

Telemetru ultrasonic cu microcontroler

1. Introducere

Detectarea non-contact a nivelului (și aflarea volumului) lichidelor toxice, puternic acide sau puternic alcaline din rezervoare, în condiții de maximă siguranță, este un deziderat major care se poate implementa cu ajutorul ecolocației.

În general, prin ecolocație se înțelege localizarea obiectelor din mediul înconjurător, cu ajutorul ultrasunetelor (unde sonore cu frecvența mai mare de 20 KHz). Fenomenul ecolocație este utilizat atât în natură (exemplu: delfini, lilieci, unele insecte, etc.) cât și în tehnică (exemplu: sonar, robotică, telementre ultrasonore, etc).

Măsurarea distanțelor cu ultrasunete se realizează cu ajutorul relației:

$$D = \frac{c \cdot (t_2 - t_1)}{2} \quad (1)$$

unde D = distanța, c = viteza sunetului, $t_2 - t_1$ = intervalul de timp în care sunetul parcurge distanța de la telemetrul ultrasonic până la obiect și înapoi (drumul dus-întors).

Pentru a măsura distanța, telemetrul cu ultrasunete emite un impuls ultrasonor cu o durată mică, apoi recepționează semnalul ecou reflectat de obiectul până la care se dorește măsurarea distanței. Dacă se măsoară intervalul de timp dintre momentul emiterii semnalului ultrasonor (t_1) și cel al recepției semnalului ecou (t_2) și dacă se cunoaște viteza de propagare a sunetului în mediul respectiv, atunci se poate determina cu ajutorul relației (1) distanța până la obiect. Viteza sunetului în aer este constantă la o temperatură constantă. De exemplu la 0 °C viteza sunetului este de 331,5 m/s, iar la 20 °C aproximativ 343,6 m/s.

Pentru aer, în practică se utilizează relația (2) care permite calculul vitezei de propagare a sunetelor în funcție de temperatură:

$$c_{\text{aer}} = (331,5 + 0,607 T) \quad [m \cdot s^{-1}] \quad (2)$$

unde T este temperatura aerului exprimată în grade Celsius.

Această formulă este aproximația liniară (primii doi termeni din seria Taylor) a funcției:

$$c_{\text{aer}} = 331,5 \sqrt{1 + \frac{T}{273,15}} \quad [m \cdot s^{-1}] \quad (3)$$

care permite calculul mai exact al acestei dependențe în ipoteza că variația cu temperatura a capacității calorice a aerului este nulă; erorile derivate din această ipoteză sunt mici în condițiile temperaturilor obișnuite din atmosferă, dar cresc în special la temperaturi înalte.

Coeficientul pentru aproximația liniară se obține astfel ca:

$$331,5 \cdot \frac{1}{2 \cdot 273,15} = 0,607 \quad (4)$$

Pentru realizarea proiectului am ales modulul HC-SR 04, care are conform fișei de catalog (data sheet) următorii parametri:

- Modulul HC-SR 04 poate realiza măsurarea distanțelor cuprinse între 2 cm și 400 cm cu o precizie de 3 mm.
- Acesta emite ultrasunete cu frecvența de 40 KHz, sub forma unui fascicul care are un unghi de 15 grade.
- Modulul HC-SR 04 funcționează la tensiunea de 5 V DC, cu un curent de 15 mA, și acceptă la pinul Trig impuls TTL cu durata de 10 microsecunde, iar la pinul Echo furnizează semnal cu nivel TTL.

În figura 1 se prezintă modulul HC-SR 04, având dimensiunile: 45x20x15 mm.

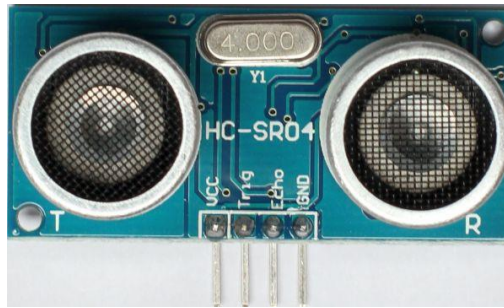


Fig 1 Senzorul ultrasonic HC-SR04 (the HC-SR04 ultrasonic ranging module)

Din punct de vedere constructiv modulul HC-SR-04 este prevăzut cu un traductor electro-acustic transmițător (T), un traductor acustic-electric receptor (R) și patru pini: Vcc, TRIG, ECHO, GND după cum se poate vedea din figura 1. Între pinul Vcc și pinul GND se aplică tensiunea de alimentare de +5 V. Atunci când se aplică la pinul TRIG un impuls de tensiune cu amplitudinea de 5V și cu durata de 10 microsecunde de la placa Arduino, oscilatorul electronic intern care funcționează la frecvența de 40 KHz alimentează traductor electro-acustic transmițător (T), astfel încât acesta va emite în spațiu o salvă (burst) de 8 impulsuri ultrasonore cu frecvența de 40 KHz, iar cronometrul intern al modulului HC-SR-04 va începe să măsoare timpul. Dacă un obiect se află în raza de acțiune, cele 8 impulsuri ultrasonore se vor reflecta înapoi la senzor. În această situație traductorul acustic-electric receptor (R), va converti semnalul ecou ultrasonor recepționat în semnal electric, iar modulul HC-SR 04 va livra la pinul ECHO un impuls TTL corespunzător care va fi utilizat la determinarea timpului de propagare a sunetului până la obiect și înapoi. După ce a fost determinat timpul de propagare al undei ultrasonore, deoarece viteza sunetului în aer este cunoscută se poate calcula cu ajutorul relației (1), distanța de la modulul HC-SR-04, până la obiect.

Funcționare în timp este reprezentată în figura 2

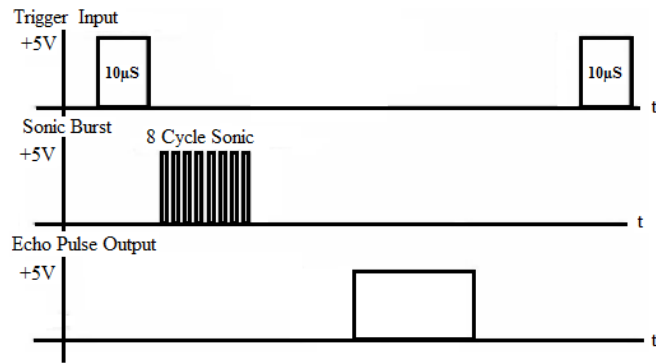


Figura 2 Timing Diagram

Dintre avantajele utilizării unui telemetrul cu ultrasunete se pot enumera:

- funcționează la fel de bine atât pe timp de lumină, cât și pe timp de întuneric, (ca urmare a propagării nestingherite a oscilației acustice de către lumină);
- poate fi utilizat în domenii ca: securitatea publică, detecția mișcări, robotică, măsurarea distanțelor, detecția obstacolelor la parcare a autovehiculelor, detectarea non-contact a nivelului lichidelor toxice, puternic acide sau puternic alcaline, etc.
- nu perturbă și nu este perturbat de comunicațiile radio, deoarece nu interferează cu undele electromagnetice.

2. Telemetru ultrasonic cu afișare pe monitor

Pentru aplicații care nu solicită o precizie deosebită, se poate realiza un telemetrul cu ultrasunete cu o construcție care utilizează un număr redus de componente, având schema electrică reprezentată în figura 3. Pentru realizarea proiectului experimental s-a folosit o placă Arduino, un modul HC-SR-04, fire de legătură și o placă breadboard după cum se poate vedea în figura 4.

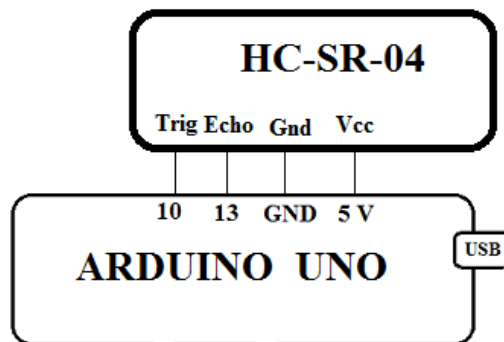


Figura 3 Schema electrică a Telemetrului ultrasonic cu afișare pe monitor

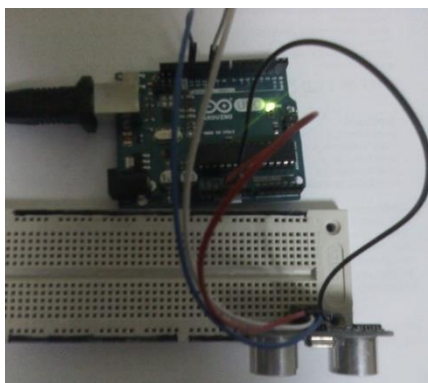


Figura 4

Montajul experimental ULTRASONIC MONITOR – HC-SR-04

Programul Ultrasonic Monitor – HC – SR - 04

În prima parte a programului se definește pinul 10 al plăcii Arduino ca fiind TRIG și se inițializează ca pin de ieșire, pinul 13 al plăcii Arduino ca fiind ECHO și se inițializează ca pin de intrare și se inițializează comunicația serială la 9600.

În partea a doua a programului, void loop, se definesc variabilele DURATA și DISTANTA.

Deoarece intervalul de timp se măsoară în microsecunde este util să exprimăm viteza sunetului la temperatura de 20 grade Celsius în centimetri pe microsecundă. Valoarea acesteia este de 0,0344 centimetri pe microsecundă, dar pentru a afla distanța până la obiect, vom înmulți timpul cu jumătatea acestei mărimi, adică 0,0172. Pentru situațiile în care distanța este mai mică de 2 cm, sau mai mare de 400 cm, pe monitor apare mesajul "IN AFARA RAZEI DE MASURARE".

Fiecare valoare măsurată va fi afișată pe monitor timp de 500 milisecunde, apoi programul se va repeta.

```
#define trigPin 10
#define echoPin 13
```

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

```
void loop() {
  float DURATA, DISTANTA;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

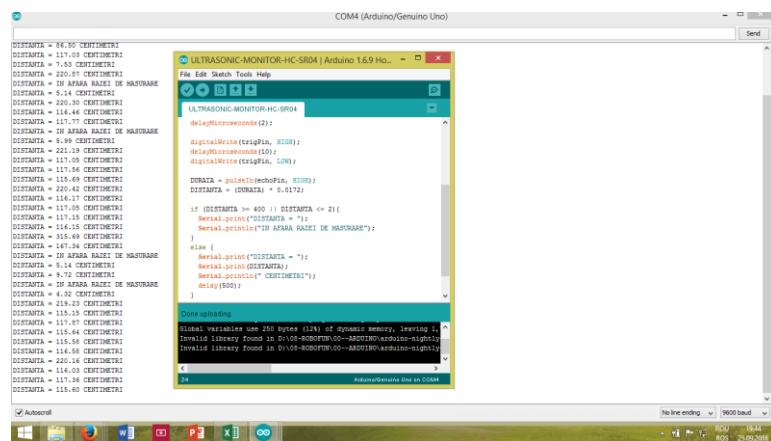
```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  DURATA = pulseIn(echoPin, HIGH);
```

```
DISTANTA = (DURATA) * 0.0172;
```

```
if (DISTANTA >= 400 || DISTANTA <= 2){  
  Serial.print("DISTANTA = ");  
  Serial.println("IN AFARA RAZEI DE MASURARE");  
}  
else {  
  Serial.print("DISTANTA = ");  
  Serial.print(DISTANTA);  
  Serial.println(" CENTIMETRI");  
  delay(500);  
}  
delay(500);  
}
```

Rezultatele experimentale se pot vedea în Figura 5 unde este un print-screen la monitor.



```
COM4 (Arduino/Genuino Uno)  
  
DISTANTA = 16.30 CENTIMETRI  
DISTANTA = 17.03 CENTIMETRI  
DISTANTA = 7.50 CENTIMETRI  
DISTANTA = 220.01 CENTIMETRI  
DISTANTA = 10 AFARA RAZEI DE MASURARE  
DISTANTA = 0.14 CENTIMETRI  
DISTANTA = 220.30 CENTIMETRI  
DISTANTA = 116.44 CENTIMETRI  
DISTANTA = 117.77 CENTIMETRI  
DISTANTA = 10 AFARA RAZEI DE MASURARE  
DISTANTA = 0.99 CENTIMETRI  
DISTANTA = 221.18 CENTIMETRI  
DISTANTA = 117.00 CENTIMETRI  
DISTANTA = 117.94 CENTIMETRI  
DISTANTA = 115.49 CENTIMETRI  
DISTANTA = 220.44 CENTIMETRI  
DISTANTA = 116.37 CENTIMETRI  
DISTANTA = 117.05 CENTIMETRI  
DISTANTA = 117.15 CENTIMETRI  
DISTANTA = 116.15 CENTIMETRI  
DISTANTA = 115.49 CENTIMETRI  
DISTANTA = 149.34 CENTIMETRI  
DISTANTA = 10 AFARA RAZEI DE MASURARE  
DISTANTA = 0.14 CENTIMETRI  
DISTANTA = 4.32 CENTIMETRI  
DISTANTA = 219.23 CENTIMETRI  
DISTANTA = 115.15 CENTIMETRI  
DISTANTA = 117.47 CENTIMETRI  
DISTANTA = 115.44 CENTIMETRI  
DISTANTA = 115.18 CENTIMETRI  
DISTANTA = 116.16 CENTIMETRI  
DISTANTA = 220.14 CENTIMETRI  
DISTANTA = 116.01 CENTIMETRI  
DISTANTA = 117.14 CENTIMETRI  
DISTANTA = 115.40 CENTIMETRI
```

Figura 5 Print-screen facut la monitor

3. Telemetru ultrasonic cu afișare pe display LCD 16x2

Pentru aplicații în care se dorește afișarea distanței, nu pe monitor, ci pe un display LCD, se poate realiza un telemetru cu ultrasunete cu o construcție care utilizează un număr redus de componente, având schema electrică reprezentată în figura 6. Pentru realizarea proiectului experimental s-a folosit o placă Arduino, un display LCD 16x2, un modul HC-SR-04, fire de legătură și o placă breadboard după cum se poate vedea în figura 7.

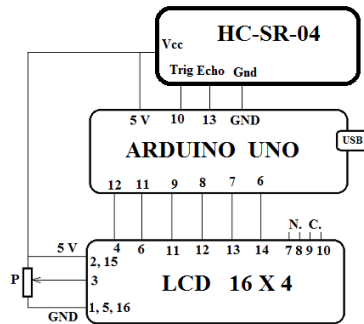


FIGURA 6 SCHEMA ELECTRICA PENTRU TELEMETRU ULTRASONIC-LCD-1

Programul Telemetru Ultrasonic – LCD - 1

În prima parte a programului, față de exemplul anterior, a mai fost inclusă biblioteca <LiquidCrystal.h>, pentru a se putea utiliza display-ul LCD 16x2 și s-au configurat pinii la care este conectat display-ul LCD 16x2.

În partea a doua a programului, void loop, se definesc variabilele DURATA și DISTANTA, iar pentru a afla distanța până la obiect, vom considera viteza sunetului de 0,0344 centimetri pe microsecundă și vom împărți timpul la 2. În cazul în care distanța este mai mică de 2 cm, sau mai mare de 400 cm, pe display-ul LCD 16x2 apare mesajul (în formă redusă) "IN AFARA RAZEI". Fiecare valoare măsurată va fi afișată pe monitor timp de 500 milisecunde, apoi programul se va repeta.

```
#include <LiquidCrystal.h>
#define trigPin 10
#define echoPin 13

LiquidCrystal lcd(12, 11, 9, 8, 7, 6);

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  float DURATA, DISTANTA;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  DURATA = pulseIn(echoPin, HIGH);
```

```

DISTANTA = (DURATA / 2) * 0.0344;

if (DISTANTA >= 400 || DISTANTA <= 2){
  lcd.print("IN AFARA RAZEI");
  delay(500);
}
else {
  lcd.print(DISTANTA);
  lcd.print(" CM ");
  delay(500);
}
delay(500);
lcd.clear();
}

```

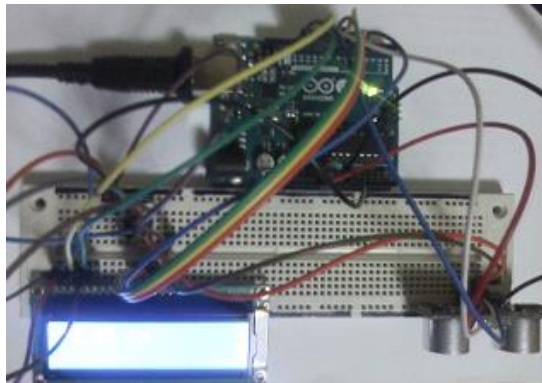


FIGURA 7 MONTAJUL EXPERIMENTAL TELEMETRU ULTRASONIC-LCD-1

4. Telemetru ultrasonic cu afișare pe monitor având compensarea temperaturii cu termistor

Deoarece viteza sunetului este dependentă de temperatură, este necesar să corectăm această mărime pentru a obține o precizie mai bună a măsurătorilor. Schema electrică a telemetrului ultrasonic cu afișare pe monitor la care se corectează viteza sunetului în funcție de temperatură cu ajutorul unui termistor este reprezentată în figura 8.

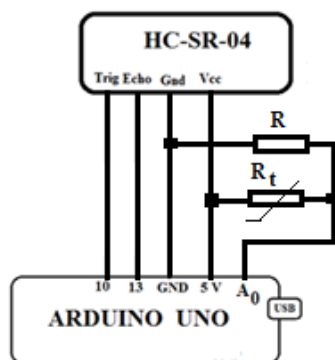


Figura 8

Schema electrică a telemetrului ultrasonic cu afișare pe monitor la care se realizează compensarea temperaturii cu termistor

În lucrarea aceasta s-a utilizat un termistor de măsurare cu coeficient de temperatură negativ – NTC, având rezistența de $10\text{ K}\Omega$, la temperatura de 25°C .

Se cunoaște că un semiconductor de tip **n**, extrinsec puternic dopat, prezintă o conductivitate electrică conform relației:

$$\sigma = e \cdot n_n \cdot \mu_n = \frac{1}{\rho} \quad (5)$$

unde:

σ este conductivitatea electrică,

e este sarcina electronului,

n_n este concentrația electronilor (dependentă de temperatură),

μ_n este mobilitatea electronilor (dependentă de temperatură),

ρ este rezistivitatea materialului.

Ținând cont de relația (5) și de variația cu temperatura a energiei de activare se obține o relație care descrie variația rezistenței în funcție de temperatură pentru un termistor NTC sub forma:

$$R(T) = A \cdot \exp\left(\frac{B}{T} + \frac{C}{T^2} + \frac{D}{T^3} + \dots\right) \quad (6)$$

Pentru modelarea dependenței rezistenței în funcție de temperatură se folosește ecuația Steinhart - Hart, care are expresia:

$$\frac{1}{T} = A + B \cdot \ln R + C \cdot (\ln R)^3 \quad (7)$$

unde:

T este temperatura absolută,

A, B, C sunt constante obținute prin măsurători (disponibile în cataloage).

Pentru realizarea proiectului experimental s-a folosit o placă Arduino, un modul HC-SR-04, un termistor de 10 K , un rezistor de 10 K , fire de legătură și o placă breadboard după cum se poate vedea în figura 9.

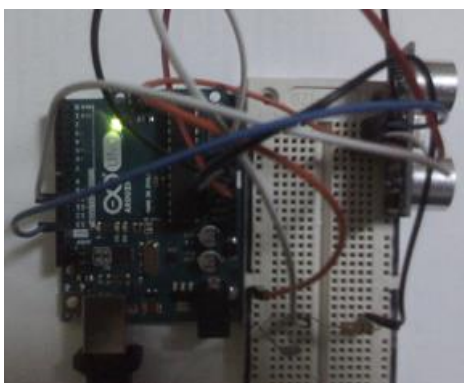


Figura 9

Montajul experimental al telemetrului ultrasonic cu afișare pe monitor la care se realizează compensarea temperaturii cu termistor

Programul Ultrasonic Monitor – Termistor – HC – SR - 04

În prima parte a programului se definește pinul 10 al plăcii Arduino ca fiind TRIG și se inițializează ca pin de ieșire, pinul 13 al plăcii Arduino ca fiind ECHO și se inițializează ca pin de intrare și se inițializează comunicația serială la 9600 – ca în primul exemplu. În plus s-a inclus biblioteca `<math.h>`, precum și ecuația Steinhart – Hart, necesară pentru a modela dependența rezistenței termistorului în funcție de temperatură.

În partea a doua a programului, void loop, se definesc variabilele durata, distanța și viteza. Apoi se calculează viteza sunetului în metri pe secundă, ținând cont de temperatură, cu ajutorul relației (2). – folosind instrucțiunea: $spdSnd = 331.5 + (0.607 * temp)$. Pentru a determina distanța, întrucât intervalul de timp se măsoară în microsecunde, exprimăm viteza sunetului în centimetri pe microsecundă, și înmulțim această valoare cu jumătatea duratei de timp necesară sunetului să parcurgă drumul dus-întors. Pentru situațiile în care distanța este mai mică de 2 cm, sau mai mare de 400 cm, pe monitor apare mesajul "IN AFARA RAZEI DE MASURARE". Fiecare valoare măsurată va fi afișată pe monitor timp de 500 milisecunde, apoi programul se va repeta.

```
#include <math.h>
#define trigPin 10
#define echoPin 13
```

```
double Thermistor(int RawADC) {
  double Temp;
  Temp = log(10000.0*((1024.0/RawADC-1)));
  Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp );
  Temp = Temp - 273.15;
  return Temp;
}
```

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
```

```

pinMode(echoPin, INPUT);
}

void loop() {
  int val;
  double temp;
  val=analogRead(0);
  temp=Thermistor(val);

  float durata, distanta;
  float spdSnd;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  durata = pulseIn(echoPin, HIGH);
  spdSnd = 331.5 + (0.607 * temp);
  distanta = (durata / 2) * (spdSnd / 10000);

  if (distanta >= 400 || distanta <= 2){
    Serial.print("DISTANTA ESTE ");
    Serial.println("IN AFARA RAZEI DE MASURARE");
  }
  else {
    Serial.print("DISTANTA ESTE = ");
    Serial.print(distanta);
    Serial.println(" CENTIMETRI");
    delay(1000);
  }
  delay(1000);
}

```



Figura 10

Rezultatele experimentale (print-screen) ale telemetrului ultrasonic cu afișare pe monitor la care se realizează compensarea temperaturii cu termistor

5. Telemetru ultrasonic cu afișare pe monitor având compensarea temperaturii și umidității cu modulul DHT-11

Modulul DHT-11 conține un senzor pentru măsurare a umidității și un termistor NTC pentru măsurarea temperaturii și se poate conecta la un microcontroler de 8 biți, oferind un răspuns rapid, un nivel anti-interferențe bun și o eficiență ridicată. Fiecare modul DHT-11 este calibrat în laborator, iar coeficienții de calibrare sunt stocați ca programe în memoria OTP, care sunt utilizate de către procesul de detectare a semnalului intern senzorului.

Principali parametri ai modulului DHT-11, sunt:

Relative humidity

Resolution: 16Bit

Repeatability: $\pm 1\%$ RH

Accuracy: At 25°C $\pm 5\%$ RH

Temperature

Resolution: 16Bit

Repeatability: $\pm 0.2^{\circ}\text{C}$

Range: At 25°C $\pm 2^{\circ}\text{C}$

Power supply: DC 3.5~5.5V

Supply Current: measurement 0.3mA standby 60 μ A

Sampling period: more than 2 seconds

Pentru aplicațiile în care se dorește o precizie mare se va ține cont că viteza sunetului este dependentă de temperatură și umiditate, conform relației:

$$c = 331.5 + (0.607 * T) + (0.0124 * H) \quad (8)$$

unde

T – este temperatura,

H – este umiditatea.

Schema electrică a telemetrului ultrasonic cu afișare pe monitor la care se realizează compensarea temperaturii și umidității cu modulul DHT-11 este reprezentată în figura 11.

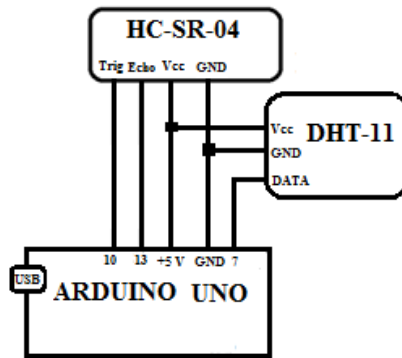


Figura 11

Schema electrică a telemetrului ultrasonic cu modul DHT-11 și afișare pe monitor

Pentru realizarea montajului experimental al telemetrului ultrasonic compensat cu temperatura și umiditatea cu modul DHT-11 și afișare pe monitor s-au utilizat: o placă de dezvoltare Arduino, un modul HC-SR-04, un modul DHT-11, fire de legătură și o placă breadboard după cum se poate vedea în figura 12.

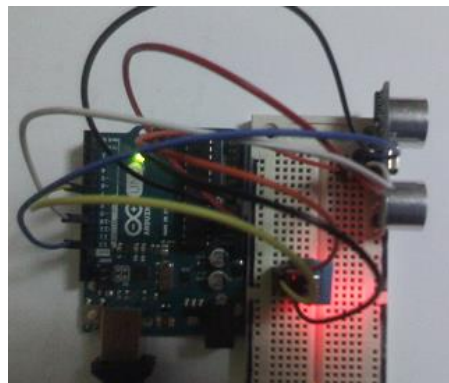


Figura 12 Montajul experimental al telemetrului ultrasonic compensat cu temperatura și umiditatea cu modul DHT-11 și afișare pe monitor

Programul Ultrasonic monitor compensat DHT-11 cu temperatura și umiditatea

În prima parte a programului se definește pinul 10 al plăcii Arduino ca fiind TRIG și se inițializează ca pin de ieșire, pinul 13 al plăcii Arduino ca fiind ECHO și se inițializează ca pin de intrare și se inițializează comunicația serială la 9600 – ca în primul exemplu. În plus s-a inclus biblioteca <dht.h>.

În partea a doua a programului, void loop, se definesc variabilele durata, distanța și viteza. Apoi se calculează viteza sunetului în metri pe secundă, ținând cont de temperatură și de umiditate, cu ajutorul relației (8). – folosind instrucțiunea: $speed = 331.5 + (0.607 * DHT.temperature) + (0.0124 * DHT.humidity)$;

Pentru a determina distanța, întrucât intervalul de timp se măsoară în microsecunde, exprimăm viteza sunetului în centimetri pe microsecundă, și înmulțim această valoare cu jumătatea duratei de timp necesară sunetului să parcurgă drumul dus-întors. Pentru situațiile

în care distanța este mai mică de 2 cm, sau mai mare de 400 cm, pe monitor apare mesajul "IN AFARA RAZEI DE MASURARE". Fiecare valoare măsurată va fi afișată pe monitor timp de 500 milisecunde, apoi programul se va repeta.

```
#include <dht.h>

#define trigPin 10
#define echoPin 13
#define DHT11_PIN 7

dht DHT;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

  float durata, distanta;
  float speed;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  durata = pulseIn(echoPin, HIGH);
  speed = 331.5 + (0.607 * DHT.temperature) + (0.0124 * DHT.humidity);
  distanta = (durata / 2) * (speed / 10000);

  if (distanta >= 400 || distanta <= 2){
    Serial.print("DISTANTA = ");
    Serial.println("IN AFARA RAZEI");
  }
  else {
    Serial.print("DISTANTA = ");
    Serial.print(distanta);
    Serial.println(" centimetri");
    delay(1000);
  }
  delay(1000);
}
```

După realizarea fizică a montajului experimental al telemetrului ultrasonic compensat cu temperatura și umiditatea cu modul DHT-11 și afișare pe monitor, încărcarea programului și rularea acestuia, se pot observa valorile în centimetri ale distanțelor măsurate, ca în figura 13.

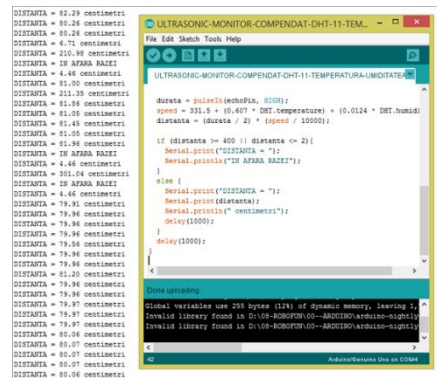


Figura 13

Rezultatele experimentale (Print-screen) ale telemetrului ultrasonic compensat cu temperatura și umiditatea cu modul DHT-11 și afișare pe monitor

6. Telemetru ultrasonic cu afișare pe LCD 16x2 având compensarea temperaturii și umidității cu modulul DHT-11

Schema electrică a telemetrului ultrasonic cu afișare pe display LCD 16x2, la care se realizează compensarea temperaturii și umidității cu modulul DHT-11 este reprezentată în figura 14.

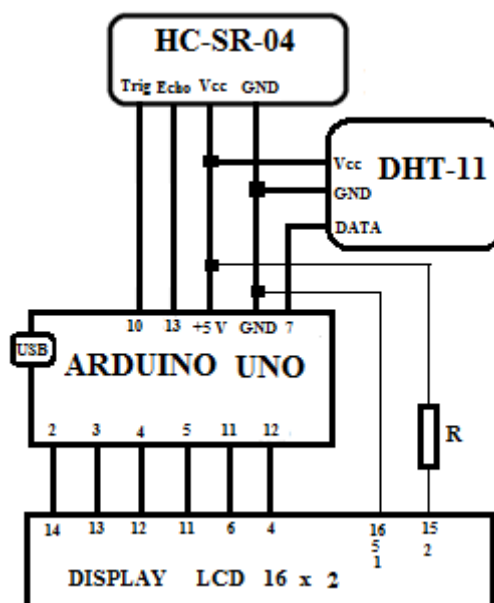


Figura 14

Schema electrică a telemetrului ultrasonic cu afișare pe display LCD 16x2 având compensarea temperaturii și umidității cu modulul DHT-11

Pentru realizarea montajului experimental al telemetrului ultrasonic compensat cu temperatura și umiditatea cu modul DHT-11 și afișare pe display LCD 16x2 s-au utilizat: o placă de dezvoltare Arduino, un modul HC-SR-04, un modul DHT-11, un display LCD 16x2, fire de legătură și o placă breadboard, iar rezultatele experimentale se pot vedea în figura 15.

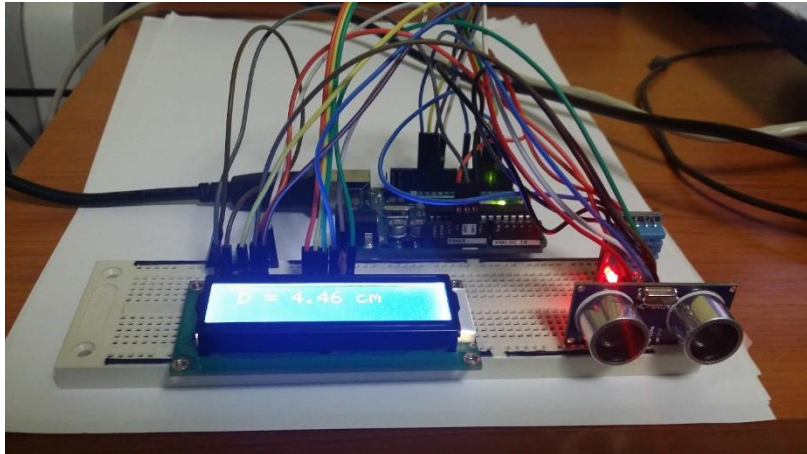


Figura 15

Montajul experimental al telemetrului ultrasonic cu afișare pe display LCD 16 x 2 având compensarea temperaturii și umidității cu modulul DHT-11

Programul telemetrului ultrasonic cu afișare pe display LCD 16x2 având compensarea temperaturii și umidității cu modulul DHT-11

În prima parte a programului se definește pinul 10 al plăcii Arduino ca fiind TRIG și se inițializează ca pin de ieșire, pinul 13 al plăcii Arduino ca fiind ECHO și se inițializează ca pin de intrare – ca în primul exemplu. În plus s-a inclus biblioteca <dht.h> și biblioteca <LiquidCrystal.h>, iar pentru a se putea utiliza display-ul LCD 16x2 s-au configurat pinii la care este conectat display-ul LCD 16x2.

În partea a doua a programului, void loop, se definesc variabilele durata, distanța și viteza. Apoi se calculează viteza sunetului în metri pe secundă, ținând cont de temperatură și de umiditate, cu ajutorul relației (8). – folosind instrucțiunea: $speed = 331.5 + (0.607 * DHT.temperature) + (0.0124 * DHT.humidity)$;

Pentru a determina distanța, întrucât intervalul de timp se măsoară în microsecunde, exprimăm viteza sunetului în centimetri pe microsecundă, și înmulțim această valoare cu jumătatea duratei de timp necesară sunetului să parcurgă drumul dus-întors. Pe durata efectuării măsurătorilor de distanță pe display-ul LCD 16x2, se va afișa D = (valoarea măsurată a distanței) cm. Pentru situațiile în care distanța este mai mică de 2 cm, sau mai mare de 400 cm, pe display-ul LCD 16x2 apare mesajul (în formă restrânsă) "IN AFARA

RAZEI". Fiecare valoare măsurată va fi afișată pe monitor timp de 500 milisecunde, apoi programul se va repeta.

```
#include <dht.h>
#include <LiquidCrystal.h>

#define trigPin 10
#define echoPin 13
#define DHT11_PIN 7

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

dht DHT;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

  float durata, distanta;
  float speed;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  durata = pulseIn(echoPin, HIGH);
  speed = 331.5 + (0.607 * DHT.temperature) + (0.0124 * DHT.humidity);
  distanta = (durata / 2) * (speed / 10000);

  if (distanta >= 400 || distanta <= 2){
    lcd.print("IN AFARA RAZEI");
    delay(500);
  }
  else {
    lcd.print(" D = ");
    lcd.print(distanta);
    lcd.print(" cm");
    delay(500);
  }
  delay(500);
  lcd.clear();
}
```


7. Concluzii

Necesitatea utilizării unui telemetru cu ultrasunete pentru realizarea de măsurători în medii toxice, în spații inaccesibile, într-o atmosferă cu poluare electromagnetică, etc., este evidentă și de necontestat.

Nucleul telemetrului cu ultrasunete este microcontrolerul Atmega 328 P-PU, cu o arhitectură RISC, având 32 registre pe 8 biți și un număr de 131 de instrucțiuni, asigurând resurse mai mult decât suficiente pentru realizarea unui telemetru cu ultrasunete. Spre deosebire de alte tipuri de microcontrolere, programarea microcontrolerului Atmega 328 P-PU, este mult ușurată atât din punct de vedere hardware – familia Arduino cuprinde mai multe plăci de dezvoltare – cât și din punct de vedere software – mediul de programare IDE Arduino este gratuit pentru sistemele de operare actuale (Windows, Mac OS X, Linux).

Prețul scăzut al microcontrolerului Atmega 328 P-PU, este un alt motiv care a stat la alegerea schemei electronice a telemetrului cu ultrasunete. Acesta după ce a fost programat, este replantat pe placa electronică a telemetrului cu ultrasunete.

Posibilitatea realizării mai multor variante de telemetre cu ultrasunete în jurul microcontrolerului Atmega 328 P-PU, oferă avantajul alegerii variantei constructive optime în funcție de aplicația dorită. Astfel pentru aplicații puțin pretențioase se poate implementa schema electronică fără compensarea temperaturi mediului, aceasta fiind și cea mai ieftină. Dacă se dorește implementarea schemei electronice cu compensarea temperaturi mediului, atunci varianta cu termistor este mai ieftină, dar mai puțin precisă decât cea cu senzorul de temperatură și umiditate DHT-11.

În concluzie, diversitatea variantelor constructive permite alegerea soluției tehnice adecvate aplicației practice, precum și posibilitatea alegerii raportului optim performanțe preț.

8. Bibliografie

1. Data sheet module HC – SR 04
2. John Boxall - ARDUINO WORKSHOP – 2013 - ISBN-13: 978-1-59327-448-1 , pag. 147 – 151. (pentru LCD)
3. Temperature and humidity module DHT-11 Product Manual
4. Emily Gertz and Patrick Di Justo – Environmental Monitoring with Arduino – 2012- ISBN: 978-1-449-31056-1 , pag. 51 – 62, (pentru DHT-11)
5. Michael McRoberts - Beginning Arduino – 2010 - ISBN-13 (electronic): 978-1-4302-3241-4 , pag. 293 – 316 (ultrasonic LV-MaxSonar EZ3)
6. ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32K BYTES IN-SYSTEM PROGRAMMABLE FLASH
7. NTC thermistors for temperature measurement.pdf
8. https://en.wikipedia.org/wiki/Steinhart-Hart_equation
9. <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

10. <https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay>
11. <https://www.arduino.cc/en/Main/Software>
12. <https://www.arduino.cc/en/Tutorial/LibraryExamples>
13. Răzvan-Daniel ALBU – Design and implementation of a thermometer with Arduino and thermistor -