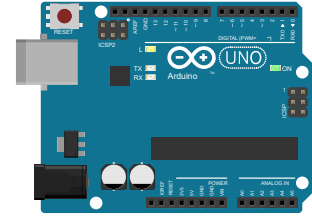
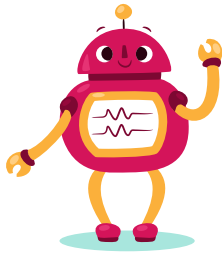


CUPRINS

INTRODUCERE	4
Despre Arduino	4
De ce să înveți Arduino ?	4
Placa Arduino - prezentare	5
Arduino UNO / NANO - principalele proprietăți	6
Mediul Arduino	9
Structura unui program Arduino	13
Tipuri de semnale	15
Breadboard-ul	19
ALGORITMI	20
Noțiunea de algoritm	20
Reprezentarea algoritmilor	22
Principiile programării structurate	25
Structura avansată a programelor Arduino	28
Variabile	30
Instrucțiuni	31
Vectori. String-uri. Matrici	34
Funcții	35
Declararea și apelarea unei funcții	35
Tipuri de funcții	35
Parametri	35
Variabile utilizate în funcții	35
PROIECTE	36
Proiect 1 - LED	36
Proiect 2 - Semafor	38
Proiect 3 - Tranzistor	40
Proiect 4 - Buton	42
Proiect 5 - LED RGB	44

Proiect 6 - Potențiomtru	46
Proiect 7 - LED luminos	48
Proiect 8 - Fotorezistorul	50
Proiect 9 - Telecomanda	52
Proiect 10 - Senzor de prezență umană	55
Proiect 11 - Senzor de distanță	57
Proiect 12 - Pianul digital	60
Proiect 13 - Termistor	62
Proiect 14 - Ecran LCD	64
Proiect 15 - Joystick	66
Proiect 16 - Senzor magnetic	68
Proiect 17 - Termometrul	70
Proiect 18 - Senzor de alcool	73
Proiect 19 - Plantă inteligentă	76
Proiect 20 - Tomberon inteligent	78
BIBLIOGRAFIE	80
Surse Bibliografice	80



Prefață

Lucrarea de față s-a născut din dorința de a facilita inițierea în domeniul roboticii a celor aflați la început de drum, fără experiență în programare.

Structurată pe trei capitole, prezenta carte nu își propune să fie una exhaustivă, ci mai degrabă un caiet de lucru pe baza căruia să se poată realiza imediat câteva aplicații relativ simple cu Arduino, înțelegând totodată și mecanismele care stau la baza realizării circuitelor aferente.

Primul capitol – intitulat Introducere în Arduino – prezintă lumea Arduino, atât din perspectivă hardware (placa de dezvoltare), cât și software (mediul de dezvoltare).

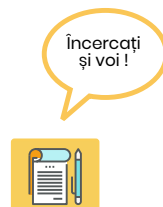
Dar, înainte de a lucra efectiv cu placa Arduino, este necesar să se înțeleagă conceptele de bază ale programării. În acest scop am conceput capitolul al doilea – Algoritmi – cu ajutorul căruia veți înțelege noțiunea de algoritm, veți învăța modalitățile de reprezentare ale algoritmilor, dar și instrucțiunile și obiectele specifice Arduino.

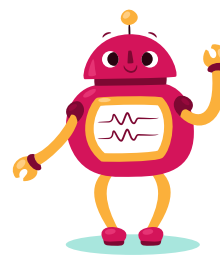
Proiectele prezentate în cel de-al treilea capitol denumit Primele proiecte cu Arduino au fost realizate în totalitate de către membrii clubului de robotică RoboCODE în cadrul atelierelor desfășurate în anii școlari precedenți. Cu această dorință în minte a fost scrisă această carte, dacă noi am reușit să realizăm aceste mici aplicații suntem convinși că și voi o puteți face la rândul vostru! Credem totodată că satisfacția finalizării unor astfel de proiecte este de neegalat!

Mai mult, cu acest prilej veți putea constata cât de bine vi se potrivește acest domeniu, vă veți putea orienta mai bine ce să faceți mai departe în viață, electrotehnică sau programare, sau, de ce nu, amândouă așa cum propunem noi prin studiul Arduino!

Ne puteți trimite feedback, sugestii sau recomandări pentru extinderea sau îmbunătățirea activităților prezentate la adresele de email:

- viorel_muscas@yahoo.com
- robnad2000@gmail.com





Despre Arduino

“Arduino” se referă în principiu la:

- placa de dezvoltare care are în componența sa un microcontroller și care astfel poate fi programată, respectiv
- soft-ul folosit pentru a scrie codul care să controleze placa.

De asemenea, Arduino este și numele companiei, dar și a comunității care gestionează dezvoltarea plăcilor și a mediului de programare.

Cu ajutorul acestui curs, veți descoperi Arduino atât din punct de vedere hardware cât și software.

De ce să înveți Arduino ?

- Pentru că în zilele noastre suntem înconjurați peste tot de dispozitive inteligente care ne ajută în viața cotidiană, deci ar fi bine să înțelegem cum funcționează acestea;
- Pentru că poți realiza proiecte educative care să îți pună mintea și creativitatea la contribuție pentru a te dezvolta pe tine însuși;
- Pentru a afla ce te pasionează – programarea și/sau electronica;
- Pentru că este cea mai populară placă de dezvoltare din lume;
- Pentru că folosind Arduino practic nu există limite în privința complexității proiectelor ce se pot realiza.



ȘTIAȚI CĂ?

Denumirea de **Arduino** provine de la numele localului din Italia unde o parte dintre fondatori obișnuiau să se întâlnească.



<https://www.arduino.cc/>



<https://youtu.be/1ENiVwk8idM>



Placa Arduino - prezentare

Arduino reprezintă o familie de plăci ce se bazează pe microcontrolere și este destinată pasionaților de electronică, mecanică și programare, într-un cuvânt robotică. Arduino și-a făcut prima apariție în 2005, iar de atunci s-a dezvoltat din ce în ce mai mult.

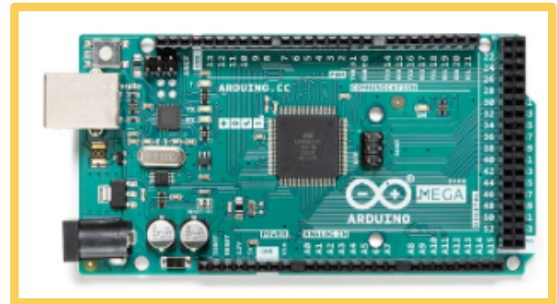
Arduino nu este singura familie de plăci de dezvoltare, dar este cel mai cunoscut brand.

Plăcile din seria Arduino reprezintă un bun punct de plecare în studiul roboticii, astfel că acest manual se va axa asupra a două dintre plăcile Arduino și anume **Arduino UNO** - cel mai cunoscut model din această serie și **Arduino NANO**.

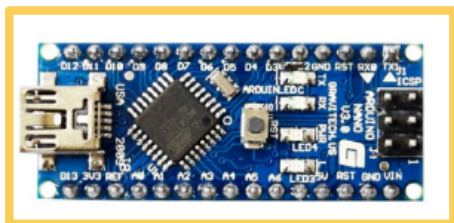
Actualmente există nu mai puțin de 17 variante de plăci Arduino, iar cele mai cunoscute sunt prezentate mai jos.



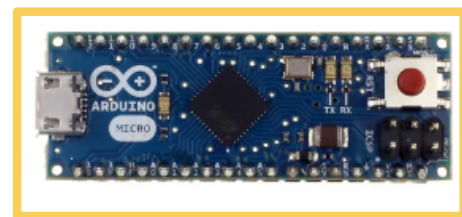
Arduino UNO - cea mai populară placă



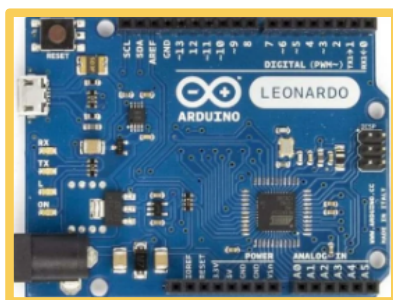
Arduino MEGA - cea mai puternică



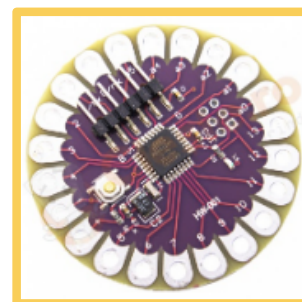
Arduino NANO - cea mai mică placă, dar la fel de capabilă ca UNO



Arduino MICRO - printre cele mai mici



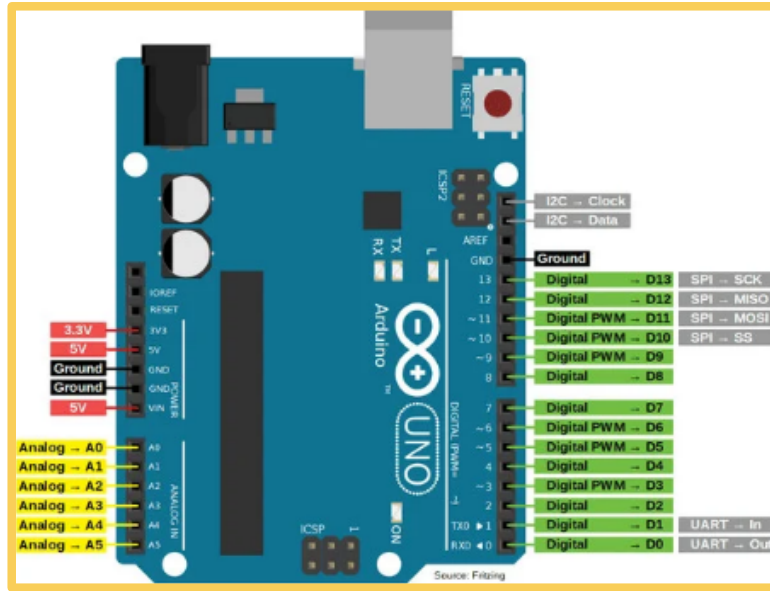
Arduino LEONARDO - mai performantă decât UNO



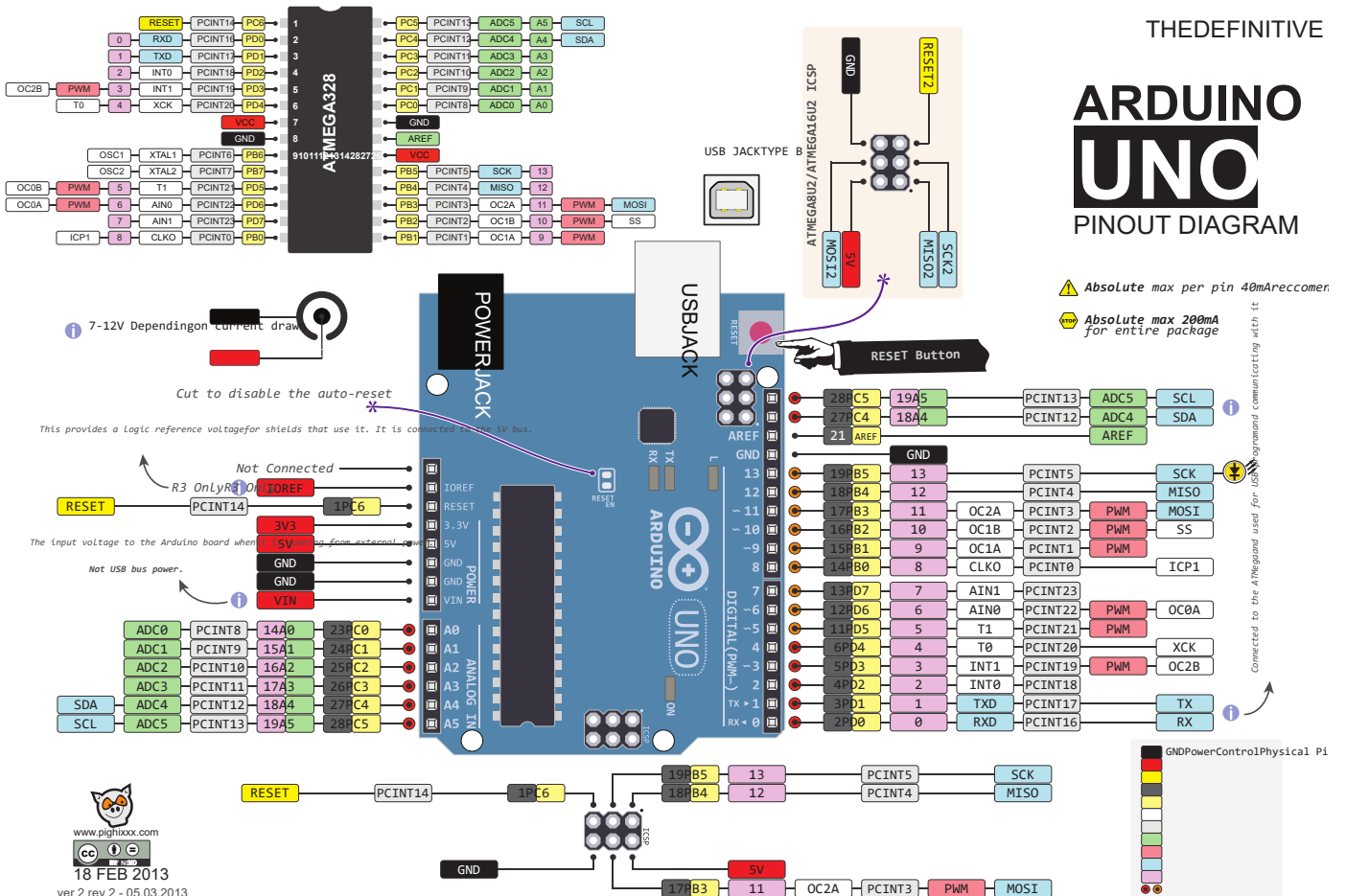
Arduino LILYPAD - pentru textile

Arduino UNO / NANO - principalele proprietăți

Ambele plăci - **Arduino UNO** și **Arduino NANO** - sunt plăci de dezvoltare open-source realizate pe baza microcontrolerului **ATmega328P**. Acesta din urmă prezintă un procesor AVR cu o arhitectură de tip RISC.



Placa de dezvoltare Arduino UNO



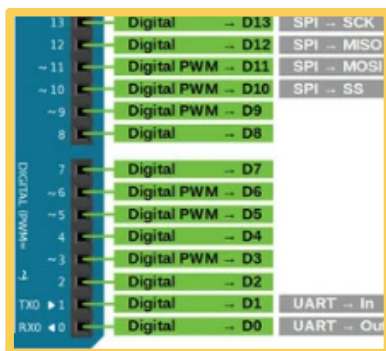
ARDUINO UNO PINOUT DIAGRAM

- ⚠ Absolute max per pin 40mA recommended
- ⚡ Absolute max 200mA for entire package

Din punct de vedere al microcontrolerului avem următoarele specificații:

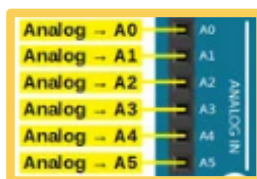
- o viteza de 20 MIPS
- o 32 kb de spațiu de stocare pentru program din care 2kb sunt ocupați de bootloader
- o 2 kb de memorie echivalentă RAM (de lucru)
- o 1 kb de spațiu de stocare EEPROM

Din punct de vedere al specificațiilor de conectare avem următorii pini:



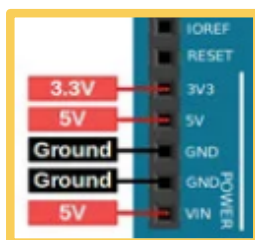
Pinii digitali

- D0-D13: 14 pini digitali de intrare/ieșire, din care:
 - 2 pini (D0 și D1) sunt RX și TX
 - 12 sunt digitali, din care:
 - 6 pini (D3, D5, D6, D9, D10, D11), cei cu semnul “~”, sunt conectabili la module PWM
 - 4 pini (D10, D11, D12, D13) conectabili la SPI / ICSP



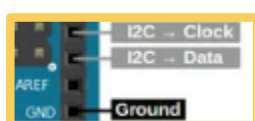
Pinii analogici

- A0-A5: 6 pini analogici de intrare, din care 2 pini sunt SDA (A4) și SCL (A5)



Pinii de alimentare

- IOREF: pin folosit ca referință de intrare/ieșire; oferă referința de tensiune cu care funcționează microcontrolerul
- RESET: pin care resetează Arduino
- 2 pini de alimentare (unul de 5V și celălalt de 3,3V)
- GND: doi pini de ground
- VIN: un port de alimentare a plăcii Arduino folosind o sursă de alimentare externă; tensiunea trebuie să se încadreze între 7-12V.



Alți pini

- I2C Clock (SCL): pin folosit pentru a sincroniza transferul de date între dispozitivele I2C
- I2C Data (SDA): pin utilizat pentru transferul efectiv de date între dispozitive
- GND: un pin de ground

De asemenea, mai regăsim:

- 1 mufă USB-B
- 1 mufă de alimentare DC



Mic dicționar de termeni!

AVR = este o familie de microcontrollere care au la bază un procesor RISC cu o arhitectură Harvard, adică unitatea centrală de procesare are memoria de program separată de memoria de date.

RISC (Reduced Instruction Set Computer) = calculator cu set de instrucțiuni reduse, este o arhitectură de microprocesor/calculator cu un set de comenzi simple și rapide, în care viteza crește datorită simplificării instrucțiunilor.

MIPS (milioane de instrucțiuni pe secundă) - unitate de măsură a vitezei procesorului.

Kb (KiloByte) - unitate de măsură a informației.

RAM (Random Access Memory) - memorie cu acces aleatoriu, se șterge la întreruperea alimentării.

EEPROM (Electrically Erasable Programmable Read-only Memory) - memorie non-volatilă (nu se șterge), pentru stocarea de informații și după oprirea alimentării.

RX și TX - folosiți pentru a recepționa (RX) și a transmite (TX) date în comunicații cu alte module sau plăci Arduino.

PWM (Pulse-Width Modulation) - pini digitali speciali care sunt capabili să genereze semnale omonime. PWM este o tehnică utilizată pentru controlul intensității sau a vitezei unui dispozitiv sau a unei componente.

SPI (Serial Peripheral Interface) este un protocol de date seriale utilizat de microcontrollere pentru a comunica cu unul sau mai multe dispozitive externe într-o conexiune de tip magistrală.

ICSP (In-Circuit Serial Programming) - numele provine din anteturile de programare în sistem (ISP). Acești pini permit utilizatorului să programeze firmware-ul plăcilor Arduino.

SCK (Serial Clock) = un semnal de ceas generat de dispozitivul master pentru sincronizarea transmisiei de date.

MISO (Master In Slave Out) = o linie pentru trimiterea datelor către dispozitivul master.

MOSI (Master Out Slave In) = linia master pentru trimiterea datelor către dispozitivele periferice.

SS (Slave Select) determină cu ce dispozitiv comunică masterul.

GND (Ground) - pinii GND sunt utilizați pentru a închide circuitul electric și pentru a oferi un nivel comun de referință logică în întregul circuit.

VIN (Voltage Input) = tensiune de intrare.

I2C - este un protocol de comunicație.

SCL (Serial Clock) - este linia de ceas proiectată pentru a sincroniza transferurile de date.

SDA (Serial Data) - este linia utilizată pentru transmiterea datelor.

USB-B (Universal Serial Bus - B) = tip de port USB.

DC (Direct Current) = curent continuu.

Mediul Arduino

Instalarea Arduino IDE

Pentru crearea și editarea codului ce trebuie încărcat pe placa Arduino, creatorii platformei au realizat și un **IDE** (**I**ntegrated **D**evelopment **E**nvironment - mediu de dezvoltare integrat) care poate satisface și cele mai complexe cerințe. În plus acesta este gratis, fiind disponibil pe site-ul dezvoltatorilor.

Pentru instalare urmați pașii de mai jos.

Pasul 1:

Descărcați **Arduino IDE** corespunzător sistemului de operare de la adresa: <https://www.arduino.cc/en/Main/Software>



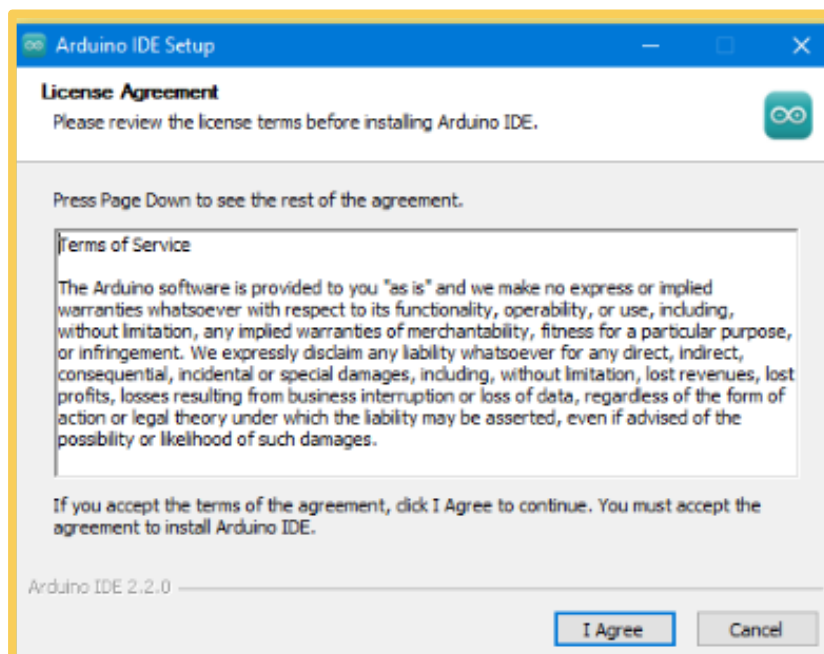
<https://www.arduino.cc/en/Main/Software>



Pagina de download – Arduino IDE

Pasul 2:

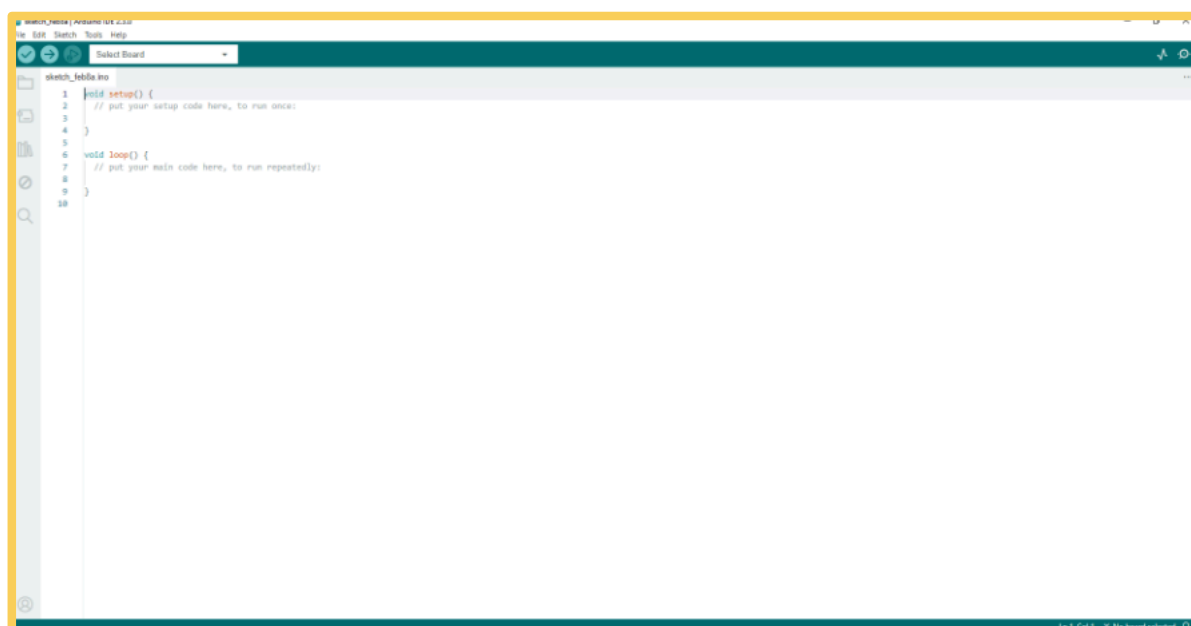
Deschideți fișierul descărcat și instalați aplicația.



Programul de instalare - Arduino IDE

Pasul 3:

După finalizarea instalării, pornim Arduino IDE și se va deschide fereastra de mai jos.



Interfața Arduino IDE

Tipuri de semnale

Digital vs. Analogic

Trăim într-o lume analogică, există o multitudine de nuanțe de culori, o multitudine de tonalități pe care le putem auzi și o multitudine de mirosuri pe care le putem percepe. Sensorii noștri biologici înregistrează constant date pe care creierul le combină și le transformă în informații inteligibile și utile.

Semnalele și obiectele digitale se află într-o lume discretă sau finită, adică pot lua un număr limitat de valori. De exemplu, doar două valori (0 și 1), 255 sau chiar 4294967296, dar nu o infinitate de valori.



Ceas digital vs. Ceas analogic

În lumea electronicii întâlnim atât semnale analogice, cât și digitale, la intrare și la ieșire.

Tehnologia analogică ("analogic" înseamnă "identic") presupune prelucrarea semnalelor electrice așa cum sunt recepționate de la diverși senzori sau emițători.

Un televizor analogic recepționează semnalul TV modulat transmis prin cablu coaxial sau prin unde radio. Acesta este demodulat și transformat în semnale electrice ce sunt trimise către tubul catodic, respectiv difuzoare pentru a deveni imagini, respectiv sunet.

În echipamentele digitale, semnalul util este reprezentat prin valorile binare 0 sau 1 care sunt interpretate fizic prin două nivele de tensiune. Un exemplu este acela în care, pentru valoarea 0, valoarea tensiunii este cuprinsă între 0V și 1V, iar pentru valoarea 1, avem o tensiune cuprinsă între 4V și 5V.

Un semnal digital este obținut prin eșantionarea (digitizarea) unui semnal analogic provenit de la diverși senzori sau emițători.

Un sistem digital de televiziune, eșantionează un semnal analogic transformându-l în valori binare: 0 sau 1. Acesta este comprimat și codificat pentru a fi transmis pe un canal de date. La recepție, semnalul digital este decodificat și decomprimat, astfel încât să poată fi redat.

Un mouse transformă semnalul analogic provenit de la mișcarea plană a acestuia pe două direcții – X și Y, în semnale digitale pe două canale X și Y. Cursorul de pe ecran se va mișca digital conform numărului de impulsuri primite pe cele două canale.



ȘTIAȚI CĂ? Un ecran tactil nu este altceva decât o matrice de senzori de presiune! Semnalul analogic produs de acești senzori este transformat în semnal digital.

INTRODUCERE

Un modem (însăși denumirea provine de la modulare-demodulare) este un dispozitiv ce permite conectarea la Internet, fiind capabil să convertească semnalele digitale în analogice, un proces numit „modulare” și este, de asemenea, capabil să convertească semnale analogice în digitale, al căror proces se numește „demodulare”, de aici și denumirea sa.

Echipamentele digitale prezintă o serie de avantaje:

- sunt foarte puțin afectate de condițiile de mediu (temperatură, umiditate, etc.);
- permit reducerea volumului echipamentelor ceea ce presupune și un consum redus de energie;
- permit o rată de transfer a informației mult mai mare decât cele analogice.

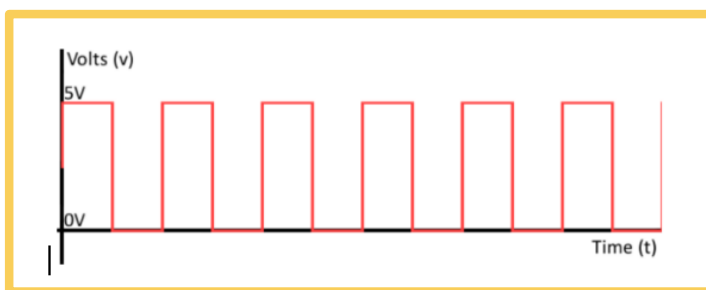
Semnale pe placa Arduino

Placa Arduino este capabilă să scrie/citească semnale atât digitale cât și analogice - folosind tehnica **PWM** (*pulse-width-modulation*) și convertoarele analog în digital.

Pinii digitali sunt utilizați pentru a citi valori ale unor mărimi și pentru a comanda dispozitive externe (led-uri, motoare). Pinii analogi sunt utilizați pentru a citi valori ale unor mărimi care variază continuu (temperaturi, distanța până la un obiect, etc.).

Semnalul digital

Semnalul digital arată astfel:



Semnalul digital

Un astfel de semnal este ușor de utilizat datorită faptului că el poate avea doar 2 valori, LOW(0) și HIGH(1). Pentru a scrie sau citi astfel de valori în Arduino folosim următoarele secvențe: “**digitalWrite**(pin, valoare)” pentru scriere și “**digitalRead**(pin)” pentru citire.

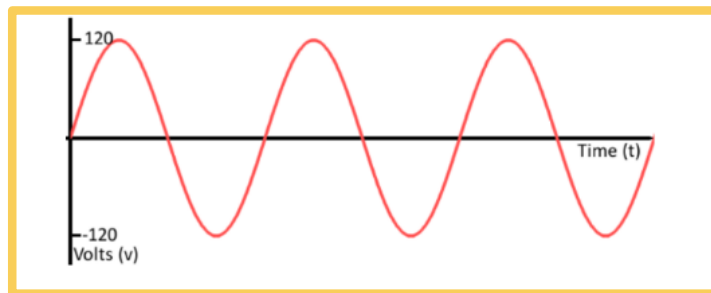


Pe placa Arduino toți pinii digitali au inițiala “D” (Ex.: D1, D2, ..., D13).

Digital	
Scriere	Citire
2 valori: 0 și 1 sau altfel spus LOW(0) și HIGH(1)	Rezultat binar (0 sau 1)
Este posibilă pe toți pinii digitali	Este posibilă pe toți pinii digitali
Instrucțiune: <code>digitalWrite(pin, LOW / HIGH);</code>	Instrucțiune: <code>digitalRead(pin);</code>

Semnalul analogic

În schimb, semnalul analogic poate arată astfel:



Semnalul analogic

Pentru scriere vom folosi sintaxa “**analogWrite**(pin, valoare)” și valorile vor fi cuprinse între LOW(0) și HIGH(255), iar pentru citire vom folosi sintaxa “**analogRead**(pin)”, cu intervalul de valori cuprins între 0 și 1023.

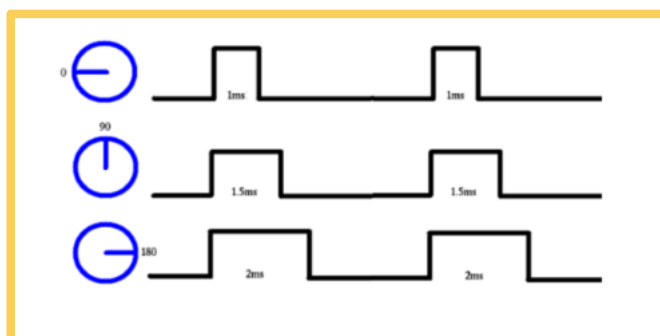


Pe placa Arduino pinii analogici au inițiala “A” (A0, A1, ..., A5).

Analogic	
Scriere	Citire
Are 256 valori de la 0 la 255, unde 0 = LOW și 255 = HIGH	Rezultat de la 0 la 1023, adică 1024 valori
Este posibilă doar pe pinii digitali marcați (~)	Se poate doar pe pinii de input analog (A0, A1, A2, A3, A4, A5)
Instrucțiune: <code>analogWrite(pin, valoare);</code>	Instrucțiune: <code>analogRead(pin);</code>

Semnalul PWM (Pulse-Width Modulation)

Semnalul PWM arată astfel:



Semnalul Pulse-Width Modulation



Pe placa Arduino funcția PWM este disponibilă pe pinii D3, D5, D6, D9, D10, D11.

Semnalul PWM este un concept important în lumea electronicii și este adesea folosit la placa Arduino pentru a controla cu precizie luminozitatea dispozitivelor cum sunt LED-urile, viteza motoarelor sau poziția servo-motoarelor.

Pentru a mapa o valoare analogică (care variază între 0 și 1023) la un semnal de ieșire PWM (care variază de la 0 la 255), putem utiliza funcția: `map(val, L1, H1, L2, H2)`. Aceasta funcție are 5 parametri: primul este variabila în care este stocată valoarea analogică, L1 și H1 sunt limitele intervalului inițial (0, 1023), iar L2 și H2 sunt limitele intervalului final (0, 255).



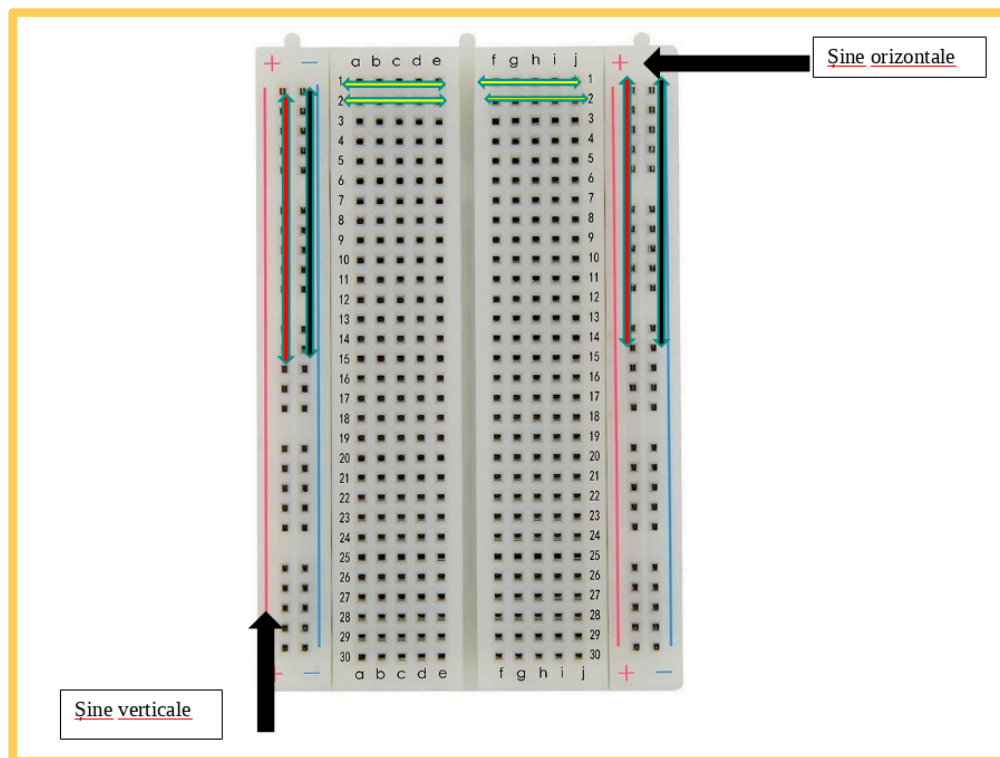
<https://docs.arduino.cc/learn/microcontrollers/analog-output/>



Breadboard-ul

Breadboard-ul este o placă de testare sau de prototipare care este folosită în general pentru realizarea rapidă a montajelor fără a fi nevoie de lipirea firelor, de exemplu pentru testarea proiectelor.

Un breadboard este de obicei împărțit în patru secțiuni, două secțiuni exterioare și două secțiuni interioare. Cele două secțiuni exterioare ale panoului sunt utilizate de obicei exclusiv pentru alimentarea cu curent electric. Aceste benzi sunt de obicei marcate cu linii roșii și albastre, cu semnele plus (+) și respectiv minus (-). Acestea mai sunt numite "șine" și sunt de obicei utilizate pentru a furniza energie electrică circuitului atunci când le conectezi la un acumulator sau la o altă sursă de alimentare externă.



Breadboard-ul

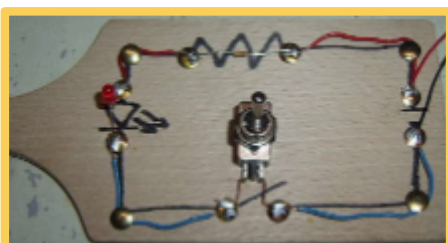
După cum se poate observa în figură există o anumită distanță pe mijlocul plăcii care o separă în 2 părți simetrice. Partea din mijloc este utilizată de obicei pentru a face conexiunea între părțile unor componente, deoarece separă pinii din partea stânga de pinii din partea dreaptă.

Dacă am desface banda adezivă de pe spatele plăcii, ea ar arăta ca în imaginea următoare:

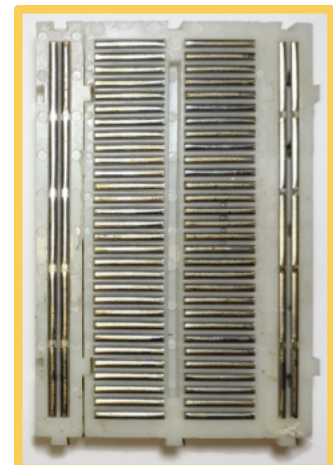


ȘTIAȚI CĂ?

Placa de testare poartă acest nume deoarece are o formă similară cu tocătoarele folosite pentru tăierea pâinii (în limba engleză "breadboard" înseamnă "tocător de pâine").



Tocător de pâine folosit ca breadboard



Spatele breadboard-ului

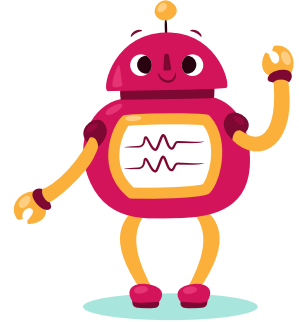
Noțiunea de algoritm

Algoritmul este o **succesiune finită de operații**, care se execută în ordinea scrierii lor, astfel încât plecând de la un set de date, numite datele problemei sau date de intrare, ce îndeplinesc anumite condiții, să obținem într-un interval de timp finit un set de valori, numite soluțiile problemei sau date de ieșire.

Exemplu de algoritm

Modul de preparare a unui ceai:

- **Pas1:** Se pune apa la fiert.
- **Pas2:** Se adaugă 2 pliculețe de ceai.
- **Pas3:** Când apa a fiert destul, se ia ceainicul de pe foc.
- **Pas4:** Se toarnă ceaiul într-o ceașcă.
- **Pas5:** Dacă se dorește se adaugă o cantitate de zahăr.



Pentru ca un algoritm să fie corect scris el trebuie să îndeplinească următoarele condiții:

A. Generalitatea (universalitate), adică să poată fi folosit pentru mai multe probleme de același tip. De exemplu, algoritmul care determină suma a două numere naturale trebuie să funcționeze pentru oricare două numere naturale, nu numai pentru anumite numere naturale.

B. Finitudinea, adică algoritmul trebuie să permită obținerea soluției problemei după parcurgerea unui număr finit de pași.

C. Determinismul (claritatea), adică pentru același set de valori de intrare se obțin întotdeauna aceleași soluții.

Pentru a se obține un algoritm cât mai bun, pe lângă cele trei condiții de mai sus, mai trebuie avută în vedere și următoarea caracteristică:

D. Eficiența constă în capacitatea algoritmului de a da soluția unei probleme într-un timp cât mai bun, folosirea unei memorii optime etc.

Exemplu de algoritm

Realizați algoritmul care calculează suma a două numere naturale.

START

Pas1: Citim cele 2 numere naturale a și b

Pas2: Calculăm suma celor 2 numere naturale $s = a + b$

Pas3: Afișăm suma s

STOP

Proiect 1 - LED



- **Scopul proiectului:** vom aprinde și stinge un LED obișnuit.
- **Componentele necesare:**
 - Arduino Uno;
 - Rezistor (220 Ω);
 - 1x LED;
 - Fire.

Diagrama de conectare

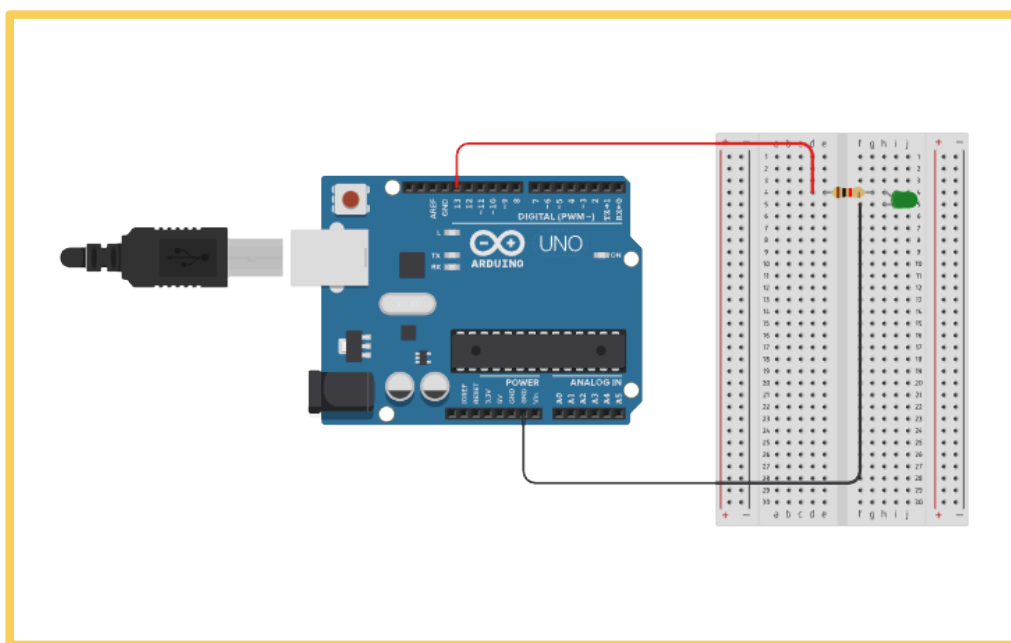


Diagrama de conectare a unui LED cu placa ARDUINO

Conexiunea firelor

- Anodul(+) se conectează la rezistorul de 220 Ω , iar apoi la D13;
- Catodul(-) se conectează la GND.

Codul sursă



<https://www.tinkercad.com/things/iYECbEQQWAK-01-led>



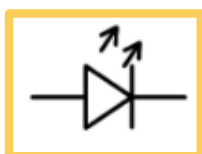
```

sketch_mar1a | Arduino IDE 2.3.0
File Edit Sketch Tools Help
Select Board
sketch_mar1a.ino
1 void setup()
2 { // Setăm pinul 13 asociat cu LED-ul integrat de pe placa Arduino pentru a fi folosit ca ieșire
3   pinMode(13,OUTPUT);
4 }
5 void loop() {
6   digitalWrite(13,LOW); //LED oprit
7   delay(1000); //Întârziere de o secundă
8   digitalWrite(13,HIGH); //LED pornit
9   delay(1000); //Întârziere de o secundă
10 }
11

```

Codul sursă

Informații



Simbolul unui LED

LED (Light Emitting Diode) este un dispozitiv electronic care emite lumină atunci când este străbătut de curent electric. Este alcătuită dintr-un anod și un catod, rolul diodei fiind acela de a lăsa curentul să treacă printr-o singură direcție (exemplu de utilizare: protecție de polaritate inversă).



Simbolul unui rezistor

Scopul principal al unui rezistor este de a se opune trecerii curentului electric. Principala caracteristică a unui rezistor este rezistența, care se măsoară în **ohmi (Ω)**. Aceasta indică cât de mult opune rezistorul curentului electric. Valoarea rezistenței este indicată pe rezistor cu un cod de culori sau cu o valoare numerică; această rezistență poate fi măsurată și cu un aparat de măsură denumit multimetru.



ȘTIAȚI CĂ?

Pielea umană are, de asemenea, o anumită rezistență electrică datorată compoziției sale. Această rezistență poate varia în funcție de umiditatea pielii și de alți factori.



<http://eelectronicce.blogspot.com/2016/07/codul-culorilor-pentru-marcarea.html>



<https://www.youtube.com/watch?v=JOP9TKuWi4w>



Proiect 2 - Semafor



- **Scopul proiectului:** vom simula un semafor format din 3 LED-uri: verde, galben și roșu.
- **Componentele necesare:**
 - Arduino Uno;
 - 3 LED-uri (preferabil verde, galben și roșu);
 - 3 rezistori de 220 ohmi;
 - Fire.

Diagrama de conectare

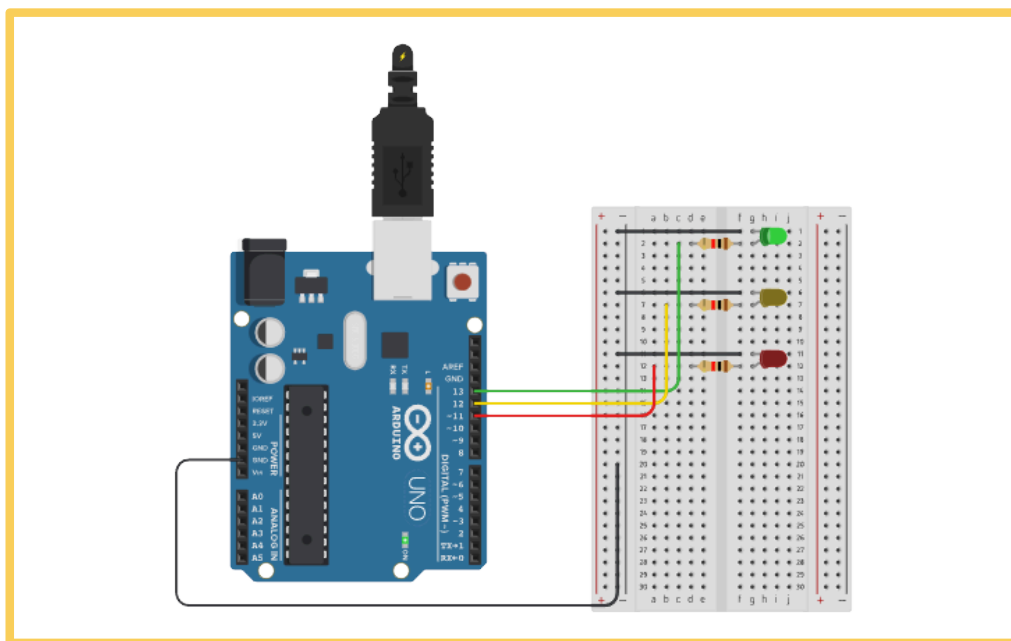


Diagrama de conectare pentru semafor

Conexiunea firelor

- Anodul(+) se conectează la rezistorul de 220Ω, iar apoi la D13, D12 și D11;
- Catodul(-) de la fiecare LED se conectează la GND.

Codul sursă



<https://www.tinkercad.com/things/3nNf3i1tBtU-02-led>





```
1 void setup() {
2   pinMode(11, OUTPUT);
3   pinMode(12, OUTPUT);
4   pinMode(13, OUTPUT);
5 }
6
7 void loop() {
8   digitalWrite(13, HIGH);
9   delay(3000);
10  digitalWrite(12, HIGH);
11  delay(2000);
12  digitalWrite(13, LOW);
13  digitalWrite(12, LOW);
14  digitalWrite(11, HIGH);
15  delay(3000);
16  digitalWrite(12, HIGH);
17  delay(2000);
18  digitalWrite(11, LOW);
19  digitalWrite(12, LOW);
20  digitalWrite(13, HIGH);
21  delay(3000);
22 }
23
```

Codul sursă

Proiect 3 - Tranzistor



- **Scopul proiectului:** vom aprinde și stinge un LED utilizând un tranzistor de tip NPN.
- **Componentele necesare:**
 - Arduino Uno;
 - LED;
 - Tranzistor NPN (2N222, care se află de obicei în kit-urile Arduino);
 - 2 rezistori de 220Ω;
 - Fire.

Diagrama de conectare

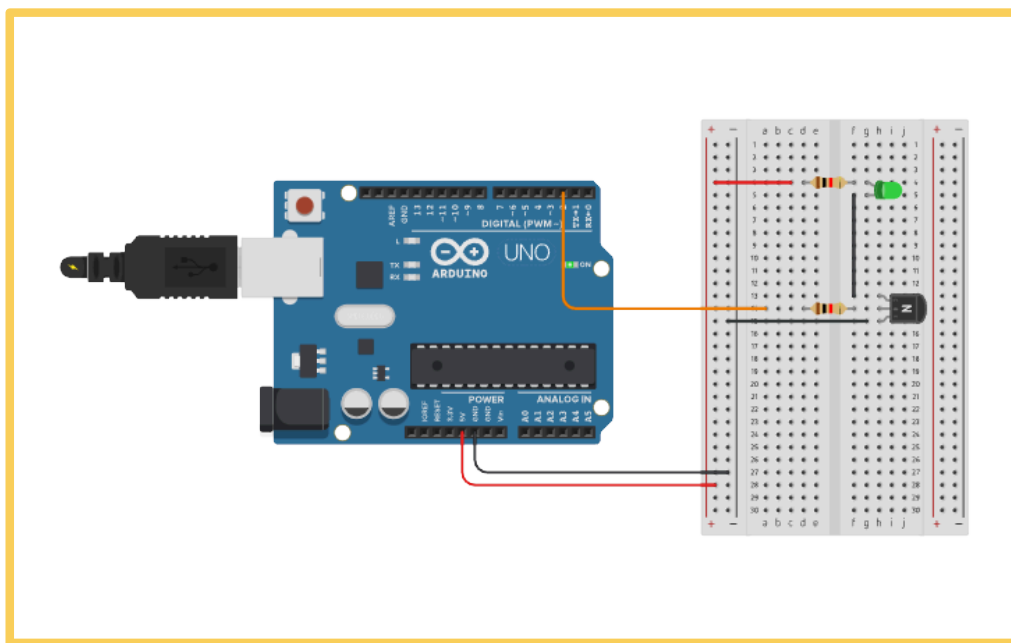


Diagrama de conectare pentru tranzistor

Conexiunea firelor

- Anodul (+) LED-ului se conectează la rezistorul de 220Ω, iar apoi la 5V;
- Catodul (-) LED-ului se conectează la pinul C (colector) al tranzistorului;
- La baza tranzistorului se leagă un rezistor de 220Ω, iar apoi se conectează la placa Arduino la pinul D2;
- Emitorul tranzistorului se conectează la GND.

Codul sursă



<https://www.tinkercad.com/things/IFDhvd8VEq2-03-led>

