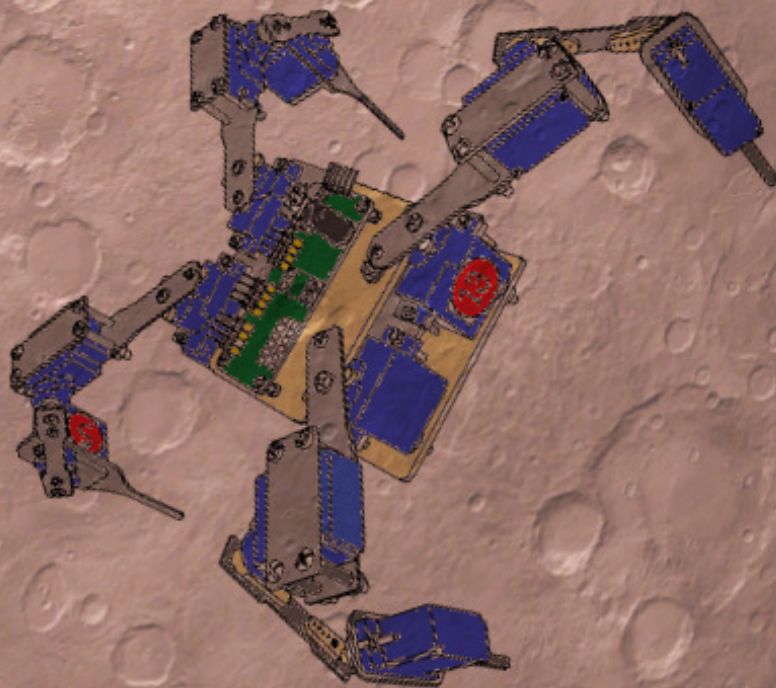


# Sonde martienne JEKERT

## Les expériences scientifiques embarquées.

### TOME 3



Landing Site



Par Nulentout : Mercredi 6 Décembre 2017.

# Développement du projet Sonde martienne JEKERT

Une sonde spatiale non équipée d'expériences scientifiques constitue une aberration technique, une merveille de technologie stérile. Compte tenu du coût faramineux engagé pour développer cette dernière, sans compter la fusée de lancement en orbite terrestre et en éjection interplanétaire qui sera définitivement perdue, il est totalement impensable de n'envoyer qu'une touriste, une promeneuse qui n'aura pour toute activité que d'avoir à se déplacer sans but précis.

Ce tome va se préoccuper de transformer JEKERT en une vraie prospectrice truffée de capteurs sophistiqués capable de prendre en compte son environnement, et ainsi mériter son titre d'exploratrice. Nous avons commencé en premier par concrétiser le vecteur de transport. Dans la réalité astronautique c'est toujours l'inverse qui se produit. De nombreux chercheurs de haute voltige inventent des concepts et font des propositions de recherche expérimentale au comité chargé d'élaborer les grandes lignes d'un nouveau projet scientifique. Ensuite, les budgets étant forcément limités, on élabore une combinaison qui semble la plus prometteuse pour faire évoluer nos connaissances du système solaire. La liste des expériences embarquées étant provisoirement définie, une première étude vérifie que leur promiscuité est possible, qu'il n'y aura pas des interactions réciproques rendant certaines inutilisables. Leur répartition géographique sur une structure particulièrement élaborée pour minimiser les nuisances réciproques, on peut alors débiter les études détaillées du moyen de transport.

Dans ce didacticiel nous avons mis la charrue avant les bœufs. Bref, nous ne serions pas des exemples à suivre dans le monde du spatial réel ! Il est grand temps de corriger cette "grossière erreur". Du coup, ce sont les expériences scientifiques très complexes qui vont devoir s'adapter à leur moyen de locomotion et pas l'inverse. Les faibles dimensions de JEKERT et la fragilité de son ossature vont influencer le "possible" et compliquer parfois singulièrement le réalisable. Peu importe, le but de ce projet réside dans la recherche, alors l'ordre dans lequel elle est conduite n'est pas primordial. Il est d'ores et déjà possible d'annoncer les caractéristiques de la sonde quand elle sera totalement achevée et que le moment sera venu de la hisser en haut du lanceur Ariane :

## Fiche signalétique de JEKERT

- **Masse à vide : 240g** intégrant :
  - \* Le châssis et les douze moteurs,
  - \* Le bouclier et les quatre boulons longs de liaison,
  - \* La plaque de support du multiplexeur,
  - \* Le multiplexeur,
  - \* Une petite carte Arduino NANO.
- **Circuit imprimé principal : 50g à 60g** comprenant :
  - \* Le circuit imprimé et tous ses composants y compris le petit C.I. de complément à l'avant,
  - \* La carte Arduino NANO sur le connecteur HE14 ainsi que le carton d'isolement.
- **Circuit imprimé du condensateur de 470 $\mu$ F : 90g** muni du capteur inertiel gyroscopique.
- **Masse totale en charge : 370g** avec tous les capteurs, les quatre chaussettes, ainsi que la charge du cordon ombilical de dialogue série et celui amenant l'énergie aux servomoteurs.

## ➤ Bilan énergétique.

- **Module Arduino 58mA** prenant en compte : (*Alimentation du circuit principal en +5Vcc.*)
  - \* La carte Arduino NANO avec la LED Moteurs OFF, la LED du répéteur potentiométrique,
  - \* Les phares au maximum d'éclairage,
  - \* Le LASER au maximum de luminosité.

Sans le LASER : 40mA. Sans le LASER et sans les Phares : 30mA.

**Intensité maximale** l'intégralité des consommateurs étant alimentée : **80 à 90mA.**



➤ Elle est petite ... mais elle a tout d'une grande !

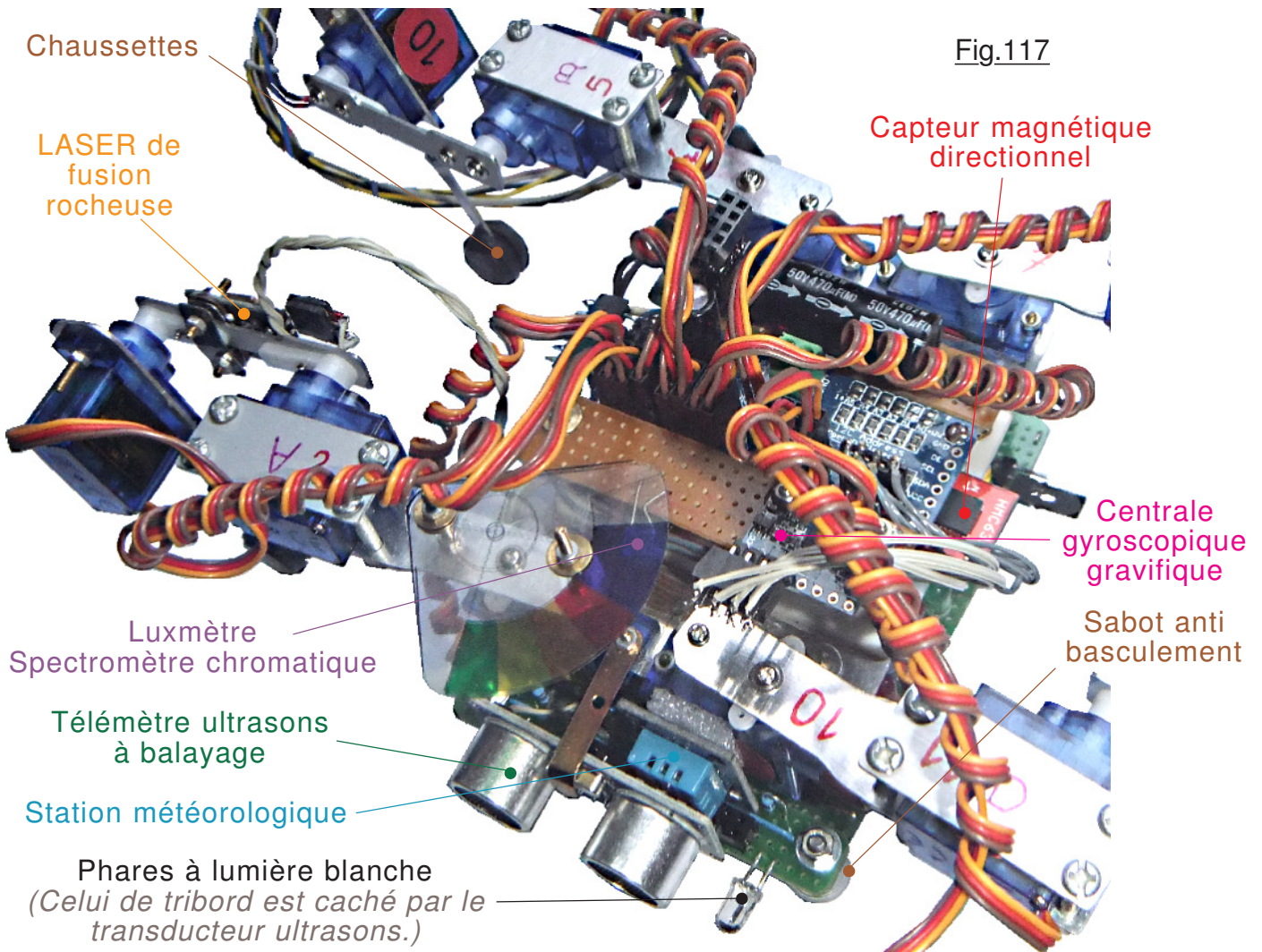


Fig.117

### Répartition des expériences scientifiques sur la sonde.

#### 25) 22/11/2017 : Le luxmètre (MJD 58079)

B ranlebas de combat en salle S10, une forteresse de propreté à l'image de la salle S6 dans laquelle JEKERT a été replacée sur son berceau pour ménager l'ensemble mécanique. Cette pièce entièrement étanche, d'une pureté "chirurgicale" jouxte S6 pour pouvoir transiter facilement de l'une à l'autre. Le professeur Mandex assiste les ingénieurs chargés de la mise en œuvre du luxmètre, un appareil scientifique chargé d'analyser le rayonnement solaire à travers l'atmosphère Martienne. C'est la première expérience embarquée qui sera intégrée sur l'insecte mécanique.

- *Dis Totoche, c'est quoi un luxmètre, un truc pour évaluer le luxe et la richesse du citoyen et ainsi déterminer son imposition sur la fortune ?*
- *Ben non Dudule, c'est une technologie pour mesurer la luminosité. Comme les unités dans ce domaine peuvent être des "Lux", le nom donné à l'appareil en dérive.*

D ans les exposés qui vont suivre, nous allons mener en parallèle l'étude des c électroniques ainsi que les logiciels associés. Pour des modules comme les informations techniques des fournisseurs seront filtrées et résumées dans des fiches dédiées pour lesquelles seules les caractéristiques indispensables seront présentes. Ce sont des résumés très condensés. Je vous conseille de les imprimer et de les avoir à disposition sur le plan de travail chaque fois qu'un démonstrateur sera téléversé sur Arduino NANO. Commencez par la [Fiche n°17](#) propose au recto le schéma électronique complet. On retrouve le microcontact bouclier, le transistor NPN du disjoncteur @ et diverses LEDs de visualisation opti de l'état de la sonde. Il est évident qu'à ce stade de l'exposé tout n'est pas

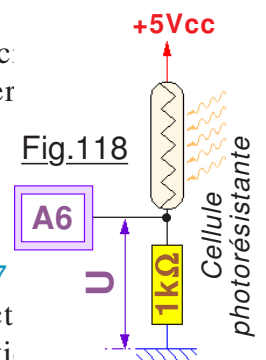


Fig.118

compréhensible. Les chapitres qui suivent vont clarifier la situation. Le verso de cette fiche anticipe sur le **TOME 4** consacré à l'intégration des systèmes, nous y reviendrons donc en temps voulu. Pour l'heure, écoutons le Professeur Mandex exposer la théorie qui préside à la présence de ce matériel météorologique sur la sonde. Le schéma électronique de cet instrument de mesure d'une complexité phénoménale est proposé en Fig.118 et doit être rapproché au montage de la Fig.110 dont il copie scandaleusement la structure. On retrouve un diviseur de tension constitué d'une résistance fixe de  $1k\Omega$  et du capteur sensible à la lumière constitué d'une photorésistance. Ce type de composant montré sur la Fig.119 est disponible à profusion sur Internet à des prix de vente dérisoires. Peu importe ses caractéristiques

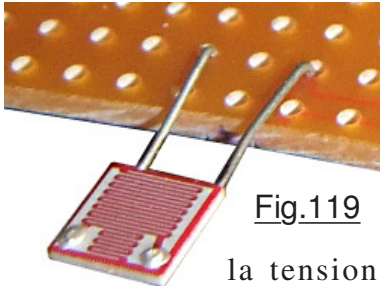


Fig.119

précises, car dans l'optique d'une approche résolument simple, la sonde se contentera de retourner la valeur numérisée sur l'entrée analogique **A6**. Pour minimiser le code généré par le compilateur, des formules complexes transposant la valeur entière en donnée exprimée dans les unités du système S.I. ont été écartées. Ce type de cellule photoélectrique voit sa résistance électrique diminuer quand la quantité de lumière reçue augmente. Ainsi placée dans la branche **+5Vcc / GND**

la tension mesurée **U** variera dans le sens de l'énergie reçue. Quand vous approvisionnez le composant photorésistant, tout modèle conviendra, qu'il soit rond ou carré peu importe. Ses caractéristiques seront voisines. Certainement qu'au lieu de retourner 652 valeur issue de la cellule du prototype vous obtiendrez 589 ou 698. Peu importe à partir du moment où l'on ne cherche pas précisément à calibrer l'instrument. Dans tous les cas, les ACR retournés par la sonde, comme visible sur la Fig.120 seront représentatif de la clarté régnant dans l'environnement de JEKERT.

Fig.120

```
m* >
Luxmètre : 652
```

Constituant un paramètre météorologique, cette valeur sera récupérée depuis la sonde par le canal des télémesures. Une instruction à un caractère "**m\***" sollicitera le logiciel pour qu'il nous retourne l'ensemble des facteurs météorologiques appréhendés par la petite station équipant JEKERT. Cette expérience scientifique embarquée est susceptible d'établir les facteurs d'assombrissement atmosphériques en fonction des autres paramètres météorologiques. En particulier ils permettent également d'estimer la pollution environnante résultant de l'arrivée d'une tempête de sable, phénomène banal sur la planète rouge. Il sera ainsi possible aux techniciens de maîtrise d'anticiper l'arrivée de ce genre de perturbation et de configurer la machine en posture VEILLE et en économie d'énergie. On se doute que sur un explorateur réel, des obturateurs sont prévus pour protéger les optiques et les instruments délicats des agressions poussiéreuses ...

Ayant installé la cellule photorésistante sur une petite plaque d'expérimentation et correctement branchée sur **+5Vcc / GND** et sur l'entrée **A6** sans oublier la résistance de  $1k\Omega$ , téléversez le démonstrateur **P14\_Sonde\_avec\_capteurs\_scientifiques.ino**, vous pouvez tester la nouvelle commande "**m\***". Comme pour le moment la station météorologique n'est pas encore complète, l'affichage du luxmètre sur le moniteur série de l'**IDE** sera correct mais accompagné d'une alerte sonore et d'un message "**!ERR 16!**". Si vous n'avez pas encore réceptionné le petit composant photosensible, rien n'interdit de le remplacer provisoirement par un potentiomètre dont les extrémités seront réunies à la résistance fixe de  $1k\Omega$  et au **+5Vcc**. Le curseur ira à l'entrée analogique **A6**. Ce fonctionnement en "dégradé" ne restituera pas les valeurs numériques d'une cellule photoélectrique. Vous obtiendrez toutefois des valeurs variables confirmant le bon comportement du logiciel.

**ATTENTION :** Le schéma dessiné sur la **Fiche n°17** concerne l'électronique définitive. Hors les démonstrateurs ont été écrits et mis au point bien avant que la ventilation définitive des broches du microcontrôleur ne soit déterminée. Aussi il importe de réaliser sur les petites plaques d'expérimentation des branchements conformes à ceux précisés dans les programmes sources C++ par les remarques du chapitre nommé **DIFFERENTS BRANCHEMENTS**.



*Ben môamôa je dis qu'imprimer toutes ces fiches c'est du gaspillage de papier, le luxmètre il va clignoter au rouge !*

## 26) 23/11/2017 : La petite station météorologique (MJD 58080)

Il n'est contournable sur un explorateur réel pour tenir compte de l'environnement et parer l'arrivée des tempêtes, mesurer les facteurs atmosphériques constituent pour toutes les sondes d'exploration des missions fondamentales ... à condition toutefois que l'astre visité possède une atmosphère ! Déterminer la composition moléculaire, la présence d'eau, la température constituent les fondamentaux de tout géophysicien consacrant sa carrière à ce que l'on nomme l'espace. La petite station météorologique installée sur JEKERT est élémentaire. Elle ne comporte qu'un petit module ajouté au luxmètre. Bien que simplissime elle n'en mesure pas moins et avec une précision raisonnable deux données primordiales dans l'air ambiant de la petite machine : La température et surtout l'hygrométrie. D'une banalité manifeste, vraiment très facile à se procurer dans le commerce en ligne pour des broutilles, le petit module utilisé DHT11 fait l'objet de la [Fiche n°18](#) qui pour des raisons d'économies de papier porte sur la face verso la répartition des broches correspondant à l'implantation définitive. On peut y observer que le module DHT11 ne mobilise qu'une seule broche binaire **D11** qui sera employée à la fois en entrée et en sortie.

Basés sur des phénomènes physiques complexes, les capteurs d'humidité exigent des traitements informatiques complexes pour traduire les mesures effectuées en valeur d'humidité relative par rapport à la saturation théorique de l'atmosphère terrestre. Représentant un marché de loisir considérable, les petites stations météorologiques domestiques ont envahi les chaumières. Aussi, pour gagner des parts de marché, les fournisseurs de capteurs spécifiques ont muni leurs produits d'une informatique locale les rendant presque simples à les mettre en service. Les bibliothèques spécialisées ont fait le reste. L'intégralité du traitement gérant le DHT11 se résume à peu de lignes de code source. Le code objet qui en résulte consomme 556 octets, affichages des valeurs comprises :

```
① #include <DHT11.h> // Déclarations pour le capteur d'humidité DHT11.  
② #define Broche_Com_DH11 11 // D11 : Ligne de dialogue avec le capteur DHT11.  
③ byte Humidite; // Capteur d'humidité DHT11.
```

Ultra classique, la séquence des directives et déclarations relative à un module particulier commence en ① par invoquer la bibliothèque développée pour le C++ d'Arduino. Ensuite, en ② on affecte l'identificateur **Broche\_Com\_DH11** à la broche du microcontrôleur qui sera chargée d'établir la Communication avec la logistique interne au DHT11. Dans notre cas on affecte **D11** qui seule restait disponible, car vous pouvez observer sur la [Fiche n°18](#) que l'intégralité des ressources de l'ATmega328 sont mobilisées. Enfin en ③ on réserve la variable **Humidite** codée sur un octet de type **byte**. C'est suffisant puisque le module retourne des valeurs de pourcentage d'humidité relative dont la fourchette sera forcément limitée entre [0 et 100].

Notons au passage que pour son fonctionnement interne ce type de capteur doit aussi mesurer la valeur de la température atmosphérique, car le taux de saturation en dépend. Aussi, le fabricant de ce circuit a ajouté un peu de logiciel à son composant. Outre l'humidité, le module fournit aussi la valeur de la température, et l'on fait d'une pierre deux coups.

Pour en bénéficier il importe de réserver une variable dédiée :

```
byte Temperature; // Capteur d'humidité DHT11.
```

Comme on se contentera d'un affichage sous forme d'entiers, les dixièmes de °C n'étant pas pertinents dans cette application, un **byte** est largement suffisant.

La procédure qui affiche tous les paramètres météorologiques est élémentaire :

```
void Valeurs_meteorologiques() {  
    // Ci-dessous : Simple cellule photorésistante montée en diviseur de tension.  
    ① Lumiere_ambiante = analogRead(Entree_Luxmetre);  
    ② Serial.println(); Aff_TEXTE_EEPROM(720,11); Serial.println(Lumiere_ambiante);  
    ③ Afficher_Hygrometrie_et_Temperature();  
}
```

Les deux premières lignes ① et ② mesurent la valeur de la luminosité ambiante et en affichent la grandeur sur la ligne série USB du moniteur de l'IDE. Minimisant le code objet les textes sont puisés en EEPROM. L'instruction de la ligne ③, pour des raisons de lisibilité du programme se contente d'invoquer **Afficher\_Hygrometrie\_et\_Temperature()** qui se charge d'afficher comme montré sur la [Fig.121](#) la valeur du taux d'humidité et celle de la température.

```
m* >  
Luxmetre : 241  
Humidite : 66%  
Temperature : 20°C  
!PGM m OK!
```

[Fig.121](#)



```

void Afficher_Hygrometrie_et_Temperature() {
  // >>> Lecture de la température et de l'humidité + gestion des erreurs <<<
  ① switch (readDHT11(Broche_Com_DH11, &Temperature, &Humidite)) {
  ②   case DHT11_SUCCESS:
        { // Affichage du taux d'humidité puis de la température.
  ③     Aff_TEXTE_EEPROM(969,11); Serial.print(Humidite);
  ④     Aff_TXT_EEPROM_et_CRLF(644,1); Aff_TEXTE_EEPROM(980,14);
  ⑤     Serial.print(Temperature);
  ⑥     Serial.print(char(186));
  ⑦     Aff_TXT_EEPROM_et_CRLF(804,1); break;}
  ⑧   case DHT11_TIMEOUT_ERROR : {Erreur_N_BIP(16); break;}
  ⑨   case DHT11_CHECKSUM_ERROR : Erreur_N_BIP(17);}
}

```

Le gros du travail de mesurage est effectué en ① par la fonction `readDHT11` de la bibliothèque `<DHT11.h>`. Utilisant la technique des "pointeurs" repérable par le caractère `&` elle détermine les valeurs de `Temperature` et `Humidite`. Elle affecte également un état aux trois indicateurs `DHT11_TIMEOUT_ERROR` et `DHT11_CHECKSUM_ERROR` et `DHT11_SUCCESS` qui rendent compte de l'échange de données entre Arduino et DHT11. L'instruction ② effectue l'aiguillage du traitement en fonction de ces trois variables d'état. Les lignes ③ et ④ vont générer un BIP d'alerte sonore et un message d'erreur n°16 ou n°17 si un incident de dialogue est détecté. Par exemple `"!ERR 16!"` si le petit module n'est pas présent sur la ligne D11. Si le dialogue binaire est validé, en ②, les instructions de ③ à ⑦ vont s'enchaîner pour afficher conformément à la Fig.121 les données du DHT11. L'instruction de la ligne ⑥ fait afficher le caractère '°' qui sur la ligne série USB est spécial, il faut en indiquer son code 186 dans la table des caractères du moniteur de l'IDE. Enfin, les divers textes qui sont affichés sur la ligne série USB sont issus de l'EEPROM et exploitent `Aff_TEXTE_EEPROM(PTR,LGR)` et `Aff_TXT_EEPROM_et_CRLF(PTR,LGR)`. Par exemple en ligne ⑦ l'instruction `Aff_TXT_EEPROM_et_CRLF(804,1)` affiche le "C" de "°C". Elle peut sembler exagérée juste pour afficher un caractère suivi d'un passage à la ligne. Si on la remplace par le classique `Serial.println('C')` on aboutit à exactement la même taille de code objet et d'encombrement de mémoire dynamique, d'où le codage du source pour l'homogénéité.

## 27) 24/11/2017 : Le LASER de fusion rocheuse (MJD 58081)

Depuis pratiquement trois mois, les travaux pour ajouter la salle S11 sont achevés. Cette pièce est très particulière. C'est une sorte de coffre d'une propreté totale percé d'un petit orifice de dix centimètres de diamètre. Le laser de fusion rocheuse y est installé, parfaitement dans l'axe longitudinal de l'orifice. De l'autre côté, la pièce dispose blindages en acier qui rendraient par comparaison un char d'assaut fin comme du papier chocolat. Dans l'axe des tirs lumineux, sur un bloc de béton peuvent être disposés divers blocs rocheux qui vont subir les attaques de ce dragon des temps modernes. La pièce est totalement étanche. Toute présence "animale" bloque l'énergie alimentant la puissante machine lumineuse. Divers types de roches sont testées pour déterminer les spectres lumineux dégagés lors de la fusion du minéral et calibrer les instruments de mesure. Pendant que les scientifiques procèdent à leurs ajustements de calibration, en S6 les ingénieurs installent sur JEKERT les supports qui recevront le fameux crayon lumineux. Le cahier des charges fonctionnel précise que le LASER pourra se voir orienté finement en hauteur et en largeur.

Pointer le LASER en hauteur et horizontalement avec précision constitue une pierre d'achoppement pour ce projet. la première idée qui préside pour satisfaire cette clause du cahier des charges fonctionnel consiste à placer au dessus du multiplexeur toute une mécanique incluant deux servomoteurs dont les arbres de sortie sont à angle droit.

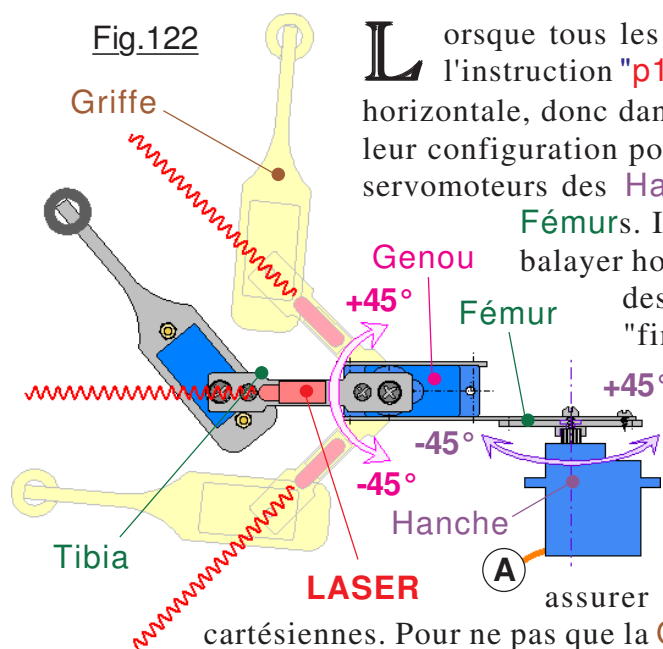
***Cette solution basique est irréaliste pour plusieurs raisons :***

- Grande difficulté à agencer mécaniquement le support car le dessus de la sonde est très encombré,
- Bilan de masse trop pénalisant, la charge utile comparée à la fragilité de la structure globale ne doit pas dépasser une charge estimée à 100g, tous circuits imprimés et capteurs confondus,
- Chargement "dans les hauts" particulièrement néfaste à la stabilité et surtout aux phénomènes inertiels lors des déplacements avec risques de basculements sur les terrains en forte pente,
- Augmentation trop importante des coûts, la motorisation spatiale dans ce domaine arrivant rapidement à des sommes "faramineuses".

☠ **DANGER** : Tout dispositif lumineux intégrant un composant de type LASER ou assimilé est potentiellement dangereux pour l'œil humain ou animal. Les diodes "MASER" appartiennent à cette catégorie de composants. **NE JAMAIS LES REGARDER DIRECTEMENT OU EN DIRIGER LE FAISCEAU LUMINEUX VERS UNE PERSONNE OU UN ANIMAL.** Sans que ce ne soit précisé par les fournisseurs, beaucoup de diodes électroluminescentes à haut rendement lumineux font partie de ces dispositifs. Aussi, lorsqu'une telle diode diffuse un rayonnement "scintillant", adopter impérativement les précautions d'usage relatives à ce type de composant optronique.

Suite à de nombreuses études et essais, les ingénieurs de la NDRMSE ont dégagé une solution à la fois viable et parfaitement élégante. On va se contenter par un bridage élémentaire, de fixer le module LASER sur le **Fémur** de la **Jambe A**. Du coup il n'y a pas besoin d'une motorisation supplémentaire, la compacité de la sonde n'étant strictement pas altérée. Une seule sortie **D6** est mobilisée sur Arduino. Cette dernière est sélectionnée car elle peut fonctionner en PWM. La luminosité du LASER sera de ce fait modulable par programme respectant ainsi les performances exigées pour l'ambitieux projet JEKERT. Le tout petit module présenté sur la **Fiche n°19** est vraiment facile à trouver dans le commerce en ligne et à un prix de vente très faible. Toutefois, rien ne vous interdit de le remplacer par une simple LED rouge associée à une résistance de limitation de courant. Ce sera bien moins "réaliste" qu'un pinceau de lumière cohérente, ce qui n'empêchera pas d'effectuer des essais en attendant de recevoir l'éventuel petit circuit commercial. Notez au passage que pour limiter le nombre de feuilles imprimées, le verso de la **Fiche n°19** fournit des informations sur l'encodeur rotatif KY-040 et anticipe largement de ce fait sur le **TOME 5** ... pour le moment on oublie. Caractéristiques du petit module en main, on constate que si la sortie **D6** fournit une tension continue de +5Vcc le courant absorbé par le module est de l'ordre de 22mA, largement compatible avec la sortance des broches binaires de l'ATmega328. Inutile de prévoir un quelconque amplificateur.

### ➤ Principe utilisé pour pointer le LASER.



Lorsque tous les moteurs sont amenés au *Neutre opérationnel* avec l'instruction "**p15\***" on peut vérifier que les Tibias sont en orientation horizontale, donc dans le prolongement des Fémurs. La Fig.122 présente leur configuration pour cette posture particulière. Hors la rotation sur les servomoteurs des Hanches permet de faire tourner horizontalement les Fémurs. Il devient alors possible en mode pilotage manuel de balayer horizontalement de  $-45^\circ$  à  $+45^\circ$  par rapport au centrage des servomoteurs. Le faisceau LASER peut alors passer "finement" de *Devant* jusqu'à *plain* travers sur *tribord*. Verticalement il suffit de commander l'orientation du Genou qui depuis le neutre à l'horizontale peut cabrer ou piquer jusqu'à  $\pm 45^\circ$ . Le programme "**p9m\***" (Mode décrit plus avant dans ce chapitre.) moyennant de sélectionner la bonne sortie du multiplexeur est apte sans autre forme de procès à assurer la fonction de pointage dans les deux directions cartésiennes. Pour ne pas que la Griffe ne risque de talonner quand on pointerait vers le

**Un peu d'histoire** : Située en haut d'un mât du fabuleux rover Curiosity, une caméra LASER développée par nos ingénieurs et scientifiques Toulousains fait preuve d'une longévité étonnante. Environ 500000 tirs LASER ont été effectués, "ChemCam" ayant collecté près de 10 000 images et enregistré plus de 15000 points de mesure. Bravo aux personnels qui l'ont conçue et qui la télécommandent à plusieurs centaines de millions de kilomètres de distance. C'est pour simuler de façon ludique le canon à faire fondre la roche que JEKERT embarque un ultra puissant LASER dont la **Fiche n°19** fournit les caractéristiques. À 5mW de puissance lumineuse ... il ne présente pas vraiment de grand danger pour les meubles et les murs du laboratoire !

**NOTE :** télécommander la sonde est forcément ludique, avec une fragrance de magie. Toutefois, pour préserver au maximum l'intégrité de la petite machine il importe de respecter les protocoles établis "avec rigueur". D'une part les procédures opérationnelles tiennent compte des préalables, et surtout elles parent certains incidents. Généralement elles sont prévues pour préserver le matériel, en particulier écarter les "brusqueries mécaniques". Toutefois dans le cas d'utilisation du petit LASER, l'entourage également se doit d'être en sécurité. Aussi, quand vous désirez expérimenter le *pilotage manuel du LASER*, il me semble *impératif d'avoir la Fiche n° 16 en main et de respecter les protocoles opérationnels*. Tout particulièrement pour la fin de l'opération car elle évite de "brutaliser" la machine en préparant la posture avant de déclencher les mouvements rapides. Par ailleurs, *on éteint le LASER pour éviter que durant ces balayages le faisceau puisse risquer de passer sur le visage des personnes présentes* dans la pièce.

bas il est recommandé de l'effacer en butée logicielle vers le haut. Ce n'est pas vraiment obligatoire, tout au plus une action esthétique qui confirme durant la campagne de tests que la machine est bien en *Configuration LASER* et que la présence des personnels en salle S11 est exclue.

➤ **Lever une jambe sur quatre fait perdre l'équilibre !**

Titre qui fleure étonnamment le "déjà vu". Et oui, c'est la faille. Quand on va pointer le LASER la sonde expérimentale ne sera plus posée que sur trois *Jambes* et nous savons qu'elle va "piquer du nez" puisque le centre de gravité sera déporté vers l'Avant et vers Tribord. La solution qui éliminera radicalement ce problème est au final élémentaire. Il suffit de modifier la surface de sustentation comme montré sur la Fig.123 sur laquelle JEKERT est vue par dessus. En jaune sont représentées les trois jambes qui supportent l'ensemble avec **b**, **c** et **d** les points de contact des chaussettes sur le sol. Compte tenu de la masse actuelle de la sonde à pleine charge, celle de la

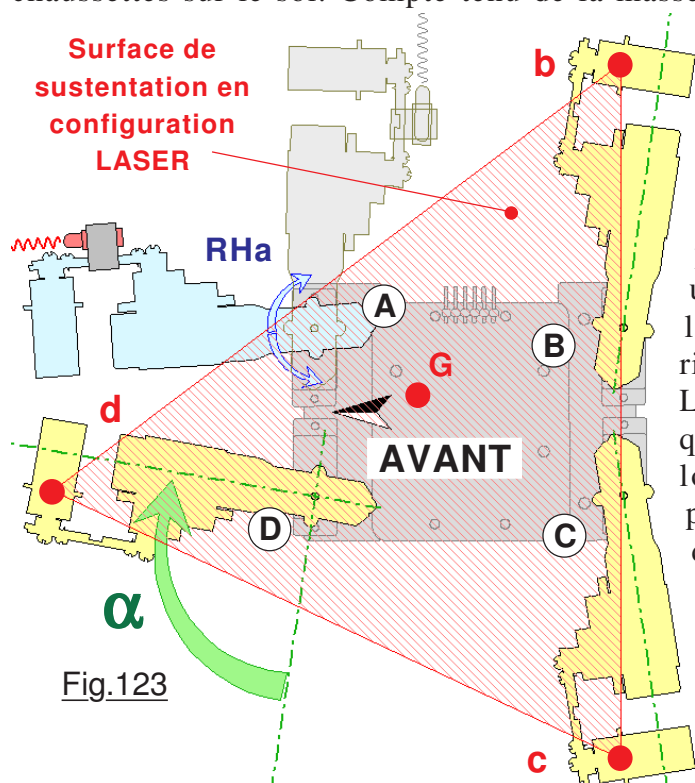


Fig.123

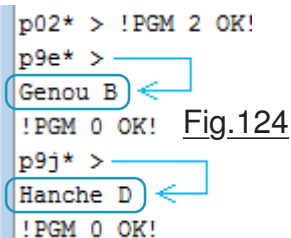
*Jambe A* devient faible par rapport au total. Le centre de gravité **G** se trouve de ce fait pas très loin du centre du triangle de sustentation. La posture adoptée est bien stable et autorise des pointages précis. Pour agrandir au maximum la surface de sustentation et la déporter le plus possible vers l'avant, l'angle  $\alpha$  est choisi pour avoir une orientation vert tribord optimale. C'est à dire le plus possible vers la droite, sans pour autant risquer une collision avec la *Jambe A* quand le LASER pointe au maximum vers la gauche. Notez qu'en réalité le pointage vers l'avant dépasse l'axe longitudinal du châssis. C'est volontaire pour pouvoir avec certitude pointer un rocher qui serait devant la sonde. Comme la motorisation le permet, l'ouverture vers tribord est également optimisée, c'est à dire la plus grande possible sans pour autant risquer une interférence entre le petit module LASER et son dispositif de bridage avec la *Jambe* arrière **B**. Pour passer de la posture initiale *Stable Transversal* à celle de *Stable LASER* par la

commande "*p12\**" on utilise un mouvement rapide qui divise le balayage  $\alpha$  en *6 étapes rapides* car il ne faut pas perdre l'équilibre. Seule la *Jambe D* est concernée et pour chaque rotation élémentaire la *Griffe* Gest soulevée du sol puis reposée pour minimiser les frottements et l'usure des chaussettes. La posture *Stable LASER* étant effective on place alors la *Hanche* de la *Jambe A* au neutre et on soulève la *Griffe* par un mouvement coordonné. L'exploitation du LASER peut alors débuter. Riches d'enseignement, les trois images *Image 11.JPG*, *Image 12.JPG* et *Image 13.JPG* rangées dans le dossier <Galerie d'Images> dévoilent une kyrielle de détails matériels et anticipent largement sur le **TOME 4**. En particulier l'*Image 12.JPG* est à rapprocher de la Fig.117 car, bien que prévue pour montrer la posture *Stable LASER* elle visualise un éventail panoramique des expériences scientifiques embarquées.



### > Pilotage manuel des moteurs.

**A**nnoncé dans le **TOME 2** dans le chapitre relatif à l'apprentissage, ce mode n'était pas effectif dans la liste des programmes disponibles sur le démonstrateur **P13**. Pour expérimenter cette commande il fallait faire appel au programme complet **P30** disponible dans les fichiers fournis. Dans le démonstrateur actuel **P14\_Sonde\_avec\_capteurs\_scientifiques.ino**, "**p9m\***" est directement utilisable et intégré dans le jeu d'instructions dédié à l'exploitation de la sonde sur le sol martien. Il me semble utile de présenter plus en détail cette fonctionnalité. Elle s'avère particulièrement utile pour positionner chaque moteur individuellement. Les ingénieurs pourront ainsi faire adopter à JEKERT toutes les postures imaginables en vue d'ouvrir l'éventail des recherches, et surtout de parfois dégager l'exploratrice d'une situation critique. Lors de la mise au point des programmes, orienter librement un moteur et ensuite se faire lister sa consigne par la commande "**i\***", peut s'avérer très salubre. Nous en verrons un exemple lorsque l'on va traiter du spectromètre chromatique. Pour rappel, quand on impose la commande "**p9m\***" au logiciel embarqué, la lettre **m** qui accompagne le programme **p9** précise le moteur à piloter. La **Fiche n°12** pour l'instruction **p9m\*** indique la correspondance entre les lettres prises dans l'ordre alphabétique et le servomoteur impliqué. Lorsque la consigne est reçue, l'ACR retourné par l'esclave est complété comme montré sur la Fig.124 par une indication en clair de l'articulation qui bougera ainsi que de la **Jambe** sur laquelle on agit. Tout mouvement si potentiomètre actif engendrera une réaction du servomoteur piloté, la motorisation n'étant pas mise sur OFF par la commande "**f\***". On peut à tout moment frapper la commande "**i\***" pour se faire lister les consignes correspondant à la posture actuelle, à condition toutefois que l'option "**c\***" soit en état validée. Piloter les moteurs sur plusieurs mobilités engendre forcément des risques de brusqueries mécaniques et de perte d'équilibre de la sonde. Dans ce type de manipulation il importe de suivre avec méthode les directives du protocole listé sur la **Fiche n°16**, avec en particulier la précaution de figer la motorisation sur OFF chaque fois que l'on change d'articulation pilotée. Et n'oubliez pas après avoir effectué les essais et mesurages envisagés, de mettre fin au mode manuel avec "**k\***". Si vous passez outre, chaque fois que vous allez invoquer des postures comme **VEILLE** ou **Stable transversal**, la sonde va prendre la configuration initiée et immédiatement le moteur resté en contrôle manuel va diverger en fonction de la position du potentiomètre. Vous penserez alors à un incident matériel ou une vermine dans le programme ... mais la LED rouge commandée par **S15** est toujours allumée !



p02\* > !PGM 2 OK!  
p9e\* > !PGM 0 OK!  
Genou B < Fig.124  
p9j\* > !PGM 0 OK!  
Hanche D <

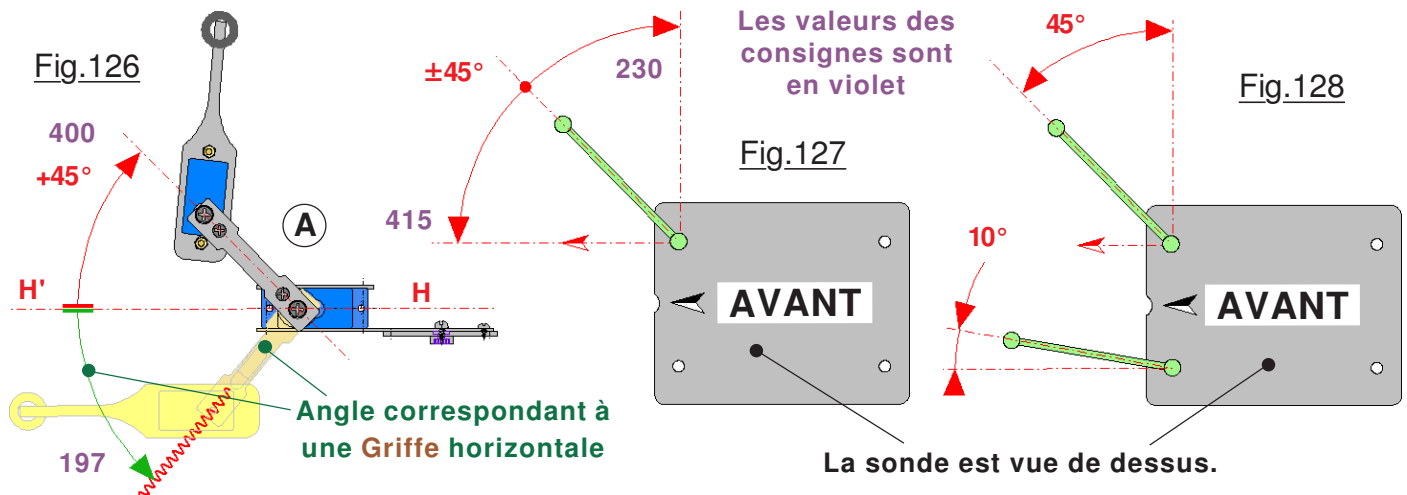
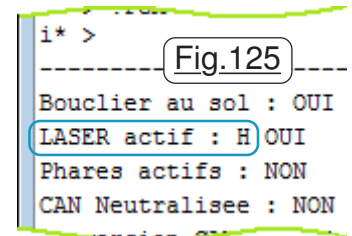
### > JEKERT a bon caractère !

**F**aciliter le travail des ingénieurs qui prendront en charge l'exploitation de la sonde posée sur la lointaine planète devrait rester une priorité. Dans la mesure où l'on peut ménager un peu de place disponible dans la mémoire programme du calculateur de bord, il importe autant que faire ce peut, d'ajouter les commandes les plus simples possibles pour télécommander JEKERT. Hors le pilotage du LASER fait partie des facettes d'utilisation opérationnelles prépondérantes. Pour en retenir plus aisément les commandes, deux instructions à un caractère ont été ajoutées dans la liste. Vous pourrez observer sur le programme complet **P30** que les vingt six lettres de notre alphabet ont été affectées, et qu'il a fallu ajouter quelques autres caractères spéciaux. Dans la mesure où c'est possible la lettre correspond directement à l'action engagée par la commande. Ce n'est pas toujours le cas. Par exemple **x**, **y** et **z** ont été affectées par défaut car toute autre lettre qui aurait été plus évocatrice était déjà attribuée. Les fonctions marginales ont donc été "sacrifiées" ...

**F**orce est de constater à l'usage que durant un balayage LASER, pour choisir le moteur à animer, des consignes comme "**h\***" et "**v\***" plus parlantes que "**p9b\***" ou "**p9a\***". Le choix des lettres retenues correspondent naturellement à un balayage **horizontal** ou **vertical**, faciles à mémoriser, alors que retenir qu'il faut solliciter le programme n°9 suivi de la lettre **a** ou **b** n'a rien de naturel. C'est le logiciel du calculateur Arduino NANO qui recevant l'ordre "**h\***" ou "**v\***" sélectionnera d'autorité la bonne sortie sur le multiplexeur. Au même titre que l'information d'état du mode pilotage manuel était visualisée par une LED rouge, l'exploitation des balayages LASER allume une LED dédiée verte pilotée par **D9**. (*Verte pour raison d'homogénéité avec la LED verte du mode Torsion.*) Avant d'aborder la technologie des autres expériences embarquées, testez abusivement pour le plaisir le pilotage manuel des moteurs et celui du LASER ... **Fiches** de protocoles en main évidemment !

Notez au passage que lorsque le LASER est actif ET que l'un des deux modes de balayage est effectif, les informations retournées par la commande "i\*" le précisent comme montré sur la Fig.125 où le OUI est complété par la lettre "H" dans l'encadré bleu clair. (Ou par "V" si c'est l'orientation verticale qui est en cours d'utilisation.)

Notez que toute commande de type programme du genre "pNN\*" désactive le mode balayage LASER, bien qu'en toute logique la première action en fin de manipulation devrait faire appel à "p13\*". Pour ceux qui désirent scruter avec attention le logiciel, les dessins Fig.126, Fig.127 et Fig.128 donnent les valeurs des consignes aboutissant à la posture **Stable LASER** et aux amplitudes angulaires de balayage.



## 28) 25/11/2017 : Le télémètre à ultrasons (MJD 58082)

Prendre des décisions pour la suite à donner à l'exploitation d'une sonde aussi éloignée dans le système solaire n'est jamais anodin. Les échanges radio entre la Terre et Mars sont laborieux et prennent un temps considérable. Par ailleurs, évaluer l'éloignement d'un obstacle à partir des images à haute résolution issues d'une caméra embarquées n'est pas évident et toujours imprégné d'une certaine imprécision aléatoire. Hors il faut décider de la trajectoire à imposer. Choisir entre poursuivre sans changer de direction ou orienter différemment le mobile. Si l'on constate que l'on s'est trompé, faire reculer la sonde puis élaborer une autre stratégie va englober des heures. Aussi, pour "rentabiliser" la machine scientifique et le cout des personnels impliqués dans sa gestion, il devient pratiquement indispensable de munir l'explorateur d'un radar. Celui installé sur JEKERT est immobile sur le châssis et orienté en capture dans l'axe vers l'avant du véhicule. Il fonctionne aux ultrasons, technologie possible car la planète visitée possède une atmosphère.

Fondamentalement, le principe de fonctionnement d'un radar ou d'un sonar est élémentaire. On projette dans le milieu observé une "pulse" sous forme d'une onde électromagnétique ou sonore. Cette dernière ricoche sur l'obstacle dont on désire déterminer la distance. Le temps **T** mis par l'onde pour parcourir l'aller et le retour est mesuré. La distance **D** à laquelle se trouve l'écho est égale à  $D = V \times T / 2$ , formule dans laquelle **V** est la vitesse de propagation de l'onde dans le milieu traversé. Il faut diviser par deux car avec l'aller et le retour de l'écho l'onde à franchi deux fois l'éloignement. Considérons la [Fiche n°20](#) qui traite du petit module HC-SR04. Ce dispositif électronique est équipé de deux transducteurs piézo-électriques identiques. Chaque composant peut fonctionner aussi bien en transmission d'une impulsion ultrasonore s'il est soumis à une sollicitation électrique calibrée à sa fréquence de résonance, qu'en capteur acoustique. Si l'air ambiant vibre à sa fréquence de résonance, la cellule piézo-électrique sera soumise à des compressions mécaniques car elle est associée à une membrane souple, exactement comme pour un petit haut-parleur. Les faibles tensions électriques générées à la réception sont alors amplifiées et traitées pour mesurer le temps mis par l'écho pour revenir suite à une impulsion initiale. Il serait possible de n'utiliser qu'un seul composant ultrasonore, ce qui compliquerait sa mise en œuvre, car un système de commutation rapide devrait effectuer l'aiguillage entre l'oscillateur du transmetteur et l'amplificateur sensible du récepteur. Ce type de composant restant d'un prix d'achat vraiment faible, il devient nettement plus simple et plus économique d'oublier l'électronique d'aiguillage et d'installer deux transducteurs.



### ➤ Ping Pong acoustique.

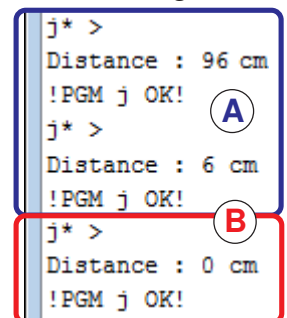
L' image donnée en Fig.3 sur [Fiche n°20](#) présente le module HC-SR04 vu de face avec à gauche l'antenne d'émission et à droite celle de réception. Les informations de branchement correspondent à ceux de JEKERT dans sa version définitive. Comme un système de commutation d'antenne a été écarté, il faut mobiliser deux broches de l'ATmega328. Des Entrées/Sorties binaires conviennent parfaitement. C'est **D12** et **D13** qui seront affectées à ce capteur. Le HC-SR04 à l'image des modules électroniques actuels intègre une logistique électronique et informatique qui en rend la mise en application élémentaire. La [Fiche n°20](#) expose le principe de mesurage qui se fait en deux temps. On déclenche la transmission d'une impulsion acoustique en plaçant un état logique "1" d'au moins 10 µs sur la broche **Trig** pilotée par **D12** qui correspond à une entrée sur le module pour déclencher une mesure. En retour, sur sa sortie **Output** (*Ou écho électronique.*) branchée sur **D13** on mesure la durée de l'impulsion fournie dont la durée est proportionnelle à la distance, si le module détecte un objet. Afin de pouvoir calculer la distance exprimée en cm, on utilise la formule de conversion  $\text{Distance en cm} = (\text{Durée de l'impulsion Echo en } \mu\text{s}) / 58$  précisée dans les documents de caractéristiques qui accompagnent le module. Notez au passage que le "Ping" sonore émis dans l'atmosphère présente une fréquence de 40kHz. Comme une oreille humaine ne détecte que des sons globalement inférieurs à 18kHz, nous n'entendons pas le signal envoyé par le transducteur. C'est la raison pour laquelle un tel "bruit" est qualifié d'Ultrason.

### ➤ Jalonner le terrain.

Concrètement, l'écriture du logiciel qui traite du télémètre à ultrasons se résume à peu de chose. L'intégralité du traitement pour le HC-SR04 est présenté sur la [Fiche n°20](#). Le code source est directement extrait du logiciel qui anime la sonde et reste d'une grande banalité. Éventuellement, deux instructions sont à souligner. Commodité apportée par le compilateur C++ d'Arduino, `delayMicroseconds(10)` réalise une temporisation de 10µs. Ainsi on peut calibrer très facilement la durée de l'impulsion de déclenchement du processus de mesurage. D'une souplesse remarquable, la fonction `pulseIn(EntéeBinaire, EtatLogique)` retourne la durée en µs d'une impulsion mesurée sur une broche initialisée en `EntéeBinaire` durant l'`EtatLogique` défini par les constantes `LOW` ou `HIGH`. (*En fait cette fonction est encore plus riche et peut tester un dépassement de délai.*) Dans le démonstrateur l'instruction `Lecture_Echo = pulseIn(Echo, HIGH)` affecte à la variable nommée `Lecture_Echo` la durée à laquelle la broche `Echo` monte et reste à l'état logique "1". Dans le traitement on va se contenter d'une précision d'un centimètre, alors que la résolution du capteur est de 3mm. C'est suffisant comme finesse dans le cadre de cette application ludique.

Fig.129

La commande qui incite la sonde à effectuer la mesure d'une distance d'obstacle "j\*" déclenche une mesure ultrasonore vers l'avant de la sonde et retourne la distance d'un éventuel obstacle exprimée en centimètres. La portée maximale est limitée par le logiciel à 2,5m car pour minimiser la taille des variables on a utilisé pour `Distance` un `byte` et pour `Lecture_Echo` un `int`. Si vous ne disposez pas du module et que le connecteur HE14 est inoccupé, la distance indiquée sera de 0cm. Sur la Fig.129 en **A** le module fonctionne correctement. En **B** il a été enlevé du connecteur. Donc ceux qui n'ont pas encore reçu leur petit capteur peuvent effectuer tous les essais logiciels qu'ils désirent sans pénalité. Il vous arrivera certainement de diriger le châssis vers un mur lointain et d'avoir en retour des distances du genre 96cm. Ce n'est pas que le module ne fonctionne pas bien. Si vous analysez les caractéristiques du HC-SR04, vous constaterez que l'angle d'ouverture de l'antenne ultrasonore (*Diagramme polaire.*) dépasse les ±25°. Du coup une pulse peut accrocher des échos latéraux sur des obstacles qui semblent visuellement très éloignés. En revanche, placez votre main bien dans l'axe et vous observerez que le dispositif fonctionne parfaitement. Ainsi il est utilisé pour l'anti-collision dans les séquences de protection logicielle avec une fiabilité remarquable. JEKERT ne risque pas de se taper le nez contre un lampadaire ! Jalonnez, jalonnez, il en restera toujours quelque chose. Autrement dit, testez ce radar pour en cerner les comportements, les réactions. Expérimentez des obstacles divers comme une plaque de carton, miroir en verre, matériau absorbant comme du polystyrène expansé, du tissu etc. Bien que ces matériaux ne seront pas rencontrés sur le chemin de notre exploratrice, il serait bien dommage de ne pas tester les réactions sur Terre du module HC-SR04 ...



## 29) 26/11/2017 : Le radar ultrasonore à balayage (MJD 58083)

**B**ien que le cône d'ouverture du transducteur HC-SR04 soit bien trop évasé pour permettre une analyse fine de la topographie de l'environnement situé à l'avant de la sonde, créer un programme qui se charge d'effectuer un balayage avec enregistrement des échos est tout à fait séduisant au point de vue informatique. Aussi nous allons munir la sonde d'un télémètre à ouverture avec balayage angulaire, enregistrement et visualisation graphique du spectre.

Enregistrer un panoramique télémétrique reste assez élémentaire en théorie. La pratique n'est pas très complexe non plus. Il suffit d'exploiter la faculté de "Torsion" de JEKERT et ainsi lui faire adopter une suite d'orientations du châssis apte à "embrasser du regard" l'environnement situé devant la sonde. Par programme on impose une torsion en butée logicielle à droite. Puis le logiciel va imposer l'adoption de 64 orientations intermédiaires pour balayer entièrement la plage du panoramique envisagée. À chaque position le démonstrateur déclenche une mesure de distance et enregistre le résultat dans la mémoire statique non volatile EEPROM du microcontrôleur. On dispose ainsi d'une sorte de "photographie télémétrique" apte à construire un spectre topographique.

**P**ourquoi 64 échantillons ? La valeur adoptée résulte d'un compromis. Chaque échantillon ne consomme qu'un octet en EEPROM, il serait tentant d'augmenter la résolution de l'instrument. Toutefois, ce serait illusoire pour deux raisons. Nous avons vu que le transducteur présente un cône de réception qui manque singulièrement de finesse. Par ailleurs, on peut toujours envisager la restitution du spectre sur un écran graphique. Dans cette hypothèse, la valeur de 64 est bien adaptée aux composants du commerce de définitions "standards". Enfin, avec 64 positions intermédiaires, on optimise la "sensibilité" d'orientation angulaire des servomoteurs. C'est la commande sur un caractère "e\*" qui déclenche un balayage télémétrique avec enregistrement topographique.

**V**isualiser le spectre enregistré se commande avec "w\*", le w faisant référence à "Waveform". (Pour programmer cette fonction, les lettres de l'alphabet étaient pratiquement toutes déjà affectées à d'autres commandes sur un caractère.) Le résultat est présenté sur la Fig.130 sur laquelle en 1 le mesurage est déclenché. Puis en 2 on sollicite la sonde pour qu'elle nous retourne sous forme graphique la "forme" de l'environnement qui a été enregistré. Ces données resteront en EEPROM aussi longtemps qu'un nouvel enregistrement ne sera pas initié avec "e\*". Il sera possible librement et à tout moment de se faire visualiser la panoramique avec "w\*". Ce dernier est montré sous forme graphique sur l'écran du P.C. avec un encadrement vertical 3 et 8 et terminé par l'ACR 9. Les 64 lignes de représentation dépassent largement en hauteur l'écran vidéo, raison pour laquelle sur la Fig.130 le repère 7 symbolise une coupure. Sur le moniteur vidéo il faudra utiliser "l'ascenseur" latéral et la souris pour exploiter verticalement cette image artificielle. Toute image radar peut être polluée par des échos scintillants parasites tels que ceux visibles par exemple en 5 et 6. C'est la raison pour laquelle, s'il s'agissait d'un système technique élaboré, il faudrait inclure dans le programme d'exploitation des techniques de filtrage pour éliminer les "échos fantômes". Sur JEKERT un tel développement logiciel ne se justifie pas, trop consommateur d'octets de programme. On se contentera donc d'une informatique simplifiée. Si ces deux fonctions sont très séduisantes pour enrichir notablement les performances opérationnelles de la petite machine robotisée, il ne faut toutefois pas en attendre des miracles. N'espérez pas retrouver sur le graphe le beau visage de votre compagne ou compagnon. L'angle d'ouverture des transducteurs est bien trop "flou" pour en faire un portraitiste de renommée mondiale !

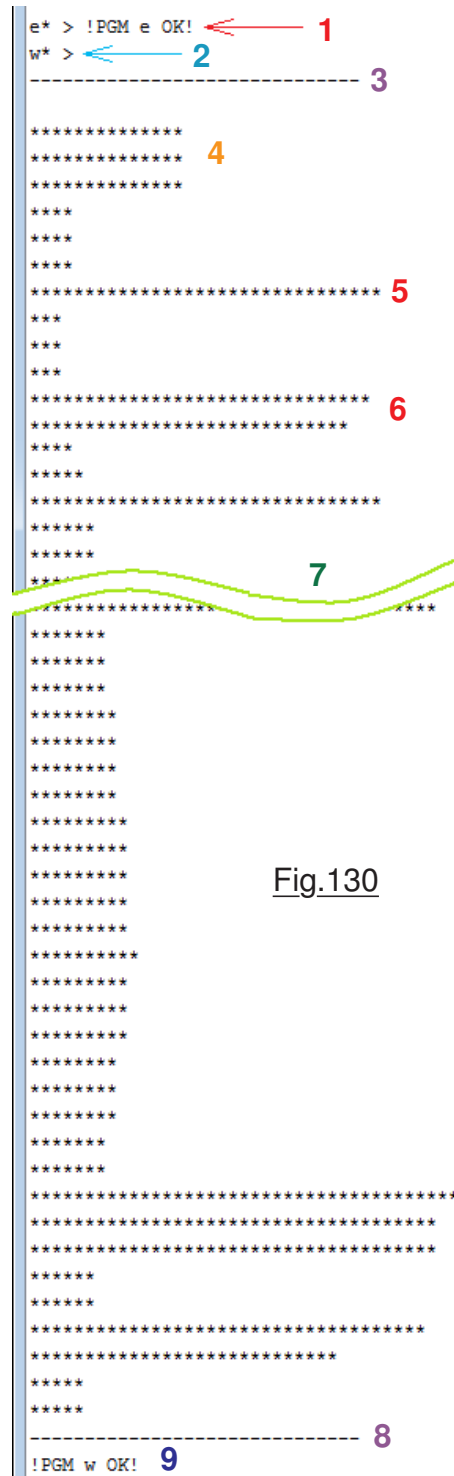


Fig.130

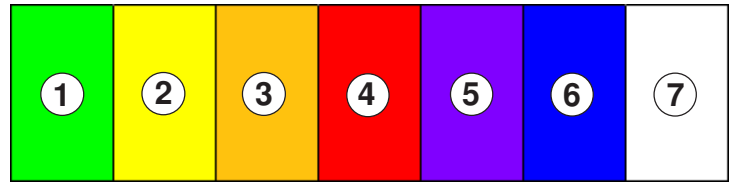


### 30) 27/11/2017 : Le spectromètre chromatique (MJD 58084)

**G** lups, mais c'est quoi ce truc avec des mots si compliqués ? Pour frimer tu sors ça à ta réunion ministérielle et tout le monde te vote les crédits que tu voulais pour le projet JEKERT !

Bon, soyons sérieux. Si ce titre accrocheur attire l'attention, il cache une réalité pratiquement élémentaire. Nous allons expérimenter les filtres colorimétriques. De quoi est-il question ? Fig.131

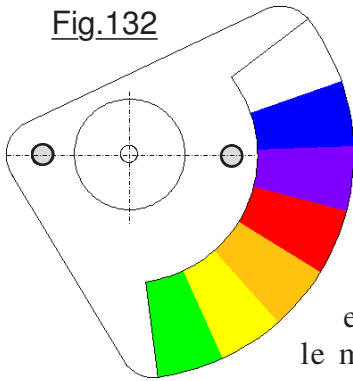
Depuis "toujours", les astrophysiciens à l'aide de leurs télescopes exploitent les couleurs pour déterminer des informations cruciales arrivant des étoiles ou les planètes, qu'ils observent avec des dispositifs matériels de plus en plus sophistiqués. L'idée de base consiste à



décomposer la lumière reçue pour en "étaier" les couleurs, un peu comme le fait naturellement cette merveille que l'on nomme arc en ciel. Lisez l'encadré qui traite de façon scandaleusement sommaire un peu de colorimétrie, ainsi nous aurons une idée de ce qui incite les chercheurs à munir JEKERT d'un spectromètre chromatique, c'est à dire un appareil sensé analyser la lumière reçue par la cellule photorésistante et filtrer couleur par couleur la quantité d'énergie lumineuse reçue.

**L** instrument réalisé est rudimentaire et n'impose strictement aucun ajout électronique au robot. Son principe de fonctionnement est basé sur l'usage de filtres colorés qui sont placés devant le capteur photosensible. Chaque filtre ne laisse passer qu'une "tranche" du spectre lumineux. Pour simplifier, nous allons nous contenter de six couleurs comme le montre la Fig.131 qui présente l'ordre dans lequel seront laissées passées les différentes longueurs d'ondes.

Fig.132



En **7** le masque sera totalement transparent. Ainsi l'appareil enregistrera six valeurs chromatiques et la quantité totale d'énergie lumineuse reçue par la sonde. Il sera alors possible d'établir des pourcentages énergétiques si on le souhaite. Notez au passage que pour concevoir notre spectromètre, on utilise les quatre couleurs fondamentales en **1, 2, 4 et 6**. Ainsi nous n'écartons au point de vue conceptuel ni la soustraction, ni l'addition. Les deux nuances intermédiaires **3 et 5** sont des "mélanges" établies "à l'estime" avec le logiciel **PAINT.exe**. Pour interposer chaque filtre coloré entre la cellule photorésistante et le ciel, l'idée de base consiste à transformer le modèle de la Fig.131 en Fig.132 un équivalent circulaire. Ainsi, le filtre

**Quelques notes sur les couleurs :** Quand à l'école primaire l'institutrice vous initiait au dessin, tout particulièrement à "peinturlurer", elle vous présentait les couleurs fondamentales. Tout artiste sait qu'avec les trois bases **ROUGE, JAUNE** et **BLEUE** on peut par des mélanges proportionnés obtenir n'importe quelle autre nuance de couleur. Par exemple bleu plus rouge bien brassé donne du violet. Newton a montré que si l'on ajoute toutes les nuances, notre vision voit ce que l'on nomme du blanc. Curieusement, si vous observez un écran lumineux, vous constaterez que les luminophores sont **ROUGE, VERT** et **BLEU**. On en déduit qu'en électronique lumineuse les couleurs fondamentales ne sont pas identiques. Pourquoi ?

Avec des crayons de couleur, *en peinture*, en imprimerie, *on procède par soustraction*. C'est à dire que des grains de couleur rouge cachent des grains bleus et réciproquement. En fonction des pourcentages de ce qui est caché, vu de loin pour ne plus distinguer les grains, l'œil humain "voit du violet". *En optronique, on procède par addition*. Les luminophores contrairement aux grains de peinture ne se masquent pas les uns des autres. Notre organe sensoriel reçoit donc deux sources d'excitations qui ajoutées l'une à l'autre vont engendrer des sensations de couleurs différentes en fonction de leur proportions énergétique. Pour l'astrophysicien on va retrouver des phénomènes un peu analogues. Une étoile émet du rayonnement. Ses ondes lumineuses s'ajoutent et le scientifique peut en déduire des informations comme la température, la composition etc, en fonction des "franges colorées". Une planète se contente de renvoyer la lumière solaire. Elle va de ce fait modifier le spectre par absorption, (*Donc soustraction.*) que ce soit par son sol rocheux ou l'atmosphère qui le masque. Dans les deux cas, les scientifiques analysent l'atténuation de la lumière réfléchiée en fonction de sa "longueur d'onde" et en déduisent une foule de paramètres comme la composition atmosphérique, la texture du sol ...

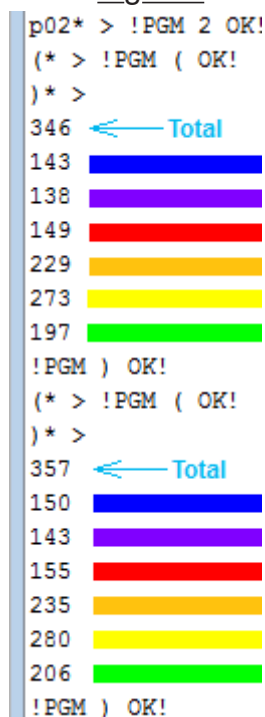
sera tout simplement immobilisé à la bonne hauteur sur le **Fémur** de la **Jambe A**. Au final, avec un trois fois rien de "mécanique" on a nobli la cellule photorésistante qui ne pouvait que mesurer une intensité lumineuse globale en un spectromètre chromatique qui ouvrira une foule d'expériences à la fois simples et très formatrices. Dans ce chapitre, expliciter la réalisation matérielle du spectroscope est hors sujet. Sa réalisation sera abordée finement dans le **TOME 4**. Contentez-vous d'aller voir [Image 14.JPG](#) dans le dossier <Galerie d'Images>. La photographie montre assez de détails pour comprendre l'agencement de l'instrument de mesure ainsi improvisé. Consultez la [Fiche n°21](#). Pour créer le filtre de couleurs il faut l'imprimer sur des feuilles de milard transparentes. ATTENTION, chaque imprimante doit utiliser des références qui lui sont propres. Quand on regarde le résultat obtenu, les couleurs jaune et orange semblent trop pâles. Aussi, sur le prototype j'ai découpé avec soin deux secteurs colorés. Ils sont immobilisés "parfaitement" l'un au dessus de l'autre. (*C'est peu visible sur la photographie mais bien réel.*) En toute rigueur deux secteurs colorés sont suffisant pour équiper la sonde. La [Fiche n°21](#) en propose douze ... *gaspillage !*

Pas forcément. Surtout attendez le **TOME 4** pour avoir les conseils relatifs à l'opération de découpage de cet élément. Ce n'est pas immédiat et les risques de loupés sont réels. Avec tous les éléments imprimés vous arriverez forcément à en façonner deux de parfaits. L'argument le plus pertinent à mon sens concerne les copains. Perdre une feuille A4 complète pour deux secteurs imprimés serait incontestablement un gaspillage patent. Les divers secteurs non utilisés seront particulièrement appréciés par les copains. Je suggérerais fortement de vous grouper. L'un d'entre vous se chargerait de l'achat du support, et de l'impression sur son ordinateur. Les autres participeraient financièrement. Ainsi vous arriveriez à un coût raisonnable pour le petit bout de transparent colorié ...

### ➤ Automatiser et enregistrer le spectre.

**J** EKERT dispose matériellement de tout ce dont nous avons besoin pour établir un spectre chromatique. Le filtre étant immobilisé sur le **Fémur** de la **Jambe A**, avec la commande "p9a\*" il suffit de faire tourner la **Hanche** pour amener chaque couleur bien au dessus de la cellule photoélectrique. Puis on déclenche un mesurage météorologique avec "m\*" et l'on note sur une feuille de papier la valeur pour le total et pour chaque tranche colorée. Totalement gratuite, puisqu'il n'y a

Fig.133



strictement rien à ajouter au logiciel, cette solution n'est pas du tout satisfaisante sur le plan opérationnel. Elle s'avère particulièrement laborieuse, car il faut en permanence se placer à la verticale au dessus de la cellule pour bien centrer le secteur coloré. Ensuite reposer le potentiomètre et revenir au clavier pour se faire afficher la valeur numérisée de la lumière. Chaque manipulation exige entre trente et cinquante secondes. Vous allez vous apercevoir à l'usage, que la cellule photorésistante est d'une sensibilité infiniment plus grande de celle de l'œil humain. Une toute petite variation lumineuse se traduit immédiatement par un changement de valeur numérisée. Un trois fois rien de nuage dans le ciel évoluant lentement devant le Soleil engendre des variations qui fausseraient complètement les mesures. Aussi, il importe de procéder rapidement à l'échantillonnage. L'analyse spectrale se doit d'être automatisée, ce qui implique de mémoriser les valeurs. Sur la Fig.133 deux enregistrements coup sur coup ont été effectués. Il est évident sur cette expérience qu'entre les deux échantillonnages pourtant séparés d'à peine vingt secondes, la luminosité ambiante a changé. La variation est insensible visuellement. La cellule photorésistante enregistre pourtant un net changement de numérisation.

Quand on observe la lumière solaire à travers les filtres colorés, on constate bien que le bleu et le violet engendrent une forte atténuation confirmée par les mesures. Par contre, à l'observation visuelle le vert semble très lumineux. Pourtant, numériquement l'atténuation est significative. C'est un autre facteur

physiologique trompeur. Notre vue est plus sensible dans le vert que dans le rouge. C'est du reste la raison pour laquelle il est si difficile d'assortir des couleurs vestimentaires vertes. La moindre différence se voit. Comme dans la nature la couleur verte est dominante, rendre notre regard plus sensible dans cette teinte est un moyen pour "la vie" d'optimiser notre vision de l'environnement. Et sans que vous ne le sachiez, les systèmes électroniques de visualisation polychrome comme les téléviseurs affectent des énergies différentes en fonction de la couleur du pixel.



### ➤ Les choix des nouvelles commandes.

Bien que l'idée d'ajouter un spectrographe colorimétrique à JEKERT hantait le laboratoire depuis les tous premiers octets soumis à l'ATmega328, c'est la dernière fonction opérationnelle qui a été émulée. Maintenant que je vous ai livré la solution toute simple qui consiste à placer la cellule photorésistante en un endroit stratégique pour pouvoir placer un filtre coloré sur le **Fémur** de la **Jambe A** semble couler de source. Y auriez-vous pensé ?

Par ailleurs, comment réaliser un filtre coloré ? Les transparents du commerce pour les imprimantes peuvent-ils réaliser un secteur coloré efficace ? Les couleurs fondamentales vont vraiment engendrer des différences significatives sur une cellule photorésistante banale ?

Toutes ces questions laissent supposer de nombreux essais, impliquant du bricolage mécanique et pas mal de temps englouti pour valider le concept. Il était en vérité plus "rentable" de consacrer du temps à la motricité de JEKERT, à développer son autonomie pour une version "définitive", à développer des expériences variées pour enrichir la petite machine. Dans ce contexte, c'est lorsque tout a parfaitement fonctionné que passer à la couleur devenait pertinent.

Ajouter deux programmes du genre "**pNN\***" à la liste actuelle "bousculerait" un peu trop la **Fiche n°12**. Comme de plus les commandes sur un seul caractère sont agréables à l'usage, il a été décidé d'allonger "la liste alphabétique". Comme caractères n'imposant ni SHIFT ni ALT ni CTRL il ne restait que '(' et ')', le choix d'ouvrir une parenthèse s'est imposé naturellement. Procédant par rotation du **Fémur** de la **Jambe A** la technique s'apparente fortement à l'exploitation du LASER. Aussi on en réutilise la posture stable, ce qui oblige à un programme préalable "**p02\***". La commande "**(\***" déclenche un enregistrement spectral en EEPROM. Associée à cette dernière, "**)\***" provoque la restitution sur l'écran du Moniteur série USB des valeurs enregistrées. Comme cette fonction a été développée de façon tardive, aucun texte spécifique ne lui est dédié. On se contente de lister dans l'ordre les valeurs mémorisées. C'est le programme complet résident sur la raquette de pilotage (*Pour le moment ce n'est qu'un lointain projet.*) qui sera chargé de présenter les données de façon claire. En premier est indiquée la valeur Totale, (*Voir la Fig.133*) puis suivent dans l'ordre les énergies filtrées en commençant par le bleu et en terminant par le vert.

### ➤ L'aspect programmation du spectromètre chromatique.

Aiguillage compris, les deux commandes ne consomment que 296 octets sur le démonstrateur **P14\_Sonde\_avec\_capteurs\_scientifiques.ino** et 226 sur le programme **P30** qui remplace celui du **TOME 2** et fourni dans les fichiers qui accompagnent le **TOME 3**. C'est à mon sens pas beaucoup pour une expérience que je trouve très séduisante. Un problème restait à résoudre : Où enregistrer le spectre, sachant que l'EEPROM est saturée, qu'il n'y reste plus un seul octet de disponible ? Un spectre colorimétrique est assez proche structurellement à celui d'un panoramique télémétrique. Par ailleurs il ne consomme pour être mémorisé que sept entiers soit quatorze octets.

Pour minimiser l'impact d'une "surcharge par écrasement", il a été décidé de loger un spectre de colorimétrie sur les derniers octets d'un panoramique télémétrique. Il en résulte deux petits inconvénients qui pour cette application sont estimés dérisoires :

- Un enregistrement chromatique faussera les 14 derniers échantillons d'un panoramique télémétrique.
  - Un enregistrement panoramique télémétrique "écrasera" un enregistrement chromatique présent.
- Ces deux inconvénients sont acceptables, car logiquement, un enregistrement qu'il soit télémétrique ou colorimétrique sera visualisé immédiatement après pour exploitation. Ce n'est que sur le long terme que l'un des deux se superposera à l'autre, ce qui n'affecte pas vraiment l'exploitation de la sonde sur le terrain.

```
void Restituer_le_spectre_colorimetrique() {  
    PTR_EEPROM = 603; Serial.println();  
    for (byte l=1; l < 8; l++) {  
        Serial.println(Lire_un_Entier_en_EEPROM(PTR_EEPROM));  
        PTR_EEPROM++; PTR_EEPROM++;  
    }  
}
```

Visualiser les valeurs fait appel à une procédure vraiment très banale. On positionne le pointeur en EEPROM en **603** qui est l'adresse relative du premier échantillon. Puis sept fois on effectue une lecture en EEPROM de la valeur, on affiche cette dernière sur le moniteur, on fait passer à la ligne. Enfin on fait avancer **PTR\_EEPROM** de deux octets pour pointer la donnée suivante. **Page 15**

## ➤ Enregistrer un spectre chromatique.

Plus étoffée que la procédure d'affichage, `Spectroscope_Chromatique()` qui enregistre les données colorimétriques en EEPROM doit enchaîner plusieurs opérations visant à placer la sonde en posture idoine, puis effectuer le balayage circulaire de filtrage et enregistrer les valeurs sans oublier de revenir à la configuration *Stable Transversal*.

```
void Spectroscope_Chromatique() { // La consigne moteur est dans Lecture_Echo.
① if (Dernier_PGM_actif == 2) {
②   Configuration_stable_LASER(); PTR_Apprentissage = 603;
③   Lecture_Echo = 211; // Positionne sur "transparent".
④   for (byte l=1; l < 8; l++) {
⑤     Mouvoir(0, Lecture_Echo); delay(200);
⑥     Lecture_Echo = Lecture_Echo + 34; // Aller à la couleur suivante.
⑦     Lumiere_ambiante = analogRead(Entree_Luxmetre);
⑧     Ecrire_un_Entier_en_EEPROM(PTR_Apprentissage, Lumiere_ambiante);
⑨     Configuration_stable_LASER(); Caractere = '(';
    else Erreur_N_BIP(21);}
}
```

Examinons pour commencer la Fig.135 qui représente le **Fémur** de la **Jambe A** vu de dessus en situation de départ quand la sonde est en posture *Stable Transversal*. Comme il sera précisé dans le **TOME 4** la cellule photorésistante sera placée pas loin de l'axe de symétrie longitudinal du châssis. Ses fils de liaison électriques seront coudés pour qu'elle soit en orientation radiale par rapport à l'axe de rotation de la **Hanche Jambe A**. Comme représenté sur la Fig.135 la cellule **C** se trouve sous le secteur bleu foncé. Il faut donc faire tourner en rotation **R** pour amener en premier le secteur transparent au dessus de la cellule photorésistante. Par des orientations manuelles, chaque secteur coloré a été placé bien au dessus de **C**.

Fig.134

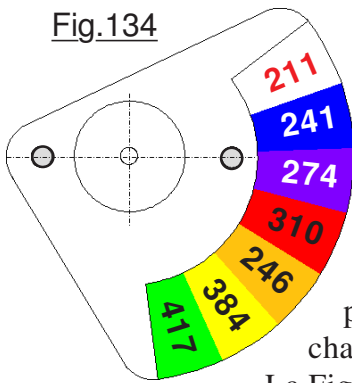
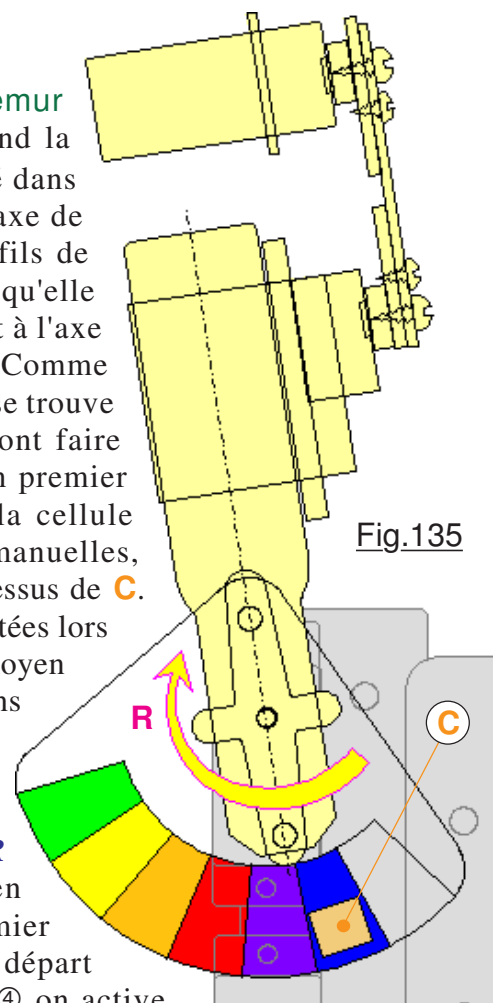


Fig.135

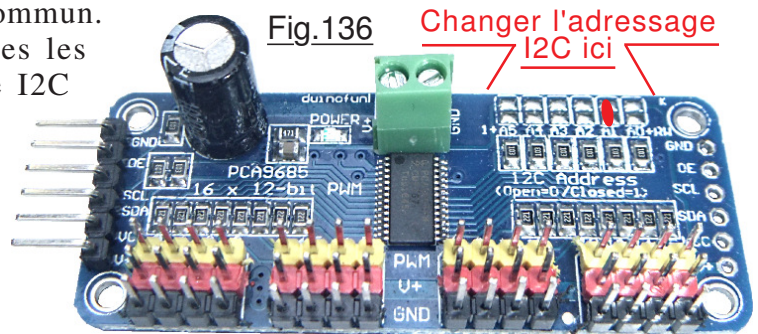


La Fig.134 résume les valeurs des **consignes** notées lors de la manipulation. Un rapide calcul montre que l'écart numérique moyen entre deux **consignes** successives avoisine la valeur **+34**. Nous avons tous les éléments pour agencer `Spectroscope_Chromatique()`. Première action en ① on vérifie la **posture préalable**. Si elle n'est pas correcte on déclenche une **alerte n°21**. Si le test est positif, on commence en ② par adopter la posture *Stable LASER* qui permet la libre rotation de la **Jambe A**. Le pointeur en EEPROM est alors positionné à **603** l'adresse relative du premier enregistrement. En ③ on impose à la **consigne** l'orientation de départ correspondant à la luminosité totale. (**Consigne 211**.) Puis en ④ on active une boucle qui va effectuer sept itérations. La première action consiste en ⑤ à placer la **Hanche** en orientation correspondant à **Lecture\_Echo**. Un délai de 0,2S laisse le temps au servomoteur d'atteindre l'angle de consigne. Puis en ⑥ on prépare la valeur de la **consigne** suivante. Pour balayer intégralement les sept secteurs colorés il faut entre chaque pas ajouter **+34** à **Lecture\_Echo**. L'instruction ⑦ effectue une numérisation sur l'**Entree\_Luxmetre** et affecte la valeur à la variable de type **int** désignée par **Lumiere\_ambiante**. C'est l'instruction ⑧ qui fait usage à la procédure dédiée `Ecrire_un_Entier_en_EEPROM()`. Notez que par principe toute procédure d'écriture met à jour le pointeur d'adressage sur la prochaine cellule EEPROM disponible. Cette procédure utilise comme pointeur **PTR\_Apprentissage**, raison pour laquelle c'est cette variable qui avait été initialisée en ②. Quand la boucle **for** a effectué les sept itérations imposées par l'index de comptage **l**, la procédure s'achève et la sonde est remplacée par ⑨ en posture *Stable Transversal*. **Page 16**



### 31) 28/11/2017 : Expériences sur réseau I2C (MJD 58085)

L'une des spécificités les plus appréciables du C++ d'Arduino est son aptitude à gérer un réseau de périphériques sur une ligne au standard I2C. Dès que l'on envisage en informatique de loisir une application un tant soit peu riche en périphériques, inévitablement nous serons confrontés à l'utilisation d'un BUS I2C. Heureusement pour nous, le C++ du compilateur intègre la bibliothèque **<Wire.h>** et les fournisseurs de modules électroniques accompagnent leurs productions de bibliothèques complémentaires (*Library.*) adaptées à leurs références. C'est le démonstrateur **P15\_Plusieurs\_modules\_I2C.ino** qui sera chargé dans un premier temps de nous montrer comment faire cohabiter plusieurs esclaves sur le bus commun. Avant de poursuivre, on s'informe de toutes les explications sommaires relatives à la ligne I2C proposées dans la **Fiche n°22**. Sans que nous n'y avons apporté une attention particulière, le bus de dialogue était déjà occupé par un module : Le multiplexeur de sorties pour les servomoteurs. La documentation de ce produit commercial indique que par défaut en sortie d'usine l'adresse de ce périphérique est 40HEX. (*01000000 exprimé en Hexadécimal.*) Sachant que ce module peut être chaîné facilement avec d'autres compères, pour en changer l'adresse I2C il suffit de forcer des "0" sur l'un des 6 "Bits" disponibles dans la zone qui sur la Fig.136 est repérée en rouge. Pour l'application en cours nous n'aurons pas besoin d'effectuer cette petite modification matérielle, autant vous l'éviter.



#### ➤ Une boussole magnétique sur GEKERT.

Consultons la **Fiche n°23** qui décrit le tout petit module magnétique qui sera ajouté sur le bus I2C de la sonde et qui va fonctionner en esclave avec le démonstrateur. Basée sur l'effet Hall, la "puce électronique" contient matériellement deux périphériques indépendants pouvant dialoguer sur le bus I2C avec pour adresses respectives 42HEX et 43 HEX. Chaque entité mesure séparément la valeur du champ magnétique dans une orientation cartésienne. Puis un module de calcul combine les deux valeurs pour déterminer l'orientation des "lignes de force" pour les traduire sous forme d'un CAP. La Fig.137 présente une copie d'écran quand on a envoyé à JEKERT la commande "o\*" qui retourne l'orientation magnétique du petit robot.

Fig.137

```
o* >
Orientation : 215°
!PGM o OK!
o* >
Orientation : 200°
!PGM o OK!
```

Initialement pour appréhender ce petit module, la bibliothèque spécialisée **hmc6352.h** avait été intégrée dans le programme d'évaluation. Mais l'utilisation actuelle sur la sonde n'en a pas besoin, il faut juste inclure les routines de gestion de la ligne I2C **<Wire.h>** qui de toute façon est dès le début indispensable pour dialoguer avec le multiplexeur PCA9685.

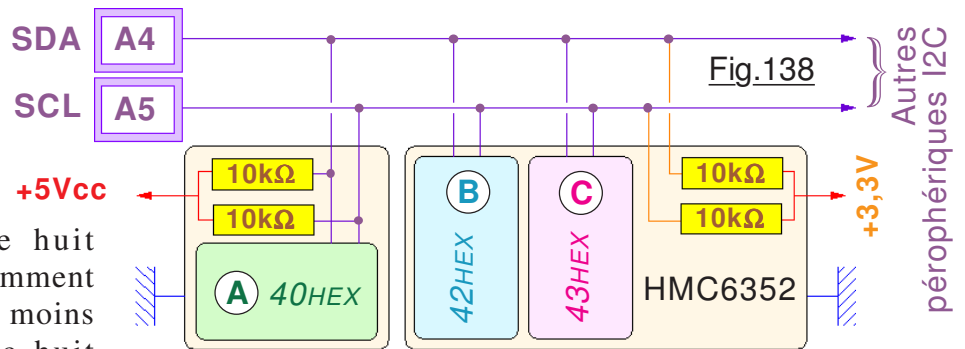
#### ➤ Conflits d'adressages sur une ligne I2C.

Interroger une personne pour dialoguer quand on est en présence d'une classe, implique de désigner de façon univoque celle à laquelle on s'adresse. Par exemple l'institutrice qui sollicite l'une de ses gamines par : **"Alice tu veux bien venir au tableau s'il te plait"** va avoir la surprise de constater que cinq de ses fillettes se lèvent pour obtempérer. Nous sommes en présence d'un conflit d'adressage verbal. Quand une "séquence Maître" sollicite un périphérique sur la ligne commune I2C, elle précise en premier l'adresse, puis la nature de la réaction attendue. Si deux entités ont une adresse identique, elles vont répondre simultanément et les données vont mutuellement polluer leur accusé de réception par le phénomène Dominant/Récessif. Le Maître recevra alors du "bruit binaire incompréhensible". Aussi, **il est IMPÉRATIF** de s'assurer **que tous les modules branchés en parallèle sur une ligne I2C soient conditionnés à des adresses matérielles différentes**.

Considérons la Fig.138 qui représente l'assemblage électronique actuel quand on branche simultanément le multiplexeur de motorisation PCA9685 et la petite boussole HMC6352.

Si vous étudiez à nouveau la Fig.22 vous constaterez qu'il n'a pas été nécessaire d'ajouter des résistances "Pull Up" au module PCA9685 pour forcer le niveau haut +5Vcc. Généralement les fournisseurs de modules prêts à l'emploi intègrent tout ce qu'il faut sur leur circuit

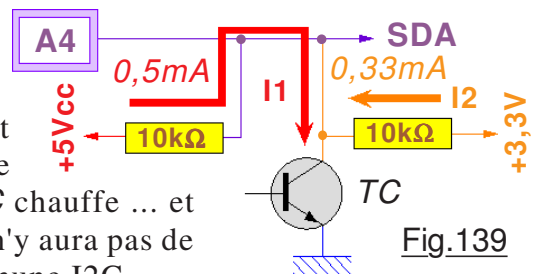
imprimé pour que l'utilisateur n'ait strictement rien d'autre à ajouter dans son application. Dans ce but, le module PCA9685 dispose de résistances de  $10k\Omega$  entre le  $+5V_{cc}$  et les deux lignes **SDA** et **SCL**. Hors ce petit module est prévu pour coexister avec plusieurs autres. Supposons que l'on chaîne huit PCA9685 en ayant bien évidemment modifié les adressages sur au moins sept d'entre eux. C'est donc huit résistances de  $10k\Omega$  qui vont se trouver montées en parallèle soit une résultante de  $1,25k\Omega$ . Dans ce cas, les transistors à collecteur ouvert devront drainer un courant de 4mA au lieu de  $50\mu A$ . Il n'y a aucune raison de s'inquiéter. Ces transistors fonctionnant en "saturé/bloqué" pourraient facilement commuter des courants cent fois plus importants. (*Sans compter qu'à ce pris on pilote cent vingt huit servomoteurs ... sacré machine !*)



### ➤ Conflits électroniques sur la ligne I2C.

Observant plus attentivement la [Fiche n°23](#) de la boussole HMC6352 on déduit que les deux adresses des entités **B** et **C** sont différentes de celle du module **A**. Sans avoir à modifier l'adresse du multiplexeur de motorisation, les trois entités peuvent dialoguer sans conflit d'adressage sur le bus I2C. Si le Maître sollicite l'un des trois, ce sera le seul à répondre et la cacophonie informatique est évitée. Beaucoup de périphériques sont conçus pour fonctionner en  $3,3V_{cc}$  régulé. C'est la raison pour laquelle les cartes Arduino anticipent en intégrant un régulateur  $+3,3V_{cc}$  que l'on peut utiliser à notre guise. Comme précisé expressément sur la Fig.1 de la [Fiche n°23](#) la minuscule carte électronique de la boussole **DOIT ÊTRE ALIMENTÉE EN  $+3V_{cc}$** .

Du coup, ses deux résistances intégrées de  $10k\Omega$  reçoivent "un contre courant" arrivant du bus qui est déjà forcé à  $+5V$  par le module PCA9685. Cette configuration ressemble fortement à un conflit électronique, et l'on pourrait craindre une surcharge du HMC6352. Rassurez-vous, ce n'est qu'une apparence, il n'en est strictement rien. Les transistors de commutation **TC** de ce dernier se trouvent dans la situation de la Fig.139 et doivent commuter la somme des courants **I1** et **I2** soit  $0,83mA$ . À ce régime il faudra placer la sonde en plein soleil pour que **TC** chauffe ... et encore ! On peut donc brancher sans problème ce module, il n'y aura pas de conflit électronique à craindre sur la ligne de dialogue commune I2C.



**NOTE :** Autant inverser le  $+5V_{cc}$  et **GND** peut endommager définitivement un module électronique, autant inverser **SDA** et **SCL** restera strictement sans effet mis à part des écrasements de données de type "Dominant/Récessif". Seule pénalité à ce type d'incident : Le dialogue I2C ne pourra pas s'établir correctement et le module en défaut restera inerte.

### ➤ La cohabitation I2C logicielle.

Trop impatients de voir fonctionner la petite boussole, vous avez rapidement branché le petit module rouge et téléversé le démonstrateur [P15\\_Plusieurs\\_modules\\_I2C.ino](#) qui fait vivre tout ce petit monde électroniqueinformatiquecorobotique. Magique, quand on tourne dans tous les sens le module, il ne perd pas trop le Nord à condition qu'il soit bien à l'horizontale. Manifestement cette manipulation démontre qu'il est possible de faire cohabiter au moins trois entités sur un bus I2C. Nous allons dans ce chapitre passer en revue les étapes et les détails logiciels qui aboutissent à ce résultat très séduisant, au prix pour le programmeur d'un minimum d'efforts. Cette simplicité à mettre en œuvre des systèmes d'une complexité technique vertigineuse résulte de programmes "faramineux" qui font tout le travail pour nous, que ce soit câblé en local sur le module électronique, ou par l'entremise de bibliothèques spécifiques bien pensées. Remercions au passage les Ingénieurs qui ont développé ces aides rendant évident et ludique des domaines qu'il y a peu de temps relevait de laboratoires pourvus de très hautes technologies, pratiquement de la science fiction ...

**P**remière étape à toute utilisation d'un quelconque périphérique : Les déclarations et les directives. Si l'un des modules branchés sur l'ATmega328 doit dialoguer sur un BUS I2C, nous savons déjà qu'il faut déclarer la bibliothèque `#include <Wire.h>` et que cette dernière en standard va s'octroyer **A4** et **A5** et qu'ils devront respectivement se brancher sur **SDA** et **SCL**. Pour **P15** c'est la directive ① qui se charge d'informer le compilateur.

```
① #include <Wire.h>
② #include <Adafruit_PWMServoDriver.h>
③ Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
④ #define Lecture_Adresse 0x41 // "A" en hexadécimal ...
⑤ #define Adresse_du_HMC6352 0x21 // Adresse pour le capteur Magnétique.
```

Puis on peut déclarer dans un ordre quelconque les bibliothèques spécifiques. Dans notre cas, historiquement nous avons commencé par celle du multiplexeur déclarée en ②. En général, il faut "instancier" un périphérique matériel pour qu'il soit connu de sa bibliothèque. C'est les cas pour la carte PCA9685. Aussi, immédiatement après avoir déclaré la "library" en ③ on déclare le module connecté à Arduino. Connue de l'**IDE**, il n'est pas obligatoire d'inclure une bibliothèque pour la boussole HMC6352. Par contre, quand on désire faire référence aux "octets internes" du module électronique, on devra utiliser des identificateurs. C'est la directive en ligne ④ qui se charge de déclarer `0x41`, soit la lettre "A" en code ASCII. ATTENTION, cette information ressemble à une adresse I2C. Rien à voir. Elle désigne une cellule binaire interne au circuit électronique qui permettra de récupérer la valeur du CAP codée en décimal sur deux octets. (*Intrinsèquement un type float.*) Enfin en ⑤ on définit une autre constante d'adressage interne dont aura besoin le logiciel.

**D**euxième étape : Réserver de la place en mémoire pour que le logiciel puisse traiter des variables et leur affecter un identificateur. Pour nos trois modules périphériques et surtout pour leurs traitements informatiques, on doit déclarer des variables. Voici les informations relatives à la petite boussole HMC6352 :

```
(Voir plus avant.) int ① byte LSB; // Utilisé par le capteur magnétique.
                        ② byte MSB; // Utilisé par le capteur magnétique.
                        ③ float CAP_construit;
```

La valeur calculée du CAP est délivrée sous forme de deux octets qui seront concaténés (*Mis bout à bout.*) pour obtenir une valeur numérique complète. Pour préserver ces deux octets temporaires, les programmes fournis pour évaluation de la boussole électronique réservent deux **byte** ce que font les lignes ① et ②. Enfin le réel **CAP\_construit** sera affecté en ③ à la valeur calculée du CAP.

**Q**uand les constantes, les variables et les bibliothèques disponibles sont mises en place, on complète le logiciel de l'application par les séquences qui exploitent toutes ces ressources. Pour la boussole HMC6352 le gros du travail se résume à une procédure copiée presque sans changement sur celle qui accompagne le composant. En réalité, sur tous les programmes que vous trouverez sur Internet, l'adresse **Adresse\_du\_HMC6352** est décrite sous forme d'une variable de type **int** qui lors de sa déclaration est initialisée à `0x42`. Ensuite pour l'utiliser on la divise par deux par un décalage logique `>> 1` pour devenir `0x21`. Autant éliminer la variable ce qui économise deux octets dans la RAM dynamique. Éliminez aussi le calcul `>> 1`, ce qui réduit encore le code. Ainsi déclarée en constante valant directement `0x21`, on gagne 30 octets dans le code objet. Examinons maintenant la procédure suggérée sur Internet pour mesurer la valeur du CAP magnétique retournée dans **CAP\_construit**. Nous verrons plus avant que cette procédure est encore loin d'être optimisée, on peut faire bien mieux. Ceci étant précisé, elle a le grand mérite de bien montrer comment procéder.

**C**ompilant le démonstrateur, lorsque l'on ajoute l'appel à cette fonction `"o"` dans la liste des commandes à un caractère, la procédure de mesure et de calcul ainsi que celle d'affichage, le code objet enfile d'environ 236 octets. (*Pour avoir cette information on compile sans, puis avec la nouvelle fonction.*) C'est au final très peu pour compléter les performances de JEKERT. Cependant, le modules fonctionnel entièrement ajouté ne comporte au total que dix sept instructions, ligne d'appel `"o"` comprise. C'est que des lignes comme `Wire.beginTransmission()` par exemple font appel au contenu de la bibliothèque et exigent rapidement des quantités d'octets pour le programme binaire. Il faut donc toujours avoir présent à l'esprit qu'une simple ligne de source faisant appel à du "code externe" peut engendrer un taux d'inflation important.



Avant de décortiquer le fonctionnement de la procédure `Mesure_orientation_magnetique()` il me semble utile d'ouvrir une petite parenthèse consacrée à du vocabulaire traditionnel. Pour illustrer ce propos, nous allons considérer en Fig.140 un cas concret. Par exemple on va supposer qu'une variable nommée `CAP_construit` est de type `int`, c'est à dire qu'elle contient 16 BITS. Chaque BIT peut se trouver à l'état "0" ou à l'état "1". Quel que soit son ordre dans la représentation du nombre rangé dans `CAP_construit` les "0" vaudront "0", c'est à dire pas grand chose. Par contre, les "1" en fonction de leur position représentent des valeurs de plus en plus grande quand on les examine de la gauche vers la droite. Dans le système décimal chaque "décalage" multiplie par dix, le tout

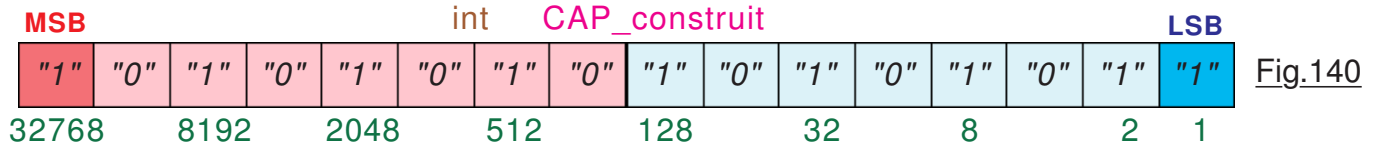


Fig.140

premier "1" valant la valeur unitaire. En binaire c'est exactement pareil, sauf que la base étant 2 au lieu de 10, on multiplie par 2. Le poids de chaque BIT vaut donc 1, puis 2, puis 4, 8 etc. En conclusion, celui le plus à droite est le moins significatif. Il est nommé **Least Significant Bit** dans la littérature spécialisée et symbolisé par **LSB**. Celui le plus à gauche devient le **Most Significant Bit** et se représente par le sigle **MSB**. Il se trouve que dans les exemples de programme donnés pour la boussole les auteurs utilisent **LSB** et **MSB**, sauf que dans ce cas ces deux identificateurs trompeurs ne font plus référence à des BITS mais à des OCTETS. Ce n'est pas tragique, mais durant l'analyse avec **LSB** il faudra penser **Least Significant Octet** et avec **MSB** raisonner sur l'Octet de poids fort, **Most Significant Octet** avec pour corollaire le fait qu'un mot binaire sur 16 BITS peut être considéré comme deux OCTETS concaténés c'est à dire placés l'un contre l'autre MSB à gauche, LSB à droite.

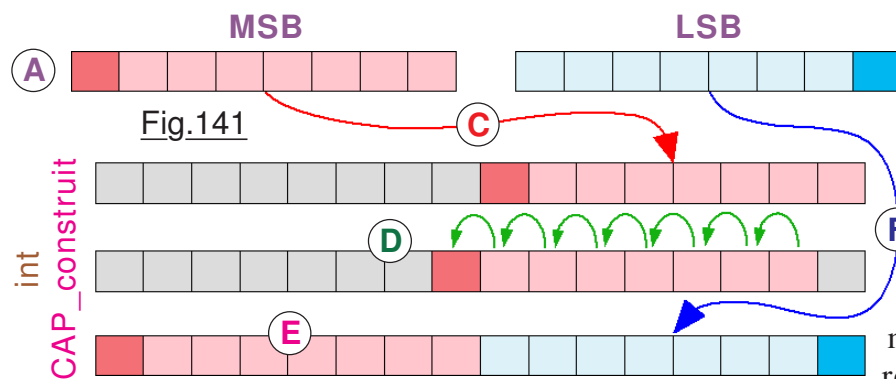
### ➤ Exemple de protocole de dialogue sur un bus I2C.

```
void Mesure_orientation_magnetique() { // Saisie de la valeur du cap magnétique.
① Wire.beginTransmission(Adresse_du_HMC6352);
② Wire.write(Lecture_Adresse); // Commande de lecture.
③ Wire.endTransmission();
④ delay(6); // Le HMC6352 a besoin d'un délai de 2ms.
⑤ Wire.requestFrom(Adresse_du_HMC6352, 2);
⑥ // Saisie des deux octets de donnée "MSB" et "LSB".
⑦ MSB = Wire.read(); LSB = Wire.read();
⑧ CAP_construit = (MSB << 8) + LSB;
```

Première action, la procédure `Mesure_orientation_magnetique()` amorce en ① un dialogue c'est à dire qu'elle signale que le périphérique habitant à `Adresse_du_HMC6352` de la rue I2C doit se mettre à l'écoute, qu'il va recevoir une instruction. Dans ce but, il devra surveiller le signal d'horloge sur **SCL** et les données sur **SDA** pour lire le code instruction. Puis le Maître par la ligne ② "écrit" sur le bus le caractère ASCII "A" qui en protocole HMC6352 demande de retourner la valeur du CAP magnétique actuel. Enfin en ③ le Maître informe l'esclave qu'il peut parler à son tour. Pour laisser le temps à l'esclave de décoder le message la documentation précise qu'il faut faire une pause de 2mS. Avec un délai de 6mS en ④, ce qui ne change pas grand chose à la rapidité de réaction de la sonde, on prend une marge de sécurité. Puis en ⑤ le logiciel prévient qu'il désire récupérer successivement les deux octets pour "A" dans la "puce électronique". C'est ce que fait l'instruction en ligne ⑦. C'est ici que la remarque donnée en préambule prend tout son sens. Ce sera d'autant plus vrai que dans le programme optimisé, la variable `CAP_construit` sera réellement déclarée comme un `int` donc une grandeur binaire codée sur 16BITS. Pour la suite des explications, considérez-donc que la déclaration définitive en tête de programme est :

```
int CAP_construit; // CAP calculé codé sur un entier et non comme un réel.
```

Considérons la Fig.141 qui résume les opérations effectuées par la procédure et dans le circuit intégré. Le premier `MSB = Wire.read();` de la ligne ⑦ impose au périphérique esclave d'envoyer l'OCTET de poids fort, la valeur est alors préservée dans `MSB`. (*Action A*) À la deuxième instruction de la ligne ⑧ c'est le poids faible du CAP magnétique qui est transmis sur la ligne I2C, puis rangé dans `LSB` par le programme. (*Action B*) La valeur calculée du CAP magnétique dans l'électronique interne au circuit intégré de la boussole a été recopié à l'extérieur de l'esclave, et se



trouve dans l'électronique du Maître dans deux cellules de RAM qui pour le programme sont des emplacements repérés par **MSB** et **LSB**. En l'état ces données sont complètement inutilisables car la valeur complète est coupée en deux morceaux de taille OCTET, il faut reconstituer la grandeur sur 16 BITS

pour pouvoir l'afficher sur le moniteur de l'IDE. C'est l'instruction de la ligne ⑧ qui se charge de ce travail en plusieurs étapes. Décortiquons le processus qui dans son intégralité s'écrit :

$$\text{CAP\_construit} = (\text{MSB} \ll 8) + \text{LSB};$$

- **CAP\_construit** = (MSB : Réalise **C** c'est à dire que les huit BITS de **MSB** vont dans les huit BITS de poids faible de la variable **CAP\_construit**. Les bits de poids forts pour le moment sont quelconques et coloriés en gris sur la Fig.141 car ce sont des résidus d'opérations "anciennes".
- **<<** : Effectue un décalage logique à gauche du contenu intégral de la variable **CAP\_construit**. Les 16 BITS se décalent d'une position à gauche, le premier à droite est forcé à "0" et colorié en gris car non significatif pour le moment. Ce décalage (*SHIFT left*) est représenté en **D**.
- **8** : Le décalage à gauche est répété huit fois, ce qui fait que **MSB** qui avant ces décalages était entièrement "à droite" se retrouve entièrement "à gauche" en **E**. C'est la place que doit occuper tout OCTET de poids fort bien élevé. L'OCTET de poids faible ne contient que des "0" non significatifs.
- **+** **LSB** : Ajouter **LSB** à **CAP\_construit** revient à "transporter" comme symbolisé en **F** l'OCTET en poids faible, car avant l'addition des huit premiers BITS ne contiennent que des "0".

Il ne reste plus qu'à afficher le résultat ; informatiquement une simple formalité :

```
void Indique_orientation_sur_USB() {
  ① Mesure_orientation_magnetique();
  ② Serial.println(); Aff_TEXTE_EEPROM(1004,14);
  ③ Serial.print(CAP_construit / 10,0); Serial.println(char (186));}
```

En ligne ① l'esclave effectue la mesure, puis la valeur est placée dans la variable **CAP\_construit**. Pour pouvoir bénéficier de la meilleure définition possible sur le capteur, la valeur est fournie multipliée par dix en degrés, donc sur quatre chiffres significatifs. La ligne ② fait passer le curseur écran à la ligne suivante sur le moniteur vidéo. Puis le texte "Orientation : " est affiché. En ligne ③ on commence par diviser **CAP\_construit** par 10. Enfin ,0 impose au programme de n'afficher aucun chiffre après la virgule. Notez au passage que si vous voulez les dixièmes de degré, il suffit de remplacer le format d'affichage par ",1". Pour notre application c'est inutile, sans compter le fait que ce chiffre ne sera pas fiable puisque les valeurs ne sont garanties qu'à 0,5° près. Enfin la ligne d'instruction se termine par l'affichage du caractère **char (186)** qui sur le moniteur série de l'IDE est le symbole du "°". (*Degré angulaire soit 360 de ces unités sur un tour complet.*)

### ➤ Optimiser le programme de traitement de la boussole.

Quelle que soit l'origine d'un exemple que l'on trouve sur Internet, le code source n'est pas forcément idéal et ce pour plusieurs raisons. Il peut avoir pour origine la gentillesse d'un programmeur amateur qui a résolu un problème et qui désire en faire profiter la collectivité. Amateurs avertis et débutants se retrouvent sur les sites spécialisés, tous contribuent à leur façon à l'édifice. Si celui qui a mis en ligne un exemple qui fonctionne est un débutant, son "Sketch" sera "brut de décoffrage". Si vous avez plus d'expérience, alors apporter de l'élégance au premier jet, améliorez ce dernier et à votre tour augmentez la richesse de ce qui se trouve en magasin. Les exemples issus de fournisseurs ne sont pas forcément optimisés. Car le facteur commercial incite à publier le plus rapidement possible, la concurrence est forte. Bien entendu, vous aurez toutes et tous compris qu'il ne s'agit surtout pas de critiquer, seulement d'apporter une amélioration quand c'est possible. Du reste on s'inscrit dans cette ligne de conduite qui consiste à toujours chercher à

*optimiser à outrance un programme qui fonctionne*, c'est un principe qui a été énoncé avec ferveur le jour de notre mémorable arrivée à l'entreprise NDRMSE. Nous allons dans ce qui suit optimiser `Mesure_orientation_magnetique()` et ainsi gagner encore de précieux octets. Et surtout, par une analyse soignée des variables, ce qui normalement doit se faire au moment où on les intègre au logiciel, améliorer l'élégance de cette procédure qui effectue un travail remarquable. Pour ne pas chambouler tous les autres démonstrateurs actuellement terminés, seul le programme définitif `P31_PGM_COMPLET_pour_Raquette.ino` bénéficiera de ces quelques optimisations.

La première optimisation consiste à déclarer la variable `CAP_construit` comme étant un `int`. Elle ne consommera que deux octets au lieu de quatre dans la RAM dynamique. Vous pourriez penser que c'est du "gagne petit". C'est oublier que maintenant le microcontrôleur n'aura plus qu'à décaler 16 BITS au lieu de 32, on gagne aussi du code objet. N'oubliez-bas que pour effectuer une opération, l'ATmega328 doit commencer par amener les données dans son unité arithmétique et logique. Il faut charger tous les octets, faire l'opération, stocker à nouveau en RAM le résultat. Mal choisir sa variable engendre un effet boule de neige sur la taille du programme, avec pour conséquence un ralentissement de son exécution. Bref, réfléchissez bien au moment de définir des variables ...

```
void Mesure_orientation_magnetique() {
    ... } Le début de la procédure reste inchangé.
    // Saisie des deux octets de donnée "MSB" et "LSB".
    ① MSB = Wire.read(); LSB = Wire.read();
    // ===== Traite la donnée et calcule l'orientation. =====
    ② // Le CAP lu en I2C sur deux octets est concaténée, "MSB" décalé 8 fois à gauche.
    // ATTENTION ne pas permuter les deux termes de l'addition.
    ③ CAP_construit = (Wire.read() << 8) + Wire.read();}
```

Notons que le module logiciel `Wire.read()` fournit par la bibliothèque n'est pas une procédure, mais une **FONCTION**. C'est à dire qu'il **RETOURNE une valeur typée**. On peut donc le placer dans toute instruction à un endroit qui attend une donnée de son type. Comme `Wire.read()` est une fonction qui retourne une donnée sous forme d'un `byte`, on peut donc parfaitement remplacer la variable `MSB` ou la variable `LSB` par `Wire.read()`. C'est exactement ce que fait la ligne ③. Du coup la ligne ① ne sert plus à rien, on l'enlève du programme source. Les deux variables `MSB` et `LSB` ne sont plus utiles non plus, et sont également effacées des déclarations. À résumer l'instruction de la ligne ③, on peut la simplifier par `CAP_construit = Valeur 1 + Valeur 2`. Hors mathématiquement une addition est commutative, calculer `CAP_construit = Valeur 2 + Valeur 1` doit donner un résultat identique. Et bien pas en informatique, car une instruction telle que celle de la ligne ③ est bien plus qu'un calcul mathématique. Se sont plusieurs opérations mathématiques et logiques **qui doivent se succéder dans un ordre bien précis**. Donc nous n'obtiendrons absolument pas le même résultat avec `CAP_construit = Wire.read() + (Wire.read() << 8);`

Reste à résoudre un petit aléas qui résulte de la façon dont le compilateur traite les affichages et en particuliers ceux de la voie série USB du moniteur de l'**IDE**. Si on laisse inchangée la procédure `Indique_orientation_sur_USB()`, l'affichage de la valeur du CAP sera étrange et incompréhensible, car la donnée numérique est un entier `int` au lieu d'un `float`. Pour que tout puisse rentrer dans l'ordre il suffit d'imposer à la procédure d'affichage `Serial.print()` de considérer que la variable `CAP_construit` n'est pas un entier, mais un réel. La correction de type est "soulignée" en

<i>Transposition de type sur un affichage :</i>	<pre>void Indique_orientation_sur_USB() {     Mesure_orientation_magnetique();     Serial.println(); Aff_TEXTE_EEPROM(1004,14);     Serial.print(<u>(float)</u> CAP_construit/10,0); Serial.println(char (186));}</pre>
---	---

rouge dans le listage. Ces quelques modifications font gagner deux octets dans la RAM dynamique et surtout 18 Octets dans le programme objet. Additionné à toutes les optimisations effectuées, le bénéfice devient significatif. Au risque de rabâcher, c'est plus "un mode de comportement" qu'un principe. Affiner un programme devrait systématiquement conclure un développement qui donne satisfaction. C'est la clef pour arriver à traiter un nombre considérable d'actions avec un microcontrôleur en saturant ses possibilités, et éviter de se voir obligé éventuellement à passer à une carte électronique plus puissante ...



Disponible dans le dossier <Galerie d'Images>, l'Image 12.JPG permet de situer l'emplacement de la boussole magnétique sur la sonde. Dans le Tome 4 seront précisé en détails l'implantation des connecteurs, les options à prendre en compte, le positionnement des divers trous etc. Dans l'attente de sa publication, vous allez vous contenter de prendre le petit module en main et de l'orienter à convenance. Néanmoins, un petit regard sur Image 15.JPG montre comment est installé le petit circuit imprimé sur la plaque qui supporte le multiplexeur et Image 16.JPG la façon de relier le connecteur au circuit électronique principal.

### ➤ La mesure pragmatique d'un CAP magnétique.

Contrairement au GPS qui permet de déterminer le CAP géographique du mobile qui s'en sert, le CAP magnétique par nature fournit une orientation fautive par rapport à la carte. Le magnétisme d'une planète, quand il est d'une intensité suffisante pour le rendre mesurable, n'est pas homogène, les lignes "de force" se déforment de façon considérable en fonction de la nature des masses rocheuses environnantes. De ce fait, une boussole magnétique prend par rapport à la référence géographique des orientations "quelconques". Considérons la Fig.142

qui représente les "lignes de force" du champ magnétique dans une région vue par dessus. Une boussole n'indique pratiquement jamais réellement la direction des points cardinaux, mais ne s'oriente qu'en fonction des "lignes de forces" du champ magnétique planétaire local. Hors ce dernier peut diverger du nord géographique jusqu'à 80° en fonction des "mascons" locaux profondément enfouis dans le sol, surtout si ces derniers sont fortement magnétiques et "aspirent" les lignes de force et ainsi peuvent dévier considérablement le "flux magnétique". Qu'elle soit constituée d'un petit aimant où de cellules à effet Hall, une boussole s'aligne en 1 sur les lignes de champ magnétique qui sont relativement parallèles. Toutes les boussoles de la région adoptent une même direction. Dans cette zone, la déclinaison magnétique  $\delta$  qui traduit l'écart entre la direction du nord géographique N et celle indiquée par la boussole Nm est relativement constante. En 2 se trouvent des masses ferreuses qui "attirent et concentrent" le flux magnétique régional. Les directions indiquées par les boussoles sont alors beaucoup plus divergentes.

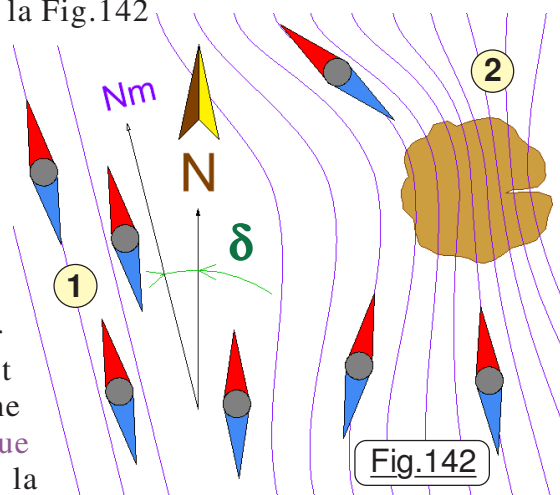


Fig.142

Deuxième source d'imprécision qu'il faudra impérativement prendre en compte : La pente du terrain sur lequel évolue JEKERT. Il est clairement précisé en Fiche n°23 sur la face (2/2) que la boussole ne donnera des résultats crédibles que si l'assiette du module est parfaitement horizontale. Si la sonde se trouve sur une pente à forte déclivité, le calcul du cap sera faussé. Une expérience amusante consiste à placer la sonde bien à l'horizontale et faire afficher son CAP. Puis incliner en ROULIS, en TANGAGE tout en vérifiant qu'en LACET elle pointe en permanence l'avant du vaisseau dans une direction constante. Aidez-vous du LASER par exemple. Vous pourrez ainsi évaluer les effets des inclinaisons transversales et longitudinales. Puis combinez les deux. Vous allez rapidement arriver à la conclusion qu'indiquer les dixièmes de degrés n'est pas vraiment pertinent.

Troisième source d'incertitude induite non pas par l'environnement, mais générée par le robot. Ce dernier comporte des masses métalliques, certaines sont ferreuses. Par ailleurs les servomoteurs intègrent dans leur technologie des aimants permanents qui joyeusement "diffusent" des champs magnétiques parasites qui faussent les lignes de force de la région où se déplace le mobile. Sur un petit avion de tourisme monomoteur à hélice, le tableau de bord est souvent équipé d'un gyroscope. Ce que les pilotes nomment le conservateur de CAP. Hors cet instrument doit être régulièrement calibré. Dans ce but, généralement placé un peu comme un rétroviseur d'automobile, un compas magnétique se trouve au centre du pare-brise. Hors le moteur sur ces avions constitue une masse ferreuse importante proche du compas et fausse les indications de ce dernier. Pour palier ce problème, la boussole de bord est accompagnée d'une table de corrections. Ainsi un pilote sérieux recalant son gyroscope compensera l'indication du compas, et ajoutera ou retranchera la valeur de la déclinaison magnétique du lieu qu'il survole. (Les cartes marines ou aviation permettent de la calculer.) Alors, pourquoi ne pas établir une correction de la boussole magnétique de JEKERT ? La technique à utiliser sera précisée dans le prochain chapitre, car nous disposerons d'une aide considérable apportée par la centrale gyroscopique.

L'avancement des études respecte assez bien les espérances du planning. Pour assembler les expériences scientifiques sur JEKERT, les dernières semaines ont été surchargées et les équipes ont largement dépassé les horaires légaux, elles ont dépensé sans compter. Aussi, la direction a décidé d'organiser un petit séjour de détente. Un petit vol transatlantique, et tout le monde se retrouve ébahi dans le musée de la NASA à contempler les machines fabuleuses qui ont permis, pour la toute première fois pour l'humanité, de poser le pied sur la Lune. Nous venons de pénétrer dans la salle qui expose les systèmes qui permettaient à Apollo de naviguer dans le vide sidéral. Le guide, un peu scolaire, nous explique que naviguer, c'est décider où l'on veut aller, et surtout comment y aller.

**N**aviguer, c'est l'action de déterminer en permanence où l'on se trouve, et s'orienter pour continuer à cheminer dans la bonne direction. Sur Terre c'est relativement facile, car l'on se contente de suivre des routes qui sont jalonnées de repères précis pour nous aider tout le long du voyage afin de nous situer. En automobile ce sont les panneaux indicateurs. En aviation ce sont des balises radio. (Remplacées actuellement par le système GPS) Quelle que soit la technologie utilisée, on constate dans ces systèmes de navigation que l'on se trouve tributaire d'installations au sol ou de satellites en orbite. Mais il existe des cas où l'on se trouve totalement isolé de toute assistance extérieure. Par exemple un sous-marin nucléaire pour des raisons de discrétion, une fusée car un téléguidage radar ne serait pas assez rapide pour réagir en temps réel, enfin, pour le cas qui nous occupe, la sonde JEKERT doit pouvoir naviguer en autarcie. Dans tous ces exemples, *l'autonomie totale réside dans l'inertie.*

➤ **La navigation inertielle.**

**Q**uelle que soit la situation d'un mobile, il sera soumis où qu'il se trouve dans l'Univers à l'inertie, phénomène totalement général et omniprésent. *L'inertie, c'est le refus que présente tout objet à voir son mouvement se modifier.* Pour changer la vitesse d'un corps quel qu'il soit, il faut forcer sur ce dernier. Réciproquement, si on force sur un objet il accélère dans la direction de

la sollicitation. L'idée fondamentale consiste donc à utiliser le phénomène d'inertie pour mesurer l'accélération subie par

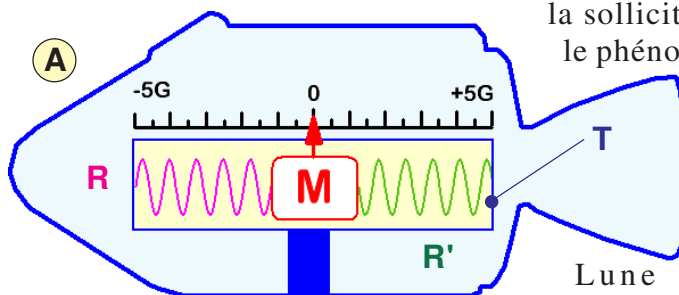
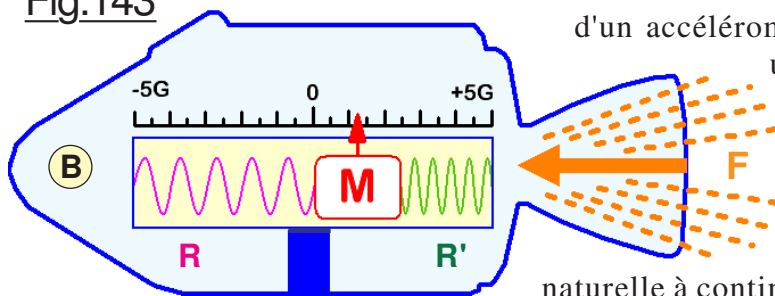


Fig.143



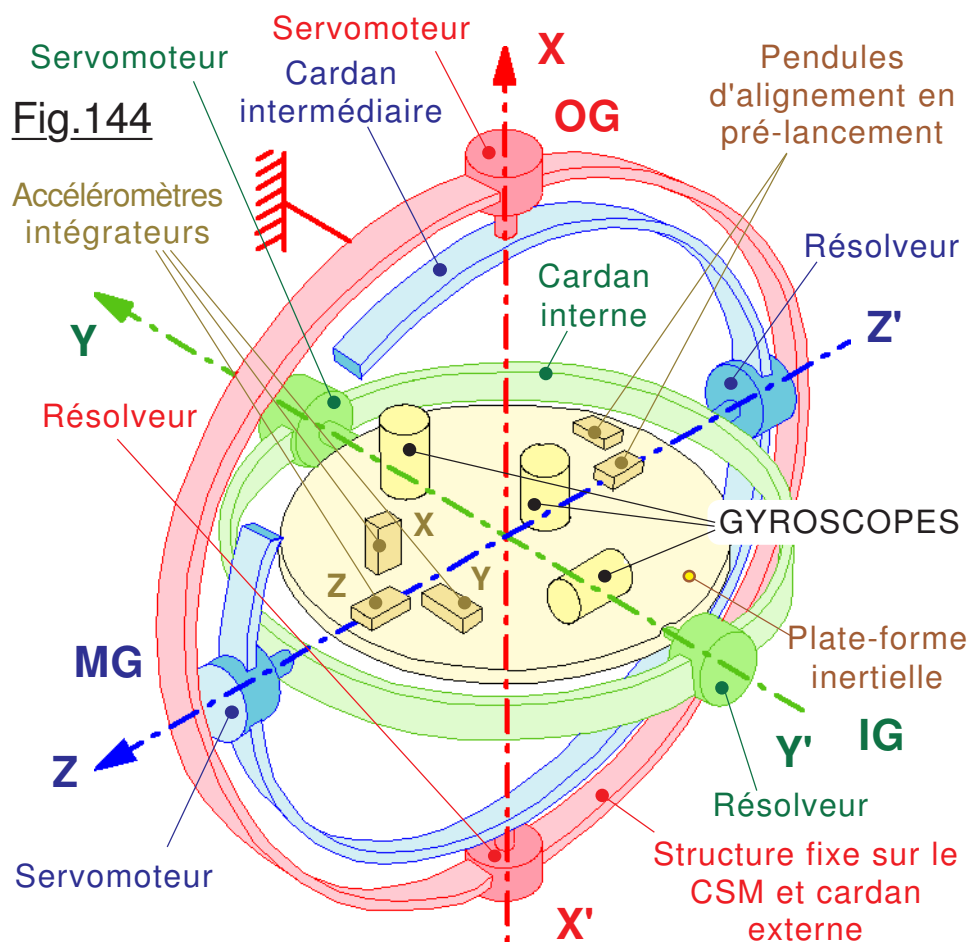
notre mobile. Pour illustrer ce propos, nous allons considérer le cas du vaisseau Apollo CSM qui à l'époque des vols lunaires habités devait se "débrouiller" seul quand il était isolé dans l'immensité du vide sidéral, ou caché derrière la Lune sans aucun secours possible des installations terrestres. Considérons la Fig.143 **A** qui montre le principe d'un accéléromètre. Une masse inerte **M** est centrée dans un guide tubulaire **T** au moyen des deux

ressorts **R** et **R'**. Allumons comme montré sur la Fig.143 **B** le moteur orbital, le vaisseau subit une poussée **F** et accélère vers l'avant. Le tube **T** accélère, mais la masse **M** par inertie présente une tendance naturelle à continuer d'un mouvement uniforme. Elle "refuse"

cette accélération. Le ressort **R'** se tasse, poussé par **T**. À son tour il force sur **M** jusqu'à ce qu'il exerce sur cette dernière une action suffisante. Du coup la masse **M** accélère et suit alors le vaisseau dans son mouvement, avec un petit retard dû au tassement de **R'**. Le ressort **R** pour son compte se détend un peu, mais il ne sera vraiment utile que pour des accélérations dans l'autre sens, quand le vaisseau accélérera à reculons avec les moteurs de manœuvre. Si nous avons soudé une flèche sur **M**, et qu'elle se déplace devant des graduations, il sera alors facile de lire directement la valeur de l'accélération, qu'elle soit vers l'avant ou vers l'arrière. Nous avons décortiqué ici le principe de fonctionnement de tout accéléromètre. Ceci dit, conçu comme celui de la Fig.143 notre appareil de mesure ne serait pas très efficace, car la masse et les deux ressorts seraient affectées d'oscillations bien gênantes durant les transitoires. Dans la pratique on utilise d'autres procédés, mais tous à cette époque étaient basés sur la présence d'une masse inerte qui agit sur un corps d'épreuve quelconque."

### ➤ La centrale inertielle d'Apollo.

Traiter de la navigation inertielle exigerait un livre plus gros qu'une encyclopédie tellement le sujet est complexe, tant au point de vue théorique que technique. Totalement hors sujet dans ce didacticiel, nous allons juste regarder dans la vitrine la petite boîte métallique qui a été ouverte pour montrer l'intérieur de la plateforme inertielle. Concevoir une centrale inertielle constitue un défi technique tellement complexe qu'à cette époque il a été estimé qu'étudier une machine spécifique à Apollo ne serait pas assez fiable. Aussi, bien que les appareils qui équipaient les missiles



intercontinentaux Polaris de l'époque présentaient de gros défauts pour naviguer dans l'espace, il a été décidé d'installer à bord ces ensembles qui prouvaient leur fiabilité sur les vecteurs intercontinentaux militaires. La clef de voûte du système est constituée par la **plate-forme inertielle** connue sous le signe IMU. (*Inertial Measurement Unit*) Le cœur d'une centrale à inertie est constitué comme montré sur la Fig.144 d'une **Plate-forme inertielle** sur laquelle sont liés trois **Accéléromètres intégrateurs** cartésiens. (*Orientés à angle droit les uns par rapport aux autres pour former un repère trirectangle*) Cette "table inertielle" est articulée sur trois joints de cardans **X'X**, **Y'Y** et **Z'Z** pour pouvoir

s'orienter librement dans toutes les directions par rapport à la structure du vaisseau. Sur la **Plate-forme inertielle** sont également liés trois **Gyroscopes** (*Mécaniques à l'époque d'Apollo*) eux même articulés "à la **Cardan**" pour pouvoir s'orienter librement dans l'espace. Des chaînes d'asservissement utilisant des **Résolveurs** commandent des **Servomoteurs** sur chaque axe. Les signaux issus des gyroscopes sont utilisés pour assujettir les servomoteurs de façon à ce que la cage (*Plate-forme*) conserve au cours du temps une orientation constante par rapport à l'univers. **Les trois accélérations sont prises en compte par des calculateurs ultrarapides pour en déduire les vitesses et les déplacements.** Nous disposons alors de l'orientation dans l'espace, et de la position actuelle du vaisseau par rapport à celle de départ quand la centrale inertielle a été initialisée.

Cette technologie qui relève de mécanismes d'horlogerie particulièrement précis était enfermée dans un boîtier antichoc, étanche et thermostaté. Il fallait pratiquement une heure pour qu'elle soit pleinement opérationnelle. Comme tout ce qui est mécanique, quelles que soient les précautions prises, il y a inexorablement des frottements "partout" où des pièces glissent les unes sur les autres. Par exemple les balais qui font transiter "l'électricité" vers des électroniques qui peuvent se mouvoir sur plusieurs tours, du glissement parasite dans les roulements des pivots ...

Conséquence de ces incontournables, la centrale inertielle dérivait lentement, et il était impératif de la recalibrer. Dans ce but, à bord du vaisseau était disponible un télescope pour pointer des étoiles à partir desquelles le calculateur de bord réinitialisait les données. *C'est pour apprécier à sa juste valeur les avancées technologiques phénoménales, que j'ai trouvé acceptable, d'encombrer avec deux pages "historiques" le cours linéaire de notre tutoriel.*



### 33) 30/11/2017 : Une centrale gyroscopique sur JEKERT (MJD 58087)

C'est probablement le module électronique le plus fabuleux intégré à la petite machine, qui lui alloue une performance technique presque magique. Missiles balistiques, vaisseaux Apollo lors des missions lunaires, les avions de ligne actuels, les sous-marins nucléaires, tous sont équipés de centrales inertielles qui leur permettent une autonomie totale pour effectuer leur navigation. Alors pouvoir équiper le petit robot d'un tel dispositif relève presque de la science fiction ... et pourtant ! Pour exactement 2,1• port compris, on peut s'offrir le summum de la technologie actuelle dont les caractéristiques de base sont proposées dans la [Fiche n°24](#) et la [Fiche n°25](#). Cerise sur le gâteau, il ne sera pas obligatoire d'aller consacrer cinq à six années d'études pointues au MIT pour pouvoir mettre en service ce fabuleux module. Avec la bibliothèque fournie, c'est pratiquement élémentaire.

Chapitre sur la boussole, nous avons apporté la preuve que trois périphériques étaient capables de cohabiter sur le réseau I2C. Avec le module MPU-6050 nous allons conforter cette certitude en y rajoutant une couche. En effet, c'est à l'adresse 68HEX que réside le circuit intégré qui intègre un calculateur ultra rapide. Nous avons de la chance, car cette adresse n'interfère avec aucune des trois autres habitations de la rue I2C. Donc la compatibilité est assurée. Le démonstrateur [P16\\_Plusieurs\\_modules\\_I2C\\_avec\\_MPU6050.ino](#) est directement issu de [P15](#) avec l'ajout on s'en doute de la petite centrale gyroscopique. C'est volontairement que je n'utilise pas dans ces lignes le vocable de *centrale inertielle*, car bien que le circuit comporte effectivement des accéléromètres et peut en fournir les valeurs, nous n'exploiterons dans cette application qu'une partie de toutes les informations disponibles. (*Simplification du logiciel oblige.*)

#### ➤ **Expérimentation du capteur gravitationnel.**

Simplifions au maximum de démonstrateur pour expérimenter les procédures d'utilisation du circuit MPU-6050, tant au point de vue matériel que logiciel. Nous adopterons pour les branchements électriques le schéma minimal indiqué en Fig.1 sur la [Fiche n°24](#). Vous pouvez constater en observant l'agencement du petit module que l'on retrouve certains aspects déjà découverts pour la boussole HMC6352 comme par exemple son fonctionnement interne en 3,3Vcc. Les deux broches **SDA** et **SCL** ne vont pas directement sur le bus I2C, des résistances de 1kΩ sont intercalées pour sécurité. En standard Arduino ce sont les broches **A4** et **A5** de l'ATmega328 qui sont affectées au dialogue sur le réseau des périphériques. Lorsque le démonstrateur démarre, il vérifie la présence du périphérique. Si ce dernier ne répond pas à l'appel, le double message d'erreur de la Fig.145 est envoyé en ACR assorti d'un BIP d'alerte sonore. Si le module inertiel est détecté, l'affichage sur le moniteur série de l'**IDE** étale une ligne de données correspondant à une première série de mesures effectuées automatiquement suivi de l'invitation à envoyer des commandes ">>>". (*Voir la Fig.146*)

Fig.145

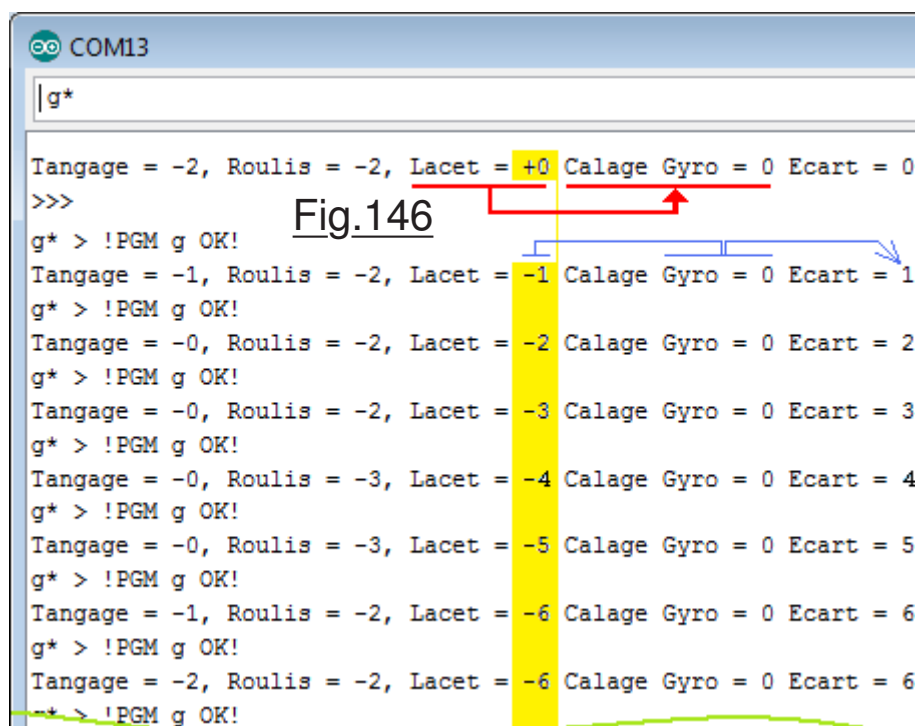
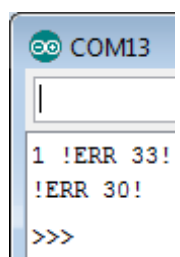
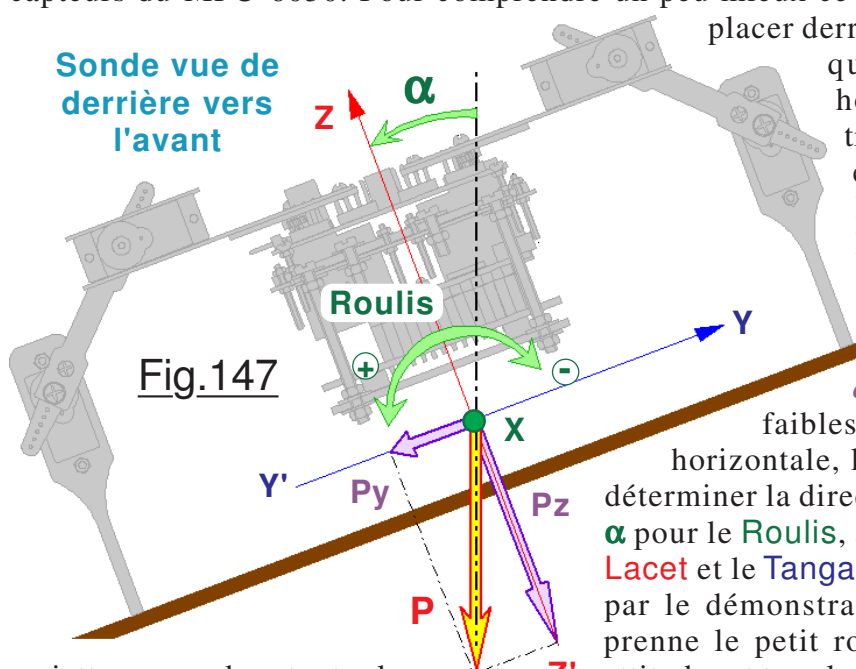


Fig.146

Lors de ce premier mesurage la référence en LACET est enregistrée, c'est à dire la valeur initiale qui est forcée à zéro dans l'unité de calcul intégrée. La donnée **Calage Gyro** contiendra la valeur mesurée du LACET chaque fois que l'on provoquera un **ReCalage du Gyroscope** virtuel. Toute commande "g\*" impose de réaliser un mesurage de la gravitation et de renvoyer les composantes angulaire du vecteur poids. Interpréter cette donnée n'est pas immédiat. Sur la [Fiche n°25](#) en Fig.2 sont montrés les axes classiques de repérage en référence au mobile qui emporte une telle centrale gyroscopique.

En particulier le repère cartésien **XYZ** est normalisé et les sens de rotation autour de ces directions trirectangles respectent la convention d'un repère dit de sens direct. **ATTENTION : Les angles d'inclinaison retournés par la centrale ne seront corrects que si le repère de référence est correctement orienté par rapport au mobile.** Il faudra vérifier que sur le module électronique que vous avez approvisionné, la minuscule "puce électronique" soit soudée avec la même orientation que celle sur le circuit imprimé du prototype. Si ce n'est pas le cas ROULIS et LACET voir TANGAGE seront permutés et les sens de rotation ne seront pas conformes à la théorie. Il vous suffira dans ce cas d'immobiliser votre module avec l'orientation correcte par rapport au châssis.

**R**ien ne permet à un accéléromètre de faire la différence entre un changement de vitesse du vaisseau dans lequel il se trouve, et le poids éventuel subit par tout corps quel qu'il soit situé dans un champ gravitationnel généré par l'astre à proximité duquel il se trouve. La pesanteur terrestre (Ou martienne dans l'hypothèse de la petite sonde.) est interprétée comme une accélération par les capteurs du MPU-6050. Pour comprendre un peu mieux ce qu'affiche le moniteur vidéo, on va se



assiette un peu dans toutes les

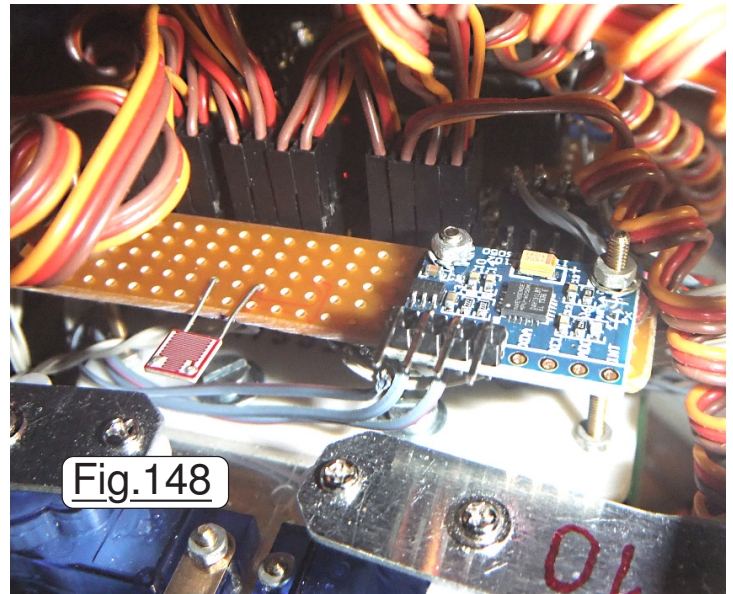
du vecteur poids **P** le calculateur interne au minuscule circuit intégré va prendre en compte les variations des accélérations angulaires et déterminer ainsi les angles balayés. C'est en brassant mathématiquement toutes ces informations que la petite merveille électronique nous retourne les valeurs déduites pour les trois angles qui caractérisent l'attitude actuelle de la sonde.

### ➤ Utilisation du capteur gravitationnel.

**C**ontrairement au "monstre de technologie" décrit sur la Fig.144 le MPU-6050 totalement statique ne comporte aucun élément mécanique. De ce fait tous les inconvénients issus du frottement sont radicalement éliminés. Ce n'est pas pour autant que cette technologie soit exempte de faiblesses. Observez à nouveau la Fig.146 qui représente une copie d'écran. Par la commande "g\*" répétée assez rapidement au moment du démarrage de l'IMU, on constate, mis en évidence par la couleur jaune, que la valeur calculée pour le **Lacet** dérive de façon "notable" pendant les quelques premières secondes. Plusieurs facteurs engendrent ce comportement. En particulier le substrat électronique monte en température ce qui influence la sensibilité des capteurs. De plus, si la sonde est pratiquement horizontale, les composantes **Px** et **Py** sont vraiment très faibles. Hors ce sont ces deux projections qui permettent de déterminer le **Lacet** à partir du vecteur **P**. Il est donc assez logique de constater ce type de divergence. En revanche, quand l'équilibre thermique est atteint, les rotations sont détectées avec une précision étonnante. (On ne s'en rend pas trop compte car le logiciel arrondi l'affichage et ne donne aucune décimale.) Quand on désirera maintenir la sonde sur un CAP précis, après avoir **Recalé le Gyroscope** de **Lacet** nous aurons à notre disposition un conservateur de CAP absolument fabuleux. Reste que, même si cette technologie était parfaite, de toute façon les valeurs fournies changeraient au cours du temps, sonde au repos sur son bouclier. N'oublions pas que la Terre tourne par rapport à l'Univers, et la centrale enregistre cette rotation.

P our faciliter les expérimentations, si vous allumez les phares avec la commande "a\*", vous armez une option de listage gyroscopique en continu. Dans ce cas, dès que vous allez frapper "g\*", l'écran vidéo va lister en permanence les trois angles d'attitude de la sonde. Orientez à votre guise, cabrez, piquez, tournez en CAP etc. Bref, manipulez librement et observez les valeurs affichées pour bien cerner ce que retourne le petit insecte robotisé. Vous pouvez à tout moment *Recaler le Gyroscope* avec la commande "f\*" qui active l'initialisation quand la LED moteurs OFF est allumée. Vous aurez rapidement compris qu'il faut réitérer "f\*" et éteindre la LED pour ne pas avoir à l'écran un écart calculé nul en permanence.

Le petit module électronique MPU-6050 est immobilisé par deux minuscules boulons  $\phi$  M2 sur le circuit imprimé sur lequel est soudée la cellule photorésistante. Il est supporté par des petites entretoises isolantes. L'implantation mécanique sera abordée très en détails dans le prochain tome, enfin consacré à l'intégration de tous les éléments sur le petit vaisseau interplanétaire. On voit bien sur la Fig.148 qui anticipe les explications pratiques, que l'on n'utilise que quatre broches pour le connecter électriquement au reste de l'électronique embarquée. Sur cette photographie la fiche HE14 femelle qui permet le raccordement n'est pas insérée à sa place car elle masque trop le petit module. Juste derrière on peut voir les nombreuses fiches qui sur le multiplexeur vont aux divers servomoteurs. Tout ce petit monde est bien tassé, peu de place est perdue. En termes de densité, avec JEKERT on approche la limite de ce que l'on peut faire tout en évitant de trop galérer. Affaire à suivre ...



A vec 9140 octets investis dans cette fonction, on note qu'elle se gloutonne à elle seule 30% de l'espace disponible pour le programme. Difficile d'imaginer moins, car les méthodes incluses dans la bibliothèque qui gère le petit module commercial sont très complexes et font appel à un outillage mathématique lourd. Au final, arriver à émuler une machine aussi complexe que celle de la Fig.144 relève assez d'un coup de baguette magique. Pour résumer, cette fonction n'aurait pas été possible sans l'optimisation permanente pour grappiller partout des "bouts de chandelle". À l'arrivée, pouvoir l'inclure dans les performances de JEKERT est sans exagération une récompense.

T erminez de vous faire plaisir en téléversant le programme de synthèse presque définitif **P30\_Programme\_COMPLET.ino** qui rassemble la totalité des fonctionnalités expérimentées sur les divers démonstrateurs. (*Presque définitif car une version visant à dialoguer avec une raquette de pilotage devra forcément le remplacer.*) Dans cette version, *des commandes sont ajoutées et listées en tête du programme*. Par exemple c'est "=" qui recale le gyroscope et "&" qui active ou suspend le listage continu des valeurs gravitationnelles. Bien que pour le moment la machine n'est qu'une structure avec des moteurs, que le calculateur est perdu sur une plaque à essais, complètement immergé dans des paquets de fils aussi envahissant que des ronces sur un champ oublié depuis des décennies, testez toutes les fonctions, toutes les commandes. Pour ménager la mécanique, la campagne d'essais ultime avant intégration se fait JEKERT posée sur le berceau.

N ous allons nous quitter provisoirement, car ce tome est pratiquement achevé. Vous attendez avec impatience le suivant qui va dévoiler tous les circuits imprimés et vous aider point par point à concrétiser votre propre robot. Tous les pièges rencontrés pour la réalisation du prototype seront abordés, et surtout certaines pratiques pour faciliter des manipulations qui seraient très indigestes si elles sont mal engagées. Bref, dans pas longtemps JEKERT va prendre du poil de la bête et devenir une exploratrice gaillarde et enthousiaste. Pour clore de troisième volet de la saga, et bien que pour le moment ce n'est pas encore possible, le chapitre qui suit décrit pas à pas la technique qui vous permettra d'établir la fiche de correction du compas de route. C'était promis. Comme je crois que le **TOME 4** sera très chargé, en vous proposant ce chapitre prématurément on allège d'autant le prochain livre. Je vous souhaite une très agréable lecture ...



### 34) 01/12/2017 : Calibrer le compas de route de JEKERT (MJD 58088)

Réaliser une fiche de correction d'un compas de route installé sur un mobile quelconque est une manipulation très ludique et particulièrement facile quand la machine est facile à orienter. La procédure qui suit va nous permettre d'établir la fiche de correction de la boussole statique installée sur JEKERT, les résultats seront consignés dans la [Fiche n°26](#) qui de surcroît est complétée par les protocoles de l'utilisation des expériences embarquées. Toute calibration d'un instrument scientifique quel qu'il soit se fait en plusieurs phases :

- 1) On rédige une procédure rigoureuse pour saisir les données de calibration,
- 2) On mesure avec précision les données de comportement sans idée préconçue,
- 3) Par l'analyse de ces données on dégage des conclusions d'ordre opérationnel,
- 4) On établit alors le document de compensation ou éventuellement on automatise une correction dans le programme qui gère la machine concernée.

#### Première phase : Saisie des données brutes.

Le programme **P30** doit être téléversé pour bénéficier de la possibilité de recalcr la centrale gyroscopique en LACET avec "=" et de faire défiler les valeurs gyroscopiques avec "&\*",

- Placer la sonde sur un berceau NON MÉTALLIQUE bien stable dans une zone bien dégagée, *(Et surtout dans un environnement proche non métallique.)*
- Avec "p02\*" imposer la configuration **Stable Transversal**, pour que la configuration "magnétique" soit celle qui sera effective lors de la mesure d'un CAP, *(La sonde étant en l'air posée sur un berceau stable garanti une assiette horizontale correcte indispensable pour mesurer.)*
- À l'aide d'une boussole "de conception sérieuse" repérer la direction du nord magnétique local, *(Surtout ne pas corriger avec la valeur de déclinaison, l'étalonnage envisagé est indépendant de la direction géographique car purement magnétique.)*
- Orientez soigneusement l'AVANT de l'axe longitudinal de JEKERT vers le Nord magnétique local,
- Caler le gyroscope avec "=" pour que l'Ecart soit nul,
- Répéter cette action tous les 10° pour couvrir les 360° d'un tour complet :
  - \* "&\*" pour faire défiler en continu les données gyroscopiques,
  - \* Changer manuellement l'orientation de JEKERT pour obtenir un Ecart de -10°, -20°, -30° etc,
  - \* "&\*" pour stopper le défilement.
  - \* "o\*" pour faire afficher l'orientation magnétique et noter cette valeur.

Rotation dans le  
sens horaire  
↑ ↑ ↑

Les **Ecart**s affichés étant calculés par soustraction entre le LACET actuel et celui mémorisé par la commande "=", le résultat sera perturbé lors du franchissement des "frontières" critiques 180° et 360°. Durant la saisie des valeurs mesurées on recalcr la boussole à la demande, action indiquée par (@) dans le tableau qui résume les données brutes notées lors de la manipulation. En (1) la sonde a été replacée orientée plein Nord pour la faire ensuite tourner à gauche.

Fig.149

Ecart	Boussole	Δ°	Ecart	Boussole	Δ°	Ecart	Boussole	Δ°	Ecart	Boussole	Δ°
0	332	-28	-100	140	+40	0 360	332	-28	100 260	252	-8
-10	346	-24	-110	151	+41	10 350	321	-29	110 250	246	-4
-20	12	-8	-120	161	+41	20 340	311	-29	120 240	240	0
-30	24	-6	-130	170	+40	30 330	302	-28	130 230	234	+4
-40	42	+2	-140	178	+38	40 320	293	-27	140 220	229	+9
-50	62	+12	-150	186	+36	50 310	285	-25	150 210	222	+12
-60	82	+22	-160	194	+34	60 300	277	-23	160 200	217	+17
-70	98	+28	-170	200	+30	70 290	270	-20	(@)	---	---
-80	114	+34	-180	207	+27	80 280	264	-16	-350 190	210	+20
-90	129	+39	---	(@) (1)	---	90 270	257	-13	-340 180	204	+24

La première colonne contient la valeur Gyroscopique. Comme pour le premier balayage angulaire on faisait tourner la sonde dans le sens horaire, les LACETS indiqués par **Ecart** sont négatifs. Arrivé à -180° on a réorienté la sonde plein nord en (1). Puis tournant cette fois dans le sens trigonométrique positif, *(Antihoraire.)* les LACETS indiqués sont alors positifs. Passant de +160 à +170 se produit un "incident de soustraction". Pour le contourner on recalcr le gyroscope qui de ce fait recalcule les **Ecart**s à partir de 360. Du coup en zone grisée -350 correspond à un **Ecart** angulaire de 170 et -340 à 180. Il reste maintenant à interpréter toutes ces données.

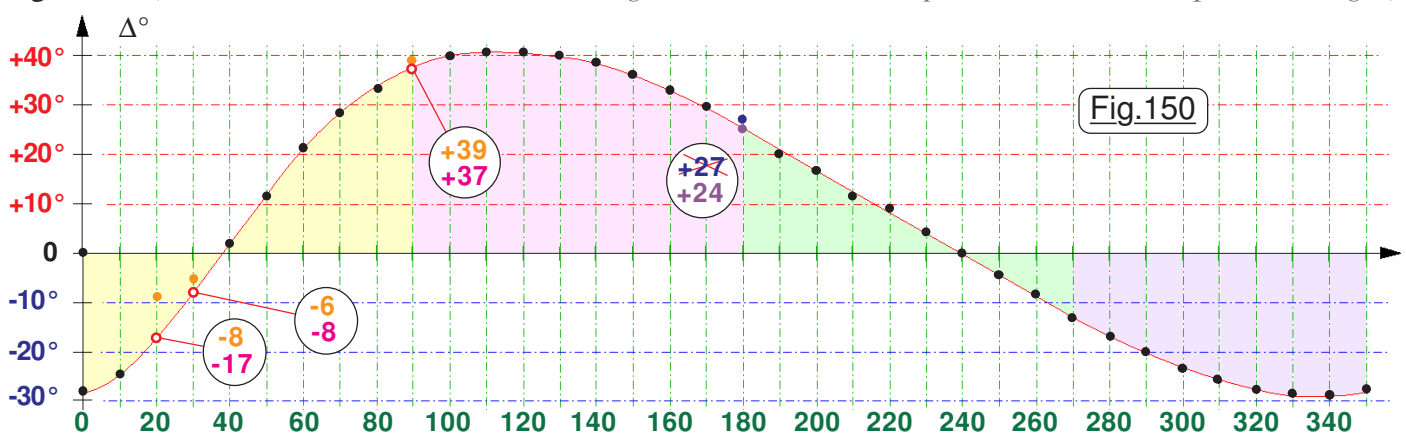
### Troisième phase : Analyse des données brutes.

**A** observer le tableau des données, on peut être étonné de telles divergences entre la direction du champ magnétique naturel et le CAP fourni par la boussole. Et pourtant la manipulation a été conduite avec rigueur. Une petite boussole de promeneur, un compas marine et un compas de relèvement professionnel ont permis de déterminer de façon fiable la direction du Nord magnétique dans le laboratoire, et à l'emplacement exact occupé par JEKERT au cours de la campagne de mesurage. Pourtant, quand la boussole est plein Nord, les deux mesures repérées par la couleur vert pastel donnent exactement  $332^\circ$  chacune. Quand JEKERT est orientée plein SUD dans un sens puis dans l'autre, on retrouve dans les deux cas surlignée en jaune  $204^\circ$  et  $207^\circ$ . Il y a bonne répétitivité entre les diverses mesures effectuées par le circuit HMC6352. À peine on tourne d'un degré qu'il accuse correctement la variation. Les résultats obtenus peuvent être considérés comme crédibles.

**L'** imprécision de trois unités pour  $207$  et  $204$  est parfaitement normale. Plusieurs phénomènes se combinent pour aboutir à de la différence entre réalité et affichage. La première source d'écart réside dans la précision assurée de la boussole qui n'est que de  $0,5^\circ$ . À cette "erreur" de mesure, s'ajoute l'arrondi inhérent à tout système d'affichage numérique qui fonctionne à  $\pm$  une unité. La centrale gyroscopique est très précise et l'écart angulaire affiché reste inférieur au degré. Toutefois, la valeur reste indéterminée à  $\pm$  une unité puisque de nature "numérique". Ces facteurs combinés aboutissent donc à une fluctuation potentielle disons d'environ  $\pm 2^\circ$ .

### Quatrième phase : Traitement des données.

**A** vant de présenter les données sous une forme facile à utiliser, il importe d'effectuer un filtrage car lors de saisie de nombreuses mesures physiques, on ne peut exclure totalement des points aberrants qu'il faudra pondérer. Puis, pour effacer en partie les imprécisions de mesurage, il importe généralement de "lisser" les résultats. Analyser une foule de valeurs numériques comme celles contenues dans le tableau de la Fig.149 n'est pas visuel. Aussi, souvent les techniciens ont recours à une représentation graphique. Par exemple le flot de valeurs sous forme de points figuratifs disposés sur un graphe agencé en coordonnées cartésiennes. Une méthode classique pour "lisser" les fluctuations résultant des mesures physiques : On fait passer une courbe de BÉSIER au plus proche des points figuratifs. (*Courbe de Bézier comme sur la Fig.MMM ou tout autre procédé mathématique de lissage.*)



On remarque immédiatement que ce qui ressemblait à du "n'importe quoi" devient une pseudo sinusoïde. Cette évolution est particulièrement rassurante, car dans la nature, dès qu'à un titre quelconque on approche un phénomène de rotation, la sinusoïde s'invite dans les modèles mathématiques ou les représentations graphiques.

Il est assez manifeste que les deux points mesurés pour  $20^\circ$  et pour  $30^\circ$  ne sont pas très crédibles. Aussi les valeurs oranges douteuses seront remplacées par les valeurs pondérées roses. Pour  $180^\circ$  on observe facilement que  $207^\circ$  est plus éloigné de la courbe de lissage que  $204^\circ$ . C'est donc  $+24$  qui sera pris en compte. Enfin, dans un tel traitement, le lissage consiste à utiliser la valeur de la courbe quand le point figuratif d'une mesure s'en éloigne manifestement trop. Par exemple pour  $90^\circ$  l'écart de  $+39^\circ$  sera remplacé par  $+37^\circ$  correspondant certainement plus à la réalité.

### Dernière phase : Présentation "rationnelle" des données.

Cette étape ultime est essentielle. Si les données sont fournies sans changement de présentation, elles seront difficiles à utiliser. Corriger le CAP indiqué par un compas de route doit

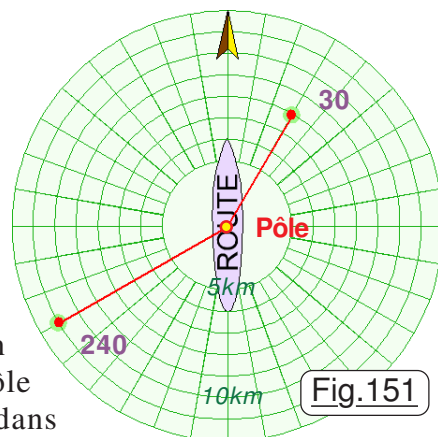
être immédiat. C'est la raison pour laquelle si un traitement informatique peut effectuer la correction pour fournir directement la bonne valeur, alors on aura contourné cette difficulté de manière idéale. Dans un petit monomoteur de tourisme, la présence d'un ordinateur caché derrière le tableau de bord est peu probable, tout au moins sur les aéronefs "économiques". Dans le cas de JEKERT, la "disponibilité informatique" reste à prouver car la place restante pour ajouter du code est réduite. Il semble assez probable que la séquence qui linéariserait les données du circuit HMC6352 ne serait pas élémentaire. Alors repoussons à plus tard une telle éventualité.

Aussi, et surtout pour aborder et traiter cet aspect d'utilisation d'un capteur "complexe", nous allons rédiger la [Fiche n°26](#) dans cet esprit.

### ➤ Diagramme polaire.

**S**ouvent il s'impose fortement quand on désire avoir une visualisation "naturelle" d'un phénomène qui embrasse les 360° d'un environnement quelconque. Ce peut être le diagramme de rayonnement d'une antenne, (*Comme celui partiel du transducteur ultrasons HC-SR04 par exemple.*) l'écran circulaire d'un radar hyperfréquences, le disque des couleurs d'outils graphiques informatiques et tout particulièrement la représentation des orientations géographiques cardinales. Fondamentalement, un Pôle est par nature un point singulier aux caractéristiques particulières dans

un domaine quelconque. Par exemple pour un tremblement de terre l'épicentre constitue un Pôle. Sur Terre l'axe de rotation de notre planète traverse le géoïde en deux points particulier qui sont le pôle Nord et le pôle Sud. Si le phénomène étudié présente une géométrie plane directionnelle, alors on use souvent de la représentation polaire. L'orientation d'un **rayon vecteur** symbolise la direction concernée, la longueur du **rayon vecteur** la grandeur du phénomène étudié dans cette direction. Considérons la Fig.151 qui représente l'écran radar de veille sur un navire marchand. Le pôle sur cette unité est constitué par le mât qui supporte l'antenne hyperfréquence du radar. Angulairement en partant du haut qui symbolise la proue on mesure le **gisement**. Les cercles partant du centre matérialisent les éloignements par rapport au pôle. Ici, dans le gisement 30 se trouve un écho à 8km. Dans le 240 à 11km on détecte une autre source de réflexion.



### ➤ Polaire de correction magnétique.

**G**énéralement les documents de correction du compas magnétique de bord sont fournis sous la forme d'un tableau comme celui de la Fig.152 que l'on trouvera sur la [Fiche n°26](#). Un tel rassemblement de valeurs est facile à utiliser soit en correction de lecture soit en prédiction. Par exemple si on obtient avec "o\*" la valeur 151° on en déduit immédiatement que le CAP actuel est de 110°. Réciproquement, si l'on désire orienter en Torsion le châssis au CAP 320° il faudra agir sur le potentiomètre pour obtenir 293° avec "o\*". Inspiré de la présentation d'un écran radar, il est avantageux de compléter ce tableau par un graphique en coordonnées polaires qui résumerait ces données. Une courbe

Réel	Indiqué	Réel	Indiqué	Réel	Indiqué	Réel	Indiqué
0	332	90	127	180	204	270	257
10	346	100	140	190	210	280	264
20	3	110	151	200	217	290	270
30	22	120	161	210	222	300	277
40	42	130	170	220	229	310	285
50	62	140	178	230	234	320	293
60	82	150	186	240	240	330	302
70	98	160	194	250	246	340	311
80	114	170	200	260	252	350	321

représentative des écarts permettrait d'obtenir les valeurs intermédiaires par interpolation. Par ailleurs, pour beaucoup d'entre nous, un petit croquis est plus parlant qu'un beau tableau de nombres bien alignés. Aussi, dans cet ordre d'idée le tableau

Fig.152

donné en Fig.152 sera complété par une telle représentation. J'imagine que vous avez toutes et tous bien compris que l'établissement de la correction du compas de bord ne peut se faire que lorsque la sonde est entièrement terminée. Néanmoins, votre exemplaire sera assez proche du mien, son comportement va certainement ressembler aux caractéristiques du mien.



Réel	Indiqué	Réel	Indiqué	Réel	Indiqué	Réel	Indiqué
0	-28	90	+37	180	+24	270	-13
10	-24	100	+40	190	+20	280	-16
20	-17	110	+41	200	+17	290	-20
30	-8	120	+41	210	+12	300	-23
40	+2	130	+40	220	+9	310	-25
50	+12	140	+38	230	+4	320	-27
60	+22	150	+36	240	0	330	-28
70	+28	160	+34	250	-4	340	-29
80	+34	170	+30	260	-8	350	-29

Fig.153

Pour construire le diagramme polaire de correction du compas magnétique, nous allons porter directement sous forme numérique sur les **rayons vecteurs**, la valeur du CAP réel en vert et celui retourné par la boussole HMC6352 en violet. (Sur la Fiche n°26) La longueur des **rayons vecteurs** va servir à définir l'**Ecart E** entre la réalité et la mesure perturbée. Pour tenir compte du signe l'idée de base consiste sur le **rayon vecteur** à définir un "éloignement central" qui sera considéré comme **Ecart nul**. Puis, par rapport à ce cercle de référence on en trace ensuite les autres, tous concentriques au pôle **P**. On peut alors sur un gisement quelconque  $\alpha$  placer le point figuratif **F** correspondant à la valeur de l'**Ecart**. Pour placer tous les points on a au préalable rédigé un tableau tel que celui de la Fig.153 résumant les données utiles. Enfin, on peut, comme montré sur la Fig.154, "lisser" une courbe de Bézier passant par tous les points **F** qui matérialise la polaire de correction magnétique du compas de bord. Sur la fiche dédiée le sens positif des **Ecart**s est dirigé vers

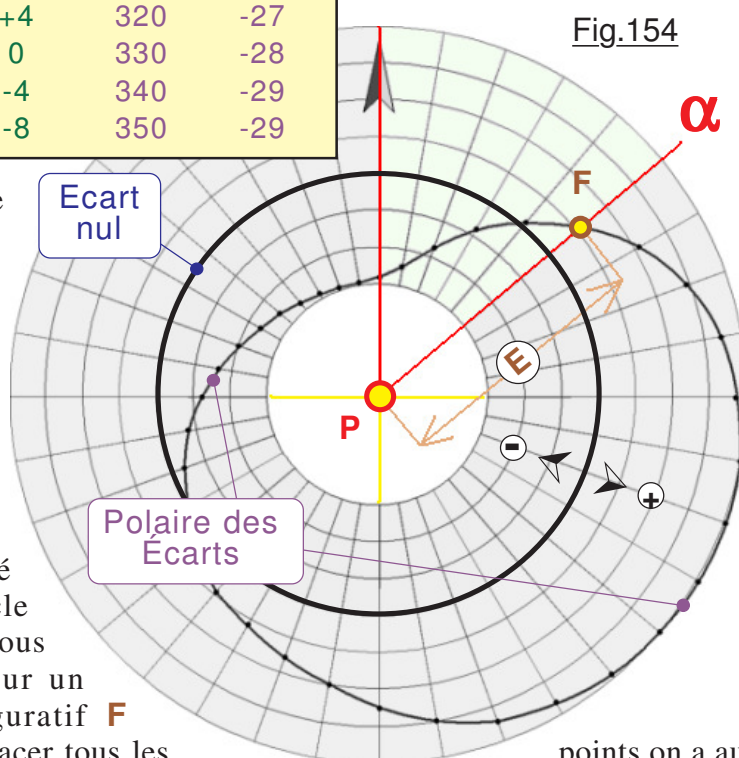


Fig.154

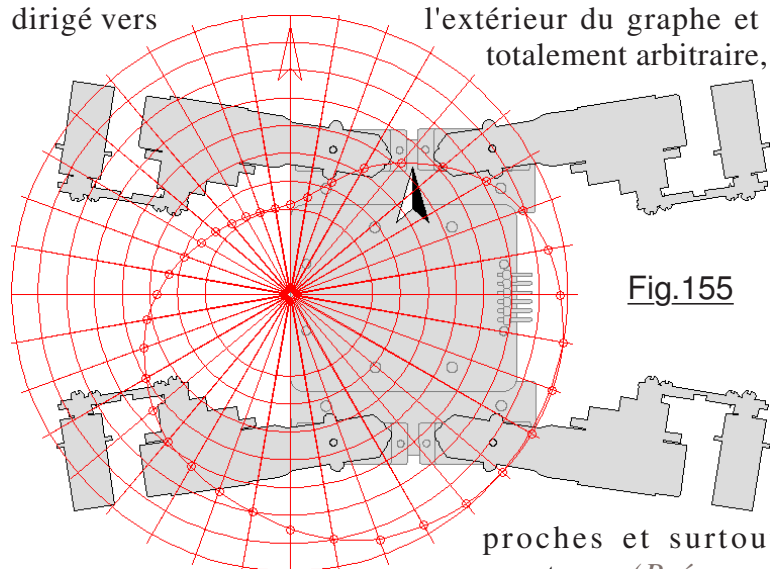


Fig.155

personnellement, quand j'ai constaté à quel point l'environnement immédiat de la boussole perturbait ses indications, j'ai été pour le moins très intrigué, imaginant presque un dysfonctionnement du capteur électronique. Je trouve très intéressant, comme représenté sur la Fig.155, d'établir une corrélation entre la polaire et la présence perturbatrice de la sonde. Le HMC6352 est sensible par conception à des champs magnétiques relativement faibles. Il ne faut pas s'étonner si les masses métalliques proches et surtout les champs parasites générés par les servomoteurs (*Présence d'aimants permanents.*) puissent engendrer de telles divergences dans la direction des "lignes de force" du champ magnétique terrestre.

**NOTE :** Le **gisement** correspond à un **repérage angulaire** d'une entité quelconque **par rapport à un mobile pris comme référence**. L'angle est compris entre 0° et 359°. Le zéro correspond à l'avant du mobile, donc la proue pour un navire. Quand ils devaient indiquer rapidement le gisement d'un ennemi qui attaquait les B17 durant le conflit de 39/45, les équipages indiquait la direction du chasser, donc son gisement, par des équivalents horaires. Par exemple à 3H pour tribord, 6H pour 180°, 9H pour 270° et 12H pour 360° ou plus exactement zéro.