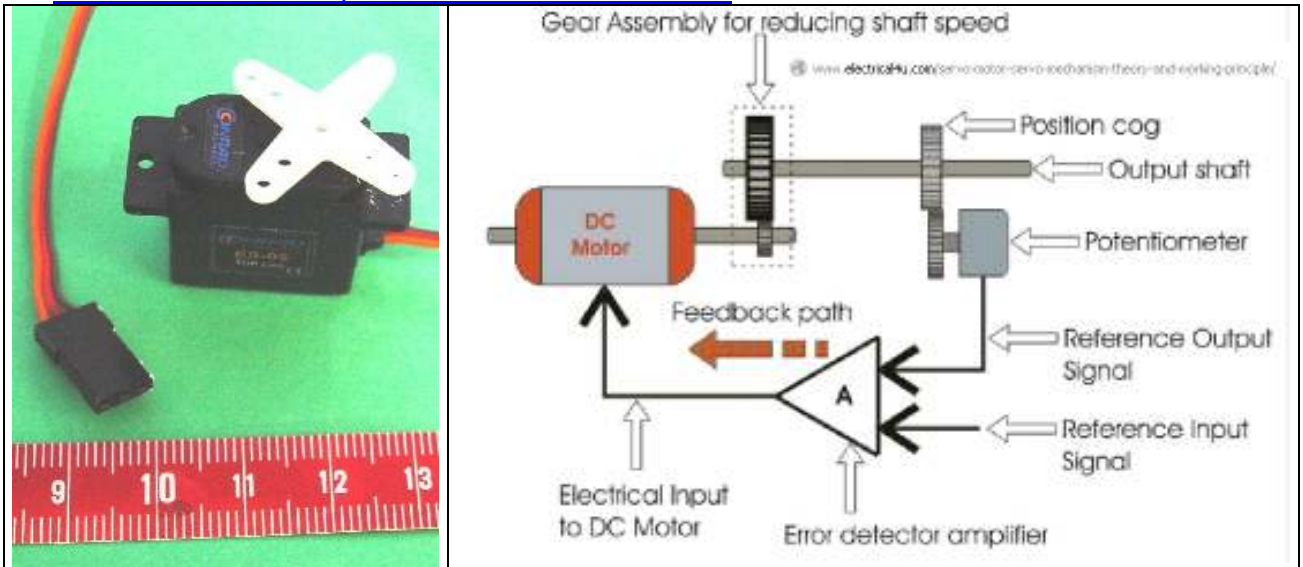
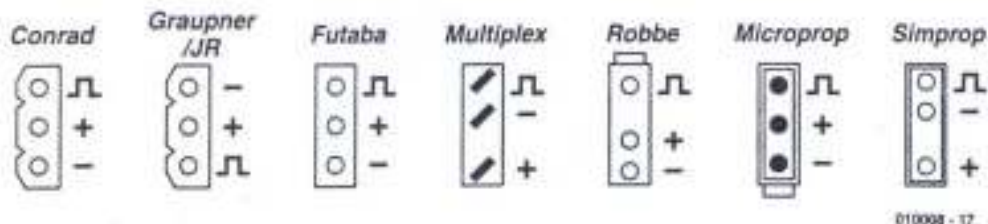


Servos de télécommande -- provisoire 140616

Le servos de télécommande sont bien connus <http://www.digitalspirit.org/wiki/docs/servo-moteurs>
 ou www.hooked-on-rc-airplanes.com/servo-tutorial.html



Leur position dépend d'une impulsion positive de 1 à 2ms, répétée toutes les 20ms.
 La tension d'alimentation est en général de 4 à 6V. Attention au brochage !



Librairie Arduino

La librairie servo est installée en standard sur Arduino.
 L'exemple ci-contre est explicite.
 Le timer 1 est utilisé. Il commande aussi le PWM sur les pins 9 et 10 (incompatibilité?).

0-180 degrés = 1000-2000 ms??

refresh() You must call this at least once every 50ms to keep the servos updated. You can call it as often as you like, it won't fire more than once every 20ms. When it does fire, it will take from .5 to 2.5 milliseconds to complete, but won't disable interrupts.

setMinimumPulse(uint16_t) set the duration of the 0 degree pulse in microseconds. (default minimum value is 544 microseconds)

setMaximumPulse(uint16_t) set the duration of the 180 degree pulse in microseconds. (default maximum pulse value is 2400 microseconds)

```
#include <Servo.h>

Servo myservo;
Servo mysecondservo;

void setup()
{
  // attaches a servo connected to pin 3
  myservo.attach(3);
  // attaches a servo connected to pin 15
  mysecondservo.attach(15);
}

void loop()
{
  // sets the servo position at 180 degrees
  myservo.write(180);
  // sets the servo position at 65 degrees
  mysecondservo.write(65);
}
```

? refresh dans la boucle?

Gestion par interruption 100 ms

Une action toutes les 100 us ne permet d'avoir que 20 pas de servo entre 0.7 ms et 2.6ms. La simplicité de la fonction à intégrer dans l'interruption 100 us encourage cette programmation s'il y a un seul servo.

```
void PosServo () {
  static int cnt;
  PORTC = cnt;
  cnt++;
  switch(next) {
    case Sstart: // On active le servo et prepare la duree
      ServoOn; cnt=0;
      next=Scnt;
      break;

    case Scnt: // On attend la fin de l'imp
      if(cnt > AngleIni + angle)
        { cnt=0; next=Sfin; }
      break;

    case Sfin: // On attend 20ms
      if(cnt > 200-angle,AngleIni)
        { cnt=0; next=Sstart; }
      break;
  }
}
```

Testeur microdules

Le microdule PoToD8
www.didel.com/08micro/PoToD8.pdf
permet de tester un servo entre 1 et 2ms.
Si le poussoir est maintenu pressé, la gamme est étendue à 0.7 à 2.6 ms



Servo 1.7g

www.didel.com/pic/Servos.pdf

www.didel.com/pic/Servos.pdf