

Sisteme cu microprocesoare

- Protocoale de comunicație utilizate în sisteme cu microprocesoare -

I. SCOPUL LUCRĂRII:

Lucrarea de laborator are ca scop:

- prezentarea unor protocoale de comunicație în sisteme cu microprocesoare [1], [2]
- prezentarea unor dispozitive compatibile cu protocoalele de comunicație [3], [4], [5], [6]
- utilizarea microcontrolerelor pentru interfațarea protocoalelor de comunicație [5], [6], [7]

II. INTRODUCERE: [1], [2], [3]

Un **protocol de comunicație**, reprezintă un **mod de asociere** și **transferare de date** între cel puțin **două** sisteme de calcul, pe baza **circuitelor periferice** aflate în dotarea sistemului de calcul. **Cadrul de date** vehiculat între cele două echipamente reprezintă orice **combinație alfa-numerică** exprimată prin intermediul protocolului. O **grupare de cadre de date**, reprezintă un **pachet de date**. În unele situații, cadrele de date reprezintă un singur bit.

În domeniul sistemelor cu microprocesoare, există două categorii principale de protocoale de comunicație, anume:

- protocoale de comunicație cu transmitere de date în mod paralel;
- protocoale de comunicație cu transmitere de date în mod serial;

Protocoalele de comunicație cu transmitere de date în mod **paralel**, pot vehicula cadrele de date pe **mai multe canale de comunicație** în **mod simultan**. Mai precis prin intermediul a patru intrări / ieșiri digitale, se pot vehicula 2^4 combinații (Fig. 1).

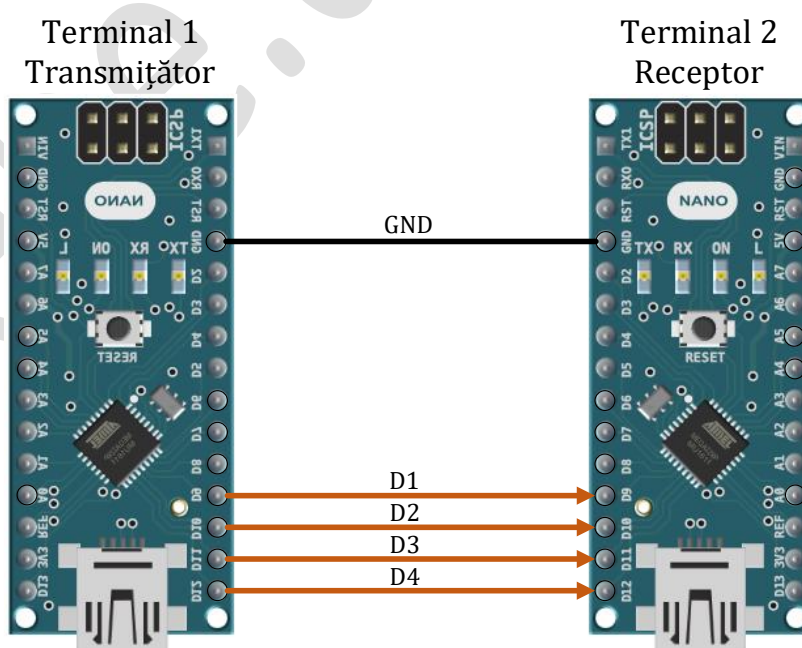


Fig. 1 – Protocolul de comunicație cu transmitere de date în mod paralel

<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

Pentru a realiza asocierea între două sisteme de calcul de tip microcontroler, **Terminalul 1** va fi configurat prin intermediul codului program, ca și **transmițător**. Canalele digitale vor fi în acest sens **ieșiri digitale**. **Terminalul 2** va fi configurat prin intermediul codului program, ca și **receptor**. Canalele digitale, în acest sens vor fi configurate ca și **intrări digitale**. Informația se va transmite în mod paralel sau simultan ca și o **combinație de patru biți**. Numărul maxim de combinații posibile în acest sens, va fi $2^4 = 16$ combinații. Protocolul de date în acest sens, poate fi considerat **unidirecțional**.

Protocoloalele de comunicație cu transmitere de date în mod **serial**, pot vehicula cadrele de date pe **două canale de comunicație în mod succesiv, serializat**. Terminalele de date, pot avea rolul atât de transmițător cât și de receptor. Astfel, protocolele de comunicație serial, pot fi **bidirecționale**. În protocolele serial există **două canale** pentru vehicularea informației (Fig. 2), anume:

- „TxD” (eng. Transmit „x” = Any Data – canal pentru transmisia oricărui tip de date);
- „RxD” (eng. Receive „x” = Any Data – canal pentru recepționarea oricărui tip de date);

Canalul pentru transmisia oricărui tip de date „TxD” este reprezentat printr-o ieșire digitală, iar datele sunt transmise sub formă de impulsuri (bit cu bit). Canalul pentru recepționarea oricărui tip de date „RxD” este reprezentat printr-o intrare digitală, iar datele sunt recepționate în mod serializat (bit cu bit).

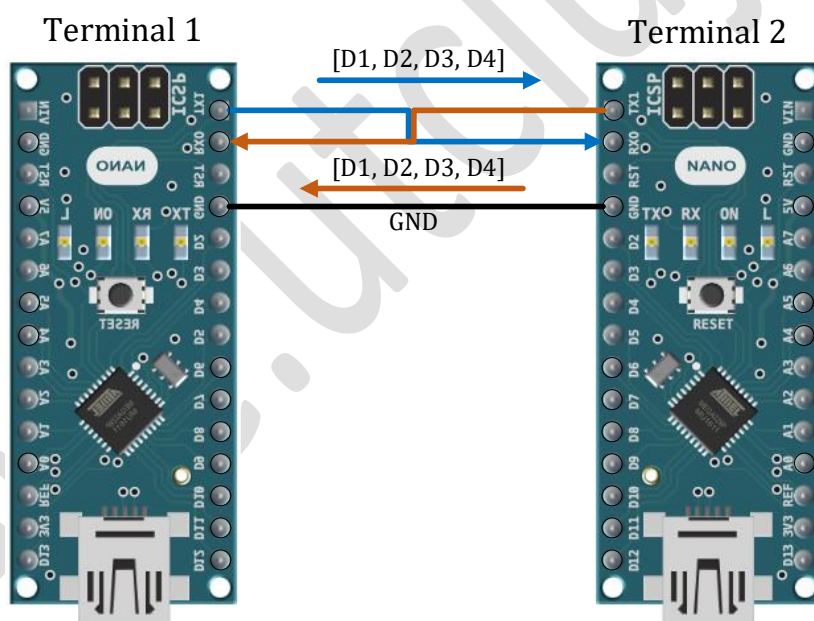


Fig. 2 – Protocolul de comunicație cu transmitere de date în mod serial

Protocoloalele de tip serial nu pot vehicula în mod simultan datele pe magistrala de comunicație, dar în schimb conferă siguranța suspendării procesului de comunicație în cazul întreruperii canalelor de comunicație. În cazul comunicației de tip paralel, la întreruperea unui canal digital de comunicație (întreruperea unui fir de date), informația se va transfera în mod eronat la terminalul receptor.

Comunicația de tip serial standard asincron (eng. Universal Asynchronous Receiver – Transmitter – UART), reprezintă modelul de bază al tuturor protocolelor de comunicație de tip serial (Fig. 3) și poate fi reprezentat la următoarele etape:

<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

A. Etapa de transmitere implică:

- conversia în binar a tipului de date care urmează a fi transmis;
- partiționarea informației binare rezultante în grupuri de opt biți (adică un byte);
- transmiterea informației în mod serializat (succesiv) a fiecărui bit rezultat;

B. Etapa de recepționare implică:

- recepționarea în mod serializat (succesiv) a fiecărui bit component al informației;
- concatenarea informației în grupuri a câte opt biți (adică un byte);
- mascarea biților care nu fac parte din cuvântul de date și conversia datelor în zecimal;

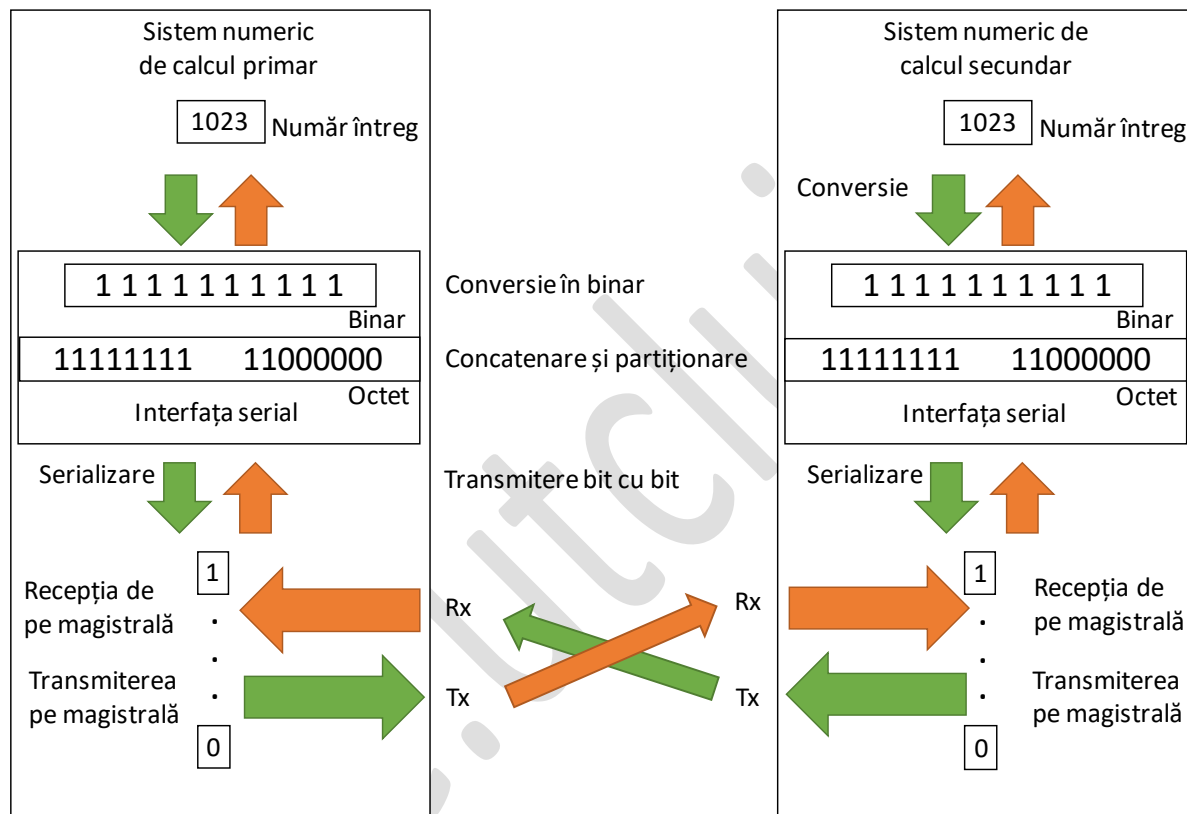


Fig. 3 – Modelul pentru vehicularea datelor în procesul de comunicație serial [3]

OBSERVAȚIE: În exemplul de mai sus (Fig. 3), este ilustrat faptul că, pe magistrala de date serial, pot fi vehiculate și alte tipuri de date având dimensiunea mai mare decât opt biți, dimensiunea maximă a unui pachet de date.

III. ASPECTE TEORETICE: [1], [2], [3], [4]

În domeniul sistemelor microprogramabile, cele mai des utilizate protocoale de tip serial sunt:

- UART – (eng. Universal Asynchronous Receiver – Transmitter) – Serial asincron;
- I²C – (eng. Inter – Integrated Circuit) – Serial sincron adresabil;
- SPI – (eng. Serial Peripheral Interface) – Serial sincron pentru dispozitive periferice;
- 1 - Wire – One Wire – Serial asincron;

<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

A. Protocolul Serial standard asincron UART:

Comunicația standard Serial asincronă (UART):

- reprezintă un protocol **fizic** (eng. **hardware**) de comunicație;
- se realizează pe **două fire**: „Tx” – eng. Transmit Data și „Rx” – eng. Receive Data (Fig. 4);
- face parte din clasa protocoalelor „**punct la punct**” (eng. Point To Point – PTP), adică, protocolul nu funcționează decât între **cel puțin și cel mult două terminale**;
- poate fi optimizat pentru a vehicula datele pe **distanțe medii** între două terminale, însă **volumul de date este redus**;
- poate funcționa în regim **bidirecțional**;

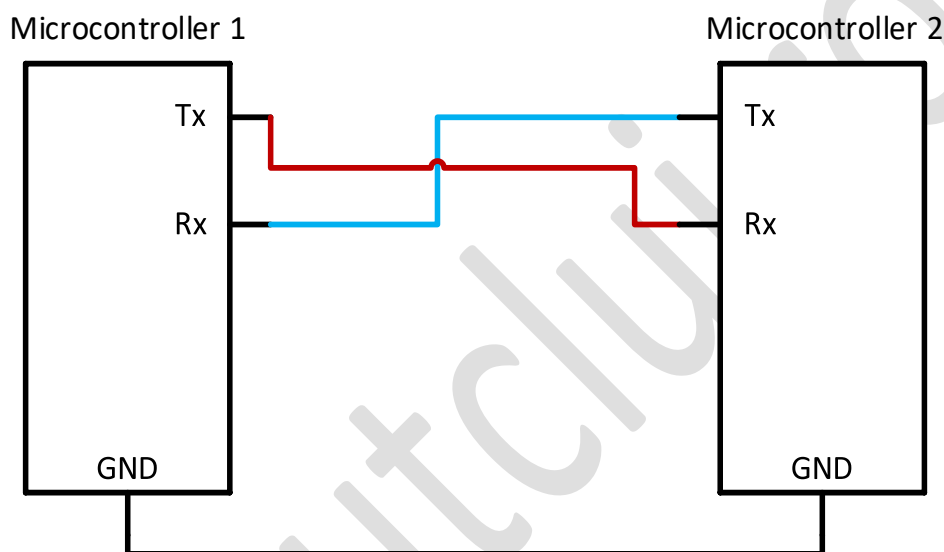


Fig. 4 – Conexiunea între terminale în cadrul protocolului serial standard asincron UART

Acest tip de protocol se utilizează pentru:

- transfer de date între două sisteme de calcul;
- diagnoză și depanare asistată software (ex. Consola Serial + Text Mode);
- trecerea de la un mediu de comunicare la altul (ex. Bluetooth – Serial HC-05);

B. Protocolul Serial sincron adresabil I²C:

Comunicația Serial sincronă cu terminale adresabile (I²C):

- reprezintă un protocol fizic (eng. hardware) de comunicație;
- se realizează pe două fire: „SCL” – eng. Serial CLock și „SDA” – eng. Serial DAta (Fig. 5);
- face parte din clasa protocoalelor **adresabile** de tip eng. „**Master – Slave**”, și poate deservi **mai mult decât două terminale**. Echipamentele sunt cuplate **în paralel** pe o **magistrală comună**, având **adrese distincte**;
- poate vehicula datele pe distanțe relativ reduse, în volum relativ mediu, la viteze cuprinse în intervalul 100 [kb / s] până la 3,5 [Mb / s];
- în vederea generării unei diferențe de potențial la nivelul canalelor de comunicație sunt necesare două rezistențe pentru conectare la potențialul sursei (eng. pull – up);

<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

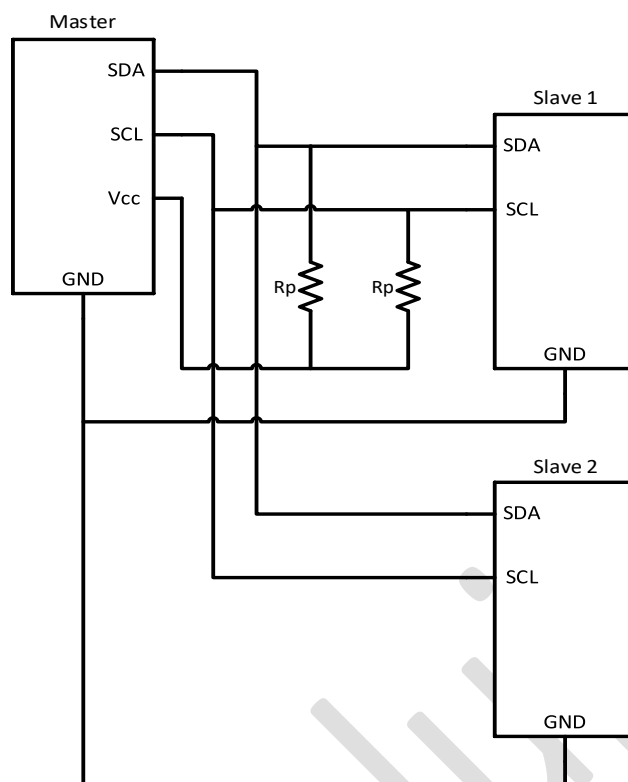


Fig. 5 - Conexiunea între terminale în cadrul protocolului serial sincron I²C

Acest tip de protocol se utilizează pentru:

- conectarea diverselor echipamente compacte la micro-controller (ex. LCD, senzori);
- inter-conectarea mai multor sisteme de calcul în vederea cooperării (Master / Slave);
- integrare de module și sisteme (ex. echipamente sau instalații multi-modulare) (Fig. 6);

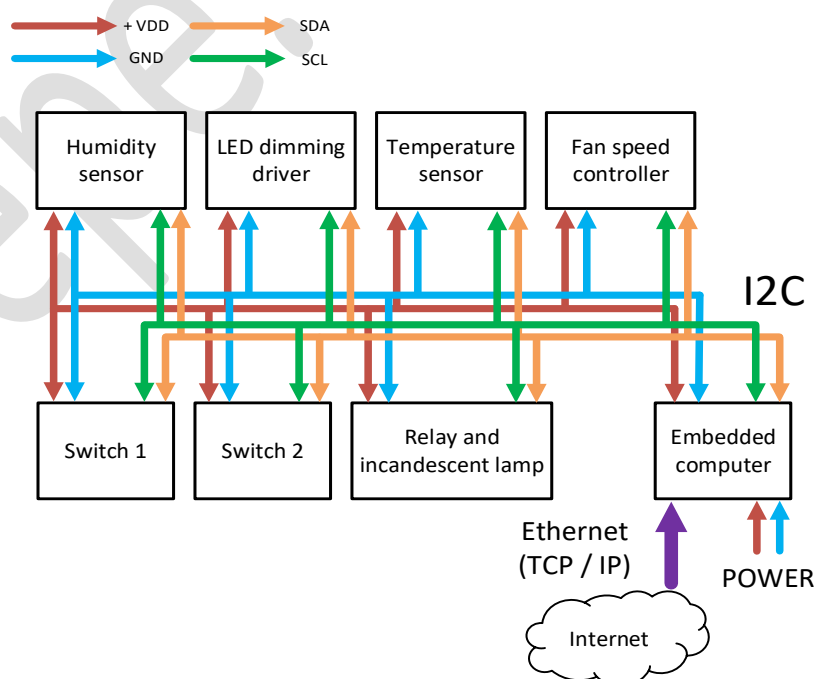


Fig. 6 – Automatizarea sistemelor pentru asigurarea confortului în domeniul rezidențial

<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

C. Protocolul Serial standard sincron SPI:

- Comunicația Serial sincronă cu terminale adresabile (SPI):
- reprezintă un protocol **fizic** (eng. hardware) de comunicație;
 - se realizează pe **patru canale** de transmiterea datelor (Fig. 7):
 - „SCLK” – eng. **S**erial **C**locK;
 - „MOSI” – eng. **M**aster **O**ut **S**lave **I**n;
 - „MISO” – eng. **M**aster **I**n **S**lave **O**ut;
 - „SS” – eng. **S**lave **S**elect;
 - face parte din clasa **protocoalelor adresabile hibride**, adică poate funcționa atât în modul de conectare **punct la punct** cât și în modul eng. „**Master – Slave**”, și poate deservi **mai mult decât două terminale**.
 - echipamentele sunt cuplate în **paralel** pe o **magistrală comună**;
 - adresarea în cazul protocolului SPI poate fi **opțională**;
 - este utilizat pentru a vehicula datele pe **distanțe relativ reduse**, în volum **relativ mare**, la viteze de maxim 60 [Mb / s];

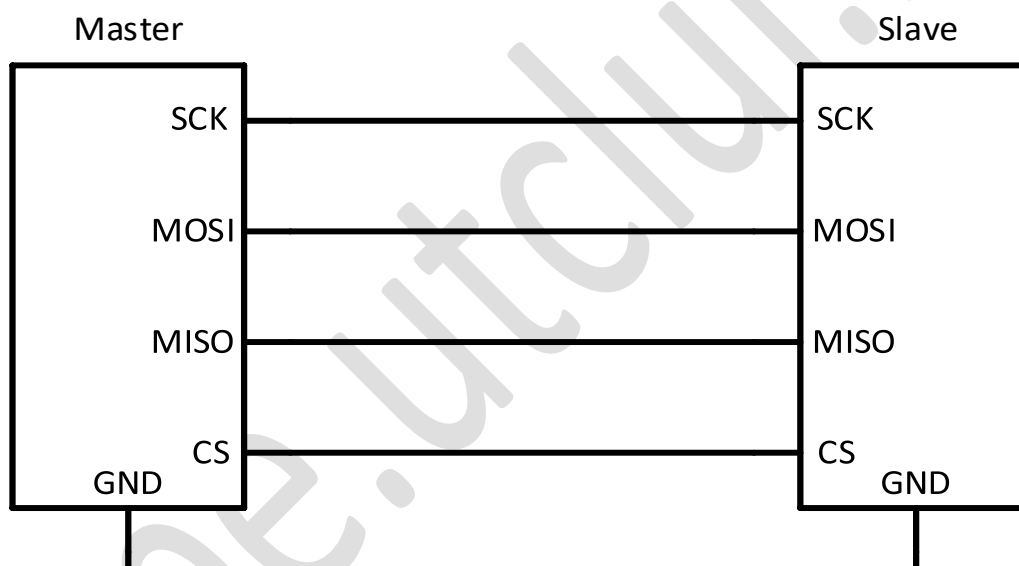


Fig. 7 - Conexiunea între terminale în cadrul protocolului serial sincron SPI

Acest tip de protocol se utilizează pentru:

- transfer de date între sisteme de calcul;
- programarea micro-controllerelor, și transferul de date direct în memorie;
- trecerea de la un mediu de comunicare la altul și conectarea altor echipamente de expansiune la sistemul de calcul dat (ex. eng. Ethernet – SPI Shield, SD card etc.);

D. Protocolul Serial asincron 1-Wire (One Wire):

- Comunicația Serial asincronă 1-Wire (One Wire):
- reprezintă un protocol fizic (eng. hardware) de comunicație;
 - se realizează prin intermediul unui singur canal de transmiterea datelor (Fig. 8);
 - funcționează în modul SMMS (eng. „Single Master and Multiple Slaves”);

<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

- funcționează în modul (eng. half - duplex), adică, un singur dispozitiv este capabil să inițieze procesul de comunicație și să transfere date în mod bidirecțional în același timp;
- prezintă viteză de transfer redusă (aprox. 15,4 [kb / s]);
- poate transfera un volum redus de date;

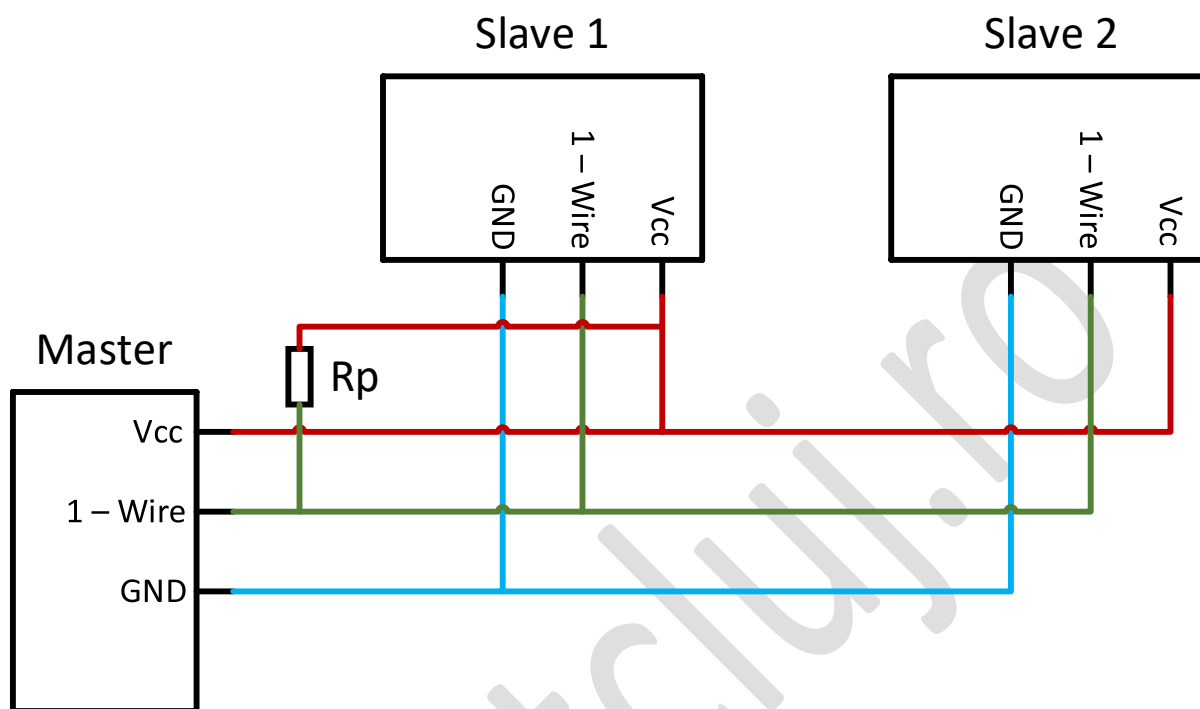


Fig. 8 - Conexiunea între terminale în cadrul protocolului serial asincron One Wire

Acest tip de protocol se utilizează pentru:

- transfer de date între sisteme de calcul;
- interfațarea traductoarelor cu sistemele care nu prezintă convertor analog - digital;
- monitorizarea parametrilor de funcționare pentru sursele de alimentare în comutație, destinate alimentării echipamentelor electronice portabile;

IV. IMPLEMENTAREA APLICAȚIILOR: [4], [5], [6], [7]

În vederea implementării aplicațiilor din cadrul acestei lucrări, vor fi utilizate două module specializate precum:

- modulul adaptor Bluetooth – Serial HC-05;
- traductorul digital DHT-11;

A. Modulul adaptor Bluetooth – Serial HC-05: [5]

Modulul HC-05 este un adaptor Bluetooth - Serial (eng. Serial Port Protocol - SPP), conceput pentru configurarea transparentă a conexiunii Serial fără fir. Este compatibil cu standardul „Bluetooth V2.0” și „EDR” (eng. Enhanced Data Rate). Tehnica de modulație se realizează la viteza de 3 [Mb / s] prin intermediul unui dispozitiv transmițător și receptor (eng. transceiver) radio cu bandă de frecvență 2,4 [GHz]. Acest modul înglobează de

<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

asemenea circuitul integrat specializat pentru aplicații în domeniul radio „Bluetooth CSR Bluecore 04 – External” cu construcție monobloc și implementare în tehnologie CMOS și tehnică de modulare AFH (eng. Adaptive Frequency Hopping Feature) (Fig. 9).

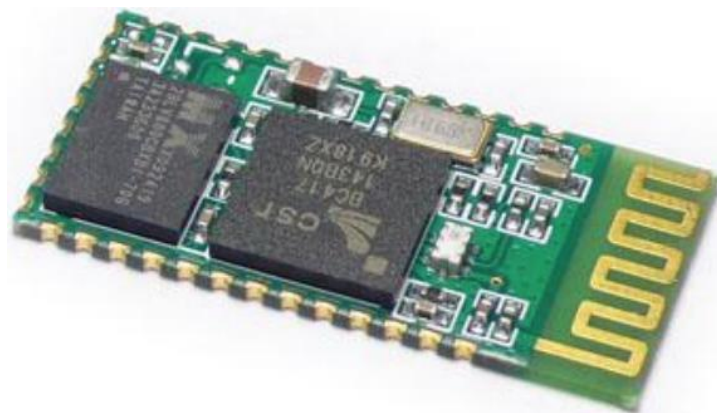


Fig. 9 – Modulul adaptor Bluetooth – Serial HC – 05 [5]

Modulul HC-05 prezintă următoarele dotări și specificații:

- nivel de sensibilitate -80 [dBm];
- putere de transmitere +4 [dBm];
- consum redus de putere, de la 1,8 [V] până la 3,6 [V] pe ieșirile digitale;
- compatibil cu protocolul serial asincron UART;
- antenă încorporată;

B. Traductorul digital DHT-11: [6]

Senzorul de temperatură și umiditate DHT11 furnizează mărimea rezultantă sub forma unui semnal digital. Acesta poate fi interpretat prin intermediul unei biblioteci de funcții (DHT.h) furnizată de către producător. Semnalul digital este generat prin intermediul unui microcontroler integrat în capsula traductorului. Protocolul de comunicație implementat la nivelul traductorului este serial de tip One Wire (Fig. 10).

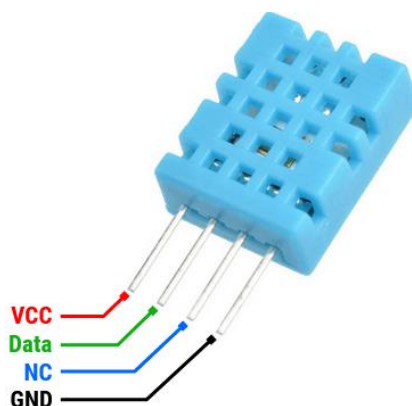


Fig. 10 – Traductorul de umiditate și temperatură DHT - 11 [6]

Pentru a evidenția conceptele prezentate anterior, se propune deci implementarea următoarelor aplicații cu ajutorul platformei Arduino Nano:

- recepționarea datelor prin interfața serial de la microcontroler;
- transmiterea datelor prin intermediul interfeței serial la microcontroler;
- recepționarea datelor prin intermediul modulului Bluetooth HC-05;
- transmiterea datelor prin intermediul modulului Bluetooth HC-05;
- preluarea datelor de la traductorul de umiditate și temperatură DHT-11;
- implementarea unei aplicații de monitorizare multipunct a parametrilor ambiantali

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;
- traductor pentru măsurarea temperaturii și a umidității ambientale DHT-11;
- rezistențe cu valoarea de 100 $[\Omega]$;
- rezistențe cu valoarea de 10 $[k\Omega]$;
- senzor de temperatura LM-35;
- 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. 11):

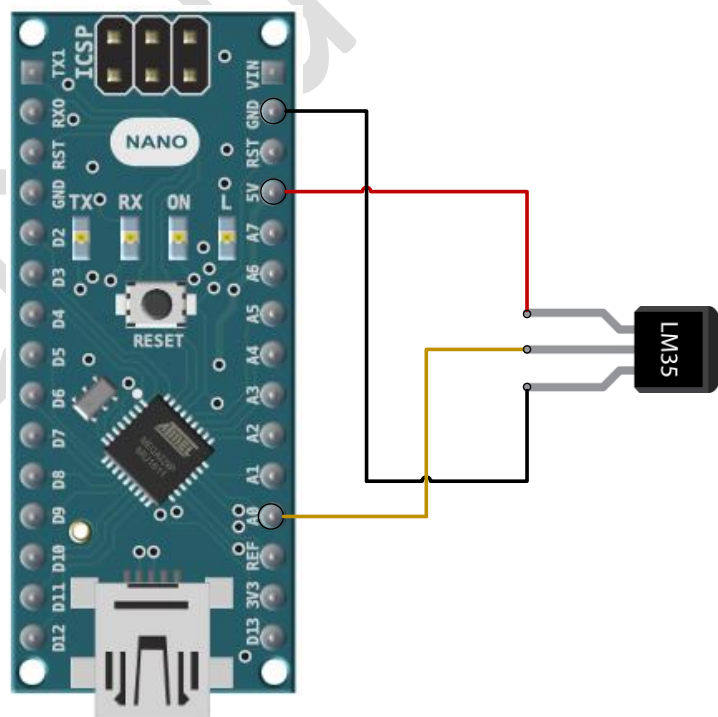


Fig. 11 – Schema electronică pentru implementarea circuitului specific aplicației 1

<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 realiza următorul montaj experimental (Fig. 12):

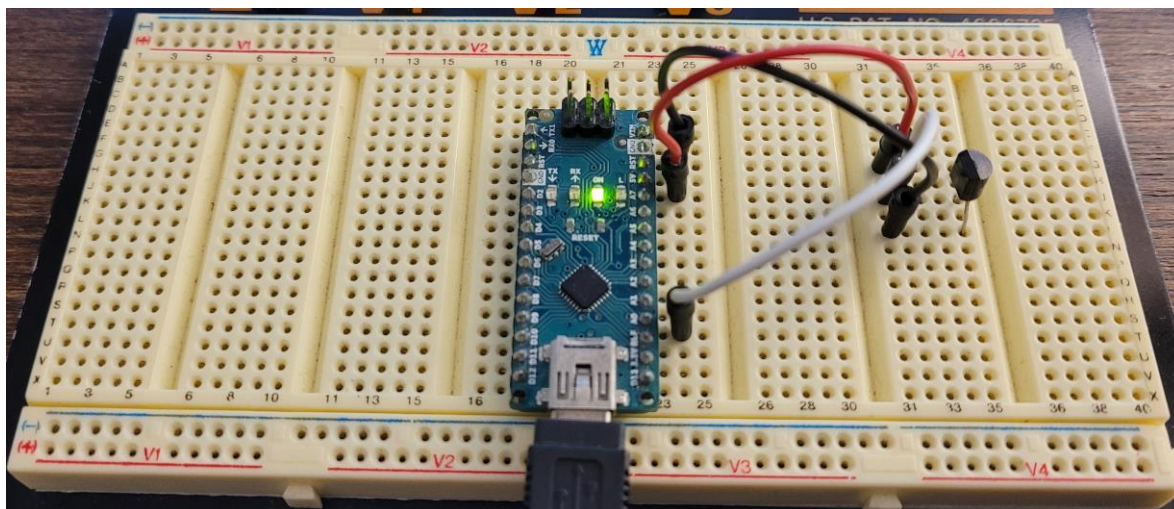


Fig. 12 – Montajul experimental specific aplicației 1

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.print("\n");
    delay(250);
}
```

<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 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 Serial (Fig. 13);

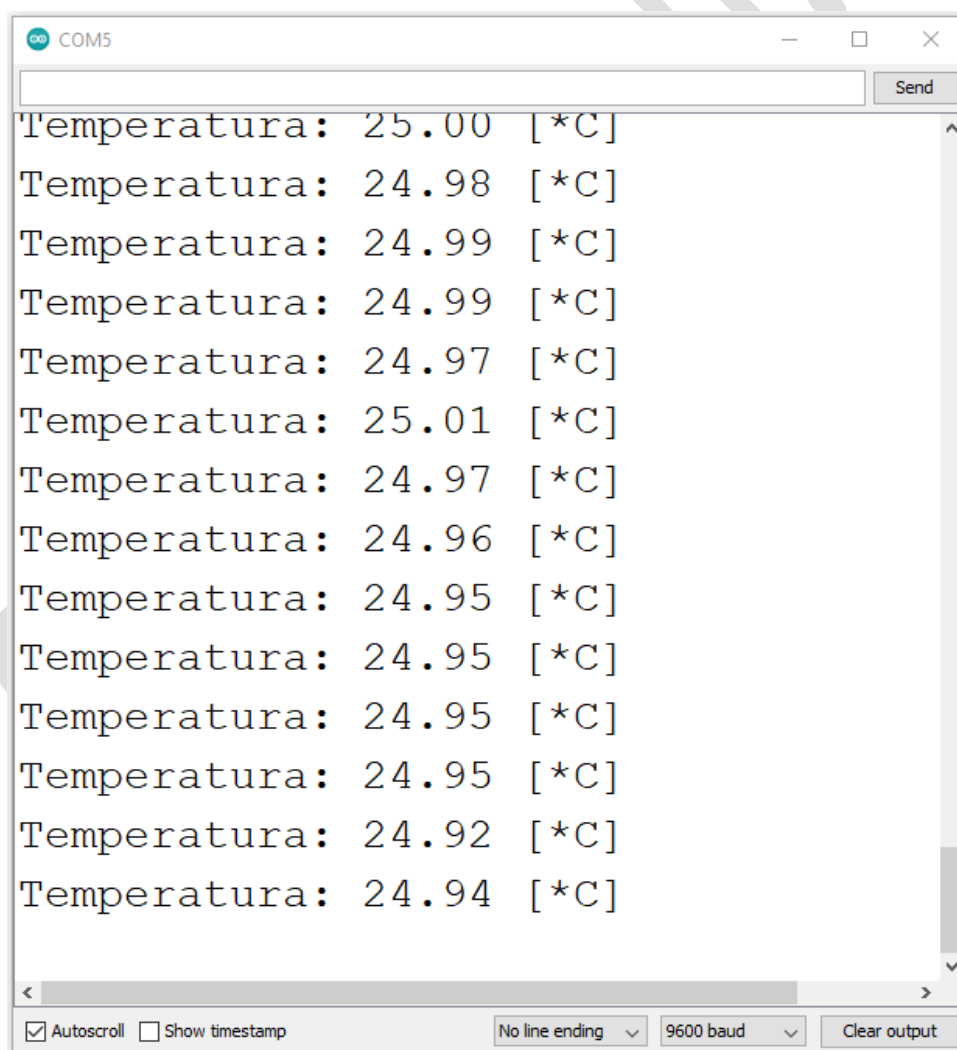


Fig. 13 – Afișarea mesajelor în consola serial

APLICAȚIA 2:

Se va implementa circuitul conform următoarei scheme (Fig. 14):

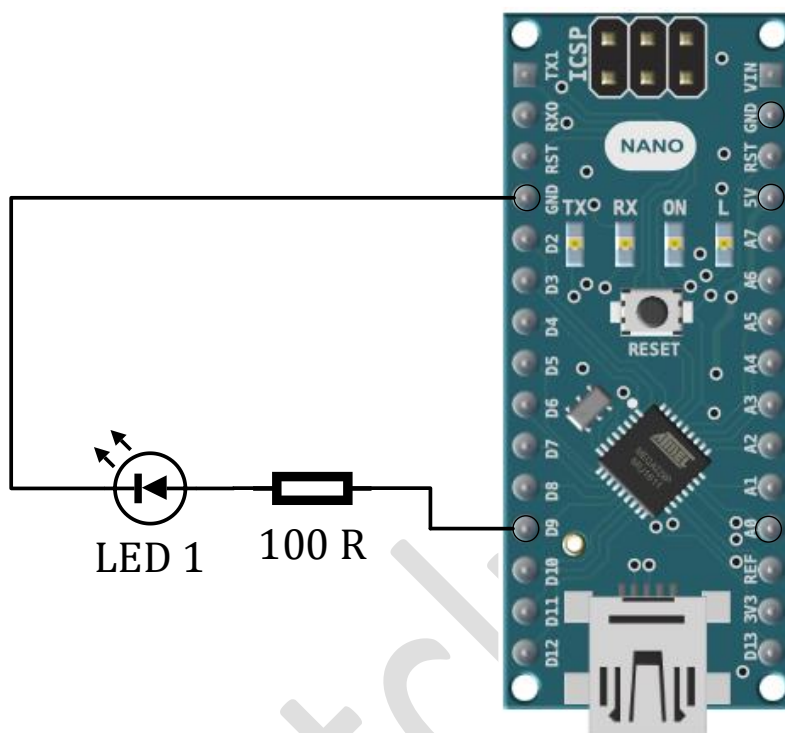


Fig. 14 – Schema electronică pentru implementarea circuitului specific aplicației 2

Se va realiza următorul montaj experimental (Fig. 15):

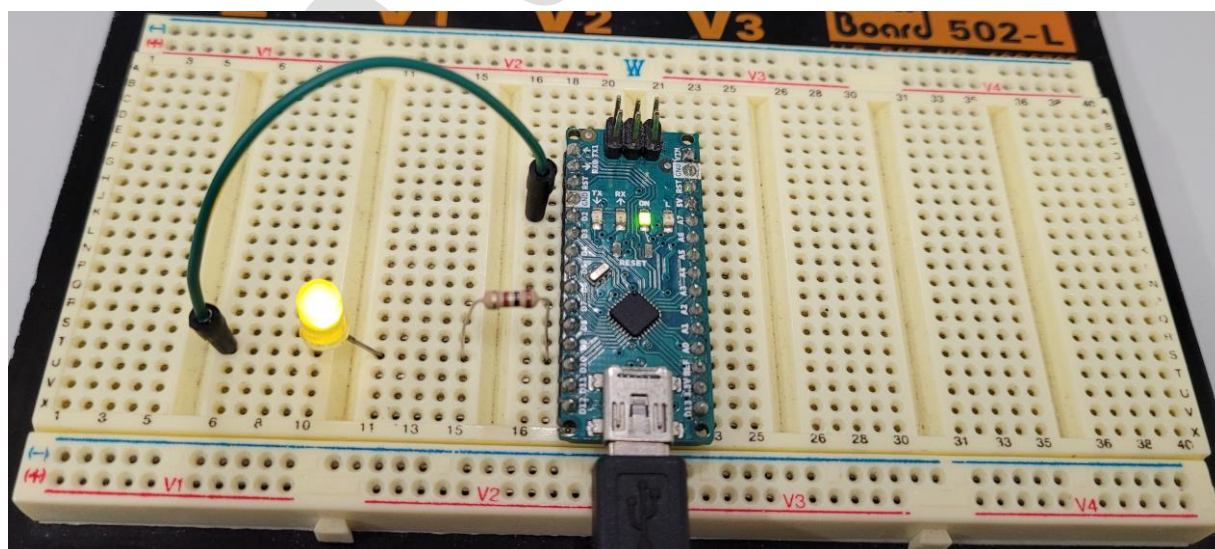


Fig. 15 – Montajul experimental specific aplicației 2

<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 pin_led = 9;
int dc = 0;

void setup() {
    pinMode(pin_led, OUTPUT);
    Serial.begin(9600);
    Serial.println("Introduceti factorul de umplere 0 - 255: ");
}

void loop() {
    if(Serial.available()){
        dc = Serial.parseInt();
        if(dc < 0){
            Serial.print("Valoarea introdusa nu este in interval!");
            Serial.print("\n");
            dc = 0;
        }
        if(dc > 255){
            Serial.print("Valoarea introdusa nu este in interval!");
            Serial.print("\n");
            dc = 0;
        }
        Serial.print("Valoarea factorului de umplere este: ");
        Serial.print(dc);
        Serial.print("\n");
    }
    analogWrite(pin_led, dc);
}
```

Implementarea aplicației nr. 2 presupune:

- declararea unei constante de tip număr întregi „pin_led” cu valoarea „9”;
- inițializarea unei variabile de tip număr întreg „dc” cu valoarea „0”;
- inițializarea terminalului digital „9” în modul de lucru „ieșire digitală”;
- inițializarea comunicației Serial la viteza de transfer 9600 [b/s]
- afișarea mesajului: "Introduceti factorul de umplere 0 - 255: ";
- verificarea disponibilității datelor de pe magistrala serial cu funcția „Serial.available()”;
- preluarea factorului de umplere de la magistrala serial cu funcția „Serial.parseInt()”;
- afișarea valorii exprimată pe 8 biți a duratei de conducție în consola serial (Fig. 16);
- ajustarea factorului de umplere pentru un tren de impulsuri furnizat pe terminalul „9”;

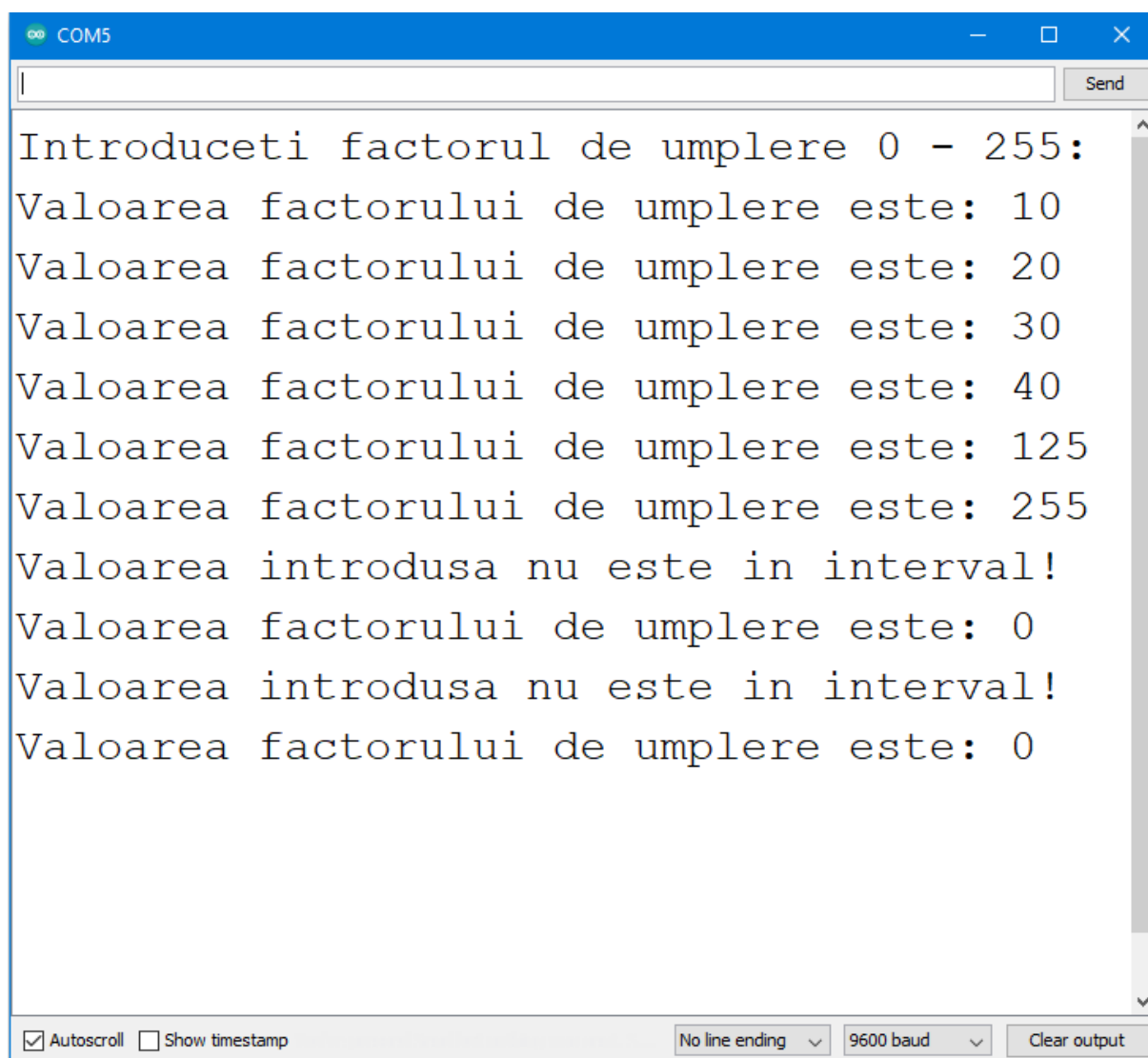


Fig. 16 – Afișarea factorului de umplere sau a duratei de conducție în consola Serial

APLICAȚIA 3:

Se va implementa următorul cod program:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(4, 5); //4 - RX, 5 - TX

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);
    BTSerial.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.print("\n");

    BTSerial.print("Temperatura: ");
    BTSerial.print(media);
    BTSerial.print(" [*C]");
    BTSerial.print("\n");
    delay(250);
}
```

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

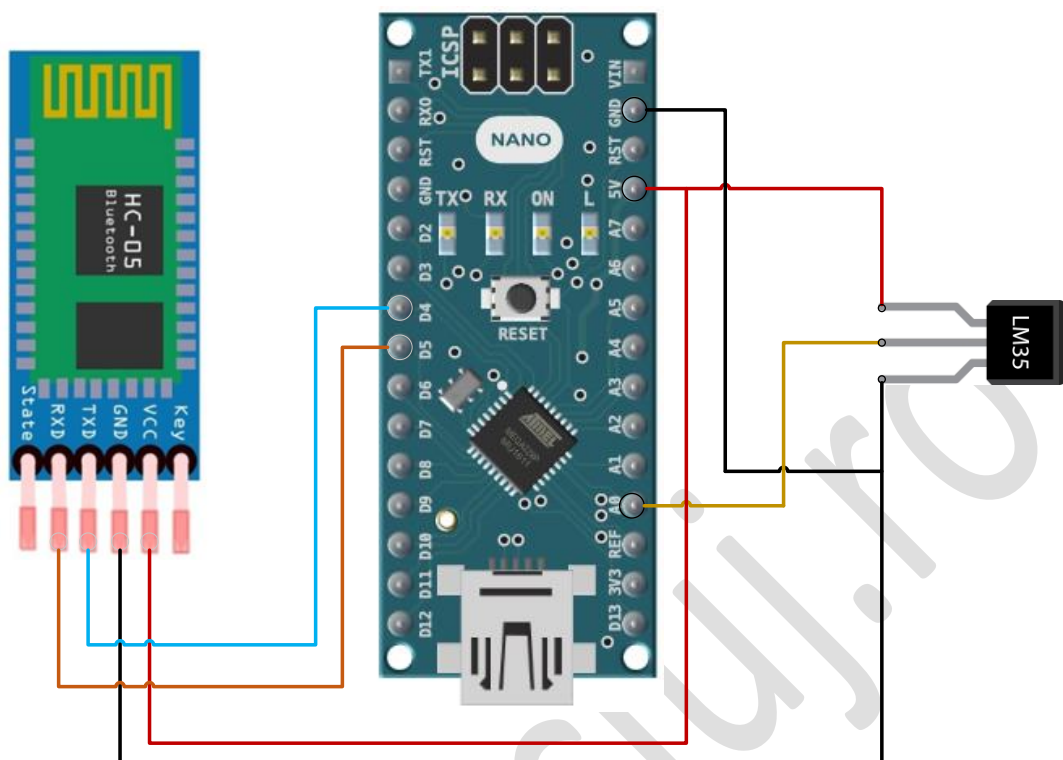


Fig. 17 - Schema electronică pentru implementarea circuitului specific aplicației 3

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

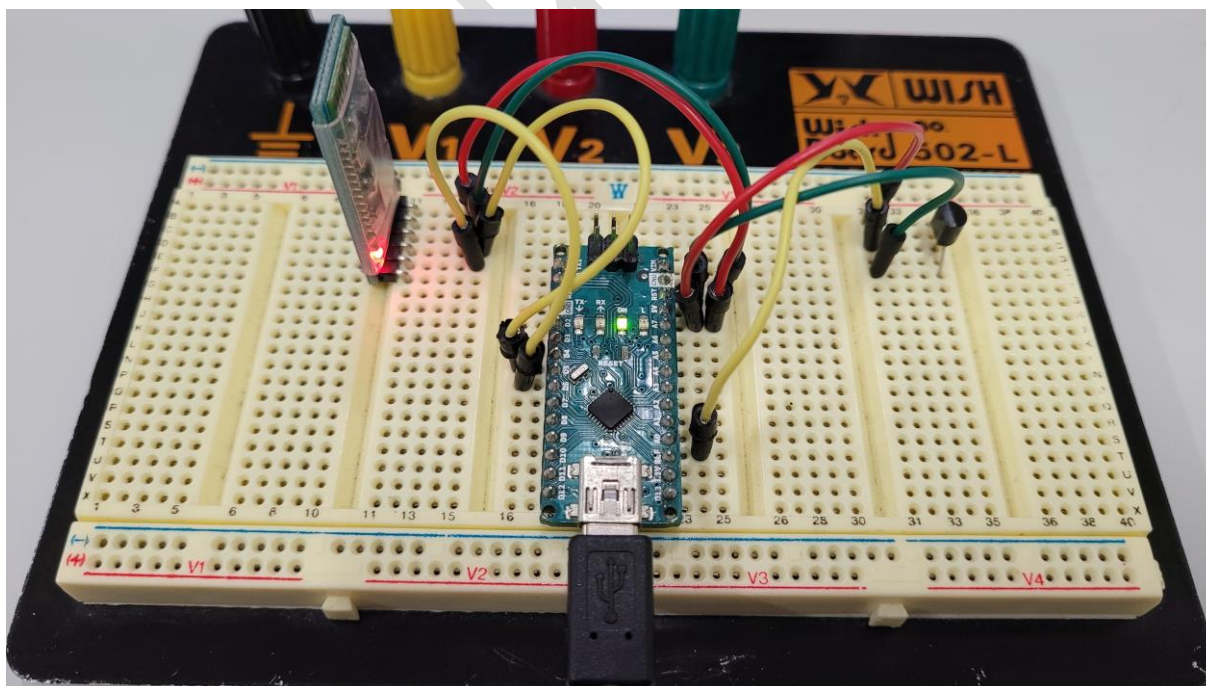


Fig. 18 - Montajul experimental specific aplicației 3

<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

OBSERVAȚII:

1. Prin intermediul bibliotecii de funcții „SoftwareSerial.h” se va inițializa comunicația serial în mod virtual prin intermediul a două terminale digitale la alegere (în cazul de față „D4” – RxD iar „D5” – TxD).
2. Pentru a verifica funcționalitatea modului Bluetooth – Serial HC-05, se va utiliza aplicația „Bluetooth Serial Terminal” pe telefonul mobil. Prin intermediul acestei aplicații va fi posibilă recepționarea mesajelor din consola serial pe telefonul mobil (Fig. 19).

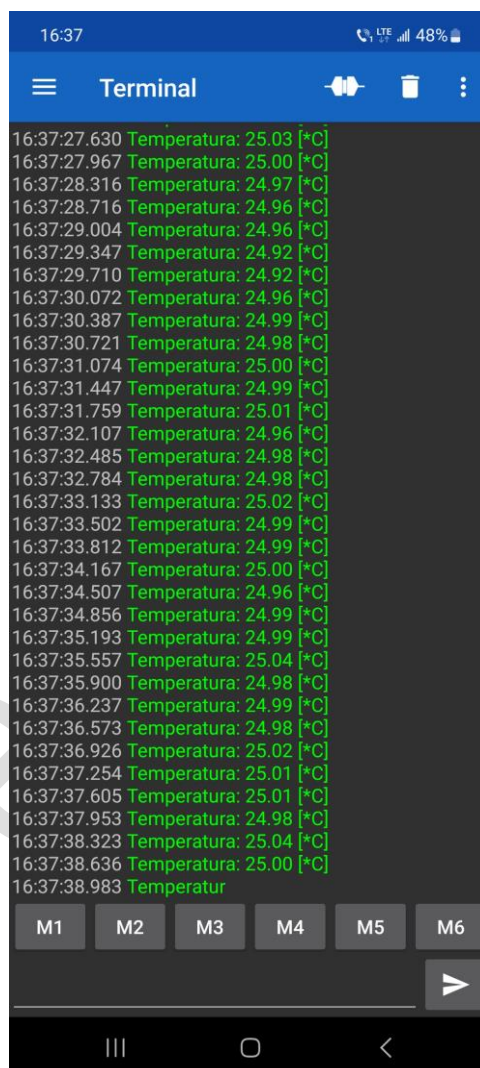


Fig. 19 – Aplicația Bluetooth Serial Terminal – vizualizarea conținutului consolei serial

Implementarea aplicației nr. 3 presupune:

- determinarea temperaturii medii achiziționată de la traductorul LM-35;
- inițializarea comunicației serial atât locală cât și la distanță;
- afișarea locală a mesajului pentru monitorizare în consola serial;
- afișarea în aplicația Bluetooth Serial Terminal a conținutului consolei serial.

<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):

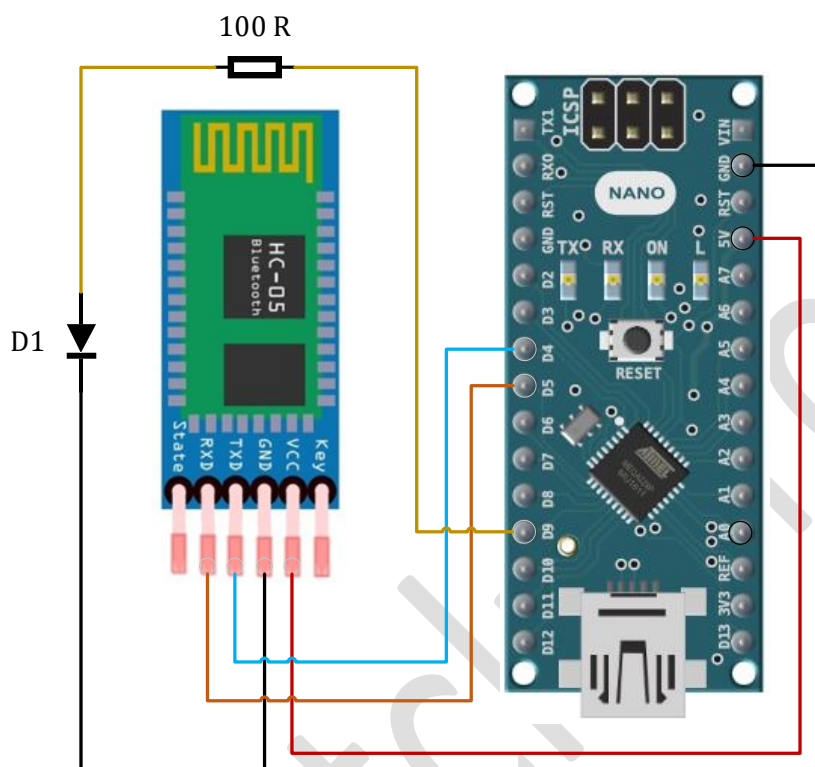


Fig. 20 - Schema electronică pentru implementarea circuitului specific aplicației 4

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

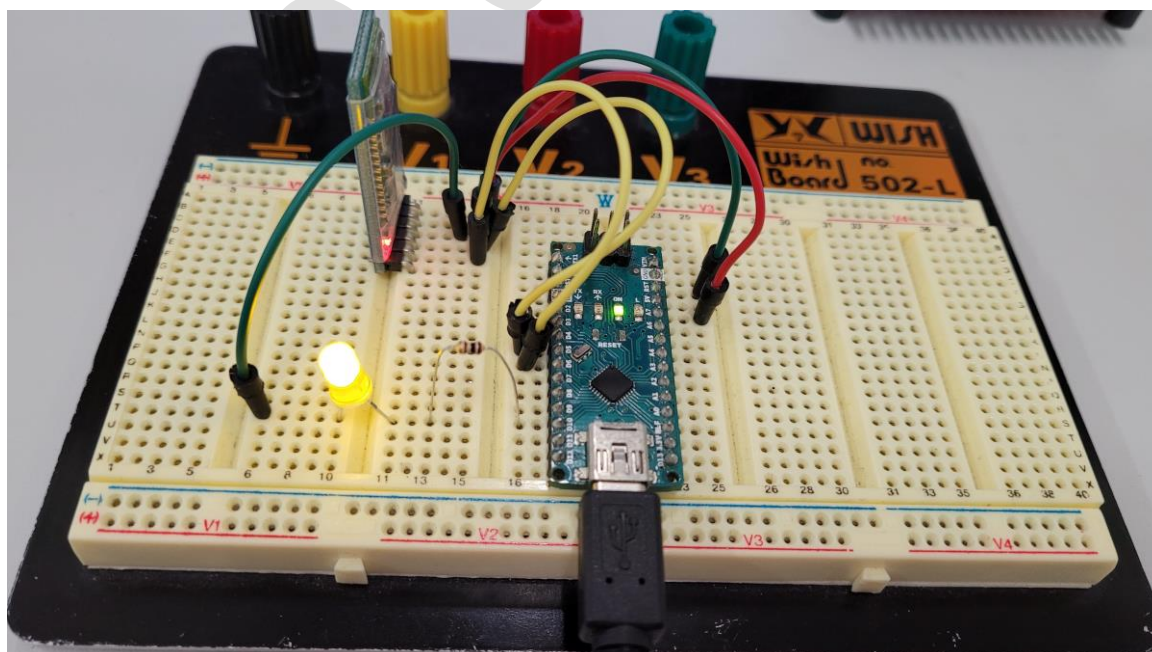


Fig. 21 - Montajul experimental specific aplicației 4

<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:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(4, 5); //4 - RX, 5 - TX

const int pin_led = 9;
int dc = 0;

void setup() {
    Serial.begin(9600);
    BTSerial.begin(9600);
    Serial.println("Introduceti factorul de umplere: ");
    BTSerial.println("Introduceti factorul de umplere: ");
}

void loop() {
    if(Serial.available()){
        dc = Serial.parseInt();
        if(dc < 0){
            Serial.print("Valoarea introdusa nu este in interval!");
            Serial.print("\n");
            dc = 0;
        }
        if(dc > 255){
            Serial.print("Valoarea introdusa nu este in interval!");
            Serial.print("\n");
            dc = 0;
        }
        Serial.print("Valoare factor de umplere: ");
        Serial.print(dc);
        Serial.print("\n");
    }

    if(BTSerial.available()){
        dc = BTSerial.parseInt();
        if(dc < 0){
            BTSerial.print("Valoarea introdusa nu este in interval!");
            BTSerial.print("\n");
            dc = 0;
        }
        if(dc > 255){
            BTSerial.print("Valoarea introdusa nu este in interval!");
            BTSerial.print("\n");
            dc = 0;
        }
    }
}
```

<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

```

    BTSerial.print("Valoare factor de umplere: ");
    BTSerial.print(dc);
    BTSerial.print("\n");
}
analogWrite(pin_led, dc);
}

```

Implementarea aplicației nr. 4 presupune:

- inițializarea comunicației Serial la viteza de transfer 9600 [b/s]
- inițializarea comunicației serial atât locală cât și la distanță;
- verificarea disponibilității datelor de pe magistrala serial cu funcția „Serial.available()”;
- verificarea disponibilității datelor de pe magistrala cu funcția „BTSerial.available()”;
- preluarea factorului de umplere de la magistrala serial cu funcția „Serial.parseInt()”;
- preluarea factorului de umplere de la magistrala serial cu funcția „BTSerial.parseInt()”;
- afișarea valorii exprimată pe 8 biți a duratei de conducție în consola serial;
- afișarea valorii pe 8 biți a duratei de conducție în aplicația Bluetooth Serial Terminal;
- ajustarea factorului de umplere pentru un tren de impulsuri furnizat pe terminalul „9”;
- preluarea atât local cât și la distanță a datelor numerice din consolă (Fig. 22);
- afișarea locală a mesajului pentru monitorizare în consola serial;
- afișarea în aplicația Bluetooth Serial Terminal a conținutului consolei serial.

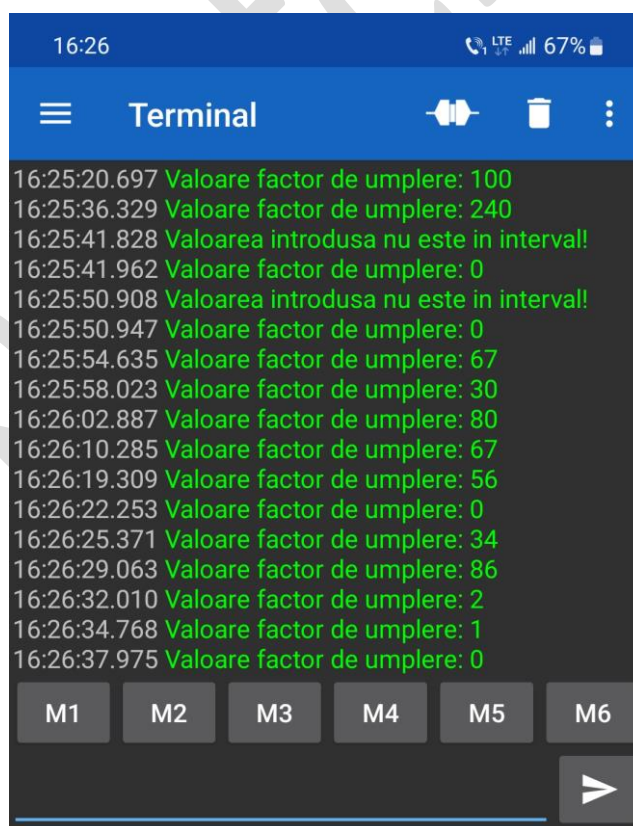


Fig. 22 – Aplicația Bluetooth Serial Terminal – preluarea valorii factorului de umplere și afișarea în consolă a mesajelor

<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 5:

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

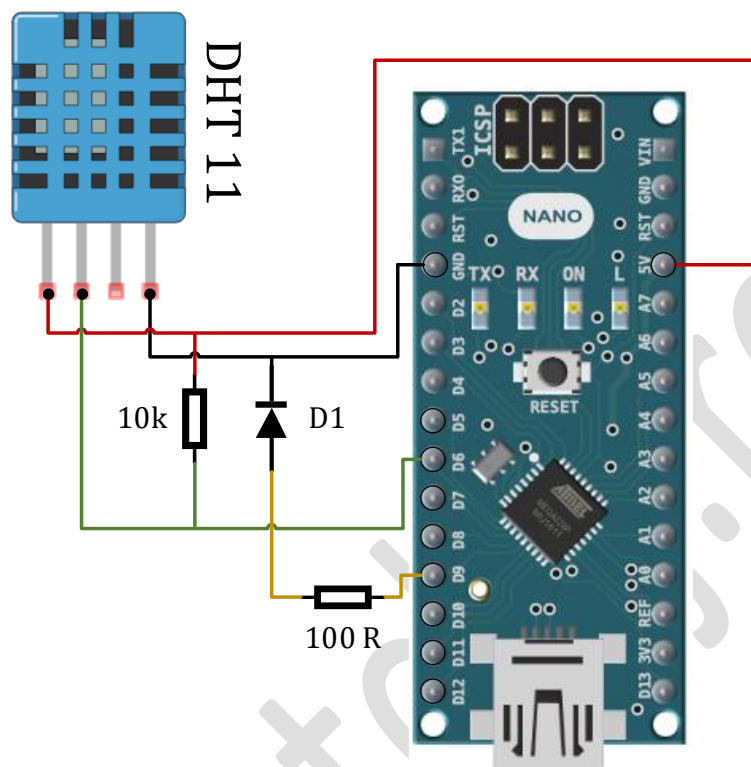


Fig. 23 - Schema electronică pentru implementarea circuitului specific aplicației 5

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

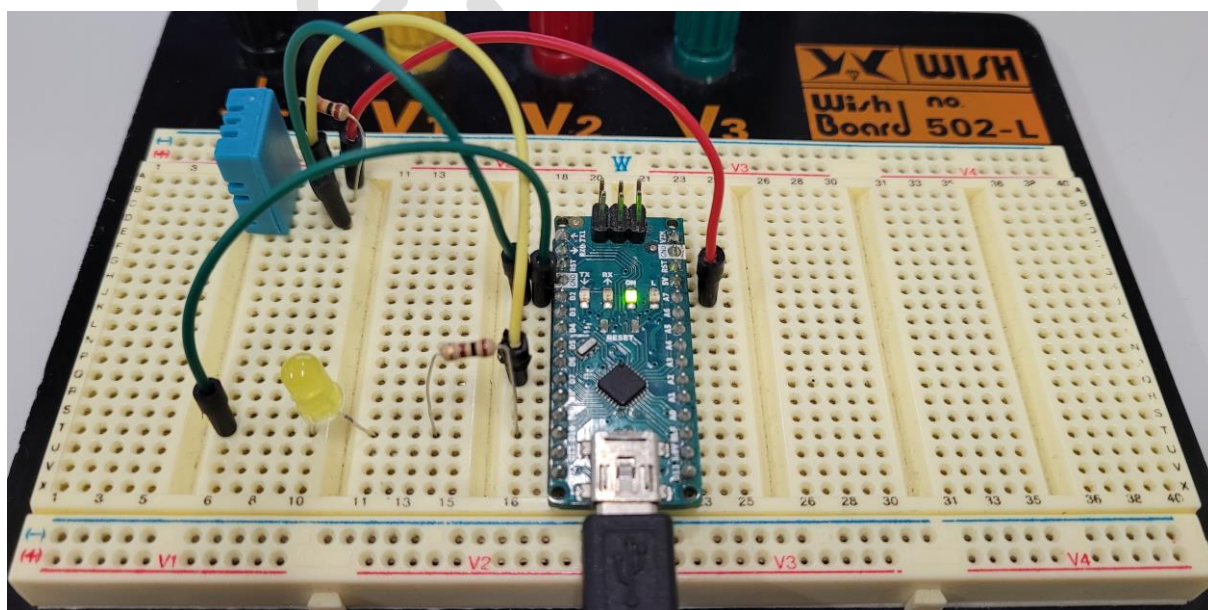


Fig. 24 - Montajul experimental specific aplicației 5

<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:

```
#include <dht.h>
#define DHT11_PIN 6

const int status_led = 9;
dht DHT;

void setup() {
    pinMode(status_led, OUTPUT);
    Serial.begin(9600);
    digitalWrite(status_led, LOW);
}

void loop() {
    int chk = DHT.read11(DHT11_PIN);
    digitalWrite(status_led, LOW);
    delay(5000);
    Serial.print("Temperatura: ");
    Serial.println(DHT.temperature);
    Serial.print("Umiditate: ");
    Serial.println(DHT.humidity);
    digitalWrite(status_led, HIGH);
    delay(100);
}
```

Implementarea aplicației nr. 5 presupune:

- inițializarea bibliotecii „dht.h” specifică traductorului de umiditate și temperatură;
- inițializarea obiectului „DHT” pentru interogarea punctuală a parametrilor descriptivi;
- interogarea traductorului cu ajutorul bibliotecii de funcții;
- preluarea de pe magistrala „1 – Wire” a informațiilor digitale de la traductor;
- afișarea în consola serial a valorilor de temperatură și umiditate (Fig. 25);
- semnalizarea prin intermediul unui LED de stare la fiecare achiziție;

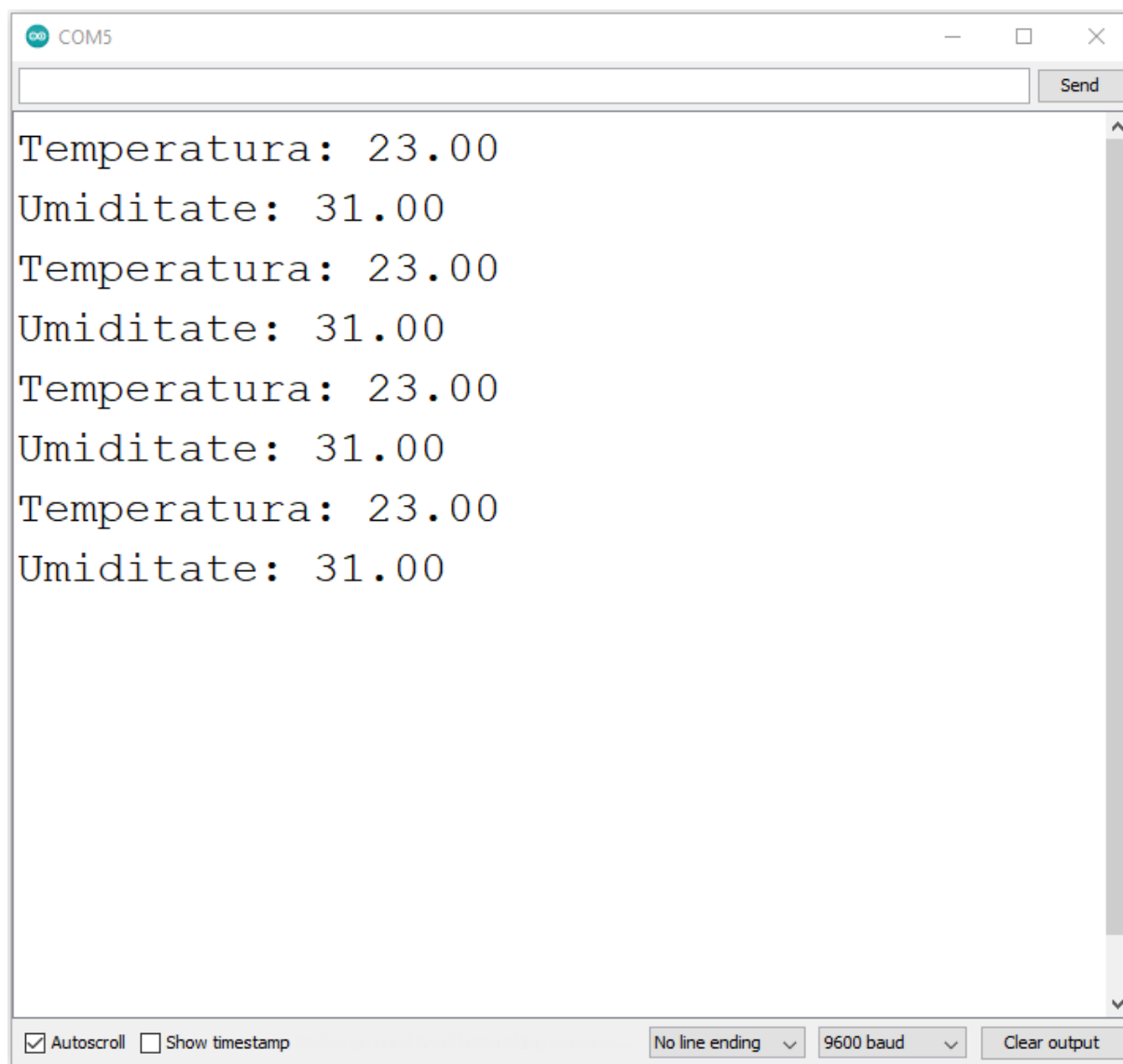


Fig. 25 – Consola serial – preluarea datelor de la traductorul digital de umiditate și temperatură DHT 11 și afișarea lor în consolă

APLICAȚIA 6:

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

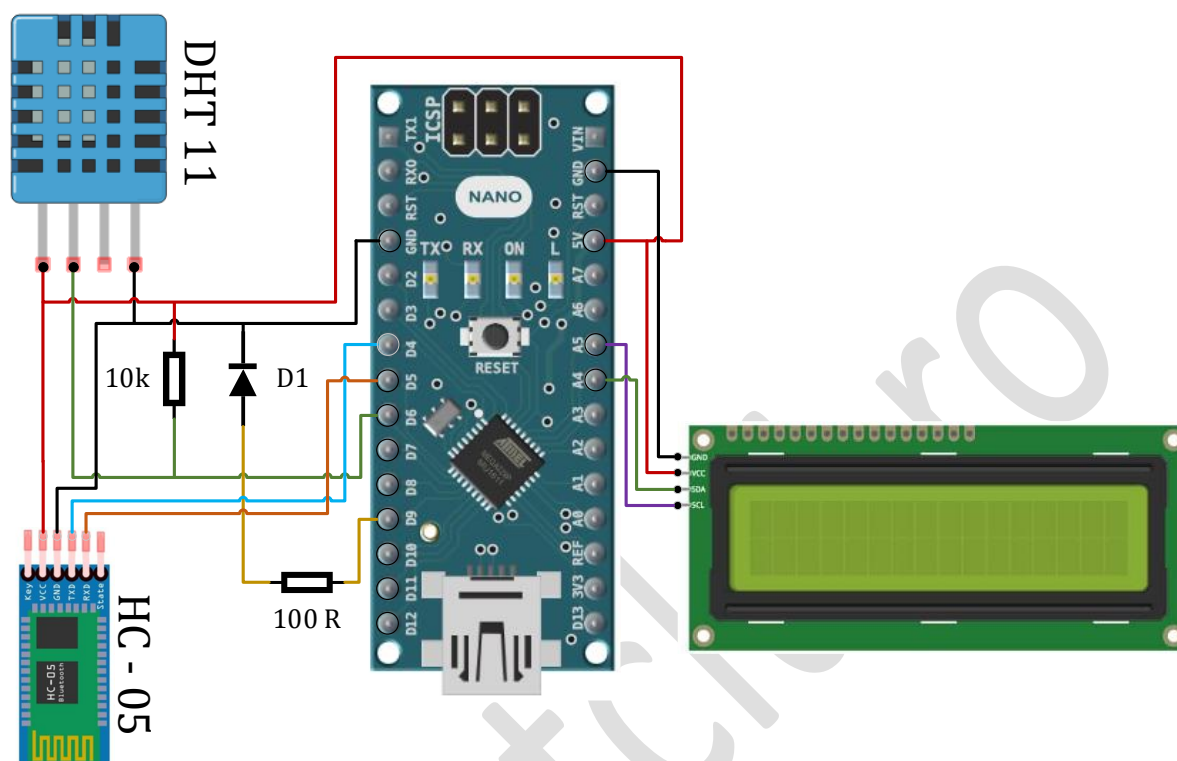


Fig. 26 - Schema electronică pentru implementarea circuitului specific aplicației 6

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

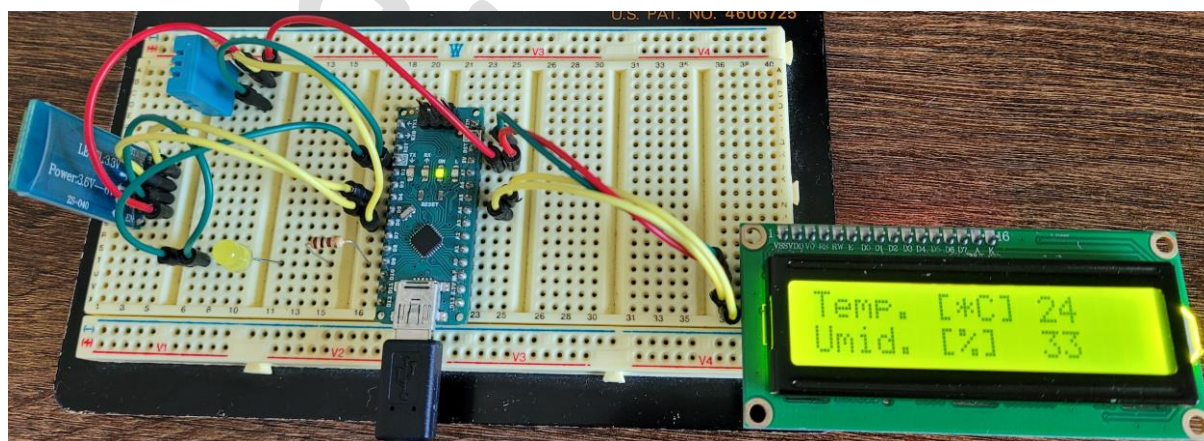


Fig. 27 - Montajul experimental specific aplicației 6

<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:

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <dht.h>
#define DHT11_PIN 6
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
SoftwareSerial BTSerial(4, 5); //4 - RX, 5 - TX
const int status_led = 9;
dht DHT;

void setup() {
  pinMode(status_led, OUTPUT);
  Serial.begin(9600);
  BTSerial.begin(9600);
  digitalWrite(status_led, LOW);
  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("Temp. [*C] ");
  lcd.setCursor(0,1);
  lcd.print("Umid. [%] ");
}

void loop() {
  int chk = DHT.read11(DHT11_PIN);
  digitalWrite(status_led, LOW);
  delay(2000);
  Serial.print("Temperatura [*C]: ");
  Serial.println(DHT.temperature, 0);
  Serial.print("Umiditate [%]: ");
  Serial.println(DHT.humidity, 0);
  BTSerial.print("Temperatura [*C]: ");
  BTSerial.println(DHT.temperature, 0);
  BTSerial.print("Umiditate [%]: ");
  BTSerial.println(DHT.humidity, 0);
  digitalWrite(status_led, HIGH);
```



```
lcd.setCursor(11,0);

if(DHT.temperature < 1000) {
  lcd.setCursor(14,0);
  lcd.print(" ");
  lcd.setCursor(11,0);
  lcd.print(DHT.temperature, 0);
}
if(DHT.temperature < 100) {
  lcd.setCursor(13,0);
  lcd.print(" ");
  lcd.setCursor(11,0);
  lcd.print(DHT.temperature, 0);
}
if(DHT.temperature < 10) {
  lcd.setCursor(12,0);
  lcd.print(" ");
  lcd.setCursor(11,0);
  lcd.print(DHT.temperature, 0);
}

lcd.setCursor(11,1);

if(DHT.humidity < 1000) {
  lcd.setCursor(14,1);
  lcd.print(" ");
  lcd.setCursor(11,1);
  lcd.print(DHT.humidity, 0);
}
if(DHT.humidity < 100) {
  lcd.setCursor(13,1);
  lcd.print(" ");
  lcd.setCursor(11,1);
  lcd.print(DHT.humidity, 0);
}
if(DHT.humidity < 10) {
  lcd.setCursor(12,1);
  lcd.print(" ");
  lcd.setCursor(11,1);
  lcd.print(DHT.humidity, 0);
}
delay(100);
}
```


Implementarea aplicației nr. 6 presupune:

- interogarea periodică a traductorului digital de umiditate și temperatură DHT – 11;
- afișarea locală (în consola serial a calculatorului gazdă) a valorilor preluate;
- afișarea la distanță a valorilor înregistrate de la traductor (Bluetooth Serial Terminal);
- expunerea valorilor preluate de la traductor pe afișajul LCD;

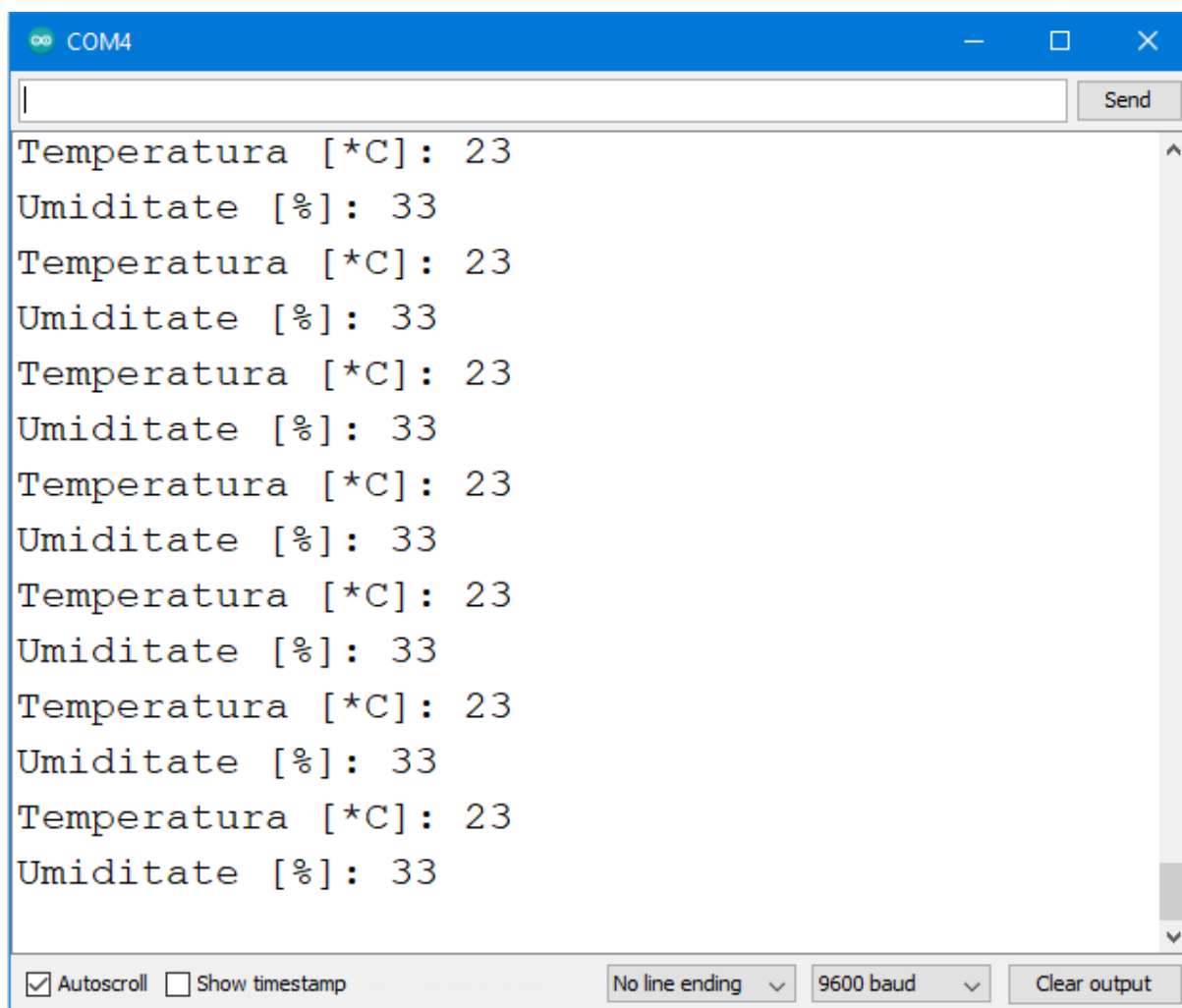


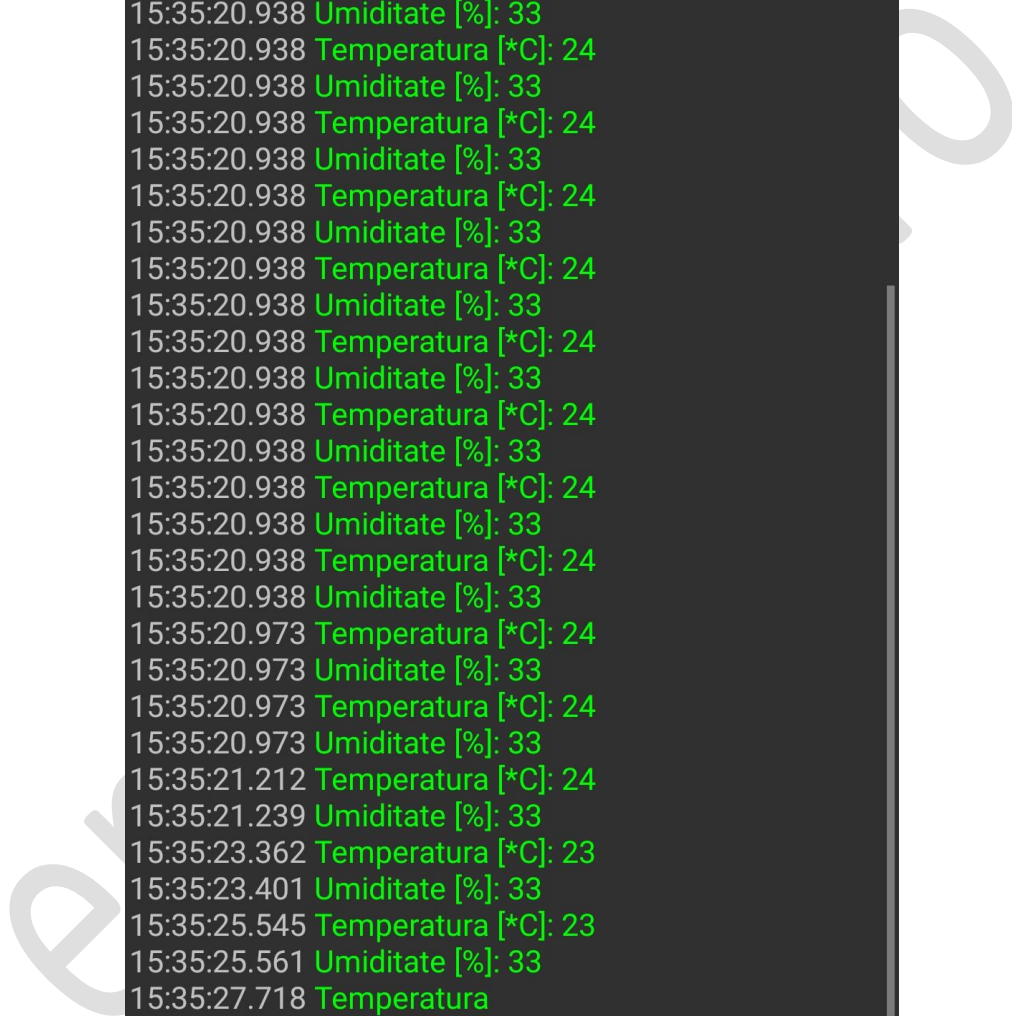
Fig. 28 – Consola serial – preluarea datelor de la traductorul digital de umiditate și temperatură DHT 11 și afișarea lor în consolă

OBSERVAȚIE: Această aplicație poate fi considerată o modalitate de tip „multipunct” de supraveghere și monitorizare a parametrilor ambientali, precum temperatura și umiditatea aerului din încăpere, deoarece, informația finală poate fi preluată atât de la fața locului (prin intermediul afișajului LCD – fizic, eng. hardware) cât și de la consola serial a calculatorului sau a telefonului mobil (logic sau virtual - eng. software). Funcționalitatea „multipunct” a aplicației poate deservi de asemenea procesul și metodele de depanare și diagnosticare a unui echipament complex (eng. debugging methods).

<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



<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

V. CONCLUZIE:

Prin intermediul protocoalelor de comunicație implementate atât la nivelul codului program (eng. software protocol) cât și la nivelul fizic (eng. hardware protocol) al unui sistem de calcul, pot fi realizate următoarele funcții:

- diagnoză și depanare (eng. debugging and repairing);
- monitorizarea la distanță a unor parametrii (eng. remote monitoring);
- accesul multipunct la mijloacele fizice ale aplicației (eng. multi point operation);
- traversarea sau conversia de la un protocol la altul (ex. adaptorul Bluetooth – Serial);
- transportarea datelor pe distanțe mari în format digital codificat;

VI. BIBLIOGRAFIE:

1. Technical University of Wien – „Communication Protocols for Embedded Systems” - <https://ti.tuwien.ac.at/>
2. Universitatea Politehnică Timișoara – „Interfațarea circuitelor - Tipuri de interfețe, Protocoale seriale de comunicare” - <https://www.aut.upt.ro>
3. Universitatea Tehnică din Cluj – Napoca: 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. Washington University – „MEMS: Micro-Electro-Mechanical Systems – What are MEMS?” - <https://courses.cs.washington.edu/>
5. ITead Studio – „HC-05 -Bluetooth to Serial Module” - <https://components101.com/>
6. Mouser Electronics and OSEPP Electronics – „DHT11 Humidity & Temperature Sensor” <https://www.mouser.com/>
7. Electronics-lab.com – „BUILD YOUR OWN I2C SENSOR” – ATTiny 85 MCU <https://www.electronics-lab.com/build-i2c-sensor/>