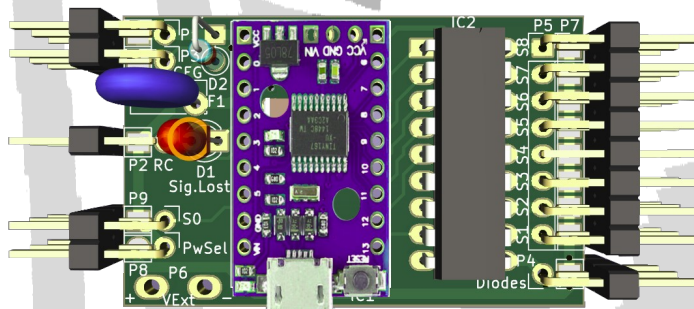


MS8-Xany V2

**Bien plus qu'un simple
MultiSwitch à 8 sorties
pour *OpenAVRc*,
OpenTx, *EdgeTx*,
certains ensembles RC
Futaba et tous les
autres ensembles RC**



Manuel Utilisateur MultiSwitch MS8-Xany V2



Copyright OpenAVRc 2023

Table des matières

1	CE DOCUMENT.....	4
1.1	Versions.....	4
1.2	Copyright.....	4
1.3	Avertissement.....	4
1.4	Contenu.....	4
2	PRESENTATION DE MS8-XANY V2.....	5
2.1	Vue d'ensemble.....	5
2.2	Spécifications du décodeurs MS8-Xany V2.....	6
3	SCHEMA DU DECODEUR MS8-XANY V2.....	8
4	REALISATION DU DECODEUR MS8-XANY V2.....	9
4.1	Circuit imprimé.....	9
4.2	Montage des circuits intégrés sur barrette tulipe.....	9
4.2.1	Montage de l'Arduino Digispark Pro sur <i>barrette</i> tulipe.....	9
4.2.2	Montage de l'ULN2803 sur support tulipe.....	9
4.2.3	Montage des connecteurs 3 points en bout de carte.....	10
4.3	Chargement du Firmware dans l'Arduino Digispark Pro.....	10
5	UTILISATION.....	11
5.1	Connexion au récepteur.....	11
5.2	Paramètres par défaut.....	11
5.2.1	<i>Câblage des « utilisations » Tout-Ou-Rien sur les sorties.....</i>	11
5.2.2	<i>Sélection de la tension d'alimentation des sorties S1 à S8.....</i>	12
5.2.3	<i>Diodes de roue libre.....</i>	12
5.2.4	<i>Commande directe de relais avec diode intégrée à l'ULN2803.....</i>	12
5.2.5	<i>Commande de relais opto-isolés.....</i>	13
5.2.6	<i>Montage conseillé pour relais 5V opto-isolés.....</i>	14
5.3	Configuration pour MS8-Xany V2.....	15
5.3.1	<i>Configuration du mode de MS8-Xany V2 pour l'EKMFA.....</i>	15
5.3.2	<i>Configuration du mode de MS8-Xany V2 pour Boutons-Poussoirs.....</i>	17
5.3.3	<i>Configuration de l'émetteur OpenAVRc.....</i>	19
5.3.4	<i>Configuration du mode de MS8-Xany V2 pour OpenAVRc.....</i>	19
5.3.5	<i>Configuration des émetteurs OpenTx et EdgeTx.....</i>	20
5.3.6	<i>Configuration du mode de MS8-Xany V2 pour OpenTx et EdgeTx.....</i>	22
5.3.7	<i>Configuration du mode de MS8-Xany V2 pour Robbe-Futaba.....</i>	22
5.4	Mode avancé/Commande de servos.....	23
5.4.1	Utilisation du port série de MS8-Xany V2.....	23
5.4.2	Les messages de commande de MS8-Xany V2.....	24
5.4.3	Exemple de configuration réelle.....	27
5.4.4	<i>Commande en mode Normal et en mode imPulsionnel.....</i>	29
6	TABLES DES COMBINAISONS DE COMMANDE SUPPORTEES.....	30
6.1	Tout ensemble RC.....	30
6.2	Ensemble OpenAVRc.....	30
6.3	Ensembles OpenTx/EdgeTx.....	30
6.4	Ensemble Futaba supportant le MultiSwitch 8 sorties Futaba N° F1513.....	30



1 CE DOCUMENT

1.1 Versions

La version de ce document est la dernière listée dans la colonne **version** de la table ci-dessous :

Version du document	Date	Raison de l'évolution
1.0	28/05/2023	Création

1.2 Copyright

Ce document est Copyright © 2018-2023 **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**, du module décodeur **MS8-Xany V2** 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 module décodeur **MS8-Xany V2** ainsi que le paramétrage pour son utilisation avec les émetteurs **OpenAVRc**, **OpenTx**, **EdgeTx** et certains émetteurs **Futaba**.

MS8-Xany V2 peut se substituer à un décodeur MultiSwitch 8 sorties N° F1513 de **Robbe - Futaba** : il peut donc fonctionner sur un ensemble RC **Robbe-Futaba** qui supporte le décodeur MultiSwitch 8 sorties N° F1513.

Bien que moins pratique, **MS8-Xany V2** peut également être commandé par coups de manche (protocole EKMFA), ce protocole fonctionne avec **tous** les ensembles RC.

MS8-Xany V2 est un module décodeur MultiSwitch disposant de 8 sorties « Tout-Ou-Rien ».

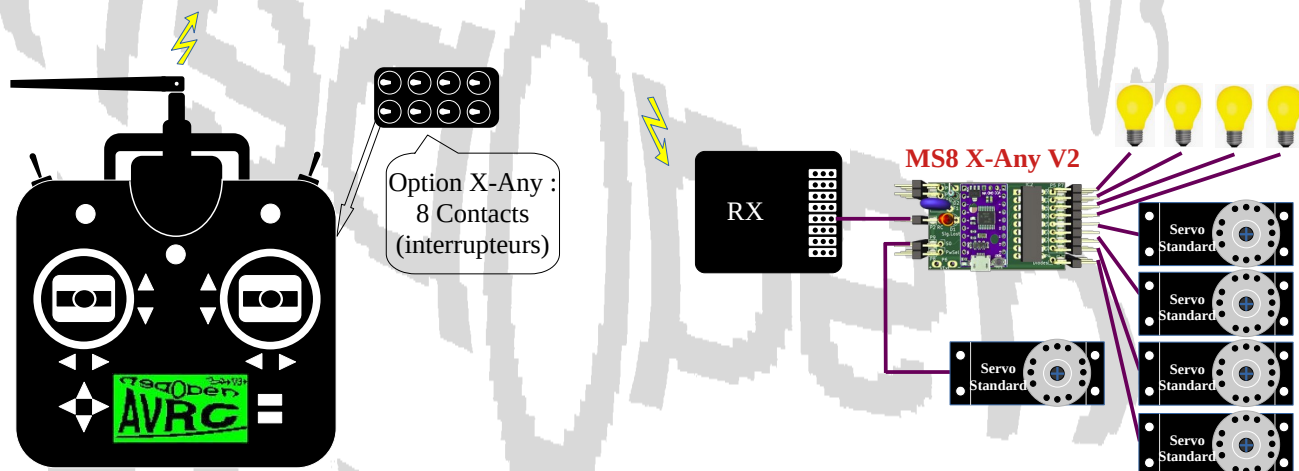
En plus de la fonction MultiSwitch classique, **MS8-Xany V2** peut être paramétré afin que ses sorties pilotent des servos en « Tout-Ou-Rien ». Les 2 positions extrêmes ainsi que le temps de déplacement entre ces 2 positions extrêmes sont paramétrables indépendamment pour chaque sortie.

Si piloté à partir d'un émetteur **OpenAVRc**, **MS8-Xany V2** peut fournir une voie proportionnelle d'appoint permettant la gestion d'un servo ou d'un ESC.

Il est également possible de commander chaque sortie en mode « impulsional » (équivalent du mode « Memory » chez Robbe-Futaba).

2 PRESENTATION DE MS8-XANY V2

2.1 Vue d'ensemble



Le décodeur **MS8-Xany V2** à 8 sorties se connecte sur le récepteur RC.

L'interface de commande sur le récepteur RC peut être :

- une sortie PWM
- une sortie CPM
- une sortie SBUS

Côté émetteur, la commande peut se faire à partir de :

- Coups de manche (Protocole **EKMFA**) : supporté par tous les ensembles RC
- D'appuis sur des boutons-poussoirs (Protocole **Boutons-Poussoirs**) connectés à une voie de l'émetteur
- Interrupteurs/bouton-poussoirs physiques/logiques (pour les émetteurs évolués : **OpenAVRc**, **OpenTx**, **EdgeTx**, certains ensembles RC **Robbe-Futaba**)

Les protocoles de commande supportés sont :

- **EKMFA** : supporté par **tous** les ensembles RC sans modification
- **Boutons-Poussoirs** : supporté par **tous** les ensembles RC (nécessite l'ajout du « clavier »)
- **RCUL/X-Any** : supporté par les émetteurs **OpenAVRc**
- **Binaire** : supporté par les émetteurs **OpenTx** et **EdgeTx**
- **Futaba** : supporté par certains ensembles RC Robbe-Futaba (Compatibilité avec le MultiSwitch 8 sorties N° F1513)

2.2 Spécifications du décodeurs MS8-Xany V2

Spécification	Valeur / Caractéristique	Note
Alimentation	+4.5V à +6.6V	Mettre le cavalier « tension récepteur » sur « = » ou « ↓ » selon la valeur de la tension fournie par le récepteur
Interface RC PWM	A connecter sur l'entrée RC de MS8-Xany V2	Ne fonctionne qu'avec les protocoles de commande EKMFA, X-Any et Futaba*. (Voir ci-dessous)
Interface RC CPPM	A connecter sur l'entrée RC de MS8-Xany V2	Ne fonctionne qu'avec les protocoles de commande EKMFA, X-Any et Futaba*. (Voir ci-dessous)
Interface RC SBUS	A connecter sur l'entrée RC de MS8-Xany V2 Pas besoin d'inverseur SBUS externe	Fonctionne avec les protocoles de commande EKMFA, X-Any, Binaire et Futaba*. (Voir ci-dessous)
Protocole de commande X-Any	- Protocole numérique universel utilisé par OpenAVRc pour les accessoires distants - Contrôle d'intégrité par checksum 8 bits - Fonctionne avec tous les protocoles, y compris en 2.4GHz : - Protocole PPM - Protocoles SPIRfMod - Protocoles MultiMod	Contrairement à bon nombre de modules MultiSwitches, MS8-Xany V2 fonctionne également avec les modules HF en 2.4GHz
Protocole de commande Futaba MultiSwitch 8 sorties	Compatibilité avec le module décodeur MultiSwitch 8 sorties N°F1513 de Robbe-Futaba	MS8-Xany V2 peut remplacer avantageusement un décodeur N°F1513 sur un ensemble supportant le décodeur MultiSwitch 8 sorties N°F1513 de Robbe-Futaba.
Protocole de commande « Binaire »	La commande des 8 sorties S1 à S8 est un mot de 8 bits. L'impulsion fournie sur l'entrée RC représente une valeur 8 bits (Valeur de 0 à 255).	Avec ce protocole, il n'est pas possible d'utiliser la sortie proportionnelle S0. OpenTx et EdgeTx doivent utiliser ce protocole.
Protocole de commande « EKMFA »	Ce protocole permet de commander les sorties S1 à S8 par coups de manche	Ce protocole de commande fonctionne sur tous les ensembles RC et fonctionne « au-dessus » des interfaces PWM, CPPM et SBUS.
Protocole de commande «Boutons-Poussoirs»	Ce protocole fonctionne avec un clavier de 8 boutons-poussoirs connecté sur une voie côté émetteur.	Une calibration est à réaliser une seule fois : les largeurs d'impulsion de voie correspondant à l'appui de chaque bouton-poussoir étant sauvegardées en EEPROM.
8 sorties Tout-Ou-Rien	- Commandées en Tout-Ou-Rien par sortie transistorisée	Sorties configurées en mode « numérique » (MultiSwitch)
Alimentation des sorties	- Interne : tension du récepteur - Externe : tension externe 5 à 24V	Sélectionnable par cavalier (commun aux 8 sorties)
Diode de roue libre	Diodes de roue libre des 8 sorties au + de l'alimentation des sorties	Sélectionnable par cavalier (commun aux 8 sorties)
8 sorties Servo	- Commandées en Tout-Ou-Rien - Positions extrêmes paramétrables - Durée du mouvement entre positions extrêmes paramétrable	- Sorties configurées en mode «Servo» - Inversion de la course des servos possible en permutant les valeurs extrêmes - Possibilité d'utiliser la tension du récepteur - Possibilité d'utiliser une tension externe (compatible avec les servos)
1 sortie S0 proportionnelle	Commande proportionnelle d'un servo ou d'un ESC de 988µs à 2008µs (0 à 255 pas de 4µs)	Uniquement disponible avec l'émetteur OpenAVRc. Présence de commande proportionnelle détectée en dynamique par MS8-Xany V2 : rien à configurer sauf côté émetteur OpenAVRc
LED rouge perte de signal	Au bout de 1 seconde sans signal	
Failsafe	- Toutes les sorties passent à 0 en cas de perte de signal RC - La sortie proportionnelle S0 conserve sa position courante	Synchrone avec LED rouge perte de signal
Port série TTL	Connecteur pour câble USB/FTDI TTL	Pour configuration avancée à l'aide d'une application type « Terminal série »
Dimensions hors tout (mm)	L x l x h : 62 x 28 x 11	

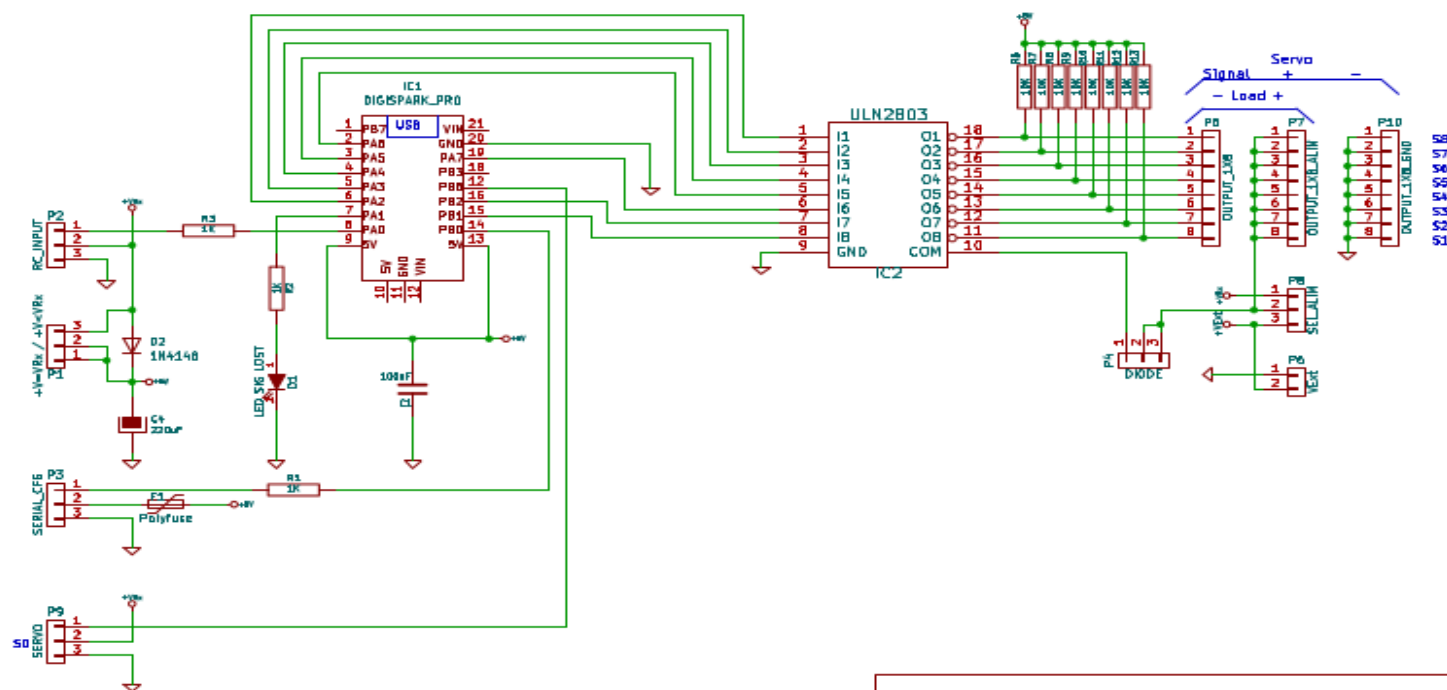
OpenV3

Page left intentionnally blank

/

Page laissée intentionnellement blanche

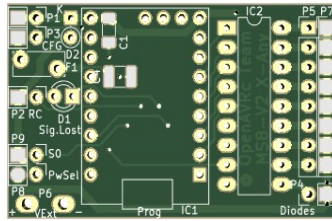
3 SCHEMA DU DECODEUR MS8-XANY V2



OpenAVRc		
Sheet: 7		
File: MultiSwitch_Sw8_V2.kicad_sch		
Title: Décodeur MS8 X-Any V2		
Size: A4	Date: 2023-02-19	Rev: 0,2
KiCad E.D.A. eeschema 7.0.0-da2b9df05c-171-ubuntu20.04.1		Id: 1/1

4 REALISATION DU DECODEUR MS8-XANY V2

4.1 Circuit imprimé

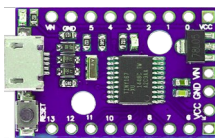


4.2 Montage des circuits intégrés sur barrette tulipe

4.2.1 Montage de l'Arduino Digispark Pro sur barrette tulipe

ATTENTION :

L'**Arduino Digispark Pro** doit être programmé **avant** son positionnement sur le circuit imprimé de **MS8-Xany V2**.



C'est pourquoi, il est fortement recommandé de le monter sur 2 portions (de 9 points) de barrette tulipe sécable :



Bien vérifier l'orientation de l'**Arduino Digispark Pro** avant de mettre sous tension **MS8-Xany V2**.

4.2.2 Montage de l'ULN2803 sur support tulipe

En cas d'erreur de câblage sur les sorties, il est possible de détruire l'**ULN2803**.



Afin de faciliter son éventuel remplacement, il est fortement recommandé de le monter sur un support tulipe de 18 points :

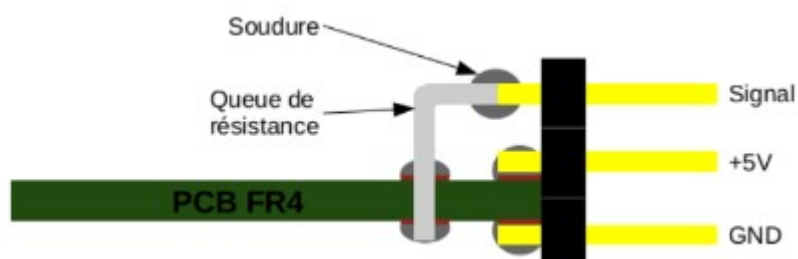


Il est également possible d'utiliser 2 portions (de 9 points) de barrette tulipe sécable :



Dans les 2 cas, bien vérifier l'orientation de l'**ULN2803** avant de mettre sous tension **MS8-Xany V2**.

4.2.3 Montage des connecteurs 3 points en bout de carte



4.3 Chargement du Firmware dans l'Arduino Digispark Pro

L'Environnement de Développement Intégré Arduino (IDE) permet de programmer les [Arduino Digispark Pro](#).

L'[Arduino Digispark Pro](#) dispose d'un programme préchargé appelé **Bootloader** qui permet, depuis un PC et par USB, le chargement initial puis les éventuelles mises à jour du firmware de [MS8-Xany V2](#).

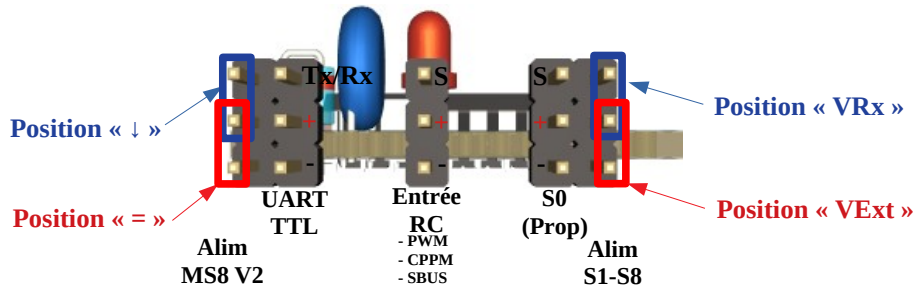
5 UTILISATION

5.1 Connexion au récepteur

Avant de connecter **MS8-Xany V2** 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. Supérieure à 5.7V, placer le cavalier « tension récepteur » sur « ↓ »
2. Inférieure à 5.7V, placer le cavalier « tension récepteur » sur « = »



5.2 Paramètres par défaut

A la première mise sous tension, **MS8-Xany V2** a les paramètres par défaut suivants :

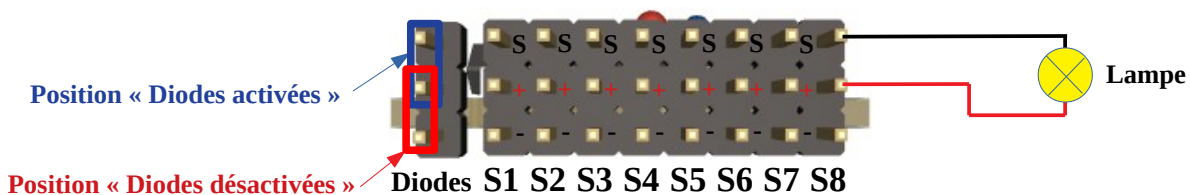
- **I=P** → L'Interface RC de commande est **PWM**
- **P=E** → Le **Protocole** de commande est **EKMFA** (Commande par coups de manche)
- **SR=N** → Le **Servo Reverse** est à **Non** pour la sortie proportionnelle S0
- **Sx=D ; N** → Les Sorties **S1** à **S8** sont en type **Digital** en mode **Normal** (Multiswitch)

Pour changer le paramétrage, se reporter au § Utilisation du port série de MS8-Xany V2.

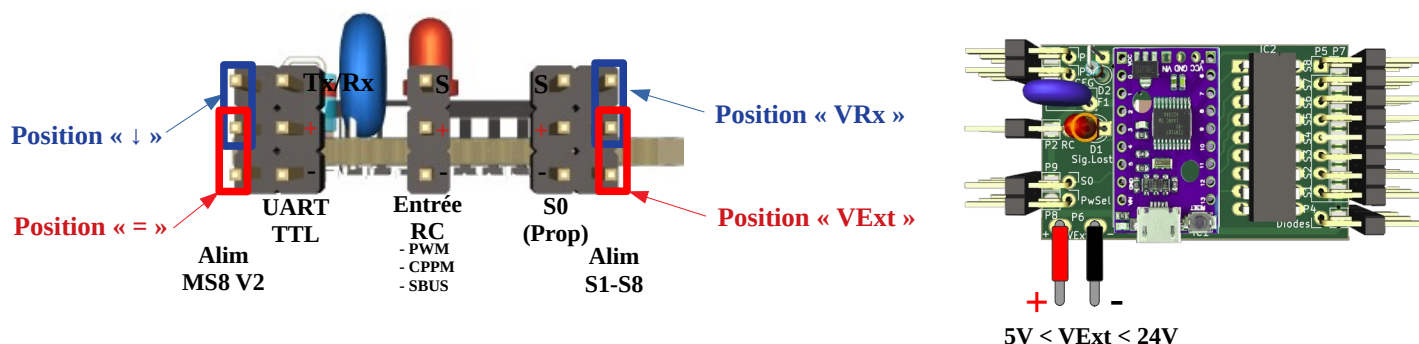
5.2.1 Câblage des « utilisations » Tout-Ou-Rien sur les sorties

Le **MS8-Xany V2** s'utilise alors comme un module MultiSwitch standard :

- Les « utilisations » (ex : une lampe) se branchent sur les sorties **S1** à **S8** entre les broches « S » et « + », la rangée 8 points de « - » en bord de carte n'est pas utilisée dans ce cas.



5.2.2 Sélection de la tension d'alimentation des sorties S1 à S8

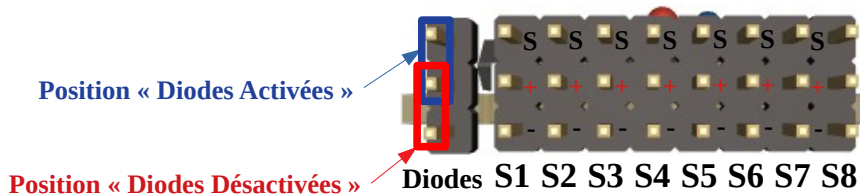


Il est possible d'alimenter les sorties **S1** à **S8** (fourniture du **+**) à partir :

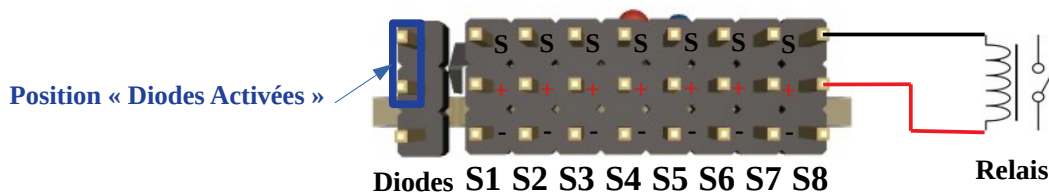
1. De la tension du **+VRx** fournie par le récepteur (Attention à la consommation sur les sorties !)
→ Cavalier de sélection alimentation sur « **VRx** »
2. D'une tension externe **VExt** (de **5V** à **24V**) appliquée sur le connecteur 2 points en bas à gauche de la carte (**ATTENTION** : ne pas dépasser 6V si au moins un servo est connecté à Sx)
→ Cavalier de sélection alimentation sur « **VExt** »

5.2.3 Diodes de roue libre

Si les « utilisations » connectées sur les sorties sont selfiques (ex :relais), il faut connecter les diodes internes de roue libre, sinon il y a risque de détruire l'ULN2803.



5.2.4 Commande directe de relais avec diode intégrée à l'ULN2803



Lorsque des charges selfiques (tels que relais ou petits moteurs) sont alimentés par les sorties S1 à S8, il est nécessaire d'activer les diodes dites « de roue libre ».

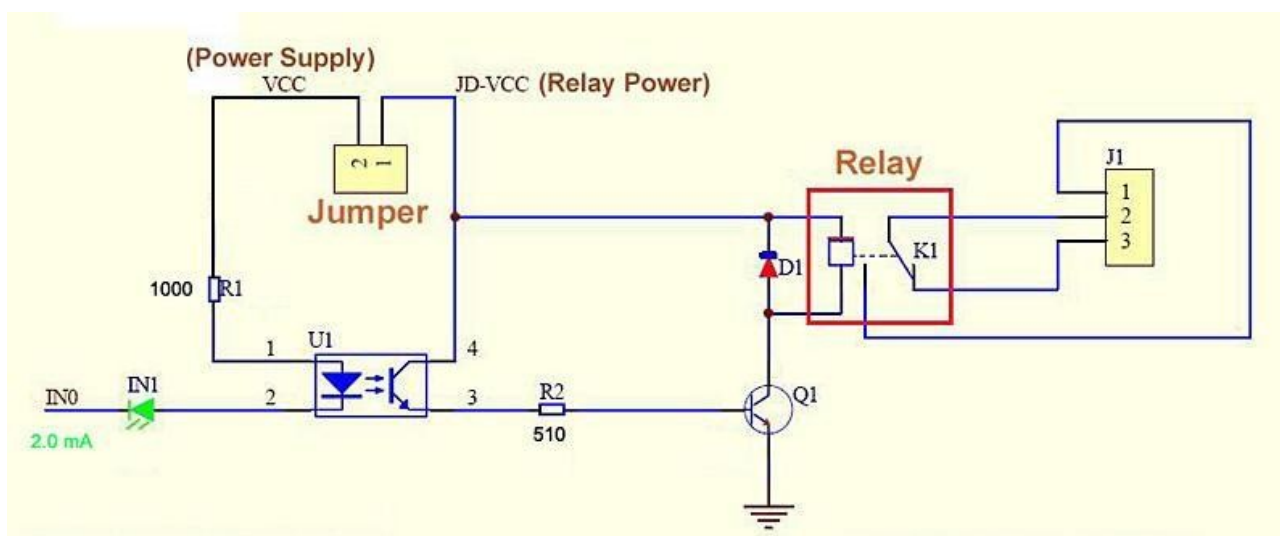
ATTENTION : ces diodes sont activées/désactivées pour les 8 sorties à la fois.

5.2.5 Commande de relais opto-isolés

Il existe des « modules relais » très bon marché souvent appelés « Arduino Relay Module ». Ces modules intègrent un opto-coupleur permettant une isolation totale entre la tension de commande et la tension d'alimentation des bobines des relais.



Le schéma équivalent d'une voie de ces modules est donné ci-dessous :

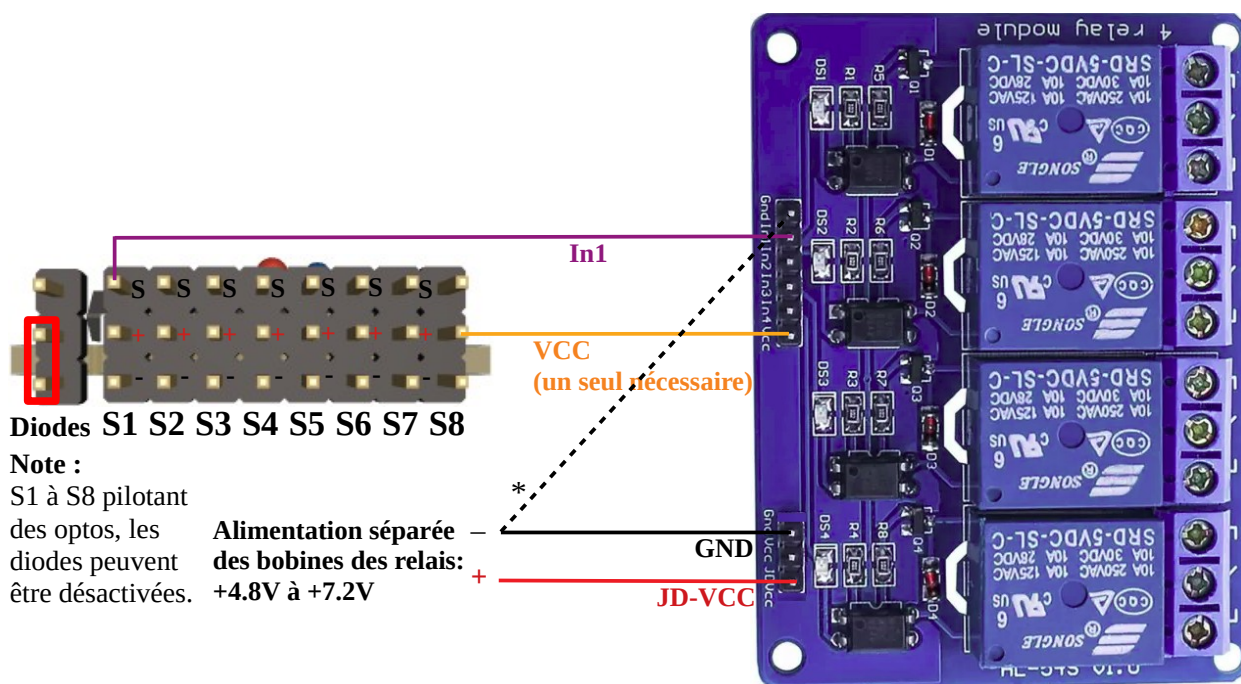


- ◆ **VCC** est la tension de commande des opto-coupleurs : accessible sur le connecteur de bord de carte
- ◆ **JD-VCC** est la tension de commande des bobines des relais : accessible sur un des points du connecteur « Jumper »

Note importante :

Pour bénéficier d'une isolation totale entre VCC et JD-VCC, le Jumper jaune (cavalier) doit être retiré.

5.2.6 Montage conseillé pour relais 5V opto-isolés



- ◆ Les opto-coupleurs sont alimentés par la tension **VRx** du récepteur (conso : quelques x10 mA)
- ◆ VCC de la carte relais est relié à un des points centraux de S1 à S8
- ◆ In1 de la carte relais est reliée à la sortie S1
- ◆ In2 de la carte relais est reliée à la sortie S2, etc
- ◆ Les bobines des relais sont alimentées par une alimentation séparée (+4.8V à +7.2V)

→ Les alimentations étant complètement isolées, la tension du récepteur ne sera pas perturbée lors des commutations des relais : aucun risque de perdre le contrôle RC.

* : sur certains modèles de « module relais », il n'y a pas de broche **GND** à proximité du **JD-VCC**. Dans ce cas, utiliser la broche **GND** à proximité immédiate de la broche **In1** sur l'autre connecteur.

5.3 Configuration pour MS8-Xany V2

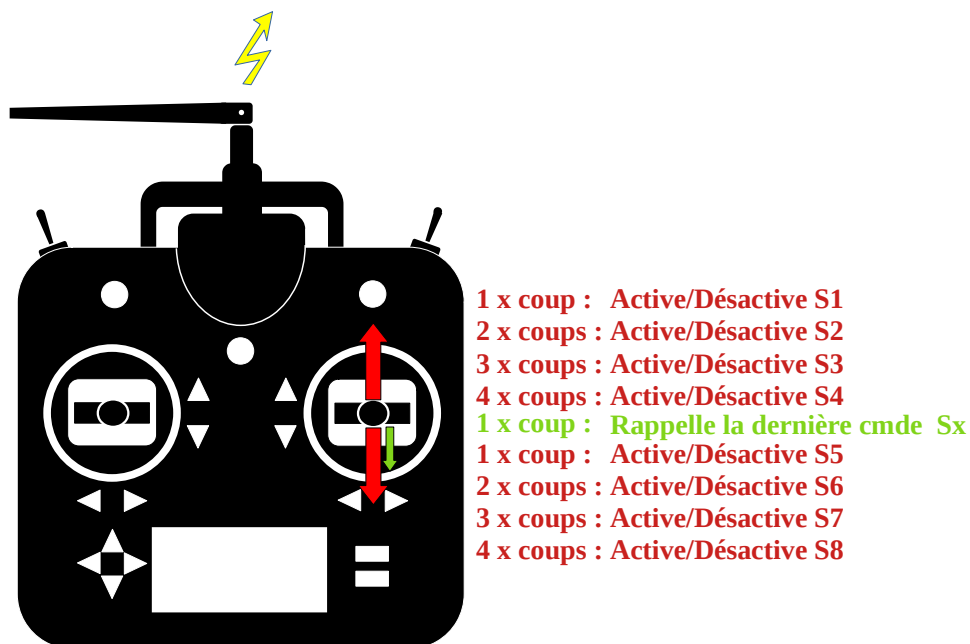
5.3.1 Configuration du mode de MS8-Xany V2 pour l'EKMFA

EKMFA signifie **E**in Kanal **M**ulti-**F**onktion **A**uswahl : un canal pour sélection de fonctions multiples.

Pour commander indépendamment plusieurs fonctions depuis l'émetteur, il suffit de donner des coups de manches. Le nombre de coups de manche, ainsi que la direction des coups de manche sélectionne la sortie à activer/désactiver.

Le comptage du nombre de coups de manche étant fastidieux, les 8 sorties ont été divisées en 2 :

- **Un à quatre** coups de manche dans **une** direction activent/désactivent les sorties **S1 à S4**
- **Un à quatre** coups de manche dans **l'autre** direction activent/désactivent les sorties **S5 à S8**



Astuce : un maintient à mi-course vers le bas (flèche verte) pendant environ 1/4 de seconde, complémente la dernière sortie commandée.

Cette manière de commander les sorties est universelle et fonctionne sur tous les ensembles RC et ne nécessite aucune modification de l'émetteur, ni de module additionnel.

De plus, avec **MS8-Xany V2**, côté récepteur, il est possible d'utiliser **EKMFA**, sur une interface RC de type PWM, CPPM ou encore SBUS !

PWM et **EKMFA** sont l'interface RC et le protocole par défaut de **MS8-Xany V2**.

A la toute première mise sous tension, **MS8-Xany V2** est déjà dans ce mode.

Si **MS8-Xany V2** a été précédemment configuré dans un autre mode, il est évidemment possible de revenir en **EKMFA** à l'aide des commandes suivantes.

Pour (re-)passer le protocole en **EKMFA** :

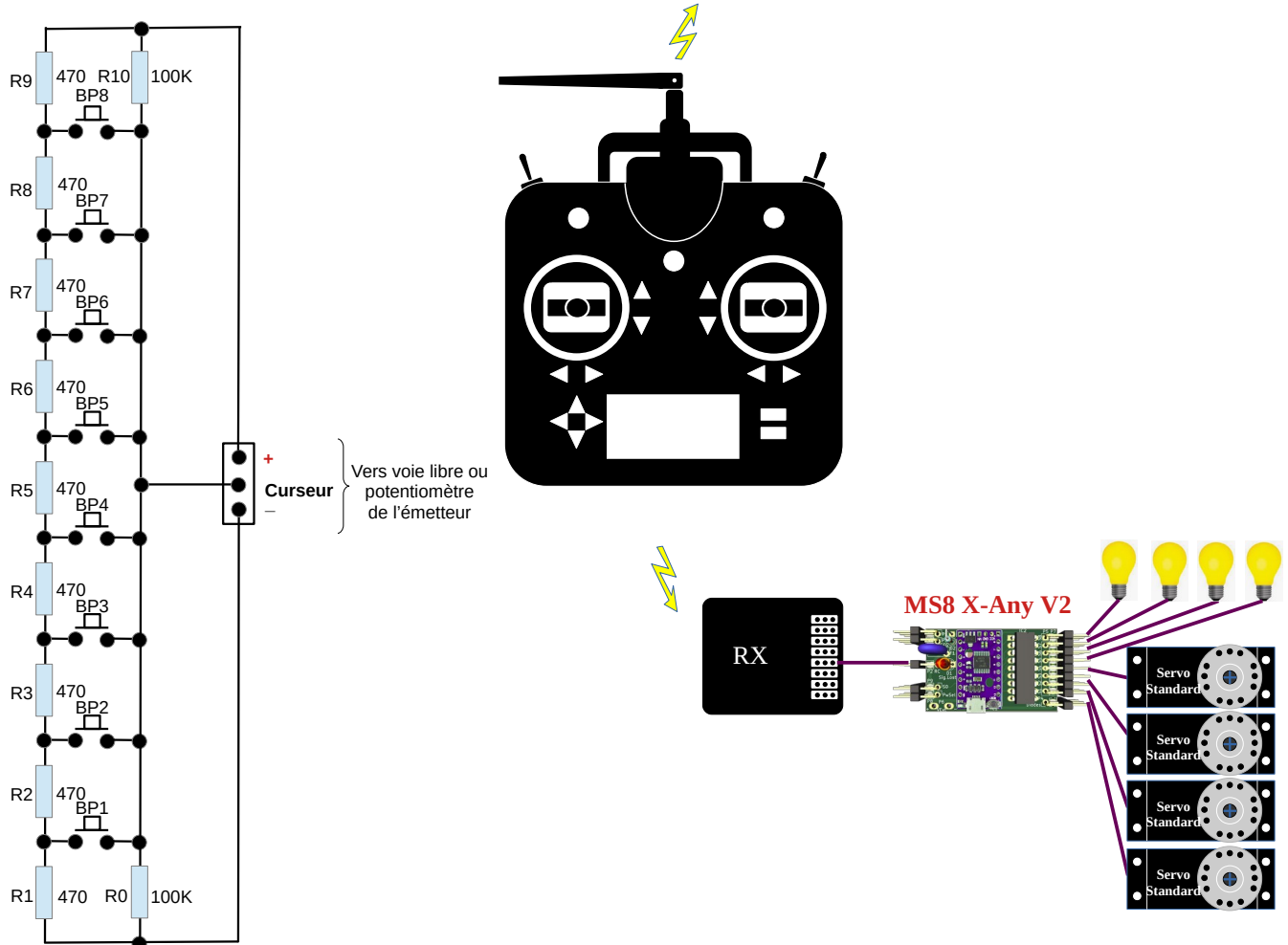
- ◆ **P=E** → Le **P**rotocol de commande est **E**KMFA

Les combinaisons d'Interface de commande possibles sont les suivantes :

- ◆ **I=P** → **MS8-Xany V2** se connecte sur une sortie « voie » **P**WM d'un récepteur RC
- ◆ **I=C5** → **MS8-Xany V2** se connecte sur une sortie **C**PPM d'un récepteur RC et utilise la voie N°5 du train **C**PPM
- ◆ **I=S5** → **MS8-Xany V2** se connecte sur une sortie **S**BUS d'un récepteur RC et utilise la voie N°5 de la trame **S**BUS

5.3.2 Configuration du mode de MS8-Xany V2 pour Boutons-Poussoirs

Dans ce mode de fonctionnement, côté émetteur, un « clavier » comprenant 8 boutons-poussoirs se connecte sur une voie libre ou se substitue à un potentiomètre de manche.



Chaque appui sur un bouton-poussoir génère une tension différente au niveau du point noté « curseur », l'émetteur va donc générer une impulsion de largeur différente qui va se retrouver côté réception. Bien que chaque sortie puisse être commandée indépendamment des autres, les appuis multiples sur le clavier ne sont pas supportés et ne conduiront pas aux commandes attendues !

Pour passer le protocole en mode **Boutons-Poussoirs** :

- ◆ **P=P** → Le **P**rotocol de commande est Boutons-**P**oussoirs

Les combinaisons d'Interface de commande possibles sont les suivantes :

- ◆ **I=P** → **MS8-Xany V2** se connecte sur une sortie « voie » **PWM** d'un récepteur RC
- ◆ **I=C3** → **MS8-Xany V2** se connecte sur une sortie **CPPM** d'un récepteur RC et utilise la voie N°3 du train **CPPM**
- ◆ **I=S3** → **MS8-Xany V2** se connecte sur une sortie **SBUS** d'un récepteur RC et utilise la voie N°3 de la trame **SBUS**

Calibration en mode Boutons-Poussoirs :

MS8-Xany V2 doit connaître la largeur d'impulsion associée à chaque appui sur les boutons-poussoirs BP1 à BP8 : il y a donc une phase de calibration initiale à réaliser (une seule fois).

Pour réaliser cette opération, il faut connecter un Terminal Série au décodeur **MS8-Xany V2**.

Pour passer en mode calibration pour les boutons-poussoirs, envoyer la commande suivante :

◆ **B=C** → Passe le protocole **B**outons-Poussoirs en mode **C**alibration

L'affichage devient :

BP1 :

Tout maintenant le Bouton-Poussoir N°1 appuyé, appuyer sur la touche Entrée du clavier du Terminal série.

La largeur d'impulsion mesurée pour le BP1 va alors être affichée (et automatiquement mémorisée) et BP2 va être affiché :

BP1 :

0988

BP2 :

Tout maintenant le Bouton-Poussoir N°2 appuyé, appuyer sur la touche Entrée du clavier du Terminal série.

Répéter l'opération jusqu'au BP8.

Une fois la largeur d'impulsion associée au BP8 affichée, la calibration est terminée.

Il est possible d'afficher les 8 largeurs d'impulsion associée aux BP1 à BP8 par la commande :

B?

B=0988;1100;1228;1320;1672;1768;1888;2016

En cas de fausse manipulation ou d'erreur, il est évidemment possible de refaire une calibration en renvoyant la commande :

◆ **B=C** → Passe le protocole **B**outons-Poussoirs en mode **C**alibration

5.3.3 Configuration de l'émetteur OpenAVRc

Reportez vous au manuel d'**OpenAVRc** pour configurer l'instance **X-Any** avec les paramètres suivants :

1. Le N° de voie doit correspondre au N° de voie sur laquelle le décodeur **MS8-Xany V2** sera connecté sur le récepteur
2. Le nombre de répétition sera en premier lieu réglé sur 3 (dès que ça fonctionnera, il sera possible de réduire cette valeur afin d'atteindre la réactivité maximale autorisée par votre ensemble HF).
3. Configurer « Sw. » sur Sw.8 : cela va transmettre l'état de 8 contacts
4. Si le servo proportionnel sera utilisé sur **MS8-Xany V2**, sélectionner un des choix proposés par « Prop. », cela va ajouter la transmission de la valeur du proportionnelle.



5.3.4 Configuration du mode de MS8-Xany V2 pour OpenAVRc

L'émetteur **OpenAVRc** intègre nativement le protocole **RCUL/X-Any** qui permet de passer les commandes « au dessus » de n'importe quelle Interface RC de commande : PWM, CPPM, SBUS.

Comme **X-Any** est un protocole entièrement numérique, il peut passer la commande pour piloter une sortie proportionnelle (S0) en même temps que la commande des 8 sorties (S1 à S8).

Les combinaisons d'Interface RC de commande possibles sont les suivantes :

- ◆ **I=P** → **MS8-Xany V2** se connecte sur une sortie « voie » **PWM** d'un récepteur RC
- ◆ **I=C5** → **MS8-Xany V2** se connecte sur une sortie **CPPM** d'un récepteur RC et utilise la voie N°5 du train CPPM
- ◆ **I=S5** → **MS8-Xany V2** se connecte sur une sortie **SBUS** d'un récepteur RC et utilise la voie N°5 de la trame SBUS

Le protocole **RCUL/X-Any** se configure comme suit :

- ◆ **P=R; N/F** → **MS8-Xany V2** utilise le Protocole **RCUL/X-Any** (N : Non filtré, F : Filtré)

5.3.5 Configuration des émetteurs OpenTx et EdgeTx

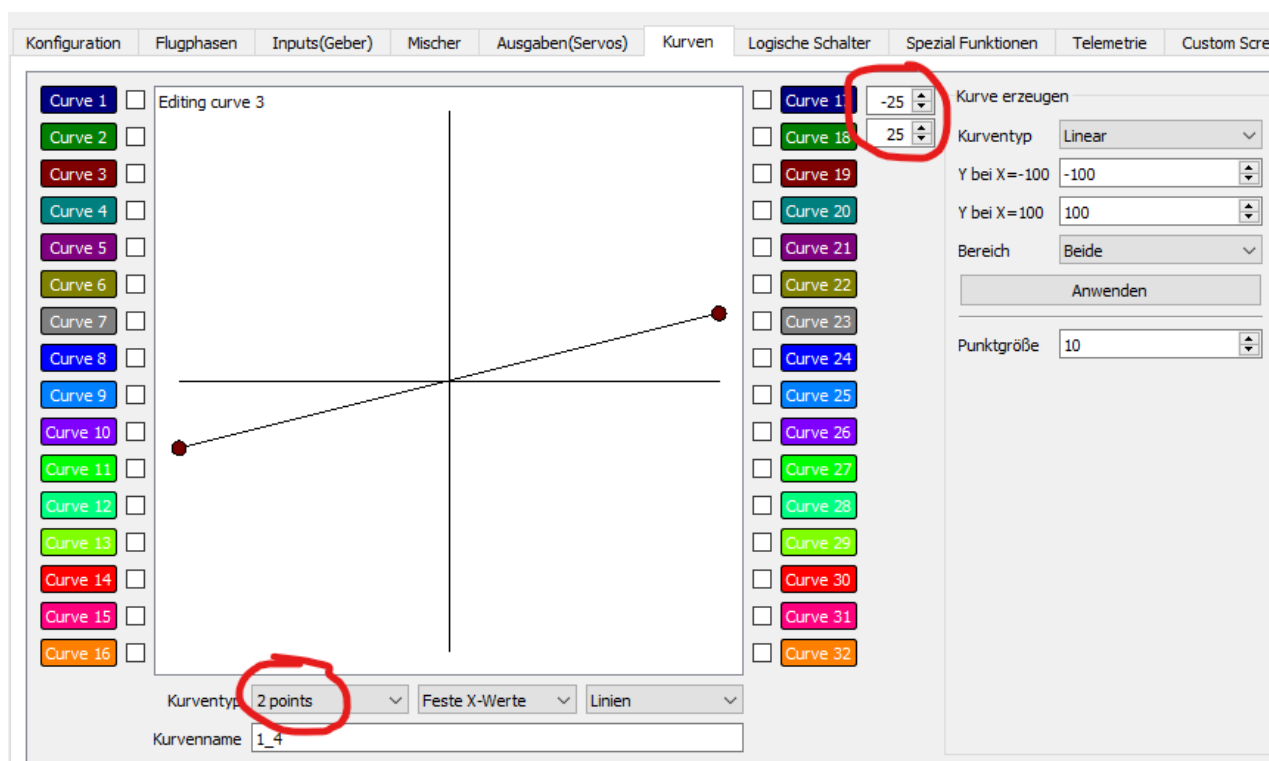
Pour commander **MS8-Xany V2** à partir de 8 interrupteurs natifs (ou logiques) des émetteurs **OpenTx** et **EdgeTx**, il est nécessaire de configurer 9 mélangeurs :

- 1 mélangeur par interrupteur → 8 mélangeurs
- 1 dernier mélangeur pour ramener les largeurs d'impulsion RC dans de la plage 1000 à 2000 µs

Chaque interrupteur représentant un bit de pondération dans le mot de 8 bits transmis sous forme de largeur d'impulsion dans la voie SBUS dédiée à **MS8-Xany V2** (voie N°5 dans l'exemple ci-dessous).

Les configurations suivantes pour **Compagnon** d'**OpenTx** et d'**EdgeTx** proviennent de :
<https://github.com/Tiefflieger68/SBUS-Switch/blob/main/docs/SBUS-Switch%20V2.4.d.pdf>

Captures d'écran d'**openTX Companion** pour la programmation de l'émetteur (**OpenTx** Version 2.4.x)



Courbe (ici Courbe 3)

Une courbe doit être définie afin de pouvoir traiter des valeurs inférieures à 1 % dans les mélangeurs.

Dans le menu "Configurations", "Chemins étendus" doit être activé. (Pas nécessaire pour le mode de compatibilité)

Selon le mode (Normal ou de Compatibilité), la sortie des 8 canaux utilisés dans la voie SBUS doit être ajustée en conséquence.

Différentes valeurs sont à utiliser :

	FrSky (Normal)	FrSky (Compatibilité)	FlySky (Compatibilité)
Sortie Min (%)	-125,0	-99,9	-102,4
Sortie Max (%)	+125,0	+100	+102,4
Neutre (µs)	1520	1520	1500

Mélangeur:

Le premier mélangeur est le mélangeur commun. Aucun interrupteur ne lui est affecté.

Les autres mélangeurs correspondent aux sorties 1 à 8.

Un commutateur physique ou logique doit être attribué dans chaque cas. La sortie de commutation correspondante est alors commutée avec cet interrupteur.

5.3.6 Configuration du mode de MS8-Xany V2 pour OpenTx et EdgeTx

Sauf **EKMFA**, la seule interface RC supportée avec **OpenTx** et **EdgeTx** est l'interface SBUS avec le protocole « Binaire » :

- ◆ **I=S5** → **MS8-Xany V2** se connecte sur une sortie **SBUS** d'un récepteur RC et utilise la voie N°5 de la trame SBUS
- ◆ **P=B** → Protocole « **Binaire** » en SBUS

Le protocole Binaire en SBUS permet l'utilisation des interrupteurs physiques et logiques, ce qui est beaucoup plus pratique que la commande par coups de manche de **l'EKMFA**.

5.3.7 Configuration du mode de MS8-Xany V2 pour Robbe-Futaba

MS8-Xany V2 est en mesure de se substituer à un décodeur MultiSwitch 8 sorties N° F1513 de **Robbe-Futaba**.

Si un ensemble RC **Robbe-Futaba** fonctionne avec le décodeur MultiSwitch 8 sorties N° F1513, il fonctionnera également avec **MS8-Xany V2**.

Les 2 seuls modes supportés avec **Robbe-Futaba** sont les modes d'interface RC PWM et CPPM avec le protocole «Futaba» :

- ◆ **I=P** → **MS8-Xany V2** se connecte sur une sortie « voie » **PWM** d'un récepteur RC
- ◆ **I=C7** → **MS8-Xany V2** se connecte sur une sortie **CPPM** d'un récepteur RC et utilise la voie N°7 du train CPPM
- ◆ **P=F** → Le **Protocole «Futaba»** en PWM ou en CPPM

5.4 Mode avancé/Commande de servos

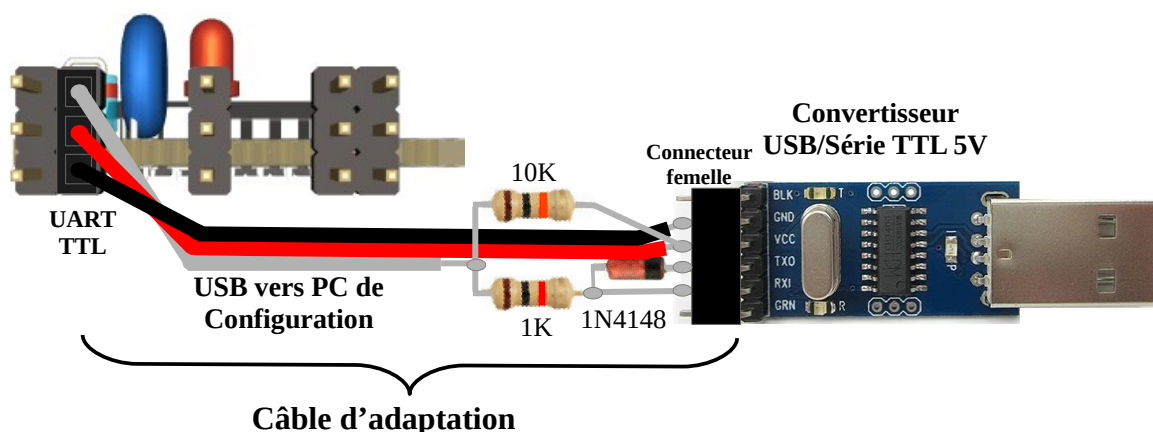
Le décodeur **MS8-Xany V2** dispose d'un accès pour les configurations avancées : un port série TTL.

C'est cet accès série qui va permettre le paramétrage de **MS8-Xany V2** pour permettre, par exemple, la sélection de l'interface RC, du protocole utilisé et l'utilisation de servos connectés aux sorties **S1** à **S8**. La liste des commandes supportées est détaillée en page suivante.

Dans ce cas, la tension « + » de sortie devra impérativement être compatible des servos !

5.4.1 Utilisation du port série de MS8-Xany V2

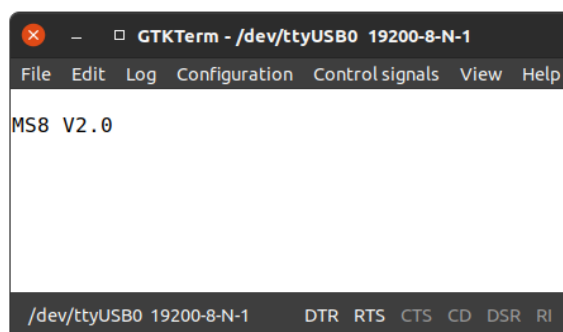
Pour accéder au port série de **MS8-Xany V2**, il faut un convertisseur USB/Série TTL 5V par exemple de type « FTDI » et un câble adaptateur réalisé à l'aide d'un cordon de servo, de 2 résistances (10K et 1K), d'une diode (1N4148) et d'un connecteur femelle :



1. Connecter le convertisseur USB/Série TTL sur le port USB d'un PC pour le côté USB
2. Connecter le câble d'adaptation au connecteur UART TTL de **MS8-Xany V2**
3. Sur le PC, ouvrir un Terminal série, par exemple, PuTTY, TeraTerm, HyperTerminal, GtTerm, 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.

4. Appuyer sur la touche « Entrée » de votre clavier, le message « MS8 V2.X » doit apparaître sur le Terminal série comme illustré ci-dessous.



Exemple de connexion avec le Terminal GtTerm sous Linux

Note :

Il est possible de configurer **MS8-Xany V2** pendant que celui-ci est connecté au récepteur RC.

Cependant, en SBUS, les configurations peuvent avoir du mal à passer : déconnecter le récepteur.

5.4.2 Les messages de commande de MS8-Xany V2

La liste des messages supportés par **MS8-Xany V2** est donnée dans la table suivante :

← Commande/ → Réponse	Action	Remarque
← Enter → MS8 V2.x	Retourne le nom et la version du Firmware de MS8-Xany V2	
← I? → I=Interface[v]	<p>Retourne l'Interface RC utilisée par MS8-Xany V2.</p> <p>Interfaces possibles :</p> <ul style="list-style-type: none"> - P: PWM. MS8-Xany V2 est connecté sur la voie PWM du récepteur définie dans le menu X-Any d'OpenAVRc. - Cv: CPPM voie v. MS8-Xany V2 est connecté sur la sortie CPPM du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Sv: SBUS voie v. MS8-Xany V2 est connecté sur la sortie SBUS du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. Le SBUS est le seul mode d'interface supporté par OpenTx et EdgeTx. 	La voie v doit être dans la plage 1 à 16.
I=Interface[v]	Définit l'Interface RC utilisée par MS8-Xany V2 . Idem, ci-dessus.	
← P? → P=R;Filtrage ou → P=F/B/E/P	<p>Retourne le Protocole utilisé sur l'interface RC :</p> <ul style="list-style-type: none"> - R;Filtrage : Protocole RCUL/X-Any. Il faut préciser si Non filtré ou si Filtré. - F : Protocole Futaba MultiSwich 8 - B : Binaire (Valeur 8 bits de 0 à 255 codée dans l'impulsion RC) - E : Protocole EKMFA (le nombre de coup de manche active/désactive une sortie) - P : Protocole Boutons-Poussoirs 	<ul style="list-style-type: none"> - Utiliser le protocole RCUL/X-Any pour OpenAVRc <p>Le paramètre de filtrage vaut :</p> <ul style="list-style-type: none"> - N : Non filtré - F : Filtré <p>Dans la plupart des cas, ce paramètre doit être positionné à N (Non filtré). Un récepteur RC effectue un filtrage quand il considère une grande variation d'impulsion RC comme probablement anormale. Par exemple, si l'impulsion RC passe brutalement de 1000µs à 2000µs, l'impulsion en sortie du récepteur va passer de 1000µs à 1500µs avant d'atteindre 2000µs.</p>

		<p>C'est le comportement des récepteurs dits « DSP » ou encore des récepteurs PCM Futaba.</p> <ul style="list-style-type: none"> - Utiliser le protocole Futaba pour remplacement d'un décodeur MultiSwitch 8 sorties N°F1513 de Robbe-Futaba - Utiliser le protocole Binaire pour OpenTx et pour EdgeTx - Pour tous les autres ensembles RC, utiliser le protocole EKMFA
← P=R;Filtrage/F/B/E/P → P	Définit le Protocole utilisé sur l'interface RC (idem ci-dessus)	
← SR? → SR=Y/N	Retourne l'inversion de servo pour la sortie S0. Y=Inversé, N=Non inversé	
← SR=Y/N → SR	Définit l'inversion de servo pour la sortie S0. Y=Inversé, N=Non inversé	
← S0? → S0=Pos:OffsetPas4us	Retourne la position courante en μ s et la commande OffsetPas4us qui est le nombre de pas de 4 μ s (valeur entre 0 et 255) à ajouter à 988 pour avoir la largeur d'impulsion en μ s pour le servo proportionnel	<p>Si SR=N : Largeur d'impulsion(us) = 988 + (OffsetPas4us x 4)</p> <p>Si SR=Y : Largeur d'impulsion(us) = 988 + [(255 - OffsetPas4us) x 4]</p>
← S0=Pos → S0	Définit la position en μ s pour le servo proportionnel	Renvoie ERR, si valeur non comprise entre 988 et 2008
← Sx? → Sx=D;M:C ou → Sx=S;M;Pos0;Pos1;Dur:C	<p>Si x est compris entre 1 et 8, retourne la configuration de la sortie N°x ainsi que l'état « C » de la Commande courante associée (0 ou 1)</p> <ul style="list-style-type: none"> - Si la sortie est configurée en sortie numérique MultiSwitch, la réponse est : Sx=D;M:C D=Digital (Numérique) M=Mode de commande (N : Normal, P : ImPulsionnel) - Si la sortie est configurée en sortie Servo, la réponse est : Sx=S;M;Pos0;Pos1;Dur:C avec S=Servo M=Mode de commande (N : Normal, P : ImPulsionnel) Pos0=la position en μs pour une commande à 0, Pos1=la position en μs pour une commande à 1, et Dur=la durée du mouvement du servo entre Pos0 et Pos1 	<p>Renvoie ERR, si</p> <ul style="list-style-type: none"> - Valeur x non comprise entre 0 et 8 - Pos0 ou Pos1 < 600 - Pos0 ou Pos1 > 2400 <p>Exemple de réponses :</p> <p>S1=D;N:0</p> <p>S2=D;P:1</p> <p>S3=S;N;1000;2000;5000:0</p> <p>S4=S;P;2300;600;8500:1</p>
← Sx=D;M → Sx	Définit la sortie x comme étant de type numérique (Digitale) et commandée en mode M (M=N : Mode Normal, M=P : Mode imPulsionnel)	<p>Exemple de commandes :</p> <p>S1=D;N</p> <p>S2=D;P</p>

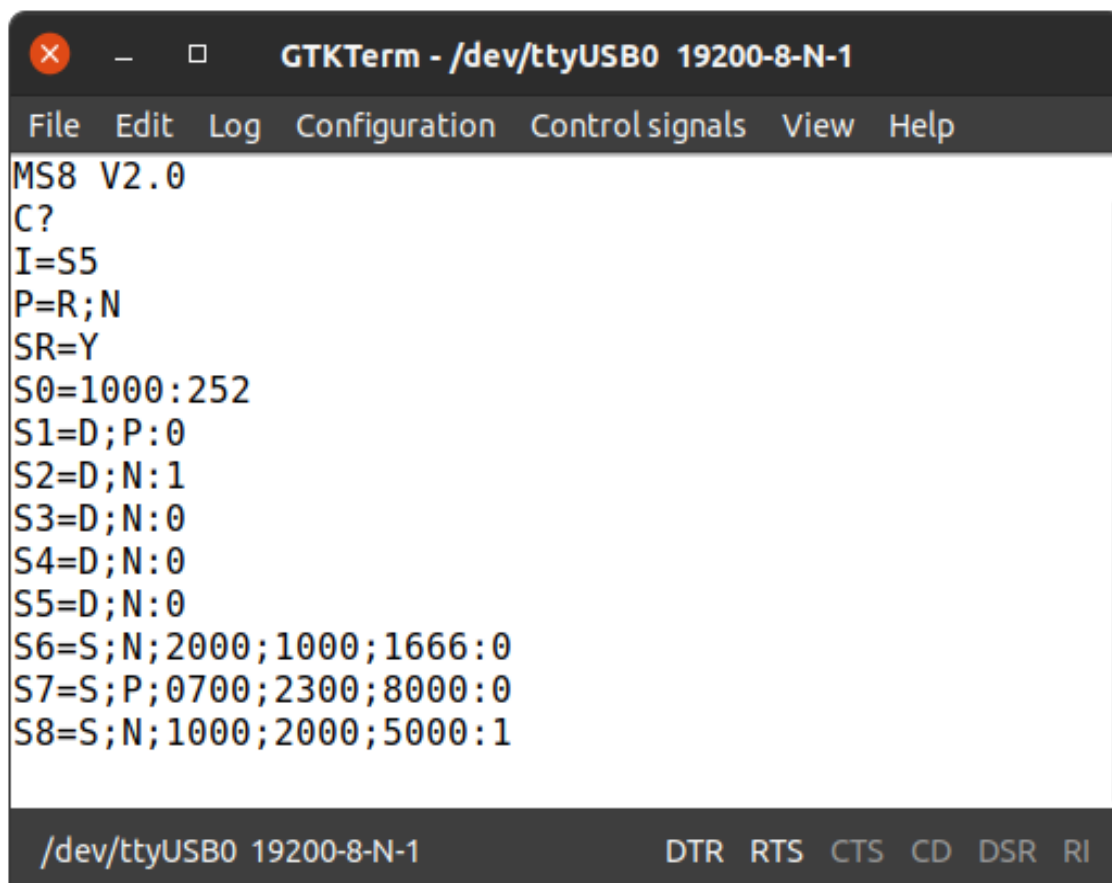
← Sx=S;M;Pos0;Pos1;Dur → Sx	Définit la sortie x comme étant de type Servo commandée en mode M (M=N : Mode Normal, M=P : Mode imPulsionnel) avec une position à Pos0 µs pour une commande à 0, une position à Pos1 µs pour une commande à 1, et une durée de mouvement à Dur ms	La valeur « Dur » en ms est recalculée en interne par MS8-Xany V2 en tenant compte des différentes résolutions/limitations et peut être différente à la relecture.
← Sx=C → Sx	Si x est compris entre 1 et 8, « C » définit la Commande (0 ou 1) pour la sortie x, que celle-ci soit de type Digital ou Servo	Très pratique pour les tests par l'accès série sans RC.
← B=C → BPx:	Entre dans la mode de calibration des boutons-poussoirs	
← B? → B=xxxx;xxxx;...;xxxx	Retourne les valeurs de calibration (µs) des 8 boutons-poussoirs	Les valeurs vont environ de 1000 à 2000µs
← C? → La conf complète	Retourne la Configuration complète de MS8-Xany V2	L'état des commandes est présent en fin de ligne après le ':'
← A? → La conf complète	Retourne la Configuration complète pour archivage	Ne contient pas les états en fin de ligne après le ':'

5.4.3 Exemple de configuration réelle

Dans l'exemple ci-dessous :

- **MS8-Xany V2** est connecté sur l'interface SBUS d'un récepteur et utilise la voie N°5
- Le protocole utilisé sur SBUS est RCUL/X-Any sans filtrage
- La sortie proportionnelle (S0) est inversée
- La largeur d'impulsion pour le servo/ESC connecté sur la sortie proportionnelle S0 vaut :
$$988 + [(255 - \mathbf{252}) \times 4] = \mathbf{1000} \mu\text{s} \quad ((255 - \mathbf{252}) \text{ et non directement } \mathbf{252} \text{ car inversé})$$

→ la commande (= le nombre de pas de 4 μs à ajouter à 988 μs) serait de 252 pour la largeur d'impulsion de 1000 μs
- Les sorties S1, S2, S3, S4 et S5 sont de type Digital (MultiSwitch), leurs commandes valent respectivement 0, 1, 0, 0 et 0. La sortie S2 est commandée en imPulsionnel.
- Les sorties S6, S7 et S8 sont de type Servo, leurs commandes valent respectivement 0, 0 et 1. La sortie 7 est commandée en imPulsionnel.

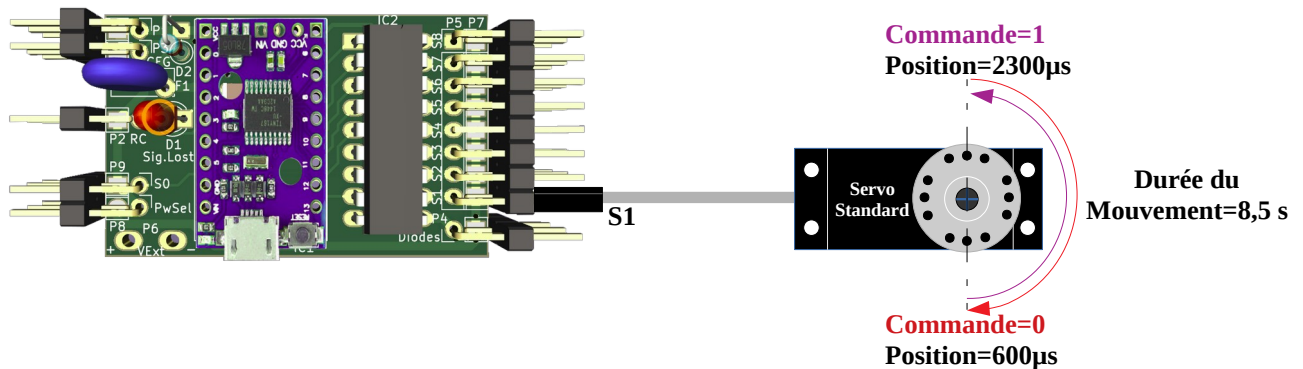


```
GTKTerm - /dev/ttyUSB0 19200-8-N-1
File Edit Log Configuration Control signals View Help
MS8 V2.0
C?
I=S5
P=R;N
SR=Y
S0=1000:252
S1=D;P:0
S2=D;N:1
S3=D;N:0
S4=D;N:0
S5=D;N:0
S6=S;N;2000;1000;1666:0
S7=S;P;0700;2300;8000:0
S8=S;N;1000;2000;5000:1
/dev/ttyUSB0 19200-8-N-1 DTR RTS CTS CD DSR RI
```

1. Mouvement de servo dans le sens contraire des aiguilles d'une montre

Si le contact N°1 côté émetteur est fermé (**Commande=1**), l'impulsion du servo connecté à la sortie N°1 va de 600 μ s à 2300 μ s (soit un mouvement d'environ 180°) en 8,5 secondes, et va de 2300 μ s à 600 μ s (soit un mouvement d'environ 180°) en 8,5 secondes si le contact N°1 côté émetteur est ouvert (**Commande=0**).

Ex : **S1=S;N;600;2400;8500**

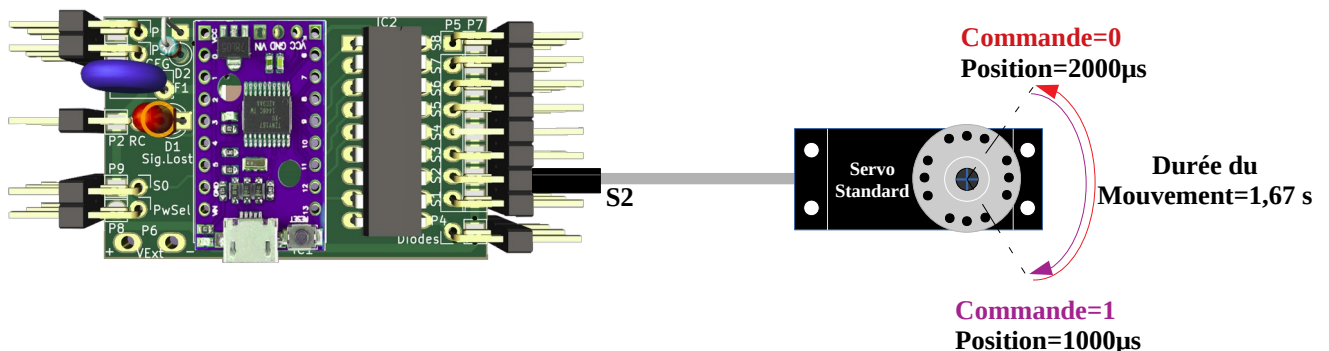


2. Mouvement de servo dans le sens des aiguilles d'une montre

Avec **MS8-Xany V2**, il est possible d'avoir un mouvement dans le sens opposé : il suffit de permuter les positions extrêmes (Pos0 et Pos1).

Si le contact N°2 côté émetteur est fermé (**Commande=1**), l'impulsion du servo connecté à la sortie N°2 va de 2000 μ s à 1000 μ s (soit un mouvement d'environ 100°) en 1,67 secondes si la commande côté émetteur vaut 1, et de 1000 μ s à 2000 μ s (soit un mouvement d'environ 100°) en 1,67 secondes si le contact N°2 côté émetteur est ouvert (**Commande=0**).

Ex : **S2=S;N;2000;1000;1666**



5.4.4 Commande en mode Normal et en mode imPulsionnel

Que la sortie soit configurée en type **Digital** (Multiswitch) ou en type **Servo**, il faut définir le mode commande :

- **Sx=D;M** (avec M valant **N** pour **Normal** ou **P** pour im**P**ulsionnel)
- **Sx=S;M;1000;2000;5000** (avec M valant **N** pour **Normal** ou **P** pour im**P**ulsionnel)

Supposons que l'on ait la configuration suivante et que, côté émetteur, S1 et S2 soient commandés par des contacts de boutons-poussoirs :

S1=D;N pour piloter une corne de brume

S2=D;P pour piloter une lumière

Dans les 2 cas, les sorties sont de type **Digital** (Multiswitch), mais **S1** est pilotée en mode **Normal**, tandis que **S2** est pilotée en mode im**P**ulsionnel.

1. Cas du mode Normal :

Tant que l'on appuie sur le contact C1, la sortie S1 est à 1 et la corne de brume fonctionne.

Dès que l'on relâche le contact C1, la sortie S1 passe à 0 et la corne de brume s'arrête.

En résumé, la sortie S1 suit l'état du contact C1.

2. Cas du mode imPulsionnel :

Tant que l'on appuie sur le contact C2, la sortie S2 est à 1 et la lumière fonctionne.

Dès que l'on relâche le contact C2, la sortie S2 reste à 1 et la lumière continue à fonctionner.

Pour éteindre la lumière, il faut appuyer une nouvelle fois sur le contact C2.

En résumé, une impulsion sur le contact C2 active la sortie S2, une nouvelle impulsion sur le contact C2 désactive la sortie S2 et ainsi de suite.

Notes :

1. Si la sortie est de type **Servo** en mode im**P**ulsionnel, une impulsion sur le bouton-poussoir, provoque le mouvement du servo jusqu'à la position N°1 et une seconde impulsion provoque le mouvement du servo jusqu'à la position N°2.

Ex : **S3=S;P;1000;2000;5000**

2. En protocole **EKMFA**, même si le mode de la sortie est configuré en **Normal**, elle est toujours traitée comme étant configurée en im**P**ulsionnel.

6 TABLES DES COMBINAISONS DE COMMANDE SUPPORTEES

6.1 Tout ensemble RC

		Protocole de commande				
		EKMFA	Boutons-Poussoirs	RCUL/X-Any	Binaire	Futaba
Interface RC	PWM	✓	✓	✗	✗	✗
	CPPM	✓	✓	✗	✗	✗
	SBUS	✓	✓	✗	✗	✗

6.2 Ensemble OpenAVRc

		Protocole de commande				
		EKMFA	Boutons-Poussoirs	RCUL/X-Any	Binaire	Futaba
Interface RC	PWM	✓	✓	✓	?	✗
	CPPM	✓	✓	✓	?	✗
	SBUS	✓	✓	✓	?	✗

6.3 Ensembles OpenTx/EdgeTx

		Protocole de commande				
		EKMFA	Boutons-Poussoirs	RCUL/X-Any	Binaire	Futaba
Interface RC	PWM	✓	✓	✗	✗	✗
	CPPM	✓	✓	✗	✗	✗
	SBUS	✓	✓	✗	✓	✗

6.4 Ensemble Futaba supportant le MultiSwitch 8 sorties Futaba N° F1513

		Protocole de commande				
		EKMFA	Boutons-Poussoirs	RCUL/X-Any	Binaire	Futaba
Interface RC	PWM	✓	✓	✗	✗	✓
	CPPM	✓	✓	✗	✗	✓
	SBUS	✓	✓	✗	✗	✗