



HAL
open science

Validation et optimisation d'une sonde de mesure de CO₂ dissous à bas coût pour les milieux aquatiques

Viet Tran Khac, Maxime Thibaudon, Philippe Quéting, Didier Jézéquel

► To cite this version:

Viet Tran Khac, Maxime Thibaudon, Philippe Quéting, Didier Jézéquel. Validation et optimisation d'une sonde de mesure de CO₂ dissous à bas coût pour les milieux aquatiques. INRAE; Pole R&D ECLA; USMB; IPGP, Université Paris Cité. 2025, pp.25. hal-04870739

HAL Id: hal-04870739

<https://hal.science/hal-04870739v1>

Submitted on 7 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright



INRAE



Validation et optimisation d'une sonde de mesure de CO₂ dissous à bas coût pour les milieux aquatiques

Synthèse des résultats de la première année 2023

Axe 1 Caractérisation et évaluation de l'état et des trajectoires des milieux lacustres, de leur biodiversité, et de leur fonctionnement

Viet Tran Khac, Maxime Thibaudon, Philippe Quetin, Didier Jézéquel

Pôle R&D ECLA

Site INRAE d'Aix-en-Provence
3275 route Cézanne – 13100 Le Tholonet

<https://professionnels.ofb.fr/fr/pole-ecla-ecosystemes-lacustres>

SOMMAIRE

I. Contexte	4
II. Matériel et méthodes	5
II.1 Sélection des matériels.....	5
II.1.1 Capteur CO2 SenseAir K33.....	5
II.1.2 Capteur CH4 Figaro NGM2611-E13.....	6
II.2 Principe du fonctionnement	6
II.3 Schéma du branchement.....	6
III. Résultats	9
III.1 Étanchéité du système	9
III.1.1 Boîtier d'étalonnage des capteurs	9
III.1.2 Étanchéité des tuyaux	9
III.2 Étalonnage du capteur K33	10
III.3 Réponse des capteurs K33	12
III.3.1 Test de dégazage	12
III.3.2 Réponse du capteur à des conditions saturée en RH	13
IV. Conclusion provisoire	16
V. Bibliographie	17
VI. Annexe	18
VI.1 Code d'Arduino.....	18



I Contexte

Pôle R&D ECLA

I. Contexte

Les gaz à effet de serre (GES), notamment le dioxyde de carbone (CO₂) et le méthane (CH₄), contribuent fortement au réchauffement climatique, et également pour le premier à l'acidification des eaux de surface. Si les masses d'eau continentales ne représentent qu'environ 3 % de la surface des continents, leurs contributions vis-à-vis des émissions de GES ne sont cependant pas négligeables. Par exemple, les émissions de CO₂ vers l'atmosphère depuis les eaux douces sont du même ordre de grandeur que le puits de CO₂ océanique (Tranvik et al., 2009).

Le sixième rapport d'évaluation du GIEC paru en 2022 a mis un focus sur le continuum de l'eau (rivières, lacs, zones humides, estuaires, zones côtières). Il est constaté que les flux de CO₂ ne sont pas explicitement pris en compte dans le bilan global source/puits de carbone, et que par ailleurs les flux de méthane sont peu étudiés et considérés comme la plus grande source d'incertitude pour le bilan global du carbone (IPCC, 2022).

A l'échelle des systèmes lacustres, ces milieux, qui sont perçus comme un réacteur biogéochimique, ont différentes fonctions dans la réception, la transformation, le stockage et le transfert du carbone. Il est donc important de comprendre leur fonctionnement biogéochimique et le bilan de carbone associé.

Les méthodes de détermination des flux de GES à l'interface eau-atmosphère sont résumées par la Figure 1. Nous pouvons estimer les émissions de GES d'un milieu aquatique soit par le biais des mesures des concentrations des gaz dissous (ou de pression partielle) dans l'eau de surface (méthode du gradient à l'interface eau-air, IEA), soit par la détermination des flux de gaz à l'interface entre l'eau et l'atmosphère (méthode directe par accumulation ou méthode de la covariance des turbulences). Les méthodes par la chambre flottantes et de covariances des turbulences nécessitent des logistiques et des financements plus importants par rapport à la méthode du gradient à l'IEA, qui est basée sur des capteurs de gaz dissous.

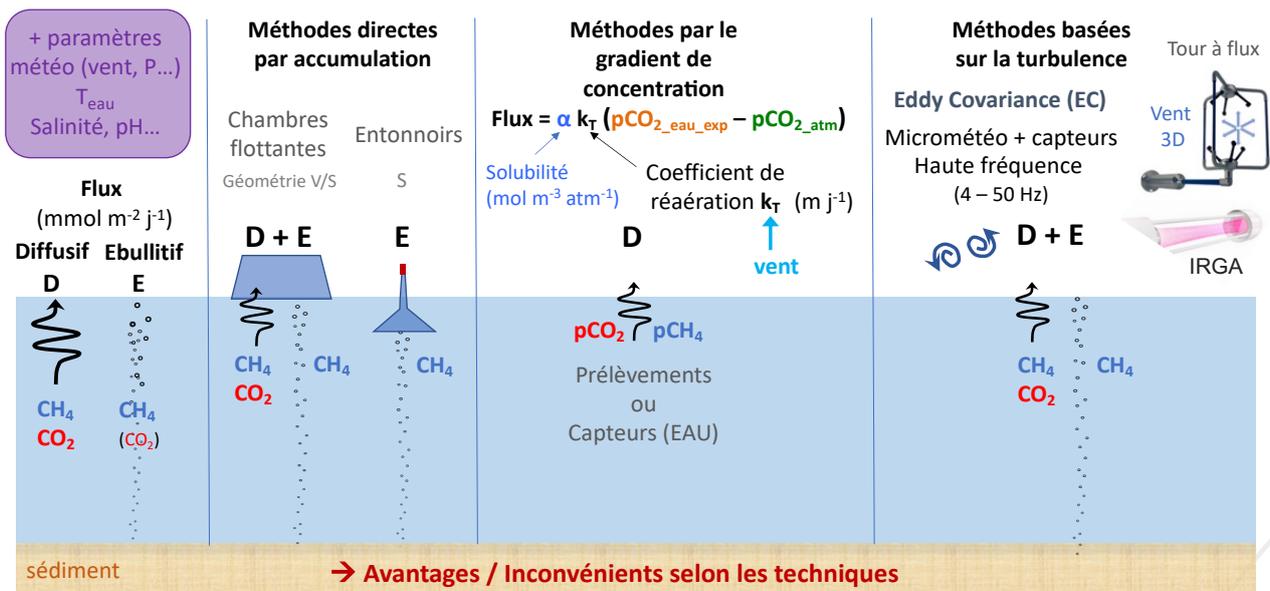


Figure 1 Panorama des méthodes de détermination des flux de GES à l'interface eau et atmosphère

A l'heure actuelle, plusieurs capteurs commerciaux permettent de mesurer les concentrations en CO₂ et CH₄. Néanmoins le prix est assez élevé : entre 3 k€ à 20 k€ pour le CO₂ (AMT Analysenmesstechnik GmbH, CONTROS HydroC) et aux environs de 50 k € pour le CH₄ (CONTROS HydroCH₄ HISEM).

A l'heure actuelle, les capteurs low-cost en CO₂ et CH₄ ne permettent que de mesurer dans l'air. Pour mesurer en milieu aquatique, le principe se base sur une mesure de concentration gazeuse dans une volume étanche en équilibre en permanence avec l'eau. En l'occurrence, nous utilisons une membrane semi-perméable en téflon poreux étanche à l'eau mais qui permet la diffusion des gaz intéressés (Johnson et al., 2012).

Ce projet vise à mesurer en continu des concentrations en CO₂ et CH₄ dissous en utilisant des capteurs à moindre coût et à augmenter la capacité de déploiement multisites grâce à la réduction des coûts.

II. Matériel et méthodes

II.1 Sélection des matériels

Les matériels à faibles coûts utilisés pour le capteur CarbonAqua sont présentés dans le Tableau 1, que nous pouvons diviser en quatre catégories :

- Datalogger : Il s'agit de la partie du système électronique embarqué qui pilote le système de circulation de l'air, effectue les mesures des GES à l'aide des deux capteurs et enregistre des données acquises.
- Capteurs : les capteurs CO₂ et CH₄ sont décrits dans les sections suivantes (II.1.1 et II.1.2)
- Alimentation : Il s'agit du système d'alimentation électrique, composé d'un panneau solaire avec régulation de charge et d'une batterie.
- Circulation de l'air : Cette partie comprend la pompe volumétrique et les tuyaux étanches aux gaz.

Tableau 1 Les matériels sélectionnés pour le montage du capteur CarbonAqua

No	Référence	Description	Prix (€)	Année
Datalogger				
1	715-4081	UNO Rév. 3 Arduino	19.79	2022
2	Amazon	Data Logger Shield compatible avec Arduino	29.99	2022
3	426-DFR0060	DFRobot Accessories Screw Shield for Arduino	6.35	2022
4	758-2578	Carte SD SDHC 16 Go	12.54	2022
Panneau solaire + Batterie				
5	SPP040201200	Victron 20W-12V Poly 440 x 350 x 25mm séries 4a	31	2022
6	BAT412015080	12V/15Ah AGM Super Cycle Batt. (Faston-tab 6.3x0.8mm)	49.7	2022
7	Régulateur batterie	Victron PWM 5A LCD/USB 12/24V	35.7	2022
Capteurs				
8	033-8-0009	Senseair K33 LP T/RH Low power sensor core for CO ₂ , temperature and RH measurement	140	2022
9	134-6661	Capteur qualité d'air Figaro, NGM2611-E13, Méthane	22	2022
Tuyaux				
10	Viton	Tuyaux Viton® Rotilabo®, FPM, noir 1 Ø int. 6 mm, Ø ext. 8 mm	71	2023
11	11703139	Saint-Gobain Tubulure Bev-A-Line IV	136	2023
12	REI-14887	14887 THOMAPOR®-Sintered Pipe made of Porous - PTFE	577	2021
Autres				
13	702-6898	Pompe volumétrique RS PRO, 380ml/min., 1,5 → 5 V Diaphragme Electrique, 11 psi	69	2023
14	Amazon	Senbeeda 6Pcs MP1584EN Convertisseur Abaisseur, Step Down Réglable Ultra Petit DC-DC 4.5-28V à 0.8-20V	10	2023
Total			1210.07	

II.1.1 Capteur CO₂ SenseAir K33

Le capteur CO₂ K33 de SenseAir est un capteur infrarouge non dispersif (NDIR) à faible coût, conçu pour mesurer la concentration en dioxyde de carbone dans l'air. La technologie NDIR repose sur le principe de l'absorption sélective des longueurs d'onde infrarouges par le CO₂, notamment à 4260nm (absorption maximale) et à 3900nm (absorption nulle). L'air ambiant diffuse à travers une membrane perméable ou surface sensitive au gas (Figure 2) et est exposé à une source de lumière infrarouge. Une partie de cette lumière est absorbée par le CO₂ présent. La quantité de lumière détectée est donc inversement proportionnelle à la concentration de CO₂ dans le milieu.

Le capteur K33 de SenseAir affiche une performance remarquable en termes de précision et d'exactitude par rapport au capteur GMP343 de Vaisala (Finlande), avec un écart quadratique moyen des résidus (RMSE) de seulement 0,032 % pour le CO₂ (Herger et al., 2020).



Figure 2 Capteur CO₂ SenseAir K33

II.1.2 Capteur CH4 Figaro NGM2611-E13

Le capteur Figaro CH4 est un module pré-calibré d'usine basé sur le Figaro TGS 2611-E00 (Figure 3). La calibration d'usine est réalisée à 5000ppm, 20°C et 65%RH. La gamme de mesure de ce capteur est donnée entre 500-10000ppm mais il a été démontré qu'il peut mesurer à la gamme de 2 à 9 ppm (van den Bossche et al., 2017).

Le capteur utilise un élément de détection à semi-conducteur à oxyde métallique (MOS) à l'état solide. En présence du méthane, celui-ci subit une réaction chimique avec le matériau MOS, provoquant une modification de la résistance électrique. Le capteur mesure le changement des signaux de résistance et convertit en concentration.

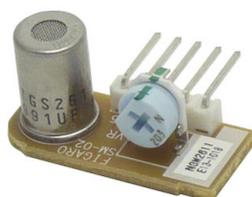


Figure 3 Capteur CH4 Figaro NGM2611-E13

II.2 Principe du fonctionnement

Le capteur CarbonAqua est basé sur la méthode par le gradient de concentration. Le boîtier monté sur une plateforme flottante comprend le système électronique embarqué et les capteurs GES. Un système de pompe assure une circulation en continu de l'air. L'équilibrateur est constitué d'un segment de tube Teflon poreux hydrophobe, agissant comme une membrane semi-perméable : perméables aux gaz et imperméable à l'eau. L'air à l'intérieur du tube téflon, en équilibre avec les pressions partielles du milieu, est ensuite pompé vers les capteurs pour être mesuré. La principale difficulté de ce système réside dans la garantie d'une étanchéité parfaite de tous ses éléments, incluant le boîtier et les tuyaux.

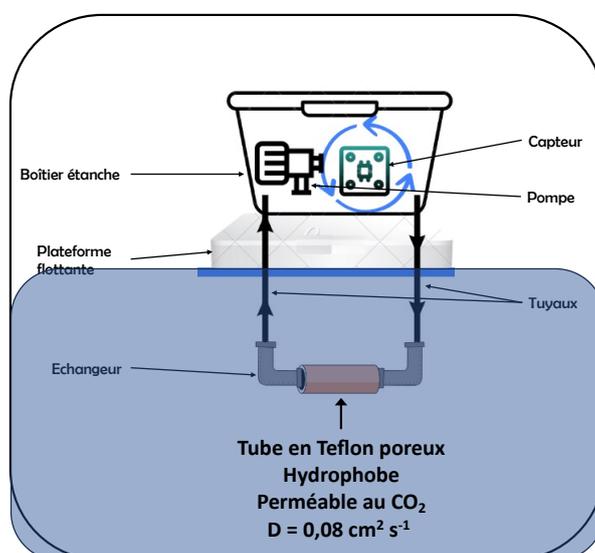


Figure 4 Principe du fonctionnement du capteur CarbonAqua

II.3 Schéma du branchement

L'ensemble des composants est monté selon les schémas de montage présentés ci-dessous (Figure 5 et Figure 6)

Carbon Aqua

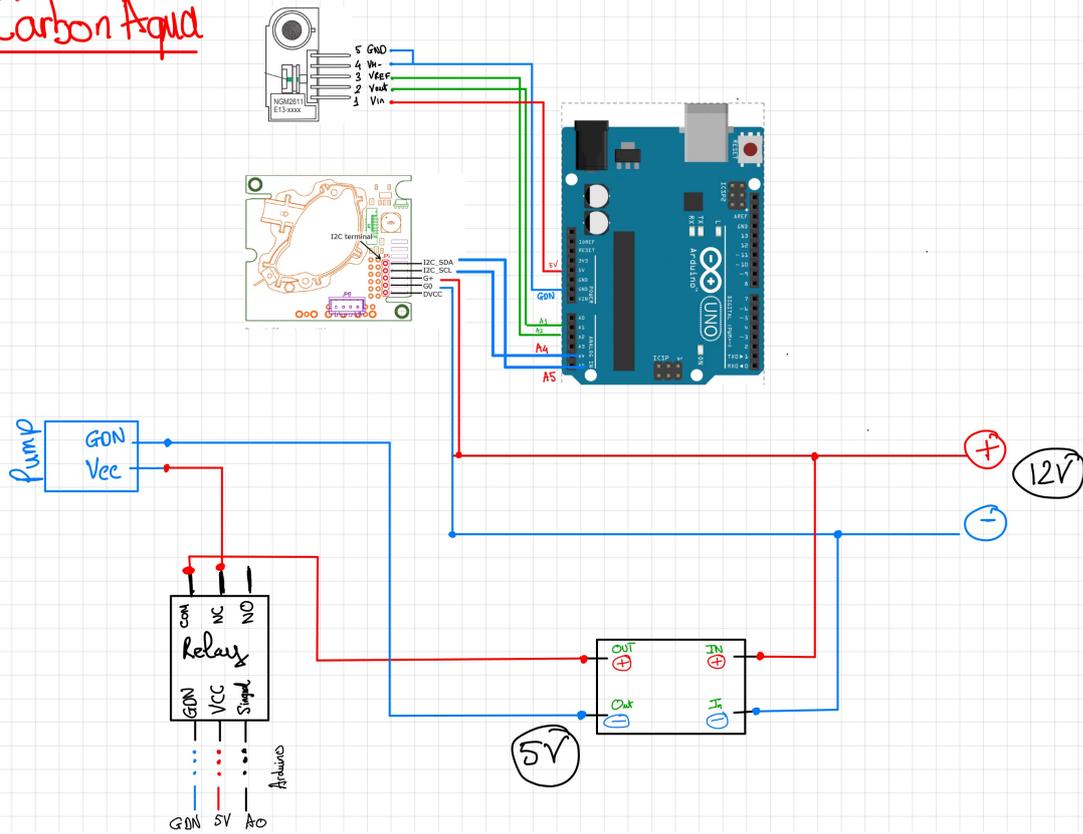


Figure 5 Schéma de branchement fonctionnel du capteur CarbonAqua

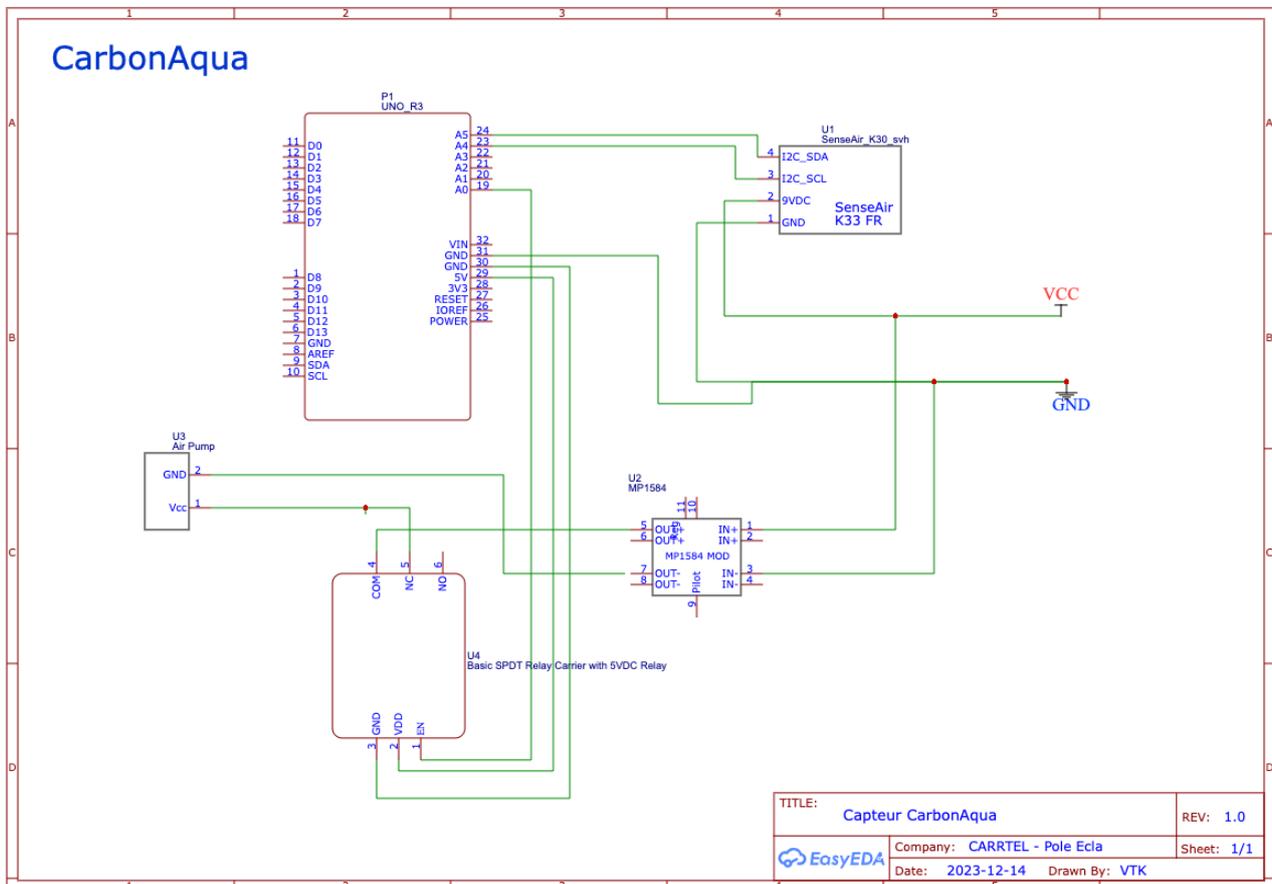


Figure 6 Schéma de circuit du capteur CarbonAqua

III Résultats



III. Résultats

III.1 Étanchéité du système

Une des difficultés principales de ce système est d'avoir une étanchéité satisfaisante à gaz. Nous avons quantifié l'étanchéité de toutes les sous parties du système.

III.1.1 Boîtier d'étalonnage des capteurs

Pour étalonner les capteurs, le boîtier doit avoir le plus petit volume mort, autrement dit le plus petit possible. Le boîtier est fabriqué en liant deux cylindres en PVC à l'aide de la colle époxy Araldite. Quant aux couvercles, on assure leur étanchéité avec du téflon en ruban et du caoutchouc thermo-rétractable. Enfin, pour permettre le passage des gaz on intègre deux embouts cannelés et un presse-étoupe pour le câble électrique afin de piloter le capteur (Figure 7).

Le test de fuite consiste à injecter un volume de CO₂ et observé la diminution de la concentration en CO₂ dans le temps. Le boîtier ne présente pas de fuite au bout d'une heure de test. Il est donc suffisant pour l'étalonnage des capteurs (Figure 8).



Figure 7 Boîtier de calibration des capteurs

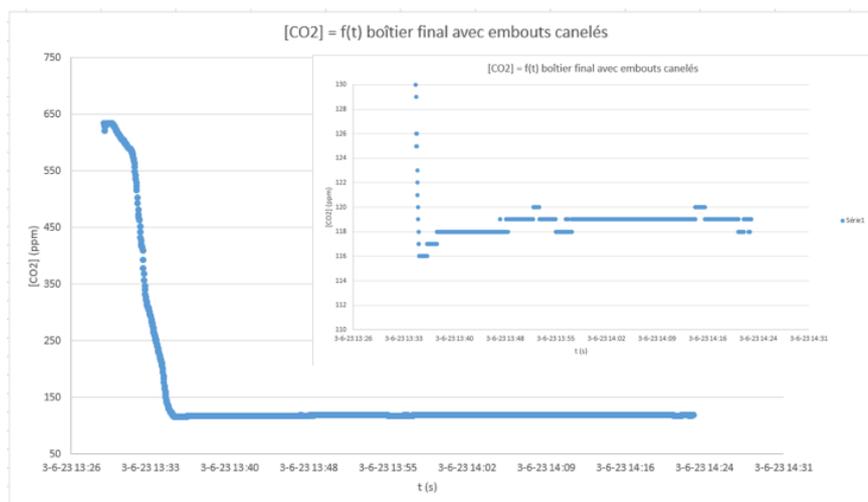


Figure 8 Test de fuite du boîtier d'étalonnage

III.1.2 Étanchéité des tuyaux

Les tuyaux constituent d'une part importante du système. Leur étanchéité est donc cruciale est quantifier car tous les matériaux ne sont pas étanches aux gaz. Le silicone est facile à utiliser grâce à sa souplesse et son élasticité. Cependant sa perméabilité aux gaz est bien connue. Au contraire, certains tuyaux sont très imperméables tel que le BEV-A-LINE. Afin d'avoir les meilleurs résultats possibles, nous réalisons plusieurs tests avec les différents tuyaux à disposition pour déterminer quel est le plus étanche. Le test de fuite est réalisé sur 4 type de tuyaux: BEV-A-LINE, Viton, Tygon et Butyl.

Afin de mener ces tests, on réalise le montage suivant Figure 9

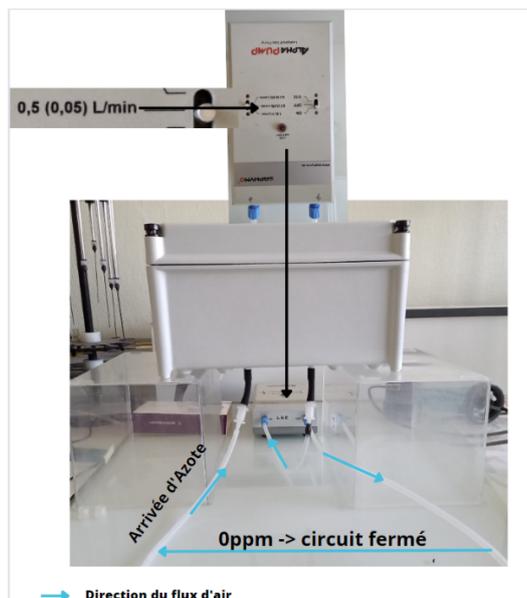


Figure 9 Montage pour le test d'étanchéité des tuyaux

Protocole du test d'étanchéité des tuyaux :

- Ajouter de l'azote dans le circuit fermé pendant 5 min avec un flux de 2.5 L.min⁻¹.
- Fermer rapidement le circuit avec le tuyau à tester.
- Régler la pompe à un débit de 0.5 L.min⁻¹.
- Laisser fonctionner le système pendant 2h.

Les résultats confirment que le Bev-a-line et le viton montrent une étanchéité comparable et plus faibles parmi les 2 autres types de tuyaux et le silicone pour la comparaison.

Tableau 2 Classement des tuyaux en fonction du débit de fuite

Tuyaux	Débit de fuite (ppm.min ⁻¹)						Qmoyen
	Q	R ²	Q	R ²	Q	R ²	
Bev-a-line	0.093	0.960	0.099	0.966	0.083	0.975	0.092
Viton	0.086	0.976	0.086	0.979	0.104	0.952	0.092
Butyl	0.133	0.957	0.096	0.953	0.079	0.963	0.103
Tygon	0.108	0.960	0.105	0.959	0.112	0.980	0.108
Silicon	0.365						0.365

III.2 Étalonnage du capteur K33

L'étalonnage du K33 consiste à effectuer trois séries de mesures croissantes et décroissantes en comparaison avec le Licor. Pour avoir des points de mesure fixes, on a à notre disposition des étalons de CO₂ à 500 ppm et 992 ppm, l'air extérieur et une bouteille d'azote pour le zéro (Figure 10).



Figure 10 Montage d'étalonnage du K33

Les résultats d'étalonnage sont présentés dans la Figure 11 et Tableau 3 . Le capteur K33 a montré des performances satisfaisantes en termes d'hystérésis, de répétabilité et de précision. L'incertitude de K33 est estimée à 4.52% de l'étendu de mesure.

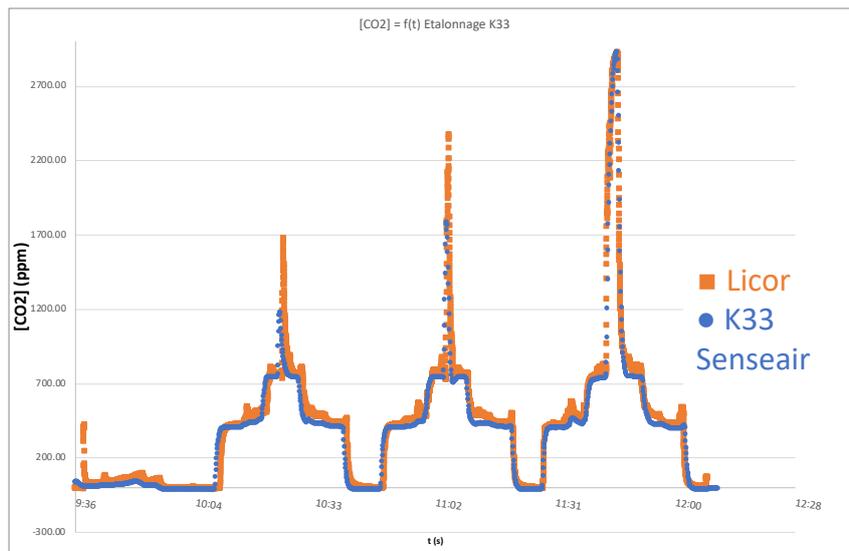


Figure 11 Étalonnage du K33 avec le Licor

Hystérésis			Répétabilité		Précision
ABS (Croissant - Décroissant) (ppm)			Max - Min (ppm)		Abs (Moyenne - Etalon) (ppm)
2.18	0.11	1.20	2.07	2.87	75.70
6.05	5.69	4.49	3.40	9.52	86.24
6.26	7.29	5.95	15.21	8.01	114.53
4.25	1.81	1.52	12.00	10.13	84.87
0.95	% d'EM		1.99 % d'EM		3.95 % d'EM
Incertitude (K33) =			4.52 % d'EM		

Tableau 3 Récapitulatif des étalonnages de K33

III.3 Réponse des capteurs K33

III.3.1 Test de dégazage

Pour expérimenter la réponse du capteur, on utilise un système un cylindre en plexiglas d'un mètre de hauteur dans lequel on peut submerger l'échangeur en téflon poreux (Figure 12). L'intérêt est de pouvoir tester le temps de réponse du capteur en pouvant contrôler les paramètres physico-chimiques de l'eau. Grâce à ces essais on peut ainsi déterminer le cycle de fonctionnement de la pompe dans le boîtier. Ce test permet également de contrôler la consommation électrique de tout le système.

Le cylindre est recouvert par un bouchon assurant l'étanchéité aux liquides et aux gaz. Ce bouchon possède deux orifices pour pouvoir ajouter des solutions faisant varier le pH et compenser ces apports, un passage pour une sonde pH et une entrée importante pour pouvoir faire passer les tuyaux et fils électriques. L'homogénéité du cylindre est quant à elle assurée par une pompe d'aquarium et un agitateur magnétique assez résistant pour supporter le poids de la manipulation.

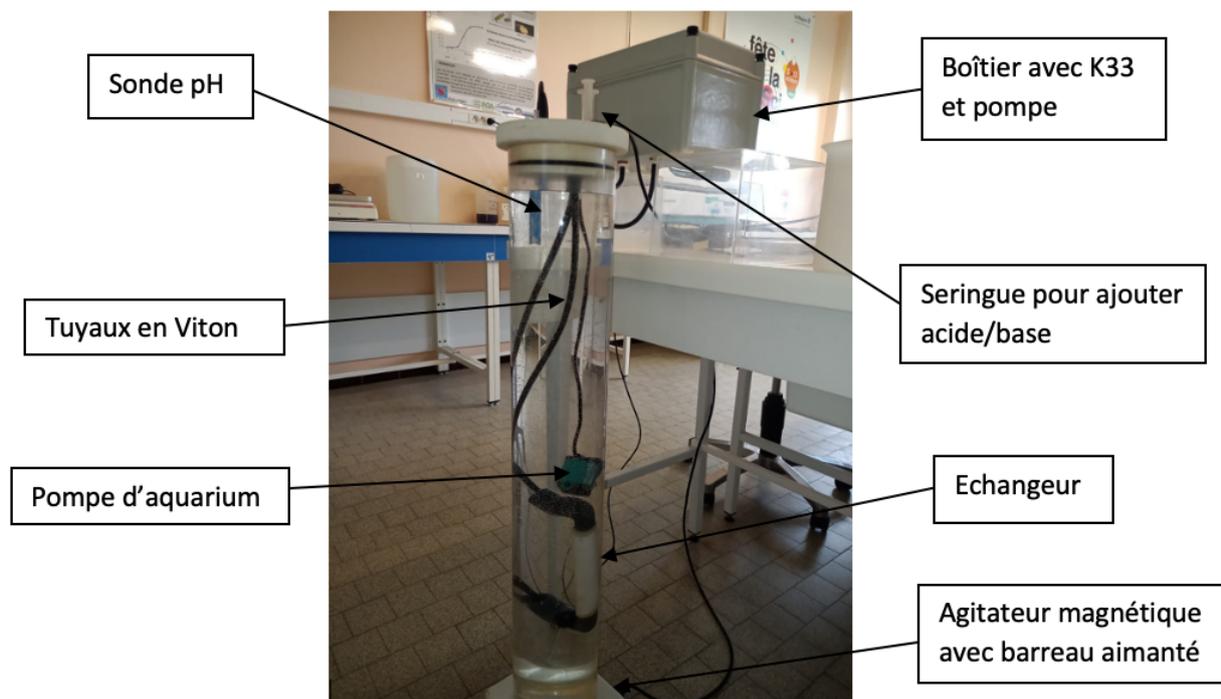


Figure 12 Système pour le test de dégazage

En faisant varier le pH, on décale l'équilibre du système carbonate. Acidifier l'eau (avec du HCl) provoque un dégazage de CO₂. Car avec l'ajout d'ions H⁺, les ions bicarbonate forment l'acide carbonique, qui elle-même se dissocie pour donner de l'eau et du dioxyde de carbone. Au contraire, en ajoutant une base telle que de la soude (NaOH) on baisse la pression partielle de CO₂ en précipitant le carbonate de calcium. Le calcium étant présent naturellement dans l'eau (sous forme d'ions Ca²⁺), avec une concentration variant en fonction des régions.

Le temps de réponse du capteur est long avec un débit d'air de 1L/min (Figure 13). Ce temps de réponse nécessite un réajustement en fonction du temps d'équilibre statique (arrêt de la pompe) et le débit de la pompe. L'hypothèse est que le débit d'air en continu pourrait empêcher la diffusion du CO₂ à l'intérieur de l'échangeur de gaz.

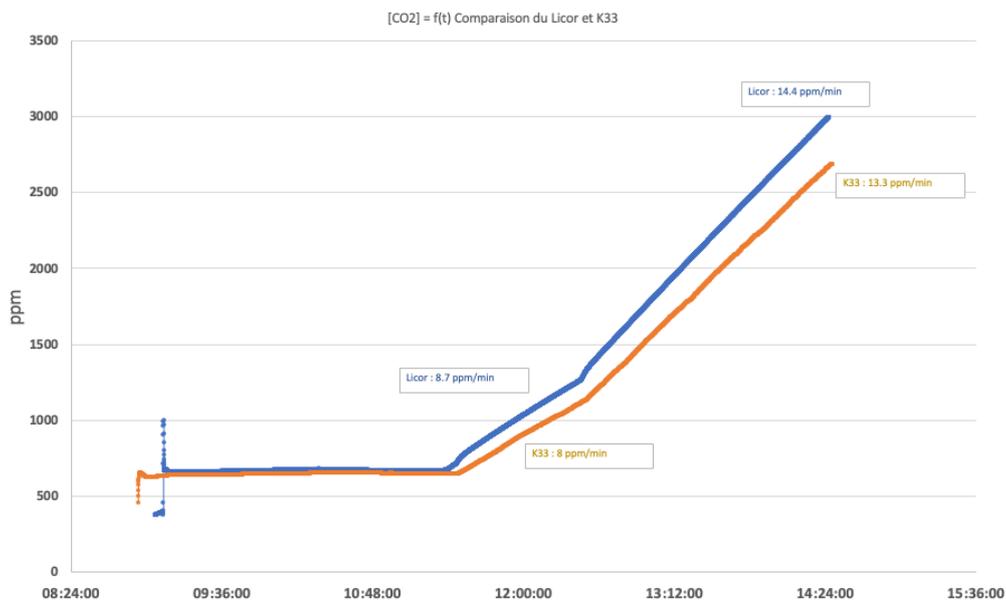


Figure 13 Temps de réponse du capteur

III.3.2 Réponse du capteur à des conditions saturée en RH

Le capteur K33 est utilisé à nu et testé mensuellement dans une chambre flottante automatique pendant 1 année au lac de Chavoley. Le capteurs K33 a donné de bons résultats par rapport au capteur Licor (résultats ne sont pas présentés ici). Néanmoins, le capteur à nu montre des instabilités lorsque l'humidité relative atteint 100 % (Figure 14).

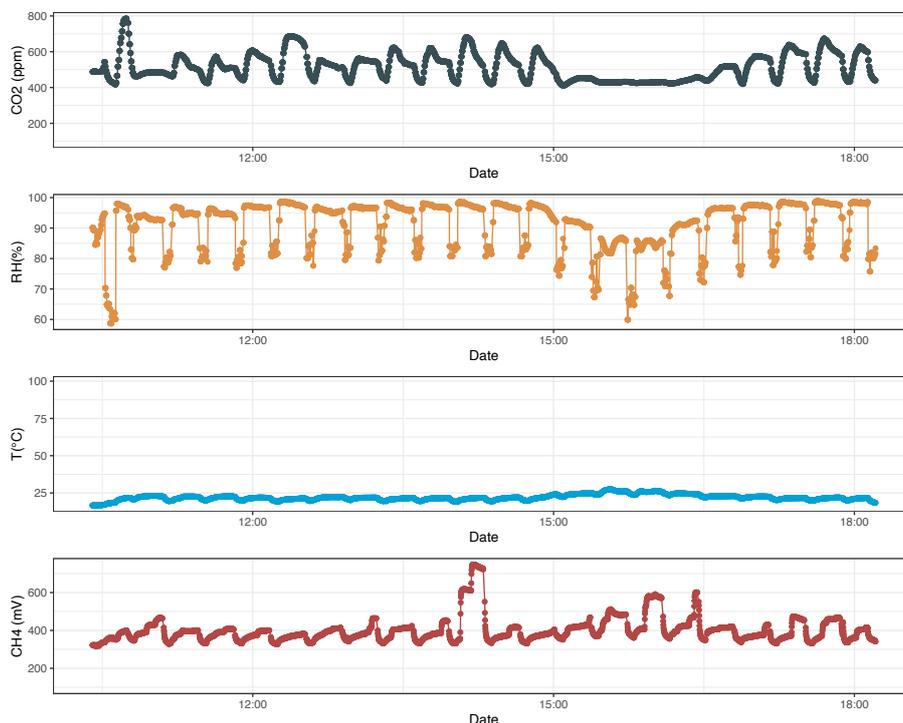


Figure 14 Réponse du capteur K33 nu à des conditions saturées en RH

Dans la configuration du capteur CarbonAqua, la membrane du Teflon joue un rôle de barrière pour l'humidité. Elle permet d'éviter la sursaturation de la vapeur d'eau. Selon nos mesures, l'humidité relative reste toujours dans la zone optimale pour le capteur K33 durant 29 heures de mesures au laboratoire.

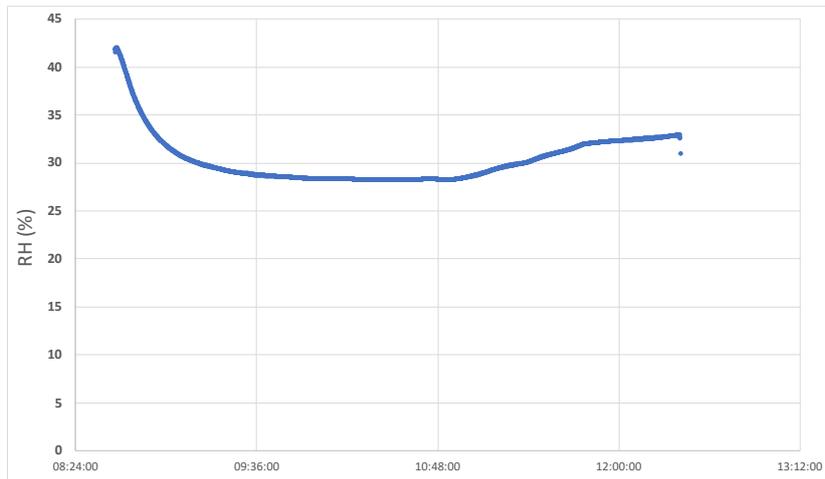


Figure 15 Evolution de la RH selon le temps

IV Conclusion

Pôle R&D ECLA

IV. Conclusion provisoire

Les résultats obtenus ont montré que le capteur CarbonAqua assure une étanchéité au gaz de façon satisfaisante. Nous avons pu valider le système d'étalonnage pour le capteur, ainsi que le type des tuyaux étanches (Bev-a-line et Viton).

Le capteur K33 a démontré des performances satisfaisantes et une robustesse dans le temps (mensuellement pendant 1an). Bien qu'il présente des instabilités aux conditions de saturation en humidité, la membrane en teflon permet de maintenir des conditions optimales d'humidité pour le fonctionnement du capteur K33. Comparativement, le capteur CarbonAqua présente donc des avantages en ce qui concerne l'humidité relative. Cependant, le temps de réponse du capteur est relativement long, ce qui nécessite des améliorations du système en réalisant des cycles de circulations de l'air pour optimiser le temps d'équilibre entre la membrane et le milieu.

Ce dernier point est en cours d'amélioration. Les résultats futurs seront intégrés à ce rapport.

V. Bibliographie

HEGER, Adrian, KLEINSCHMIDT, Volker, GRÖNGRÖFT, Alexander, *et al.* Application of a low-cost NDIR sensor module for continuous measurements of in situ soil CO₂ concentration. *Journal of Plant Nutrition and Soil Science*, 2020, vol. 183, no 5, p. 557-561.

IPCC, 2022: Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [H.-O.

JOHNSON, Mark S., BILLET, Michael F., DINSMORE, Kerry J., *et al.* Direct and continuous measurement of dissolved carbon dioxide in freshwater aquatic systems—method and applications. *Ecohydrology: Ecosystems, Land and Water Process Interactions, Ecohydrogeomorphology*, 2010, vol. 3, no 1, p. 68-78.

TRANVIK, Lars J., DOWNING, John A., COTNER, James B., *et al.* Lakes and reservoirs as regulators of carbon cycling and climate. *Limnology and oceanography*, 2009, vol. 54, no 6part2, p. 2298-2314.

VAN DEN BOSSCHE, Michael, ROSE, Nathan Tyler, *et* DE WEKKER, Stephan Franz Joseph. Potential of a low-cost gas sensor for atmospheric methane monitoring. *Sensors and Actuators B: Chemical*, 2017, vol. 238, p. 501-509.

VI. Annexe

VI.1 Code d'Arduino

// une fermeture 33 secondes, mesurer 15 minutes, ouverture 33 secondes, laisser ouvert 2 minutes, fermer 33 secondes et mesurer 15 minutes

```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <RTClib.h>
#include <kSeries.h>
#include <SoftwareSerial.h>

#define LOG_INTERVAL 2000
#define SYNC_INTERVAL 5000

uint32_t syncTime = 0;
#define ECHO_TO_SERIAL 1
#define WAIT_TO_START 0
const int RELAY_PIN = 8;

RTC_DS1307 RTC;
const int chipSelect = 10;
File logfile;
double CO2ValueK33 = 0;
double RHValue = 0;
double TempValue = 0;
int co2Addr = 0x7F;

void wakeSensor() {

  TWCR &= ~(1<<2); // Disable I2C Engine
  DDRC |= (1<<4); // Set pin to output mode
  PORTC &= ~(1<<4); // Pull pin low
  delay(1);
  PORTC |= (1<<4); // Pull pin high again
  TWCR |= (1<<2); // I2C is now enabled
  delay(1);
}

void initPoll() {
  Wire.beginTransmission(co2Addr);
  Wire.write(0x11);
  Wire.write(0x00);
  Wire.write(0x60);
  Wire.write(0x35);
  Wire.write(0xA6);

  Wire.endTransmission();
  delay(20);
  Wire.requestFrom(co2Addr, 2);

  byte i = 0;
  byte buffer[2] = {0, 0};

  while(Wire.available()) {
    buffer[i] = Wire.read();
    i++;
  }
}
```

```

double readCO2(){
  int co2_value = 0;
  Wire.beginTransmission(co2Addr);
  Wire.write(0x22);
  Wire.write(0x00);
  Wire.write(0x08);
  Wire.write(0x2A);
  Wire.endTransmission();
  delay(50);
  Wire.requestFrom(co2Addr, 4);
  byte i = 0;
  byte buffer[4] = {0, 0, 0, 0};
  while (Wire.available()){
    buffer[i] = Wire.read();
    i++;
  }
  co2_value = 0;
  co2_value |= buffer[1] & 0xFF;
  co2_value = co2_value << 8;
  co2_value |= buffer[2] & 0xFF;
  byte sum = 0;
  sum = buffer[0] + buffer[1] + buffer[2];
  if (sum == buffer[3]){
    return ((double)co2_value * (double) 1);
  } else {
    delay(10);
    return ((double) -1);
  }
}

```

```

double readRH() {
  int RH_value = 0;
  Wire.beginTransmission(co2Addr);
  Wire.write(0x22);
  Wire.write(0x00);
  Wire.write(0x14);
  Wire.write(0x36);
  Wire.endTransmission();
  delay(20);

  Wire.requestFrom(co2Addr, 4);
  byte i = 0;
  byte buffer[4] = {0, 0, 0, 0};

  while(Wire.available()) {
    buffer[i] = Wire.read();
    i++;
  }

  RH_value = 0;
  RH_value |= buffer[1] & 0xFF;
  RH_value = RH_value << 8;
  RH_value |= buffer[2] & 0xFF;

  byte sum = 0;
  sum = buffer[0] + buffer[1] + buffer[2];
  if(sum == buffer[3]) {
    return ((double)RH_value / (double) 100);
  } else {
    return ((double) -1);
  }
}

```

```

double readTemp() {
  int Temp_value = 0;
  Wire.beginTransmission(co2Addr);
  Wire.write(0x22);
  Wire.write(0x00);
  Wire.write(0x12);
  Wire.write(0x34);
  Wire.endTransmission();
  delay(20);
  Wire.requestFrom(co2Addr, 4);
  byte i = 0;
  byte buffer[4] = {0, 0, 0, 0};
  while(Wire.available()) {
    buffer[i] = Wire.read();
    i++;
  }
  Temp_value = 0;
  Temp_value |= buffer[1] & 0xFF;
  Temp_value = Temp_value << 8;
  Temp_value |= buffer[2] & 0xFF;
  byte sum = 0;
  sum = buffer[0] + buffer[1] + buffer[2];
  if(sum == buffer[3]) {
    return ((double)Temp_value / (double) 100);
  } else {
    return ((double) -1);
  }
}

```

```

void setup(void){
  Serial.begin(9600);
  Serial.println();
  pinMode(RELAY_PIN, OUTPUT);

  Serial.print("Initialisation carte SD...");
  pinMode(10, OUTPUT);
  if (!SD.begin(chipSelect)) {
    Serial.println("Erreur. Carte non détectée.");
    while (true);
  } else {
    Serial.println("Initialisation terminée.");
  }
}

```

```

char filename[] = "LOGGER00.CSV";
for (uint8_t i = 0; i < 100; i++) {
  filename[6] = i/10 + '0';
  filename[7] = i%10 + '0';
  if (!SD.exists(filename)) {
    logfile = SD.open(filename, FILE_WRITE);
    break;
  }
}

```

```

Serial.print("Logging to: ");
Serial.println(filename);

```

```

Wire.begin();
RTC.begin();
if (!RTC.begin()) {
  logfile.println("RTC failed");
  #if ECHO_TO_SERIAL
  Serial.println("RTC failed");

```

```

#endif
}
//RTC.adjust(DateTime(__DATE__, __TIME__));

logfile.println("millis,stampunix,datetime,CO2ppm_K33,RH%,Temp");
#if ECHO_TO_SERIAL
Serial.println("millis, stampunix, datetime, CO2ppmK33, RH%, Temp");
#endif
}

void loop(){

digitalWrite(RELAY_PIN, LOW);

for (int t=0; t<2; t++){

wakeSensor();
initPoll();
delay(3000);

DateTime now;
String dataString = "";
wakeSensor();
double CO2ValueK33 = readCO2();
delay(50);

wakeSensor();
double RHValue = readRH();
delay(20);

wakeSensor();
double TempValue = readTemp();
delay(20);

now = RTC.now();
logfile.print(now.unixtime());
logfile.print(", ");
logfile.print(now.day(), DEC);
logfile.print("/");
logfile.print(now.month(), DEC);
logfile.print("/");
logfile.print(now.year(), DEC);
logfile.print(" ");
logfile.print(now.hour(), DEC);
logfile.print(":");
logfile.print(now.minute(), DEC);
logfile.print(":");
logfile.print(now.second(), DEC);

#if ECHO_TO_SERIAL
Serial.print(now.unixtime());
Serial.print(", ");
Serial.print(now.day(), DEC);
Serial.print("/");
Serial.print(now.month(), DEC);
Serial.print("/");
Serial.print(now.year(), DEC);
Serial.print(" ");
Serial.print(now.hour(), DEC);
Serial.print(":");
Serial.print(now.minute(), DEC);
Serial.print(":");
Serial.print(now.second(), DEC);

```

```
#endif

logfile.print(" ");
logfile.print(CO2ValueK33+90);
logfile.print(" ");
logfile.print(RHValue);
logfile.print(" ");
logfile.print(TempValue);

#if ECHO_TO_SERIAL
  Serial.print(" ");
  Serial.print(CO2ValueK33);
  Serial.print(" ");
  Serial.print(RHValue);
  Serial.print(" ");
  Serial.print(TempValue);
#endif //ECHO_TO_SERIAL

logfile.println();
#if ECHO_TO_SERIAL
  Serial.println();
#endif
logfile.flush();
delay(2000);
}

digitalWrite(RELAY_PIN, HIGH);
delay(5000);

}
```