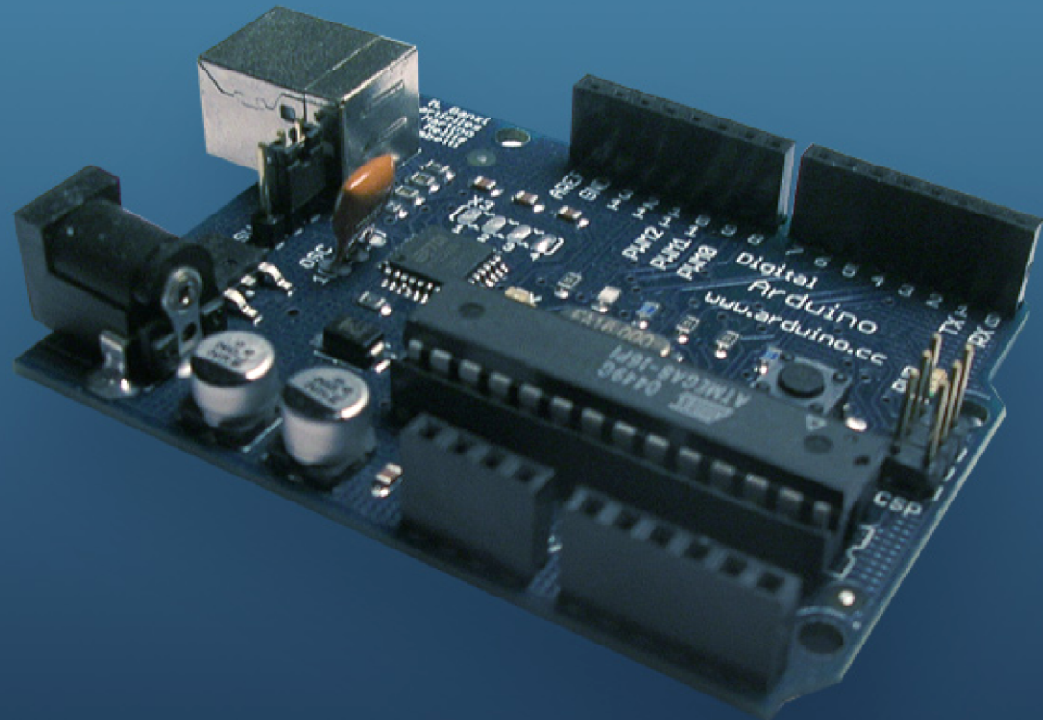


Arduino
Physical Computing I/O board



11[^] parte : Come gestire i servomotori con Arduino



Author: Ing. Sebastiano Giannitto (ITIS "M.BARTOLO" –PACHINO)

Un **servomotore** è un dispositivo capace di eseguire dei movimenti meccanici in funzione del segnale applicato al suo ingresso.

Sono in pratica dei dispositivi di precisione che servono per gestire la posizione angolare di un disco rotante o di un braccio meccanico.

Sostanzialmente un servo è composto da un motore elettrico, un motoriduttore ed un circuito di feedback per la gestione della posizione.

In commercio, esiste una vasta scelta di servi, ciascuno caratterizzabile per valore di coppia e precisione.

Quindi le caratteristiche comuni tra i servo sono principalmente, l'angolo di rotazione, la velocità di rotazione e la coppia motrice.

Il servo trattato in questo articolo è il **B2232**, un piccolo servo economico, ideale per i primi esperimenti didattici.

Questo tipo di servo ha un angolo di rotazione di circa 180° (la rotazione effettiva è un po' inferiore), inoltre esistono altri tipi di servi che hanno una rotazione continua e, anziché pilotarne la posizione, ne possiamo gestire la velocità.



Qualunque sia il servo scelto, ci troveremo davanti sempre **3 contatti** di cui due sono di alimentazione (V_+ e GND) mentre il terzo è il pin di controllo che nel nostro caso andrà collegato con un pin di arduino.

Per il servo in questione (i colori sono differenti dalla foto) il filo rosso corrisponde a V_+ , il filo viola è il GND mentre il filo giallo è quello del segnale.

Se vogliamo pilotare dei servo prima di tutto dobbiamo sapere che la complessità dell'opera nasce dalla necessità di utilizzare un **duty-cycle** per impartire comandi a questi meravigliosi oggetti in grado di trasformare la volontà elettronica in movimento.

Arduino dispone, secondo i modelli, di uscite PWM ossia pin digitali in grado di generare un duty-cycle utilizzabile per far muovere un servo.

Il modo più semplice per pilotare i servomotori è utilizzare una libreria, che traduca angoli in segnali e ci eviti di dover impazzire con i duty-cycle e i calcoli dei tempi; la libreria in questione è la [Servo](#) spiegata molto dettagliatamente sulla pagina ufficiale presente sul sito ufficiale dell'Arduino.



Questa classe ti mette a disposizione alcuni metodi che semplificano di molto l'obiettivo:

attach(): permette di specificare su quale pin è connesso il nostro servo e legarlo all'oggetto Servo;

attached(): controlla che un oggetto di tipo Servo sia collegata ad un pin;

detach(): rimuove il collegamento tra l'oggetto Servo e il pin a cui era legata;

read(): legge la posizione angolare del nostro servo, restituisce l'ultimo valore passato con write();

write(): impartisce al servo l'angolo a cui posizionarsi, su servo a rotazione continua imposta la velocità di rotazione 0=velocità massima in un senso, 90=fermo, 180=velocità massima nella direzione inversa;

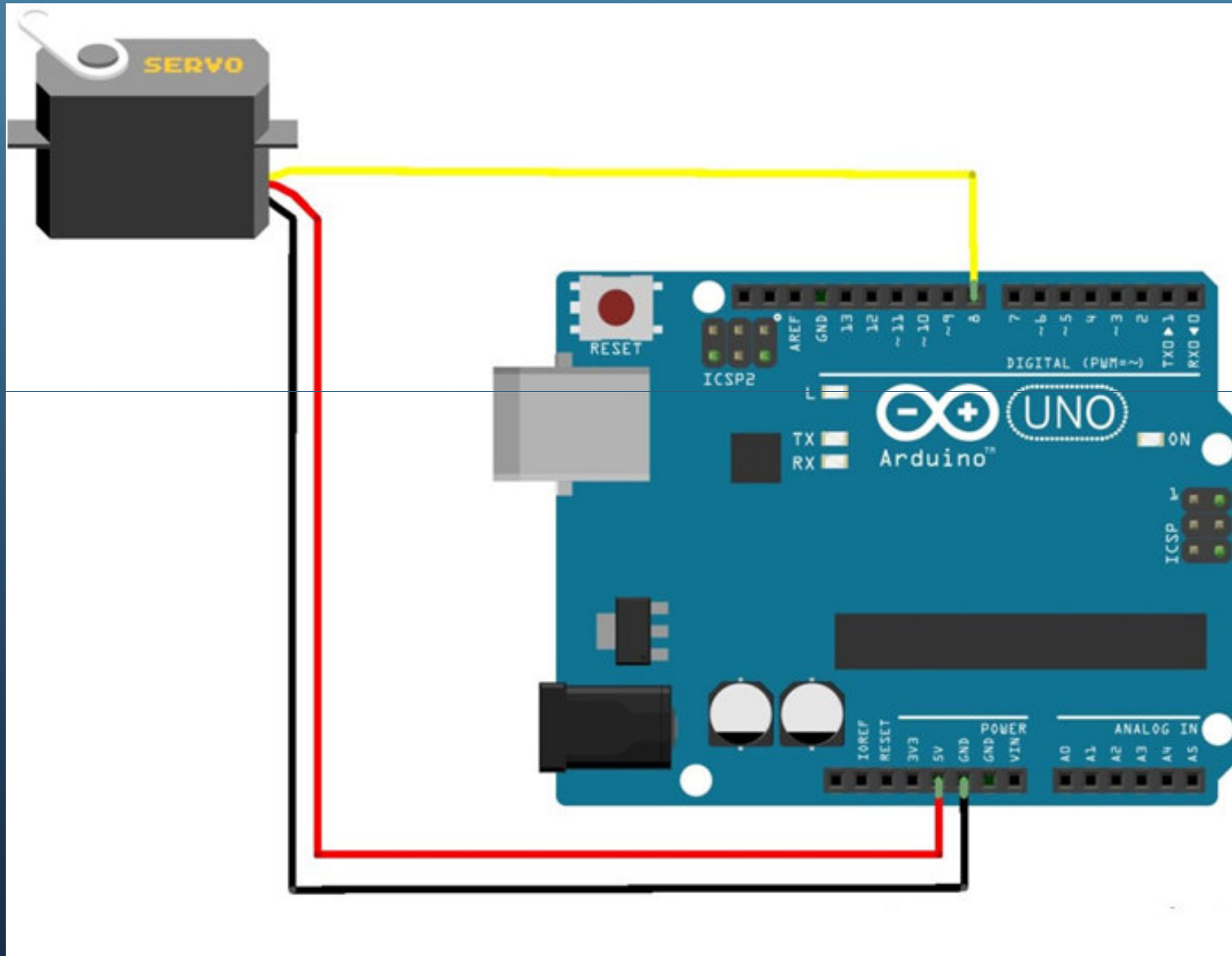
writeMicroseconds(): imposta la velocità di rotazione del servo, in un servo standard il valore va da 1000 a 2000, in un servo a rotazione continua si comporta allo stesso modo della write().

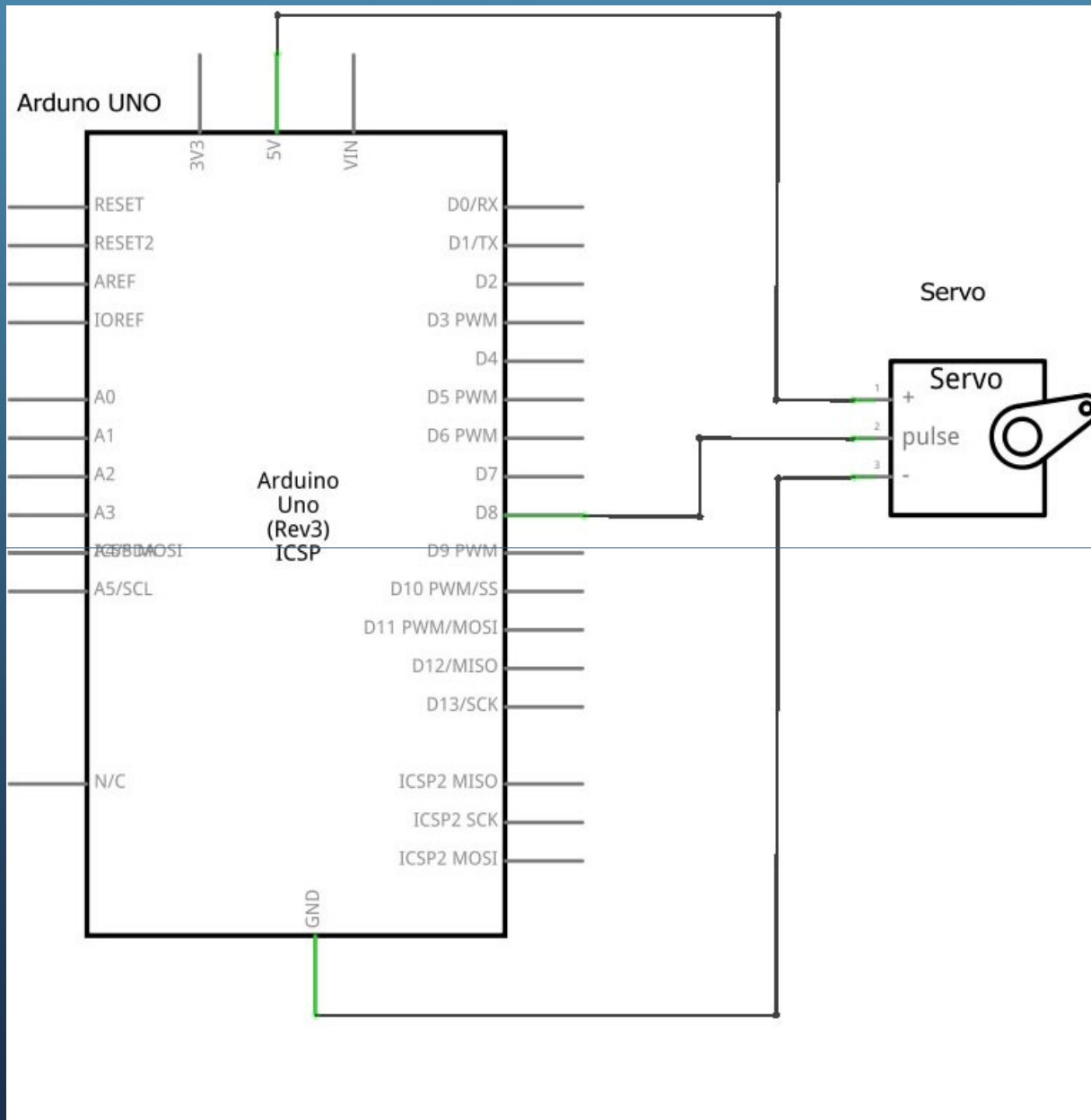
L'uso di questa libreria è molto semplice, vediamo quest'esempio preso direttamente dal sito.

Esempio 1

In questo esempio vedremo il servomotore compiere una rotazione da 0 a 180° e viceversa.

Incominciamo collegando il nostro servo come nella figura seguente:





E' importante tenere conto che con la classe Servo possiamo utilizzare tutti i pin digitali e analogici di Arduino e non solo le uscite di tipo PWM, questo disabilita la funzione analogWrite() normalmente utilizzabile con le uscite PWM di Arduino.

```
#include <Servo.h> // include la Libreria Servo.h
Servo myservo; // crea l'oggetto di tipo Servo, myservo sarà l'oggetto su cui opererai
int pos = 0; // inizializza una variabile di tipo intero pos il cui valore sarà la posizione da impartire al servo

void setup()
{ myservo.attach(8); // lega l'oggetto myservo al pin a cui abbiamo collegato il nostro servo, in questo caso il pin 8
}

void loop()
{ for(pos = 0; pos < 180; pos += 1) // imposta un ciclo con valori che vanno da 0 a 180,
// saranno i gradi di spostamento del nostro servo
{
myservo.write(pos); // con il metodo write() passi all'oggetto myservo la posizione che deve raggiungere,
// il servo si sposterà gradualmente dalla sua posizione 0° alla posizione 180°
delay(15); // imposta un ritardo di 15 millesimi di secondo per ogni ciclo del for.
// Più sarà alto il ritardo più il servo sarà lento.
}
for(pos = 180; pos >= 1; pos -= 1) // In questo caso imposta un ciclo con valori che vanno da 180 a 0
{
myservo.write(pos);
delay(15);
}
}
```


Con l'istruzione **Servo myservo;**

dichiariamo un oggetto di tipo Servo, e lo chiamiamo myservo.

Invece per indicare a quale pin è collegato l'oggetto myservo usiamo la funzione attach(), di cui possiamo vedere la sintassi all'interno del void setup().

Per pilotare il servo usiamo la funzione myservo.write(pos) dove con pos si indica la posizione in gradi.

Dopo aver visto la sintassi vediamo un esempio pratico.

Esempio 2

Nel secondo esempio commanderemo la rotazione di un servomotore con un potenziometro o trimmer.

