APLICAȚII ALE PROGRAMULUI C++ PENTRU MANAGEMENTUL CASEI INTELIGENTE

Pițigoi Andrei, Prof. Ing., Colegiul Tehnic **Petru Maior**, Bucuresti,
pitigoi4111@yahoo.com
Gaidoș Nicoleta, Prof. Ing., Colegiul Tehnic **Mircea cel Batran**, Bucuresti,
gaidosnil@yahoo.com

Rezumat. Pentru Simpozionul SINUC 2017 organizat de Facultatea de Utilaj Tehnologic, Universitatea Tehnica de Constructii București, profesorii Pițigoi Andrei și Gaidoș Nicolatea au realizat o temă care se încadrează la secțiunea "Case inteligente" și anume "Aplicații ale programului C++ pentru managementul casei inteligente" . Realizarea comenzilor prin GSM are drept obiectiv optimizarea funcționării instalațiilor și echipamentelor electrice, utilizând instrucțiuni de programare în limbajul C++.

1. NOȚIUNI INTRODUCTIVE

Conceptul de casă inteligentă (fig.1), prin contrast cu o casă tipică, reprezintă de exemplu: pornirea aerului condiționat, închiderea luminii sau reglarea căldurii.
Electronicele și electrocasnicele într-o casă inteligentă pot fi comandate centralizat, de la distanță. Utilizatorii vor putea crea principii de automatizare de genul "pornește centrala termică și verifică temperatura".
În prima fază inteligența se va traduce în posibilitatea de a controla de la distanță dispozitivele din casă și setarea de reguli automate pentru funcționarea acestora. Poate în viitor vom vorbi și de inteligențe artificiale care vor decide ce este mai bine pentru noi, însă momentan nu este cazul.

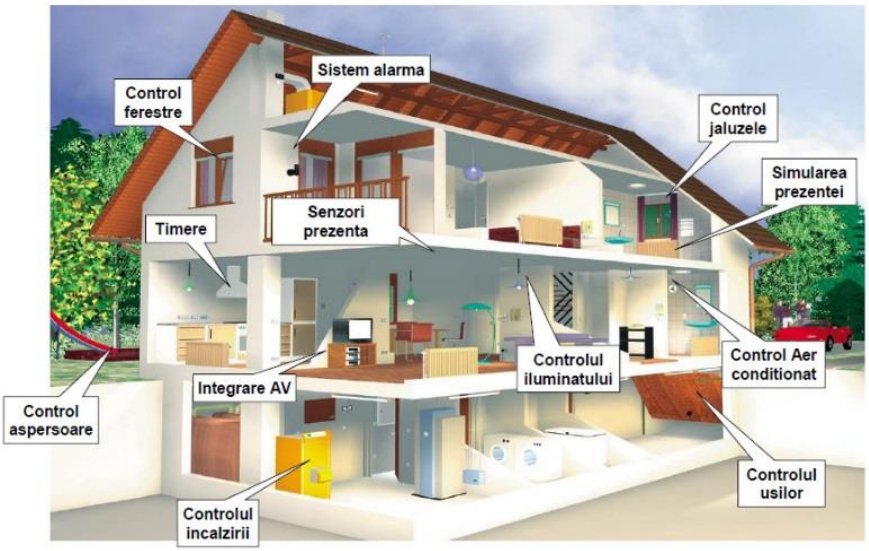

Fig.1

2. ASPECTE CONSTRUCTIVE

Pentru realizarea proiectului am avut în vedere să proiectez următoarele componente: componenta hardwer și componenta softwer.
componenta hardwer cuprinde următoarele componente: conductoare electrice,
relee de comandă, motor de c.c., modul GSM, dulie, bec, radiator electric, senzor de temperatură, tastatură, senzor fotoelectric, alimentatoare și leduri.
Componenta software cuprinde următorul program: Arduino.
În figura 2 se prezintă schema bloc pentru simplificarea funcționării comenzilor GSM [5].

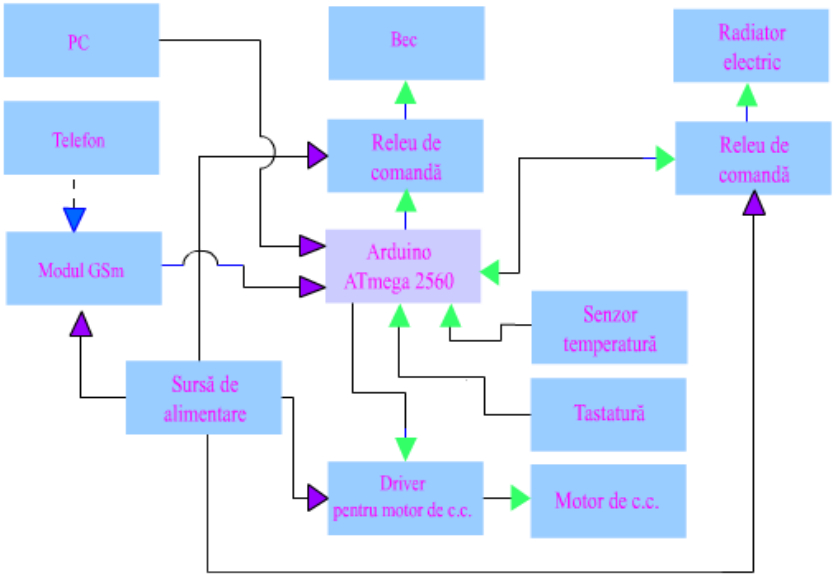

Fig. 2

3. FUNCȚIONAREA ELEMENTELOR COMPONENTE

3.1. ARDUINO ATMEGA 2560

În figura 3 se prezintă placa de dezvoltare Arduino Atmega 2560.

Fig. 3

3.2. RELEU DE COMANDĂ

Fig. 4

Program:
Releul de comandă (fig. 4) acționează aprindere și stingerea unui led.

```
void setup() {
    Serial.begin(9600);
    pinMode(36,OUTPUT);
}
void loop() {
  bec();
}
void bec(){
  digitalWrite(36,HIGH);
  delay(1000);
  digitalWrite(36,LOW);
  delay(1000);
}
```

3.3. MODUL GSM

Fig. 5

Program EMISIE:
Modul GSM (fig 5) transmite la telefon o valoare numerică sub formă zecimală.

```
char b[10];
int c;
void setup() {
  Serial.begin(115200);
  Serial2.begin(38400);
}
void loop() {
  while(Serial2.available()) Serial.write(Serial2.read());
  while(Serial.available()) Serial2.write(Serial.read());
```

```
  trimitere();
  delay(100);
}
void trimitere()
{
  Serial2.print("\r");
  delay(1000);
  Serial2.print("AT+CSCS=\"GSM\"\r");
  delay(1000);
  Serial2.print("AT+CMGF=1\r");
  delay(1000);
  Serial2.print("AT+CMGS=\"+40727083523\"\r");
  delay(1000);
  c=250;
  b[10]=itoa(c,b,10);
  Serial2.print(b);
  delay(1000);
  Serial2.write(0x1A);
  delay(1000);
}
```

Program RECEPȚIE:
Telefonul transmite un SMS la modul GSM.

```
int a;
bool ledStatus;
void setup()
{
  Serial.begin(115200);
  Serial2.begin(38400);

  delay(100);
  Serial2.print("AT+CMGF=1\r");
   delay(100);
   Serial2.print("AT+CSCS=\"GSM\"");
   Serial2.print("\r");
   delay(100);
  Serial2.print("AT+CMGF=1\r");
  delay(100);
  Serial2.print("AT+CNMI=1,2,0,0,0\r");
  delay(100);
  Serial2.println("AT+CMGD=1");
  delay(100);
 }
char currentLine[500] = "";
int currentLineIndex = 0;
bool nextLineIsMessage = false;
void loop() {
 if(Serial2.available()){
    char lastCharRead = Serial2.read();
     if(lastCharRead == '\r' || lastCharRead == '\n'){
```

```
      String lastLine = String(currentLine);

      if(lastLine.startsWith("+CMT:")){
        Serial.println(lastLine);
        nextLineIsMessage = true;
      }else if(lastLine.length() > 0) {
        if(nextLineIsMessage) {
          Serial.println(lastLine);
     a=atoi(lastLine.c_str());
          Serial.println(a);
           if(a==44){
            Serial.println("adevarat");
           }else{
            Serial.println("fals");
           }
          Serial.println("jhgft");

          if(lastLine.indexOf("LED ON") >= 0){
            ledStatus = 1;
           }else if(lastLine.indexOf("LED OFF") >= 0) {
            ledStatus = 0;
           }
         nextLineIsMessage = false;
         }
       }

      for( int i = 0; i < sizeof(currentLine);  ++i ) {
       currentLine[i] = (char)0;
      }
      currentLineIndex = 0;
    }else{
      currentLine[currentLineIndex++] = lastCharRead;
    }
  }
}
```

3.4. DRIVER PENTRU MOTOR DE C.C.

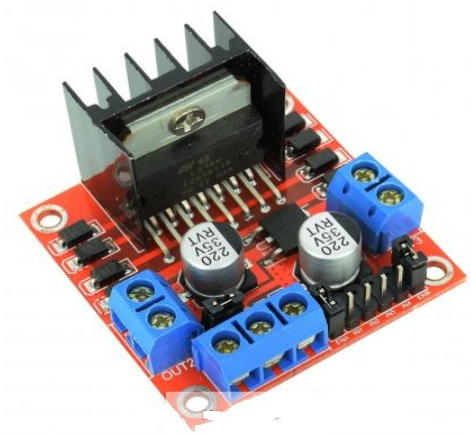

Fig. 6

Program:
Driver-ul (fig. 6) acționează motorul electric de c.c.

```
int MOTOR1_PIN1 = 2;
int MOTOR1_PIN2 = 3;
void setup() {
  pinMode(MOTOR1_PIN1, OUTPUT);
  pinMode(MOTOR1_PIN2, OUTPUT);
  Serial.begin(9600);
}
void loop() {
   motor();
}
void motor() {
     analogWrite(MOTOR1_PIN1,200);
     analogWrite(MOTOR1_PIN2,0);
     delay (1000);
     analogWrite(MOTOR1_PIN1,0);
     analogWrite(MOTOR1_PIN2,200);
     delay(1000);
 }
```

3.5. TASTATURĂ

Fig. 7

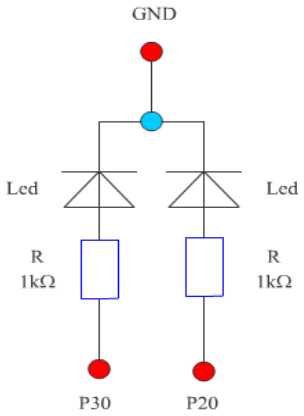

Fig. 8

Program:
Tastatura (fig. 7) pornește motorul electric de c.c. și aprinde ledurile (fig. 8).

```
#include <Keypad.h>
int MOTOR1_PIN1 = 2;
int MOTOR1_PIN2 = 3;
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','4','7','*'},
  {'2','5','8','0'},
  {'3','6','9','#'},
  {'A','B','C','D'}
};
```

```
byte rowPins[ROWS] = {44, 42, 40, 38};
byte colPins[COLS] = {52, 50, 48, 46};

Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void setup(){
  Serial.begin(9600);
  pinMode(30,OUTPUT);
  pinMode(20,OUTPUT);
  pinMode(MOTOR1_PIN1, OUTPUT);
  pinMode(MOTOR1_PIN2, OUTPUT);
}
void loop(){
   tastatura();
}
void tastatura(){
  char customKey = customKeypad.getKey();
  if(customKey=='1'){
     Serial.println(customKey);
     digitalWrite(30,HIGH);
     digitalWrite(20,LOW);
     analogWrite(MOTOR1_PIN1,200);
     analogWrite(MOTOR1_PIN2,0);
  }else if(customKey=='2'){
     Serial.println(customKey);
     digitalWrite(20,HIGH);
     digitalWrite(30,LOW);
     analogWrite(MOTOR1_PIN1,0);
     analogWrite(MOTOR1_PIN2,200);
  }else if(customKey=='3'){
     Serial.println(customKey);
     digitalWrite(30,LOW);
     digitalWrite(20,LOW);
     analogWrite(MOTOR1_PIN1,0);
     analogWrite(MOTOR1_PIN2,0);
  }
}
```

3.6. SENZOR DE TEMPERATURĂ

Fig. 9

Program:
Senzorul de temperatură (fig. 9) citește temperatura într-un mediu ambiant.

```
int count=10;
int reztemp;
void setup() {
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop() {
  temperatura();
  Serial.println(reztemp);
  delay(200);
}
void temperatura(){
  int sumaTemperatura=0;
  int temperatureCelsius=0;
   for (int i =0; i<count; i++) {
    int reading=analogRead(A0);
     float voltage =reading*5.0;
     voltage/=1024.0;
     temperatureCelsius=(voltage-0.5)*100;
     sumaTemperatura = sumaTemperatura + temperatureCelsius;
     reztemp=sumaTemperatura/count;
    }
}
```

3.7. SENZOR FOTOELECTRIC

Fig. 10

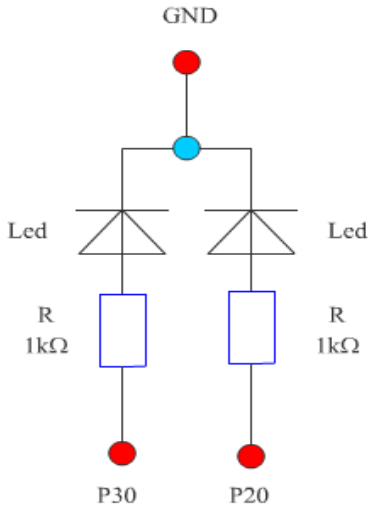

Fig. 11

Program:
Senzorul fotoelectric (fig. 10) comandă digital aprindere și stingerea unor leduri (fig. 11).

```
int cu;
void setup(){
```

```
   Serial.begin(9600);
   pinMode(20,OUTPUT);
   pinMode(30,OUTPUT);
}
void loop(){
   contact_use();
}
void contact_use(){
  cu=digitalRead(32);
  if(cu==0){
    digitalWrite(20,LOW);
    digitalWrite(30,HIGH);
  }else if(cu==1){
    digitalWrite(20,HIGH);
    digitalWrite(30,LOW);
  }
}
```

4. LIMBAJ DE PROGRAMARE C++

```
// bibliotecă pentru tastatură
   #include <Keypad.h>
//pin motoare
  int MOTOR1_PIN1 = 2;
   int MOTOR1_PIN2 = 3;
//pentru senzor ușe
   int cu;
//int temperatură
  int count=10;
  int reztemp;
//tastatură
  char* secretCode = "5843";
  int position = 0;
  int position1=0;
  const byte rows = 4;
  const byte cols = 4;
  char keys[rows][cols] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
   };
//conexiunile pentru pinii de la rănduri
  byte rowPins[rows] = {52, 50, 48, 46};
// conexiunile pentru pinii de la coloane
  byte colPins[cols] = {44, 42, 40, 38};
  Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);
 int a;
 bool ledStatus;
void setup(){
```

```
 //Seriale
  Serial.begin(115200);
  Serial2.begin(38400);
 //aprinde radiator
   pinMode(34,OUTPUT);
   pinMode(26,OUTPUT);
 //aprinde bec 220
   pinMode(36,OUTPUT);
 //pin bec use
    pinMode(30,OUTPUT);
    pinMode(28,OUTPUT);
  //pini motor
    pinMode(MOTOR1_PIN1, OUTPUT);
    pinMode(MOTOR1_PIN2, OUTPUT);
  //semnal GSM
   delay(100);
   Serial2.print("AT+CMGF=1\r");
   delay(100);
   Serial2.print("AT+CSCS=\"GSM\"");
   Serial2.print("\r");
   delay(100);
   Serial2.print("AT+CMGF=1\r");  // trimite SMS-ul sub formă de text
   delay(100);
   Serial2.print("AT+CNMI=1,2,0,0,0\r");
   delay(100);
   Serial2.println("AT+CMGD=1"); // șterge toate SMS-urile 1,4
   delay(100);

}
  char currentLine[500] = "";
  int currentLineIndex = 0;
  bool nextLineIsMessage = false;
 void loop() {
  contact_use();
  tastatura();
  if(Serial2.available()){
     char lastCharRead = Serial2.read();
       //Citiți fiecare caracter din ieșirea seriei până la atingerea lui \ r sau \ n (care indică sfârșitul
       liniei)
     if(lastCharRead == '\r' || lastCharRead == '\n'){
        String lastLine = String(currentLine);
        //Dacă a fost citită ultima linie + CMT, s-au primit mesaje SMS noi.
        //Prin urmare, următoarea linie este conținutul mesajului.
        if(lastLine.startsWith("+CMT:")){
           Serial.println(lastLine);
           nextLineIsMessage = true;
        }else if(lastLine.length() > 0) {
           if(nextLineIsMessage) {
             Serial.println(lastLine); //afișare string
             a=atoi(lastLine.c_str()); //convertire din string în număr
```

```
  while(lastLine=="deschis"){
      analogWrite(MOTOR1_PIN1,0);
      analogWrite(MOTOR1_PIN2,120);
      delay(300);
      analogWrite(MOTOR1_PIN1,0);
      analogWrite(MOTOR1_PIN2,0);
      while(Serial2.available()) Serial.write(Serial2.read());
      while(Serial.available()) Serial2.write(Serial.read());
      Serial2.print("\r");
      delay(100);
      Serial2.print("AT+CSCS=\"GSM\"\r");
      delay(100);
      Serial2.print("AT+CMGF=1\r");
      delay(100);
      Serial2.print("AT+CMGS=\"+40724914405\"\r");
      delay(100);
      Serial2.print("Use deschisa");
      delay(100);
      Serial2.write(0x1A);
      delay(100);
      lastLine="ande";
  }
  while(lastLine=="inchis"){
     analogWrite(MOTOR1_PIN1,120);
     analogWrite(MOTOR1_PIN2,0);
     delay(300);
     analogWrite(MOTOR1_PIN1,0);
     analogWrite(MOTOR1_PIN2,0);
      while(Serial2.available()) Serial.write(Serial2.read());
      while(Serial.available()) Serial2.write(Serial.read());
      Serial2.print("\r");
      delay(100);
      Serial2.print("AT+CSCS=\"GSM\"\r");
      delay(100);
      Serial2.print("AT+CMGF=1\r");
      delay(100);
      Serial2.print("AT+CMGS=\"+40724914405\"\r");
      delay(100);
      Serial2.print("Use inchisa");
      delay(100);
      Serial2.write(0x1A);
      delay(100);
     lastLine="nimic";
  }
   analogWrite(MOTOR1_PIN1,0);
   analogWrite(MOTOR1_PIN2,0);

//aprindere si stinge bec
 if(lastLine=="aprins"){
      digitalWrite(36,HIGH);
```

```
         while(Serial2.available()) Serial.write(Serial2.read());
        while(Serial.available()) Serial2.write(Serial.read());
        Serial2.print("\r");
        delay(100);
        Serial2.print("AT+CSCS=\"GSM\"\r");
        delay(100);
        Serial2.print("AT+CMGF=1\r");
        delay(100);
        Serial2.print("AT+CMGS=\"+40724914405\"\r");
        delay(100);
        Serial2.print("Bec aprins");
        delay(100);
        Serial2.write(0x1A);
        delay(100);
 }else if(lastLine=="stins"){
      digitalWrite(36,LOW);
        while(Serial2.available()) Serial.write(Serial2.read());
        while(Serial.available()) Serial2.write(Serial.read());
        Serial2.print("\r");
        delay(100);
        Serial2.print("AT+CSCS=\"GSM\"\r");
        delay(100);
        Serial2.print("AT+CMGF=1\r");
        delay(100);
        Serial2.print("AT+CMGS=\"+40724914405\"\r");
        delay(100);
        Serial2.print("Bec stins");
        delay(100);
        Serial2.write(0x1A);
        delay(100);
 }

//aprinde radiator
   if(lastLine=="pornit"){
        digitalWrite(34,HIGH);
        digitalWrite(26,HIGH);
        while(Serial2.available()) Serial.write(Serial2.read());
        while(Serial.available()) Serial2.write(Serial.read());
        Serial2.print("\r");
        delay(100);
        Serial2.print("AT+CSCS=\"GSM\"\r");
        delay(100);
        Serial2.print("AT+CMGF=1\r");
        delay(100);
        Serial2.print("AT+CMGS=\"+40724914405\"\r");
        delay(100);
        Serial2.print("Radiator aprins");
        delay(100);
        Serial2.write(0x1A);
        delay(100);
```

```
    }else if(lastLine=="oprit"){
       digitalWrite(34,LOW);
       digitalWrite(26,LOW);
       while(Serial2.available()) Serial.write(Serial2.read());
       while(Serial.available()) Serial2.write(Serial.read());
       Serial2.print("\r");
       delay(100);
       Serial2.print("AT+CSCS=\"GSM\"\r");
       delay(100);
       Serial2.print("AT+CMGF=1\r");
       delay(100);
       Serial2.print("AT+CMGS=\"+40724914405\"\r");
       delay(100);
       Serial2.print("Radiator stins");
       delay(100);
       Serial2.write(0x1A);
       delay(100);
  }

 //trimitere semnal
 if(lastLine=="temperatura"){
     temperatura();
     while(Serial2.available()) Serial.write(Serial2.read());
     while(Serial.available()) Serial2.write(Serial.read());
     Serial2.print("\r");
     delay(100);
     Serial2.print("AT+CSCS=\"GSM\"\r");
     delay(100);
     Serial2.print("AT+CMGF=1\r");
     delay(100);
     Serial2.print("AT+CMGS=\"+40724914405\"\r");
     delay(100);
     Serial2.print("t = ");
     delay(100);
     Serial2.print(reztemp);
     delay(100);
     Serial2.print("  grade celsius");
     delay(100);
     Serial2.write(0x1A);
     delay(100);

  }

//Citiți conținutul mesajului și setați starea în funcție de conținutul mesajelor SMS
  if(lastLine.indexOf("LED ON") >= 0){
   ledStatus = 1;
  }else if(lastLine.indexOf("LED OFF") >= 0) {
    ledStatus = 0;
  }
nextLineIsMessage = false;
```

```
          }
        }

       // Ștergeți matricea de caractere pentru următoarea linie de citire
        for( int i = 0; i < sizeof(currentLine);  ++i ) {
         currentLine[i] = (char)0;
        }
        currentLineIndex = 0;
    }else{
        currentLine[currentLineIndex++] = lastCharRead;
    }
  }
}

//functie use
void contact_use(){
  cu=digitalRead(32);
  if(cu==0){
    digitalWrite(30,HIGH);
    digitalWrite(28,LOW);
  }else if(cu==1){
    digitalWrite(28,HIGH);
    digitalWrite(30,LOW);
  }
}

//functie temperatura
void temperatura(){
  int sumaTemperatura=0;
  int temperatureCelsius=0;
  for (int i =0; i<count; i++) {
    int reading=analogRead(A0);
     float voltage =reading*5.0;
     voltage/=1024.0;
     temperatureCelsius=(voltage-0.5)*53;
     sumaTemperatura = sumaTemperatura + temperatureCelsius;
     reztemp=sumaTemperatura/count;
  }
}

//functie tastatura
void tastatura(){

 char key = keypad.getKey();
  while(key=='0'){
      analogWrite(MOTOR1_PIN1,110);
     analogWrite(MOTOR1_PIN2,0);
     delay(300);
     key='3';
  }
```

```
    analogWrite(MOTOR1_PIN1,0);
    analogWrite(MOTOR1_PIN2,0);
  if(key){
   Serial.println(key);
   position1++;
  }
  if (key){
    delay(100);
  }
  if (key == '*' || key == '#'){
    position = 0;
    position1=0;
   }
  if (key == secretCode[position]){
    Serial.print(position);
    position ++;
  }
  if (position == 4&&position1==4){
    analogWrite(MOTOR1_PIN1,0);
    analogWrite(MOTOR1_PIN2,110);
    delay(300);
  }
  delay(100);
}
```

5. CONCLUZII

Posibilitățile de extindere a studiului început în cadrul acestei teme se poate baza pe: creșterea numărului de senzori; optimizarea funcționării senzorilor; utilizarea temei în domeniul auto; utilizarea temei în domeniul medicinii.
În realizarea proiectului s-au utilizate materiale și dispozitive care asigură un consum redus de energie electrică, costuri minime și fără agresarea mediului înconjurător.

BIBLIOGRAFIE

1. Barna E., Barna V., Cucu C., Miron, C., Mecanică fizică și acustică (II), Editura Universității Bucureşti, 2010.
2. Alexandrescu L, Acustică aplicată, Editura Orator, Braşov 2004.
3. Alexandru M., Sisteme de măsurare cu traductoare, Editura Matrixrom, 2012.
4. Barlea N-M, Fizica senzorilor, Editura Albastra, Cluj Napoca, 2000.
5. Gaidoș N., Pițigoi A., Sisteme de comandă de la distanță prin GSM, SINUC 2017, UTCB, Bucuresti.