SCHÉMAS ÉLECTRONIQUES.

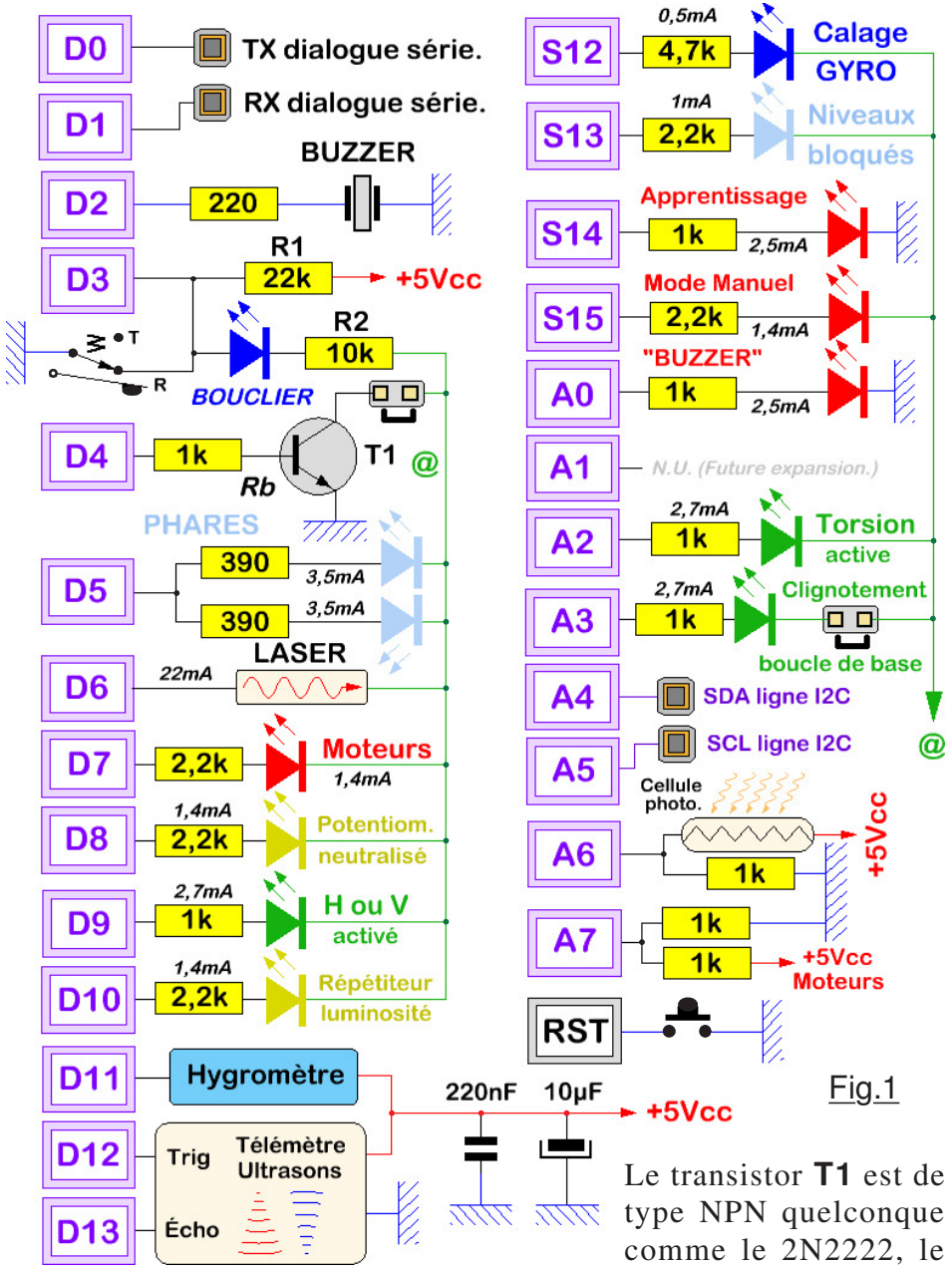


Fig.1

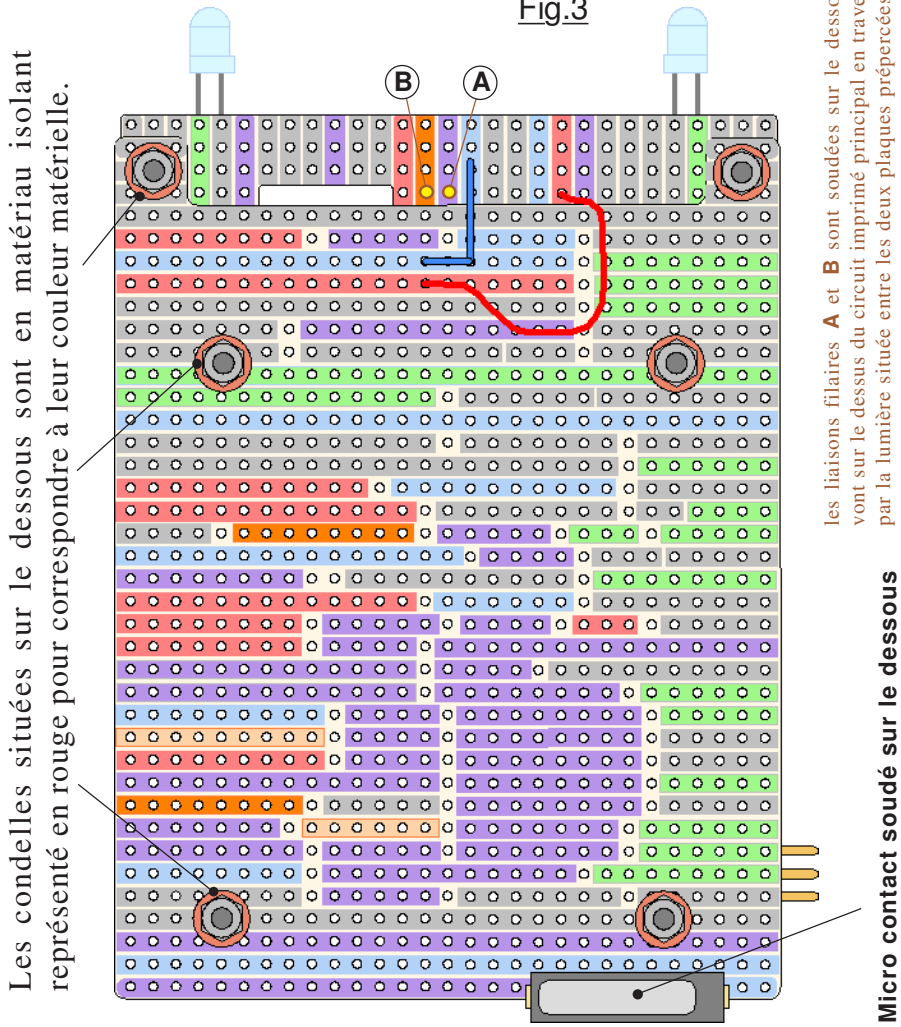
Le transistor **T1** est de type NPN quelconque comme le 2N2222, le 2N1711 etc. Il fonctionne en mode commutation. Sa résistance de base **Rb** permet de le saturer entièrement quand **D4** est à "1".

Fiche n°17

Circuit imprimé principal. (2/3)

Sans que ce ne soit systématique, d'une façon générale :
Le bleu représente des pistes à la masse GND,
Le rouge le +5Vcc, le orange clair le +3.3Vcc,
Le orange foncé le +9V issu d'un éventuel accumulateur déporté,
Le vert la masse coupée par le transistor NPN notée @.
Le violet des Entrées / Sorties sur la carte Arduino Nano,
Le gris des liaisons "banales".

Fig.3



Circuit entièrement câblé vu coté pistes cuivrées.

Capteur Température / Humidité DHT11

Montré sur la Fig.1, ce module de mesure est constitué d'un senseur de température à base d'une thermistance à coefficient négatif de température et d'un capteur d'humidité résistif. Un microcontrôleur intégré s'occupe de faire les mesures, de les convertir et de les transmettre. Le DHT11 fait partie de ces circuits numériques qui dialoguent au moyen d'une seule ligne série bidirectionnelle "à collecteur ouvert" de type OneWire. *(Un fil)*

Caractéristiques techniques du capteur DHT11 :

La mise en œuvre sur ARDUINO est d'autant plus facile que la bibliothèque DHT11.h est disponible en ligne. ***Elle se charge entièrement des protocoles de dialogue de type série.***

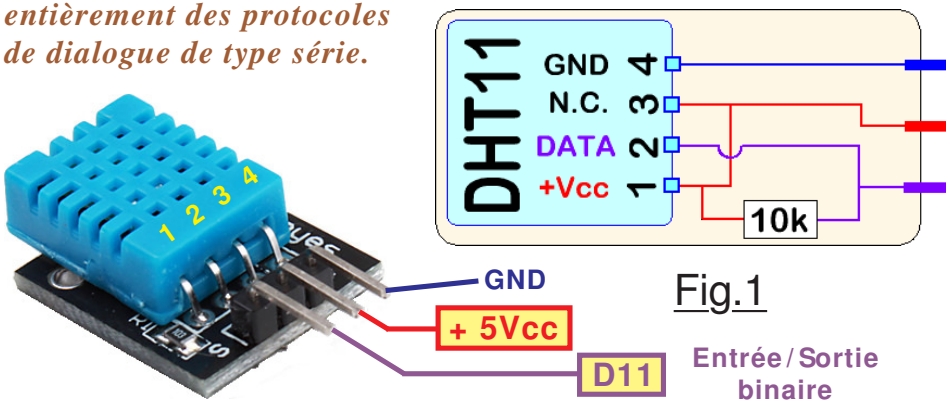


Fig.1

- Mesure de l'humidité relative. *(RH)*
- Alimentation comprise entre 3 Vcc et 5.5 Vcc.
- Courant d'alimentation : 100 µA maximum en veille, 2.5 mA maximum en dialogue.
- Mesures des températures comprises entre 0 °C et 50 °C.
- Mesures d'humidité entre 30 % et 90% RH à 0°C.
  entre 20 % et 90% RH à 25°C.
  entre 20 % et 80% RH à 50°C.
- Résolution 1 % RH. • Précision à 25 °C : ± 5% RH.

Lorsque le P.C. envoie un signal de démarrage, le DHT11 passe du mode faible consommation électrique au fonctionnement en mode dialogue et envoie l'accusé de réception suivi des données. *(Si le signal de départ est valide)* Il repasse ensuite en mode veille jusqu'à la prochaine sollicitation.

Fiche n°18

Câblage des Entrées / Sorties.

D0 : Libre pour F.E. *(Dialogue série entre deux Arduino.)*
D1 : Libre pour F.E. *(Dialogue série entre deux Arduino.)*
D2 : Pilotage du BUZZER.
≈ D3 : Détection du contact du bouclier avec le sol.
D4 : Coupure de masse de toutes les LED. *(Pilotage de la base du transistor de commutation.)*
≈ D5 : Éclairage analogique *(PWM)* des phares.
≈ D6 : Pilotage du LASER.
D7 : Pilotage LED témoin Moteurs figés.
D8 : Pilotage LED témoin Potentiomètre neutralisé.
≈ D9 : Pilotage LED témoin mode H ou V.
≈ D10 : Pilote la LED répétiteur du CAN potentiomètre.
≈ D11 : Capteur Humidistance et de température.
D12 : Transmetteur ultrasons.
D13 : Récepteur ultrasons.

A0 : Sortie de pilotage de la LED "BUZZER".
A1 : Libre pour F.E
A2 : Pilotage LED de torsion active.
A3 : Clignotement de la LED de "boucle de PGM active".
A4 : SDA pour la ligne I2C.
A5 : SCL pour la ligne I2C.
A6 : Mesure de la lumière pour le luxmètre.
A7 : Mesure de la tension d'alimentation des moteurs.

Sortie S12 Multiplexeur : LED de la commande '=*' armée.
(Mémorisation du Gyroscope sur le Lacet actuel.)
Sortie S13 Multiplexeur : LED de l'option 'B*' active.
(Niveau éclairages des phares et du LASER Bloqués.)
Sortie S14 Multiplexeur : LED du mode 'D*' actif. .....
(Enregistrement d'une suite de déplacements en cours.)
Sortie S15 Multiplexeur : LED du mode "P9n*" en cours.
(Pilotage des moteurs manuellement et individuellement.)

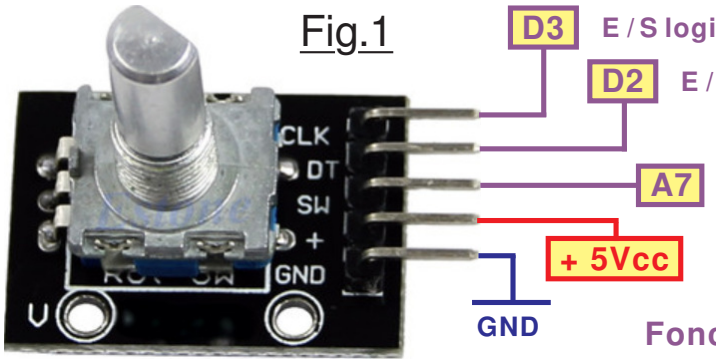
Encodeur rotatif KY-040.

Le KY-40 est un codeur **sans butée** à 20 points par tour pourvu d'un "bouton poussoir" par appui sur la tige de commande centrale. La sortie se fait par deux lignes pilotées par des capteurs de type codage Gray. La Fig.1 donne le schéma des branchements à réaliser sur ARDUINO pour utiliser le programme "de la Raquette" P40.

Caractéristiques techniques du module KY-040 :

- Consommation maximale : 10 mA sous 5 Vcc.
- Température de fonctionnement : - 30 à + 70 °C.
- Température de stockage : - 40 à + 85 °C.
- Durée de vie du capteur de rotation : Minimum 30 000 cycles.
- Durée de vie du contact central : Minimum 20 000 cycles.
- Résistance de passage du contact de RAZ : 3 Ω maximum.

Fig.1



L'entrée **A7** est choisie, car elle ne peut fonctionner qu'en entrée.

Fonctionnement :

Concrètement les trois sorties sont alimentées au + 5 Vcc par des résistances de 10 kΩ. La sortie **SW** est celle de l'inverseur piloté par appui sur la tige du rotor. Les deux sorties **CLK** et **DT** sont en réalité deux sorties classiques avec déphasage de 90° souvent nommées **A** et **B** pour ce type de capteur. Le déphasage de 90° électriques des signaux **CLK** et **DT** permet de déterminer le sens de rotation. *(Voir Fig.2)* ***Le capteur est traité par interruptions.***

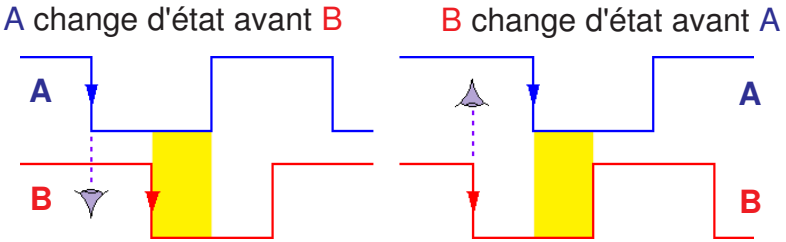


Fig.2

Fiche n° 19

Petit module LASER.

Caractéristiques techniques du petit module LASER :

- C'est une simple diode LASER associée à une résistance de 103 Ω montée en série pour limiter le courant.
- Longueur d'onde 650 nm. *(Couleur rubis)*
- Puissance lumineuse 2 à 5 mW.
- Le +5Vcc peut être appliqué directement sur la broche d'alimentation, mais pour augmenter la durée de vie de la diode LASER il est possible d'insérer une résistance. Le tableau proposé ci-dessous précise le courant d'alimentation en fonction de **R**, la tension d'alimentation étant de +5Vcc.
- Le courant maximal consommé est compatible avec la sortance d'une broche binaire de l'ATmega328. Il est parfaitement possible de moduler la luminosité par PWM, la fréquence de 490Hz de découpage étant largement assez élevé pour donner l'impression d'une clarté constante.

R	0	22 Ω	47 Ω	82 Ω	100 Ω
I	22 mA	19 mA	17 mA	14 mA	14 mA

Fig.1

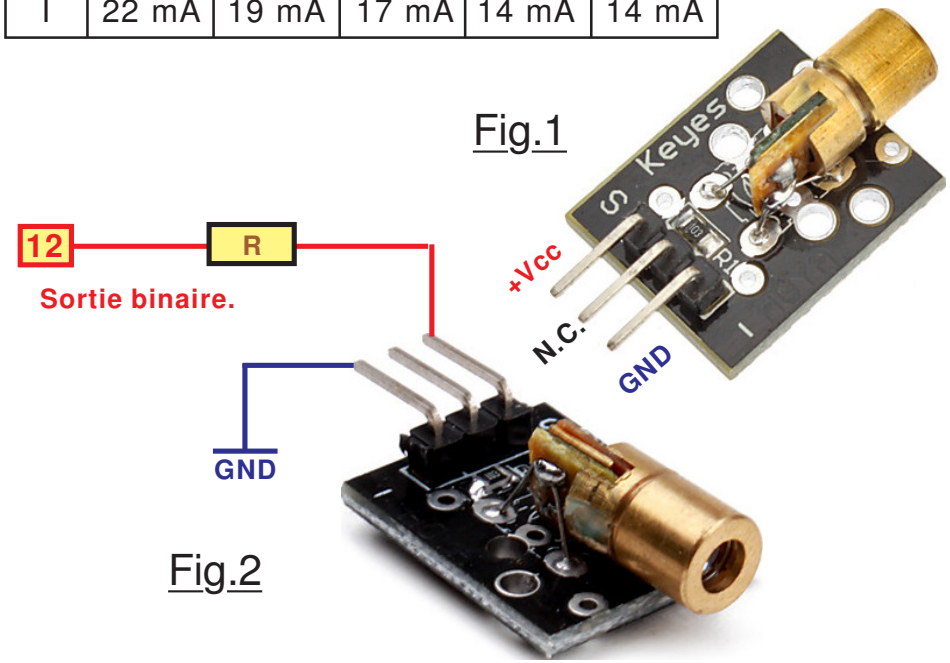


Fig.2

Module ultrasons HC-SRO4.

Caractéristiques techniques du module HC-SR04 :

- Alimentation : 5Vcc.
- Consommation en utilisation : 15 mA.
- Gamme de distance : 2 cm à 4 m. *(Limité ici 2,5m par le byte.)*
- Résolution : 3 mm.
- Angle de mesure : < 15°.

Mise en œuvre :

La Fig.3 montre les branchements à effectuer. Il faut envoyer une impulsion état logique "1" d'au moins 10 µs sur la broche Trig Input pour déclencher une mesure. *(Voir Fig.1)* En retour la sortie Output ou *(Echo)*, va restituer une impulsion d'environ + 5v dont la durée est proportionnelle à la distance si le module détecte un objet. Afin de pouvoir calculer la distance exprimée en cm, on utilisera la formule suivante :

Distance en cm = (Durée de l'impulsion Echo en µs) / 58

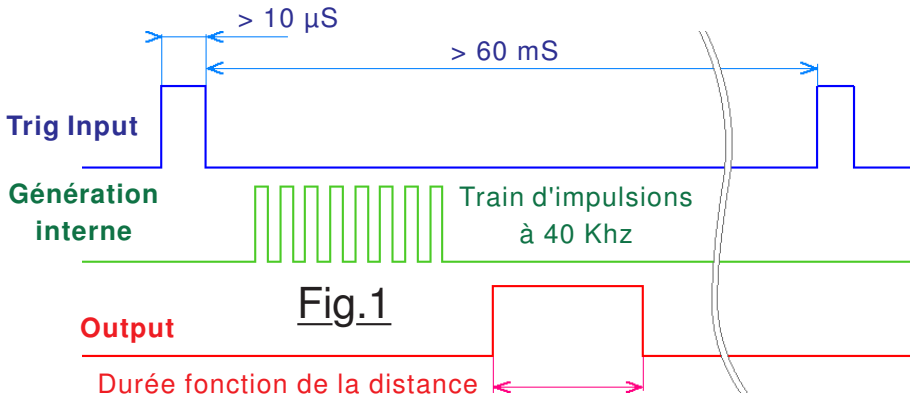
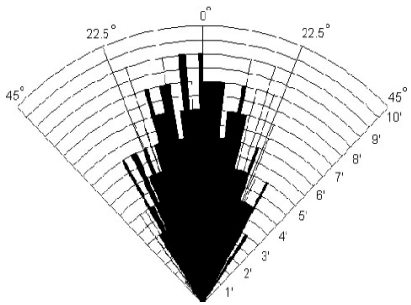


Fig.1



Les distances sont en pouces

Diagramme polaire

Fig.2

Fiche n° 20

Exemple de programme :

```
#define TX_Pulse_sonard 12
#define Echo 13
byte Distance;
int  Lecture_Echo;

void setup() {
pinMode(Trig, OUTPUT); digitalWrite(Trig, LOW);
pinMode(Echo, INPUT); }

void Telemetre_ultrasons() {
  // Déclencher une mesure :
  digitalWrite(TX_Pulse_sonard, HIGH);
  delayMicroseconds(10);
  digitalWrite(TX_Pulse_sonard, LOW);
  // Analyser le retour sur la ligne de réception :
 Lecture_Echo = pulseIn(Echo, HIGH);
 Distance = Lecture_Echo / 58;}
```

La procédure Telemetre_ultrasons effectue une mesure et préserve le résultat dans la variable Distance.

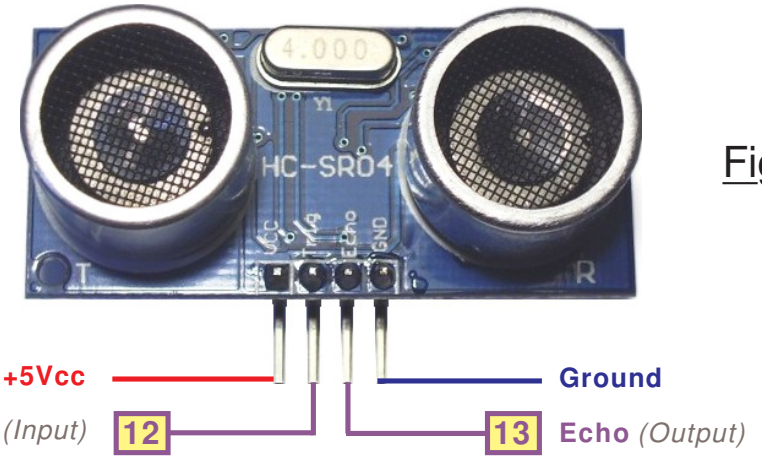
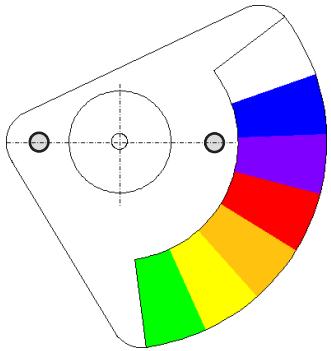
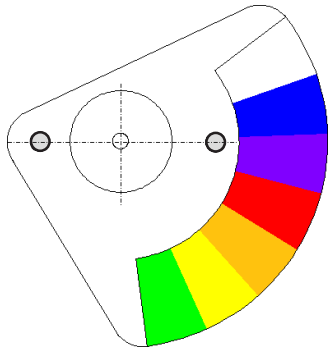
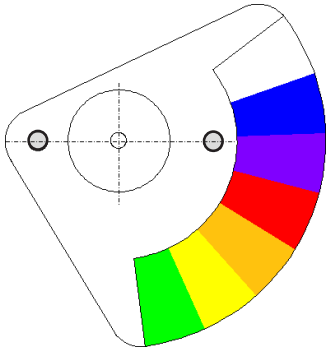
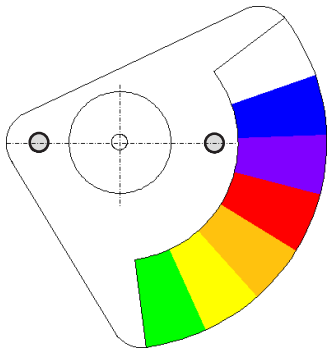
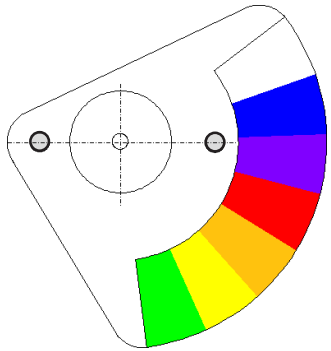
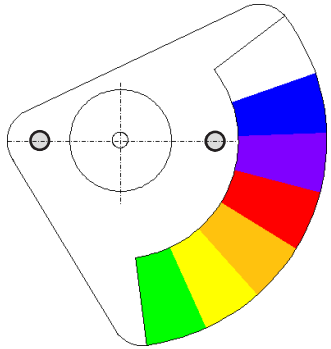


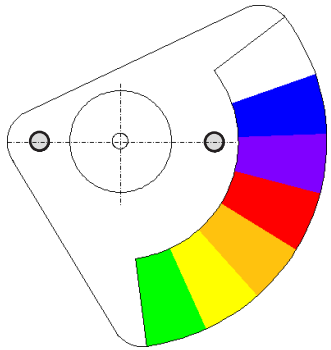
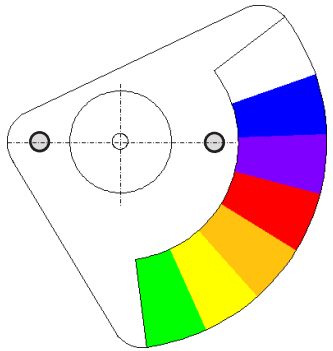
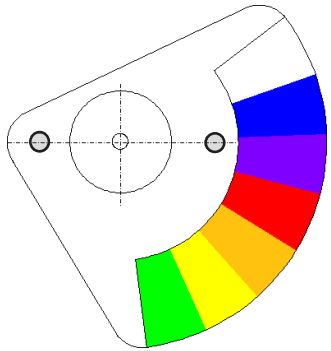
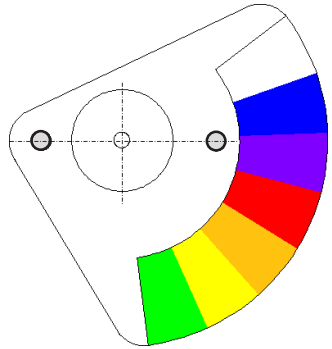
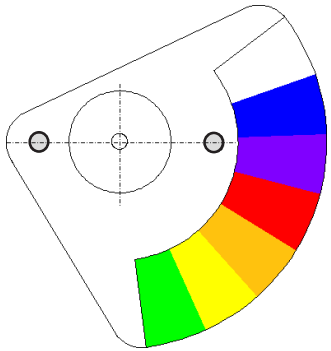
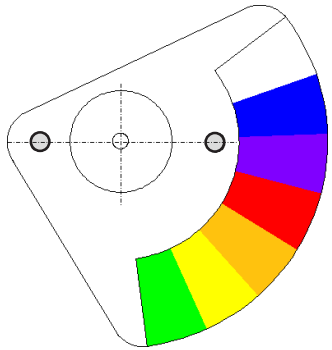
Fig.3

Comme le montre la Fig.2 le capteur à ultrasons HC-SR04 présente un angle d'ouverture trop important *(Cône d'environ 45 degrés)* qui interdit une analyse fine et ne permet pas réellement de différencier la forme des objets situés dans le champ du balayage.

Filtres colorimétriques (1/2)

Filtres colorimétriques (2/2)

BUS série au standard I2C. (1/2)

Développé initialement par Philips en 1982, le bus I2C s'est largement imposé dans le domaine des microprocesseur, microcontrôleurs et applications industrielles diverses. Sa désignation dérive de Inter-Integrated Circuit. Il fut à l'origine conçu pour des applications de domotique et d'électronique domestique.
La norme I2C est basée sur un ***bus série synchrone bidirectionnel*** fonctionnant en "half-duplex", où plusieurs périphériques maîtres ou esclaves peuvent communiquer entre eux. Les dialogues ont toujours lieu entre un seul maître et un ou tous les esclavesprésents et l'échange de données et toujours déclenché à l'initiative du maître. *(Jamais de maître à maître ou d'esclave à esclave.)*

➢ Constitution matérielle du bus I2C.

Outre une masse commune GND la ligne n'utilise que deux fils :
- **SDA** *(**S**erial **Da**ta Line)* : Ligne de données bidirectionnelle,
- **SCL** *(**S**erial **Cl**ock Line)* : Ligne d'horloge pour la synchronisation également de type bidirectionnel.

Les deux lignes sont maintenues à l'état logique "1" par un niveau de tension +VDD à travers des résistances de forçage. *(Pull-Up.)*
Dans le standard I2C le nombre maximal de périphériques est limité à 128 par le nombre d'adresses disponibles, 7 Bits d'adressage et un Bit R/W. *(Lecture ou Écriture.)* Bien qu'une foule de combinaisons complexes soit possible, dans le cas d'Arduino c'est la carte ATmega328 qui sera toujours le Maître et les modules périphériques les esclaves. Traditionnellement si le programme d'application doit intégrer une ligne I2C, comme représenté sur la Fig.1 ci-dessous ce ***sont* A4 *et* A5** qui ***sont respectivement affectées à* SDA *et* SCL**. Les bibliothèques qui accompagnent les modules du commerce sont basées sur ce schéma.

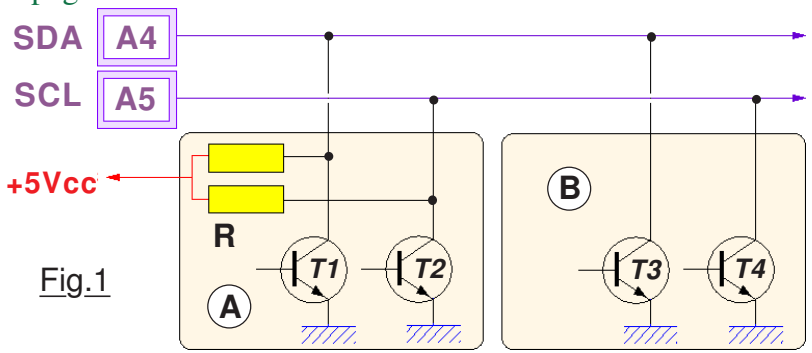


Fig.1

... / ...

BUS série au standard I2C. (2/2)

➢ Signaux échangés sur un bus I2C.

Le niveau HIGH ou LOW de la ligne **SDA** doit être maintenu stable pendant le niveau HIGH sur la ligne **SCL** servant à déclencher la lecture successive des bits du protocole de dialogue. *(Voir la Fig.2)*

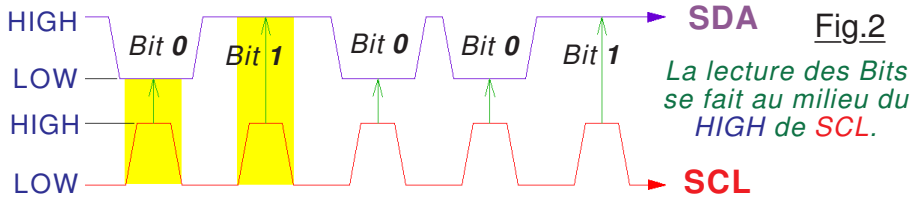


Fig.2

La lecture des Bits se fait au milieu du HIGH de SCL.

Comme montré sur la Fig.1 les équipements sont connectés au bus par des électroniques de type drain ouvert. *(Ou collecteur ouvert.)*
Fonctionnant en ET câblés deux périphériques tels que **A** et **B** peuvent "parler" simultanément. Dans ce cas un état logique "0" "écrase" un état logique "1". Pour caractériser ce genre d'incident potentiel sur un bus I2C on utilise le vocable :
- L'état logique "0" LOW est un état dominant,
- L'état logique "1" HIGH est un état récessif.

Lorsque le bus n'est pas utilisé, il est forcé niveau haut par les résistances telles que **R** de l'un des modules connectés. Il suffit d'un seul rappel à +5Vcc pour que la ligne fonctionne. Si plusieurs modules sont pourvus de résistances de forçage, le courant qui devra être drainé par l'ATmega328 et les électroniques branchées sera plus important mais reste généralement faible car le nombre de périphériques est classiquement faible pour des applications ordinaires.
Pour les modules dédiés à Arduino les vitesses de transmission sont généralement comprises entre 100kb/s et 400kb/s.

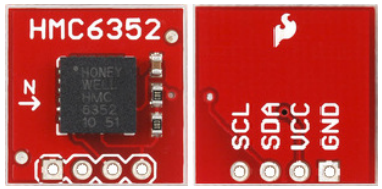
➢ Les bibliothèques de programme.

Pour simplifier le travail des programmeurs et se positionner sur le marché, les concepteurs de modules électroniques dialoguant en I2C fournissent des bibliothèques dédiées à leurs produits. Outre les "library" propres à chaque référence commerciale, la bibliothèque Wire.h est spécialisée pour gérer sur carte Arduino les protocoles I2C/TWI et devra parfois accompagner celle qui accompagne un module électronique spécialisé. C'est elle qui impose l'usage de **A4** et **A5**.

... / ...

BOUSOLE statique HMC6352. *(1/2)*

Comme tous les modules qui utilisent une SPI pour réaliser le dialogue avec Arduino, ce petit circuit est extrêmement facile à mettre en œuvre. Le schéma du petit module est donné en Fig.1 avec le circuit imprimé représenté par dessus. La flèche verte sur le dessin représente la direction du "Nord magnétique" de cette boussole statique. La petite flèche notée **N** sur le circuit imprimé définit ce pôle du circuit intégré. Quand cette flèche est orientée vers le Nord magnétique local, le HMC6352 retourne la grandeur 0° pour la valeur du Cap.



Caractéristiques techniques du module HMC6352 :

- Tension de fonctionnement : 2,7Vcc à 5,2Vcc.
- Consommation 2mA. Maximum 10mA sous 5Vcc.
- Compas avec donnée de sortie en valeur décimale pour le CAP.
- Intègre deux axes magnétiques complets.
- Logiciel local de traitement d'acquisition, de calibrage et de calcul de CAP inclus au circuit intégré.
- Interface série de type I²C. ***(Adresses : 42HEX et 43 HEX)***
- Fonctionne en esclave. Fréquence d'horloge possible 100kHz.
- Compensé en température.
- Résolution pour le CAP calculé 0,5° avec une répétabilité de 1°.
- Peut être utilisé dans un environnement de champs magnétiques importants.

ATTENTION : Bien que réputé protégé contre les champs magnétiques importants, sa mise en présence d'un aimant de gestion de la tête d'un disque dur, (Aimant très puissants) a engendré un mauvais fonctionnement du circuit. Pour retrouver sa sensibilité il a fallu soumettre la "puce" à un champ inverse, puis attendre plusieurs heures que la "rémanence" magnétique se dissipe.

- Plage de champ magnétique ±2Oe.

L'oersted (Symbole Oe) est l'unité ÉLECTROMAGNÉTIQUE CGS à trois dimensions d'excitation magnétique ou de champ magnétique. L'oersted, nommé ainsi en l'honneur de Hans Christian Ørsted.

... / ...

(2/2)

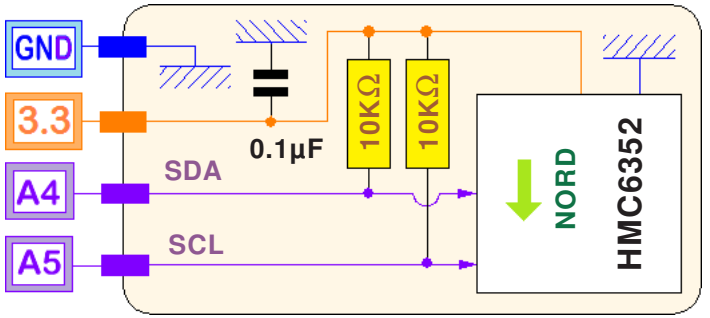


Fig.1

Outre le condensateur de découplage sur l'alimentation en 3,3Vcc on peut noter sur la Fig.1 que les deux lignes de la SPI sont forcées à l'état logique "1" par des résistances de 10 kΩ incluses. C'est un impératif s'il n'y a pas d'autre module I²C en ligne, car ce type de périphérique est à "collecteur ouvert" pour permettre le multiplexage de plusieurs esclaves. En standard pour du dialogue I2C sur Arduino on utilise les deux broches analogiques **A4** pour des données **SDA** et **A5** pour le signal de synchronisation **SCL**.

ATTENTION : Le module boussole HMC6352 ne fournit une information de ***Cap magnétique correcte que s'il se trouve dans une attitude parfaitement horizontale***. Il importe donc de veiller à cette contrainte lors des expérimentations et pour son assemblage sur la structure de l'application. Conformément à tout dispositif influencé par un champ magnétique, il donnera une information relative aux "lignes de champ" locales. ***Toute masse magnétique ou aimant dans son environnement faussera l'information relative au champs magnétique terrestre***.

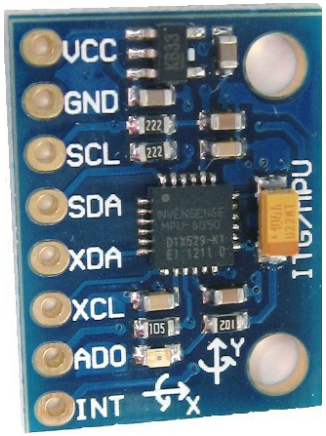
Comme pour tout compas magnétique, l'utilisation d'un tel dispositif doit s'accompagner d'un graphe donnant les ***courbes de correction*** angulaire pour tenir compte des perturbations engendrées par la structure du mobile sur lequel est employé ce module.

Initialement, pour pouvoir tester ce petit module la bibliothèque spécialisée hmc6352.h avait été intégrée dans l'environnement de développement. Mais le actuel d'utilisation sur la sonde n'en a pas besoin, il faut juste inclure les routines de gestion de la ligne I²C avec la déclaration :

```
#include <Wire.h>
```

... / ...

Centrale gyroscopique 3 axes MPU-6050. (1/4)



Nombreuses sont les applications qui doivent prendre en compte l'orientation d'un dispositif quelconque tel qu'une manette de jeu, un téléphone portable, un drone, un télescope, un caméscope gyro-stabilisé etc. Compte tenu de la complexité engendrée par la mesure des phénomènes d'accélérations tant linéaires qu'angulaires, un marché considérable à incité l'industrie à produire des composants de plus en plus sophistiqués, associés à des bibliothèques de programme pour en faciliter l'intégration dans les projets en cours de développement. Le MPU-6050 en est un exemple, qui comme bien d'autres circuits intégrés, à poussé les fournisseurs à mettre sur le marché des petits circuits imprimés tel que celui de la photographie donnée ci-dessus associant au composant de base quelques éléments électroniques pour en faire un dispositif prêt à l'emploi. Le circuit intégré basé sur un MPU-6050 combine un accéléromètre 3 axes et un gyroscope 3 axes avec un circuit DMP™. *(Digital Motion Processor™)* Le circuit de traitement DMP™ permet de résoudre les problèmes d'alignement sans rencontrer les erreurs générées par les composants discrets.

Caractéristiques de base du circuit MPU-6050 :

- **Détecteur de mouvements intégrant 3 axes gyroscopiques et 3 axes accélérométriques linéaires et angulaires.**
- Alimentation DC : 2,3 à 3,4 Vcc.
- Consommation maximale : 3,9 mA avec les 6 capteurs activés.

Accéléromètres :

- Plages de mesure : ±2 g / ±4 g / ±8 g / ±16 g.
- Tolérance de calibration : ±3%.

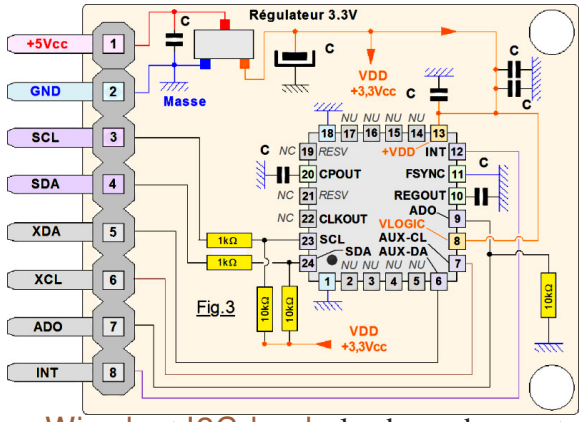
Gyroscopes :

- Plages de mesure: ±250/500/1000/2000 °/s
- Tolérance de calibration : ±3%
- Interface de dialogue I2C fonctionnant jusqu'à 400kHz.
- Capteur de température intégré.
- Température de service : -40°C à +85°C

Fiche n° 24

Centrale gyroscopique 3 axes MPU-6050. (2/4)

Le schéma électronique du petit circuit imprimé est donné ci-cntre. On peut y voir également le brochage du MPU-6050. Dans la version la plus simple de son utilisation, les quatre broches 1 à 4 sont suffisantes pour sa mise en œuvre. Si on développe le programme avec usage des bibliothèques Wire.h et I2Cdev.h, les branchements imposés sont ceux de la Fig.1 donnée ci-dessous.



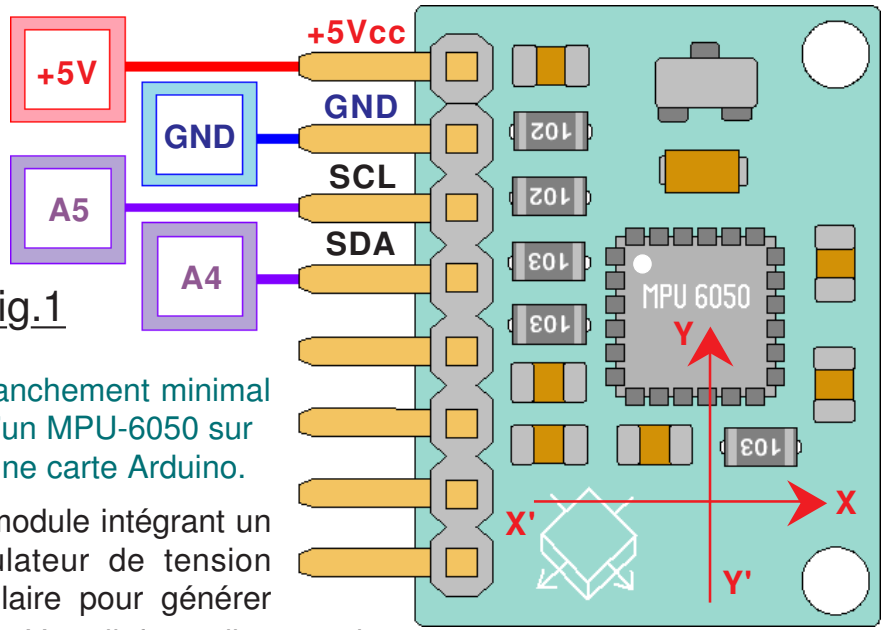


Fig.1

Branchement minimal d'un MPU-6050 sur une carte Arduino.

Le module intégrant un régulateur de tension tripolaire pour générer le 3.3Vcc, il faut alimenter le module en +5Vcc.

ATTENTION : L 'orientation des axes internes est celle montrée en rouge sur la Fig.1 et dépend de celle de l'implantation du 6050 sur le circuit imprimé. *(Ces axes sont fonction des développeurs.)*

Centrale gyroscopique 3 axes MPU-6050. (3/4)

Principe de fonctionnement :

Le gyroscope / accéléromètre MPU-6050 comporte 6 axes avec une conversion analogique-numérique codée sur 16 bits simultanée sur chaque canal, et inclus une interface de dialogue I2C.
La lecture des mesures brutes de ce capteur est aisée, car il dispose d'un registre FIFO d'une taille de 1024 octets que le microcontrôleur d'Arduino peut lire par requête d'un signal d'interruption.
Le module fonctionne en esclave sur le bus I2C par rapport à Arduino *(Broches* ***SDA*** *et* ***SLC****.)* mais peut aussi contrôler un autre dispositif en aval avec les broches AUX-DA et AUX-CL. Le capteur avec son DMP est apte à réaliser des calculs rapides directement sur la puce à partir des mesures brutes du capteur, mais cette fonction est mal documentée, il est alors plus simple de traiter les mesures brutes avec une carte Arduino.

Filtrage des mesures brutes :

Un filtrage des mesures brutes s'impose si on ne veut pas cumuler des petites erreurs au cours du temps et laisser dériver le calcul de la position réelle. L'accéléromètre doit être filtré si on veut piloter un asservissement de manière fluide et sans oscillations permanentes avec surcompensations. On peut utiliser une simple moyenne glissante de 40 mesures de 12 ms chacune par exemple pour lisser les micro fluctuations ou utiliser un filtre de Kalman cette méthode étant nettement plus complexe.

Nature des données habituellement utilisées :

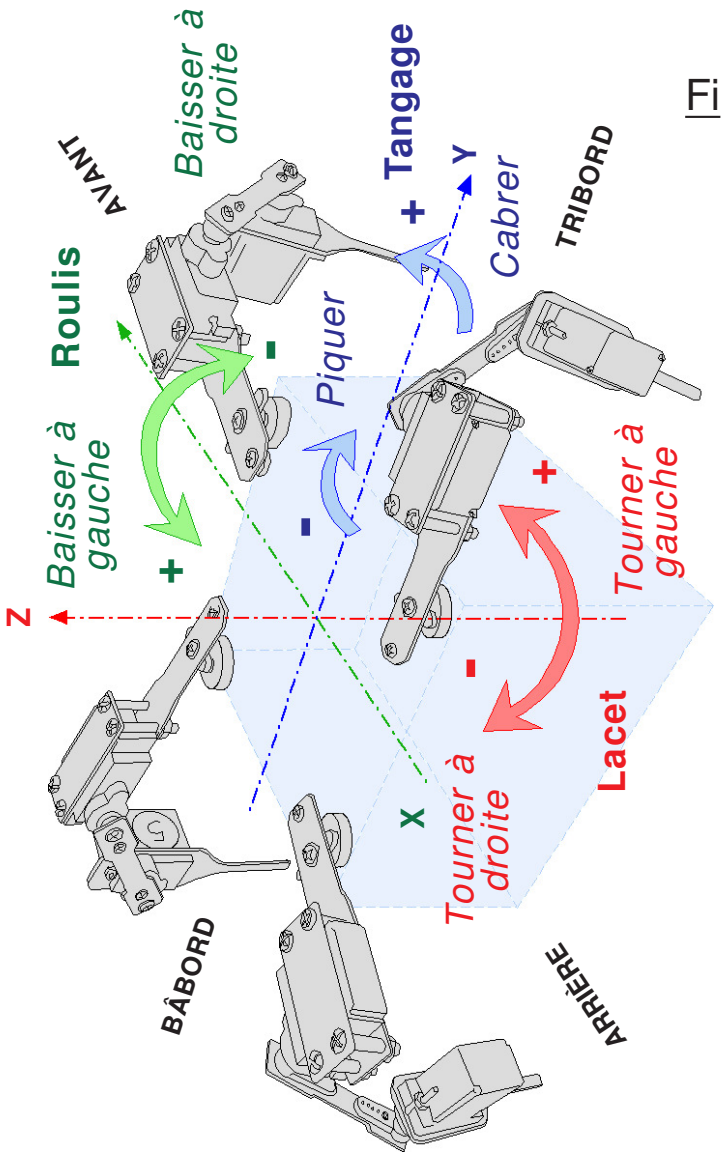
- **Les quaternions** *(w, x, y, z)* **:** Trois quaternions permettent de définir l'orientation d'un axe, et le quatrième précise la valeur de la rotation autour de cet axe. *(Cette méthode contourne les limitations au delà de 180°.)*
- **Les angles d'Euler** *(Psi, téta, phi)* **:** Les angles d'Euler sont constitués par trois rotations autour des axes X, Y et Z.
- **Les angles en référence d'un mobile :** *(Voir la Fig.2)*
  (C'est la forme d'informations généralement la plus commode.)
  YAW : LACET. *(> 0 si la rotation est vers la droite.)*
  PITCH : TANGAGE. *(> 0 si la rotation est à piquer.)*
  ROLL : ROULIS. *(> 0 si l'inclinaison est vers la gauche.)*

Centrale gyroscopique 3 axes MPU-6050. (4/4)

Les sens de rotation correspondent à ceux d'un système trirectangle de sens direct. *(Sens triginométrique positif.)*

Nom des axes gyroscopiques et sens de variation :

Fig.2



Les angles en référence d'un mobile :
YAW : LACET. *(> 0 si la rotation est vers la gauche.)*
PITCH : TANGAGE. *(> 0 si la rotation est à cabrer.)*
ROLL : ROULIS. *(> 0 si l'inclinaison est vers la gauche.)*

Protocoles des expériences embarquées.

Enregistrement d'un spectre colorimétrique :

- Avec "p02*" imposer la configuration ***Stable Transversal***,
- Éteindre les phares avec "a*" et couper le LASER avec "a*",
- Vérifier que les moteurs ne sont pas figés avec "f*",
- Déclencher une numérisation avec "(*". *(Il y a "écrasement" des quatorze derniers échantillons d'un panoramique télémétrique.)*

La visualisation sur le moniteur de l'**IDE** se fait avec la commande ")*". Tout enregistrement d'un spectre télémétrique "écrasera" un éventuel spectre chromatique actuellement mémorisé.

Utilisation du capteur gravitationnel :

- Poser la sonde sur le bouclier avec "p01*"
- Déclencher une mesure avec "g*",
- Si une répétition continue est désirée : Consigne "&*",
- Réitérer la commande "&*" pour sortir du mode et stopper l'affichage continu.

Utiliser la centrale inertielle :

- "g*" liste sur la ligne USB les valeurs gyroscopiques.
- "=*" Arme la mémorisation du LACET actuel. Toute activation du listage que ce soit par "g*" ou durant un listage continu libère la consigne.
- "&*" déclenche un listage en continu des valeurs de la centrale inertielle. C'est une bascule de type OUI/NON. Durant le listage "=*" provoque l'enregistrement du LACET actuel. *(Ecart représente la différence entre le LACET actuel et la valeur du CAP enregistrés avec "=*".)*
- "t*" active la Torsion qui permet d'orienter finement la centrale gyroscopique en LACET.

Utiliser de la boussole magnétique :

- "o*" affiche sur la ligne USB l'orientation magnétique.

Deux capteurs à effet Hall mesurent la valeur du champ magnétique dans deux directions cartésiennes. Les deux valeurs sont combinées pour calculer l'orientation du vecteur résultant en gisement. Exprimé en degrés angulaires le CAP magnétique indiqué ne sera crédible que si le châssis de la sonde est parfaitement horizontal. (Voir la fiche de correction du compas magnétique coté verso.)

... / ...

Fiche de correction du compas magnétique.

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