

# SISTEM DE TESTARE A MODULELOR ARDUINO

**AUTORI :** Marilena Florentina IANC<sup>1</sup> ; [florentina.ianc8@gmail.com](mailto:florentina.ianc8@gmail.com)

Nicolae CHIRIAC<sup>2</sup> ; [nicuchiriac710@gmail.com](mailto:nicuchiriac710@gmail.com)

**CORDONATORI :**prof.univ.habil.dr.ing Monica LEBA<sup>3</sup>

drd .ing Remus SIBIȘANU<sup>4</sup>

<sup>1</sup>Univesitatea din Petroșani ; Facultatea de Inginerie Mecanica și Electrica ;specializare: Calculatoare ;anul : 3

<sup>2</sup>Universitatea din Petroșani ; Facultatea de Inginerie Mecanica și Electrica ; specializare Automatica ; anul : 3

<sup>3</sup>Universitatea din Petroșani ; Facultatea de Inginerie Mecanica și Electrica ; Departamentul de Automatica ,Calculatoare, Inginerie Electrica și Energetica.

**Rezumat:** *Testarea modulelor Arduino poate fi realizată folosind o varietate de tehnici, inclusiv teste funcționale pentru a verifica funcționalitatea specifică a modulelor și interacțiunea cu componentele periferice, teste de integrare pentru a asigura că modulele funcționează corect împreună și teste de performanță pentru a evalua timpul de răspuns și eficiența generală a sistemului. Analiza și interpretarea rezultatelor testelor sunt importante pentru identificarea și remedierea eventualelor probleme sau erori în codul sau în funcționalitatea modulelor. Testarea este un aspect crucial al procesului de dezvoltare a proiectelor Arduino și contribuie la asigurarea calității și fiabilității produselor finite. În concluzie, testarea modulelor Arduino este esențială pentru dezvoltarea și implementarea proiectelor cu succes, asigurându-se că acestea îndeplinesc cerințele specifice și că funcționează corect în diferite condiții de utilizare.*

**Cuvinte cheie:** modul , arduino , testare.

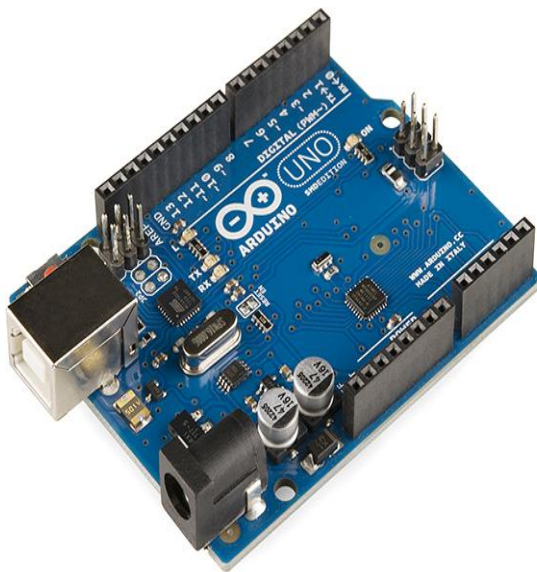
## 1.Introducere

Testarea modulelor electronice este un proces esențial în dezvoltarea și producția electronicelor moderne. Aceasta implică evaluarea și verificarea funcționării, performanței și fiabilității modulelor electronice în diferite condiții de operare. Scopul principal al testării este de a identifica și corecta orice defecte sau probleme care ar putea afecta calitatea sau performanța produsului final. O componentă importantă a testării modulelor electronice este analiza și interpretarea rezultatelor testelor pentru a identifica orice anomalii sau probleme potențiale. În plus, testarea este adesea integrată în întregul proces de dezvoltare a produsului, de la etapele inițiale de proiectare până la producția în masă și livrarea către clienți.

În concluzie, testarea modulelor electronice este un pas critic în asigurarea calității și fiabilității produselor electronice și joacă un rol crucial în satisfacerea cerințelor și așteptărilor utilizatorilor finali.

## 2. Prezentarea componentelor folosite

**2.1** Arduino este o companie [open-source](#) care produce atât plăcuțe de dezvoltare bazate pe [microcontrolere](#), cât și partea de [software](#) destinată funcționării și programării acestora. Pe lângă acestea include și o comunitate uriașă care se ocupă cu creația și distribuirea de proiecte care au ca scop crearea de dispozitive care pot sesiza și controla diverse activități sau procese în lumea reală.



*Fig. 1(Arduino Uno)*

Proiectul este bazat pe designul plăcilor cu microcontroler produse de câțiva furnizori, folosind diverse tipuri de microcontrolere. Aceste plăci pun la dispoziția utilizatorului pini I/O, digitali și analogici, care pot fi interfațați cu o gamă largă de plăcuțe numite scuturi (shield-uri) și/sau cu alte module. Plăcile au interfețe de comunicații seriale, inclusiv [USB](#) pe unele modele, pentru a încărca programe din [calculatoarele personale](#). Pentru programarea microcontrolerelor, Arduino vine cu un [mediu de dezvoltare integrat](#) (IDE) bazat pe proiectul [Processing](#), care include suport pentru [limbaje de programare](#) ca [C](#) și [C++](#).

Primul Arduino a fost lansat în 2005, având ca țintă asigurarea unei soluții ieftine și simple pentru începători și profesioniști spre a crea dispozitive capabile să interacționeze cu mediul, folosind senzori și sisteme de acționare. Cele mai comune exemple sunt dispozitivele pentru utilizatorii începători precum: roboții simpli, termostatele și/sau detectoarele de mișcare. O plăcuță Arduino este compusă dintr-un microcontroler [Atmel AVR](#) de 8-, 16- sau 32-biți (deși începând cu 2015 s-au folosit microcontrolere de la alți producători) cu componente complementare care facilitează programarea și încorporarea în alte module. Un aspect important la Arduino este că acesta dispune de conectori standard, care permit utilizatorului să conecteze plăcuța cu procesorul la diferite module interschimbabile numite shield-uri. Unele shield-uri comunică cu Arduino direct

prin pinii digitali sau analogici, dar altele sunt adresabile individual prin magistrala serială **I<sup>2</sup>C** permițând utilizarea mai multor module în paralel.

## 2.2 Analiza lexicala

Analiza lexicala este conversia unui text în *simboluri lexicale* semnificative (semantic sau sintactic) aparținând unor categorii definite de un program „lexer”. În cazul unui limbaj natural, acele categorii includ substantive, verbe, adjective, punctuații etc. În cazul unui limbaj de programare, categoriile includ identificatori, operatori, simboluri de grupare și **tipuri de date** . Analizarea lexicală este legată de tipul de analiza utilizat în **Modelele de limbaj mari** (LLM), dar cu două diferențe. În primul rând, analiza lexicală se bazează de obicei pe o **gramatică lexicală** , în timp ce analizele LLM sunt de obicei bazate pe **probabilități** . În al doilea rând, analizele LLM efectuează un al doilea pas care convertește jetoanele în valori numerice.

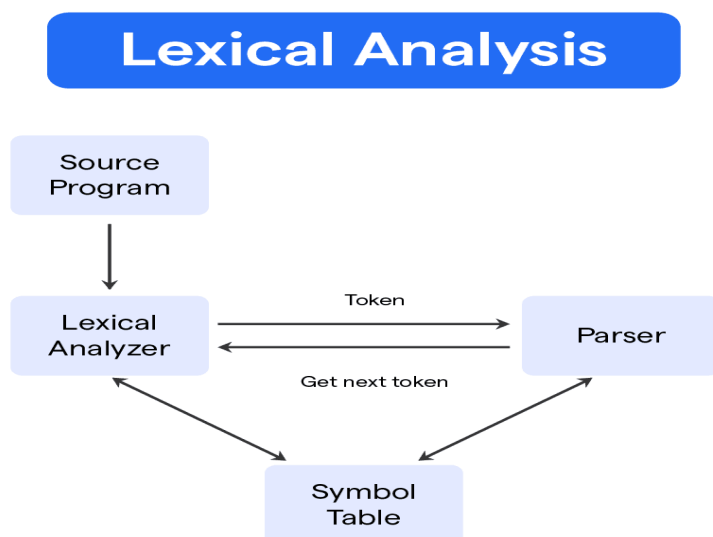


Fig. 2 (Analiza lexicala)

Lexeri și analizatorii sunt cel mai adesea folosiți pentru compilatoare, dar pot fi folosiți pentru alte instrumente de limbaj informatic, cum ar fi **prettyprinters** sau **linters** . Lexarea poate fi împărțită în două etape: *scanarea* , care segmentează șirul de intrare în unități sintactice numite *lexeme* și le clasifică în clase de jetoane; și *evaluarea* , care convertește lexemele în valori procesate.

Lexerii sunt, în general, destul de simpli, cu cea mai mare parte a complexității deferite fazelor de **analiză sintactică** sau **analiză semantică** și pot fi adesea generați de un **generator de lexer** , în special **lex** sau derivate. Cu toate acestea, lexerii pot include uneori o anumită complexitate, cum ar fi procesarea **structurii frazei** pentru a ușura introducerea și a simplifica analizatorul și pot fi scrise parțial sau integral de mână, fie pentru a sprijini mai multe caracteristici, fie pentru performanță.

## 2.3 Firmata

Firmata este un protocol pentru comunicarea cu microcontrolere din software-ul de pe un computer gazdă. Protocolul poate fi implementat în firmware pe orice arhitectură de microcontroler , precum și în software pe orice pachet de software pentru computerul gazdă. Depozitul Arduino descris aici este o bibliotecă Firmata pentru dispozitive Arduino .Există două modele principale de utilizare a Firmata. Într-un model, autorul schiței Arduino folosește diferitele metode oferite de biblioteca Firmata pentru a trimite și a primi în mod selectiv date între dispozitivul Arduino și software-ul care rulează pe computerul gazdă. De exemplu, un utilizator poate trimite date analogice către gazdă folosind `Firmata.sendAnalog(analogPin, analogRead(analogPin))` sau poate trimite date împachetate într-un șir folosind `Firmata.sendString(stringToSend)`.

### 3.Descrierea lucrării

Un sistem de testare a modulelor Arduino este o configurație hardware și software specializată pentru evaluarea și verificarea funcționării modulelor și a proiectelor bazate pe platforma Arduino. Acest sistem este conceput pentru a efectua teste funcționale și de performanță pentru modulele individuale și pentru întregul sistem integrat.

Componentele principale ale unui sistem de testare a modulelor Arduino pot include:

Interfețe hardware: Aparatură și conexiuni necesare pentru a interacționa cu modulele Arduino și cu componentele periferice, cum ar fi senzori, actuatori sau alte dispozitive.

Software de testare: Aplicații și scripturi software care controlează și monitorizează modulele Arduino în timpul testelor, precum și analizează rezultatele pentru a identifica problemele sau erorile.

Senzori și echipamente de măsurare: Dispozitive care măsoară diferiți parametri, cum ar fi temperatură, umiditate, lumină, sau alte variabile relevante pentru proiectul sau aplicația testată.

Sursă de alimentare și echipament de siguranță: Asigură alimentarea corectă și sigură a modulelor și a altor componente ale sistemului de testare.

Configurație de testare automată (opțională): Un sistem automatizat care poate executa teste repetate și standardizate asupra modulelor Arduino fără intervenție umană, facilitând testarea de mare volum și asigurând consistența rezultatelor.

### 4. Proiectare software

```
#include <Servo.h>
```

```

char buff;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    Serial.println("Sistem de testare a modulelor electronice Arduino");
    meniu();
}

void meniu(){
    Serial.println("1.Serial echo");
    Serial.println("2.Blink");
}

void check(){
    if(Serial.available()>0){
        buff=Serial.read();
        if(buff=='1'){
            for(int thisByte=33;thisByte<=126;thisByte++){
                Serial.write(thisByte);
                Serial.print(", dec: ");
                Serial.print(thisByte);
                Serial.print(", hex: ");
                Serial.print(thisByte, HEX);
                Serial.print(", oct: ");
                Serial.print(thisByte, OCT);
                Serial.print(", bin: ");
                Serial.println(thisByte, BIN);
            }
            Serial.println("=====");
            meniu();
        }
    }
}

```

```
}  
  
if(buff=='2'){  
    pinMode(13, OUTPUT);  
    for(int i=0;i<5;i++){  
        digitalWrite(13,HIGH);  
        delay(1000);  
        digitalWrite(13,LOW);  
        delay(1000);  
    }  
    Serial.println("TASK: Blink");  
    Serial.println("=====");  
    meniu();  
}  
  
}  
  
}  
  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    check();  
}
```