



**IL TUTORIAL PER LO STARTER KIT PIÙ  
COMPLETO PER UNO**

**V1.0.17.7.10**

# 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>

Negozi Amazon Stati Uniti: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1>

Negozi Amazon CA: <http://www.amazon.ca/shops/A2WWHQ25ENKVJ1>

Negozi Amazon UK: <http://www.amazon.co.uk/shops/A1780XYQ9DFQM6>

Negozi Amazon DE: <http://www.amazon.de/shops/A1780XYQ9DFQM6>

Negozi Amazon FR: <http://www.amazon.de/shops/A1780XYQ9DFQM6>

Negozi Amazon ES: <http://www.amazon.de/shops/A1780XYQ9DFQM6>

Negozi Amazon IT: <http://www.amazon.de/shops/A1780XYQ9DFQM6>

## **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](mailto:service@elegoo.com) o [EUservice@elegoo.com](mailto: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

Servo Motor  
(SG90)  
1PC



ULN2003 Stepper  
Motor Driver  
Board  
1PC



Power Supply  
Module  
1PC



GY-521  
Module  
1PC



Elegoo UNO R3  
Controller  
Board  
1PC



LCD 1602 Module  
(with pin header)  
1PC



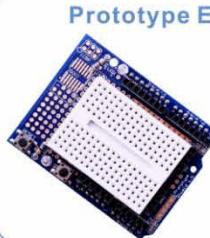
Rc522  
RFID Module  
1PC



Stepper Motor  
1PC

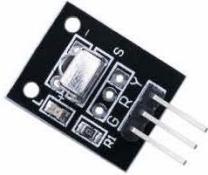


Prototype Expansion  
Board  
1PC



Contact us : service@elegoo.com

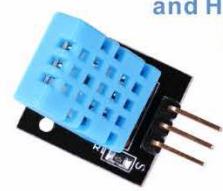
IR Receiver  
Module  
1PC



Joystick Module  
1PC



DHT11 Temperature  
and Humidity  
Module  
1PC



Rotary Encoder  
Module  
1PC



Ultrasonic Sensor  
1PC



Ds3231 RTC  
Module  
1PC



HC-SR501 PIR  
Motion Sensor  
1PC



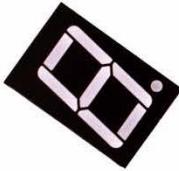
Sound Sensor  
Module  
1PC



Water Lever Detection  
Sensor Module  
1PC



1 Digit 7-Segment  
Display  
1PC



4 Digit 7-Segment  
Display  
1PC



74HC595  
1PC



Potentiometer  
(10K)  
2PCS



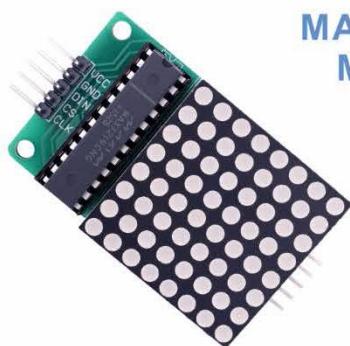
L293D  
1PC



Remote  
1PC



MAX7219  
Module  
1PC

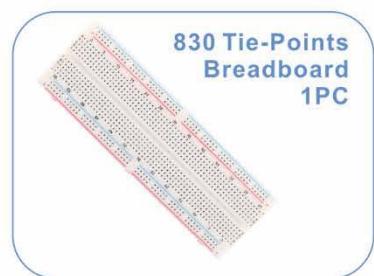


Active Buzzer  
1PC



Passive Buzzer  
1PC





Electrolytic Capacitor  
(100uF 50V)  
2PCS



Electrolytic Capacitor  
(10uF 50V)  
2PCS



NPN Transistor  
(S8050)  
5PCS



## Resistor 120PCS



Button (Small)  
5PCS



Thermistor  
1PC



Diode Rectifier  
5PCS



Tilt Ball Switch  
1PC





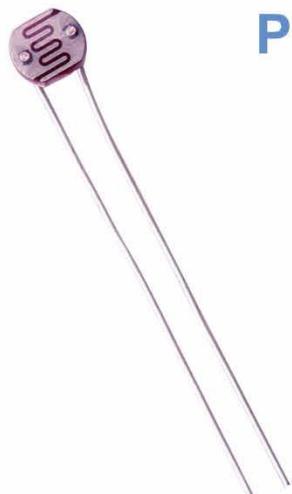
Red LED  
5PCS



Yellow LED  
5PCS



Blue LED  
5PCS



Photoresistor  
(Photocell)  
2PCS



RGB LED  
1PC



White LED  
5PCS



22pF Ceramic  
Capacitor  
5PCS



104pF Ceramic  
Capacitor  
5PCS

# Contenuti

<b>Lezione 0 Installazione IDE .....</b>	12
<b>Lezione 1 Aggiunta delle librerie e apertura della porta seriale .....</b>	23
<b>Lezione 2 Lampeggio (Blink) .....</b>	32
<b>Lezione 3 LED .....</b>	43
<b>Lezione 4 LED RGB.....</b>	50
<b>Lezione 5 Input Digitali .....</b>	59
<b>Lezione 6 Cicalino Attivo (Active Buzzer).....</b>	64
<b>Lezione 7 Cicalino Passivo (Passive Buzzer).....</b>	68
<b>Lezione 8 Interruttore ad Inclinazione .....</b>	72
<b>Lezione 9 Servo .....</b>	76
<b>Lezione 10 Modulo Sensore ad Ultrasuoni .....</b>	80
<b>Lezione 11 Modulo Switch a Membrana.....</b>	85
<b>Lezione 12 Sensore di Temperatura e umidità DHT11 .....</b>	91
<b>Lezione 13 Modulo Joystick Analogico.....</b>	97
<b>Lezione 14 Modulo di Ricezione IR.....</b>	102
<b>Lezione 15 Modulo Display 8x8 MAX7219 LED Dot Matrix.....</b>	108
<b>Lezione 16 Modulo GY-521.....</b>	112
<b>Lezione 17 Sensore di Movimento HC-SR501 .....</b>	121
<b>Lezione 18 Modulo Sensore del livello dell'Acqua .....</b>	131
<b>Lezione 19 Modulo Orologio a tempo reale (Real Time Clock) .....</b>	136
<b>Lezione 20 Modulo Sensore del Suono .....</b>	141
<b>Lezione 21 Modulo RFID RC522.....</b>	147
<b>Lezione 22 Display LCD .....</b>	152
<b>Lezione 23 Termometro.....</b>	157
<b>Lezione 24 Otto LED con 74HC595 .....</b>	162
<b>Lezione 25 Il Monitor Seriale.....</b>	170
<b>Lezione 26 Fotocellula .....</b>	176
<b>Lezione 27 Display a segmenti e 74HC595 .....</b>	182

<b>Lezione 28 Display a sette segmenti con quattro cifre.....</b>	<b>188</b>
<b>Lezione 29 Motore DC .....</b>	<b>193</b>
<b>Lezione 30 Relè .....</b>	<b>203</b>
<b>Lezione 31 Motore Passo-Passo .....</b>	<b>208</b>
<b>Lezione 32 Controllare il motore passo-passo da remoto.....</b>	<b>216</b>
<b>Lezione 33 Controllare il motore passo-passo con un encoder.....</b>	<b>220</b>

# Lezione 0 Installazione IDE

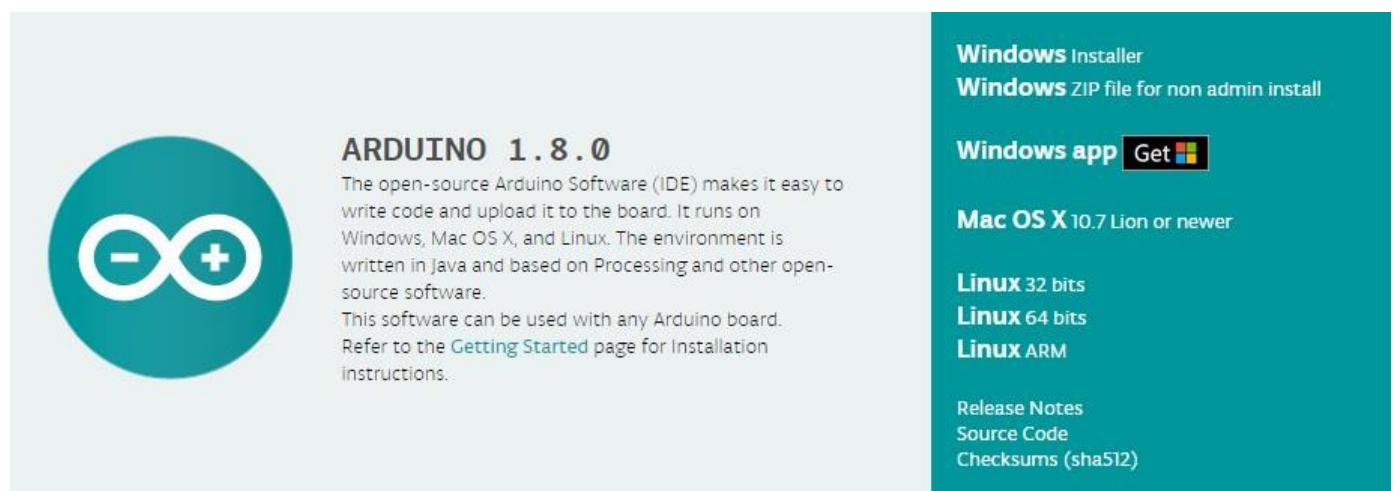
## Introduzione

"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.

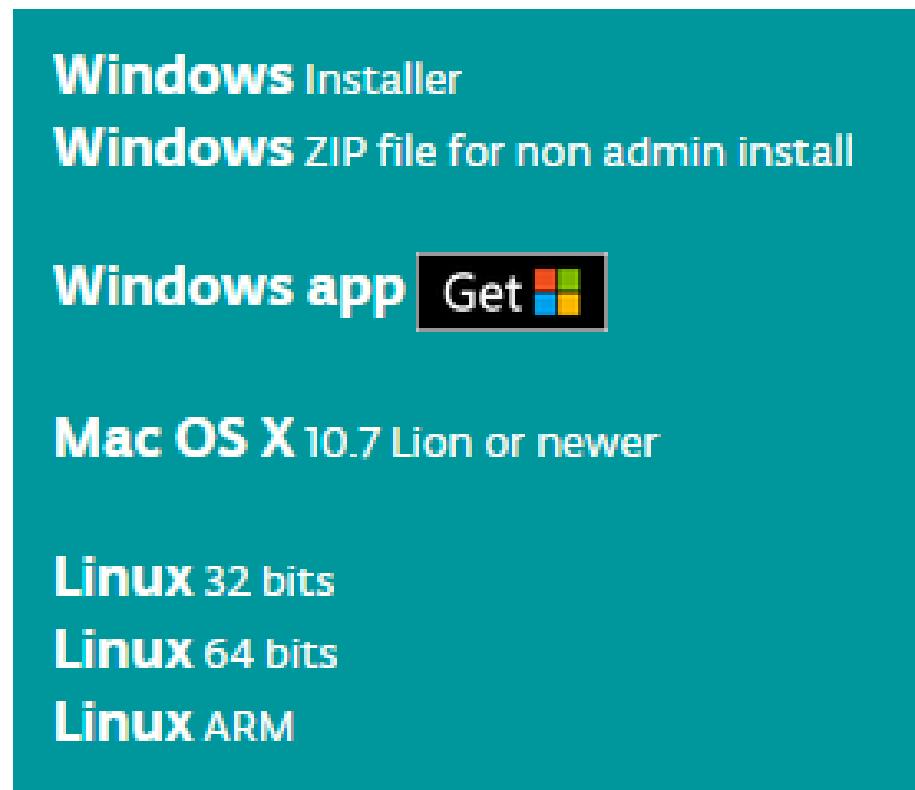
Il software di Arduino che verrà utilizzato per programmare la piattaforma Arduino è disponibile per Windows, Mac e Linux. Il processo di installazione è diverso per le tre piattaforme e purtroppo necessita di una certa quantità di lavoro manuale per essere installato

**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*.

## Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



Premi su *JUST DOWNLOAD*.

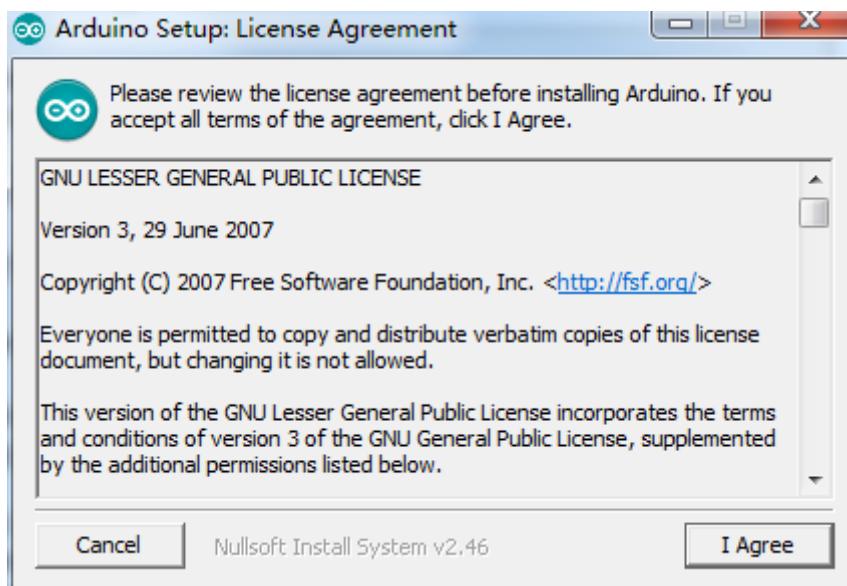
La versione 1.8.0 è disponibile anche nel materiale che ti abbiamo fornito.

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

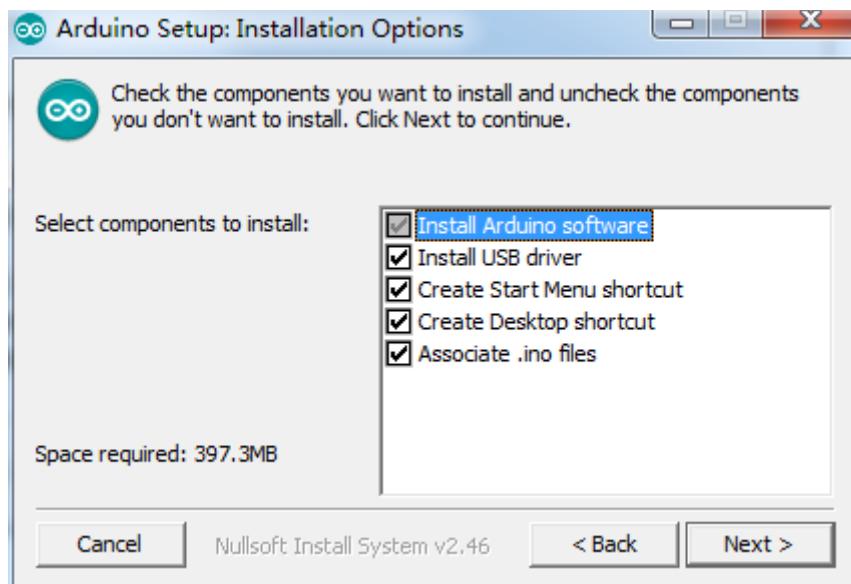
-  arduino-1.8.0-linux32.tar.xz
-  arduino-1.8.0-linux64.tar.xz
-  arduino-1.8.0-macosx.zip
-  arduino-1.8.0-windows.exe
-  arduino-1.8.0-windows.zip

## Installazione Arduino (Windows)

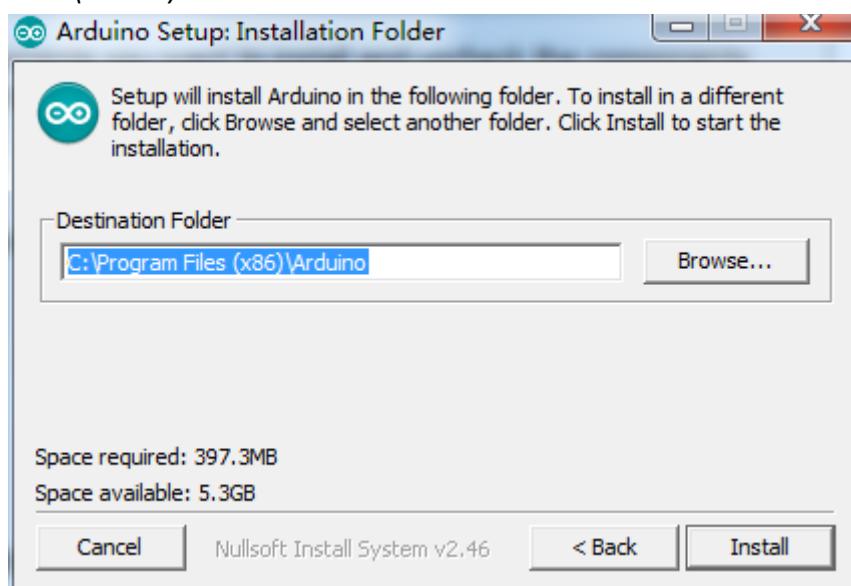
Installa Arduino con il pacchetto di installazione exe.



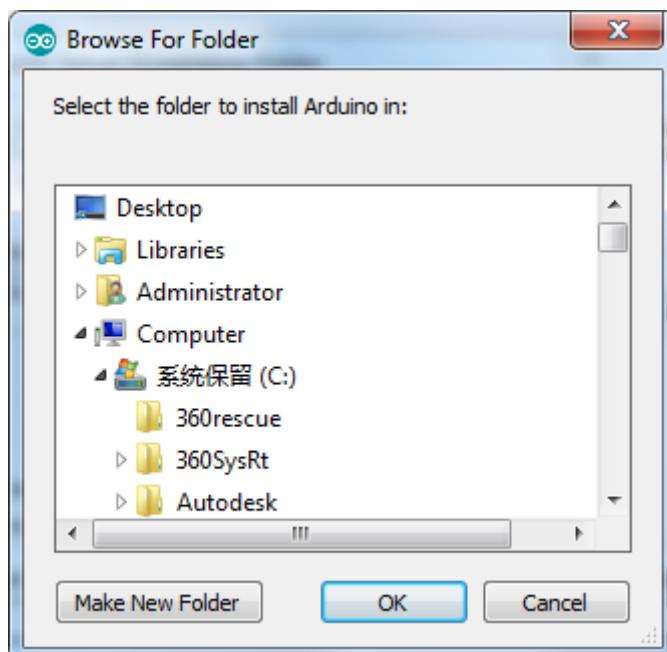
Premi su *I Agree*(Accetto) per proseguire con la prossima interfaccia.



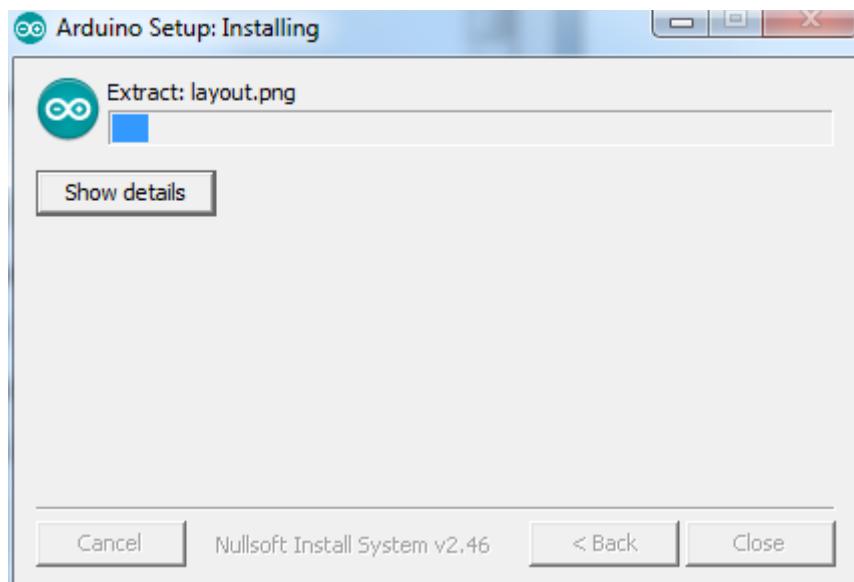
Premi su **Next(Avanti)**



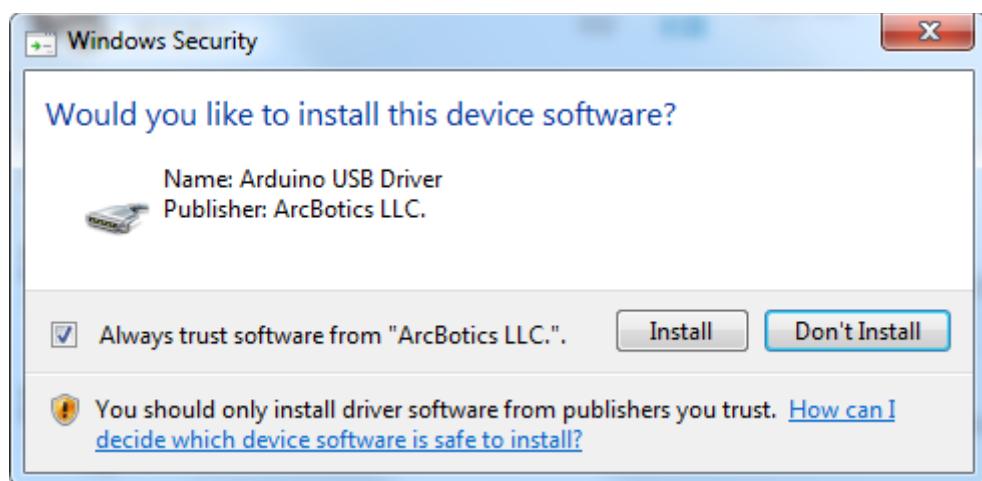
**Puoi premere su **Browse(Naviga)...** per scegliere il percorso di installazione o digitare direttamente la cartella che vuoi utilizzare.**



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



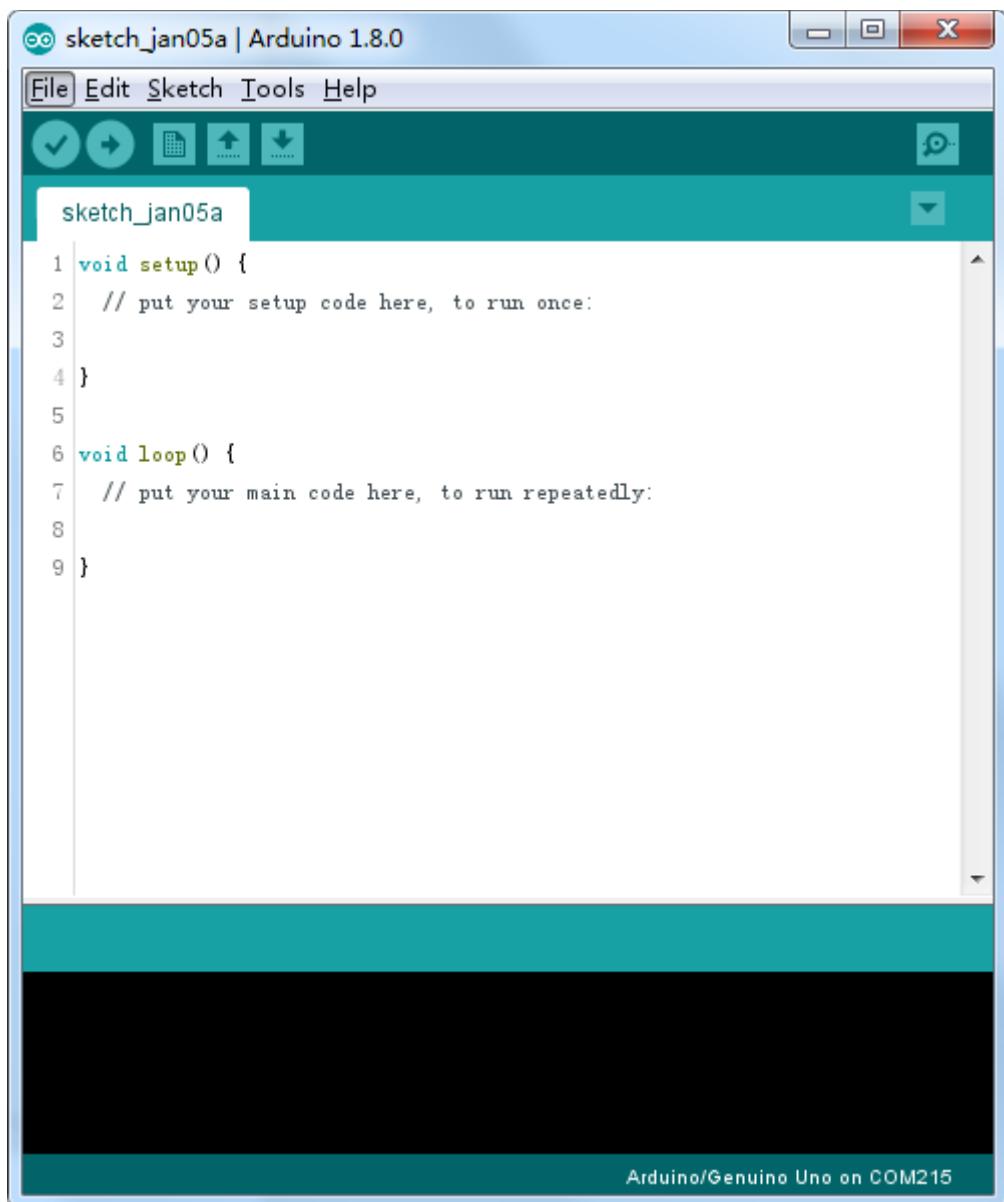
Al termine, apparirà la seguente interfaccia, premi su *Install(installa)* per avviare l'ultimo passaggio dell'installazione.



Successivamente, verrà visualizzata la seguente icona sul desktop.



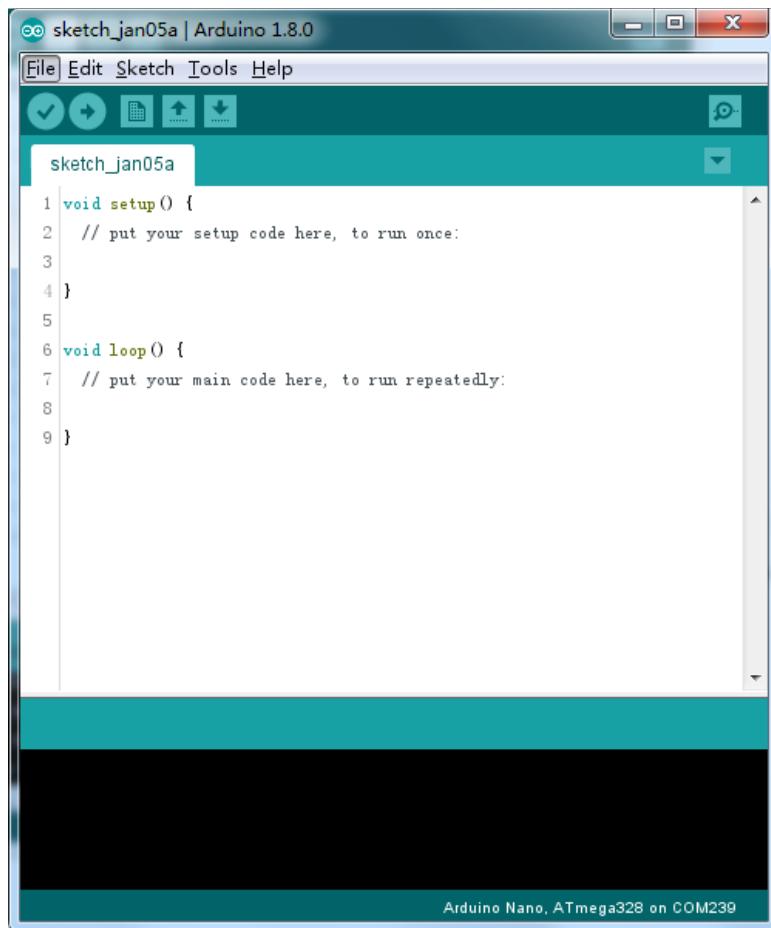
Fare doppio clic sull'icona per accedere all'ambiente di sviluppo.



Puoi scegliere direttamente il pacchetto di installazione e passare direttamente alla sezione successiva, 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.

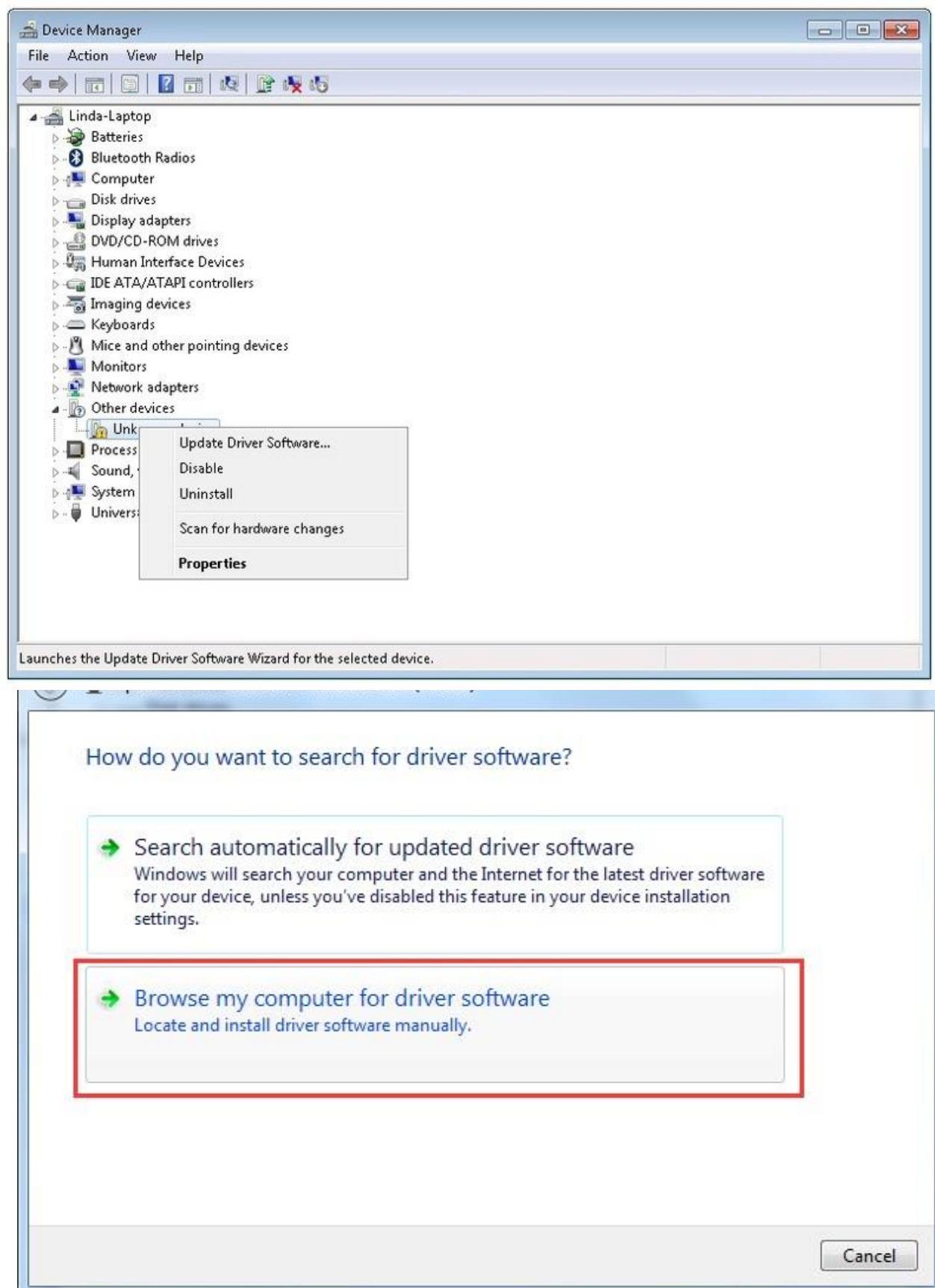
Prima di lanciare il software di Arduino, ricordati di installare i driver USB come spiegato nei passaggi seguenti:

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.**

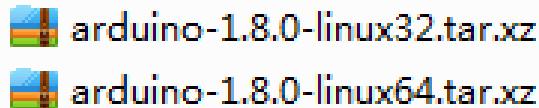
## Installazione Arduino (Mac OS X)

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.

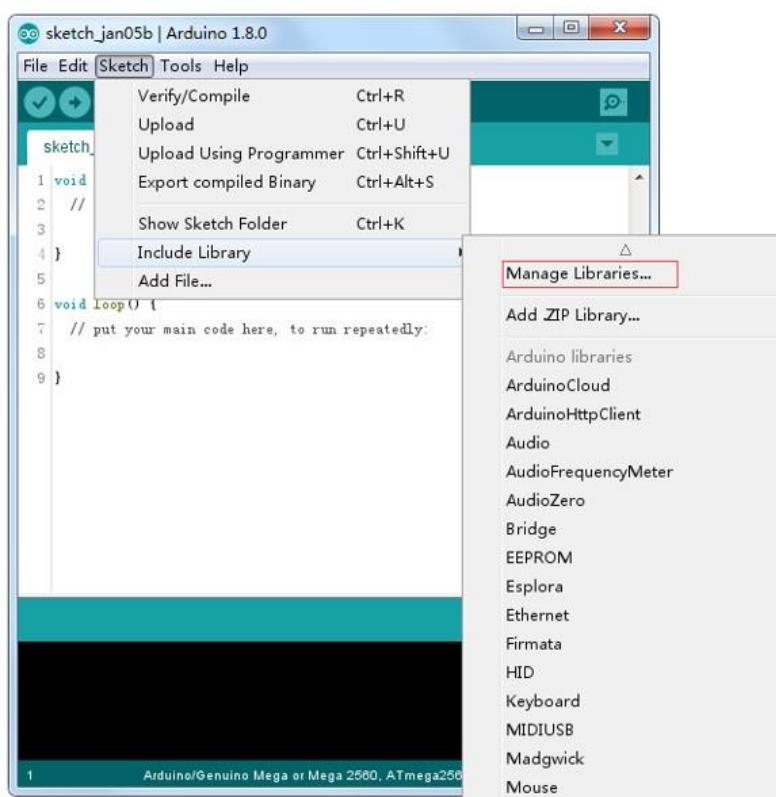
## Cosa sono le librerie?

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 dovrà 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".

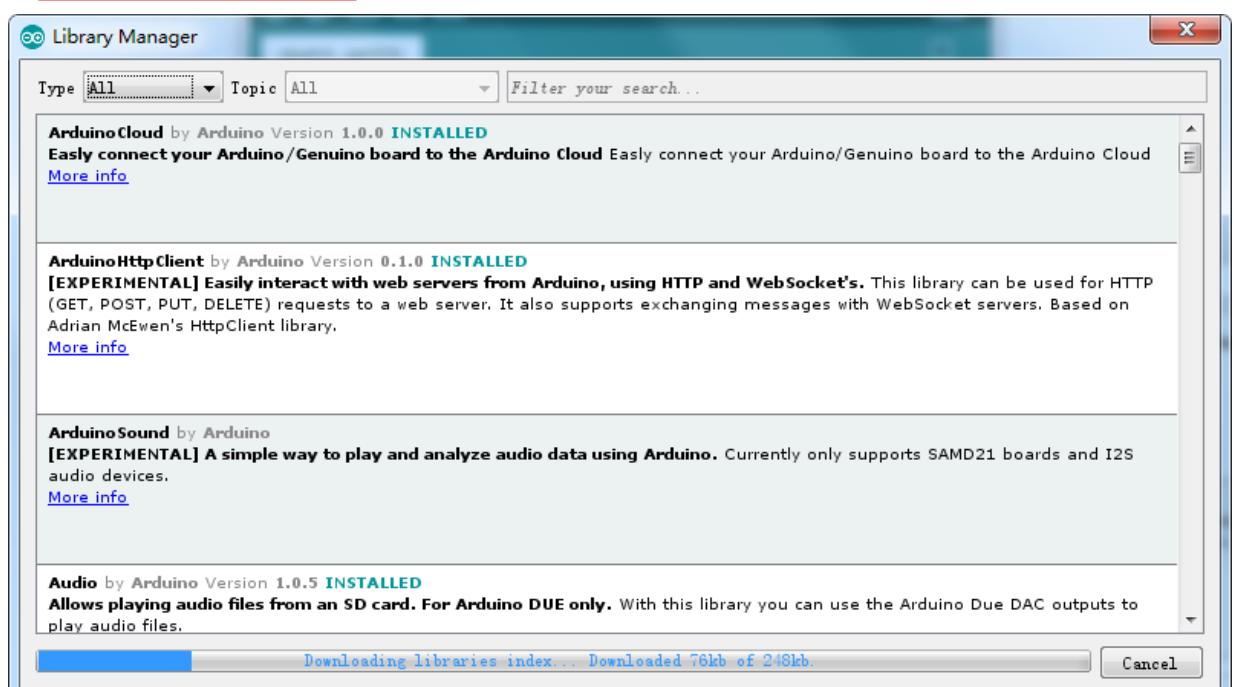
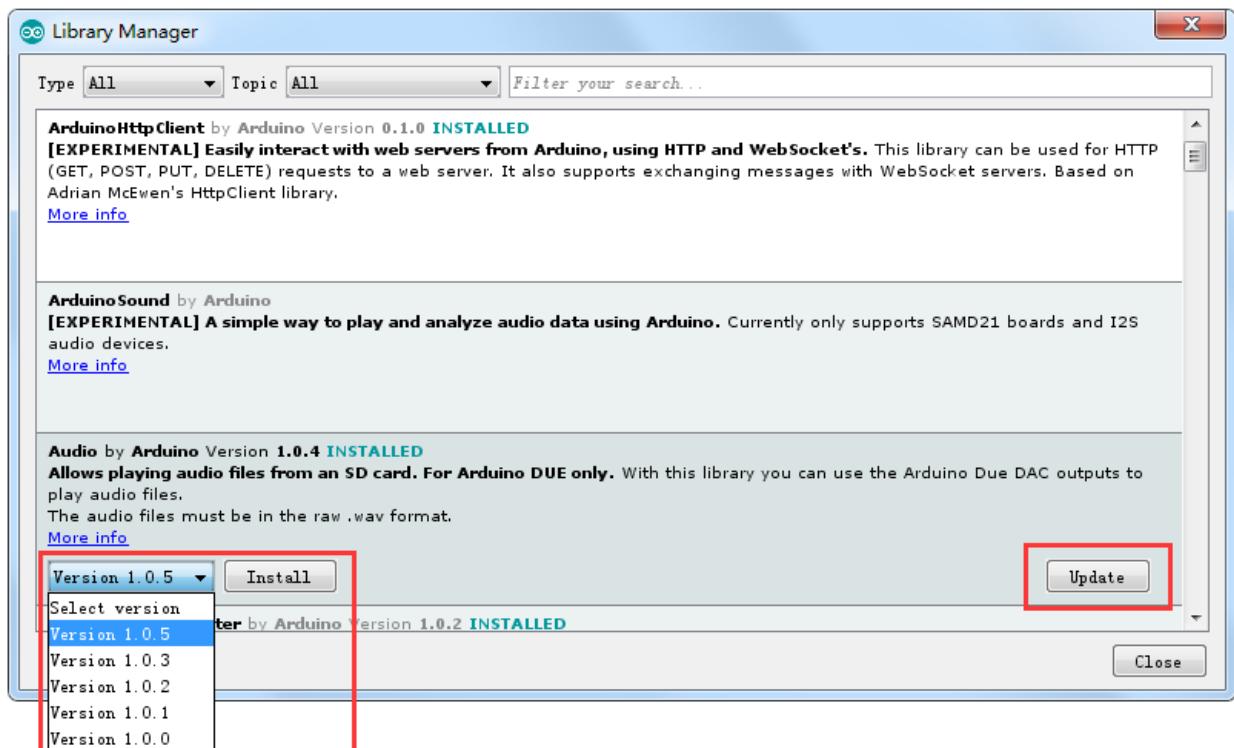


A questo punto si aprirà il gestore delle librerie e troverai un elenco di librerie già installate o pronte per l'installazione.

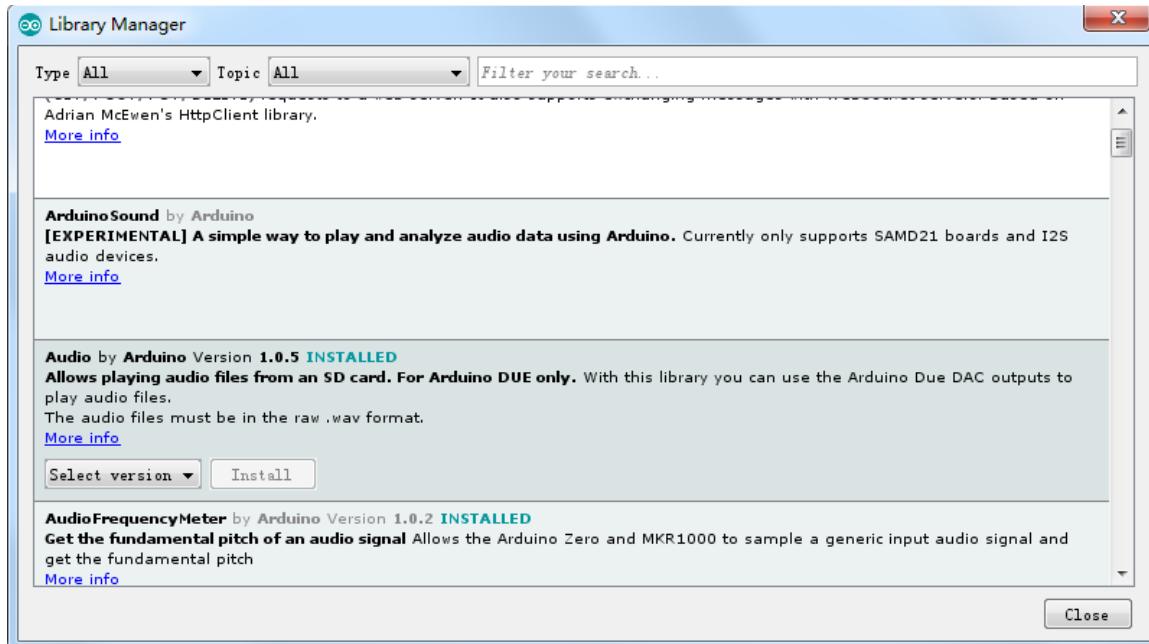
In questo esempio installeremo la libreria Bridge. Scorri l'elenco per trovarla, quindi seleziona la versione della libreria che vuoi installare. A volte solo una versione della libreria è disponibile. Se non viene visualizzato il menu di selezione della versione, non preoccuparti: è normale.

**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.

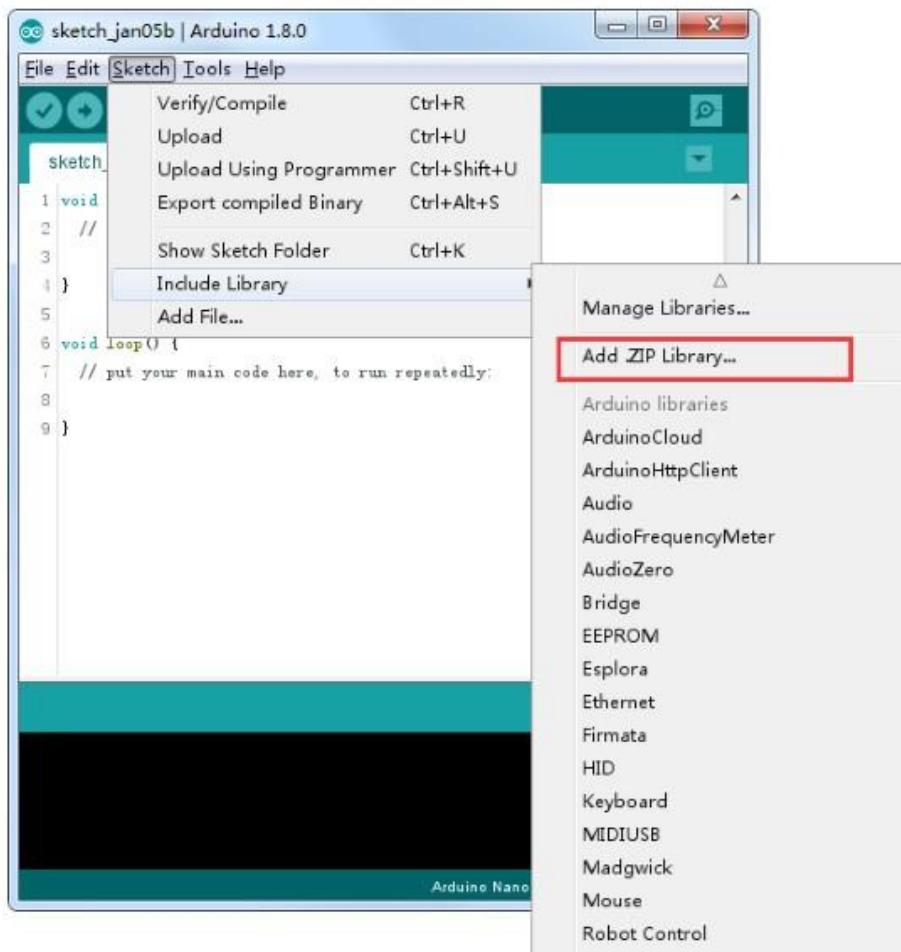


Ora ti sarà possibile trovare la nuova libreria disponibile nel menu: Library.

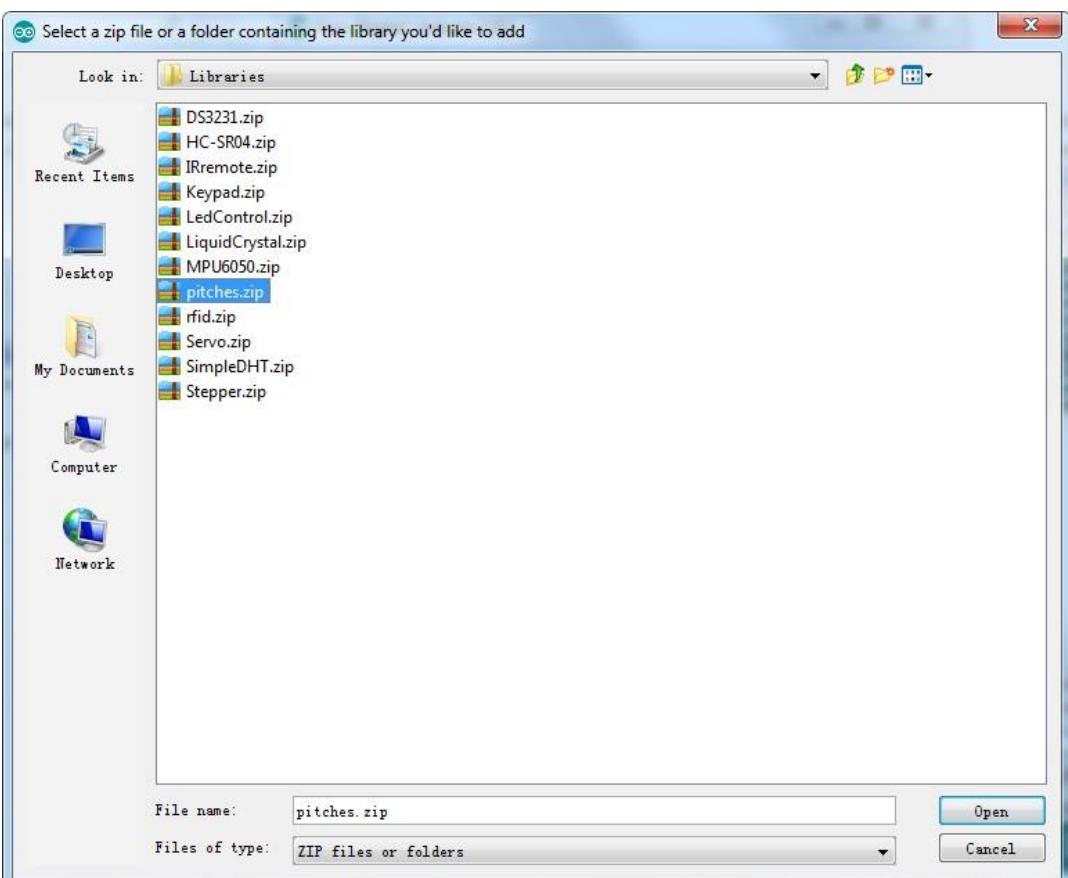
Se vuoi aggiungere una tua libreria all'elenco delle librerie disponibili in questa raccolta puoi fare una richiesta sull'apposita repository di [Github](#).

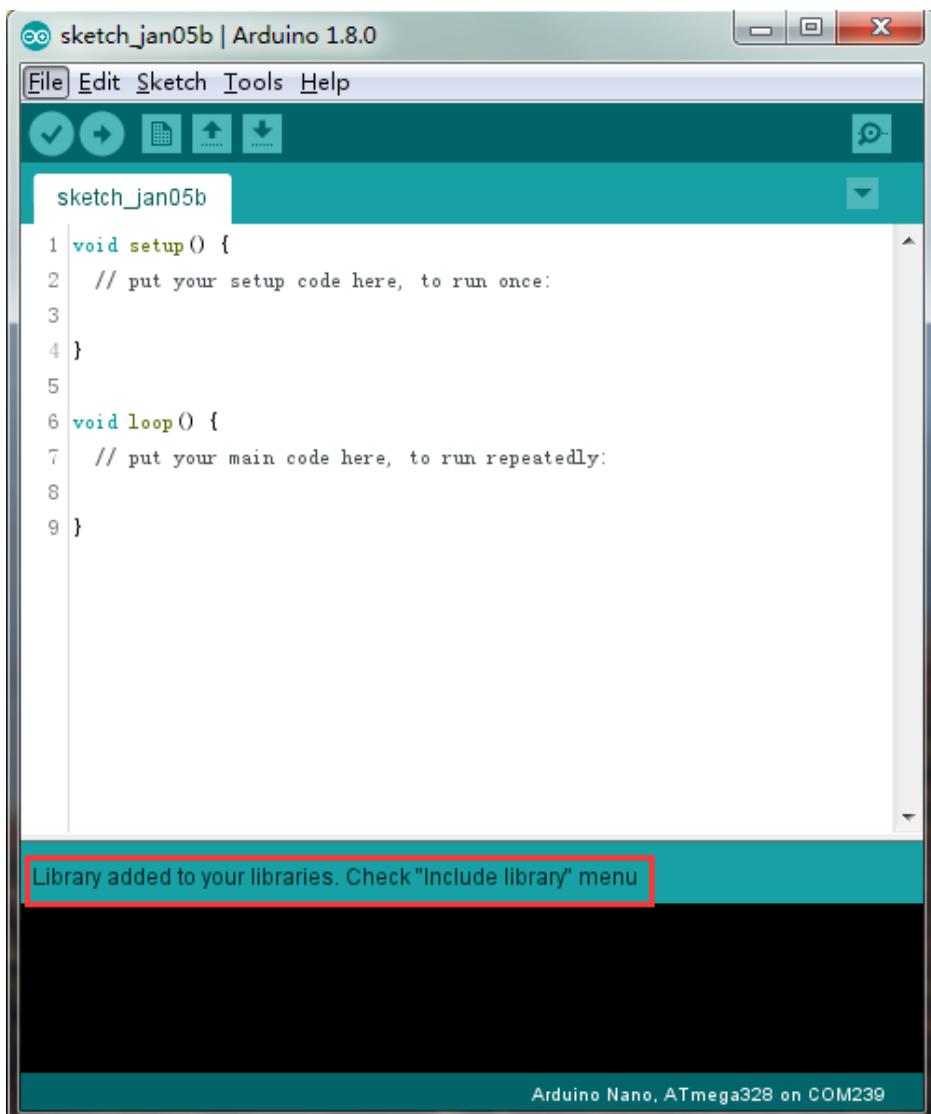
## Importare una libreria in formato .zip

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 dovrà 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!

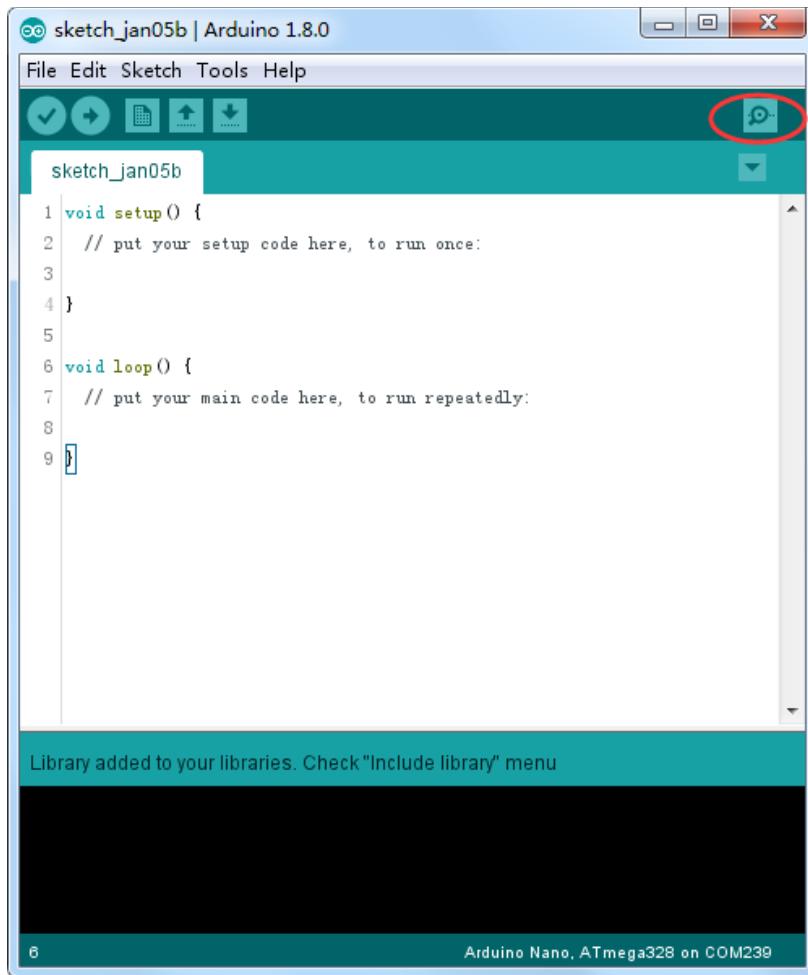
## Arduino Serial Monitor (Windows, Mac, Linux)

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.

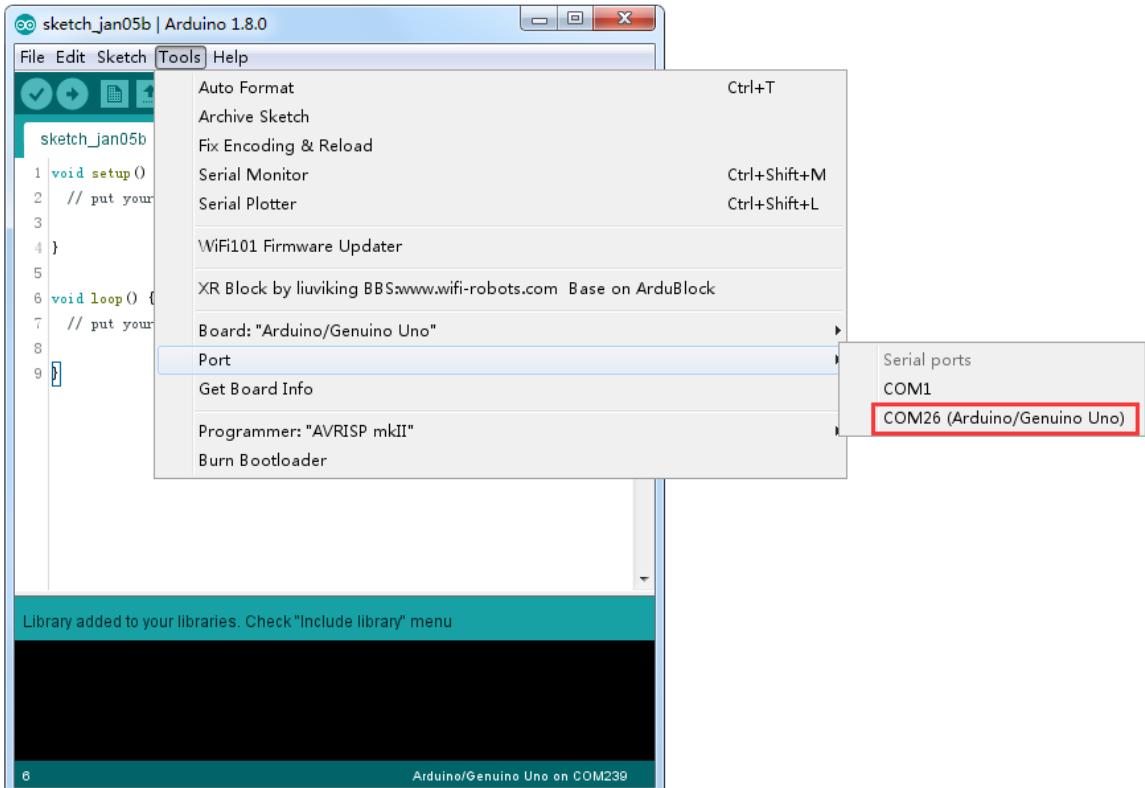
## Connettersi al Monitor Seriale

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.

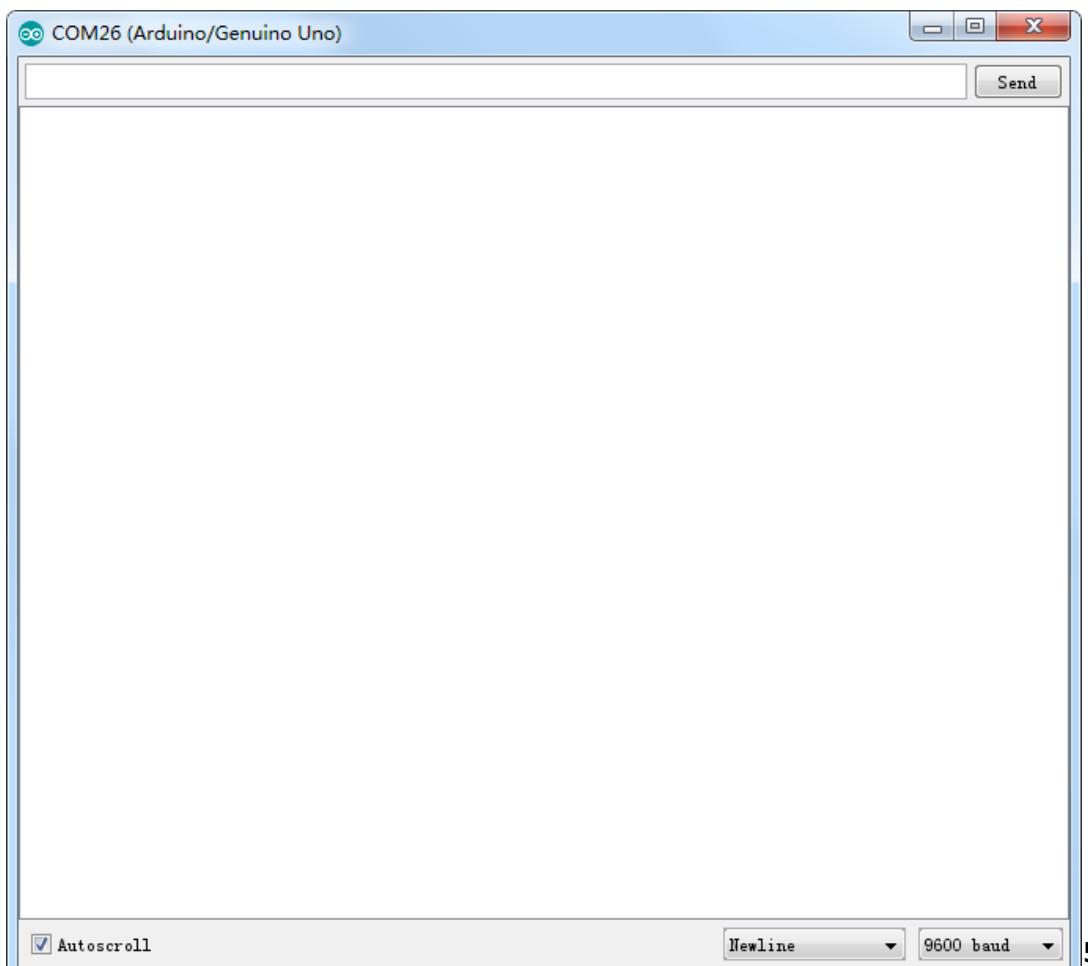


Nello stesso modo in cui si seleziona una porta per il caricamento di codice Arduino, si sceglie quale porta aprire nel Serial Monitor. Andare in Tools -> Serial Port, e selezionare la porta corretta.

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

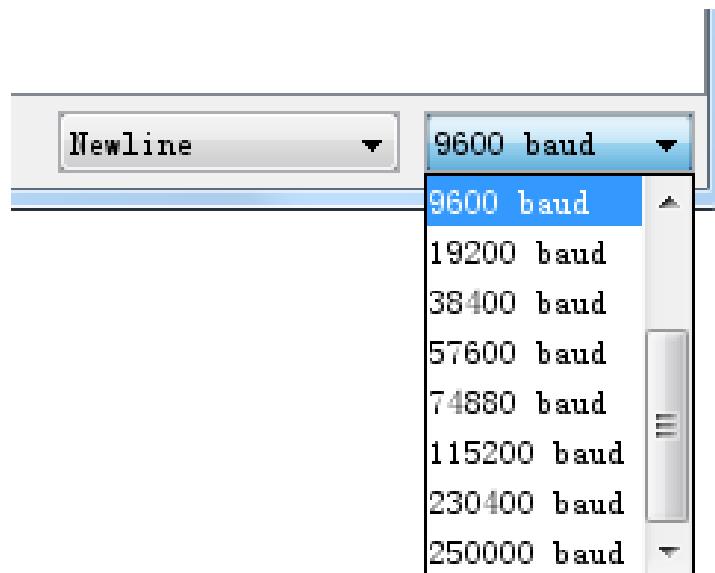


Una volta aperto dovresti vedere qualcosa di simile:

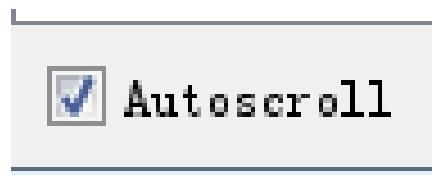


## Impostazioni

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 R3 per far lampegnare il LED già presente sulla scheda e come scaricare il programma in piccoli passi.

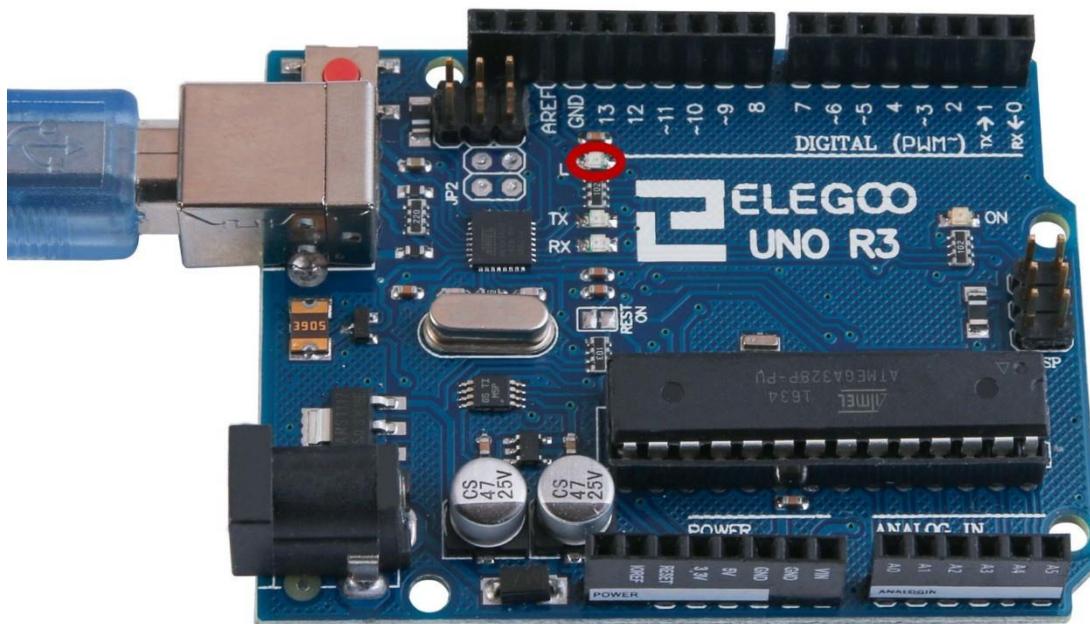
### Componenti Richiesti:

- (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.



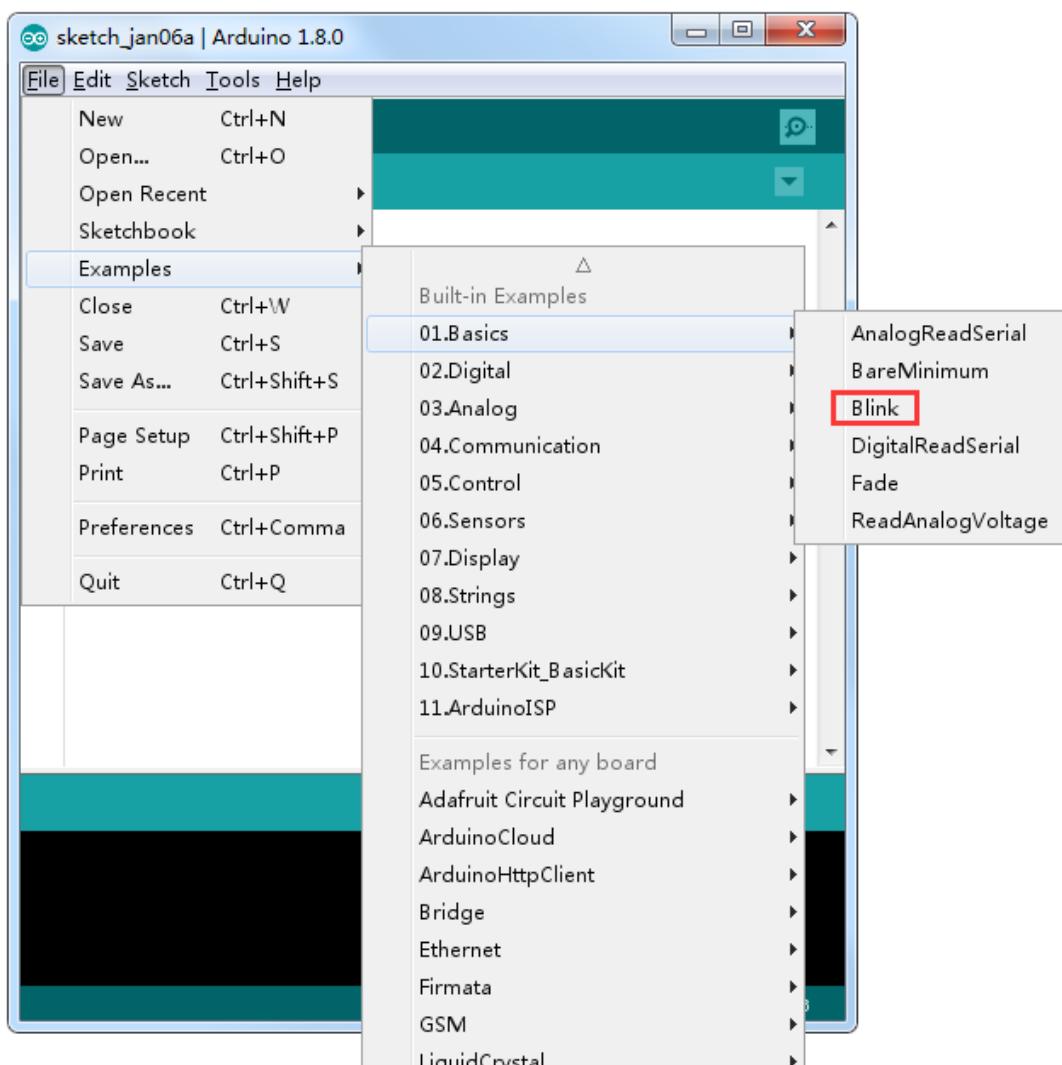
Potresti notare che la tua scheda UNO R3 abbia il led 'L' già lampegnante quando la colleghi ad una presa USB. Questo perché le schede sono in genere fornite con il codice 'Blink' pre-installato.

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

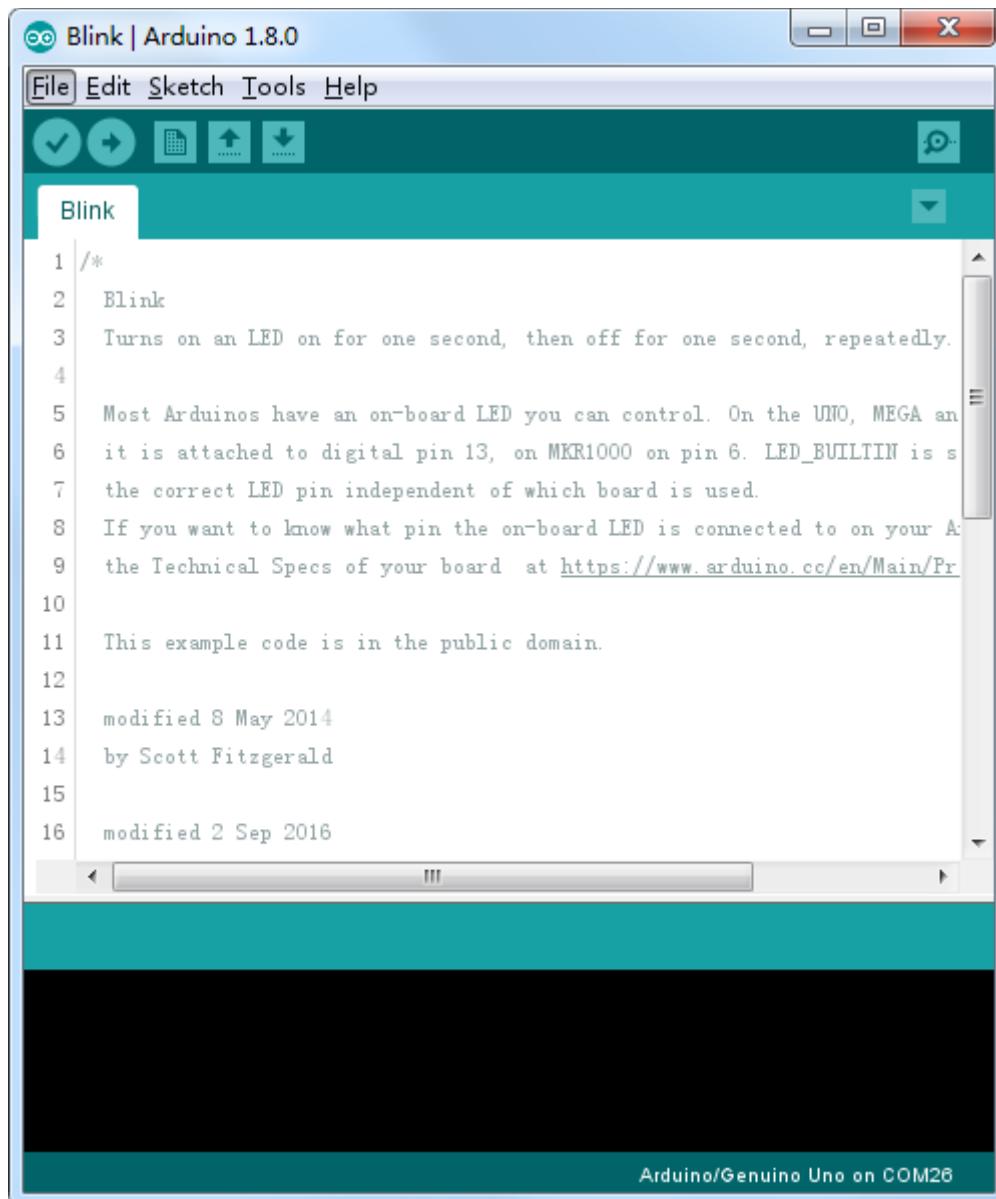
Nella lezione 0, hai impostato l'IDE di Arduino e hai fatto in modo di impostare la porta seriale giusta per poter collegare la tua scheda UNO R3. Ora è il momento di mettere alla prova il collegamento e programmare la scheda UNO R3.

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 menu dell'IDE in File> Esempi> 01.Basics



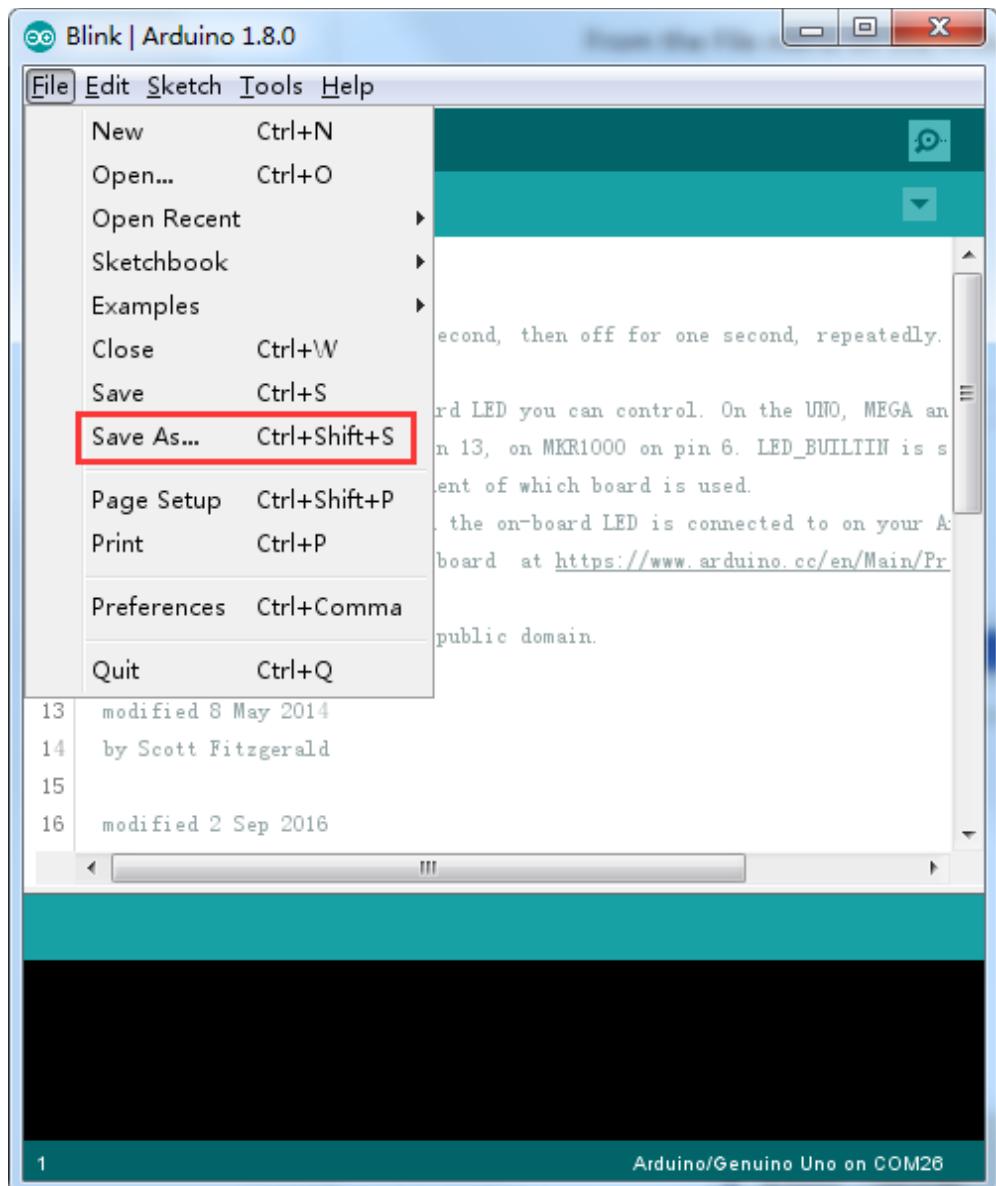
Quando si apre la finestra del codice, ingrandirla in modo che si possa vedere l'intero codice nella finestra.

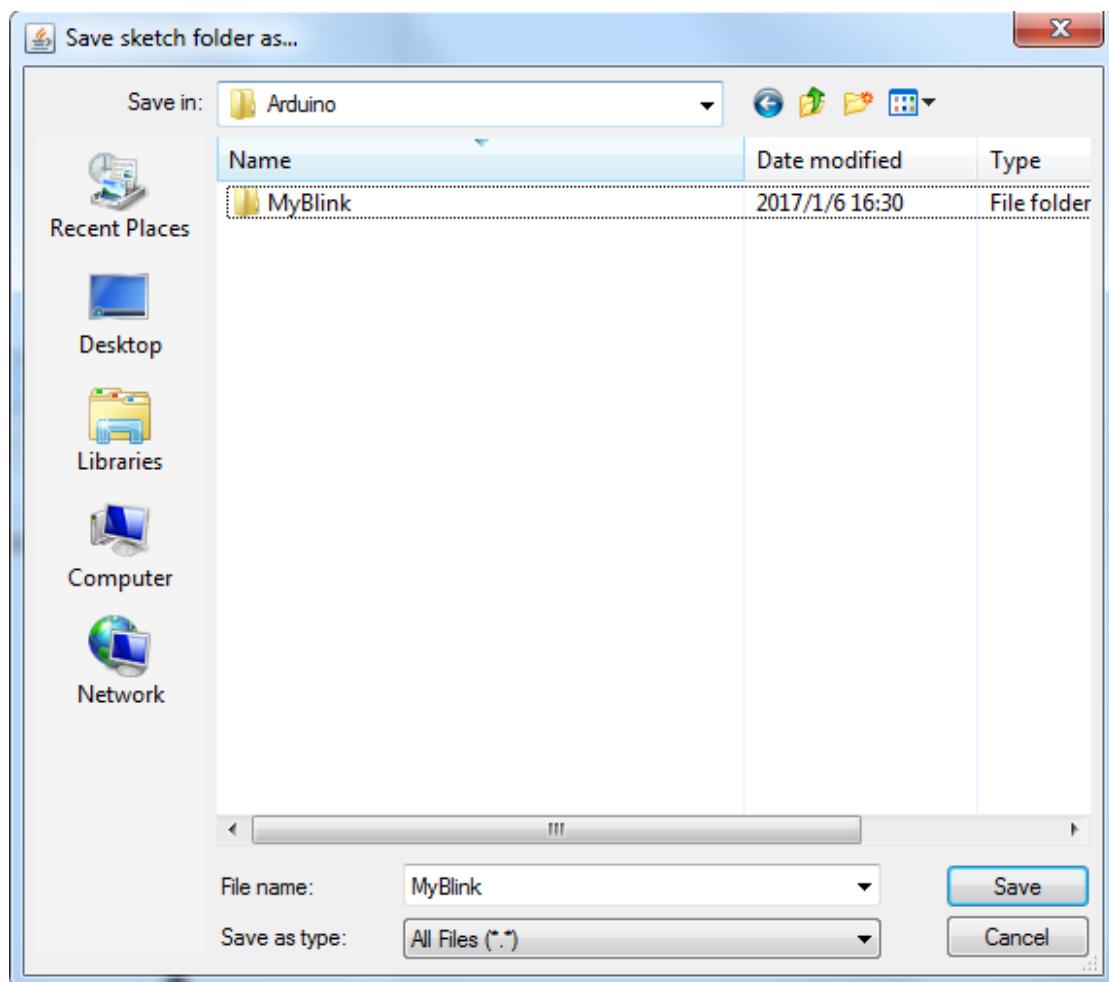


Gli esempi di codici inclusi con l'IDE di Arduino sono in 'sola lettura'. Cioè, è possibile caricarli su una scheda UNO R3, ma se li si cambia, non è possibile salvarli come lo stesso file.

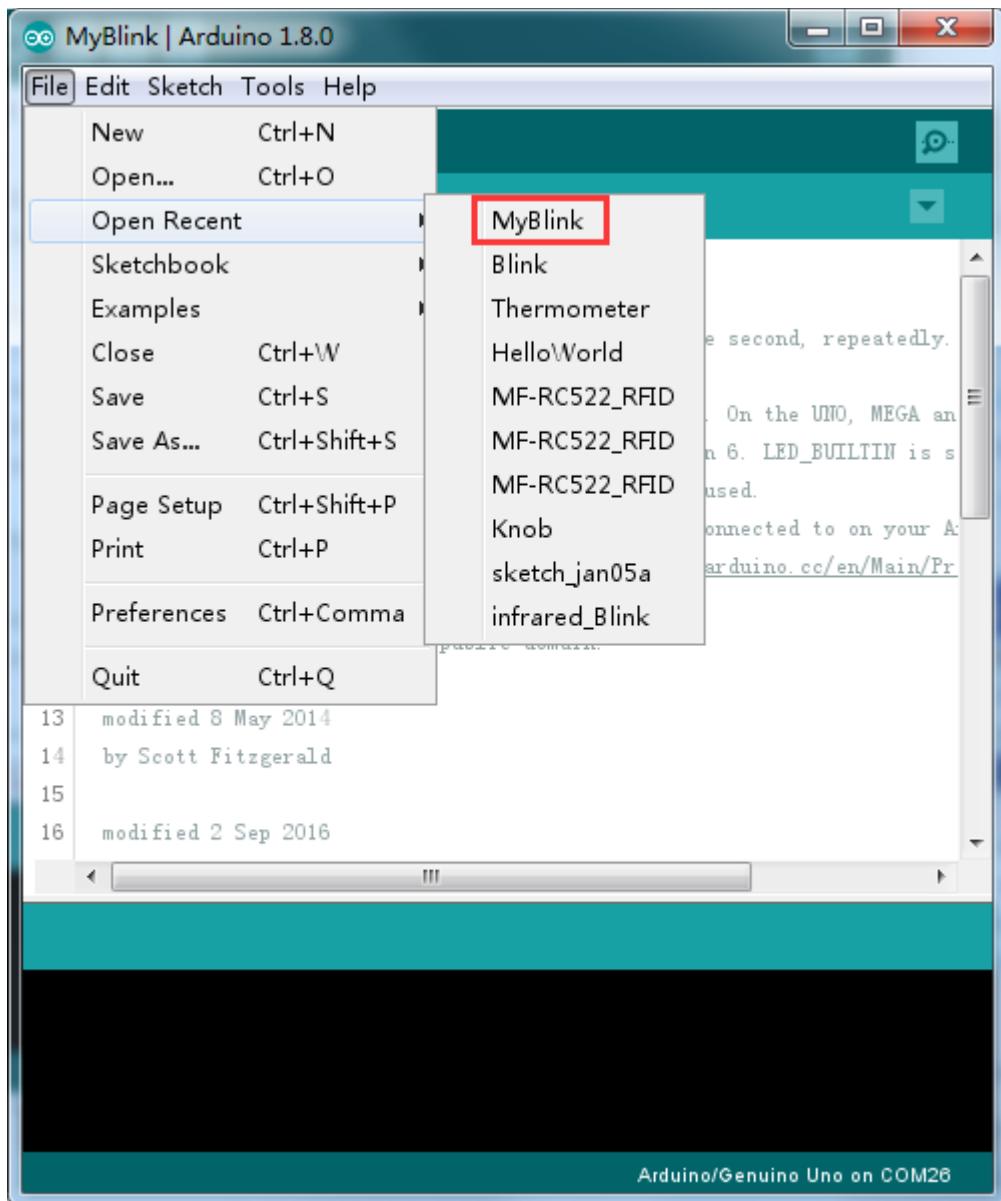
Dal momento che ci accingiamo a cambiare questo codice, la prima cosa che devi fare è salvare la tua copia di codice.

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

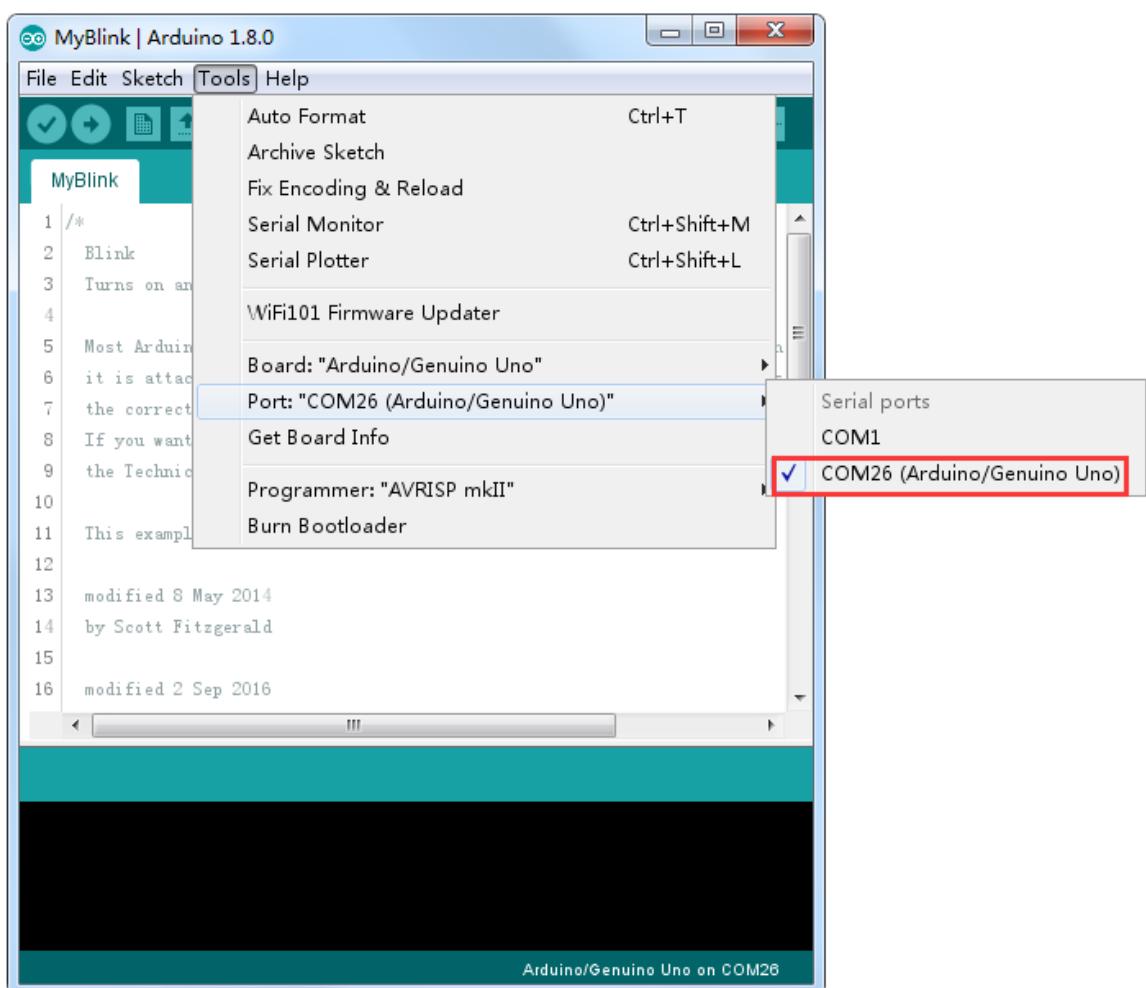
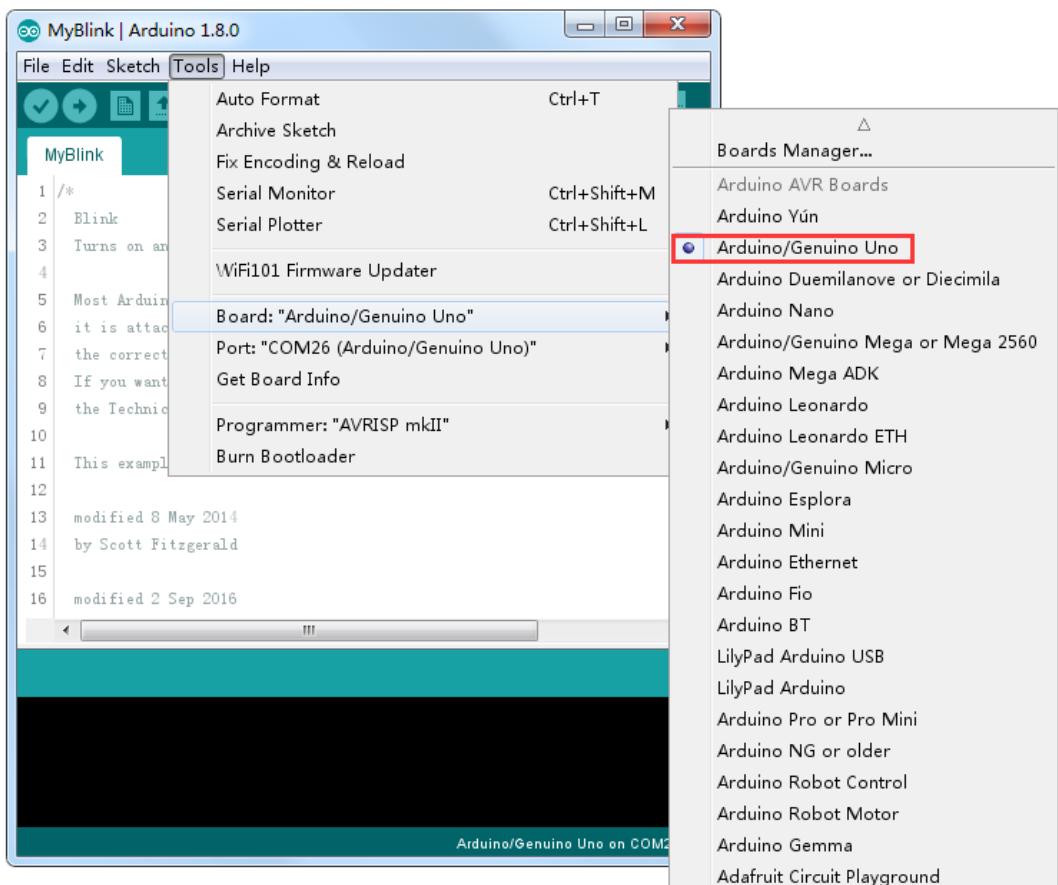




Ora hai salvato la tua copia di 'Blink' nella tua raccolta di codici. Questo significa che se tu volessi trovare il tuo codice potresti aprirlo cliccando su File > Open Recent.

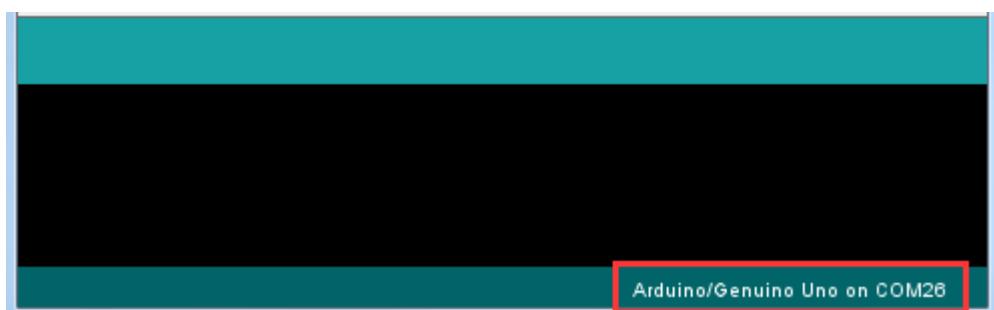


Collega la tua scheda Arduino al computer con il cavo USB e verifica che 'Board Type' e 'Serial Port' siano impostate correttamente, come da screen seguenti.



**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'. È 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.

Uploading... [Progress Bar]

```
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -I bin sketch.ihex
```

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

Arduino/Genuino Uno on COM26

Infine, lo stato passerà a 'Fatto'.

Done uploading.

```
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -I bin sketch.ihex
```

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

Arduino/Genuino Uno on COM26

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:

Problem uploading to board. See <http://www.arduino.cc/en/Troubleshooting/BoardUpload> [Copy error messages]

```
avrduude: st1500_recv(): programmer is not responding
avrduude: st1500_getsync() attempt 10 of 10: not in sync: resp=0x22
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload
```

Arduino/Genuino Uno on COM1

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

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 durante 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. È 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.

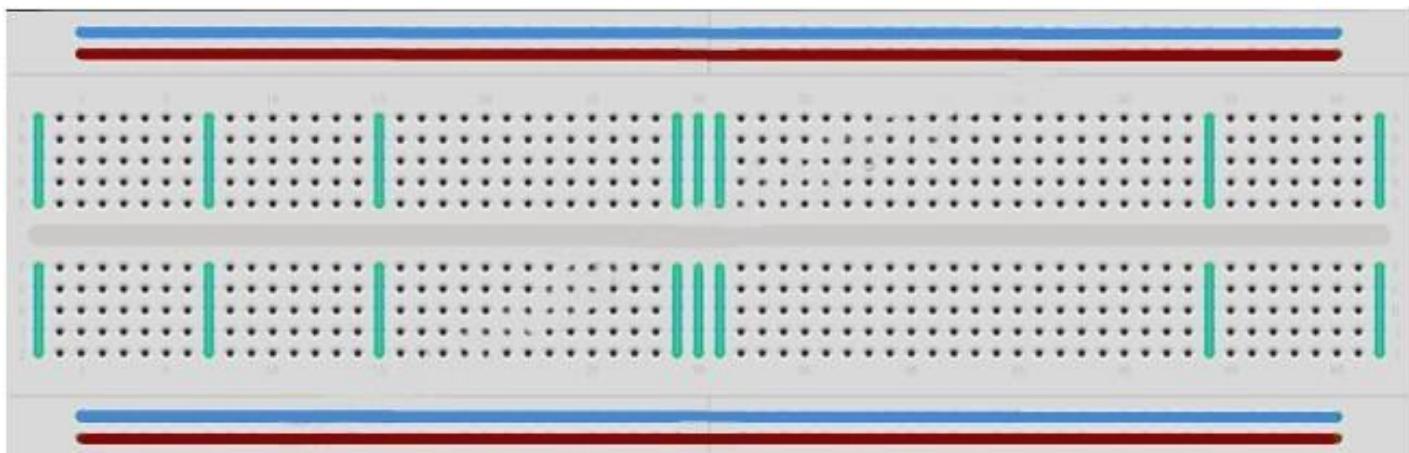
### Componenti Richiesti:

- (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 di plastica. All'interno sono presenti strisce di metallo che mettono in connessione differenti fori su tutte righe corte (quelle verdi 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 lì 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 3mm 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 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.

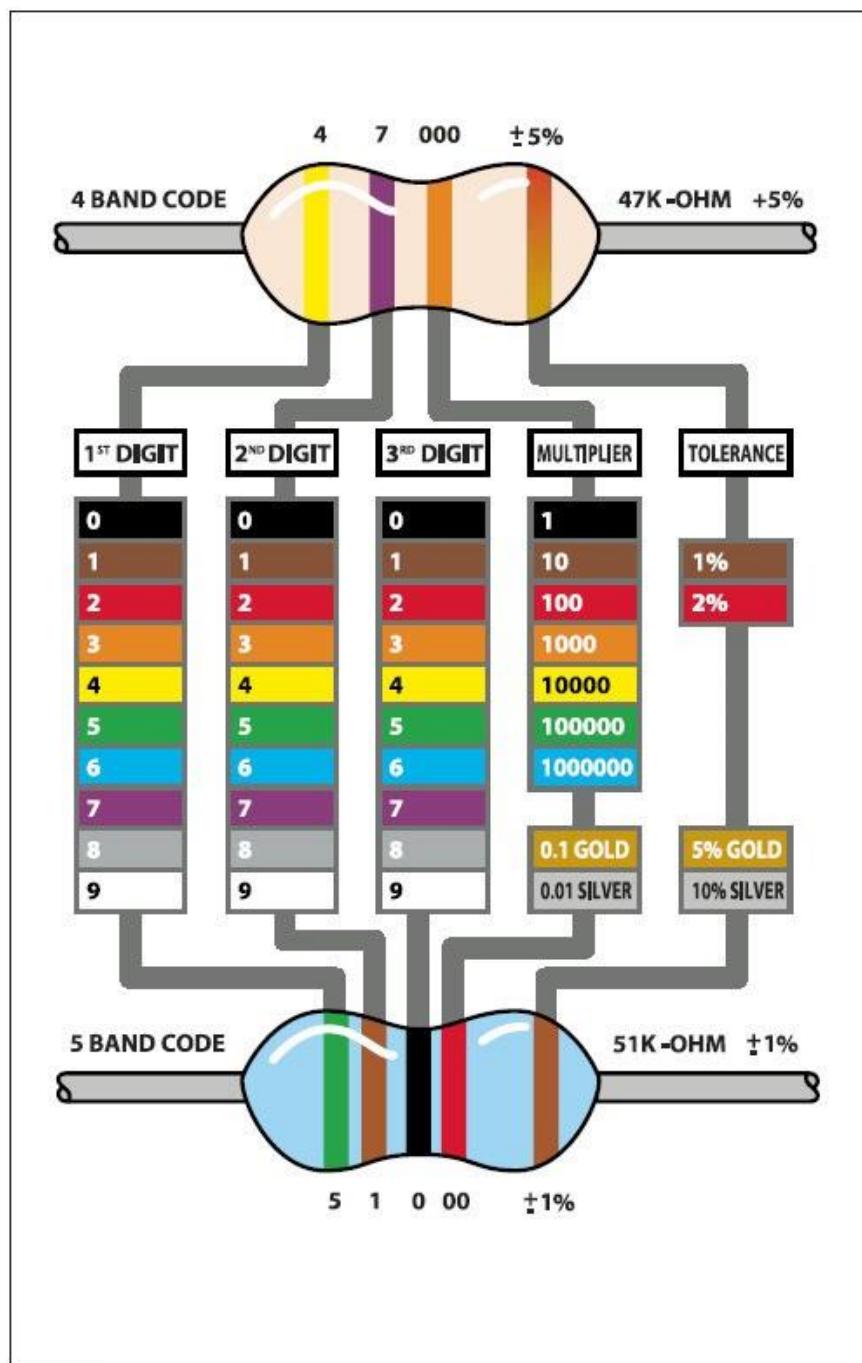


Ma prima diciamo qualcosa di più sulle resistenze...

L'unità di resistenza è chiamata Ohm, che solitamente è abbreviata con la lettera greca  $\Omega$  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  $k\Omega$  (1,000  $\Omega$ ) e  $M\Omega$  (1,000,000  $\Omega$ ). I due valori si leggono come kilo-ohm e mega-ohm.

In questa lezione utilizzeremo tre diversi tipi di resistenza tutte con valori diversi: 220 $\Omega$ , 1k $\Omega$  and 10k $\Omega$ . 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.



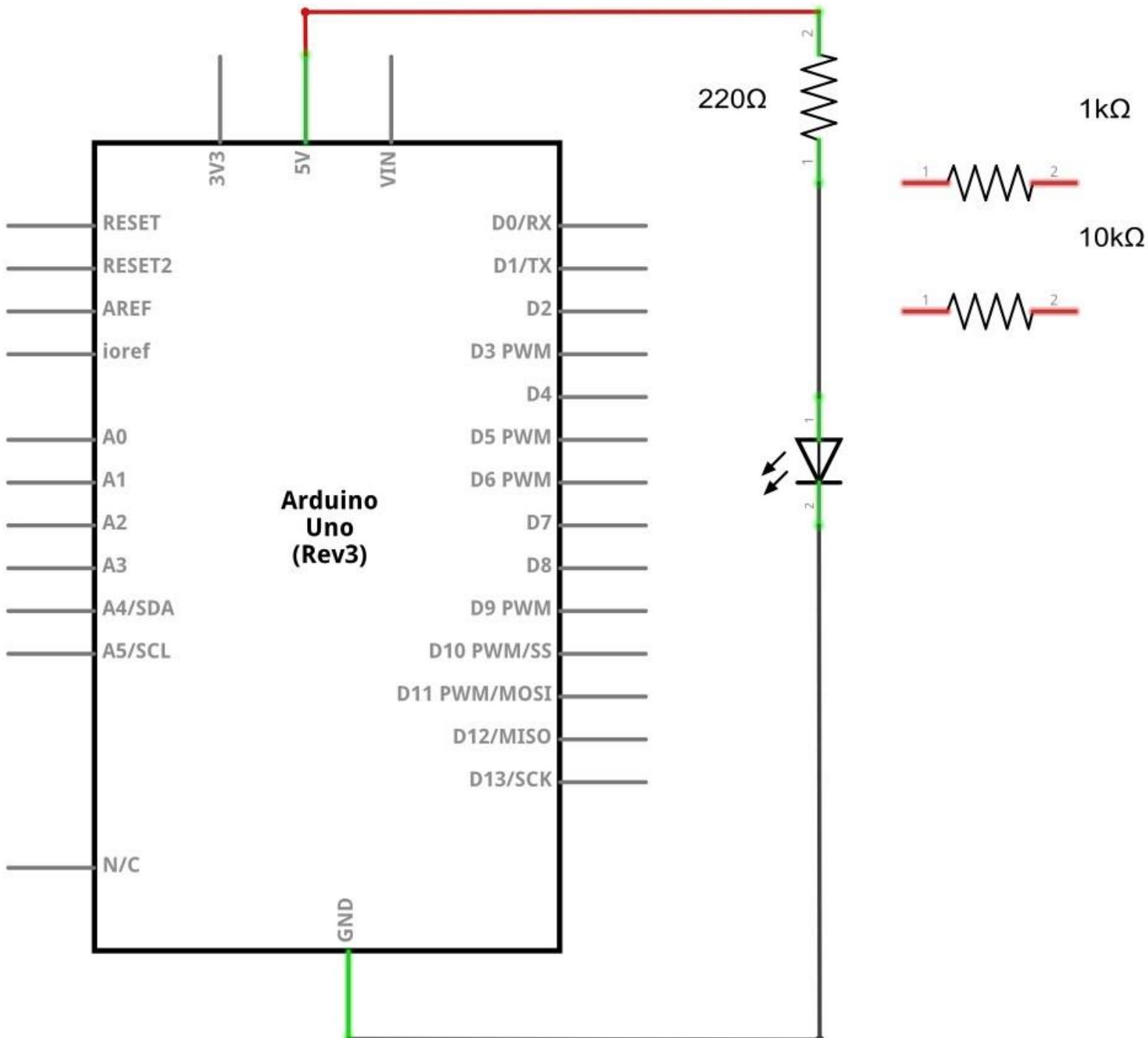
A differenza dei LED, le resistenze non hanno un capo negativo ed uno positivo, e possono essere collegati indifferentemente in un verso o nell'altro.

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

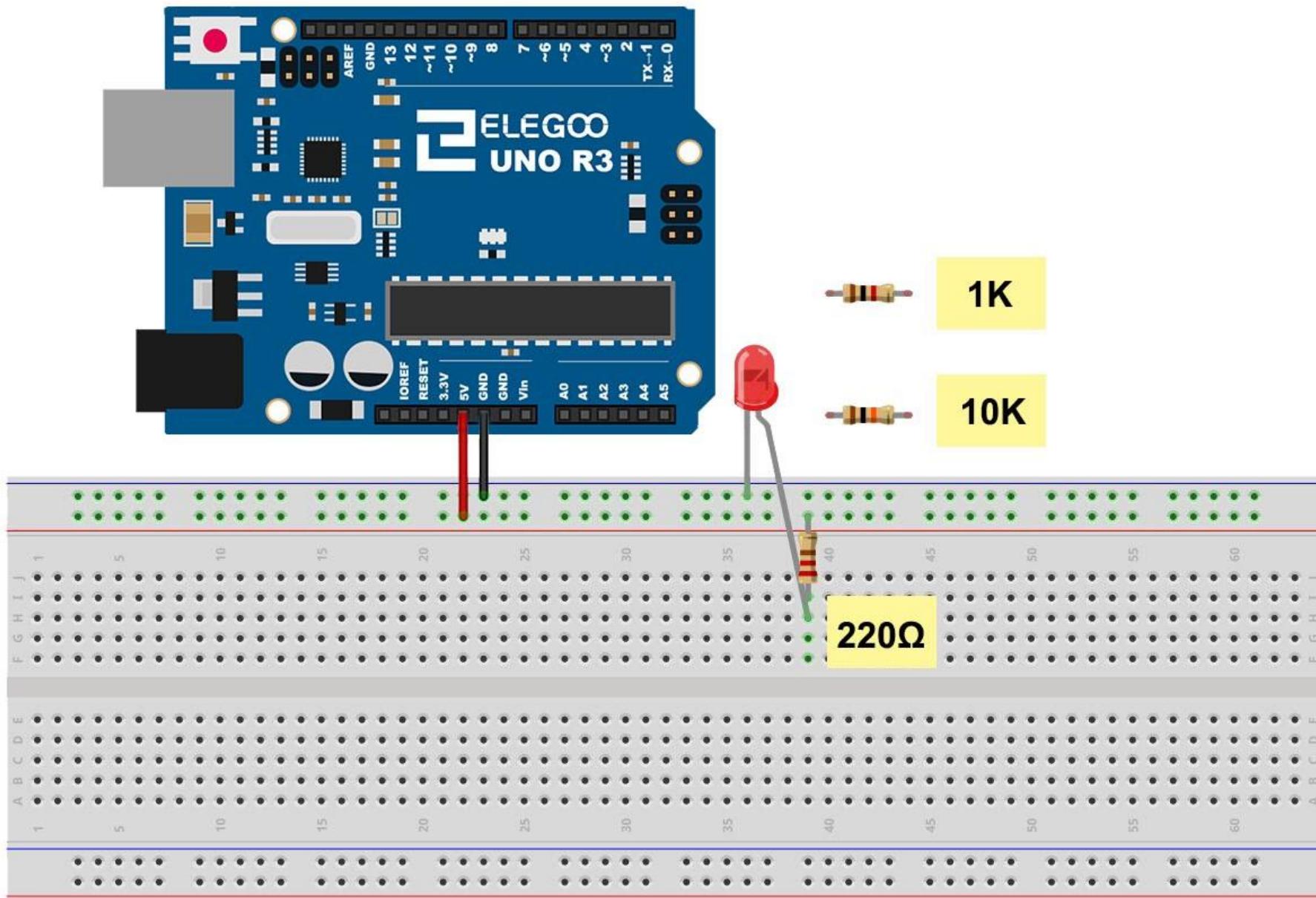
Oppure puoi usare un multimetro digitare.

## 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.

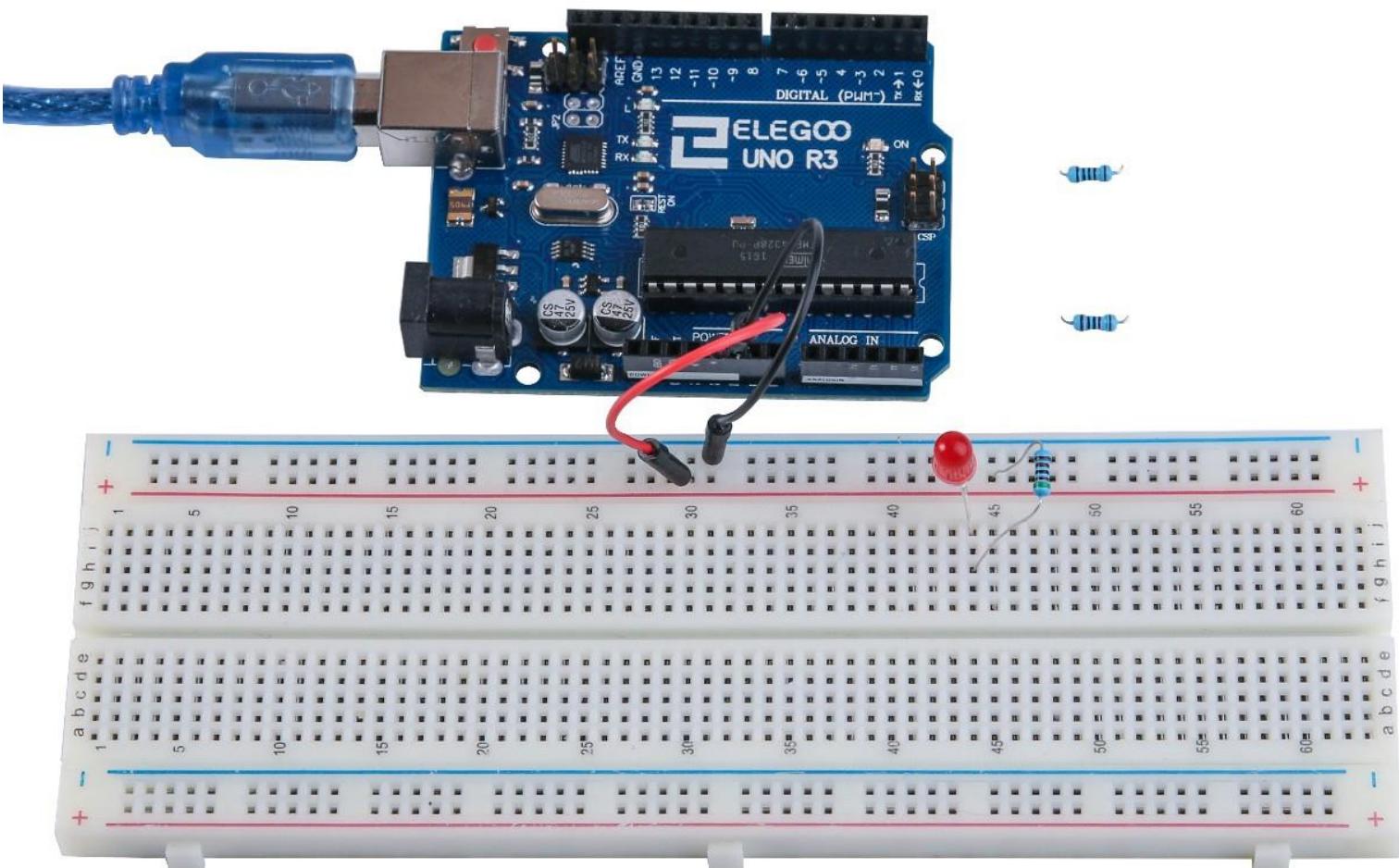
Con collegata la resistenza da  $220\ \Omega$ , il LED ti apparirà molto luminoso. Se cambi la resistenza e invece della  $220\ \Omega$  utilizzi una resistenza da  $1k\ \Omega$  il led apparirà leggermente meno luminoso. Infine collegando la resistenza da  $10k\ \Omega$ , il led sarà molto poco luminoso,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

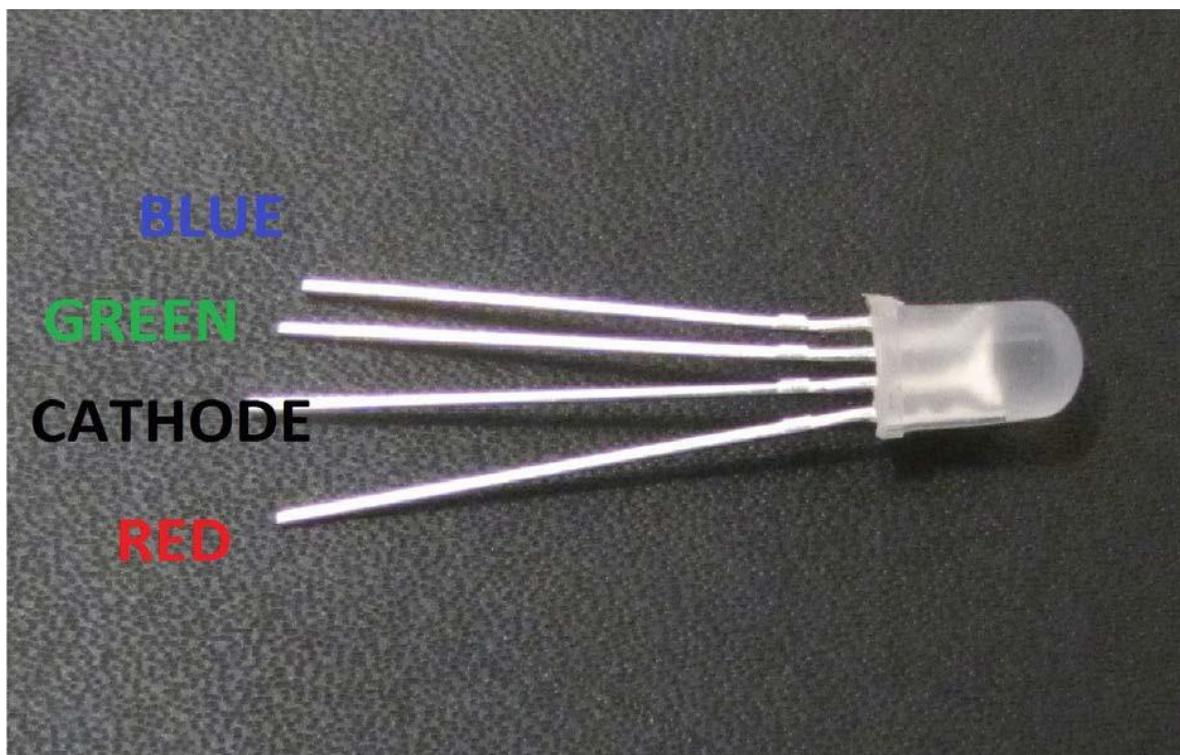
### LED RGB:

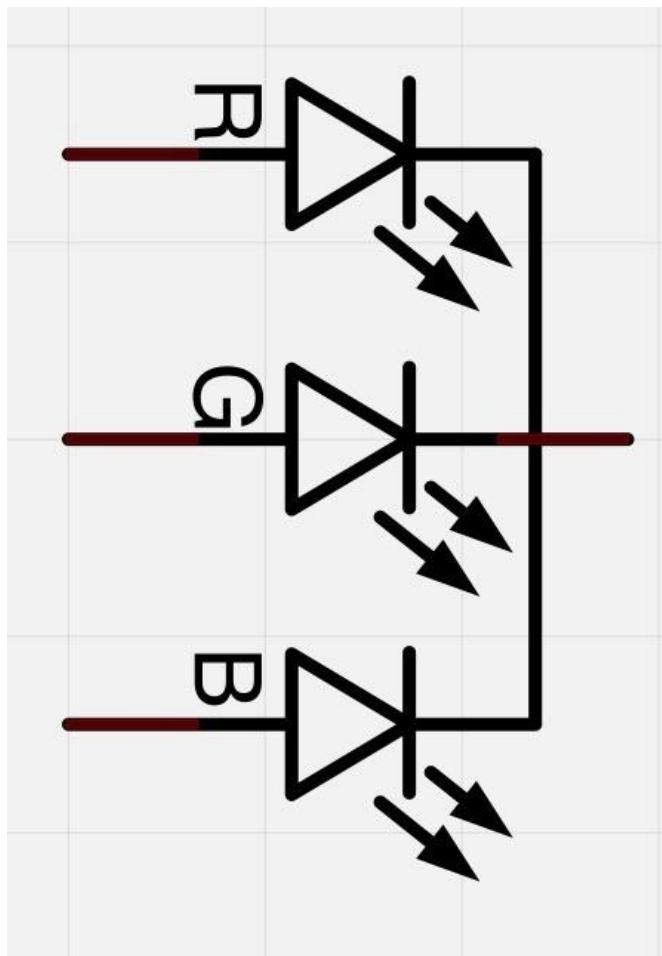
A prima vista i LED RGB (Red, Green, Blue) sembrano dei normali led. Tuttavia all'interno del solito corpo del led ci sono 3 led differenti, uno Rosso, uno Verde e ovviamente uno Blu. Controllando la luminosità di ogni led individualmente puoi controllare e mischiare praticamente qualsiasi colore tu 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.

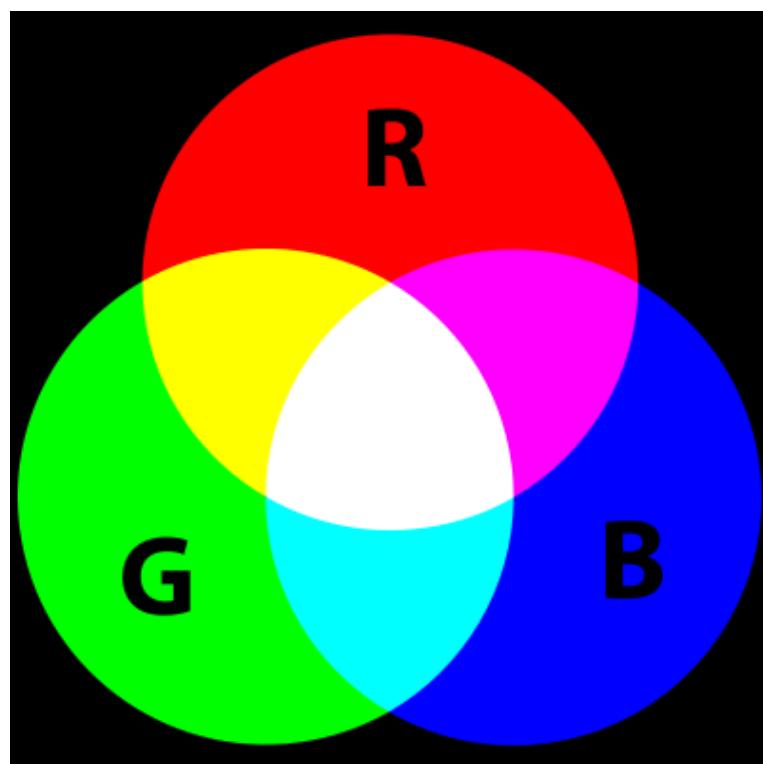
Esso è anche il piedino più lungo dei 4 e verrà connesso alla messa a terra.

Ogni led interno al corpo necessita una resistenza da  $220\Omega$  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  $220\Omega$ .

## 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 blu). 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.

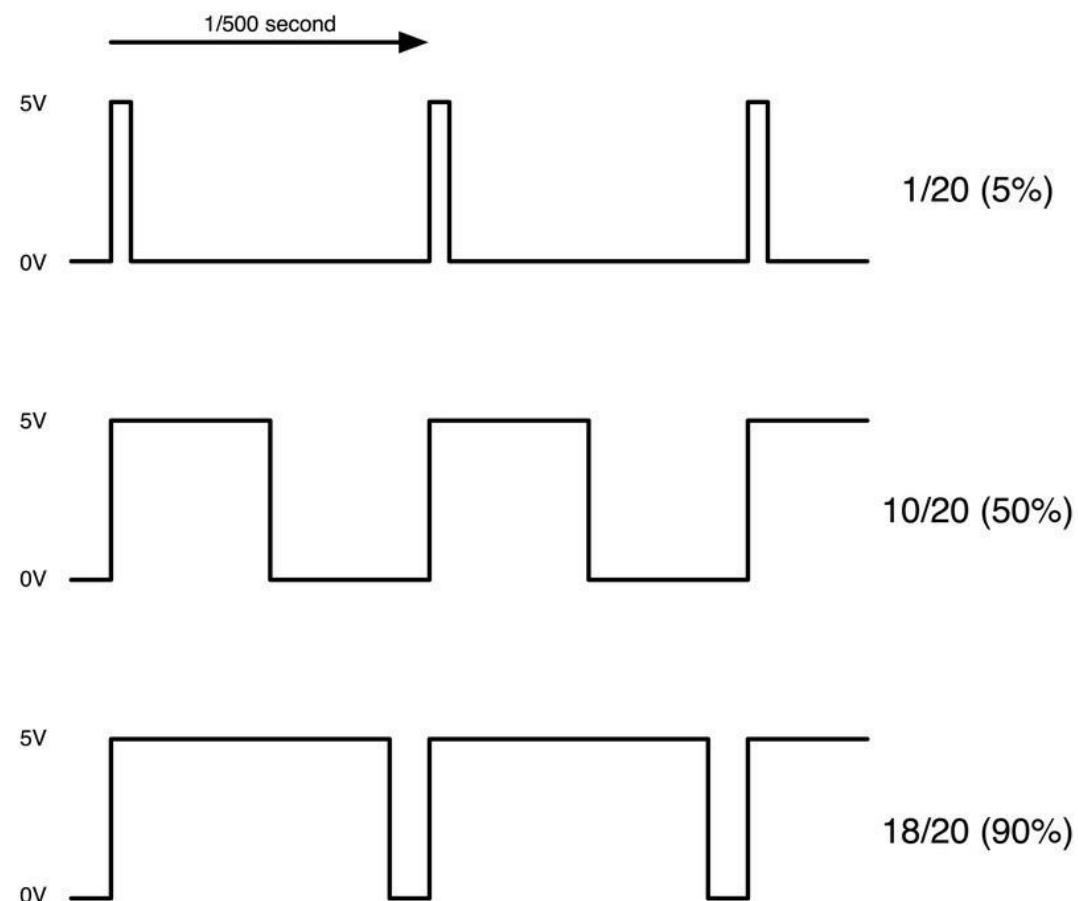
Il nero non è considerato un colore, ma semplicemente l'assenza di luce. Infatti, l'unico modo per avvicinarci a quel colore è semplicemente spegnendo tutti e tre i colori.

## 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 funzione 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

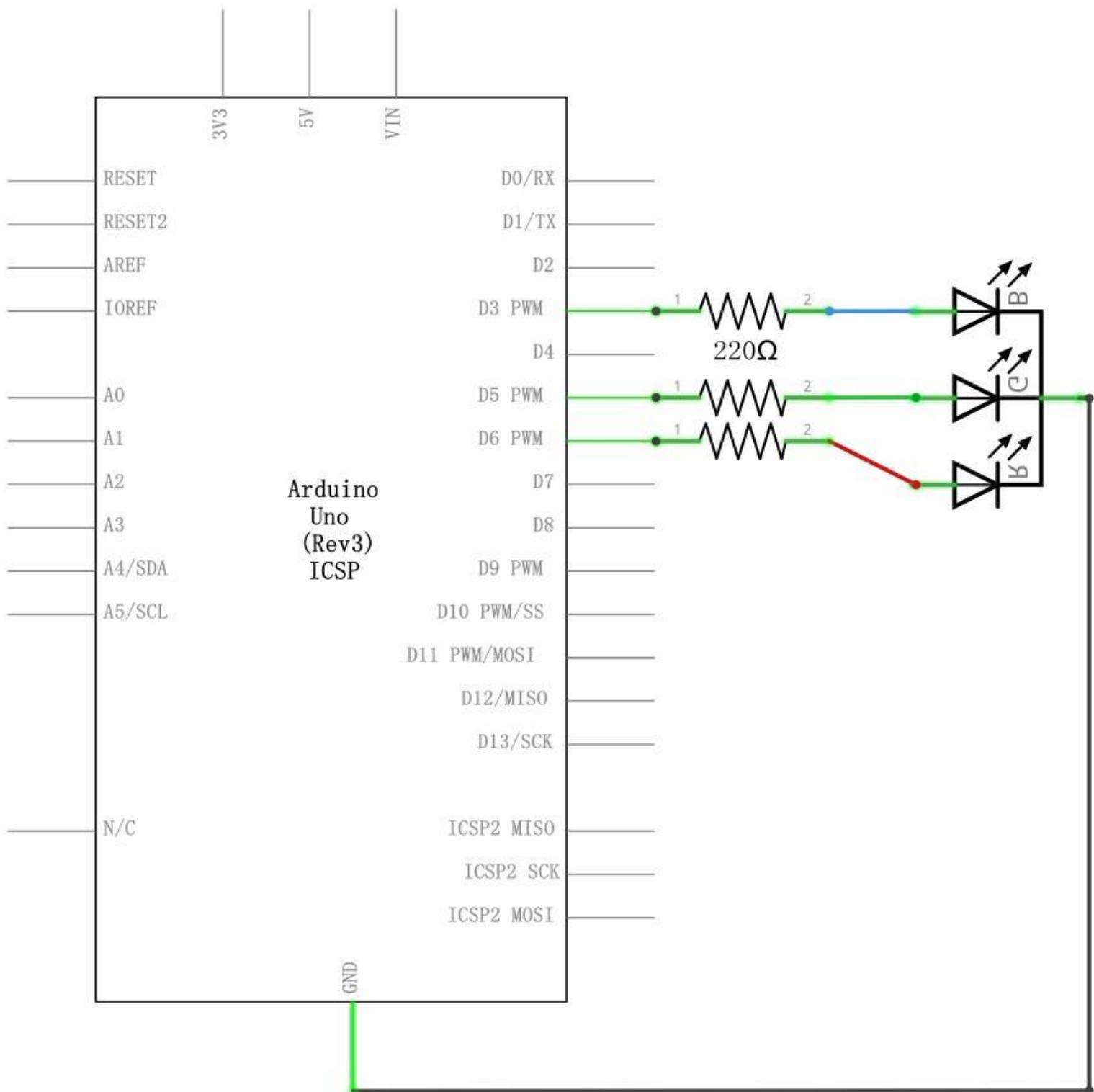
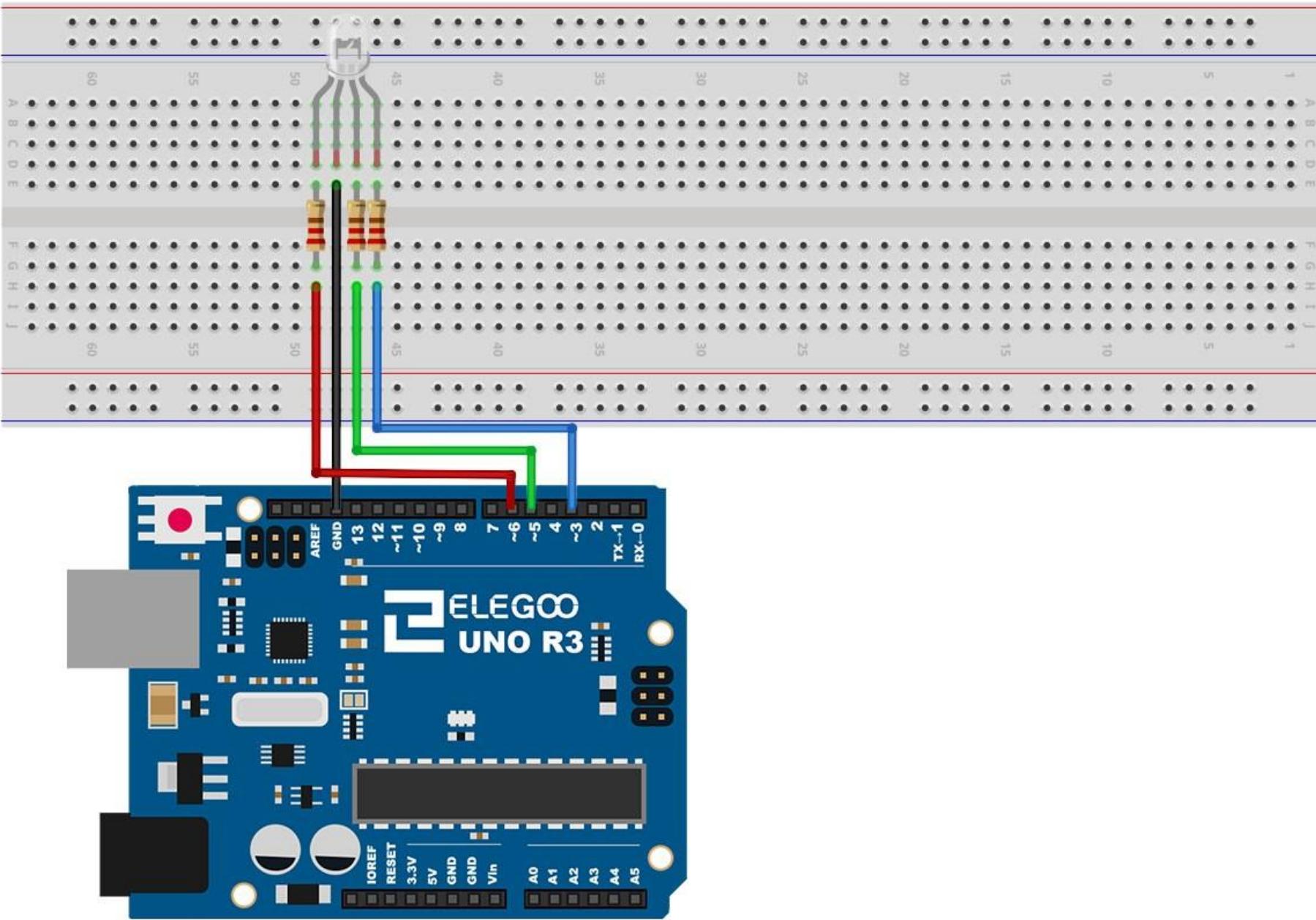


Diagramma di collegamento



## 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.

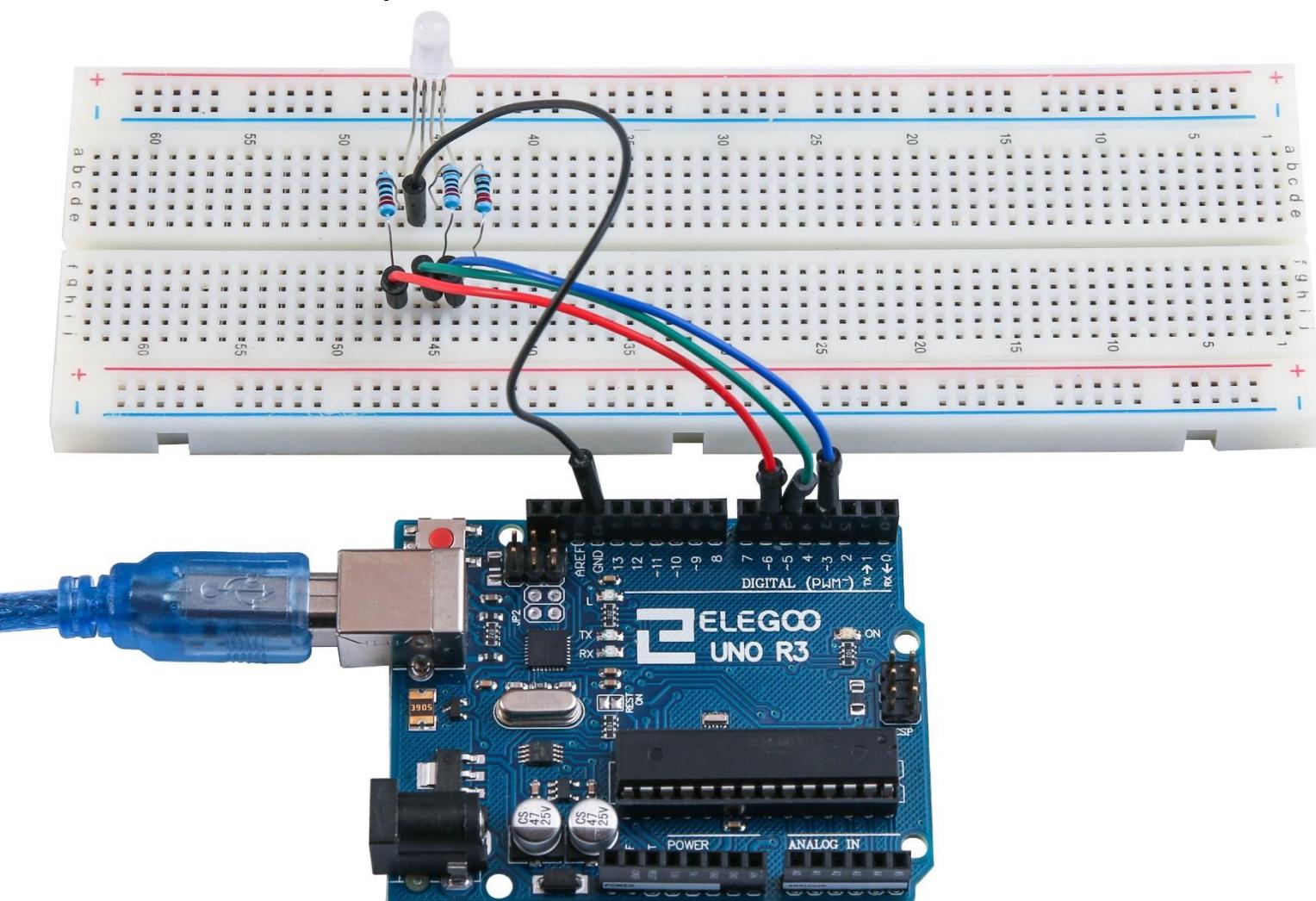
Se guardi la funzione “loop” puoi vedere che viene settata la quantità di rosso verde e blu che vogliamo visualizzare, dopodiché viene fatta una pausa di un secondo prima di iniziare con il prossimo colore.

```
#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:

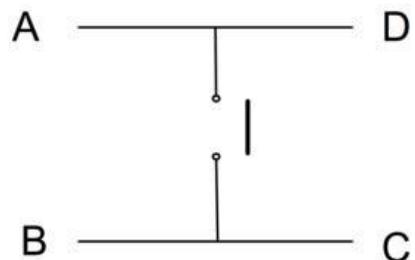
- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di aggancio
- (1) x LED rosso da 5mm
- (1) x Resistenza da 220 ohm
- (2) x Interruttore a pressione (bottone)
- (7) x Connettori M-M (Connettori Maschio-Maschio)

## 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.

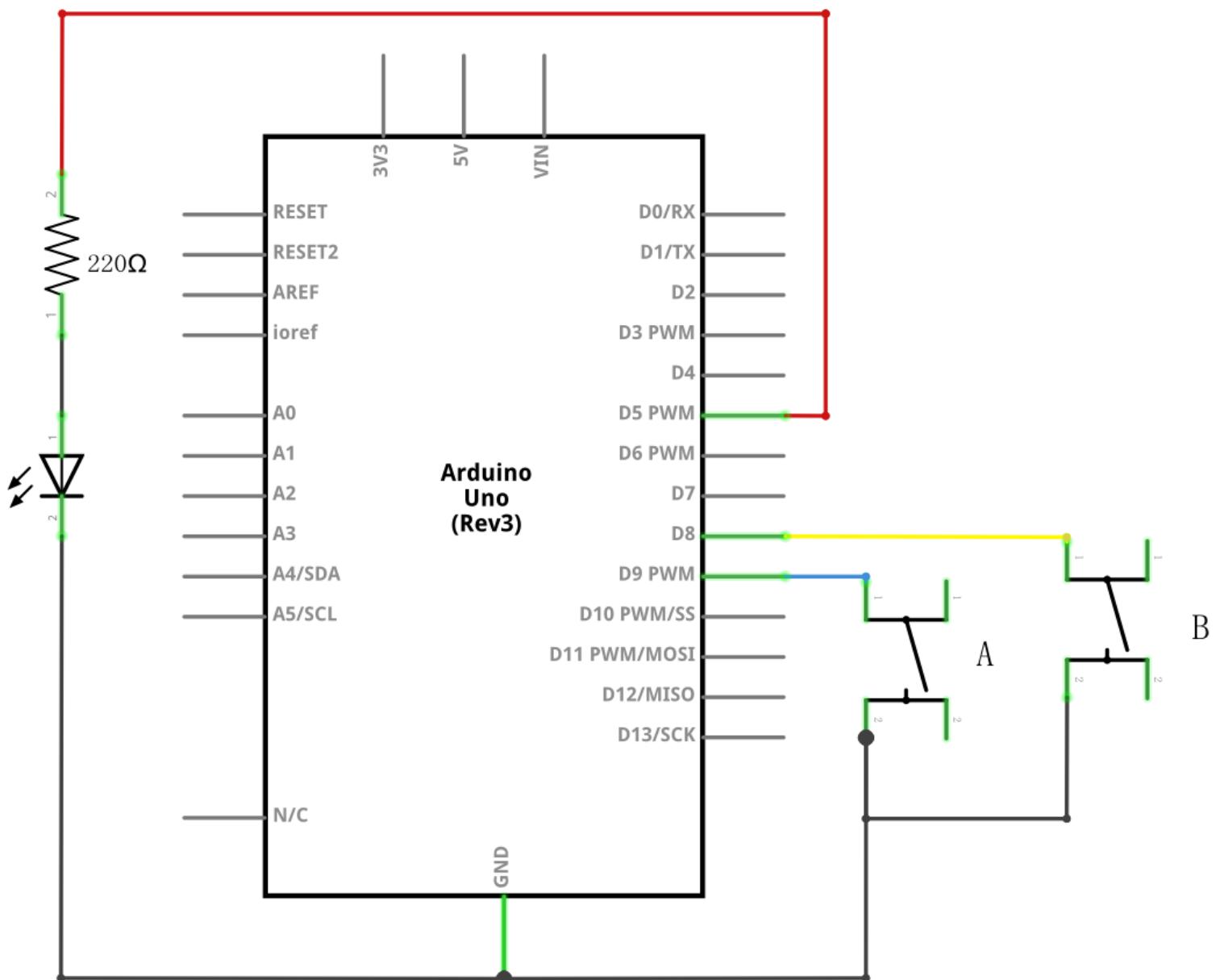
Il piccolo interruttore tattile che viene utilizzato in questa lezione ha quattro connessioni, ciò spesso mette in confusione.



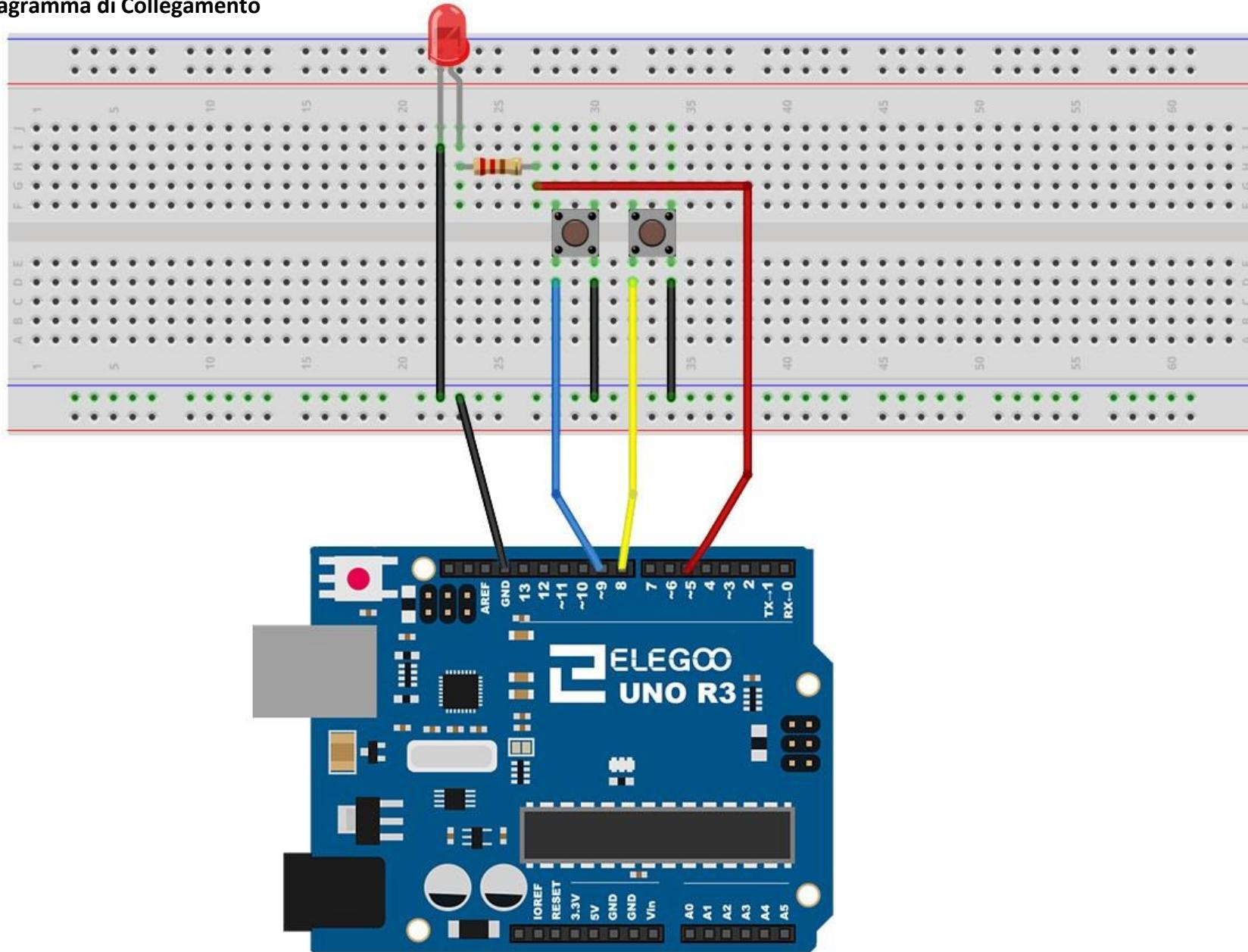
In realtà ci sono solamente due connessioni elettriche. All'interno dell'interruttore i pin B e C sono connessi insieme come lo sono anche A e D.

## 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 tre 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);
    }
}

```

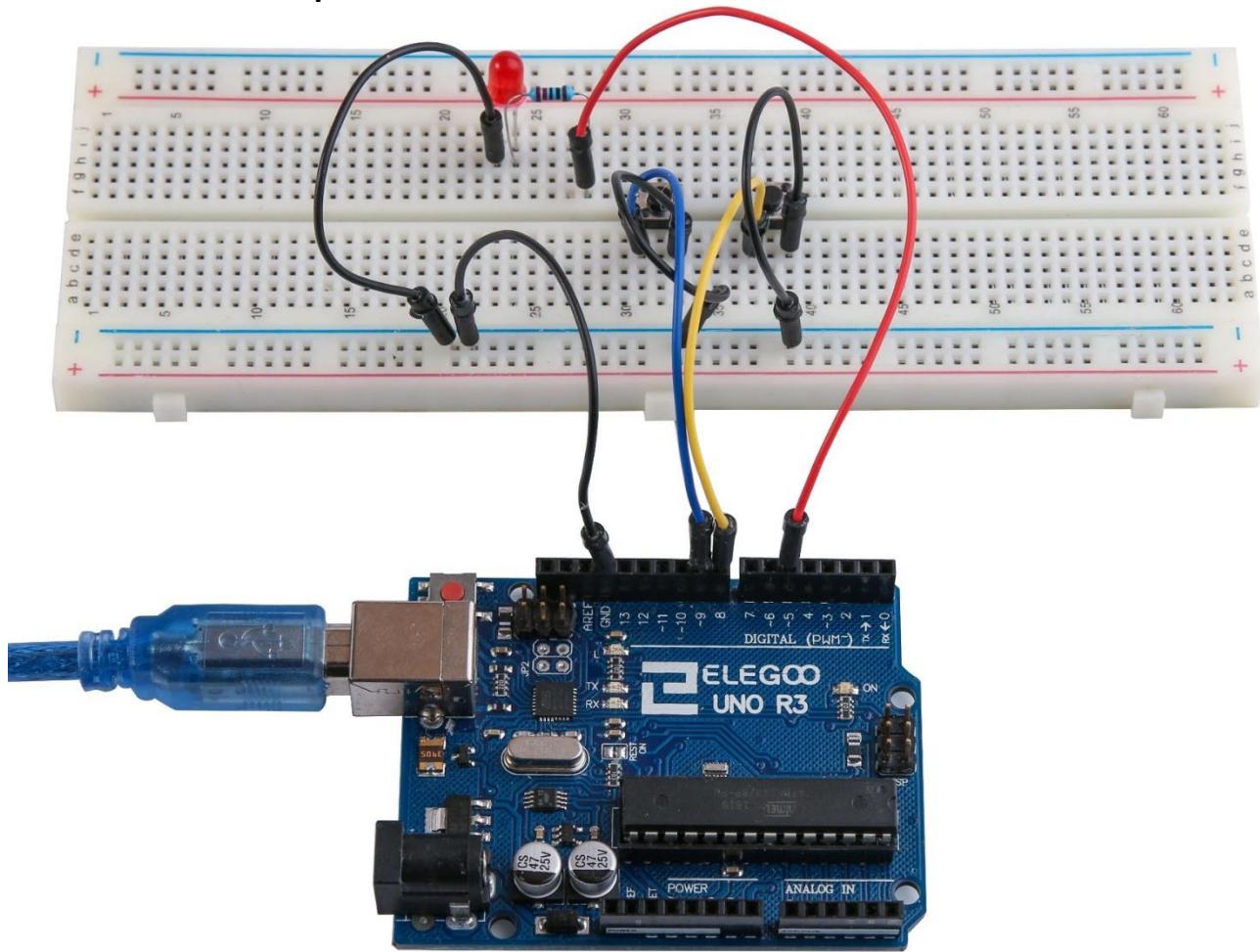
Nella funzione di “loop” ci sono due istruzioni “if”. Una per ogni bottone.

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.

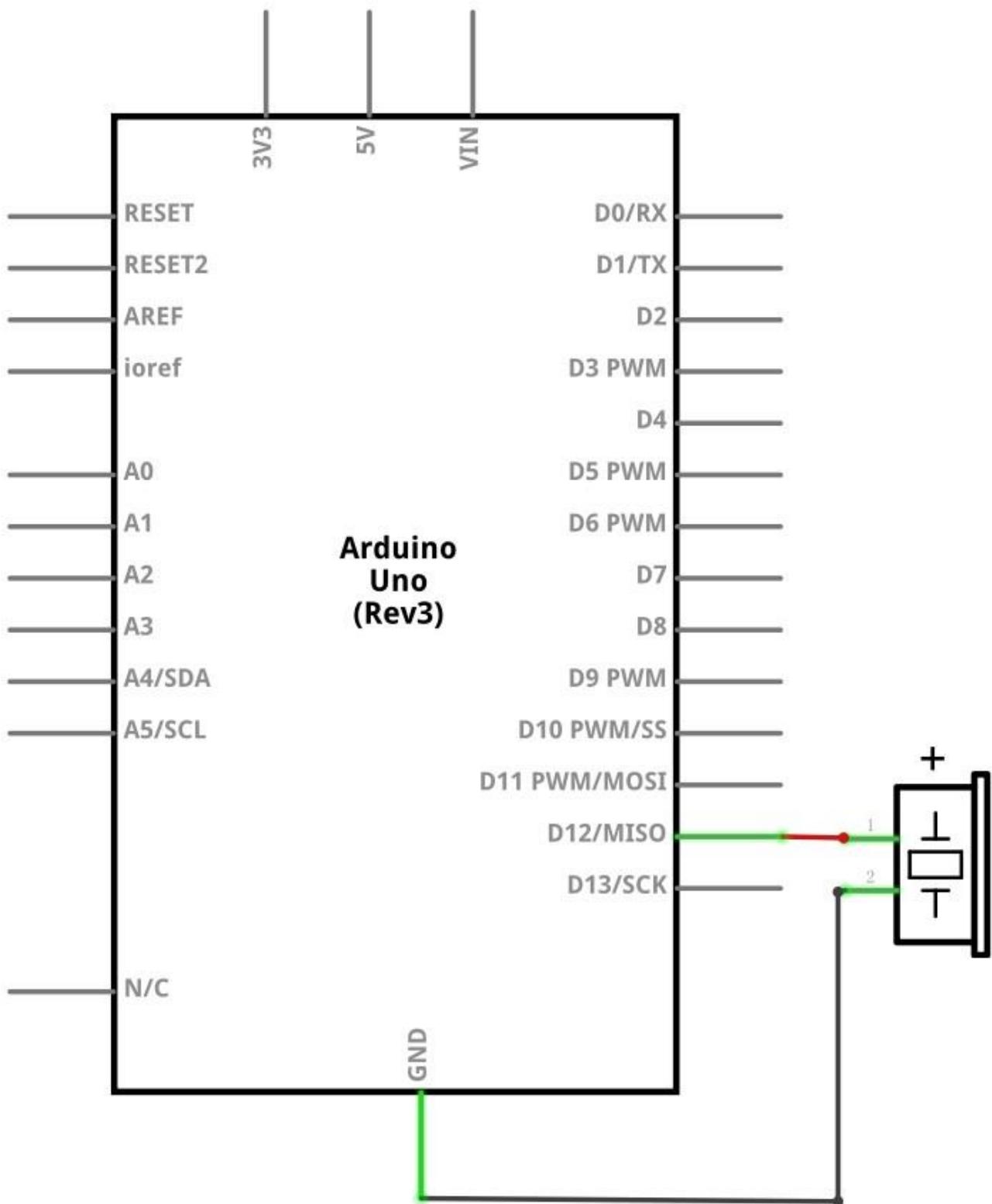
La differenza tra i due è che il cicalino attivo ha un ingresso oscillante, perciò genererà suono quando elettrificato. Un cicalino passivo al contrario, non ha tale ingresso, perciò non farà nessun suon quando riceverà della corrente continua come segnale. Con il cicalino passivo dovrà utilizzare un 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.

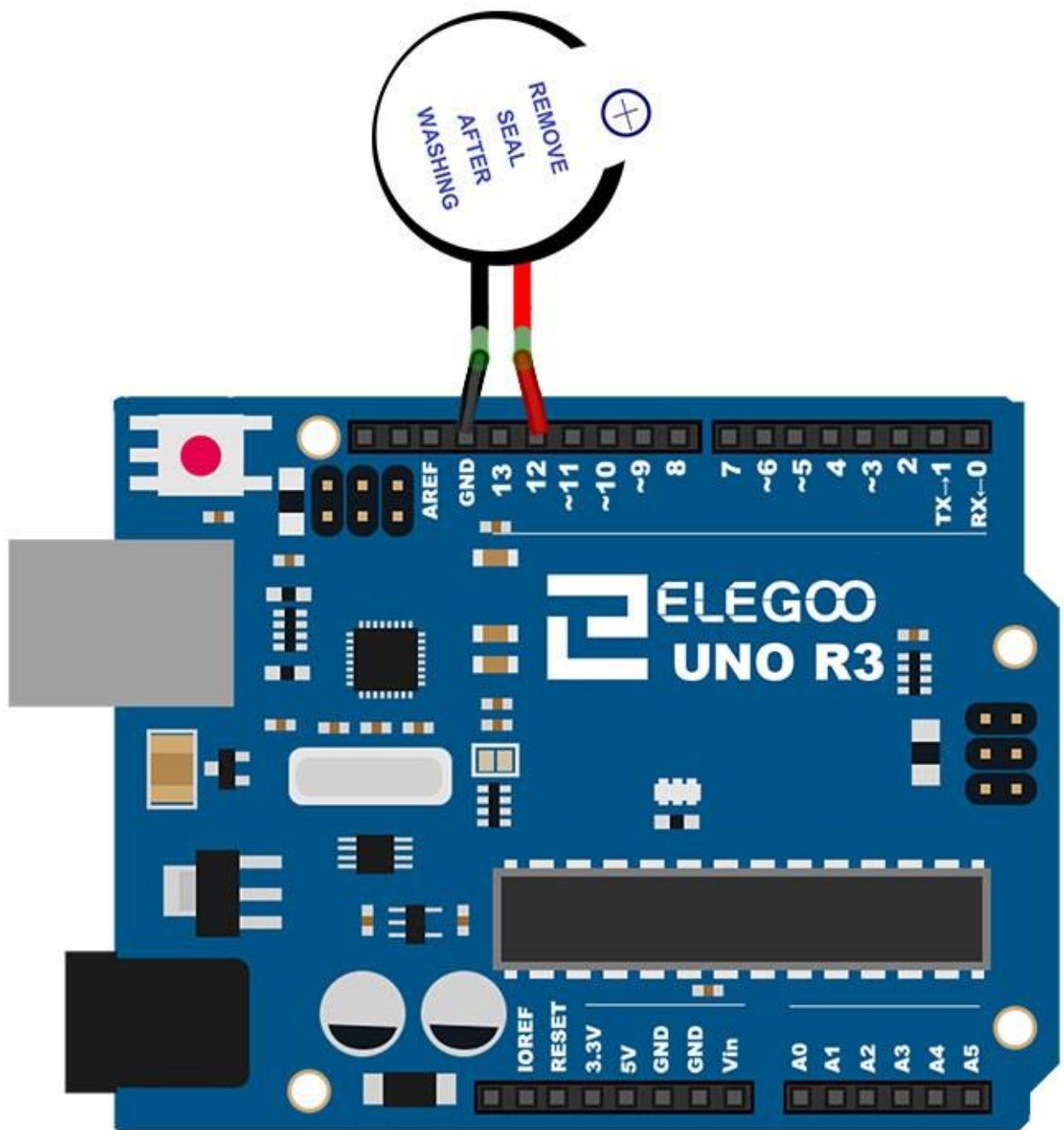


## Connessione

### Schema



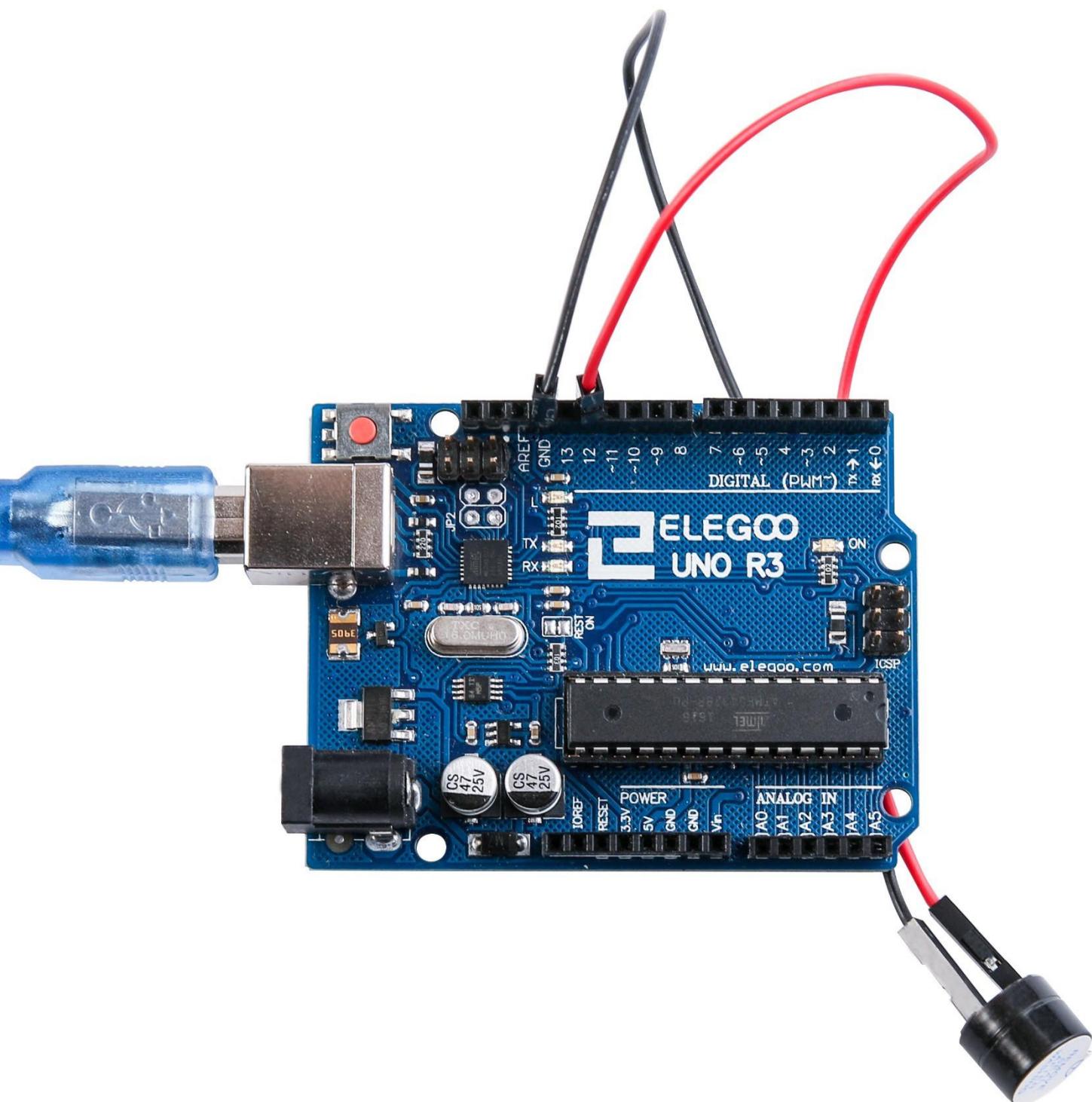
## Diagramma di Collegamento



## 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 Cicalino Passivo (Passive Buzzer)

### Introduzione

In questa lezione imparerai ad utilizzare un cicalino passivo.

L'obiettivo di questo esperimento è di generare otto suoni differenti, ognuno con una durata di 0.5 secondi: Do (523Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), Sol (784Hz), La (880Hz), Si (988Hz) Do Acuto (1047Hz).

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Cicalino Passivo (Passive buzzer)
- (2) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

### Introduzione ai Componenti

#### Cicalino Passivo:

Il principio di funzionamento del cicalino passivo si fonda sull'utilizzo della modulazione PWM le cui onde permettono la vibrazione dell'aria e la generazione del suono.

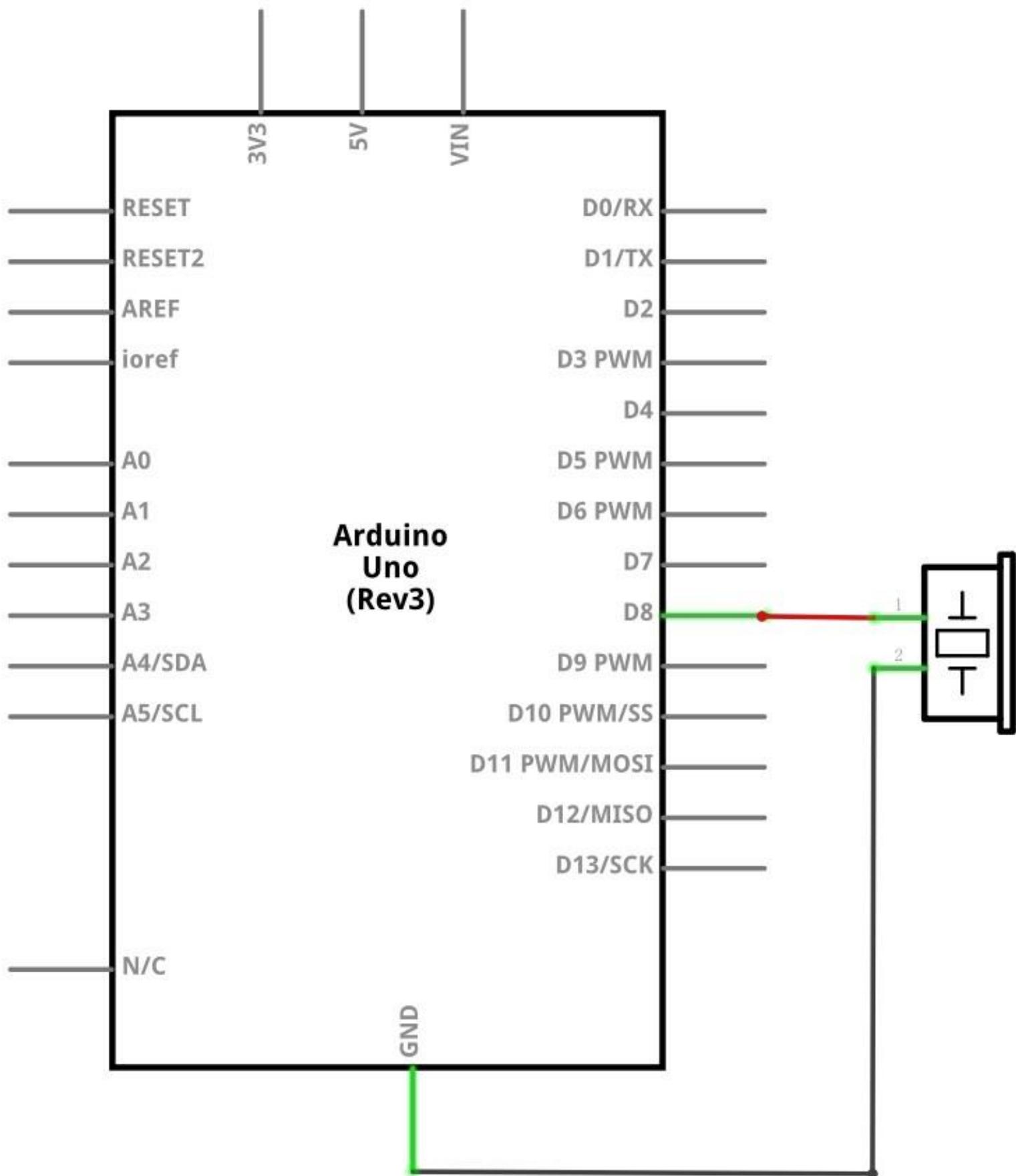
Impostando la frequenza delle onde correttamente, si possono generare suoni differenti. Per esempio inviando una pulsazione di 523Hz si genera un Do, con una pulsazione di 587Hz si genera un re, con una pulsazione di 659Hz si genera un Mi. Mettendo insieme tutto ciò si può costruire e suonare una canzone.

Devi solo prestare attenzione a non utilizzare la funzione analogWrite() della scheda UNO per generare le pulsazioni per il cicalino, in quanto le pulsazioni generate da analogWrite sono fisse a 500Hz

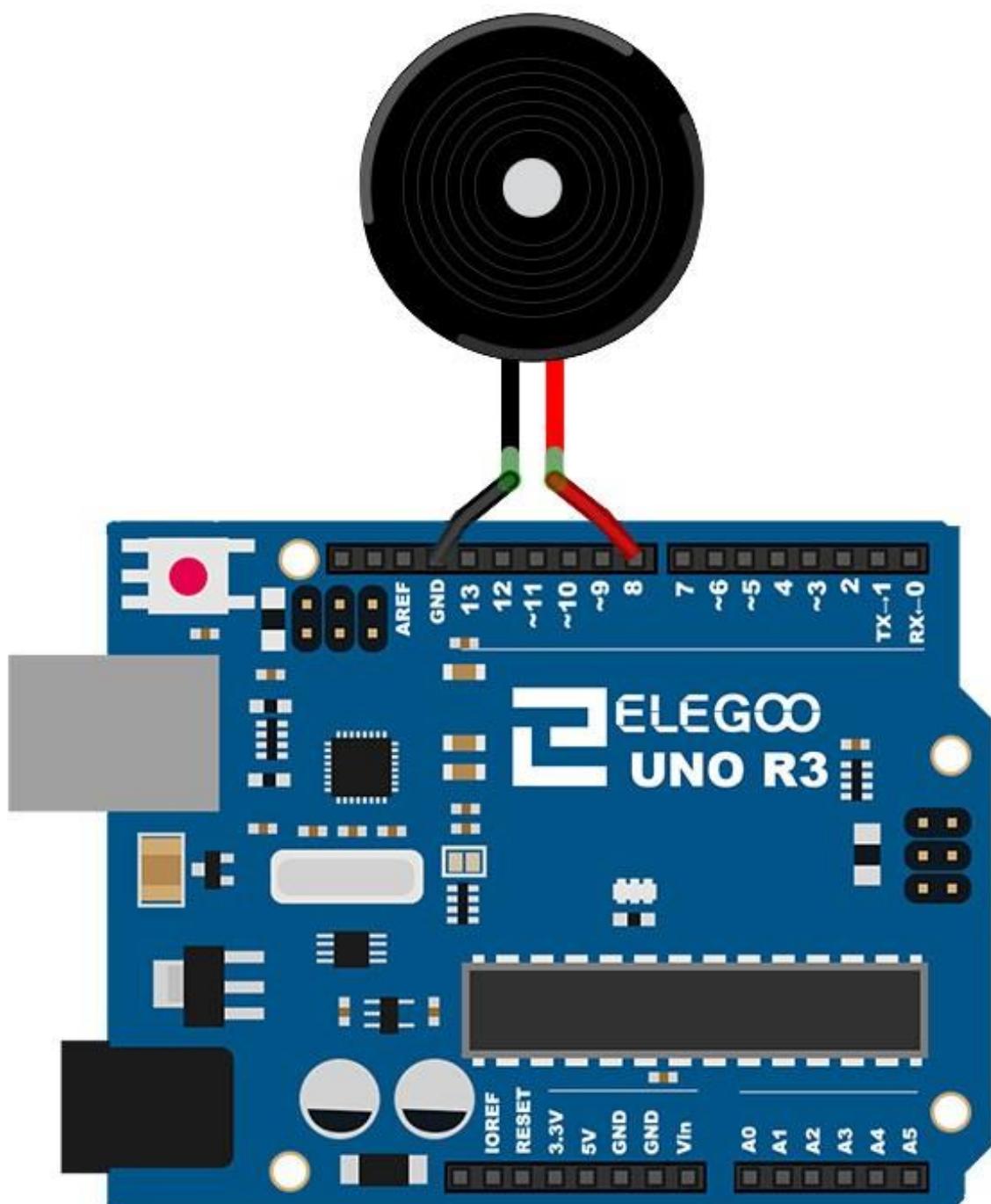


## Connessioni

### Schema



## Diagramma di Collegamento



Collega il Cicalino alla scheda UNO, il cavo rosso (positivo) al pin 8 e il cavo nero (negativo) alla messa a terra GND

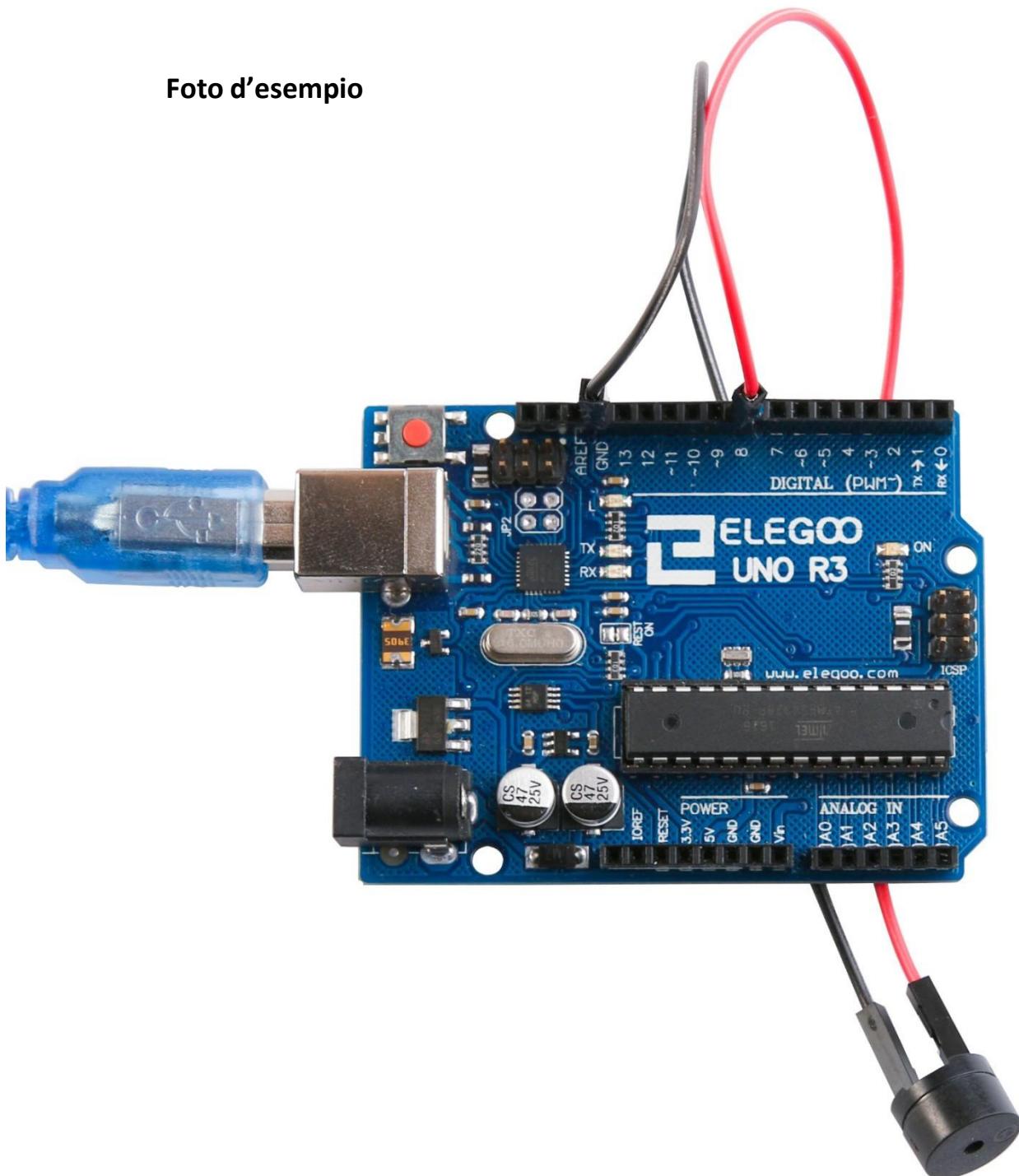
## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 7 Passive Buzzer”, 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.

Prima di eseguire questo codice, controlla di aver installato la libreria <pitches> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d'esempio



## Lezione 8 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 conduttriva 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

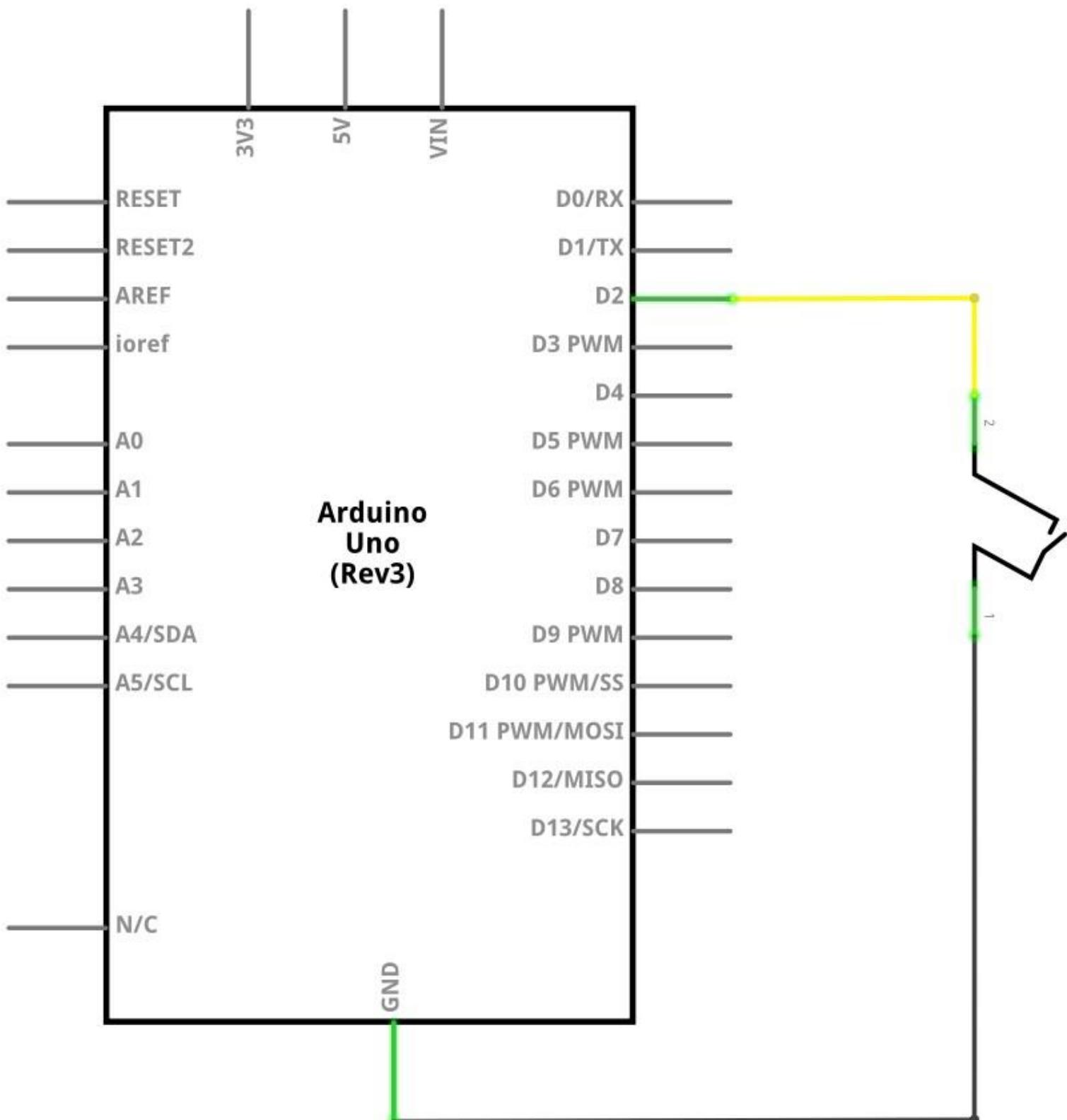
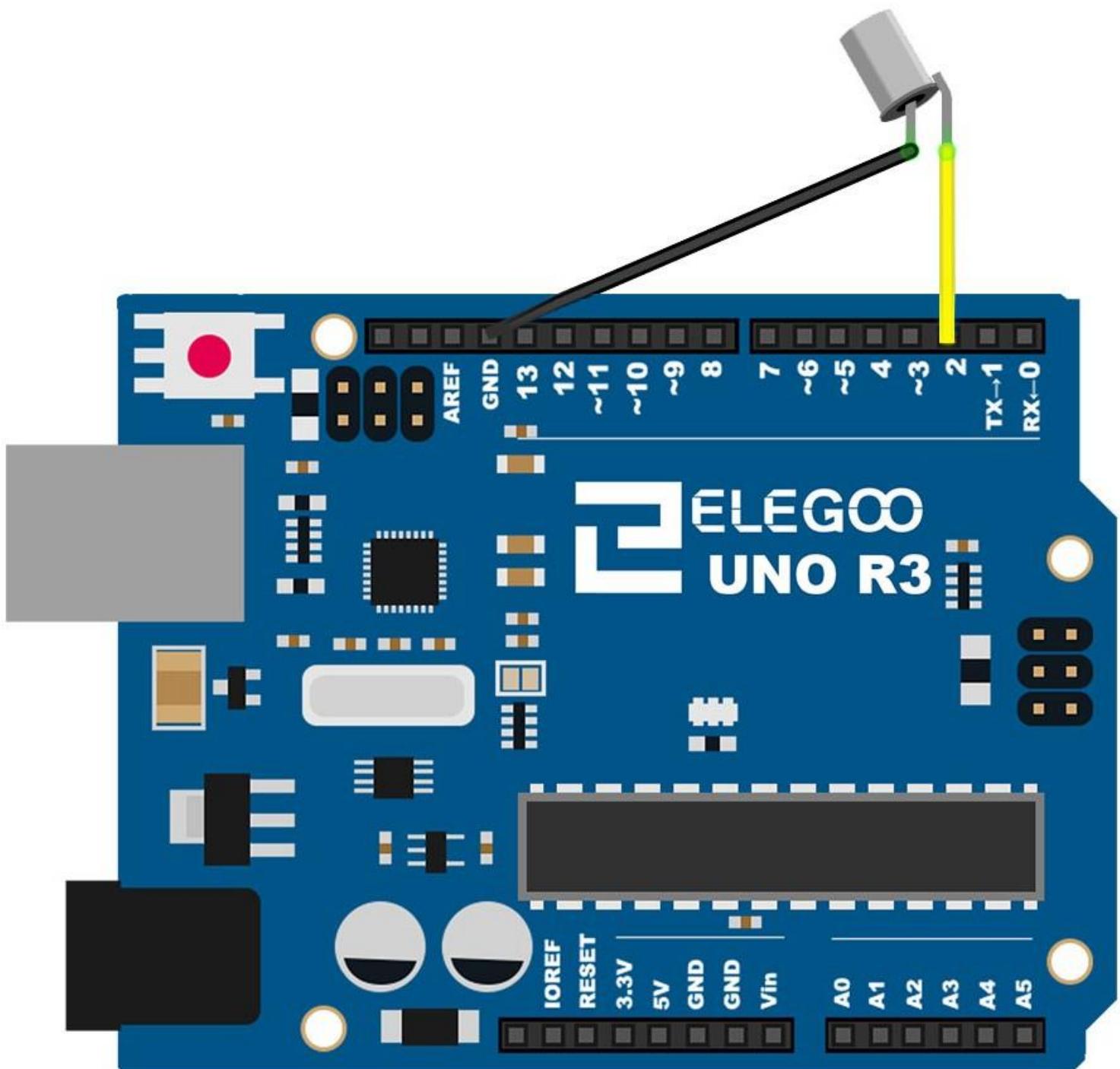


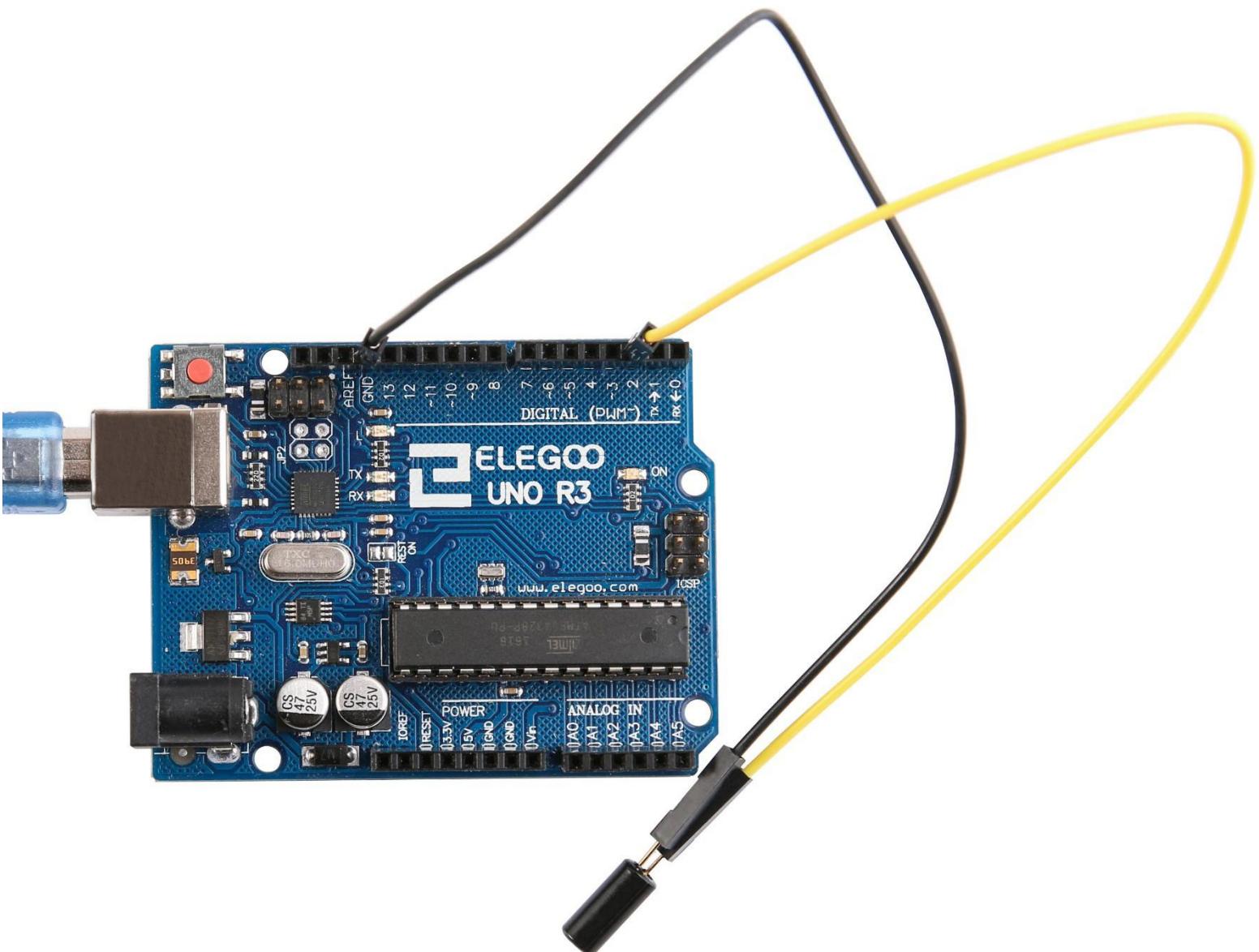
Diagramma di collegamento



## 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 9 Servo

### Introduzione

Il Servo è un tipo di motore che può ruotare solamente di 180 gradi. Viene controllato inviando impulsi elettrici tramite la scheda UNO. Queste pulsazioni indicano al servo in quale posizione deve ruotare.

Il Servo ha tre cavi: il cavo marrone è la messa a terra e va connessa alla porta GND della scheda UNO., il rosso è il cavo di alimentazione e va connesso alla porta dei 5v, infine il cavo arancio è il cavo del segnale e va connesso al pin numero 9.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Servo (SG90)
- (3) x Connettori M-M (Connettori Maschio-Maschio)

### Introduzione ai componenti

#### Servo SG90

- Universale per connettori JR e FP
- Lunghezza cavo: 25cm
- Nessun carico; velocità operativa: 0.12 sec / 60 gradi (4.8V), 0.10 sec / 60 gradi (6.0V)
- Coppia di stallo (4.8V): 1.6kg/cm
- Temperatura: -30~60'C
- Larghezza di banda morta: 5us
- Voltaggio di lavoro: 3.5~6V
- Dimensioni: 1.26 in x 1.18 in x 0.47 in (3.2 cm x 3 cm x 1.2 cm)
- peso: 4.73 oz (134 g)



## Connessione

### Schema

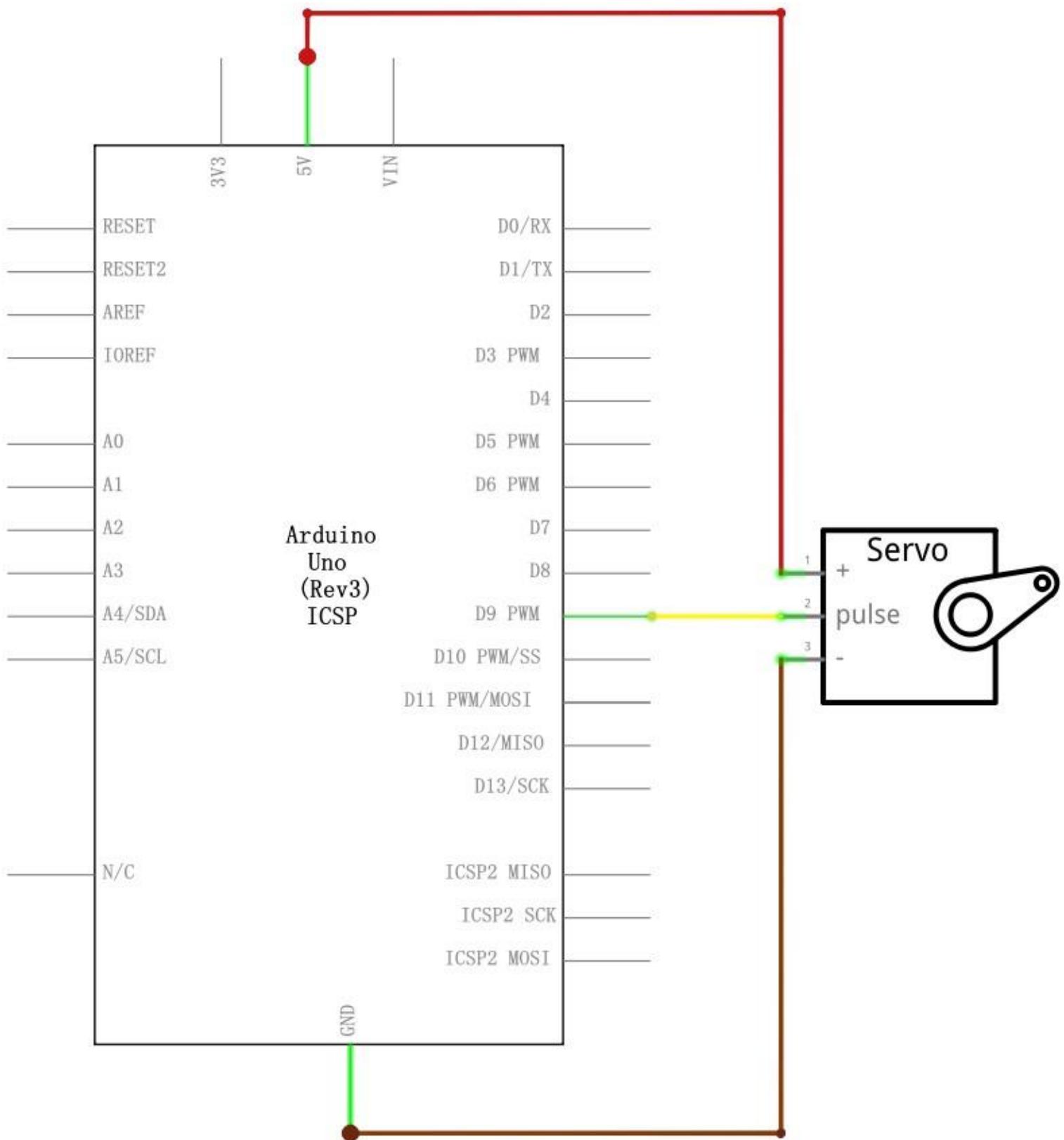
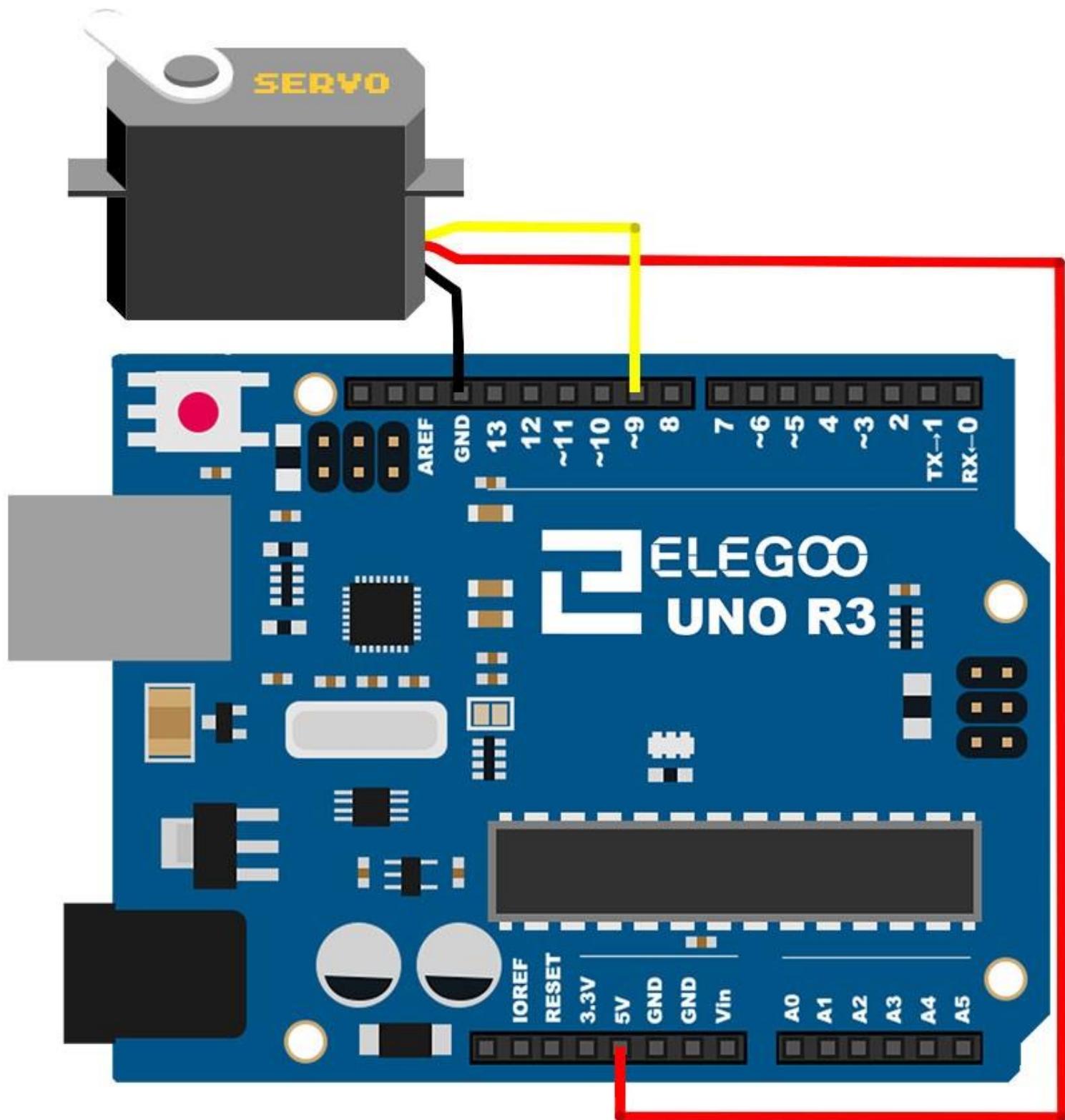


Diagramma di collegamento

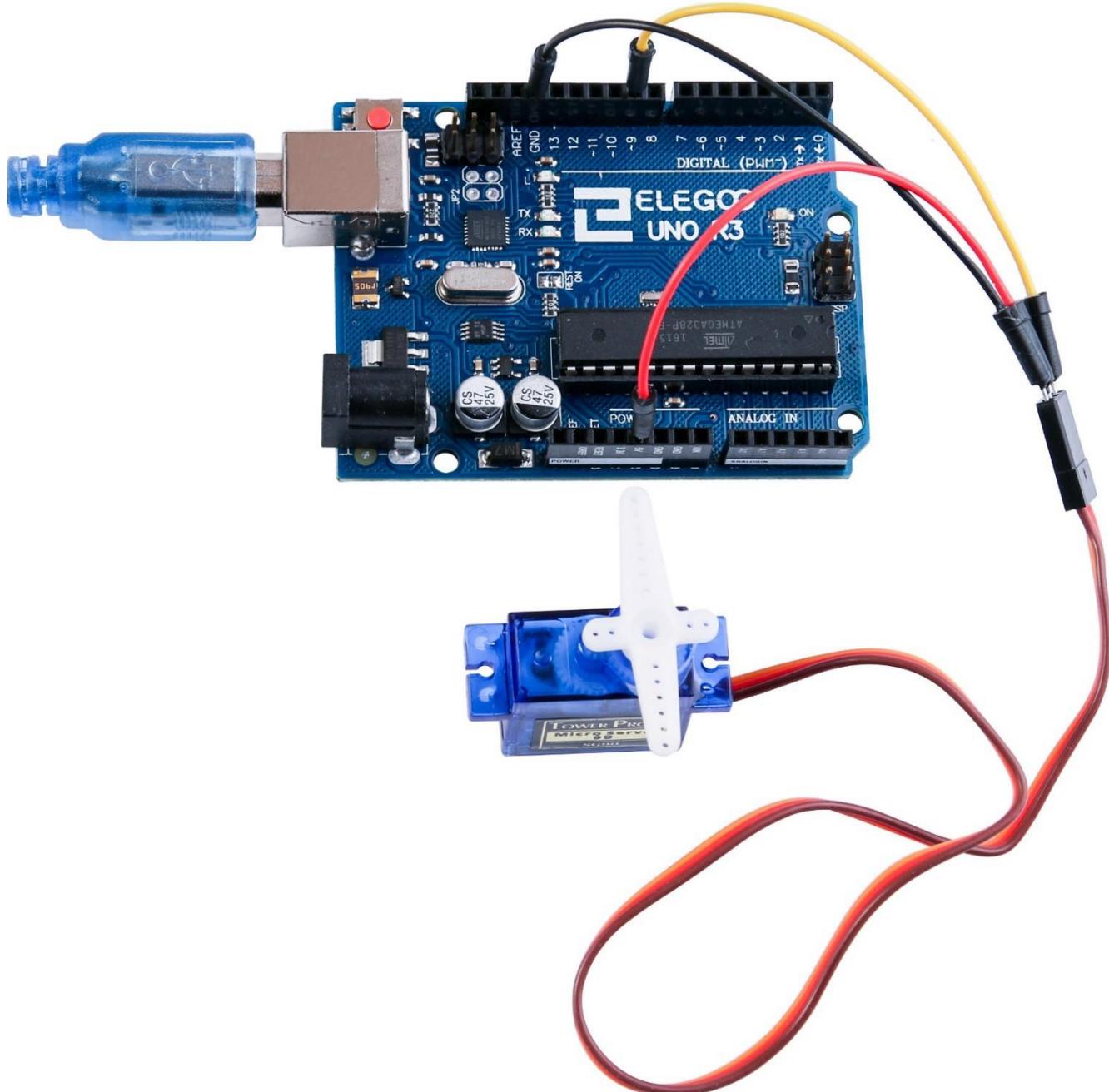


## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 9 Servo”, 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.  
Prima di eseguire questo codice, controlla di aver installato la libreria <Servo> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.  
Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d'esempio

Nella foto, il cavo marrone del servo è collegato tramite il cavo nero M-M, il cavo rosso del servo è collegato tramite il cavo rosso M-M, il cavo arancio del servo è collegato tramite il cavo giallo M-M.

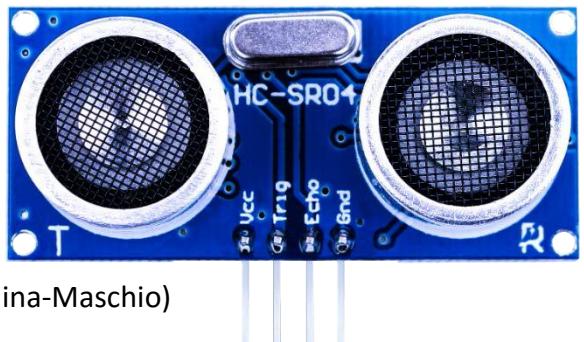


# Lezione 10 Modulo Sensore ad Ultrasuoni

## Introduzione

Il sensore ad ultrasuoni è grande per tutti i tipi di progetti che necessitano di misurare le distanze, progetti che necessitano di evitare gli ostacoli per esempio.

Il sensore HC-SR04 è economico e facile da utilizzare dato che utilizza una libreria specifica, scritta per questi sensori.



## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo sensore ad Ultrasuoni
- (4) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

## Introduzione ai componenti

### Sensore Ultrasonico

Il modulo sensore ultrasonico HC-SR04 permette un range di misura 2cm-400cm senza contatto, l'accuratezza del sensore può raggiungere i 3mm. Il modulo include un trasmettitore ad ultrasuoni, un ricevitore e un circuito di controllo.

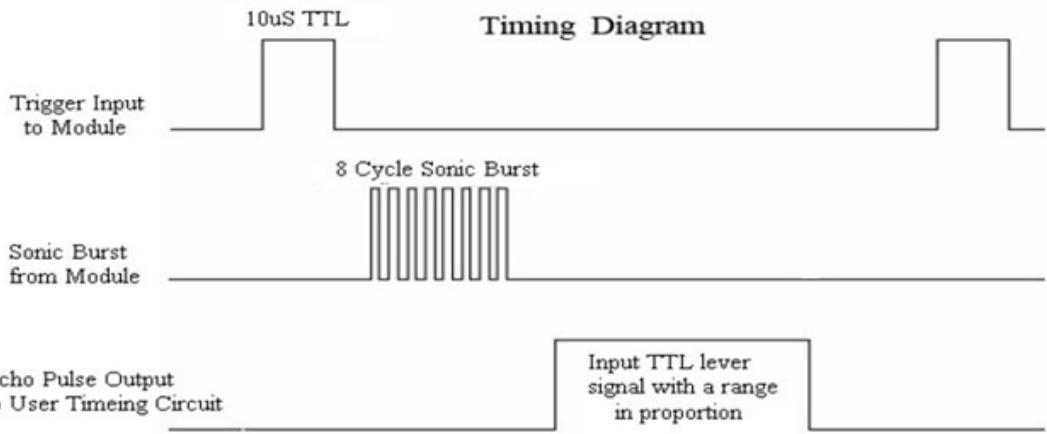
Questo è il principio base di funzionamento:

- (1) utilizzando il pin Trigger si invia un segnale HIGH per almeno 10us
- (2) il modulo invierà automaticamente otto onde a 40kHz ed individuerà l'eventuale ritorno una pulsazione di segnale.
- (3) il sensore risponderà sul pin Echo con un impulso HIGH della durata corrispondente a quelle del viaggio delle onde sonore.

Distanza Test =  $(\text{tempo di livello alto} \times \text{velocità del suono (340m/s)}) / 2$

Il diagramma del tempo è visualizzato qui sotto. Devi semplicemente inviare una veloce pulsazione 10us all'input del trigger per iniziare la misura, dopodiché il modulo invierà 8 veloci cicli di ultrasuoni a 40kHz per intercettare l'eco. Tramite queste pulsazioni è possibile calcolare la distanza tra il sensore e l'oggetto incontrato. Il range viene calcolato inviando un segnale di trigger e ricevendo il segnale su Echo. La formula per il calcolo è questa:  $\text{us}/58 = \text{centimetri}$  oppure  $\text{us}/148 = \text{pollici}$ ; oppure ancora, il range = tempo di HIGH \* velocità (340m/s) / 2;

Ti suggeriamo di usare cicli di misure non inferiori ai 60ms per prevenire l'accavallamento di più misurazioni.



## Connessioni

### Schema

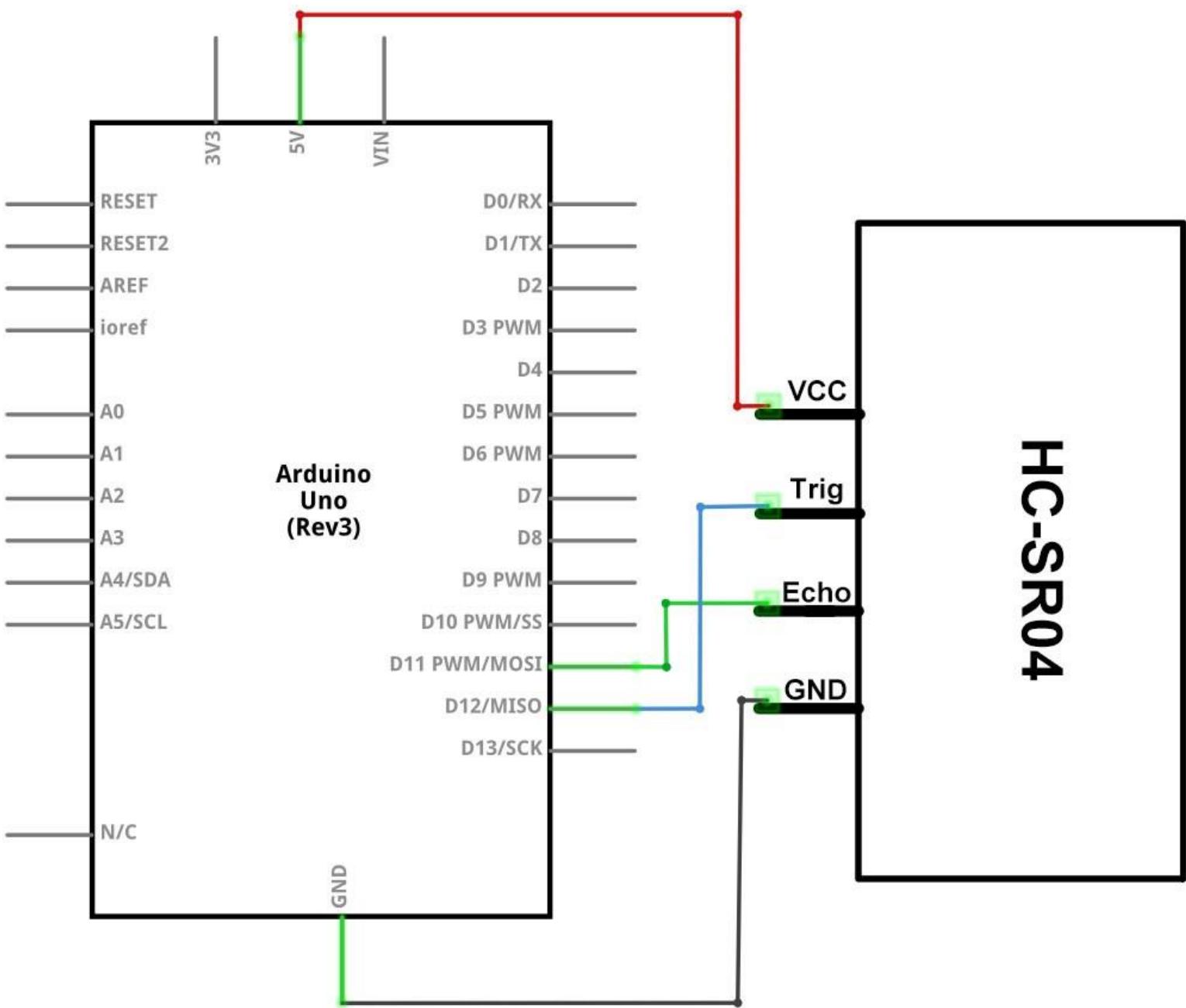
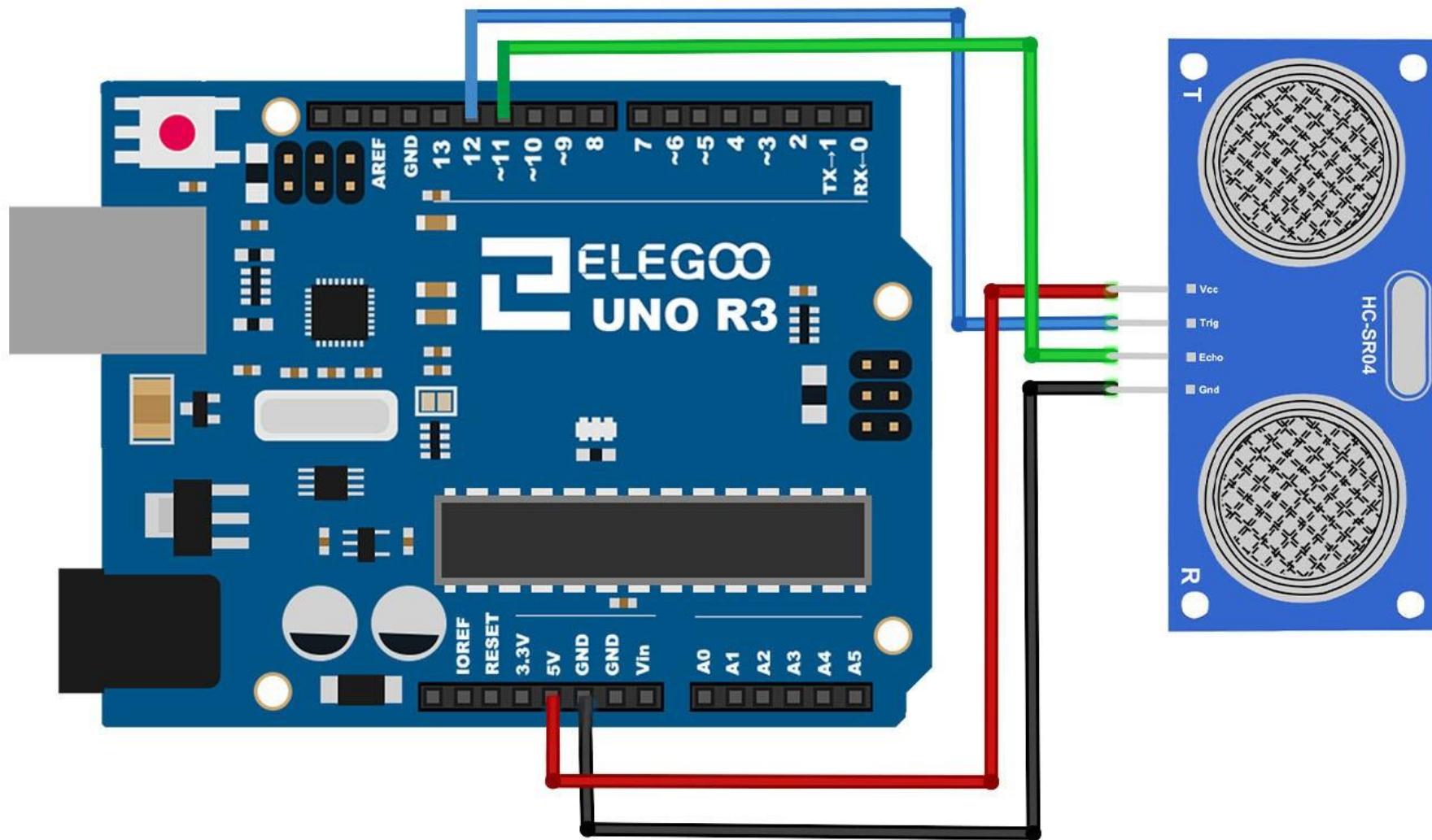


Diagramma di Collegamento



## Codice

Utilizzare la libreria scritta per questo sensore ci permetterà di avere un codice molto semplice e corto.

Il codice include la libreria all'inizio del nostro codice, in seguito vengono usati semplici comandi che permettono di utilizzare il sensore.

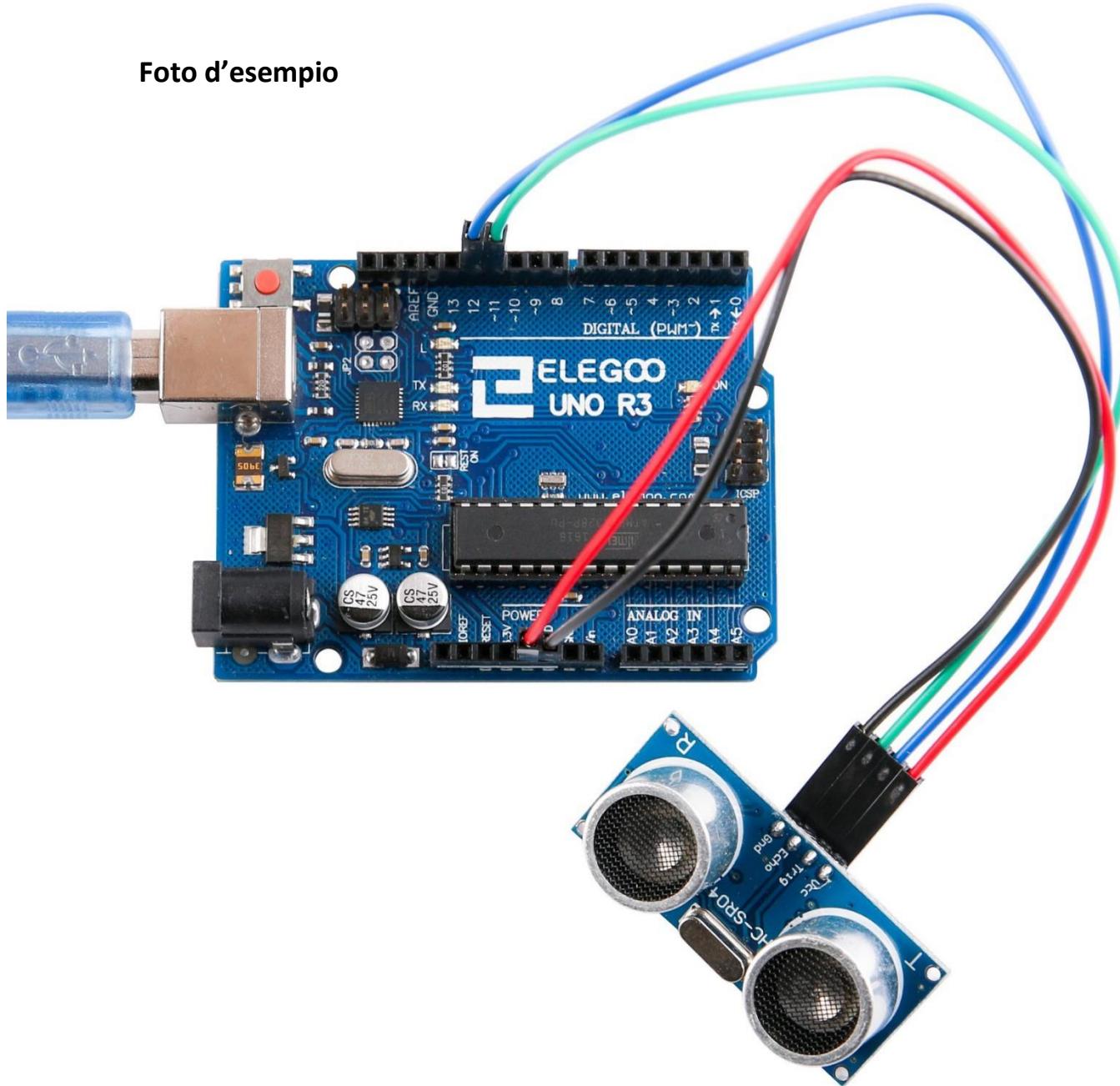
Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lesson 10 Ultrasonic Sensor Module", clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria < HC-SR04> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

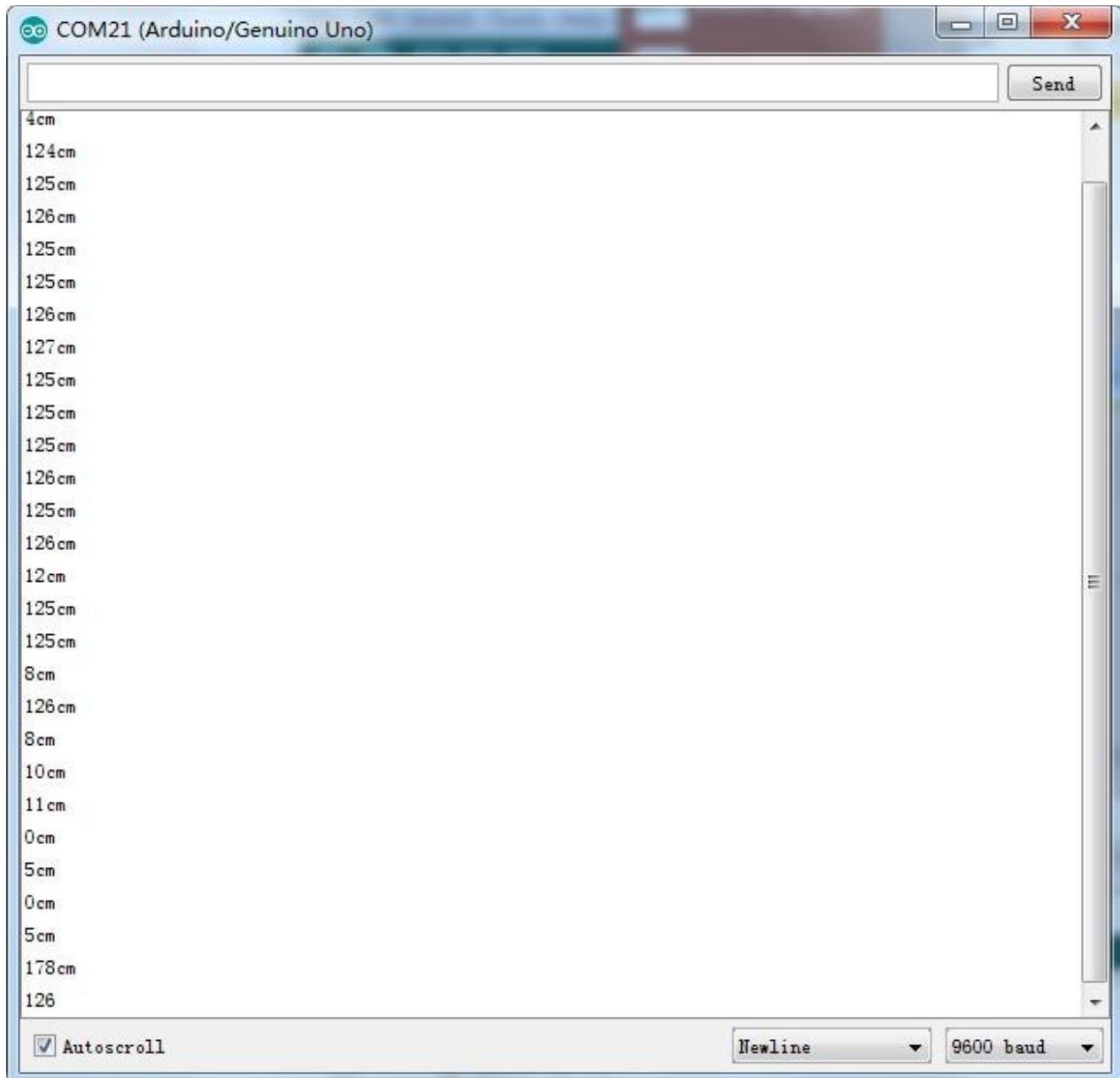
Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d'esempio



Apri il monitor seriale e vedrai i dati come quelli d'esempio qui sotto:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.



The screenshot shows a Windows-style application window titled "COM21 (Arduino/Genuino Uno)". The main window contains a scrollable list of text entries representing sensor measurements in centimeters. The entries are as follows:

- 4cm
- 124cm
- 125cm
- 126cm
- 125cm
- 125cm
- 126cm
- 127cm
- 125cm
- 125cm
- 125cm
- 126cm
- 125cm
- 126cm
- 126cm
- 125cm
- 125cm
- 125cm
- 126cm
- 12cm
- 125cm
- 125cm
- 8cm
- 126cm
- 8cm
- 10cm
- 11cm
- 0cm
- 5cm
- 0cm
- 5cm
- 178cm
- 126

At the bottom of the window, there are three control buttons: "Autoscroll" (checked), "Newline" (dropdown menu), and "9600 baud" (dropdown menu).

## Lezione 11 Modulo Switch a Membrana

### Introduzione

In questo progetto vedremo come integrare un tastierino con la nostra scheda UNO R3 in modo che essa possa leggere i tasti premuti dall'utente sul tastierino.

I tastierini sono utilizzati in tutti i tipi di dispositivi, inclusi telefoni cellulari, macchine per fax, microonde, forni, lucchetti per porte etc. essi sono praticamente ovunque, moltissimi dispositivi elettronici li utilizzano per permettere all'utente di inserire dei valori.

In questo modo sapendo connettere il tastierino al nostro microcontrollore UNO R3, saremo in grado di costruire un sacco di prodotti commerciali.

Alla fine della lezione, quando tutto sarà connesso a dovere e programmato, ogni volta che un tasto viene premuto, il valore corrispondente verrà visualizzato sul monitor seriale del tuo computer. Per motivi di semplicità inizieremo a visualizzare semplicemente il tasto premuto sul computer.

Per questo progetto il tastierino che utilizzeremo sarà un tastierino a matrice, questo è un tastierino che segue uno schema di codici che permette di avere numero di output minore rispetto al numero di tasti presenti sul tastierino. Per esempio la matrice che noi usiamo ha 16 tasti (0-9, A-D, \*, #) ma solamente 8 pin di output. Con un tastierino lineare, avremmo necessitato 17 pin di output (uno per ogni tasto più uno per la messa a terra) per farlo funzionare.

Lo schema di codifica della matrice permette di avere meno pin di output e perciò meno connessioni da fare per permettere il corretto funzionamento del tastierino.

In questo modo, questi tastierini sono molto più efficienti di quelli lineari avendo meno connessioni da effettuare.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Tastierino (Membrane switch module)
- (8) x Connatori M-M (Connatori Maschio-Maschio)



## Connessioni

### Schema

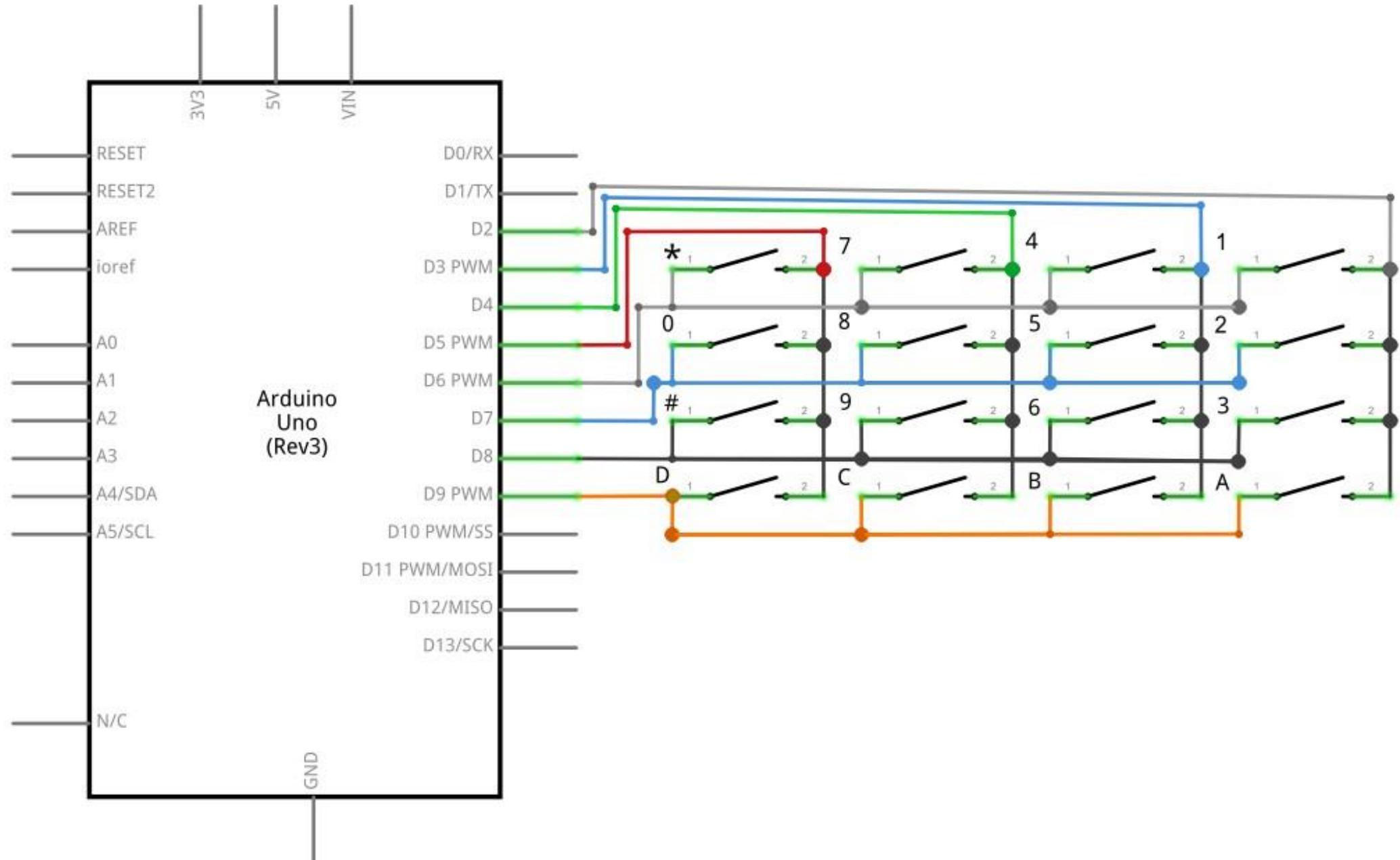
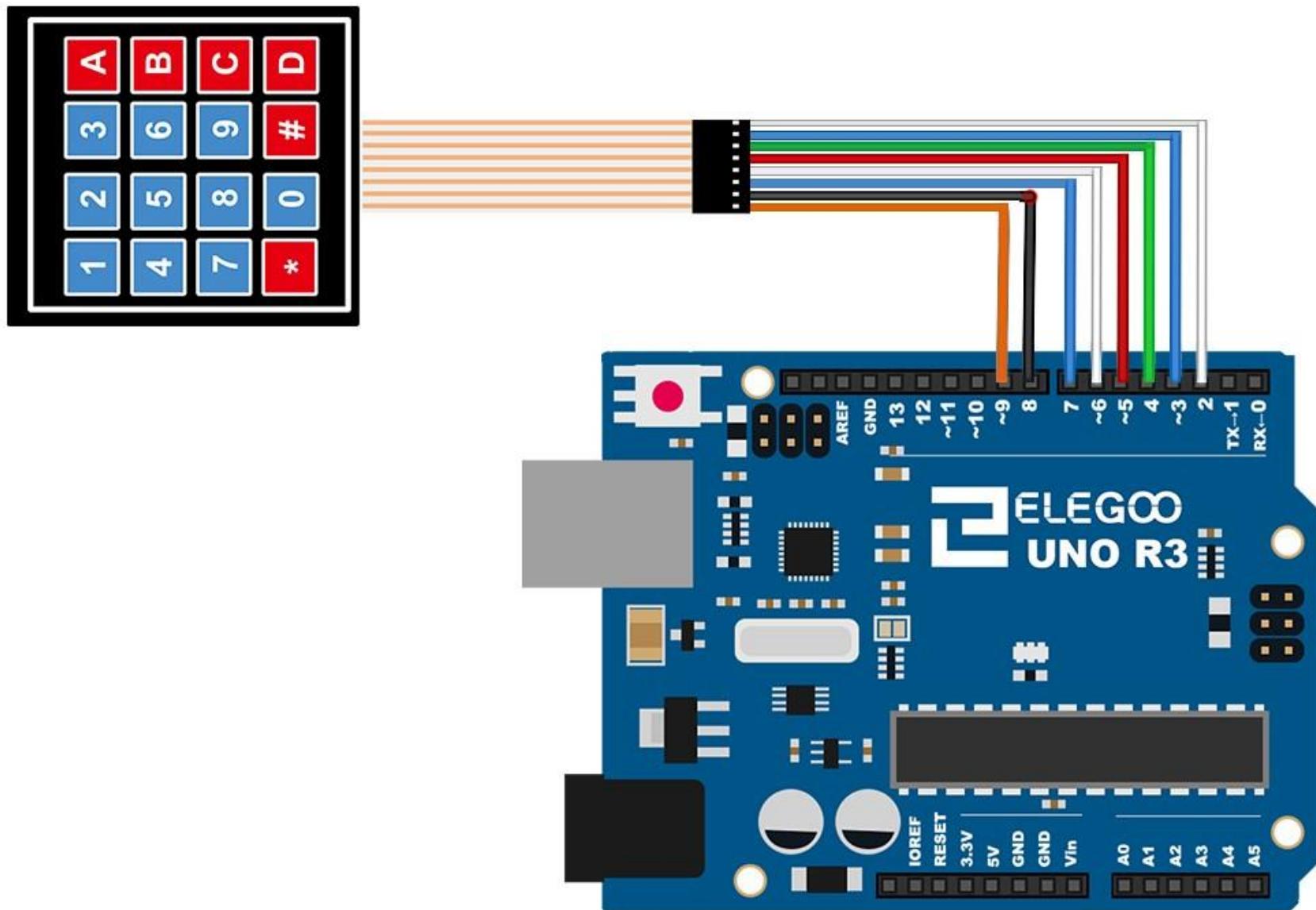


Diagramma di Collegamento



Quando si connettono i pin alla scheda UNO i pin dovranno essere connessi ai pin di output digitale D9-D2. Connetteremo il primo pin del tastierino al D9, il secondo al pin D8, il terzo pin al D7, il quarto pin al D6, il quinto pin al D5, il sesto pin al D4, il settimo pin al D3 e infine l'ottavo pin al D2.

Questa è la tabella riassuntiva delle connessioni:

Keypad Pin	Connects to Arduino Pin...
1	D9
2	D8
3	D7
4	D6
5	D5
6	D4
7	D3
8	D2

## Codice

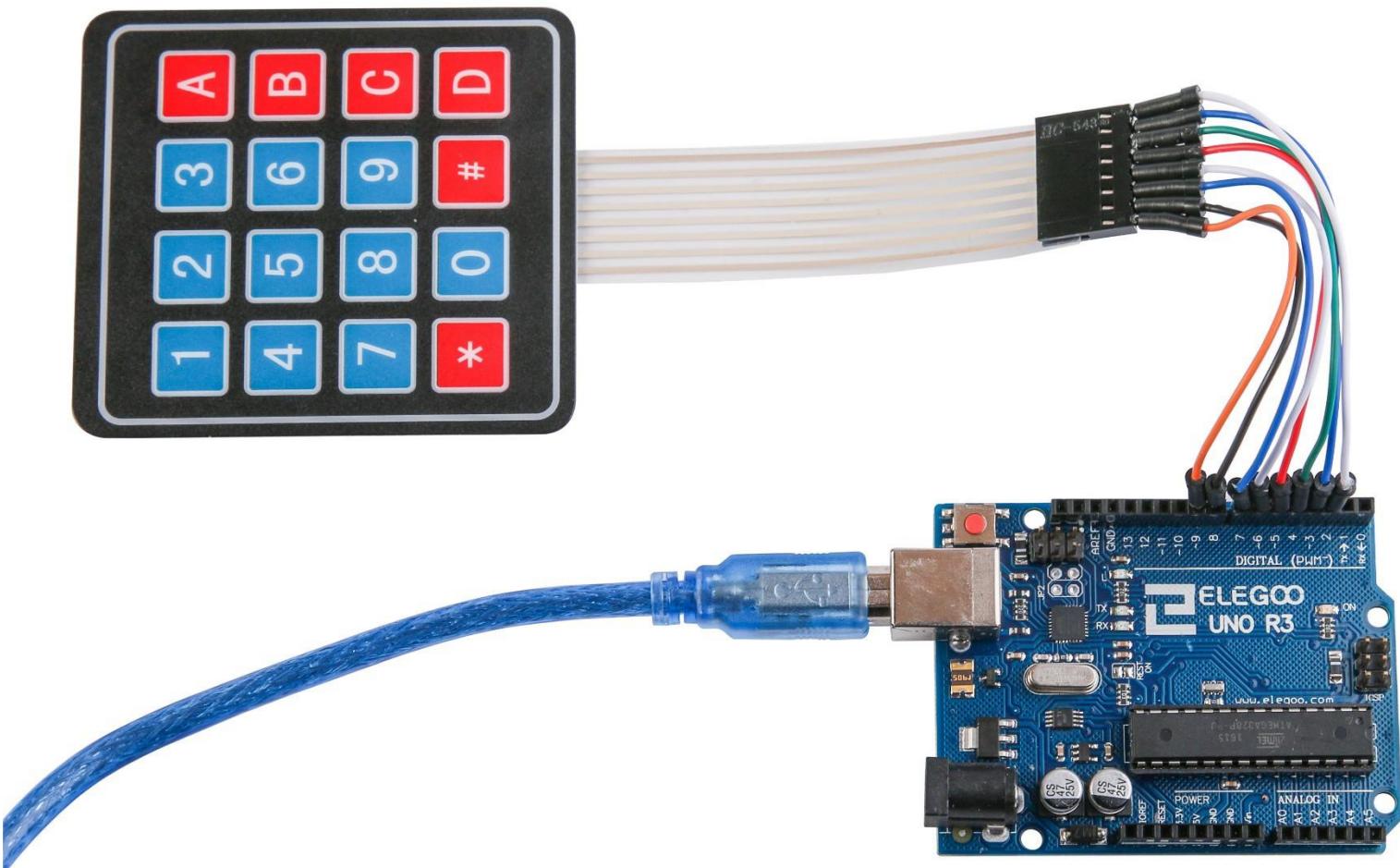
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 11 Membrane Switch Module”, clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria < Keypad > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

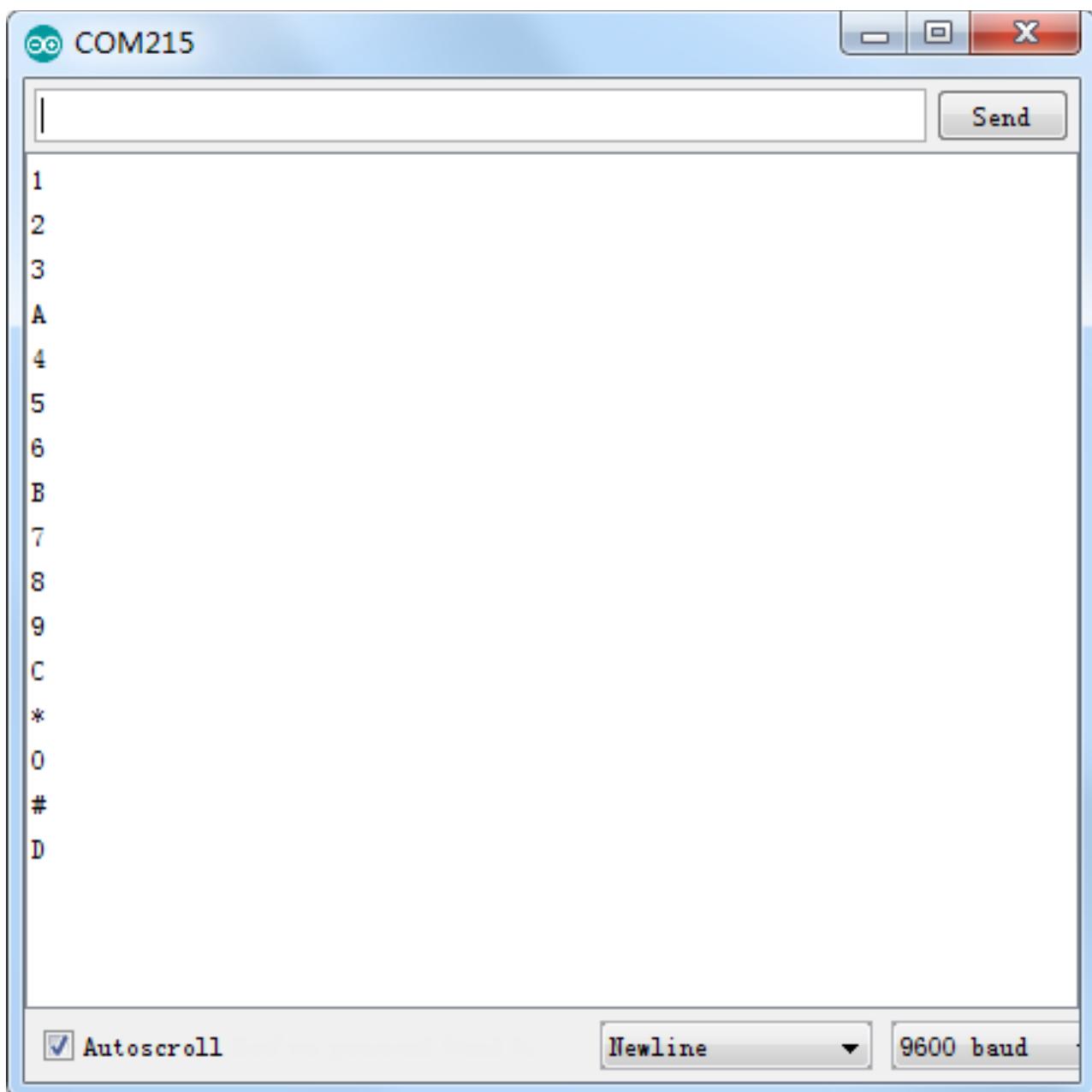
Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d'esempio



Dopo aver compilato e caricato il codice sulla scheda UNO R3, con questo codice ogni volta che si premerà un tasto sul tastierino, verrà visualizzato sul monitor seriale del pc.

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.



# Lezione 12 Sensore di Temperatura e umidità DHT11

## Introduzione

In questo tutorial imparerai ad utilizzare il sensore di temperatura ed umidità DHT11.

Il sensore è abbastanza accurato per la maggior parte dei progetti che necessitano di tenere traccia delle letture di temperatura ed umidità.

Anche questa volta useremo una libreria specificatamente scritta per questi sensori che accorcerà il codice e ne renderà facile la scrittura.

## Componenti Richiesti:

(1) x Elegoo UNO R3

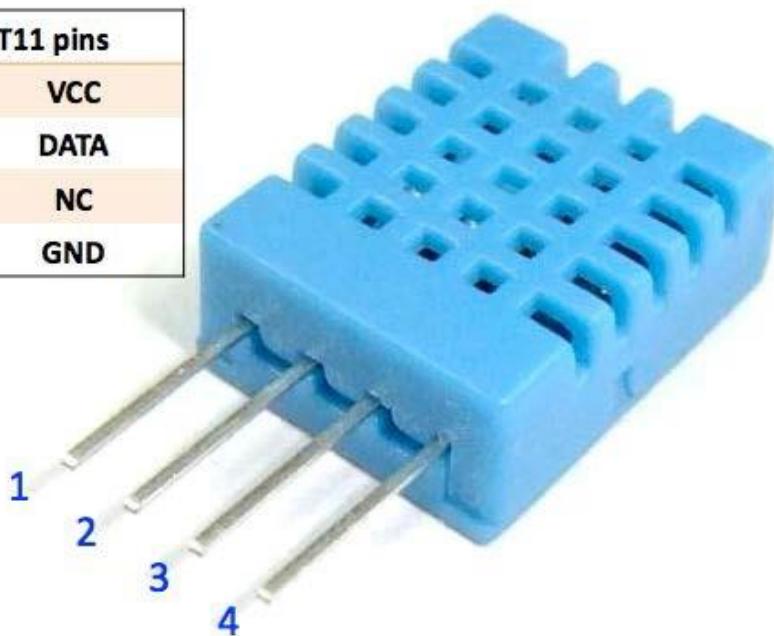
(1) x Modulo rilevatore di Temperatura ed umidità DHT11

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

## Introduzione ai Componenti

### Sensore di temperatura ed umidità

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Il sensore di temperatura ed umidità DHT11 è un sensore composto che permette di avere in output segnali digitali calibrati di temperatura ed umidità

Questo modulo digitale contiene una tecnologia che permette di assicurare eccellente affidabilità e stabilità per un lungo periodo.

Il sensore include un sensore resistivo di umidità e un dispositivo NTC di misura della temperatura, entrambi connessi con un microcontrollore ad alte performance ad 8-bit

Applicazioni: HVAC, deumidificatori, test e ispezione di equipaggiamenti, beni di consumo, automotive, controlli automatici, data logger, stazioni meteo, applicazioni casalinghe, regolatori di umidità, medico e altre misure e controllo di umidità.

Parametri del prodotto:

Relativi all'umidità:

Risoluzione: 16Bit

Ripetibilità:  $\pm 1\%$  RH

Accuratezza: At  $25^{\circ}\text{C}$   $\pm 5\%$  RH

Intercambiabilità: pienamente intercambiabile

Tempo di risposta: 1 / e (63%) of  $25^{\circ}\text{C}$  6s

1m / s air 6s

Isteresi:  $<\pm 0.3\%$  RH

Stabilità a lungo termine:  $<\pm 0.5\%$  RH / anno

Temperatura:

Risoluzione: 16Bit

Ripetibilità:  $\pm 0.2^{\circ}\text{C}$

Range: a  $25^{\circ}\text{C}$   $\pm 2^{\circ}\text{C}$

Tempo di risposta: 1 / e (63%) 10S

Caratteristiche Elettriche

Voltaggio richiesto: DC  $3.5 \sim 5.5\text{V}$

Corrente richiesta: misurata  $0.3\text{mA}$  standby  $60\mu\text{A}$

Periodo di campionatura: maggiore di 2 secondi

Descrizione dei Pin:

1, Voltaggio VDD (+)  $3.5 \sim 5.5\text{V}$  DC

2 DATA dati seriali, bus singolo

3, NC, pin vuoto

4, GND messa a terra (-)

## Connessioni

### Schema

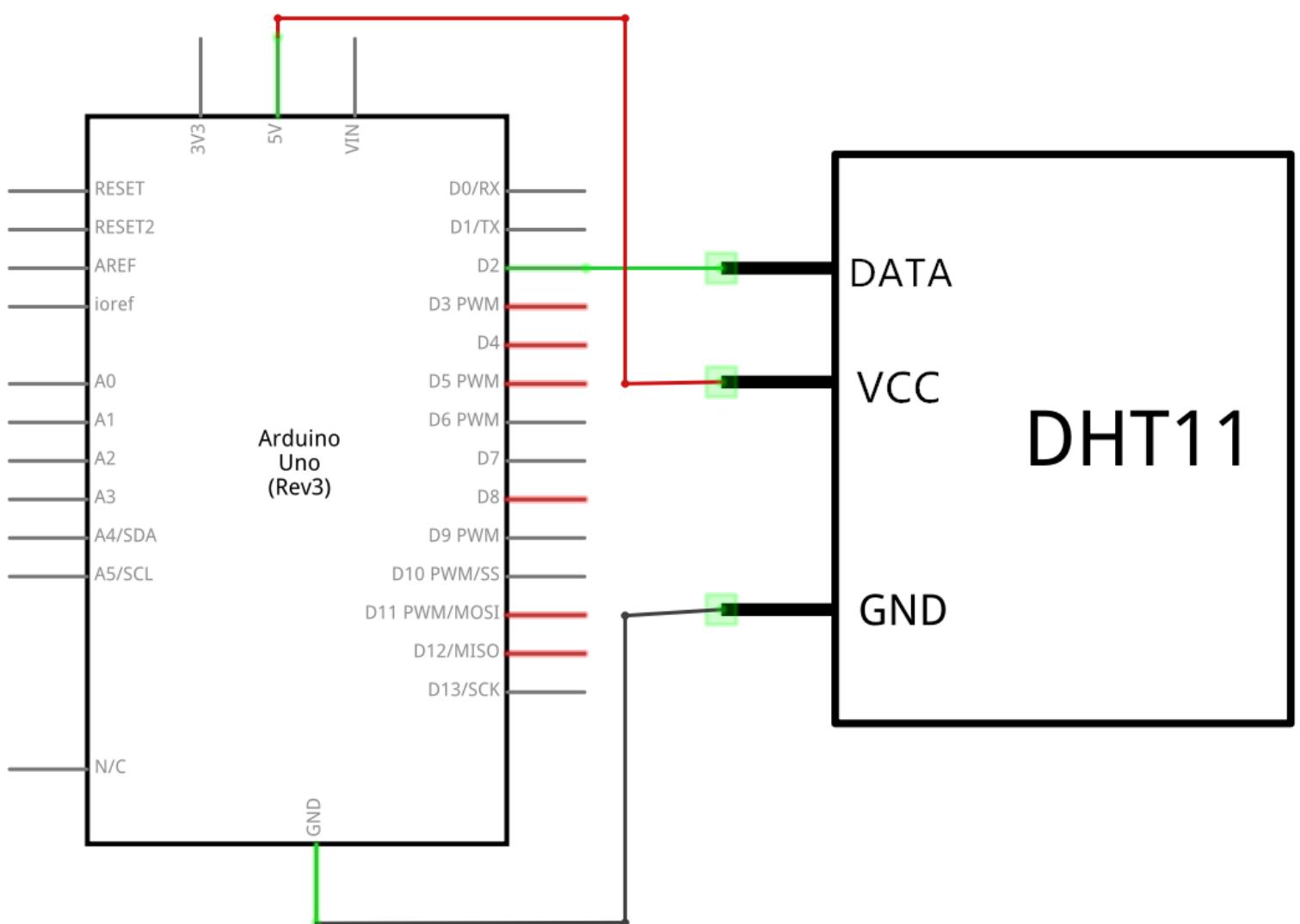
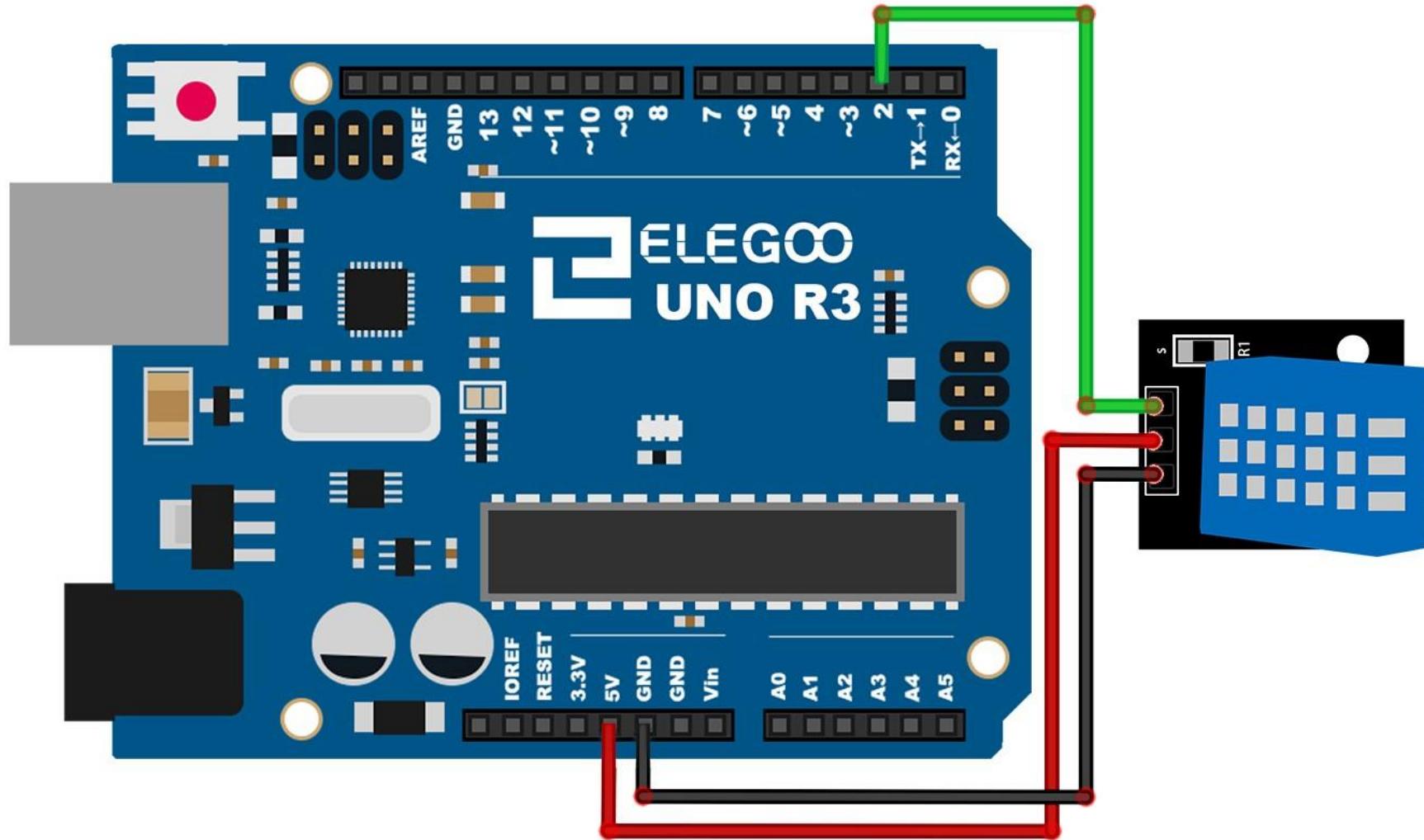


Diagramma dei collegamenti



Come puoi vedere per il sensore sono necessarie solo 3 connessioni, in quanto un pin non viene utilizzato.

Le connessioni sono Voltaggio, messa a terra e segnale. L'ultimo può essere connesso ad un qualsiasi pin sulla nostra scheda UNO

## Codice

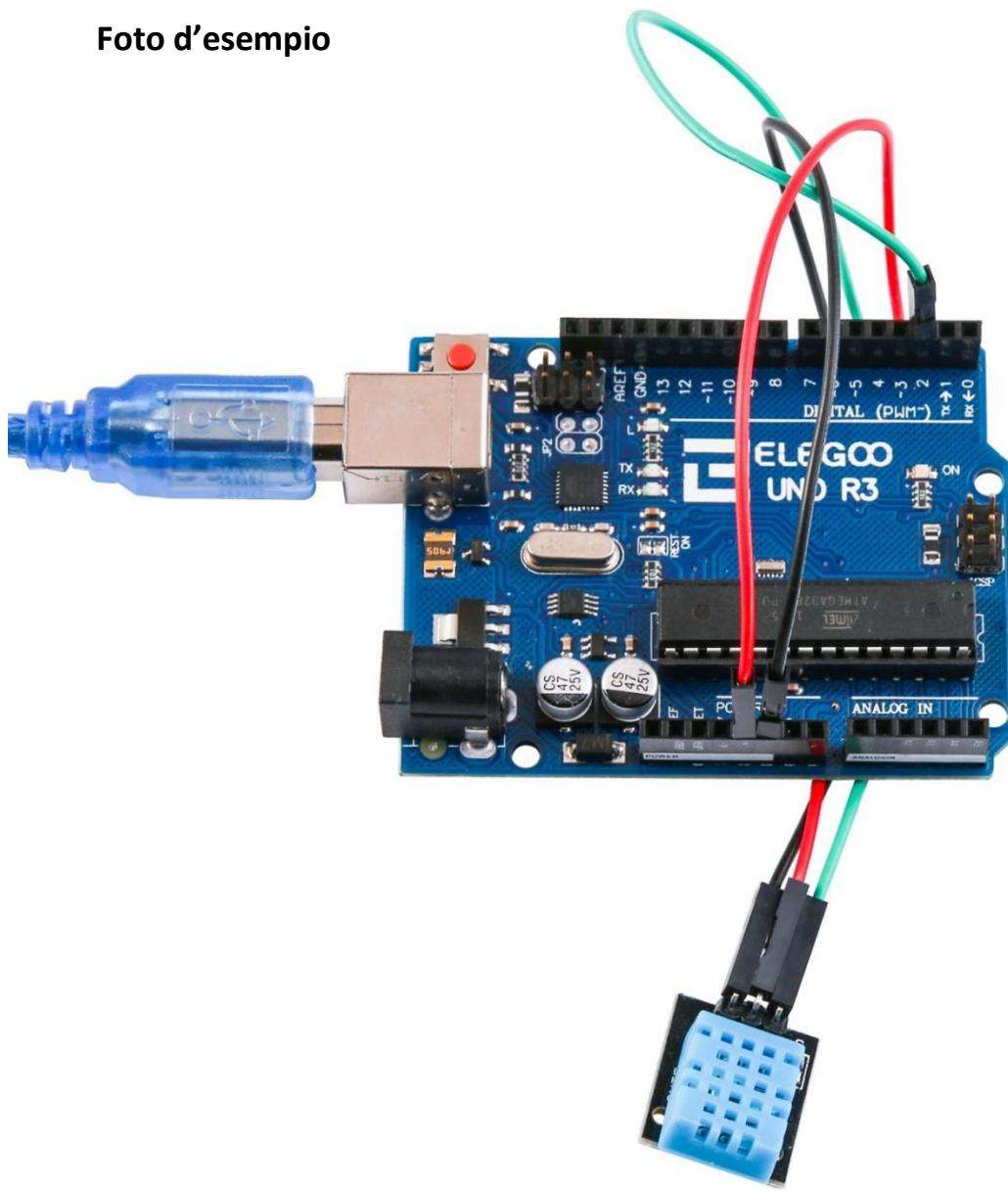
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 12 DHT11 Temperature and Humidity Sensor”, clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria <SimpleDHT> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

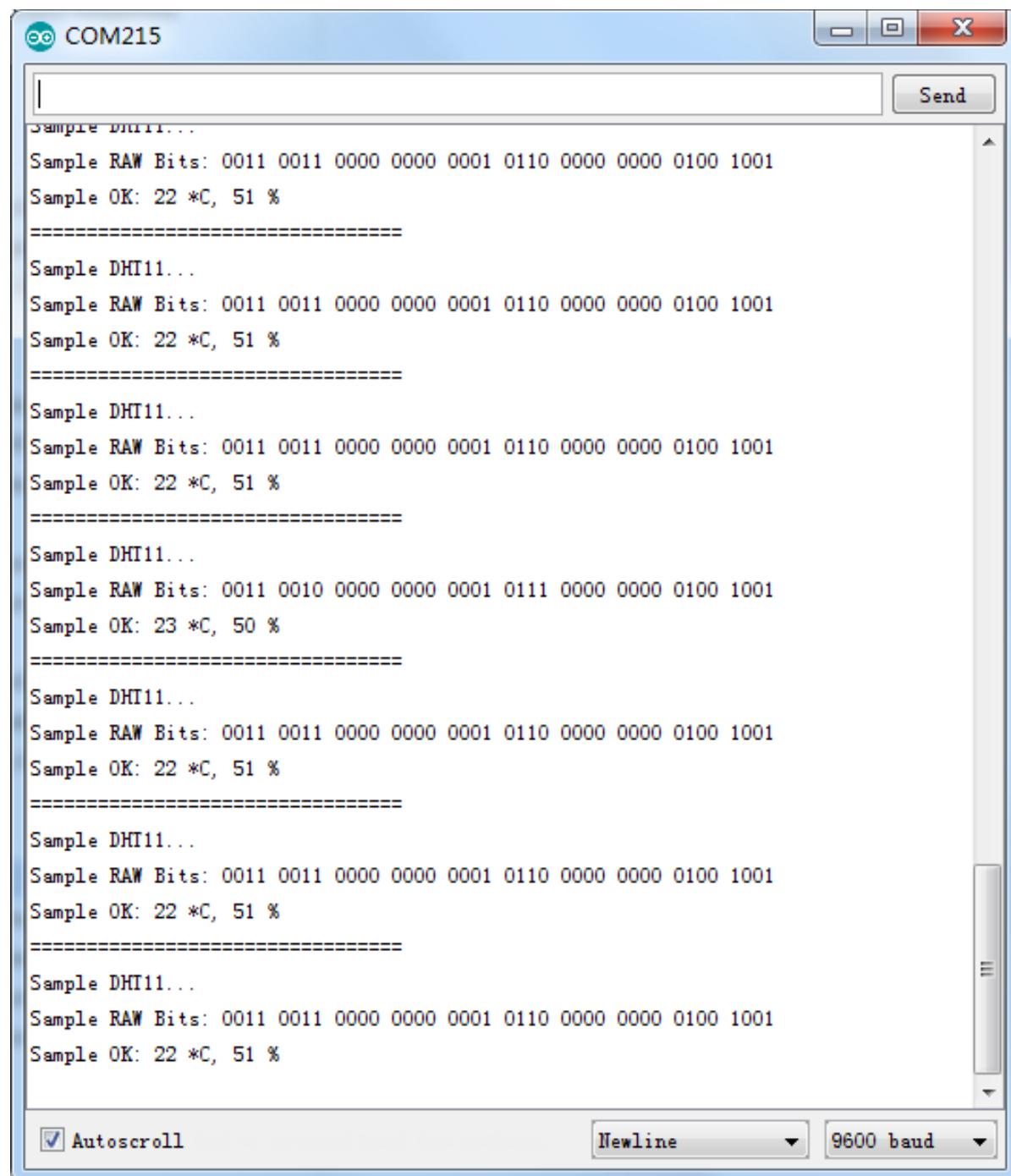
## Foto d’esempio



Carica il programma e poi apri il monitor seriale, potrei vedere i dati come qui sotto:

(viene visualizzata la temperatura dell'ambiente, nel nostro esempio sono 22 gradi)

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.



The screenshot shows a Windows-style application window titled "COM215". The main area displays a series of sensor readings from a DHT11 module. Each reading consists of a header, raw binary data, and a temperature/humidity pair. The headers are "Sample DHT11...", followed by "=====". The raw data is shown as a sequence of 16 binary digits (0 or 1). The temperature and humidity values are given as "Sample OK: 22 \*C, 51 %". The window has standard operating system controls (minimize, maximize, close) at the top right. At the bottom, there are three configuration options: "Autoscroll" (checked), "Newline" (dropdown menu), and "9600 baud" (dropdown menu).

```
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0010 0000 0000 0001 0111 0000 0000 0100 1001
Sample OK: 23 *C, 50 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %

=====
```

# Lezione 13 Modulo Joystick Analogico

## Introduzione

I joystick analogici sono un ottimo modo per aggiungere controlli al tuo progetto.

In questo tutorial imparerai ad utilizzare il modulo del Joystick analogico.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo Joystick
- (5) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

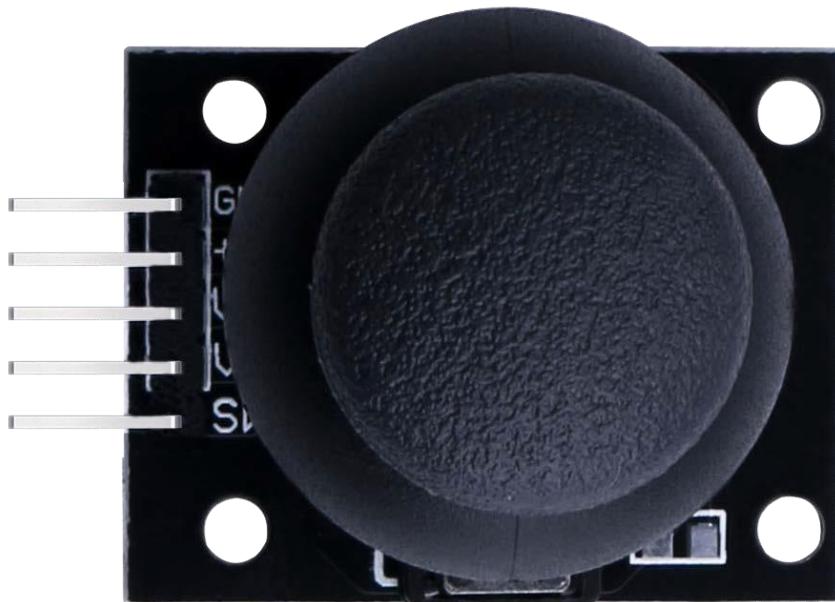
## Introduzione ai Componenti

### Joystick

Il modulo ha 5 pin: VCC, Ground, X, Y, Key. Nota che le scritte sul modulo potrebbero essere leggermente differenti in base a dove hai comprato il modulo.

Il joypad per il pollice è analogico e può fornire letture più accurate di un semplice joystick direzionale il quale solitamente utilizza qualche tipo di bottone o interruttore meccanico. Inoltre, puoi premere il joystick come se fosse un semplice bottone, il quale invierà un segnale di pressione.

In questo esempio utilizzeremo due pin analogici per leggere i dati di X e Y, un terzo pin analogico verrà utilizzato per leggere le pressioni del joystick, o altrimenti può essere lasciato sconnesso. Per ricevere letture stabili dal pin KEY>Select, è necessario connettere la tensione VCC tramite una resistenza di pull-up. Il resistore interno della scheda Arduino presente nei pin digitali può essere utilizzato a questo scopo.



## Connessioni

### Schema

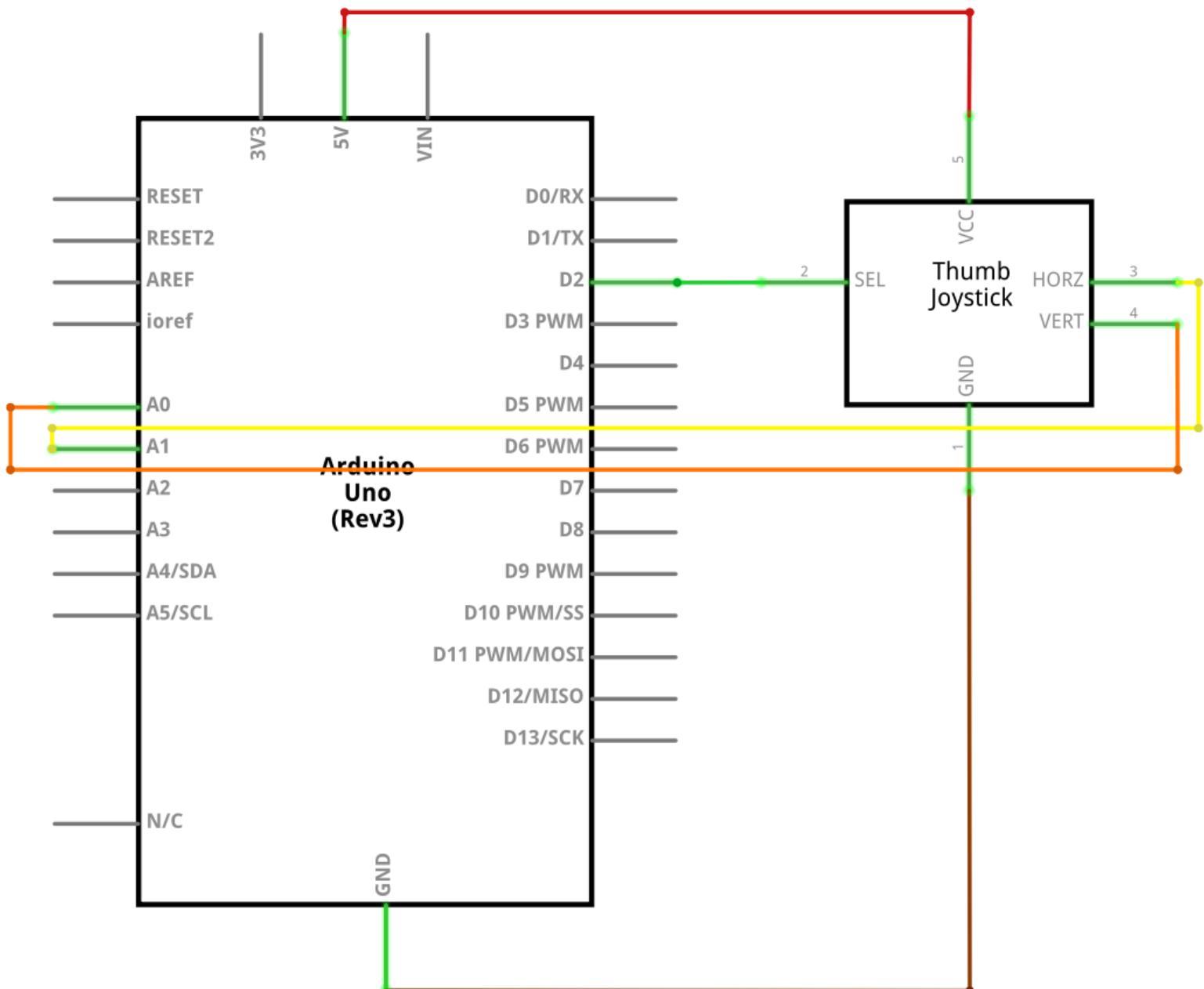
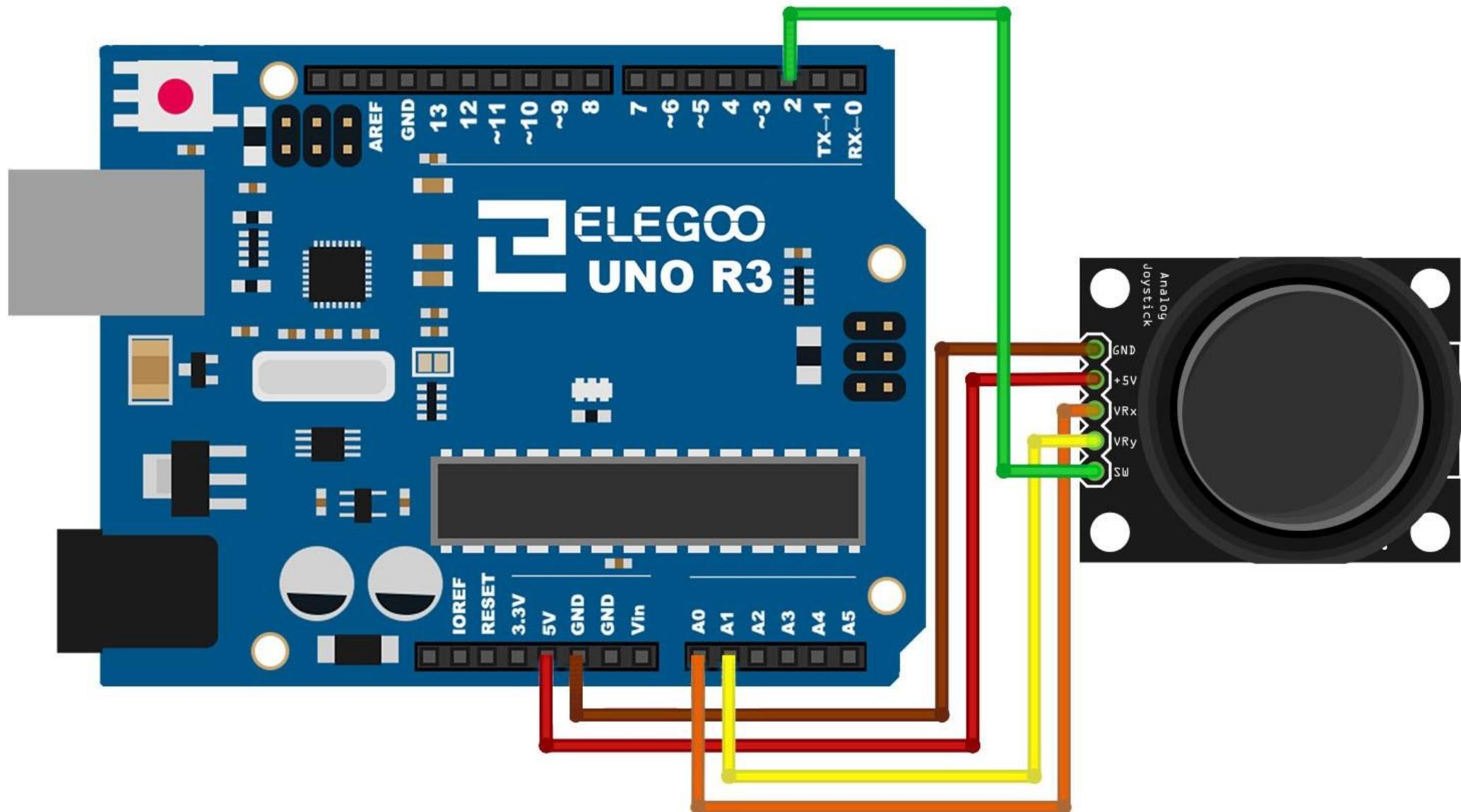


Diagramma di collegamento



Necessitiamo 5 connessioni al joystick.

Le connessioni sono: Key, Y, X, Voltaggio e messa a terra.

Y e X sono analogici mentre Key è digitale. Se non ti serve utilizzare la pressione del joystick puoi semplicemente non utilizzarlo ed utilizzare solamente gli altri 4 pin.

## Codice

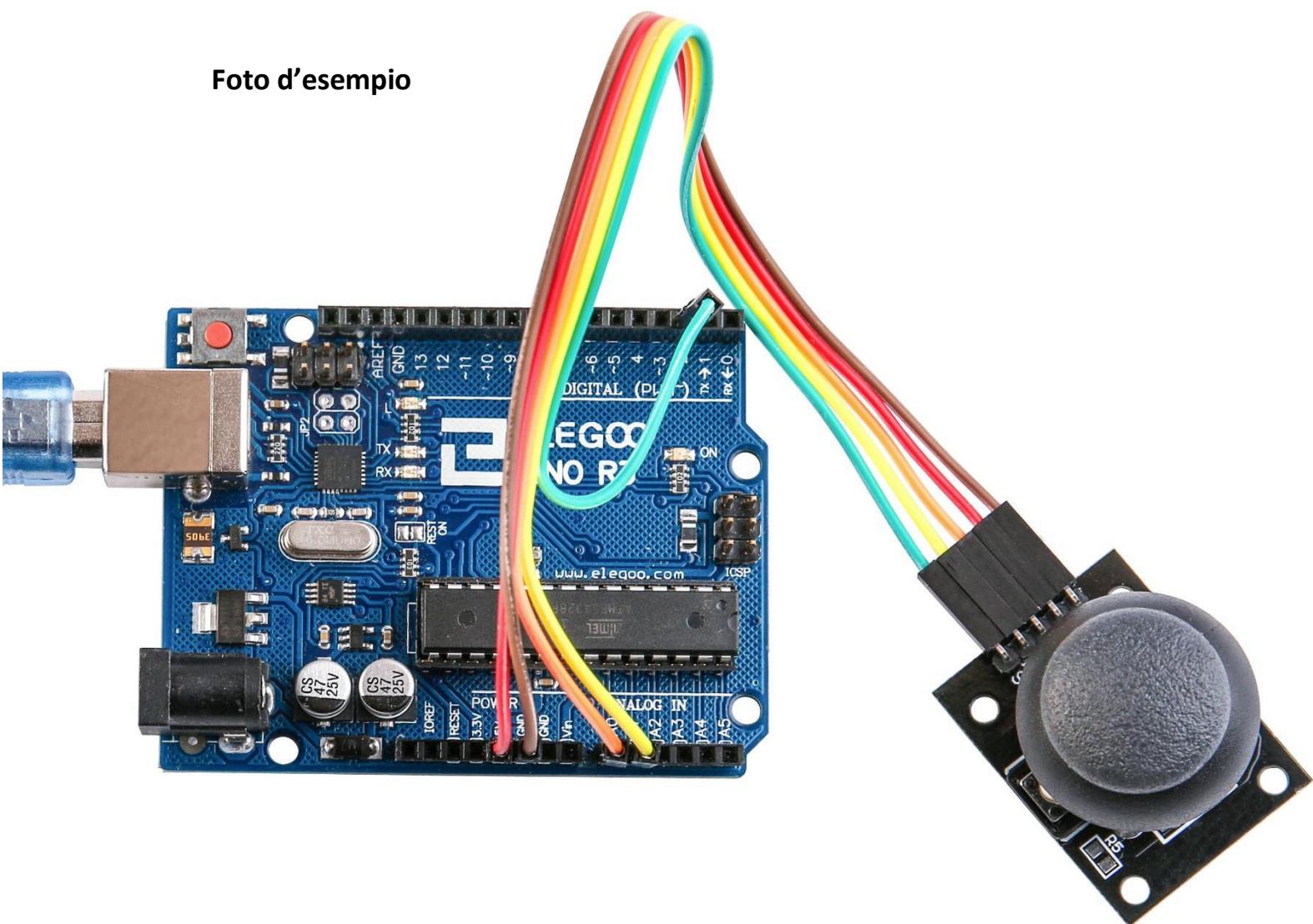
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 13 Analog Joystick Module”, clicca su UPLOAD per caricare il programma.

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

I joystick analogici sono dei semplici potenziometri, perciò restituiscono dei valori analogici.

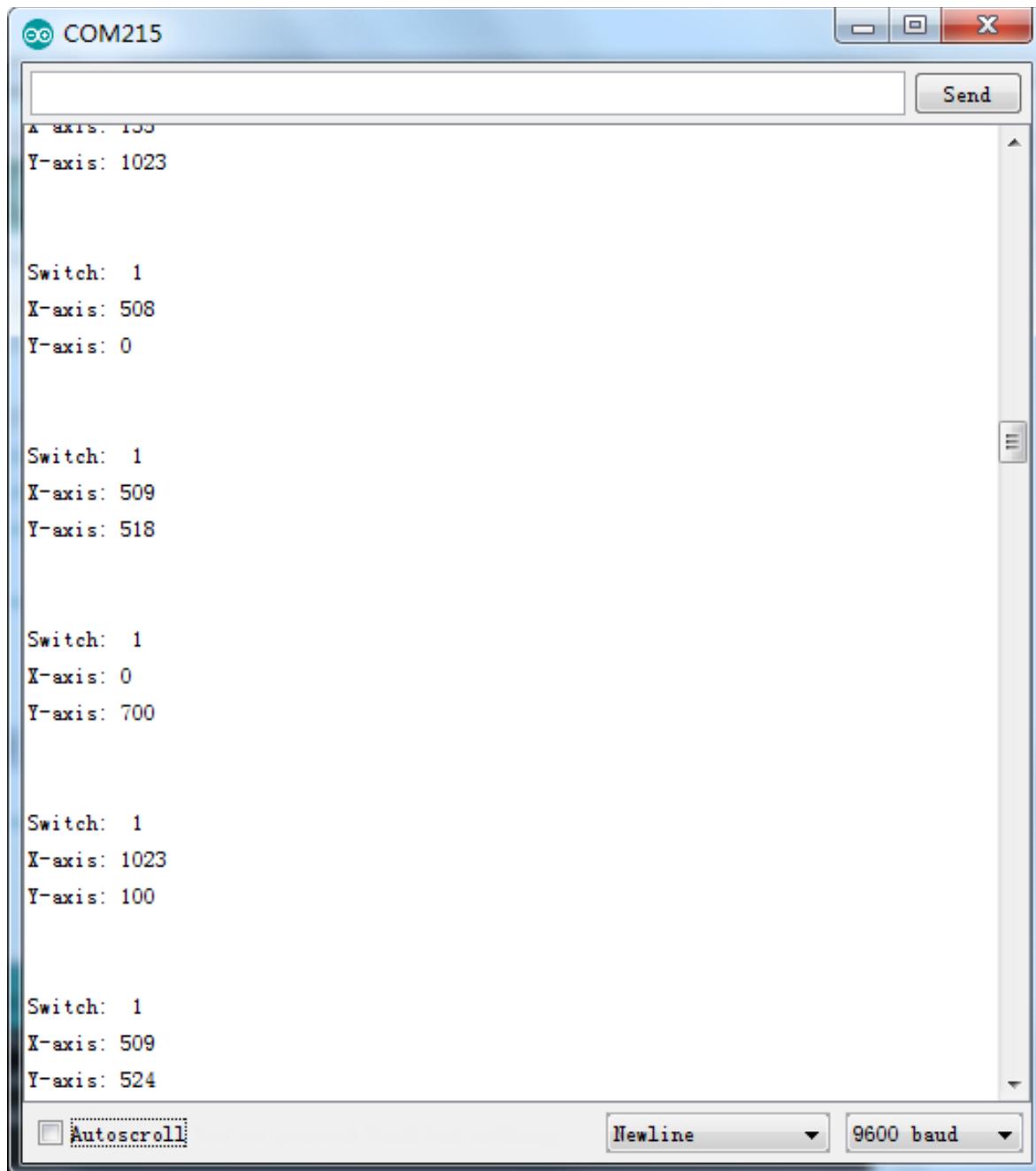
Quando un joystick è nella posizione centrale dovrebbe fornire un valore intorno al 512. Il range di valori varia tra 0 e 1024

## Foto d'esempio



Apri il monitor seriale per vedere dei dati simili ai seguenti:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.



The screenshot shows a Windows-style application window titled "COM215". The main area contains several lines of text representing sensor data. The data is organized into five distinct sections, each starting with "Switch: 1" followed by "X-axis" and "Y-axis" values. The sections are separated by blank lines. The bottom of the window features a toolbar with buttons for "Autoscroll", "Newline", and "9600 baud".

```
Switch: 1
X-axis: 100
Y-axis: 1023

Switch: 1
X-axis: 508
Y-axis: 0

Switch: 1
X-axis: 509
Y-axis: 518

Switch: 1
X-axis: 0
Y-axis: 700

Switch: 1
X-axis: 1023
Y-axis: 100

Switch: 1
X-axis: 509
Y-axis: 524
```

Autoscroll      Newline      9600 baud

## Lezione 14 Modulo di Ricezione IR

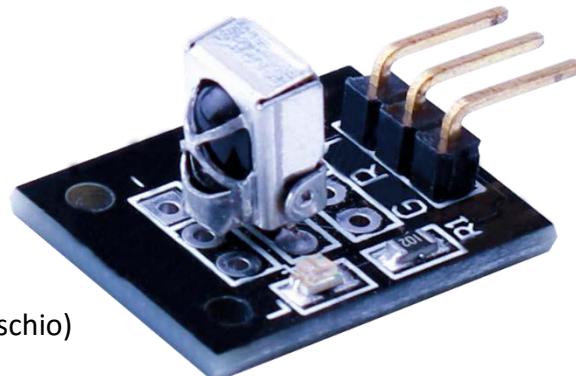
### Introduzione

Utilizzare un modulo ad infrarossi (IR) è un ottimo modo per controllare da remoto i tuoi progetti. I moduli ad infrarossi sono molto semplici da utilizzare. In questo tutorial connetteremo il ricevitore IR alla nostra scheda UNO, poi utilizzeremo una libreria scritta appositamente per questo sensore.

Nel nostro file di codice sono presenti tutti i codici esadecimali che sono disponibili nel telecomando ad infrarossi. Il codice sarà anche in grado di riconoscere se un tasto riceve una pressione prolungata.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Ricevitore IR
- (1) x Telecomando IR
- (3) x F-M cavi (Cavi di DuPont Femmina-Maschio)



### Introduzione ai componenti

#### Sensore ricevitore di infrarossi:

I rilevatori di infrarossi sono piccoli microchips con una fotocellula sensibile alla luce infrarossa. Questi sensori sono quasi sempre utilizzati per il controllo remoto – tutte le TV e lettori DVD hanno uno di questi sensori sulla parte frontale per ricevere i comandi inviati dal telecomando. All'interno dei telecomandi è presente il led ad infrarossi che emette pulsazioni alla giusta frequenza di luce che comanda al televisore di accendersi, spegnersi o cambiare canale. La luce ad infrarossi non è visibile agli occhi umani. Questo significa che il sensore necessita di un po' di test prima di essere impostato correttamente.

Ci sono alcune differenze tra questi sensori ed i fotoresistori (CdS):

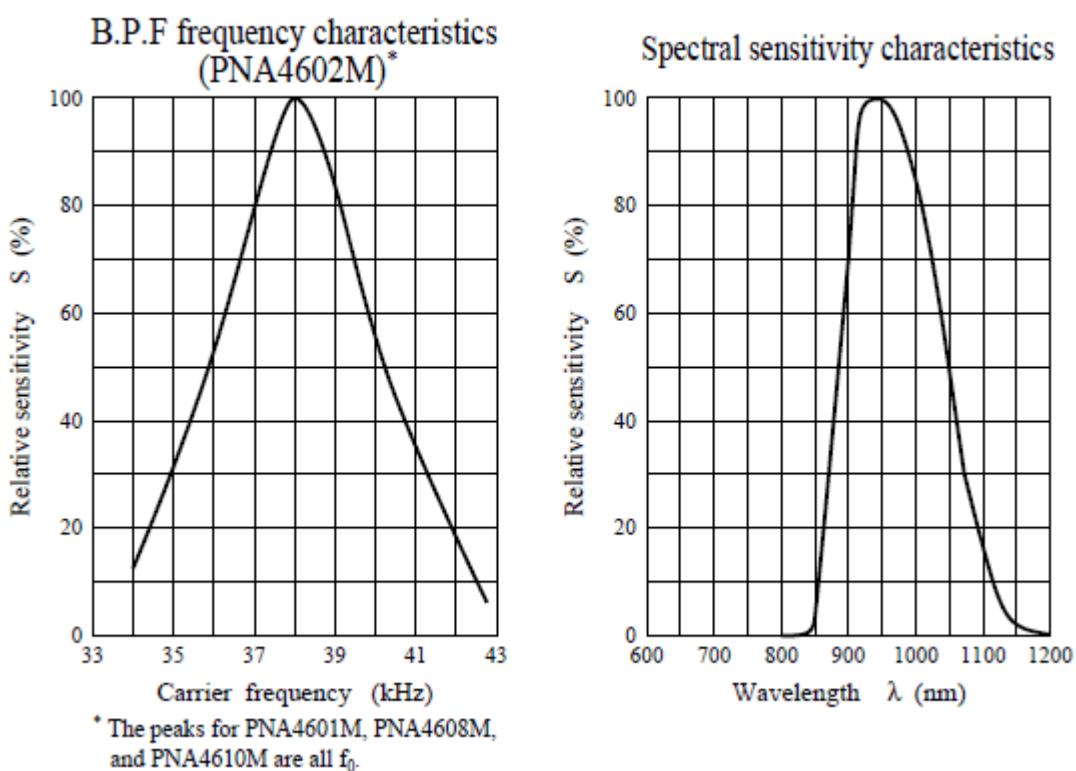
I rilevatori ad infrarossi hanno speciali filtri per la luce ad infrarossi, essi non sono ottimi per rilevare la luce visibile. D'altro canto, i fotoresistori sono ottimi per rilevare la luce gialla/verde ma non lo sono per la luce ad infrarossi.

I rilevatori ad infrarossi hanno al loro interno un demodulatore che rileva la luce ad infrarossi a 38kHz. Un semplice led ad infrarossi non viene rilevato, necessita una modulazione PWM a 38kHz.

I fotoresistori al contrario non hanno nessun tipo di demodulatore e possono rilevare qualsiasi tipo di frequenza (compresa al corrente continua), limitati solo della velocità di risposta del fotoresistore (che è circa 1kHz)

I rilevatori ad infrarossi hanno un uscita digitale – se rilevano un segnale infrarosso a 38kHz portano l'uscita a LOW (0V) oppure non rilevano nessun segnale e portano l'uscita a HIGH (5V). I fotoresistori invece hanno un comportamento simile alle resistenze, cambiano il valore di resistenza in base alla quantità di luce alla quale sono esposti.

### Cosa puoi misurare



Come puoi nel grafico dei dati qui sopra, il picco di sensibilità si trova a 38 kHz ed il picco di illuminazione del led è a 940 nm. Si possono usare frequenze tra i 35 kHz ed i 41 kHz ma avremo una notevole riduzione di sensibilità e quindi una riduzione di sensibilità rispetto al caso precedente.

Allo stesso modo si possono utilizzare LED con una frequenza d'onda compresa tra 850 nm e 1100 nm ma non funzioneranno bene come quelli con frequenze d'onda comprese tra 900 nm e 1000 nm, perciò fai attenzione a scegliere i led che si accoppino a dovere. Controlla il datasheet (scheda dati) dei tuoi led quando per verificare la lunghezza d'onda. Cerca di acquistarli a 940 nm e ricorda che 940 nm è una luce non visibile all'occhio umano.

## Connessione

### Schema

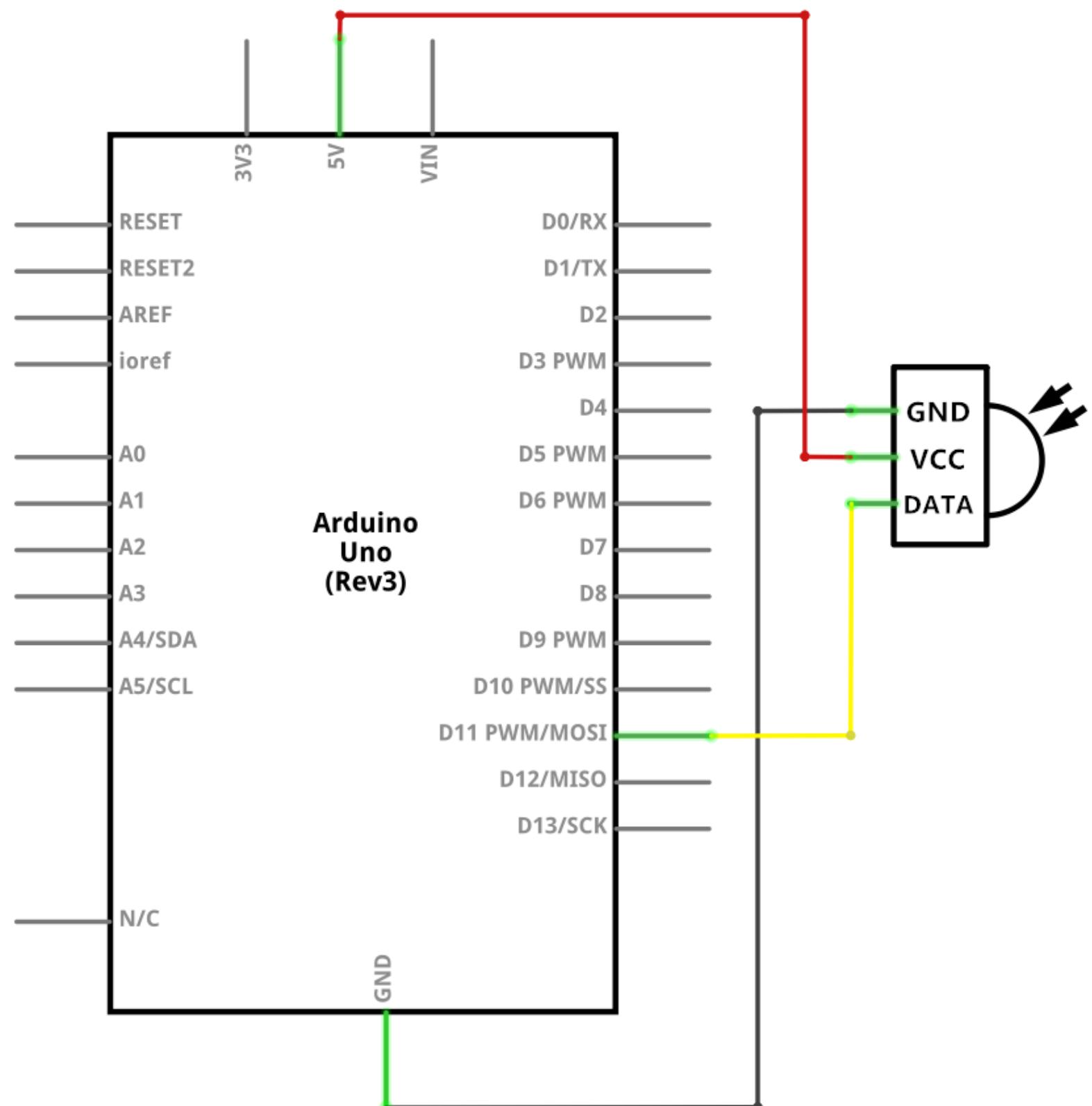
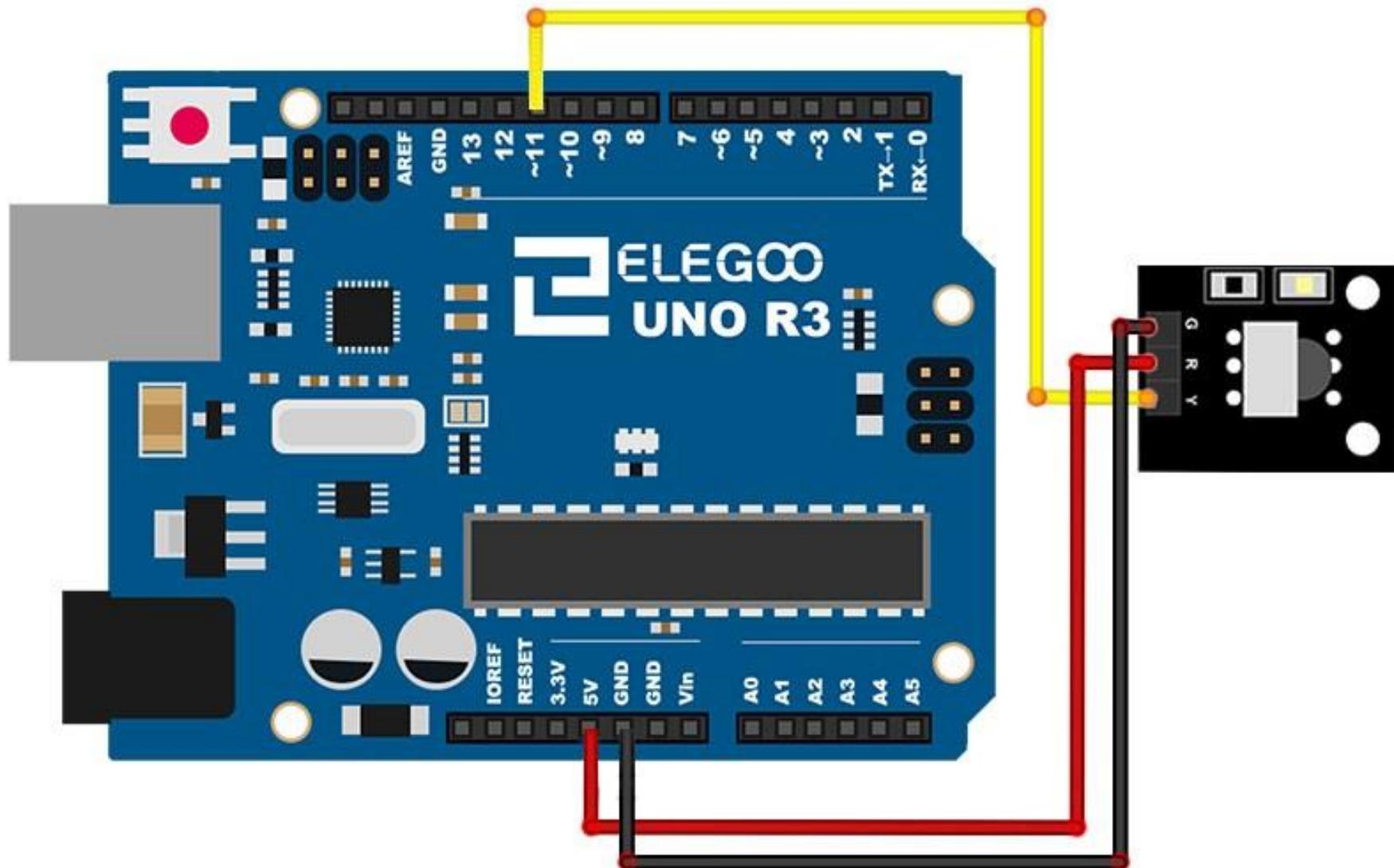


Diagramma di collegamento



Il ricevitore ad infrarossi è dotato di 3 connessioni.

Le connessioni sono: Segnale, Voltaggio, Messa a terra.

Il “-“ è la messa a terra, “S” è il segnale, e il pin in mezzo è il voltaggio 5V.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 14 IR Receiver Module”, 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.

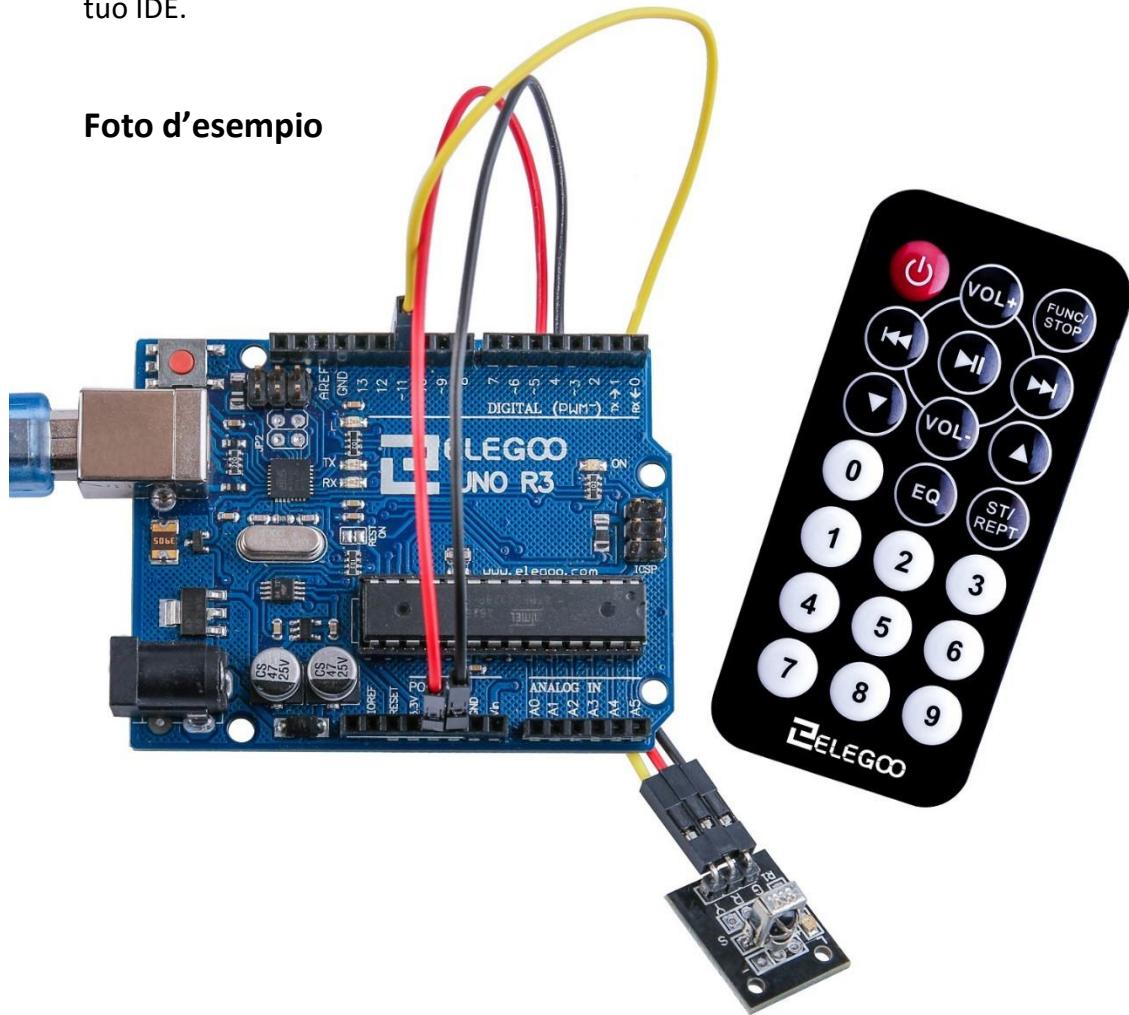
Prima di eseguire questo codice, controlla di aver installato la libreria <IRremote> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

Per prima cosa spostiamo <RobotIRremote> all’esterno della cartella della libreria, va effettuato questo passaggio perché quella libreria è in conflitto con quella che utilizzeremo noi. Puoi semplicemente rispostare la cartella rimossa all’interno della libreria nel momento in cui avrai terminato di programmare il microcontrollore.

Una volta che avrai installato la libreria devi semplicemente procedere e riavviare il tuo IDE.

## Foto d’esempio



Apri il monitor seriale e potrai vedere i dati qui sotto.

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.

The screenshot shows a Windows-style application window titled "COM21 (Arduino/Genuino Uno)". The main text area displays a list of button codes from an IR receiver. The list includes:

- IR Receiver Button Decode
- POWER
- VOL+
- FUNC/STOP
- FAST BACK
- PAUSE
- FAST FORWARD
- DOWN
- VOL-
- UP
- 0
- EQ
- ST/REPI
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

At the bottom of the window, there are three control buttons: "Autoscroll" (checked), "Newline" (dropdown menu), and "9600 baud" (dropdown menu).

# Lezione 15 Modulo Display 8x8 MAX7219 LED Dot Matrix

## Introduzione

In questo tutorial conserveremo il modulo MX7219 e faremo scorrere del testo su di esso. Dato che questo modulo utilizza il chip MAX7219 per comandare i led, saremo in grado tramite esso di accendere e spegnere ogni singolo dei 64 utilizzando solamente 3 pin della nostra schedina UNO.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo Max7219
- (5) x F-M connettori (Connettori di DuPont Femmina-maschio)

## Introduzione al Componente

### Modulo matrice LED MAX7219

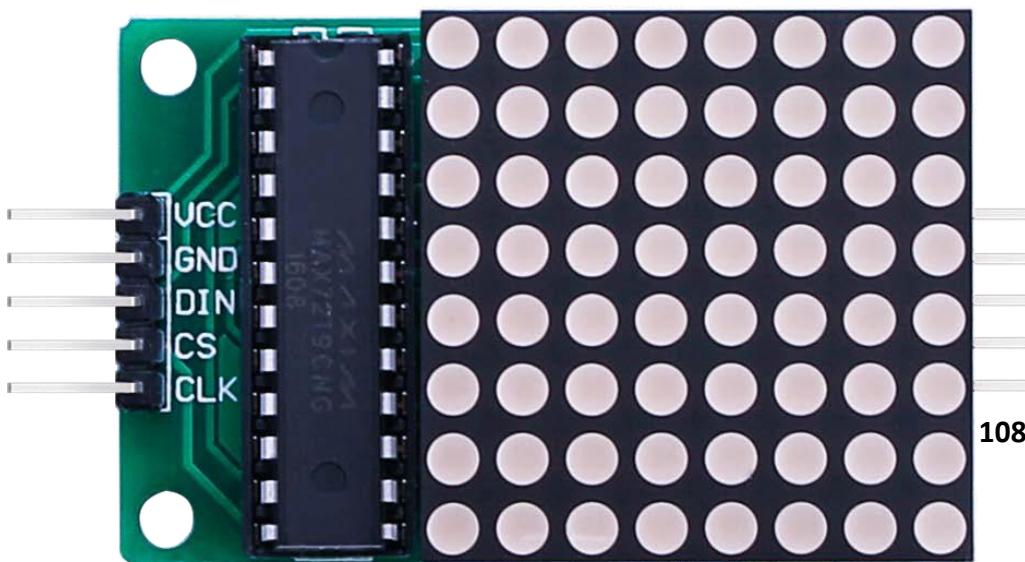
Il nostro progetto, sarà una scheda Arduino collegata in serie con un chip MAX7219 che controllerà la matrice LED. Il circuito integrato MAX7219 è un chip che permette di avere un input e output seriale con il catodo in comune in grado di interfacciarsi con display a 7 segmenti a LED, display con grafici a barre oppure ancora con display composti da 64 led comandati individualmente. Per comodità utilizzeremo una matrice di led 8x8 con integrato il chip MAX7219. Le specifiche tipo del componente in esame (il modulo matrice di led) sono le seguenti:

Range di Voltaggio: DC 4.7V – 5.3V

Voltaggio standard: 5V

Corrente standard: 320mA

Massima corrente operativa: 2A



## Connessioni

### Schema

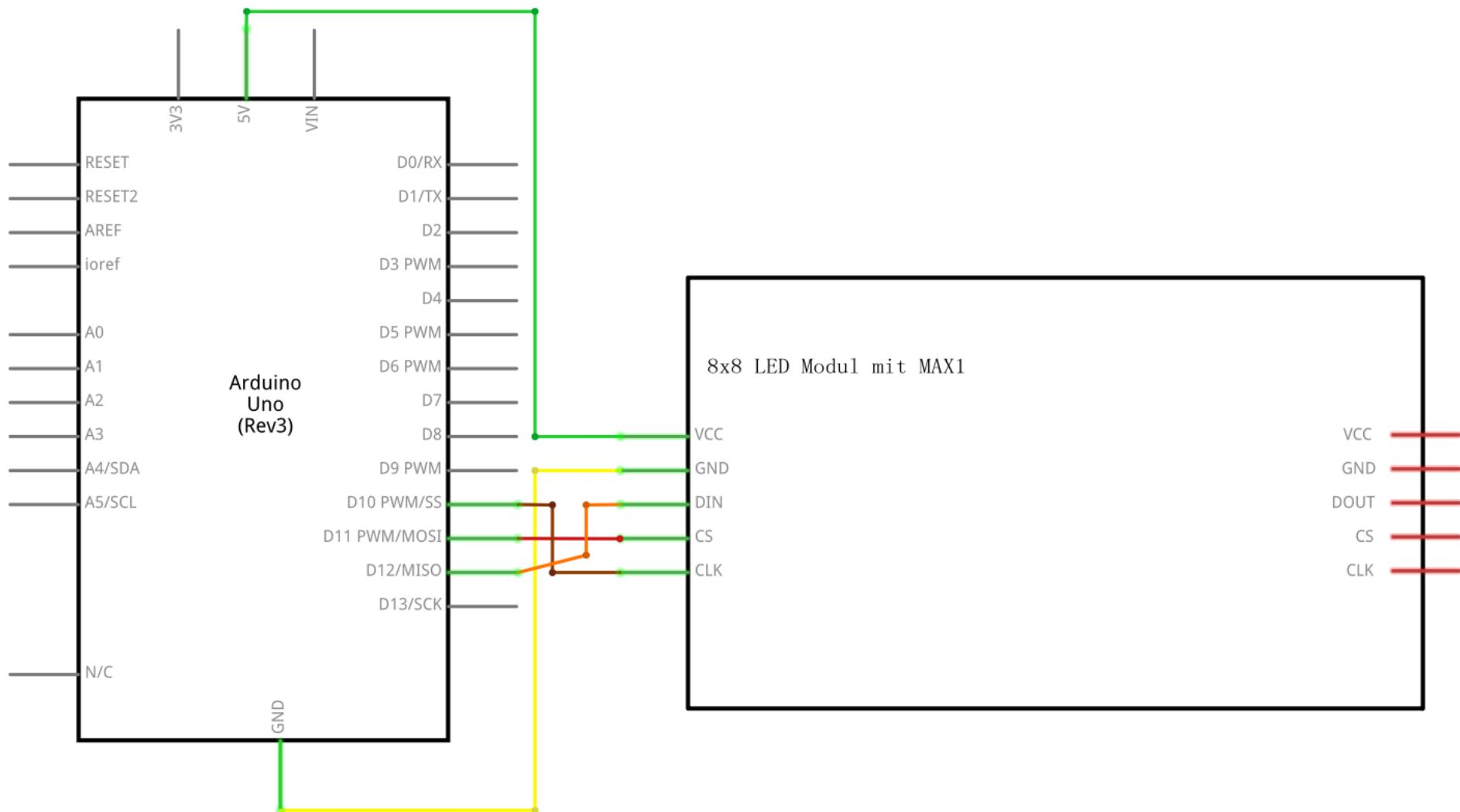
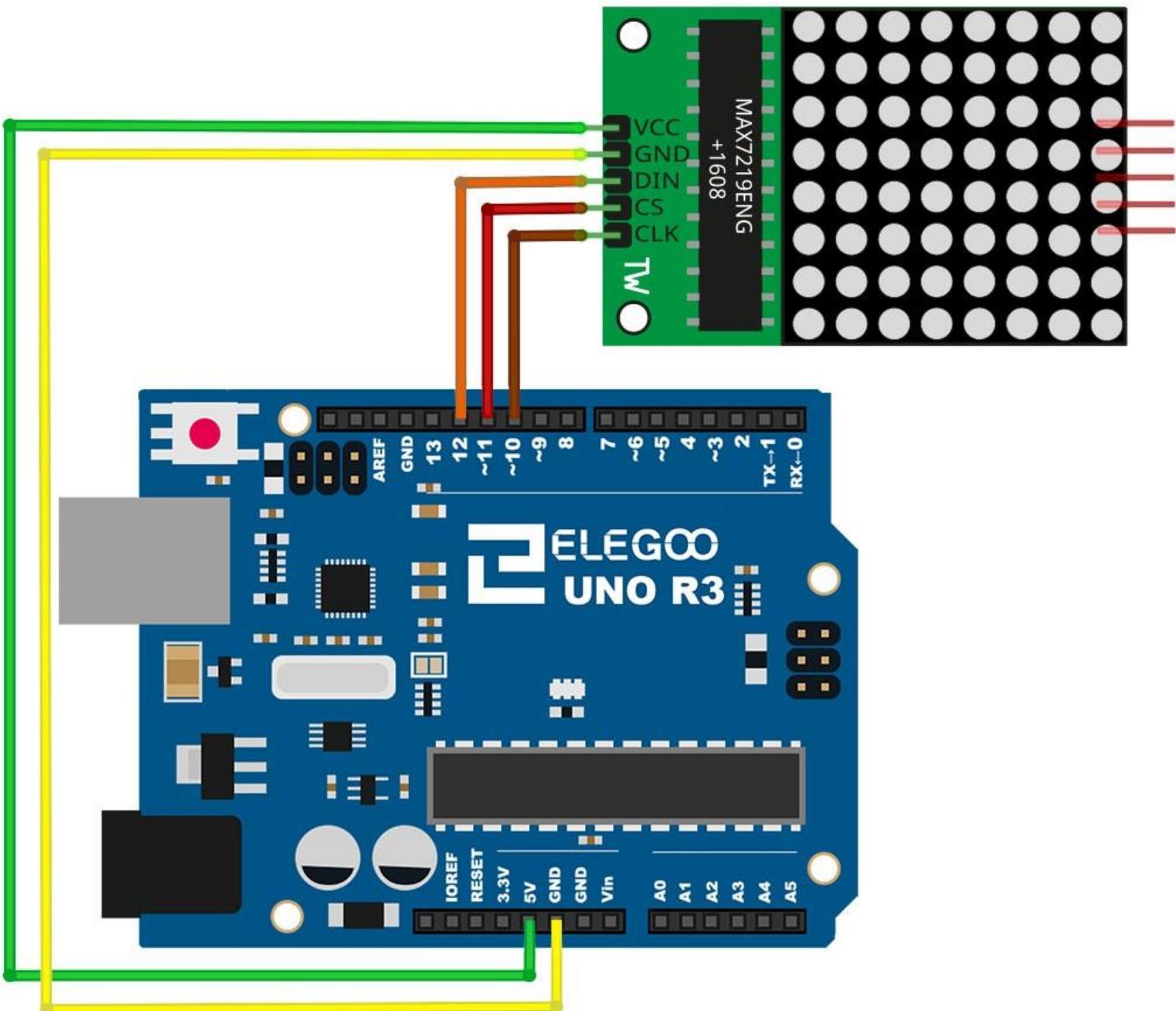


Diagramma di Collegamento



Connetti la messa a terra e la VCC alla scheda UNO

Il pin va connesso a DIN, il Pin 11 va connesso a CS e infine il pin 10 va connesso a CLK.

## Codice

Il nostro codice farà uso della libreria “Maxmatrix” per comunicare con il modulo MAX7219.

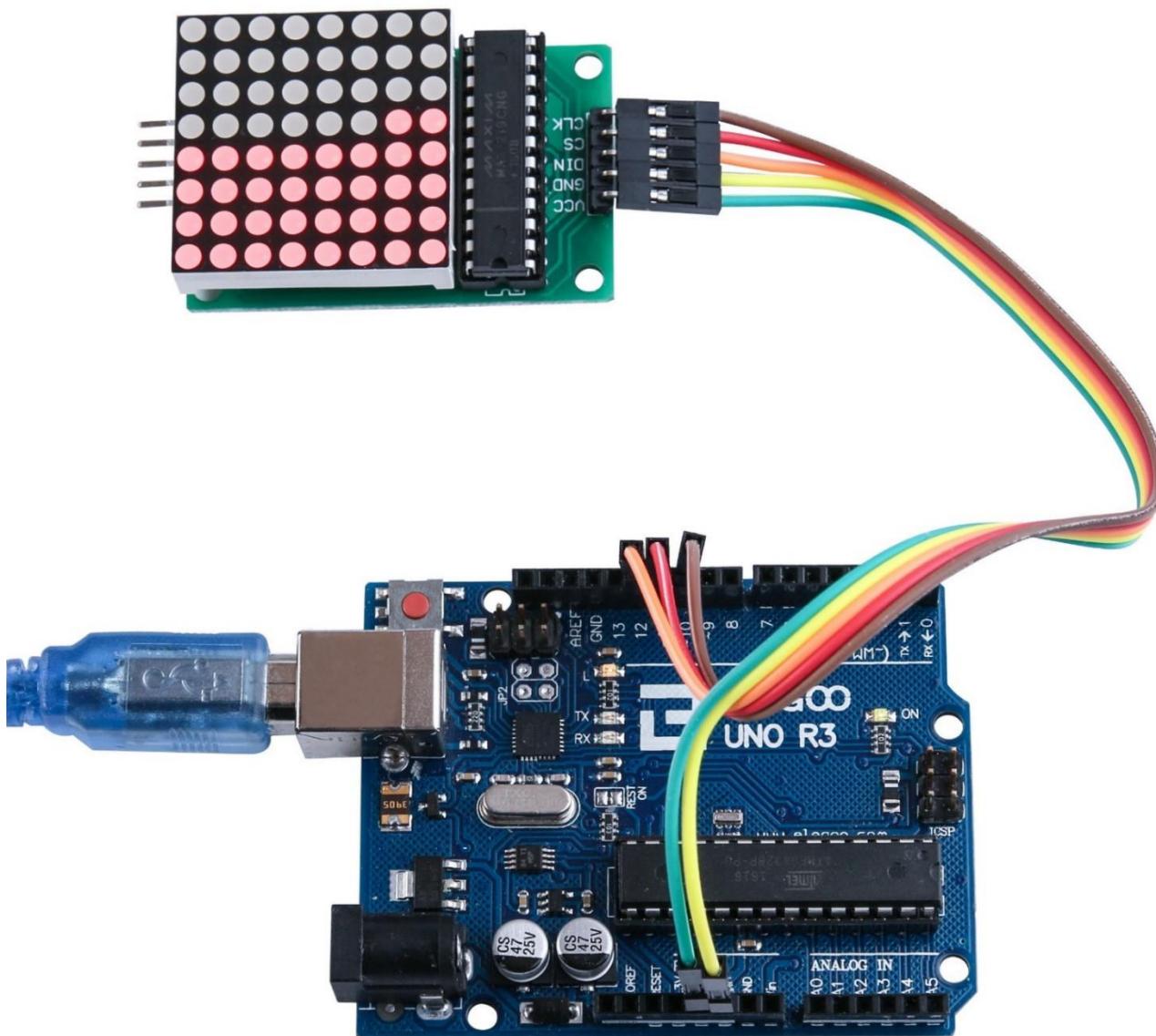
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 15 MAX7219 LED Dot Matrix Module”, clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria < LedControl > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d’esempio



## Lezione 16 Modulo GY-521

### Introduzione

In questa lezione imparerai come utilizzare il modulo GY-521, esso uno de migliori sensori IMU (Inertia Measurement Unit – Sensore di misura dell'inerzia) compatibili con Arduino. I sensori IMU come l'GY-521 sono utilizzati per robot auto-bilanciati (capaci di stare in equilibrio), UAV, smartphone, etc.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo GY-521
- (4) x Connettori F-M

### Introduzione ai componenti

#### Sensore GY-521

Il sensore GY-521 sviluppato dalla società InvenSense contiene accelerometri e giroscopi MEMS (Micro Electro-Mechanical System). Questo sensore è molto preciso e contiene anche un convertitore analogico digitale a 16 bit per ogni canale.

Perciò rileva contemporaneamente l'inerzia sugli assi x, y, e z. Il sensore utilizza la connessione I2C-bus per interfacciarsi con Arduino.

Il sensore GY-521 non è molto costoso, considerato soprattutto che contiene sia accelerometro che giroscopio.



I sensori IMU sono parte dei sensori più utilizzati al giorno d'oggi per tutti i tipo di gadget elettronici. Essi si possono trovare negli smartphones, nei dispositivi indossabili, nei controller delle piattaforme di gioco, etc. I sensori IMU ci aiutano a stabilire la posizione dell'oggetto e stabilire la sua posizione nelle tre dimensioni.

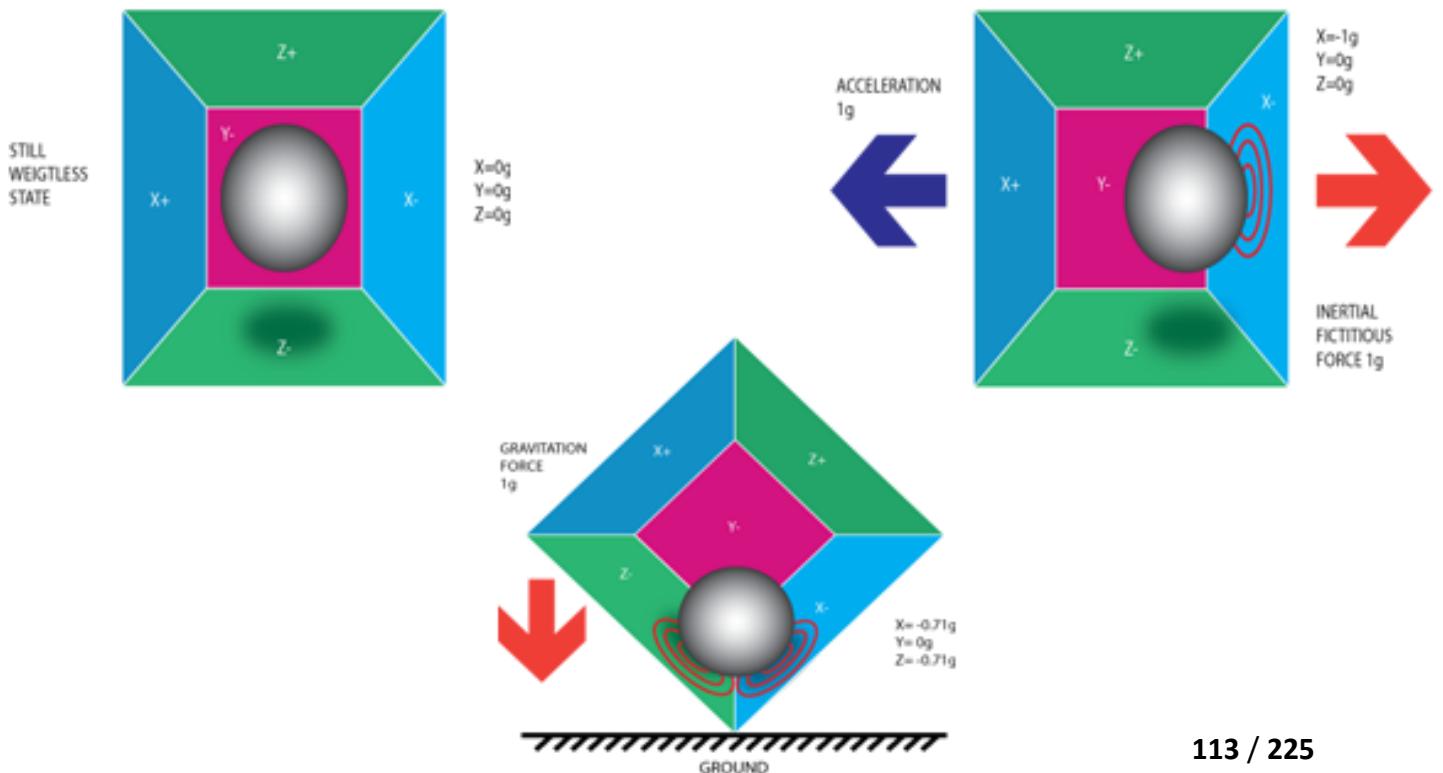
I 3 valori solitamente valutati in gradi ci aiutano a determinare la posizione dell'oggetto. Perciò, sono spesso usati negli smartphone per determinarne l'orientamento. Nei dispositivi indossabili come nike fuel band o fit bit vengono utilizzati questi sensori per tracciare i movimenti.

### Come funziona il sensore?

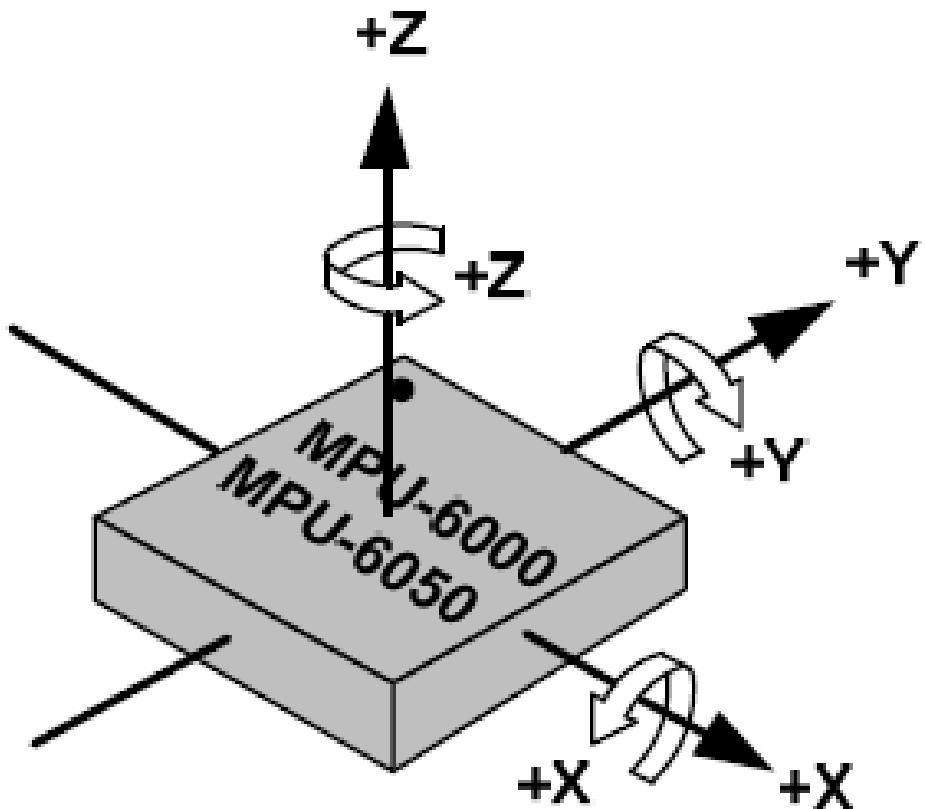
I sensori IMU sono composti da due o più parti, elencandole per priorità esse sono: accelerometro, giroscopio, magnetometro e altimetro. Il sensore GY-521 ha 6 gradi di libertà (DOF - Degree Of Freedom) detto anche sensore a sei assi, ciò significa che sarà in grado di fornire sei valori come output. Tre valori saranno ricavati dall'accelerometro e 3 dal giroscopio. Il sensore MPU 650 è un sensore basato sulla tecnologia MEMS (Micro Electro Mechanical Systems).

Accelerometro e giroscopio, sono entrambi contenuti all'interno di un singolo chip. Questo chip utilizza il protocollo di comunicazione I2C (Inter Integrated Circuit).

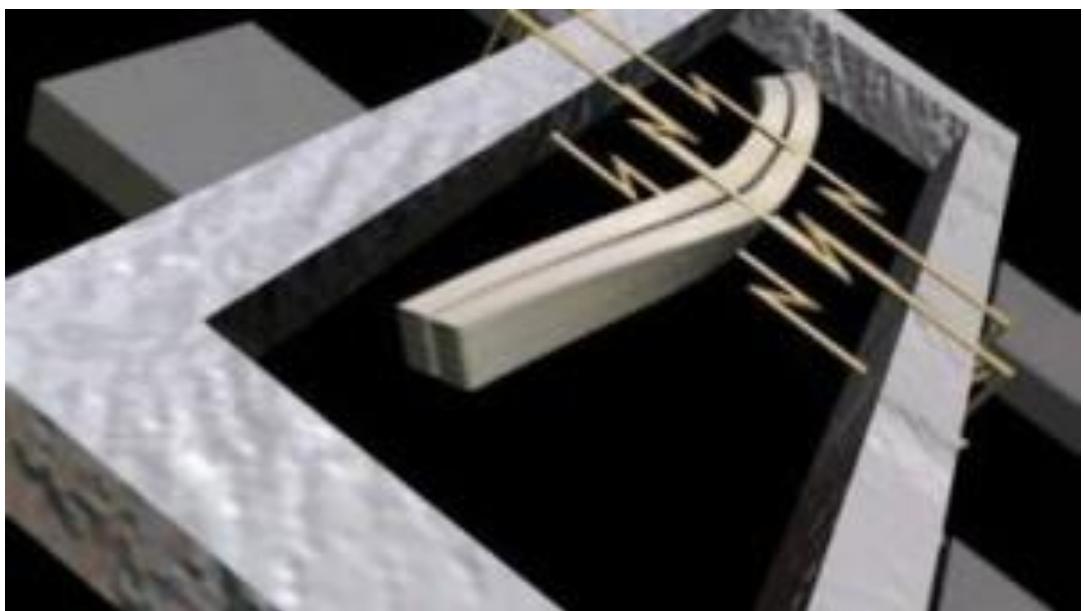
### Come funziona l'accelerometro?



Un accelerometro funziona sfruttando il principio dell'effetto piezoelettrico. Immagina una scatola di forma cubica con una piccola sferetta all'interno con nell'immagine qui sopra. Le pareti di questa scatola sono fatte di materiale cristallino piezoelettrico. Quando ruoti il box la sferetta è forzata a muoversi, ad opera della forza di gravità, nella direzione in cui viene inclinata la scatola. Le pareti contro cui la sferetta collide, creano delle leggerissime scariche di piezoelettricità. La scatola è composta da un totale di 3 paia di pareti opposte. Ogni paio corrisponde ad un asse nello spazio tridimensionale: secondo le assi X, Y, Z. In base alla corrente piezoelettrica prodotta dalle pareti cristalline si può determinare la direzione dell'inclinazione.



### Come funziona il Giroscopio?



Il giroscopio funziona sfruttando il principio di accelerazione di Coriolis. Immagine che ci sia una struttura simile ad una forca che ha un movimento costante, avanti ed indietro. Questa struttura è tenuta in posizione da un cristallo piezoelettrico. Ogni qualvolta la struttura viene mossa il cristallo subisce una forza nella direzione dell'inclinazione, questa forza è causata dal risultato della forza di inerzia applicata alla forca. Il cristallo in questo modo produce una corrente dovuta all'effetto piezoelettrico e tale corrente viene amplificata. Questi valori sono raffinati poi dal microcontrollore.

## Connessione

### Schema

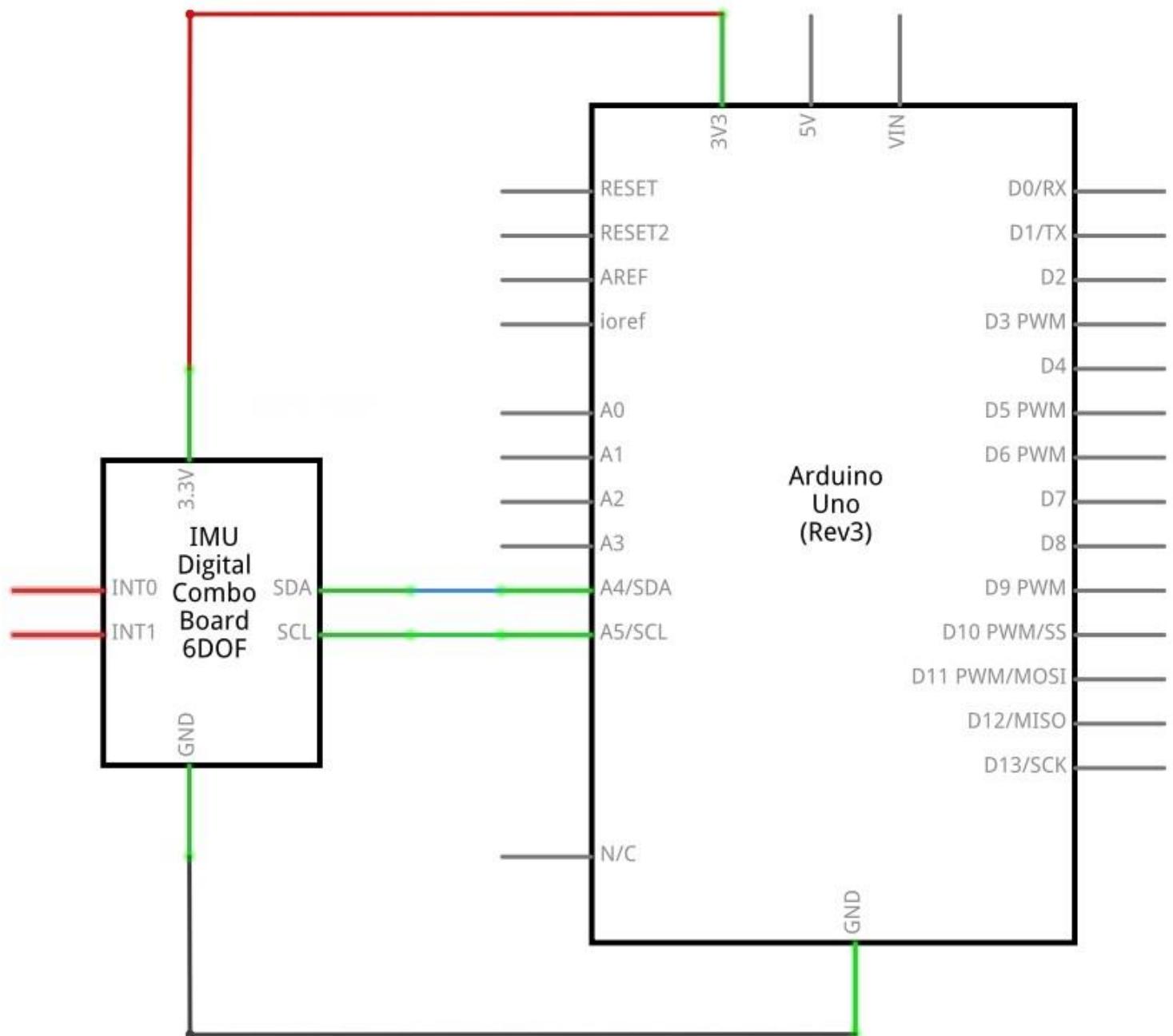
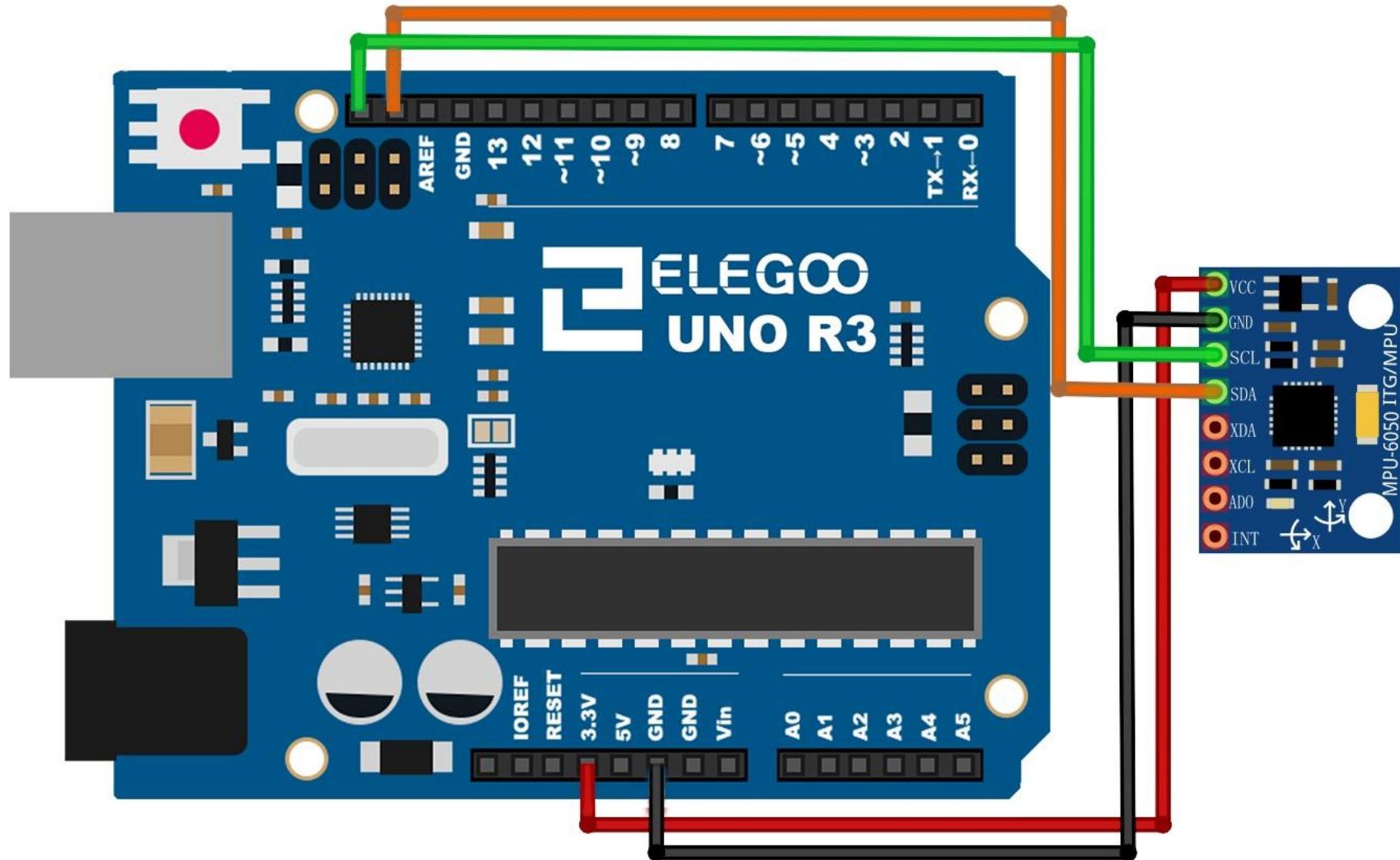


Diagramma di Collegamento



Ora dovrà settare le connessioni I2C, per queste connessioni collega il pin SDA sul sensore GY-521 al pin analogico di Arduino numero 4. Il pin etichettato con SCL sul sensore GY-521 dovrà connetterlo al pin analogico 5 su Arduino. Avrai terminato così le connessioni con il sensore MPU 5060.

## **Libreria necessaria**

MPU-6050

## **Codice**

Il breve esempio ci mostra tutti i valori grezzi (accelerometro, giroscopio, temperatura). Il codice funziona con Arduino UNO, Nano, Leonardo e Due.

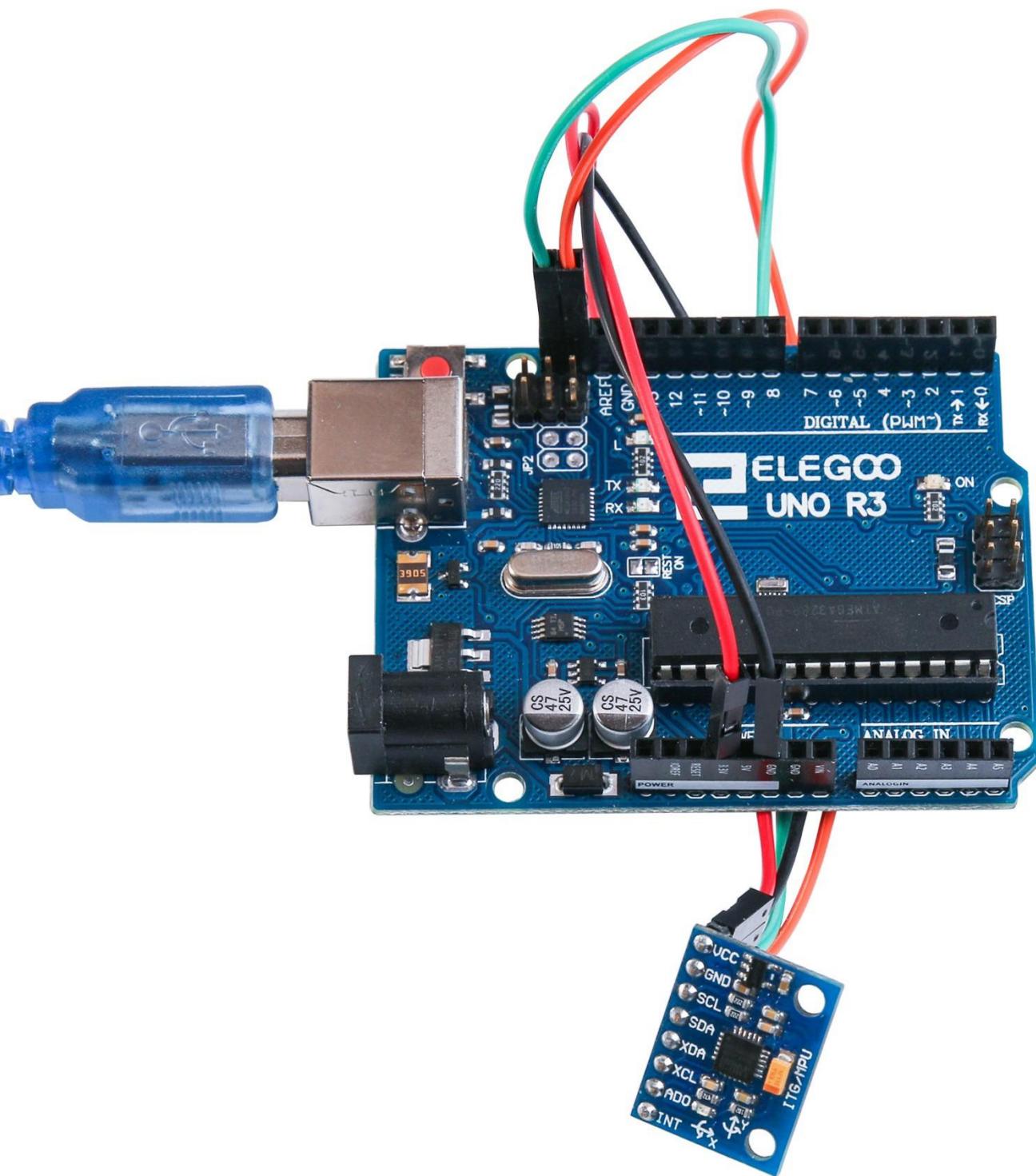
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 16 GY-521 Module”, clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria < GY-521 > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto d'esempio



Apri il monitor serial monitor e potrai vedere dati simili a questi:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.

The screenshot shows the Arduino Serial Monitor window titled "COM21 (Arduino/Genuino Uno)". The window displays a continuous stream of sensor data. Each line of data consists of six variables separated by vertical bars: AcX, AcY, AcZ, Imp, GyX, GyY, and GyZ. The values for each variable range from approximately -5920 to 15976. The "Send" button is located in the top right corner of the window. At the bottom, there are three buttons: "Autoscroll" (checked), "Newline", and a baud rate selection dropdown set to "9600 baud".

```
AcX = 15976 | AcY = -4280 | AcZ = -596 | Imp = 24.62 | GyX = 230 | GyY = -26 | GyZ = -1231
AcX = 15940 | AcY = -4408 | AcZ = -648 | Imp = 24.53 | GyX = -357 | GyY = -590 | GyZ = -519
AcX = 15852 | AcY = -4328 | AcZ = -668 | Imp = 24.62 | GyX = -189 | GyY = -79 | GyZ = -198
AcX = 15844 | AcY = -3972 | AcZ = -708 | Imp = 24.62 | GyX = -254 | GyY = -216 | GyZ = -44
AcX = 15740 | AcY = -4232 | AcZ = -968 | Imp = 24.53 | GyX = -319 | GyY = -141 | GyZ = -227
AcX = 15900 | AcY = -4936 | AcZ = -1008 | Imp = 24.62 | GyX = 126 | GyY = 111 | GyZ = 3870
AcX = 15356 | AcY = -5080 | AcZ = -1192 | Imp = 24.58 | GyX = -1670 | GyY = -1741 | GyZ = -2571
AcX = 14592 | AcY = -6504 | AcZ = -5700 | Imp = 24.53 | GyX = 662 | GyY = 264 | GyZ = 3219
AcX = 13740 | AcY = -7020 | AcZ = -2744 | Imp = 24.58 | GyX = 8265 | GyY = 4962 | GyZ = 8163
AcX = 3600 | AcY = -16556 | AcZ = 4244 | Imp = 24.58 | GyX = -17048 | GyY = -12197 | GyZ = 3845
AcX = 12248 | AcY = -12292 | AcZ = 7256 | Imp = 24.62 | GyX = 12046 | GyY = 24428 | GyZ = -5483
AcX = 588 | AcY = -3832 | AcZ = 19208 | Imp = 24.53 | GyX = 9258 | GyY = -4420 | GyZ = -4557
AcX = 1896 | AcY = -3784 | AcZ = 6320 | Imp = 24.62 | GyX = -7486 | GyY = -32768 | GyZ = 2677
AcX = 32767 | AcY = -19068 | AcZ = -1920 | Imp = 24.58 | GyX = -9262 | GyY = -19403 | GyZ = 25320
AcX = -19160 | AcY = 12004 | AcZ = -2452 | Imp = 24.58 | GyX = -32768 | GyY = -32768 | GyZ = -4809
AcX = -25124 | AcY = 1616 | AcZ = 32767 | Imp = 24.62 | GyX = 7628 | GyY = 7064 | GyZ = 6299
AcX = 11976 | AcY = -8432 | AcZ = -32600 | Imp = 24.53 | GyX = 29381 | GyY = 32767 | GyZ = -19841
AcX = 972 | AcY = -22992 | AcZ = -12480 | Imp = 24.62 | GyX = -31051 | GyY = -32768 | GyZ = 32767
AcX = -27260 | AcY = 16868 | AcZ = 10704 | Imp = 24.62 | GyX = 32767 | GyY = 28603 | GyZ = -20636
AcX = 32268 | AcY = -32468 | AcZ = -21952 | Imp = 24.58 | GyX = -27684 | GyY = -32768 | GyZ = 32767
AcX = -22476 | AcY = -8436 | AcZ = -3976 | Imp = 24.58 | GyX = 32156 | GyY = 32767 | GyZ = 25696
AcX = -3836 | AcY = -13428 | AcZ = -9628 | Imp = 24.58 | GyX = -30925 | GyY = -32768 | GyZ = 32767
AcX = 3164 | AcY = -5392 | AcZ = -19464 | Imp = 24.48 | GyX = -30769 | GyY = -17986 | GyZ = 17236
AcX = 3408 | AcY = -3584 | AcZ = -13752 | Imp = 24.58 | GyX = 1820 | GyY = -2660 | GyZ = -186
AcX = 4404 | AcY = -5552 | AcZ = -15216 | Imp = 24.58 | GyX = -578 | GyY = -234 | GyZ = -425
AcX = 4160 | AcY = -5456 | AcZ = -15304 | Imp = 24.53 | GyX = -445 | GyY = -154 | GyZ = -277
AcX = 4152 | AcY = -5192 | AcZ = -15300 | Imp = 24.53 | GyX = -404 | GyY = -114 | GyZ = -262
```

## Lezione 17 Sensore di Movimento HC-SR501

### Introduzione

In questa lezione imparerai come utilizzare un sensore di movimento PIR con un la tua scheda UNO. La scheda UNO sarà il cuore di questo progetto, essa “ascolta” il sensore PIR ed in base al rilevamento o meno di movimento si accende o spegne il led sulla scheda UNO

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Sensore di movimento HC-SR501
- (3) x Connettori F-M ( Connettori di DuPont Femmina – Femmina )

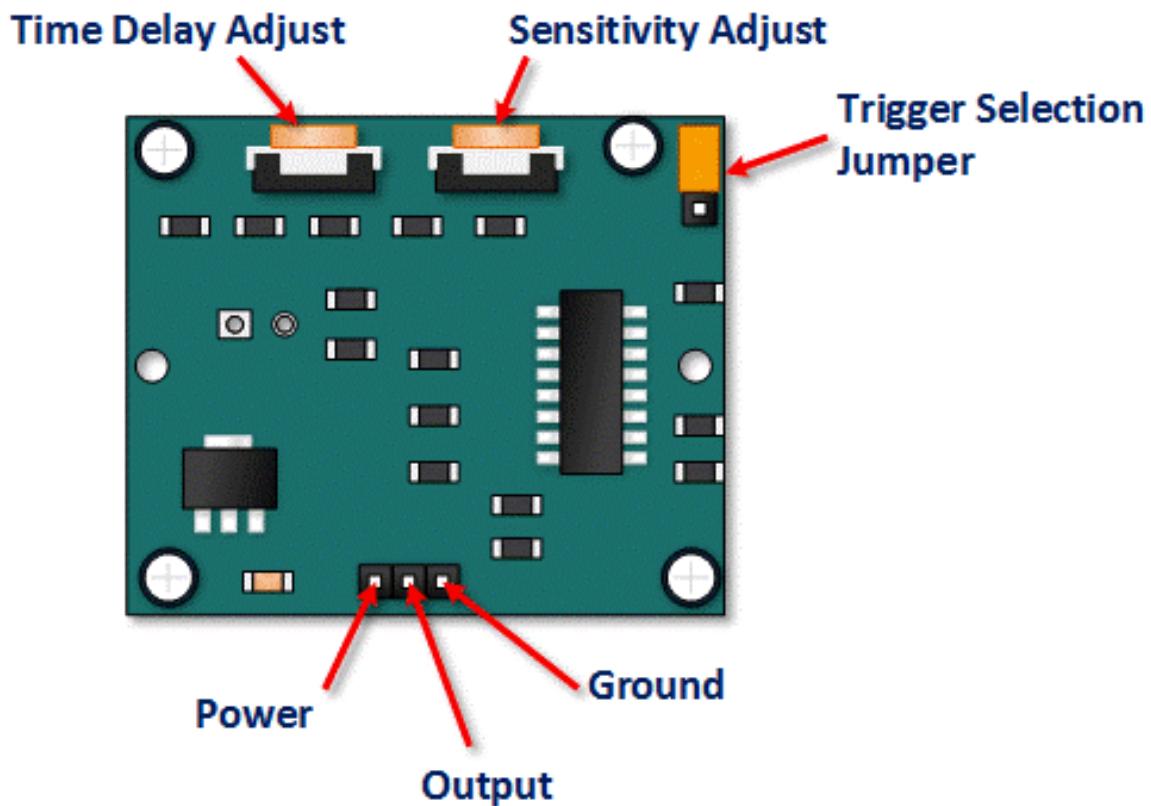
### Introduzione ai componenti

#### SENSORE PIR:

Il sensore PIR è molto più complicati di altri sensori spiegati in questo tutorial (per esempio fotoresistenze, FSR e interruttore ad inclinazione) perché ci sono molte variabili che hanno effetto sugli input ed output del sensore.

Il sensore PIR stesso ha due slot. Ognuno di essi è fatto di un materiale speciale che è sensibile agli Infrarossi. Quando il sensore è inattivo, entrambi gli slot rilevano la stessa quantità di infrarossi radiati dalla stanza, ambiente, parti, etc. Quando un corpo caldo come un umano o un animale, passa davanti al sensore, viene intercettato dalla prima metà del sensore il quale causa un cambio di differenziale tra le due metà. Quando un corpo caldo lascia l'area avviene l'opposto, e viene generato un cambio di differenziale negativo. Queste pulsazioni vengono rilevate e trasmesse.





Pin o Controllo	Funzione
Aggiustamento ritardo	Imposta per quanto tempo l'output rimane in posizione di HIGH dopo aver rilevato il movimento. Il range varia da 5 secondi a 5 minuti.
Aggiustamento di sensibilità	Imposta il range di rilevamento, che può variare tra i 3 e i 7 metri.
Jumper di selezione	Imposta il rilevamento singolo o a ripetizione.
Pin Messa a Terra	Input di messa a terra
Pin di Output	LOW quando non viene rilevato del movimento, HIGH quando viene rilevato del movimento. Il valore di High è 3.3V.
Power Pin	Input di tensione con un range tra i 5 ed i 20 V di corrente continua.

## HC SR501 PIR Descrizione del Funzionamento

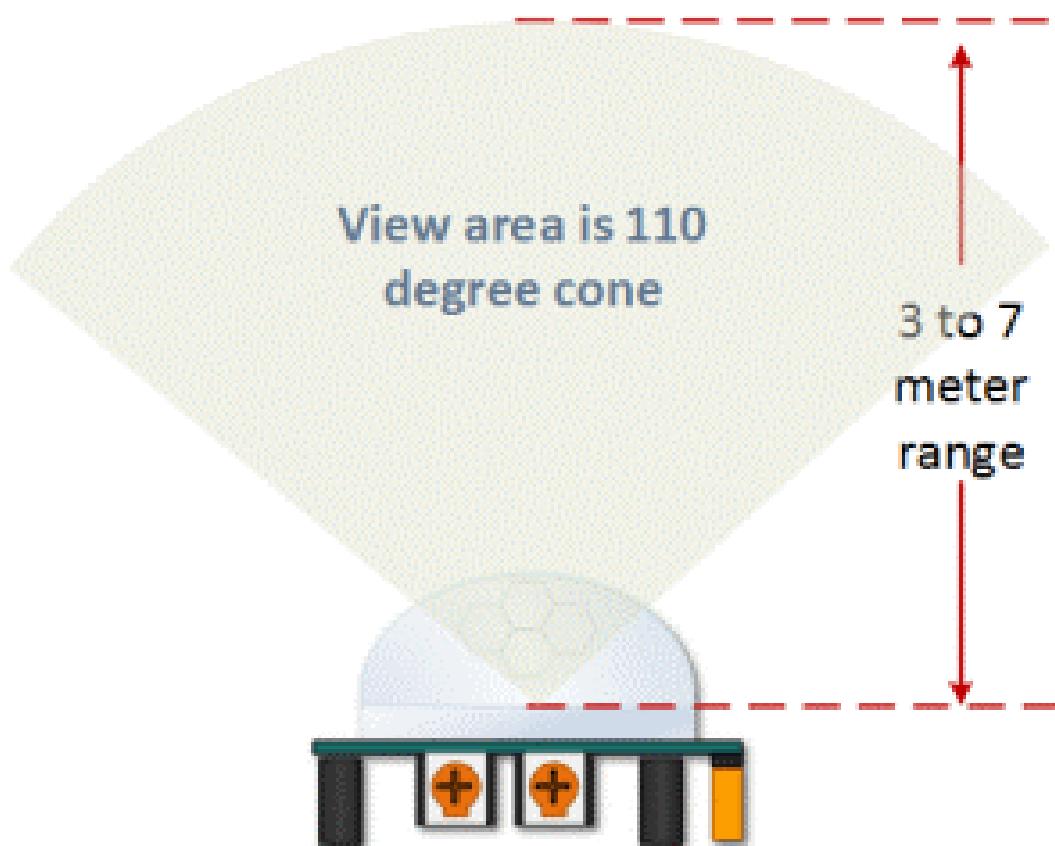
Il sensore SR501 è in grado di rilevare cambiamenti della luce ad infrarossi e se interpretati come movimento, portano l'output ad HIGH. Cosa viene interpretato come movimento o meno dipende largamente da dalle impostazioni e gli aggiustamenti impostati sul sensore.

### Inizializzazione del dispositivo

Il dispositivo necessita circa un minuto per essere pronto all'uso, durante questo periodo potrebbero presentarsi dei segnali di rilevamento errati. Il circuito e la logica del controllore necessitano di questo breve periodo di tempo per iniziare a lavorare correttamente.

### Area di rilevamento del dispositivo

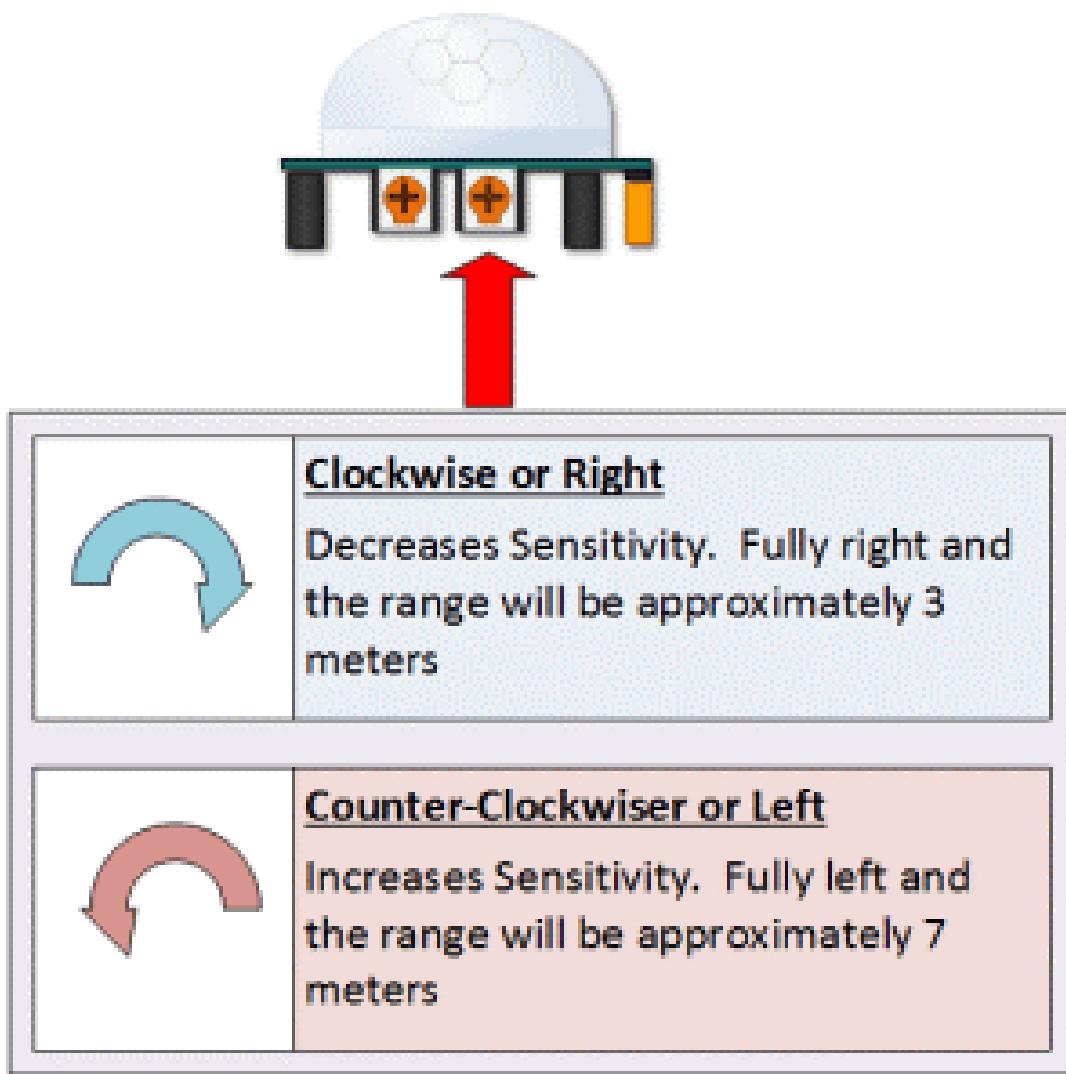
Il dispositivo rileva il movimento all'interno di un angolo di 110 gradi in un range compreso tra i 3 e i 7 metri.



## HC SR501 Raggio di visione

Aggiustare la sensibilità del range del sensore PIR.

Come detto prima, il range di distanza tra cui è possibile rilevare i movimenti è circa tra i 3 e i 7 metri. L'illustrazione qui sotto mostra come impostare la sensibilità.

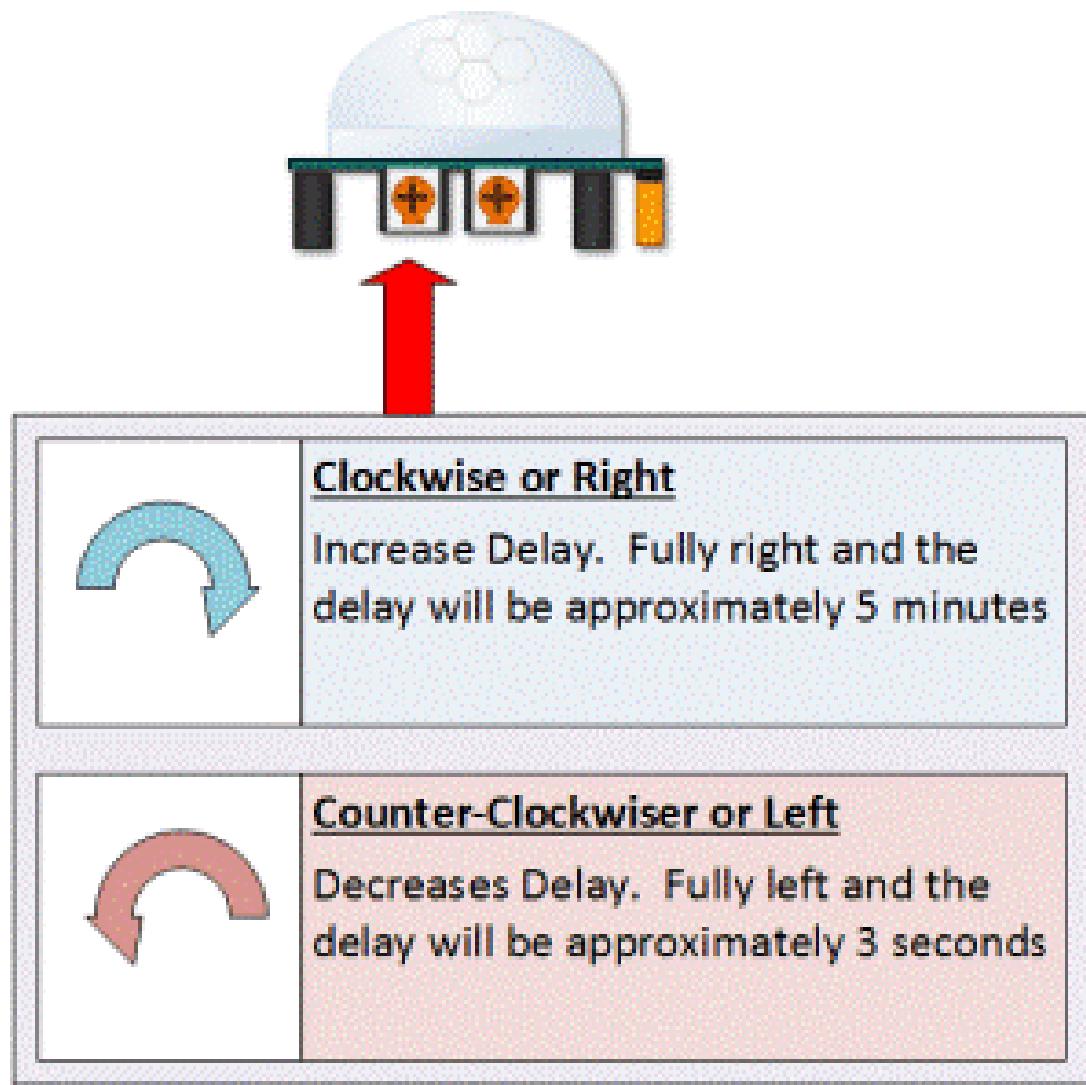


## HC SR501 Correzione dell'intervallo di tempo

HC SR501 Aggiustare la sensibilità dell'intervallo di tempo

L'intervallo di tempo determina quanto dura l'intervallo di tempo in cui il modulo de sensore PIR rimane in posizione HIGH dopo aver rilevato il movimento. Il range varia tra i 3 secondi ed i 5 minuti.

**IMPORTANTE** – Dopo che sarà terminato l’intervallo di tempo che abbiamo impostato per il rilevamento di movimento, l’output del sensore andrà a LOW (OFF) per circa 3 secondi. Tutti i movimenti non saranno rilevati durante questo periodo di 3 secondi



### ESEMPIO:

Immagina tu sia in modalità innesco singolo ( single trigger ) ed il tuo intervallo di tempo sia 5 secondi.

Il sensore PIR rileverà il movimento e terrà l’output HIGH per 5 secondi.

Dopo 5 secondi il sensore PIR porterà l’output a LOW per circa 3 secondi.

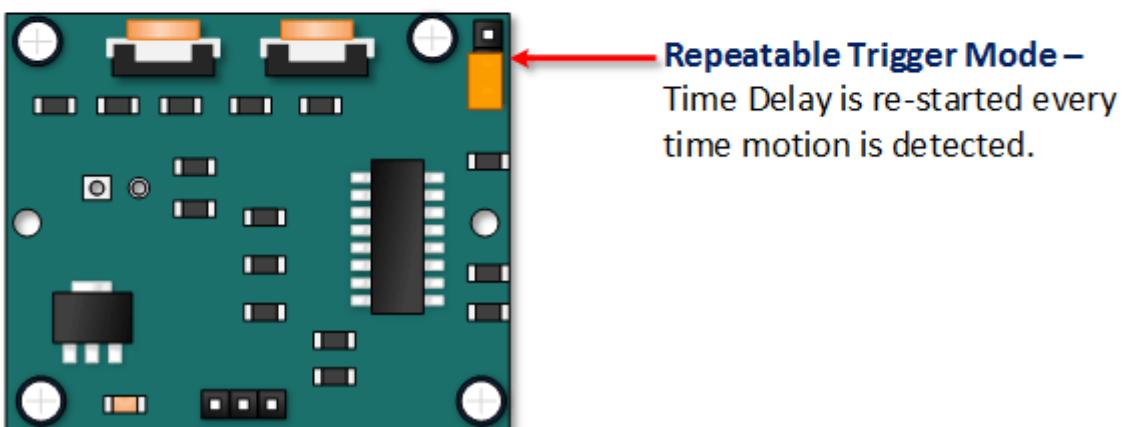
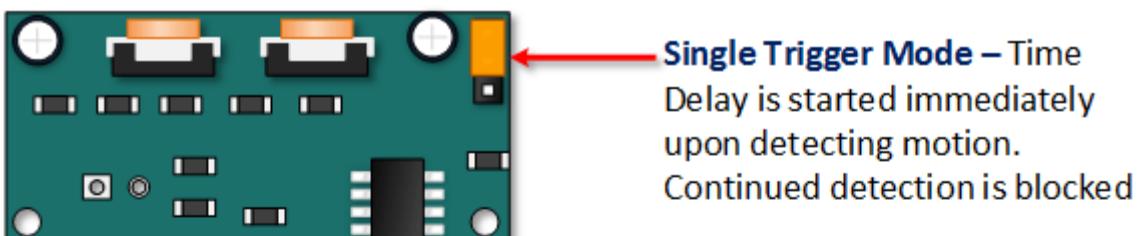
Durante questi 3 secondi il sensore non rileverà alcun movimento.

Terminati i 3 secondi il sensore PIR tornerà attivo e sarà pronto a rilevare di nuovo i movimenti ed una volta rilevati tornerà a portare l’uscita ad HIGH e ricomincerà il ciclo qui descritto.

## Jumper di selezione della modalità di innesco

Il jumper per la selezione della modalità di innesco ti permette di scegliere tra singolo innesco o innesco ripetibile. L'effetto della selezione di questo jumper determina quando inizia l'intervallo di tempo.

- **INNESCO SINGOLO ( SINGLE TRIGGER )** – L'intervallo di tempo inizia immediatamente subito dopo che il primo movimento viene rilevato.
- **INNESCO RIPETIBILE (REPEATABLE TRIGGER)** – Ogni rilevazione di movimento resetta l'intervallo di tempo. Perciò l'intervallo di tempo inizia quando l'ultimo movimento viene rilevato.

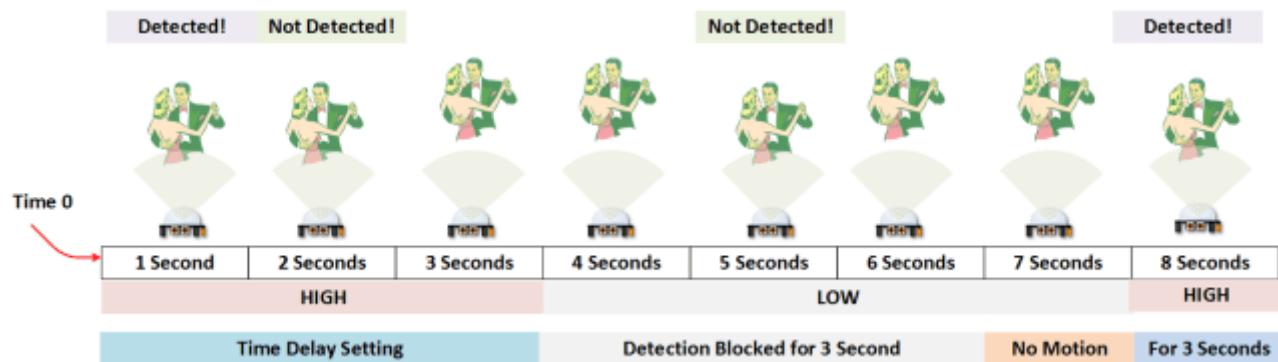


## HC-SR501 Esempio di applicazione per la pista da ballo

Immagine che tu voglia controllare le luci di una pista da ballo in base a dove sono posizionati i ballerini. Puoi benissimo immaginare come l'intervallo di tempo e la modalità di innesco siano necessarie a controllare le luci nella maniera che vuoi.

## Esempio uno

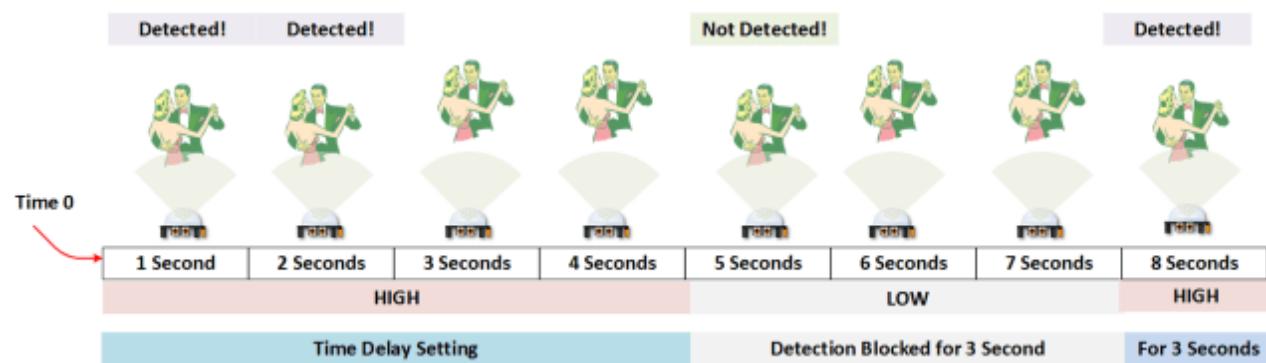
Nel primo esempio l'intervallo di tempo è settato a 3 secondi e la modalità di innesco impostata a singola. Come puoi vedere nell'illustrazione qui sotto, il movimento non è sempre rilevato. Infatti, c'è un periodo di sei secondi in cui c'è del movimento e non viene rilevato.



## Esempio due

In questo secondo esempio l'intervallo di tempo è ancora impostato a 3 secondi, ma la modalità di innesco in questo caso è a ripetizione. Nell'illustrazione qui sotto puoi vedere che l'intervallo di tempo riparte. Tuttavia, dopo che tre secondi, il rilevamento sarà bloccato per altri tre secondi.

Come ho già detto in precedenza, è possibile ignorare il periodo di stop di 3 secondi con del codice creativo, ma non prendere in considerazione questa opzione. Alcuni dei componenti elettronici usati potrebbero surriscaldarsi senza questo intervallo di tempo. I tre secondi lasciano al componente un po' di riposo prima di iniziare di nuovo a rilevare.



## Connessione

### Schema

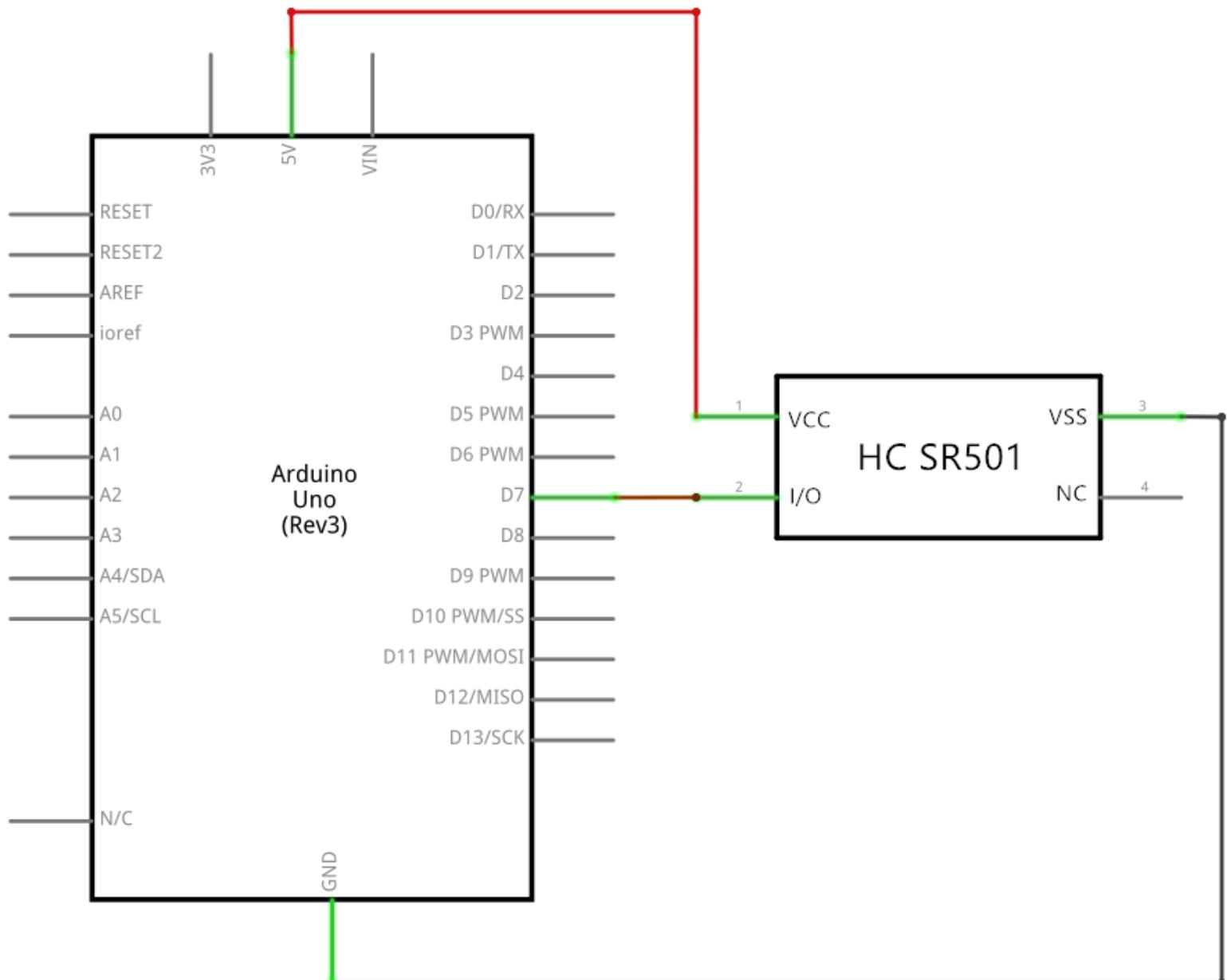
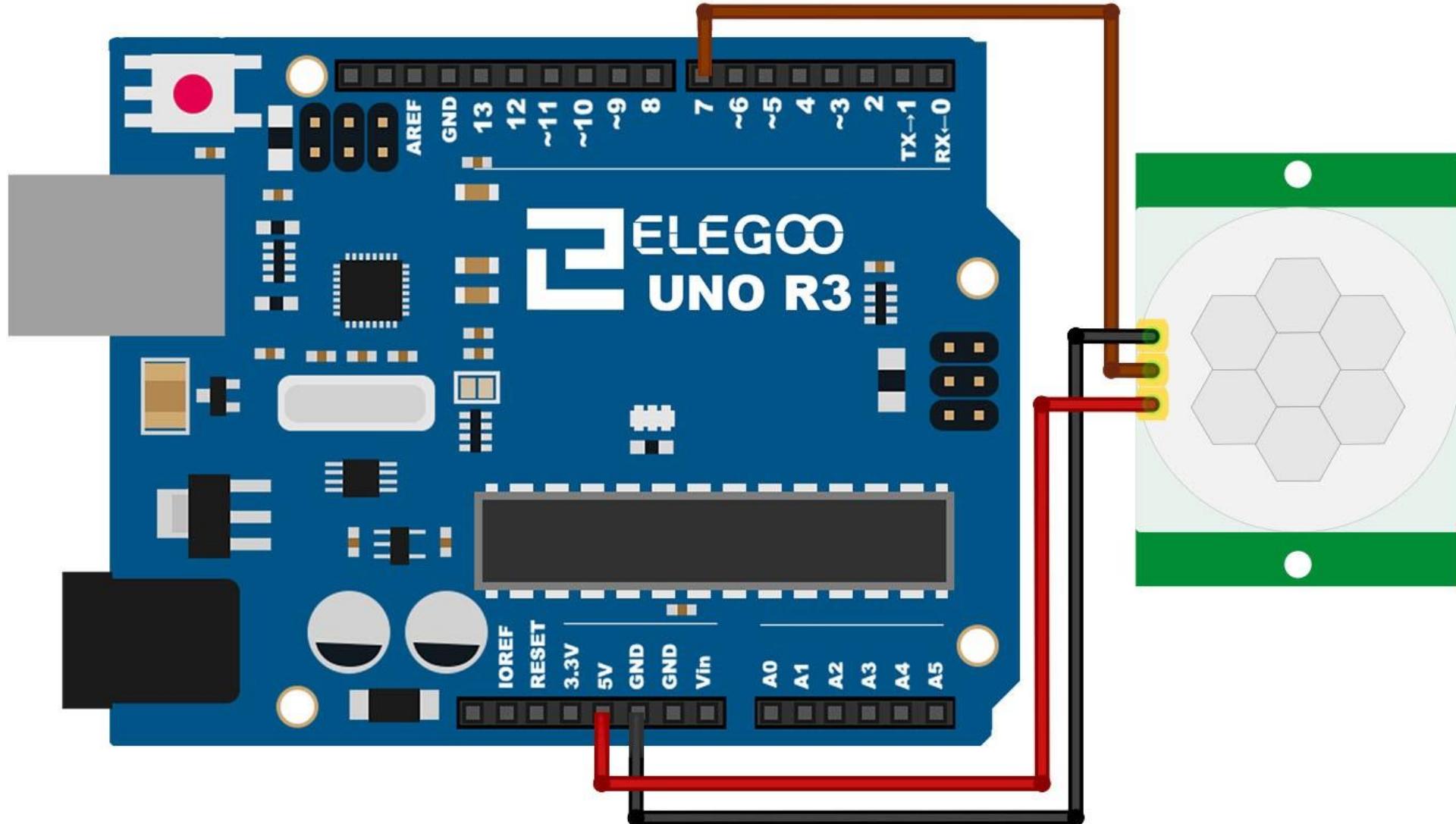


Diagramma di collegamento



Collegare il sensore PIR ad un microcontrollore è molto semplice. Il sensore PIR agisce come output digitale, perciò tu devi semplicemente mettere il microcontrollore in ascolto sul pin di output del sensore per rilevare lo stato di HIGH (rilevato movimento) e LOW (nessun rilevamento).

Alimenta il sensore pin con 5V e connetti la messa a terra con la messa a terra. Dopodiché connetti l'output ad un pin digitale, in questo esempio utilizzeremo il pin 7.

## Codice

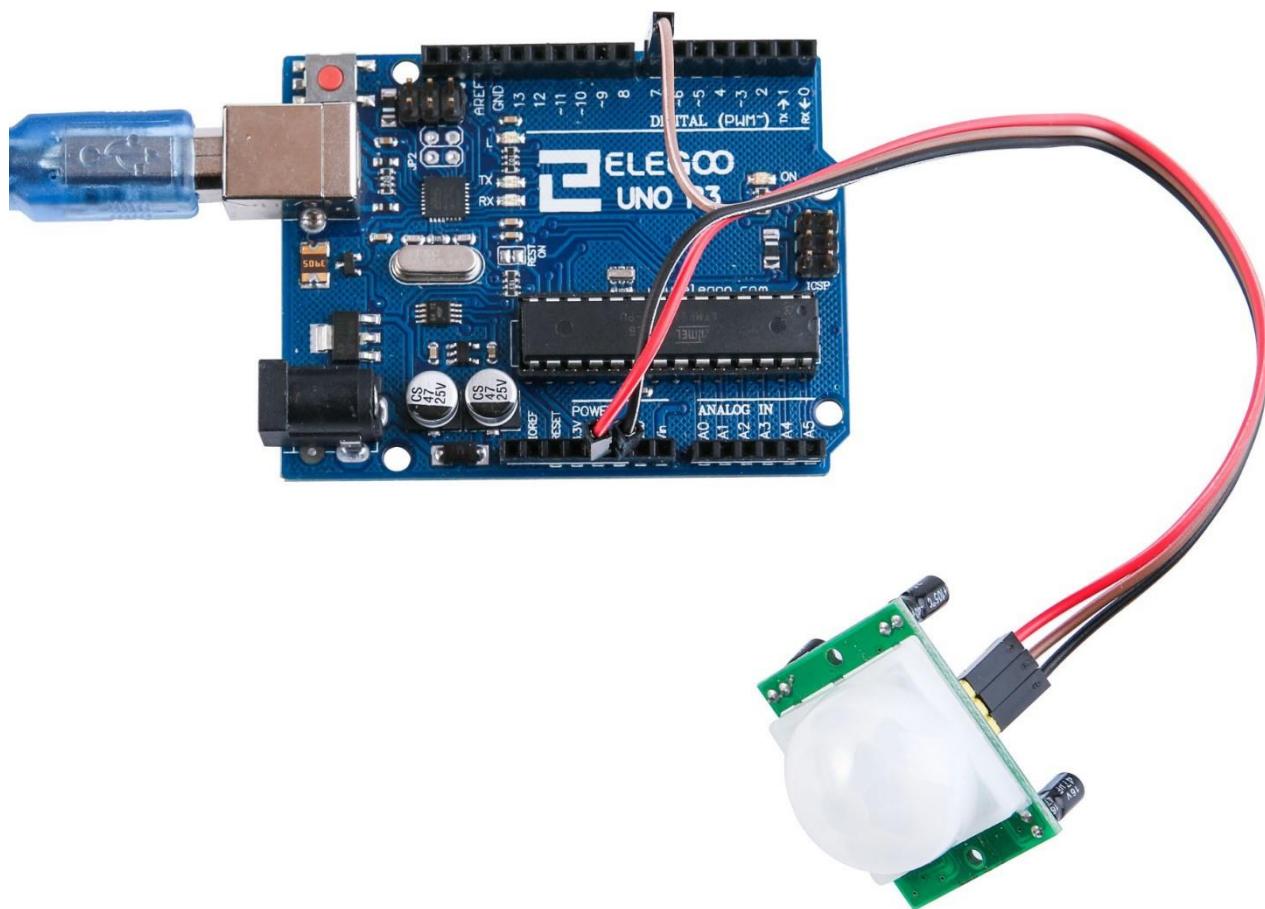
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 17 HC-SR501 PIR Sensor”, clicca su UPLOAD per caricare il programma.

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

Questo codice è molto semplice, permette di accedere il LED connesso al pin 13 ogniqualvolta viene rilevato del movimento.

Ricordati di includere l’attesa di un minuto per l’inizializzazione del sensore quando sviluppi il codice.

## Foto d’esempio



# Lezione 18 Modulo Sensore del livello dell'Acqua

## Introduzione

In questa lezione imparerai come utilizzare il modulo sensore del livello dell'Acqua. Questo modulo è in grado di percepire la profondità dell'acqua, il nucleo di questo componente è un circuito amplificatore costituito da un transistor e più linee di circuito stampato. Quando inserito nell'acqua queste linee di circuito stampato formano una resistenza che varia in base alla profondità dell'acqua. Poi, il segnale della profondità dell'acqua viene convertito in un segnale elettrico che permette di conoscere il cambiamento di profondità rilevata attraverso una funzione di conversione analogico digitale del UNO R3.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (3) x Connettore F-M ( Connettore di DuPont Femmina – Maschio )
- (1) x Modulo sensore del livello dell'acqua

## Introduzione ai componenti



### Sensore del livello dell'acqua:

Questo sensore è progettato per rilevare la profondità dell'acqua, viene largamente utilizzato per rilevare la pioggia, le perdite di acqua e il livello dell'acqua. Il modulo si può suddividere in tre parti, un connettore, una resistenza da  $1\text{ M}\Omega$ , e numerose linee di materiale leggermente conduttivo.

Questo sensore lavora con una serie di linee esposte e connesse alla messa a terra.

Le linee presenti tra le linee della messa a terra sono le linee di percezione.

Le linee di percezione hanno una resistenza di pull-up da  $1\text{ M}\Omega$ . La resistenza tiene il valore del sensore ad un valore alto fino a che una goccia d'acqua mette in cortocircuito la messa a terra con l'altra linea. Difficile da credere, ma questo circuito funzionerà con i pin di Input e output digitale della scheda UNO. In alternativa puoi utilizzare un pin analogico per rilevare la quantità di acqua presente tra le tracce del sensore e le tracce della messa a terra.

Questo oggetto valuta il livello dell'acqua attraverso una serie di fili paralleli esposti, il sensore è in grado di rilevare gocce d'acqua e profondità. Esso può cambiare la

misura dell'acqua in valore analogico per essere utilizzata direttamente in una funzione del programma per realizzare una funzione si allerti in base al livello dell'acqua.

Il sensore ha un basso consumo ed un'alta sensibilità.

Specifiche:

- 1、 Voltage di funzionamento: 5V
- 2、 Corrente di funzionamento: <20ma 3、  
Interfaccia: Analogica
- 4、 Larghezza di rilevamento: 40mm×16mm 5、  
temperatura di lavoro: 10°C~30°C 6、 Segnale di  
output (volt): 0~4.2V

## Connessione

### Schema

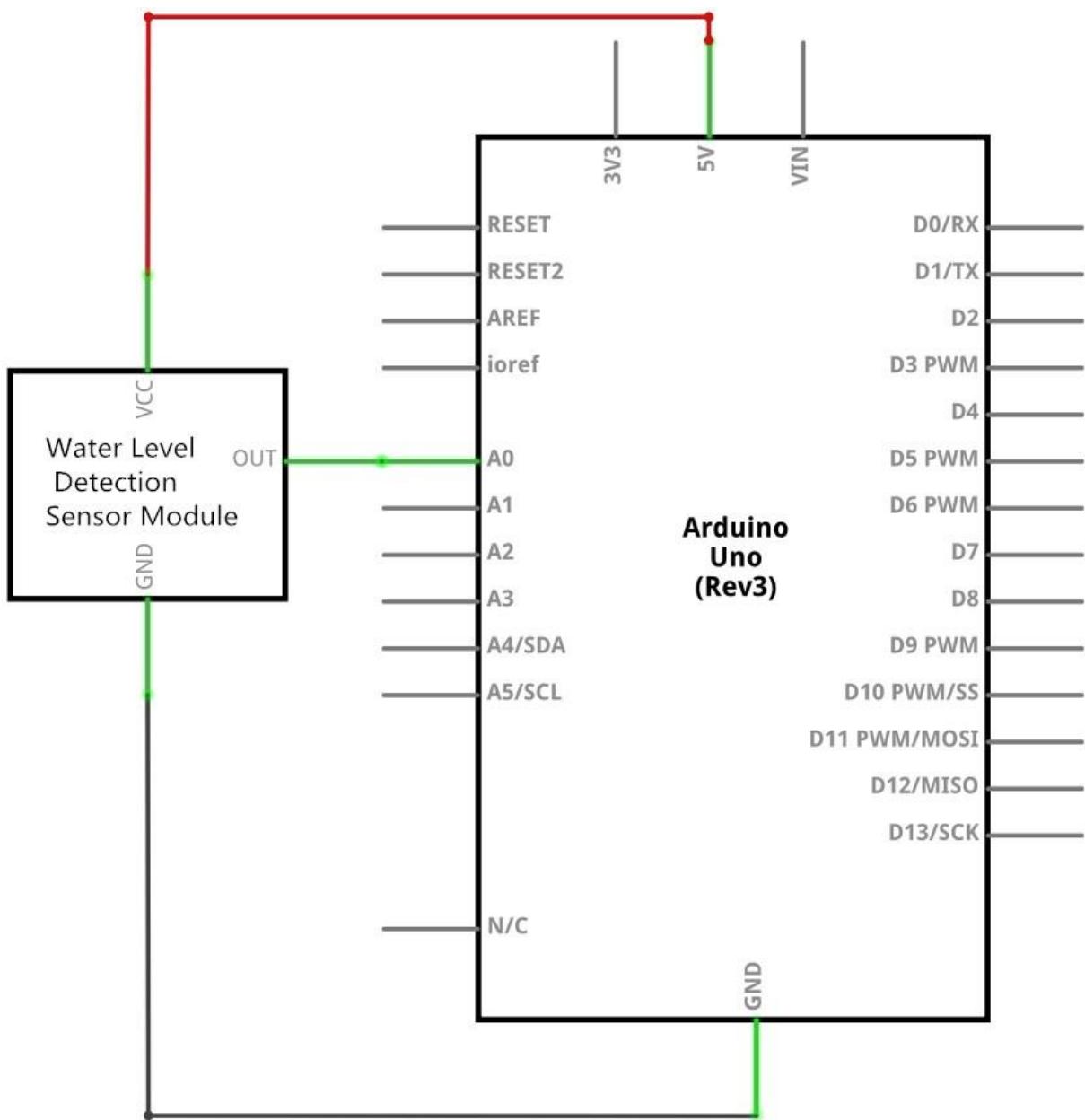
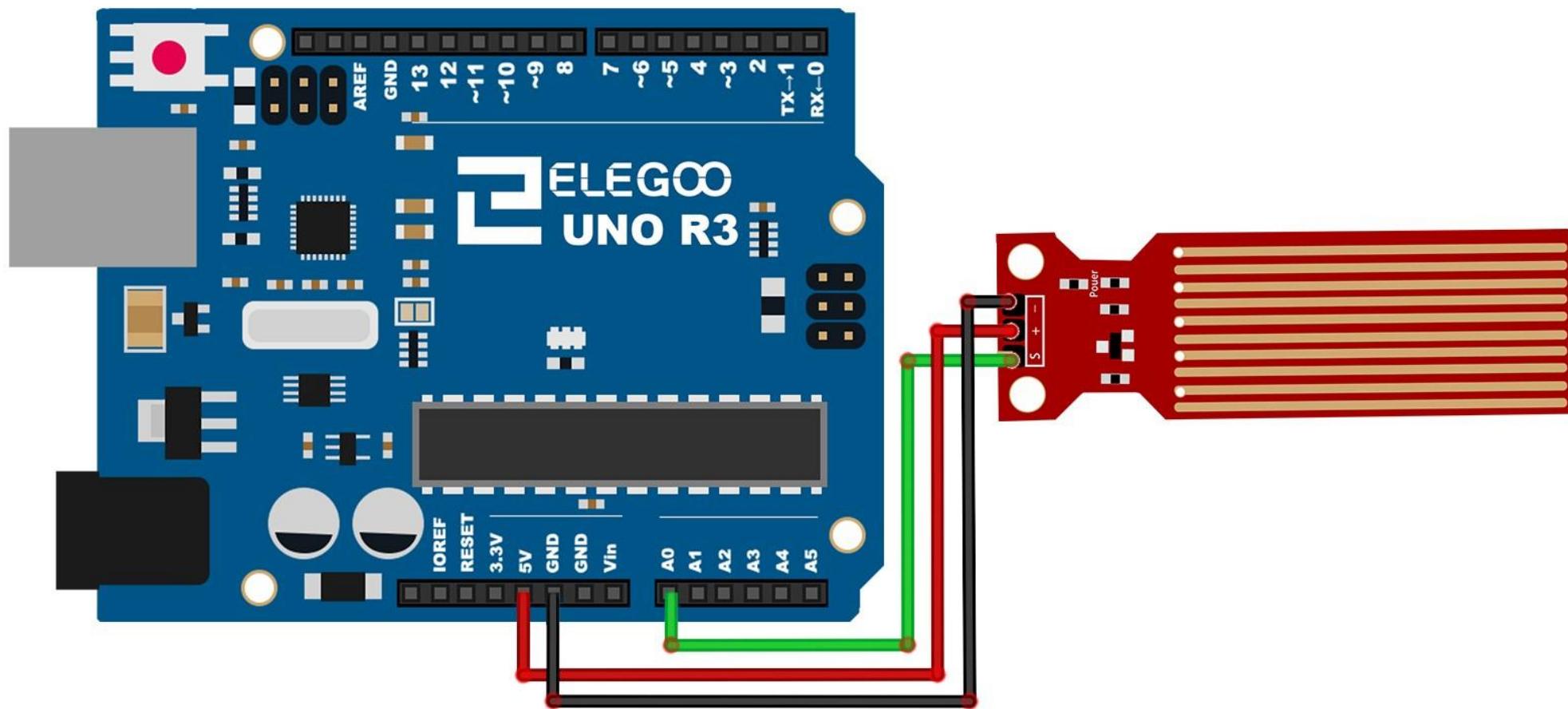


Diagramma di Collegamento

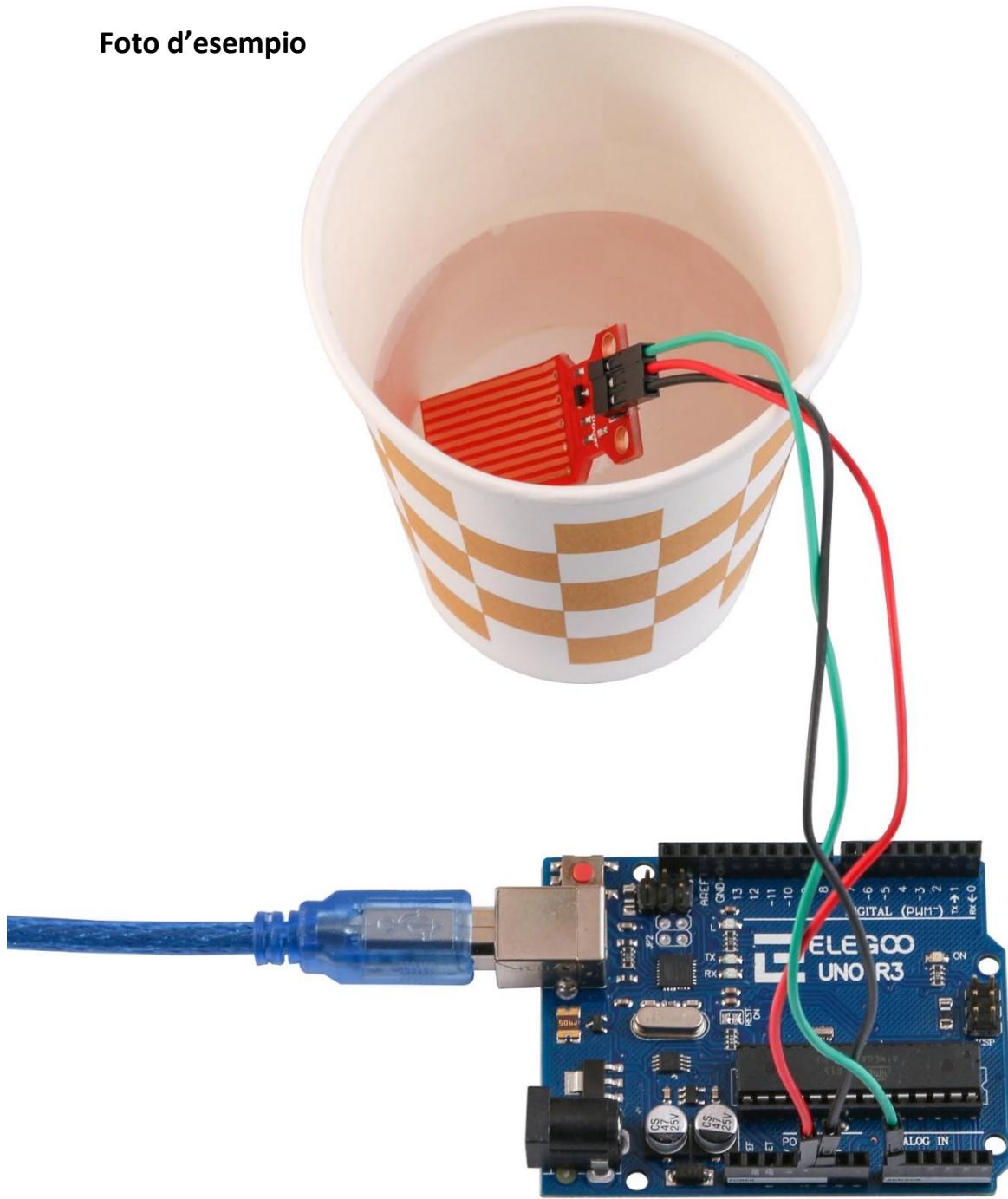


Suggerimenti per il collegamento: Il piedino di alimentazione (+) va connesso ai 5V della scheda UNO R3, la messa a terra (-) è connessa alla messa a terra GND. Il segnale di output (S) va connesso ad una delle porte analogiche (A0-A5), il piedino S sarà quello che darà il segnale del sensore. Una porta qualsiasi delle porte comprese tra la A0 e la A5 va bene. Ricordati però di definire la stessa porta che usi nel codice.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lessona 18 Water Level Detection Sensor Module”, clicca su UPLOAD per caricare il programma.

## Foto d'esempio



Aprendo il monitor seriale dovresti vedere dei dati simili a quelli qui sotto:

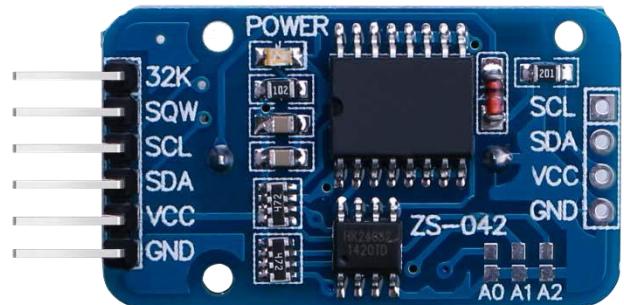
Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.

```
ADC0 level is 11
ADC0 level is 22
ADC0 level is 33
ADC0 level is 44
ADC0 level is 55
ADC0 level is 66
ADC0 level is 77
ADC0 level is 88
ADC0 level is 99
ADC0 level is 110
ADC0 level is 127
ADC0 level is 138
ADC0 level is 149
ADC0 level is 160
ADC0 level is 171
ADC0 level is 182
ADC0 level is 193
ADC0 level is 204
ADC0 level is 215
ADC0 level is 226
ADC0 level is 237
ADC0 level is 248
ADC0 level is 259
ADC0 level is 270
ADC0 level is 259
ADC0 level is 248
ADC0 level is 237
ADC0 level is 226
ADC0 level is 215
```

# Lezione 19 Modulo Orologio a tempo reale (Real Time Clock)

## Introduzione

In questa lezione imparerai ad utilizzare il modulo DS3231, questo modulo permette di visualizzare l'anno, il mese, il giorno, il minuto, il secondo e la settimana. Il modulo è alimentato da una batteria a bottone e continua ad andare anche se non è connesso alla scheda UNO con i 3 cavi dati.



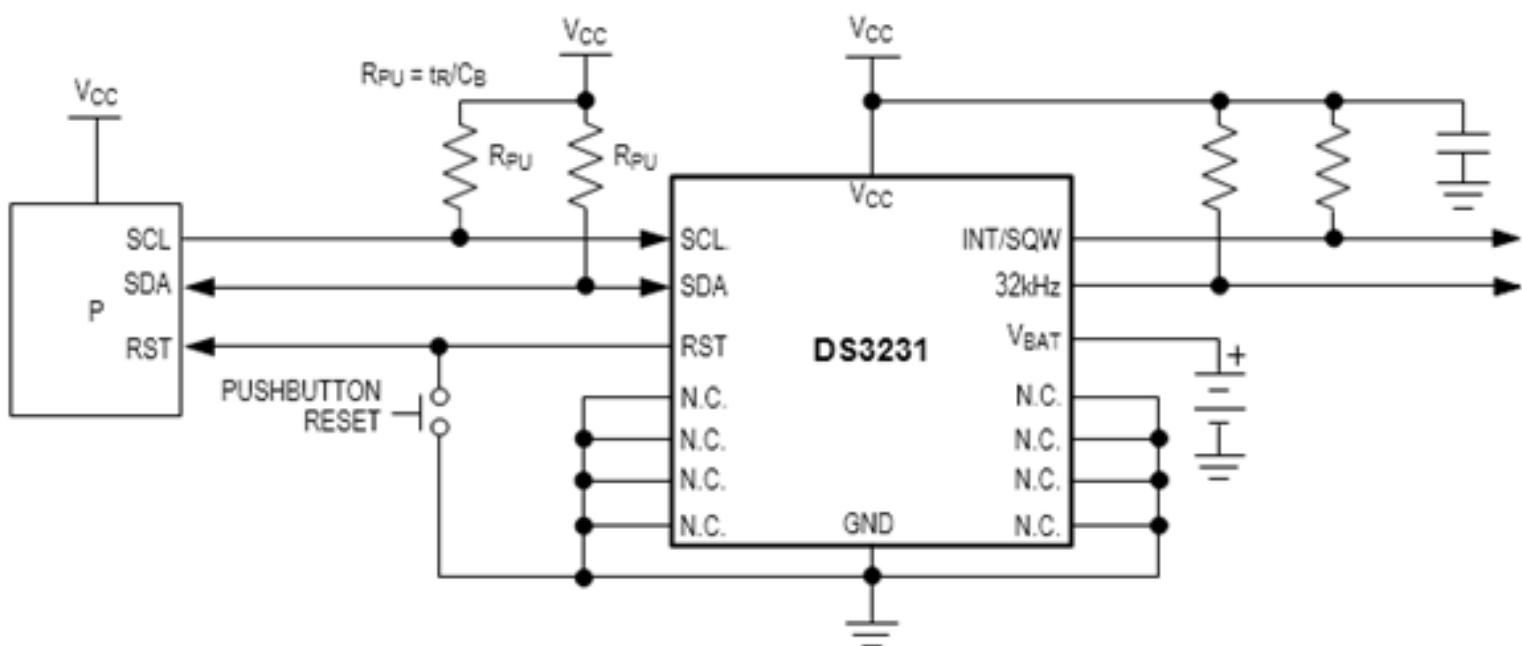
## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo RTC DS3231
- (4) x Connettori F-M (Connettori di DuPont Femmina – Maschio )

## Introduzione ai componenti

### DS3231

Il modulo DS3231 è un semplice chip che mantiene il tempo. Il modulo ha una batteria interna in modo che l'orologio non si fermi anche se sconnesso dalla scheda principale o se la scheda principale perde l'alimentazione.



## Connessione

### Schema

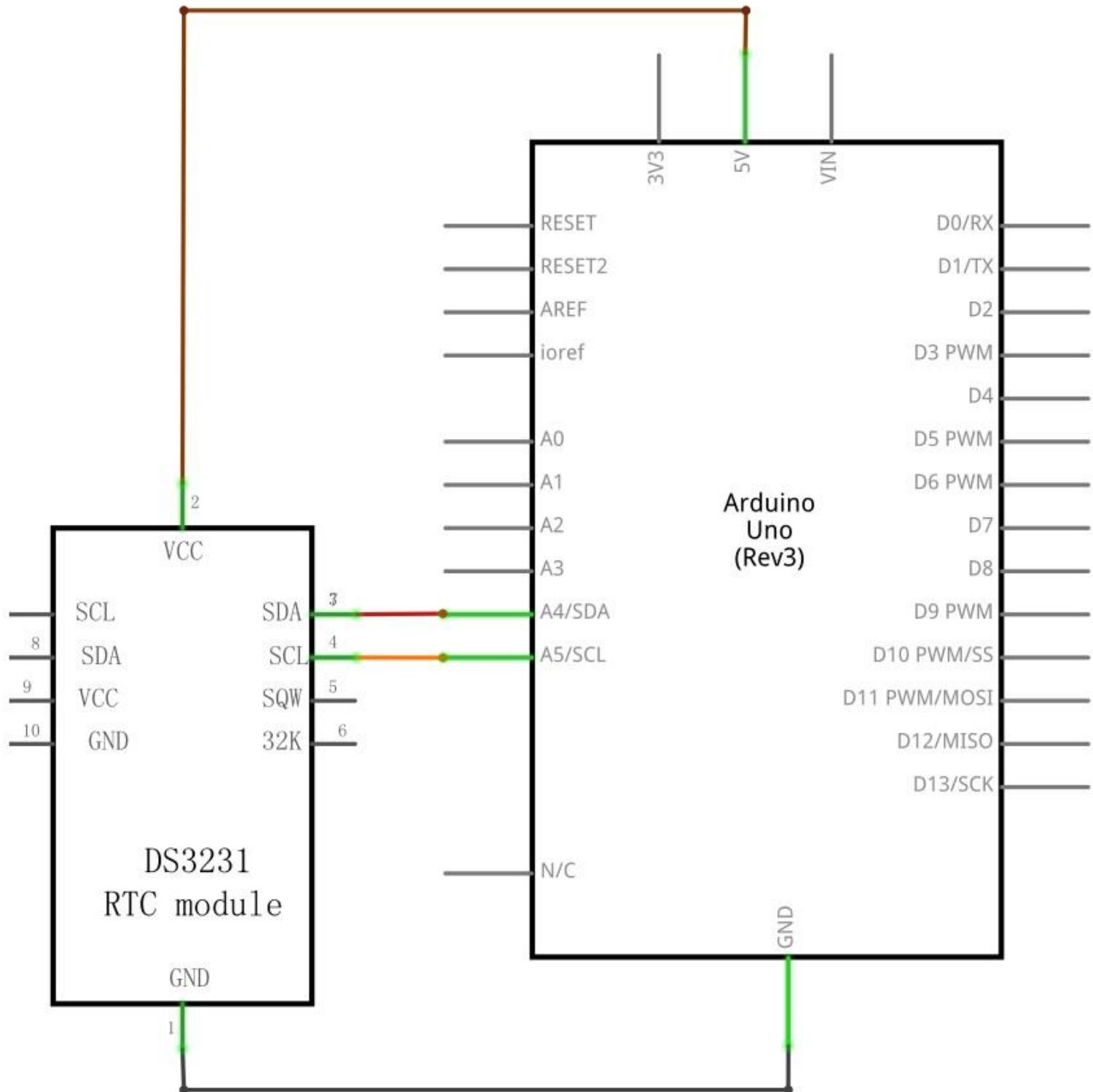
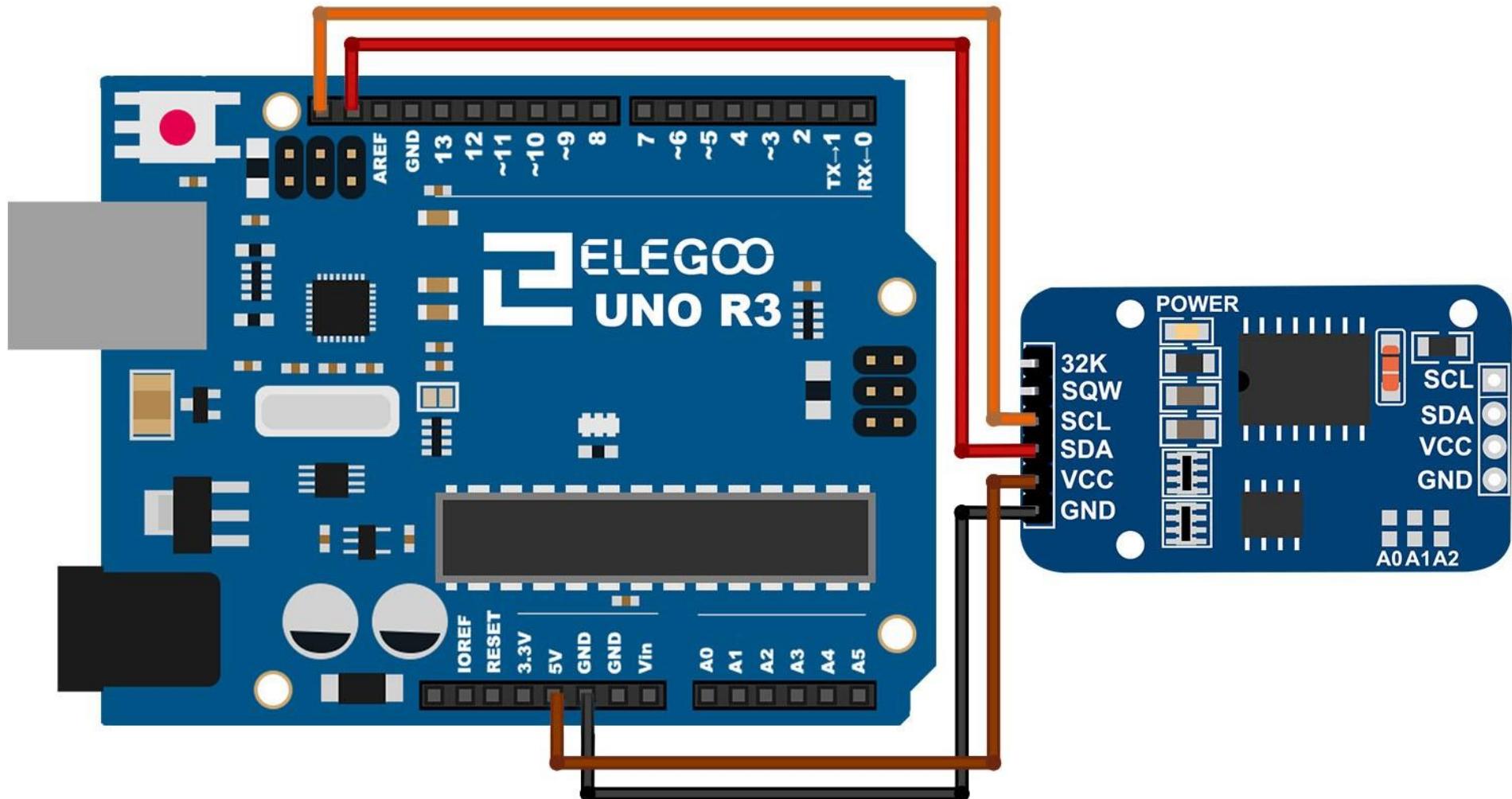


Diagramma di Collegamento



Imposta i collegamenti seguendo l'immagine successiva

Ignora i pin 32K e SQW, non ne avrai necessità. Connetti il pin SCL alla porta SCL della schedina UNO, ed il pin SDA alla porta SDA della schedina UNO.

Collega il pin VCC alla porta 5V e il pin GND alla porta GND.

## Codice

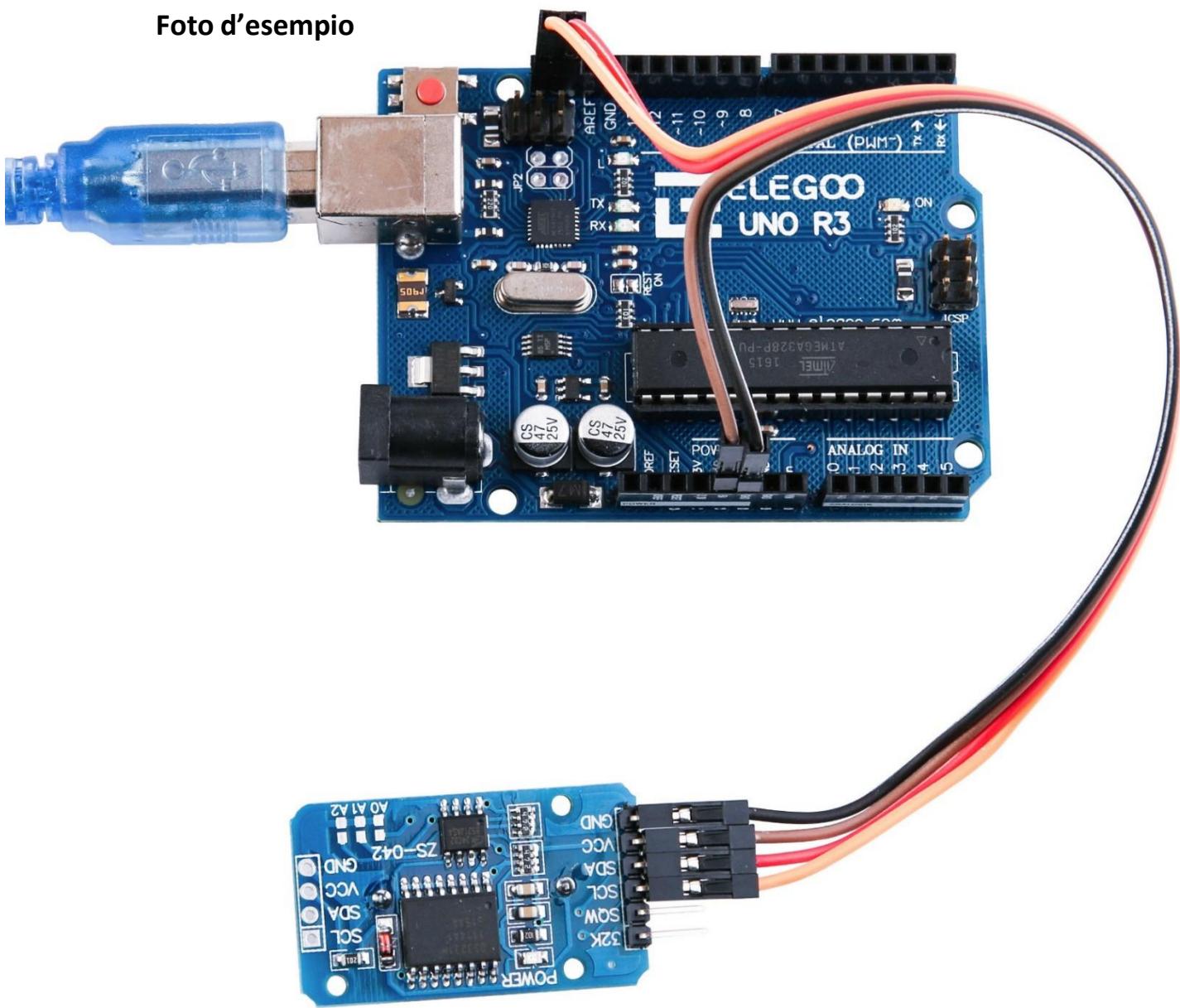
Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lessons 19 Real Time Clock Module", clicca su UPLOAD per caricare il programma.

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

Prima di eseguire questo codice, controlla di aver installato la libreria < DS3231 > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1

## Foto d'esempio



Aprendo il monitor seriale potrai vedere le letture del tempo rilevate dal monitor come qui sotto:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione.

```
Long format with month name: 09 December 2016 11:47:15
Short format with 12h mode: 9th Dec 16, 11:47am
Today is: Friday, 342 days of the year.
Actual month has: 31 days.
Unixtime: 1481194035

Long number format: 09-12-2016 11:47:17
Long format with month name: 09 December 2016 11:47:17
Short format with 12h mode: 9th Dec 16, 11:47am
Today is: Friday, 342 days of the year.
Actual month has: 31 days.
Unixtime: 1481194037

Long number format: 09-12-2016 11:47:18
Long format with month name: 09 December 2016 11:47:18
Short format with 12h mode: 9th Dec 16, 11:47am
Today is: Friday, 342 days of the year.
Actual month has: 31 days.
Unixtime: 1481194038

Long number format: 09-12-2016 11:47:19
Long format with month name: 09 December 2016 11:47:19
Short format with 12h mode: 9th Dec 16, 11:47am
Today is: Friday, 342 days of the year.
Actual month has: 31 days.
Unixtime: 1481194039
```

## Lezione 20 Modulo Sensore del Suono

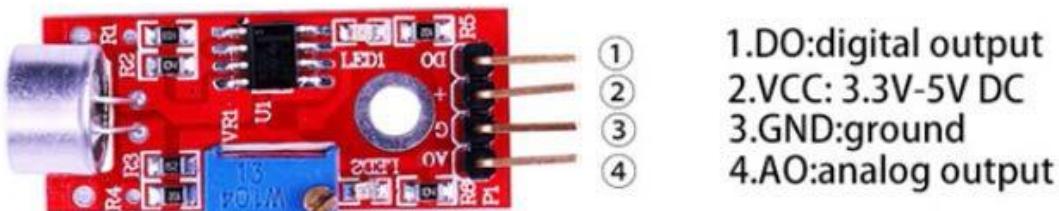
### Introduzione

In questa lezione imparerai ad utilizzare il Modulo con il sensore del suono. Questo modulo presenta due output:

AO: Output Analogico, è il segnale di voltaggio del microfono in tempo reale.

DO: quando l'intensità di segnale raggiunge e supera una certa soglia il segnale digitale viene portato ad HIGH. La soglia della sensibilità del suono può essere aggiustata variandone il valore tramite il potenziometro.

Per essere sicuri che il microfono riconosca la tua voce normalmente, prova a cambiare la sensibilità ruotando il potenziometro presente sul modulo. Data l'alta precisione, è necessario fare almeno 10 giri prima di avere qualche risposta.



### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo sensore del suono
- (4) x Connettori F-M (Connettori di DuPont Femmina-Maschio)

### Introduzione ai componenti

#### Microfono

I Trasduttori sono dispositivi che convertono l'energia da una forma all'altra. Un microfono è un trasduttore in grado di convertire l'energia del suono in segnale elettrico. Il microfono funziona al contrario di un altoparlante. I microfoni sono disponibili in diverse forme e misure. In base all'applicazione che ne si vuole fare, il microfono è composto con differenti tecnologie in grado di convertire il suono in segnale elettrico. Nel nostro caso, parleremo del microfono a condensatore, esso è largamente utilizzato in smartphones, computer, etc.

Come suggerisce il nome, questo tipo di microfono è composto da un condensatore a piastre parallele che lavora secondo il principio della capacità variabile. Esso consiste in due piatti, uno fisso (chiamato il piatto nero) ed uno mobile (chiamato diaframma), con un piccolo divario tra i due. Un potenziale elettrica carica i due piatti. Quando un suono colpisce il diaframma esso inizia a muoversi modificando la capacità dei due piani e di conseguenza modificato la corrente che li attraversa.



Questo tipo di microfoni sono largamente utilizzati nei circuiti elettronici per misurare suoni lievi e vibrazioni dell'aria che vengono poi convertiti in segnali elettrici. I due piedini che puoi vedere nell'immagine qui sopra sono utilizzati per connettere il sensore al circuito.



Un corpo conduttivo di solido metallo contiene le varie parti del microfono. La faccia superiore è coperta da un materiale poroso che sta in posizione con l'aiuto di colla. Questo materiale serve per filtrare le particelle di polvere. I segnali sonori e le vibrazioni dell'aria passano attraverso il materiale poroso ed interferiscono con il diaframma attraverso il buco che puoi vedere nell'immagine qui sopra.

## Connessione

### Schema

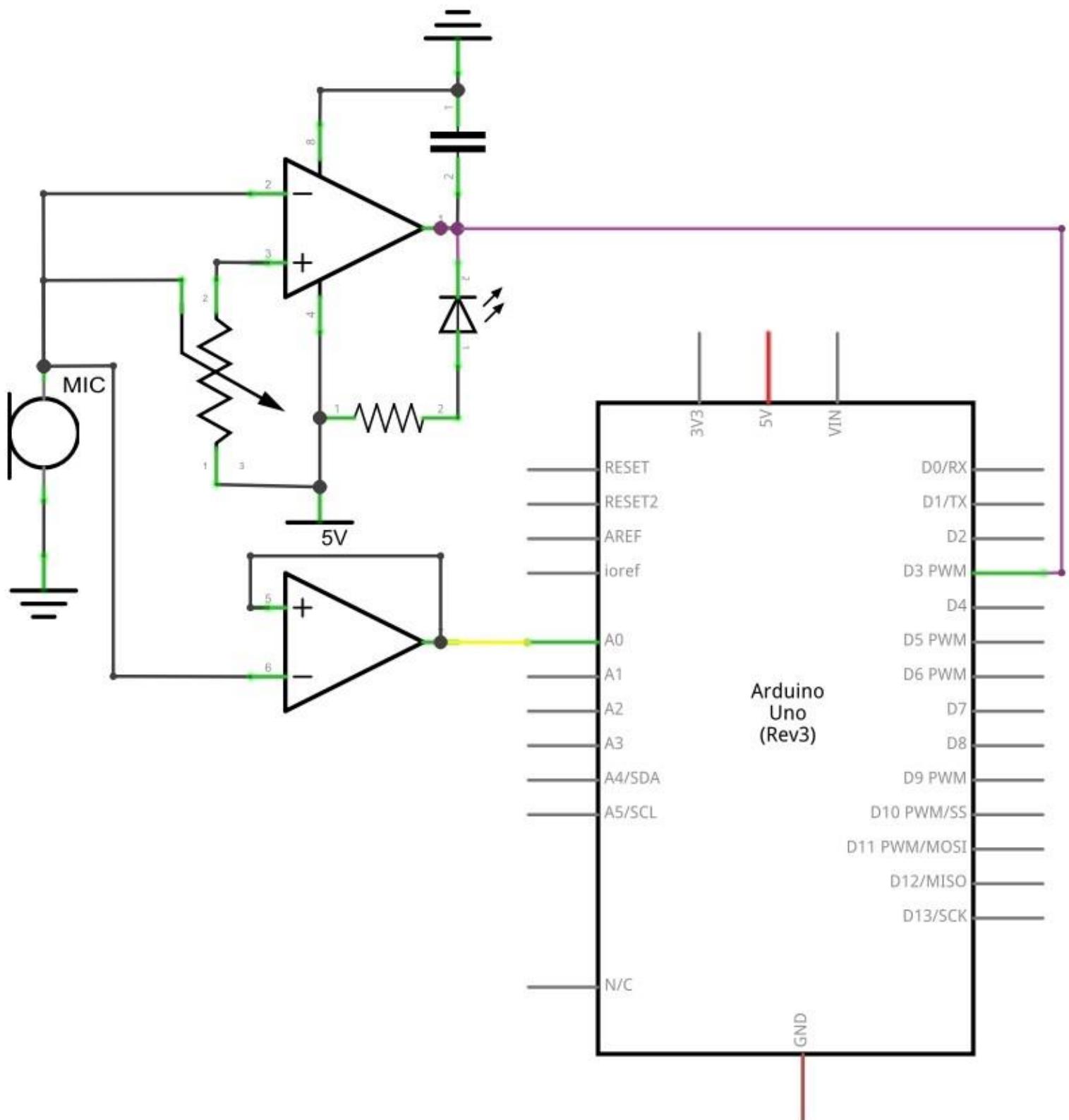
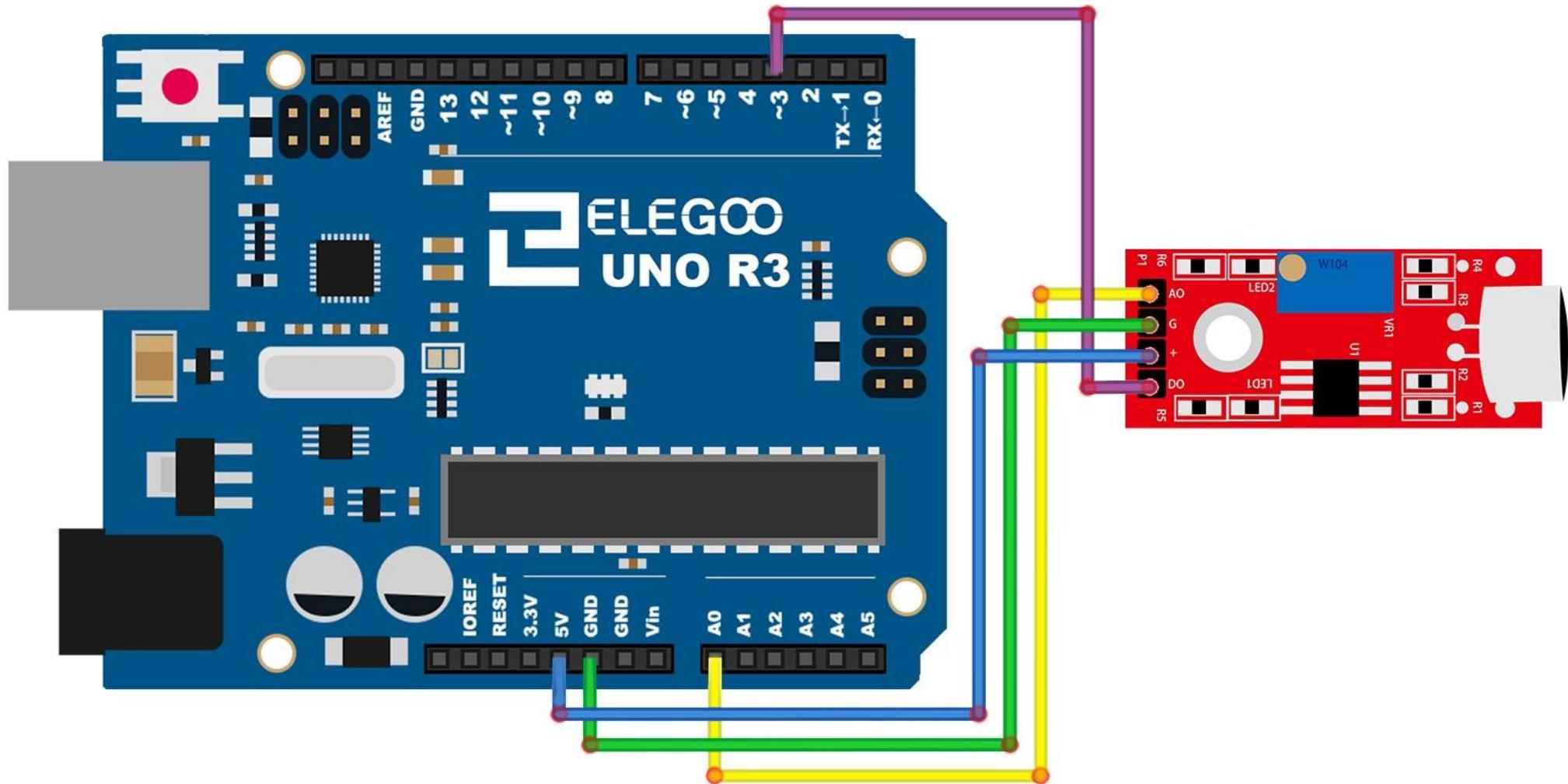


Diagramma di Collegamento



## Codice

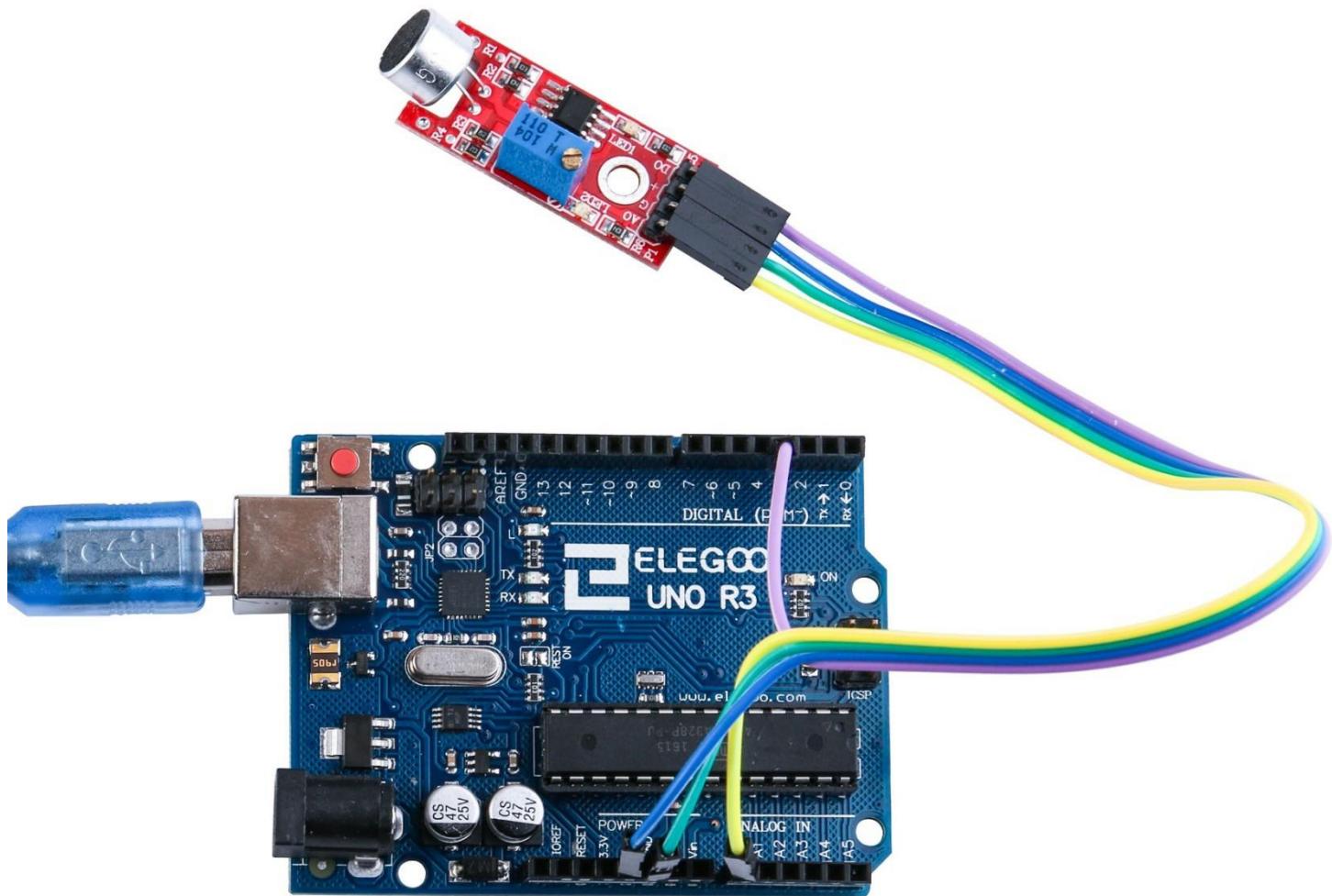
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 20 Sound Sensor Module”, clicca su UPLOAD per caricare il programma.

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

Avendo questo modulo due segnali di output è possibile utilizzarlo con diverse modalità, perciò avviamo scritto due file di codici `digital_signal_output` e `analog_signal_output`. Il codice `digital_signal_output` riconosce quando la voce raggiunge un certo valore, ed attiva il segnale digitale sul pin 11 della scheda Arduino portando il segnale ad HIGH e l’indicatore LED L si accenderà in quel preciso istante. La sensibilità di innesco del segnale può essere cambiata come spiegato sopra, con il potenziometro.

Il codice `analog_signal_output` legge il valore analogico rilevato dal modulo e lo trasmetterà alla scheda Arduino che lo visualizzerà sul monitor seriale. Questo valore può essere cambiato tramite gli aggiustamenti di sensibilità di cui abbiamo parlato sopra.

## Foto d’esempio



Aprendo il monitor seriale potrai vedere delle letture come qui sotto:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione

```
52
51
52
52
51
52
52
50
48
48
50
51
53
124
50
54
52
46
49
51
52
52
53
52
52
52
54
```

Autoscroll      Newline      9600 baud

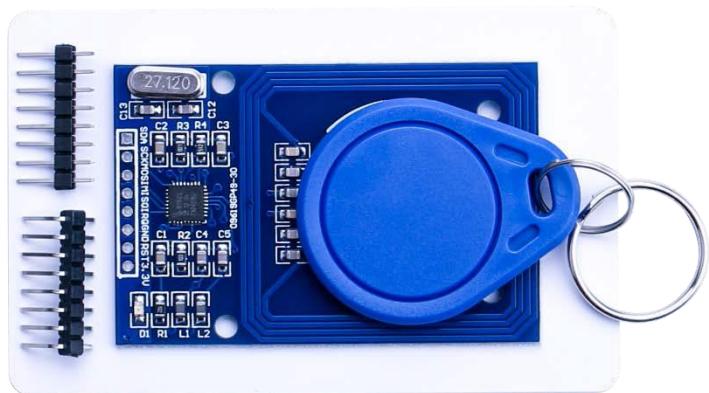
## Lezione 21 Modulo RFID RC522

### Introduzione

In questa lezione imparerai come utilizzare il modulo RFID RC5522 con la tua scheda UNO R3. Questo modulo utilizza un'interfaccia di comunicazione seriale (Serial Peripheral Interface - SPI) per comunicare con controllori come Arduino, raspberry etc.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo RFID RC522
- (7) x Connettori F-M (connettori di DuPont Femmina Maschio)



### Introduzione ai componenti

#### RC522

Il modulo MFRC522 ha integrato un lettore/scrittore per comunicazioni contactless a 13.65MHz. Il modulo MFRC522 supporta lo standard ISO 14443A / MIFARE®.

I componenti interni del MFRC5522 adoperati a trasmettere wireless sono in grado di leggere e scrivere tramite un'antenna disegnata per comunicare con carte e transponder con circuiti attivi secondo lo standard ISO/IEC 14443A/MIFARE®.

La parte ricevente ha a disposizione un robusto ed efficiente circuito di demodulazione e decodifica per i segnali ISO/IEC 14443A/MIFARE® ricevuti da carte e transponder compatibili. La parte digitale comprende un completo frame ISO/IEC 14443A con rilevazioni d'errori (accoppiamento e controllo a ridondanza ciclica).

Il modulo MFRC5522 è compatibile con i prodotti MIFARE®Classic (e.g. MIFARE® Standard). Il modulo MFRC5522 supporta le comunicazioni MIFARE® contactless che utilizzano una velocità di trasferimento superiore a 848kbit/s in entrambe le direzioni.

Varie interfacce sono implementate:

- Interfaccia SPI
- Seriale UART (Simile alla RS232 con livelli di voltaggio adeguati)
- Interfaccia I2C

L'immagine qui sotto mostra un diagramma tipico del circuito da implementare per

utilizzare il modulo MFRC5522.

## Connessione

### Schema

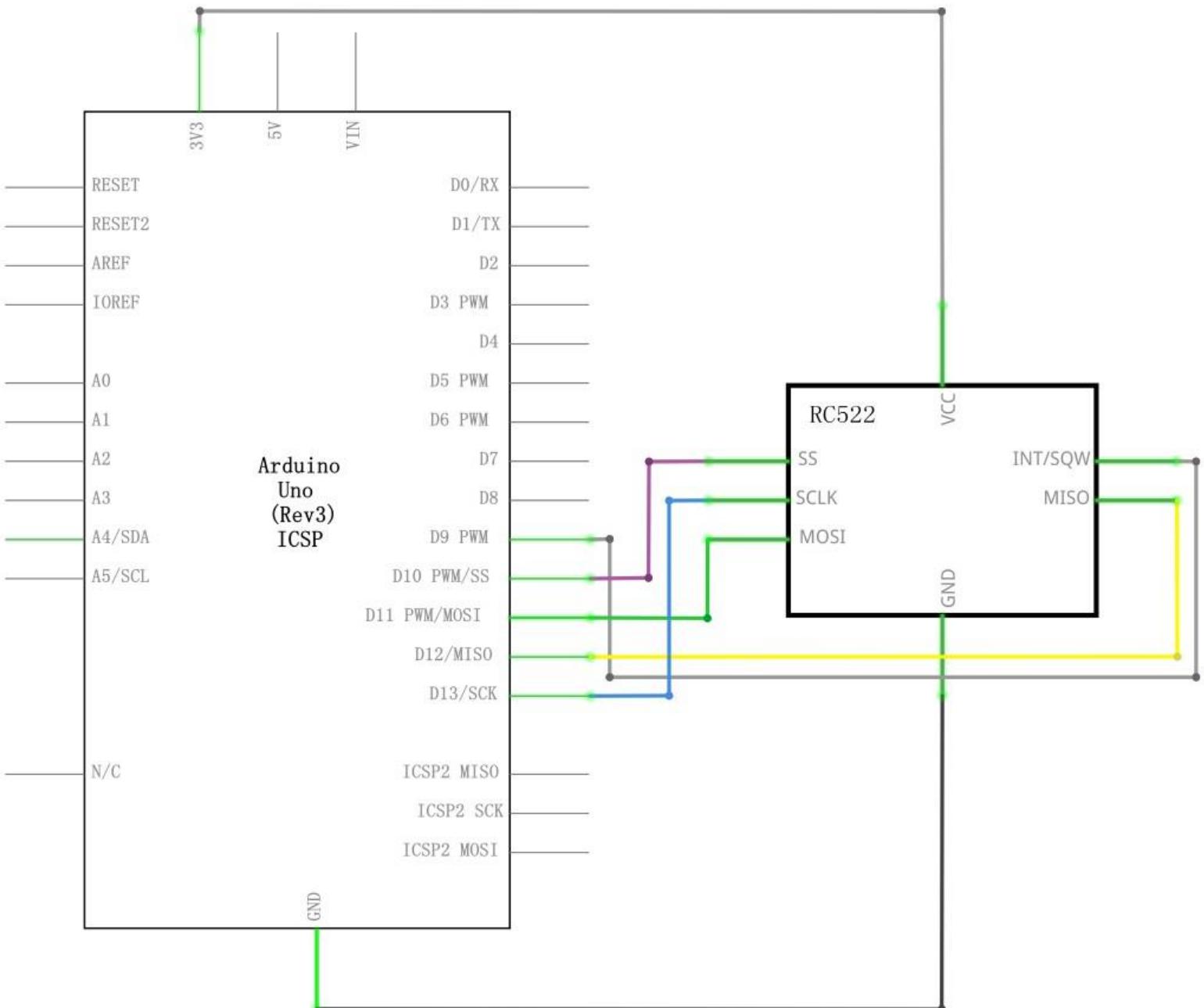
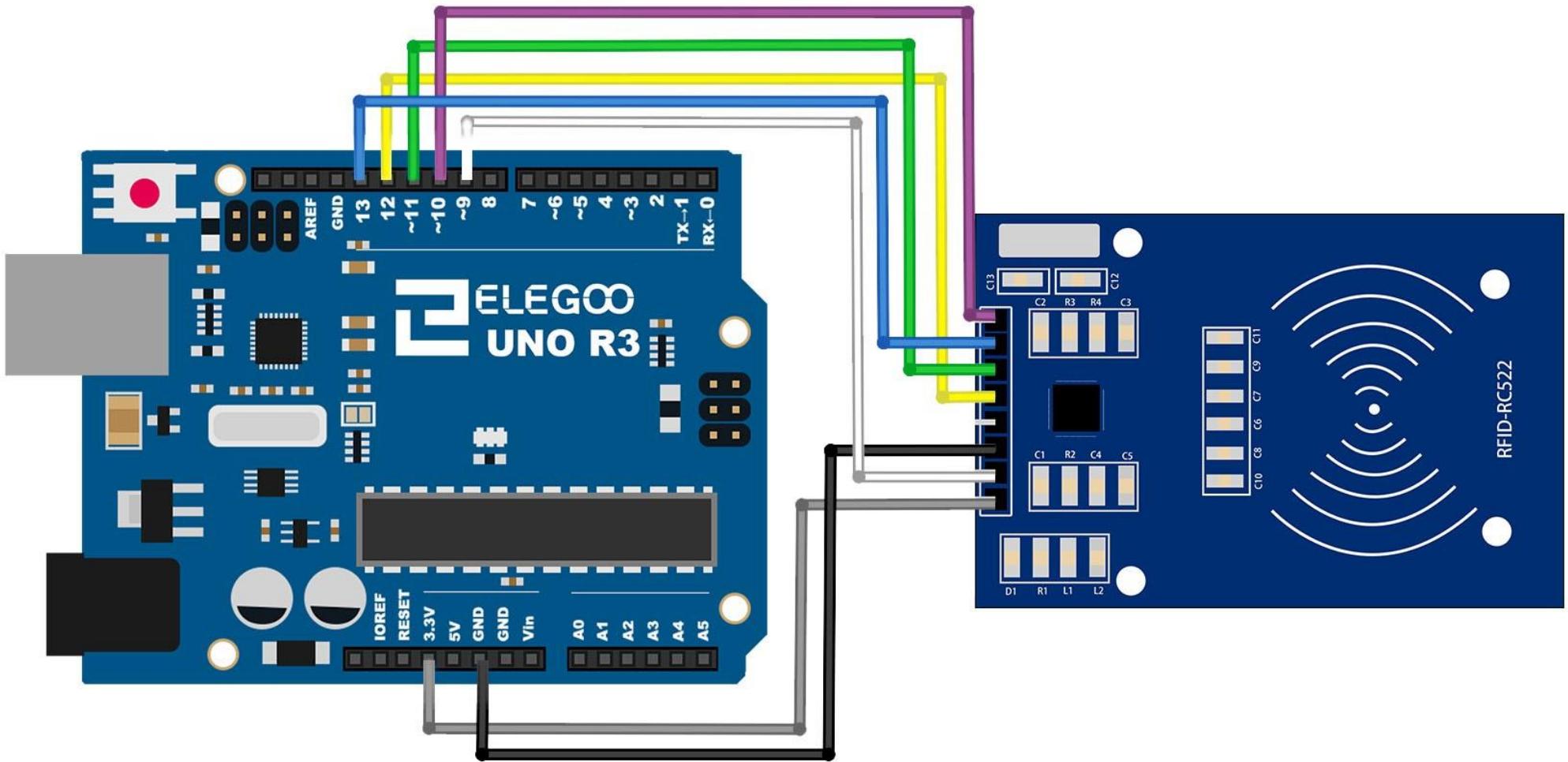


Diagramma di collegamento



## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 21 RC522 RFID Module”, 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.

Prima di eseguire questo codice, controlla di aver installato la libreria < rfid > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

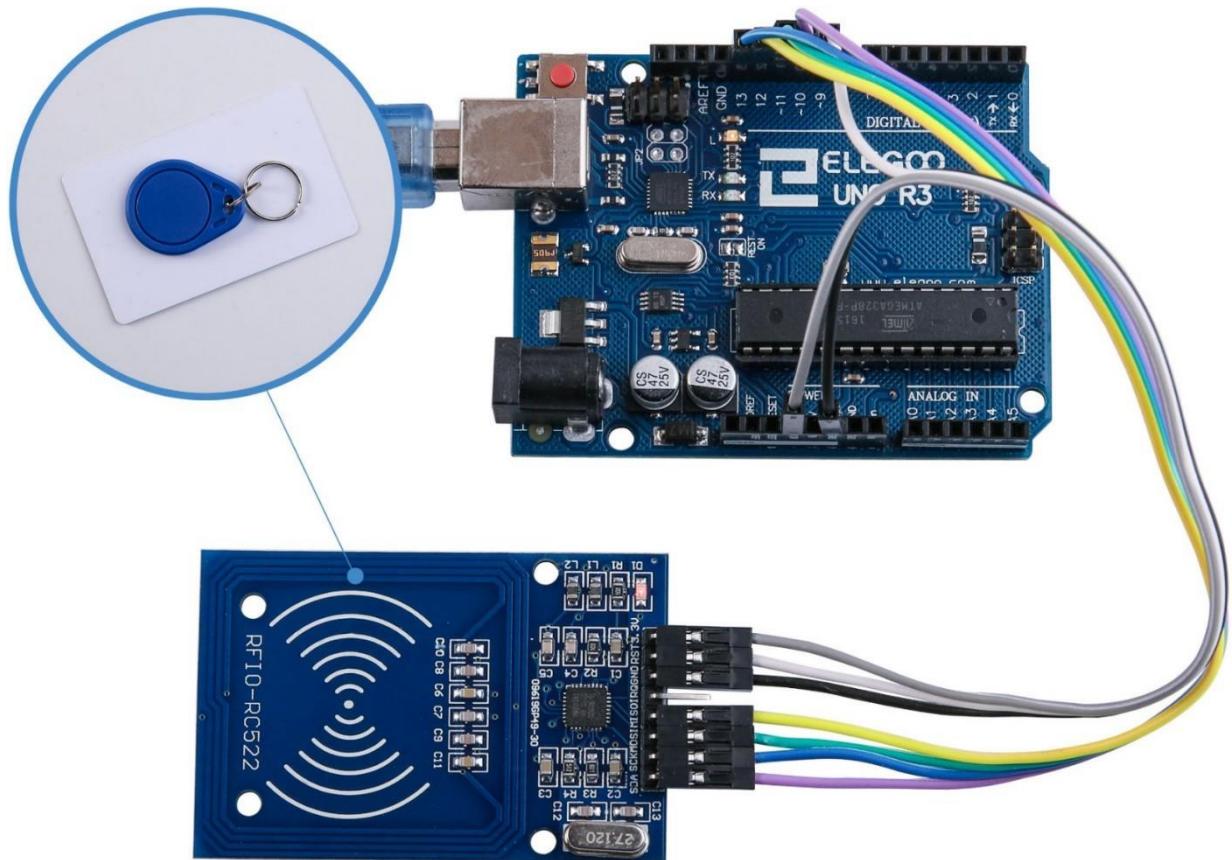
	MFRC522	Arduino Uno	Arduino Mega	Arduino Nano v3	Arduino Leonardo/Micro	Arduino Pro Micro
Signal	Reader/PCD	Pin	Pin	Pin	Pin	Pin
RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
SPI SS	SDA (SS)	10	53	D10	10	10
SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

```
#define RST_PIN      5      // Configurable, see typical pin layout above
```

```
#define SS_PIN       53     // Configurable, see typical pin layout above
```

La posizione del pin SPI varia in base al chip che utilizziamo perciò forse avrai da fare piccole modifiche alla funzione.

## Foto d'esempio



Aprendo il monitor seriale potrai vedere delle letture come qui sotto:

Clicca sul tasto del monitor seriale per attivarlo. Le basi riguardo al monitor seriale sono state introdotte nella lezione 1, se hai qualche dubbio torna a tale lezione

The screenshot shows a Windows-style application window titled "COM6 (Arduino/Genuino Mega or Mega 2560)". Inside the window, there is a text area containing the following command-line output:

```
Warning: this example overwrites the UID of your UID changeable card, use with care
Card UID: 66 E3 F4 74
Card did not respond to 0x40 after HALT command. Are you sure it is a UID changeable?
Error name: Timeout in communication.
Activating the UID backdoor failed.
New UID and contents:
Card UID: 66 E3 F4 74
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
 15   63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
        62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
 14   59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
        58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
 13   55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
        54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
 12   51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
        50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
 11   47 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
        46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
```

At the bottom of the window, there are three buttons: "Autoscroll" (checked), "Newline" (dropdown menu), and "9600 baud" (dropdown menu).

## Lezione 22 Display LCD

### Introduzione

In questa lezione, imparerai come collegare ed utilizzare un display alfanumerico LCD. Questo display ha una retroilluminazione LED e può visualizzare due righe da 16 caratteri ognuna. Sul display puoi vedere un rettangolino per ogni carattere del display ed i pixel che formano ogni carattere. Il display è solo bianco e blu ed è pensato per visualizzare solo del testo.

In questa lezione eseguiremo il codice d'esempio per la libreria del display LCD, ma nella prossima lezione utilizzeremo il display per mostrare la temperatura rilevata da un sensore.



### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo LCD1602
- (1) x Potenziometro (10k)
- (1) x Breadboard con 830 fori
- (16) x Connettori M-M (connettori Maschio-Maschio)

### Introduzione ai componenti:

#### LCD1602

Specifiche dei pin del modulo LCD1602:

**VSS:** Il pin da connettere alla messa a terra

**VDD:** Il pin che verrà connesso alla sorgente di voltaggio +5 V

**VO:** Il pin atto ad aggiustare il contrasto del display LCD1602

**RS:** Il pin di selezione del registro, che ti permette di controllare su quale area della memoria del display LCD si andrà a scrivere il nuovo carattere. Puoi selezionare il pin del registro dei dati, che mantiene ciò che andremo a scrivere sullo schermo, oppure il registro delle istruzioni, il quale riceve le istruzioni su cosa fare diseguito.

**R/W:** Questo pin serve a selezionare la modalità di lettura o scrittura.

**E:** il pin di abilitazione, quando alimentato con bassi livelli di energia, determina l'esecuzione di istruzioni rilevanti sul modulo LCD.

**D0-D7:** Pin che leggono e scrivono i dati.

**A and K:** Pin che controllano la retroilluminazione.

## Connessione Schema

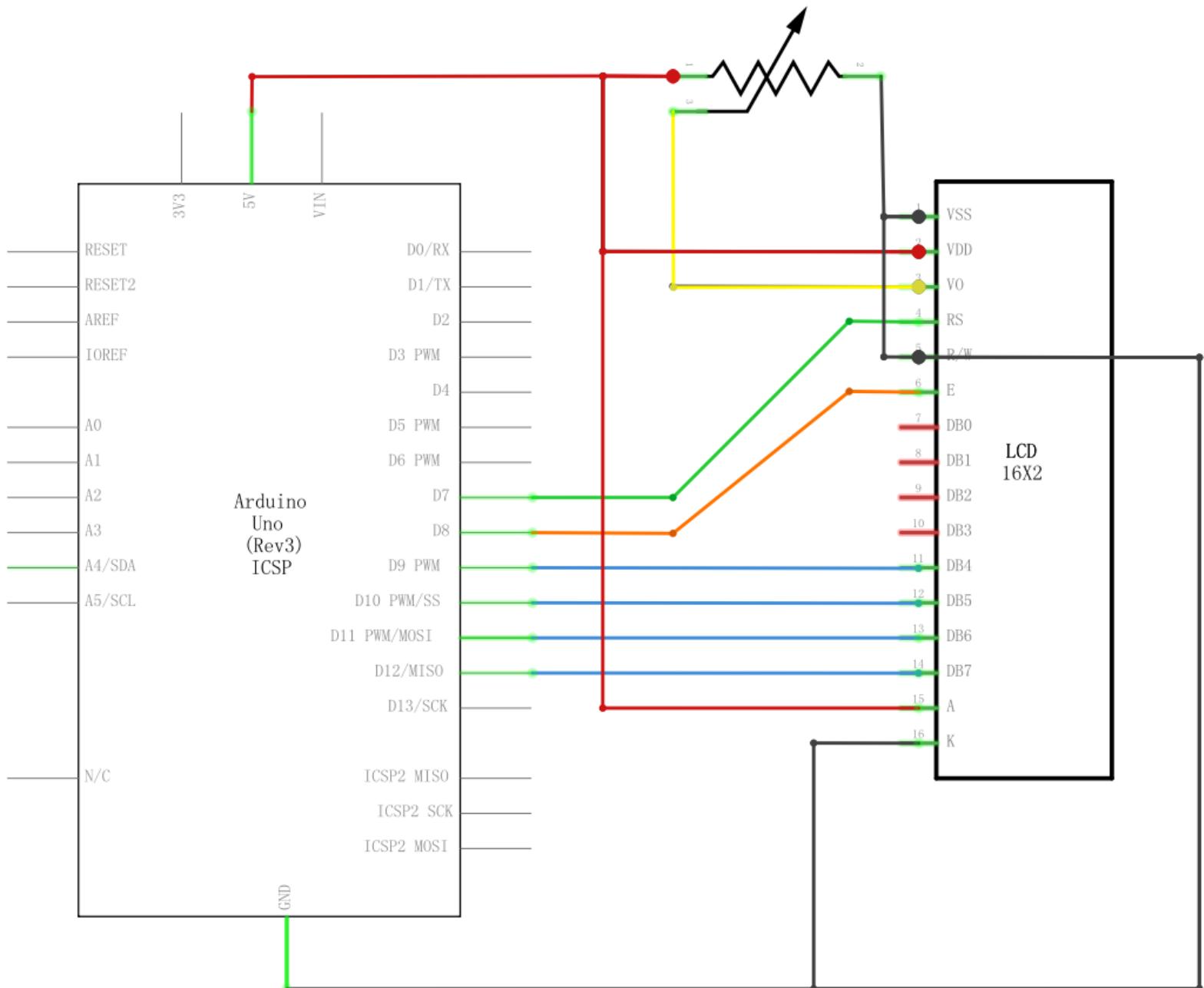
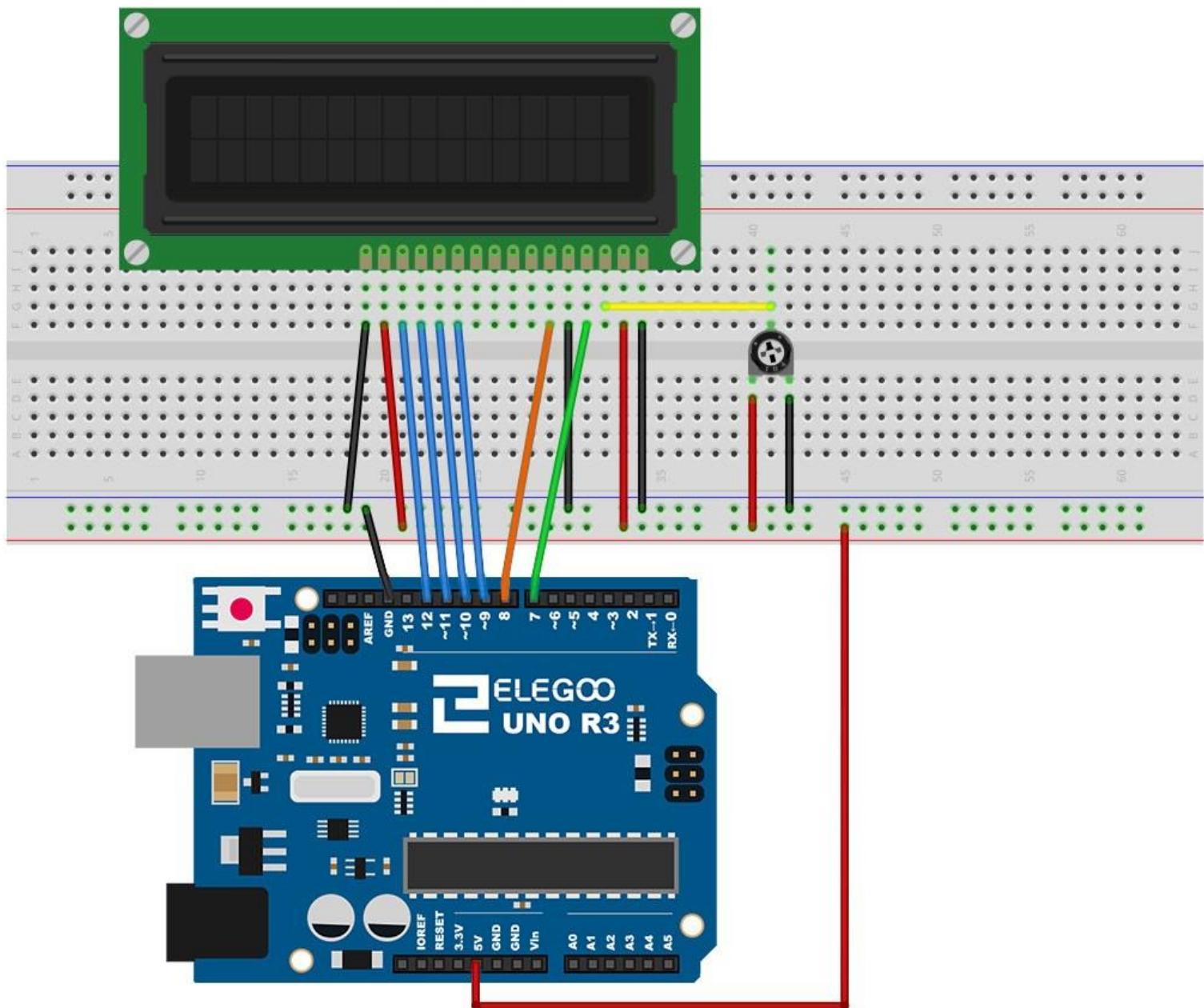


Diagramma di collegamento



Il display LCD necessita di sei pin di Arduino, tutti i pin devono essere output digitali. Il display necessita anche di essere connesso ai 5V e alla messa a terra GND. Ci sono numerose connessioni da effettuare, allineare il display alla parte alta della breadboard può risultarti utili ad identificare i pin senza fare troppa confusione, specialmente se la breadboard ha le righe numerate. Non dimenticare di collegare il cavo giallo (in figura) che connette il potenziometro al pin 3 del display. Il potenziometro viene utilizzato per controllare il contrasto del display. Potresti scoprire che il display viene alimentato senza il pin di testa collegato ad esso. Se è così, segui le istruzioni nella prossima sezione.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 22 LCD Display”, 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. Prima di eseguire questo codice, controlla di aver installato la libreria < LiquidCrystal > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare. Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

Dopo aver caricato il codice sulla scheda Arduino dovresti vedere il messaggio 'hello, world' apparire sul display, seguito da un numero crescente a partire da zero.

La prima cosa che trovi nel codice è la linea:

`#include <LiquidCrystal.h>`

Questa linea specifica ad Arduino di utilizzare la libreria Liquid Crystal.

Di seguito trovi una linea da modificare. Essa definisce quali pin di Arduino sono i corrispettivi connessi al display.

`LiquidCrystal lcd(7, 8, 9, 10, 11, 12);`

Dopo aver caricato questo codice assicurati che la retroilluminazione sia accesa e regola il potenziometro in modo da vedere il messaggio di testo predisposto.

Nella funzione “setup” ci sono due comandi:

`lcd.begin(16, 2);`

`lcd.print("Hello, World!");`

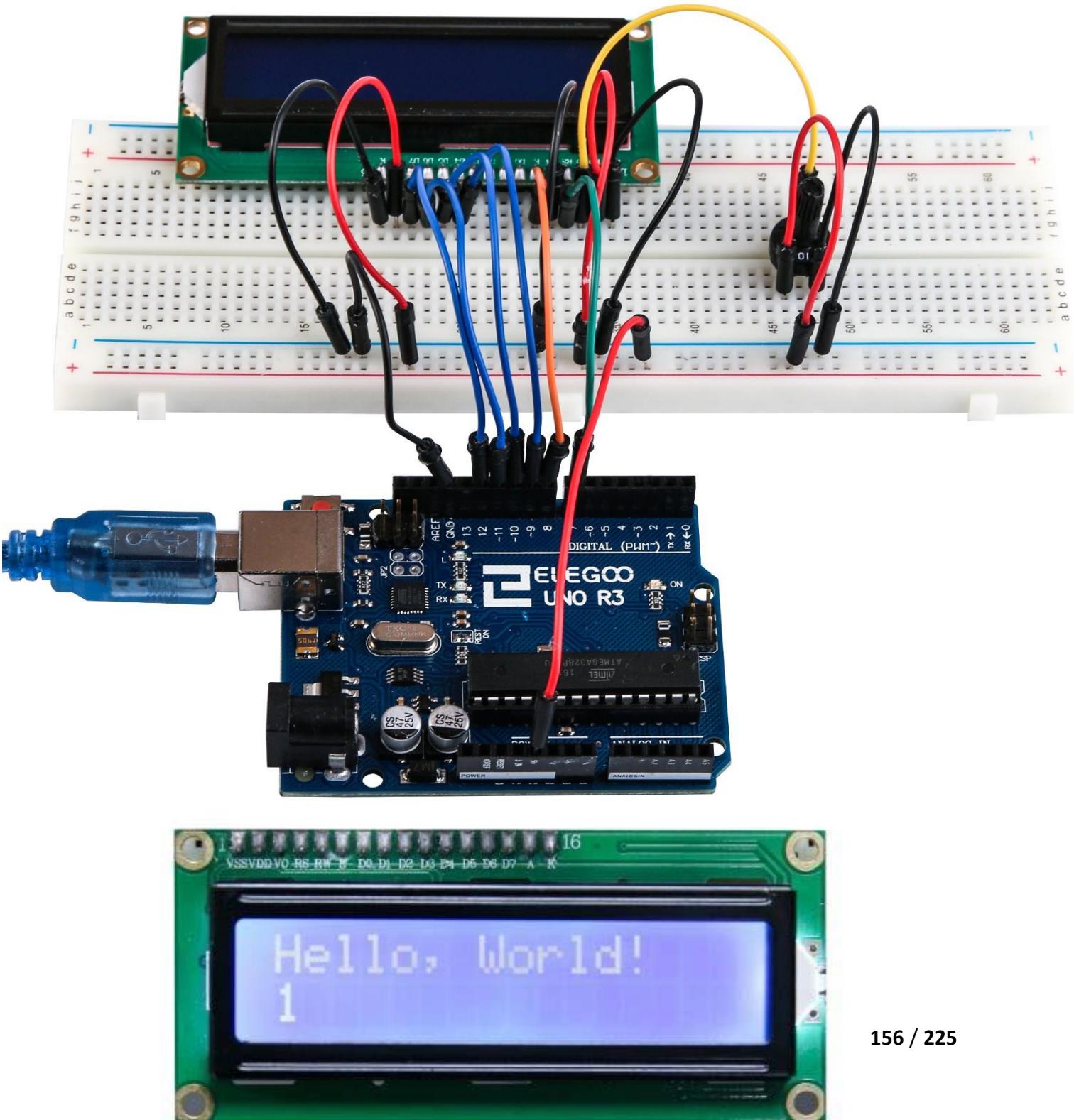
Il primo definisce nella libreria Liquid Crystal quante colonne e righe ha il display con il quale andremo ad interfacciarcì. La seconda linea visualizza il messaggio che vedremo sullo schermo.

Nella funzione di “loop” abbiamo altri due comandi:

```
lcd.setCursor(0, 1);  
lcd.print(millis()/1000);
```

Il primo imposta la posizione del cursore, perciò dove il prossimo testo apparirà (nel nostro caso colonna 0 e riga 1). Colonne e righe iniziano la numerazione da 0 e non da 1. La seconda linea visualizza il numero di millisecondi passati da quando la scheda Arduino è stata resettata.

### Foto d'esempio



## Lezione 23 Termometro

### Introduzione

In questa lezione, utilizzerai un display LCD per visualizzare la temperatura.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Modulo LCD1602
- (1) x Resistenza 10k ohm
- (1) x Termistore
- (1) x Potenziometro
- (1) x Breadboard con 830 punti di collegamento
- (18) x Connettori M-M (Connettori Maschio – Maschio)

### Introduzione ai componenti

#### Termistore

Un Termistore è una resistenza termica – una resistenza che cambia la sua resistenza in base alla sua temperatura. Tecnicamente tutte le resistenze sono termistori, infatti la loro resistenza cambia leggermente con la temperatura, ma questo cambiamento è solitamente molto lieve e difficile da misurare. I termistori sono costruiti in modo che il loro livello di resistenza cambi drasticamente con il variare della temperatura, questa variazione può essere anche 100 ohm per grado!

Ci sono due tipi di termistori, NTC (negative temperature coefficient) e PTC (positive temperature coefficient). In generale vedrai per misurare la temperatura sensori NTC. I termistori PTC sono spesso utilizzati come fusibili resettabili, un aumento di temperatura incrementa la resistenza, questi significa che più corrente passa attraverso di essi più essi si scalzano e bloccano la corrente, questo è molto utile per proteggere i circuiti.

## Connessione Schema

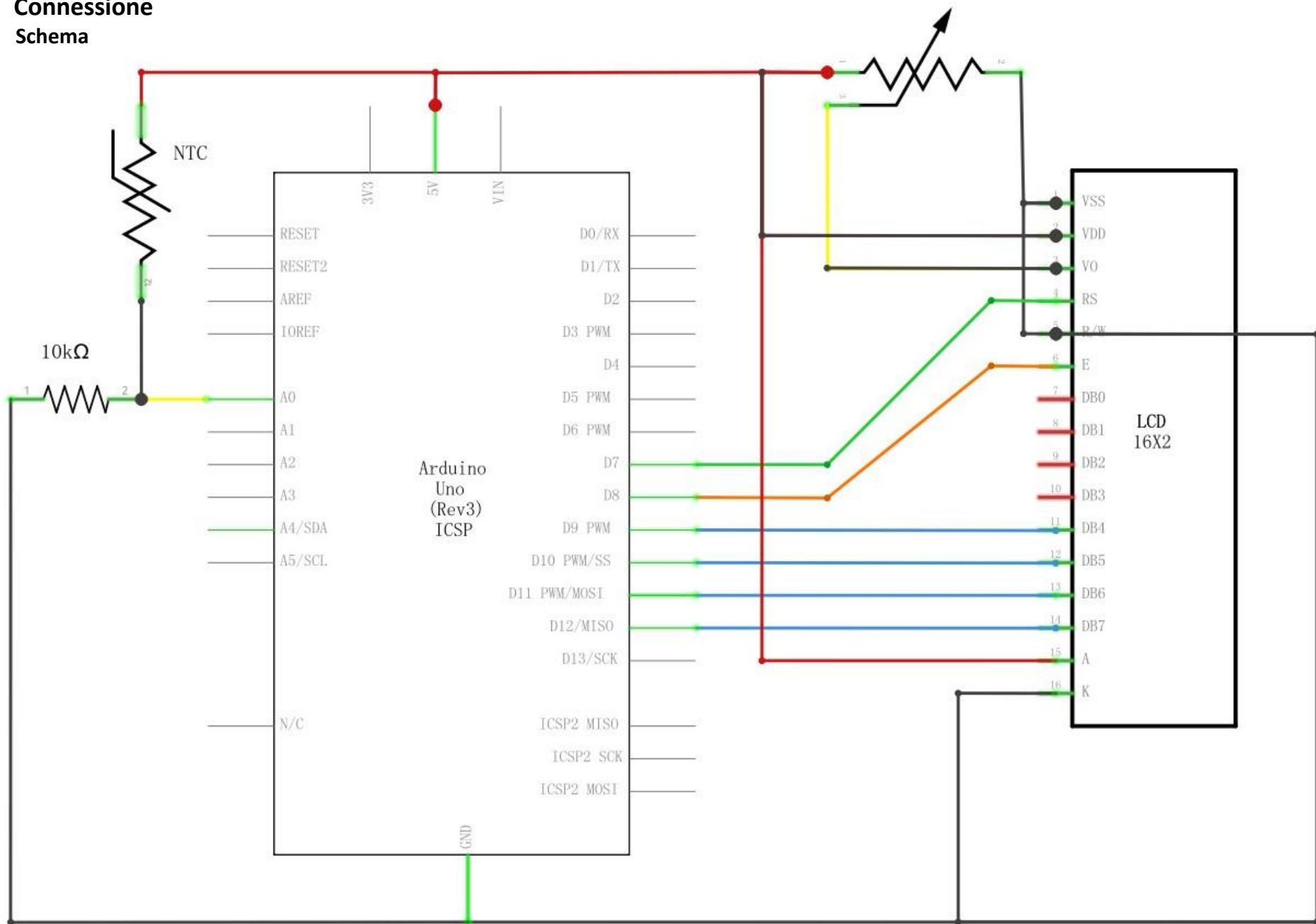
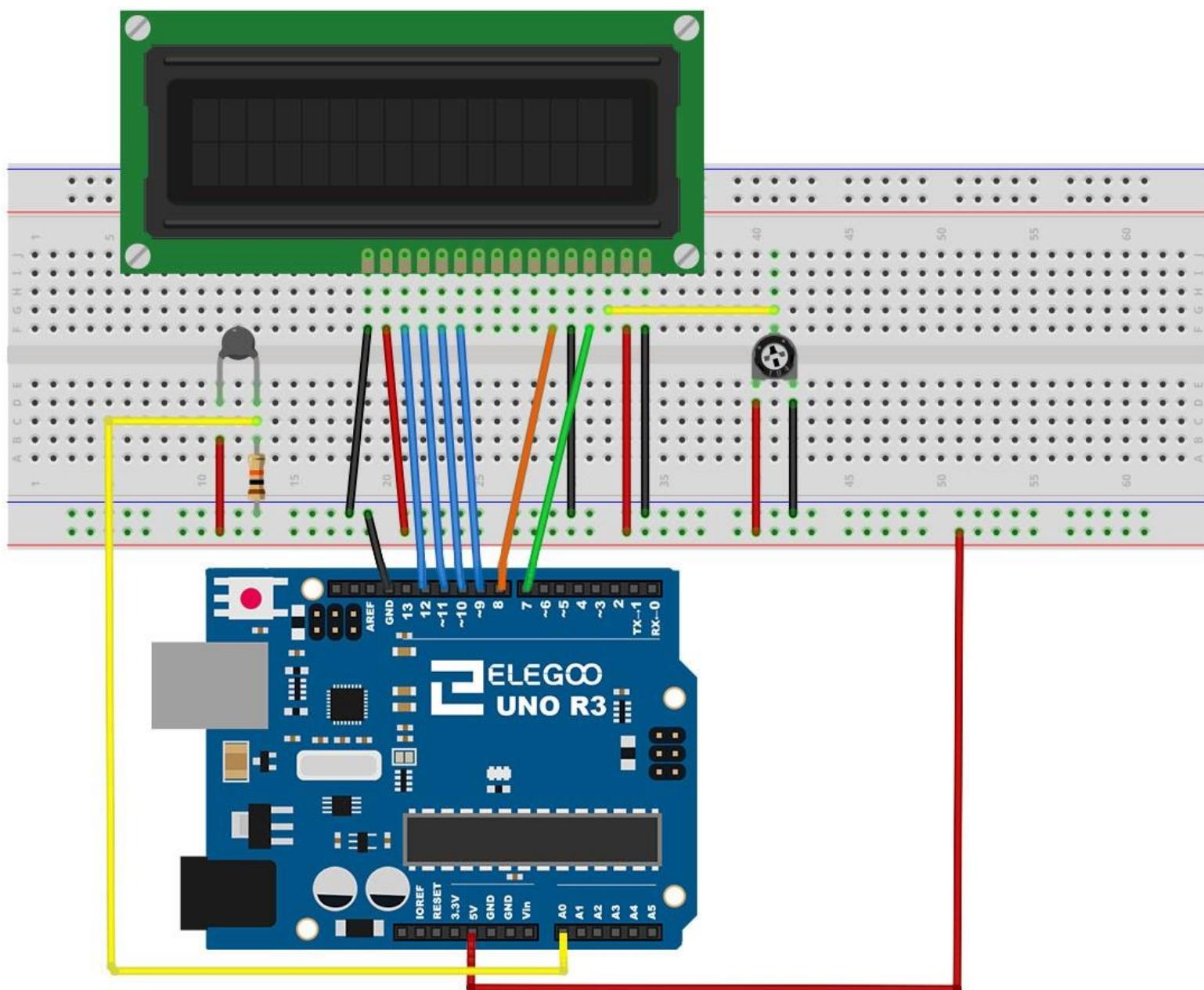


Diagramma di collegamento



Il layout montato sulla breadboard è basato sul layout visto nella lezione 22, in modo da semplificare l'apprendimento di ciò che si ha sulla breadboard.

Ci sono alcuni cavetti di collegamento vicino al potenziometro che sono stati leggermente spostati in questo layout.

La resistenza da 10 kΩ ed il termistore sono entrambi delle aggiunte rispetto alla lezione già vista.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 23 Thermometer”, 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.

Prima di eseguire questo codice, controlla di aver installato la libreria <LiquidCrystal> oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

Il codice di questo esercizio è basato sul codice della lezione 22. Caricalo su Arduino e dovresti vedere la temperatura sul display cambiare semplicemente stringendo tra le tue dita il termistore.

Abbiamo pensato potesse essere utile inserire una linea di commento sopra al comando dell’LCD.

```
// BS E D4 D5 D6 D7  
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Questo ti rende più facile la situazione nel caso tu decidessi di cambiare i pin utilizzati.

Nella funzione “loop” accadono due cose interessanti. Primo è necessario convertire l’ingresso analogico della lettura di temperatura del sensore nella temperatura attuale. In seguito dobbiamo poi eseguire delle operazioni per visualizzare il dato ottenuto.

Per prima cosa, guarda come viene calcolata la temperatura.

```
int tempReading = analogRead(tempPin);  
double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));  
tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK ))  
* tempK );  
float tempC = tempK - 273.15;
```

```
float tempF = (tempC * 9.0) / 5.0 + 32.0;
```

Visualizzare il cambio di temperatura sul display LCD può risultare leggermente complicato. Il problema maggiore nella scrittura può essere che il nuovo numero non avrà lo stesso numero di cifre di quello precedente. Così per esempio, se la temperatura passa da 101.50 a 99.00 c'è il pericolo che un numero extra della vecchia lettura resti sul display. Per evitare questo, riscrivi l'intera linea del display LCD ogni ciclo di loop.

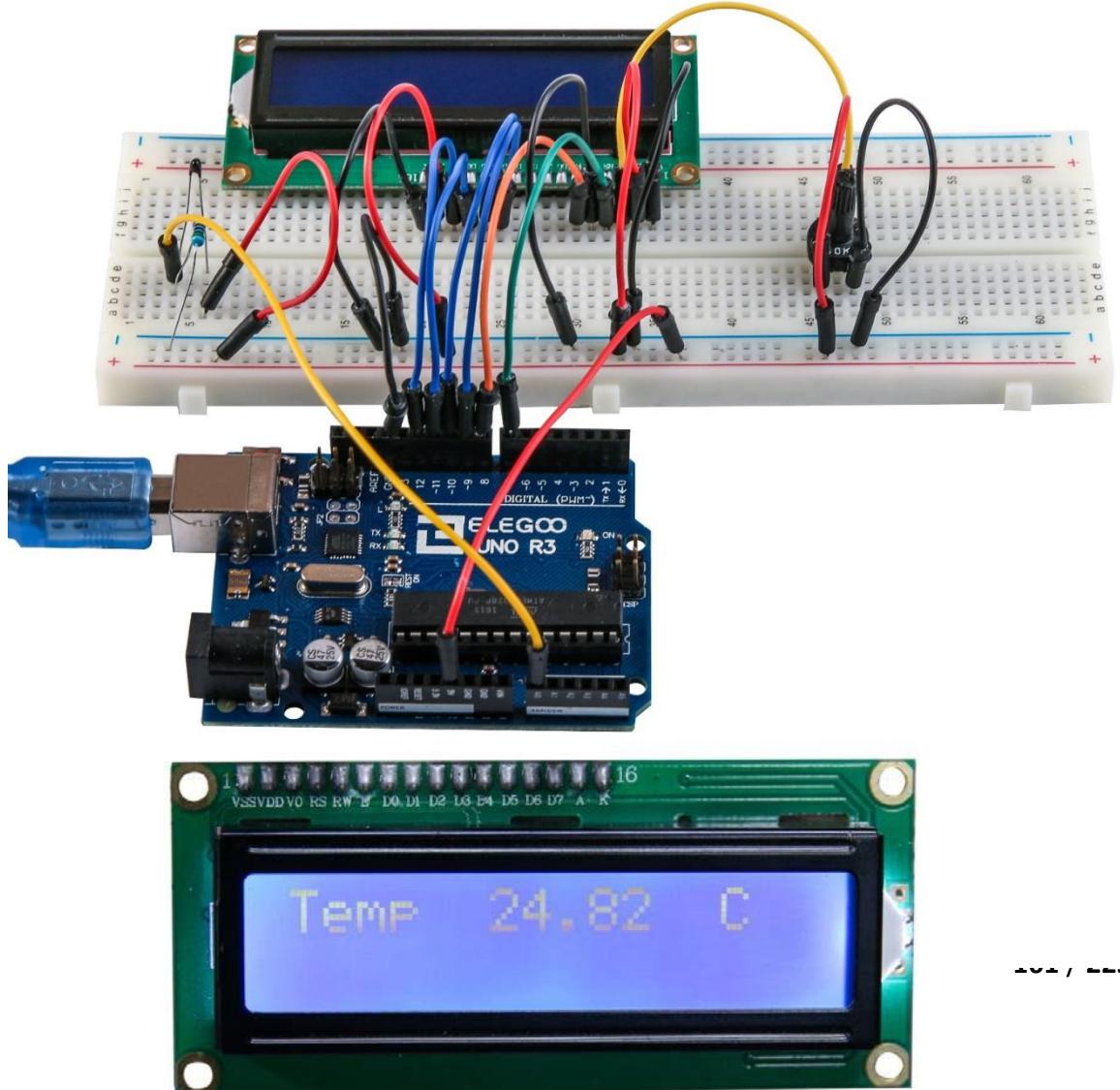
```
lcd.setCursor(0, 0);  
lcd.print("Temp C ");  
lcd.setCursor(6, 0);  
lcd.print(tempF);
```

Il commento che ti sembrerà strano Serve per ricordarti delle 16 colonne del display.

Tu puoi stampare una stringa di tale lunghezza con soli spazi dove andrà la lettura attuale.

Per riempire gli spazi vuoti, imposta il cursore sulla posizione in cui dovrebbe comparire la lettura e quindi stampa il valore.

### Foto d'esempio



## Lezione 24 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. Così, 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:

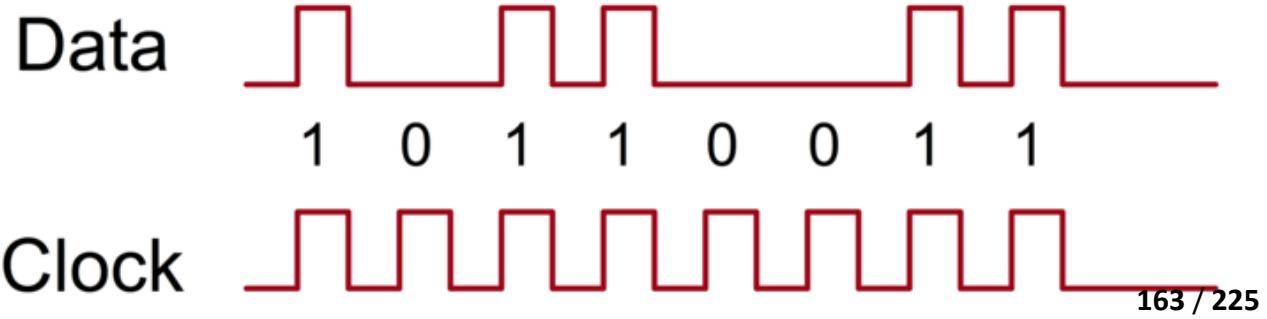
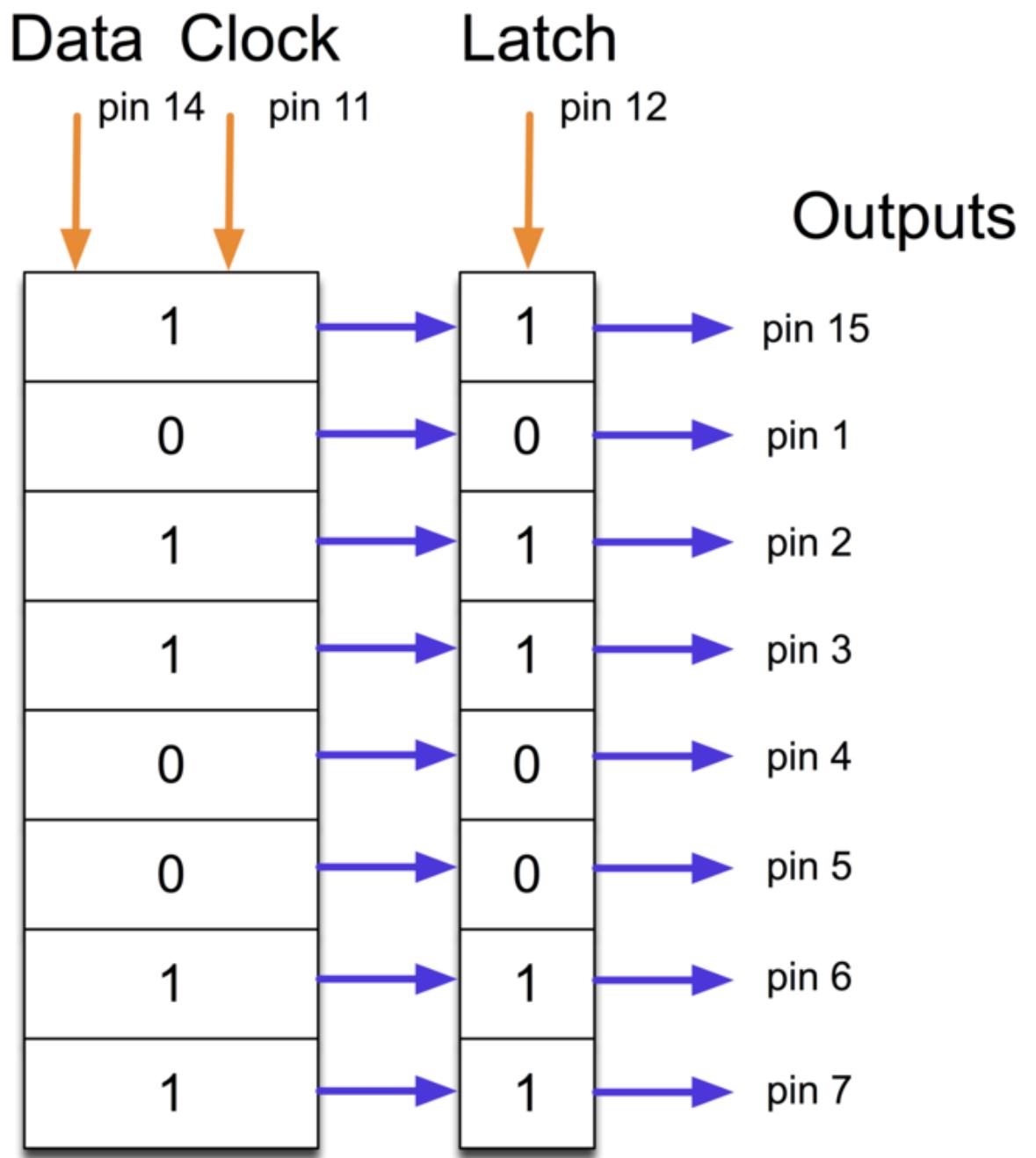
- (1) x Elegoo UNO R3
- (1) x breadboard con 830 punti di collegamento
- (8) x LED
- (8) x Resistenze da 220 ohm
- (1) x Chip 74hc595
- (14) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)



### 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.



Il pin del clock necessita di ricevere otto pulsazioni. Ogni pulsazione se il pin del dato 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

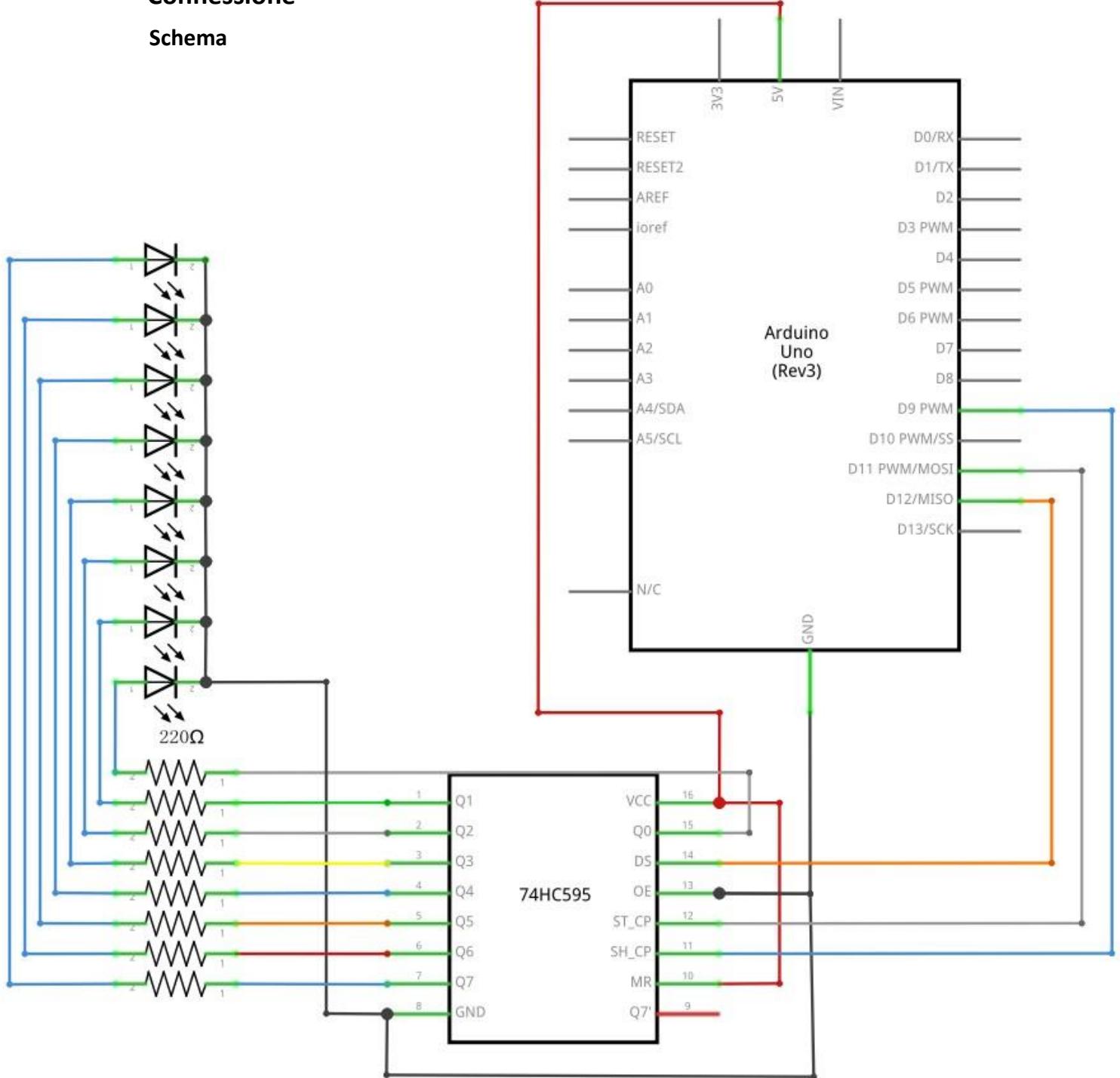
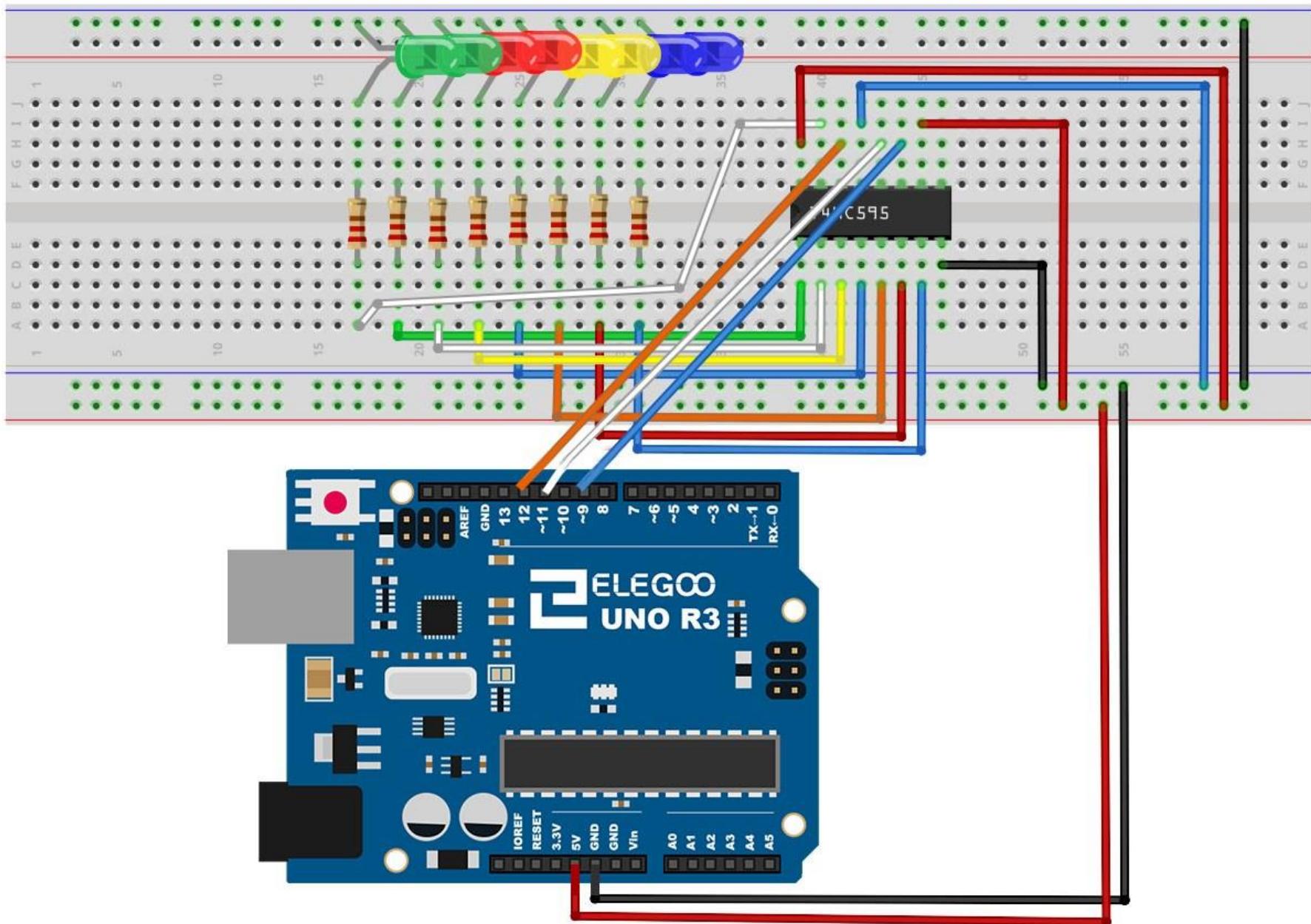


Diagramma di collegamento



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 utilizzata per mantenere il pattern di quali led sono attualmente accesi o spenti. 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 che 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++)
```

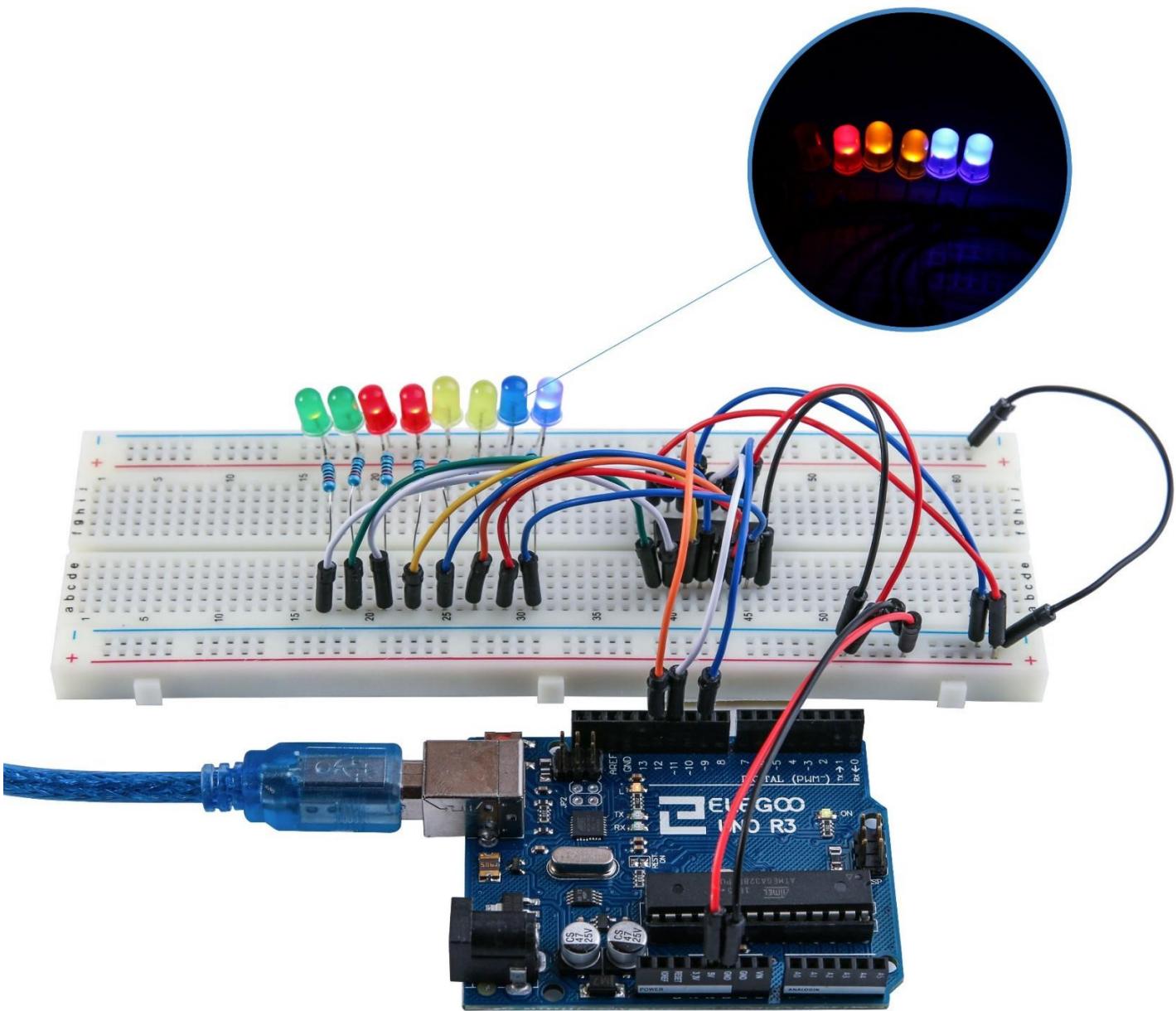
```
{  
    bitSet(leds, i);  
    updateShiftRegister();  
    delay(500);  
}  
}
```

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 dovrà semplicemente chiamare la funzione 'updateShiftRegister' per aggiornare i LED.

## Foto d'esempio



## Lezione 25 Il Monitor Seriale

### Introduzione

In questa lezione, utilizzeremo il progetto costruito nella lezione 24 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 24, perciò se non lo hai già fatto, torna ora alla lezione 24.

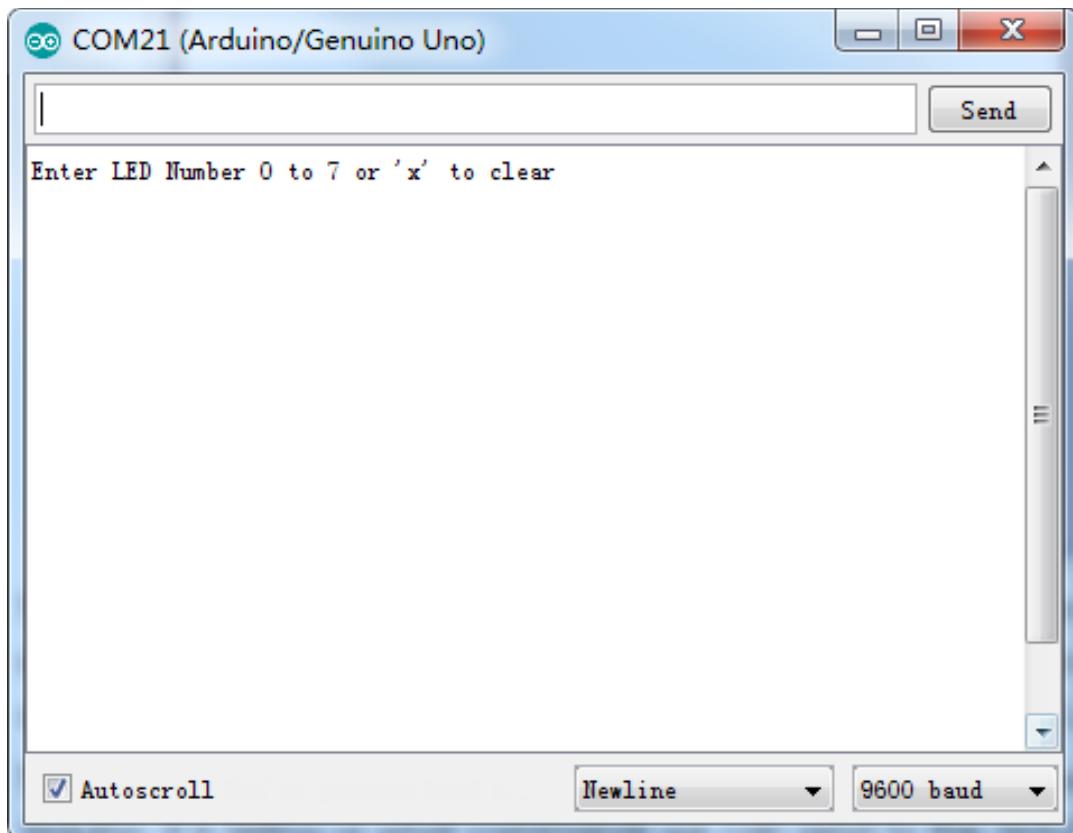
### Passaggi

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.

```
1 //www.elegoo.com
2 //2016.12.9
3
4 int latchPin = 11;
5 int clockPin = 9;
6 int dataPin = 12;
7
8 byte leds = 0;
9 void updateShiftRegister()
10 {
11     digitalWrite(latchPin, LOW);
12     shiftOut(dataPin, clockPin, LSBFIRST, leds);
13     digitalWrite(latchPin, HIGH);
14 }
15 void setup()
16 {
17     pinMode(latchPin, OUTPUT);
```

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.



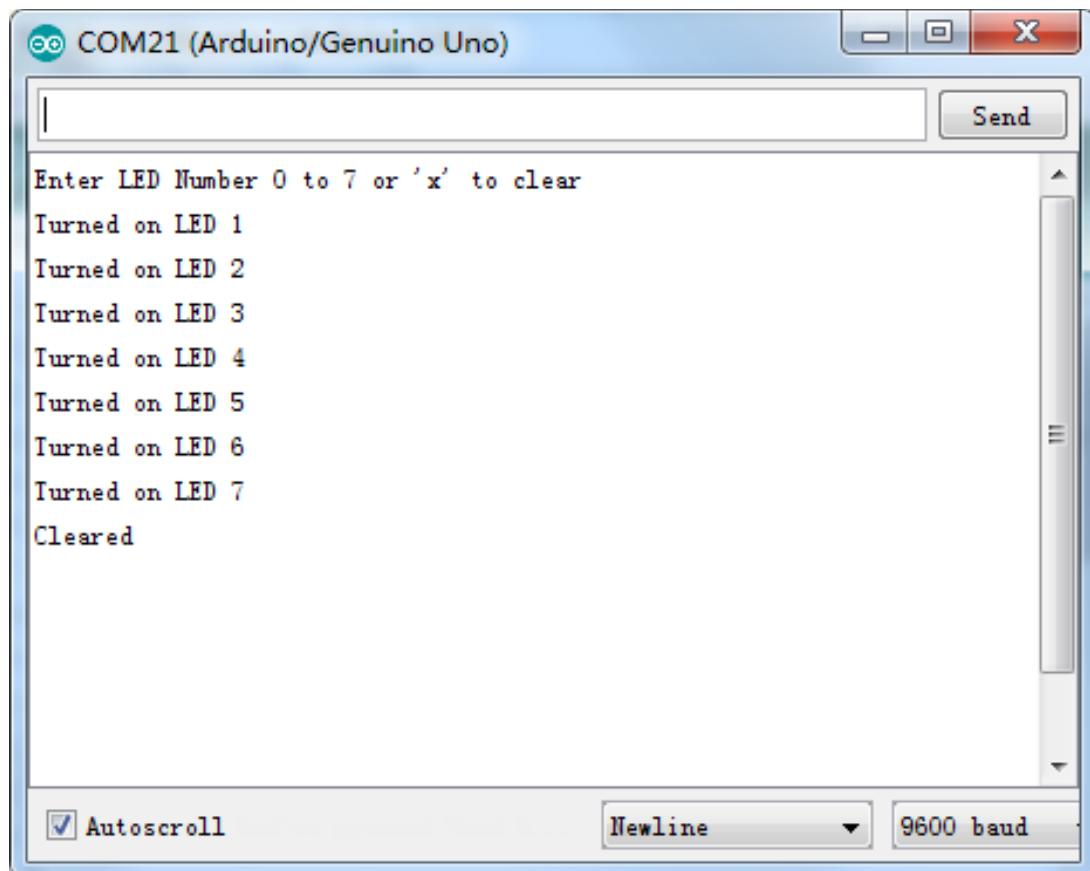
Questa finestra è chiamata Serial Monitor ed è parte del software IDE di Arduino. Il suo lavoro è permettere di inviare messaggi dal tuo computer alla scheda UNO (tramite USB) ma anche di ricevere messaggi da essa.

Il messaggio “Enter LED Number 0 to 7 or ‘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 superiore 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 dovrà cambiargli 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 automaticamente 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  $\geq$  e  $\leq$  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 stringhe 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 26 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)



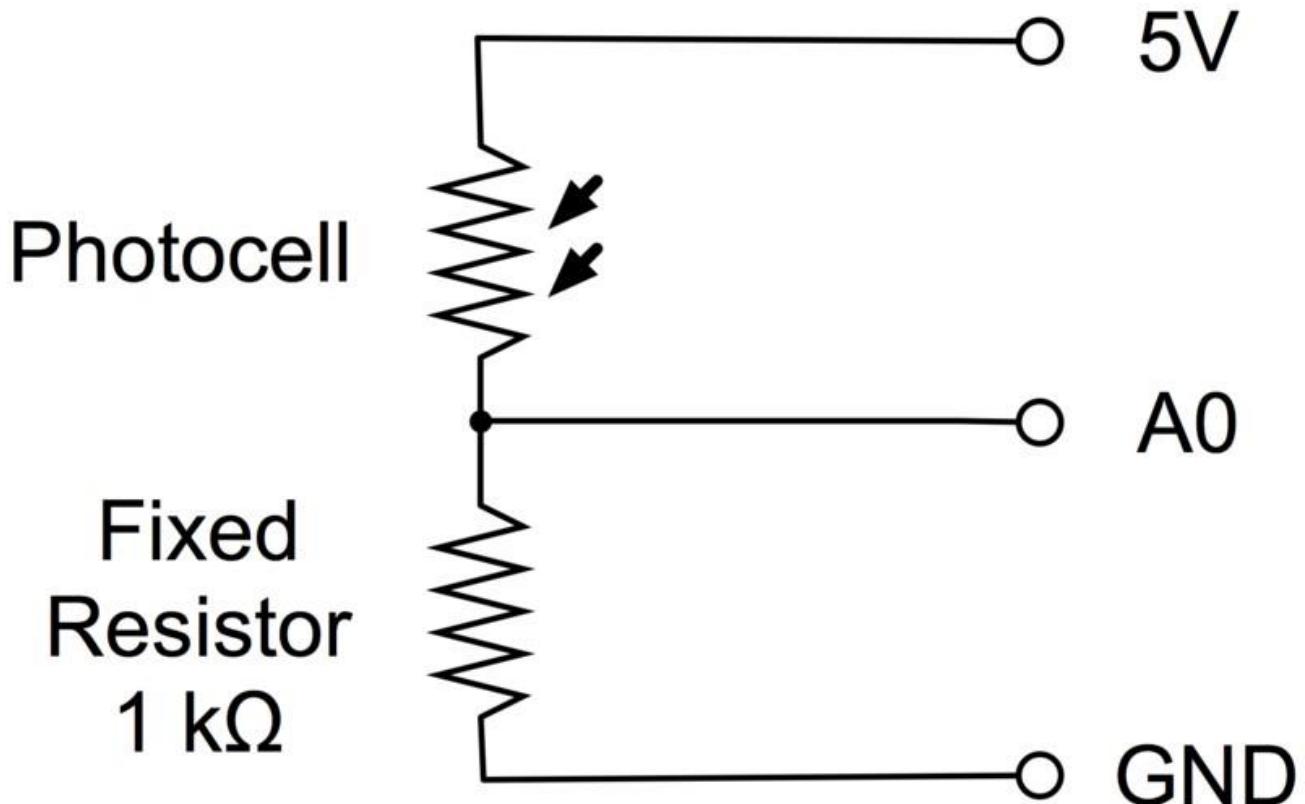
### Introduzione al componente

#### 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  $50\text{ k}\Omega$  al buio e circa  $500\ \Omega$  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 fotoresistore 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  $1\text{ k}\Omega$ , 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

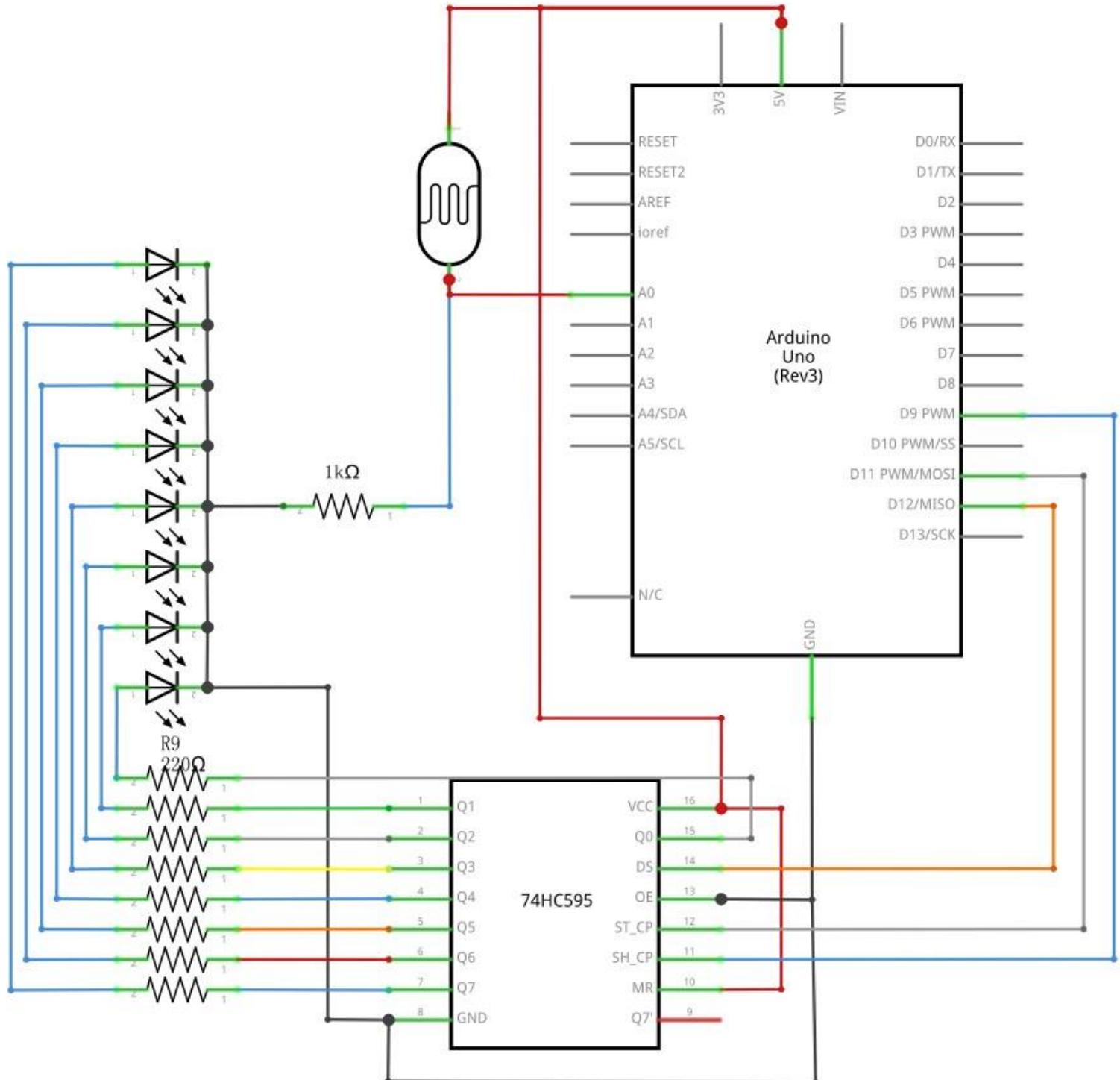
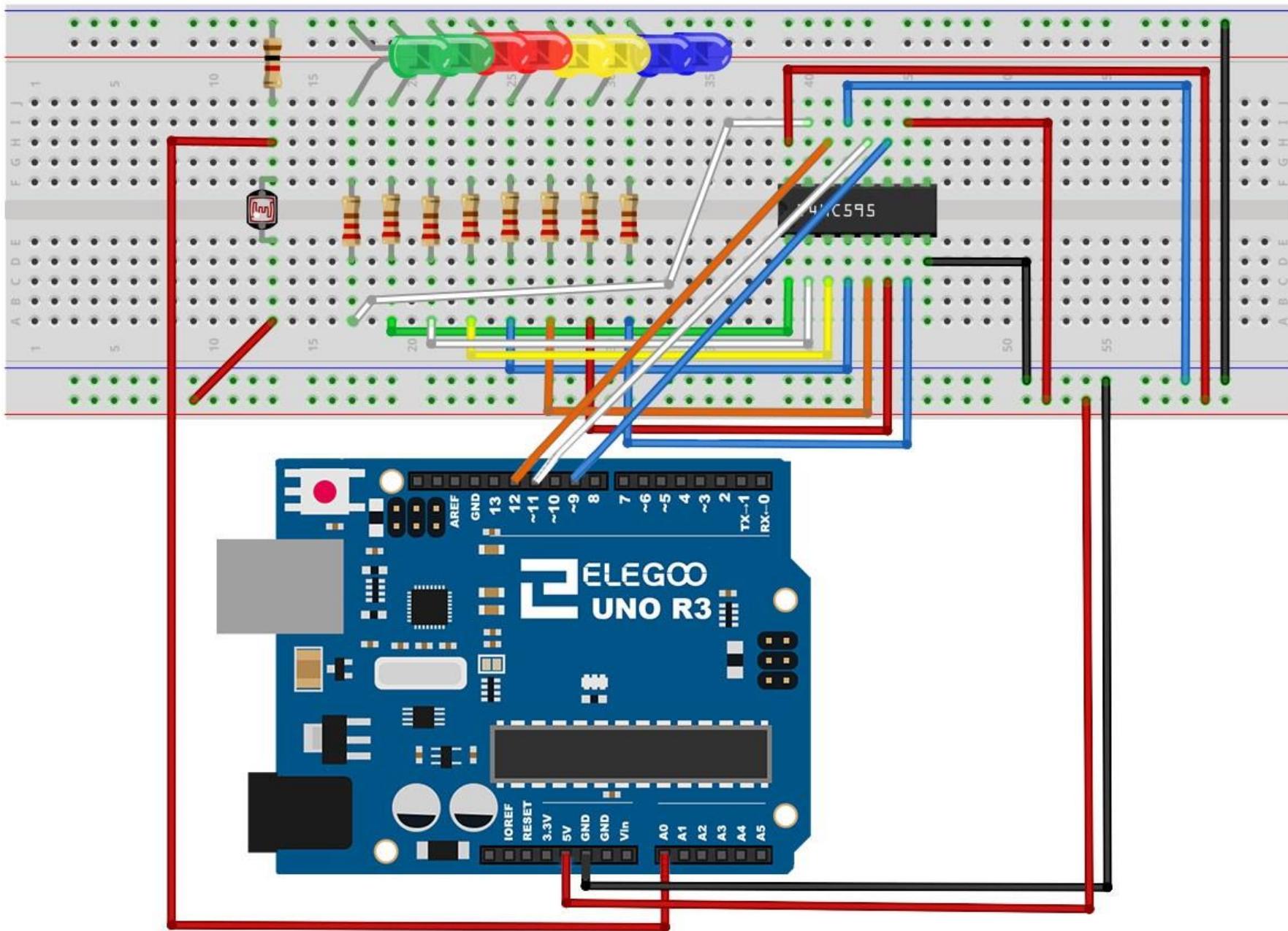


Diagramma di collegamento



## 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.

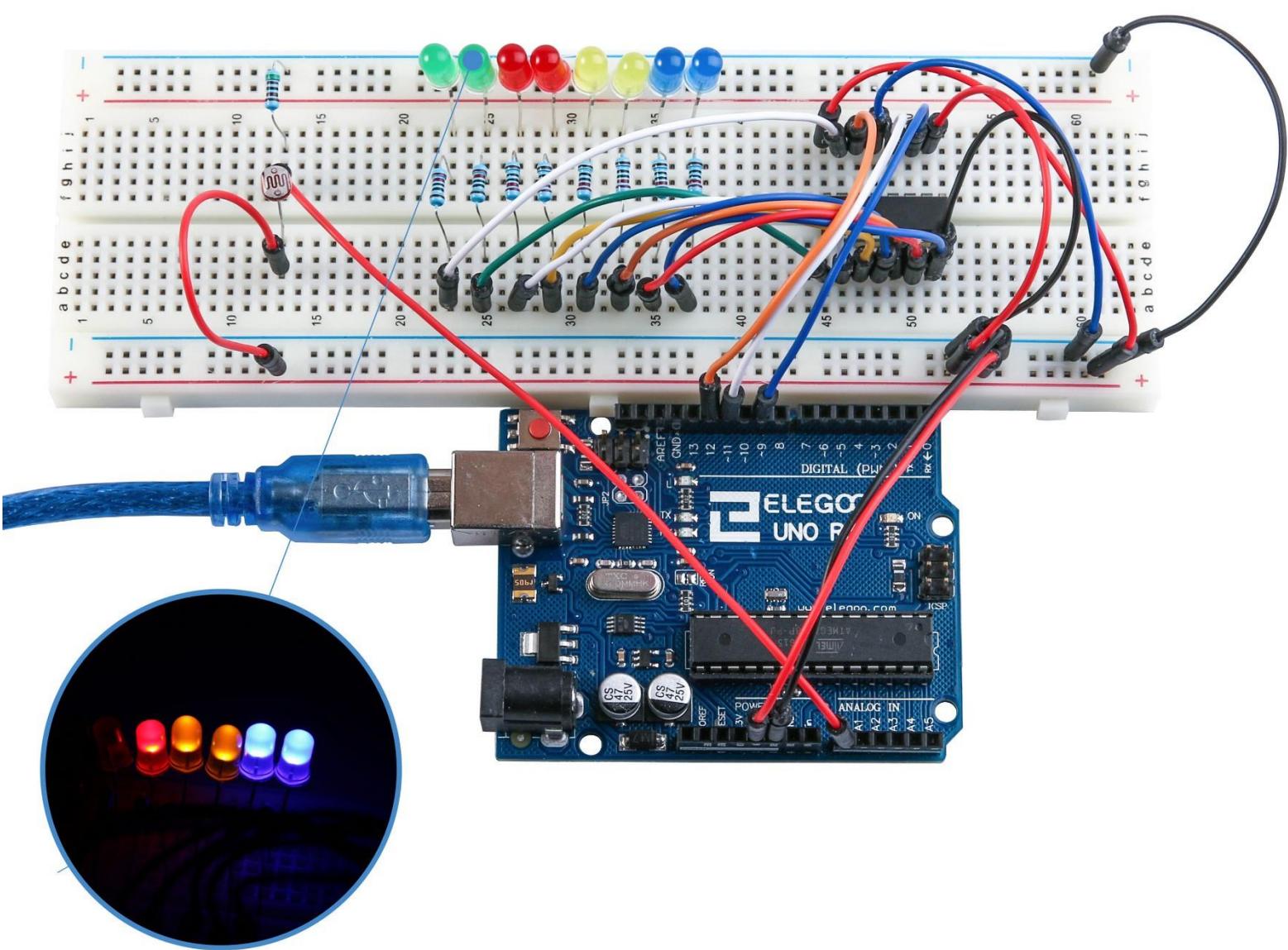
Se hai dubbio riguardo all’uploading del programma o se riscontri qualche errore durante il caricamento torna a vedere la lezione 2.

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



## Lezione 27 Display a segmenti e 74HC595

### Introduzione

Dopo aver imparato le lezioni 24, 25 e 26; utilizzeremo il Circuito integrato 74HC595 (registro a scorrimento) per controllare un display a segmenti. Tale display a segmenti ha la possibilità di visualizzare i numeri da 9 a 0.

### Componenti Richiesti:

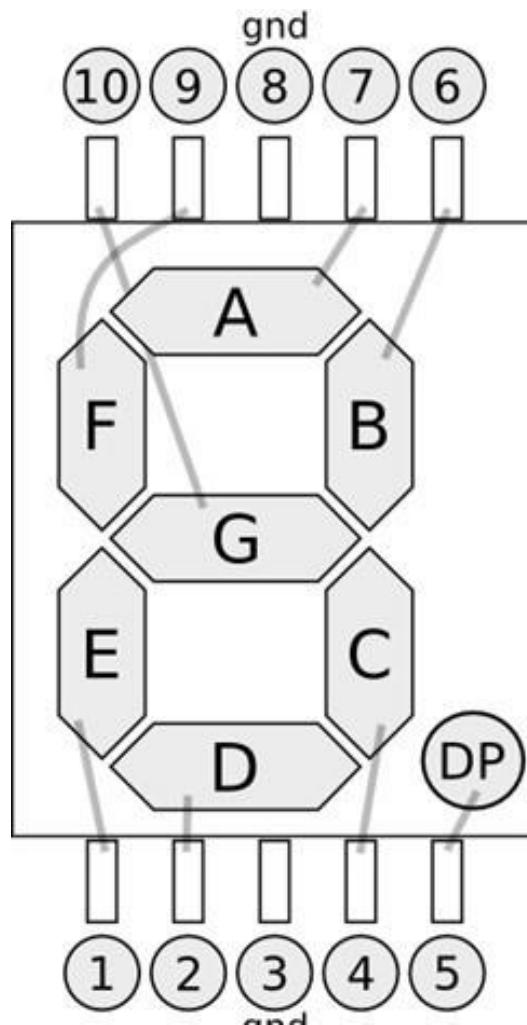
- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento
- (1) x Circuito integrato 74HC595
- (1) x Display ad 1 Cifra e 7-Segmenti
- (8) x Resistenza 220 ohm
- (26) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)



### Introduzione ai componenti

#### Display a sette segmenti:

Sotto qui puoi vedere il diagramma del display a sette segmenti.



Le dieci cifre da 0 a 9 corrispondono all'accensione di ogni segmento come segue (la seguente tabella viene applicata a display a sette segmenti con catodo in comune, nel caso tu voglia usare l'anodo in comune, dovrà sostituire tutti gli 1 con degli 0).

Display digital	dp	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

## Connessione Schema

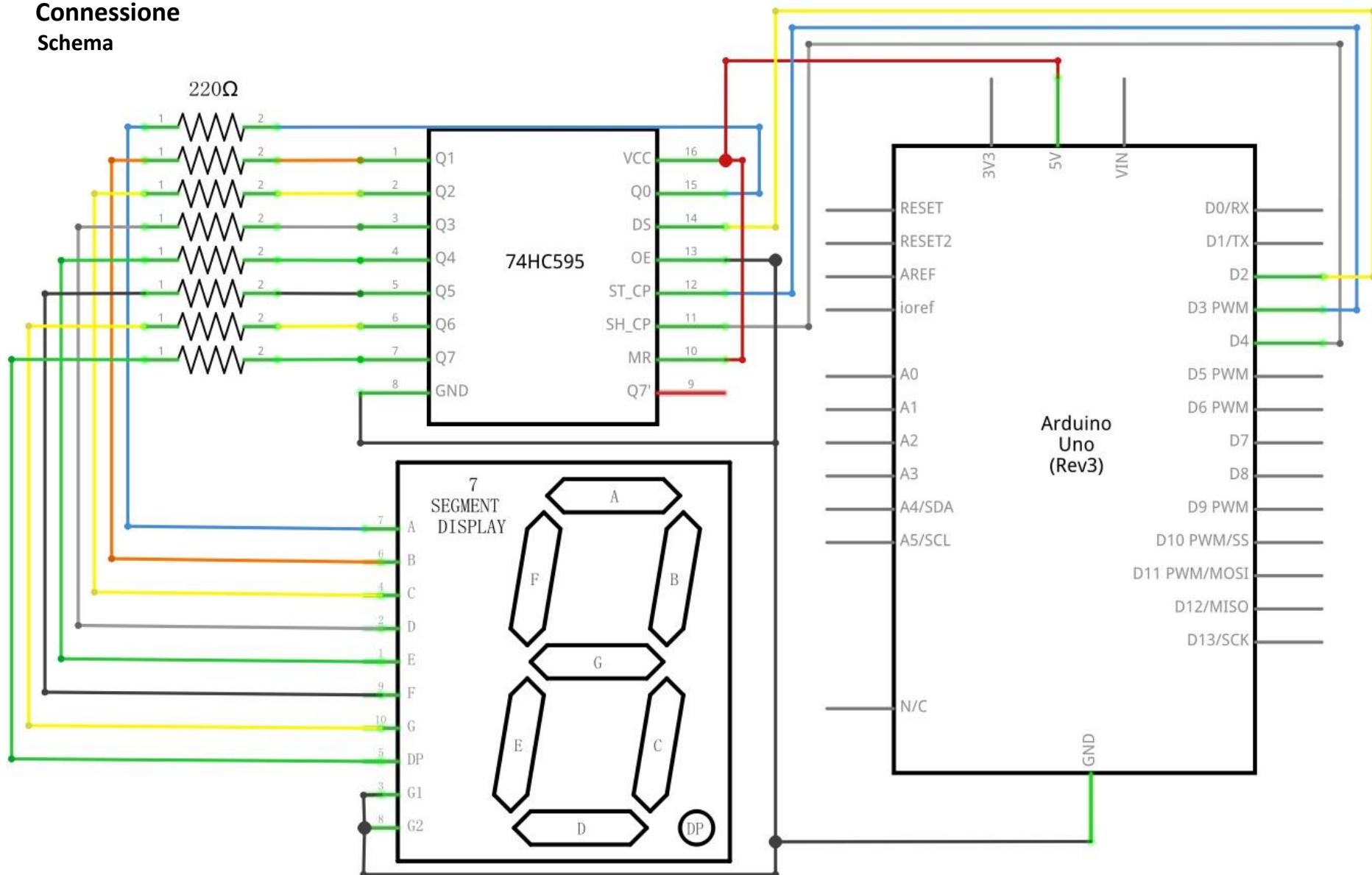
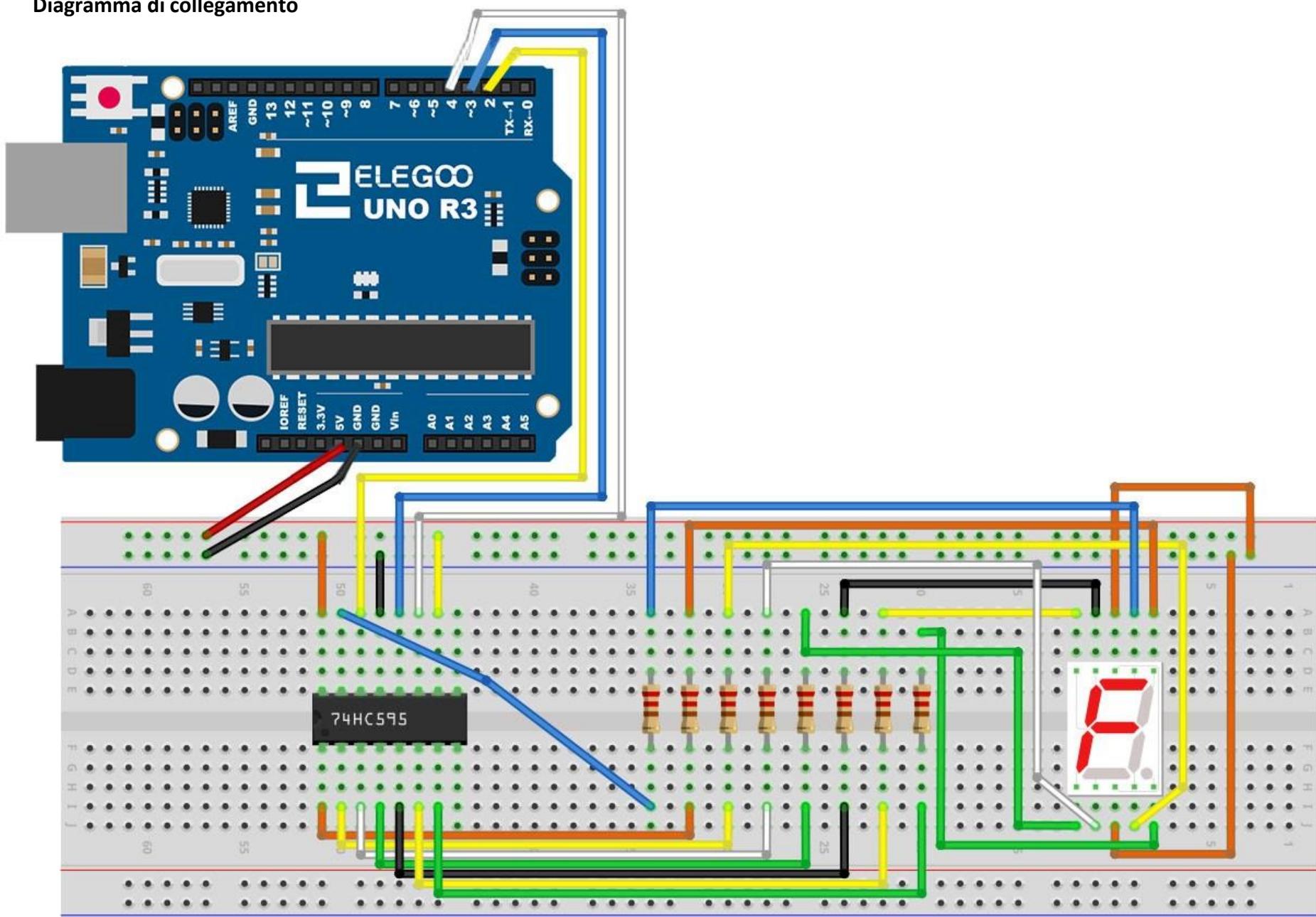


Diagramma di collegamento



La seguente tabella mostra la corrispondenza dei pin del display a sette segmenti con i pin del circuito integrato 74HC595.

74HC595 pin	Seven shows remarkable control pin (stroke)
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

Passaggio uno: connetti il circuito integrato 74HC595

Per prima cosa connetti il positivo e la messa a terra:

**VCC** (pin 16) e **MR** (pin 10) connesso ai 5V

**GND** (pin 8) e **OE** (pin 13) alla messa a terra

Connetti i pin **DS**, **ST\_CP** e **SH\_CP**:

**DS** (pin 14) connesso al pin 2 della scheda UNO R3 (La linea gialla nella figura sottostante)

**ST\_CP** (pin 12, latch pin) connesso al pin 3 della scheda UNO (La linea blu nella figura sottostante)

**SH\_CP** (pin 11, clock pin) connesso al pin 5 della scheda UNO R3 (La linea bianca nella figura sottostante)

Passaggio due: connetti il display a sette segmenti

Connetti i pin 3 e 8 del display a sette segmenti con il **GND** della scheda UNO (questo esempio utilizza il catodo in comune, se utilizzi l'anodo in comune, connetti i pin 3 e 8 del display a sette segmenti con il +5V della scheda UNO)

In accordo con la tabella qui sopra connetti i pin del circuito integrato 74HC595 Q0

~ Q7 con i corrispondenti pin (A ~ G e DP) del display a sette segmenti, e poi per

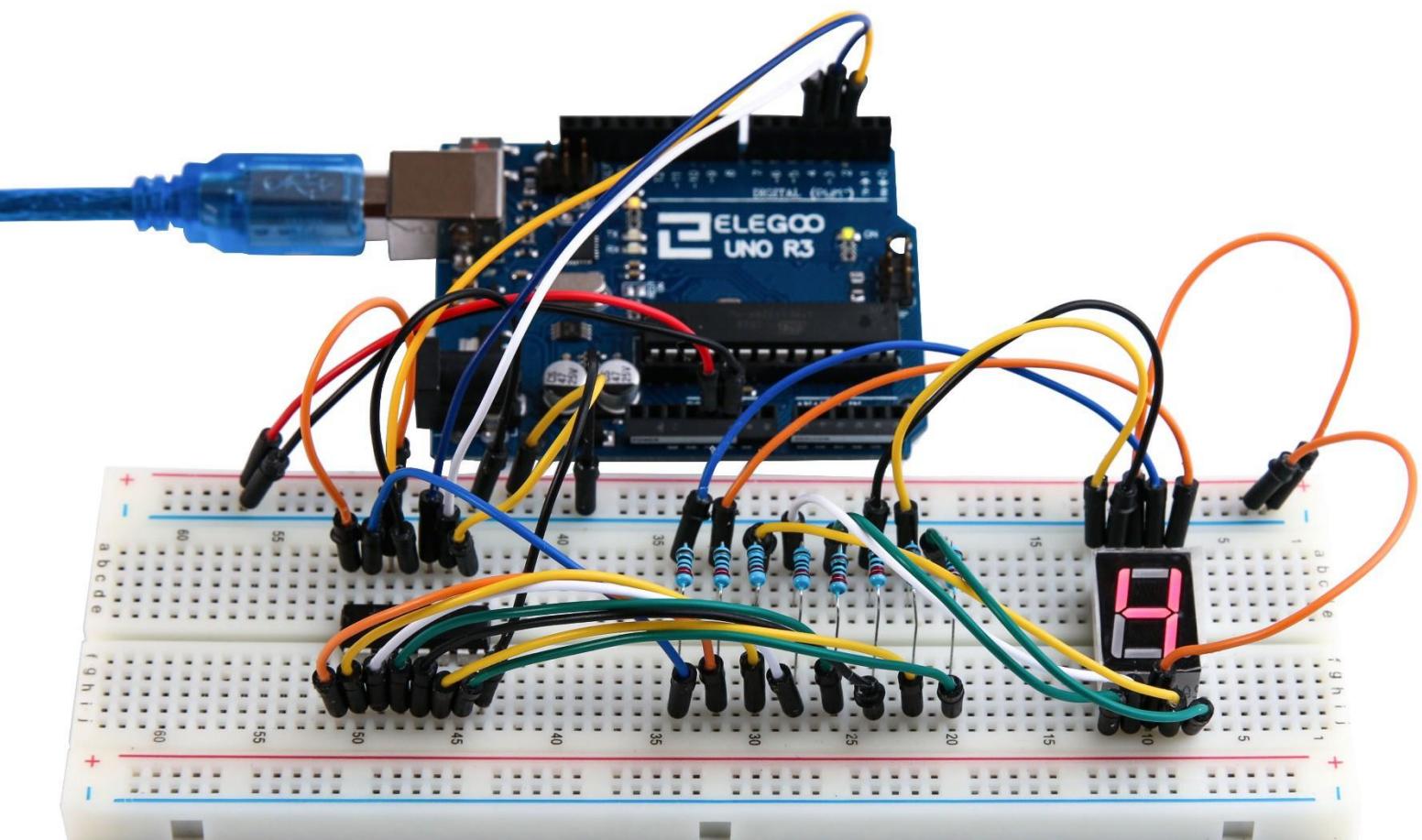
ogni piedino aggiungi in serie una resistenza da 220 ohm

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 27 74HC595 And Segment Display”, 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 28 Display a sette segmenti con quattro cifre

### Introduzione

In questa lezione imparerai come utilizzare un display a 7 segmenti con quattro cifre.

Quando si utilizza un display a 7 segmenti ed 1 cifra, nota che se utilizzi l'anodo in comune, esso viene collegato alla sorgente +5V, se invece si utilizza il catodo in comune, i pin del catodo in comune vanno connessi al pin GND.

Quando si utilizza un display a 7 segmenti e 4 cifre, il pin dell'anodo o catodo in comune è utilizzato per controllare qualche cifra viene visualizzata, perciò si può utilizzare solamente una cifra alla volta, il principio della persistenza della visione ti permetti di visualizzare tutti i numeri visualizzati in quanto la velocità di ciclo è talmente veloce che il suo intervallo è molto difficile da notare.

### Componenti Richiesti:

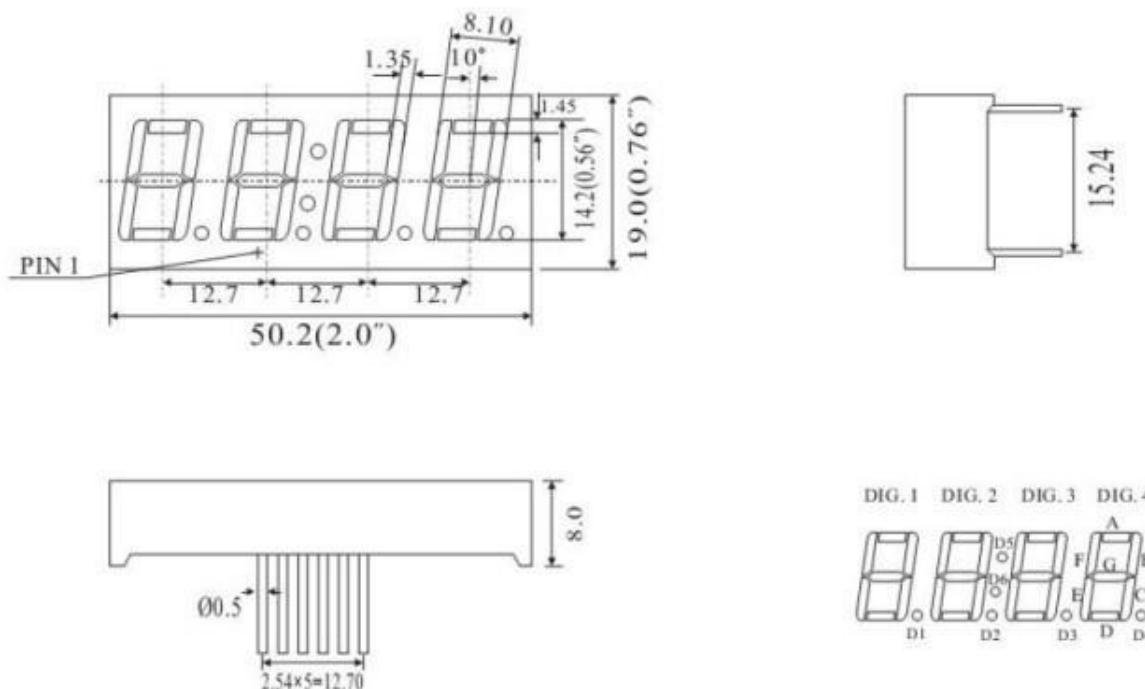
- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento
- (1) x Circuito integrato 74HC595
- (1) x Display a 4 Cifre e 7-Segmenti
- (4) x Resistenza 220 ohm
- (23) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)



**Introduzione ai componenti**  
**Display a sette segmenti e quattro cifre**

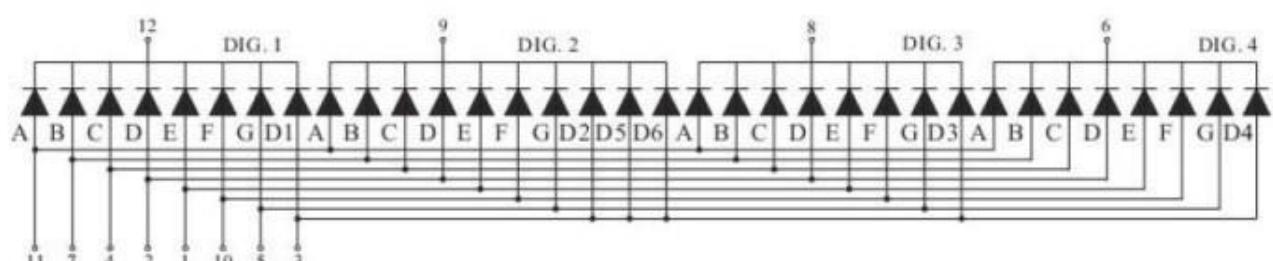
**Package Dimensions**

**CPS05643AB**

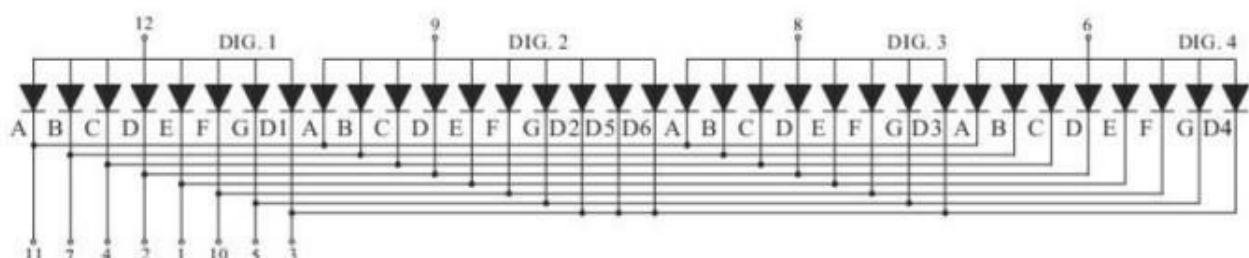


UNIT: MM(INCH) TOLERANCE:±0.25(0.01")

**Internal Circuit Diagram**

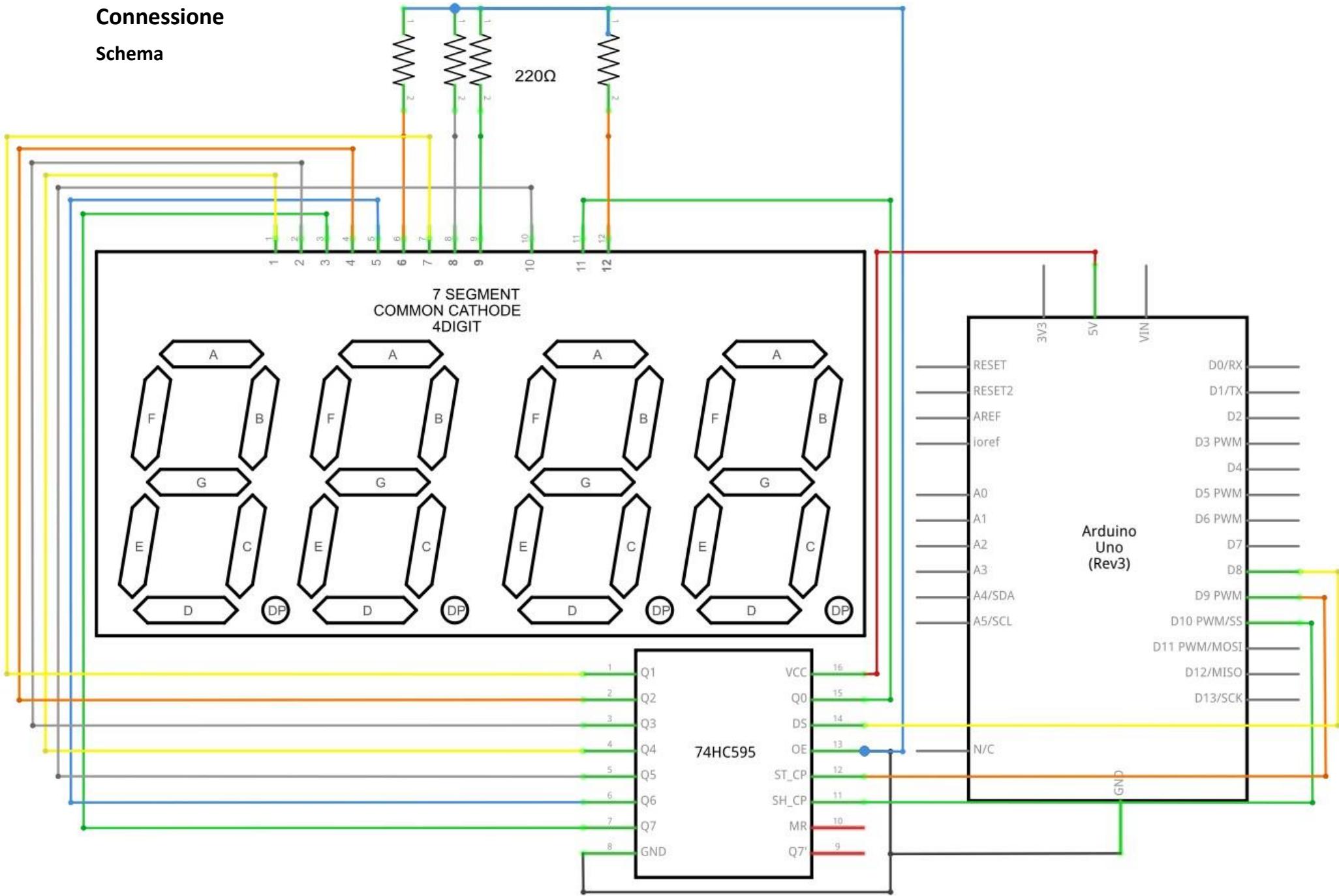


**5643A**

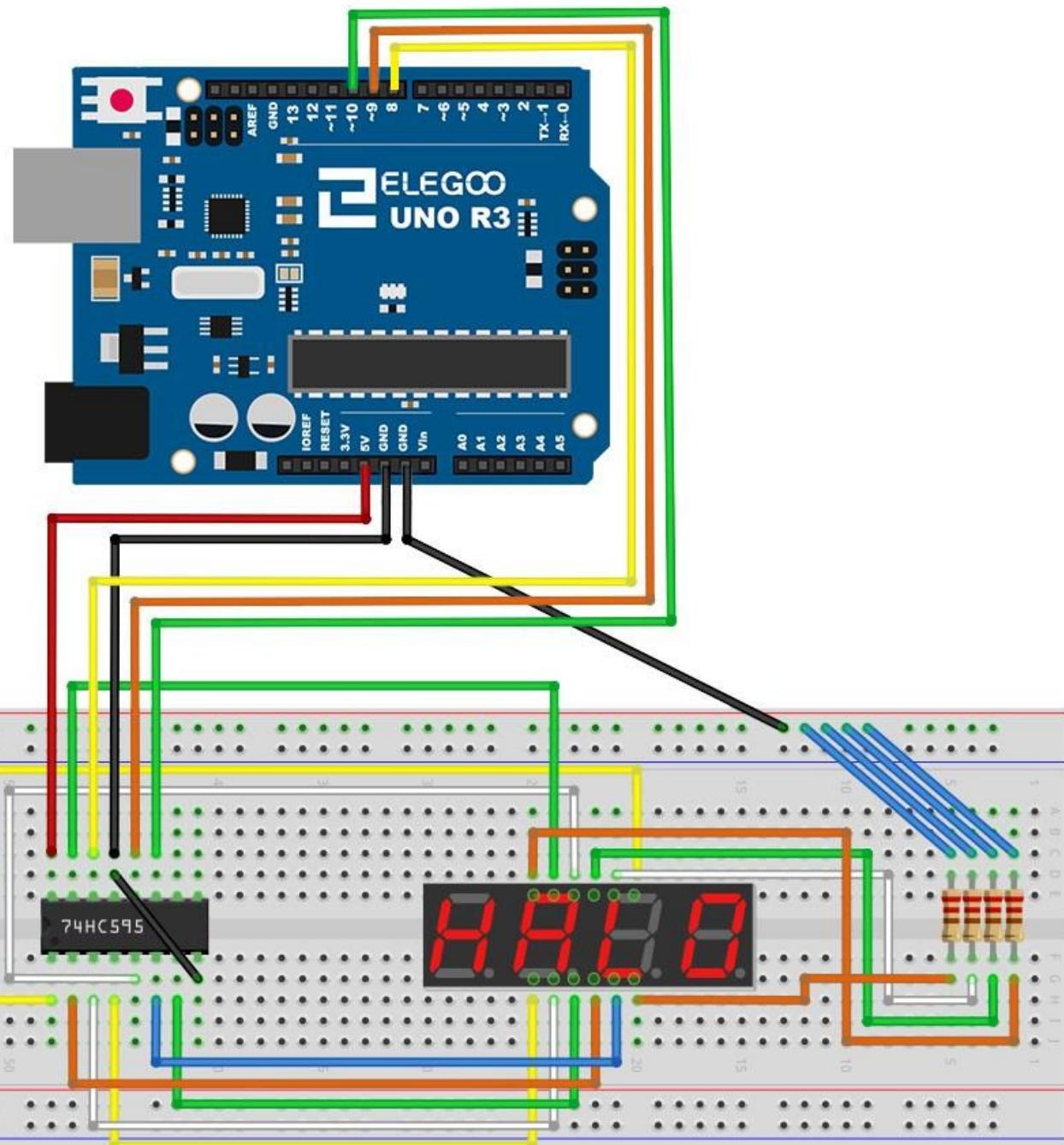


**5643B**

## Connessione Schema



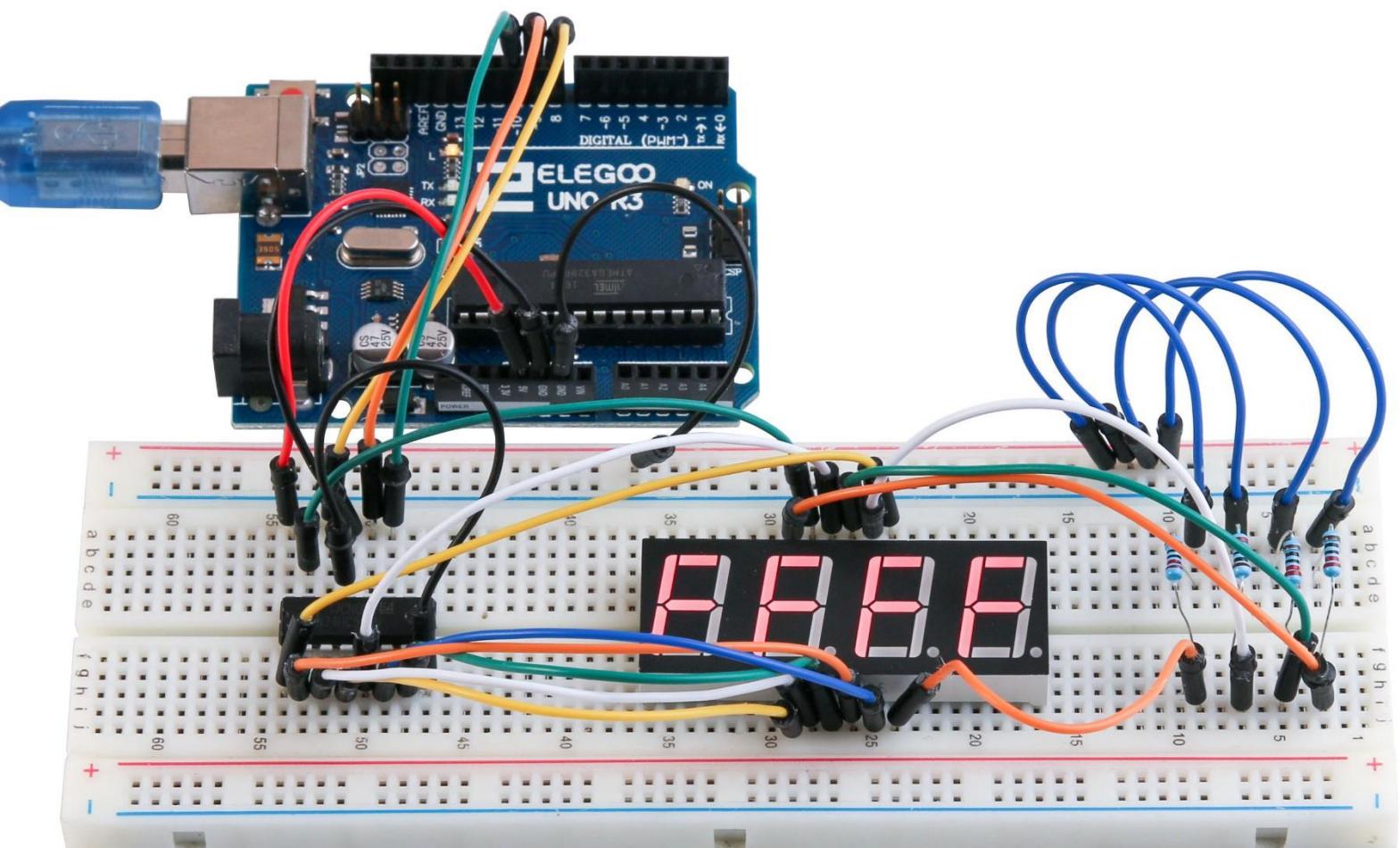
## Diagramma di collegamento



## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 28 Four Digital Seven Segment Display 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 29 Motore DC

## Introduzione

In questa lezione imparerai come controllare un piccolo motore DC utilizzando la tua scheda UNO R3 ed un transistor.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento
- (1) x Circuito integrato L293D
- (1) x Pala della ventola e motore 3-6 V
- (5) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)
- (1) x Modulo di alimentazione elettrica
- (1) x adattatore 9V1A

## Introduzione ai componenti

### Modulo di alimentazione elettrica

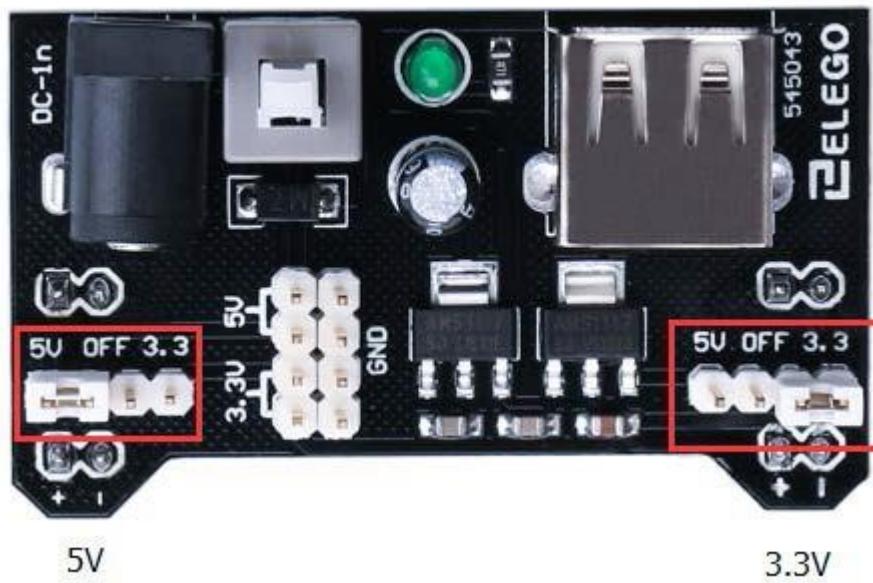
Il piccolo motore DC utilizza più energia di quella che la schedina UNO R3 può sopperire direttamente. Se provassimo a collegare il motore direttamente con il pin della scheda UNO c'è una alta possibilità di rovinare la scheda UNO. Per questo motivo utilizzeremo un modulo di alimentazione elettrica in modo da fornire la corretta alimentazione.



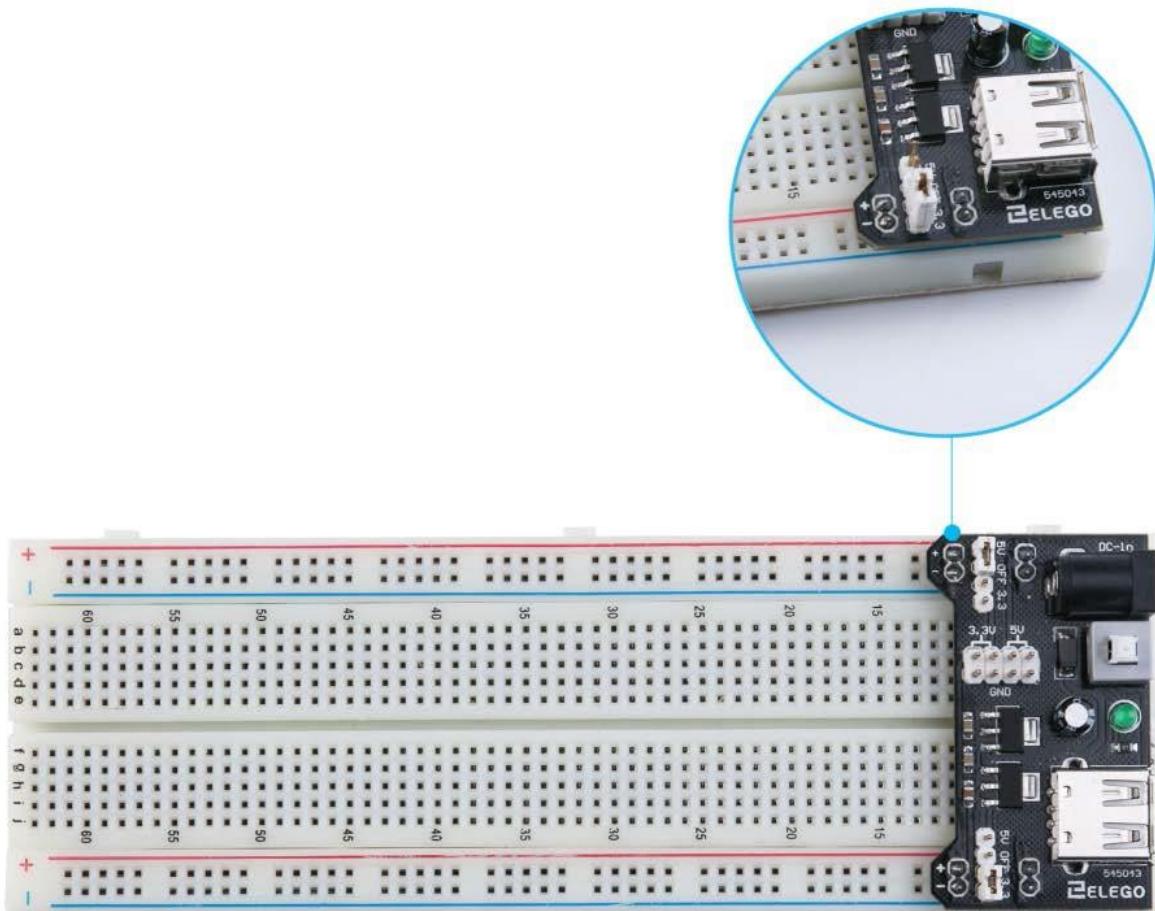
## Specifiche del prodotto

- Interruttore di blocco On/Off
- LED indicatore di accensione
- Voltaggio in Input: 6.5-9v (DC) tramite plug 5.5mm x 2.1mm
- Voltaggio in output: 3.3V/5v
- Corrente massima in output: 700 mA
- Linea di controllo output indipendente 0v, 3.3v, 5v con la breadboard
- Piedini di uscita per un comodo uso esterno
- Dimensioni: 2.1 in x 1.4 in
- Connettore USB incorporato per alimentare un dispositivo esterno.

## Impostare il voltaggio di output



Il voltaggio a sinistra e quello di destra possono essere configurati indipendentemente. Per selezionare il voltaggio di output, sposta il ponticello al pin corrispondente. Nota: il Led di accensione e le linee collegate alla breadboard non vengono alimentate se entrambi i ponticelli sono in posizione di "OFF".

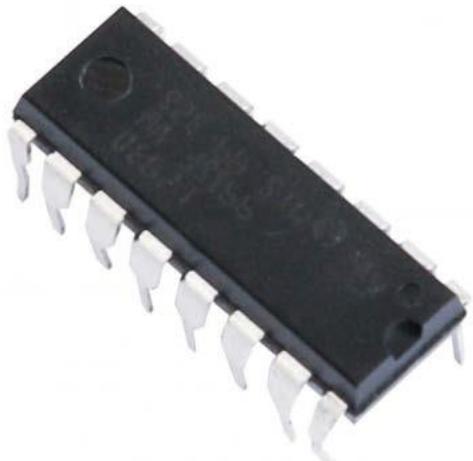


### **Nota importante:**

Fa attenzione ad allineare il modulo correttamente sulla breadboard. Il pin negativo (-) sul modulo corrisponde alla linea blu (-) sulla breadboard ed il pin positivo (+) corrisponde alla linea rossa (+). Sbagliare questi collegamenti può portare ad una inversione di corrente sul tuo progetto.

### **L293D**

Questo è un chip molto utile. Può controllare due motori indipendentemente. Noi useremo solamente metà delle potenzialità di questo chip in questa lezione. La maggior parte dei pin sul lato di destra del chip sono a disposizione per il secondo motore.



## Specifiche del prodotto:

- Componenti L293 e L293D Prodotti ora da Texas Instruments
- Banda di Voltaggio in ingresso molto larga: 4.5 V - 36 V
- Logica di input separata
- Protezione ESD interna
- Spegnimento Termico
- input Immuni agli alti rumori
- Funzionalità simili a SGS L293 e SGS L293D
- Corrente in Output: 1 A per canale (600 mA per L293D)
- Picchi di corrente in output: 2 A per canale (1.2 A per L293D)
- Morsettiera per diodi in uscita, utilizzata per soppressione induttiva dei transistori.



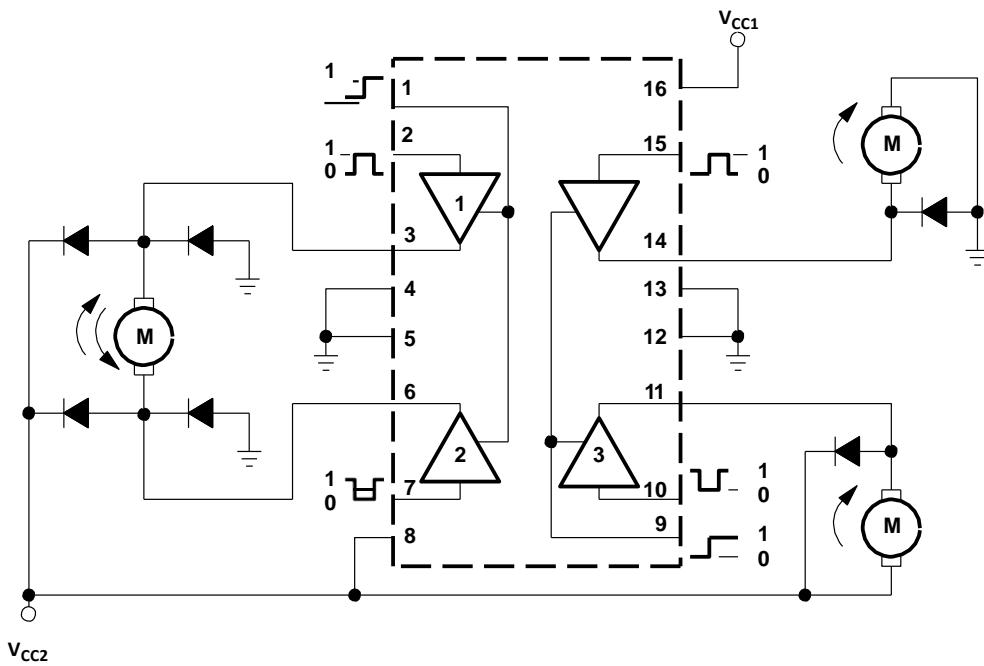
## Informazioni di descrizione/Comandi

L293 e L293D sono dispositivi a half-H quadrupli ad alta corrente.

L293 è costruito per fornire una corrente guida bidirezionale con un voltaggio compreso tra 4.5V e 36V e corrente superiore a 1A. L293D è costruito per fornire una corrente guida bidirezionale con un voltaggio compreso tra 4.5V e 36V e corrente superiori a 600mA.

Entrambi i dispositivi sono strutturati per comandare carichi indutttivi tipo solenoidi, relè, dc e motori passo-passo, così come altri carichi ad alta-corrente/alto-voltaggio in applicazioni con correnti di alimentazione positive. Tutti gli input sono compatibili con la logica dei circuiti integrati. Ogni uscita è un circuito di pilotaggio completo, con un transistore Darlington e una pseudo- fonte Darlington. Gli attuatori sono abilitati in coppia, si possono attivare gli attuatori 1 e due con il comando 1,2EN e gli attuatori 3 e 4 con il comando 3,4EN. Quando il comando di abilitazione è HIGH gli attuatori associati sono attivi, e la loro uscita è attiva, in fase con i loro input. Quando il comando di abilitazione è LOW, i rispettivi attuatori sono disabilitati, quindi il loro output è su off, in uno stato di alta impedenza. Con i giusti dati in ingresso, ogni paio di attuatori forma un full-H (detto anche ponte), un comando reversibile adatto a comandare solenoidi o per applicazioni con i motori.

## Diagramma del blocco



Essendo sono stufi di diagrammi di piedinatura indecifrabiili all'interno schede, ti abbiamo disegnato uno schema adatto che pensiamo posa fornirti informazioni più pertinenti.

Ci son o 3 cavi connessi con Arduino, due connessi al motore ed uno connesso alla batteria.

### L293D

M1 PWM	1	16	Battery +ve
M1 direction 0/1	2	15	M2 direction 0/1
M1 +ve	3	14	M2 +ve
GND	4	13	GND
GND	5	12	GND
M1 -ve	6	11	M2 -ve
M1 direction 1/0	7	10	M2 direction 1/0
Battery +ve	8	9	M2 PWM

#### Motor 1

#### Motor 2

Per utilizzare questi pin:

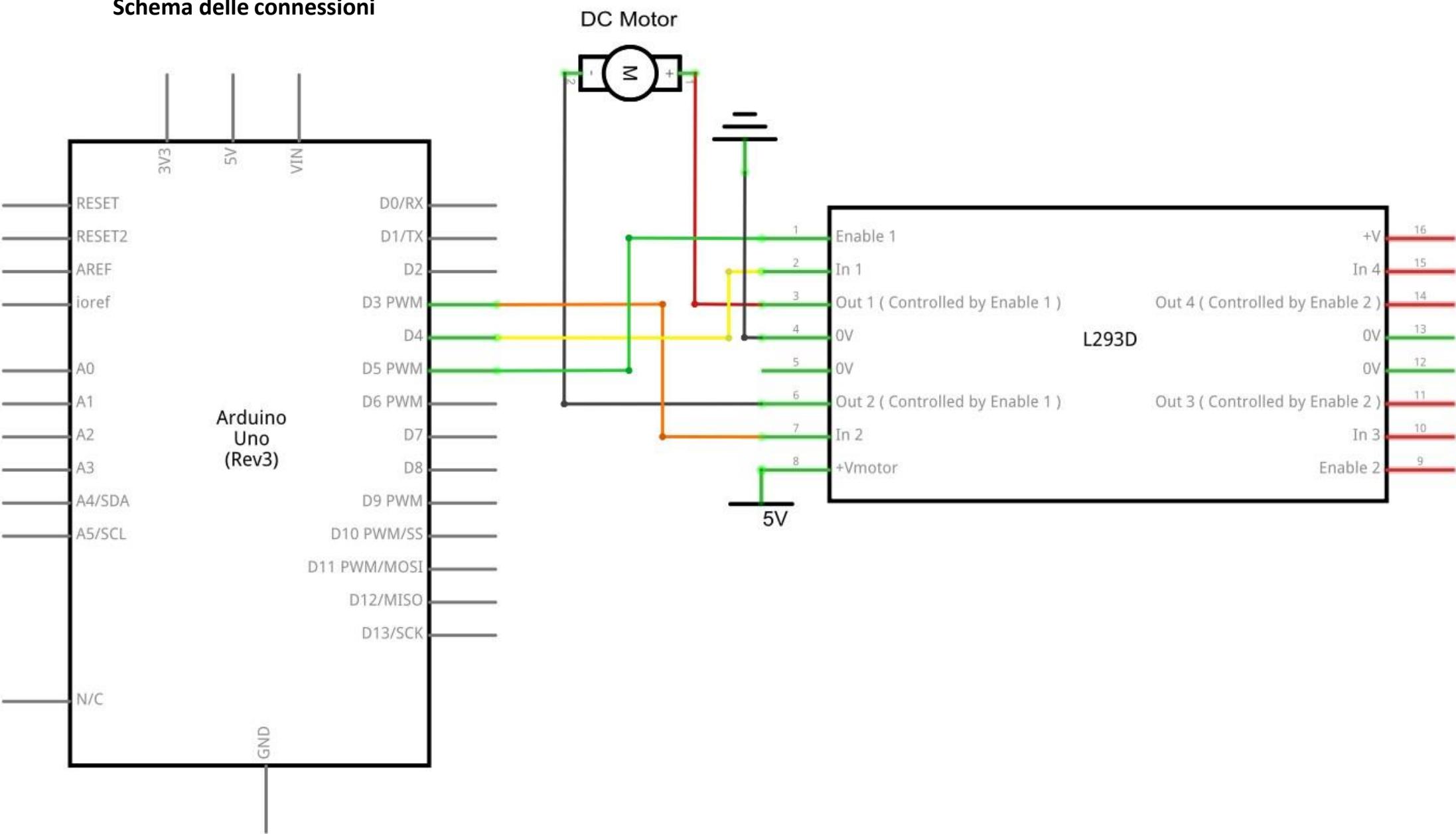
Il lato sinistro si occupa per primo motore, il lato alla destra si occupa del secondo. E puoi utilizzare questo circuito con un solo motore connesso.

## **Connessione con Arduino**

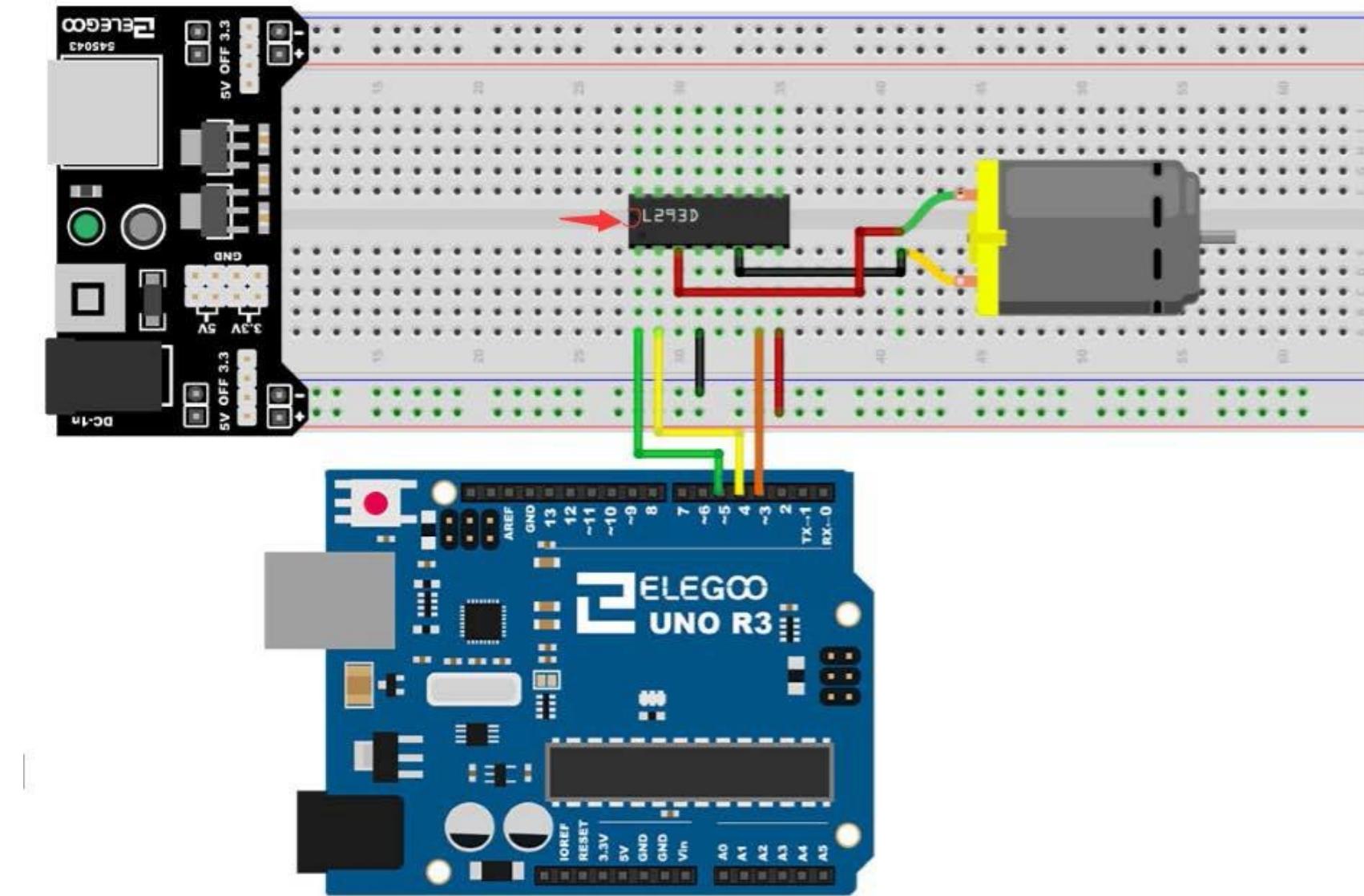
M1 PWM – Connotti questo ad un pin PWM della scheda Arduino. Essi sono etichettati il pin 5 è un esempio. Utilizza in uscita un valore intero compreso tra 0 e 255, 0 indicherà spento, 128 mezza velocità e 255 massima velocità.

M1 direction 0/1 e M1 direction 1/0 – Connotti questi due a due pin digitali su Arduino. Uno dei due pin dovrà essere impostato su HIGH e l'altro su LOW, in questo modo il motore girerà in una direzione. Invertendo le uscite a LOW la prima e HIGH la seconda il motore girerà nella direzione opposta.

## Schema delle connessioni

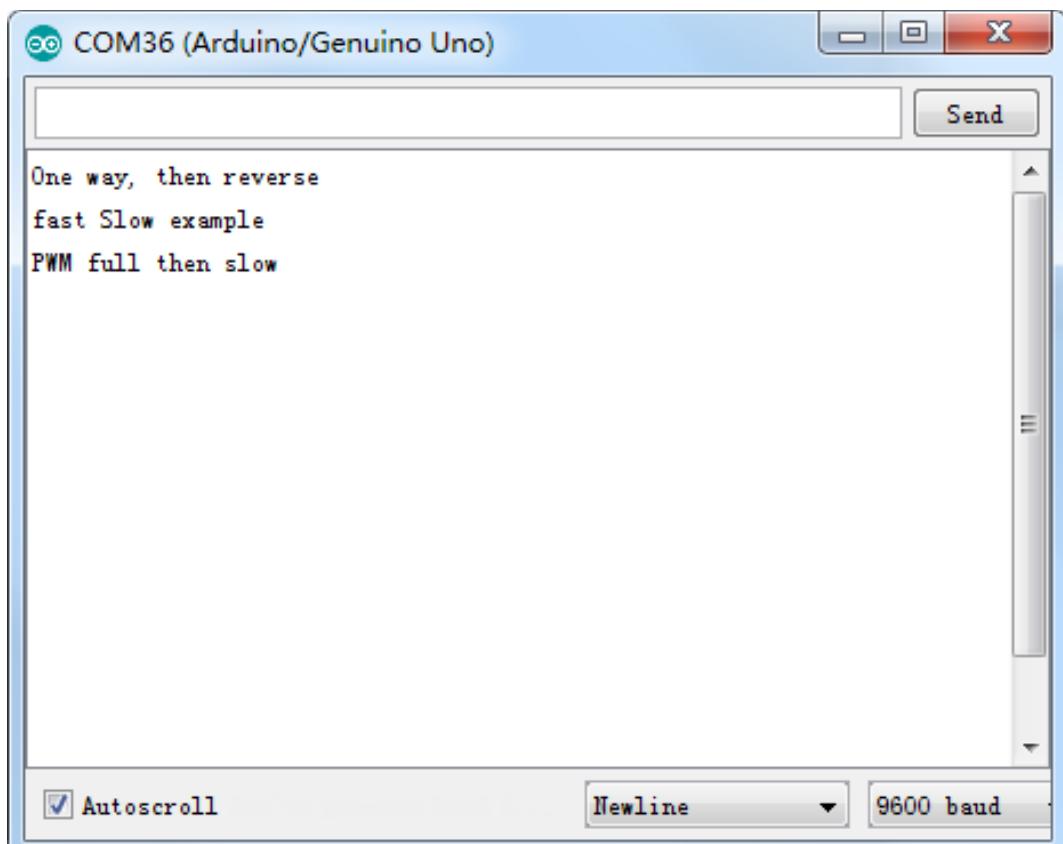


Schema di collegamento



Il codice qui sotto non utilizza una sorgente di energia separata (esempio una batteria), al suo posto utilizza l'uscita 5V proveniente da Arduino. Nota. Questo potrebbe essere rischioso senza il controllo del circuito L293D.

Non dovresti mai collegare un motore direttamente ad Arduino, perché quando spegni un motore si forma una retroazione elettrica. Con un piccolo motore questo potrebbe danneggiare l'Arduino, con un grande motore c'è il rischio che la scheda faccia scintille e si incendi.



## Codice

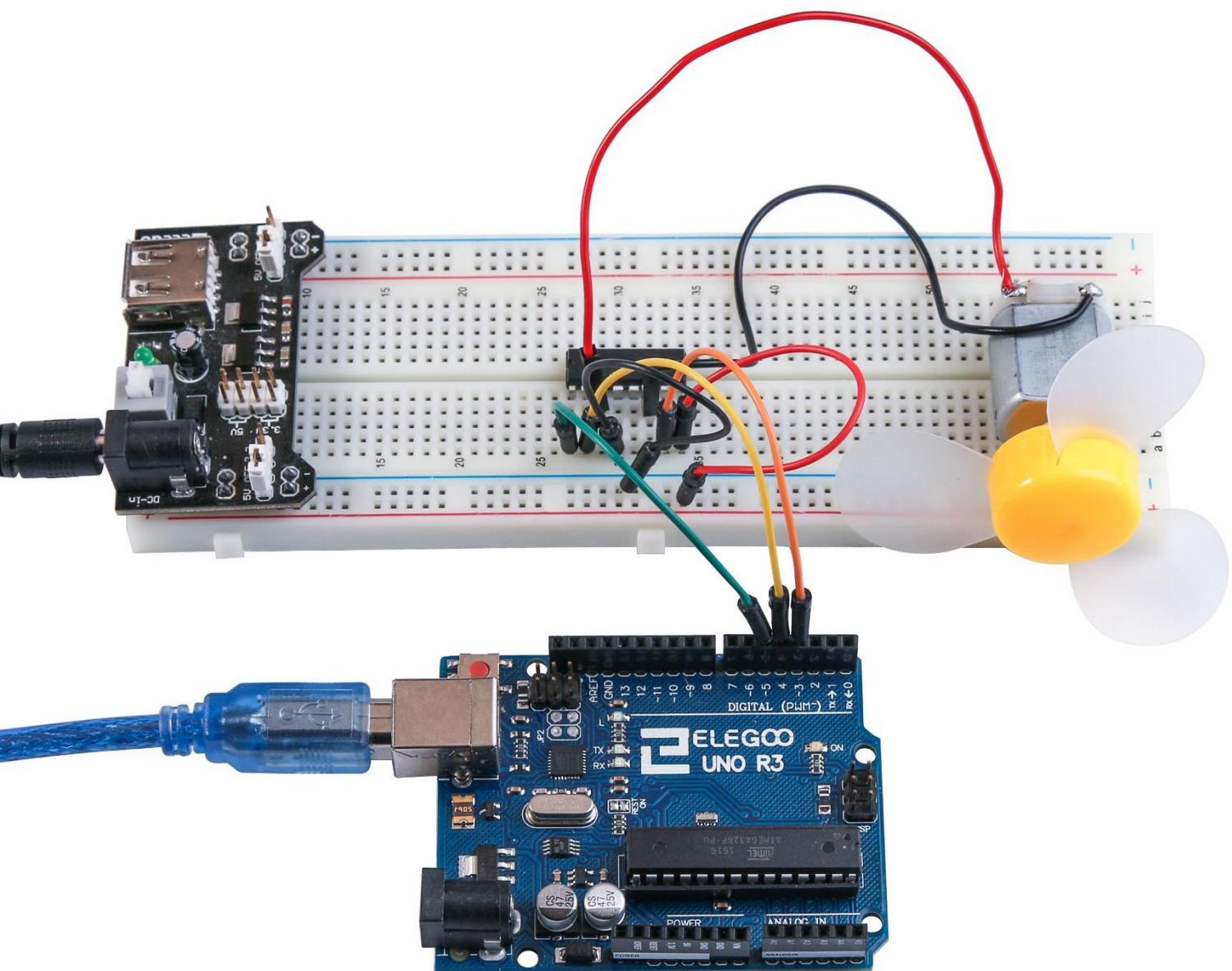
Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 29 DC Motors” clicca su UPLOAD per caricare il programma.

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

Dopo che si è caricato il programma, accendi tutto tramite gli interruttori. Il motore inizierà a girare leggermente in senso orario e antiorario per 5 volte. Poi, inizierà a girare più velocemente in senso orario. Dopo una piccola pausa, lo farà in senso antiorario. Poi la scheda di controllo invierà un segnale PWM per comandare il motore che diminuirà la sua velocità al minimo e la aumenterà poi al massimo.

Alla fine si stopperà per circa 10s dopodiché ricomincerà da capo.

## Foto d'esempio



# Lezione 30 Relè

## Introduzione

In questa lezione imparerai ad utilizzare un relè.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento
- (1) x Pala della ventola e motore 3-6 V
- (1) x Circuito Integrato L293D
- (1) x 5v Relè
- (1) x Modulo di alimentazione supplementare
- (1) x Adattatore 9V 1A
- (8) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)



## Introduzione ai componenti

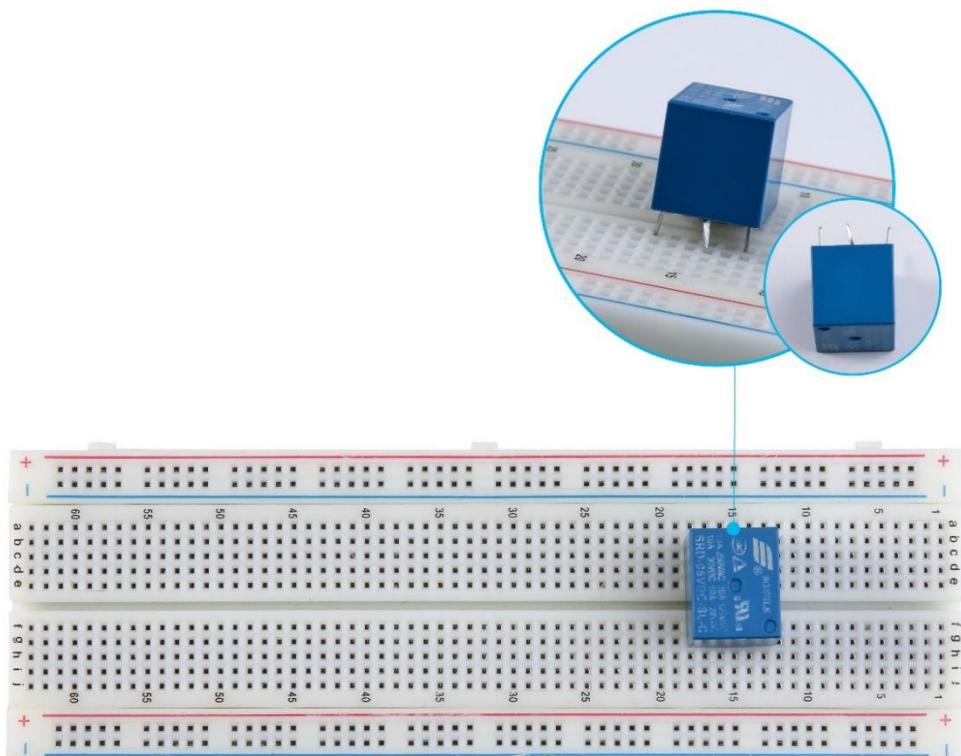
### Relè:

Il relè è un interruttore attivato elettricamente. Molti relè utilizzano un elettromagnete che controlla un interruttore, ma altri utilizzano il principio del relè a stato solido. I relè sono utilizzati quando è necessario controllare un circuito con un segnale a bassa potenza (con un isolamento elettrico completo tra il circuito controllore ed il circuito controllato) oppure quando più circuiti devono essere controllati da un singolo segnale. Il primo relè venne usato come amplificatore nei circuiti del telegrafo a lunga distanza. Essi ripetevano il segnale ricevuto da un circuito e lo ritrasmettevano ad un altro. I relè vennero usati estensivamente negli scambi telefonici e nei primi computer per svolgere operazioni logiche.

I relè che possono gestire l'alta potenza richiesta per controllare direttamente un motore elettrico o altri carichi vengono chiamati contattori. I relè a stato solido controllano circuiti senza parti mobili, utilizzando dispositivi con semiconduttori per attivare gli interruttori. Relè con caratteristiche operative calibrate e spesso multiple bobine vengono utilizzati per proteggere i circuiti da guasti e sovraccarichi. Nei sistemi elettrici moderni queste funzioni vengono eseguite da strumenti digitali chiamati "relè protettivi".

Sotto qui vedi uno schema su come comandare un relè con Arduino.

Potresti fare confusione su come inserire il relè all'interno della breadboard. Come ti mostra l'immagine qui sotto, devi piegare leggermente uno dei pin del relè, dopodiché potrai inserirlo nella breadboard.



## Schema di connessione

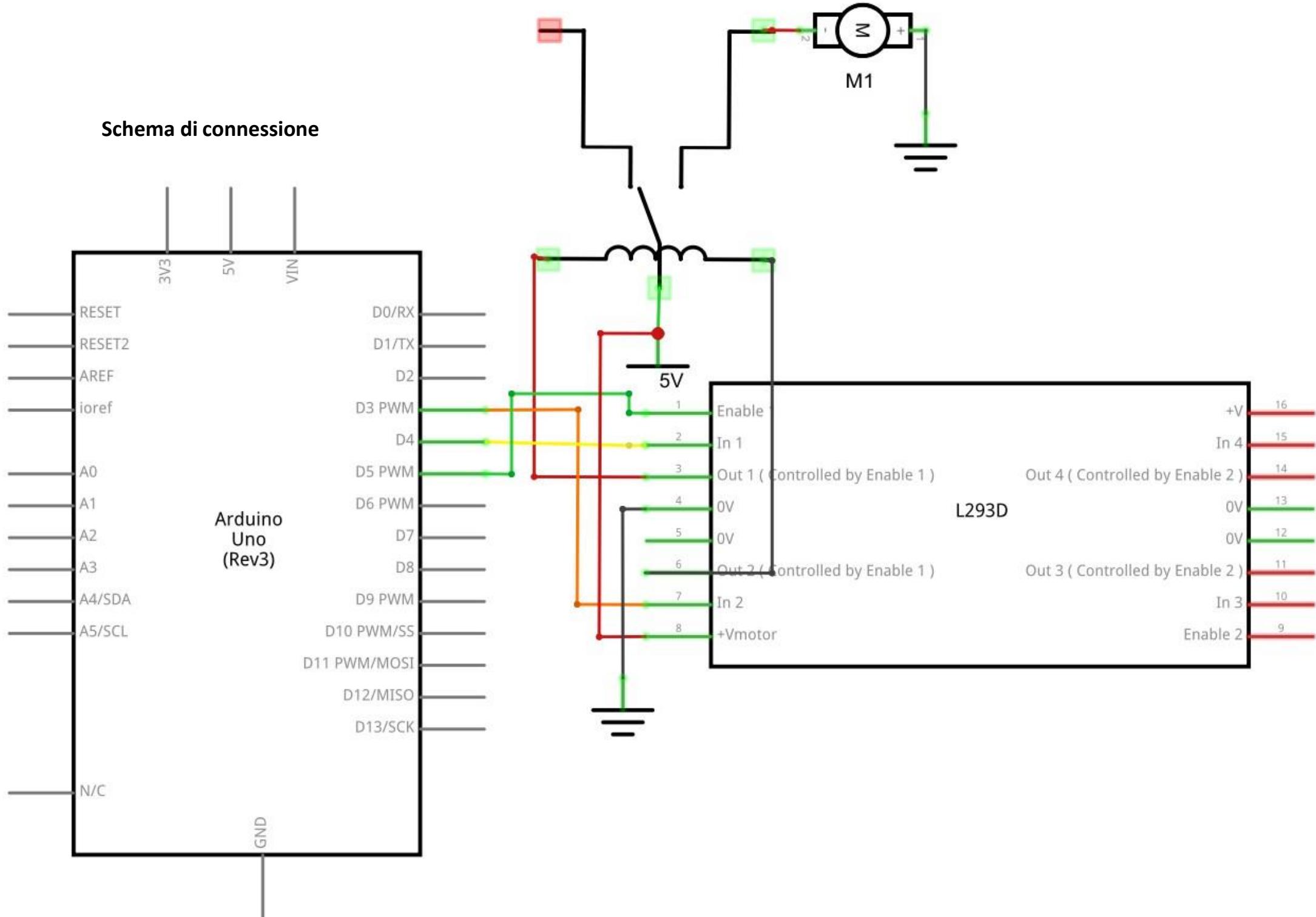
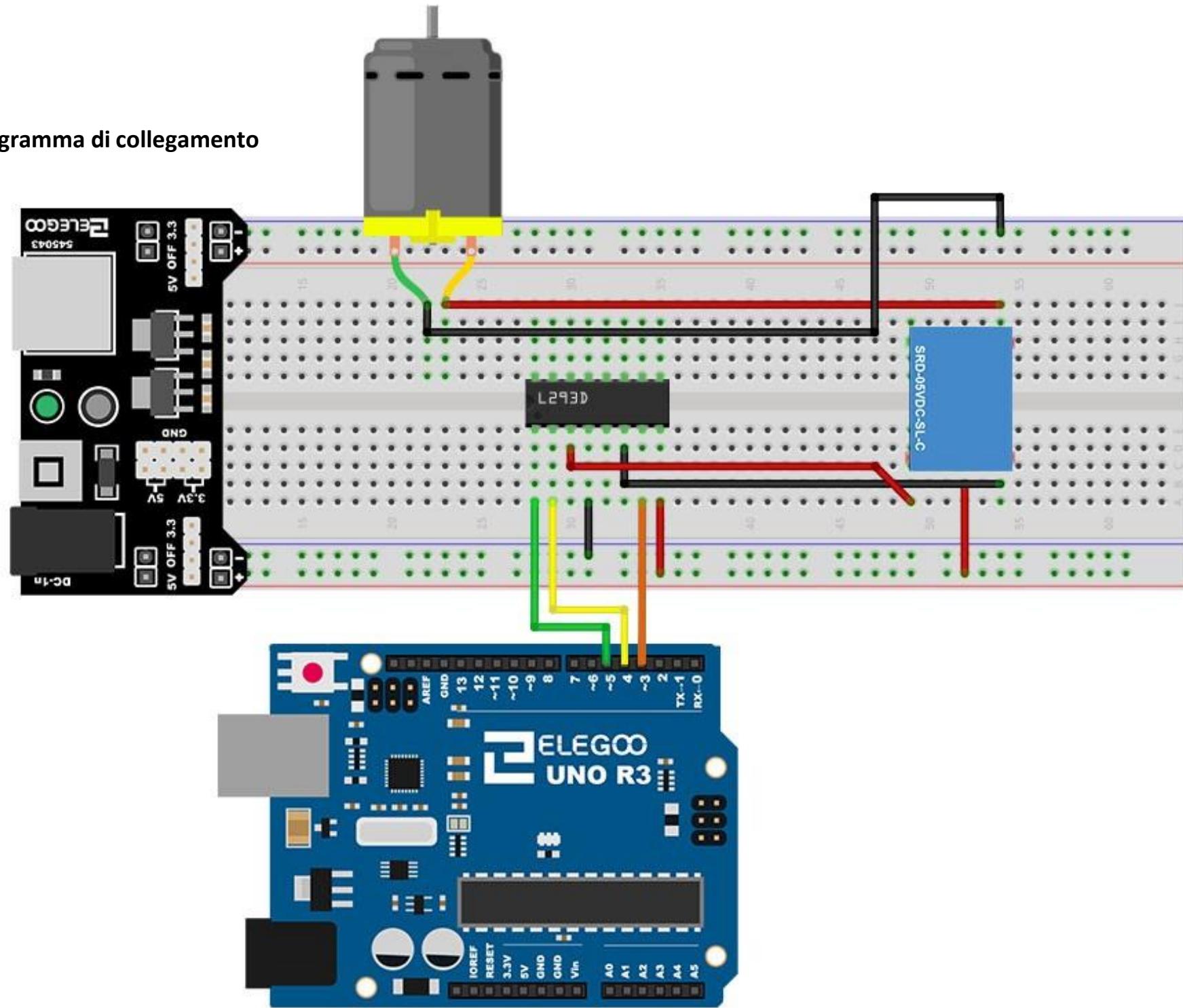


Diagramma di collegamento



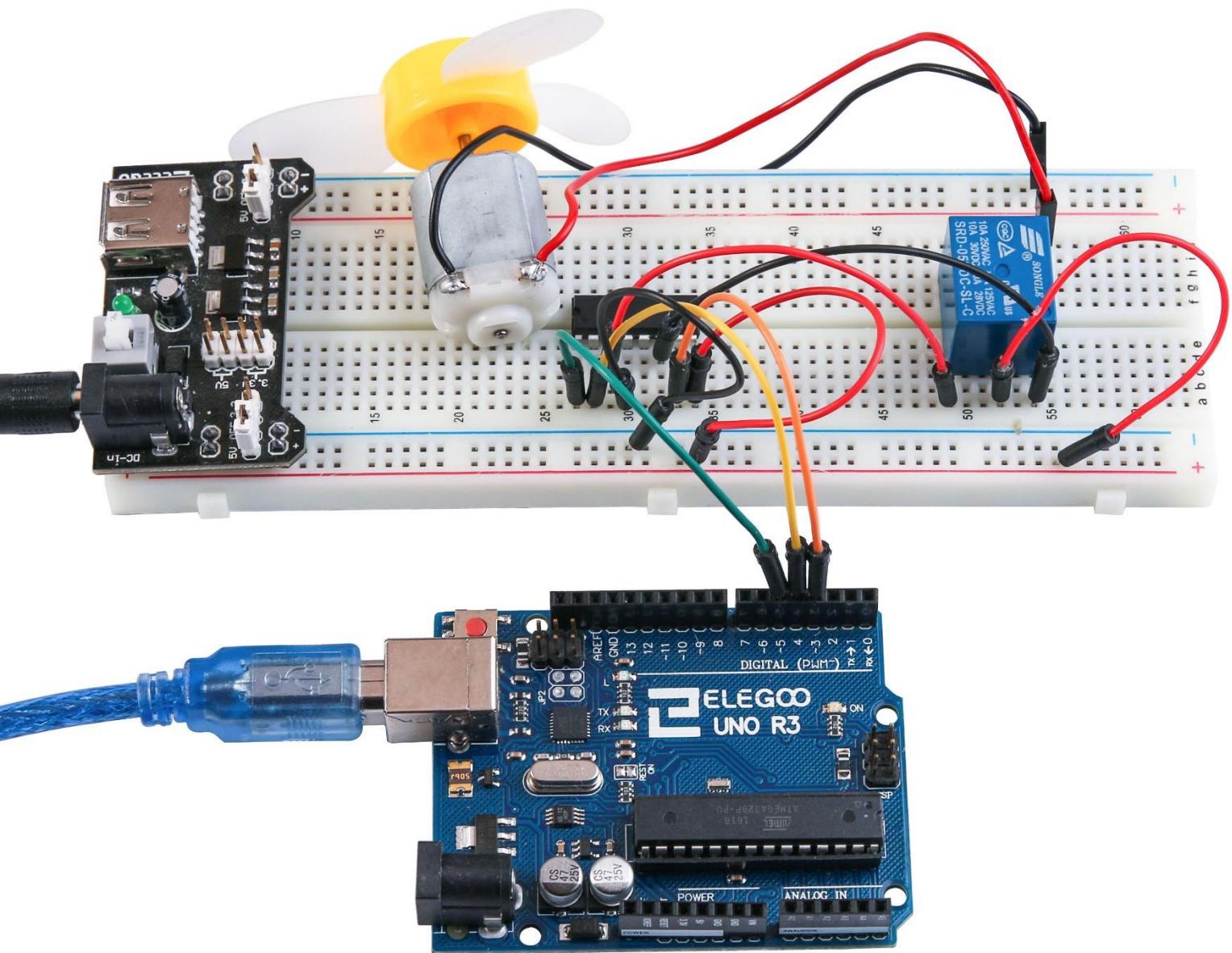
## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella “code” e apri il programma “Lesson 30 Relay” 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

Dopo che si sarà caricato il programma accendi tutti gli interruttori ed il relè emetterà un suono squillante. Poi, il motore girerà. Passato un attimo, il relè verrà rilasciato ed il motore si fermerà.

## Foto d'esempio



## Lezione 31 Motore Passo-Passo

### Introduzione

In questa lezione imparerai un metodo divertente e facile di comandare un motore passo-passo.

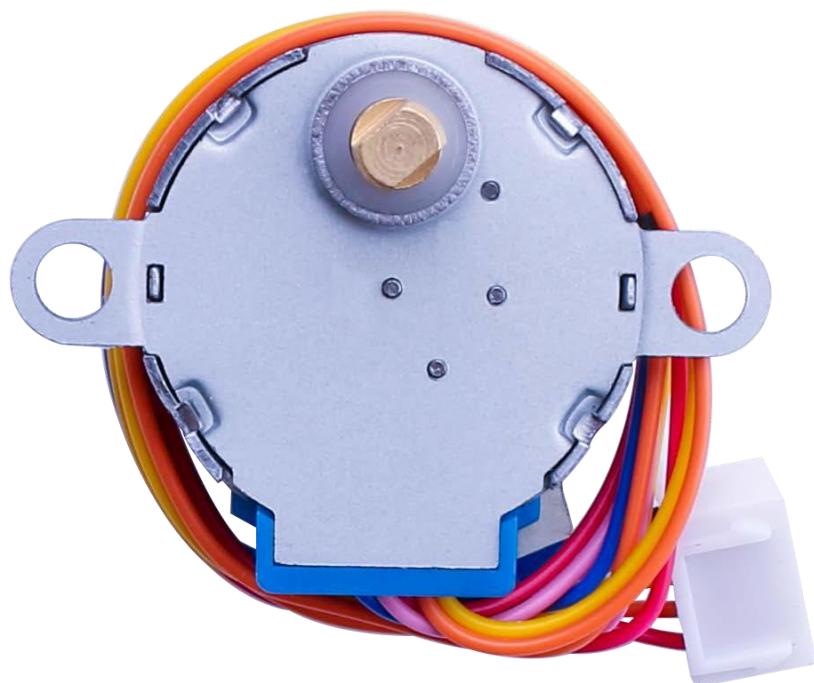
Il motore che utilizzeremo ha già la sua scheda di movimento, ciò permette di connetterlo facilmente al UNO.

### Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento
- (1) x Modulo di guida del motore passo-passo ULN2003
- (1) x Motore passo-passo
- (1) x Adattatore 9V 1A
- (1) x Modulo di alimentazione supplementare
- (6) x F-M cavetti (Cavetti di collegamento di DuPont Femmina – Maschio)
- (1) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)

### Introduzione ai componenti

#### Motore passo passo



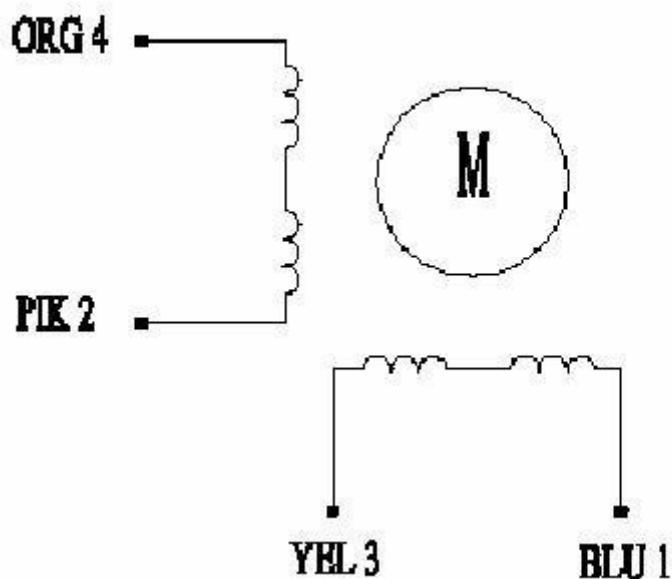
Un motore passo-passo è un dispositivo elettromeccanico che converte impulsi elettronici in movimenti meccanici discreti. L'albero o il mandrino del motore passo-passo ruota di passi predefiniti che incrementano all'aumentare delle pulsazioni elettriche in grado di comandarlo che lo raggiungono nella giusta sequenza. La rotazione del motore ha diversi rapporti diretti con questi impulsi in ingresso applicati. La sequenza degli impulsi applicati è direttamente correlata alla direzione di rotazione dell'albero motore. La velocità di rotazione dell'albero motore è direttamente correlata alla frequenza degli impulsi di ingresso e la durata della rotazione è direttamente correlata al numero di impulsi di ingresso applicati. Uno dei vantaggi più significativi di un motore passo-passo è la sua capacità di essere controllato con precisione in un sistema a ciclo aperto. Il controllo in ciclo aperto significa che non è necessaria alcuna informazioni di feedback sulla posizione. Questo tipo di controllo elimina la necessità di costosi dispositivi di rilevamento e di feedback quali encoder ottici. La posizione è conosciuta semplicemente tenendo traccia degli impulsi in input.

#### **Parametri del motore passo-passo 28BYJ-48**

- Modello: 28BYJ-48
- Voltaggio: 5VDC
- Numero di fasi: 4
- Indice di variazione di velocità: 1/64
- Angolo di passo:  $5.625^\circ / 64$
- Frequenza: 100Hz
- Resistenza :  $50\Omega \pm 7\% (25^\circ C)$
- Frequenza in trazione quando inattivo: > 600Hz
- Frequenza non in trazione quando inattivo: > 1000Hz
- Torsione in trazione > 34.3mN.m(120Hz)
- Torsione di posizionamento > 34.3mN.m
- Torsione in Frizione: 600-1200 gf.cm
- Torsione in traino : 300 gf.cm
- Resistenza isolata > 10MΩ(500V)
- Potenza elettrica isolata: 600VAC/1mA/1s
- Grado di isolamento: A
- Aumento di temperatura < 40K(120Hz)
- Rumore < 35dB(120Hz,No load,10cm)

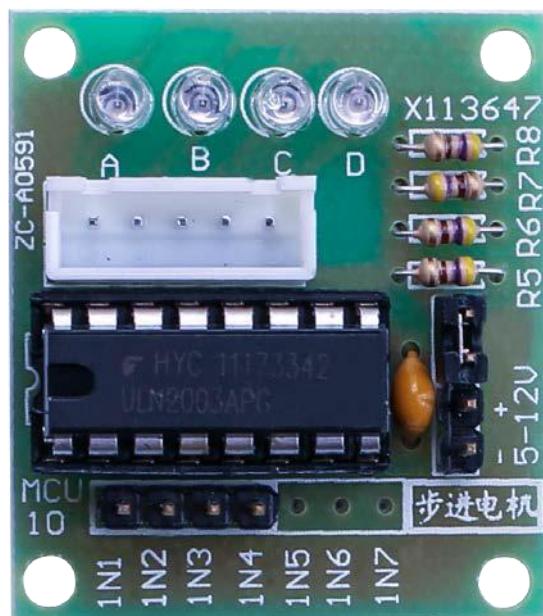
## Circuito di interfaccia

### WIRING DIAGRAM



Il motore passo-passo bipolare solitamente ha quattro cavi che escono da esso. Diversamente dai motori unipolari, i motori bipolari non hanno in centro una connessione comuni. Essi hanno insiemi di bobine indipendenti. Puoi distinguere questo tipo di motori da quelli unipolari misurando la resistenza tra i cavi. Dovresti trovare due paia di motori con resistenza uguale. Se hai due cavo del misuratore connessi con due cavi che non sono collegati (cioè, non collegati alla stessa bobina) dovresti vedere infinita resistenza (non continuità)

## Scheda di guida ULN2003



### Descrizione del prodotto

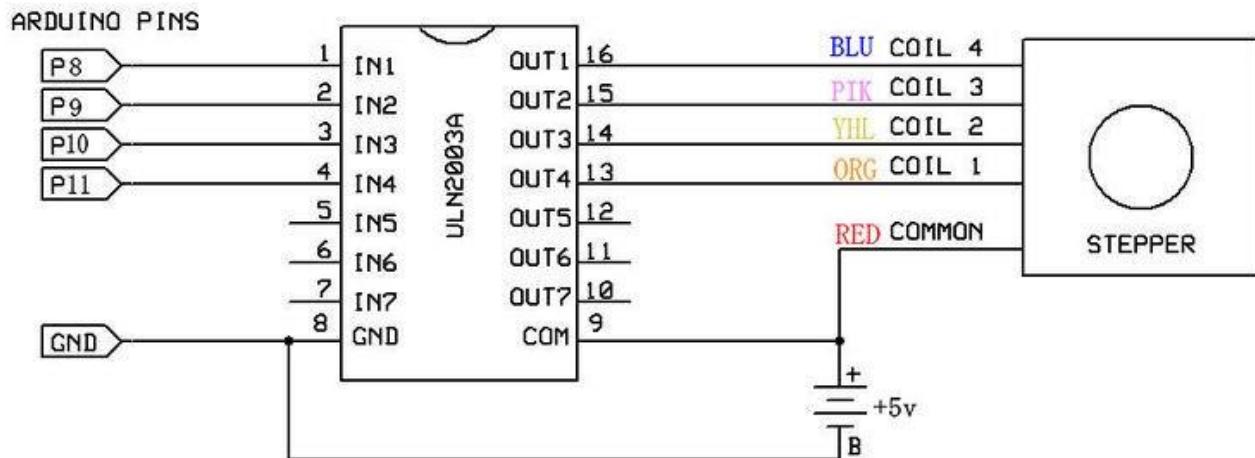
- Dimensioni: 42mmx30mm
- Chip utilizzato :ULN2003, 500mA
- LED A. B. C. D indicanti il funzionamento delle quattro fasi del motore.
- Il connettore bianco è il connettore standard per la quarta fase del motore passo-passo.
- Pin di alimentazione separati
- Lasciamo il resto dei pin del chip ULN2003 per tue future prototipazioni.

Il modo più semplice per interfacciarsi un motore passo-passo unipolare con Arduino è utilizzare un transistor di breakout del chip ULN2003A. Il circuito ULN2003A è in grado di controllare sette Darlington transistor, ciò come avere sette transistor TIP120 tutti in una struttura. Il circuito ULN2003A può alimentare con più di 500mA ogni singolo canale ed ha una caduta di tensione interna di circa 1V quando acceso. Esso contiene anche dei diodi in grado di dissipare picchi di tensione

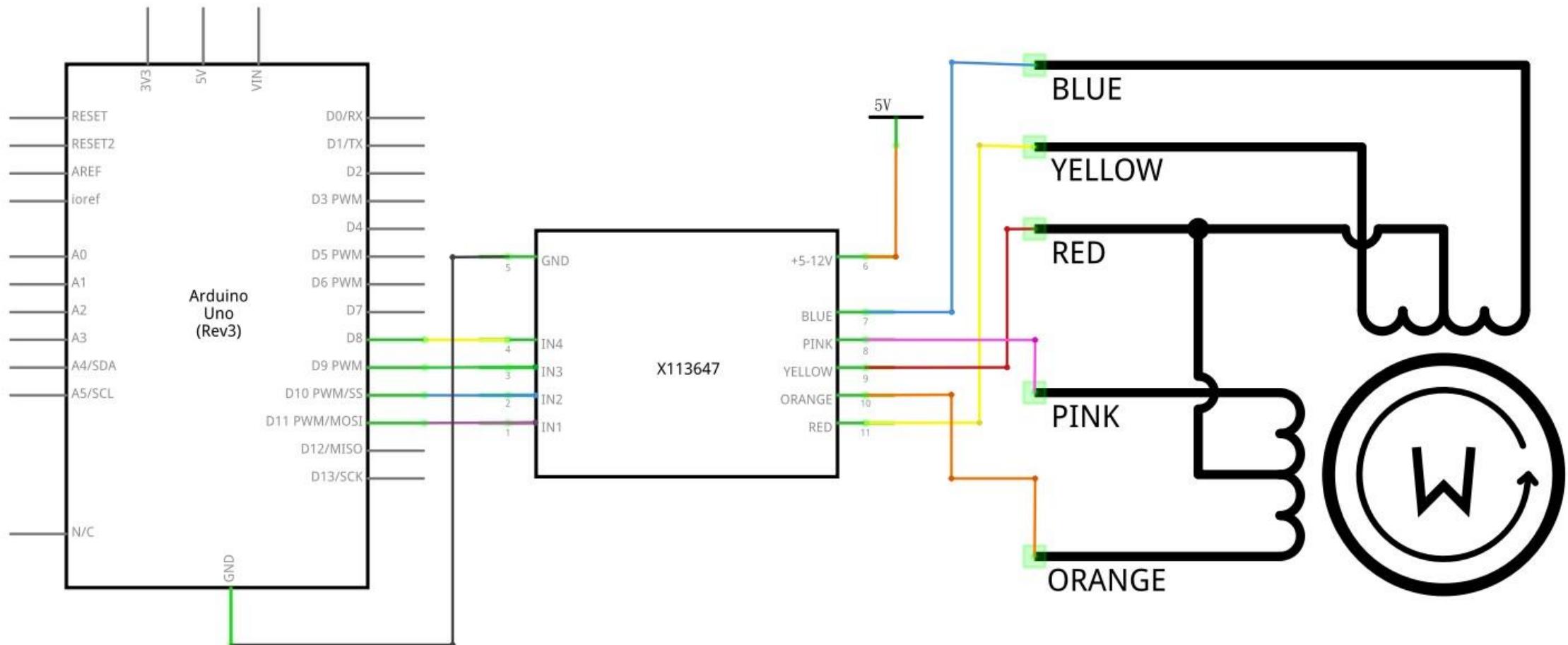
quando comanda carichi induttivi. Per controllare il motore, applica il voltaggio ad ogni bobina in una specifica sequenza. La sequenza va similmente a questa:

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

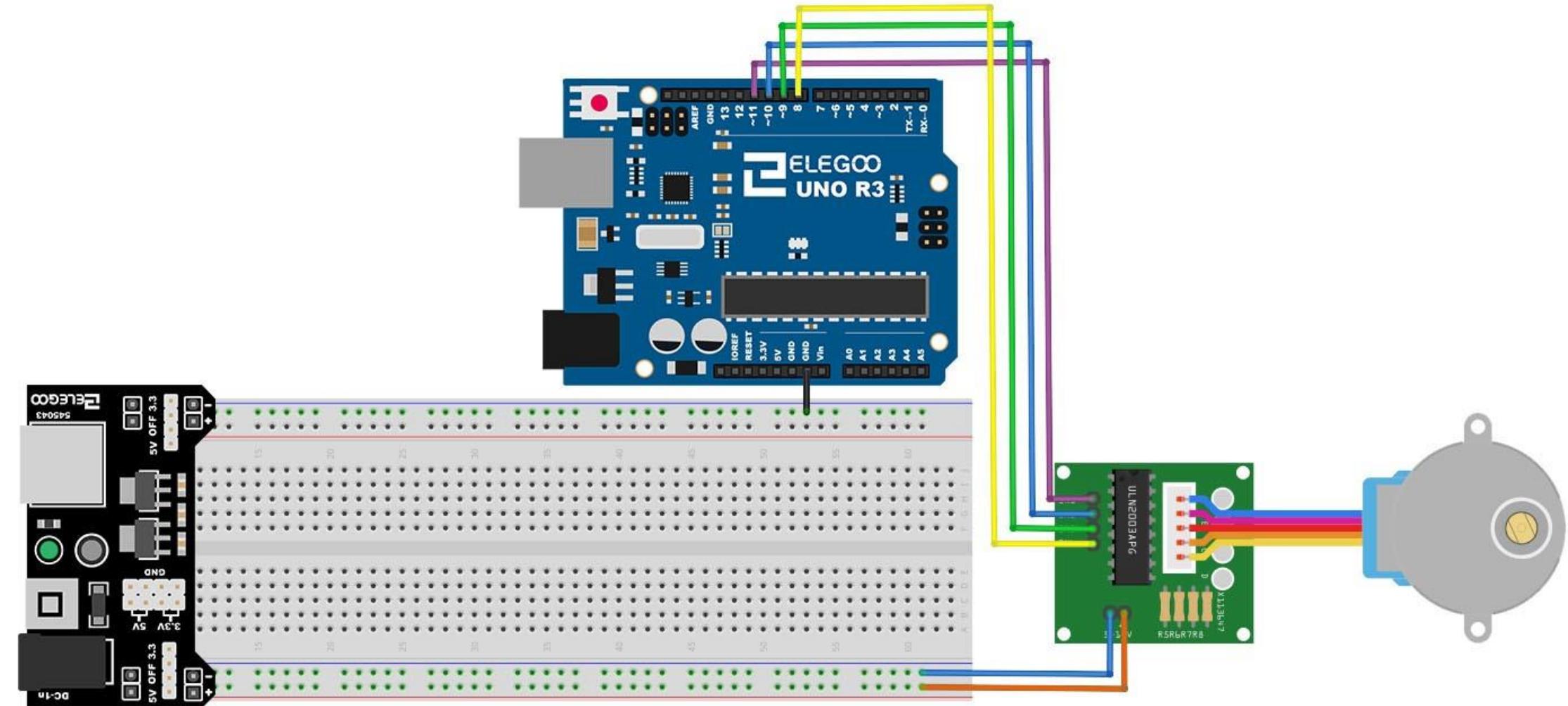
Qui sotto puoi vedere lo schema di come interfacciare un motore unipolare ai quattro pin di controllo utilizzati dal ULN2003A, ti mostra inoltre come interfacciarlo utilizzando quattro collegamenti.



## Connessioni Schema



## Diagramma di collegamento



Stiamo utilizzando 4 pin per controllare il motore.

I Pin 8-11 Stanno controllando il Motore passo-passo.

Abbiamo connesso la messa a terra dal UNO al motore passo-passo.

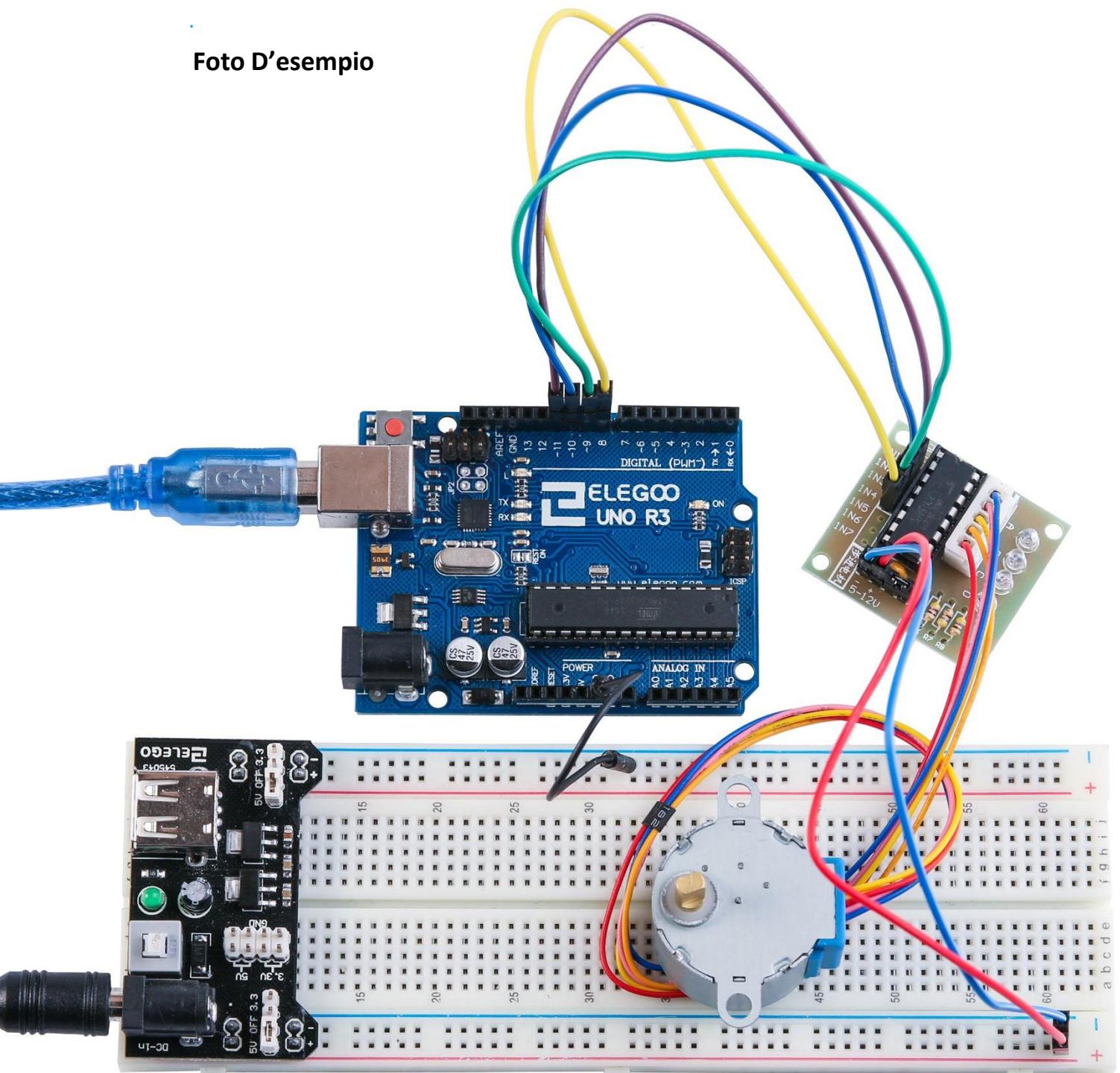
## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lesson 31 Stepper Motor", 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.

Prima di eseguire questo codice, controlla di aver installato la libreria < Stepper > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

## Foto D'esempio



## **Lezione 32 Controllare il motore passo-passo da remoto**

### **Introduzione**

In questa lezione imparerai un metodo veloce e semplice di controllare il motore passo-passo utilizzando il comando ad infrarossi.

Il motore che useremo viene venduto con la propria scheda di controllo, che rende facile la connessione con il UNO.

Dato che non vogliamo controllare il motore direttamente dalla UNO, utilizzeremo un piccolo e comodo alimentatore che si connetterà direttamente alla nostra breadboard per alimentarla con 9V e 1A.

Il sensore ad infrarossi è connesso alla UNO direttamente dato che non necessita di una grande potenza.

### **Componenti Richiesti:**

- (1) x Elegoo UNO R3
- (1) x Modulo ricevitore IR
- (1) x comando IR
- (1) x Modulo di comando motore passo-passo ULN2003
- (1) x Motore passo-passo
- (1) x Modulo di alimentazione supplementare
- (1) x Adattatore 9V 1A
- (9) x F-M cavetti (Cavetti di collegamento di DuPont Femmina – Maschio)
- (1) x M-M cavetti (Cavetti di collegamento Maschio –Maschio)

## Connessione Schema

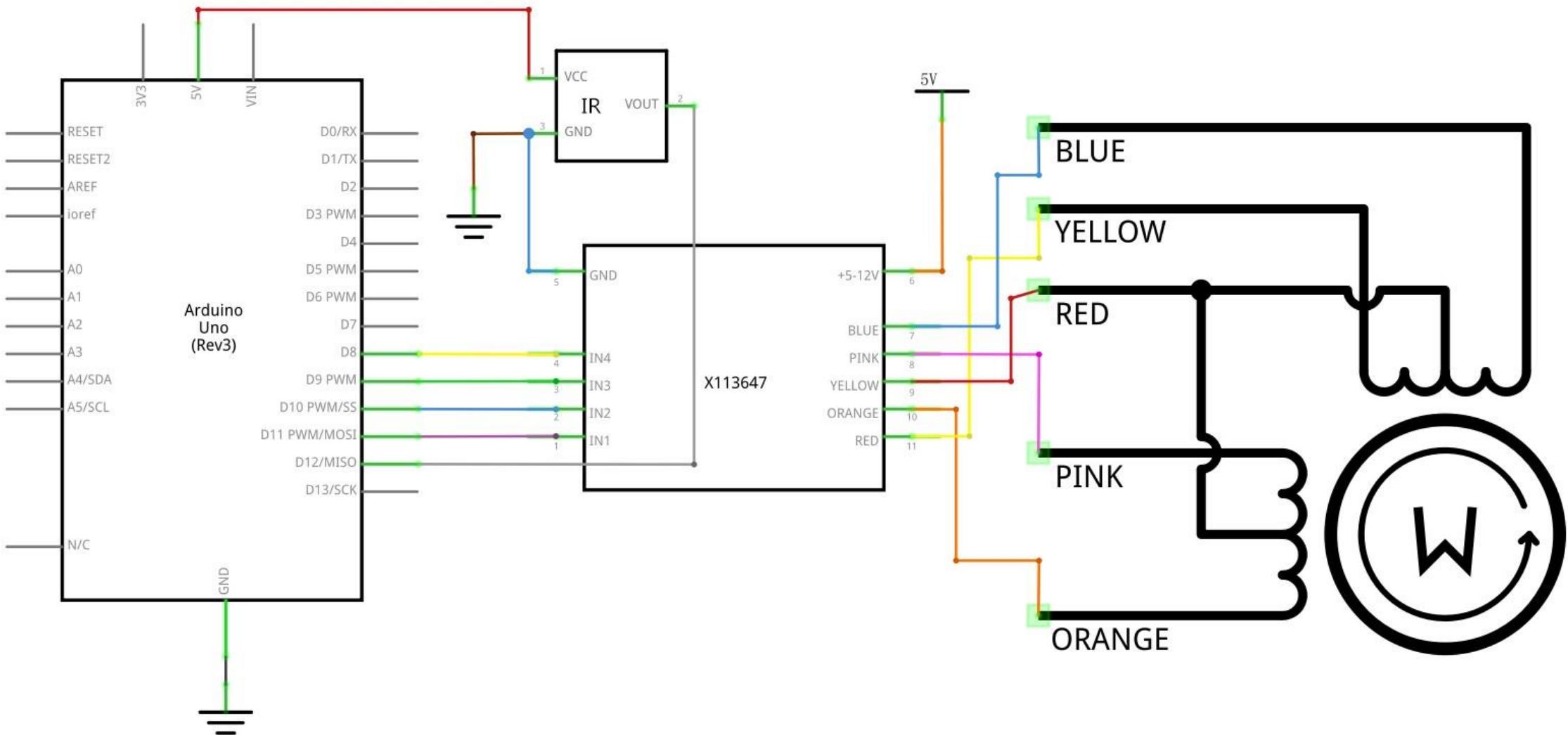
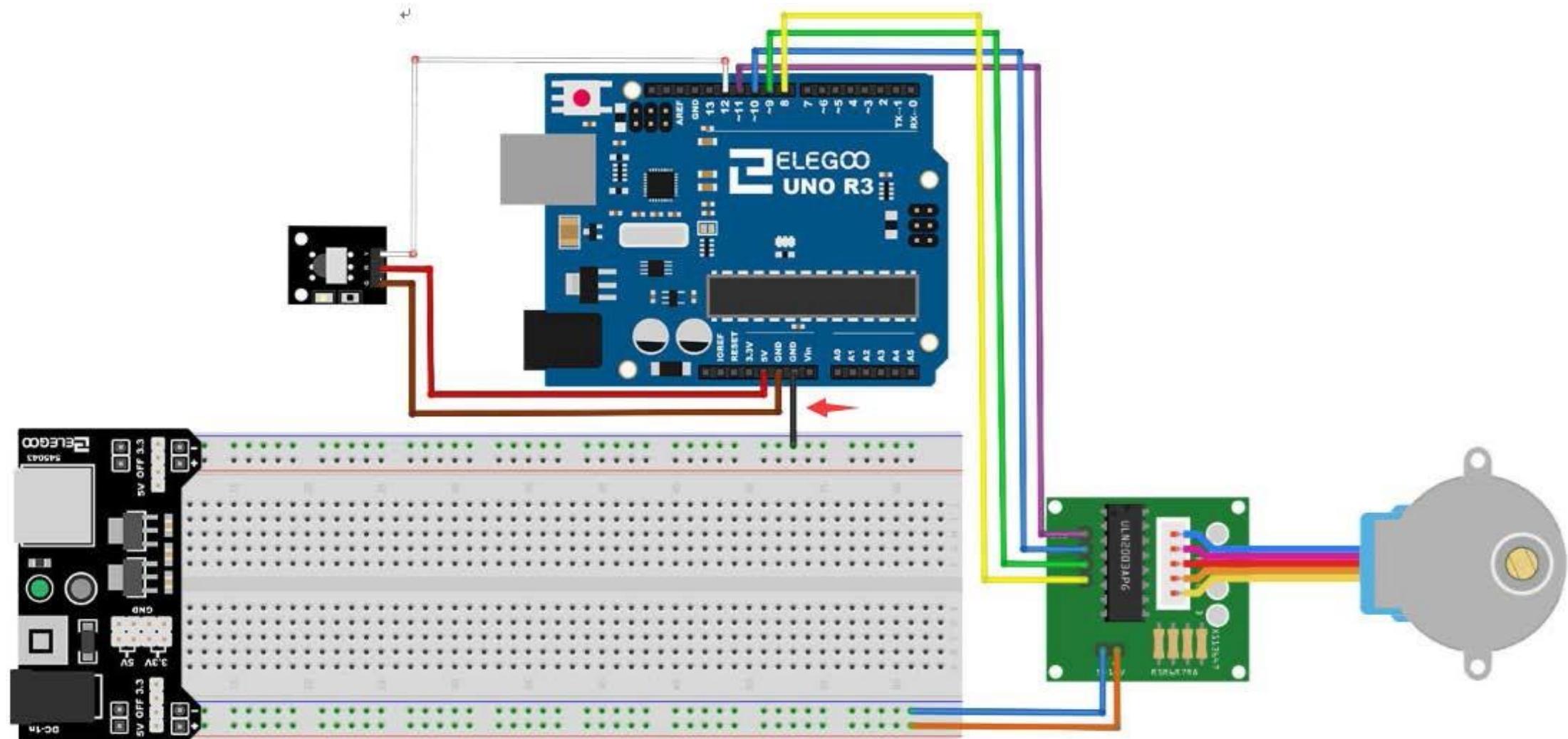


Diagramma di collegamento



Stiamo utilizzando 4 pin per controllare il motore ed un pin per il sensore ad infrarossi.

I pin 8-11 controllano il motore passo-passo ed il pin 12 viene utilizzato per ricevere le informazioni dal motore ad infrarossi.

Connetteremo i 5V e la messa a terra del UNO al sensore. Per precauzione, utilizzeremo l'alimentazione supplementare per il motore passo-passo dato che utilizza molta potenza e non vogliamo rischiare di danneggiare l'alimentazione del UNO.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lesson 32 Controlling Stepper Motor With Remote", 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.

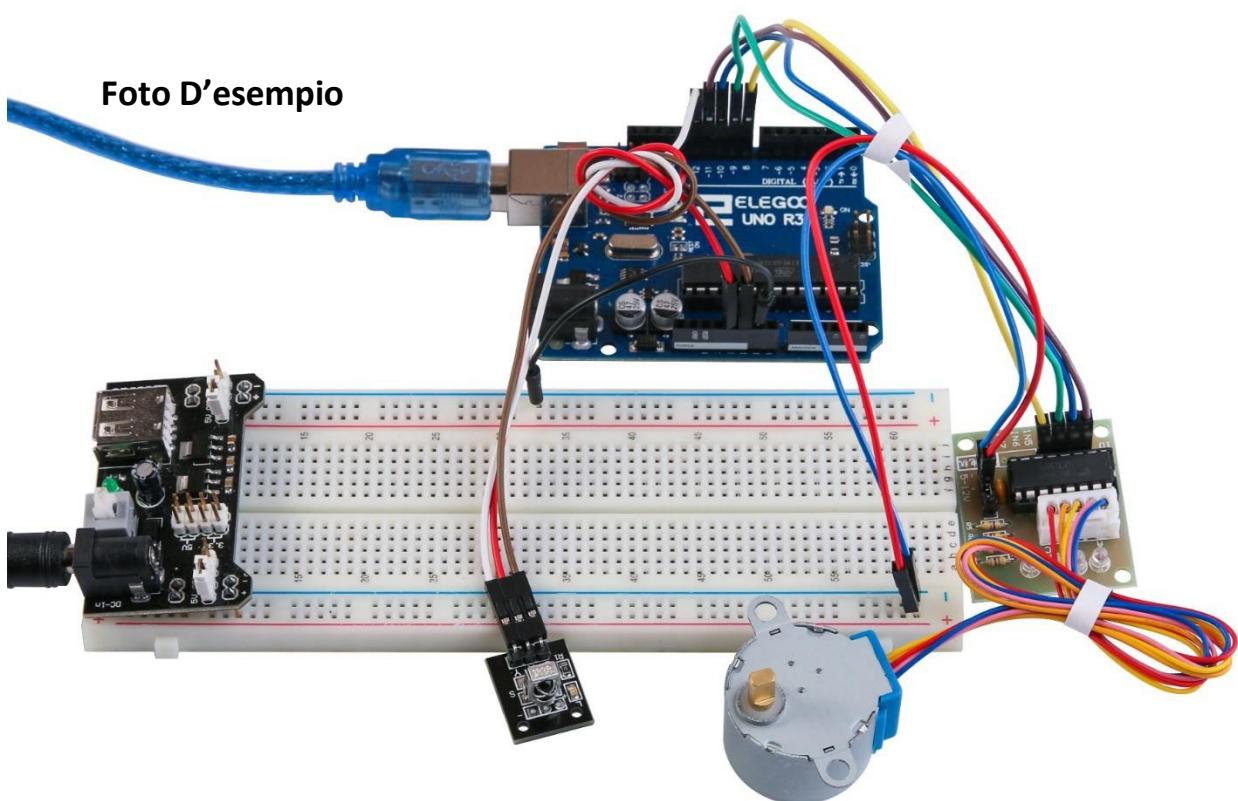
Prima di eseguire questo codice, controlla di aver installato le librerie < IRremote > e < Stepper > oppure reinstallale se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

Il codice riconosce solamente due valori dal comando ad infrarossi: VOL+ e VOL-.

Quando viene premuto il VOL+ il motore farà una rotazione completa in senso orario, premendo invece VOL- si avrà una rotazione in senso anti-orario.

## Foto D'esempio



# Lezione 33 Controllare il motore passo-passo con un encoder

## Introduzione

In questa lezione imparerai come controllare il motore passo-passo con un encoder a rotazione. Utilizzeremo un motore economico e popolare che viene venduto con la scheda di controllo: il motore 28BYJ-48 con la scheda ULN2003.

Il motore 28BYJ-48 non è molto forte o molto veloce, ma è ottimo per i principianti che iniziano le sperimentazioni di controllo de motore passo-passo con Arduino.

Scriveremo del codice per controllare il motore e muoverlo nella direzione in cui ruoteremo l'encoder, e terremo anche traccia di quanti passi sono stati fatti, così potremo riportarlo nella posizione di partenza premendo l'interruttore dell'encoder.

## Componenti Richiesti:

- (1) x Elegoo UNO R3
- (1) x Breadboard con 830 punti di collegamento 830
- (1) x Modulo encoder di rotazione
- (1) x Modulo di comando motore passo-passo ULN2003
- (1) x Motore passo-passo
- (1) x Modulo di alimentazione supplementare
- (1) x Adattatore 9V 1A
- (9) x F-M cavetti (Cavetti di collegamento di DuPont Femmina – Maschio)
- (1) x M-M cavetti (Cavetti di collegamento Maschio – Maschio)

## Introduzione ai componenti

### Encoder a rotazione

Un encoder a rotazione, anche chiamato encoder ad asta, è un dispositivo elettromeccanico che converte una posizione angolare, un movimento dell'asta o dell'asse in un valore analogico o digitale.

Ci sono due tipi di valori: assoluto e incrementale (relativo). L'uscita dell'encoder assoluto indica la posizione corrente dell'asta tramite posizione angolare. L'uscita incrementale fornisce informazioni riguardo al movimento dell'asta, che è tipicamente nuovamente trasformato in informazioni quali la velocità, la distanza e la posizione. Gli encoder a rotazione sono usati in molte applicazioni che necessitano rotazioni infinite – inclusi controlli industriali, robotica, speciale lenti fotografiche, dispositivi di input per pc (come mouse e trackball) e piattaforme radar rotanti.



## Connessione Schema

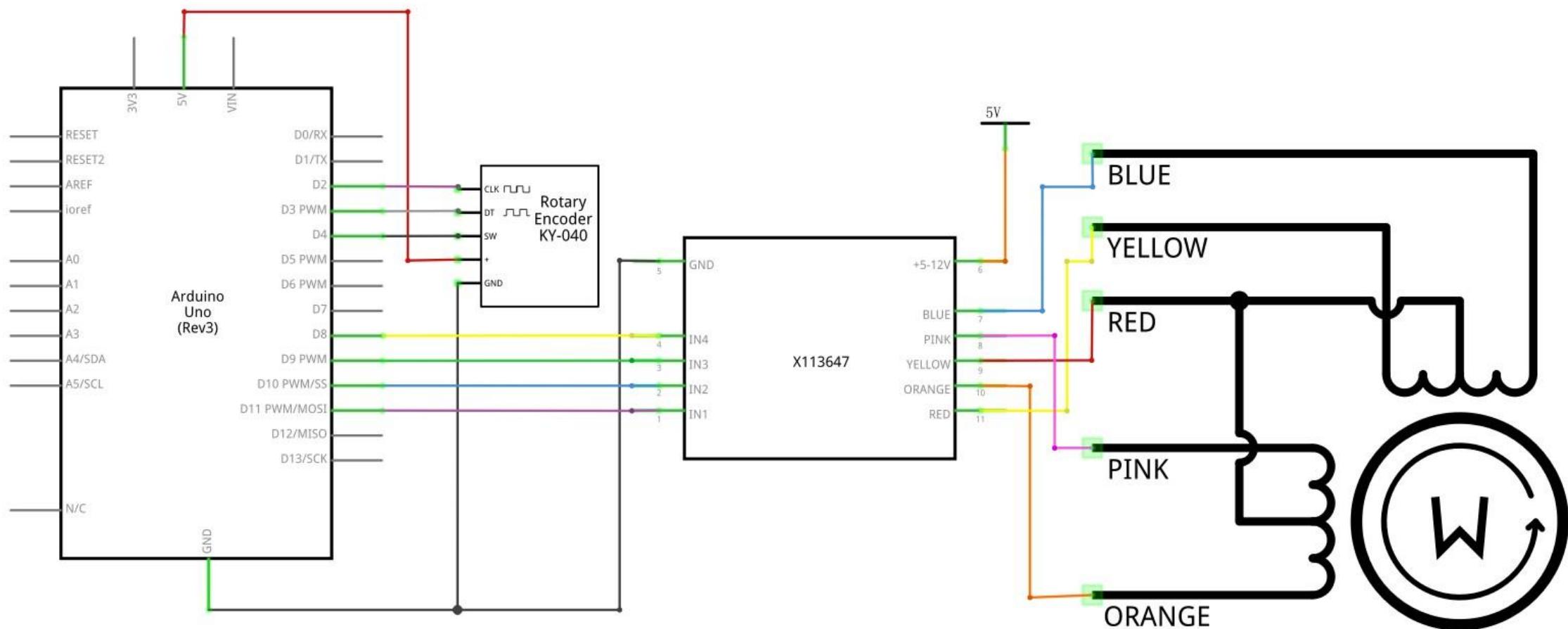
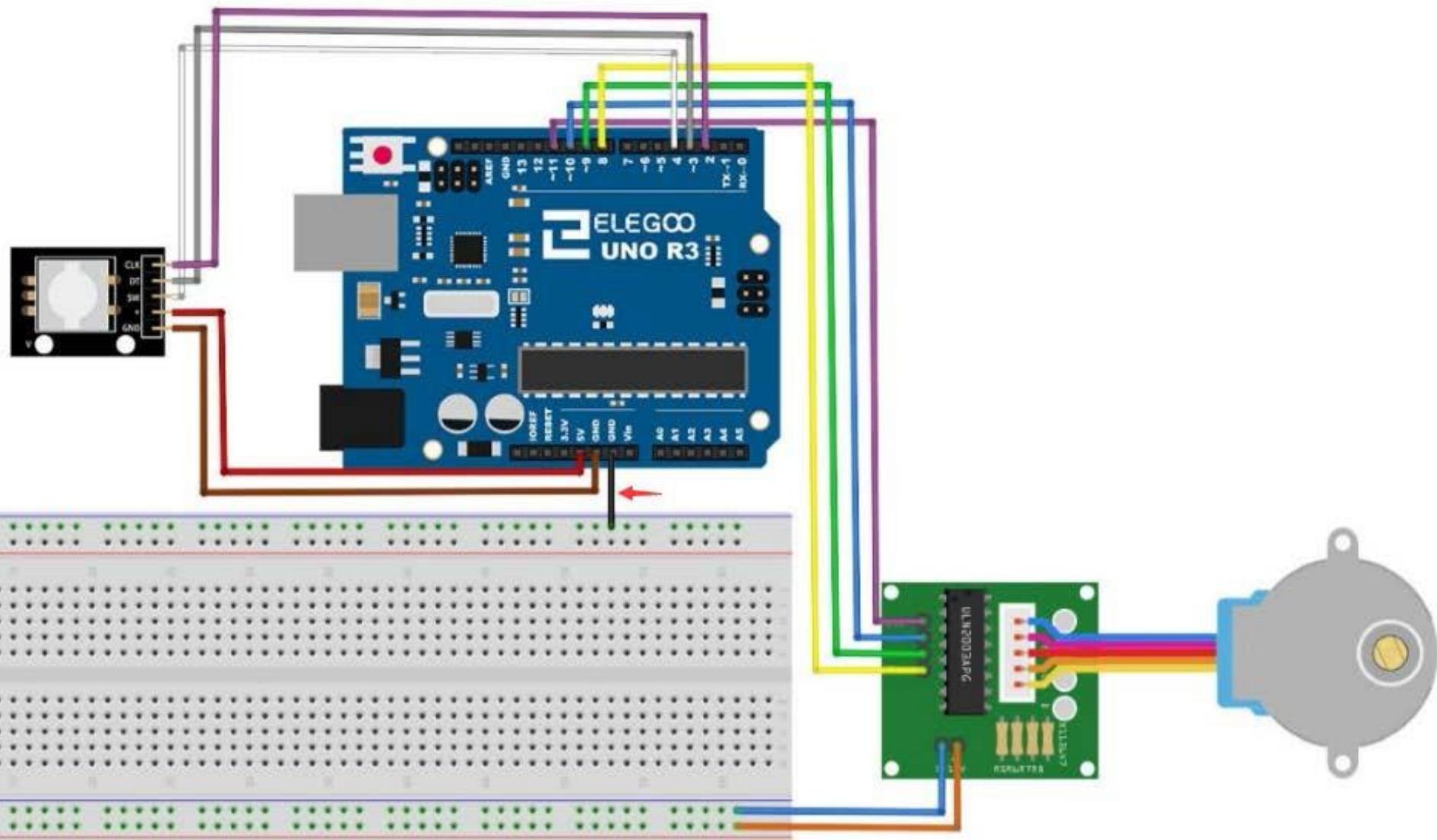


Diagramma di collegamento



Utilizzeremo 4 pin per controllare il motore e 3 pin per il modulo encoder.

I pin 8-11 controllano il motore passo-passo e i pin 2-4 ricevono informazioni dall'encoder.

Connetteremo i 5V e la messa a terra del UNO all'encoder e per precauzione l'alimentazione del motore la ricaveremo dall'alimentazione supplementare.

Possiamo anche connettere il UNO alla messa terra della breadboard per utilizzarla come riferimento.

## Codice

Dopo aver effettuato i collegamenti, naviga fino alla cartella "code" e apri il programma "Lesson 33 Controlling Stepper Motor With Rotary Encoder", 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.

Prima di eseguire questo codice, controlla di aver installato la libreria < Stepper > oppure reinstallala se necessario. Altrimenti il codice non riuscirà a funzionare.

Per i dettagli su come caricare i file delle librerie, guarda la lezione 1.

Utilizzeremo alcune variabili per salvare la posizione corrente, dato che vogliamo tenere traccia della posizione del motore passo-passo in modo da poterlo riportare alla posizione di partenza.

Includeremo inoltre alcuni controlli sugli errori in modo da essere sicuri che l'encoder non perda qualche passo, rendendo così la posizione del mostro motore più accurata.

## Foto d'esempio

