

## MĂSURAREA TEMPERATURII PRIN INTERMEDIUL INSTRUMENTULUI ARDUINO

**Dorin AFANAS**, doctor, conferențiar universitar

<https://orcid.org/0000-0001-7758-943X>

**Gheorghe DADU**, locotenent colonel, AMFA

**Tudor TIMERCAN**, maior, AMFA

**Vadim BODEANCIUC**, locotenent colonel, AMFA

**Rezumat.** Există un număr imens de proiecte Arduino [1, 2, 3, 4, 5]. Aceasta se datorează faptului că Arduino este o platformă de prototip electronic cu sursă deschisă. A luat naștere în anul 2005 în cadrul Institutului de proiectare a interacțiunilor (Interaction Design Institute) din Italia. Unul dintre principalii fondatori ai acestui proiect a fost Massimo Banzi. În prezentul articol vom prezenta circuitul fizic și codul programului pentru măsurarea temperaturii unui obiect.

**Cuvinte cheie:** arduino, temperatură, circuit, codul programului.

**Abstract.** There are a huge number of Arduino projects [1, 2, 3, 4, 5]. This is because Arduino is an open source electronic prototyping platform. It was born in 2005 within the Interaction Design Institute in Italy. One of the main founders of this project was Massimo Banzi. In this article we will present the physical circuit and program code for measuring the temperature of an object.

**Keywords:** arduino, temperature, circuit, program code.

### I. Introducere

În cadrul acestui proiect vom realiza un circuit electric ce conține:

- fire pentru conexiune;
- trei rezistoare de  $220\Omega$ ;
- trei LED-uri roșii;
- un senzor de temperatură TMP36.

La finele acestei activități vom putea măsura temperatura unui corp.

Obiectivele acestei activități sunt:

- conștientizarea intrărilor analogice;
- formarea competenței de utilizare a monitorului serial;
- formarea competenței de utilizare a senzorului de temperatură TMP36;
- formarea competenței de a construi un circuit electric;
- conștientizarea tipurilor de prezentare ale circuitelor electrice;
- conștientizarea necesității reprezentării circuitelor electrice prin două moduri;
- realizarea unor calcule matematice simple cu fracțiile;
- conștientizarea instrucțiunilor utilizate;
- conștientizarea variabilelor utilizate;

- formarea competenței de a crea programul la calculator aferent circuitului electric construit.

## II. Noțiuni preliminare

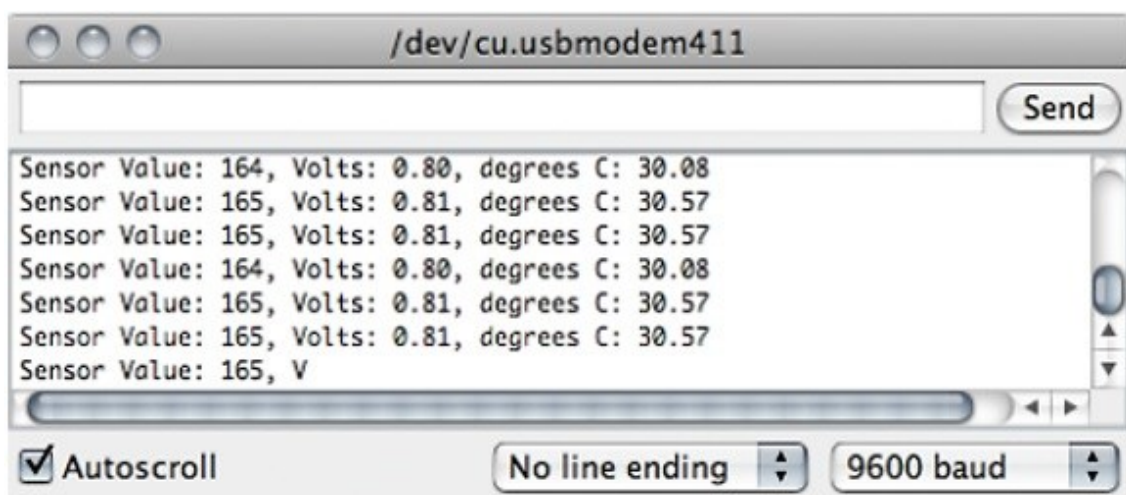
În lumea fizicii există multe procese destul de interesante. Chiar dacă Arduino este un instrument digital, totuși este posibil ca acesta să primească informații de la senzorii analogici pentru a măsura careva lucruri precum temperatura sau lumina. Pentru realizarea acestei activități, vom profita de convertorul analog-digital încorporat (ADC) al Arduino. Pini analogici A0 – A5 poate raporta o valoare între 0 și 1023, care se mapează la un interval de la 0 volți la 5 volți.

Vom utiliza un senzor de temperatură (fig. 1) care poate genera o tensiune variabilă în funcție de temperatura pe care o simte. Posedă trei pini: unul se conectează la sol, altul se conectează la sursa de energie, iar al treilea generează o tensiune variabilă către Arduino. În sketch-ul (schița) acestui proiect, vom citi ieșirea senzorului și îl vom folosi pentru a porni și a stinge LED-urile, indicând temperatura măsurată. Există diferite modele ale senzorilor de temperatură. Acest model, TMP36, este convenabil pentru noi, deoarece produce o tensiune care se schimbă direct proporțional cu temperatura în grade Celsius.



**Figura 1. Senzor de temperatură**

Software-ul Arduino (IDE) vine cu un instrument numit monitor serial (fig. 2) care ne permite să vizualizăm și să înregistrăm rezultatele din microcontroler. Folosind monitorul serial, putem obține informații despre starea senzorilor și vom conștientiza ce se întâmplă în circuitul nostru și în cod în timp ce rulează programul.



**Figura 2. Monitor serial**

## II. Prezentarea circuitului

Circuitul este reprezentat în figurile 3 și 4.

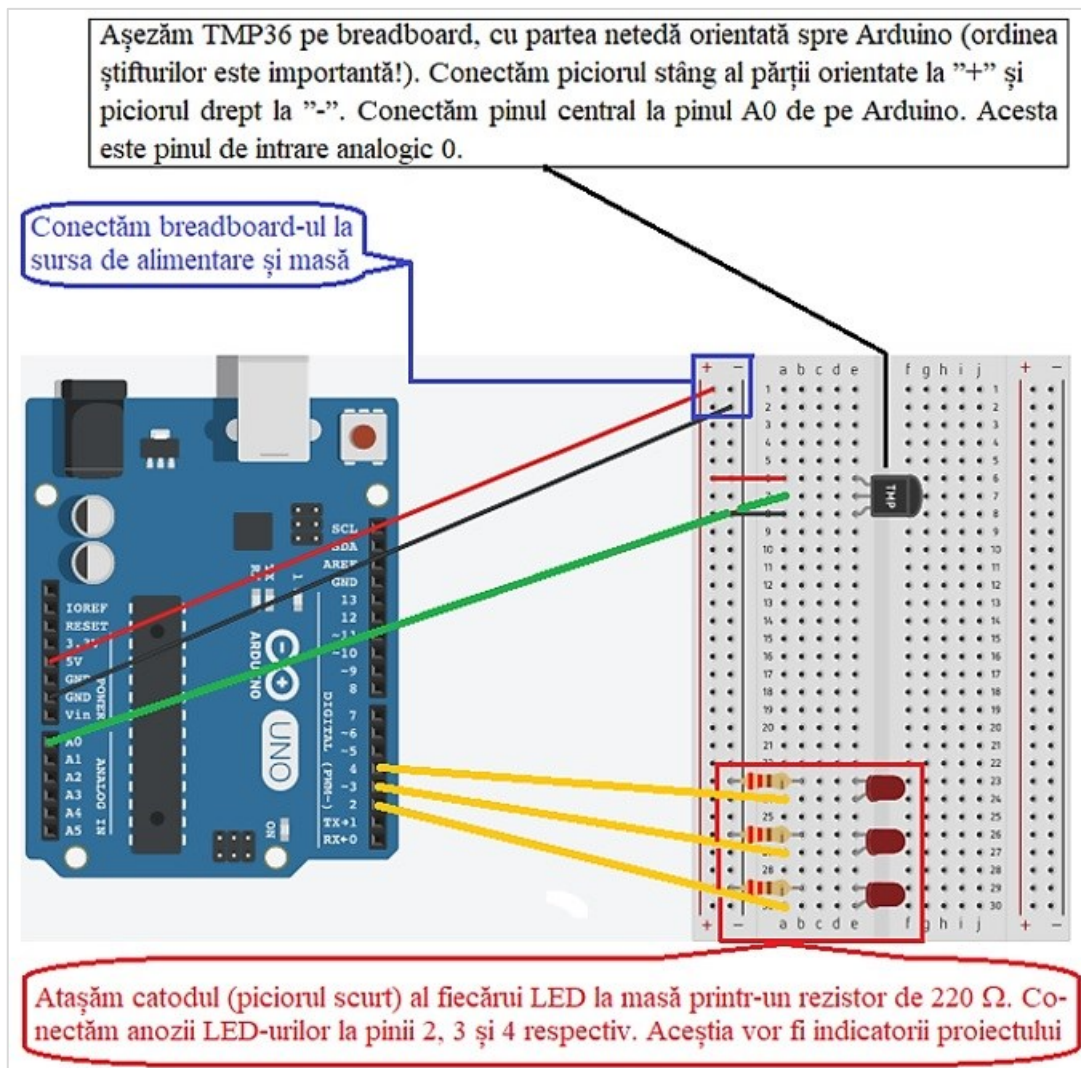


Figura 3. Ilustrația circuitului

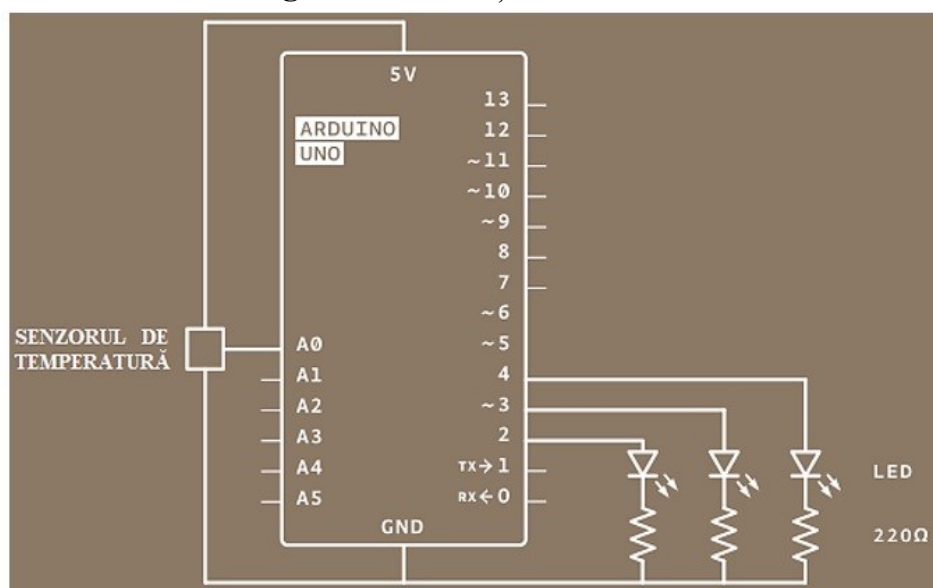
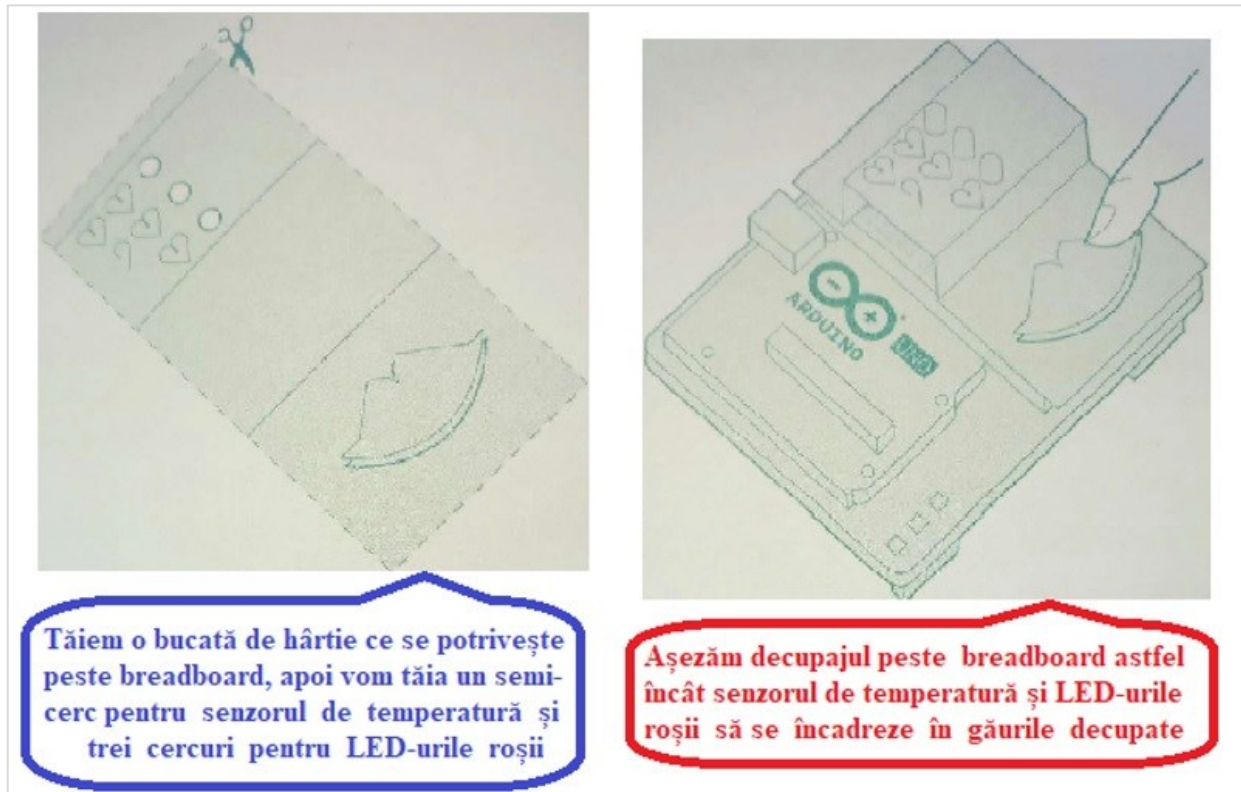


Figura 4. Prezentarea schematică a circuitului

**Notă.** Se recomandă de verificat temperatura ambientă din încăpere înainte de a continua. Putem verifica manual chiar acum, însă acest lucru poate fi realizat și prin calibrare. Este posibil să utilizăm un buton pentru a seta temperatura de bază sau să lăsăm Arduino să preia un eșantion înainte de a începe bucla **loop()** și să-l utilizăm ca punct de referință.



**Figura 5. Decupaj de hârtie**

În cadrul acestor activități se recomandă de creat o interfață pentru senzorul nostru cu scopul ca proiectul să devină cât mai atractiv (fig. 5). Pentru aceasta putem folosi un decupaj de hârtie. De asemenea, putem eticheta LED-urile pentru a le da un anumit sens. De exemplu, aprinderea unui LED înseamnă că temperatura în încăpere este joasă, aprinderea a două LED-uri înseamnă că temperatura în încăpere este normală, iar aprinderea a trei LED-uri – temperatura este prea înaltă.

### III. Codul programului

Pentru scrierea codului respectăm următorul algoritm:

**3.1. O pereche de constante utile.** Constantele sunt similare cu variabilele prin faptul că ne permit să definim în mod unic un proces din program, dar spre deosebire de variabile nu se pot modifica. Denumim intrarea analogică pentru o referință și creăm o altă constantă pentru a menține temperatura de referință. Pentru fiecare 2 grade Celsius pe linia de bază se va aprinde un LED. Mai sus am făcut cunoștință cu datele **int** utilizate pentru a identifica ce senzor este pornit. Temperatura este stocată ca un număr flotant. Acest tip de număr

poate conține o cifră după virgulă și este utilizat pentru numere care pot fi exprimate și ca fracții. Cele două constante în programul nostru arată astfel:

```
1  const int sensorPin = A0;
2  const float baselineTemp = 20.0;
```

**3.2. Inițializarea portului serial la viteza dorită.** În configurație vom utiliza o comandă nouă **Serial.begin()**. Ea deschide o conexiune între Arduino și computer, astfel încât să putem vedea pe ecran valorile de la intrarea analogică.

Argumentul 9600 este viteza cu care Arduino va comunica, adică Arduino va comunica cu 9600 biți pe secundă. Vom utiliza monitorul serial al IDE Arduino pentru a vizualiza informațiile pe care le alegem să le trimitem de la microcontrolerul nostru. Când deschidem monitorul serial al IDE, verificăm dacă rata de transmisie este într-adevăr 9600.

```
3  void setup(){
4  Serial.begin(9600); // deschidem un port serial
```

**3.3. Inițializarea direcțiilor pin-ului digital.** Următoarea comandă este **for()** pentru a seta pinii la care am atașat LED-urile noastre. În loc să le dăm nume unice și să tastăm funcția **pinMode()** pentru fiecare pin, putem utiliza operatorul **for()** pentru a le parcurge rapid. Aceasta este util când avem la îndemână un număr mare de lucruri similare pe care dorim să le repetăm într-un program. În cazul nostru vom spune operatorului **for()** să ruleze secvențial prin pinii 2, 3 și 4. Tutorialul pentru operatorul **for()** o putem găsi pe adresa: [arduino.cc/for](http://arduino.cc/for).

```
5  for(int pinNumber = 2; pinNumber<5; pinNumber++){
6  pinMode(pinNumber, OUTPUT);
7  digitalWrite(pinNumber, LOW);
8  }
9  }
```

**3.4. Citirea datelor (informației) de la senzorul de temperatură.** În operatorul **loop()** vom utiliza o variabilă locală numită **sensorVal** pentru a stoca datele de la senzorul TMP36. Pentru a obține valoarea de la senzor vom apela la funcția **analogRead()** care ia un argument: pe ce pin ar trebui să ia o tensiune. Valorile cuprinse între 0 și 1023 ne reprezintă tensiunea pe pin.

```
10 void loop(){
11     int sensorVal = analogRead(sensorPin);
```

**3.5. Expedierea valorilor senzorului de temperatură la computer.** Funcția **Serial.print()** trimite informații de la Arduino la un computer conectat. Putem vizualiza

aceste informații pe monitorul serial. Dacă dăm funcției **Serial.print()** un argument între ghilimele, acesta va imprima textul pe care l-am introdus. Dacă îi vom da o variabilă ca argument, atunci va imprima valoarea acelei variabile.

```
12 Serial.print("Sensor Value: ");
13 Serial.print(sensorVal);
```

**3.6. Convertirea datelor (citirii) senzorului în tensiune.** Aplicând calcule matematice simple vom putea determina tensiunea reală pe pin. Tensiunea va avea o valoare între 0 – 5 volți și va avea o parte fracționară (de exemplu, ar putea fi de 2,5 de volți). Deci va trebui să o stocăm într-un **float**. Cu acest scop vom crea o variabilă numită tensiune pentru a menține acest număr. Împărțim **sensorVal** la 1024,0 și multiplicăm cu 5,0. Numărul obținut și va reprezenta valoarea tensiunii pe pin.

```
14 // convertim datele (citirea) ADC în tensiune
15 float voltage = (sensorVal/1024.0) * 5.0;
```

Similar ca pentru valoarea senzorului, vom imprima datele obținute pe monitorul serial.

```
16 Serial.print(" , Volts: ");
17 Serial.print(voltage);
```

**3.7. Convertirea tensiunii în temperatură și expedierea valorii la computer.** În fișa tehnică a senzorului există informații despre gama tensiunii de ieșire. Fișele tehnice sunt ca niște manuale pentru componente electronice. În cazul nostru fișa tehnică pentru acest senzor explică că fiecare schimbare de 10 milivolți față de senzor este echivalentă cu o modificare a temperaturii de 1 grad Celsius. De asemenea, indică faptul că senzorul poate citi și temperaturi sub 0 grade Celsius. Din acest motiv, va trebui să creăm un offset pentru valori sub 0 grade. Pentru a obține temperatura exactă în grade Celsius din tensiunea citită scădem 0,5 și înmulțim cu 100. Stocăm numărul obținut într-o variabilă cu virgulă numită temperatură.

```
18 Serial.print(" , degrees C: ");
19 // convertim tensiunea în grade Celsius
20 float temperature = (voltage - .5) * 100;
```

După ce am obținut temperatura reală, o imprimăm pe monitorul serial. Deoarece variabila de temperatură este ultimul lucru pe care îl vom imprima, vom folosi o comandă: **Serial.println()**. Această comandă va crea o nouă linie în monitorul serial după ce va trimite valoarea. Aceasta face ca informația să fie mai ușor de citit atunci când sunt tipărite.

```
21 Serial.println(temperature);
```

**3.8. Oprirea LED-urilor pentru o temperatură scăzută.** Cu temperatura reală putem configura o declarație `if()...else` pentru a aprinde LED-urile. Folosind temperatura de bază ca punct de plecare, vom aprinde un LED pentru fiecare 2 grade de creștere a temperaturii. Vom căuta o gamă de valori pe măsură ce ne vom deplasa pe scara temperaturii.

```
22 if(temperature < baselineTemp+2){
23   digitalWrite(2, LOW);
24   digitalWrite(3, LOW);
25   digitalWrite(4, LOW);
```

**3.9. Pornirea unui LED pentru o temperatură scăzută.** Operatorul `&&` înseamnă „și”, într-un sens logic. Putem verifica condiții multiple: ”dacă temperatura este cu 2 grade mai mare decât cea de bază și dacă este mai mică cu 4 grade față de cea de bază”.

```
26 }else if(temperature >= baselineTemp+2 &&
    temperature < baselineTemp+4){
27   digitalWrite(2, HIGH);
28   digitalWrite(3, LOW);
29   digitalWrite(4, LOW);
```

**3.10. Pornirea unui LED pentru o temperatură medie.** Dacă temperatura este între 4 și 6 grade Celsius mai mare decât cea de bază, atunci acest bloc de coduri aprinde și LED-ul pinului 3.

```
30 }else if(temperature >= baselineTemp+4 &&
    temperature < baselineTemp+6){
31   digitalWrite(2, HIGH);
32   digitalWrite(3, HIGH);
33   digitalWrite(4, LOW);
```

**3.11. Pornirea unui LED pentru o temperatură ridicată.** Codul este prezentat mai jos:

```
34 }else if(temperature >= baselineTemp+6){
35   digitalWrite(2, HIGH);
36   digitalWrite(3, HIGH);
37   digitalWrite(4, HIGH);
```

Convertorul analog-digital poate citi atât de repede și deci ar trebui să punem o mică întârziere chiar la finele operatorului `loop()`. Dacă vom citi din el prea des, atunci valorile vor apărea neregulate.

```
38 }  
39 delay(1);  
40 }
```

După ce codul a fost încărcat pe Arduino, facem clic pe pictograma monitorului serial. Ar trebui să apară un flux de valori, formatat astfel:

Senzor: 200, Volți: .70, grade C: 17

Vom încerca să punem degetele în jurul senzorului în timp ce acesta este conectat la panoul de verificare și observăm ce se întâmplă cu valorile din monitorul serial. Notăm care este temperatura când senzorul este lăsat în aer liber.

După realizarea acestor activități putem închide monitorul serial și schimbăm constanta de bază de timp în programul nostru la valoarea la care am observat temperatura. Încărcăm codul din nou și încercăm să ținem senzorul în degete. Pe măsură ce temperatura crește, ar trebui să vedem LED-urile pornind unul câte unul.

## CONCLUZII

Intenția proiectului Arduino constă în reducerea pragului pentru studenții din cadrul AMFA să creeze lucrări interactive. Pe lângă costul redus se ia în considerație și dificultatea învățării, adică proiectul dat nu necesită mari investiții financiare și nu necesită cunoștințe profunde de programare pentru crearea diferitor lucrări în funcție de creativitatea studenților.

Pentru beneficiul și în folosul mai multor entuziaști proiectul Arduino respectă acordul de hardware open source, adică oricine are acces la desenele de proiectare ale plăcii de circuit hardware Arduino și la codul programului.

Arduino nu numai că simplifică procesul de lucru cu un microcontroler, ci oferă cadrelor didactice, studenților și pasionaților avantaje și caracteristici pe care alte sisteme pur și simplu nu le posedă.

*Articol realizat de către membrii laboratorului Dispozitive electrice și electronice, Catedra Comunicații și Informatică, Academia Militară a Forțelor Armate "Alexandru cel Bun" în cadrul proiectului de stat „Metodologia implementării TIC în procesul de studiu a științelor reale în sistemul de învățământ al Republicii Moldova din perspectiva inter/transdisciplinarității (conceptul STEAM)” cu cifrul 20.80009.0807.20.*

## Bibliografie

1. AFANAS, D. Programarea servomotorului și sugestii de utilizare. In: *Conference on Applied and Industrial Mathematics: CAIM 2022*, Ed. 29, 25-27 august 2022,



- Chișinău. Chișinău, Republica Moldova: Bons Offices, 2022, Ediția a 29 (R), pp. 44-50. ISBN 978-9975-76-401-8.
2. AFANAS, D.; GUȚULEAC, L.; BARDARI, L. Activități de laborator la fizică și informatică în concept STEAM. În: *Materialele Conferinței științifice internaționale "Abordări inter/transdisciplinare în predarea științelor reale, (Concept STEAM)"* dedicată aniversării a 70 de ani de la nașterea profesorului universitar Anatol Gremalschi, 29 – 30 octombrie 2021. Volumul II: Abordări inter/transdisciplinare în procesul de predare-învățare a fizicii și științelor tehnice (concept STEAM) & Integrarea STEAM în procesul de studiere a biologiei, chimiei și geografiei, (ISBN 978-9975-76-358-5. pp. 8 – 14 (link: [https://ibn.idsi.md/ro/vizualizare\\_articol/142004](https://ibn.idsi.md/ro/vizualizare_articol/142004)).
  3. Bureau U. S. of Naval Personnel. *Basic Machines and How They Work*. New York: Dover Publications, 1971.
  4. MACAULAY, D. *The Way Things Work*. Boston: Houghton Mifflin, 1988.
  5. KAMM, L. J. *Designing Cost-Efficient Mechanisms: Minimum Constraint Design, Designing with Commercial Components, and Topics in Design Engineering*. New York: McGraw-Hill, 1990. Also published in paperback by the Society of Automotive Engineers, Inc., 1993.