

## Petite alimentation 5Vcc - 1.

Prévu pour fonctionner sous 5Vcc le petit ventilateur **V** consomme 65mA. Possibilité d'augmenter la tension en sortie avec le potentiomètre ajustable jusqu'à 7Vcc.

**Dans ce cas couper le ventilateur.**

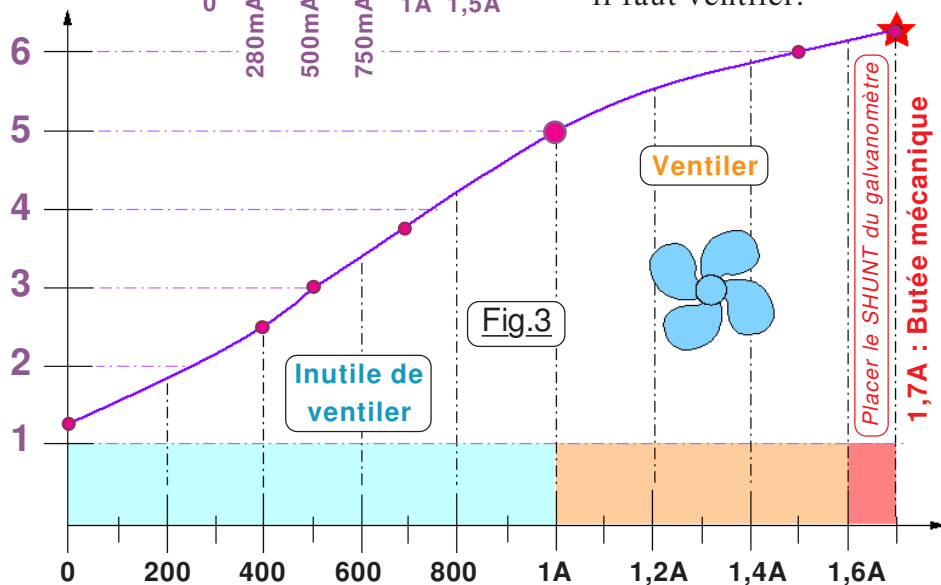
U <sub>sortie</sub>	I <sub>MAX</sub>	I <sub>indiqué</sub>
6v	1A	5,7v
7v	700mA	6,8v

**Bouton de Panique BP :** Sur incident BP annule immédiatement la tension en sortie mais laisse tourner le ventilateur. La LED **D2** s'éteint. **Débrancher la sortie ou le bloc secteur avant de relâcher le bouton poussoir** qui au repos rétablit le contact électrique. (Voir la Fig.1)

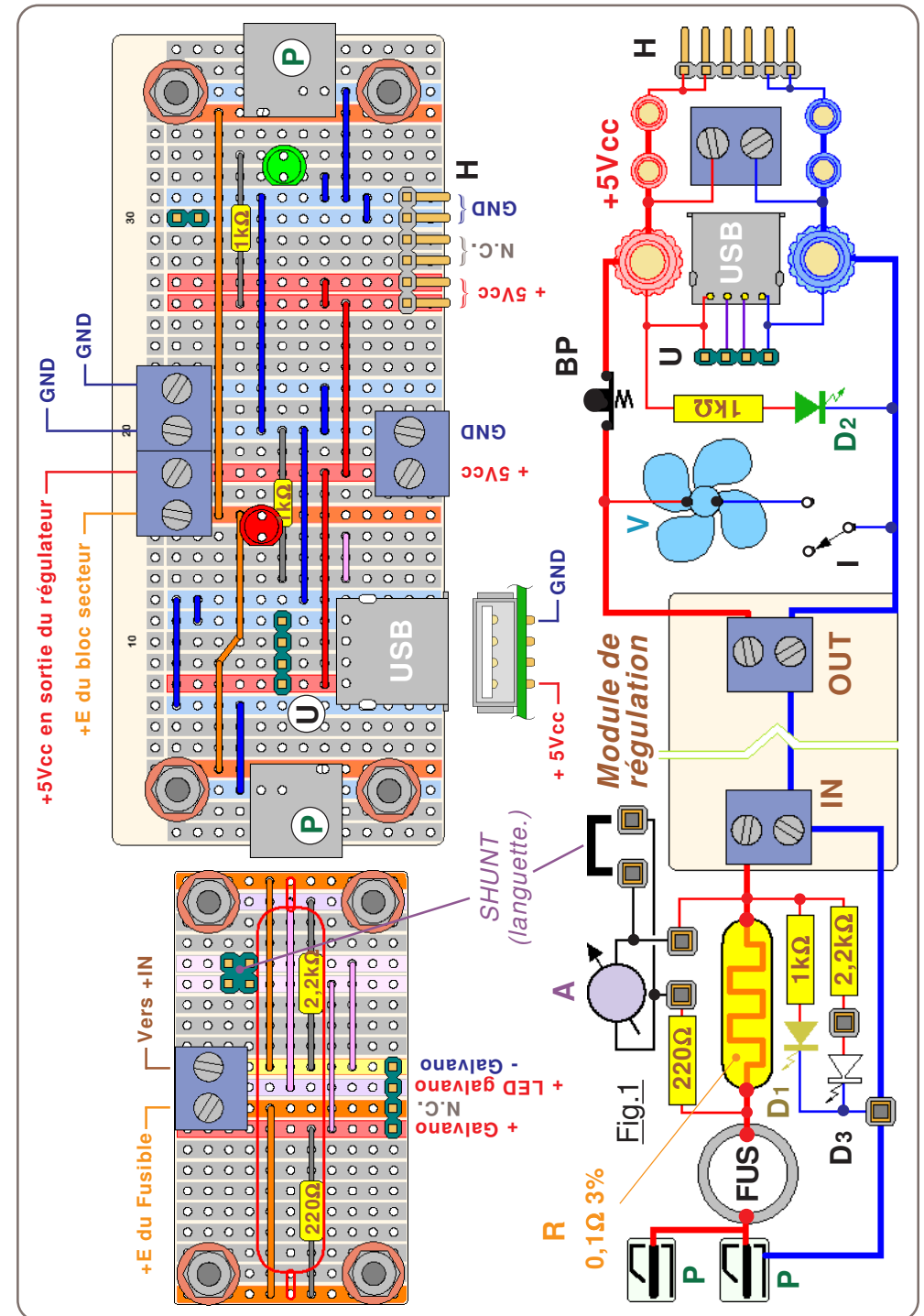
### ► Comportement du galvanomètre.

Compact et rétroéclairé, il présente toutefois l'inconvénient de commencer les graduations non pas à zéro mais à **1**, compliquant de ce fait l'interprétation des affichages. La Fig.2 présente l'étalonnage, le graphe de la Fig.3 précisant qu'à partir de **4** il faut ventiler.

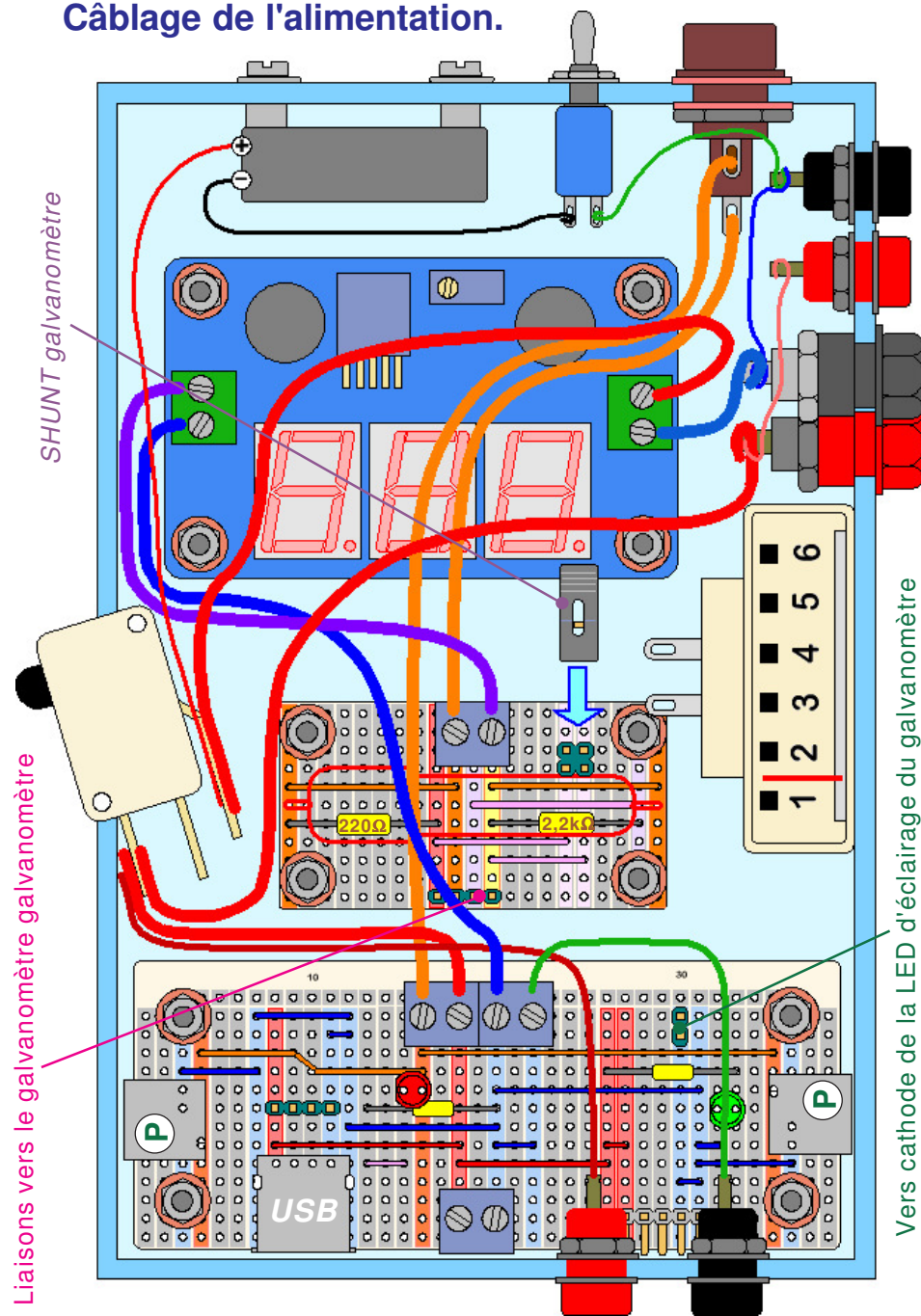
Fig.2



Fiche n° 1



## Câblage de l'alimentation.



Fiche n° 2

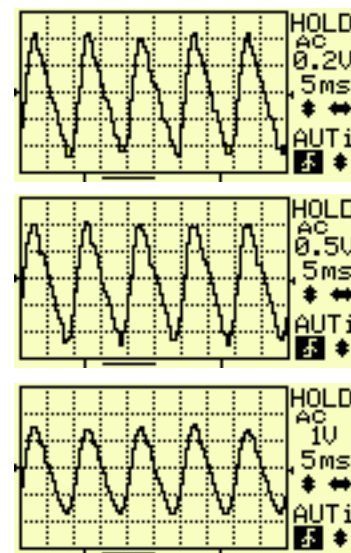
## Petite alimentation 5Vcc - 2.

### ► Comportement du module régulateur de tension.

Jusqu'à un débit de 750mA il n'est pas utile de ventiler. Dès que l'indication d'intensité dépasse la graduation **4** il importe de mettre en fonctionnement le petit ventilateur. Les indications de tension d'entrée et de sortie sur l'afficheur ne sont pas très précises. Le tableau donné ci-dessous précise les valeurs réelles en fonction de celles affichées ainsi que les conditions de fonctionnement.

V <sub>IN</sub>	Aff.	I	V <sub>OUT</sub>	Aff.
11V	11,2	à vide	5,05v	4.8
10,2V	9,7 à 10,2	500mA	5v	4.8
10V	9,9 à 10,3	400mA	5v	4.8
9V	9,2 à 9,7	700mA	5v	4.7
9,6V	8,9 à 9,7	750mA	4.98v	4.8
9V	8,2 à 9,1	1A	4.85v	4.8
7,95V	6,5 à 7,7	1,5A	4.7v	4.4
7,2V	6,2 à 7,7	1,7A	4.5v	4.5

On observe que la tension affichée en entrées du régulateur fluctue rapidement. C'est assez normal car le bloc basse tension secteur présente une filtration sommaire. Les oscillogrammes de la Fig.4



ont été saisis en mode alternatif. Celui montré en **A** présente un RAC résiduel de 0,9V crête à crête environ pour une intensité consommée sur **OUT** de 750mA. On constate en **B** que pour une intensité débitée de 1A la composante alternative avoisine 2,5V crête à crête. Sur le tracé **C** la tension alternative arrive à 3V crête à crête pour une intensité débitée de 1,5A sur **OUT**. La marge de tension différentielle entre la sortie et l'entrée du régulateur commence à devenir faible quand la tension redressée passe par sa valeur minimale.



### Petite alimentation 5Vcc - 3.

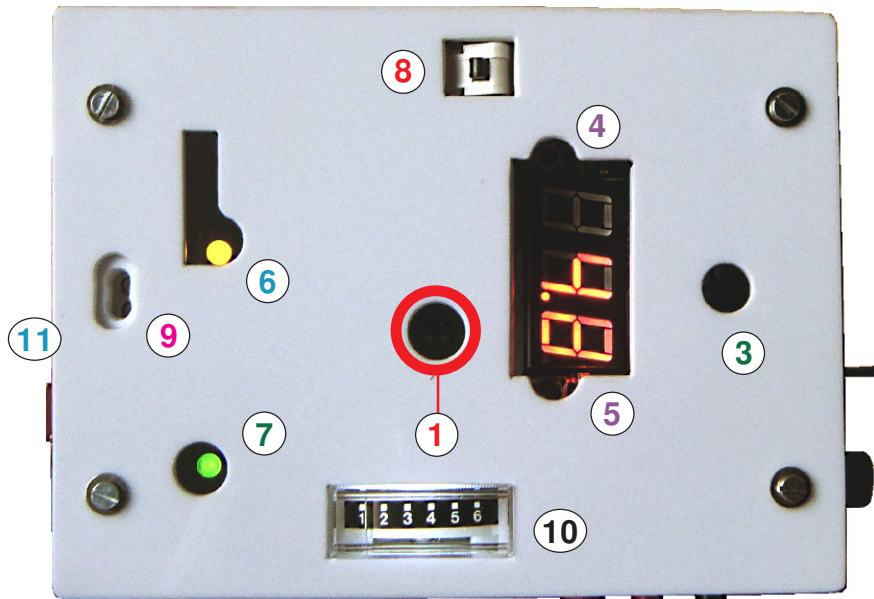


Fig.1

➤ **Mise en œuvre.**

Il est possible de faire débiter cette alimentation à 1,7A en permanence à condition d'activer impérativement le petit ventilateur. Le galvanomètre **10** est alors en surcharge. Sa déviation exagérée sur le long terme pourrait en diminuer la fiabilité. Par ailleurs, certains moteurs peuvent engendrer des pics d'intensité à leur démarrage qui souquent énergiquement le cadre mobile, brutalité qui ne va pas dans le sens de la longévité. Pour parer ces deux cas de figure relativement peu fréquents, la possibilité de placer **10** en court-circuit est prévue sur le circuit imprimé sous la forme d'un connecteur HE14 double et de grande hauteur. Ainsi, par le truchement d'une petite languette en **1** il devient facile d'établir le contact ou au contraire de le supprimer. Le connecteur HE14 est utilisé pour avoir les broches proches du couvercle et ainsi insérer facilement la languette de court-circuit coffret fermé.

Autre risque potentiel : Les broches du petit connecteur HE14 de sortie sur le côté droit en **11** dépassent légèrement du coffret. De ce fait un risque de contact accidentel avec l'environnement ou d'un

court-circuit malencontreux devient trop important si elles ne sont pas utilisées. En **2** un bouchon isolant pare ce risque.

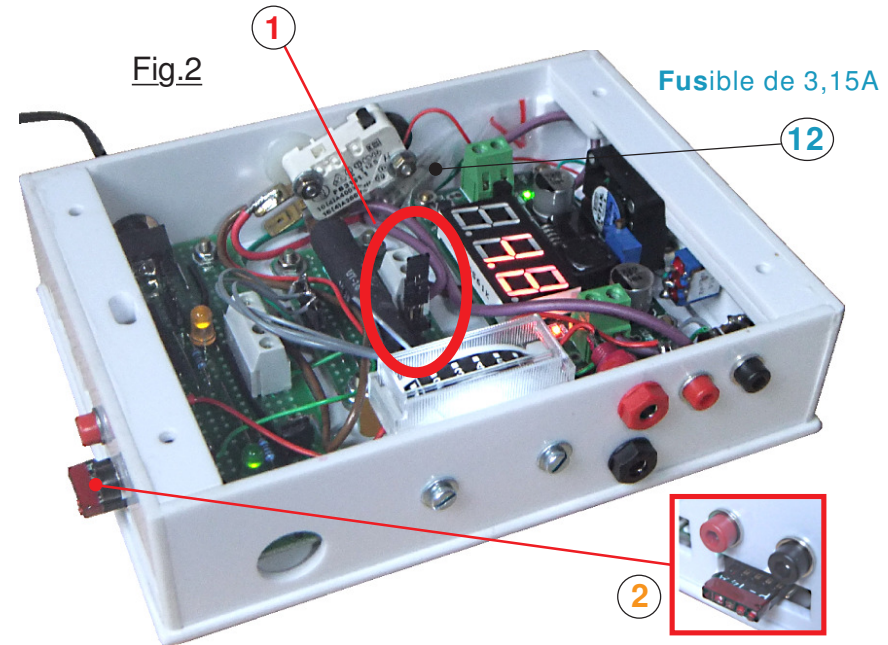
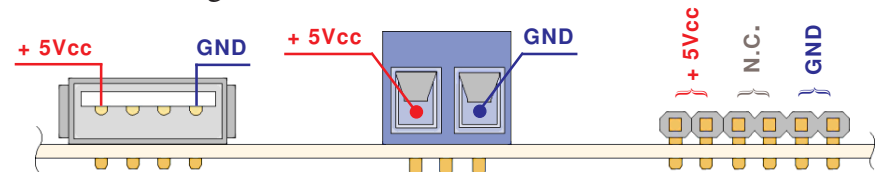


Fig.2

Fusible de 3,15A

En **3** on peut ajuster la tension de sortie avec un petit tournevis idoine. Un quelconque stylet permet d'allumer ou d'éteindre l'afficheur en **4** et de choisir la valeur de la tension d'entrée ou de sortie indiquée en **5**. En **6** la lumière est assez grande pour permettre la sortie des fils qui seraient reliés au connecteur USB et pour voir la LED témoin de la présence de tension en entrée du régulateur. L'orifice **7** laisse voir la LED témoin de la tension en sortie qui s'éteint quand on active en **8** le "bouton de panique". Enfin en **9** il sera facile avec un petit tournevis de brancher des fils de section suffisante sur le bornier de sortie disponible sur le côté gauche.

Pas très visible sur la Fig.2 on peut observer en **12** un petit sachet en matière thermoplastique transparente dans lequel est disponible un fusible de rechange calibré en nominal à 3,15A.



## Méthodes de la bibliothèque Adafruit\_PWMServoDriver.h.

**S**pécifique au multiplexeur PCA9685 à 16 canaux cette bibliothèque permet facilement de gérer chaque moteur indépendamment l'un de l'autre. L'effet d'une commande reste effectif jusqu'à une nouvelle consigne pour le canal concerné. Toute utilisation de cette bibliothèque doit être précédée de la déclaration :

```
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
```

### Fréquence du signal généré sur les 16 canaux.

```
pwm.begin();
```

Cette instruction doit être placée en premier dans **void setup()** et permet de générer une instance du module PCA9685.

### Fréquence du signal généré sur les 16 canaux.

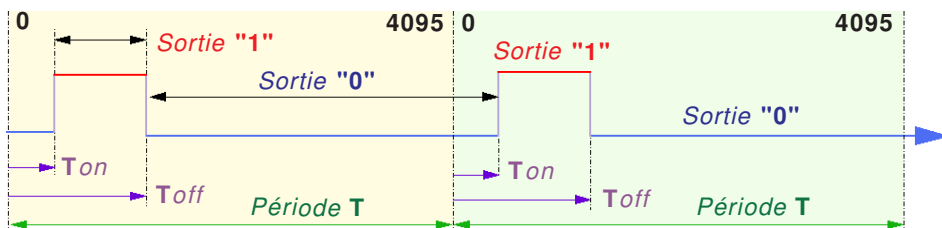
```
pwm.setPWMFreq(Fréquence);
```

Cette instruction doit être placée dans **void setup()** et définit la fréquence des signaux PWM générés sur les sorties. La valeur sera comprise entre 40 et 1000, mais en standard adopter 50Hz.

(La période **T** fait alors 20mS pour des servomoteurs.)

### Principe de la génération PWM.

La génération PWM est basée sur un compteur à 12 BITS qui fonctionne à une cadence définie par **pwm.setPWMFreq()** et qui



en détermine la période **T**. Le signal présentera une période identique sur toutes les sorties d'un même module. L'instruction qui conditionne une sortie contient deux paramètres **Ton** et **Toff** qui permettent à notre guise de définir la durée à l'état "1". Ce sont les valeurs du compteur qui déclenchent les changements d'état :

**Sortie "1"** = **Toff - Ton**.

**Sortie "0"** = **Période T - Sortie "1"**.

Le rapport cyclique et la durée des deux états est donc directement fonction de la fréquence de répétition programmée.

## Méthodes de la bibliothèque Adafruit\_PWMServoDriver.h.

**F**ondamentalement le module PCA9685 est créé pour générer de la PWM dans des applications quelconques. C'est la raison pour laquelle on peut librement définir la fréquence de répétition et le rapport cyclique. On peut par exemple piloter "analogiquement" des éclairages, des résistances chauffantes etc. *Les exemples et la formule donnée ci-dessous sont donc restrictives à une période imposée "arbitrairement" à 20mS pour piloter des servomoteurs.*

### Largeur de l'impulsion "positive" générée.

```
pwm.setPWM(Num_sortie, Ton, Toff)
```

Cette instruction engendre sur la sortie **Num\_moteur** un signal

PWM à la **Fréquence** prédéfinie dont la durée de l'état "1" est définie par les valeurs **Toff - Ton** sur une "portée" 4095.

**Pour s'affranchir de la différence, Ton sera généralement égal à 0.**

Compteur = 4095 pour 20mS soit 20000µS.

On a un comptage pour 20000 / 4095 soit toutes les 4,884µS.

**Toff à indiquer = Sortie "1" / 4,884** (Sortie "1" désirée en µS.)

*Exemple : Sortie "1" désirée = 2272µS.*

*Toff = 2272 / 4,884 = 465 ⇒ pwm.setPWM(Num\_Sortie,0,465);*

### Pilotage en degrés des servomoteurs.

Chaque moteur peut en principe balayer un angle de 180°. Toutefois la dispersion de caractéristique fait que d'un moteur à l'autre la position adoptée n'est pas rigoureusement identique pour un même signal. Par exemple le moteur testé présente les limites suivantes :

-90° pour 2272µS soit **Sup** = 465 valeur nommée **Tmini**.

+90° pour 667µS soit **Sup** = 136 valeur nommée **Tmaxi**.

L'instruction qui transpose l'angle désiré en durée PWM est :

**Pulse = map(Angle, -90, +90, Tmini, Tmaxi);**

**Ligne I2C et adressage du module :** La bibliothèque impose pour la ligne I2C le signal **SDA** sur **A4** et le signal **SLC** sur **A5**. L'adressage par défaut sur le matériel est en 0x40. On peut à la demande, ce qui sera obligatoire si on chaîne plusieurs modules, modifier l'adresse matériellement par établissement de contacts sur des ponts disponibles sur le circuit imprimé. (Voir documentation.)

## Protocoles de dialogues avec la sonde.

**M**ode alternat de type Maître/Esclave. La sonde est en écoute jusqu'à réception d'une consigne. Après analyse syntaxique elle accuse réception. Si l'ordre est cohérent elle réalise l'action souhaitée. Sur erreur de réception elle repasse immédiatement en écoute. Deux types de consignes sont envisagés :

- L'appel d'un programme.
- Une consigne de rotation sur un moteur individuel.

*Quel que soit le type de consigne les messages doivent se terminer par le caractère "\*" de fin de transmission en voie montante.*

Démonstrateur : `P04_Piloter_un_moteur.ino`

### ➤ Appel d'un programme.

Les messages présentent tous une longueur de quatre caractères commencent par "p" et sont de la forme "pNN\*" ou NN est le numéro de programme à invoquer **codé obligatoirement sur deux caractères**.

### ➤ Consigne de position pour un moteur.

Elles sont de la forme : "mN±PP\*". Le caractère "m" précise qu'il s'agit d'une consigne de motorisation. N désigne le numéro du moteur et sera compris entre [1 et 12]. **Attention, ne pas mettre un zéro en tête**. Le numéro de sortie sera obligatoirement suivi du signe "-" ou "+" pour la position, y compris pour le neutre opérationnel zéro. Enfin PP précise la position angulaire sur un ou deux chiffres la valeur étant comprise entre [0 et 90].

💀 **ATTENTION : Un zéro en tête pour désigner la sortie moteur génère un angle décodé égal au n° N.** 💣

Exemple : "m02+45\*" sera décodé Sortie n°1 Angle 2 !ERR 5!

Erreur	Nature de l'erreur de syntaxe.
1	Longueur du message inférieure à 4 caractères.
2	Chaîne non terminée par "*".
3	Le message ne commence pas par "p".
4	Le message ne commence pas par "m".
5	Signe "-" ou signe "+" attendu.
6	Programme invoqué 00 ou supérieur au nombre possible.
7	N° de moteur supérieure à 11 ou égal à 0.
8	Angle de position supérieur à +90°.
9	Angle de position inférieur à -90°.

## Pilotage individuel des moteurs avec le potentiomètre.

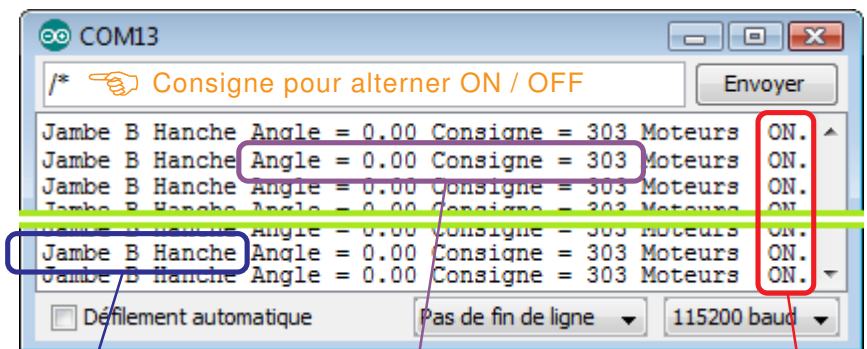
**C**ommode en cours d'élaboration des programmes, c'est `P05_Piloter_au_potentiometre.ino` qui permet d'agir individuellement sur chaque servomoteur. Le dialogue se fait par l'intermédiaire de la ligne série USB du Moniteur de l'IDE.

### ➤ Consigne de position pour un moteur.

Elles sont de la forme : "N\*". N désigne le numéro du moteur et sera compris entre [1 et 12]. Les zéros en tête sont tolérés. On peut à tout moment configurer la motorisation en ON / OFF. La commande "/" se comporte comme une bascule. Sur OFF les affichages continuent pour permettre l'ajustement du potentiomètre sans incidence sur les mouvements effectif. Basculer sur ON engendre immédiatement le positionnement du servomoteur concerné. Toutes consignes différentes de celles précisées ci-dessus engendreront un BIP sonore d'alerte pour l'opérateur.

*Le protocole pour agir sur un moteur est le suivant :*

- 1) Consigne "/" pour imposer le mode OFF,
- 2) Centrer le bouton du potentiomètre pour afficher l'angle 0.00°,
- 3) Consigne "N\*" pour valider un servomoteur individuel,
- 4) Consigne "/" pour rétablir le mode ON, le moteur concerné va immédiatement se positionner au "neutre opérationnel",
- 5) Tourner le bouton du potentiomètre pour positionner le moteur.



Articulation pilotée    Consignes "potentiométriques"    État actuel

**ATTENTION :** Les limites de consigne sont étendues et les moteurs peuvent parfaitement diverger en rotation continues. **Surveiller attentivement quand les valeurs angulaires approchent de +90°.**



### Pilotage des servomoteurs.

Actuellement tous les servomoteurs utilisés en petite robotique fonctionnent sur le même principe. Bien que les fournisseurs ne respectent pas forcément ce qui semble devenir une norme de fait, globalement tous s'approchent de ce qui devient un standard. Ces moteurs intégrant une électronique d'interfaçage et un capteur de position se raccordent avec seulement trois fils :

- Le fil noir ou **marron** va à la masse **GND**.
- Le fil central **rouge** va au **+5Vcc**. (1)
- Le fil blanc ou **orange** reçoit le signal de pilotage.

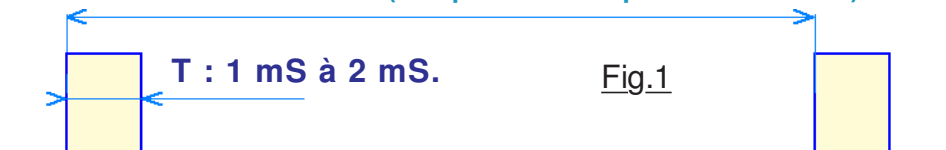
Comme montré sur la Fig.1 la fréquence de répétition standard choisie pour leur pilotage est de 50 Hz, soit une période de 20 mS. C'est la durée **T** de l'impulsion à l'état "1" qui positionne le moteur en absolu par rapport à son amplitude possible de rotation. Mis à part les moteurs sans limitation de rotation angulaire, les plages de débattement angulaire les plus courantes avoisinent les 180°.



#### **En standard :**

- $T = 1 \text{ mS}$  : Le moteur se place en position minimale.
- $T = 1,5 \text{ mS}$  : Le moteur se place en position neutre. (2)
- $T = 2 \text{ mS}$  : Le moteur se place en position maximale.

Période : 20 mS MAX (Fréquence de répétition :  $\approx 50\text{Hz}$ )



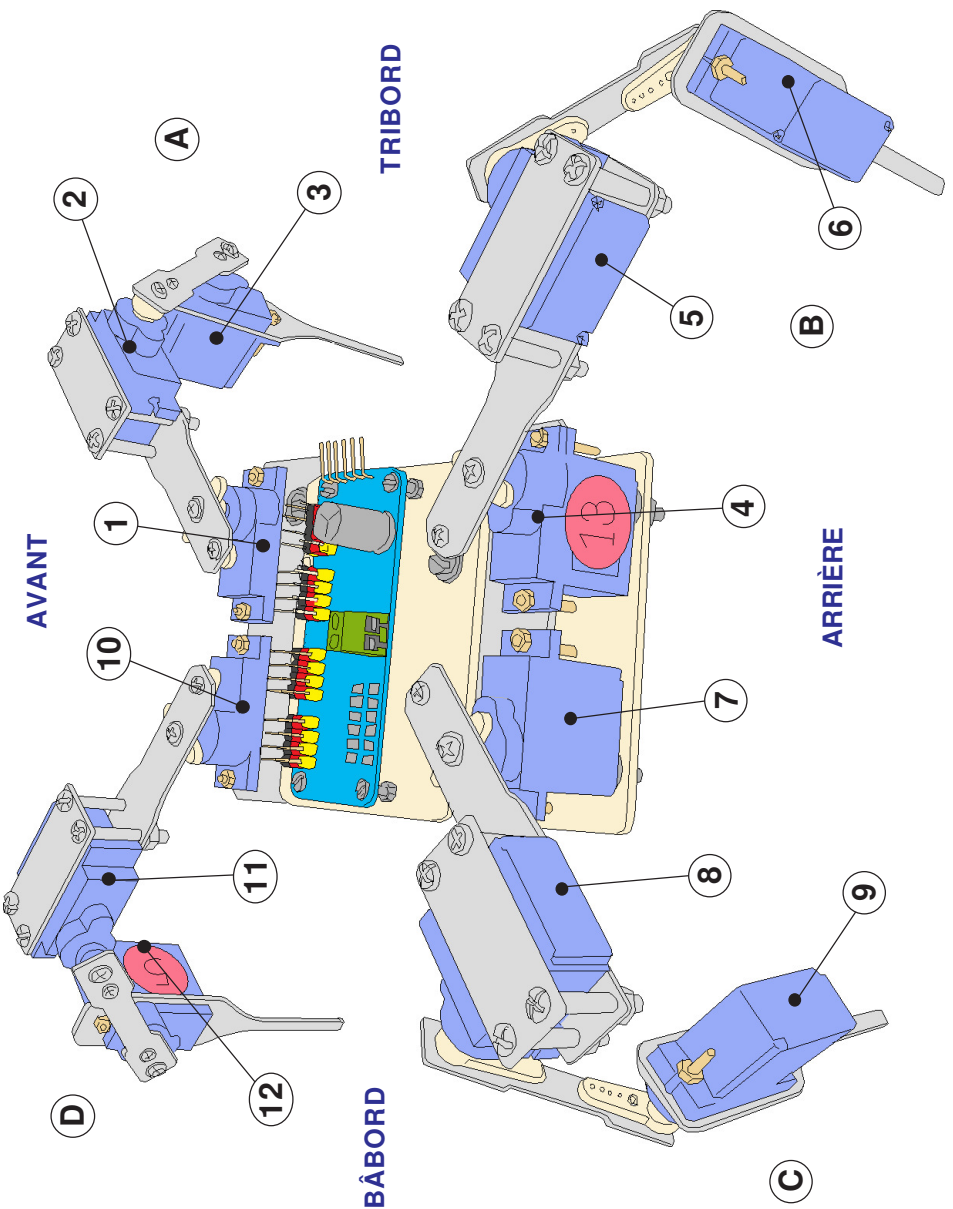
*Attention : La proportion temporelle n'est pas respectée.*

(1) Les moteurs à courant continu génèrent des appels de courant importants quand ils démarrent. Chaque surintensité transitoire corrompt la tension d'alimentation. Pour qu'Arduino ne soit pas perturbé, il faut impérativement qu'il reçoive son **+5Vcc** d'une **alimentation indépendante de celle des moteurs**. En revanche **GND est commun aux deux sources d'alimentation**.

(2) La position neutre correspond à la position moyenne qui pour la course de 180° se trouve au centre à 90°.

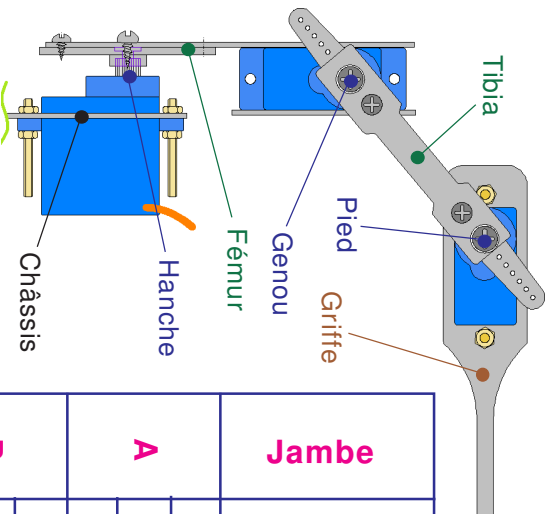
### Notes personnelles.

## Répartition des moteurs sur l'interface.



Fiche n° 7

## Repérage et caractéristiques des articulations.



Jambe robotisée.

Repéré en vert  
les membres  
principaux.

Repéré en bleu  
les articulations.

Jambe	Articulation	Traçabilité	Sortie	Butée -	Centrage	Butée +	Moteur
A	Hanche						1
	Genou						2
	Pied						3
B	Hanche						4
	Genou						5
	Pied						6
C	Hanche						7
	Genou						8
	Pied						9
D	Hanche						10
	Genou						11
	Pied						12

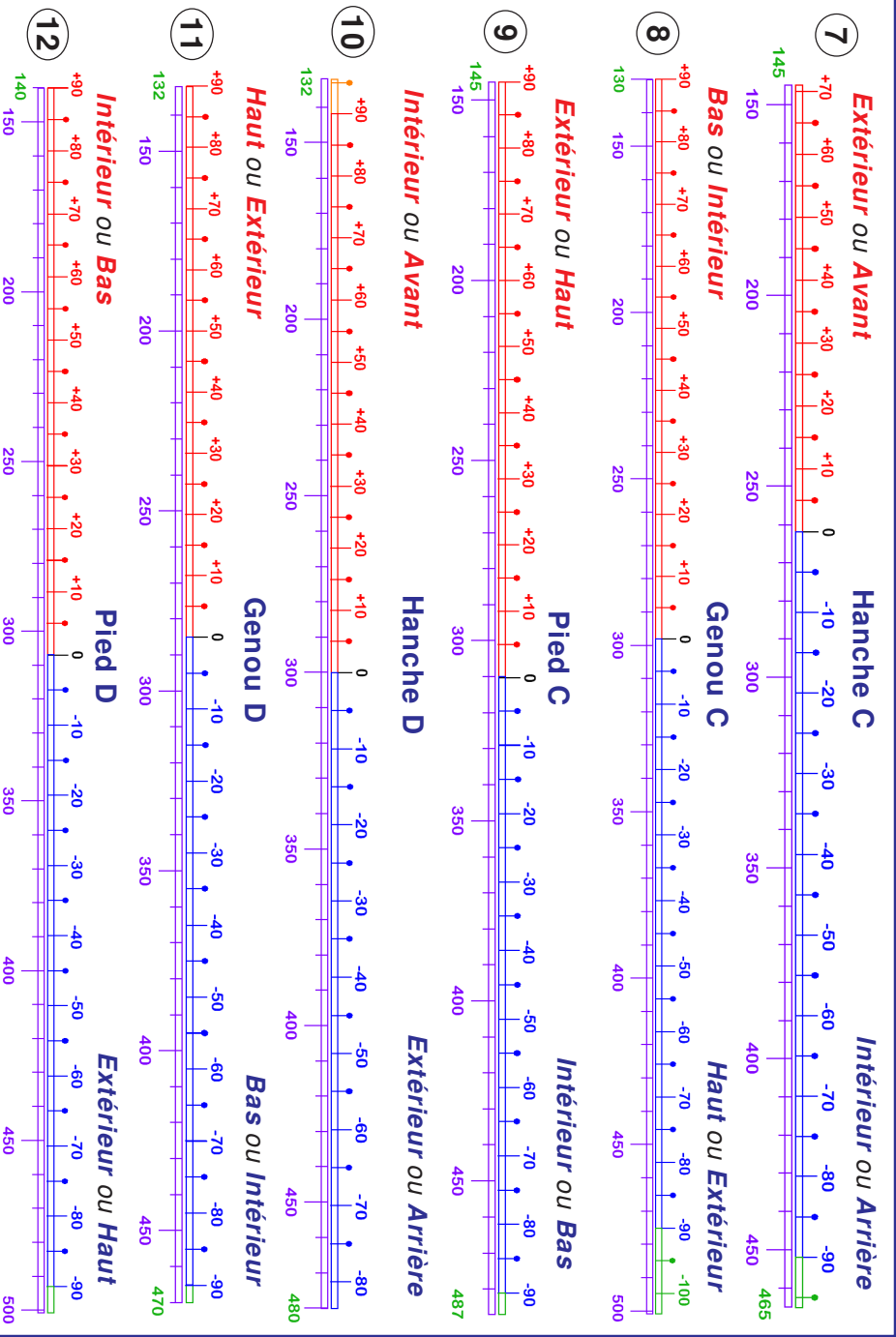
[illegible]

Fiche n° 8

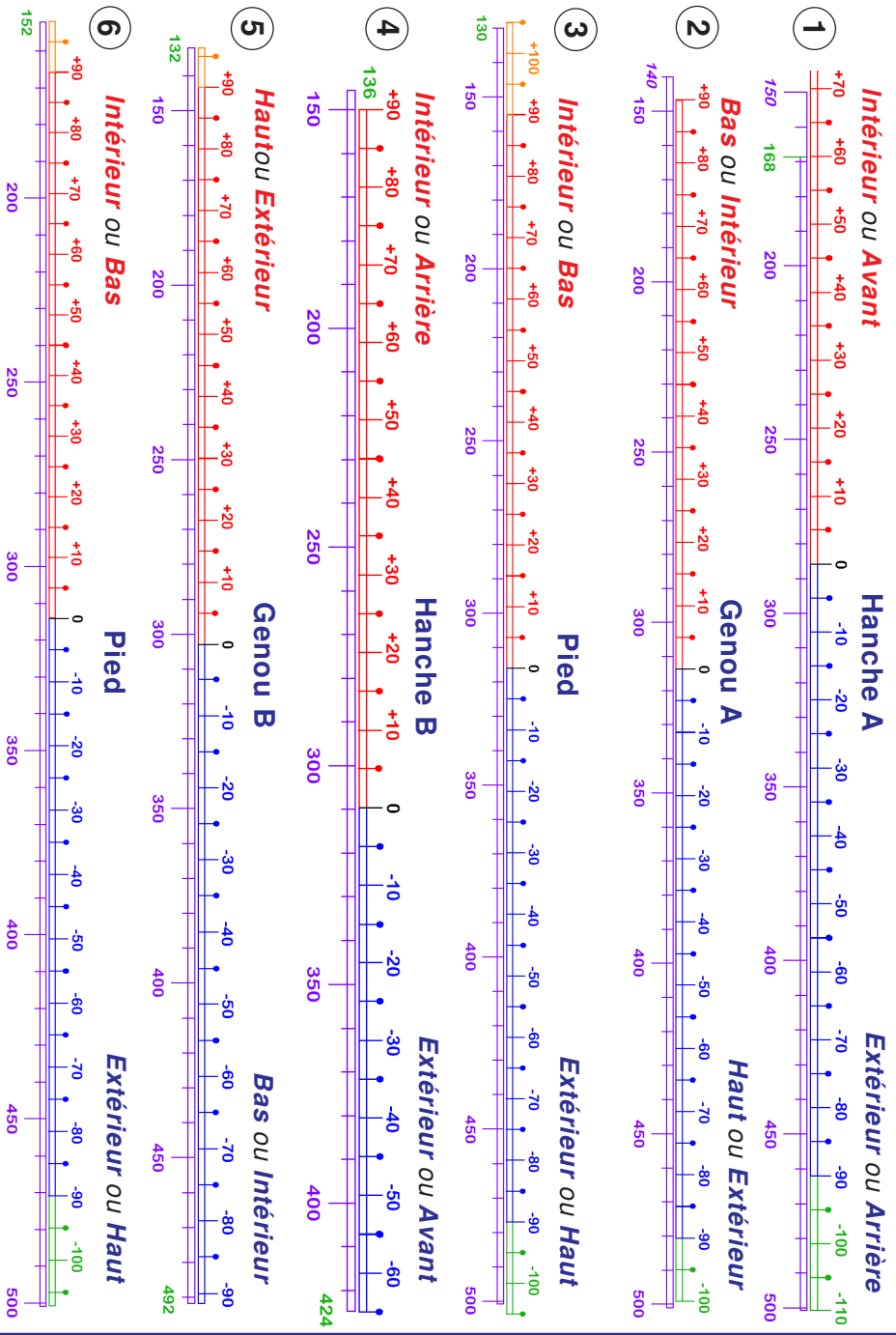
[illegible]







01.04.21



## Liste des **ERREURS** en exploitation.

### **ERR Nature de l'erreur de syntaxe.**

- 1 Le message ne commence pas par "p".
- 2 Message non terminé par "\*\*".
- 3 Une consigne sur trois caractères est invalide.
- 4 Consigne à plus de quatre caractères invalide.
- 5 Lettre de répétition de mouvement incorrecte.
- 6 Le caractère "C\*" n'est pas valide.
- 7 "pNN\*" avec 00 ou supérieur au nombre possible.
- 8 Pas de programme enregistré en mémoire EEPROM.
- 9 PGM présent en EEPROM. (*L'effacer avec "y\*"*)
- 10 Tentative d'effacer un programme en EEPROM alors que la double sécurité "Phares allumés" et ACR validé n'est pas satisfaite.
- 11 Saturation de la mémoire d'enregistrement.
- 12 Bouclier non posé sur le sol martien. (*Ou sur le berceau dans le lanceur Ariane.*)
- 13 "h\*" ou "v\*" alors que le LASER n'est pas actif.
- 14 "h\*" ou "v\*" sonde non "Configurée LASER".
- 15 "p03\*" avec un obstacle à moins de 8cm.
- 16 Capteur Humidité : Temps d'attente dépassé.
- 17 Capteur Humidité : Erreur de "Checksum".
- 18 "p9c\*" : Caractère c pour désigner le moteur incorrect.
- 19 "p09\*" : Pas de caractère c pour désigner le moteur.
- 20 Tentative d'enregistrer une posture en EEPROM alors que la double sécurité "Phares allumés" et ACR validé n'est pas satisfaite.
- 21 Invoquer un programme alors que le programme préalable n'est pas correct. (*Voir la liste des conditions préalables sur la fiche : [Liste des PROGRAMMES.](#)*)
- 22 Tension motorisation inférieure à 4.0Vcc.

### ----- Problèmes sur la centrale gravitationnelle -----

- 30 Échec de connexion avec le MPU6050.
- 32 Débordement FIFO en dialogue "Inertiel".
- 33 Précédé de 1 : Échec de chargement initial.  
Précédé de 2 : Échec MAJ Config du MPU.

## Quelques détails géométriques.

Genou	A	B	C	D
Moteur	2	5	8	11
45°	213	406	215	388
Vertical	152	486	139	463
Hanches	A	B	C	D
Transversal	287	335	222	347

## **Justification des butées logicielles**

