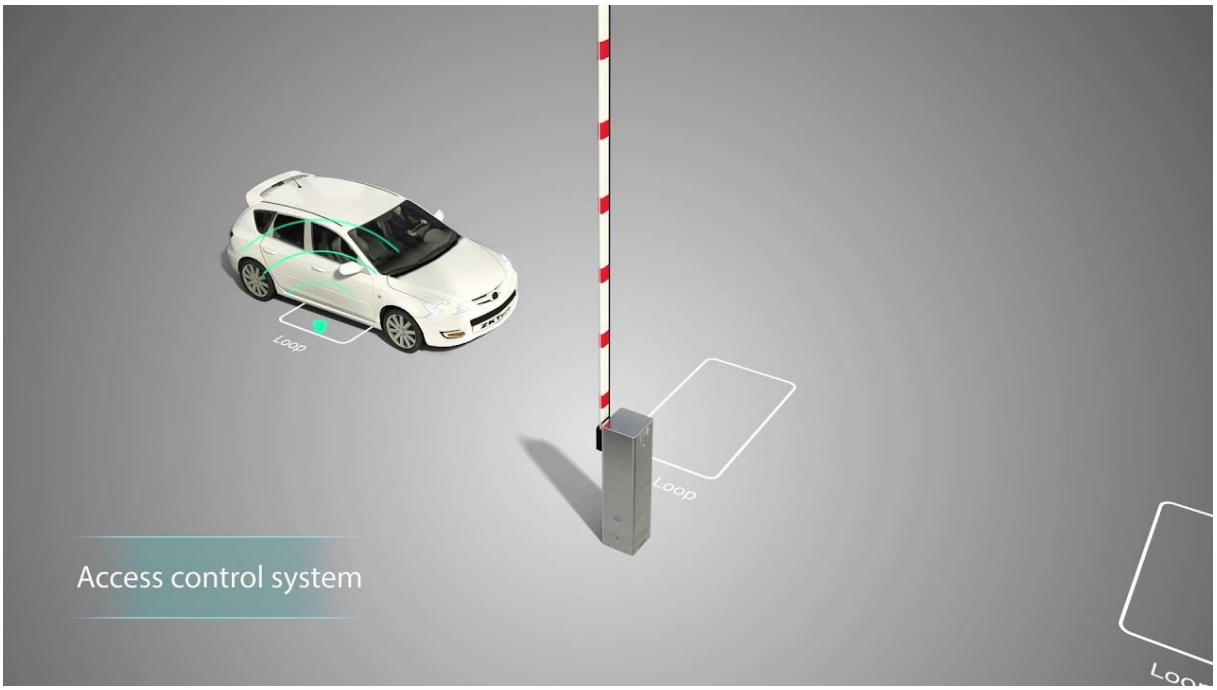
Raport Tehnic al Proiectului:

Barieră Auto Automatizata



Cuprins

1. Introducere

Acest proiect are ca scop realizarea unei bariere auto controlate de un senzor de proximitate. Bariera se va ridica automat atunci când un vehicul se apropie și va reveni în poziția inițială după ce vehiculul a trecut. În plus, bariera poate fi controlată manual prin intermediul unei telecomenzi cu infraroșu (IR).Controlul cu telecomanda a fost gandit in scopul utilizarii sistemului chiar daca senzorul ar ceda .

Sistemul realizat poate fi aplicat în diverse domenii și poate fi integrat în multiple tipuri de sisteme, după cum urmează:

Sisteme de Control al Accesului: În clădiri de birouri, complexe rezidențiale sau unități industriale.

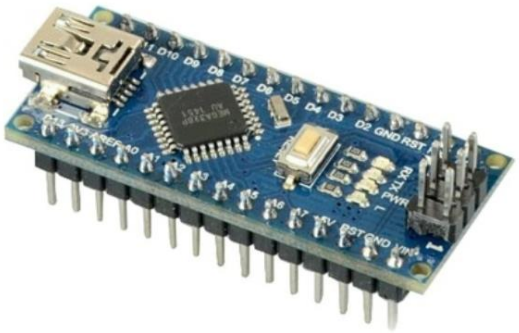
Gări și Aeroporturi: În aceste locații, managementul eficient al fluxului de vehicule este esențial. Bariera auto poate fi utilizată pentru a gestiona accesul la zonele de încărcare și descărcare, precum și la parcările pentru pasageri.

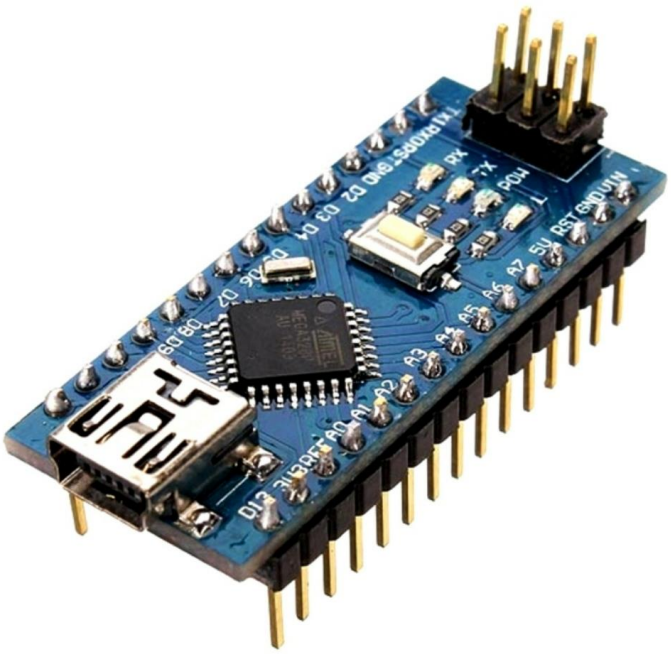
Unități Medicale: În spitale și clinici, unde accesul rapid și controlat este crucial.

Usi Glisante și Sisteme de Control Acces pentru Pietoni: Tehnologia poate fi adaptată pentru a controla ușile glisante automate în clădiri comerciale sau rezidențiale.

2. Componente Hardware utilizate

Arduino Nano: Placa de dezvoltare principală.





Specificatii Tehnice:

Tensiune de funcționare: 5 V;

Tensiune de alimentare Jack: 7 V - 12 V;

Pini I/O: 14;

Pini PWM: 6 (din cei 14 de I/O);

Pini analogici: 8;

Memorie flash: 32 KB, din care 2 KB ocupați de bootloader;

Frecventa de funcționare: 16 MHz.

Dimensiuni: 45 mm x 18 mm;

.....

Servo Motor: Pentru ridicarea și coborârea barierei.

Caracteristici tehnice:

Tensiune de alimentare: 4.8V - 6V;

Consum redus de curent;

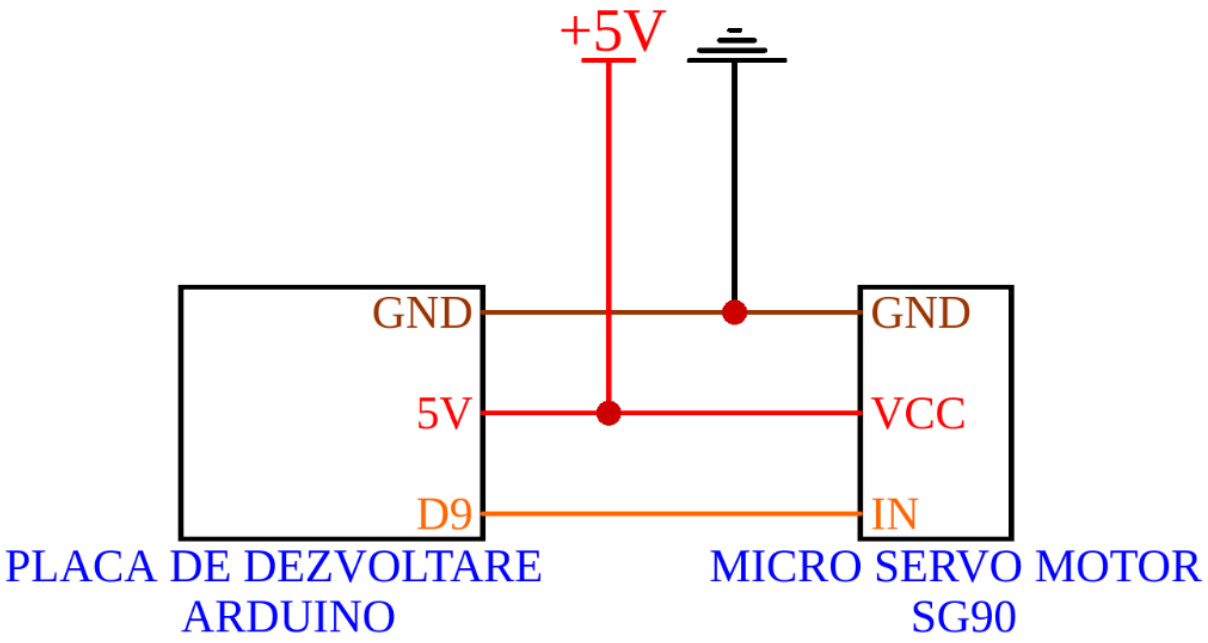
Viteza de funcționare: 0.12 s/60o @ 4.8 V;

Cuplu în blocare la 4.8V: 1.8 kgf*cm;

Frecvență PWM: 50Hz (conform datasheet-ului anexat);

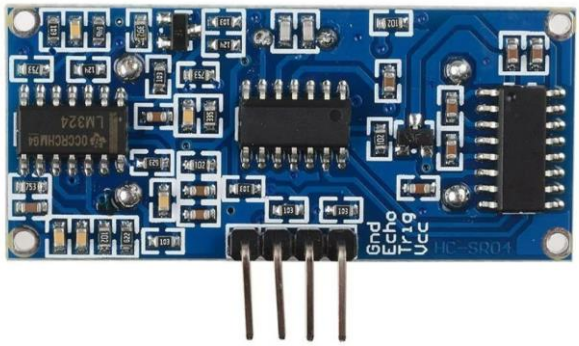
Temperatura de funcționare: -30° C - +60° C.

Dimensiuni: 21.5 x 11.8 x 22.7 mm



.....

Senzor de proximitate (HC-SR04): Pentru detectare



Specificații:

Tensiune de alimentare: 5 VDC

Consum de curent static: mai puțin de 2 mA

Unghiul fasciculului senzorului: mai puțin de 15°

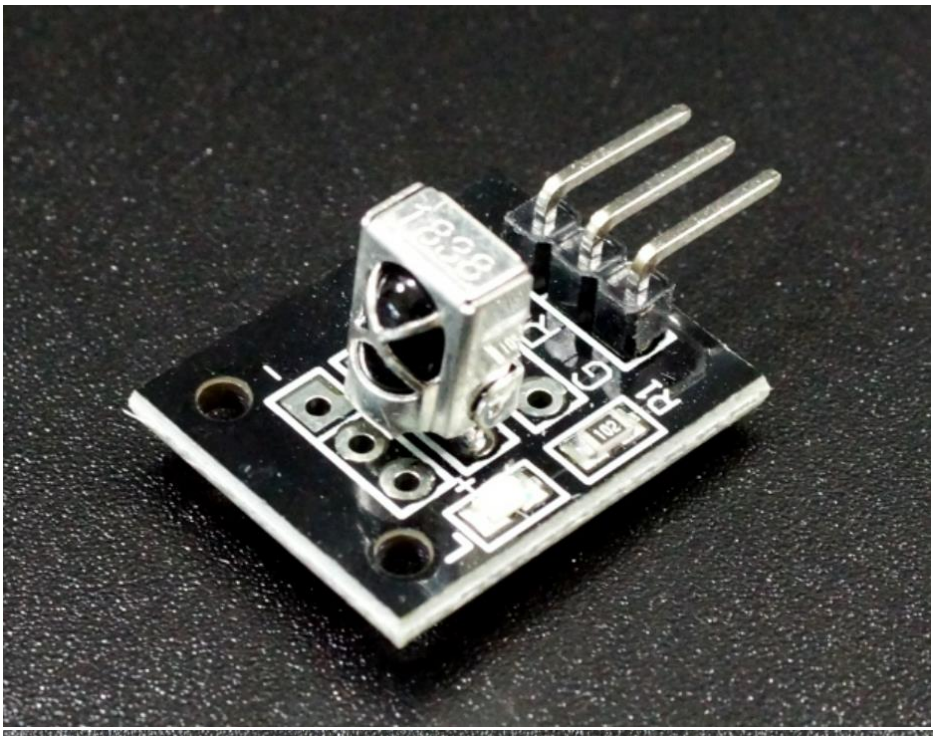
Interval de detectare: 2 cm până la 450 cm

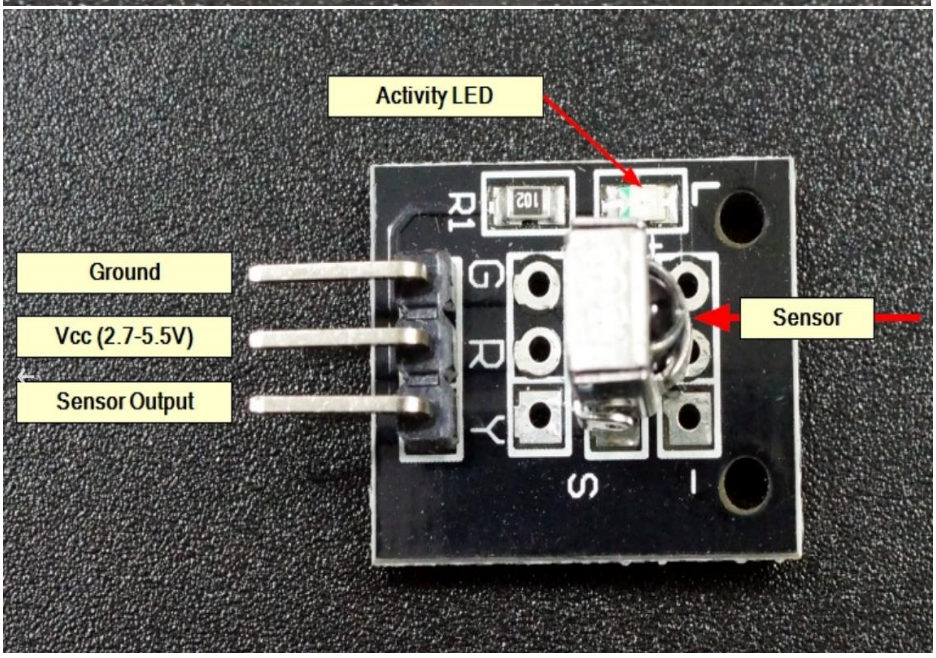
Rezolutie: 0,3 cm

Notă: Nu conectați senzorul în timp ce circuitul este alimentat. Acest lucru poate duce la deteriorarea sau o ușoară degradare a performanței.

.....

Receptor IR (TSOP4838 sau similar): Pentru recepționarea semnalelor de la telecomandă.





DESCRIERE:

Modulul receptor IR VS1838 detectează infraroșu (IR) în spectrul utilizat în mod obișnuit pentru telecomandă IR sau senzori de intruziune IR.

G or '-' = Ground

Pin central sau R = Vcc (de obicei 3.3 or 5V)

Y or S = Digital Output. De obicei se conecteaza la pinul de intrare digitala de pe MCU

.....

Telecomandă IR: Pentru controlul manual al barierei.

Caracteristici tehnice:

Tensiune de alimentare: 3V;

Baterie litiu CR2032;

Frecvența de modulație: 38 KHz;

Raza de acțiune: 8-10 m

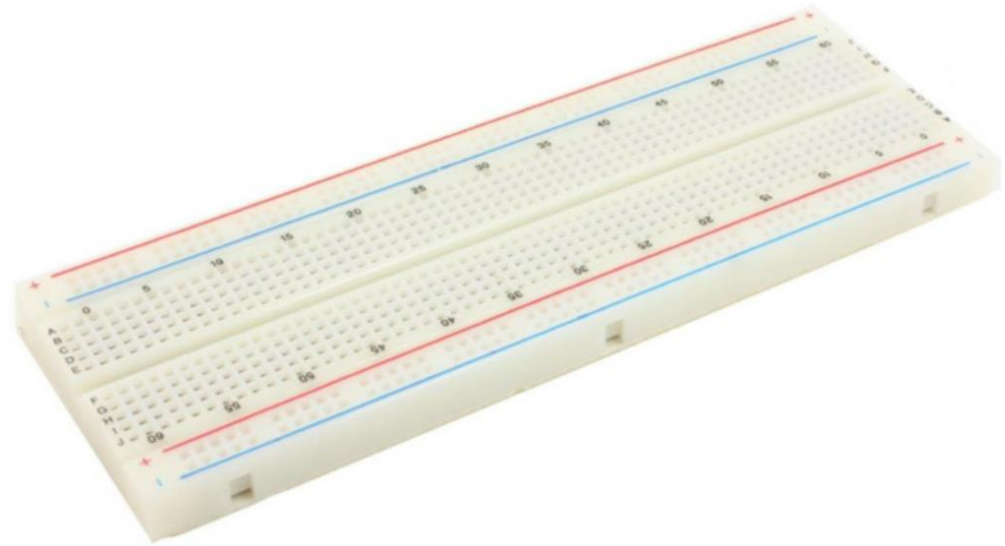
Prezintă buton de schimbare a canalelor și de control al volumului.

Telecomanda poate fi folosită împreună cu un receptor infraroșu și este utilă pentru a trimite comenzi la distanță.

Are o rază de acțiune destul de mare și poate fi utilizată, de exemplu, pentru a porni și opri roboți sau, având un număr destul de mare de butoane, pentru a controla un MP3 Player sau alte dispozitive complexe.

.....

Breadboard:



Caracteristici tehnice:

Dimensiuni: 16.5 x 5.4 x 0.85mm;

Număr de puncte: 830;

Diametru fir necesar: 0.8mm.

Acest breadboard de calitate bună este perfect pentru proiectele dumneavoastră de test în care doriți să vedeți rapid cum funcționează un montaj. Pentru un contact cat mai bun al firelor, conexiunile interne sunt placate cu un aliaj de nichel și bronz.

Conexiunile interne de la dreadboard sunt realizate orizontal sau vertical, conform pozei atașate.

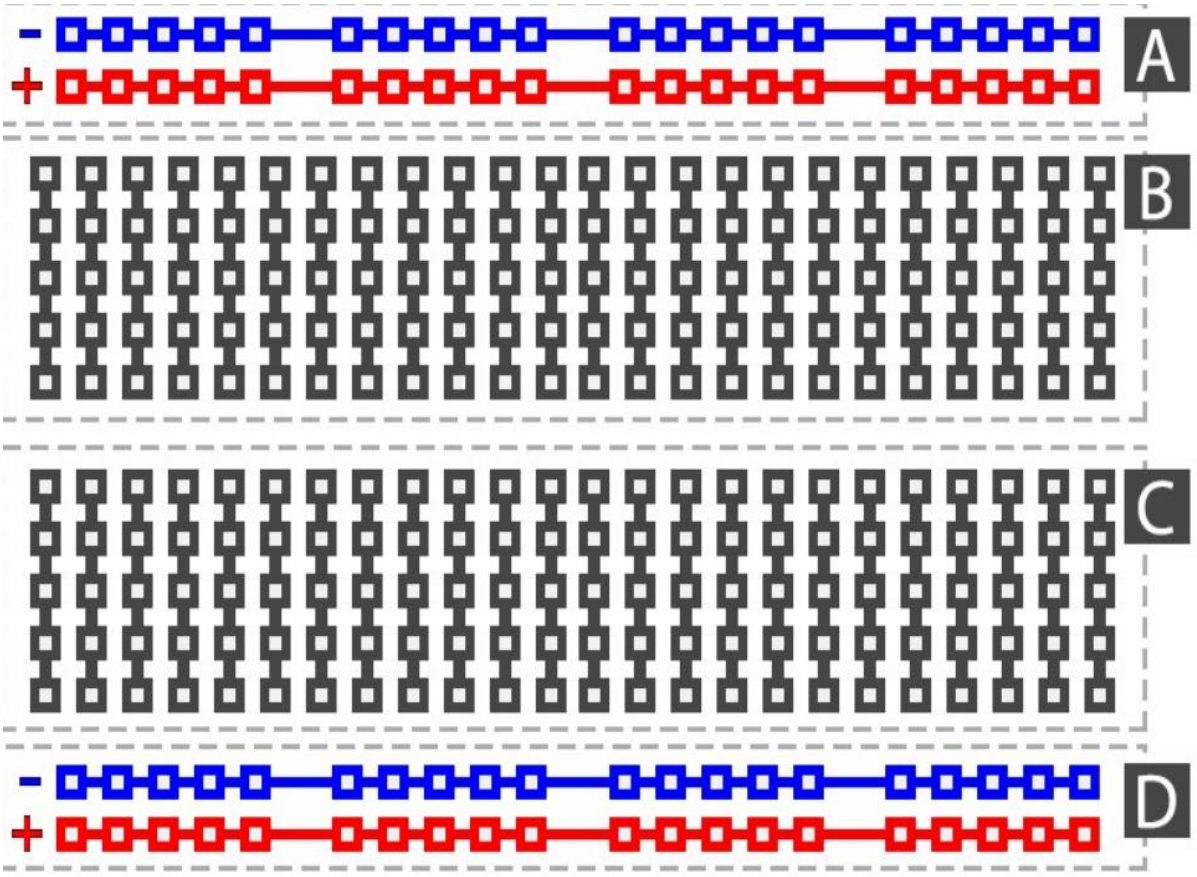
Aplicațiile acestui produs sunt foarte multe, fiind recomandat atunci când începeți să învățați electronică, putând crea foarte multe montaje fără să fiți nevoiți să faceți lipituri.

Utilizarea unui breadboard este recomandată în momentul în care ne dorim construirea rapidă și ușoară a unui circuit. Acesta este o placă de plastic cu găuri și contacte metalice în interior, ce permite conectarea prin fire a elementelor de circuit, fără a mai fi nevoie să le lipim.

Utilizare împreună cu Arduino

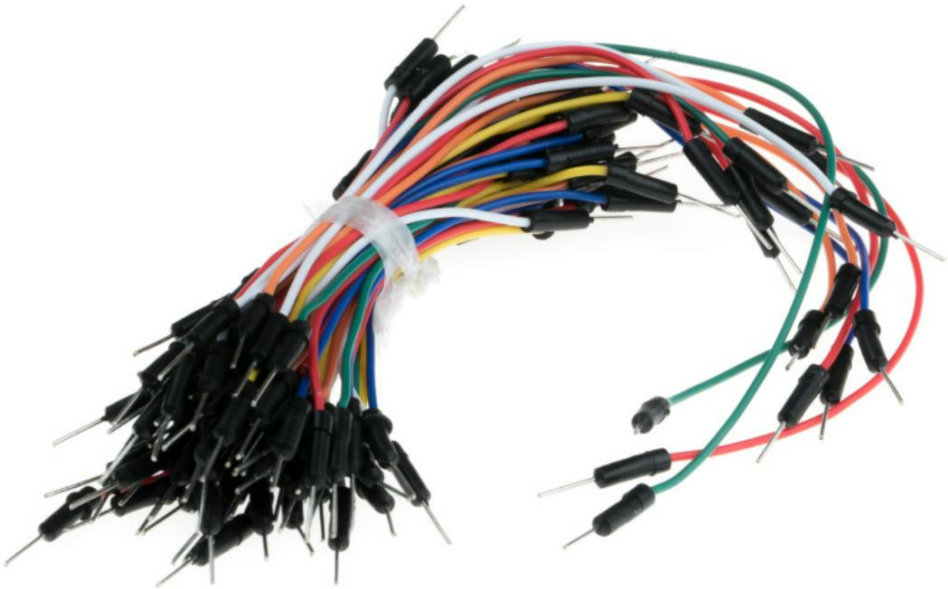
Breadboardurile pot avea diverse dimensiuni. "Anatomia" de bază a acestora se poate observa în imaginea alaturata:

- **secțiunile orizontale A și D vor fi folosite de obicei pentru alimentare;**
- **în secțiunile verticale B și C vom insera elementele de circuit;**


A
B
C
D

.....

Fire de conexiune :



INFORMAȚII

Acest set conține aproximativ 65 de cabluri cu următoarele lungimi:

25 cm

20.3 cm

16.7 cm

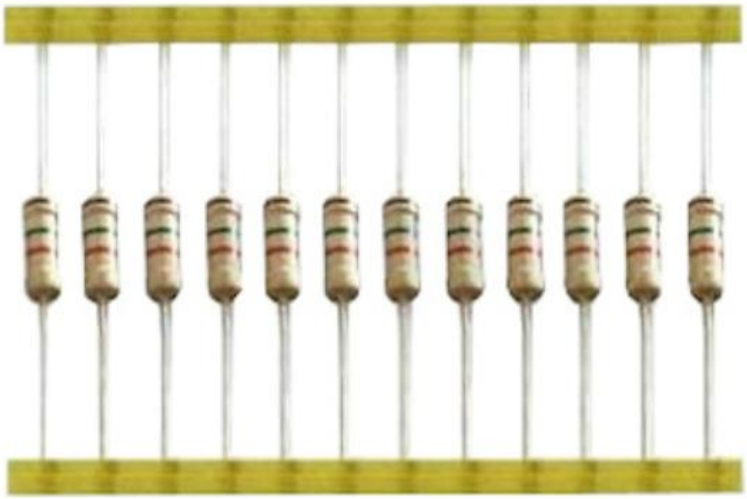
15.7 cm

11.7 cm

Cablurile sunt de tip tată - tată și sunt ideale pentru semnale analogice sau digitale date de plăcuțe de dezvoltare sau module sau circuite integrate.

.....

Rezistențe:



<u>INFORMAȚII</u>

Rezistor 0.25W 220Ω

Tip: CFR1/4

Valori 0.1 Ohm pana la 22M Ohm

Toleranta: ±5%

Putere maxima la 70°C: 0.25W

Dimensiuni: 2.5 x 6,8 mm

.....

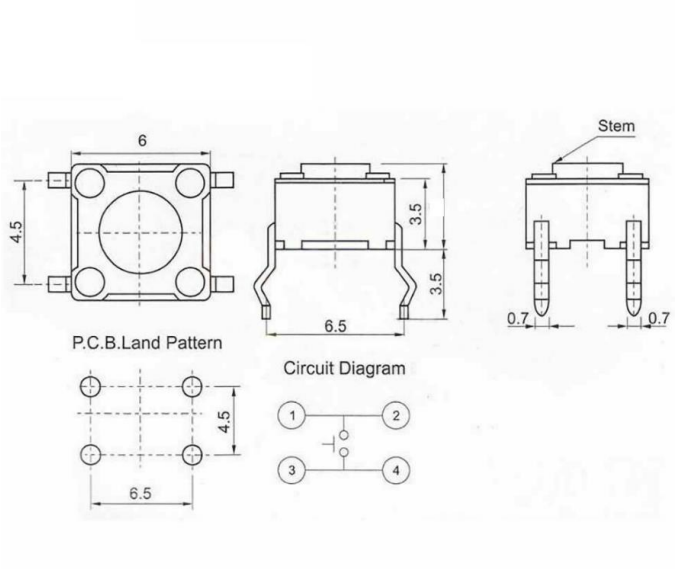
BUTON MINI 6X6X5, 4 PINI:

Buton de mici dimensiuni cu 4 pini, compatibil cu Breadboard.

Specificatii:

Dimensiuni mm: 6 x 6 x 5

Distanta intre pini: 4.5mm x 6.5mm



5 Leduri de diferite culori :

La acest proiect avem nevoie de 2 rosii, 2 verzi,1 galben

INFORMAȚII

- **2 LED-uri roșii 3 mm cu lentile difuze;**
- **1 LED galbene 3 mm cu lentile difuze;**
- **2 LED-uri verzi 3 mm cu lentile difuze;**

.....

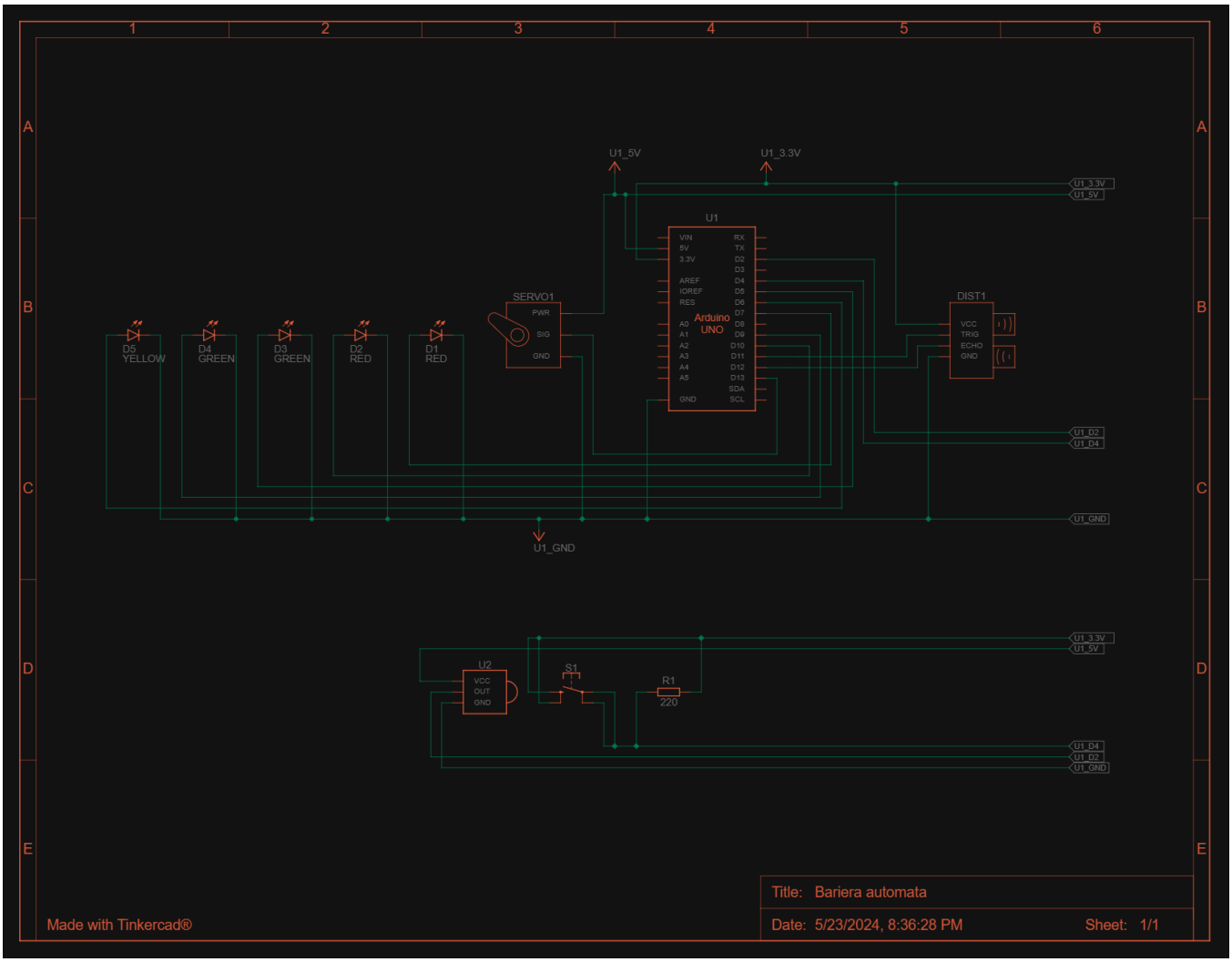
Cablu miniUSB(in cazul in care arduino nano nu vine cu el)

Cu acest cablu puteti conecta dispozitive cu port miniUSB la calculator sau laptop (pe port USB 2.0)

Lungime cablu: 1.5m

.....

Schema Bloc circuit



Pseudocod

Variabile Globale

Servo servo // Obiect pentru controlul servo motorului

int trigPin = 11 // Pin pentru semnalul de declanșare al senzorului de proximitate

int echoPin = 12 // Pin pentru semnalul de ecou al senzorului de proximitate

long duration // Durata semnalului de ecou

int distance // Distanța măsurată de senzor

const int RECV_PIN = 2 // Pin pentru receptorul IR

IRrecv irrecv(RECV_PIN) // Obiect pentru recepția semnalelor IR

decode_results results // Structură pentru rezultatele decodificării semnalelor IR

int mode = 0 // Mod de operare curent (0 pentru automat, 1 pentru telecomandă)

int butonPin = 4 // Pin pentru buton

bool lastButtonState = LOW // Starea precedentă a butonului

bool currentButtonState = LOW // Starea curentă a butonului

Funcția setup()

INITIALIZEAZĂ comunicarea serială la 9600 baud

ACTIVEAZĂ receptorul IR

ATAȘEAZĂ servo motorul la pinul 13

SETARE servo motor la poziția 180 grade

AȘTEAPTĂ 2 secunde

CONFIGUREAZĂ pinii pentru LED-uri ca output

CONFIGUREAZĂ pinul butonului ca input

CONFIGUREAZĂ trigPin ca output

CONFIGUREAZĂ echoPin ca input

// Funcția pentru modul automat

Funcția ridica_automat()

SET trigPin LOW

AȘTEAPTĂ 2 microsecunde

SET trigPin HIGH

AȘTEAPTĂ 10 microsecunde

SET trigPin LOW

MĂSOARĂ durata semnalului de ecou folosind echoPin

CALCULEAZĂ distanța în baza duratei semnalului de ecou

Dacă distanța este mai mică sau egală cu 10 cm:

SETARE servo motor la 180 grade (ridică bariera)

APRINDE LED-ul verde

STINGE LED-ul roșu

AȘTEAPTĂ 3 secunde

STINGE LED-ul verde

APRINDE LED-ul roșu

Altfel:

SETARE servo motor la 100 grade (coboară bariera)

APRINDE LED-ul roșu

Dacă distanța este mai mare de 20 cm:

APRINDE LED-ul 1 (verde)

STINGE LED-ul 2 și LED-ul 3

Altfel dacă distanța este între 14 și 20 cm:

APRINDE LED-ul 1 și LED-ul 2

STINGE LED-ul 3

Altfel:

APRINDE LED-ul 1, LED-ul 2 și LED-ul 3

<u>Funcția mod_telecomanda()</u>

Dacă receptorul IR primește un semnal:

Afișează codul IR primit

Dacă semnalul este repetitiv:

Continuă decodificarea următorului semnal

Dacă codul IR este 0xF30CFF00 (comanda pentru servo la 180 grade):

SETARE servo motor la 180 grade (ridică bariera)

APRINDE LED-ul verde

STINGE LED-ul roșu

Afișează mesajul "rotit cu 180"

Altfel dacă codul IR este 0xE718FF00 (comanda pentru servo la 100 grade):

SETARE servo motor la 110 grade (coboară bariera)

STINGE LED-ul verde

APRINDE LED-ul roșu

Afișează mesajul "rotit cu 130"

Continuă cu următoarea decodificare

<u>Funcția loop()</u>

// Funcția principală de execuție

CITEȘTE starea curentă a butonului

Dacă butonul a fost apăsat (tranziție de la LOW la HIGH):

AȘTEAPTĂ 50 milisecunde (debouncing)

Dacă butonul este încă apăsat:

Comută între modurile de operare (automat sau telecomandă)

Dacă modul curent este automat:

Afișează mesajul "Mod: ridica_automat"

Altfel:

Afișează mesajul "Mod: mod_telecomanda"

ACTUALIZEAZĂ starea precedentă a butonului

Dacă modul curent este automat:

APELEAZĂ funcția ridica_automat()

Altfel:

APELEAZĂ funcția mod_telecomanda()

Cod

```
#include <IRremote.h>
#include <Servo.h>

Servo servo;
int trigPin = 11;
int echoPin = 12;
long duration;
int distance;
const int RECV_PIN = 2;  // Pinul pe care este conectat receptorul IR
IRrecv irrecv(RECV_PIN);
decode_results results;
int state = 0; // Inițializăm variabila de control a while
int mode = 0;
int currentMode = 0;          // 0 pentru ridica_automat, 1 pentru mod_telecomanda
int butonPin = 4;
bool lastButtonState = LOW;    // Starea precedentă a butonului
bool currentButtonState = LOW; // Starea curentă a butonului
void setup() {
  Serial.begin(9600);  // Inițializează comunicația serială cu PC-ul
```

```
  irrecv.enableIRIn(); // Activează receptorul IR pentru a primi semnale
  servo.attach(13); // Conectează servo la pinul 9
  Serial.println("Ready to receive IR signals");
  servo.attach(13);
 servo.write(180);
 delay(2000);
 pinMode(7,OUTPUT);
 pinMode(6,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(10,OUTPUT);
 pinMode(9,OUTPUT);
 pinMode(4,INPUT);
 //Sets the trigPin as an Output
 pinMode(trigPin,OUTPUT);
 //Sets the echopin as an Input
 pinMode(echoPin, INPUT);
}

void ridica_automat(){
//functia pentru modul automat
 digitalWrite(trigPin, LOW);
          delayMicroseconds(2);
          digitalWrite(trigPin, HIGH);
```

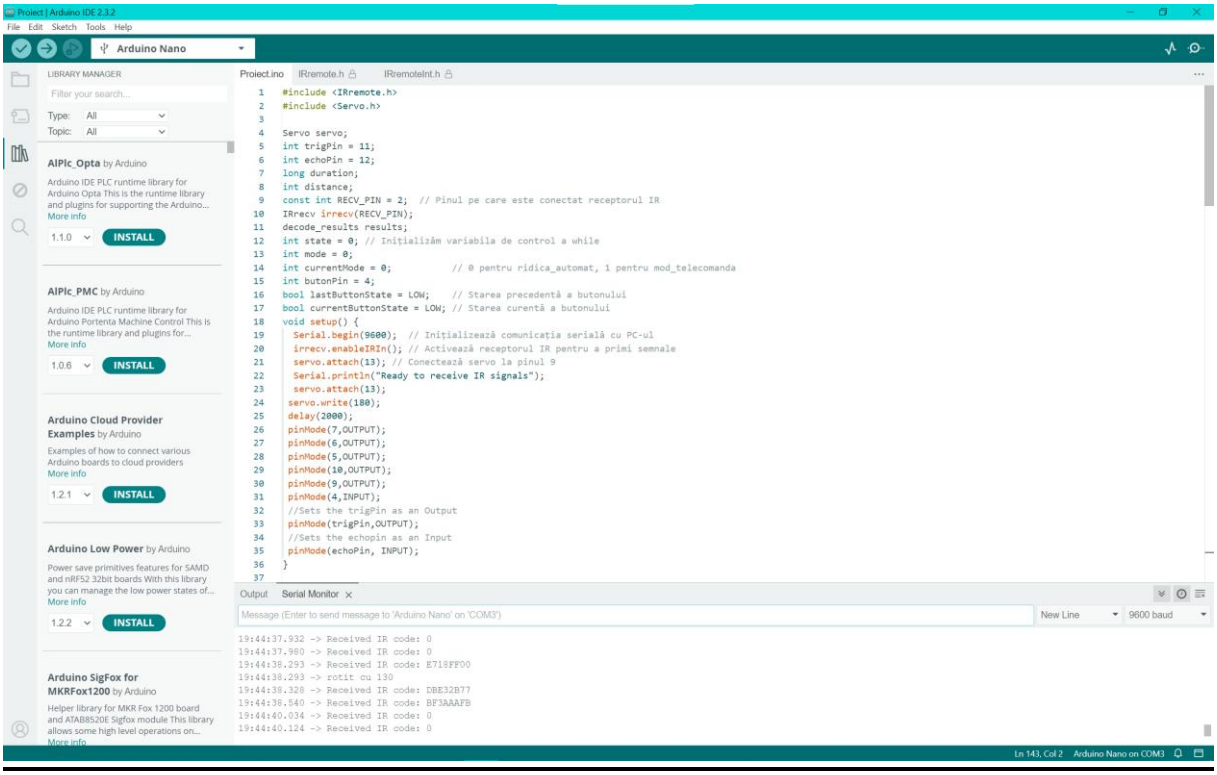
```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin,HIGH);
  distance = duration*0.034/2;
  Serial.println("Distance:");
  Serial.print(distance);
  if(distance<= 10)
  {
  servo.write(180);
   digitalWrite(9, HIGH);
   digitalWrite(10,LOW);
  delay(3000);
  digitalWrite(9,LOW);
  digitalWrite(10,HIGH);
  }
  else
  {
    servo.write(100);
    digitalWrite(10, HIGH);
        }

if (distance > 20) {
  digitalWrite(7, HIGH); // Aprindem LED-ul 1
  digitalWrite(6, LOW);  // Stingem LED-ul 2
```

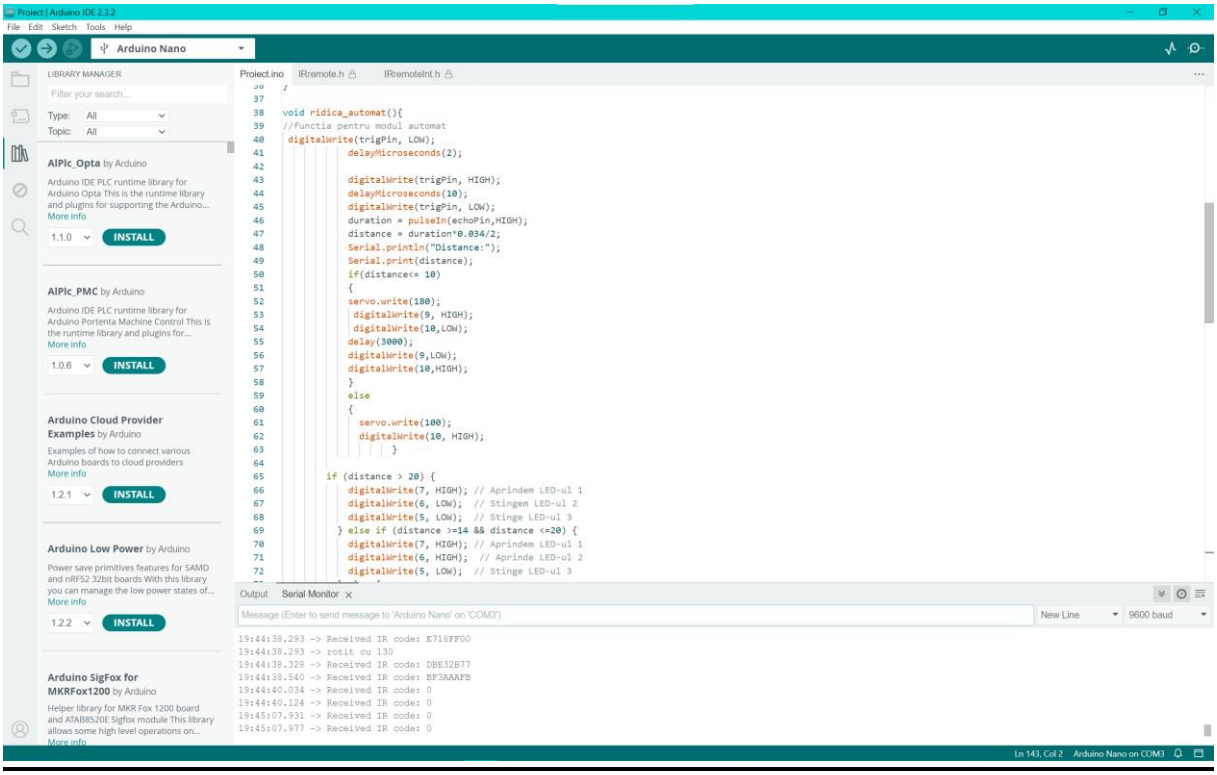
```
      digitalWrite(5, LOW);  // Stinge LED-ul 3
    } else if (distance >=14 && distance <=20) {
      digitalWrite(7, HIGH); // Aprindem LED-ul 1
      digitalWrite(6, HIGH);  // Aprinde LED-ul 2
      digitalWrite(5, LOW);  // Stinge LED-ul 3
    } else {
      digitalWrite(7, HIGH); // Aprindem LED-ul 1
      digitalWrite(6, HIGH);  // Aprinde LED-ul 2
      digitalWrite(5, HIGH);  // Aprinde LED-ul 3
    }

}

 void mod_telecomanda(){
//functia pentru modul telecomanda
  if (irrecv.decode()) {
    // Identificăm butonul și afișăm codul IR asociat
    Serial.print("Received IR code: ");
    Serial.println(irrecv.decodedIRData.decodedRawData, HEX);

    if (irrecv.decodedIRData.flags & IRDATA_FLAGS_IS_REPEAT) {
      irrecv.resume(); // Continuă cu următoarea decodificare
      return;
    }
```

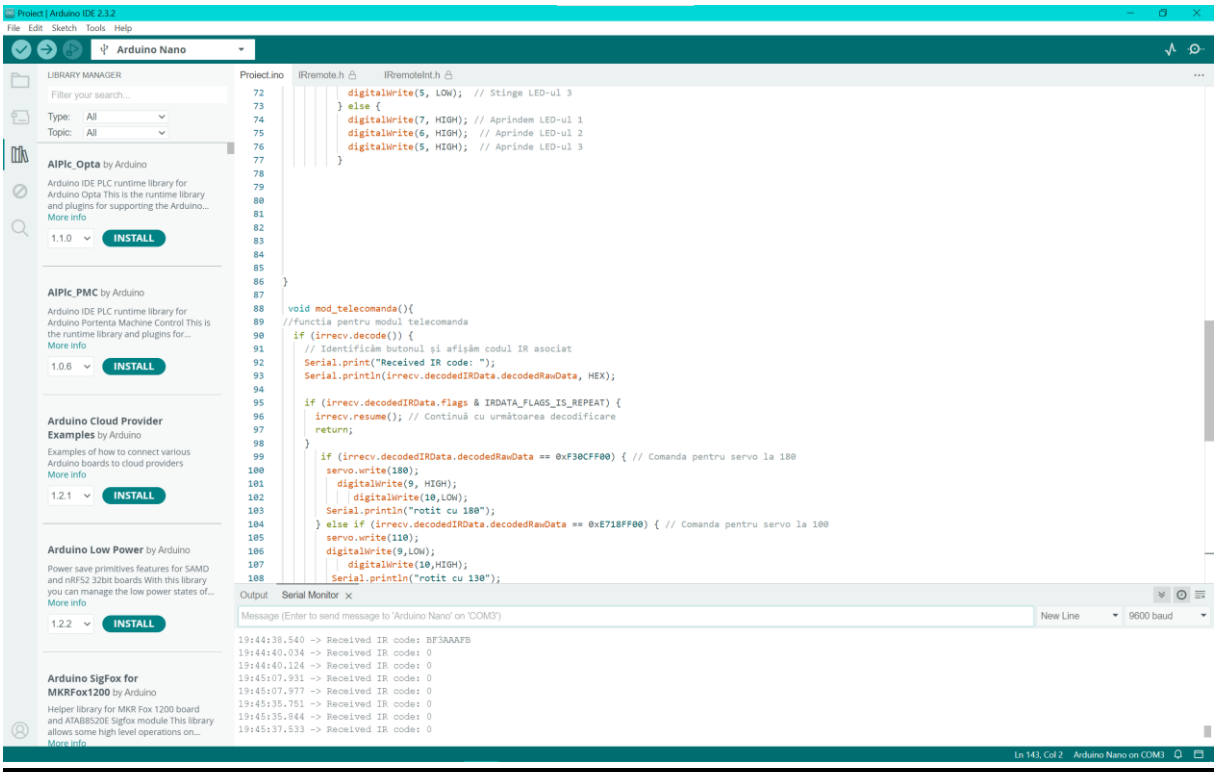
```
    if (irrecv.decodedIRData.decodedRawData == 0xF30CFF00) { // Comanda pentru servo la 180
      servo.write(180);
       digitalWrite(9, HIGH);
         digitalWrite(10,LOW);
      Serial.println("rotit cu 180");
    } else if (irrecv.decodedIRData.decodedRawData == 0xE718FF00) { // Comanda pentru servo la 100
      servo.write(110);
      digitalWrite(9,LOW);
        digitalWrite(10,HIGH);
       Serial.println("rotit cu 130");
    }
     irrecv.resume();
 }
 }
void loop() {

 currentButtonState = digitalRead(butonPin); // Citește starea butonului

  // Detectează o apăsare de buton (tranziție de la LOW la HIGH)
  if (currentButtonState == HIGH && lastButtonState == LOW) {
   // Debouncing - așteaptă puțin și verifică din nou starea butonului
```

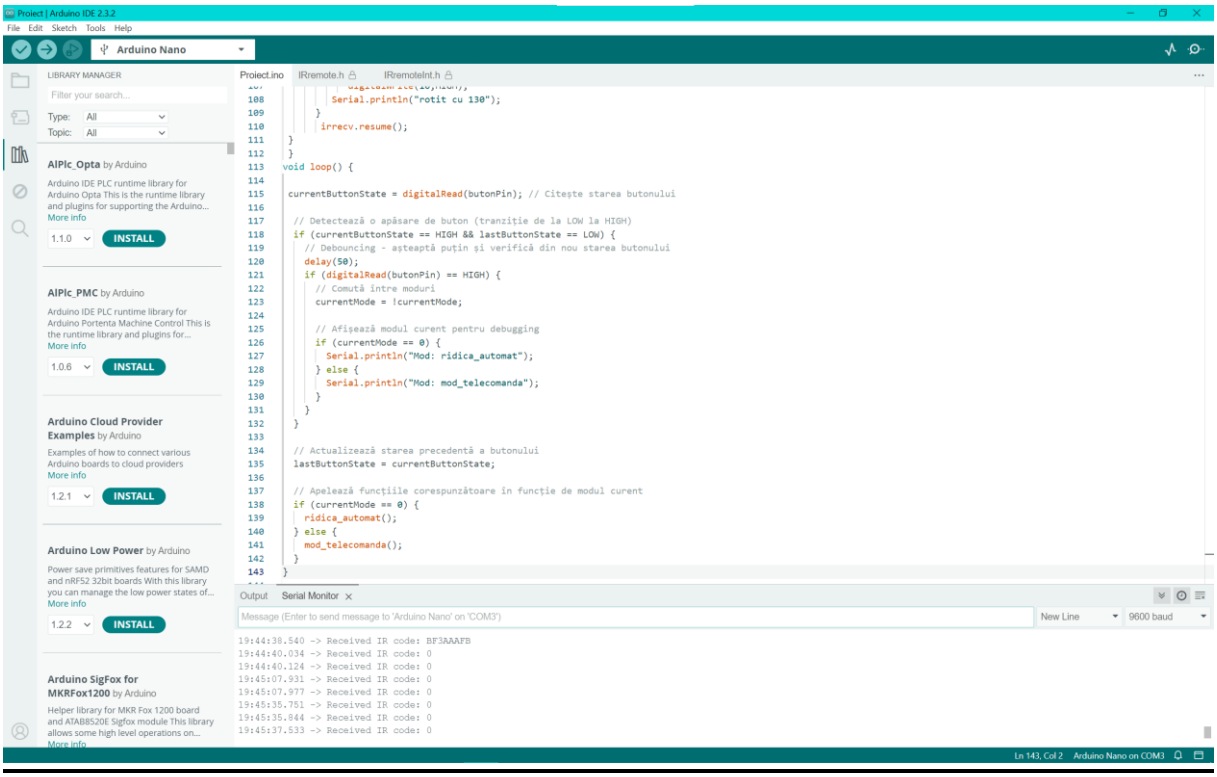
```
    delay(50);
    if (digitalRead(butonPin) == HIGH) {
      // Comută între moduri
      currentMode = !currentMode;

      // Afișează modul curent pentru debugging
      if (currentMode == 0) {
        Serial.println("Mod: ridica_automat");
      } else {
        Serial.println("Mod: mod_telecomanda");
      }
    }
  }

  // Actualizează starea precedentă a butonului
  lastButtonState = currentButtonState;

  // Apelează funcțiile corespunzătoare în funcție de modul curent
  if (currentMode == 0) {
    ridica_automat();
  } else {
    mod_telecomanda();
}
}
```

Fotografii Proiect






Proiect | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Nano
LIBRARY MANAGER
Filter your search...
Type: All
Topic: All
AIPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins for supporting the Arduino...
More info
1.1.0 INSTALL
AIPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime library and plugins for...
More info
1.0.6 INSTALL
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info
1.2.1 INSTALL
Arduino Low Power by Arduino
Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of...
More info
1.2.2 INSTALL
Arduino SigFox for MKRFox1200 by Arduino
Helper library for MKR Fox 1200 board and ATAB8520E Sigfox module This library allows some high level operations on...
More info
Proiect.ino
IRremote.h
IRremoteInt.h
digitalWrite(5, LOW); // Stinge LED-ul 3
} else {
digitalWrite(7, HIGH); // Aprindem LED-ul 1
digitalWrite(6, HIGH); // Aprinde LED-ul 2
digitalWrite(5, HIGH); // Aprinde LED-ul 3
}
}
void mod_telecomanda(){
//functia pentru modul telecomanda
if (irrecv.decode()) {
// Identificăm butonul și afișăm codul IR asociat
Serial.print("Received IR code: ");
Serial.println(irrecv.decodedIRData.decodedRawData, HEX);
if (irrecv.decodedIRData.flags & IRDATA_FLAGS_IS_REPEAT) {
irrecv.resume(); // Continuă cu următoarea decodificare
return;
}
if (irrecv.decodedIRData.decodedRawData == 0xF30CFF00) { // Comanda pentru servo la 180
servo.write(180);
digitalWrite(9, HIGH);
digitalWrite(10,LOW);
Serial.println("rotit cu 180");
} else if (irrecv.decodedIRData.decodedRawData == 0xE718FF00) { // Comanda pentru servo la 100
servo.write(110);
digitalWrite(9,LOW);
digitalWrite(10,HIGH);
Serial.println("rotit cu 130");
Output Serial Monitor
Message (Enter to send message to 'Arduino Nano' on 'COM3')
New Line 9600 baud
19:44:38.540 -> Received IR code: BF3AAAFB
19:44:40.034 -> Received IR code: 0
19:44:40.124 -> Received IR code: 0
19:45:07.931 -> Received IR code: 0
19:45:07.977 -> Received IR code: 0
19:45:35.751 -> Received IR code: 0
19:45:35.844 -> Received IR code: 0
19:45:37.533 -> Received IR code: 0
Ln 143, Col 2 Arduino Nano on COM3


Proiect | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Nano
LIBRARY MANAGER
Filter your search...
Type: All
Topic: All
AIPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins for supporting the Arduino...
More info
1.1.0 INSTALL
AIPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime library and plugins for...
More info
1.0.6 INSTALL
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info
1.2.1 INSTALL
Arduino Low Power by Arduino
Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of...
More info
1.2.2 INSTALL
Arduino SigFox for MKRFox1200 by Arduino
Helper library for MKR Fox 1200 board and ATAB8520E Sigfox module This library allows some high level operations on...
More info
Proiect.ino
IRremote.h
IRremoteInt.h
Serial.println("rotit cu 130");
}
irrecv.resume();
}
}
void loop() {
currentButtonState = digitalRead(butonPin); // Citește starea butonului
// Detectează o apăsare de buton (tranziție de la LOW la HIGH)
if (currentButtonState == HIGH && lastButtonState == LOW) {
// Debouncing - așteaptă puțin și verifică din nou starea butonului
delay(50);
if (digitalRead(butonPin) == HIGH) {
// Comută între moduri
currentMode = !currentMode;
// Afișează modul curent pentru debugging
if (currentMode == 0) {
Serial.println("Mod: ridica_automat");
} else {
Serial.println("Mod: mod_telecomanda");
}
}
}
// Actualizează starea precedentă a butonului
lastButtonState = currentButtonState;
// Apelează funcțiile corespunzătoare în funcție de modul curent
if (currentMode == 0) {
ridica_automat();
} else {
mod_telecomanda();
}
}
Output Serial Monitor
Message (Enter to send message to 'Arduino Nano' on 'COM3')
New Line 9600 baud
19:44:38.540 -> Received IR code: BF3AAAFB
19:44:40.034 -> Received IR code: 0
19:44:40.124 -> Received IR code: 0
19:45:07.931 -> Received IR code: 0
19:45:07.977 -> Received IR code: 0
19:45:35.751 -> Received IR code: 0
19:45:35.844 -> Received IR code: 0
19:45:37.533 -> Received IR code: 0
Ln 143, Col 2 Arduino Nano on COM3