

Document créé le: 04/02/2012
révisé le : 21/04/2012

MANUEL D'UTILISATION DE TRISERCOM

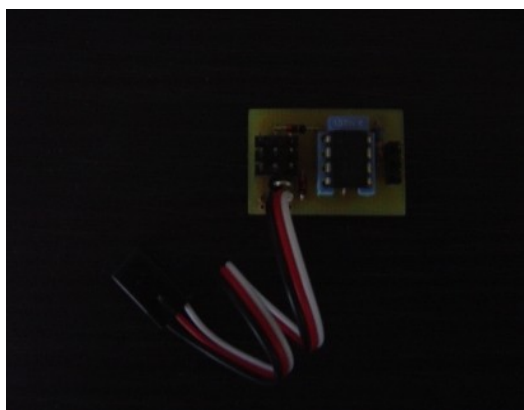


Table des matières

<u>I. PRESENTATION</u>	<u>2</u>
<u>II. CARACTERISTIQUES</u>	<u>2</u>
<u>III. SCHEMA</u>	<u>3</u>
<u>IV. NOMENCLATURE</u>	<u>3</u>
<u>V. CIRCUIT IMPRIME</u>	<u>4</u>
<u>V.1. CONNECTIQUE</u>	<u>4</u>
<u>VI. LES 3 TYPES DE COMMANDE POSSIBLES</u>	<u>4</u>
<u>VI.1. COMMANDE PAR UN SIMPLE INTERRUPTEUR</u>	<u>5</u>
<u>VI.2. COMMANDE PAR SORTIE TRANSISTORISEE</u>	<u>5</u>
<u>VI.3. COMMANDE PAR UNE VOIE D'UN RECEPTEUR RC</u>	<u>6</u>
<u>VII. CORDON DE PARAMETRAGE</u>	<u>6</u>
<u>VIII. COMMANDES ACCEPTEES EN MODE TERMINAL</u>	<u>7</u>
<u>IX. REALISATION</u>	<u>9</u>
<u>X. PARAMETRAGE A L'AIDE D'UN TERMINAL</u>	<u>10</u>
<u>X.1. EXEMPLE DE COMMANDE DE TRAIN D'UN HELICOPTERE AIRWOLF</u>	<u>10</u>
<u>X.1.1. Etapes de programmation pour le mouvement A→B (sortie train)</u>	<u>10</u>
<u>X.1.2. Etapes de programmation pour le mouvement B→A (rentrée train)</u>	<u>11</u>
<u>XI. PARAMETRAGE A L'AIDE DE TRISERCOM.EXE</u>	<u>12</u>
<u>XII. MENTIONS LEGALES</u>	<u>12</u>
<u>XIII. REVISIONS DU DOCUMENT</u>	<u>12</u>

I. PRESENTATION

Le montage [TRISERCOM](#) est un [SIMSERCOM](#) amélioré permettant de séquencer 3 servos, le tout d'une taille de 20 x 30 mm et constitué de seulement 7 composants !

Le montage [TRISERCOM](#) basé sur un micro-contrôleur PIC12F683 de chez [Microchip](#) permet de fixer les limites des positions extrêmes de 3 servos standards ainsi que la vitesse de déplacement du palonnier entre ces 2 positions et permet surtout de séquencer les 3 servos entre eux.

La commande est : soit une sortie Tout-Ou-Rien (simple interrupteur, contact de relais ou sortie Multi-Switch), soit une sortie voie d'un récepteur RC.

[TRISERCOM](#) utilise une interface série RS232 ultra-simplifiée pour le paramétrage en mode terminal à l'aide d'un PC.

Cela signifie qu'il n'y a pas obligatoire de disposer d'un programme spécifique à [TRISERCOM](#) pour être paramétré.

Cependant, afin de faciliter le paramétrage de [TRISERCOM](#), l'auteur a développé l'application Window\$ "trisercom.exe", disponible sur le site de l'auteur : [RC NAVY](#).

Il s'agit d'un programme graphique convivial qui permet de paramétrer très facilement [TRISERCOM](#) sans avoir besoin de connaître la messagerie de commande.

Si l'utilisateur ne veut pas installer "trisercom.exe" sur son PC, il suffit d'utiliser le terminal "HyperTerminal" déjà inclus à Window\$ XP. Malheureusement, pour les utilisateurs de Window\$ VISTA ou SEVEN, le Terminal n'est plus fourni par Micro\$oft. Il est donc nécessaire d'installer un Terminal de type "TeraTerm" par exemple.

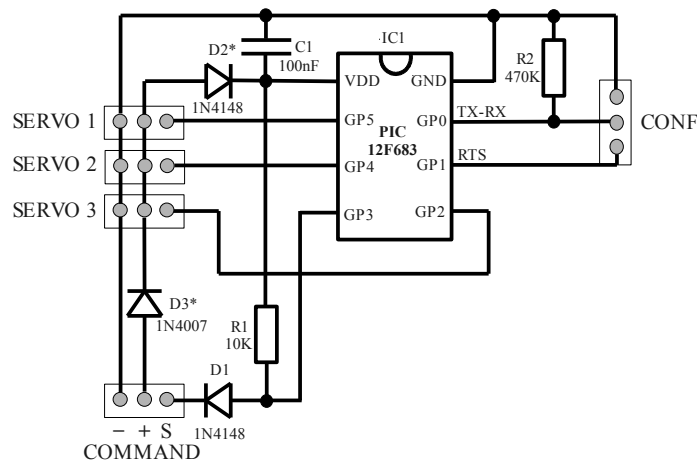
Sous Linux, le terminal "Minicom" ou "GtkTerm" fera parfaitement l'affaire.

Aujourd'hui, presque tout le monde dispose d'un PC, c'est pourquoi, cette solution est d'actualité.

II. CARACTERISTIQUES

- Tout comme [SIMSERCOM](#), la commande de [TRISERCOM](#) peut être, soit un simple interrupteur, soit une sortie TOR, ou encore une sortie "voie" d'un récepteur RC,
- Les paramètres de réglage des 3 servos sont complètement indépendants et sont :
 - ◆ Position extrême A du palonnier,
 - ◆ Position extrême B du palonnier,
 - ◆ Vitesse pour aller de la position A à B,
 - ◆ Vitesse pour aller de la position B à A indépendante de la vitesse pour aller de A à B,
 - ◆ Critère de démarrage pour le sens A→B:
 - Démarrage immédiat dès la commande reçue,
 - Démarrage conditionnel sur position d'un des autres servos (seuil réglable),
 - ◆ Critère de démarrage pour le sens B→A:
 - Démarrage immédiat dès la commande reçue,
 - Démarrage conditionnel sur position d'un des autres servos (seuil réglable),
- En résumé, chaque servo peut démarrer immédiatement ou dès qu'un des autres servos a atteint une certaine course, et cela, dans les 2 directions, et à des vitesses indépendantes,
- Support natif de l'inversion de sens des servos (tout simplement en programmant une position extrême B inférieure à la position extrême A),
- Interface de paramétrage série de type pseudo RS232,
- Pas d'obligation d'y connecter les 3 servos : fonctionne avec 1, 2 ou 3 servos,
- Alimentation de 4,8V à 6V.

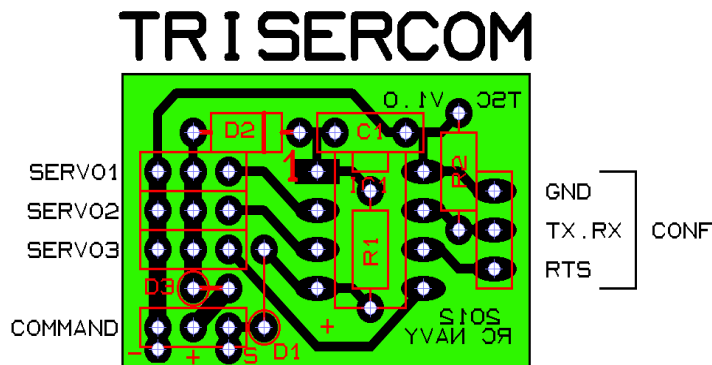
III. SCHEMA



IV. NOMENCLATURE

Repère	Valeur	Référence	Désignation	Remarque
Micro-contrôleur				
IC1		PIC 12F683	Micro-contrôleur Microchip	
Condensateur				
C1	100nF		Condensateur milfeuïl	
Résistance				
R1	10K		Résistance ¼ W	
R2	470K		Résistance ¼ W	
Diode				
D1		1N4148	Diode faible signal	
D2		1N4148	Diode faible signal	*: D2 montée pour alimentation en +6V, D2 remplacée par un strap pour alimentation en 4,8V
D3		1N4007	Diode de redressement	*: D3 montée pour alimentation en +6V, D3 remplacée par un strap pour alimentation en 4,8V
Connectique				
COMMAND			Cordon de servo	Pour signal d'entrée du récepteur RC
SERVO 1			Picots 3 points	Pour connecter le servo N°1
SERVO 2			Picots 3 points	Pour connecter le servo N°2
SERVO 3			Picots 3 points	Pour connecter le servo N°3
CONF			Picots 3 points	Pour connecter le cordon de paramétrage

V. CIRCUIT IMPRIME



Le circuit imprimé est très compact, puisqu'il ne mesure que 20 x 30 mm.

Le circuit imprimé est disponible sur le site de l'auteur au format TCI : [RC NAVY](#)

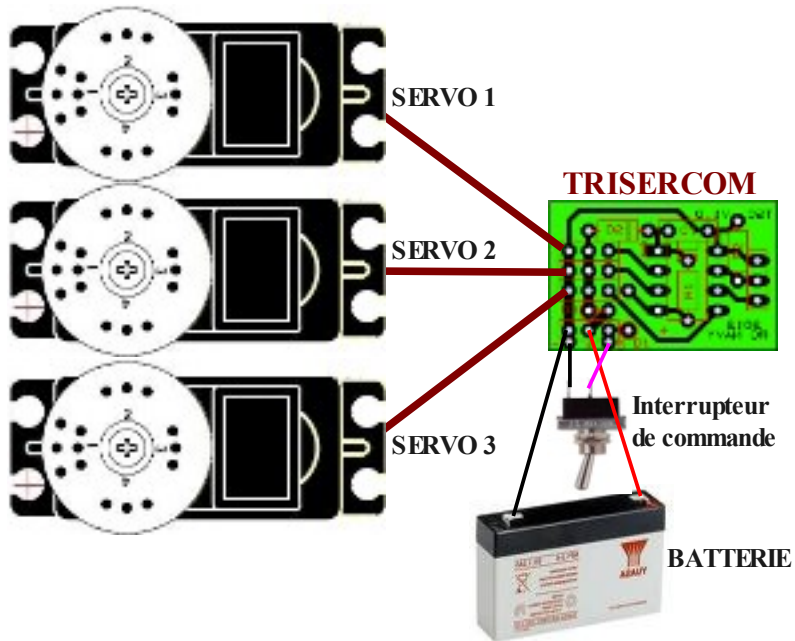
V.1. CONNECTIQUE

REPERE SUR LE CIRCUIT IMPRIME	RÔLE
COMMAND: ENTREE	Cordon de servo vers voie du récepteur RC ou vers batterie + TOR de commande
SERVO1: SORTIE	3 picots pour accueillir le cordon du servo N°1
SERVO2: SORTIE	3 picots pour accueillir le cordon du servo N°2
SERVO3: SORTIE	3 picots pour accueillir le cordon du servo N°3
CONF: CORDON DE PARAMETRAGE	3 picots pour accueillir le cordon de paramétrage

VI. LES 3 TYPES DE COMMANDE POSSIBLES

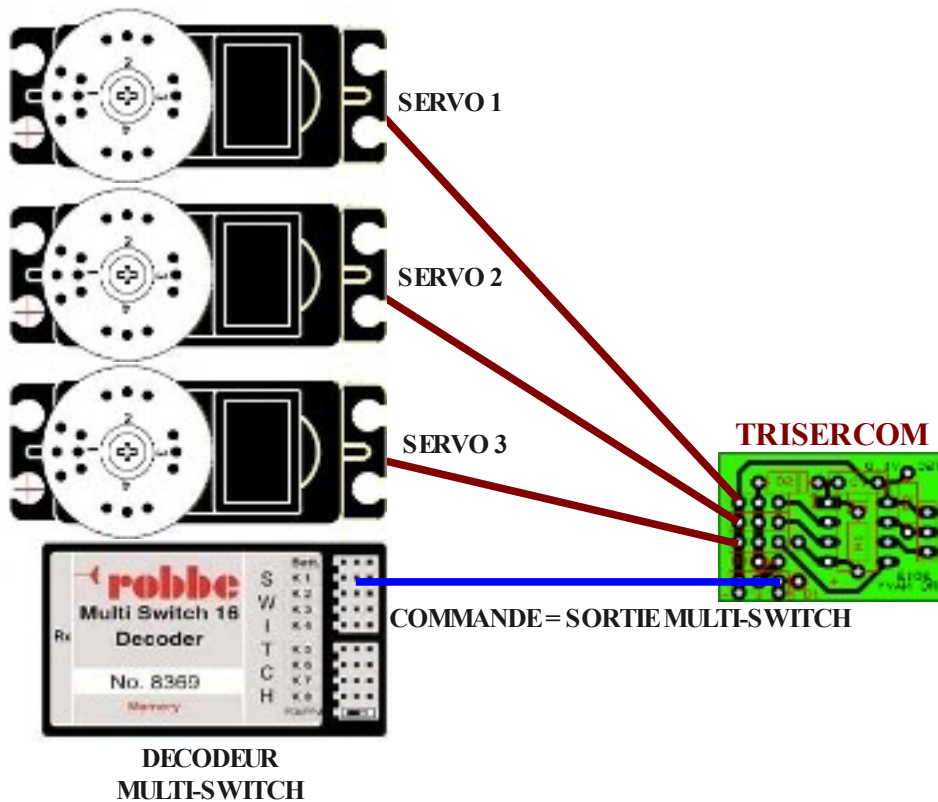
Utilisation	Signal de commande	Positions A	Positions B
1	Simple interrupteur ou contacts de relais	Contact fermé	Contact ouvert
2	Sortie transistorisée de type Tout-Ou-Rien : - collecteur ouvert - push-pull - totem-pole - inverseur CMOS - Multi-switch	$0 < \text{Tension} < 1,5 \text{ V}$	$2,5\text{V} < \text{Tension} < 24\text{V}$
3	Signal d'une voie proportionnelle issue d'un récepteur RC	Impulsion $< 1,5 \text{ ms}$ (au-dessous du neutre)	Impulsion $> 1,5 \text{ ms}$ (au-dessus du neutre)

VI.1. COMMANDE PAR UN SIMPLE INTERRUPTEUR

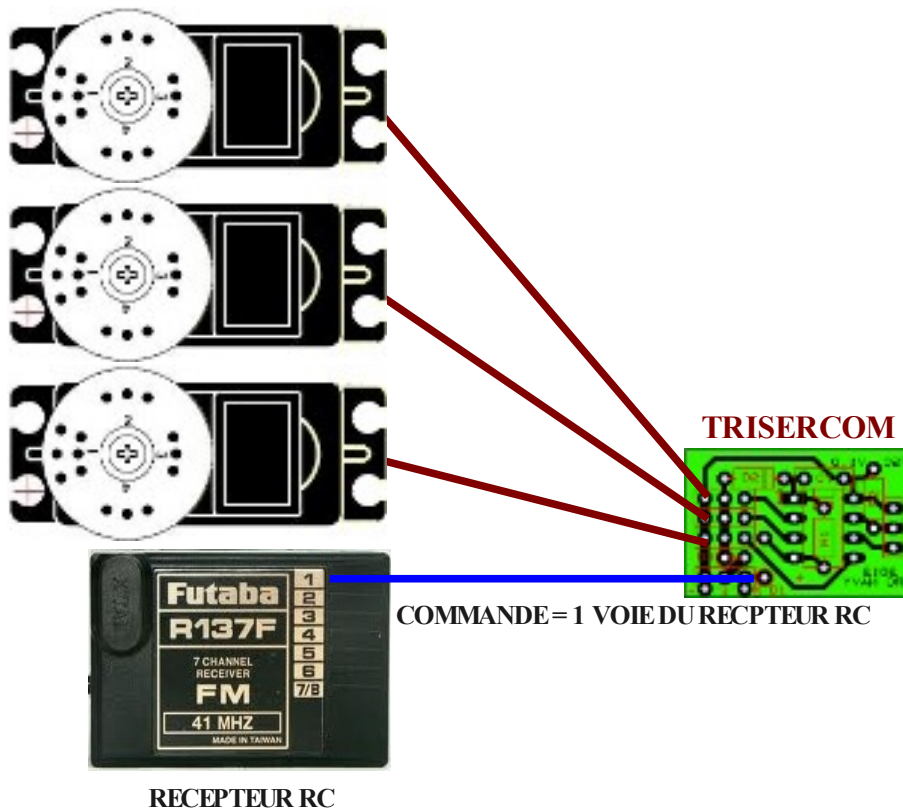


Dans cette configuration, aucun ensemble RC n'est nécessaire : le montage est autonome.

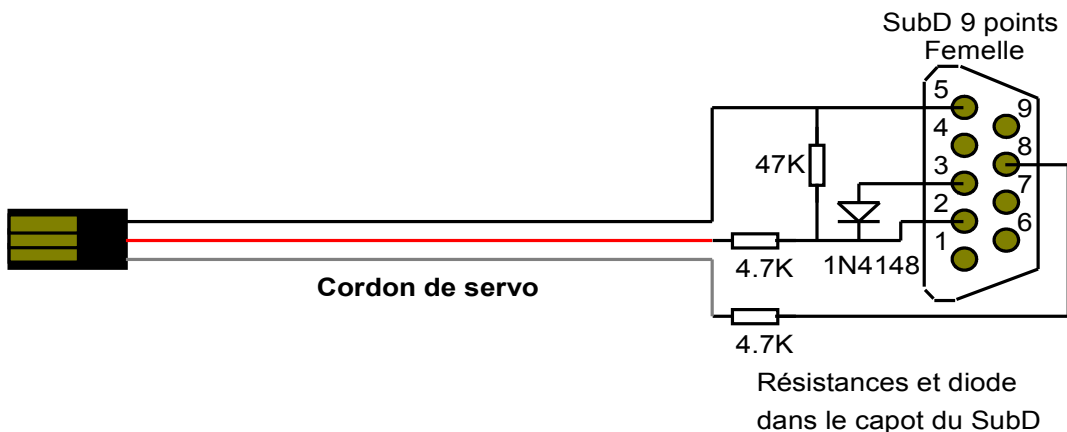
VI.2. COMMANDE PAR SORTIE TRANSISTORISEE



VI.3. COMMANDE PAR UNE VOIE D'UN RECEPTEUR RC



VII. CORDON DE PARAMETRAGE



Le cordon de paramétrage de [TRISERCOM](#) est strictement identique à celui de [SIMSERCOM](#). Associé à un PC et à un programme "Terminal", il permet d'ajuster finement :

- La position extrême A de chaque servo,
- La position extrême B de chaque servo,
- La vitesse pour aller de la position A à la position B de chaque servo,
- La vitesse pour aller de la position B à la position A de chaque servo,
- Le N° du servo qui doit déclencher le démarrage de chaque servo,
- La course que doit atteindre le servo en question pour déclencher le démarrage de chaque servo.

Une fois les paramètres ajustés et sauvegardés dans la mémoire EEPROM du micro-contrôleur PC,

ce cordon ne sert plus à rien, il doit être retiré du montage [TRISERCOM](#).

Il est bien sûr possible de retoucher le paramétrage ultérieurement sans avoir à recharger le programme (fichier HEX) dans le PIC. Il suffit juste de réutiliser le cordon de paramétrage pour entrer les nouveaux paramètres de fonctionnement.

Ce cordon de paramétrage n'est pas spécifique à [TRISERCOM](#), il est désormais utilisé pour tous les montages de l'auteur qui se paramètrent par une liaison série RS232 : c'est devenu un "standard".

Par exemple, ce cordon est déjà utilisé pour paramétrer les montages [P&B](#) et [SIMSERCOM](#) de l'auteur.

VIII. COMMANDES ACCEPTEES EN MODE TERMINAL

Dans la table ci-dessous:

- "A" désigne la 1^{ère} position extrême de chaque servo,
- "B" désigne la 2^e position extrême de chaque servo,
- "S" désigne la vitesse de chaque servo,
- "s" désigne le N° de servo auquel s'applique la commande.

COMMANDE	ACTION	REMARQUE
As=DDDD	Pour le servo "s", programme la consigne de largeur d'impulsion de servo correspondant à la position extrême A.	Unité : μ s Réponse : OK
As+	Pour le servo "s", incrémente de 1 μ s la consigne de position extrême A.	Réponse : OK
As-	Pour le servo "s", décréméte de 1 μ s la consigne de position extrême A.	Réponse : OK
As?	Pour le servo "s", interroge la consigne courante de largeur d'impulsion de servo correspondant à la position extrême A.	Réponse: As=DDDD en μ s
Bs=DDDD	Pour le servo "s", programme la consigne de largeur d'impulsion de servo correspondant à la position extrême B.	Unité : μ s Réponse : OK
Bs+	Pour le servo "s", incrémente de 1 μ s la consigne de position extrême B.	Réponse : OK
Bs-	Pour le servo "s", décréméte de 1 μ s la consigne de position extrême B.	Réponse : OK
Bs?	Pour le servo "s", interroge la consigne courante de largeur d'impulsion de servo correspondant à la position extrême B.	Réponse: Bs=DDDD en μ s
Ss=DDDD	Pour le servo "s", programme la consigne vitesse pour aller de la position extrême A à la position extrême B.	Unité : μ s Réponse : OK
Ss+	Pour le servo "s", incrémente de 1 μ s la consigne de vitesse	Réponse : OK
Ss-	Pour le servo "s", décréméte de 1 μ s la consigne de vitesse	Réponse : OK
Ss?	Pour le servo "s", interroge la consigne courante de vitesse	Réponse: Ss=DDDD en μ s
Vs=DDDD	Pour le servo "s", programme la consigne vitesse pour aller de la position extrême B à la position extrême A.	Unité : μ s Réponse : OK

Vs+	Pour le servo "s", incrémente de 1 μ s la consigne de vitesse	Réponse : OK
Vs-	Pour le servo "s", décrémente de 1 μ s la consigne de vitesse	Réponse : OK
Vs?	Pour le servo "s", interroge la consigne courante de vitesse	Réponse: Ss=DDDD en μ s
Ts=M	Pour le servo "s", programme le N° de servo qui va servir de déclencheur pour le mouvement A vers B.	Réponse : OK (M=1, 2 ou 3 et 0 pour départ immédiat)
Ts?	Pour le servo "s", interroge le N° de servo qui va servir de déclencheur pour le mouvement A vers B.	Réponse: Ts=M
Hs=N	Pour le servo "s", programme le N° de servo qui va servir de déclencheur pour le mouvement B vers A.	Réponse : OK (N=1, 2 ou 3 et 0 pour départ immédiat)
Hs?	Pour le servo "s", interroge le N° de servo qui va servir de déclencheur pour le mouvement B vers A.	Réponse: Hs=N
Cs=DDDD	Programme le seuil de largeur d'impulsion que doit atteindre le servo M pour que le servo "s" démarre le mouvement A vers B.	Unité : μ s Réponse : OK
Cs+	Incrémente de 1 μ s le seuil de largeur d'impulsion que doit atteindre le servo M pour que le servo "s" démarre le mouvement A vers B.	Réponse : OK
Cs-	Décrémente de 1 μ s le seuil de largeur d'impulsion que doit atteindre le servo M pour que le servo "s" démarre le mouvement A vers B.	Réponse : OK
Cs?	Interroge le seuil de largeur d'impulsion que doit atteindre le servo M pour que le servo "s" démarre le mouvement A vers B.	Réponse: Cs=DDDD en μ s
Ds=DDDD	Programme le seuil de largeur d'impulsion que doit atteindre le servo N pour que le servo "s" démarre le mouvement B vers A.	Unité : μ s Réponse : OK
Ds+	Incrémente de 1 μ s le seuil de largeur d'impulsion que doit atteindre le servo N pour que le servo "s" démarre le mouvement B vers A.	Réponse : OK
Ds-	Décrémente de 1 μ s le seuil de largeur d'impulsion que doit atteindre le servo N pour que le servo "s" démarre le mouvement B vers A.	Réponse : OK
Ds?	Interroge le seuil de largeur d'impulsion que doit atteindre le servo N pour que le servo "s" démarre le mouvement B vers A.	Réponse: Ds=DDDD en μ s
R	Enregistre les paramètres courants dans la mémoire EEPROM du micro-contrôleur	Réponse : OK

Notes importantes:

- Les commandes agissent en dynamique, c'est-à-dire, qu'il n'est pas nécessaire d'arrêter puis de remettre sous tension [TRISERCOM](#) pour que les nouveaux paramètres prennent effet,
- Une fois les paramètres ajustés, ne pas oublier d'envoyer la commande **R** (**R**ecord=**E**n**R**egistrer) pour les sauvegarder dans la mémoire EEPROM.
- Plus on augmente la valeur de **S** (**S**peed=**V**ite**S**se) et **V** (**V**itesse), et plus le servo est rapide : en fait il s'agit du pas d'incrément ou de décrétement qui est appliqué au servo toutes les :
 - 20 ms environ (période fixée par l'émetteur si l'entrée de [TRISERCOM](#) est une voie du récepteur),
 - 30 ms si l'entrée de [TRISERCOM](#) est une commande TOR(Tout-Ou-Rien)
- La dernière position extrême atteinte est mémorisée en EEPROM interne. Ainsi, à la mise sous tension, les servos ne bougent pas.

IX. REALISATION

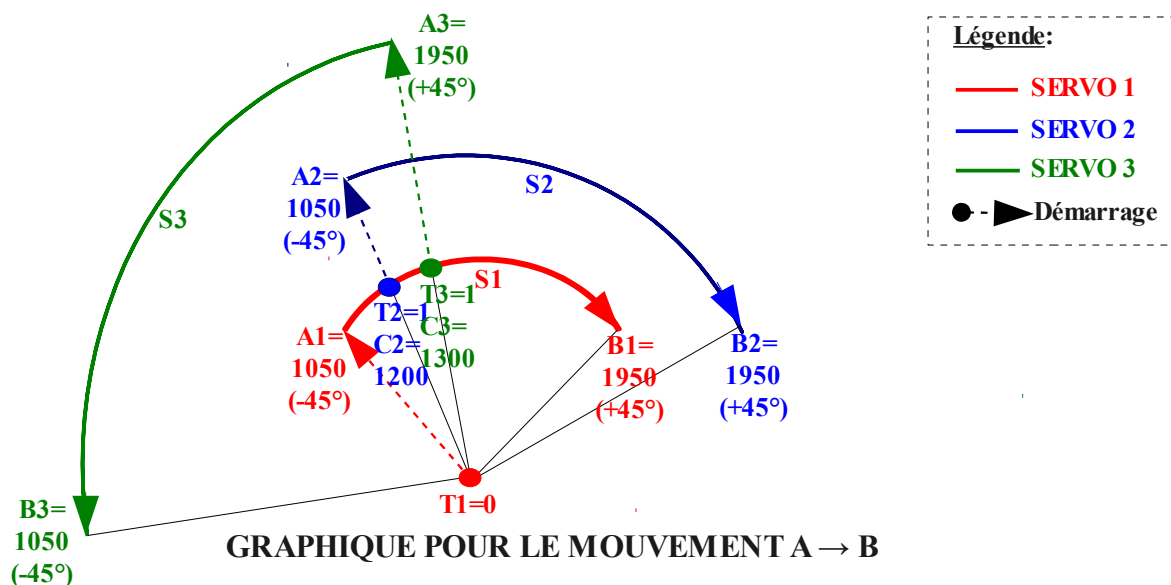
1. Faire le circuit imprimé décrit au §V,
2. Charger le programme "**trisercom.hex**" dans la mémoire Flash du PIC : c'est l'opération la plus délicate. Pour obtenir le fichier "**trisercom.hex**", contacter l'auteur sur le site: [RC NAVY](#)
3. Réaliser le cordon de paramétrage tel que décrit au §VII, et le connecter sur [TRISERCOM](#) et sur un port série (port COM) du PC.
Sur les PC récents, il n'y a plus de port COM RS232. La solution consiste à investir dans un convertisseur USB→RS232. On en trouve pour quelques € sur ebay, par exemple.
4. Mettre [TRISERCOM](#) sous tension,
5. Sous Window\$, pour paramétrer [TRISERCOM](#) (régler la valeur des positions extrêmes et la vitesse), 2 solutions sont possibles:
 - Utiliser l'interface graphique conviviale "**trisercom.exe**": aller directement au §X,
 - Utiliser le programme HyperTerminal intégré à Window\$ XP ou TeraTerm pour Window\$ VISTA et Window\$ SEVEN:
continuer ci-dessous.
6. Pour Window\$ XP, lancer l'HyperTerminal :
(Démarrer→Programme→Accessoire→Communication→Hyperterminal)
 - Ajuster les paramètres de communication à 2400, n, 8, 1
 - Activer le contrôle de flux : Hardware
7. Ajuster les positions extrêmes à l'aides des commandes définies dans la table au §VIII,
8. Une fois les réglages terminés, ne pas oublier d'envoyer la commande **R** pour sauvegarder les paramètres.

X. PARAMETRAGE A L'AIDE D'UN TERMINAL

X.1. EXEMPLE DE COMMANDE DE TRAIN D'UN HELICOPTERE AIRWOLF

- Hypothèse : la course des 3 jambes de trains (avant, gauche et droite) est de 90° . Afin de travailler autour du neutre mécanique du servo, les servos auront des positions extrêmes de 45° de part et d'autre du neutre (de -45° à $+45^\circ$, la course est bien de 90°),
- Une variation de $1\mu\text{s}$ sur la largeur d'impulsion de commande induit une variation d'environ 0.1° au niveau du palonnier du servo : $\text{Val}(\mu\text{s}) = 1500 + \text{Val}(\circ) \times 10$,
- Le servo N°1 correspond à la jambe de train avant,
- Le servo N°2 correspond à la jambe de train côté gauche,
- Le servo N°3 correspond à la jambe de train côté droit,
- Les servos N°2 et N°3 étant montés de manière symétrique, ils se déplacent en sens opposé : cette exemple illustre donc également la notion d'inversion de sens de rotation,
- Il ne s'agit que d'un exemple, il est évidemment possible de programmer des variantes.

X.1.1. Etapes de programmation pour le mouvement A → B (sortie train)



1. Programmer les **positions extrême A et B** de chaque servo:

A1=1050 (en effet, pour -45° , la valeur de l'impulsion en μs est : $1500 - 45 \times 10 = 1050$)

B1=1950 (en effet, pour $+45^\circ$, la valeur de l'impulsion en μs est : $1500 + 45 \times 10 = 1950$)

A2=1050 (-45°)

B2=1950 ($+45^\circ$)

A3=1950 ($+45^\circ$)

B3=1050 (-45°)

2. Programmer les **critères de démarrage** de chaque servo (immédiat ou après une certaine course d'un des autres servos) pour le mouvement A→B:

T1=0 (0 signifie départ immédiat du servo N°1 dès que la commande A→B est reçue)

T2=1 (signifie que le servo N°2 démarrera après une certaine course du servo N°1)

T3=1 (signifie que le servo N°3 démarrera après une certaine course du servo N°1)

C1=XXXX (Comme T1=0, la valeur de C1 n'a pas d'importance et est ignorée)

C2=1200 (cela signifie que le servo N°2 démarre dès que le servo N°1 a atteint -30°, la valeur en μs est : $1500 - 30 \times 10 = 1200$)

C3=1300 (cela signifie que le servo N°3 démarre dès que le servo N°1 a atteint -20°, la valeur en μs est : $1500 - 20 \times 10 = 1300$)

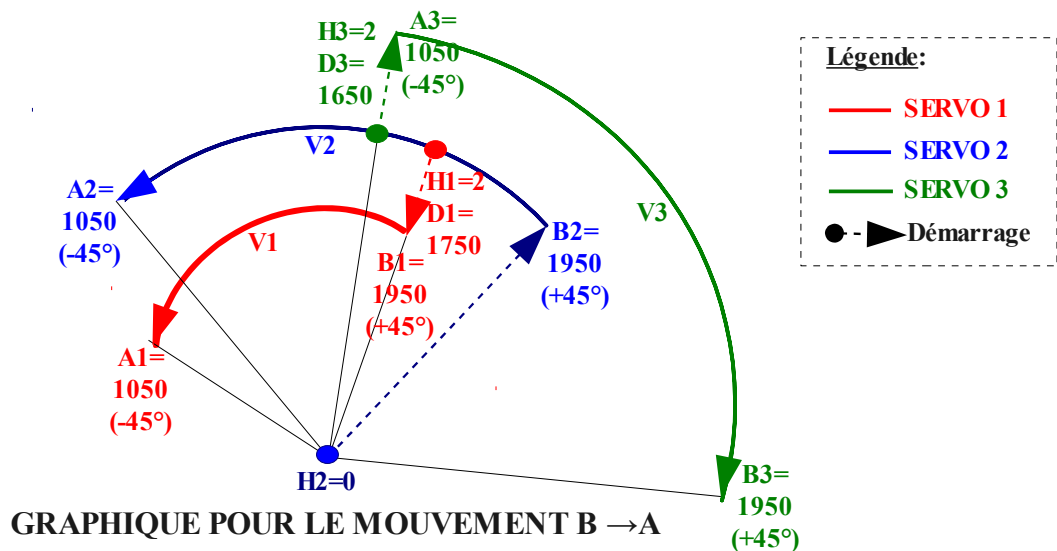
3. Programmer la vitesse de chaque servo pour le mouvement A→B:

S1=2 (signifie que l'impulsion de commande varie de $2\mu\text{s}$ toutes les 20ms environ si **TRISERCOM** est commandé par une voie d'un récepteur RC, soient $0.2^\circ/20\text{ms}=10^\circ/\text{s}$ ou de $2\mu\text{s}$ toutes les 30ms si **TRISERCOM** est commandé par une une sortie TOR, soient $0.2^\circ/30\text{ms}=6.7^\circ/\text{s}$)

S2=3 ($15^\circ/\text{s}$ si commande RC ou $10^\circ/\text{s}$ si commande TOR)

S3=5 ($25^\circ/\text{s}$ si commande RC ou $16.7^\circ/\text{s}$ si commande TOR)

X.1.2. Etapes de programmation pour le mouvement B→A (rentrée train)



1. Pas besoin de programmer les **positions extrême A et B** de chaque servo, (déjà fait):

A1=1050 (en effet, pour -45° , la valeur de l'impulsion en μs est : $1500 - 45 \times 10 = 1050$)

B1=1950 (en effet, pour $+45^\circ$, la valeur de l'impulsion en μs est : $1500 + 45 \times 10 = 1950$)

A2=1050 (-45°)

B2=1950 ($+45^\circ$)

A3=1950 ($+45^\circ$)

B3=1050 (-45°)

2. Programmer les **critères de démarrage** de chaque servo (immédiat ou après une certaine course d'un des autres servos) pour le mouvement B→A:
 - H1=2** (signifie que le servo N°1 démarrera après une certaine course du servo N°2)
 - H2=0** (0 signifie départ immédiat du servo N°2 dès que la commande B→A est reçue)
 - H3=2** (signifie que le servo N°3 démarrera après une certaine course du servo N°2)
 - D1=1750** (cela signifie que le servo N°1 démarre dès que le servo N°2 a atteint +25°, la valeur en μs est : $1500 + 25 \times 10 = 1750$)
 - D2=XXXX** (Comme T2=0, la valeur de D2 n'a pas d'importance et est ignorée)
 - D3=1650** (cela signifie que le servo N°3 démarre dès que le servo N°2 a atteint +15°, la valeur en μs est : $1500 + 15 \times 10 = 1650$)
3. Programmer la vitesse de chaque servo pour le mouvement B→A:
 - V1=1** (signifie que l'impulsion de commande varie de $1\mu\text{s}$ toutes les 20ms environ si **TRISERCOM** est commandé par une voie d'un récepteur RC, soient $0.1^\circ/20\text{ms}=5^\circ/\text{s}$ ou de $1\mu\text{s}$ toutes les 30ms si **TRISERCOM** est commandé par une une sortie TOR, soient $0.1^\circ/30\text{ms}=3.3^\circ/\text{s}$)
 - V2=2** ($10^\circ/\text{s}$ si commande RC ou $6.7^\circ/\text{s}$ si commande TOR)
 - V3=4** ($20^\circ/\text{s}$ si commande RC ou $13.3^\circ/\text{s}$ si commande TOR)
4. **R** (EnRegisterer les paramètres dans la mémoire EEPROM de **TRISERCOM**).

XI. PARAMETRAGE A L'AIDE DE TRISERCOM.EXE

Télécharger le paquetage d'installation "**trisercom.zip**" sur le site de l'auteur et l'installer sur le PC Window\$.

Lancer l'application "**trisercom.exe**" par :

Démarrer → Programmes → TRISERCOM → TRISERCOM

La fenêtre suivante apparaît :

Photos à insérer ici

XII. MENTIONS LEGALES

L'auteur n'est en aucun cas responsable des dommages qui pourraient découler de la mauvaise utilisation ou d'un éventuel dysfonctionnement du montage [TRISERCOM](#). Il appartient donc à l'utilisateur d'en mesurer et d'en assumer les risques.

XIII. REVISIONS DU DOCUMENT

- 04/02/2012 : création
- 12/02/2012 : ajout paramétrage de la vitesse qui devient indépendante de la direction,
- 26/02/2012 : ajout d'un exemple de paramétrage en mode Terminal (commande de train d'atterissage d'un hélicoptère Airwolf)
- 07/04/2012 : remplacement de D2 et D3 par strap si alimentation en 4.8V,
- 21/04/2012 : correction câblage du cordon de paramétrage et ajout de la mémorisation de la dernière position extrême atteinte.