Sisteme cu microprocesoare
– Aplicații generale de interfațare –

I. SCOPUL LUCRĂRII:

Lucrarea de laborator are ca scop:
- prezentarea conceptului „aplicație de interfațare” [1] [2]
- prezentarea metodelor și procedeelor pentru optimizarea unei aplicații [1] [2]
- prezentarea modului de funcționare al unui senzor ultrasonic [3]
- prezentarea modului de funcționare al unui afișaj LCD pe interfață I²C [4] [5]
- implementarea unor aplicații de interfațare [1] [2] [3] [4] [5]

II. INTRODUCERE:

Conceptul de **aplicație** reprezintă orice formă de **implementare a unui circuit electronic** atât pe baza **perifericelor specializate** ale microcontrolerelor cât și pe baza altor sisteme de calcul specializate în acest sens (ex. comunicare cu calculatorul gazdă sau cu telefonul mobil) [1] [2].

Conceptul de **aplicație de interfațare**, presupune implementarea unui etaj de circuit pe baza perifericelor specializate ale unui microcontroler (ex. intrări – ieșiri digitale, intrări analogice, interfețe de comunicare). Perifericele de intrare și ieșire ale microcontrolerului pot fi atașate la elemente de circuit precum: afișaje LCD, butoane, senzori sau elemente de execuție precum motoare, rezistențe sau semnalizatoare acustice. Rolul dispozitivului rezultat în urma implementării, este de a **intermedia operația de reglare** a procesul tehnologic final realizată de factorul uman. Funcția îndeplinită în acest sens se numește „interfațare” (Fig. 1).

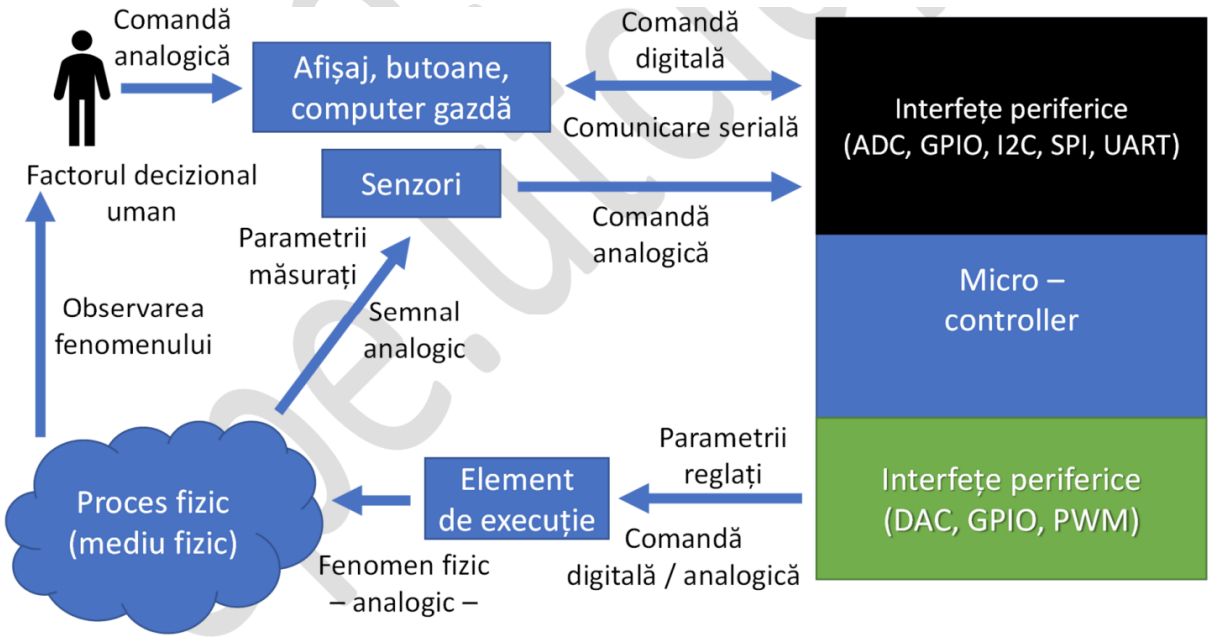

Fig. 1 – Funcția de interfațare [1] [2]

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

Există trei tipuri de aplicaţii [1] [2]:
- aplicaţii independente (cu algoritm de condiţionare proprie);
- aplicaţii dependente de elementele de interfaţare (LCD şi butoane sau calculator);
- aplicaţii hibride (comandate de utilizator dar cu algoritm de condiţionare propriu);

A. Aplicaţii independente:

Acest mod de implementare, se bazează pe programul propriu inscripţionat de către producătorul echipamentului în memoria microcontrolerului, iar utilizatorul nu poate modifica modul de funcţionare şi nici nu poate furniza o comandă de modificare a stării în care se află dispozitivul. Interacţiunea cu sistemul de calcul se va realiza prin intermediul senzorilor (sau mai precis al perifericelor de intrare) iar semnalul rezultant va fi furnizat către un element de execuţie (prin intermediul perifericelor de ieşire).

În această categorie se încadrează aplicaţii precum: detectoare, sesizoare, elemente de semnalizare, elemente de execuţie cu algoritm condiţional propriu (Fig. 2).

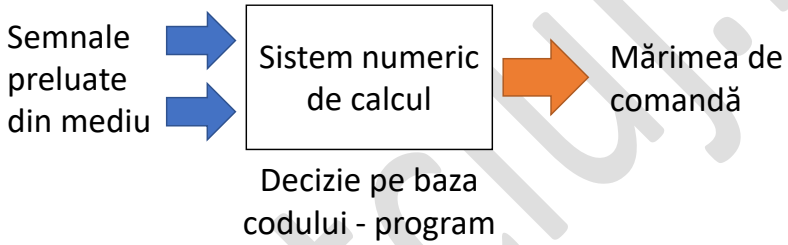

Fig. 2 – Aplicaţii independente [1] [2]

B. Aplicaţii dependente de elementele de interfaţare:

Acest mod de implementare necesită o comandă furnizată de către operator pentru a schimba modul de funcţionare al aplicaţiei. Interacţiunea dintre operator şi dispozitiv sau procesul tehnologic final, se va realiza prin intermediul elementelor de interfaţare ataşate la perifericele microcontrolerului (ex. LED ataşat la ieşirea digitală pentru semnalizare, buton ataşat la intrarea digitală pentru preluarea comenzilor, afişaj LCD pentru transmiterea mesajelor text de informare asupra stării de funcţionare).

În această categorie se încadrează aplicaţii precum: dispozitivele periferice ale computerului, panouri frontale cu afişaj şi butoane, echipamente de interacţiune umană cu procesul fizic (Human Interface Device – HID) (Fig. 3).

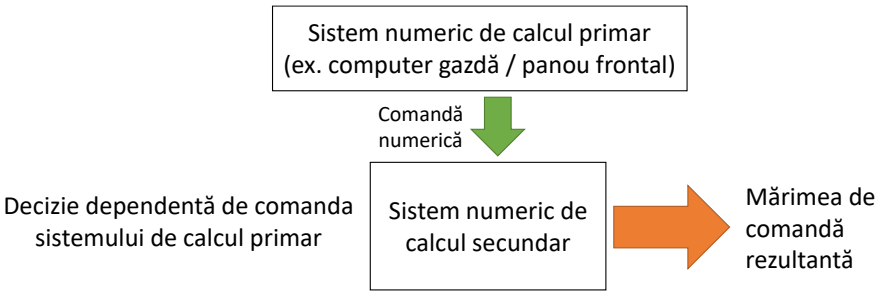

Fig. 3 – Aplicaţii dependente de elementele de interfaţare [1] [2]

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

C. Aplicaţii hibride:

Acest mod de implementare are la baza funcţionării atât programul pre-implementat de către producător în memorie microcontrolerului cât şi un program pentru interacţiune cu procesul din exterior (adică prin intermediul unui alt sistem de calcul sau prin intermediul propriilor elemente de interfaţare cu afişaj LCD şi butoane). Modul de funcţionare al acestui tip de aplicaţii poate fi influenţat atât prin intermediul elementelor proprii de interfaţare ale dispozitivului cât şi prin intermediul unui alt sistem de calcul precum calculatorul personal sau telefonul mobil (Fig. 4).

Câteva exemple din această categorie pot fi dispozitivele complexe care pot fi configurate şi asistate de la distanţă pentru a deservi un proces fizic: termostate digitale programabile atât de la distanţă cât şi prin intermediul panoului frontal propriu, ventilatoarele industriale comandate de la distanţă dar care au şi posibilitatea adjustării în funcţie de datele preluate de la senzori, imprimante multifuncţionale etc.

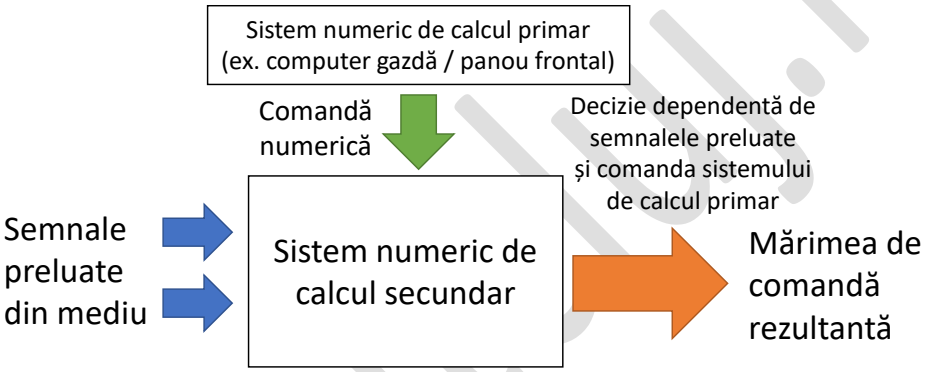

Fig. 4 - Aplicaţii hibride [1] [2]

III. ASPECTE TEORETICE:

Există diverse metode şi procedee utilizate în vederea îmbunătăţirii modului de funcţionare al aplicaţiei concepute, precum [1] [2]:
- introducerea unui timp de întârziere (prin intermediul funcţiei „delay()”);
- introducerea constrângerilor simple (pentru limitarea unui interval de variaţie);
- introducerea unui algoritm de mediere în timp a semnalului;

A. Introducerea unui timp de întârziere:

Prin intermediul funcţiei „delay()”, pot fi impuse **intervale de timp constante** la care să aibă loc o anumită operaţie impusă prin codul – program. Spre exemplu, în situaţia preluării temperaturii de la un senzor în cele mai multe cazuri **nu este necesară achiziţia de date la frecvenţă mare**, deoarece, variaţia temperaturii în timp reprezintă un proces **relativ lent**. Având în vedere modul de funcţionare al convertorului analog – digital, se constată că procesul de achiziţie al semnalului furnizat de traductor, este oarecum susceptibil la **erorile cauzate de frecvenţa la care se realizează achiziţia de date**.

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Acest impediment, deci, poate fi corectat introducând un **timp de întârzie,** (Fig. 5) astfel va avea loc şi stabilizarea termică a traductorului!

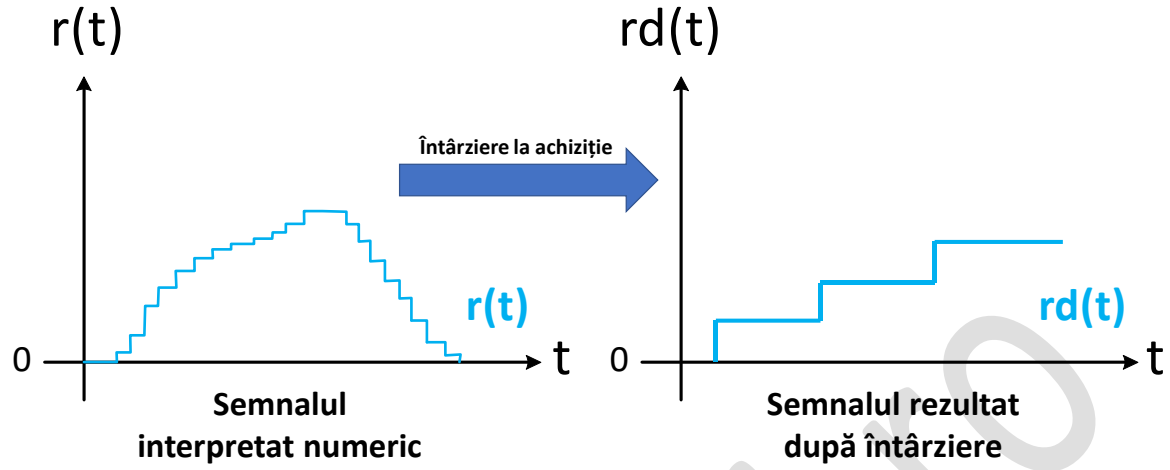

Fig. 5 - Aplicaţii hibride [1] [2]

B. Introducerea unor constrângeri:

În unele cazuri este necesară **constrângerea** sau **restrângerea** intervalului de variaţie al unei mărimi. În vederea realizării acestui lucru, există două procedee:
- constrângerea unui semnal cu variaţie infinită la un interval de variaţie finit (mărginire);
- restrângerea sau lărgirea unui interval de variaţie finit (scalare);

Spre exemplu:
- constrângerea sau mărginirea se poate aplica în cazul în care se doreşte limitarea variabilei care se incrementează sau decrementează în vederea reglării factorului de umplere pentru PWM cu două butoane (Fig. 6);
- restrângerea sau scalarea se poate aplica în cazul în care se doreşte redefinirea intervalului de variaţie în vederea reglării factorului de umplere pentru PWM folosind rezultatul conversiei analog – digitale (Fig. 7);

Pentru o variabilă oarecare în intervalul [**constantă_1 constantă_2**] se va impune o constrângere simplă prin următorul procedeu:

```
if (variabilă <= constantă_1) {
      variabilă = constantă_1;
      }
if (variabilă >= constantă_2) {
      variabilă = constantă_2;
      }
```

Există şi metoda alternativă:

```
constrain (variabilă, constantă_1, constantă_2);
```


Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

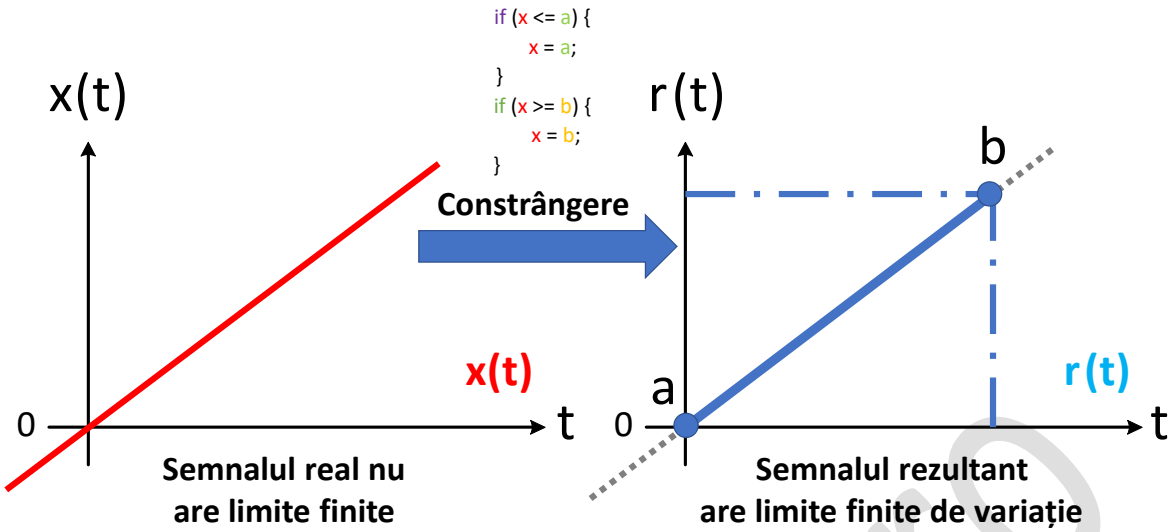

Fig. 6 – Constrângere sau mărginire simplă a unui interval infinit [1] [2]

Restrângerea intervalului de variație finit se va realiza prin intermediul funcției „map()" sub forma următoarei sintaxe:

```
variabila_2 = map (variabila_1, valoare_inferioară_1,
valoare_superioară_1, valoare_inferioară_2, valoare_superioară_2);
```

Spre exemplu: 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);
```

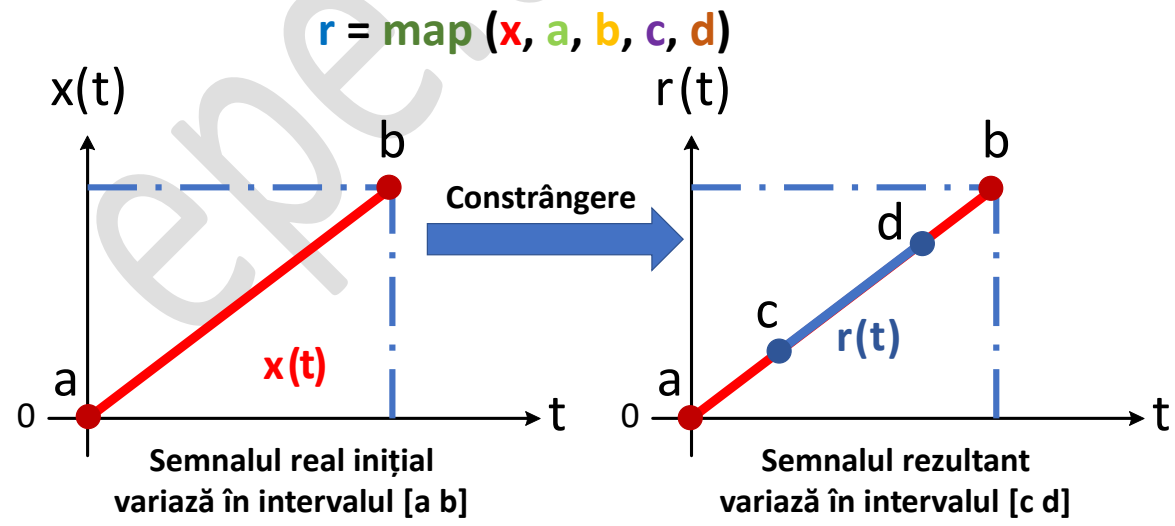

Fig. 7 - Restrângerea sau scalarea unui interval finit [1] [2]

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

C. Introducerea unui algoritm de mediere în timp a semnalului:

Semnalul analogic preluat de la convertorul analog – digital, în caz real este susceptibil zgomotelor şi erorilor de achiziţie. Astfel, unele valori achiziţionate nu reprezintă tocmai mărimea reală, se procedează deci, la excluderea datelor nesemnificative din şirul de măsurători (Fig. 8). Media aritmetică, reprezintă una din metodele numerice de filtrare a semnalului, şi poate fi exprimată sau implementată prin intermediul următoarei relaţii de calcul:

$$X(t) = \frac{1}{N} \cdot \sum_{i=1}^{N} x(t)$$

Unde „X(t)" reprezintă media aritmetică a valorii instantanee „x(t)", „N" este numărul de eşantioane sau măsurători achiziţionate, iar „i" reprezintă o variabilă – contor.

Algoritmul de mediere se implementează cu ajutorul unei structuri iterative „for":

```
void loop {
      int suma = 0;
            for (int i = 1; i <= N_măsurători; i++) {
                  x = analogRead (pin_analogic);
                  suma += x;
            }
            X = suma / N_măsurători;
}
```

OBSERVAŢII:
- algoritmul se implementează în bucla infinită „void loop()";
- variabila „suma" trebuie iniţializată cu valoarea zero la începutul structurii „void loop()", pentru a realiza re-iniţializarea algoritmului pentru fiecare ciclu.

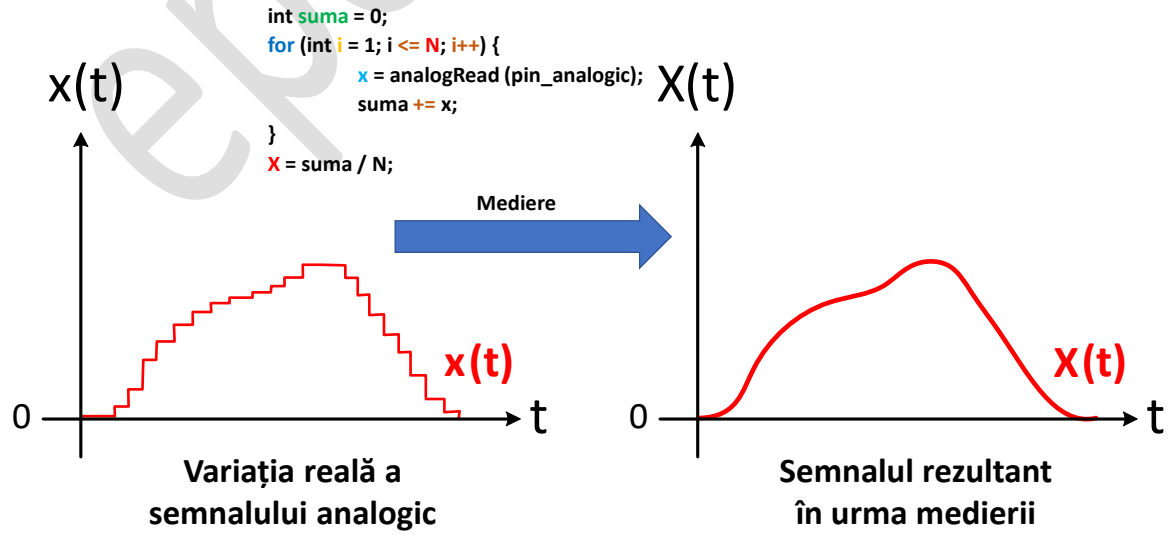

Fig. 8 – Media aritmetică a semnalului de măsură [1] [2]

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

IV. IMPLEMENTAREA APLICAȚIILOR:

În vederea implementării aplicațiilor din cadrul acestei lucrări, vor fi utilizate două module specializate precum:
- traductorul de distanță HC – SR04 [3];
- modulul LCD QAPASS cu adaptor I²C M.H. [4] [5];

A. Principiul de funcționare al traductorului HC – SR04:

Traductorul HC – SR04, funcționează pe baza efectului de reflexie și propagare a undei sonore. Modulul are în componență un dispozitiv transmițător („T") de tip difuzor de înaltă frecvență pe baza efectului piezoelectric și un receptor („R") de tip microfon de înaltă frecvență, funcționând pe baza efectului invers piezoelectric (Fig. 9) [3].

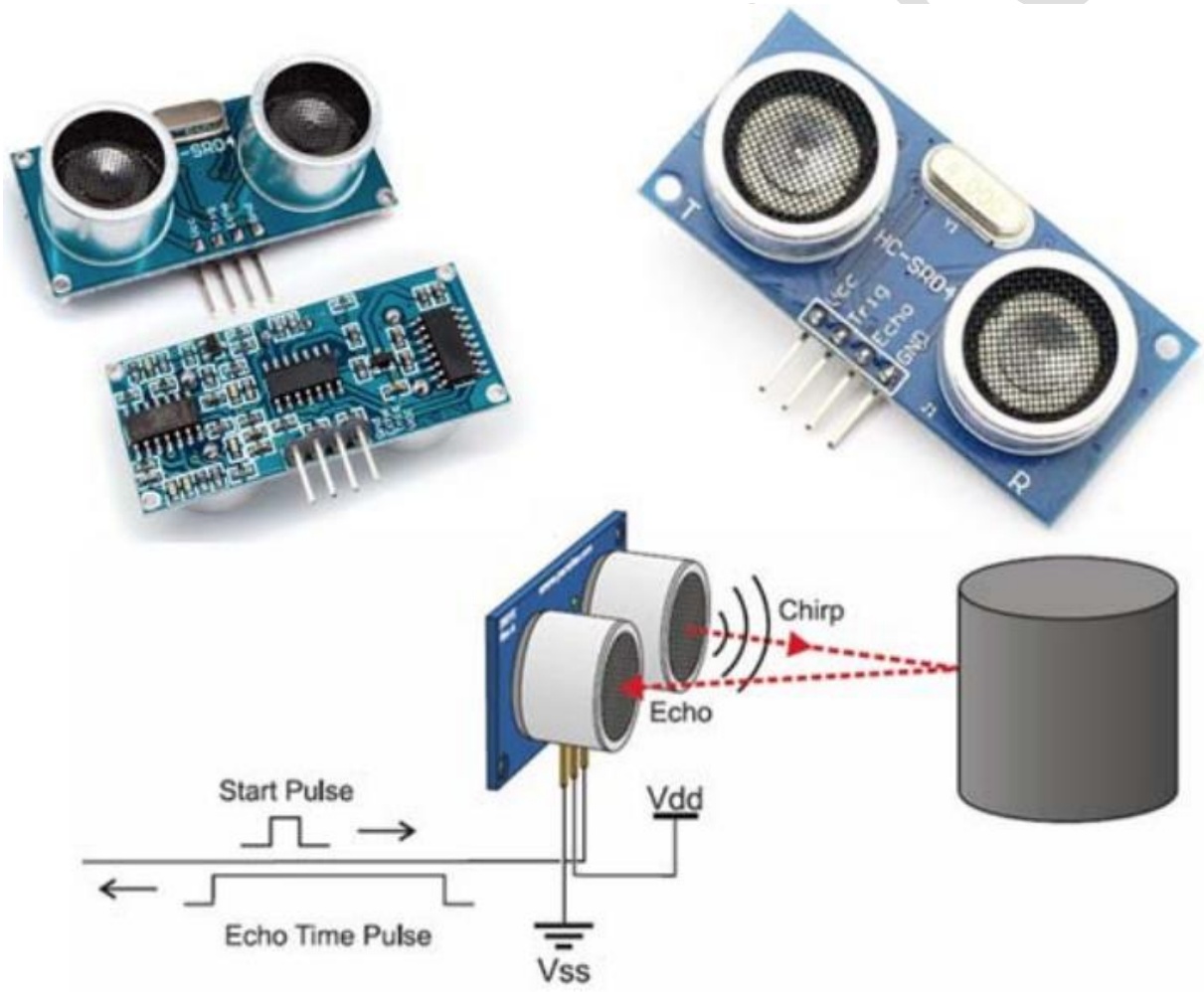

Fig. 9 – Traductorul HC-SR04 [3]

Frecvența undei sonore pe care o vehiculează modulul traductor este 40 [kHz]. Raza de acțiune este între 3 [cm] și 4 [m] la un unghi de incidență de 15°. Rezoluția aproximativă de măsură este de 1 [cm] / pas. Tensiunea de alimentare a modulului este 5 [V]. Semnalele furnizate la ieșirea modulului sunt de natură digitală (Fig. 10) [3].

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Pe baza frecvenţei undei sonore şi a vitezei sale de deplasare se poate estima distanţa de la modul până la obstacolul întâmpinat în calea de propagare conform relaţiilor de calcul, cunoscând constanta v_{sunet} = 340 [m/s]:

$$v = \frac{d}{t} \rightarrow t = \frac{d}{v} \rightarrow d = \frac{0.034 \cdot t}{2} \quad [cm]$$

Microcontrolerul măsoară distanţa pe baza timpul de răspuns echivalent timpului de propagare al undei sonore prin mediul de transmitere (Fig. 10).

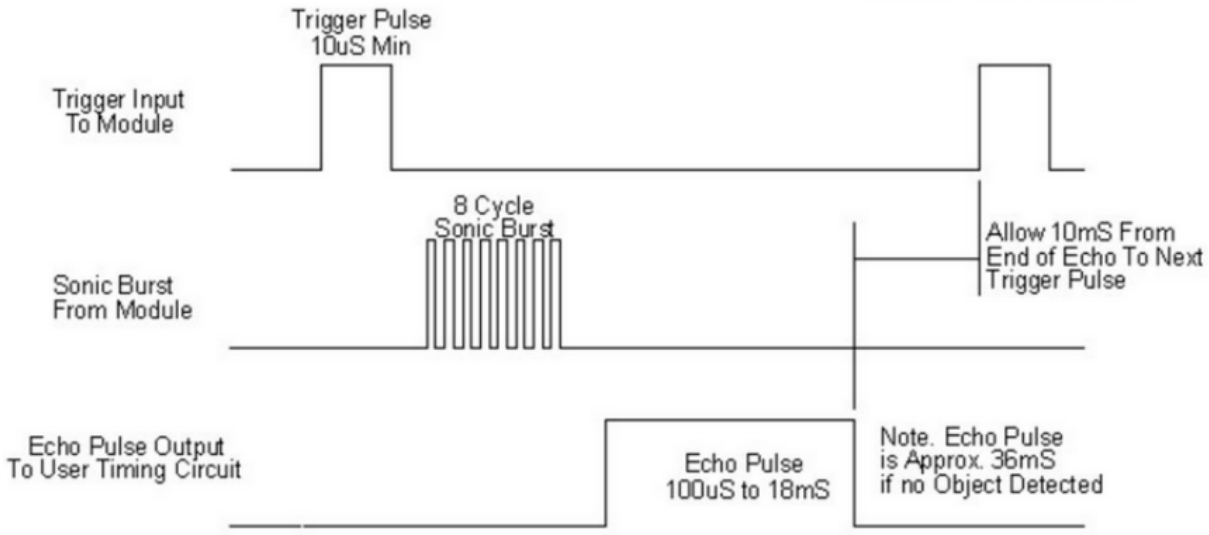

Fig. 10 – Semnalele vehiculate de modulul HC-SR04 [3]

B. Principiul de funcţionare al modulului cu afişaj LCD:

Principiul de funcţionare al afişajului LCD (eng. Liquid Crystal Display) se bazează pe efectul de magnetizare al cernelii ferofluidice (cristale lichide). În construcţia afişajului se regăseşte o matrice de bobine având rolul de electromagneţi. Prin intermediul micro-electro-magneţilor cerneala ferofluidică este dirijată printr-un sistem de vase comunicante transparente care au forma caracterelor ce pot fi afişate (Fig. 11) [4].

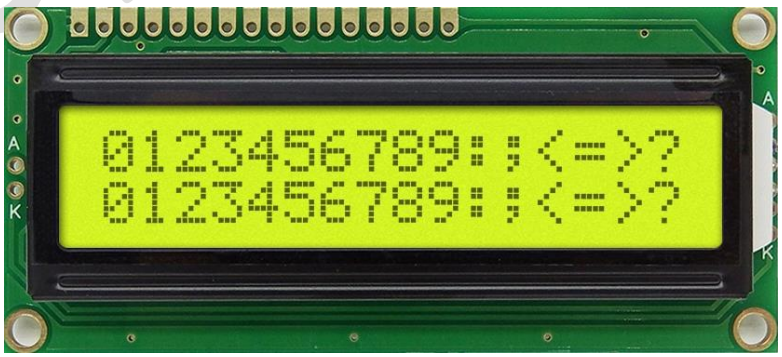

Fig. 11 – Afişaj LDC caracter [4]

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Modulul LCD caracter utilizat în această lucrare dispune de 32 celule caracter dispuse pe două rânduri, rezultând deci o rezoluţie 16 x 2. Caracterul se construieşte pe baza unor matrici de puncte (celulele). Pachetul modular dispune de 16 terminale de racord (cu posibilitatea reglării luminiţei de fond). Substratul constructiv de siliciu înglobează un microcontroller pentru gestionarea modulului. Acceptă comenzi pe 4 sau 8 biţi vehiculaţi în mod paralel la terminalele de date (DB0 – DB7). Pot fi transferate seturi suplimentare de caractere în memoria modulului LCD [4].

Pentru a reduce numărul de terminale conectate la platforma de dezvoltare Arduino Nano, împreună cu modulul LCD se va utiliza adaptorul pentru interfaţă de comunicare de tip Paralel la interfaţă de tip I²C (eng. LCD I²C back Pack). În acest sens, pentru conectarea modulului LCD la platforma de dezvoltare Arduino Nano, se vor utiliza patru terminale, anume [5]:

- Vcc – pentru alimentare la 5 [V];
- GND – pentru conectare la potenţial zero;
- SDA (eng. Serial Data) – pentru conectare la terminalul A4;
- SCL (eng. Serial Clock) – pentru conectare la terminalul A5;

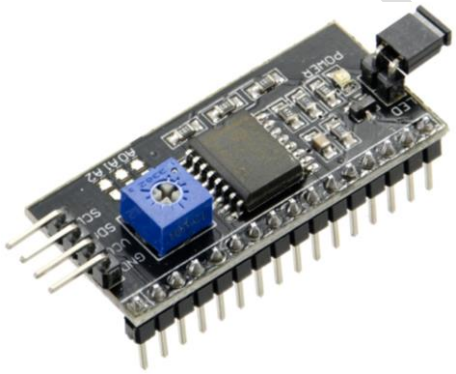

Fig. 12 – Adaptor I²C la paralel pentru afişaj LCD [5]

Pentru a utiliza afişajul LCD împreună cu adaptorul I²C la interfaţă Paralel, este necesară instalarea şi includerea bibliotecii de funcţii „Wire.h” pentru iniţializarea comunicaţiei pe interfaţa I²C şi a bibliotecii de funcţii „LiquidCrystal_I2C.h” [5].

Pentru a preveni efectul remanent al caracterelor de pe ecran este necesară implementarea unui algoritm pentru ştergere şi actualizare a zonei de afişare. În principal, acest fenomen poate fi observat la afişarea numerelor fracţionare. Odată cu afişarea unui sub-multiplu, celula din faţă se va popula cu un caracter „0”. Procesul de eliminare a caracterelor remanente sau a cifrelor „0” din faţa fracţiei (ex. 005,0) poartă denumirea (în literatura de specialitate) de (eng.) Leading Zeros Clearing, spre exemplu:

```
if(media < 1000) {          // verificarea numărului de caractere
lcd.setCursor(3,1);         // plasarea cursorului de scriere la coordonate
lcd.print(" ");       // afişarea unui spaţiu gol la coordonate
lcd.setCursor(0,1);         // plasarea cursorului de scriere la coordonate
lcd.print(media);           // afişarea variabilei la coordonate
}
```


Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Pentru a evidenția conceptele prezentate anterior, se propune deci implementarea următoarelor aplicații cu ajutorul platformei Arduino Nano:
- controlul digital al factorului de umplere pentru un semnal modulat în lățime;
- preluarea temperaturii de la un traductor și filtrarea zgomotelor;
- măsurarea distanței cu ajutorul traductorului ultrasonic HC – SR04;
- utilizarea unui afișaj de tip LCD;
- implementarea unui termometru digital;
- implementarea unei rigle digitale;

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;
- butoane cu apăsare și revenire;
- rezistențe cu valoarea de 100 [Ω];
- rezistențe cu valoarea de 10 [kΩ];
- senzor de temperatura LM-35;
- traductor de distanță pe bază de efect ultrasonic HC-SR04;
- modulul pentru afișare LCD QAPASS cu adaptor I²C la Paralel M.H.;
- fire pentru conexiune rapidă compatibile cu placa de testare;
- calculator gazdă având mediul Arduino IDE instalat;
- cablu adaptor USB A la mini USB;

APLICAȚIA 1:

Se va implementa circuitul conform următoarei scheme (Fig. 13):

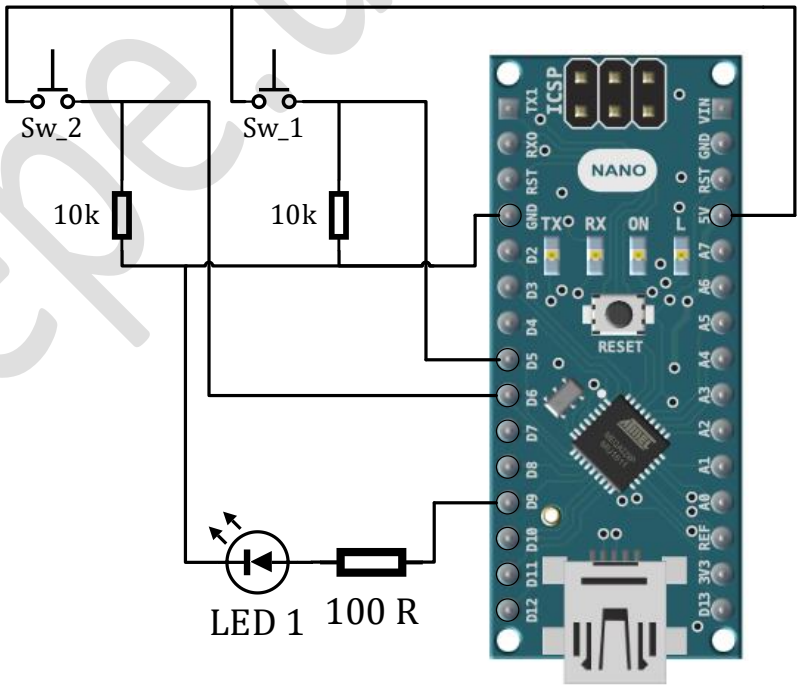

Fig. 13 – Schema electronică pentru implementarea circuitului specific aplicației 1

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 pin_sw_1 = 5;
const int pin_sw_2 = 6;
const int pin_led = 9;

int sw_1_state = 0;
int sw_2_state = 0;
int dc = 0;

void setup() {
  pinMode(pin_sw_1, INPUT);
  pinMode(pin_sw_2, INPUT);
  pinMode(pin_led, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  sw_1_state = digitalRead(pin_sw_1);
  sw_2_state = digitalRead(pin_sw_2);

  if(sw_1_state == 1) {
    dc++;
    if (dc >= 255) {
      dc = 255;
    }
  }

  if(sw_2_state == 1) {
    dc--;
    if (dc <= 0) {
      dc = 0;
    }
  }
  Serial.print("Factor de umplere: ");
  Serial.print(dc);
  Serial.println("");
  analogWrite(pin_led, dc);
  delay(10);
}
```


Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

Montajul experimental se va realiza conform (Fig. 14):

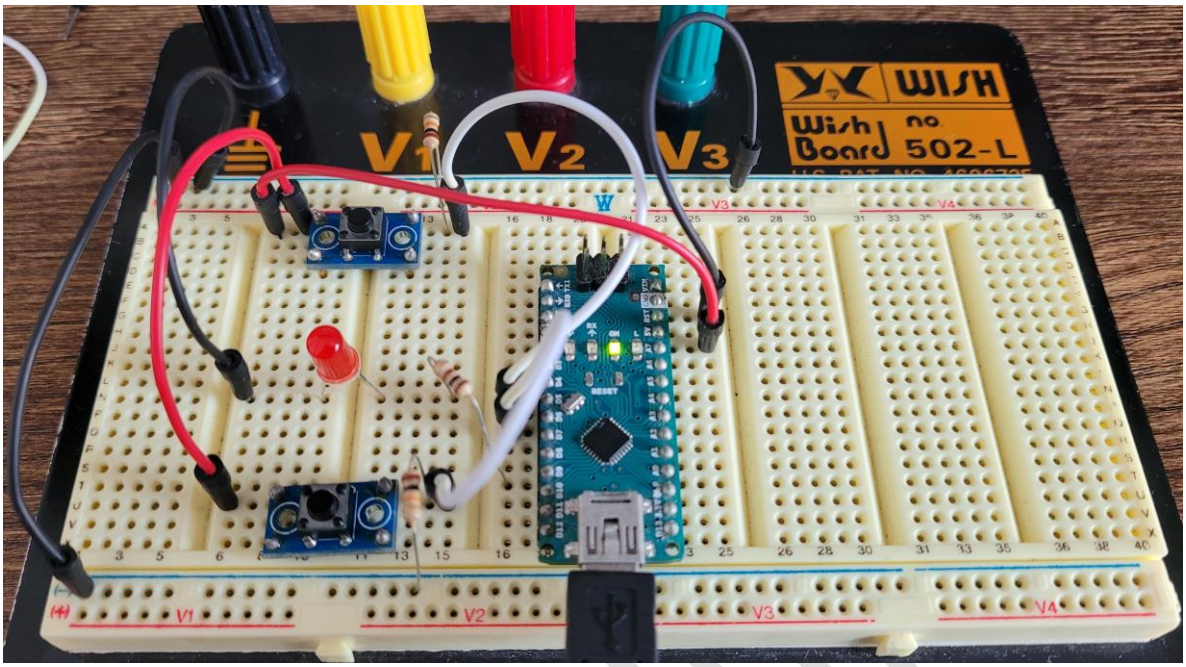

Fig. 14 – Montaj experimental pentru aplicaţia 1

Implementarea aplicaţiei nr. 1 presupune:

- declararea a trei constante globale, în vederea stabilirii numerelor de ordine ale terminalelor corespondente atât butoanelor cu apăsare şi revenire cât şi diodei (LED);
- stabilirea modului de lucru „intrare digitală" pentru terminalele butoanelor;
- stabilirea modului de lucru „ieşire digitală" pentru terminalul diodei (LED);
- preluarea stărilor logice ale terminalelor alese pentru atașarea butoanelor cu apăsare și revenire prin intermediul instrucțiunii „digitalRead ()";
- incrementarea sau decrementarea factorului de umplere „dc" la menţinerea apăsată sau apăsarea repetată a butoanelor;
- impunerea unor constrângeri simple;
- afișarea în consola Serial a valorii pentru reglarea factorului de umplere (Fig. 15 / 16);
- generarea unui semnal modulat în lățime prin intermediul funcției „analogWrite()";

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

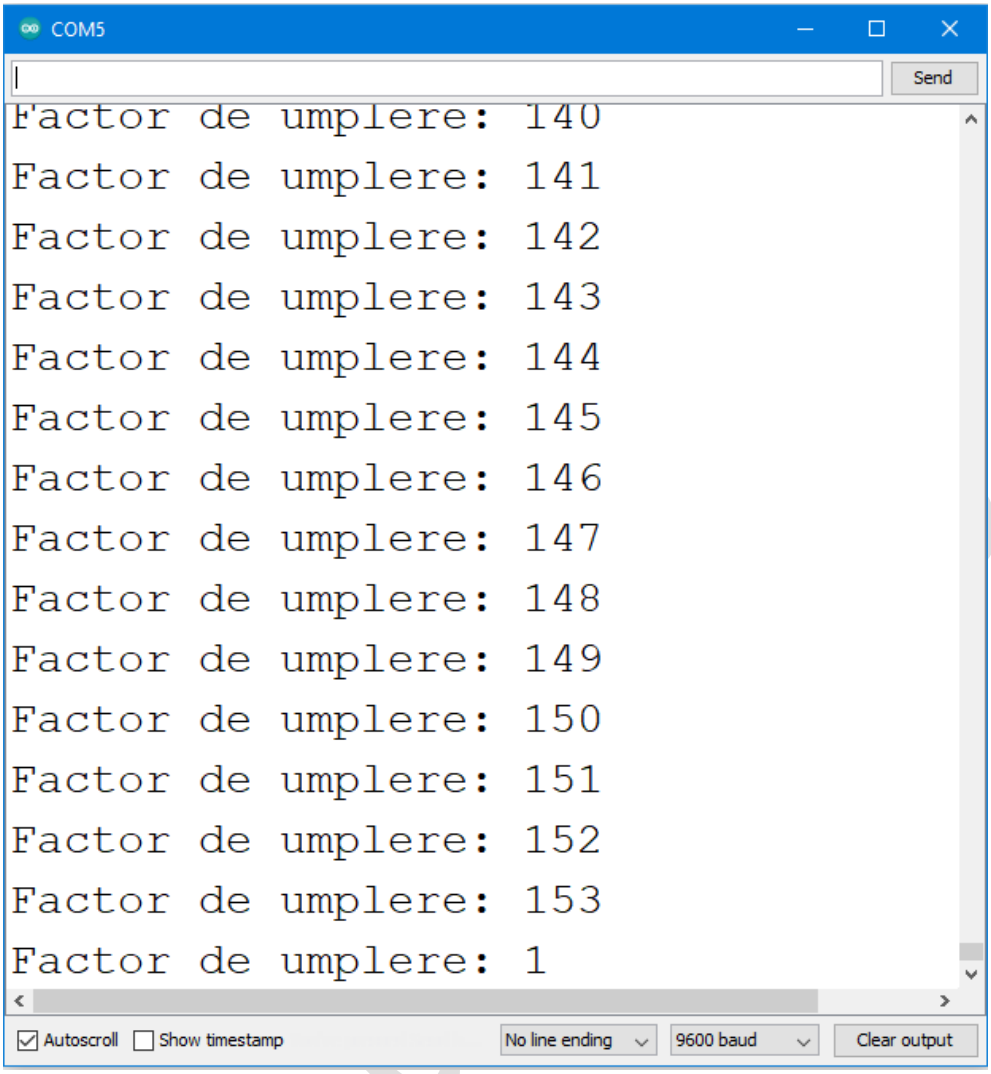

Fig. 15 – Afișarea valorii zecimale a factorului de umplere în consola Serial

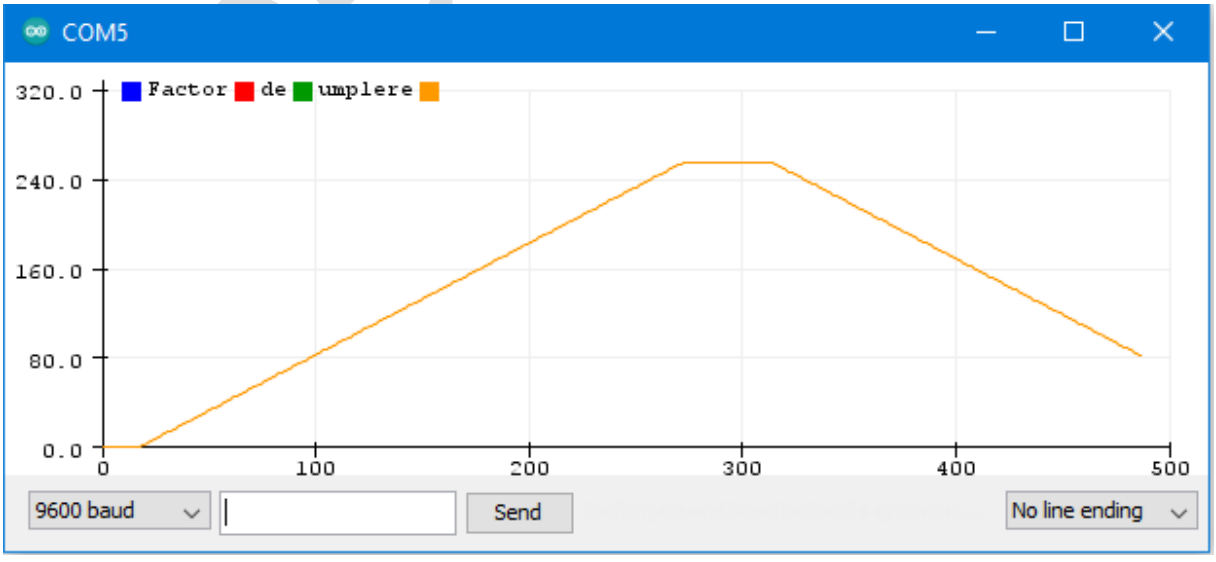

Fig. 16 – Afișarea valorii zecimale a factorului de umplere în consola grafică

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAȚIA 2:

Se va implementa circuitul conform următoarei scheme (Fig. 17):

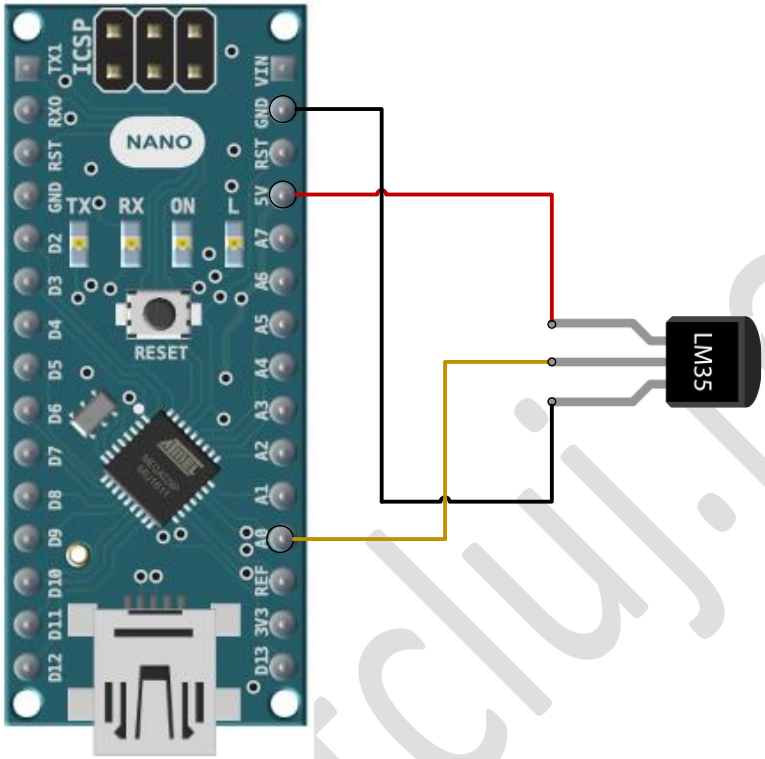

Fig. 17 – Schema electronică pentru implementarea circuitului specific aplicației 2

Se va realiza următorul montaj experimental (Fig. 18):

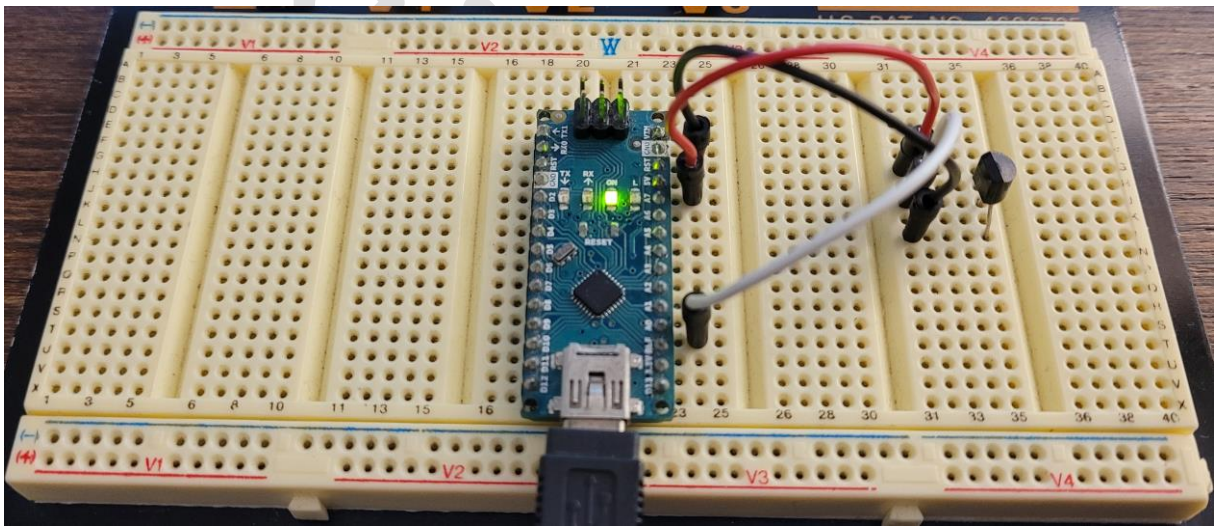

Fig. 18 – Montajul experimental specific aplicației 2

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;
int N = 500;
float U = 0.00;
float temp = 0.00;
float media = 0.00;

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

void loop() {
  float suma = 0.00;
  for (int i = 1; i <= N; i++) {
  ADC_val = analogRead(analog_pin);
  U = (4.80 / 1023.00) * ADC_val;
  temp = 100.00 * U;
  suma += temp;
  }
  media = suma / N;
  Serial.print("Temperatura: ");
  Serial.print(media);
  Serial.print(" [*C]");
  Serial.println("");
  delay(250);
}
```

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 număr întreg „N” 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 unei variabile de tip fracționar „media” cu valoarea „0.00”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s];
- inițializarea unei variabile locale cu denumirea „suma”;
- 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;
- însumarea a 500 de valori prin intermediul structurii iterative de tip „for ()”;
- determinarea mediei aritmetice prin intermediul raportului dintre suma tuturor valorilor înregistrate în variabila „suma” și numărul de iterații „N”;
- afișarea valorii medii a temperaturii în consola grafică;

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

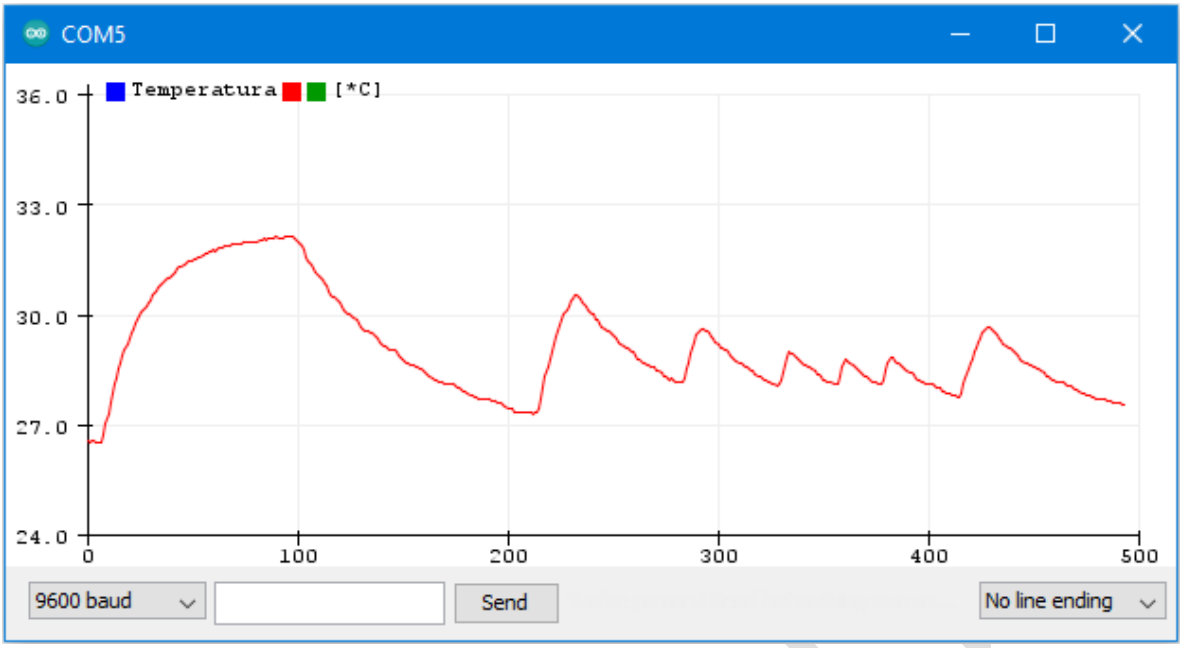

Fig. 19 – Afișarea valorilor medii ale temperaturii în consola grafică

APLICAȚIA 3:

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

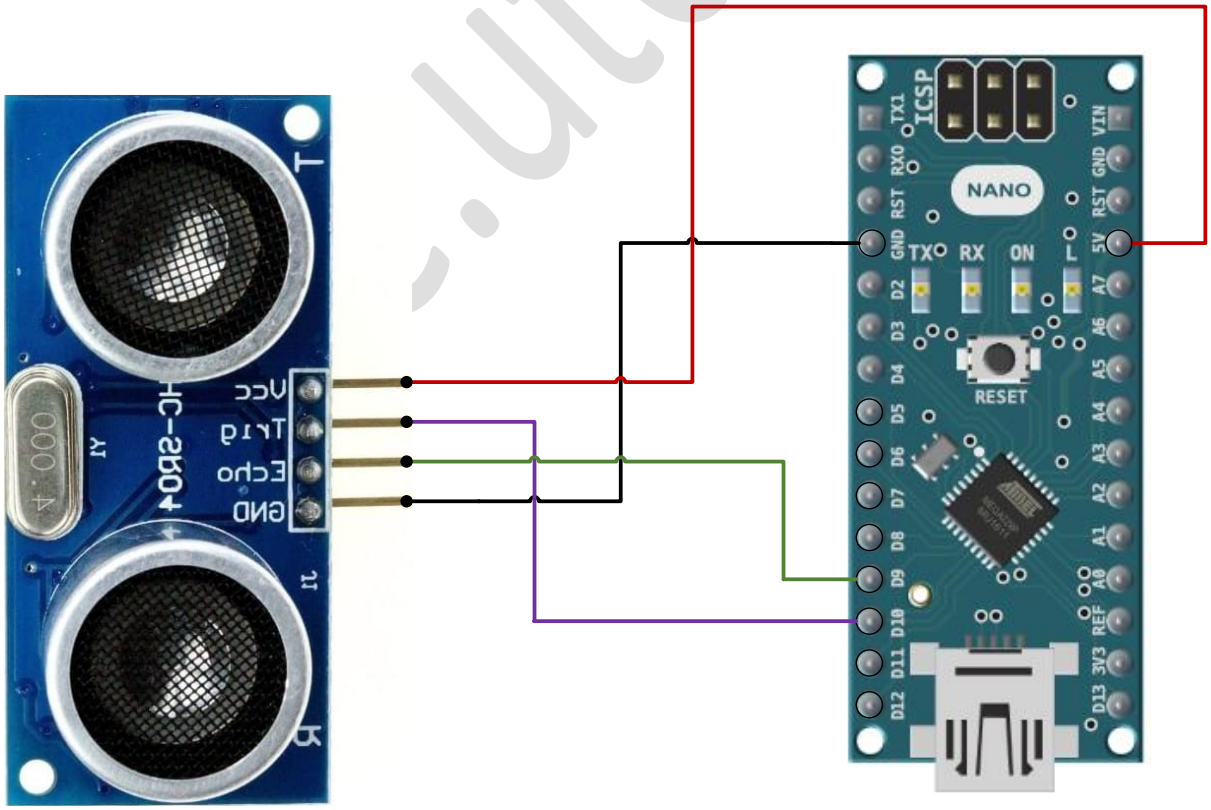

Fig. 20 – Schema electronică pentru implementarea circuitului specific aplicației 3

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

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

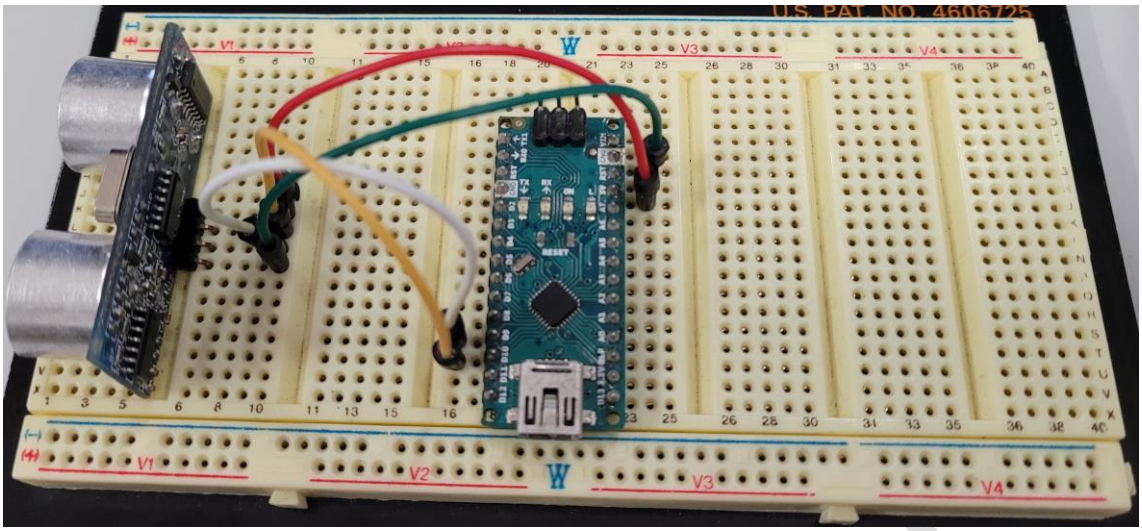

Fig. 21 - Montajul experimental specific aplicației 3

Se va implementa următorul cod program:

```
const int echo = 9;
const int trigger = 10;
long t = 0.0;
long d = 0.0;

void setup() {
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);
  t = pulseIn(echo, HIGH);
  d = t * 0.034 / 2.0;
  Serial.print("Distanta: ");
  Serial.print(d);
  Serial.print(" [cm]");
  Serial.println("");
  delay(100);
}
```


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. 3 presupune:

- declararea unei constante de tip număr întreg „echo” având ca și valoare „9”;
- declararea unei constante de tip număr întreg „trigger” având ca și valoare „10”;
- inițializarea unei variabile de tip număr întreg „t” cu valoarea „0”;
- inițializarea unei variabile de tip număr întreg „d” cu valoarea „0”;
- stabilirea modului de lucru „ieșire digitală” pentru terminalul „9”;
- stabilirea modului de lucru „intrare digitală” pentru terminalul „10”;
- inițializarea comunicației Serial;
- generarea unei secvențe de semnalizare acustică la o frecvență de 40 [kHz];
- determinarea timpului de propagare a undei sonore cu ajutorul funcției „pulseIn ()”;
- determinarea distanței pe baza formulei de calcul a vitezei de propagare a undei sonore;
- afișarea în consola Serial a rezultatului măsurării (Fig. 22);

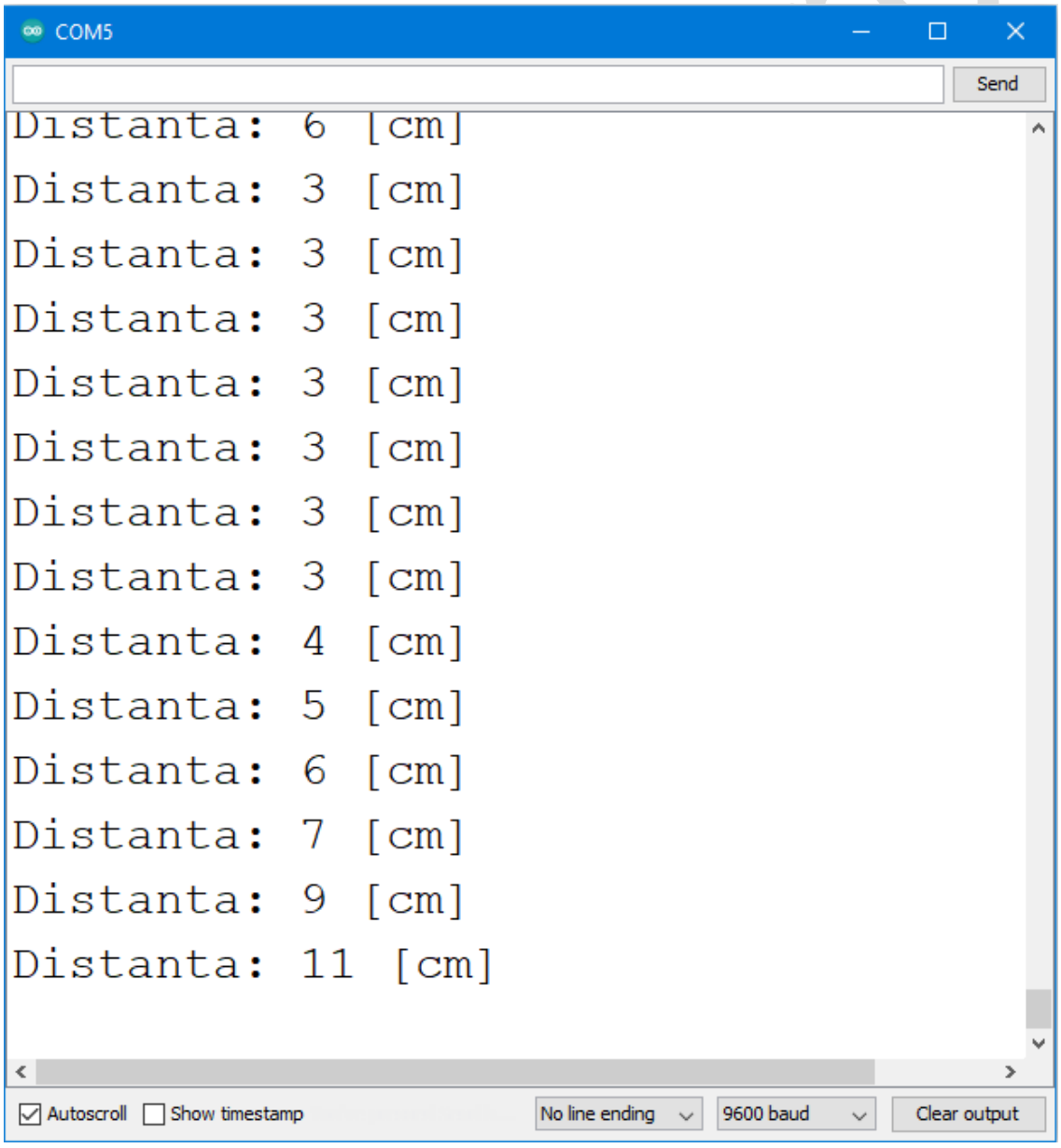

Fig. 22 – Afișarea în consola Serial a rezultatului de măsurare a distanței

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAŢIA 4:

Se va implementa circuitul conform următoarei scheme (Fig. 23):

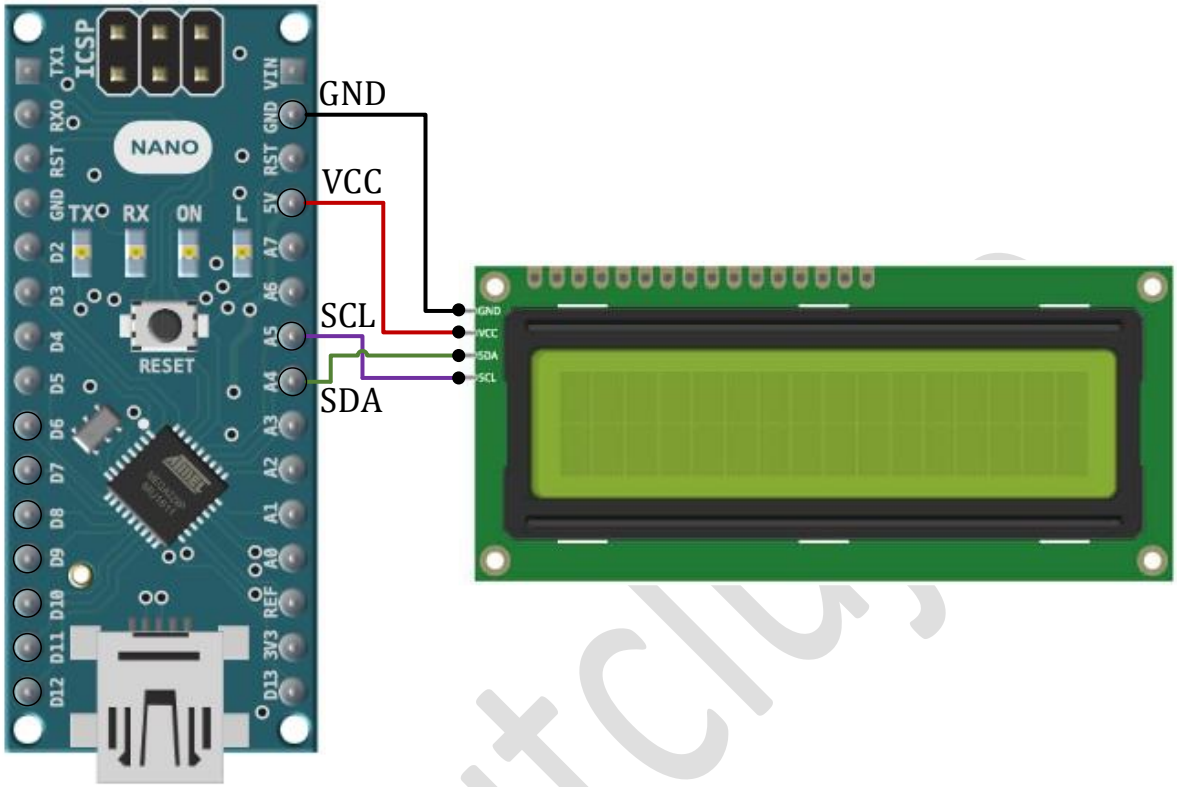

Fig. 23 - Schema electronică pentru implementarea circuitului specific aplicaţiei 4

Se va realiza următorul montaj experimental (Fig. 24):

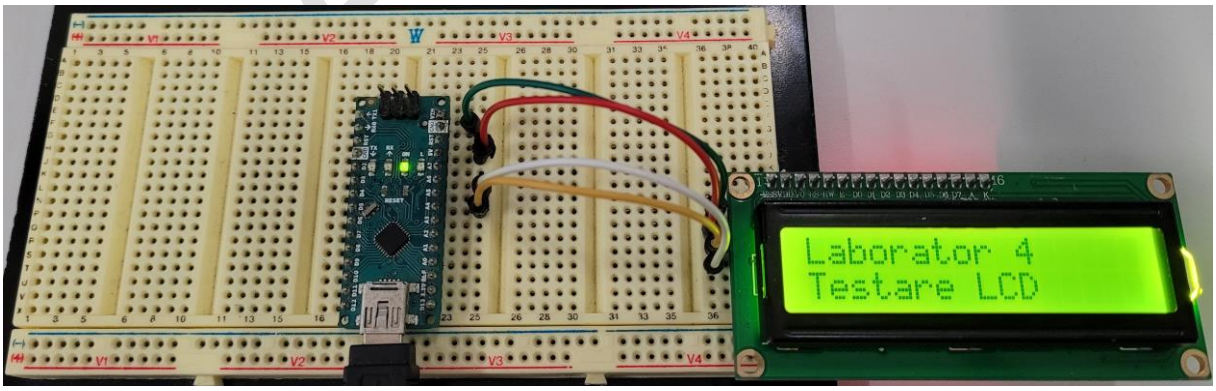

Fig. 24 - Montajul experimental specific aplicaţiei 4

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:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

void setup() {
  lcd.begin(16,2);
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight();
}

void loop() {
  lcd.setCursor(0,0);
  lcd.print("Laborator 4");
  lcd.setCursor(0,1);
  lcd.print("Testare LCD");
}
```

Implementarea aplicației nr. 4 presupune:

- inițializarea bibliotecii de funcții „Wire.h" pentru comunicația I²C;
- inițializarea bibliotecii de funcții „LiquidCrystal_I2C.h";
- inițializarea parametrilor aferenți obiectului „lcd" (ex. adresa I²C și terminale);
- inițializarea rezoluției afișajului prin intermediul funcției „lcd.begin ()";
- inițializarea luminiței de fundal a ecranului printr-o secvență de licărire;
- menținerea permanent aprinsă a luminiței prin intermediul funcției „lcd.backlight ()";
- plasarea cursorului de scriere cu ajutorul funcție „lcd.setCursor ()";
- afișarea unui mesaj pe ecran cu ajutorul funcției „lcd.print ()";

OBSERVAȚII:

1. Adresa dispozitivului adaptor al ecranului atașat pe interfața I²C (în cazul de față) este „0x27". Aceasta se poate determina pe baza datelor de catalog sau prin intermediul codului program „i2c_scanner" regăsit în spațiul public al domeniului „arduino.cc". Cele mai uzuale adrese ale dispozitivelor adaptoare sunt „0x27" și „0x3F".

2. Funcția „lcd.SetCursor" are ca și argumente lcd.SetCursor(<celulă>, <rând>);

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAȚIA 5:

Se va implementa circuitul conform următoarei scheme (Fig. 25):

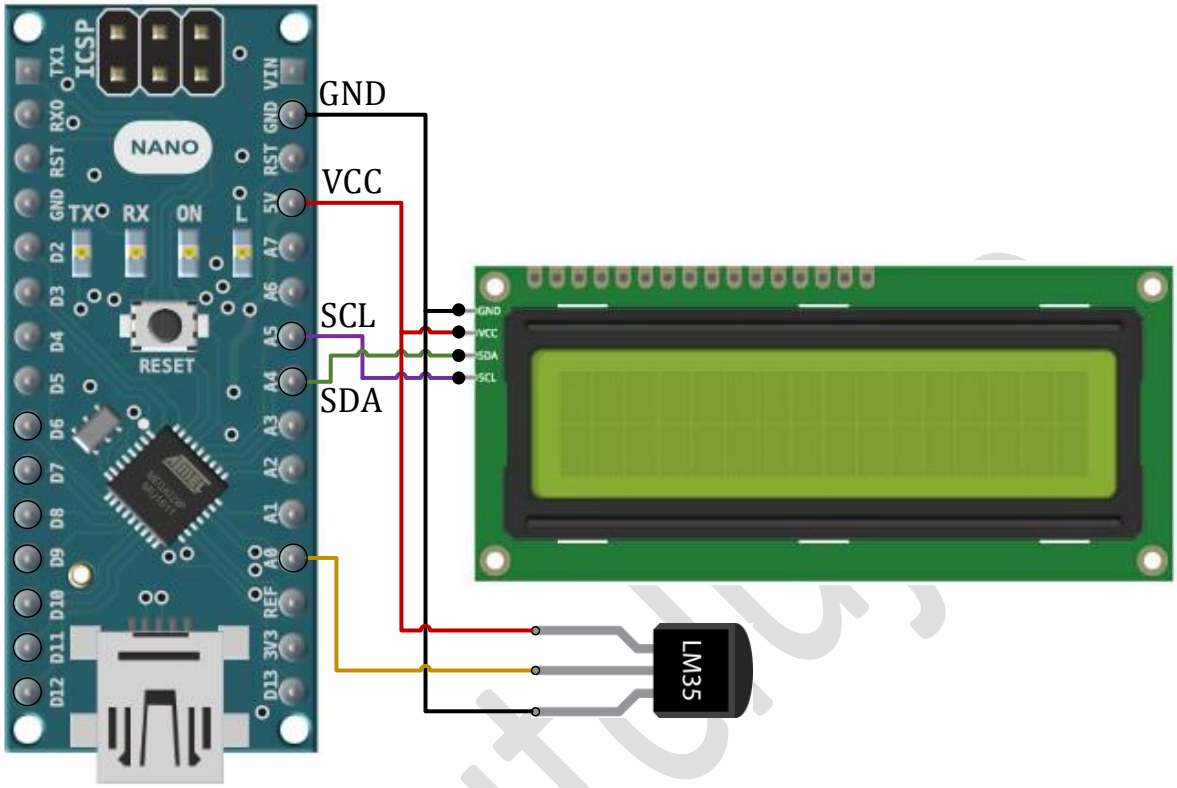

Fig. 25 - Schema electronică pentru implementarea circuitului specific aplicației 5

Se va realiza următorul montaj experimental (Fig. 26):

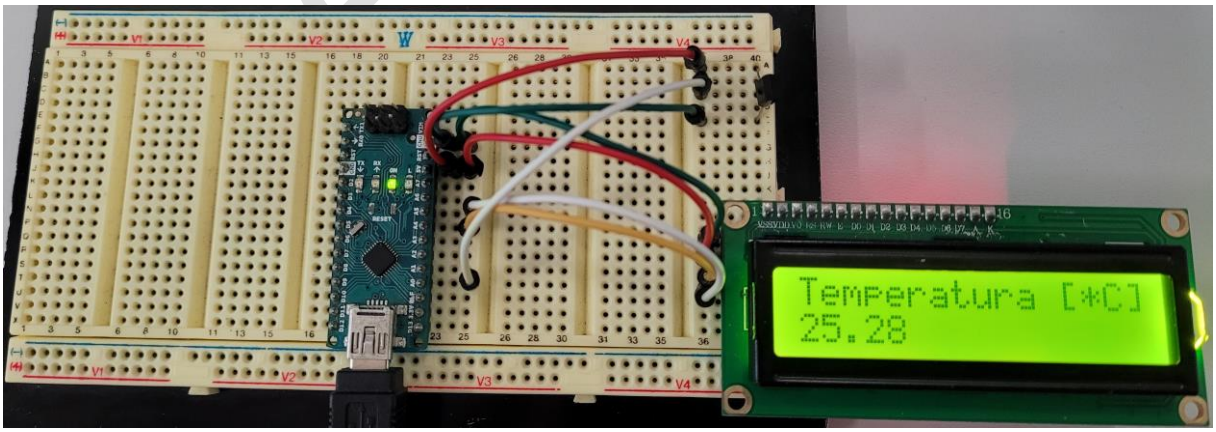

Fig. 26 - Montajul experimental specific aplicației 5

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:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
const int analog_pin = 0;
int ADC_val = 0;
int N = 500;
float U = 0.00;
float temp = 0.00;
float media = 0.00;

void setup() {
  lcd.begin(16,2);
  for(int i = 0; i< 3; i++) {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Temperatura [*C]: ");
}

void loop() {
  float suma = 0.00;

  for (int i = 1; i <= N; i++) {
  ADC_val = analogRead(analog_pin);
  U = (4.78 / 1023.00) * ADC_val;
  temp = 100.00 * U;
  suma += temp;
  }

  media = suma / N;

  lcd.setCursor(0,1);

    if(media < 1000) {
    lcd.setCursor(3,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(media);
  }
```


Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

```
    if(media < 100) {
    lcd.setCursor(2,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(media);
  }
    if(media < 10) {
    lcd.setCursor(1,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(media);
  }
  delay(500);
}
```

Implementarea aplicației nr. 5 presupune:

- inițializarea bibliotecii de funcții „Wire.h" pentru comunicația I^2C;
- inițializarea bibliotecii de funcții „LiquidCrystal_I2C.h";
- inițializarea parametrilor aferenți obiectului „lcd" (ex. adresa I^2C și terminale);
- inițializarea unui terminal analogic „A0";
- inițializarea variabilelor pentru determinarea conținutului registrului convertorului analog – digital, a tensiunii de măsură și a temperaturii medii;
- inițializarea rezoluției afișajului prin intermediul funcției „lcd.begin ()";
- inițializarea luminiței de fundal a ecranului printr-o secvență de licărire;
- afișarea mesajului inițial „Temperatura [*C]: " pe primul rând la începutul ecranului;
- menținerea permanent aprinsă a luminiței prin intermediul funcției „lcd.backlight ()";
- determinarea temperaturii medii pe baza datelor furnizate de către traductorul LM-35 conectat la o intrare analogică a convertorului analog – digital (similar aplicației 2);
- plasarea cursorului de scriere cu ajutorul funcție „lcd.setCursor ()";
- afișarea unui mesaj și a valorii măsurate pe ecran cu ajutorul funcției „lcd.print ()";
- actualizarea valorilor numerice prin intermediul algoritmului „Leading Zeros Clearing";

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

APLICAȚIA 6:

Se va implementa circuitul conform următoarei scheme (Fig. 27):

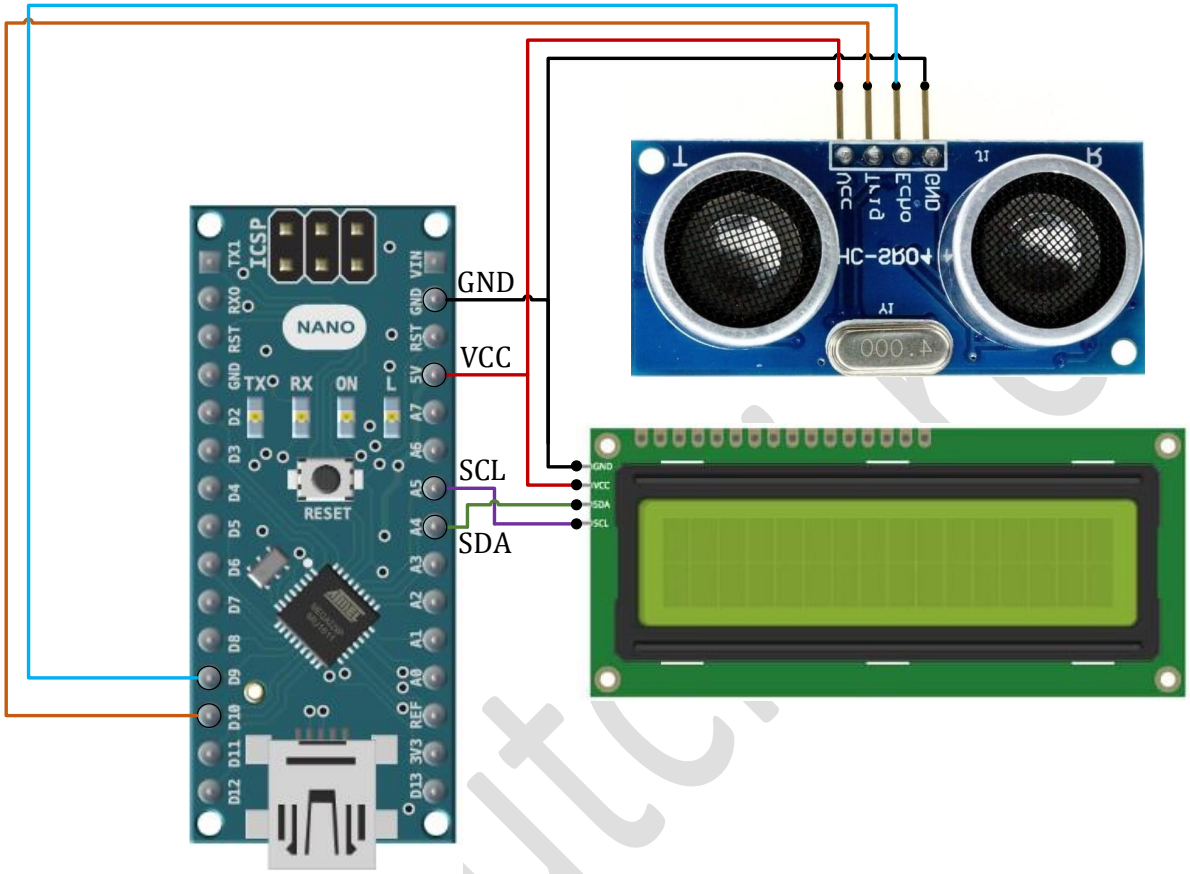

Fig. 27 – Schema electronică pentru implementarea circuitului specific aplicației 6

Se va realiza următorul montaj experimental (Fig. 28):

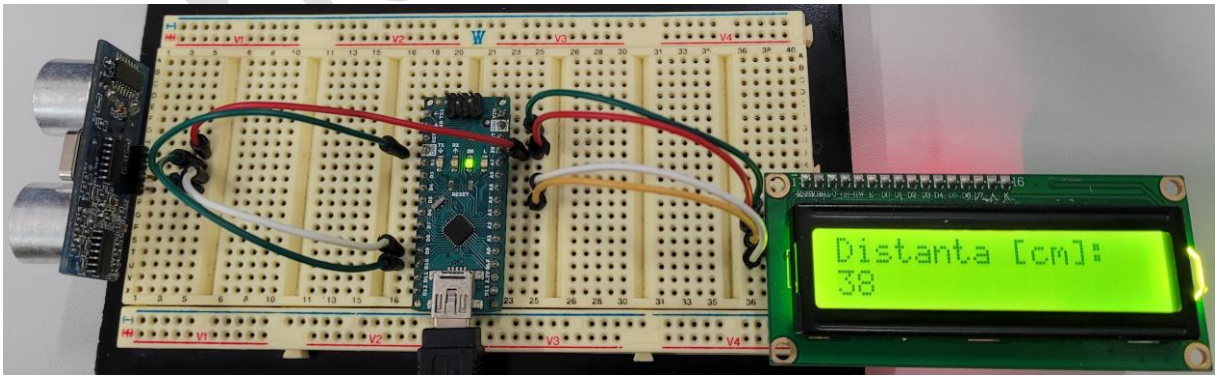

Fig. 28 – Montajul experimental specific aplicației 6

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:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

const int echo = 9;
const int trigger = 10;
long t = 0.0;
long d = 0.0;

void setup() {
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
  lcd.begin(16,2);
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Distanta [cm]: ");
}

void loop() {
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);

  t = pulseIn(echo, HIGH);
  d = t * 0.034 / 2.0;

  lcd.setCursor(0,1);

    if(d < 1000) {
    lcd.setCursor(3,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(d);
  }
```


Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

```
    if(d < 100) {
    lcd.setCursor(2,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(d);
  }
    if(d < 10) {
    lcd.setCursor(1,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(d);
  }
  delay(500);
}
```

Implementarea aplicației nr. 6 presupune:

- inițializarea bibliotecii de funcții „Wire.h" pentru comunicația I^2C;
- inițializarea bibliotecii de funcții „LiquidCrystal_I2C.h";
- inițializarea parametrilor aferenți obiectului „lcd" (ex. adresa I^2C și terminale);
- inițializarea unui terminalului „D10" ca și ieșire digitală;
- inițializarea unui terminalului „D9" ca și intrare digitală;
- inițializarea variabilelor pentru determinarea duratei de propagare și a distanței;
- inițializarea rezoluției afișajului prin intermediul funcției „lcd.begin ()";
- inițializarea luminiței de fundal a ecranului printr-o secvență de licărire;
- afișarea mesajului inițial „Distanta [cm]: " pe primul rând la începutul ecranului;
- menținerea permanent aprinsă a luminiței prin intermediul funcției „lcd.backlight ()";
- determinarea distanței pe baza traductorului HC – SR04 (similar aplicației 3);
- plasarea cursorului de scriere cu ajutorul funcție „lcd.setCursor ()";
- afișarea unui mesaj și a valorii măsurate pe ecran cu ajutorul funcției „lcd.print ()";
- actualizarea valorilor numerice prin intermediul algoritmului „Leading Zeros Clearing";

V. CONCLUZIE:

Aplicațiile implementate pe baza microcontrolerelor pot funcționa în mod independent de calculator dacă există mijloace periferice de intrare sau ieșire atașate la terminalele microcontrolerului care pot înlocui rolul calculatorului gazdă.

Funcționarea în mod independent de calculatorul gazdă al aplicației finale presupune aplicarea unor algoritmi de optimizare atât al modului de funcționare cât și al modului de execuție specific codului program.

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro

VI. BIBLIOGRAFIE:

1. 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

2. 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/

3. Handson Technology – „HC-SR04 Ultrasonic Sensor Module User Guide” - Ultrasonic Sensor V2.0, SKU: MDU-1014 - www.handsontec.com
https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf

4. Shenzhen Eone Electronics CO. LTD. – „Specification for LCD module 1602A-1 V1.2”
https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf

5. Handson Technology – „I2C Serial Interface 1602 LCD Module” - SKU: DSP-1182
https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

Conf. Dr. Ing. Ioana – Cornelia Gros – Ioana.Gros@emd.utcluj.ro
Asist. Dr. Ing. Lucian – Nicolae Pintilie – Lucian.Pintilie@emd.utcluj.ro