

Quanto Tempo Hai Premuto il Pulsante?

Obiettivo: Determinare per quanto tempo un pulsante è stato premuto.

Componenti elettronici:

- Arduino UNO
- Breadboard
- Pulsante
- Resistenza (1k0hm)

Prerequisiti

[*Blinking Led Senza Delay: MILLIS\(\)*](#)

[*Pulsante come Interruttore*](#)

Teoria: Poter misurare il tempo di pressione di un tasto può risultare utile in molte applicazioni. Infatti, questa informazione permette di discriminare le differenti modalità di iterazione con il pulsante come il click (tasto premuto) ed

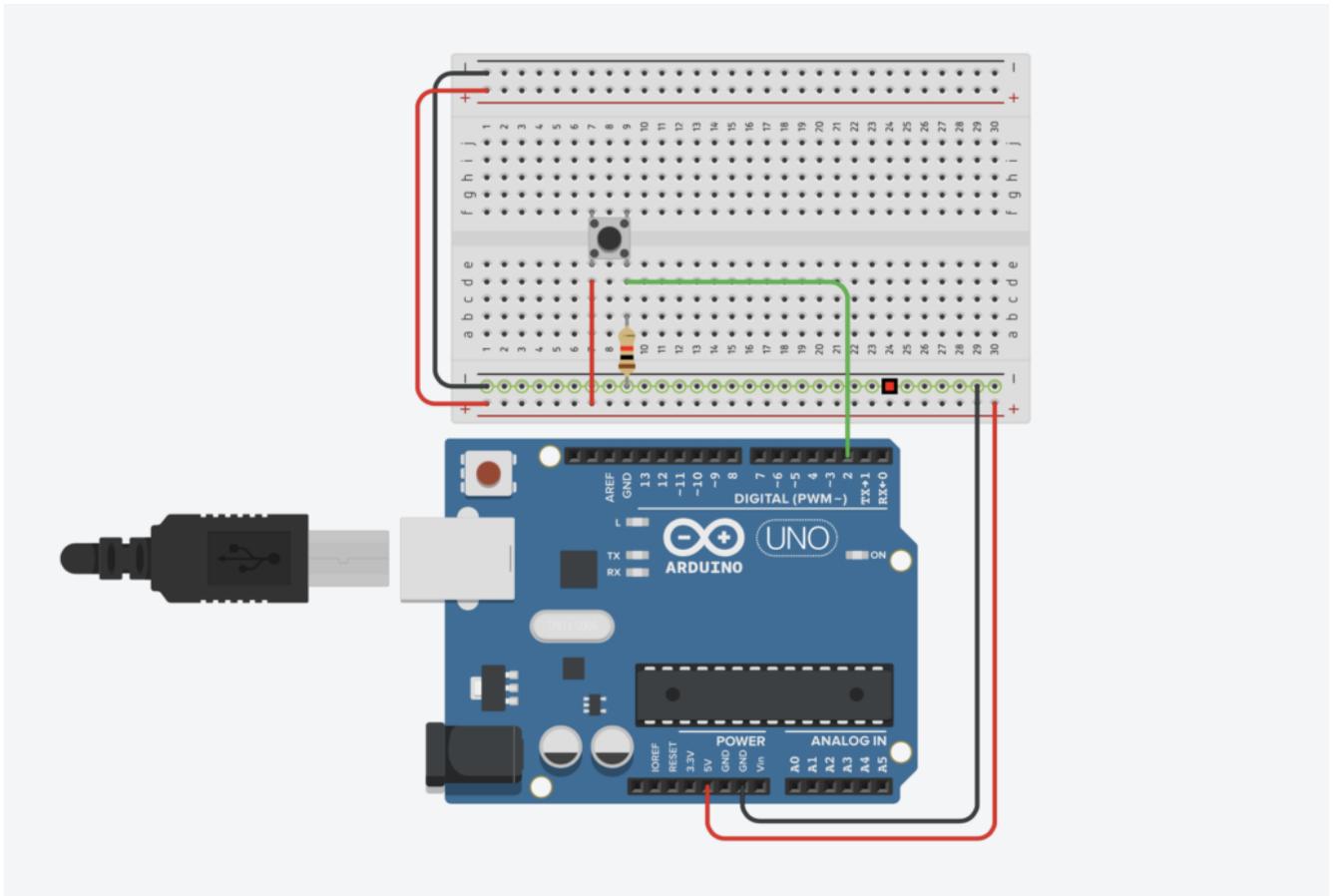
il long click (tasto premuto a lungo). Potere discriminare questi comportamenti permette di abilitare il pulsante a differenti funzioni. Ad esempio il single click potrebbe essere utilizzato per accendere un led mentre il long click potrebbe essere utile per farlo lampeggiare.

Dal punto di vista hardware il circuito necessario per realizzare questa applicazione è molto semplice ed è costituito dal singolo pulsante collegato a vcc e al ground mediante resistenza di pull-down.

Elemento centrale di questa esercitazione è la scrittura di un codice corretto che permetta di misurare esattamente lo scorrere quel tempo. Questo codice si basa sull'impiego di due elementi fondamentali:

- La funzione millis: questa funzione restituisce il numero di millisecondi che sono passati da quando la board Arduino ha eseguito il programma corrente. Questo numero si riavvolge dopo circa 50 giorni. L'impiego di questa funzione è fondamentale per misurare il tempo di pressione del pulsante. Nel dettaglio, questa operazione può essere svolta semplicemente eseguendo la differenza tra le misure temporali prese quando il pulsante è premuto e quando il pulsante è rilasciato.
- La variabile di stato: questa variabile (di natura globale) permette di determinare quando un pulsante è premuto e quando il pulsante è rilasciato. Nello specifico grazie a questa variabile è possibile determinare un passaggio dallo stato logico basso a quello alto e viceversa.

Collegamento Circuitale:



Collegamento Circuitale

Codice: A seguire viene riportato il codice utilizzato per determinare il tempo di pressione di un pulsante. Nello specifico il codice utilizza la variabile di stato `valButtonOld` per memorizzare lo stato del pulsante relativo al ciclo passato.

Quando i valori di `valButton` e `valButtonOld` differiscono, allora c'è stato un passaggio di stato.

<code>valButtonOld</code>	<code>valButton</code>	Evento
LOW	HIGH	Il pulsante è stato premuto
HIGH	LOW	Il pulsante è stato rilasciato

Nel caso specifico del passaggio di stato viene effettuata una misura del tempo trascorso mediante la funzione `millis()`. Per determinare il tempo trascorso basta semplicemente effettuare

una differenza tra le due misure realizzate.

Personalizzazioni: E' possibile modificare l'hardware introducendo due led. Quando il pulsante viene premuto per meno di un secondo deve accendersi il primo led, quando invece il pulsante viene premuto per più di un secondo deve accendersi il secondo.

Blinking Led Senza Delay: MILLIS()

Obiettivo: Realizzazione del classico blinking led senza utilizzare la funzione Delay

Componenti elettronici:

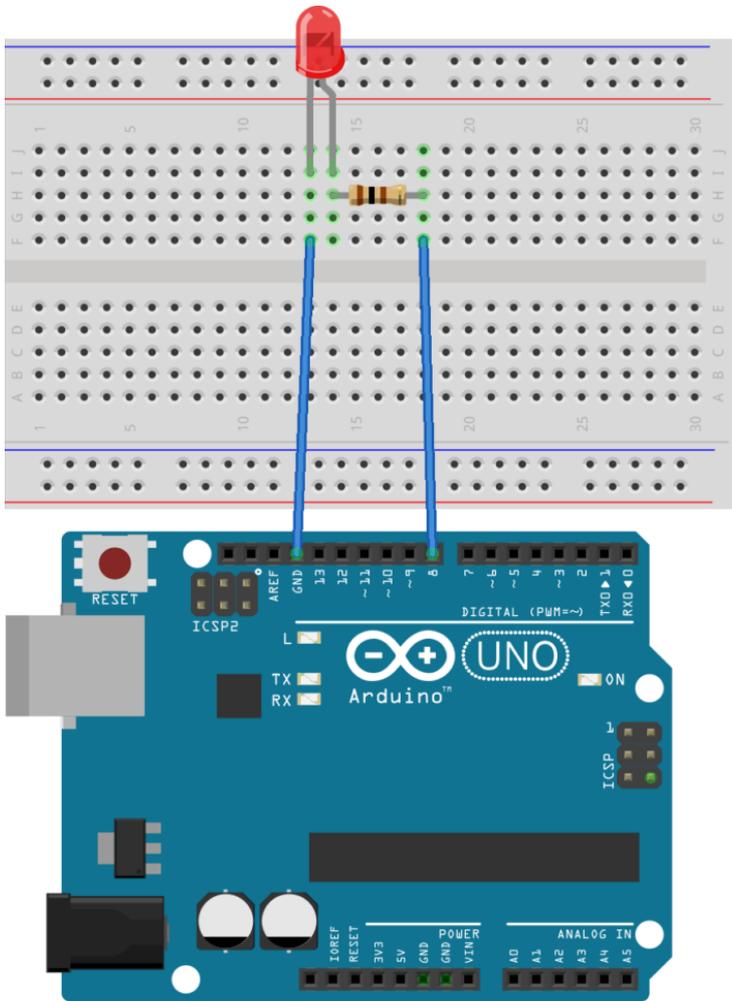
- Arduino UNO
- Breadboard
- Led
- Resistenza (100 Ohm)

Teoria: Se da un certo punto di vista l'impiego della funzione delay è particolarmente utile per la realizzazione di semplici applicativi; da un altro punto di vista molteplici potrebbero essere gli inconvenienti legati all'utilizzo di questa funzione.

E' infatti, molto importante, considerare che l'istruzione delay è un'istruzione bloccante. Questa istruzione "congela" Arduino nel suo stato corrente, pertanto negli istanti di delay non è possibile fare aggiornamenti, gestire input attraverso letture o comandare attuatori mediante scritture. Ad esempio, nel caso della realizzazione di un semaforo per perdoni, dove è possibile effettuare la chiamata mediante la pressione di un pulsante, l'impiego della funzione delay è altamente sconsigliato per la gestione del semaforo perchè questo renderebbe invisibile la pressione del pulsante. Per tutte queste ragioni è opportuno scrivere codici senza l'impiego della funzione delay. Nel dettaglio è opportuno sostituire la funzione delay con la più funzionale ma meno pratica funzione [millis](#).

La funzione millis restituisce il numero di millisecondi che sono passati da quando la board Arduino ha eseguito il programam corrente. Questo numero si riazzera dopo circa 50 giorni.

Collegamento Circuitale:



fritzing

Collegamento Circuitale

Codice:

A seguire viene riportato il codice utilizzato per fare lampeggiare il led senza delay. Nello specifico il codice memorizza l'ultimo istante in cui è avvenuta una azione (il led ha cambiato stato) nella variabile `previousMillis` mentre nella variabile `currentMillis` viene memorizzato il tempo corrente. Attraverso la differenza tra questi due tempi si comprendere se il led deve cambiare stato (accendersi o spegnersi). Nel dettaglio, se la differenza tra la variabile `currentMillis` e `previousMillis` è maggiore di 1000 millisecondi allora lo stato del led cambia.

Personalizzazioni: E' possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *interval*. Modificando il suo valore infatti cambia la frequenza di lampeggiamento del LED.

E' inoltre possibile modificare il pin digitale utilizzato per pilotare il LED cambiando rispettivamente hardware e software.

L'utilizzo di una resistenza, in serie al LED, serve appunto per limitare la quantità di corrente presente sul diodo emettitore di luce.