

ACTIVITĂȚI DE LABORATOR LA FIZICĂ ȘI INFORMATICĂ ÎN CONCEPT STEAM

Dorin AFANAS, dr., conf. univ.

Leonid GUȚULEAC, dr., conf. univ.

Universitatea de Stat din Tiraspol

Lilia BARDARI

Rezumat. În prezentul articol sunt evidențiate unele probleme care persistă în învățământul național din Republica Moldova. Sunt propuse unele soluții pentru diminuarea lor. Una din aceste soluții ar fi integrarea învățământului STEAM prin intermediul lucrărilor de laborator, având la bază proiectul Arduino.

Abstract. This article highlights some problems that persist in national education in the Republic of Moldova. Some solutions are proposed to reduce them. One of these solutions would be the integration of STEAM education through laboratory work, based on the Arduino project.

Cuvinte chee: Arduino, circuit electric, codul programului, sunet.

Keywords: Arduino, electrical circuit, program code, sound.

1. Introducere

La etapa actuală de dezvoltare a societății contemporane cu toții suntem nevoiți să utilizăm tehnologia, indiferent de cunoștințele pe care le posedăm la moment în acest domeniu. Această tehnologie poate fi sofisticată sau mai puțin sofisticată, pentru unii – atractivă, iar pentru alții nu prea. Însă indiferent din ce categorie facem parte, noi suntem nevoiți să apelăm la ea zi de zi în activitățile noastre.

O soluție pentru a spori atractivitatea tehnologiilor performante existente ține de învățământul actual în instituțiile de învățământ din țară care nu mai satisface cerințelor actuale la nivelul cerut de societate. În acest sens, o soluție ar putea fi introducerea treptată a învățământului STEAM, mai ales că asemenea experiențe reușite există deja în alte țări. Însă introducerea învățământului STEAM în Republica Moldova se confruntă cu mai multe probleme:

- rezistența din partea unei categorii de profesioniști care privesc sceptic la un astfel de învățământ;
- lipsa specialiștilor în domeniul STEAM;
- costul ridicat al utilajelor care este folosit în învățământul STEAM;
- imposibilitatea susținerii financiare a unui asemenea învățământ de către stat;
- reducerea interesului față de științele exacte mai ales față de matematică, fizică, chimie, geografie, iar pe alocuri chiar și față de informatică.

Astfel apare o dilemă sau un paradox: pe de o parte societatea contemporană utilizează tehnologii performante, iar pe de alta – se măresc carențele la capitolul specialiști în științe exacte și în tehnică. Problemele elucidate mai sus, după părerea unor specialiști, sunt greu de rezolvat și prin urmare se crează impresia că nu pot fi rezolvate în viitorul apropiat. Însă după părerile altor specialiști, numiți optimiști în sensul bun al cuvântului, asemenea probleme pot

fi rezolvate cu mult mai repede și cu efort minim depus, mai ales pentru a crește atractivitatea față de științele exacte.

Pentru creșterea atractivității față de disciplinele menționate ne poate veni în ajutor proiectul Arduino [1]. Menționăm doar că încă în anul 2013 peste 700.000 de plăcuțe oficiale erau în posesia utilizatorilor [2].

2. Teremin bazat pe lumină

În cadrul acestui proiect vom realiza un circuit ce conține:

- fire pentru conexiune;
- un piezo element;
- un fototranzistor;
- un rezistor de 10 k Ω .

La finele acestei activități vom cunoaște cum putem crea sunete în dependență de iluminarea externă.

Obiectivele acestei activități sunt:

- formarea competenței de a crea sunete cu ajutorul funcției **tone()**;
- formarea competenței de calibrare a senzorilor analogici;
- formarea competenței de a construi un circuit electric;
- conștientizarea tipurilor de prezentare ale circuitelor electrice;
- conștientizarea necesității reprezentării circuitelor electrice prin două moduri;
- conștientizarea instrucțiunilor utilizate;
- conștientizarea variabilelor utilizate;
- conștientizarea constantelor utilizate;
- formarea competenței de a crea programul la calculator aferent circuitului electric construit.

2.1. Noțiuni preliminare. Tereminul este un instrument care scoate sunete pe baza mișcărilor mâinilor unui interpret (muzician). Tereminul detectează unde sunt mâinile unui interpret în raport cu două antene citind schimbarea capacitivă pe antene. Aceste antene sunt conectate la circuite analogice ce creează sunetul. O antenă controlează frecvența sunetului, iar cealaltă – volumul. În timp ce Arduino nu poate reproduce exact sunetele misterioase din acest instrument, este posibil să le emulăm folosind funcția **tone()**. În figura 1 este reprezentată diferența dintre impulsurile emise de **analogWrite()** și **tone()**. Acest lucru permite unui traductor ca, de exemplu, un difuzor sau un piezo să se deplaseze înainte și înapoi la viteze diferite.

În loc să detectăm capacitatea cu Arduino, vom folosi un fototranzistor pentru a detecta cantitatea de lumină. Mișcând mâinile peste senzor, vom schimba cantitatea de lumină ce cade pe fototranzistor. Schimbarea tensiunii pe pinul analogic va determina ce notă de frecvență să redăm.

Vom conecta fototranzistorul la Arduino generând un curent și deci o tensiune pe rezistorul de 10 kΩ. Rezistența fixă care se conectează la sol (la "–") limitează capătul inferior al gamei, iar luminozitatea limitează capătul înalt. În loc să lucrăm cu o gamă limitată, vom calibra citirile sensorului obținând valorile înalte și joase, mapându-le la frecvențe de sunet utilizând funcția `map()` pentru a obține cât mai multă autonomie din teren. Acest lucru va avea avantajul suplimentar de a regla citirile senzorilor ori de câte ori vom muta circuitul într-un mediu nou, cum ar fi o încăpere cu condiții de lumină diferite.

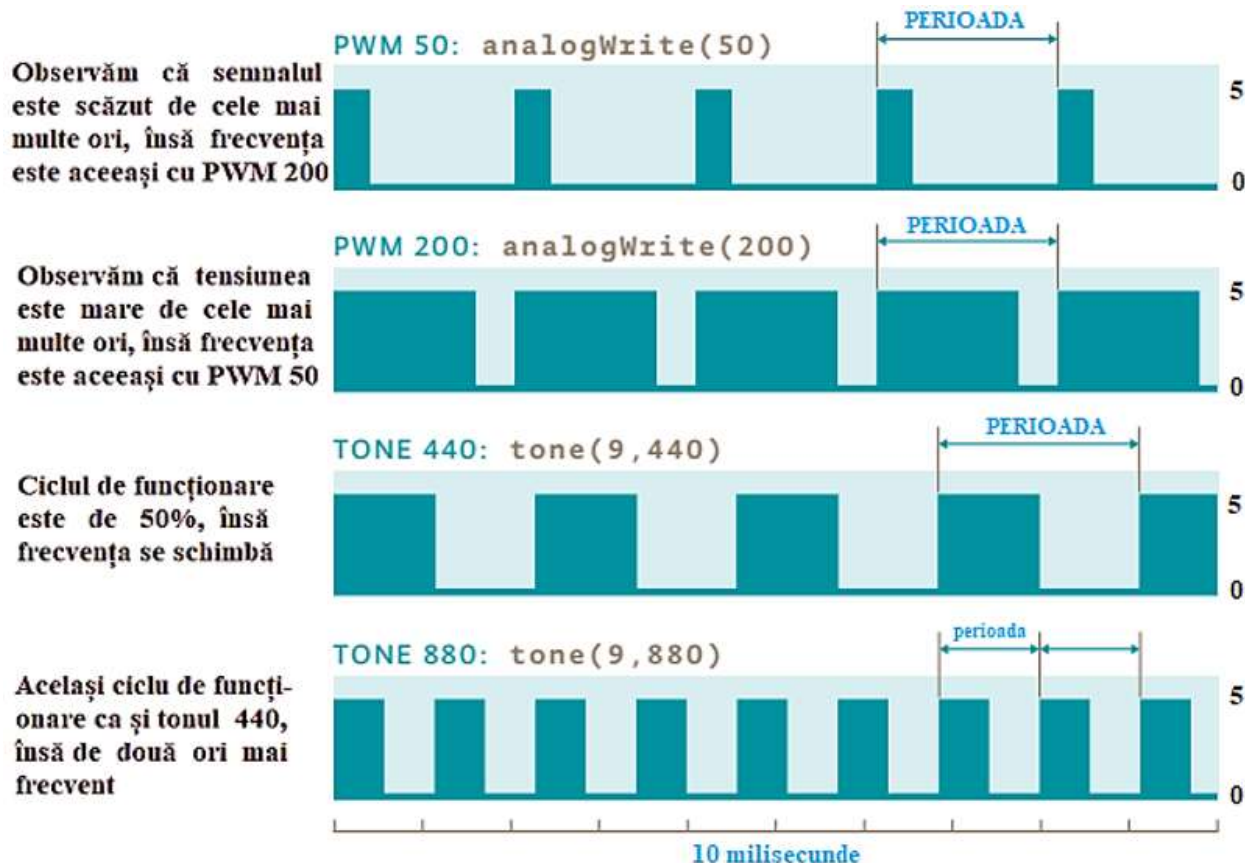


Figura 1.

Un piezo este un element mic care vibrează atunci când primește electricitate. Când se mișcă, deplasează aerul în jurul său, creând unde sonore.

2.2. Prezentarea circuitului. Se recomandă de prezentat circuitul în două moduri: o vizualizare pe panou (fig. 2) numită *ilustrația circuitului*, așa cum arată componentele reale și altul schematic (fig. 3) numită *prezentare schematică*, care este un mod mai abstract de a arăta relațiile dintre componente într-un circuit. Schemele nu arată întotdeauna unde componentele sunt plasate în realitate, ci arată mai mult cum sunt conectate între ele.

Tereminele tradiționale pot controla frecvența și volumul sunetului. În acest proiect vom putea controla numai frecvența. Deși nu putem controla volumul prin Arduino, este posibil să modificăm nivelul de tensiune care ajunge manual la difuzor.

Cadrul didactic va scoate în evidență următoarele întrebări:

- Ce se întâmplă dacă includem în circuit un potențiometrul în serie cu pinul 8 și un piezo?
- Dar un alt fototranzistor ?

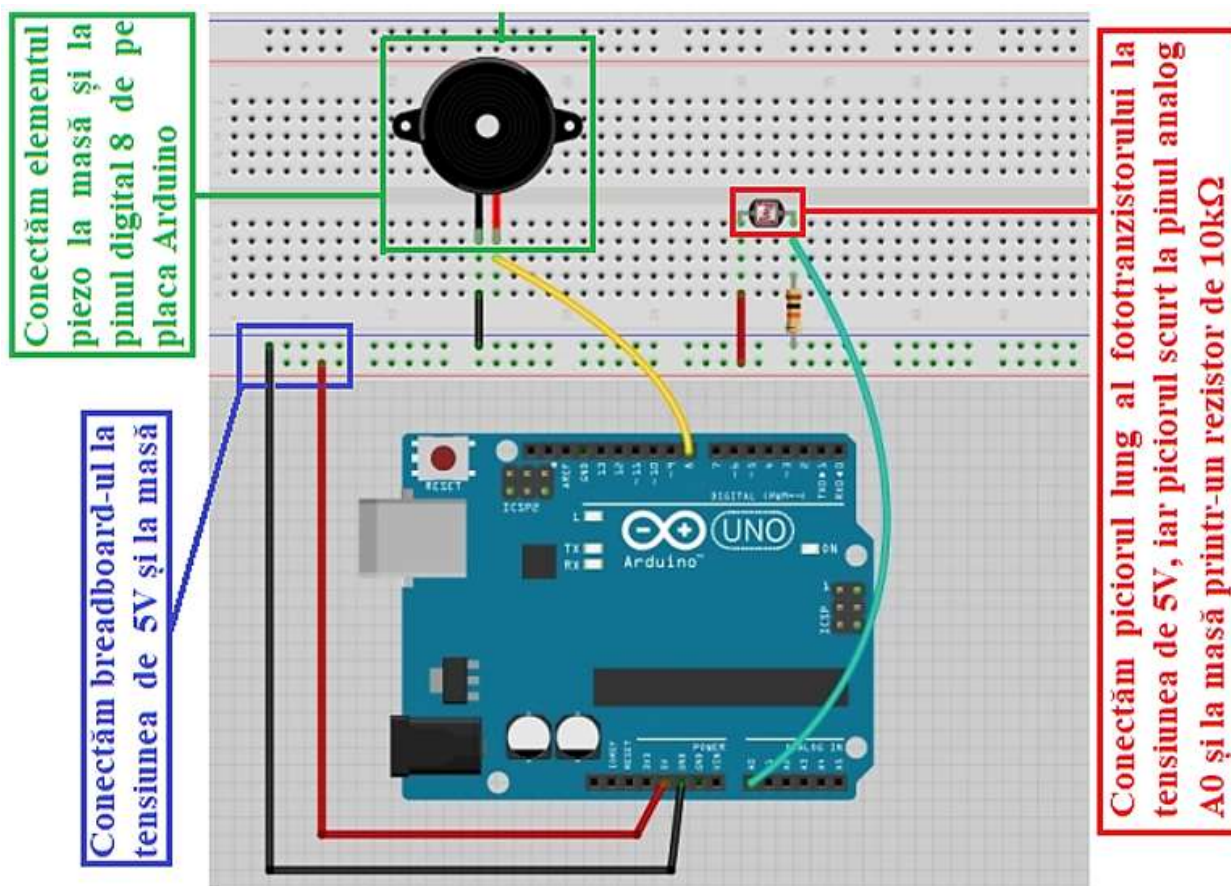


Figura 2.

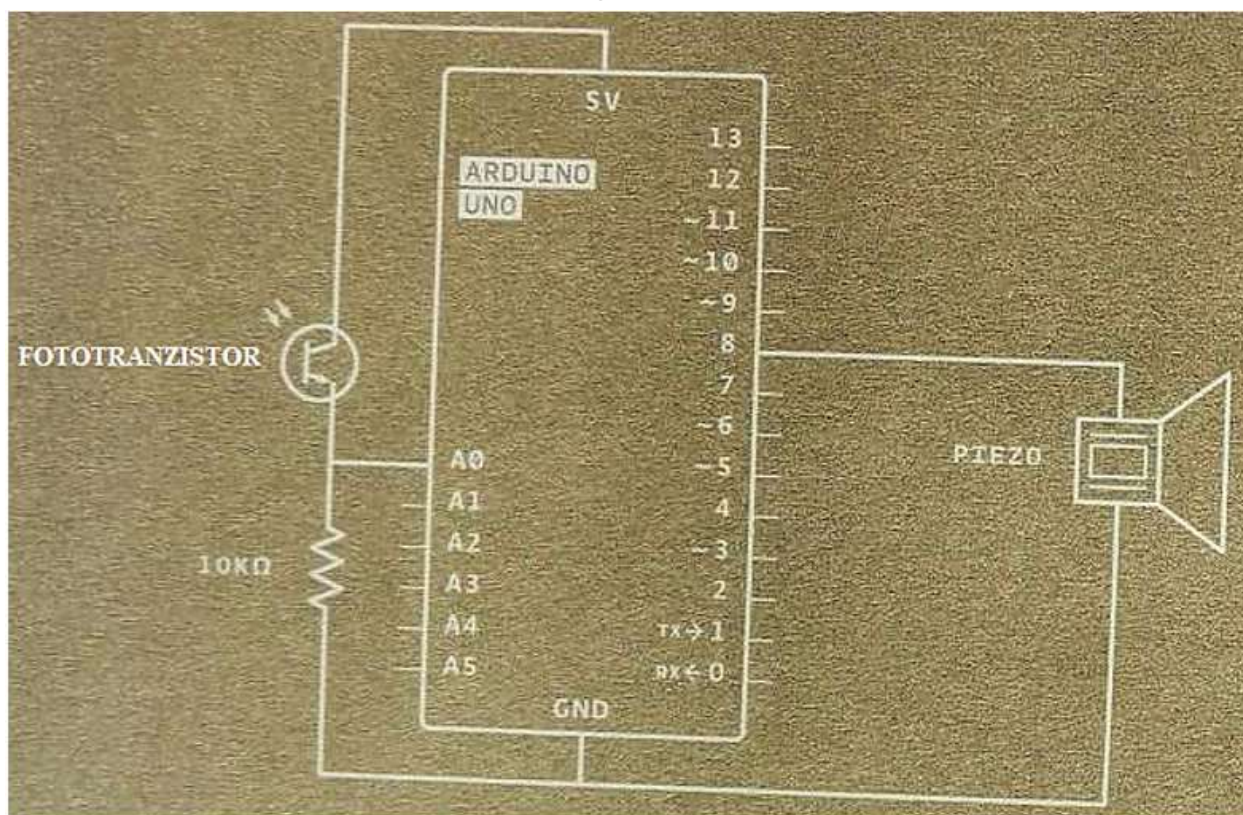


Figura 3.

2.3. Codul programului. Pentru scrierea codului respectăm următorul algoritm:

2.3.1. Creăm variabile pentru calibrarea senzorului. Creăm o variabilă pentru a păstra valoarea **analogRead()** din fototranzistor. Apoi creăm variabile pentru valorile mari și mici. Vom seta valoarea inițială în variabila **sensorLow** la 1023 și valoarea variabilei **sensorHigh** la 0. Când rulăm pentru prima dată programul, vom compara aceste numere cu citirile senzorului pentru a găsi maximul real și valorile minime.

```
1 int sensorValue;
2 int sensorLow = 1023;
3 int sensorHigh = 0;
```

2.3.2. Creăm constante pentru indicatorul de calibrare. Creăm o constantă numită **ledPin**, care ne va indica faptul dacă senzorul a finalizat cu succes calibrarea. În cazul nostru vom utiliza LED-ul integrat conectat la pinul 13.

```
4 const int ledPin = 13;
```

2.3.3. Setăm direcția pinului digital și setăm HIGH. În **setup()**, schimbăm **pinMode()** al **ledPin**-ului la OUTPUT și pornim lumina.

```
5 void setup() {
6   pinMode(ledPin, OUTPUT);
7   digitalWrite(ledPin, HIGH);
```

2.3.4. Utilizăm instrucțiunea while() pentru calibrare. Următorii pași vor calibra valorile maxime și minime ale senzorului. Vom utiliza o instrucțiune **while()** pentru a rula o buclă timp de 5 secunde, în timp ce buclele **while()** rulează până când se îndeplinește o anumită condiție. În acest caz vom utiliza funcția **millis()** pentru a verifica ora curentă. Funcția **millis()** raportează cât timp a funcționat Arduino de la ultima pornire sau resetare.

```
8 while (millis() < 5000) {
```

2.3.5. Comparăm valorile senzorului pentru calibrare. În buclă vom citi valoarea senzorului. Dacă valoarea este mai mică decât **sensorLow** (inițial 1023), vom actualiza acea variabilă, iar dacă este mai mare decât **sensorHigh** (inițial 0), acesta se actualizează.

```
9   sensorValue = analogRead(A0);
10  if (sensorValue > sensorHigh) {
11    sensorHigh = sensorValue;
12  }
13  if (sensorValue < sensorLow) {
14    sensorLow = sensorValue;
15  }
16 }
```

2.3.6. Indicăm finalizarea calibrării. Când au trecut 5 secunde, bucla **while()** se va încheia. Oprim LED-ul atașat la pinul 13. Vom utiliza senzorul cu valori înalte și joase tocmai înregistrate, pentru a scala frecvența în partea principală a programului.

```
17 digitalWrite(ledPin, LOW);
18 }
```

2.3.7. Citim și stocăm valoarea senzorului. În bucla **loop()**, citim valoarea de pe A0 și o stocăm în **sensorValue**.

```
19 void loop() {
20   sensorValue = analogRead(A0);
```

2.3.8. Mapăm valoarea senzorului la o frecvență. Creăm o variabilă numită **pitch**. Valoarea **pitch**-ului va fi mapată din **sensorValue**. Vom utiliza **sensorLow** și **sensorHigh** ca limite pentru valorile de intrare. Pentru valorile inițiale pentru ieșire, încercăm de la 50 la 4000. Aceste numere stabilesc intervalul de frecvențe pe care îl va genera Arduino.

```
21   int pitch = map(sensorValue, sensorLow, sensorHigh, 50, 4000);
```

2.3.9. Redăm frecvența. Apelăm la funcția **tone()** pentru a reda un sunet. Sunt necesare trei argumente: pe ce pin să redăm sunetul (în acest caz pinul 8), pe ce frecvență să redăm (determinată de variabila de ton) și cât timp să redăm nota (este recomandabil să încercăm 20 de milisecunde pentru început).

```
22   tone(8, pitch, 20);
```

Apoi, apelăm o întârziere **delay()** de 10 milisecunde pentru a acorda sunetului ceva timp de redare.

```
23   delay(10);
```

```
24 }
```

Când pornim Arduino pentru prima dată, există o fereastră de 5 secunde pentru a calibra senzorul. Pentru a face acest lucru, mișcăm mâna în sus și în jos peste fototranzistor, schimbând cantitatea de lumină ce ajunge la el. Cu cât replicăm mișcărilor pe care ne așteptăm să le folosim în timp ce ”cântăm” la instrument, cu atât calibrarea va fi mai bună.

După 5 secunde, calibrarea va fi completă, iar LED-ul de pe Arduino se va stinge. Când se întâmplă acest lucru, ar trebui să auzim ceva zgomot provenind din piezo! Pe măsură, ce cantitatea de lumină ce cade asupra senzorului se schimbă, la fel se va schimba și frecvența pe care o joacă piezo.

Gama din funcția **map()** care determină tonul este destul de largă. Vom încerca să schimbăm frecvențele pentru a le găsi pe cele potrivite pentru stilul nostru muzical.

Funcția **tone()** funcționează foarte mult ca **PWM** în **analogWrite()**, dar cu o diferență semnificativă. În **analogWrite()** frecvența este fixă. Modificăm raportul impulsurilor în acea perioadă de timp pentru a varia ciclul de funcționare. Cu **tone()** putem trimite în continuare impulsuri, dar schimbând frecvența acestora. Cu **tone()** sunetul pulsează întotdeauna la un ciclu de funcționare de 50% (jumătate din timpul în care pinul este ridicat, cealaltă jumătate din timp este scăzut).

Funcția **tone()** ne oferă posibilitatea de a genera frecvențe diferite atunci când pulsează un difuzor sau un piezo. Când utilizăm senzori într-un circuit divizor de tensiune, probabil că

nu vom obține o gamă completă de valori între 0–1023. Prin calibrarea senzorilor, este posibil să mapăm intrările într-un interval utilizabil.

Concluzii

- În rezultatul unor asemenea activități pot fi rezolvate următoarele probleme:
- formarea competențelor tineretului studios aferente disciplinelor fizica și informatica în domeniul STEAM;
- mărirea interesului față de științele fizica și informatica;
- micșorarea costului utilajelor care este folosit în învățământul STEAM;
- diminuarea abandonării de către potențialii studenți a specialităților aferente disciplinelor fizica și informatica.

Acest articol a fost elaborat în cadrul proiectului de cercetări științifice „Metodologia implementării TIC în procesul de studiere a științelor reale în sistemul de educație din Republica Moldova din perspectiva inter/transdisciplinarității (concept STEAM)”, Programul „Program de stat” (2020- 2023), Prioritatea IV: Provocări societale, cifrul 20.80009.0807.20.

Bibliografie

1. *Arduino* - Introduction. arduino.cc.
2. *Arduino* FAQ – With David Cuartielles. Malmö University. 5 aprilie 2013.