Analyseur Trimix :

Principe :

- Utilisation d'une cellule oxygène pour mesurer le taux d'O2 du mélange (30€)
- Utilisation d'un capteur de conductivité thermique MD62 pour mesurer le taux d'hélium (25€)
- Utilisation d'une carte Arduino pour gérer logiciellement le calibrage des capteurs, compenser la non linéarité, afficher le Trimix ou Nitrox analysé et le MOD associé (3€)
- Utilisation d'une carte ADS1115 pour mesurer précisément des tensions entre 5mV et 200mV (ce que ne sait pas faire la carte Arduino nativement) (3€)
- Utilisation d'un générateur de tension LM2596 pour alimenter le pont de Wheatstone du capteur MD62 en 3V (2€)
- Utilisation d'un limiteur de flux DIN pour doser le flux de gaz circulant dans l'analyseur (49€)
- Pile 9V remplacée par une batterie LiPo pour plus d'autonomie
- Le gaz à analyser rentre à l'extrémité d'un bout de tube PVC diamètre 32, passe sur le capteur MD62, puis sur la cellule O2, et ressort à l'autre extrémité du tube.

les ingrédients :

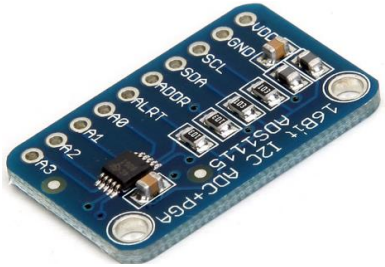
carte Arduino

écran LCD 4 lignes
I2C 2004

résistance variable
500 ohm

générateur de tension
LM2596

capteur de conductivité
thermique MD62



convertisseur ADC 16 bits
ADS1115

batterie liPo 11,1V

2 résistances 2k

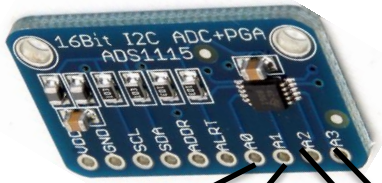
cellule oxy AO2 Citycell

cablage pont Wheatstone et mesure O2 :

on ajuste la résistance variable pour avoir une tension en entrée de l'ADS 1115 proche de 0mV à l'air libre (quand le taux d'hélium = 0)

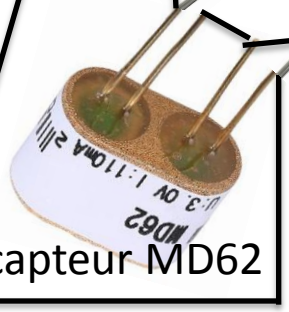
générateur de tension réglé à 3V

convertisseur ADS1115



entrées analogiques

résistance variable 500 ohm



capteur MD62

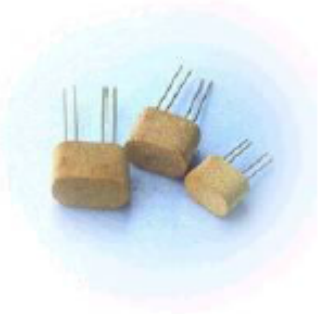
cellule Oxy

Datasheet du MD62:

MD62 gas sensor consists of an active element and a reference element with the same resistance, both elements are placed in a wheatstone bridge circuit, The analyzing gas contents changes, the overall thermal coefficient of mixed gases changed correspondingly; when the active element meet the combustible gas, its resistance become smaller, when It meet other gas, , Its resistance become larger(air background), the bridge circuit output the voltage change, this change increase according to gas concentration, the reference element as a benchmark while for temperature compensation.

Features

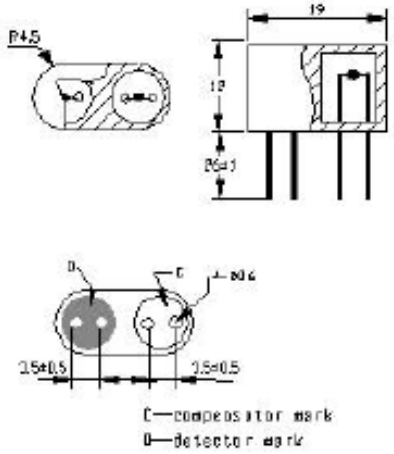
- Wide Detecting Range (0—100%VOL)
- Linear output signal
- Quick response
- Good reproducibility and reliable performance
- Resistant to toxicosis
- Detecting without Oxygen or short of oxygen



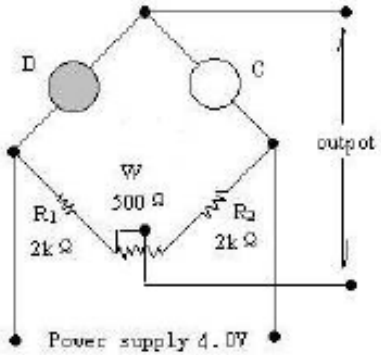
Application

Domestic, Industrial spot for CO2、CcL4、freon、Natural gas, LPG etc detecting.

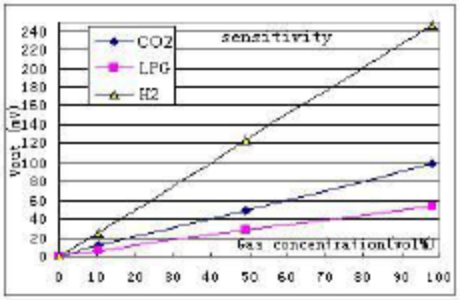
Structure

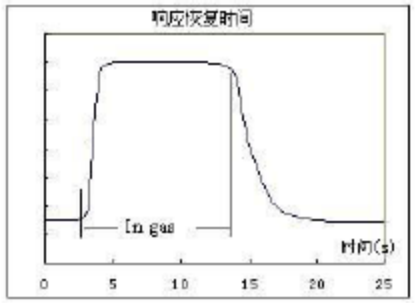


Basic testing circuit

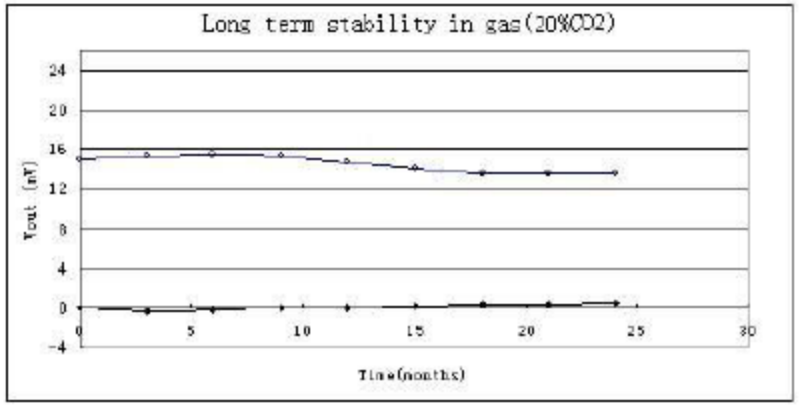


Sensitivity and response characteristic



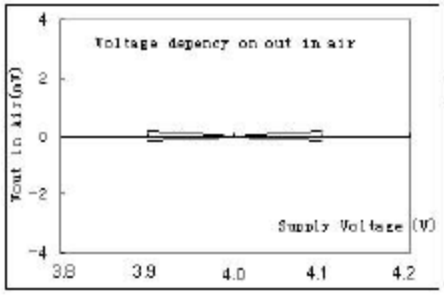


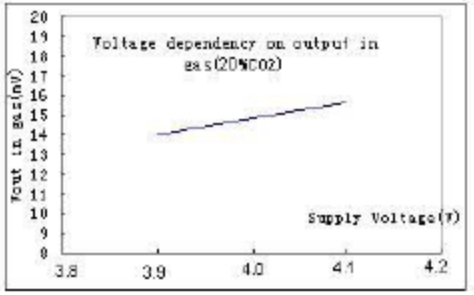
Long term stability



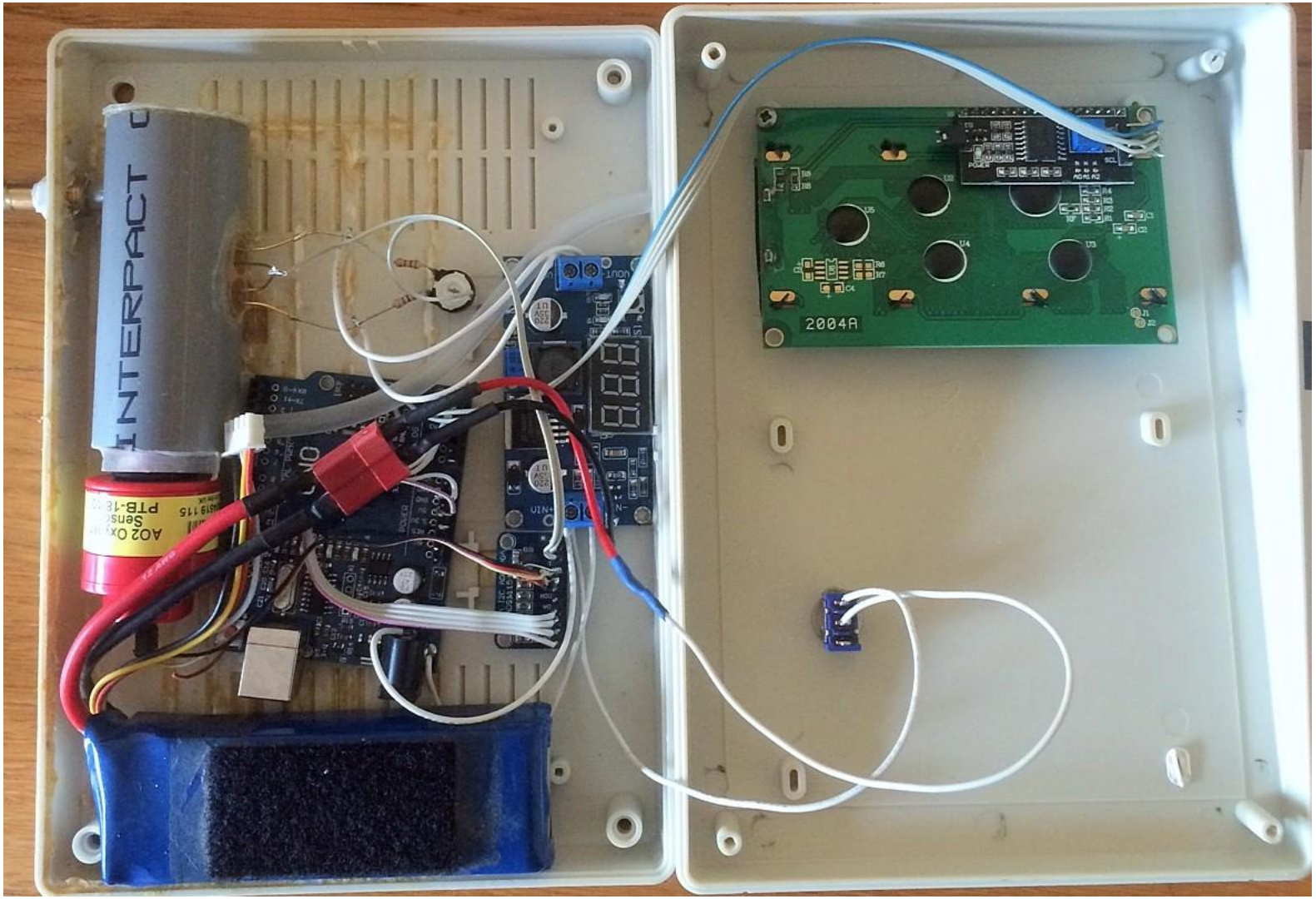
The drift in air is less than 2 mV per year, in 20%CO2 the drift is less than 2mV. for a short period storage (in 2 weeks), the sensor need 30mins' preheating to stabilize, for more than one year storage, it need more than 24 hours' preheating.

MD62 output singnal dependency on working voltage

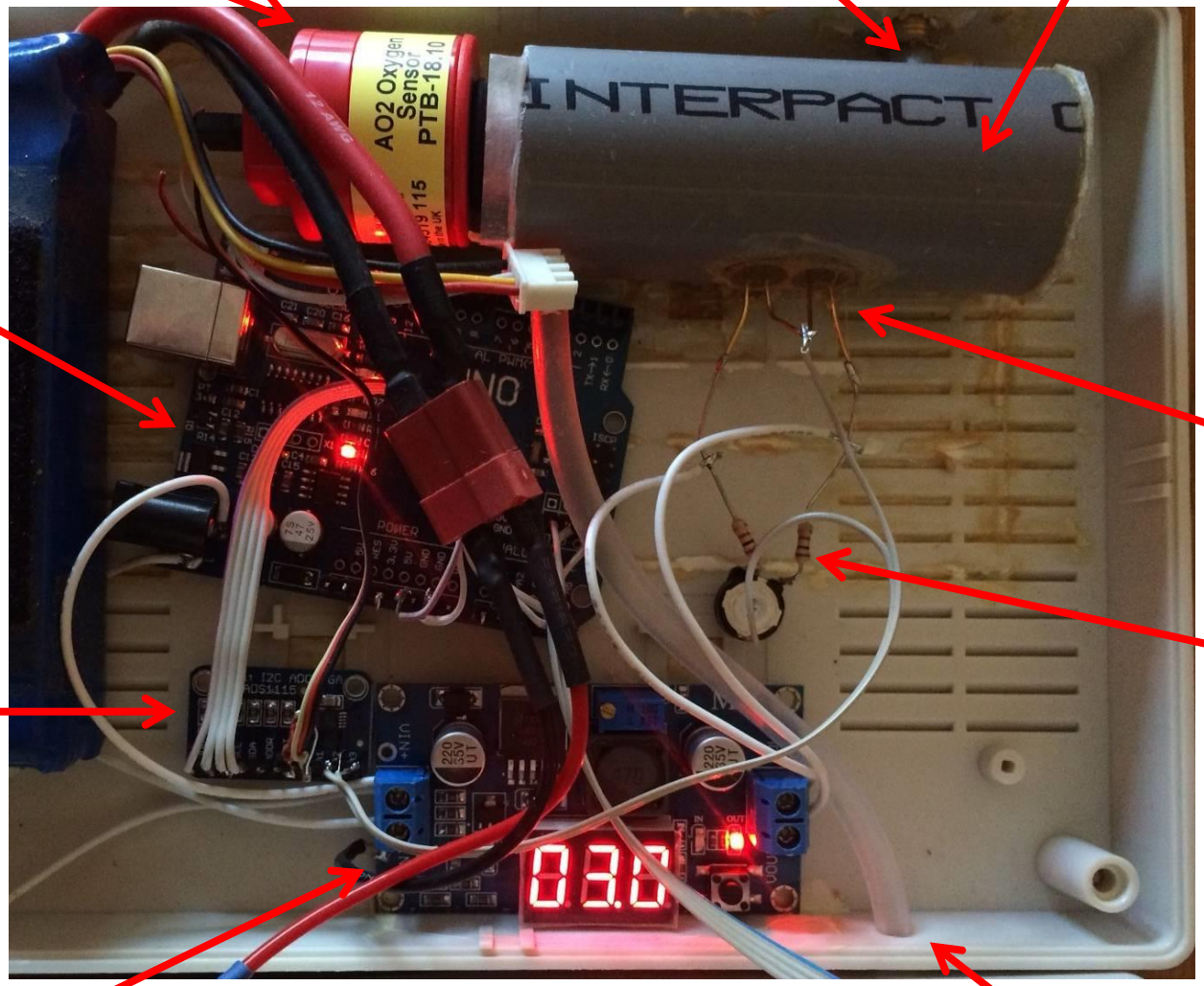




vue d’ensemble :



zoom sur le montage :



Le programme dans l'Arduino (déclarations) :

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

LiquidCrystal_I2C lcd(0x27, 20, 4);  // set the LCD address to 0x27 for a 20 chars and 4 line display
Adafruit_ADS1115 ads;    // convertisseur analogique --> digital ADS1115

// variables will change:
float TensionCalib = 0;      // mise a 0 de la tension de calibrage de la cellule
float voltage = 0;           // tension mesuree sur cellule
float wheatstone = 0;        // tension sur pont de wheatsone

float gain = 0.03125;      // gain du convertisseur ADS1115
float calibMD62 = 661.26;      // valeur de la tension du pont avec 100% helium
float WheatCalib = 10;      // mise a 0 de la tension de calibrage du pont à l'air
float CorrFroid = 0 ;       // correction de lecture du fait de lecture a froid (capteur pas assez chaud)
unsigned long time;         // mesure du temps depuis allumage pour ajuster correction

RunningAverage RA0(10);      // moyennage O2 sur 10 valeurs
RunningAverage RA1(10);      // moyennage He sur 10 valeurs
```

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("  Analyseur Trimix");

 ads.setGain(GAIN_FOUR); // 4x gain 1 bit = 0.03125mV
 ads.begin();

 int16_t adc0;
 int16_t adc1;

 adc0 = ads.readADC_Differential_0_1();
 RA0.addValue(adc0);
 voltage = abs(RA0.getAverage()*gain);

 adc1 = ads.readADC_Differential_2_3();
 RA1.addValue(adc1);
 wheatstone = RA1.getAverage()*gain;

 // affichage de la tension
 lcd.setCursor(0,2);
 lcd.print("V cell = ");
 lcd.print(voltage,2);
 lcd.print("mV");
 lcd.setCursor(0,3);
 lcd.print("V pont = ");
 lcd.print(wheatstone,2);
 lcd.print("mV");
 delay(2000);

  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print(" Calib. en cours ...");
  lcd.setCursor(0,2);
  lcd.print("(utiliser de l'air)");

 // determination de la tension moyenne de la cellule à l'air libre
 int i = 0;
 float tensionMoyenne = 0;

 for(i = 1; i <10 or (abs (voltage - (tensionMoyenne / (i-1)))) > 0.001; i++)
  {
    adc0 = ads.readADC_Differential_0_1();
    RA0.addValue(adc0);
    voltage = abs(RA0.getAverage()*gain);
    tensionMoyenne = tensionMoyenne + voltage;
    delay(200);
  }

 lcd.clear();
 lcd.setCursor(0,1);
 lcd.print("    Calibrage OK");
 tensionMoyenne = tensionMoyenne / (i - 1);
 TensionCalib = tensionMoyenne;
 lcd.setCursor(0,2);
 lcd.print("V calib = ");
 lcd.print(TensionCalib,2);
 lcd.print("mV");

 delay(2000);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Prechauffage du");
 lcd.setCursor(0,1);
 lcd.print("  capteur Helium...");
 delay(500);

 while(wheatstone > 10){
     adc1 = ads.readADC_Differential_2_3();
     RA1.addValue(adc1);
     wheatstone = RA1.getAverage()*gain;
     lcd.setCursor(4,3);
     lcd.print("Vpont=");
     lcd.print(wheatstone,0);
     lcd.print("mV  ");
     delay(50);
     }

 lcd.clear();
 lcd.setCursor(0,1);
 lcd.print(" Capteur He OK");
 delay(2000);
 lcd.clear();

}
```

Le programme dans l'Arduino (loop) :

```
void loop() {
  // put your main code here, to run repeatedly:

  int16_t adc0;
  int16_t adc1;
  adc0 = ads.readADC_Differential_0_1();
  adc1 = ads.readADC_Differential_2_3();

  time = millis();     // temps en ms depuis démarrage programme

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

  RA1.addValue(adc1);
  wheatstone = RA1.getAverage()*gain;

  float nitrox = 0;
  int MOD = 0;              // valeur de MOD du mélange
  int EAD = 0;              // prof equivalente Air pour narcose 30m
  float helium = 0;

    nitrox = voltage * (20.9 / TensionCalib);
    MOD = 10 * ( (160/nitrox) - 1);

    lcd.setCursor(0,0);
    if (voltage > 1) {
      lcd.print("Oxygene = ");
      lcd.print(nitrox,1);
      lcd.print("%   ");
      lcd.setCursor(12,2);
      lcd.print(" MOD");
      lcd.print(MOD);
      lcd.print("m");
      if (MOD < 100) {
        lcd.print(" ");
        }
      }
      else {
        lcd.print("Cell O2 HS !!! ");
        }
    lcd.setCursor(0,3);
    lcd.print("Vpont=");
    lcd.print(wheatstone,0);
    lcd.print("mV  ");

    wheatstone = wheatstone - WheatCalib;

    if (time < 480000) {  CorrFroid = 1 ; }     // correction de la tension lue en fonction du temps de
chauffe du capteur
    if (time < 360000) {  CorrFroid = 2 ; }
    if (time < 300000) {  CorrFroid = 3 ; }
    if (time < 270000) {  CorrFroid = 4 ; }
    if (time < 240000) {  CorrFroid = 5 ; }
    if (time < 210000) {  CorrFroid = 6 ; }
    if (time < 180000) {  CorrFroid = 7 ; }
    if (time < 165000) {  CorrFroid = 8 ; }
    if (time < 150000) {  CorrFroid = 9 ; }
    if (time < 120000) {  CorrFroid = 10 ; }
    if (time < 105000) {  CorrFroid = 11 ; }
    if (time < 90000) {  CorrFroid = 12 ; }
    if (time < 80000) {  CorrFroid = 13 ; }
    if (time < 70000) {  CorrFroid = 14 ; }
    if (time < 60000) {  CorrFroid = 15 ; }
    if (time < 50000) {  CorrFroid = 16 ; }
    if (time < 40000) {  CorrFroid = 17 ; }
    if (time < 30000) {  CorrFroid = 18 ; }

    wheatstone = wheatstone - CorrFroid;          // ajustement car capteur pas assez chaud


    lcd.setCursor(0,1);
    lcd.print("Helium = ");
    helium = 100 * wheatstone / calibMD62;
    if (helium > 50) {
      helium = helium * (1 + (helium - 50) * 0.4 / 100);
      }
    if (helium > 2) {
      lcd.print(helium,1);
      lcd.print("%    ");
      }
    else {
      helium = 0;
      lcd.print("0%    ");
      }
```

Le programme dans l'Arduino (loop) :

```
    lcd.setCursor(0,2);
    if (helium > 0) {
     lcd.print("Trimix ");
     lcd.print(nitrox,0);
     lcd.print("/");
     lcd.print(helium,0);
     lcd.print(" ");
     lcd.setCursor(12,3);
     lcd.print(" EAD");
     EAD = 10 * ( 100 * 3.2 / (100 - helium - nitrox) -1 );
     lcd.print(EAD);
     lcd.print("m");
     if (EAD < 100) {
       lcd.print(" ");
       }
     }
    else {
     lcd.print("Nitrox ");
     lcd.print(nitrox,0);
     lcd.print("    ");
     lcd.setCursor(12,3);
     lcd.print("       ");
     }
}
```

Les éléments à acheter :

- Carte Arduino Uno : ici
- Ecran LCD 2004 I2C : ici
- Carte ADS1115 : ici
- Générateur de tension LM2596 : ici
- Capteur MD62 : ici
- Cellule oxy : ici
- Batterie LiPo 11,1V : ici

- le boitier : ici
- Limiteur de flux : ici