

S4A: uno Scratch per Arduino



for



Versione 1.2
31/03/2017

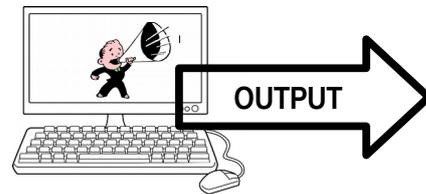
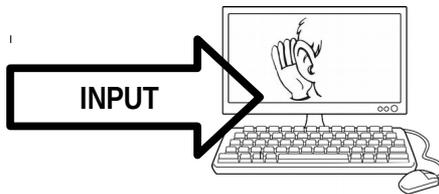
Schema dei canali

Per collegare un componente alla scheda arancione TinkerKit ricordarti che questo può essere:

1) di input (sensore)

oppure

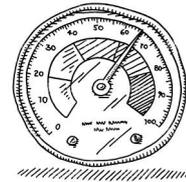
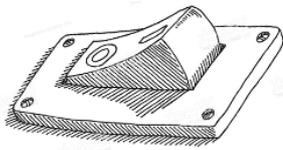
di output (attuatore);



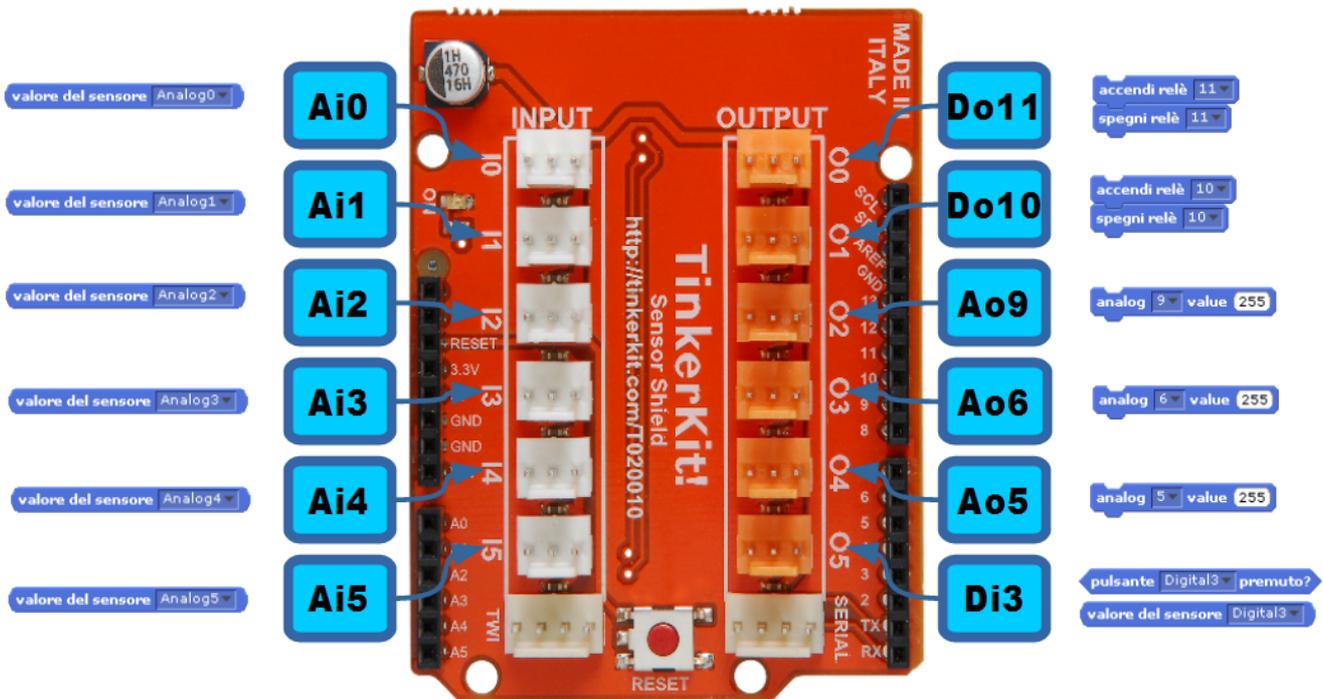
2) digitale (acceso/spento)

oppure

analogico (valori continui).



Capito quindi perché i canali hanno tutti dei nomi che cominciano per **Ai**, **Ao**, **Di** e **Do**?

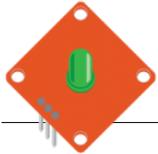
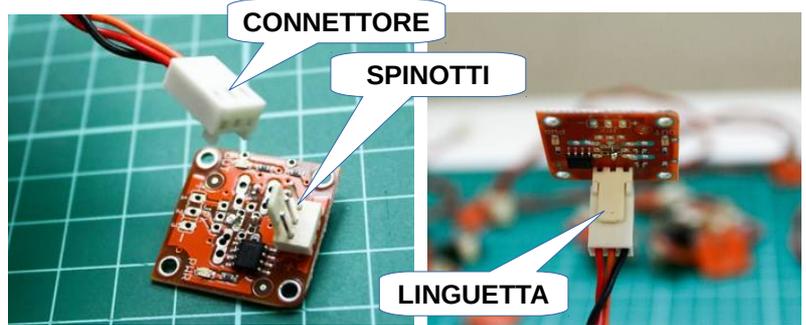




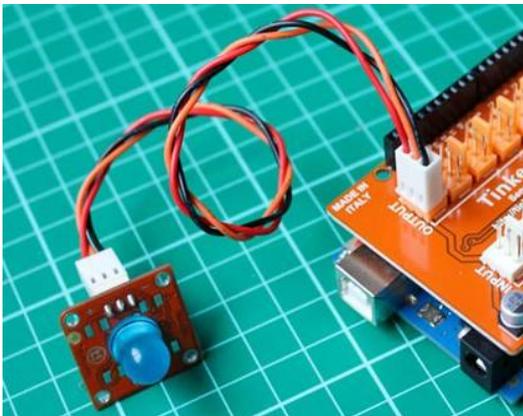
Connettori TinkerKit

Occhio alla linguetta!

- quando colleghi, devi spingere fino a sentire un *click*!
- quando stacchi i collegamenti devi sollevare la linguetta di plastica e sfilare il connettore.



Progetto 1 – LED lampeggiante



Il **LED** (light emitting diode) è un componente che emette luce ed è il più semplice elemento che possiamo controllare in uscita.

Collegiamolo al canale **Do11** e scriviamo in S4A lo script qui a destra.

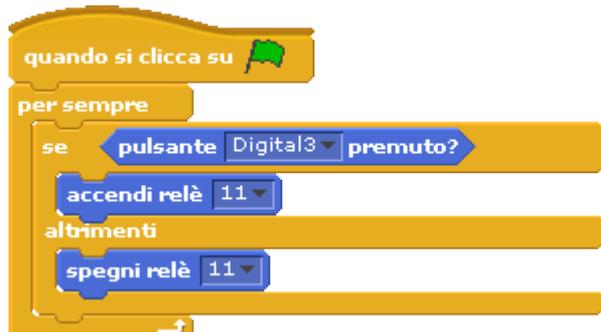
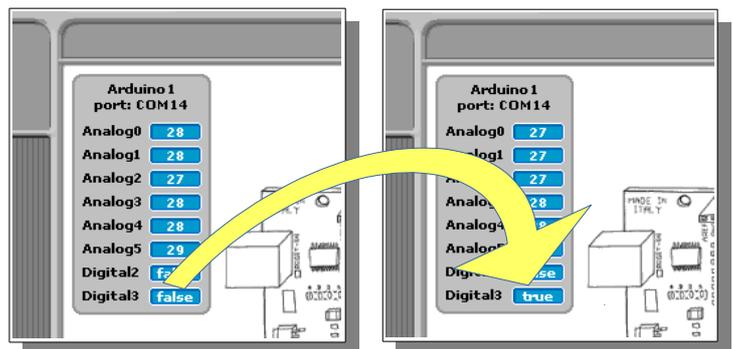
Attachiamo il cavetto USB, attendiamo che S4A si colleghi ad Arduino, clickiamo sulla bandierina verde e via!



Progetto 2 – Pulsante e LED

Il **pulsante** è il sensore più semplice: se lo colleghiamo ad un canale digitale di input, leggeremo TRUE quando è premuto e FALSE altrimenti.

Attachiamolo quindi al canale **Di3**: premendolo, vediamo che il valore del segnale Digital3 sulla board cambia valore.



Al posto dello script precedente, usiamo quello riportato qui a sinistra: così possiamo far accendere il LED (ancora attaccato al canale **Do11**) solo quando premiamo il pulsante.



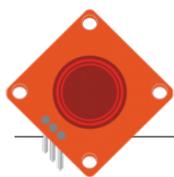
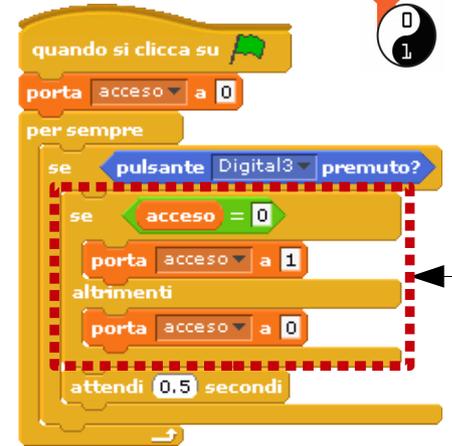
Sfida.

Dopo aver preparato una variabile di nome "acceso" (menù "Variabili", comando "Nuova variabile") sostituisci lo script precedente con i due script qui a destra.

Funziona tutto come prima? A cosa serve l'attesa di mezzo secondo? Se la togli, cosa succede?

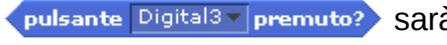
Per cambiare il valore della variabile "acceso" abbiamo usato il blocco "se" che sta più all'interno.

Riesci a trovare un modo molto più semplice per sostituire quel blocco "se" con una semplice operazione matematica? (Atsopsir: 



Progetto 3 – Pulsante «touch»

Anche il **pulsante "touch"** è un sensore digitale di input (ci dice "vero" o "falso", proprio come l'altro bottone): dovremmo utilizzarlo con il canale **Di3**, ma se questo è già occupato possiamo usare un canale di input analogico (ad esempio **Ai0**) e confrontare il valore letto con una soglia pari ad esempio a 500.

La condizione  sarà quindi sostituita da 

Per questo progetto, colleghiamo due LED ai canali **Do11** e **Do10**; colleghiamo come prima il bottone al canale **Di3** ed il pulsante "touch" all' **Ai0**.

Scriviamo poi il programma qui a fianco.

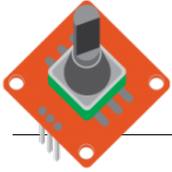


Cosa abbiamo realizzato? É in pratica il meccanismo per un gioco a quiz! Ogni concorrente controlla un LED ed il primo che preme il pulsante, lo fa accendere per 5 secondi: se durante questo tempo il giocatore non riesce a dare la risposta, l'altro giocatore può prenotarsi per rispondere.

Sfida.

Usa dei costumi diversi per mostrare anche a video quale giocatore che si è prenotato: mentre si legge la domanda, comincia con il costume del "?" poi passa ad un costume o all'altro in base al concorrente che si è prenotato per primo.





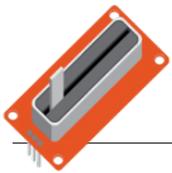
Progetto 4 – Potenzenziometro a rotazione

Collegiamo il **potenziometro a rotazione (o angolare)** al canale analogico di input **Ai0**: se giriamo piano la manopola vediamo che passa da valori piccoli (circa 10) a valori superiori a 900, attraversando tutti i valori intermedi.

Collegiamo poi un LED a **Ao9**: dopo aver creato la variabile “luminosità”, possiamo leggere il valore da **Ai0** (che teoricamente va da 0 a 1023), dividerlo per 4 ed arrotondarlo, per comandare il LED.



Questi calcoli servono, perché su un canale analogico di uscita possiamo scrivere solo dei valori interi da 0 a 255.



Progetto 5 – Potenzenziometro lineare

Anche il **potenziometro lineare** è un sensore che andrà collegato ai canali **Ai**. Sembra un po' il joystick di un videogioco, ed infatti in questo gioco lo useremo per controllare la posizione di uno sprite all'interno dello stage; dobbiamo solo ricordarci di trasformare i valori che leggeremo sul canale d'ingresso (valori al massimo tra 0 e 1023) in modo che il nostro personaggio si possa spostare nell'area del palcoscenico (le coordinate X vanno circa da -250 a 250): se colleghiamo il potenziometro al canale **Ai0** e dividiamo il valore di Analog0 per 2, otterremo valori compresi circa tra 0 e 500; se a questi sottraiamo 250, avremo l'intervallo di valori desiderato.

Siamo pronti per creare il nostro gioco!

- 1) per lo stage importiamo lo sfondo “forest”;
- 2) cambiamo il costume ad Arduino scegliendo quello della scimmia (“monkey1”);
- 3) aggiungiamo il suono “Waterdrop” (nel tab “Suoni”, lo troviamo nella directory “Effects”);
- 4) introduciamo un nuovo personaggio (troviamo lo sprite “bananas1” nella directory “Things”) e lo ridimensioniamo opportunamente;
- 5) infine aggiungiamo una variabile “Percentuale pancia piena” che ci servirà per il punteggio;
- 6) se già non l'hai fatto, connetti il potenziometro al canale **Ai0** e collega Arduino al PC col cavo USB.



I due personaggi “comunicano” tramite il messaggio “gnam!” ed hanno i seguenti script.



```

quando si clicca su
  porta Percentuale pancia piena a 0
  vai a x: 0 y: -110
  per sempre
    vai dove x è valore del sensore Analog0 / 2 - 230

quando ricevo gnam!
  cambia Percentuale pancia piena di 10
  produci suono WaterDrop
  se Percentuale pancia piena = 100
    dire Burp! per 2 secondi
    ferma tutto
  
```

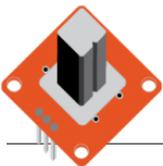
```

quando si clicca su
  per sempre
    se Percentuale pancia piena < 100
      mostra
      vai a x: numero a caso tra -200 e 200 y: 190
      ripeti fino a quando posizione y < -170 o sta toccando Arduino
        cambia y di -4
      se sta toccando Arduino
        nascondi
        invia a tutti gnam!
        attendi 1 secondi
    
```

Sfida.

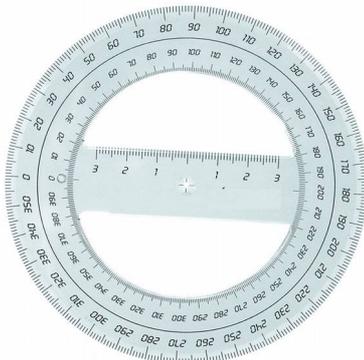
Modifica il gioco a piacere! Prova ad introdurre un nuovo sprite che cade ad intervalli casuali (magari delle banane acerbe che fanno venire il mal di pancia) e che fa perdere punti alla scimmia.

Per giocare con un amico, puoi anche provare ad aggiungere un secondo personaggio goloso di banane che potrà essere comandato dal potenziometro angolare collegato ad un altro canale Ai (siccome Arduino1 è l'unico sprite che conosce i valori dei sensori, per comunicarli agli altri personaggi dovrai fare uso di nuove variabili).



Progetto 6 – Tilt sensor

Il **tilt sensor** è un componente sensibile all'inclinazione: non può misurarla in gradi, ma grazie alla sferetta al suo interno (prova a scuoterlo per sentirla) ci dice se è orientato verso l'alto oppure se è ribaltato. Dovrà quindi essere collegato ad un canale **Di** (nel progetto qui sotto, usiamo il **Di3**).



Visto il suo funzionamento, possiamo fissarlo nel mezzo di un goniometro a 360° ed utilizzarlo come se fosse il manubrio di un'auto (certo, è un'automobile un po' strana, perché può andare solo a destra o a sinistra, ma mai dritta...).

Prova questo script! →

```

quando si clicca su
  hide board
  passa al costume car1_spr
  porta dimensione al 30 %
  vai a x: 0 y: 0
  punta in direzione 90
  per sempre
    fai 2 passi
    se pulsante Digital3 premuto?
      ruota di 3 gradi
    altrimenti
      ruota di -3 gradi
  
```

Se invece lo agganciamo al coperchio di una scatola possiamo utilizzarlo come “antifurto”, oppure potremmo fingere che sia una leva per controllare il ponte levatoio di un castello: usando sensori e costumi degli sprite possiamo fare questo ed altro!





Progetto 7 – Sensore di luminosità



La fotoresistenza è un **sensore di luminosità** che riesce a misurare l'intensità della luce nell'ambiente: vediamo ora come utilizzarlo per avere un programma che cambia automaticamente lo sfondo da giorno a notte e viceversa!

- 1) attacchiamo il sensore di luminosità ad **Ai0**;
- 2) colleghiamo un LED a **Do11**;
- 3) aggiungiamo questo script ad Arduino; nota che il LED viene acceso, quando la luminosità è bassa, proprio come i lampioni di sera!



- 4) infine nello **stage** importiamo i due sfondi "desert" e "stars" illustrati qui a fianco ed aggiungiamo i due brevi script che reagiscono ai messaggi inviati da Arduino.



Progetto 8 – Sensore di temperatura

L'ultimo componente che presentiamo è il termistore, ovvero il **sensore di temperatura**: lo attacchiamo ad **Ai0** ed osserviamo i valori sulla board. Una volta stabilizzato sulla temperatura ambiente, possiamo far variare i numeri tenendo il termistore tra le dita o anche solo alitandoci sopra!

Questo sensore reagisce più lentamente, ma se scegliamo in maniera attenta un valore di soglia, possiamo utilizzarlo in un gioco come abbiamo fatto prima col sensore di luminosità.

Cominciamo!

- 1) Se già non l'abbiamo fatto, colleghiamo il termistore al canale **Ai0**;
- 2) per lo stage scegliamo lo sfondo "castle";

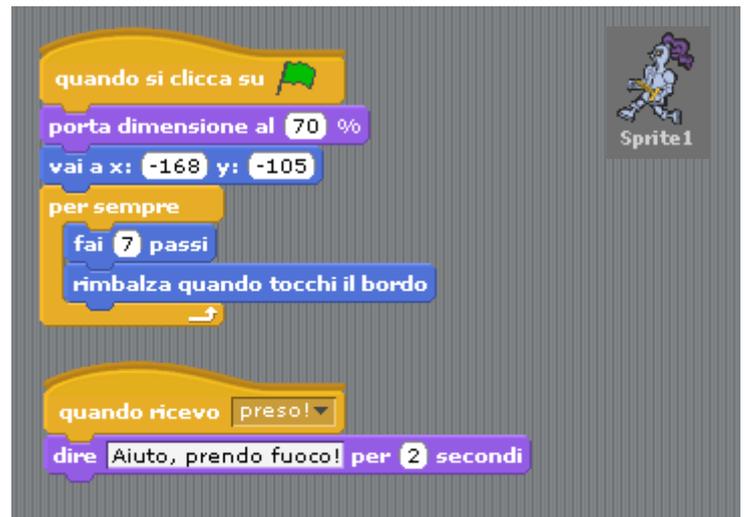




3) per lo sprite di Arduino carica i due costumi del drago e seleziona “Solo destra-sinistra” come stile di rotazione poi aggiungi lo script riportato qui sotto (cambia il numero 686 mettendo un valore che vada bene per il tuo sensore):



4) aggiungi poi un nuovo sprite, importando il costume del cavaliere (“knight1”) e siccome anche lui correrà avanti ed indietro per lo schermo, scegliamo lo stile di rotazione “Solo destra-sinistra”; aggiungi poi gli script trascritti qui a destra:



Trova questo ed altri tutorial al nostro indirizzo: coderdojotrento.it/arduino

Ringraziamenti

Questo tutorial:

- * è in larga parte ispirato al materiale del workshop su Arduino tenuto da Giulio Pilotto (@giulio_pilotto) & Mirco Piccin (@mircopiccin),
- * fa parte delle “risorse” del CoderDojo Trento,
- * è stato scritto con il supporto di CoderDolomiti.

Crediti per le immagini:

TinkerKit e disegno componenti: [Canada Robotix](#) e [Arduino Home](#) / Immagine dei pin di Arduino: [SparkFun Electronics](#) (CC BY-SA 3.0)

