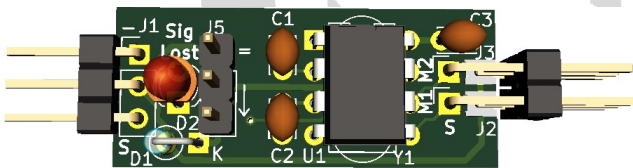


Xany2Msx

**Un cordon
d'adaptation pour
interfacer les
modules MultiSwitch
du commerce avec
*OpenAVRc***



Manuel Utilisateur Adaptateur Xany2Msx



Copyright *OpenAVRc* 2018

Table des matières

1	CE DOCUMENT.....	3
1.1	Versions.....	3
1.2	Copyright.....	3
1.3	Avertissement.....	3
1.4	Contenu.....	3
2	PRESENTATION DE XANY2MSX.....	4
2.1	Vue d'ensemble.....	4
2.2	Facilité de migration vers le 2.4GHz.....	5
2.3	Résistance aux perturbations radio-électriques.....	5
2.4	Spécifications du cordon adaptateur Xany2Msx.....	5
3	SCHEMA DE L'ADAPTATEUR XANY2MSX.....	6
4	REALISATION DE L'ADAPTATEUR XANY2MSX.....	7
4.1	Liste de composants.....	7
4.2	Câblage « en volant ».....	7
4.3	Réalisation d'un circuit imprimé.....	8
4.4	Montage de l'Attiny85 sur support tulipe.....	9
4.5	Chargement du Firmware dans l'Attiny85.....	9
5	UTILISATION.....	10
5.1	Connexion au récepteur.....	10
5.2	Mode avancé.....	11
5.2.1	Utilisation du port série de Xany2Msx.....	11
5.2.2	Les messages de commande de Xany2Msx.....	12
5.3	Exemple de configurations réelles.....	13
5.3.1	Contrôle d'un Multi-Switch 8 sorties.....	13
5.3.2	Contrôle d'un Multi-Switch 16 sorties.....	13
5.3.3	Contrôle de 2 Multi-Switches 8 sorties.....	14
5.3.4	Contrôle d'un décodeur Robbe MS12 + 2x PROP.....	14
5.3.5	Contrôle d'un décodeur Multiplex MultiNaut Top.....	15
5.3.6	Contrôle d'un décodeur Graupner Nautic Expert.....	15

1 CE DOCUMENT

1.1 Versions

Version	Date	Raison de l'évolution
0.1	20/12/2018	Création
0.2	04/04/2021	Adaptation au schéma V1.0

1.2 Copyright

Ce document est Copyright © 2018 - 2021 **OpenAVRc**.

1.3 Avertissement

L'équipe **OpenAVRc** n'est aucunement responsable des dommages qui pourraient découler de la mauvaise utilisation ou d'un éventuel dysfonctionnement de l'émetteur **OpenAVRc**, de l'adaptateur **Xany2Msx** et/ou des logiciels associés.

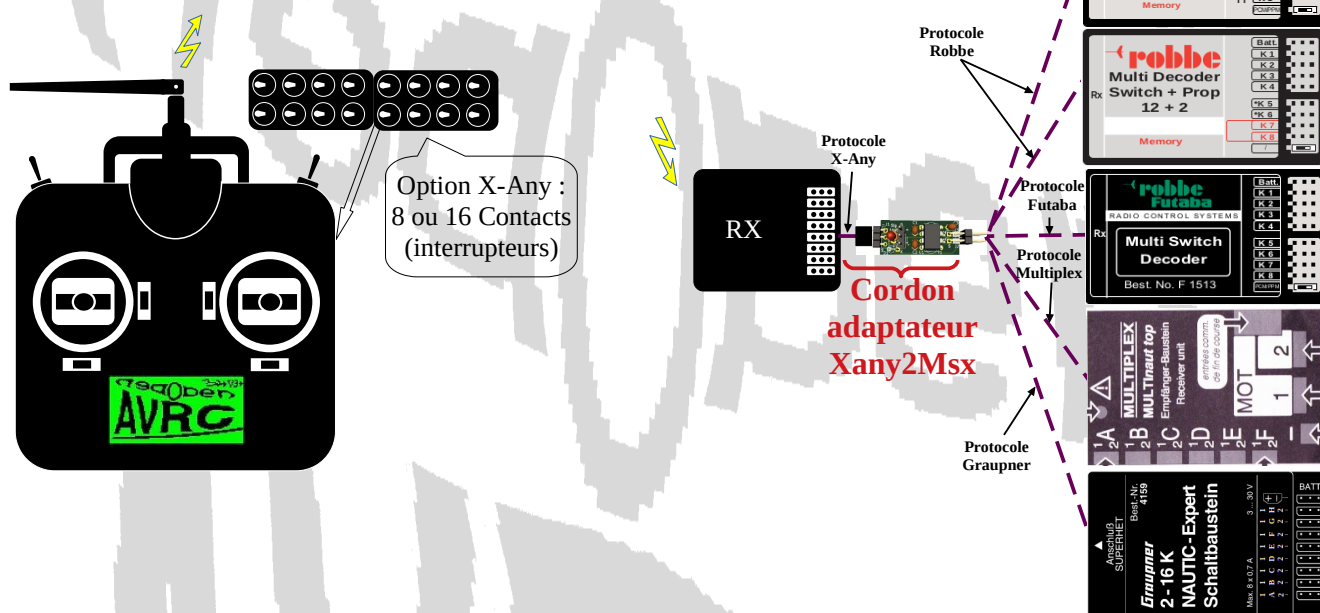
Il appartient donc à l'utilisateur final d'en mesurer, d'en assumer les risques et de respecter la législation en vigueur selon le pays d'utilisation.

1.4 Contenu

Ce document décrit la réalisation du cordon adaptateur **Xany2Msx** pour l'interfaçage avec les modules Multi-Switches supportés du commerce (Robbe, Futaba, Multiplex, Graupner), ainsi que le paramétrage pour son utilisation avec l'émetteur **OpenAVRc**.

2 PRESENTATION DE XANY2MSX

2.1 Vue d'ensemble



Le cordon adaptateur **Xany2MSx** permet d'utiliser un décodeur MS16 ou jusqu'à 2 décodeurs MS8 du commerce sur une voie « X-Any » de l'ensemble RC **OpenAVRc**.

Les décodeurs Multi-Switches/Prop du commerce actuellement supportés sont :

- | | | | |
|-----------------|-----------------------------|----------|--------------------|
| 1. Robbe | Multi Switch Decoder | No 8369 | (MS16) |
| 2. Robbe | Multi Decoder Switch + Prop | No 8370 | (MS12 + 2 PROP) |
| 3. Robbe-Futaba | Multi Switch Decoder | No F1513 | (MS8) |
| 4. MULTIPLEX | MULTInaut top | No 75882 | (MS12 + 2 MOTeurs) |
| 5. GRAUPNER | NAUTIC Expert | No 4159 | (MS16) |

Note :

1. Côté émetteur, il n'y a pas besoin du codeur spécifique correspondant au modèle de décodeur.
2. Le codeur unique et universel est intégré à l'émetteur **OpenAVRc** et utilise le protocole **X-Any**. Il est possible de déclarer jusqu'à 4 instances d' **X-Any** côté émetteur afin d'autoriser jusqu'à 64 commandes Tout-Ou-Rien (soient côté réception : 4 x décodeurs Multi-Switches 16 sorties, ou 8 x décodeurs Multi-Switches 8 sorties, ou un mélange des 2 combinaisons). Se reporter à la documentation relative à **OpenAVRc** pour configurer **X-Any** côté émetteur.
3. Pour les modules décodeurs de type Multi-Prop, les sorties proportionnelles sont gérées en Tout-Ou-Rien. Le cordon adaptateur **Xany2MSx** utilise l'état des contacts 13, 14, 15 et 16 :
 - Contacts 13+14 et 15+16 pour les 2 commandes voies proportionnelles du Multi Decoder Switch + Prop : chaque sortie proportionnelle pouvant prendre 3 valeurs fixes (mais paramétrables)
 - Contacts 13+14 et 15+16 pour les 2 commandes de moteurs du MULTInaut top : chaque commande de moteur prendre 3 valeurs fixes (mais paramétrables) : Marche Arrière/Stop/Marche Avant

2.2 Facilité de migration vers le 2.4GHz

Bon nombre de modélistes adeptes d'animations (donc de modules Multi-Switches) ne migrent pas vers les radios 2.4 GHz, puisque les anciens codeurs/décodeurs Multi-Switches sont rarement supportés en 2.4 GHz, ou bien, il est nécessaire de réinvestir dans des modules compatibles avec le 2.4 GHz.

Avec le cordon adaptateur **Xany2Msx**, il est très facile de migrer un ensemble RC FM/PPM vers l'émetteur **OpenAVRc** en 2.4 GHz tout en conservant le bon fonctionnement des modules Multi-Switches existants : en effet, **X-Any** est un protocole universel fonctionnant aussi bien en FM/PPM qu'en 2.4 GHz.

2.3 Résistance aux perturbations radio-électriques

Du fait de l'utilisation du protocole **X-Any** pour transporter les ordres entre l'émetteur et le récepteur, les corruptions liées aux perturbations radio-électriques sont détectées à la réception à l'aide de la somme de contrôle (checksum 8 bits).

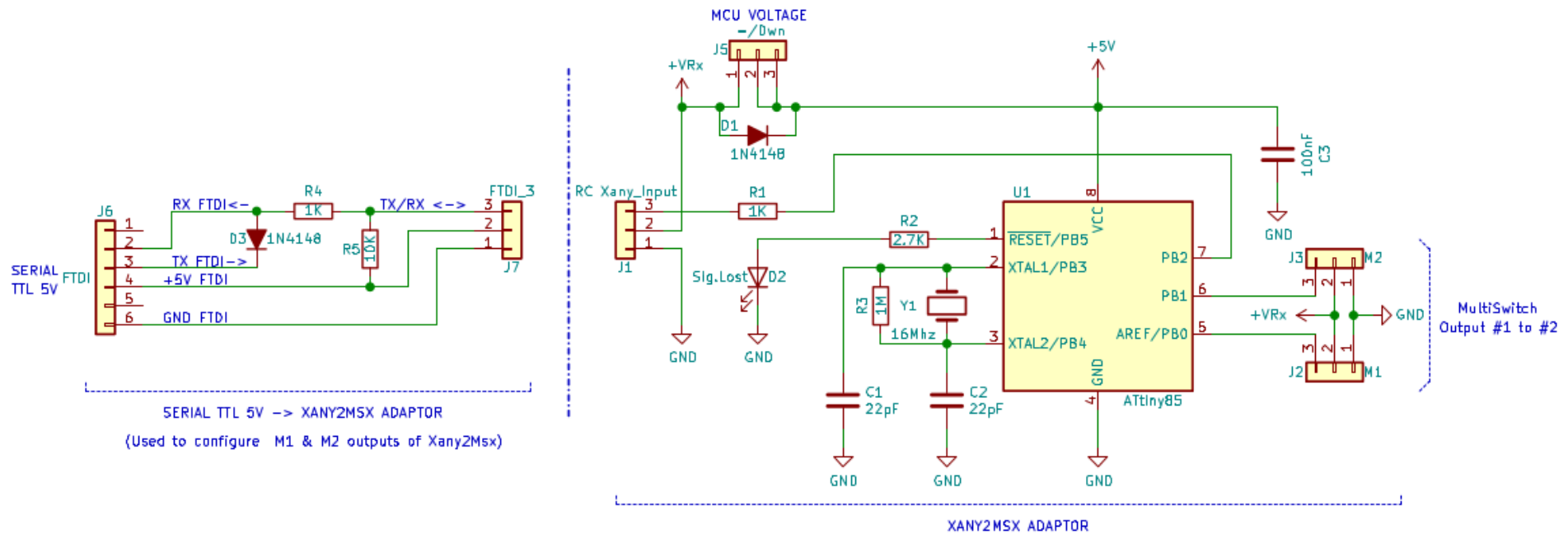
Contrairement au protocole FM/PPM, le cordon adaptateur **Xany2Msx** est en mesure de détecter la corruption lors du transport et peut ainsi passer en mode Failsafe en fournissant des commandes à 0 au(x) décodeur(s) Multi-Switch connecté(s) sur la(les) sortie(s) de **Xany2Msx**.

Donc, finis les soubresauts incontrôlés en sortie de décodeur Multi-Switch !

2.4 Spécifications du cordon adaptateur Xany2Msx

Spécification	Valeur / Caractéristique	Note
Alimentation	+3.3V à +6.6V	Mettre le cavalier « tension récepteur » sur « = » ou « ↓ » selon la valeur de la tension fournie par le récepteur
Protocole Entrant	<ul style="list-style-type: none">- Protocole numérique X-Any universel utilisé par OpenAVRc pour les accessoires distants- Contrôle d'intégrité par checksum 8 bits- Fonctionne avec tous les protocoles, y compris en FM/PPM et en 2.4 GHz :<ul style="list-style-type: none">- Protocole PPM- Protocoles SPIRfMod- Protocoles MultiMod	Contrairement à bon nombre de modules Multi-Switches, l'adaptateur Xany2Msx pour Multi-Switches fonctionne également avec les modules HF en 2.4 GHz
Protocoles Sortants	Les protocoles des modèles de Multi-Switches actuellement supportés sont : <ul style="list-style-type: none">- Robbe Multi Switch 16- Robbe Multi Switch Prop 12+2- Robbe-Futaba Multi Switch 8- Multiplex MultiNaut Top- Graupner Nautic Expert	Chacune des 2 sorties Multiswitch M1 à M2 peut être configurée avec chacun de ces protocoles de Multi-Switch.
Failsafe	Désactive toutes les sorties si pas de signal valide pendant 1,5s : La Led rouge « Signal Lost » s'allume.	En présence de signal valide, a Led rouge « Signal Lost » est éteinte.

3 SCHEMA DE L'ADAPTATEUR XANY2MSX



An adaptor for using many commercial Multiswitch decoders with OpenAVRc transmitter
It behaves like a protocol converter: X-Any protocol → Manufacturer protocol
Supported decoders: Futaba MS8 (1513), Robbe MS16 (8369), Robbe MS12+2xProp (8370)
Multiplex Multinaut MS12+2xEngines (75882), Graupner Nautic MS16 (4159)

OpenAVRc Team (C) 2018 – 2021

Sheet: /

File: Xany2Msx.sch

Title: Xany2Msx Adaptor

Size: A4

Date: 2021-04-04

Rev: 1.0

KiCad E.D.A. eeschema 5.1.9-73d0e3b20d88ubuntu20.04.1

Id: 1/1

4 REALISATION DE L'ADAPTATEUR XANY2MSX

4.1 Liste de composants

Composants	Quantité	Remarque
Support « tulipe » 8 points	1	Indispensable, car présence Quartz
Quartz 16MHz	1	
Condensateur céramique 22 pF	2	
Condensateur film plastique MKP 100 nF	1	Découplage d'alimentation
Résistance 1K 1/4 W	2	
Résistance 2.7K 1/4 W	1	
Résistance 10K 1/4 W	1	
Led rouge diamètre 3 mm haut rendement	1	
Diode 1N4148	2	
Microcontrôleur ATtiny85-20PU	1	A charger avec le HEX du Firmware
Cordon de servo avec connecteur 3 points femelle	1	
Barrette de connecteur 3 points au pas de 2.54 mm	6	2 pour les sorties M1 à M2 , 1 pour la sélection du niveau d'alimentation et 3 pour l'adaptateur FTDI
Cavalier pour barrette 3 points	1	Pour sélection du niveau d'alimentation
Gaine thermorétractable (diamètre à froid : 18 mm)	2 cm	Uniquement si câblage « en volant »

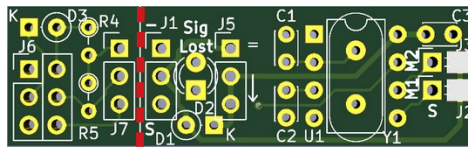
4.2 Câblage « en volant »

Le cordon adaptateur [Xany2Msx](#) peut être réalisé autour d'un support « tulipe » 8 points et d'une poignée de composants. Tout est soudé « en volant » : il n'y a aucun circuit imprimé à réaliser. Une fois assemblé, l'ensemble est emprisonné dans de la gaine thermorétractable.

TO DO : décrire l'assemblage

4.3 Réalisation d'un circuit imprimé

Autre possibilité, plus propre, mais moins compacte : la réalisation d'un circuit imprimé.

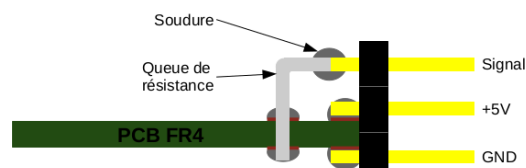
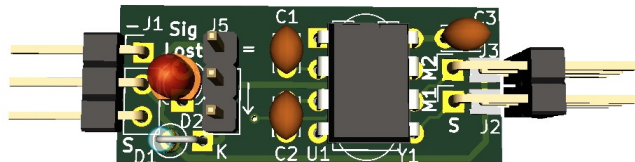


Note :

A réception du circuit imprimé, il est nécessaire de le couper au niveau de la ligne en pointillé **rouge** afin de séparer les 2 parties.

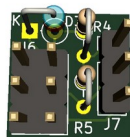
Le circuit imprimé proposé est composé de 2 parties :

1. L'adaptateur **Xany2Msx** proprement dit : (partie droite)



Montage des connecteurs J2 et J3 de M1 et M2

2. Un adaptateur FTDI vers **Xany2Msx**: (partie gauche)



Cet adaptateur FTDI vers **Xany2Msx** est nécessaire pour le paramétrage avancé de **Xany2Msx**.

Une fois paramétré, cet adaptateur FTDI vers **Xany2Msx** est déconnecté et le montage se limite à l'adaptateur **Xany2Msx** (montage plus compact dans le modèle réduit).

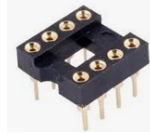
4.4 Montage de l'ATtiny85 sur support tulipe

ATTENTION :

L'**ATtiny85** doit être programmé **avant** son positionnement sur le circuit imprimé de **Xany2Msx**.

De plus, pour plus de compacité, le **quartz de 16MHz** est logé sous l'**ATtiny85**.

C'est pourquoi, il est nécessaire de le monter **l'ATtiny85** sur un support tulipe de 8 points :



Il est également possible d'utiliser 2 portions (de 4 points) de barrette tulipe sécable :



Dans les 2 cas, bien vérifier l'orientation de l'**ATtiny85** avant de mettre sous tension **Xany2Msx**.

4.5 Chargement du Firmware dans l'ATtiny85

TO DO

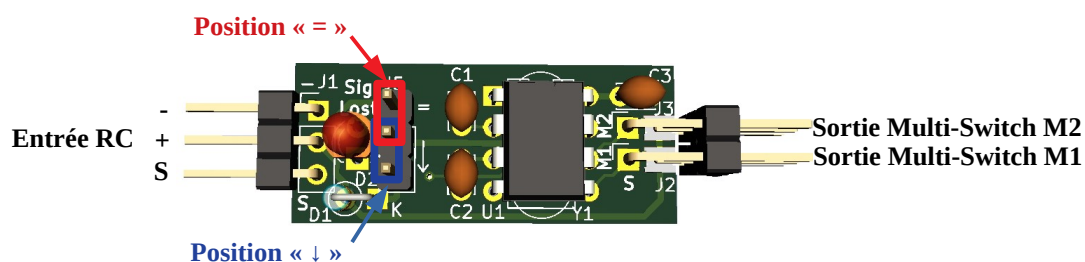
5 UTILISATION

5.1 Connexion au récepteur

Avant de connecter le cordon adaptateur **Xany2Msx** au récepteur, il est impératif de mesurer la tension fournie par le récepteur.

Si la tension disponible entre les broches – et + du connecteur 3 points de la voie utilisée est :

1. Inférieure à 5.7V, placer le cavalier « tension récepteur » sur « = »
2. Supérieure à 5.7V, placer le cavalier « tension récepteur » sur « ↓ »



5.2 Mode avancé

Le cordon adaptateur **Xany2Msx** dispose d'un accès pour les configurations avancées : un port série TTL.

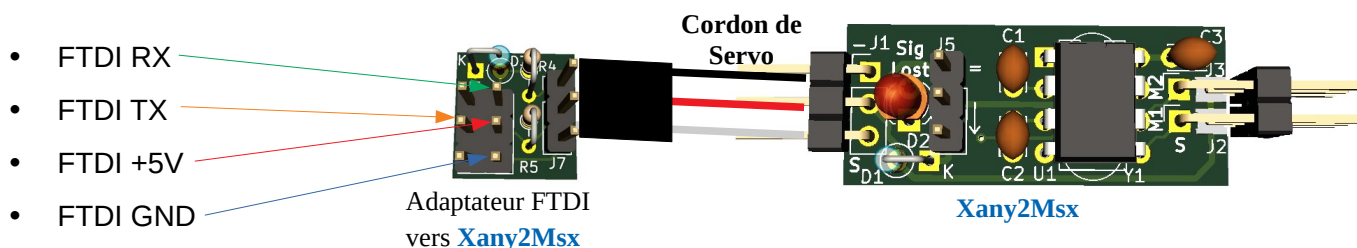
Pour l'utiliser, il est nécessaire de déconnecter l'entrée RC de **Xany2Msx** afin de ne pas perturber le fonctionnement de l'accès série.

C'est cet accès série qui permet de définir le modèle de décodeur Multi-Switch à connecter aux sorties **M1** et **M2**.

5.2.1 Utilisation du port série de Xany2Msx

Pour accéder au port série de **Xany2Msx**, il faut un câble USB/Série TTL par exemple de type « FTDI ».

Les points nécessaires sur le câble USB/Série TTL sont :

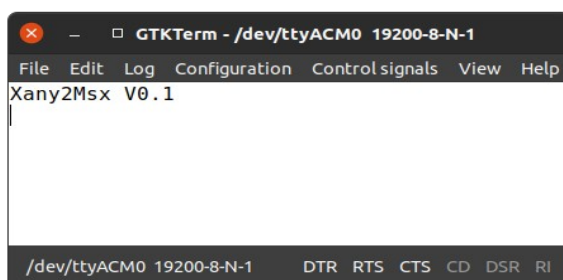


Le câble USB/Série TTL doit être capable de fournir le +5V pour alimenter **Xany2Msx** car pendant la configuration avancée, **Xany2Msx** doit être déconnecté du récepteur.

1. Connecter le câble USB/Série TTL sur le port USB d'un PC pour le côté USB
2. Sur le PC, ouvrir un Terminal série, par exemple, PuTTY, TeraTerm, HyperTerminal, GtKTerm, ou encore CoolTerm avec les paramètres suivants : 19200 bauds, 8 bits de données, 1 bit de stop, pas de parité.

Selon le Terminal série, il peut être nécessaire d'activer les retours à la ligne automatique sur réception de CR/LF pour avoir un bon affichage.

3. Connecter le câble USB/Série TTL sur le connecteur J6 de l'adaptateur FTDI vers **Xany2Msx**
4. Connecter J7 de l'adaptateur FTDI vers **Xany2Msx** à J1 de **Xany2Msx**, ceci va alimenter **Xany2Msx**
5. **Dans les 3 secondes** après le branchement sur **Xany2Msx**, appuyer sur la touche « Entrée » de votre clavier, le message « Xany2Msx VX.Y » doit apparaître sur le Terminal série comme illustré ci-dessous. Si ce n'est pas le cas, déconnecter J7 de J1 de **Xany2Msx** et recommencer à l'étape 4 ci-dessus.



Exemple de connexion avec le Terminal GtKTerm sous Linux

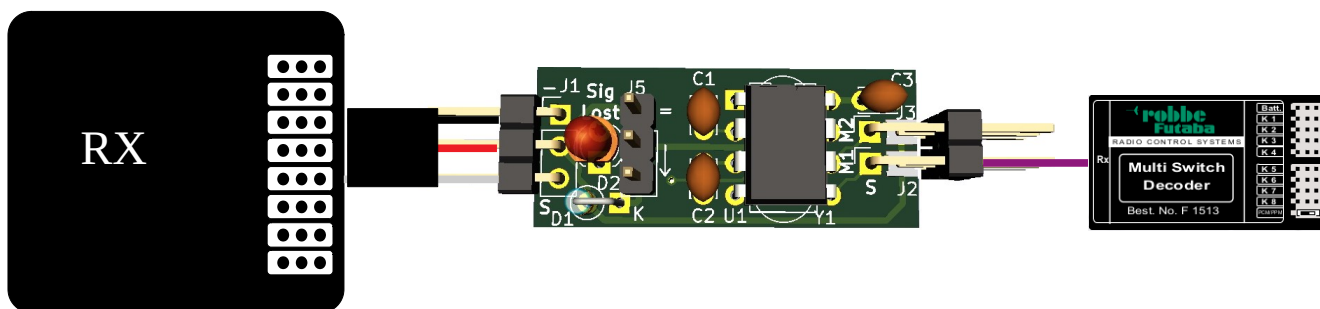
5.2.2 Les messages de commande de Xany2Msx

La liste des messages supportés par **Xany2Msx** est donnée dans la table suivante :

← Commande/ → Réponse	Action	Remarque
← Enter → Xany2Msx Vx.y	Si envoyée pendant les 3 secondes après la mise sous tension, passe en mode Terminal	Si échec et 3 secondes écoulées, débrancher puis rebrancher le connecteur J7 3 points du câble USB/Série TTL
← Mx? → Mx=FP ou Mx=OFF	Si x est compris entre 1 et 2, retourne la configuration de la Sortie Multiswitch N°x : - F étant le Fabricant: - R pour Robbe - F pour Robbe-Futaba - M pour Multiplex - G pour Graupner - P étant le code Produit chez le Fabricant - 0 pour tous les autres - OFF : la Sortie est désactivée	Renvoie ERR, si : - Valeur x non comprise entre 1 et 2 Exemple de réponses : M1=R0 M1=R1 M2=F0 M2=M0 M2=G0
← Mx=FP ou Sx=OFF → Mx	Définit la sortie Multiswitch x comme étant du Fabricant F pour le code Produit P ou désactive la Sortie (OFF)	M1=R1 M2=OFF
← Mx=Ky;Pos0;PosOff;Pos1 → Mx	Pour le type R1 (Robbe 12+2), les sorties proportionnels K7 et K8 peuvent être paramétrés avec 2 valeurs extrêmes de position	Renvoie ERR, si : - Valeur y non comprise entre 7 et 8 Exemple de commandes: M1=K7;1000;1500;2000 M1=K8;2000;1500;1000
← Mx=Ez;VitMar;VitStop;VitMaV → Mx	Pour le type M0 (Multiplex MultiNaut Top), les moteurs (Engines) 1 et 2 peuvent être paramétrés avec 3 vitesses (une pour chaque sens et une pour l'arrêt)	Renvoie ERR, si : - Valeur z non comprise entre 1 et 2 Exemple de commandes: M1=E1;1100;1300;1500 M1=E2;1500;1300;1100
← Q	Quitte le mode Terminal : Xany2Msx peut être connecté au récepteur	Ne pas oublier de déconnecter le câble USB/Série TTL !

5.3 Exemple de configurations réelles

5.3.1 Contrôle d'un Multi-Switch 8 sorties



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 8 contacts (et pas 16).

Pour commander un Multi-Switch **Futaba MS8** sur la sortie M1, il suffit de paramétrer **Xany2Msx** avec les commandes :

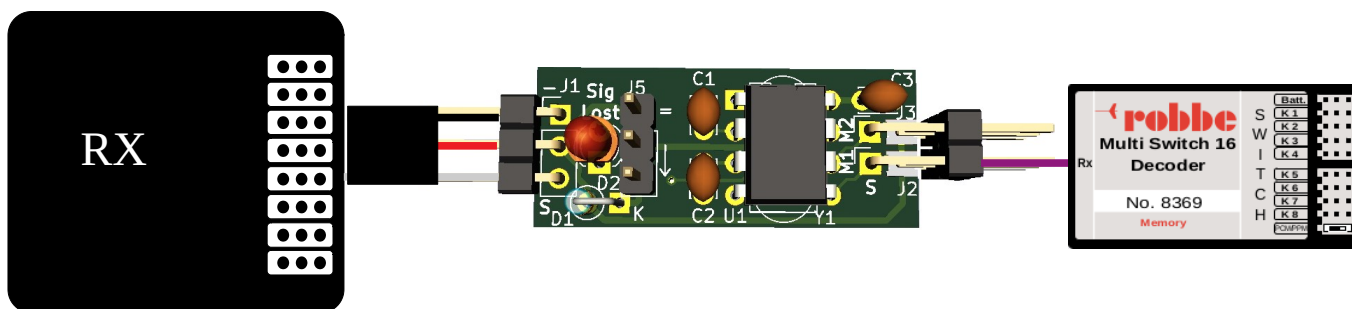
M1=F0

M2=OFF

Note :

Il est important de mettre M2 à OFF, c'est cela qui indique à **Xany2Msx** qu'il va recevoir uniquement l'état de 8 contacts.

5.3.2 Contrôle d'un Multi-Switch 16 sorties



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 16 contacts (et pas 8).

Pour commander un Multi-Switch **Robbe MS16** sur la sortie M1, il suffit de paramétrer **Xany2Msx** avec la commande :

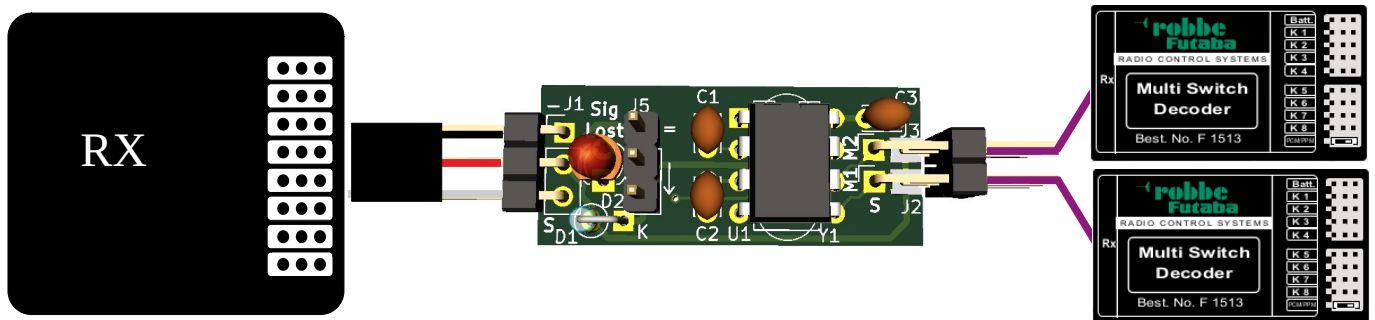
M1=R0

M2=OFF

Note :

M2 peut être paramétrée avec n'importe quelle valeur.

5.3.3 Contrôle de 2 Multi-Switches 8 sorties



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 16 contacts (et pas 8).

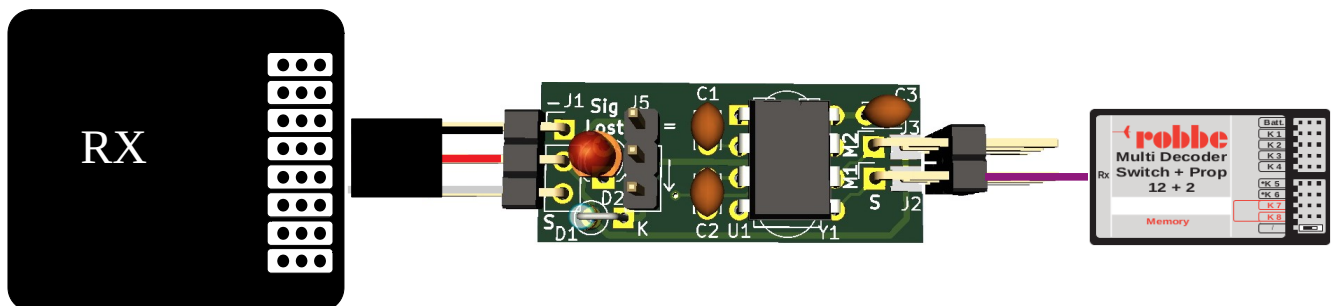
Pour commander des Multi-Switch **Robbe MS8** sur les sorties M1 et M2, il suffit de paramétrer **Xany2Msx** avec les commandes :

M1=F0

M2=F0

- Le décodeur Multi-Switch à 8 sorties connecté sur la sortie M1 réagira à l'état des contacts 1 à 8 de l'émetteur
- Le décodeur Multi-Switch à 8 sorties connecté sur la sortie M2 réagira à l'état des contacts 9 à 16 de l'émetteur

5.3.4 Contrôle d'un décodeur Robbe MS12 + 2x PROP



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 16 contacts (et pas 8).

Pour commander des Multi-Switch **Robbe MS12 + 2x PROP** sur le sortie M1, il suffit de paramétrer **Xany2Msx** avec les commandes :

M1=R1

M1=K7;1000;1500;2000

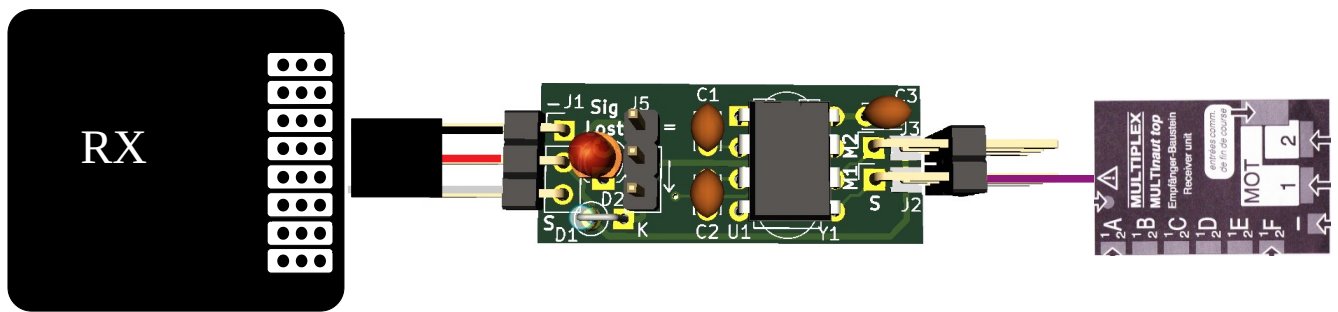
M1=K8;2000;1500;1000

M2=OFF

Note :

M2 peut être paramétrée avec n'importe quelle valeur.

5.3.5 Contrôle d'un décodeur Multiplex MultiNaut Top



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 16 contacts (et pas 8).

Pour commander un décodeur **Multiplex Multinaut Top (MS12 + 2x Engines)** Sur la sortie M1, il suffit de paramétrer **Xany2Msx** avec les commandes :

M1=M0

M1=E1;1100;1300;1500

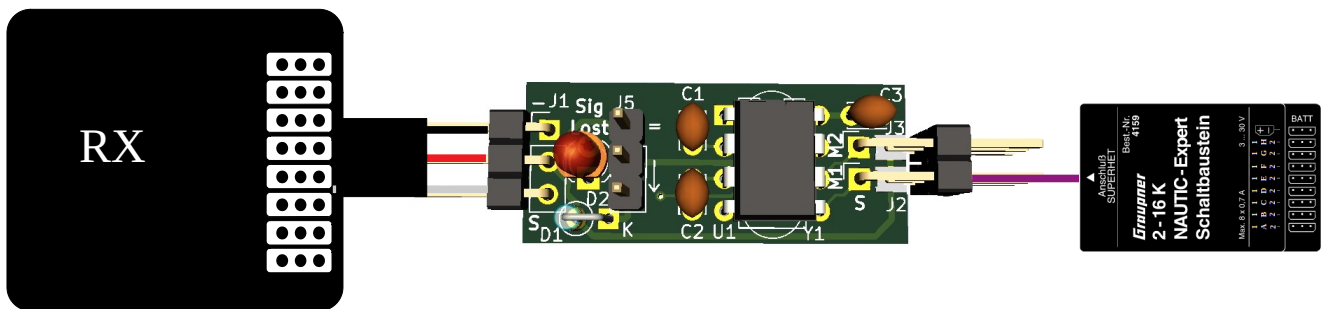
M1=E2;1500;1300;1100

M2=OFF

Note :

M2 peut être paramétrée avec n'importe quelle valeur.

5.3.6 Contrôle d'un décodeur Graupner Nautic Expert



Régler l'émetteur **OpenAVRc** pour que l'instance **X-Any** envoie l'état de 16 contacts (et pas 8).

Pour commander un décodeur MS16 **Graupner Nautic Expert** sur la sortie M1, il suffit de paramétrer **Xany2Msx** avec les commandes :

M1=G0

M2=OFF

Note :

M2 peut être paramétrée avec n'importe quelle valeur.