

Sisteme cu microprocesoare - Manipularea intrărilor analogice -

I. SCOPUL LUCRĂRII:

Lucrarea de laborator are ca scop:

- prezentarea modului de funcționare al convertorului analog – digital sau numeric ^{[1] [2]}
- prezentarea modului de funcționare al convertorului numeric sau digital – analog ^[2]
- prezentarea modului de funcționare al unității pentru generare a impulsurilor ^{[1] [2]}
- implementare aplicațiilor specifice procesării semnalelor analogice ^{[1] [2] [3]}

II. INTRODUCERE:

Microcontrolerul ATmega 328P ^[1] din cadrul platformei de dezvoltare Arduino Nano, conferă posibilitatea achiziționării și prelucrării atât a semnalelor digitale cât și a **semnalelor analogice**. Terminalele marcate pe cablajul imprimat cu indicativul „Ax” (unde „x” reprezintă numărul de ordine), reprezintă un set de **intrări analogice** (Fig.1).

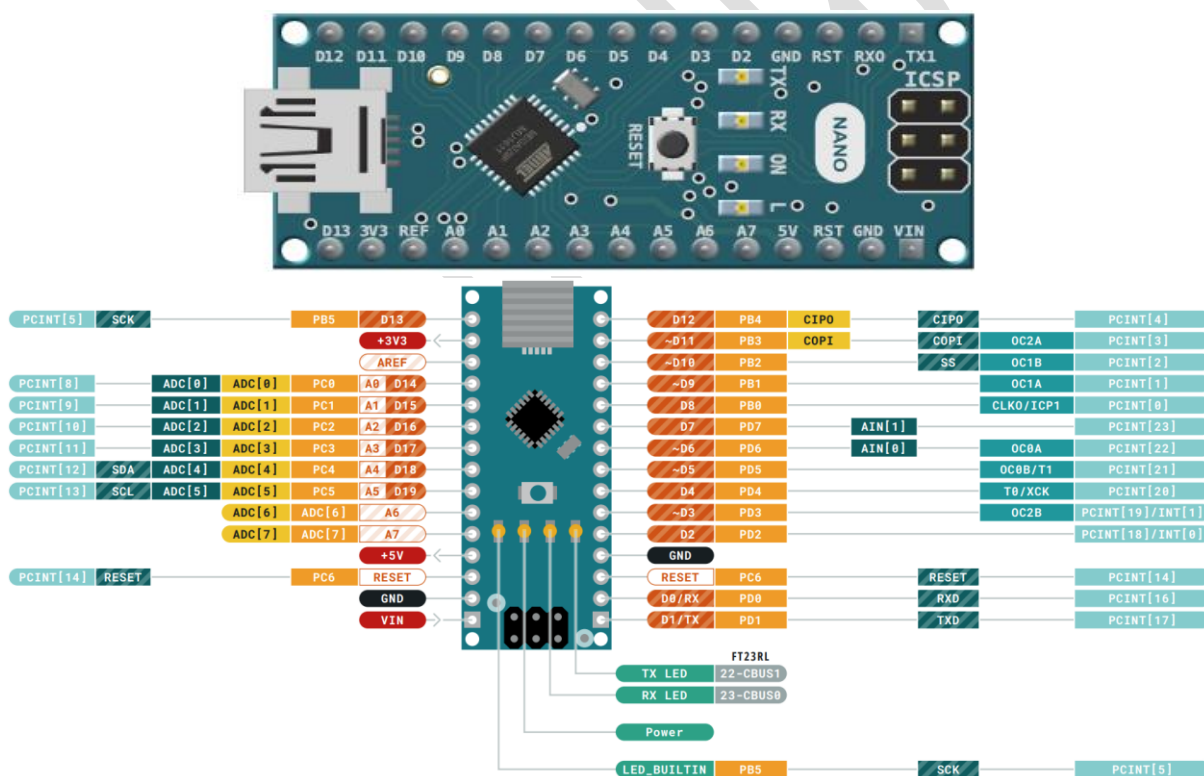


Fig. 1 – Harta funcțiilor specifice terminalelor platformei Arduino Nano ^[4]

Etajul din cadrul arhitecturii care permite achiziționarea semnalelor analogice, poartă denumirea de convertor analog – digital (eng. Analog to Digital Converter – ADC). Prin intermediul acestuia, microcontrolerul ATmega 328P poate măsura semnale de tensiune variabile în timp cu amplitudinea cuprinsă între 0 și 5 [V].

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

III. ASPECTE TEORETICE:

Într-un circuit electronic, semnalele analogice sunt reprezentate ca și o tensiune cu amplitudine variabilă în timp (Fig. 2).

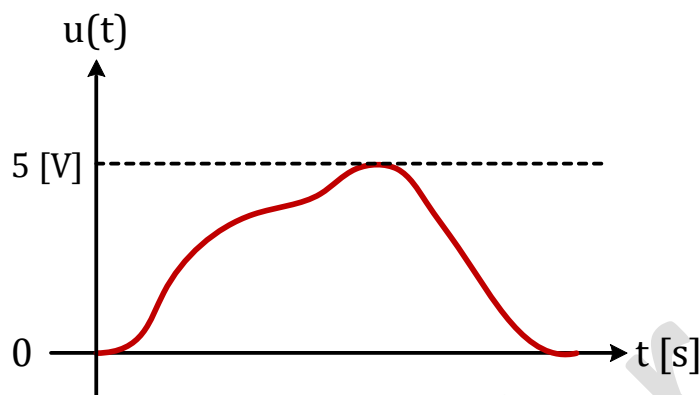


Fig. 2 – Semnal analogic [3]

Semnalele de tensiune variabile în timp provin în mare parte de la traductoare. Rolul unui traductor este de a transforma o mărime de natură ne-electrică într-o mărime de natură electrică precum tensiune sau curent cu amplitudine variabilă. Există și situația în care, mărimea ne-electrică măsurată de traductor, este proporțională cu un anumit parametru de circuit precum: rezistența electrică, capacitatea, inductivitatea sau impedanța echivalentă. Prin urmare, există deci, două clase de traductoare:

- traductoare active (care nu necesită alimentare deoarece produc tensiune sau curent);
- traductoare pasive (care necesită alimentare deoarece se bazează pe variația unui parametru de circuit precum rezistența, capacitatea sau inductivitatea);

În vederea adaptării semnalului de măsură la sistemul de achiziție există de asemenea două categorii de circuite sau etaje intermediare:

- circuite formatoare de semnal pentru traductoarele active (ex. amplificator);
- circuite adaptoare de impedanță pentru traductoarele pasive (ex. filtre pasive);

Cel mai simplu circuit de adaptare a impedanței și atenuare a semnalului este **divizorul de tensiune** (Fig. 3).

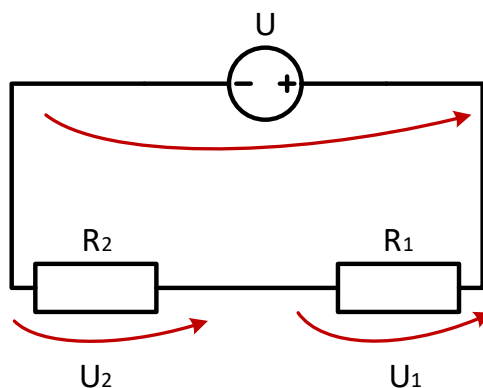


Fig. 3 – Divizorul de tensiune [3]

Există următoarele relații de calcul pentru tensiune (în cazul divizorului):

$$U_1 = R_1 \cdot \frac{U}{R_1 + R_2} \quad U_2 = R_2 \cdot \frac{U}{R_1 + R_2}$$

Traductoarele pasive pot fi atașate la microcontroler prin intermediul divizorului de tensiune având o rezistență variabilă (ex. montaj în semi-punte sau potențiometric). Rezistența variabilă reprezintă însuși traductorul (ex. termistor) (Fig. 4).

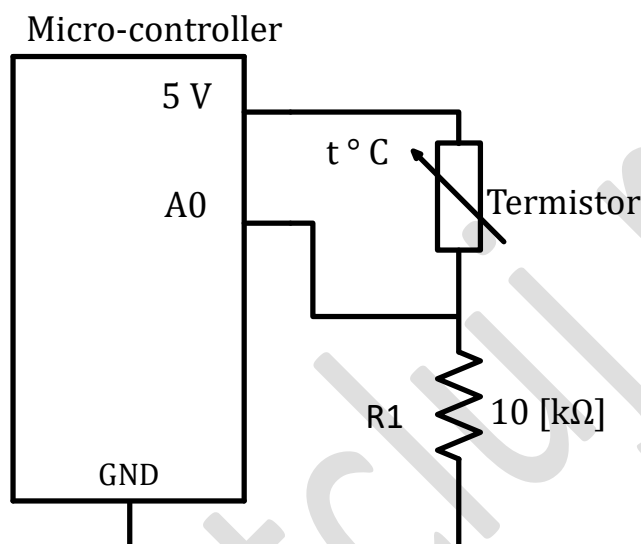


Fig. 4 – Atașarea traductoarelor pasiv-rezistive la microcontroler [3]

Cel mai simplu etaj electronic din componența arhitecturii microcontrolerului ATmega 328P, care poate prelucra semnalul analogic este **comparatorul** [1] (Fig. 5).

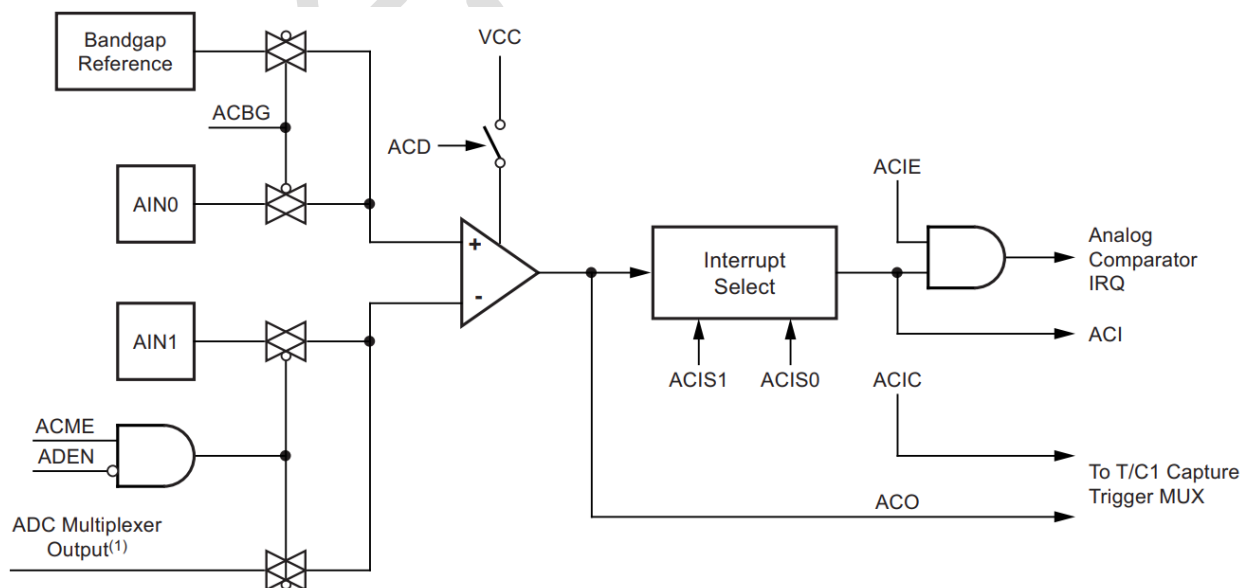


Fig. 5 – Comparatorul din cadrul arhitecturii microcontrolerului ATmega 328P [1]

Comparatorul electronic are rolul de a „sesiza” diferența de potențial dintre tensiunea de referință și semnalul analogic măsurat „ $u(t)$ ”. Tensiunea de referință „ref” poate fi „0”, diferită de „0” sau egală cu tensiunea de alimentare „ V_{cc} ”. Când tensiunea măsurată devine egală cu tensiunea de referință, comparatorul produce un semnal digital la ieșire cu nivelul 1 logic (adică tensiunea de alimentare se va regăsi la ieșire) (Fig. 6).

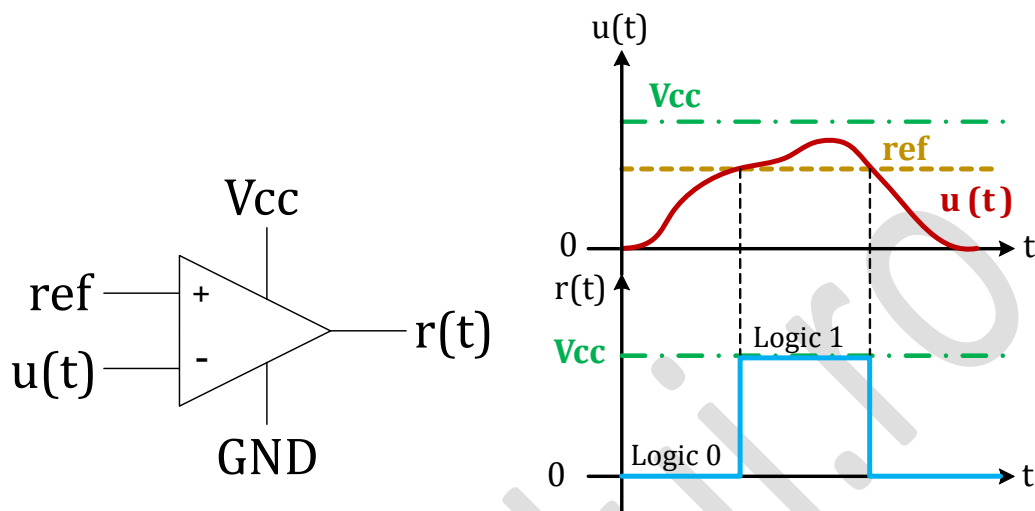


Fig. 6 – Principiul de funcționare al comparatorului electronic [3]

Convertorul analog – digital (eng. Analog to Digital Converter – ADC) reprezintă o serie progresivă de comparatoare. Numărul de comparatoare redă rezoluția convertorului (ex. 4 comparatoare = 2^4 combinații posibile). În componența convertorului analog – digital există un divizor rezistiv cu un număr finit de rezistențe egal cu numărul de comparatoare. Fiecare comparator verifică nivelul de tensiune corespunzător căderii la bornele rezistenței corespunzătoare (Fig. 7).

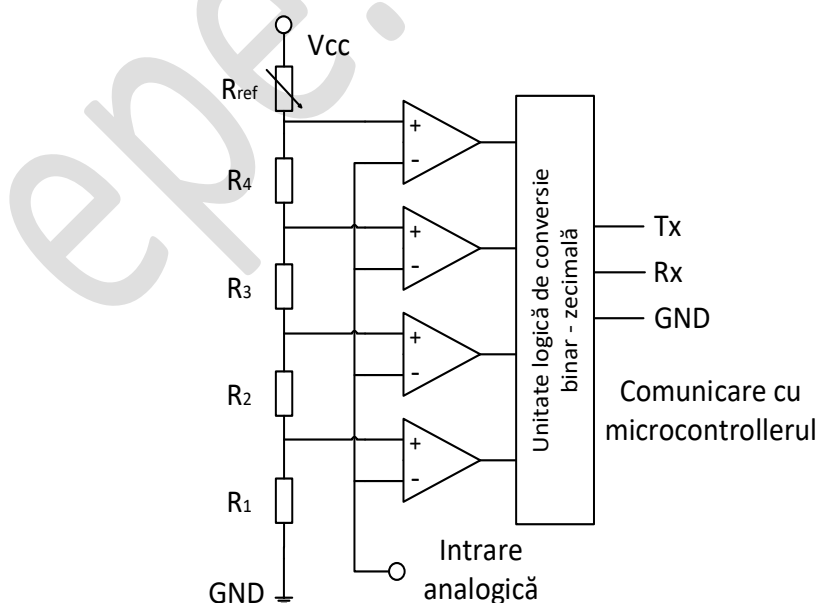


Fig. 7 – Schema electronică principală a convertorului analog digital [3]

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Semnalul rezultat „r(t)” înregistrat în memoria microcontrolerului de către convertorul analog – digital, va fi de natură digitală, anume un semnal discretizat (construit prin variația în trepte a amplitudinii) (Fig. 8).

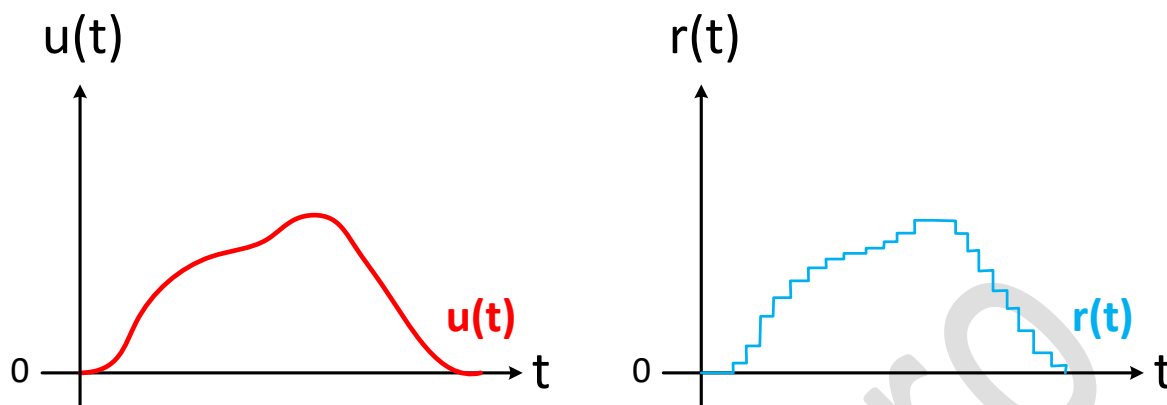


Fig. 8 – Semnalul analogic discretizat [3]

Fidelitatea semnalului înregistrat depinde de **rezoluția convertorului** analog – digital. Rezoluția depinde de **numărul de comparatoare** din componența convertorului analog – digital. **Precizia** convertorului analog – digital depinde de **rezoluție** și **tensiunea de alimentare**. Spre exemplu:

Fie, un convertor analog – digital având rezoluția pe 10 biți, alimentat cu tensiunea de referință 5 [V]. Să se calculeze precizia convertorului:

$$U_{ref} = 5 [V], \quad r = 10 [bit] = (2^{10}) - 1 = 1023$$

$$x = \frac{U_{ref}}{r} = \frac{5}{1023} \approx 0,004 [V] \rightarrow 4 [mV / treaptă];$$

Unde, „U_{ref}” reprezintă tensiunea de referință, „r” rezoluția, iar „x” precizia convertorului analog – digital. **Precizia** reprezintă **constanta de calibrare în tensiune** a convertorului.

Fie „x”, precizia unui convertor analog – digital, „r” rezoluția egală cu 10 biți, iar tensiunea de referință 5 [V]. Să se determine **nivelul de tensiune măsurat cu ajutorul convertorului analog - digital**, pentru care **indicația numerică** în format zecimal din registrul convertorului ar fi ADC = 345:

$$U_{m\grave{a}s} = ADC \cdot x = ADC \cdot \frac{U_{ref}}{r}$$

$$ADC = 345, \quad U_{m\grave{a}s} = ADC \cdot x = ADC \cdot \frac{U_{ref}}{r} = 345 \cdot \frac{5}{1023} \approx 1,38 [V]$$

Unde, „ADC” reprezintă indicația numerică din registrul convertorului analog - digital, iar „U_{măș}” reprezintă tensiunea măsurată pe intrarea analogică.

Circuitul care realizează **funcția inversă** conversiei analog – digitale (numerice), este **convertorul digital – analog** (eng. Digital to Analog Converter – DAC). Rolul acestui circuit este de a **furniza un semnal de tensiune cu amplitudine variabilă** în funcție de **comanda numerică dată**. În componența convertorului digital – analog există un divizor de tensiune și o serie de comutatoare electronice care cuplează mai multe valori de tensiune la intrarea unui amplificator sumator (Fig. 9).

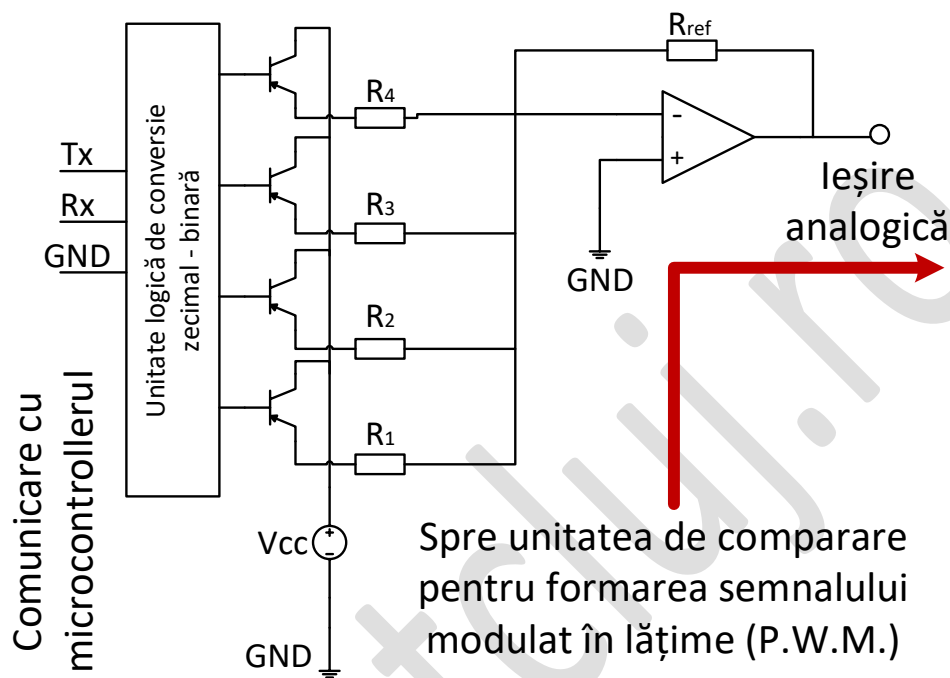


Fig. 9 – Schema bloc principală a convertorului digital – analog [2]

În cadrul unor arhitecturi de microcontroler sau procesor digital de semnal, nivelul de tensiune rezultat „DAC” [2] al convertorului digital – analog este utilizat în vederea **generării semnalului dreptunghiular modulat în lățime** „r(t)” (eng. Pulse Width Modulation – PWM). Convertorul digital – analog reprezintă în acest sens, o **sursă de tensiune cu amplitudine controlabilă în mod digital**. Tensiunea generată de către convertor, poate fi furnizată înspre un **etaj comparator** împreună cu un **semnal triunghiular** „Timer” cu **frecvență constantă** (Fig. 10 ~ 11). Semnalul triunghiular reprezintă „**unda purtătoare**”, iar tensiunea cu amplitudine variabilă reprezintă „**unda modulatoare**”. Semnalul rezultat, este de natură **digitală** având **frecvența** și **amplitudinea constantă** iar **lățimea variabilă** (Fig. 11).

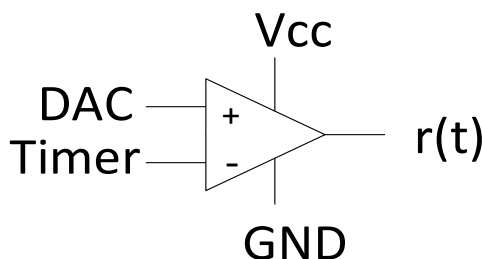


Fig. 10 – Etaj comparator intern pentru generarea semnalului modulat în lățime [2] [3]

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Semnalul triunghiular va fi generat de un circuit numărător sau temporizator (eng. counter ~ timer).

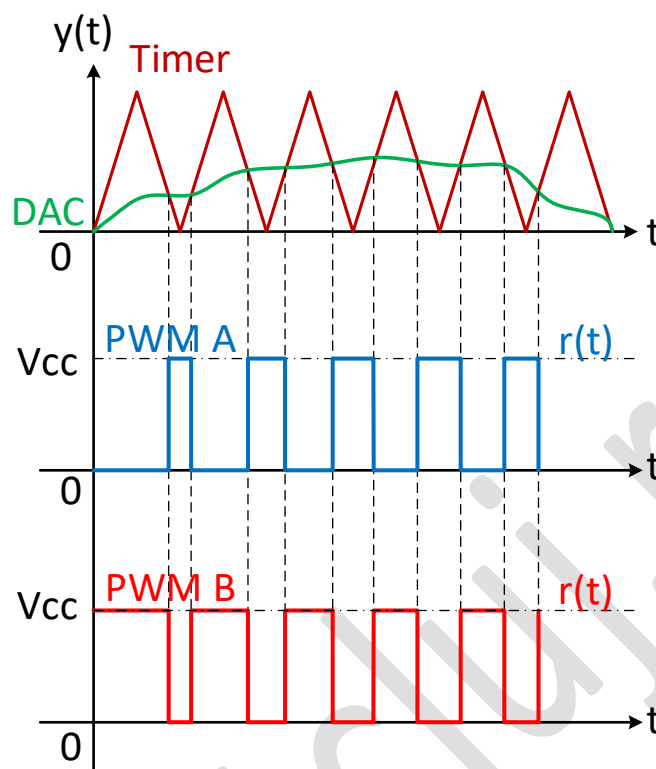


Fig. 11 – Generarea semnalului modulat în lăţime [1] [2] [3]

Fidelitatea pentru reprezentarea factorului de umplere al semnalului modulat în lăţime depinde de **rezoluţia numărătorului** şi **rezoluţia convertorului digital - analog** (dacă semnalul este generat pe baza unei strategii de comparare). Spre exemplu:

Fie indicaţia numerică a registrului „PWM = 125”. Pentru o unitate de generare a semnalului modulat în lăţime având rezoluţia „r” pe 8 biţi, să se determina factorul de umplere „d”:

$$r = 8 \text{ [bit]} = (2^8) - 1 = 255$$

$$x = \frac{1}{r} = \frac{1}{255} \approx 0,0039$$

$$d_{[\%]} = PWM \cdot \frac{1}{r} = 125 \cdot \frac{1}{255} \approx 0,49 \rightarrow 49 \text{ [\%]}$$

Unde „PWM” este indicaţia numerică a factorului de umplere (între [0 – 255]), iar „d[%]” factorul de umplere exprimat procentual cu fracţie sub-unitară. Motivul pentru care factorul de umplere are valori sub-unitare, constă în faptul că, amplitudinea undei purtătoare (triunghiulară) este maxim 1.

În vederea implementării aplicațiilor specifice procesării semnalelor analogice cu ajutorul microcontrolerului ATmega 328P, se vor avea în vedere două instrucțiuni:

- instrucțiunea condițională „if ()” (ro. dacă) și „else ()” (ro. altfel), spre exemplu:

```
if (variabila_1 > variabila_2) {  
    instrucțiune_1 (argument_1, ARGUMENT_2);  
}  
else {  
    instrucțiune_2 (argument_1, ARGUMENT_2);  
}
```

- instrucțiunea de constrângere și scalare „map ()” (ro. restrângerea unui domeniu de variație la un anumit interval de finit prin regula de trei simplă). Spre exemplu:

```
variabila_2 = map (variabila_1, valoare_inferioară_interval_1,  
    valoare_superioară_interval_1, valoare_inferioară_interval_2,  
    valoare_superioară_interval_2);
```

Fie variabila „p” cu valori în intervalul [0 1023]. Să se scrie funcția necesară pentru care, variabila „q” variază **direct - proporțional** cu variabila „p” în intervalul [0 255]:

```
q = map (p, 0, 1023, 0, 255);
```

Fie variabila „p” cu valori în intervalul [0 1023]. Să se scrie funcția necesară pentru care, variabila „q” variază **invers - proporțional** cu variabila „p” în intervalul [0 255]:

```
q = map (p, 0, 1023, 0, 255);
```

IV. IMPLEMENTAREA APLICAȚIILOR:

Se vor utiliza următoarele componente:

- placă pentru testare rapidă a circuitelor electronice (Wisher WBU-502L);
- platformă de dezvoltare Arduino NANO cu microcontroler ATmega 328;
- diode electro-luminiscente;
- rezistențe cu valoarea de 100 [Ω];
- senzor de temperatura LM-35;
- potențiomtru cu valoarea maximă a rezistenței 10 [kΩ];
- fire pentru conexiune rapidă compatibile cu placa de testare;
- calculator gazdă având mediul Arduino IDE instalat;
- cablu adaptor USB A la mini USB;

În vederea studierii aspectelor și conceptelor teoretice se propune implementarea unei serii a câte șase aplicații pe baza platformei Arduino Nano, precum:

- determinarea rezoluției convertorului analog – digital (cu afișare în monitorul serial);
- determinarea preciziei convertorului analog – digital și a valorii de tensiune măsurată;
- determinarea temperaturii cu ajutorul traductorului LM-35;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

- implementarea unui comparator numeric cu prag digital reglabil (funcția „if”);
- implementarea unei coloane luminoase indicatoare de nivel (funcția „map”);
- generarea unui semnal dreptunghiular modulat în lățime (funcția „map”);

APLICAȚIA 1:

Se va implementa circuitul conform următoarei scheme (Fig. 12):

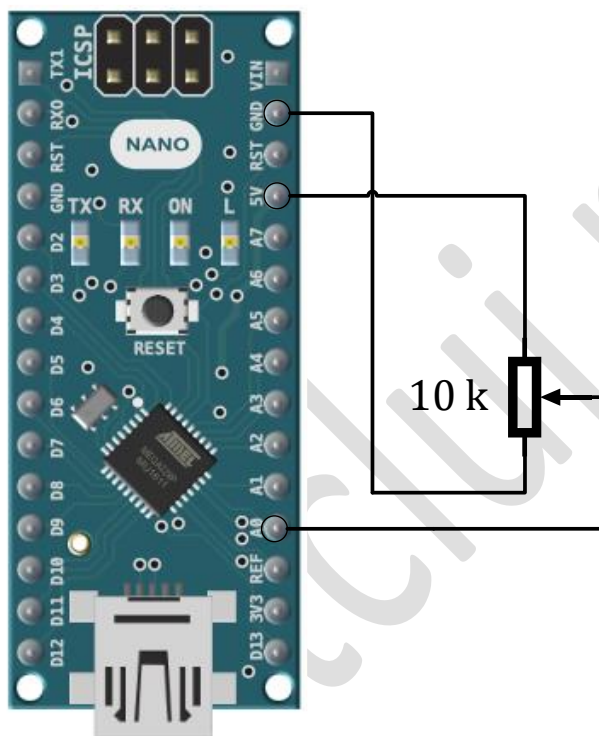


Fig. 12 – Schema electronică pentru implementarea aplicației 1 și 2 [3] [4]

Se va realiza următorul montaj experimental (Fig. 13):

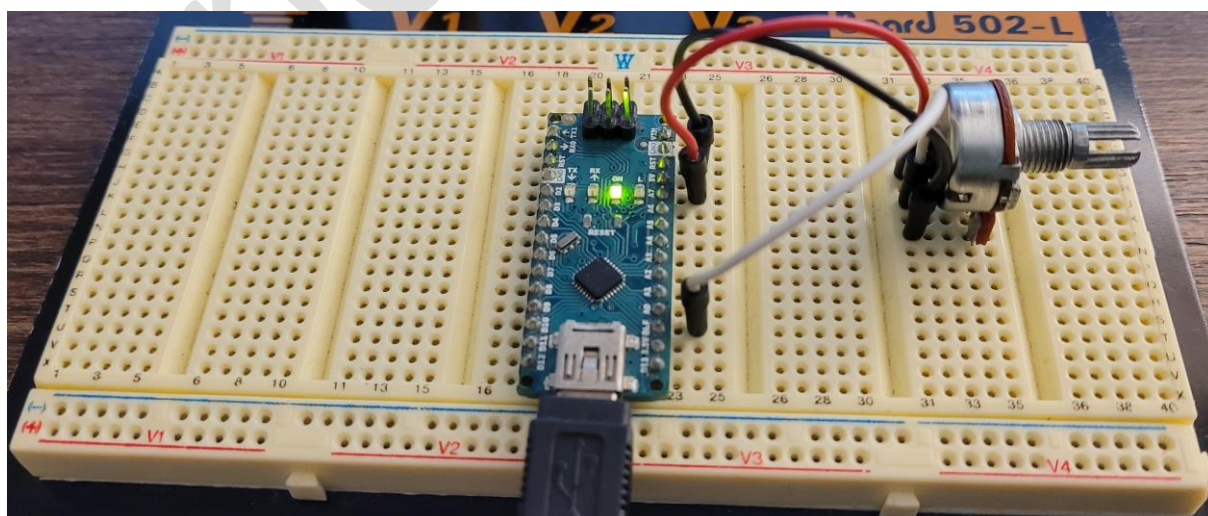


Fig. 13 – Montajul experimental specific aplicației 1 și 2

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Se va implementa următorul cod program:

```
const int analog_pin = 0;
int ADC_val = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  ADC_val = analogRead(analog_pin);
  Serial.print("ADC: ");
  Serial.print(ADC_val);
  Serial.println("");
  delay(250);
}
```

În urma implementării codului program la nivelul microcontrolerului ATmega 328P, se va deschide consola Serial (meniul „Tools” --> „Serial Monitor”) (Fig. 14).

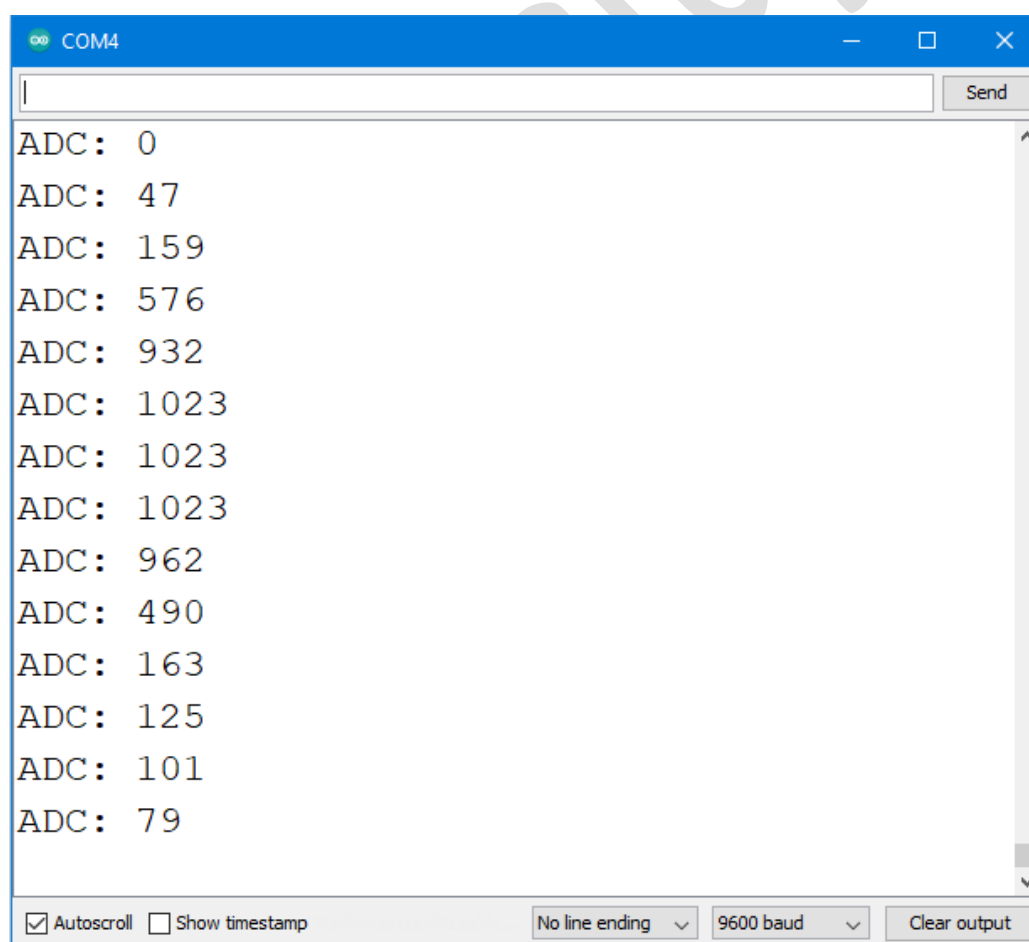


Fig. 14 – Consola Serial – Rezultatul conversiei analog - digital

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Implementarea aplicației nr. 1 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s];
- preluarea valorii zecimale rezultante în urma procesului de conversie analog – digital;
- afișarea rezultatului de conversie în consola serial însoțit de mesajul „ADC: ”;

Se constată faptul că intervalul de variație al rezultatului conversiei analog – digitale este cuprins între [0 1023] (Fig. 14). Valorile obținute corespund unei rezoluții de 10 biți, astfel valoarea maximă în format zecimal poate fi determinată pe baza relației:

$$r = 10 \text{ [bit]} = (2^{10}) - 1 = 1023$$

APLICAȚIA 2:

Utilizând același montaj ca și în cazul aplicației anterioare (Fig. 1 și 2), se va determina (pe baza următoarelor relații) atât precizia convertorului analog – digital cât și valoarea nivelului de tensiune înregistrat în mod dinamic de către convertor.

$$x = \frac{U_{ref}}{r} \rightarrow U_{m\acute{a}s} = ADC \cdot \frac{U_{ref}}{r}$$

Se va implementa următorul cod program:

```
const int analog_pin = 0;
int ADC_val = 0;
float U = 0.00;

void setup() {
  Serial.begin(9600);
}

void loop() {
  ADC_val = analogRead(analog_pin);
  U = (5.00 / 1023.00) * ADC_val;
  Serial.print("Tensiune: ");
  Serial.print(U);
  Serial.print(" [V]");
  Serial.println("");
  delay(250);
}
```

În urma implementării codului program se va deschide consola Serial (Fig. 15).

Precizia convertorului analog – digital este aproximativ 4 [mV / pas], iar tensiunea măsurată variază în intervalul [0 5] [V], în situația în care nivelul tensiunii de referință sau nivelul tensiunii de alimentare este egal cu 5 [V], (Fig. 15).

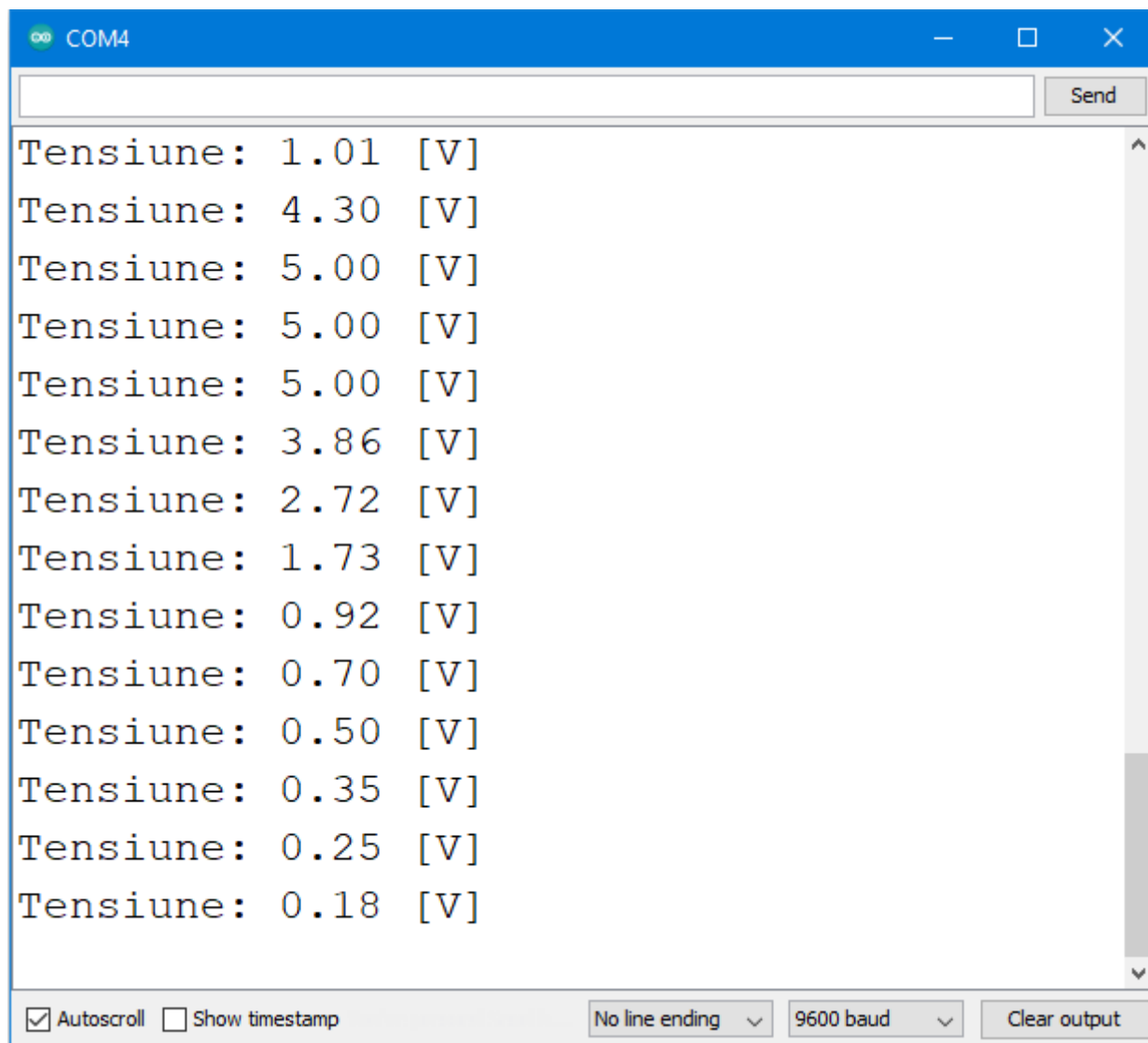


Fig. 15 – Consola Serial – Afișarea nivelului de tensiune măsurat de convertor

Implementarea aplicației nr. 2 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea unei variabile de tip fracționar „U” cu valoarea „0.00”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s];
- preluarea valorii zecimale rezultante în urma procesului de conversie analog – digital;
- determinarea tensiunii de măsură pe baza preciziei convertorului analog – digital;
- afișarea nivelului de tensiune în consola Serial însoțit de mesajul „Tensiune: ____ [V]”;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAȚIA 3:

Se va implementa circuitul conform următoarei scheme (Fig. 16):

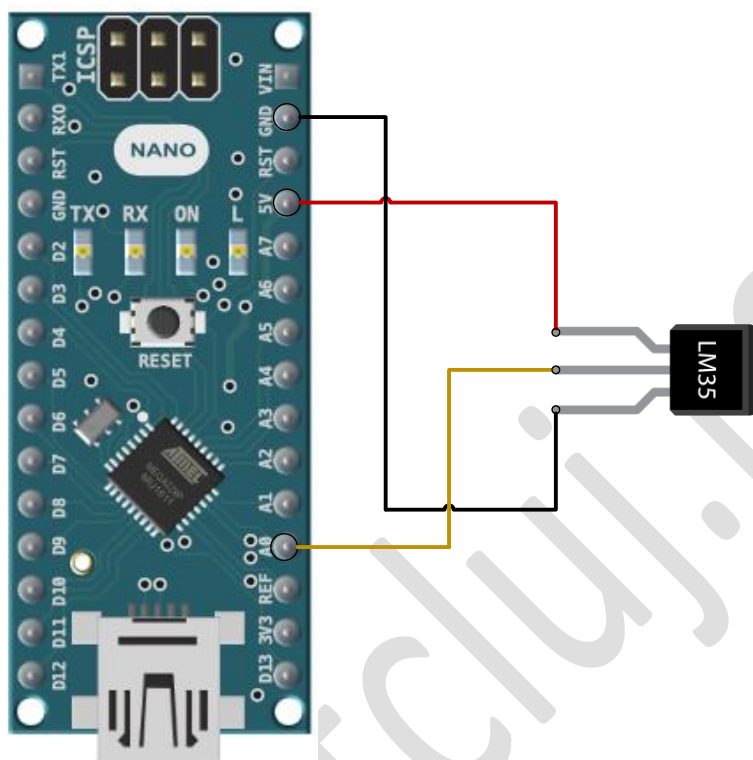


Fig. 16 – Schema electronică pentru implementarea circuitului specific aplicației 3 [3] [4]

Se va realiza următorul montaj experimental (Fig. 17):

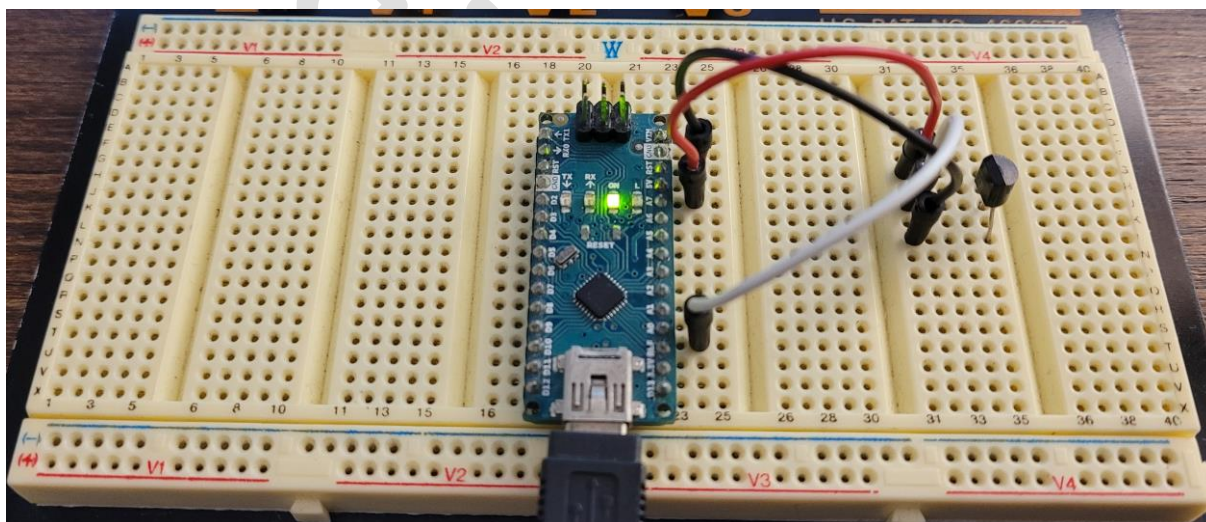


Fig. 17 – Montajul experimental specific aplicației 3

Pe baza datelor de catalog a fost identificată constanta de calibrare $10 \text{ [mV / } ^\circ\text{C]}$.

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Se va implementa următorul cod program:

```
const int analog_pin = 0;
int ADC_val = 0;
float U = 0.00;
float temp = 0.00;

void setup() {
  Serial.begin(9600);
}

void loop() {
  ADC_val = analogRead(analog_pin);
  U = (5.00 / 1023.00) * ADC_val;
  temp = 100.00 * U;
  Serial.print("Temperatura: ");
  Serial.print(temp);
  Serial.print(" [*C]");
  Serial.println("");
  delay(250);
}
```

Pe baza instrumentului grafic de analiză (Serial Plotter), se va observa evoluția în timp a temperaturii înregistrată de traductorul LM-35 (Fig. 18):

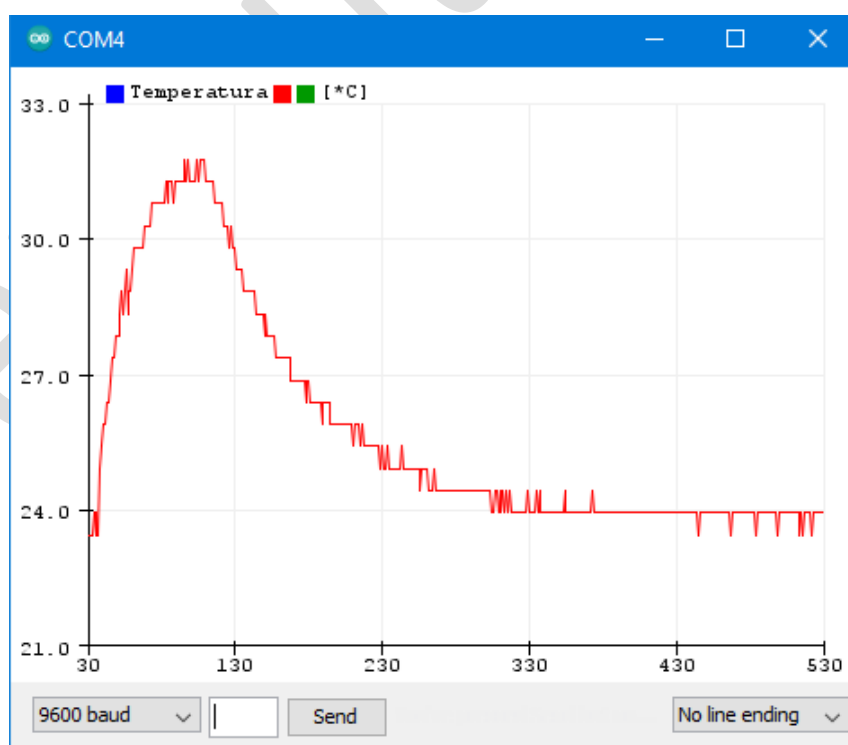


Fig. 18 – Variația temperaturii înregistrată de traductorul LM-35 în raport cu timpul

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Se va observa de asemenea modul de transmitere în consola Serial al mesajului care indică temperatura ambientală (Fig. 19).

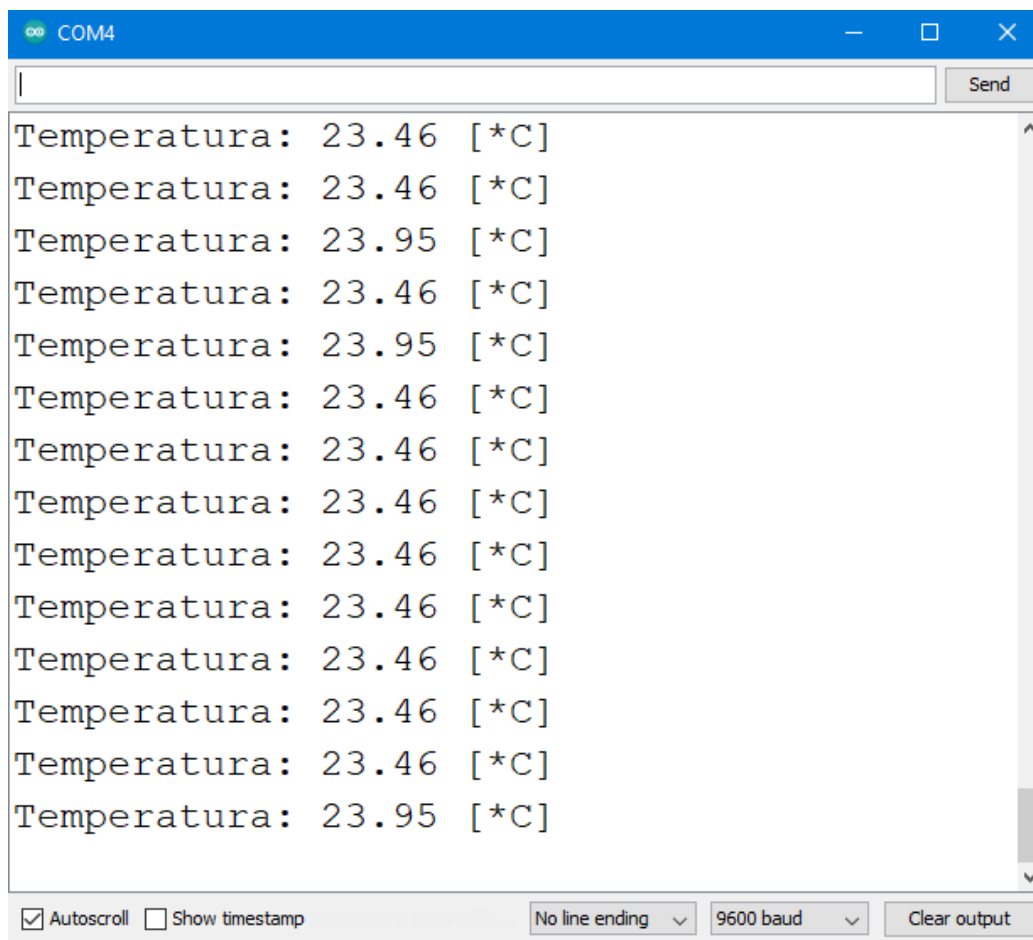


Fig. 19 – Consola Serial – Variația temperaturii în raport cu timpul

Se constată faptul că, față de valoarea reală a temperaturii, există o serie de abateri, care pot fi percepute ca și erori de măsurare sau „zgomote” induse în semnalul de măsură. Din acest motiv, va fi necesar ca semnalul de măsură achiziționat să fie filtrat cu ajutorul algoritmului de mediere dinamică (determinarea în mod dinamic a mediei aritmetice pe baza punctelor de măsură înregistrate).

Implementarea aplicației nr. 3 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea unei variabile de tip fracționar „U” cu valoarea „0.00”;
- inițializarea unei variabile de tip fracționar „temp” cu valoarea „0.00”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s];
- preluarea valorii zecimale rezultante în urma procesului de conversie analog – digital;
- determinarea tensiunii de măsură pe baza preciziei convertorului analog – digital;
- determinarea temperaturii pe baza tensiunii de măsură și a constantei de calibrare;
- afișarea valorii temperaturii în consola Serial însoțită de mesajul „Temperatura: _ [*C] ”;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAȚIA 4:

Se va implementa circuitul conform următoarei scheme (Fig. 20):

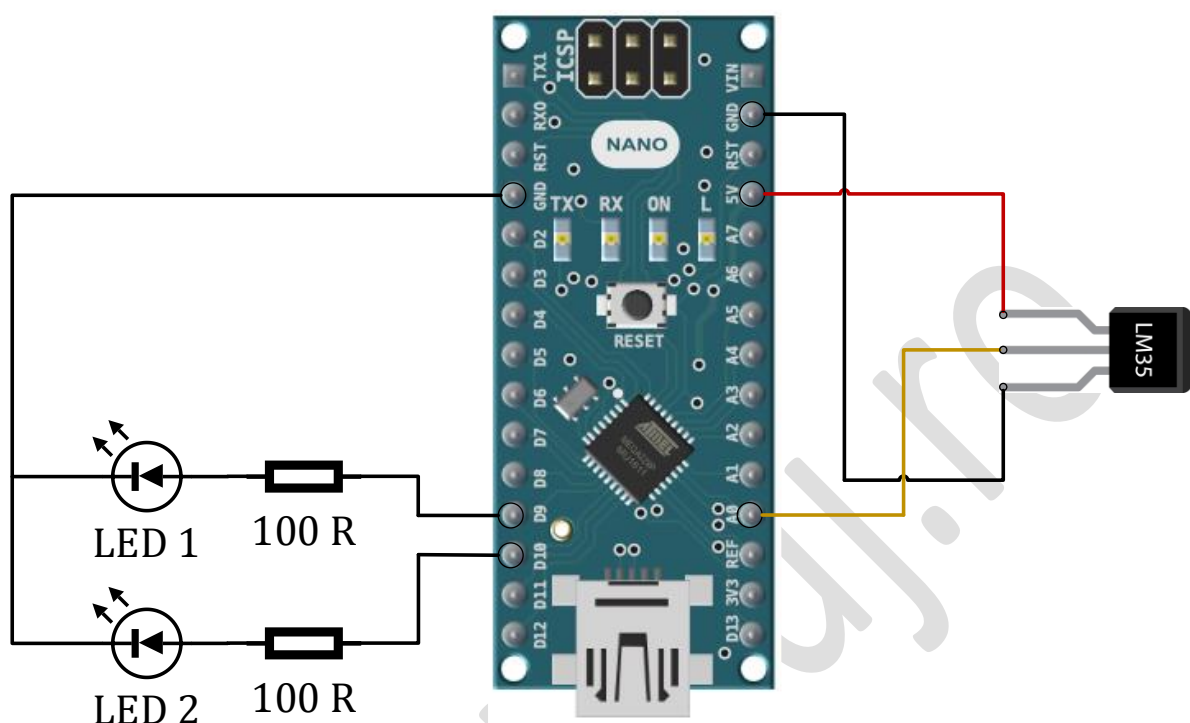


Fig. 20 – Schema electronică pentru implementarea circuitului specific aplicației 4 [3] [4]

Se va realiza următorul montaj experimental (Fig. 21):

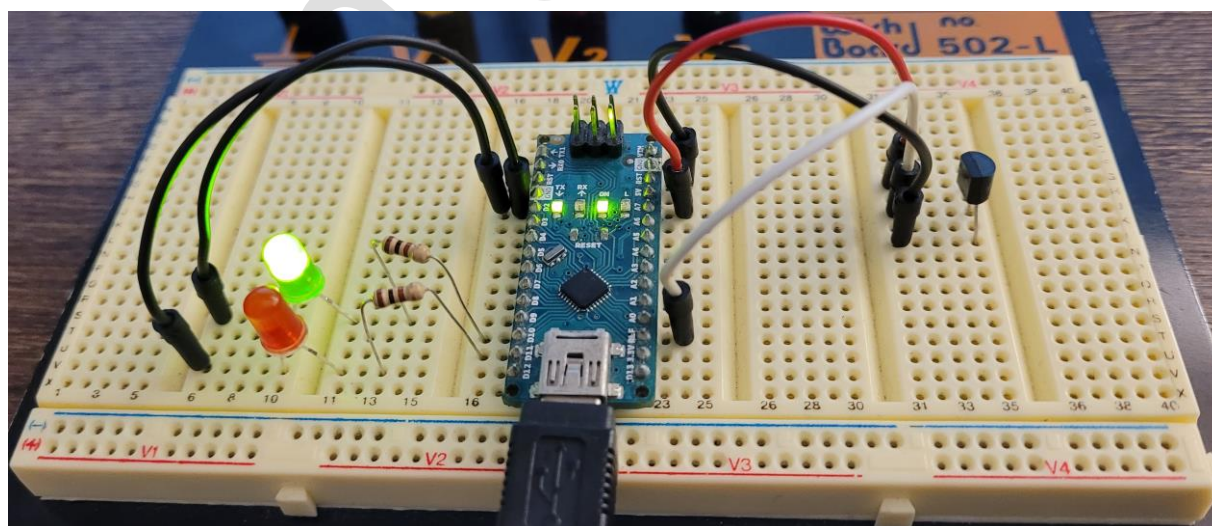


Fig. 21 – Montajul experimental specific aplicației 4

Se va implementa următorul cod program:

```
const int analog_pin = 0;
const int led_1 = 9;
const int led_2 = 10;
int ADC_val = 0;
float U = 0.00;
float temp = 0.00;
int prag = 28;

void setup() {
    pinMode(led_1, OUTPUT);
    pinMode(led_2, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    ADC_val = analogRead(analog_pin);
    U = (5.00 / 1023.00) * ADC_val;
    temp = 100.00 * U;
    if (temp < prag) {
        digitalWrite(led_1, HIGH);
        digitalWrite(led_2, LOW);
    }
    else {
        digitalWrite(led_1, LOW);
        digitalWrite(led_2, HIGH);
    }
    Serial.print("Temperatura actuala: ");
    Serial.print(temp);
    Serial.print(" [*C]");
    Serial.print(" Prag: ");
    Serial.print(prag);
    Serial.print(" [*C]");
    Serial.println("");
    delay(250);
}
```

Implementarea aplicației nr. 4 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- declararea unei constante de tip număr întreg „led_1” având ca și valoare „9”;
- declararea unei constante de tip număr întreg „led_2” având ca și valoare „100”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea unei variabile de tip fracționar „U” cu valoarea „0.00”;
- inițializarea unei variabile de tip fracționar „temp” cu valoarea „0.00”;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

- inițializarea unei variabile de tip număr întreg „prag” cu valoarea „28”;
- stabilirea modului de lucru „ieșire” pentru terminalele digitale „D9” și „D10”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s];
- preluarea valorii zecimale rezultante în urma procesului de conversie analog – digital;
- determinarea tensiunii de măsură pe baza preciziei convertorului analog – digital;
- determinarea temperaturii pe baza tensiunii de măsură și a constantei de calibrare;
- compararea nivelului temperaturii actuale cu valoarea de prag impusă inițial;
- afișarea în consolă atât a valorii actuale cât și a pragului de temperatură;

Se va deschide consola Serial pentru vizualizarea valorilor (Fig. 22).

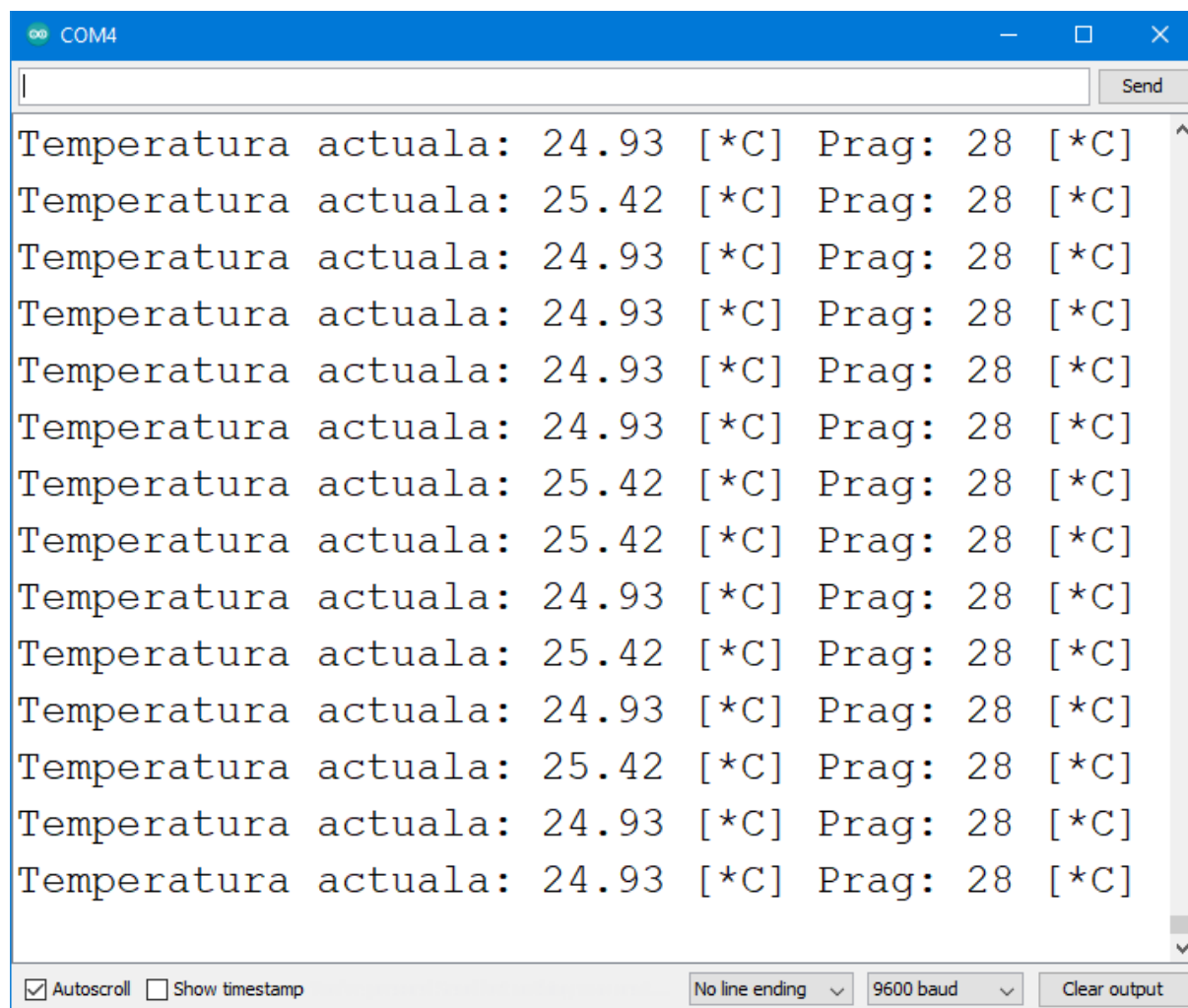


Fig. 22 – Consola Serial – Variația temperaturii în raport cu valoarea de prag

Pentru vizualizare se va utiliza de asemenea instrumentul pentru analiză grafică a variației temperaturii ambientale în raport cu o valoare de prag impusă (Fig. 23).

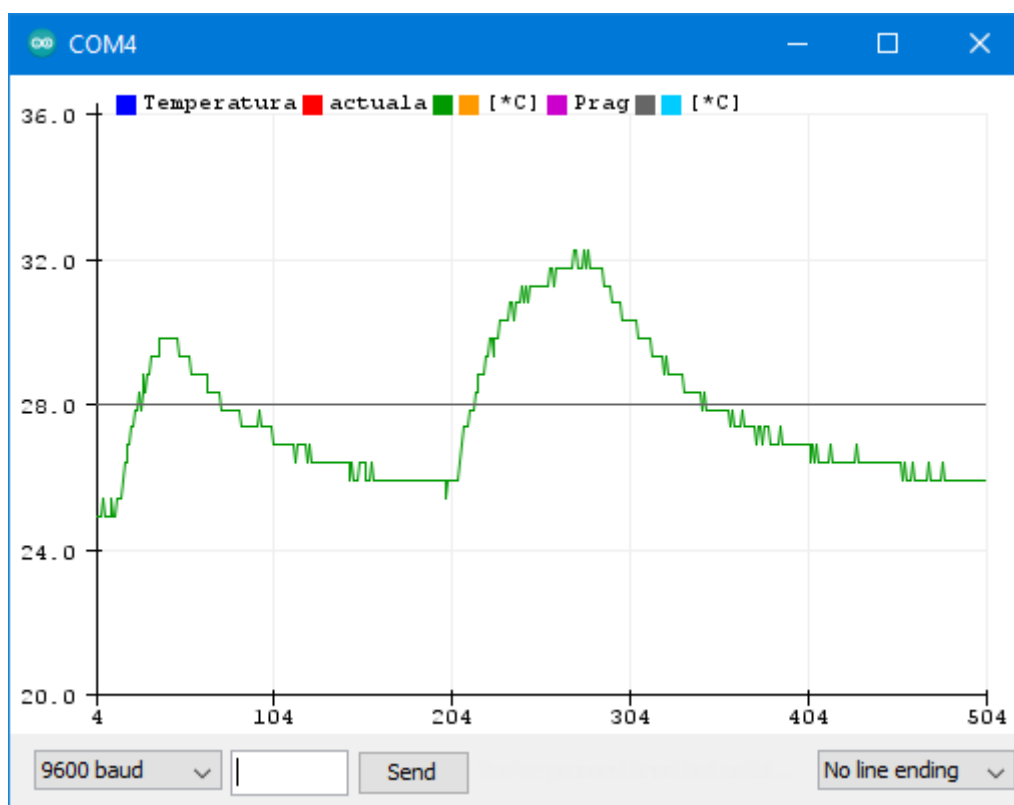


Fig. 23 – Variația temperaturii în raport cu valoarea de prag impusă inițial

APLICAȚIA 5:

Se va implementa circuitul conform următoarei scheme (Fig. 24):

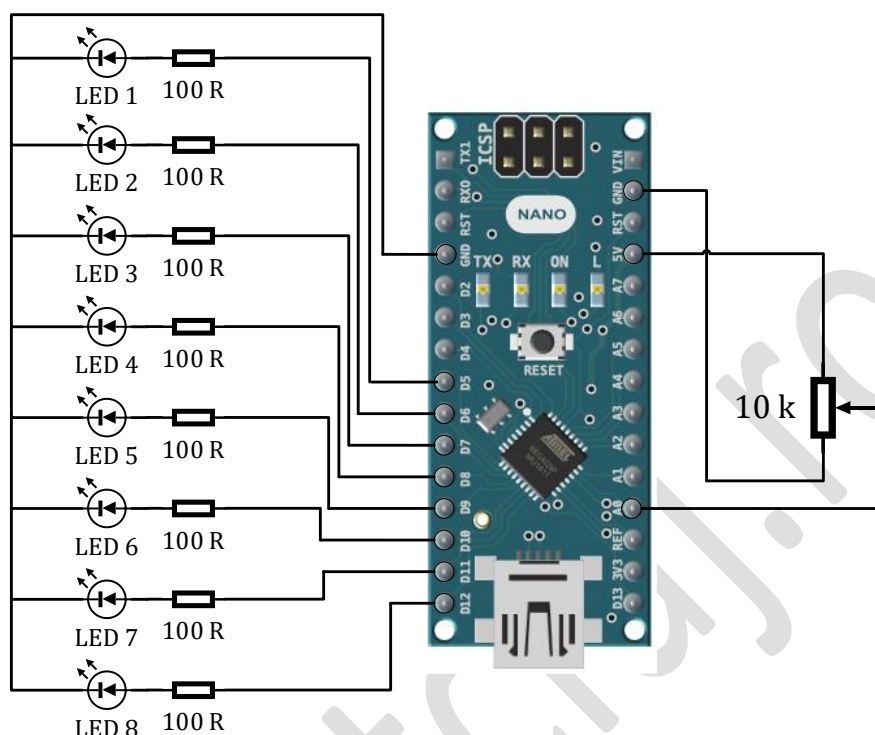


Fig. 24 – Schema electronică pentru implementarea circuitului specific aplicației 5 [3] [4]

Se va realiza următorul montaj experimental (Fig. 25):

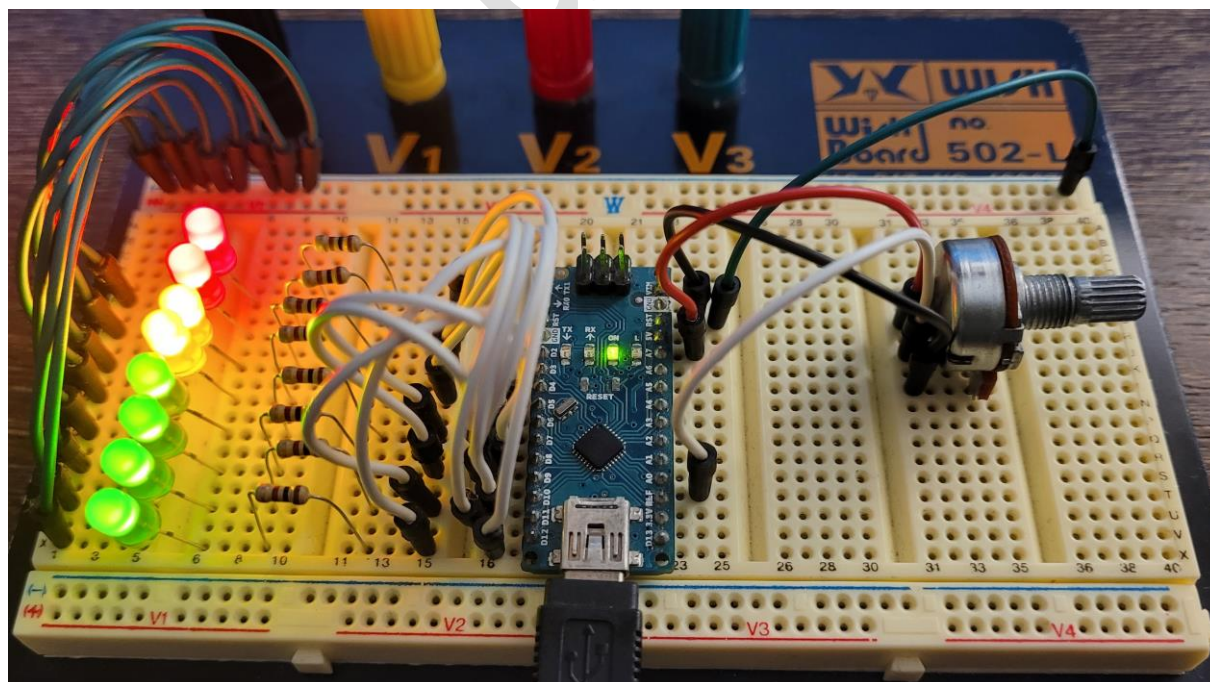


Fig. 25 – Montajul experimental specific aplicației 5

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Se va implementa următorul cod program:

```
const int analog_pin = 0;
int pin_led[] = {5, 6, 7, 8, 9, 10, 11, 12};

int ADC_val = 0;
int nivel = 0;

void setup() {
  for (int i = 0; i <= 7; i++) {
    pinMode(pin_led[i], OUTPUT);
  }
}

void loop() {
  ADC_val = analogRead(analog_pin);
  nivel = map(ADC_val, 0, 1023, 0, 8);
  for (int i = 0; i <= 7; i++) {
    if (i < nivel) {
      digitalWrite(pin_led[i], HIGH);
    }
    else {
      digitalWrite(pin_led[i], LOW);
    }
  }
}
```

Implementarea aplicației nr. 5 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- declararea a unui șir finit de opt numere întregi „pin_led []”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea unei variabile de tip număr întreg „nivel” cu valoarea „0”;
- inițializarea terminalelor digitale din șir în modul de lucru „ieșire digitală”;
- preluarea valorilor de la convertorul analog – digital;
- scalarea domeniului de variație al convertorului la numărul de terminale din șir;
- parcurgerea șirului în sens crescător și activarea stării digitale la fiecare terminal;
- parcurgerea șirului în sens descrescător și dezactivarea stării digitale la fiecare terminal;

OBSERVAȚIE: Montajul și codul program realizat în cadrul aplicației nr. 5, poate fi adaptat la orice tip de senzor analogic, care furnizează o tensiune cu nivel variabil. De asemenea, numărul de elemente semnalizatoare, poate fi modificat (pentru coloane indicatoare cu un număr mai mare de elemente).

APLICAȚIA 6:

Se va implementa circuitul conform următoarei scheme (Fig. 26):

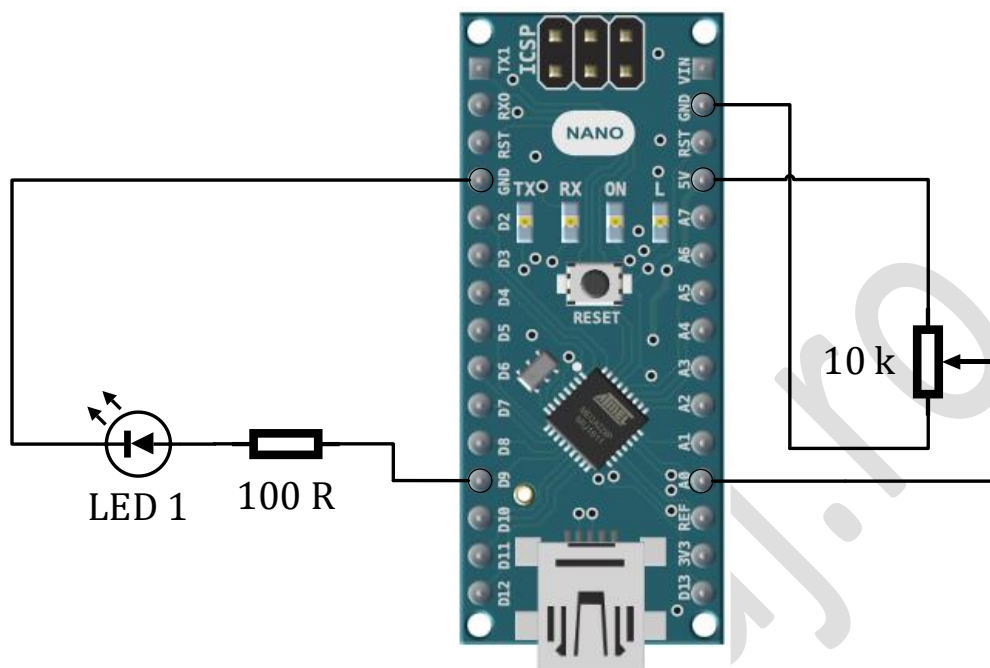


Fig. 26 – Schema electronică pentru implementarea circuitului specific aplicației 6 [3] [4]

Se va realiza următorul montaj experimental (Fig. 27):

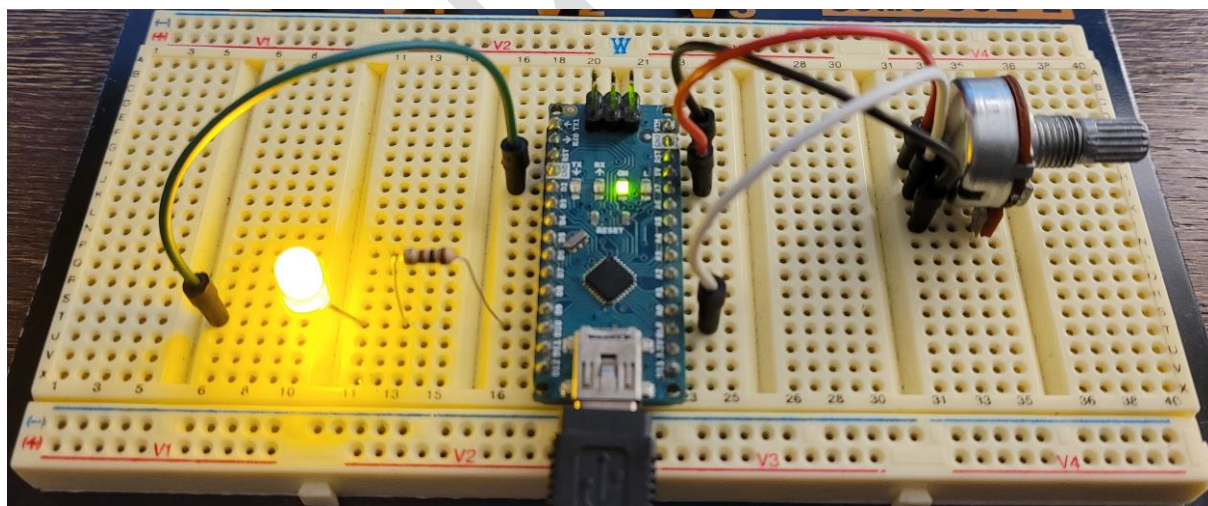


Fig. 27 – Montajul experimental specific aplicației 6

Se va implementa următorul cod program:

```
const int analog_pin = 0;
const int led_1 = 9;

int ADC_val = 0;
int dc_1 = 0;
float dc_1_p = 0.00;

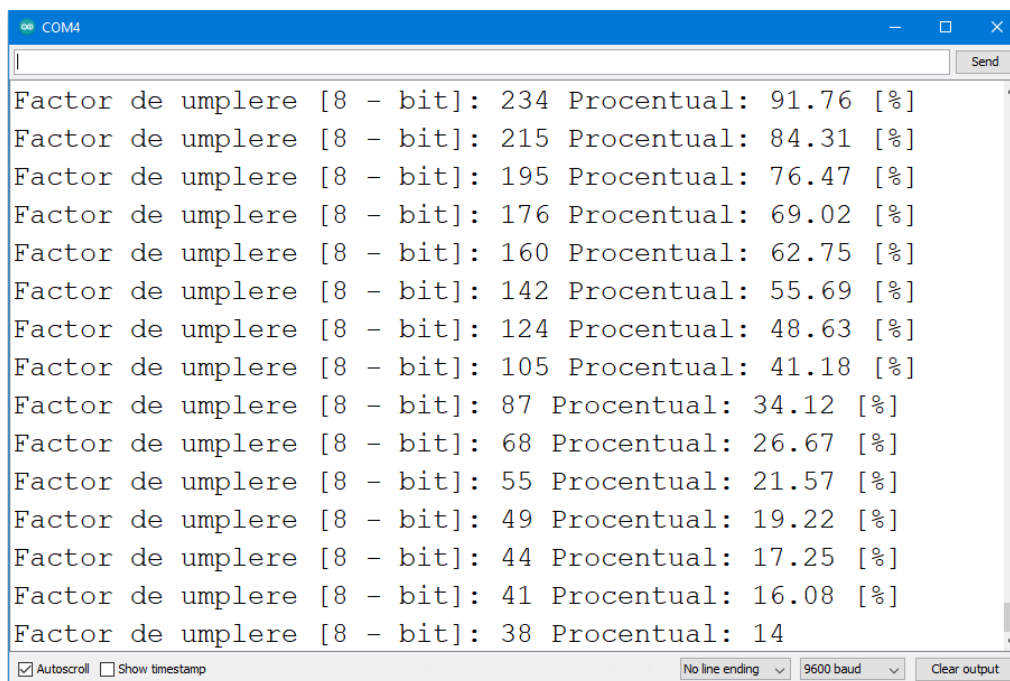
void setup() {
    pinMode(led_1, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    ADC_val = analogRead(analog_pin);
    dc_1 = map(ADC_val, 0, 1023, 0, 255);
    dc_1_p = ((1.00 / 255.00) * dc_1) * 100;
    Serial.print("Factor de umplere [8 - bit]: ");
    Serial.print(dc_1);
    Serial.print(" Procentual: ");
    Serial.print(dc_1_p);
    Serial.print(" [%]");
    Serial.println("");
    analogWrite(led_1, dc_1);
}
```

Implementarea aplicației nr. 6 presupune:

- declararea unei constante de tip număr întreg „analog_pin” având ca și valoare „0”;
- declararea unei constante de tip număr întreg „led_1”;
- inițializarea unei variabile de tip număr întreg „ADC_val” cu valoarea „0”;
- inițializarea unei variabile de tip număr întreg „dc_1” cu valoarea „0”;
- inițializarea unei variabile de tip fracționar „dc_1_p” cu valoarea „0.00”;
- inițializarea terminalului digital „9” în modul de lucru „ieșire digitală”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s]
- preluarea valorilor de la convertorul analog – digital;
- scalarea domeniului convertorului la valoarea maximă reprezentată pe 8 biți;
- afișarea atât a valorii exprimată pe 8 biți a duratei de conducție cât și procentuală;
- ajustarea factorului de umplere pentru un tren de impulsuri furnizat pe terminalul „9”;

Se va deschide consola Serial pentru a vizualiza datele furnizate de către microcontroler înspre calculator (Fig. 28).



```

COM4
Factor de umplere [8 - bit]: 234 Procentual: 91.76 [%]
Factor de umplere [8 - bit]: 215 Procentual: 84.31 [%]
Factor de umplere [8 - bit]: 195 Procentual: 76.47 [%]
Factor de umplere [8 - bit]: 176 Procentual: 69.02 [%]
Factor de umplere [8 - bit]: 160 Procentual: 62.75 [%]
Factor de umplere [8 - bit]: 142 Procentual: 55.69 [%]
Factor de umplere [8 - bit]: 124 Procentual: 48.63 [%]
Factor de umplere [8 - bit]: 105 Procentual: 41.18 [%]
Factor de umplere [8 - bit]: 87 Procentual: 34.12 [%]
Factor de umplere [8 - bit]: 68 Procentual: 26.67 [%]
Factor de umplere [8 - bit]: 55 Procentual: 21.57 [%]
Factor de umplere [8 - bit]: 49 Procentual: 19.22 [%]
Factor de umplere [8 - bit]: 44 Procentual: 17.25 [%]
Factor de umplere [8 - bit]: 41 Procentual: 16.08 [%]
Factor de umplere [8 - bit]: 38 Procentual: 14
Autoscroll Show timestamp No line ending 9600 baud Clear output
  
```

Fig. 28 – Afișarea factorului de umplere sau a duratei de conducție în consola Serial

V. CONCLUZIE:

Procesarea semnalelor atât de natură analogică cât și de natură digitale, poate fi realizată prin intermediul oricărei strategii de implementare a codului program la nivelul microcontrolerului ATmega 328P.

VI. BIBLIOGRAFIE:

1. Atmel Corporation © 2015, Rev.: 7810D – AVR – 01 / 15 – „ATmega328P datasheet - 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash”;
2. Atmel Corporation © 2015, Rev.: Atmel-11057C-ATARM-SAM3X-SAM3A – „SAM3X / SAM3A Series – Atmel SMART ARM-based MCU datasheet”
3. Ioana - Cornelia Gros, Lucian - Nicolae Pintilie, Teodor Crișan Pană – „SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ - GHID DE APLICAȚII”, Editura UTPRESS, Cluj-Napoca, 2020, ISBN 978-606-737-431-5
4. Arduino Store © 2021 Arduino SRL - Partita IVA 09755110963 – „Arduino Nano”
<https://store.arduino.cc/products/arduino-nano>
5. Electronics and Power electronics (EPE) Brings power and electronics together © 2017 – „Documentație pentru laboratorul de Sisteme cu Microprocesoare”
<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro

Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro