IL TUTORIAL DI BASE STARTER KIT PER UNO

V1.0.19.7.24

Prefazione

La nostra Compagnia

Fondata nel 2011, Elegoo Inc. è una fiorente società che si occupa di tecnologia dedicata alla ricerca e sviluppo, produzione e commercializzazione di hardware open-source. Situati a Shenzhen, la Silicon Valley della Cina, siamo cresciuti fino ad avere più di 150 dipendenti con più di 1.000 metri quadrati di fabbrica.

Le nostre linee di prodotti variano tra fili di DuPont, schede UNO R3, fino a starter kit pensati per clienti di qualsiasi livello interessati ad imparare e migliorarsi con Arduino. Vendiamo inoltre di accessori per Raspberry Pi come schermo 2.8'' TFT touch e STM32. In futuro vorremmo dedicare più energia e gli investimenti per le stampanti 3D e simili. Tutti i nostri prodotti seguono le norme internazionali di qualità e sono molto apprezzati in una grande varietà di mercati in tutto il mondo.

Sito ufficiale: http://www.elegoo.com

US Amazon storefront: http://www.amazon.com/shops/A2WWHQ25ENKVJ1

CA Amazon storefront: http://www.amazon.ca/shops/A2WWHQ25ENKVJ1

UK Amazon storefront: http://www.amazon.co.uk/shops/AZF7WYXU5ZANW

DE Amazon storefront: http://www.amazon.de/shops/AZF7WYXU5ZANW

FR Amazon storefront: http://www.amazon.fr/shops/AZF7WYXU5ZANW

ES Amazon storefront: http://www.amazon.es/shops/AZF7WYXU5ZANW

IT Amazon storefront: http://www.amazon.it/shops/AZF7WYXU5ZANW

Il nostro Tutorial

Questo Tutorial è stato progettato per i principianti. Imparerete tutte le nozioni di base su come utilizzare il controllore Arduino, sensori e componenti. Se volete studiare Arduino in modo più approfondito, vi consigliamo di leggere “Arduino Cookbook” scritto da Michael Margolis.

Alcuni codici in questo tutorial sono a cura di Simon Monk. Simon Monk è autore di una serie di libri relativi all'Open Source Hardware. Sono disponibili su Amazon: “Programming Arduino”, “30 Arduino Projects for the Evil Genius” e “Programming the Raspberry Pi”.

Assistenza Clienti

Essendo un’azienda di tecnologia siamo in costante e veloce crescita, continuando a fare del nostro meglio per offrire eccellenti prodotti e servizi di qualità. Per soddisfare le vostre aspettative e potete mettervi in contatto con noi semplicemente scrivendo a service@elegoo.com o EUservice@elegoo.com. Saremo molto felici di avere vostri commenti, critiche e suggerimento. Avrete risposta a tutte le vostre domande e ai vostri eventuali problemi entro 12 ore (24 ore durante le vacanze)

Packing list

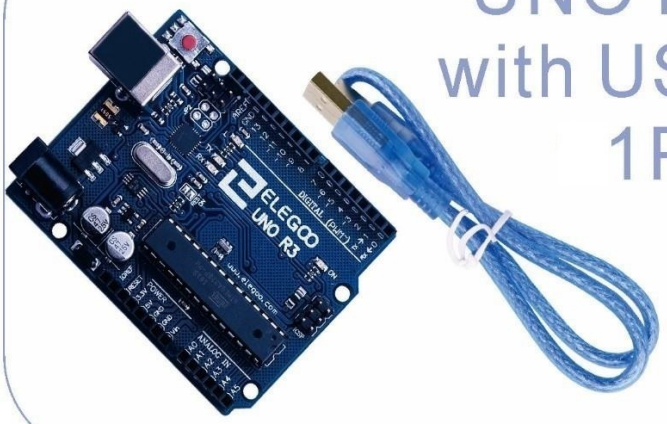
www.elegoo.com

RGB LED
2PCS

Photoresistor
(photocell)
1PC

Resistor
120PCS

UNO R3
with USB
1PC



Tilt Ball Switch
1PC

LED
30PCS

Button(Small)
5PCS

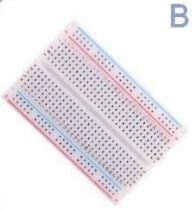
Active Buzzer
1PC

74HC595 IC
1PC

F-M Dupont Wire
5PCS

Breadboard
Jumper Wire
65PCS

Breadboard
1PC



Contact us : service@elegoo.com

Contenuti

Lezione 0 Installazione IDE

Introduzione

"Arduino Integrated Development Environment", IDE o Ambiente di Sviluppo Integrato è il lato software della piattaforma Arduino.

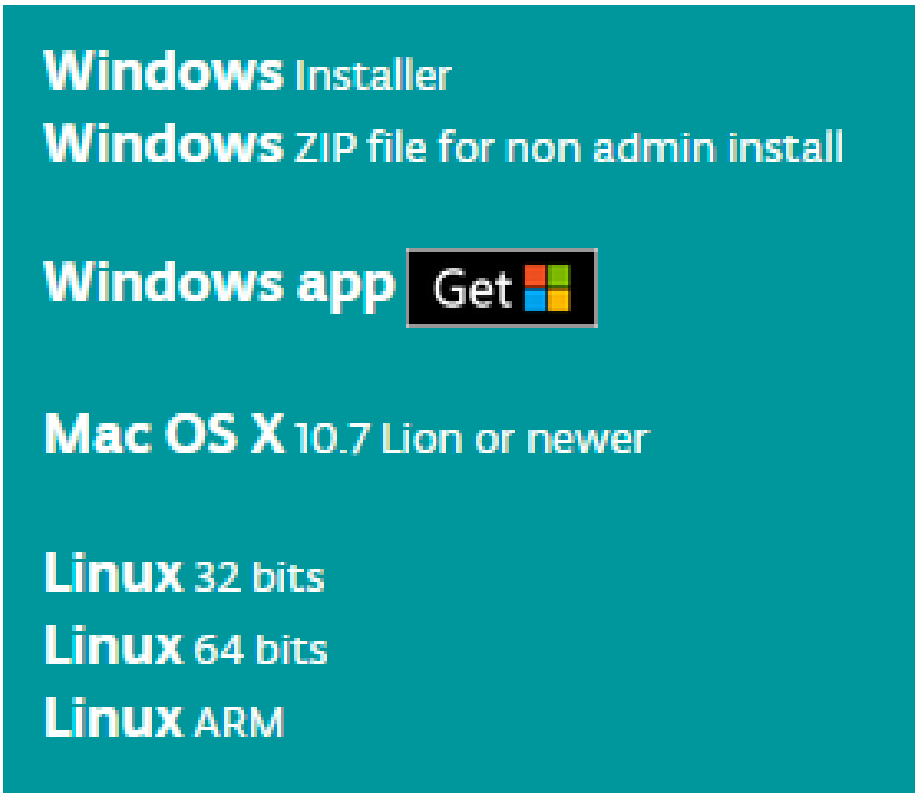
In questa lezione imparerai come configurare il computer per utilizzare Arduino e come impostarlo per le lezioni che seguono.

Il software di Arduino che verrà utilizzato per programmare la piattaforma Arduino è disponibile per Windows, Mac e Linux. Il processo di installazione è diverso per le tre piattaforme e purtroppo necessita di una certa quantità di lavoro manuale per essere installato

STEP 1: Visita https://www.arduino.cc/en/Main/Software e troverai la seguente schermata.

La versione disponibile sul sito è la più recente, potrebbe non corrispondere a quella in foto qui sopra.

STEP2： Scarica il software di sviluppo che è compatibile con il sistema operativo del computer. In questo esempio utilizzeremo Windows.

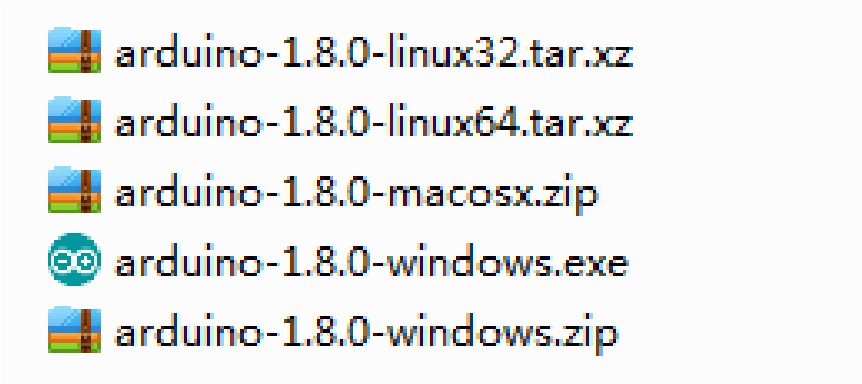


Clicca su *Windows Installer*.

Premi su *JUST DOWNLOAD*.

La versione 1.8.0 è disponibile anche nel materiale che ti abbiamo fornito.

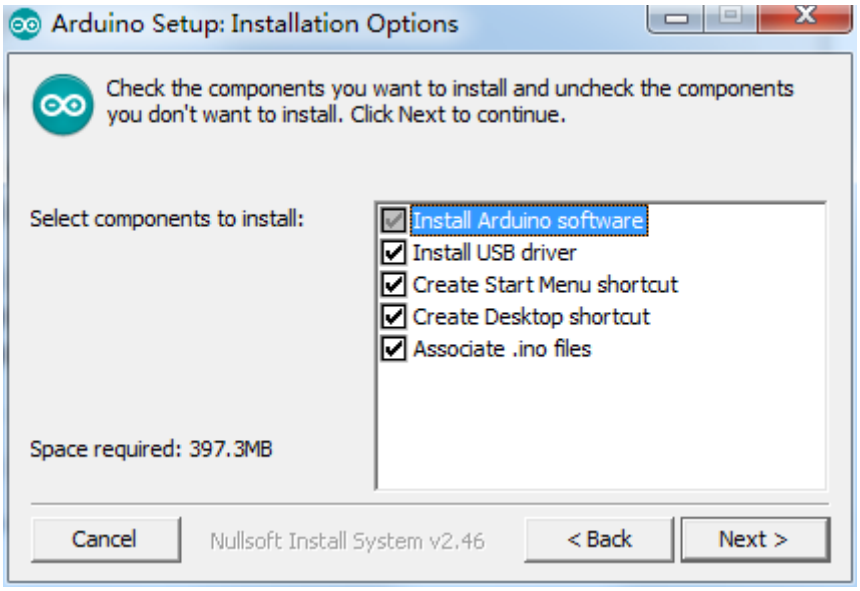
Nota: Quando è stato scritto questo tutorial la 1.8.0 era la versione più recente esistente.



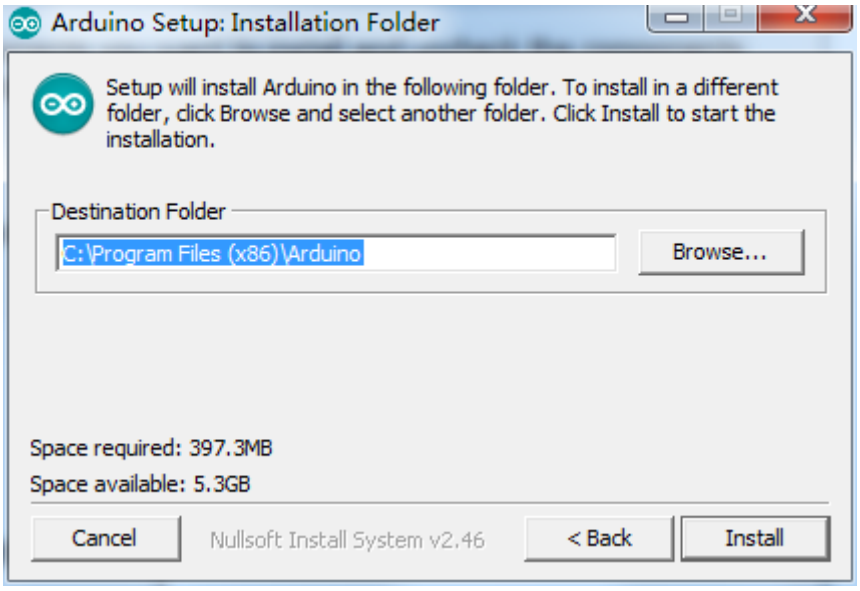
Installazione Arduino (Windows)

Installa Arduino con il pacchetto di installazione exe.

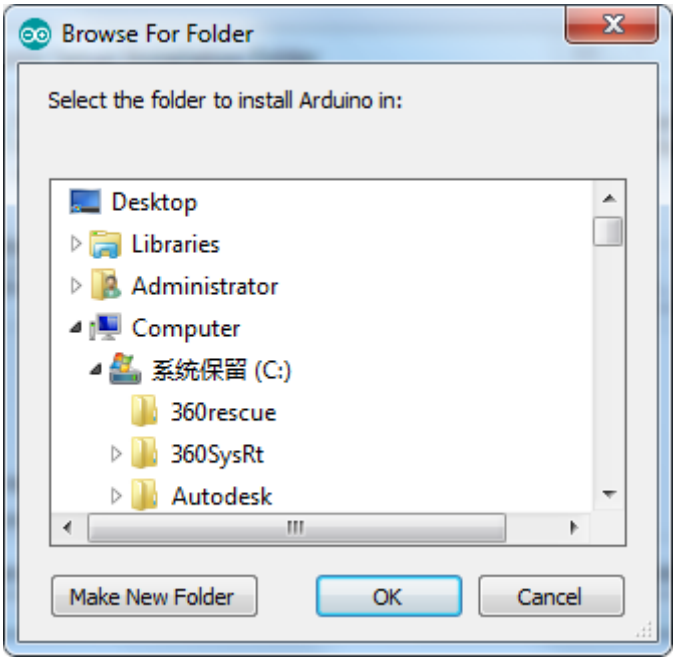
Premi su *I Agree(Accetto)* per proseguire con la prossima interfaccia.



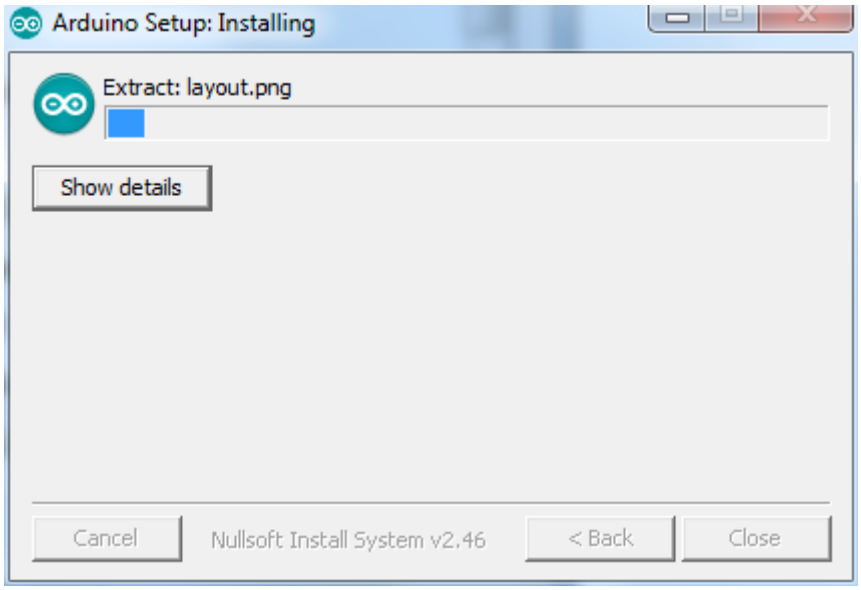
Premi su *Next(Avanti)*



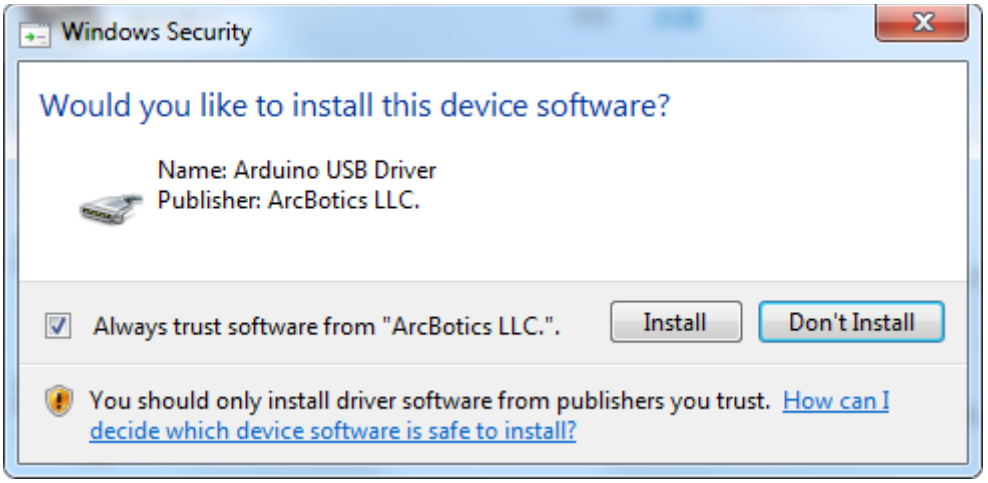
Puoi premere su Browse(Naviga)… per scegliere il percorso di installazione o digitare direttamente la cartella che vuoi utilizzare.



Premi su *Install(installa)* per dare via al processo di installazione.

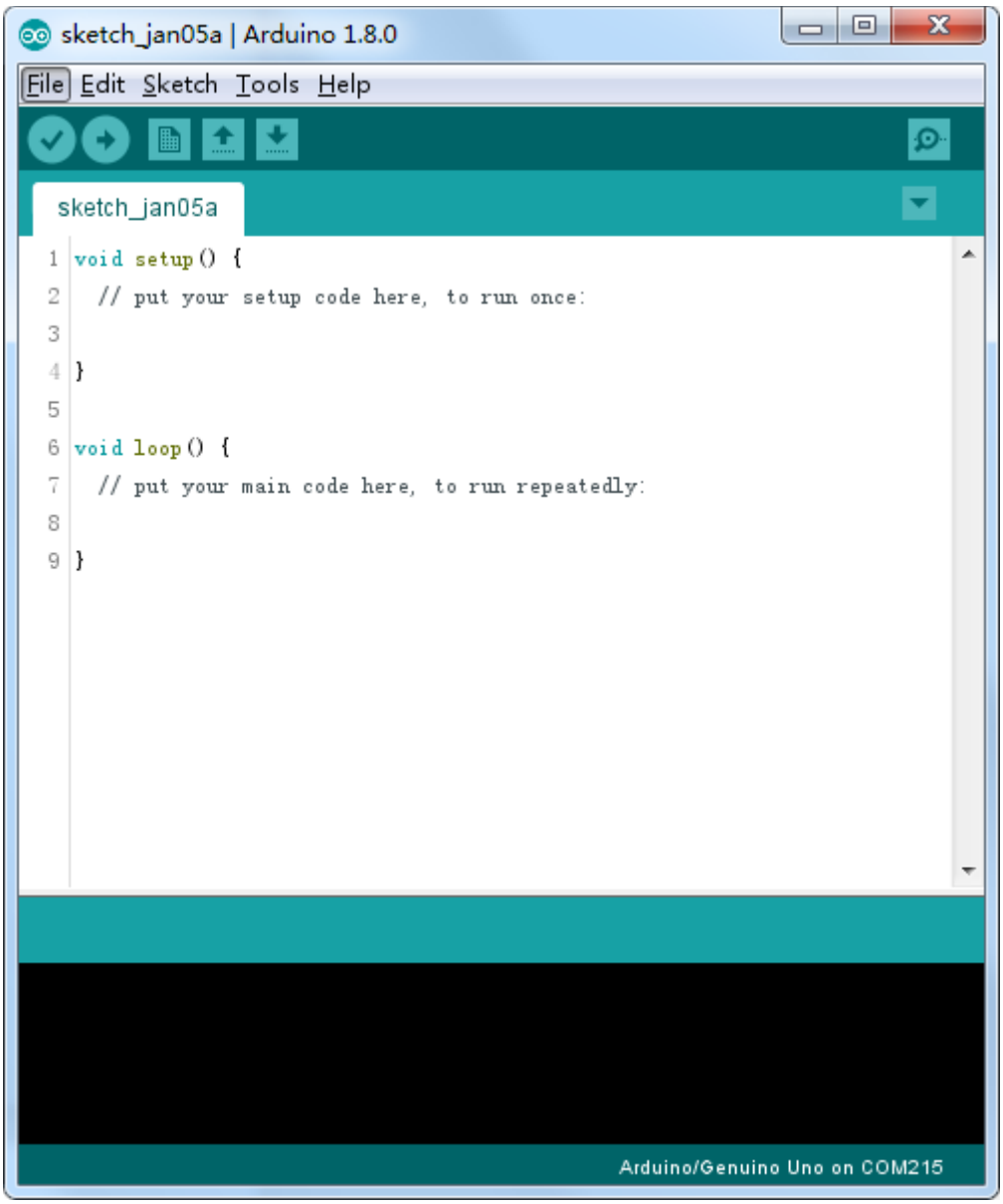


Al termine, apparirà la seguente interfaccia, premi su *Install(installa)* per avviare l'ultimo passaggio dell'installazione.



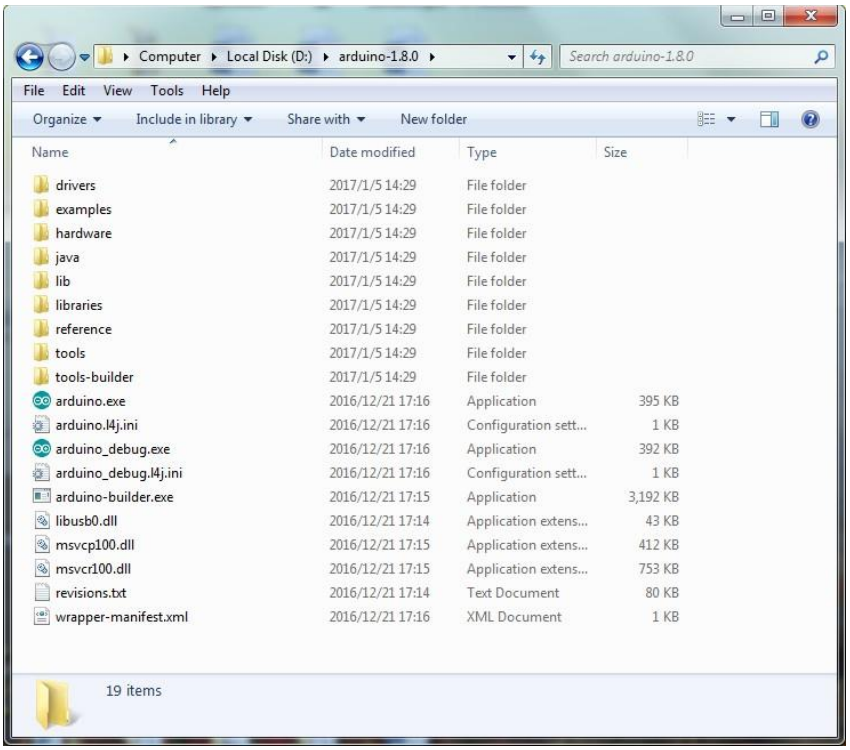
Successivamente, verrà visualizzata la seguente icona sul desktop.

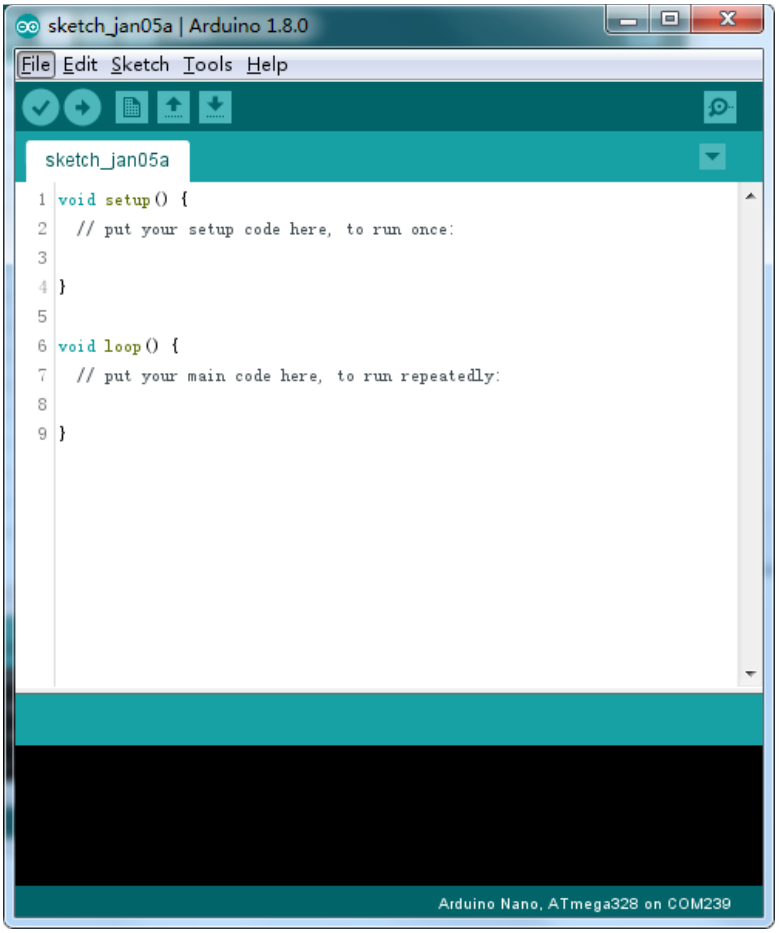
Fare doppio clic sull’icona per accedere all'ambiente di sviluppo.



Puoi scegliere direttamente il pacchetto di installazione e passare direttamente alla sezione successive, saltando I prossimi passaggi. Se invece vuoi imparare alcuni metodi diversi per installare l’IDE continua a leggere questa sezione.

Decomprimi il file zip scaricato, fai doppio clic su “Arduino.exe” per aprire il programma ed accedere all'ambiente di sviluppo di Arduino.





Tuttavia, questo metodo di installazione richiede l'installazione separata di driver.

La cartella Arduino contiene sia il programma di sviluppo Arduino sia i driver che permettono ad Arduino di interagire con il computer tramite un cavo USB.

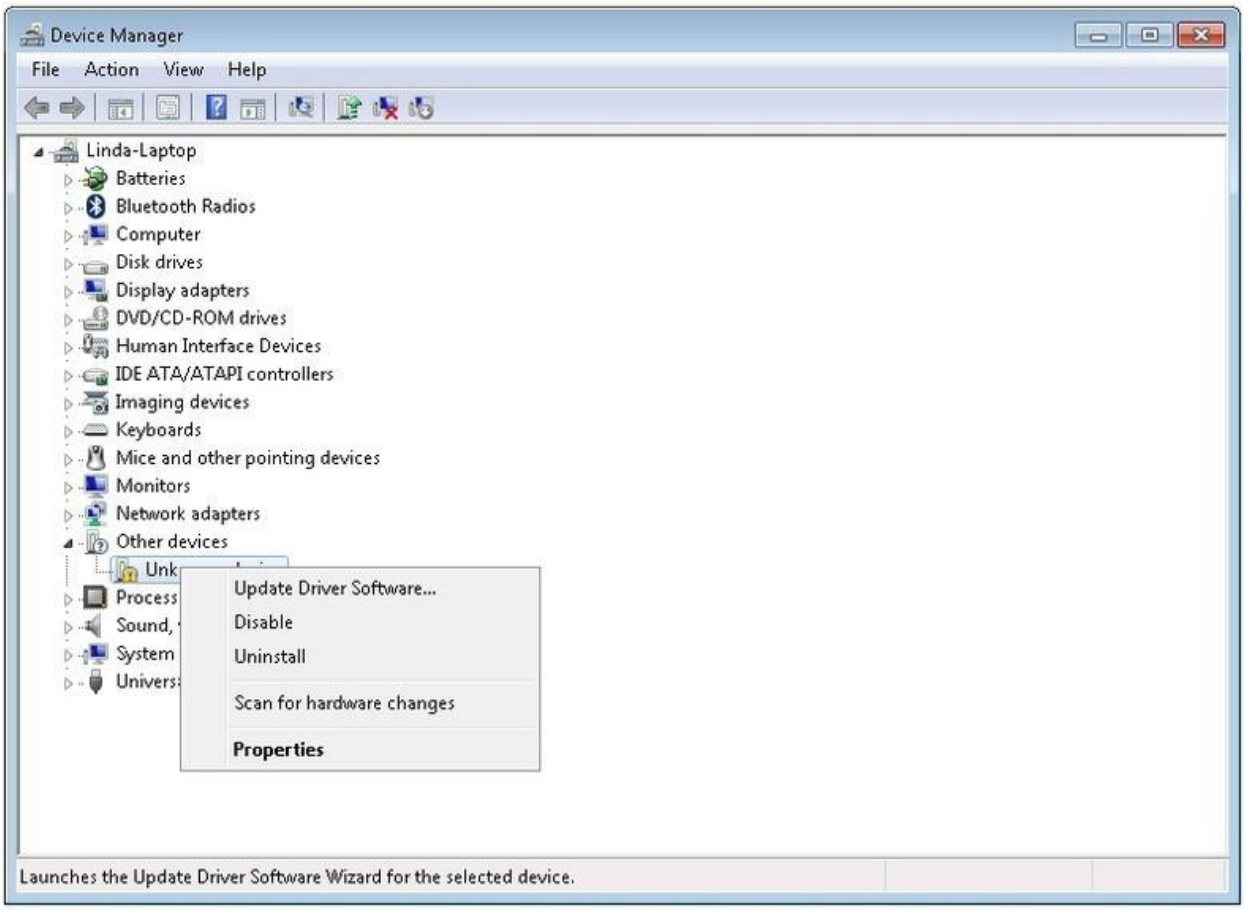
Prima di lanciare il software di Arduino, ricordati di installare i driver USB come spiegato nei passaggi seguenti:

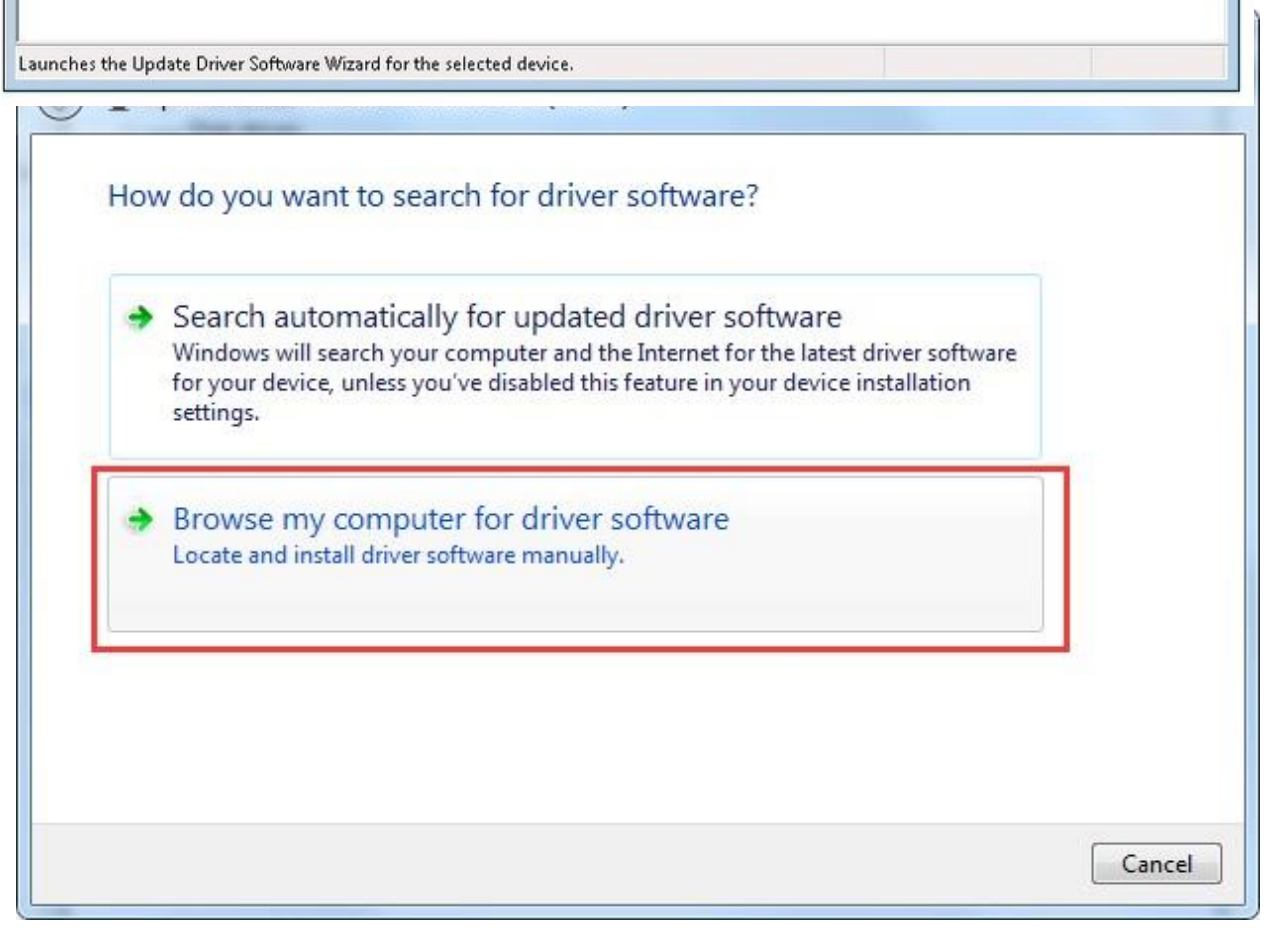
Collega un'estremità del cavo USB alla scheda Arduino e l'altra in una presa USB del computer. Si accenderà la spia LED di alimentazione della scheda Arduino e potresti visualizzare sul pc un messaggio simile a “Trovato nuovo hardware”. Ignora il messaggio e annulla tutti gli eventuali tentativi di Windows di installazione automatica dei driver di Arduino.

Il metodo più affidabile per installare dei driver USB è quello di utilizzare Gestione periferiche. Per accedere al pannello di Gestione Risorse sono possibili modi diversi a seconda della versione di Windows.

In Windows 7, è necessario innanzitutto aprire il Pannello di controllo, quindi selezionare l'opzione di visualizzazione per icone, e cliccare poi su Gestione periferiche nell'elenco.

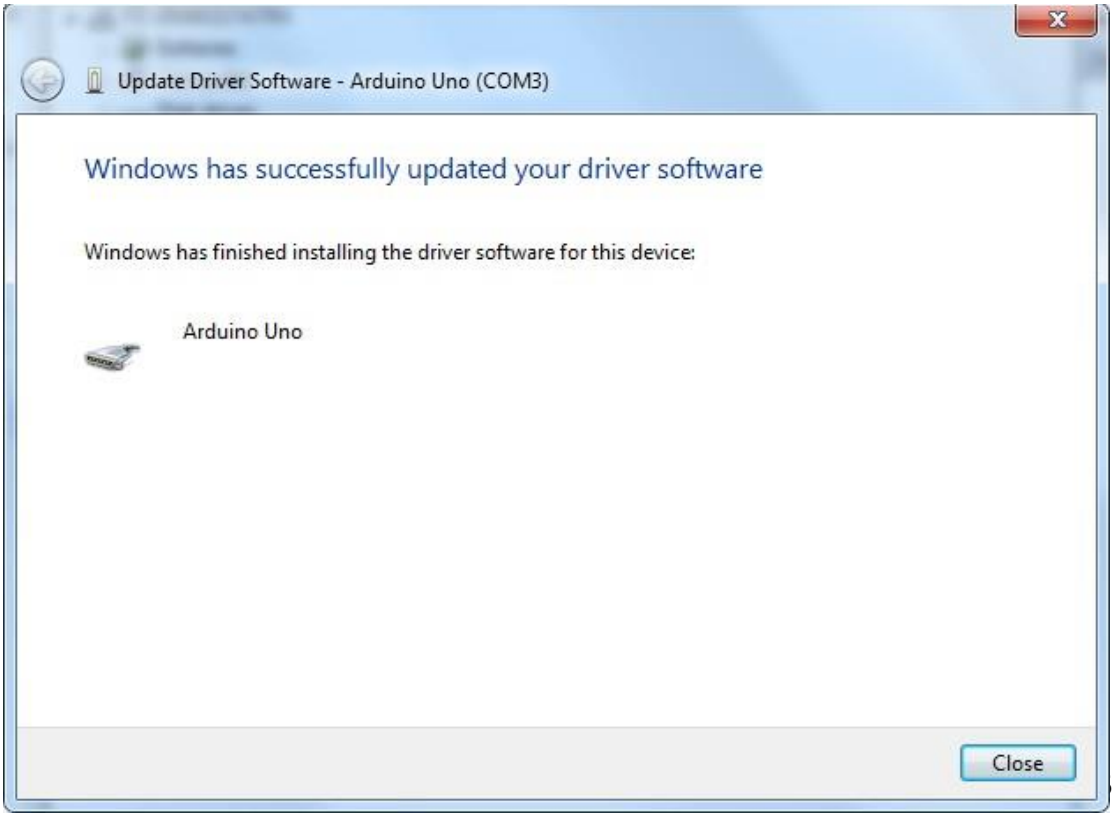
Sotto "altri dispositivi", troverai 'periferica sconosciuta' con un piccolo triangolo di avvertimento giallo sull'icona. Questa è l'icona relativa al tuo Arduino.





Ti Verrà quindi chiesto di scegliere tra 'Cerca automaticamente un driver aggiornato' o 'Cerca sul computer il software del driver'. Selezionare l'opzione per cercare il driver sul pc e navigare fino alla cartella scaricata in precedenza X:\ arduino1.8.0 \ drivers.

Fare clic su 'Avanti' e verrà visualizzato un avviso di protezione, consentire al software di proseguire con l'installazione. Una volta che il software sarà stato installato, visualizzerai un messaggio di conferma.



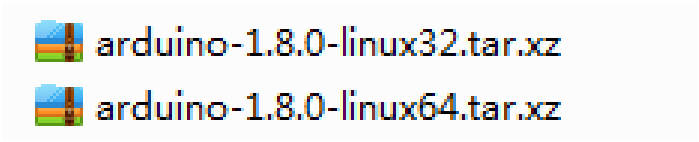
Gli utenti Windows possono saltare le istruzioni di installazione per i sistemi Mac e Linux ed andare direttamente alla Lezione 1.
Gli utenti Mac e Linux possono continuare a leggere questa sezione.

Installazione Arduino (Mac OS X)

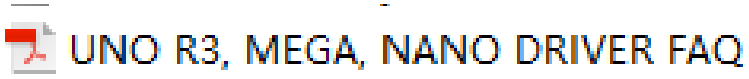
Scarica e decomprimi il file zip, fai doppio clic su Arduino.app per accedere all'IDE di Arduino. Il sistema ti chiederà di installare la libreria di runtime Java se non la hai già installata nel tuo computer. Una volta completata l'installazione ti sarà possibile eseguire l'IDE di Arduino.

Installazione Arduino (Linux)

Dovrai utilizzare il comando make install. Se utilizzi il sistema Ubuntu, ti consigliamo di installare Arduino IDE dal centro software di Ubuntu.



TIPS: In caso di problemi nell'installazione dei driver, si prega di fare riferimento alla ONU R3, UNO, NANO DRIVER FAQ.



Lezione 1 Aggiunta delle librerie e apertura della porta seriale

Installare Ulteriori Librerie Arduino

Una volta che avrai preso dimestichezza con il software di Arduino e utilizzato le funzioni di base, ti consigliamo di potenziare le capacità del tuo Arduino con librerie aggiuntive.

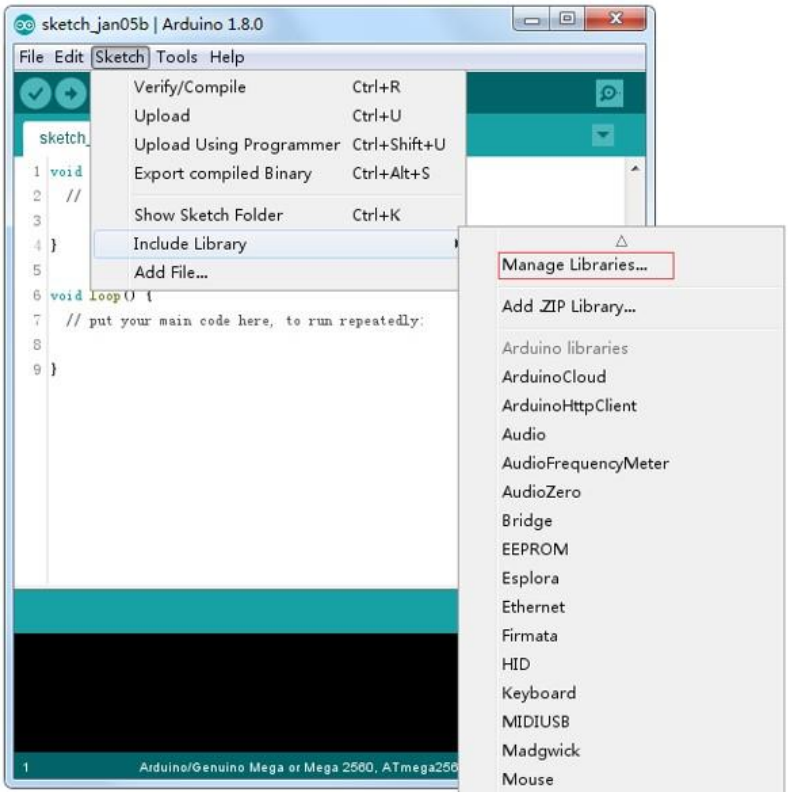
Cosa sono le librerie?

Le librerie sono delle raccolte di codici già scritti che rendono facile ed intuitivo il l'utilizzo di sensori, la visualizzazione dei dati, ecc. Ad esempio, la libreria incorporata LiquidCrystal rende molto semplice la comunicazione tra la scheda Arduino e il modulo display LCD. Ci sono centinaia di librerie disponibili per il download su Internet. Le librerie integrate e alcune di queste librerie aggiuntive verranno utilizzate in questi tutorial. Per utilizzare le librerie aggiuntive, è necessario che vengano installate.

Come installare una libreria

Utilizzando il gestore di librerie.

Per installare una nuova libreria nel tuo ambiente di sviluppo Arduino dovrai utilizzare il Gestore librerie (disponibile da IDE versione 1.8.0). Aprire l'IDE e fare clic sul menu "Sketch" e poi: “Include Library” > “Manage Libraries”.

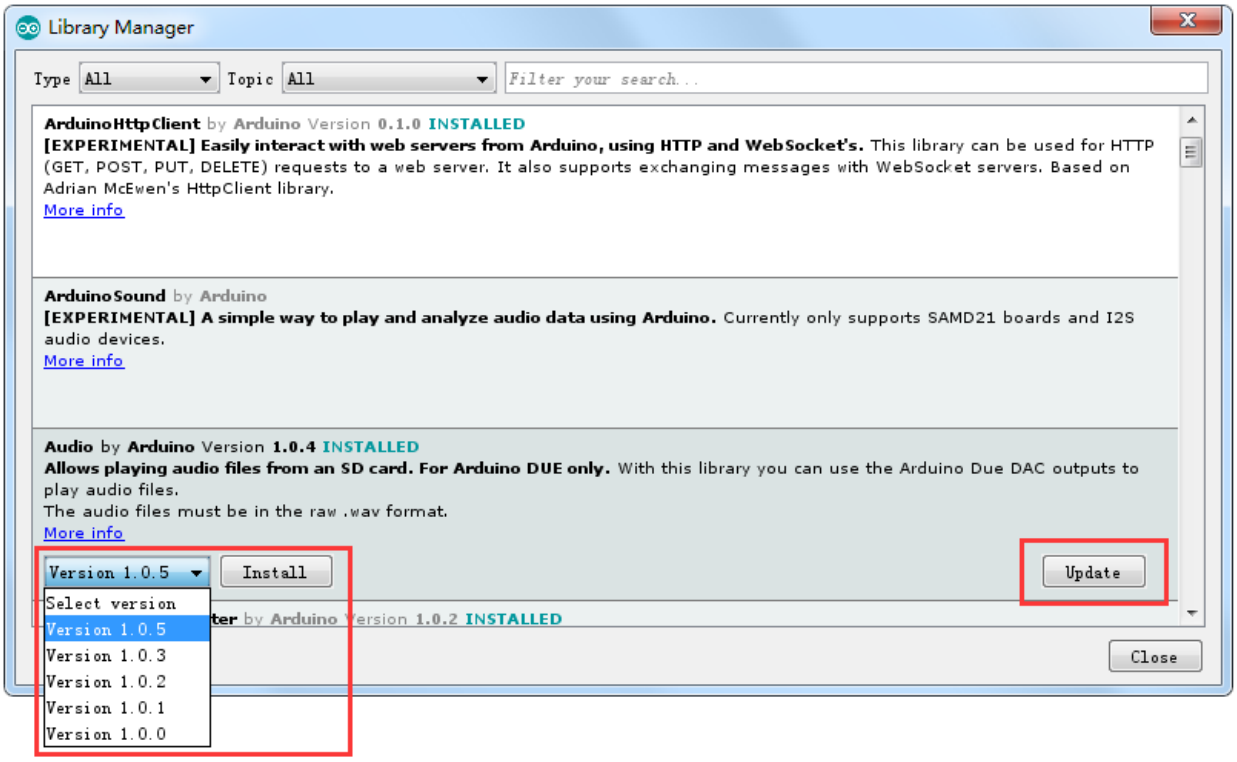


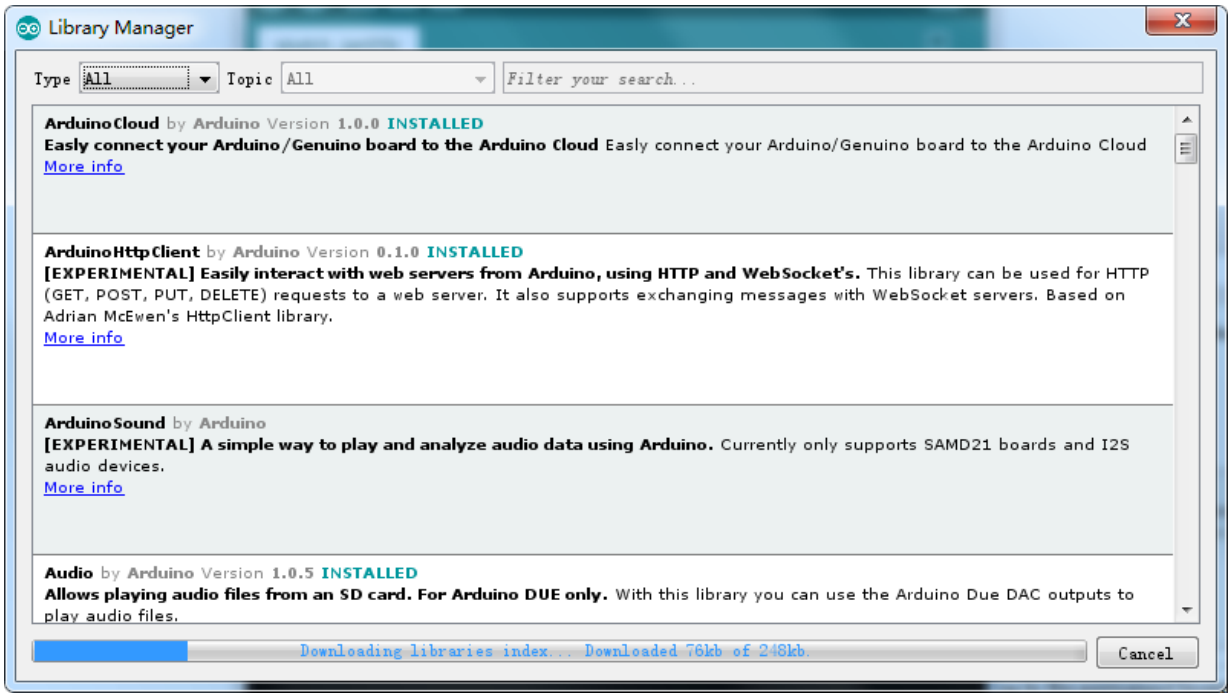
A questo punto si aprirà il gestore delle librerie e troverai un elenco di librerie già installate o pronte per l'installazione.

In questo esempio installeremo la libreria Bridge. Scorri l'elenco per trovarla, quindi seleziona la versione della libreria che vuoi installare. A volte solo una versione della libreria è disponibile. Se non viene visualizzato il menu di selezione della versione, non preoccuparti: è normale.

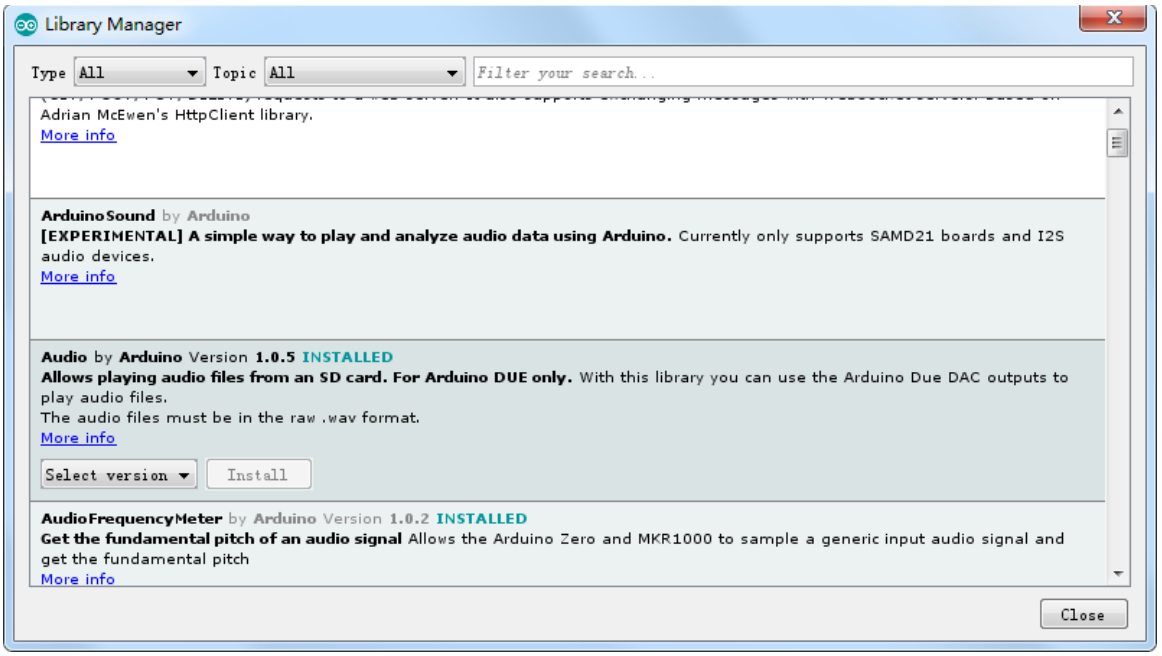
Capita di dover essere pazienti con il gestore di librerie, proprio come mostrato in figura.

Clicca su aggiorna e attendi.





Infine, fare clic su installa e attendi che l'IDE installi la nuova libreria. Il download può richiedere del tempo a seconda della velocità di connessione. Una volta finita, il tag “INSTALLED” dovrebbe comparire accanto alla libreria Bridge. Puoi ora chiudere la gestione delle librerie.



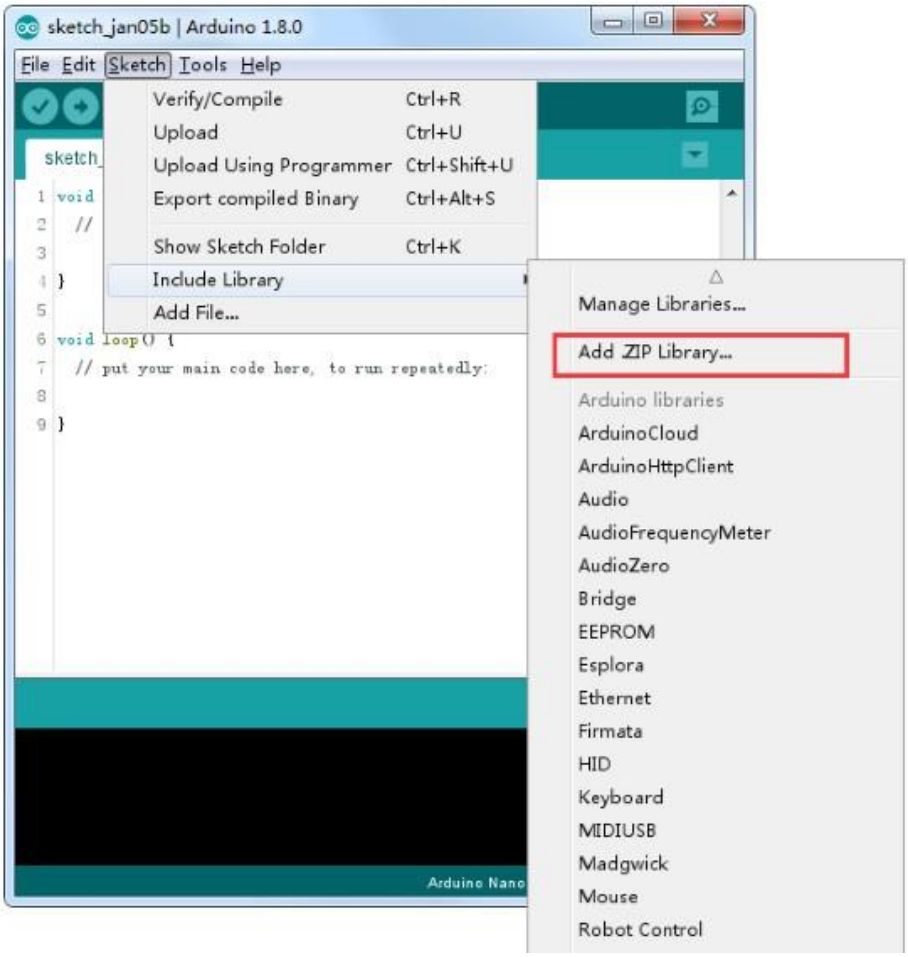
Ora ti sarà possibile trovare la nuova libreria disponibile nel menu: Library.

Se vuoi aggiungere una tua libreria all’elenco delle librerie disponibili in questa raccolta puoi fare una richiesta sull’apposita repository di Github.

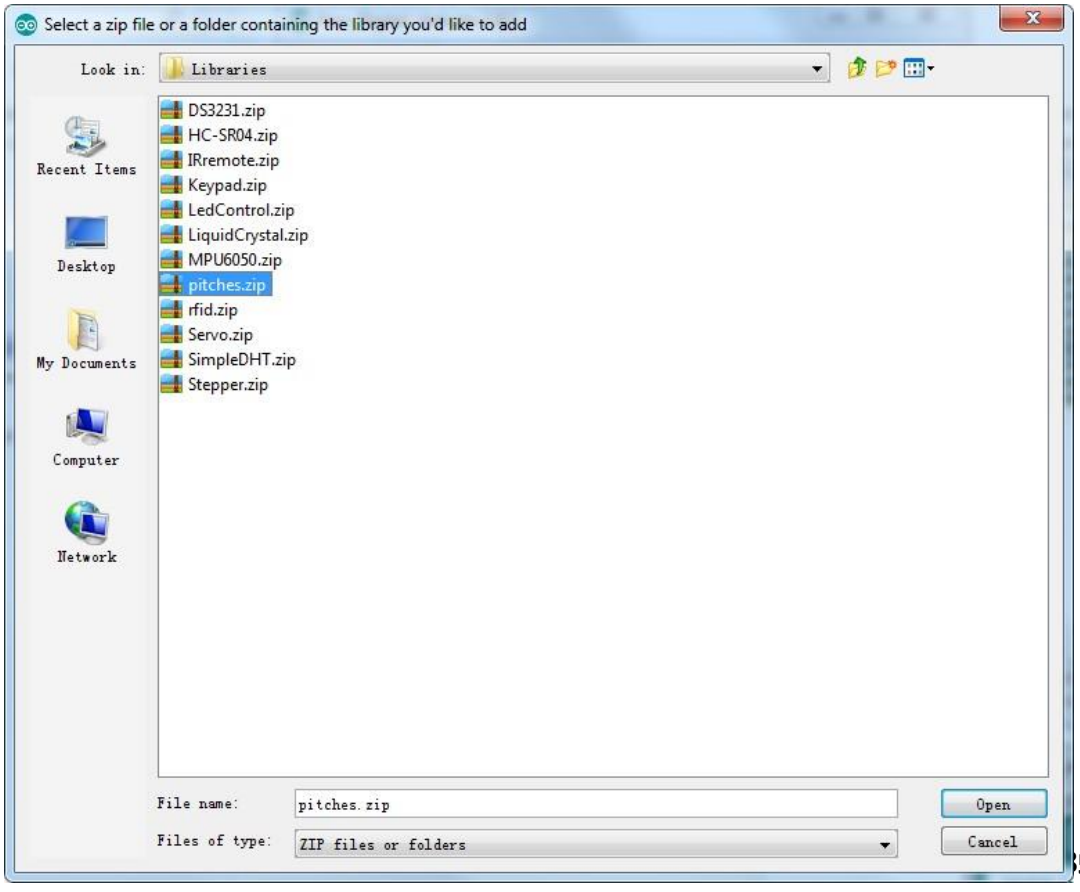
Importare una libreria in format .zip

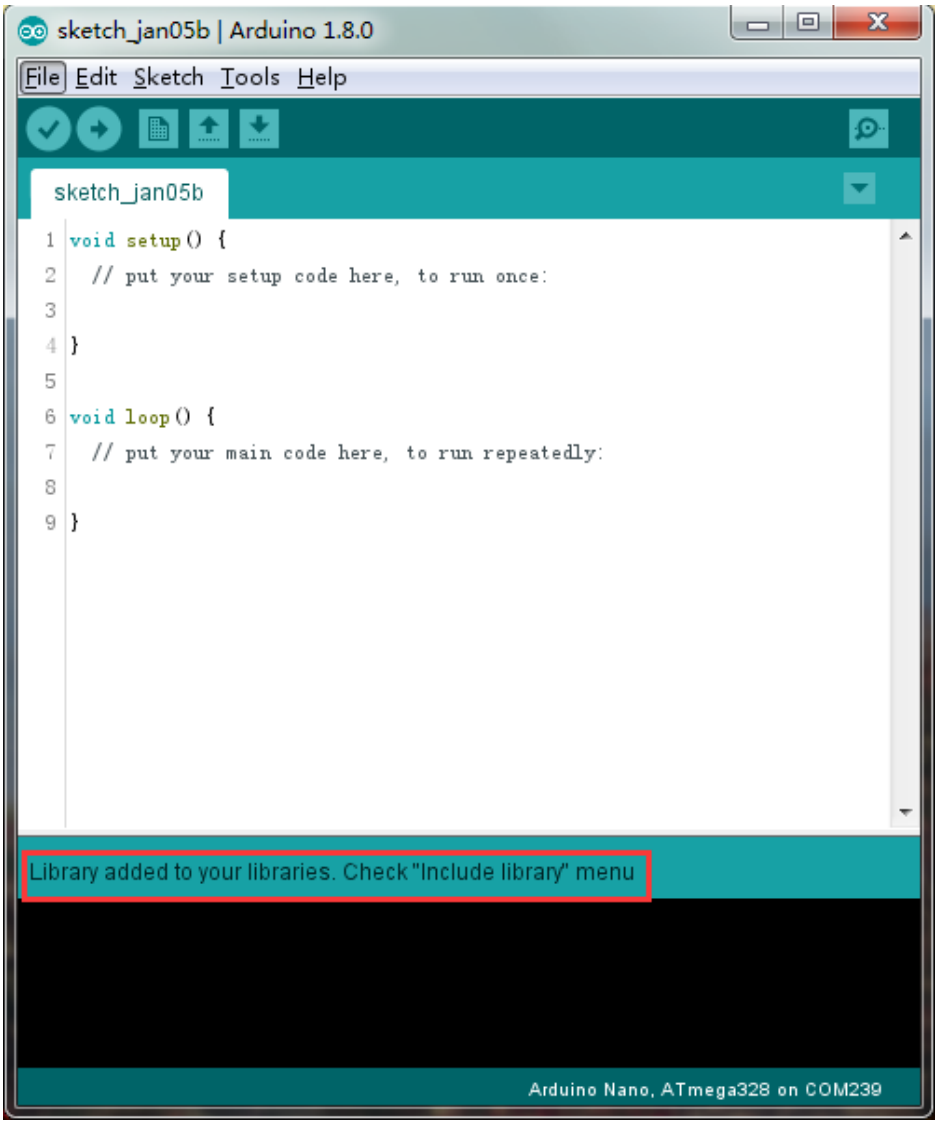
Le librerie sono spesso distribuite come file ZIP o come cartelle. Il nome della cartella è il nome della libreria. All'interno della cartella ci sarà un file .cpp, un file .h e spesso un file keywords.txt, la cartella con gli esempi, e altri file richiesti dalla libreria. A partire dalla versione 1.0.5, è possibile installare le librerie di terze parti nell'IDE. Non è necessario decomprimere le librerie zip scaricate, lasciarle così come sono.

Nel IDE di Arduino, navigare in Sketch > Include Library. Nella parte superiore del menu a tendina e selezionare l'opzione "Add .ZIP Library".



Ti verrà richiesto di selezionare la libreria che desideri aggiungere. Navigare fino alla posizione del file .zip scelto e aprirlo.





Torna al menu Sketch> Import Library. Ora dovresti vedere la libreria in fondo al menu a tendina. Ora è pronta per essere utilizzata nel vostro codice.

Il file zip sarà stato decompresso nella cartella library di Arduino.

NB: la libreria sarà disponibile da utilizzare, ma gli esempi per la libreria non saranno visibili in File> Esempi fino a che l'IDE sarà riavviato.

Quei due sono I metodi più comuni. i sistemi Mac e Linux possono essere gestiti allo stesso modo. L'installazione manuale che verrà introdotta qui di seguito può essere utile in alcuni rari casi e se non ne vedi la necessità puoi saltare il seguente paragrafo.

Installazione manuale

Per installare la libreria, prima chiudere l'applicazione Arduino. Poi decomprimere il file ZIP contenente la libreria. Ad esempio, se si sta installando una libreria chiamata

"ArduinoParty", decomprimere ArduinoParty.zip. Dovrebbe contenere una cartella chiamata ArduinoParty, contenente i file di cui si parlava al paragrafo sopra ArduinoParty.cpp e ArduinoParty.h. (Se i file cpp e .h non sono in una cartella, è necessario crearne una. In questo caso, devi creare una cartella chiamata "ArduinoParty" e spostare in essa tutti i file che erano nel file ZIP: ArduinoParty.cpp, ArduinoParty.h, etc.)
Trascinare la cartella ArduinoParty in nella tua cartella delle librerie. In Windows, la troverai probabilmente nel percorso "My Documents \ Arduino \ libraries". Per gli utenti Mac, troverai probabilmente nel percorso "Documenti / Arduino / librerie". Su Linux, troverai la cartella "library" nel tuo sketchbook .

La cartella library di Arduino dovrebbe apparire così (su Windows):

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp
My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h
My Documents\Arduino\libraries\ArduinoParty\examples

O così (su Mac e Linux):

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp
Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h
Documents/Arduino/libraries/ArduinoParty/examples
....

Ci potrebbero essere più file oltre ai soliti .cpp e .h, l'importante è assicurarsi che siano tutti lì. (La library non funziona se si mette I file cpp e .h direttamente nella cartella library o se sono annidati in una cartella di troppo. Ad esempio: Documents \ Arduino \ libraries \ ArduinoParty.cpp e Documents \ Arduino \ le librerie \ ArduinoParty \ ArduinoParty \ ArduinoParty.cpp non funzioneranno.)
Riavviare l'applicazione Arduino. Assicurarsi che la nuova libreria compaia nel menu di Sketch->Import Library del software. Questo è tutto! Hai installato una libreria!

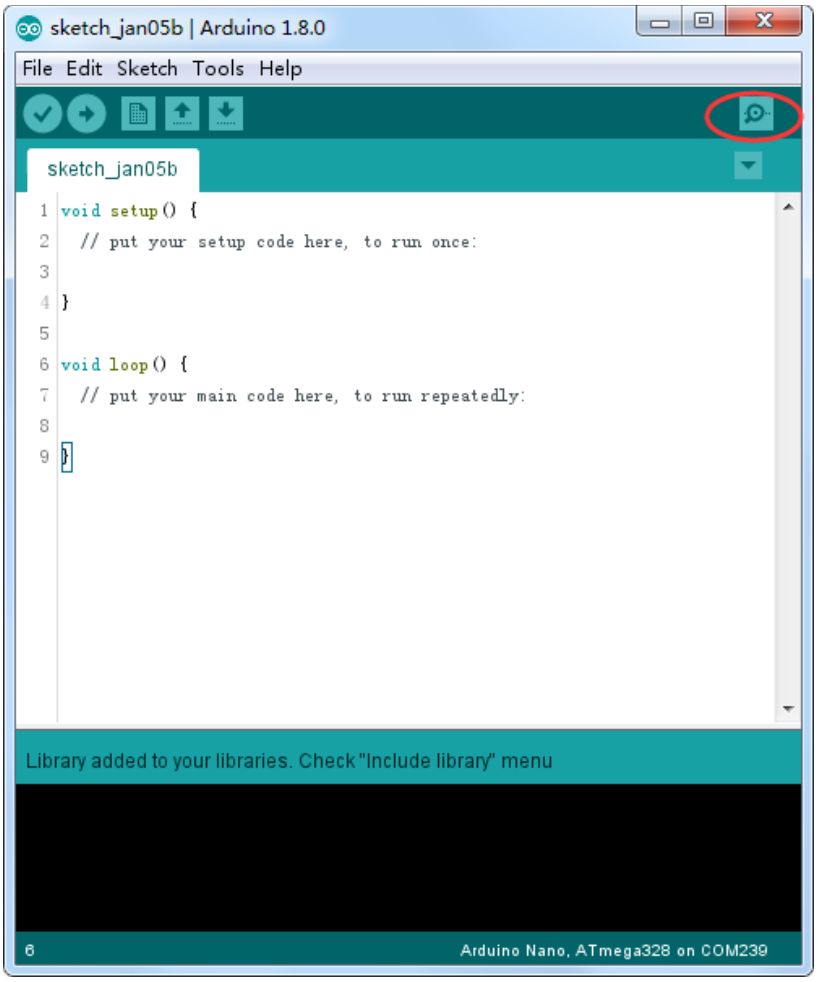
Arduino Serial Monitor (Windows, Mac, Linux)

L'ambiente integrato di sviluppo (IDE) di Arduino è il lato software della piattaforma Arduino. E dato che l'utilizzo del terminale è una grossa fetta del lavoro con Arduino e altri microcontrollori, gli sviluppatori hanno deciso di includere un terminale

seriale nel software. All'interno dell'IDE di Arduino, questo terminale è chiamato il Serial Monitor.

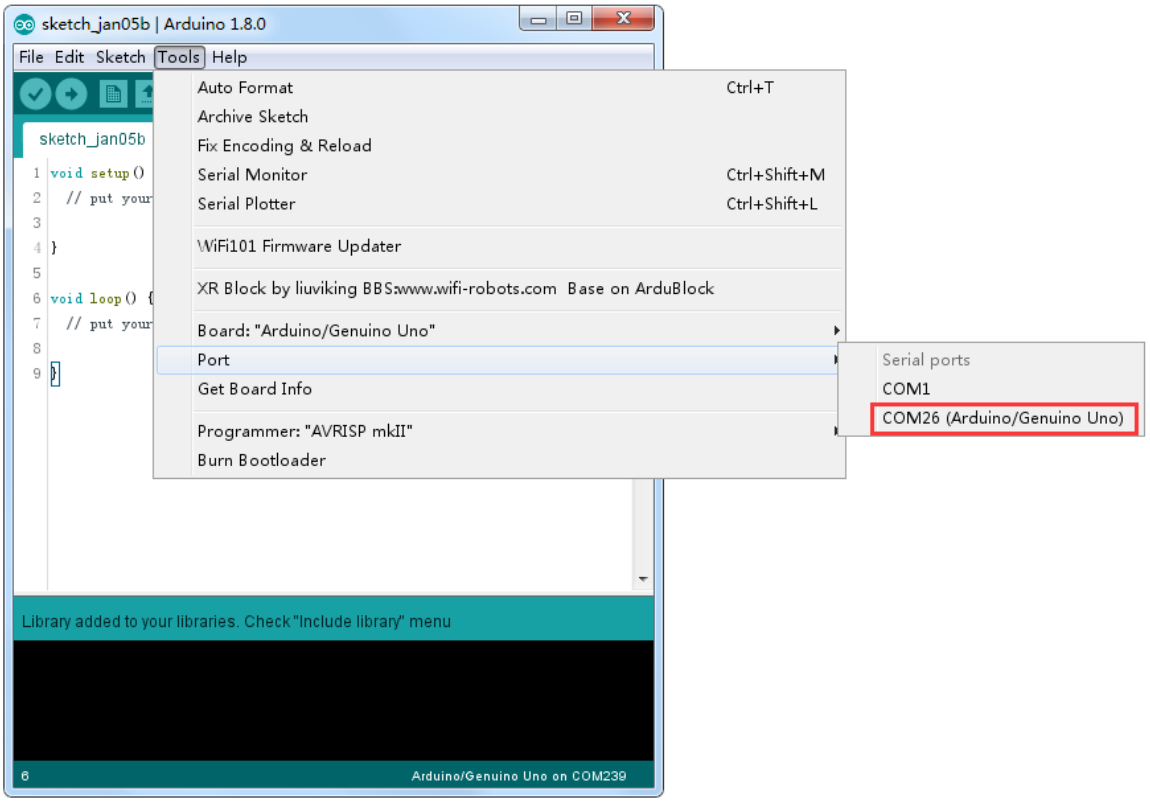
Connettersi al Monitor Seriale

Il Monitor Seriale è presente di ogni versione di Arduino. Per aprirlo, è sufficiente fare clic sull'icona del monitor seriale, come da immagine sottostante.

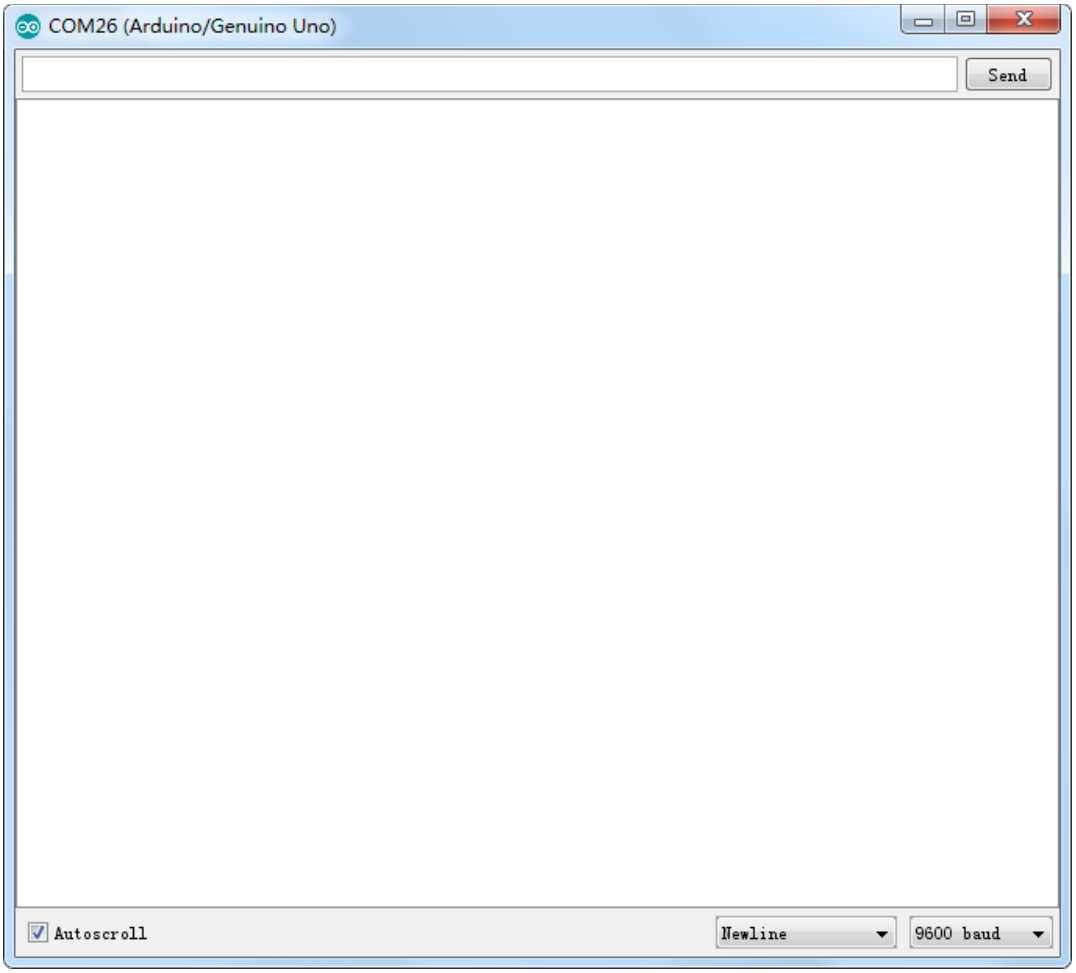


Nello stesso modo in cui si seleziona una porta per il caricamento di codice Arduino, si sceglie quale porta aprire nel Serial Monitor. Andare in Tools -> Serial Port, e selezionare la porta corretta.

Consiglio: scegli la stessa porta COM che hai trovato in precedenza in Gestione periferiche.

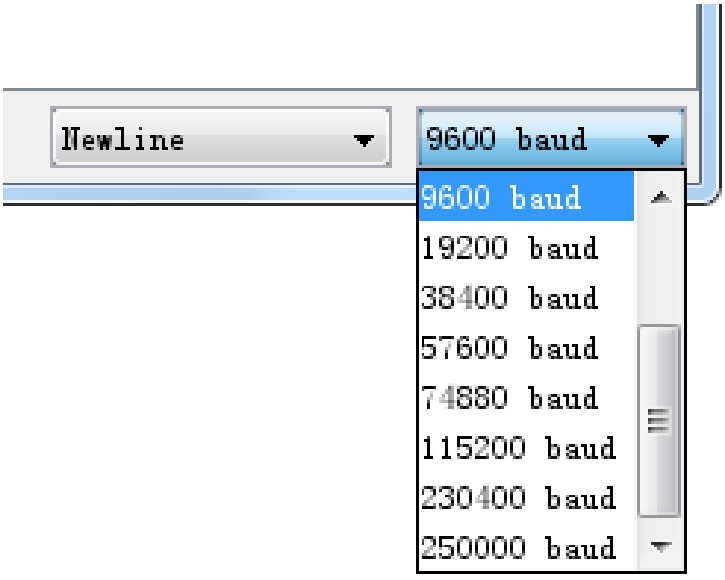


Una volta aperto dovresti vedere qualcosa di simile:

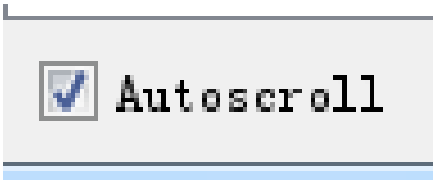


Impostazioni

Il Serial Monitor ha impostazioni limitate, ma sufficienti per gestire la maggior parte delle esigenze di comunicazione seriale. La prima impostazione modificabile è la velocità di trasmissione. Fai clic sul menu a discesa e seleziona la velocità di trasmissione corretta. (9600 baud)



Infine puoi scegliere se impostare l'Autoscroll o meno utilizzando la checkbox nell'angolo in basso a sinistra



Pro

Il Serial Monitor è un modo ottimo, rapido e semplice per stabilire una connessione seriale con Arduino. Se si sta già lavorando in Arduino, non c'è alcun bisogno di aprire un terminale separato per visualizzare i dati.

Contro

La mancanza di impostazioni lascia molto a desiderare nel Serial Monitor, e, per le comunicazioni seriali avanzate, potrebbe non essere abbastanza.

Lezione 2 Lampeggio (Blink)

Introduzione

In questa lezione imparerai come programmare il tuo UNO R£ per far lampeggiare il LED già presente sulla scheda e come scaricare il programma in piccoli passi.

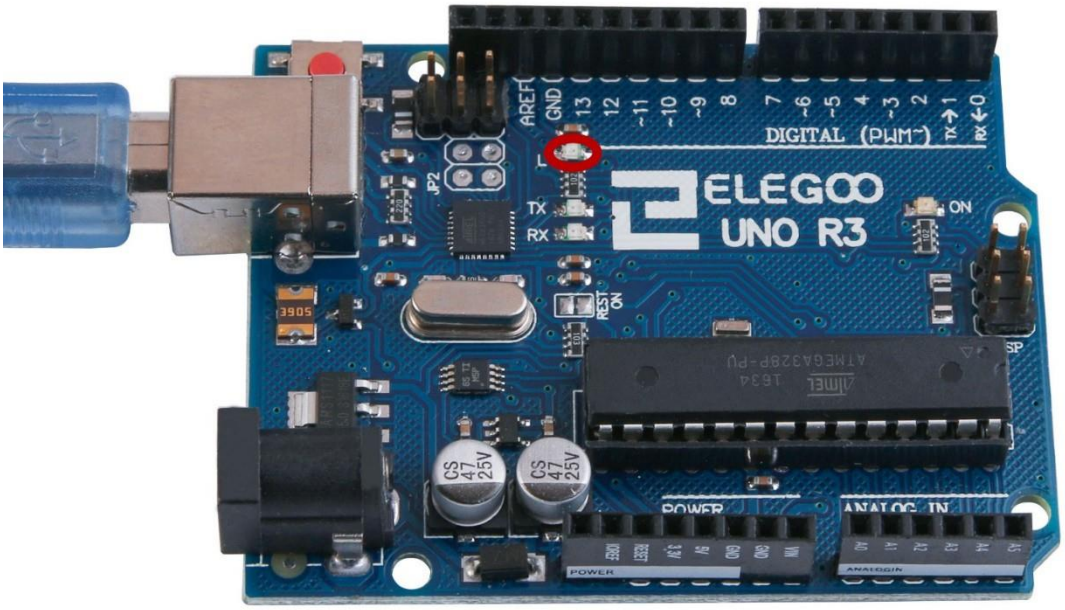
Componenti Richiesti:

(1) x Elegoo UNO R3

Informazioni di Base

La scheda UNO R3 ha linee di connettori lungo i tre lati che vengono utilizzati per la connessione a vari dispositivi elettronici e schede “shield” che estendono le sue capacità.

La scheda ha anche un LED che è possibile controllare direttamente dal codice. Questo LED è situato in alto a sinistra sulla scheda UNO R3 ed è spesso indicato con il simbolo 'L' LED come quello cerchiato sull’immagine sottostante.



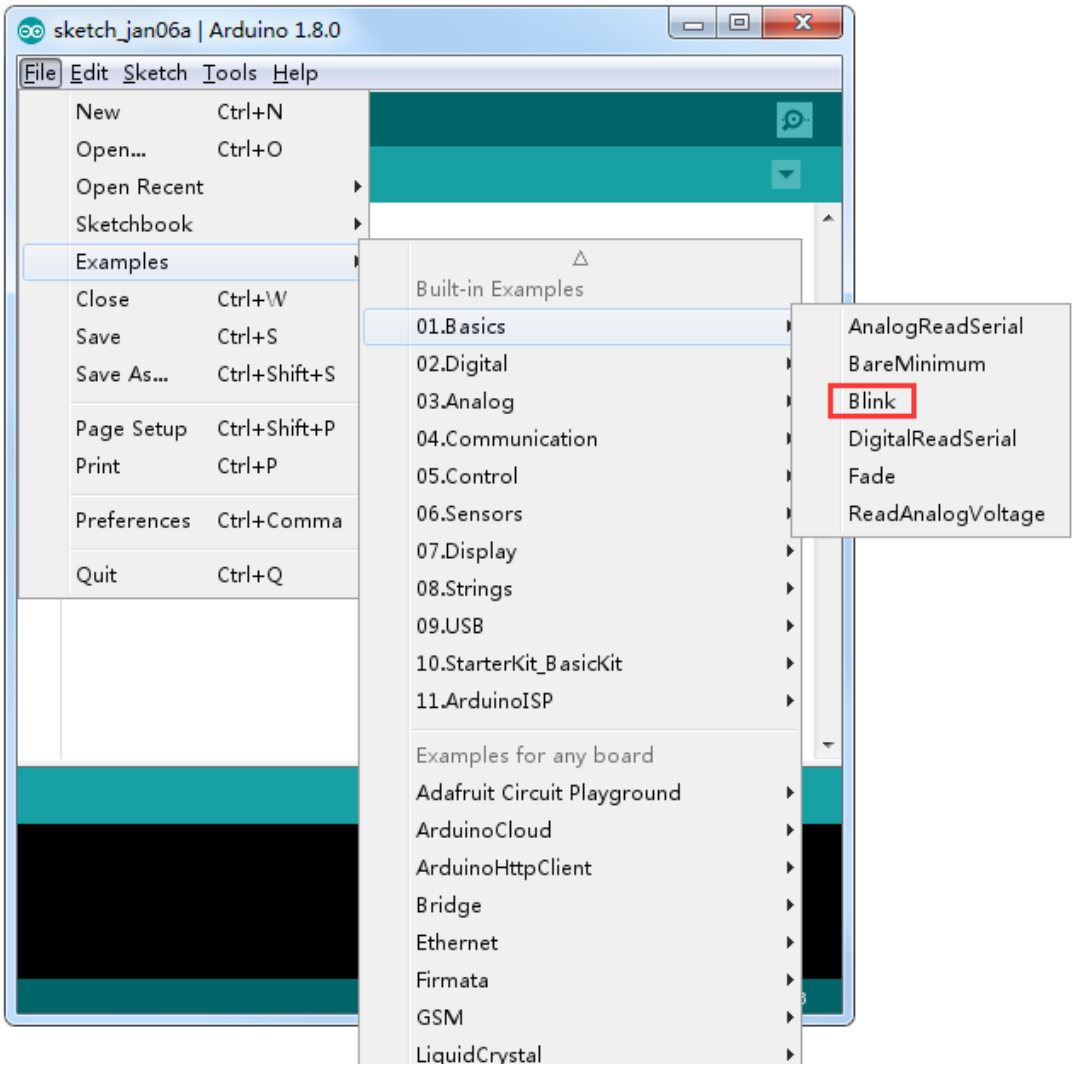
Potresti notare che la tua scheda UNO R3 abbia il led 'L' già lampeggiante quando la colleghi ad una presa USB. Questo perché le schede sono in genere fornite con il codice 'Blink' pre-installato.

In questa lezione, riprogrammerai la scheda UNO R3 con il nostro codice Blink e quindi potrai modificare la velocità con cui il LED lampeggia.

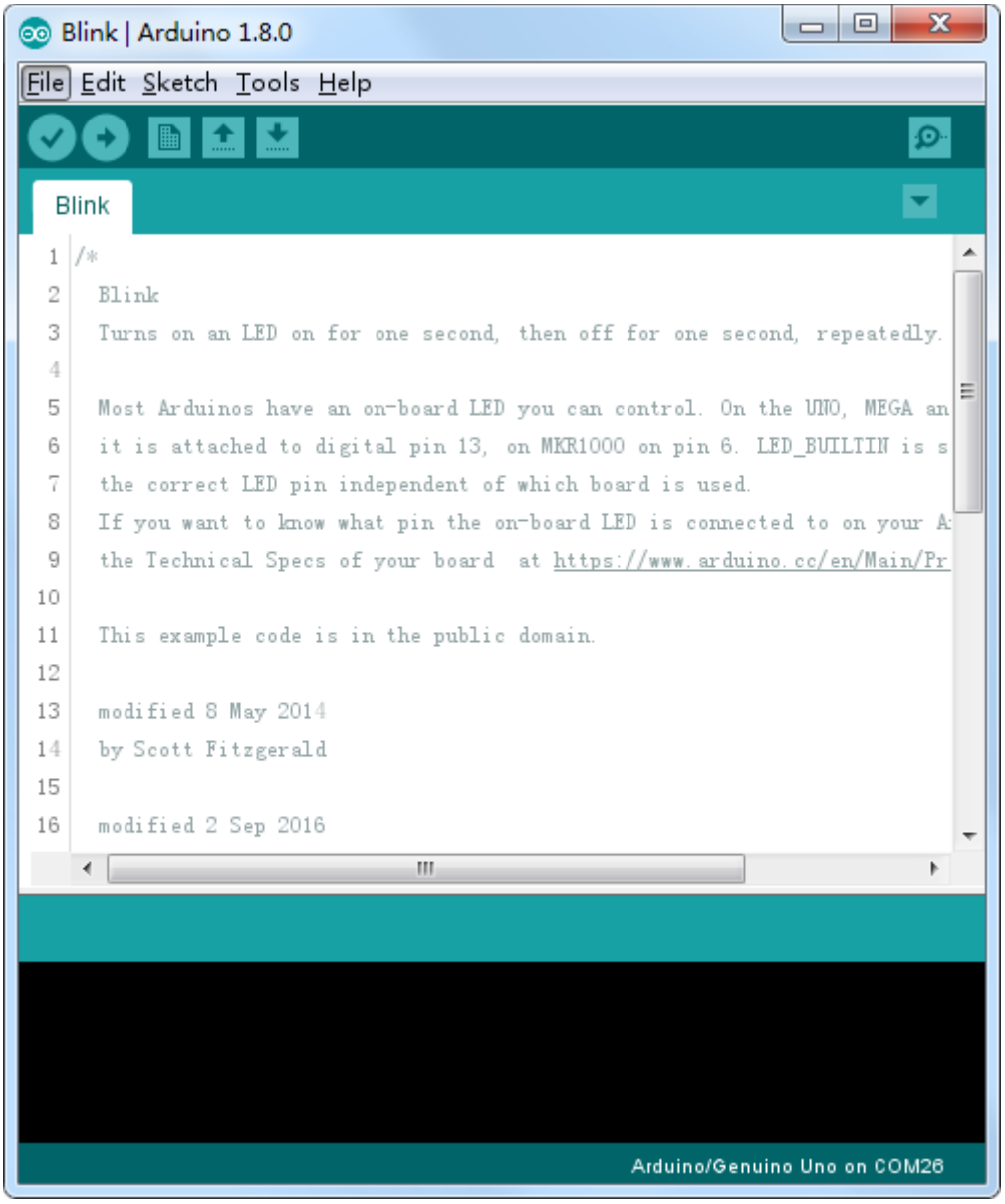
Nella lezione 0, hai impostato l'IDE di Arduino e hai fatto in modo di impostare la porta seriale giusta per poter collegare la tua scheda UNO R3. Ora è il momento di mettere alla prova il collegamento e programmare la scheda UNO R3.

L'IDE di Arduino include una vasta collezione di esempi di codici che è possibile caricare e utilizzare. Tra questi vi è un esempio che permette far lampeggiare il LED 'L'.

Carica il codice 'Blink' che trovi nel di menu dell'IDE in File> Esempi> 01.Basics



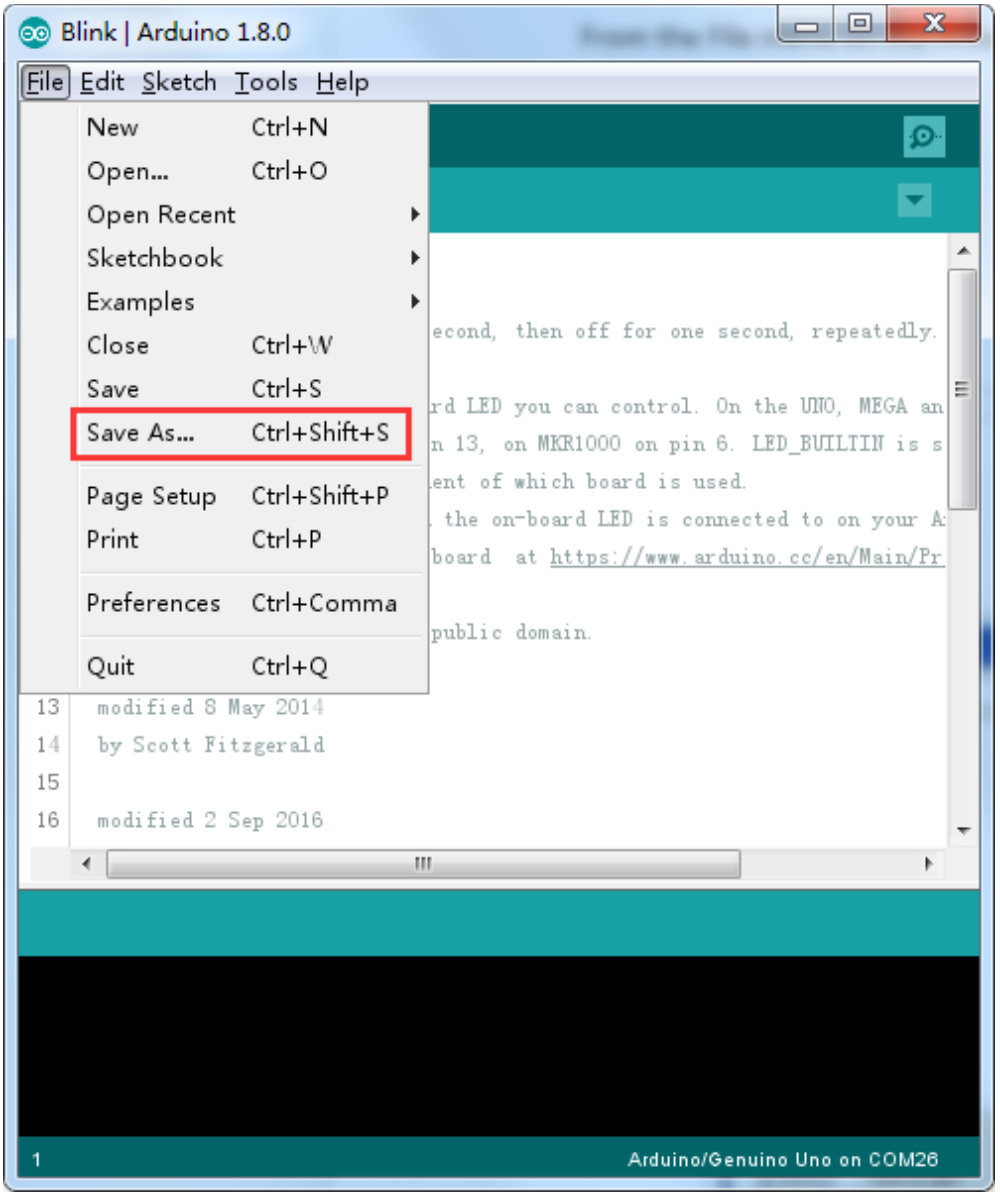
Quando si apre la finestra del codice, ingrandirla in modo che si possa vedere l'intero codice nella finestra.



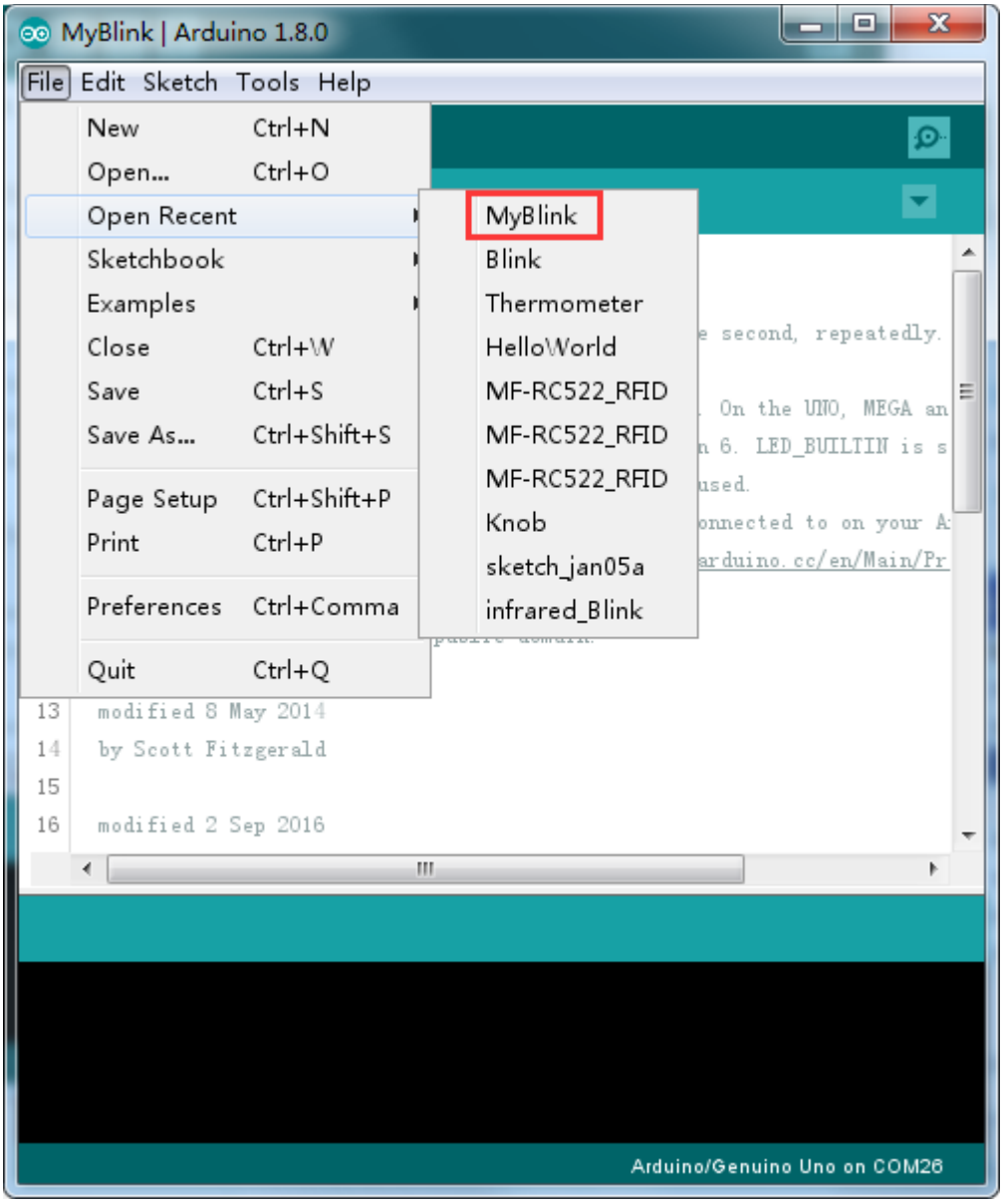
Gli esempi di codici inclusi con l'IDE di Arduino sono in 'sola lettura'. Cioè, è possibile caricarli su una scheda UNO R3, ma se li si cambia, non è possibile salvarli come lo stesso file.

Dal momento che ci accingiamo a cambiare questo codice, la prima cosa che devi fare è salvare la tua copia di codice.

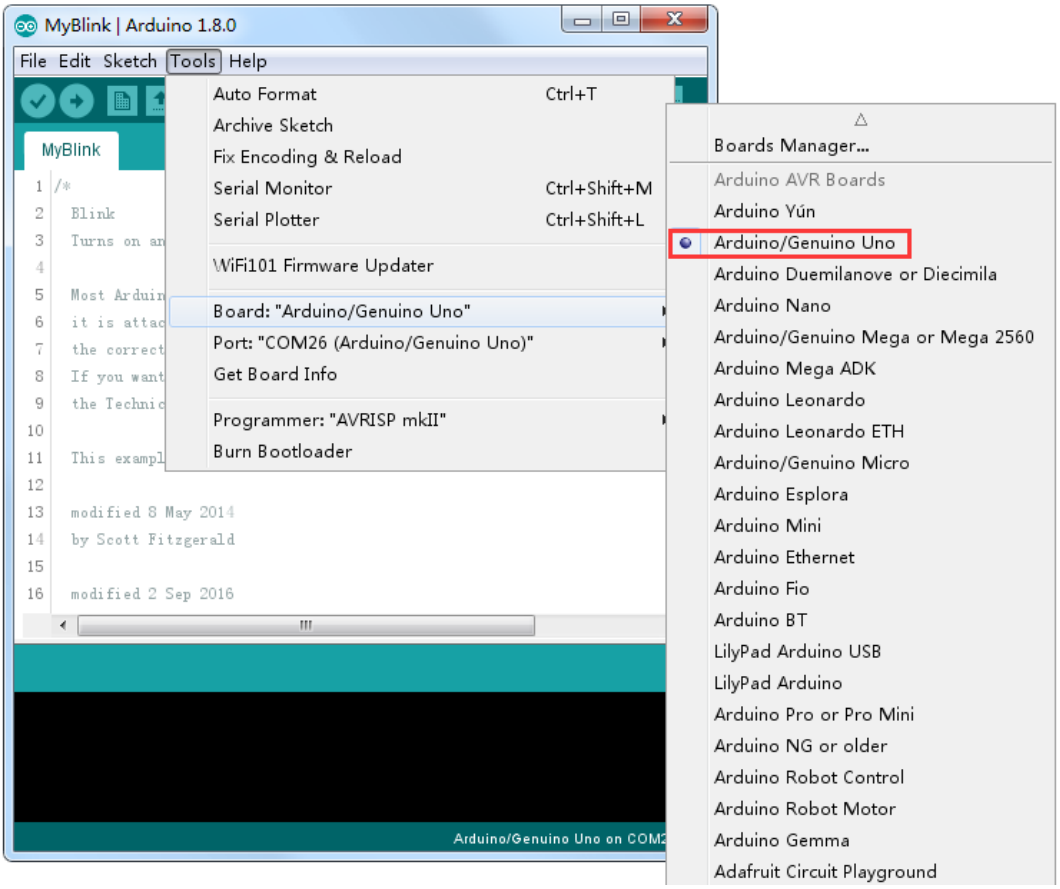
Dal menu File sul Arduino IDE, selezionare 'Salva con nome...' e quindi salvare il disegno con il nome di 'MyBlink'.

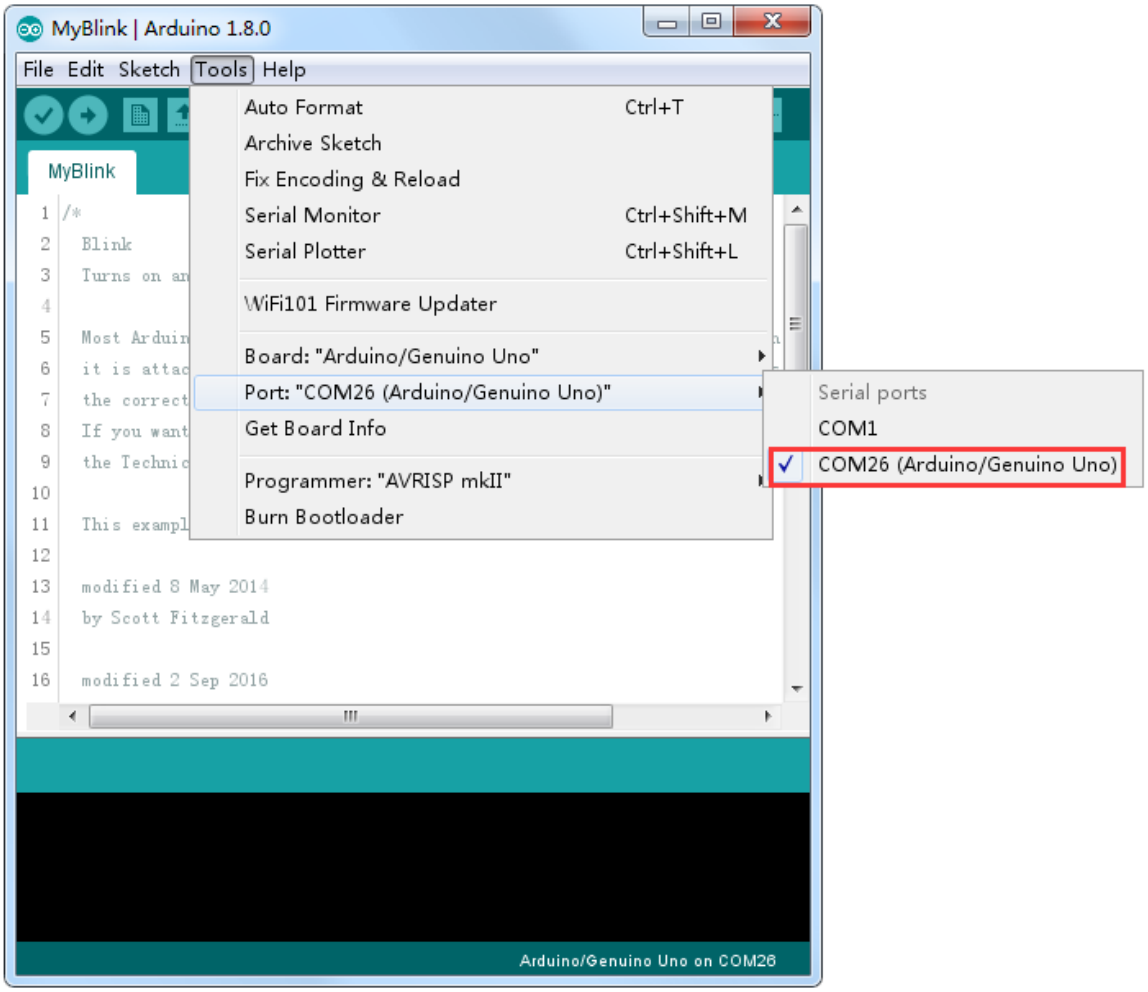

Blink | Arduino 1.8.0
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
13 modified 8 May 2014
14 by Scott Fitzgerald
15
16 modified 2 Sep 2016
1
Arduino/Genuino Uno on COM26

Ora hai salvato la tua copia di 'Blink' nella tua raccolta di codici. Questo significa che se tu volessi trovare il tuo codice potresti aprirlo cliccando su File > Open Recent.



Collega la tua scheda Arduino al computer con il cavo USB e verifica che 'Board Type' e 'Serial Port' siano impostate correttamente, come da screen seguenti.


MyBlink | Arduino 1.8.0
File Edit Sketch Tools Help
MyBlink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM26 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino/Genuino Uno on COM

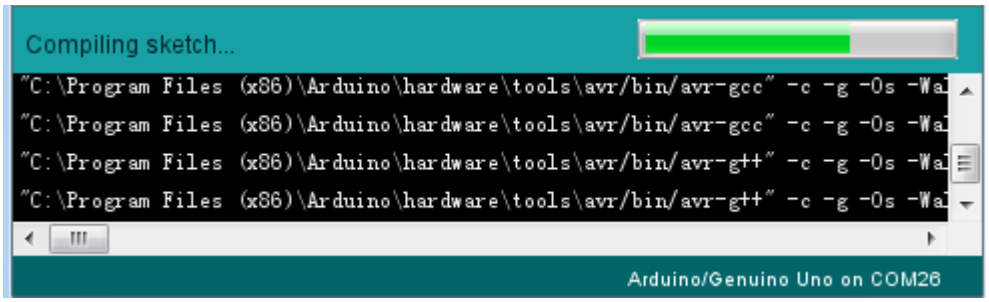

MyBlink | Arduino 1.8.0
File Edit Sketch Tools Help
MyBlink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM26 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM26 (Arduino/Genuino Uno)
Arduino/Genuino Uno on COM26

Nota: il tipo scheda e la porta seriale da utilizzare non sono necessariamente le stesse mostrate in figura. Se si utilizza UNO, allora si dovrà scegliere UNO come tipo di scheda, altre scelte possono essere fatte nello stesso modo. La porta seriale visualizzata per ognuno è diversa, nonostante sia stata scelta COM 26 nell'immagine seguente, la porta corretta sul tuo computer potrebbe essere COM3 o COM4. Una porta COM giusta dovrebbe essere COMX (Arduino XXX), che soddisfa i criteri di certificazione.

La IDE di Arduino vi mostrerà le impostazioni per scheda corrente nella parte inferiore della finestra.

Fai clic sul pulsante 'Carica'. É il secondo pulsante da sinistra nella barra degli strumenti.

Se guardi l'area di stato dell'IDE, vedrai una barra di avanzamento e una serie di messaggi. In un primo momento, vedrai 'Compilazione Sketch ...'. Questo trasforma il codice in un formato adatto per il caricamento sulla scheda.



Successivamente, lo stato passerà a 'Caricamento'. A questo punto, i LED su Arduino dovrebbero iniziare a lampeggiare mentre il codice viene trasferito sulla scheda.

Infine, lo stato passerà a 'Fatto'.

Il messaggio seguente ci dice che il disegno sta usando 928 byte dei 32,256 byte disponibili. Dopo la 'Compiling Sketch..'(compilazione del codice) potresti ottenere il seguente messaggio di errore:

Può significare che la scheda non sia collegato affatto oppure non siano stati installati i driver (se necessari) o che sia stata selezionata la porta seriale sbagliata. Se si verifica questo problema, tornare alla Lezione 0 e verificare l'installazione. Attendere che il caricamento sia completato, la scheda Arduino dovrebbe riavviarsi e iniziano a lampeggiare.

Apri il codice

Si noti che una parte enorme di questo codice è composto da commenti. Queste non sono istruzioni del programma; Servono semplicemente spiegare come funziona il programma. Sono lì per aiutare chi legge il codice. Tutto ciò che si trova tra / * e * / nella parte superiore del disegno è un blocco di commento; spiega il a cosa serve questo codice.

I commenti su una singola riga iniziano con // e tutto ciò che è presente fino fine di quella linea è considerato un commento.

La prima linea di codice è:

```
//define led pin
int led = 13;
```

Come il commento di cui sopra spiega, questo comando definisce un nome per il pin a cui è collegato il LED. Sulla maggior parte degli Arduino, tra cui UNO e Leonardo, è il pin 13. Successivamente, abbiamo la funzione di 'setup'. Di nuovo, come spiega il commento, questa funzione viene eseguita quando viene premuto il pulsante di reset sulla scheda. Viene eseguito anche ogni volta che avviene il ripristino della scheda, per qualsiasi motivo, come per esempio quando viene connesso ad una fonte di alimentazione, oppure dopo un nuovo codice viene caricato.

```
void setup() {
// initialize the digital pin as an output.
pinMode(led, OUTPUT);
}
```

Ogni codice di Arduino deve avere una funzione di 'setup', e la zona corretta dove si aggiungono le istruzioni da eseguire durane il setup è proprio tra le due parentesi graffe: {...}.In questo caso, vi è un solo comando tra le parentesi, che, come il commento spiega, imposta sulla scheda Arduino il pin LED come uscita. É anche obbligatorio per il codice di avere una funzione di 'loop'. A differenza della funzione di 'setup' che viene eseguito solo una volta, dopo un reset, la funzione di 'loop' sarà eseguita dopo aver terminato l'esecuzione del setup, ed una volta terminata andrà in loop, cioè verrà ripetuta all'infinito, se non definiamo ulteriori comandi.

```
void loop() {
   digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
   delay(1000);                        // wait for a second
   digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
   delay(1000);                        // wait for a second
}
```

All'interno della funzione di loop, il primo comando accende il LED impostando il relativo pin su HIGH, poi interviene 'ritardo' di 1000 millisecondi (1 secondo), in cui

la scheda semplicemente attende, poi il LED viene spento, impostando il pin a LOW e viene richiamata una ulteriore attesa di 1 secondo. Se vuoi far lampeggiare il LED più velocemente, come puoi immaginare, la chiave sta nel cambiare il parametro di ritardo, tra le parentesi tonde () della funzione di delay.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

Questo tempo di ritardo è espresso in millisecondi, quindi se desideri che il led lampeggi due volte più velocemente, ti basterà modificare il valore da 1000 a 500. Questo imposterà una pausa di mezzo secondo, invece che di un secondo. Caricando il nuovo codice e dovresti vedere il LED lampeggiare più rapidamente.

Lezione 3 LED

Introduzione

In questa lezione imparerai a cambiare la luminosità di un led usando differenti valori di resistenza.

Componenti Richiesti:

(1) x Elegoo UNO R3

(1) x LED rosso da 5mm

(1) x Resistenza da 220 ohm

(1) x Resistenza da 1k ohm

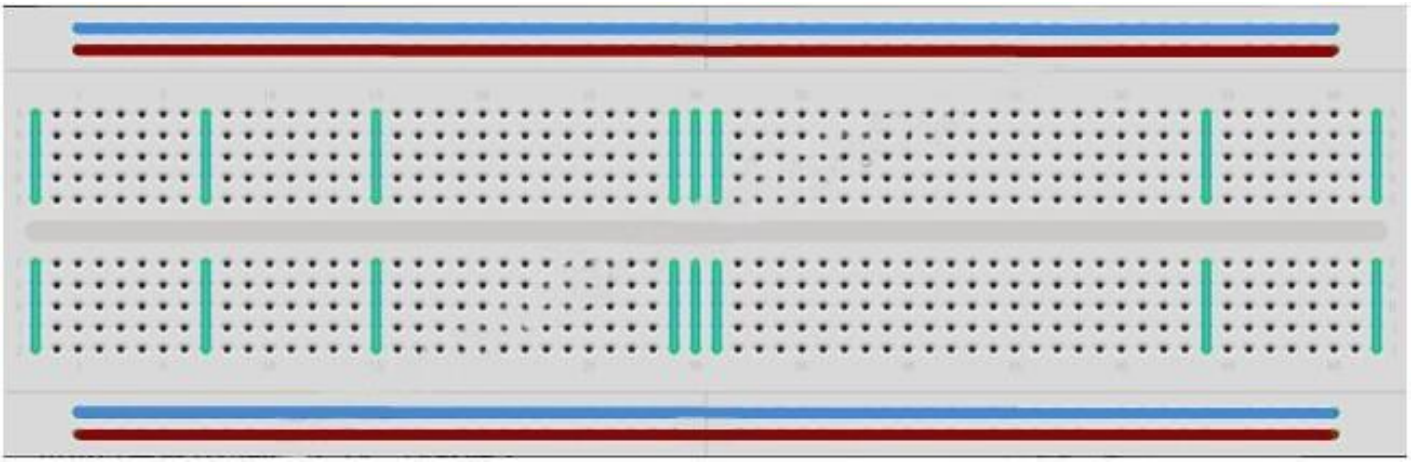
(1) x Resistenza da 10k

(2) x M-M connettori (Connettori maschio-maschio)

Introduzione ai componenti

BREADBOARD MB-102：

La breadboard ti permette di costruire dei prototipi di circuiti velocemente, senza necessità di saldare le connessioni, sotto un esempio.



Esistono breadboard di diverse dimensioni e configurazioni. Il tipo più semplice è costituito da una griglia di buchi in un blocco ci plastica. All'interno sono presenti strisce di metallo che mettono in connessione differenti fori su tutte righe corte(quelle verdine nell'immagine). Inserendo i piedini di diversi componenti nella stessa riga, si avrà un collegamento elettrico diretto tra di essi.

Un canale più profondo è visibile al centro della breadboard, indica che li non sono presenti connessioni, perciò si potrà inserire un chip con i piedini metà nella parte superiore e metà in quella inferiore della breadboard senza che facciano cortocircuito (quindi senza che si colleghi un piedino con l'altro).

Alcune breadboard hanno due strisce di buchi che si sviluppano lungo i bordi e che sono separate dalla griglia centrale. Esse hanno strisce di collegamento per tutta la lunghezza e sono solitamente usate per i per i voltaggi comuni. Essi sono comunemente usati per +5 volt e per la messa a terra (GND). Queste strisce sono dei binari che ti permettono di dare energia a più componenti o punti della breadboard.

Le breadboard sono molto comode per la prototipazione, ma hanno alcune limitazioni, per esempio le connessioni sono temporanee e non sono affidabili come le connessioni saldate, se ti capita di avere problemi di intermittenza con i circuiti potrebbe essere per colpa delle connessioni sulla breadboard.

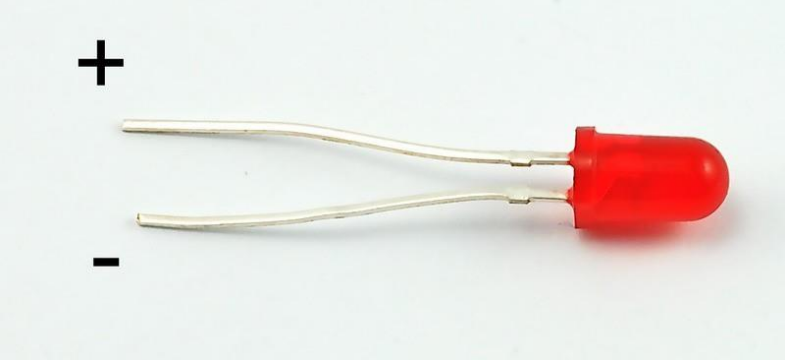
LED:

I LED sono ottimi indicatori luminosi, essi utilizzano una quantità molto esigua di elettricità ed hanno una vita quasi infinita.

In questa lezione utilizzerai uno dei led più comuni tra quelli esistenti. Il led da 5mm. 5mm si riferisce al diametro del LED. Altre dimensioni comuni sono la 33mm e la 10mm.

Non puoi collegare direttamente il LED alla batteria o all'erogatore ti voltaggio, in quanto:

1) Il LED ha un lato positivo ed uno negativo, e non si accenderà se piazzato nel senso sbagliato.
2) Il LED va utilizzato con una resistenza per limitare la quantità di corrente che fluisce attraverso di esso, altrimenti si brucia!



Se non utilizzi una resistenza con il led potrebbe distruggersi immediatamente avendo troppa corrente che fluisce in esso, la corrente surriscalderà e distruggerà la giunzione interna che produce la luce.

Ci sono due modi per sapere qual è il lato positivo e quello negativo nel led.

In primo luogo il piedino più lungo è quello positivo.

In alternativa, dove il piedino negativo entra nel corpo del led c'è un piano metallico di dimensioni maggiori rispetto al positivo.

Se ti capita di avere un led con il lato interno piano più grande in corrispondenza del piedino più lungo devi assumere che il piedino più lungo corrisponde al positivo.

RESISTENZE:

Come suggerisce il nome, le resistenze oppongono resistenza al flusso di elettricità. Più p alto il valore della resistenza, più resiste, e meno corrente elettrica fluirà attraverso di esso.

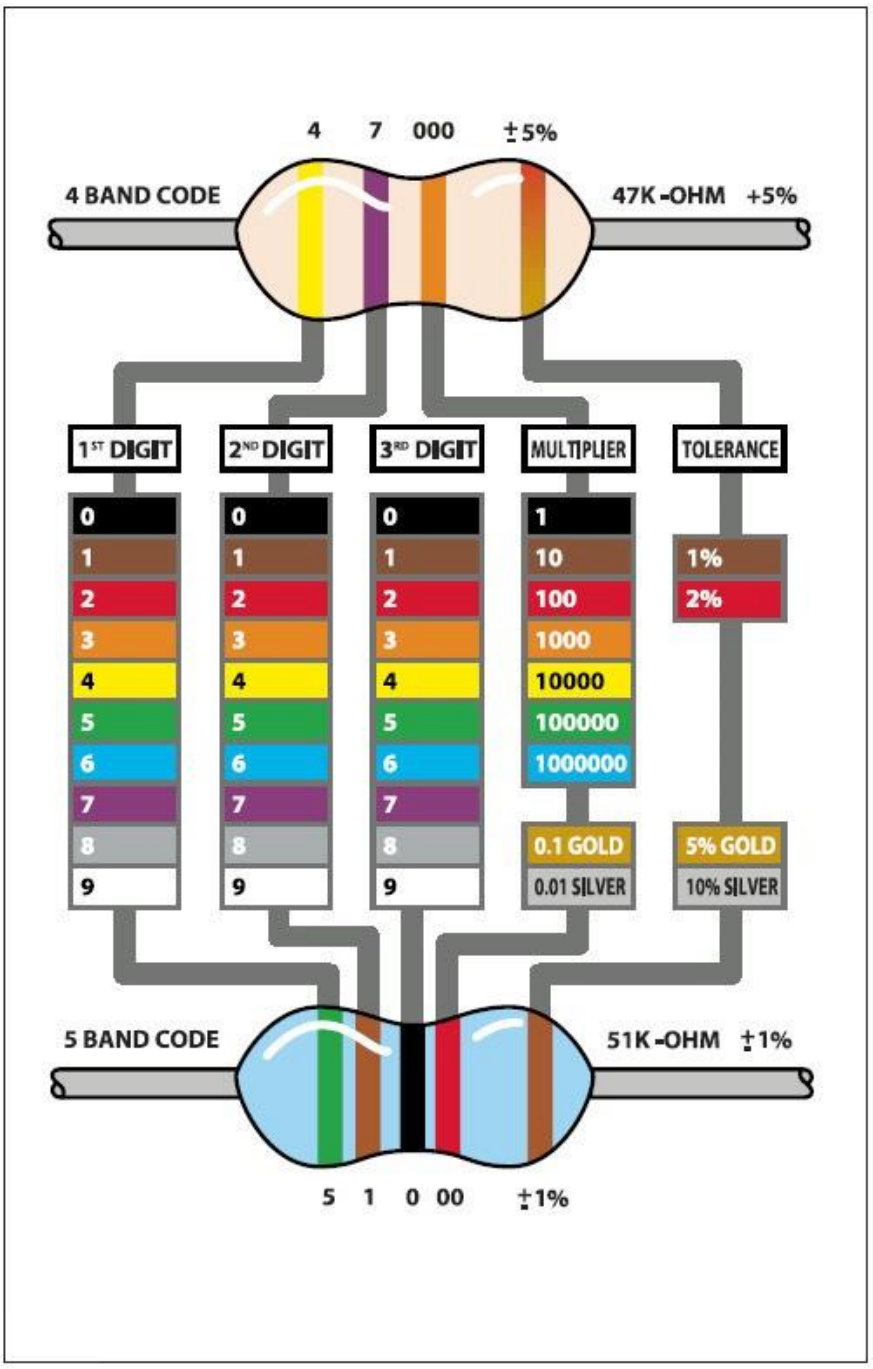
Noi utilizzeremo i resistori per controllare quanta elettricità fluisce attraverso il led e di conseguenza controllare quanta luce produrrà il LED.

Ma prima diciamo qualcosa di più sulle resistenze…

L'unità di resistenza è chiamata Ohm, che solitamente è abbreviata con la lettera greca Ω Omega. Dato che un Ohm è un valore di resistenza molto masso (non oppone quasi nessuna resistenza), si accompagna spesso il valore di resistenza con kΩ (1,000 Ω) e MΩ (1,000,000 Ω). I due valori si leggono come kilo-ohm e mega-ohm.

In questa lezione utilizzeremo tre diversi tipi di resistenza tutte con valori diversi: 220Ω, 1kΩ and 10kΩ. Queste resistenze sembrano tutte uguali, eccetto per le strisce colorate su di esse. Queste strisce indicano il valore della resistenza.

Il codice dei colori delle resistenze ha tre strisce colorate ed una ora ad una estremità.



A differenza dei LED, le resistenze non hanno un capo negativo ed uno positivo, e possono essere collegati indifferentemente in un verso o nell'altro.

Se trovi questo tipo di approccio troppo complicato, puoi semplicemente leggere il flag sui nostri pacchi di resistenze per determinarne il valore di resistenza!

Oppure puoi usare un multimetro digitare.

Connessione

Schema

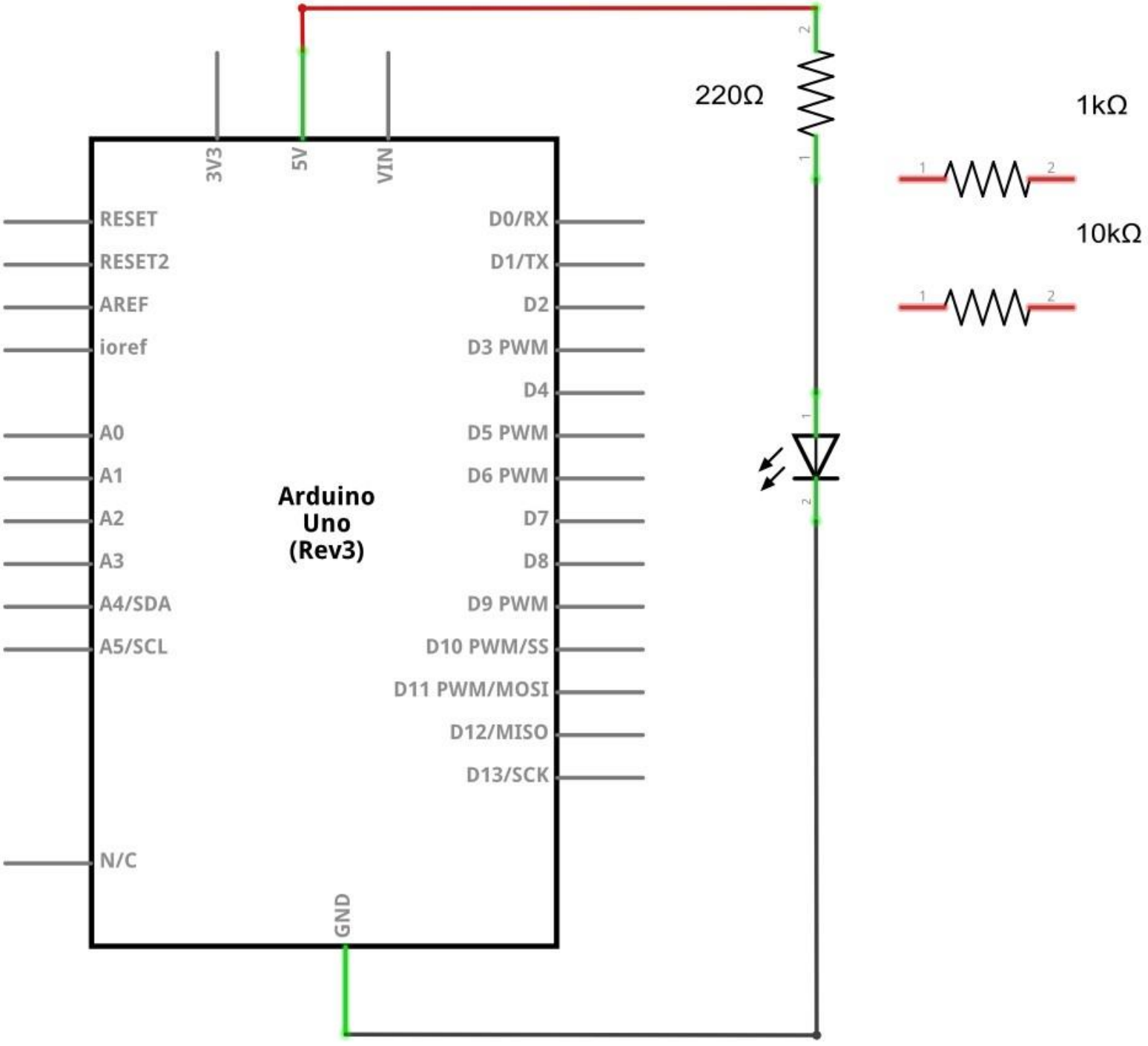
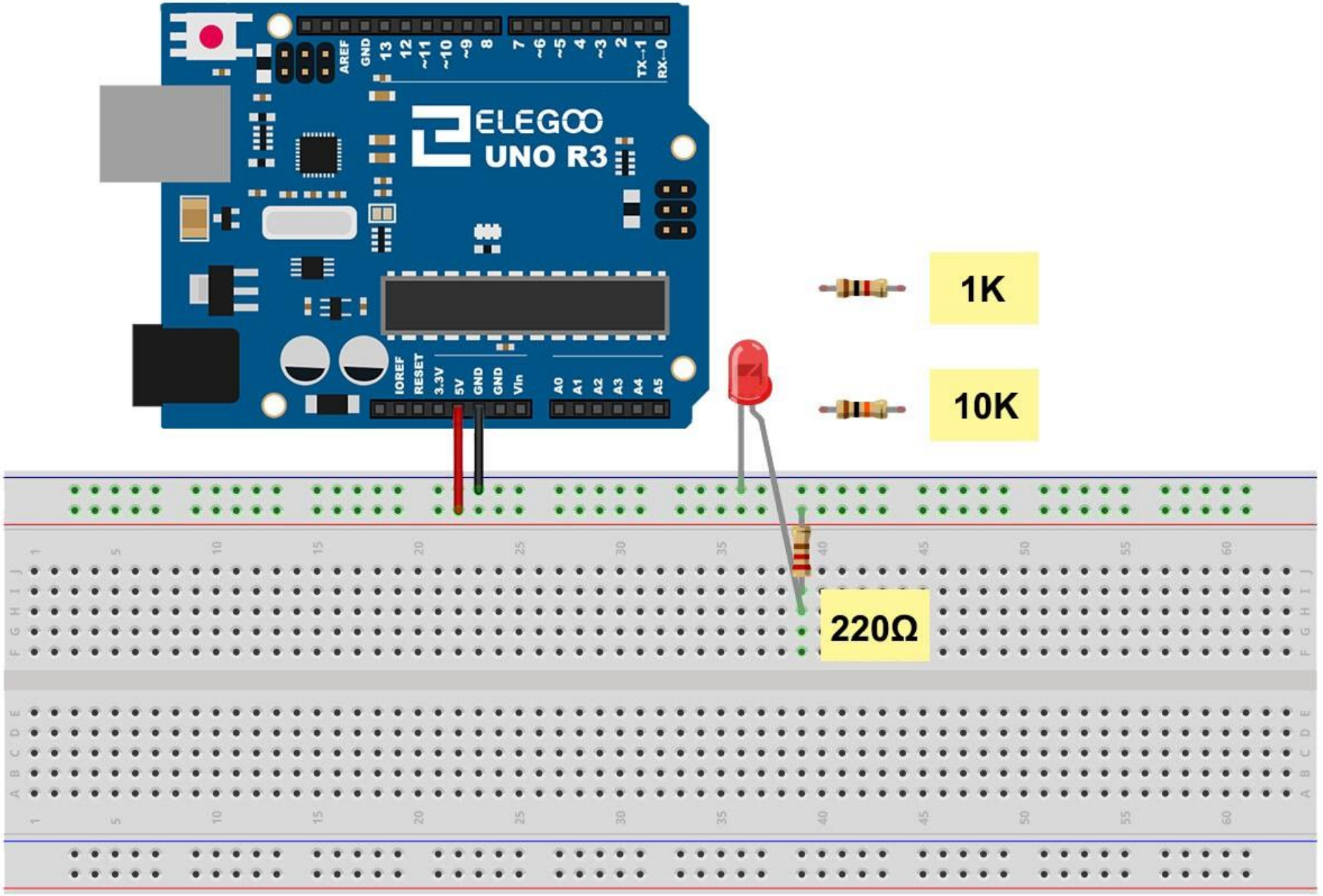


Diagramma di Collegamento



La scheda UNO è un ottimo generatore di 5 volt, che noi utilizzeremo per dare energia a led e resistenze. Non necessiti di fare niente oltre al collegare la tua scheda UNO con il cavo usb.

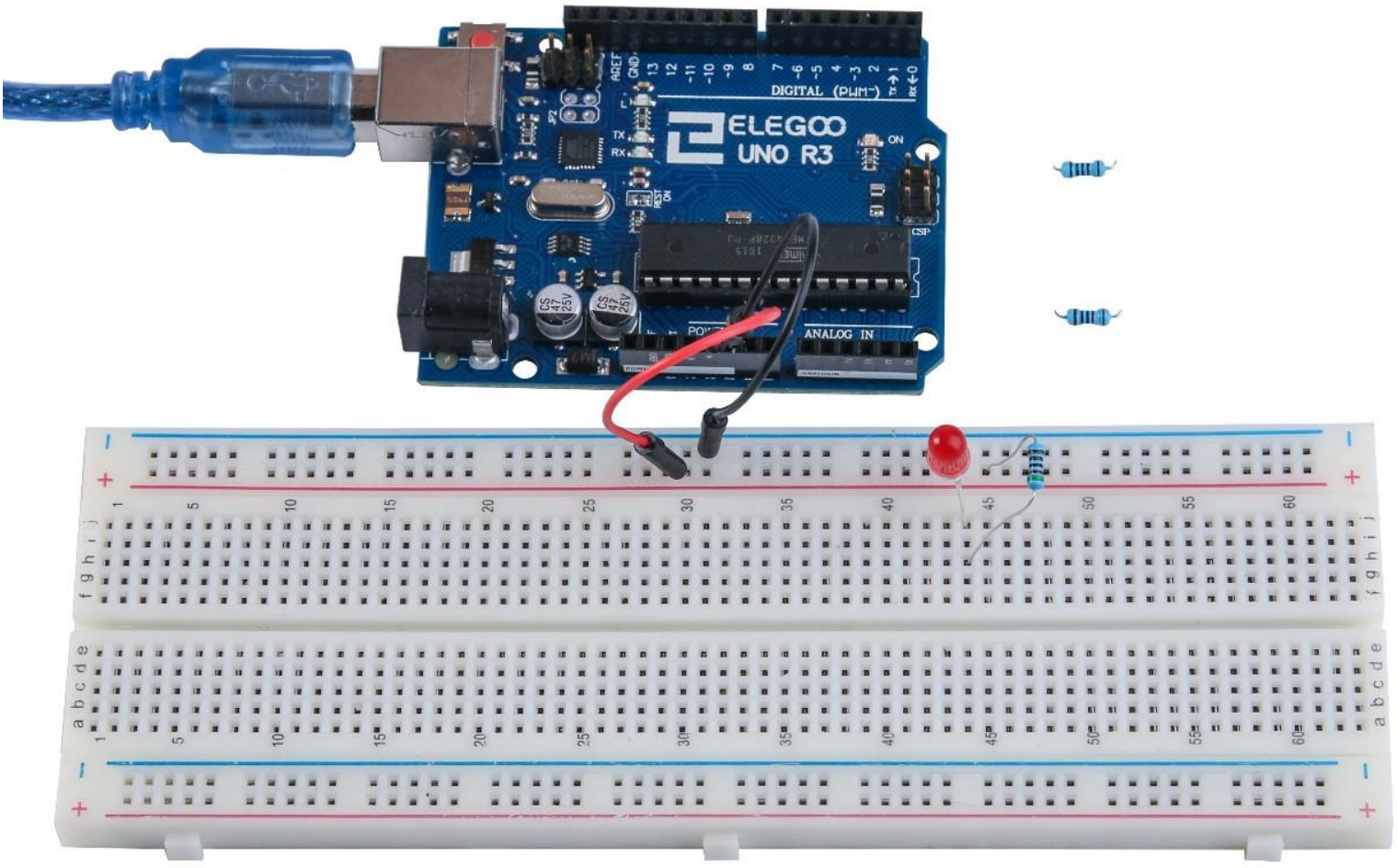
Con collegata la resistenza da 220 Ω, il LED ti apparirà molto luminoso. Se cambi la resistenza e invece della 220 Ω utilizzi una resistenza da 1k Ω il led apparirà leggermente meno luminoso. Infine collegando la resistenza da 10k Ω, il led sarà molto poco luminoso, quasi spento.

Utilizza il collegamento con del cavo rosso, togliendolo dalla breadboard e rimettendolo al suo posto, funzionerà da bottone di accensione, e dovresti riuscire a notare la differenza tra acceso e spento.

Al momento hai una tensione di 5V sul primo piedino della resistenza, il secondo piedino della resistenza è collegato ad un capo del led e poi all'altro capo del LED è collegato con la messa a terra GND. Se spostiamo la resistenza da prima a dopo il LED come indicato nell'immagine di sotto, il LED continuerà a rimanere acceso.

Vorrai probabilmente rimettere la resistenza al suo posto, non importa in quale lato del LED viene posizionata la resistenza, l'importante è che sia posizionata da qualche parte!

Foto d'esempio



Lezione 4 LED RGB

Introduzione:

I LED RGB sono un modo divertente di aggiungere un po' di colore ai tuoi progetti… Dato che sono come 3 led regolari in uno, il loro utilizzo e la loro connessione non è molto differente rispetto a quelli già visti.

Ne esistono 2 versioni, la prima, con Anodo in comune e la seconda con Catodo in comune.

La versione con l'Anodo in comune utilizza i 5V sul pin comune mentre la seconda lo vuole connesso alla messa a terra GND.

Come per ogni led necessitiamo di connettere qualche resistenza in linea (3 totali) in modo da limitare la corrente che lo attraversa.

Nel nostro codice accenderemo il led nello stato di colore rosso, poi si dissolverà nel colore verde, in seguito si dissolverà nel colore Blu che tornerà rosso.

Facendo così cicleremo su tutti i colori che possono essere ottenuti.

Componenti Richiesti:

(1) x Elegoo UNO R3

(1) x Breadboard con 830 punti di aggancio

(4) x Connettori M-M (Connettori Maschio-Maschio)

(1) x LED RGB

(3) x Resistenza da 220 ohm

Introduzione ai componenti

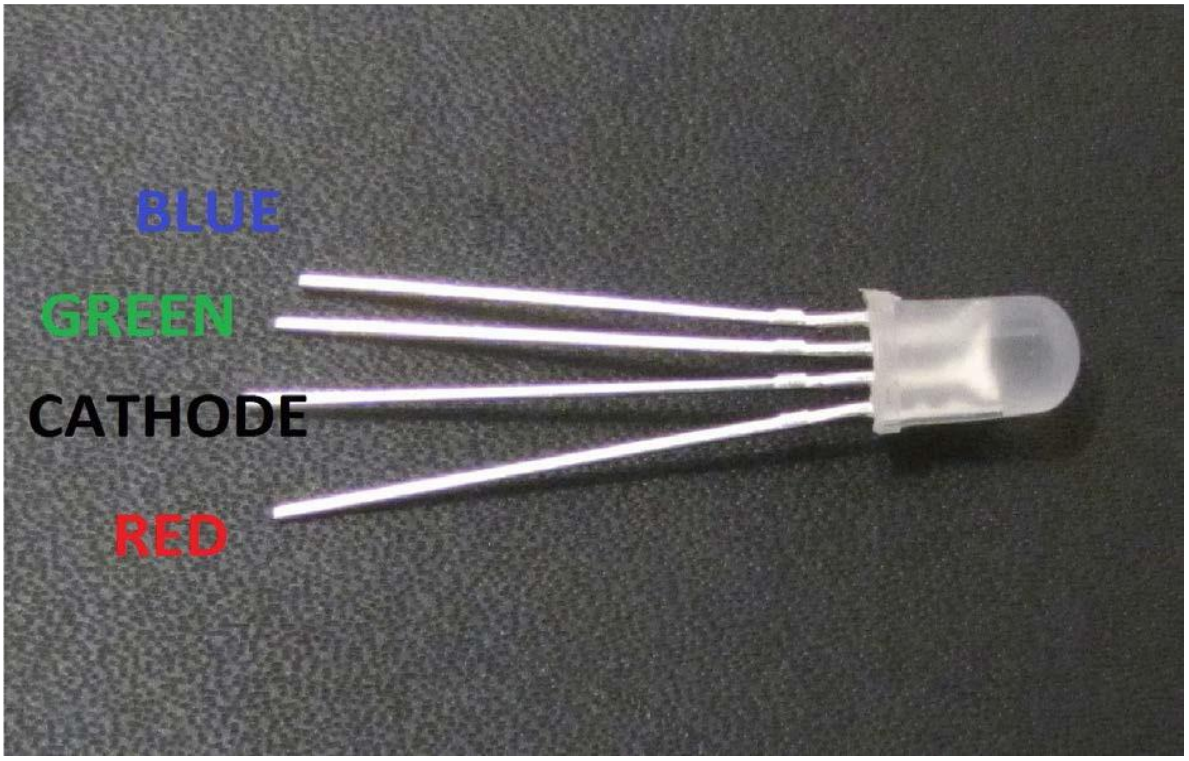
LED RGB:

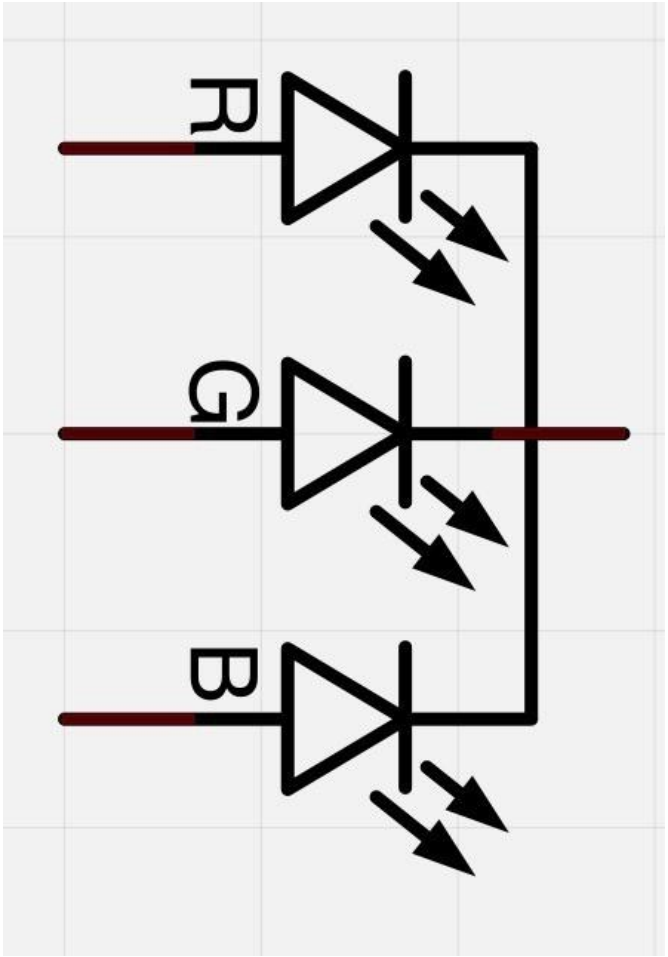
A prima vista i LED RGB (Red, Green, Blue) sembrano dei normali led. Tuttavia all'interno del solito corpo del led ci sono3 ledi differenti, uno Rosso, uno Verde e ovviamente uno Blu. Controllando la luminosità di ogni led individualmente puoi controllare e mischiare praticamente qualsiasi colore tu voglia.

I colori si mischiano nello stesso modo in cui si mischiano i colori in una tavolozza – aggiustando la luminosità in ognuno dei tre LED. La parte più difficile di tutto questo è utilizzare differenti valori di resistenza (o resistenze variabili) come abbiamo fatto nella lezione 2, ma questo comporta un sacco di lavoro!

Fortunatamente per noi la scheda UNO ha una funzione analogWrite che puoi utilizzare sui pin marcati con il simbolo ~, la quale permette di fornire una quantità variabile di potenza per ogni singola uscita, e quindi per ogni singolo LED.

Il LED RGB ha quattro piedini. Tre di essi sono l'ingresso positivo dei tre led presenti nel corpo, un capo per ogni led, il quarto è un l'ingresso negativo in comune per tutti e 3 i led.





Qui in fotografia puoi vedere 4 Elettrodi LED. Un pin separato per ognuno dei colori, verde, blu o rosso, esso è chiamato Anodo. Dovrai sempre collegare il “+” ad esso. Il catodo va collegato al “-“ (GND, messa a terra), se colleghi i piedini in un modo differente il led non si accenderà.

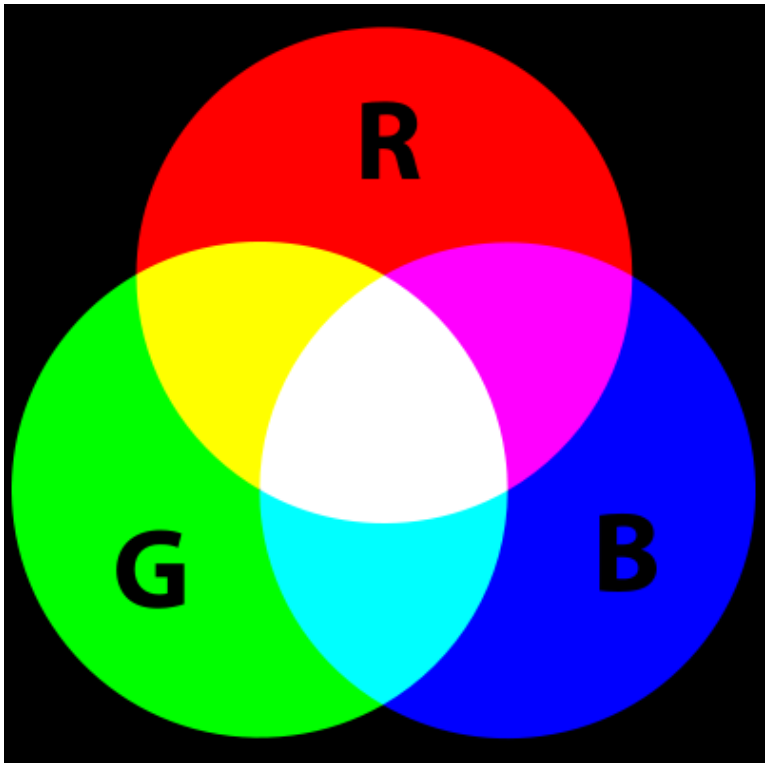
L’ingresso negativo in comune per i tre led è il secondo pin dal lato con il piatto. Esso è anche il piedino più lungo dei 4 e verrà connesso alla messa a terra.

Ogni led interno al corpo necessita una resistenza da 220Ω per prevenire che troppa corrente fluisca attraverso di esso. I tre capi dei LED (uno rosso, uno verde, ed uno blu) vanno connessi ai pin di output della scheda UNO utilizzando le resistenze da 220Ω.

COLORI:

Il poter creare ogni tipo di colore mischiando e variando la quantità di Rosso, Verde e Blu è dovuto al fatto che i tuoi occhi hanno tre tipi di ricettori per la luce in essi (verde, rosso e blue). I tuoi occhi e il tuo cervello processano la quantità di rosso, verde e blu, e la convertono in uno spettro di colori.

In questo modo utilizzando 3 led molto vicini, stiamo semplicemente imbrogliando gli occhi, facendogli credere che sia un colore solo invece che tre. La stessa idea è utilizzata in TV dove gli schermi LCD hanno puntini rossi, verdi e blu molto vicini uno all'altro formando ogni trio forma un pixel.



Se impostiamo la stessa luminosità per tutti e tre i led, il colore finale che si svilupperà sarà il bianco. Se spegniamo il led Blue, quindi tenendo acceso solo rosso e verde alla stessa luminosità, la luce apparirà gialla.

Possiamo controllare la luminosità di ognuno dei tre colori Rosso, verde e blu separatamente, mischiandoli e creando qualsiasi colore vogliamo.

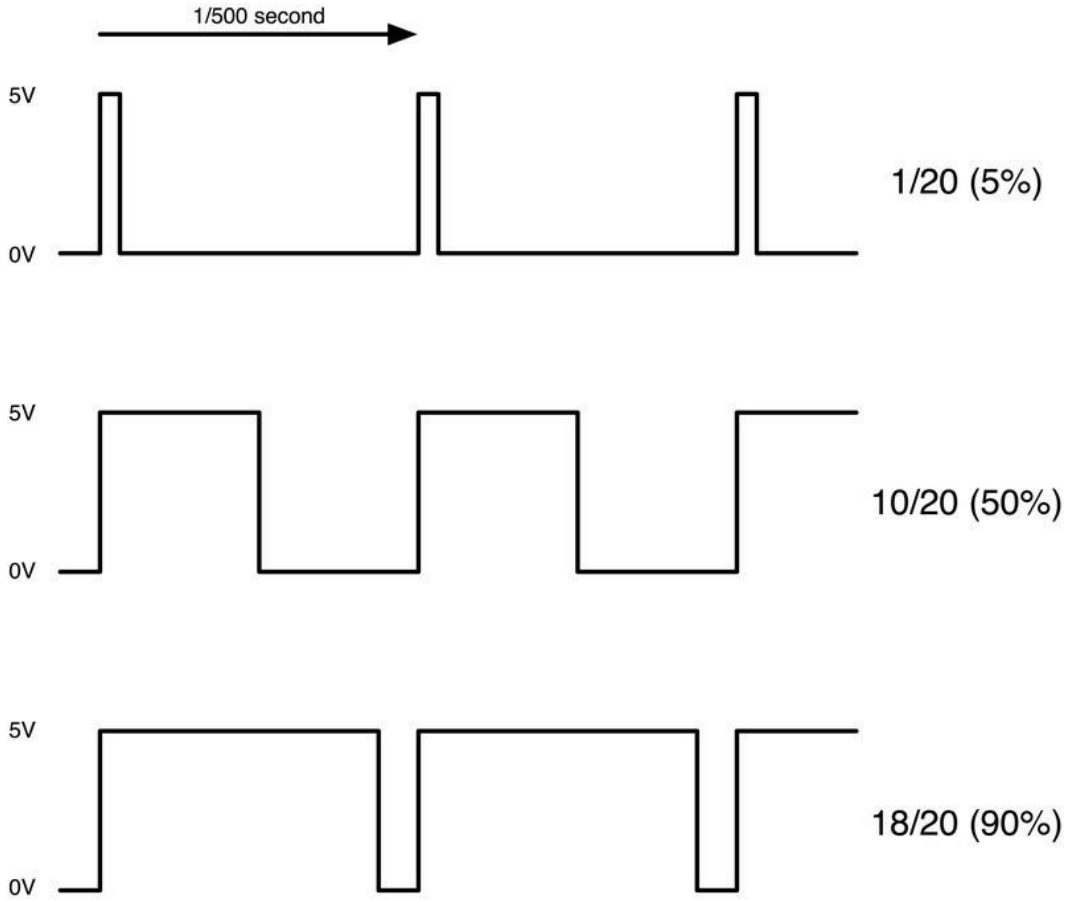
Il nero non è considerato un colore, ma semplicemente l'assenza di luce. Infatti, l'unico modo pe avvicinarci a quel colore e semplicemente spegnendo tutti e tre i colori.

Teoria (PWM)

Pulse Width Modulation (PWM) è una tecnica per controllare la potenza.

Si usa anche qui per controllare la luminosità di ogni singolo LED.

Il diagramma qui sotto mostra il segnale da uno dei pin PWM della scheda UNO.



Approssimativamente ogni 1/500 di secondo l'uscita PWM produce una pulsazione. La lunghezza di tale pulsazione è controllata dalla funzione analogWrite. Perciò analogWrite(0) non produrrà alcuna pulsazione, mentre analogWrite(255) produrrà una pulsazione che durerà fino all'inizio della prossima pulsazione, perciò l'uscita sarà allo stato di ON per tutto il periodo.

Se specifichiamo un valore nella funziona analogWrite che è compresa tra 0 e 255, verrà prodotto a una pulsazione. Se l'uscita è alta, per esempio, solo per il 5% del tempo, qualsiasi cosa ci sarà connessa a tale uscita riceverà il 5% della potenza disponibile.

Se l'output è a 5V per il 90% del tempo, il carico riceverà il 90% della potenza emessa.

Non possiamo vedere il LED accendersi e spegnersi a tale velocità, perciò, al nostro occhio sembrerà semplicemente un cambio di luminosità.

Connessioni

Schema

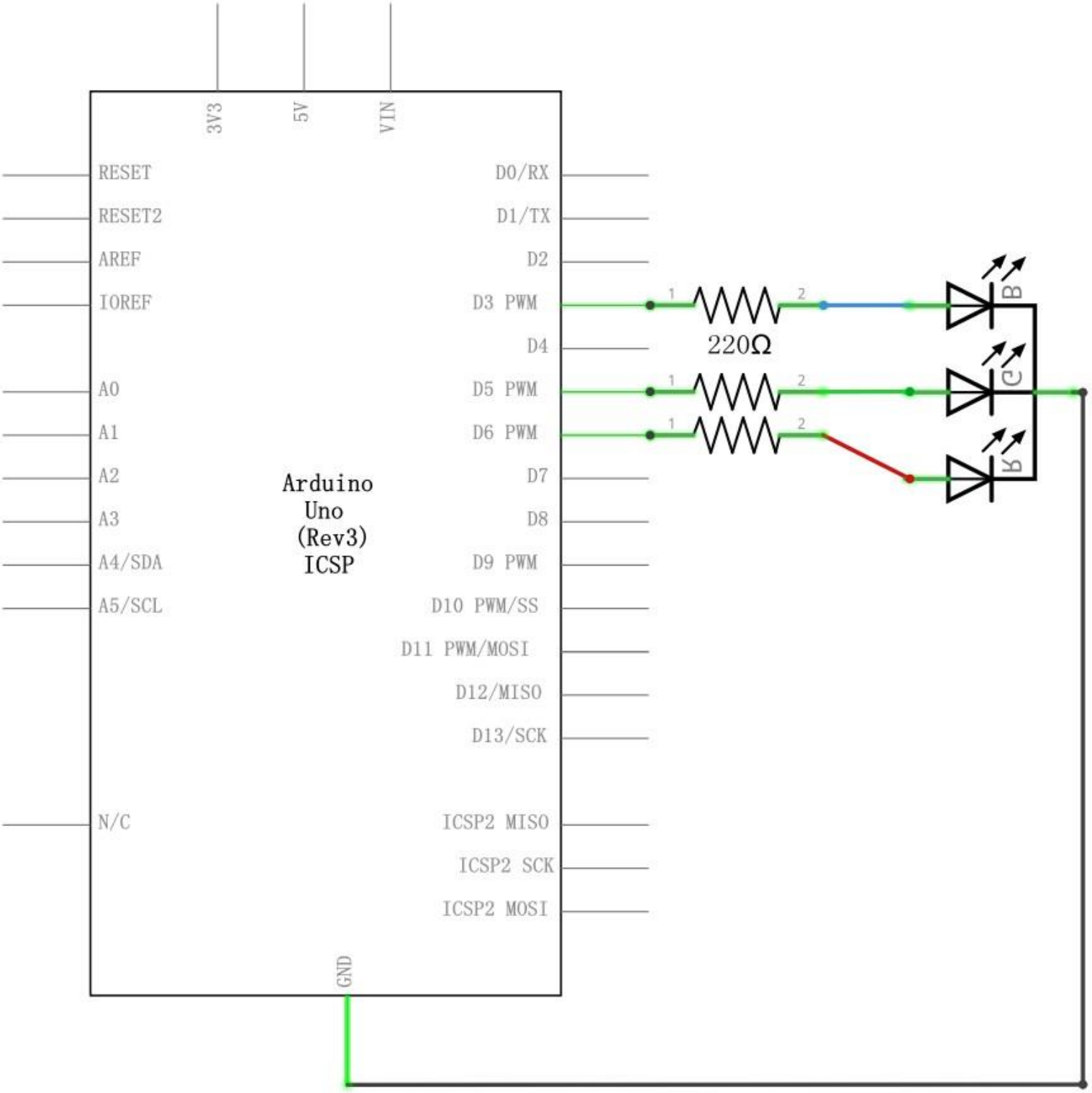
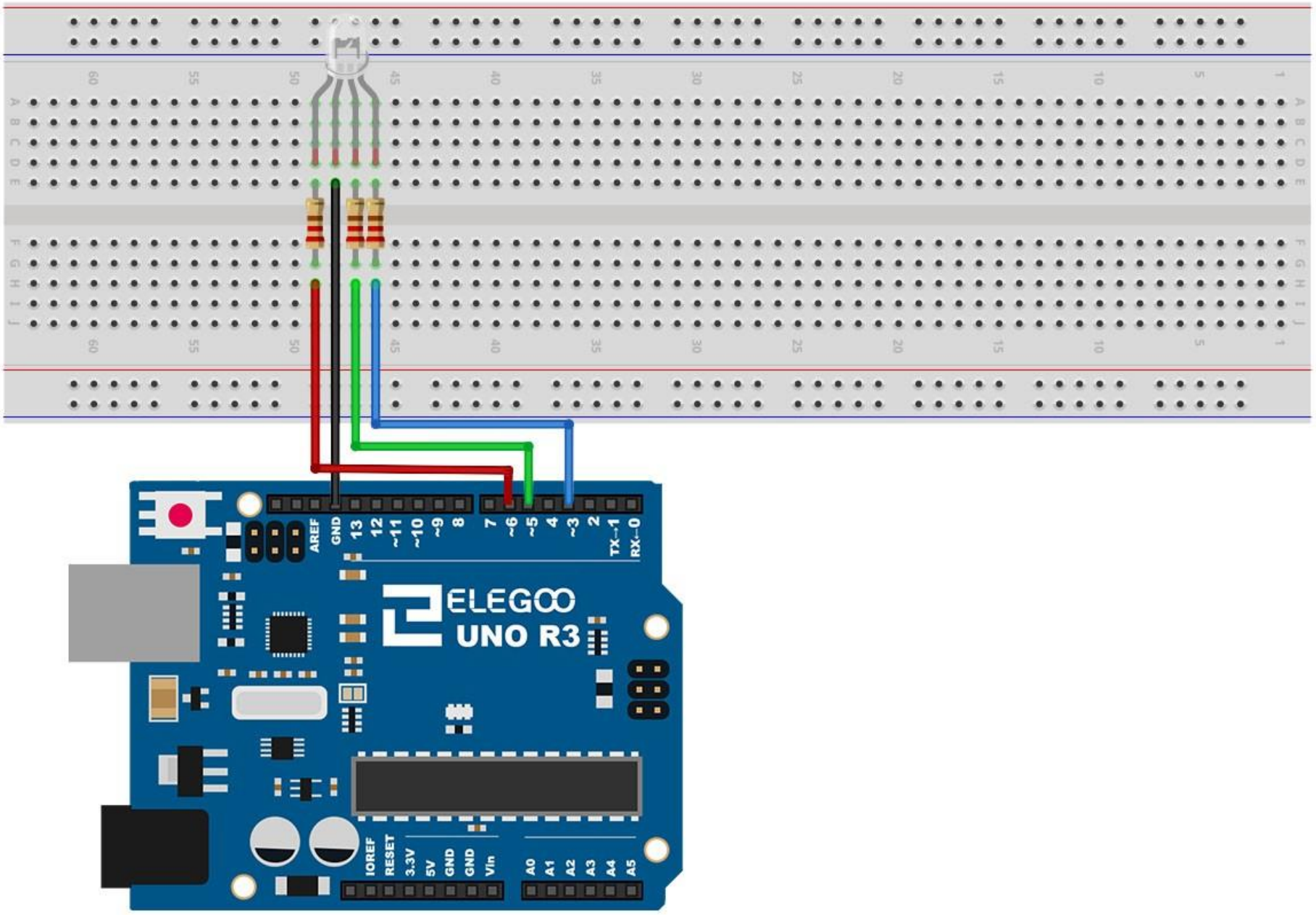


Diagramma di collegamento



Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 4 RGB LED”, clicca su UPLOAD per caricare il programma.
Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

Il nostro codice utilizza il ciclo di FOR per ciclare tra i vari colori.
Il primo ciclo di FOR sposterà il colore tra ROSSO e VERDE.
Il secondo ciclo di FOR sposterà il colore tra VERDE e BLUE.
L’ultimo ciclo di FOR sposterà il colore tra BLUE e ROSSO.
Prova il codice e in seguito te lo spiegheremo nei dettagli.

Il codice inizia specificando quali pin verranno usati per quali colori:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

Il prossimo passo è quello di scrivere la funzione di setup. Come abbiamo imparato nelle lezioni precedenti, la funzione di setup viene eseguita una sola volta dopo che viene resettata la scheda di Arduino. In questo caso, tutto quello che fa la funzione è definire che i tre pin che andremo ad utilizzare saranno degli output.

```
void setup()
{
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);
digitalWrite(RED, HIGH);
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);
}
```

Prima di guardare la funzione di “loop”, andiamo a vedere l’ultima funzione nel codice.

La definizione delle variabili.

```
redValue = 255; // choose a value between 1 and 255 to change the color.
greenValue = 0;
blueValue = 0;
```

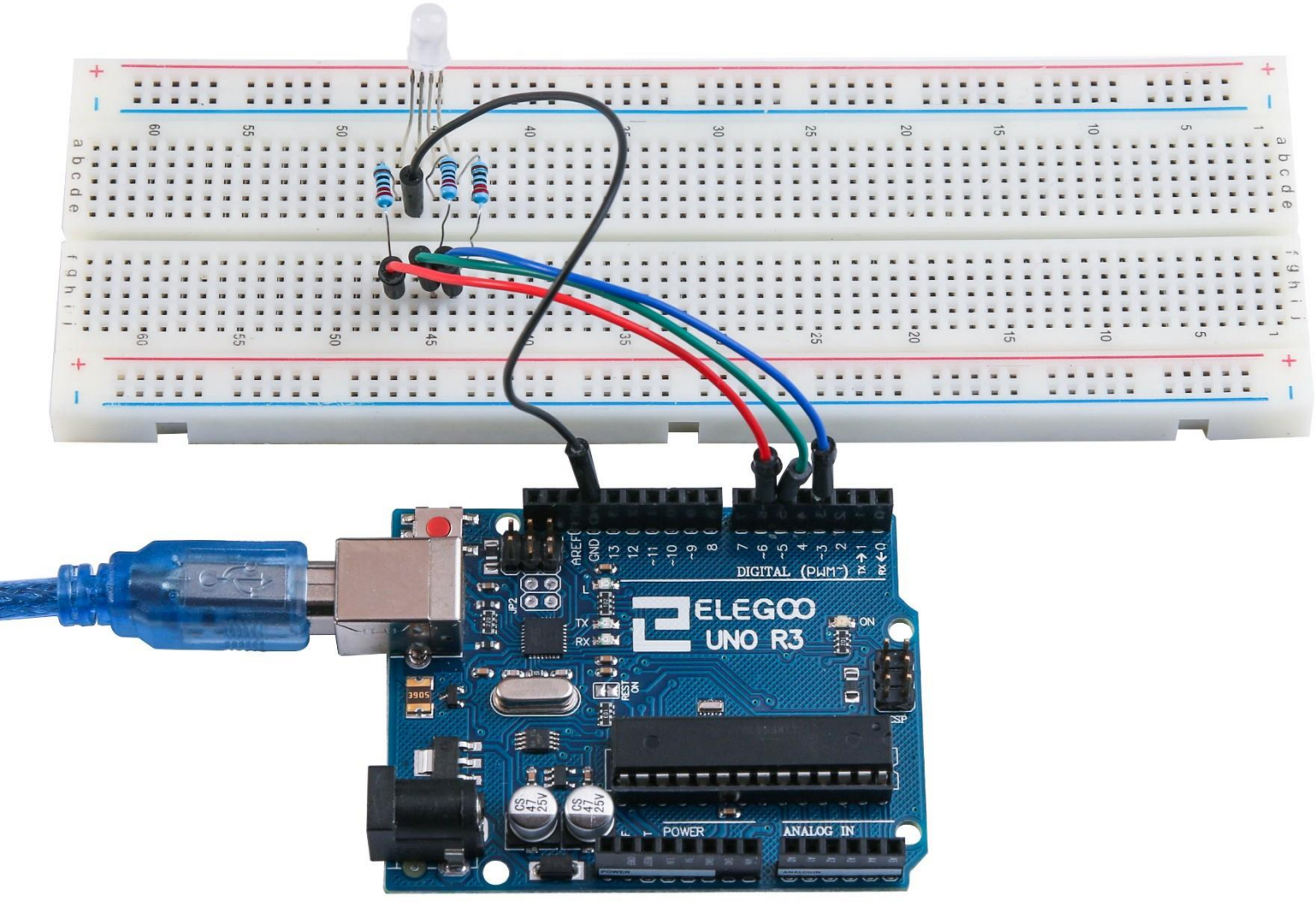
Questa funzione ha tre argomenti, uno per la luminosità del rosso, uno per il verde e uno per il led blu. In ogni caso i tre numeri dovranno avere un range compreso tra 0 e 255, dove 0 significa spento e 255 significa massima luminosità. La funzione si occuperà di chiamare “analogWrite” per impostare la luminosità di ogni led.

Se guardi la funzione “loop” puoi vedere che viene settata la quantità di rosso verde e blu che vogliamo visualizzare, dopodiché viene fatta una pausa di un secondo prima di iniziare con il prossimo colore.

```
#define delayTime 10 // fading time between colors
delay(delayTime);
```

Prova tu stesso ad aggiungere diversi colori al tuo codice e caricalo sulla scheda Arduino per vedere l’effetto del LED.

Foto d’esempio



Lezione 5 Input Digitali

Introduzione

In questa lezione imparerai ad utilizzare il bottone a pressione con un input digitale per accedere e spegnere un LED.

Premendo un primo bottone si accenderà il LED, mentre premendo il secondo bottone il led verrà spento.

Componenti Richiesti:

(1) x Elegoo UNO R3

(1) x Breadboard con 830 punti di aggancio

(1) x LED rosso da 5mm

(1) x Resistenza da 220 ohm

(2) x Interruttore a pressione (bottone)

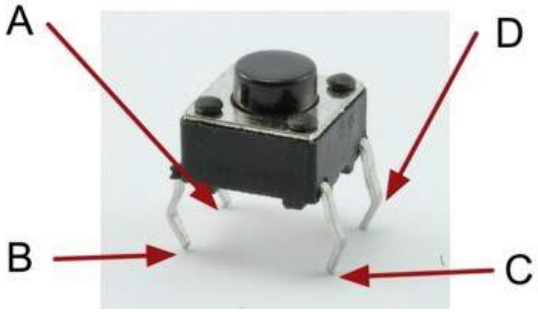
(7) x Connettori M-M (Connettori Maschio-Maschio)

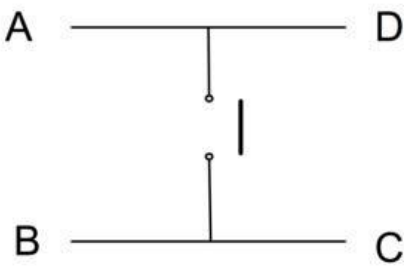
Introduzione ai componenti

INTERRUTTORI A PRESSIONE:

Gli interruttori sono componenti molto semplici. Quando premi un bottone vengono connessi due contatti, in modo che l'elettricità possa fluire attraverso di essi.

Il piccolo interruttore tattile che viene utilizzato in questa lezione ha quattro connessioni, ciò spesso mette in confusione.





In realtà ci sono solamente due connessioni elettriche. All'interno dell'interruttore i pin B e C sono connessi insieme come lo sono anche A e D.

Connessioni

Schema

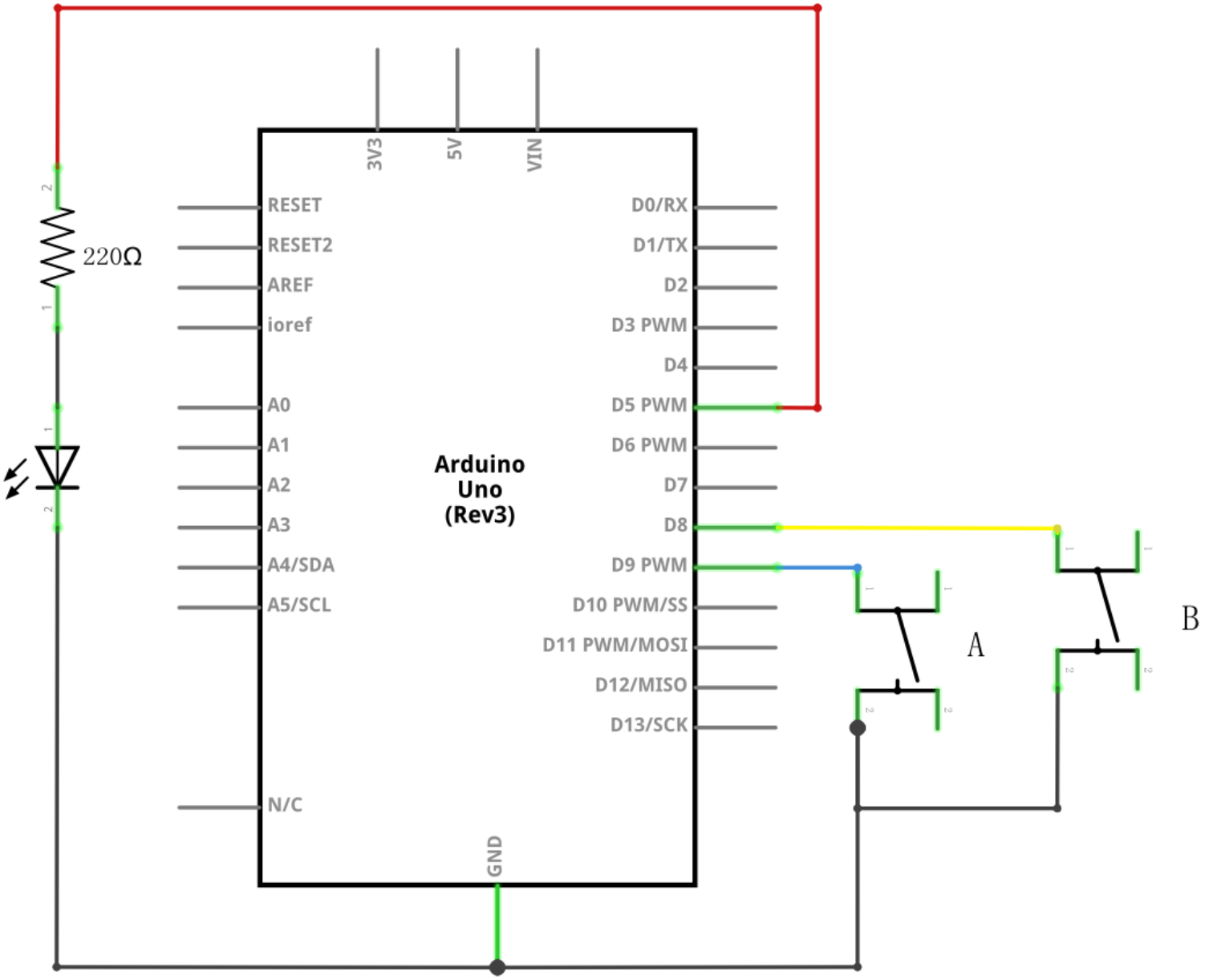
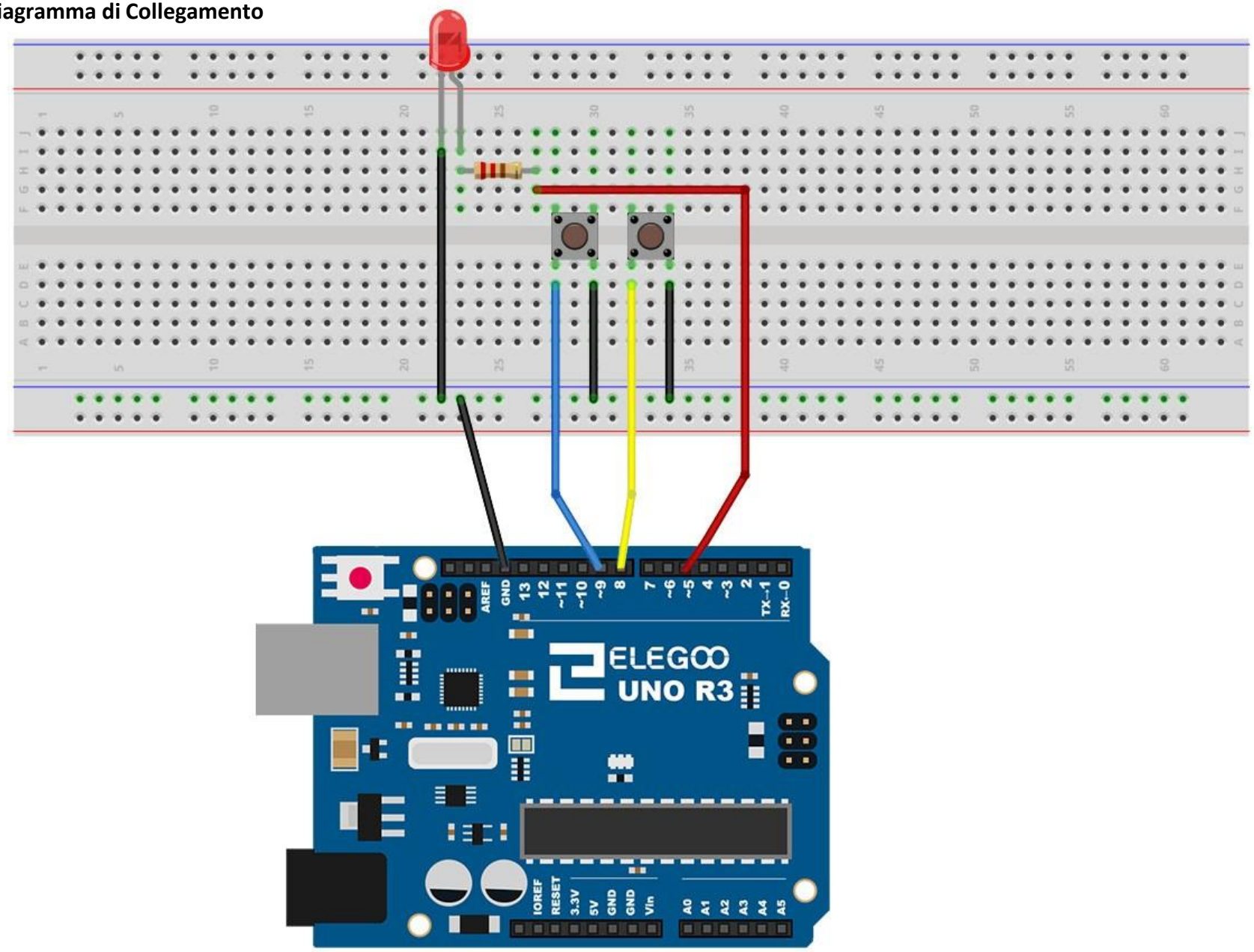


Diagramma di Collegamento



Nonostante il corpo dell'interruttore sia quadrato, i pin sporgono da due lati opposti dell'interruttore. Questo significa che i pin permettono l'inserimento dell'interruttore solamente in un verso sulla breadboard, questo è l'unico verso corretto per evitare cortocircuiti.
Ricordati che il LED deve avere il piedino più corto (negativo) collegato a sinistra.

Codice

Dopo aver effettuato tutti i collegamenti apri il programma che trovi nella cartella "code" chiamato "Lesson 5 Digital Inputs" e premi Upload per caricare il programma. Se visualizzi qualche errore, torna alla lezione 2 per i dettagli su come caricare programmi sulla schedina Arduino.

Dopo aver caricato il codice sulla tua scheda UNO. Premendo il bottone di sinistra vedrai accendersi il led, premendo invece quello di destra il led verrà spento.
La prima parte del codice definisce tre variabili, una per ogni singolo dei te pin che verranno usati. Il "ledPin" è il pin di output, "buttonApin" è il primo dei due bottoni per accendere e spegnere il led, mentre "buttonBpin" è il secondo bottone dei due.
La funzione setup definisce ledPin come un OUTPUT normale, ma ora andiamo a vedere come comportarci con gli altri due input. In questo caso il pinMode dei due interruttori verrà settato così come vedi di seguito ad "INPUT_PULLUP":

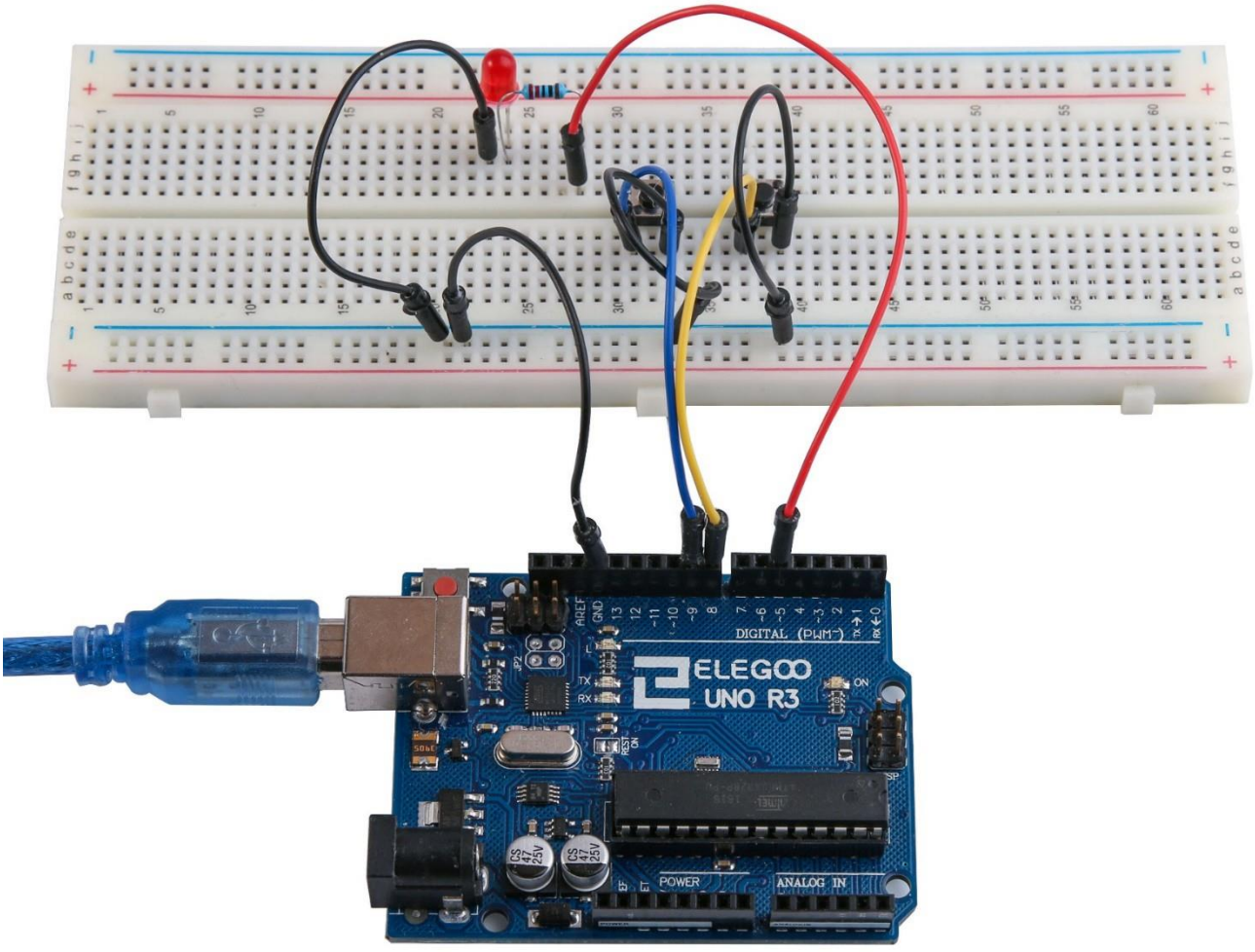
```
pinMode(buttonApin, INPUT_PULLUP);
pinMode(buttonBpin, INPUT_PULLUP);
```

Impostare la modalità di input come INPUT_PULLUP significa che il pin verrà utilizzato come input, ma che se nient'altro è connesso al pin di output esso dovrà essere "pulled up" quindi in modalità HIGH. In altre parole, il valore di default dell'input sarà HIGH a meno che venga portato a LOW della pressione del bottone che ha l'altra coppia di piedini è connesso all'uscita GND.
Quando un bottone viene premuto, connette il pin di input al pin GND perciò il primo non sarà più in modalità HIGH.
Dato che l'input è normalmente in HIGH, e viene portato a LOW solamente quando viene premuto il bottone, il senso del bottone è impostato al contrario. Ci occuperemo di questo in seguito, nella funzione "loop".

```
void loop()
{
   if (digitalRead(buttonApin) == LOW)
   {
      digitalWrite(ledPin, HIGH);
   }
   if (digitalRead(buttonBpin) == LOW)
   {
      digitalWrite(ledPin, LOW);
   }
}
```

Nella funzione di “loop” ci sono due istruzioni “if”. Una per ogni bottone.
Ognuna di esse esegue una lettura “digitalRead” sull’opportuno input.
Ricorda che quando un bottone viene premuto il corrispondente input verrà impostato a LOW, se il bottone A è portato a LOW allora la funzione “digitalWrite” su ledPin verrà impostato ad HIGH, accendendo il led.
Ugualmente, quando il bottone B viene premuto, viene impostato ledPin a LOW spegnendo il led.

Foto d’esempio



Lezione 6 Cicalino Attivo (Active Buzzer)

Introduzione

In questa lezione, imparerai come generare del suono con un cicalino attivo.

Componenti Richiesti:

(1) x Elegoo UNO R3

(1) x Cicalino Attivo (Active buzzer)

(2) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

Introduzione ai componenti

BUZZER:

Il cicalino elettronico è alimentato a corrente continua ed equipaggiato con un circuito integrato.

Essi sono largamente usati nei computer, stampanti, fotocopiatrici, allarmi, giochi elettronici, dispositivi elettronici per autoveicoli, telefoni, cronometri ed altri prodotti di elettronica per dispositivi vocali.

I cicalini vengono divisi in due categorie, attivi e passivi. Girando i due cicalini con i piedini verso l'alto, quello con il circuito verde è il cicalino passivo, al contrario quello chiuso con la superficie nera liscia è quello attivo.

La differenza tra i due è che il cicalino attivo ha un ingresso oscillante, perciò genererà suono quando elettrificato. Un cicalino passivo al contrario, non ha tale ingresso, perciò non farà nessun suon quando riceverà della corrente continua come segnale. Con il cicalino passivo dovrai utilizzare un segnale ad onde quadre la cui frequenza dovrà essere compresa tra i 2K e i 5K.

Il cicalino attivo è spesso più costoso di uno passivo, ciò è dovuto al fatto che all'interno di esso sono presenti alcuni circuiti oscillanti.

Connessione

Schema

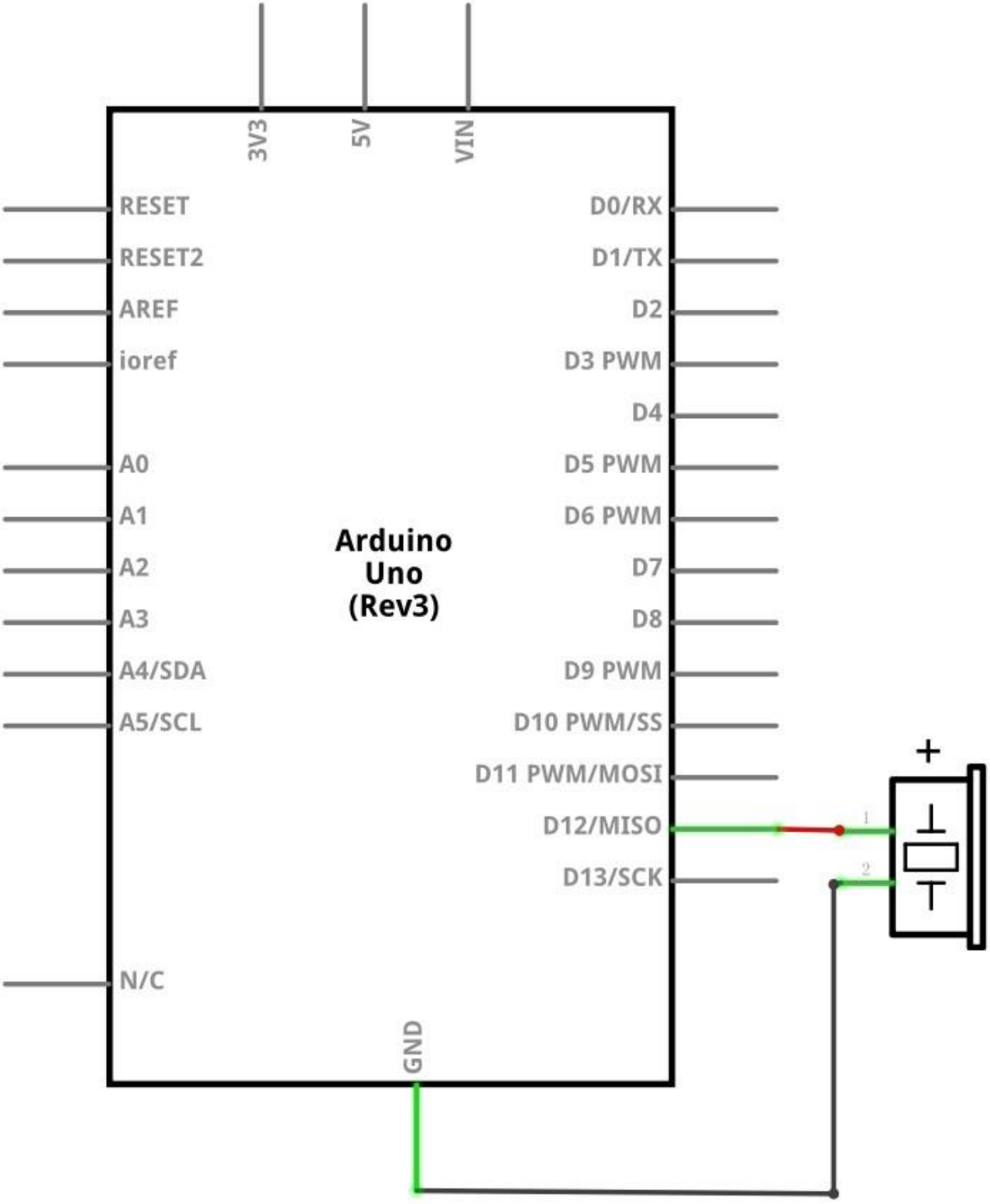
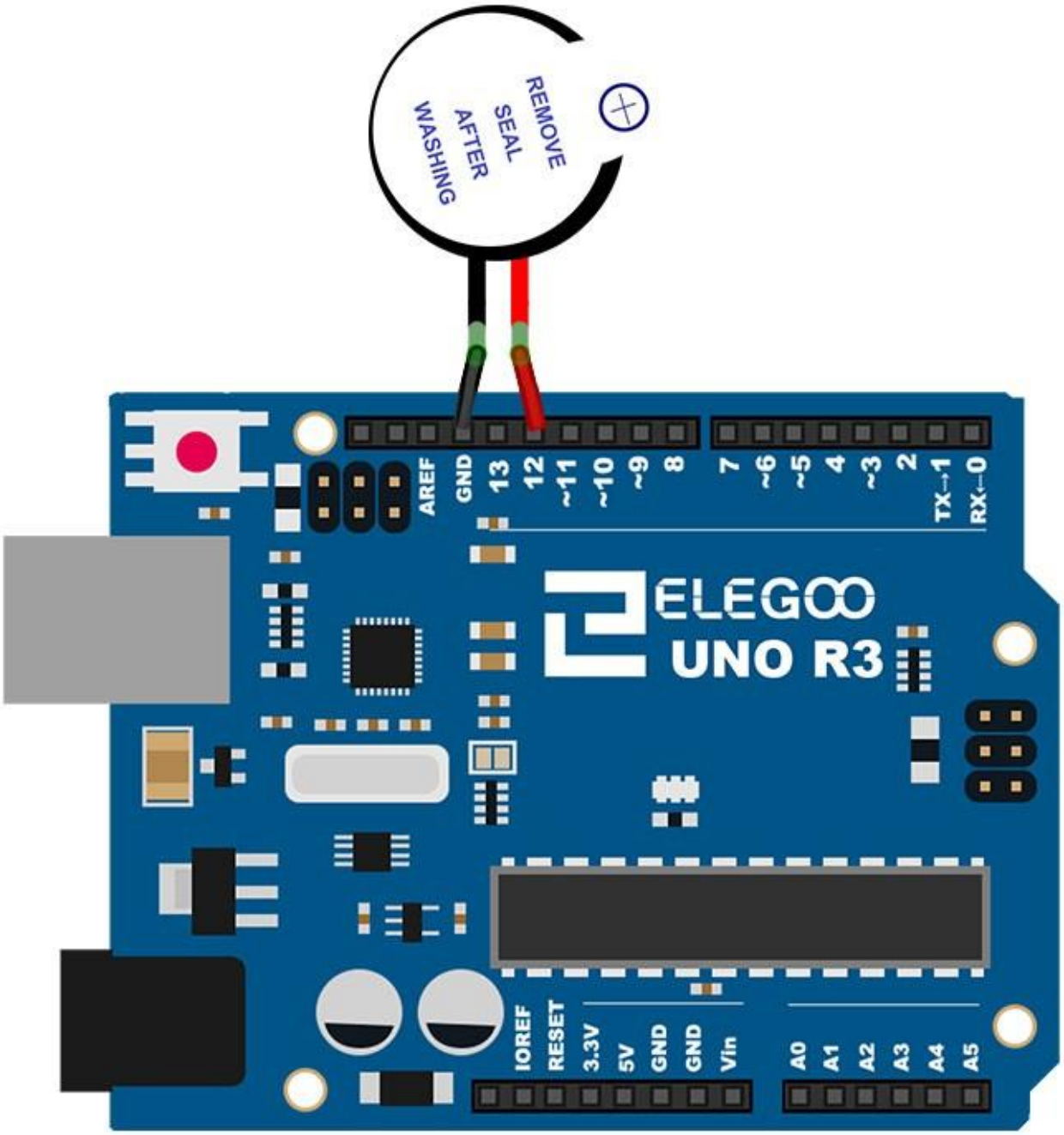


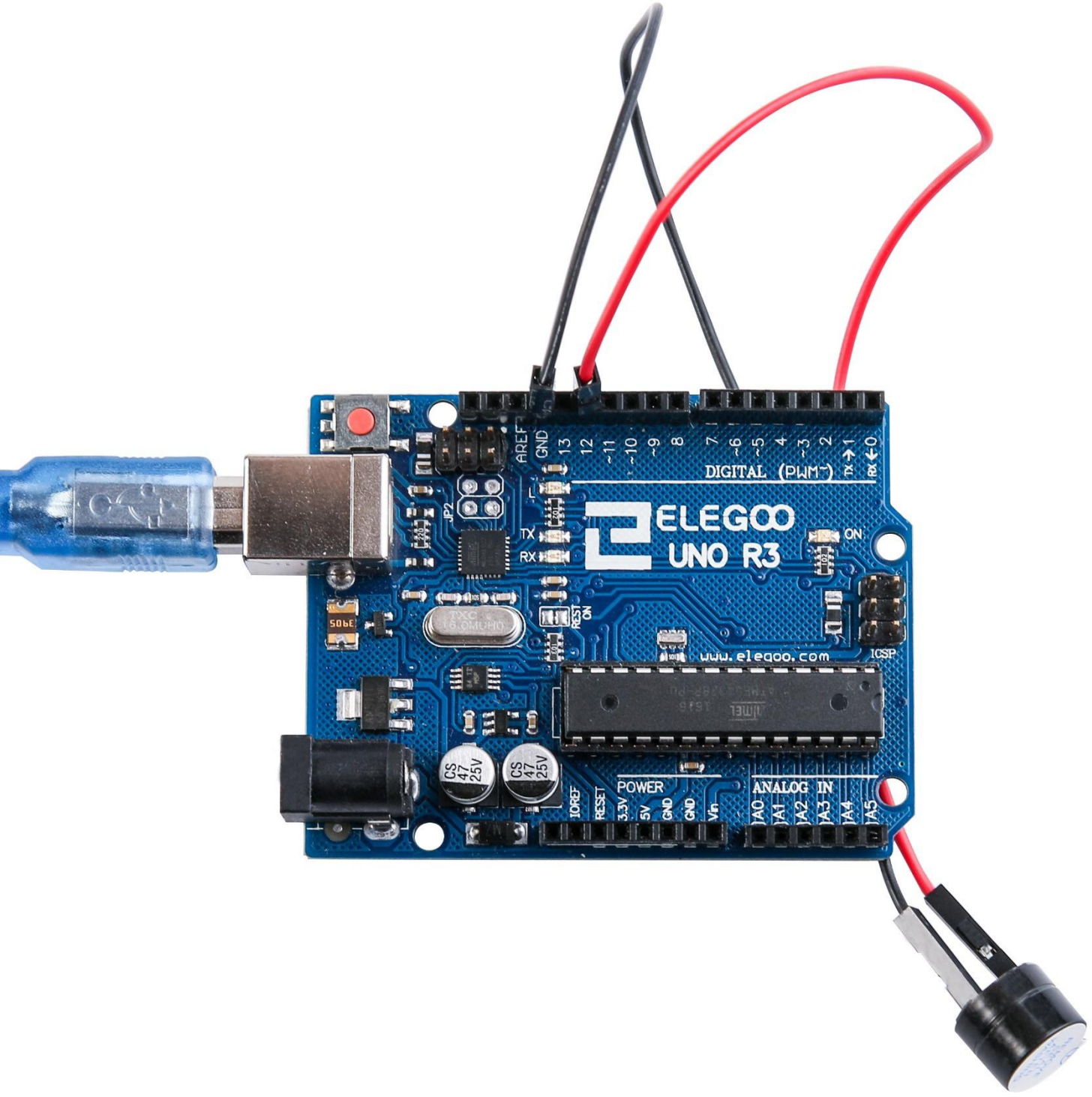
Diagramma di Collegamento



Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 6 Making Sounds”, clicca su UPLOAD per caricare il programma. Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

Foto d’esempio



Lezione 7 Interruttore ad Inclinazione

Introduzione

In questa lezione, imparerai come utilizzare un interruttore ad inclinazione in modo da percepire inclinazione di piccoli angoli.

Componenti Richiesti:

(1) x Elegoo UNO R3

(1) x Interruttore ad inclinazione (Tilt Ball switch)

(2) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

Introduzione ai Componenti

Interruttore ad inclinazione:

L'interruttore ad inclinazione (tilt ball switch) ti permette di percepire orientamento o inclinazione del sensore. Questi sensori sono piccoli, poco costosi, consumano poca corrente e sono semplici da utilizzare. Se usati correttamente questi sensori non si consumano. La loro semplicità li ha resi popolari per giochi, gadget e vari strumenti. Spesso sono chiamati "mercury switches"(interruttori al mercurio), "tilt switches" (interruttori di inclinazione o "rolling ball sensors" (sensori di rotolamento a sfera) per ovvie ragioni.

Questi sensori sono spesso costituiti da qualche tipo di cavità (quella cilindrica è la più comune, ma non l'unica) con una massa mobile conduttiva all'interno, come una bolla di mercurio o una pallina libera di rotolare. Uno dei due estremi della cavità ha due elementi che conducono elettricità (poli). Quando il sensore cambia angolo e viene posto sottosopra, la massa rotola contro i poli formando un cortocircuito, è un comportamento analogo al premere un bottone.

Nonostante non siano precisi o flessibili come dei completi accelerometri, i sensori di inclinazione sono sensibili al movimento e all'orientamento.

Questo tipo di sensori hanno la qualità di poter essere utilizzati da soli senza extra circuiti extra che ne analizzino i segnali.

Gli accelerometri, al contrario, hanno output digitali o analogici che vanno analizzati con circuiti extra

Connessione

Schema

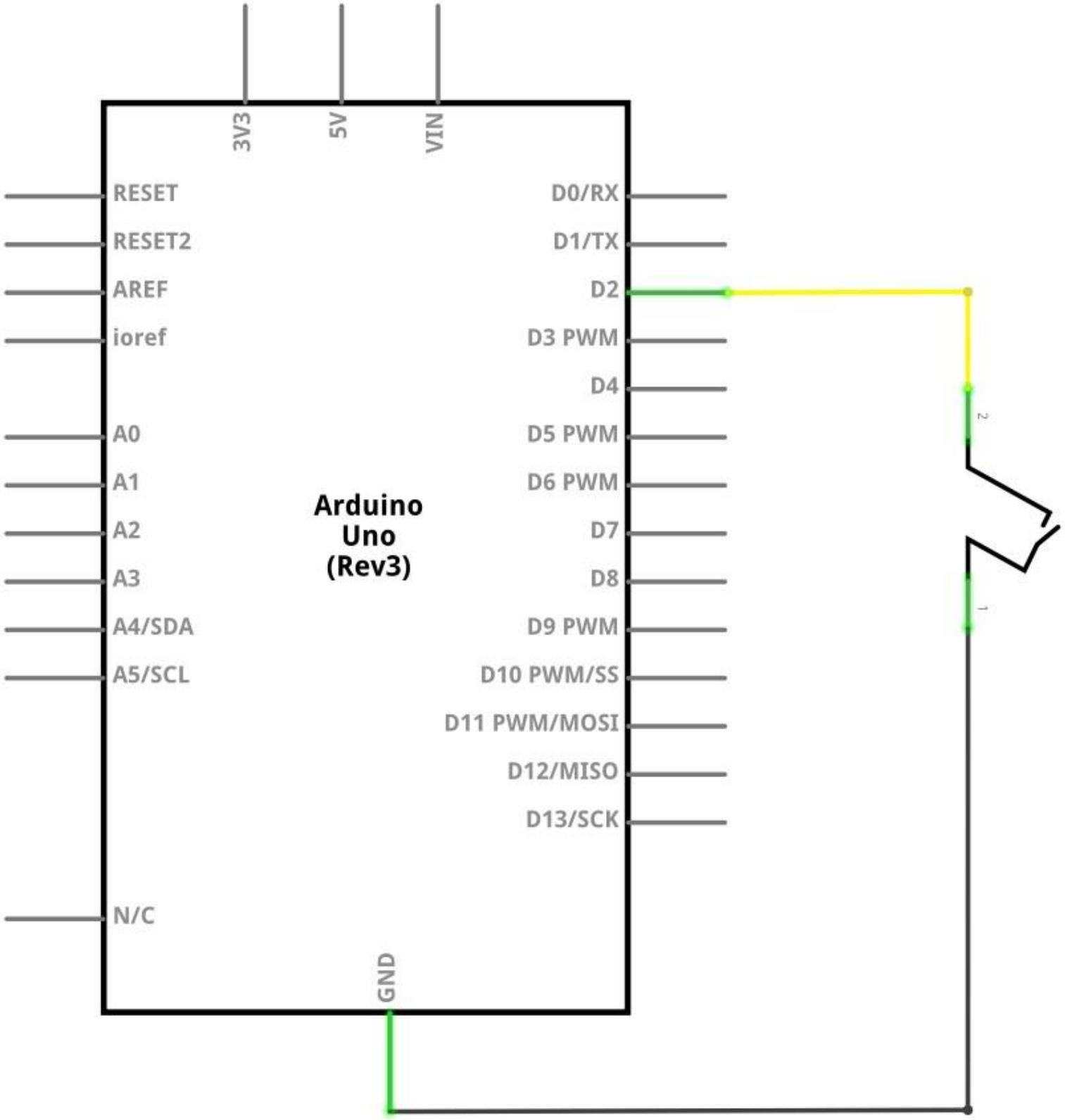
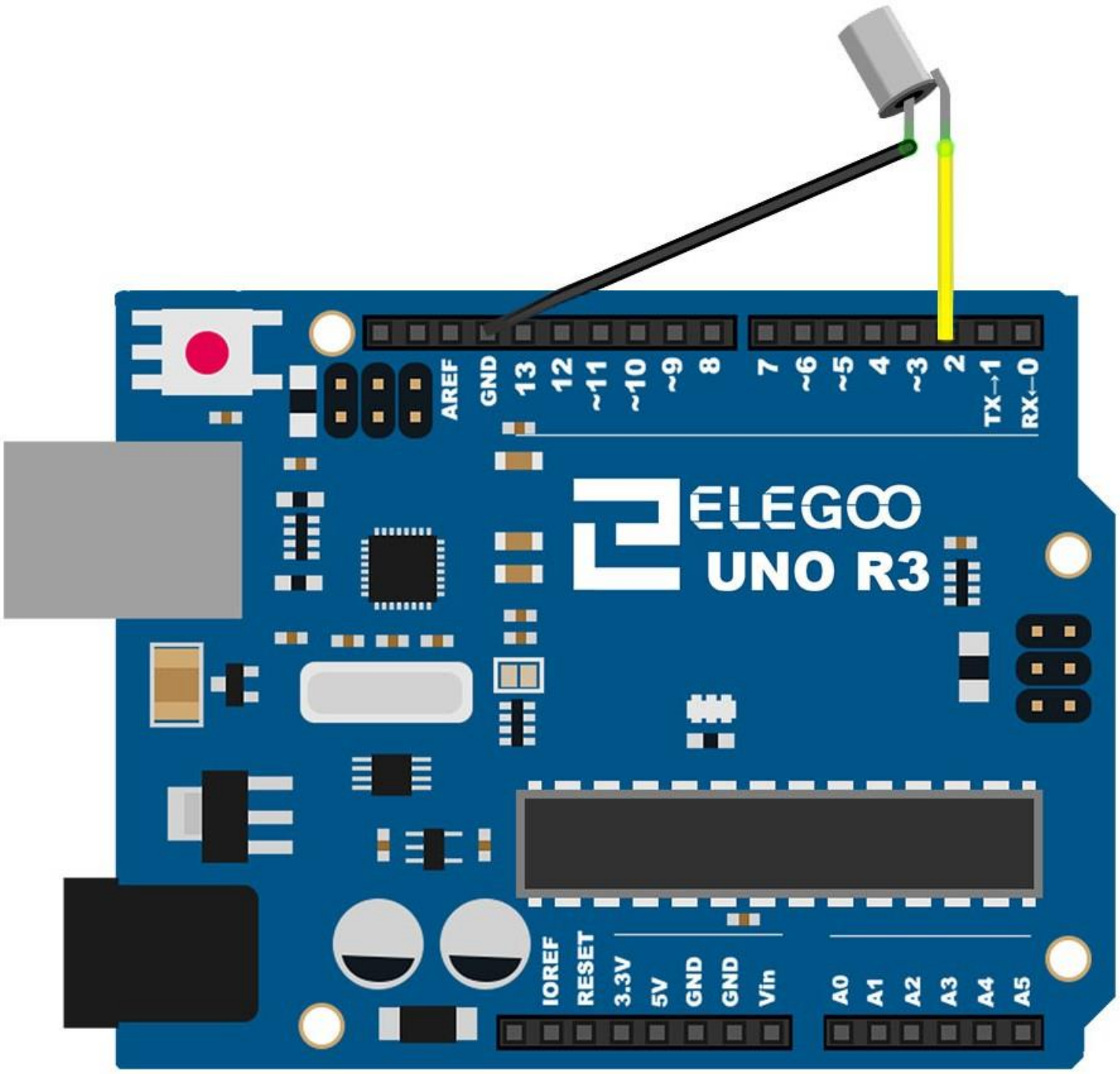


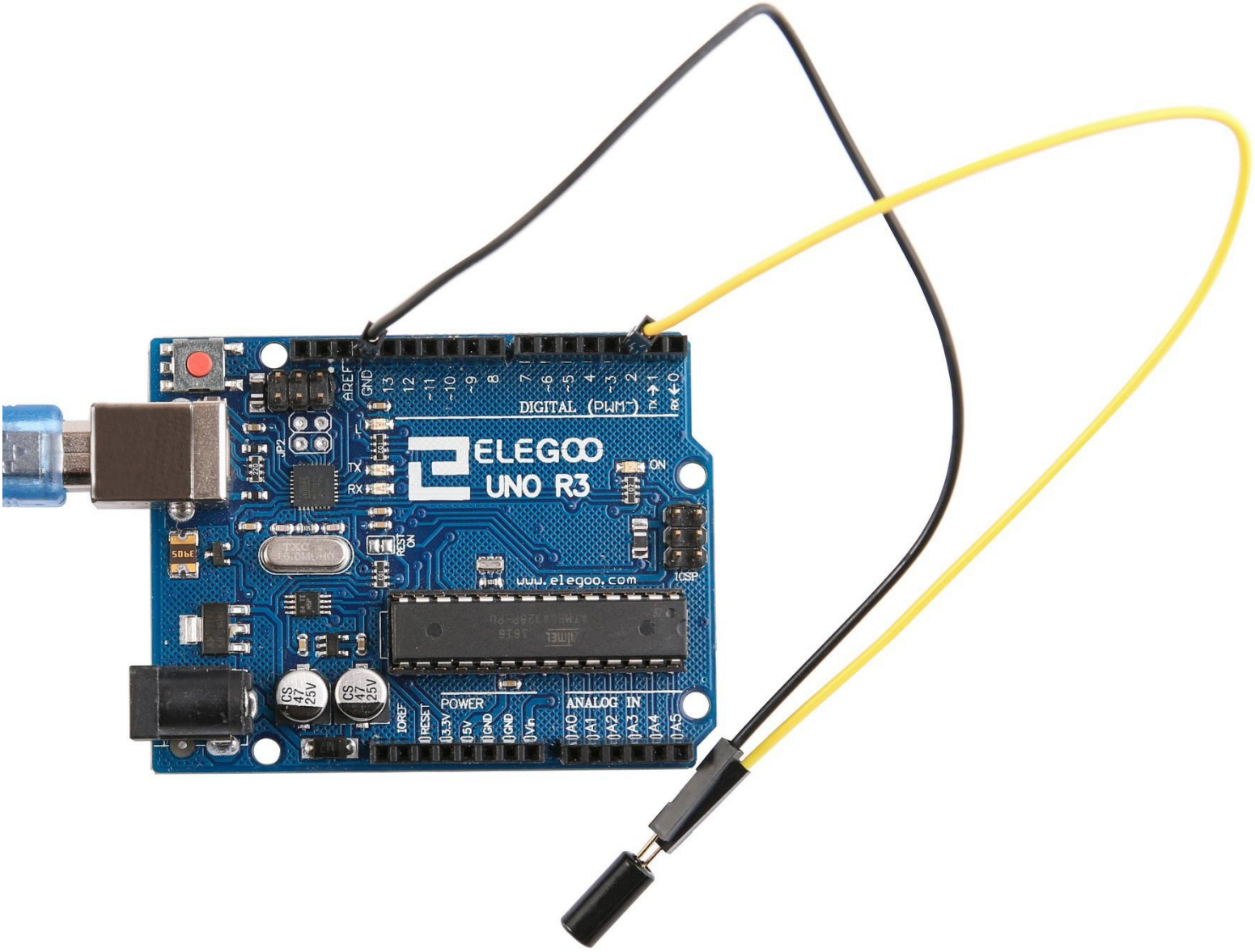
Diagramma di collegamento



Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 8 Ball Switch”, clicca su UPLOAD per caricare il programma.
Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

Foto d’esempio



Lezione 8 Otto LED con 74HC595

Introduzione

In queste lezione imparerai come utilizzare otto led con la tua scheda UNO senza necessità di utilizzare 8 pin di output!
Sebbene tu possa collegare più di 8 led ognuno con una resistenza alla scheda UNO, presto finirai i pin di output disponibili. Se non hai necessità di collegare molti componenti alla scheda non sarà un problema, ma spesso necessitiamo bottoni, sensori, servo, etc. e prima ancora di accorgerti avari finito i pin disponibili. Cosi, invece di far succedere questo, andremo ad utilizzare un chip convertitore da seriale a parallelo chiamato 74HC595. Questo chip ha 8 output (perfetto) e tre input utilizzati per comunicare dati al chip un poco alla volta.
Questo chip rende leggermente più lento il processo che comanda i LED (puoi comandare i led circa 500.000 volte al secondo invece di 8.000.000 al secondo) ma resta comunque molto veloce, e soprattutto più veloce di quanto un occhio umano possa vedere, ne vale quindi la pena!

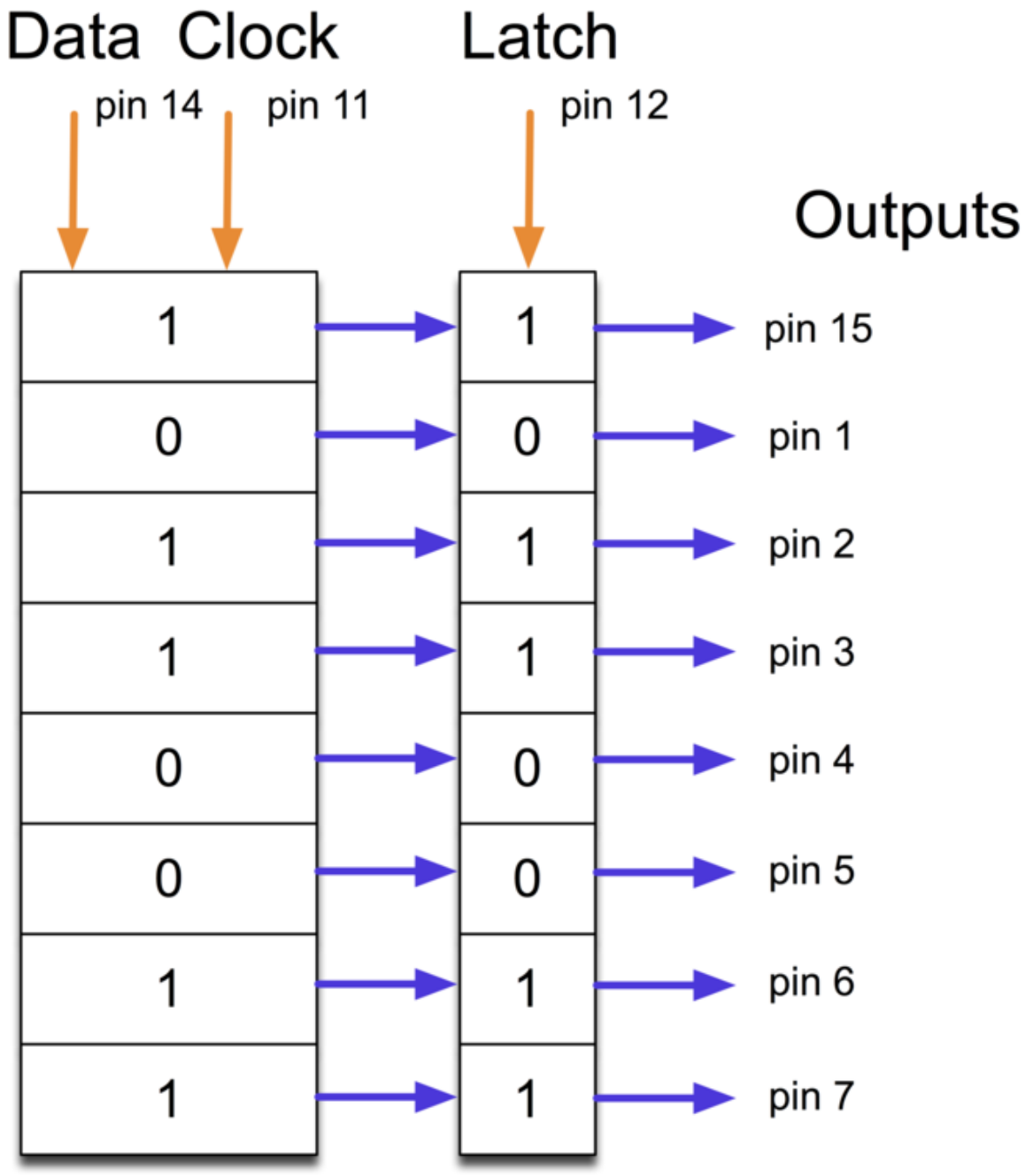
Componenti Richiesti:

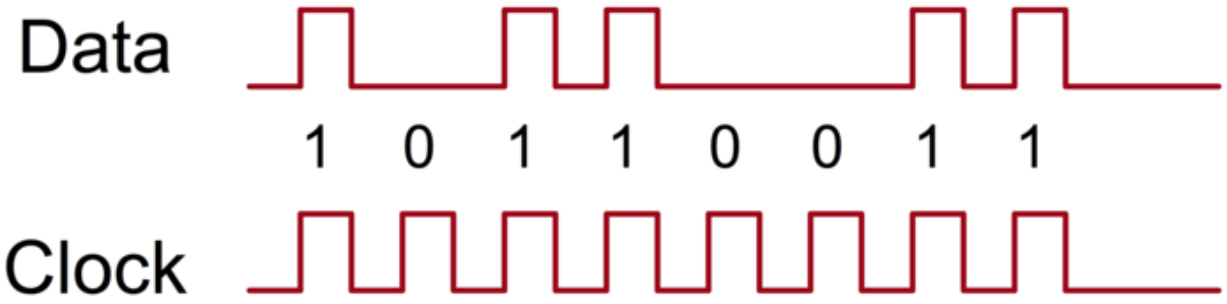
(1) x Elegoo UNO R3
(1) x breadboard con 830 punti di collegamento
(8) x LED
(8) x Resistenze da 220 ohm
(1) x Chip 74hc595
(14) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)

Introduzione al componente

74HC595 Registro a scorrimento:

Il registro a scorrimento è un tipo di chip che mantiene ciò che gli viene comandato per ognuna delle sue 8 zone di memoria, ognuna di essa può registrare i valori 1 o 0. Per impostare questi valori on oppure off, inseriremo i dati utilizzando i pin 'Data' e 'Clock' sul chip.





è HIGH, un 1 viene inserito nel registro a scorrimento, altrimenti viene inserito uno

0. Quando tutti e 8 i valori sono ricevuti, abilitando il pin ‘Latch’ il chip tutti questi valori vengono copiati sul registro di Latch. Questo è necessario altrimenti, mentre viene impostato il valore per i led nel registro a scorrimento vedremmo tremolare la luce dei led.

Questo chip ha inoltre un pin di abilitazione output (OE), esso è utilizzato per abilitare o disabilitare tutti gli output in una volta sola. Puoi collegare questo ad un pin PWM della scheda UNO ed utilizzare ‘analogWrite’ per controllare la luminosità dei led. Questo pin è attivo a LOW, perciò possiamo collegarlo al pin GND.

Connessione

Schema

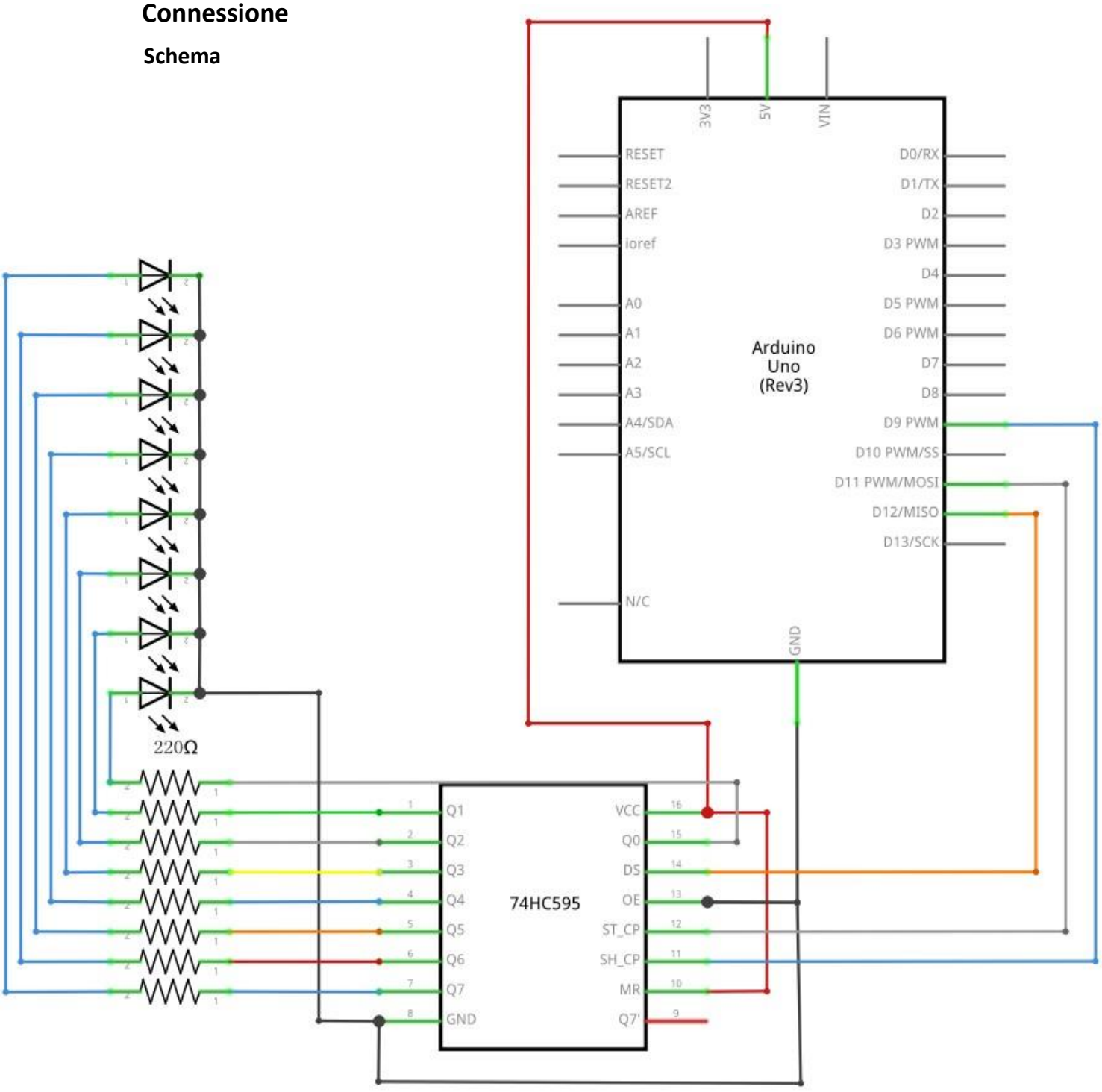
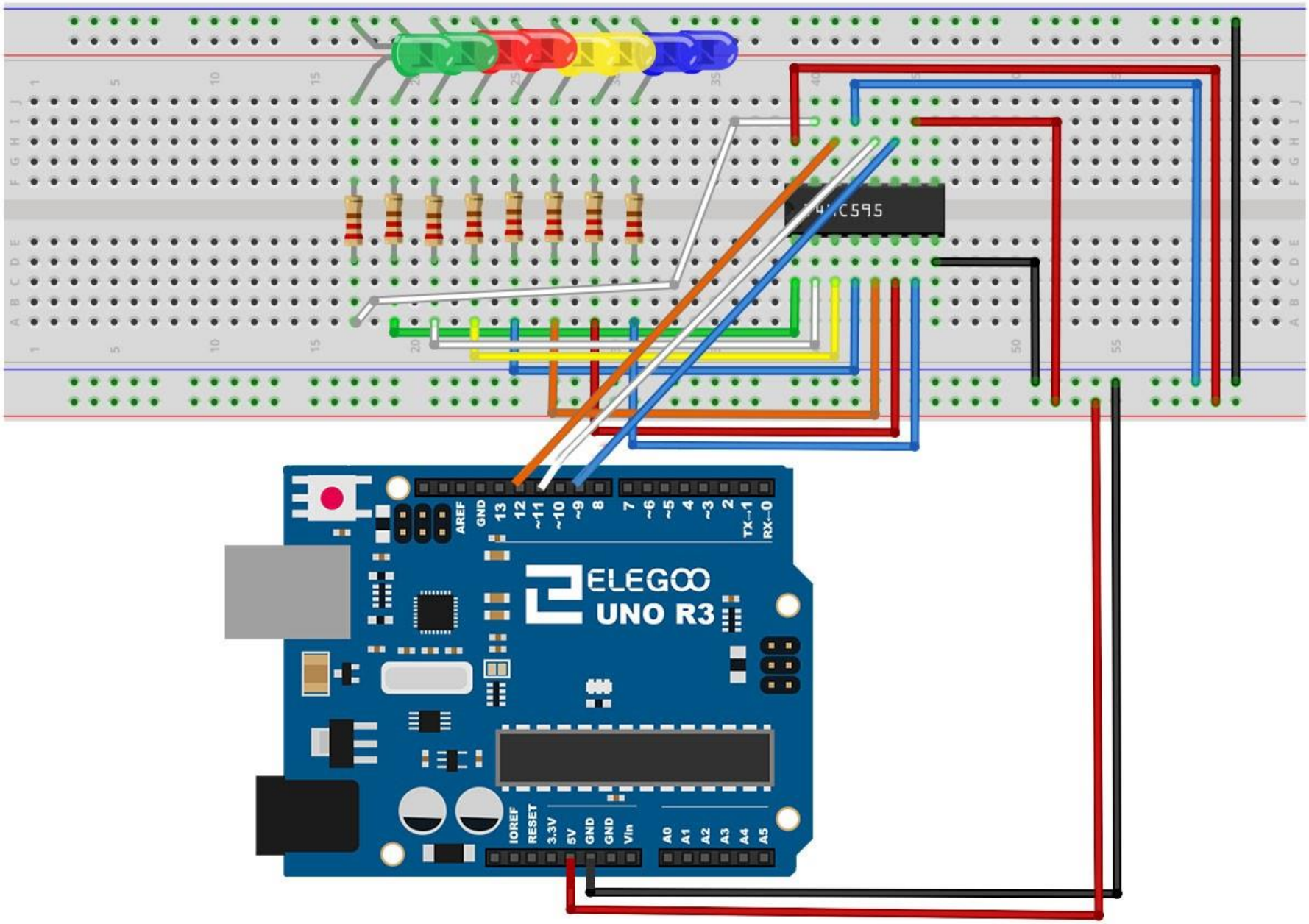


Diagramma di collegamento



Visto che abbiamo otto led ed otto resistenze da utilizzare, ci sono un bel po' di connessioni da fare.

È probabilmente più facile collegare prima il chip 74HC595 dato che quasi tutte le connessioni passeranno attraverso di esso. Collegatelo con il piccolo intaglio a forma di U sulla parte superiore rivolta verso la parte alta della breadboard. Il pin 1 del chip è quello alla sinistra dell'intaglio.

Il Pin digitare 12 dalla scheda UNO va collegato al pin #14 del chip a scorrimento di registro.

Il Pin digitare 11 dalla scheda UNO va collegato al pin #12 del chip a scorrimento di registro.

Il Pin digitare 19 dalla scheda UNO va collegato al pin #11 del chip a scorrimento di registro.

Tutti gli output tranne uno del chip sono posizionato sul lato sinistro del circuito. Quindi per semplificare le connessioni, metteremo i led da quel lato.

Dopo aver posizionato il chip, metti in posizione anche i resistori. Devi prestare attenzione che nessun capo dei resistori ne tocchi con un altro. Ti suggeriamo di controllare questa cosa ancora prima di alimentare la scheda UNO. Se trovi difficoltà nell'aggiustare la posizione di resistori, puoi aiutarti tagliando una parte del loro filo metallico, accorciandolo, e facendo in modo che il resistore stia più vicino alla superficie delle breadboard.

Ora posizioni i led sulla breadboard. Il piedino più lungo del led è quello positivo ed esso va collegato in direzione del chip, indipendentemente dalla posizione della breadboard alla quale sono collegati.

Collega i cavetti di collegamento come mostrato qui sotto. Non dimenticarti quello che collega il pin 8 del circuito integrato alla colonna della messa a terra GND della breadboard. Carica il codice indicato qui sotto e prova i led! Ogni led dovrebbe illuminarsi a turno prima che tutti siano accesi, dopodiché si spegneranno tutti ed il ciclo si ripeterà.

Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lesson 24 Eight LED with 74HC595", clicca su UPLOAD per caricare il programma.

Se hai dubbio riguardo all'uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

La prima cosa da fare è definire i 3 pin che andremo ad utilizzare. Essi sono i pin di output digitale della scheda UNO che saranno connessi a Latch, clock e Data del circuito integrato 74HC595.

```
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
```

Ora, la variabile ‘leds’ è definita. Questa verrà utilizzate per mantenere il pattern di quali led sono attualmente accesi o sopenti. Il tipo di dato ‘byte’ rappresenta numeri utilizzando otto bit. Ogni bit può essere on oppure off (0 o 1), ciò è perfetto per tenere traccia degli otto led per conoscerne il loro stato.

```
byte leds = 0;
```

La funzione 'setup' semplicemente setta i tre pin digitali che useremo come output.

```
void setup()
{
   pinMode(latchPin,  OUTPUT);
   pinMode(dataPin,  OUTPUT);
   pinMode(clockPin, OUTPUT);
}
```

La funzione ‘loop’ inizialmente spegne tutti i led, dando alla variabile ‘leds’ il valore 0. Dopo di che chiama la funzione 'updateShiftRegister', che invierà il pattern al registro a scorrimento, così tutti i led verranno spenti. Ci occuperemo del funzionamento della funzione 'updateShiftRegister' più avanti.

La funzione di loop attende per mezzo secondo, dopodiché inizia a contare da 0 a 7, utilizzando il ciclo ‘for’ e la variabile ‘i’. Ogni volta, utilizza la funzione ‘bitSet’ per settare il bit he controlla il led nella variabile ‘leds’. Chiama anche la funzione 'updateShiftRegister' in modo da aggiornare i led e farli corrispondere al valore della variabile ‘leds’.

C’è mezzo secondo di intervallo tra l’aggiornamento del valore della ‘i’ e l’illuminazione del LED.

```
void loop()
{
   leds = 0;
   updateShiftRegister();
   delay(500);
   for (int i = 0; i < 8; i++)
```

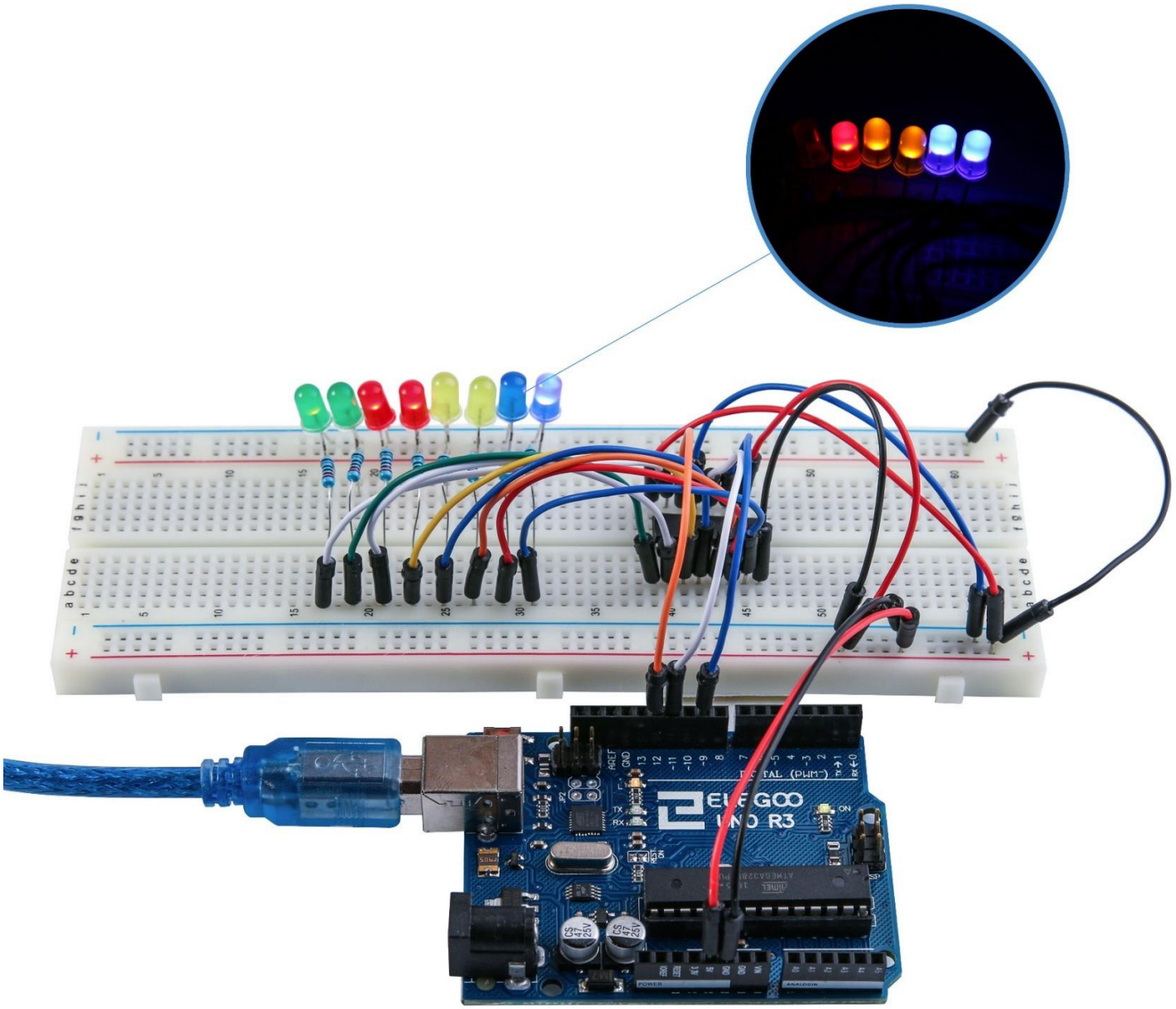
```
  {
    bitSet(leds, i);
    updateShiftRegister();
    delay(500);
  }
}
```

La funzione 'updateShiftRegister', per prima cosa imposta il valore di ‘latchPin’ a LOW, in seguito chiama la funzione 'shiftOut' prima di impostare nuovamente ‘latchPin’ ad HIGH. La funzione prende quattro parametri. Due si essi sono i pin utilizzati per Data e Clock. Il terzo parametro specifica da quale lato del registro si vuole iniziare. Noi inizieremo da destra con il bit meno significativo 'Least Significant Bit' (LSB). L’ultimo parametro è il dato che verrà inserito nel registro, nel nostro caso sarà ‘leds’.

```
void updateShiftRegister()
{
   digitalWrite(latchPin, LOW);
   shiftOut(dataPin, clockPin, LSBFIRST, leds);
   digitalWrite(latchPin, HIGH);
}
```

Nel caso tu voglia accendere un LED piuttosto che spegnerlo, puoi utilizzare una funzione di Arduino (bitClear) sulla variabile ‘leds0. Questa imposterà il bit scelto della variabile ‘leds’ a 0 e dovrai semplicemente chiamare la funzione 'updateShiftRegister' per aggiornare i LED.

Foto d’esempio



Lezione 9 Il Monitor Seriale

Introduzione

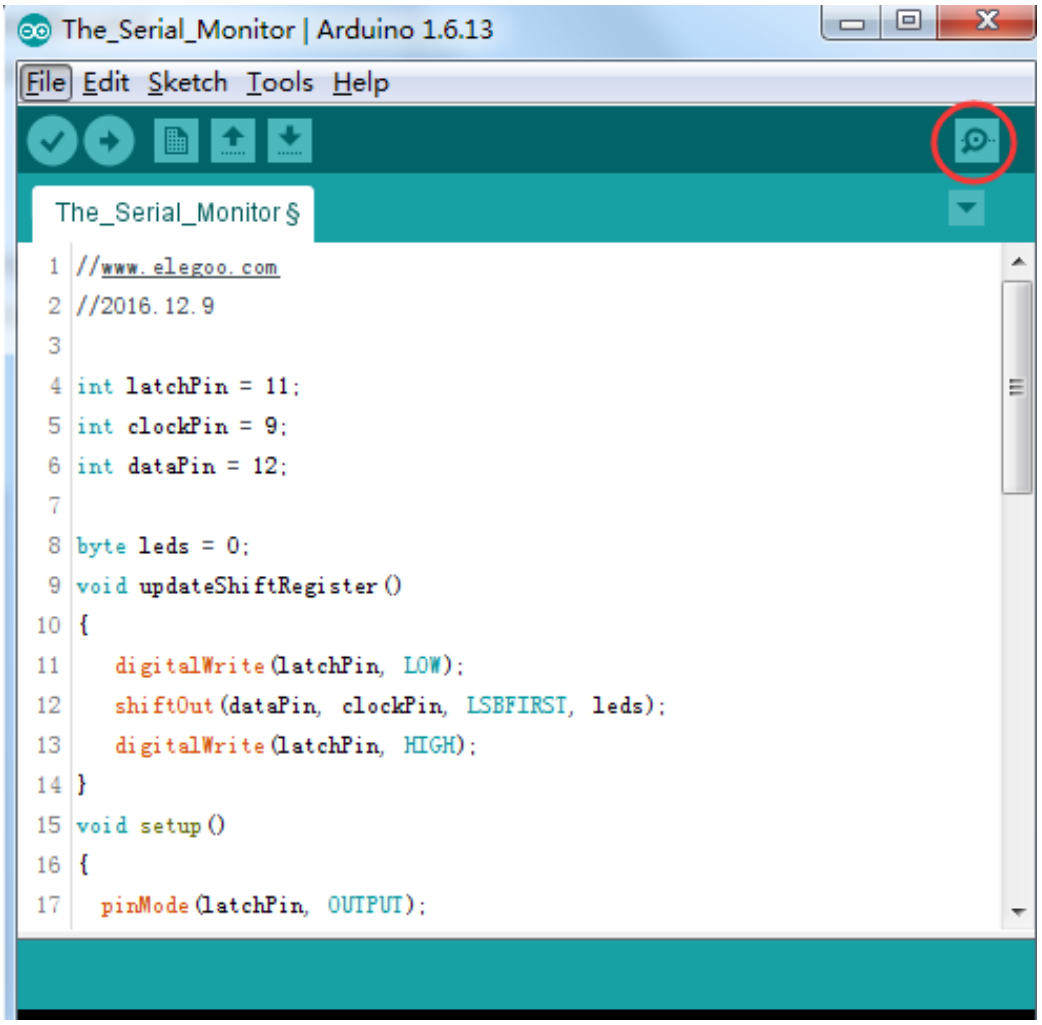
In questa lezione, utilizzeremo il progetto costruito nella lezione 8 aggiungendo la possibilità di controllare i LED dal computer utilizzando il controllo seriale di Arduino. Il monitor seriale è il 'collegamento' tra il tuo PC e la tua scheda UNO. Esso ti permette di inviare e ricevere messaggi di testo, utilizzato spesso per debug della piattaforma e controllo della scheda UNO tramite la tastiera!

Per esempio, sarai in grado di inviare comandi dal tuo pc ed accedere i LED.

In questa lezione utilizzerai esattamente le stesse parti ed un layout della breadboard simile alla lezione 8, perciò se non lo hai già fatto, torna ora alla lezione 8.

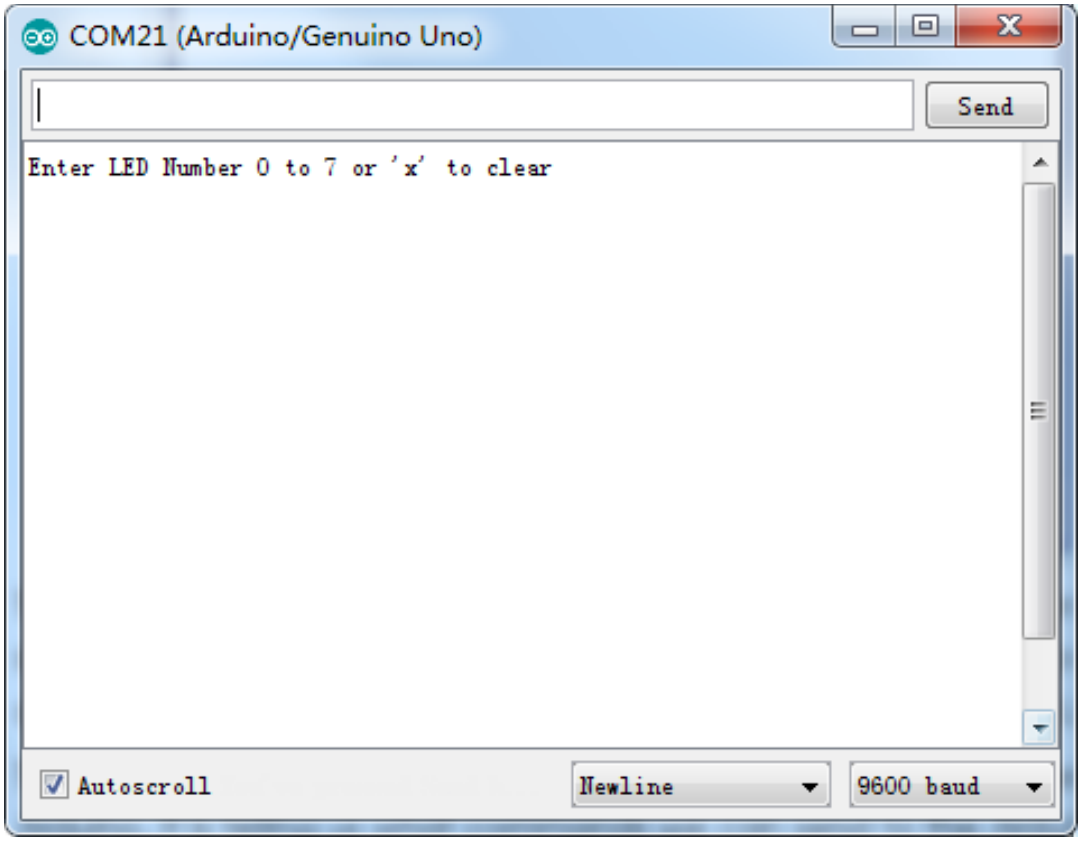
Passaggi

Dopo aver caricato il codice sulla tua scheda UNO clicca in alto a destra, sul bottone presente sulla barra degli strumenti dell'IDE di Arduino. Il bottone è lo stesso che vedi cerchiato qui sotto.



La seguente finestra si aprirà.

Clicca sul bottone del monitor seriale per attivare il monitor seriale. Le vasi del monitor seriale sono state introdotte in dettaglio nella lezione numero 1.



Questa finestra è chiamata Serial Monitor ed è parte del software IDE di Arduino. Il suo lavoro è permettere di inviare messaggi dal tuo computer alla scheda UNO (tramite USB) ma anche di ricevere messaggi da essa.

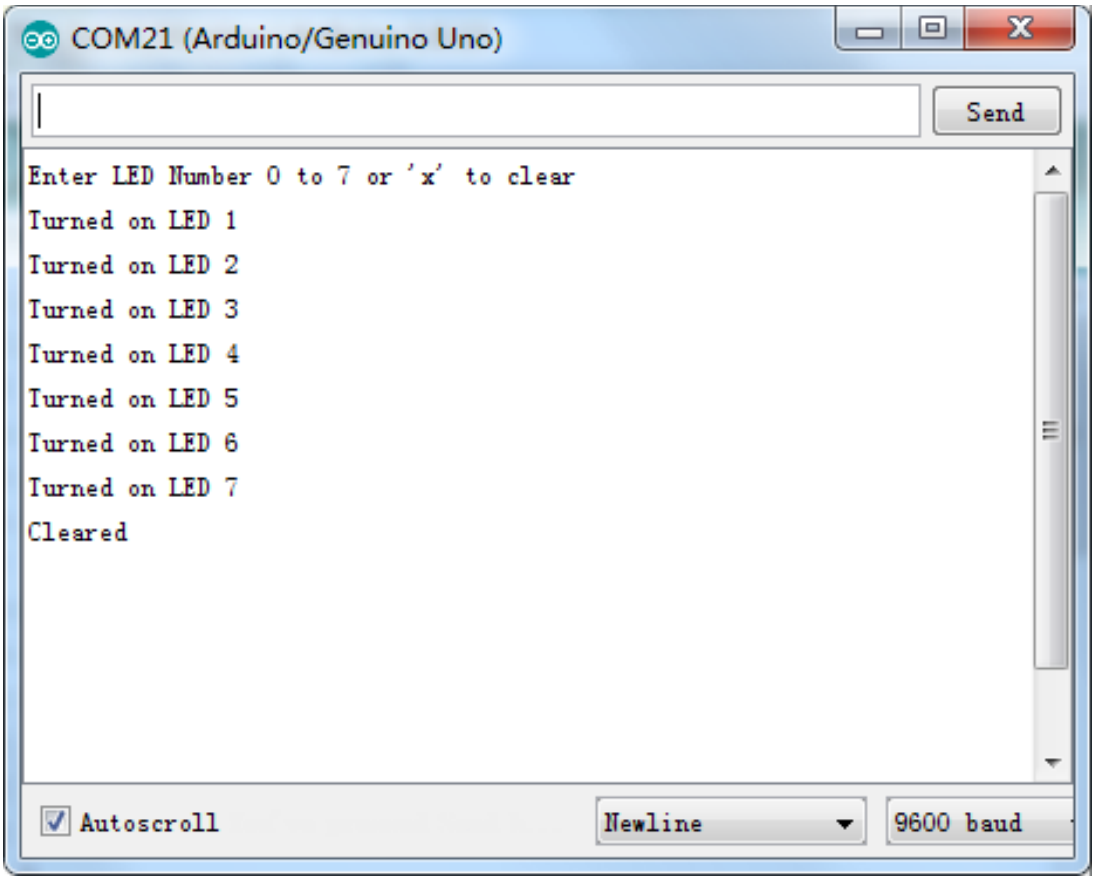
Il messaggio “Enter LED Number 0 to 7or 'x' to clear” viene inviato da Arduino e ci spiega quale comando possiamo inviare ad esso: possiamo mandare ‘x’ (per spegnere tutti i LED) oppure il numero del led che vogliamo accendere (dove 0 è il LED più in basso, 1 è il successivo verso l’alto, fino al 7 che è l’ultimo più in alto).

Prova a digitare i seguenti comando nell’area superiori del monitor seriale, quella allo stesso livello del bottone “send”. Premi “send” dopo aver digitato ogni carattere:

x 0 3 5

Digitare x non avrà alcun effetto in quanto tutti i led sono già spenti, ma quando digiterai ogni numero il corrispondente led si accenderà e riceverai il messaggio di

conferma dalla scheda UNO. Il monitor seriale apparirà come mostrato qui sotto.



Digita nuovamente x e premi “send!” per spengere tutti i LED.

Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 25 The Serial Monitor”, clicca su UPLOAD per caricare il programma.

Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

Come potresti esserti aspettato, il codice è basato sul codice della lezione 24. Così, ora ti spiegheremo solo una parte di esso. Troverai utile far riferimento al codice completo presente sul tuo IDE Arduino.

Alla fine della funzione di ‘setup’ ci sono tre nuove linee.

```
void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
```

```
  pinMode(clockPin, OUTPUT);
  updateShiftRegister();
  Serial.begin(9600);
  while (! Serial); // Wait until Serial is ready - Leonardo
  Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

Per prima cosa, abbiamo il comando 'Serial.begin(9600)'. Questo apre la comunicazione sul monitor seriale, in modo che la scheda UNO possa inviare comandi tramite la connessione USB. Il valore 9600 è chiamato 'baud rate' della connessione. Esso indica la velocità con cui i dati sono inviati. Puoi modificare questo valore ed alzarlo, ma dovrai cambiarlo anche sul serial monitor di Arduino inserendo lo stesso valore. Parleremo di questo più avanti, ma per ora lasciamolo a 9600.

La linea inizia con un ciclo di 'while' per assicurarsi che ci sia qualcuno dall'altro capo del cavo USB con cui comunicare prima di iniziare ad inviare messaggi.

Altrimenti, il messaggio potrebbe essere inviato ma non visualizzato. Questa linea è necessaria solamente se stai utilizzando un Arduino Leonardo perché la scheda Arduino UNO Si resetta in automatico quando apri il monitor seriale, al contrario questo non succede con la scheda Leonardo.

L'ultima delle nuove linee in 'setup' invia il messaggio che vedremo in cima al monitor seriale.

La funzione 'loop' è dove avvengono tutte le azioni.

```
void loop()
{
  if (Serial.available())
  {
    char ch = Serial.read();
    if (ch >= '0' && ch <= '7')
    {
      int led = ch - '0';
      bitSet(leds, led);
      updateShiftRegister();
      Serial.print("Turned on LED ");
      Serial.println(led);
    }
    if (ch == 'x')
```

```
      {
        leds = 0;
        updateShiftRegister();
        Serial.println("Cleared");
      }
    }
}
```

Tutto ciò che avviene all'interno del loop è contenuto all'interno della condizione if. Così, se la chiamata della funzione 'Serial.available()' non ritorna il valore 'true', nulla accade.

Serial.available() ritorna 'true' se dei dati sono stati inviati alla scheda UNO ed essa è pronta a processare. I messaggi in ingresso sono mantenuti in quello che è chiamato buffer e 'Serial.available()' ritorna 'true' se il buffer non è vuoto.

Se un messaggio è stato ricevuto, allora sarà nella successiva linea di codice:

```
char ch = Serial.read();
```

Questa funzione legge il prossimo carattere presente nel buffer e lo rimuove da esso. Assegna alla variabile 'ch' il valore letto. La variabile 'ch' è di tipo 'char' che è l'abbreviazione di 'character' e come suggerisce il nome, permette di salvare un singolo carattere.

Se hai seguito le istruzioni nel prompt in cima al monitor seriale, questo carattere sarà una singola digitazione, e sarà o un numero compreso tra 0 e 7 inclusi oppure la lettera 'x'.

La dichiarazione 'if' nella linea seguente controlla se è il carattere digitato è una singola digitazione controllando che 'ch' sia maggiore o uguale al carattere '0' e minore o uguale al carattere '7'. Può sembrarti strano comparare dei caratteri in questa maniera, ma è perfettamente accettabile.

Ogni carattere è rappresentato da un unico numero, chiamato valore ASCII. Questo significa che comparando dei caratteri usando >= e <= si comparano i rispettivi valori ASCII.

Se il test è passato, allora si prosegue alla prossima linea di codice:

```
int led = ch – '0';
```

Ora stiamo eseguendo dei calcoli aritmetici con dei caratteri. Stiamo sottraendo il carattere '0' da qualsiasi valore sia stato digitato. In questo modo se è stato digitato '0' allora '0' – '0' equivale a 0. Se è stato digitato '7' allora '7' – '0' equivale a 7 perché attualmente è il valore ASCII utilizzato nella sottrazione.

Dato che conosciamo il numero del LED che vogliamo accendere necessitiamo solamente di aggiornare il corretto bit nella variabile ‘leds’ ed aggiornare il registro a scorrimento.

```
bitSet(leds, led);
updateShiftRegister();
```

Le seguenti due linee visualizzano un messaggio di conferma sul monitor seriale.

```
Serial.print("Turned on LED ");
Serial.println(led);
```

La prima linea utilizza Serial.print invece di Serial.println. La differenza tra le due è che Serial.print non va a capo dopo aver stampato il messaggio passatogli come parametro. Usiamo questo nella prima linea, perché stiamo stampando il messaggio in due parte, prima il messaggio generico 'Turned on LED ' e di seguito il numero del LED scelto.

Il numero del led è salvato in una variabile ‘int’ invece di essere in una stringa di testo. Serial.print è in grado di ricevere come parametri sia stringe di testo incluse tra doppi apici, sia variabili ‘int’ o per questo motivo quasi ogni tipo di variabile.

Dopo che lo la condizione ‘if’ è stata controllata, quando ogni singola digitazione è stata manipolata, c’è un secondo ‘if’ che si occupa di controllare se il carattere ‘ch’ equivale alla lettera ‘x’.

```
if (ch == 'x')
{
   leds = 0;
   updateShiftRegister();
   Serial.println("Cleared");
}
```

Se così è, Allora pone a 0 tutti i valori dei led, aggiorna il registro a scorrimento e invia un messaggio di conferma.

Lezione 10 Fotocellula

Introduzione

In questa lezione imparerai come misurare l'intensità della luce utilizzando un input analogico. Nella lezione costruirai un circuito in grado di usare il livello della luce per controllare la quantità di LED che si accenderanno. La fotocellula sarà situata nella parte inferiore della breadboard ad ha un funzionamento simile ad un potenziometro.

Componenti Richiesti:

(1) x Elegoo UNO R3
(1) x Breadboard con 830 punti di collegamento
(8) x LED
(8) x Resistenze da 220 ohm
(1) x Resistenze da 1k ohm
(1) x Circuito integrato 74hc595
(1) x Fotoresistore (Fotocellula)
(16) x M-M Cavetti (Cavetti di collegamento Maschio - Maschio)

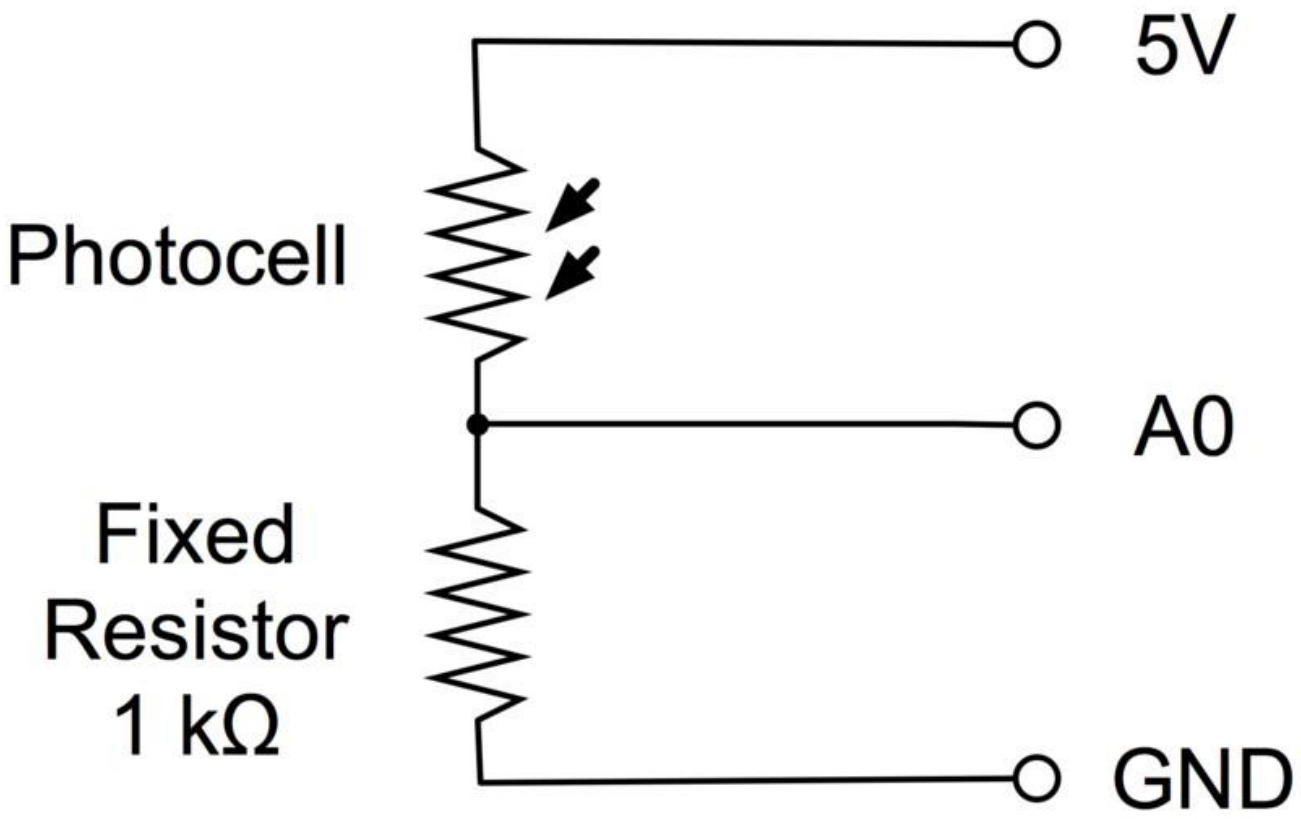
Introduzione al componente

Fotocellula:

La fotocellula utilizzata è un tipo di resistore dipendente dalla luce, spesso abbreviato con LDR. Come suggerisce il nome questi componenti agiscono similmente a dei resistori, fatta eccezione per il fatto che la resistenza cambia in base alla quantità di luce che la illumina.

La fotocellula che utilizzeremo ha una resistenza di circa 50 kΩ al buio e circa 500 Ω quando sottoposta ad alta luminosità. Per convertire questo valore di resistenza variabile in qualcosa che possiamo misurare con un input analogico nella nostra scheda UNO R3 misureremo il voltaggio.

Il modo più semplice di fare questa cosa è combinare il fotoresitore con un resistore fisso.



Il resistore e la fotocellula insieme agiscono come un potenziometro. Quando la luce è intensa la resistenza del fotoresistore è molto bassa se comparata con il valore della resistenza fissa, in questo caso è come se il potenziometro fosse girato verso il massimo.

Quando la fotocellula è sottoposta ad una luce molto bassa o nulla, la resistenza sarà molto più alta rispetto alla seconda resistenza fissa da 1 kΩ, così sarà come se il potenziometro sia girato verso la messa a terra GND.

Carica il codice che ti abbiamo fornito e prova a coprire la fotocellula con il dito, dopodiché avvicinala ad una sorgente di luce.

Connessione

Schema

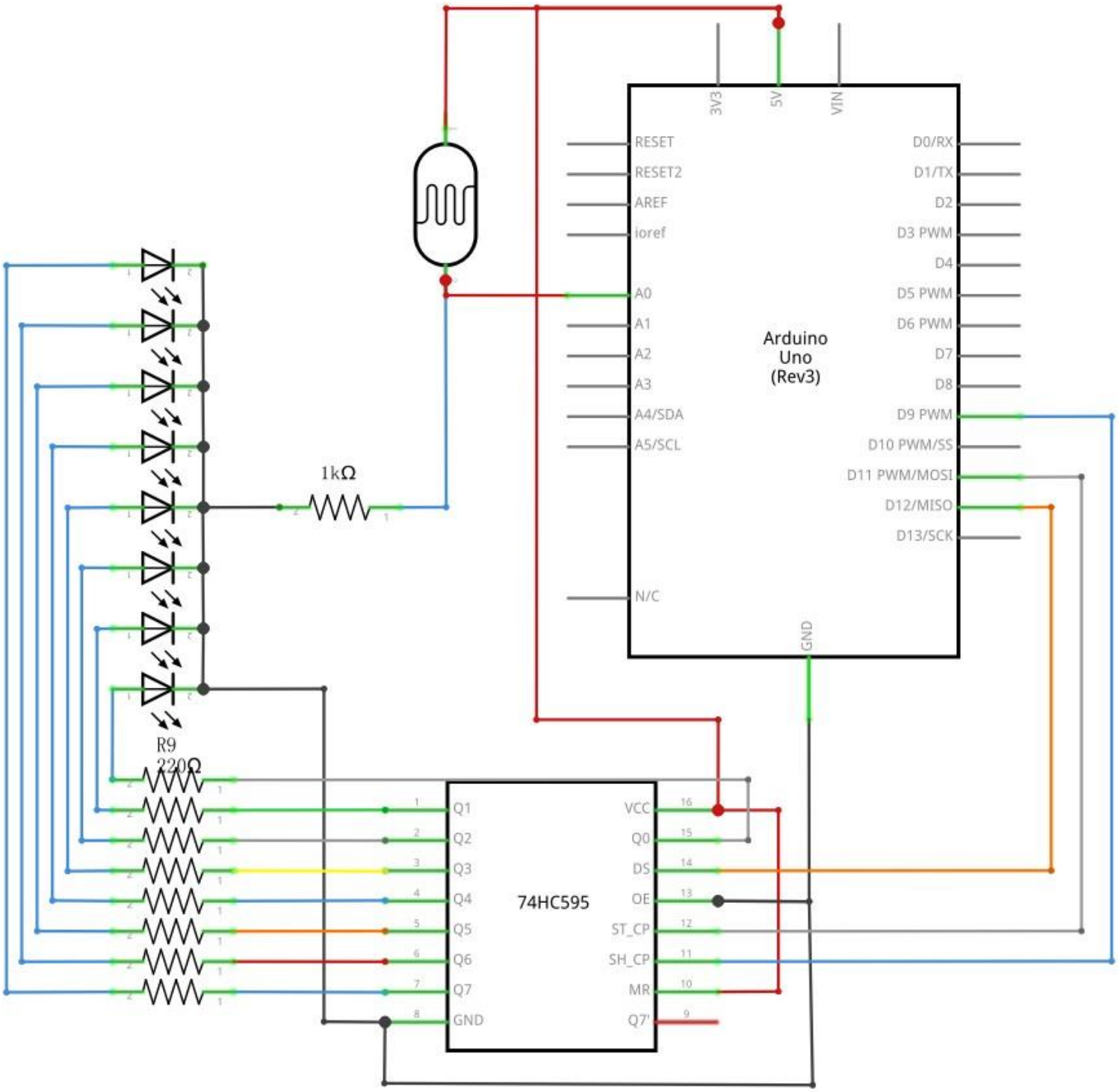
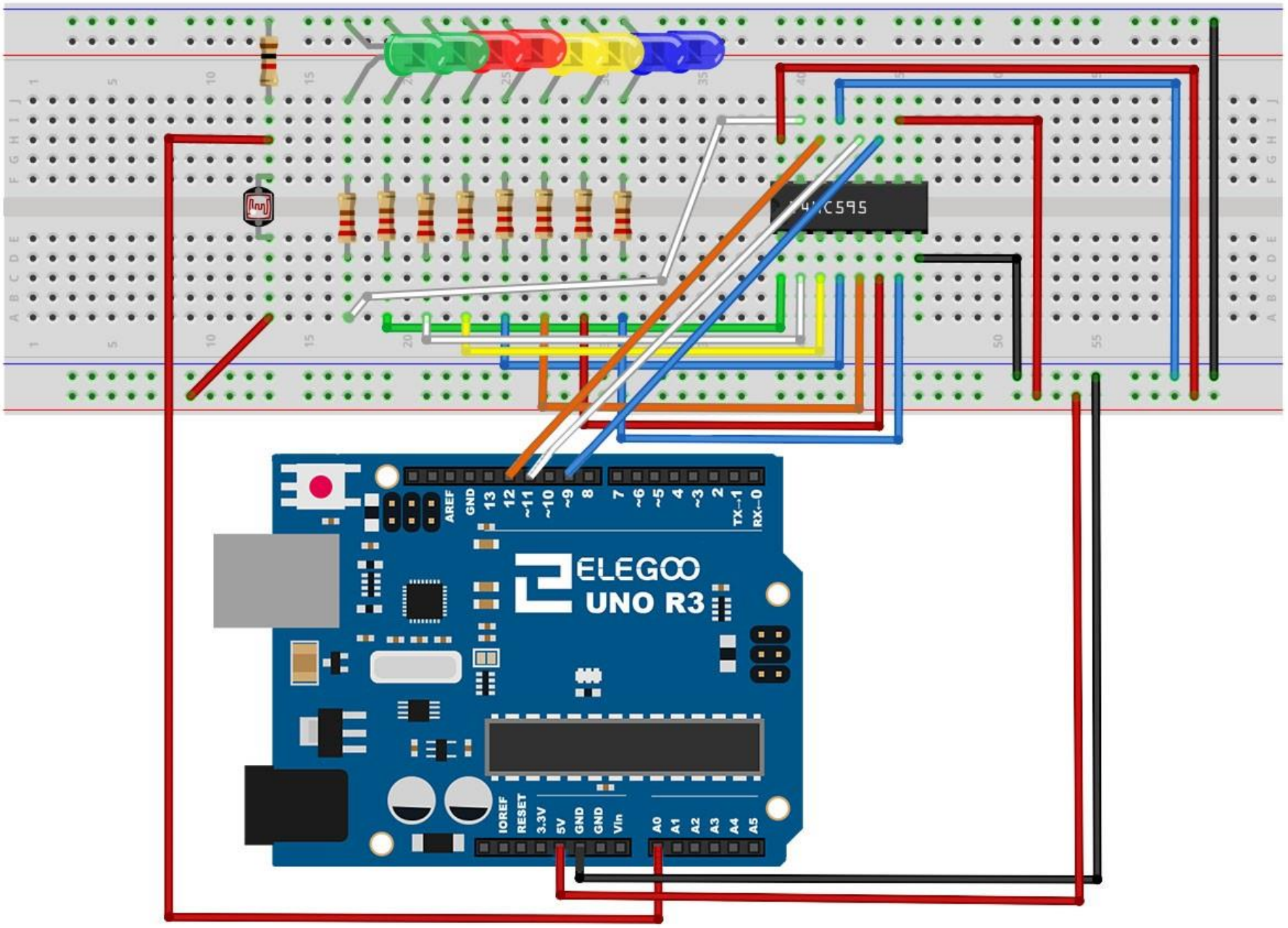


Diagramma di collegamento



Code

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 26 Photocell”, clicca su UPLOAD per caricare il programma.
Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

La prima cosa da notare è che abbiamo cambiato il nome del pin analogico chiamandolo ‘lightPin’ al posto di ‘potPin’ dato che non avremo un potenziometro collegato. L’unico cambiamento sostanziale al codice è la linea che calcola quanti led vengono accesi.

```
int numLEDSLit = reading / 57;    // all LEDs lit at 1k
```

Questa volta, dividiamo il valore grezzo letto per 57 invece che per 114. In altre parole lo dividiamo per la metà rispetto a quanto abbiamo fatto con il potenziometro, vogliamo infatti dividere per nove zone il range di valori, il primo corrisponderà a tutti i led spenti e l’ultimo corrisponderà a tutti i led illuminati. Questo fattore è dovuto alla resistenza fissa da 1 kΩ. Questo perché quando la fotocellula ha una resistenza pari a 1 kΩ (la stessa del resistore fisso), la lettura grezza sarà 1023 / 2 = 511. Questo significa che tutti i led vengono accesi e quindi numLEDSlit sarà pari a 8.

Foto d’esempio