

## INGRESSI DIGITALI USCITE DIGITALI

Dopo aver testato le uscite digitali, il passo successivo è il test degli ingressi digitali.

In questo caso è necessario collegare un pulsante ad un ingresso digitale e per vedere l'effetto dell'azione del pulsante conviene riportare lo stato del pulsante sull'uscita digitale già collegata nell'esperimento precedente.

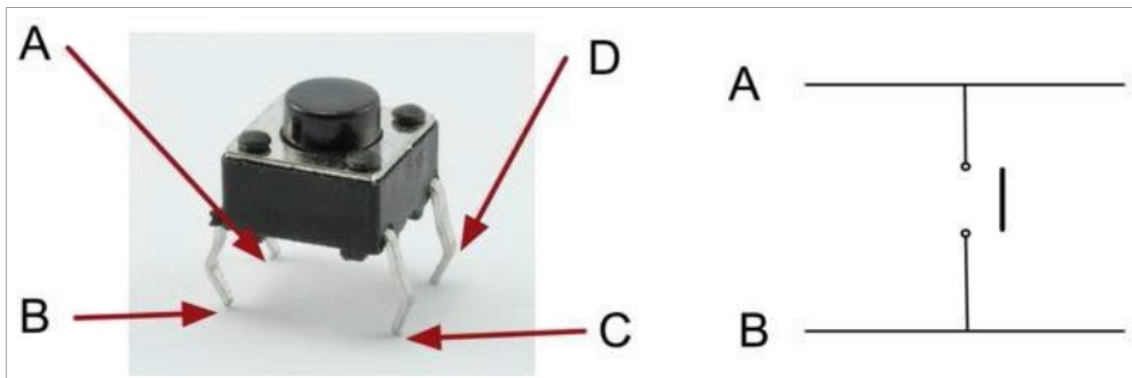
E' inoltre possibile collegare lo stato del pulsante ad azioni dello sprite.

Questo esercizio richiede montaggio dell'esercizio precedente (Arduino, breadboard, LED, resistenza, ponticelli) con l'aggiunta del pulsante e di due ulteriori ponticelli.

## MONTAGGIO DEL PROGETTO

Supponiamo di utilizzare ancora il pin digitale 12 come uscita con il LED già collegato e di utilizzare il pin digitale 2 come ingresso collegando il pulsante.

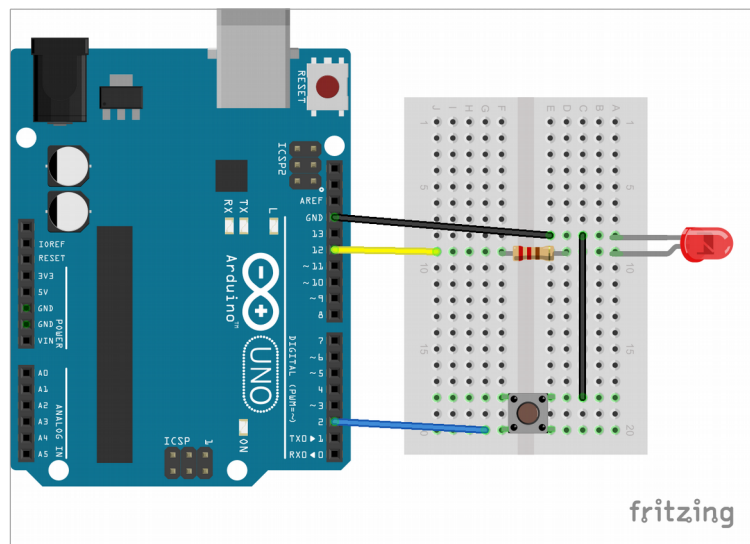
Il pulsante è un dispositivo dotato di quattro piedini.



I piedini sono accostati a coppie due lati opposti del dispositivo. I piedini corrispondenti su due lati opposti sono elettricamente collegati tra loro (A con D e B con C) e quindi è come se fossero lo stesso piedino. I piedini adiacenti sullo stesso lato (A e B o C e D) invece sono normalmente non collegati e diventano collegati solo quando si preme il pulsante.

Per realizzare un ingresso digitale si deve collegare uno dei piedini adiacenti al pin di ingresso digitale e l'altro piedino adiacente (o il suo opposto) alla massa della scheda (GND).

Schema di montaggio:



In questo esempio il piedino B del pulsante è collegato al pin digitale 2 di Arduino mentre il piedino D è collegato allo stesso nodo del pin negativo del LED che era già stato collegato in precedenza a massa (GND).

Si sarebbe anche potuto usare il pin B del pulsante per il collegamento a massa; la scelta del piedino D è stata fatta per ottenere uno schema più ordinato.

**ATTENZIONE!** Con questo tipo di collegamento del pulsante in condizioni di riposo (pulsante non premuto) l'ingresso digitale 2 di Arduino vale "Vero" mentre quando il pulsante viene premuto l'ingresso vale "Falso".

Il comportamento del segnale è quindi l'opposto di quello che ci si aspetterebbe secondo una logica comune. Questa organizzazione, che si chiama "Logica negativa" (perché ribalta le condizioni logiche: vero se non faccio azione, falso se la faccio) è di uso piuttosto comune nella tecnologia perché semplifica la struttura hardware (Per realizzare una "Logica positiva" ci sarebbe voluta una ulteriore resistenza e altri due ponticelli).

Sarà sufficiente tenere conto di questa inversione nella scrittura del codice.

## PROGRAMMI SNAP4ARDUINO

Il primo programma che si può realizzare è un programma che riconosce la pressione del pulsante ed accende il LED quando il pulsante è premuto e lo spegne quando il pulsante non è premuto.



La lettura dell'ingresso è inserita in un blocco "Operatore" di confronto e lo stato letto viene confrontato con il valore "vero". Il risultato dell'operazione può essere vero o falso; se il risultato è vero viene eseguito il primo ramo del blocco "se" mentre se il risultato è falso viene eseguito il secondo blocco ("altrimenti").

Notare l'inversione della logica: con risultato vero (pulsante non premuto) si spegne il LED, con risultato falso (pulsante premuto) si accende il LED.

Il codice di questo esempio ( 03\_ingresso\_digitale\_uscita\_digitale.xml ) si può scaricare dal sito dedicato all'URL:

[https://www.schoolmakerday.it/wp-content/uploads/2018/03/tutorial\\_primaria.zip](https://www.schoolmakerday.it/wp-content/uploads/2018/03/tutorial_primaria.zip)

Con una piccola modifica si può anche associare la pressione del tasto ad azione sullo sprite.

Nell'esempio seguente lo sprite si muove alla pressione del tasto.

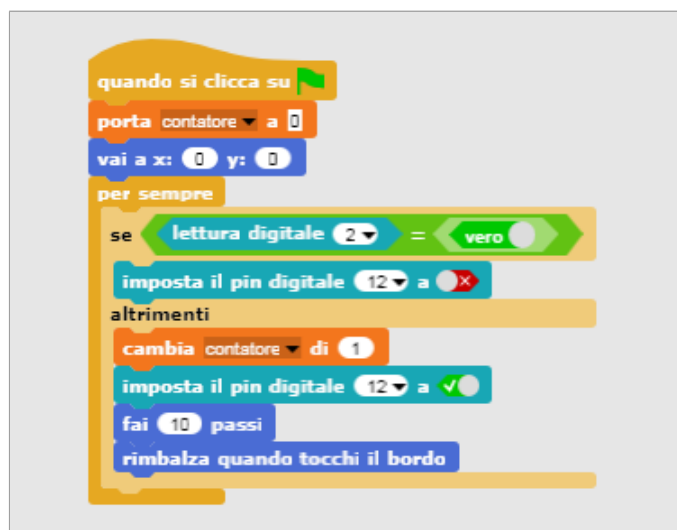


Il codice di questo esempio(04\_ingresso\_digitale\_movimento\_sprite.xml) si può scaricare dal sito dedicato all'URL:

[https://www.schoolmakerday.it/wp-content/uploads/2018/03/tutorial\\_primaria.zip](https://www.schoolmakerday.it/wp-content/uploads/2018/03/tutorial_primaria.zip)

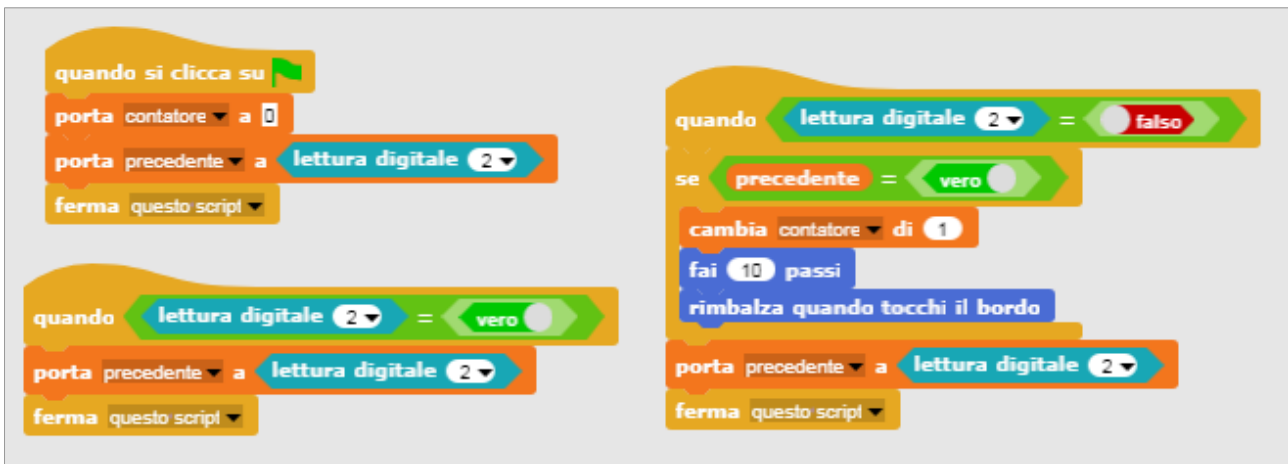
Un ulteriore programma che si può sperimentare è il riconoscimento di un evento. Riconoscere un evento vuol dire riconoscere il momento del passaggio dello stato di un ingresso da uno stato ad un altro.

Ad esempio potrebbe essere utile per realizzare un contatore. Si può infatti verificare che se si realizzasse un contatore a partire dall'esempio precedente aggiungendo una variabile ed incrementandola ogni volta che il tasto è premuto il contatore non funzionerebbe bene.



Si aggiunge una variabile "contatore" e lo si incrementa quando il pulsante è premuto. Si nota però che ogni volta che l'utente preme il pulsante il contatore si incrementa di un numero variabile di unità. Il motivo è che il blocco "per sempre" viene ripetuto molte volte durante la pressione del pulsante falsando il risultato del conteggio.

Si può risolvere il problema aggiungendo una seconda variabile che ricordi lo stato precedente del pulsante; confrontando lo stato attuale del pulsante con lo stato precedente si può riconoscere la "transizione negativa" (passaggio da non premuto a premuto) e contare solo in occasione di questo evento



L'azione di conteggio invece che in un "ripeti per sempre" avviene in un blocco "Quando".

Il blocco "Quando si clicca su bandierina verde" fa solo l'inizializzazione delle variabili: "contatore" a 0, "precedente" allo stato del pulsante.

Quando il pulsante risulta premuto (falso) se nella lettura precedente era vero si riconosce la transizione negativa e si fanno le azioni di conteggio e movimento mentre se anche prima era falso significa che è la continuazione di una pressione già contata. Affinchè questo meccanismo funzioni è necessario ogni volta che si legge un ingresso, indipendentemente dall'azione fatta, effettuare un aggiornamento dello stato precedente.