

# Document de réalisation

## du gestionnaire de moteur



# Table des matières

1 Introduction.....	3
2 Outils nécessaires.....	4
3 Composants nécessaires.....	4
4 Description du module principal et des périphériques.....	6
5 Descriptions des périphériques.....	7
5.1 Le branchement du gestionnaire sur l'alternateur.....	7
5.2 Le thermomètre du moteur (ou de son compartiment).....	7
5.3 Contrôle du ventilateur du compartiment.....	8
6 Description de la carte électronique principale.....	10
6.1 L'électronique d'adaptation de la ligne W de l'alternateur.....	11
6.2 Le sous-module traitant la tension de la batterie du moteur.....	11
6.3 Le sous-module du buzzer.....	12
6.4 Le sous-module des boutons pousoirs.....	13
6.5 Module d'écriture et de lecture sur carte SD.....	13
6.6 L'écran LCD.....	14
6.7 Le module bluetooth HC-06.....	14
7 Configuration du gestionnaire de moteur.....	16
7.1 Configuration du dongle Bluetooth HC-06.....	16
7.2 Configuration des capacités de gasoil.....	18
7.3 Configuration des variables du moteur.....	18
7.4 Configuration des paramètres de consommation.....	19
7.4.1 Configuration de la constante de l'alternateur.....	20
7.4.2 Configuration du type de moteur.....	21
7.4.3 Configuration de constantes empiriques.....	22

# **1 Introduction**

Ce document est destiné à ceux qui veulent réaliser le gestionnaire de moteur. Il guide étape par étape dans sa réalisation et sa configuration.

Il est nécessaire de savoir souder correctement, d'avoir quelques outils de base de l'électronicien et de savoir compiler ou suivre un tutoriel informatique.

Le gestionnaire de moteur est à monter soi-même et à configurer en fonction du type du moteur et de la capacité des réservoirs de l'embarcation.

## 2 Outils nécessaires

- fer à souder avec une pointe fine
- une pompe pour dessouder ou une tresse de cuivre à utiliser dans le cas d'une erreur dans le soudage
- un support pour le fer et une éponge humide est bien pratique
- de l'étain à souder de diamètre 0.8 était assez pratique pour moi
- une pince à couper du câble de petite dimension
- des câbles électrique de diamètre très fin (type fil de couleur dans un câble ethernet)

## 3 Composants nécessaires

Nom du produit	Référence Banggood	Prix indicatif
LM35	SKU119146	1,50 €
1 résistance de 75 Ω		
1 résistance de 5 kΩ		
2 résistances de 22 kΩ		
1 résistance de 10 kΩ		
1 résistance de 20 kΩ		
4 résistances de 100 Ω		
1 diode Zener 1N4148	SKU136990	1,60 € les 100 pièces
1 transistor d'usage général	SKU032148	0,61 €
1 condensateur de 1 µF		
1 condensateur de 0.1 µF		
1 condensateur de 0.5 µF		
Relais optocouplé	SKU093781	1,57 €
Carte PROTO MEGA	SKU106089	2,75 €
Fils mâle vers femelle	SKU067904	1,60 €
Socket mâle et femelle	SKU039144	0,76 €
3 blocs de connecteurs	SKU113230	3,21 €
Fusible de 0.5A et son porte fusible		
Abaisseur de tension	SKU116802	1,47 €
1 buzzer	SKU127305	1,41 €

Un clavier de 4 boutons super fin	SKU094590	1,18 €
Module SD	SKU075678	1,24 €
Module bluetooth HC-06	SKU106868	5,26 €
Ecran LCD 4 lignes de 20 caractères	SKU106856	5,80 €
Arduino UNO	SKU141102	6,80 €
<b>Total</b>		<b>36,76 €</b>

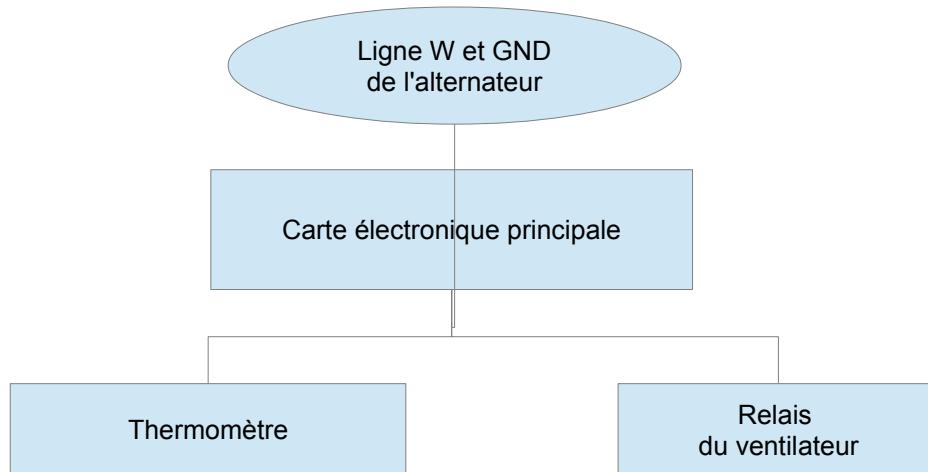
## 4 Description du module principal et des périphériques

Le module principal est contenu dans un boîtier (une boite en bois réalisée par mes soins pour ce qui me concerne) dans lequel se trouve la carte électronique principale, le buzzer et le module bluetooth. La façade de ce boîtier comporte l'interface utilisateur composé d'un écran LCD de 4 lignes de 20 caractères chacune et de 4 boutons poussoir.

Les prises USB et d'alimentation de l'arduino sont toujours accessibles car j'ai fait des ouvertures sur le côté de mon boîtier. Cependant, elles ne sont que peu utiles car étant donné qu'il est **obligatoire de débrancher le module bluetooth pour pouvoir uploader un programme dans l'arduino** via sa prise USB, il est nécessaire d'extraire l'électronique de sa boite.

Pour ce qui concerne des composants périphériques, il y a :

- Une sonde de température située sur une durite du moteur ou dans le compartiment du moteur. Ma sonde de température se trouve dans le compartiment du moteur dans une boite de dérivation électrique avec le relais du ventilateur et l'interrupteur du ventilateur.
- Un relais contrôlant un ventilateur pour le refroidissement du moteur.
- Le branchement de la carte électronique principale sur l'alternateur du moteur via la masse de l'alternateur et la ligne W.



## 5 Descriptions des périphériques

### 5.1 Le branchement du gestionnaire sur l'alternateur

Le gestionnaire de moteur se base sur le régime du moteur afin d'estimer la consommation instantanée en gasoil du moteur. Le gestionnaire du moteur pourra ensuite estimer par exemple la quantité de gasoil restante ou encore une moyenne de la consommation pour un parcours.

Le gestionnaire de moteur a été conçu pour un moteur Nanni 3.30 qui dispose d'un alternateur avec une sortie W. Il doit être possible d'adapter le gestionnaire de moteur pour d'autres types de moteur (celui-ci doit posséder néanmoins une ligne W sur son alternateur) juste en modifiant quelques paramètres du programme à uploader dans l'arduino. Nous verrons ceci dans la partie concernant la configuration du gestionnaire de moteur.

Le gestionnaire de moteur est capable de lire directement la ligne W de l'alternateur car la carte électronique principale comporte l'électronique nécessaire pour adapter le signal de la ligne W de l'alternateur pour pouvoir ensuite la lire et l'interpréter par le microcontrôleur de l'Arduino. Ainsi la partie externe à la carte électronique principale concernant le branchement à l'alternateur n'est qu'un branchement d'un double fil sur l'alternateur (GND et ligne W). Il est conseillé de :

- **torsader** ces 2 fils pour limiter les parasites pouvant perturber par exemple notre thermomètre distant...
- **gainer** car ce double fil se trouve dans le compartiment moteur où l'ambiance est très chaude, grasse, humide ?, ...

La partie électronique traitant le signal de la ligne W étant embarquée dans la carte électronique principale, elle sera décrite comme un sous module de la carte électronique principale.

#### Matériels nécessaires :

Du câble 2 fils

De la gaine annulaire

### 5.2 Le thermomètre du moteur (ou de son compartiment)

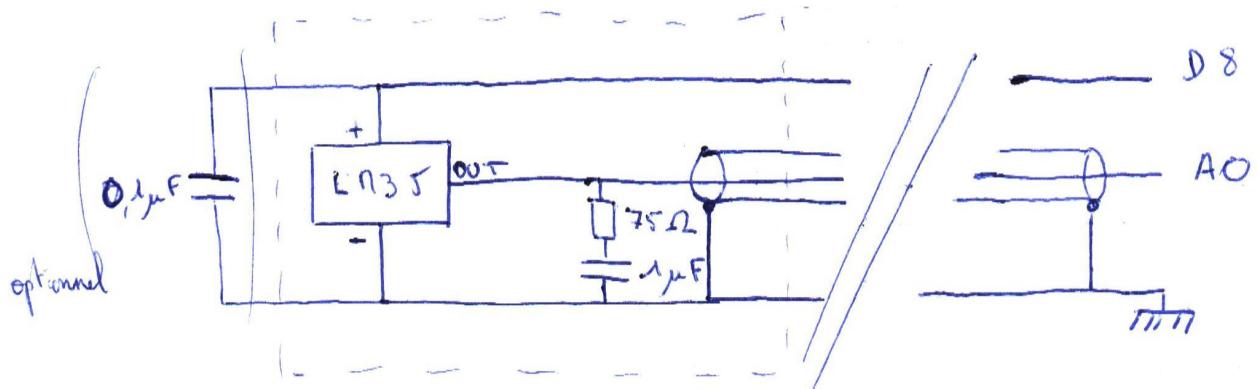
Ce n'est que après avoir terminé mon installation, que j'ai réalisé qu'il aurait peut-être été plus judicieux de positionner le capteur de température sur la durite du liquide de refroidissement rentrant dans l'échangeur de température. Cela m'aurait donné directement la température du moteur .

Dans mon installation, le capteur de température se trouve dans une boîte de dérivation (avec le relais du ventilateur) dans le compartiment du moteur. Ainsi la température renvoyée est celle du compartiment du moteur et non celle du moteur.

Suivant l'installation que vous réaliserez, les températures limites qui servent à déclencher l'alarme sonores et du ventilateur devront être adaptées à travers l'interface utilisateur (Cf Manuel d'utilisateur du gestionnaire de moteur).

Le thermomètre se base sur le composant LM35 qui est peu onéreux. Etant donné que ce composant est distant de la carte électronique principale et qu'il est situé dans un environnement très perturbés (présence de relais, d'alternateur, etc. ), il est nécessaire de l'accompagné d'un peu d'électronique et de prendre quelques précautions pour son branchement jusqu'à la carte électronique principale.

Voici le schéma électronique du thermomètre :



Je me suis largement inspiré du schéma et des informations données dans la datasheet du composant LM35.

Pour ma réalisation, je n'ai pas utilisé la capacité optionnelle. J'ai soudé le LM35, la résistance de 75 ohms et la capacité de 1uF sur une plaque pré-trouée avec un bornier à 3 entrées pour faciliter la connexion à cette petite plaque électronique. Ensuite je l'ai installée dans une boîte de dérivation située dans le compartiment moteur.

Un câble double fil (GND et le +5V d'alimentation du capteur) et un câble coaxiale permet de relier la petite carte électronique du thermomètre à la carte électronique principale. Il faut noter que la tresse du câble coaxiale est connecté au GND à chaque extrémité ...

Le câble coaxiale permet de protéger le faible signal de sortie du LM35 des perturbations extérieures.

Dans le cas d'une installation sur une durite du moteur, j'avais pensé à embarquer la petite électronique dans un petit tube de comprimé Efferalgant avec les câbles rentrant d'un côté et le capteur LM35 sortant de quelques centimètres seulement de l'autre côté afin de pouvoir le scotché directement sur une durite.

Pour ce qui concerne la longueur des câbles, dans mon installation, j'ai environ 1.5m de longueur et j'ai une température qui semble suivre correctement l'évolution de la température du compartiment.

Une autre remarque concerne l'alimentation de ce capteur par la ligne D8. Cela permet de mettre sous tension le capteur que lorsque cela est nécessaire. En effet, j'ai remarqué que si l'on garde le capteur sous tension, le capteur chauffe de quelques degrés et par conséquent injecte une erreur de quelques degrés dans la mesure de la température. De plus, alimenter le capteur que lorsque cela est nécessaire permet d'économiser quelques milli-ampère.

#### Matériels nécessaires :

LM35

résistance de 75 Ω

Condensateurs de 1μF et de 0.1μF

Un câble double fil pour alimenter le capteur

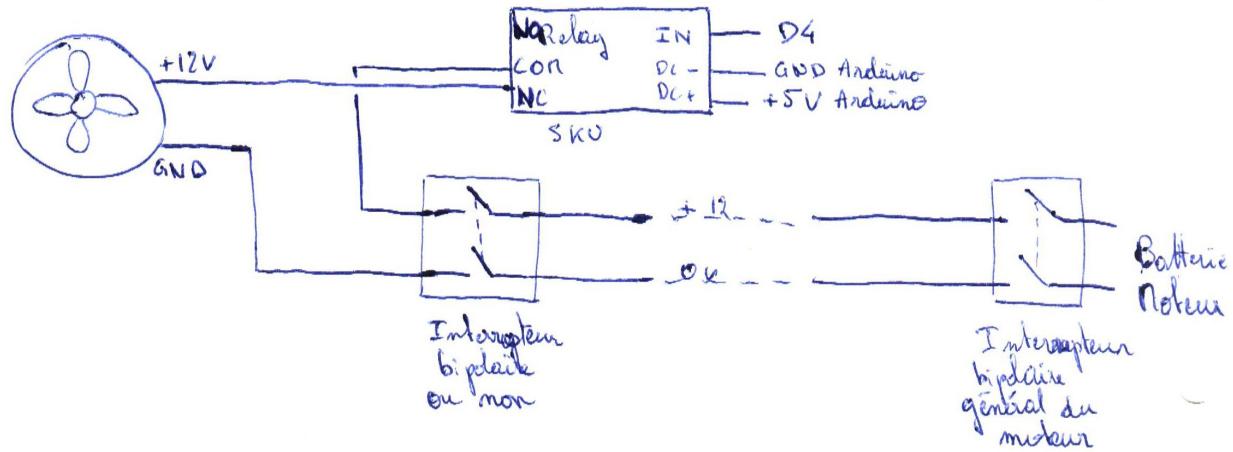
Un câble coaxial pour transporter la donnée

### **5.3 Contrôle du ventilateur du compartiment**

Afin d'économiser la durée de vie limitée du ventilateur situé dans le compartiment moteur, de

laisser aussi le moteur monter en température, le gestionnaire de moteur permet de contrôler la mise sous tension d'un ventilateur en agissant comme le ferait un thermostat. La température limite de déclenchement est configurable à travers l'interface utilisateur (Cf. Manuel utilisateur du gestionnaire de moteur).

Voici comment le ventilateur du compartiment et son relais contrôlé par l'Arduino sont branchés dans mon installation :



#### Matériels utilisés :

La référence pour le relais utilisé chez Banggood est SKU093781

## 6 Description de la carte électronique principale

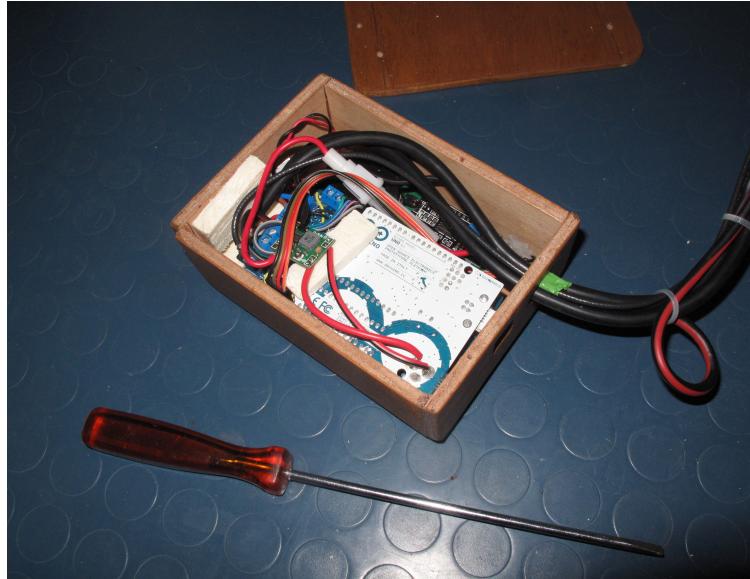
La carte électronique principale est contenu dans un boîtier visible par l'utilisateur car il comporte en façade l'écran LCD et les boutons poussoirs permettant d'interagir avec l'utilisateur.



L'électronique embarquée dans ce boîtier peut être décomposé en 7 sous-module:

- La carte Arduino Uno comportant le microcontrôleur ATMega avec toutes ses entrées/sorties digitale et analogique
- L'électronique adaptant le signal de l'alternateur pour le microcontrôleur
- Le régulateur de la tension des batteries pour obtenir le 7V nécessaire au fonctionnement de l'Arduino. Ce sous-module comporte aussi la protection par fusible du gestionnaire de moteur mais aussi l'adaptation de la tension d'entrée (tension de batterie moteur) pour la mesurer (simple pont diviseur de tension).
- L'écran LCD de 4 lignes à 20 caractères pour afficher les informations utiles à l'utilisateur
- Le module bluetooth pour permettre de créer un logiciel de gestion de la maintenance du moteur pour PC ou tablette.
- Le buzzer pour alerter en cas d'une tension de batterie moteur trop faible ou d'une température trop élevée.
- Les 4 boutons poussoirs pour naviguer dans les menus
- Le module de lecture et écriture sur carte SD permettant de sauvegarder certaines informations même après l'extinction du gestionnaire de moteur.

L'écran, les 4 boutons poussoirs, le module bluetooth, le buzzer et la carte Arduino sont interconnectés entre eux grâce à la carte PROTO SHIELD MEGA. Cette carte est sensée être utilisée avec l'Arduino MEGA mais il est tout à fait possible de l'utiliser avec la carte Arduino UNO.



J'ai utilisé cette carte pour y souder tous les composants électroniques (résistances, condensateurs, bornier et rangés de fiche male et femelle). Son format et son pré-cablage est pratique. Il permet de simplifier le cablage, d'économiser du temps et de la soudure.

Pour ce système, étant donné que pas moins de 8 fils provenant de l'extérieur doivent être branchés sur cette carte proto, l'utilisation de bornier soudé sur la plaque rend l'installation plus commode, plus propre et plus solide.

J'ai aussi utilisé des rangés de connecteur mâle pour pouvoir superposer la carte PROTO sur l'Arduino UNO. Des rangés de fils soudés sur la carte PROTO terminées par des connecteurs femelle permettent de brancher le module bluetooth et l'écran.

#### Matériels utilisés :

Carte PROTO MEGA (Banggood SKU106089)

Fils mâle vers femelle (Banggood SKU067904)

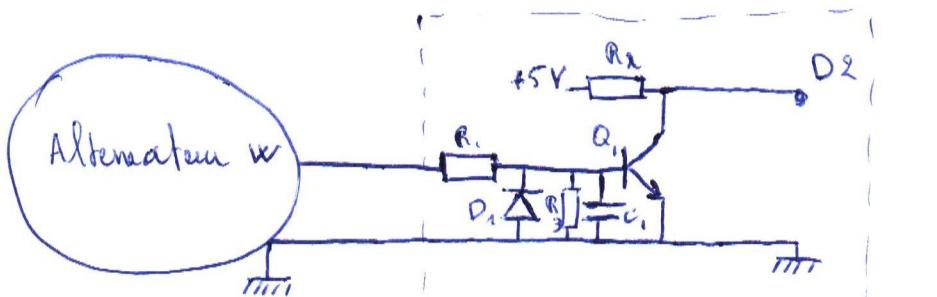
Socket mâle et femelle (Banggood SKU039144)

Bloc de connecteurs (Banggood SKU113230)

Carte Arduino UNO (Banggood SKU141102)

### **6.1 L'électronique d'adaptation de la ligne W de l'alternateur**

Pour cette partie, je me suis inspiré d'un schéma trouvé sur sonelec que je reproduis ici :



$$\begin{aligned}
 R_1 &= 5 \text{ k}\Omega && (2 \text{ (marron noir change) marron (bleu)}) \\
 R_2 &= 22 \text{ k}\Omega && (\text{rouge rouge noir rouge) marron (bleu}) \\
 R_3 &= 22 \text{ k}\Omega \\
 D_1 &= 1N4148 && 1N4448 \\
 C_1 &= 0.5 \mu\text{F} \\
 Q_1 &= 2N5551
 \end{aligned}$$

J'ai recalculé les valeurs de R1 et C1 pour avoir un filtre passe-bas dont la fréquence de coupure est de l'ordre de 640 Hz.

Pour le moteur nanni, si le rupteur envoie un signal de 640 Hz, cela voudrait dire que le régime du moteur serait de 4170 tr/min, ce qui est impossible sur mon moteur.

L'électronique présenté ici est soudé sur la carte PROTO. Deux emplacements sont utilisés sur le bornier soudé afin de brancher le GND et le W de l'alternateur.

Comme indiqué sur le schéma, c'est sur D2 que les impulsions adaptées pour l'Arduino et provenant de l'alternateur sont lues.

On remarque dans ce schéma que la masse du moteur est commune avec la masse de l'arduino d'où la nécessité d'alimenter le gestionnaire de moteur par la batterie du moteur. **Il n'est pas possible d'alimenter le gestionnaire de moteur avec les batteries de servitude.**

#### Matériels nécessaires :

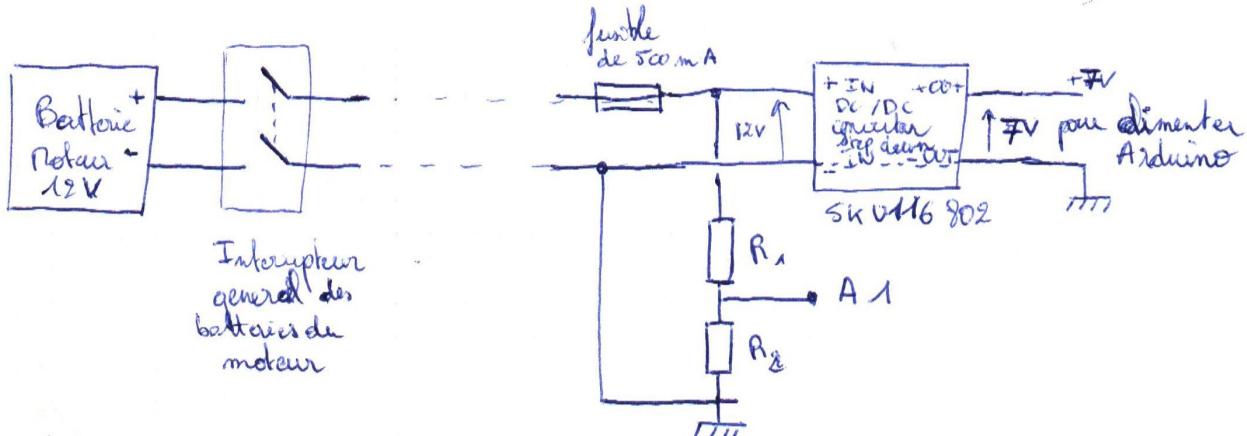
- 1 résistance de 5 kΩ
- 2 résistances de 22 kΩ
- 1 diode Zener (ici 1N4148)
- 1 condensateur de 0.5 μF
- 1 transistor d'usage général (ici 2N5551)

## 6.2 Le sous-module traitant la tension de la batterie du moteur

Ce sous module permet de :

- protéger le gestionnaire de moteur par un fusible monté sur la ligne +12V. Si vous souhaitez protéger le gestionnaire de moteur d'une inversion de polarité, vous pouvez monter un fusible identique sur le négatif.
- Il permet aussi d'adapter la tension en sortie de la batterie moteur afin de mesurer sa tension grâce à une entrée analogique du microcontrôleur.
- Il permet de réguler la tension des batteries pour obtenir une tension de 7V afin d'alimenter correctement le gestionnaire de moteur.

Voici son schéma :



$$R_2 = 10 \text{ k}\Omega \quad (\text{marron noir noir rouge})$$

$$R_1 = 20 \text{ k}\Omega \quad (2 \text{ série marron noir noir rouge})$$

On voit que la tension de la batterie du moteur est divisée par 3. Cela implique que les batteries du moteur ne doivent pas excéder 15 volts ... La tension est mesurée sur l'entrée A1.

Le convertisseur DC/DC permet de réguler la tension de la batterie à une tension permettant d'alimenter l'Arduino UNO. J'ai utilisé le composant dont la référence Banggood est SKU116802. Un potentiomètre permet de régler la tension de sortie désirée. Ce composant est très pratique et pas cher. Un autre composant encore moins cher permet de faire la même chose et à l'avantage de pouvoir être soudé sur la plaque du proto : C'est le composant LM317 accompagné de 2 résistances et de 2 condensateurs.

#### Matériels nécessaires :

Un fusible de 0.5A et son porte fusible

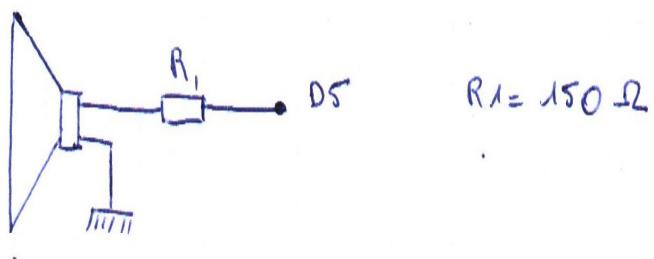
Un abaisseur de tension du type LM317 ou Banggood SKU116802

1 résistance de 10 kΩ

1 résistance de 20 kΩ

### 6.3 Le sous-module du buzzer

Pour le branchement du buzzer, c'est classique :



#### Matériels nécessaires :

Un buzzer (Si Banggod SKU127305 pas besoin de résistance)

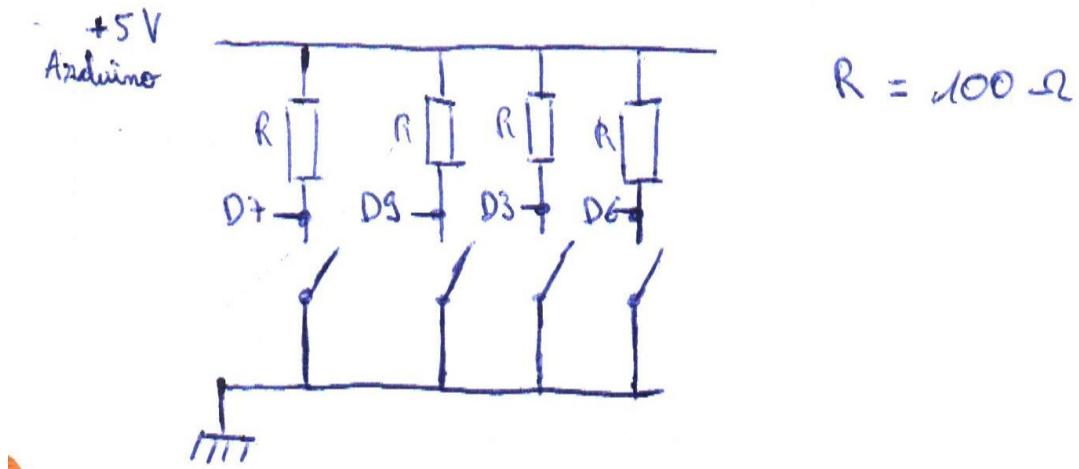
1 résistance de  $150\ \Omega$  si buzzer classique

## 6.4 Le sous-module des boutons poussoirs

Pour les boutons permettant une interaction entre le système et l'utilisateur, j'ai voulu tester des boutons poussoir que l'on peut trouver sur des photocopieuse par exemple. Ils ne sont pas cher, semblent bien résister à l'humidité et sont esthétiquement pas vilains.

Il faut prévoir de les coller « à vie » avec une colle de type bi-composant ou glue car l'auto adhésif n'est pas convaincant.

Afin de protéger les entrées digitales d'une sur-intensité, il faut utiliser une résistance :



Matériels utilisés :

4 résistances de  $100\ \Omega$

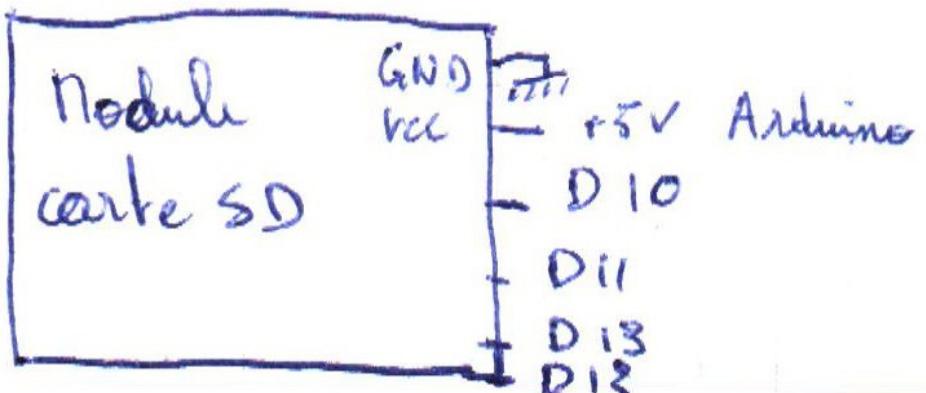
Un clavier de 4 boutons super fin (Banggood SKU094590)

## 6.5 Module d'écriture et de lecture sur carte SD

Cette solution technique n'est pas vraiment justifiée étant donné la faible quantité d'information sauvegardée. J'ai entendu parlé de circuit imprimé simple d'utilisation contenant de la mémoire flash en petite quantité ce qui aurait été suffisant pour le besoin. Cela aurait certainement pu permettre d'économiser de l'espace, de l'argent et quelques milli-ampère.

On peut se consoler en se disant que le module de lecture/écriture pourra servir un jour à logger l'utilisation du moteur un peu comme le ferait une boîte noire ... Cependant, pour faire un bon logger, il faudrait connaître l'heure ...

Le module de lecture/écriture est directement soudé sur la carte PROTO et câblé de la façon suivante :



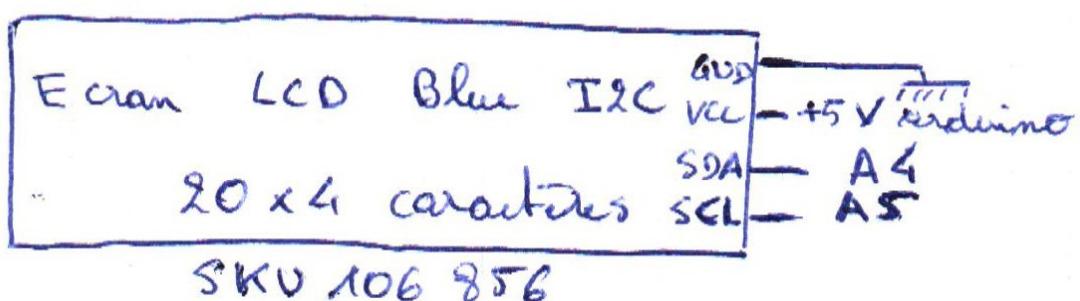
#### Matériels utilisés :

Module SD (Banggood SKU075678)

### 6.6 L'écran LCD

Le gestionnaire de moteur utilise un écran LCD de 4 lignes à 20 caractères afin de restituer des informations à l'utilisateur. C'est un écran LCD rétro-éclairé en bleu à fort contraste agréable à lire.

Voici comment la branchement est prévu dans le gestionnaire de moteur :

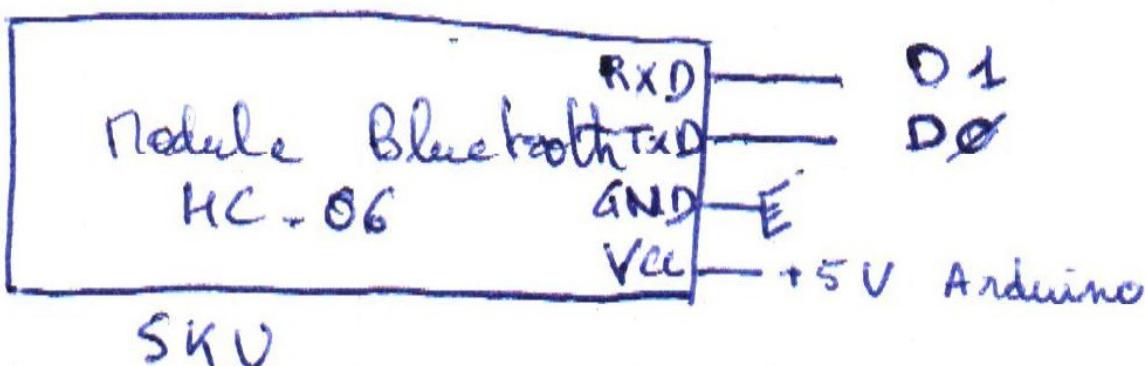


#### Matériels utilisés :

Ecran LCD 4 lignes de 20 caractères (Banggood SKU106856)

### 6.7 Le module bluetooth HC-06

Branchement simpliste pour le module bluetooth :



#### Matériels utilisés :

Module Bluetooth HC-06 (Banggood SKU106868)

## 7 Configuration du gestionnaire de moteur

Cette partie du document concerne la configuration du système pour l'adapter à votre propre embarcation. Il est considéré que toute la partie matériel (les périphériques, la carte électronique principale et ses sous-modules) a été réalisé.

Le système n'attend plus que le microcontrôleur reçoive le sketch. Mais avant de charger le sketch, il est nécessaire d'apporter quelques modifications dans le sketch lui-même, de configurer la module bluetooth et de créer un fichier de propriété sur la carte SD (située dans le module SD).

Chaque embarcation est différente, le moteur, la capacité des réservoirs de gasoil, la capacité du réservoir principale, la méthode d'installation du thermomètre (durite ou dans compartiment moteur) peuvent varier d'une embarcation à l'autre.

Il existe deux types de configuration :

- La configuration utilisateur qui s'effectue à travers l'interface utilisateur. Cette configuration est traitée dans le manuel de l'utilisateur du gestionnaire de moteur.
- La configuration du système est effectuée avant l'installation. Elle ne s'effectue généralement qu'une seule fois. Elle est stockée « en dur » dans le programme et concerne des paramètres qui ne sont pas supposés changer :
  - Le nom du gestionnaire de moteur visible en bluetooth
  - La vitesse en baud du bluetooth
  - Le pin du bluetooth
  - La capacité totale de gasoil
  - La capacité du réservoir principal
  - Les paramètres de la fonction permettant de calculer la consommation instantanée à partir du régime moteur. Ces paramètres sont liés au type du moteur et de l'embarcation.

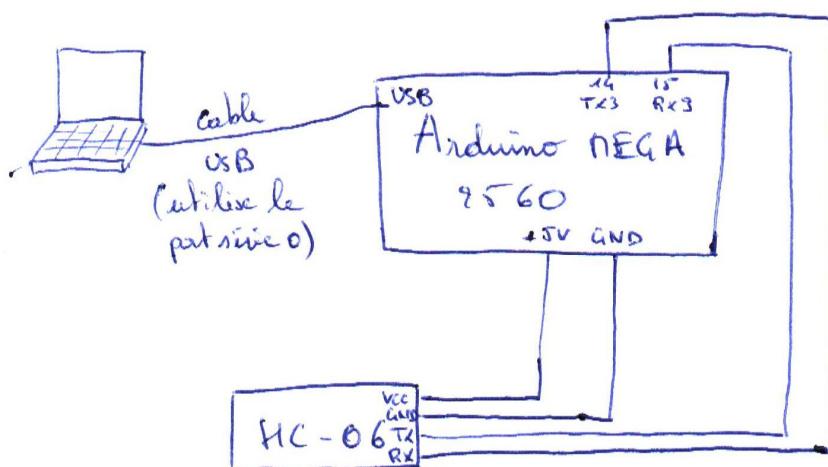
### 7.1 Configuration du dongle Bluetooth HC-06

Je décris la méthode extrêmement simple que j'ai utilisé pour configurer le dongle bluetooth du gestionnaire de moteur avant de le brancher au système et de le caler dans la boîte.

Le principe est d'envoyer des commandes AT au dongle Bluetooth grâce à un port série. Pour ce faire je vais utiliser le port série d'un microcontrôleur Arduino.

Sur une plaque d'essais, j'ai brancher le dongle HC-06 avec un Arduino Mega. L'arduino Mega est très pratique ici car il dispose de plusieurs port série. Le port série 0 sert à la saisie des commandes et le port série 3 va servir à transmettre (et recevoir) vers (et depuis) le dongle HC-06.

Voici le schéma de branchement :



Après avoir réalisé ce branchement, lancer l'IDE de l'Arduino.

Ecrivez ce sketch :

```
void setup() {  
    Serial.begin(9600);  
    Serial3.begin(9600);  
}  
  
void loop() {  
    if(Serial.available()){  
        Serial3.write(Serial.read());  
    }  
    if(Serial3.available()){  
        Serial.write(Serial3.read());  
    }  
}
```

Ce sketch considère que votre dongle HC-06 est configuré par défaut à 9600 bauds.

Téléverser le sketch dans l'Arduino Mega.

Ouvrez le moniteur série (Ctrl Maj M).

Vérifier que 9600 baud et « Pas de fin de ligne » soient bien sélectionnés.

Le moniteur série de l'IDE permet de taper au clavier des commandes dans le champ de texte et de les envoyer sur le port série 0 (via le cable USB) grâce au bouton Envoyer. Il permet aussi d'afficher le retour du port série 0.

Pour vérifier que la communication entre le dongle HC-06 et l'Arduino Mega s'effectue correctement, taper la commande suivante :

*AT*

L'envoie de cette commande devrait faire apparaître en retour le message OK.

Voici les commandes AT reconnues par le dongle HC-06 :

COMMAND	RESPONSE	COMMENT
AT	OK	Used to verify communication
AT+VERSION	OKlinvorV1.8	The firmware version (version might depend on firmware)
AT+NAMExyz	OKsetname	Sets the module name to "xyz"
AT+PIN1234	OKsetPIN	Sets the module PIN to 1234
AT+BAUD1	OK1200	Sets the baud rate to 1200
AT+BAUD2	OK2400	Sets the baud rate to 2400

AT+BAUD3	OK4800	Sets the baud rate to 4800
AT+BAUD4	OK9600	Sets the baud rate to 9600
AT+BAUD5	OK19200	Sets the baud rate to 19200
AT+BAUD6	OK38400	Sets the baud rate to 38400
AT+BAUD7	OK57600	Sets the baud rate to 57600
AT+BAUD8	OK115200	Sets the baud rate to 115200
AT+BAUD9	OK230400	Sets the baud rate to 230400
AT+BAUDA	OK460800	Sets the baud rate to 460800
AT+BAUDB	OK921600	Sets the baud rate to 921600
AT+BAUDC	OK1382400	Sets the baud rate to 1382400

Attention à changer la vitesse en baud lors de la dernière commande car autrement, vous devrez téléverser à nouveau dans l'Arduino MEGA un sketch avec la correcte vitesse ...

Dans mon système, j'ai configuré le dongle comme suivant :

Nom : EngineMonitor

Vitesse : 4800 baud

Pin : 0000

J'ai mis 4800 baud car c'est largement suffisant. De plus cela permet de limiter la bande passante dédiée pour ce système.

**Le dongle est maintenant prêt à être utilisé mais il doit être branché seulement APRES le chargement du sketch dans l'Arduino UNO sinon vous obtiendrez un message d'erreur lors de l'upload du sketch.**

## 7.2 Configuration des capacités de gasoil

Pour le moment, les capacités totale de gasoil du navire et du réservoir principale sont dans le SKETCH et ne peuvent être modifiés. Afin d'adapter le système à votre embarcation, il est nécessaire d'adapter ces valeurs.

Ouvrez le sketch du gestionnaire de moteur puis modifier les 2 lignes suivantes :

- #define QTELIMIT 220000000
- #define QTMAINLIMIT 72000000

La première ligne concerne la quantité totale de gasoil que peut transporter le navire (géricanes inclus) en micro litre. Dans la ligne ci-dessous, 220 litres a été saisi.

La seconde ligne concerne la capacité du réservoir principal. Dans l'exemple, 72 litres sont renseignés.

Il suffit maintenant de modifier ces 2 lignes aux contenus de votre embarcation.

## 7.3 Configuration des variables du moteur

Cette étape de la configuration du système est en quelque sorte son initialisation en vue du premier démarrage. Ainsi grâce à cette étape, le système pourra démarrer avec la réelle quantité de gasoil des réservoirs, mais aussi avec l'age réel du moteur, une valeur cohérente des seuils de température,

...

Ces variables sont mises à jour périodiquement par le gestionnaire de moteur. Elles se trouvent donc dans la mémoire de la carte SD qui peut être lu et écrite à volonté.

En effet, le lecteur de carte SD doit comporter une carte SD de faible capacité car un seul fichier y sera contenu.

Sur cette carte SD, il faut créer un fichier nommé *P* qui contiendra les variables du moteur.

Il est important de respecter la syntaxe qui sera reconnue par le gestionnaire de moteur. J'utilise l'éditeur Notepad++ qui me permet d'afficher TOUT les caractères d'un fichier texte.

Ce fichier *P* est très simple, il contient un entier par ligne codé sur 11 caractères et de 2 caractères permettant de coder le retour à la ligne (\n\r) représenté comme ceci dans NotePad++ :

**CRLF**

Voici un exemple du fichier *P* :

```
1 3690000... CRLF
2 220000000... CRLF
3 0..... CRLF
4 0..... CRLF
5 0..... CRLF
6 0..... CRLF
7 50..... CRLF
8 40..... CRLF
9 72000000... CRLF
10 CRLF
```

Voici sa description ligne par ligne :

1. Age du moteur en seconde. Dans l'exemple, 1025h est renseigné.
2. La quantité du gasoil en micro Litre. Dans l'exemple, 220L est renseigné.
3. Renseigner 0
4. Renseigner 0
5. Renseigner 0
6. Renseigner 0
7. Température en degré Celsius à laquelle le buzzer émettra une alarme.
8. Température en degré Celsius à laquelle le ventilateur sera mis en route.
9. Quantité de gasoil dans le réservoir principal en micro litre. Dans l'exemple, 72 litres est renseigné.

**A ce moment de la réalisation et de l'installation du gestionnaire de moteur, vous êtes prêt à charger le sketch dans l'arduino UNO.**

Il est nécessaire de charger le sketch, de mettre pour la première fois en route le gestionnaire de moteur pour effectuer la suite de la configuration.

**Attention** : Lors du 1er démarrage, et tant que la configuration ne sera pas terminée, le régime moteur et sa consommation instantanée risquent d'être incohérents avec la réalité.

## 7.4 Configuration des paramètres de consommation

La configuration des paramètres liés au moteur se déroule en 3 étapes :

1. Configurer la constante de l'alternateur afin de retrouver le régime moteur à partir de la fréquence du signal transporté par la ligne W.
2. Ensuite, il faut initialiser les 3 constantes permettant de calculer la consommation à partir du régime moteur.
3. Et enfin, il faut ajuster empiriquement les variables rapprochant la consommation théorique de la consommation réelle.

#### **7.4.1 Configuration de la constante de l'alternateur**

La constante de l'alternateur permet de retrouver le régime moteur en rotation par minute en fonction de la fréquence du signal de la ligne W de l'alternateur en Hz.

Sur mon moteur Nanni 3.30, la relation est linéaire, c'est à dire que la fonction de conversion est de la forme :

$$\text{RPM} = a * F$$

avec :

- $RPM$  le régime du moteur en tour par minute
- $F$  la fréquence en Hz du signal transporté par la ligne W de l'alternateur
- $a$  vaut 6.51 pour notre moteur Nanni 3.30 et notre alternateur de 100A

Afin d'adapter le programme à votre embarcation, il est nécessaire de définir cette constante.

Pour ce qui me concerne, j'ai défini cette constante de façon empirique en utilisant notre compte-tour analogique d'origine.

Avec l'aide de votre coéquipier favori, effectuer une dizaine de relevés du régime moteur affiché par le gestionnaire de moteur et le compte tour analogique d'origine et ce à plusieurs régime moteur afin de remplir les 2 premières colonnes du tableau suivant :

<b>Régime compte-tour analogique</b>	<b>Régime moteur affiché par le gestionnaire de moteur</b>	<b>Divisé le régime du gestionnaire par 6.51</b>	<b>Divisé le régime du compte-tour analogique par le résultat de la colonne à gauche</b>

Ensuite remplissez, les 2 colonnes de droite.

Vous devriez trouver dans la colonne la plus à droite, 10 valeurs qui sont assez proches les une des

autres si toutefois il existe une relation linéaire entre le régime de votre moteur et la fréquence du signal de la ligne W.

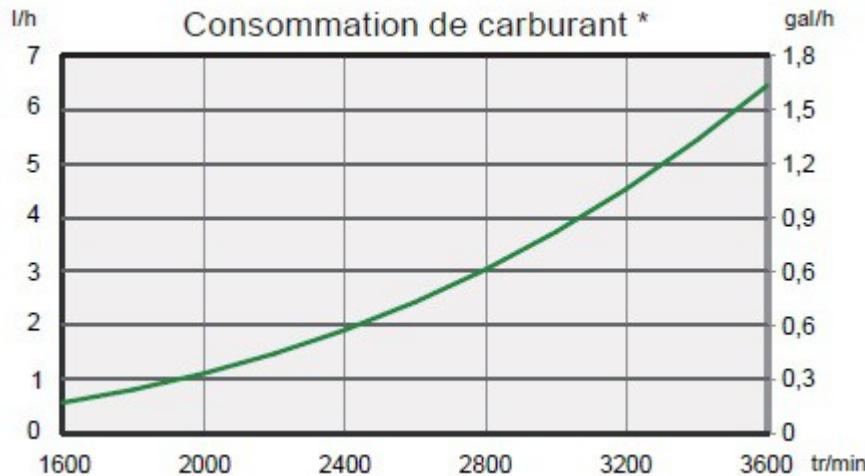
En effectuant la moyenne arithmétique de ces 10 valeurs, vous obtenez la constante de l'alternateur : Félicitations.

Dans le sketch remplacer 6.51 par la valeur que vous venez de calculer dans la ligne suivante :

```
const float ALPHA_RPM = (float)6.51;
```

## 7.4.2 Configuration du type de moteur

Chaque moteur possède une courbe de consommation de carburant. Chez Nanni, j'ai trouvé cette courbe dans le pdf nommé « Feuillet N3.30 FRA »:



Dans un tableur, je me suis amusé à relever dans cette courbe avec la plus grande précision possible la consommation de carburant en L/h au régime suivant : 1600, 2000, 2400, 2800, 3200, 3600.

Ensuite, j'ai demandé au tableur OpenOffice Calc de **calculer la courbe de tendance de l'ensemble des couples de valeurs relevés (régime, consommation) en appliquant une régression de type « puissance »**.

Open Office Calc est capable de vous sortir une équation de ce type pour les prélèvements que vous avez effectué :

$$Conso(RPM) = a * 10^b * RPM^c$$

La courbe de tendance que j'ai obtenue est très proche de ce qui a été prélevé. Effectuer le calcul du R<sup>2</sup> afin de vérifier la corrélation de la courbe de tendance avec les données prélevées : Plus la valeur de R<sup>2</sup> est proche de 1, plus la courbe de tendance est proche de tous les points prélevés. Pour mon cas, le R<sup>2</sup> est très proche de 1 (0.99996) ce qui indique une excellente corrélation.

J'ai essayé les autres type de régression mais aucune des autres type de regression possède un R<sup>2</sup> aussi proche de 1 ce qui me laisse penser que la courbe de consommation d'un moteur est de ce type.

Voici l'équation de la courbe de tendance calculé par OpenOffice Calc pour les valeurs que j'ai prélevés sur la courbe fourni par Nanni présentée précédemment :

$$Conso(RPM) = 1,1330288410882 * 10^{-10} * RPM^{3,0243858131}$$

Afin de déterminer la consommation du moteur, le sketch a besoin de connaître les 3 valeurs a, b et c. Il faut remplacer les lignes dans le sketch suivantes avec les constantes a, b et c correspondant à la

consommation de votre moteur :

```
const float a = (float)1.1330288410882;  
const float b = (float) -10;  
const float c = (float)3.0243858131;
```

### 7.4.3 Configuration de constantes empiriques

C'est pas fini avec les constantes permettant de définir la consommation du moteur ! En effet, je me suis rendu compte sans surprise que la courbe de consommation définie comme précédemment ne correspond malheureusement pas à la réalité.

Ainsi, il est nécessaire d'appliquer quelques corrections qui rapproche l'équation précédente (basé sur les données constructeur uniquement) de la réalité.

Ces 2 constantes viennent s'insérer comme ci-dessous :

$$ConsoReelle(RPM) = d * a * 10^b * RPM^c + e$$

Une constante  $d$  multiplicatrice et une constante additionnelle  $e$ .

Voici les valeurs de  $d$  et  $e$  définies pour mon embarcation :

```
const float d = (float)1.05;  
const float e = (float)0.5;
```

Au final, j'obtiens une consommation estimée comprise entre la consommation réelle et la consommation réelle + 5%.

Ces constantes sont définies de façon empirique, c'est à dire que vous devez utiliser le gestionnaire de moteur (et votre moteur). De temps en temps, il faut contrôler que la quantité de gasoil restante estimée correspond à la quantité de gasoil restante réelle. Si ce n'est pas le cas, il faut ajuster ces 2 constantes ...

**Il est conseillé que votre estimation soit supérieur à votre consommation réelle afin de ne pas avoir de mauvaise surprise en mer.**

Arrivé à cette étape, votre système doit être opérationnel. Vous ne devriez plus avoir besoin de recompiler et uploader de sketch dans l'Arduino. Ainsi, vous pouvez brancher le module Bluetooth permettant à d'autres système de profiter des informations provenant du moteur ...