

# **Localisation sous-marine**

Système de logging pour déplacement de module  
sous-marin.

**Ali Zoubir**

Rapport de projet



Génie électrique  
École supérieure  
Suisse

26 janvier 2023

## Table des matières

<b>1</b>	<b>Caractéristiques du projet</b>	<b>3</b>
1.1	Description . . . . .	3
1.2	Aperçu . . . . .	3
1.3	Tâches à réaliser . . . . .	4
1.4	Description des blocs . . . . .	5
1.5	Jalons principaux . . . . .	6
1.6	Livrable . . . . .	6
<b>2</b>	<b>Pré-étude</b>	<b>7</b>
2.1	Fonctionnement du système . . . . .	7
2.1.1	Schéma bloc . . . . .	7
2.2	Choix des composants importants . . . . .	9
2.2.1	Senseur absolu . . . . .	9
2.2.2	Capteur de pression . . . . .	11
2.2.3	Affichage . . . . .	11
2.2.4	Carte SD . . . . .	12
2.2.5	Real Time Clock . . . . .	13
2.2.6	Microcontrôleur . . . . .	14
2.2.7	Batterie, charge et régulation . . . . .	15
2.3	Estimation des coûts . . . . .	16
2.4	Conclusion et perspectives . . . . .	17
<b>3</b>	<b>Développement schématique</b>	<b>18</b>
3.1	Choix des composants . . . . .	18
3.1.1	Microcontrôleur . . . . .	18
3.2	Dimensionnements . . . . .	19
3.2.1	Vue d'ensemble schématique . . . . .	19
3.2.2	Autonomie du système . . . . .	20
3.2.3	LED Interface . . . . .	22
3.2.4	Adaptation mécanique . . . . .	23
3.2.5	Bus de communications . . . . .	24
3.2.6	Chargeur de batterie . . . . .	25
3.2.7	Régulateur 3.3V . . . . .	25
3.2.8	Régulateur 5V . . . . .	25



# Localisation sous-marine 2022, V0.0

## 1 Caractéristiques du projet

### 1.1 Description

L'objectif de ce projet, et de stocker des données de mesures du déplacement d'un module sous-marin par une centrale inertielle, dans le but de mathématiquement le localiser depuis son point de départ (référence). Ceci, car la localisation sous-marine n'est pas une tâche aisée due aux différentes contraintes de communication sous-marine notamment que les ondes électromagnétiques ne se propagent pas facilement.

### 1.2 Aperçu

- Sauvegarde d'un set de données chaque 100ms.
- Profondeur d'utilisation maximum, de 60m.
- 2 heures de logging dans carte SD.
- Sensing sur 9 axes :
  - Mesures [Il est souhaitable que les capteurs choisis aient une faible dérive] ;
    - Accéléromètre 3-axes.
    - Gyroscope 3-axes.
    - Magnétomètre 3-axes.
    - Senseur de température
    - Profondimètre [0->10bar] [Res 1/10]
    - 3 à 5 slots libres MikroE pour autres mesures.
- Possibilité de sauvegarder la localisation de points d'intérêt par :
  - Bouton de sauvegarde [A définir : Magnétique, Optique, Mécanique ou autre].
- Batterie, autonomie minimum de 2 heures [ $\sim 10^{\circ}\text{C}$ ].
- Charge de la batterie par connecteur USB.
- (Optionnel) Lecture des données par connecteur USB (Interfaçage électronique, software optionnel dans cette version).
- (Optionnel) Interface LED ou petit écran.

### 1.3 Tâches à réaliser

Développement et intégration d'un PCB avec capteurs et logging sur carte SD dans une lampe de plongée étanche.

Développement schématique

- Fonctionnement MCU.
- Périphériques de mesures et de sauvegarde / Bus de communication.
- Gestion batterie

Routage pour intégration dans boîtier de lampe de plongée 200x45mm.

Programmation mesure et sauvegarde chaque 100ms.

- Configuration MCU.
- Configuration des périphériques de mesure pour 9-DOF.
- Configuration des périphériques de sauvegarde (Carte SD).
- Configuration et communication avec l'interface.
- Communication et traitement des données mesurées.

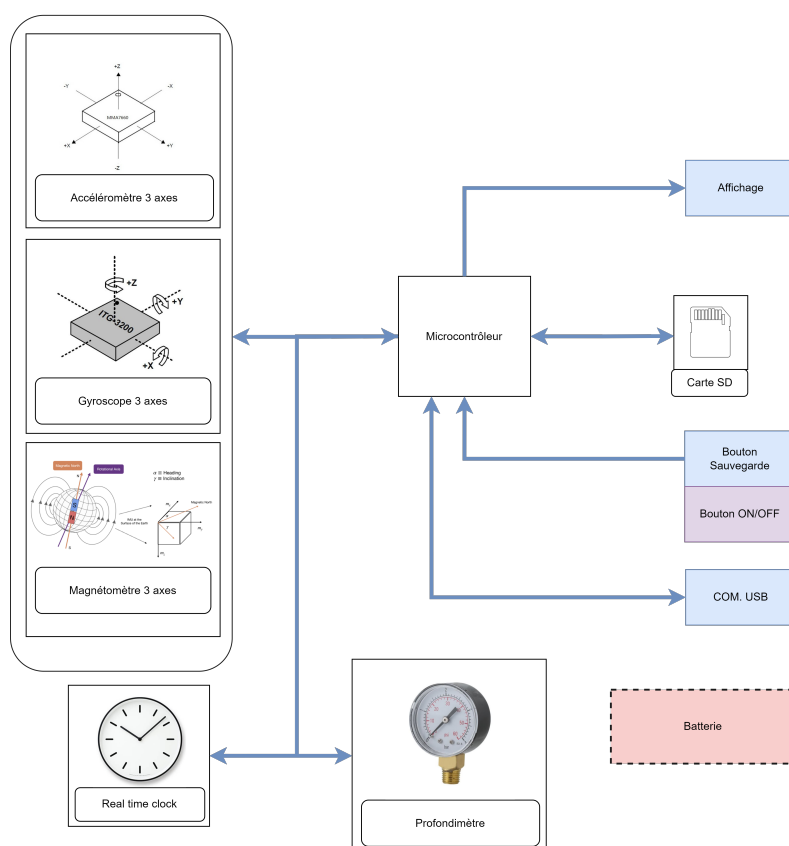


FIGURE 1 – Schéma de principe  
Source : Auteur

## 1.4 Description des blocs

1. **Carte SD :**  
Stockage des données de mesures chaque 100ms, cur du projet.
2. **Accéléromètre-gyroscope-magnétomètre :**  
Lecture des données individuelles brute ainsi que de fusion des capteurs, pour mesurer les déplacements sur 9 degrés de libertés.
3. **Profondimètre :**  
Mesure la pression pour déduire la profondeur, afin de corroborer les autres mesures des capteurs.
4. **Real time clock :**  
Permet de sauvegarder la temporalité du set de mesure dans la carte SD.
5. **Affichage :**  
Affichage LED ou écran, pour affichage pas encore définis (ex. Profondeur, état batterie)
6. **Bouton sauvegarde :**  
Permet la mise en valeur dun set de mesure. La forme de ce bouton nest pas encore définie. Il sera peut-être fusionné avec le bouton ON/OFF.
7. **Bouton ON/OFF :**  
Permet dallumer ou déteindre le système.
8. **Batterie :**  
Batterie du système, technologie à définir dans la pré-étude.
9. **COM. USB :**  
Permet de charger les batteries. Il faudra également prévoir dans cette version linterface électronique pour la lecture de la carte SD directement par le port USB.
10. **Microcontrôleur :**  
Lis et traite les valeurs des capteurs, sauvegarde dans la carte SD...

## 1.5 Jalons principaux

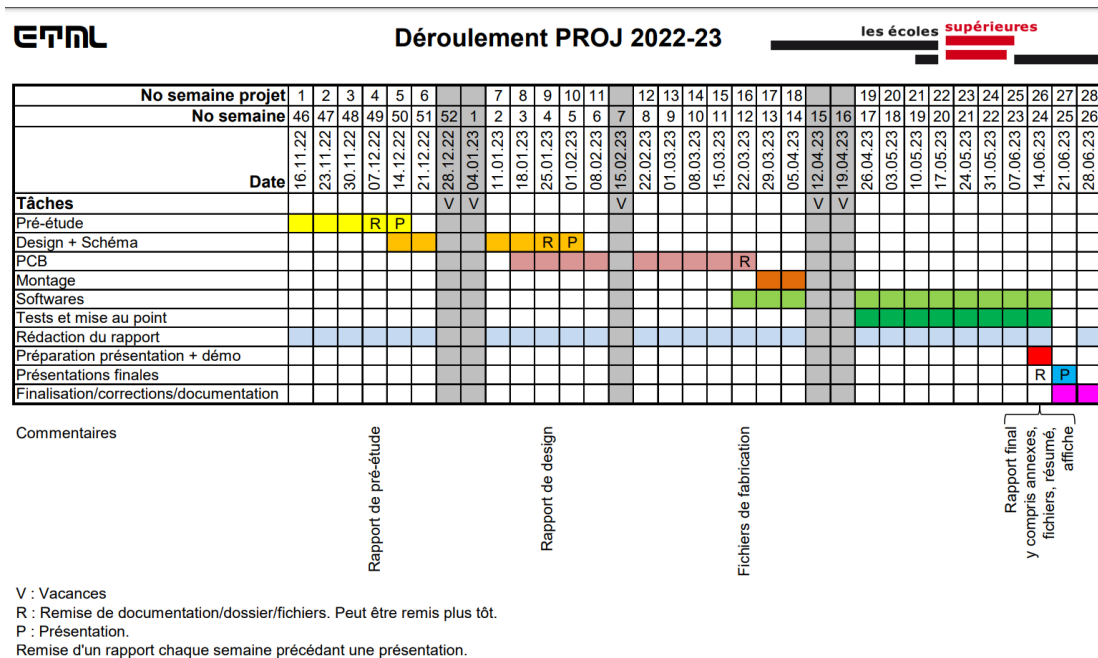


FIGURE 2 – Jalons principaux

## 1.6 Livrable

Les fichiers sources de CAO électronique des PCB réalisés  
 Tout le nécessaire à fabriquer un exemplaire hardware de chaque :  
 fichiers de fabrication (GERBER) / liste de pièces avec références  
 pour commande / implantation  
 Prototype fonctionnel  
 Modifications / dessins mécaniques, etc  
 Les fichiers sources de programmation microcontrôleur (.c / .h)  
 Tout le nécessaire pour programmer les microcontrôleurs (logiciel  
 ou fichier .hex)  
 Un calcul / estimation des coûts  
 Un rapport contenant les calculs - dimensionnement de composants  
 - structogramme, etc.

## 2 Pré-étude

L'objectif de cette pré-étude, est de se pencher sur le fonctionnement plus fondamental du système, faire des petits dimensionnements ainsi que de survoler différents aspects techniques liés au projet.

### 2.1 Fonctionnement du système

#### 2.1.1 Schéma bloc

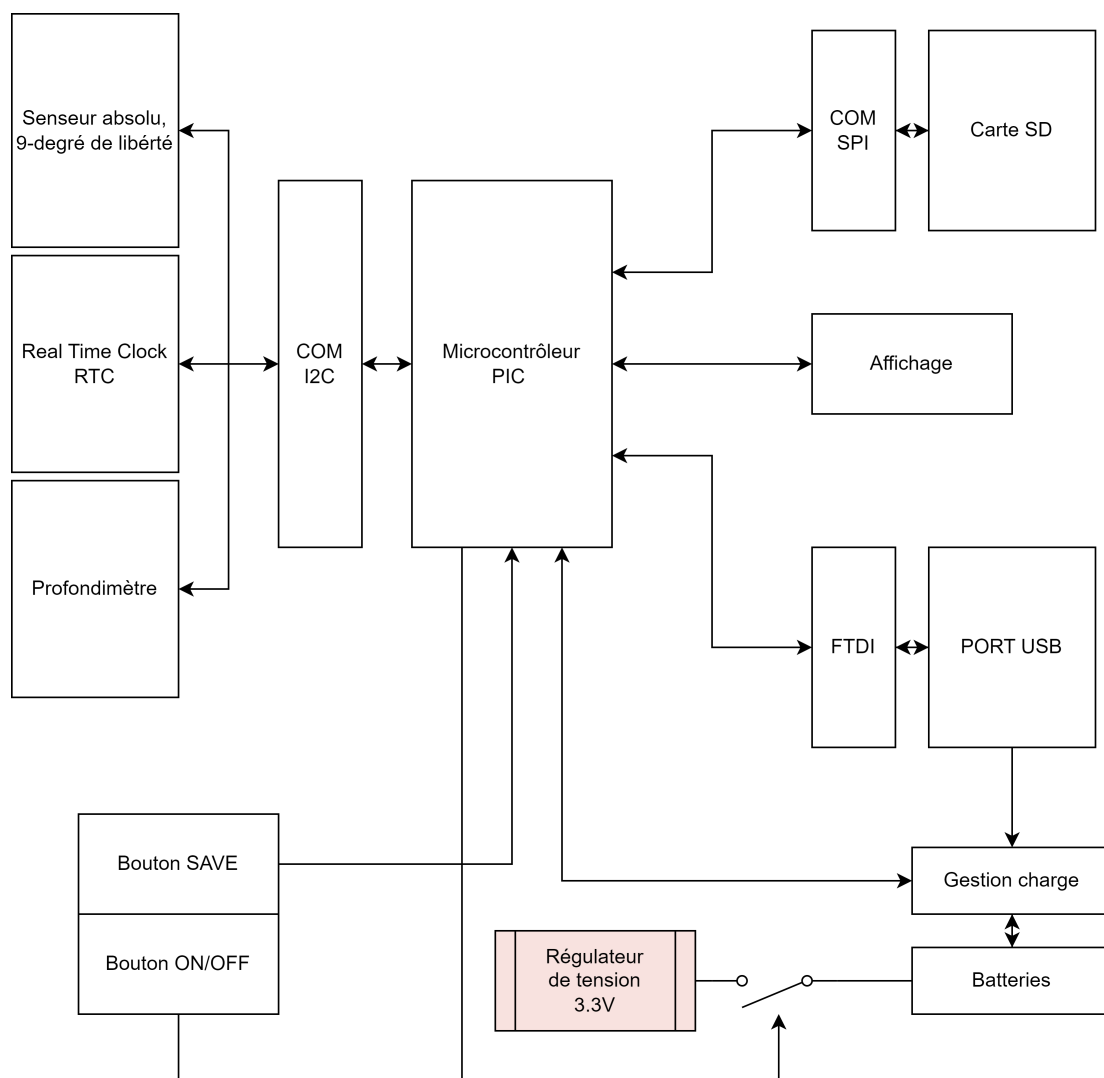


FIGURE 3 – Schéma bloc du module  
Source : Auteur

### Capteurs :

Les différents capteurs sont interfacés sur le même bus, et ont comme master le microcontrôleur en communication bidirectionnel, afin d'à la fois configurer les registres des périphériques et de lire leurs mesures.

### Carte SD :

La carte SD est interfacée en SPI et va contenir les données des différents capteurs ainsi que leurs éventuels flags d'importance (sauvegarde), sa taille sera dimensionnée ultérieurement.

### Port USB & charge :

Un port USB est présent, afin charger les batteries par un IC de gestion de charge connecté directement au 5V. De plus le port USB est communiquant avec le microcontrôleur par un driver FTDI, afin d'éventuellement ajouter un système de lecture de la carte SD, directement par USB. Ceci dans cette version ou une ultérieure. Le port USB pourrait aussi servir à fixer la référence de la RTC.

### Bouton multifonction :

Sachant qu'un bouton étanche est déjà présent sur le module, l'exploiter en tant que bouton multifonction est une solution ergonomique pour ne pas mettre en péril l'étanchéité globale. Ce bouton ferait office de ON/OFF et de "sauvegarde" de point d'intérêt. Pour se faire, le bouton contrôlerait par un transistor de commutation l'alimentation du système, puis lors de l'allumage du microcontrôleur, le MCU prendrait la relève en maintenant le système allumé à son tour, permettant ainsi de lire le bouton et de sur une pression longue déconnecter l'alimentation.

### Affichage :

L'affichage permettra de visualiser différentes données, dont les plus importantes tel que la pression ou le statut de la batterie.

La forme de l'affichage est encore à définir selon la mécanique du module, mais le plus élégant, serait l'utilisation d'un petit écran OLED.

### Capteur de pression :

Le capteur de pression devra avoir un contact direct avec l'eau, cela impliquera de la mécanique et de la gestion d'étanchéité. Une autre possibilité aurait été de mesurer optiquement la déformation du boîtier pour en déduire la pression, mais la complexité est trop importante.



## 2.2 Choix des composants importants

### 2.2.1 Senseur absolu

Pour le senseur absolu, il existe des IC permettant directement de faire la fusion des senseurs (**Accéléromètre, gyroscope, magnétomètre et thermomètre**), ce qui épargne toute une phase de calcul chronophage, en permettant directement de lire les **quaternion, angles de Euler, vecteurs de rotations, cap de direction etc...** directement sur le composant. Il existe différents IC dont deux ce sont montrés très intéressants, le **BNO85** et le **BNO55**, les deux étant PIN-Compatible, j'ai décidé d'opter pour le **BNO055**<sup>1</sup>.

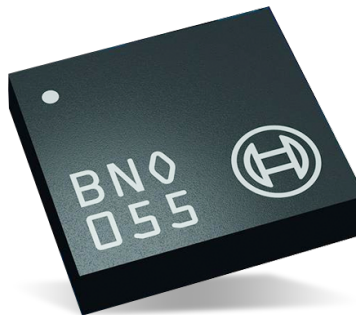


FIGURE 4 – Schéma bloc du module

Source : <https://www.mouser.ch/new/bosch/bosch-bno55-sensor/>

Sachant que la brazure de ce type de boîtier est compliquée et également dans un but de simplification du projet, j'ai décidé d'utiliser les cartes d'évaluation d'adafruit N° : **4646** qui ont des connections bergs ainsi que tous les composants externes passifs déjà montés.

#### Caractéristiques importantes :

Résolution gyroscope	: 16	[bits]
Résolution accéléromètre	: 14	[bits]
Résolution magnétomètre	: ~0.3	[ $\mu$ T]
$I_{DD}$	: 12.3	[mA]
Dérive de température	: $\pm 0.03$	[%/K]
Dérive accéléromètre	: 0.2	[%/V]
Dérive gyroscope	: <0.4	[%/V]

Nous allons par la suite voir sur la figure 5, quelles données du BNO055 sont disponibles ainsi que leurs tailles mémoires.

1. K:/ES/PROJETS/SLO/2221\_LocalisationSousMarine/doc/composants/9DOF-BNO055

Table 3-36: Temperature Data

Parameter	Data type	bytes
TEMP	signed	1

Table 3-34: Gravity Vector Data

Parameter	Data type	bytes
GRV_Data_X	signed	2
GRV_Data_Y	signed	2
GRV_Data_Z	signed	2

Table 3-32: Linear Acceleration Data

Parameter	Data type	bytes
LIA_Data_X	signed	2
LIA_Data_Y	signed	2
LIA_Data_Z	signed	2

Table 3-30: Compensated orientation data in quaternion format

Parameter	Data type	bytes
QUA_Data_w	Signed	2
QUA_Data_x	Signed	2
QUA_Data_y	Signed	2
QUA_Data_z	Signed	2

Table 3-28: Compensated orientation data in Euler angles format

Parameter	Data type	bytes
EUL_Heading	Signed	2
EUL_Roll	Signed	2
EUL_Pitch	Signed	2

Table 3-27: Yaw rate data

Parameter	Data type	bytes
Gyr_Data_X	signed	2
Gyr_Data_Y	signed	2
Gyr_Data_Z	signed	2

Table 3-26: Magnetic field strength data

Parameter	Data type	bytes
Mag_Data_X	signed	2
Mag_Data_Y	signed	2
Mag_Data_Z	signed	2

Table 3-25: Acceleration data

Parameter	Data type	bytes
Accel_Data_X	signed	2
Accel_Data_Y	signed	2

FIGURE 5 – Donnée de sortie de l'IC (43 bytes)

Source : [https://cdn-shop.adafruit.com/datasheets/BST\\_BNO055\\_DS000\\_12.pdf](https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf)

### 2.2.2 Capteur de pression

Pour le capteur de pression, une modification mécanique du boîtier sera très probablement nécessaire. J'ai pu trouver un capteur correspondant aux caractéristiques demandée du projet, celui-ci est plutôt générique et peut communiquer en I2C :

**PTE7300-14AN-1B016BN**



FIGURE 6 – Illustration capteur de pression  
Source : Distrelec, PTE7300-14AN-1B016BN

L'avantage avec le capteur ci-dessus est le système hermétique pour le trou, un autre capteur peut être utilisé lors de l'étude, néanmoins la modification mécanique étant probablement inévitable, le système de vissage de la figure 6 est intéressant.

### 2.2.3 Affichage

Pour l'affichage, je vais essayer d'opter pour un petit afficheur OLED, en gardant la possibilité en cas de complication lors de l'étude, l'utilisation de simples LEDS d'indications.

Il existe plusieurs affichages OLED rond petits formats, sur lesquels je me pencherais plus en détail lors de l'étude.

## 2.2.4 Carte SD

### Taille mémoire :

Afin de dimensionner la taille de stockage de la carte SD, il faut utiliser les différentes caractéristiques du projet. Normalement la taille de la carte SD n'est clairement pas un problème, sachant que seulement du texte est enregistré et que les tailles mémoires disponibles peuvent être très élevées. Néanmoins il est intéressant de faire le dimensionnement pour connaître le minimum, et pour éventuellement adapter le projet avec d'autres systèmes de mémorisation.

Où :

$T_{rec}$	=	7200'000	[ms]	Temps a enregistrer
$T_{ech}$	=	100	[ms]	Temps d'un échantillon
$S_{mes}$	=	43	[bytes]	Taille de toutes les données de mesures
$S_{timestamp}$	=	~23	[bytes]	Taille de l'information de temporalité
$S_{flag}$	=	1	[bytes]	Taille de l'indication d'importance

Nombre de mesure a effectuer :

$$Nb_{mesures} = \frac{T_{rec}}{T_{ech}} \quad (1)$$

D'après (1), nous avons un nombre de mesure de 72'000.

Taille minimum :

$$Taille_{min} = Nb_{mesures} * (S_{mes} + S_{timestamp} + S_{flag}) \quad (2)$$

D'après (2), la taille mémoire minimum doit être de **~5MB**.

Nous pouvons donc constater que pour une utilisation standard de 2h, la mémoire occupée est très faible, d'où l'intérêt de sauvegarder dans la carte SD la date, afin de pouvoir faire plusieurs "expéditions" en "une fois", sans avoir à vider la carte.

### 2.2.5 Real Time Clock

L'objectif de la RTC, est de donner l'information de la temporalité de la mesure (timestamp), afin de lors du traitement des donnée avoir accès à ce paramètre.

Sachant que l'échantillonnage des mesures est de 100ms, la RTC devrait permettre cette résolution. Néanmoins une autre information importante, comme mentionnée lors de la section 2.2.4, est la date de la mesure, afin de permettre plusieurs expéditions par utilisation de la carte.

J'ai donc décidé d'utiliser une RTC pour l'heure grossière de départ (Année, date, heure, minute, seconde) et les compteur du MCU pour faire le delta entre chacune des mesures en ms.

La RTC devra pouvoir tenir le minimum de 2 heure d'utilisation, à cette fin, la batterie LI-ION déjà présente sera suffisante.

La RTC devra avoir une faible consommation, le calendrier ainsi qu'une bonne précision. A cette fin, la RTC **S-35390A-T8T1G** est assez générique et possède une bonne documentation.

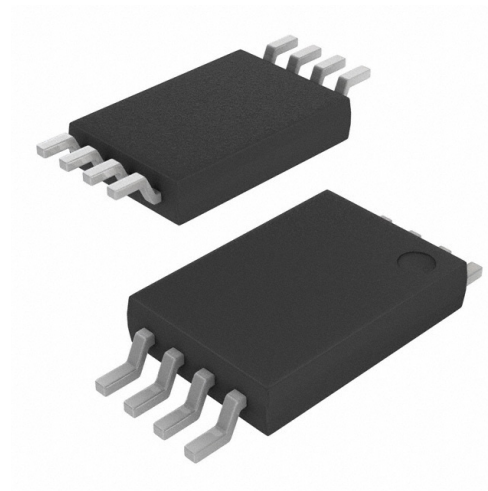


FIGURE 7 – Illustration de la RTC

Source : <https://www.digikey.com/en/products/detail/ablic-inc/S-35390A-T8T1G/1628383>

### 2.2.6 Microcontrôleur

Le microcontrôleur devra avoir un nombre suffisant de communications, sachant que beaucoup sont présentes dans le projet (**I2C, SPI, UART...**), ce qui signifie un nombre de pattes élevées.

Des calculs peuvent aussi être nécessaire, si il s'avère qu'il faille faire un traitement des données préliminaire, il faudrait donc opter pour un MCU 32bits si possible.

La famille PIC est celle standardisée par l'école supérieure, c'est donc pour cette famille-ci que je vais opter.

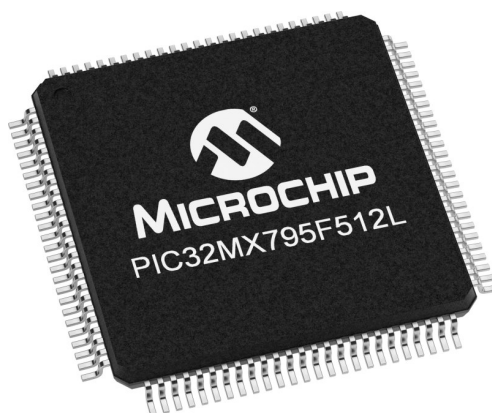


FIGURE 8 – Illustration du modèle MCU du kit ETML-ES  
Source : <https://www.microchip.com/en-us/product/PIC32MX795F512L>

### 2.2.7 Batterie, charge et régulation

Pour la technologie de batterie, en utilisation sous-marine, j'ai trouvé ce tableau de comparaison :

Chemistry	Energy Density (Whr/kg)	Pressure Compensatable (Whr/kg)	Outgassing	Cycles	Comments
Alkaline	140	No	Possible, at higher temperatures	1	Inexpensive, easy to work with
Li Primary	375	No		1	Very high energy density
Lead Acid	31.5	Yes (46)	Yes, even with sealed cells	~100	Well established, easy to work with technology
Ni Cad	33	No	If overcharged	~100	Very flat discharge curves
Ni Zn	58.5	Possibly (160)	None	~500	Emerging Technology
Li Ion	144	No	None	~500	In wide use in small packs
Li Polymer	193	Possibly	None	~500	Only "credit card" form factor currently available
Silver zinc	100	No	Yes	~30	Can handle very high power spikes

FIGURE 9 – Comparaison des technologies de batteries  
Source : Power Systems for Autonomous Underwater Vehicles[1]

Pour des raisons de praticité et étant-donné la documentation plus importante, j'ai décidé d'utiliser la technologie **LI-ION** :

Avantages	Inconvénient
Haute densité d'énergie	Risque d'éclatement
Poids léger	Risque d'enflammement avec l'eau
Haute durée de vie	Sensible a la température
Charge rapide	Décharge complète altérante

Malgré les risque dûs au contact de l'eau (**Enflammement, éclatement...**) la technologie LI-ION est souvent utilisée pour les application sous-marines dû a ses différents avantages, c'est pour cela que j'opterais pour cette technologie.

### 2.3 Estimation des coûts

Ici je vais me baser sur les composants que j'ai pu trouver et estimer le coût moyen de ceux-ci, c'est à titre purement indicatif, (les prix sont généralement estimés à la hausse).

Composant	Estimation
Profondimètre	70.-
Centrale inertielle	35.-
RTC	5.-
Microcontrôleur	5.-
Carte SD	20.-
Affichage OLED	45.-
FTDI	4.-
Batterie LI-ION	20.-
IC chargeur	4.-
Traco-power 3.3V	10.-
PCB	40.-
Total	258.-

L'estimation des prix étant plutôt élevée, des économies peuvent être très facilement réalisées, en changeant l'affichage OLED pour des LEDS ou en modifiant le PCB (Le simplifier ou changer de fournisseur (eurocircuit)).



## 2.4 Conclusion et perspectives

J'ai pu lors de cette pré-étude, établir le fonctionnement global du système, choisir certaines technologies et composants importants, ainsi que pu procéder a certains dimensionnements utiles quant au futur développement.

Par la suite, je vais affiner les différents éléments abordés lors de la pré-étude, effectuer le développement plus détaillé de chacun des blocs et réaliser la schématique du projet.

Lors de la pré-étude, je n'ai pas eu accès au boîtier mécanique du projet, ce qui a restreint mon champs d'action lors de certains dimensionnement, tandis que pendant l'étude j'aurais accès a celui-ci, ce qui risque d'impacter/modifier certains aspect fixés lors des section antérieures.

Je suis très intéressé par le projet et me réjouis grandement de poursuivre son développement.

### 3 Développement schématique

#### 3.1 Choix des composants

##### 3.1.1 Microcontrôleur

Lors de la recherche de composants, j'ai décidé d'utiliser l'un des PIC32 standards de l'ES : **PIC32MX130F064D-I/PT**.

Device	Pins	Program Memory (KB) <sup>(1)</sup>	Data Memory (KB)	Remappable Peripherals					Analog Comparators	USB On-The-Go (OTG)	I <sup>2</sup> C	PMP	DMA Channels (Programmable/Dedicated)	CTMU	10-bit 1 Msps ADC (Channels)	RTCC	I/O Pins	JTAG	Packages
				Remappable Pins	Timers <sup>(2)</sup> /Capture/Compare	UART	SPI/I <sup>2</sup> S	External Interrupts <sup>(3)</sup>											
PIC32MX130F064D	44	64+3	16	32	5/5/5	2	2	5	3	N	2	Y	4/0	Y	13	Y	35	Y	VTLA, TQFP, QFN

FIGURE 10 – Périphériques disponibles du PIC  
Source : PIC32MM0256GPM064 family datasheet

Nous pouvons constater sur la figure 10 que les critères minimaux de mon projet sont respectés :

1 - I2C    1 - SPI    1 - UART    1 - RTCC

3.2 Dimensionnements

3.2.1 Vue d'ensemble schématique

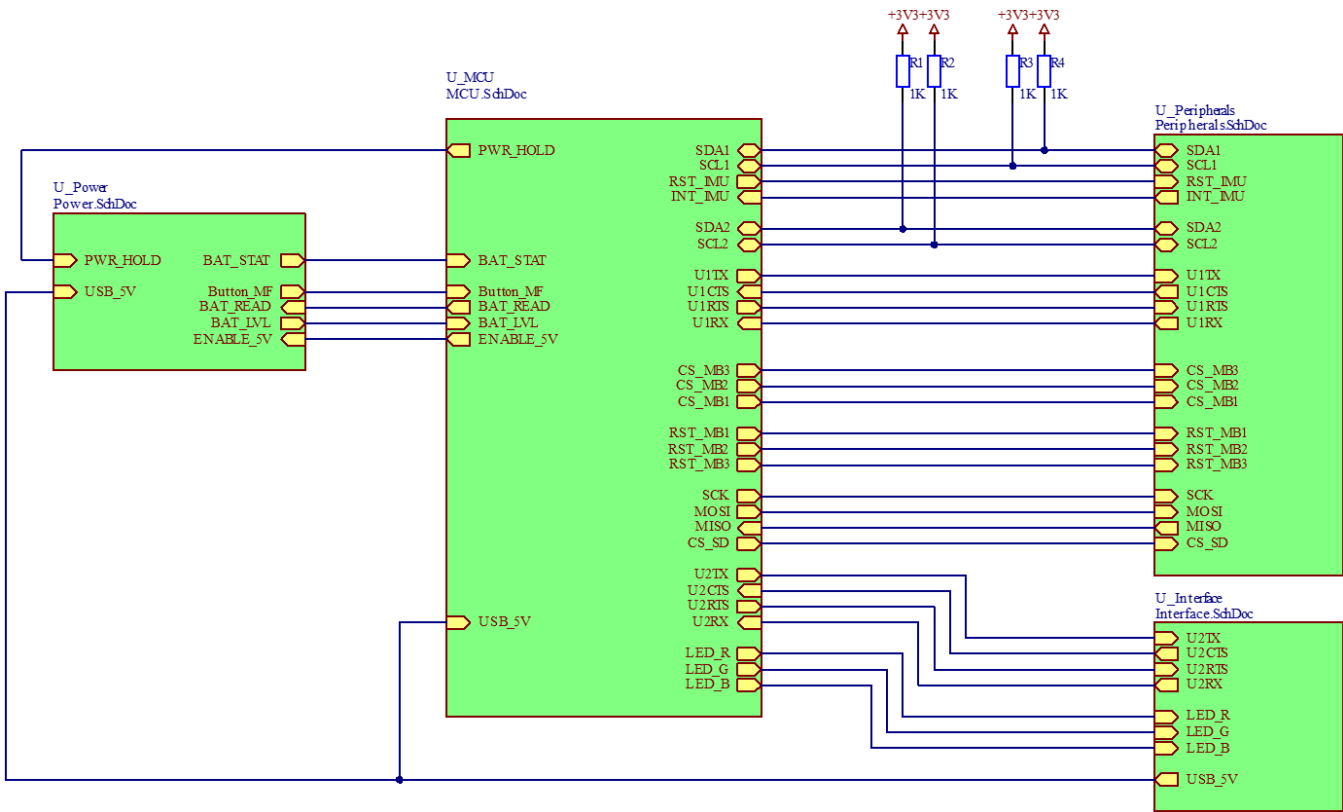


FIGURE 11 – Schéma bloc de la schématique  
Source : Auteur

Nous pouvons constater sur la figure 11 la structure des différents blocs du schéma :

Bloc	Description
Power	Contient les différents régulateurs du système, ainsi que la gestion de charge de la batterie.
MCU	Contient l'intelligence du système, avec le microcontrôleur ainsi que tous ses composants passifs associés.
Peripherals	Périphériques du système : Carte-SD, Centrale inertielle, Capteur de pression, Slots MikroE.
Interface	Connecteur USB avec convertisseur serial (FTDI) et tous les composants passifs de sécurité. Interface LED RGB pour le statut.

### 3.2.2 Autonomie du système

Afin de proportionner la batterie du circuit, il a fallu dimensionner les différentes consommations des composants, ceci par le biais de leurs documentations :

MCU - 30mA	BNO055 - 12.3mA	Capt. Pression - 4mA
Carte-SD - 100mA	MikroE - ??mA	Régulateurs - 40uA
LED RGB - 25mA		

Nous pouvons constater que la plus grande consommation vient de la carte micro-SD, qui au maximum peut induire 100mA.<sup>2</sup>

Afin d'obtenir une autonomie d'au moins 2h (selon CDC), il faudrait une capacité de :

$$Capacite = Consommation_{tot} * Temps \quad (3)$$

Ce qui nous fait une capacité de  $\sim 342.68\text{mAh}$ , valeur facilement atteignable par les batteries li-ion du marché. Étant-donné que différents projets utilisaient des batteries 3400mAh, dans un objectif de conformité et de simplification des commandes, j'ai choisis cette même valeur. Ce qui signifie une autonomie de  $\sim 20$  heures, sans compter les différents mécanismes d'économie d'énergie.

C'est un temps largement suffisant pour la durée de plusieurs expéditions, néanmoins la RTCC du microcontrôleur requiert d'être alimentée en permanence, j'ai donc décidé de déterminer un fonctionnement, où lorsque l'on charge la batterie la date se mettrait à jour et le mode "éteint" serait juste un mode de veille qui attendrait un niveau positif sur le switch avant de commencer le logging avec un timestamp principale contenant la date, puis, seulement des deltas entre les mesures. Un diagramme des états est présents à la figure 12.

2. Selon datasheet SanDisk : <https://images-na.ssl-images-amazon.com/images/I/91tTtUMDM3L.pdf>

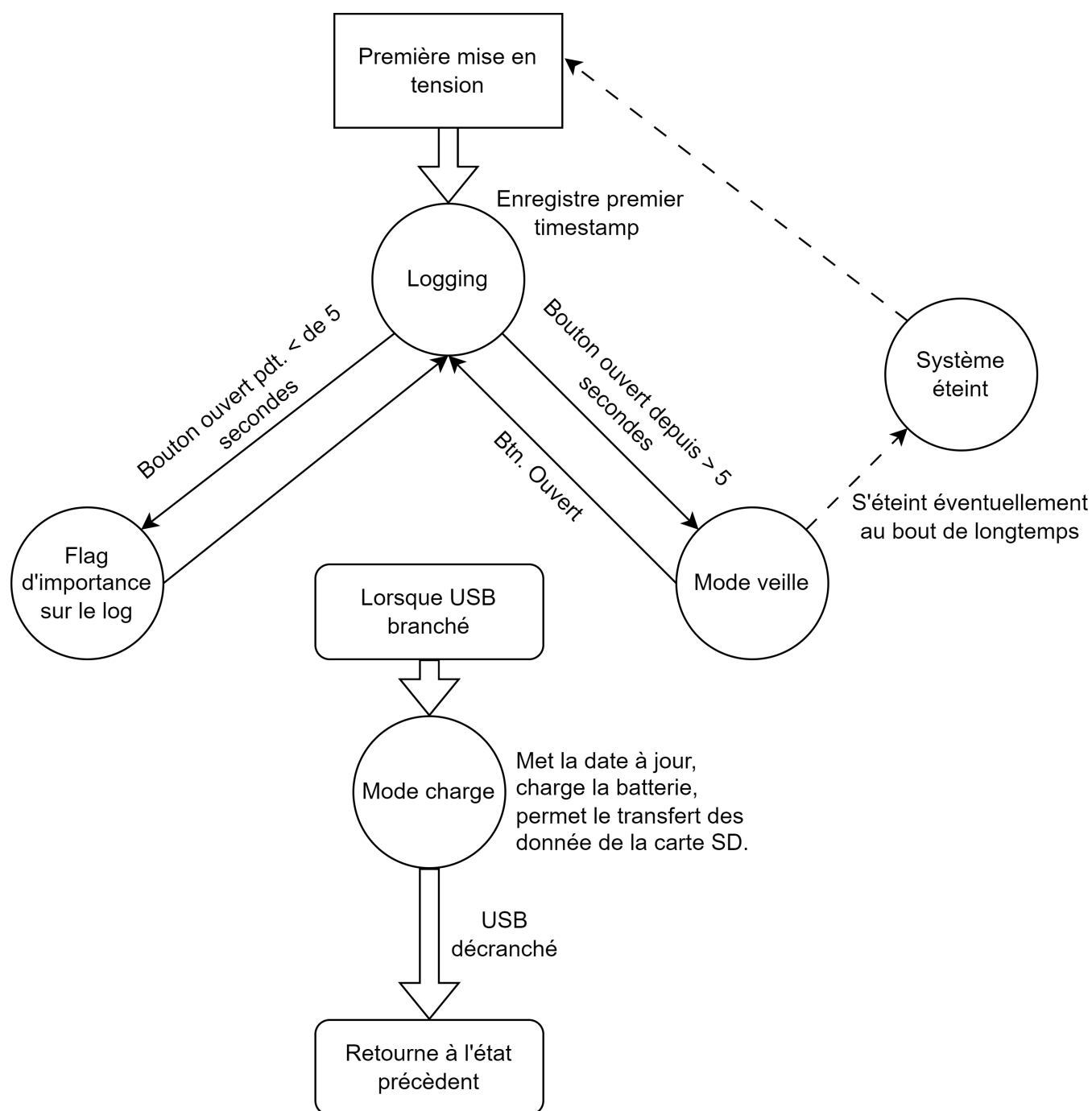


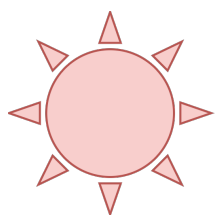
FIGURE 12 – Diagramme des états du système  
Source : Auteur

### 3.2.3 LED Interface

Afin d'informer l'utilisateur de ce qu'il se passe dans le système, j'ai décidé d'implémenter en tant qu'interface, une led RGB. Celle-ci sera un minimum puissante, afin de pouvoir être lisible lors de l'utilisation sous-marine du module.

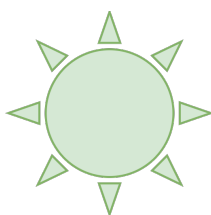
La consommation de la led RGB étant relativement importante, des mécanismes d'économie d'énergie seront mis en place dans le développement firmware.

#### Diagramme d'interface :



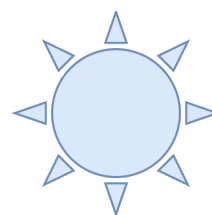
#### Rouge

- Constamment allumé : Indique que la batterie doit être chargée.
- Clignote rapidement ; Indique le passage en mode veille ou le passage en mode éteint.
- Clignote lentement: Indique que la carte SD est pleine.



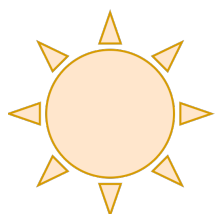
#### Vert

- Clignote lentement : Indique que le logging est en cours.
- Clignote rapidement : Indique que la charge est en cours
- Allumé et carte branché : Indique que la charge est complète.



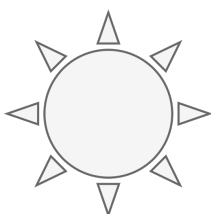
#### Bleu

- S'allume : Indique que le flag d'importance a été enregistré.
- Clignote rapidement : Indique qu'un transfert de fichier est en cours.



#### Orange

- Indique une erreur autre.



#### Eteint

- Constamment éteint : Indique que l'appareil est en veille ou entièrement éteint.

FIGURE 13 – Définitions des états de la LED RGB  
Source : Auteur

### 3.2.4 Adaptation mécanique

L'idée étant d'obtenir une mesure de pression sans modification mécaniques sur le boîtier originale, plusieurs idées ont émergées :

- 1) Mesurer une déformation mécanique a-même le module, dans le but de déduire la pression (Développement d'un capteur).
- 2) Ajout d'une rallonge cylindrique au module, afin de fixer un capteur de pression à plat sur celui-ci, tout en permettant les modifications mécaniques sans altération du boîtier originale.

Par sa complexité moins importante et due aux contraintes de temps, la seconde option sera-celle développée lors de cette version du projet. Voici des ébauches (Pas à l'échelle) du concept :

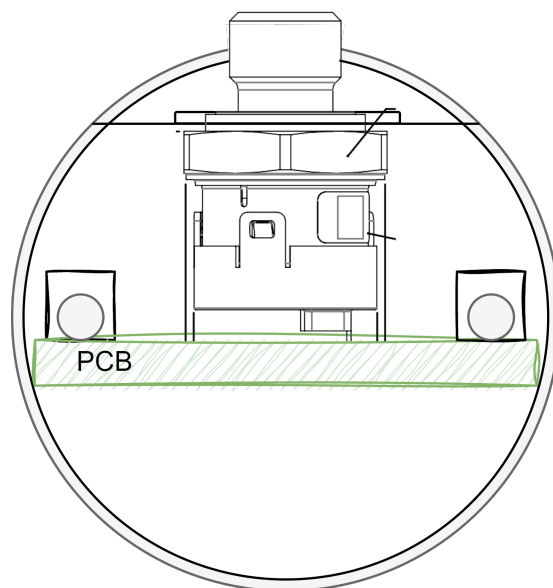


FIGURE 14 – Ébauche intérieur du cylindre  
Source : Auteur

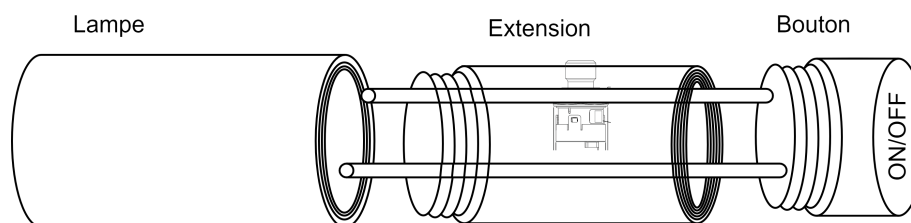


FIGURE 15 – Ébauche globale  
Source : Auteur

### 3.2.5 Bus de communications

**UART (1) :**

Utilisation :

Pinning :

**UART (2) :**

Utilisation :

Pinning :

**SPI :**

Utilisation :

Pinning :

**I2C (1) :**

Utilisation :

Pinning :

**I2C (2) :**

Utilisation :

Pinning :



**3.2.6 Chargeur de batterie****3.2.7 Régulateur 3.3V****3.2.8 Régulateur 5V**

## Références

- [1] A. Bradley, M. Feezor, H. Singh, and F. Yates Sorrell, “Power systems for autonomous underwater vehicles,” vol. 26, no. 4, pp. 526–538. Conference Name : IEEE Journal of Oceanic Engineering.
- [2] N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, “Multi-sensor fusion for underwater vehicle localization by augmentation of RBF neural network and error-state kalman filter,” vol. 21, no. 4, p. 1149. Number : 4 Publisher : Multidisciplinary Digital Publishing Institute.
- [3] A. S. Zaki, T. B. Straw, M. J. Obara, and P. A. Child, “High accuracy heading sensor for an underwater towed array.”