

**Mihai Todica**

**Experimente cu Arduino,  
Processing și Soundcard Scope**

**PRESA UNIVERSITARĂ CLUJEANĂ**

**2025**

## Cuprins

<b>Introducere.....</b>	<b>9</b>
<b>Capitolul 1. Platforma <i>Arduino</i> și aplicații.....</b>	<b>11</b>
1.1. Instalarea platformei <i>Arduino</i> .....	11
1.2. Instalarea librărilor.....	12
1.3. Conectarea unui afișaj <i>LCD</i> sau <i>Oled</i> .....	14
1.4. Senzori și actuatori.....	15
1.4.1. Senzori și date analogice. Măsurarea unei tensiuni continue.....	16
1.4.2. Senzori și date digitale. Măsurarea temperaturii și umidității.....	17
1.4.3. Senzori neconvenționali. Celula fotovoltaică.....	19
1.5. Actuatori.....	21
Bibliografie.....	23
<b>Capitolul 2. Platforma <i>Processing</i> și aplicații.....</b>	<b>25</b>
2.1. Instalarea programului <i>Processing</i> .....	25
2.2. Facilitatea <i>mouse</i> a calculatorului și <i>Processing</i> .....	26
2.2.1. Senzor de mișcare cu <i>mouse</i> . Citirea coordonatelor în planul $xOy$ .....	27
2.2.2. Afișarea traiectoriei unui <i>mouse</i> .....	31
2.2.3. Reprezentarea coordonatelor în funcție de timp.....	32
2.2.4. Mișcarea pe direcția $Ox$ cu reactualizarea ecranului.....	35
2.2.5. Mișcarea pe direcția $Oy$ cu reactualizarea ecranului.....	37
2.2.6. Deplasări înregistrate cu <i>scroll</i> .....	39
2.2.7. Studiul mișcărilor cu frecare folosind facilitatea <i>scroll</i> .....	41
2.2.8. Studiul oscilațiilor amortizate.....	42
2.3. Facilități oferite de camera <i>web</i> .....	45
2.3.1. Activarea camerei <i>web</i> .....	45
2.3.2. Monitorizarea intensității luminoase a unui pixel în timp real.....	46
2.3.4. Încărcarea unei imagini statice.....	47
2.2.5. Vizualizarea oscilațiilor unui pendul matematic.....	49
Bibliografie.....	51
<b>Capitolul 3. Experimente cu programul <i>Soundcard Scope</i>.....</b>	<b>53</b>
3.1. Determinarea lungimii de undă și a vitezei de propagare a sunetelor.....	53
3.2. Interferența sunetelor.....	57

3.3. Analiza spectrală a unui sunet.....	59
3.4. Studiul circuitelor de curent alternativ. Circuitul RC serie .....	61
3.5. Vizualizare impulsurilor unei telecomenzi IR.....	64
3.6. Evidențierea câmpurilor magnetice de autoinducție generate de un motor de curent continuu (DC).....	66
3.7. Vizualizarea tensiunilor de comandă ale unui motor de curent continuu fără perii colectoare, (BLDC).....	69
3.8. Generarea tensiunilor alternative cu un motor pas cu pas.....	71
3.9. Măsurarea vitezei de rotație a unui corp .....	73
Bibliografie .....	75
<b>Capitolul 4. Experimente combinate Arduino-Processing .....</b>	<b>77</b>
4.1. Comunicarea Arduino-Processing .....	77
4.1.1 Transmisia unei singure variabile Arduino-Processing.....	77
4.1.2. Transmiterea a două variabile Arduino-Processing.....	79
4.1.3. Comunicarea Processing-Arduino .....	81
4.2. Verificarea legii lui Ohm pentru o porțiune de circuit .....	83
4.3. Caracteristica volt-amperică a unui LED .....	86
4.4. Studiul puterilor într-un circuit electric simplu.....	88
4.5. Fenomene tranzitorii în circuite RC .....	93
Bibliografie .....	98
<b>Capitolul 5. Sisteme de comunicație bilaterală.....</b>	<b>99</b>
5.1. Transceiverul Si 1278.....	99
5.2. Date și comenzi unilaterale. Date analogice .....	100
5.3. Transmiterea datelor multiple .....	102
5.4. Legături duplex și feedback .....	105
Bibliografie .....	108
<b>Capitolul 6. Module cu ecran tactil pentru comenzi și transmisii de date ....</b>	<b>109</b>
6.1. Setarea platformei IDE pentru ESP32 CYD.....	109
6.2. Măsurarea temperaturii și umidității cu ESP32 CYD și DHT 11 .....	110
6.3. Comenzi senzoriale cu ESP32 CYD. Comanda unui led RGB și a unui servomotor .....	111
6.4. Voltmetru cu ESP32 CYD .....	113
6.5. Conexiuni pereche (P2P) cu ESP32 CYD.....	115
6.5.1. Transmiterea datelor digitale .....	115
6.5.2. Transmiterea datelor analogice.....	117

6.6. Folosirea ecranului tactil. Telecomandă senzorială unidirecțională.....	118
6.7. Telecomandă senzorială cu feedback .....	120
Bibliografie .....	122
<b>Capitolul 7. Date și comenzi prin rețele locale și web cu aplicații speciale....</b>	<b>123</b>
7.1. Accesarea platformei <i>RemoteXY</i> .....	123
7.2. Modulul <i>ESP32</i> setări și instalare .....	125
7.3. Comunicații prin <i>Bluetooth</i> .....	126
7.3.1. Comanda unu led <i>RGB</i> prin <i>Bluetooth</i> .....	127
7.3.2. Voltmetru analogic cu transmiterea datelor prin <i>Bluetooth</i> .....	129
7.4. Comunicații prin internet .....	130
7.4.1. Transmiterea datelor de temperatură și umiditate prin internet .....	132
7.4.2. Comunicații bilaterale și <i>feedback</i> .....	133
7.4.3. Comunicații cu <i>ESP32 CYD</i> .....	134
Bibliografie .....	135

## Capitolul 1.

# Platforma *Arduino* și aplicații

Această platformă a fost concepută pentru a controla anumite elemente fizice externe, *senzori*, în vederea culegerii unor date numerice, analiza lor, luarea unor decizii și apoi comanda unor dispozitive executante, *actuatori*, pe baza deciziilor luate. Conține două părți, o parte *hardware* reprezentată de modulele electronice folosite și o parte *software* ce conține *codurile* de control ale acestora. Există mai multe variante de module folosite, *Arduino UNO*, *MEGA*, *NANO*, etc., precum și o serie de periferice, senzori și actuatori compatibile cu acestea, (Fig. 1.1).

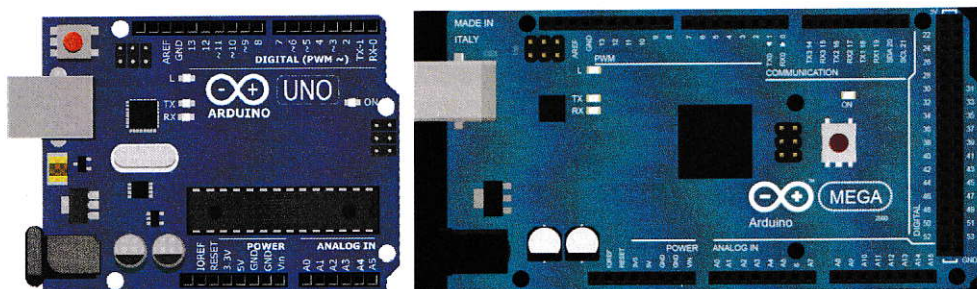


Fig. 1.1. Imaginea modulelor *Arduino UNO* și *MEGA*.

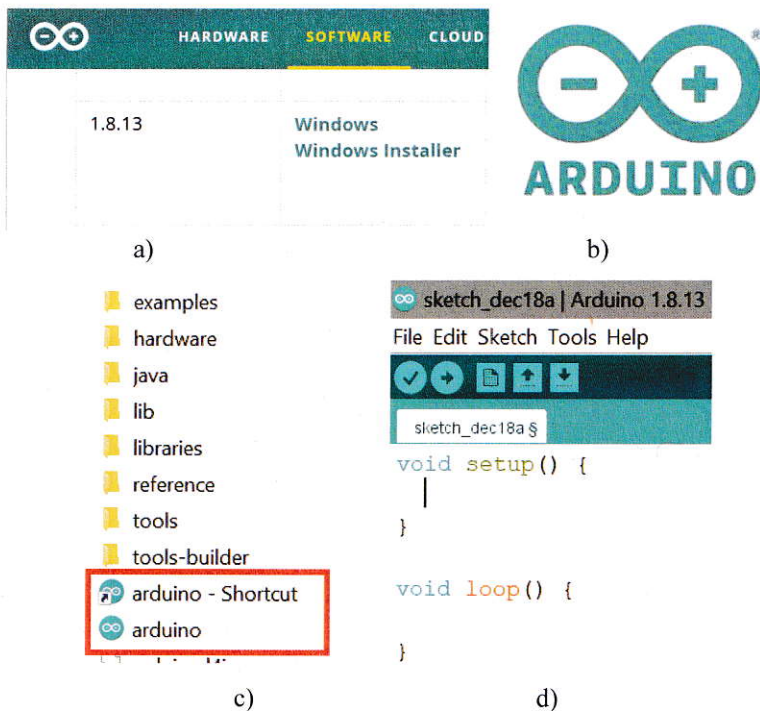
Odată editate codurile pot fi încărcate în oricare din modulele hardware compatibile, obținându-se un sistem programabil inteligent, gata de utilizare. Dar pentru utilizare sunt necesare parcurgerea câtorva etape, legate atât de partea de programare, cât și de partea hardware, etape care sunt general valabile pentru toate aplicațiile prezentate în continuare.

## 1.1. Instalarea platformei *Arduino*

Prima etapă în utilizarea platformei *Arduino* presupune descărcarea și instalarea ei pe un calculator. Aceasta se va descărca de pe internet folosind linkul din referință, [1], (Fig. 1.2.a). Arhiva *Arduino* se salvează într-un dosar, (*folder*), nou creat în spațiul de lucru al utilizatorului și se dezarchivează. Apar mai multe fișiere, (*files*), după care se deschide fișierul *exe*, ceea ce va iniția automat instalarea programului, (Fig. 1.2.c). Pentru instalarea corectă se vor urmări instrucțiunile afișate în ferestrele de dialog. Dacă totul decurge bine pe *desktop* va apărea *sigla* aplicației, (Fig. 1.2.b). O apăsare dublă pe aceasta va deschide aplicația, (Fig. 1.2.d). De aici pot fi apelate *coduri* gata finalizate sau se pot demara proiecte noi.

Orice nou proiect conține cel puțin două secțiuni, *void setup()* și *void loop()*. Prima secțiune include instrucțiuni care se execută o singură dată, cum ar fi

definirea unor variabile, apelarea unor librării, setarea unor parametri inițiali etc., iar a doua secțiune conține instrucțiunile care sunt rulate periodic, în buclă închisă, ce execută programului propriu zis.



**Fig. 1.2.** a) Pagina oficială de descărcare a programului Arduino.  
 b) Sigla aplicației. c) Fișierele conținute în arhiva Arduino.  
 d) Pagina de lucru afișată la deschiderea programului.

Pentru familiarizarea cu programul este indicată rularea unor exemple simple conținute în folderul *examples* din program, dar rularea lor efectivă implică conectarea unei plăci Arduino la computer, așa cum va fi prezentat în proiectele incluse în lucrare.

## 1.2. Instalarea librărilor

Pentru operarea majorității perifericelor conectate la placa *Arduino* codul *IDE* trebuie să facă apel la *subrutine* special concepute numite *librării*. Anumite librării sunt conținute în folderul *libraries* al pachetului inițial de instalare al platformei *Arduino*, dar altele trebuie introduse manual. Procesul implică mai multe etape.

În primul rând librăria dorită trebuie descărcată de pe un site specializat, de exemplu *github.com*, salvată pe calculator într-un folder particular, de exemplu *Librării descărcate* și apoi dezarhivată, [2]. Folderul dezarhivat, cu numele specific

librării descărcate, de exemplu *Adafruit GFX*, se introduce în folderul *libraries* al platformei *Arduino*. De obicei acesta se găsește în partiția *C* a hard discului, *Users / nume user / Documents / Arduino / libraries*, (Fig. 1.3).

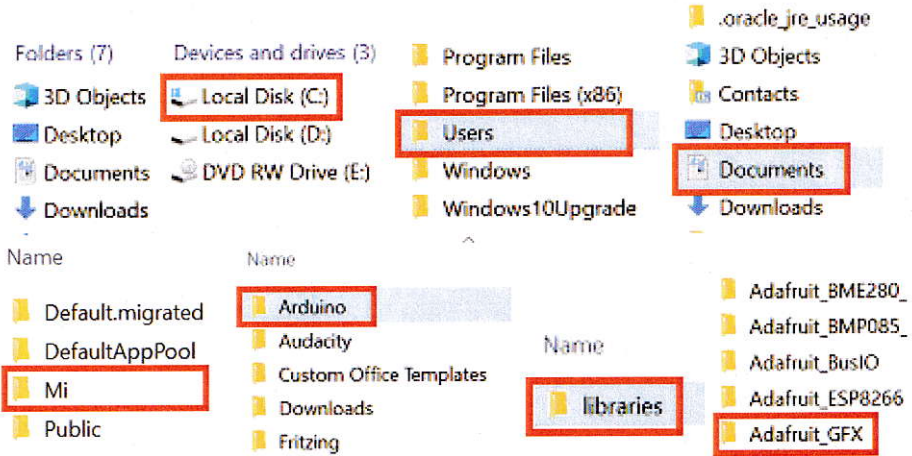


Fig. 1.3. Instalarea librărilor.

După instalarea librărilor dorite se deschide din nou platforma *Arduino* și programul este gata de lucru.

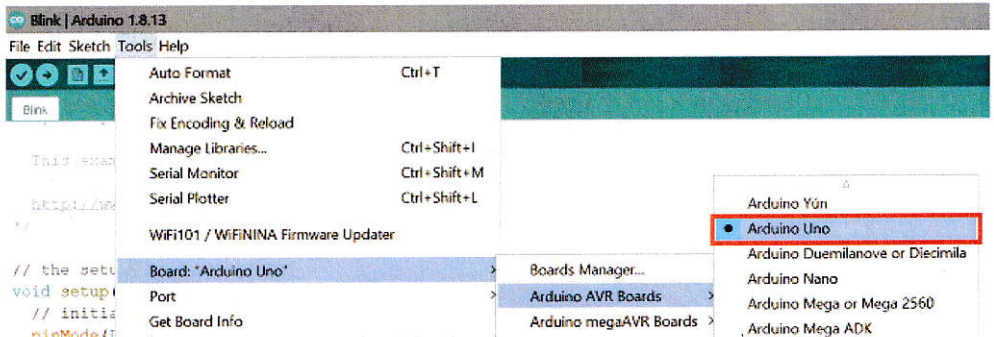


Fig. 1.4. Lansarea platformei *Arduino*.

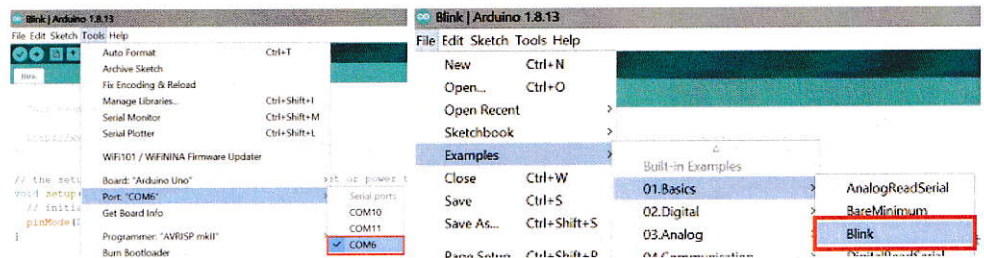


Fig. 1.5. Setarea platformei *Arduino*.

Pentru verificare se conectează placa *Arduino UNO* la computer, se deschide platforma *Arduino IDE*, se selectează placa utilizată, în acest caz *Arduino UNO*, se selectează portul de comunicare cu placa, *COM 6* de exemplu, se deschide folderul *examples*, se selectează *Blink*, se încarcă codul și apoi se rulează. Ledul de pe placa începe să clipească. Procedura este exemplificată prin figurile 1.4 și 1.5.

### 1.3. Conectarea unui afișaj LCD sau Oled

În numeroase aplicații este utilă afișarea locală a unor informații fără a mai face uz de calculator. Acest lucru este posibil prin conectarea unui afișaj *LCD* (*Liquid Crystal Display*), sau *Oled* plăcii *Arduino*, [3, 4]. Există mai multe tipuri de afișoare dar aici am optat pentru modulul *LCD 2x16*, conținut în majoritatea kiturilor comerciale *Arduino*, care poate afișa 16 caractere pe 2 rânduri. Se poate conecta la placa *Arduino* fie în modul *SPI*, (*Serial Peripheral Interface*), fie în modul *I2C*. În primul caz conexiunea este mai complicată implicând folosirea mai multor conductori de legătură, dar comunicația cu placa *Arduino* este mai rapidă și nu mai necesită alte module intermediare. În cazul *I2C* comunicația este puțin mai lentă, dar conectarea se face mult mai simplu, doar cu 4 conductori, în schimb este necesară folosirea unui adaptor, (Fig. 1.6). În cele ce urmează vom folosi a doua variantă pentru simplitate ei, dar codurile rămân valabile și pentru conexiunea directă. Modulele *Arduino* au pini dedicați comunicațiilor *I2C* care nu trebuie definiți în cod. Aceștia sunt *A4* și *A5* pentru *Arduino UNO*. Conectarea modulului *LCD* la *Arduino UNO* se face conform tabelului 1.1 și a figurii 1.6.

Tabel 1.1.

Adaptor I2C	SCL	SDA	Vcc	GND
Arduino UNO	A5	A4	+5V	GND

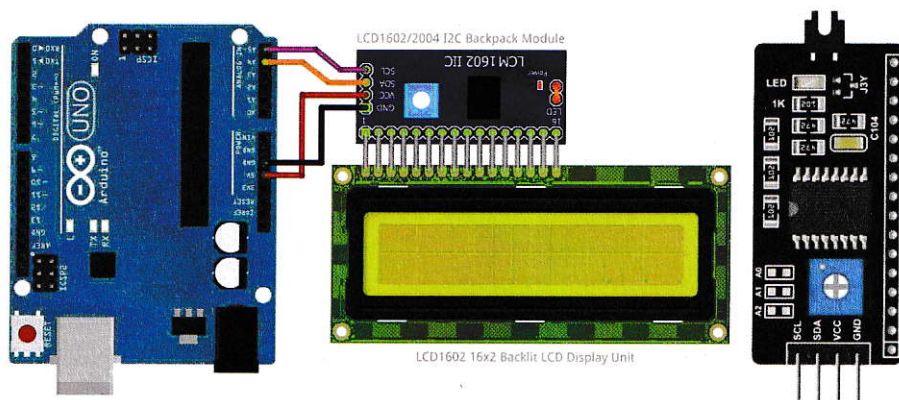
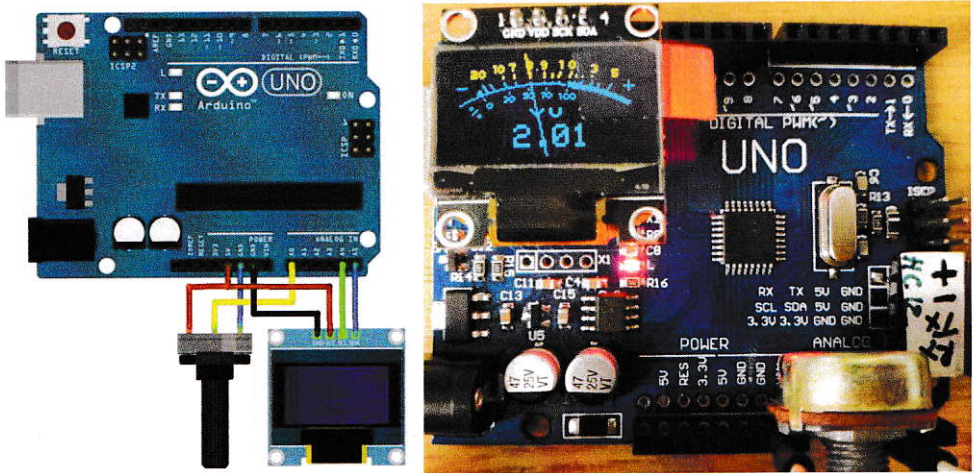


Fig. 1.6. Conectarea afișajului LCD la placa *Arduino UNO* folosind adaptorul *I2C* și imaginea acestuia.

Pentru activarea afișajului este necesară instalarea și folosirea librăriei *LiquidCrystal*, [5].



*Fig. 1.7. Conectarea afișajului Oled 0.96" la placa Arduino UNO și realizarea practică a circuitului.*

O altă variantă constă în utilizarea unui afișaj *Oled 0.96"*, care, deși are dimensiuni mai mici, poate afișa mai multe caractere, având o rezoluție de 128 x 64 pixeli. Trebuie avut în vedere că acest afișaj lucrează cu semnale de +3.3V, dar poate fi alimentat și cu +5V deoarece are integrat un stabilizator de tensiune. Se conectează la placa Arduino UNO conform figurii 1.7 și a tabelului 1.2.

*Tabel 1.2.*

Oled 0.96"	SCL	SDA	Vcc	GND
Arduino UNO	A5	A4	+5V	GND

Pentru activarea afișajului sunt necesare librăriile *Adafruit-GFX* și *Adafruit\_SSD1306*, [6, 7].

## 1.4. Senzori și actuatori

Pentru a putea fructifica facilități unui PC în experimentul de fizică trebuie să folosim elemente adecvate de culegere a datelor asociate unui fenomen și care să le transpună într-un format digital, compatibil cu un computer. Culegerea datelor se face cu dispozitive speciale, *senzorii*, care transformă un parametru fizic, de exemplu presiunea, temperatura, intensitatea luminoasă, etc., într-un semnal electric, singura formă care poate fi gestionată de computer. La fel, pentru

materializarea deciziilor luate de computer este necesară utilizarea unor dispozitive care să transforme comanda primită în acțiuni fizice concrete, așa numiții *actuatori*. Atât senzorii cât și actuatorii nu pot fi atașați direct unui PC obișnuit decât prin intermediul unei plăci adecvate, în cazul nostru *Arduino*, care poate prelua informațiile primite de la senzori și poate lua singură deciziile adecvate pe baza unui program prestabilit, dar care poate asigura și legătura cu PC-ul. În cele ce urmează vom prezenta câteva exemple de senzori și actuatori compatibili cu placa *Arduino*.

### 1.4.1. Senzori și date analogice. Măsurarea unei tensiuni continue

Există două tipuri de senzori, *analogici* și *digitali*. Primii furnizează la ieșire o tensiune electrică proporțională cu valoarea parametrului fizic măsurat, iar cei digitali o succesiune de impulsuri electrice, *zero* sau *unu*, care reprezintă un număr binar asociat mărimii fizice investigate. Aceste date pot fi procesate direct de placa *Arduino*, în schimb cele analogice trebuie mai întâi transpuse în *cod binar* prin intermediul unui *convertor analog/digital*. Placa *Arduino* conține un astfel de convertor pe 10 biți, ceea ce înseamnă că un semnal de intrare este eșantionat în maximum 1024 de trepte, fiecareia dintre ele asociindu-se un număr binar cuprins între 000000000 și 111111111. [8]. Înălțimea unui *pas* de eșantionare, exprimat în volți, este de  $5/1024$ , adică aproximativ 0.00488 V. Aceste valori rezultă din faptul că valoarea maximă a tensiunii suportate de convertorul *A/D* este de 5 V. Convertorul este conectat la cele 5 intrări analogice ale plăcii numerotate cu A0, A1, A4. Vom prezenta câțiva astfel de senzori și modul lor de gestionare. Vom începe cu senzorii analogici.

*Potențiometrul*, poate fi considerat cel mai simplu senzor analogic care permite aplicarea unei tensiuni electrice variabile plăcii *Arduino*. Datorită limitărilor de protecție impuse acestei plăci, tensiunile aplicate trebuie să fie continue, cu polaritatea negativă la masă și mai mici de 5V, dar se pot aplica și tensiuni mai mari dacă se folosește un divizor rezistiv. Vom prezenta un caz simplu de măsurare a unei tensiuni continue și afișarea ei pe un display *LCD* sau *Oled*. Folosim circuitele din figurile 1.6 și 1.7 la care am conectat în plus un potențiometrul de 10 K $\Omega$  la GND, +5V și intrarea A0 a plăcii. Vom prezenta două situații, prima în care valoarea tensiunii măsurate va fi exprimată direct în număr de pași de eșantionare, iar în al doilea caz, exprimată direct, în cod zecimal. În primul caz nu este necesară nici operație matematică, tensiunea măsurată fiind afișată direct în pași de eșantionare, de exemplu 1023 pentru 5V, 512 pentru 2.5 V, etc. În al doilea caz este necesară o operațiune matematică de scalare. Folosim codul:

```
Uno_LCD_Analog_Voltage
```

pentru afișarea cu LCD și codul:

```
Uno_Oled_Analog_Voltage
```