensure smooth variations of the temperature. The precise process is explained in the Firmware section of Chapter 5.

Chapter 5

Temperature feedback loop electronics

5.1 Purpose and scope

This chapter describes the implementation of the temperature feedback loop electronics needed for the Breath-Fit project. In this context the electronics involve the hardware of the driving and reading circuitry, the software/firmware of the micro-controller and the graphic user interface. This second version was needed to enable the constant temperature driving of the sensors' heaters and the flow sensor. Furthermore, the partner Flusso changed the design of the flow sensor, the new design is single-filament.

Certain components of the hardware and firmware were taken from the existing solutions (previous Breath-Fit electronics and EVK02). The GUI was completely re-designed. Most of the internship was dedicated to the development of the electronics

5.2 Hardware

5.2.1 Overview

In this section, the design and implementation of the second version of the breath-fit hardware is detailed. The hardware was designed with Altium Designer, manufactured by Eurocircuits and assembled in house. It was designed to drive and readout four quad-sensors, the ENS210 and communicate with a GUI. Its key-features are the same as the previous version (see table 3.1), only F2 is slightly changed; only one digital sensor is included in the design. The implementation resembles the first version with a sensor board, an adaptor board and a main board. Each board is described below.

5.2.2 Sensor board

The sensor board has two layers of connectivity and contains four 16-pins sockets (DIL16 sockets) to attach the four combo-chips. The sockets are collinear with respect to the flow direction in the chamber and the spacing between them is minimized to avoid any perturbation in the laminar flow that could cause turbulences. The board is longer than the previous version to fit in a longer chamber. The digital sensors ENS220 and MS5637 were removed from the design. Only the ENS210 remains and will be attached to the roof of the chamber, it won't be on board as in the previous version.

The board was designed according to the bonding diagram shown in Figure 5.1, this sensor is not commercialized yet, so its design was hidden. Basically the die

is divided in 4 areas, one per sensor, the flow sensor was placed at the top - left position of the die (bottom - left in Figure 5.1). A 64 pins connector at the edge of the board ensures the attachment to the adaptor board. The final PCB layout of the sensor board is shown in Figure 5.2, the four footprints of the 16-pins package are designated by IC1-4 and the 64-pin connector is at the bottom edge. The white rectangular outline represents the chamber, the four holes are for the screws that will anchor the chamber to the board. The dimensions of the sensor board are 137 mm * 65 mm.

The flow enters at the right side, going from IC4 to IC1 and the extra room on this side ensures that a laminar flow is established before reaching the sensors. Turbulent flows are undesirable and affect the accuracy of the flow sensor measurements.

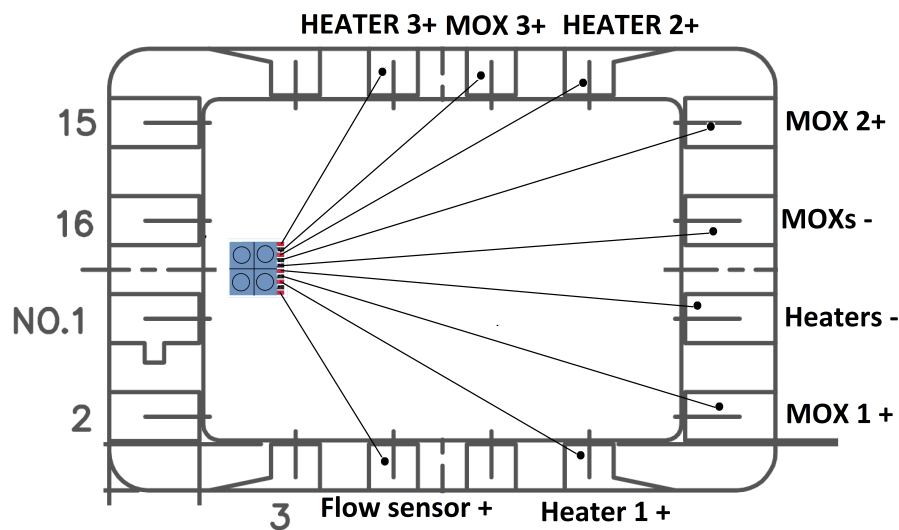


FIGURE 5.1: Bonding diagram of the combo-chip onto the 16-pins package

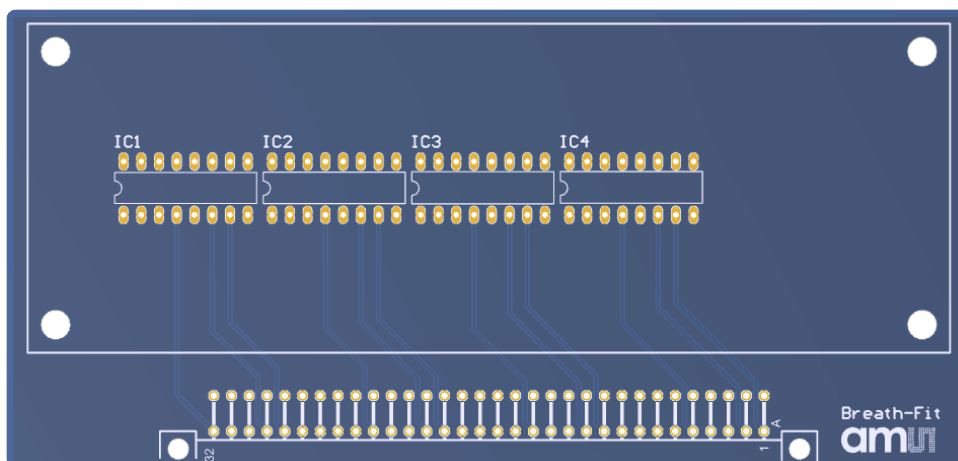


FIGURE 5.2: Top view of the sensor board PCB layout

5.2.3 Adaptor board

This board has two layers of connectivity and contains four 26-pins sockets to attach four main boards. Each main board drives one combo-chip. The firsts elements of

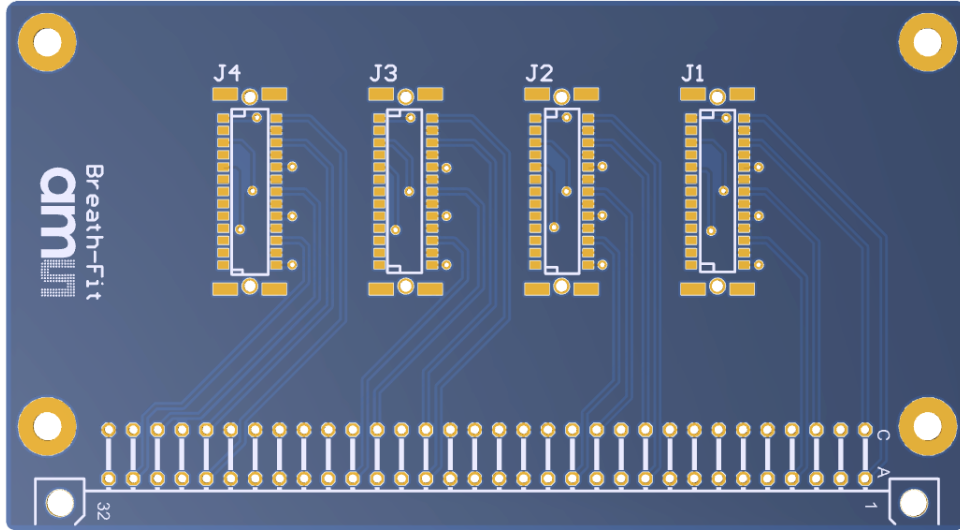


FIGURE 5.3: Top view of the adaptor board PCB layout

circuitry are on the bottom side of the board. For each gas sensor there is a potential divider necessary to read-out the resistance of the MOx layer (R_{MOx}). This potential divider is composed of R_{MOx} and a 330 k Ω fixed resistor. The value of this resistor is not arbitrary, it must be the closest to the value R_{MOx} to allow the most accurate reading. The value of R_{MOx} is calculated from the equation below:

$$R_{MOx} = 300 \cdot 10^3 \cdot \frac{V_{MOx}}{V_{DAC} - V_{MOx}}$$

V_{DAC} being the voltage across the potential divider, here set by an external DAC. V_{MOx} is read by an ADC in the main board.

The final PCB design of this board is shown in Figure 5.3, the four 26-pins connectors are designated by J1-4 and the 64-pins connector complementary to the one of the sensor-board is on the bottom edge.

5.2.4 Main board

Overview

This board has 4 levels of connectivity: two internal copper planes for VCC and GND and two signal layers (top and bottom). All the circuitry is on this board, besides the main components enumerated in Table 5.1, there are many passive devices such as resistors, capacitors, inductors and ferrites forming the necessary filters, potential dividers and protective elements of the design.

One main board can drive and readout 1 combo chip (3 gas sensors and 1 flow sensor). It is connected to the adaptor board with a 26 pins connector complementary to the one on the adaptor board.

Besides the main change which is the constant-temperature driving for the heaters and the flow sensor, two modifications were made with respect to the former version: micro-USB connection instead of mini-USB and higher reference voltage (2.5 V instead of 2.048V) for a larger range of ADC readings and DAC output values.

The PCB design of the main board is shown in Figure 5.4. The 26-pins connector J1 is on the bottom edge, the MCU UC1 on the top left and the USB connector J1 on

Component	Designator	Comment / Use
XMEGA128A3U MCU	UC1	Microcontroller, internal ADC, external DAC driving, I2C comm. ENS210, HID communication.
DAC8554, External DAC	U5 & U6	4 channels DAC. Powers MOx's IDE, heaters and flow sensor.
16MHz quartz oscillator	Q1	Clock reference for the MCU.
Voltage regulator 3.3V	IC5	Board voltage VCC.
Voltage reference 2.5V	U7	Voltage reference for ADCs and external DACs.
op-amp AD8456	U1 & U2 & U3 & U4	Double channel buffer for sensor signals.
Diode bridge	DN1	Diode bridge for USB power protection.
4 pins connector	P1	4 pins connector for I2C ENS210 Interface.
XMEGA PDI programmer	PDI	6 pins connector for the ATMEL - ICE programming interface.
Micro-USB connector	J1	Board powering and HID communication.
26 pins connector	J1	Connection to the adaptor board.

TABLE 5.1: Main board components

the top right. Note that two components were designated by J1, this is an error made during the design.

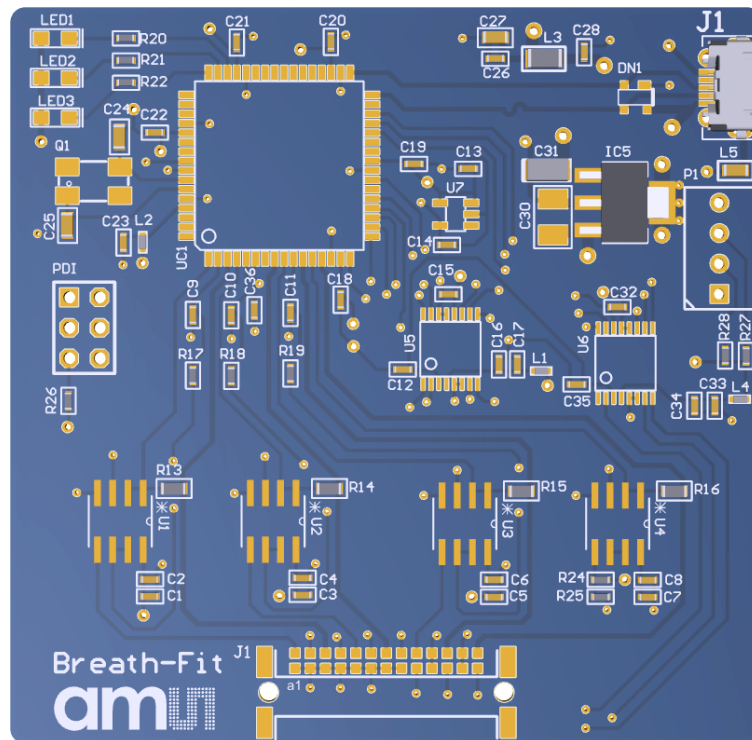


FIGURE 5.4: Top view Main board PCB layout

IDE Sensing bloc

This bloc was kept unchanged from the first version (see part 3.2.4). There are three of these blocs in the board, one for each MOx sensor. The three buffers before the ADC are op-amp AD8456 in follower mode (U1, U2 and U3), the low pass filters are formed by C9, C10, C11, R17, R18 and R11, the pull-up voltages of the three potential dividers are set by the DAC (U6).

Heaters and flow sensor bloc

The heaters block drives and reads all the heaters present in the combo chip. Its schematic is shown in Figure 5.5. The two ADC readings are used to calculate the resistance (hence temperature) of the heater. The MCU compares this resistance to the setpoint value and calculates the new DAC voltage necessary to the re-establishment of the right temperature.

There are four of these blocs in the main board, three for the sensors' heaters and one for the flow sensor. However the parameters of the PI controller are not the same for the flow sensor and for the heaters. The three buffers are op-amp AD8456 in follower mode (U1, U2, U3 and U4) and the DAC2 (U5) sets the pull-up voltages across the four potential dividers. As the resistance of the potential divider is very small (hundreds of ohms) the buffers are necessary to provide enough power to both resistors.

Theoretically the ADC reading of the voltage across the potential divider is not required as it is equal to DAC2, however as we cannot be completely sure of the accuracy of the DAC, we chose to base the calculation on the ADC reading.

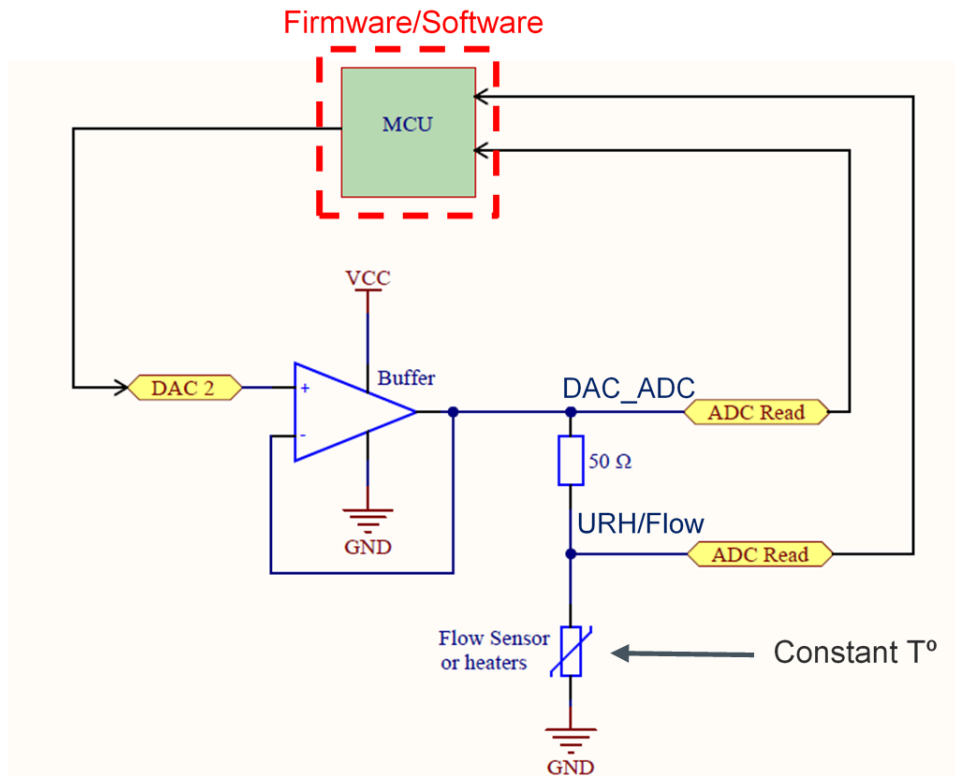


FIGURE 5.5: Heaters and flow sensor bloc

5.2.5 Component placement and routing

The routing and the component placement is the most demanding part in the design of a PCB. The sensor and adaptor board were straightforward to design, the main board required more time and a few iterations. There are digital and analog signals circulating in this PCB, they must be sorted out and prioritized. The analog signals from the sensors (flow sensor, heaters and IDE) must be routed in priority and kept clear from any noise. The 2.5V reference voltage also should be routed in priority,

this value is crucial for an accurate DAC and ADC operation. Also, a special attention must be paid to the routing of the differential USB pair D+ and D-, their length must be the same and their routing as clean as possible (no via nor proximity with other noisy signals)

In this board the main sources of noise are the digital signals: clock, USB data, DAC input, I2C signals... They must be kept away from the analog signals. In our application, noise in digital signals is not a problem (except for the USB differential pair mentioned above), it takes a strong noise source to reverse the value of a bit. Hence the strategy was to place and route the analog bloc first and arrange the digital bloc in the remaining space.

The analog bloc (buffers, potential dividers and low pass filters) was placed close to the 26-pins connector on the bottom part and the digital bloc on the upper part. The MCU can be seen as the bridge between analog and digital signals, it was oriented so that the ADC pins were close to the analog signals and USB and SPI pins close to the USB ports and DAC SPI ports. It is important to keep in mind that on contrary of the other components each MCU pin has not a single function. In fact, several modules of a MCU can be accessed through a single pin and different pins can access a same MCU module. It is advised to route the signals to the appropriated pin region of the MCU (reported in its datasheet) and adapt the firmware in consequence.

The reference voltage generator (U7) must be close to the MCU's ADC and external DACs and was therefore placed between them.

Another important consideration is the placement of the decoupling capacitors, they must be the closest to the power pins of the active components (buffer, DAC and MCU) in order to quickly provide power if the power supply temporarily drops its voltage. They also absorb the high-frequency noise in the power supply signal.

5.2.6 Component choosing

Several components were carefully chosen in order to meet the required specifications in terms of device accuracy or power delivery. There are no major constraints for the resistances and connectors of the sensor and adaptor board. However the following components of the main board should respect a few constraints.

- External DAC: Should have a high precision as the constant temperature control depends on them. The DACs should be multi-channel as they set seven voltages in the board (three pull-up voltages for the IDE and four heaters voltages). The DAC8554 has four output channels and a 16 bits precision.
- Operational amplifiers: We chose to have the same op-amp for the MOx sensors bloc and for heaters bloc, hence two main conditions had to be fulfilled. The input bias current must be very small to prevent any distortion of the MOx resistance reading. I_{bias} of the AD8656 is maximum 10 pA. This value is five order of magnitude smaller than the current circulating in IDE's branch (typically $\frac{2V}{1M\Omega} = 2\mu A$).
The headroom should be small enough to enable the heater to reach its maximum temperature. The headroom is the difference between V_{CC} and $V_{\text{out max}}$. The smaller the load the bigger the headroom. To calculate the maximum headroom we need the specifications of the heaters. The maximum temperature is reached when $I = 17\text{mA}$, this corresponds to $R_{\text{Heater}} = 90\ \Omega$, hence $V_{\text{out max}} = (90 + 50) * I_{\text{max}} = 2.4\text{ V}$. As $V_{CC} = 3.3\text{V}$, the maximum headroom is

0.7V at 17mA. The headroom curve of the AD8656 shows that at 17mA, $V_{OH} \approx 100\text{mV}$.

- Potential divider resistors: To maximize the accuracy of the ADC readings, the value of the pull-up resistors of the potential dividers should be perfectly known. For these specific resistors a tolerance of 0.1% was chosen.
- Microcontroller: It should have enough ADC channels to read simultaneously enough signals. Here eight signals can be read at the same time. It should also be quick enough to perform and send ten measurements per second. The operation frequency of the XMEGA128 is 32MHz.
- External oscillator: The MCU will set its internal clock according to the crystal frequency, this frequency can range from 0.4 MHz to 16 Mhz. The chosen crystal oscillates at 16 MHz.

For the other constraints, the datasheet of each component should be consulted.

Regarding the size of the components, most of the active components have a fixed size. However all the passive components are available in different sizes, for a compact design it is preferable to take the smallest one. The drawback of this strategy is the difficulty of the soldering during the assembly. In the main board most of the passive components were [04 02] sized (roughly 1mm* 0.5mm), it is not easy task to solder them.

5.2.7 Board manufacturing and assembly

Before the manufacturing, the gerber and drill files of the design must be created. The drill files list all the holes in the PCB (vias for electric signals and mechanic holes). The gerber files are intermediate manufacturing files that provide a complete description of each PCB layer (top overlay, copper layers, bottom overlay ...). The layer stack of each PCB can be seen in Figure 5.6. This figure shows the two connectivity layers for the sensor and adaptor board and the four connectivity layers for the main board. Before fabrication, it is also important to ensure that the manufacturer guidelines such as minimum clearance, via size, routing width are followed. These guidelines can usually be found on the manufacturer's website. In our case the Altium design rules of the boards were already set to these guidelines.

To assemble the board we first soldered the Surface Mounted Devices (SMD) and then the through Hole components. Only the main and adaptor board have SMD and required stencils. The process to solder SMD is described below:

1. Dispose the stencil on top of the board ensuring that the holes are in front of the SMD pads.
2. Apply the solder paste on the stencil and carefully remove it.
3. Arrange the SMD components on their pads (now covered by solder paste).
4. Put the board in a oven to reflow the paste. After this short process the components should be fixed to the board.
5. Check the paste has properly attached the pins and there are no bridges (short circuits) between close pins.

The through hole components should be placed in their holes and soldered with a soldering iron and a soldering wire.

Once the assembly is completed and all the components connected to their pads, the first verifications can start. The integrity of the most important signals was checked with a multimeter in continuity mode. For instance, it was ensured that VCC, GND and the USB power were not shorted. Then it was verified that the MCU was programmable with Atmel studio and basic functions were achieved, e.g., ADC readings, LED blinking, USB communication. The most frequent problems are due to shorts between consecutive MCU pins or clock issues. The phase-locked loop of the clock had to be properly set according to the frequency of the external oscillator. The USB communication is done at 48MHz, if the clock has issues the communication will be impossible.

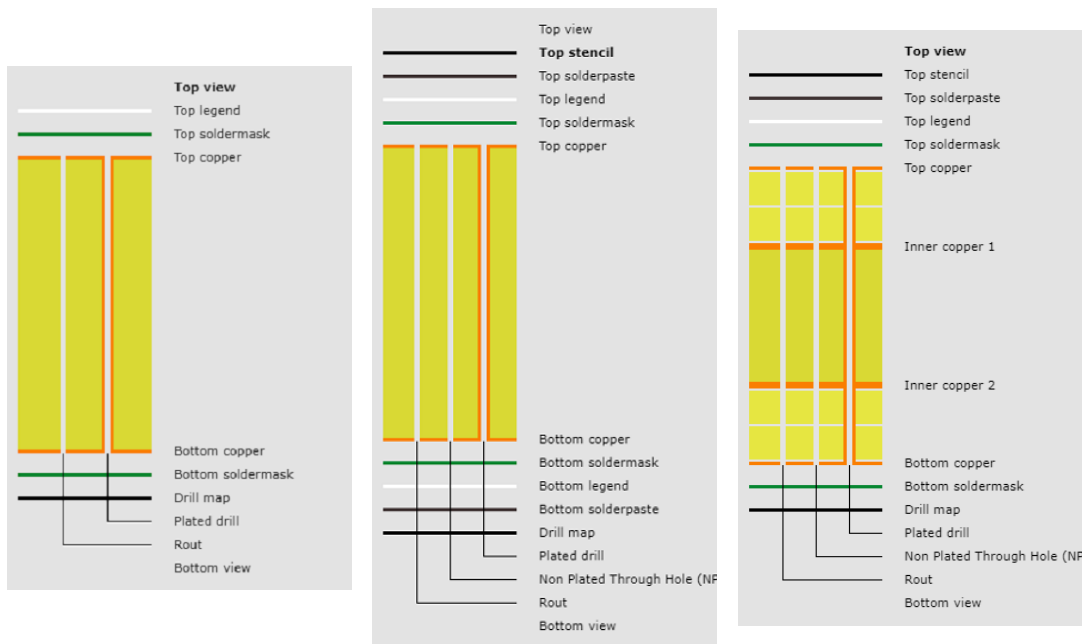


FIGURE 5.6: Layer stack (from left to right): sensor board, adaptor board, main board

5.3 Firmware

5.3.1 Overview

A firmware is an embedded software that is written on a non-volatile memory of a microcontroller. This software is meant to provide a low-level control for the device's hardware. Typically, a firmware can manipulate data, execute pre-defined functions, store data.

In this application the firmware has to drive and readout the gas sensors (IDE + heater), the ENS210 and the flow sensor and communicate with the GUI. The firmware is written on the flash memory of the main board's MCU, the XMEGA128A3U. The firmware was developed with Atmel Studio, a very practical tool that provides many libraries and embedded resources to operate Atmel microcontrollers. The firmware was not written from scratch, several functions enabling many operations already existed in the previous version of the breath-fit firmware:

- ADC operations (initializations and readings).

- External DAC operations.
- USB HID communication.
- ENS210 I2C communication.

And many other functions necessary to the good working of a microcontroller.

5.3.2 Minor changes

Besides the main change which is the implementation of the feedback loop control, a few modifications were made in the source code.

- ADC: the new channel assignment according to each measurement mode is described in Table 5.2. The meaning of each voltage can be seen in Figures 3.2 and 5.5. Also, the ADC is now operated in unsigned mode as there are no negative voltages to measure.

ADC Channel	Measurement mode 1	Measurement mode 2
ADCA Ch0	URS_ADC_1	DAC_ADC_1
ADCA Ch1	URS_ADC_2	DAC_ADC_2
ADCA Ch2	URS_ADC_3	DAC_ADC_3
ADCA Ch3	URH_1	URH_1
ADCB Ch0	/	DAC_ADC_4 (flow sensor)
ADCB Ch1	URH_2	URH_2
ADCB Ch2	URH_3	URH_3
ADCB Ch3	Flow	Flow

TABLE 5.2: ADC channel assignment

- 2 bytes communication function: The USB communication is based on exchanges of 8 bits packets. Most of the information to be sent through USB is 16 bits coded, hence the 16 bits must be split into 2 bytes before sending. The new function *comm_put16bits* splits the 16 bits numbers and writes them into the HID report. The most significant bits are written first (therefore received first) and then the least significant bits.
- Clock setting: The frequency of the external oscillator changed from 12MHz to 16MHz, hence in order to reach the 48MHz necessary to the USB communication the phase-locked loop clock multiplier must be changed from 4 to 3 in the source code.
- Commands and parameters: the possibility of requesting specific data during the main sensor routine such as calibration information or variable value has been removed. This previous implementation was a feature of the EVK02 but had no use in this application and allowed a simplified GUI.

5.3.3 The feedback loop control

The main implementation in this second version is the feedback temperature control, meant to keep the flow sensor and the heaters of the gas sensors at constant temperature. The function *set_heaters* ensures this task and its operation is given below:

For each heater/flow sensor:

1. The measurement mode is set to 1 (see Table 5.2), the two voltages URH/Flow and DAC_ADC are read by the MCU internal ADC and transformed into a resistance by the function *get_heat_resistance*.
2. The difference between the targeted value (set-point chose by the user with the GUI) and actual resistance is assessed (error calculation $e(t) = \text{setpoint} - \text{current value}$).
3. The error is fed to a PI (proportional integral) controller that calculates the new voltage required to re-set the heater/flow-sensor at its required temperature, this new voltage is then fed to the DAC that will re-adjust DAC_ADC to the calculated value. The global formula of the PI controller is the following:

$$V(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) d\tau + V_0$$

$V(t)$ here is the voltage across the potential divider (DAC_ADC), it is set by DAC2.

K_p is the proportional term that accounts for the current error, if the value of the error is significant and positive, the output voltage will be proportionally increased (or vice-versa).

K_i is the integral term that accounts for the past values of the error, if the error has been the same for several iterations the output voltage will be increased proportionally to this error sum.

V_0 is the voltage that keeps the temperature at its setpoint when there is no flow. This term is necessary to have a stable behavior even when the error has been zero since the beginning. Without it the voltage would drop to zero if there was be no error. This would of course increase the error and increase the voltage but it's better to avoid this situation and have a PI control strictly depending on the flow intensity and not on internal instability.

As the concept of mathematical integration is hardly definable at this level, the approach here is to use a discrete sum. At each iteration of the feedback control, the value of the error will be added to the sum of the previous errors. The voltage the formula becomes:

$$V(t) = K_p \cdot e(t) + K_i \sum_{i=0}^t e(i) + V_0$$

The PI controller needs a very precise calibration of its parameters (proportional gain K_i , integral gain K_p and windup) to minimize the overshoot while keeping a sufficiently small settling time. A windup limit is useful in case the error has been important during a long time, in this situation the integral correction will have an excessively important weight in the voltage correction. This over correction will continue to affect $V(t)$ even if the current error is very small. This windup limit will set a maximum value of the error sum $\sum e(i)$.

Board operation

Once the board is powered-up the firmware stored in the flash starts its execution. The MCU will sequentially perform the following operations:

1. Initialization of all the relevant sub-systems such as ADCs, clock, USB communication, external DACs, pins or registers.
2. Internal ADC measurement to determine the V_{ref} detected by the MCU (theoretically 2.5V, in practice V_{ref} detected $\approx 2.49V$).
3. Awaiting the GUI to send a number of important parameters (IDE pull-up voltages, heaters and flow-sensor setpoints, parameters of the PI controller and acquirement frequency).
4. Set or store these parameters and send V_{ref} measured to the GUI.
5. Entering the routine loop until powering-down of the device:
 - (a) Calculates the resistance of the heaters and flow sensor, deduces the required voltage to keep them at constant T° and feeds this new value to the DACs (feedback temperature control described above).
 - (b) Measures all the voltages necessary to the plotting of the MO_x resistances, heaters resistance, flow sensor resistance and power consumptions.
 - (c) Sends HID report with all the measured values and ENS210 data to the GUI.
 - (d) Returns to the beginning of the loop.

The flow diagram of the firmware is shown in Figure 5.7. One iteration lasts approximatively 340ms, the limiting measurement is the one of the ENS210 which takes about 270ms. In order to reach the minimum frequency established at 10Hz, the ENS210 is only read once every ten measurements, in this way ten measurements can be performed in 1s (with one measurement significantly longer than the others).

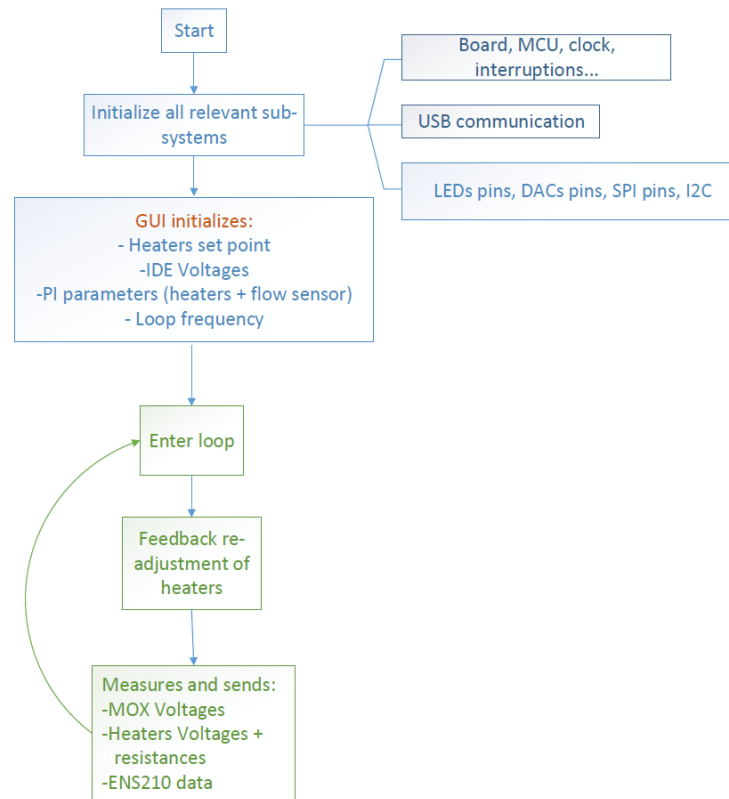


FIGURE 5.7: Flow diagram of the firmware operation

5.4 Graphical User Interface

5.4.1 Purpose and scope

The GUI was developed with Visual Studio in c# and uses many HID class functions of the .NET framework. Its purpose is to exchange data with the MCU of the main board through the USB port and plot in real time the sensors output. On the contrary of the previous GUI this one is not based on the ENS Dashboard (the ams GUI compatible with all the environmental sensors of ams). Concretely the GUI should fulfill four functions:

1. Send the initialization data to the MCU.
2. Receive and store the data sent by the MCU.
3. Plot in real time 4 charts.

Chart 1: 3 MOx layers resistances

Chart 2: Resistances of the flow sensor and 3 heaters

Chart 3: Power consumption of the flow sensor and 3 heaters

Chart 4: Temperature and humidity (ENS210)

4. write all the data in a log file for further analysis.

The communication between the MCU and the GUI is achieved through HID protocols, which are based on interruptions. At each interruption, a 64 bytes long report is send or received.

5.4.2 Operation and description of the GUI

When the GUI is executed a first window appears (see Figure 5.8): the Initialization window, several actions are possible:

- **Find my device:** The PC will try to find a HID device whose IDs match the vendor ID and product ID of the GUI. These IDs are written in the firmware of the MCU, the Vendor ID of our device is 0x3EB and the product ID is 0x2402.
- **Initialize:** Once the initialization parameters filled in the text boxes and “Initialize” pressed, the GUI will send the data to the MCU. Before sending the initialization parameters, the GUI formats IDE_1 , IDE_2 , IDE_3 so that the MCU won’t have to calculate the required 16 bits number that must be fed to the DAC. These three voltages represent the pull-up voltages of the MOx potential divider. It is important to make all the possible computations on the PC rather than on the MCU which has less computational resources. If the MCU is not in the initialization state the GUI does not receive the confirmation from the MCU and an error is displayed. Once the initialization done, the reference voltage measured by the MCU is written in the V_{ref} text box.
- **Get data:** The GUI enters the routine state and a new window will be opened (See Figure 5.9). All the data sent by the MCU will be plotted in the charts. Parallel threading had to be introduced in this process. While the GUI is waiting for the MCU to send the data it cannot keep the two windows responsive, therefore two threads are needed. The first thread controls the receiving of the data from the MCU and stores it and the second keeps the User Interface active and refreshes the charts when new data is available (UI thread). This button will also trigger the writing of the data in a file named after the Data name text box.
- **Stop charts:** the chart window will be closed but the data is still written in the log file. This function is interesting for long tests during which the live charts are not really useful but only the stored data is. Besides, most of the crashes of the GUI are due to the double threading since the chart plotting is very demanding in the UI thread. The GUI is more stable without this process. After stopping the charts the button will change to “Plot charts”, by clicking on it the live data will be plotted again in the charts window.
- **Refresh graphs:** If the charts stop refreshing one can click this button.
- **Restart:** The GUI closes and opens again.

The data is saved in a .txt file, the header and data format must follow some guidelines to be compatible with the ams Matlab scripts. The writing of the data is done in the non-UI thread, right after the GUI receives the HID report from the MCU. This is the only task performed in this thread between the reception of each HID report. This in-between work cannot be too long or the GUI will miss the arrival of the report and the data will start to accumulate.

The plots we see in Figure 5.9 were obtained with an ams quad-MOx gas sensor, we can see how the resistances and power consumption of the heaters increase during the first seconds of operation and how they quickly stabilize to reach the set-points defined in the Initialization window (Figure 5.8). This shows that, at least for the initialization, the temperature control works properly.

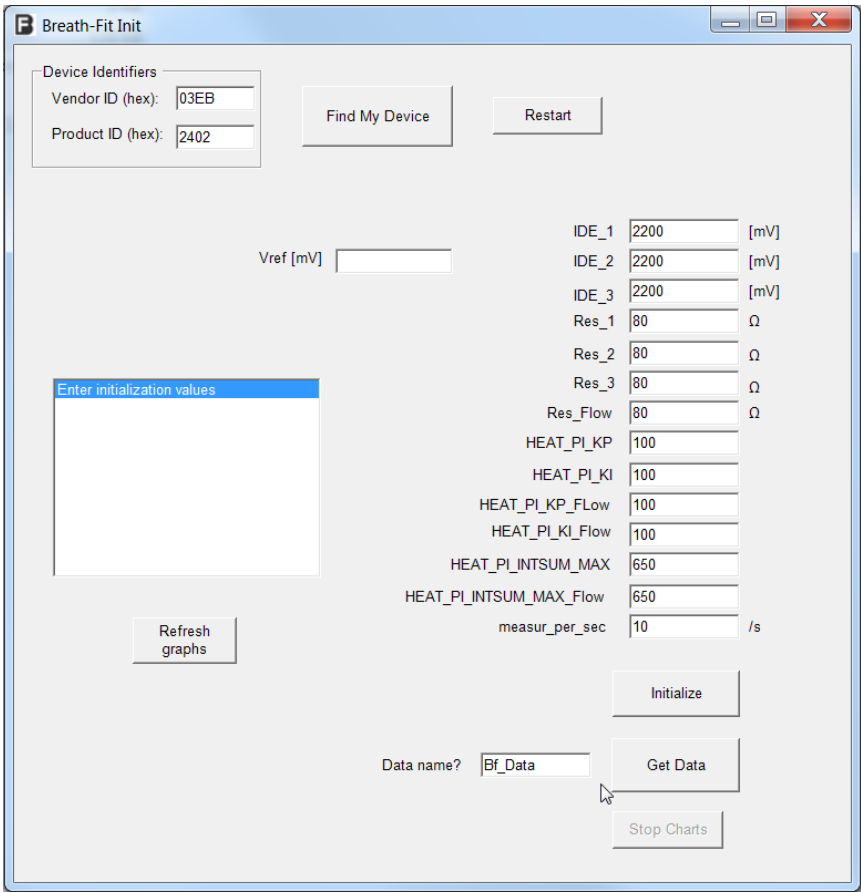


FIGURE 5.8: GUI Initialization window

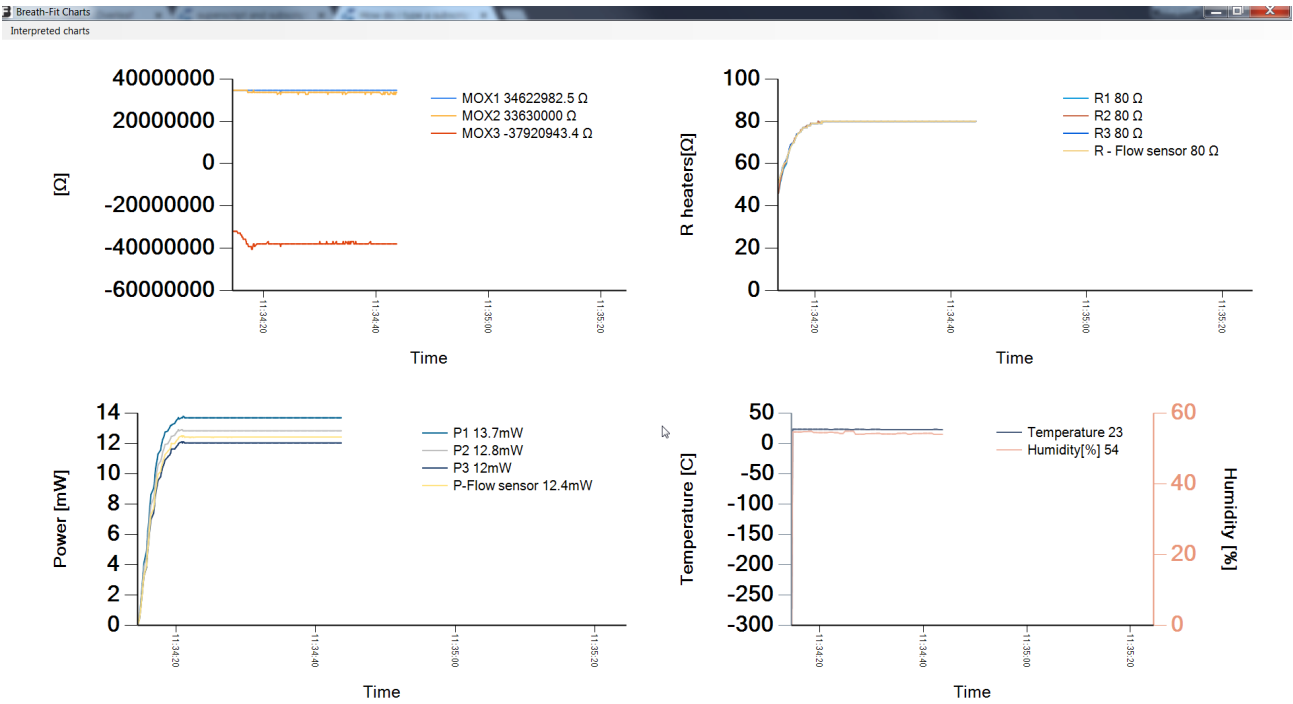


FIGURE 5.9: GUI charts window

Chapter 6

Gas tests and characterization

6.1 Purpose and scope

During the last part of the project several MOx materials were characterized and the response of the MOx towards several gases was tested. The purpose of these tests is to find the most suitable materials for coating the 3 MOX sensors of the quad-sensor. The main criteria to assess the suitability of the materials are: gas sensitivity, cross-sensitivity, resistance range, reproducibility, baseline drift and operational temperature. Perhaps, cross-sensitivity remains one of the most challenging aspect of MOx gas sensors which can be in part overcome by extrapolating the concentration of the gas of interest by combining the response from multiple gas sensors. For instance, the responses from a sensor sensitive to acetone and ethanol and another one specifically sensitive to ethanol, can be combined in order to determine the acetone concentration by subtraction.

We also carried out some gas tests to assess the impact of packaging processes on the MOX and membrane integrity. For instance, testing the impact of epoxy glues dispensed in proximity of the sensing film or the exposure to UV light which is used for curing these resins.

We also run one short gas test with the new electronics. The purpose of this test was to check if the new hardware, firmware, GUI and temperature control were functional and could be used for a gas test.

6.2 Sensor preparation

Before the gas test is run the parts must be prepared. The first step is the preparation of the MOx paste which is then deposited on the sensor, annealed and conditioned, e.g., micro-heater operated for a certain time, often 48 hours before testing.

1. **MOx paste:** The MOx paste is prepared by mixing the powder with a solvent. Additional binders can be added in order to achieve the right viscosity for the deposition.
2. **MOx deposition:** The paste is deposited on open-cap packaged parts. Deposition conditions depend on the past rheology. After deposition, part are baked in order to remove solvent traces. Each part is observed under microscope and the faulty ones are trashed. The most common defects are: MOX not deposited on IDE, MOx film deposited beyond the edges of the membrane, the sensor is not functional due to failures such as broken membrane and defective wire bonding.
3. **Annealing:** For approximatively 1 hour, the heater of the sensor is turned-on and its voltage is tuned so that its temperature follows a specific profile.



FIGURE 6.1: MOx dispensing system

Usually this profile starts with a slow ramp-up, then a long plateau at constant temperature, a quick period at very high temperature (400-500°C) and a ramp-down. This step is crucial to achieve the desired sensitivity for the sensor.

4. **Burn-in:** This last step is the longest, for approximatively one week the heater is kept at the operational temperature (typically 250-400°C) in order to stabilize the electrical baseline.

At the end of this process the sensors are ready and can undergo a gas test.

6.3 Gas Tests

6.3.1 Test 1: Material characterization

For the first gas test we tested four Metal Oxides at two different temperatures. These tests were run with single-membrane sensors. Even if the Breath-Fit sensor is a quad-sensor, for this test we were comparing materials, not sensors, that's why we used the simplest sensors, easier to handle. For statistical significance purposes we tested 4 identical parts per series, hence $4 \times 2 \times 4 = 32$ sensors (4 MOx, 2 temperatures, 4 devices per series). The four metal oxides and the gases that we were targeting with each of them are:

- MOx 1: Cr based metal oxide composite, tested at 320°C and 370°C, meant for ethanol detection.
- MOx 2: Sn based metal oxide composite, tested at 320°C and 370°C, meant for acetone detection although ethanol cross-sensitivity expected.
- MOx 3: Pd based metal Oxide, tested at 320°C and 370°C, meant for VOCs detection.
- MOx 4: Al based metal Oxide composite tested at 200°C and 235°C, meant for NO₂ detection.

The gas sequence, described in Table 6.1, is composed of 6 gases injected at two different relative humidities, first at 50% then at 80%. Between each gas, air is injected so that the sensor can retrieve its baseline.

The sensors are placed into sealed chambers with an inlet and outlet for the gas injection. Each chamber can fit 16 sensors, we used two chambers connected in "series" to test the 32 sensors. The variations of the MOx resistance during the test are plotted in Figures 6.2 to 6.5. A green point indicates the injection of a new gas and a red one indicates the end of the injection. The shaded areas represent the time during which the sensor was exposed to the gas. There are not always 4 curves per chart as some sensors failed the test.

Each material was assessed in order to determine: the sensitivity to the targeted gas, the cross-sensitivity to other gases and the trend with the operational temperature.

The first observation is the decrease of resistance with the increasing temperature (roughly 1 order of magnitude less for 50°C more), that was expected as most of the semiconductors have a negative temperature coefficient. In figure 6.2 we see that MOx 1 is sensitive to ethanol and NO₂. However its sensitivity to ethanol is not high enough and it was decided that parts should be re-tested at a higher temperature since the increase of temperature from 320°C to 370°C produces larger sensitivity towards ethanol and reduces cross-sensitivity towards NO₂.

The results of MOx 2 in figure 6.3 also suggest that this material should be operated at higher temperature. At 370°C there is a very small response to ethanol (predicting the appearance to the acetone response in the same range of temperature) and the NO₂ sensitivity decreases, we expect an enhancement of these two behaviors at high temperature.

The MOx 3 is widely used on sensors monitoring air quality, we see in figure 6.4 that it reacts with all the gases without any specificity. This material is good to take in account other VOC which may be present in breath apart from acetone and ethanol.

For MOx 4 we only have results at 200°C, all the parts used at 235°C failed. However it shows a very good sensitivity to NO₂ and some small cross-sensitivity to CO. This material is thought to be critical to assess the presence of NO₂ in breath which may occur under certain circumstances, e.g., lung inflammation.

6.3.2 Test 2: Test of the new electronics

For this first gas test with the new electronics we tested two types of sensors: 2 identical ams quad MOx gas sensor and 2 identical Flusso's flow sensors (single hot wire). The combo chip did not exist yet, otherwise we would have tested four of them. For each quad-MOx gas sensor:

- Sensor 1 was left bare and operated as a flow sensor (to check if the heater's power consumption could be linked to the flow, see Part 4.2). The heater was heated at 400°C.
- Sensor 2 and 4 were coated with MOx 3 and operated at 320°C
- Sensor 3 was coated with MOx 4 and operated at 200°C

The flow sensors were set to operate at 110 Ω. The gas sequence presented in Table 6.2 was run three times, at three increasing flow speeds: 200 sccm (0.2 L/min), 300 sccm and 400 sccm.

Sequence	Gas	Concentration [ppm]	Flow [sccm]	rH [%]	Duration [min]
1	Air	/	100	50	15
2	Acetone	1.5	100	50	15
3	Air	/	100	50	15
4	Ethanol	50	100	50	15
5	Air	/	100	50	30
6	NO ₂	0.1	100	50	15
7	Air	/	100	50	15
8	H ₂	20	100	50	15
9	Air	/	100	50	30
10	CO	30	100	50	15
11	Air	/	100	50	15
12	CH ₄	20	100	50	15
13	Air	/	100	80	15
14	Acetone	1.5	100	80	15
15	Air	/	100	80	15
16	Ethanol	50	100	80	15
17	Air	/	100	80	30
18	NO ₂	0.1	100	80	15
19	Air	/	100	80	15
20	H ₂	20	100	80	15
21	Air	/	100	80	30
22	CO	30	100	80	15
23	Air	/	100	80	15
24	CH ₄	20	100	80	15
25	Air	/	100	80	15
					7h15

TABLE 6.1: Gas sequence test 1

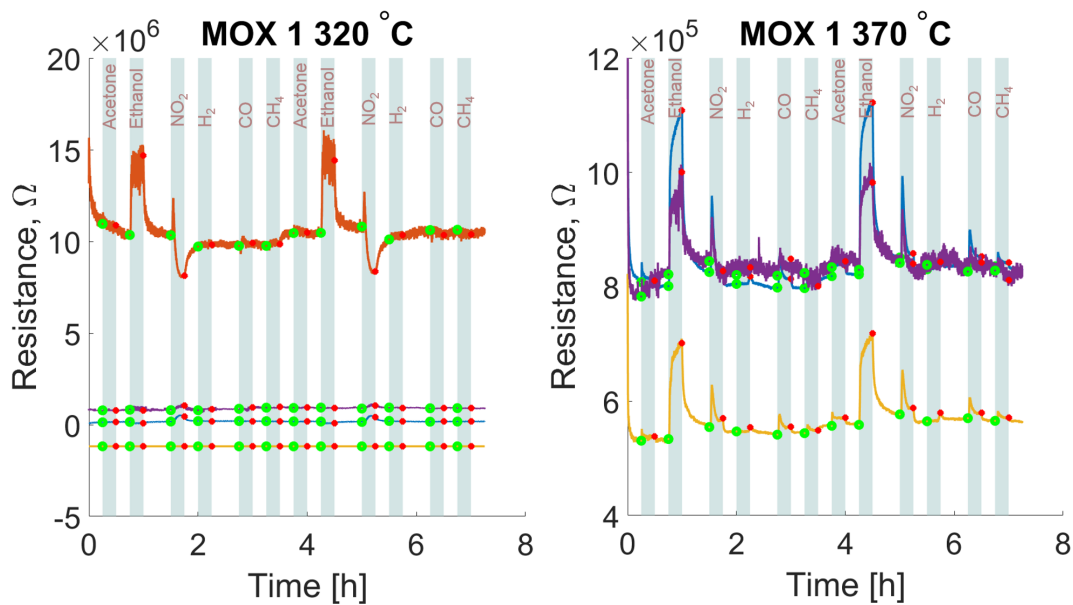


FIGURE 6.2: Response of the MOx 1 at 2 temperatures

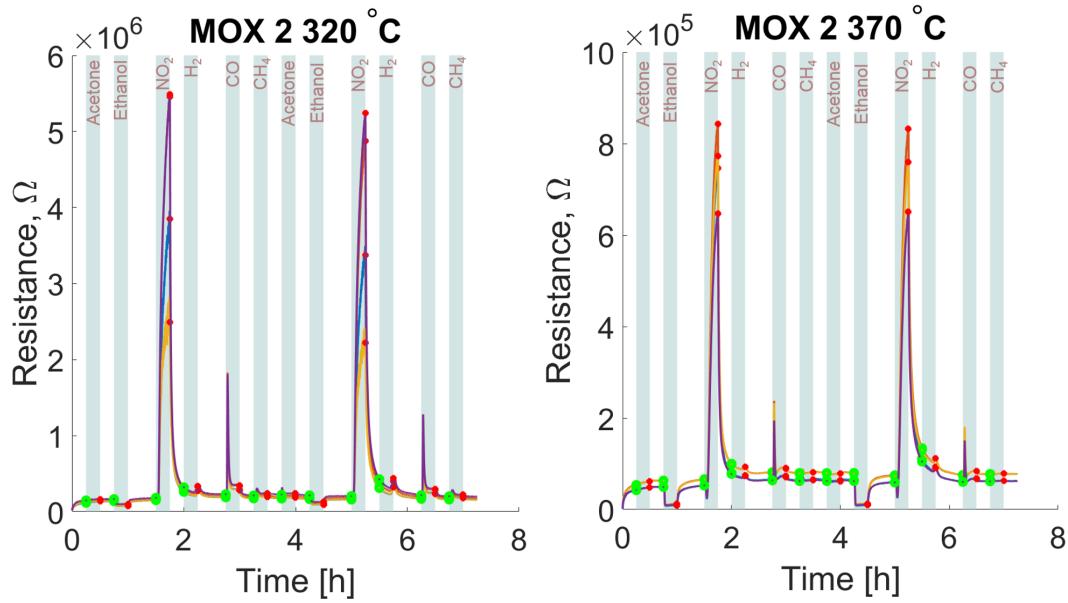


FIGURE 6.3: Response of the MOx 2 at 2 temperatures

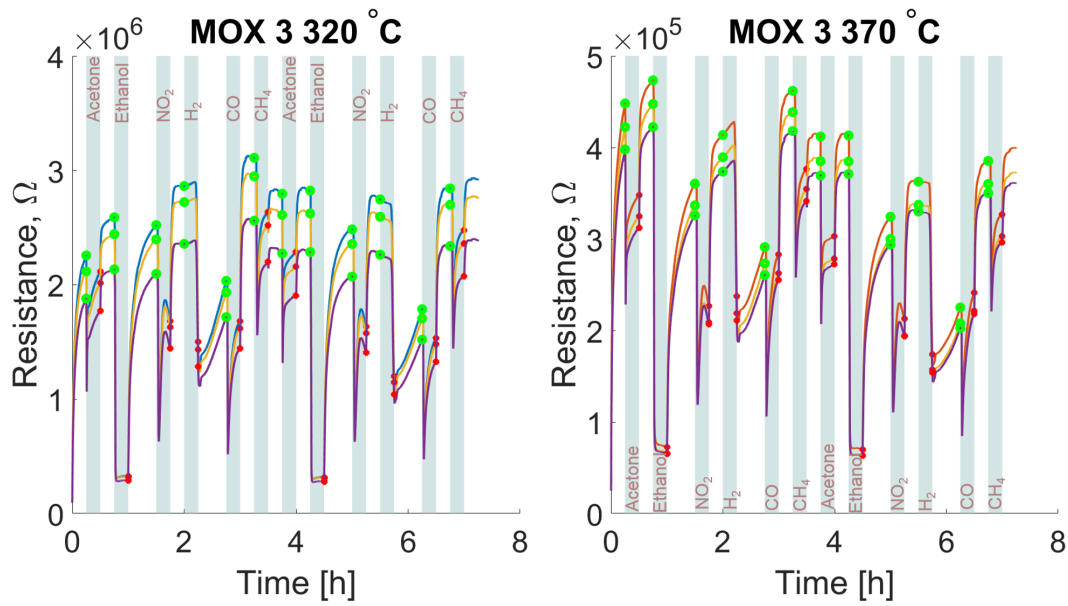


FIGURE 6.4: Response of the MOx 3 at 2 temperatures

The first observation is the comparison between the power consumption of the flow sensors and the two heaters of the sensors left bare. We see in Figure 6.6 that the power of the dedicated flow sensors increases with the flow velocity. On the other hand, the power consumption of the heaters barely depends on the flow velocity, only one of the two heaters increased its power consumption when the flow was increased from 0.3L/min to 0.4L/min. We can conclude that for this range of flow velocities, a dedicated flow sensor seems compulsory, as expected the heaters are not sensitive enough to the flow.

The second observation is the response of both MOx, plotted in in Figure 6.7 and 6.8. The responses are similar to the ones we got in test 1; MOx 3 is sensitive to both gases and MOx 4 only responded to NO_2 . We can note that the signals are slightly

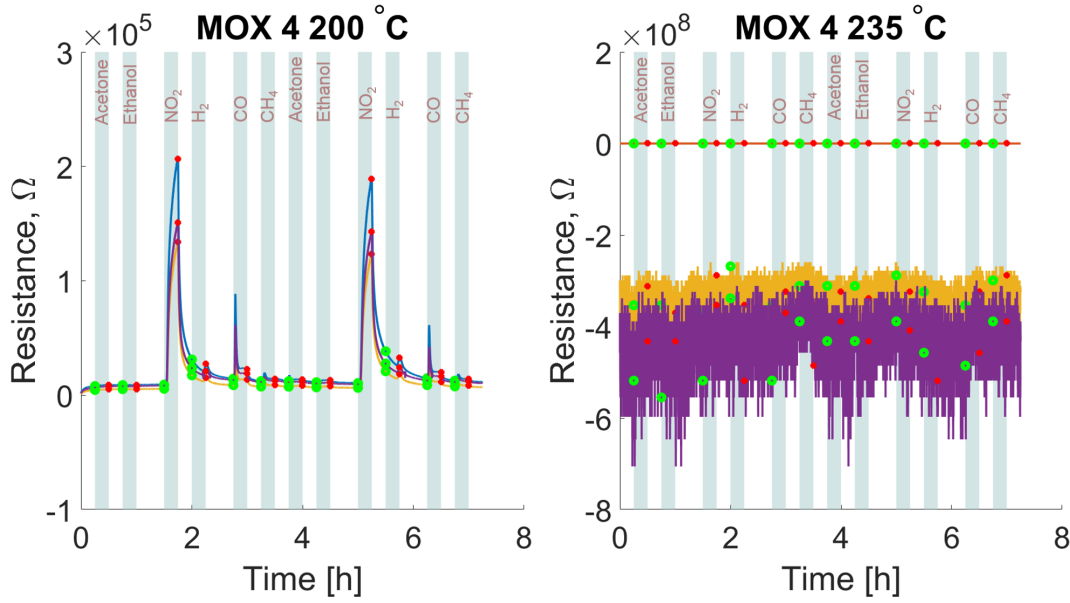


FIGURE 6.5: Response of the MOx 4 at 2 temperatures

more noisy than the ones we obtain with the usual ams measurement set-ups (in Test 1).

These results are very positive, we managed to measure many meaningful signals with the new electronics. Furthermore, the temperature control seems to work, we can see in Figure 6.9 that most of the time the resistances of the flow sensors are equal to 110 Ω . We can however notice the drops of the flow sensor resistance (or peaks in the power consumption) when the flow velocity changes. These are due to the overshoot and settling time of the PI controller and could eventually be reduced by adapting the parameters of the temperature control.

Sequence	Gas	Concentration	rH%	Flow [sccm]	Duration [min]
1	Air	/	50	200/300/400	15
2	Acetone	5 ppm	50	200/300/400	15
3	Air	/	50	200/300/400	15
4	NO2	0.1 ppm	50	200/300/400	15

TABLE 6.2: Gas sequence test 2

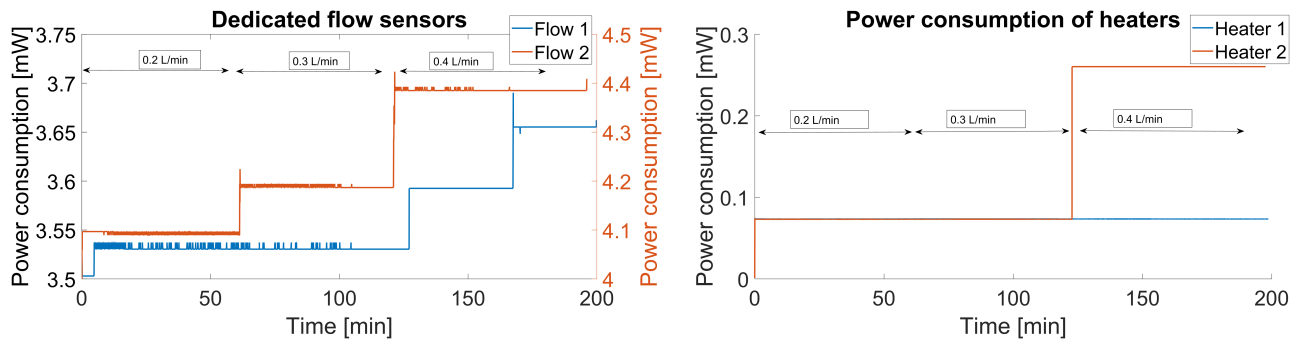


FIGURE 6.6: Flow sensor and heater power consumption

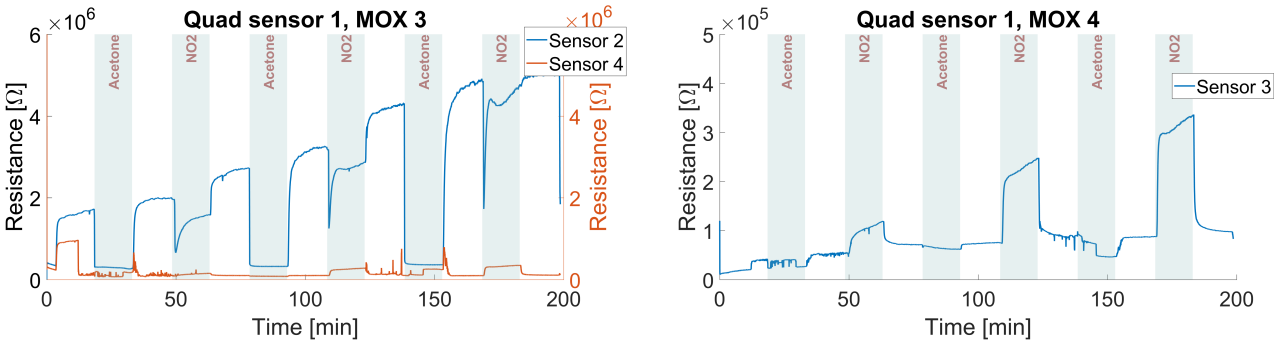


FIGURE 6.7: Response of Quad sensor 1

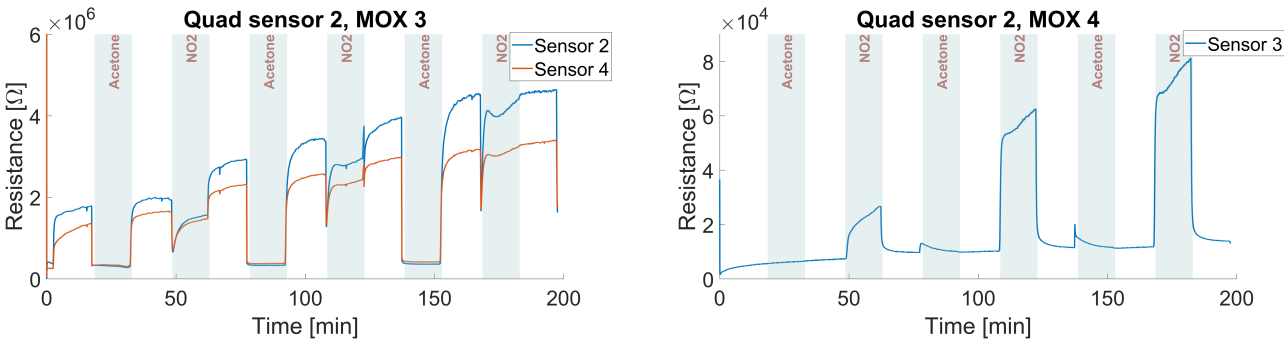


FIGURE 6.8: Response of Quad sensor 2

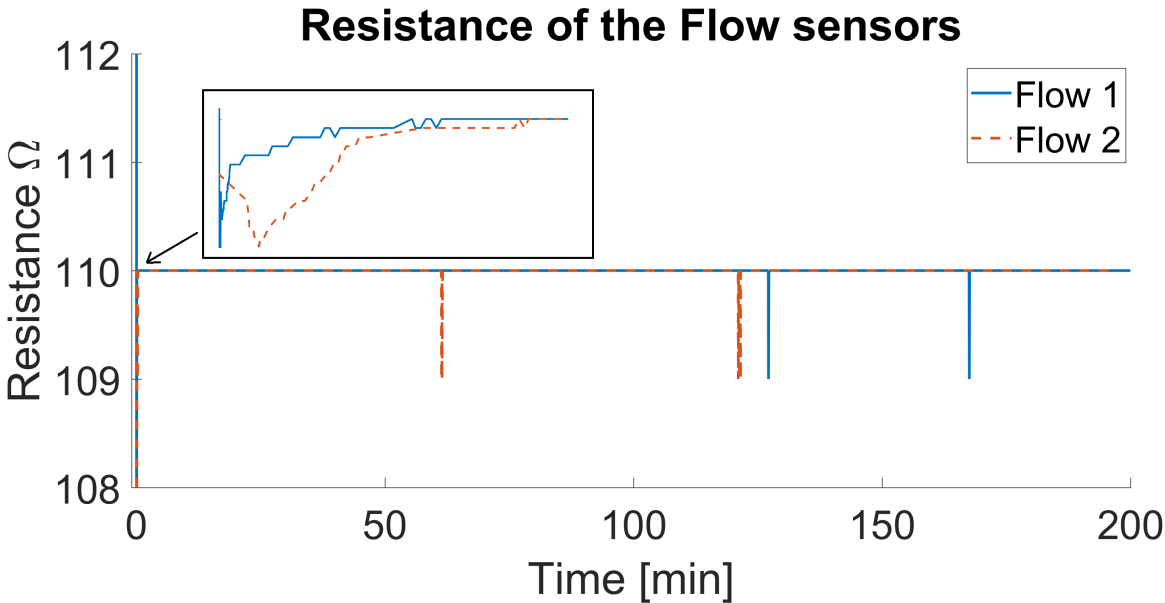


FIGURE 6.9: Resistance of the flow sensor

Chapter 7

Project conclusion

The purpose of the project was to re-design the electronics of the Breath-Fit project to include the constant temperature driving of the flow sensor and heaters. This allowed a simplification of the hardware as many components were now software driven. This second version was also needed after the re-design of the flow sensor by Flusso. This company is recent, at the time the first electronics were being developed, the design of their flow sensor was not finalized. In terms of design, the objectives have been met, a new hardware, firmware and GUI have been developed and are operational. The first test with the new electronics has been successful, the heaters and flow sensor kept their temperature constant throughout the test and the MOx readings are comparable to the ones obtained with the typical ams set-ups. Nonetheless, a few aspects of the new electronics could be improved:

- The response of the PI controller when the flow changes has an important overshoot and settling time, this could be solved by adding a derivative correction (PID controller).
- The MOx resistivity readout are slightly more noisy than the ones obtained with the classical AMS set-ups. This can be due to a bad noise management in the hardware, or the firmware that does not have the required precision. Improving this requires good skills in PCB and analog design.

In terms of future developments, a lot of material characterization is yet to be done to find the three right MOx for the combo-chip. Subsequently, the algorithm calculating the in-breath acetone concentration from the reading of the four sensors should be developed.

Bibliography

- [1] O. B. Crofford, R. E. Mallard, R. E. Winton, N. L. Rogers, J. C. Jackson, and U. Keller, "Acetone in breath and blood", *Transactions of the American clinical and climatological association*, pp. 128–139, 1977.
- [2] J. C. Anderson, "Measuring breath acetone for monitoring fat loss: Review", *Obesity*, vol. 23, no. 12, pp. 2327–2334, Feb. 2015. DOI: [10.1002/oby.21242](https://doi.org/10.1002/oby.21242).
- [3] Y. Qiao, Z. Gao, Y. Liu, Y. Cheng, M. Yu, L. Zhao, Y. Duan, and Y. Liu, "Breath ketone testing: A new biomarker for diagnosis and therapeutic monitoring of diabetic ketosis", *BioMed Research International*, vol. 2014, pp. 1–5, 2014. DOI: [10.1155/2014/869186](https://doi.org/10.1155/2014/869186).
- [4] C. Wang, L. Yin, L. Zhang, D. Xiang, and R. Gao, "Metal oxide gas sensors: Sensitivity and influencing factors", *Sensors*, vol. 10, no. 3, pp. 2088–2106, 2010. DOI: [10.3390/s100302088](https://doi.org/10.3390/s100302088).
- [5] A. Rothschild and Y. Komem, "The effect of grain size on the sensitivity of nanocrystalline metal-oxide gas sensors", *Journal of Applied Physics*, vol. 95, no. 11, pp. 6374–6380, 2004. DOI: [10.1063/1.1728314](https://doi.org/10.1063/1.1728314).
- [6] G. F. Fine, L. M. Cavanagh, A. Afonja, and R. Binions, "Metal oxide semiconductor gas sensors in environmental monitoring", *Sensors*, vol. 10, no. 6, pp. 5469–5502, Jan. 2010. DOI: [10.3390/s100605469](https://doi.org/10.3390/s100605469).
- [7] H. Meixner and U. Lampe, "Metal oxide sensors", *Sensors and Actuators B: Chemical*, vol. 33, no. 1-3, pp. 198–202, 1996. DOI: [10.1016/0925-4005\(96\)80098-0](https://doi.org/10.1016/0925-4005(96)80098-0).
- [8] S. Jeevan, P. Gowthaman, M. Venkatachalam, and D. Manickam, "A review of gas sensors based on semiconducting metal oxide", vol. 12, Mar. 2017.
- [9] R. Ionescu, E. Espinosa, R. Leghrib, A. Felten, J. Pireaux, R. Erni, G. V. Tendeloo, C. Bittencourt, N. Cañellas, and E. Llobet, "Novel hybrid materials for gas sensing applications made of metal-decorated mwcnts dispersed on nano-particle metal oxides", *Sensors and Actuators B: Chemical*, vol. 131, no. 1, pp. 174–182, 2008, ISSN: 0925-4005.
- [10] H. Gao, H. Jia, B. Bierer, J. Wöllenstein, Y. Lu, and S. Palzer, "Scalable gas sensors fabrication to integrate metal oxide nanoparticles with well-defined shape and size", *Sensors and Actuators B: Chemical*, vol. 249, pp. 639–646, 2017. DOI: [10.1016/j.snb.2017.04.031](https://doi.org/10.1016/j.snb.2017.04.031).
- [11] S. Lee, "Electrodes for semiconductor gas sensors", *Sensors*, vol. 17, no. 4, p. 683, 2017. DOI: [10.3390/s17040683](https://doi.org/10.3390/s17040683).
- [12] J. Courbat, D. Briand, and N. D. Rooij, "Reliability improvement of suspended platinum-based micro-heating elements", *Sensors and Actuators A: Physical*, vol. 142, no. 1, pp. 284–291, 2008. DOI: [10.1016/j.sna.2007.04.006](https://doi.org/10.1016/j.sna.2007.04.006).
- [13] S. Ali, F. Udrea, W. Milne, and J. Gardner, "Tungsten-based soi microhotplates for smart gas sensors", *Journal of Microelectromechanical Systems*, vol. 17, no. 6, pp. 1408–1417, 2008. DOI: [10.1109/jmems.2008.2007228](https://doi.org/10.1109/jmems.2008.2007228).

- [14] G. Korotcenkov, "Practical aspects in design of one-electrode semiconductor gas sensors: Status report", *Sensors and Actuators B: Chemical*, vol. 121, no. 2, pp. 664–678, 2007. DOI: [10.1016/j.snb.2006.04.092](https://doi.org/10.1016/j.snb.2006.04.092).
- [15] K. Kasama, H. Fukuda, and S. Nomura, "Highly sensitive mosfet gas sensors with porous pt-snox gate electrode for co gas sensing applications", *Electronics Letters*, vol. 34, no. 14, p. 1393, 1998. DOI: [10.1049/e1:19980970](https://doi.org/10.1049/e1:19980970).
- [16] T. W. J. Kuo, L. Yu, and E. Meng, "Micromachined thermal flow sensors—a review", *Micromachines*, 2012, ISSN: 2072-666X.
- [17] N. Nguyen, "Micromachined flow sensors—a review", *Flow Measurement and Instrumentation*, vol. 8, no. 1, pp. 7–16, 1997. DOI: [10.1016/s0955-5986\(97\)00019-8](https://doi.org/10.1016/s0955-5986(97)00019-8).