

Stick Nitrox et Trimix Arduino :

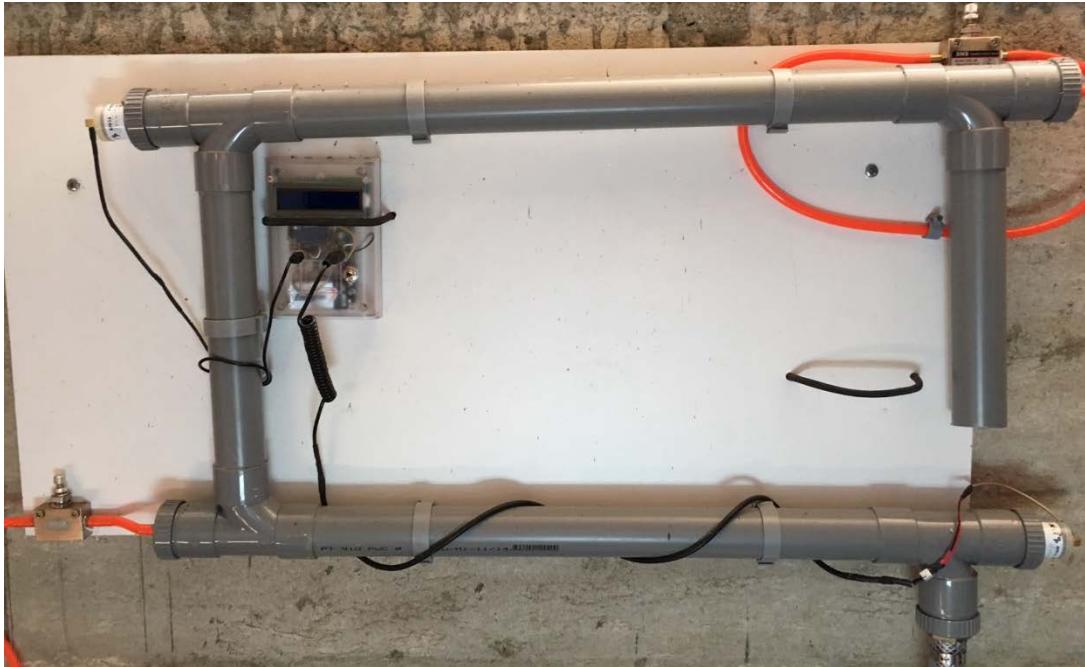


Principe :

- Utilisation de 2 cellules oxygène pour mesurer le taux d'O₂ du mélange en 2 points du stick : mesure après injection d'hélium, puis mesure après injection d'oxygène (60€), de ces 2 mesures on peut déduire le pourcentage d'oxygène et d'hélium du mélange fabriqué
- Utilisation d'une carte Arduino pour gérer logiciellement le calibrage des capteurs, et afficher le mélange fabriqué et le MOD associé (3€)
- Utilisation d'une carte ADS1115 pour mesurer précisément des tensions entre 5mV et 100mV (ce que ne sait pas faire la carte Arduino nativement) (3€)
- Affichage sur un écran LCD 16 caractères sur 2 lignes, avec interface I2C (3€)

Partie tuyauterie :

- fabrication du stick sur la base de celui décrit sur <http://barbuzard.over-blog.com/article-30676976.html>
- j'utilise des détendeurs Oxygène et Hélium fournissant maximum 10 bar en sortie, pour pouvoir ajuster le débit précisément



les ingrédients :



écran LCD 2 lignes
I2C 1602



convertisseur ADC 16 bits
ADS1115



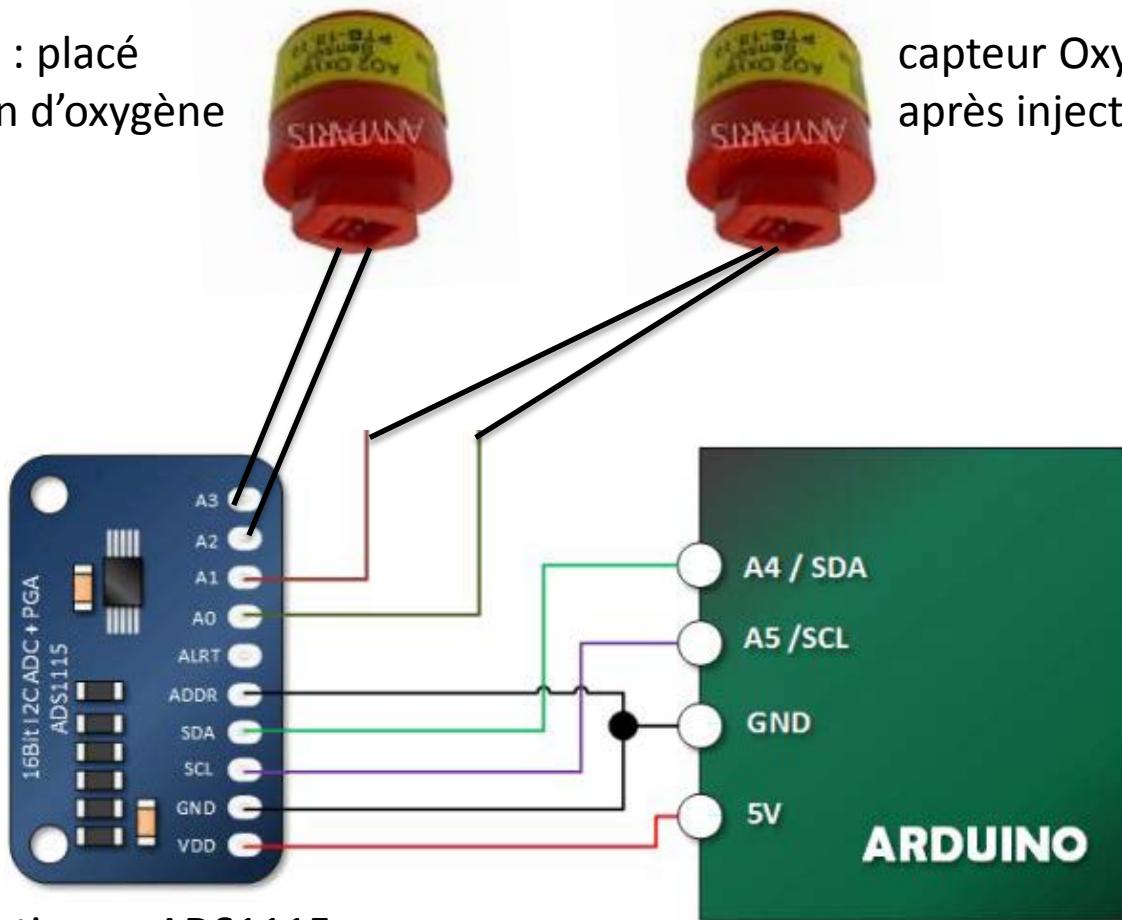
carte Arduino



2 cellules oxy AO2 Citycell

cablage cellules O2 :

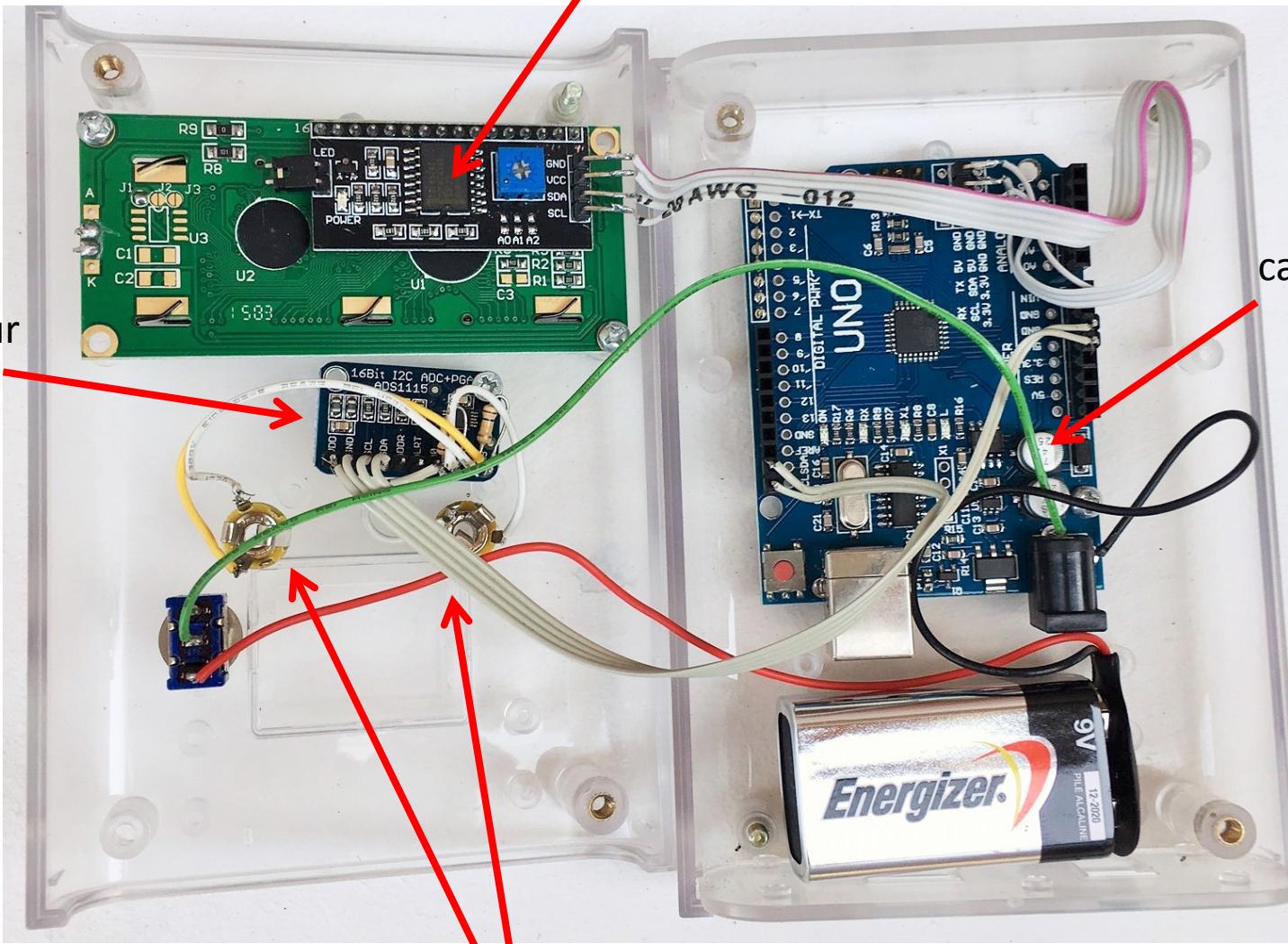
capteur Oxy 2 : placé
après injection d'oxygène



capteur Oxy 1 : placé
après injection d'hélium

convertisseur ADS1115

vue d'ensemble :



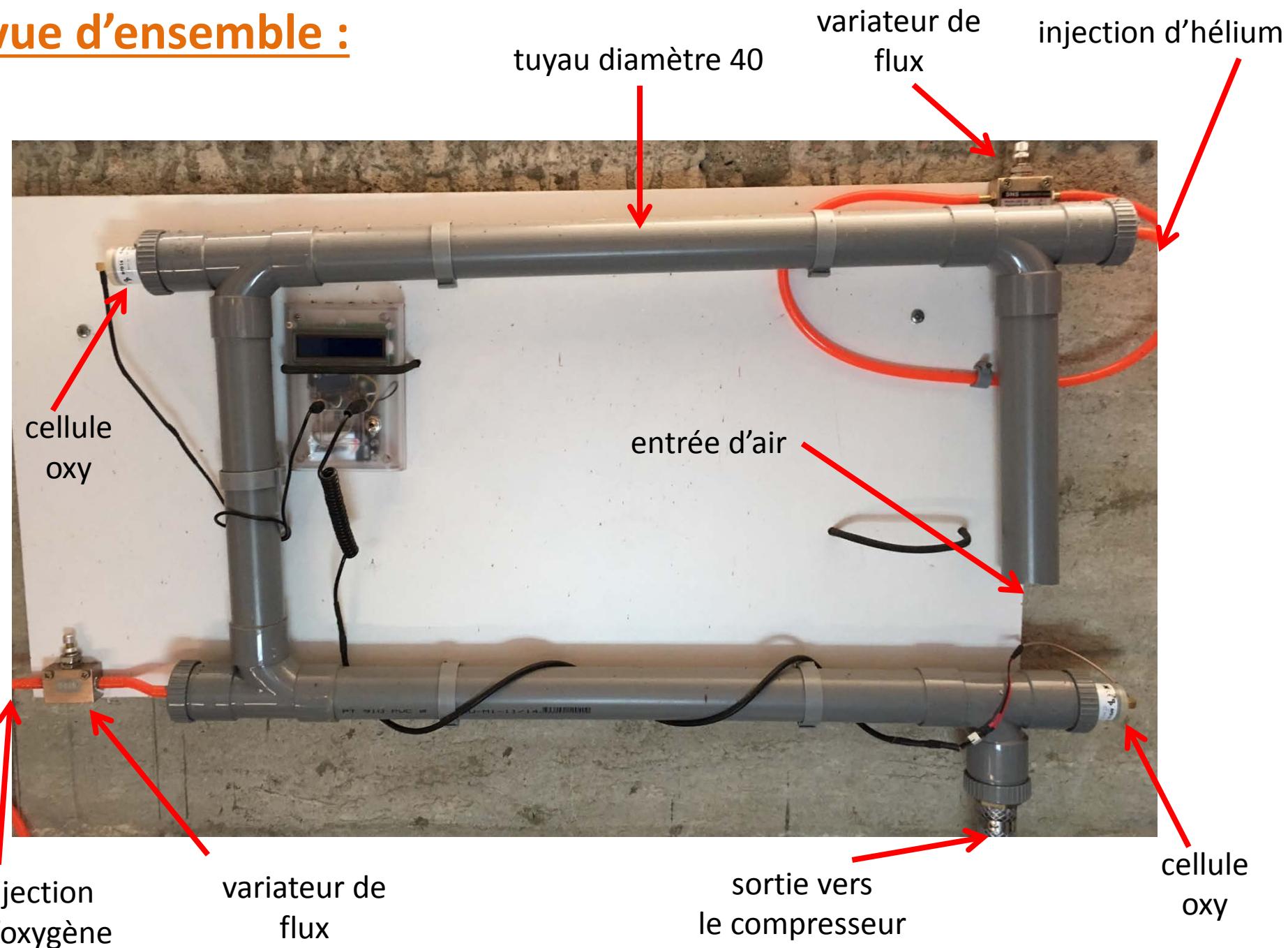
écran LCD

carte Arduino

convertisseur
ADS1115

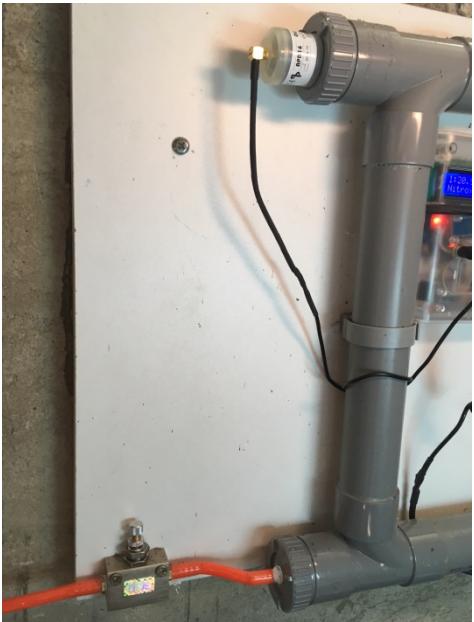
prises jack pour
les 2 capteurs Oxy

vue d'ensemble :



détails :

convertisseur
ADS1115



partie gauche



détendeur 4 bar Hélium



contrôleur de flux
sur Aliexpress

4.7 ★★★★
Evaluations 7 Commandes Liste de vœux



partie droite



Livraison gratuite 10 pcs/lot haute qualité
tuyau connecteur conique barbelé 1/8 " fileté
mâle raccord en laiton

€ 9,28 / lot (10 pieces / lot)

Prix vrac 5% off (>= 3 pieces)



branchement sur le
compresseur

Le programme dans l'Arduino (déclarations) :

```
#include <Wire.h>
#include <Adafruit_ADS1015.h>
#include <RunningAverage.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x20 for a 16 chars and 2 line display
Adafruit_ADS1115 ads;

float TensionCalib1 = 0;      // mise a 0 de la tension de calibrage
float TensionCalib2 = 0;      // mise a 0 de la tension de calibrage
float voltage0 = 0;
float voltage1 = 0;
int MOD = 0;                // valeur de MOD du mélange
float gain = 0.0078125;
boolean Cell1HS = false;     // dans le cas ou la cellule 1 est HS ou absente
boolean Cell2HS = false;     // dans le cas ou la cellule 2 est HS ou absente

RunningAverage RA0(10);
RunningAverage RA1(10);
```

Le programme dans l'Arduino (setup) :

```
void setup() {  
  
    // initialize serial communication at 9600 bits per second:  
    Serial.begin(9600);  
  
    lcd.begin();  
    lcd.backlight();  
    lcd.print(" Stick Trimix");  
    lcd.setCursor(0,1);  
    lcd.print("avec Helium en 1");  
    delay(2000);  
  
    ads.setGain(GAIN_SIXTEEN); // 16x gain 1 bit = 0.0078125mV  
    ads.begin();  
  
    int16_t adc0;  
    int16_t adc1;  
  
    adc0 = ads.readADC_Differential_0_1();  
    RA0.addValue(adc0);  
    voltage0 = abs(RA0.getAverage()*gain);  
  
    adc1 = ads.readADC_Differential_2_3();  
    RA1.addValue(adc1);  
    voltage1 = abs(RA1.getAverage()*gain);  
  
    // affichage de la tension  
    lcd.clear();  
    lcd.print("cell 1 : ");  
    lcd.print(voltage0,2);  
    lcd.print("mV");  
    lcd.setCursor(0,1);  
    lcd.print("cell 2 : ");  
    lcd.print(voltage1,2);  
    lcd.print("mV");  
    delay(2000);  
  
    lcd.clear();  
    lcd.print("Calib. en cours");  
  
    int i = 0;  
    int j = 0;  
  
    float tensionMoyenne0 = 0;  
    for(i = 1; i < 10 or (abs (voltage0 - (tensionMoyenne0 / (i-1))) > 0.002;  
i++)  
    {  
        adc0 = ads.readADC_Differential_0_1();  
        RA0.addValue(adc0);  
        voltage0 = abs(RA0.getAverage()*gain);  
        tensionMoyenne0 = tensionMoyenne0 + voltage0;  
        delay(200);  
    }  
  
    RA1.addValue(adc1);  
    voltage1 = abs(RA1.getAverage()*gain);  
  
    float tensionMoyenne1 = 0;  
  
    for(j = 1; j < 10 or (abs (voltage1 - (tensionMoyenne1 / (j-1))) >  
0.002; j++)  
    {  
        adc1 = ads.readADC_Differential_2_3();  
        RA1.addValue(adc1);  
        voltage1 = abs(RA1.getAverage()*gain);  
        tensionMoyenne1 = tensionMoyenne1 + voltage1;  
        delay(200);  
    }  
  
    lcd.clear();
```

Le programme dans l'Arduino (setup) :

```
if ((voltage0 < 2) && (voltage1 < 2)) {  
    lcd.print("2 Cellules HS !!");  
    Cell1HS = true;  
    Cell2HS = true;  
    delay(3000);  
}  
else {  
    lcd.print("Calibrage OK");  
    if (voltage0 < 2) {  
        Cell1HS = true;  
        lcd.setCursor(0,1);  
        lcd.print("Cellule 1 HS");  
        delay(3000);  
    }  
    if (voltage1 < 2) {  
        Cell2HS = true;  
        lcd.setCursor(0,1);  
        lcd.print("Cellule 2 HS");  
        delay(3000);  
    }  
}  
delay(1000);  
tensionMoyenne0 = tensionMoyenne0 / (i - 1);  
tensionMoyenne1 = tensionMoyenne1 / (j - 1);  
TensionCalib1 = tensionMoyenne0;  
TensionCalib2 = tensionMoyenne1;  
lcd.clear();  
lcd.print("V1 : ");  
lcd.print(TensionCalib1,2);  
lcd.print("mV");  
lcd.setCursor(0,1);  
lcd.print("V2 : ");  
lcd.print(TensionCalib2,2);  
lcd.print("mV");  
delay(1000);  
lcd.clear();  
lcd.print("1:");  
lcd.setCursor(8,0);  
lcd.print("2:");  
}
```

Le programme dans l'Arduino (loop) :

```
void loop() {
    int16_t adc0;
    int16_t adc1;
    adc0 = ads.readADC_Differential_0_1();
    adc1 = ads.readADC_Differential_2_3();

    RA0.addValue(adc0);
    voltage0 = abs(RA0.getAverage()*gain);

    RA1.addValue(adc1);
    voltage1 = abs(RA1.getAverage()*gain);

    // dans le cas ou seule la cellule 1 est connectée
    if (Cell2HS) {
        voltage1 = voltage0;
        TensionCalib2 = TensionCalib1;
    }
    // dans le cas ou seule la cellule 2 est connectée
    if (Cell1HS) {
        voltage0 = voltage1;
        TensionCalib1 = TensionCalib2;
    }

    float oxy1 = 0;
    float oxy2 = 0;
    float He1 = 0;
    float He2 = 0;

    oxy1 = voltage0 * (20.9 / TensionCalib1);
    oxy2 = voltage1 * (20.9 / TensionCalib2);
    MOD = 10 * ((160/oxy2) - 1);
    He1 = 100 - oxy1 - (79.1 / 20.9) * oxy1; // déduction du taux d'hélium sur
    le mélange Héliair
    He2 = 100 - oxy2 - (1 - (oxy2 - oxy1)/100) * 79.1/20.9 *oxy1; // déduction
    du taux d'hélium sur le mélange Trimix final
    if (He2 < 0) {
        He2 = 0;
    }

    // si les 2 cellules sont à 0 on met tout à 0
    if ((Cell1HS) && (Cell2HS)) {
        lcd.clear();
        lcd.print("2 cellules HS !!");
        goto CellulesHS;
    }

    lcd.setCursor(2,0);
    lcd.print(oxy1,1);
    lcd.print("% ");

    // si on n'injecte pas d'hélium en 1, on passe en mode Nitrox
    if (oxy1 < 20.5) {
        if (oxy1 == oxy2) { // s'il manque une cellule, on est en Héliair
            lcd.setCursor(8,0);
            lcd.print("MOD ");
            lcd.print(MOD);
            lcd.print("m ");
            lcd.setCursor(0,1);
            lcd.print("Heliair ");
            lcd.print(oxy2,0);
            lcd.print("/");
            lcd.print(He2,0);
        }
    }
}
```

Le programme dans l'Arduino (loop) :

```
else {      // mode Trimix
  lcd.setCursor(8,0);
  lcd.print("2:");
  lcd.print(oxy2,1);
  lcd.print("% ");
  lcd.setCursor(0,1);
  lcd.print("Tx ");
  lcd.print(oxy2,0);
  lcd.print("/");
  lcd.print(He2,0);
  lcd.print(" MOD ");
  lcd.print(MOD);
  lcd.print("m ");
}
}

else {      // mode Nitrox
  lcd.setCursor(0,1);
  lcd.print("Nitrox ");
  lcd.print(oxy2,1);
  lcd.print("%   ");
  lcd.setCursor(8,0);
  lcd.print("MOD ");
  lcd.print(MOD);
  lcd.print("m ");
}

CellulesHS:
delay(500);    // delay in between reads for stability

}
```

Les éléments à acheter :

- Carte Arduino Uno : [ici](#)
- Ecran LCD 1602 I2C : [ici](#)
- Carte ADS1115 : [ici](#)
- Cellule oxy : [ici](#)
- le boitier : [ici](#)
- le variateur de flux : [ici](#)
- détendeurs Oxygène et Hélium : sur Leboncoin (matériel de soudure)
- Tuyau basse pression pour Oxygène et Hélium : [ici](#)