

ADAPTATION D'INSTRUMENTS REELS POUR UTILISATION SUR SIMULATEUR DE VOL

Par YVECA _Oct 2013_ Rev00_



SOMMAIRE

1/ GENERALITES

2/ OUTILLAGE

3/ TURN COORDINATOR

4/ VOR

5/ TACHYMETRE

6/ GYROCOMPAS

7/ ALTIMETRE

8/ HORIZON ARTIFICIEL

9/ AIR SPEED (IAS)

10/ VERTICAL SPEED (VARIO)

11/ ADF

12/ CARTE INTERFACE

13/ CODES SIOC

1/ GENERALITES

L'adaptation d'instruments réels est à la portée de bon nombre de passionnés de simulation. La première étape étant de trouver les instruments en état relativement correct mais évidemment non fonctionnels (ce qui rend leur prix accessible , autour des 30 euros en moyenne) On trouve ces instruments évidemment sur le net ou par relations , ou par hasard (brocante , puces aéro ..)

On utilisera les cartes OpenCockpit suivantes :

-Master

-USB Servo

-USB Stepper

Le tout est piloté par un code SIOC

2/ OUTILLAGE

Au minimum il faut un étau , une scie sauteuse , une perceuse à colonne (pas indispensable mais préférable) , des tarauds M3 , une lime électrique à bande et l'outillage courant du bon bricoleur ...

Pour ma part , j'ai un petit tour d'établi (utile pour faire les bagues d'accouplement et quelques petites pièces) on peut faire sans tour en travaillant avec la perceuse et la lime électrique .

Réalisation du Turn Coordinator

La réalisation nécessite 2 servos : un pour la silhouette avion et un pour la bille.
Les débattements sont faibles (inférieurs à 90 °) et on attaque en direct via un petit accouplement souple fait maison.
La pièce la plus délicate est le système de balancier qui porte la bille.

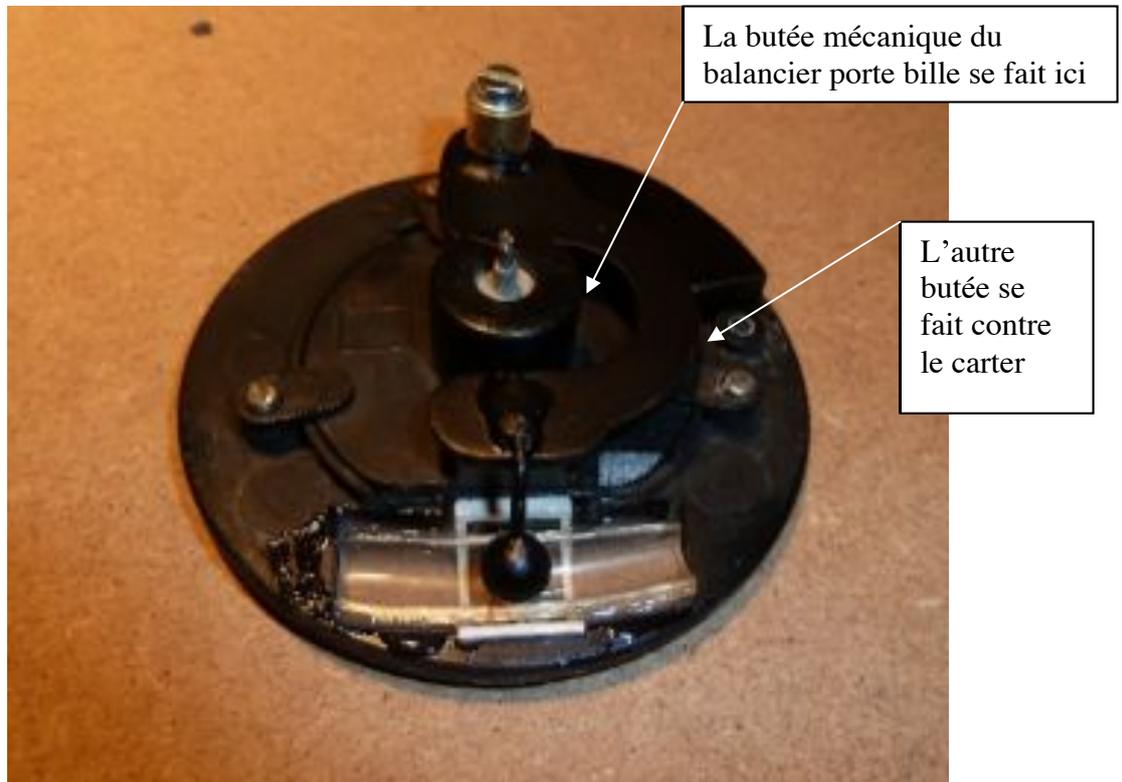
La bille est en acier doux mise en forme sur une perceuse et à la lime .Le tout est brasé sur le support.

Le balancier et la bille



La fausse capsule de verre est réalisée avec un morceau de plastique transparent.
 Sur la photo il s'agissait d'un essai avec un morceau de tuyau plastique , mais le résultat n'était pas trop bon car le tuyau était trop épais.Pas facile d'imiter un tube de verre avec une fente en haut pour passer la cde de la bille ..

Montage balancier



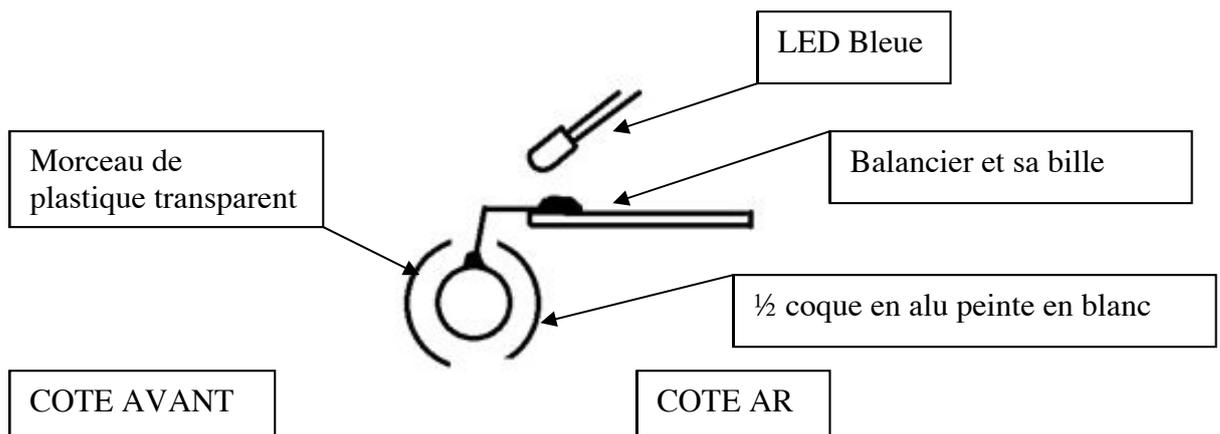
La butée mécanique du balancier porte bille se fait ici

L'autre butée se fait contre le carter

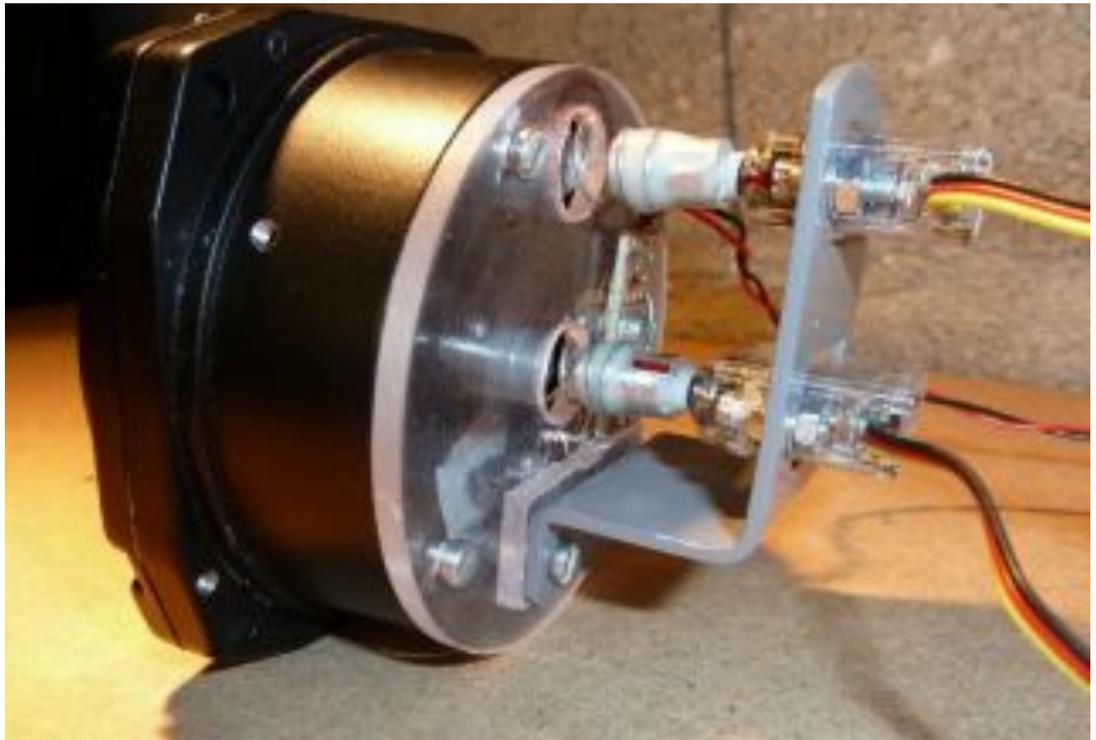
A l'arrière de la bille , j'ai placé une petite pièce en forme de 1/2 coque réalisée en alu (boite de conserve) et peinte en blanc.

L'éclairage se fait par une led bleutée placée assez loin de la fenêtre de manière à faire un éclairage assez faible qui se reflète sur le blanc de la pièce en forme de coque.

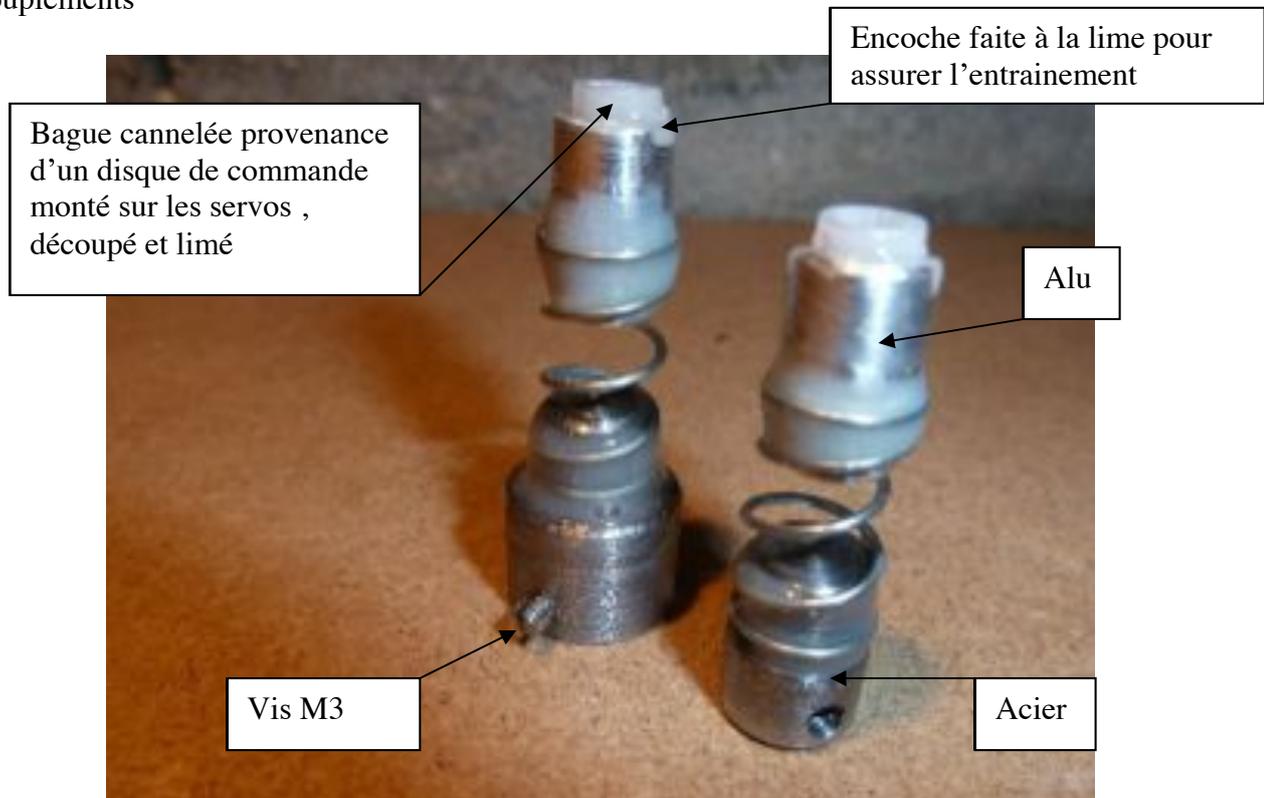
Avec le plastique devant qui imite le tube , l'illusion de la bille dans son tube de verre est assez bonne..



Montage des servos (Type 9G)



Les accouplements



L'instrument fini



Réalisation d'un VOR

Pour cette gauge , j'ai choisi de nouveau de commander l'aiguille par un servo. Bien que dans le cas présent , il soit possible (je pense..) de faire dévier l'aiguille (qui est un galvanomètre) en appliquant une tension continue adéquate sur le cadre de l'équipage mobile.

Dans mon cas , l'instrument était un peu endommagé et l'aiguille pas en bonne condition , donc la commande par servo s'imposait.

Cette gauge comporte également un dispositif de drapeaux à cde électromagnétique (là aussi certainement pilotable électriquement par notre équipement).

J'ai opté pour une signalisation par 3 LED remplaçant les drapeaux .Il y a donc 3 fenêtres à éclairer : FROM / TO / LOC (en rouge).

Ici ,on a une déviation de quelques dizaines de degrés de part et d'autre de la position centrale ,donc un servo en cde directe est tout à fait approprié.

Dans un première version de l'instrument , je n'avais pas placé de butée mécanique pour la course angulaire en comptant uniquement sur les réglages dans le code Sioc pour réaliser les butées .Le problème est que l'aiguille traverse le masque avant dans une sorte de découpe en forme de haricot et que c'est cette découpe qui risque de bloquer l'aiguille puis la tordre si on dépasse la butée programmée (ce peut etre le cas si on travaille en mode réglage avec le sioc monitor et les curseurs !)

Bref je recommande vivement de placer des butées mécaniques capables de bloquer la rotation aux extrêmes , sans que l'aiguille ne touche le masque avant .

On voit ici le montage du servo sur lequel est placé une bague fendue destinée à recevoir l'aiguille et un taquet pour la butée mécanique (qui sera une simple vis M3)

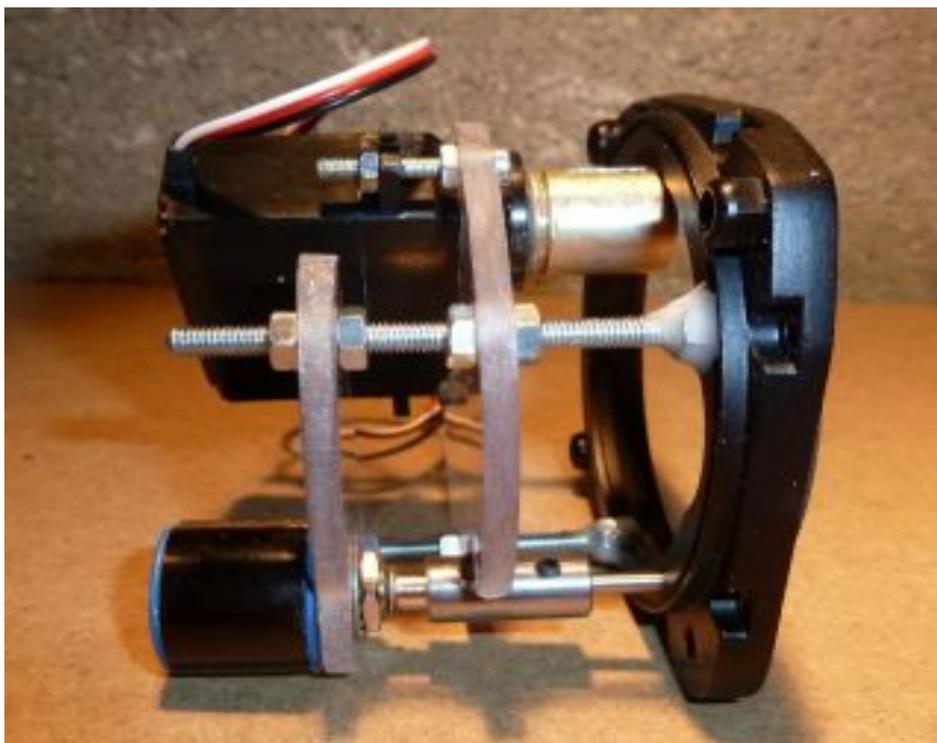


Pour le réglage de la radiale j'utilise un potentiomètre 10 tours commandé par le bouton d'origine qui entraîne donc aussi le cadran
L'ensemble des pièces avec l'aiguille sur son support , le potentiomètre 10 tours et son accouplement :



Support d'aiguille en Alu

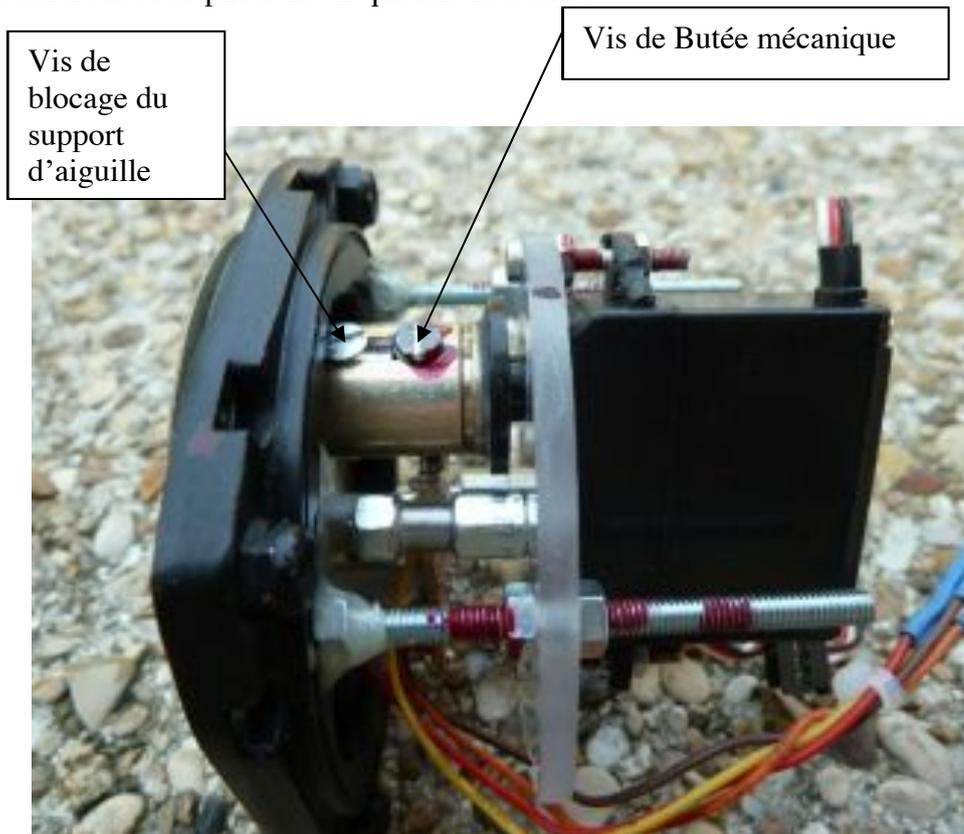
Ci dessous ,le début du montage avec le servo et le potentiomètre



Voici le système de butée mécanique , l'angle exact de rotation a été ajusté simplement en modifiant le décolletage sur la pièce hexagonale (on peut aussi faire une lumière pour déplacer la butée) :



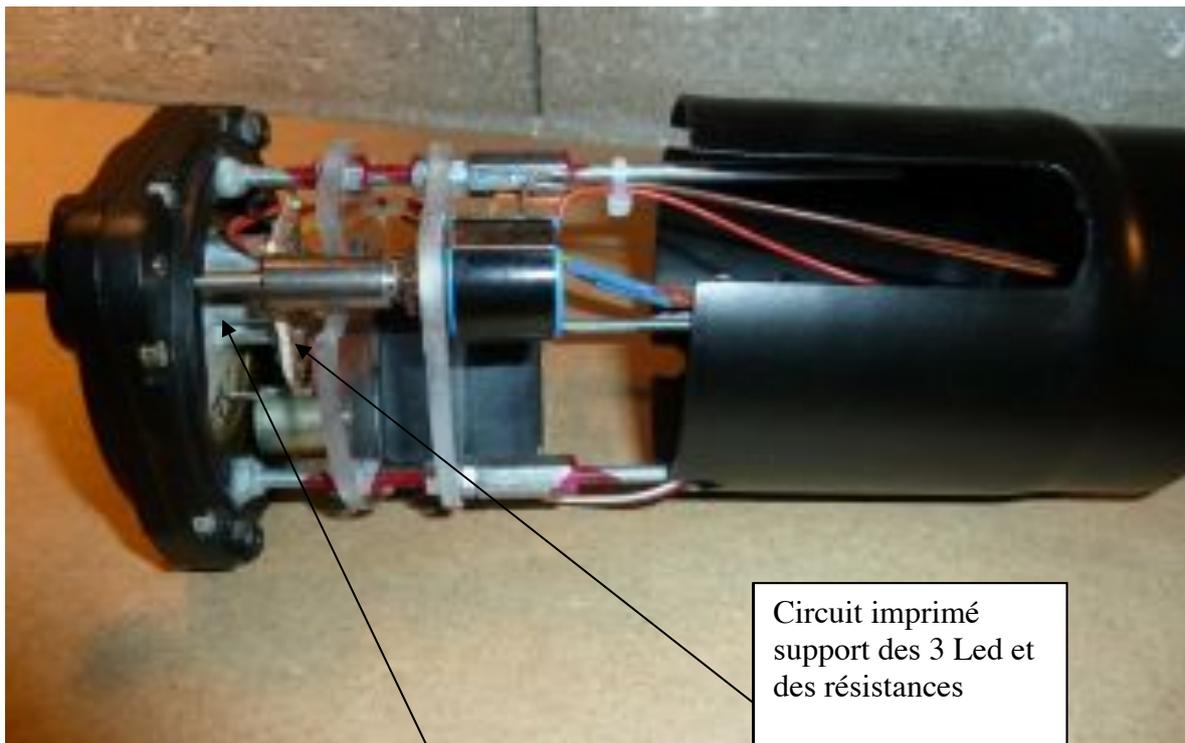
On voit aussi sur cette photo la vis qui sert de butée :



Le capot est réalisé en PVC (tuyau évacuation dia 80mm) et le fond en alu est le fond d'origine de l'instrument :



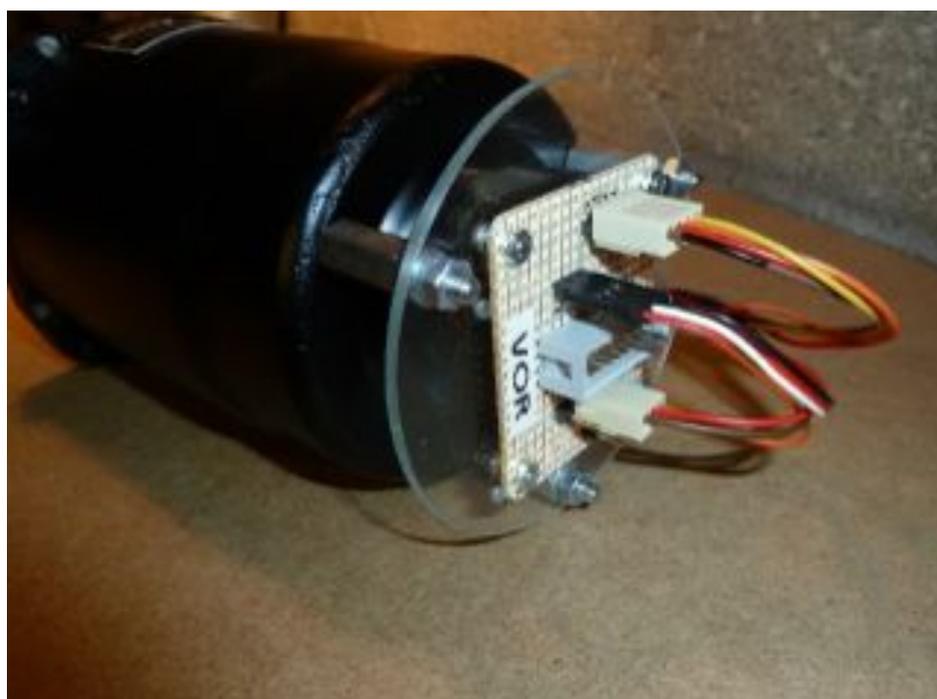
Pour les led il faut réaliser 3 tubes en alu et placer la led au fond de chaque tube, les tubes reçoivent à l'avant un petit disque trans lucide blanc découpé avec un emporte-pièce et collé à l'extrémité du tube Le système des tubes alu évite d'éclairer les fenêtres voisines.



Circuit imprimé support des 3 Led et des résistances

3 Tubes Alu

Comme pour les autres réalisations on utilise un connecteur pour cable plat :



Et voici l'instrument fini en vue de face :



Réalisation d'un Tachymètre (diam 50 mm)

Pour cette gauge , je n'utilise que très peu de pièces d'origine.

En fait l'instrument de départ était un manomètre (provenance aéronautique).

Donc la description peut s'appliquer à n'importe quelle gauge à 1 aiguille,demandant un angle de rotation de 360 ° approx (rapport engrenages 1/3 avec le servo)

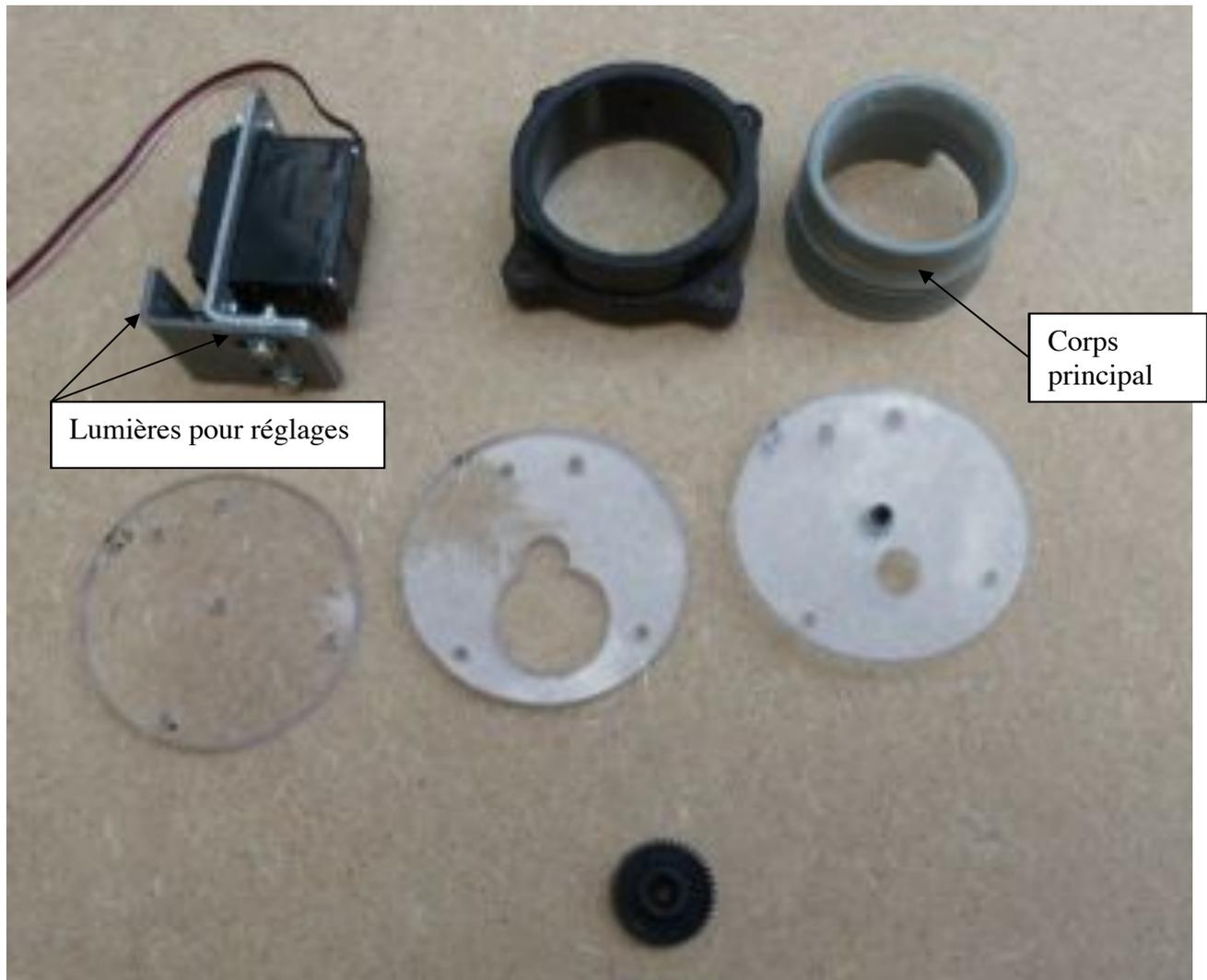
La pièce principale récupérée est la couronne avant avec sa vitre ,ceux qui n'ont pas cette pièce peuvent la fabriquer avec un empilage de plexi collé par exemple sur un tube PVC.

On réalise 3 trous à 120 ° pour mettre les vis de fixation de la partie principale (les vis sont juste en pression sur le corps principal) L'autre pièce récupérée a été l'aiguille , elle sera collée sur un axe en laiton de 3 mm .



On fabriquera les pièces suivantes

Notez le montage du servo sur une double équerre avec les trous en forme de lumière , ceci permet de régler le servo en position suivant 2 directions (très pratique)

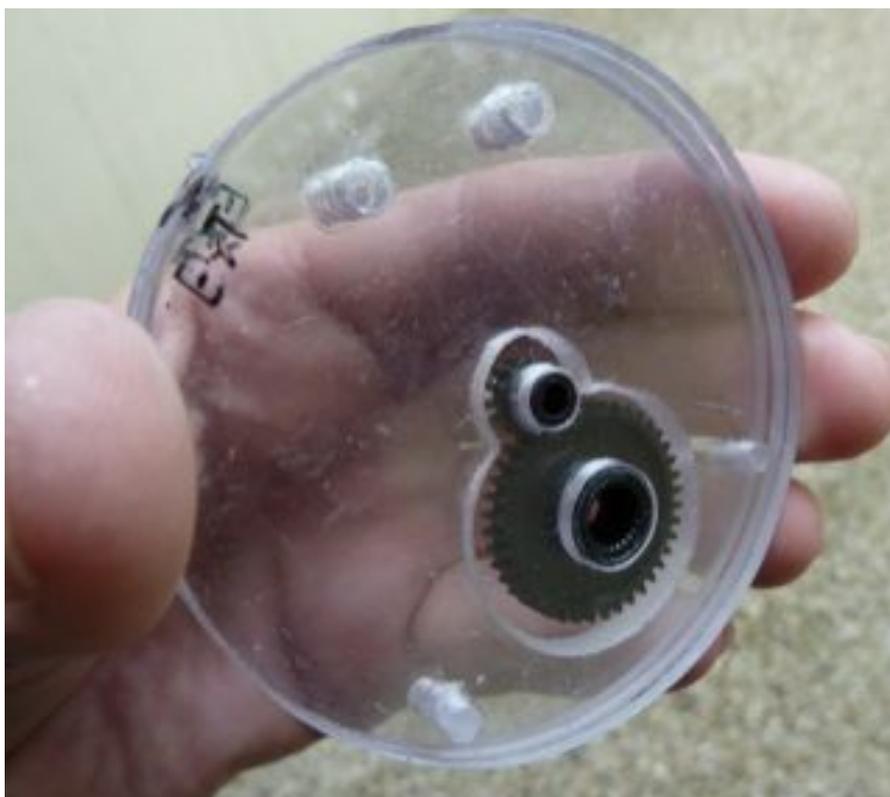


A droite on voit la pièce formant le corps principal .Elle est réalisée avec un raccord PVC (raccord de plomberie pour évacuation en dia 50mm). Elle a été entaillée généreusement avec 3 dégagements placés à 120 ° ceci facilite l'accès à l'accouplement pour le montage/réglage.

Le corps principal une fois collé sur son disque en plexi.



La cage recevant les pignons est réalisée avec 3 disques en plexi



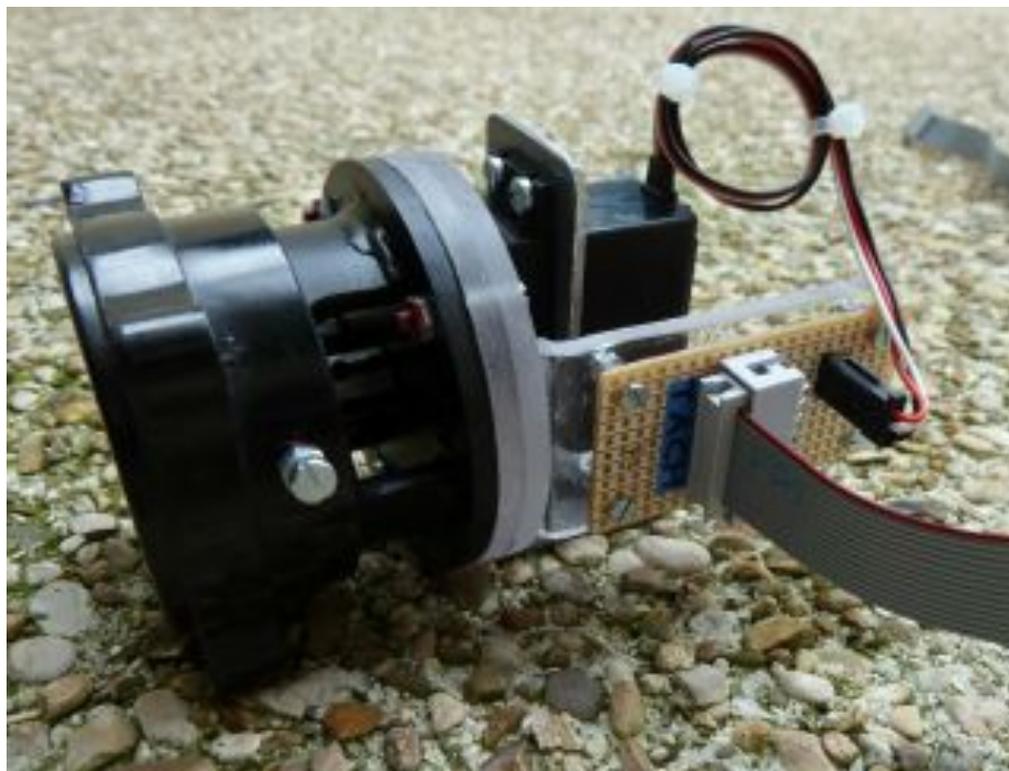
Une fois les pièces préparées on assemble et on obtient ça :



En vue de face , avec le cadran réalisé sur papier photo satiné collage sur le plexi avec colle en spray.



On installe les connecteurs , le cable plat a ici , de la réserve (on peut installer un éclairage par la suite)



Une fois monté sur le Tableau de Bord



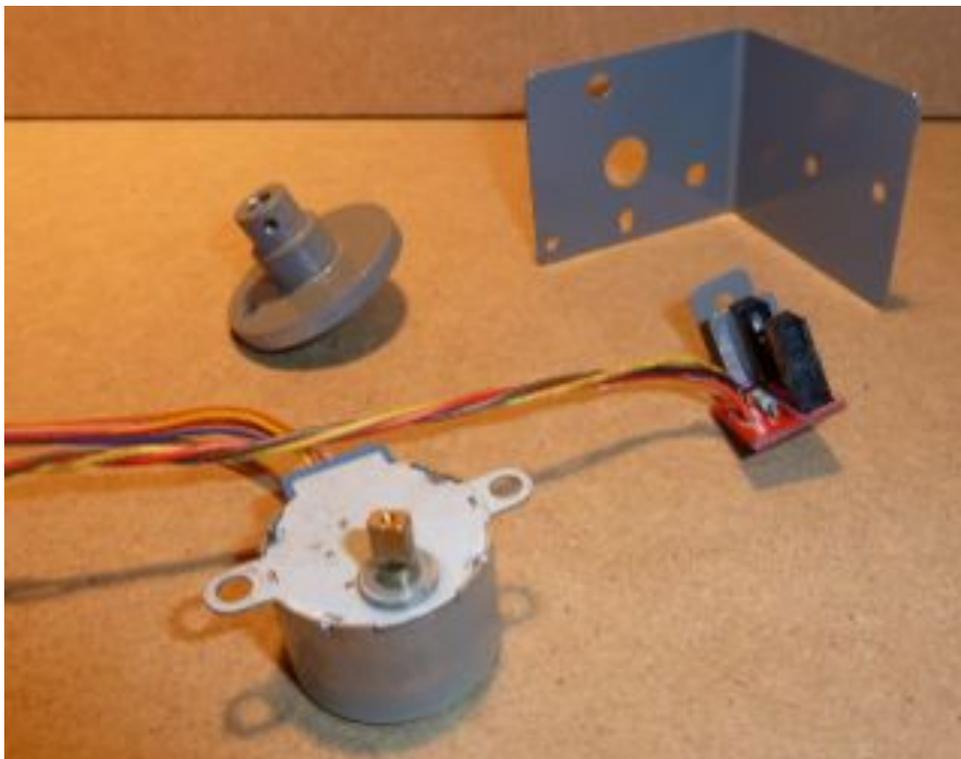
Réalisation d'un Gyrocompas

Le gyroscope (dans mon cas c'était un électrique) est déposé et on récupère en gros les pièces suivantes (la vitre sérigraphiée n'est pas sur la photo mais on la récupère bien):



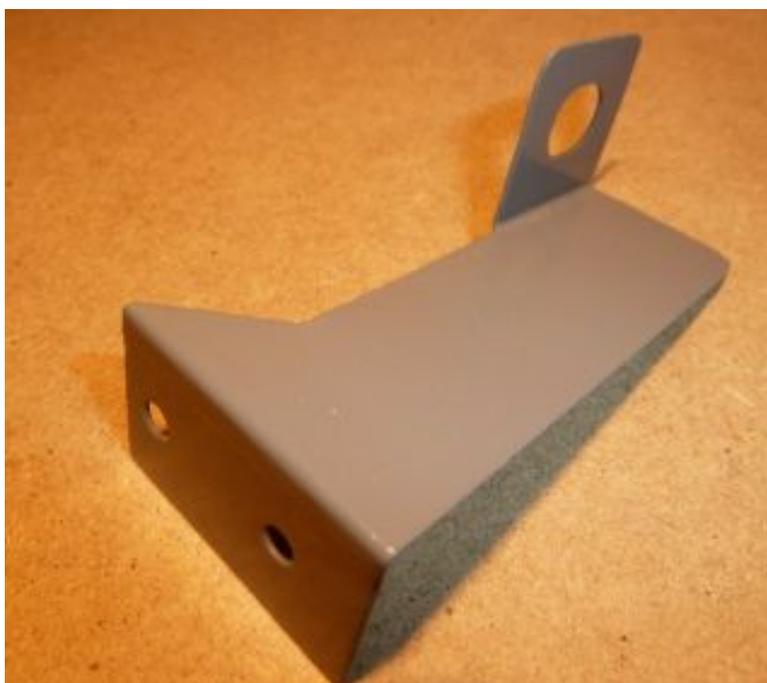
On fabrique les nouvelles pièces :support moteur,accouplement,support cellule (pour le calage du zéro)

Le stepper est un unipolaire avec réducteur 1/16 incorporé en 12v avec 48 pas x 16 = l'équivalent de 768 pas



On trouve une solution pour monter le potentiomètre de réglage (ici une sorte d'équerre mais c'est différent suivant votre instrument)

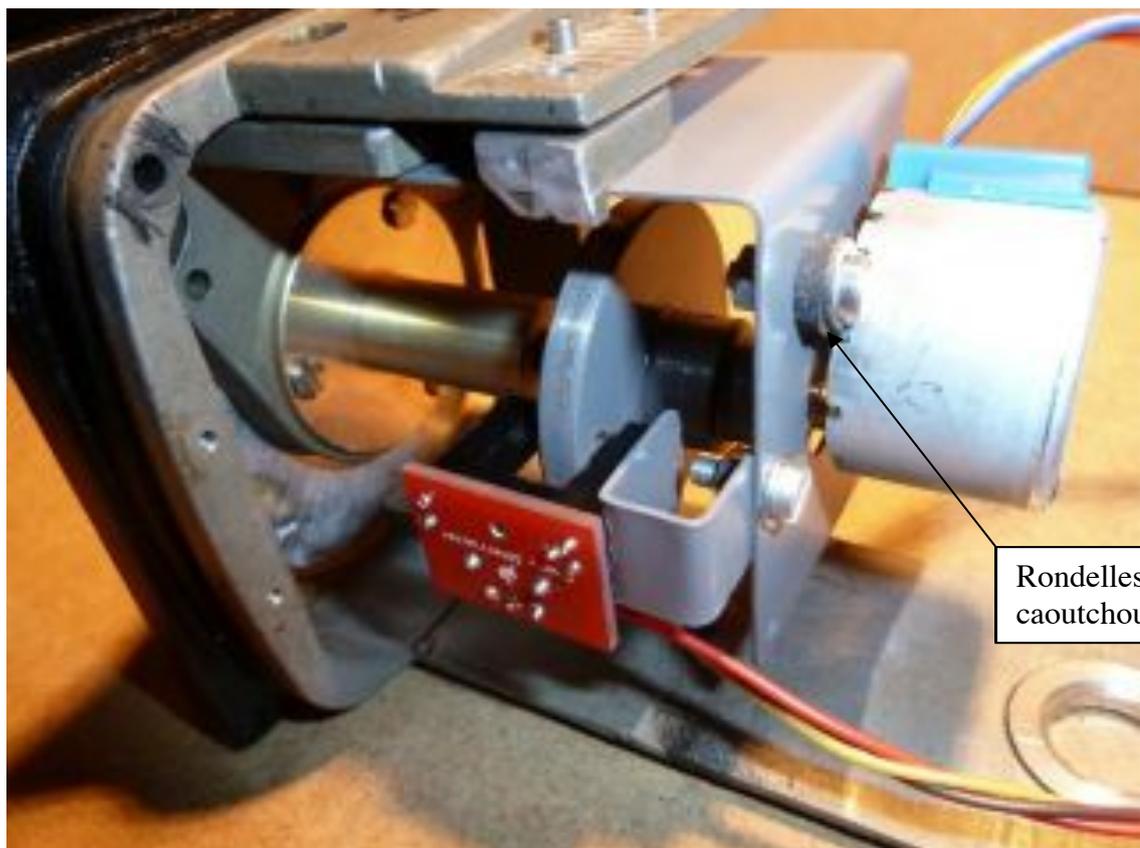
A noter le trou oblong pour le montage du pot afin de bien trouver l'alignement avec l'axe .



L'accouplement a deux fonctions : porter le disque obturateur pour la cellule et faire la jonction entre l'axe du cadran et le moteur.
En fait il est réalisé avec 2 pièces collées ,l'axe est fait avec un bout de rond acier et le disque est une rondelle plate ,le tout collé à l'Araldite.

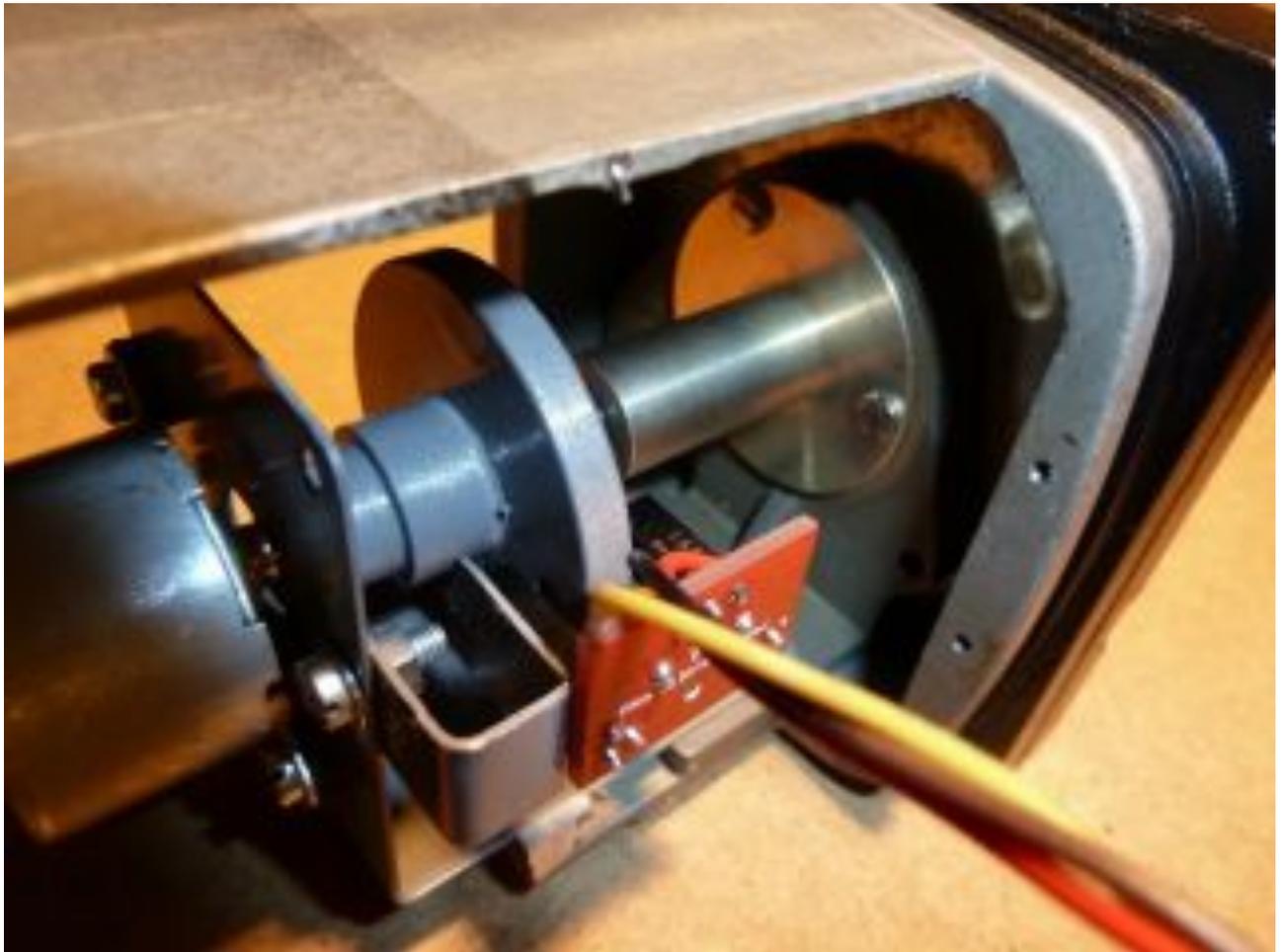


Vue de l'ensemble avec le capteur monté (notez les rondelles caoutchouc sur les pattes du moteur)

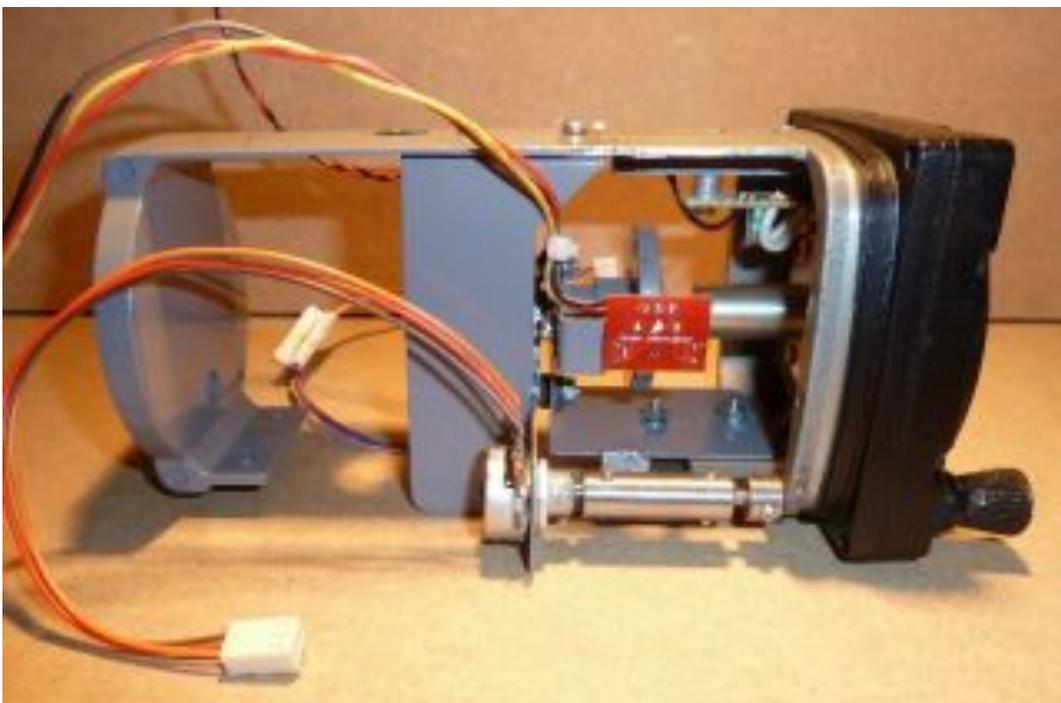


Rondelles
caoutchouc

Vue intérieure



Une fois tout monté on obtient ça , sur la photo le potentiomètre était un provisoire ,le définitif est un 10 Tours.
Le pot est raccordé à une entrée analogique de la Master.



Notez qu'il est possible et certainement préférable de remplacer le potentiomètre par un encodeur pour ne pas limiter la rotation du bouton aux 10 tours.

l'instrument prêt à être monté..



Réalisation d'un altimètre

Utilisation d'un altimètre réel (bouton du qnh faisant tourner correctement le disque gradué , sans point dur)

Voici les pièces principales récupérées (les aiguilles ne sont pas sur la photo):



On utilisera trois plateaux supports moteurs décalés à 120° et trois plateaux de guidage des pignons.

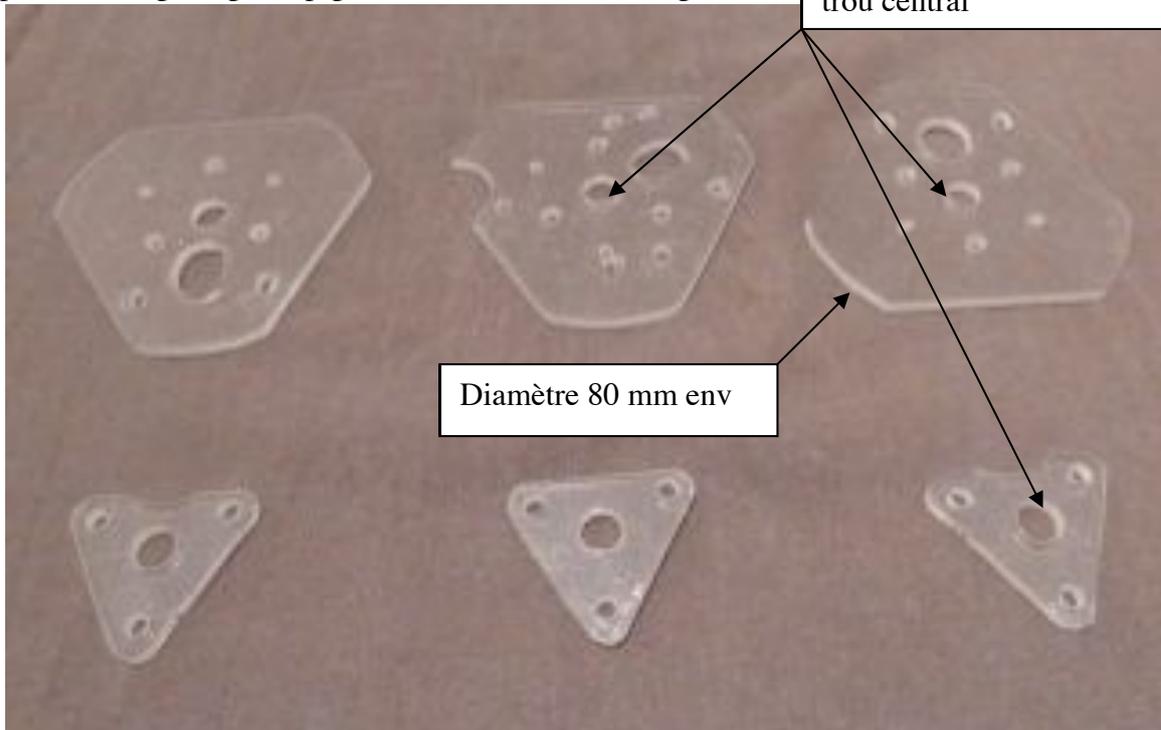
Les moteurs sont des pas à pas unipolaires avec réducteur en 12 v (Lextronic ref ADA 918)

Commencer par fabriquer en plexi (4 mm) les plateaux:

-3 plateaux supports moteurs (les grands)

- 3 plateaux de guidage du pignon d'entrainement de l'aiguille

Utiliser le trou central comme référence et percer les plateaux ensemble en les assemblant par le trou central



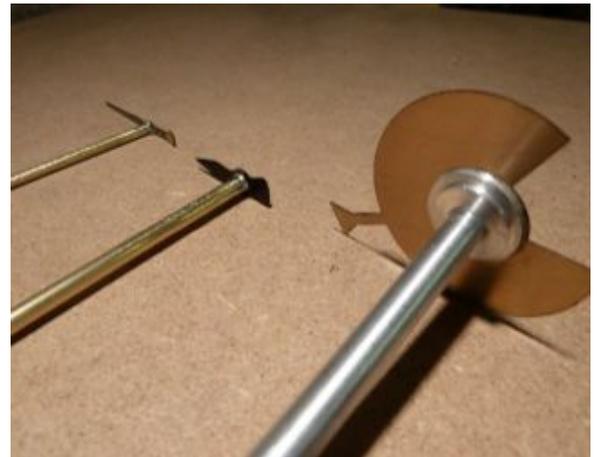
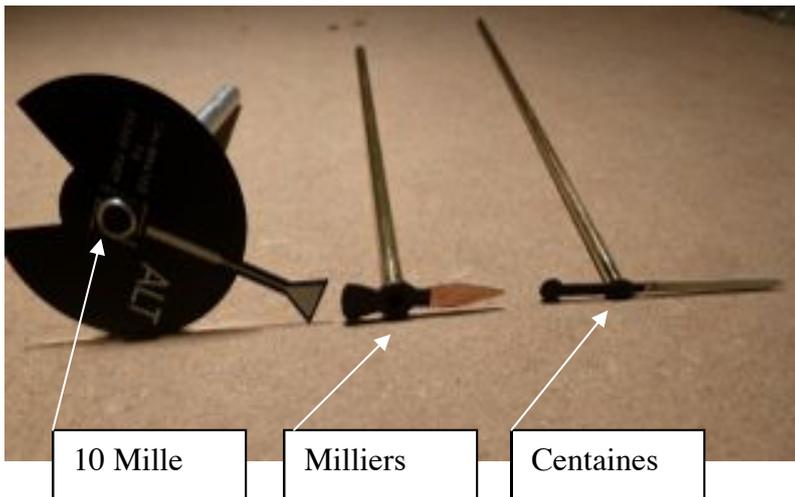
Préparer les axes:

-2 axes creux (tubes alu ou laiton) qui doivent tourner l'un dans l'autre sans trop de jeu.

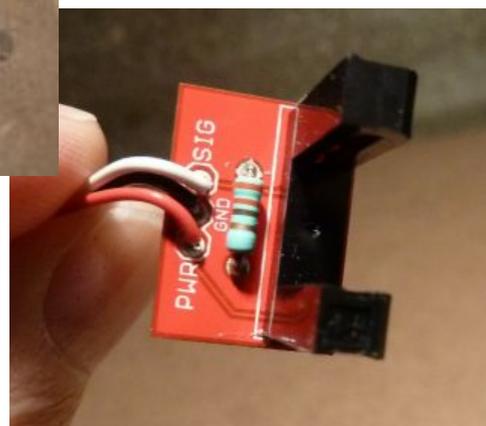
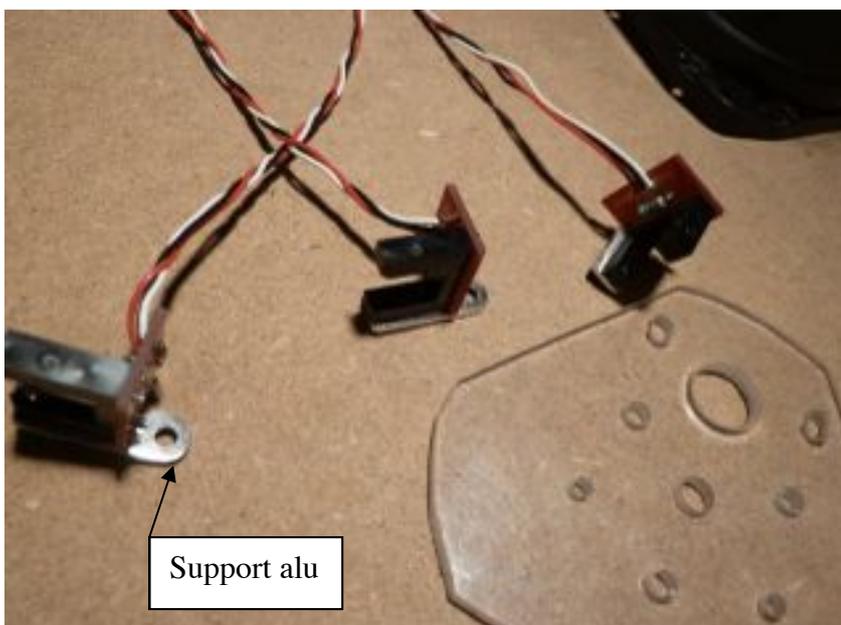
-1 axe plein (tige laiton diam 2 ou 3 mm)

Pour les aiguilles, j'ai utilisé celle des cents et des 10 milles d'origine et refabriquée celle des milles dont la taille a été augmentée pour être fixée sur l'axe.

Pour la fixation araldite ou brasure.

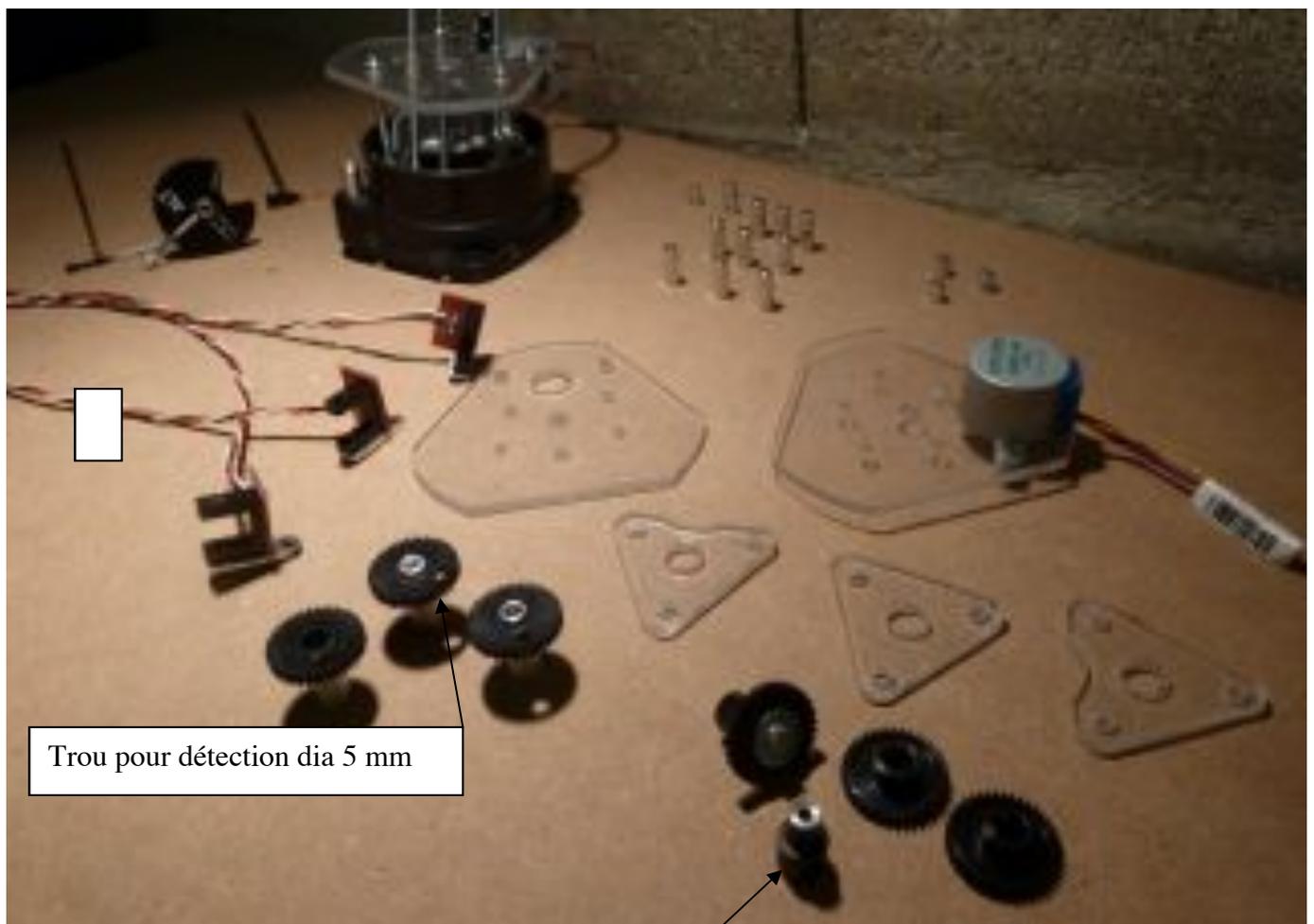


Réaliser les cellules de détection (cellules de chez Lextronic) collées sur un petit support alu (pour le réglage)



Les pignons sont dans le rapport 1/1 (diamètre environ 20 mm)
Le pignon sur l'axe de l'aiguille comporte un trou (dia 5 mm) pour la détection du zéro

fabriquer les tiges filetées à longr et les entretoises en tube alu .
Une fois tout ça terminé ça donne cet ensemble de pièces :



Trou pour détection dia 5 mm

Bague pour montage pignon
sur l'axe moteur

Le carter d'origine en alu doit être scié , meulé etc pour avoir le minimum d'encombrement en longueur
et permettre de fixer (araldite) les 3 premières tiges filetées:



Débuter le montage par le plateau moteur de l'axe des 10 milles (l'axe le plus gros):

1er plateau



Axe aiguille



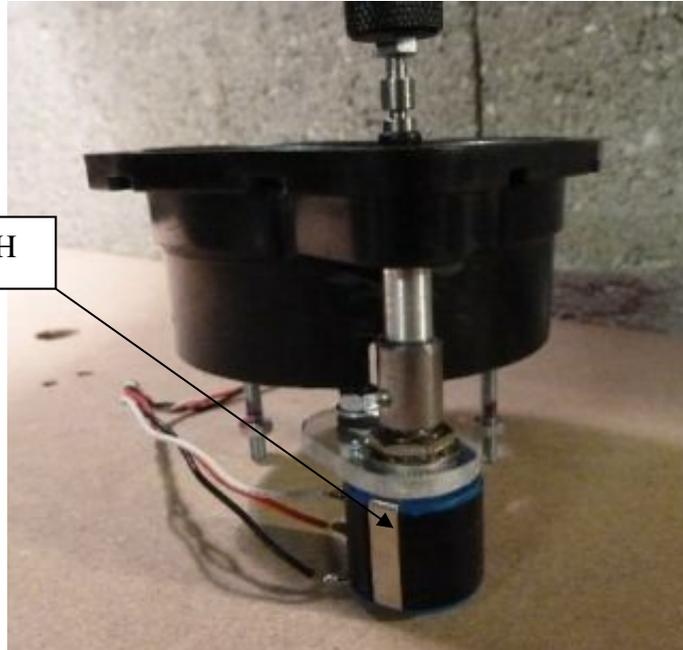
La cellule de detection est montée
Coté pignon relié à l'aiguille

Monter le potentiomètre de réglage QNH (10 tours 10 kOhm) ,il est monté en souple grâce à un morceau de caoutchouc (genre joint de robinet avec un trou coupé en deux et placé de chaque coté de la plaxe support).

L'accouplement es un morceau en acier avec trous M3 pour vis pointeau.

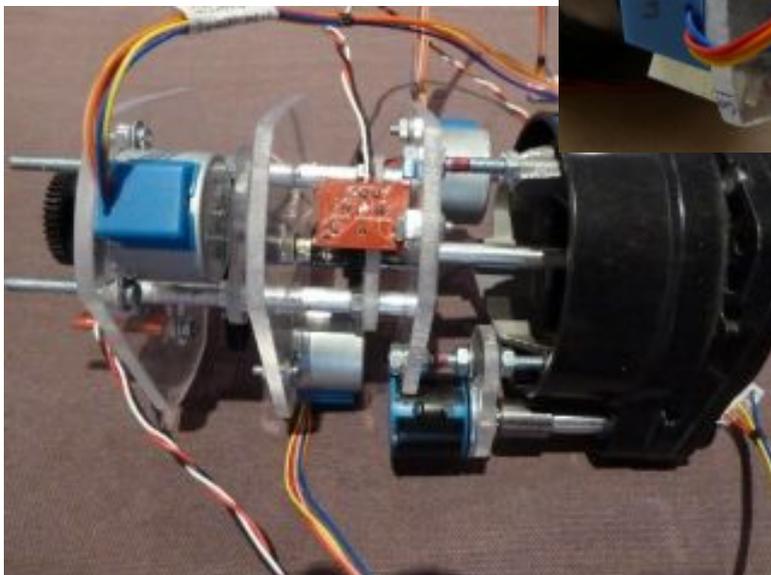
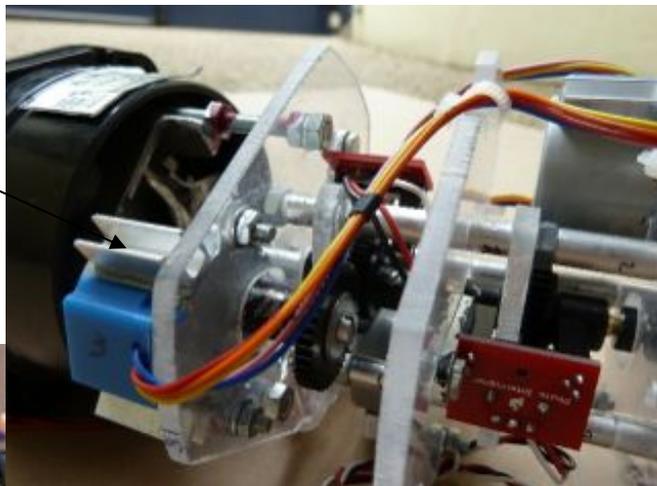
Potentiomètre de réglage QNH

Potentiomètre de réglage QNH

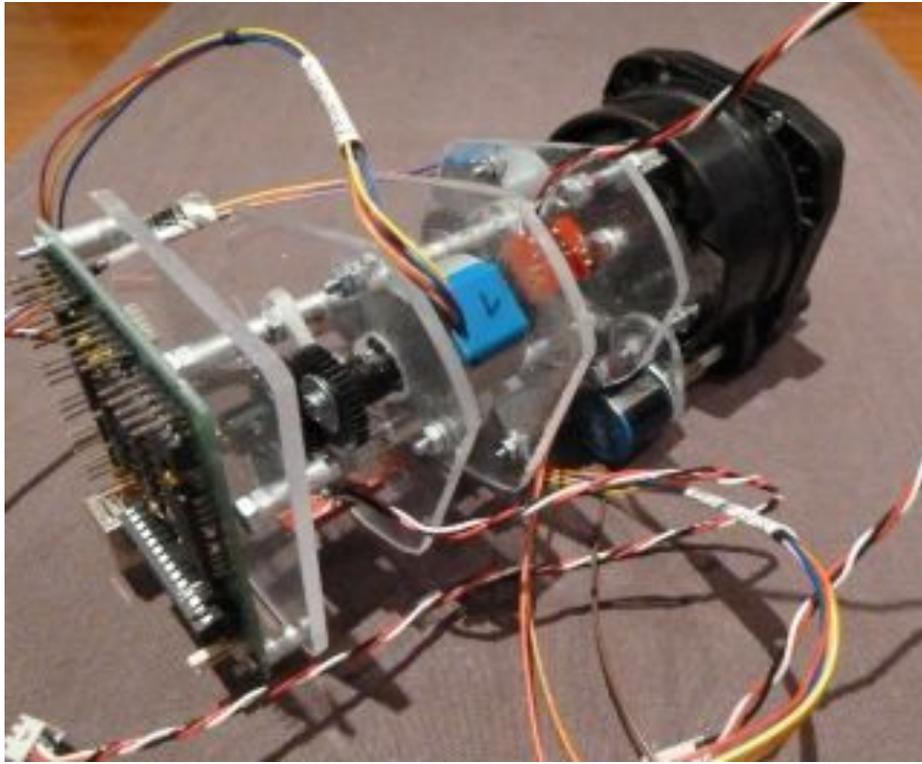


Les moteurs chauffent légèrement (normal) mais j'ai placé des petites ailettes collées (araldite) visibles sur la photo.

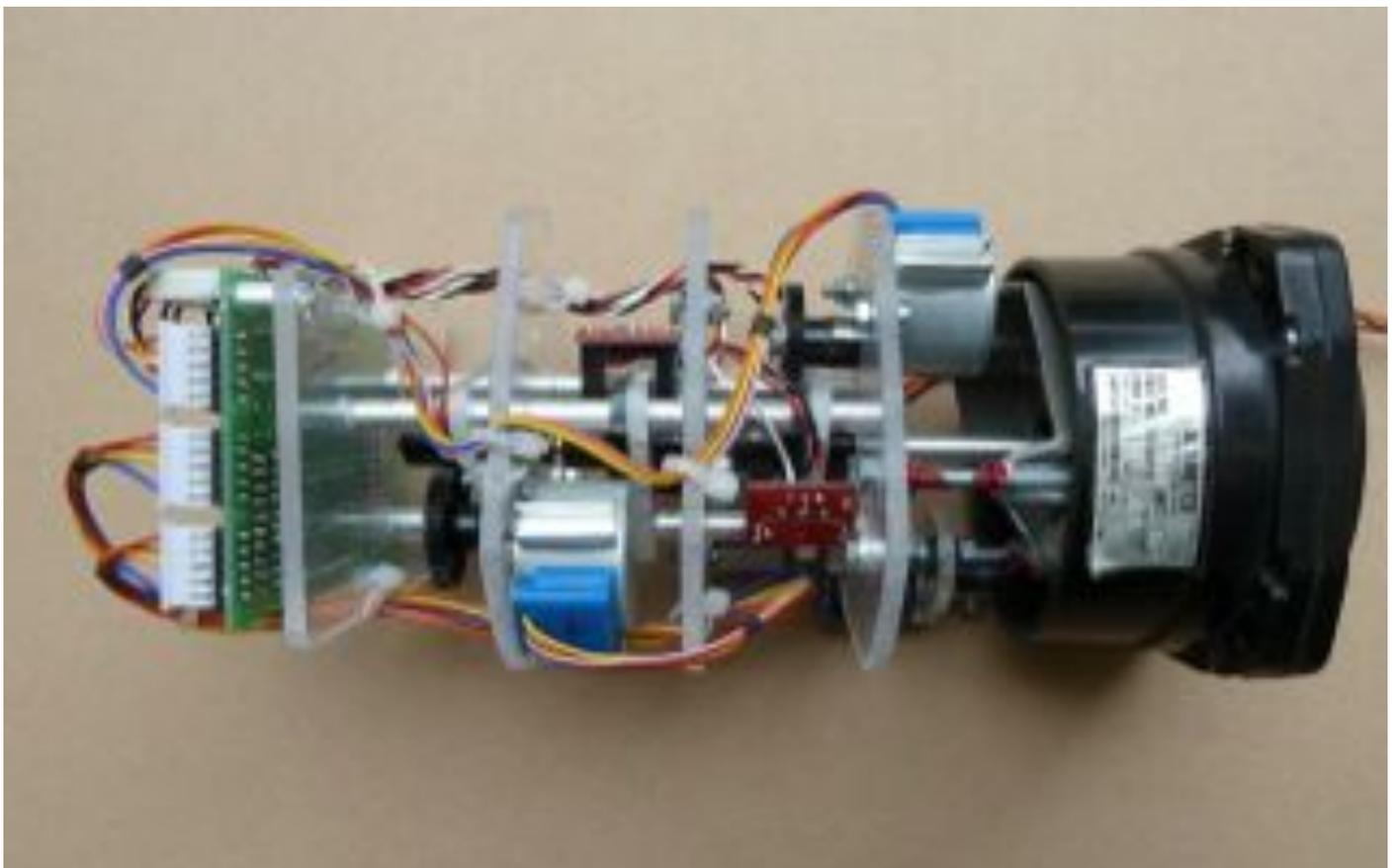
Detail ailettes moteur



La carte OC usb stepper est montée à l'arrière , sur une plaque plexi , le moteur est fourni avec un connecteur qui se branche directement sur la carte (le brochage correspond exactement à ce qu'il faut):



Ensemble terminé :



Pour le code sioc , je ne suis pas spécialiste mais j'ai réussi à sortir qqchose qui fonctionne.

Le principe:

Les 3 moteurs sont totalement indépendants

La valeur de l'altitude lue dans fsx par fsuipc est décomposée en 3 nombres en utilisant la fonction Modulo (reste de la division)

La valeur est ensuite mise en forme pour correspondre à un angle à envoyer au moteur.

Pour le calage au zéro :

La carte OC envoie la séquence de calage automatiquement dès la mise sous tension , reste une fois les 3 zéros détectés , à caler mécaniquement les aiguilles

sur la graduation zéro , le mieux possible (vis pointeau sur les pignons) puis on peut régler finement dans le programme en jouant

sur une variable d'ajustement (dans mon code c'est la &Ajust_Mill pour l'aiguille des milliers

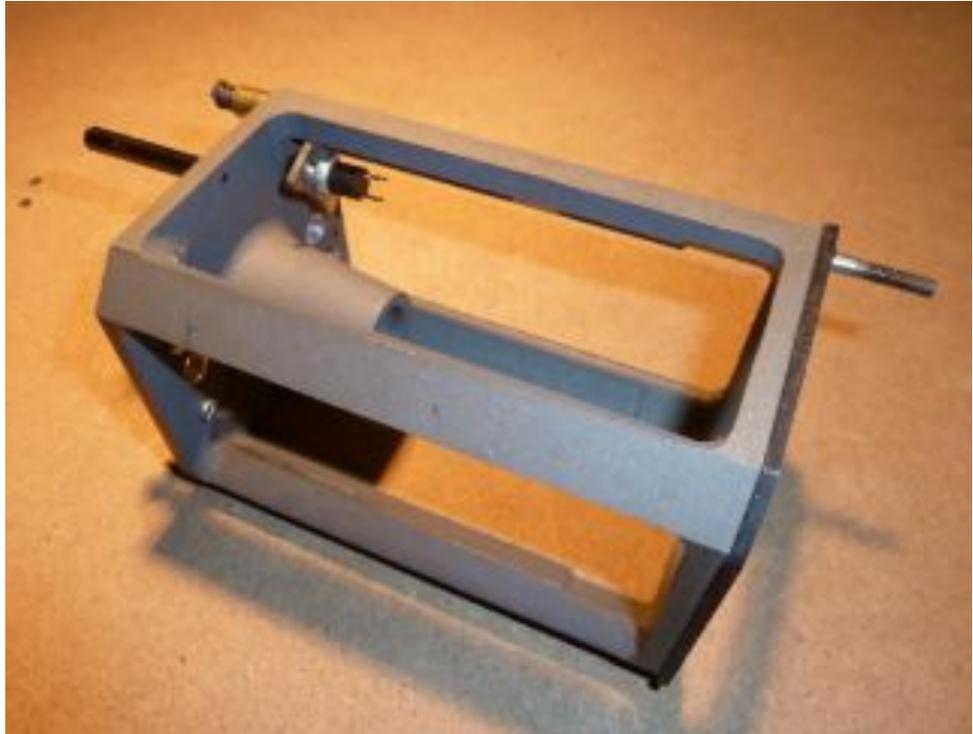
) Une fois tout réglé on colle au vernis les réglages des

vis poiteau , on retombe parfaitement au zéro à chaque mise sous tension et dès que fsx est lancé l'altitude est indiquée en une fraction de seconde ..

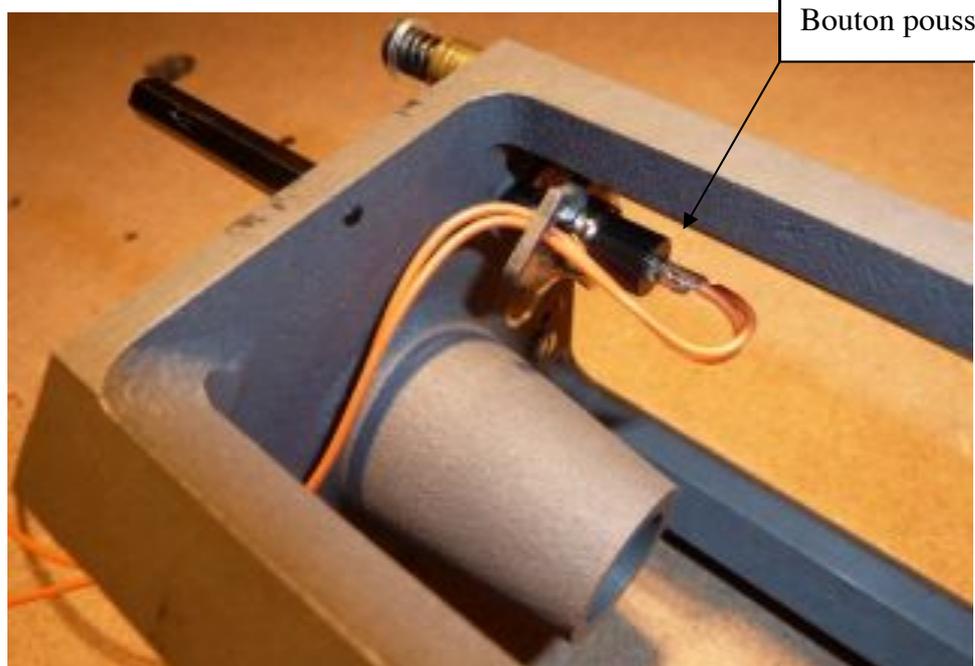
Réalisation d'un Horizon Artificiel

Le matériel dans mon cas est un "Badin-Crouzet" des années 70 en assez non état.
Après démontage soigneux on se retrouve avec un bati fixe et une partie mobile montée sur roulements.

Le bati fixe

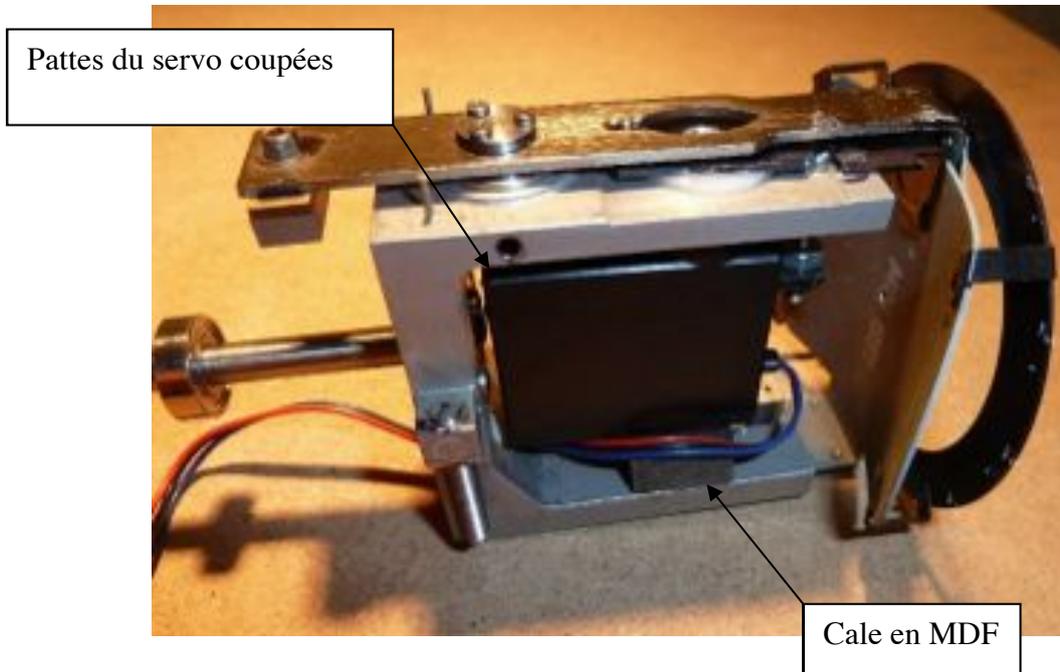


j'ai placé un bouton poussoir (contact normalement fermé) commandé par le bouton à tirer en façade ,ce bouton sert à recaler l'équipage mobile horizontal avec l'instrument , c'est une came ,pour l'instant j'ai raccordé le bp sur la master mais pas programmé cette fonction.



Pour la partie mobile , le servo est fixé sur une cale en medium et par 2 vis (il a fallu couper 2 pattes du servo)

Le maneton est fixé sur le servo par simple collage à l'araldite



Une plaque pour le montage du servo AR est réalisée en acier ou alu (ici c'était en acier)

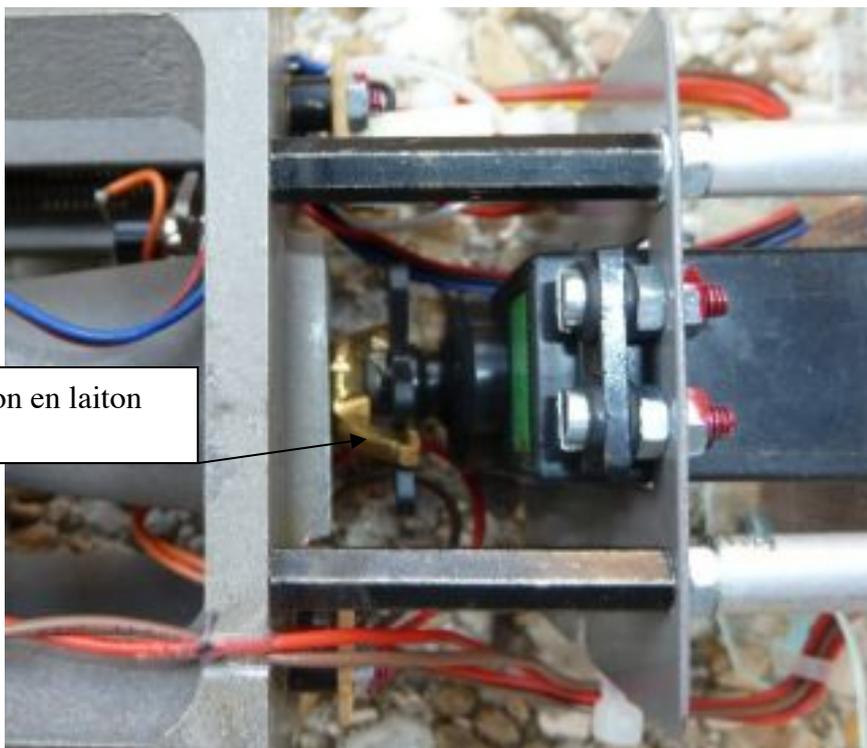


L'accouplement a été réalisé simplement avec un écrou à oreilles en laiton (écrou papillon), un peu retravaillé à la lime.

L'écrou est un M6 (je crois) et correspond à l'extrémité de l'équipage mobile qui se termine par un bout d'arbre fileté.

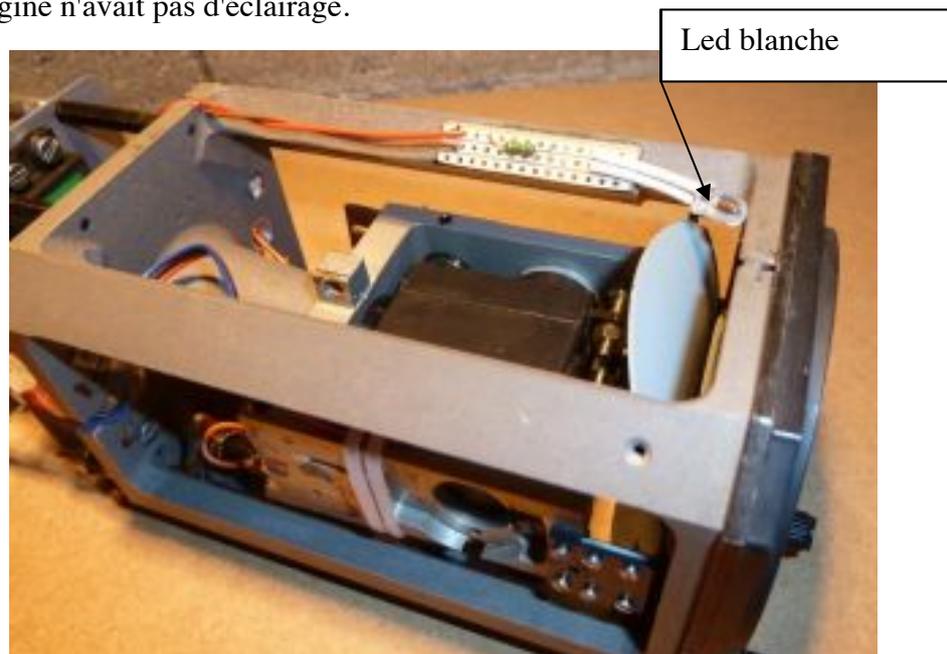
Le servo comporte une croix standard , on a donc un accouplement croix/fourche qui laisse un peu de liberté

dans l'alignement mais n'a pas de jeu parasite (si bien réglé)



Écrou papillon en laiton

Ne pas oublier l'éclairage j'ai utilisé 1 seule Led , je conseille d'en mettre 2 (une de chaque coté).
L'instrument d'origine n'avait pas d'éclairage.

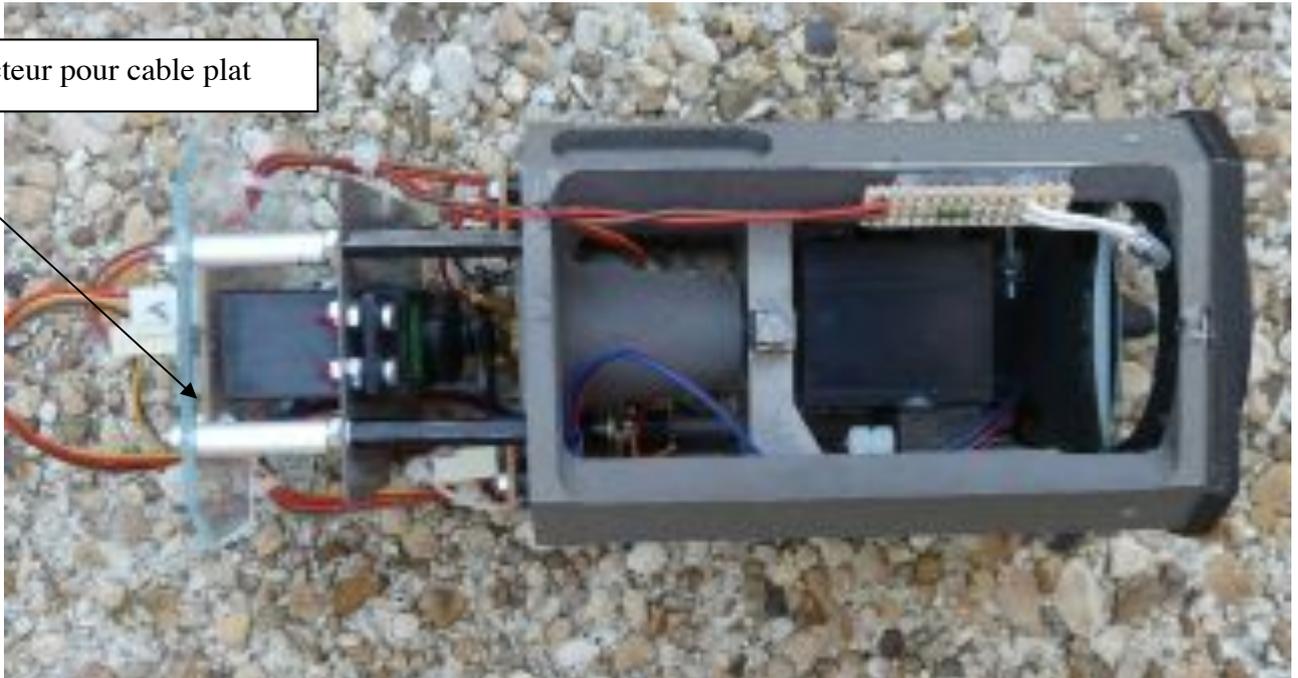


Vue latérale de l'ensemble



Vue de dessus

Connecteur pour cable plat



L'instrument prêt à être monté



Résultat final



Réalisation de la gauge IAS

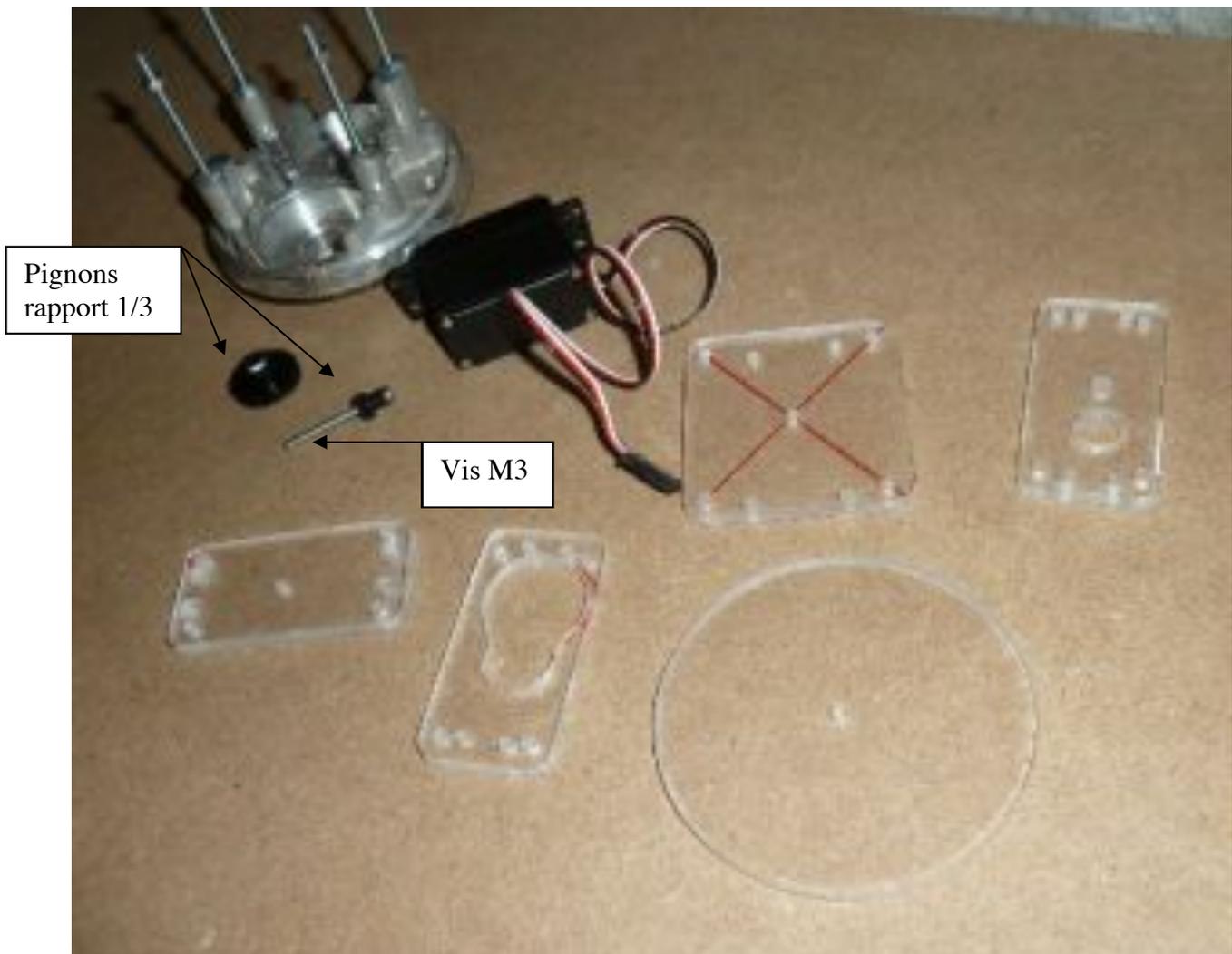
Simple dans sa mécanique un peu plus d'attention sur la programmation car ce n'est pas une loi linéaire ,mais à la portée d'un novice en sioc comme je

J'ai utilisé un servo basic Futaba et une démultiplication par pignons pour un rapport de 3 (l'axe de sortie peut donc faire quasiment 360° sans être en butée mécanique).

L'axe du pignon de sortie est une simple vis M3 collée à l'araldite sur le pignon et l'aiguille est faite avec un moyeu réalisé avec un écrou M3 brasé sur le corps de l'aiguille.

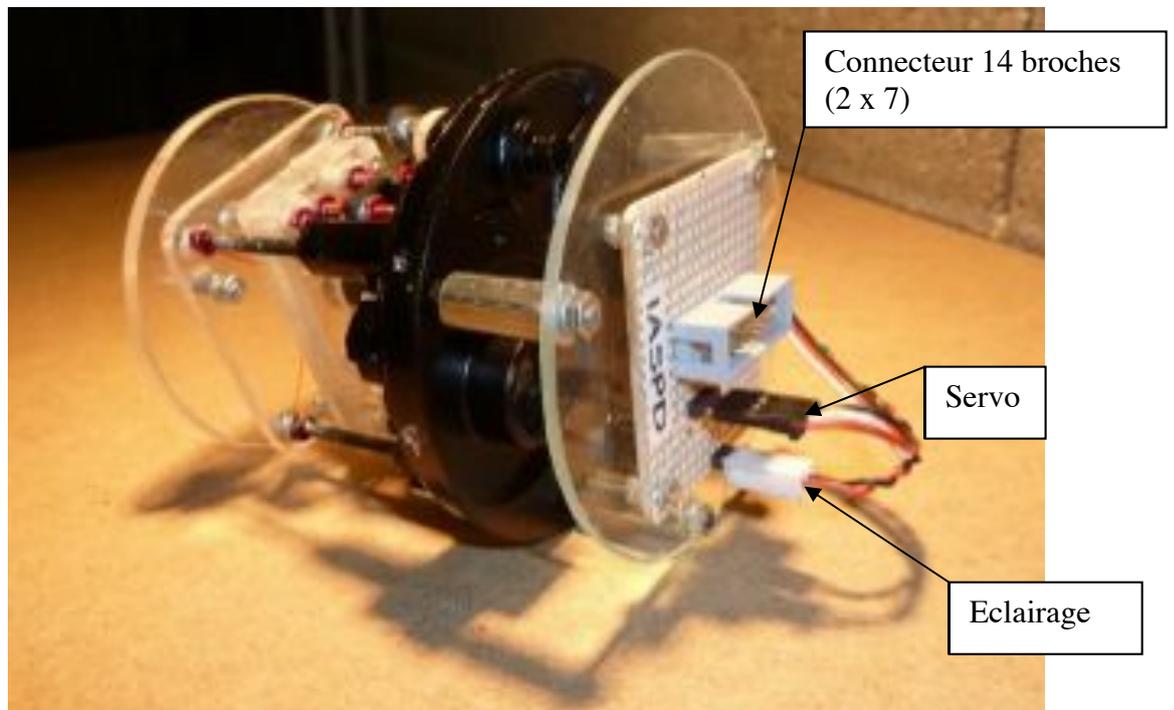
L'aiguille (fabrication maison en laiton) est donc vissée en butée sur cet axe .

L'ensemble des pièces



Le panneau de connexion sur l'arrière comporte le connecteur pour câble plat à 14 broches. J'ai adopté ce standard pour tous mes instruments ,ça évite les fils partout (il y en a déjà beaucoup)

Vue AR



La peinture a été refaite pour avoir du noir mat (au moment des photos ,c'était du brillant par erreur..) Pour les graduations c'est une photo (papier satiné) collée sur le plexi (colle en spray)



Réalisation de la gauge VSI (Vario)

Dans le principe , c'est la même chose que la gauge IAS mais j'ai monté les pignons dans une sorte de cage (idem le tachymètre).L'axe de sortie est aussi une vis M3 collée sur le petit pignon.

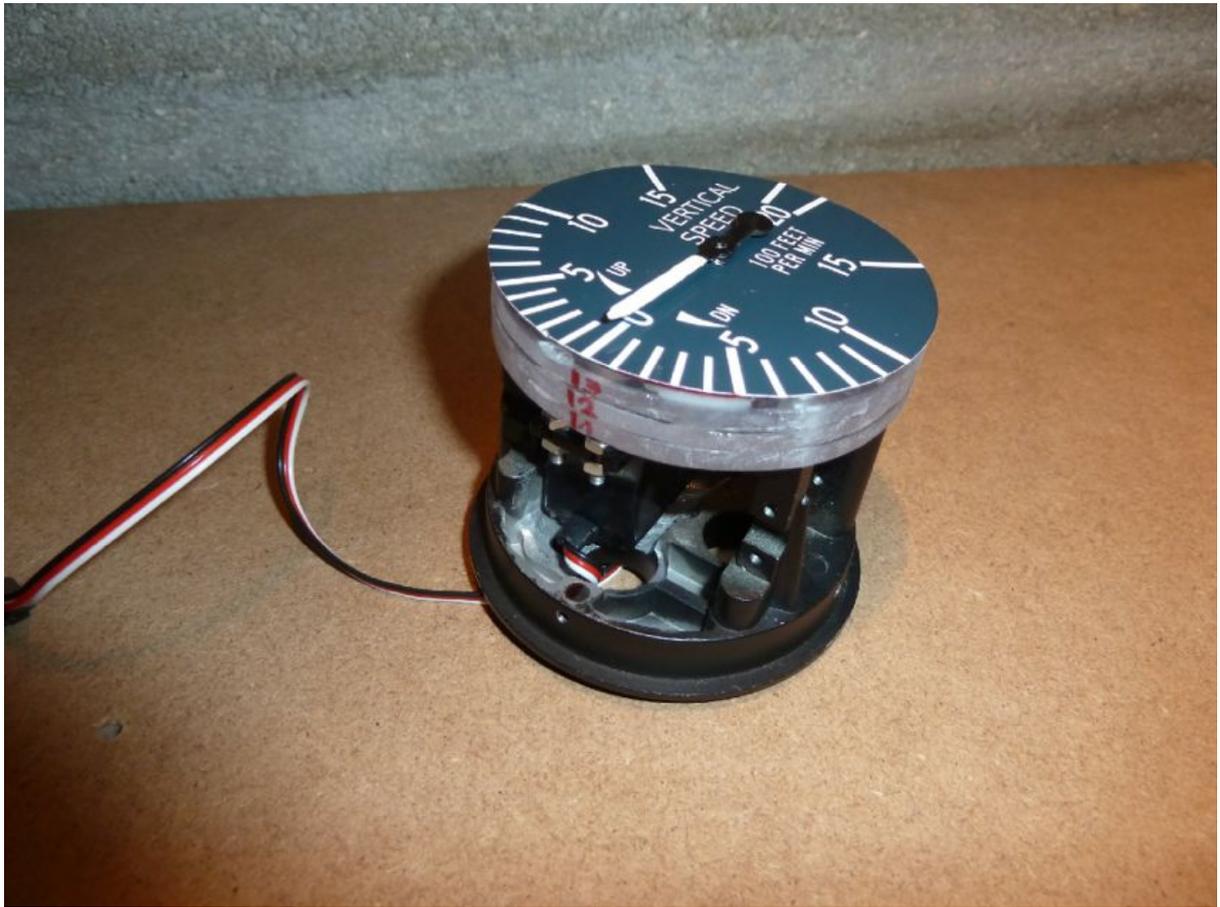
De mémoire cette gauge était en réalité un indicateur de température ou de pression
Ici , l'aiguille a été récupérée sur l'instrument d'origine , alors un petit décolletage en bout de la vis/axe (1.5 mm) juste pour servir de centrage à l'aiguille qui sera collée



Montage de la cage des pignons



Le cadran est fait par photo sur papier satiné et collé sur le disque plexi





ADF (Automatic Direction Finder)

Pour cette gauge , je n'ai pas de photos en cours de montage .

Ce qu'on peut dire c'est que le principe a adopter est exactement le même que pour le GyroCompas.

La différence réside dans le fait que pour le GyroCompas , c'est le disque cadran qui tourne devant un repère fixe et dans l'ADF , c'est l'aiguille qui tourne devant des graduation (disque) qui peuvent être mobiles en les faisant tourner à l'aide du bouton HDG

Donc ici , le stepper motor va attaquer en direct l'axe portant l'aiguille , on placera le même dispositif pour la synchronisation (disque avec trou)



A noter que dans cet instrument , j'ai choisi de ne pas renvoyer pour l'instant la position du cadran commandé par le bouton HDG vers Fsx , mais c'est possible en utilisant un encodeur.

CARTE INTERFACE

Pour relier tous les instruments sur les cartes OpenCockpit j'ai utilisé des câbles plats avec connecteurs .

Les câbles sont des 14 fils et les connecteurs organisés en 2 rangs de 7 pins.

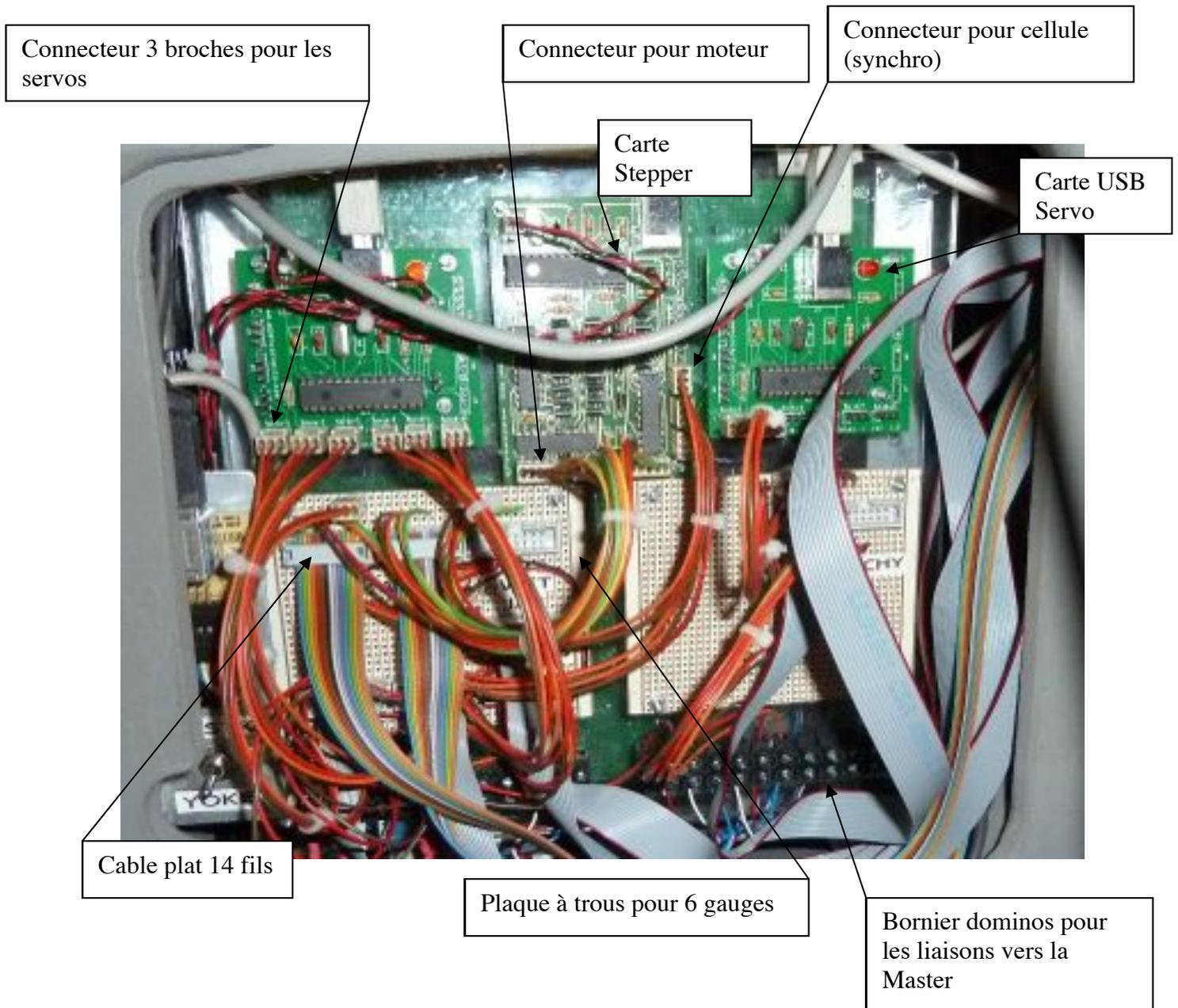
Tous les câbles sont identiques (hormis les longueurs)

Le sertissage des connecteurs se fait dans l'étau.

Les connecteurs (mâles) sont soudés sur un circuit imprimé à trous et en bandes parallèles.

Il suffit de couper (Dremel) les pistes pour faire les séparations des différents circuits.

Ensuite des petits câbles avec connecteurs (3 , 4 , ou 5 broches) assurent la liaison entre les connecteurs plats et les cartes OC .



CODE SIOC

Je joins le code sioc utilisé pour information uniquement .

Ceci est mon premier programme Sioc , il est loin d'être parfait mais il fonctionne correctement.

Il faudra adapter les équations à votre réalisation en faisant un étalonnage.

Les fonctions utilisées sont toutes de la forme $y=ax+b$ vous aurez à trouver vos coefs a et b en utilisant le sioc monitor pour faire bouger les aiguilles avec les curseurs et en notant les valeurs de commande à envoyer au servo. Des tutos et des feuilles excel existent sur le net pour faire ce calibrage.

Certaines gauges ne sont pas linéaires et il faut découper en tranches , en gros , de tant à tant , utiliser cette équation et de tant à tant utiliser une autre équation (cest le cas de l'IAS et du VSI)

Je n'ai pas procédé au nettoyage du code (supression des lignes et espaces en trop , mélange de l'anglais et français , etc , optimisation pour éviter les répétitions de code , etc ..)

Utilisez ces codes comme base pour écrire votre propre programme .

Ceci étant dit , voici le code complet pour les instruments :

HORIZONARTIFICIEL

//////////////////////////////// PITCH (servo n°2)////////////////////////////////

Var 0000, name init, Value 0 // initialisation

{

// &alt_fs_h = 0

}

Var 0025, name A2

Var 0026, name B2

Var 0023, name servo2, Link USB_SERVOS, Device 3, Output 2, PosL 330, PosC 569, PosR 781

Var 0024, name pi_value, Link FSUIPC_INOUT, Offset \$2F70, Length 8

{

 &A2 = &pi_value * 5

 &B2 = 569 - &A2

 IF &B2 > 781 // UPPER LIMIT

 {

 &B2 = 781

 }

 IF &B2 < 330 // LOWER LIMIT

 {

 &B2 = 330

 }

 &servo2 = &B2

}

//////////////////////////////// PITCH (servo n°2)////////////////////////////////

//////////////////////////////// BANK (servo n°1)////////////////////////////////

Var 0015, name A1

Var 0016, name B1

Var 0013, name servo1, Link USB_SERVOS, Device 3, Output 1, PosL 240, PosC 617, PosR 1020

Var 0014, name bk_value, Link FSUIPC_INOUT, Offset \$2F78, Length 8

```
{
```

```
&A1 = &bk_value * 4.58
```

```
&B1 = 617 + &A1
```

```
IF &B1 > 1020 // UPPER LIMIT
```

```
{
```

```
    &B1 = 1020
```

```
}
```

```
IF &B1 < 240 // LOWER LIMIT
```

```
{
```

```
    &B1 = 240
```

```
}
```

```
&servo1 = &B1
```

```
}
```

```
////////////////////////////////// BANK (servo n°1)//////////////////////////////////
```

AIR SPEED IAS

//////////////////////////////// IAS (servo n°3)////////////////////////////////

Var 0017, name A3

Var 0019, name B3

Var 0018, name servo3, Link USB_SERVOS, Device 3, Output 3, PosL 181, PosC 478, PosR 743

Var 0021, name C3

Var 0020, name ias_value, Link FSUIPC_INOUT, Offset \$02BC, Length 4

{

&B3 = &ias_value / 128 // &B3 est la vitesse en KNOTS

IF &B3 > 250 // UPPER LIMIT

{

&B3 = 250

}

IF &B3 <= 30 // COURBE 1 equation --> $y=0.80x + 194$

{

IF &B3 >= 0

{

&C3 = 0.8 * &B3

&C3 = &C3 + 194

&servo3 = &C3 // Sortie cde servo

}

}

IF &B3 <= 60 // COURBE 2 equation --> $y=1.63x + 169$

{

IF &B3 > 30

{

&C3 = 1.63 * &B3

&C3 = &C3 + 169

&servo3 = &C3 // Sortie cde servo

}

}

```

IF &B3 <= 100 // COURBE 3 equation --> y=2.58x + 112.5
{
  IF &B3 > 60
  {
    &C3 = 2.58 * &B3
    &C3 = &C3 + 112.5
    &servo3 = &C3 // Sortie cde servo
  }
}

```

```

IF &B3 <= 140 // COURBE 4 equation --> y=2.8x + 90
{
  IF &B3 > 100
  {
    &C3 = 2.8 * &B3
    &C3 = &C3 + 90
    &servo3 = &C3 // Sortie cde servo
  }
}

```

```

IF &B3 <= 180 // COURBE 5 equation --> y=2.9x + 76
{
  IF &B3 > 140
  {
    &C3 = 2.9 * &B3
    &C3 = &C3 + 76
    &servo3 = &C3 // Sortie cde servo
  }
}

```

```

IF &B3 <= 250 // COURBE 6 equation --> y=2.23x + 196.86
{
  IF &B3 > 180
  {
    &C3 = 2.23 * &B3
    &C3 = &C3 + 196.86
    &servo3 = &C3 // Sortie cde servo
  }
}

```

////////////////////////////////// IAS (servo n°3)//////////////////////////////////

VERTICAL SPEED (VSI)

```
//////////////////////////////// VERTSPD (servo n°4)////////////////////////////////
```

```
// Info: Vertical speed, signed,
```

```
// as 256 * metres/sec.
```

```
// For the more usual ft/min you need to apply the conversion *60*3.28084/256
```

```
Var 0040, name A4
```

```
Var 0041, name B4
```

```
Var 0042, name servo4, Link USB_SERVOS, Device 3, Output 4, PosL 299, PosC 584, PosR 854
```

```
Var 0043, name C4
```

```
Var 0044, name vertspd_value, Link FSUIPC_INOUT, Offset $02C8, Length 4
```

```
{
```

```
    &A4 = &vertspd_value * 60
```

```
    &A4 = &A4 * 3.28084
```

```
    &A4 = &A4 / 256
```

```
IF &A4 > 2000 // UPPER LIMIT
```

```
{
```

```
    &A4 = 2000
```

```
}
```

```
IF &A4 < -2000 // LOWER LIMIT
```

```
{
```

```
    &A4 = -2000
```

```
}
```

```
IF &A4 < 15 // Force Zero si VS entre 0 et 15 ft/min
```

```
{
```

```
    IF &A4 >= 0
```

```
    {
```

```
        &C4 = 0
```

```
        &servo4 = &C4 // Sortie cde servo
```

```
    }
```

```
}
```

```
IF &A4 > -15 // Force Zero si VS entre 0 et -15 ft/min
```

```
{
```

```
    IF &A4 <= 0
```

```
    {
```

```
        &C4 = 0
```

```
        &servo4 = &C4 // Sortie cde servo
```

```
    }
```

```
}
```

```

IF &A4 <= 500
{
    IF &A4 >= 0 // COURBE 5 equation --> y=0.12x + 584
    {
        &C4 = 0.12 * &A4
        &C4 = &C4 + 584
        &servo4 = &C4 // Sortie cde servo
    }
}

IF &A4 <= 1000
{
    IF &A4 > 500 // COURBE 6 equation --> y=0.14x + 572
    {
        &C4 = 0.14 * &A4
        &C4 = &C4 + 572
        &servo4 = &C4 // Sortie cde servo
    }
}

IF &A4 <= 1500
{
    IF &A4 > 1000 // COURBE 7 equation --> y=0.13x + 588
    {
        &C4 = 0.15 * &A4
        &C4 = &C4 + 560
        &servo4 = &C4 // Sortie cde servo
    }
}

IF &A4 <= 2000
{
    IF &A4 > 1500 // COURBE 8 equation --> y=0.15x + 558
    {
        &C4 = 0.14 * &A4
        &C4 = &C4 + 575
        &servo4 = &C4 // Sortie cde servo
    }
}

```

// valeurs negatives

```

IF &A4 >= -500
{

```

```

IF &A4 <= 0 // COURBE 4 equation --> y=0.12x + 584
{
    &C4 = 0.12 * &A4
    &C4 = &C4 + 584
    &servo4 = &C4 // Sortie cde servo
}
}

IF &A4 >= -1000
{
    IF &A4 < -500 // COURBE 3 equation --> y=0.15x + 600
    {
        &C4 = 0.15 * &A4
        &C4 = &C4 + 600
        &servo4 = &C4 // Sortie cde servo
    }
}

IF &A4 >= -1500
{
    IF &A4 < -1000 // COURBE 2 equation --> y=0.17x + 624
    {
        &C4 = 0.17 * &A4
        &C4 = &C4 + 624
        &servo4 = &C4 // Sortie cde servo
    }
}

IF &A4 >= -2000
{
    IF &A4 < -1500 // COURBE 1 equation --> y=0.13x + 567
    {
        &C4 = 0.13 * &A4
        &C4 = &C4 + 567
        &servo4 = &C4 // Sortie cde servo
    }
}

}
//////////////////////////////// VERTSPD (servo n°4)////////////////////////////////

```

TURN COORDINATOR

```
////////////////////////////////// TURN COORD_Ball (servo n°5)//////////////////////////////////
//Turn co-ordinator ball position : 036E (slip and skid). -128 is extreme left, +127 is extreme
right, 0 is balanced.
//Offset:036E Length:1
Var 0050, name A5
Var 0051, name B5
Var 0052, name servo5, Link USB_SERVOS, Device 3, Output 5, PosL 707, PosC 631, PosR
555
Var 0053, name C5
Var 0054, name ball_position, Link FSUIPC_INOUT, Offset $036E, Length 1, Type 1
{
```

```
    &A5 = &ball_position
```

```
// equation  $y = -0.608x + 631$ 
    &B5 = 0 // valeur pour ajuster la position zero (centre)
    &C5 = -0.608 * &A5
    &C5 = &C5 + 631
    &C5 = &C5 + &B5
```

```
    IF &C5 >= 707
    {
        &C5 = 707
    }
```

```
    IF &C5 <= 555
    {
        &C5 = 555
    }
```

```
    &servo5 = &C5 // Sortie cde servo
```

```
}
```

```
////////////////////////////////// TURN COORD_Ball (servo n°5)//////////////////////////////////
```

```
////////////////////////////////// TURN COORD_Turn Rate (servo n°6)//////////////////////////////////
// Turn Rate (for turn coordinator). 0=level ou 65535, (65535-512)=2min Left, +512=2min
Right // tricky !
//Offset:037C Length:2
Var 0060, name A6
Var 0061, name B6
Var 0062, name servo6, Link USB_SERVOS, Device 3, Output 6, PosL 300, PosC 650, PosR
1000
Var 0063, name C6
```

Var 0064, name turn_rate, Link FSUIPC_INOUT, Offset \$037C, Length 2

```
{
    IF &turn_rate >= 0
    {
        &A6 = &turn_rate
    }
    IF &turn_rate > 60000
    {
        &A6 = &turn_rate - 65535
    }

// equation y=0.211x + 735
&B6 = 0 // valeur pour ajuster la position zero (centre)
&C6 = 0.211 * &A6
&C6 = &C6 + 735
&C6 = &C6 + &B6

IF &C6      >= 940 // limite sup right angle de approx 6 deg/sec (1 tour en 1 min)
{
    &C6 = 940
}

IF &C6      <= 525 // limite sup left angle de approx 6 deg/sec (1 tour en 1 min)
{
    &C6 = 525
}

&servo6 = &C6 // Sortie cde servo

}

////////////////////////////////////// TURN COORD_Turn Rate (servo n°6)//////////////////////////////////////
```

GYROCOMPAS

```
////////// GYROCOMPAS //////////////////////////////////////
// Rotation of dial (not needle) clockwise =decrease angle

Var 0070, Link USB_STEPPER, Device 2, Output 1, PosL 5, PosC 0, PosR 4, Type H
Var 0072, name Ajst_Centr_GY
Var 0073, name Angle_ValueGY
Var 0074, name PotGyro_FSuipc, Link FSUIPC_OUT, Offset $0C3E, Length 2 // Pot value
send to FSUIPC *360/65536 for degrees
Var 0075, name intrGy1
Var 0077, name intrGy2

// Bouton de reglage

Var 0076, name POT_GYRO, Link IOCARD_ANALOGIC, Device 0, Input 3, PosL 0, PosC
127, PosR 255 // Input potentiomètre de 10kohm 10 tours Bouton Gyro//
{

    V0075 = V0076 * 1.412 // avec 1.412 = 360/255 donc V0075 est en degres
    V0077 = V0075 * 65536
    V0074 = V0077 / 360 // send to fsuipc

}

Var 0071, Link FSUIPC_IN, Offset $2B00, Length 8, Numbers 1 // FSimulator
gyroCompass variable 2B00
{
    &Ajst_Centr_GY = -200 // gauge center
    &Angle_ValueGY = V0071
    &Angle_ValueGY = &Angle_ValueGY * 10

    V0070 = &Angle_ValueGY + &Ajst_Centr_GY // V0070 send degrees to stepper
motor
    V0070 = 36000 - V0070 // Inversion du sens de rotation
}

////////// GYROCOMPAS //////////////////////////////////////
```

ALTIMETRE

////////// ALTIMETER //

```
Var 0082, name Ajust_Cent
Var 0083, name Ajust_Mill
Var 0084, name Ajust_DixMil
Var 0085, name Intermediaire1
Var 0086, name Intermediaire2
Var 0087, name Intermediaire3
Var 0088, Link USB_STEPPER, Device 1, Output 1, PosL 5 PosC 0, PosR 4, Type H //
device0=1ere carte stepper et output1=moteur1
Var 0089, Link USB_STEPPER, Device 1, Output 2, PosL 5, PosC 0, PosR 4, Type H
Var 0090, Link USB_STEPPER, Device 1, Output 3, PosL 5, PosC 0, PosR 4, Type H
Var 0081, name alt_fs_h, Link FSUIPC_IN, Offset $3324, Length 4 // Altimetre READING
from FS it is alt in feets
{
    &Ajust_Cent = 100 // gauge center (ie : 8500 = needle indicates 85 deg angle instead
of 0)
    &Ajust_Mill = 100
    &Ajust_DixMil = 100

// Aiguille Centaines (Moteur N°1)
L0 = MOD &alt_fs_h ,1000 // L0 is a real variable
V0085 = L0 * 36
V0088 = V0085 + &Ajust_Cent // send angle value to motor

// Aiguille Mill (Moteur N°2)
L1 = MOD &alt_fs_h ,10000 // L1 is a real variable
V0086 = L1 * 3.6
V0089 = V0086 + &Ajust_Mill // send angle value to motor

// Aiguille DixMille (Moteur N°3)
L2 = MOD &alt_fs_h ,100000 // L2 is a real variable
V0087 = L2 * 0.40 // 0.40 au lieu de 0.36 pour compenser erreur
V0090 = V0087 + &Ajust_DixMil // send angle value to motor
}

////REGLAGE du QNH

// Var 0100, name QNH_FSuipc, Link FSUIPC_OUT, Offset $0F48, Length 2 // QNH to
FSUIPC
Var 0100, name QNH_FSuipc, Link FSUIPC_OUT, Offset $0330, Length 2 // Pressure
window to FSUIPC
Var 0102, name Intermediaire4
Var 0103, name Intermediaire5
```

//Remarque la IOCARD_ANALOGIC, Device 0 est la MASTER entrée analogique N°1 (sur 5)

Var 0101, name POT_QNH, Link IOCARD_ANALOGIC, Device 0, Input 1, PosL 0, PosC 127, PosR 255 // Input potentiomètre de 10kohm 10 tours QNH//

```
{  
    V0102 = &POT_QNH * 0.26876 // ax avec a=0.26876  
    V0102 = -1 * V0102  
    V0103 = V0102 + 1033.115 // y=ax+b avec b=1033.115  
    &QNH_FSuipc = V0103 * 16 // send setting to FSUIPC (16=mise a l'echelle pour  
fsuipc)  
}
```

////REGLAGE du QNH

////////// ALTIMETER //

ADF

```
////////// ADF INDICATOR //////////////////////////////////////
```

```
// Motor drives the Rotation of needle (not dial) clockwise =increase angle
```

```
Var 0110, Link USB_STEPPER, Device 2, Output 2, PosL 5, PosC 0, PosR 4, Type H
```

```
Var 0112, name Ajst_Centr_AD
```

```
Var 0113, name Angle_ValueAD
```

```
Var 0114, name Interadf1
```

```
Var 0115, name Interadf2
```

```
// Var 0111, Link FSUIPC_IN, Offset $02D8, Length 2 // FSimulator ADF2
```

```
Var 0111, Link FSUIPC_IN, Offset $0C6A, Length 2 // FSimulator ADF1
```

```
{
```

```
    v0114 = V0111 * 360
```

```
    v0115 = V0114 / 65536 // c'est le gisement :l'angle entre axe avion et la direction vers  
la balise
```

```
    &Ajst_Centr_AD = 100 // gauge center (ie : 8500 = needle indicates 85 deg angle  
instead of 0)
```

```
// le 100 pour moi c'est 1 graduation sur mon ADF l'aiguille indiquait 1 graduation vers la  
droite au lieu de zero
```

```
    &Ajst_Centr_AD = 36000 - &Ajst_Centr_AD
```

```
    &Angle_ValueAD = V0115
```

```
    &Angle_ValueAD = &Angle_ValueAD * 100
```

```
    V0110 = &Angle_ValueAD + &Ajst_Centr_AD // send degrees to stepper motor
```

```
}
```

```
////////// ADF INDICATOR //////////////////////////////////////
```

VOR

//////////////////////////////// VOR servo N°1 de la carte USB Servos N°2////////////////////////////////

Var 0120, name A12

Var 0121, name vor_needle

Var 0122, name servo7, Link USB_SERVOS, Device 4, Output 1, PosL 460, PosC 621, PosR 778

Var 0123, name vor_fsnedle, Link FSUIPC_INOUT, Offset \$0C48, Length 1 // NAV1

//Var 0123, name vor_fsnedle, Link FSUIPC_INOUT, Offset \$0C59, Length 1 // NAV2

{

 IF &vor_fsnedle > 127

 {

 &vor_needle = &vor_fsnedle - 257 // correction suite erreur dans fsuipc
retrancher 257 pour avoir les -127

 }

 ELSE

 {

 &vor_needle = &vor_fsnedle

 }

// Equation $y = 1.25 x + 621$ avec -127 left max et + 127 right max

 &A12 = 1.25 * &vor_needle

 &A12 = &A12 + 621

 IF &A12 > 778

 {

 &A12 = 778

 }

 IF &A12 < 460

 {

 &A12 = 460

 }

 &servo7 = &A12 // Send to servo

}

//partie reglage Bouton OBS sur master Input analogic N°2

//Var 0124, name VOROBS_FSuipc, Link FSUIPC_OUT, Offset \$0C5E, Length 2 // send
obs angle to FSUIPC o ->359 deg NAV2

Var 0124, name VOROBS_FSuipc, Link FSUIPC_OUT, Offset \$0C4E, Length 2 // send obs
angle to FSUIPC o ->359 deg NAV1

Var 0125, name Intermediaire6

Var 0126, name Intermediaire7

```
Var 0128, name Count
Var 0129, name Intermediaire8
Var 0140, name Stockpot
```

```
//Remarque la IOCARD_ANALOGIC, Device 0 est la MASTER entrée analogique N°2
```

```
Var 0127, name POT_VOR, Link IOCARD_ANALOGIC, Device 0, Input 2, PosL 0, PosC
127, PosR 255 // Input potentiomètre de 10kohm 10 tours VOR OBS//
{
```

```
    V0125 = V0127 * 3 // ax avec a=3//
    V0126 = V0125 - 18 // y=ax+b avec b=-18
    &VOROBS_FSuipc = V0126 // send setting to FSUIPC
```

```
}
```

```
//Partie Fleches UP DOWN STOP not active(Centre)
```

```
Var 0131, name LedStop, Link IOCARD_OUT, Output 12 // output 12 is led Stop
Var 0132, name LedTo, Link IOCARD_OUT, Output 13 // output 13 is led To (up)
Var 0133, name LedFrom, Link IOCARD_OUT, Output 11 // output 11 is led From (down)
```

```
// Var 0130, name LedFlag_FSuipc, Link FSUIPC_INOUT, Offset $0C5B, Length 1 //
receive led state from fsuipc not active NAV2
```

```
Var 0130, name LedFlag_FSuipc, Link FSUIPC_INOUT, Offset $0C4B, Length 1 // receive
led state from fsuipc not active NAV1
```

```
{
    IF &LedFlag_FSuipc = 0 // Stop non active
    {
        V0131 = 1
    }
    ELSE
    {
        V0131 = 0
    }
}
```

```
IF &LedFlag_FSuipc = 1 // To (Fleche Up)
{
    V0132 = 1
}
ELSE
{
    V0132 = 0
}
}
```

```
IF &LedFlag_FSuipc = 2 // From (Fleche Down)
{
```


TACHYMETRE

//////////////////////////////////// Tachymetre

Var 0280, name FS_RPM, Link FSUIPC_IN, Offset \$0898, Length 2, Type 1 // RPM from FSUIPC

```
{
    V0283 = &FS_RPM * &FS_RPM_Scaler
    V0284 = V0283 / 65536 // divided by 65536 to have RPM
//    V0285 = V0284 / 100 // divided by 100 to have hundreds RPM
```

// limites de la course du servo

```
IF V0284    >= 3500 // limite sup
{
    V0284 = 3500 // graduation 35
}
```

```
IF V0284    <= 300 // limite inf
{
    V0284 = 300 // graduation 3
}
```

```
V0286 = V0284 * 0.12 // conversion des RPM en position servo --> ax
V0287 = V0286 + 222 // conversion des RPM en position servo --> +b avec b=222
&servo_tachy = V0287 // Sortie cde servo
```

}

Var 0281, name FS_RPM_Scaler, Link FSUIPC_IN, Offset \$08C8, Length 2, Type 1 // Scaler from FSUIPC

Var 0282, name servo_tachy, Link USB_SERVOS, Device 4, Output 2, PosL 250, PosC 455, PosR 660

Var 0283

Var 0284

Var 0285

Var 0286

Var 0287

//////////////////////////////////// Tachymetre

Pour voir la séquence de synchronisation au lancement de sioc et de FSX :

<http://www.youtube.com/watch?v=iYywdp2NAzs>

C'est la fin de ce tuto , j'espère qu'il aura été utile à ceux qui veulent se lancer dans l'interfaçage des instruments.

Bons vols

Yveca

