

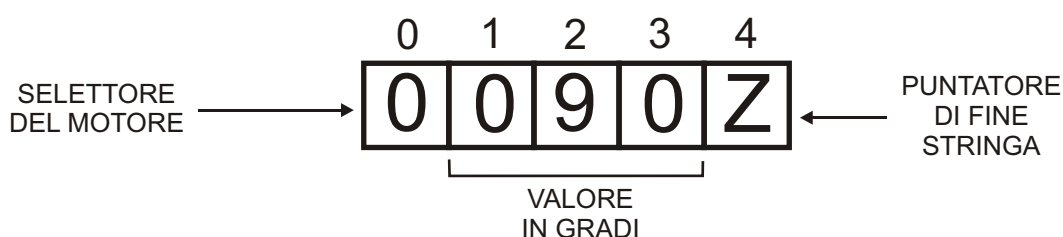
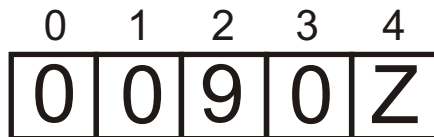
**Pilotaggio di un manipolatore senza scheda SSC32.**

Lo sketch seguente è relativo al pilotaggio di un braccio robotico della Linxmotion senza utilizzare la scheda SSC-32 USB. La prima parte del programma interpreta la stringa in arrivo (da C# o da App Inventor), estraendone il numero da fornire come parametro alla subroutine che si incaricherà di pilotare il servo corrispondente. La subroutine dovrà prevedere un algoritmo che identifichi la posizione precedente da quella che si vuole raggiungere per poter opportunamente incrementare o decrementare la rotazione del servomotore.

```
void loop()
{
  if(Serial.available()>0)
  //la seriale deve essere quella che veicola i dati
  {
    datoInput= Serial.readStringUntil('Z');
    //controlla la fine della stringa in arrivo
    delay(50);
    datoInputGradi=datoInput.substring(1,4);
    //estrae dalla stringa il secondo, terzo e quarto termine (1,2,3)
    //(nell'esempio scartando 0 e Z)

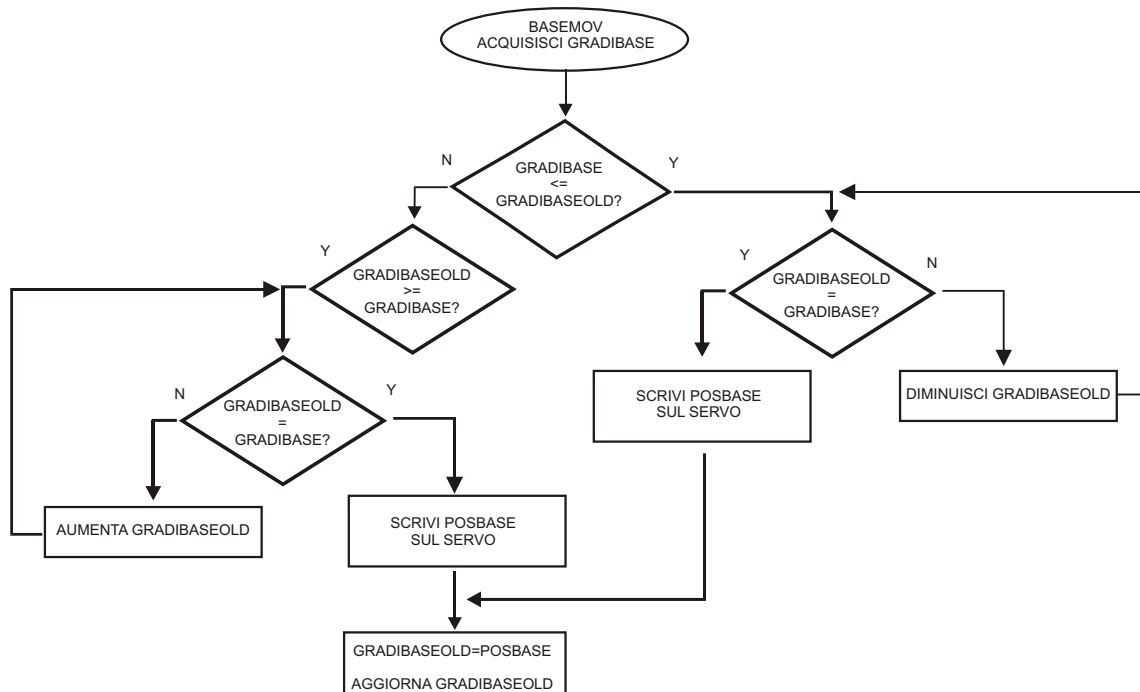
    delay(50);
    gradi=datoInputGradi.toInt();
    //trasforma la stringa in intero
    delay(50);

    if(datoInput.startsWith("0")==true)
    //effettua la selezione del motore (0=base)
    {
      basemov(gradi);
    }
    //chiama la sub basemov passando il parametro "gradi"
    delay(50);
  }
}
```



Passiamo ora all'analisi della subroutine.

```
void basemov(int gradibase)
//acquisisce il parametro gradi denominandolo "gradibase"
{
if(gradibase<=gradibaseold)
//se gradibase è minore di gradibaseold
{
    for (posbase = gradibaseold; posbase > gradibase; posbase --)
//diminuisce il valore di posbase fino a renderlo uguale a gradibase
    {
        base.write(posbase);
//scrivi posbase sul motore della base
        Serial.print("posbase e' ");
        Serial.println(posbase);
        delay(tempo);
//tempo regolerà la velocità di movimento del giunto
    }
    gradibaseold=posbase;
//aggiorna il valore de gradibaseold
}
else if (gradibase>=gradibaseold)
//se gradibase è maggiore di gradibaseold
{
    for (posbase = gradibaseold; posbase < gradibase; posbase ++ )
//aumenta il valore di posbase fino a renderlo uguale a gradibase
    {
        base.write(posbase);
//scrivi posbase sul motore della base
        delay(tempo);
    }
    gradibaseold=posbase;
//aggiorna il valore de gradibaseold
}
}
```



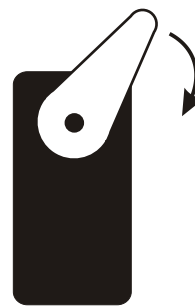
Ed ecco un esempio pratico:



Gradibaseold (la posizione precedente) è a  $135^\circ$ . Se il parametro passato "posbase" che prende il nome di "gradibase" è minore di  $135^\circ$ , il motore dovrà ruotare in senso antiorario.

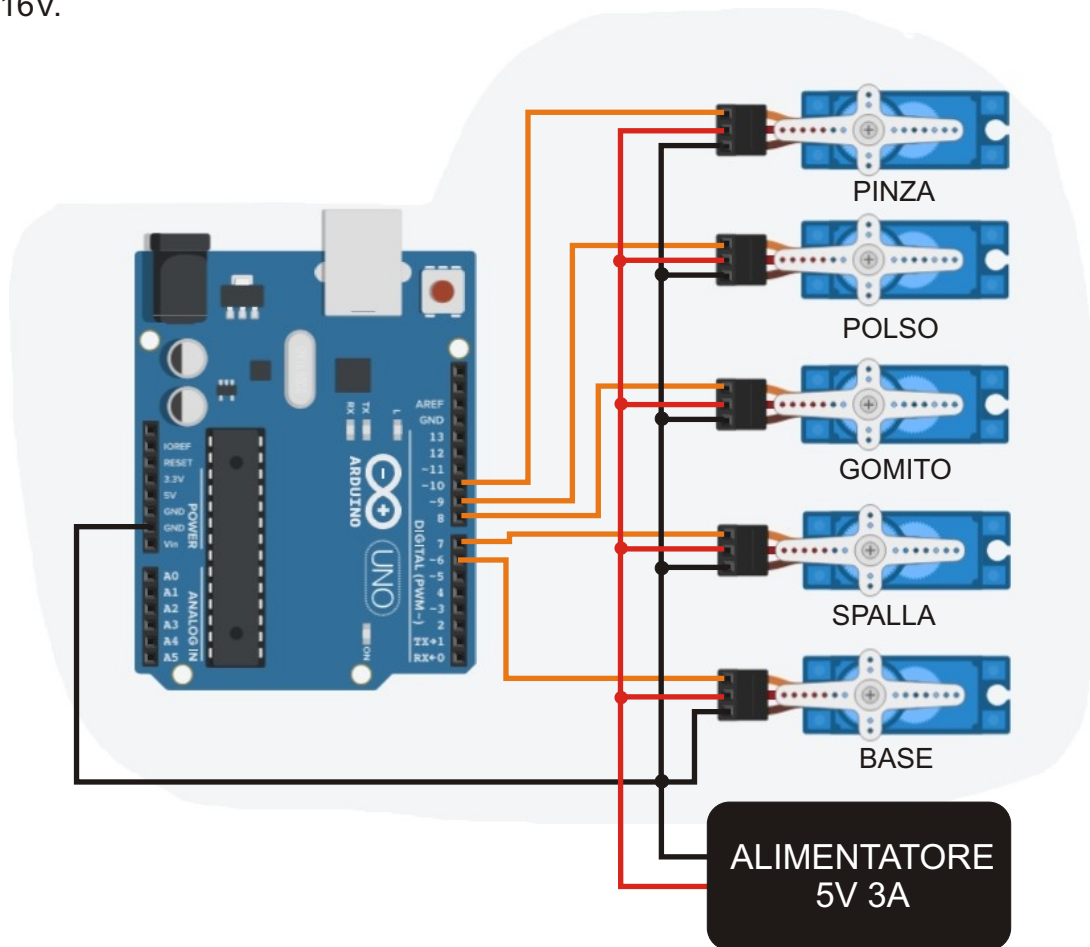


Gradibaseold (la posizione precedente) è a  $135^\circ$ . Se il parametro passato "posbase" che prende il nome di "gradibase" è uguale a  $135^\circ$ , il motore non si muoverà.



Gradibaseold (la posizione precedente) è a  $135^\circ$ . Se il parametro passato "posbase" che prende il nome di "gradibase" è maggiore di  $135^\circ$ , il motore dovrà ruotare in senso orario.

Per quanto riguarda l'alimentazione bisogna collegare i singoli motori al rispettivo pin di comando, mantenendo le alimentazioni in comune. Si consiglia un alimentatore a 5V da almeno 3A. In presenza di malfunzionamenti sullo spunto dei singoli servo, provare a collegare tra il + ed il - dei 5V un condensatore da 1000uF 16V.



Un esempio di uso del manipolatore con comando Bluetooth si può trovare sul libro **L@aborobotica vol B** a pagina 116.