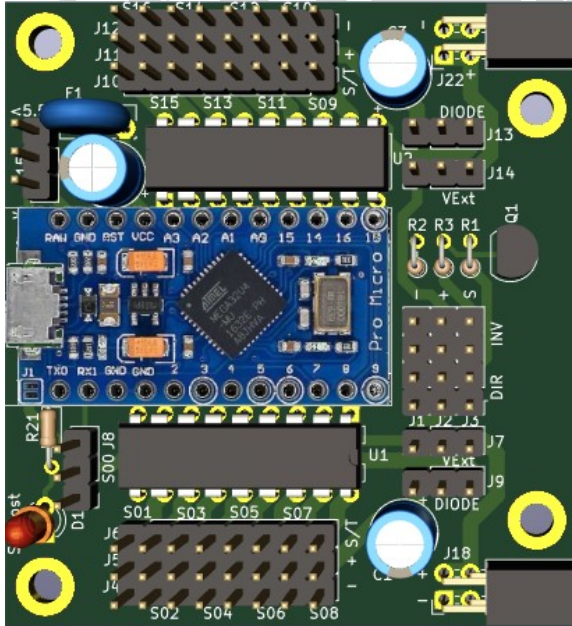


# MS16-Xany

**Bien plus qu'un  
simple MultiSwitch  
à 16 sorties pour  
OpenAVRc**



## Manuel Utilisateur MultiSwitch MS16-Xany



Copyright OpenAVRc 2021

V3

## 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 MS16-XANY.....	5
2.1	Vue d'ensemble.....	5
2.2	Spécifications du décodeur MS16-Xany.....	5
3	SCHEMA DU DECODEUR MS16-XANY.....	7
4	REALISATION DU DECODEUR MS16-XANY.....	8
4.1	Circuit imprimé.....	8
4.2	Arduino Pro Micro.....	8
4.3	Chargement du Firmware dans l'arduino Pro Micro.....	8
4.3.1	Utilitaire XLoader modifié.....	8
5	UTILISATION.....	9
5.1	Connexion au récepteur.....	9
5.2	Mode standard MultiSwitch/Commutation Tout-Ou-Rien.....	9
5.2.1	Câblage des « utilisations » Tout-Ou-Rien sur les sorties.....	9
5.2.2	Sélection de la tension d'alimentation des sorties S01 à S08.....	10
5.2.3	Alimentations externes VExt1 et VExt2.....	10
5.2.4	Modules optionnels d'alimentation VExt1 et VExt2.....	11
5.2.5	Module optionnel d'alimentation VExt1.....	11
5.2.6	Module optionnel d'alimentation VExt2.....	11
5.2.7	Diodes de roue libre.....	12
5.2.8	Commande directe de relais avec diode intégrée à l'ULN2803.....	12
5.2.9	Commande de relais opto-isolés.....	12
5.2.10	Montage conseillé pour relais 5V opto-isolés.....	13
5.3	Configuration X-Any côté émetteur OpenAVRc.....	14
5.4	Mode avancé/Commande de servos.....	15
5.4.1	Utilisation du port série USB de MS16-Xany.....	15
5.4.2	Les messages de commande de MS16-Xany.....	16
5.4.3	Exemple de configuration réelle.....	19
5.4.4	Modes de commande RC et câblage.....	21
5.4.5	Outil de diagnostic intégré.....	22
5.4.6	Commande en mode Normal et en mode imPulsionnel.....	23
5.5	Archivage et restauration des configurations.....	24
5.5.1	Archivage de la configuration courante.....	24
5.5.2	Restauration à partir de la configuration sauvegardée.....	25



# 1 CE DOCUMENT

## 1.1 Versions

Version du Manuel	Date	Raison de l'évolution
0.1	23/05/2020	Création
0.2	25/07/2021	Ajout des commandes en mode « impulsif » (Firmware Release MS16 V0.2)
0.3	29/07/2021	<ul style="list-style-type: none"><li>- Documentation de la commande C? déjà existante permettant d'afficher la configuration complète ainsi que l'état des commandes</li><li>- Correction des liens hyper-textes</li><li>- Ajout photo du câble USB micro</li><li>- Ajout de la commande A? pour Archiver la configuration courante (Firmware Release MS16 V0.3)</li></ul>
0.4	10/11/2022	Ajout paramètre de filtrage RC pour la commande Mode (Firmware Release MS16 V0.5)

## 1.2 Copyright

Ce document est Copyright © 2021-2022 **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 **MS16-Xany** 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 **MS16-Xany** ainsi que le paramétrage pour son utilisation avec l'émetteur **OpenAVRc**.

**MS16-Xany** est un module décodeur MultiSwitch disposant de 16 sorties « Tout-Ou-Rien ».

Il supporte les protocoles de commande RC suivants : PWM, CPPM, SBUS, SRXL, SUMD, IBUS, JETI.

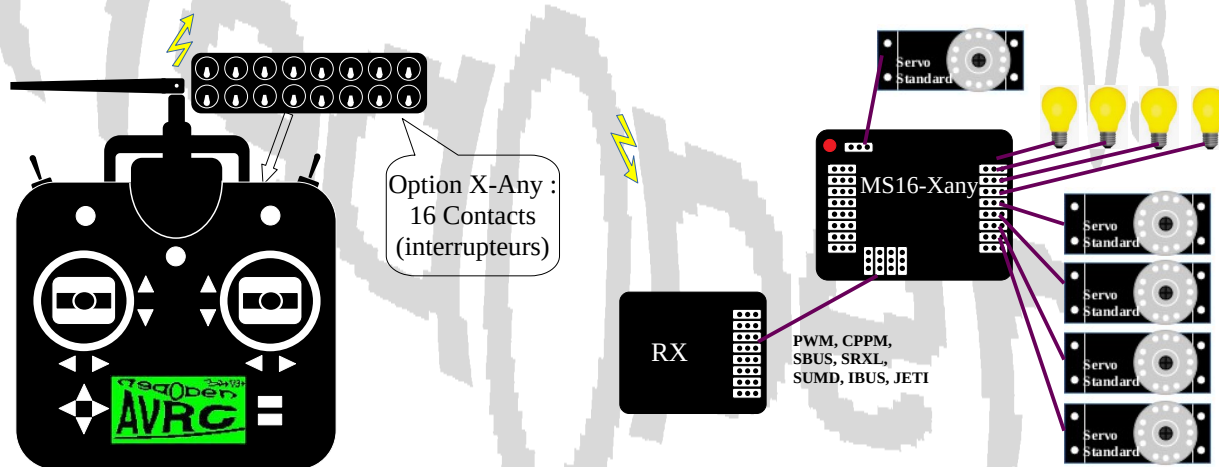
En plus de la fonction MultiSwitch classique, **MS16-Xany** peut être paramétré afin que ses sorties pilotent des servos en « Tout-Ou-Rien ».

Enfin, **MS16-Xany** peut fournir une voie proportionnelle d'appoint permettant la gestion d'un servo ou d'un variateur/contrôleur de vitesse.

Il est également possible de commander chaque sortie en mode « impulsif ».

## 2 PRESENTATION DE MS16-XANY

### 2.1 Vue d'ensemble



### 2.2 Spécifications du décodeur MS16-Xany

Spécification	Valeur / Caractéristique	Note
Alimentation CPU	+3.3V à +6.6V	Mettre le cavalier « tension récepteur » sur « = » ou « ↓ » selon la valeur de la tension fournie par le récepteur
Interface RC avec le récepteur	PWM, CPPM, SBUS, SRXL, SUMD, IBUS, JETI	
Protocole X-Any	<ul style="list-style-type: none"> <li>- Protocole numérique universel utilisé par OpenAVRc pour les accessoires distants</li> <li>- Contrôle d'intégrité par checksum 8 bits</li> <li>- Fonctionne avec tous les protocoles, y compris en 2.4GHz : <ul style="list-style-type: none"> <li>- Protocole PPM</li> <li>- Protocoles SPIRfMod</li> <li>- Protocoles MultiMod</li> </ul> </li> </ul>	Contrairement à bon nombre de modules MultiSwitches, <b>MS16-Xany</b> fonctionne également avec les modules HF en 2.4GHz
16 sorties Tout-Ou-Rien	- Commandées en Tout-Ou-Rien par sortie transistorisée	Sorties configurées en mode « numérique » (MultiSwitch)
Alimentation des sorties	<ul style="list-style-type: none"> <li>- Interne : tension du récepteur</li> <li>- Externe : tension externe 5 à 24V</li> </ul>	Sélectionnable par cavalier (commun par bloc de 8 sorties)
Diode de roue libre	Diodes de roue libre des 8 + 8 sorties au + de l'alimentation des sorties	Sélectionnable par cavalier (commun par bloc de 8 sorties)
16 sorties Servo	<ul style="list-style-type: none"> <li>- Commandées en Tout-Ou-Rien</li> <li>- Positions extrêmes paramétrables</li> <li>- Durée du mouvement entre positions extrêmes paramétrable</li> </ul>	<ul style="list-style-type: none"> <li>- Sorties configurées en mode « Servo »</li> <li>- Inversion de la course des servos possible en permutant les valeurs extrêmes</li> <li>- Possibilité d'utiliser la tension du récepteur</li> <li>- Possibilité d'utiliser un tension externe (compatible avec les servos)</li> </ul>
1 sortie Servo proportionnelle	Commande proportionnelle d'un servo de 988µs à 2008µs (0 à 255 pas de 4µs)	<b>MS16-Xany</b> doit être configuré explicitement par la commande P=SW+P pour gérer la sortie proportionnelle.
LED rouge perte de signal	Au bout de 1.5 secondes sans signal	Non gérée si servo proportionnel utilisé (broche partagée)
Failsafe	<ul style="list-style-type: none"> <li>- Toutes les sorties passent à 0 en cas de perte de signal RC</li> <li>- Le servo proportionnel conserve sa position courante (Hold)</li> </ul>	Synchrone avec LED rouge perte de signal
Port série USB	USB 2.0	Pour configuration avancée à l'aide d'une application type « Terminal série »
Dimensions	Carte N°1 (Principale) : L x l = 60 x 55 Carte N°2 (Alim VExt1) : L x l = 60 x 22 Carte N°2 (Alim VExt2) : L x l = 60 x 16	Cartes N°2 et N°3 optionnelles

OpenV3

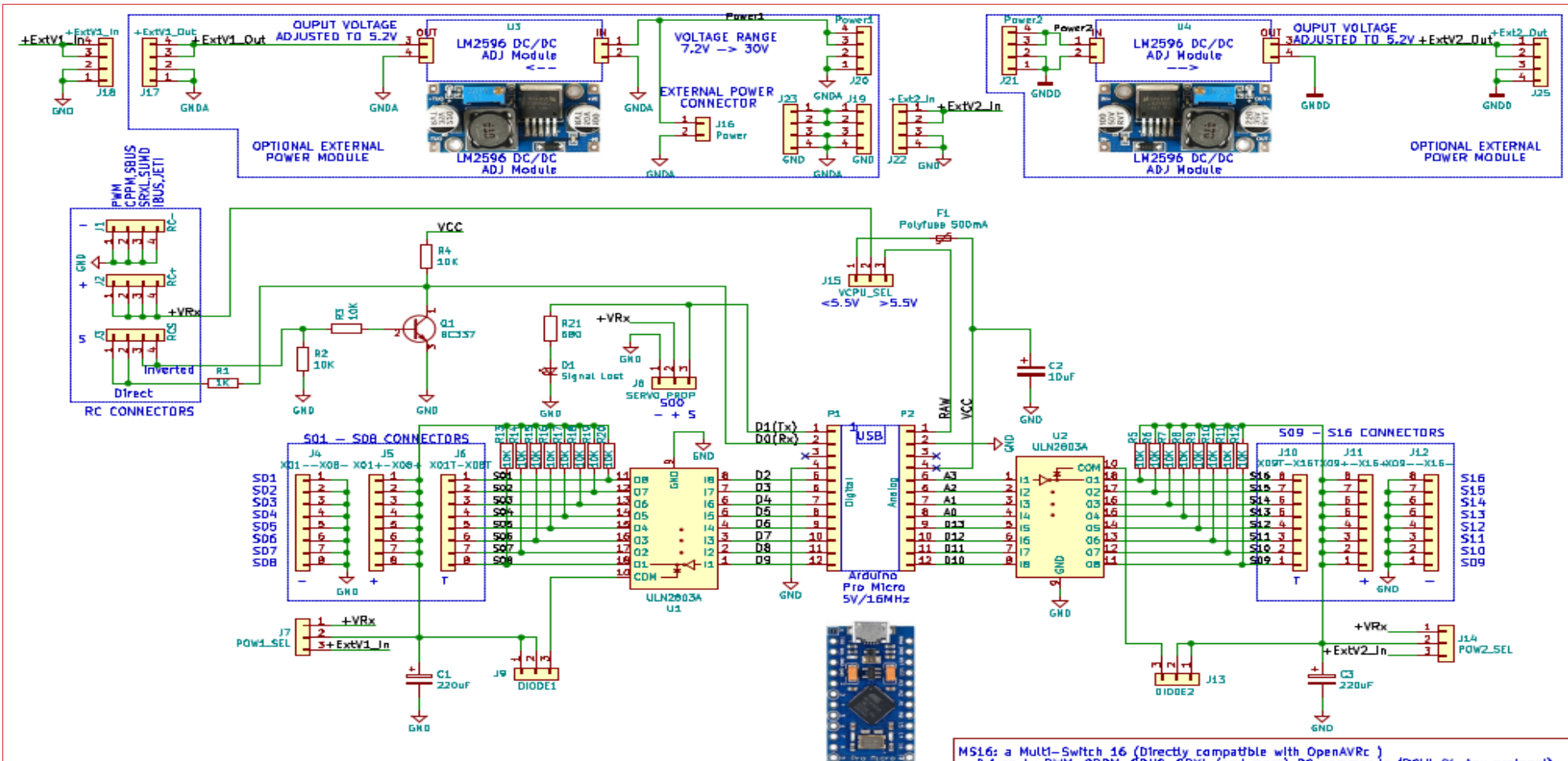
Page left intentionally blank

/

Page laissée intentionnellement blanche

AVRc

### 3 SCHEMA DU DECODEUR MS16-XANY



MS16: a Multi-Switch 16 (Directly compatible with OpenAVRc )  
 - Driven by PWM, CPM, SBUS, SRXL (and more) RC commands (RCUL/X-Any protocol)  
 - Drive digital outputs (Leds, relays)  
 - Drive regular servos (up to 2 x built-In 5V/2A power supply)

By RC NAVY and Pierrotm777 © Team OpenAVRc

Sheet: /

File: MultiSwitch\_Sw16\_ProMicro.sch

**Title: MultiSwitch\_Sw16\_ProMicro**

Size: A4

Date: 2020-05-24

Rev: V1.0

KiCad E.D.A. kicad (5.1.5)-3

Id: 1/1

## 4 REALISATION DU DECODEUR MS16-XANY

### 4.1 Circuit imprimé

La conception du circuit imprimé du décodeur **MS16-Xany** a été réalisée sous KiCad. Le [dossier de fabrication](#) est disponible sur le dépôt GitHub du projet **OpenAVRc**.

### 4.2 Arduino Pro Micro

Le décodeur **MS16-Xany** fait appel à un arduino Pro Micro.

**ATTENTION** : Il en existe plusieurs modèles, il faut prendre l'arduino **Pro Micro 5V à 16MHz**.

### 4.3 Chargement du Firmware dans l'arduino Pro Micro

Le chargement du Firmware (micro-logiciel embarqué) consiste à charger le contenu d'un fichier HEX dans la mémoire Flash du micro-contrôleur contenu dans l'arduino **Pro-Micro**.

Le fichier HEX [MS16-Xany Vx y. HEX](#) est à télécharger sur le dépôt GitHub du projet **OpenAVRc**.

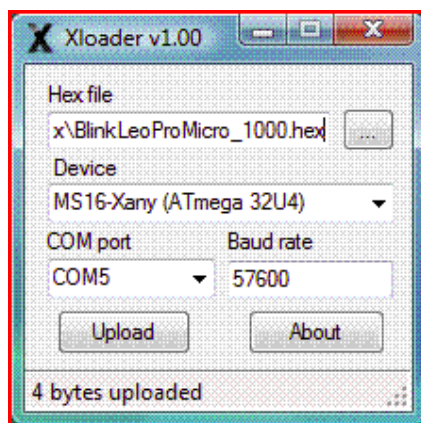
#### 4.3.1 Utilitaire XLoader modifié

Sous Windows, le chargement s'effectue à l'aide de l'utilitaire **XLoader**. Il s'agit d'une version légèrement modifiée (de la version « standard » disponible sur internet) qui permet de faire le **Reset** automatique de l'arduino **Pro Micro** nécessaire pour charger le fichier HEX.

L'archive [ArdUpload.zip](#) contenant l'utilitaire **XLoader** est à télécharger sur le même dépôt GitHub.

L'utilisation de cet utilitaire est trivial :

1. Double-cliquer sur **XLoader.exe** : l'interface graphique s'ouvre



2. Sélectionner la carte à programmer (**Device**): sélectionner **MS16-Xany (ATmega 32U4)**  
→ le débit du port série s'affiche automatiquement (**Baud rate**)
3. Sélectionner le fichier HEX à charger (**Hex file**): sélectionner **MS16-Xany\_Vx\_y.HEX**
4. Sélectionner la port série utilisé par Windows (**COM port**): **COMx**
5. Cliquer sur **Upload**
6. La fin du téléchargement est indiquée par **xx bytes uploaded** (la valeur xx est fausse : il s'agit d'un petit bug sans conséquence).



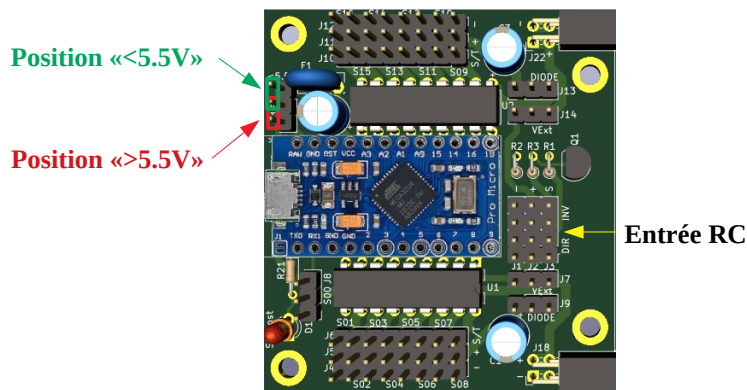
## 5 UTILISATION

### 5.1 Connexion au récepteur

Avant de connecter **MS16-Xany** 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 5.5V, placer le cavalier « tension récepteur » sur « **<5.5V** »
2. Supérieure à 5.5V, placer le cavalier « tension récepteur » sur « **>5.5V** »



### 5.2 Mode standard MultiSwitch/Commutation Tout-Ou-Rien

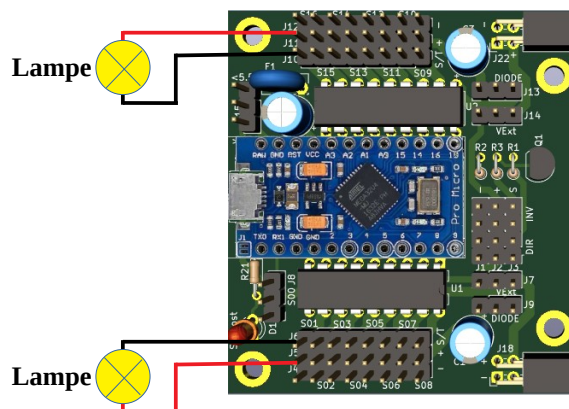
MultiSwitch, c'est le mode par défaut après chargement du Firmware : le mode commutation Tout-Ou-Rien des 16 sorties **S01** à **S16**. Il n'y a donc rien à configurer pour fonctionner en mode MultiSwitch.

Le mode RC par défaut est le PWM : il faut donc connecter **MS16-Xany** sur une voie du récepteur.

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

Le **MS16-Xany** s'utilise alors comme un module MultiSwitch standard :

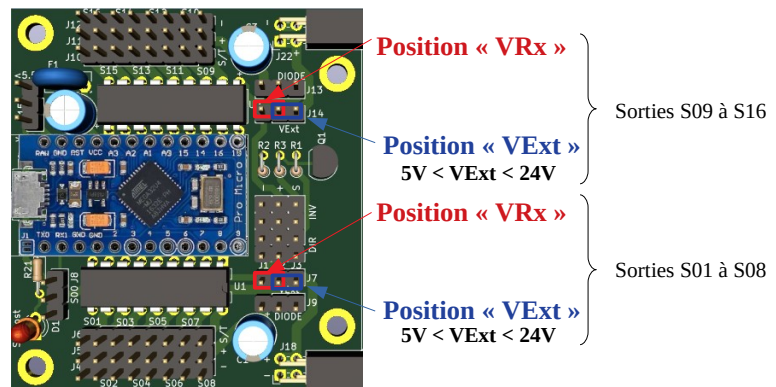
- Les « utilisations » (ex : une lampe) se branchent sur les sorties **S01** à **S16** entre les broches « **S/T** » et « **+** », la rangée 8 points de « **-** » en bord de carte n'est pas utilisée dans ce cas.



## 5.2.2 Sélection de la tension d'alimentation des sorties S01 à S08

Il est possible d'alimenter les sorties **S01 à S08** et **S09 à S16** (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**)  
→ Cavalier de sélection alimentation sur « **VExt** »

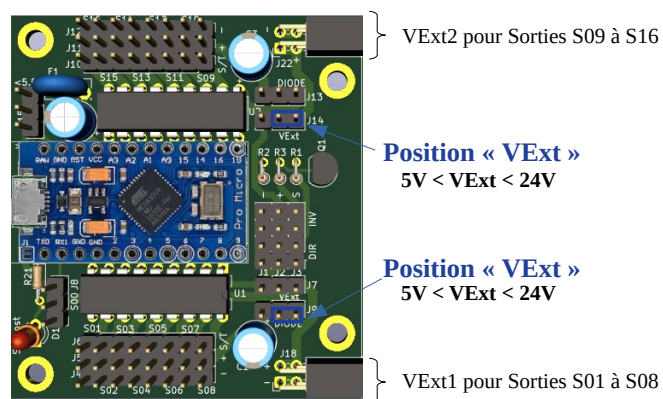


Chaque cavalier s'applique à un groupe de 8 sorties. Il est donc possible que le récepteur fournisse le **+** « **VRx** » à un des groupes de 8 sorties et que « **VExt** » fournisse le **+** à l'autre groupe de 8 sorties.

## 5.2.3 Alimentations externes VExt1 et VExt2

Lorsque les cavaliers sont sur VExt, il est nécessaire de fournir une alimentation externe.

- ◆ Pour « **VExt1** », l'accès se fait via la connecteur 2 x 2 points du coin inférieur droit de la carte
- ◆ Pour « **VExt2** », l'accès se fait via la connecteur 2 x 2 points du coin supérieur droit de la carte



### 5.2.4 Modules optionnels d'alimentation VExt1 et VExt2

Les utilisations connectées sur les sorties **S01** à **S08** et **S09** à **S16** ne s'alimentent pas forcément sous la même valeur de tension.

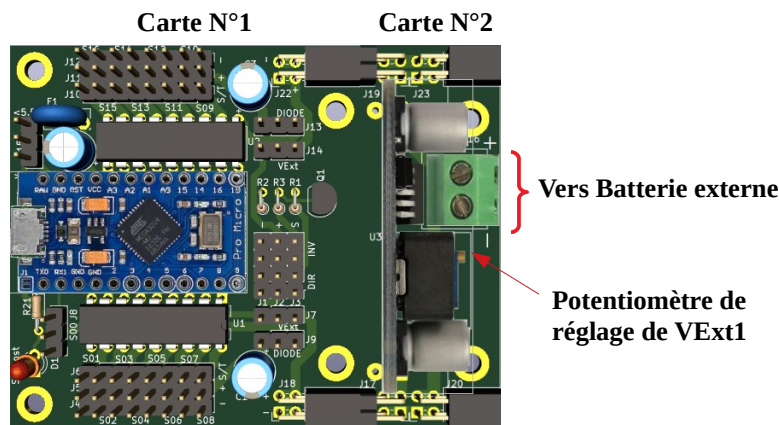
Par exemple, **S01** à **S08** pourraient être utilisées pour commander des servos en 5.5V et **S09** à **S16** pourraient être utilisées pour commander des relais en 12V.

Afin d'éviter de multiplier les sources de tension (batteries), **MS16-Xany** dispose de 2 modules optionnels hébergeant des convertisseurs de tension DC/DC qui supportent de 7.2V à 30V en entrée et qui disposent d'une sortie réglable de 5V à 24V (par potentiomètre).

Ainsi, il est possible d'alimenter **S01** à **S08** et **S09** à **S16** avec la même batterie.

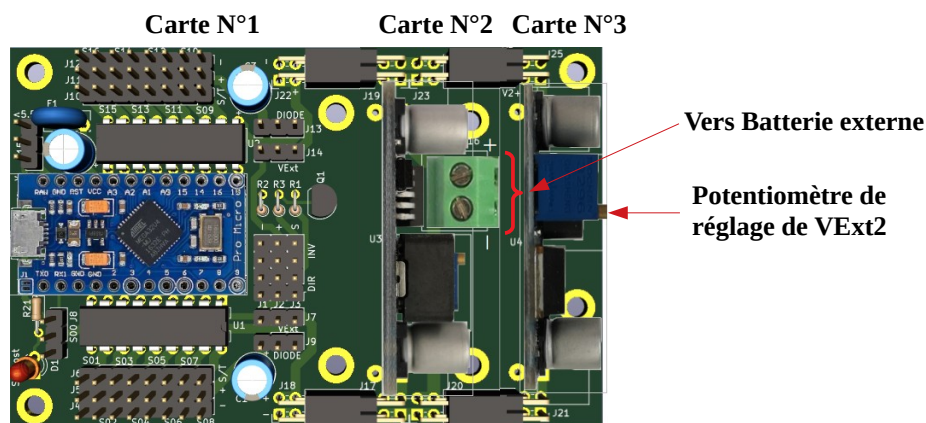
### 5.2.5 Module optionnel d'alimentation VExt1

Il s'agit d'une 2<sup>e</sup> carte (carte N°2) qui vient se connecter à la carte principale. Elle embarque un convertisseur DC/DC à bas coût (LM2596 DC to DC, adjustable, step down converter/regulator) capable de fournir 2A en continu et 3A en pointe.



### 5.2.6 Module optionnel d'alimentation VExt2

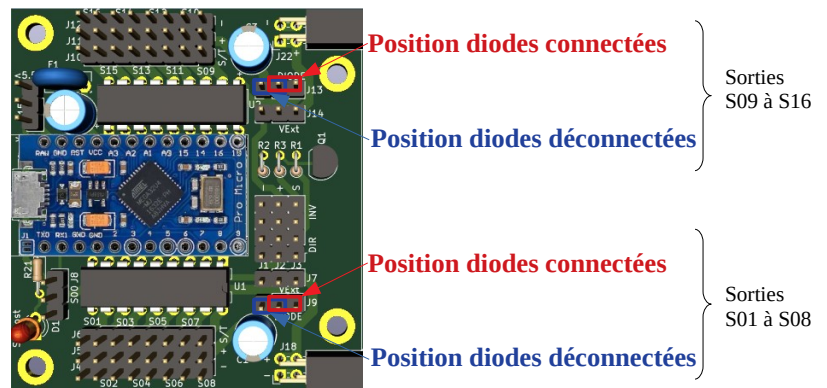
Il s'agit d'une 3<sup>e</sup> carte (carte N°3) qui vient se connecter à la 2<sup>e</sup> carte. Elle embarque également un convertisseur DC/DC à bas coût capable de fournir 2A en continu et 3A en pointe.



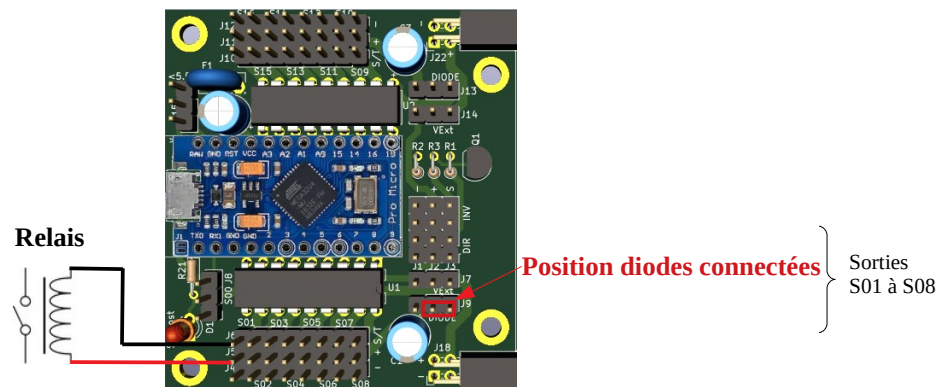
Info : Cette 3<sup>e</sup> carte peut être réglée à une tension différente de la carte N°2.

### 5.2.7 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.

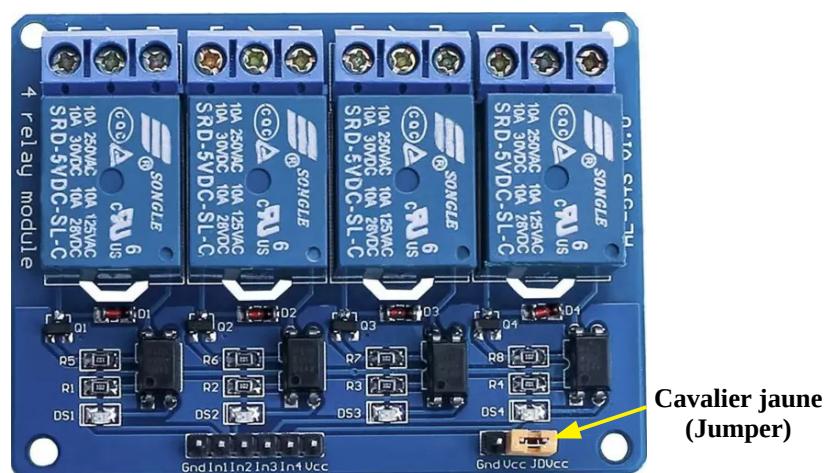


### 5.2.8 Commande directe de relais avec diode intégrée à l'ULN2803



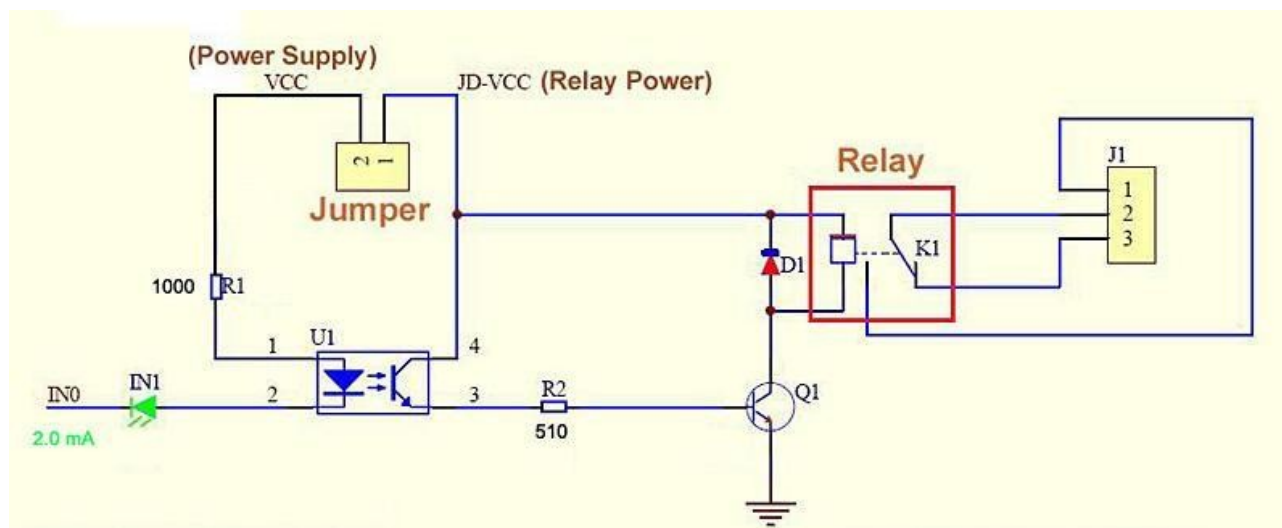
### 5.2.9 Commande de relais opto-isolés

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 :

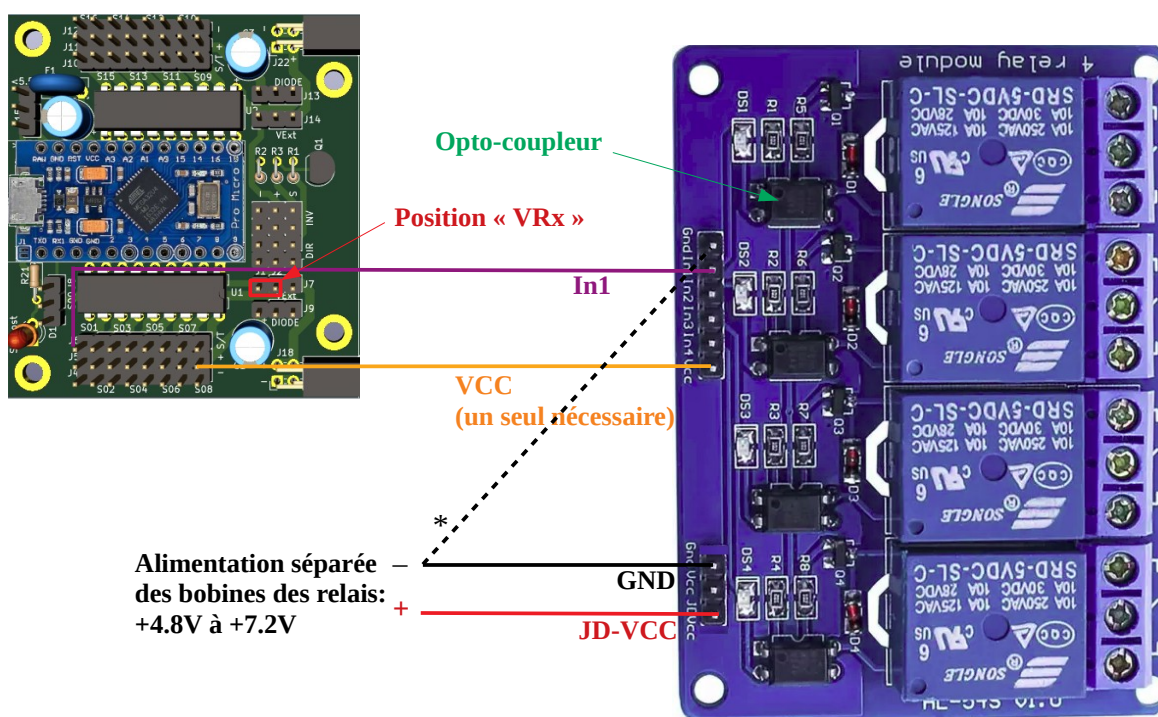


- ◆ **VCC** est la tension de commande des opto-coupleurs : accessible sur le connecteur de bord de carte
- ◆ **JD-VCC** est la tension de commande des bobines des relais : accessible sur un des points du connecteur « Jumper »

### Note importante :

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

### 5.2.10 Montage conseillé pour relais 5V opto-isolés



- ◆ Les opto-coupleurs sont alimentés par la tension du récepteur (conso : quelques x10 mA)
- ◆ VCC de la carte relais est relié à un des points de barrette de contact +
- ◆ In1 de la carte relais est reliée à la sortie S01 (comme dans la figure, par exemple)
- ◆ In2 de la carte relais est reliée à la sortie S02, etc (toujours par exemple)
- ◆ 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 X-Any côté 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 **MS16-Xany** 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.16 : cela va transmettre l'état de 16 contacts
4. Si le servo proportionnel sera utilisé sur **MS16-Xany**, sélectionner un des choix proposé par « Prop. », cela va ajouter la transmission de la valeur du proportionnelle.



Exemple de configuration X-Any sur l'émetteur OpenAVRc

Note :

Avec X-Any, côté émetteur, chaque contact peut être :

- un interrupteur OFF/ON
- un bouton-poussoir
- un contact de rotacteur à n positions

## 5.4 Mode avancé/Commande de servos

Le décodeur **MS16-Xany** dispose d'un accès pour les configurations avancées : le port série USB de l'arduino **Pro Micro**.

C'est cet accès série qui va permettre :

- ◆ l'utilisation de servos connectés aux sorties **S00** et **S01** à **S16**
- ◆ le changement du mode de commande RC (choix entre PWM, CPM, SBUS, SRXL, SUMD, IBUS et JETI) et le mode de filtrage RC (Non filtré, Filtré : voir la table des commandes)
- ◆ la définition du contenu du message X-Any envoyée par l'émetteur (P=SW ou P=SW+P)
- ◆ Commander les sorties **S00** et **S01** à **S16** même si l'émetteur RC est coupé ou absent
- ◆ Faire un diagnostic en cas de non fonctionnement
- ◆ Faire des sauvegardes de configuration et des restaurations de configuration à partir des sauvegardes

### 5.4.1 Utilisation du port série USB de MS16-Xany

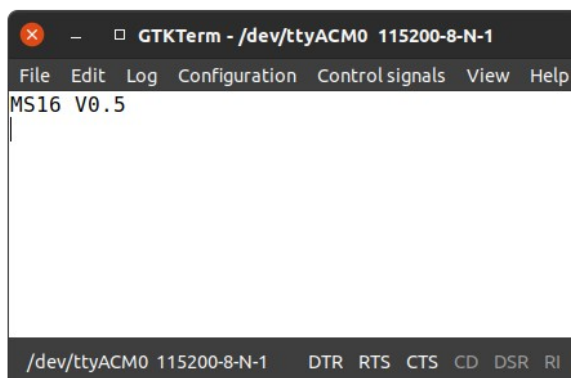
Pour accéder au port série de **MS16-Xany**, il faut un câble USB type micro afin de se connecter à un PC.



1. Connecter le câble USB entre le connecteur micro USB du Pro Micro et le port USB d'un PC
2. Sur le PC, ouvrir un Terminal série, par exemple, PuTTY, TeraTerm, HyperTerminal, GtTerm, ou encore CoolTerm avec les paramètres suivants : 115200 bauds, 8 bits de données, 1 bit de stop, pas de parité.

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

3. L'USB du PC va alimenter **MS16-Xany** pendant cette phase de configuration.
4. Appuyer sur la touche « Entrée » de votre clavier, le message « MS16 VX.Y » doit apparaître sur le Terminal série comme illustré ci-dessous. Si ce n'est pas le cas, débrancher/rebrancher le câble USB et recommencer à l'étape 4 ci-dessus.



Exemple de connexion avec le Terminal GtTerm sous Linux

## 5.4.2 Les messages de commande de MS16-Xany

La liste des messages supportés par **MS16-Xany** est donnée dans la table suivante :

← Commande/ → Réponse	Action	Remarque
← Enter → MS16 Vx.y	Vérification de la connectivité entre <b>MS16-Xany</b> et le PC	Bien vérifier que le Terminal série envoie un caractère «retour chariot».
← P? → P=SW ou → P=SW+P	Retourne le contenu attendu dans le message X-Any envoyé par l'émetteur OpenAVRc. Ce contenu peut être : - SW : uniquement la position des 16 Switches - SW+P : position des 16 Switches + valeur proportionnelle	Doit être aligné avec le paramétrage défini dans le menu de l'instance X-Any de l'émetteur OpenAVRc.
← P=ContenuMessage → P	Définit le contenu attendu dans le message X-Any envoyé par l'émetteur OpenAVRc ContenuMessage peut être : - SW : uniquement la position des 16 Switches - SW+P : position des 16 Switches + valeur proportionnelle	Doit être aligné avec le paramétrage défini dans le menu de l'instance X-Any de l'émetteur OpenAVRc.
← M? → M=Proto[v];Filtrage	Retourne le Mode RC (Protocole) utilisé par <b>MS16-Xany</b> . Protocoles possibles : - P;N: PWM non filtré. <b>MS16-Xany</b> est connecté sur la voie PWM du récepteur définie dans le menu X-Any d'OpenAVRc. - Cv;N: CPPM voie v. <b>MS16-Xany</b> est connecté sur la sortie CPPM du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Sv;N: SBUS voie v. <b>MS16-Xany</b> est connecté sur la sortie SBUS du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Xv;N: SRXL voie v. <b>MS16-Xany</b> est connecté sur la sortie SRXL du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Dv;N: SUMD voie v. <b>MS16-Xany</b> est connecté sur la sortie SUMD du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Iv;N: IBUS voie v. <b>MS16-Xany</b> est connecté sur la sortie IBUS du récepteur et utilise la voie v définie dans le menu X-Any d'OpenAVRc. - Jv;N: JETI voie v. <b>MS16-Xany</b> est connecté sur la sortie JETI du récepteur et utilise la voie v définie	D'autres protocoles séries pourront être ajoutés dans le futur. La voie v doit être dans la plage 1 à 16. Le paramètre de filtrage vaut : - N : Non filtré - F : Filtré 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. C'est le comportement des récepteurs dits « DSP » ou encore des récepteurs PCM Futaba.



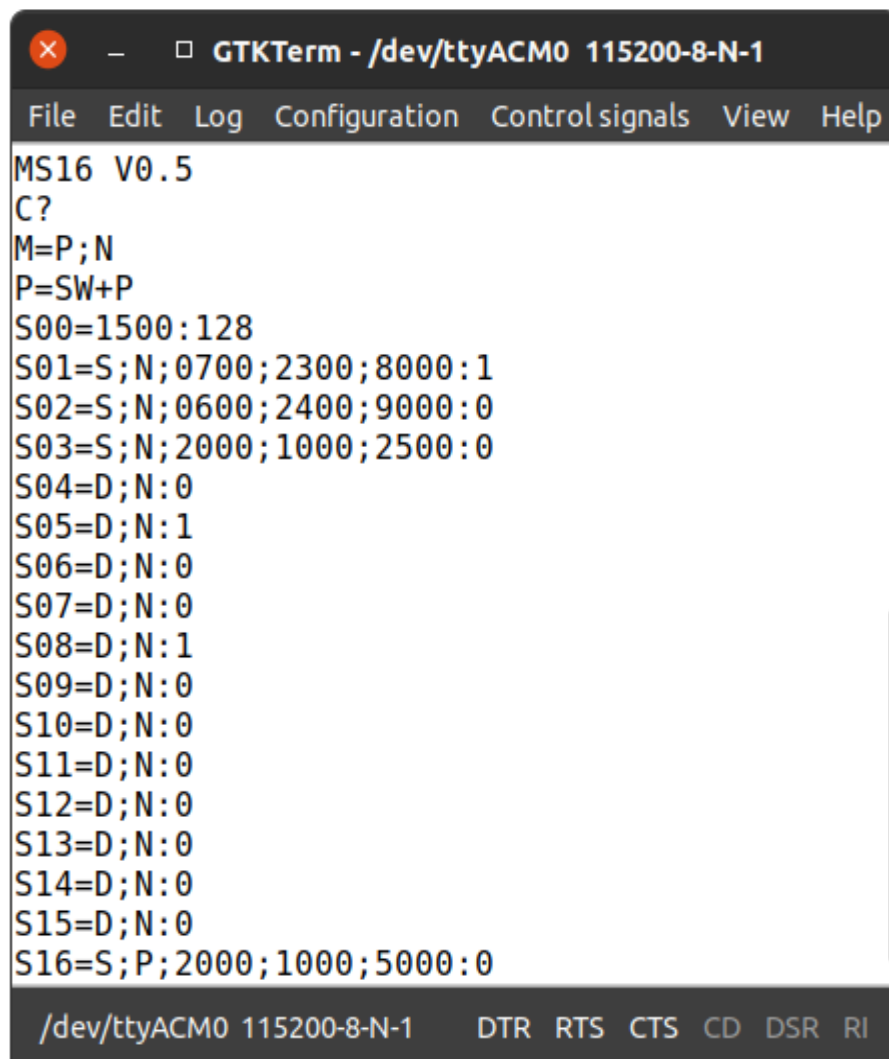
	dans le menu X-Any d'OpenAVRc.	
M=Proto[v];Filtrage	Définit le Mode de fonctionnement RC (Protocole) utilisé par <b>MS16-Xany</b> . Idem, ci-dessus.	Filtrage : - N : Non filtré - F : Filtré
← S00? → S00=Pos:OffsetPas4us	Retourne la position courante en µs et la commande OffsetPas4us qui est le nombre de pas de 4µs (valeur entre 0 et 255) à ajouter à 988 pour avoir la largeur d'impulsion en µs pour le servo proportionnel	Largeur d'impulsion(us) = 988 + (OffsetPas4us x 4)  Note : uniquement accessible si P=SW+P. Répond ERR si P=SW.
← S00=Pos → S00	Définit la position en µs pour le servo proportionnel	Renvoie ERR, si valeur non comprise entre 988 et 2008 Note : uniquement accessible si P=SW+P. Répond ERR si P=SW.
← Sxx? → Sxx=D;M:C ou → Sxx=S;M;Pos0;Pos1;Dur:C	Si xx est compris entre 01 et 16, retourne la configuration de la sortie N°xx ainsi que l'état « C » de la Commande courante associée (0 ou 1) - Si la sortie est configurée en sortie numérique MultiSwitch, la réponse est : Sxx=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 : Sxx=S;M;Pos0;Pos1;Dur:C avec 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	Renvoie ERR, si - Valeur xx non comprise entre 01 et 16 - Pos0 ou Pos1 < 600 - Pos0 ou Pos1 > 2400  Exemple de réponses :  S01=D;N:0 S02=D;P:1 S03=S;N;1000;2000;5000:0 S04=S;P;2300;600;8500:1
← Sxx=D;M → Sxx	Définit la sortie xx comme étant de type numérique (Digitale) et commandée en mode M (M=N : Mode Normal, M=P : Mode imPulsionnel)	Exemple de commandes :  S01=D;N S02=D;P
← Sxx=S;M;Pos0;Pos1;Dur → Sxx	Définit la sortie xx 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 <b>MS16-Xany</b> en tenant compte des différentes résolutions/limitations et peut être différente à la relecture.
← Sxx=C → Sxx	Si xx est compris entre 01 et 16, « C » définit la Commande (0 ou 1) pour la sortie x,x que celle-ci soit de type Digitale ou Servo	Très pratique pour les tests par l'accès série sans RC.

← Q	Quitte le mode Terminal : <b>MS16-Xany</b> peut être commandé par radio	Détecte à nouveau les pertes « radio ».
← D? → D=NiveauDiag	Retourne le niveau d'affichage de Diagnostic. D=0 : pas d'affichage D=1 : affichage valeurs SW et Prop D=2 : affichage période message	Très utile pour faire un diagnostic et pour vérifier la bonne réception des commandes de l'émetteur.
← D=NiveauDiag → D	Définit niveau d'affichage de Diagnostic. Voir ci-dessus.	Toujours à 0 après un redémarrage de <b>MS16-Xany</b> .
← C?	Retourne la <b>C</b> onfiguration complète	Pratique pour avoir une vue de la configuration globale et l'état des commandes pour chaque sortie.
← A?	Retourne la configuration complète pour <b>A</b> rchivage	Pratique pour faire une sauvegarde de la configuration par un simple copier/coller depuis la console série.

### 5.4.3 Exemple de configuration réelle

Dans l'exemple ci-dessous :

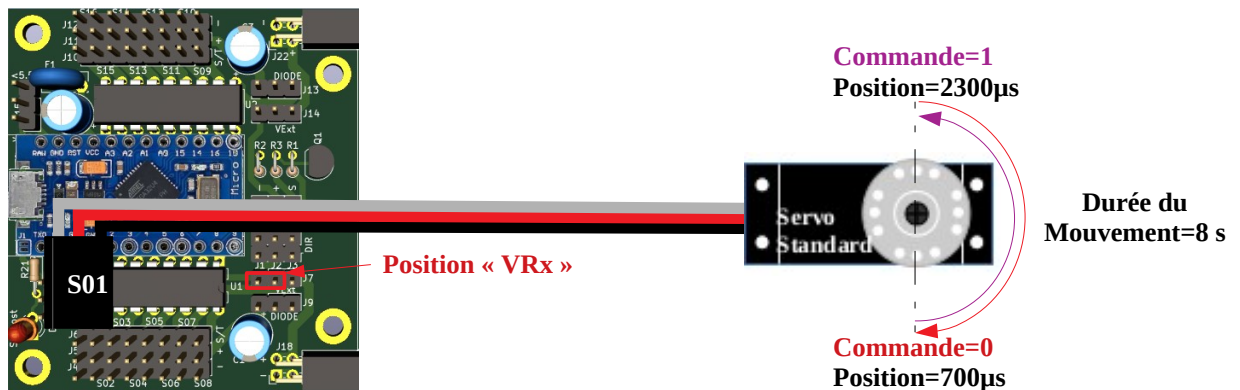
- Le mode de commande RC est PWM sans filtrage des impulsions RC : (M=P;N)
- Le message envoyé par l'émetteur contient la position des 16 contacts plus la valeur Proportionnelle : (P=SW+P). La sortie S00 est donc commandée.
- La largeur d'impulsion pour le servo connecté sur la sortie proportionnelle S00 vaut :  
 $[988 + (128 \times 4)] = 1500 \mu s$   
→ la commande (= le nombre de pas de 4  $\mu s$  à ajouter à 988  $\mu s$ ) serait de 128 pour la largeur d'impulsion de 1500  $\mu s$
- Les sorties S01, S02, S03 et S16 sont de type Servo, leurs commandes valent respectivement 1, 0, 0 et 1
- Les sorties S04, S05, S06, S07, S08, S09, S10, S11, S12, S13, S14 et S15 sont de type Digital (MultiSwitch), leurs états valent respectivement 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0 et 1



```
GTKTerm - /dev/ttyACM0 115200-8-N-1
File Edit Log Configuration Control signals View Help
MS16 V0.5
C?
M=P;N
P=SW+P
S00=1500:128
S01=S;N;0700;2300;8000:1
S02=S;N;0600;2400;9000:0
S03=S;N;2000;1000;2500:0
S04=D;N:0
S05=D;N:1
S06=D;N:0
S07=D;N:0
S08=D;N:1
S09=D;N:0
S10=D;N:0
S11=D;N:0
S12=D;N:0
S13=D;N:0
S14=D;N:0
S15=D;N:0
S16=S;P;2000;1000;5000:0
/dev/ttyACM0 115200-8-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 700  $\mu$ s à 2300  $\mu$ s (soit un mouvement d'environ 180°) en 8 secondes, et va de 2300  $\mu$ s à 600  $\mu$ s (soit un mouvement d'environ 180°) en 8 secondes si le contact N°1 côté émetteur est ouvert (**Commande=0**). Ex : **S01=S;N;700;2300;8000**

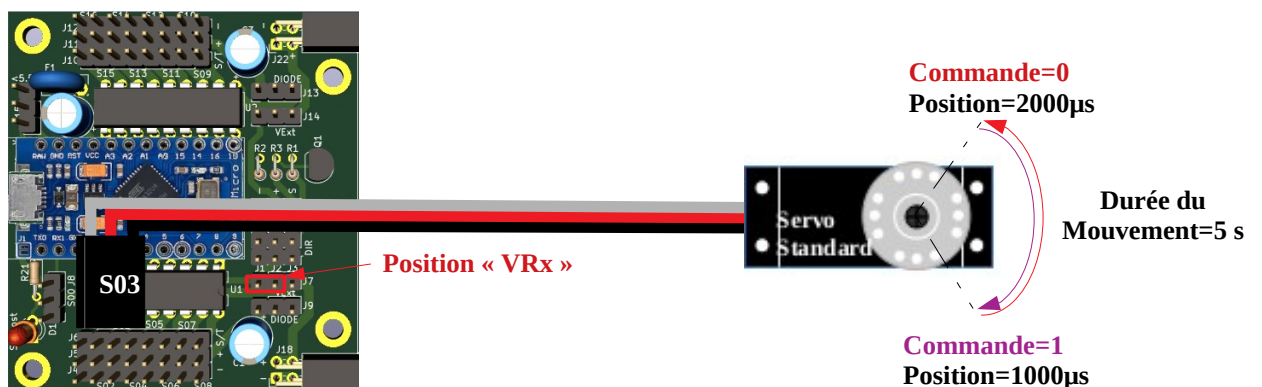


## 2. Mouvement de servo dans le sens des aiguilles d'une montre

Avec **MS16-Xany**, 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°3 côté émetteur est fermé (**Commande=1**), l'impulsion du servo connecté à la sortie N°3 va de 2000  $\mu$ s à 1000  $\mu$ s (soit un mouvement d'environ 100°) en 5 secondes si la commande côté émetteur vaut 1, et de 1000  $\mu$ s à 2000  $\mu$ s (soit un mouvement d'environ 100°) en 5 secondes si le contact N°2 côté émetteur est ouvert (**Commande=0**).

Ex : **S03=S;N;2000;1000;5000**



#### 5.4.4 Modes de commande RC et câblage

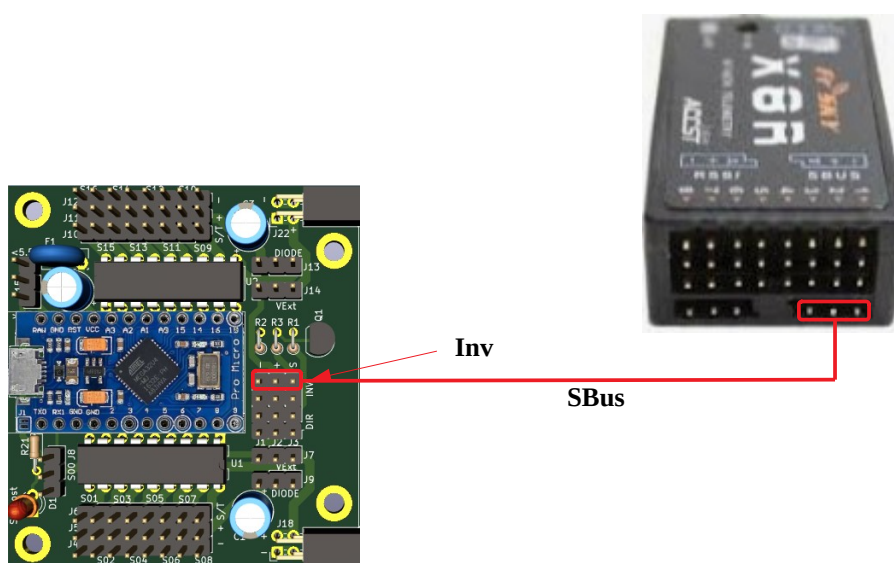
La commande en mode Terminal pour définir le mode de commande RC est M : M=Proto[v];Filtrage.

Voir le détail de la commande au chapitre « Les messages de commande de MS16-Xany ».

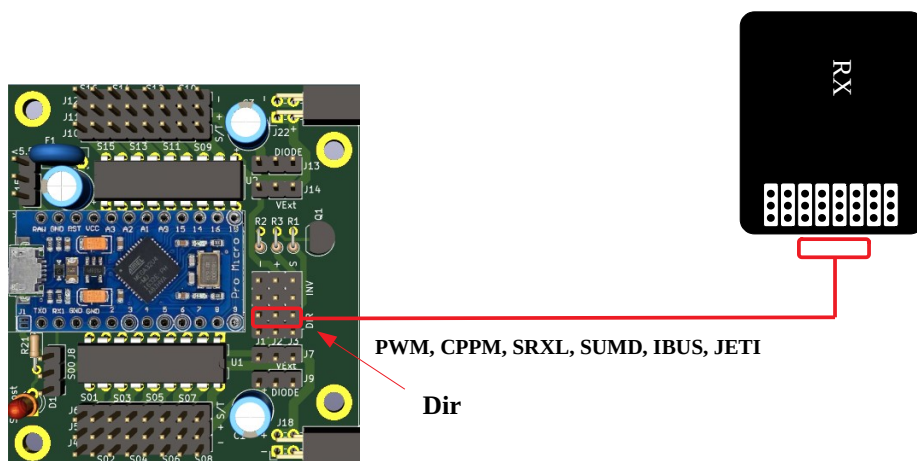
Quelque soit le mode de commande RC utilisé, il faut utiliser un câble « type servo » femelle/femelle.

Contrairement à tous les modes de commande RC actuellement supportés par **MS16-Xany** (PWM, CPPM, SBUS, SRXL, SUMD, IBUS, JETI), seul le SBus requiert un inverseur de signal. Celui-ci est déjà intégré sur la carte **MS16-Xany**. Il suffit juste de se connecter sur l'une des 2 rangées de 3 points en face de « Inv » (Inversé).

Ci-dessous, le câblage en mode SBus avec un récepteur FrSky X8R.



Pour tous les autres modes de commande RC (PWM, CPPM, SRXL, SUMD, IBUS, JETI), il faut se connecter sur l'une des 2 rangées de 3 points en face de « Dir » (Direct = Non inversé).



### 5.4.5 Outil de diagnostic intégré

Le décodeur **MS16-Xany** intègre plusieurs outils de diagnostic.

1. Quand il ne reçoit pas le message attendu par l'émetteur, toutes les 1.5 seconde :
  - ◆ il passe en mode Failsafe
  - ◆ maintient la dernière valeur proportionnelle de S00
  - ◆ met les sorties numériques de S01 à S16 à 0
  - ◆ affiche « Failsafe ! » sur le Terminal série
2. Si la commande D=1 est envoyée par le Terminal série, les valeurs de commande des 16 Switches ainsi que la valeur proportionnelle (si  $P=SW+P$ ) sont affichées toutes les 500 ms :

```
GtkTerm - /dev/ttyACM0 115200-8-N-1
File Edit Log Configuration Controlsignals View Help
MS16 V0.1
D=1
11111111
6543210987654321
SW:0100000010000001
P: 59
11111111
6543210987654321
SW:0100000010000001
P: 59
11111111
6543210987654321
SW:0100000010000001
P: 59
/dev/ttyACM0 115200-8-... DTR RTS CTS CD DSR RI
```

Sortie S16

Valeur de la sortie S16

Valeur proportionnelle S00 (de 0 à 255)

3. Si la commande D=2 est envoyée par le Terminal série, la période de réception du message **X-Any** envoyé par l'émetteur est affichée : il s'agit du temps de réaction du système.

```
GtkTerm - /dev/ttyACM0 115200-8-N-1
File Edit Log Configuration Controlsignals View Help
MS16 V0.1
D=2
T=159ms
T=164ms
T=160ms
T=162ms
T=165ms
T=162ms
T=161ms
T=160ms
T=165ms
T=159ms
T=163ms
T=164ms
T=159ms
T=162ms
T=165ms
/dev/ttyACM0 115200-8-... DTR RTS CTS CD DSR RI
```

### 5.4.6 *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 :

- **Sxx=D;M** (avec M valant **N** pour **Normal** ou **P** pour **imPulsionnel**)
- **Sxx=S;M;1000;2000;5000** (avec M valant **N** pour **Normal** ou **P** pour **imPulsionnel**)

Supposons que l'on ait la configuration suivante et que, côté émetteur, S01 et S02 soient commandés par des contacts de boutons-poussoirs :

**S01=D;N** pour piloter une corne de brume

**S02=D;P** pour piloter une lumière

Dans les 2 cas, les sorties sont de type **Digital** (Multiswitch), mais **S01** est pilotée en mode **Normal**, tandis que **S02** est pilotée en mode **imPulsionnel**.

#### 1. Cas du mode Normal :

Tant que l'on appuie sur le contact C1, la sortie S01 est à 1 et la corne de brume fonctionne.

Dès que l'on relâche le contact C1, la sortie S01 passe à 0 et la corne de brume s'arrête.

En résumé, la sortie S01 suit l'état du contact C1.

#### 2. Cas du mode imPulsionnel :

Tant que l'on appuie sur le contact C2, la sortie S02 est à 1 et la lumière fonctionne.

Dès que l'on relâche le contact C2, la sortie S02 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 S02, une nouvelle impulsion sur le contact C2 désactive la sortie S02 et ainsi de suite.

#### Note :

Si la sortie est de type **Servo** en mode **imPulsionnel**, 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 : **S03=S;P;1000;2000;5000**



## 5.5 Archivage et restauration des configurations

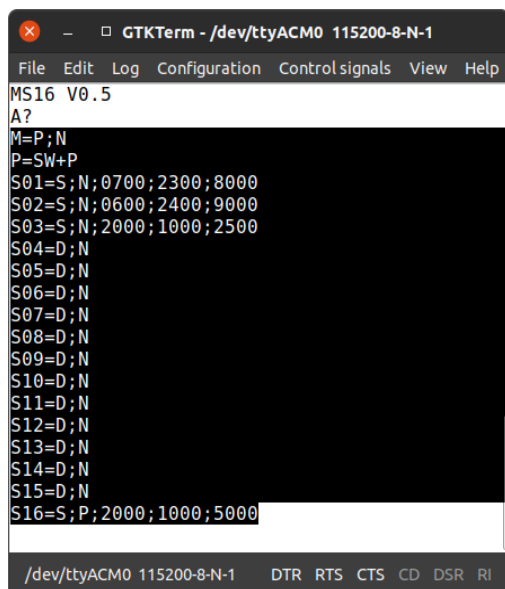
Le décodeur **MS16-Xany** ayant beaucoup de paramètres, il peut être intéressant de sauvegarder la configuration pour un modèle particulier, afin de pouvoir la restaurer plus tard. Ceci est intéressant dans le cas où le décodeur **MS16-Xany** est utilisé dans plusieurs modèles avec des configurations différentes.

Une commande d'Archivage (la commande « **A?** ») aide à réaliser cette opération.

### 5.5.1 Archivage de la configuration courante

Pour archiver la configuration courante : (exemple avec le Terminal série GTKTerm)

1. Taper la commande « **A?** », la configuration courante s'affiche sur plusieurs lignes



```
GTKTerm - /dev/ttyACM0 115200-8-N-1
File Edit Log Configuration Control signals View Help
MS16 V0.5
A?
M=P;N
P=SW+P
S01=S;N;0700;2300;8000
S02=S;N;0600;2400;9000
S03=S;N;2000;1000;2500
S04=D;N
S05=D;N
S06=D;N
S07=D;N
S08=D;N
S09=D;N
S10=D;N
S11=D;N
S12=D;N
S13=D;N
S14=D;N
S15=D;N
S16=S;P;2000;1000;5000
/dev/ttyACM0 115200-8-N-1 DTR RTS CTS CD DSR RI
```

2. Sélectionner les lignes et faire **Edit** → **Copier**
3. Ouvrir n'importe quel éditeur de texte (Notepad++, VSCode, Gedit, Kate, etc)
4. Dans cet éditeur de texte, coller en faisant **Edit** → **Coller**
5. Puis, faire **Fichier** → **Enregistrer sous** et donner le nom que vous voulez (ex : MaConf.txt)

#### **Important :**

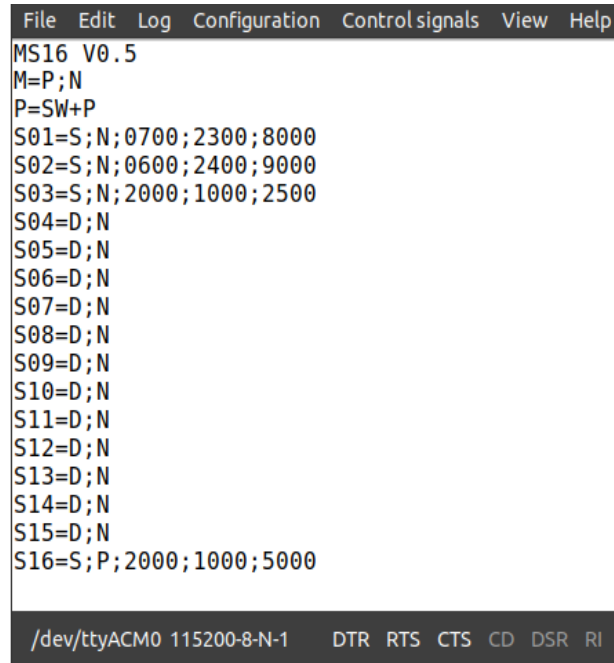
S'assurer que l'éditeur de texte a bien sauvegardé le fichier avec les terminateurs de ligne « CR » (Carriage Return), parfois appelés terminateurs de ligne « Macintosh ».



### 5.5.2 Restauration à partir de la configuration sauvegardée

Pour restaurer à partir d'une configuration sauvegardée: (exemple avec le Terminal série GTKTerm)

1. Faire **File** → **Send RAW file**, sélectionner le fichier sauvegardé (ex : MaConf.txt), puis OK. Les commandes sauvegardées sont envoyées automatiquement dans le terminal série.



```
File Edit Log Configuration Controls signals View Help
MS16 V0.5
M=P;N
P=SW+P
S01=S;N;0700;2300;8000
S02=S;N;0600;2400;9000
S03=S;N;2000;1000;2500
S04=D;N
S05=D;N
S06=D;N
S07=D;N
S08=D;N
S09=D;N
S10=D;N
S11=D;N
S12=D;N
S13=D;N
S14=D;N
S15=D;N
S16=S;P;2000;1000;5000
/dev/ttyACM0 115200-8-N-1 DTR RTS CTS CD DSR RI
```

2. Ça y est, c'est terminé : le décodeur **MS16-Xany** est reconfiguré à partir de la sauvegarde !