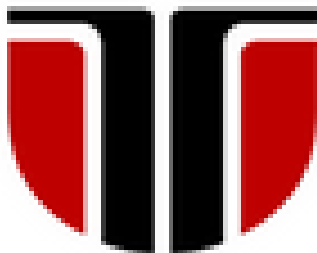


Universitatea Tehnica din Cluj-Napoca



Catedra de Calculatoare

Aplicatie Bluetooth pentru dezvoltarea unui sistem de monitorizare “SmartHome”

Studenti:

Irimus Ileana Maria

Voicu Diana-Georgiana

Indrumator de proiect:

Itu Razvan

Grupa 30233

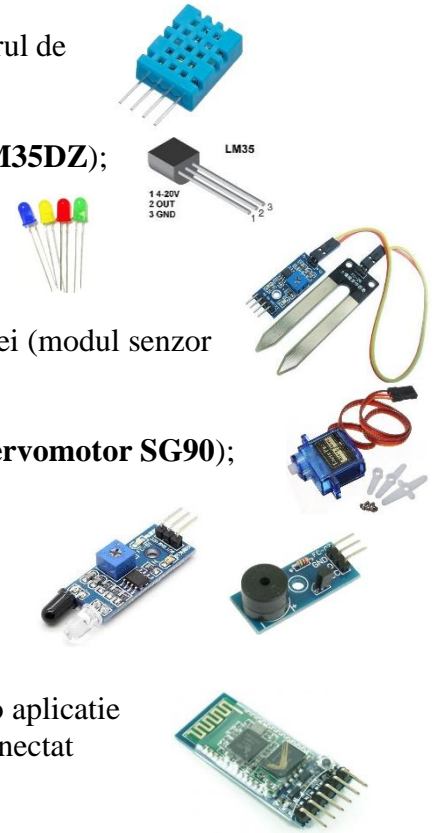
Cuprins

1. Descriere proiect	3
2. Proiectare si implementare	4
3. Schema circuit.....	8
4. Poze rezultate experimentale.....	9
Bibliografie	11

1. Descriere proiect

Pentru acest proiect am dorit sa realizam o casa inteligenta. Astfel, cu ajutorul mai multor senzori am reusit sa simulam anumite functionalitati specifice unui smart home:

- ✓ masurarea temperaturii interioare si a umiditatii aerului (senzorul de temperatura si umiditate digital **DHT11**);
- ✓ masurarea temperaturii exterioare (senzorul de temperatura **LM35DZ**);
- ✓ simularea aprinderii/stingerii becurilor din diferitele incaperi;
- ✓ masurarea nivelului de umiditate sol al florilor de pe terasa casei (modul senzor umiditate sol – **higrometru**);
- ✓ simularea inchiderii/deschiderii unui geam (cu ajutorul unui **Servomotor SG90**);
- ✓ sistem de alerta care va notifica printr-un semnal audio de fiecare data cand cineva intra/iese din casa (**modul senzor IR infrarosu** - evita obstacole impreuna cu un **modul buzzer**).



Pentru a controla toate aceste aspecte ale casei inteligente am utilizat o aplicatie Bluetooth pe telefon (<http://ai2.appinventor.mit.edu/>) la care ne-am conectat folosind un modul **Bluetooth HC-05**.

2. Proiectare si implementare

Codul implementat poate fi structurat in 3 parti:

- Includerea bibliotecilor si declararea variabilelor si a pinilor necesari;
- Functia setup in care sunt setati pinii INPUT si OUTPUT;
- Functia loop cu o structura logica IF folosita pentru a comunica cu modulul Bluetooth si setarea valorilor pentru leduri, senzorul IR si buzzer in functie de variabilele declarate anterior.

Includerea bibliotecilor si declararea variabilelor si a pinilor necesari

Pentru acest proiect a fost nevoie de includerea a trei biblioteci, astfel: <SoftwareSerial.h> pentru modulul Bluetooth, <Servo.h> pentru servomotor si <dht.h> pentru senzorul de temperatura si umiditate DHT11.

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <dht.h>
```

Pentru inceput, a fost conectat modulul Bluetooth la pinii 53 si 52 corespunzatori lui TX si RX, iar unghiul de rotatie al elicelor servomotorului este initial 0. Fiecare led este conectat la pinul corespunzator (pin 4 = LED_hol , pin 5 = LED_baie, pin 6 = LED_terasa, pin 7 = LED_dormitor), iar variabilele de tip int responsabile cu starea ledurilor sunt initializate la 0 (ledurile sunt stinse). Sensorii utilizati sunt conectati la pinii placutei Arduino Mega 2560, iar variabilele ce vor primi valorile citite de pe acestia sunt initializate la 0 (Senzorul de temperatura LM35 = pinul A0, senzorul de temperatura si umiditate DHT11 = pinul 8, senzorul de umiditate sol = pinul A1, senzorul IR = pinul 9, buzzer = pinul 10). Datele primite de modulul Bluetooth de la aplicatia mobila vor fi stocate intr-o variabila de tip String numita dataIn, initializata cu un string gol.

Funcția setup

Primul pas reprezintă setarea baud rate-ului implicit pentru modulul Bluetooth pe care îl vom utiliza pentru conectarea la aplicația mobilă.

```
void setup()
{
    // Default baud rate of the Bluetooth module
    Bluetooth.begin(9600);
```

Apoi, urmează setarea pinilor pentru ledurile casei inteligente, a senzorului cu infraroșu și a buzzer-ului prin funcția predefinită pinMode.

```
// Declare the LEDs as outputs
pinMode(LED_hol, OUTPUT);
pinMode(LED_terasa, OUTPUT);
pinMode(LED_dormitor, OUTPUT);
pinMode(LED_baie, OUTPUT);

// Set the IR sensor as input.
pinMode(IR_pin, INPUT);
// Set the buzzer as output.
pinMode(buzzer_pin, OUTPUT);
```

În cazul servomotrului am utilizat două funcții specifice: attach pentru conectarea la pinul corespunzător (11) și write pentru setarea unghiului de rotație al elicelor.

```
servo.attach(11);
servo.write(angle);
```

Functia loop

Functia void loop este si cea mai consistenta parte a programului. Astfel, incepem prin a verifica daca s-a realizat conexiunea dintre aplicatia mobila si modulul Bluetooth. In caz afirmativ, dataIn primeste stringul apasat in aplicatie, asteapta 20ns pentru a ne asigura ca citirea datelor este completa, dupa care se verifica ce buton a fost apasat: in cazul ledurilor modificam valoarea variabilelor is<Incapere>On (pentru aprindere/stingere).

```
// If the Bluetooth connection is available
if (Bluetooth.available() > 0)
{
    // Read the string received from the Bluetooth in <dataIn>
    dataIn = Bluetooth.readString();
    delay(20);

    //----- LEDS

    if (dataIn.equals("HOL"))
    {
        isHolOn = 1;
    }
    else if (dataIn.equals("HOL_OFF"))
    {
        isHolOn = 0;
    }
    else if (dataIn.equals("TERASA"))
    {
        isTerasaOn = 1;
    }
    else if (dataIn.equals("TERASA_OFF"))
    {
        isTerasaOn = 0;
    }

    else if (dataIn.equals("DORMITOR"))
    {
        isDormitorOn = 1;
    }
    else if (dataIn.equals("DORMITOR_OFF"))
    {
        isDormitorOn = 0;
    }
    else if (dataIn.equals("BAIE"))
    {
        isBaieOn = 1;
    }
    else if (dataIn.equals("BAIE_OFF"))
    {
        isBaieOn = 0;
    }
}
```

In cazul citiri temperaturii exterioare se executa 10 citiri pentru a obtine un rezultat cat mai exact. Pentru calculul temperaturii am folosit o formula de conversie a datelor citite in volti si retinand informatia in valoarea medie si am convertit-o apoi in milivolti. La final valoarea a fost transmisa modului Bluetooth si tiparita in aplicatia mobila.

```
//----- TEMP EXT

if (dataIn.equals("TEMP"))
{
    float v_med = 0;
    float v_final = 0;
    for (int j = 0; j < 10; j++)
    {
        // Read the analog value.This voltage is stored as a 10bit number
        int reading = analogRead(temp_pin);
        // Converting that reading to voltage
        float voltage = reading * 5.0;
        voltage /= 1024.0;
        v_med = v_med + voltage;
        delay(200);
    }

    // Divide it by 10 beacuse each degree rise results in a 10 millivolt increase
    v_final = v_med / 10;
    // Multiplied by 1000 to convert it to millivolt
    temperatura = v_final * 100 ;
    // Send the value of the temperature via Bluetooth
    Bluetooth.println(temperatura);
}
```

Pentru citirea temperaturii interioare si a umiditatii aerului si mai apoi pentru afisarea acestora inapoi in aplicatie, se utilizeaza functiile specifice bibliotecii <dht.h> si anume: read11 si temperature/humidity.

```
//----- TEMP INT //----- UMID AER
if (dataIn.equals("TEMP_INT"))
{
    dht_reading = DHT.read11(dht_pin);
    // Send the value of the temperature via Bluetooth
    Bluetooth.println(DHT.temperature);
    delay(200);
}

if (dataIn.equals("AER"))
{
    dht_reading = DHT.read11(dht_pin);
    // Send the value of the humidity via Bluetooth
    Bluetooth.println(DHT.humidity);
    delay(200);
}
```

Pentru aflarea umiditatii solului, se citesc datele primite prin functia analogRead si se transforma utilizand functia map, astfel: noi avem nevoie sa scoatem din valoarea citita procentajul in functie de range-ul maxim al senzorului. Stiind ca formula functiei map este: $(x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min$, antetul functiei fiind `map(long x, long in_min, long in_max, long out_min, long out_max)`, realizam ca trebuie sa ajungem la o formula de procentaj => inmultire cu 100 si impartire cu 1023 (ca in cazul senzorului de temperatura). Dar senzorul are un range de valori intre 0-1023 => din 1023 trebuie scazuta valoarea citita pentru a ajunge la valoarea dorita ce va fi transformata in procente. Acestea din urma vor fi tiparite pe ecranul telefonului in aplicatie.

```
//----- UMID SOL

if (dataIn.equals("SOL"))
{
    // Read the analog value
    int reading = analogRead(umid_sol_pin);
    // Transform the data read in percentage
    umiditate_sol = map(reading, 1023, 0, 0, 100);
    // Send the value of the humidity via Bluetooth
    Bluetooth.println(umiditate_sol);
    delay(200);
}
```

Inchiderea si deschiderea unuia dintre geamurile casutei inteligente sunt realizate prin rotirea elicelor servomotrului cu un unghi de 90 de grade, care se face treptat.

```
//----- GEAM

if (dataIn.equals("GEAM_CLOSE"))
{
    for(angle = 0; angle < 90; angle++)
    {
        servo.write(angle);
        delay(15);
    }
}

delay(2000);

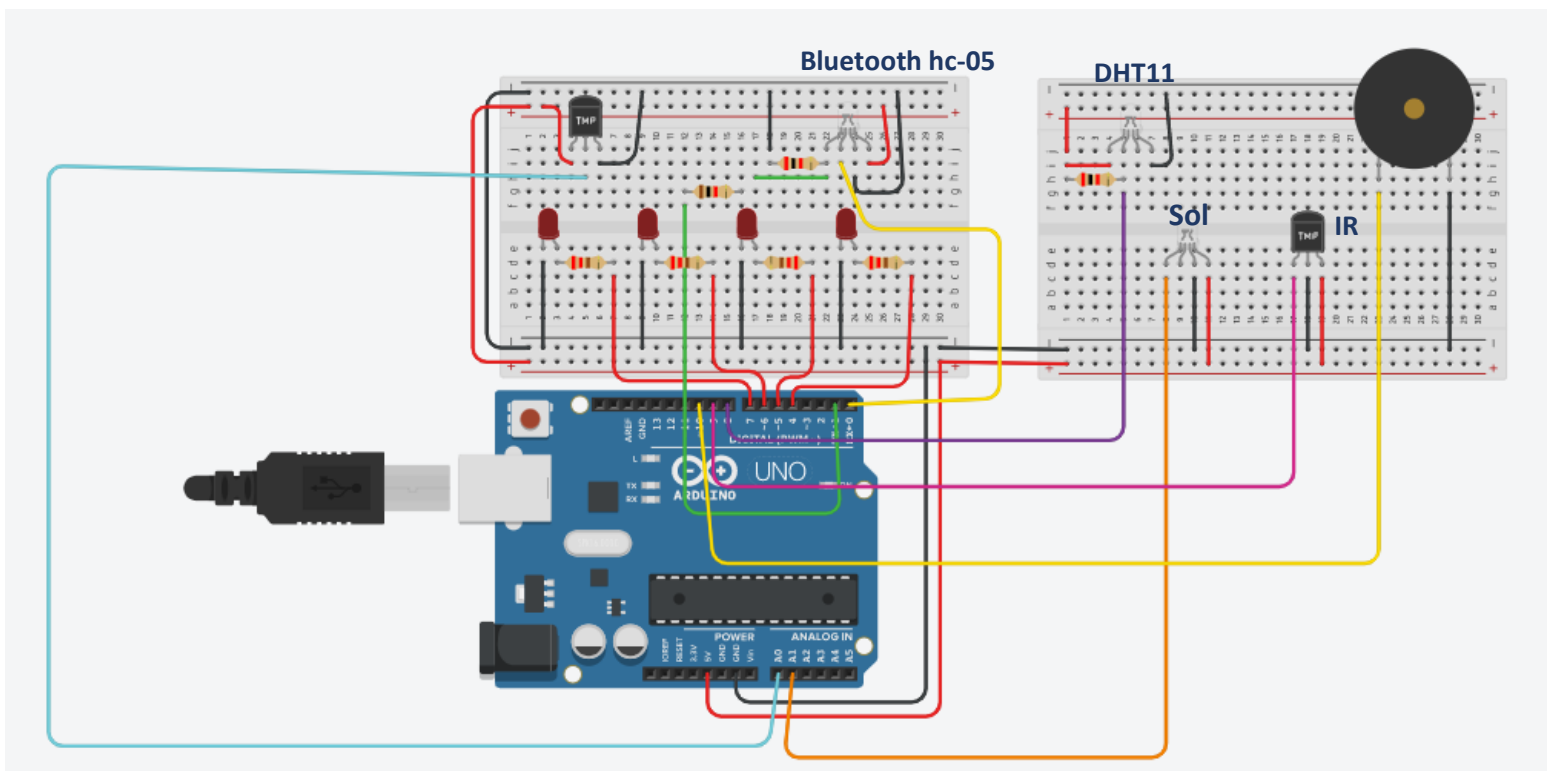
if (dataIn.equals("GEAM"))
{
    for(angle = 90; angle > 0; angle--)
    {
        servo.write(angle);
        delay(15);
    }
}
```

Pe langa datele citite de la senzori si tiparite in aplicatia Bluetooth, pe macheta se vor aprinde/stinge cele 4 leduri. In ceea ce priveste sistemul de alerta al casei, in momentul in care senzorul IR detecteaza miscare, se va citit valoarea logica HIGH, buzzerul va fi setat tot pe HIGH si va genera un semnal sonor.

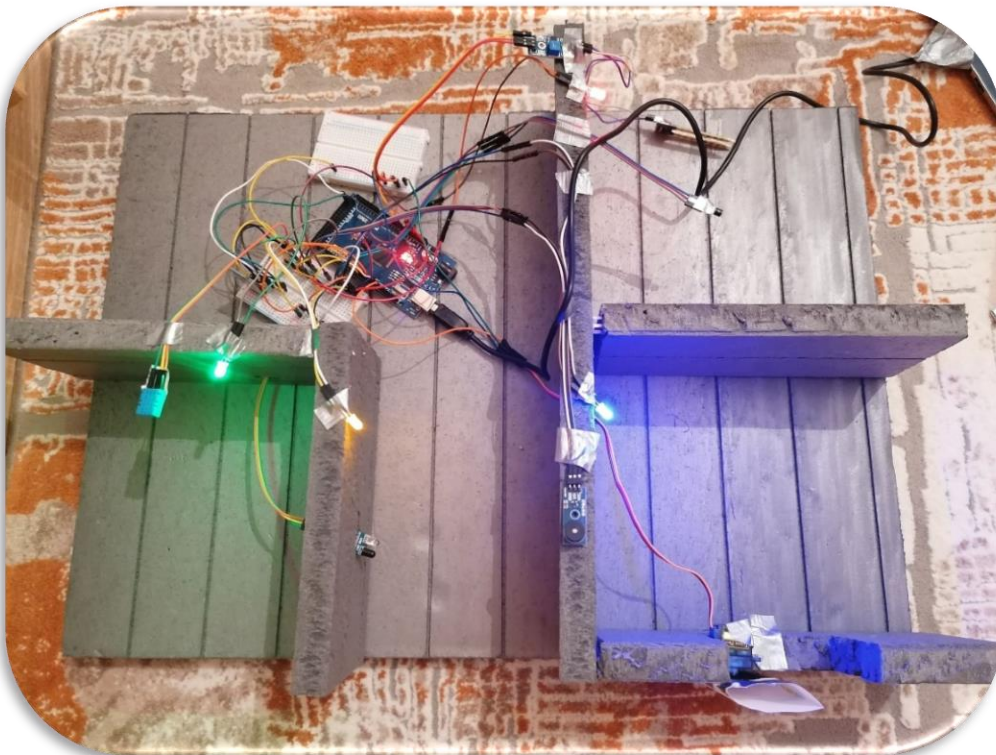
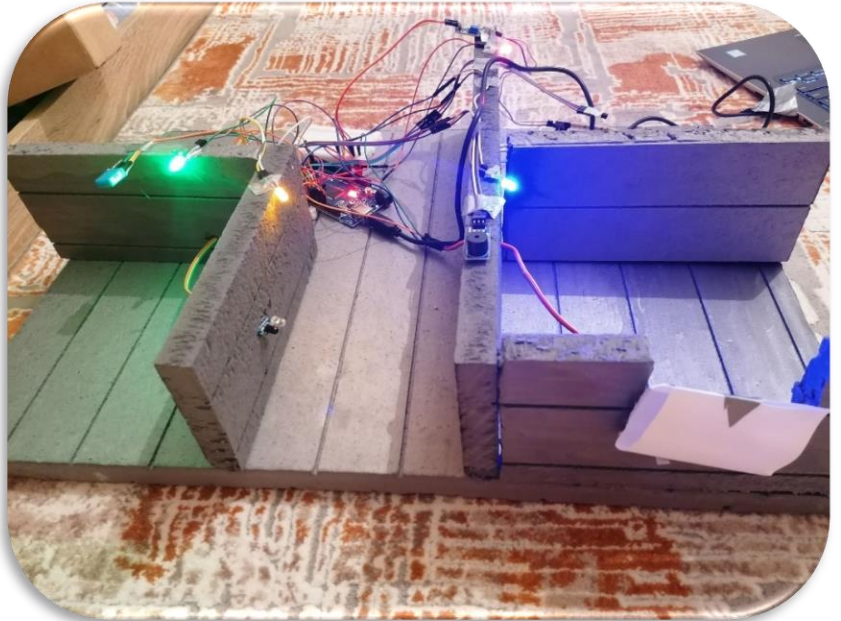
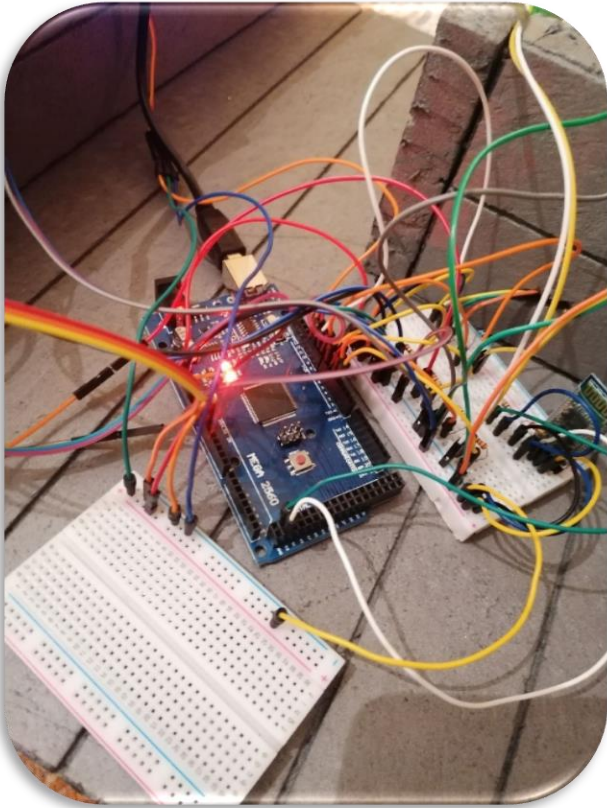
```
//----- hol
if (isHolOn == 1)
{
    digitalWrite(LED_hol, HIGH);
}
else
{
    digitalWrite(LED_hol, LOW);
}

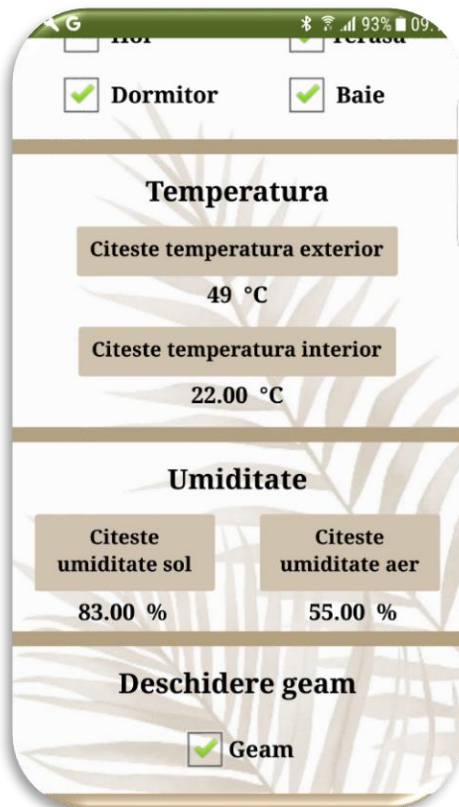
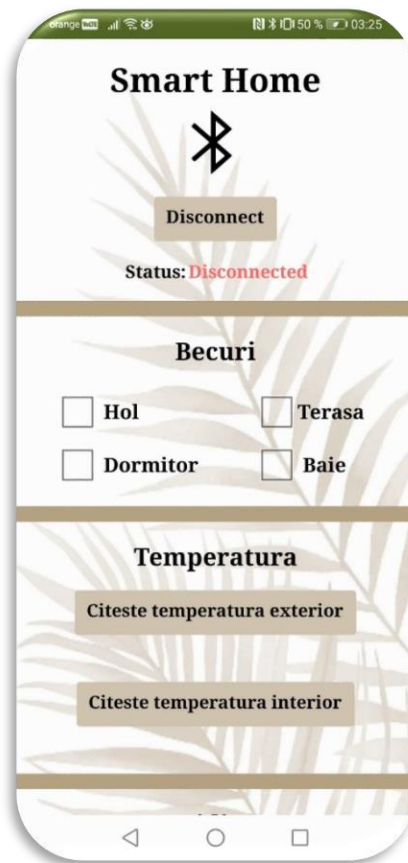
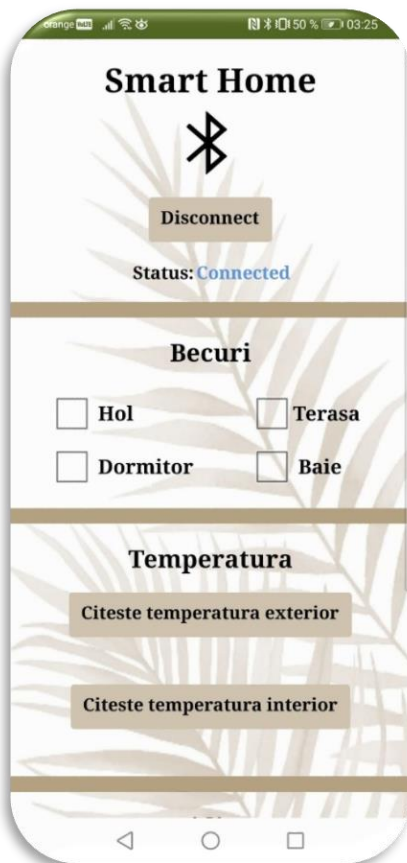
//----- IR
if (digitalRead(IR_pin) == LOW)
{
    digitalWrite(buzzer_pin, LOW);
}
if (digitalRead(IR_pin) == HIGH)
{
    digitalWrite(buzzer_pin, HIGH);
}
```

3. Schema circuit



4. Poze rezultate experimentale





Bibliografie

[1] HC-05 Bluetooth Module with Arduino-MIT App Inventor:

<https://www.youtube.com/watch?v=aQcJ4uHdQEA>

[2] Senzor temperatura si umiditate digital DHT11:

<https://www.youtube.com/watch?v=OogldLc9uYc>

[3] Senzor temperatura LM35DZ:

<https://www.youtube.com/watch?v=3Xc2sPhwWEc>

<https://www.instructables.com/Measure-Temperature-Using-Lm35-and-Arduino/>

[4] Modul sensor umiditate sol – higrometru :

https://ardushop.ro/ro/home/44-modul-senzor-umiditate-sol-higrometru.html?search_query=Modul+senzor+umiditate+sol+-+higrometru&results=433

[5] Servomotor SG90:

<https://ardushop.ro/ro/electronica/93-servomotor-sg90.html>

[6] Modul Buzzer:

<https://ardushop.ro/ro/home/89-modul-buzzer.html>

[7] Modul sensor IR infrarosu – Evita obstacolele:

<https://ardushop.ro/ro/home/41-modul-senzor-ir-infrarosu-evita-obstacole.html>