

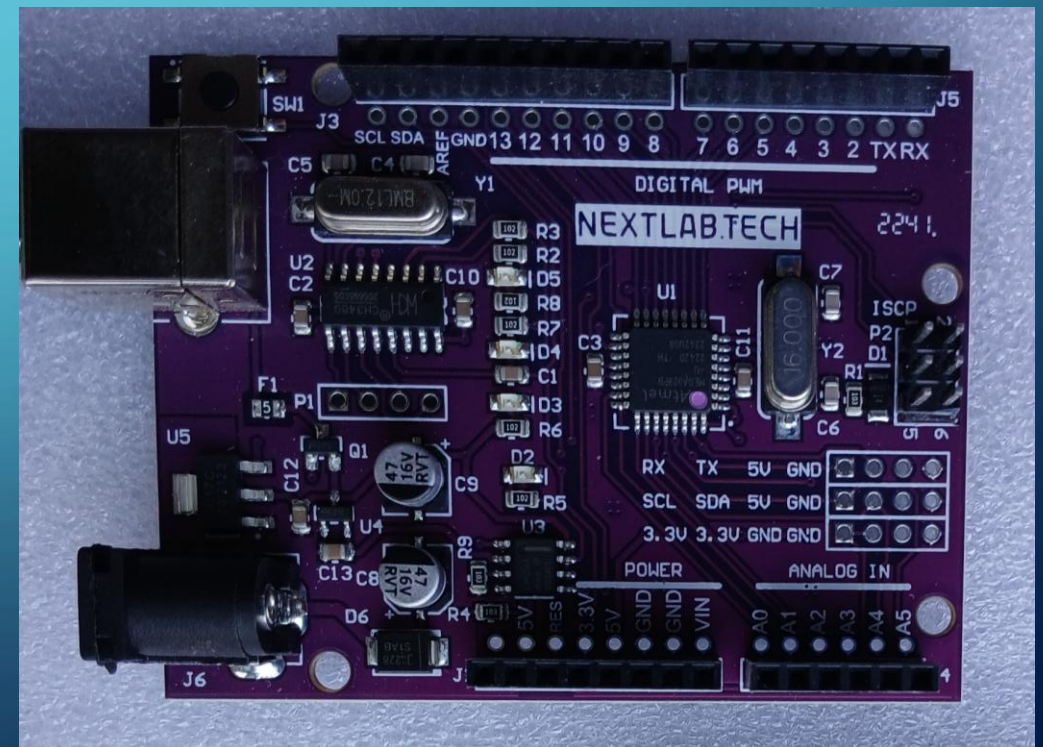
A decorative graphic on the left side of the slide, consisting of white and light blue lines and circles that resemble a circuit board or a network diagram. The lines are vertical and horizontal, with some diagonal connections, and the circles are small and white, some with a light blue outline.

INTRODUCERE ÎN UTILIZAREA PLĂCILOR PROGRAMABILE ARDUINO

NOȚIUNI SISTEME ROBOTIZATE

STRUCTURA UNEI PLĂCI PROGRAMABILE

- O placă programabilă este un mic „calculator” care poate controla dispozitive electronice (LED-uri, senzori, motoare).
- Arduino este o platformă simplă și populară pentru începători, folosită pentru proiecte interactive.
- Microcontroller – „creierul” plăcii
- Pini digitali (0–13) – pentru ON/OFF, pot primi sau furniza curent (valoarea 1) sau nu (valoarea 0)
- Pini analogici (A0–A5) – pentru citirea valorilor (ex: temperatură), care pot primi sau furniza curent cu valori variabile (între 0 și 5 V)
- Port USB – conectare la calculator
- Alimentare – 5V sau GND (ground/pământare), 0 V
- LED integrat (pin 13)



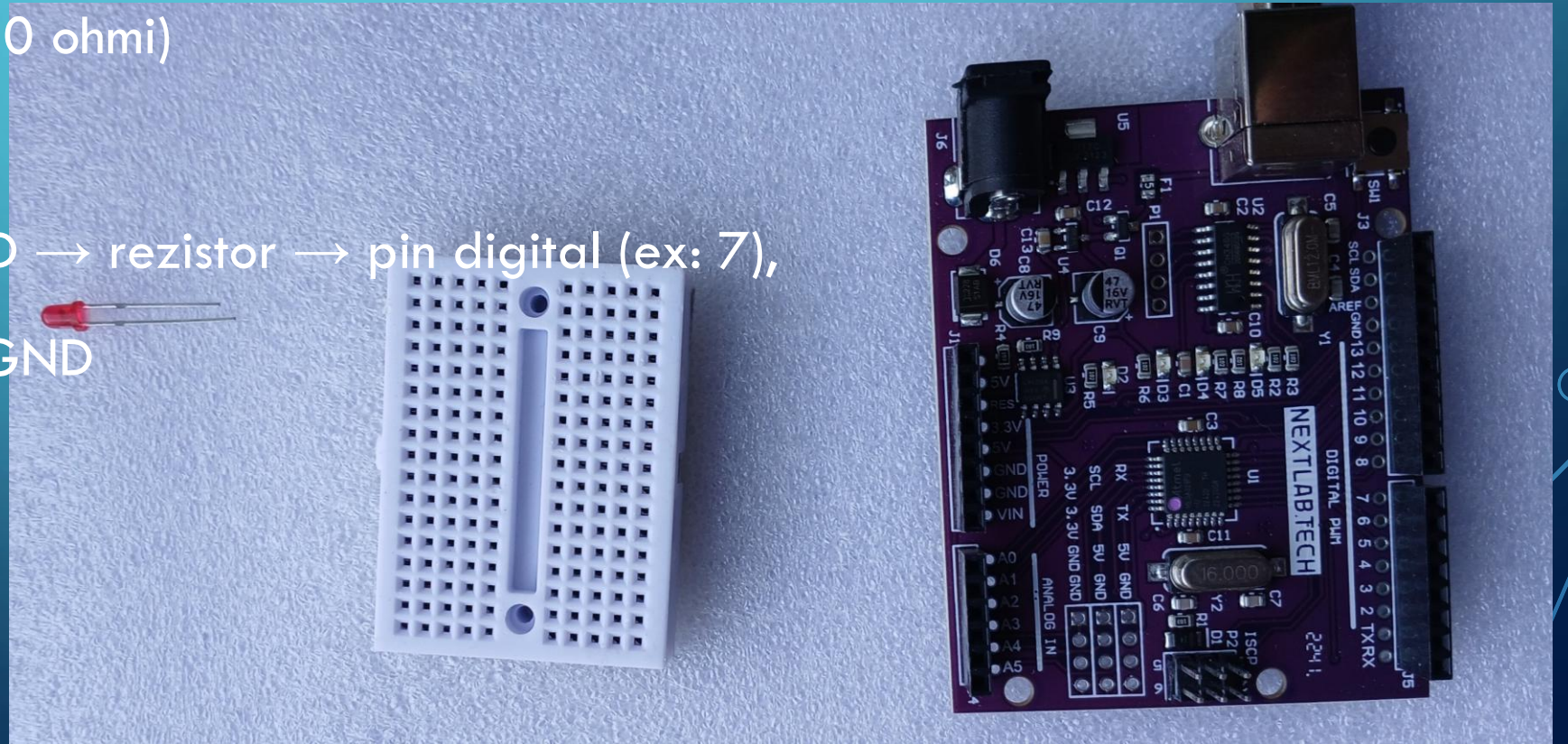
PLACA BREADBOARD ȘI CONEXIUNI

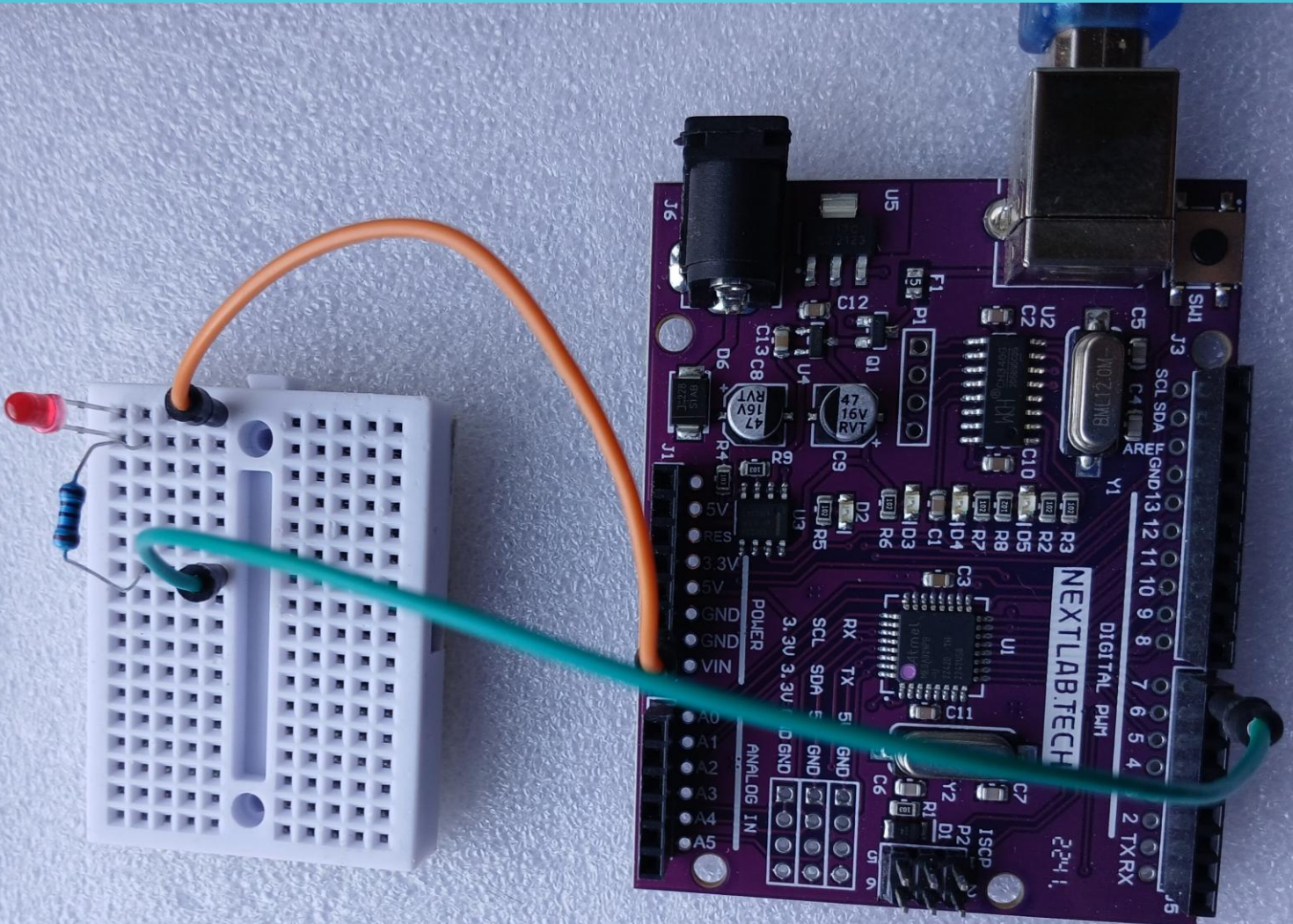
Placa Breadboard este o placă pe care putem conecta componente fără lipire.

Conectarea unui LED:

- LED
- Rezistor (220–330 ohmi)
- Conectori

Pași: Picior lung LED → rezistor → pin digital (ex: 7),
iar Picior scurt → GND





PROGRAMAREA PLĂCII ARDUINO

- Mediul de lucru se numește Arduino IDE
- Structura unui program, folosind limbajul C++:

```
void setup() {  
    // se execută o singură dată  
}
```

```
void loop() {  
    // se repetă continuu  
}
```

Comenzi de bază:

- `pinMode(pin, OUTPUT);`
- `digitalWrite(pin, HIGH/LOW);`
- `delay(ms);`

```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }  
10
```

EXEMPLUL 1 - UN LED CARE SE STINGE ȘI SE APRINDE LA UN INTERVAL CONTROLAT (400 MS):

Conectarea:

LED (+) → rezistor → pin 7

LED (-) → GND

Cod C++:

```
void setup() {  
    pinMode(7, OUTPUT);           //configurăm pin-ul 7 să trimită curent  
}  
  
void loop() {  
    digitalWrite(7, HIGH);       //LED-ul se aprinde  
    delay(400);                  //așteptăm 400 ms  
    digitalWrite(7, LOW);       //LED-ul se stinge  
    delay(400);                  //așteptăm 400 ms, apoi se reia totul  
}
```

EXEMPLUL 2 - UN LED CARE SE STINGE ȘI SE APRINDE LA UN INTERVAL VARIABIL

Conectarea:

LED (+) → rezistor → pin 7

LED (-) → GND

Cod C++:

```
int timp = 600; //definim o variabilă, timp, care va avea valoarea 600
void setup() { //configurăm pin-ul 7 să trimită curent
    pinMode(7, OUTPUT);
}
void loop() {
    digitalWrite(7, HIGH); //LED-ul se aprinde
    delay(timp); //așteptăm un interval de timp
    digitalWrite(7, LOW); //LED-ul se stinge
    timp = timp - 50; //valoarea timpului scade cu 50
    if (timp < 100) { //verificăm dacă nu este prea mic timpul
        timp = 600; //reinițializăm timpul la 600
    }
    delay(timp); //așteptăm un interval de timp, apoi se reia totul
}
```

EXEMPLUL 3 - DOUĂ LED-URI CARE CLIPESC ALTERNATIV:

Conectarea:

LED1 (+) → rezistor → pin 7

LED1 (-) → GND

LED2 (+) → rezistor → pin 8

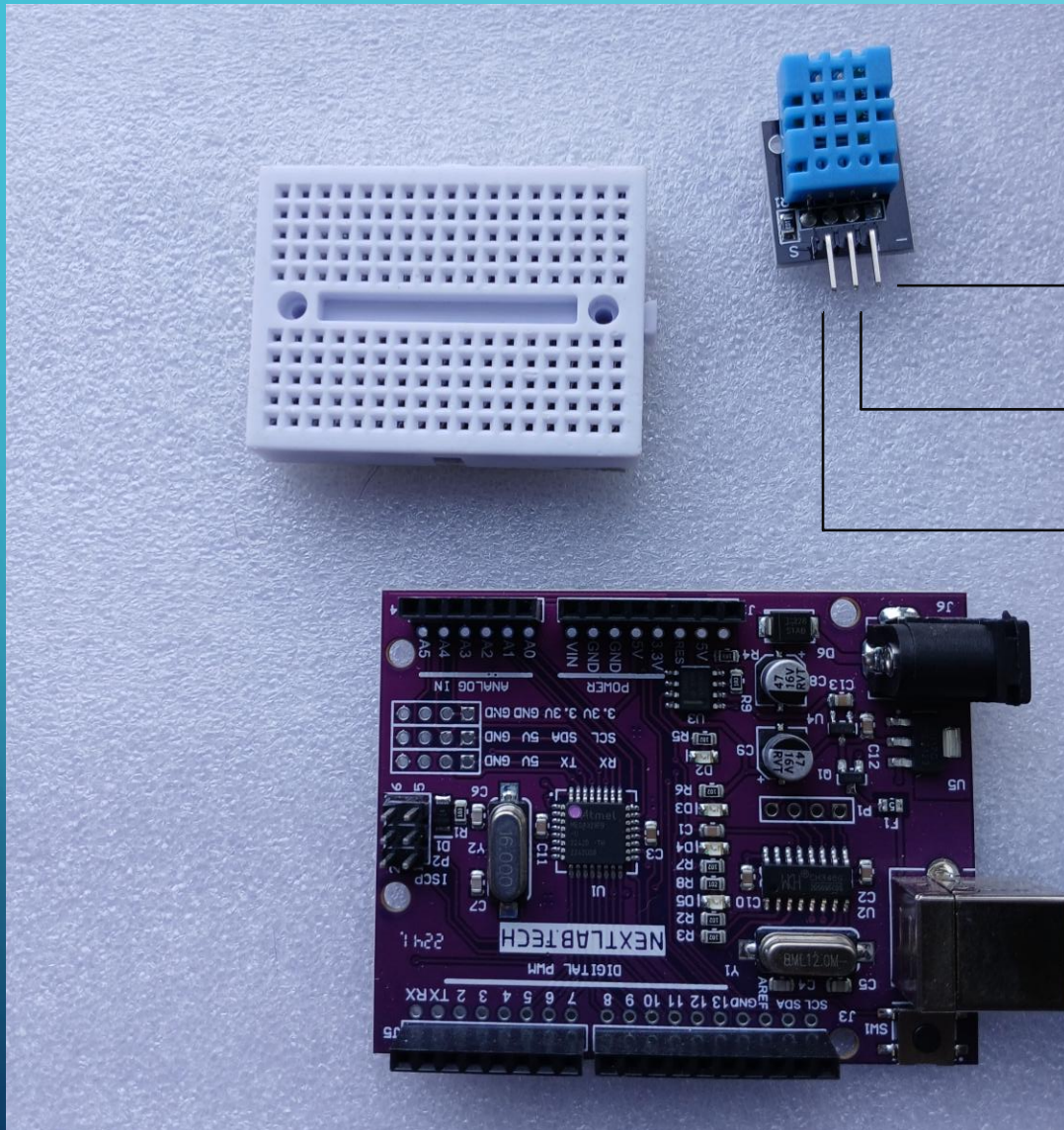
LED2 (-) → GND

Cod C++:

```
void setup() {  
  pinMode(7, OUTPUT);           //configurăm pin-ul 7 să trimită curent  
  pinMode(8, OUTPUT);           //configurăm pin-ul 8 să trimită curent  
}
```

```
void loop() {  
  digitalWrite(7, HIGH);         //LED1 se aprinde  
  digitalWrite(8, LOW);          //LED2 se stinge  
  delay(400);  
  digitalWrite(7, LOW);          //LED1 se stinge  
  digitalWrite(8, HIGH);         //LED2 se aprinde  
  delay(400);  
}
```

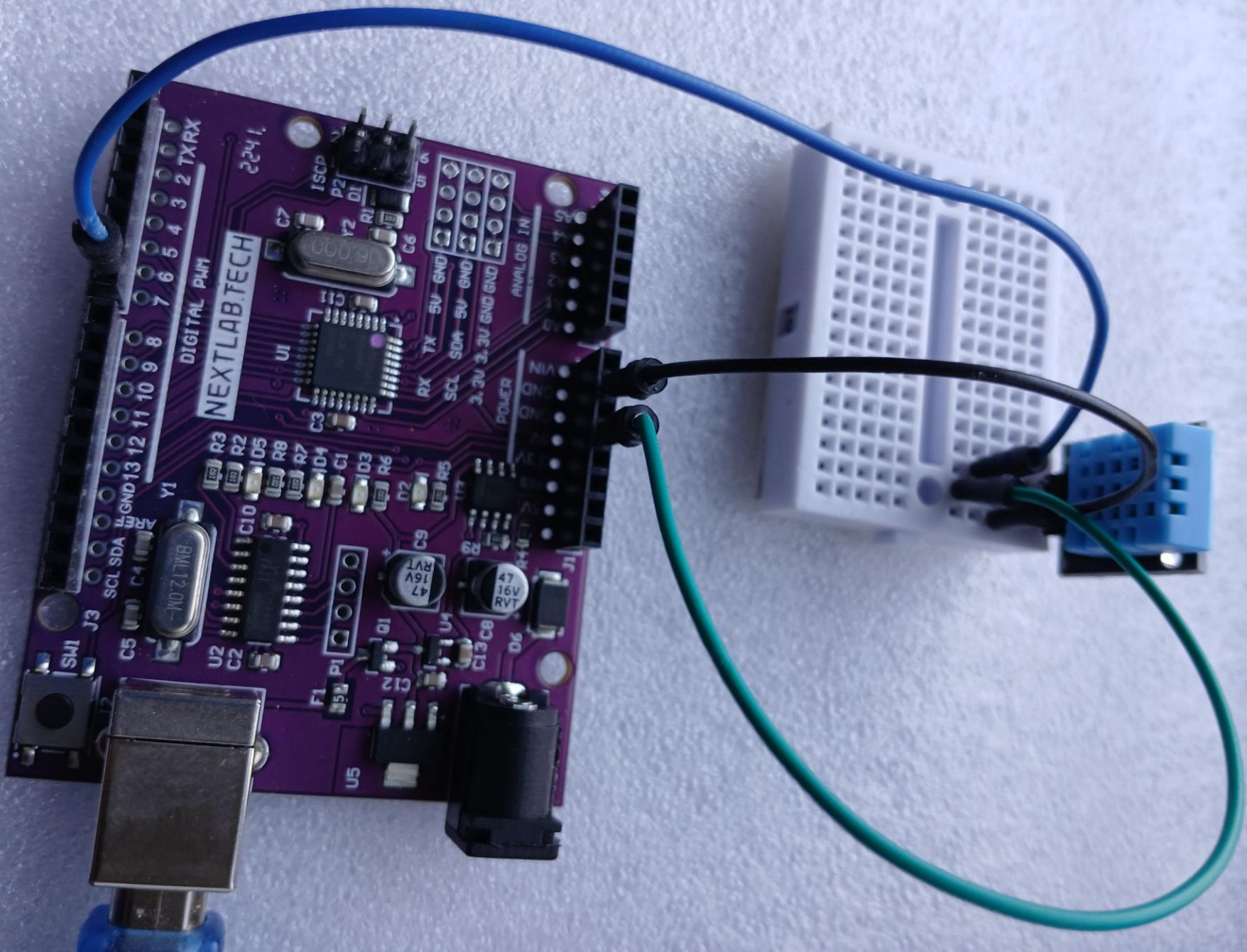
EXEMPLUL 4 – UTILIZAREA UNUI SENZOR DE TEMPERATURĂ



GND – ground

5 V – curent

Semnal – pin digital



Cod C++

```
#include "DHT.h"
```

```
DHT dht(5, DHT11);
```

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
  pinMode(7, OUTPUT);  
}
```

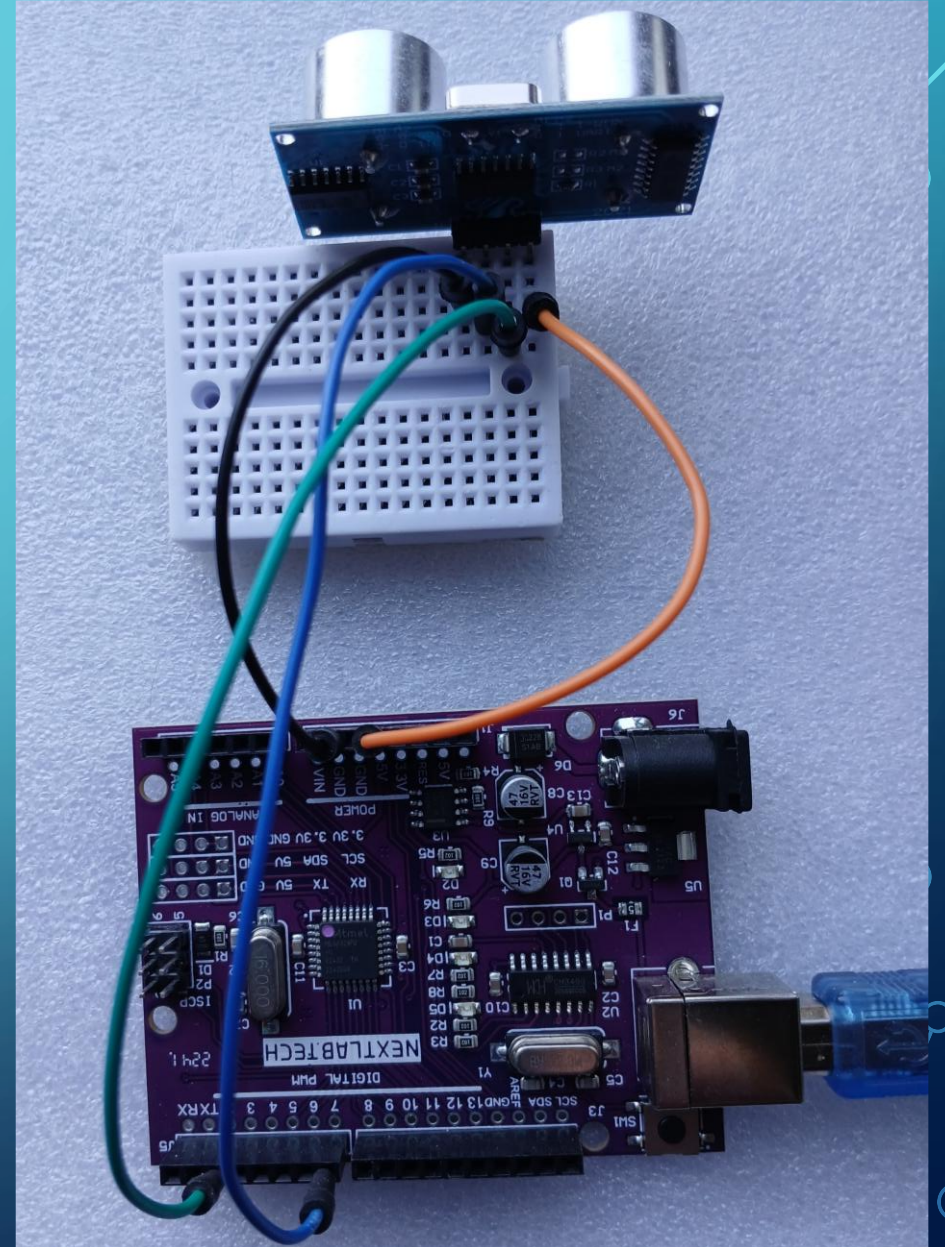
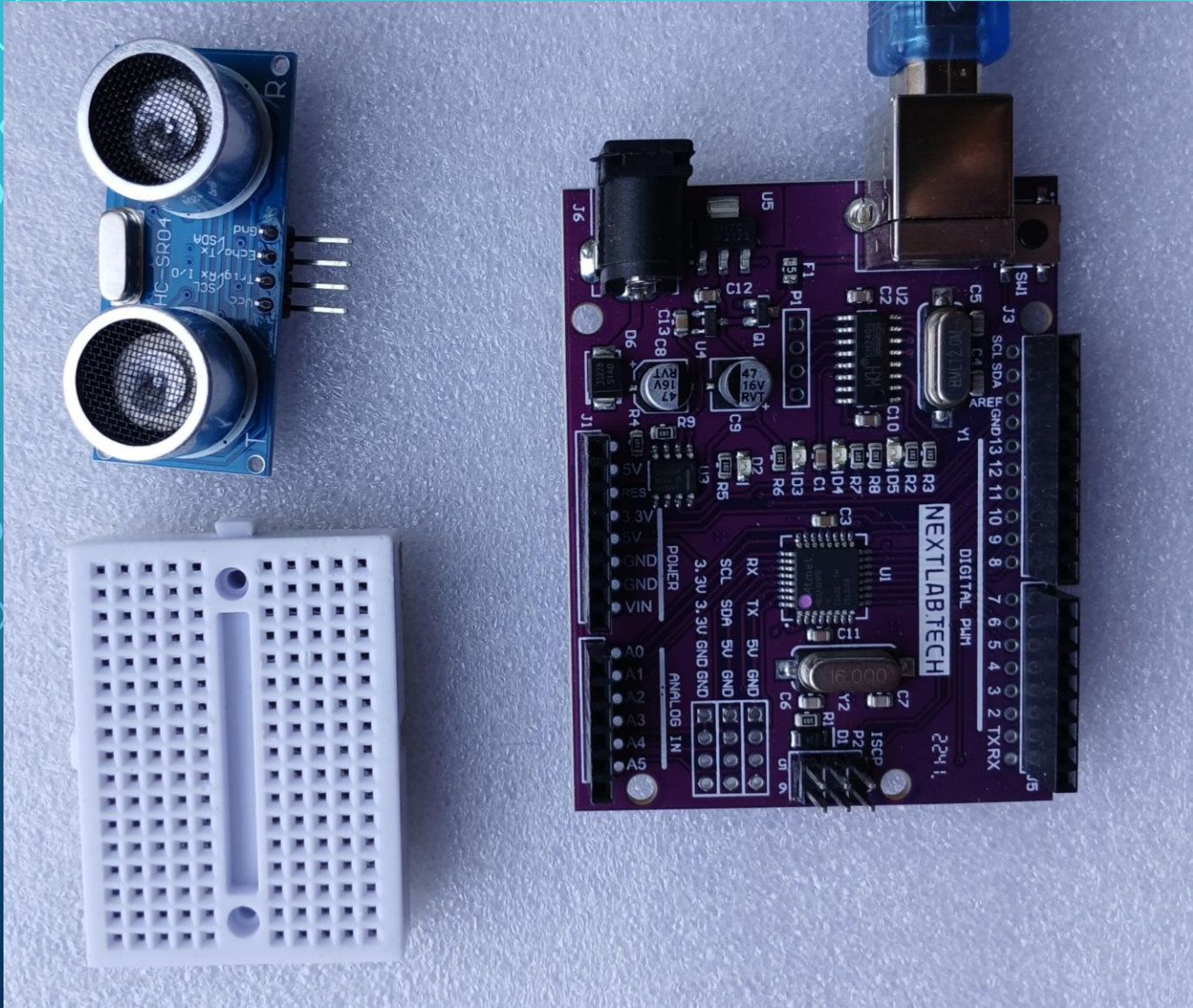
```
void loop() {  
  delay(2000);
```

```
  float umiditate = dht.readHumidity();  
  float temperatura = dht.readTemperature();
```

```
  Serial.print("umiditate: ");  
  Serial.println(umiditate);  
  Serial.print("temperatura: ");  
  Serial.println(temperatura);
```

```
  if (temperatura > 28) { // dacă temperatura  
    digitalWrite(7, HIGH); // depășește 28 grade, se  
                           // aprinde LED-ul  
  }  
  else { // altfel LED-ul se stinge  
    digitalWrite(7, LOW);  
  }  
}
```

EXEMPLUL 5 – UTILIZAREA UNUI SENZOR DE DISTANȚĂ



Cod C++

```
#include <HCSR04.h>

const int TRIGGER_PIN = 2; // pin 1 stanga senzor 5 V
const int ECHO_PIN = 7; // pin 2 stanga senzor
// pin 3 stanga senzor
// pin 4 senzor GND

HCSR04 senzor(TRIGGER_PIN, ECHO_PIN);

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

void loop() {
  int d=senzor.getDistance();
  Serial.print("Distance: ");
  Serial.print(d);
  Serial.println(" cm");
  delay(500);
}
```