



Plan de Lecție - Informatică

Subject: Instrucțiunea alternativă IF – aplicație în programarea unui sistem de securitate ultrasonic

Grup țintă: elevi clasa a IX-a (începători)

Obiective/ Competențe

- Obj1. Explicarea funcționării instrucțiunii alternative IF și a IF-urilor imbricate
- Obj2. Construirea dispozitivului Arduino
- Obj3. Implementarea codului dispozitivului folosind instrucțiunea IF
- Obj4. Testarea funcționalității dispozitivului

Metode didactice: conversația, explicația, problematizarea, algoritmizarea, demonstrația

Mijloace/Instrumente/Tehnologii educaționale

Calculator, Internet, online Arduino editor,
Componente pentru proiect (Arduino Uno x 1, breadboard x 1, senzor ultrasonic x 1, leduri rosu x 1, galben x 1, verde 1 x 1, rezistor x 1, cabluri x 10)

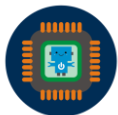
Proiectarea activității

Elevii vor fi împărțiți în 3 echipe care vor realiza în paralel dispozitivul și îl vor prezenta. În fiecare echipă de 10 elevi vor fi elevi care vor construi dispozitivul și elevi care îl vor programa.

Durata	Activitate	Metode/mijloace
5 min	Introducerea instrucțiunii alternative IF. Explicarea sintaxei C++ și a principiului de execuție. IF-uri imbricate, reguli de asociere ELSE.	Explicația, conversația
5 min	Prezentarea dispozitivului de securitate ultrasonic (Pas 1- anexa)	Problematizarea, Explicația, conversația
20 min	Construirea dispozitivului (Pas 2- Pas 6 anexa)	Explicația, conversația, problematizarea
15 min	Programarea dispozitivului	Algoritmizarea,
5 min	Testarea funcționării dispozitivului	Demonstrația

Evaluare/Feedback:

Testarea funcționalității dispozitivului pentru fiecare situație implementată prin instrucțiunile IF (poziționarea la distanțe diferite și verificarea semnalelor luminoase și sonore).



ROBOSTEM Project

Agreement no: 2019-1-RO01-KA202-063965

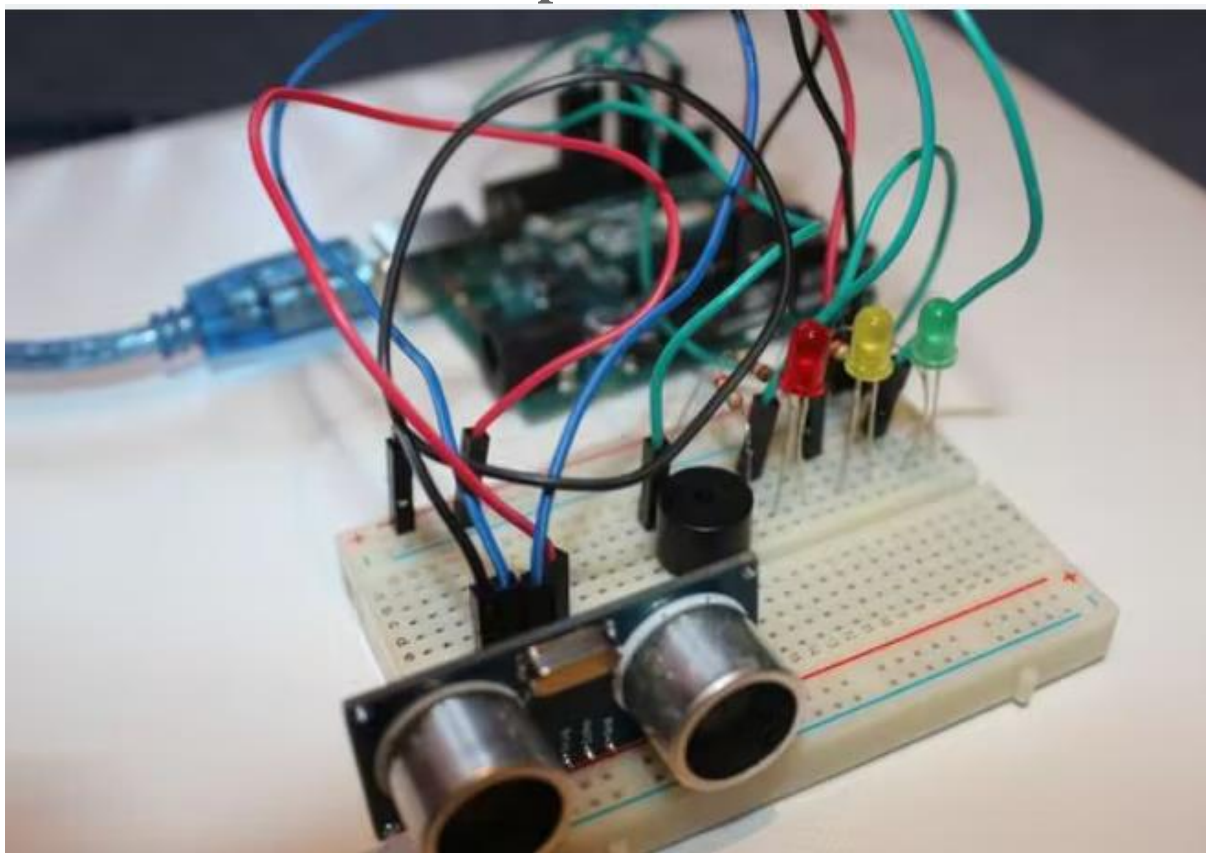


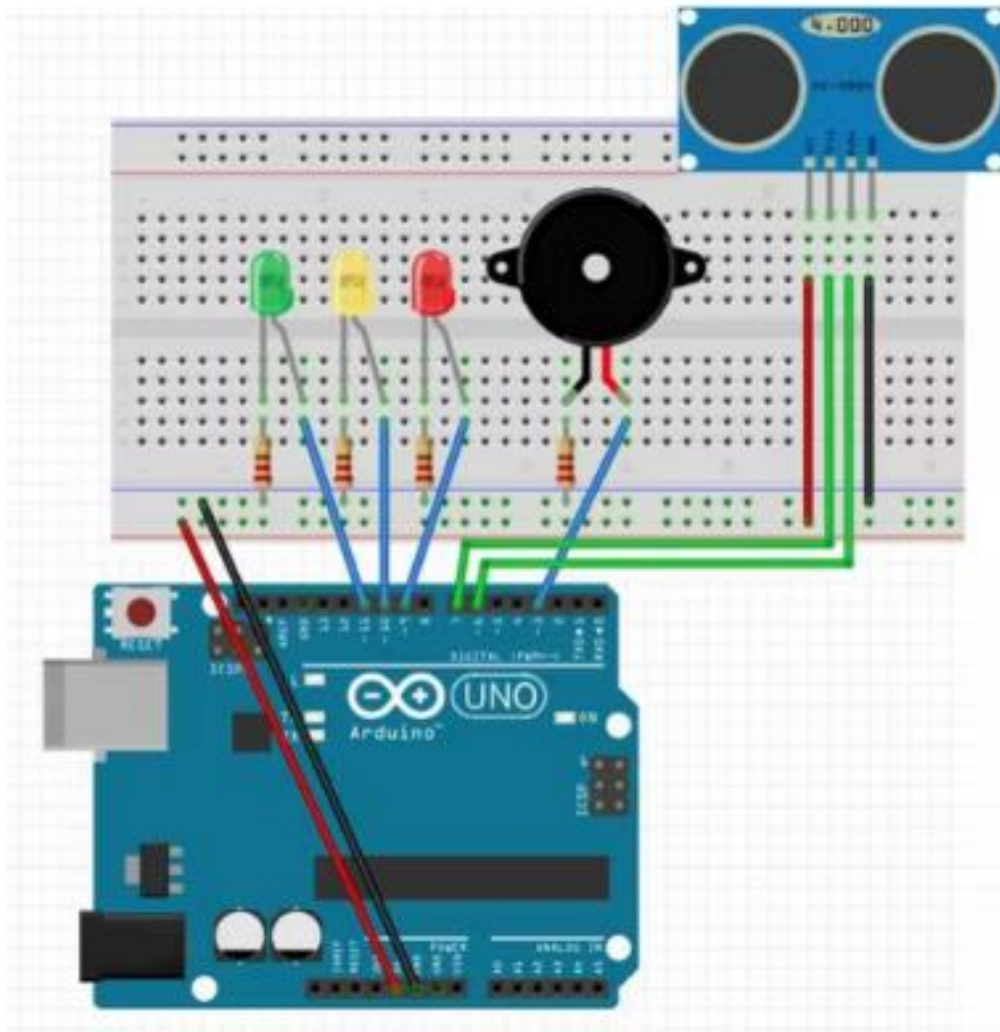
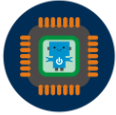
Bibliografie/Webografie:

https://create.arduino.cc/projecthub/Krepak/ultrasonic-security-system-3afe13?ref=tag&ref_id=kids&offset=3

Anexa 1

Pas 1: Prezentarea dispozitivului

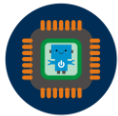




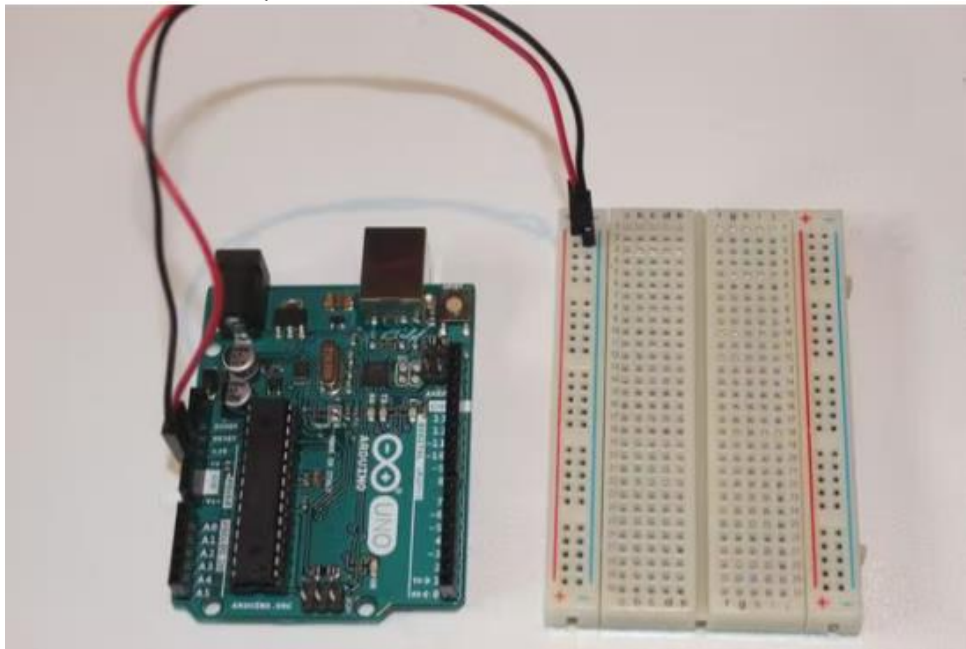
Pas 2: Conetați:

- cablul roșu de la pin 5V la canalul pozitiv de pe breadboard
- cablul negru de la GND pin de pe Arduino la canalul negativ de pe breadboard
- Buzzer = pin 7
- Ultrasonic Sensor:
 - Echo = pin 3
 - Trig = pin 2
- LED uri:
 - RedLED = pin 4
 - YellowLED = pin 5
 - GreenLED = pin 6

Cablurile verzi conectează LED urile în line astfel: LED pozitiv cu LED negativ la canalul negativ de pe breadboard, folosind un 220 ohm rezistor.

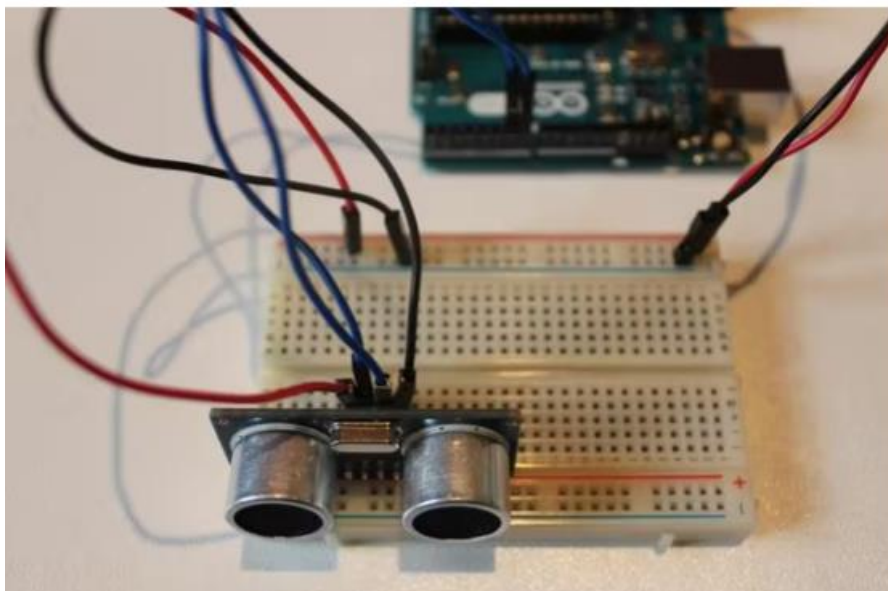


Pas 3: Asamblați – Breadboard



Mai întâi 5V și GND pin de pe Arduino la breadboard.

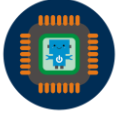
Pas 4: Asamblați - Ultrasonic Sensor



HC-SR04 ultrasonic sensor! Plasați ultrasonic sensor cu fața în sus cât mai în dreapta posibil.

Conectați:

- GND pin de la ultrasonic sensor la canalul negativ pe breadboard.
- Trig pin de pe sensor la pin 2 de pe Arduino
- Echo pin de pe sensor la pin 3 pe Arduino.



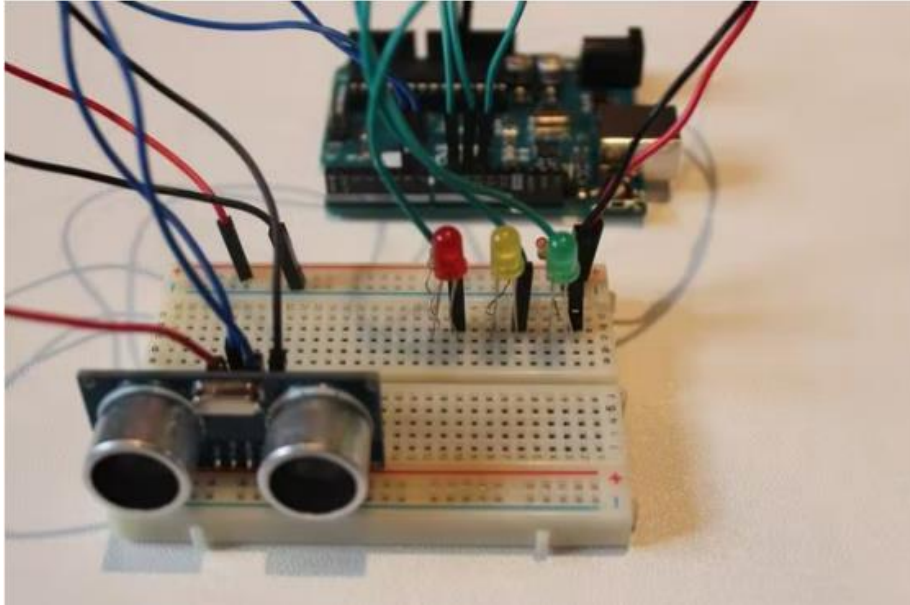
ROBOSTEM Project

Agreement no: 2019-1-RO01-KA202-063965

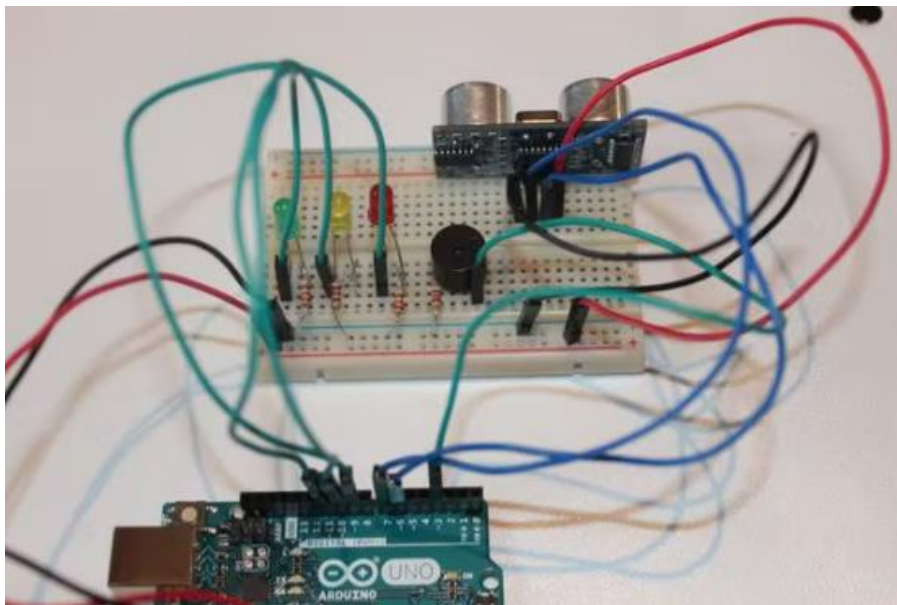


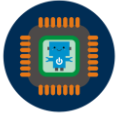
- VCC pin de pe ultrasonic sensor la canalul pozitiv pe breadboard.

Pas 5: Asamblați - LEDs



Pas 6: Asamblați – Buzzer



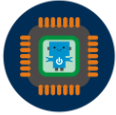


ROBOSTEM Project

Agreement no: 2019-1-RO01-KA202-063965



```
const int trigPin = 2;
const int echoPin = 3;
const int LEDlampRed = 4;
const int LEDlampYellow = 5;
const int LEDlampGreen = 6;
const int buzzer = 7;
int sound = 500;
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDlampRed, OUTPUT);
  pinMode(LEDlampYellow, OUTPUT);
  pinMode(LEDlampGreen, OUTPUT);
  pinMode(buzzer, OUTPUT);
}
void loop() {
  long durationindigit, distanceincm;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
}
```



ROBOSTEM Project

Agreement no: 2019-1-RO01-KA202-063965



```
digitalWrite(trigPin, LOW);
durationindigit = pulseIn(echoPin, HIGH);
distanceincm = (durationindigit * 0.034) / 2;
if (distanceincm > 50) {
    digitalWrite(LEDlampGreen, LOW);
    digitalWrite(LEDlampYellow, LOW);
    digitalWrite(LEDlampRed, LOW);
    noTone(buzzer);
}
else if (distanceincm <= 50 && distanceincm > 20) {
    digitalWrite(LEDlampGreen, HIGH);
    digitalWrite(LEDlampYellow, LOW);
    digitalWrite(LEDlampRed, LOW);
    noTone(buzzer);
}
else if (distanceincm <= 20 && distanceincm > 5) {
    digitalWrite(LEDlampYellow, HIGH);
    digitalWrite(LEDlampGreen, HIGH);
    digitalWrite(LEDlampRed, LOW);
    tone(buzzer, 500);
}
else if (distanceincm <= 0) {
    digitalWrite(LEDlampGreen, LOW);
    digitalWrite(LEDlampYellow, HIGH);
    digitalWrite(LEDlampRed, LOW);
    noTone(buzzer);
}
else {
    digitalWrite(LEDlampGreen, HIGH);
    digitalWrite(LEDlampYellow, HIGH);
    tone(buzzer, 1000);
    digitalWrite(LEDlampRed, HIGH);
    delay(300);
    digitalWrite(LEDlampRed, LOW);
}
Serial.print(distanceincm);
Serial.println(" cm");
delay(300);
}
```