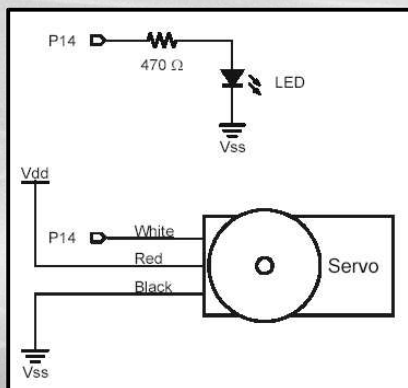




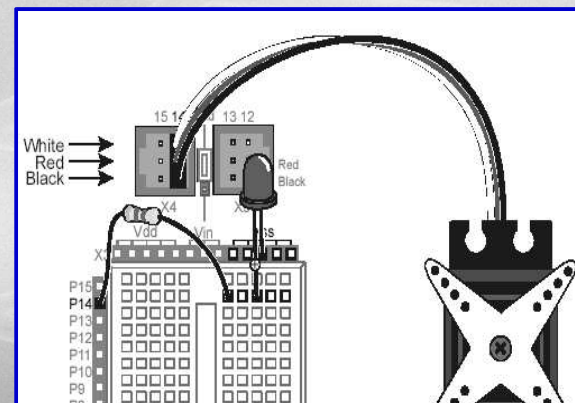
Connexions à au BASIC Stamp:



Attention : ne pas alimenter le servomoteur avec plus de 9V (risque de destruction).

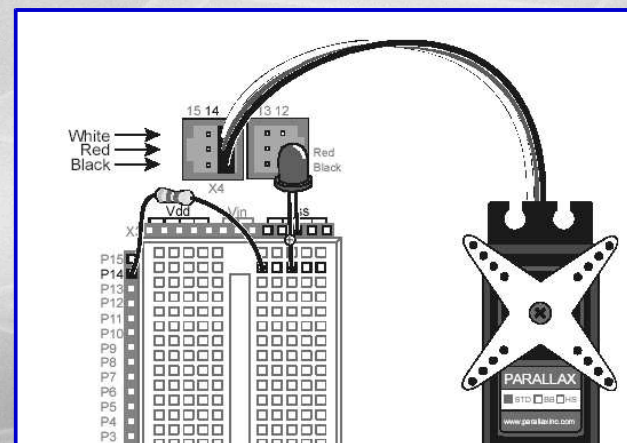


Connexion à la carte de développement



Connexion à la carte de développement

Débrancher la carte avant toute manipulation





Programmer le servo moteur

What's a Microcontroller?

Les servo sont controlés par des signaux de forme rectangulaire de 1 à 2 ms largeur (niveau haut) et espacés de 20 ms (niveau bas).

L'instruction PULSOUT est utilisée pour générer ces signaux:

PULSOUT *pin, durée*

pin: Définie quel E/S doit être utilisée.

durée: définie combien de temps l'impulsion doit durer (ce temps n'est pas en ms).



5



What's a Microcontroller?

Ce programme envoie un groupe d'impulsions de durée:

500 = 1 mS

750 = 1.5 mS

1000 = 2 mS

Exemple de code:

```
FOR counter = 1 TO 150
  PULSOUT 14, 1000
  PAUSE 20
NEXT
```

En fonctionnement, l'impulsion de 2ms est envoyée 150 fois avec des pauses de 20 ms.



7



What's a Microcontroller?

L'incrément de temps de PULSOUT est de 2 microsecondes (uS).

1 μ S = .000001 secondes.

1 mS = 1000 μ S

Pour la commande: PULSOUT 14,750

Le micro controleur envoie une impulsion de
 $750 \times 2 \mu\text{S} = 1500 \mu\text{S}$ ou 1.5mS
sur la pin 14.



6



What's a Microcontroller?

```
counter VAR Word
DEBUG "Counterclockwise 10 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 1000
  PAUSE 20
NEXT

DEBUG "Clockwise 2 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 500
  PAUSE 20
NEXT

DEBUG "Center 12 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 750
  PAUSE 20
NEXT

DEBUG "All done."
END
```



8



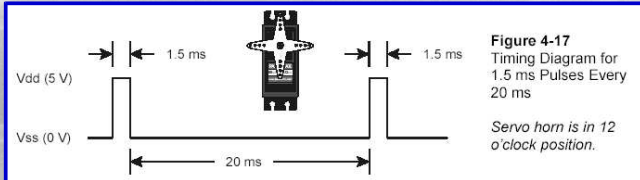
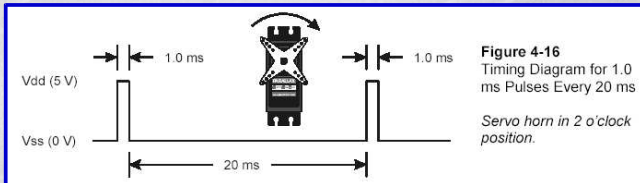
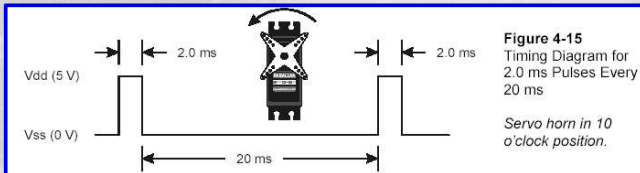
Avec un servo moteur classique, le servo devrait tourner de la position 10 heures à la position 2 heures pour finir à la position 12h.

La largeur des impulsions contrôle la position absolue du servo moteur (en général le max est de 90 ou 180 degrés).



La commande **DEBUGIN** permet de lire des données venant du PC via la liaison série, ex :

DEBUGIN DEC Pulses



```

' {$STAMP BS2}
' {$PBASIC 2.5}
counter VAR Word
pulses VAR Word
duration VAR Word

DO
  DEBUG CLS, "Enter number of pulses:", CR
  DEBUGIN DEC pulses
  DEBUG "Enter PULSOUT duration:", CR
  DEBUGIN DEC duration
  DEBUG "Servo is running...", CR
  FOR counter = 1 TO pulses
    PULSOUT 14, duration
    PAUSE 20
  NEXT
  DEBUG "DONE"
  PAUSE 1000
LOOP

```



Rotation en continu de servo moteurs (contrôle vitesse)

What's a Microcontroller?

Les servo standard utilise un mécanisme de feedback connecté à l'arbre de transmission. Lorsque la fréquence de contrôle est lue, la position courante de l'axe est comparée à la commande. Le circuit de contrôle interne au servo tente alors de réduire cette erreur en faisant tourner le servo dans le bon sens jusqu'à ce que l'erreur soit nulle.



13



Contrôler au clavier les déplacements de votre robot

What's a Microcontroller?

On enverra le chiffre 1 pour tout droit, 2 pour à droite et 3 pour à gauche.

Une fois ce programme terminer, écrire un nouveau programme permettant de faire réaliser à votre robot une trajectoire automatisée:

- décrire une trajectoire rectangulaire pendant 10s
- décrire une tajectoire circulaire pendant 10 s et recommencer

Pour finir, contrôler le changement de trajectoire par l'appuie sur le bouton du robot. A chaque fois que l'on appuie sur la touche le robot change de trajectoire.



15



What's a Microcontroller?

Les servos du robot sont modifier pour tourner de manière continue (suppression de la butée mécanique et déconnexion de la lecture de l'orientation de l'arbre de transmission. Le mécanisme de feedback a été ajusté à la main pour que le moteur s'arrète pour des impulsions de 750 or 1.5mS (valeur centrale).

Des impulsions plus longues ou plus courtes permettent de faire tourner le servo à une vitesse plus ou moins grande dans les 2 directions (remplace avantageusement une électronique de régulation de vitesse complexe tant que la locomotion peut se contenter de servo moteurs).



14