



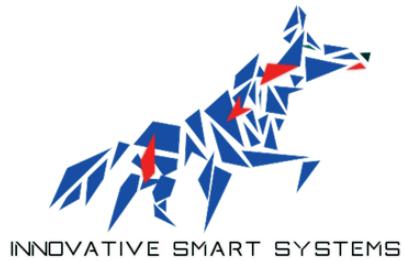
INSTITUT NATIONAL DES SCIENCES  
APPLIQUÉES  
TOULOUSE

2024 - 2025

---

## MOSH Gas Sensor Project Report

---



Achile Caute, Amine Benchekroun, Hugo Michel, Alexandre Masson

January 29, 2025

# Contents

<b>Introduction</b>	<b>2</b>
<b>1 MOx sensor manufacturing</b>	<b>3</b>
1.1 MOx sensor architecture . . . . .	3
1.2 Photolithography . . . . .	3
1.2.1 Polysilicium heating element manufacturing: . . . . .	3
1.2.2 Metalization and metal etching: . . . . .	3
1.3 Metal Oxide (MOx) synthesis . . . . .	4
1.3.1 Tungsten trioxide crystals synthesis: . . . . .	4
1.3.2 Dielectrophoresis: . . . . .	5
1.4 Packaging . . . . .	5
<b>2 Sensor characterization</b>	<b>7</b>
2.1 Thermistor . . . . .	7
2.2 The MOx resistance . . . . .	7
2.2.1 Nominal resistance . . . . .	7
2.2.2 Sensibility . . . . .	7
2.3 remarks . . . . .	8
<b>3 Integration of the MOx gas sensor into a smart device</b>	<b>10</b>
3.1 A smart device using Arduino . . . . .	10
3.1.1 Integration to the LoRa Network . . . . .	11
3.1.2 Integration of monitoring on our device . . . . .	11
3.1.3 Including Bluetooth connection . . . . .	12
3.1.4 Overall solution . . . . .	12
3.2 Development of a Node-RED Application . . . . .	13
3.2.1 Flow Design . . . . .	13
3.2.2 Development of a Small IoT Application . . . . .	14
3.3 System Components . . . . .	16
3.3.1 The Amplifier . . . . .	16
3.3.2 The Gas sensor . . . . .	17
3.3.3 Sensor Connections and PCB Design . . . . .	17
<b>4 Conclusion</b>	<b>20</b>

# Introduction

Our project focuses on the integration of a gas sensor designed during a training period at AIME. As part of the **Innovative Smart Systems** major, we conducted a comprehensive process that included design and development within the AIME laboratory, creation of schematics, routing, wiring, programming, as well as the management of wireless communication and network aspects.

This project serves as a comprehensive assessment of our skills in key areas such as networking, programming, and electronics. Its main objective is to transmit the sensor data collected via a LoRa network to a dashboard for visualization and analysis. This report outlines the various stages of the project, highlighting the challenges overcome and the skills developed throughout its execution.

# Chapter 1

## MOx sensor manufacturing

### 1.1 MOx sensor architecture

### 1.2 Photolithography

#### 1.2.1 Polysilisium heating element manufacturing:

The polysilicium heating element is used to heat up the MOx resistor in order to decrease its resistance and evaporate the water residue on it. We did not make it ourselves, it was already done by the professionals that work at the AIME clean room. It was manufactured by first making a layer of polysilicium and then etching into it to keep the shape that we wanted, i.e. the polysilicium rode. The etching has been done using a photo-resist resin that created a mask over the part we wanted to protect from the etching process. The resin is activated with UV light in an photolithography machine. There is a lot more to tell about the manufacturing of IC, however it would take too much time, and we are not the most well placed people to talk about it.

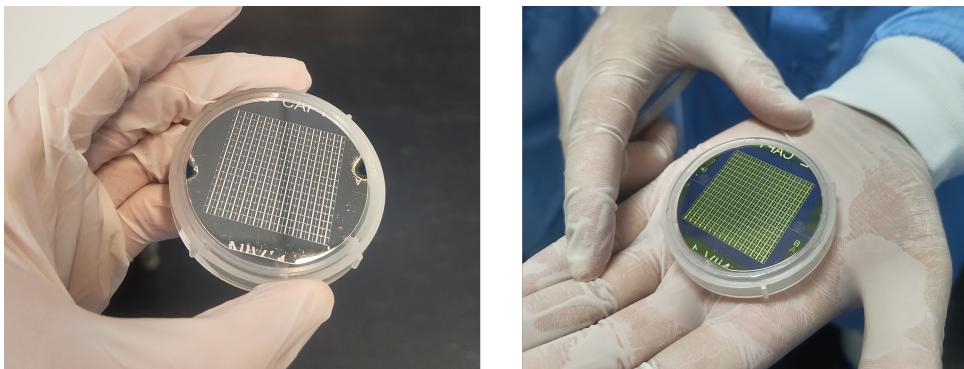


Figure 1.1: Photos of our PCB before and after the photolithography steps

#### 1.2.2 Metalization and metal etching:

### Metallization

Metallization involves depositing a thin 500 nm layer of aluminum on the component side of the chip. This deposition is carried out using RF sputtering under controlled conditions, including an initial pressure of  $10^{-7}$  mbar, a deposition pressure of  $2.10^{-3}$

mbar, an RF power of 250 W, and a target-to-substrate distance of 75 mm. The total duration of the deposition process is 10 minutes. This step is crucial for forming the electrodes and electrical contacts of the sensors.

## Metal Etching

Metal etching follows metallization and defines the patterns of the electrodes and electrical contacts by removing excess aluminum. This process begins with the homogenization of the etching bath using an ultrasonic bath, followed by a sequence of steps:

- Deposition of Shipley S1813 resist via spin coating,
- Exposure using a specific mask (No. 3) for 5 seconds,
- Development, and
- Two annealing cycles on a hot plate (100°C and 120°C).

The aluminum etching is performed in a bath composed of 40 volumes of phosphoric acid ( $H_3PO_4$ ), 7 volumes of nitric acid ( $HNO_3$ ), and 7 volumes of water. An optical microscope inspection is conducted after each step to ensure the quality of the etched patterns. Finally, the resist is removed with a mixture of acetone and deionized water, followed by spin drying.



Figure 1.2: Microscopic visualization of our PCB

## 1.3 Metal Oxide (MO<sub>x</sub>) synthesis

### 1.3.1 Tungsten trioxide crystals synthesis:

The synthesis of WO<sub>3</sub> crystals involves two main steps: seed preparation and nanowire growth. The seeds are obtained by mixing a solution of  $Na_2WO_4 \cdot 2H_2O$  with a diluted HCl solution, followed by stirring and centrifugation to isolate the precipitate. The seeds are then transferred into a hydrothermal bomb containing NaSO, where they undergo a hydrothermal annealing process at 180°C for 1 hour. After cooling and rinsing, the nanowires are collected by centrifugation and stored. This method enables the production of homogeneous nanostructures suitable for applications such as gas sensors.

### 1.3.2 Dielectrophoresis:

The integration of WO nanoparticles onto the interdigitated electrodes of the sensor is performed via dielectrophoresis. A drop of aqueous solution containing the nanoparticles ( $10 \mu\text{L}$ ) is deposited onto the chip. The electrodes are then polarized with a sinusoidal voltage (100 kHz, 10 V) for 60 seconds, aligning the nanoparticles through dielectrophoretic forces.

After integration, excess particles are removed by rinsing with deionized water, and the remaining water is gently absorbed. Finally, the device is inspected under an optical microscope to verify the alignment of the nanoparticles, thus ensuring a uniform and functional active layer ready for testing.

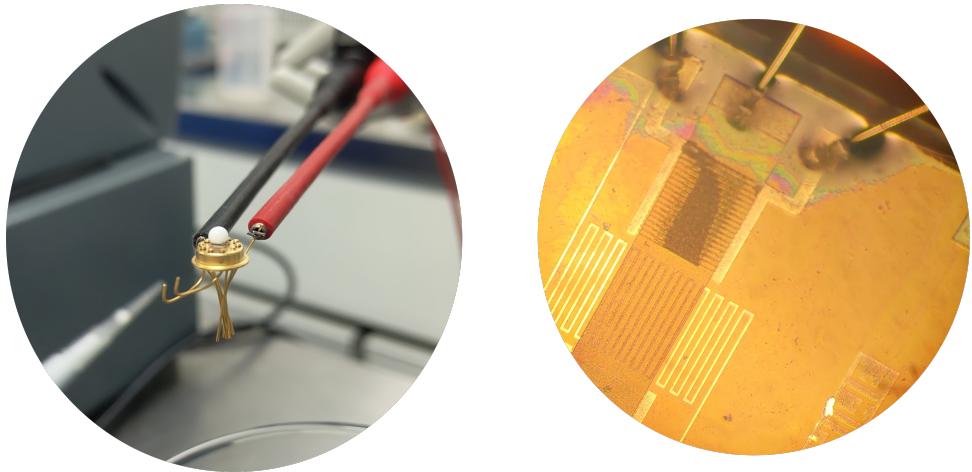


Figure 1.3: Photos before and after Dielectrophoresis

## 1.4 Packaging

For the packaging we used JEDEC's transistor outline 5 with 10 pins, it was actually used by Fairchild semiconductor when it still exist. The sensor was placed over a piece of glass to isolate it, into the package. We then use wire bonding to connect the chip to the pins of the package. Wire bonding is a technic where a small arm that hold a tiny wire wiggle it with ultrasonic vibration, and this weld it.

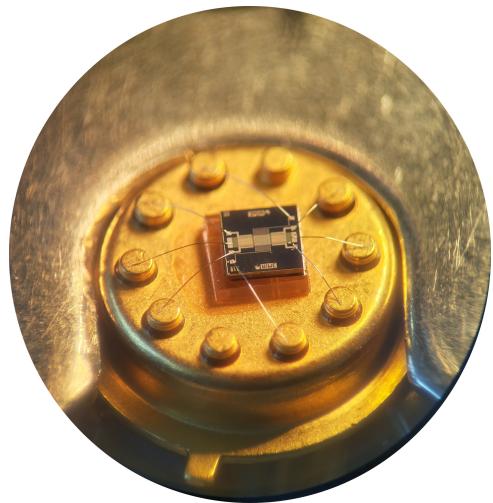


Figure 1.4: Photo of the welded gas cell on its support

# Chapter 2

## Sensor characterization

### 2.1 Thermistor

The thermistor  $R_{alu}$  is a positive temperature coefficient thermistor, which means that its resistance increase with heat. Also, we did not characterize it to know the said coefficient, and take the arbitrary value of  $1^\circ\text{C}$  per  $\Omega$ , as it is said to have the same behavior as a pt100, which are well known platinum resistor for temperature measure.

We should have characterize it with a heat source and a reference thermometer. But is has not been done. The approximation that the resistance varies as said, is good enough for what we want to do, i.e. use a MOx sensor.

When it comes to its resistance to room temperature, to get the reference resistance. We used a waveform generator and an instrumentation board to plot the relation between the current and the tension, and thank to Ohm's law, we were able to retrieve its nominal resistance as you can see in figure 2.1. The nominale value we find was  $86\Omega$ .

Note that the resistance is non-linear when the voltage increase to much, which is probably due to self warming.

### 2.2 The MOx resistance

#### 2.2.1 Nominal resistance

The MOx resistance when in its reduce form is in the  $\text{G}\Omega$ . It is actually quite difficult to mesure such a resistance as it needs a very precise current generator, and it produce a lot of thermic noise. Hence the value we get was a crude approximation of the real value, still it performs quite well at  $16 \text{ G}\Omega$ .

#### 2.2.2 Sensibility

The characterization of the sensibility of the sensor, and its sensibility limit with a given conditioner is a hard task. What we have done is not the best way to do it and we will go in deeper details later concerning what we should have done to yield more useful data.

We measure the sensor sensibility to ethanol. To set it up, we first increase the temperature to  $550^\circ\text{K}$ , such that the water evaporate, and that the resistor became lower. Then we proceed in two cycles, a regeneration one, where we put the sensor under a dry air flow to regenerate the sensor, i.e. reduce the oxide on the edges of the  $\text{W}_0_3$  crystals. And a measure one, where we put the sensor under a ethanol air mixture of approximately

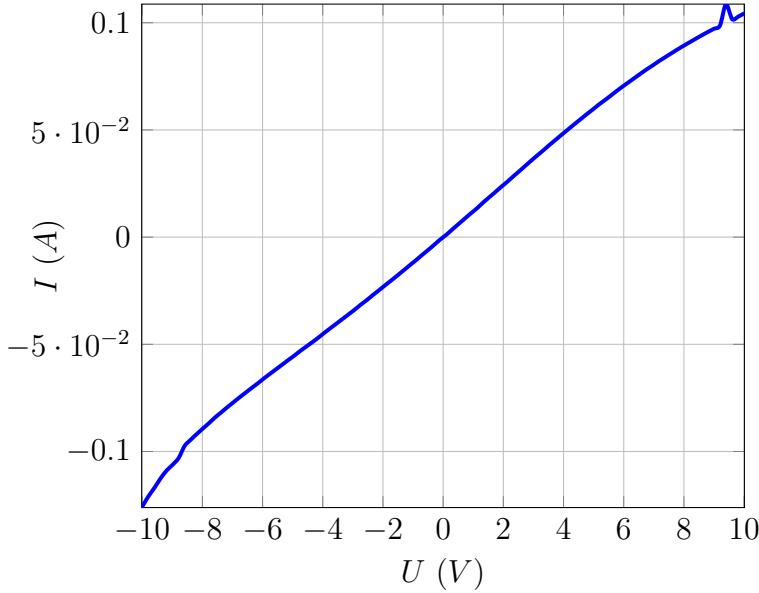


Figure 2.1: Thermistance characteristic ( $T_{\text{ambient}} = 21^\circ\text{C}$ ), non linearities over maximal range.

1000 ppm of ethanol, which lead to a resistance drop as you can see in figure 2.2, following an exponential behaviour.

The first thing we measure is the relative dynamic of the sensor to such an experiment, we get  $\Delta R/\Delta R_0 = 64\%$ , we also measure the sensitivity coefficient (see eq.2.1) and the response time to get from the dry air resistance to 10% of the final value of the ethanol mixture resistance.

$$\frac{R_{\text{air}} - R_{\text{eth}}}{R_{\text{air}}} \quad (2.1)$$

Finally we get the sensitivity in  $\Omega/\text{ppm}$ , which is computed with  $\delta R/\delta C$  as our sensor is non-linear (exponential behaviour, see fig.2.2), and with  $C$ , the concentration on ppm. We also we get the sensitivity limit, which is the lowest variation of gas concentration we can measure for a given conditioner. We computed it by computing the lowest resistance we could measure with the arduino ADC (10 bits), multiply by the gain of the amplifier, and dividing this value by the sensitivity.

You can find all the values we found in table 2.1. All its measures have been done in an ambient environment at  $T_{\text{ambient}} = 21^\circ\text{C}$ .

## 2.3 remarks

There are a lots of things that should have been done better. First of all we test the sensor on ethanol only, and with a precarious setup. In fact, we are not really sure that the ethanol concentration really is 1000 ppm, as the mixture has been done with a bubbler.

---

<sup>1</sup>Max variation of the MOx resistor over the whole sensor range.

<sup>2</sup>Sensitivity coefficient

<sup>3</sup>Response time to ethanol (10% of the final value)

<sup>4</sup>Sensibility limit, lowest measurable change in gas concentration

Parameter	Test Conditions	Min	Typ.	Max	Unit
<b>Temperature sensor (Aluminium thermistor)</b>					
$R_{temp}$	$U_{temp} = 10 \text{ V}$	80	86	94	$\Omega$
$S_{temp}$	$U_{temp} = 10 \text{ V}$		1		$\Omega/\text{°C}$
<b>Poly-silicium heating element</b>					
$R_{poly}$	$U_{barreau} = 5 \text{ V}$	79	87	94	$\Omega$
<b>Tungsten trioxide element (gas sensor)</b>					
$R_{gas}$	$T_{MOx} = 21\text{°C}$ , gas = N <sub>2</sub> O <sub>2</sub> (dry air)	10	16	20	$\text{G}\Omega$
$\Delta R/R_0^1$	$T_{MOx} = 277\text{°C}$ , gas = ethanol, $C_{eth} = 1000 \text{ ppm}$	35	64	237	%
$k_{eth}^2$	$T_{MOx} = 277\text{°C}$ , gas = ethanol, $C_{eth} = 1000$		0.34		ppm
$t_{eth}^3$	$T_{MOx} = 277\text{°C}$ , gas = ethanol, $C_{eth} = 1000 \text{ ppm}$		88		s
$S$	$T_{MOx} = 277\text{°C}$ , gas = ethanol, $C_{eth} = 1000 \text{ ppm}$		85000		$\Omega/\text{ppm}$
$S_{lim}^4$	$T_{MOx} = 277\text{°C}$ , gas = ethanol, $C_{eth} = 1000 \text{ ppm}$		38.3		ppm

Table 2.1: Electrical characteristics of the MOx gas sensor

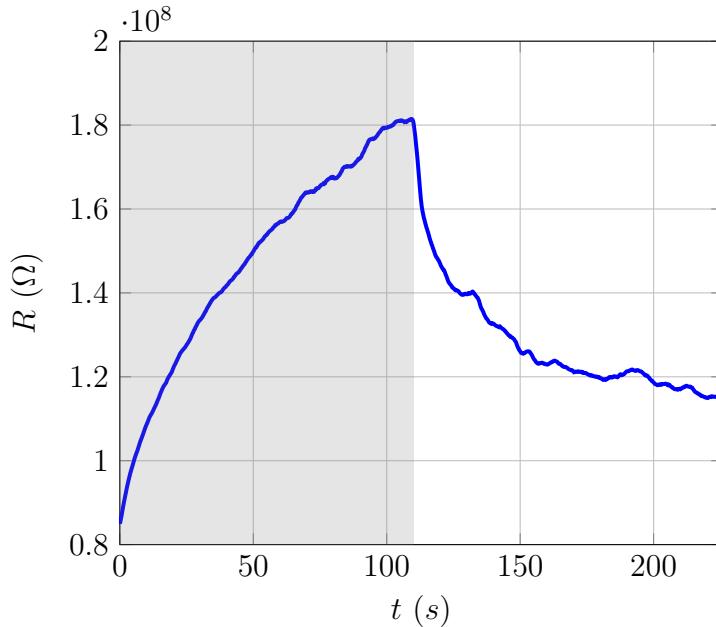


Figure 2.2: Evolution of the MOx resistance during a regeneration phase (in gray) no gas presence, and a detection phase. ( $T_{MOx} = 277\text{°C}$ , gas = ethanol vapor.)

Also, we have not characterize the thermistor properly. We should have measure its PTC right with a control thermometer.

# Chapter 3

## Integration of the MOx gas sensor into a smart device

### 3.1 A smart device using Arduino

We will integrate our gas sensor on a shield and then connect it to our designed solution on the Arduino board. You will find the complete code to integrate each module on the board joined to the report.

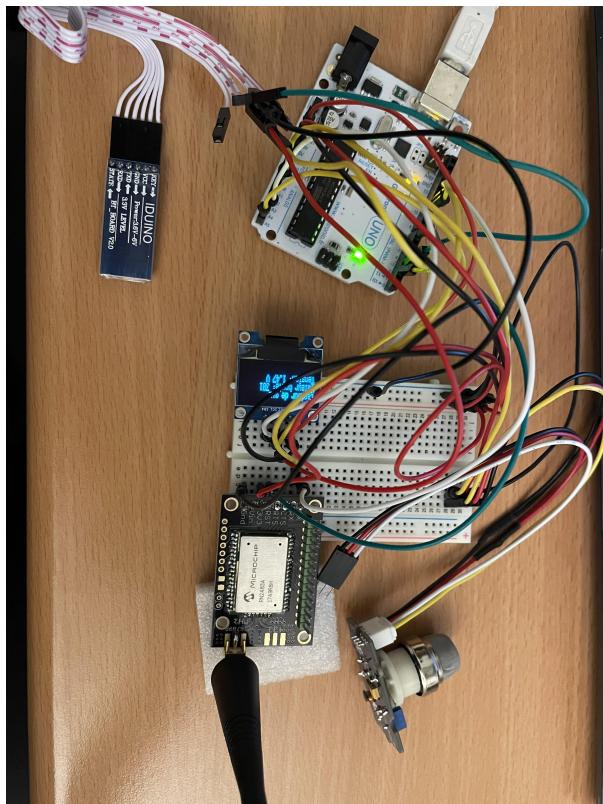


Figure 3.1: Our solution including the Arduino board, the LoRa module, the OLED screen, the bluetooth module and a gas sensor that simulate the integration of our gas sensor

### 3.1.1 Integration to the LoRa Network

As the first step of our project, we focused on establishing a connection between the Arduino and the LoRa network. This was achieved using the RN2483 chip manufactured by Microchip, a module designed specifically for LoRa network communication. Due to the small size of the chip's connectors, it was necessary to solder the RN2483 onto a custom board with larger connectors to ensure compatibility and facilitate communication with the Arduino.

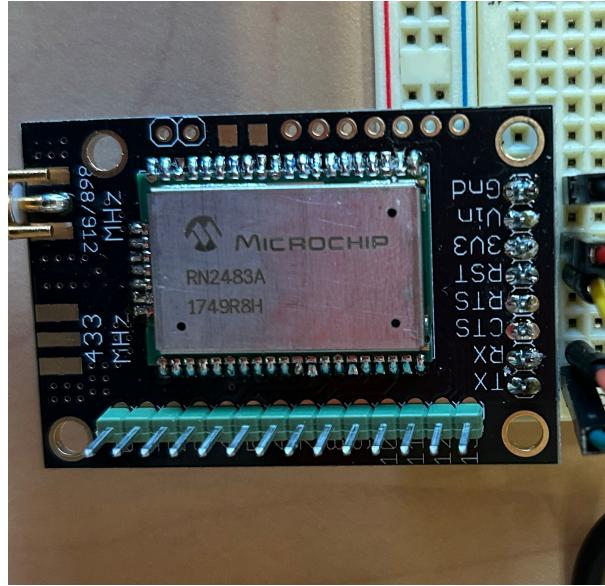


Figure 3.2: The RN2483 LoRa Module

After configuring the RN2483 module and completing the initial integration steps, we did not successfully connected our Arduino to INSA's LoRa network. The problem on was that the gateway didn't let us transmit our data despite the full configuration of the module on our device. That's why we put the blame on the LoRa module (in consultation with the supervisor on the project) and we decided to focus on the other aspects of the integration on the smart device.

### 3.1.2 Integration of monitoring on our device

We used an OLED screen (SBC-OLED01 module) that displays in real time the measure of the gas sensor (the grove sensor that simulates our manufactured gas sensor).

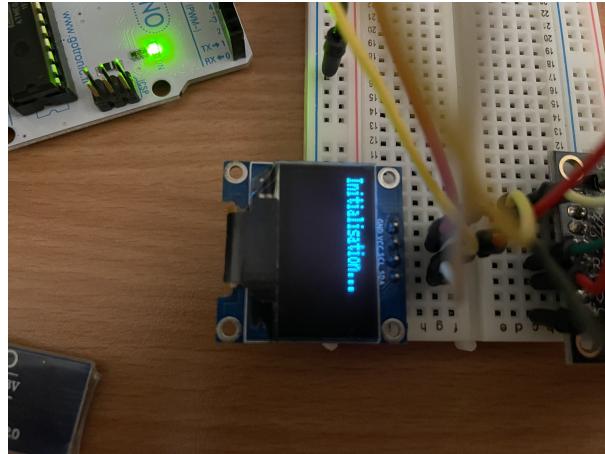


Figure 3.3: The OLED screen

### 3.1.3 Including Bluetooth connection

Using the module HC05 for Bluetooth connection we allow an mobile android application to be connected for the monitoring of the data from our gas sensor remotely.

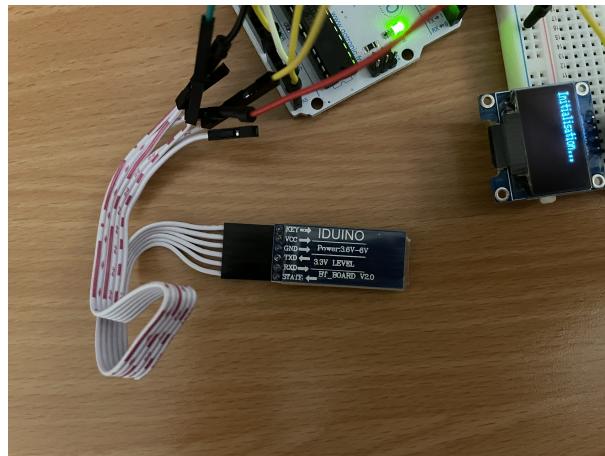


Figure 3.4: The HC05 Bluetooth module

### 3.1.4 Overall solution

Aggregating all this modules using the Arduino board, we performed (although the connection to the LoRa network wasn't successful) to create a platform that collect the data and transmit it in three ways: directly displaying it on the device, sending it to a long range network and sending it to a mobile using a short range communication. This way we explored different approaches and different solution to monitor and treat the data coming from our gas sensor (even though due to the lack of time the shield wasn't created and therefore we performed our tests using the Grove gas sensor).

## 3.2 Development of a Node-RED Application

### 3.2.1 Flow Design

#### Node-RED Integration

Node-RED is a web-based flow editor that simplifies the connection and management of various data flows.

In our implementation, we utilized Node-RED's MQTT subscription block to receive data transmitted by ChirpStack. The incoming data, originally encoded in base64 by ChirpStack, was processed using a custom JavaScript function block designed specifically to decode the base64 format. After decoding, the processed data was directed to a dashboard block in Node-RED. This final step enabled the clear and accessible visualization and monitoring of the data directly on the dashboard.

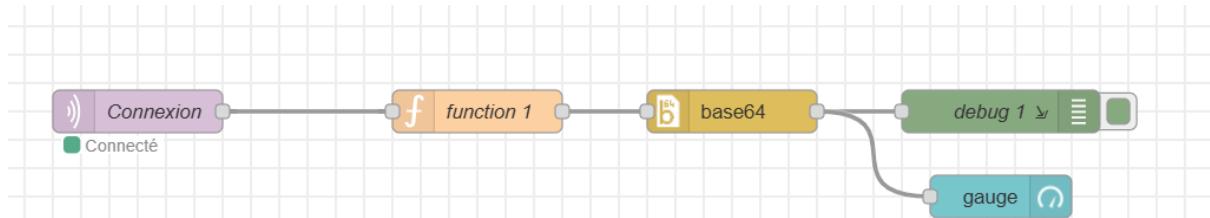


Figure 3.5: Node-RED Flow for Data Processing

#### Sensor Data and Dashboard Visualization

The sensor values detected and processed by our system are displayed on a dynamic dashboard using Node-RED's visualization tools. Below, we present both the raw data received and the corresponding dashboard interface that visualizes these values in real-time.

**Raw Data Received** The raw sensor data received by the system is displayed in the debug section of Node-RED. The data values are shown in a buffer format and are updated continuously as the sensor detects changes. Figure 3.6 shows an example of the raw data received.

```

18/12/2024 10:19:35 noeud: debug 1
application/feb26b6b-94e0-45c9-ab1c-
1f8aec7935bd/device/0004a30b00e9b37d/event/up :
msg.payload : buffer[1]
▶ [ 75 ]

18/12/2024 10:21:39 noeud: debug 1
application/feb26b6b-94e0-45c9-ab1c-
1f8aec7935bd/device/0004a30b00e9b37d/event/up :
msg.payload : buffer[1]
▶ [ 66 ]

18/12/2024 10:22:14 noeud: debug 1
application/feb26b6b-94e0-45c9-ab1c-
1f8aec7935bd/device/0004a30b00e9b37d/event/up :
msg.payload : buffer[1]
▶ [ 44 ]

18/12/2024 10:22:31 noeud: debug 1
application/feb26b6b-94e0-45c9-ab1c-
1f8aec7935bd/device/0004a30b00e9b37d/event/up :
msg.payload : buffer[1]
▶ [ 78 ]

```

Figure 3.6: Raw sensor data received and displayed in Node-RED debug panel.

**Dashboard Visualization** The processed sensor values are visualized using a gauge on the Node-RED dashboard. This gauge dynamically updates to reflect the real-time value detected by the sensor. For instance, when the sensor detects a value of 78, the gauge displays this value on a scale. Figure 3.7 illustrates the dashboard visualization.

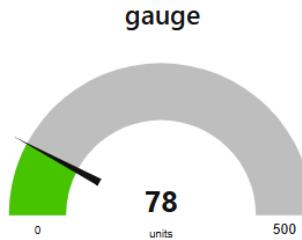


Figure 3.7: Dashboard gauge showing the real-time sensor value (78).

### 3.2.2 Development of a Small IoT Application

To showcase possible use cases for the IoT system of our gas sensor, we developed a small application capable of communicating real-time data from the sensor.

To achieve this, we used a tool called MIT App Inventor, introduced during the Open Source Tools course. This platform allows us to create Android applications using a block-based programming approach similar to Scratch. The interface is designed by dragging and dropping components. While this method seems practical at first, it requires a good understanding of the available tools and their locations. Additionally, the functionality is somewhat limited.

The purpose of the application is to display real-time data from the gas sensor and to set alert levels for triggering specific actions.

In our implementation:

- When the first threshold is reached, the background color changes to indicate potential danger.

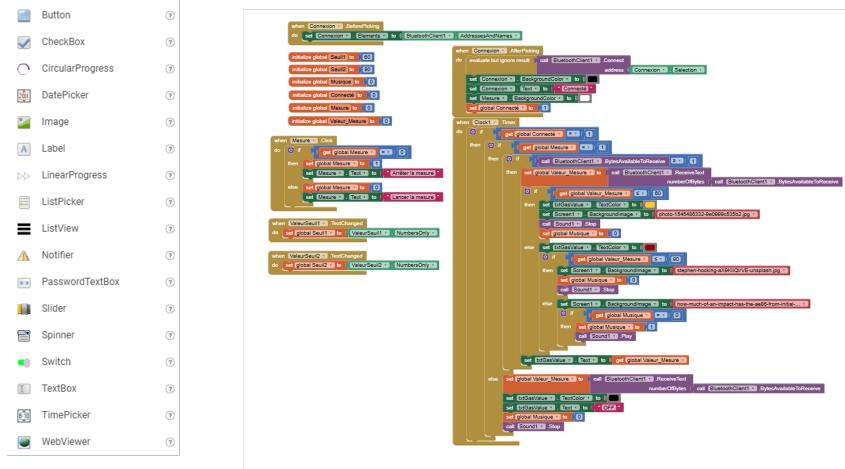


Figure 3.8: MIT App Inventor Interface

- When the second threshold is exceeded, an audio alert is triggered.

The application features:

- A connection button to initialize communication with the microcontroller.
- A button to start or stop the measurement process.
- A display for the measured data.
- Two adjustable thresholds to set alert levels.



Figure 3.9: Screenshots of our application

This application, while not directly useful on its own, demonstrates an open-source monitoring solution that can be easily implemented in various setups.

### 3.3 System Components

#### 3.3.1 The Amplifier

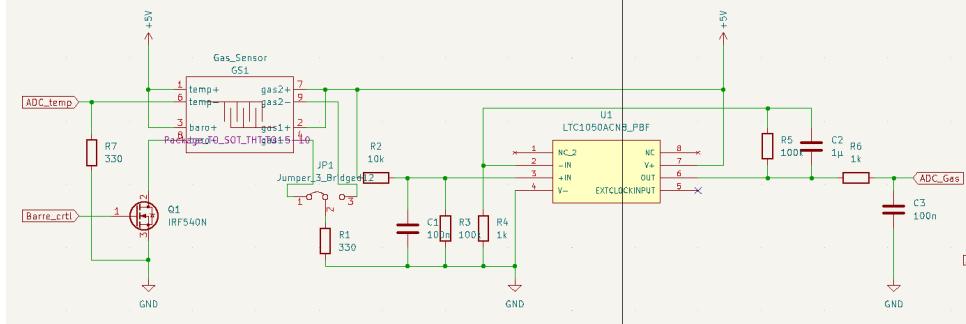


Figure 3.10: Conditioner schematic

In our project, the first step in processing sensor data involves utilizing a transimpedance amplifier. This component plays a vital role in conditioning the signal to ensure compatibility with the Arduino's ADC. To optimize its performance, we simulated the circuit using LTSpice, enabling us to analyze the behavior of the transimpedance amplifier and evaluate the effectiveness of different filters. The design and operation of this amplifier are essential for achieving accurate signal detection and reliable data processing within our system.

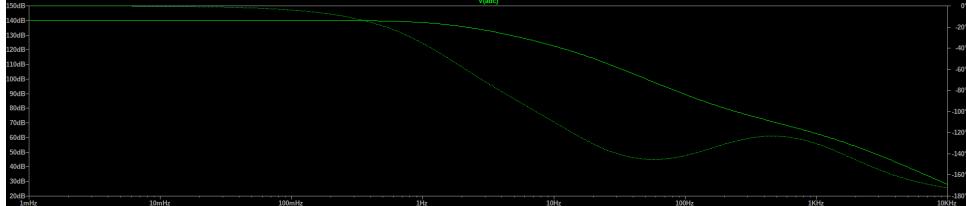


Figure 3.11: Transimpedance amplifier transfer function

The amplifier has a gain of 140 dB, which is equivalent to  $10^7$ , as you can see in figure 3.11. This high gain is essential to measure the resistance as this one is very high, hence the current in the voltage divider bridge is fairly low. This conditioner is actually not the best for this application as it might end up having a poor SNR, as the thermic noise of a resistor is higher as its impedance is great.

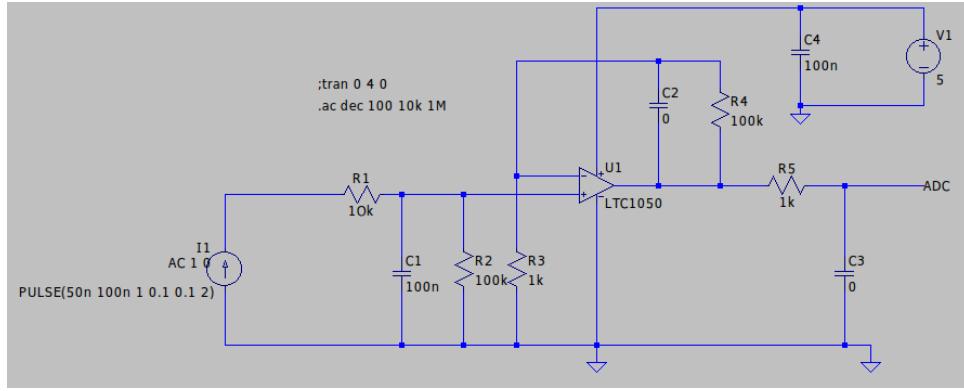


Figure 3.12: Trans-impedance amplifier schematic

As you can see in figure 3.12, the amplifier is a simple non-inverter amplifier surrounded by filters on the input, output and feedback. You can find their cut-off frequency in table 3.1. The goal of those filters is to cut-off eventual noise and get a better measure.

Also, to prevent 50 Hz, EMC noise, we add a low pass filter on the alimentation, as it is often an important source of noise.

Filter	Theoretical Cut-off Frequency	Simulated Cut-off Frequency
1 C1 = 100 nF	$F_C = \frac{1}{2\pi R_1 C_1} = 159.15 \text{ Hz}$	160 Hz
2 C2 = 1μF	$F_C = \frac{1}{2\pi R_4 C_2} = 15.91 \text{ Hz}$	16 Hz
3 C3 = 100 nF	$F_C = \frac{1}{2\pi R_5 C_3} = 1591.54 \text{ Hz}$	1.62 kHz

Table 3.1: Non-inverter's filter cut-off frequencies

### 3.3.2 The Gaz sensor

Name	Pin number	Function
Temp+	1	Positive terminal for the internal aluminium termistance
Temp-	6	Negative terminal for the internal aluminium termistance
Barreau+	3	Positive terminal for the internal heating element
Barreau-	8	Negative terminal for the internal heating element
Gas1+	2	Positive terminal for the first gas sensor
Gas1-	4	Negative terminal for the first gas sensor
Gas2+	7	Positive terminal for the second gas sensor
Gas2-	9	Negative terminal for the second gas sensor

Table 3.2: Gas sensor pinout

### 3.3.3 Sensor Connections and PCB Design

To integrate all the components of our sensor system into a single cohesive unit, we designed a dedicated Printed Circuit Board (PCB). This PCB consolidates the various

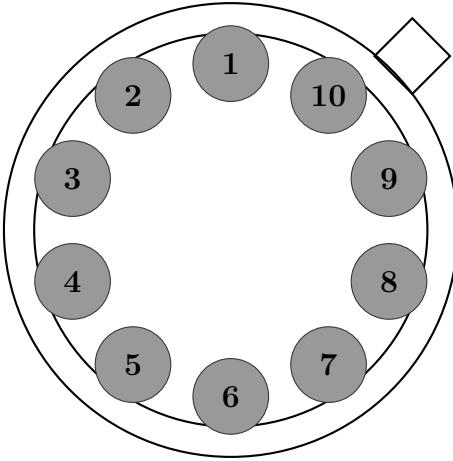


Figure 3.13: TO-5 10 bottom view

stages of the system, including signal conditioning, amplification, filtering, and data acquisition, into an efficient and compact layout. By centralizing these components, the PCB not only ensures seamless interaction between different parts of the system but also enhances the overall reliability and performance.

The design process involved careful consideration of the placement and routing of components to minimize noise, optimize signal integrity, and ensure compatibility with the sensor system's specifications. Advanced design tools were used to create the layout, taking into account thermal management, power distribution, and mechanical constraints.

However, as we had to route using single layer PCB, the routing is clearly unoptimal. First of all, the mass plan is cut out at various location, making current loops, generation EMC issues. As well the layout is not the most beautiful we could have make. Note that we mount the resistors vertically to save on place, hence, the copper screen might seem odd at first.

The final PCB design, shown in the figure 3.14 below, represents the culmination of these efforts, providing a robust and integrated platform for the operation of our sensor system.

Name	Pin number	Function
Display_SCL	A5	I <sup>2</sup> C clock
Display_SDA	A4	I <sup>2</sup> C data
RX_BLE	13	RX for the UART communication to the ??? BLE module
TX_BLE	12	TX for the UART communication to the ??? BLE module
RX_LoRA	11	RX for the UART communication to the RN2483A LoRa module
TX_LoRA	10	TX for the UART communication to the RN2483A LoRa module
rst_LoRa	7	Reset pin for the RN2483A LoRa module (low reset)
LED	6	LED control pin (PWM available)
buzzer	5	Buzzer pin (PWM available)
barre_ctrl	4	Gas sensor heating element control pin
ADC_Gas	A0	Gas senso ADC pin (post amplifier)
ADC_temp	A1	Gas sensor internal temperature pin (post conditionner)

Table 3.3: Gas sensor shield pinout

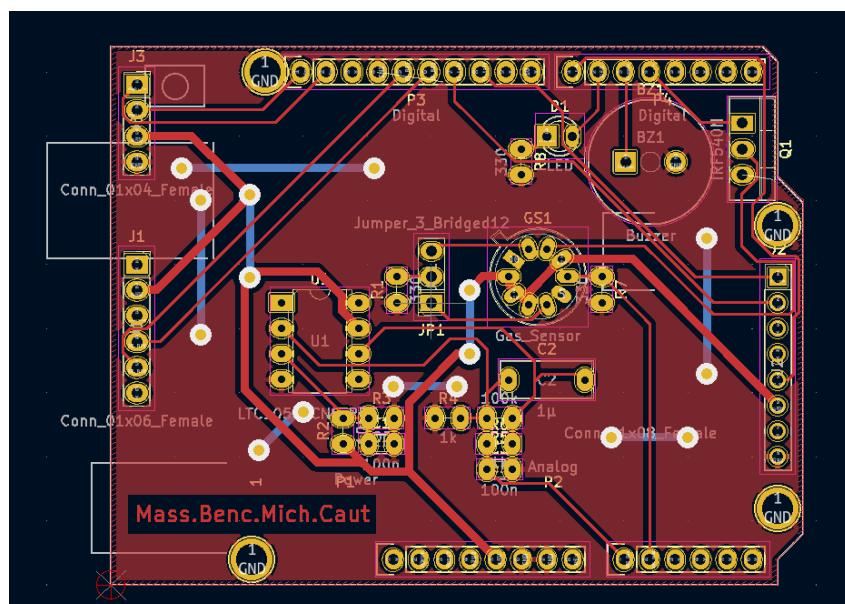


Figure 3.14: Gas sensor node shield for arduino UNO

# Chapter 4

## Conclusion

This project has been an incredible journey, guiding us through the entire process of designing and manufacturing a smart gas sensor from the ground up. We explored the chemistry behind reactive metal oxide resistors, employed photolithography to fabricate the sensor structure, and developed the signal conditioning circuit. Our goal was to integrate everything into a fully functional smart gas sensor node, transmitting data via LoRa to a Node-RED server for real-time visualization.

However, despite our efforts, we faced significant challenges. Persistent issues with Microchip's LoRa module, coupled with the impossibility of manufacturing our PCB as our course schedule tightened, ultimately stalled the project. While we successfully completed the design phase, we were unable to produce a working prototype in time. Nonetheless, the experience provided invaluable hands-on knowledge in sensor fabrication, circuit design, and IoT integration, reinforcing our skills in real-world problem-solving and project management.