

Universitatea Tehnica Cluj-Napoca

Rover Arduino cu Control prin Bluetooth



Echipa: Barz Sebastian

Buzgan Paul

Grupa: 30236

1. Obiective

Obiectivul acestui proiect este implementarea unui robot care se deplaseaza cu ajutorul a doua motoare, a unui sistem de tractiune bazat pe 3 roti si comandat prin Bluetooth printr-o aplicatie de mobil. Pe langa acest aspect robotul mai are o multitudine de senzori ce ii imbunatatesc comenzile si de asemenea aceste valori pot fi transmise pentru citire cu usurinta catre terminalul Bluetooth de control pentru telefon.

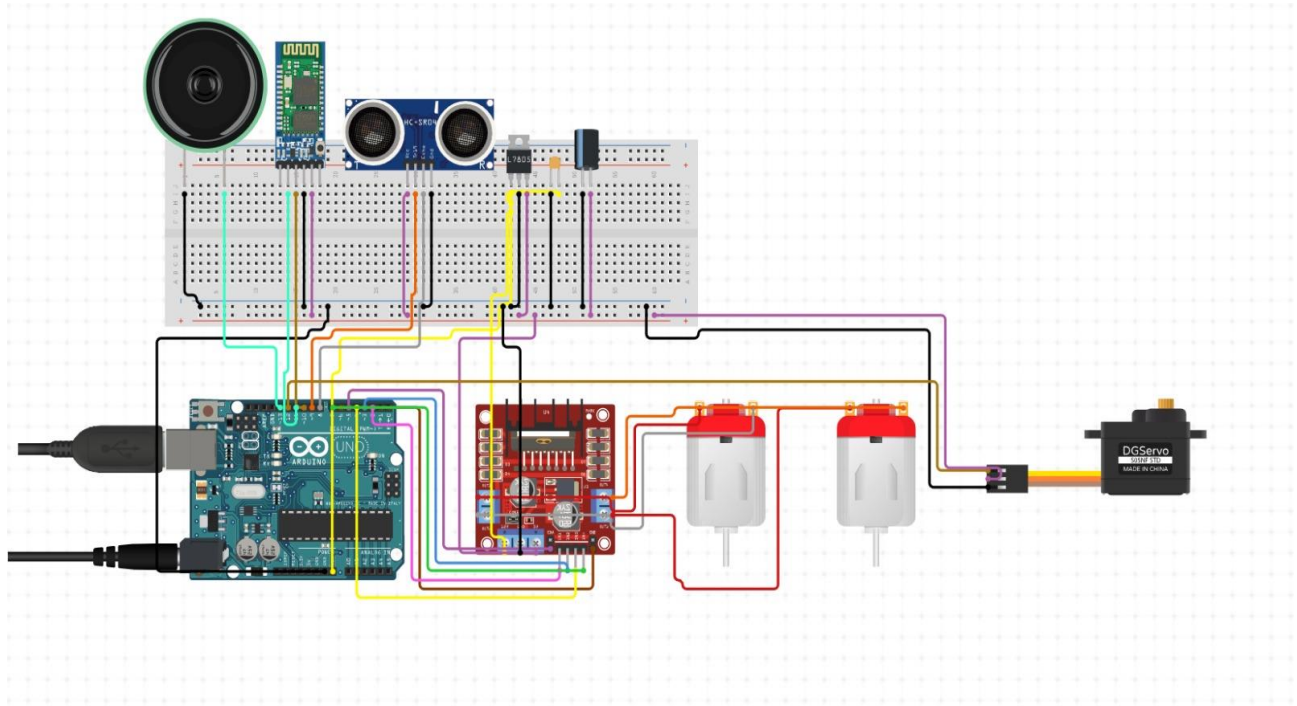
Acest robot se foloseste de o placuta Aduino Uno pentru a controla actiunile pe care le face impreuna cu inca un set de piese: un modul Bluetooth HC-5, doua motoare DC pentru controlul rotilor, doua module de detectare a turatiei, un detector de distanta bazat pe ultrasunete, un shield pentru controlul motoarelor si un difuzor.

Dupa cum am mentionat si mai sus robotul este controlat din telefon prin Bluetooth avand de asemenea in aplicatie o consola ce permite controlarea usoara a robotului si atat ca miscari precum si ca viteza de deplasare. Alte butoane din consola au fost customizate pentru a face ca robotul sa transmita informatii prin intermediul

Bluetooth-ului catre terminalul telefonului pentru a afla distanta catre obiectele apropiate sau turatia rotilor robotului si de asemenea avem si o comanda pentru a activa difuzorul atasat robotului pentru a scoate un sunet.

De asemenea dispozitivul este programat astfel incat sa previna unele actiuni care ar duce la deteriorarea sa. Una din aceste functii este oprirea deplasarii in fata daca cumva exista un obstacol la mai putin de 10 cm de el detectat cu ajutorul senzorului cu ultrasunete pentru distanta. Acest aspect functioneaza si daca robotul este deja in miscare si gaseste un obiect aparut in fata sa cat si daca dorim sa pornim robotul cu un obiect deja aflat in apropierea zonei sale frontale.

2.Schema hardware + componente



3.Algoritm

```
#include <Servo.h>
#include <SoftwareSerial.h>
// Pinii motor 1
#define mpin00 5
#define mpin01 6
// Pinii motor 2
#define mpin10 3
#define mpin11 11
Servo srv;
char comanda;
char comandaMiscare;

int vitezaComanda = 100;
int vitezaCitita;

const int trigPin = 4;
const int echoPin = 10;
// defines variables
long duration;
int distance;

int encoder_pin = 2; // pulse output from the module
long int rpm; // rpm reading
volatile byte pulses; // number of pulses
long int timeold;
// number of pulses per revolution
// based on your encoder disc
long int pulsesperturn = 20;
void counter()
{
    //Update count
    pulses++;
}

SoftwareSerial mySerial(12, 13); // RX, TX
void setup() {
    // configurarea pinilor motor ca iesire, initial valoare `0
    digitalWrite(mpin00, 0);
    digitalWrite(mpin01, 0);
    digitalWrite(mpin10, 0);
    digitalWrite(mpin11, 0);
    pinMode (mpin00, OUTPUT);
    pinMode (mpin01, OUTPUT);
    pinMode (mpin10, OUTPUT);
    pinMode (mpin11, OUTPUT);
    pinMode (A0, OUTPUT);
```

```

// pin LED
pinMode(13, OUTPUT);

// set the data rate for the SoftwareSerial port
mySerial.begin(9600);
mySerial.println("Hello, world?");
Serial.begin(9600);
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT);

pinMode(encoder_pin, INPUT);
//Interrupt 0 is digital pin 2
//Triggers on Falling Edge (change from HIGH to LOW)
attachInterrupt(0, counter, FALLING);
// Initialize
pulses = 0;
rpm = 0;
timeold = 0;

}
// Functie pentru controlul unui motor
// Intrare: pinii m1 si m2, directia si viteza
void StartMotor (int m1, int m2, int forward, int speed)
{

    if (speed == 0) // oprire
    {
        digitalWrite(m1, 0);
        digitalWrite(m2, 0);
    }
    else
    {
        if (forward)
        {
            digitalWrite(m2, 0);
            analogWrite (m1, speed); // folosire PWM
        }
        else
        {
            digitalWrite(m1, 0);
            analogWrite (m2, speed);
        }
    }
}

// Functie de siguranta
// Executa oprire motoare, urmata de delay
void delayStopped(int ms)
{
    StartMotor (mpin00, mpin01, 0, 0);
    StartMotor (mpin10, mpin11, 0, 0);
    delay(ms);
}
// Utilizare servo

```

```

// Pozitionare in trei unghiuri
// La final, ramane in mijloc (90 grade)
void playWithServo(int pin)
{
    srv.attach(pin);
    srv.write(0);
    delay(1000);
    srv.write(180);
    delay(1000);
    srv.write(90);
    delay(1000);
    srv.detach();
}
void loop() {
    // Cod avertizare
    // Blink lent

    /*
    for (int i = 0; i < 10; i++)
    {
        digitalWrite(13, 1);
        delay(200);
        digitalWrite(13, 0);
        delay(200);
    }
    // Blink rapid. Scoateti cablul USB!!!!
    for (int i = 0; i < 10; i++)
    {
        digitalWrite(13, 1);
        delay(100);
        digitalWrite(13, 0);
        delay(100);
        playWithServo(8);
    }
    */
    digitalWrite(13, 1);
    // Pornirea motorului Servo

    /*
    // Acum se pornesc motoarele DC
    StartMotor (mpin00, mpin01, 0, 128);
    StartMotor (mpin10, mpin11, 0, 128);

    delay (500); // Cat timp e motorul pornit
    delayStopped(500); // Cat timp e oprit

    StartMotor (mpin00, mpin01, 1, 128);
    StartMotor (mpin10, mpin11, 1, 128);

    delay (500);
    delayStopped(500);
    StartMotor (mpin00, mpin01, 0, 128);
    StartMotor (mpin10, mpin11, 1, 128);

    delay (500);
    delayStopped(500);

```

```

StartMotor (mpin00, mpin01, 1, 128);
StartMotor (mpin10, mpin11, 0, 128);

delay (500);
delayStopped(500);
*/

if (mySerial.available()){ // Citire de pe Bluetooth, trimite la PC
comanda=mySerial.read();
}

if('0'<= comanda && comanda <='9')
{
    vitezaComanda =100+ (10 * (comanda - '0'));
}
else
{
    comandaMiscare = comanda;
}

if (millis() - timeold >= 1000) {
    //Don't process interrupts during calculations
    detachInterrupt(0);
    rpm = ((long)60 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses;
    timeold = millis();
    pulses = 0;
    Serial.print("RPM = ");
    Serial.println(rpm,DEC);
    //Restart the interrupt processing
    attachInterrupt(0, counter, FALLING);
}

Serial.print("\n Comanda: ");
Serial.print(comanda);
Serial.print("\n vitezaData: ");
Serial.print(vitezaComanda);
Serial.print("\n COmanda Miscare: ");
Serial.print(comandaMiscare);

Serial.print("RPM = ");
    Serial.println(rpm,DEC);
Serial.print("\n");

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor

```



```

if(comanda == 'K'){
mySerial.print("Distance: ");
mySerial.println(distance);
comanda = 'S';

}

if(comanda == 'M'){
tone(A0, 440, 2000);

}

/*
Serial.print("Distance: ");
Serial.println(distance);
Serial.print("\n");
*/
if(distance < 10 && comandaMiscare!='B')
{
    delayStopped(100);
}

if(comandaMiscare=='S')
{
    delayStopped(100);
}

    if(distance >10 && comandaMiscare=='F'){
        StartMotor (mpin00, mpin01, 1, vitezaComanda);
        StartMotor (mpin10, mpin11, 1, vitezaComanda);
        //delay(5000);
    }

if(comandaMiscare=='B')
{
    StartMotor (mpin00, mpin01, 0, vitezaComanda);
    StartMotor (mpin10, mpin11, 0, vitezaComanda);
    //delay(5000);
}

if(comandaMiscare=='L')
{
    StartMotor (mpin00, mpin01, 1, vitezaComanda);
    StartMotor (mpin10, mpin11, 0, vitezaComanda);
    //delay(5000);
}
if(comanda=='R')
{
    StartMotor (mpin00, mpin01, 0, vitezaComanda);
    StartMotor (mpin10, mpin11, 1, vitezaComanda);
    //delay(5000);
}

```

}

}

4.Rezultate

Robotul a fost testat de o multitudine de ori in timpul dezvoltarii acestui proiect. Toate obiectivele proiectului au fost indeplinite obtinant un mic rover ce poate fi controlat cu usurinta din telefon imediat dupa ce a a fost conectat la el. De asemenea pentru eficienta maxima roverul nostru este alimentat prin conexiunea de alimentare a placutei Arduino (5 V). Acest lucru face din acest proiect un dispozitiv interesant si util pentru diverse aplicatii.

5.Concluzie

Limbajul Arduino impreuna cu diversele sale module si placi de dezvoltare precum si cu platformele sale de cod de tip Open Source au dus la creerea unor aplicatii ingenioase si foarte practice care pot fi folosite intr-o sumedenie de domenii.

Acest aspect a facut ca atunci cand avem nevoie de un dispozitiv nou sa nu fie neaparat nevoie sa cumparam

unul gata facut ci sa il contruim si modificam dupa necesitatile noastre.

Chiar daca pana in momentul de fata acest robot este limitat ca aplicabilitate acesta se poate dezvolta pentru a dezvolta si mai multe functii si a avea o functionalitate mai larga. Cateva exemple ar fi implementarea unei camere video alaturi de senzorul de distanta pentru a vedea live miscarile robotului precum si alti senzori pentru a putea dezvolta si mai mult datele transmise de catre acest dispozitiv precum si salvarea acestora pentru o utilizare ulterioara.