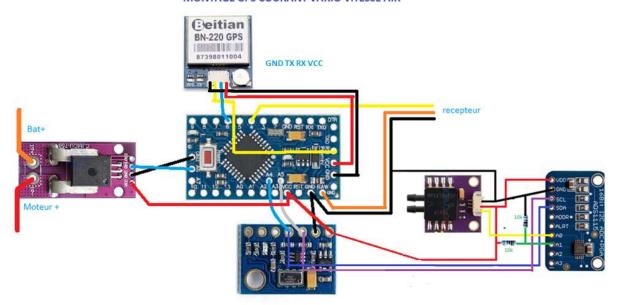
Table des matières

| PΙ | RESE | NTATION | 2 |
|----|------|---|---|
| | 1. | Le matériel : | 2 |
| | | Les logiciels : | |
| | | Programmation et mise en service des capteurs | |
| | | Programmation de la carte | |
| | | Amélioration de la précision Chapitre 6.1 et 6.5 dans oXs_config_advanced : | |
| | | Programmation du GPS | |
| | | Programmation dans openTX | |
| | | Activation des Log et récupération des valeurs du gps dans Google Hearth | |

OpenXsensor

MONTAGE GPS COURANT VARIO VITESSE AIR



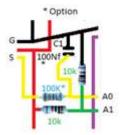


Fig1.

PRESENTATION

Création d'une carte d'acquisition de données provenant de capteurs low cost :

1. Le matériel:

Une carte arduino pro mini 5v 328

Un capteur de courant ACS 758 100A

Un capteur de pression barométrique MS5611

Un GPS BEITIAN BN-220

Un capteur de pression MPVX7002

Un module ADS1115

Un FTDI to USB

Des rallonges servo ou des fils électriques de faible section.

Gaine Thermo-rétractable de 1 mm et 3.5cm.

Un fer à souder

Un Ohmmètre

Un Ampèremètre (optionnel)

2 Résitances 10 kOhms

2. Les logiciels:

IDE Arduino à télécharger sur le site : https://www.arduino.cc/en/main/software

U-center: https://www.u-blox.com/en/product/u-center

Télécharger les applications et lancer les installations :

Pour IDE Arduino laisser le logiciel faire ne cherchez pas à l'installer ailleurs que sur le chemin d'accès préconisé ou installez le en mode portatif sur clé usb ou encore sur un autre pc si les transferts ne se font pas avec succès. Les problèmes de transfert de données proviennent bien souvent des autorisations d'accès (mode admin) sur tel ou tel répertoire ou du parefeu.

1 télécharger openXsensor : https://github.com/openXsensor/openXsensor

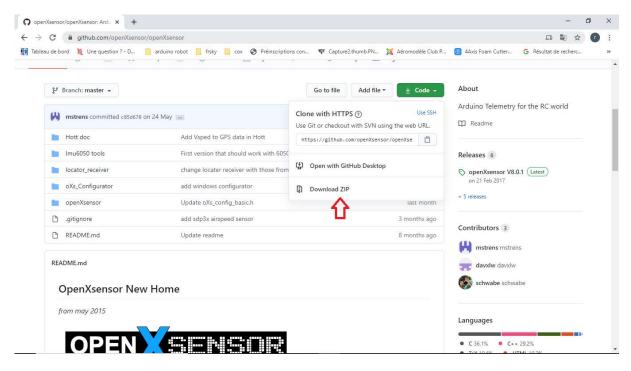
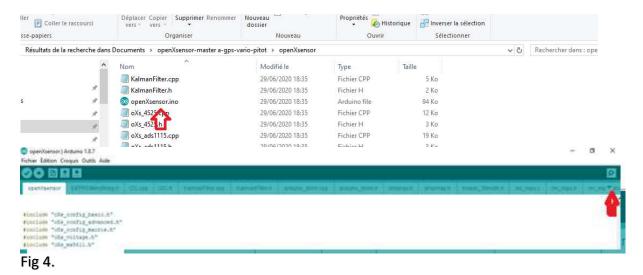


Fig 2.

Déziper les fichiers dans votre répertoire Mes documents et renommer le répertoire principal en fonction de votre configuration par exemple : openxsensor-master-gps-vario-a (vous aurez une configuration pour chaque montage et moins de risque d'erreur par la suite), dans ce répertoire, ouvrez le répertoire openxsensor et double-cliquez sur le fichier openxsensor.ino



Vous êtes maintenant dans l'IDE arduino nous allons paramétrer deux onglets oXs.config_basic et oXs.config_Advanced. Pour sélectionner l'onglet config basic cliquez sur la flèche à droite (fig 4) des onglets qui ouvre un menu déroulant fig(5)

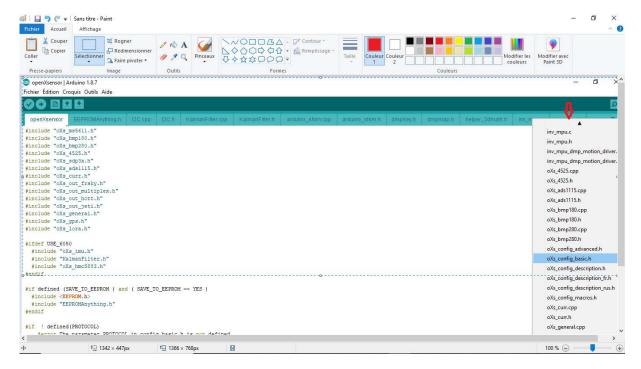


Fig 5.

- 3. Programmation et mise en service des capteurs
 - Sélectionnez oXs_config_basic.h double dic.(fig. 4 et 5)

Chapitre 1 : Sélectionnez le type de protocol Jeti, hott etc et copiez le à la suite de define PROTOCOL dans notre cas FRSKY SPORT

```
// ----- 1 - Telemetry protocol ------
```

#define PROTOCOL FRSKY_SPORT // select between FRSKY_SPORT , FRSKY_HUB , FRSKY_SPORT_HUB , MULTIPLEX , HOTT, JETI

Chapitre 2 : décommenter la source A3 qui prendra la valeur de d'ADS_VOLT_1 (valeur optimiser du capteur MPVX7002 par l'ADS 1115)

Décommenter la source T1 auquel sera affecté la valeur PPM , Valeur qui déterminera le passage du vario normal MS5611 vers un le vario compensé (mixage entre le capteur MPVX7002 et le MS5611) par son signe positif ou négatif et qui ajustera la finesse du MS5611 ou la compensation par le MPVX7002 par sa valeur absolue au travers d'un inter et d'un slider.

// ----- 2 - Data to transmit ----- uncomment the lines below if you want to transmit some extra oXs measurements into some telemetry fields

```
// **** 2.1 - Frsky data *****
#define A3 SOURCE ADS VOLT 1 // select between VOLT 1, VOLT 2, VOLT 3,
VOLT 4, VOLT 5, VOLT 6, ADS VOLT 1, ADS VOLT 2, ADS VOLT 3, ADS VOLT 4
#define T1 SOURCE
                     PPM
                                  // select between TEST 1, TEST 2, TEST 3,
GLIDER_RATIO, SECONDS_SINCE_TO, AVERAGE_VSPEED_SINCE_TO, SENSITIVITY, PPM,
VOLT_1, VOLT_2, VOLT_3, VOLT_4, VOLT_5, VOLT_6, ADS_VOLT_1, ADS_VOLT_2,
ADS VOLT 3, ADS VOLT 4
#define T2 SOURCE
                    SENSITIVITY
                                         // select between TEST 1, TEST 2, TEST 3, ,
GLIDER_RATIO, SECONDS_SINCE_TO, AVERAGE_VSPEED_SINCE_TO, SENSITIVITY, PPM,
VOLT_1, VOLT_2, VOLT_3, VOLT_4, VOLT_5, VOLT_6, ADS_VOLT_1, ADS_VOLT_2,
ADS VOLT 3, ADS VOLT 4
Chapitre 4 : Sélectionnez le type de BARO utilisé dans notre cas le MS5611 si ce n'est pas
déjà le cas.
Source de la vitesse proviendra du MS5611 ou du Vario compensé par le signe positif
(ms5611) ou négatif (MPXV7002) du signal PPM a la valeur VSPEED SOURCE.
La vitesse AIR viendra du MPXV7002
// ----- 4 - Vario settings -----
// **** 4.1 - Connecting 1 or 2 barometric sensor(s) *****
#define FIRST_BARO_SENSOR_USE MS5611 // select between NO_BARO , MS5611,
GY86, BMP085, BMP180, GY87, BMP280
// **** 4.2 - Type of Vspeed to transmit *****
#define VSPEED SOURCE PPM SELECTION // select between FIRST_BARO,
BARO AND IMU, SECOND BARO, AVERAGE FIRST SECOND, AIRSPEED COMPENSATED or
PPM SELECTION
#define AIRSPEED SENSOR USE MPXV7002 // select between NO AIRSPEED , MS4525
, MPXV7002, SDP3X
```

Chapitre 6.5: mettez YES pour activer le capteur de courant

```
// **** 6.5 - Current parameters *****
                                                             see also
oXs config advanced.h for additionnal parameters if you want to measure current with a
current sensor
#define ARDUINO MEASURES A CURRENT YES
                                                                  // select between
YES, NO
Chapitre 6.6
Il faut déclarer l'ADS1115, sont rôle améliorer la précision de la valeur du MPXV7002
// **** 6.6 - Ads1115 parameters *****
                                                              see
oXs_config_advanced.h for additional parameters when AN_ADS1115_IS_CONNECTED is YES
#define AN_ADS1115_IS_CONNECTED YES // select between YES , NO
Chapitre 9: mettez YES pour activer le GPS
// ----- 9 - GPS -----
                                                   see oXs config advanced.h for
additionnal parameters (normally no need to change them)
                                            // select between YES , NO
#define A GPS IS CONNECTED YES
Les valeurs des chapitres suivant doivent être à NO : Chapitre 6.2 7 8 10 11 12
Chapitre 6.3 (valeur 0) dans le cas contraire le pin (A6 dans la version à télécharger dans
oXs advanced) voltage est réservé pour la tension des éléments
#define NUMBEROFCELLS 0

    Sélectionnez oXs_advanced.h double dic. (fig 4 et 5)

Définir le pin de communication du Smartport pin 4 dans notre cas :
// ***** 1.1 - Pin connected to Rx ******
#define PIN_SERIALTX 4
                                 // The pin which transmits the serial data to the
telemetry receiver, Usually pin 4 (otherwise pin 2)
Chapitre 4.4
Le potard sur la radio affecté au MS5611 devra varié entre +10 et + 40 pourcent, la valeur
positive ou négative sera envoyé par un inter. L'échelle de sensibilité sera comprise entre 20
```

et 200. Décommenter les lignes suivantes :

// **** 4.4 - Sensitivity adjusted from the TX *****

```
#define SENSITIVITY_MIN_AT_PPM 10 // sensitivity will be changed by OXS only when PPM signal is between the specified range enlarged by -5 / +5
```

```
#define SENSITIVITY_MAX_AT_PPM 40

#define SENSITIVITY_PPM_MIN 20 // common value for vario is 20

#define SENSITIVITY PPM MAX 100 // common value for vario is 100
```

Chapitre 4.6 valeur de l'inter comprise entre min 10 et 90 poucent, le signe de la valeur permettra de basculer sur le MS5611 (positif) ou le MPXV7002 (négatif)

```
// ***** 4.6 - Vertical speeds calculations based on PPM ****

#define VARIO_PRIMARY FIRST_BARO // select between FIRST_BARO,

SECOND_BARO , AVERAGE_FIRST_SECOND, AIRSPEED_COMPENSATED , BARO_AND_IMU

#define VARIO_SECONDARY AIRSPEED_COMPENSATED // select between

FIRST_BARO, SECOND_BARO , AVERAGE_FIRST_SECOND, AIRSPEED_COMPENSATED ,

BARO_AND_IMU

#define SWITCH_VARIO_MIN_AT_PPM 10

#define SWITCH_VARIO_MAX AT_PPM 90
```

5 Airspeed ici la plage du potard va être comprise entre -60 et -90 le signe déterminera l'activation de la compensation par le MPXV7002, le reset du MPXV7002 sera effectué quand le switch 3 positions sera sur la troisième position (pour résumé : switch en bas plage du potard 10 40, switch au milieu plage du potard -60 -90, switch en haut valeur renvoyée 100 quand le manche dir est a gauche -100 et l'inter écolage en haut.

```
// ----- 5 - Airspeed settings ------
```

//#define AIRSPEED_AT_SEA_LEVEL_AND_15C // if this line is commented, airspeed is calculated using baro pressure and temperature (so being "true" airspeed instead of normalised airspeed)

//#define AIRSPEED_IN_KMH // uncomment this line if airspeed has to be in km/h instead of knot/h (except some old versions, openTx expects knot/h)

#define AIRSPEED RESET AT PPM 100

```
#define COMPENSATION_MIN_AT_PPM 60

#define COMPENSATION_MAX_AT_PPM 90

#define COMPENSATION_PPM_MIN 80

#define COMPENSATION_PPM_MAX 140
```

Décommenter #define REFERENCE_VOLTAGE 4970 et commenter les autres lignes avec des // dans le chapitre 6

```
// ----- 6 - Voltages & Current sensor settings ------
```

```
// **** 6.1 - Voltage Reference to measure voltages and current *****
```

//#define USE_INTERNAL_REFERENCE // uncomment this line if you use 1.1 volt internal reference instead of Vcc (voltage divider mst be used to reduce voltages to 1.1 volt max)

//#define USE_EXTERNAL_REFERENCE // uncomment this line if you use an external reference instead of Vcc

#define REFERENCE_VOLTAGE 4970 // set value in milliVolt; if commented, oXs will use or 1100 (if internal ref is used) or 5000 (if internal ref is not used)

Ces valeurs sont provisoires : voir 5. Amélioration de la précision Chapitre 6.1 et 6.5 dans oXs_config_advanced :

Chapitre 6.5 capteur de courant

nous allons paramétrer le capteur de courant ACS 758 100A il faudra remplir le numéro du port analogique dans notre cas le port 6. Après lecture du datasheet la tension a 0 ampère est de 2500 millivolt soit 2.5 v, et nous avons une tension de 20mv/A

```
// ***** 6.5 - Current parameters *****

#define PIN_CURRENTSENSOR 6 // Arduino pin used to measure the voltage provided by a current sensor

#define MVOLT_AT_ZERO_AMP 2500 // in millivolt

#define MVOLT_PER_AMP 20 // in millivolt per Amp

#define RESISTOR_TO_GROUND_FOR_CURRENT 0 // put as comment or set to 0 if no divider is used (e.g. 19.8 for 1.1 internal ref)
```

#define RESISTOR_TO_CURRENT_SENSOR 0 // put as comment or set to 0 if no divider is used (e.g 39 for 1.1 internal ref)



🔨 Ces valeurs sont provisoires : voir 5. Amélioration de la précision Chapitre 6.1 et 6.5 dans oXs config advanced:

Chapitre 6.6 paramétrage de l'Ads1115, la comparaison est faite entre l'entrée analogique A0 et A1 (équipé d'un pont diviseur), ADS FULL SCALE à MV2048 2048 millivolts le pont diviseur composé de deux résistances 10 kOhms divise la tension par 2 notre capteur est alimenté en 4.75v environs soit 4750 millivolts. La vitesse de rafraichissement est de 2ms (rapide) par L' I2C (bus intégré qui permet d'avoir le vario MS5611 et le MPVX7002 sur les mêmes broche A4 A5), ADS AVERAGING est défini a 10 (la valeur retenue correspond à la moyenne de 10 valeurs, ce qui permet de lisser les données, c'est la solution retenue dans oXs pour améliorer la pertinence des résultats). Ensuite la valeur est affectée à VOLT_1.

// **** 6.6 - Ads1115 parameters *****

#define ADS_MEASURE A0_TO_A1 , ADS_OFF , ADS_OFF , ADS_OFF // select 4 values between A0 TO A1, A0 TO A3, A1 TO A3, A2 TO A3, A0 TO GND, A1 TO GND, A2_TO_GND, A3_TO_GND, ADS_OFF

#define ADS_FULL_SCALE_VOLT_MV2048, MV2048, MV2048, MV2048 // select between MV6144 MV4096 MV2048 MV1024 MV512 MV256

#define ADS OFFSET 0, 0, 0, 0 // must be an integer (positive or negative)

#define ADS SCALE 1, 1, 1, 1 // can be a float

#define ADS_RATE_MS2, MS2, MS2, MS2 // select between MS137, MS69, MS35, MS18, MS9, MS5, MS3, MS2

#define ADS_AVERAGING_ON 10, 20, 30, 50 // number of values used for averaging (must be between 1 and 254)

//#define ADS_CURRENT_BASED_ON ADS_VOLT_1 // uncomment if current, and comsumption have to be calculated based on one of ADS voltage measurement; select then the voltage to be used between ADS_VOLT_1, ADS_VOLT_2, ADS_VOLT_3, ADS_VOLT_4

#define ADS_AIRSPEED_BASED_ON ADS_VOLT_1 // uncomment if airspeed (and dte) have to be calculated based on one of ADS voltage measurement; select then the voltage to be used between ADS_VOLT_1, ADS_VOLT_2, ADS_VOLT_3, ADS_VOLT_4

Il faut enlever les commentaire // de la ligne #define GPS_SPEED_3D, le taux de rafraichissement est mis sur 5 dans notre cas , c'est amplement suffisant, le gps va chauffer

//#define GPS_SPEED_IN_KMH // uncomment this line if GPS speed has to be sent in km/h instead of knot/h (only for Frsky protocol)

#define GPS_SPEED_3D // uncomment this line if GPS speed has to be the 3d speed instead of the 2d speed (note: 3d is probably less accurate - to test)

#define GPS_REFRESH_RATE 5 // rate at which GPS sent new data; select between 1, 5 or 10 (Hz). Default = 5 Hz; Ublox NEO6 does not support 10 hz.

Une fois ces modifications terminées il faut téléverser le programme dans l'arduino.

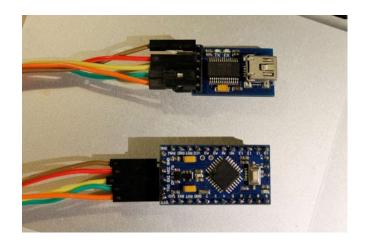
L'ensemble des modifications précédentes peuvent aussi être réalisées en ouvrant les fichiers oXs_config_basic et oXs_ advanced qui se trouvent dans le répertoire openxsensor dans un éditeur de texte type notepad++ ou word, il faudra être vigilant lors de l'enregistrement et vérifier que l'extension du fichier soit la bonne et repasser par l'IDE ARDUINO pour programmer la carte

- 4. Programmation de la carte
 - o Téléversement du programme dans l'arduino

ATTENTION LE GPS DOIT ETRE DEBRANCHE PHISIQUEMENT SINON LE TELECHARGEMENT NE SE FERA PAS

La pro mini ne possède pas de port de communication intégré, il faut donc utiliser un câble FTDI pour programmer l'arduino

Fig. 6



Connections FTDI USB-to-TTL 6pins to Arduino Mini Pro:

- 1. FTDI USB-to-TTL: TX (yellow) -> Arduino Mini Pro RX
- 2. FTDI USB-to-TTL: RX (green) -> Arduino Mini Pro TX
- 3. FTDI USB-to-TTL: GND (brown) -> Arduino Mini Pro GND
- 4. FTDI USB-to-TTL: +5Vcc (red) -> Arduino Mini Pro VCC
- 5. FTDI USB-to-TTL: DTR (orange) -> Arduino Mini Pro Reset
- 6. FTDI USB-to-TTL: CTS (non connected)

Une fois raccordé, vérifiez que vous avez sélectionné la bonne carte, vérifiez le numéro du port com affecté à votre FTDI dans votre gestionnaire de périphérique (dans la barre des taches tapez gestionnaire de périphérique)

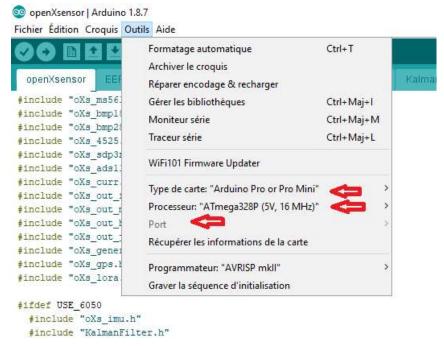


Fig.7

Pour téléverser :

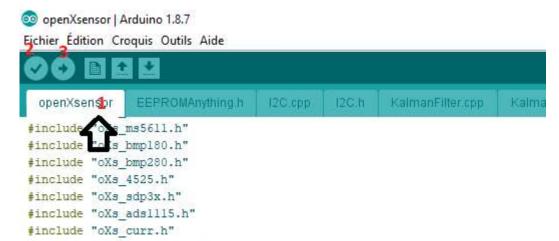


Fig.8

1 cliquer sur l'onglet openxsensor (si vous êtes sur un autre onglet, le téléversement peux générer un message d'erreur)

2 cliquer sur la vérification

3 sur téléverser

Si l'opération s'est bien déroulée vous aurez (2) compilation terminée puis (3) téléversement terminé :

Fig.9

Pour U-center rien de particulier concernant l'installation.

5. Amélioration de la précision Chapitre 6.1 et 6.5 dans oXs config advanced :

Comme indiqué précédemment les valeurs de tension de la carte sont provisoires, après la première programmation et mise en service, il faudra mesurer la tension entre le pin VCC et ground sur l'arduino, convertir la valeur obtenue en MVolt (Ex. :4.85v =4850) et remplacer la

valeur dans #define REFERENCE_VOLTAGE 4850 (4.85v =4850) par la nouvelle valeur obtenue et téléverser à nouveau. La précision sera grandement améliorée!

Il en est de même pour le capteur de courant après mise à jour de la tension de référence il faudra mesurer la tension du signal sur le capteur (OUT1 ou A6 sur l'arduino et le GND) sans aucun raccordement de batterie pour obtenir la tension a 0 Ampère, elle peut varier d'un capteur à l'autre. Si l'on a un ampère (optionnel) mètre il est possible d'affiner la tension par ampère consommé en prenant la tension aux bornes du capteur (signal et Ground) a 5 10 15 20 ampères de consommation par exemple, une fois les valeurs mesurées on remplace les valeurs millivolt par ampère(optionnel) et millivolt à 0 Ampère par des valeurs plus précises. Et téléverser à nouveau.

6. Programmation du GPS

Le GPS fonctionne généralement sans faire aucune modification mais la mise en route du GPS BEITIAN BN 220 peut être améliorée, message d'erreur et surtout prise en compte des satellites GALILEO. Pour cela ouvrez le logiciel U-CENTER, connectez votre module GPS sur votre câble FTDI.

Pin Description:



FTDI GPS

VCC VCC

TX RX

RX TX

GND GND

| | PIN | PIN Name | I/O | Description | |
|--|-----|----------|-----|--|--|
| | 1 | GND | G | Ground | |
| | 2 | TX | 0 | Serial Data Output. | |
| | 3 | RX | I | Serial Data Input. | |
| | 4 | VCC | I | DC 3.0V - 5.5V supply input, Typical: 5.0V | |

Fig.10

Dans le logiciel UCENTER

Cliquez sur l'onglet en haut : View

Puis Configuration View

Puis Menu de gauche GNSS Config

Cochez sur Galiléo ENABLE

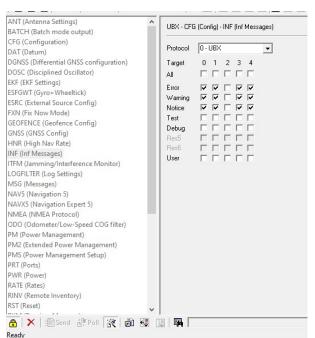
Décochez Bédou CONFIGURE

ET ENABLE

Ouvrir le cadenas cliquez sur

SEND fermez le Cadenas cliquez sur SEND

Fig.11



Toujours dans le menu de gauche cliquez sur INF (inf Message) puis cochez error warnig notice sur la cible 0 1 3 4.

UEX - CFG (Config) - GNSS (GNSS Config)

Configure

V

V

V

V

V

V

N 4

P 0

8

0

GNSS

GPS

SBAS

Galleo

SeiDou

IMES

gzss.

IRNSS

GLONASS

Number of channels available

Number of channels to use

ID

0

3

4

5

6 0 X

Signals

₩ E1

₩ B1

₹ L1C/A

₩ L10F

₹ L1C/A T L1SAIF

₩ L1C/A

₩ L1C/A

16

16

14

☐ Auto set

Ouvrir le cadenas cliquez sur

Configure - GNSS Configuration

ANT (Antenna Settings)

DGNSS (Differential GNSS configur

DOSC (Disciplined Oscillator)

ESRC (External Source Config)

GEOFENCE (Geofence Config) GNSS (GNSS Canfig)

ITFM (/amming/Interference Moni-

商 × 圖Send 등 Poll 등 回 ■

DYNSEED (Dynamic Seed)

CFG (Configuration)

EKF (EKF Settings) ESFGWT (Gyro+Wheeltick)

FIXSEED (Fixed Seed)

EXN (Fix Now Mode)

HNR (High Nav Rate)

LOGFILTER (Log Settings)

INF (Inf Messages)

MSG (Messages)

NAV5 (Navigation 5)

DAT (Datum)

SEND fermez le Cadenas cliquez sur SEND

Fig.12

Dans PRT (port) modifiez le Target en UART1

Le protocol in 0+1+2 UBX

Le protocol out 0+1 UBX NMEA

Le BAUDRATE a 9600

Ouvrir le cadenas cliquez sur

SEND fermez le Cadenas cliquez sur SEND

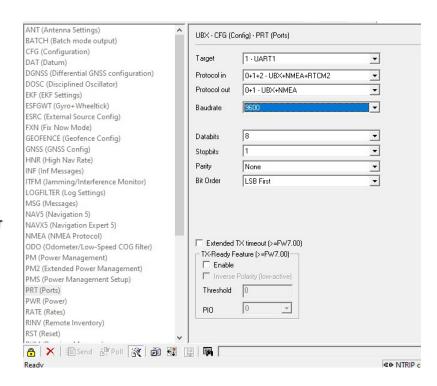
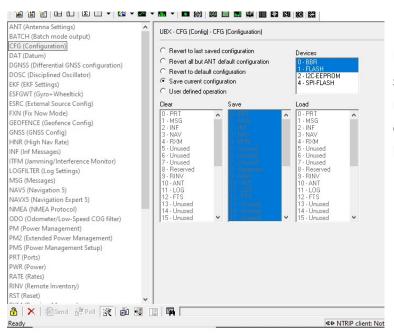


Fig.13



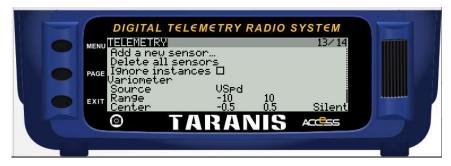
Dans CFG (configuration) cochez save current configuration si ce n'est pas déjà le cas puis Ouvrir le cadenas cliquez sur SEND fermez le Cadenas cliquez sur SEND.

Fig.14

7. Programmation dans openTX

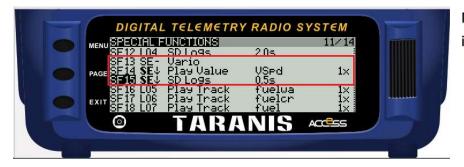
Pour la jauge à carburant celle de LAPINFOU est très bien vous trouverez toutes les explications ici : jauge a carburant lapin fou

Le vario:



Mettre la source du Vario sur Vspd

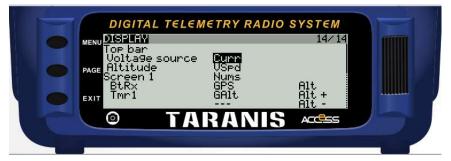
Fig.15



Mettre le vario sur un inter ici SE

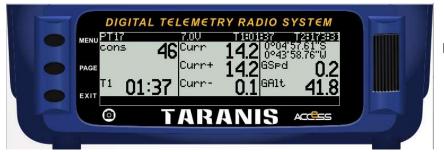
Fig.16

L'affichage des données :



Une disposition possible

Fig.17



Le Résultat

Fig.18

Concernant le vario compensé voici les mixages à créer pour la voie 9

L'inter SA permet de déterminer les plages du Slider S2 donc le passage Vario au Vario compensé ainsi que la sensibilité (+) ou la compensation (-)

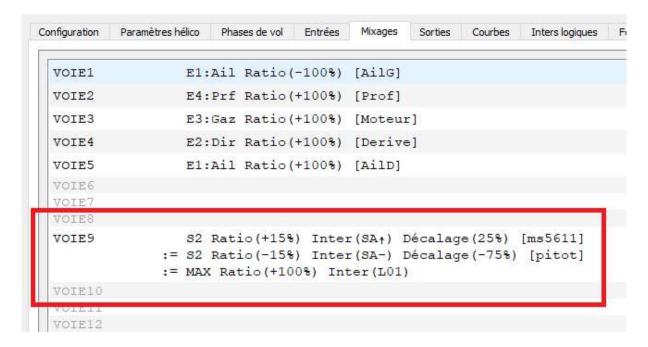


Fig.18

Inter logique quand la direction et a -100 (à gauche) et l'inter écolage en haut le MPXV7002 est remis à zéro



Fig.19

Copier le fichier LUA oxsppm dans le répertoire MIXES puis activez les fonctions spéciales

| FS8 | ON ▼ | Exécuter script ▼ | oxsppm | |
|-----|------|-------------------|--------|--|
| | | | | |

Fig.20

8. Activation des Log et récupération des valeurs du gps dans Google Hearth

Sur la Radio:

Ajouter dans les fonctions spéciales le démarrage des logs, dans mon cas, il démarre lorsque je démarre le moteur (inter logique LO1 sur la voie des gaz qui permet de couper le moteur quand SF↑) ,un enregistrement toute les demi secondes.

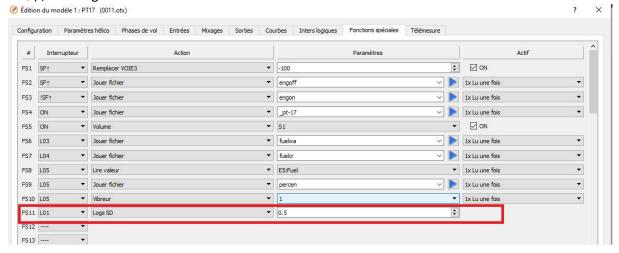


Fig. 21
Sur l'onglet télémesure pensez à activer les logs sur le capteur

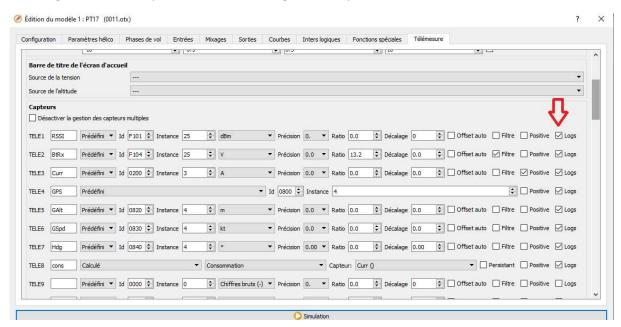


Fig. 22

Sur votre ordinateur:

Copier les fichiers logs de votre Taranis qui se trouve dans le répertoire log sur la carte SD dans mes documents \LOG par exemple, puis ouvrez openTX, cliquez fichier et ouvrir un log.

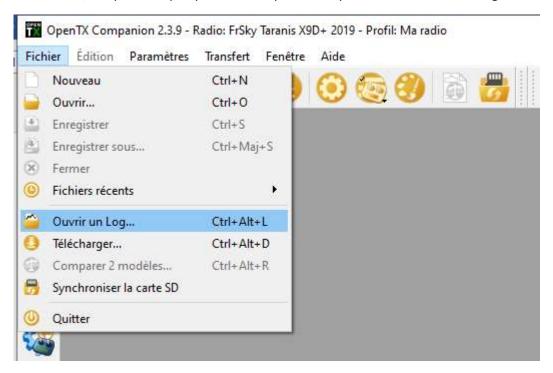


Fig.23

Cliquez sur ouvrir un log et sélectionnez votre fichier .csv

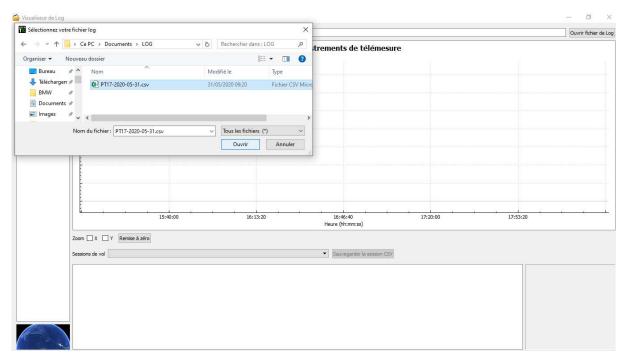


Fig.24

Cliquez sur la terre (c'est un bouton en fait)

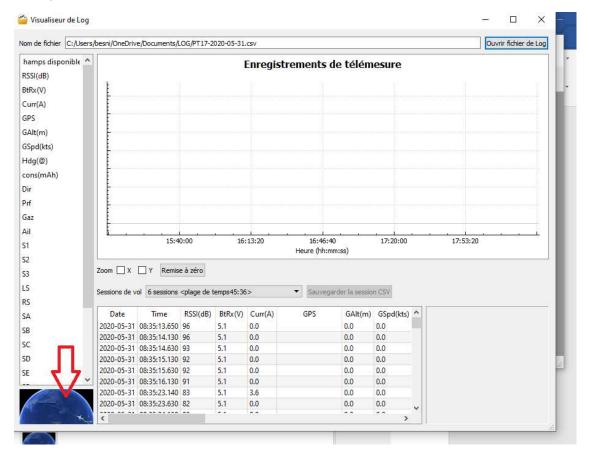


Fig. 25

Google Earth va s'ouvrir, le tracé du vol va apparaître! bon là ça ne ressemble à rien!

