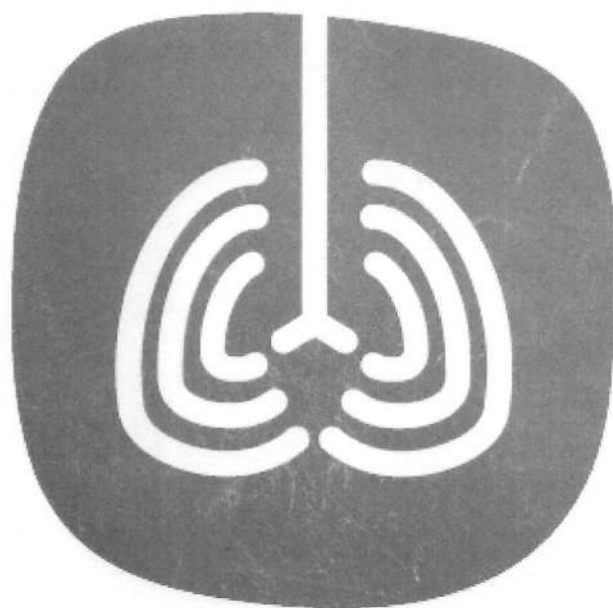


# DOSSIER TECHNIQUE

Partie commune



# MakAir

GABORIT Antonin  
BARRETEAU Paul  
ROUSSEL Mathis  
ROUET Teddy

# TABLE DES MATIÈRES

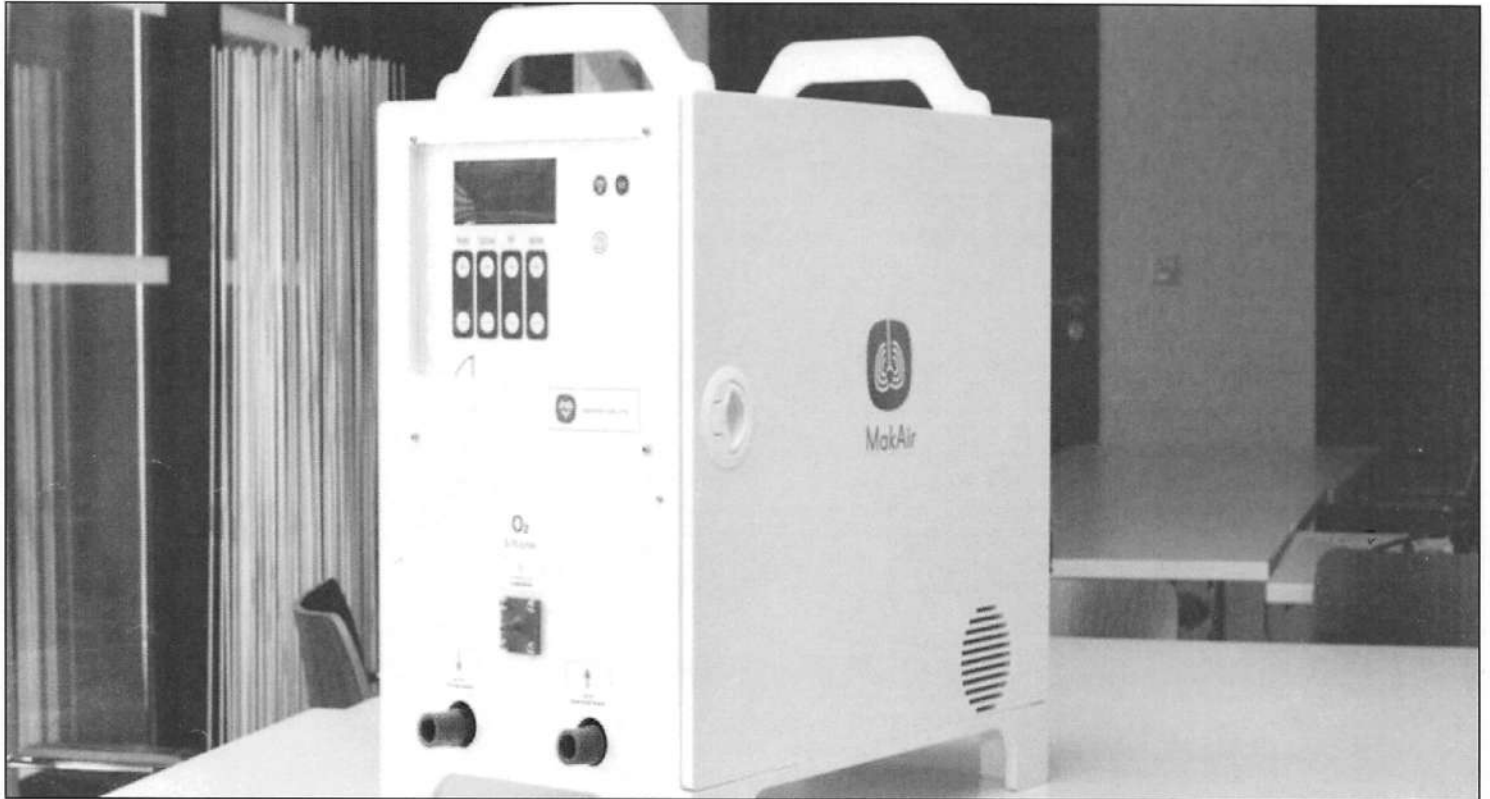
|   |           |
|---|-----------|
| <b>INTRODUCTION</b>                                 | <b>2</b>  |
| SITUATION DU PROJET DANS SON CONTEXTE INDUSTRIEL    | 2         |
| Synoptique général du système                       | 2         |
| Missions du système                                 | 3         |
| 2.1 Diagramme de contexte (bas niveau)              | 4         |
| 2.1 Diagramme de contexte (haut niveau)             | 4         |
| 2.3 Diagramme de cas d'utilisation                  | 5         |
| 2.4 Diagramme de déploiement d'exploitation         | 6         |
| 3. CATALOGUE DES ACTEURS                            | 7         |
| <b>II. Présentation du projet</b>                   | <b>8</b>  |
| Planification du GANTT(générale)                    | 8         |
| <b>III. Schéma structurelle de la carte</b>         | <b>9</b>  |
| <b>IV. Dossier de fabrication</b>                   | <b>13</b> |
| <b>V. Fiche de test</b>                             | <b>17</b> |
| V.1 Mesure de VBAT                                  | 17        |
| V.2 Test des LED alarmes                            | 19        |
| V.3 Vérification de la mesure de température        | 23        |
| V.4 Mesurer le débit                                | 27        |
| V.5 Fiche de test du capteur de pression MPXV5010DP | 28        |
| V.6 Fiche de test des moteurs FAULHABER BX4         | 29        |

# I. INTRODUCTION

---

## 1.1 SITUATION DU PROJET DANS SON CONTEXTE INDUSTRIEL

Synoptique général du système



MakAir est « *Un respirateur artificiel exclusivement dédié au traitement du Covid-19* » qui a pour but de proposer une aide d'urgence aux personnes en souffrance respiratoire sévère. MakAir a été conçu pour aider les hôpitaux à accueillir autant de patients que nécessaire en réanimation.

Ainsi, MakAir est un respirateur artificiel open source facilement reproductible, à bas coût.

- Open source (logiciel et matériel)
- Adapté au besoin
- Conçu au regard des normes

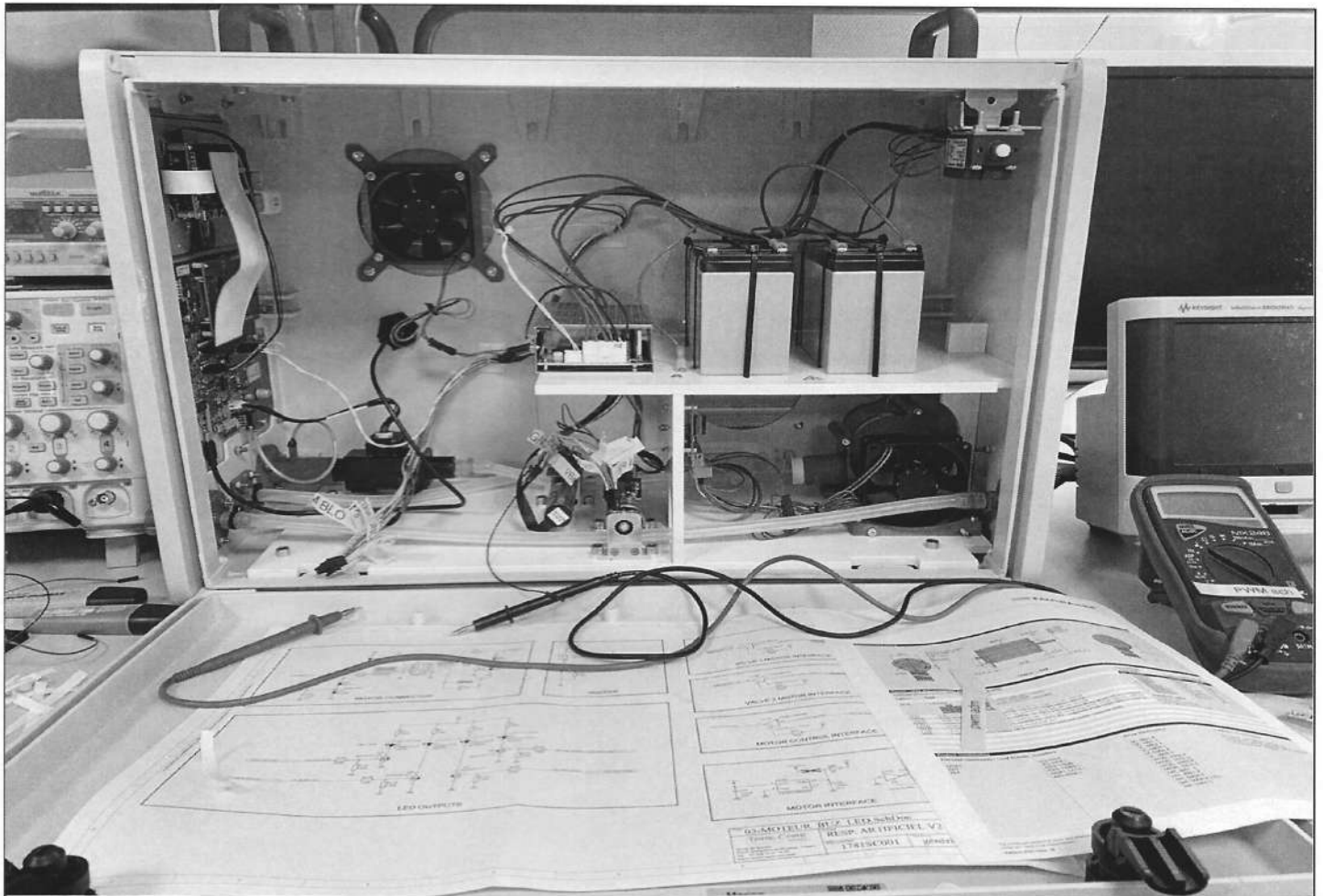
## Missions du système

Le système envisagé consiste à améliorer sur certains points, décrits ci-dessous, la version V1 qui a été prêtée à l'établissement.

Le système doit pouvoir :

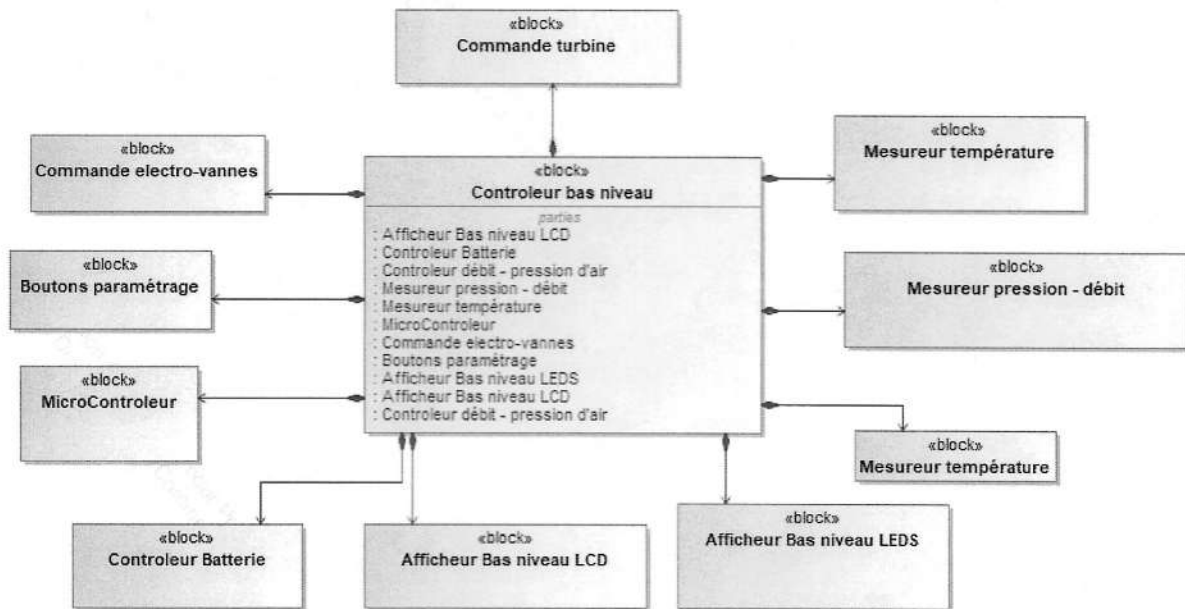
- Superviser la batterie de façon à indiquer son niveau de charge
- Assurer la régulation du débit d'air fourni au patient
- Gérer la température de l'intérieur du boîtier

Ces informations doivent pouvoir être affichées « en local » sur un écran embarqué, et sur un écran déporté plus grand et offrant plus d'ergonomie de lecture.

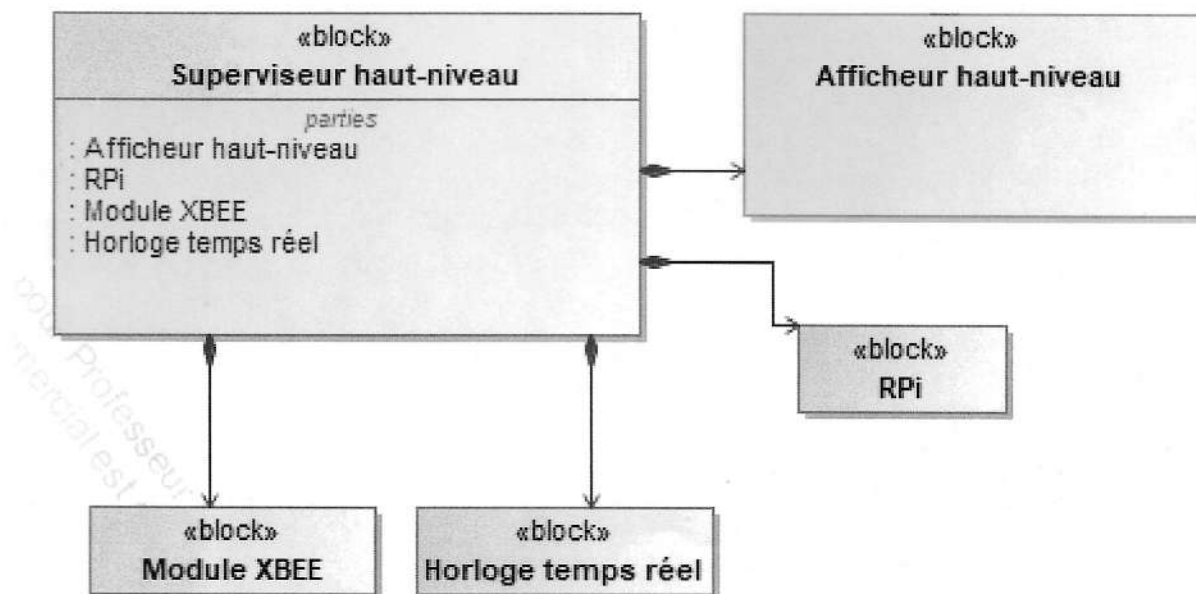


UAC -

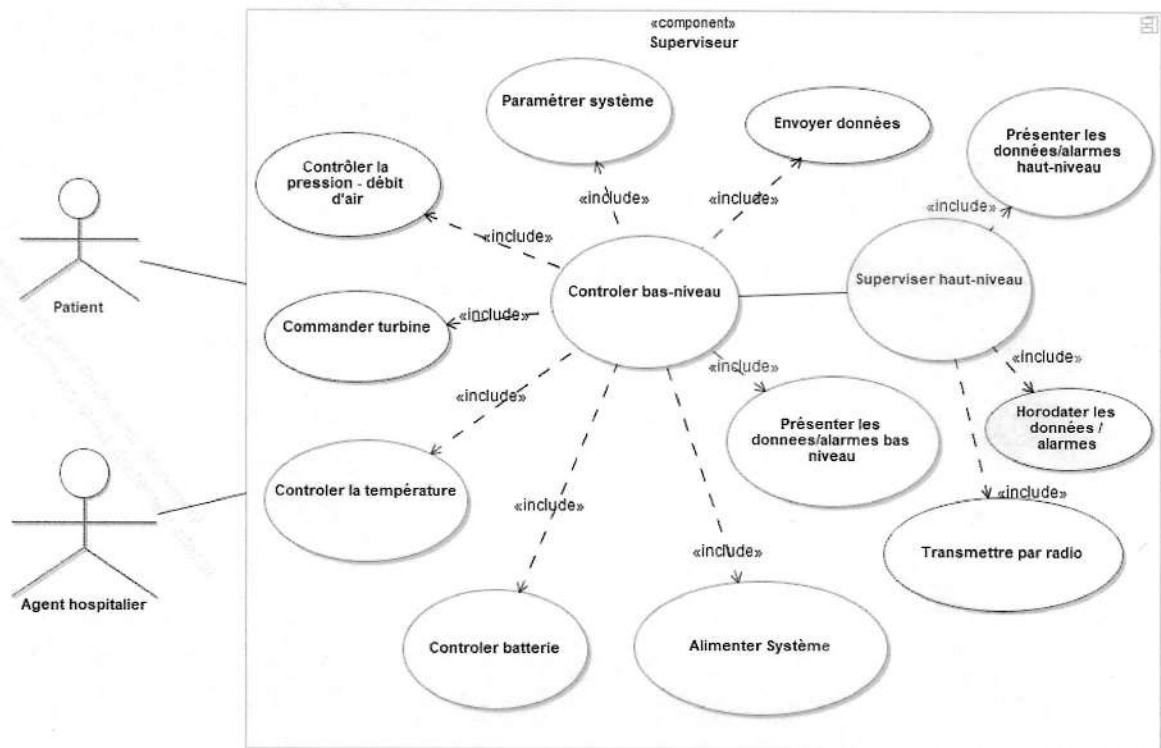
## 2.1 Diagramme de contexte (bas niveau)



## 2.1 Diagramme de contexte (haut niveau)

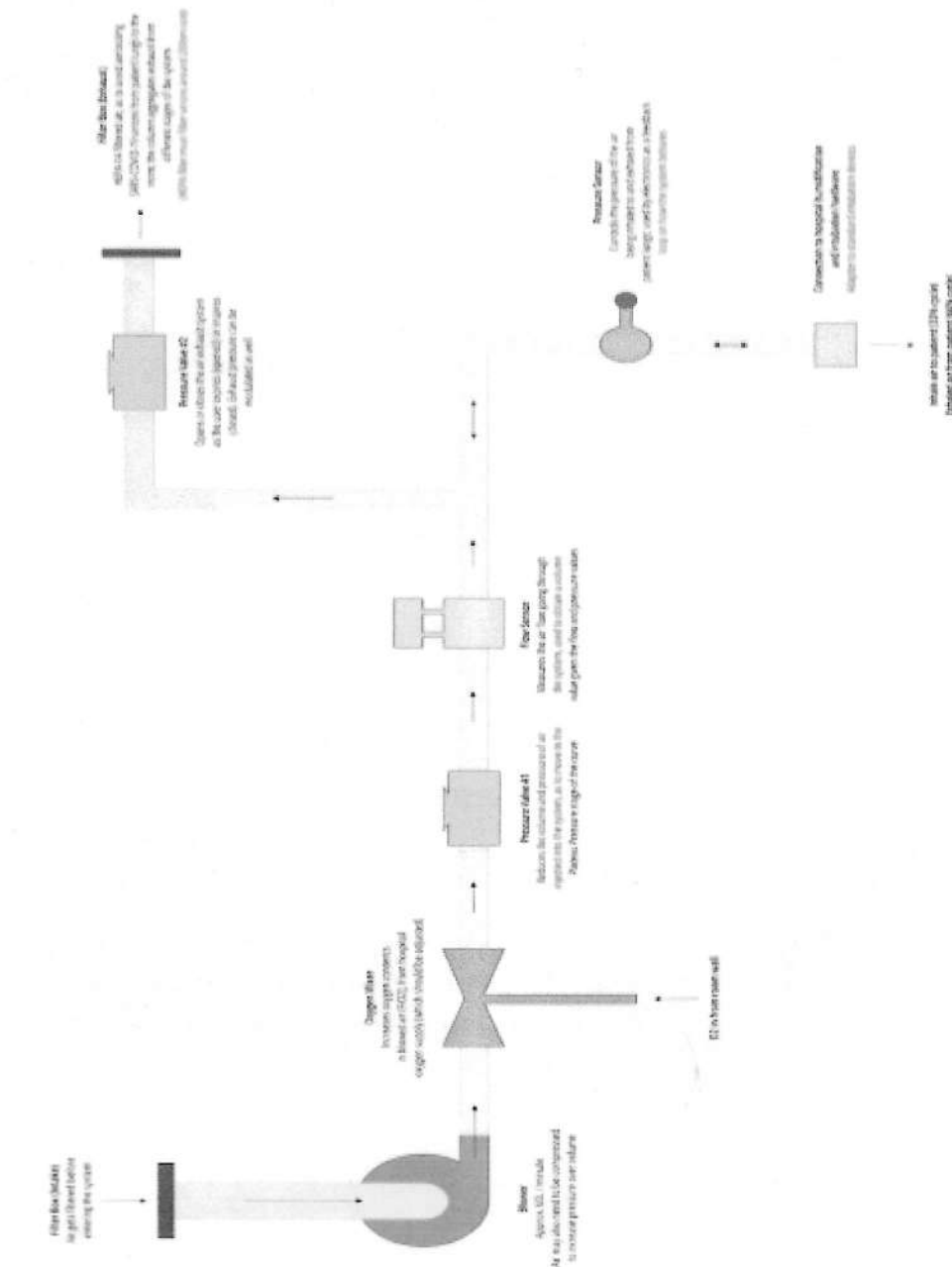


### 2.3 Diagramme de cas d'utilisation



- CATALOGUE DES ACTEURS :

| Acteurs           | Description  |
|-------------------|--|
| Agent Hospitalier | <p><i>Acteur principal :</i></p> <p>Il déclenche, manuellement, la mise en œuvre du MakAir. Il gère la relation entre le MakAir et le patient. Il observe les affichages, bas niveau ou haut niveau.</p> |
| Patient           | <p><i>Acteur secondaire :</i></p> <p>Le patient est sous assistance respiratoire. Cette assistance est fournie par le MakAir.</p>  |



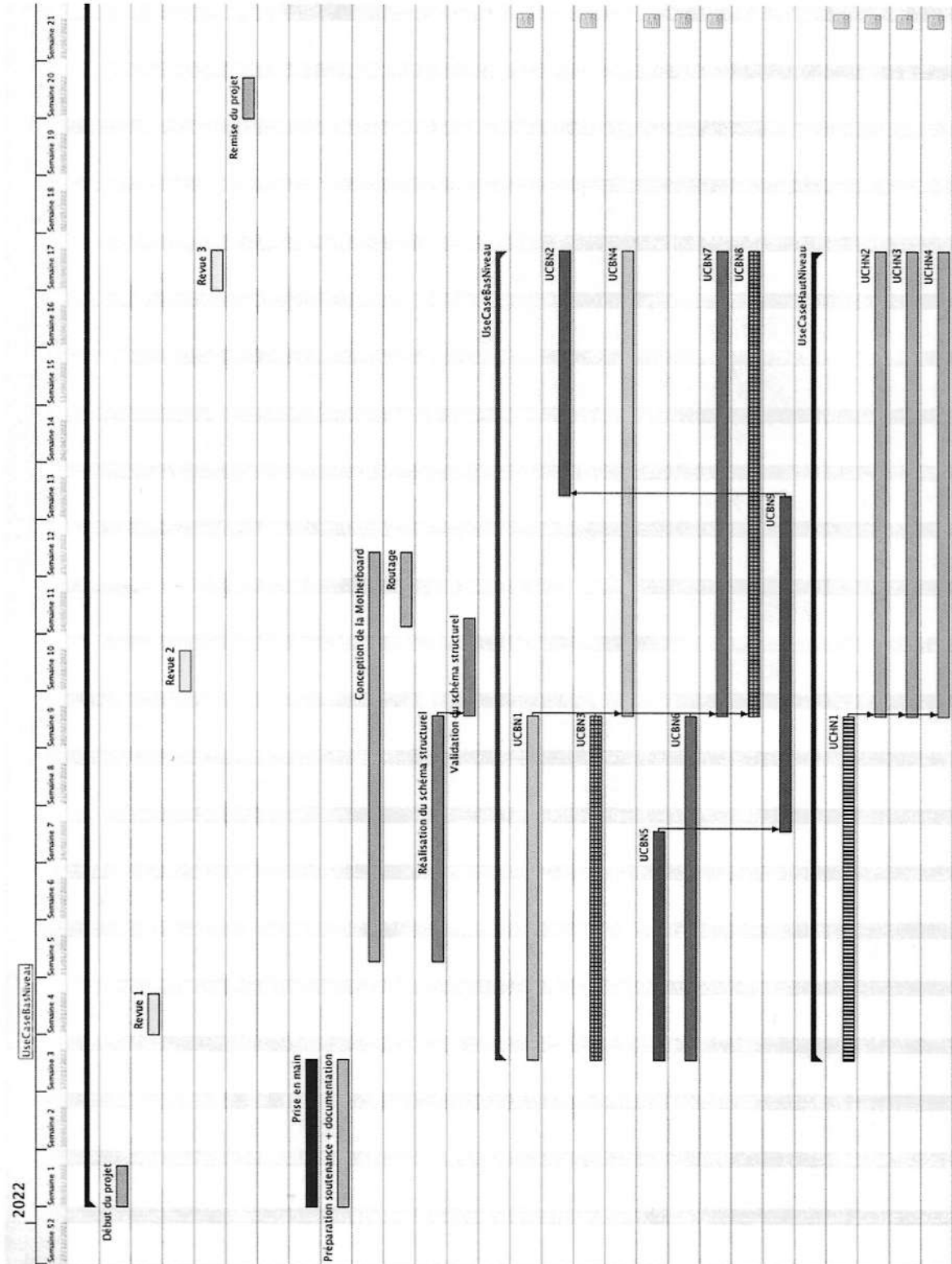
### 3. CATALOGUE DES ACTEURS

| Acteurs         | Rôles  |
|-----------------|--|
| GABORIT Antonin | <p>-&gt; Pour le <b>UCBN3</b> il est demandé de contrôler la pression et le débit d'air de manière à ce que la pression d'air soit correcte pour l'assistance respiratoire.</p> <p>-&gt; Pour le <b>UCBN8</b> il est demandé de présenter les données / alarmes bas niveau de manière à ce que Les données soit correctement présentées.</p> |
| BARRETEAU Paul  | <p>-&gt; Pour le <b>UCBN3</b> il est demandé de mesurer le débit d'air.</p> <p>-&gt; Pour le <b>UCBN4</b> il est demandé de commander la turbine (PWM).</p>  |
| ROUSSEL Mathis  | <p>-&gt; Pour le <b>UCBN2</b> il est demandé de paramétrer le système.</p> <p>-&gt; Pour le <b>UCBN5</b> il est demandé de contrôler la température de manière à ce qu'elle soit surveillée.</p> <p>-&gt; Pour le <b>UCBN9</b> il est demandé d'envoyer les données (TX).</p>  |
| ROUET Teddy     | <p>-&gt; Pour le <b>UCBN6</b> il est demandé de contrôler la batterie de manière à ce que La tension de la batterie soit correctement acquise et régulée.</p> <p>-&gt; Pour le <b>UCBN7</b> il est demandé d'alimenter le système de manière à ce que l'alimentation pour tout le système soit correcte.</p>                                 |

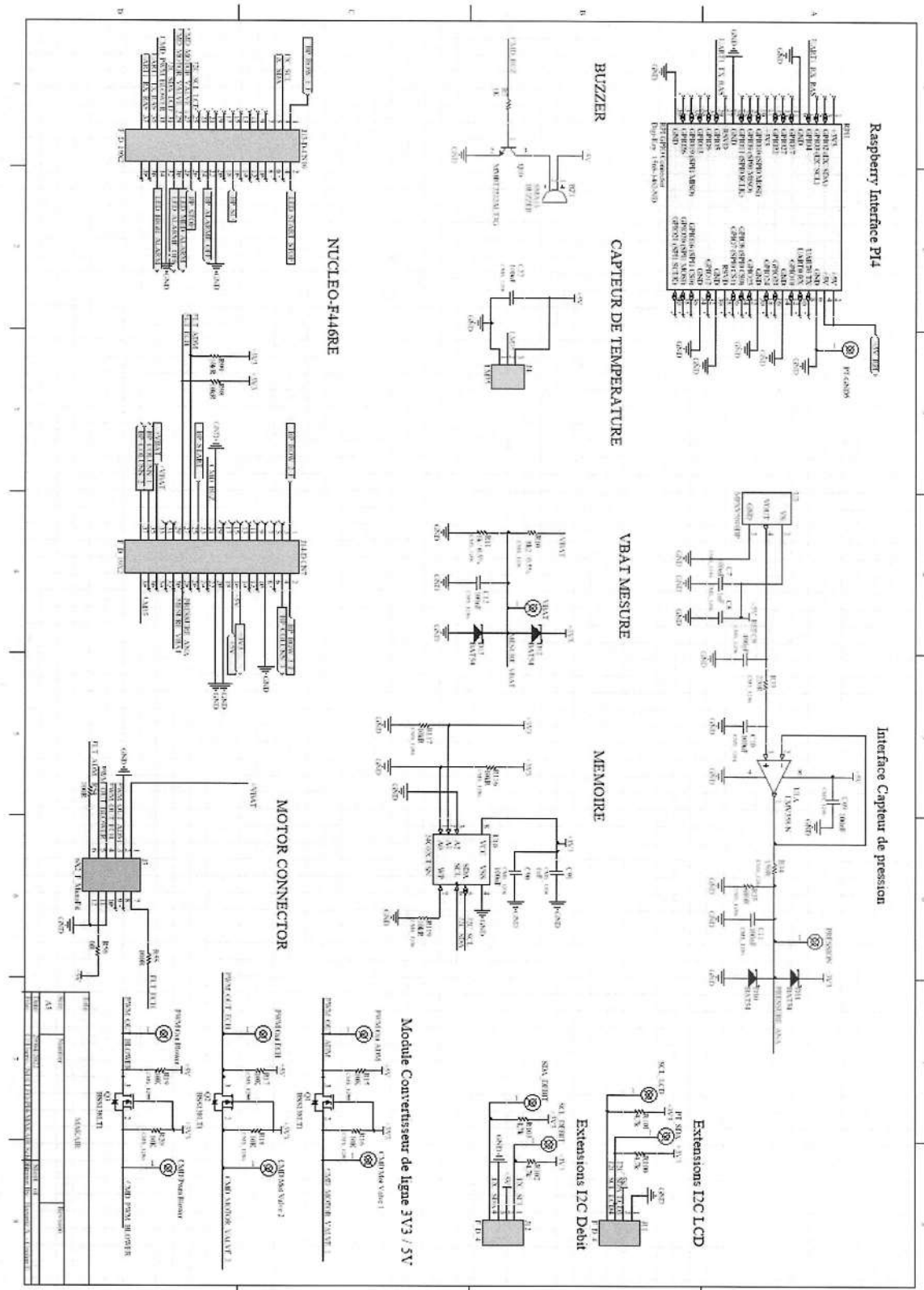


## II. Présentation du projet

### 1. Planification du GANTT(générale)



### III. Schéma structurelle de la carte

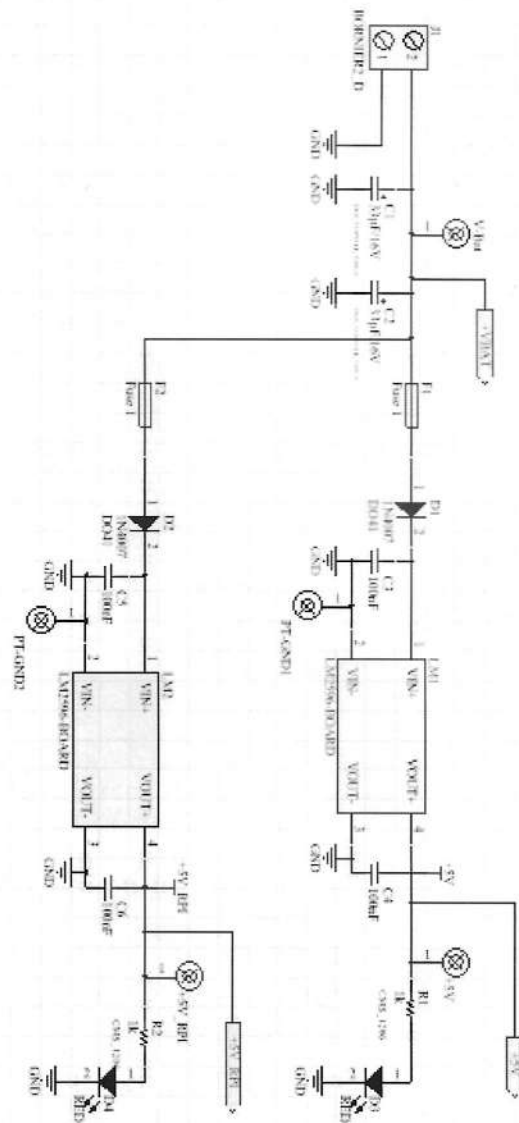


[illegible]

|       |                            |
|-------|----------------------------|
| Title |                            |
| Supp  | Number                     |
| A4    |                            |
| Date  | 25.04.2025                 |
| File  | C:\Users\Tobias.ALF\SchDoc |
|       | Sheet of                   |
|       | Pravna Ber                 |



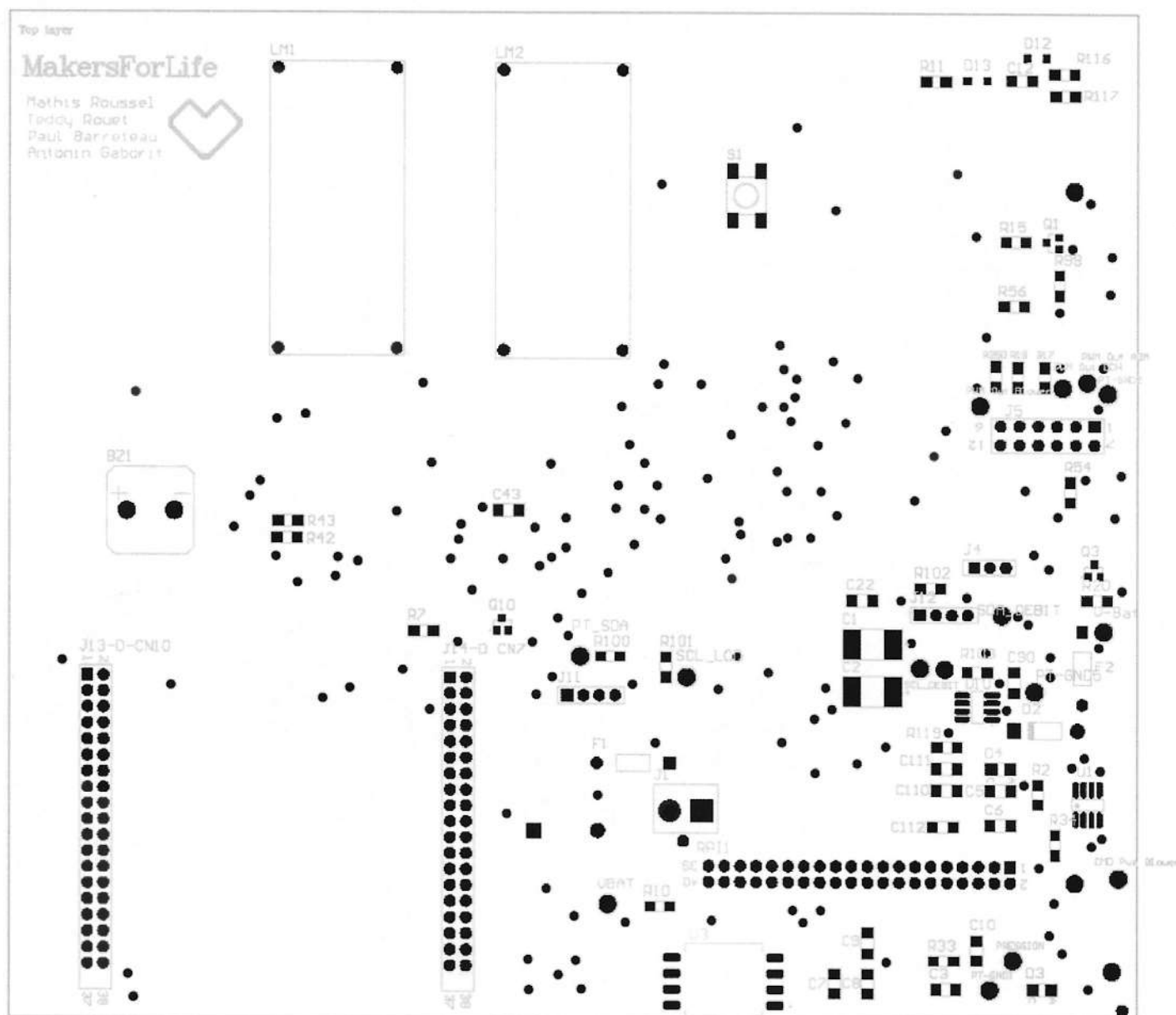
# ALIMENTATION



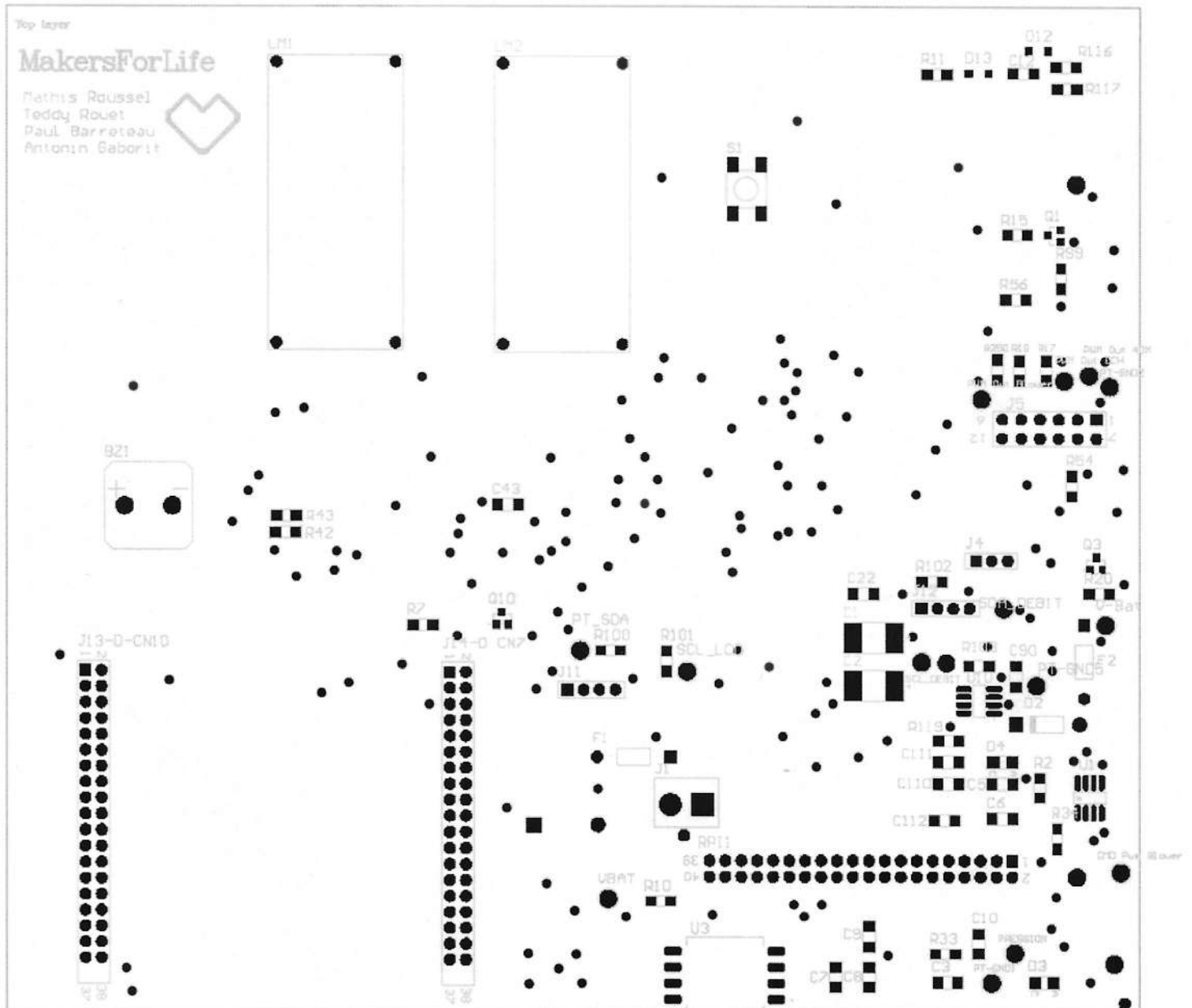
|       |        |  |
|-------|--------|--|
| Titre | Niveau |  |
| Nom   | Niveau |  |
| Mat   | Niveau |  |
| Thème | Niveau |  |
| Unité | Niveau |  |

## IV. Dossier de fabrication

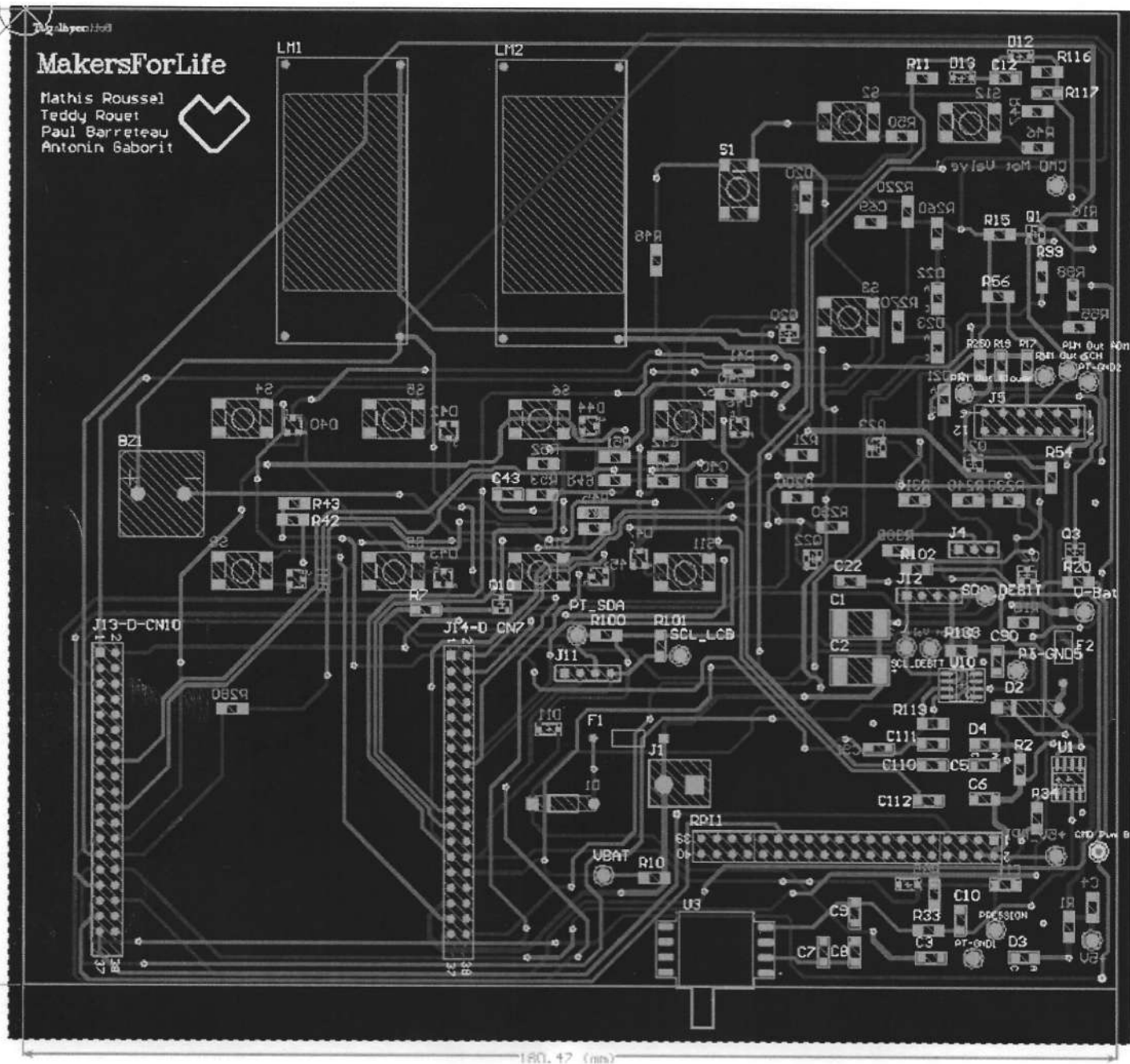
### III.1 – Implantation des composants, Côté Top



### III.2 – Implantation des composants, Côté Bottom



### III.3 – PCB (Printed Circuit Board)





### III.4 – Nomenclature

| REF  | Description  | Emplacement                                   | Quantité | Ref. Fabric                                 | Prix   | Prix Total |
|--|--|---|----------|---|--------|------------|
| SP SCHEM3<br>SV, *SV_PRL<br>T-GND1 PT-GND2<br>T-GND3 PT_GSA<br>CL_DEST1: SOL_LCD<br>DA_DEST1: V881               | POINT TEST   | POINT_TEST                                    | 9        | POINT_TEST                                  | NA     |            |
| Z1   | Buzzer SMA13   | Buzzer_SMA13                                  | 1        | Buzzer_SMA13                                | 6,38 € | 6,38 €     |
| MD Mot Valve 1<br>MD Mot Valve 2<br>MD Pwm Blower<br>PSSON: PWM Out<br>OM: PWM Out<br>Lower PWM Out<br>CH: HUBAT | POINT TEST   | POINT_TEST                                    | 3        | POINT_TEST                                  | NA     |            |
| 1_F2   | Fuse   | FUSE 1  | 2        | PIH-V2/E2 B                                 | NA     |            |
| 1  | BORNIER 2 PLOTS DROIT A VIS 5,08mm   | BORNIER2 DROIT A VIS                          | 1        | BORNIER2_A_VIS                              | NA     |            |
| 4  | BARRETTE 3 CONTACTS 1 RANGEE FEMELLE DROITE                                  | BARRETTE_3 CONTACTS_1 RANGEE FEMELLE DROITE   | 1        | BARRETTE_3_F_D                              | NA     |            |
| 8  | BARRETTE 6 CONTACTS 2 RANGEE FEMELLES DROITES                                | BARRETTE_6 CONTACTS_2 RANGEE FEMELLES DROITES | 1        | BARRETTE_6X2_F_DS                           | NA     |            |
| 11_112   | BARRETTE 4 CONTACTS 1 RANGEE FEMELLE DROITE                                  | BARRETTE_4 CONTACTS_1 RANGEE FEMELLE DROITE   | 2        | BARRETTE_4_F_D                              | NA     |            |
| 13-D-0V10_114-D<br>N7  | BARRETTE_19 CONTACTS_2 RANGEE FEMELLE DROITE                                 | BARRETTE_19 CONTACTS_2 RANGEE FEMELLE DROITE  | 2        | BARRETTE_19X2_F_D                           | NA     |            |
| 11_1M2   | CONVERTISSEUR DC/DC_BUCK_LM2596-BOARD  | LM2596-BOARD                                  | 2        | LM2596-BOARD                                | 0,51 € | 1,02 €     |
| 1_Q1_Q3_Q20<br>21_Q32_Q23<br>10  | Power MOSFET 200 mA 30V, N-Channel, 3-Pin SOT-23, Tape and Reel              | BS6518L71                                     | 7        | ONSC-SOT-23-3-318-OS_V                      | 0,10 € | 0,70 €     |
| 1_Q2   | General Purpose Transistor NPN Silicon, 3-Pin SOT-23, Pb-Free, Tape and Reel | MMBT2222ALT3G                                 | 1        | ONSC-SOT-23-3-318-OS_V                      | 0,02 € | 0,02 €     |
| 1_Q2   | DIODE DE REDRESSEMENT DO41 1N4007  | DIODE_REDRESSEMENT_DO41 1N4007                | 2        | DIODE_DO41                                  | NA     |            |
| 3_Q4_Q32   | LED CMS 1206 Rouge   | LED_CMS_1206                                  | 3        | LED_CMS_1206                                | NA     |            |
| 23   | LED CMS 1206 Jaune   | LED_CMS_1206                                  | 1        | LED_CMS_1206                                | NA     |            |
| 20_Q21   | LED CMS 1206 Vert  | LED_CMS_1206                                  | 2        | LED_CMS_1206                                | NA     |            |
| 40_Q41_Q42_Q43<br>44_Q45_Q46_Q47   | DIODE CMS COMMUTATION RAPIDE SOT23 BAS16                                     | DIODE_CMS_COMMUTATION_RAPIDE_SOT23 BAS16      | 3        | DIODE_SOT23                                 | 0,11 € | 0,88 €     |
| 10_Q11_Q12_Q13   | Schottky Rectifier   | BAT54   | 4        | 1N4148W                                     | 0,10 € | 0,40 €     |
| 3_Q4_Q5_Q6_Q7_Q10<br>11_Q12_Q13_Q40_Q41<br>42_Q43_Q69_Q90<br>110_Q111_Q112                                       | CONDENSATEUR CMS CERAMIQUE 1206 /"100nF"/                                    | CMS_1206                                      | 13       | Magasin St Gob                              | NA     |            |
| 8_Q91  | CONDENSATEUR CMS CERAMIQUE 1206 /"1uF"/                                      | CONDENSATEUR_CMS_1206                         | 1        | Magasin St Gob                              | NA     |            |
| 9  | CONDENSATEUR CMS CERAMIQUE 1206 /"490pF"/                                    | CONDENSATEUR_CMS_1206                         | 1        | Magasin St Gob                              | NA     |            |
| 1_Q2   | DIODE DE REDRESSEMENT DO41 1N4007 /"33uF/16V"/                               | CMS_TANTALE_T343-S1                           | 2        | CONDENSATEUR_CMS_POLARISE_TANTALE_D/T343-S1 | NA     |            |

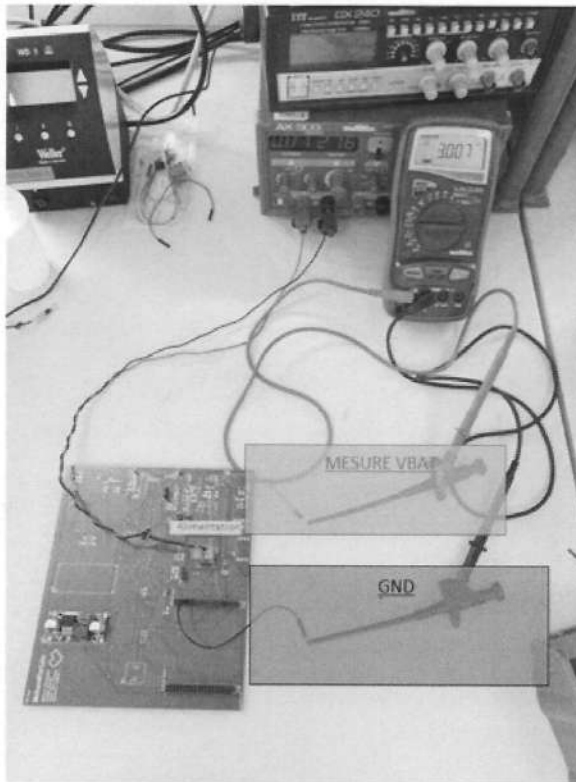
|   |  |                                |    |                      |         |         |
|---|--|--------------------------------|----|----------------------|---------|---------|
| 3 R16 R17 R18 R19<br>3 R21 R41 R43 R48<br>7 R49 R51 R53 R58<br>9 R116 R117 R119<br>40 R190 R310 | RESISTANCE CMS 1206, /*10K ohm*/   | CMS_1206                       | 22 | Magasin St Gab       | NA      |         |
| 3 R42 R44 R46 R48<br>3 R52 R54 R56<br>20 R220 R230<br>40 R250 R260                              | RESISTANCE CMS 1206, /*100 ohm*/   | RESISTANCE_CMS_1206            | 15 | RESISTANCE_CMS_1206  | NA      |         |
| 50 R270   | RESISTANCE CMS 1206, /*180 ohm*/   | RESISTANCE_CMS_1206            | 2  | Magasin St Gab       | NA      |         |
| 1   | RESISTANCE CMS 1206, /*120 ohm*/   | RESISTANCE_CMS_1206            | 1  | Magasin St Gab       | NA      |         |
| 4   | RESISTANCE CMS 1206, /*150 ohm*/   | RESISTANCE_CMS_1206            | 1  | Magasin St Gab       | NA      |         |
| 1   | RESISTANCE CMS 1206, /*680 ohm*/   | RESISTANCE_CMS_1206            | 1  | Magasin St Gab       | NA      |         |
| R2 R7 R11   | RESISTANCE CMS 1206, /*1Kohm*/   | RESISTANCE_CMS_1206            | 4  | Magasin St Gab       | NA      |         |
| 3   | RESISTANCE CMS 1206, /*8K2 ohm*/   | RESISTANCE_CMS_1206            | 1  | Magasin St Gab       | NA      |         |
| 1   | RESISTANCE CMS 1206, /*0 ohm*/   | RESISTANCE_CMS_1206            | 1  | Magasin St Gab       | NA      |         |
| 1   | GPIO Header for Raspberry Pi 4+/5+/Pi 2/Pi 3 - 2x20  | RPi GPIO Connector             | 1  | rpil                 | 1,62 €  | 1,62 €  |
| 52, 53, 54, 55, 56,<br>58, 59, 510, 511,<br>1   | BOULTON POUSSOIR MINIATURE CMS ON/OFF  | BOULTON_POUSSOIR_MINIATURE_CMS | 12 | BOULTON_POUSSOIR_CMS | 0,56 €  | 0,56 €  |
| A   | DS91C03 Wide-Bandwidth High-Output-Drive Operational Amplifier, 4.5 to 16 V, -40 to 125 degC, 8-pin SOIC (DS), Green (RoHS & no Sb/Br) | LMV358-N                       | 1  | T-DS_N               | 0,60 €  | 0,60 €  |
|   | Pressure Sensor 1.45Psi_10KPa_Vented Gauge Male - 0.13_3.17mm_Tube 0.2 V 4.7 V S-SMD, Gull Wing, Side Port                             | MPXV50100P                     | 1  | SOIC254P1821X762-SN  | 30,00 € | 30,00 € |
| D   | 2Kbit, 400KHz, 5V, 12C Serial EEPROM, 8-Pin SOIC 150mil, Commercial Temperature Tape and Reel  | 24C02CT/5N                     | 1  | SOIC-SN8_N           | 0,27 €  | 0,27 €  |
|   |  |                                |    | Total                | 40,27 € | 42,27 € |

## V. Fiche de test

### V.1 Mesure de VBAT

#### 1. test physique

##### Test n°1 :



Ce test consiste à vérifier si le pont diviseur de l'atténuation fonctionne

Pour cela j'ai eu besoin d'un voltmètre avec le « + » brancher à la pin « Vbat mesure » et le – brancher sur le GND de la carte.

La carte est alimentée grâce à l'alimentation pour simuler la décharge de la batterie.

On peut effectuer un premier calcul pour vérifier si le pont diviseur est fonctionnel

$$\text{Mesure } \underline{V_{bat}} = (R11 / (R11 + R10)) * \underline{V_{lim}}$$

$$\text{Mesure } \underline{V_{bat}} = (1k / (1k + 8k2)) * 27,6V$$

$$\text{Mesure } \underline{V_{bat}} = 3V$$

##### Test n°2 :



Test avec l'alimentation baissée à 15V

Le voltmètre affiche 1,633V

$$\text{Mesure } \underline{V_{bat}} = (R11 / (R11 + R10)) * \underline{V_{lim}}$$

$$\text{Mesure } \underline{V_{bat}} = (1k / (1k + 8k2)) * 15V$$

$$\text{Mesure } \underline{V_{bat}} = 1,630V$$

Ce programme permet de récupérer le résultat obtenu de la pin Mesure VBAT dans la nucléo et de la transformer en pourcentage pour l'afficher sur un afficheur LCD.

```
#include <Arduino.h>

// 1Les bibliothèques et création d'objets
#include <Wire.h> // ajoute la bibliothèque pilotant le bus I2C
#include <LiquidCrystal_I2C.h> // ajoute la bibliothèque pilotant l'écran
LCD I2C
#define MESURE_VBAT PA1

void setup() {
  Serial.begin(9600);
  delay(5100);
}

void loop() {

  analogReadResolution(12);
  float x = analogRead(MESURE_VBAT); //pin A1
  x = (x * 0.8056640625)/100*0.9283018869; //Q = 3.3/2^12 = 0.805 // /1000 *
9.283018869
  // 100 pour passer du ms en s // 0.92 pour être le plus précis possible
  int y = (x * 100) / 27.6; // mettre le résultat en %

  Serial.println(x);
  Serial.println(y);
}
```

Le programme est composé :

- D'une transformation en résolution de 12 bits
- D'une variable x qui indique la tension reçue de la pin PA1
- D'une variable y qui transforme le résultat x en pourcentage avec un calcul en croix  $((x*100)/27,6)$ .
- D'un affichage avec le terminale `Serial.println()`

## V.2 Test des LED alarmes

### 1. Programme des LED

Ce programme permet de piloter les LED ont fonction de la variable « x » (la tension de Vbat mesure)

4 leds sont commander 2 vertes, 1 rouge et 1 jaune.

```
#define LED_START PC8
#define LED_MED PB14
#define LED_ALARME_OFF PB13
#define LED_HIGH_ALARME PC4

void setup()
{
  Serial.begin(9600);
  pinMode(LED_START, OUTPUT);
  pinMode(LED_MED, OUTPUT);
  pinMode(LED_ALARME_OFF, OUTPUT);
  pinMode(LED_HIGH_ALARME, OUTPUT);

  delay(5100);
}

void loop()
{
  if (x > 0.8)
  {
    digitalWrite(LED_START, HIGH);
  }
  else
  {
    digitalWrite(LED_START, LOW);
    digitalWrite(LED_ALARME_OFF, HIGH);
    delay(1000);
  };

  ///Allumage LED MED///

  if (x >= 15 && x <= 22)
  {
    digitalWrite(LED_MED, HIGH); //alarme moyenne propriété si tension < 22
  }
  else
  {
    digitalWrite(LED_MED, LOW);
    digitalWrite(LED_ALARME_OFF, HIGH);
    delay(1000);
  }
}
```

```

////Allumage LED HIGH////

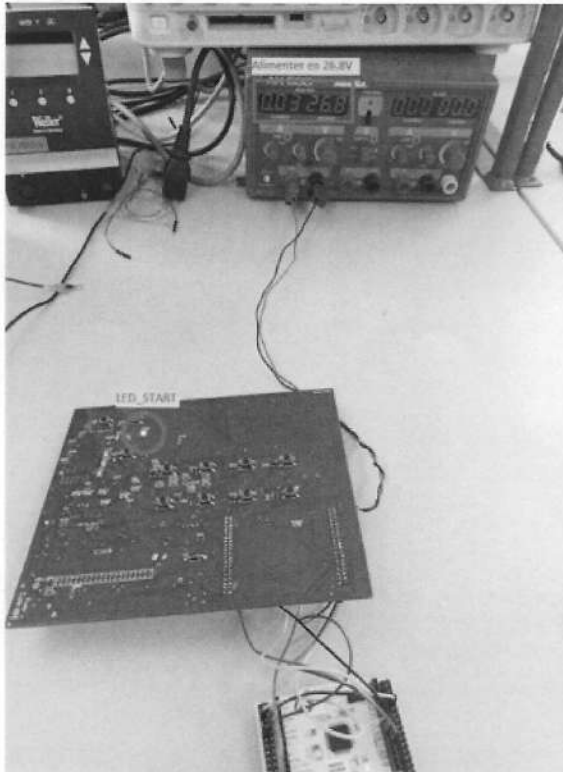
if (x <= 15)
{
    digitalWrite(LED_HIGH_ALARME, HIGH); //alarme moyenne propriété si tension
< 22
}
else
{
    digitalWrite(LED_HIGH_ALARME, LOW);
    digitalWrite(LED_ALARME_OFF, HIGH);
    delay(1000);
};

digitalWrite(LED_ALARME_OFF, LOW);
delay(300);
}

```

| LED             | Condition   |
|-----------------|---|
| LED_START       | Est allumé si la carte est alimenté(tension supérieur à 0,8V                                      |
| LED_MED         | Est allumé seulement si l'alimentation est à sa limite (tension inférieur à 22V au lieu de 27,6V) |
| LED_HIGH_ALARME | Est allumé lorsque l'alimentation devient critique (tension inférieur à 15V)                      |
| LED_ALARME_OFF  | Est allumé si chaque alarme est éteinte   |

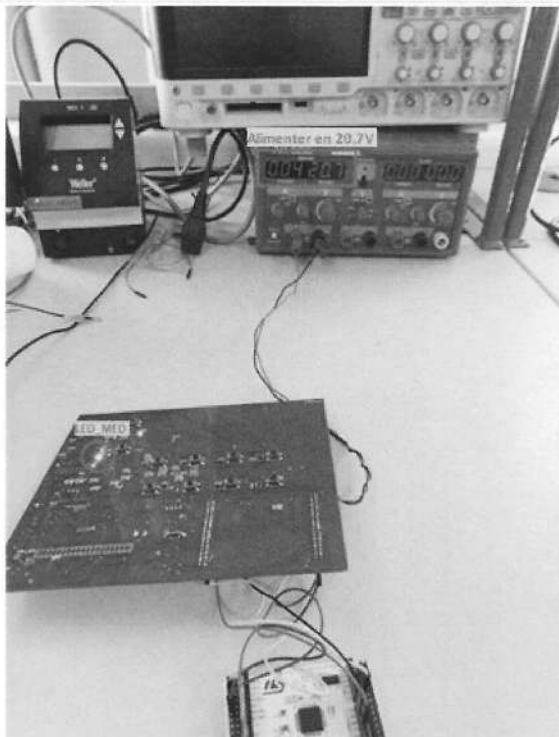
## 2. Test physique



LED\_START

Tension d'alimentation : 26,8V

Vérification que la LED\_START soit la seul à s'allumé pour indiquer que le MAKAIR est alimenter.

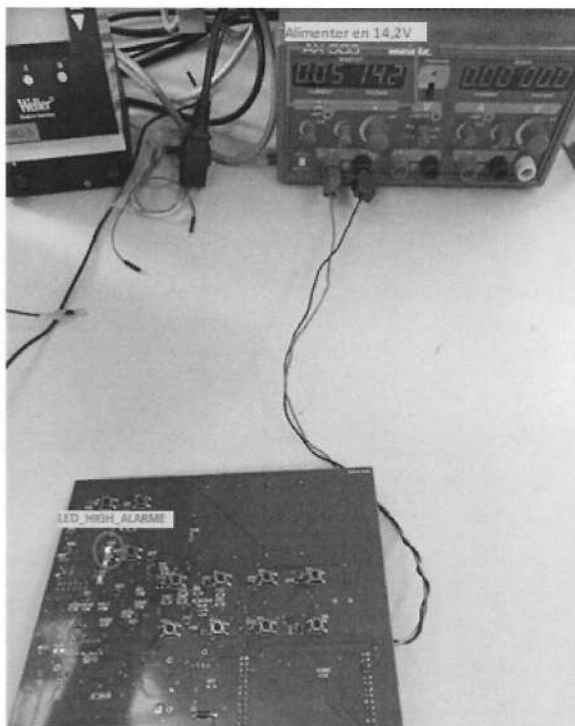


#### LED\_MED

Tension d'alimentation : 20,7V

Vérification de la condition est respecter.

Lorsque la tension descend en dessous de 22V la LED\_MED est allumé sans allumé la LED\_HIGH\_ALARM



#### LED\_HIGH\_ALARM

Tension d'alimentation : 14,2V

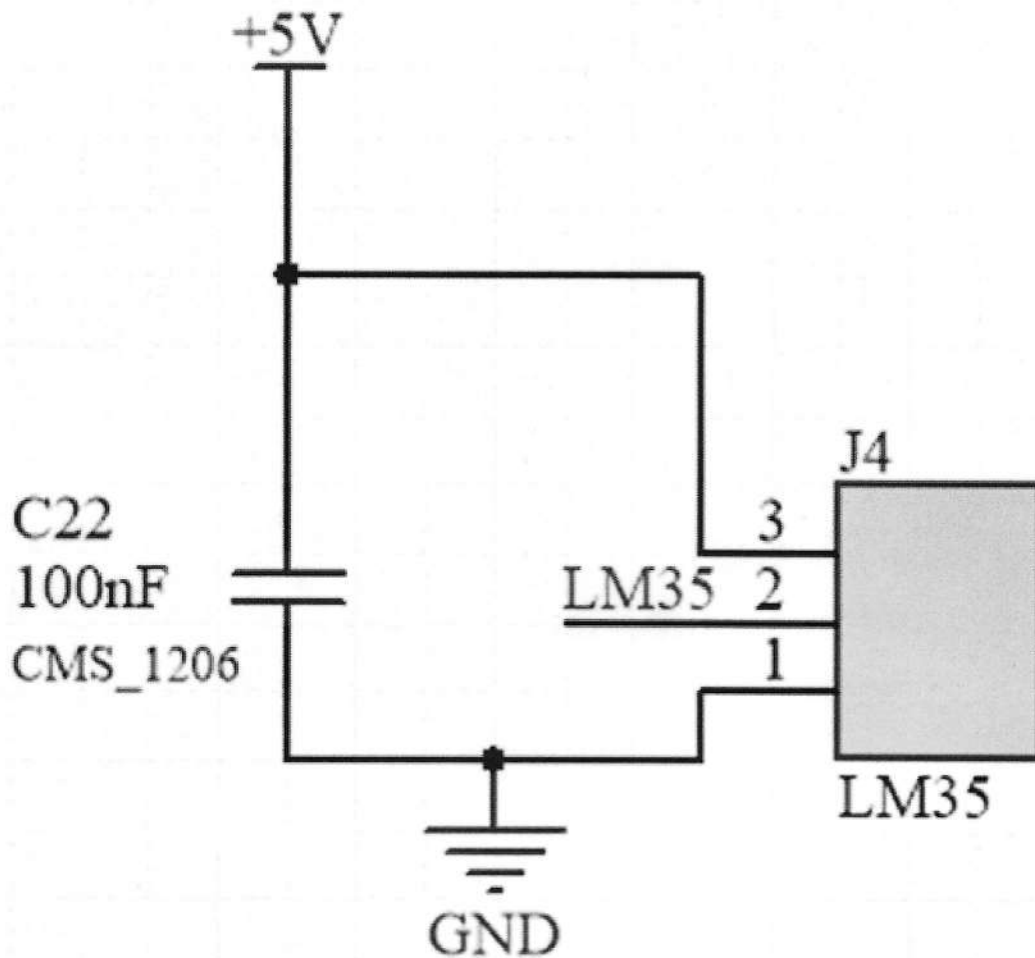
Vérification de la condition est respecter.

Lorsque la tension descend en dessous de 15V la LED\_HIGH\_ALARM est allumé



### V.3 Vérification de la mesure de température

Schéma structurel concernée par le test :



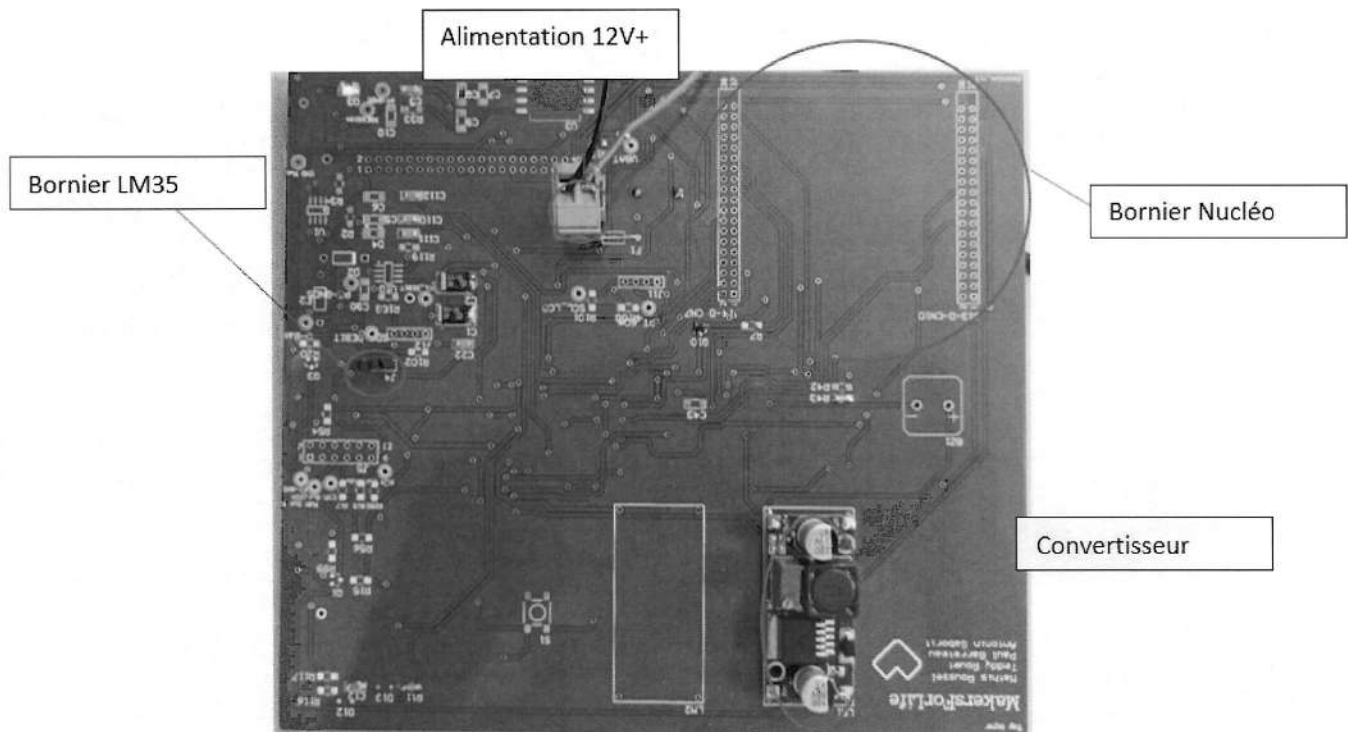
Ce test a un objectif:

- Vérifier la température dans le MakAir

Matériel nécessaire :

- La carte mère
- Une alimentation de laboratoire (tension de +12V nécessaire)
- Une Nucléo
- Visual Code
- Un thermomètre





- 1) Allumez l'alimentation de laboratoire, puis régler la consigne à +12V
- 2) Visser les deux fils sur le bornier, puis les brancher sur l'alimentation en respectant les polarités.
- 3) Brancher la carte Nucléo sur son bornier
- 4) Compiler le programme pour la conversion de température sur la nucléo
- 5) Contrôler la valeur de la température sur la nucléo par l'interface serial
- 6) Vérification au thermomètre

Configuration Platform IO pour utiliser l'environnement Visual Studio.

Platform : défini la plateforme utiliser (Stm32)

Board : Défini la carte Utiliser (Nucleo F446RE)

Framework : environnement utiliser (Arduino)

Monitor\_speed : définir la vitesse de transmission séries

Upload\_protocol : définir le système d'exploitation

```
[env:nucleo_f446re]
platform = ststm32
board = nucleo_f446re
framework = arduino
monitor_speed = 9600

;upload_protocol = stlink
upload_protocol = mbed
```

Void setup() :

Serial.begin(9600) : Il permet de définir la vitesse de transmission séries avec le PC et en lien avec la configuration fait sur « platform IO ».

```
void setup() {  
  // Initialise la communication avec le PC  
  Serial.begin(9600);  
  
  // Mappage des Pins SDA et SCL  
  Wire.setSDA(PB9);  
  Wire.setSCL(PB8);  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  analogReadResolution(12); // Conversion AnalogRead 10 bits --> 12 bits  
  int valeur_brute = analogRead(PA_0); // Lecture du port A0  
  float temperature_celcius = (float) valeur_brute * (3.3 / 4096.0 * 100.0);  
  Serial.println(temperature_celcius); // Afficher la valeur  
  
  delay(1000);  
}
```

Void loop ()

La fonction « analogReadResolution (12) » permet de modifier la fonction qui est en 10 bits vers du 12 bit

« AnalogRead » permet de faire la lecture de la valeur du capteur en brut.

La variable temperature\_celcius définit en float pour avoir les nombres après la virgule permet le calcul de la température.

La fonction Serial.println permet d'afficher la variable « temperature\_celcius » pour l'afficher dans le port Serial

La fonction delay permet d'afficher toute les 1 secondes la température en lien avec la demande du cahier des charges.

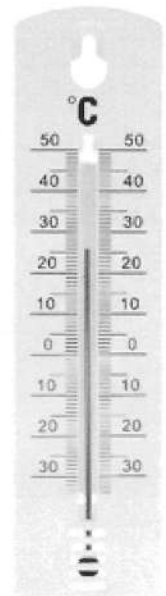
Terminal Série :

La température est bien affichée et subit des modifications lorsqu'une pression est appliquée sur le capteur.

```
PROBLÈMES  SORTIE  TERMINAL  
26.67  
27.39  
27.31  
26.75  
27.47  
26.99  
27.47  
27.47  
26.43  
26.83  
27.15  
26.75  
26.91  
27.39  
26.43  
26.75  
26.75  
27.63  
28.20  
29.33  
29.97  
30.53  
31.18  
31.02  
31.02  
30.70  
30.78  
30.29  
  
Valeur stable  
  
Pression sur le capteur
```

Pour une vérification avancée de la stabilité du capteur de température, un test à l'aide d'un thermomètre a été effectué en même temps que le test sur Visual studio et dans le même environnement.

Après un test effectué sur une plage de temps de 1 heure pour permettre au thermomètre de s'acclimater à la température de la salle, le capteur de température et le thermomètre donnait la même température.



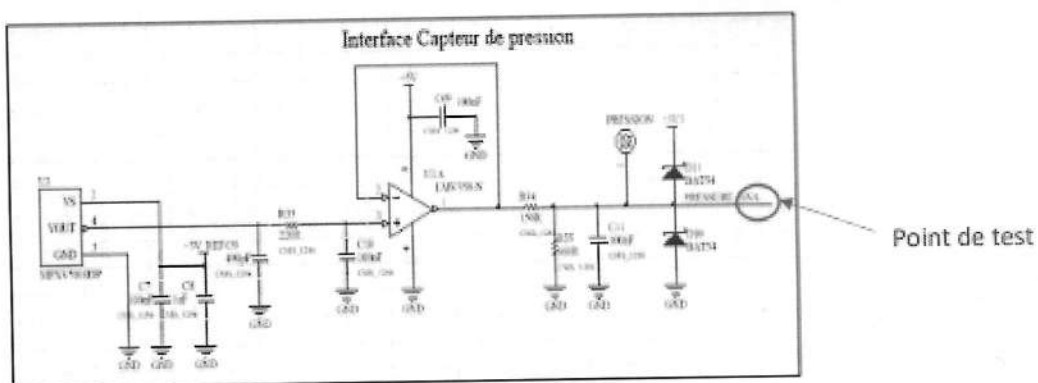
## V.4 Mesurer le débit

### Fiche de test du débitmètre

| Élément testé :              | Débitmètre   |   |   |
|------------------------------|--|---|---|
| Objectif du test :           | Vérifier son intégration dans la carte<br>Vérifier son fonctionnement  |   |   |
| Nom du testeur :             | Barreteau Paul   | Date : 10/05/2022                                       |   |
| Moyens mis en œuvre :        | Logiciel : Visual Studio code  | Matériel : Débitmètre                                   | Outil de développement : Visual Studio code |
| Réalisation :                | <p>Procédure du test : Au début du test j'ai commencé par tester le débitmètre sur une carte nucléo seule afin de tester de son fonctionnement.</p> <p>Puis après avoir reçus la carte et soudé les composants nécessaires au fonctionnement de la partie I2C j'ai connecté le débitmètre pour ensuite regarder les trames à l'aide d'un oscilloscope et pour vérifier le bon fonctionnement de mon code</p> |   |   |
| Élément testé                | Résultat attendu   | Résultat obtenu   | Validation (O/N)                            |
| Intégration sur la carte     | Intégration sans problème  | Bonne intégration mais problème d'alimentation du 3,3v  | OUI   |
| Fonctionnement du débitmètre | Bon fonctionnement   | Fonctionne correctement et réponds aux attentes du code | OUI   |
| Conclusion du test :         | Le test m'a permis de tout vérifier. Le fonctionnement du débitmètre avec son code et son intégration sur la carte sont réussis. Il faudrait comme amélioration une alimentation directe du bornier en 3V3   |   |   |

### V.5 Fiche de test du capteur de pression MPXV5010DP

|                       |   |  |   |                          |
|-----------------------|---|--|---|--------------------------|
| Élément testé :       |   | Capteur MPXV5010DP   |   |                          |
| Objectif du test :    |   | Relevés de la pression   |   |                          |
| Nom du testeur :      |   | GABORIT Antonin  | Date : 20/05/22                                       |                          |
|                       |   |  |   |                          |
| Moyens mis en œuvre : |   | Logiciel : Visual Studio Code  | Matériel : Multimètre + pinces de test + oscilloscope | Outil de développement : |
| Procédure du test :   |   |  |   |                          |
| Id                    | Description du vecteur de test                                  | Résultat attendu   | Résultat obtenu                                       | Validation (O/N)         |
|                       | Mise sous tension du MakAir                                     | Le capteur est bien sous tension et en fonctionnement.   | Capteur fonctionnel                                   | O                        |
|                       | Relevés de pression pendant la phase d'inspiration + expiration | La valeur de la tension en sortie du capteur est acquise   | La tension est acquise                                | O                        |
|                       | Calcul de la pression   | Tension de sortie converti en pression grâce à la fonction de transfert.   | La valeur de la tension est interprétée               | O                        |
| Conclusion du test :  |   | Ce test m'a permis d'interpréter la tension sur la broche de sortie Vs du capteur, pour pouvoir calculer la pression en mmH2O. |   |                          |



# V.6 Fiche de test des moteurs FAULHABER BX4

|   |   |  |                               |                  |
|---|---|--|-------------------------------|------------------|
| Élément testé :   | Moteur FAULHABER CC   |  |                               |                  |
| Objectif du test :  | Observer la variation de vitesse                            |  |                               |                  |
| Nom du testeur :  | GABORIT Antonin   | Date : 20/05/22  |                               |                  |
| Moyens mis en œuvre :   | Logiciel : Visual Studio Code                               | Matériel : Multimètre + pinces de test + oscilloscope  | Outil de développement :      |                  |
| Procédure du test :   |   |  |                               |                  |
| <ul style="list-style-type: none"><li>- Souder un câble sur les broches PWM ECH/ADM du bornier</li><li>- Brancher les pinces sur les fils et câbler les sondes sur l'oscilloscope</li><li>- Mise sous tension du MakAir et de la carte</li><li>- Observation des signaux en concordance des temps</li></ul> |   |  |                               |                  |
| Id  | Description du vecteur de test                              | Résultat attendu   | Résultat obtenu               | Validation (O/N) |
|   | Observation des signaux et la variation du rapport cyclique | Le rapport cyclique varie quand une électrovanne s'actionne  | Variation du rapport cyclique | O                |
| Conclusion du test :  |   | Pour conclure, le test a été satisfaisant et a permis de pouvoir observer l'actionnement des électrovannes en concordance des temps et mieux comprendre leurs fonctionnements. |                               |                  |