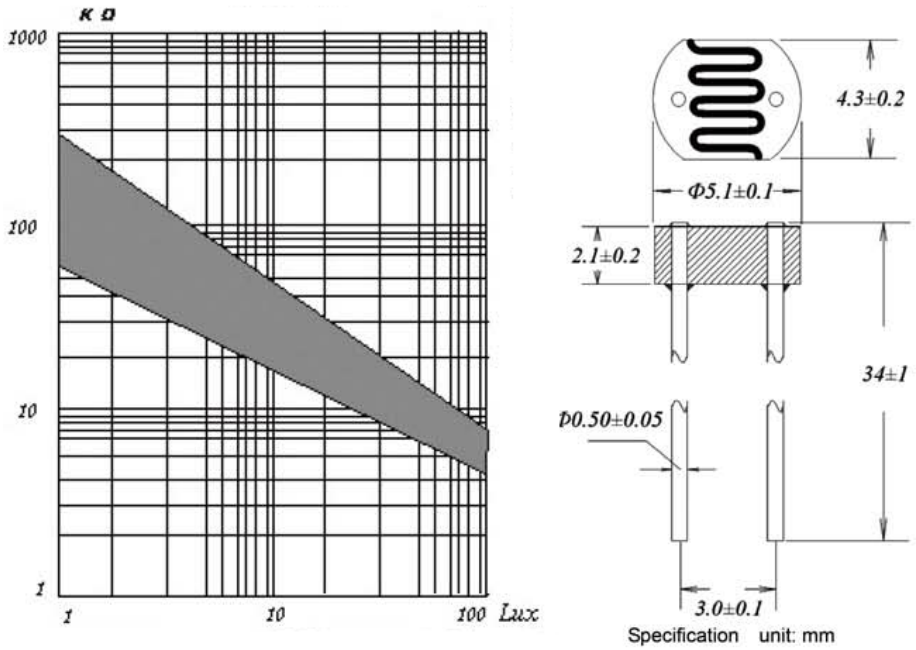
Crepuscolare (Smart Lamp)

Obiettivo: Accensione automatica di un LED al diminuire dell’intensità di luce rilevata.

Componenti elettronici:

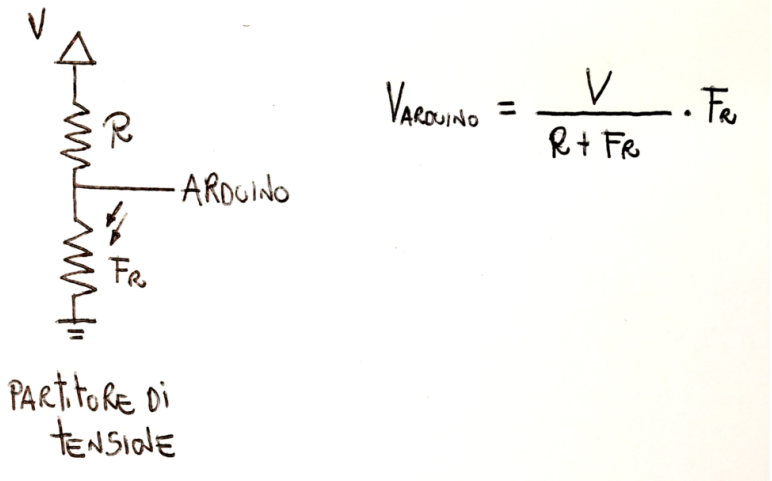
- Arduino UNO
- Breadboard
- 1 Led
- 1 Foto-resistenza
- 1 Resistenza (2.2kOhm)

Teoria: La foto resistenza è un componente elettronico la cui resistenza è inversamente proporzionale alla quantità di intensità luminosa che lo colpisce. Questo significa che la corrente (inversamente proporzionale alla resistenza) aumenta all’aumentare dell’intensità luminosa.
A seguire, viene riportato la caratteristica Ohm/Lux di una fotoresistenza tipicamente impiegata in applicazioni realizzate mediante Arduino.



Datasheet Fotoresistenza

Nel caso specifico, è importante ricordare che Arduino non può rilevare né variazioni di resistenza né variazioni di corrente. Il microcontrollore può infatti analizzare solamente valori di tensione. Per questo motivo, l'utilizzo di una resistenza è indispensabile al fine di collegare correttamente la fotoresistenza ad Arduino. Nel dettaglio, il circuito previsto per trasformare la variazione di resistenza in una variazione di tensione è il partitore di tensione.



Partitore di Tensione e Fotoresistenza

Nel circuito presentato, la variazione di luminosità produce una variazione del valore della fotoresistenza. Di conseguenza anche il valore della corrente risulta funzione dell'intensità luminosa e di conseguenza anche il valore di tensione in ingresso ad Arduino. Nel dettaglio:

- Un **incremento** della luminosità porta ad un decremento della tensione.
- Un **decremento** della luminosità porta ad un incremento della tensione.

Attraverso l'utilizzo del comando **analogRead** è possibile leggere la tensione su uno specifico pin analogico (A0-A5) di Arduino. La funzione analogRead restituisce un valore compreso tra 0 e 1023 a seconda della tensione letta dal microcontrollore.
A titolo di esempio, se il valore di tensione letto utilizzando la funzione analogRead sul pin A0 di Arduino risulta pari a 613. Il valore di tensione può essere facilmente calcolato:

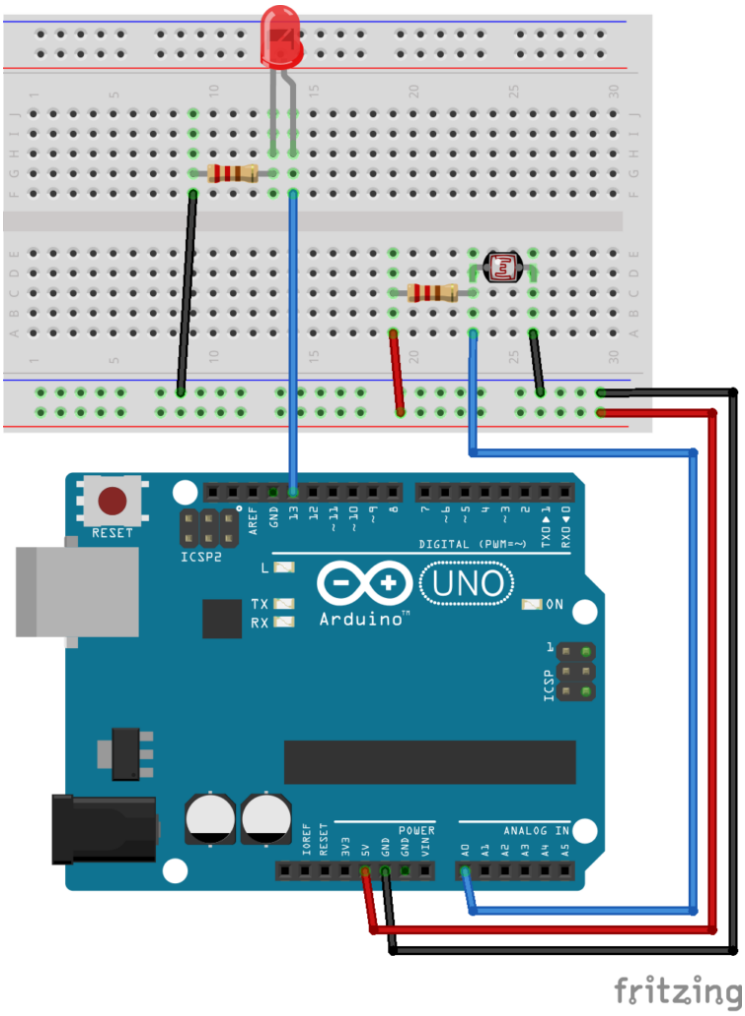
valoreTensioneAnalogico = 613/1023*5 = 3V

dove:

- Il valore analogico di tensione letto utilizzando l'istruzione analogRead è pari a 613
- Il valore di tensione massimo che può essere letto dalla funzione analogRead è pari a 1023
- La tensione massima in uscita ad Arduino è pari a 5V

Tale valore può essere facilmente utilizzato per controllare un led mediante l’**istruzione condizionale** IF.

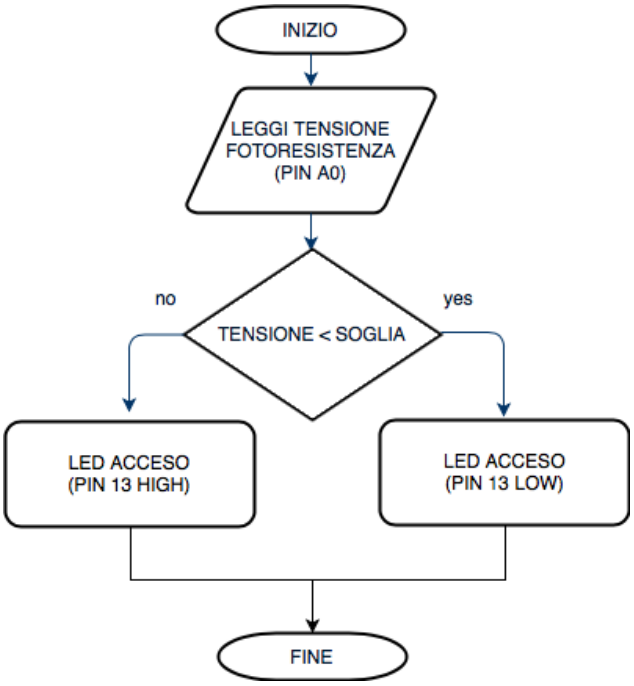
Collegamento Circuitale:



Collegamento Circuitale

Codice:

A seguire viene riportata la schematizzazione mediante flowchart dell’algoritmo utilizzato per realizzare il programma.



Flowchart

[crayon-66275c2570d72833059749/]

Personalizzazioni:

E’ possibile modificare il circuito aggiungendo altri led. E’ inoltre possibile modificare il codice al fine di realizzare una lampada che accenda un numero differente di led in funzione della luminosità.

Controllo di un LED mediante

un Potenziometro

Obiettivo: Controllare un LED utilizzando un potenziometro

Pre-Requisiti

Fading led

Componenti elettronici:

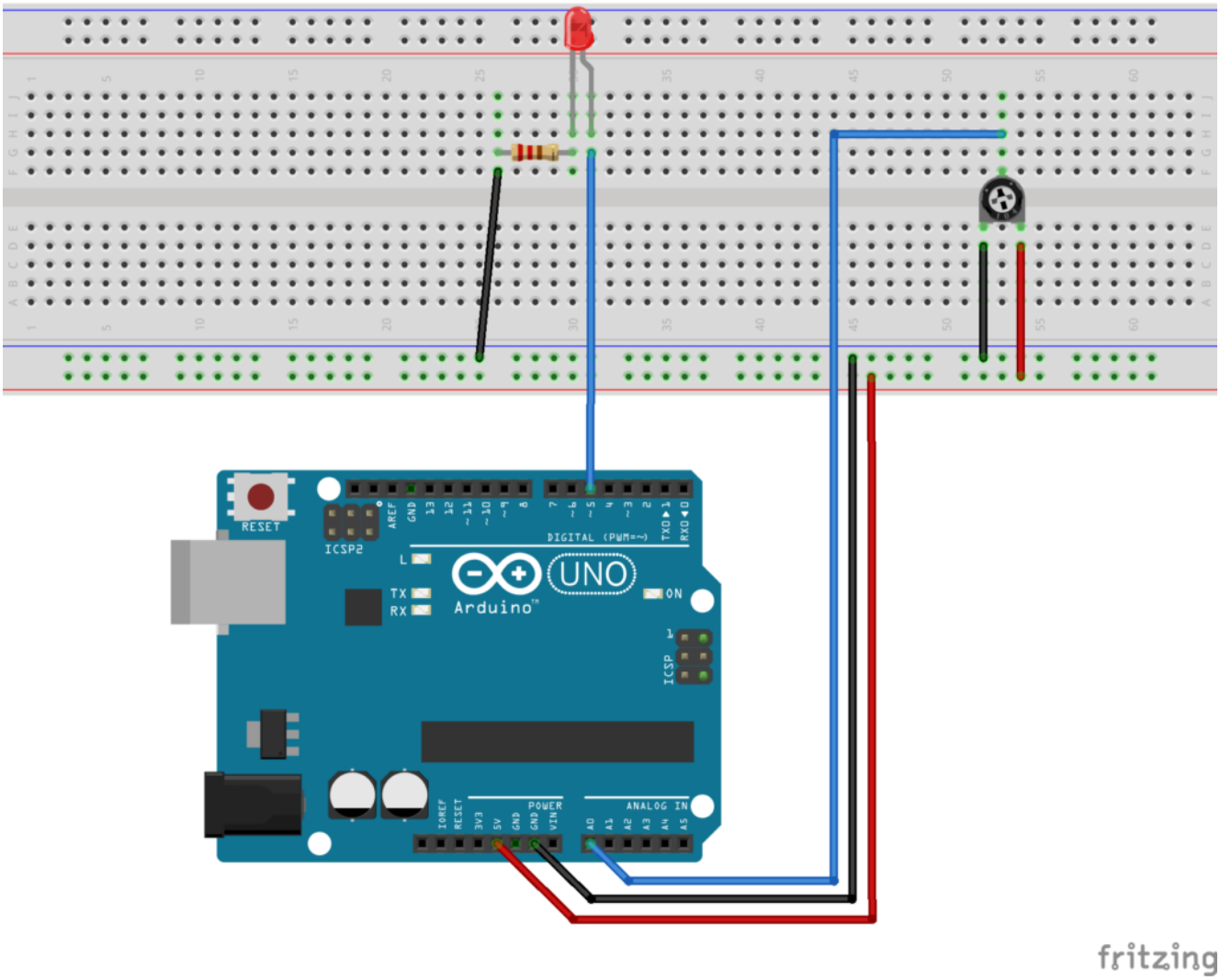
- Arduino UNO
- Breadboard
- 1 Led
- 1 Resistenza (100 Ohm)
- 1 Trimmer (4.7KOhm o similari)

Teoria: In questo articolo si propone l'utilizzo di un potenziometro per regolare in modo manuale la luminosità di un LED. E' importante considerare che (come riportato nei pre-requisiti) l'impiego della funzione **digitalWrite** non permette di modulare l'intensità luminosa del LED. Difatti, attraverso l'utilizzo di questa istruzione digitale, il LED può trovarsi solamente in due stati logici LOW (spento) o HIGH (acceso).
Per raggiungere lo scopo prefissato è pertanto necessario l'utilizzo di una differente funzione denominata: **analogWrite**.

Questa funzione permette infatti di modulare l’intensità luminosa del LED fornendo 256 differenti livelli di luminosità. L’istruzione analogWrite permette infatti di emulare un finto segnale analogico attraverso l’impiego della tecnica **PWM** (Pulse Width Modulation). Solamente sei PIN (quelli contrassegnati dal simbolo tilde ~) possono essere utilizzati per fornire un segnale “analogico”.

Tuttavia è importante considerare che se l’istruzione analogwrite permette di gestire la luminosità del LED, questa funzione non permette di controllare la posizione del potenziometro essendo il potenziometro un dispositivo di input (dato da leggere). Pertanto per la gestione del potenziometro sarà effettuata utilizzando una differente funzione denominata **analogRead**. Questa funzione permette infatti di leggere un livello di tensione compreso tra 0 e 5 Volt e mapparlo in un intervallo discreto composto da 1024 livelli (0-1023).

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Pulsante come Interruttore

Obiettivo: Utilizzo di un pulsante come interruttore per l’accensione di un led (stato ON/OFF)

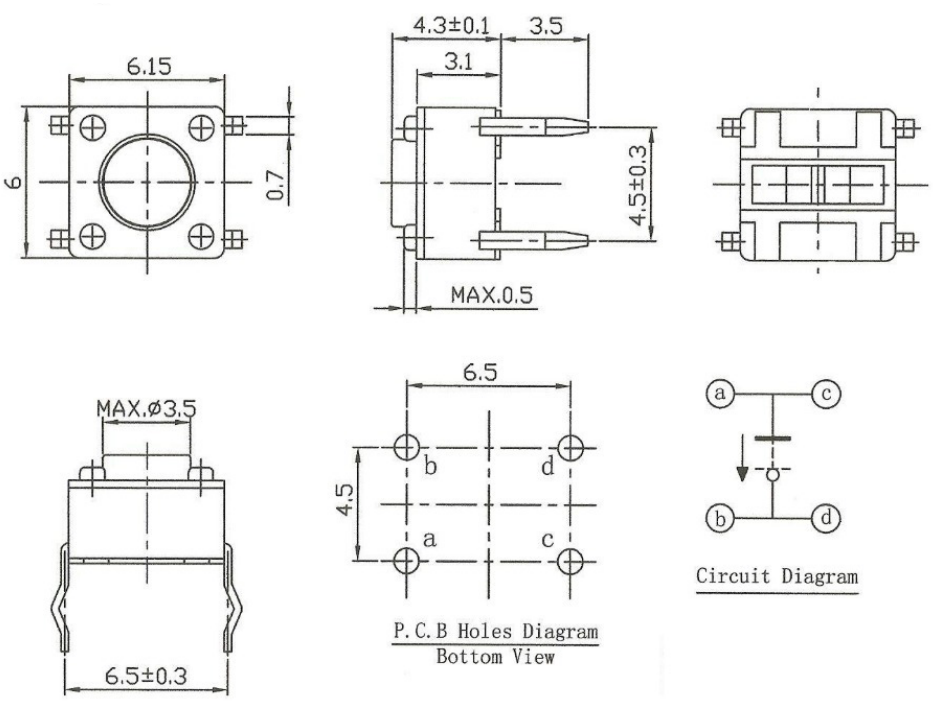
Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Led
- 1 Pulsante
- 1 Resistenza (100 Ohm)
- 1 Resistenza (1kOhm)

Pre-Requisiti

LED e Pulsante

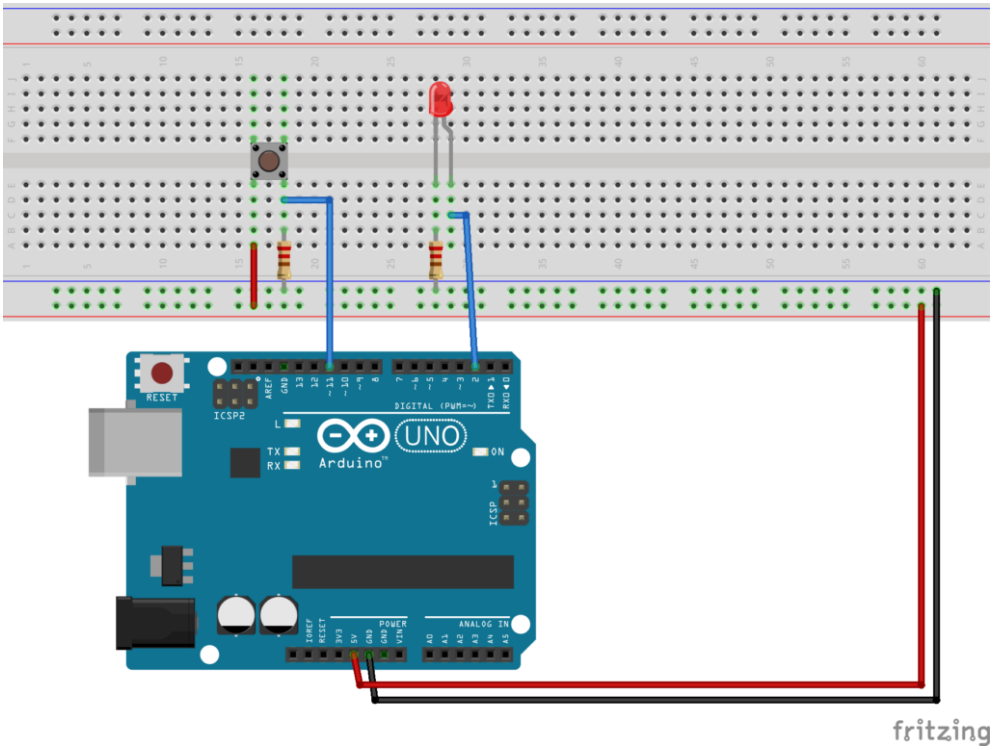
Teoria: Da un punto di vista pratico mantenere una luce accesa tenendo premuto il pulsante (vedi lezione pre-requisiti) non è proprio il massimo della vita. Se da un punto di vista elettronico il componente tipicamente utilizzato per gestire lo stato ON/OFF di una lampada è l'interruttore, attraverso una corretta gestione del software di controllo è possibile "trasformare un pulsante in interruttore". Per questo motivo, agendo direttamente sul software e lasciando invariato l'hardware proposto nell'esperienza LED e Pulsante, è possibile trasformare il sistema in questione in un controllo ON/OFF dove il pulsante è utilizzato come interruttore.



Datasheet Pulsante

Anche se in questa esperienza il pulsante viene utilizzato come interruttore è importante ribadire che l’impiego di resistenze di Pull-up o Pull-down sono necessarie per un corretto funzionamento del circuito.Maggiori informazioni sulle resistenze di Pull-up o Pull-down possono essere trovate qui.

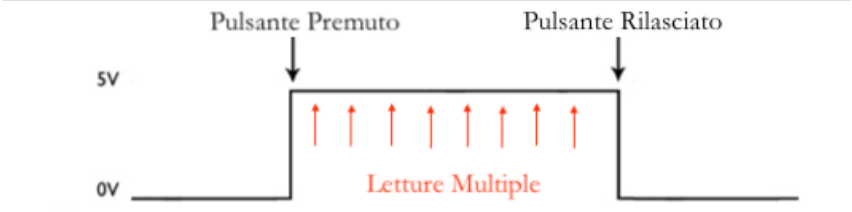
Collegamento Circuitale:



Collegamento Circuitale

Codice: Il codice utilizzato per trasformare il pulsante in interruttore si basa sull'impiego di due differenti variabili globali (create fuori dalle funzioni loop e setup, pertanto visibili in tutto il codice). Queste variabili sono:

- **valButtonOld**: Memorizza la lettura precedente effettuata sul pulsante. Attraverso questa variabile è possibile evitare letture multiple. E' importante considerare che essendo il clock di Arduino più elevato del tempo di reazione umano, premendo il pulsante anche per un istante brevissimo il controllore eseguirà più letture con stato del pulsante HIGH eseguendo in seguito il codice corrispondente. Per questo motivo si preferisce leggere lo stato di transizione da basso ad alto piuttosto che lo stato alto di un pulsante.



Esempio di lettura multipla

- **ledState**: Memorizza lo stato del led (se acceso o spento). Premuto il pulsante, nel caso in cui il led sia acceso questo viene spento viceversa nel caso in cui il led sia spento questo viene acceso. Per questo motivo è indispensabile utilizzare una variabile per controllare lo stato del pulsante.

Personalizzazioni:

L'impiego della variabile valButtonOld, utilizzata per evitare letture multiple, può essere evitato aggiungendo un delay dopo la lettura del pulsante. Si provi ad implementare il circuito ed il relativo codice.

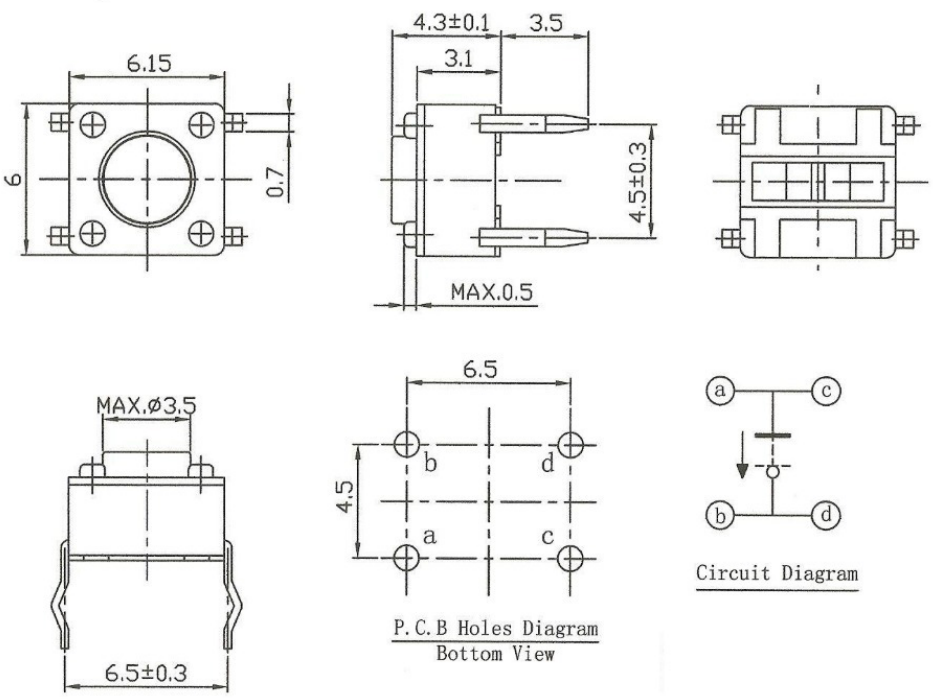
LED e Pulsante

Obiettivo: Accensione di un LED mediante un pulsante.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Led
- 1 Pulsante
- 1 Resistenza (100 Ohm)
- 1 Resistenza (1kOhm)

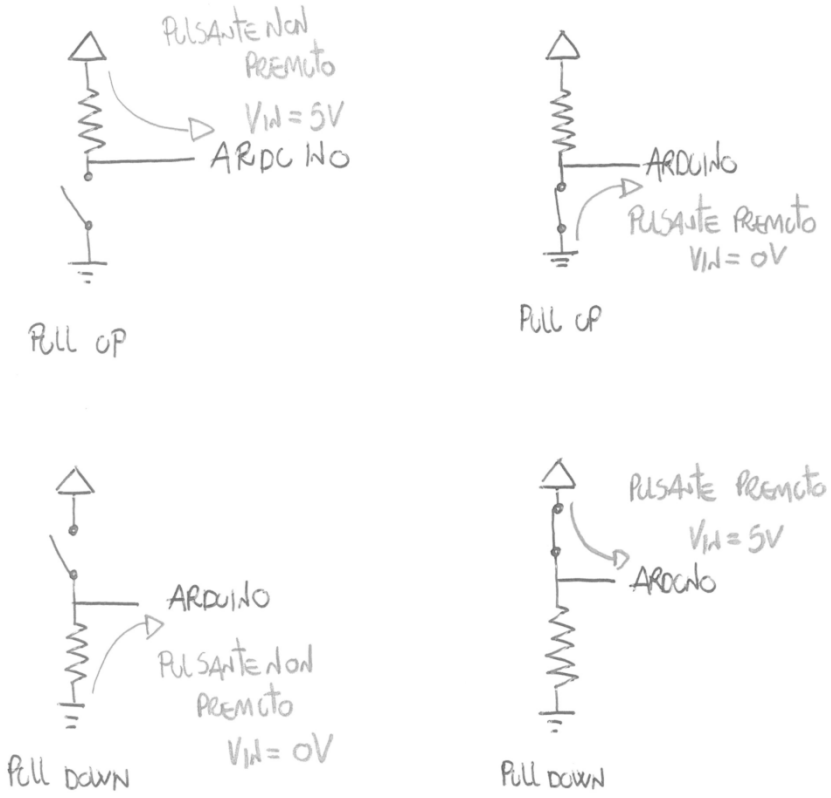
Teoria: Il pulsante è un dispositivo elettronico a due stati (ON, OFF) con una sola posizione monostabile. Nel caso specifico i pulsanti permettono di aprire o chiudere un circuito e pertanto collegare a GND (0V) o a VCC (5V) una specifica uscita. A seguire, viene riportato lo schema circuitale di un pulsante tipicamente impiegato in applicazioni realizzate mediante Arduino.



Datasheet Pulsante

L’utilizzo di una resistenza è indispensabile al fine di collegare correttamente il pulsante ad Arduino evitando cortocircuiti. A seconda del collegamento realizzato, la resistenza prende il nome di:

- Resistenza di **Pull Up**: la resistenza viene collegata direttamente all’alimentazione.
- Resistenza di **Pull Down**: la resistenza viene collegata a massa.



Resistenze di Pull Up e di Pull Down

Il comportamento del circuito e la tensione letta dal microcontrollore dipende dalla tipologia di collegamento circuitale utilizzato. In particolare:

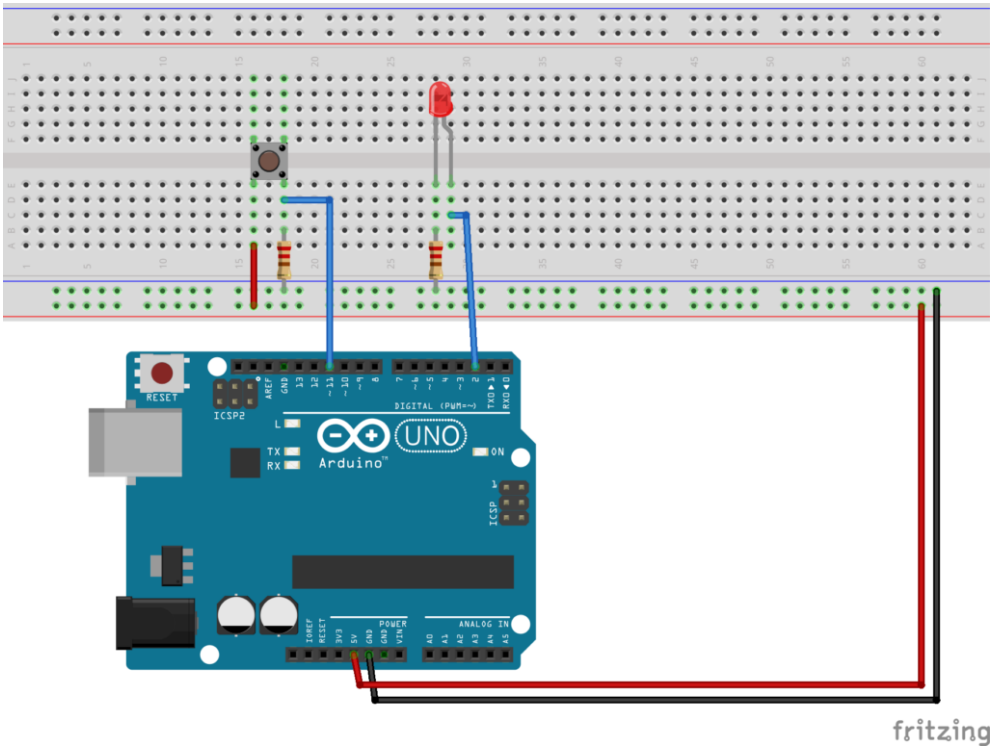
- Resistenza di **Pull Up**:

        - se il pulsante viene premuto la tensione in ingresso ad Arduino è pari a 0.
        - se il pulsante non viene premuto la tensione in ingresso ad Arduino è pari a Vcc (5V)
    - Resistenza di **Pull Down**:
        - se il pulsante viene premuto la tensione in ingresso ad Arduino è pari a Vcc (5V).
        - se il pulsante non viene premuto la tensione in ingresso ad Arduino è pari a 0.

Attraverso l'utilizzo del comando **digitalRead** è possibile leggere la tensione su uno specifico pin digitale (0-13) di Arduino. La funzione digitalRead restituisce un valore (i.e., LOW o HIGH) a seconda della tensione letta dal microcontrollore. Tale valore può essere facilmente utilizzato per controllare un led mediante l'**istruzione condizionale** IF.

A seguire viene riportato lo schema elettrico ed il codice utilizzato per l'accensione di un pulsante sfruttando una resistenza di Pull Down.

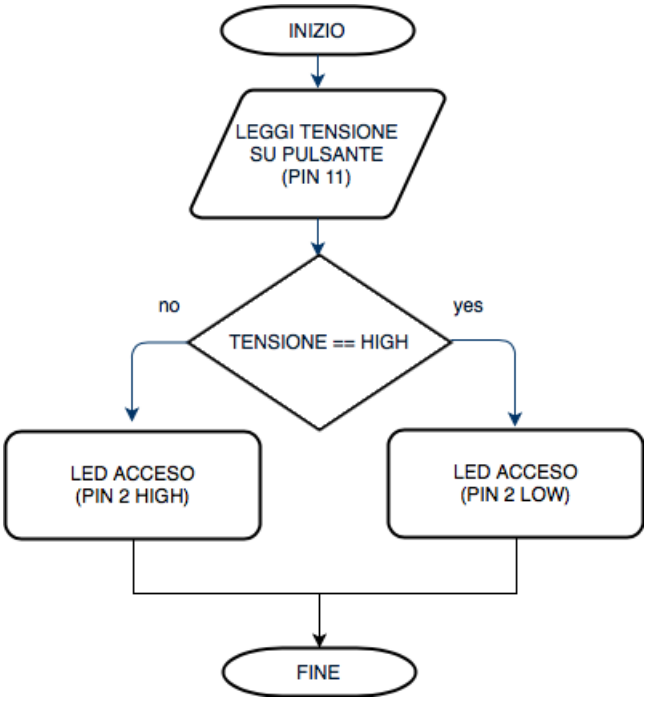
Collegamento Circuitale:



Collegamento Circuitale

Codice:

A seguire viene riportata la schematizzazione mediante flowchart dell’algoritmo utilizzato per realizzare il programma.



Flowchart

Personalizzazioni:

E’ possibile modificare il circuito aggiungendo pulsanti e led. E’ inoltre possibile modificare il codice al fine di realizzare una lampada che rimanga acceso fino a quando il pulsante non venga premuto una seconda volta (Attenzione possibili problematiche di rimbalzo).

Il LED RGB

Obiettivo: Accensione di un LED RGB (Red Green Blue) .

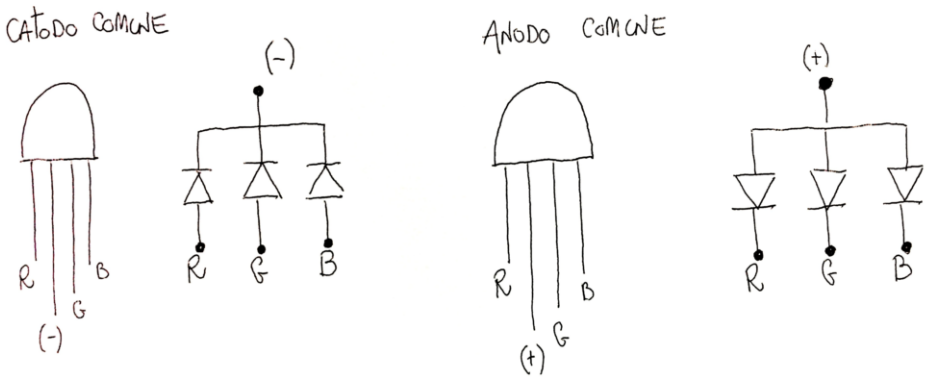
Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Led RGB
- 3 Resistenza (100 Ohm)

Teoria: Gli RGB rappresentano una particolare sottocategoria di LED capaci di riprodurre differenti colori attraverso la combinazione dei colori fondamentali Rosso Verde e Blu. Nel dettaglio, questi dispositivi sono costituiti da tre differenti led azionati in funzione del colore da riprodurre.

I LED RGB sono dotati di 4 differenti morsetti e sono così classificati:

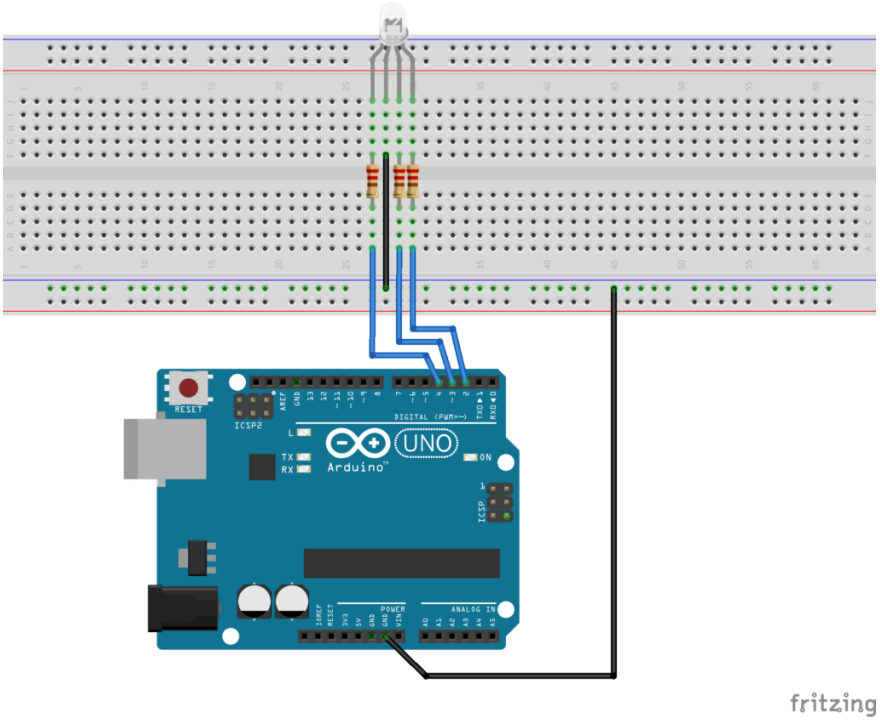
- **RGB ad Anodo Comune:** il pin più lungo del LED deve essere collegato a massa (GND) mentre gli altri 3 differenti PIN sono utilizzati per comandare i 3 LED (rosso, verde, blue). Nel caso in questione i singoli LED sono accesi attraverso l'istruzione Arduino digitalWrite(pinLed, HIGH);
- **RGB a Catodo Comune:** il pin più lungo del LED deve essere collegato a VCC (5V) mentre gli altri 3 differenti PIN sono utilizzati per comandare i 3 LED (rosso, verde, blue). Nel caso in questione i singoli LED sono accesi attraverso l'istruzione Arduino digitalWrite(pinLed, LOW);



LED RGB (Anodo e Catodo comune)

Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per comandare il LED RGB. Nel dettaglio il LED impiegato è della modalità catodo comune.



Collegamento Circuitale

Codice:

Considerando l’impiego di un LED a catodo comune i singoli LED (R, G, B) possono essere accesi mediante l’istruzione digitalWrite(pinLed,HIGH);

Personalizzazioni:

E’ possibile modificare il circuito aggiungendo pulsanti

Buzzer Passivo

Obiettivo: Realizzazione di un semplice segnalatore acustico (basato su un buzzer passivo).

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer passivo

Teoria: Il Buzzer (cicalino) è uno dispositivo elettronico che

permette di emettere un suono (bzzz) se correttamente alimentato.
I buzzer si suddividono in:

- Buzzer Attivi: alimentati con una tensione opportuna riproducono un tono ad una frequenza pre-impostata.
- Buzzer Passivi: non emettono alcun suono pre-impostato, ma necessitano di una forma d’onda specifica per fare vibrare la membrana interna. Possono produrre toni differenti in funzione del segnale di alimentazione utilizzato.

Buzzer Attivo (Funzione DigitalWrite)

Buzzer Passivo (Funzione Tone)

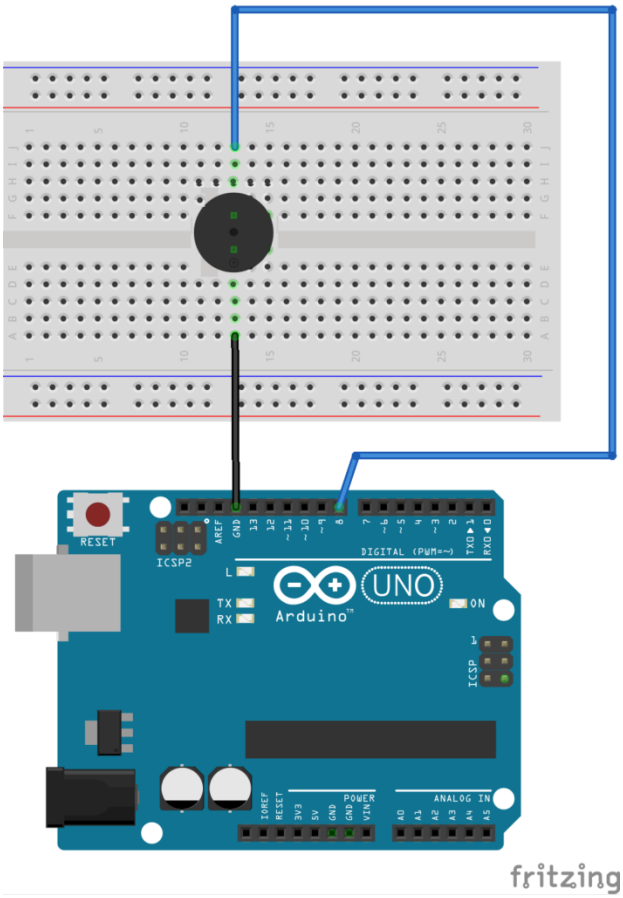
L’utilizzo di un buzzer passivo permette di creare delle melodie personalizzate o differentemente riprodurre piccole canzoni famose. Questo perché il buzzer passivo, a differenza del buzzer attivo, può generare differenti tonalità (note).
Le differenti note sono generate utilizzando la funzione tone (nativa nella libreria Arduino). Nello specifico viene riportata in seguito la sintassi di tale funzione.

tone(pin, frequenza)

tone(pin, frequenza, durata)

Per interrompere un tono è possibile utilizzare la funzione noTone()

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-66275c2574287194523340/]

Personalizzazioni: E’ possibile modificare il comportamento del circuito in questione intervenendo sul valore delle frequenze caratteristiche dei toni e sulle pause introdotte attraverso la funzione delay.

Buzzer Attivo

Obiettivo: Realizzazione di un semplice segnalatore acustico (basato su un buzzer attivo).

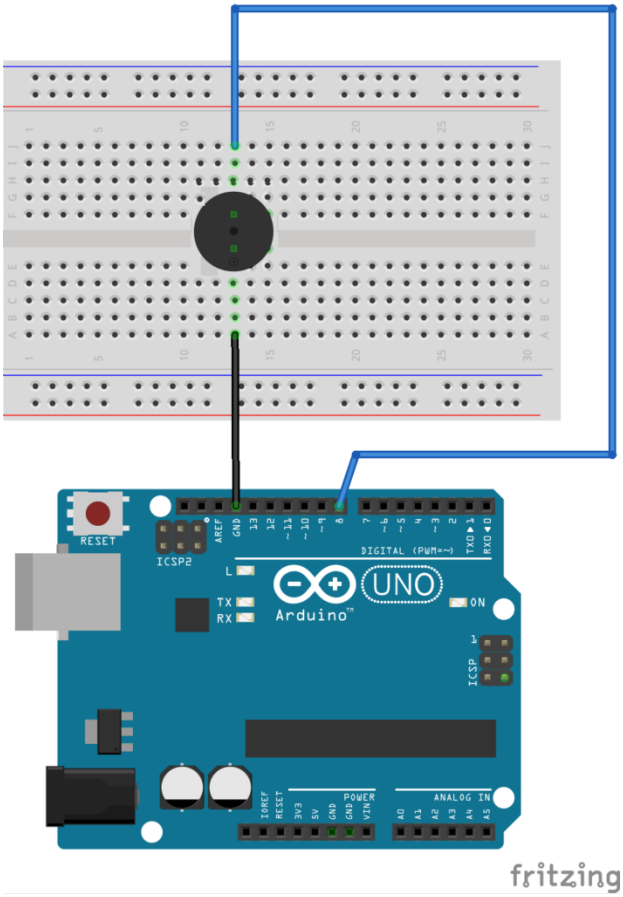
Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer Attivo

Teoria: Il Buzzer (cicalino) è uno dispositivo elettronico che permette di emettere un suono (bzzz) se correttamente alimentato.
I buzzer si suddividono in:

- Buzzer Attivi: alimentati con una tensione opportuna riproducono un tono ad una frequenza pre-impostata.
- Buzzer Passivi: non emettono alcun suono pre-impostato, ma necessitano di una forma d'onda specifica per fare vibrare la membrana interna. Possono produrre toni differenti in funzione del segnale di alimentazione utilizzato.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-66275c2574818055111726/]

Personalizzazioni: E’ possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *buzzerTime*. Modificando il suo valore infatti cambia la pausa generata tra un tono del buzzer ed il successivo.

Fading led

Obiettivo: Realizzazione di un LED con dissolvenza (fading).

Componenti elettronici:

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

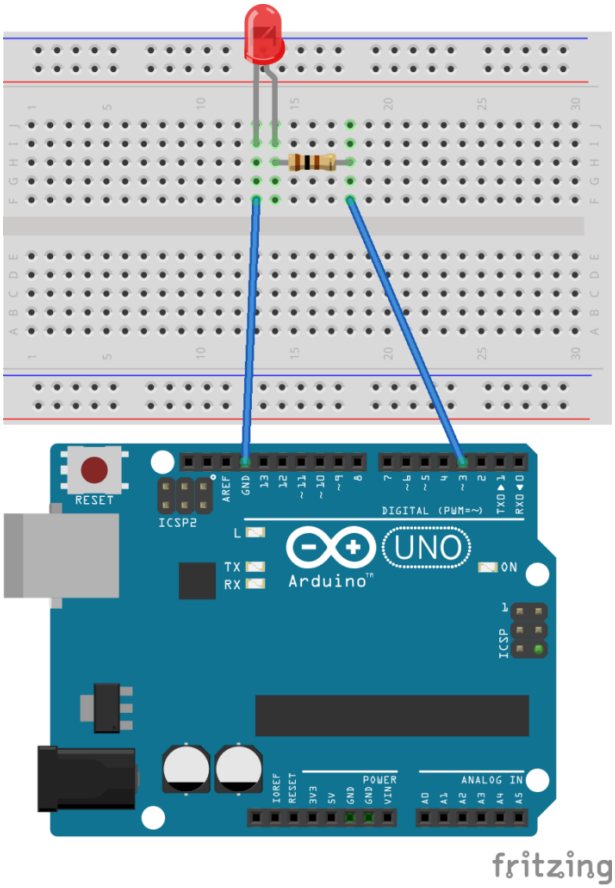
Teoria: L'utilizzo dell'istruzione **digitalWrite** non permette di modulare/regolare la luminosità di un LED. Attraverso l'utilizzo di questa istruzione digitale infatti il LED può trovarsi solamente in due stati logici LOW (spento) o HIGH (acceso).
Per raggiungere lo scopo prefissato è pertanto necessario l'utilizzo di una differente funzione denominata: **analogWrite**. Questa funzione permette infatti di modulare l'intensità luminosa del LED fornendo 256 differenti livelli di luminosità.
L'istruzione analogWrite permette infatti di emulare un finto segnale analogico attraverso l'impiego della tecnica **PWM** (Pulse Width Modulation). Solamente sei PIN (quelli contrassegnati dal simbolo tilde ~) possono essere utilizzati per fornire un segnale "analogico".
A titolo di esempio, volendo alimentare un dispositivo con una tensione analogica pari a 3V il valore da utilizzare come parametro della funzione analogWrite può essere così calcolato:

valore = 3/5*255 = 153

dove:

- Il valore analogico che si vuole produrre è pari a 3V
- La tensione massima in uscita ad Arduino è pari a 5V
- Il valore massimo utilizzabile dalla funzione analogWrite è pari a 255

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-66275c2574d5d593113296/]

Barra LED (Knight Rider) [Avanzato]

Obiettivo: Realizzazione della barra LED utilizzata nella serie TV Knight Rider.

Prerequisiti:

Barra LED (Knight Rider)

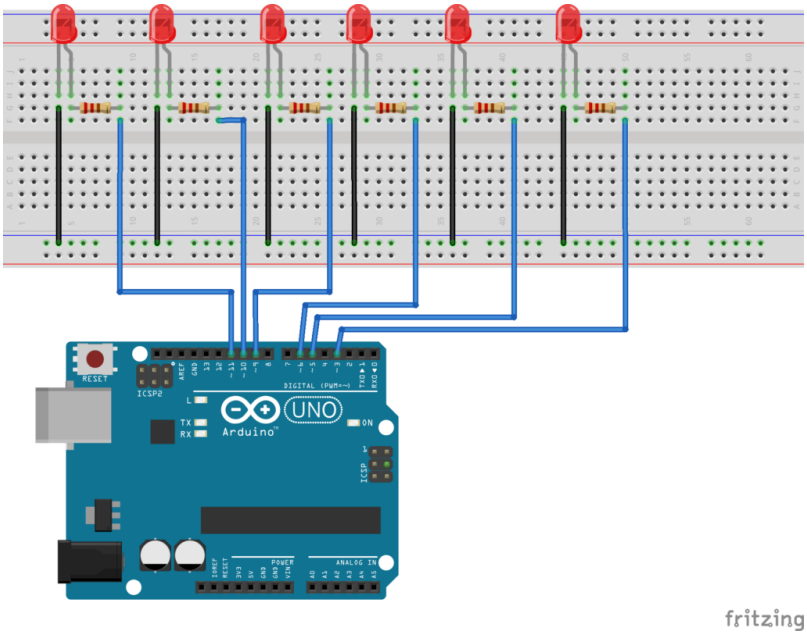
Fading led

Componenti elettronici:

- Arduino UNO
- Breadboard
- 6 Led
- 6 Resistenze (100 Ohm)

Teoria: Al fine di realizzare una barra LED (Light Emitting Diode), 6 diodi ad Emettitore di Luce sono stati utilizzati e collegati a differenti PIN digitali di Arduino. Come nelle lezioni precedenti ad ogni LED è associata una resistenza al fine di limitare il passaggio di corrente.
A differenza della barra led riportata nei prerequisiti, dove un solo led era acceso, in questa attività più led sono accesi in contemporanea. Nello specifico attraverso l'istruzione analogWrite l'intensità luminosa viene modulata in quattro differenti livelli.

Collegamento Circuitale:



Schema Circuitale

Codice:

Personalizzazioni: E’ possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *ledTime*. Modificando il suo valore infatti cambia la velocità di lampeggiamento della barra LED.

Barra LED (Knight Rider)

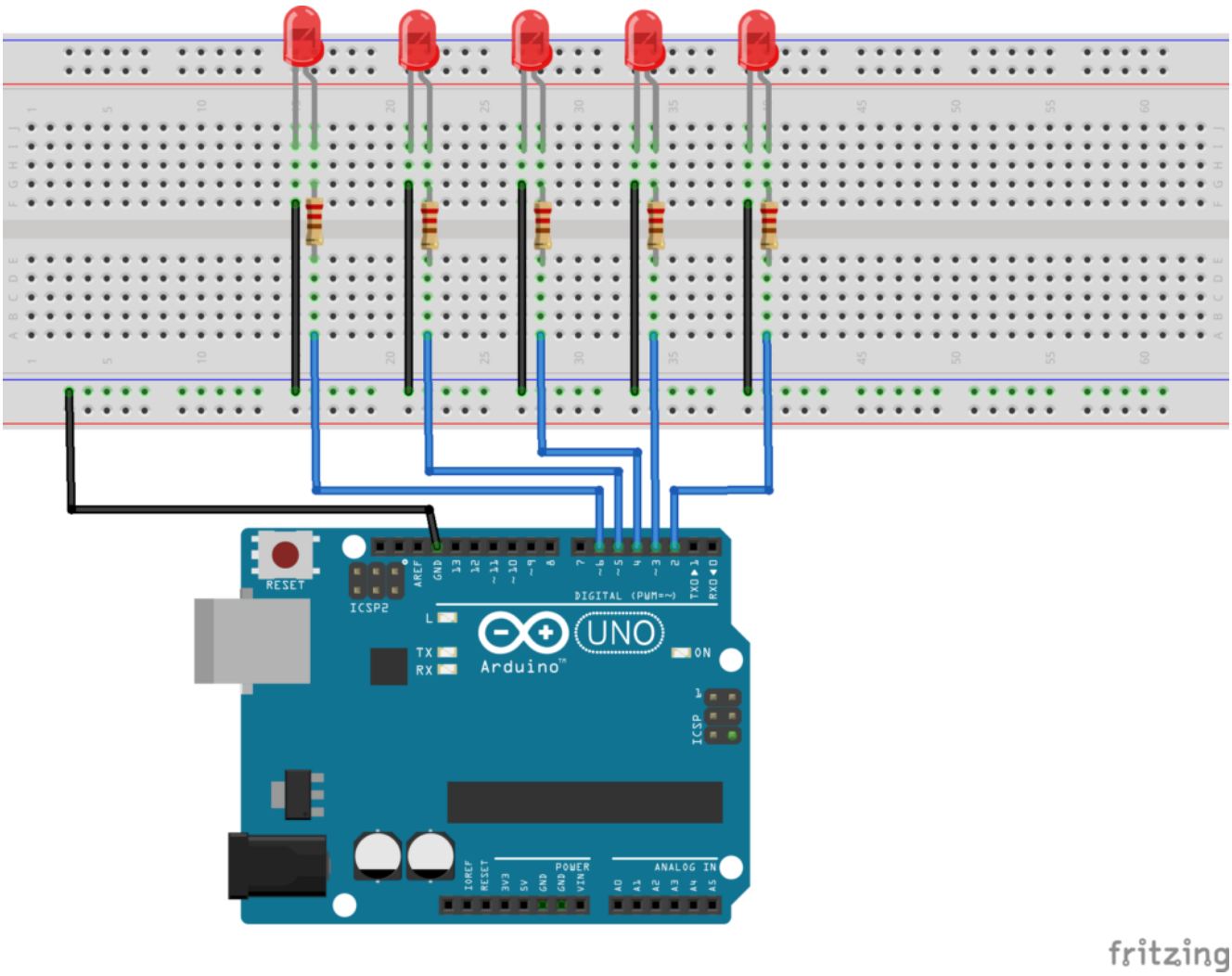
Obiettivo: Realizzazione di una barra LED.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 5 Led
- 5 Resistenze (100 Ohm)

Teoria: Al fine di realizzare una barra LED (Light Emitting Diode), 5 diodi ad Emettitore di Luce sono stati utilizzati e collegati a differenti PIN digitali di Arduino. Come nelle lezioni precedenti ad ogni LED è associata una resistenza al fine di limitare il passaggio di corrente.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-66275c2574fe5521475812/]

Tinkercad:

Personalizzazioni: E’ possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *ledTime*. Modificando il suo valore infatti cambia la

velocità di lampeggiamento della barra LED.
E’ inoltre possibile aggiungere ulteriori LED cambiando rispettivamente l’hardware ed il software presentato.