



## Projet SEB

# Sonde de mesure sous-marine de l'Étang de Berre

---

ALONZO ALVAREZ, THOMAS BERMEJO, LORINE CARCELLER, ELIA CLINCHAMP,  
SOULEYMAN EL BYA, ADAM HELLAL, JIHANE KHATTABI, CAMILLE MANCINI, MATHYS  
PAYET ET LUCAS TERNAT

L'Étang de Berre, parmi les plus grandes lagunes d'eau saumâtre d'Europe, est un écosystème unique, fragilisé par les usines et les apports d'eau douce qui dérèglent régulièrement son équilibre. Un suivi régulier et fiable de la qualité de l'eau est donc primordial pour établir l'influence de cet environnement industriel. Le projet SEB a pour but de développer une sonde de mesure sous-marine autonome, permettant de caractériser des grandeurs physico-chimiques telles que la température, la turbidité, le pH, la conductimétrie et la teneur en O<sub>2</sub> dissous. Alimenté par une batterie et un panneau solaire, un microcontrôleur enregistre les données sur une carte SD et les envoie sur un serveur Web en libre accès. L'ensemble est protégé par une enceinte étanche, soutenue par une bouée insubmersible. Ce prototype ouvre la porte aux sciences participatives, mettant à la portée de tous le moyen de suivre en temps réel et de caractériser à l'aide de critères scientifiques la santé de l'Étang.

COLLABORATIONS : *8 Vies pour la planète* et *GIPREB*

SITE WEB : <https://explorelabz.org/projet-seb/>

VIDÉO CONCOURS CGÉNIAL : <https://youtu.be/6kwMmotUzXM>

LYCÉE COCTEAU  
AVENUE JEAN COCTEAU, 13140 MIRAMAS

ANNÉE SCOLAIRE 2022–2023  
PROJET ENCADRÉ PAR BASTIEN GUIGUE

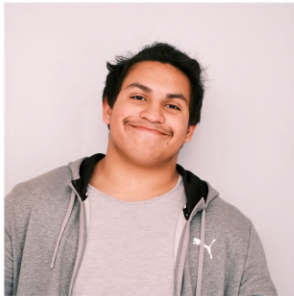


# Présentation de l'équipe



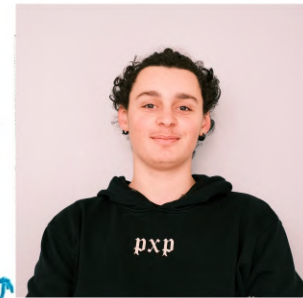
← Lorine  
aime l'aéronautique et coder  
future pilote

Adam  
passionné d'aéronautique  
expert soudure →



← Alomgo  
Rugbyman programmeur  
futur ingénieur en aéronautique

Lucas  
Bricoleur inventif  
expert sketchup →



← Elia  
fan de sciences et surtout d'Ariana Grande  
future Architecte

Carille  
N'a pas peur de se mouiller les pieds  
future psychiatre →



← Souleyman  
Lecteur acharné minutieux et perfectionniste  
futur comédien

Mathys  
Un scientifique qui bosse  
futur ingénieur en aérospatial →



← Thomas  
engagé dans la vie lycéenne  
futur astrophysicien

Jihame  
responsable photo et site WEB  
Monteuse vidéo fan de Zendaya →



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	L'écosystème de l'Étang de Berre . . . . .	1
1.2	Caractériser la pollution aquatique . . . . .	1
1.3	Description du projet et collaborations . . . . .	2
1.4	Problématique . . . . .	3
<b>2</b>	<b>Cahier des charges</b>	<b>3</b>
2.1	Design de la sonde . . . . .	3
2.2	Choix du microcontrôleur . . . . .	5
2.3	Choix des capteurs . . . . .	6
2.4	Alimentation autonome . . . . .	7
<b>3</b>	<b>Étalonnage des capteurs</b>	<b>7</b>
3.1	Thermomètre étanche . . . . .	7
3.2	pH-mètre . . . . .	8
3.3	Conductimètre . . . . .	9
3.4	Turbidimètre . . . . .	10
3.5	Oxygène dissous . . . . .	11
<b>4</b>	<b>Branchement et interfaçage des capteurs</b>	<b>12</b>
4.1	Schéma général . . . . .	12
4.2	Code Arduino . . . . .	13
<b>5</b>	<b>Acquisition des données, serveur Web et mise en sommeil</b>	<b>13</b>
5.1	Sauvegarde des données sur carte SD . . . . .	13
5.2	Envoi des données sur un serveur Web . . . . .	14
5.3	Mise en sommeil . . . . .	15
<b>6</b>	<b>Construction de la sonde</b>	<b>16</b>
<b>7</b>	<b>Tests d'étanchéité et résultats préliminaires</b>	<b>20</b>
<b>8</b>	<b>Communication autour du projet SEB</b>	<b>21</b>
<b>9</b>	<b>Conclusion et perspectives</b>	<b>22</b>

# 1 Introduction

## 1.1 L'écosystème de l'Étang de Berre

L'Étang de Berre est une lagune d'eau saumâtre, avec une superficie d'environ 155 km<sup>2</sup>, ce qui en fait une des plus grande d'Europe. Sa profondeur peut atteindre 9 mètres, et son bassin versant naturel (eau récoltée par ses affluents) est de 1700 km<sup>2</sup>. Sur les pourtours de l'étang sont installées de nombreuses industries, parmi lesquelles la raffinerie pétrolière de Fos-sur-Mer au sud et la centrale hydroélectrique EDF située à l'embouchure du canal Durance-Verdon au nord. L'étang est alors alimenté de manière importante et artificielle par la centrale EDF, dont les rejets d'eau douce représentent 3 million de m<sup>3</sup> par an, soit 3,7 fois le volume de l'étang. Cet apport d'eau douce important et localisé a plusieurs effets sur l'écosystème. La salinité de l'eau d'origine saumâtre diminue, et la stratification du milieu, du fait de la présence d'eau salée et d'eau douce de densités différentes, limite fortement le brassage. Cela provoque des conditions d'anoxie (forte diminution de la concentration en oxygène dissous) dans les couches inférieures. Le taux de poissons a par exemple nettement diminué, ce qui contraint les pêcheurs locaux à être subventionnés. À l'été 2018, suite aux rejets de la centrale, des fortes chaleurs et d'une absence de vent, l'étang a connu une de ses pires crises anoxiques, avec la mort de milliers de poissons, perturbant la chaîne alimentaire et dérégulant complètement l'équilibre de l'étang. Il a par ailleurs été remarqué que la colorimétrie de l'eau changeait en présence des rejets d'eau de l'entreprise, indicatrice de l'altération des algues et du plancton (rouges, verts, et luminescents la nuit).

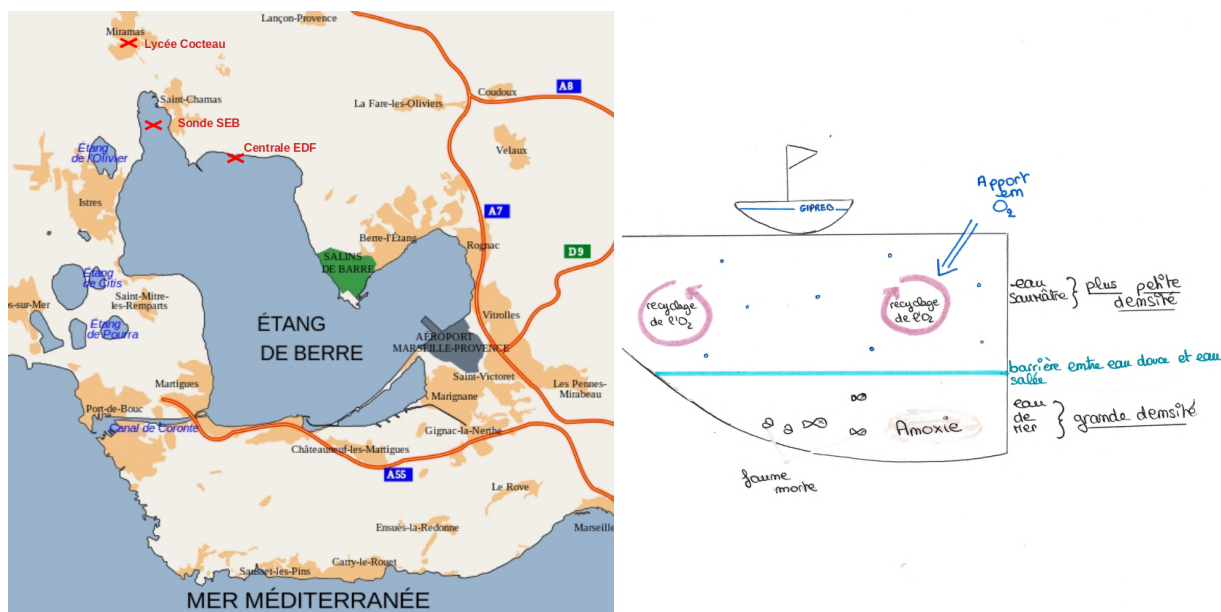


FIGURE 1 – Gauche : Carte de l'Étang de Berre, avec l'emplacement du lycée, de la centrale EDF et de la sonde SEB (croix rouges). Droite : Schéma de la stratification de l'étang conduisant à l'anoxie.

Ainsi, si la pollution de l'air du fait des industries pétrolières est importante dans la région, c'est avant tout la pollution aquatique résultant des rejets de la centrale EDF qui vient perturber l'écosystème de l'Étang de Berre.

## 1.2 Caractériser la pollution aquatique

Le terme de pollution aquatique, relativement vague, englobe en fait plusieurs paramètres physico-chimiques, et constitue une altération sur plus ou moins long terme de ces paramètres avec pour conséquence un impact sur la faune et la flore de l'écosystème.

- La **température** est un paramètre physique particulièrement important, car elle peut être un facteur primordial pour la survie des espèces aquatiques (faune et flore). La température va

également influencer d'autres paramètres, et en particulier la concentration en gaz dissous dans l'eau, dont l'oxygène.

- La **turbidité** est un paramètre physique représentant la quantité de particules solides en suspension dans un milieu aqueux. Plus simplement, une mesure de la turbidité va permettre de savoir si l'eau est plus ou moins trouble.
- La **conductivité** est un paramètre physique permettant de quantifier la capacité d'un milieu à conduire l'électricité. Elle est directement liée à la quantité d'ions dissous dans l'eau, porteurs de charges électriques et responsables de la conductivité en milieu aqueux. Une mesure de la conductivité va permettre de détecter principalement deux sources de pollution : un changement de la salinité (comme c'est le cas ici) ou la présence de produits chimiques (par exemple des métaux lourds, possédant une très grande conductivité).
- Le **pH** est un paramètre chimique représentant la concentration en ions  $H^+$  dans un milieu. Un changement de l'acidité d'un écosystème peut altérer les conditions de survie de la faune et de la flore. Un rejet de produits chimiques peut facilement faire varier le pH.
- La **concentration en  $O_2$  dissous** est un paramètre chimique important, l'oxygène étant nécessaire à la survie des espèces aquatiques. La variation de la concentration en  $O_2$  dissous peut être causée par plusieurs paramètres : la stratification du milieu du fait de masses d'eau de densités différentes, la température de l'eau, une absence de vent pour renouveler l'apport en oxygène...

### 1.3 Description du projet et collaborations

Le projet SEB (pour Sonde de mesure sous-marine de l'Étang de Berre) est mené par l'atelier scientifique du lycée Cocteau à Miramas. L'équipe est constituée d'élèves volontaires de classe de 1ère (générale et STMG). L'étang de Berre se situe à 5 km du lycée, et la plupart d'entre nous habite encore plus près. Nous nous baignons dans l'étang et bronçons sur ses bords, y faisons de la planche à voile, de la pêche... Et nous sommes tous témoins de sa pollution (poissons morts échoués en 2018, proliférations régulières d'algues vertes, déchets en tous genres). Le but de notre projet est de créer une sonde de mesure réunissant plusieurs capteurs (température, conductimétrie, pH, turbidité, oxygène dissous) et de la placer dans l'étang, afin de surveiller sa santé au quotidien et d'étudier ensuite l'influence des industries qui le bordent sur la qualité de l'eau. Nous avons également choisi de lui donner la capacité d'envoyer les mesures en temps réel sur un serveur Web en libre accès. En effet, il nous semblait important que quiconque puisse avoir accès à ses mesures et puisse les utiliser. Le fait qu'elles soient disponibles en temps réel permet aussi de réagir rapidement si une anomalie trop importante est détectée.

La création de cet atelier était donc pour nous l'opportunité de mettre nos connaissances et notre passion pour la science en commun et au service de l'écologie (et plus particulièrement de l'Étang de Berre). Nous avons tous tenu plusieurs rôles, et avons essayé de nous partager les tâches – à part Jihane, qui s'est occupée seule de prendre l'ensemble des photos, de créer le blog et de mettre les articles et ressources en ligne. Pour beaucoup d'entre nous, nous avons pu apprendre de nouvelles compétences : soudure, perçage et découpage, étalonnage des capteurs, programmation Arduino, fonctionnement d'un panneau solaire, fonctionnement du réseau 4G, création d'un site Web et d'un blog... etc !

Au cours de ce projet, nous avons également rencontré plusieurs personnes travaillant au service de l'Étang. Nous avons collaboré avec l'association 8 Vies pour la planète, basée à Saint-Chamas, qui possède un Fablab et un bateau permettant de faire des mesures dans l'étang. Damien, le président de l'association, et Guillaume, sont régulièrement venus nous voir au lycée pour nous aider dans l'avan-

cement du projet. Ils nous ont également aidés pour obtenir les autorisations pour placer notre sonde dans l'étang. Nous avons également collaboré avec le GIPREB, syndicat mixte assurant le suivi scientifique de l'étang. Le GIPREB nous a apporté un soutien logistique en particulier pour les autorisations autour de la mise à l'eau. Le GIPREB fournit également des données (température de l'eau, force et direction du vent, salinité) afin de comparer nos résultats et les interpréter.

Enfin, le projet a été documenté à travers un blog, que nous avons alimenté régulièrement pour partager notre aventure avec nos familles et nos amis. À la fin de l'année, nous avons prévu de mettre en ligne toutes les ressources de notre projet sur le site pour que d'autres lycées et associations puissent s'en servir et développer des projets similaires. Ce site est consultable en suivant [ce lien](#) ! Il sera mis à jour au fil des prochains mois avec de nouveaux postes, ressources et liens. Aujourd'hui, la sonde SEB est presque prête à être mise à l'eau. Sa construction est terminée, les capteurs sont étalonnés et le code Arduino permettant d'enregistrer les données et de les envoyer sur un serveur Web fonctionne. Seuls deux problèmes (étanchéité et interférences entre capteurs) subsistent, et nous espérons les régler rapidement.

## 1.4 Problématique

La fabrication de cette sonde devrait permettre de répondre à plusieurs problématiques, gagnant en complexité à mesure que le projet SEB évoluera. Cette année, avec la fabrication d'une première sonde autonome permettant de mesurer différents paramètres physico-chimiques sur une longue durée dans l'étang, nous devrions déjà pouvoir donner plusieurs éléments de réponse à la problématique :

**Comment la proximité des industries, et plus particulièrement la centrale hydroélectrique de Saint-Chamas et ses rejets d'eau douce, influencent-ils la qualité de l'eau dans l'Étang de Berre ?**

La sonde sera placée à proximité immédiate de la zone protégée du parc de la Poudrerie, située au nord de l'étang et proche de l'embouchure du canal Durance-Verdon et de la centrale. En mesurant la température, la conductivité, le pH, la turbidité et la concentration en  $O_2$  dissous régulièrement pendant un mois, nous pourrions détecter d'éventuelles anomalies voire caractériser des changements se faisant sur des échelles de temps plus importantes. Afin d'interpréter l'origine de ces anomalies dans le comportement des indicateurs physico-chimiques qui nous intéressent, nous pourrions chercher d'éventuelles corrélations avec des données météorologiques (température, anémométrie, pluviométrie notamment) ainsi qu'avec les informations que fournit EDF sur la programmation et le volume de ses rejets d'eau douce.

## 2 Cahier des charges

### 2.1 Design de la sonde

Avant d'aboutir à notre design actuel, nous avons imaginé la sonde sous plusieurs formes. Il était nécessaire de prendre en compte tous les paramètres : comment et où placer les capteurs ? Le micro-contrôleur ? Voulions-nous une sonde flottante, ou totalement immergée ? Comment faire en sorte que la sonde ne coule pas dans le fond de l'étang, et résiste aux vagues ? Comment la rendre parfaitement étanche ? Comment procéder afin de lui garantir une totale autonomie d'énergie ? Comment lire les résultats de nos mesures à distance ?

Comme nous le verrons à la section suivante, nous avons opté pour une alimentation utilisant un panneau solaire, et l'envoi de données par 4G avec une antenne : la sonde devait donc être flottante. Par ailleurs, nous pensions qu'il fallait mettre les capteurs à une certaine profondeur plutôt qu'en surface, pour que les mesures soient plus représentatives.

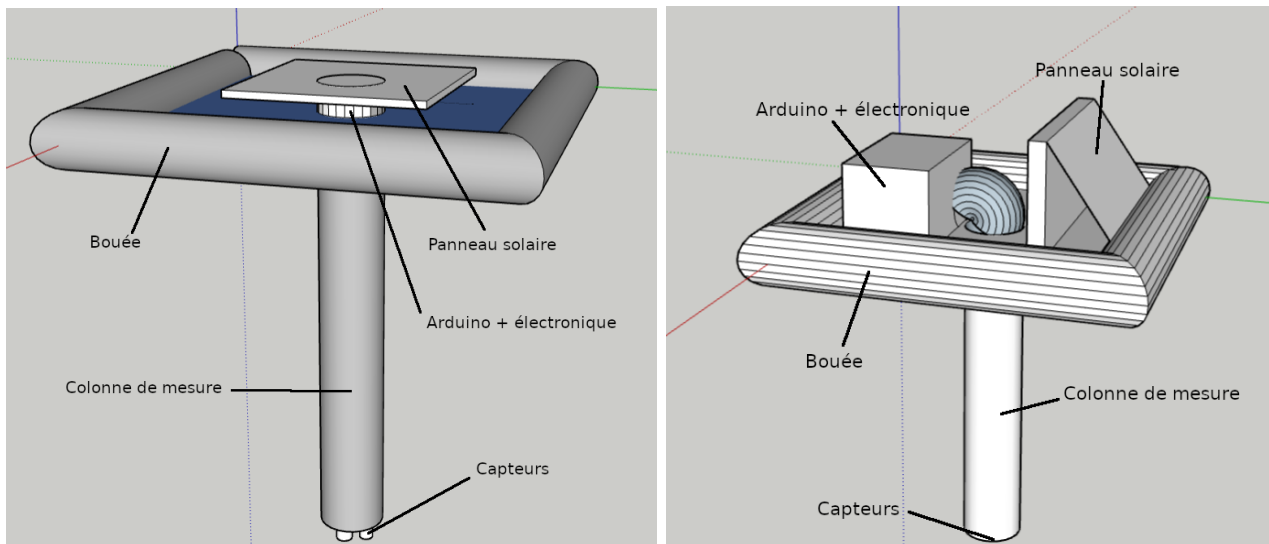


FIGURE 2 – Modélisations SketchUp des designs concurrents de la sonde, sur le principe d’une bouée carrée supportant une plateforme et d’une colonne de mesure maintenant les capteurs sous l’eau. Gauche : L’électronique est embarquée dans la colonne de mesure et le panneau solaire est placé au sommet. Droite : L’électronique et le panneau solaire sont placés sur la plateforme.

Nous avons prévu d’utiliser un tube en PVC de 10 cm de diamètre pour amener les capteurs à une profondeur d’environ 70 cm. Son extrémité, bouchée avec un tampon PVC, est percée afin de laisser passer les capteurs. Cette colonne de mesure est maintenue en place grâce à une bouée, elle aussi construite à partir de tubes PVC et de coudes droits. La bouée forme un carré de 50 cm de côté, avec des tuyaux de 10 cm de diamètre. Le PVC répond ici parfaitement à notre besoin puisqu’il est facile à trouver en magasin, peu coûteux, étanche et facile à travailler.

Pour savoir si la bouée suffit à maintenir l’ensemble hors de l’eau, il faut calculer sa flottabilité, c’est à dire la différence entre la masse d’eau déplacée par le volume de la bouée, et la masse de la sonde. La bouée peut être modélisée par 4 cylindres de diamètre 10 cm et de longueur 50 cm. Son volume est donc d’environ  $4 \times 5,0 \times \pi(0,50)^2 = 16 \text{ dm}^3$ . La densité de l’eau salée valant environ 1,03, on a une masse d’eau déplacée d’environ 16 kg. Or la bouée pèse environ 5,3 kg : elle peut donc supporter une charge de 10,7 kg, largement suffisant pour supporter un panneau solaire, une batterie et toute l’électronique.

L’interface entre la bouée et la colonne de mesure est une plaque PVC carrée de 50 cm de côté, et 4 mm d’épaisseur, sur laquelle nous allons fixer des équerres métalliques qui maintiendront la colonne de mesure. La plaque est percée et attachée à la bouée avec des serflex. Si cette plaque ne possède pas d’autre utilité pour le moment, nous avons pensé qu’il pourrait être utile d’avoir un espace pour de futures modifications de notre sonde (par exemple l’ajout de capteurs de mesure de qualité de l’air et d’anémométrie).

Enfin, il a fallu concevoir la partie accueillant le microcontrôleur, la batterie et les cartes électroniques des capteurs. Nous avons longtemps hésité entre deux possibilités, présentées sur les figures ci-dessus : tout mettre à l’intérieur du tube de mesure, et couvrir le tout avec le panneau solaire, afin de minimiser le nombre de trous à percer, ou bien utiliser une boîte extérieure étanche, plus facile d’accès. Nous avons finalement opté pour la première option, et nous avons conçu un support cylindrique pour fixer toute la partie électronique et la glisser dans la colonne de mesure. Le panneau solaire sera attaché au bouchon de la colonne de mesure, et sera ainsi le seul fil électrique à faire passer au travers.

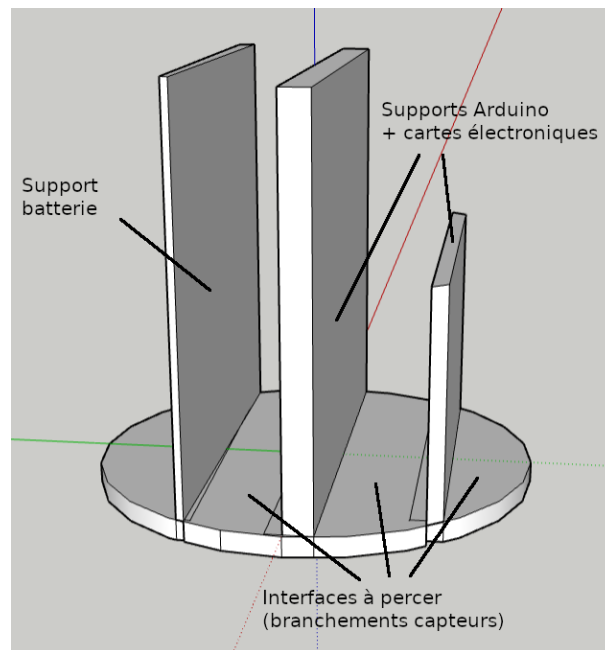


FIGURE 3 – Modélisation SketchUp du support Arduino / cartes électroniques. Le disque doit être percé pour permettre le passage des fils. Arduino et les cartes des capteurs sont vissés sur les deux plaques de droite, la batterie se loge entre les deux plaques de gauche. L'ensemble se glisse dans la colonne de mesure.

## 2.2 Choix du microcontrôleur

Il existe des dizaines de microcontrôleurs différents, et il a fallu en choisir un qui réponde à nos besoins :

- Il doit être facile à programmer (Python ou Arduino)
- Cela doit être facile de lui ajouter des capteurs
- Il doit pouvoir fonctionner sur batterie et être peu gourmand en énergie
- Il doit pouvoir enregistrer des données sur une carte SD
- Il doit pouvoir se connecter à Internet

Les deux familles principales sont microbit et Arduino. microbit se code en Python, fonctionne sur batterie, mais tout ce qui est capteurs complexes et Internet sont plus difficiles à mettre en place. Chez Arduino, il y a l'Arduino Uno (le plus basique), le Mega (le plus gros), et la famille MKR. Uno et Mega consomment beaucoup d'énergie et la connectivité Internet est limitée. La famille MKR fonctionne à 3,3 V (contre 5 V pour Uno et Mega) et consomme donc moins d'énergie. Elle est plus petite, elle est conçue pour se connecter à Internet, et possède un emplacement pour brancher directement une batterie LiPo et un emplacement pour carte SD. Pour la connectivité Internet, il y a plusieurs possibilités. Comme la bouée sera au milieu de l'étang, on ne peut pas se connecter en Wifi, mais l'étang est couvert par la 4G : nous avons donc choisi la carte Arduino MKR NB1500, qui possède un petit modem 4G LTE-M, et nous avons acheté une carte SIM auprès de ThingsMobile. Nous avons également choisi une antenne Antenova SRFL026 compatible avec les bandes de fréquences 4G LTE-M.



## 2.3 Choix des capteurs

Le choix des capteurs s’est fait selon plusieurs critères : peu coûteux, pouvant fonctionner en 3,3 V (la tension de fonctionnement de l’Arduino MKR NB1500), étanches et résistants à l’eau salée.

- **Capteur de température** : Le capteur de température que nous avons choisi est un thermomètre étanche basé sur une sonde DS18B20 et proposé par DFRobot. Il s’agit d’un thermomètre digital, pré-calibré, utilisant un transistor comme capteur de température. Étant digital, il n’y a pas de dégradation du signal même sur de longues distances (environ 1 mètre ici). Ce capteur respecte parfaitement notre cahier des charges : la tension de fonctionnement doit être comprise entre de 3,0 à 5,5 V, la plage de température de fonctionnement est de  $-55$  à  $125^{\circ}\text{C}$ . L’erreur de mesure est de  $0,5^{\circ}\text{C}$ . Son coût est très bas (environ 5 euros).
- **pH-mètre** : Le pH-mètre que nous avons choisi est la sonde SEN0161-V2 proposée par DFRobot. Elle est couramment utilisée pour les tests environnementaux dans l’eau, car sa construction est prévue pour résister à une utilisation plus intensive et à l’eau salée. Sa tension d’utilisation doit être entre de 3,3 à 5,5 V. Sa précision est de plus ou moins 0,1 unité de pH et sa plage de température de fonctionnement de 5 à  $60^{\circ}\text{C}$  (ce qui la rend inutilisable en hiver). Cette sonde est relativement chère (60 euros environ) mais sa résistance à l’eau salée est un atout.
- **Conductimètre** : Le conductimètre choisi est la référence DFR0300-H proposée par DFRobot. Il s’agit d’une sonde prévue pour une utilisation en laboratoire mais qui possède une constante de cellule élevée. Ce point a conditionné notre choix : l’eau salée possède une conductivité très élevée et il faut une sonde capable de mesurer une conductivité de l’ordre de plusieurs dizaines de mS/cm. Cela exclut donc les sondes bon marché de type “TDS” (Total Dissolved Solids), ainsi que les sondes plus élaborées avec une constante de cellule de l’ordre de 1. La tension de fonctionnement de cette sonde doit être comprise entre 3 et 5 V, et elle fonctionne entre 0 à  $40^{\circ}\text{C}$ . L’erreur de mesure est d’environ 5%. La durée de vie de ce capteur est limitée, et est de 6 mois environ en utilisation continue. Son coût est élevé (environ 100 euros). Ici, c’est la compatibilité avec une conductivité très élevée qui a imposé notre choix.
- **Turbidimètre** : Le capteur choisi correspond à la référence SEN0189 de chez DFRobot. Il s’agit d’un capteur rudimentaire, et donc peu coûteux (environ 5 euros). Sa tension de fonctionnement est de 5 V, ce qui n’est pas compatible a priori avec notre Arduino NB1500. Mais nous avons modifié une résistance sur sa carte électronique pour le rendre compatible avec 3,3 V. Une autre possibilité aurait été d’utiliser un “booster” de tension pour l’entrée et un pont diviseur de tension pour la sortie analogique. Il supporte des températures entre 5 et  $90^{\circ}\text{C}$ , ce qui le rend inutilisable en hiver. Il n’existe pas d’alternatives intermédiaires en terme de prix ou de performances : un meilleur capteur aurait été beaucoup plus cher, et nous nous sommes donc contentés de celui-ci pour ce premier prototype.
- **Capteur d’ $\text{O}_2$  dissous** : Nous avons choisi la référence SEN0237 de chez DFRobot, seule sonde de ce genre disponible. Elle possède un capuchon à membrane devant être rempli d’une solution de NaOH à  $0,5 \text{ mol.L}^{-1}$ . Cette membrane est très fragile et doit être changée au bout de 5 mois maximum pour une utilisation continue. La solution de NaOH doit être renouvelée tous les mois : cela limite donc l’immersion de notre sonde à une durée d’un mois maximum sans la moindre intervention. La durée de vie globale de la sonde est d’environ 1 an, et elle est relativement chère (environ 80 euros). Sa tension de fonctionnement est comprise entre 3,3 et 5,5 V, et accepte une plage de températures entre 0 et  $40^{\circ}\text{C}$ . Ces caractéristiques correspondant à nos besoins, et aucune alternative n’existait, nous avons choisi cette sonde malgré les contraintes sur la durée d’immersion.

## 2.4 Alimentation autonome

Comme la sonde est placée au milieu de l'étang, elle ne peut évidemment pas être branchée sur le secteur ou sur un ordinateur. Il faut donc qu'elle soit autonome grâce à une source d'énergie embarquée. Par ailleurs, comme nous comptons la laisser en place pendant un mois, il faut que cette source d'énergie soit suffisamment importante, ou bien qu'elle se renouvelle. Le choix évident est d'utiliser une batterie avec un panneau solaire : quand il fait jour, le panneau solaire fait fonctionner la sonde, et charge la batterie. Quand il fait nuit, ou nuageux, la batterie prend le relais.

Nous avons choisi un panneau solaire 6 V 9 W et une batterie LiPo 3,7 V 10000 mAh de chez Adafruit. Si les batteries LiPo sont très compactes, elles supportent mal d'être chargées tout en devant délivrer du courant. Nous avons donc également installé une carte Adafruit bq24074, qui permet de gérer les différents besoins (consommation de la carte Arduino et charge de la batterie) de manière intelligente. La priorité est donnée à la consommation Arduino, et la batterie n'est chargée que lorsque le panneau solaire produit plus de courant que nécessaire.

La tension de 6 V du panneau solaire est la valeur nécessaire au bon fonctionnement de la carte Adafruit bq24074. Nous avons choisi une puissance de 9 W car c'est le maximum proposé dans cette gamme, permettant ainsi de nous trouver le plus régulièrement possible dans la configuration où il y aura suffisamment de courant pour charger la batterie en plus de faire fonctionner Arduino.

La tension de 3,7 V de la batterie LiPo est celle compatible avec la carte Arduino MKR NB1500, qui fonctionne en 3,3 V mais avec un emplacement pour batteries 3,7 V. Nous avons choisi une capacité de 10000 mAh pour la batterie pour permettre à la sonde de fonctionner suffisamment longtemps en cas de mauvaise météo. En effet, le processeur et le modem de la carte consomment environ 100 mA lorsqu'ils sont allumés, et 1 mA environ en mode veille. Les capteurs consomment environ 20 mA chacun lorsqu'ils sont mis sous tension. Toute la sonde consomme donc environ 200 mA lorsqu'elle est allumée, et 1 mA en mode veille. Si on allume la sonde pendant 10 minutes, toutes les 3 heures, cela fait un total de 1,2 h d'utilisation sur une journée. La sonde consomme donc 256 mAh en un jour. Avec une batterie de 10000 mAh, cela correspond à un mois de fonctionnement environ.

## 3 Étalonnage des capteurs

L'étalonnage des capteurs est une étape très importante du projet. Un capteur est un outil basé sur un composant/système électronique dont la tension aux bornes est dépendante d'un paramètre physico-chimique. La fonction exacte liant la tension et ce paramètre dépend souvent de plusieurs facteurs propres au capteur (dimensions de certaines surfaces actives, état de ces surfaces, valeurs précises des composants électroniques...) et il est donc nécessaire d'étalonner chaque capteur indépendamment avant de les utiliser. Nous avons décidé d'étalonner nos capteurs directement, sans passer par les bibliothèques Arduino proposées par le fabricant, car celles-ci imposent de re-calibrer les capteurs avant chaque expérience. L'étalonnage direct des capteurs nous a aussi permis de mieux comprendre le fonctionnement de chacun d'entre eux.

### 3.1 Thermomètre étanche

Le thermomètre étanche (basé sur une sonde DS18B20) est le seul capteur que nous n'avons pas calibré, car il utilise un protocole de communication un peu spécial (à "un fil"). Nous avons donc rédigé un code Arduino affichant simplement la température en degrés Celsius toutes les secondes, et nous l'avons comparée avec celle donnée par un thermomètre commercial calibré. Nous avons fait ces mesures pour plusieurs températures, dans la plage de valeurs typiques pour l'Étang de Berre. Nous avons utilisé un mélange d'eau chaude (35°C environ) et d'eau froide (10°C environ), dans des proportions différentes pour atteindre la température souhaitée. Le résultat de ces mesures est donné dans le tableau suivant :

Sonde DS18B20 (°C)	10,8	14,8	20,7	25,2	29,9	35,0
Thermomètre commercial (°C)	9,5	15,0	20,4	25,0	29,3	33,9

Pour vérifier l'étalonnage de la sonde de température, il faut maintenant tracer ces données sur un graphe et les comparer à la droite d'équation  $y = x$ .

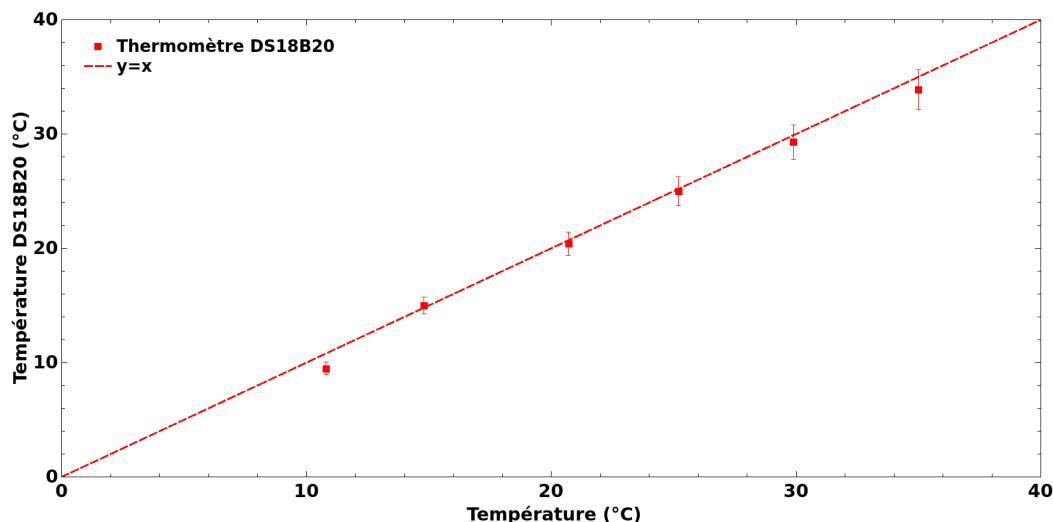


FIGURE 4 – Température mesurée par la sonde DS18B20 en fonction de la température mesurée par un thermomètre commercial étalonné. L'erreur commise est de l'ordre de 0,5% d'après la documentation fabricant.

Le point autour de 10°C est le seul dont les barres d'erreur sont en dehors de la droite, indiquant une erreur qui n'est pas uniquement celle de la sonde. La solution étant la plus froide, il est probable que le temps de faire les deux mesures elle se soit davantage réchauffée que dans le cas des autres solutions. Toutefois, à part ce point et en prenant en compte les erreurs de mesure, on voit que l'étalonnage de la sonde DS18B20 est parfaitement acceptable, les points étant bien en accord avec la droite théorique  $y = x$ .

### 3.2 pH-mètre

La relation entre la tension aux bornes du pH-mètre et le pH doit être une fonction affine. Afin de trouver les constantes propres à notre sonde, nous avons utilisé un pH-mètre commercial étalonné, ainsi que 5 solutions : 3 solutions "étalons" (pH 4,0, pH 7,0 et pH 10,0) et 2 solutions de pH intermédiaires. Celles-ci sont un mélange de 46,40 mL d'une solution de  $\text{Na}_2\text{HPO}_4$  à  $0,50 \text{ mol.L}^{-1}$  et de 16,80 mL d'acide citrique à  $0,50 \text{ mol.L}^{-1}$  (pH  $\approx 5,6$ ), ainsi qu'un mélange de 189 mL d'une solution de  $\text{Na}_2\text{HPO}_4$  à  $0,067 \text{ mol.L}^{-1}$  et de 11,0 mL d'une solution de  $\text{KH}_2\text{PO}_4$  à  $0,067 \text{ mol.L}^{-1}$  (pH  $\approx 8,0$ ). La sonde pH-métrique est utilisée avec Arduino, le code affiche la tension aux bornes de la sonde toutes les secondes. Pour plus de précision, Arduino fait l'acquisition de 8 mesures à la suite, qui sont ensuite moyennées. Le résultat de ces mesures est donné dans le tableau suivant :

Solution (pH théorique)	4,0	5,6	7,0	8,0	10,0
Sonde SEN0161-V2 (mV)	92,80	503,3	1201	1737	2019
pH-mètre commercial (pH)	4,0	5,6	7,0	8,1	9,1

Il faut maintenant tracer la droite d'étalonnage, et faire une régression linéaire pour trouver la fonction reliant la tension  $U$  et le pH.

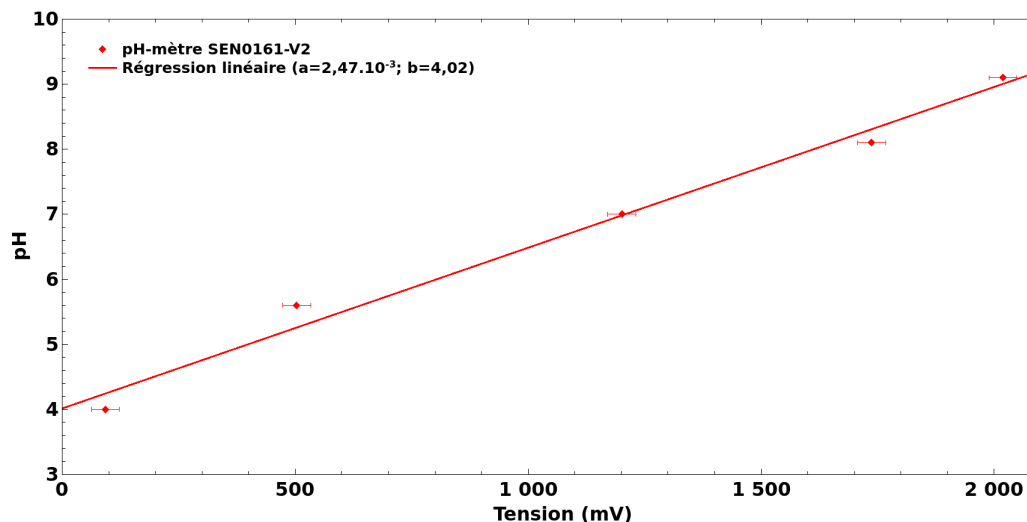


FIGURE 5 – pH de la solution en fonction de la tension aux bornes de la sonde. En rouge, régression linéaire de nos mesures, avec les paramètres de régression  $a = 2,47 \cdot 10^{-3} \text{ mV}^{-1}$  et  $b = 4,02$ . On trouve  $r^2 = 0,98$ . L'erreur commise sur le pH est de l'ordre de 0,1 d'après le fabricant, soit environ 10 mV. L'erreur statistique lors de la lecture est de l'ordre de 30 mV.

Grâce à la régression linéaire, on trouve la fonction reliant le pH à la tension  $U$  aux bornes de la sonde :

$$\boxed{\text{pH} = 2,47 \cdot 10^{-3} \times U + 4,02} \quad (1)$$

Cette fonction sera utilisée dans le code Arduino final pour que l'on mesure directement le pH de l'eau.

### 3.3 Conductimètre

La relation entre la tension aux bornes du conductimètre et la conductivité doit être une fonction affine. Toutefois, la conductivité dépend de la température, et il faut, d'après les études, prendre en compte un coefficient de compensation dépendant de la température  $T$  :

$$C_T = 1,0 + 0,0185 \times (T - 25,0) \quad (2)$$

La tension "compensée" que l'on doit utiliser pour l'étalonnage de la sonde vaut, en fonction de la tension mesurée  $U_{brut}$  et du coefficient  $C_T$  :

$$V = \frac{U_{brut}}{C_T} \quad (3)$$

Afin de trouver les constantes propres à notre sonde, nous avons donc utilisé un conductimètre commercial étalonné, un thermomètre (pour mesurer le coefficient de compensation et l'introduire dans le code Arduino) et 6 solutions de sel NaCl de concentrations massiques  $6,0 \text{ g.L}^{-1}$ ,  $12 \text{ g.L}^{-1}$ ,  $18 \text{ g.L}^{-1}$ ,  $24 \text{ g.L}^{-1}$ ,  $30 \text{ g.L}^{-1}$  et  $36 \text{ g.L}^{-1}$ . La sonde conductimétrique est utilisée avec Arduino, le code affiche la tension (compensée) aux bornes de la sonde toutes les secondes. Pour plus de précision, Arduino fait l'acquisition de 8 mesures à la suite, qui sont ensuite moyennées. Le résultat de ces mesures est donné dans le tableau suivant :

Concentration NaCl ( $\text{g.L}^{-1}$ )	6,0	12	18	24	30	36
Sonde DFR0300-H (mV)	284,5	558,1	772,4	959,9	1157	1353
Conductimètre commercial ( $\text{mS.cm}^{-1}$ )	11	21	30	38	46	53

Il faut maintenant tracer la droite d'étalonnage, et faire une régression linéaire pour trouver la fonction reliant la tension et la conductivité.

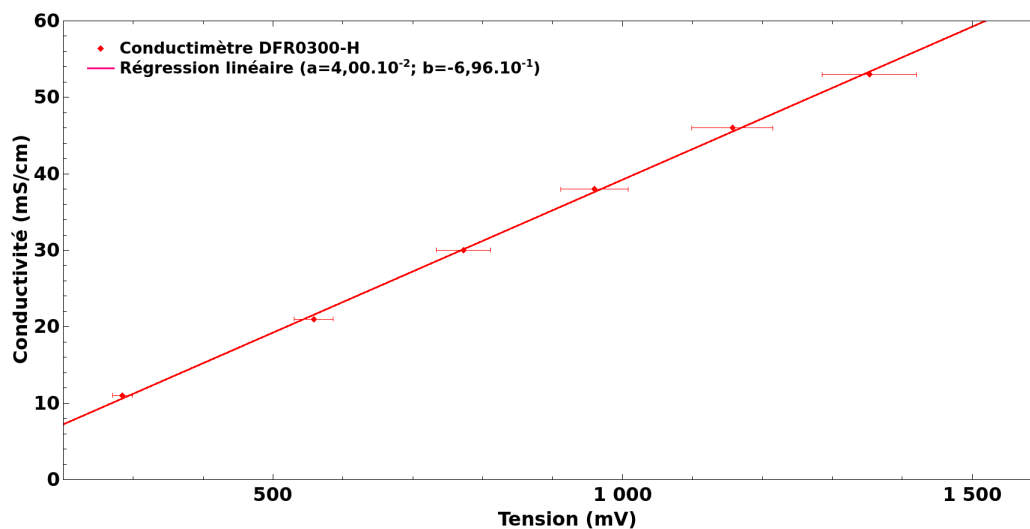


FIGURE 6 – Conductivité de la solution en fonction de la tension aux bornes de la sonde. En rouge, régression linéaire de nos mesures, avec les paramètres de régression  $a = 4,00.10^{-2} \text{ mS.cm}^{-1}.\text{mV}^{-1}$  et  $b = -6,96.10^{-1} \text{ mS.cm}^{-1}$ . On trouve  $r^2 = 0,99$ . L'erreur commise sur la tension est de l'ordre de 5% d'après le fabricant, supérieure à l'erreur statistique de lecture.

Grâce à la régression linéaire, on trouve la fonction reliant la conductivité  $\sigma$  à la tension  $U$  aux bornes de la sonde :

$$\sigma = 4,00.10^{-2} \times U - 6,96.10^{-1} \quad (4)$$

Cette fonction sera utilisée dans le code Arduino final pour que l'on mesure directement la conductivité de l'eau (en  $\text{mS.cm}^{-1}$ ).

### 3.4 Turbidimètre

La relation entre la tension aux bornes du turbidimètre et la turbidité doit être une fonction polynomiale (d'ordre 2). Afin de trouver les constantes propres à notre sonde, nous avons utilisé 8 solutions obtenues à partir de la dilution d'une solution de formazine étalonnée, possédant une turbidité de 4000 NTU (Nephelometric Turbidity Unit). La turbidité étant proportionnelle à la concentration molaire, une solution obtenue à partir de cette solution mère diluée  $n$  fois possède une turbidité de  $4000/n$ . Ainsi, nous avons dilué la solution mère pour obtenir des solutions de turbidité 4000 NTU, 2000 NTU, 1000 NTU, 500,0 NTU, 250,0 NTU, 125,0 NTU et 62,50 NTU. La sonde de turbidité est utilisée avec Arduino, le code affiche la tension aux bornes de la sonde toutes les secondes. Pour plus de précision, Arduino fait l'acquisition de 8 mesures à la suite, qui sont ensuite moyennées. Le résultat de ces mesures est donné dans le tableau suivant :

Turbidité (NTU)	4000	2000	1000	500,0	250,0	125,0	62,50
Sonde SEN0189 (mV)	415	688	883	1005	1067	1103	1123

Il faut maintenant tracer la courbe d'étalonnage, et faire une régression polynomiale pour trouver la fonction reliant la tension  $U$  et la turbidité.

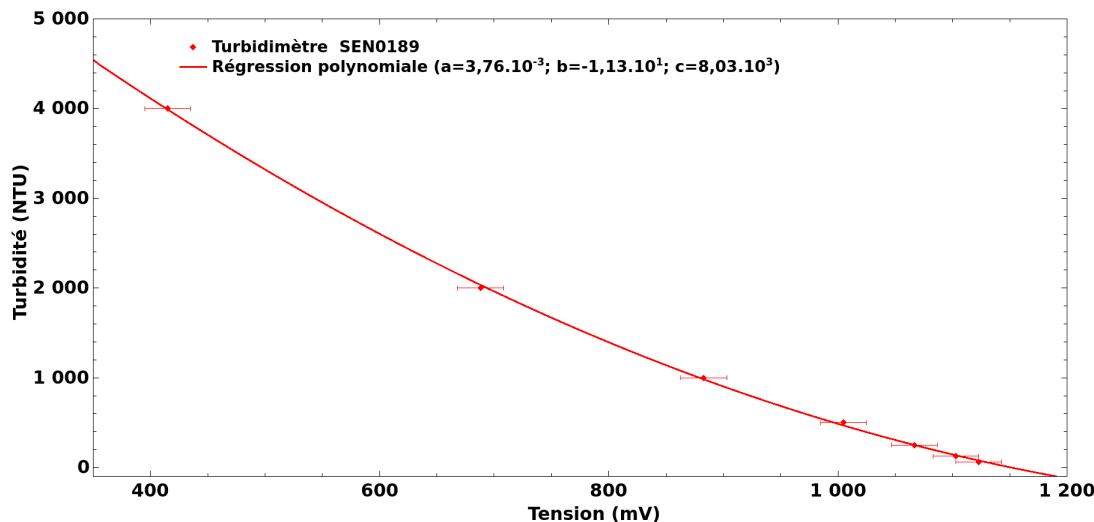


FIGURE 7 – Turbidité de la solution en fonction de la tension aux bornes de la sonde. En rouge, régression polynomiale de nos mesures, avec les paramètres de régression  $a = 3,76 \cdot 10^{-3}$  NTU.mV $^{-2}$ ,  $b = -1,13 \cdot 10^1$  NTU.mV $^{-1}$  et  $c = 8,03 \cdot 10^3$  NTU. On trouve  $r^2 = 0,96$ . L'erreur statistique commise sur la lecture de la tension est de l'ordre de 20 mV.

Grâce à la régression polynomiale, on trouve la fonction reliant la turbidité à la tension  $U$  aux bornes de la sonde :

$$\boxed{\text{Turbidité} = 3,76 \cdot 10^{-3} \times U^2 - 1,13 \cdot 10^1 \times U + 8,03 \cdot 10^3} \quad (5)$$

Cette fonction sera utilisée dans le code Arduino final pour que l'on mesure directement la turbidité de l'eau (en NTU). On note cependant que les mesures seront peu précises : la turbidité de l'eau se situe typiquement entre 0 et 100 NTU. La précision de la sonde permet difficilement de différencier ces valeurs.

### 3.5 Oxygène dissous

Dans l'eau pure, la relation entre la concentration à saturation  $DO_{sat}$  et la température  $T$  est connue et tabulée. Celle-ci vaut, avec  $DO$  en  $\mu\text{g.L}^{-1}$  :

$$DO_{sat} = 3,869 + 10,65 \times \exp(-T/27,99) \quad (6)$$

Pour une température fixée  $T_1$ , la relation  $DO = f(U)$  est une droite passant par 0 et par le point  $(U_{sat}, DO_{sat})$  (saturation). Il suffit donc de se placer expérimentalement dans cette condition pour mesurer  $U_{sat}$ , calculer  $DO_{sat}$  à partir de la formule (6), et en déduire l'équation de la droite  $DO = f(U)$ .

Cependant, cette droite n'a d'intérêt qu'à  $T_1$ , et la saturation en oxygène dissous dépend exponentiellement de la température. Il faut donc effectuer une mesure de  $U_{sat}$  pour un second point de température  $T_2$  différent de  $T_1$ . Ces deux points sont suffisants pour obtenir ainsi la relation affine  $U_{sat} = f(T)$ .

Nous avons utilisé deux béchers d'eau distillée dans un bain de glace et un bain marie (40°C) respectivement, un thermomètre commercial et la sonde de mesure remplie d'une solution de NaOH à 0,5 mol.L<sup>-1</sup>. Pour saturer l'eau en oxygène, nous avons placé les béchers sur un agitateur magnétique pendant 10 mn. Nous avons ensuite effectué la mesure, et avons obtenu les résultats suivants :

Température (°C)	13,6	30,4
Sonde SEN0237 (mV)	1000	1440

Ces deux points forment la droite d'équation  $U_{sat} = 26,2 \times T + 643$  (avec  $U_{sat}$  en mV).

Enfin, comme la relation  $DO = f(U)$  est une droite linéaire, passant par 0 et par le point  $(U_{sat}, DO_{sat})$ , sa pente vaut  $a = DO_{sat}/U_{sat}$ . Son équation est donc :

$$DO = \frac{DO_{sat}}{U_{sat}} \times U = \frac{3,869 + 10,65 \times \exp(-T/27,99)}{26,2 \times T + 643} \times U \quad (7)$$

Il s'agit de la fonction reliant la concentration en oxygène dissous  $DO$  à la tension  $U$  aux bornes de la sonde. Cette fonction sera utilisée dans le code Arduino final pour que l'on mesure directement la concentration en oxygène dissous (en  $\mu\text{g.L}^{-1}$ ).

## 4 Branchement et interfaçage des capteurs

### 4.1 Schéma général

Une fois les capteurs étalonnés, ils peuvent être utilisés avec la sonde. Plusieurs éléments doivent être branchés : les capteurs, la batterie, le panneau solaire et la carte bq24074, l'enveloppe SD et l'antenne 4G. L'enveloppe SD est prévue pour être simplement emboîtée dans la carte Arduino, et occupe le port 28. Le panneau solaire et la batterie sont branchés à la carte bq24074, elle-même connectée à la carte Arduino. L'antenne est branchée sur un le port correspondant.

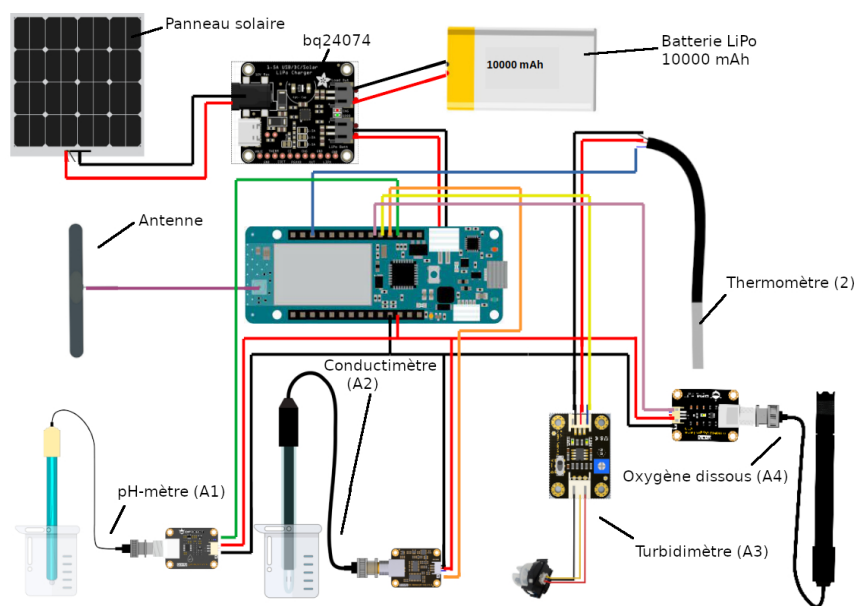


FIGURE 8 – Schéma de branchement des différents capteurs à la carte Arduino. En rouge, l'alimentation (3,3 V). En noir, la masse. Le fil bleu est branché sur le pin 2, les fils vert, orange, jaune et violet sont branchés sur les pins A1, A2, A3 et A4.

Le capteur de température est un capteur digital. On branche l'alimentation sur Vcc (3,3 V) et on relie la masse (GND). Il faut ensuite brancher le fil correspondant au signal sur un port digital, ici le port 2. Les autres capteurs sont connectés à une carte électronique, fournie avec chaque capteur, via un câble BNC. La carte électronique est ensuite connectée à l'alimentation (Vcc, ici 3,3 V), la masse (GND) et une entrée analogique. Nous avons utilisé les ports A1, A2, A3 et A4 respectivement pour les capteurs de pH, conductimétrie, turbidité et oxygène dissous.

## 4.2 Code Arduino

Chaque capteur est commandé par la carte Arduino via une fonction définie de manière similaire, appelée dans la partie loop() du code à chaque fois que l'on souhaite faire une mesure. Les ports sur lesquels sont branchés les capteurs sont d'abord déclarés en en-tête, et la sonde de thermométrie initialisée. Le code correspondant au thermomètre est complexe, et nous l'avons copié et adapté d'après la documentation. Pour les autres capteurs, le code suit ce modèle, ici pour le capteur de pH :

```

1 float getpH(){
  unsigned long int avg_pH; // definition valeur moyenne de la mesure
3  int buf[10],k;
  for(int i=0;i<10;i++){ // fait 10 mesures afin d'obtenir une valeur moyenne
5    buf[i]=analogRead(pin_pH); // lit la tension sur le port analogique
    delay(10);}
7  for(int i=0;i<9;i++){ // classe les valeurs par ordre croissant
    for(int j=i+1;j<10;j++){
9      if(buf[i]>buf[j]){
        k=buf[i];
11       buf[i]=buf[j];
        buf[j]=k;}}
13  avg_pH=0;
  for(int i=2;i<8;i++) // calcule la valeur moyenne
15    avg_pH+=buf[i];
  float V = avg_pH * VREF / 1024.0 / 6.0; // convertit la en mesure volts
17  float pH = 0.00247 * V + 4.02; // convertit la tension en pH (etalonnage)
  return pH; // retourne la valeur calculee
19 }
```

Pour adapter ce code, il suffit de changer la fonction d'étalonnage par celle correspondant au capteur. Pour les sondes de conductimétrie et d'oxygène dissous, la mesure dépend de  $T$ , et il faut donc que la fonction prenne  $T$  comme argument. Pour la sonde de conductimétrie, il faut également rajouter une ligne correspondant à la compensation en température. Les différentes fonctions sont détaillées dans le code final complet téléchargeable sur notre site dans la section Ressources (trop long pour ce rapport) en suivant [ce lien](#).

## 5 Acquisition des données, serveur Web et mise en sommeil

### 5.1 Sauvegarde des données sur carte SD

Les données mesurées sont sauvegardées sur une carte SD de 8 Go. Cela permet d'avoir une copie des données au cas où la connexion 4G serait perdue. Les données sont sauvegardées au format csv, pour être traitées lorsque l'on récupérera la sonde avec Python ou un tableur (Excel ou Calc par exemple). Nous avons rédigé le code suivant pour tester l'enregistrement SD avec un capteur de température et d'humidité (DHT22).

```

1 #include <SPI.h> // librairies SD et DHT22
  #include <SD.h>
```



```

3 #include <DHT.h>

5 const int chipSelect = 4; // pin carte SD (pour le NB1500)
  #define DHTPIN 7 // DHT22 branche sur le pin 7
7 #define DHTTYPE DHT22 // definition du capteur DHT22
  DHT dht(DHTPIN, DHTTYPE);
9
11 void setup() {
  SD.begin(chipSelect); // initialisation communication SD
  dht.begin(); // initialisation DHT22
13 File dataFile = SD.open("datalog.txt", FILE_WRITE); // ouverture fichier
  String titleString = ""; // creation de l'en-tete
15 titleString = String("Temperature (C)")+"\t"+String("Humidity(%)"); // en-tete
  dataFile.println(titleString); // ecriture dans le fichier
17 dataFile.close(); // fermeture du fichier
  }
19 void loop() {
  float Temp = dht.readTemperature(); // mesure de T
21 float Humidity = dht.readHumidity(); // mesure de H
  String dataString = ""; // definition de la ligne de texte
23 dataString += String(Temp)+"\t"+String(Humidity); // ajout mesures au texte
  File dataFile = SD.open("datalog.txt", FILE_WRITE); // ouverture du fichier
25 dataFile.println(dataString); // ecriture dans le fichier
  dataFile.close(); // fermeture du fichier
27 }

```

## 5.2 Envoi des données sur un serveur Web

Pour envoyer des données sur le Web, il faut d'abord configurer un serveur. Nous avons choisi ThingSpeak, gratuit et simple d'utilisation. Il faut créer un compte, puis ouvrir un canal ("channel" en anglais). Ce canal comporte plusieurs champs de données, nous en avons définis 5, pour chaque paramètre que l'on mesure. Le canal est associé à un numéro unique d'identification, et une clé secrète pour écrire des informations. Après la sauvegarde sur carte SD, Arduino se connecte au réseau 4G. Les données sont envoyées sur le serveur ThingSpeak, en précisant le numéro de canal, la clé secrète d'écriture et les champs concernés par l'envoi de données. On ferme ensuite la connexion 4G. Les utilisateurs peuvent se connecter gratuitement au serveur et accéder à ces données, en temps réel. Elles sont affichées sous forme de graphe, et il est possible de télécharger le fichier csv correspondant à chaque champs de mesure. Nous avons testé l'envoi de données sur ThingSpeak par 4G avec une simple mesure de température et d'humidité dans la salle, avec le code suivant :

```

1 #include <MKRNB.h> // librairies NB1500, ThingSpeak et DHT22
  #include <ThingSpeak.h>
3 #include <DHT.h>

5 const char PINNUMBER[] = "0000"; // code PIN SIM
  const char APN[] = "TM"; // APN fournisseur d'accès
7 NBClient client; // initialisation des librairies NB1500
  GPRS gprs;
9 NB nbAccess;
  unsigned long myChannelNumber = 0000000; // canal ThingSpeak
11 const char* myWriteAPIKey = "XXXX0000"; // cle secreta ThingSpeak
  #define DHTPIN 7 // DHT22 branche sur le pin 7
13 #define DHTTYPE DHT22 // definition du capteur DHT22
  DHT dht(DHTPIN, DHTTYPE);
15
17 void setup() {
  dht.begin(); // initialisation DHT22

```

```

}
19 void loop() {
    float Temp = dht.readTemperature(); // mesure de T
21    float Humidity = dht.readHumidity(); // mesure de H
    boolean connected = false; // etat de la connexion
23    while (!connected) { // boucle while pour forcer la connexion
        if ((nbAccess.begin(PINNUMBER, APN) == NB_READY) && // allume modem et SIM
25            (gprs.attachGPRS() == GPRS_READY)) { // trouve la connexion 4G
            connected = true; }
27        else {
            delay(1000); } }
29    ThingSpeak.begin(client); // initialise ThingSpeak
    delay(1000);
31    ThingSpeak.setField(1, Temp); // lie les mesures aux champs ThingSpeak
    ThingSpeak.setField(2, Humidity);
33    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey); // ecriture canal TS
    delay(240000); // delai d'ecriture du modem
35    nbAccess.shutdown(); // extinction du modem
    delay(240000); // delai d'extinction
37 }

```

La connexion au modem et à ThingSpeak a été mise dans la partie `loop()`, avec à chaque fois l’extinction du modem, en prévision du fait que l’on ne fera des mesures que toutes les 3h voire plus. Il faut donc fermer la connexion 4G pour économiser de l’énergie.

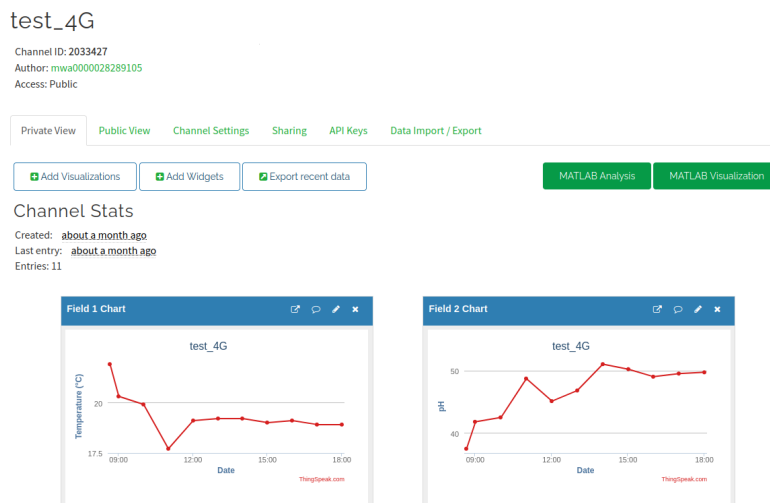


FIGURE 9 – Capture d’écran de notre canal ThingSpeak “test” avec deux champs pour la température et l’humidité. L’accès en ligne à ces graphes est libre et ne nécessite aucune inscription. L’onglet “Data Import/Export” permet de télécharger toutes les données au format csv.

L’interface de ThingSpeak se présente sous la forme de graphiques, avec en ordonnée la grandeur mesurée et en abscisse le temps. Ces graphiques sont mis à jour dès que le serveur reçoit de nouvelles mesures. On peut également choisir de télécharger les données au format csv. Nous avons configuré le canal pour qu’il soit public : nos données pourront être vues par n’importe quel utilisateur librement. Ci-dessus sont affichés les deux canaux correspondant aux mesures de température et d’humidité faites dans la salle pour tester le code.

### 5.3 Mise en sommeil

Une fois l’enregistrement sur carte SD fait et les données envoyées sur le serveur Web, nous avons prévu de mettre Arduino en mode “veille” afin d’économiser de l’énergie. L’Arduino NB1500 possède

une horloge interne. Il suffit alors dans le code d'introduire une alarme, afin de "réveiller" la sonde périodiquement. En veille, le microcontrôleur ne consomme plus que quelques mA, contre 100 à 200 mA en fonctionnement normal. Le code Arduino correspondant à cette fonction est le suivant :

```

1 #include <DHT.h> // librairies horloge et et DHT22
#include <RTCZero.h>
3
#define DHTPIN 7 // DHT22 branche sur le pin 7
5 #define DHTTYPE DHT22 // definition du capteur DHT22
DHT dht(DHTPIN, DHTTYPE);
7 RTCZero rtc; // initialisation de la librairie de l'horloge
bool matched = true; // booleen condition alarme
9
void setup() {
11   Serial.begin(9600); // frequence communication serie
   while (!Serial) {
13     ;} // attente connexion port serie
   dht.begin(); // initialisation DHT22
15   rtc.begin(); // initialisation horloge
}
17 void loop() {
   if (matched){ // condition : si booleen "matched" vrai
19     matched = false; // booleen devient faux
     float Temp = dht.readTemperature(); // mesure de T
21     float Humidity = dht.readHumidity(); // mesure de H
     Serial.print(Temp); // affiche T sur l'ecran
23     Serial.print("_");
     Serial.println(Humidity); // affiche H sur l'ecran
25     int alarmMinutes = rtc.getMinutes(); // enregistre l'heure (minutes)
     int alarmHours = rtc.getHours(); // enregistre l'heure (heures)
27     alarmMinutes += 0; // réglage de la prochaine alarme (intervalle)
     alarmHours += 1;
29     if (alarmMinutes > 59) { // boucle empechant les minutes de dépasser 60
       alarmMinutes -= 60;
31       alarmHours += 1;}
     if(alarmHours > 23) { // boucle empechant les heures de dépasser 24
33       alarmHours -= 24;}
     rtc.setAlarmTime(alarmHours, alarmMinutes, 00); // regle la prochaine alarme
35     rtc.enableAlarm(rtc.MATCH_HHMMSS); // active l'alarme
     rtc.attachInterrupt(alarmMatch); // alarme = change booleen "matched"
37     rtc.standbyMode();} // mise en veille
}
39 void alarmMatch(){ // fonction changeant "matched" en "true"
   matched=true;
41 }

```

Ce code permet de mesurer la température et l'humidité, de les afficher à l'écran, puis de mettre en veille Arduino pendant un certain intervalle de temps (ici 1h).

La mise en commun de ces trois codes permet donc l'enregistrement des données sur carte SD, leur envoi sur le serveur ThingSpeak et la mise en veille de la sonde. Le code final complet est téléchargeable sur notre site Web dans la section Ressources en suivant [ce lien](#).

## 6 Construction de la sonde

La colonne de mesure a été fabriquée à partir d'un tube PVC de diamètre 10 cm. Pour déterminer sa longueur, que nous voulions suffisante pour amener les capteurs en profondeur, nous avons du prendre

en compte la taille des câbles des capteurs (nous ne voulions pas avoir à allonger les câbles dans la mesure du possible). Les longueurs sont les suivantes :

Sonde	Longueur de câble (cm)	Diamètre (mm)
Capteur température	95	7
Capteur pH	525	28
Capteur conductimétrie	74	12
Capteur turbidité (mV)	30	28
Capteur O <sub>2</sub> dissous	208	12

Le capteur turbidité possède le câble le plus court, 30 cm, mais il est facile de l'allonger (3 fils simples). Le capteur conductimétrie possède un câble de 74 cm, mais il s'agit d'un câble type BNC, plus compliqué à rallonger : nous avons donc décidé de nous baser sur sa longueur pour déterminer la taille de la colonne de mesure. Celle-ci fait donc 70 cm, de sorte à pouvoir tirer le câble du capteur conductimétrie et le brancher facilement à Arduino.

Les extrémités de la colonne sont bouchées par des tampons de visite PVC. Afin de laisser passer les capteurs hors de la sonde pour être en contact avec l'eau de l'étang, nous avons percé le tampon inférieur à la taille des différents capteurs (voir tableau ci-dessus). Puis, pour assurer l'étanchéité une fois les capteurs passés, nous avons recouvert le tampon de mastic polymère (TEC7). Ce mastic résiste à l'eau salée, et a une bonne adhérence sur le PVC. Un joint circulaire plastique assure l'étanchéité entre le pas de vis du tampon et la colonne.



FIGURE 10 – Gauche : Le tampon inférieur percé pour accueillir les capteurs. Droite : Le tampon inférieur avec les capteurs en place, et du mastic TEC7 pour assurer l'étanchéité.

Le tampon supérieur est collé au panneau solaire. Nous avons découpé une plaque PVC rectangulaire de la même dimension que le panneau, et nous l'avons fixée dessus. Nous avons ensuite percé le tampon pour laisser passer le câble du panneau solaire, et étanchéifié l'ensemble avec le mastic polymère. Nous avons enfin allongé le câble, pour que l'on puisse visser le panneau-bouchon sans qu'il ne se tende.

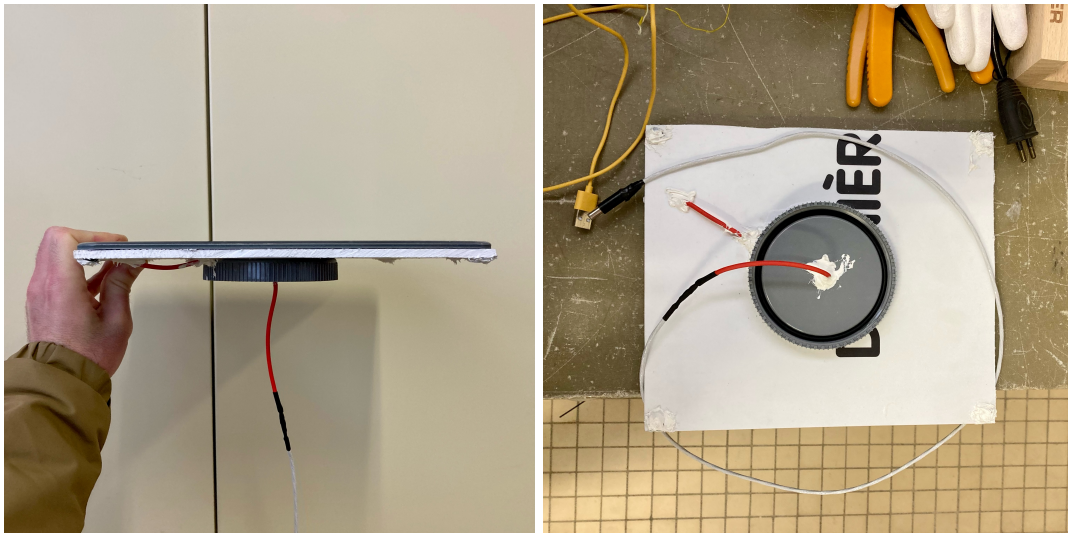


FIGURE 11 – Gauche : Vue de côté du tampon supérieur. Le panneau est vissé à une plaque PVC, elle-même collée au tampon. Droite : Vue de dessous du tampon supérieur. Le fil traverse le tampon et du mastic assure l'étanchéité.

Pour fabriquer un support adapté aux différentes cartes électroniques et à Arduino, nous avons découpé une plaque PVC 5 mm en un disque de diamètre 8 cm et deux rectangles sur lesquels viennent se fixer les cartes et Arduino. La batterie vient se loger dans le dos du support. Le disque est percé pour laisser passer les embouts mâles des cartes électroniques.

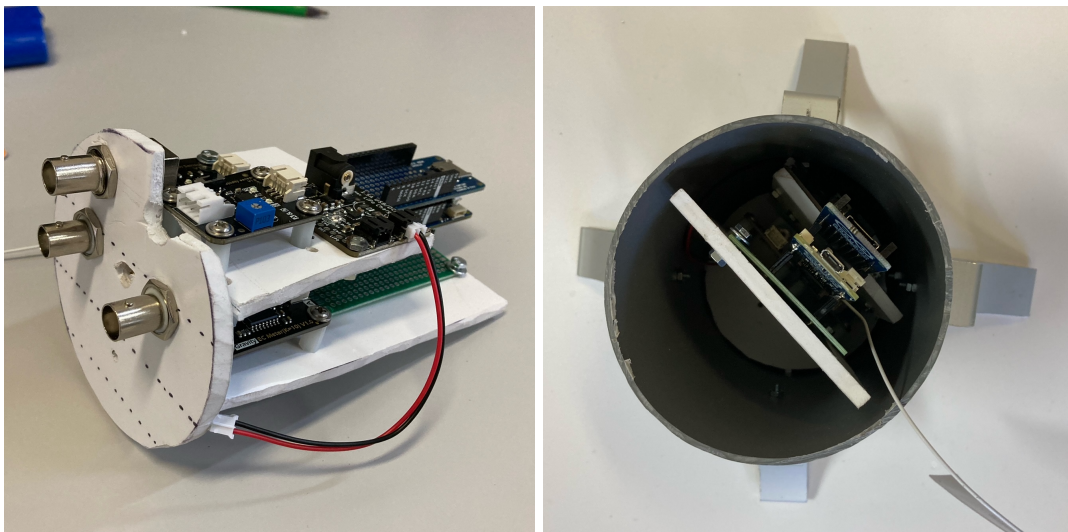


FIGURE 12 – Gauche : Le support PVC monté avec les cartes électroniques et la carte Arduino fixées dessus. Droite : Le support PVC glissé dans la colonne de mesure.

Arduino vient se brancher sur la plaque de prototypage. Nous ne l'avons pas soudé, pour éviter de chauffer ses composants, et pouvoir le retirer facilement à la fin du projet. Les câbles reliant les cartes électroniques à Arduino ont été soudés sur la borne GND (la masse) et Vcc (3,3 V), et sur leurs pins respectifs (A0, A1, A2, A3 et 2). Une fois le support monté et les différents composants en place, il suffit alors de brancher les câbles des sondes sur les embouts mâles et de glisser le support dans la colonne de mesure.

Pour lier la colonne de mesure à sa bouée, nous avons utilisé une plaque extrudé de côté 50 cm percée au milieu par un trou de 10 cm, permettant d'accueillir la colonne de mesure. Pour attacher la colonne à la plaque, nous avons recyclé 4 équerres en aluminium que nous avons percées et taraudées. Deux vis

maintiennent chaque équerre en place sur la plaque, et deux autres vis maintiennent la colonne. Les différentes interfaces (vis / équerre / colonne) ont été enduites de mastic TEC7 étanche, et l'intérieur de la colonne de peinture PVC, pour éviter toute entrée d'eau.



FIGURE 13 – Gauche : Détail lors du perçage de la plateforme pour fixer les équerres. Droite : La colonne de mesure fixée à la plateforme avec les 4 équerres vissées.

Afin de maintenir la colonne hors de l'eau, nous avons fabriqué une bouée à partir de 4 tubes PVC de 10 cm de diamètre et longs de 35 cm, avec des coudes PVC droits. Cela forme un carré de 50 cm de côté. Nous avons collé l'ensemble avec de la colle PVC, qui "soude" les pièces PVC très rapidement : il faut faire attention lors de l'assemblage car en quelques secondes c'est collé ! Pour étanchéifier la bouée, nous avons fait des joints en mastic polymère à chaque interface tube/coude.

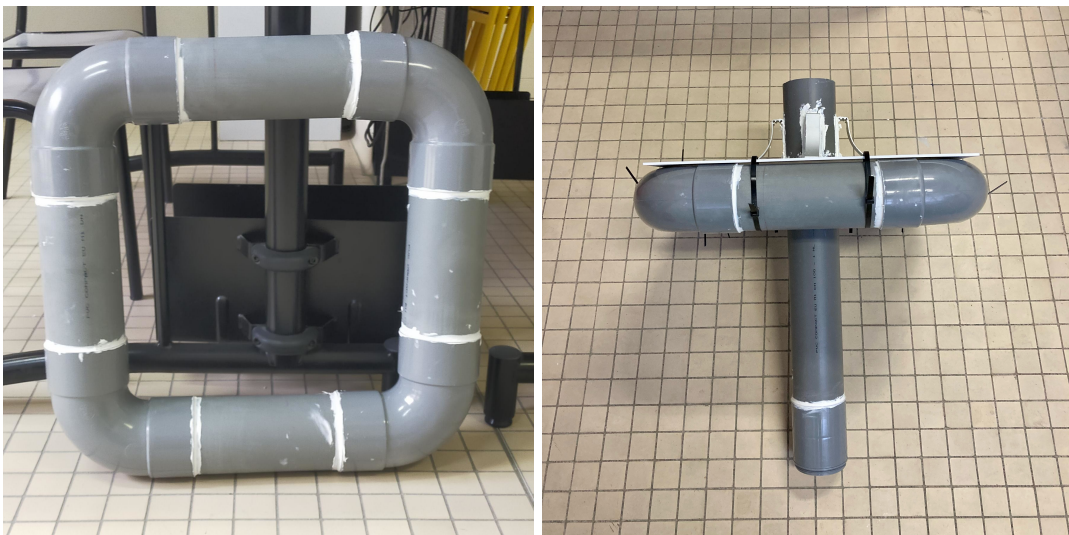


FIGURE 14 – Gauche : La bouée vue du dessus, composée de 4 tuyaux PVC de 35 cm (diamètre 100 mm) et de 4 coudes droits, collés à la colle PVC. Du mastic TEC7 assure l'étanchéité. Droite : L'ensemble colonne + sonde lié avec des serflex.

Nous avons percé des trous à 2 cm des bords de la plaque, afin de laisser passer des serflex pour attacher la plaque à la bouée de manière simple. L'espace libre sur la plaque est prévu pour accueillir d'autres capteurs dans des versions améliorées de la sonde SEB.

## 7 Tests d'étanchéité et résultats préliminaires

Nous avons effectué les premiers tests d'étanchéité, ainsi que de l'électronique complète, au port de plaisance de Saint-Chamas. Nous avons dans un premier temps testé la flottabilité de la sonde. Bonne nouvelle, elle flotte ! Toutefois, vu le volume d'air que la colonne de mesure représente, la sonde a tendance à se retourner facilement.

La partie immergée de la colonne est longue d'environ 50 cm, avec un diamètre de 10 cm. Elle déplace donc un volume d'eau de  $5 \times \pi 0,5^2 \approx 4 \text{ dm}^3$ . La densité de l'eau salée étant de 1,03 environ, nous avons lesté la colonne avec 4 kg de graviers pour compenser la poussée d'Archimède. La sonde reste alors bien en place.



FIGURE 15 – Gauche : Test d'étanchéité et de flottabilité de la sonde. Droite : La sonde SEB complète, prête à être mise à l'eau !

Concernant l'étanchéité, nous avons détecté une petite fuite, a priori à l'interface entre le tampon et un capteur (et pas dans le pas de vis du tampon). Nous pensons que le mastic, très visqueux, ne s'est pas correctement étalé entre les capteurs et des bulles d'air ont pu se former et créer un passage pour l'eau. Nous allons donc appliquer une surcouche d'un composé du style colle epoxy bi-composant, très liquide lors de l'application, et très dure après séchage, afin d'avoir une surface bien plus homogène et combler les éventuelles fuites.

Le code Arduino complet fonctionne. Nous avons réussi à enregistrer nos données sur carte SD et à les envoyer sur le serveur ThingSpeak. Toutefois, lorsque le conductimètre et le pH-mètre sont utilisés en même temps, les mesures semblent être perturbées. En n'utilisant qu'un capteur, les résultats obtenus sont parfaitement cohérents avec les mesures attendues. Le tableau ci-dessous montre nos premières mesures obtenues dans l'étang. Ces grandeurs sont cohérentes avec ce que l'on a mesuré directement (température et pH) et ce que l'on attend pour de l'eau salée légèrement trouble, mais il faudra davantage de mesures pour en tirer une vraie interprétation.

Grandeur	Mesure SEB	Valeur théorique
Température (°C)	16,4	16,8 (sonde calibrée)
pH	8,2	8,4 (sonde calibrée)
Conductivité (mS.cm <sup>-1</sup> )	32	entre 0 (eau douce) et 56 (eau de mer)
Turbidité (NTU)	25	30 (eau légèrement trouble)
O <sub>2</sub> dissous (mg.L <sup>-1</sup> )	6,2	7,5 (saturation pour de l'eau salée)

Lorsque les capteurs sont alimentés ensemble, le pH tombe à 4 environ, la conductivité monte à 80 mS.cm<sup>-1</sup> et la concentration en O<sub>2</sub> dissous dévie un peu. Une hypothèse pour expliquer ce problème est que ces sondes interfèrent entre elles, car elles partagent la même alimentation et fonctionnent donc simultanément. Plongées ensemble dans l'eau, les courants électriques qu'elles induisent viennent modifier les différences de potentiel qu'elles mesurent. La solution la plus évidente serait de les éloigner suffisamment, mais cela n'est pas possible ici. Une autre solution serait donc d'isoler ces deux sondes avec un sous-circuit comportant un transistor ou un MOSFET, commandé par un pin digital, ou une alimentation "personnelle" (un Arduino Nano par exemple) communiquant avec l'Arduino principal. Ainsi, les alimentations de ces deux sondes pourraient être allumées tour à tour et bien que plongées ensemble dans l'eau les mesures ne seraient pas impactées. Cette solution permettrait aussi de couper complètement la consommation en énergie des capteurs lorsque la sonde est mise en veille. Toutefois, cela rend notre circuit électronique plus complexe, et plus volumineux.

## 8 Communication autour du projet SEB

Pour communiquer autour de notre projet, nous avons mis en place plusieurs initiatives :

- **Un site Web avec un blog et des ressources en ligne**

Chaque séance, nous avons fait un petit résumé du travail accompli, et pris des photos. Tout cela était ensuite mis en ligne sur le blog. Pour les ressources, nous les avons mises en ligne tardivement : il s'agit de nos codes Arduino, de la liste du matériel, de notre rapport et de notre vidéo... L'idée est que d'autres personnes puissent utiliser ce que nous avons fait pour s'en inspirer et développer des projet similaires facilement. Il faut suivre [ce lien](#) pour y accéder et consulter notre blog et nos ressources (dont le code Arduino complet!).

- **Un stand au mois de mai au lycée Cocteau à l'occasion de la Semaine Verte**

Chaque année le lycée organise une semaine autour de l'environnement, et fait venir des associations etc. Cette année, nous tiendrons un stand, pour présenter notre travail et la sonde SEB, la problématique de l'Étang de Berre... Et recruter des élèves pour l'an prochain!

- **Une participation à une exposition impliquant plusieurs associations**

À la fin de l'année, notre travail sera présenté au cours d'une journée de conférences à Berre l'Étang réunissant les acteurs locaux pour la préservation de l'environnement (dont 8 Vies pour la planète et le GIPREB avec qui nous avons collaboré). Nous imprimerons des photos du projet et un poster pour expliquer notre démarche et montrer nos premiers résultats.

- **Une conférence auprès d'un atelier scientifique d'un autre lycée**

Notre atelier était jumelé avec celui du lycée Étienne Gautier Ressins, à Nandax. Nous allons donc leur présenter notre projet, nos résultats, et vice-versa!



## 9 Conclusion et perspectives

S'il reste encore un peu de travail pour que la sonde SEB soit opérationnelle, elle devrait être mise à l'eau dans les mois qui viennent, comme prévu initialement. La zone prévue de la Poudrerie a été validée par le GIPREB : zone protégée d'intérêt, proche de la centrale EDF, avec peu de profondeur (et donc moins de vagues).

Depuis le lancement du projet mi-octobre, nous avons terminé la fabrication de notre sonde sous-marine autonome. Un petit problème d'étanchéité subsiste, mais nous avons bon espoir de le régler rapidement avec une surcouche étanche et homogène au niveau du passage des capteurs. Le code Arduino a été validé, et la sonde est capable d'envoyer des mesures en 4G sur un serveur que nous avons configuré pour un suivi en temps réel, et de les enregistrer sur une carte SD. Si les capteurs ont été correctement calibrés (et l'étalonnage semble durer dans le temps), la sonde de pH et celle de conductimétrie se perturbent mutuellement lorsqu'elles sont mises toutes les deux dans l'eau et alimentées simultanément. Pour résoudre ce problème, la solution idéale semble être l'isolation des alimentations de ces capteurs, pour qu'ils ne fonctionnent qu'alternativement. Cela devrait également nous apporter un gain en terme d'économie d'énergie lorsque la sonde est en veille.

L'organisation de la mise à l'eau est en bonne voie, puisque nous avons obtenu toutes les autorisations nécessaires pour la placer en bordure de la zone protégée du parc de la Poudrerie au nord de l'Étang, et nous nous sommes arrangés avec la base nautique de Saint-Chamas pour la location d'un bateau. Nous espérons une mise à l'eau début mai, pour pouvoir laisser la sonde pendant 1 mois dans l'étang et la récupérer avant la fin de l'année scolaire.

Une fois la sonde en place, il faudra étudier les données fournies par EDF ainsi que les données de météo afin de corrélérer nos mesures avec ces informations, pour détecter une éventuelle influence de ces divers paramètres sur la pollution de l'étang (en particulier les rejets de la centrale). Une seconde sonde placée à un emplacement témoin proche de la centrale aurait été idéale, afin de comparer les mesures des deux sondes, mais le groupe chargé de sa conception n'a finalement plus été en mesure de venir à l'atelier, et nous nous sommes donc concentrés sur cette sonde uniquement.

La fabrication d'une deuxième sonde pourrait être la prochaine étape du projet SEB, elle servirait alors de témoin dans notre expérience. La plateforme qu'offre la bouée devrait aussi permettre d'embarquer de nouveaux capteurs, par exemple des capteurs météo, pour ne pas avoir à utiliser les données MétéoFrance dans notre étude. D'autres pistes à explorer pourraient être la conception de petites sondes, plus simples de conception (pas de 4G, pas de capteurs de pH ou d'O<sub>2</sub> dissous...) et moins chères à réaliser afin de cartographier l'étang et l'influence des rejets sur une étendue plus large, ou bien une colonne de mesure plus longue munie de capteurs régulièrement répartis permettant de caractériser le phénomène de stratification. Tout notre travail sur la fabrication et l'étanchéité de la sonde SEB pourrait facilement être transposé et utilisé.

L'atelier scientifique nous aura permis d'acquérir de nouvelles compétences : bricolage varié, soudure, codage (Arduino et LaTeX), électronique, étalonnage des capteurs, création d'un site Web... Les tests au port de Saint-Chamas ont été un moment fort, et nous attendons avec impatience la mise à l'eau définitive. Cette expérience nous a permis d'apprendre à monter un projet du début à la fin : développer l'idée originale, la réaliser, faire face aux problèmes et les résoudre, puis rédiger un rapport afin d'expliquer et partager nos recherches. L'intérêt généré par notre projet, que ce soit au sein du lycée ou à Saint-Chamas, est également très excitant. La mairie, le journal local et plusieurs associations nous ont contacté pour en savoir plus suite à nos tests au port, et nous espérons que la sonde SEB fournira des mesures utiles aux habitants.