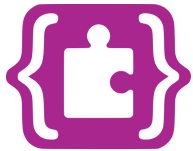


I SERVI FANNO MUOVERE LE COSE!

#R1AS13

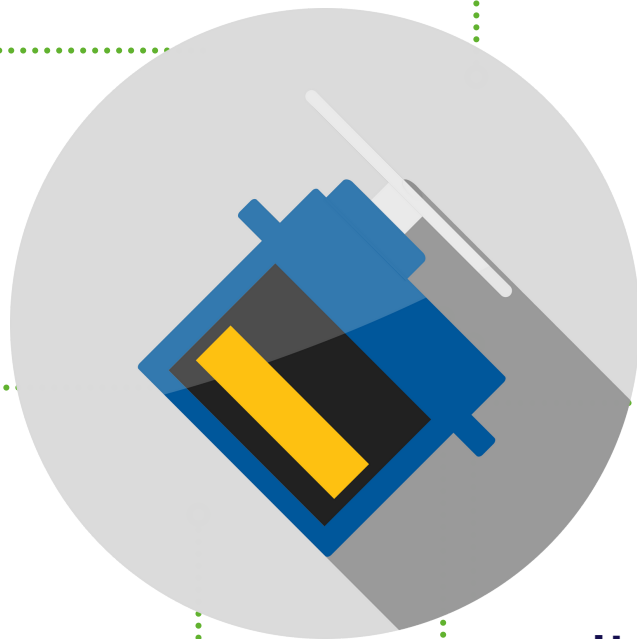


Disponibile su



Prerequisiti

- R1AS03 - Pulsanti e display a LED



Che cos'è?

Il servo è un driver per mantenere la posizione. È adatto a controllare un sistema con un cambiamento d'angolo costante e può mantenere il suo stato

Durata

25 minuti

Livello di difficoltà

Intermedio

Material

- 1 Scheda di programmazione "**STM32 IoT Node Board**"
- 1 cavo Micro-B USB
- 1 SG-90 Mini Servo(1.6kg)
- Fili di collegamento

OBIETTIVI DI APPRENDIMENTO

- Mettere in movimento un oggetto





Un Servo è un motore con un insieme di sistemi di controllo automatico, che consiste in un **motore DC** ordinario (motori elettrici rotativi che convertono l'energia elettrica in corrente continua in energia meccanica), un riduttore, un **potenziometro** (divisore di tensione usato per misurare il potenziale elettrico o la tensione) e un circuito di controllo. Può definire l'angolo di rotazione dell'albero di uscita inviando dei segnali. Di solito, un servo ha un angolo di rotazione massimo (per esempio, 180 gradi).

Risorse: https://en.wikipedia.org/wiki/DC_motor, <https://en.wikipedia.org/wiki/Potentiometer>

Il servo sistema può essere controllato da un impulso, che può cambiare la sua ampiezza. Usiamo un cavo di controllo per trasmettere l'impulso. Il ciclo di un segnale di riferimento servo è di 20ms e la larghezza è di 1,5ms. La posizione definita dal segnale di riferimento del servo è la posizione centrale. Poiché il servo ha un angolo di rotazione massimo, la definizione di posizione intermedia è da questa posizione dove il valore massimo e il valore minimo sono uguali.



PASSO 1 - ASSEMBLAGGIO DEI COMPONENTI



Collegare il servo alla scheda

Ci sono molti modi per collegare un servo alla tua scheda. Puoi usare qualsiasi pin di uscita analogica (pin PWM) per collegare il pin di controllo. Nel nostro esempio, useremo il pin **D4**. Il servo sarà collegato come segue:

- Nero per **GND**
- Rosso per **V+ (3V3)**
- Arancione per **SIG (D3)**

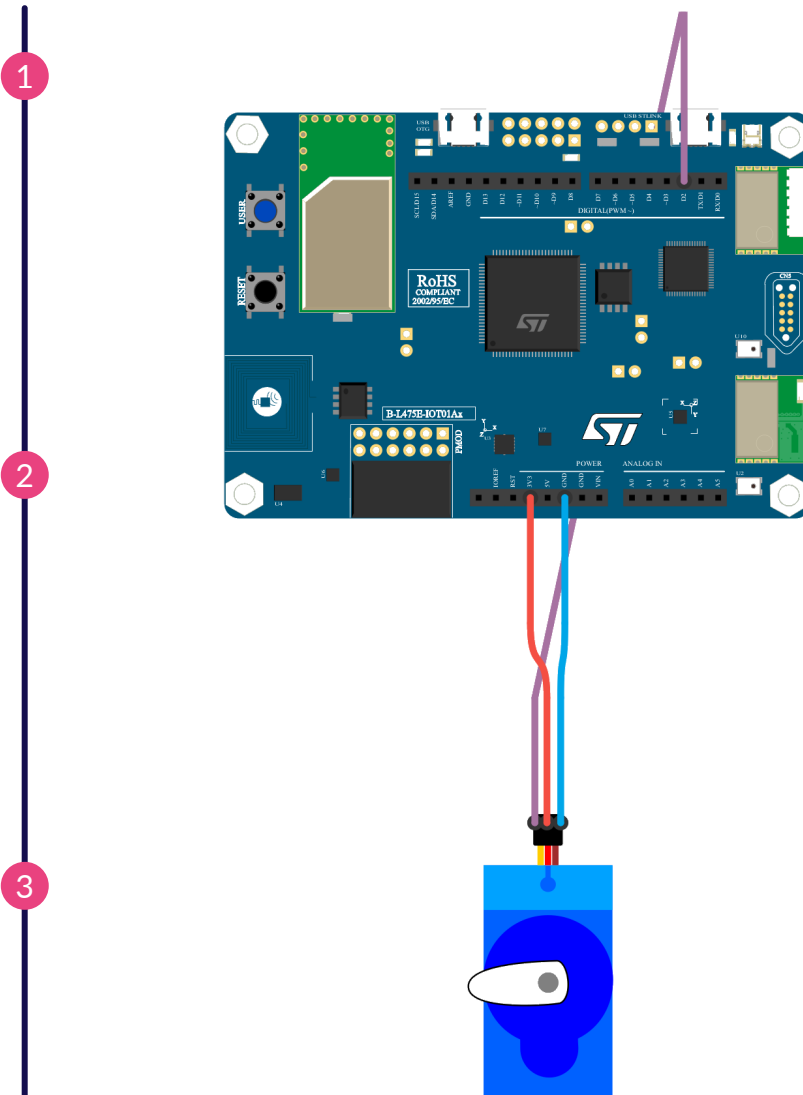
Collegare la scheda al computer

Con il tuo cavo USB, collega la scheda al tuo computer usando il **connettore USB ST-LINK** (nell'angolo destro della scheda). Se tutto sta andando bene dovresti vedere un nuovo drive sul tuo computer chiamato **DIS_L4IOT**. Questo drive è usato per programmare la scheda semplicemente copiando un file binario.

Aprire MakeCode e creare un nuovo progetto vuoto

Vai all'editor [Let's STEAM MakeCode](https://makecode.lets-steam.eu). Nella home page, crea un nuovo progetto cliccando sul pulsante "Nuovo progetto". Dai un nome al tuo progetto più espressivo di "Senza titolo" e lancia il tuo editor.

Risorsa: makecode.lets-steam.eu



Collegare il servo alla scheda



PASSO 1 - ASSEMBLAGGIO DEI COMPONENTI



Dopo aver creato il tuo nuovo progetto, otterrai la schermata predefinita "pronto a partire" mostrata qui.

Programma la tua scheda

All'interno del MakeCode Javascript Editor, copiate/incollate il codice disponibile nella **sezione Codice** qui sotto.

Prima di provare questo programma sulla scheda, potete provarlo direttamente nel simulatore. Se cambiate i valori 0 e 180, vedrete direttamente il risultato.

Se non è già stato fatto, pensate a dare un nome al vostro progetto e cliccate sul pulsante **"Scarica"**. Copia il file binario sul drive **DIS_L4IOT**, aspetta che la scheda finisca di lampeggiare e il tuo servo inizierà a muoversi!

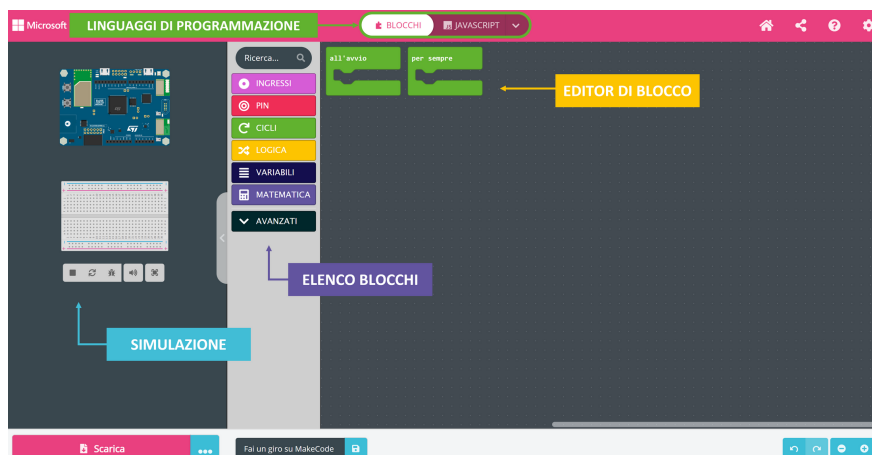
Eseguire, modificare, giocare

Il tuo programma verrà eseguito automaticamente ogni volta che lo salvi o resetti la tua scheda (premi il pulsante etichettato RESET).

Se tutto funziona bene, il vostro servo comincerà a muoversi.

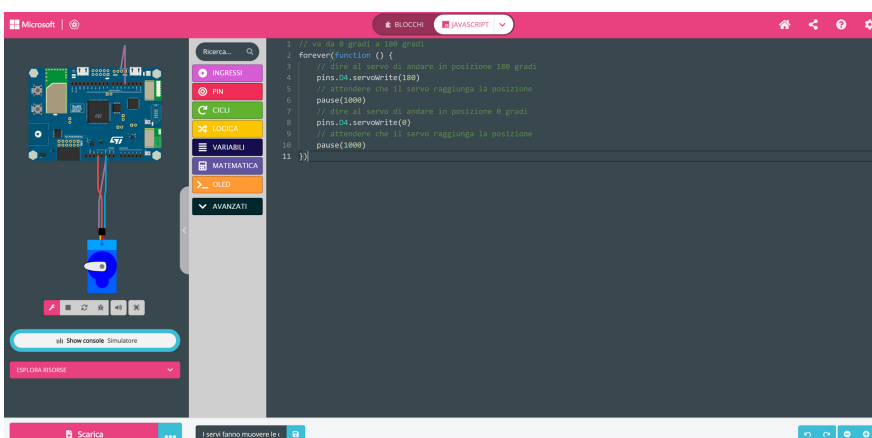
Cerca di capire l'esempio e inizia a modificarlo cambiando il periodo tra le due mosse.

4

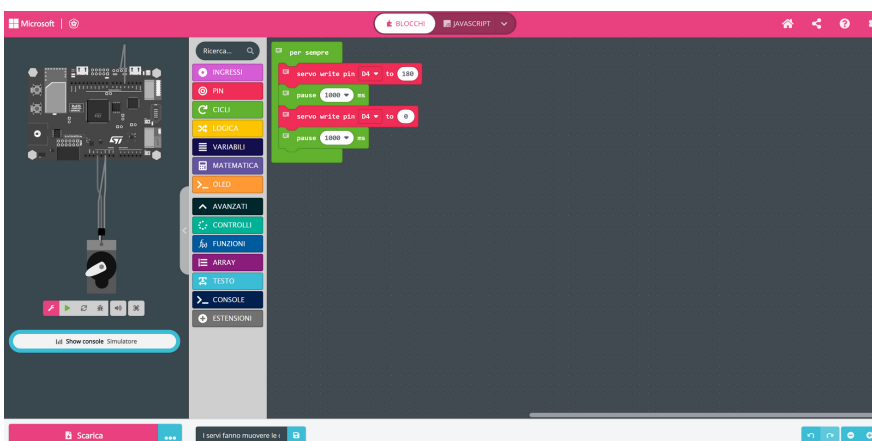


Schermata predefinita "ready to go"

5



Editor Javascript Makecode



Il servo inizia a muoversi



PASSO 3 - CODICE



```
// va da 0 gradi a 180 gradi
forever(function () {
  // dire al servo di andare in posizione 180 gradi
  pins.D4.servoWrite(180)
  // attendere che il servo raggiunga la posizione
  pause(1000)
  // dire al servo di andare in posizione 0 gradi
  pins.D4.servoWrite(0)
  // attendere che il servo raggiunga la posizione
  pause(1000)
})
```

Come funziona?

Questo esempio è abbastanza semplice, poiché si tratta del classico "blink" adattato a un servo.

L'istruzione principale è `pins.D2.servoWrite(XXX)`. Questa istruzione chiede al servo di ruotare ad un angolo di `XXX` gradi (come impostato dalle vostre esigenze specifiche a seconda del progetto che state sviluppando).

Per spostarsi tra due posizioni, il servo impiega un po' di tempo, quindi dobbiamo sempre aggiungere un ritardo prima di iniziare un altro movimento.

Questo programma spazza a destra e a sinistra per sempre!



Rispetto a un normale motore a corrente continua, un servo ruota solo entro una certa gamma di angoli, mentre un normale motore a corrente continua ruota in cerchio.

Un servo non può ruotare in cerchio. Un normale motore a corrente continua non può darci un feedback sull'angolo di rotazione, ma un servo può farlo. I loro usi sono quindi diversi.

I motori DC ordinari usano una rotazione del cerchio intero come potenza, mentre il servo usa un certo angolo di un oggetto controllato, come un giunto di un robot.



PASSO 3 - MIGLIORARE

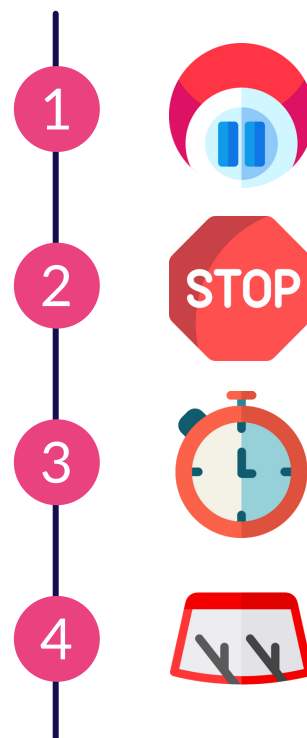


Cercate di ridurre il più possibile il **valore della pausa** per eliminare qualsiasi arresto del movimento.

Aggiungere le istruzioni per fare una **breve sosta nella posizione centrale**. Adattare il ritardo della pausa per essere sicuri che la fermata sia molto breve.

Trasformate questo programma per **fare un timer con un servo**. Ad ogni passo, spostare il servo di 3 gradi. Adattate il ritardo in modo che ogni passo richieda circa 1s.

Inizia lo **sweep move** solo quando il pulsante USER è stato cliccato.



ANDARE OLTRE



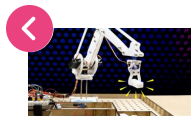
Servomotor - Per saperne di più sul meccanismo e il funzionamento di controllo del servomotore.
<https://en.wikipedia.org/wiki/Servomotor>



Servo motori con micro:bit - Tutto sui pulsanti e il loro uso in MakeCode con Shawn Hymel, Technical Content Creator.
<https://www.youtube.com/watch?v=okxooamdAP4&t=200s>, <https://shawnhymel.com>



DIY Color Sorting Robotic Arm - Impara a fare il tuo braccio robotico fai da te per ordinare i colori usando sensori a ultrasuoni e IR.
<https://thetempedia.com/project/diy-color-sorting-robotic-arm/>



Fogli di attività collegati

R1AS14 - Creare un timer per le uova

