



## **Elegoo Super Starter Kit pentru UNO**

**V1.0.18.12.18**

# Prefață

## Compania noastră

Înființată în 2011, Elegoo Inc. este o companie de tehnologie înfloritoare dedicată cercetării și dezvoltării, producției și marketingului hardware opensource. Situat în Shenzhen, Silicon Valley din China, am crescut la peste 150 de angajați cu o fabrică de peste 10.763 de metri pătrați.

Liniile noastre de produse au variat de la fire DuPont, plăci UNO R3 până la kituri de pornire complete concepute pentru clienții de orice nivel pentru a învăța cunoștințele Arduino. În plus, vindem și produse de accesorii Raspberry Pi, cum ar fi 2.8" TFT touch și STM32. În viitor, vom dedica mai multă energie și investiții produselor pentru imprimante 3D și așa mai departe. Toate produsele noastre respectă standardele internaționale de calitate și sunt foarte apreciate pe o varietate de piețe diferite din întreaga lume.

Site oficial: <http://www.elegoo.com>

Vitrina Amazon din SUA: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1> Vitrina

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

Marea Britanie: <http://www.amazon.co.uk/magazine/AZF7WYXU5ZANW> DE Vitrina

Amazon: <http://www.amazon.de/shops/AZF7WYXU5ZANW> FR Vitrina Amazon: <http://www.amazon.fr/shops/AZF7WYXU5ZANW>

ES Vitrina Amazon: <http://www.amazon.es/shops/AZF7WYXU5ZANW> IT

Vitrina Amazon: <http://www.amazon.it/shops/AZF7WYXU5ZANW>

## Tutorialul nostru

Acest tutorial este conceput pentru începători. Veți învăța toate informațiile de bază despre cum să utilizați placa de control, senzorii și componentele Arduino. Dacă doriți să studiați Arduino mai în profunzime, vă recomandăm să citiți Cartea de bucate Arduino scrisă de Michael Margolis.

Unele coduri din acest tutorial sunt editate de Simon Monk. Simon Monk este autorul unui număr de cărți referitoare la Hardware Open Source. Sunt disponibile pe Amazon: Programare Arduino, 30 de proiecte Arduino pentru Geniul Rău și Programarea Raspberry Pi.

## Serviciu clienți

Ca o companie de tehnologie continuă și cu creștere rapidă, continuăm să ne străduim tot posibilul

să vă oferim produse excelente și servicii de calitate pentru a vă satisface așteptările și ne puteți contacta pur și simplu la adresa [service@elegoo.com](mailto:service@elegoo.com) sau [EUservice@elegoo.com](mailto:EUservice@elegoo.com) . Așteptăm cu nerăbdare să auzim de la tine și orice comentariu sau sugestie critică ar fi foarte valoroasă pentru toate.

Și orice probleme și întrebări pe care le aveți cu produsele noastre vor primi un răspuns prompt de către inginerii noștri experimentați în termen de 12 ore (24 de ore în timpul vacanței)

# Lista de ambalare

 [www.elegoo.com](http://www.elegoo.com)

Motor pas cu pas 1buc



Servomotor (SG90)  
1buc



Modul receptor IR  
1PC



Releu 5V 1buc



Placă de  
control Uno R3  
1buc



Modul LCD 1602 (cu  
antet pin) 1buc



ULN2003 Modul driver de  
motor pas cu pas  
1buc



Modul de alimentare  
1PC



Modul de extindere prototip  
1PC



Contactați-ne: [service@elegoo.com](mailto:service@elegoo.com)

Afișaj cu 4 cifre și 7 segmente

1buc



Afișaj 1 cifră 7 segmente

1buc



Comutator cu bilă înclinată

1buc



Lama ventilatorului

și motor

3-6V DC

(cu fir)

fiecare

1buc



Modul Joystick 1PC



DHT11 Temperatura

și Modul de umiditate

1PC S



Senzor cu ultrasunete 1buc



Buton (mic) 5BUC



Potențiometru 1buc



Buzzer pasiv 1buc



Baterie de 9 V cu

clemă pentru conector snap-on

1buc



Buzzer activ 1PC



Telecomanda

1PC



Cablu USB 1buc



Sârmă Dupont  
femelă-mască

10 buc



Sârmă Jumper pentru

Placă 65BUC



74HC595 IC 1buc



L293D

1buc



830 Tie-Points  
Breadboard

1buc



Contactați-ne: [service@elegoo.com](mailto:service@elegoo.com)

Termistor 1buc



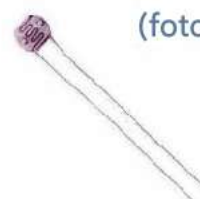
Redresor cu diode 2 buc



LED  
25BUC



Fotorezistor  
(fotocelula) 2BUC



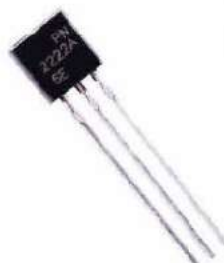
Rezistori  
120 buc



RGB  
2BUC



Tranzistor NPN PN2222 2BUC



## Conținut

|                                                                  |     |
|------------------------------------------------------------------|-----|
| Lecția 0 Instalarea IDE.....                                     | 9   |
| Lecția 1 Adăugați biblioteci și deschideți monitorul serial..... | 22  |
| Lecția 2 Clipește.....                                           | 31  |
| Lecția 3 LED.....                                                | 39  |
| Lecția 4 LED RGB.....                                            | 46  |
| Lecția 5 Intrări digitale.....                                   | 55  |
| Lecția 6 Sonerie activă.....                                     | 60  |
| Lecția 7 Buzzer pasiv.....                                       | 64  |
| Lecția 8 Comutator cu bilă înclinată.....                        | 68  |
| Lecția 9 Servo.....                                              | 72  |
| Lecția 10 Modulul senzorului cu ultrasunete.....                 | 76  |
| Lecția 11 Senzor de temperatură și umiditate DHT11.....          | 81  |
| Lecția 12 Modulul Joystick analogic.....                         | 87  |
| Lecția 13 Modulul receptor IR.....                               | 92  |
| Lecția 14 Display LCD.....                                       | 98  |
| Lecția 15 Termometru.....                                        | 103 |
| Lecția 16 Opt LED-uri cu 74HC595.....                            | 108 |
| Lecția 17 Monitorul serial.....                                  | 115 |
| Lecția 18 Fotocelulă.....                                        | 121 |
| Lecția 19 74HC595 și afișaj cu segmente.....                     | 126 |
| Lecția 20 Afișaj digital cu șapte segmente.....                  | 132 |
| Lecția 21 Motoare cu curent continuu.....                        | 136 |
| Lecția 22 Stafeta.....                                           | 146 |
| Lecția 23 Motorul pas cu pas.....                                | 151 |
| Lecția 24 Controlul motorului pas cu telecomandă.....            | 159 |

## Lecția 0 Instalarea IDE

### Introducere

Arduino Integrated Development Environment (IDE) este partea software a platformei Arduino.

În această lecție, veți învăța cum să vă configurați computerul pentru a utiliza Arduino și cum să stabiliți lecțiile care urmează.

Software-ul Arduino pe care îl veți folosi pentru a vă programa Arduino este disponibil pentru Windows, Mac și Linux. Procesul de instalare este diferit pentru toate cele trei platforme și, din păcate, există o anumită cantitate de muncă manuală pentru a instala software-ul.

**PASUL 1:** Accesați <https://www.arduino.cc/en/Main/Software> și găsiți pagina de mai jos.



### ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer  
**Windows** ZIP file for non admin install

**Windows app** 

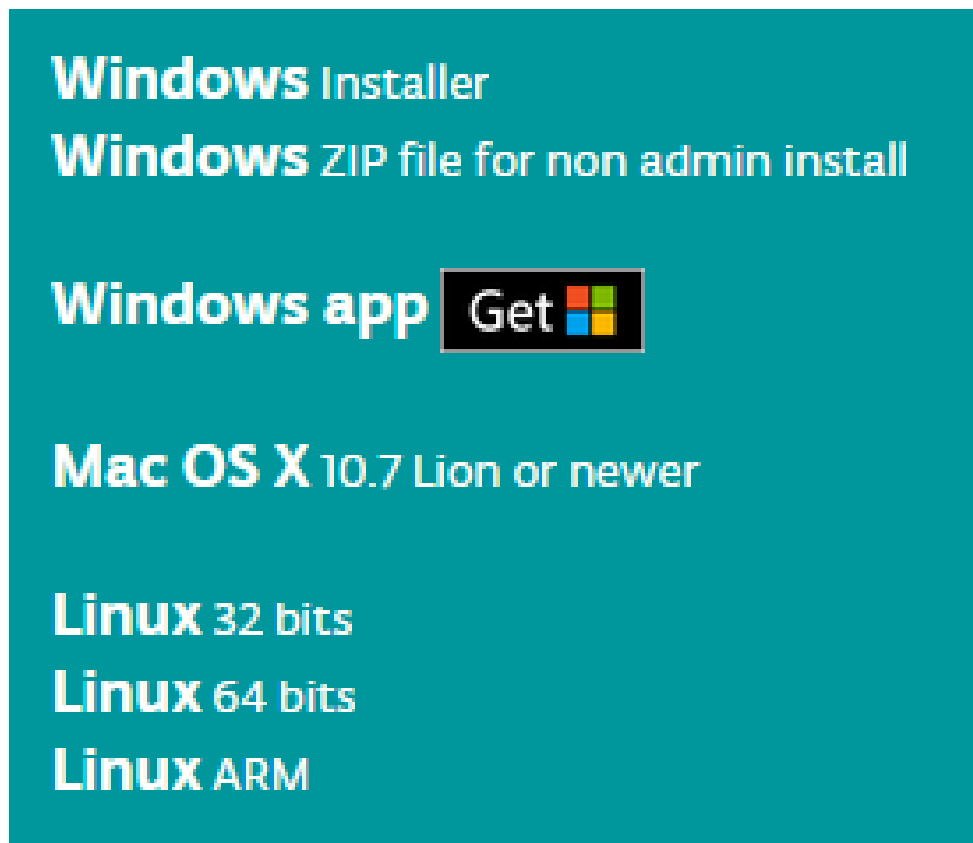
**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

**Versiunea disponibilă pe acest site web este de obicei cea mai recentă versiune, iar versiunea reală poate fi mai nouă decât versiunea din imagine.**

**PASUL 2: Descărcați software-ul de dezvoltare compatibil cu sistemul de operare al computerului dvs.**[Luați Windows ca exemplu aici.](#)



Clic *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.

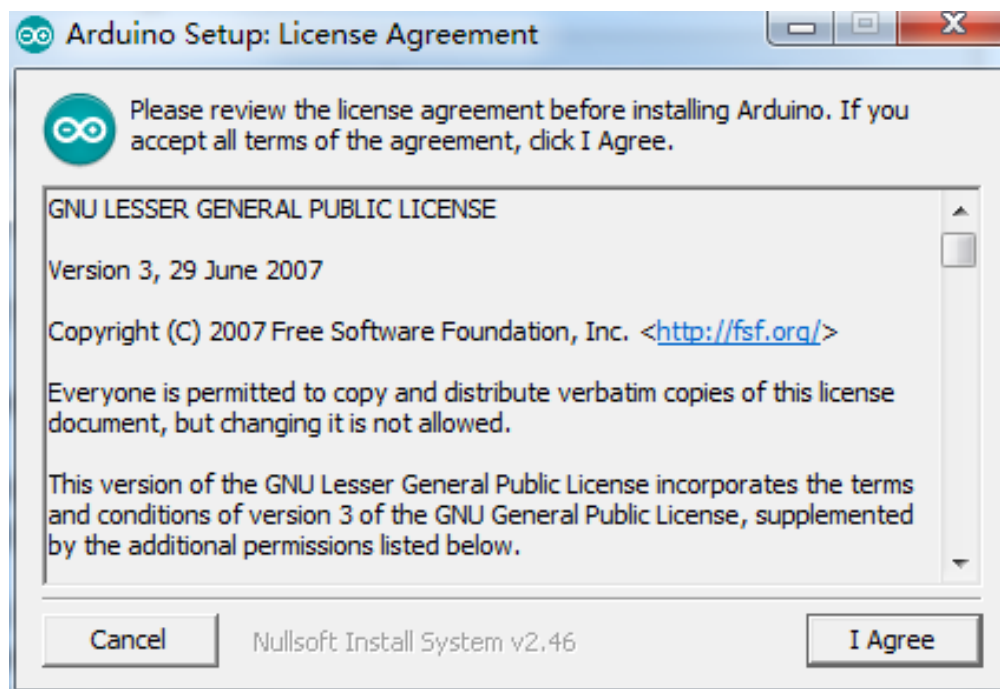
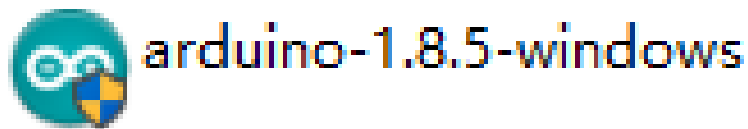


Clic *DOAR DESCARCĂ*.

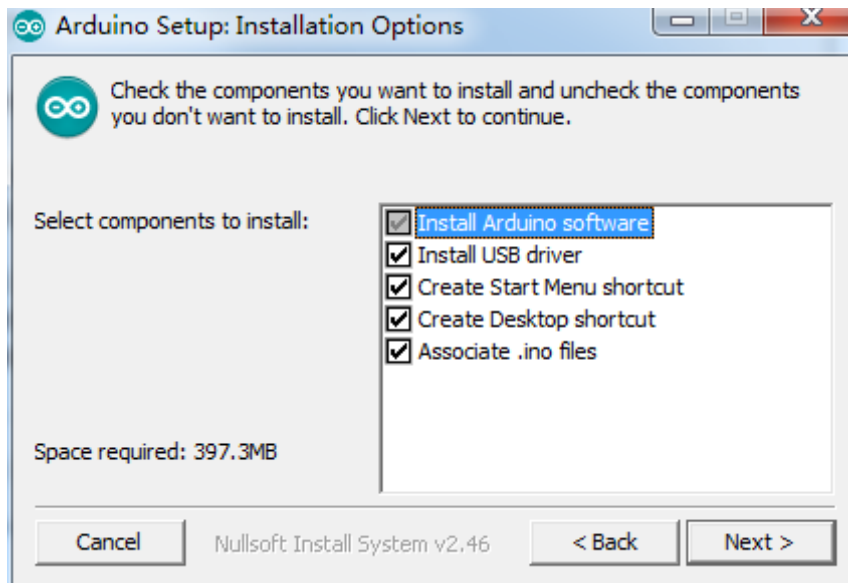
În același timp, puteți descărca și Arduino IDE de pe site-ul nostru oficial. <http://www.elegoo.com/download/>.

## Instalarea Arduino (Windows)

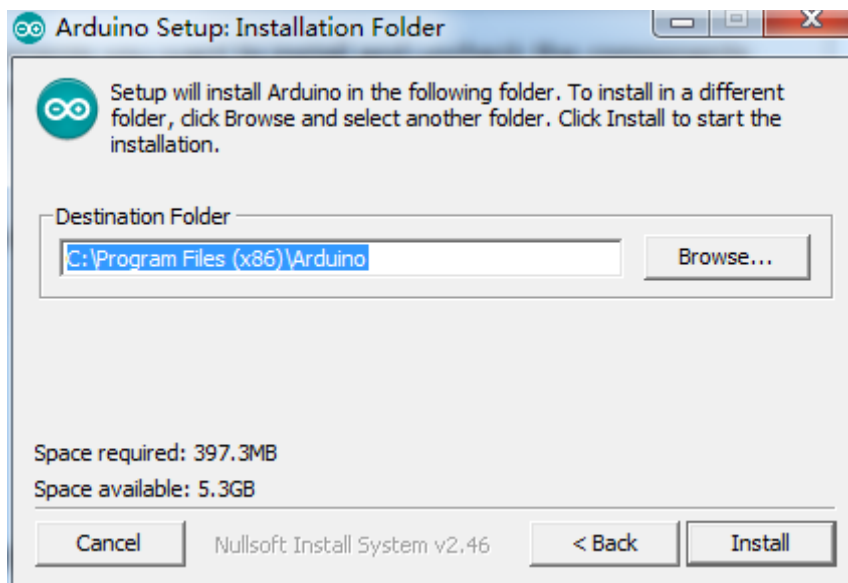
Instalați Arduino cu exe. Pachetul de instalare.



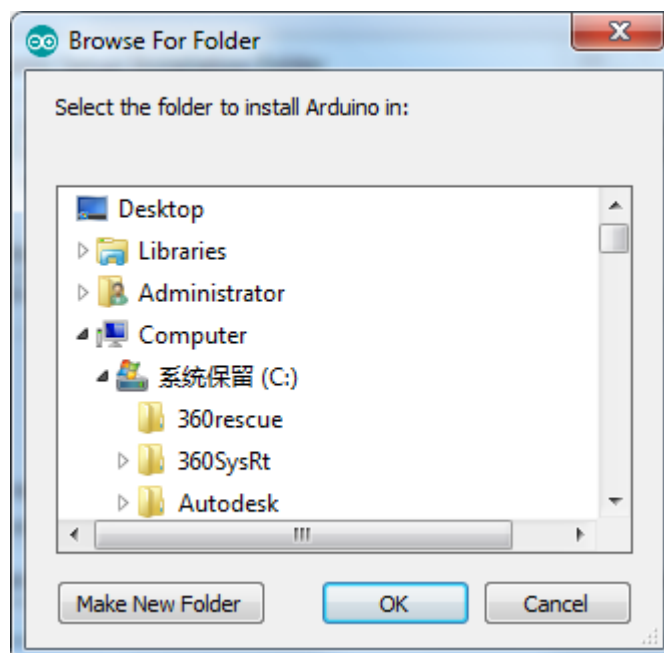
Clic *Sunt de acord* pentru a vedea următoarea interfață



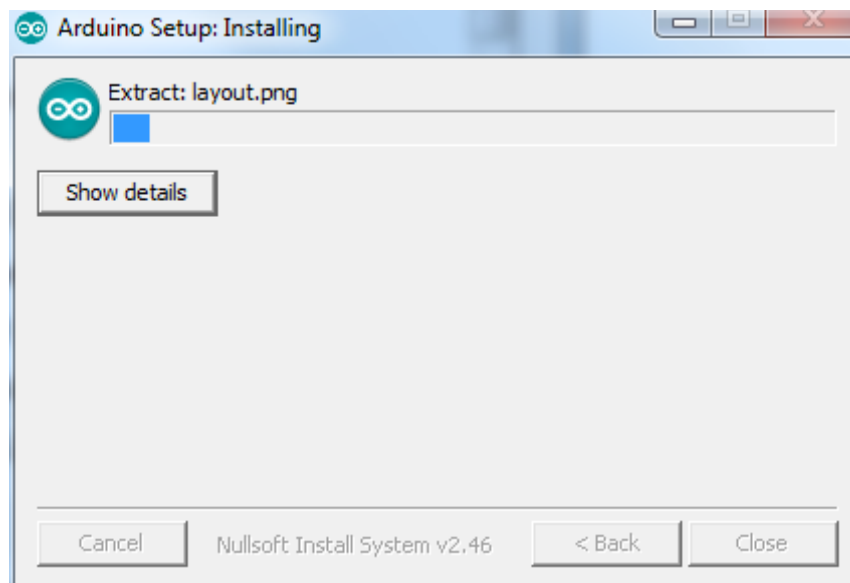
ClicUrmătorul



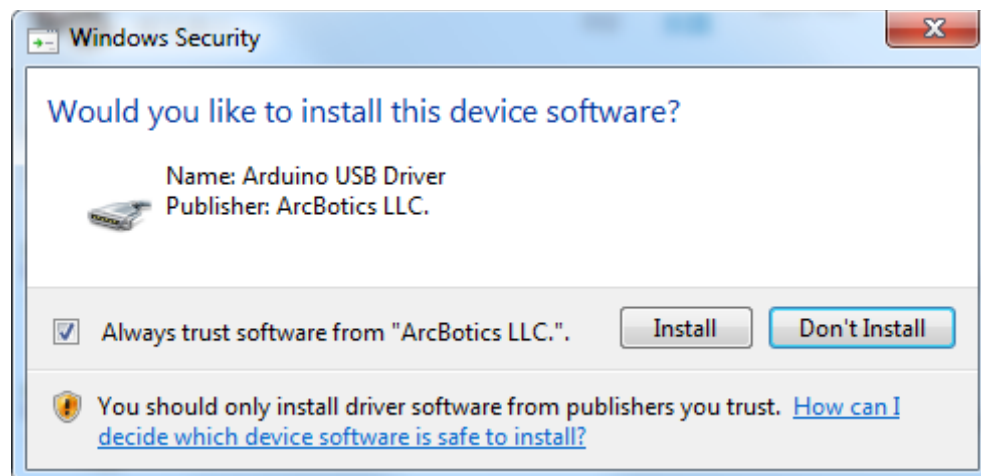
Puteți apăsa Răsfoire... pentru a alege o cale de instalare sau puteți introduce direct directorul dorit.



Clic *Instalare* pentru a iniția instalarea



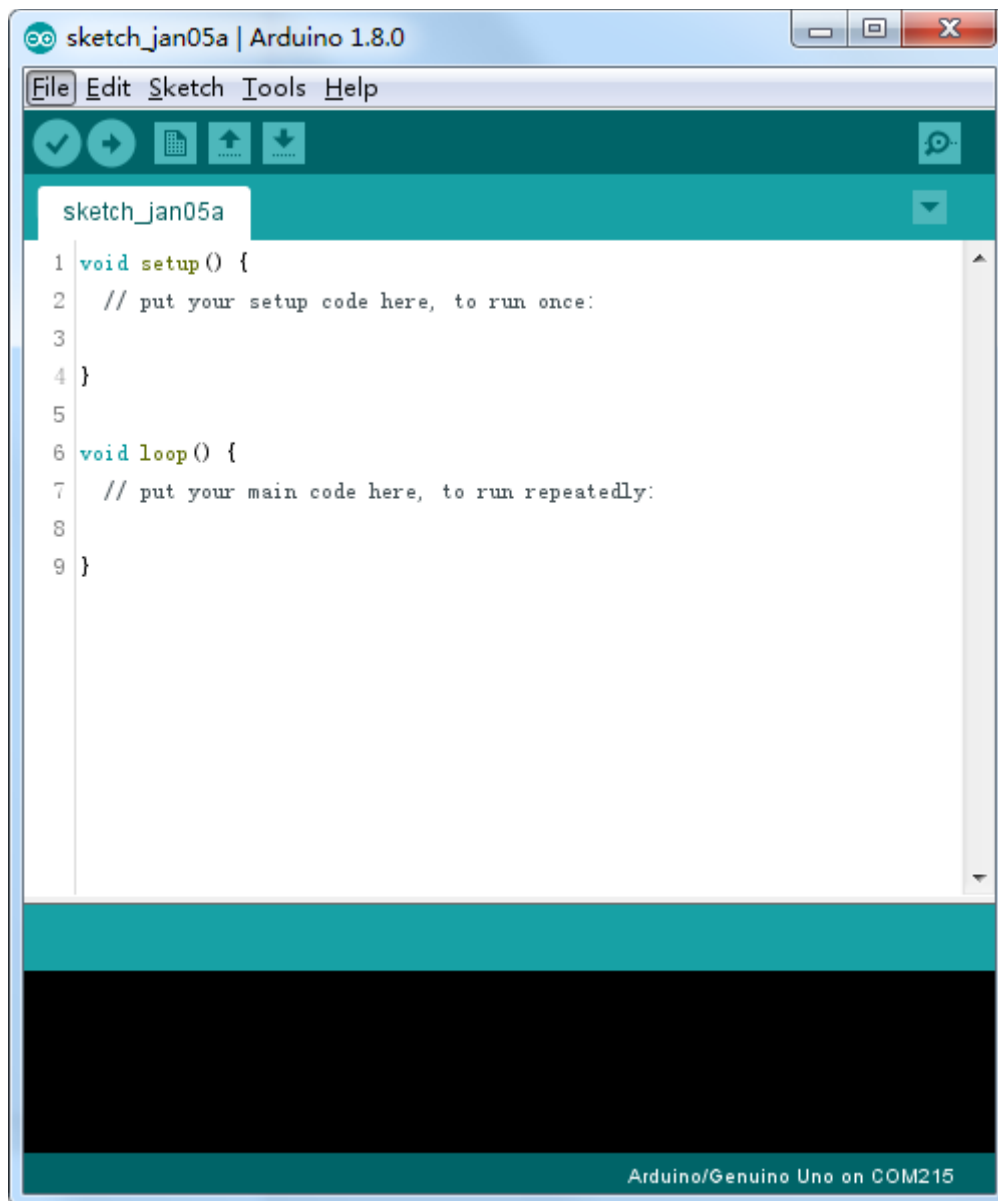
În cele din urmă, apare următoarea interfață, faceți clic *Instalare* pentru a finaliza instalarea.



Apoi, următoarea pictogramă apare pe desktop



Faceți dublu clic pentru a intra în mediul de dezvoltare dorit



Dacă alegeți APK-ul pentru a instala direct, vă rugăm să omiteți conținutul următor și să continuați lecția următoare. Dar dacă doriți să aflați cealaltă metodă de instalare, cu excepția APK-ului, vă rugăm să citiți acest cuvânt.

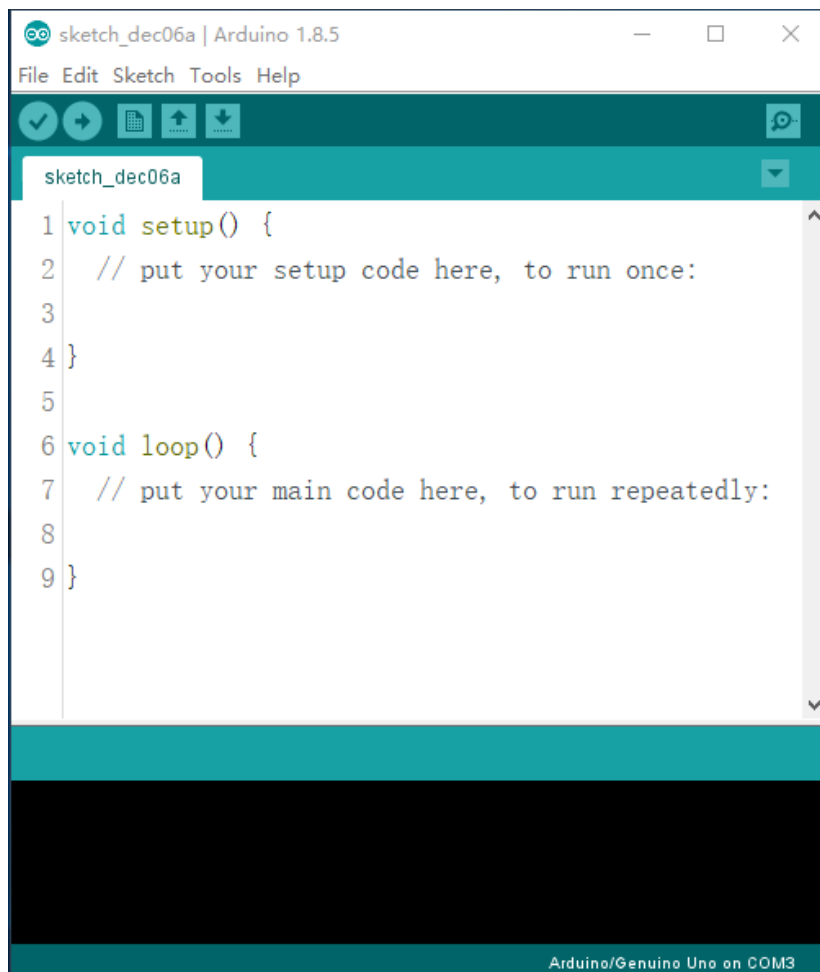
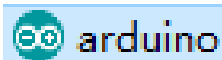
Dezarhivați fișierul zip descărcat, faceți dublu clic pe program și introduceți IDE-ul de ce aveți nevoie.



arduino-1.8.5-windows

此电脑 > Storage (D:) > arduino 1.8.5 > arduino-1.8.5-windows > arduino-1.8.5

| 名称                | 修改日期            | 类型     | 大小       |
|-------------------|-----------------|--------|----------|
| drivers           | 2017/12/6 10:15 | 文件夹    |          |
| examples          | 2017/12/6 10:15 | 文件夹    |          |
| hardware          | 2017/12/6 10:15 | 文件夹    |          |
| java              | 2017/12/6 10:15 | 文件夹    |          |
| lib               | 2017/12/6 10:15 | 文件夹    |          |
| libraries         | 2017/12/6 10:15 | 文件夹    |          |
| reference         | 2017/12/6 10:15 | 文件夹    |          |
| tools             | 2017/12/6 10:15 | 文件夹    |          |
| tools-builder     | 2017/12/6 10:15 | 文件夹    |          |
| arduino           | 2017/10/2 15:37 | 应用程序   | 395 KB   |
| arduino.l4j       | 2017/10/2 15:37 | 配置设置   | 1 KB     |
| arduino_debug     | 2017/10/2 15:37 | 应用程序   | 393 KB   |
| arduino_debug.l4j | 2017/10/2 15:37 | 配置设置   | 1 KB     |
| arduino-builder   | 2017/10/2 15:37 | 应用程序   | 3,214 KB |
| libusb0.dll       | 2017/10/2 15:37 | 应用程序扩展 | 43 KB    |
| msvcp100.dll      | 2017/10/2 15:37 | 应用程序扩展 | 412 KB   |
| msvcr100.dll      | 2017/10/2 15:37 | 应用程序扩展 | 753 KB   |
| revisions         | 2017/10/2 15:37 | 文本文档   | 84 KB    |
| wrapper-manifest  | 2017/10/2 15:37 | XML 文档 | 1 KB     |



Cu toate acestea, această metodă de instalare necesită instalarea separată a driverului.

Folderul Arduino conține atât programul Arduino în sine, cât și driverele care permit conectarea Arduino la computer printr-un cablu USB. Înainte de a lansa software-ul Arduino, veți instala driverele USB.

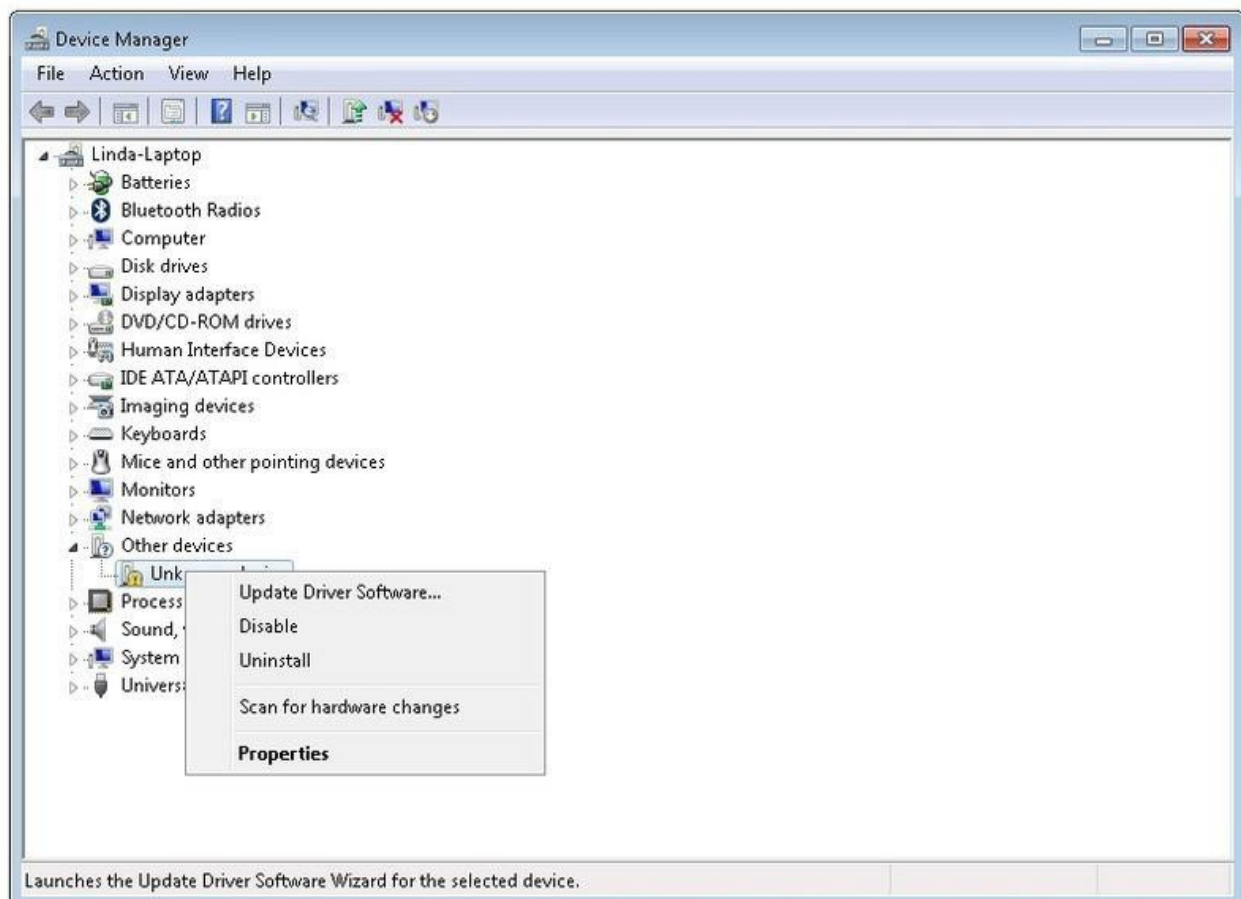
Conectați un capăt al cablului USB la Arduino și celălalt la o priză USB de pe computer.

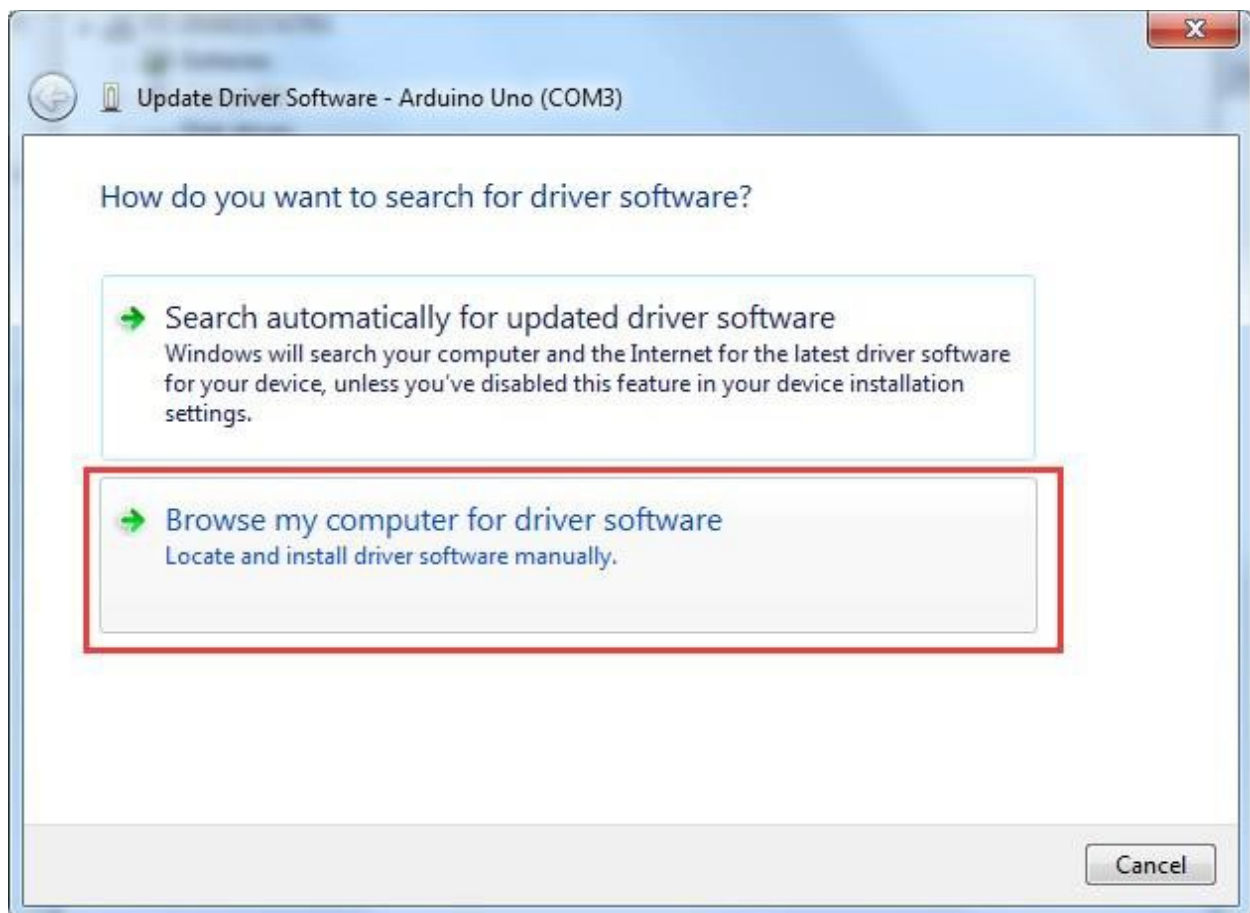
Ledul de alimentare de pe LED se va aprinde și este posibil să primiți mesajul

„Hardware nou găsit” de la Windows. Ignorați acest mesaj și anulați orice încercare pe care Windows le face de a încerca să instaleze drivere automat pentru dvs.

Cea mai fiabilă metodă de instalare a driverelor USB este utilizarea Device Manager. Acesta este accesat în moduri diferite, în funcție de versiunea dvs. de Windows. În Windows 7, mai întâi trebuie să deschideți Panoul de control, apoi să selectați opțiunea de vizualizare a pictogramelor și ar trebui să găsiți Managerul de dispozitive în listă.

Sub „Alte dispozitive”, ar trebui să vedeți o pictogramă pentru „dispozitiv necunoscut” cu un mic triunghi galben de avertizare lângă ea. Acesta este Arduino dvs.





Faceți clic dreapta pe dispozitiv și selectați opțiunea din meniul de sus (Actualizare software driver...). Apoi vi se va solicita fie „Căutați automat software-ul de driver actualizat”, fie „Răsfoiți computerul pentru software-ul de driver”. Selectați opțiunea pentru a naviga și a naviga la X\arduino1.8.0\driver.



Faceți clic pe „Următorul” și este posibil să primiți un avertisment de securitate, dacă da, permiteți instalarea software-ului. Odată ce software-ul a fost instalat, veți primi un mesaj de confirmare.



Utilizatorii Windows pot sări peste instrucțiunile de instalare pentru sistemele Mac și Linux și să sară la Lecția 1. Utilizatorii Mac și Linux pot continua să citească această secțiune.

## Instalarea Arduino (Mac OS X)

### Pasul 1: Descărcați software-ul Arduino (IDE)

- Deschide URL-ul: <https://www.arduino.cc/en/Main/Software> cu browser



- Faceți clic pe Mac OSX10.7 Lion sau mai nou

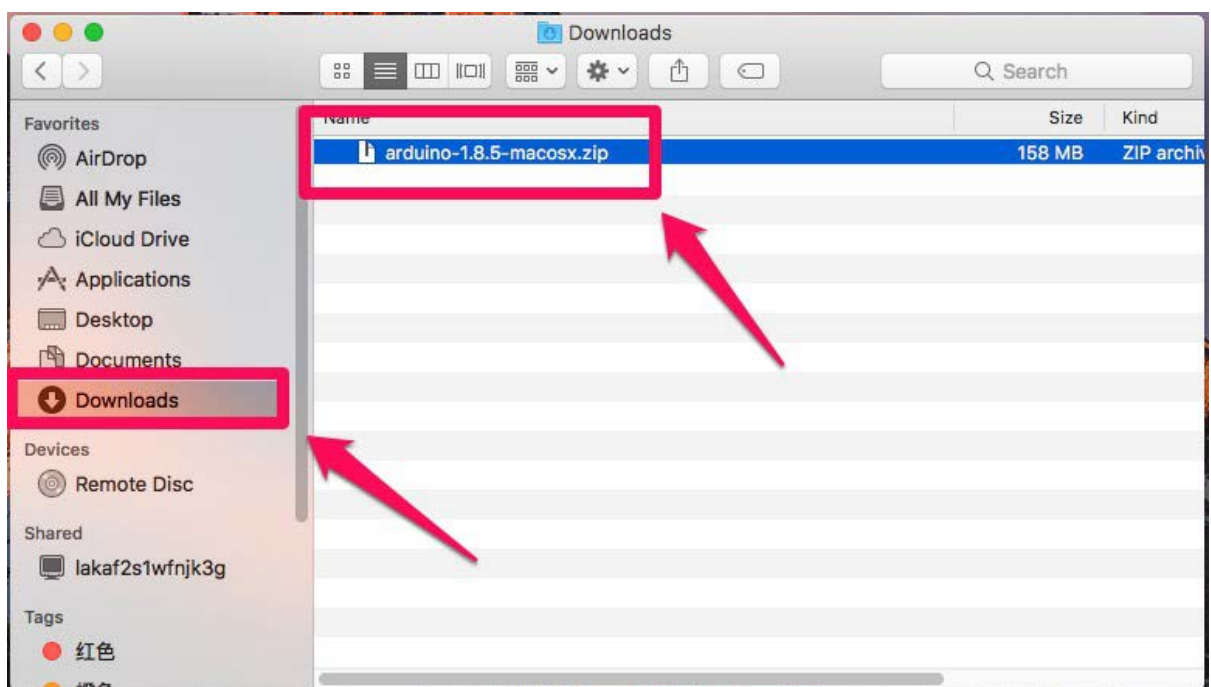
- Faceți clic DOAR DESCARCĂ

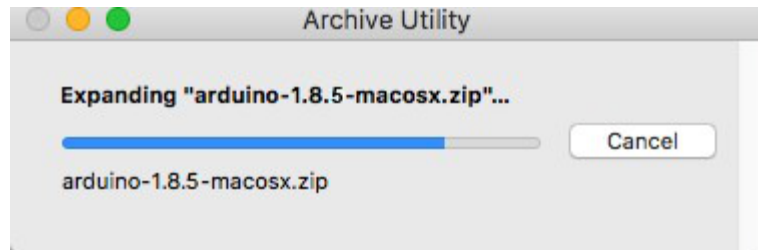
## Support the Arduino Software

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.

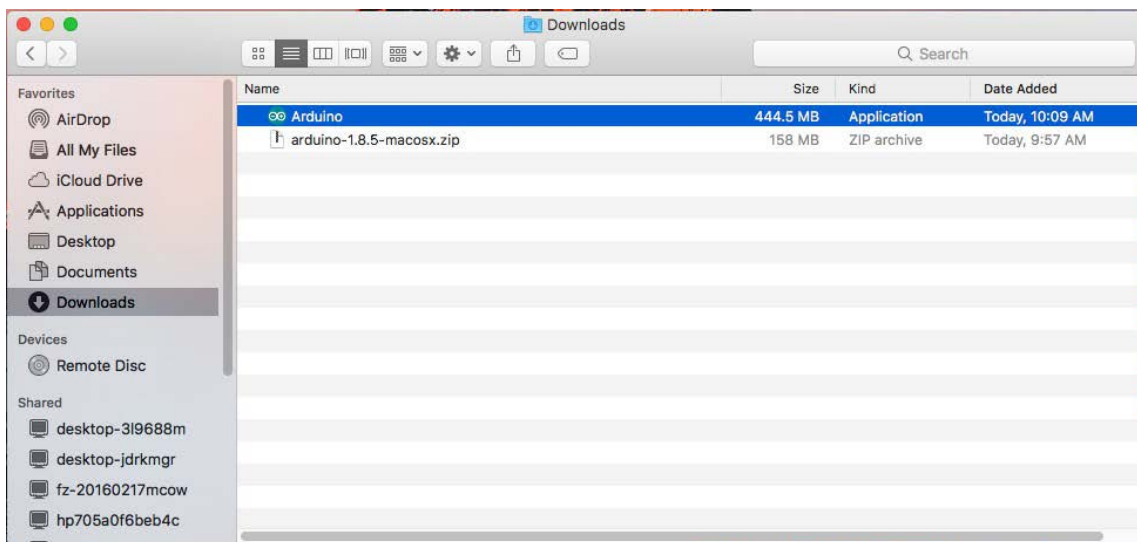


Descărcați și dezarhivați fișierul zip, faceți dublu clic pe Arduino.app pentru a intra în Arduino IDE; sistemul vă va cere să instalați biblioteca de rulare Java dacă nu o aveți în dvs calculator. Odată ce instalarea este finalizată, puteți rula IDE-ul Arduino.

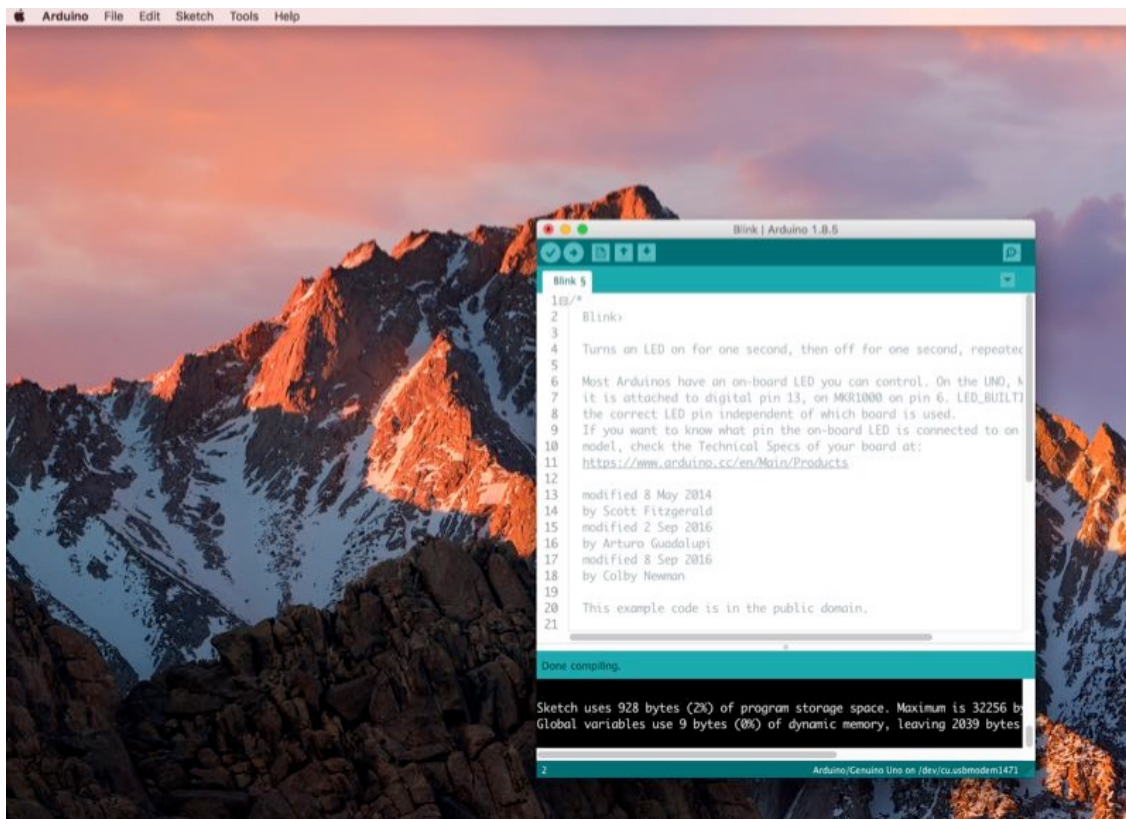




- Fișierul arduino este aplicația Arduino, doar faceți dublu clic pentru a-l deschide



- Arduino IDE

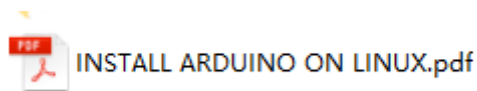


## Instalarea Arduino (Linux)

Va trebui să utilizați comanda `make install`. Dacă utilizați sistemul Ubuntu, așa este recomandat să instalați Arduino IDE din centrul de software al Ubuntu.

 `arduino-1.8.5-linux32.tar.xz`  
 `arduino-1.8.5-linux64.tar.xz`

**SFATURI:** Dacă întâmpinați probleme la instalarea driverelor, vă rugăm să consultați [Întrebările frecvente privind driverele UNO R3, MEGA, NANO](#).



## Lecția 1 Adăugați biblioteci și deschideți SerialMonitor

### Instalarea bibliotecilor Arduino suplimentare

Odată ce vă simțiți confortabil cu software-ul Arduino și utilizați funcțiile încorporate, poate doriți să extindeți capacitatea Arduino cu biblioteci suplimentare.

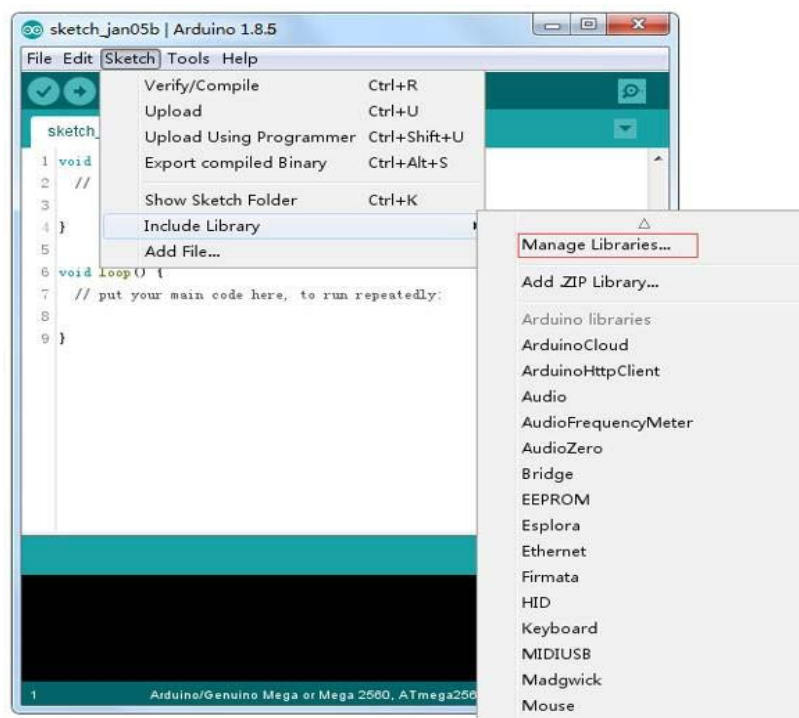
### Ce sunt Bibliotecile?

Bibliotecile sunt o colecție de coduri care vă facilitează conectarea la un senzor, afișaj, modul etc. De exemplu, biblioteca încorporată cu cristale lichide facilitează comunicarea cu afișajele LCD cu caractere. Există sute de biblioteci suplimentare disponibile pe Internet pentru descărcare. Bibliotecile încorporate și unele dintre aceste biblioteci suplimentare sunt listate în referință. Pentru a utiliza bibliotecile suplimentare, va trebui să le instalați.

### Cum se instalează o bibliotecă

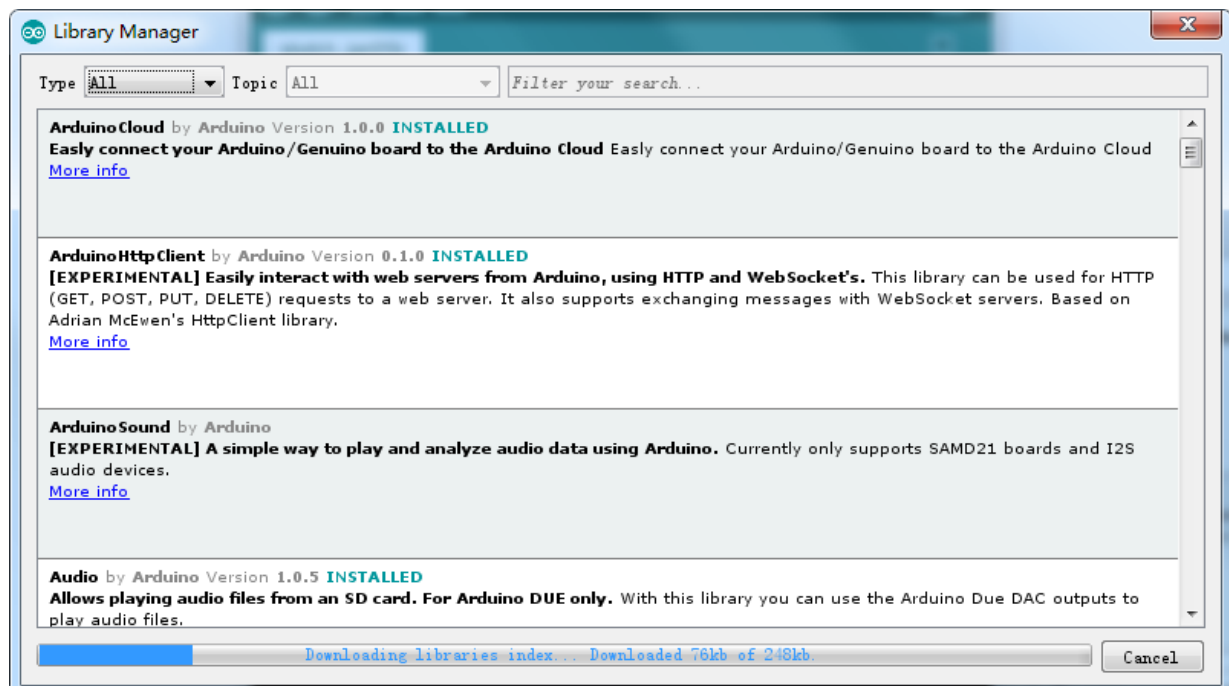
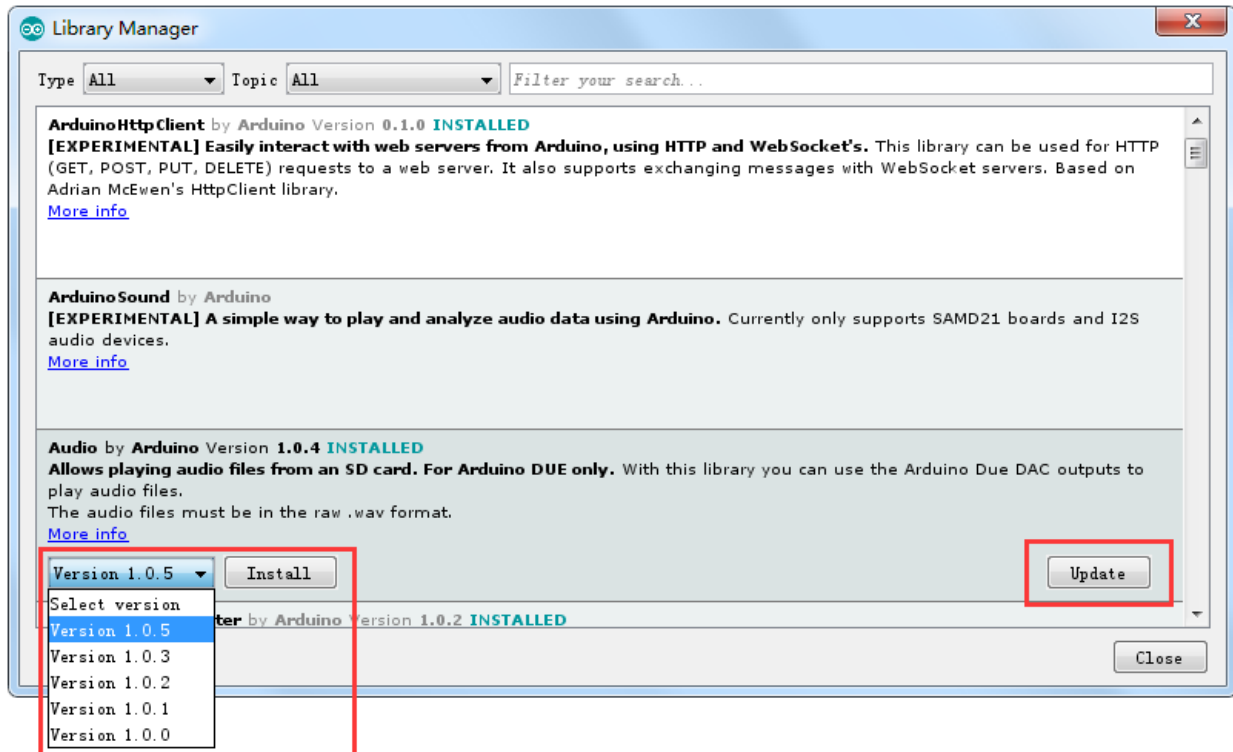
Folosind Managerul bibliotecii

Pentru a instala o nouă bibliotecă în IDE-ul tău Arduino, poți folosi Library Manager (disponibil din versiunea IDE 1.8.5). Deschideți IDE-ul și faceți clic pe meniul „Schiță”, apoi Includeți biblioteca > Gestionată bibliotecile.

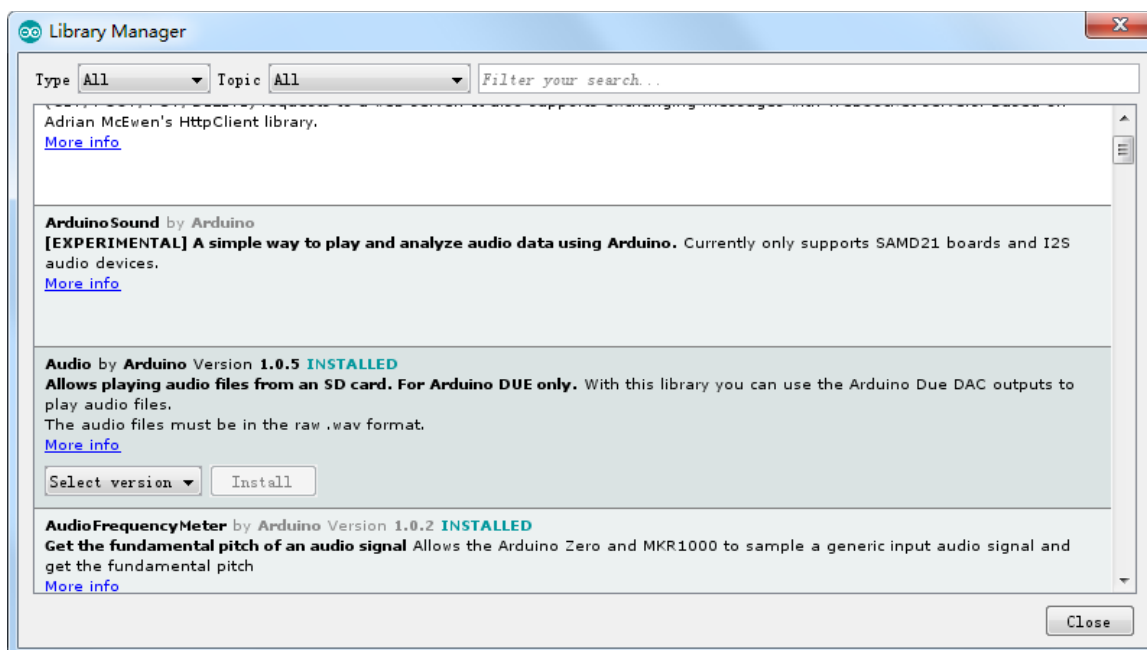


Apoi managerul de bibliotecă se va deschide și veți găsi o listă de biblioteci care sunt deja instalate sau gata de instalare. În acest exemplu vom instala biblioteca Bridge. Derulați lista pentru a o găsi, apoi selectați versiunea bibliotecii pe care doriți să o instalați. Uneori este disponibilă o singură versiune a bibliotecii. Dacă nu apare meniul de selectare a versiunii, nu vă faceți griji: este normal.

Există momente în care trebuie să aveți răbdare cu el, așa cum se arată în figură. Vă rugăm să o reîmprospătați și așteptați.



În cele din urmă faceți clic pe instalare și așteptați ca IDE-ul să instaleze noua bibliotecă. Descărcarea poate dura timp în funcție de viteza conexiunii. Odată ce s-a terminat, o etichetă Instalată ar trebui să apară lângă biblioteca Bridge. Puteți închide managerul bibliotecii.

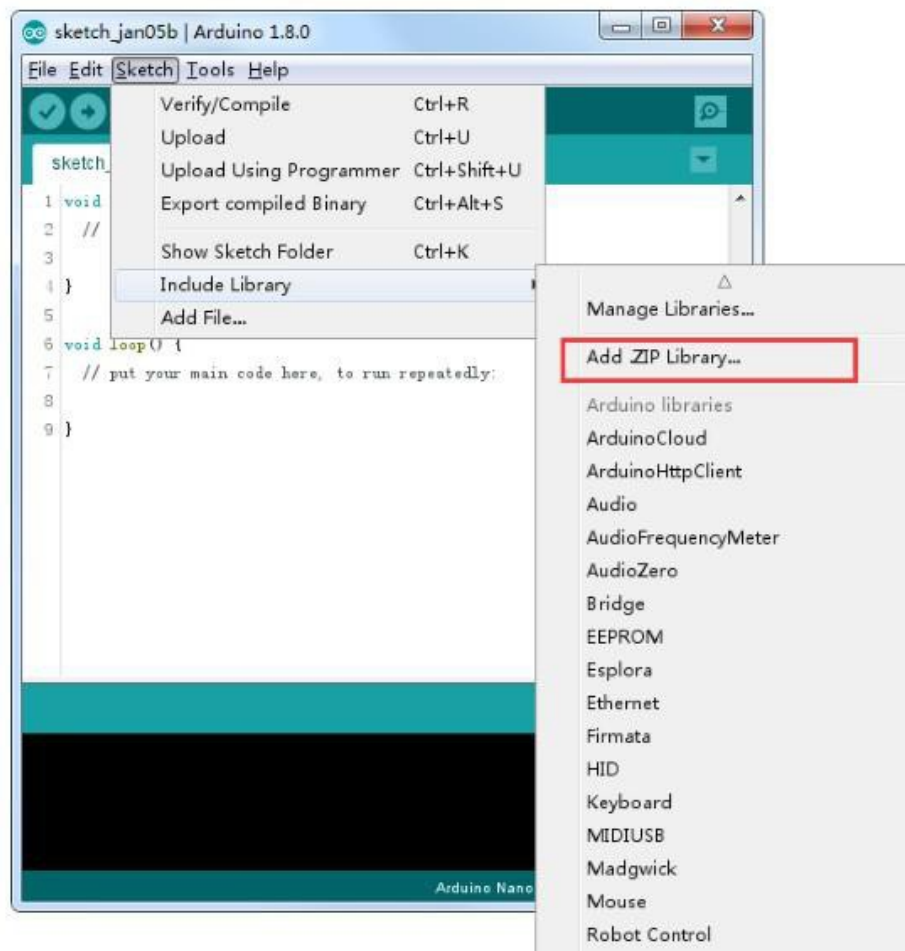


Acum puteți găsi noua bibliotecă disponibilă în meniul Includere bibliotecă. Dacă doriți să adăugați propria bibliotecă, deschideți o nouă problemă pe [Github](#).

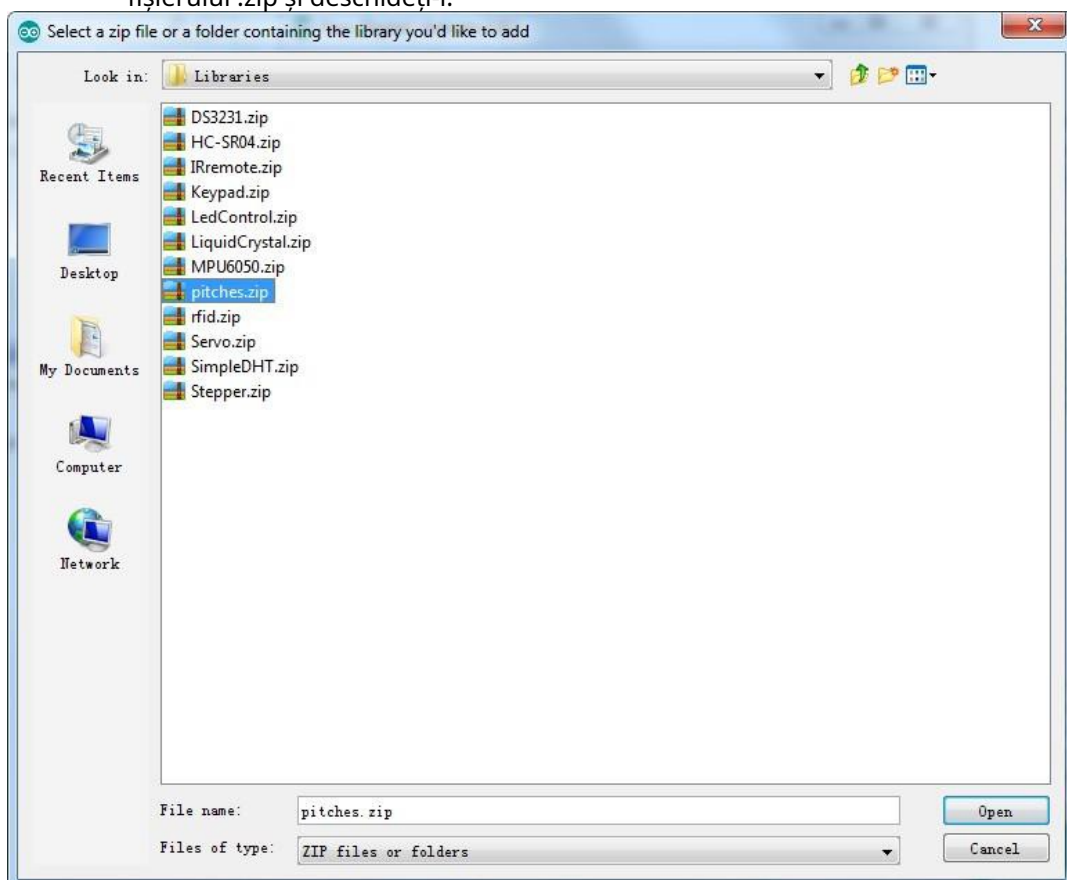
### Importul unei biblioteci .zip

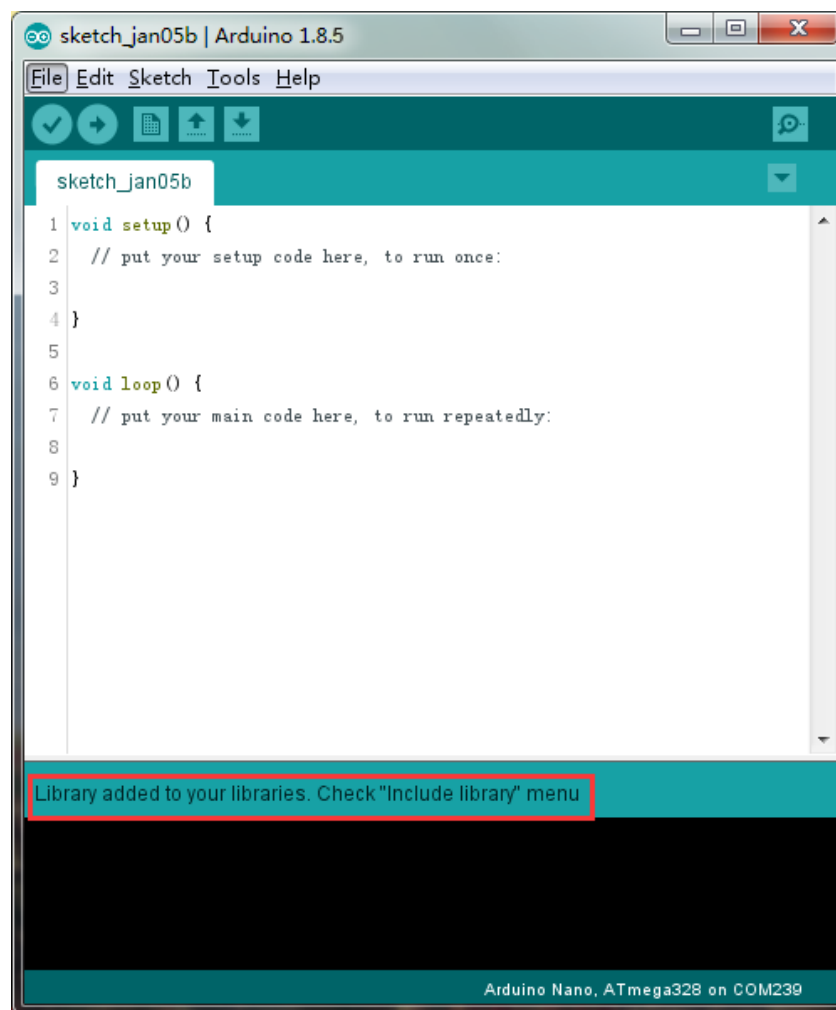
Bibliotecile sunt adesea distribuite ca fișier ZIP sau folder. Numele folderului este numele bibliotecii. În interiorul folderului va fi un fișier .cpp, un fișier .h și adesea un fișier keywords.txt, un folder de exemple și alte fișiere necesare bibliotecii. Începând cu versiunea 1.0.5, puteți instala biblioteci terță parte în IDE. Nu dezarhivați biblioteca descărcată, lăsați-o așa cum este.

În Arduino IDE, navigați la Sketch > Include Library. În partea de sus a listei derulante, selectați opțiunea „Adăugați .ZIPLibrary”.



Vi se va solicita să selectați bibliotecă pe care doriți să o adăugați. Navigați la locația fișierului .zip și deschideți-l.





Reveniți la meniul Schiță > Import bibliotecă. Ar trebui să vedeți acum biblioteca în partea de jos a meniului derulant. Este gata pentru a fi folosit în schița dvs. Fișierul zip va fi extins în folderul bibliotecii din directorul de schițe Arduino. **NB: Biblioteca va fi disponibilă pentru utilizare în schițe, dar exemplele pentru bibliotecă nu vor fi expuse în Fișier > Exemple decât după repornirea IDE-ului.**

Aceste două sunt cele mai comune abordări. Sistemele MAC și Linux pot fi gestionate la fel. Instalarea manuală care urmează să fie introdusă mai jos ca alternativă poate fi folosită rar, iar utilizatorii fără nevoi o pot sări peste ea.

## Instalare manuală

Pentru a instala biblioteca, închideți mai întâi aplicația Arduino. Apoi decompriați fișierul ZIP care conține biblioteca. De exemplu, dacă instalați o bibliotecă numită

„ArduinoParty”, decompriți ArduinoParty.zip. Ar trebui să conțină un folder numit ArduinoParty, cu fișiere precum ArduinoParty.cpp și ArduinoParty.h. (Dacă fișierele .cpp și .h nu se află într-un folder, va trebui să creați unul. În acest caz, ați crea un folder numit "ArduinoParty" și veți muta în el toate fișierele care erau în ZIP. fișier, cum ar fi ArduinoParty.cpp și ArduinoParty.h.)

Trageți folderul ArduinoParty în acest folder (dosarul bibliotecile dvs.). În Windows, se va numi probabil „My Documents\Arduino\libraries”. Pentru utilizatorii de Mac, acesta se va numi probabil „Documente/Arduino/biblioteci”. Pe Linux, acesta va fi folderul „biblioteci” din caietul de schițe.

Dosarul bibliotecii Arduino ar trebui să arate acum astfel (pe Windows):

Documentele mele\Arduino\libraries\ArduinoParty\ArduinoParty.cpp

Documentele mele\Arduino\libraries\ArduinoParty\ArduinoParty.h

Documentele mele\Arduino\libraries\ArduinoParty\exemple

sau așa (pe Mac și Linux):

Documente/Arduino/libraries/ArduinoParty/ArduinoParty.cpp

Documente/Arduino/libraries/ArduinoParty/ArduinoParty.h

Documente/Arduino/libraries/ArduinoParty/exemple

....

Este posibil să existe mai multe fișiere decât fișierele .cpp și .h, doar asigurați-vă că sunt toate acolo. (Biblioteca nu va funcționa dacă puneți fișierele .cpp și .h direct în folderul biblioteci sau dacă sunt imbricate într-un folder suplimentar. De exemplu:

Documents\Arduino\libraries\ArduinoParty.cpp și

Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp nu va funcționa.)

Reporniți aplicația Arduino. Asigurați-vă că noua bibliotecă apare în Sketch-> Elementul de meniu Import Library al software-ului. Asta este! Ai instalat o bibliotecă!

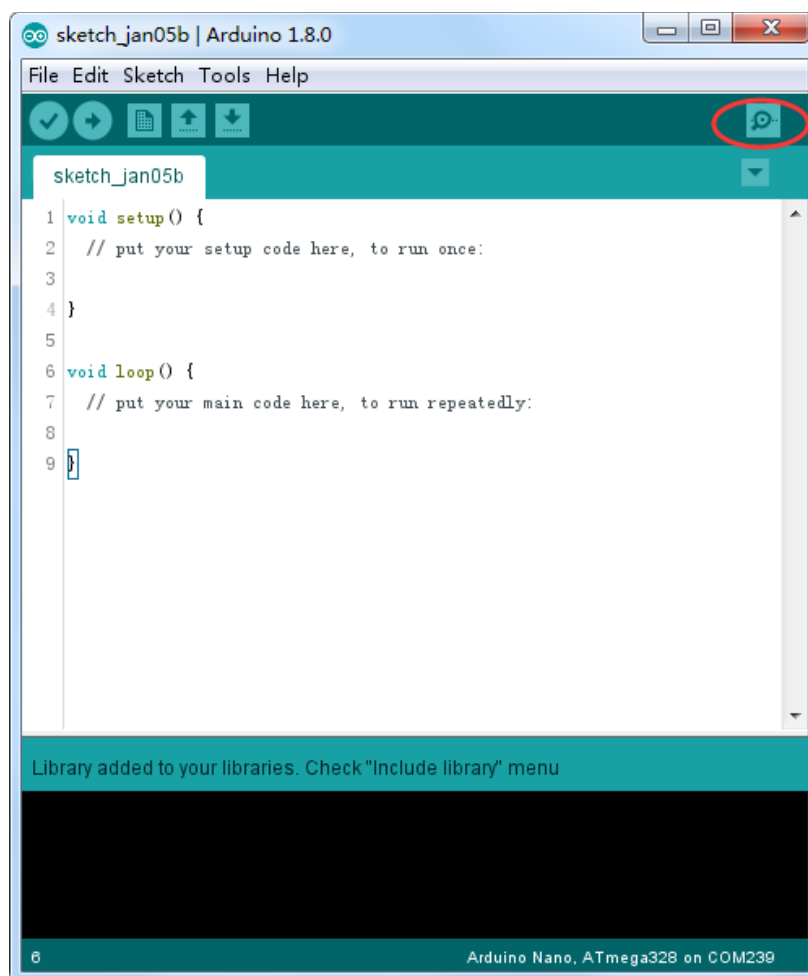
## Monitor serial Arduino (Windows, Mac, Linux)

Arduino Integrated Development Environment (IDE) este partea software a platformei Arduino. Și, pentru că utilizarea unui terminal este o parte atât de mare a lucrului cu el

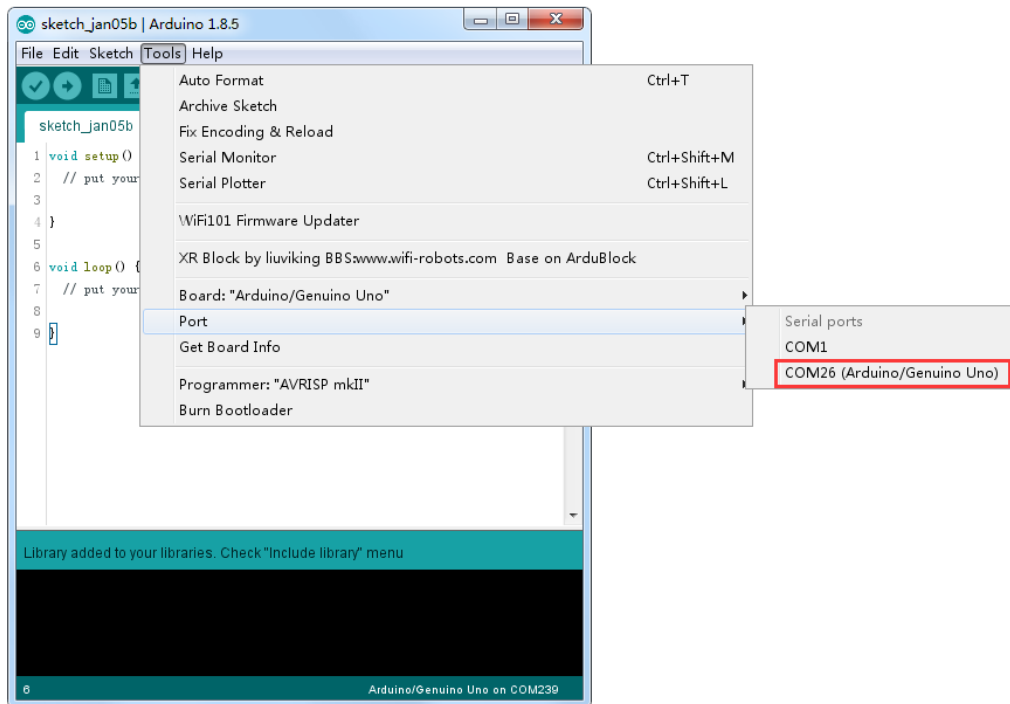
Arduinos și alte microcontrolere, au decis să includă un terminal serial cu software-ul. În mediul Arduino, acesta se numește Monitor serial.

### Realizarea unei conexiuni

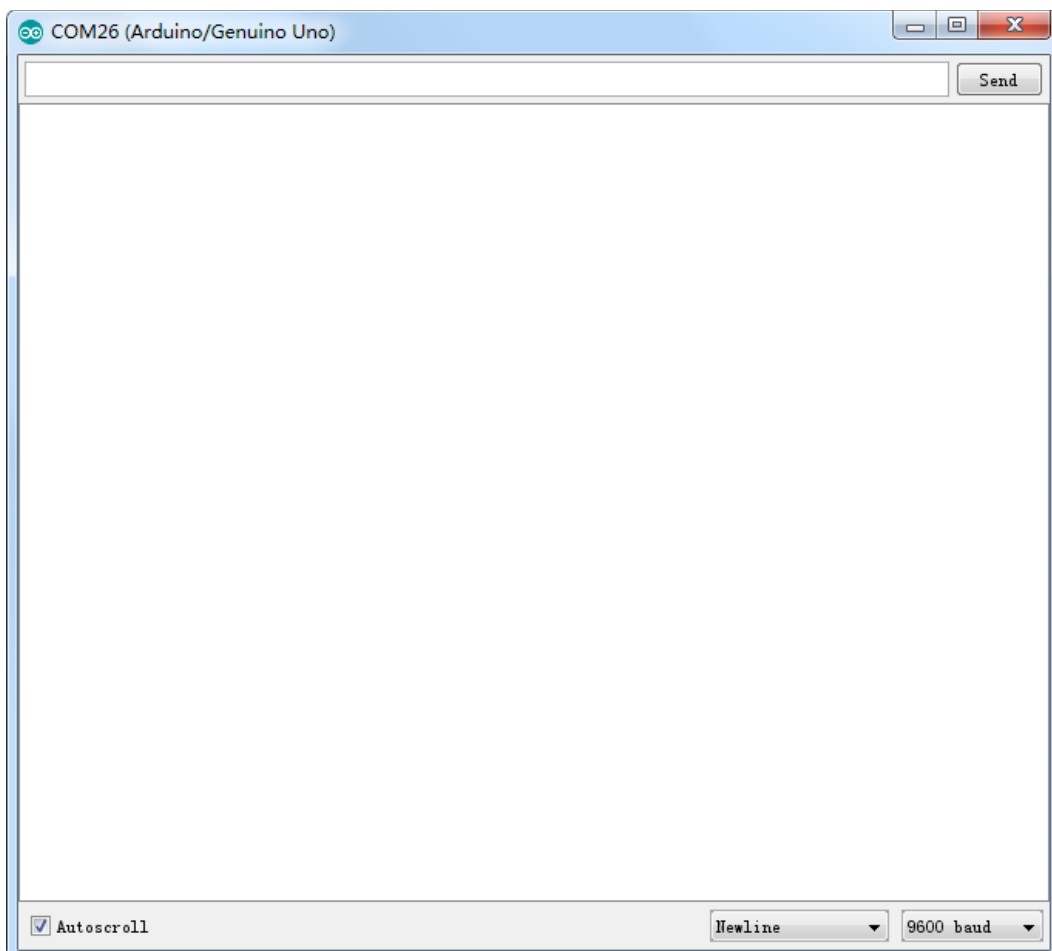
Monitorul serial vine cu orice versiune a IDE-ului Arduino. Pentru a-l deschide, faceți clic pe pictograma Monitor serial.



Selectarea portului de deschis în monitorul serial este la fel cu selectarea unui port pentru încărcarea codului Arduino. Accesați Instrumente -> Port serial și selectați portul corect. **Sfaturi:** Alegeți același port COM pe care îl aveți în DeviceManager.

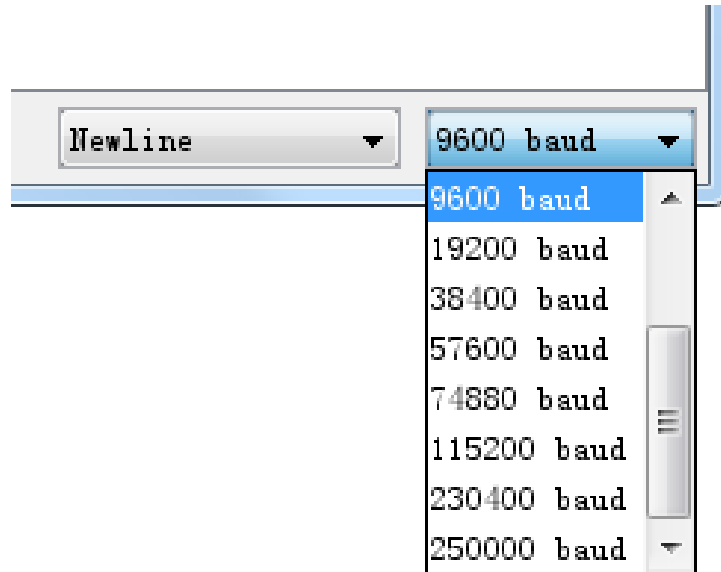


Odată deschis, ar trebui să vedeți ceva de genul:

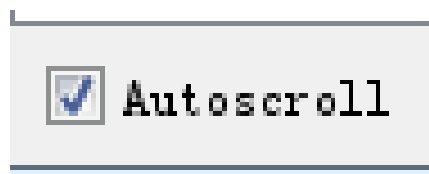


## Setări

Monitorul serial are setări limitate, dar suficiente pentru a face față majorității nevoilor dvs. de comunicare în serie. Prima setare pe care o puteți modifica este viteza de transmisie. Faceți clic pe meniul derulant al vitezei de transmisie pentru a selecta viteza de transmisie corectă. (9600 baud)



În sfârșit, puteți seta terminalul la Autoscroll sau nu bifând caseta din colțul din stânga jos.



## Pro

Monitorul serial este o modalitate rapidă și ușoară de a stabili o conexiune serială cu Arduino. Dacă lucrați deja în IDE-ul Arduino, chiar nu este nevoie să deschideți un terminal separat pentru a afișa date.

## Contra

Lipsa setărilor lasă mult de dorit în monitorul serial și, pentru comunicațiile seriale avansate, s-ar putea să nu facă truc.

## Lecția 2 Clipește

### Prezentare generală

În această lecție, veți învăța cum să programați placa de control UNO R3 pentru a clipi LED-ul încorporat al Arduino și cum să descărcați programe prin pașii de bază.

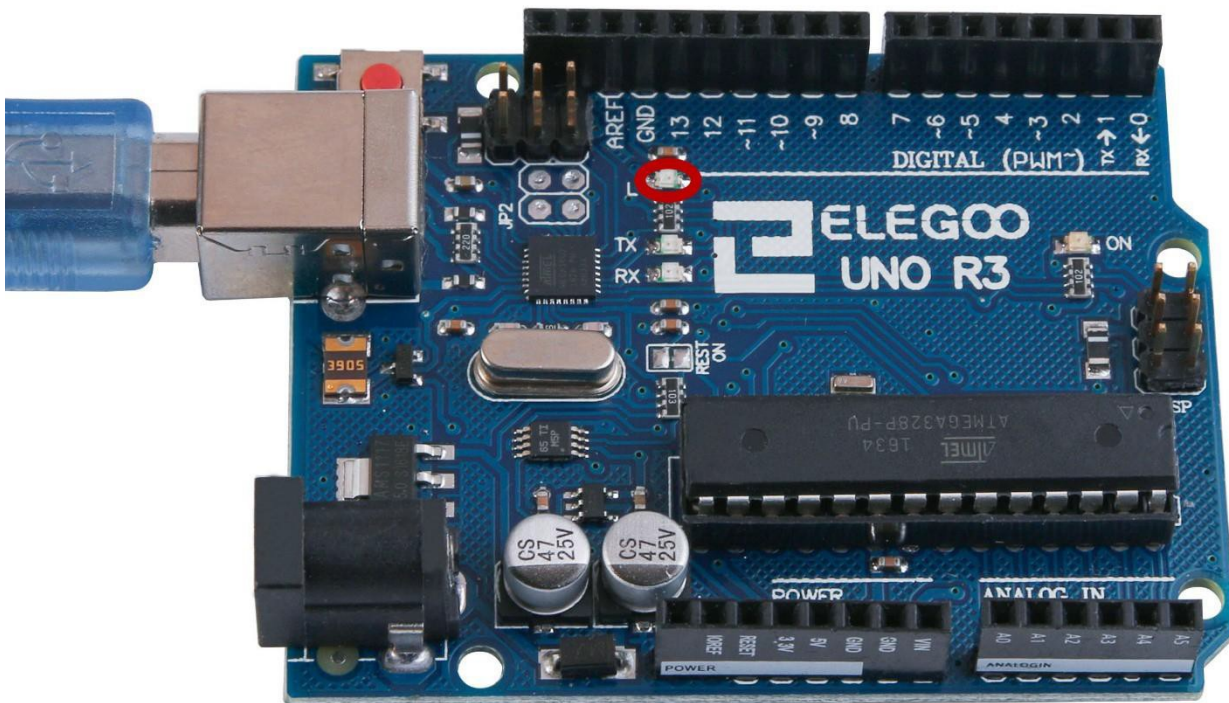
### Componentă necesară:

(1) x Elegoo UnoR3

### Principiu

Placa UNO R3 are șiruri de conectori de-a lungul ambelor părți care sunt utilizați pentru a se conecta la mai multe dispozitive electronice și „scuturi” plug-in care îi extind capacitatea.

De asemenea, are un singur LED pe care îl puteți controla din schițele dvs. Acest LED este construit pe placa UNO R3 și este adesea denumit LED-ul „L”, deoarece așa este etichetat pe placă.



Este posibil să descoperiți că LED-ul „L” al plăcii dumneavoastră UNO R3 clipește deja când îl conectați la o mufă USB. Acest lucru se datorează faptului că plăcile sunt în general livrate cu schița „Blink” preinstalată.

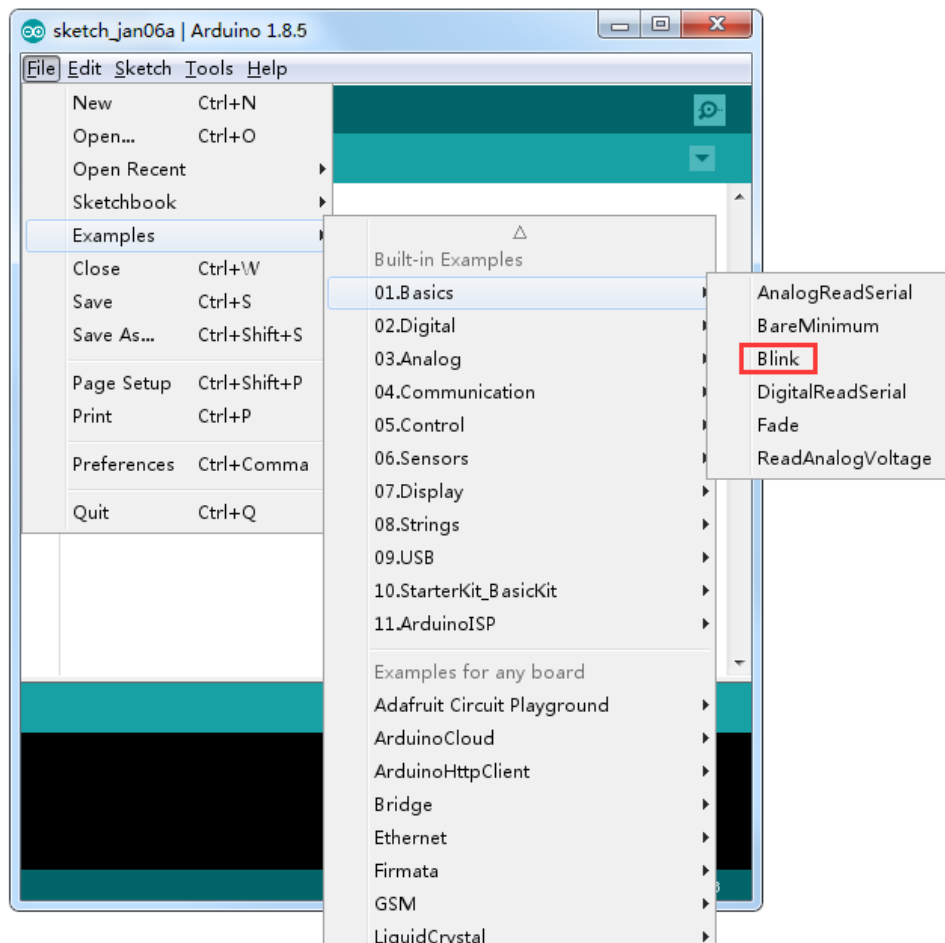
În această lecție, vom reprograma placa UNO R3 cu propria noastră schiță Blink și apoi vom schimba viteza cu care clipește.

În lecția 0, v-ați configurat Arduino IDE și v-ați asigurat că puteți găsi portul serial potrivit pentru a se conecta la placa UNO R3. Acum a sosit momentul să puneți acea conexiune la test și să vă programați placa UNO R3.

Arduino IDE include o colecție mare de exemple de schițe pe care le puteți încărca și utiliza.

Aceasta include un exemplu de schiță pentru a face LED-ul „L” să clipească.

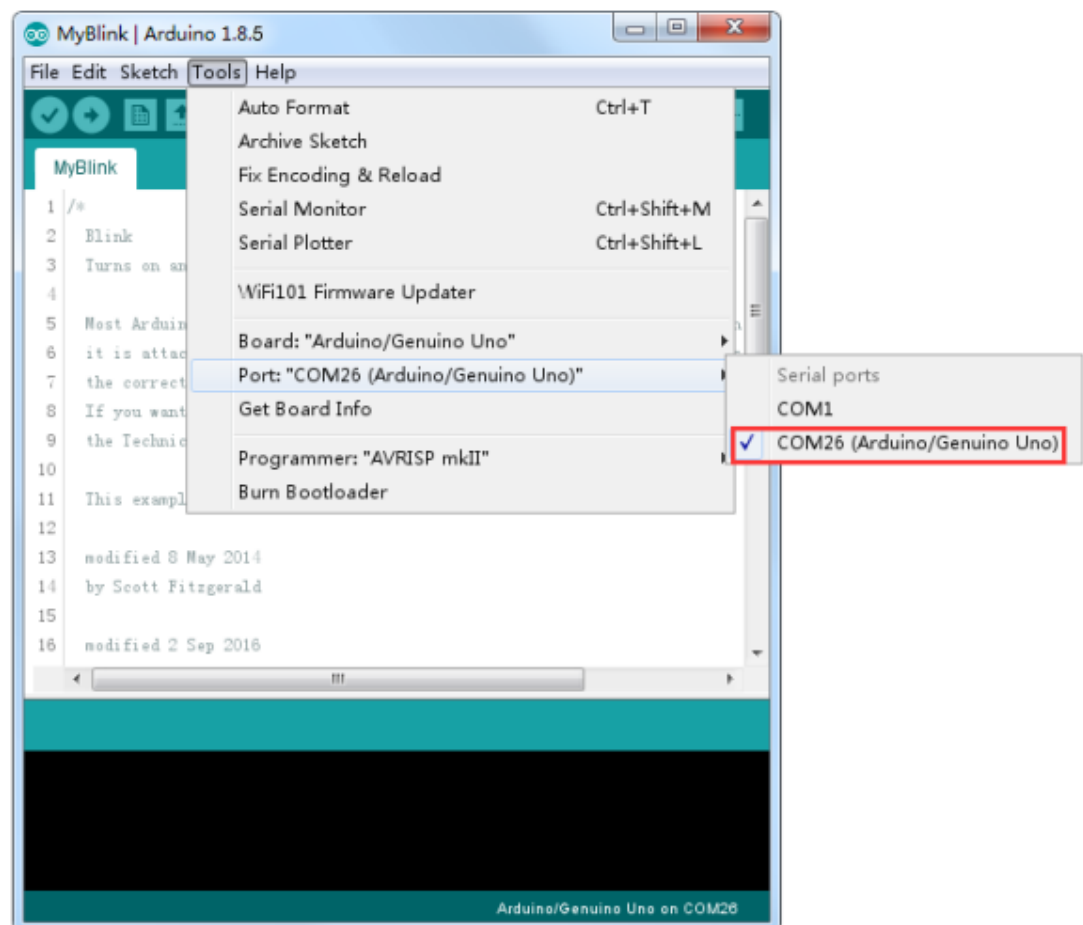
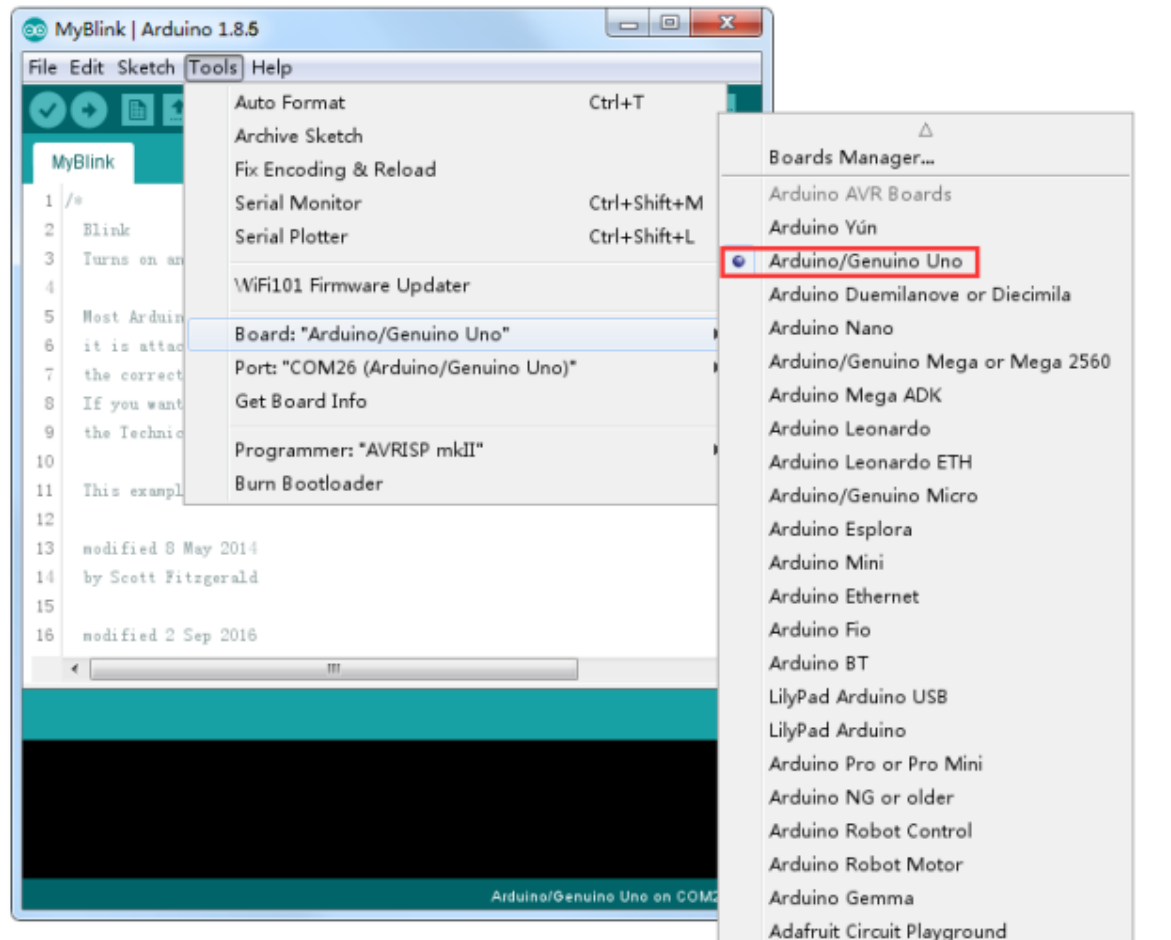
Încărcați schița „Blink” pe care o veți găsi în sistemul de meniuri al IDE-ului sub Fișier > Exemple > 01.



Când se deschide fereastra de schiță, mărește-o astfel încât să poți vedea întreaga schiță în fereastră.

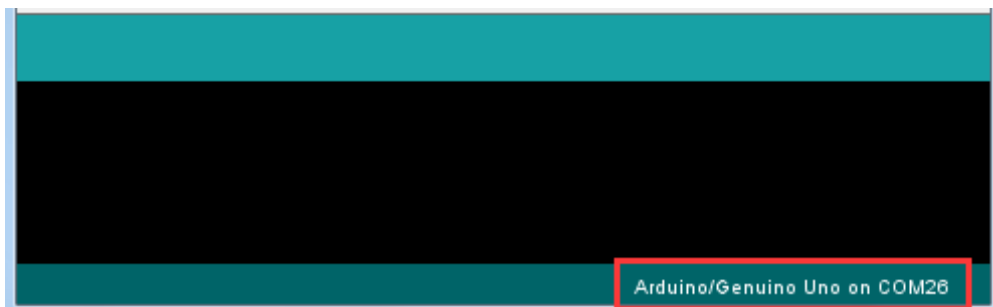


Atașați placa Arduino la computer cu cablul USB și verificați dacă „Tipul plăcii” și „Portul serial” sunt setate corect.



**Notă: Tipul plăcii și portul serial de aici nu sunt neapărat aceleași ca cele prezentate în imagine. Dacă utilizați 2560, atunci va trebui să alegeți Mega 2560 ca tip de placă, alte alegeri pot fi făcute în același mod. Și portul serial afișat pentru toată lumea este diferit, în ciuda COM 26 ales aici, ar putea fi COM3 sau COM4 pe computer. Un port COM corect ar trebui să fie COMX (arduino XXX), care este conform criteriilor de certificare.**

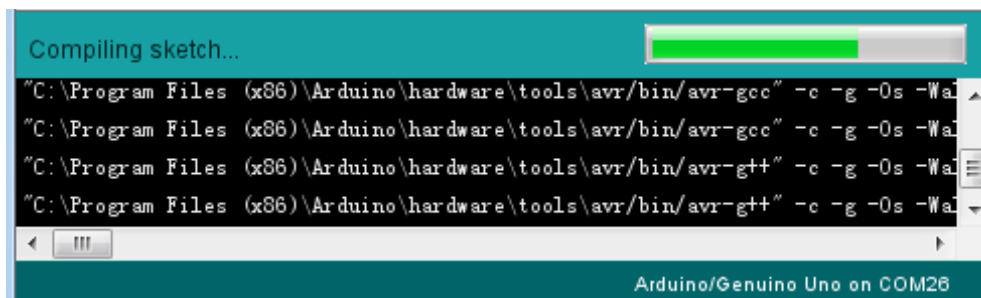
Arduino IDE vă va arăta setările curente pentru placă în partea de jos a ferestrei.



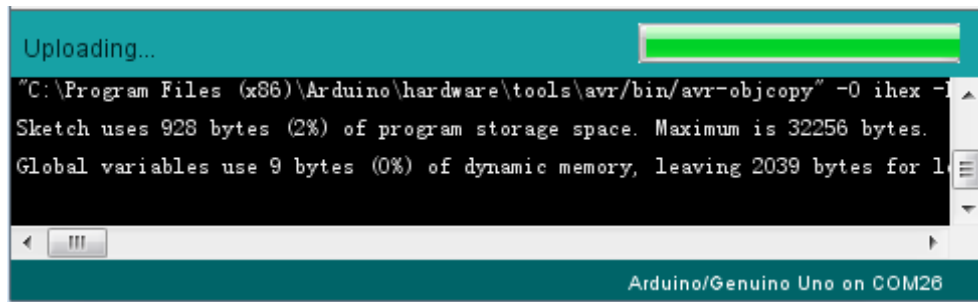
Faceți clic pe butonul „Încărcare”. Al doilea buton din stânga pe bara de instrumente.



Dacă urmăriți zona de stare a IDE-ului, veți vedea o bară de progres și o serie de mesaje. La început, va spune „Compilare schiță...”. Acest lucru transformă schița într-un format potrivit pentru încărcare pe tablă.



În continuare, starea se va schimba în „Încărcare”. În acest moment, LED-urile de pe Arduino ar trebui să înceapă să pâlpâie pe măsură ce schița este transferată.



În cele din urmă, starea se va schimba în „Terminat”.



Celălalt mesaj ne spune că schița folosește 928 de octeți din cei 32.256 de octeți disponibili. După etapa „Compilarea schiței..”, puteți primi următorul mesaj de eroare:



Poate însemna că placa dumneavoastră nu este conectată deloc, sau driverele nu au fost instalate (dacă este necesar) sau că este selectat portul serial greșit.

Dacă întâlniți acest lucru, reveniți la Lecția 0 și verificați instalarea. Odată ce încărcarea s-a încheiat, placa ar trebui să repornească și să înceapă să clipească. Deschide codul

Rețineți că o mare parte a acestei schițe este compusă din comentarii. Acestea nu sunt instrucțiuni de program reale; mai degrabă, ei explică doar cum funcționează programul. Sunt acolo pentru beneficiul tău.

Tot ceea ce este între /\* și \*/ din partea de sus a schiței este un comentariu de bloc; explică pentru ce este schița.

Comentariile pe o singură linie încep cu // și totul până la sfârșitul acelei rânduri este considerat un comentariu.

Prima linie de cod este:

```
int led = 13;
```

După cum explică comentariul de mai sus, acesta dă un nume pinului de care este atașat LED-ul. Acesta este 13 pe majoritatea Arduinos, inclusiv UNO și Leonardo.

În continuare, avem funcția „setup”. Din nou, după cum spune comentariul, acest lucru este executat atunci când este apăsat butonul de resetare. De asemenea, se execută ori de câte ori placa se resetează din orice motiv, cum ar fi alimentarea care i-a fost aplicată pentru prima dată sau după ce o schiță a fost încărcată.

```
void setup() {  
  // inițializați pinul digital ca ieșire.  
  pinMode(led, OUTPUT);  
}
```

Fiecare schiță Arduino trebuie să aibă o funcție de „configurare”, iar locul în care ați putea dori să adăugați instrucțiuni proprii este între { și }.

În acest caz, există o singură comandă acolo, care, după cum se arată în comentariu, spune plăcii Arduino că vom folosi pinul LED ca ieșire.

De asemenea, este obligatoriu ca o schiță să aibă o funcție „bucă”. Spre deosebire de funcția „setup” care rulează o singură dată, după o resetare, funcția „loop” va porni imediat din nou, după ce a terminat de rulat comenzile.

```
void loop() {  
  digitalWrite(led, HIGH);      // aprinde LED-ul (HIGH este nivelul de tensiune)  
  întârziere (1000);           // așteptați o secundă  
  digitalWrite(led, LOW);      // stinge LED-ul făcând voltage LOW  
  întârziere (1000);           // așteptați o secundă  
}
```

În interiorul funcției de buclă, comenzile pornesc mai întâi pinul LED-ului (HIGH), apoi „întârzie” timp de 1000 de milisecunde (1 secundă), apoi opresc pinul LED-ului și întrerupeți încă o secundă.

Acum veți face LED-ul să clipească mai repede. După cum probabil ați ghicit, cheia acestui lucru constă în schimbarea parametrului din () pentru comanda „întârziere”.

```
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 }
```

Această perioadă de întârziere este în milisecunde, așa că dacă doriți ca LED-ul să clipească de două ori mai repede, modificați valoarea de la 1000 la 500. Aceasta ar întrerupe apoi o jumătate de secundă pentru fiecare întârziere, mai degrabă decât o secundă întreagă.

Încărcați din nou schița și ar trebui să vedeți că LED-ul începe să clipească mai repede.

## Lecția 3 LED

### Prezentare generală

În această lecție, veți învăța cum să schimbați luminozitatea unui LED utilizând diferite valori ale rezistenței.

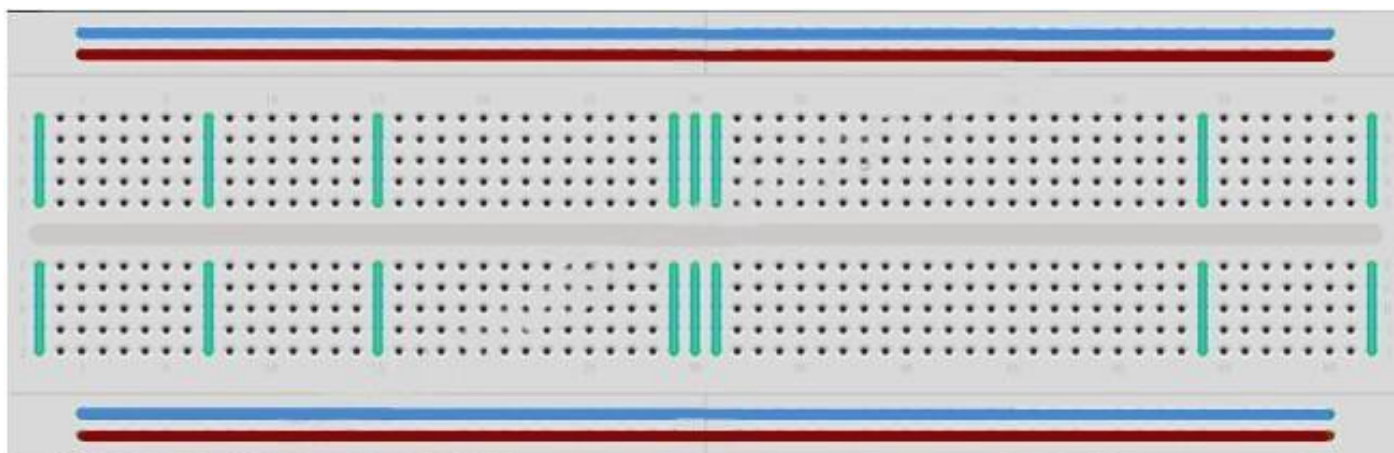
### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) LED roșu de 5 mm
- (1) x rezistor de 220 ohmi
- (1) x rezistență de 1k ohm
- (1) x rezistor de 10k ohmi
- (2) x fire MM (fire jumper de sex masculin la masculin)

### Introducere componente

#### TABLĂMB-102:

O placă de breadboard vă permite să prototipați circuite rapid, fără a fi nevoie să lipiți conexiunile. Mai jos este un exemplu.



Breadboard-urile vin în diferite dimensiuni și configurații. Cel mai simplu tip este doar o rețea de găuri într-un bloc de plastic. În interior sunt benzi de metal care asigură conexiunea electrică între găurile din rândurile mai scurte. Împingerea picioarelor a două componente diferite în același rând le unește electric. Un canal adânc care curge în mijloc indică faptul că există o întrerupere a conexiunilor acolo, ceea ce înseamnă că puteți împinge un cip cu picioarele de fiecare parte a canalului fără a le conecta împreună. Unele plăci au două benzi de găuri care trec de-a lungul marginilor lungi ale plăcii, care sunt separate de grila principală. Acestea au benzi care se desfășoară pe lungimea plăcii în interior și oferă o modalitate de a conecta o tensiune comună. Ele sunt de obicei în perechi pentru +5 volți și masă. Aceste benzi sunt denumite șine și vă permit să conectați alimentarea la multe componente sau puncte de pe placă.

În timp ce panourile sunt grozave pentru prototipare, ele au unele limitări. Deoarece conexiunile sunt împingere și temporare, nu sunt la fel de fiabile ca conexiunile lipite. Dacă aveți probleme intermitente cu un circuit, aceasta ar putea fi din cauza unei conexiuni proaste pe o placă.

### **LED:**

LED-urile fac lumini indicatoare grozave. Ei folosesc foarte puțină electricitate și durează aproape pentru totdeauna.

În această lecție, veți folosi poate cel mai comun dintre toate LED-urile: un LED roșu de 5 mm. 5 mm se referă la diametrul LED-ului. Alte dimensiuni comune sunt 3mm și 10mm. Nu puteți conecta direct un LED la o baterie sau la o sursă de tensiune, deoarece 1) LED-ul are un conductor pozitiv și unul negativ și nu se va aprinde dacă este plasat greșit și 2) un LED trebuie utilizat cu un rezistor pentru a limita sau „sufoca” cantitatea de curent care curge prin ea; altfel, se va arde!



Dacă nu utilizați un rezistor cu LED, atunci acesta poate fi distrus aproape imediat, deoarece prea mult curent va trece, încălzindu-l și distrugând „joncțiunea” în care este produsă lumina.

Există două moduri de a spune care este cablul pozitiv al LED-ului și care este negativul.

În primul rând, avantajul pozitiv este mai lung.

În al doilea rând, acolo unde conductorul negativ intră în corpul LED-ului, există o margine plată pe carcasa LED-ului.

Dacă se întâmplă să aveți un LED care are o latură plată lângă cablul mai lung, ar trebui să presupuneți că cablul mai lung este pozitiv.

### **REZISTENTE:**

După cum sugerează și numele, rezistențele rezistă fluxului de electricitate. Cu cât valoarea rezistorului este mai mare, cu atât rezistă mai mult și cu atât mai puțin curent electric va trece prin el. Vom folosi acest lucru pentru a controla cât de multă electricitate curge prin LED și, prin urmare, cât de puternic strălucește.

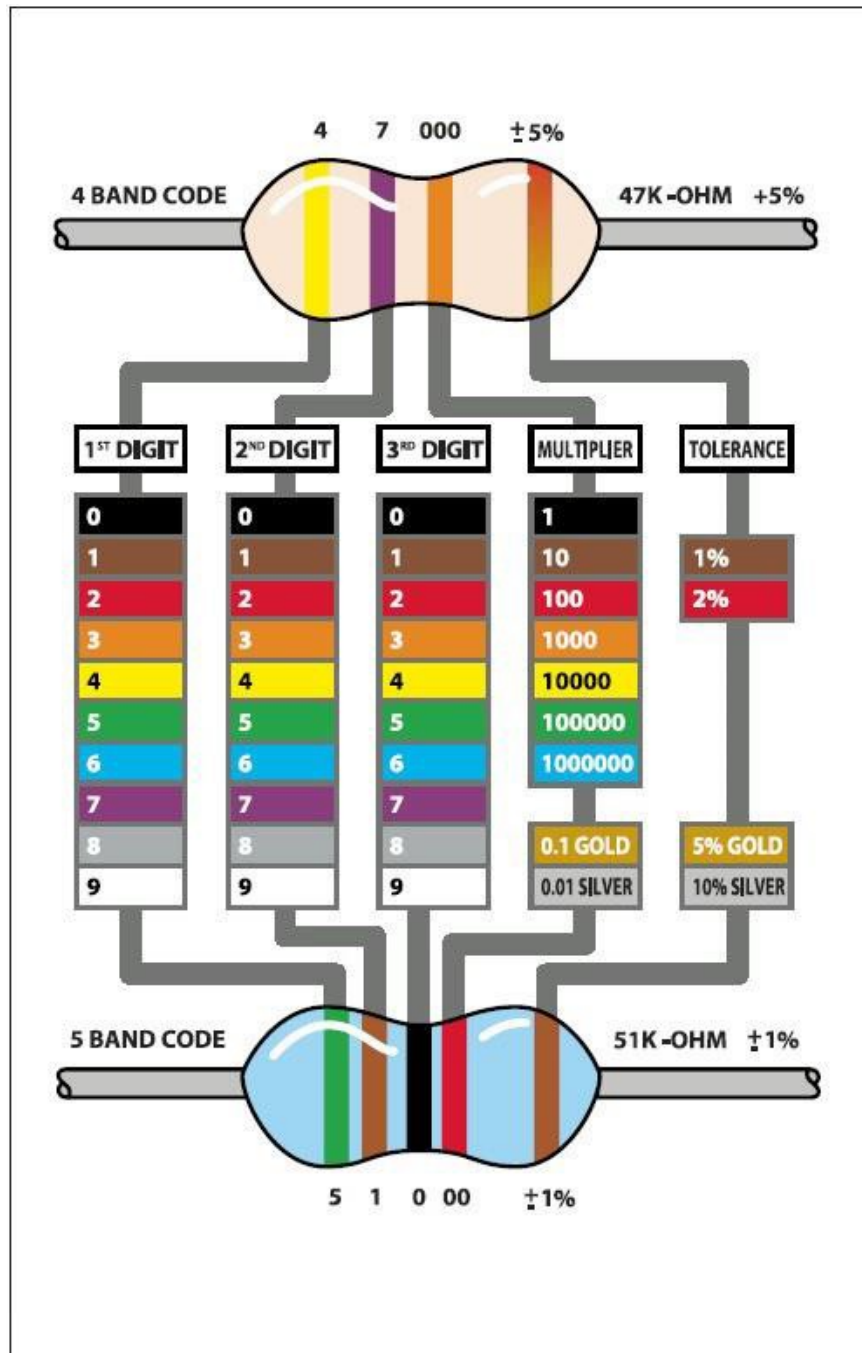


Dar mai întâi, mai multe despre rezistențe...

Unitatea de rezistență se numește Ohm, care este de obicei scurtat la  $\Omega$  litera greacă Omega. Deoarece un Ohm este o valoare scăzută a rezistenței (nu rezistă deloc), notăm și valorile rezistențelor în  $k\Omega$  (1.000  $\Omega$ ) și  $M\Omega$  (1.000.000  $\Omega$ ). Acestea se numesc kilo-ohmi și mega-ohmi.

În această lecție, vom folosi trei valori diferite ale rezistenței: 220 $\Omega$ , 1k $\Omega$  și 10k $\Omega$ . Aceste rezistențe arată toate la fel, cu excepția faptului că au dungi de culori diferite pe ele. Aceste dungi vă spun valoarea rezistorului.

Codul de culoare al rezistenței are trei dungi colorate și apoi o bandă aurie la un capăt.

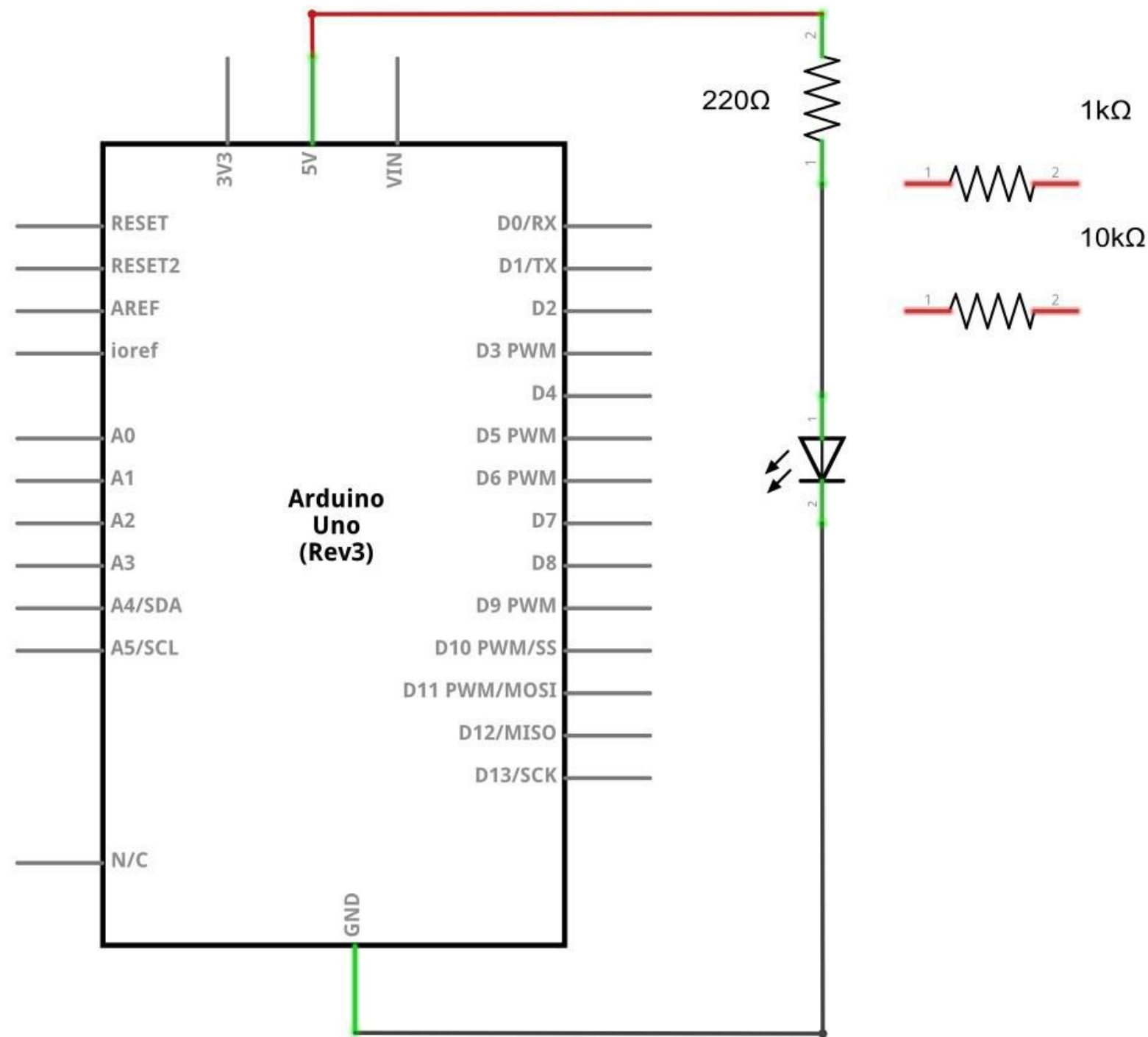


Spre deosebire de LED-uri, rezistențele nu au un fir pozitiv și negativ. Ele pot fi conectate în orice sens.

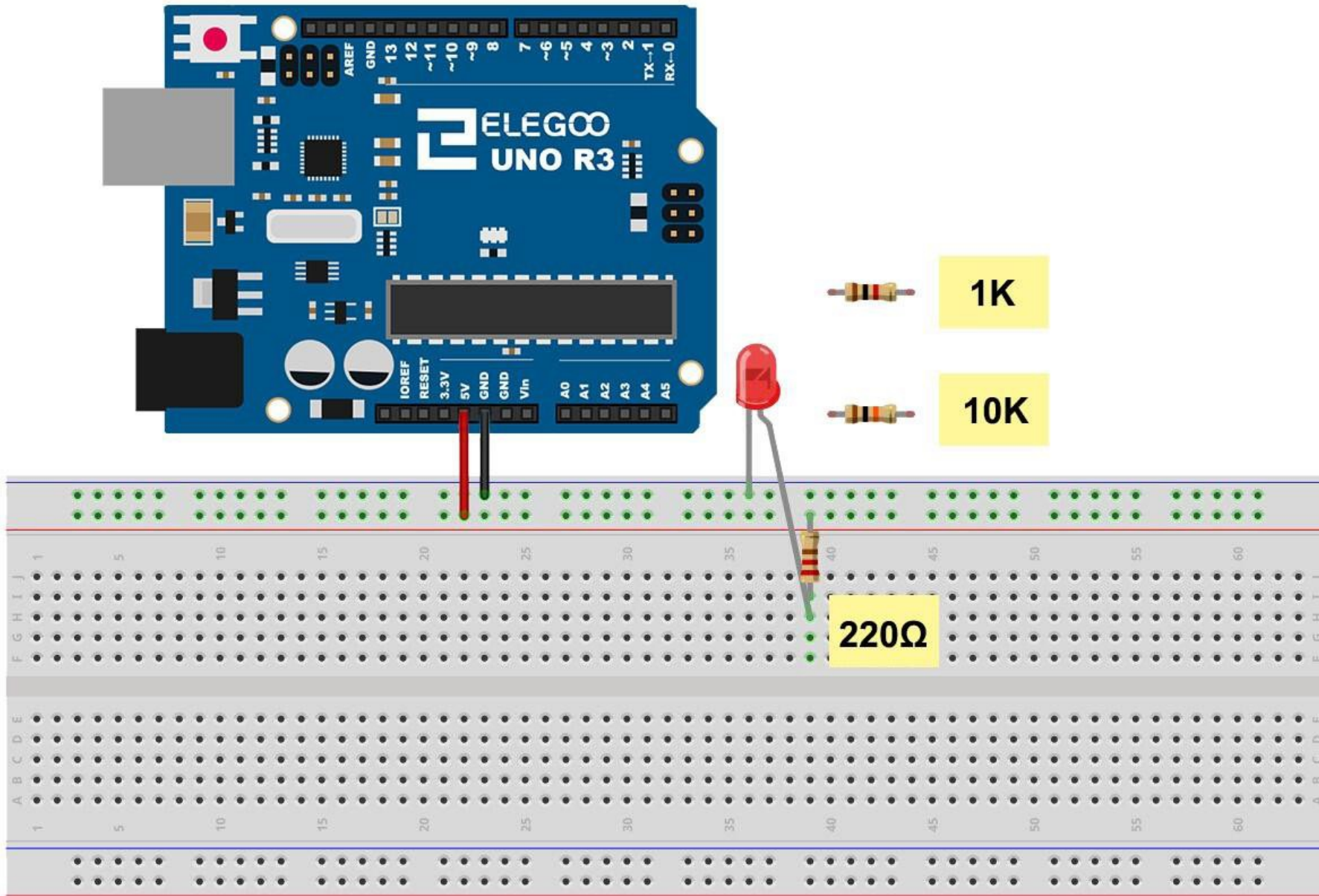
Dacă vi se pare că această metodă de abordare este prea complicată, puteți citi inelul de culoare direct pe rezistențele noastre pentru a determina valoarea rezistenței sale. Sau puteți folosi un multimetru digital în schimb.

Conexiune

Schematic



Schema de cablare



UNO este o sursă convenabilă de 5 volți, pe care o vom folosi pentru a furniza energie LED-ului și rezistenței. Nu trebuie să faceți nimic cu UNO, decât să îl conectați la un cablu USB.

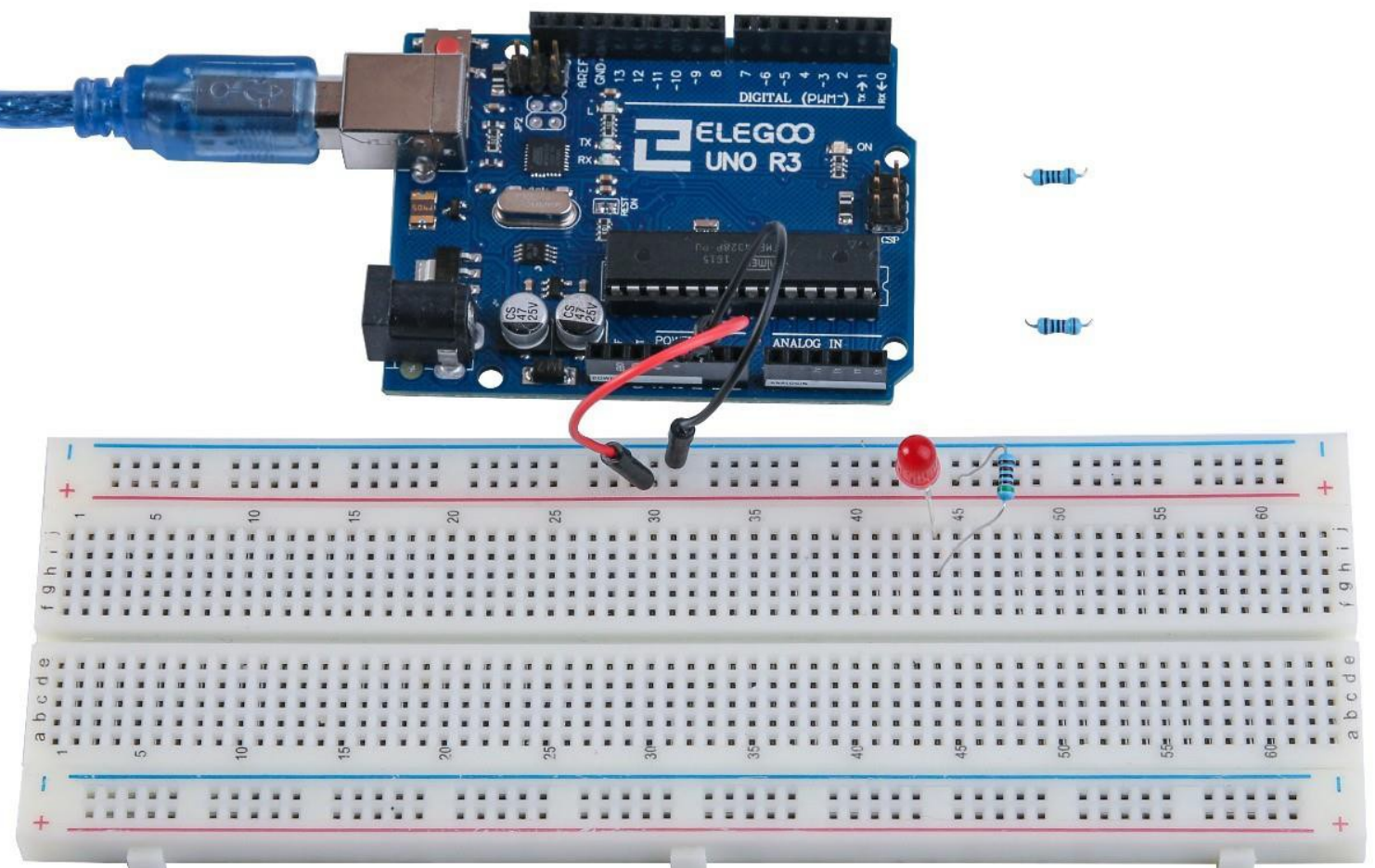
Cu rezistorul de  $220\ \Omega$  pe loc, LED-ul ar trebui să fie destul de luminos. Dacă schimbați rezistența de  $220\ \Omega$  cu rezistorul de  $1\text{ k}\Omega$ , atunci LED-ul va apărea puțin mai slab. În cele din urmă, cu rezistența de  $10\text{ k}\Omega$  pe loc, LED-ul va fi aproape vizibil. Trageți cablul roșu al jumperului din placa de breadboard și atingeți-l în orificiu și scoateți-l, astfel încât să acționeze ca un comutator. Ar trebui doar să poți observa diferența.

În acest moment, aveți 5V la un picior al rezistorului, celălalt picior al rezistorului merge la partea pozitivă a LED-ului și cealaltă parte a LED-ului merge la GND. Cu toate acestea, dacă am mutat rezistorul astfel încât să vină după LED, așa cum se arată mai jos, LED-ul se va aprinde în continuare.

Probabil că veți dori să puneți la loc rezistorul de  $220\ \Omega$ .

Nu contează pe ce parte a LED-ului punem rezistența, atâta timp cât este acolo undeva

### Imagine exemplu



## Lecția 4 LED RGB

### Prezentare generală

LED-urile RGB sunt o modalitate distractivă și ușoară de a adăuga un pic de culoare proiectelor tale.

Deoarece sunt ca 3 LED-uri obișnuite într-unul, modul de utilizare și conectare a acestora nu este mult diferit. Ele vin în principal în 2 versiuni: anod comun sau catod comun.

Anodul comun folosește 5V pe pinul comun, în timp ce catodul comun se conectează la masă.

Ca și în cazul oricărui LED, trebuie să conectăm niște rezistențe în linie (3 în total), astfel încât să putem limita curentul absorbit.

În schița noastră, vom începe cu LED-ul în starea de culoare roșie, apoi vom estompa la verde, apoi vom decolora la albastru și, în final, înapoi la culoarea roșie. Făcând acest lucru, vom trece prin cea mai mare parte a culorii care poate fi obținută.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x 830 Tie Points Breadboard

(4) x fire MM (fire de legătură de tip masculin la masculin)

(1) x LED RGB

(3) rezistențe x 220 ohmi

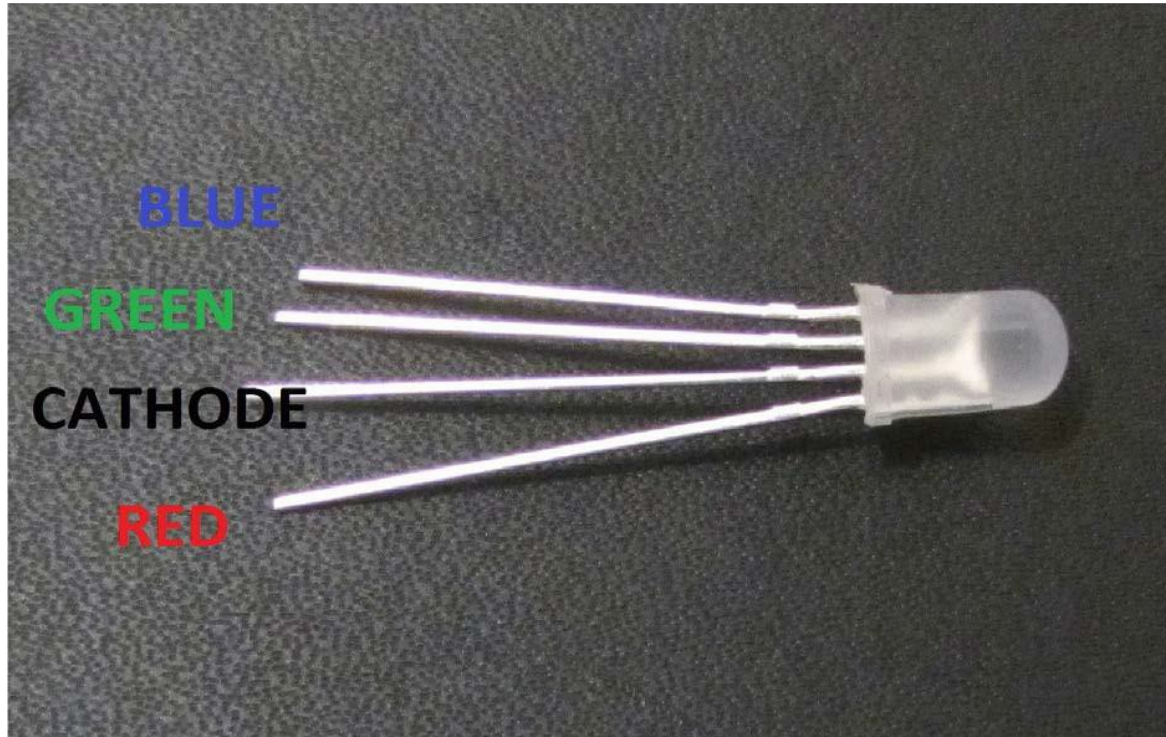
## Introducere componente

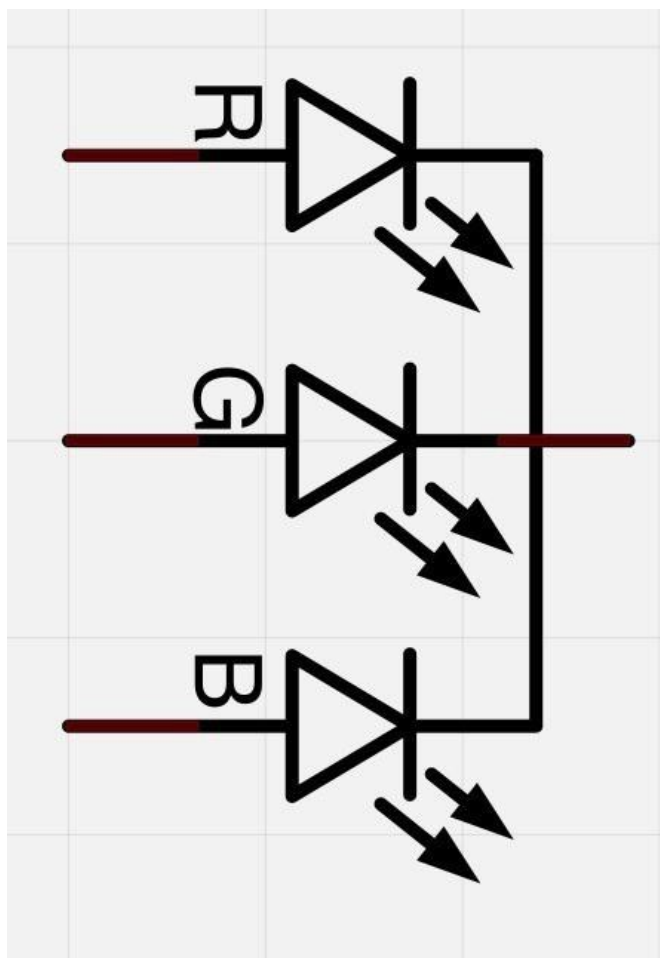
### RGB:

La prima vedere, LED-urile RGB (roșu, verde și albastru) arată exact ca LED-urile obișnuite. Cu toate acestea, în interiorul pachetului obișnuit de LED-uri, există de fapt trei LED-uri, unul roșu, unul verde și da, unul albastru. Controlând luminozitatea fiecărui LED-uri individuale, puteți amesteca aproape orice culoare doriți.

Amestecăm culorile în același mod în care ați amesteca vopseaua pe o paletă - prin ajustarea luminozității fiecăruia dintre cele trei LED-uri. Cel mai greu mod de a face acest lucru ar fi să folosiți rezistențe de valoare diferite (sau rezistențe variabile) așa cum am făcut-o în Lecția 2, dar asta este multă muncă! Din fericire pentru noi, placa UNO R3 are o funcție `analogWrite` pe care o puteți folosi cu pini marcați cu ~ pentru a furniza o cantitate variabilă de putere către LED-urile corespunzătoare.

LED-ul RGB are patru fire. Există un cablu care merge la conexiunea pozitivă a fiecărui LED din pachet și un singur cablu care este conectat la toate cele trei părți negative ale LED-urilor.





Aici pe fotografii puteți vedea 4 LED-uri cu electrozi. Fiecare pin separat pentru culoarea Verde, Albastru sau Roșu se numește Anod. Veți conecta întotdeauna „+” la acesta. Catodul merge la „-”(sol). Dacă îl conectați invers, LED-ul nu se va aprinde.

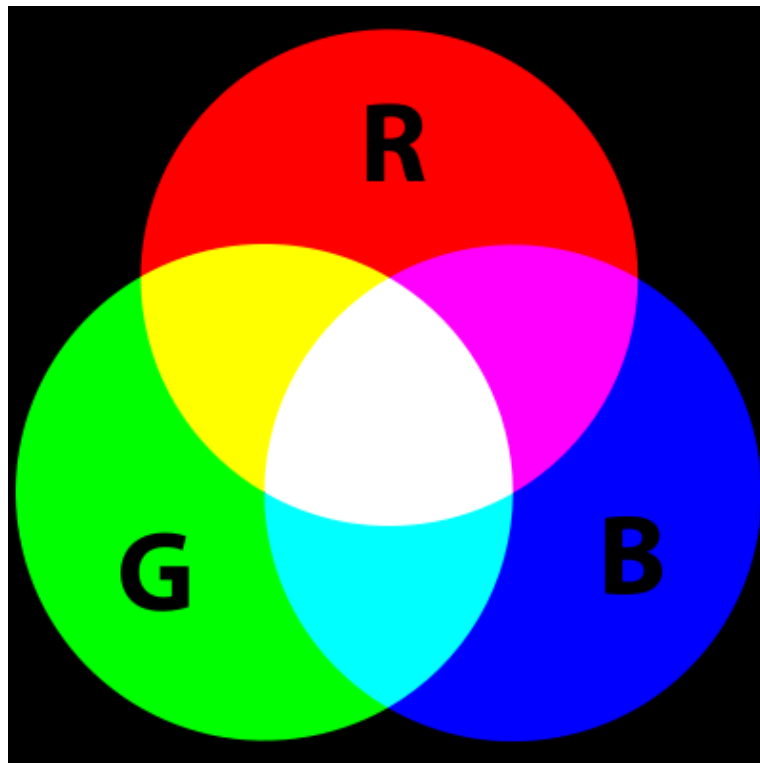
Conexiunea negativă comună a pachetului LED este al doilea pin din partea plată. Este, de asemenea, cel mai lung dintre cele patru cabluri și va fi conectat la pământ. Fiecare LED din interiorul pachetului necesită propriul rezistor de  $220\Omega$  pentru a preveni curgerea prea mare a curentului prin el. Cele trei fire pozitive ale LED-urilor (unul roșu, unul verde și unul albastru) sunt conectate la pinii de ieșire UNO folosind aceste rezistențe.

#### CULOARE:

Motivul pentru care poți amesteca orice culoare dorești variind cantitățile de lumină roșie, verde și albastră este că ochiul tău are trei tipuri de receptori de lumină (roșu, verde și albastru).

Ochiul și creierul tău procesează cantitățile de roșu, verde și albastru și le transformă într-o culoare a spectrului.

Într-un fel, folosind cele trei LED-uri, jucăm o păcăleală. Aceeași idee este folosită în televizoare, unde LCD-ul are puncte de culoare roșie, verzi și albastre unul lângă celălalt, alcătuind fiecare pixel.



Dacă setăm luminozitatea tuturor celor trei LED-uri să fie aceeași, atunci culoarea generală a luminii va fi albă. Dacă stingem LED-ul albastru, astfel încât doar LED-urile roșii și verzi să aibă aceeași luminozitate, atunci lumina va apărea galbenă.

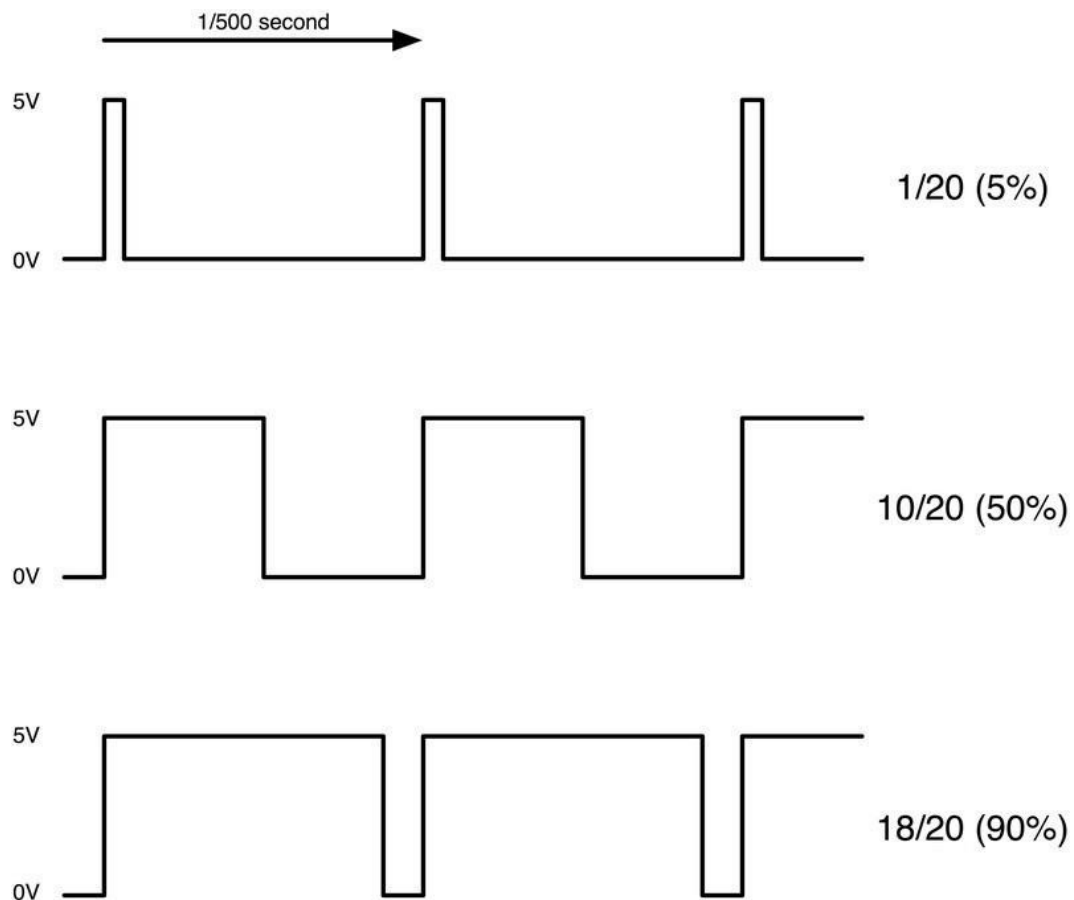
Putem controla luminozitatea fiecăreia dintre părțile roșii, verzi și albastre ale LED-ului separat, făcând posibilă amestecarea oricărei culori.

Negrul nu este atât o culoare, cât o absență a luminii. Prin urmare, cel mai aproape de negru cu LED-ul nostru este să oprim toate cele trei culori.

## Teorie (PWM)

Modularea lăţimii pulsului (PWM) este o tehnică de control al puterii. De asemenea, îl folosim aici pentru a controla luminozitatea fiecărui LED-uri.

Diagrama de mai jos arată semnalul de la unul dintre pinii PWM de pe UNO.



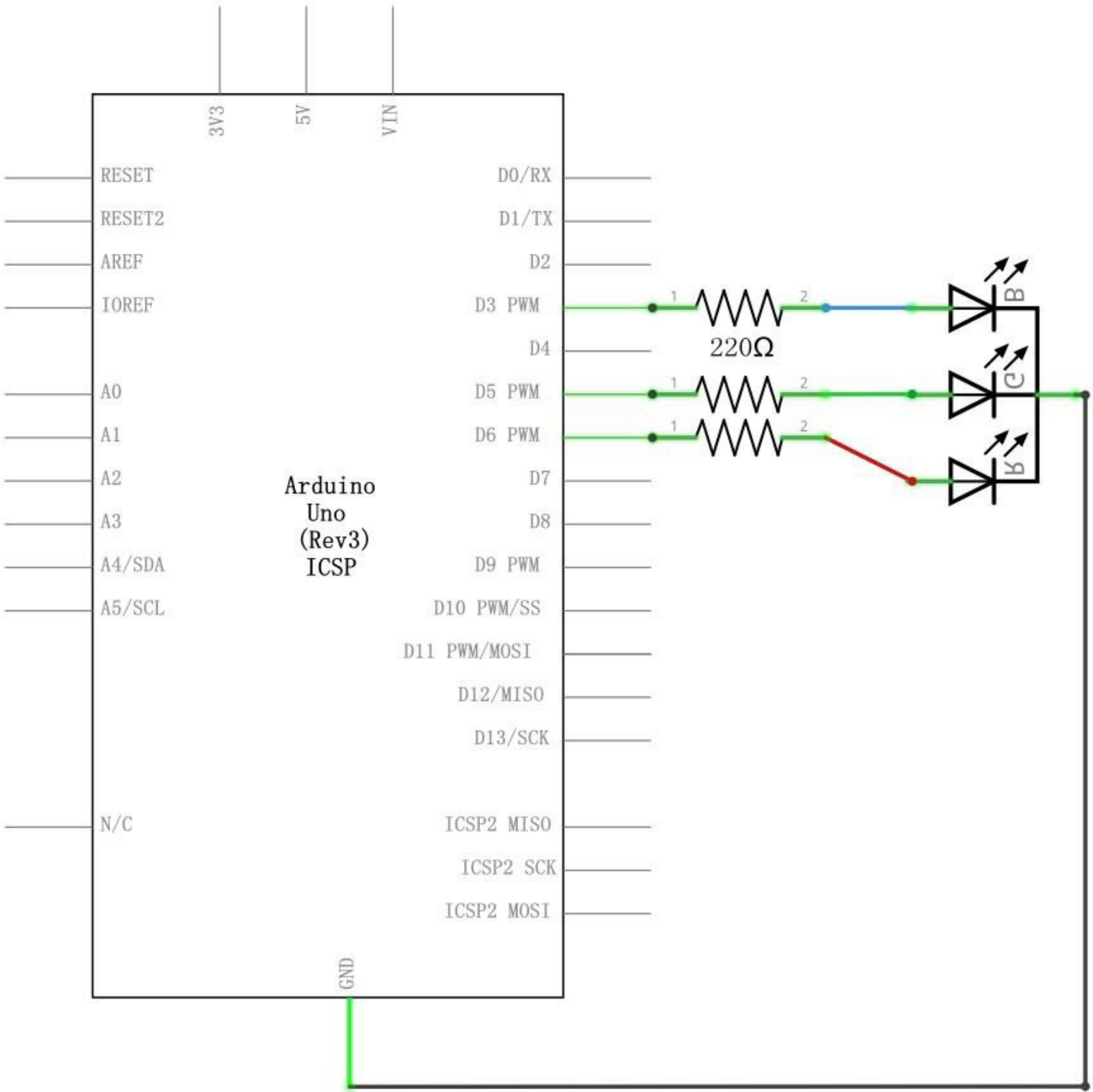
Aproximativ la fiecare 1/500 de secundă, ieşirea PWM va produce un impuls. Lungimea acestui impuls este controlată de funcţia „analogWrite”. Aşadar, „analogWrite(0)” nu va produce niciun impuls, iar „analogWrite(255)” va produce un impuls care durează până la următorul impuls, astfel încât ieşirea să fie de fapt activată tot timpul.

Dacă specificăm o valoare în analogWrite care este undeva între 0 şi 255, atunci vom produce un impuls. Dacă impulsul de ieşire este ridicat doar pentru 5% din timp, atunci orice conducem va primi doar 5% din puterea maximă.

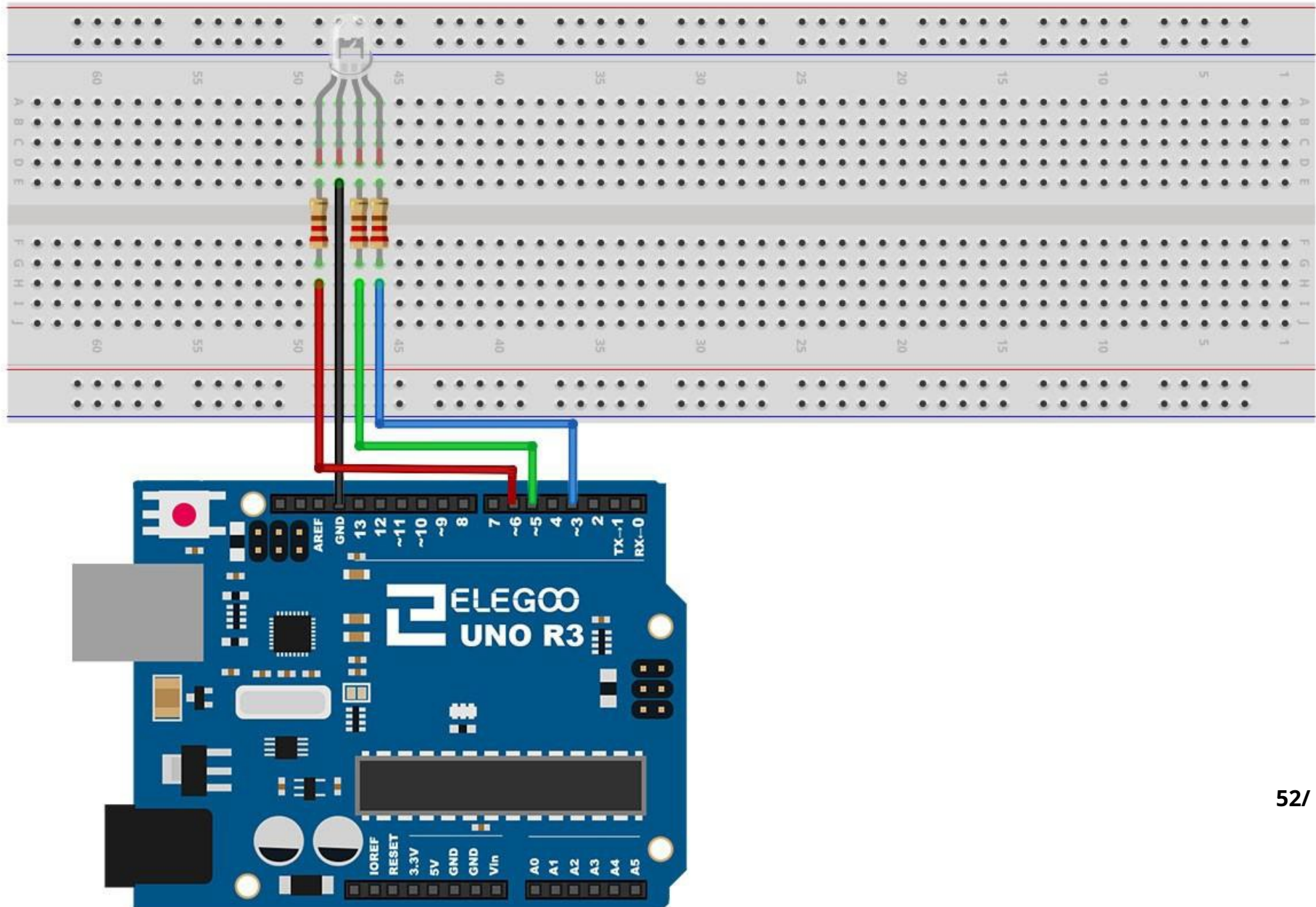
Dacă, totuşi, ieşirea este la 5V timp de 90% din timp, atunci sarcina va primi 90% din puterea furnizată. Nu putem vedea LED-urile pornind şi oprindu-se la acea viteză, aşa că pentru noi se pare că luminozitatea se schimbă.

Conexiune

Schematic

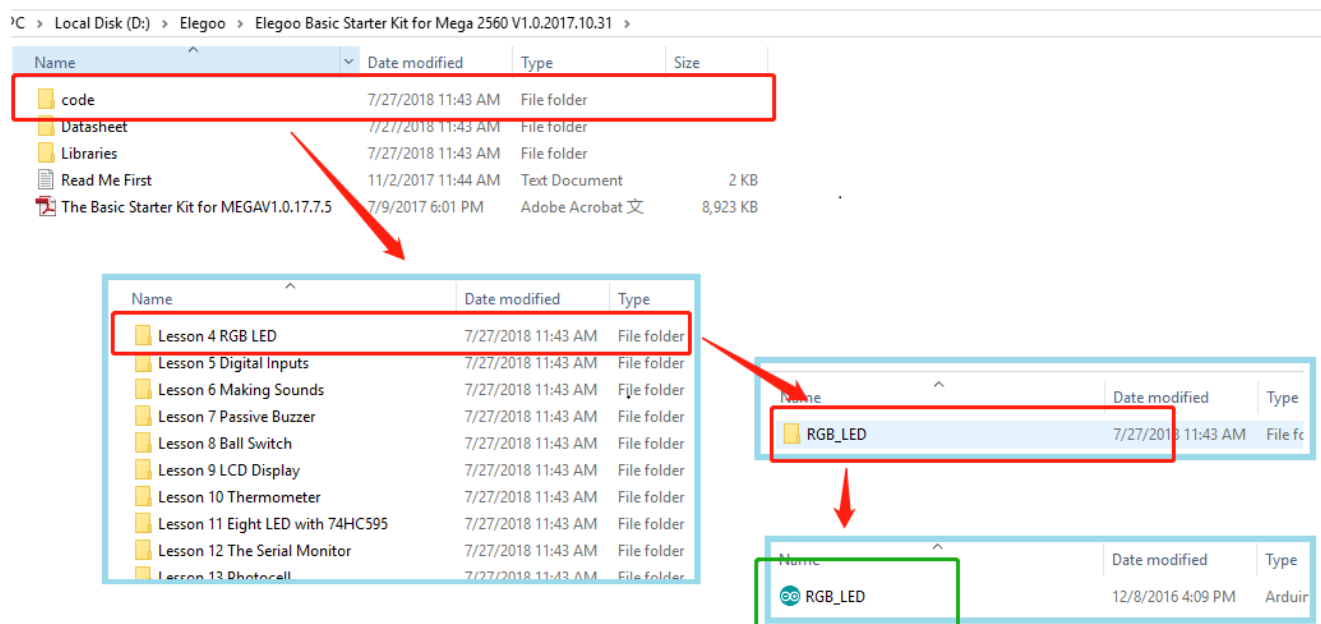


## Schema de cablare



## Cod

După cablare, deschideți calea Schiță în folder: Tutorial > Engleză > cod > Lecția 4 RGB LED > RGB\_LED și faceți clic pe ÎNCĂRCARE pentru a încărca programul.



Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Schița începe prin a specifica ce pini vor fi utilizați pentru fiecare dintre culori: //

### Definiți pinii

# definește ALBASTRU 3

# definește VERDE 5

# definește roșu 6

Următorul pas este să scrieți funcția „setup”. După cum am învățat în lecțiile anterioare, funcția de configurare rulează doar o dată după ce Arduino s-a resetat. În acest caz, tot ce trebuie să facă este să definească cei trei pini pe care îi folosim ca ieșiri.

```
void setup()
{
  pinMode(ROȘU, IEȘIRE);
  pinMode(VERDE, IEȘIRE);
  pinMode(ALBASTRU, IEȘIRE);
  digitalWrite(ROȘU, ÎNALT);
  digitalWrite(VERDE, JOS);
  digitalWrite(ALBASTRU, JOS); }
```

Înainte de a arunca o privire la funcția „bucă”, să ne uităm la ultima funcție din schiță.

Variabilele definite

```
redValue = 255; // alegeți o valoare între 1 și 255 pentru a schimba culoarea.
```

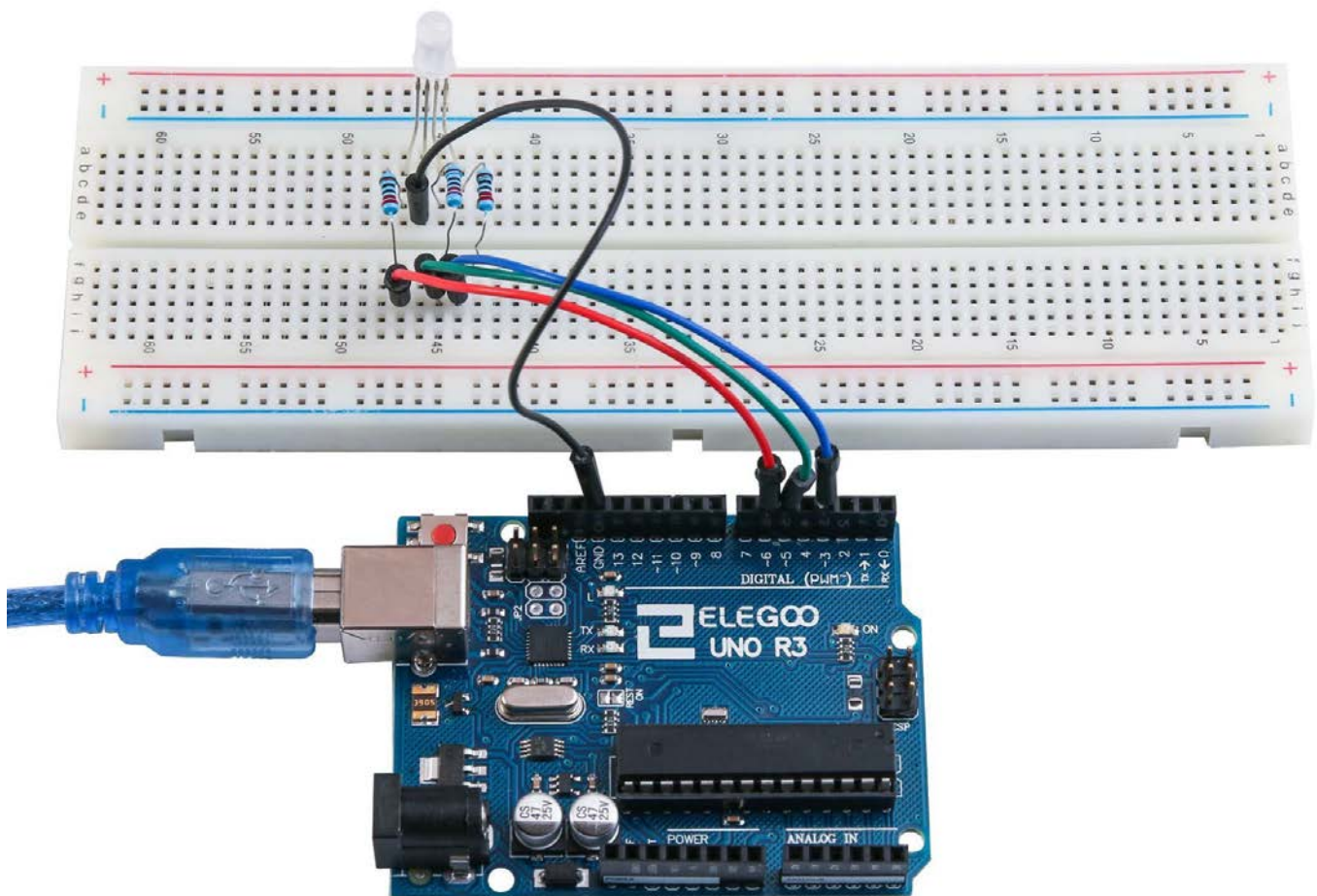
```
greenValue = 0;
```

```
blueValue = 0;
```

Această funcție are trei argumente, unul pentru luminozitatea LED-urilor roșii, verzi și albastre. În fiecare caz, numărul va fi în intervalul de la 0 la 255, unde 0 înseamnă oprit și 255 înseamnă luminozitate maximă. Funcția apelează apoi „analogWrite” pentru a seta luminozitatea fiecărui LED.

Încercați să adăugați câteva culori proprii la schiță și urmăriți efectul pe LED-ul dvs.

### Imagine exemplu



## Lecția 5 Intrări digitale

### Prezentare generală

În această lecție, veți învăța să utilizați butoanele cu intrări digitale pentru a porni și opri un LED.

Apăsarea butonului va aprinde LED-ul; apăsarea celui alt buton va stinge LED-ul.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x 830 Breadboard cu puncte de legătură

(1) LED roșu de 5 mm

(1) x rezistor de 220 ohmi

(2) x întrerupătoare cu apăsare

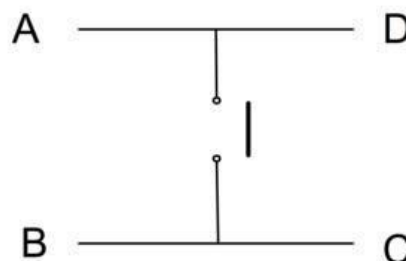
(7) fire x MM (fire de legătură de sex masculin la masculin)

### Introducere componente

#### Întrerupătoare cu apăsare:

Comutatoarele sunt componente cu adevărat simple. Când apăsați un buton sau răsturnați o pârghie, acestea conectează două contacte împreună, astfel încât electricitatea să poată circula prin ele.

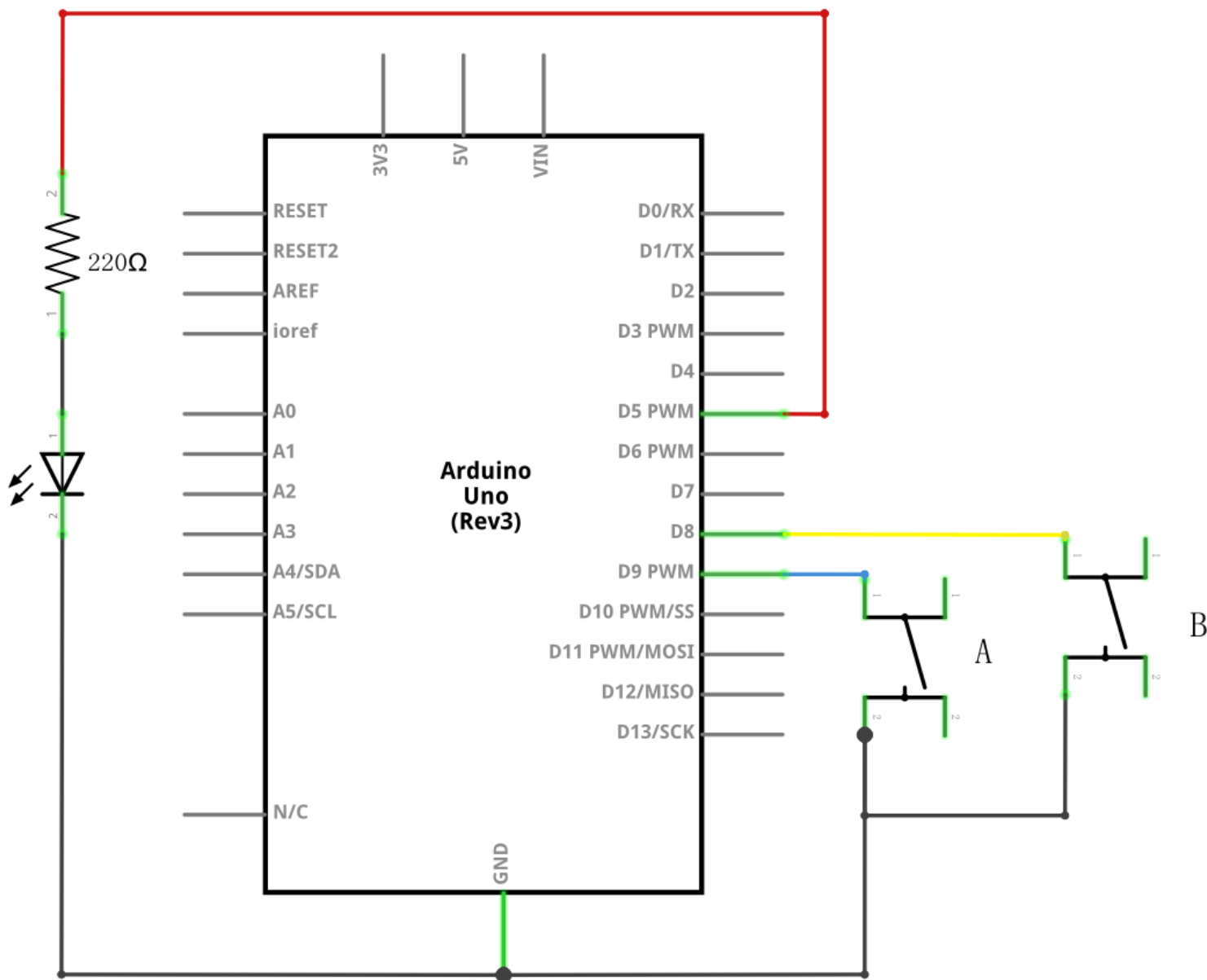
Micile comutatoare tactile care sunt folosite în această lecție au patru conexiuni, ceea ce poate fi puțin confuz.



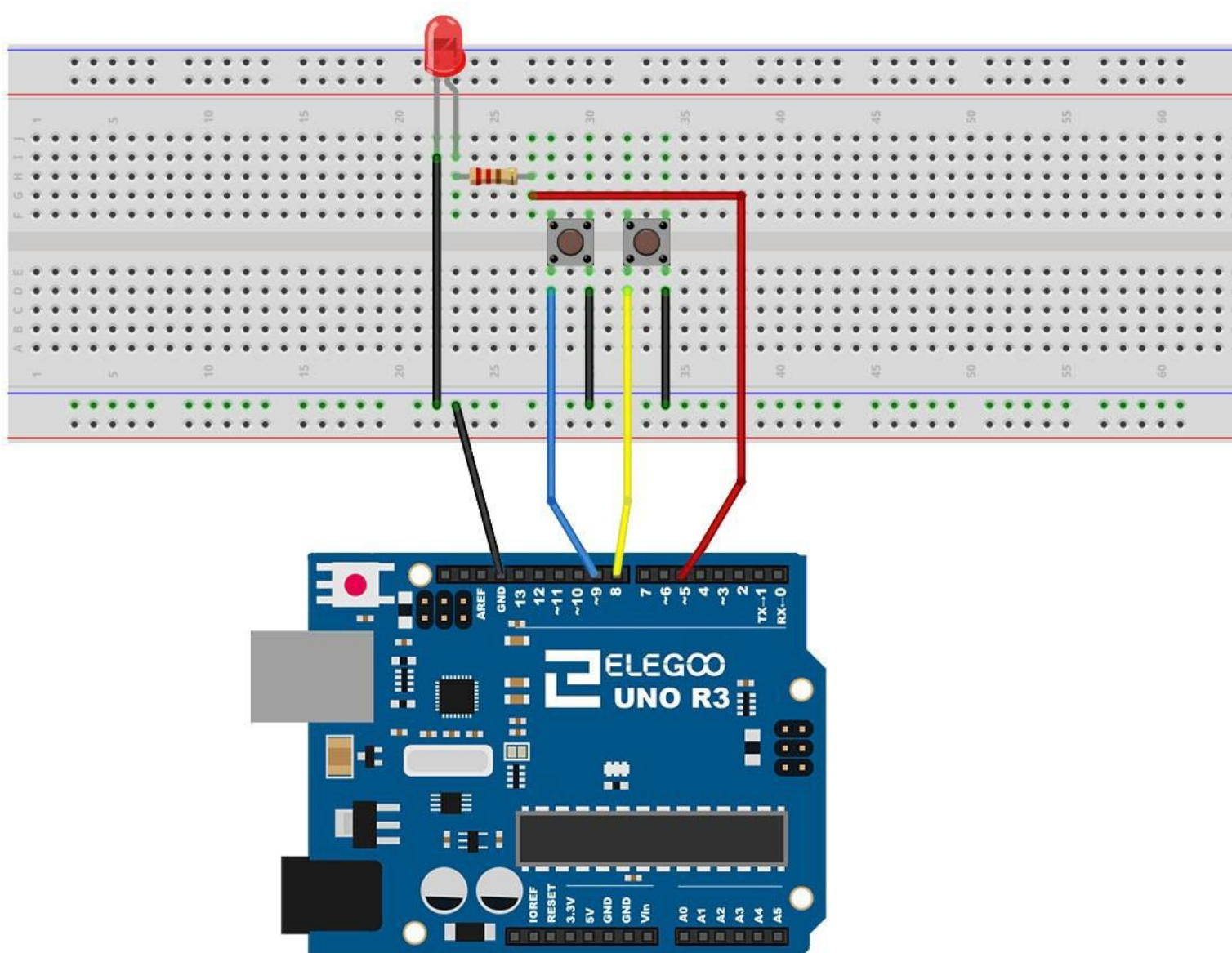
De fapt, există doar două conexiuni electrice. În interiorul pachetului de comutatoare, pinii B și C sunt conectați împreună, la fel ca și A și D.

Conexiune

Schematic



## Schema de cablare



Deși corpurile comutatoarelor sunt pătrate, pinii ies din părțile opuse ale comutatorului. Aceasta înseamnă că pinii vor fi suficient de departe unul de celălalt atunci când sunt așezați corect pe placa.

Amintiți-vă că LED-ul trebuie să aibă conductorul negativ mai scurt spre stânga.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 5 Intrări digitale și apăsați UPLOAD pentru a încărca programul. Dacă sunt solicitate erori, consultați Lecția 2 pentru detalii despre tutorialul despre încărcarea programului.

Încărcați schița pe placa UNO. Apăsarea butonului din stânga va aprinde LED-ul, în timp ce apăsarea butonului din dreapta îl va stinge.

Prima parte a schiței definește trei variabile pentru cei trei pini care urmează să fie utilizați.

„ledPin” este pinul de ieșire, iar „buttonApin” se va referi la comutatorul aflat mai aproape de partea superioară a panoului și „buttonBpin” la celălalt comutator.

Funcția „setup” definește ledPin-ul ca fiind o IEȘIRE ca în mod normal, dar acum avem cele două intrări de care trebuie să ne ocupăm. În acest caz, folosim setarea pinMode să fie „INPUT\_PULLUP” astfel:

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

Modul PIN al INPUT\_PULLUP înseamnă că pinul trebuie utilizat ca intrare, dar dacă nu este conectat nimic altceva la intrare, acesta ar trebui să fie „tras în sus” la HIGH. Cu alte cuvinte, valoarea implicită pentru intrare este HIGH, cu excepția cazului în care este trasă LOW prin acțiunea de a apăsa butonul.

Acesta este motivul pentru care comutatoarele sunt conectate la GND. Când un comutator este apăsat, acesta conectează pinul de intrare la GND, astfel încât să nu mai fie HIGH.

Deoarece intrarea este în mod normal ÎNALTĂ și devine JOSĂ doar când butonul este apăsat, logica este puțin inversată. Ne vom ocupa de asta în funcția „bucă”.

bucă goală ()

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

```

{
  digitalWrite(ledPin, LOW);
}
}

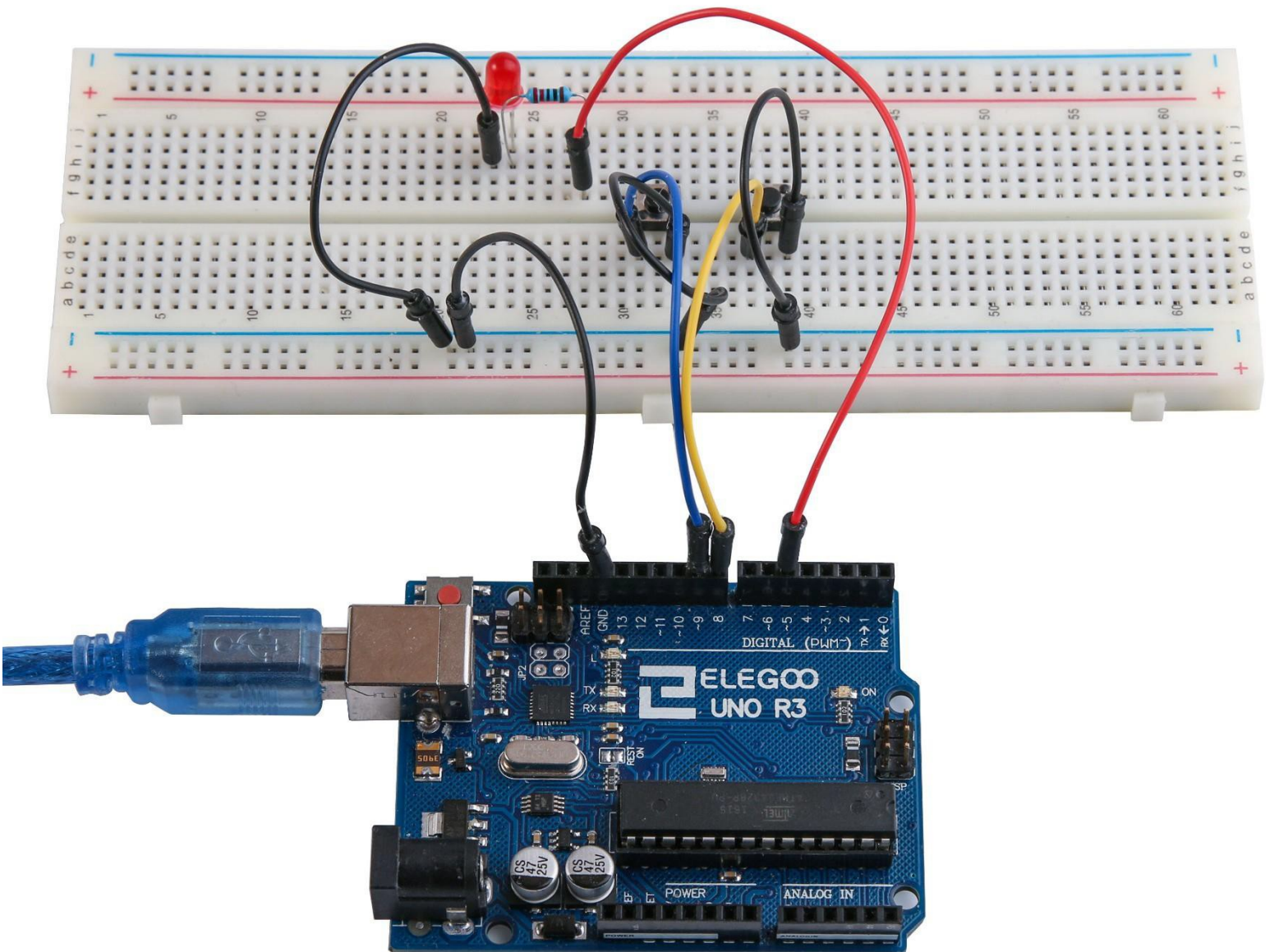
```

În funcția „bucă” există două instrucțiuni „dacă”. Unul pentru fiecare buton. Fiecare face un „digitalRead” la intrarea corespunzătoare.

Amintiți-vă că dacă butonul este apăsat, intrarea corespunzătoare va fi LOW, dacă butonul A este scăzut, atunci un „digitalWrite” de pe ledPin-ul se aprinde.

În mod similar, dacă butonul B este apăsat, pe theledPin este scris un LOW.

### Imagine exemplu



## Lecția 6 Sonerie activă

### Prezentare generală

În această lecție, veți învăța cum să generați un sunet cu un sonerie activă.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x Sonerie activă
- (2) x fire FM (fire DuPont de la femeie la masculin)

### Introducere componente

#### BUZZER:

Soneriile electronice sunt alimentate cu curent continuu și sunt echipate cu un circuit integrat.

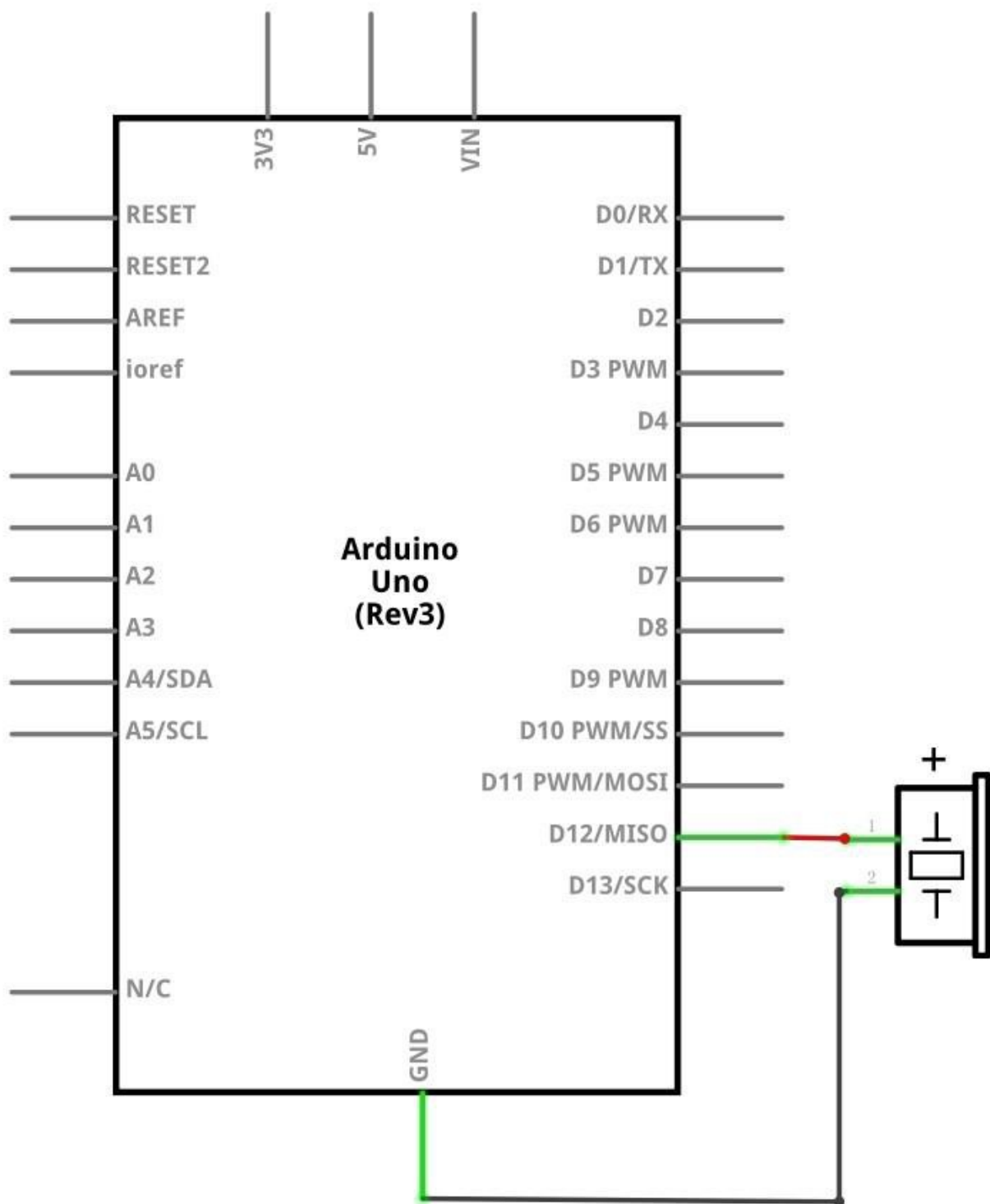
Sunt utilizate pe scară largă în calculatoare, imprimante, fotocopiatore, alarme, jucării electronice, dispozitive electronice auto, telefoane, cronometre și alte produse electronice pentru dispozitive vocale. Buzzerile pot fi clasificate ca active și pasive. Întoarceți știfturile a două sonerie cu fața în sus. Cel cu o placă de circuit verde este un buzzer pasiv, în timp ce celălalt închis cu o bandă neagră este unul activ.

Diferența dintre cele două este că un buzzer activ are o sursă oscilantă încorporată, astfel încât va genera un sunet atunci când este electrificat. Un buzzer pasiv nu are o astfel de sursă, astfel încât să nu va tweek dacă sunt utilizate semnale DC; în schimb, trebuie să utilizați unde pătrate a căror frecvență este între 2K și 5K pentru a-l conduce. Soneria activă este adesea mai scumpă decât cea pasivă din cauza multiplelor circuite oscilante încorporate.

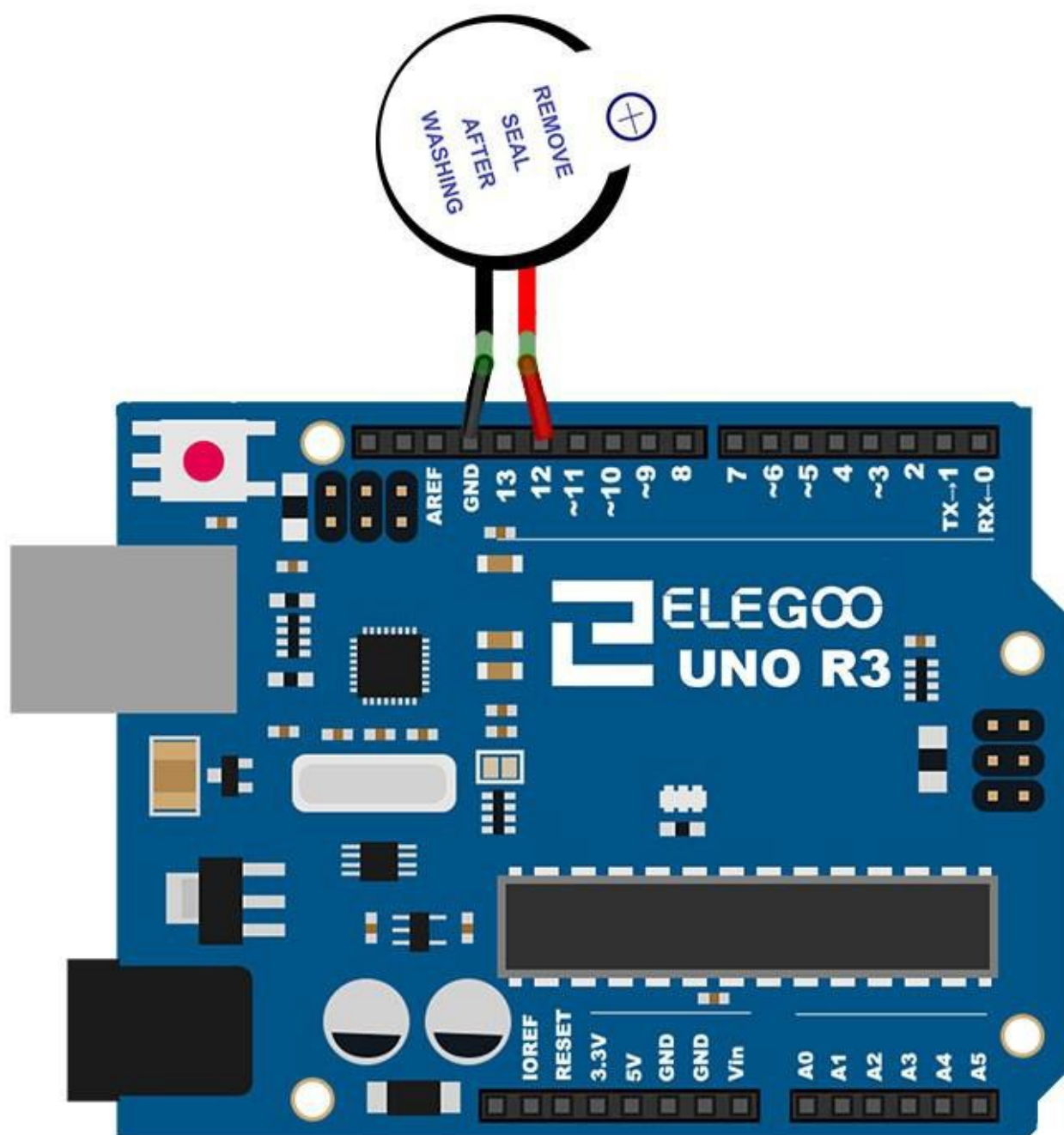


## Conexiune

### Schematic



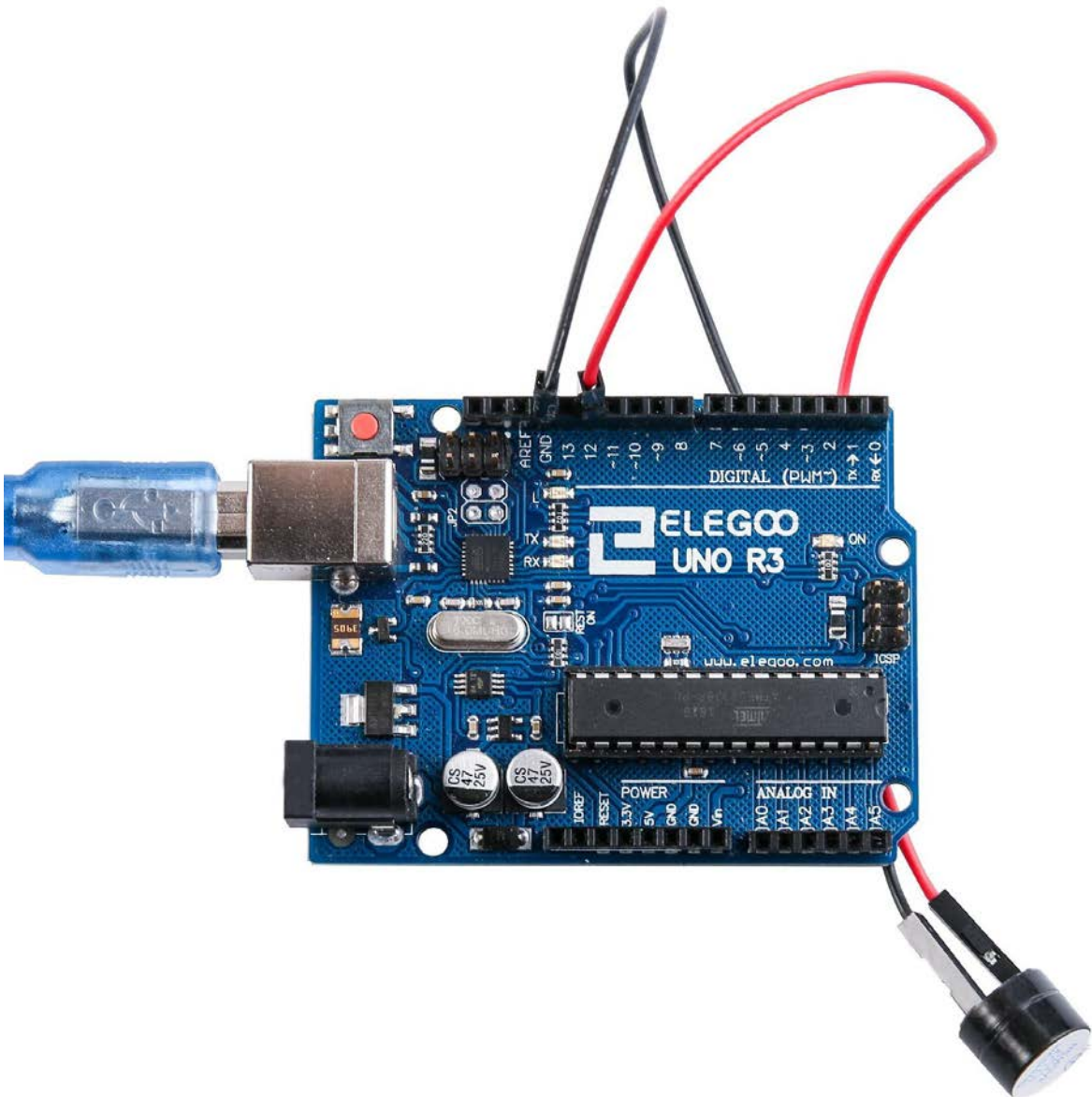
## Schema de cablare



## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 6 Emiterea de sunete și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

## Imagine exemplu



## Lecția 7 Buzzer pasiv

### Prezentare generală

În această lecție, veți învăța cum să utilizați un semnal sonor pasiv.

Scopul experimentului este de a genera opt sunete diferite, fiecare sunet durând 0,5 secunde: de la Alto Do (523Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz) la Treble Do (1047Hz).

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x Sonerie pasivă

(2) x fire FM (fire DuPont de la femeie la masculin)

### Introducere componente

#### Buzzer pasiv:

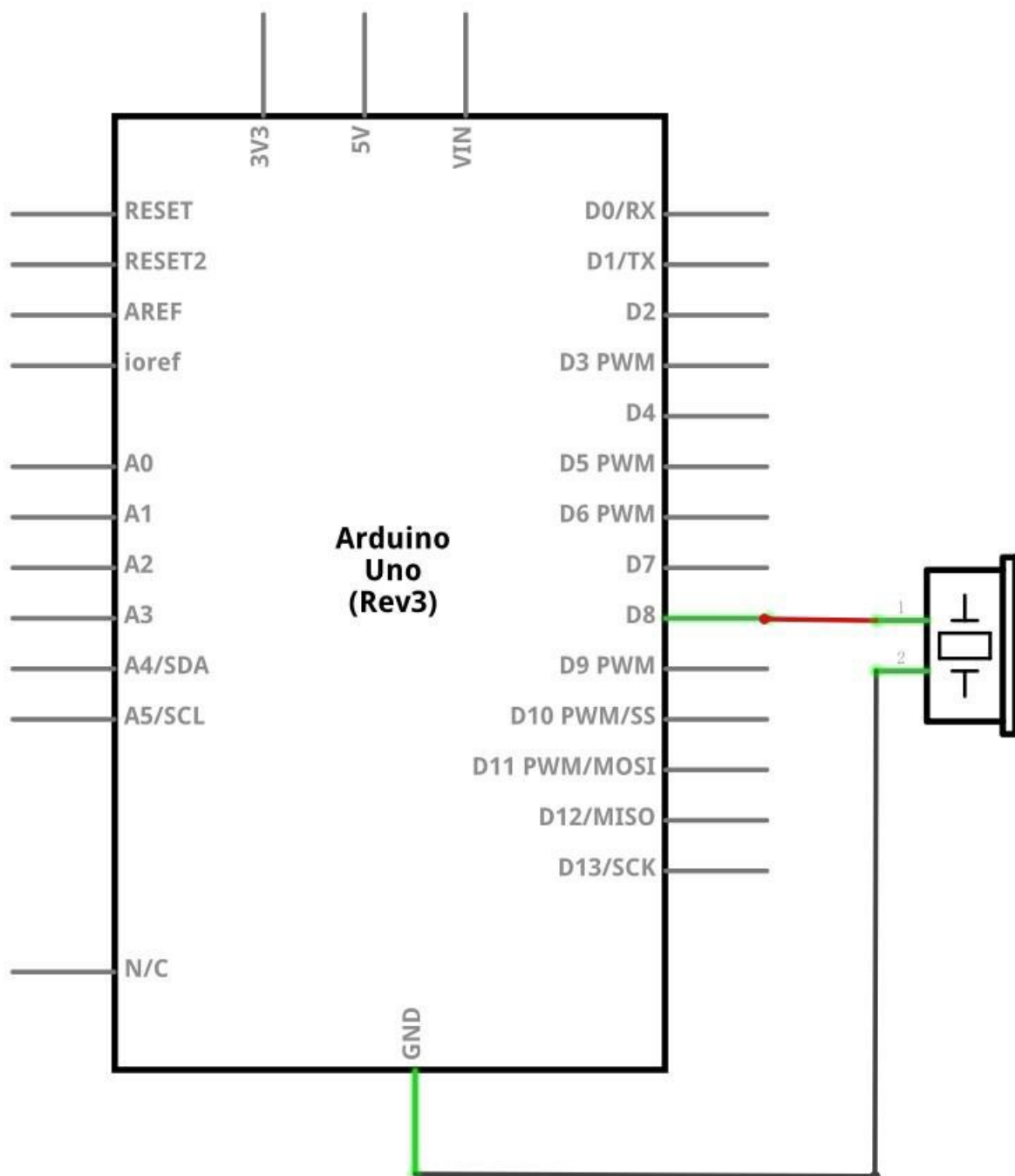
Principiul de funcționare al soneriei pasive este utilizarea audio care generează PWM pentru a face aerul să vibreze. Modificat corespunzător, atâta timp cât frecvența de vibrație, poate genera sunete diferite. De exemplu, trimițând un puls de 523 Hz, poate genera Alto Do, puls de 587 Hz, poate genera Re medii, puls de 659 Hz, poate produce gama medie Mi. Cu ajutorul soneriei, puteți reda o melodie.

Ar trebui să avem grijă să nu folosim funcția analogică Write () a plăcii UNO R3 pentru generați un impuls la sonerie, deoarece ieșirea impulsului de scriere analogică () este fixă (500 Hz).

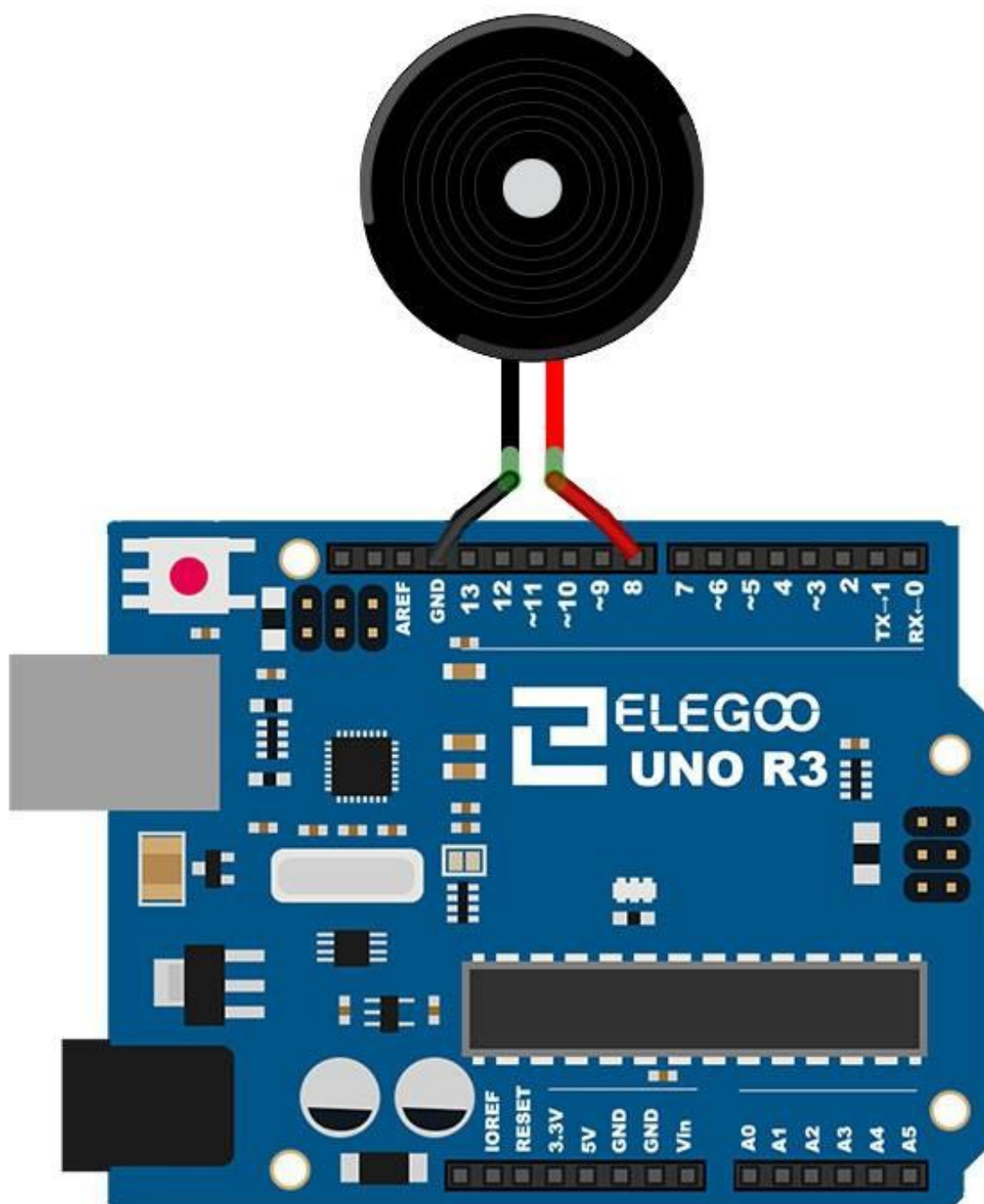


## Conexiune

### Schematic



## Schema de cablare



Cablajul soneriei conectat la placa UNO R3, cel roșu (pozitiv) la pin8, firul negru (negativ) la GND.

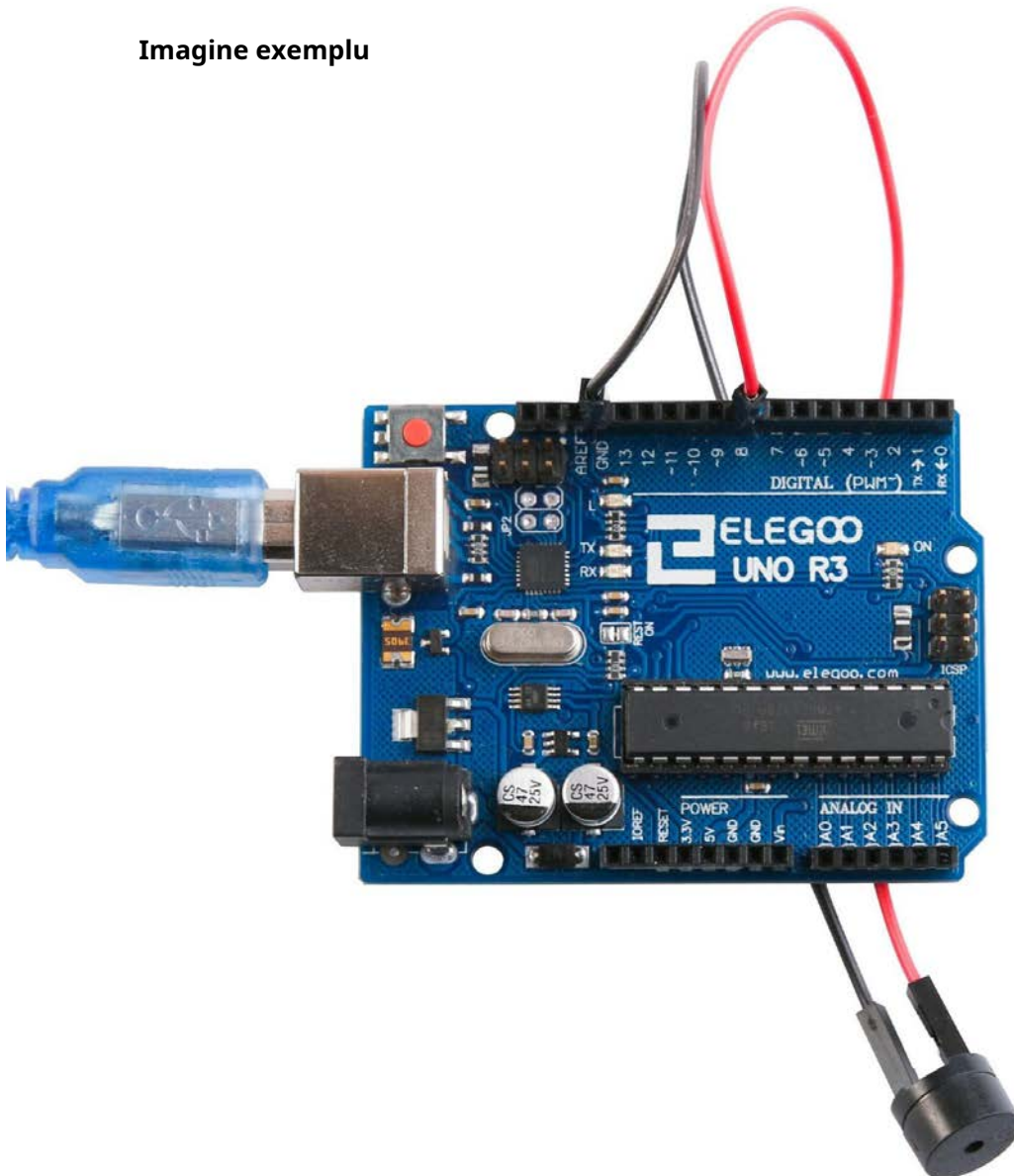
## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 7 Passive Buzzer și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca <pitch> sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați Lecția 1.

## Imagine exemplu



## Lecția 8 Comutator cu bilă înclinată

### Prezentare generală

În această lecție, veți învăța cum să utilizați un comutator cu bilă de înclinare pentru a detecta un unghi mic de înclinare.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x întrerupător cu bilă înclinată

(2) x fire FM (fire DuPont de la femeie la masculin)



### Introducere componente

#### Senzor de înclinare:

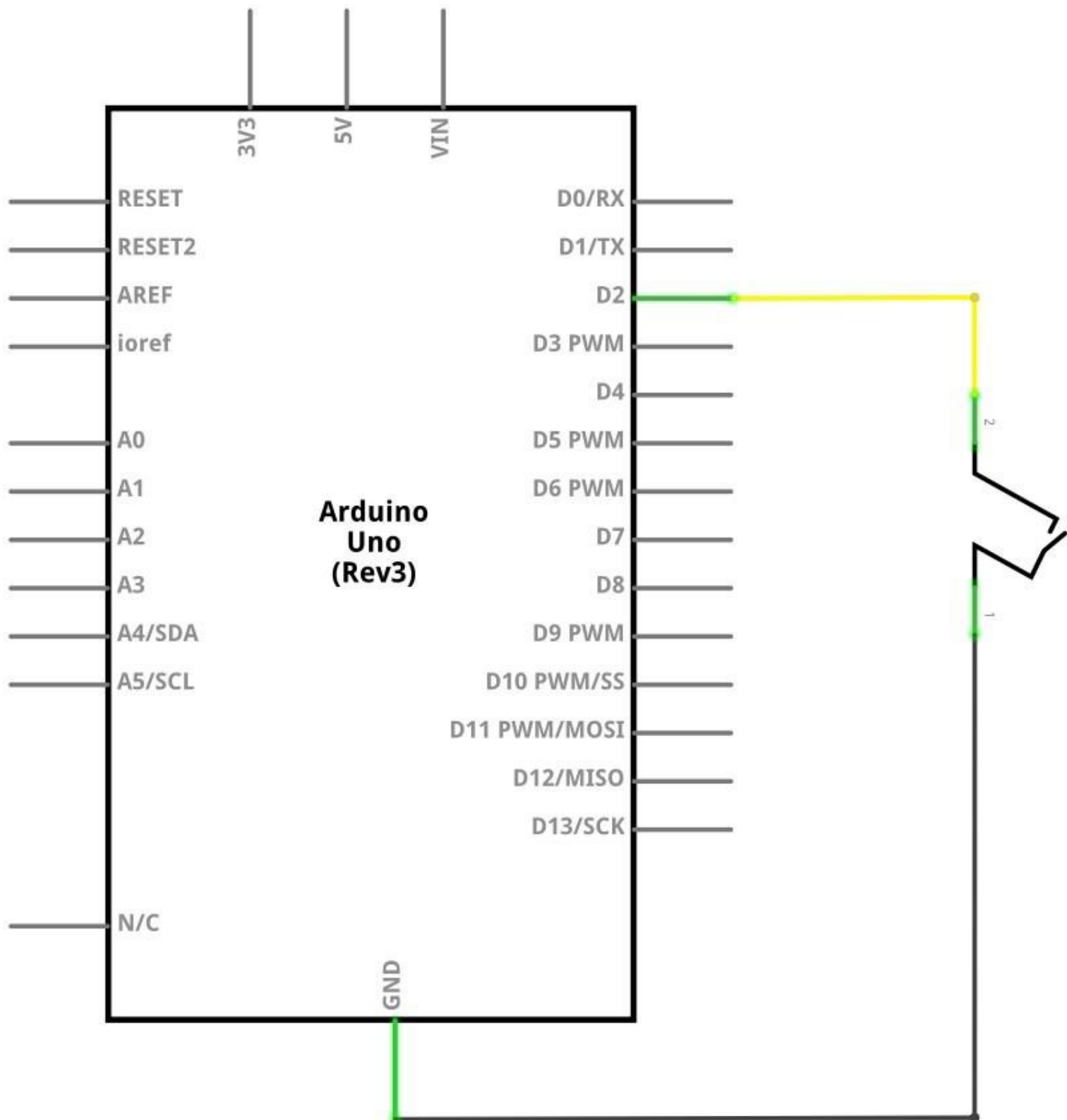
Senzorii de înclinare (comutator cu bile de înclinare) vă permit să detectați orientarea sau înclinarea. Sunt mici, ieftine, cu putere redusă și ușor de utilizat. Dacă sunt folosite corect, nu se vor uza. Simplitatea lor le face populare pentru jucării, gadgeturi și electrocasnice. Uneori, ele sunt denumite „comutatoare cu mercur”, „comutatoare de înclinare” sau „senzori cu bile de rulare” din motive evidente.

Ele sunt de obicei alcătuite dintr-o cavitate de un fel (cilindric este popular, deși nu întotdeauna) cu o masă liberă conductivă în interior, cum ar fi o stropire de mercur sau o bilă rulantă. Un capăt al cavității are două elemente conductoare (poli). Când senzorul este orientat astfel încât capătul respectiv să fie în jos, masa se rostogolește pe stâlpi și îi scurtează, acționând ca o aruncare a comutatorului.

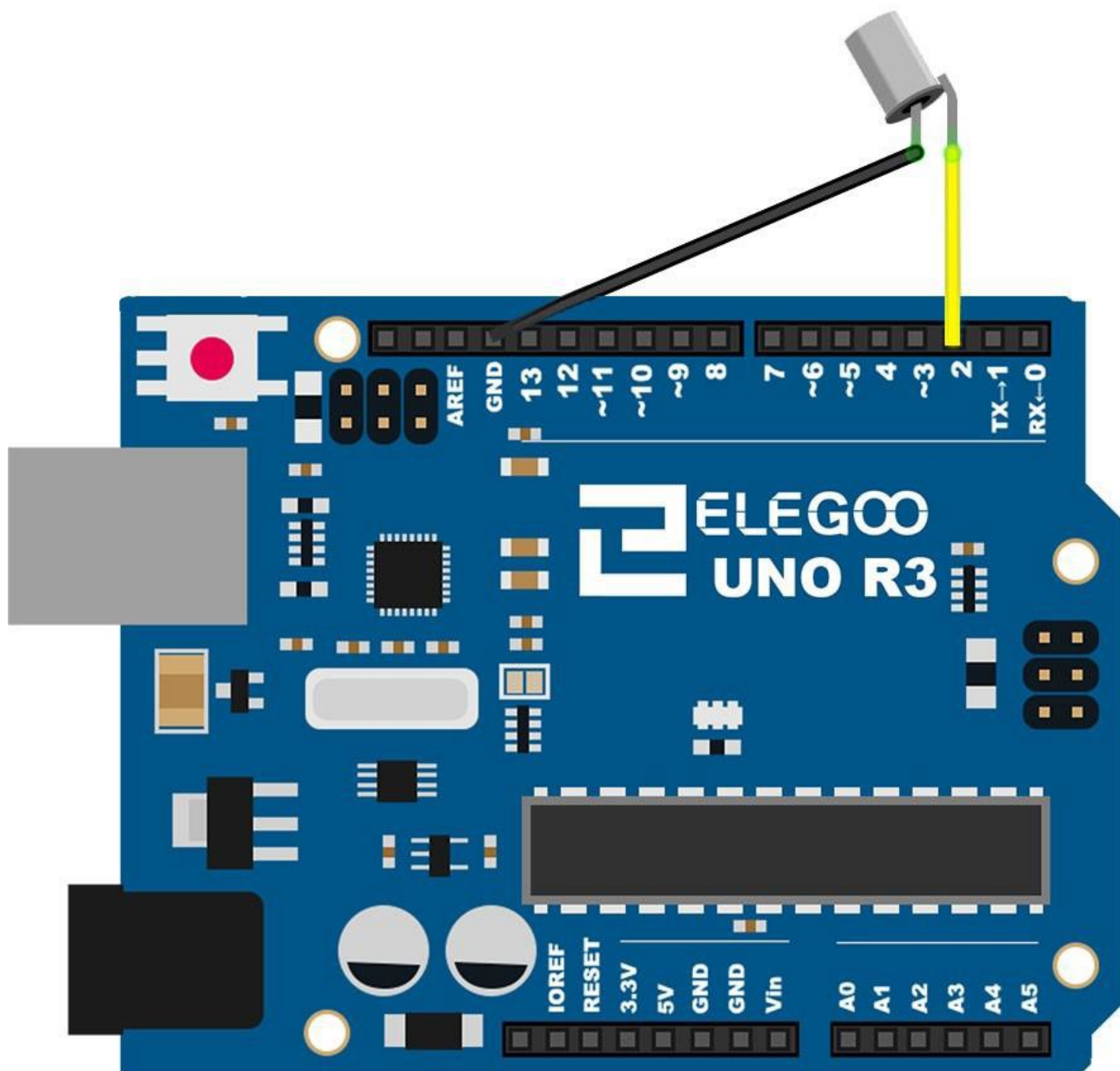
Deși nu sunt la fel de precise sau flexibile ca un accelerometru complet, comutatoarele de înclinare pot detecta mișcarea sau orientarea. Un alt beneficiu este că cei mari pot comuta singuri puterea. Accelerometrele, pe de altă parte, produc tensiune digitală sau analogică care trebuie apoi analizată folosind circuite suplimentare.

## Conexiune

### Schematic



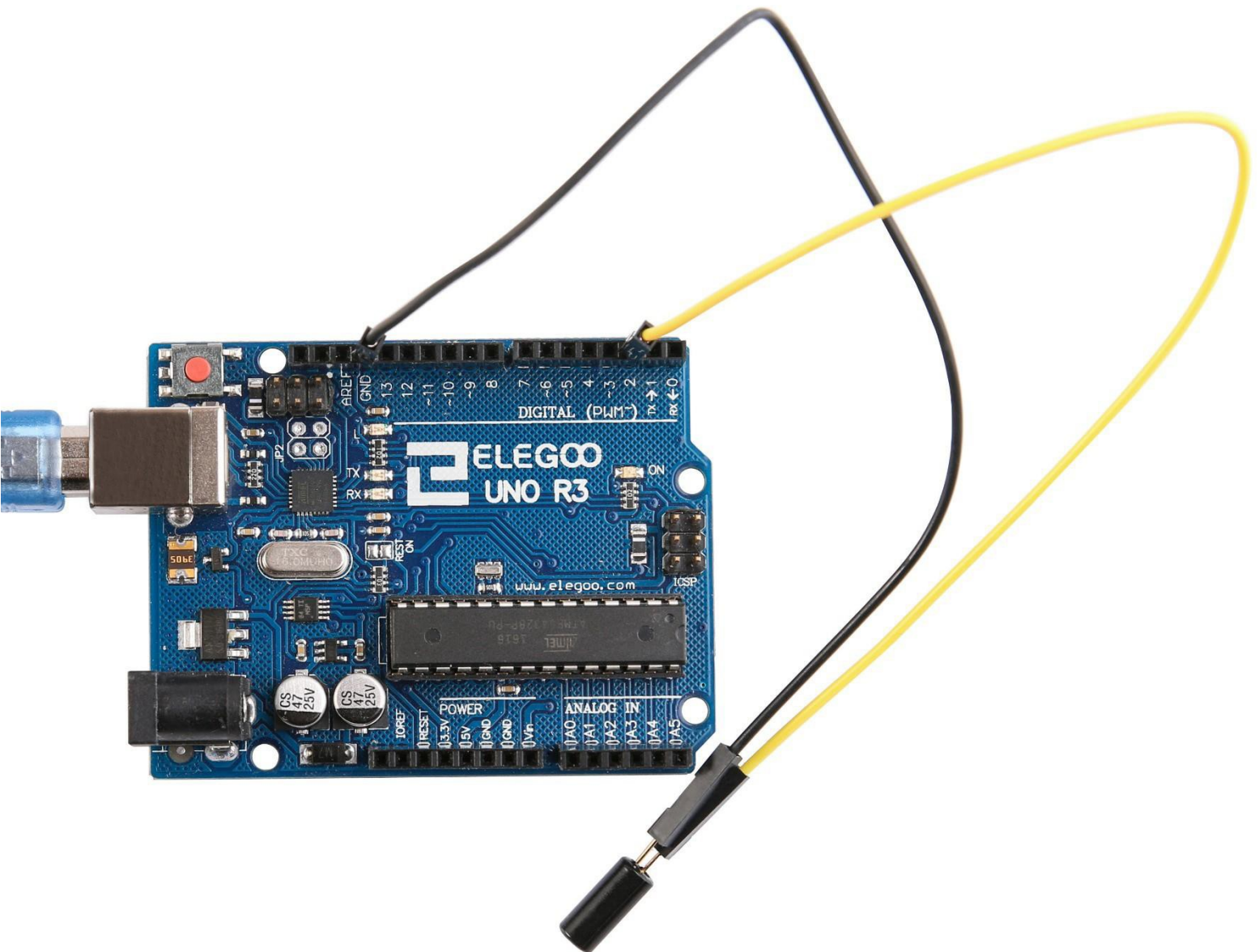
## Schema de cablare



## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 8 Ball Switch și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

## Imagine exemplu



## Lecția 9 Servo

### Prezentare generală

Servo este un tip de motorreductor care se poate roti doar 180 de grade. Este controlat prin trimiterea de impulsuri electrice de la placa dumneavoastră UNO R3. Aceste impulsuri îi spun servo-ului în ce poziție ar trebui să se miște. Servo are trei fire, dintre care cel maro este firul de împământare și trebuie conectat la portul GND al UNO, cel roșu este firul de alimentare și trebuie conectat la portul de 5v, iar cel portocaliu este firul de semnal și ar trebui să fie conectat la portul Dig #9.

### Componentă necesară:

(1) x Elegoo Uno R3

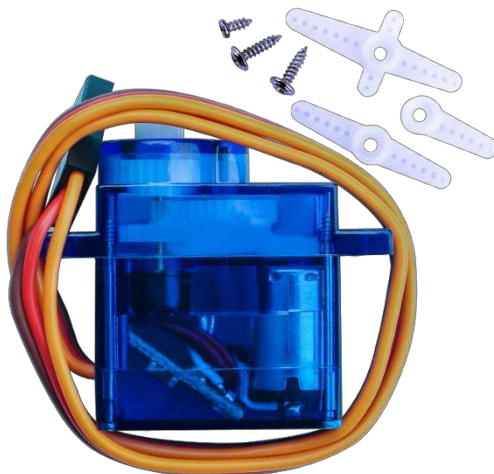
(1) x Servo (SG90)

(3) x fire MM (fire de legătură de tip masculin la masculin)

### Introducere componente

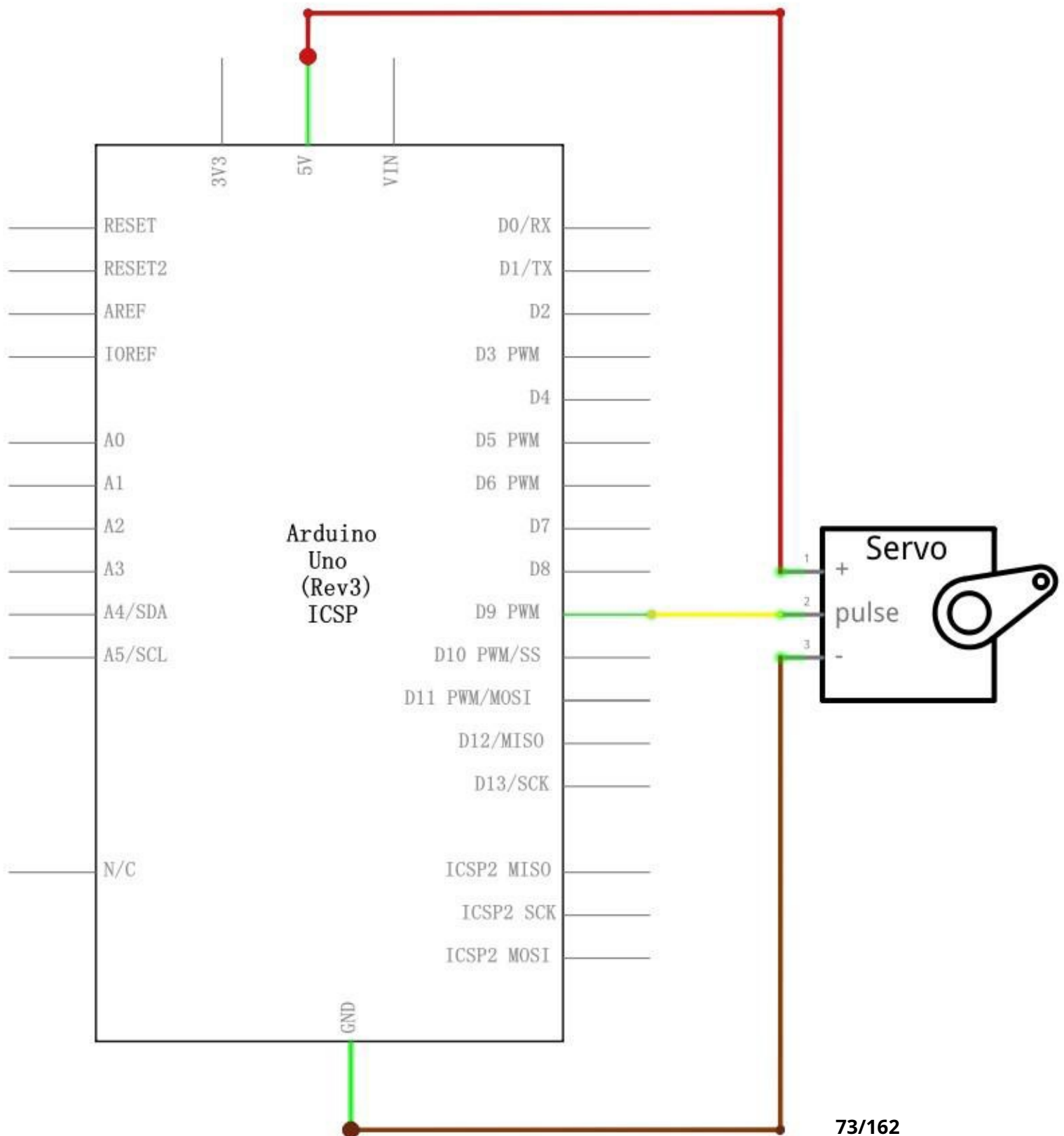
#### SG90

- Universal pentru conector JR și FP
- Lungime cablu: 25 cm
- Fără sarcină; Viteza de operare: 0,12 sec / 60 de grade (4,8 V), 0,10 sec / 60 de grade (6,0 V)
- Cuplu de oprire (4,8 V): 1,6 kg/cm
- Temperatura: -30~60°C
- Lățime de bandă moartă: 5us
- Tensiune de lucru: 3.5~6V
- Dimensiune: 3,2 cm x 3 cm x 1,2 cm (1,26 in x 1,18 in x 0,47 in)
- Greutate: 4,73 oz (134 g)

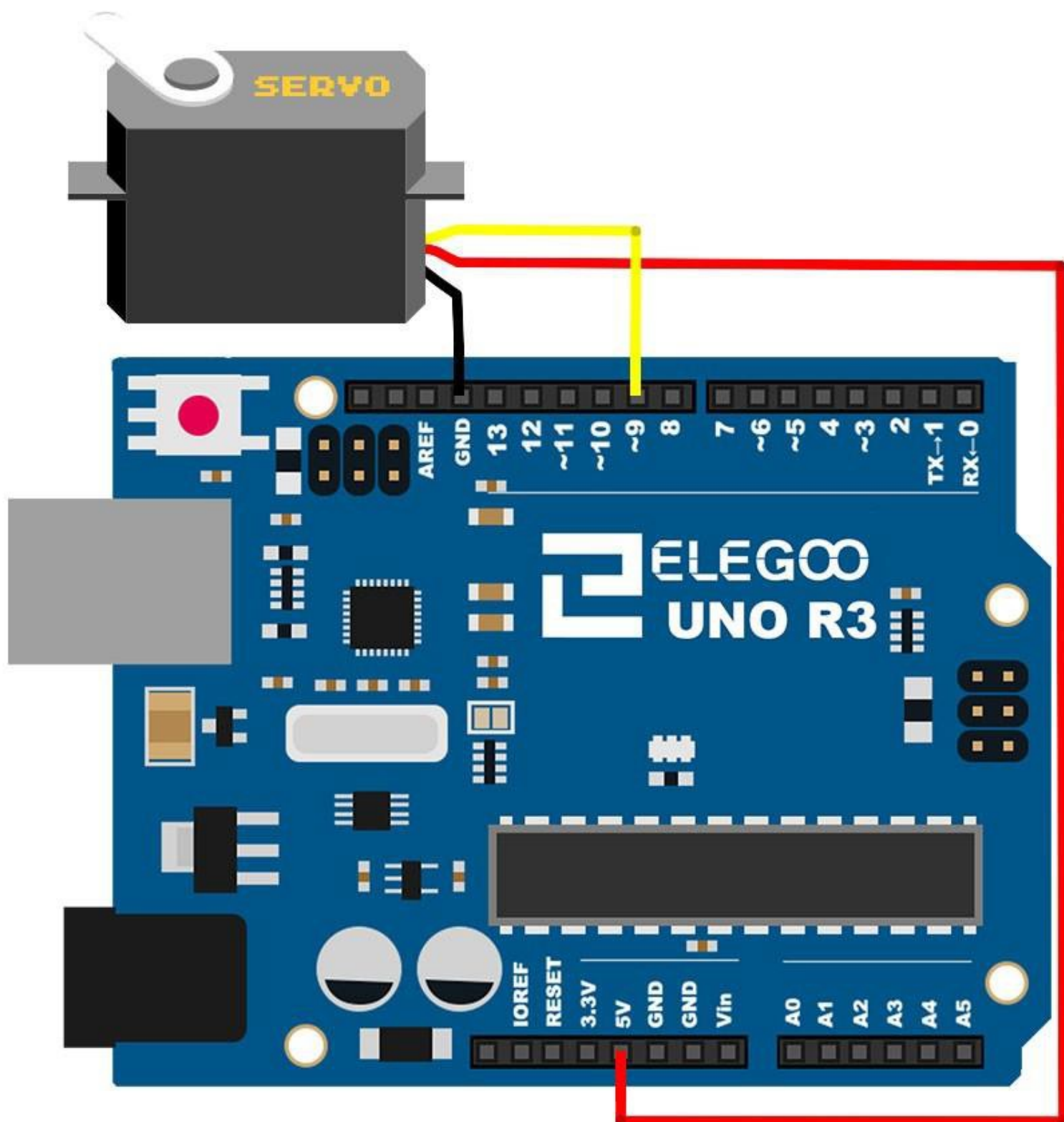


## Conexiune

### Schematic



Schema de cablare



## Cod

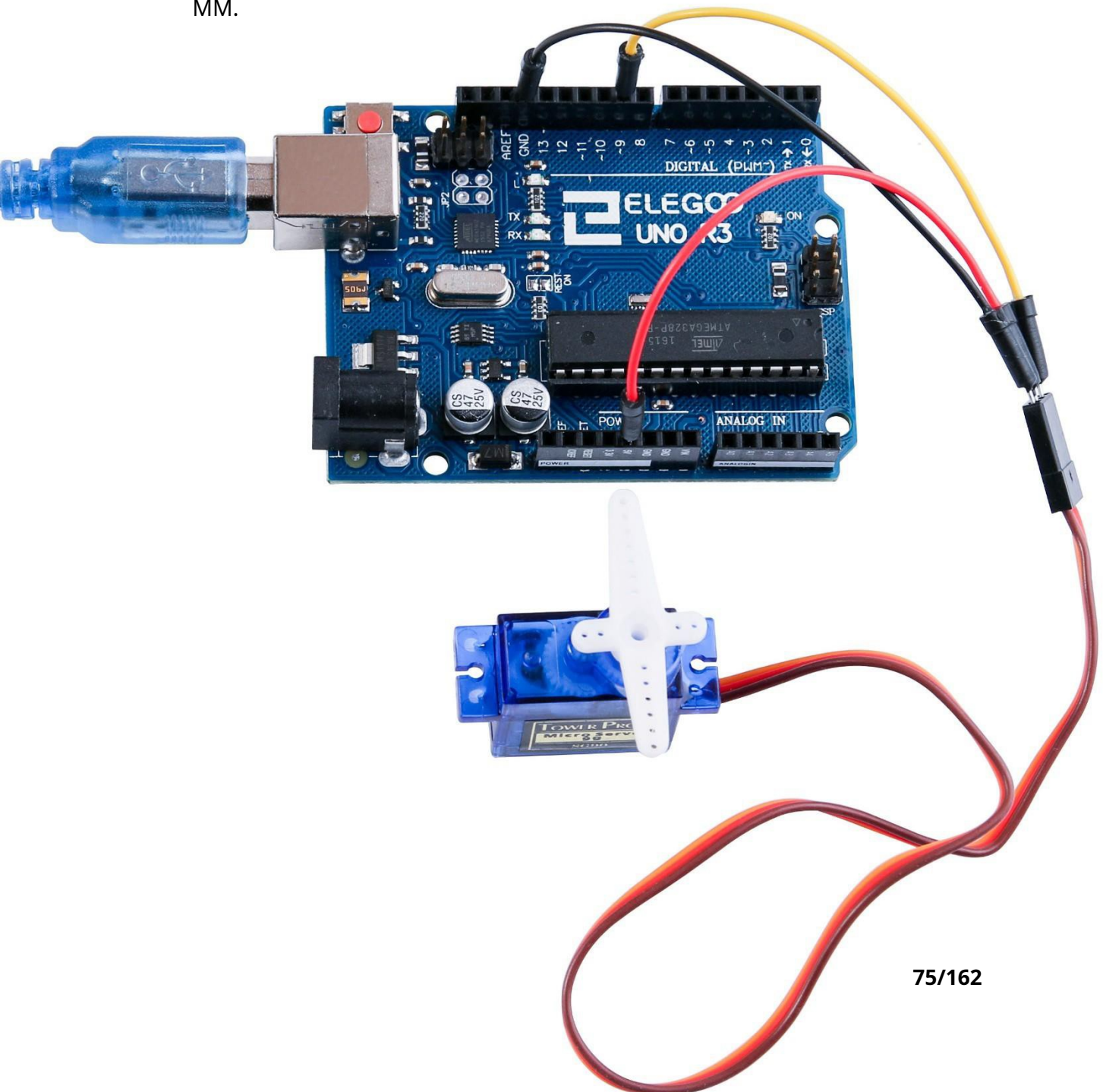
După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 9 Servo și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < Servo> sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați Lecția 1.

## Imagine exemplu

În imagine, firul maro al servo este adaptat prin firele negre MM, cel roșu este adaptat prin firele roșii MM, iar cel portocaliu este adaptat prin firele galbene MM.



## Lecția 10 Modulul senzorului cu ultrasunete

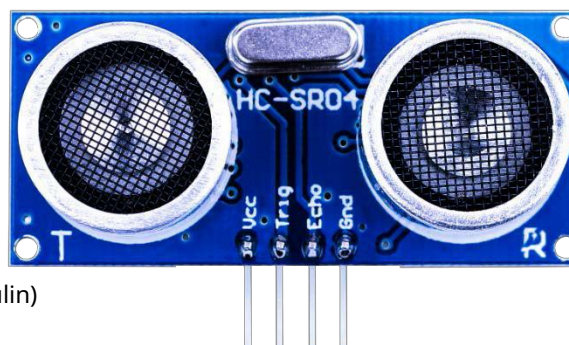
### Prezentare generală

Senzorul cu ultrasunete este excelent pentru toate tipurile de proiecte care necesită măsurători de distanță, evitând obstacolele ca exemple.

HC-SR04 este ieftin și ușor de utilizat, deoarece vom folosi o bibliotecă special concepută pentru acești senzori.

### Componentă necesară:

- (1) x Elegoo Uno R3 |
- (1) x Modul senzor cu ultrasunete
- (4) x fire FM (duPontwires de la femeie la masculin)



### Introducere componente

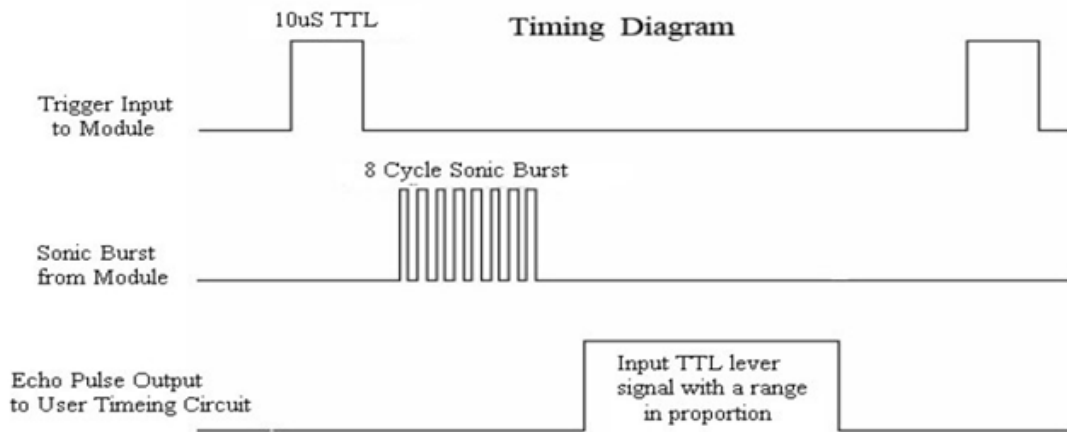
#### Senzor cu ultrasunete

Modulul senzor cu ultrasunete HC-SR04 oferă funcție de măsurare fără contact de 2 cm-400 cm, precizia de măsurare poate ajunge la 3 mm. Modulele includ transmițătoare cu ultrasunete, receptor și circuit de control. Principiul de bază al muncii:

- (1) Folosind declanșatorul IO pentru un semnal de nivel înalt de cel puțin 10us,
- (2) Modulul trimite automat opt 40 kHz și detectează dacă există un semnal de impuls înapoi.
- (3) DACĂ semnalul înapoi, la nivel înalt, timpul de ieșire ridicată a duratei IO este timpul de la trimiterea roților ultrasonice.

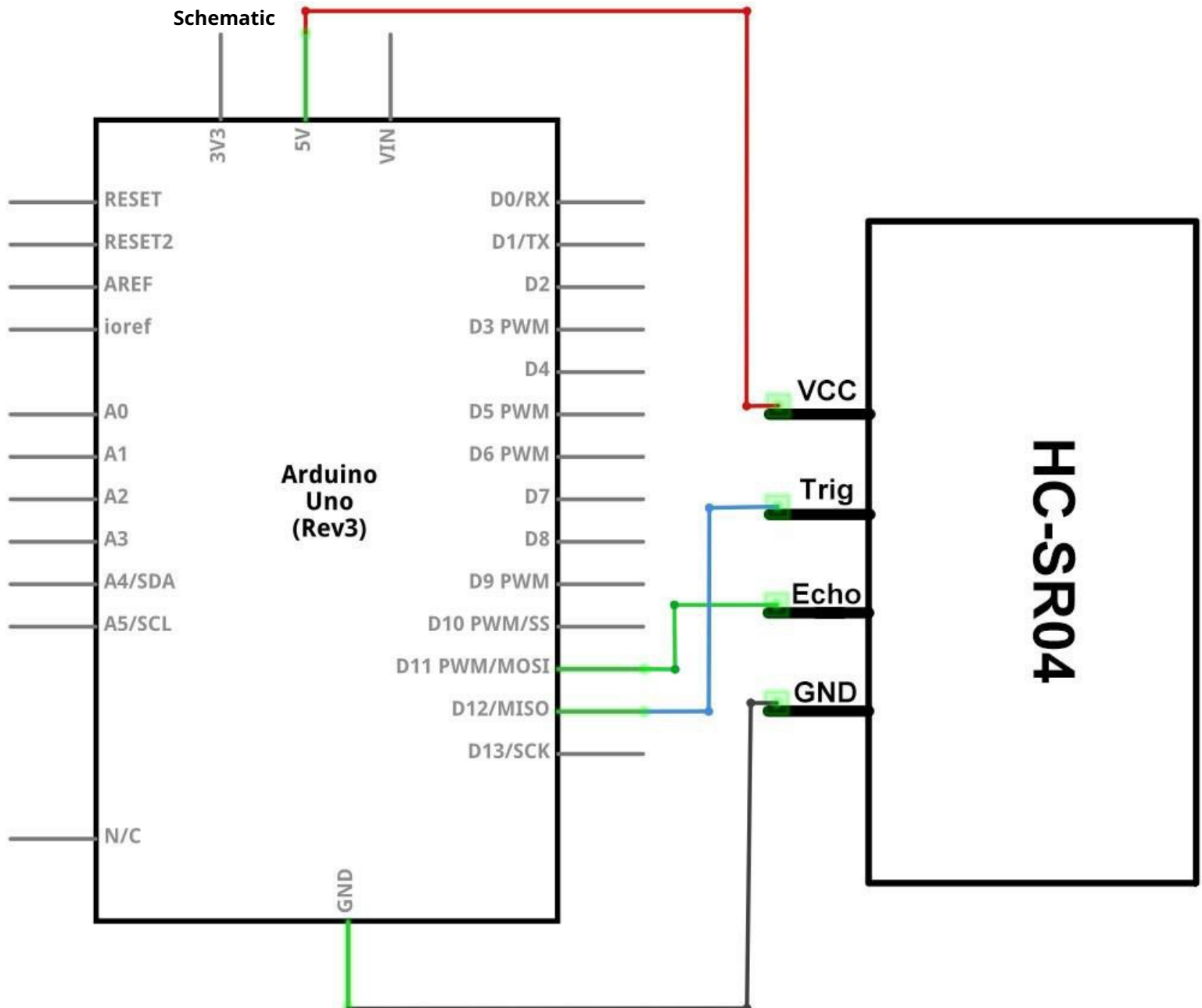
Distanța de testare = (timp de nivel înalt × viteza sunetului (340 m/s) / 2

Diagrama de timp este prezentată mai jos. Trebuie doar să furnizați un impuls scurt de 10 us la intrarea de declanșare pentru a începe intervalul, iar apoi modulul va trimite o rafală de ultrasunete cu 8 cicluri la 40 kHz și va ridica ecoul. Ecoul este un obiect de distanță care are lățimea pulsului și intervalul proporțional. Puteți calcula intervalul prin intervalul de timp dintre trimiterea semnalului de declanșare și primirea semnalului de eco. Formula:  $us / 58 = \text{centimetri}$  sau  $us / 148 = \text{inch}$ ; sau:  $\text{intervalul} = \text{timp de nivel înalt} * \text{viteza (340M/S)} / 2$ ; Vă sugerăm să utilizați un ciclu de măsurare de peste 60 ms, pentru a preveni semnalul de declanșare a semnalului de eco.

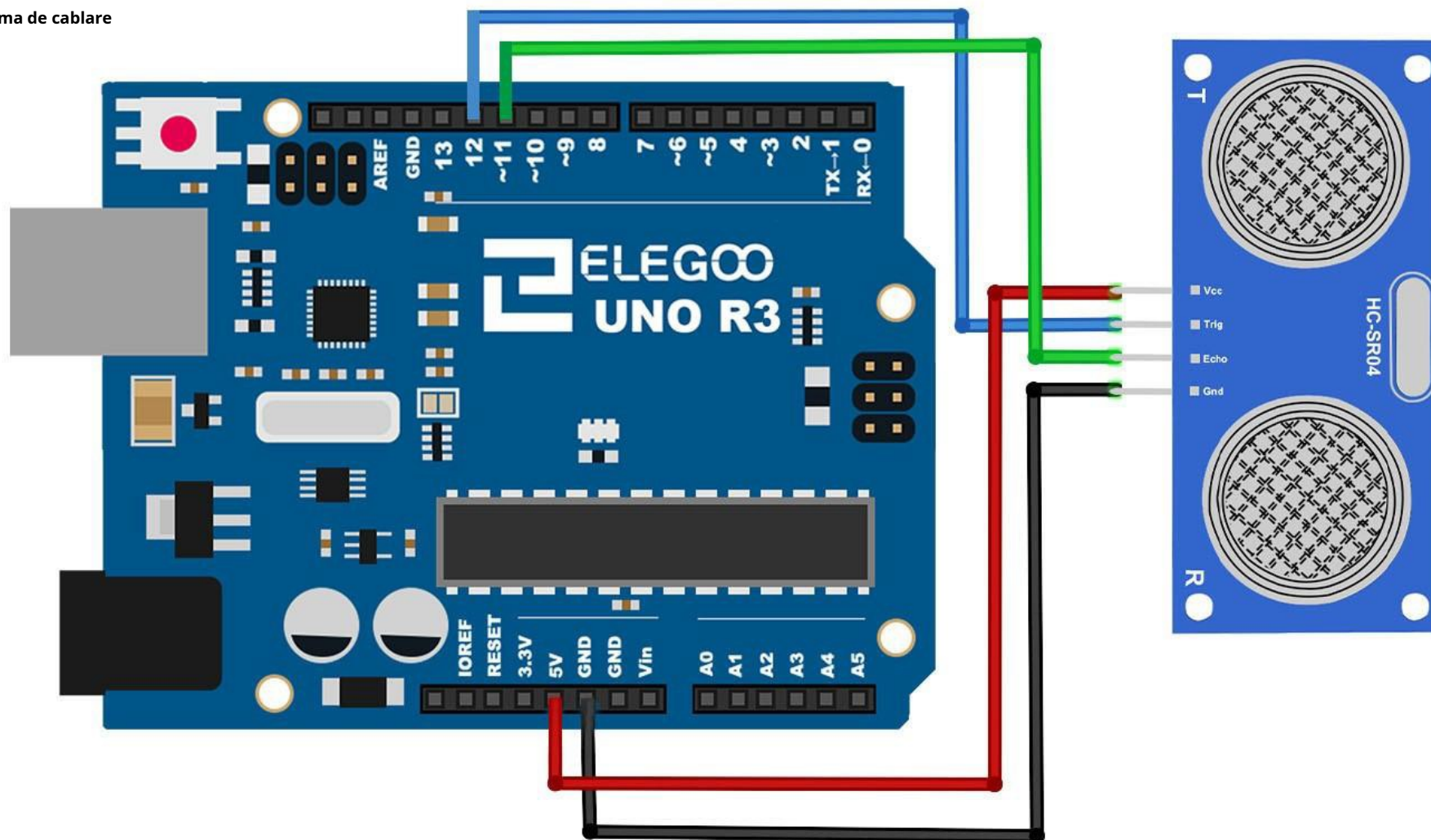


## Conexiune

### Schematic



Schema de cablare



## Cod

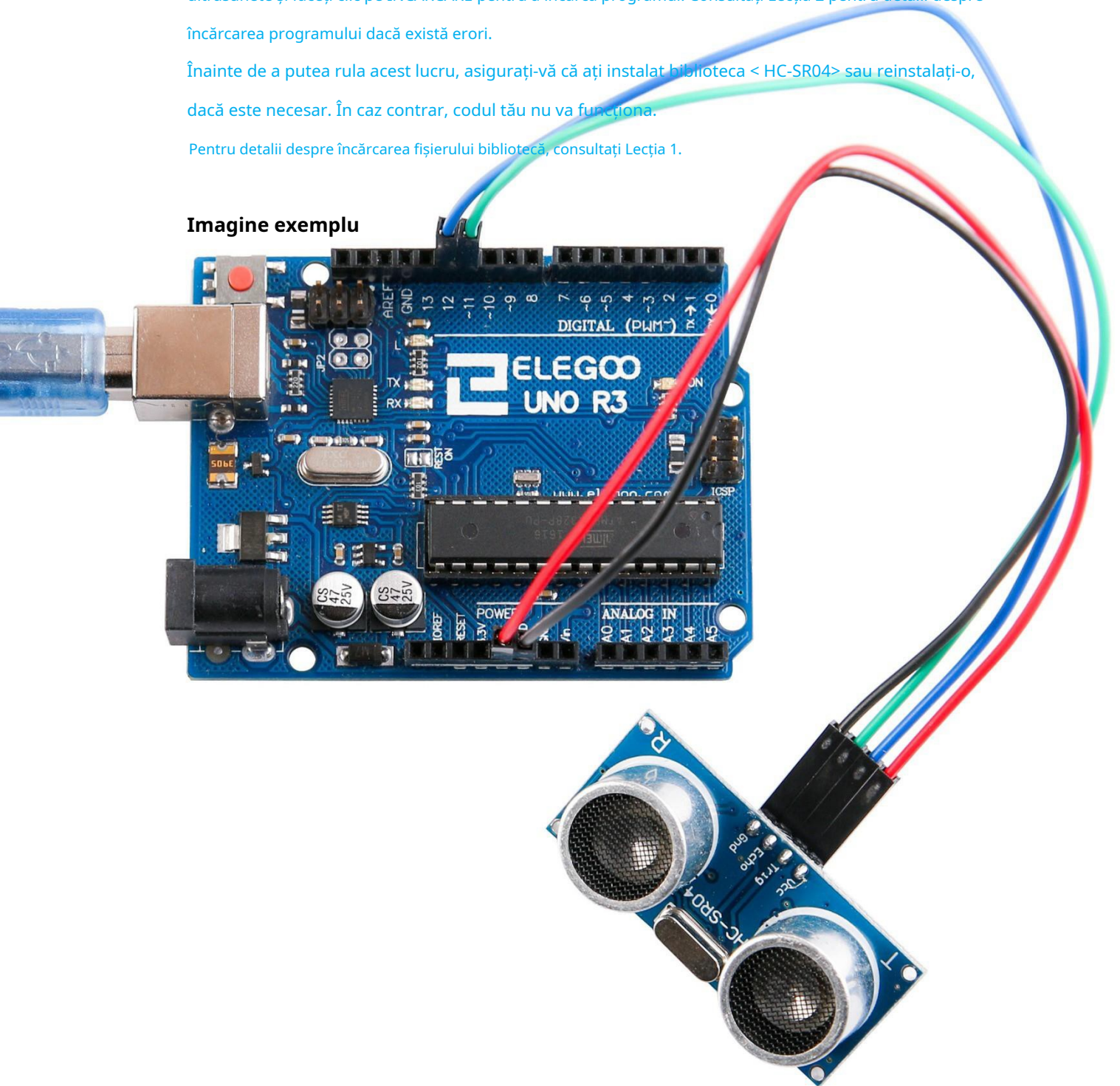
Utilizarea unei biblioteci concepute pentru acești senzori va face codul nostru scurt și simplu. Includem biblioteca la începutul codului nostru, iar apoi folosind comenzi simple putem controla comportamentul senzorului.

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 10 Modulul senzorului cu ultrasunete](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < HC-SR04 > sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

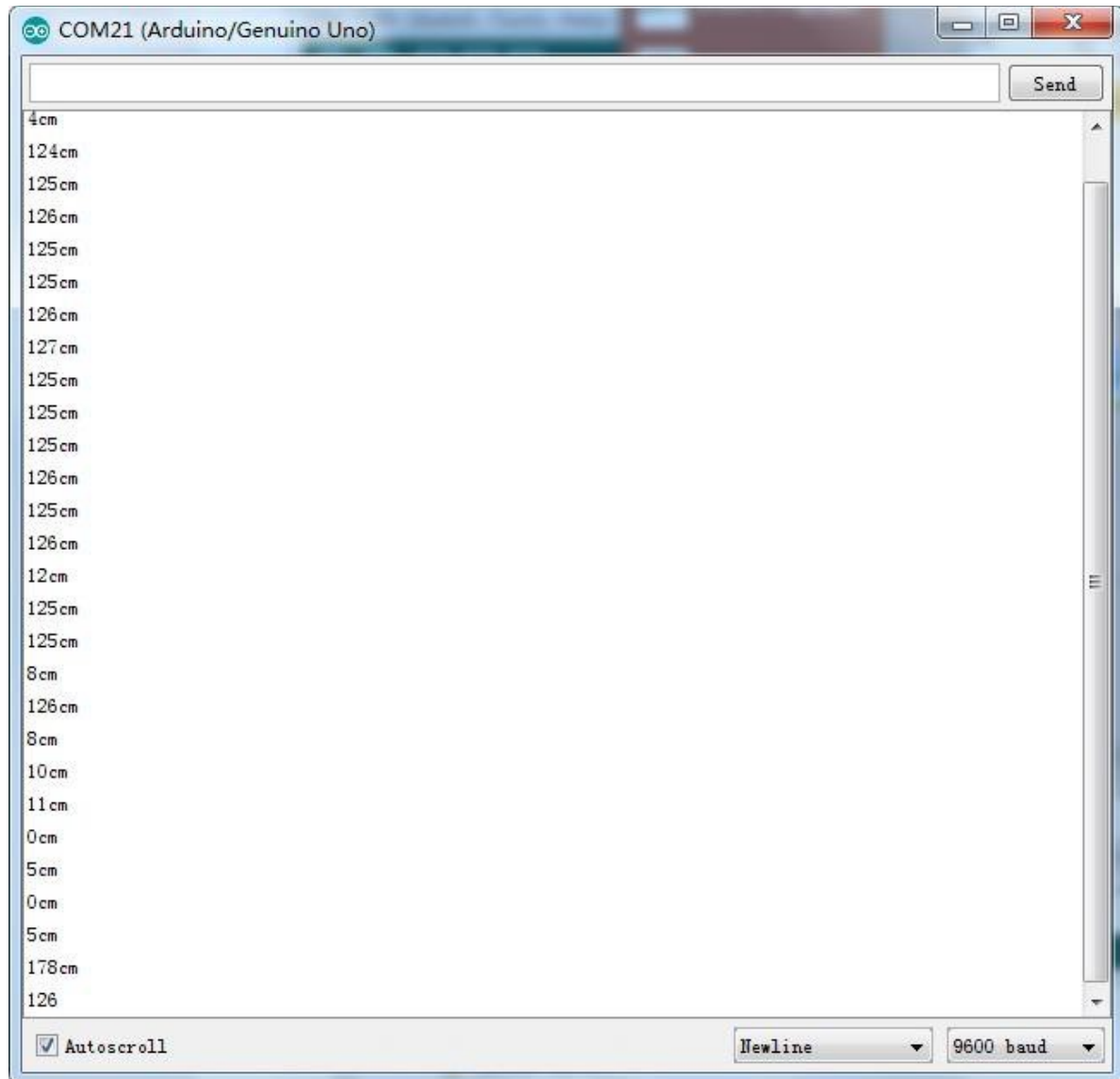
Pentru detalii despre încărcarea fișierului bibliotecă, consultați [Lecția 1](#).

## Imagine exemplu



Deschideți monitorul, apoi puteți vedea datele ca:

Faceți clic pe butonul Monitor serial pentru a porni monitorul serial. Elementele de bază despre monitorul serial sunt introduse în detalii în Lecția 1.



## Lecția 11 DHT11 Senzor de temperatură și umiditate

### Prezentare generală

În acest tutorial vom învăța cum să folosim un senzor de temperatură și umiditate DHT11. Este suficient de precis pentru majoritatea proiectelor care trebuie să țină evidența citirilor de umiditate și temperatură.

Din nou vom folosi o bibliotecă special concepută pentru acești senzori, care va face codul nostru scurt și ușor de scris.

### Componentă necesară:

(1) x Elegoo Uno R3

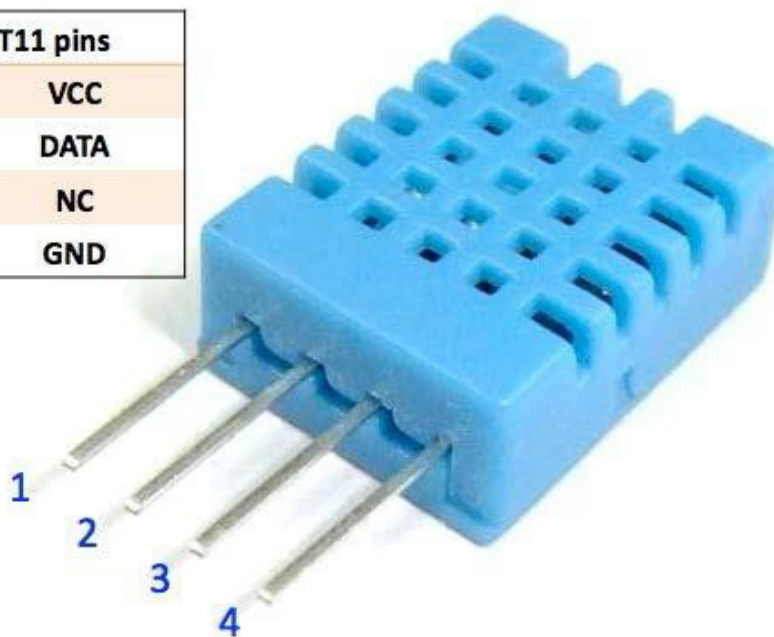
(1) x Modul de temperatură și umiditate DHT11

(3) x fire FM (duPontwires de la femeie la masculin)

### Introducere componente

Senzor de temperatură și umiditate:

| DHT11 pins |      |
|------------|------|
| 1          | VCC  |
| 2          | DATA |
| 3          | NC   |
| 4          | GND  |



Senzorul digital de temperatură și umiditate DHT11 este un senzor compozit care conține un semnal digital calibrat de ieșire a temperaturii și umidității. Tehnologia de colectare a modulelor digitale dedicate și tehnologia de detectare a temperaturii și umidității sunt aplicate pentru a se asigura că produsul are fiabilitate ridicată și

stabilitate excelentă pe termen lung. Senzorul include o senzație rezistivă a componentelor umede și un dispozitiv de măsurare a temperaturii NTC și se conectează la un microcontroler de înaltă performanță pe 8 biți.

Aplicații: HVAC, dezumidificator, echipamente de testare și inspecție, bunuri de larg consum, auto, control automat, înregistratoare de date, stații meteo, aparate electrocasnice, regulator de umiditate, măsurare și control medical și de altă natură a umidității.

Parametrii produsului

Umiditate relativa:

Rezoluție: 16 biți

Repetabilitate:  $\pm 1\%$  RH

Precizie: la  $25^{\circ}\text{C} \pm 5\%$  RH

Interschimbabilitate: complet interschimbabil

Timp de răspuns:  $1/e$  (63%) din  $25^{\circ}\text{C}$  6s

1m/s aer 6s

Histerezis:  $< \pm 0,3\%$  RH Stabilitate pe termen

lung:  $< \pm 0,5\%$  RH / an în temperatură:

Rezoluție: 16 biți

Repetabilitate:  $\pm 0,2^{\circ}\text{C}$

Interval: la  $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$

Timp de răspuns:  $1/e$  (63%) 10s

Caracteristici electrice

Alimentare: DC 3,5~5,5 V

Curent de alimentare: măsurare 0,3mA standby 60μA Perioada

de eșantionare: mai mult de 2 secunde

Descrierea pinului:

1. sursa de alimentare VDD 3.5~5,5 V DC

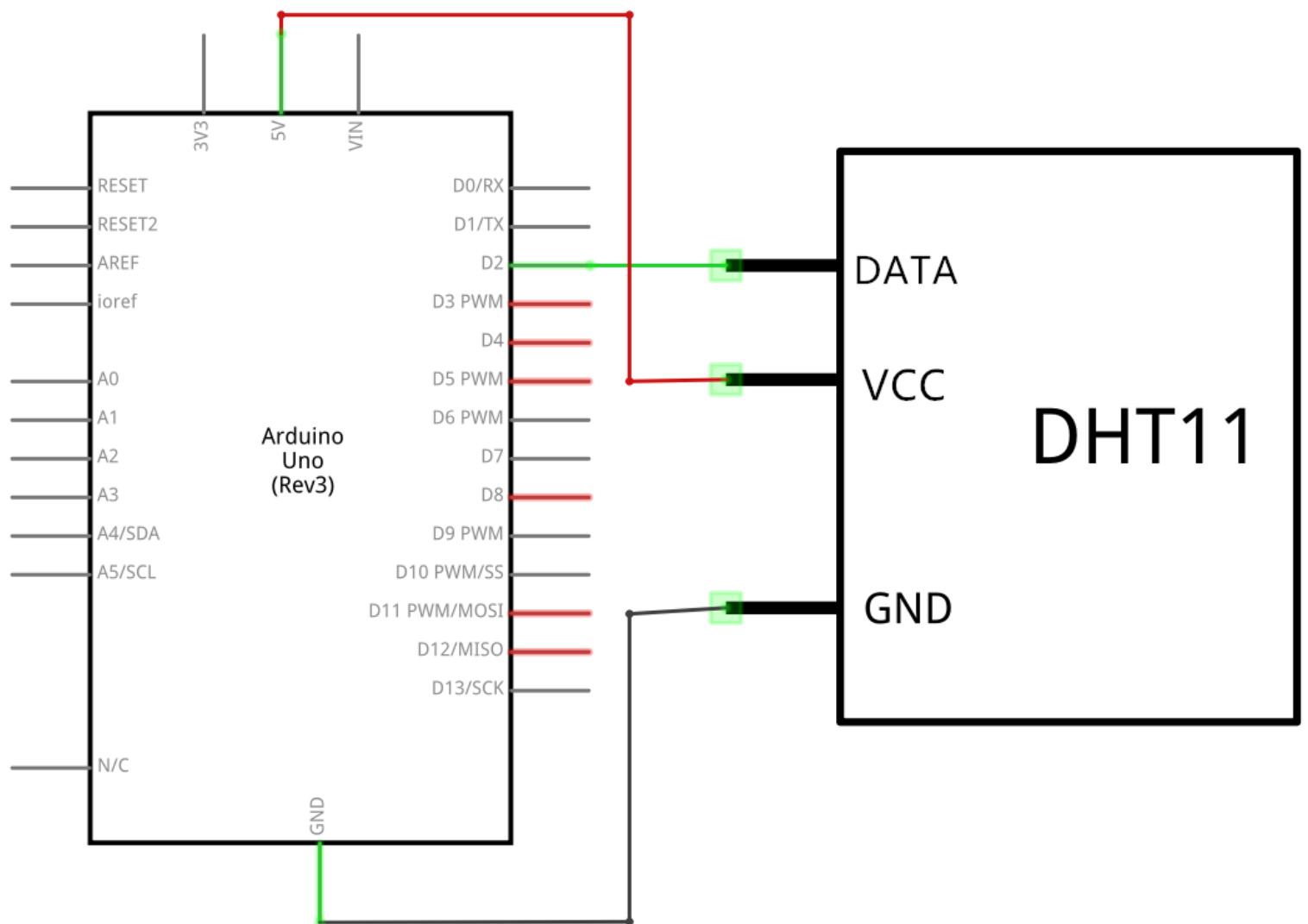
2. DATE date seriale, o singură magistrală

3. NC, pin gol

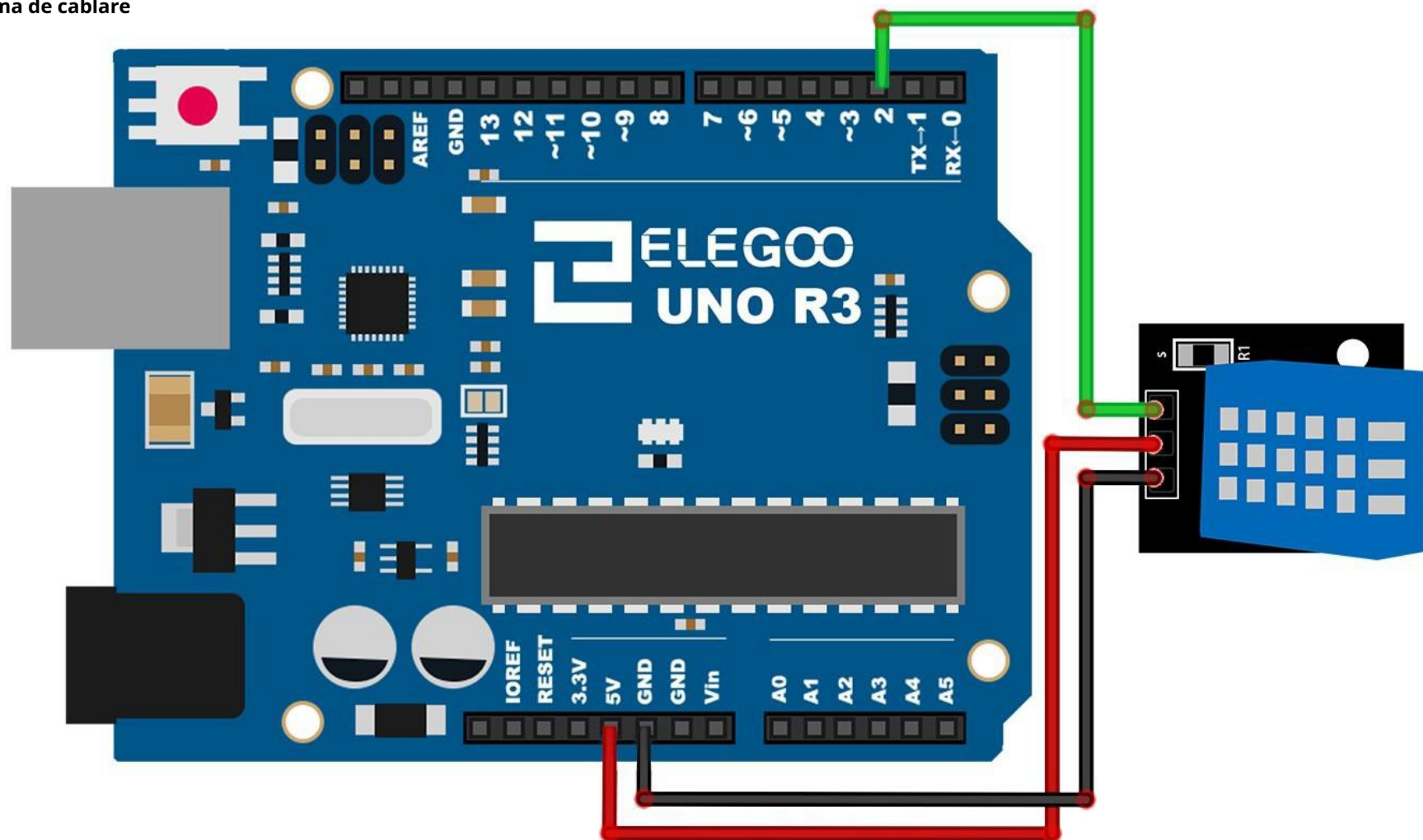
4. Masa GND, puterea negativă

## Conexiune

### Schematic



Schema de cablare



După cum puteți vedea, avem nevoie doar de 3 conexiuni la senzor, deoarece unul dintre pin nu este folosit.

Conexiunile sunt: Tensiune, Masă și Semnal care pot fi conectate la orice Pin de pe UNO.

## Cod

După cablare, deschideți programul în folderul de coduri - [Lecția 11 Senzorul de temperatură și umiditate DHT11](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul.

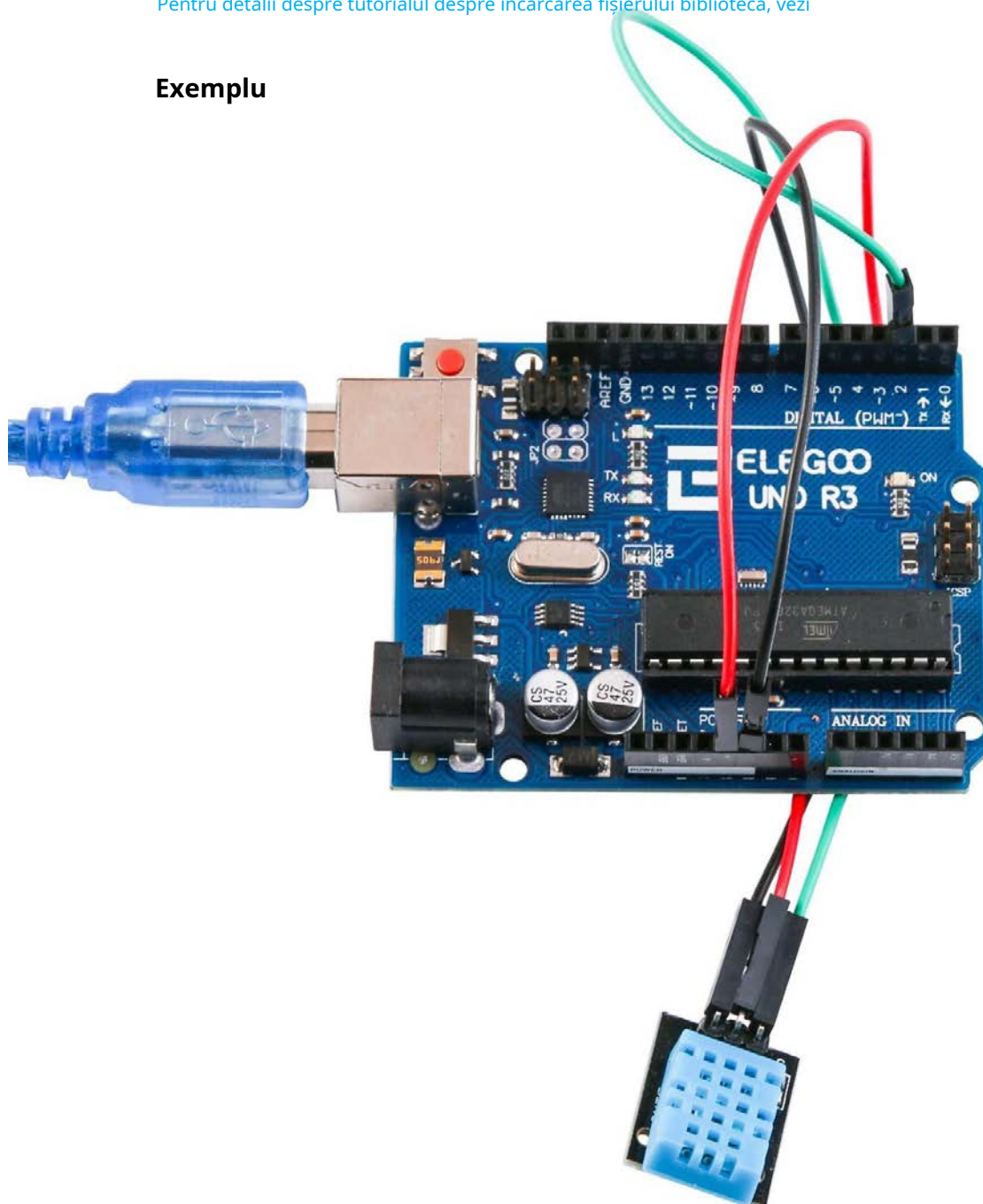
Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < DHT >

sau reinstalați-l, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre tutorialul despre încărcarea fișierului bibliotecă, vezi

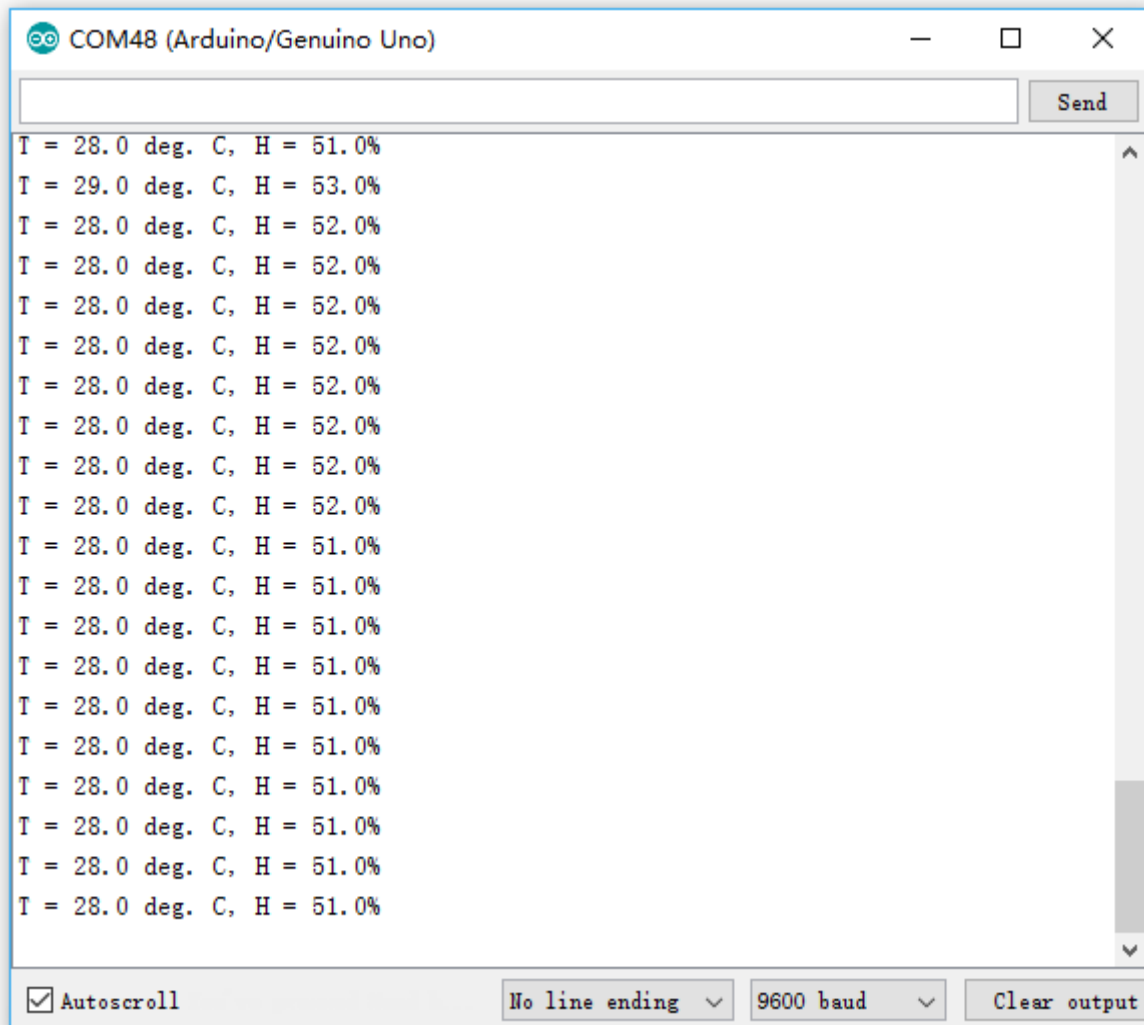
## Exemplu



Încărcați programul apoi deschideți monitorul, putem vedea datele după cum urmează:

(Afișează temperatura mediului, putem vedea că este de 28 de grade)

Faceți clic pe butonul [Monitor serial](#) pentru a porni monitorul serial. Elementele de bază despre monitorul serial sunt introduse în detalii în [Lecția 1](#).



## Lecția 12 Modulul Joystick analogic

### Prezentare generală

Joystick-urile analogice sunt o modalitate excelentă de a adăuga ceva control în proiectele dvs. În acest tutorial vom învăța cum să folosim modulul joystick analogic.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x modul Joystick

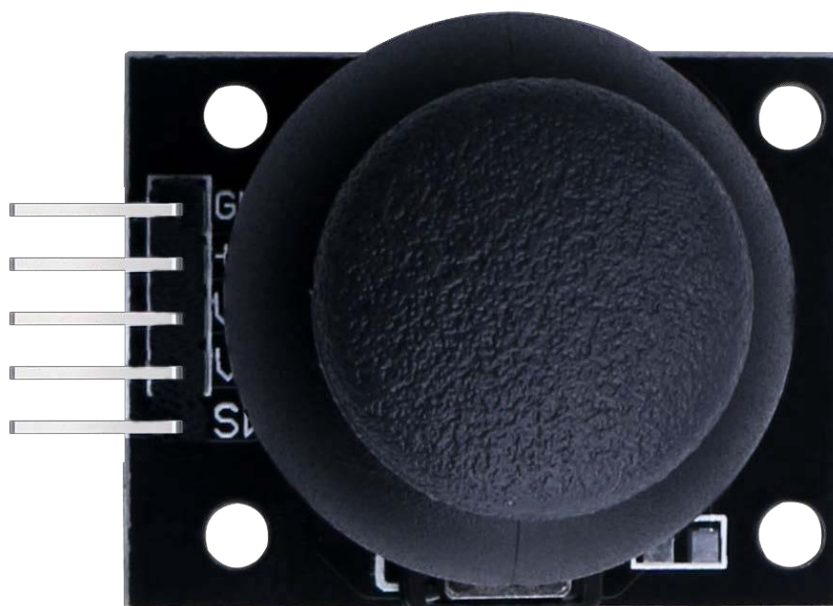
(5) x fire FM (duPontwires de la femeie la masculin)

### Introducere componente

#### Joystick

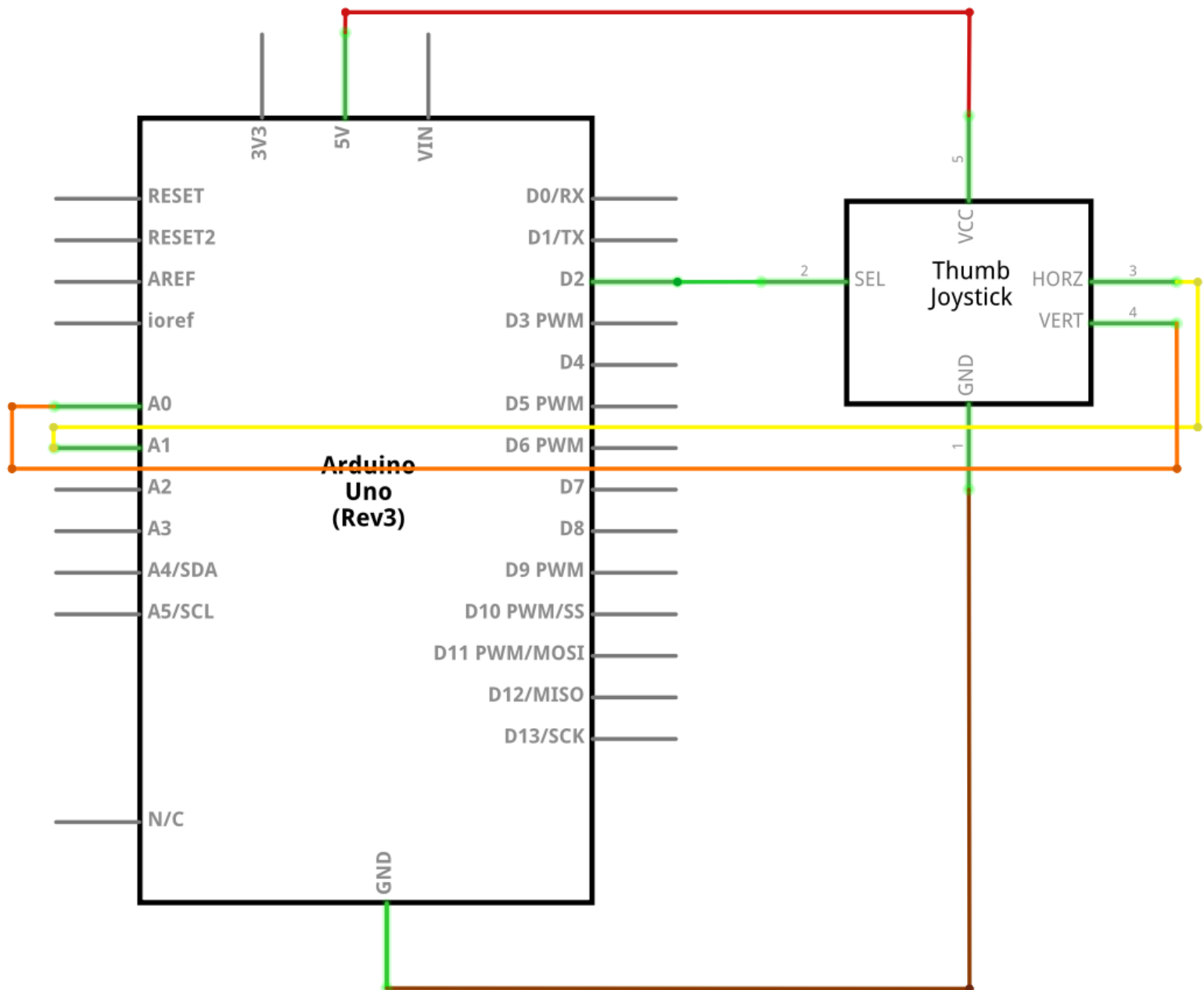
Modulul are 5 pini: VCC, Ground, X, Y, Key. Rețineți că etichetele de pe ale dvs. pot fi ușor diferite, în funcție de locul de unde ați luat modulul. Stick-ul pentru degetul mare este analogic și ar trebui să ofere citiri mai precise decât simplele joystick-uri „direcționale” tact folosesc anumite forme de butoane sau comutatoare mecanice. În plus, puteți apăsa joystick-ul în jos (mai degrabă tare pe al meu) pentru a activa un buton „apăsați pentru a selecta”.

Trebuie să folosim pini Arduino analogici pentru a citi datele de la pinii X/Y și un pin digital pentru a citi butonul. Pinul cheii este conectat la masă, atunci când joystick-ul este apăsat în jos, și în caz contrar plutește. Pentru a obține citiri stabile de la pinul Key/Select, acesta trebuie să fie conectat la VCC printr-un rezistor pull-up. Pot fi utilizate rezistențele încorporate pe pinii digitali Arduino. Pentru un tutorial despre cum să activați rezistențele pull-up pentru pinii Arduino, configurați ca intrări

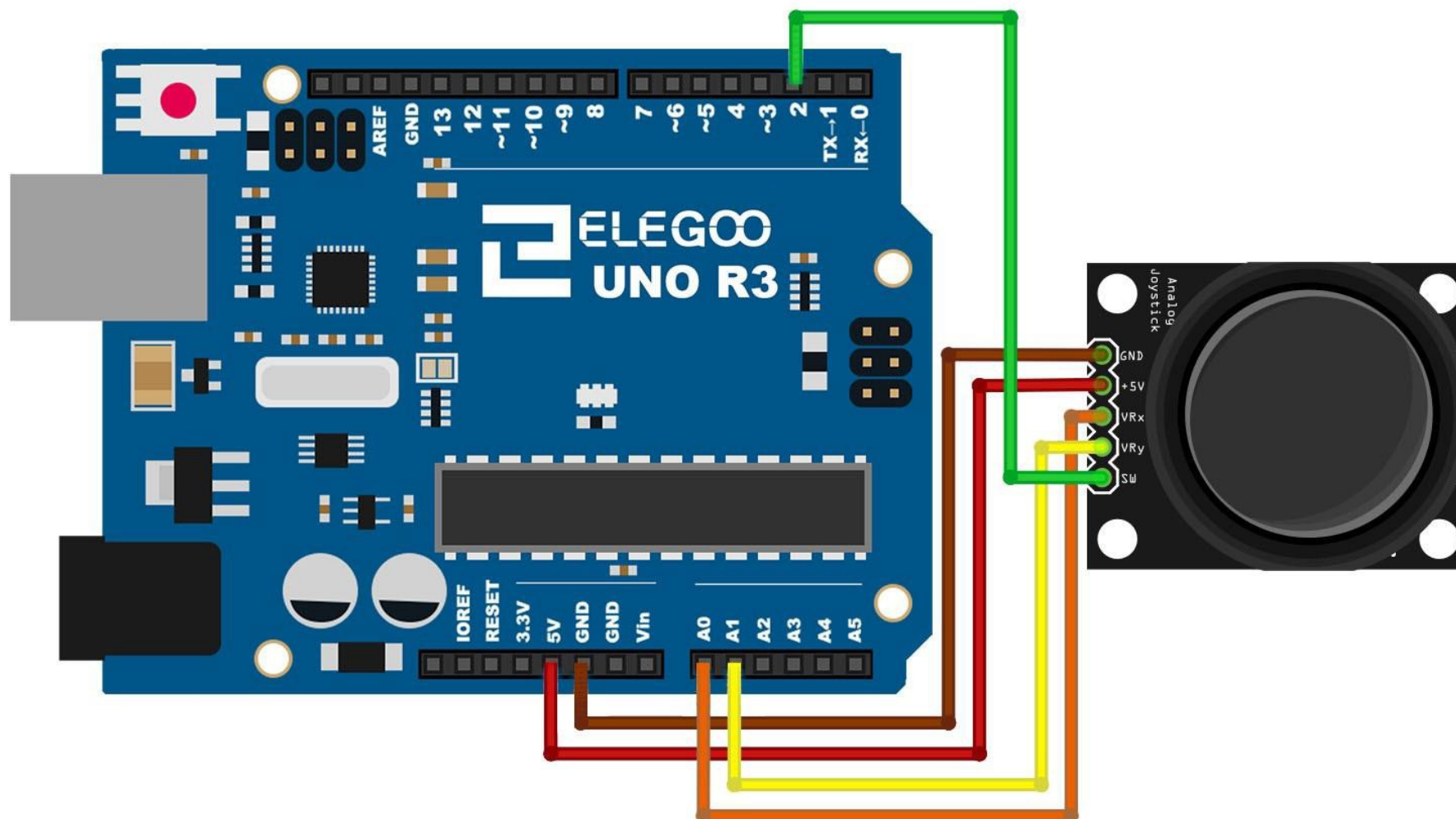


## Conexiune

### Schematic



Schema de cablare



Avem nevoie de 5 conexiuni la joystick.

Conexiunile sunt: Cheie, Y, X, Tensiune și Masă.

„Y și X” sunt analogice, iar „Cheie” este digitală. Dacă nu aveți nevoie de comutator, puteți utiliza doar 4 pini.

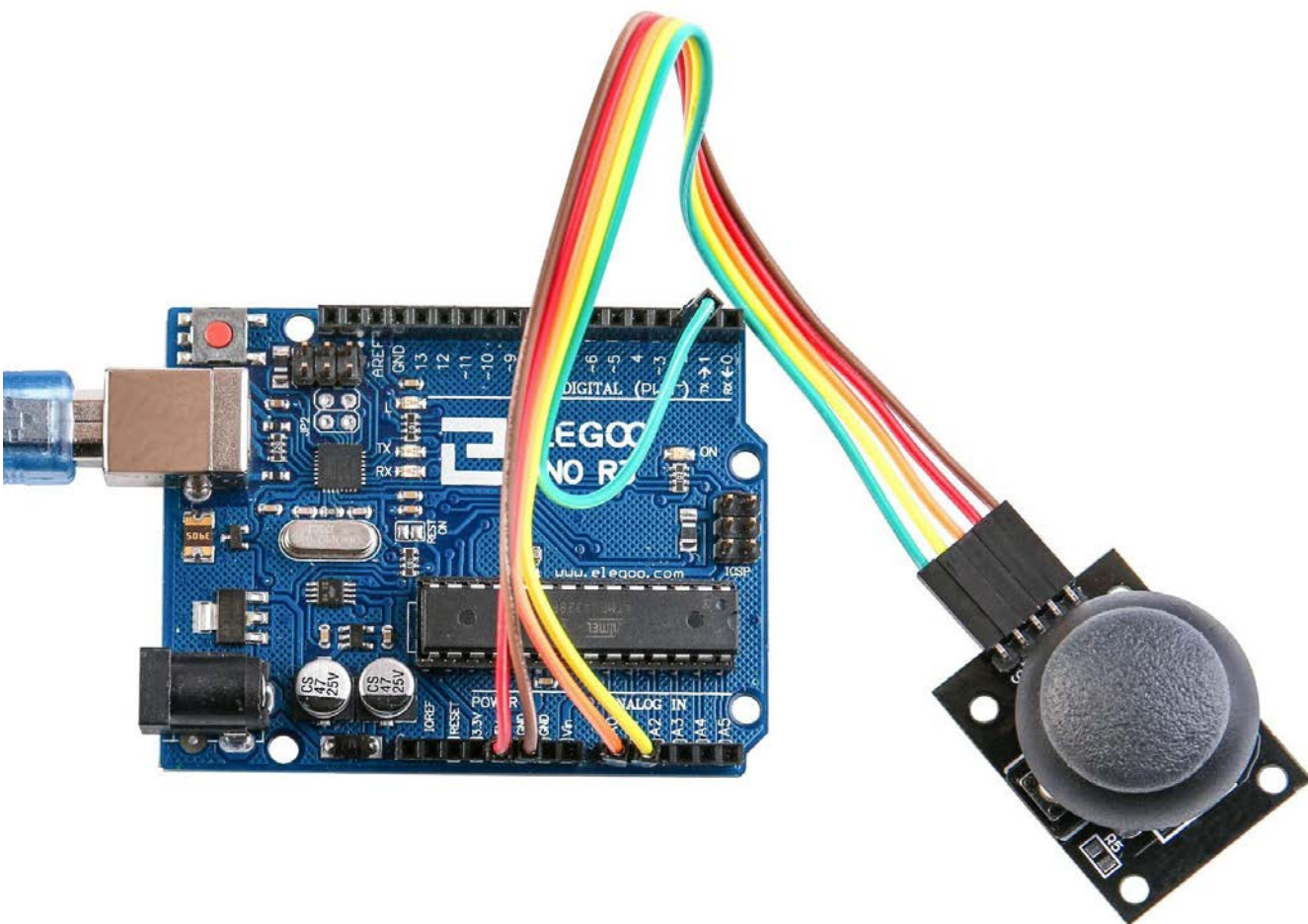
## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Modulul Joystick Analog Lecția 12 și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Joystick-urile analogice sunt practic potențiometre, astfel încât returnează valori analogice. Când joystick-ul este în poziția de repaus sau în mijloc, ar trebui să returneze o valoare de aproximativ 512.

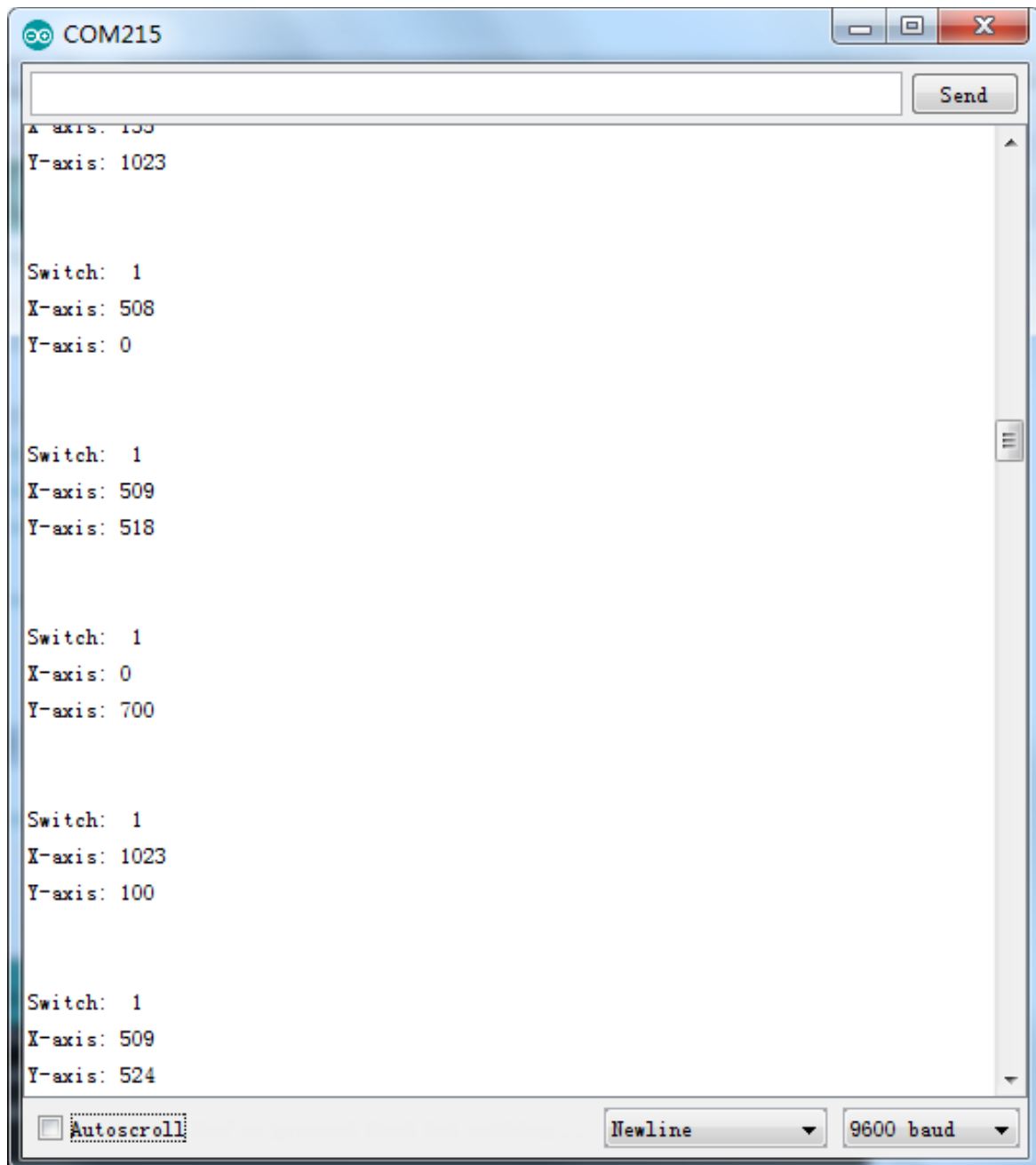
Intervalul de valori variază de la 0 la 1024.

## Imagine exemplu



Deschideți monitorul, apoi puteți vedea datele ca:

Faceți clic pe butonul Monitor serial pentru a porni monitorul serial. Elementele de bază despre monitorul serial sunt introduse în detalii în Lecția 1.



## Lecția 13 Modulul receptor IR

### Prezentare generală

Utilizarea unei telecomenzi IR este o modalitate excelentă de a avea control fără fir asupra proiectului dvs. Telecomenzile cu infraroșu sunt simple și ușor de utilizat. În acest tutorial vom conecta receptorul IR la UNO și apoi vom folosi o bibliotecă care a fost concepută pentru acest senzor special.

În schița noastră vom avea toate codurile hexazecimale IR care sunt disponibile pe această telecomandă, vom detecta și dacă codul a fost recunoscut și, de asemenea, dacă ținem apăsată o tastă.

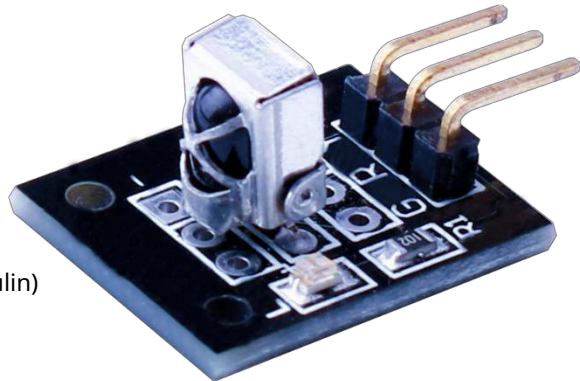
### Componentă necesară:

(1) x Elegoo Uno R3

(1) x modul receptor IR

(1) x telecomandă IR

(3) x fire FM (duPontwires de la femeie la masculin)



### Introducere componente

#### SENSOR RECEPTOR IR:

Detectoarele IR sunt mici microcipuri cu o celulă foto care sunt reglate pentru a asculta lumina infraroșie. Ele sunt aproape întotdeauna folosite pentru detectarea telecomenzii - fiecare televizor și DVD player are unul dintre acestea în față pentru a asculta semnalul IR de la clicker. În interiorul telecomenzii se află un LED IR potrivit, care emite impulsuri IR pentru a spune televizorului să pornească, să oprească sau să schimbe canalele. Lumina IR nu este vizibilă pentru ochiul uman, ceea ce înseamnă că este nevoie de ceva mai multă muncă pentru a testa o configurație.

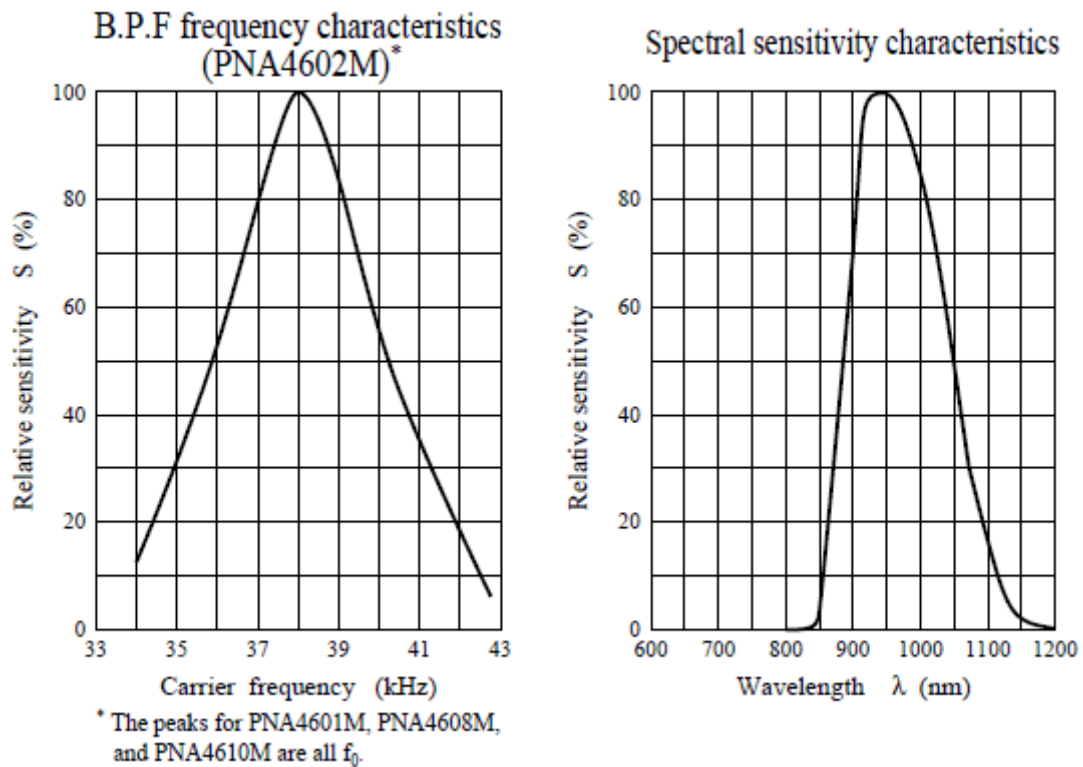
Există câteva diferențe între acestea și spunem o fotocelule CdS:

Detectoarele IR sunt filtrate special pentru lumina IR, nu sunt bune la detectarea luminii vizibile. Pe de altă parte, fotocelulele sunt bune la detectarea luminii vizibile galbene/verde și nu sunt bune la lumina IR.

Detectoarele IR au un demodulator în interior care caută IR modulată la 38 KHz. Doar aprinderea unui LED IR nu va fi detectată, trebuie să fie PWM care clipește la 38KHz. Fotocelulele nu au nici un fel de demodulator și pot detecta orice frecvență (inclusiv DC) în viteză de răspuns a fotocelulei (care este de aproximativ 1KHz)

Detectoarele IR au ieșire digitală - fie detectează semnal IR de 38KHz și ies la emisie scăzută (0V),  
fie nu detectează niciunul și ies la nivel ridicat (5V). Fotocelulele acționează ca niște rezistențe,  
rezistența se modifică în funcție de cât de multă lumină sunt expuse.

## Ce poți măsura

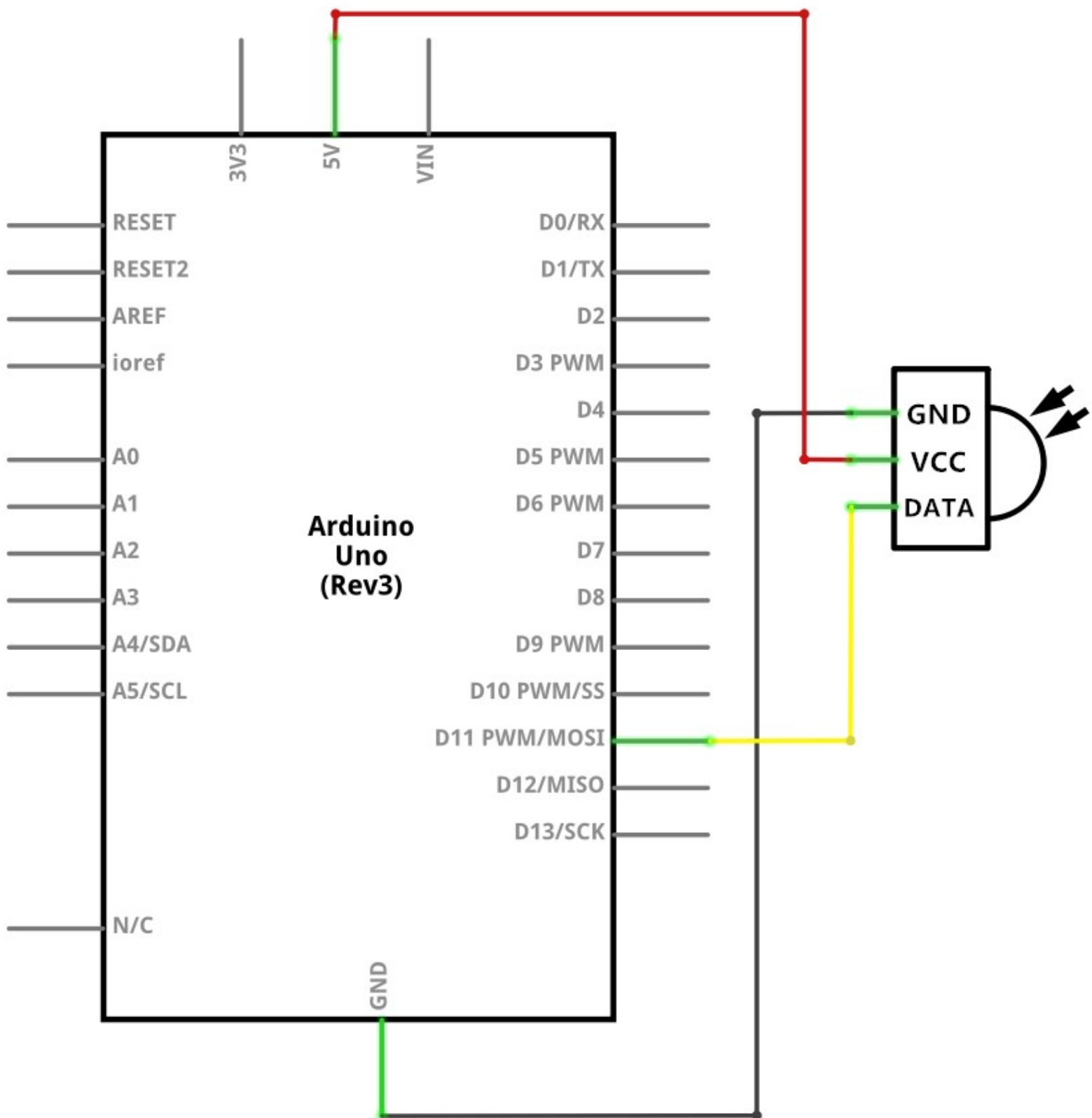


După cum puteți vedea din aceste grafice de date, detectarea frecvenței de vârf este la 38 KHz, iar culoarea LED-ului de vârf este de 940 nm. Puteți utiliza de la aproximativ 35 KHz la 41 KHz, dar sensibilitatea va scădea astfel încât să nu detecteze la fel de bine de la distanță. De asemenea, puteți utiliza LED-uri de la 850 la 1100 nm, dar acestea nu vor funcționa la fel de bine ca și la 900 la 1000 nm, așa că asigurați-vă că aveți LED-uri potrivite! Verificați fișa de date pentru LED-ul IR pentru a verifica lungimea de undă.

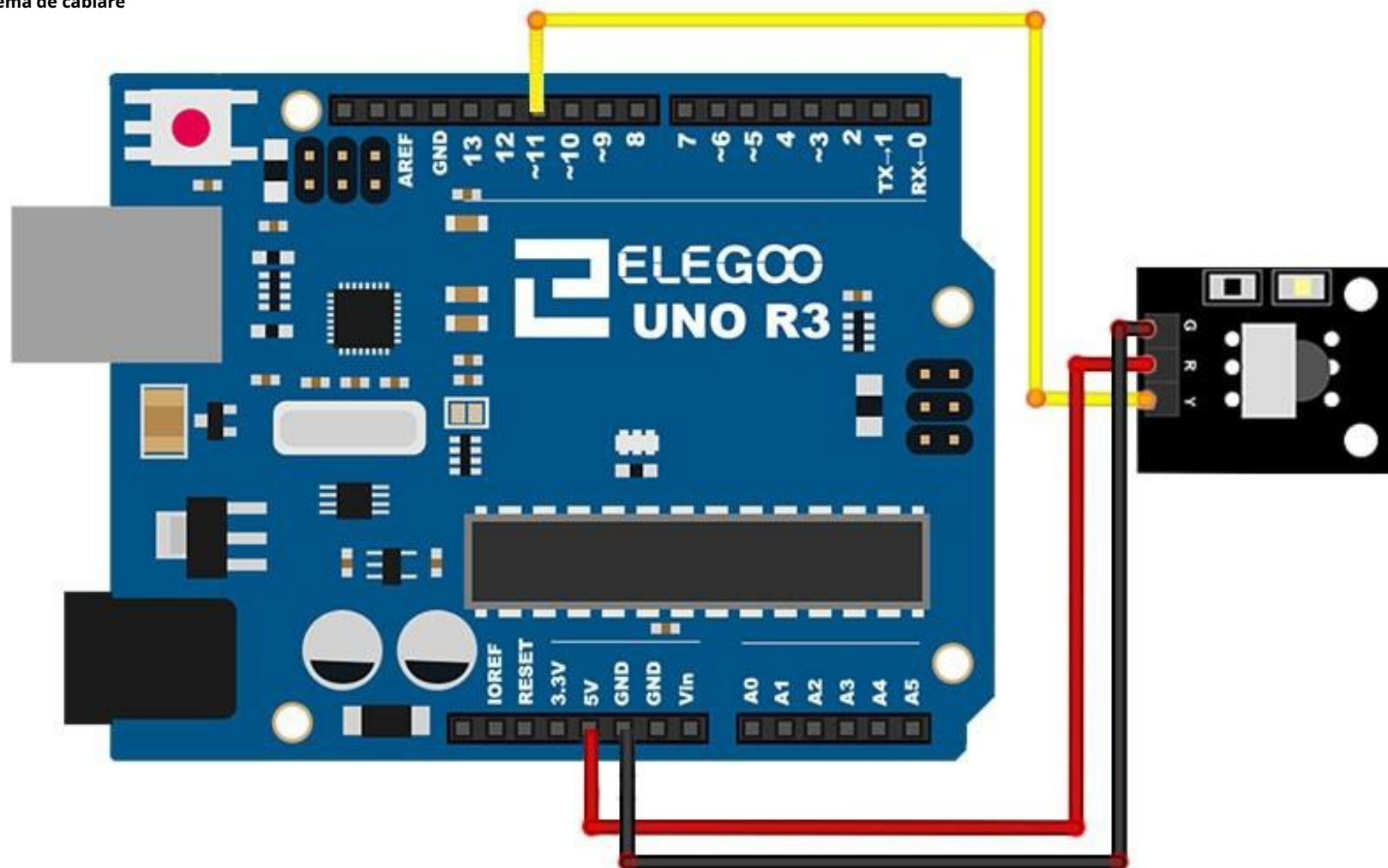
Încercați să obțineți un 940nm - amintiți-vă că 940nm nu este lumină vizibilă!

## Conexiune

### Schematic



Schema de cablare



Există 3 conexiuni la receptorul IR.

Conexiunile sunt: semnal, tensiune și masă.

„-” este masa, „S” este semnalul, iar pinul din mijloc este tensiunea de 5V.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 13 Modulul receptor IR](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

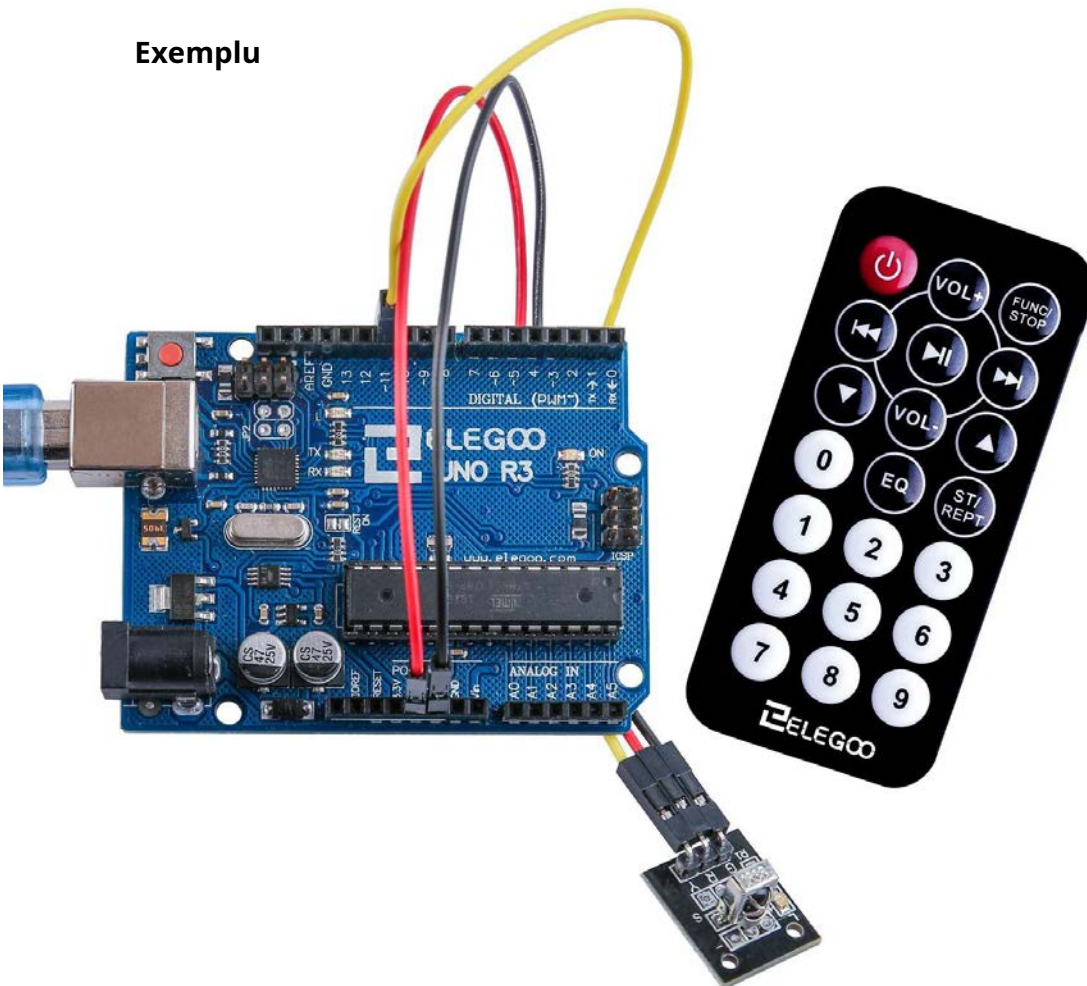
Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca <IRremote> sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați [Lecția 1](#).

În continuare, vom muta <RobotIRremote> din folderul Bibliotecă, facem asta deoarece biblioteca respectivă intră în conflict cu cea pe care o vom folosi. Puteți să-l trageți înapoi în dosarul bibliotecii după ce ați terminat de programat microcontrolerul.

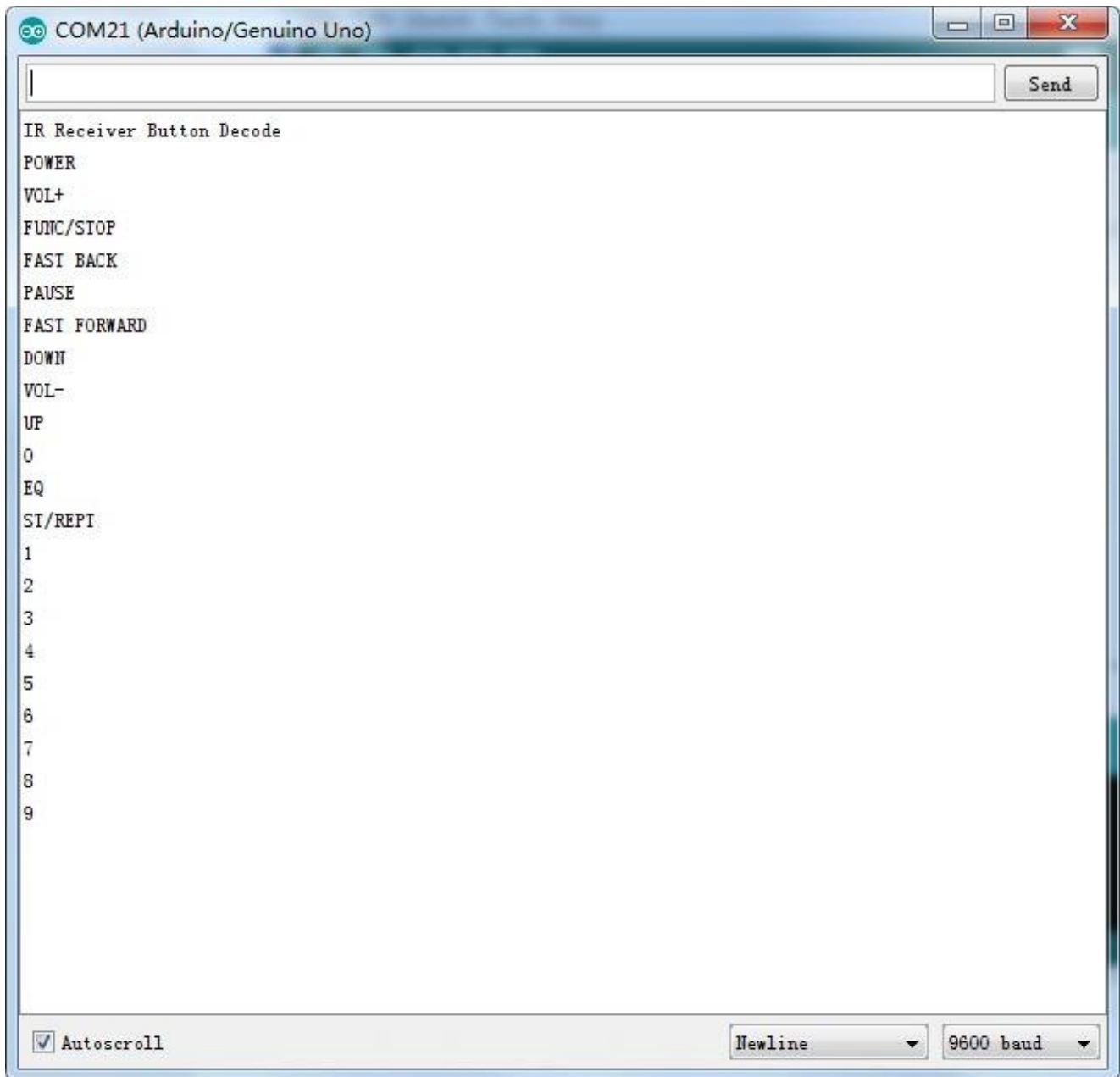
După ce ați instalat Biblioteca, continuați și reporniți-vă

## Exemplu



Deschideți monitorul, apoi puteți vedea datele ca:

Faceți clic pe butonul Monitor serial pentru a porni monitorul serial. Elementele de bază despre monitorul serial sunt introduse în detalii în Lecția 1.



## Lecția 14 Display LCD

### Prezentare generală

În această lecție, veți învăța cum să conectați și să utilizați un afișaj LCD alfanumeric.

Ecranul are o lumină de fundal LED și poate afișa două rânduri cu până la 16 caractere pe fiecare rând. Puteți vedea dreptunghiurile pentru fiecare caracter pe afișaj și pixelii care alcătuiesc fiecare caracter. Afișajul este doar alb pe albastru și este destinat să afișeze text.

În această lecție, vom rula exemplul de program Arduino pentru biblioteca LCD, dar în lecția următoare, vom face ca afișajul nostru să arate temperatura, folosind senzori.

### Componentă necesară:

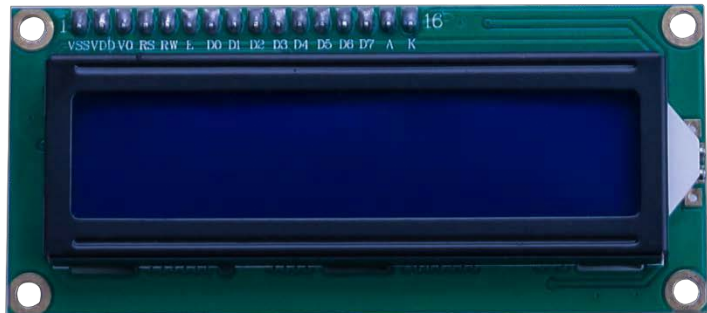
(1) x Elegoo Uno R3

(1) x modul LCD1602

(1) x potențiomtru (10k)

(1) x 830 de puncte de legătură Breadboard

(16) fire x MM (fire de legătură de tip masculin la masculin)



### Introducere componente

#### LCD1602

Introducere în pinii LCD1602: **VSS:**Un

știft care se conectează la pământ

**VDD:**Un pin care se conectează la o sursă de alimentare

de +5V **VO:**Un pin care reglează contrastul LCD1602

**RS:**Un pin de selectare a registrului care controlează locul în care scrieți datele din memoria LCD.

Puteți selecta fie registrul de date, care deține ceea ce apare pe ecran, fie un registru de instrucțiuni, care este locul în care controlerul LCD-ului caută instrucțiuni despre ce să facă în continuare.

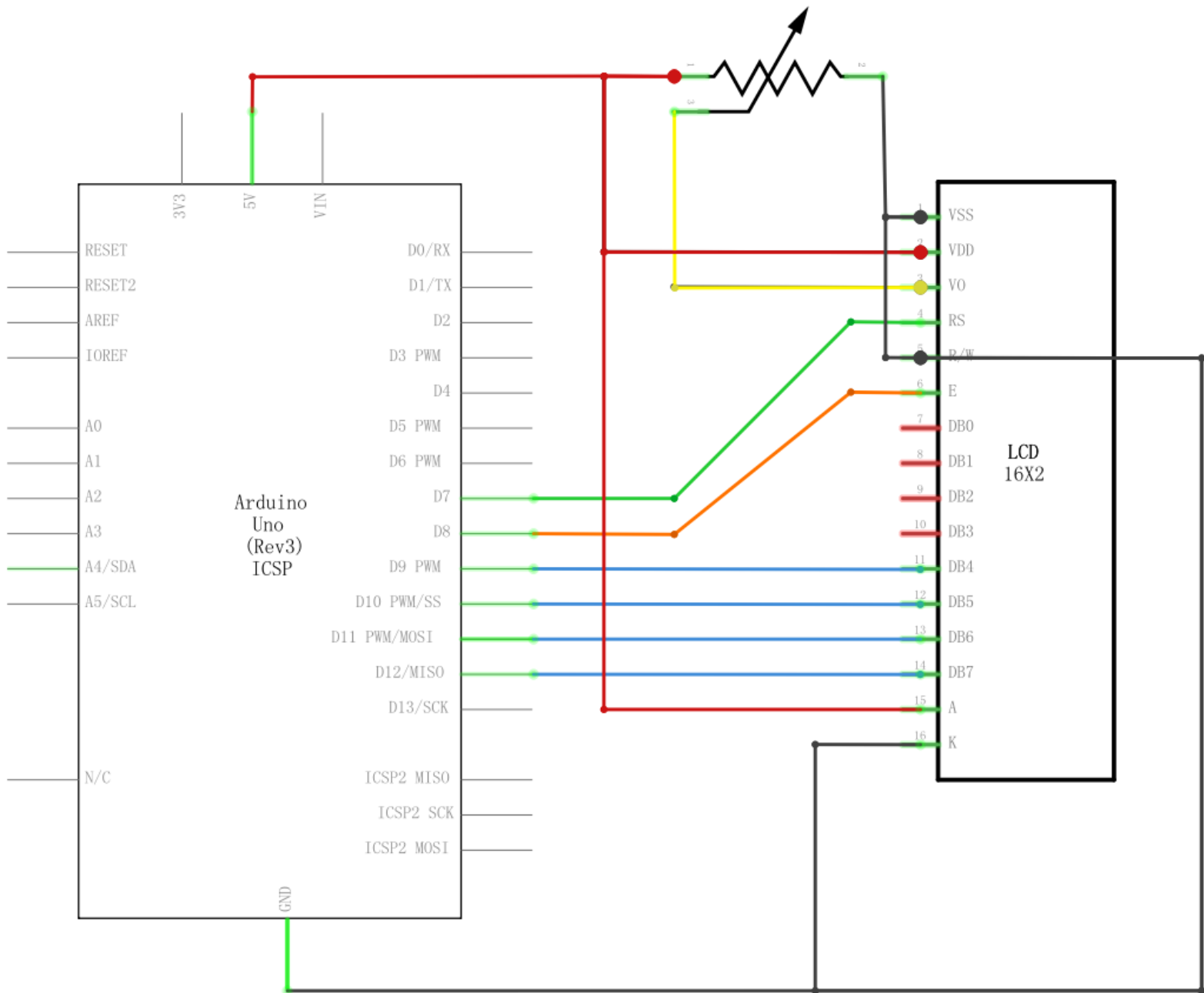
**R/W:**Un pin de citire/scriere care selectează modul de citire sau modul de scriere

**E:**Un pin de activare care, atunci când este furnizat cu energie de nivel scăzut, face ca modulul LDC să execute instrucțiuni relevante.

**D0-D7:** Pinuri care citesc și scriu date **A și K:**Pini

care controlează iluminarea LED-urilor

## Conexiune Schematic





Ecranul LCD are nevoie de șase pini Arduino, toate setate să fie ieșiri digitale. De asemenea, are nevoie de conexiuni de 5V și GND.

Există o serie de conexiuni de făcut. Alinierea afișajului cu partea superioară a plăcii de test ajută la identificarea pinii acestuia fără a număra prea mult, mai ales dacă placa de laborator are rândurile sale numerotate cu rândul 1 ca rândul superior al plăcii. Nu uitați, plumbul galben lung care leagă cursorul oală de pinul 3 al afișajului. „Oala” este folosită pentru a controla contrastul afișajului.

Este posibil să descoperiți că afișajul dvs. este furnizat fără știfturi atașate la el. Dacă da, urmați instrucțiunile din secțiunea următoare.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 14 Afișaj LCD](#) și faceți clic pe [ÎNCĂRCARE](#) pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca [< LiquidCrystal >](#) sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați [Lecția 1](#).

Încărcați codul pe placa Arduino și ar trebui să vedeți afișat mesajul „hello, world”, urmat de un număr care numără până la zero.

Primul lucru de notat în schiță este linia:

```
# include <LiquidCrystal.h>
```

Acest lucru îi spune lui Arduino că dorim să folosim biblioteca de cristale lichide.

În continuare avem linia pe care a trebuit să o modificăm. Aceasta definește ce pini ai Arduino vor fi conectați la ce pini ai afișajului.

```
LCD cristal lichid (7, 8, 9, 10, 11, 12;
```

După ce ați încărcat acest cod, asigurați-vă că lumina de fundal este aprinsă și reglați potențiometrul până când vedeți mesajul text

În funcția „setare”, avem două comenzi:

```
lcd.begin(16, 2;
```

```
lcd.print("Bună ziua, lume!");
```

Primul spune bibliotecii de cristale lichide câte coloane și rânduri are afișajul. A doua linie afișează mesajul pe care îl vedem pe prima linie a ecranului.

În funcția „bucă”, avem și două comenzi:

```
lcd.setCursor(0, 1;
```

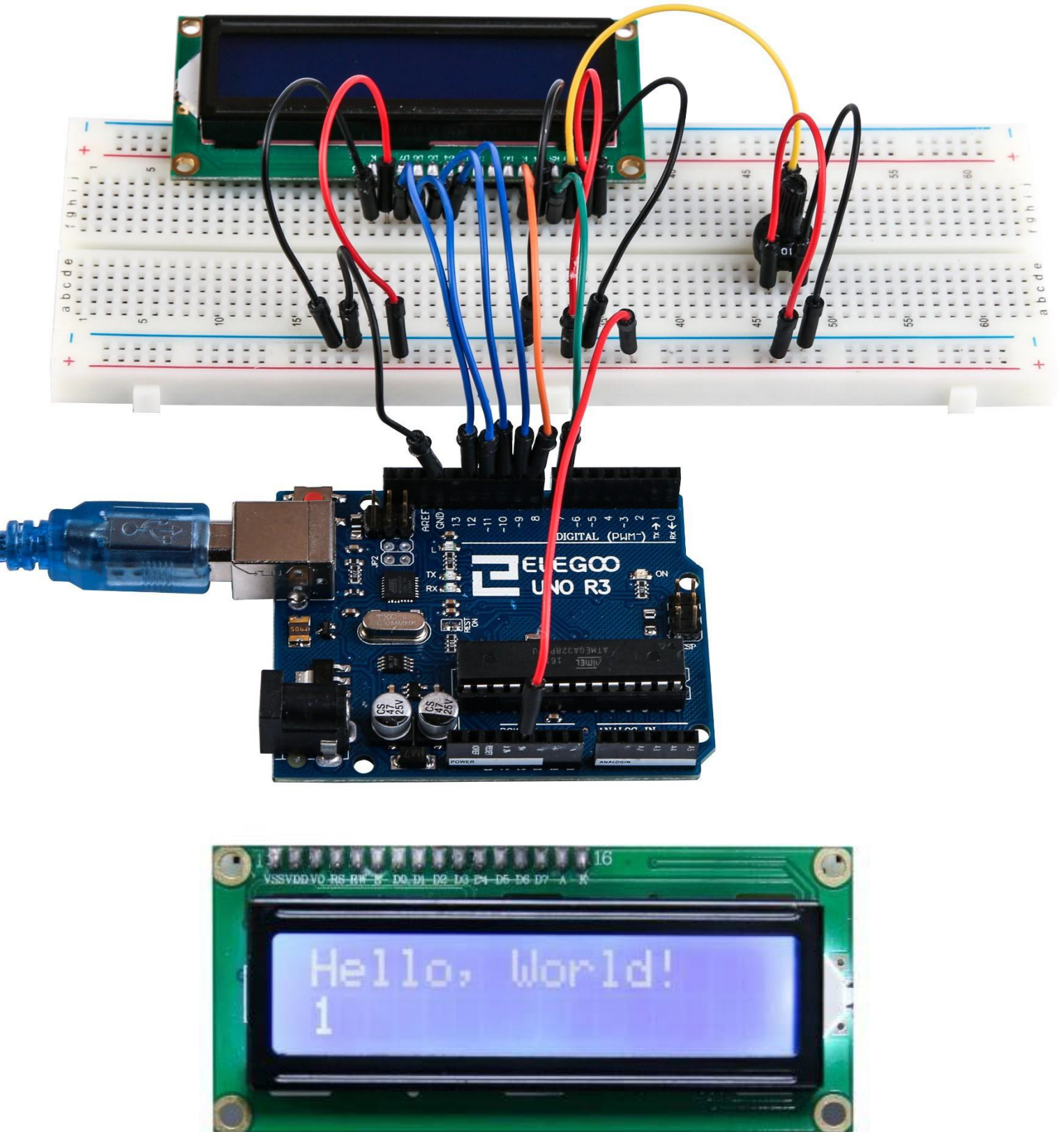
```
lcd.print(milis(/1000;
```

Primul setează poziția cursorului (unde va apărea următorul text) la coloana 0 și la rândul

1. Atât numerele de coloană, cât și de rând încep la 0 și nu la 1.

A doua linie afișează numărul de milisecunde de când Arduino a fost resetat.

**Imagine exemplu**



# Lecția 15 Termometru

## Prezentare generală

În această lecție, veți folosi un afișaj LCD pentru a afișa temperatura.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x Modul LCD1602

(1) x rezistor de 10k ohmi

(1) x termistor

(1) x potențiomtru

(1) x 830 de puncte de legătură Breadboard

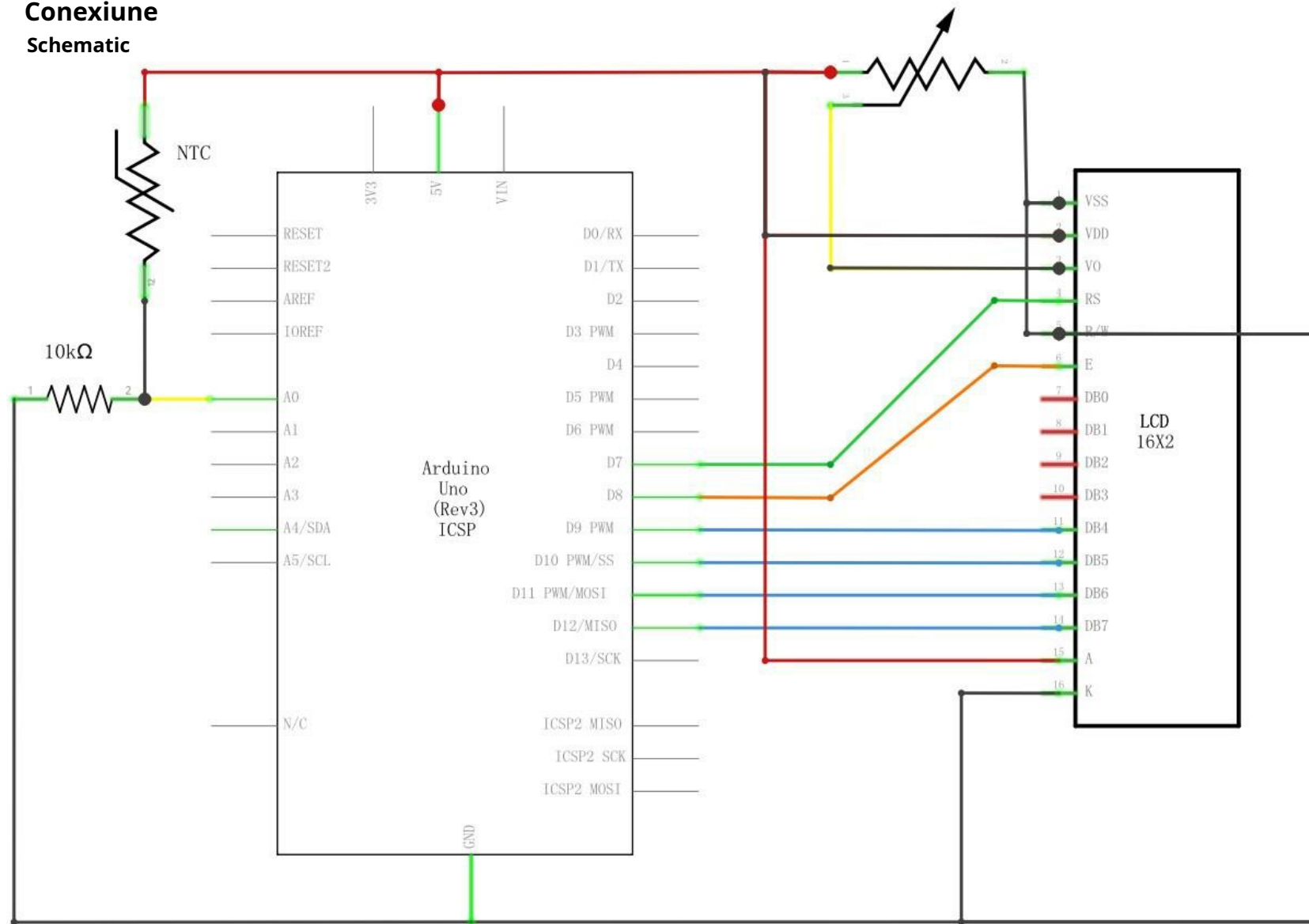
(18) fire x MM (fire de legătură de tip masculin la masculin)

## Introducere componente

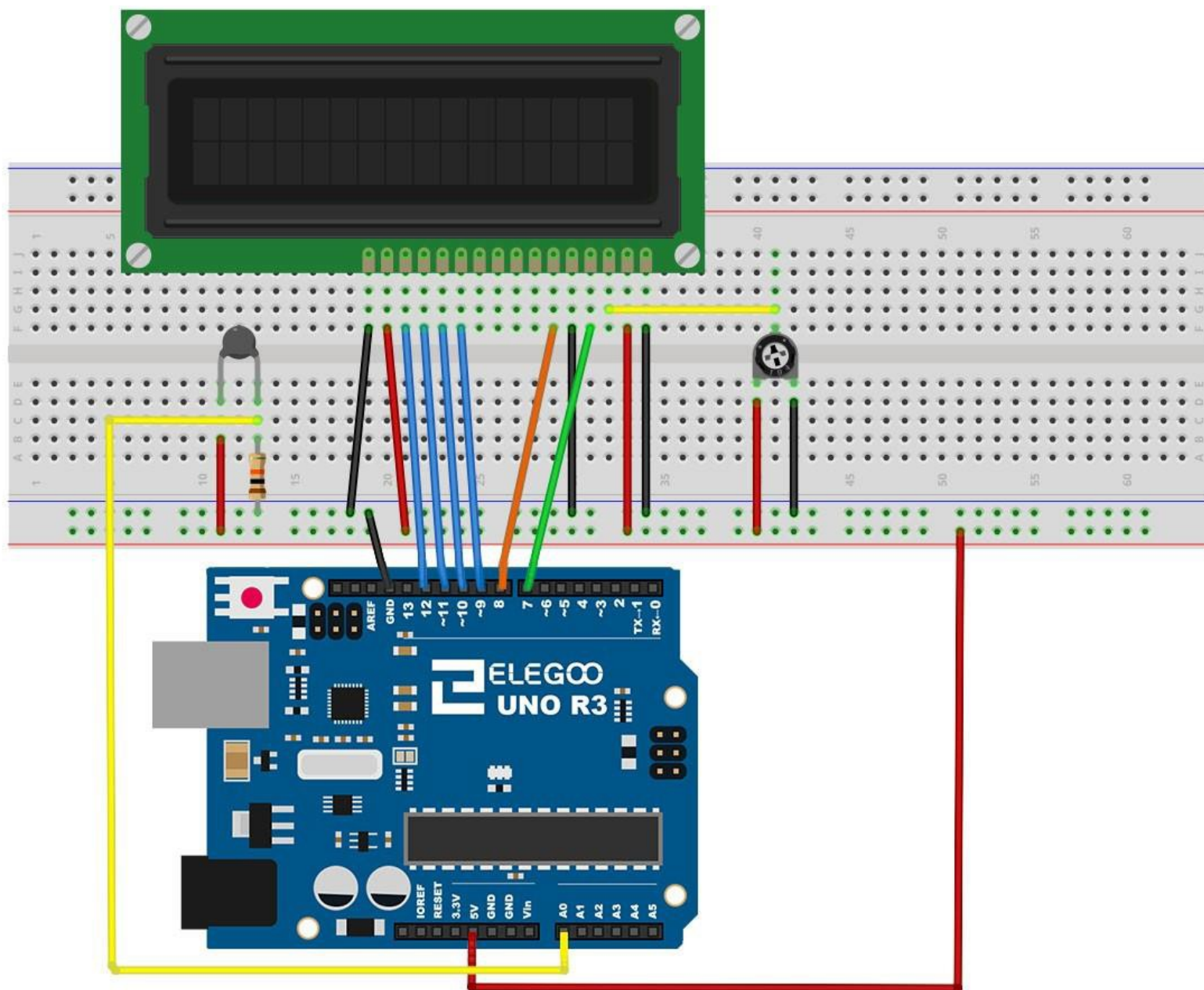
### Termistor

Un termistor este un rezistor termic - un rezistor care își modifică rezistența în funcție de temperatură. Din punct de vedere tehnic, toate rezistențele sunt termistori - rezistența lor se modifică ușor cu temperatura - dar schimbarea este de obicei foarte mică și greu de măsurat. Termistorii sunt fabricați astfel încât rezistența să se schimbe drastic cu temperatura, astfel încât să poată fi de 100 ohmi sau mai mult de schimbare pe grad! Există două tipuri de termistori, NTC (coeficient de temperatură negativ) și PTC (coeficient de temperatură pozitiv). În general, veți vedea senzori NTC folosiți pentru măsurarea temperaturii. PTC-urile sunt adesea folosite ca siguranțe resetabile - o creștere a temperaturii crește rezistența, ceea ce înseamnă că, pe măsură ce trece mai mult curent prin ele, se încălzesc și „sufocă” curentul, destul de util pentru protecția circuitelor!

## Conexiune Schematic



## Schema de cablare



Aspectul plăcii de laborator se bazează pe aspectul din lecția 22, așa că va simplifica lucrurile dacă mai aveți acest lucru pe placa de laborator.  
Există câteva fire jumper lângă oală care au fost ușor mutate pe acest aspect.

Rezistorul de 10 kΩ și termistorul sunt toate adăugări noi la placă.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 15 Termometru](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < Liquid Crystal > sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați [Lecția 1](#).

Schița pentru aceasta se bazează pe cea a [lesso14](#). Încărcați-l pe Arduino și ar trebui să descoperiți că încălzirea senzorului de temperatură punând degetul pe acesta va crește citirea temperaturii.

Mi se pare util să pun o linie de comentariu deasupra comenzii

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

Acest lucru ușurează lucrurile dacă decideți să schimbați ce pini folosiți.

În funcția „bucă” se întâmplă acum două lucruri interesante. În primul rând, trebuie să convertim analogul de la senzorul de temperatură într-o temperatură reală, iar în al doilea rând trebuie să aflăm cum să le afișăm.

În primul rând, să ne uităm la calcularea temperaturii.

```
int tempReading = analogRead(tempPin);  
dublu 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;
```

Afișarea citirilor în schimbare pe un afișaj LCD poate fi dificilă. Problema principală este că citirea poate să nu fie întotdeauna același număr de cifre. Deci, dacă temperatura s-a schimbat de la 101.50 la 99.00 atunci cifra suplimentară din citirea veche riscă să fie lăsată pe afișaj.

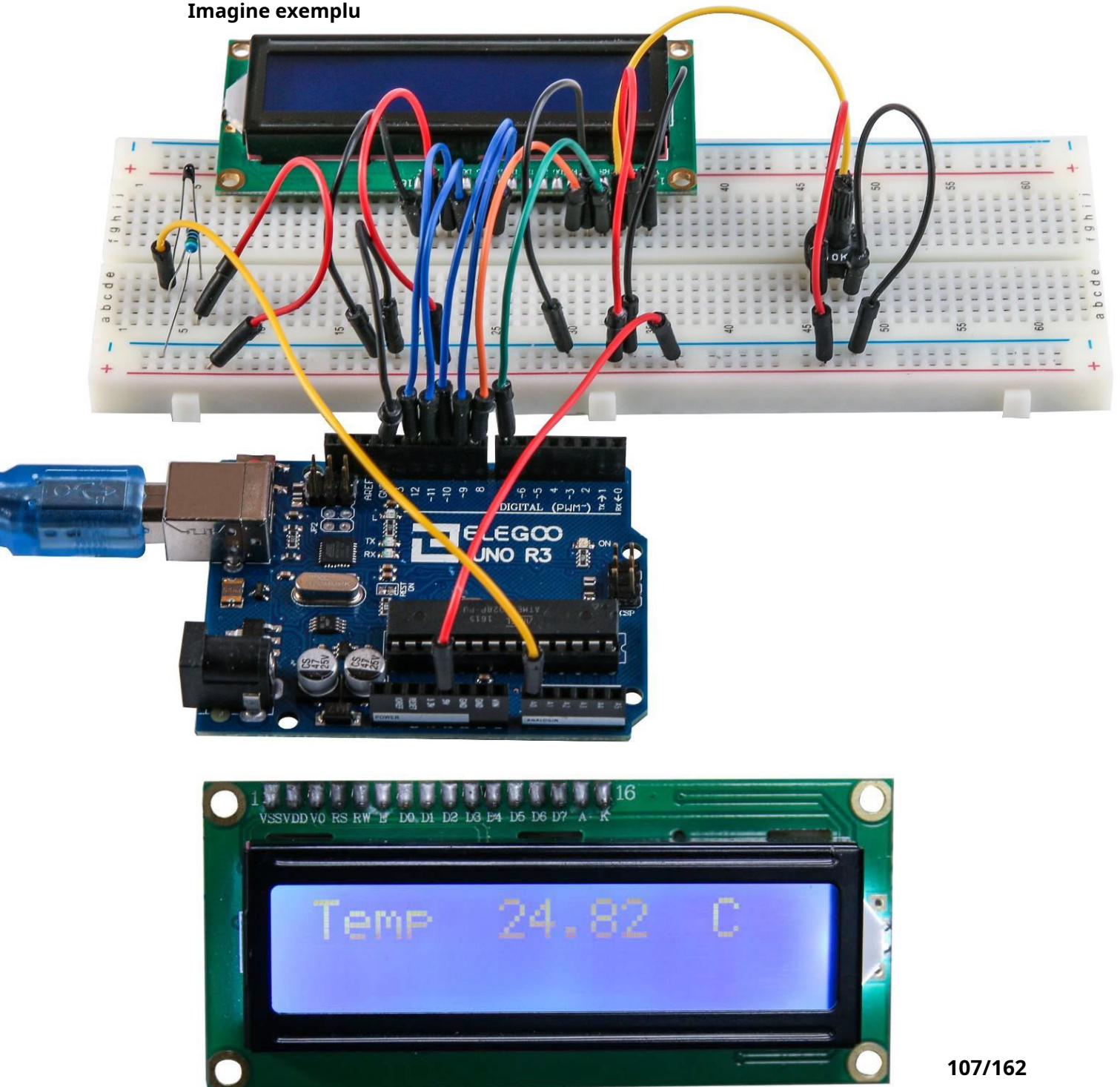
Pentru a evita acest lucru, scrieți întreaga linie a LCD-ului de fiecare dată în jurul buclei.

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

Comentariul destul de ciudat servește pentru a vă aminti de cele 16 coloane ale afișajului. Apoi puteți imprima un șir de aceea lungime cu spații unde va merge citirea reală.

Pentru a completa spațiile libere, setați poziția cursorului unde ar trebui să apară citirea și apoi imprimați-o.

### Imagine exemplu



## Lecția 16 Opt LED-uri cu 74HC595

### Prezentare generală

În această lecție, veți învăța cum să utilizați opt LED-uri roșii mari cu un UNO fără a fi nevoie să renunțați la 8 pini de ieșire!

Deși ați putea conecta opt LED-uri fiecare cu un rezistor la un pin UNO, veți începe rapid să rămâneți fără pini pe UNO. Dacă nu aveți multe lucruri conectate la UNO. Este în regulă să faceți acest lucru - dar de multe ori vrem butoane, senzori, servomotoare etc. și înainte să vă dați seama nu mai aveți ace. Deci, în loc să faci asta, vei folosi un cip numit 74HC595 Serial to Parallel Converter. Acest cip are opt ieșiri (perfecte) și trei intrări pe care le folosiți pentru a introduce date în el pe rând.

Acest cip face un pic mai lent pentru a conduce LED-urile (puteți schimba LED-urile doar de aproximativ 500.000 de ori pe secundă în loc de 8.000.000 pe secundă), dar este încă foarte rapid, mult mai rapid decât poate detecta oamenii, așa că merită!

### Componentă necesară:

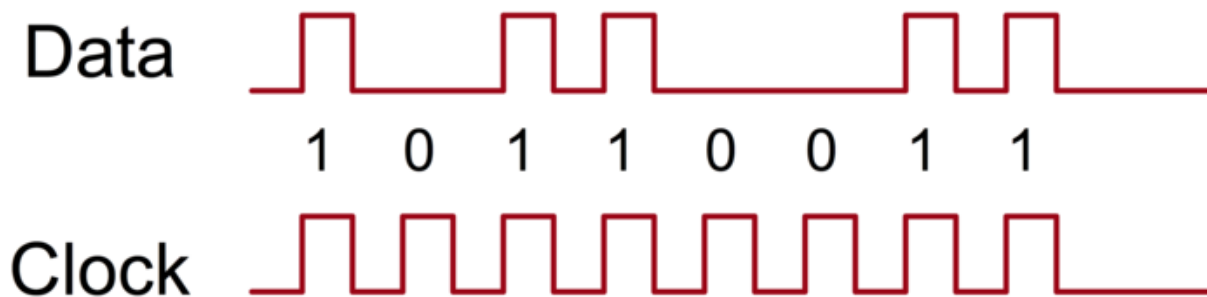
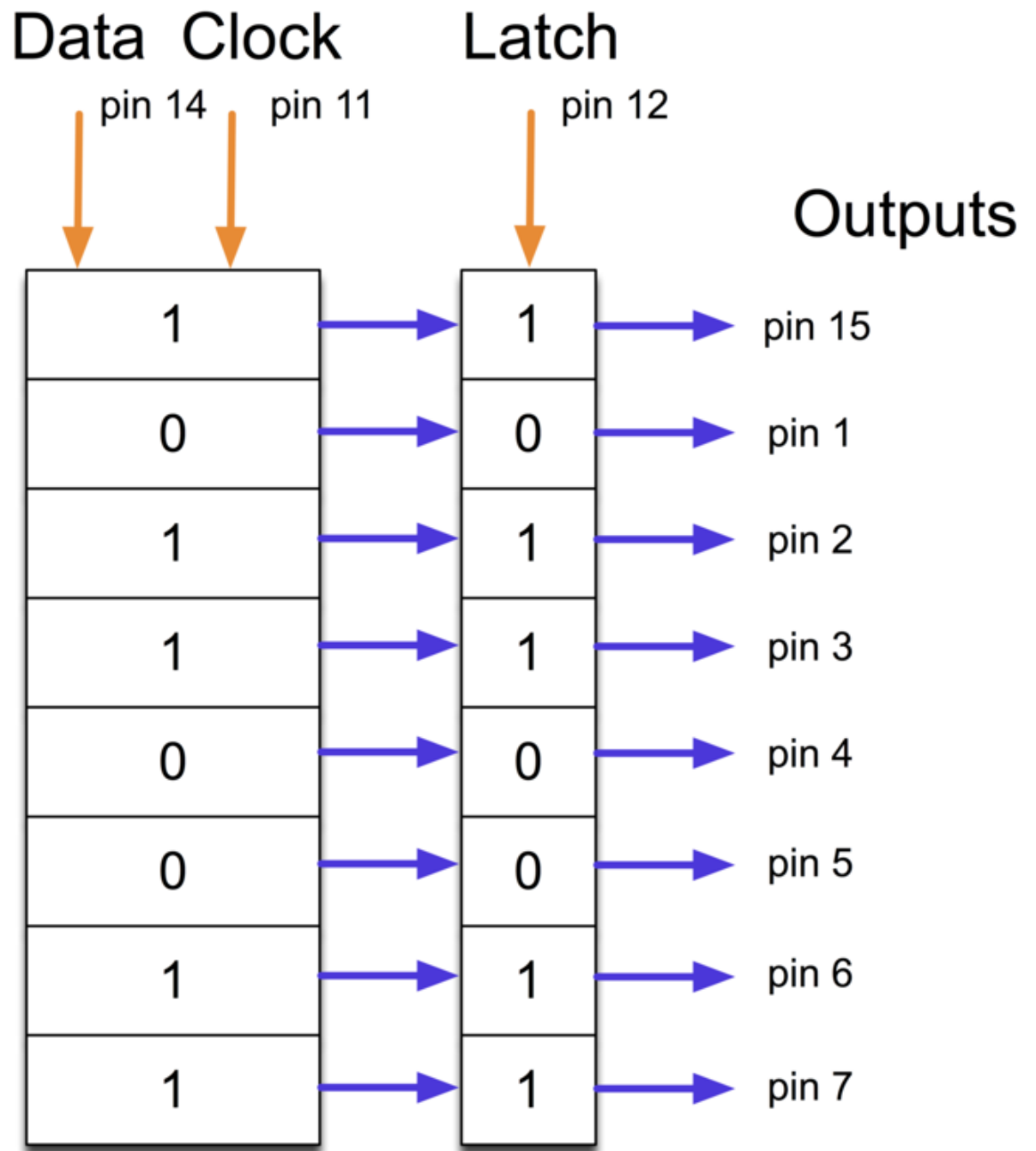
- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (8) x leduri
- (8) rezistențe x 220 ohmi
- (1) x 74hc595 IC
- (14) fire x MM (fire de legătură de tip masculin la masculin)



### Introducere componente

#### 74HC595 Registrul de schimb:

Registrul de deplasare este un tip de cip care deține ceea ce poate fi considerat ca fiind opt locații de memorie, fiecare dintre acestea putând fi fie 1, fie 0. Pentru a activa sau dezactiva fiecare dintre aceste valori, introducem datele utilizând „Pini „Date” și „Clock” ai cipului.

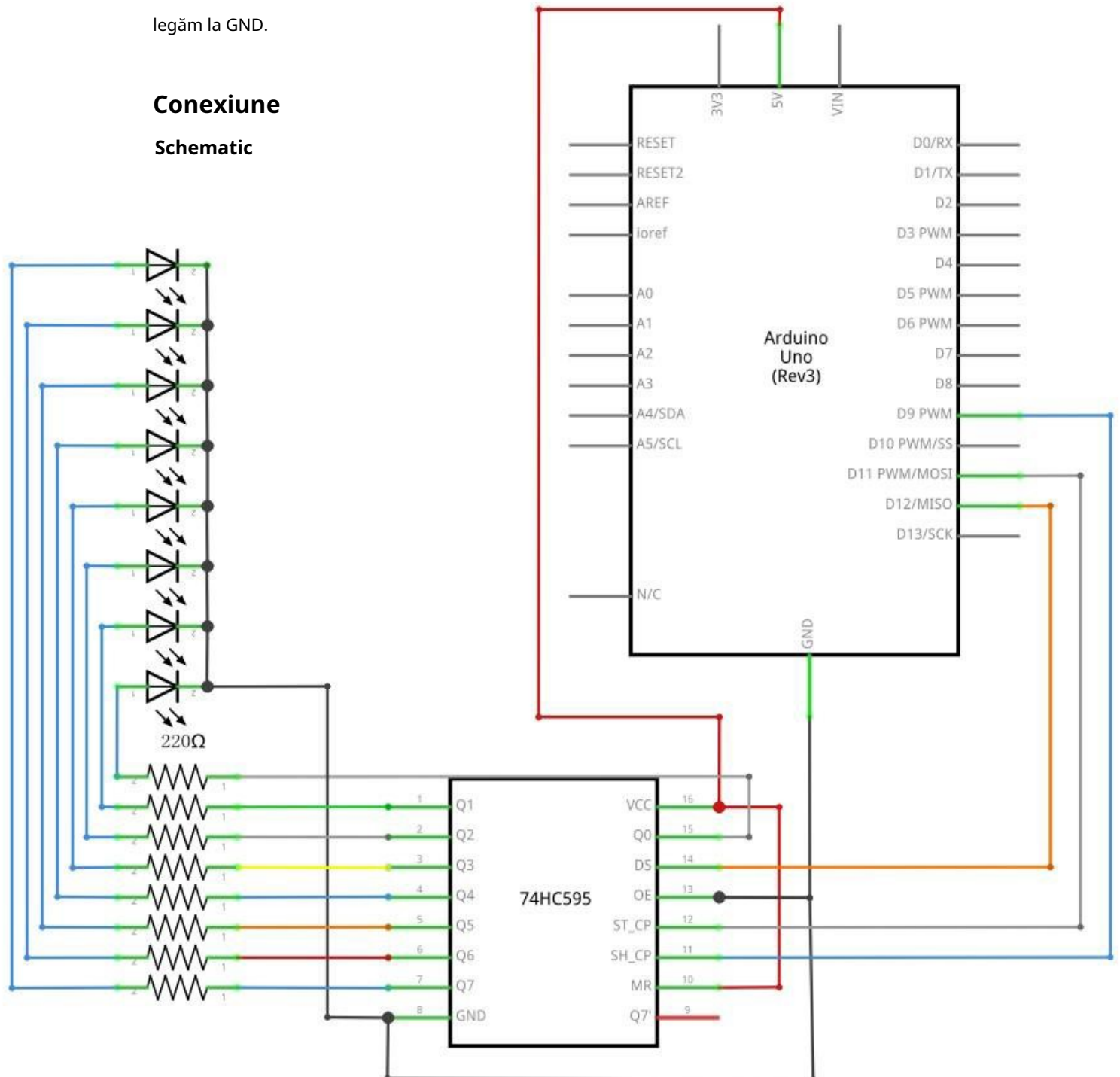


Pinul ceasului trebuie să primească opt impulsuri. La fiecare impuls, dacă pinul de date este ridicat, atunci un 1 este împins în registrul de deplasare; în caz contrar, un 0. Când toate cele opt impulsuri au fost primite, activarea pinului „Latch” copiază acele opt valori în registrul de blocare. Acest lucru este necesar; în caz contrar, LED-urile greșite ar pâlpâi pe măsură ce datele sunt încărcate în registrul de deplasare.

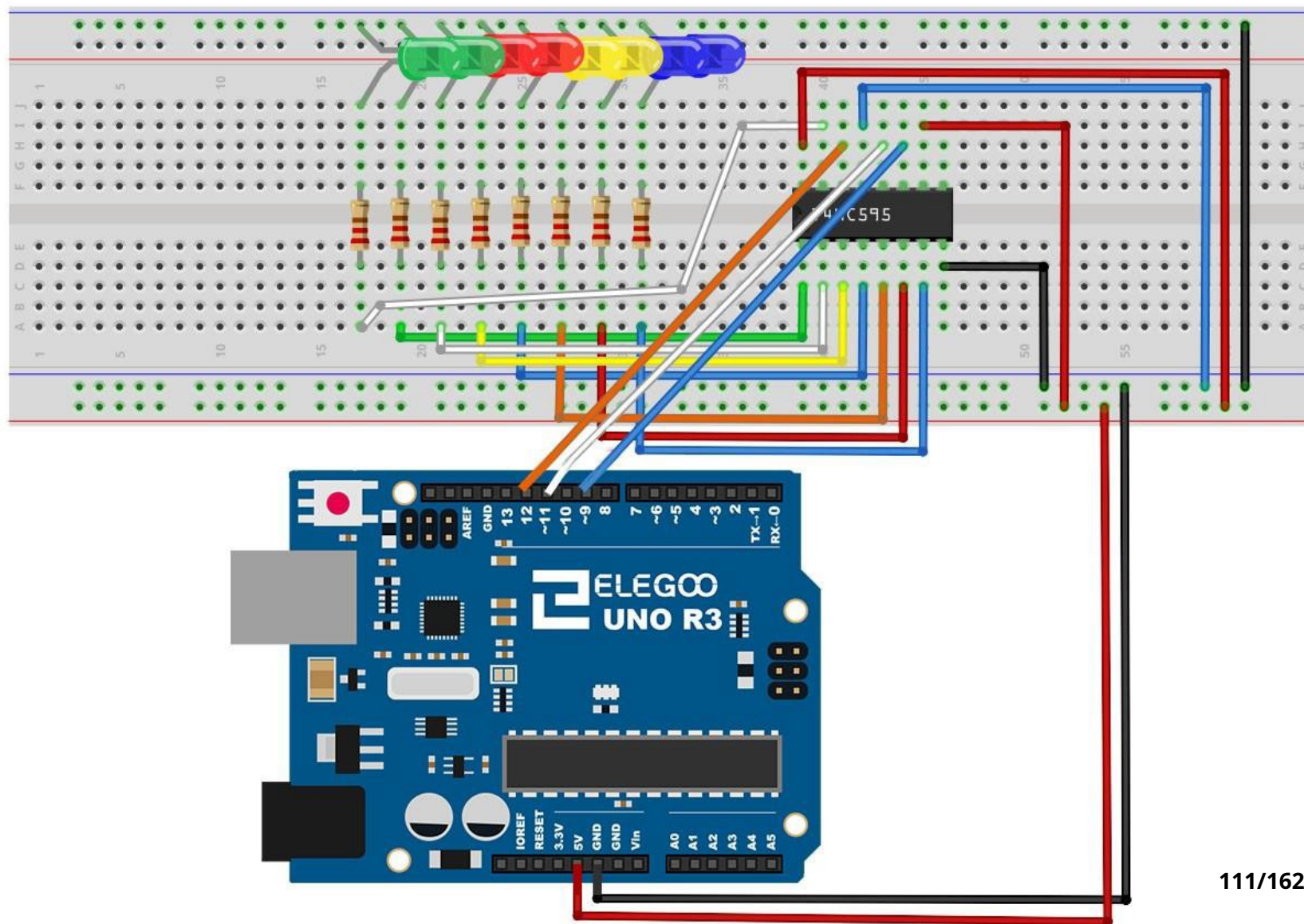
Cipul are, de asemenea, un pin de activare a ieșirii (OE), care este folosit pentru a activa sau dezactiva ieșirile dintr-o dată. Puteți atașa acest lucru la un pin UNO compatibil PWM și puteți utiliza „analogWrite” pentru a controla luminozitatea LED-urilor. Acest pin este activ la nivel scăzut, așa că îl legăm la GND.

## Conexiune

### Schematic



## Schema de cablare



Deoarece avem opt LED-uri și opt rezistențe de conectat, sunt de fapt destul de multe conexiuni de făcut.

Probabil cel mai ușor este să puneți primul cipul 74HC595, deoarece aproape orice altceva se conectează la el. Puneți-o astfel încât creștătura mică în formă de U să fie înspre partea de sus a plăcii. Pinul 1 al cipului este în stânga acestei creștături.

Digital 12 de la UNO merge la pinul #14 al registrului de deplasare

Digital 11 de la UNO merge la pinul #12 al registrului de deplasare

Digital 9 de la UNO merge la pinul #11 al registrului de deplasare

Toate, cu excepția uneia, ieșirile de la IC se află pe partea stângă a cipului. Prin urmare, pentru ușurința conexiunii, acolo sunt și LED-urile.

După cip, puneți rezistențele la locul lor. Trebuie să aveți grijă ca niciunul dintre cablurile rezistențelor să nu se atingă. Ar trebui să verificați acest lucru din nou înainte de a conecta alimentarea la UNO. Dacă vă este dificil să aranjați rezistențele fără ca cablurile lor să se atingă, atunci vă ajută să scurtați cablurile, astfel încât acestea să se afle mai aproape de suprafața plăcii.

Apoi, plasați LED-urile pe placa. LED-urile pozitive mai lungi trebuie să fie toate către cip, indiferent de partea pe care se află ele.

Atașați cablurile jumper așa cum se arată mai sus. Nu uitați-l pe cel care merge de la pinul 8 al IC-ului la coloana GND a plăcii.

Încărcați schița listată puțin mai târziu și încercați-o. Fiecare LED ar trebui să se aprindă pe rând până când toate LED-urile sunt aprinse, apoi toate se sting și ciclul se repetă.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 16 Opt LED cu

74HC595 și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Primul lucru pe care îl facem este să definim cei trei pini pe care îi vom folosi. Acestea sunt ieșirile digitale UNO care vor fi conectate la zăvorul, ceasul și pinii de date ai 74HC595.

```
int latchPin = 11;
```

```
int clockPin = 9;
```

```
int dataPin = 12;
```

În continuare, este definită o variabilă numită „led-uri”. Acesta va fi folosit pentru a menține modelul a cărui LED-uri sunt pornite sau oprite în prezent. Datele de tip „octet” reprezintă numere care folosesc opt biți. Fiecare bit poate fi activat sau oprit, deci este perfect pentru ținând evidența

care dintre cele opt LED-uri ale noastre sunt aprinse sau stinse.

```
led-uri octet = 0;
```

Funcția „setare” setează doar cei trei pini pe care îi folosim să fie ieșiri digitale. **void**

```
setup()
```

```
{
```

```
    pinMode(latchPin, OUTPUT);
```

```
    pinMode(dataPin, OUTPUT);
```

```
    pinMode(clockPin, OUTPUT);
```

```
}
```

Funcția „bucă” stinge inițial toate LED-urile, dând variabilei „led-uri” valoarea 0. Apoi apelează „updateShiftRegister” care va trimite modelul „led-urilor” la registrul de schimbare, astfel încât toate LED-urile să se stingă. Ne vom ocupa de modul în care funcționează „updateShiftRegister” mai târziu.

Funcția buclă se oprește pentru o jumătate de secundă și apoi începe să numere de la 0 la 7 folosind bucla „for” și variabila „i”. De fiecare dată, folosește funcția Arduino „bitSet” pentru a seta bitul care controlează acel LED în variabila „led-uri”. Apoi apelează, de asemenea, „updateShiftRegister”, astfel încât ledurile să se actualizeze pentru a reflecta ceea ce este în variabila „leds”. Există apoi o întârziere de jumătate de secundă înainte ca „i” să fie incrementat și următorul LED să se aprindă. **bucă goală ()**

```
{
```

```
    leduri = 0;
```

```
    updateShiftRegister();
```

```
    întârziere (500);
```

```
    pentru (int i = 0; i < 8; i++) {
```

```
        bitSet(led-uri, i);
```

```
        updateShiftRegister();
```

```
        întârziere (500);
```

```
    }
```

```
}
```

Funcția „updateShiftRegister”, în primul rând setează latchPin-ul la nivel scăzut, apoi apelează funcția UNO „shiftOut” înainte de a pune din nou „latchPin-ul” ridicat. Acest lucru necesită patru parametri, primii doi sunt pinii de utilizat pentru Date și, respectiv, Ceas.

Al treilea parametru specifică la ce capăt al datelor doriți să începeți. Vom începe cu partea cea mai potrivită, care se numește „Cel mai puțin semnificativ”.

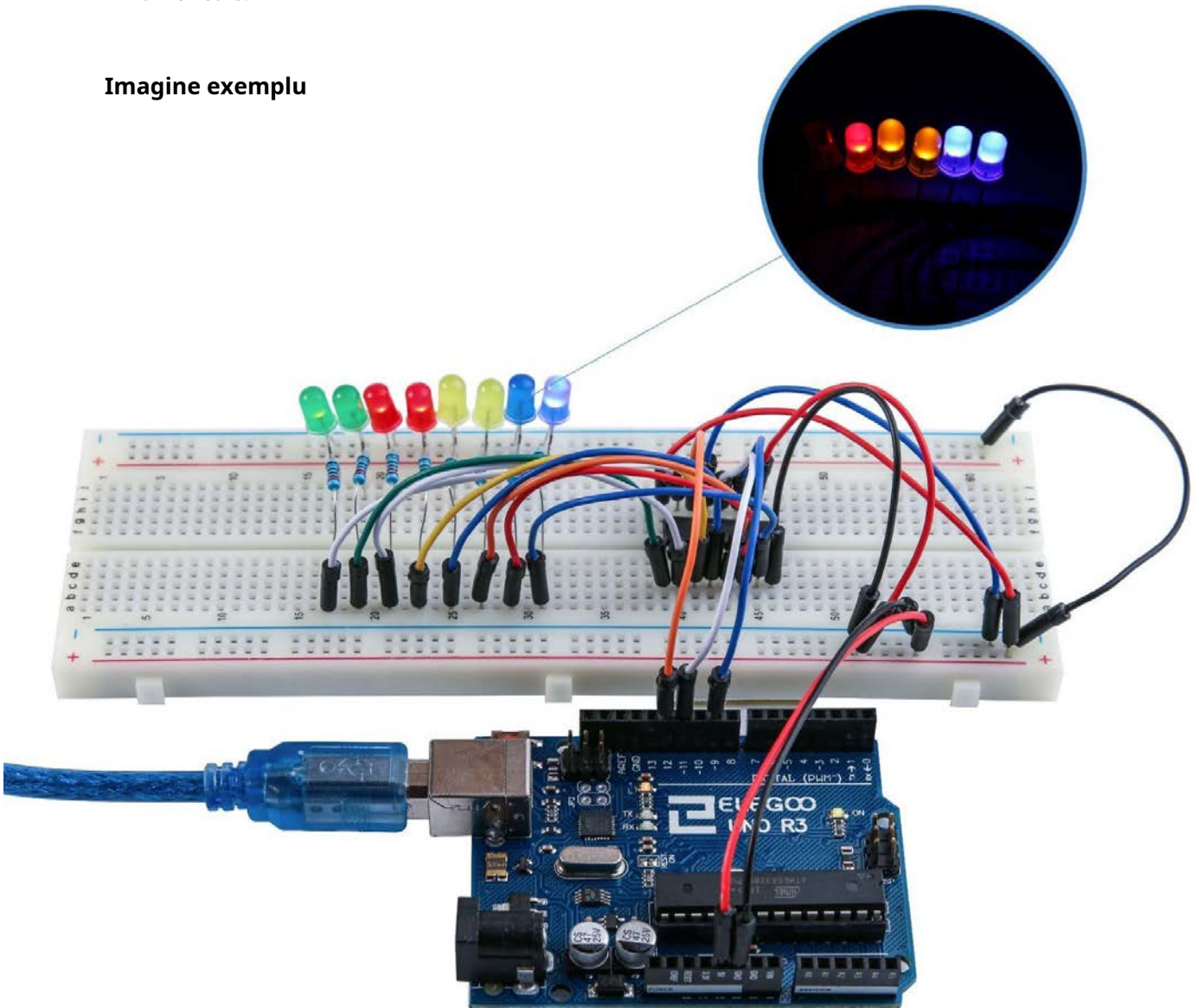
Bit' (LSB).

Ultimul parametru este datele reale care trebuie mutate în registrul de deplasare, care în acest caz este „led-uri”.

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

Dacă doriți să opriți unul dintre LED-uri mai degrabă decât să porniți, ați apela o funcție Arduino similară (bitClear) cu variabila „leds”. Acest lucru va seta acel bit de „led-uri” la 0 și atunci ar trebui doar să îl urmați cu un apel la „updateShiftRegister” pentru a actualiza LED-urile reale.

### Imagine exemplu



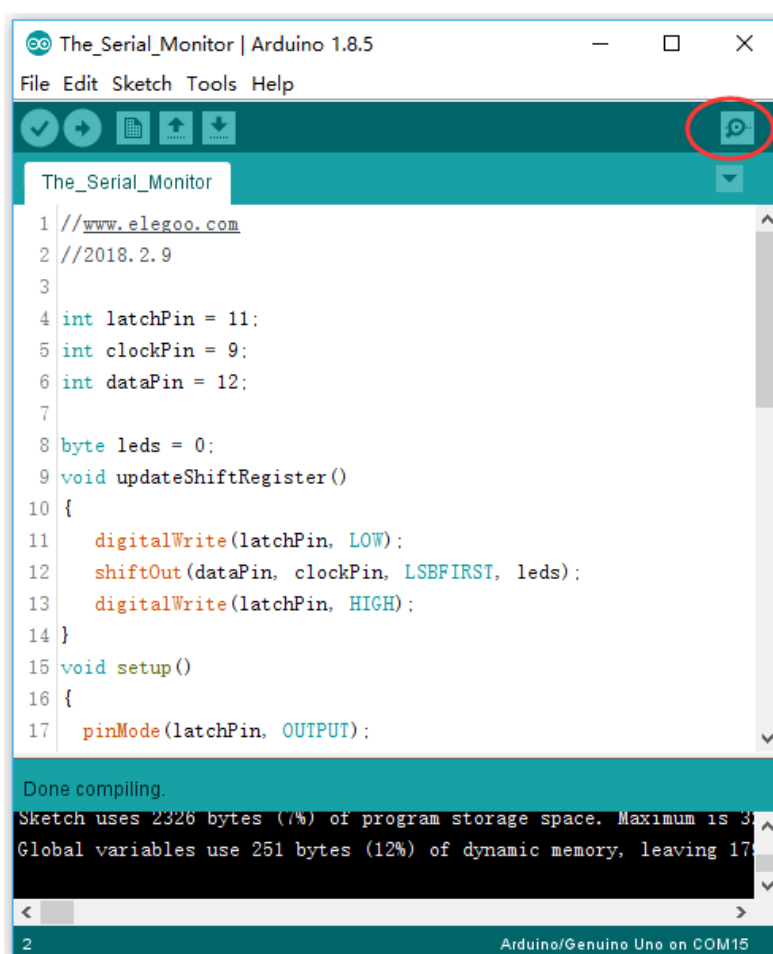
## Lecția 17 Monitorul serial

### Prezentare generală

În această lecție, veți construi pe lecția 16, adăugând posibilitatea de a controla LED-urile de pe computer folosind monitorul serial Arduino. Monitorul serial este „legatul” dintre computer și UNO. Vă permite să trimiteți și să primiți mesaje text, la îndemână pentru depanare și, de asemenea, pentru controlul UNO de la o tastatură! De exemplu, veți putea trimite comenzi de pe computer pentru a porni LED-urile. În această lecție, veți folosi exact aceleași părți și un aspect similar pentru placa de laborator ca în Lecția 16. Deci, dacă nu ați făcut-o deja, urmați Lecția 16 acum.

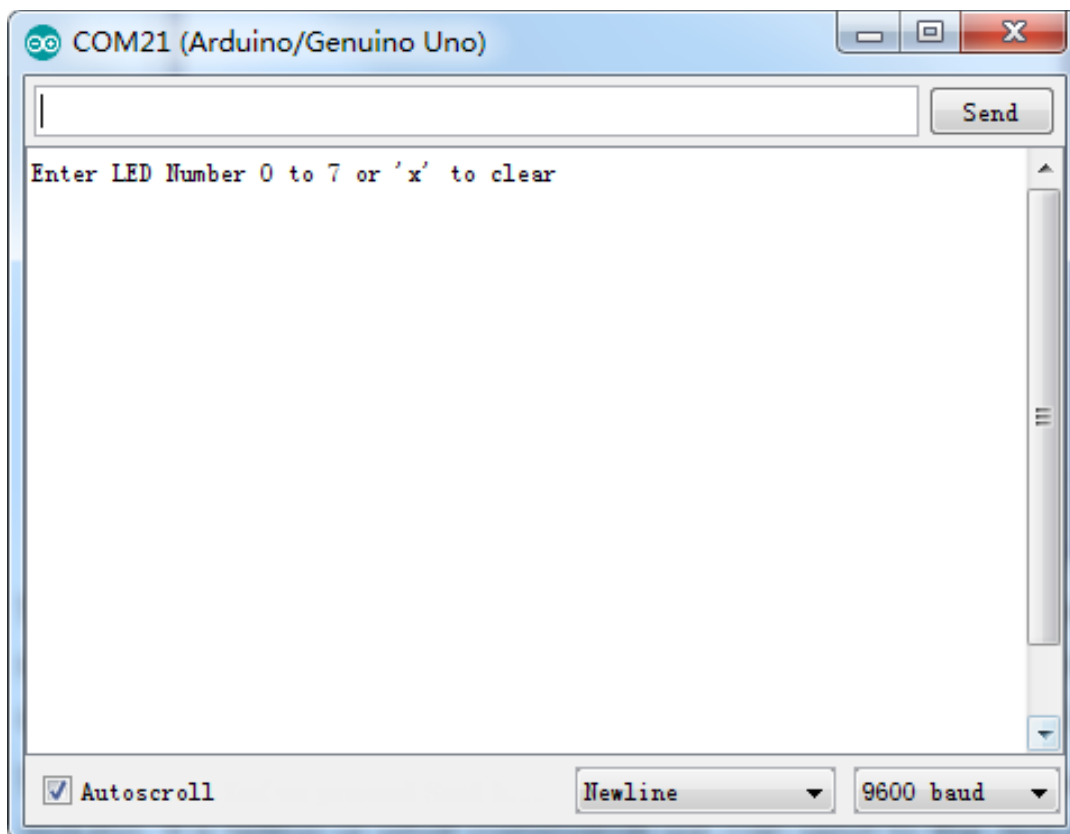
### Pașii făcuți

După ce ați încărcat această schiță pe UNO, faceți clic pe butonul din dreapta din bara de instrumente din Arduino IDE. Butonul este încercuit mai jos.



Se va deschide următoarea fereastră.

Faceți clic pe butonul Monitor serial pentru a porni monitorul serial. Elementele de bază despre monitorul serial sunt introduse în detalii în Lecția 1.

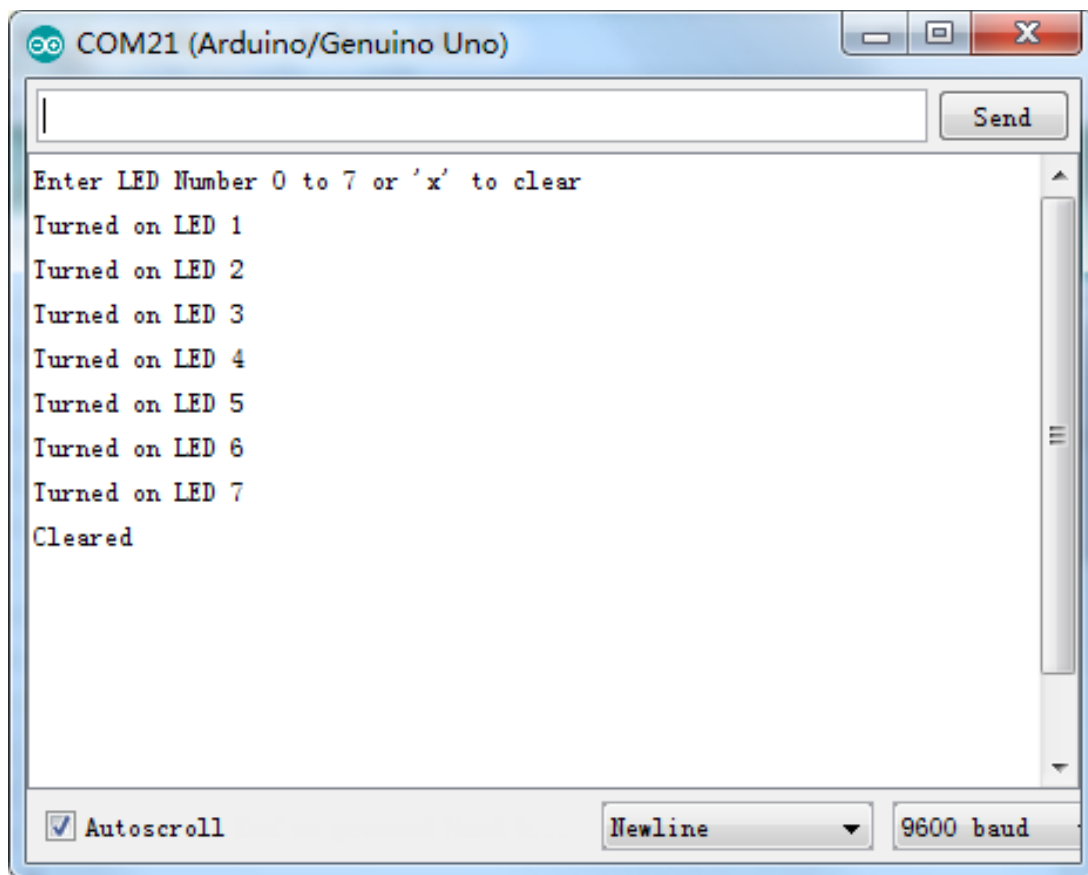


Această fereastră se numește Serial Monitor și face parte din software-ul Arduino IDE. Sarcina sa este să vă permită atât să trimiteți mesaje de pe computer la o placă UNO (prin USB), cât și să primiți mesaje de la UNO.

Mesajul „Introduceți numărul LED de la 0 la 7 sau „x” pentru a șterge” a fost trimis de Arduino. Ne spune ce comenzi putem trimite la Arduino: fie trimiteți „x” (pentru a stinge toate LED-urile), fie numărul LED-ului pe care doriți să-l porniți (unde 0 este LED-ul de jos, 1 este următorul unul în sus, până la 7 pentru LED-ul superior).

Încercați să tastați următoarele comenzi în zona de sus a monitorului serial, care este la nivelul butonului „Trimite”. Apăsați „Trimite”, după ce ați introdus fiecare dintre aceste caractere: x 0 3 5

Tastarea x nu va avea efect dacă LED-urile sunt deja stinse, dar pe măsură ce introduceți fiecare număr, LED-ul corespunzător ar trebui să se aprindă și veți primi un mesaj de confirmare de la placa UNO. Monitorul serial va apărea așa cum se arată mai jos.



Tastați x din nou și apăsați „Trimite” pentru a stinge toate LED-urile.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 17 Monitorul serial](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

După cum v-ați putea aștepta, schița se bazează pe schița folosită în [Lecția 16](#). Așadar, vom acoperi doar noile biți aici. Veți găsi util să faceți referire la schița completă din IDE-ul dvs. Arduino.

În funcția „setare”, există trei linii noi la sfârșit: **void**

**setup()**

**{**

**pinMode(latchPin, OUTPUT);**

**pinMode(dataPin, OUTPUT);**

**pinMode(clockPin, OUTPUT);**

**updateShiftRegister();**

**Serial.begin(9600);**

```

    în timp ce (! Serial); // Așteptați până când serialul este gata - Leonardo

    Serial.println(„Introduceți numărul LED de la 0 la 7 sau „x” pentru a șterge”);
}

```

În primul rând, avem comanda „Serial.begin(9600)”. Aceasta pornește comunicarea în serie, astfel încât UNO să poată trimite comenzi prin conexiunea USB. Valoarea 9600 se numește „rată de transmisie” a conexiunii. Acesta este cât de repede vor fi trimise datele. Puteți schimba aceasta la o valoare mai mare, dar va trebui să schimbați și monitorul serial Arduino la aceeași valoare. Vom discuta despre asta mai târziu; deocamdată, lasă-l la 9600.

Linia care începe cu „în timp ce” asigură că există ceva la celălalt capăt al conexiunii USB cu care Arduino poate vorbi înainte de a începe să trimită mesaje. În caz contrar, mesajul ar putea fi trimis, dar să nu fie afișat. Această linie este de fapt necesară numai dacă utilizați un Arduino Leonardo, deoarece Arduino UNO resetează automat placa Arduino atunci când deschideți monitorul serial, în timp ce acest lucru nu se întâmplă cu Leonardo.

Ultima dintre noile linii din „configurare” trimite mesajul pe care îl vedem în partea de sus a monitorului serial.

Funcția „buclă” este locul în care se întâmplă toată acțiunea: **buclă**

```

goală ()
{
    dacă (Serial.available()) {

        char ch = Serial.read(); dacă
        (ch >= '0' && ch <= '7') {

            int led = ch - '0';
            bitSet(led-uri, led);
            updateShiftRegister();
            Serial.print("LED pornit");
            Serial.println(led);
        }

        dacă (ch == 'x')
        {
            leduri = 0;
            updateShiftRegister();

```

```

        Serial.println("Șters");
    }
}
}

```

Tot ceea ce se întâmplă în interiorul buclei este conținut într-o instrucțiune „if”. Deci, dacă apelul la funcția Arduino încorporată „Serial.available()” este „adevărat”, atunci nu se va întâmpla nimic altceva.

Serial.available() va returna „adevărat” dacă datele au fost trimise la UNO și sunt acolo gata pentru a fi procesate. Mesajele primite sunt păstrate în ceea ce se numește un buffer și

Serial.available() returnează adevărat dacă acel buffer este Not gol.

Dacă a fost primit un mesaj, atunci acesta este pe următoarea linie de cod:

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

Aceasta citește următorul caracter din buffer și îl elimină din buffer. De asemenea, îl atribuie variabilei „ch”. Variabila „ch” este de tip „char”, care înseamnă „caracter” și, așa cum sugerează și numele, deține un singur caracter.

Dacă ați urmat instrucțiunile din promptul din partea de sus a monitorului serial, atunci acest caracter va fi fie un număr cu o singură cifră între 0 și 7, fie litera „x”.

Declarația „dacă” de pe linia următoare verifică dacă este o singură cifră, văzând dacă „ch” este mai mare sau egal cu caracterul „0” și mai mic sau egal cu caracterul „7”. Pare puțin ciudat comparând personaje în acest fel, dar este perfect acceptabil.

Fiecare caracter este reprezentat de un număr unic, numit valoarea sa ASCII. Aceasta înseamnă că atunci când comparăm caractere folosind <= și >=, de fapt, valorile ASCII au fost comparate.

Dacă testul trece, atunci ajungem la următoarea linie:

```
int led = ch - '0';
```

Acum facem aritmetică pe personaje! Scădem cifra „0” din orice cifră a fost introdusă. Deci, dacă ați tastat „0”, atunci „0” – „0” va fi egal cu 0. Dacă ați introdus „7”, atunci „7” – „0” va fi egal cu numărul 7, deoarece de fapt valorile ASCII sunt folosite în scădere.

Din moment ce știm numărul LED-ului pe care vrem să-l pornim, trebuie doar să setăm acel bit în variabila „led-uri” și să actualizăm registrul de deplasare.

```

bitSet(led-uri, led);
updateShiftRegister();

```

Următoarele două rânduri scriu înapoi un mesaj de confirmare pe monitorul serial. `Serial.print("LED pornit"); Serial.println(led);`

Prima linie folosește `Serial.print` mai degrabă decât `Serial.println`. Diferența dintre cele două este că `Serial.print` nu începe o nouă linie după imprimarea a ceea ce este în parametrul său. Folosim acest lucru în prima linie, deoarece tipărim mesajul în două părți. În primul rând, bitul general: „LED pornit” și apoi numărul LED-ului.

Numărul LED-ului este păstrat într-o variabilă „int” în loc să fie un șir de text.

`Serial.print` poate lua fie un șir de text cuprins între ghilimele duble, fie un „int” sau, de altfel, aproape orice tip de variabilă.

După instrucțiunea „dacă” care se ocupă de cazul, când a fost tratată o singură cifră, există o a doua instrucțiune „dacă” care verifică dacă „ch” este litera „x”.

```
dacă (ch == 'x')
{
    leduri = 0; updateShiftRegister();
    Serial.println("Șters");
}
```

Dacă este, atunci șterge toate LED-urile și trimite un mesaj de confirmare.

## Lecția 18 Fotocelulă

### Prezentare generală

În această lecție, veți învăța cum să măsurați intensitatea luminii folosind o intrare analogică. Veți construi pe baza lecției 16 și veți folosi nivelul de lumină pentru a controla numărul de LED-uri care trebuie aprinse.

Fotocelula se află în partea inferioară a tablei, acolo unde se afla oala deasupra.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (8) x leduri
- (8) rezistențe x 220 ohmi
- (1) x rezistență de 1k ohm
- (1) x 74hc595 IC
- (1) x fotorezistor (fotocelula)
- (16) fire x MM (fire de legătură de tip masculin la masculin)



### Introducere componente

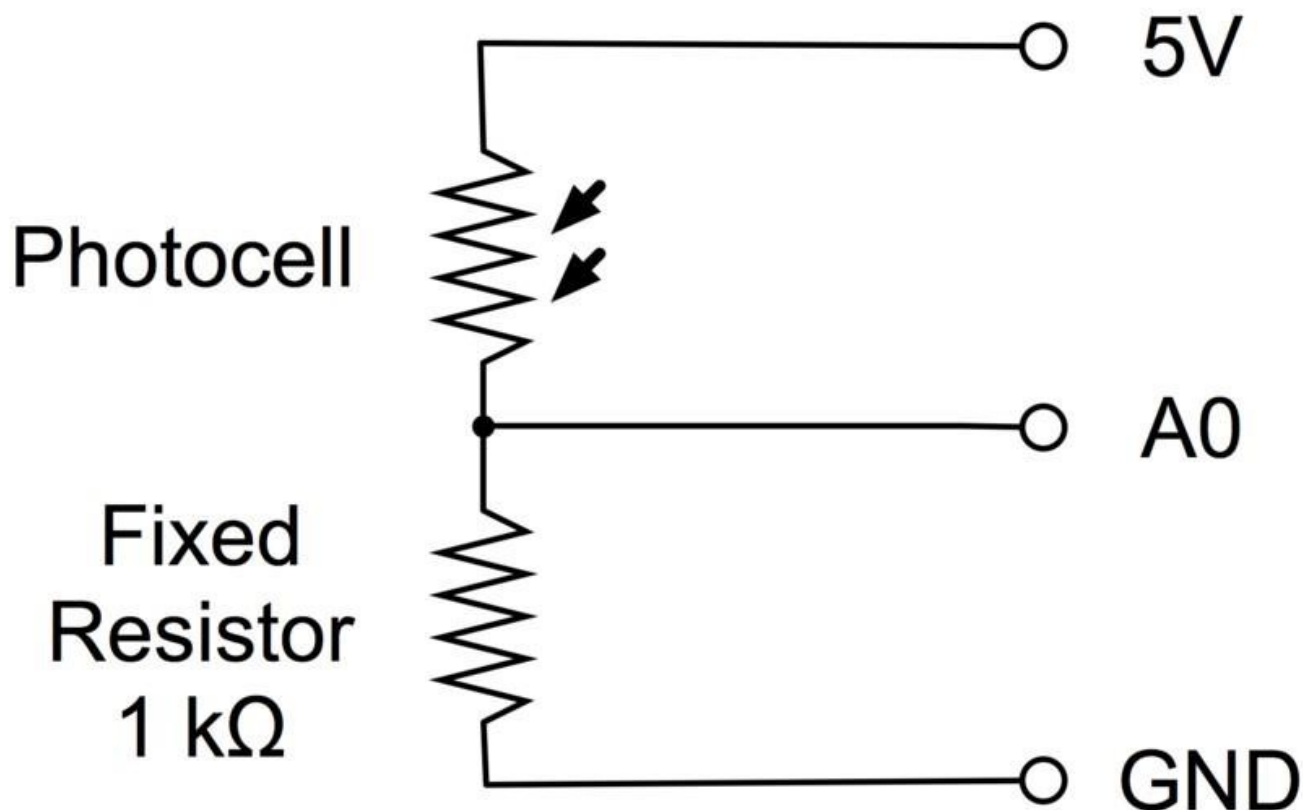
#### FOTOCELULA:

Fotocelula utilizată este de un tip numit rezistor dependent de lumină, numit uneori LDR. După cum sugerează și numele, aceste componente acționează exact ca un rezistor, cu excepția faptului că rezistența se modifică ca răspuns la cantitatea de lumină care cade asupra lor.

Acesta are o rezistență de aproximativ 50 k $\Omega$  în aproape întuneric și 500  $\Omega$  în lumină puternică.

Pentru a converti această valoare variabilă a rezistenței în ceva ce putem măsura pe intrarea analogică a plăcii UNO R3, aceasta trebuie convertită în tensiune.

Cel mai simplu mod de a face acest lucru este să îl combinați cu un fixresistor.

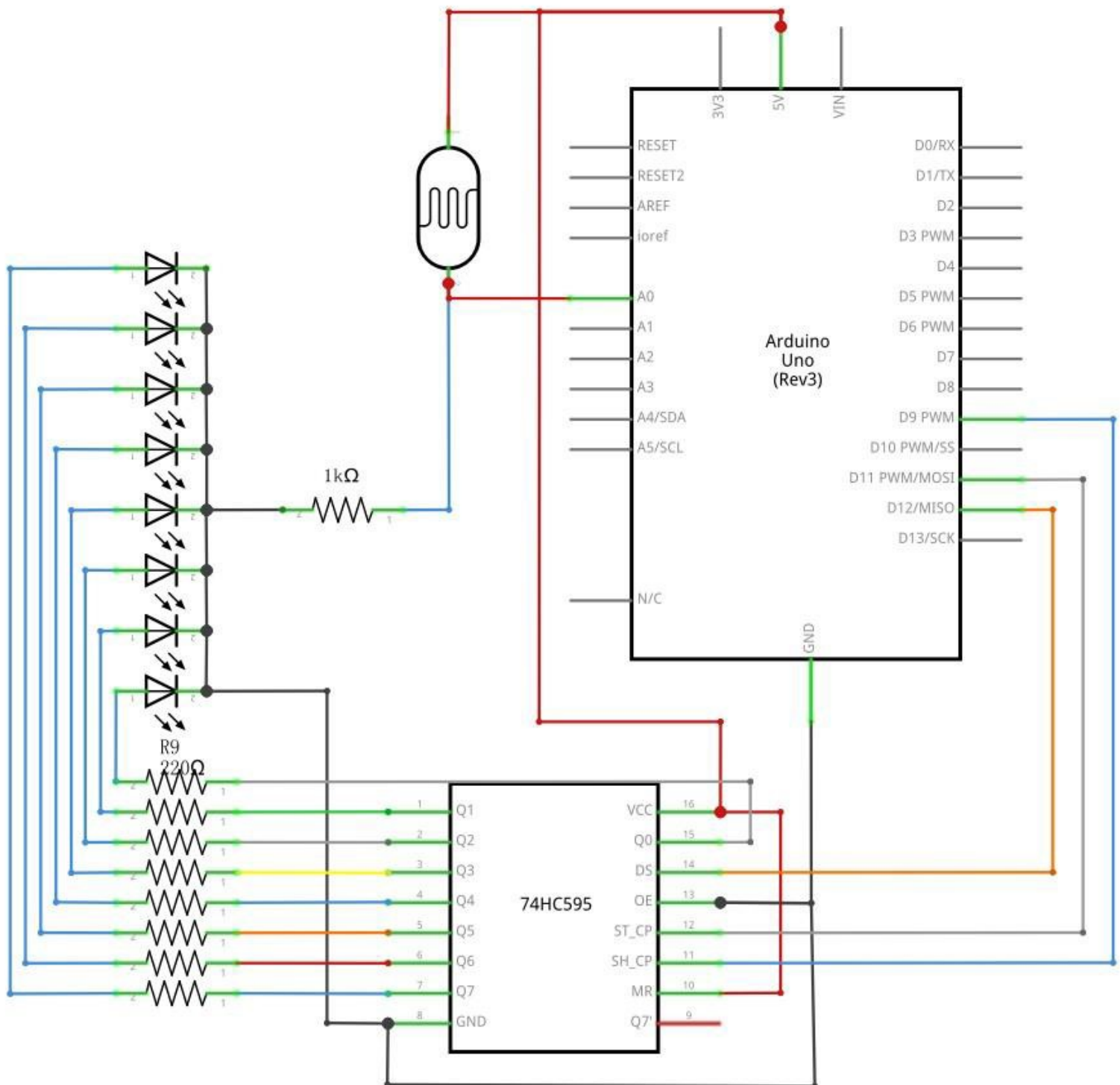


Rezistorul și fotocelula se comportă împreună ca o oală. Când lumina este foarte strălucitoare, atunci rezistența fotocelulei este foarte scăzută în comparație cu rezistența cu valoare fixă, și astfel este ca și cum oala ar fi întors la maxim.

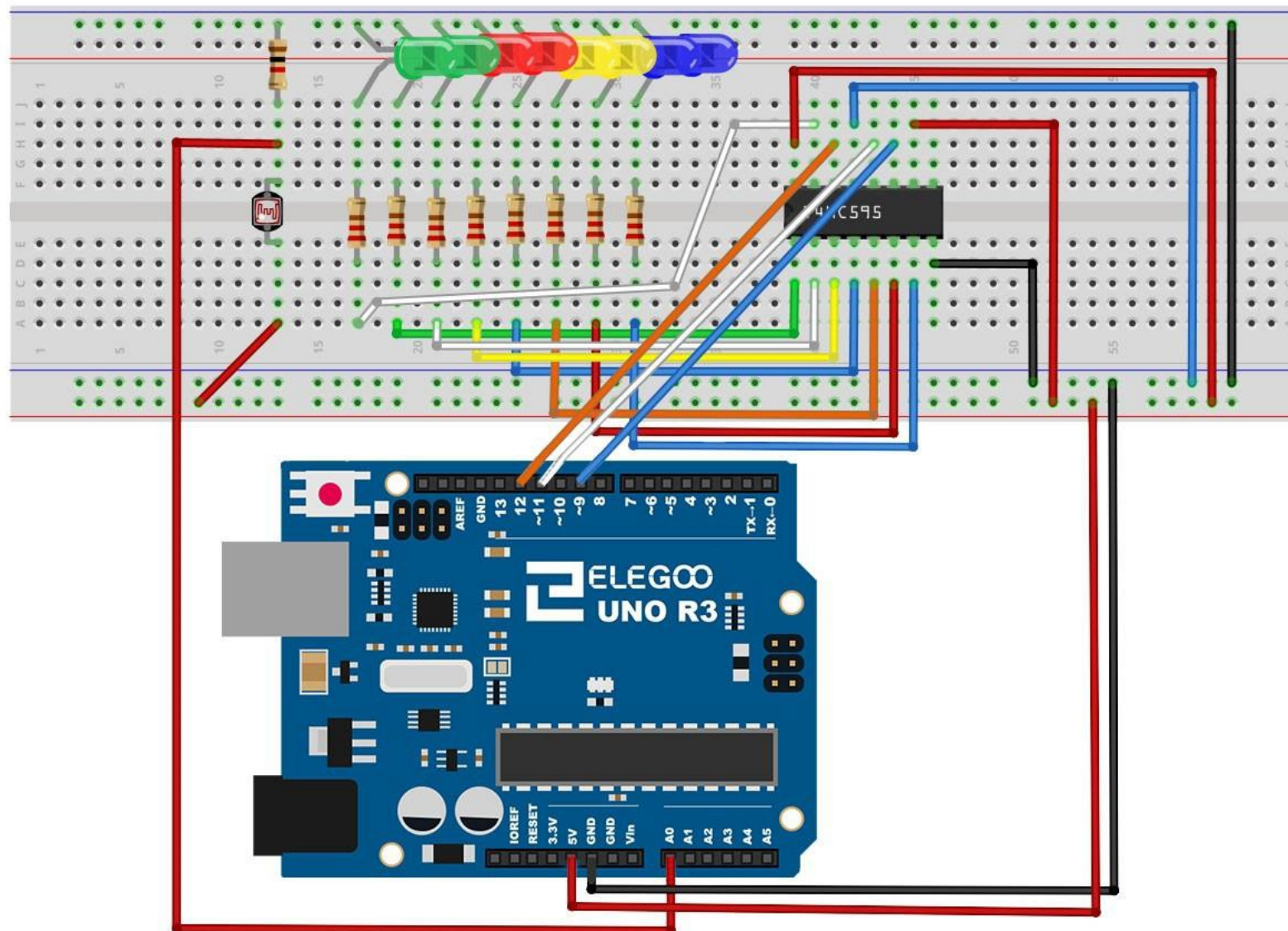
Când fotocelula este în lumină terasă, rezistența devine mai mare decât rezistența fixă de  $1\text{ k}\Omega$  și este ca și cum oala ar fi întors spre GND.

Încărcați schița dată în secțiunea următoare și încercați să acoperiți fotocelula cu degetul, apoi țineți-o lângă o sursă de lumină.

## Conexiune Schematic



Schema de cablare



## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 18 Celula foto și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

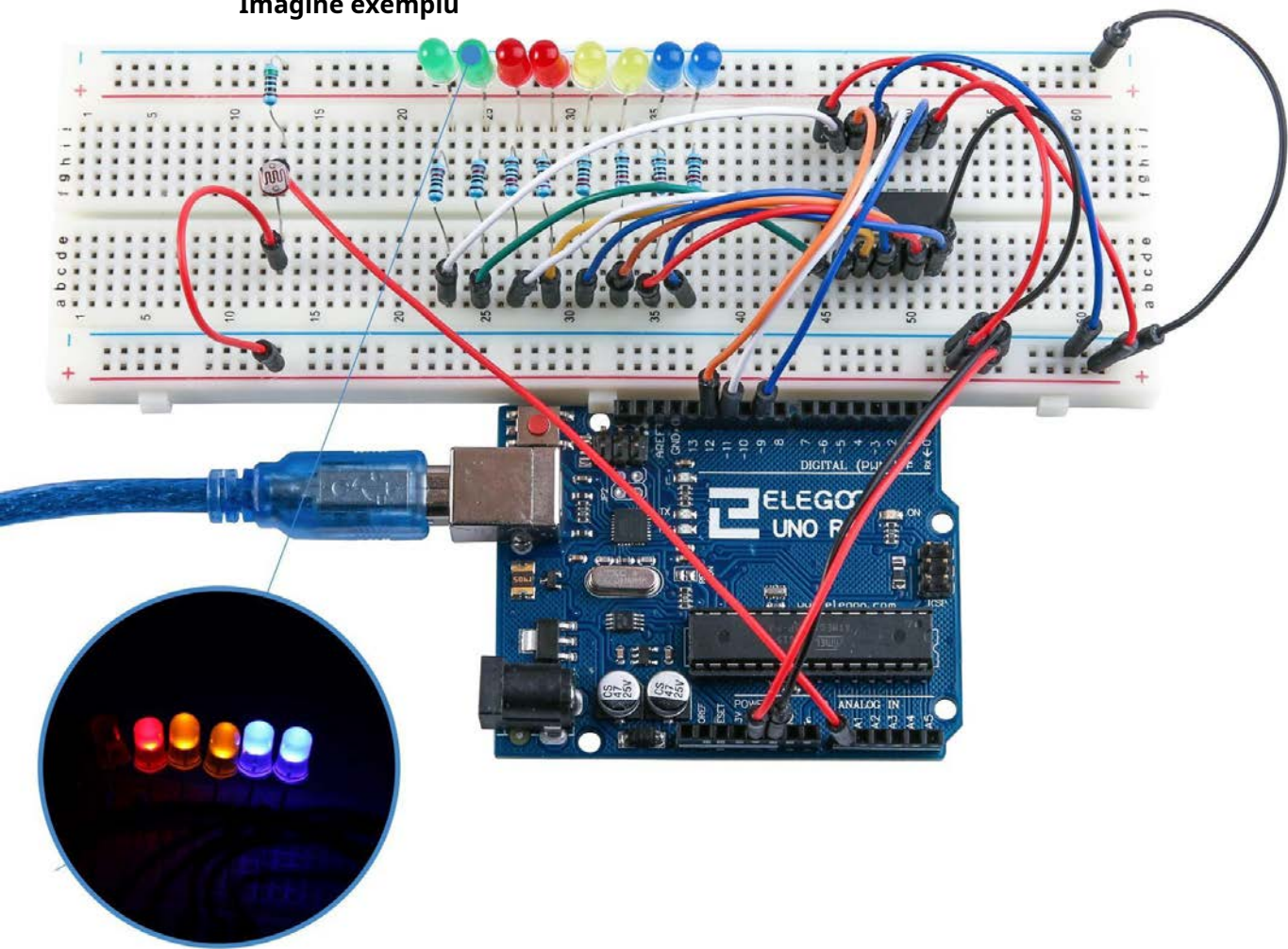
Primul lucru de remarcat este că am schimbat numele pinului analogic pentru a fi „lightPin” mai degrabă decât „potPin”, deoarece nu mai avem un pot conectat.

Singura altă modificare substanțială a schiței este linia care calculează câte LED-uri trebuie aprinse:

```
int numLEDsLit = citire / 57; // toate LED-urile aprinse la 1k
```

De data aceasta, împărțim citirea brută cu 57, în loc de 114. Cu alte cuvinte, o împărțim la jumătate din cât am făcut-o cu oala pentru a o împărți în nouă zone, de la niciun LED aprins la toate cele opt aprinse. Acest factor suplimentar trebuie să țină cont de rezistența fixă de 1 k $\Omega$ . Aceasta înseamnă că atunci când fotocelula are o rezistență de 1 k $\Omega$  (la fel ca și rezistența fixă), citirea brută va fi  $1023 / 2 = 511$ . Aceasta va echivala cu toate LED-urile aprinse și apoi un pic (numLEDsLit va fi 8).

### Imagine exemplu



## Lecția 19 74HC595 și afișare cu segmente

### Prezentare generală

După ce a învățat lecția 16, 17 și Lecția 18, vom folosi registrul de deplasare 74HC595 pentru a controla afișarea segmentului. Afișajul segmentului va afișa un număr de la 9 la 0.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x 830 de puncte de legătură

(1) x 74HC595 IC

(1) x 1 Afișaj cu 7 segmente

(8) rezistențe x 220 ohmi

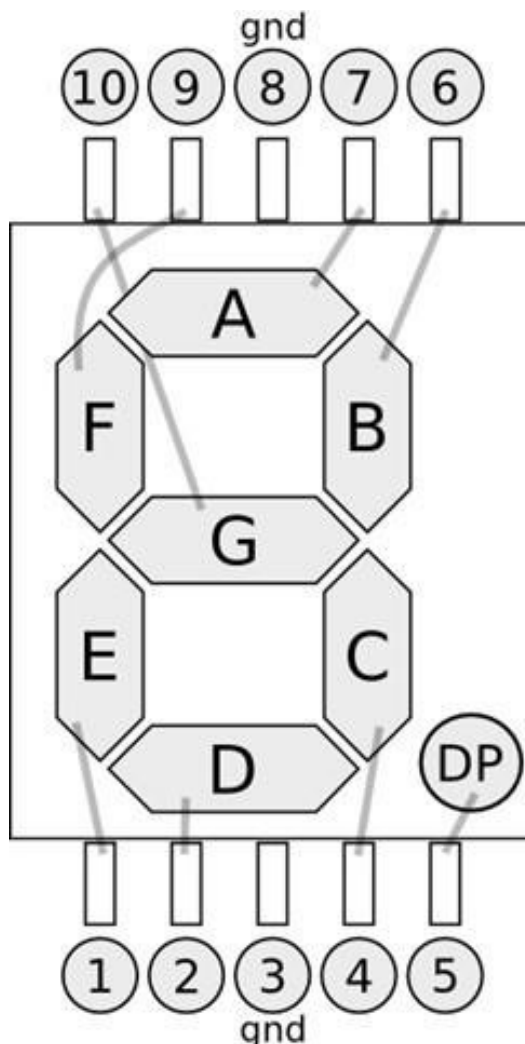
(26) fire x MM (fire de legătură de tip masculin la masculin)



### Introducere componente

#### Afișaj cu șapte segmente

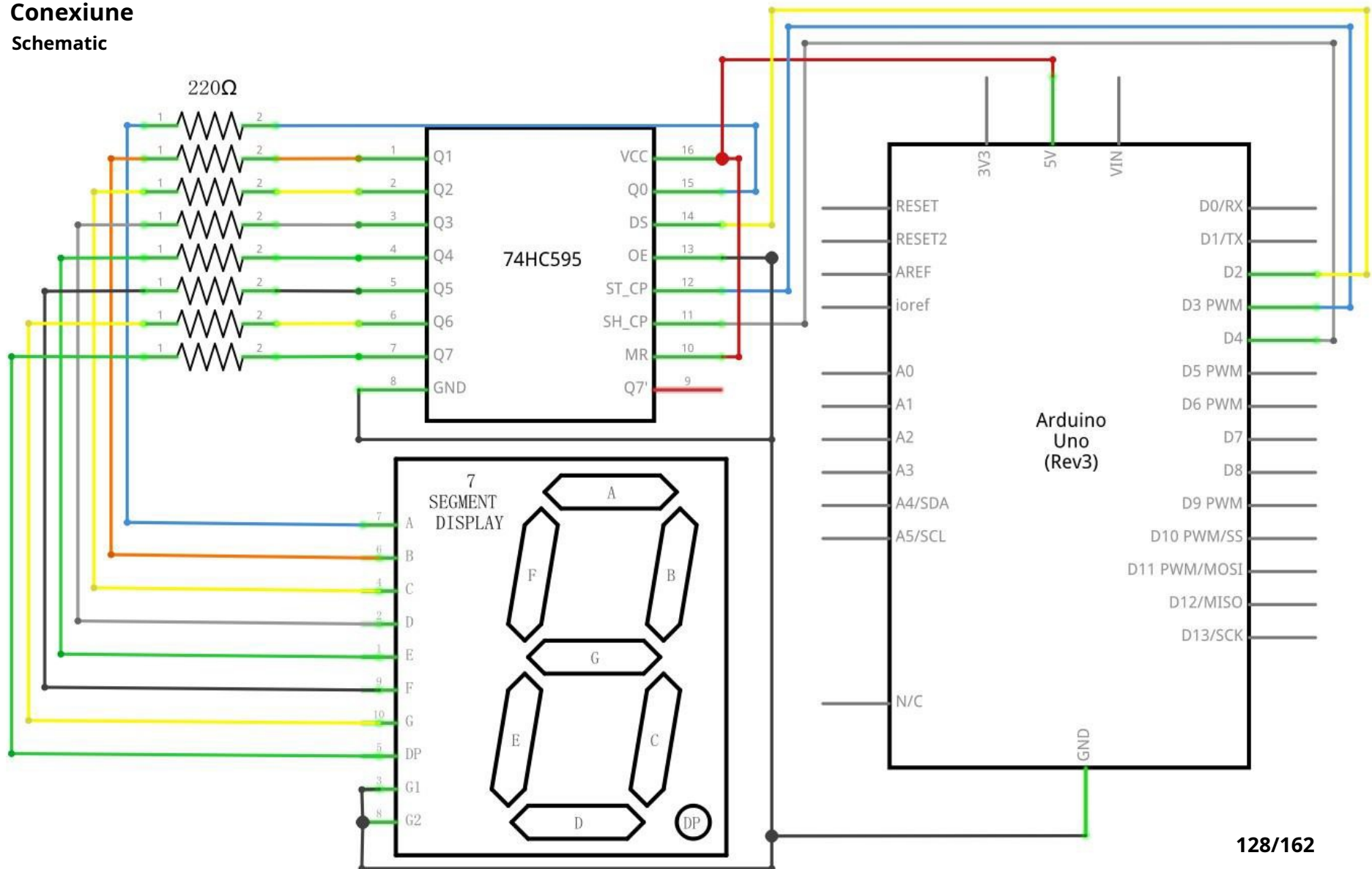
Mai jos este diagrama pin cu șapte segmente.



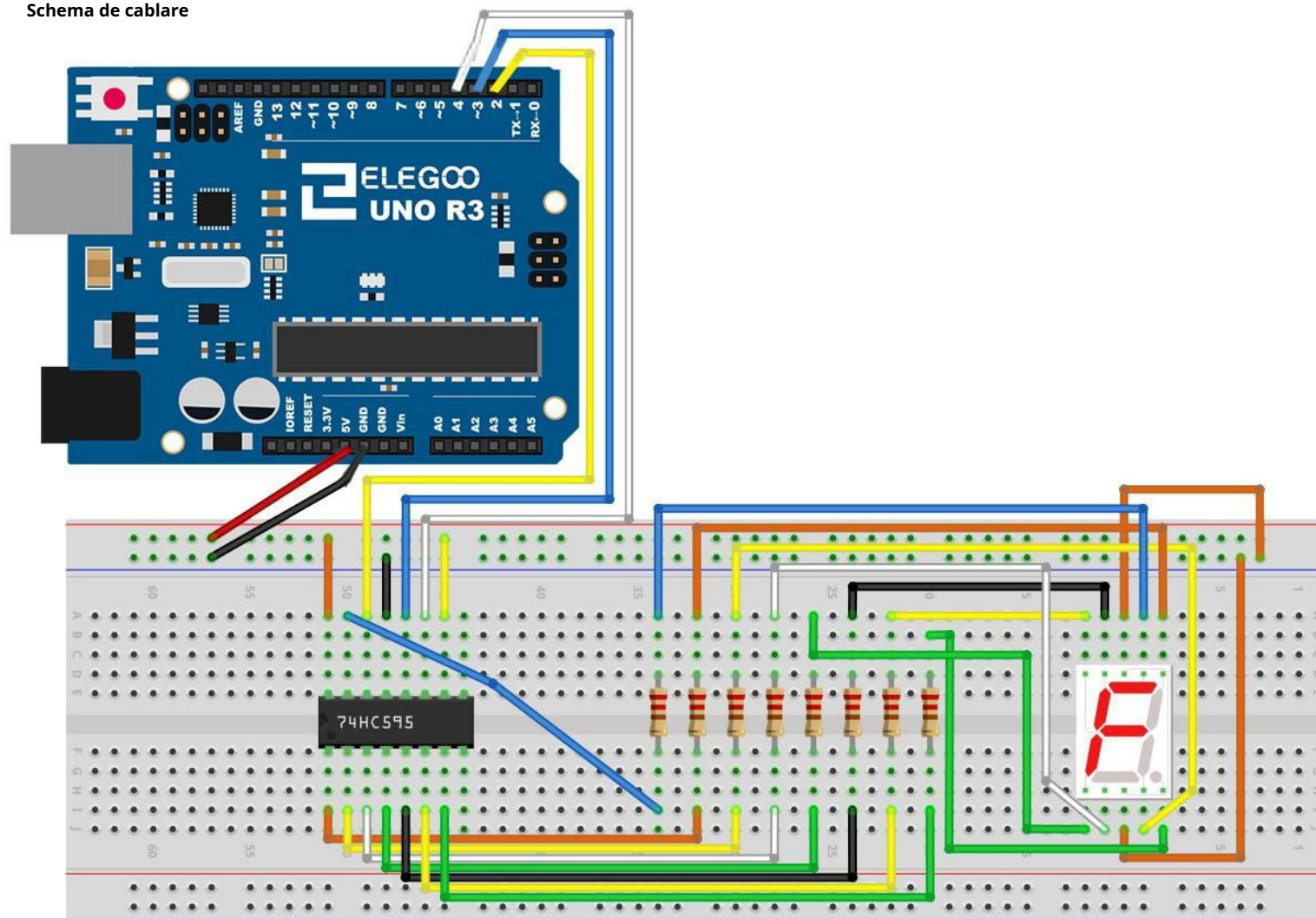
0-9 zece cifre corespund fiecărui segment sunt după cum urmează (tabelul următor se aplică dispozitivului de afișare cu șapte segmente cu catod comun, dacă utilizați un anod comun, tabelul trebuie înlocuit la fiecare 1 0 0 ar trebui să fie înlocuit cu 1):

| Displaydigital | dp | o | 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 |

## Conexiune Schematic



Schema de cablare



Următorul tabel arată tabelul de corespondență a pinii 74HC595 cu afișajul cu șapte segmente:

| 74HC595 pin | Șapte spectacole remarcabile<br>pin de control (cursă) |
|-------------|--------------------------------------------------------|
| Q0          | 7 (A)                                                  |
| Î1          | 6 (B)                                                  |
| Q2          | 4 (C)                                                  |
| Q3          | 2 (D)                                                  |
| Î4          | 1 (E)                                                  |
| Î5          | 9 (F)                                                  |
| Î6          | 10 (G)                                                 |
| Î7          | 5 (DP)                                                 |

Pasul unu: Conectați 74HC595

În primul rând, cablajul este conectat la alimentare și

la masă: **VCC**(pin 16) și **MR** (pin 10) conectate la 5V

**GND**(pin 8) și **OE** (pin 13) la masă Conexiune DS,

**ST\_CP** și **SH\_CP** pin:

**DS**(pinul 14) conectat la pinul 2 al plăcii UNO R3 (figura de sub linia galbenă) **ST\_CP**(pinul 12, pinul de blocare) conectat la pinul 3 al plăcii UNO R3 (linia albastră FIG de mai jos) **SH\_CP**(pinul 11, pinul ceasului) conectat la pinul 4 al plăcii UNO R3 (figura de sub linia albă)

Pasul doi: Conectați afișajul cu șapte segmente

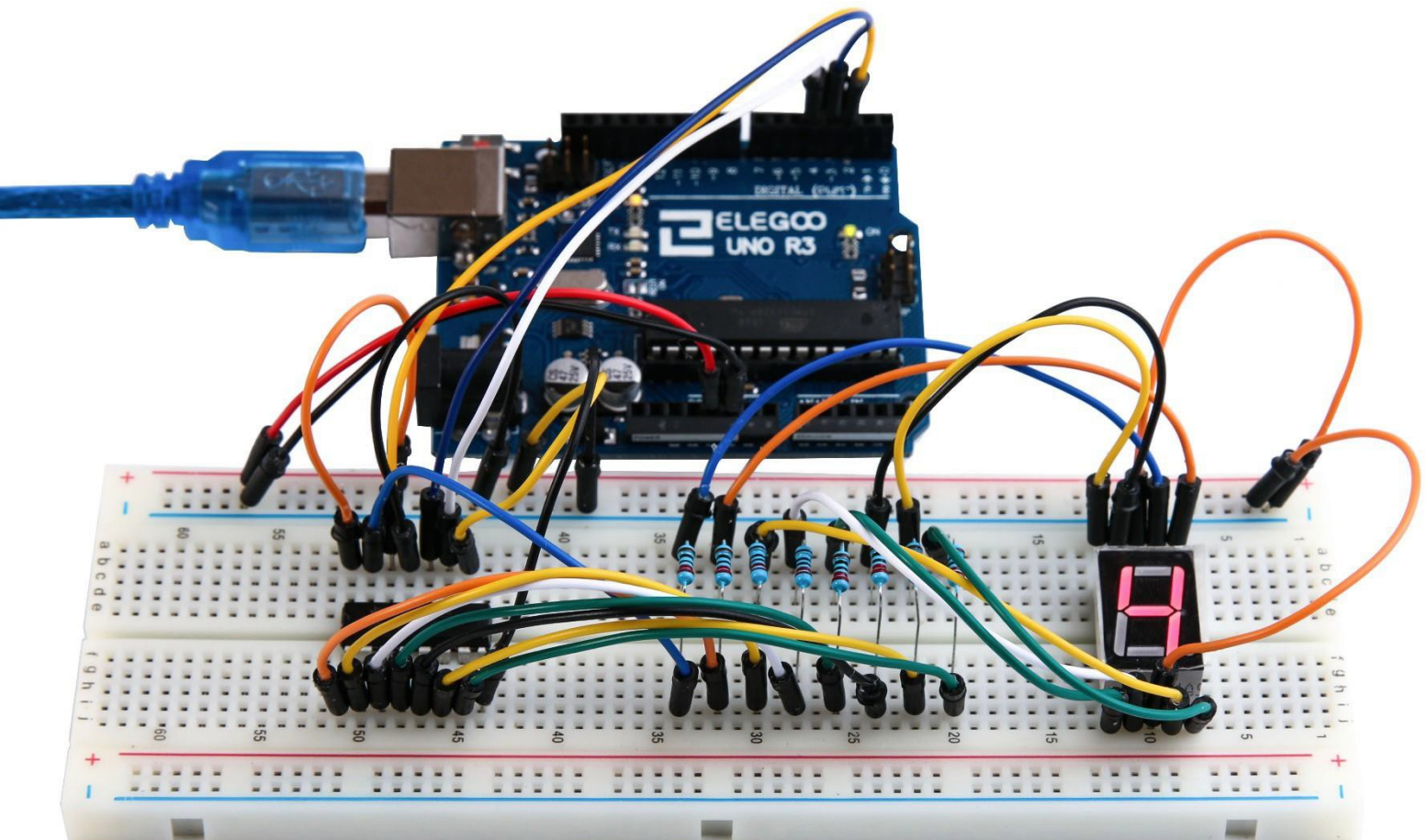
Afișajul cu șapte segmente 3, 8 pini la placa UNO R3 GND (Acest exemplu folosește catodul comun, dacă utilizați anodul comun, vă rugăm să conectați 3, 8 pini la placa UNO R3 + 5V)

Conform tabelului de mai sus, conectați 74HC595 Q0 ~ Q7 la pinul corespunzător al afișajului cu șapte segmente (A ~ G și DP), apoi fiecare picior într-un rezistor de 220 ohmi în serie.

## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 19 74HC595 și afișare segment și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

## Imagine exemplu



## Lecția 20 Patru Afișaj digital cu șapte segmente

### Prezentare generală

În această lecție, veți învăța cum să utilizați un afișaj cu 4 cifre și 7 segmente.

Când utilizați un afișaj cu 1 cifre și 7 segmente, vă rugăm să observați că, dacă este un anod comun, pinul anodului comun se conectează la sursa de alimentare; dacă este un catod comun, pinul catodului comun se conectează la GND.

Când utilizați un afișaj cu 4 cifre și 7 segmente, anodul comun sau pinul catodului comun este utilizat pentru a controla ce cifră este afișată. Chiar dacă există o singură cifră care funcționează, principiul Persistenței vederii vă permite să vedeți toate numerele afișate, deoarece fiecare viteză de scanare este atât de rapidă încât cu greu observați intervalele.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (1) x 74HC595 IC
- (1) x 4 cifre cu 7 segmente
- (4) rezistențe x 220 ohmi
- (23) fire x MM (fire jumper tată la masculin)

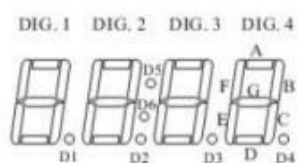
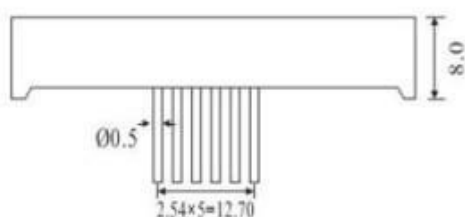
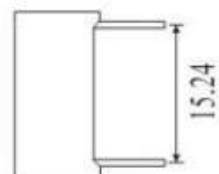
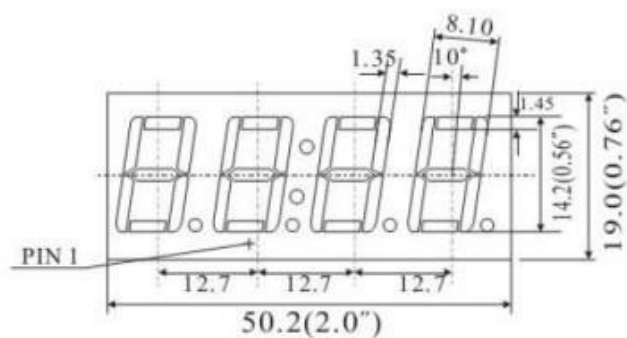


## Introducere componente

Afişaj cu patru segmente digitale cu şapte

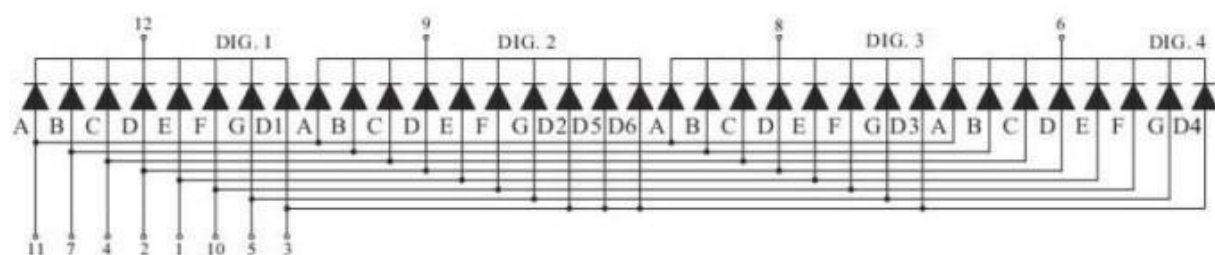
### Package Dimensions

**CPS05643AB**

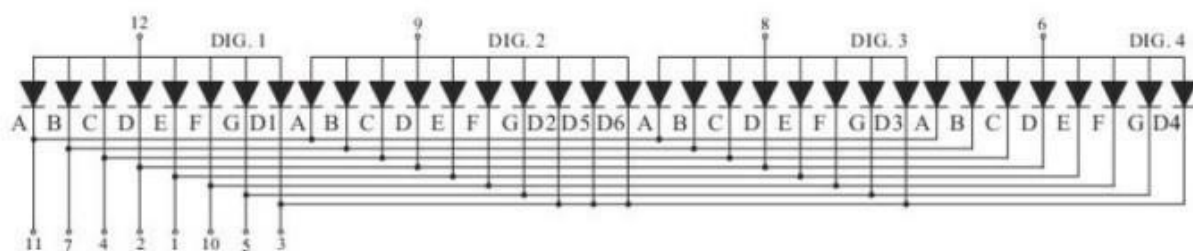


UNIT: MM(INCH) TOLERANCE:  $\pm 0.25(0.01")$

### Internal Circuit Diagram



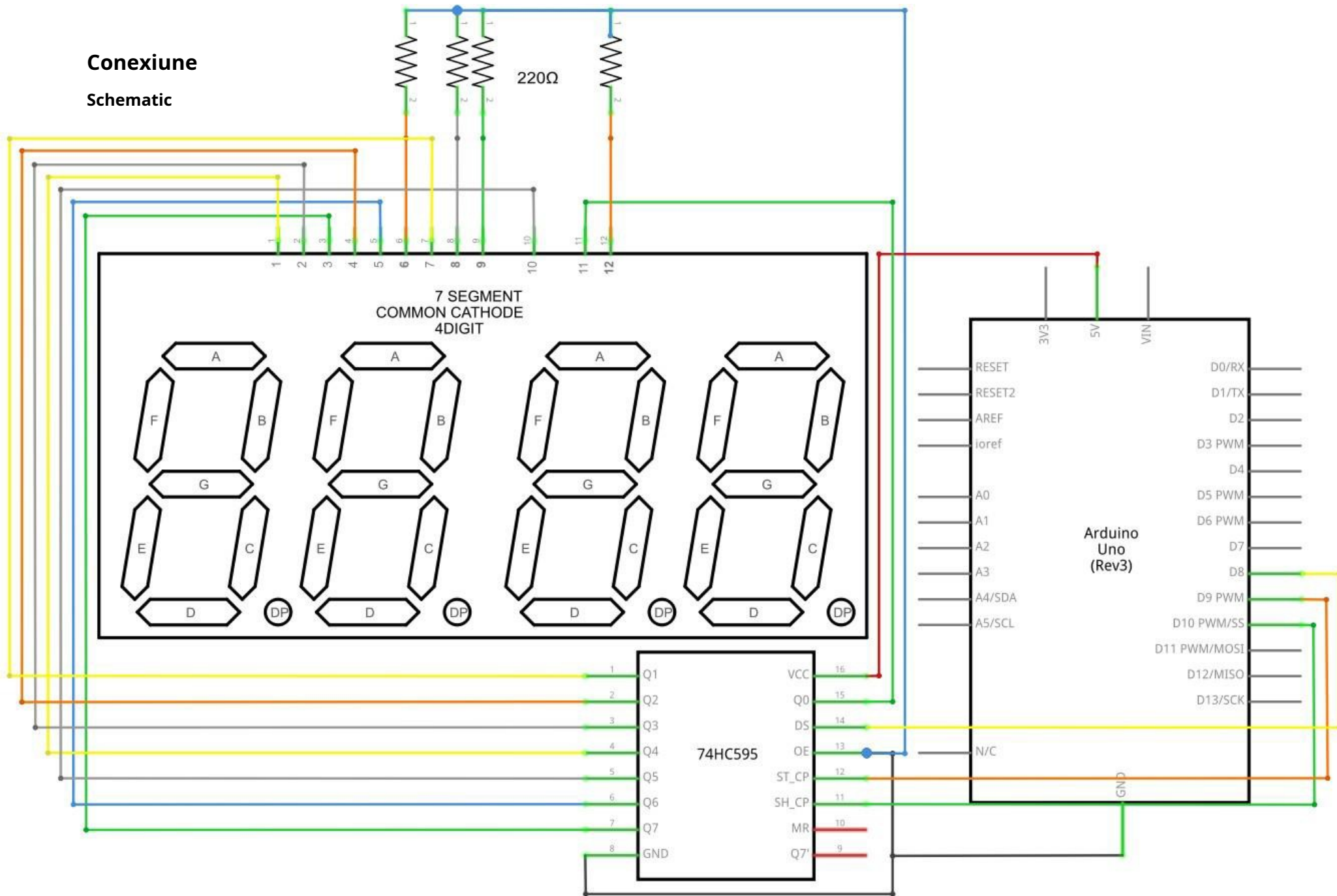
**5643A**



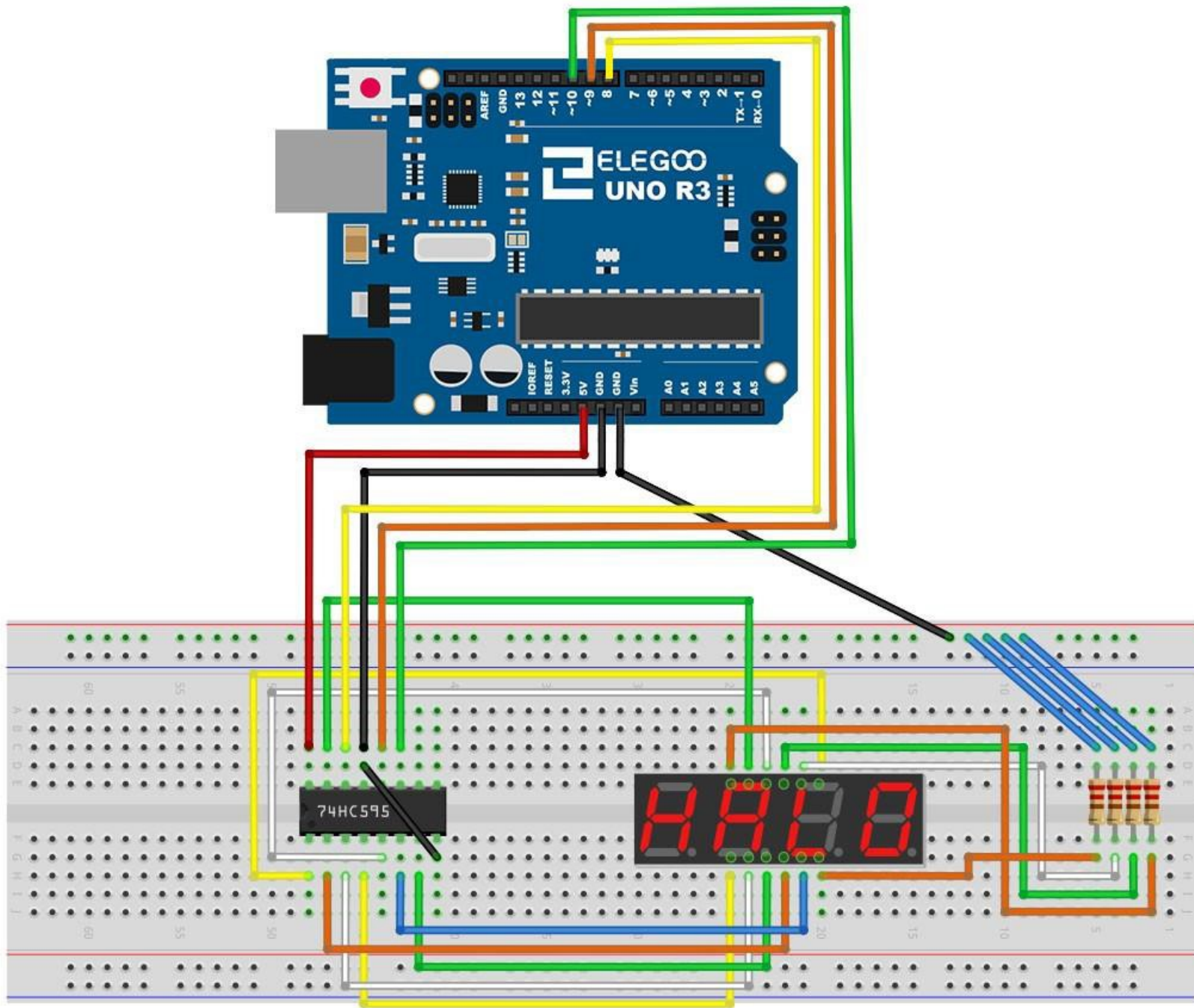
**5643B**

# Conexiune

## Schematic



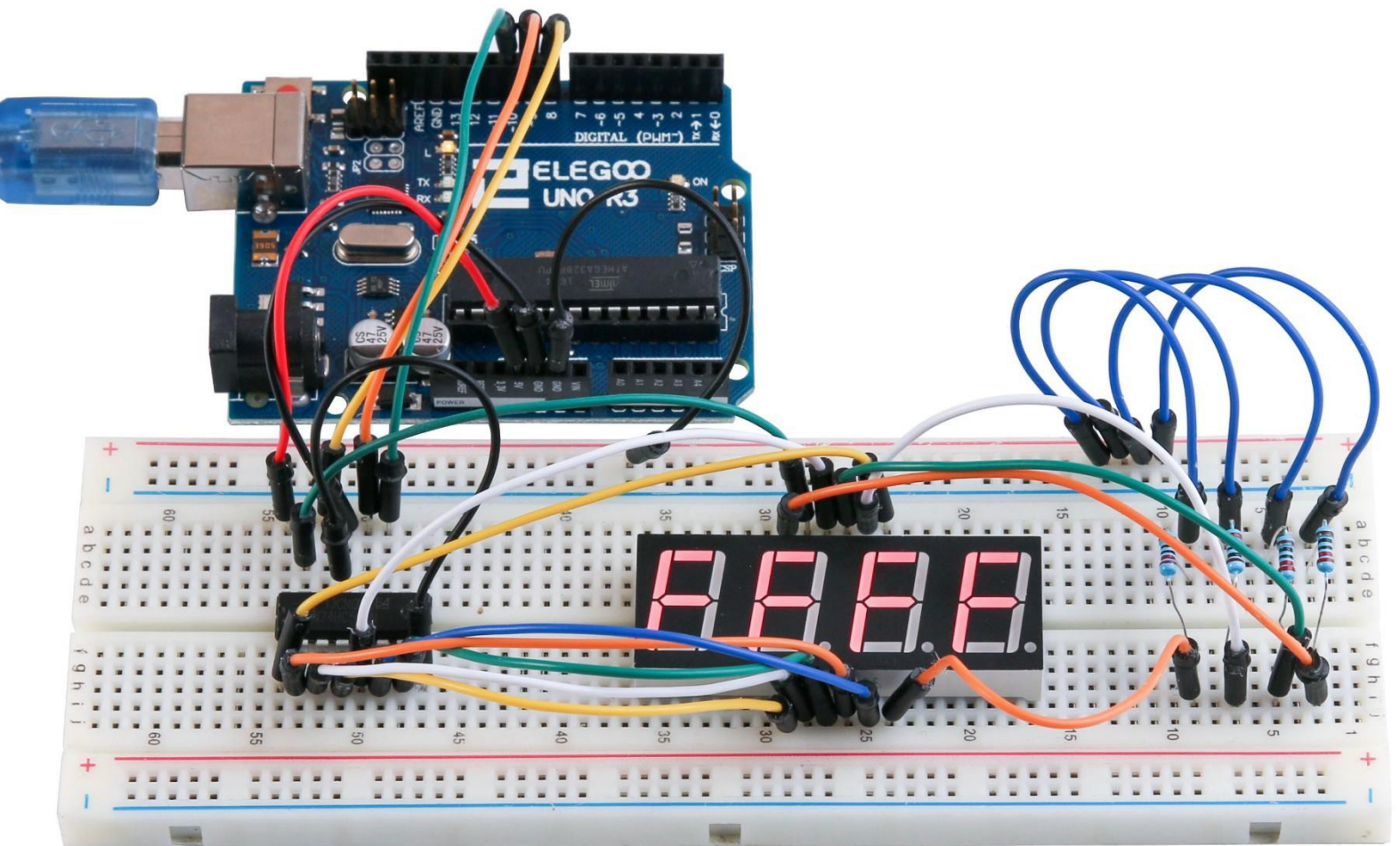
### Schema de cablare



## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 20 Afișare cu patru segmente digitale](#) și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

## Imagine exemplu



### Prezentare generală

În această lecție, veți învăța cum să controlați un mic motor DC folosind un UNO R3 și un tranzistor.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (1) x L293D IC
- (1) x lamă ventilator și motor 3-6v
- (5) fire x MM (fire de legătură de sex masculin la masculin)
- (1) x Modul de alimentare
- (1) x adaptor 9V1A

### Introducere componente

#### Sursă de alimentare pentru placa

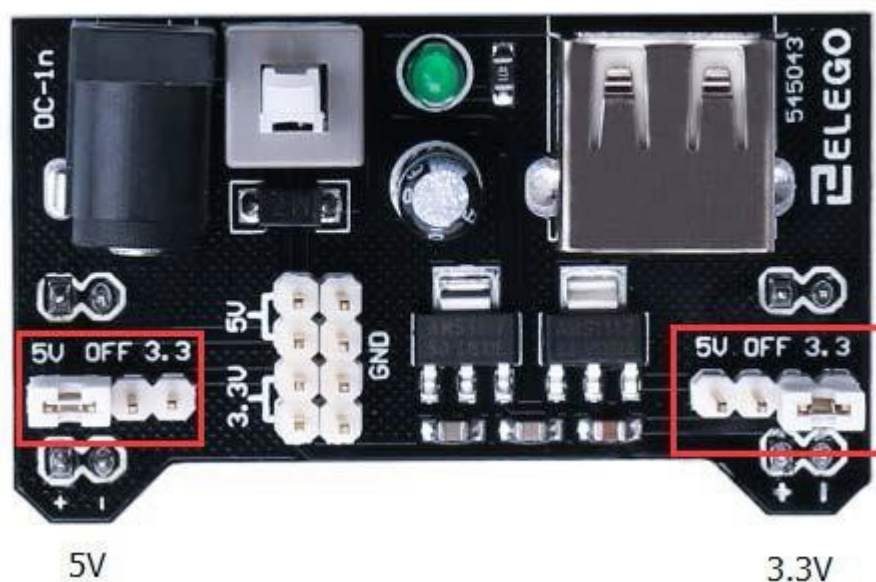
Micul motor de curent continuu este probabil să utilizeze mai multă putere decât o poate gestiona direct o ieșire digitală a plăcii UNO R3. Dacă am încercat să conectăm motorul direct la un pin al plăcii UNO R3, există șanse mari ca acesta să deterioreze placa UNO R3. Deci, folosim un modul de alimentare care asigură alimentarea cu energie



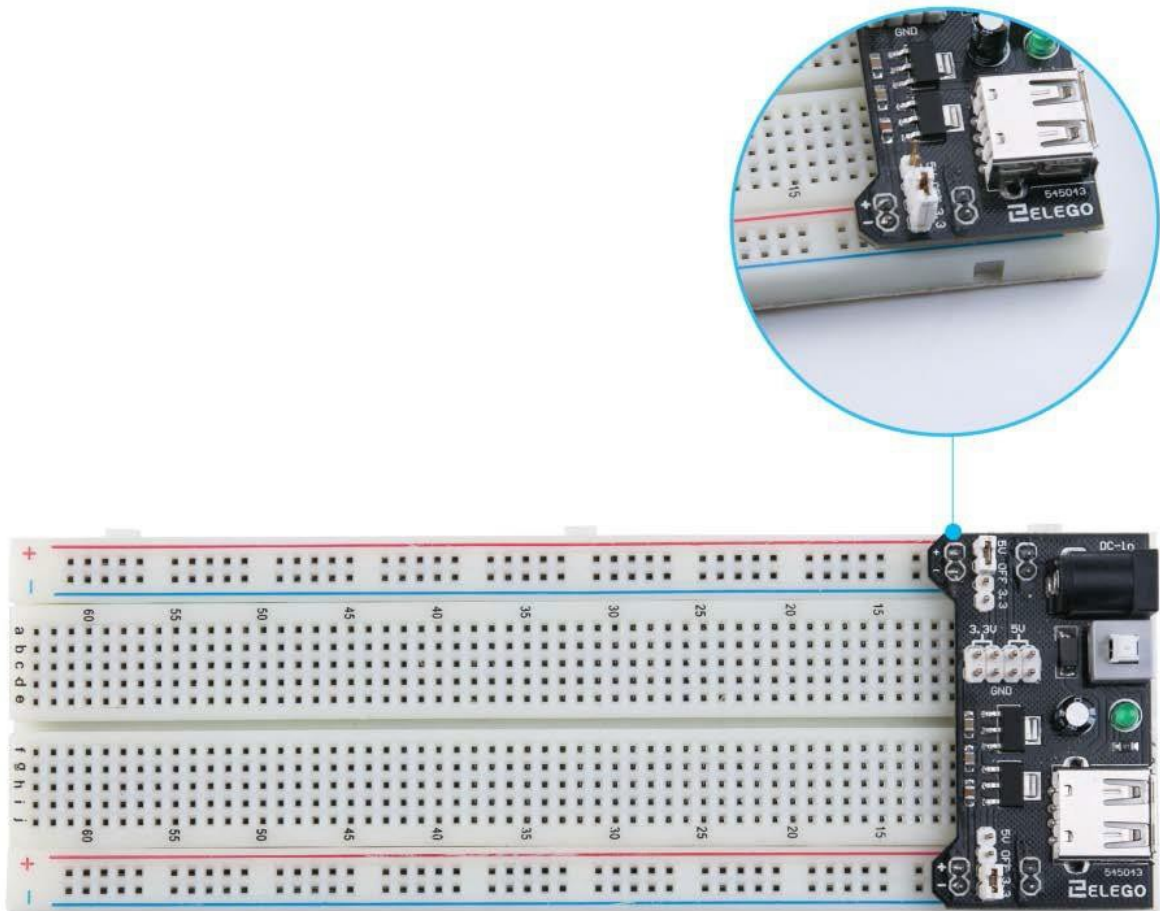
### Specificatii produsului:

- Comutator de blocare pornit/oprit
- Indicator LED de alimentare
- Tensiune de intrare: 6,5-9 V (DC) prin mufa de 5,5 mm x 2,1 mm
- Tensiune de ieșire: 3,3 V/5 V
- Curent maxim de ieșire: 700mA
- Ieșire șină de control independentă. 0v, 3.3v, 5v la placa de laborator
- Pini antet de ieșire pentru utilizare externă convenabilă
- Dimensiune: 2,1 in x 1,4 in
- Conector de dispozitiv USB la bord pentru a alimenta dispozitivul extern

### Configurarea tensiunii de ieșire:



Ieșirea de tensiune din stânga și din dreapta poate fi configurată independent. Pentru a selecta tensiunea de ieșire, mutați jumperul la pinii corespunzători. Notă: LED-ul indicator de alimentare și șinele de alimentare ale plăcii nu se vor porni dacă ambele jumperi sunt în poziția „OPRIT”.



### Notă importantă:

Asigurați-vă că aliniați corect modulul pe placa. Pinul negativ (-) de pe modul se aliniază cu linia albastră (-) de pe placa de breadboard și pinul pozitiv (+) se aliniază cu linia roșie (+). Nerespectarea acestui lucru ar putea duce la inversarea accidentală a alimentării proiectului dumneavoastră

### L293D

Acesta este un cip foarte util. De fapt, poate controla două motoare independent. Folosim doar jumătate din cip în această lecție, majoritatea pinii din partea dreaptă a cipului sunt pentru controlul unui al doilea motor.



## Specificatii produsului:

- Dispunând de produse Unitrode L293 și L293D acum de la Texas Instruments
- Gamă largă de tensiune de alimentare: 4,5 V până la 36 V
- Alimentare de intrare-logică separată
- Protecție ESD internă
- Opreire termică
- Intrări cu imunitate ridicată la zgomot
- Asemănător funcțional cu SGS L293 și SGS L293D
- Curent de ieșire 1 A pe canal (600 mA pentru L293D)
- Curent de ieșire de vârf 2 A per canal (1,2 A pentru L293D)
- Diode cu clemă de ieșire pentru suprimarea tranzitorilor inductivi (L293D)

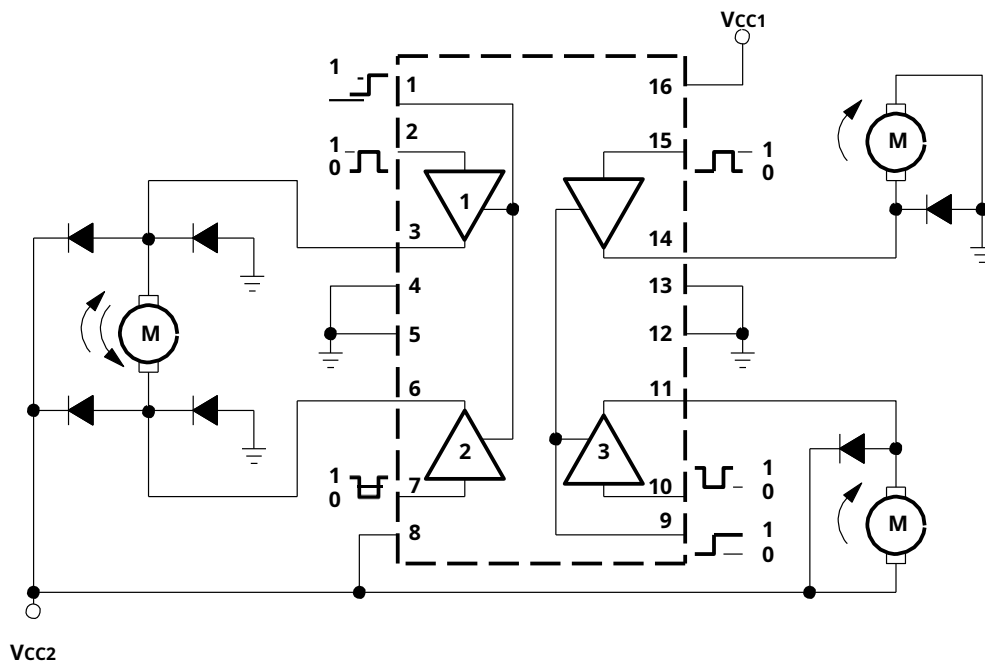


## Descriere/informații de comandă

L293 și L293D sunt drivere cvadruple semi-H cu curent ridicat. L293 este proiectat să furnizeze curenți bidirecționali de acționare de până la 1 A la tensiuni de la 4,5 V la 36 V. L293D este proiectat să furnizeze curenți bidirecționali de acționare de până la 600 mA la tensiuni de la 4,5 V la 36 V. Ambele dispozitive sunt concepute pentru a conduce sarcini inductive, cum ar fi relee, solenoizi, motoare pas cu curent continuu și bipolare, precum și alte sarcini de curent/întă tensiune în aplicații cu alimentare pozitivă.

Toate intrările sunt compatibile TTL. Fiecare ieșire este un circuit complet de antrenare totem-pole, cu o chiuvetă de tranzistor Darlington și o sursă pseudo-Darlington. Driverele sunt activate în perechi, driverele 1 și 2 activate de 1,2EN și driverele 3 și 4 activate de 3,4EN. Când o intrare de activare este ridicată, driverele asociate sunt activate, iar ieșirile lor sunt active și în fază cu intrările lor. Când intrarea de activare este scăzută, acele drivere sunt dezactivate, iar ieșirile lor sunt oprite și în starea de impedanță ridicată. Cu intrările adecvate de date, fiecare pereche de drivere formează o unitate reversibilă full-H (sau punte) potrivită pentru aplicații cu solenoid sau motor.

## Diagrama bloc



M-am săturat de diagrame de pinout indescifrabile din foile de date, așa că mi-am conceput-o pe ale mele care cred că oferă informații mai pertinente.

Există 3 fire conectate la Arduino, 2 fire conectate la motor și 1 fir conectat la o baterie.

**Notă: Pinul 16 trebuie alimentat cu curent, L293D poate funcționa normal.**

| 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 |                  |

Pentru a utiliza acest pinout:

Partea stângă se ocupă de primul motor, partea dreaptă se ocupă de un al doilea motor.

Da, îl puteți rula cu un singur motor conectat.

### Conexiuni Arduino

M1 PWM - conectați-l la un pin PWM de pe Arduino. Sunt etichetate pe Uno, pinul 5 este un exemplu.

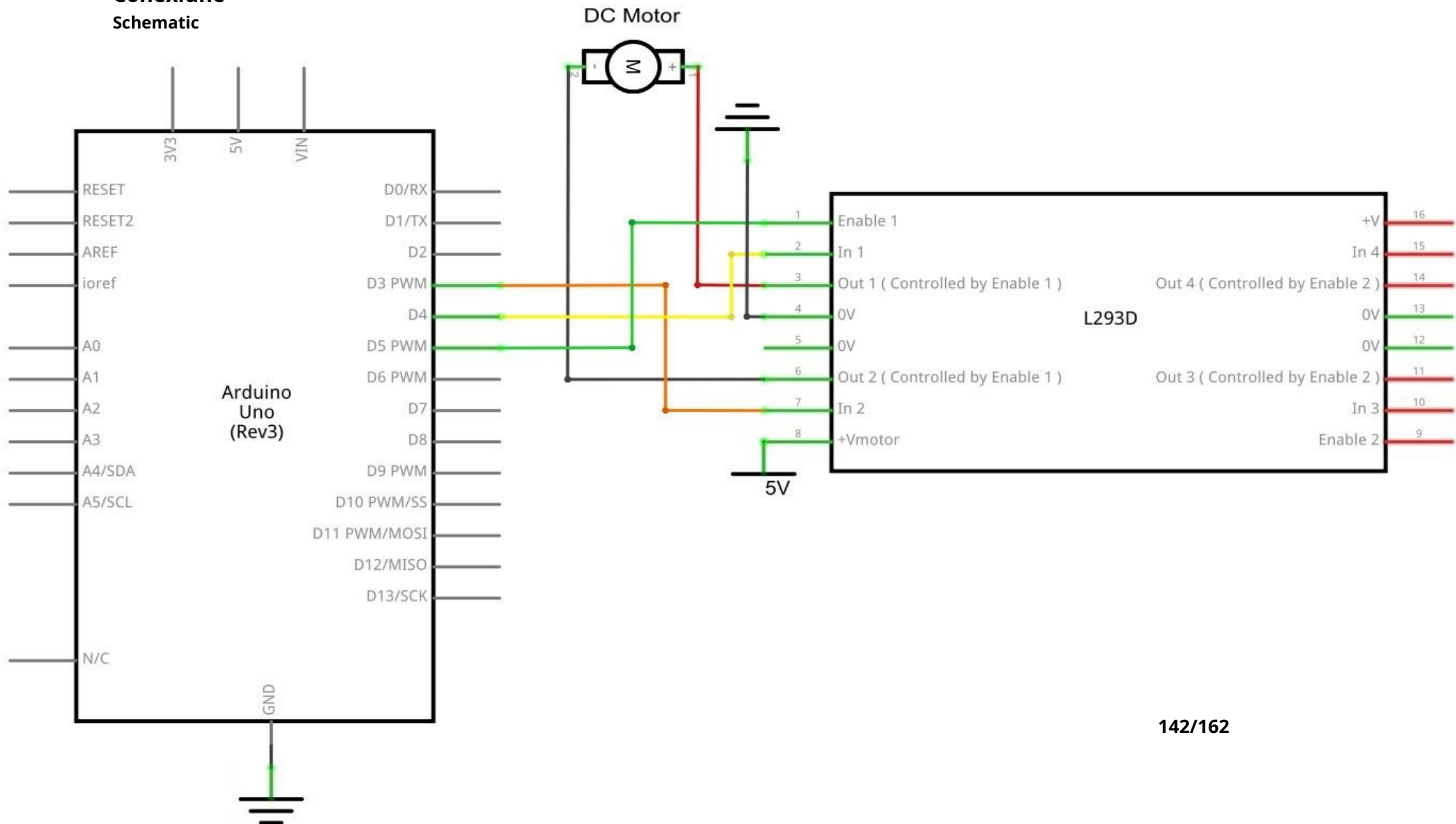
Ieșiți orice număr întreg între 0 și 255, unde 0 va fi oprit, 128 este jumătate de viteză și 255 este

viteza maxima.

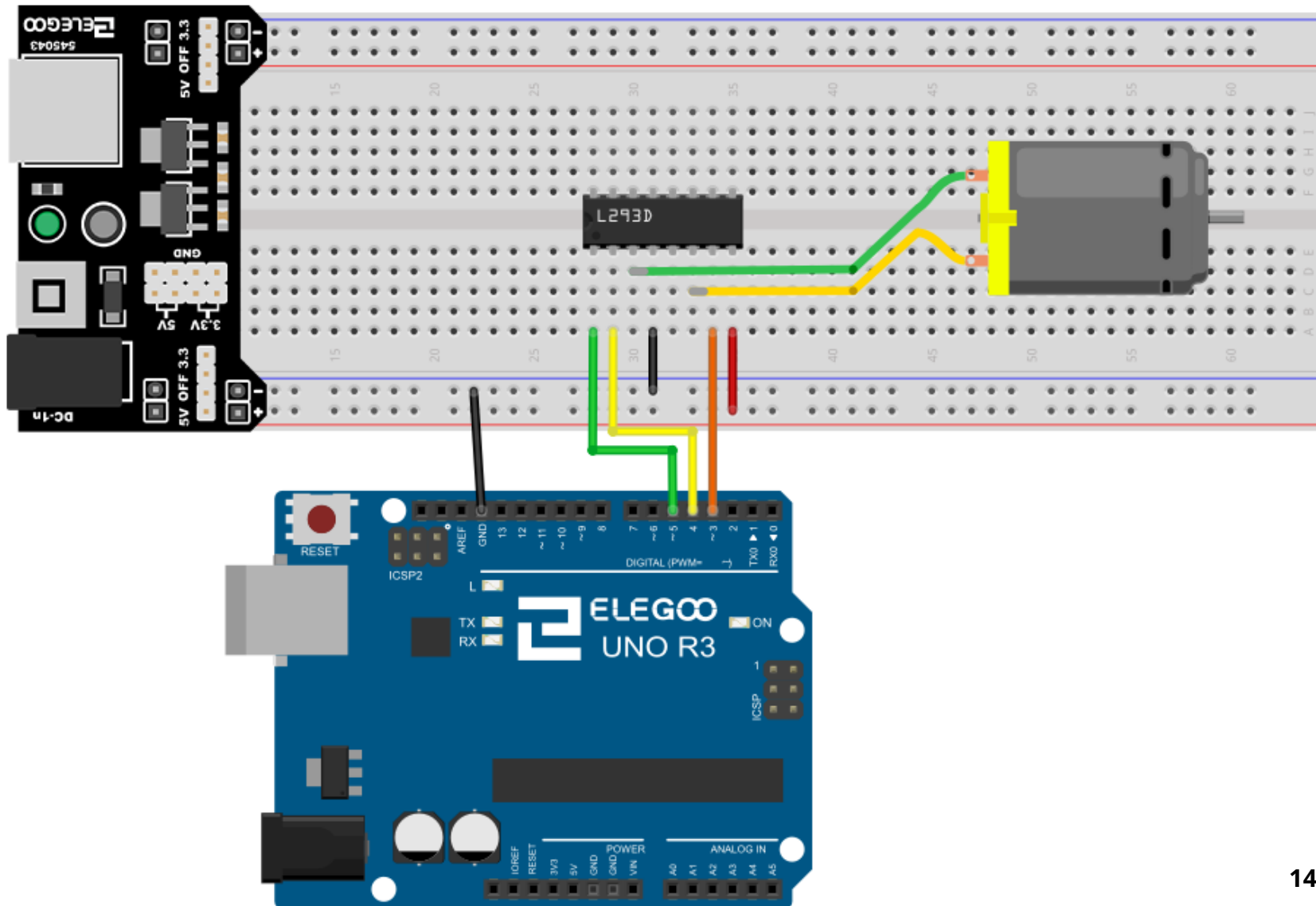
Direcția M1 0/1 și direcția M1 1/0 - Conectați acești doi la doi pini Arduino digitali. Ieșiți un pin ca HIGH și celălalt pin ca LOW, iar motorul se va învârti într-o direcție.

Inversați ieșirile la LOW și HIGH, iar motorul se va învârti în cealaltă direcție.

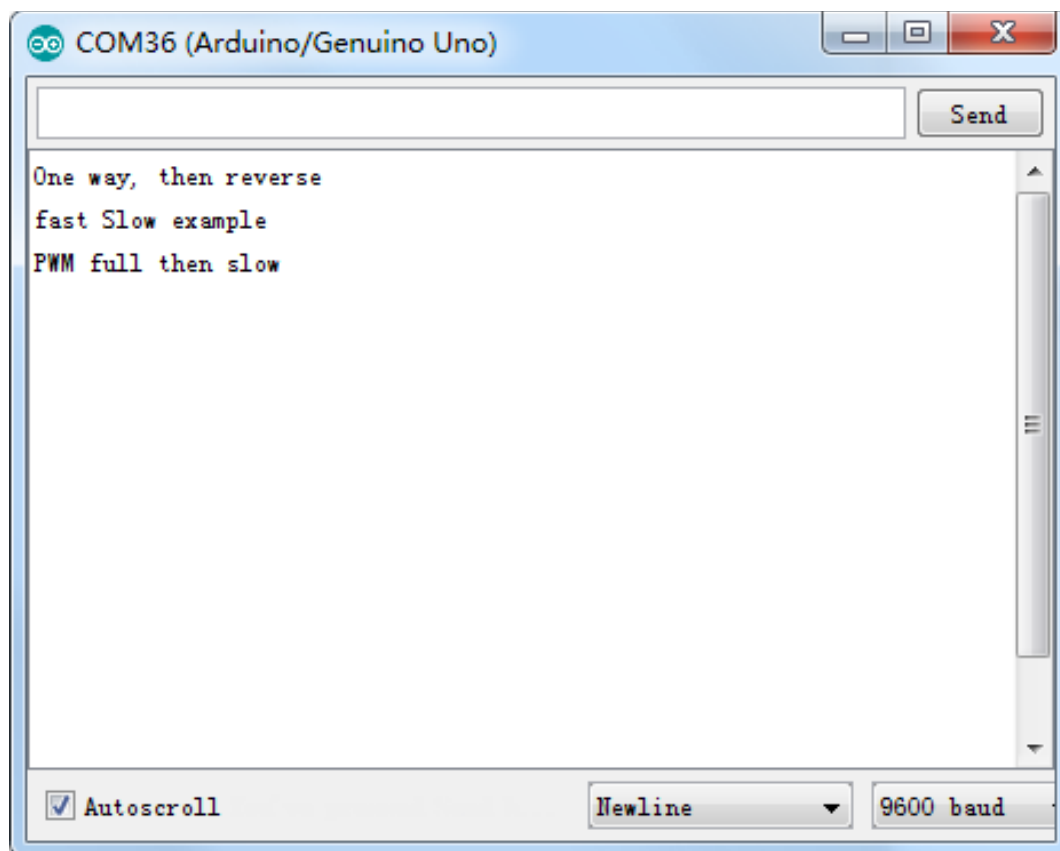
## Conexiune Schematic



## Schema de cablare



Nu ar trebui să conectați niciodată un motor direct la Arduino, deoarece atunci când opriți un motor primiți un feedback electric. Cu un motor mic, acest lucru vă va deteriora Arduino, iar cu un motor mare, puteți urmări un efect interesant de flacără și scântei.

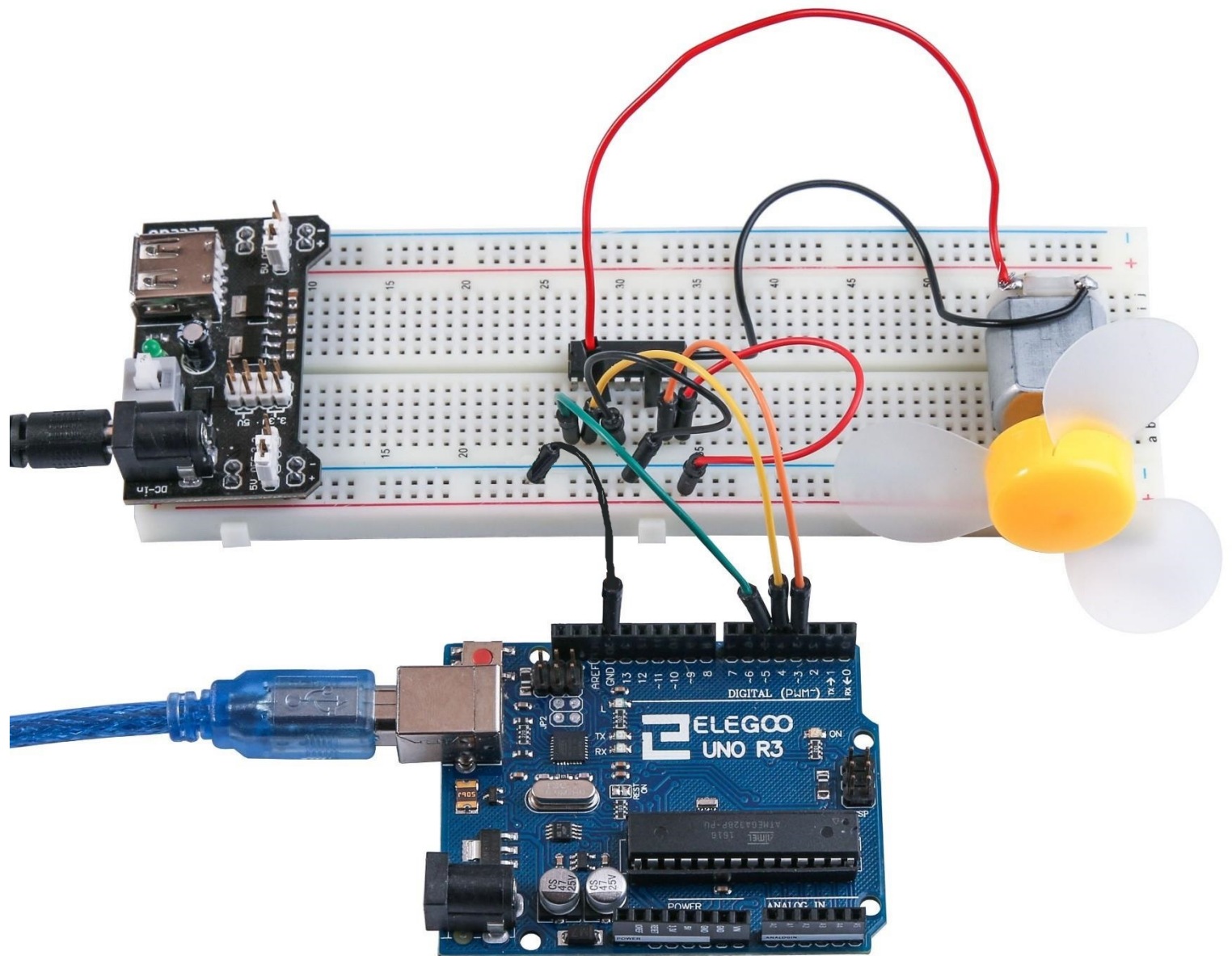


## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - [Lecția 21 Motoare CC și faceți clic pe ÎNCĂRCARE](#) pentru a încărca programul. Consultați [Lecția 2](#) pentru detalii despre încărcarea programului dacă există erori.

După încărcarea programului, porniți toate comutatoarele de alimentare. Motorul se va roti ușor în sensul acelor de ceasornic și în sens invers acelor de ceasornic de 5 ori. Apoi, va continua să se rotească dramatic în sensul acelor de ceasornic. După o scurtă pauză, se va roti dramatic în sens invers acelor de ceasornic. Apoi placa de control va trimite semnal PWM pentru a conduce motorul, motorul își va reduce încet RPM maxim la minim și va crește din nou la maxim. În cele din urmă, se oprește timp de 10 secunde până când începe următorul ciclu.

Imagine exemplu



## Lecția 22 Stafeta

### Prezentare generală

În această lecție, veți învăța cum să utilizați un releu.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (1) x lamă ventilator și motor 3-6v dc
- (1) x L293D IC
- (1) x releu 5v
- (1) x Modul de alimentare
- (1) x adaptor 9V1A
- (8) fire x MM (fire de legătură de tip masculin la masculin)



## Introducere componente

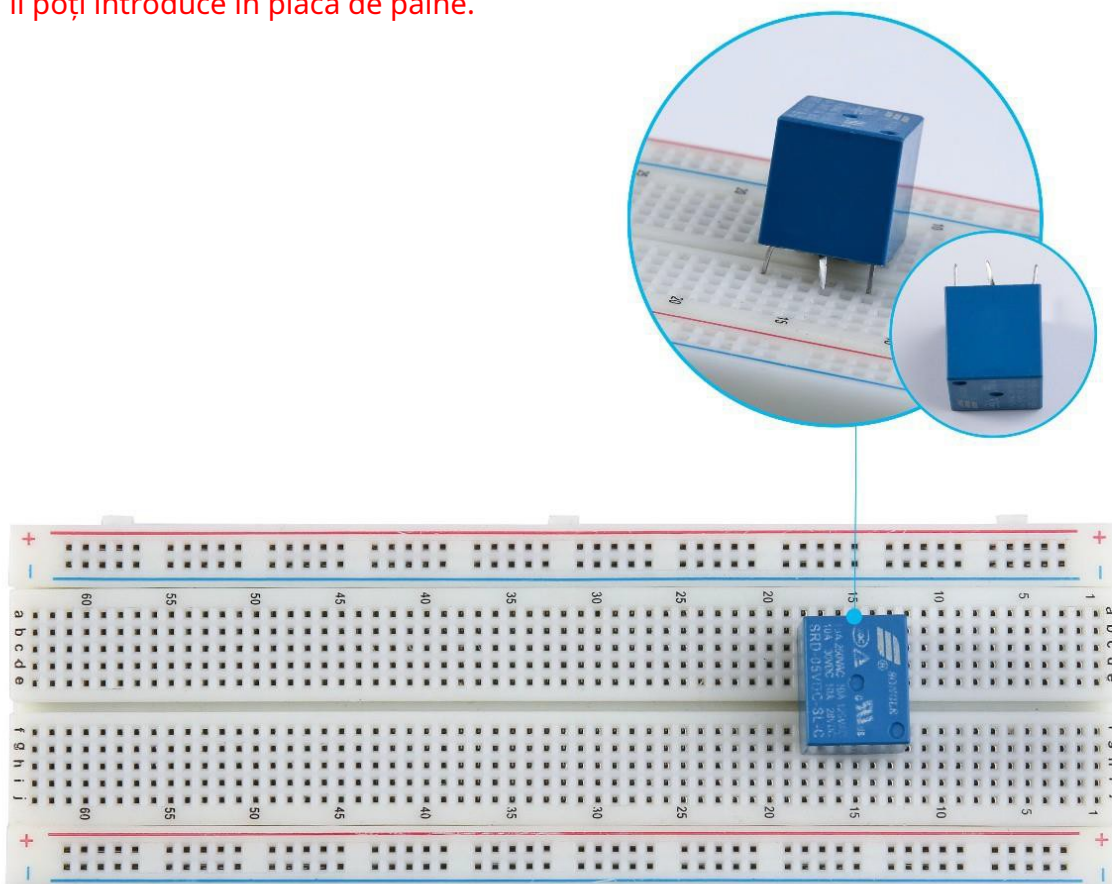
### Releu:

Un releu este un comutator actionat electric. Multe releu folosesc un electromagnet pentru a acționa mecanic un comutator, dar alte principii de funcționare sunt, de asemenea, utilizate ca în releele cu stare solidă. Releele sunt utilizate acolo unde este necesar să se controleze un circuit printr-un semnal de putere redusă (cu izolare electrică completă între circuitele de comandă și cele controlate), sau unde mai multe circuite trebuie controlate printr-un singur semnal. Primele releu au fost folosite în circuitele telegrafice pe distanțe lungi ca amplificatoare. Au repetat semnalul venit dintr-un circuit și l-au retransmis pe alt circuit. Releele au fost utilizate pe scară largă în centralele telefonice și în calculatoarele timpurii pentru a efectua operațiuni logice.

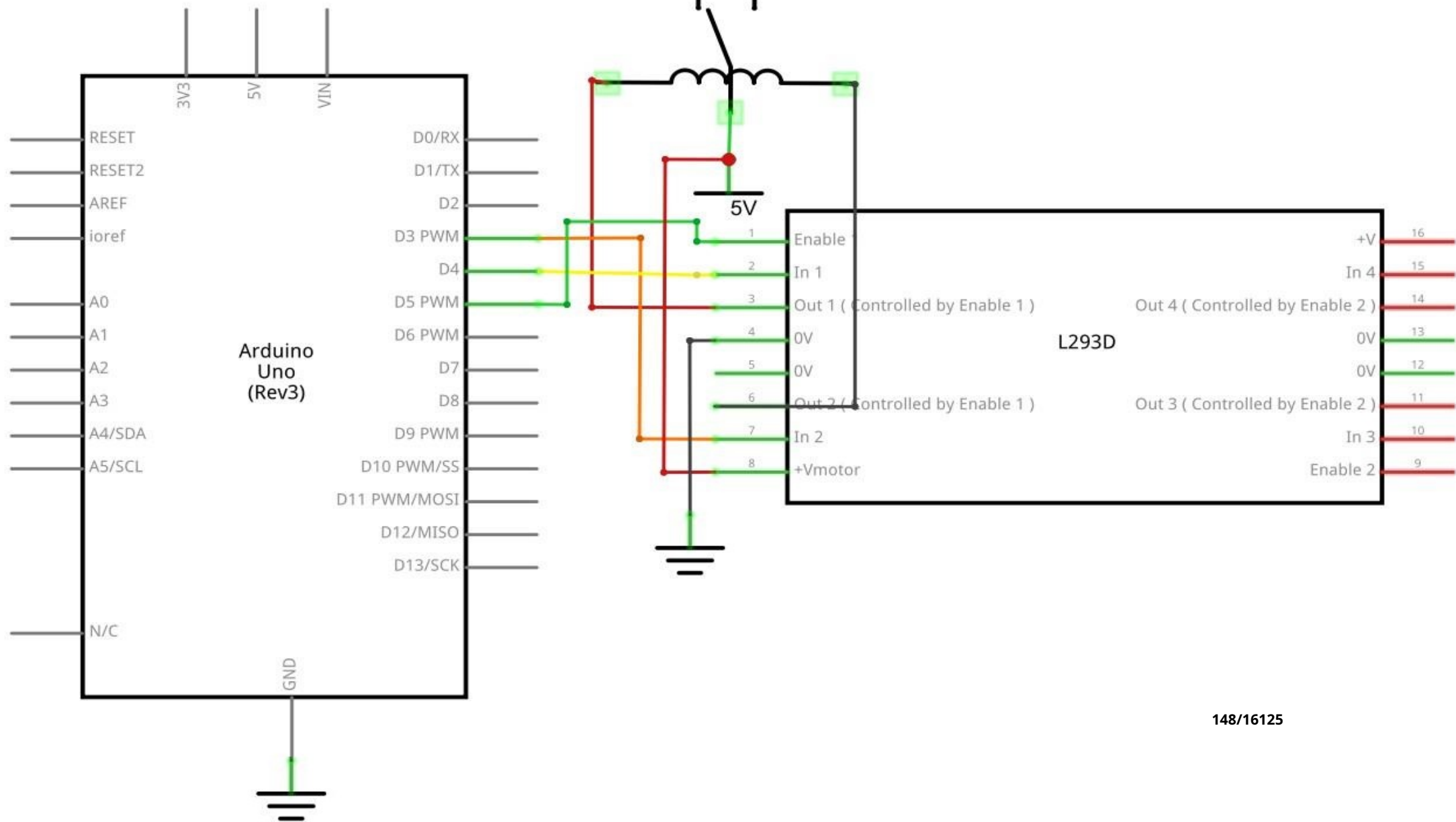
Un tip de releu care poate gestiona puterea mare necesară pentru a controla direct un motor electric sau alte sarcini se numește contactor. Releele cu stare solidă controlează circuitele de putere fără părți mobile, folosind în schimb un dispozitiv semiconductor pentru a efectua comutarea. Releele cu caracteristici de funcționare calibrate și uneori bobine de funcționare multiple sunt utilizate pentru a proteja circuitele electrice de suprasarcină sau defecțiuni. În sistemele electrice moderne, aceste funcții sunt îndeplinite de instrumente digitale numite „releu de protecție”.

Mai jos este schema de a conduce releul cu Arduino.

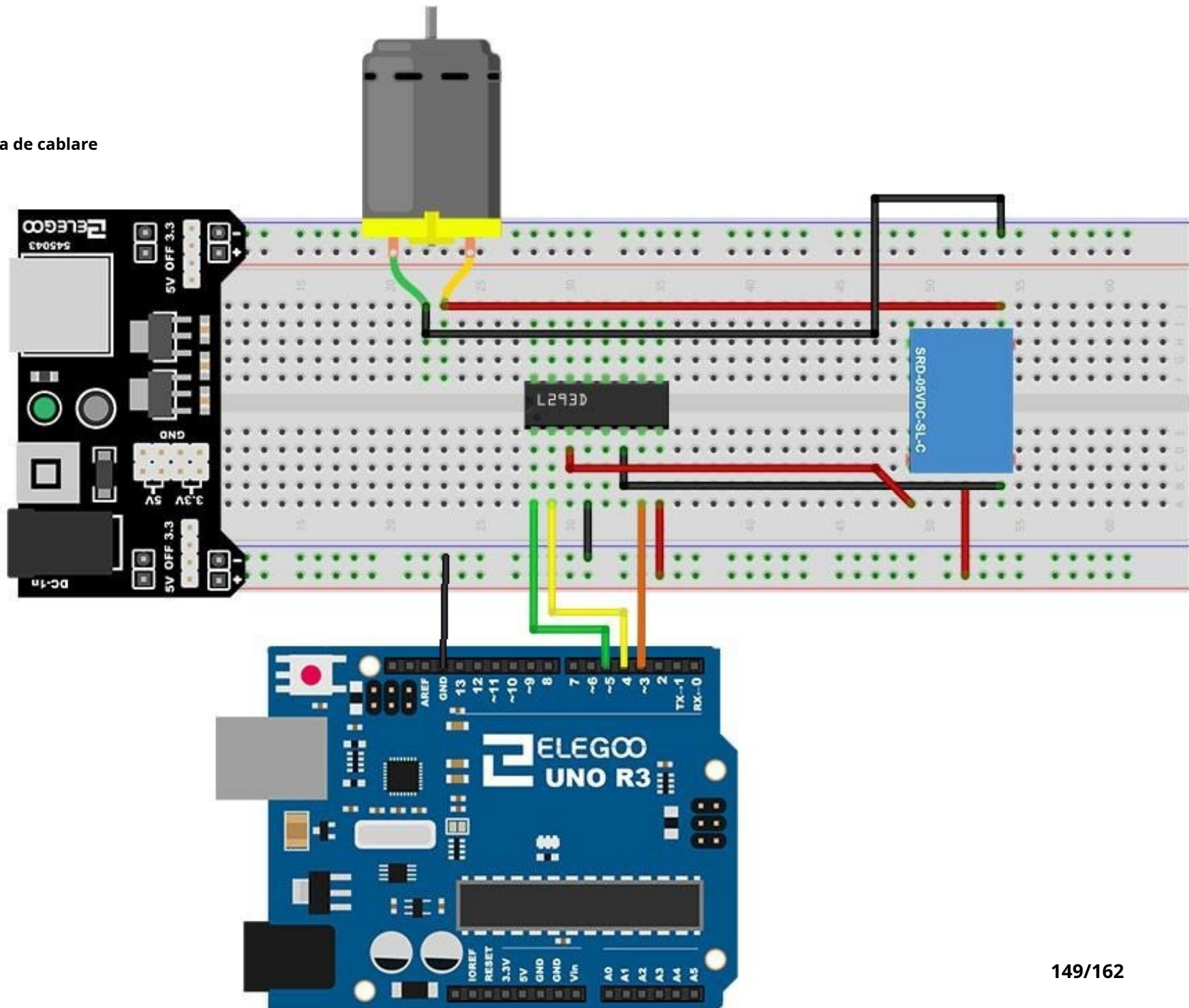
S-ar putea să fii confuz cu privire la modul de inserare a releului în placa de pâine. Ca imaginea de mai jos arată, va trebui să îndoiți ușor unul dintre pinii releului atunci îl poți introduce în placa de pâine.



## Conexiune Schematic



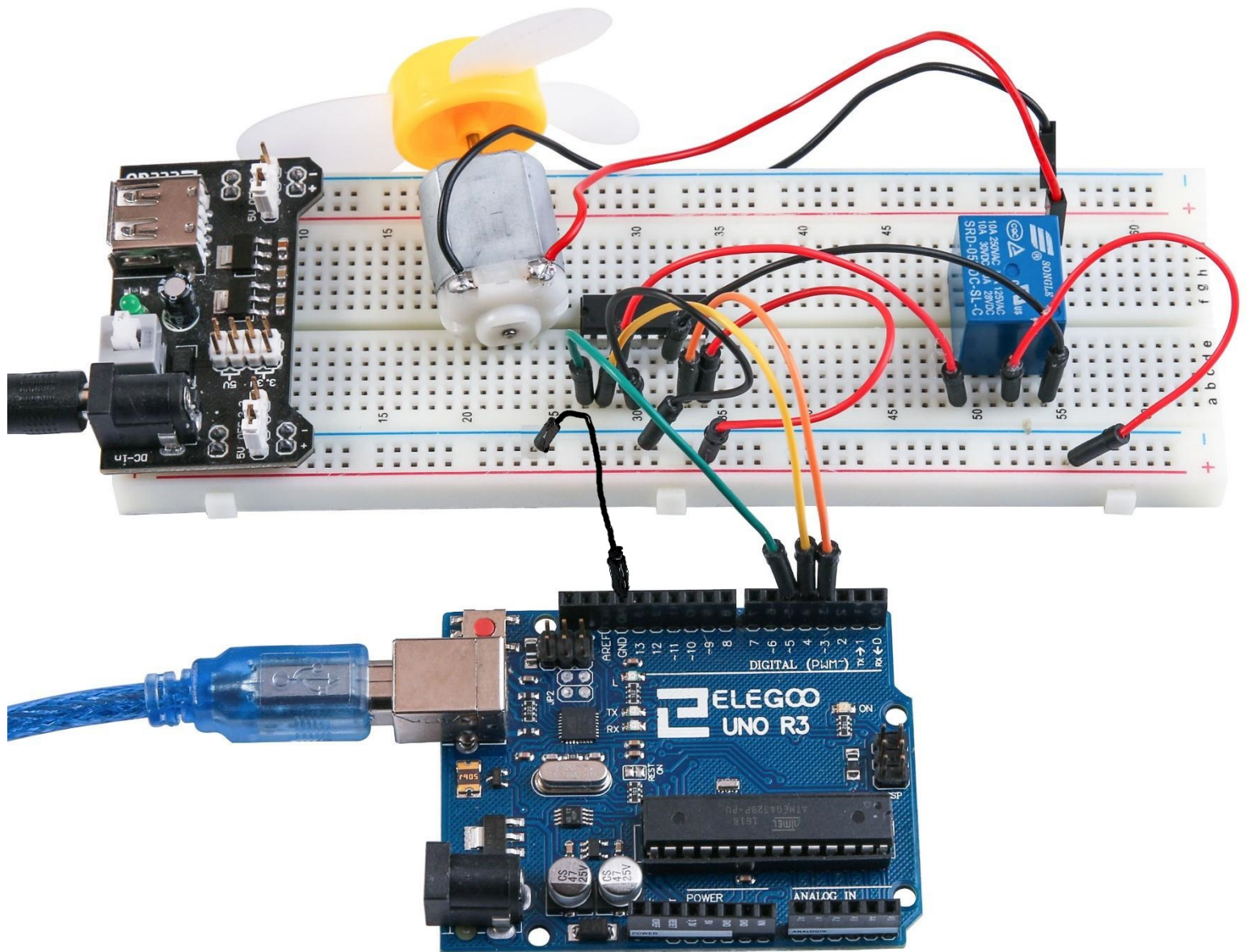
Schema de cablare



## Cod

După cablare, vă rugăm să deschideți programul în folderul de coduri - Lecția 22 Releu și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori. După încărcarea programului, porniți toate comutatoarele de alimentare. Releul va porni cu un sunet de apel. Apoi, motorul se va roti. După o perioadă de timp, releul va fi eliberat, iar motorul se oprește.

## Imagine exemplu



## Lecția 23 Motorul pas cu pas

### Prezentare generală

În această lecție, veți învăța un mod distractiv și ușor de a conduce un motor pas cu pas.

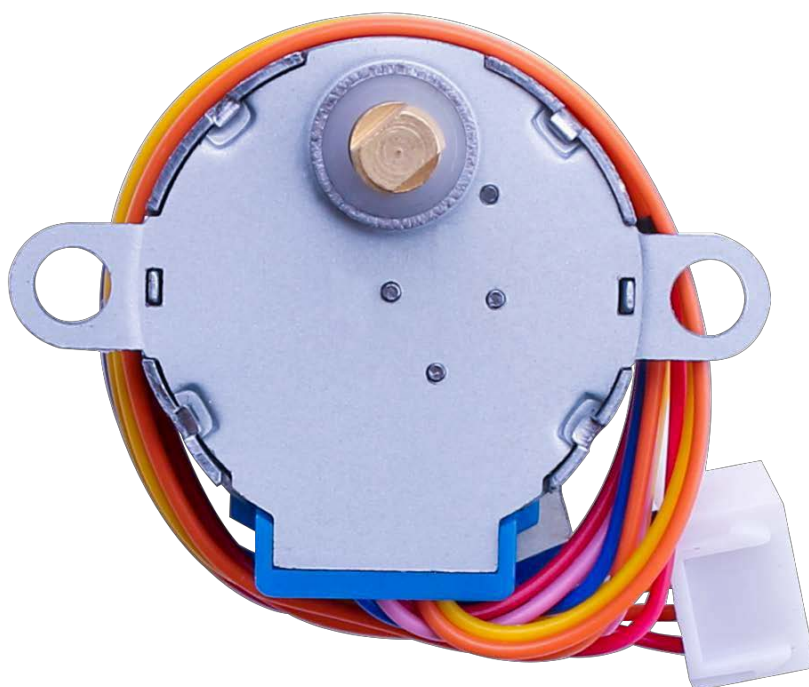
Stepper-ul pe care îl folosim vine cu propria sa placă de driver, ceea ce face ușoară conectarea la UNO.

### Componentă necesară:

- (1) x Elegoo Uno R3
- (1) x 830 de puncte de legătură
- (1) x ULN2003 modul driver motor pas cu pas
- (1) x motor pas cu pas
- (1) x adaptor 9V1A
- (1) x Modul de alimentare
- (6) x fire FM (duPontwires de la femeie la masculin)
- (1) x fir MM (sârmă de legătură de sex masculin la masculin)

### Introducere componente

#### Motor pas cu pas

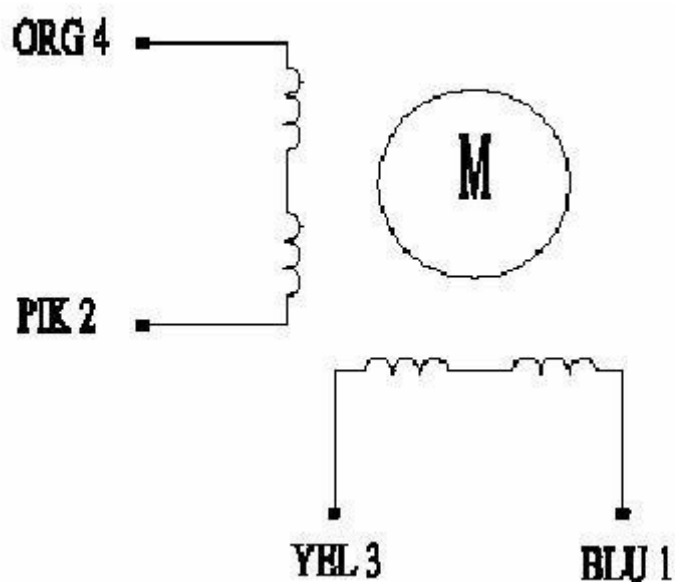


Un motor pas cu pas este un dispozitiv electromecanic care convertește impulsurile electrice în mișcări mecanice discrete. Arborele sau axul unui motor pas cu pas se rotește în trepte discrete atunci când i se aplică impulsuri de comandă electrică în secvența corespunzătoare. Rotația motoarelor are mai multe relații directe cu aceste impulsuri de intrare aplicate. Secvența impulsurilor aplicate este direct legată de direcția de rotație a arborilor motorului. Viteza de rotație a arborilor motorului este direct legată de frecvența impulsurilor de intrare, iar lungimea de rotație este direct legată de numărul de impulsuri de intrare aplicate. Unul dintre cele mai semnificative avantaje ale unui motor pas cu pas este capacitatea sa de a fi controlat cu precizie într-un sistem în buclă deschisă. Controlul în buclă deschisă înseamnă că nu sunt necesare informații de feedback despre poziție. Acest tip de control elimină nevoia de dispozitive costisitoare de detectare și feedback, cum ar fi codificatoarele optice. Poziția dvs. este cunoscută pur și simplu prin urmărirea impulsurilor pasului de intrare.

#### **Motorul pas cu pas 28BYJ-48 Parametri**

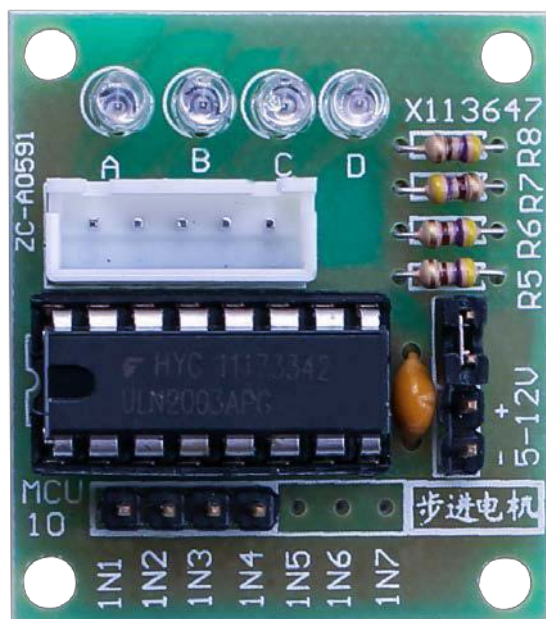
- Model: 28BYJ-48
- Tensiune nominală: 5VDC
- Număr de faze: 4
- Raport de variație a vitezei: 1/64
- Unghi de pas:  $5.625^\circ / 64$
- Frecvență: 100 Hz
- Rezistență DC:  $50\Omega \pm 7\%$  (25°C)
- Frecvența de ralanti la tracțiune: > 600 Hz
- Frecvența de ieșire la ralanti: > 1000 Hz
- Cuplu de tracțiune > 34,3 mN.m (120 Hz)
- Cuplu de autopозиționare > 34,3 mN.m
- Cuplu de frecare: 600-1200 gf.cm
- Cuplu de tragere: 300 gf.cm
- Rezistența izolată >  $10M\Omega$  (500V)
- Putere electrică izolată: 600VAC/1mA/1s
- Grad de izolare: O
- Creșterea temperaturii < 40K (120Hz)
- Zgomot < 35 dB (120 Hz, fără încărcare, 10 cm)

## WIRING DIAGRAM



Motorul pas cu pas bipolar are de obicei patru fire care ies din el. Spre deosebire de stepperele unipolare, stepperele bipolare nu au o conexiune centrală comună. Au în schimb două seturi independente de bobine. Le puteți distinge de stepperele unipolare prin măsurarea rezistenței dintre fire. Ar trebui să găsiți două perechi de fire cu rezistență egală. Dacă aveți cablurile contorului conectate la două fire care nu sunt conectate (adică nu sunt atașate la aceeași bobină), ar trebui să vedeți rezistență infinită (sau lipsă de continuitate).

## Placa driverului ULN2003



## Descriere produs

- o Dimensiune: 42mmx30mm
- o Utilizați cip driver ULN2003, 500mA
- o LED A B C D care indică starea de funcționare a motorului pas cu pas în patru faze. Jack alb
- o este mufa standard cu patru faze pentru motorul pas cu pas.
- o Pinii de alimentare sunt separați
- o Am păstrat pinii de rest ai cipului ULN2003 pentru prototipuri ulterioare.

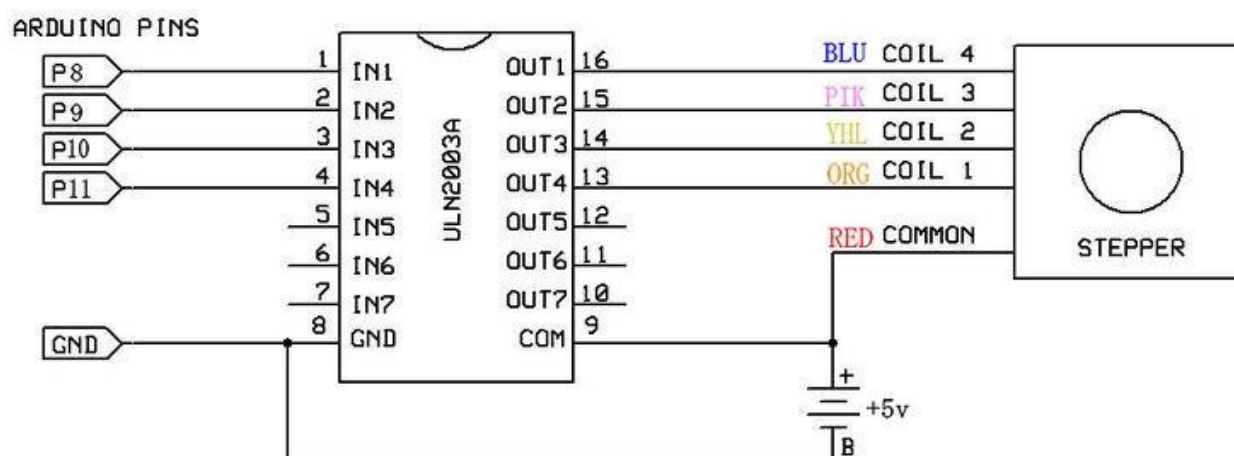
Cea mai simplă modalitate de a conecta un stepper unipolar cu Arduino este de a folosi un breakout pentru cip de matrice de tranzistori ULN2003A. ULN2003A conține șapte drivere pentru tranzistori Darlington și este oarecum ca și cum ar avea șapte tranzistoare TIP120 într-un singur pachet.

ULN2003A poate trece până la 500 mA pe canal și are o cădere de tensiune internă de aproximativ 1V când este pornit. Conține, de asemenea, diode de prindere interne pentru a disipa vârfurile de tensiune atunci când conduc sarcini inductive. Pentru a controla stepper-ul, aplicați tensiune fiecărei bobine într-o anumită secvență.

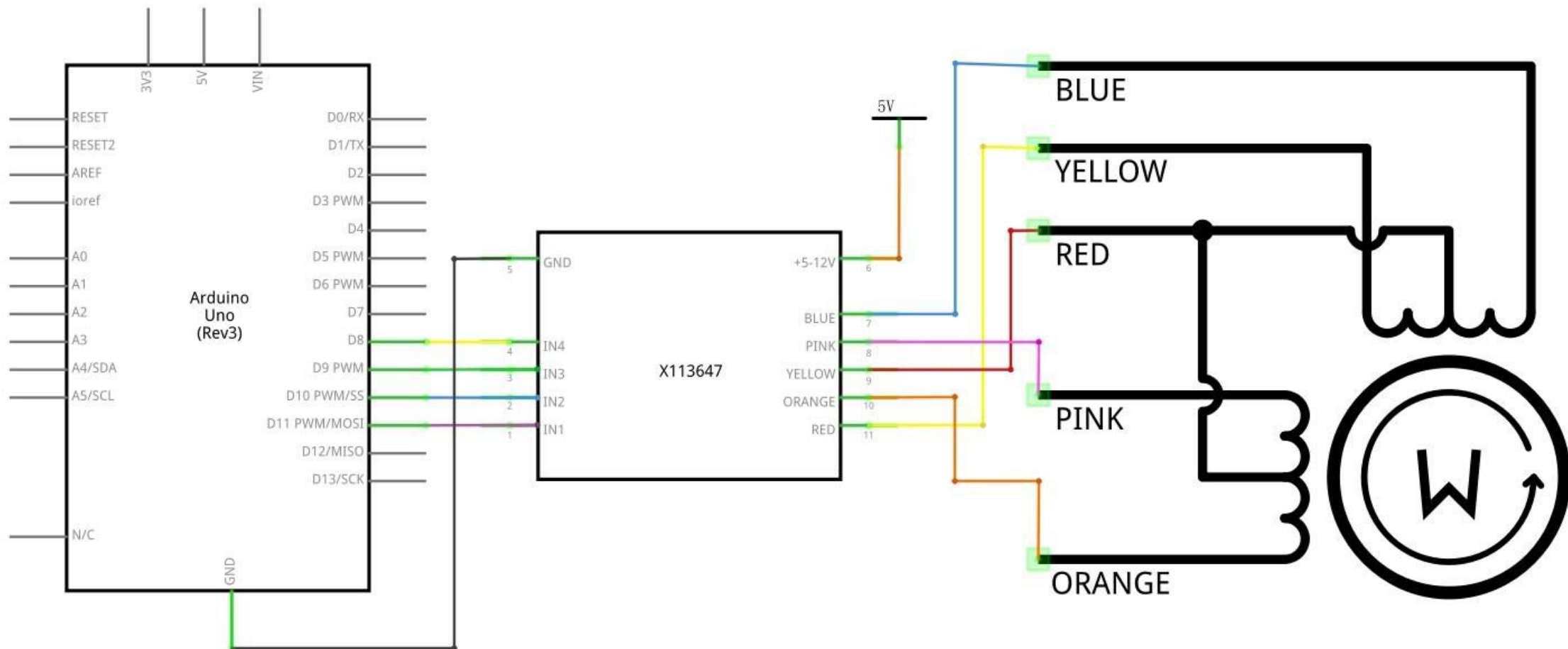
Secvența ar decurge astfel:

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

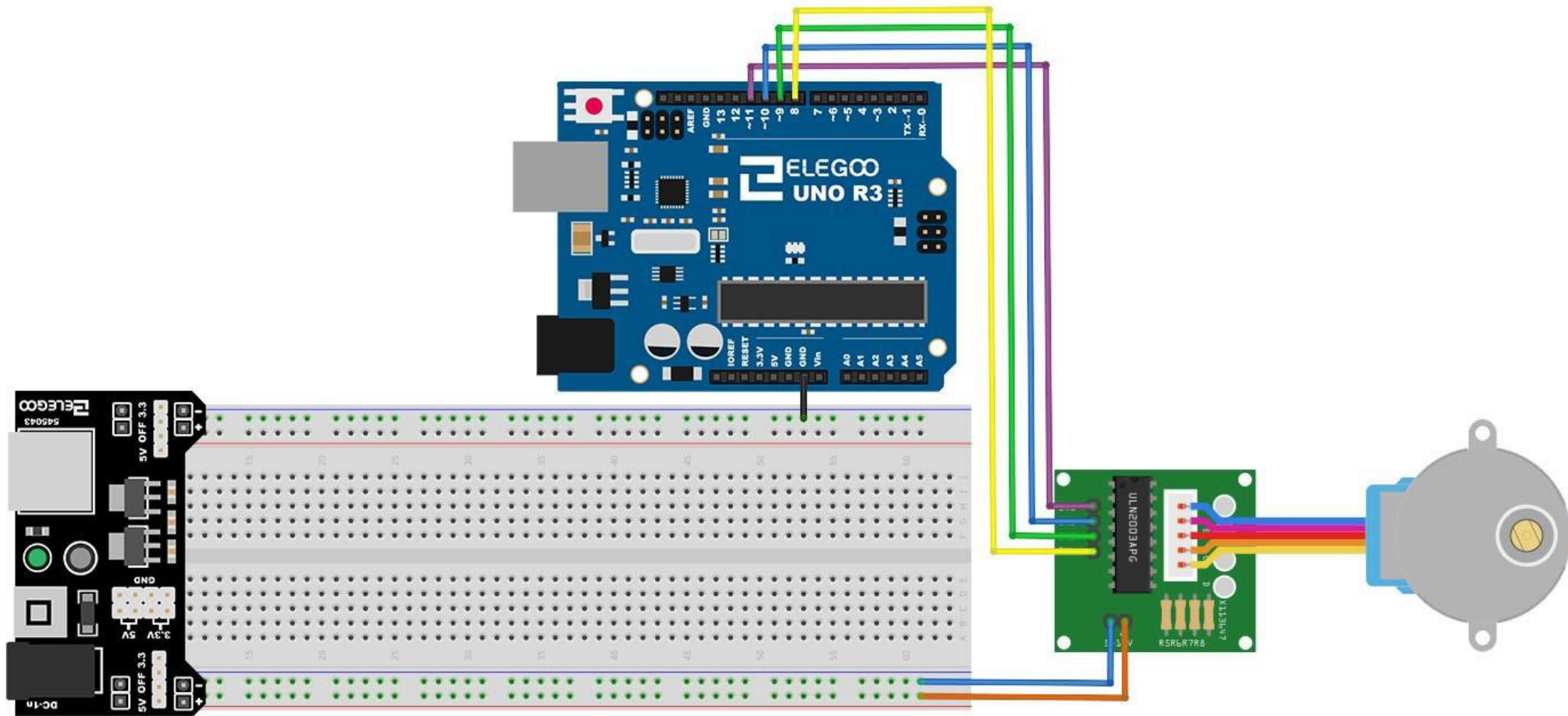
Iată scheme care arată cum să interfați un motor pas cu pas unipolar la patru pini de controler folosind un ULN2003A și care arată cum să interfațezi folosind patru com.



## Conexiune Schematic



### Schema de cablare



Folosim 4 pini pentru a controla Stepper-ul.

Pinul 8-11 controlează motorul pas cu pas.

Conectăm pământul de la UNO la motorul pas cu pas.

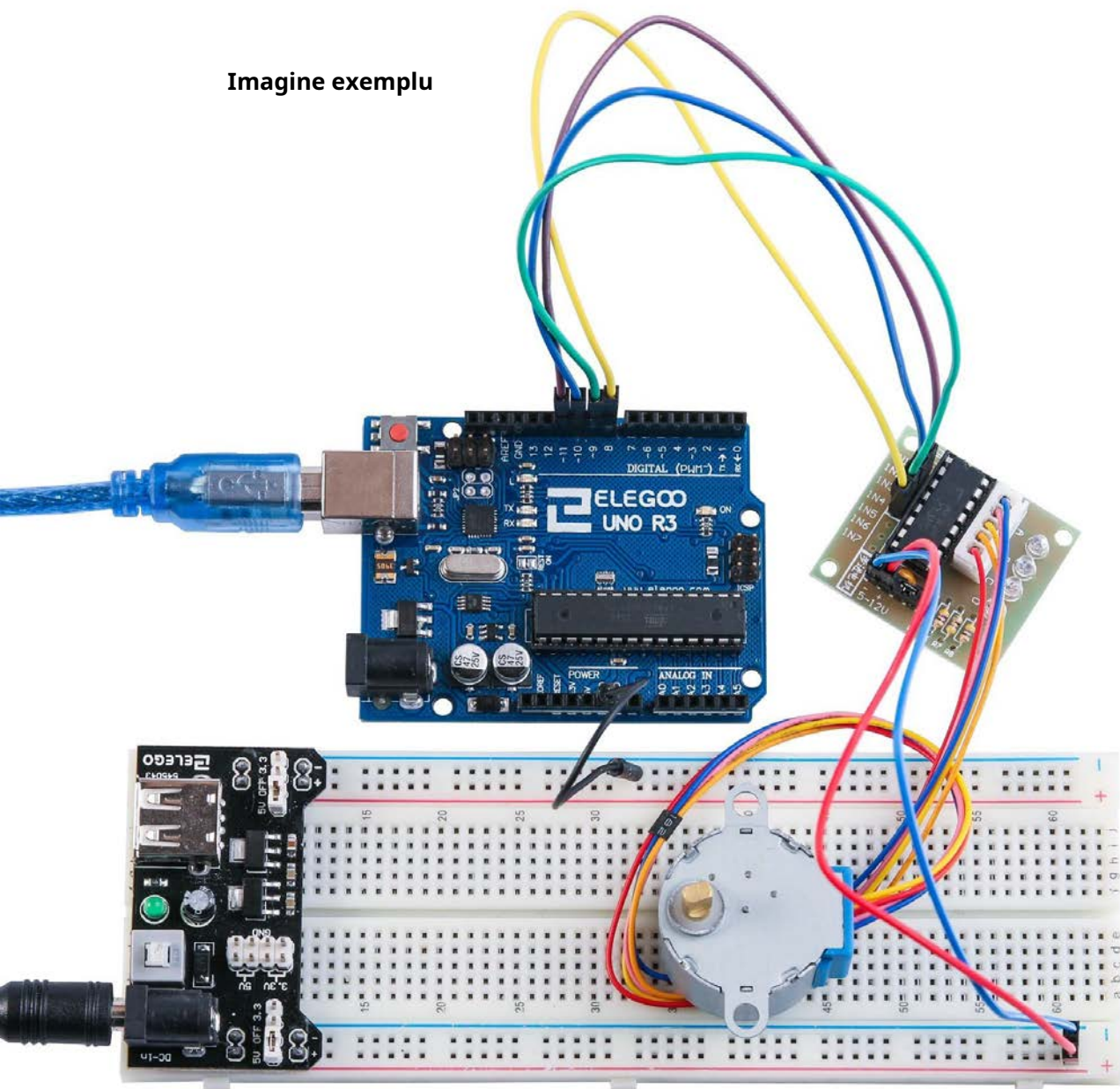
## Cod

După cablare, deschideți programul în folderul de coduri - Lecția 23 Motor pas cu pas și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < Stepper > sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa.

Pentru detalii despre încărcarea fișierului bibliotecă, consultați Lecția 1.

## Imagine exemplu



## Lecția 24 Controlul motorului pas cu telecomandă

### Prezentare generală

În această lecție, veți învăța o modalitate distractivă și ușoară de a controla un motor pas cu pas de la distanță folosind o telecomandă IR.

Stepper-ul pe care îl folosim vine cu propria sa placă de driver, ceea ce face ușoară conectarea la UNO.

Deoarece nu dorim să conducem motorul direct de la UNO, vom folosi o sursă mică de alimentare pentru placa de breadboard, ieftină, care se conectează direct la placa noastră și o alimentează cu o sursă de alimentare de 9V 1Amp.

Senzorul IR este conectat direct la UNO, deoarece aproape că nu folosește energie.

### Componentă necesară:

(1) x Elegoo Uno R3

(1) x 830 de puncte de legătură

(1) x modul receptor IR

(1) x telecomandă IR

(1) x ULN2003 modul driver motor pas cu pas

(1) x motor pas cu pas

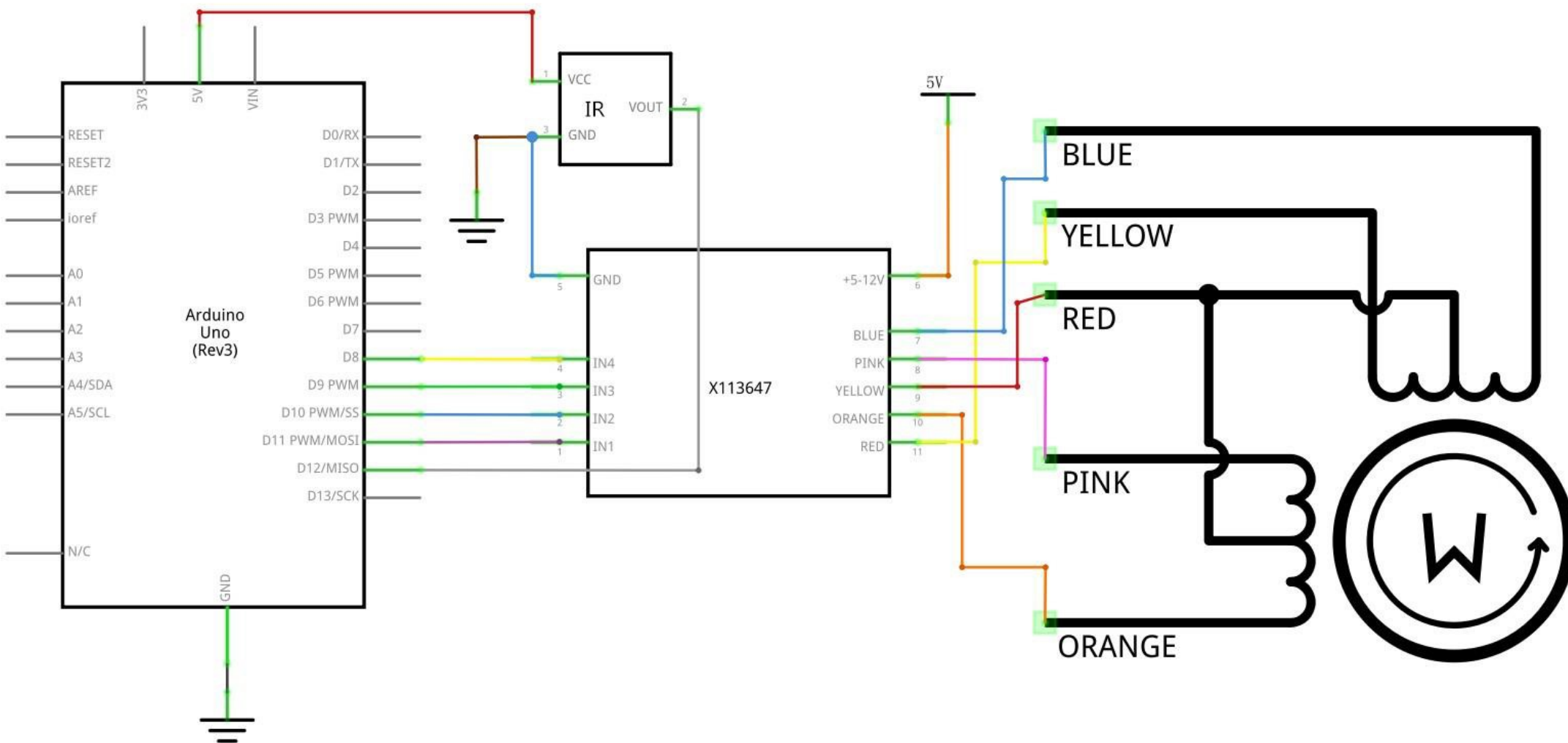
(1) x Modul de alimentare

(1) x adaptor 9V1A

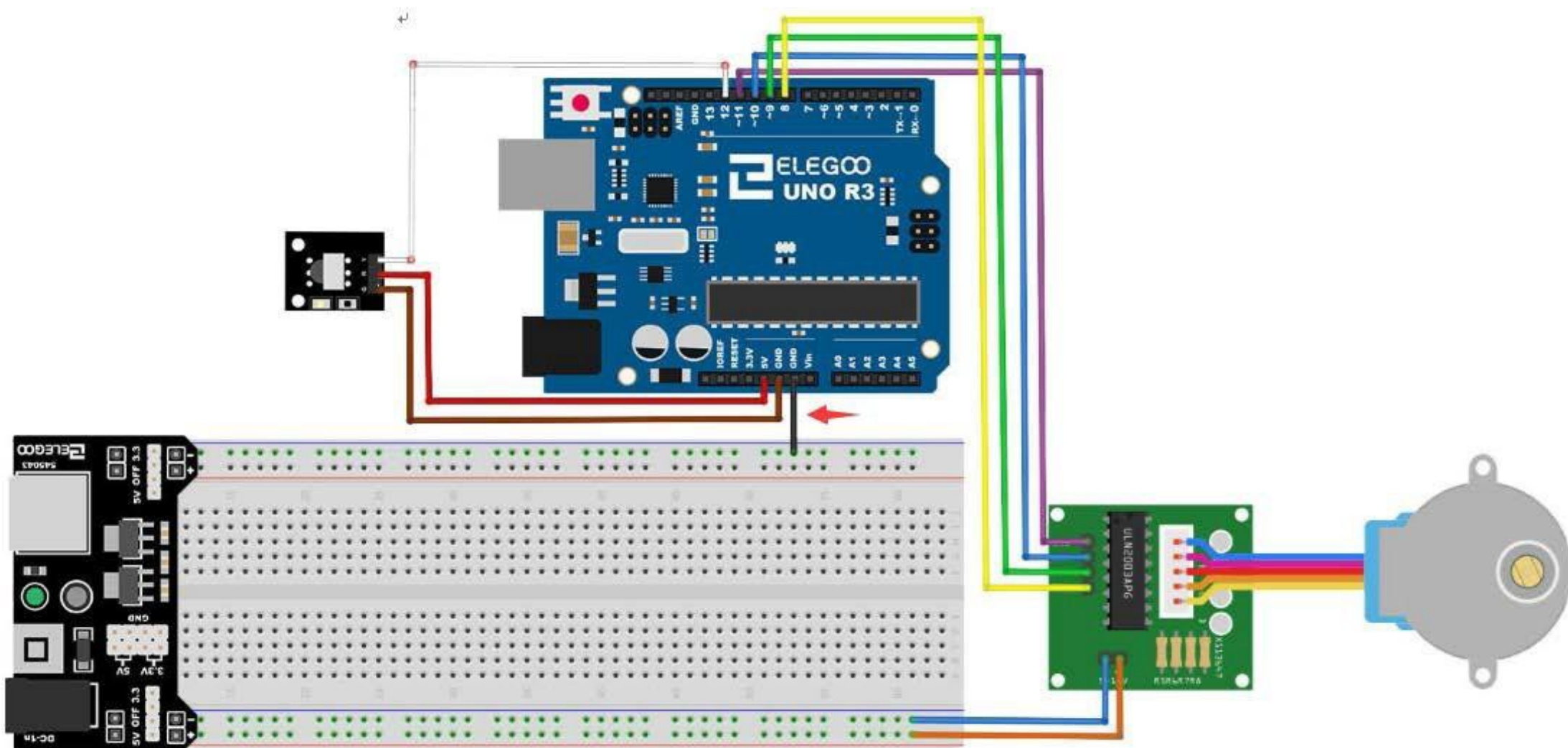
(9) x fire FM (duPontwires de la femeie la masculin)

(1) x fir MM (sârmă de legătură de sex masculin la masculin)

## Conexiune Schematic



# Schema de cablare



Folosim 4 pini pentru a controla Stepper-ul și 1 pin pentru IRsensor.

Pinii 8-11 controlează motorul pas cu pas și pinul 12 primește informații IR.

Conectăm 5V și masă de la UNO la senzor. Ca măsură de precauție, utilizați o sursă de alimentare pentru a alimenta motorul pas cu pas, deoarece poate folosi mai multă putere și nu dorim să deteriorăm sursa de alimentare a UNO.

## Cod

După cablare, deschideți programul în folderul de coduri - Lecția 24 Controlul motorului pas cu telecomandă și faceți clic pe ÎNCĂRCARE pentru a încărca programul. Consultați Lecția 2 pentru detalii despre încărcarea programului dacă există erori.

Înainte de a putea rula acest lucru, asigurați-vă că ați instalat biblioteca < IRremote > < Stepper > sau reinstalați-o, dacă este necesar. În caz contrar, codul tău nu va funcționa. Pentru detalii despre încărcarea fișierului bibliotecă, consultați Lecția 1.

Codul recunoaște doar 2 valori de la telecomanda IR: VOL+ și VOL-. Când VOL+ este apăsat pe telecomandă, motorul va face o rotație completă în sensul acelor de ceasornic. VOL- va face o rotație completă în sens invers acelor de ceasornic.

Imagine exemplu

