

# **oXs RP2040**

***Nouvelle version  
openXsensor  
pour un RP2040-Zero***

**Manuel Utilisateur oXs RP2040**

# Table des matières

1	CE DOCUMENT.....	3
1.1	Versions.....	3
1.2	Remerciement.....	3
1.3	Avertissement.....	3
1.4	Contenu.....	3
1.5	Protocoles supportés.....	3
1.6	Fonctions du module oXs RP2040.....	4
2	Présentation d'oXs RP2040.....	5
2.1	Vue d'ensemble.....	5
2.2	Spécifications du module oXs RP2040.....	5
3	Schéma du module oXs RP2040 Full.....	6
4	Réalisation du module oXs RP2040 Full.....	7
4.1	Circuit imprimé.....	7
4.2	Carte WaveShare RP2040-Zero.....	7
4.3	Téléchargement du Firmware dans le RP2040.....	7
4.4	Chargement du firmware dans le RP2040.....	7
5	Configuration.....	7
5.1	Connexion au PC.....	7
5.2	Configuration.....	8
5.2.1	Visualiser la configuration du module.....	8
5.2.2	Détails de la configuration.....	9
5.2.3	Définition du protocole de télémétrie.....	11
Remarque : les broches entre () signifient qu'elles sont facultatives.....		11
6	Câblage.....	12
6.1	Alimentation du module.....	12
6.2	Limite du RP2040.....	12
7	Les protocoles.....	13
7.1	S-Port, Hott, M-Link, Ibus, Jeti Ex.....	13
7.2	Sbus2, Srxl2, Jeti ExBus, Fbus.....	13
7.2	ExpressLRS.....	14
8	Utilisation de deux récepteurs.....	14
9	Les capteurs.....	15
9.1	Le GPS.....	15
9.2	Sonde Pitot (capteur de pression différentielle).....	16
9.3	Le baromètre.....	16
9.4	Les convertisseurs analogiques ADS1115.....	17

# 1 CE DOCUMENT

## 1.1 Versions

Version du Manuel	Date	Raison de l'évolution
1.0	11/04/2023	Création

## 1.2 Remerciement

Ce projet est basé sur le travail de Michel Strens qui en a réalisé le code .

Merci à lui pour ce formidable travail ainsi que son écoute.

Le code est téléchargeable à l'adresse suivante :

[https://github.com/mstrens/oXs\\_on\\_RP2040](https://github.com/mstrens/oXs_on_RP2040).

## 1.3 Avertissement

Je ne suis aucunement responsable des dommages qui pourraient découler de la mauvaise utilisation ou d'un éventuel dysfonctionnement du module oXs RP2040, et/ou du logiciel associé.

Il appartient donc à l'utilisateur final d'en mesurer, d'en assumer les risques et de respecter la législation en vigueur selon le pays d'utilisation.

## 1.4 Contenu

Ce document décrit la réalisation d'un module de télémétrie **oXs RP2040** ainsi que son paramétrage.

## 1.5 Protocoles supportés

**oXs RP2040** est MULTI protocoles.

Il supporte les protocoles de télémétrie RC suivants :

ExpressLRS(CRSF), FRSKY (S-Port ou FBUS), Graupner (HOTT), Multiplex (M-LINK), FLYSKY (IBUS) , Futaba ((SBUS2), Spektrum (SRXL2), Jeti Ex, Jeti ExBus.

Ce projet peut être interfacé avec 1 ou 2 récepteurs selon le protocole.



## 1.6 Fonctions du module oXs RP2040

Ce projet est prévu pour générer :

- des données de télémétrie (par exemple, lorsqu'un contrôleur de vol n'est pas utilisé).
- sorties PWM pour des servos.
- signaux Sbus.

Pour la télémétrie, il peut fournir :

- jusqu'à 4 mesures de tensions analogiques (avec mise à l'échelle et décalage) (en option).
- une mesure RPM ; une mise à l'échelle (SCALE4) peut être utilisée pour prendre en compte par ex. le nombre de pâles (facultatif).
- l'altitude et la vitesse verticale lorsqu'il est connecté à un capteur de pression (en option)
- la vitesse anémométrique lorsqu'il est connecté à un capteur de pression différentielle (et un tube de Pitot) (facultatif).
- vitesse verticale compensée lorsqu'il est connecté à un baromètre + un capteur de pression différentielle.
- Pitch et Roll lorsqu'il est connecté à un capteur MP6050 (en option).
- Données GPS (longitude, latitude, vitesse, altitude,...) (en option) Remarque : la vitesse verticale est améliorée lorsque le capteur baro est combiné avec le capteur MP6050.

Il peut également fournir jusqu'à 16 canaux PWM RC à partir d'un CRSF/ELRS ou à partir de 1 ou 2 signaux Sbus (par exemple Frsky ou Jeti).

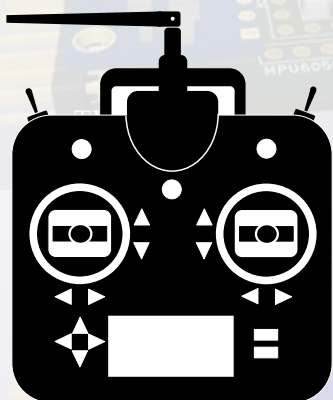
Il peut également fournir un signal SBUS (par exemple à partir de 1 ou 2 récepteurs ELRS).

Lorsqu'il est connecté à 2 récepteurs, les signaux PWM et Sbus générés seront émis à partir des derniers canaux Rc reçus. Cela fournit donc une sorte de redondance/diversité.

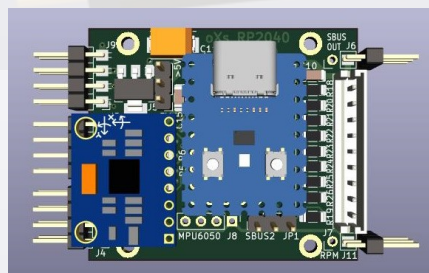
Chaque fonction (télémétrie/PWM/SBUS) peut être utilisée seule ou combinée avec les autres.

## 2 Présentation d'oXs RP2040

### 2.1 Vue d'ensemble



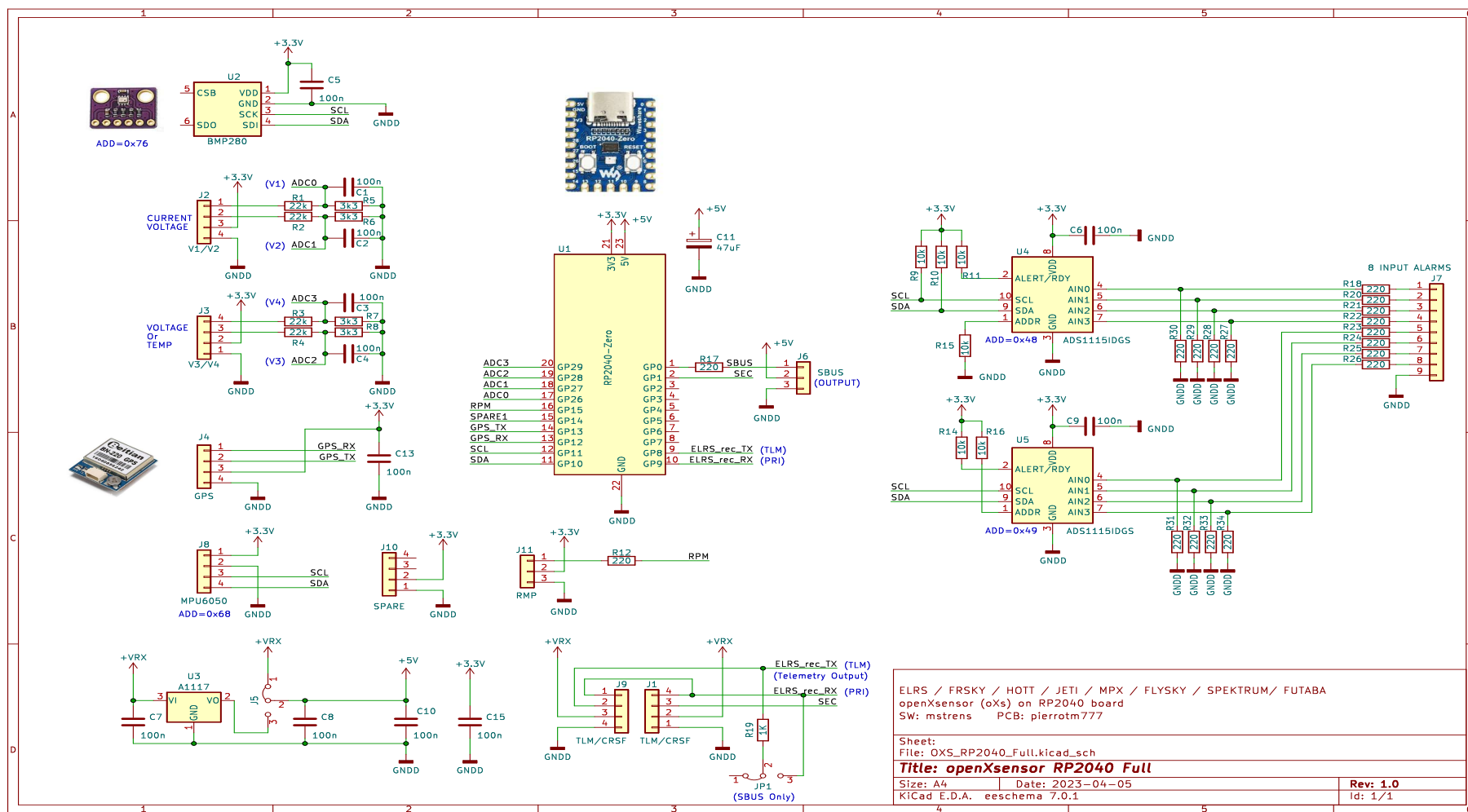
ELRS / FRSKY (sport + Fbus) / HOTT / JETI  
Ex/ MPX / FLYSKY /  
Futaba (SBUS2) /  
Spektrum (SRXL2)



### 2.2 Spécifications du module oXs RP2040

Spécification	Valeur / Caractéristique	Note
Alimentation oXs	+5V à +12V	Mettre le cavalier « tension récepteur » sur «5v» ou «>5v» selon la valeur de la tension fournie par le récepteur
Protocoles possibles	ELRS / FRSKY (sport + Fbus) / HOTT / JETI Ex/ MPX / FLYSKY / Futaba (SBUS2) / Spektrum (SRXL2)	
Port série USB		Pour configuration avancée à l'aide d'une application type « Terminal série »
Dimensions		

### 3 Schéma du module oXs RP2040 Full



## 4 Réalisation du module oXs RP2040 Full

### 4.1 Circuit imprimé

La conception du circuit imprimé du décodeur **oXs RP2040** a été réalisée sous KiCad.

Le [dossier de fabrication](#) est disponible sur GitHub.

### 4.2 Carte WaveShare RP2040-Zero

Le module de télémétrie utilise un Rasperry Pico de type RP2040-Zero.

#### RP2040-Zero de Waveshare



### 4.3 Téléchargement du Firmware dans le RP2040

Le chargement du Firmware (micro-logiciel embarqué) consiste à copier le contenu d'un fichier uf2 dans la mémoire du micro-contrôleur.

Le fichier **oXs.uf2** est à télécharger sur le dépôt [GitHub](#).

### 4.4 Chargement du firmware dans le RP2040

Le RP2040 doit être connecter à votre PC via son connecteur USB tout en maintenant le bouton BOOT.

Une fenêtre doit s'ouvrir. Il suffit de copier le fichier oXs.uf2 pour charger le firmware dans le RP2040.

## 5 Configuration

### 5.1 Connexion au PC

La configuration du module se fait au travers d'un terminal .

Sur Windows, j'utilise [Termite](#) de Compuphase .

A la connexion du RP2040, vous devriez voir apparaître un port COM qui faudra sélectionner.

## 5.2 Configuration du terminal Termit

Paramètres du port série & réglages du programme

Configuration du port série		Trame d'émission	Options
Port	COM1	<input type="radio"/> Suffixe : Aucun	<input type="checkbox"/> Toujours visible
Vitesse (b/s)	COM1	<input type="radio"/> Suffixe : CR	<input checked="" type="checkbox"/> Fermer par <Echap>
	COM5	<input type="radio"/> Suffixe : LF	<input checked="" type="checkbox"/> Saisie semi-auto
Bits données	8	<input checked="" type="radio"/> Suffixe : CR-LF	<input type="checkbox"/> Keep history
Bits stop	1	<input checked="" type="checkbox"/> Echo local	<input type="checkbox"/> Ferme le port si inactif
Parité	aucune	Trame de réception	Plug-ins
Ctrl de flux	DTR/DSR	Polling	<input type="checkbox"/> Auto Reply
Redirection	aucune	100 ms	<input type="checkbox"/> Function Keys
		Max. lines	<input type="checkbox"/> Hex View
			<input type="checkbox"/> Highlight
		Police	<input type="checkbox"/> Load File
		standard	
		<input type="checkbox"/> Retour à la ligne auto	
Langue de l'interface		Français (fr)	Annuler OK

Configurez le terminal, puis sélectionnez le port COM (ici le pour COM5).



## 5.3 Configuration

### 5.3.1 Visualiser la configuration du module

Utiliser la touche ENTREE .

Le module doit retourner le résultat suivant :

```
processing cmd
```

```
Cmd to execute:
```

```
Version = 2.2.12
```

```
Function                               Pin    Change entering XXX=yyy (yyy=255 to
disable)
Primary channels input      = 255 (PRI    = 5, 9, 21, 25)
Secondary channels input    = 255 (SEC    = 1, 13, 17, 29)
Telemetry . . . . . = 255 (TLM    = 0, 1, 2, ..., 29)
GPS Rx . . . . . = 255 (GPS_RX  = 0, 1, 2, ..., 29)
GPS Tx . . . . . = 255 (GPS_TX  = 0, 1, 2, ..., 29)
Sbus OUT . . . . . = 255 (SBUS_OUT= 0, 1, 2, ..., 29)
RPM . . . . . = 255 (RPM    = 0, 1, 2, ..., 29)
SDA (I2C sensors) . . . . = 255 (SDA    = 2, 6, 10, 14, 18, 22, 26)
SCL (I2C sensors) . . . . = 255 (SCL    = 3, 7, 11, 15, 19, 23, 27)
PWM Channels 1, 2, 3 ,4     = 255 255 255 255 (C1 / C16= 0, 1, 2, ..., 15)
PWM Channels 5, 6, 7 ,8     = 255 255 255 255
PWM Channels 9,10,11,12     = 255 255 255 255
PWM Channels 13,14,15,16    = 255 255 255 255
Voltage 1, 2, 3, 4         = 255 255 255 255 (V1 / V4 = 26, 27, 28, 29)
```

```
Protocol is Sport (Frsky)
```

```
CRSF baudrate = 420000
```

```
Voltage parameters:
```

```
    Scales : 1.000000 , 1.000000 , 1.000000 , 1.000000
```

```
    Offsets: 0.000000 , 0.000000 , 0.000000 , 0.000000
```

```
    No temperature sensors are connected on V3 and V4
```

```
RPM multiplier = 1.000000
```

```
Baro sensor is not detected
```

```
Acc/Gyro is not detected
```

```
Airspeed sensor is not detected
```

```
    No Vspeed compensation channel defined; oXs uses default settings
```

```
First analog to digital sensor is not detected
```

```
Second analog to digital sensor is not detected
```

```
Foreseen GPS type is Ublox (configured by oXs) :GPS is not (yet) detected
```

```
Failsafe type is HOLD
```

```
Config parameters are OK
```

```
Press ? + Enter to get help about the commands
```

```
>>
```

### 5.3.2 Détails de la configuration

**PRI** Entrée FBUS SBUS ou EXBUS ou SRXL2

**SEC** Deuxième Entrée FBUS SBUS ou EXBUS ou SRXL2

**TLM** Sortie Télémétrie S-PORT JETI M-LINK HOTT

**GPS\_RX** Connexion RX du GPS

**GPS\_TX** Connexion TX du GPS

**SBUS\_OUT** Sortie SBUS (résultat du meilleur signal entre PRI et SEC)

**RPM** Entrée tachymètre

**C1 à C16** Sortie PWM (servos)

**V1 à V4** Entrées analogiques (V1 dédié à la tension d'un accus, V2 au courant consommé, V3 et V4 à deux tensions supplémentaires ou deux températures)

Voici la configuration possible fonction du circuit imprimé réalisé :

```
Version = 2.2.12
Function                               Pin   Change entering XXX=yyy (yyy=255 to
disable)
Primary channels input                 =    9   (PRI      = 5, 9, 21, 25)
Secondary channels input               =    1   (SEC      = 1, 13, 17, 29)
Telemetry . . . . .                   =    8   (TLM      = 0, 1, 2, ..., 29)
GPS Rx . . . . .                      =   13   (GPS_RX   = 0, 1, 2, ..., 29)
GPS Tx . . . . .                      =   12   (GPS_TX   = 0, 1, 2, ..., 29)
Sbus OUT . . . . .                    =    0   (SBUS_OUT= 0, 1, 2, ..., 29)
RPM . . . . .                         =   15   (RPM      = 0, 1, 2, ..., 29)
SDA (I2C sensors) . . . . .           =   10   (SDA      = 2, 6, 10, 14, 18, 22, 26)
SCL (I2C sensors) . . . . .           =   11   (SCL      = 3, 7, 11, 15, 19, 23, 27)
PWM Channels 1, 2, 3 ,4                =    2    3    4    5 (C1 / C16= 0, 1, 2, ..., 15)
PWM Channels 5, 6, 7 ,8                =  255  255  255  255
PWM Channels 9,10,11,12                =  255  255  255  255
PWM Channels 13,14,15,16               =  255  255  255  255
Voltage 1, 2, 3, 4                     =  255  255  255  255 (V1 / V4 = 26, 27, 28, 29)
```

Dans ce premier tableau, on affecte une fonction à une patte du RP2040.

L'affectation des pattes ne doit bien sûr pas être modifiée car il y aurait incompatibilité avec le circuit imprimé.

Les pattes ayant la valeur 255 ne sont pas définies.

**PRI** sera toujours égal à la patte **9** (mais il peut être aussi égal à 5, 9, 21 ou 25).

**SEC** sera toujours égal à la patte **1** ( mais il peut aussi être égal à 1, 13, 17 ou 29).

etc...

Mais, en fonction du protocole choisi et des fonctions utilisées, certaines pattes resteront à **255** donc non définies.

Pour définir une patte, il suffit de taper dans le terminal **TLM=8** par exemple ou **TLM=255** pour la désaffecter.

Pour connaître toutes les commandes disponibles, il suffit de taper ? puis **ENTREE**.

Commands can be entered to change the config parameters

- To activate a function, select the pin and enter function code = pin number (e.g. PRI=1)

Function	Code	Valid pins number
Primary channels input	PRI	= 5, 9, 21, 25
Secondary channels input	SEC	= 1, 13, 17, 29
Telemetry	TLM	= 0, 1, 2, ..., 29
GPS Rx	GPS_RX	= 0, 1, 2, ..., 29
GPS Tx	GPS_TX	= 0, 1, 2, ..., 29
Sbus OUT	SBUS_OUT	= 0, 1, 2, ..., 29
RPM (only for Sport)	RPM	= 0, 1, 2, ..., 29
SDA (baro sensor)	SDA	= 2, 6, 10, 14, 18, 22, 26
SCL (baro sensor)	SCL	= 3, 7, 11, 15, 19, 23, 27
PWM Channels 1, ..., 16	C1 / C16	= 0, 1, 2, ..., 15
Voltage 1, ..., 4	V1 / V4	= 26, 27, 28, 29

- To disable a function, set pin number to 255

-To debug on USB/serial the telemetry frames, enter DEBUGTLM=Y or DEBUGTLM=N (default)

-To change the protocol, enter PROTOCOL=x where x= S(Sport Frsky), F(Fbus Frsky), C(CRSF/ELRS), H(Hott), M(Mpx), 2(Sbus2 Futaba), J(Jeti), E(jeti Exbus), X (spektrum SRXL2) ,or I(IBus/Flysky)

-To change the CRSF baudrate, enter e.g. BAUD=420000

-To change voltage scales, enter SCALEx=nnn.ddd e.g. SCALE1=2.3 or SCALE3=0.123

Enter SCALEx=0 to avoid sending voltage x to the Transmitter (for Frsky or Jeti)

-If a TMP36 is used on V3, enter TEMP=1 (if a second one is on V4, enter TEMP=2)-To change voltage offset, enter OFFSETx=nnn.ddd e.g. OFFSET1=0.6789

-To change GPS type: for an Ublox, enter GPS=U (configured by oXs) or E (configured Externally) and for a CADIS, enter GPS=C

-To change RPM multiplicator, enter e.g. RPM\_MULT=0.5 to divide RPM by 2

-To force a calibration of MP6050, enter MPUCAL

-To use a channel to setup Airspeed compensation factor and/or to select between the 2 Vspeed, enter the channel with ACC=1...16-To select the failsafe mode to HOLD, enter FAILSAFE=H

-To set the failsafe values on the current position, enter SETFAILSAFE

-To get the internal telemetry values currently calculated by oXs, enter FV (meaning Field Values)

-To test a protocol, you can force the internal telemetry values to some dummy values for dummy positive values, enter FVP; for dummy negative values, enter FVN

-To get the current PWM values (in micro sec, enter PWM)-To get the current config, just press Enter

Note: some changes require a reset to be applied (e.g. to unlock I2C bus)

En cas d'erreur de configuration le module en fait la remarque, par exemple :

Error in parameters: a pin is defined for Sbus Out but not for Primary nor Secondary channels input (PRI or SEC)

Attention: error in config parameters

### 5.3.3 Définition du protocole de télémétrie

Selon le protocole, les broches utilisées pour les canaux RC PRIMAIRES/SECONDAIRES et pour la télémétrie (TLM) varient

Protocole	PRI	SEC	TLM	Commentaire
C(ELRS)	(TX de Rx1)	(TX de Rx2)	(RX de RX1)	
S(Frsky sport)	(Sbus de Rx1)	(Sbus de Rx2)	(Sport de RX1 ou Rx2)	(1)
F(Frsky Fbus)	Fbus de Rx1	(Sbus de Rx2)	Non utilisé	(2)
J(Jeti ex)	(Sbus de Rx1)	(Sbus de Rx2)	(Ex de Rx1 ou Rx2)	
E(Jeti Exbus)	Exbus de Rx1	(Sbus de Rx2)	Non utilisé	(2)
H(Hott)	(Sbus de Rx1)	(Sbus de Rx2)	(tlm de RX1 ou Rx2)	(1)
M(Multiplex)	(Sbus de Rx1)	(Sbus de Rx2)	(tlm de RX1 ou Rx2)	(1)
I(Flysky Ibus)	(Sbus de Rx1)	(Sbus de Rx2)	(Ibus de RX1 ou Rx2)	(1)
L(Spektrum Srxl2)	Srxl de Rx1	Non utilisé	Non utilisé	(2)
2(Futaba Sbus2)	Sbus2 de Rx1	(Sbus de Rx2)	Sbus2 de Rx1 via 1Kohm	(3)

Remarque : les broches entre () signifient qu'elles sont facultatives.

(1) pour la sécurité, insérez une résistance de 1 kOhm entre la broche TLM et Rx

(2) pour la sécurité, insérez une résistance de 1 kOhm entre la broche PRI et Rx

(3) Pour Futaba, la broche TLM doit être égale à la broche PRI - 1 et insérer une résistance de 1 kOhm entre PRI et TLM

Pour plus de détails, lire le fichier [Readme.md](#) .



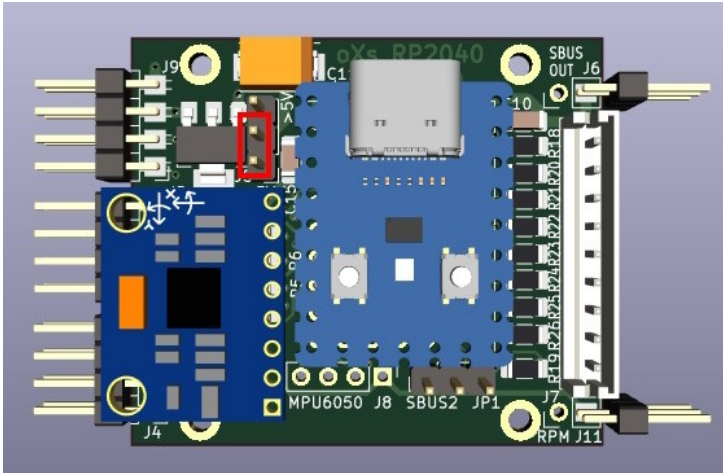
## 6 Câblage

### 6.1 Alimentation du module

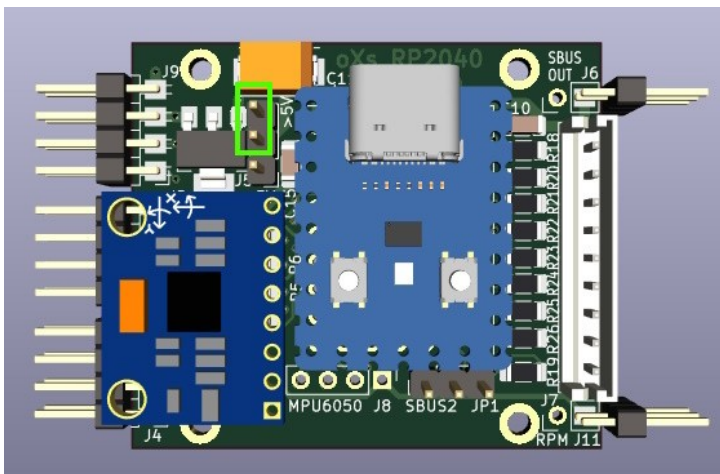
Le circuit imprimé est équipé d'un régulateur de tension 5v.

Il peut être mis en ou hors fonction selon l'accu de réception utilisé.

Accus inférieur ou égal à 5V :



Accus supérieur à 5V :



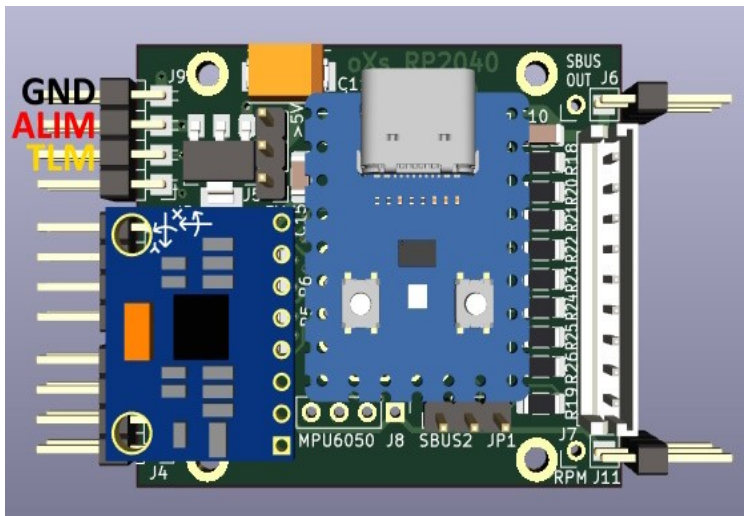
### 6.2 Limite du RP2040

**Le RP2040 ne supporte pas de tension supérieure à 3,3v en entrée !!!!**

**Toute tension supérieure détruira le composant !!!!**

## 7 Les protocoles

### 7.1 S-Port, Hott, M-Link, Ibus, Jeti Ex



S-Port **PROTOCOL=S**                      **PRI=255 SEC=255 TLM=8**

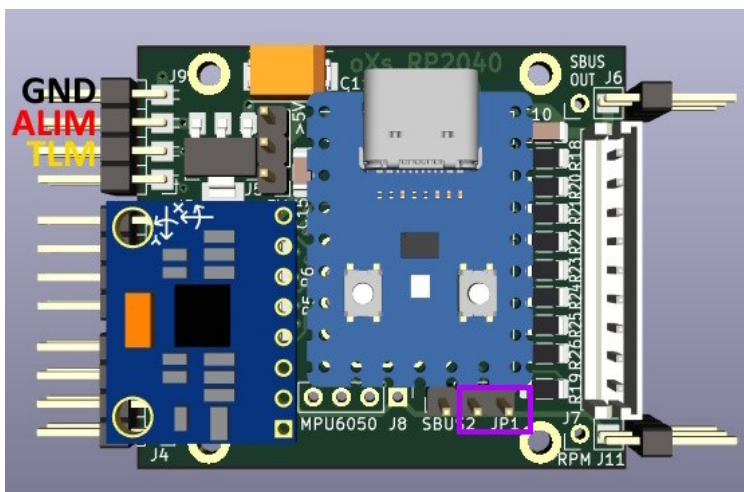
Hott    **PROTOCOL=H**

M-link **PROTOCOL=M**

Ibus    **PROTOCOL=I**

Jeti Ex **PROTOCOL=J**

### 7.2 Sbus2, Srxl2, Jeti ExBus, Fbus



Sbus2    **PROTOCOL=2**

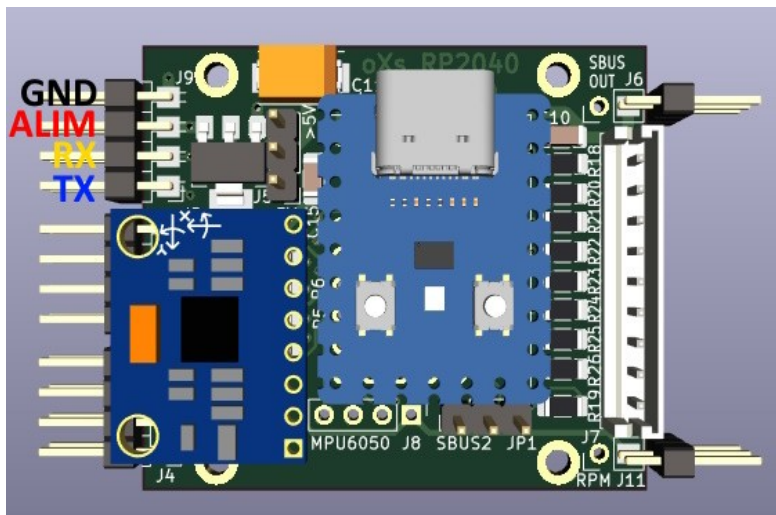
Fbus    **PROTOCOL=F**

Srxl2    **PROTOCOL=L**

**PRI=9 SEC=255 TLM=255**

Jeti Exbus **PROTOCOL=E**

## 7.2 ExpressLRS



ExpressLRS **PROTOCOL=C**      **PRI=9 SEC=255 TLM=8**

## 8 Utilisation de deux récepteurs

Tous les récepteurs ayant une sortie SBUS sont utilisables indépendamment du type de télémétrie.

On utilise les entrées **PRI=9**, **SEC=1** et **SBUS\_OUT=0**.

Le module version Full n'a pas de sortie PWM directement accessibles sur le circuit imprimé, mais il est possible de souder des connecteurs sur les pattes 1 à 7 du RP2040.

Il faut alors affecter ces 6 sorties PWM au RP2040.

**C1=2** à **C6=7**.

On peut très bien affecter ces sorties selon nos besoins sachant que le RP2040 supporte un maximum de 16 voies.

Par exemple **C9=2** à **C14=7**.

Les signaux PWM et Sbus générés seront émis à partir des derniers canaux Rc reçus. Cela fournit donc une sorte de redondance/diversité.

## 9 Les capteurs

Les capteurs utilisent différentes connexions au RP2040.

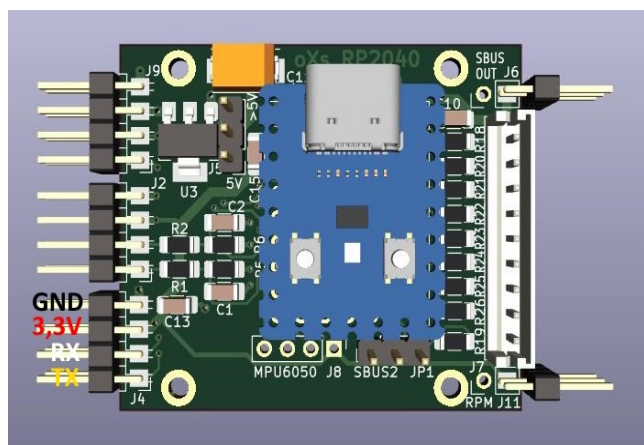
- Le GPS utilise une liaison série.
- Les 4 tensions V1 à V4 utilisent une 4 entrées analogiques.
- Le tachymètre utilise une entrée numérique.
- Tous les autres capteurs utilisent une connexion I2C.
  - \* capteur de pression (carte GY63 ou GY86 basée sur MS5611, SPL06 ou BMP280) pour obtenir l'altitude et la vitesse verticale
  - \* un capteur de pression différentielle MS4525D0\_A ou SDP3X (x=1,2,3) ou SDP8xx pour obtenir la vitesse (et la vitesse verticale compensée)
  - \* un MP6050 (acc+gyroscope ex GY86) pour améliorer le temps de réaction du vario ou pour obtenir du tangage/roulis
  - \* 1 ou 2 ADS1115 si vous souhaitez mesurer plus de 4 tensions analogiques

### 9.1 Le GPS

Différents récepteurs GPS sont utilisables :



Câblage sur la carte :





## 9.2 Sonde Pitot (capteur de pression différentielle)

La sonde Pitot permet de connaître la vitesse horizontale du modèle.

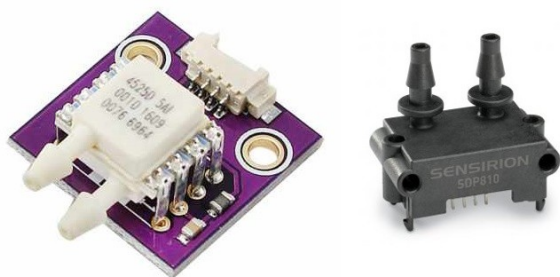
Il permet aussi de calculer la vitesse verticale compensée.

Différents capteurs sont utilisables :

\* MS4525D0\_A ou SDP3X (x=1,2,3) ou SDP8xx .

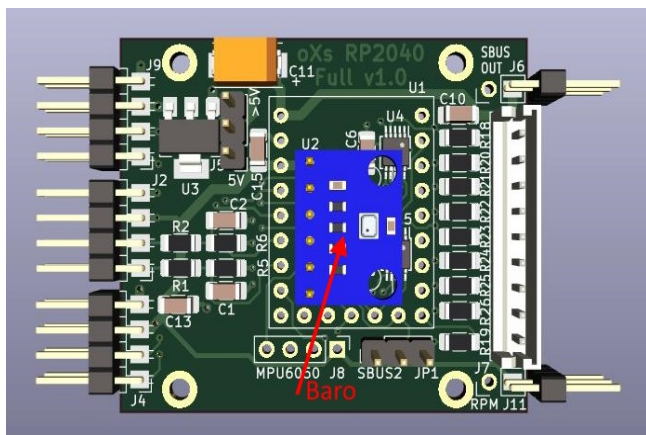
Ces capteurs sont chers, les versions SDP sont donc à privilégier.

Ces capteurs sont à câbler sur J8 avec 4 fils, 3,3v, GND, SDA et SCL.



## 9.3 Le baromètre

Le capteur BMP280 est soudé sous le RP2040.

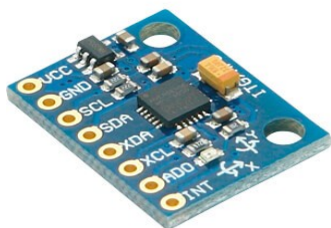


## 9.4 Accéléromètre gyroscope

Ce capteur deux en un permet d'améliorer le temps de réaction du vario ainsi que d'obtenir les données de tangage et roulis.

On peut utiliser un MPU6050 .

Ce capteur est à câbler sur J8 avec 4 fils, 3,3v, GND, SDA et SCL.



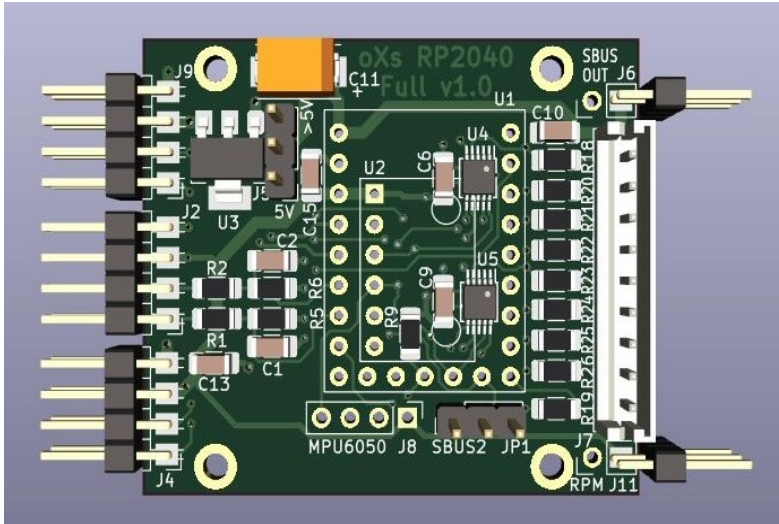
Après câblage de ce composant, celui-ci doit être calibré.  
Pour cela, il doit être placé bien à plat.

Puis on lance la commande suivante :  
**MPUCAL** puis **ENTREE**

## 9.5 Les convertisseurs analogiques ADS1115

Les ADS1115, U4 et U5, sont des quadruples convertisseurs analogique/numériques. Ils permettent de convertir 4 tensions en valeurs numériques. L'idée d'origine était de palier à la limite de 4 entrées analogiques du RP2040 et de pouvoir mesurer un accumulateur LIPO 8S maximum (connecteur blanc de droite).

Une autre idée serait d'utiliser ces ADS1115 comme huit entrées d'alarmes. Il faudrait alors imaginer des alarmes dont le seuil serait défini entre 0 et 3,3v.



**Pour rappel, le RP2040 ne supporte pas de tension supérieure à 3,3v en entrée !!!!**  
**Toute tension supérieure détruira le composant !!!!**

## 10 Réglage du Failsafe

oXs possède une fonction de Failsafe sur les sorties PWM et SBUS.

\* Pour le protocole ELRS, oXs ne reçoit aucune donnée du canal RC ou des récepteurs lorsque la connexion RF est perdue. Si oXs est connecté à 2 récepteurs (via PRI et SEC), oXs générera des signaux PWM et Sbus sur les dernières données reçues. Si oXs ne reçoit plus de données du ou des récepteurs, il continuera à générer des signaux PWM et/ou SBUS en fonction de la configuration de sécurité intégrée stockée dans oXs.

\* Pour les protocoles Frsky/Jeti... où Sbus, les valeurs de sécurité sont normalement définies à l'intérieur du récepteur et le récepteur continue à générer un signal Sbus même si la connexion RF est perdue. Pourtant, lorsque la connexion est perdue, le signal Sbus contient des drapeaux indiquant que certaines données sont manquantes ou que la sécurité intégrée a été appliquée. Lorsque oXs est connecté à 2 récepteurs différents, il donne la priorité au signal Sbus PRI sauf si le signal SEC est OK et PRI n'est pas OK (pas de signal, trame manquante, failsafe). Ainsi, pour Frsky/Jeti, oXs n'a pas à s'occuper de sa propre configuration de sécurité (sauf si oXs ne reçoit plus de signal Sbus - par exemple en raison d'un problème de câblage).

\* Le failsafe oXs a 3 options :

- \* "Hold" = failsafe sera les dernières valeurs des canaux Rc connus juste avant que la connexion ne soit perdue ; pour sélectionner cette option, utilisez l'interface série avec la commande **FAILSAFE=H**.

- \* Sauvegarder comme sécurité intégrée les valeurs actuelles des canaux RC à l'aide de l'interface série avec la commande **SETFAILSAFE**.

- \* Sauvegarder en tant que sécurité intégrée les valeurs actuelles des canaux RC à l'aide du bouton "boot" du RP2040. Pour activer cette option, double-cliquez sur le bouton. La led doit devenir bleue fixe. Dans les 5 secondes suivantes, appuyez et maintenez le bouton "boot". La led deviendra blanche pendant 2 secondes confirmant que les valeurs sont enregistrées.

Pour les 2 dernières options, l'émetteur doit être allumé et générer les valeurs des canaux que vous souhaitez enregistrer dans oXs.