

inverseur de sens de servo pour modèles radiocommandés

Mais qu'est-ce que c'est que ce bor***! !
 Je bouge le levier à gauche et mon modèle vire à droite ! Je pousse, il monte !
 Voici une solution simple à ce problème.
 Elle vous évitera bien des jurons.



Les servos sont des pièces électromécaniques asservies, cruciales sur beaucoup de modèles radiocommandés dont ils commandent la direction. Ils sont généralement commandés par modulation de largeur d'impulsion (MLI) : une impulsion de 1,5 ms correspond à la position neutre (« tout droit ») tandis que 1 ms et 2 ms représentent les positions extrêmes. Le sens de rotation d'un servo peut être inversé simplement, permettant à une impulsion de 1 ms de produire le même effet qu'une de 2 ms et vice versa. Tout ce qu'il y a à faire c'est changer la longueur des impulsions de commande.

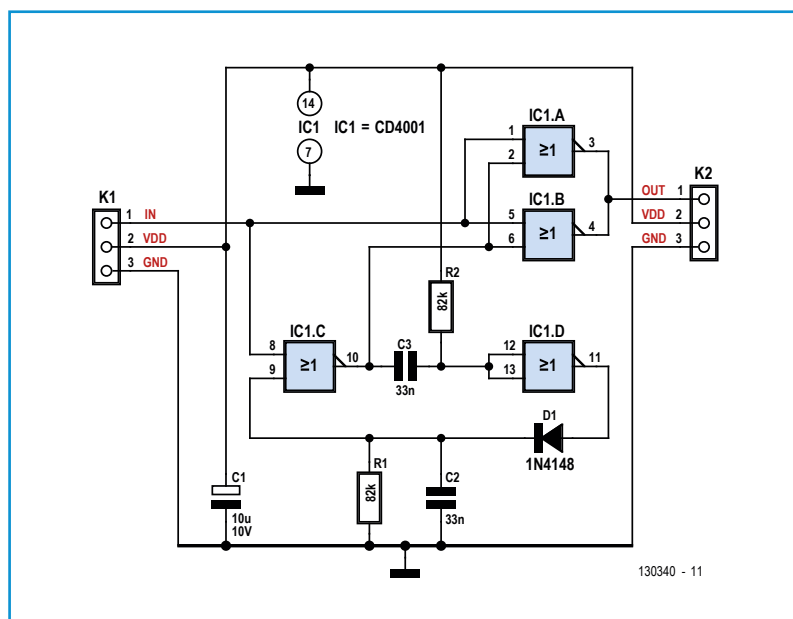
Étant donné que dans les deux cas, la position neutre correspond à une impulsion de 1,5 ms, il est possible de se baser sur cette valeur pour calculer la largeur des impulsions pour toutes les autres positions. Cela peut se faire plutôt simplement en soustrayant la largeur d'impulsion d'origine de 3 ms, ce qui donne la valeur pour la direction opposée. Pour inverser le sens de rotation d'un servo il suffit donc de soustraire les impulsions de commande d'une impulsion de référence de largeur 3 ms. On trouve deux types différents de servos : ceux pour lesquels l'impulsion de commande commence sur un front montant et ceux pour lesquels c'est un front descendant. Le circuit décrit ici ne convient que pour le premier type ; fort heureusement cela représente la grande majorité des servos disponibles dans le commerce.

Le matériel

Dans le circuit de la **figure 1** les portes NAND IC1c et IC1d d'un CD4001 couplées à C2, C3, R1, R2 et D1 forment un multivibrateur monostable dont les impulsions durent environ 3 ms. Il est déclenché par le signal de commande du servo (IN), qui attaque également une entrée de IC1b et IC1a. Les autres entrées de ces deux portes reçoivent les impulsions en logique négative d'une durée d'environ 3 ms depuis la sortie d'IC1c. Étant donné qu'IC1a et IC1b sont des portes NOR (non-OU) comme IC1c et IC1d, leurs sorties produisent une impulsion à logique positive dont la durée est de

Sunil Malekar
 (Elektor.Labs Inde)

Figure 1.
 Schéma de l'inverseur de sens de servo.



3 ms moins la durée de l'impulsion de commande d'origine. IC1a et IC1b sont câblés en parallèle pour augmenter la capacité de commande (en courant) du circuit.

Le monostable du prototype produisait des impulsions de 3,15 ms de manière fiable. Dans la plupart des applications, le servo est partie intégrante du modèle et il faudra régler la tolérance de la largeur des impulsions au niveau de l'émetteur. Si toutefois il vous fallait des impulsions durant exactement 3 ms, il faudrait changer C2 et C3 pour des condensateurs de 27 nF auxquels vous relieriez de petits condensateurs en parallèle jusqu'à obtenir exactement 3 ms sur l'oscilloscope. Si vous ne possédez pas d'oscillo, il reste possible de régler la largeur d'impulsion en modifiant légèrement la valeur de C2 et C3 jusqu'à ce que le servo reste exactement sur la position neutre, avec ou sans l'inverseur de sens.

En pratique

Le circuit est très simple à construire à l'aide de composants conventionnels sur un circuit imprimé à simple face (fig. 2). La carte terminée (fig. 3) pourra être testée avec une source d'impulsions réalisée avec

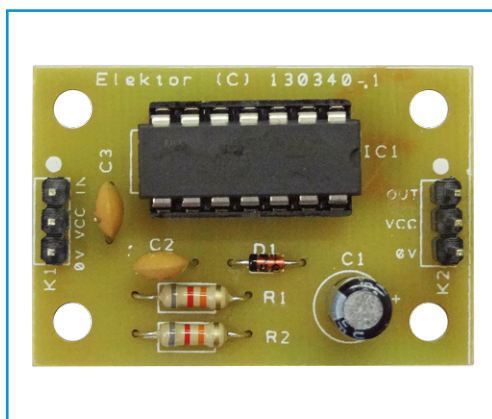


Figure 3.
La carte terminée.

Liste des composants

Résistances

R1, R2 = 10 kΩ

Condensateurs

C1 = 10 μF 10 V (de préférence au tantale)
C2, C3 = 33 nF

Semiconducteurs

IC1 = CD4001
D1 = 1N4148

Divers

K1, K2 = barrette sécable 3 voies au pas de 2,54 mm
C.I. réf. 130340

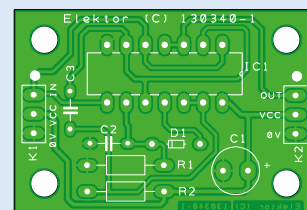


Figure 2.
Circuit imprimé de l'inverseur de sens de servo.

un Arduino (fig. 4) par exemple. Cet inverseur de sens consomme très peu de courant (1 mA), et n'est que très peu affecté (<2 %) par les variations de la tension d'alimentation entre 3 et 10 V. Il est recommandé d'utiliser un modèle au tantale pour C1 afin de réduire l'espace occupé par le montage. La configuration symétrique du circuit (R1 = R2; C2 = C3) lui confère un faible coefficient de température.

(130340 - version française : Kévin Petit)

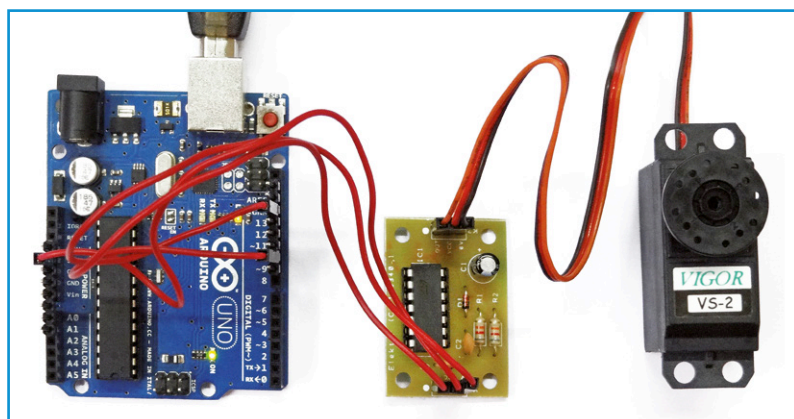


Figure 4.
Le bon fonctionnement du circuit a été vérifié à l'aide d'une carte Arduino.

