

55. srečanje mladih raziskovalcev Slovenije 2021

Daljinsko Vodenje Vozila

Raziskovalno področje: Elektrotehnika, elektronika in robotika

Raziskovalna naloga

Avtor: Gregor Kepe

Mentor: Bojan Dežman

Šola: Srednja elektro-računalniška šola Maribor

Maribor, februar 2021

Kazalo vsebine

1	POVZETEK.....	3
2	UVOD.....	4
3	ZAHVALA.....	4
4	VSEBINSKI DEL.....	5
4.1	Arduino uno.....	5
4.2	Programsko okolje.....	7
4.3	HC-12 Modul.....	8
4.4	Servo motor.....	9
4.5	H-Mostič.....	10
4.6	Daljinec – krmilna ročica.....	11
5	PROGRAMIRANJE.....	13
5.1	Programiranje HC-12 komunikacijskega modula.....	13
6	ZAKLJUČEK.....	15
7	DRUŽBENA ODGOVORNOST.....	15
8	CILJI.....	15
9	VIRI.....	16
10	PRILOGE.....	17
10.1	Programska koda.....	17
10.1.1	Oddajnik (krmilna ročica)	17
10.1.2	Sprejemnik (vozilo)	18

Kazalo slik

Slika 1:	Arduino UNO (vir. Arduino.cc).....	5
Slika 2:	PWM Graf (vir. Google slike).....	6
Slika 3:	Arduino IDE (vir. Google slike).....	7
Slika 4:	HC-12 Modul (vir. Google slike).....	8
Slika 5:	Servo motor (vir. Google slike).....	9
Slika 6:	H-Mostič BTS7960 (vir. Google slike).....	10
Slika 7:	Krmilna ročica (vir. Avtorji naloge).....	11
Slika 8:	Potenciometer (vir. https://en.wikipedia.org/wiki/Potentiometer).....	12

1 POVZETEK

V nalogi smo raziskovali daljinsko vodenje vozila. Za krmiljenje smo uporabili razvojno ploščo arduino tako na krmilni ročici kot na vozilu. Za brezžično komunikacijo med ročico in vozilom pa smo uporabili modul HC-12, ki omogoča oddajanje in sprejemanje radijskega signala na frekvenci 433 MHz do razdalje 1,8 km. Enosmerni motor na vozilu smo krmilili s pomočjo H-Mostiča BTS7960, ki s pomočjo pulzne modulacije omogoča krmiljenje hitrosti vrtenja elektromotorja in spremembo smeri vrtenja.

Za krmiljenje smeri vožnje vozila smo uporabili servo motor, ki preko mehanizma obrača sprednja kolesa levo in desno. Program za krmilno ročico in za krmiljenje vozila je napisan v programskem jeziku C++ v okolju Arduino IDE.

Ang.

In this research paper, we explored the remote vehicle control. We steered the vehicle and the remote with Arduino control panel. We used HC-12 module for wireless data transformation from remote control to the vehicle. HC-12 module enables transmitting and receiving of the radio signal at frequency 433 MHz up to distance of 1,8 km. We steered DC motor on the vehicle with the BTS7960 H-bridge. We steer speed and direction of the motor with pulse width modulation.

With Servo motor we steer driving direction of vehicle. Servo motor is turning the front wheels left and right with the help of mechanism. Programming language is C++ and programming environment is Arduino IDE.

2 UVOD

Najprej smo sestavili podlago za vse module in kolesa. Za zadnji dve kolesi smo vzeli kolesa, ki so imela motor, za sprednji dve kolesi pa smo sestavili mrhanizem, da so se kolesa lahko premikala levo in desno. Nato smo na vozilo namestili arduino mikrokrmilnik in H-Mostič in ga povezali z motorjem, ki poganja zadnji dve kolesi. Na mikrokrmilnik arduino-uno smo priključili modul HC-12 za sprejemanje radijskega signala iz krmilne ročice. Za obračanje sprednjih koles smo namestili servo motor, ki nam omogoča, da se kolesa obračajo v tisto smer, v katero premaknemo krmilno ročico. Za napajanje elektronike in motorjev na vozilu smo uporabili Litij-polimerno baterijo naperosti 12 V.

Krmilno ročico smo sestavili iz ohišja, v katerega smo namestili mikrokrmilnik arduino-uno, dvojni potenciometer z ročico, ki se vrača v nevtralni poližaj ter modul HC-12 za oddajanje radijskega signala. Za napajanje krmilne ročice smo uporabili Litij-polimerno baterijo naperosti 7,4 V.

3 ZAHVALA

Zahvaljujem se mentorju za vso pomoč pri delu raziskovalne naloge. Hvala tudi staršem, ki so me podpirali pri raziskovanju.

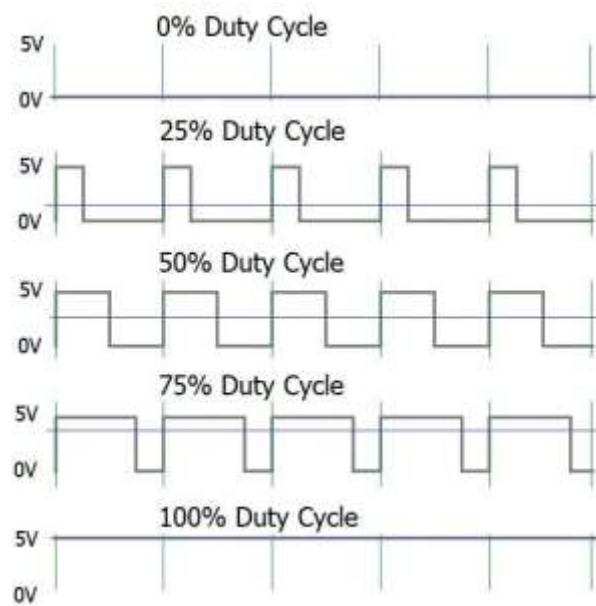
4 VSEBINSKI DEL

4.1 Arduino uno



Slika 1: Arduino UNO (vir. Arduino.cc)

Arduino uno je razvojna plošča, ki vsebuje mikrokontroler ATmega 328P. Vsebuje digitalne vhode/izhode, od tega jih je 6 z možnostjo PWM. Pulse-width modulation, ali po slovensko pulzno širinska modulacija pomeni, da lahko s širino pulza krmilimo na primer hitrost nekega motorja in s tem ustvarjamo delovni cikel. Daljši je pulz, hitreje bo motor deloval in obratno, kot kaže PWM graf (Slika 2). Pulzno širinska modulacija deluje s frekvenco 500 Hz.



Slika 2: PWM Graf (vir. Google slike)

Arduino uno vsebuje še analogne vhode, priključek za napetostno napajanje Vin, napetostna izhoda 5 V in 3,3 V ter tri priključke za maso (GND). Za mikrokrmilnik Arduino uno smo se odločili, ker je zelo razširjen in omogoča priklop številnih dodatnih modulov ter je cenovno ugoden . Za naše potrebe je tudi dovolj zmogljiv. Na internetu najdemo tudi vso pomoč za programiranje modula v programskem jeziku C++.

4.2 Programsko okolje

Programsko okolje, ki nam je na voljo za programiranje, je Arduino IDE. Programira se v programskem jeziku C++. Programsko okolje se uporablja za pisanje programov in nalaganje programov na mikrokrmilno ploščo Arduino, tako da jo najprej povežemo z računalnikom preko USB priključka. Programsko okolje je narejeno na osnovi Java programskega jezika. Je odprtokodno in brezplačno.

The image shows a screenshot of the Arduino IDE software. The window title is "Blink | Arduino 1.8.5". The main editor area contains the following C++ code for a blink program:

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

The bottom status bar shows "12" and "Arduino/Genuino (no on COM)".

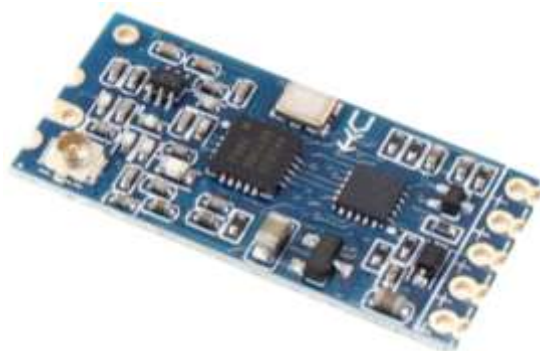
Slika 3: Arduino IDE (vir. Google slike)

4.3 HC-12 Modul

Komunikacijo med vozilom in daljincem sem želel vzpostaviti brezžično. Zato sem potreboval brezžični komunikacijski modul za oddajanje in sprejemanje radijskega signala. Uporabil sem modul HC-12, ki omogoča brezžični prenos informacij med več arduino ploščami, na razdalji do 1,8 km. Krmilimo ga s arduino mikrokrmilnikom. Lahko ga uporabljamo kot oddajnik in kot sprejemnik, kar programsko določimo v nastavitvah. Njegovo frekvenčno območje je med 433 in 473 MHz. Ima 6 pinov:

- VCC - pin za napajanje,
- RX - pin za sprejemanje podatkov,
- TX - pin za oddajanje podatkov,
- GND - pin za maso,
- SET - pin za dodatne nastavitve
- ANT - antena.

Z HC-12 modulom smo oddajali signal od daljinca do vozila. Za lažje programiranje HC-12 modula smo uporabili «SoftwareSerial.h» knjižnico, ki smo jo vključili v programsko kodo.



Slika 4: HC-12 Modul (vir. Google slike)

4.4 Servo motor

MG 996R je kovinski zobniški servo motor, sestavljen iz elektromotorja, zobniškega prenosa in elektronike. Vsebuje ročico, ki se iz nevtralnega položaja vrti za +90 stopinj in -90 stopinj. Servo ima tri žice. Rdeča in rjava sta za napajanje, oranžna pa je za signal. Servo motor krmilimo s pomočjo pulzno širinske modulacije. Po oranžni žici pošiljamo pulzni signal vsakih 20 milisekund, širina signala pa je 1,5 milisekunde za nevtralni položaj, pri širini impulza 1 milisekunde se ročica servo motorja obrne za 90 stopinj levo, pri širini impulza 2 milisekunde pa se ročica servo motorja obrne za 90 stopinj desno. Frekvenca signala za servo motor je po navadi okoli 50-60 Hz. Servo motor smo uporabili za krmiljenje smeri vozila (levo, desno). Za lažje programiranje servo motorja smo uporabili «Servo.h» knjižnico, s pomočjo katere se ročica servota obrne za željeni kot in smer.



Slika 5: Servo motor (vir. Google slike)

4.5 H-Mostič

H-Mostič BTS7960 je modul za krmiljenje hitrosti in smeri vrtenja elektromotorjev, ki ga krmilimo preko arduino mikrokrmilnika. Ima 8 krmilnih pinov s katerim krmilimo smer in hitrost vrtenja elektromotorja. Dva od tega sta pina za napajanje (Vcc (+) in masa (-)), dva sta PWM pina (R_PWM in L_PWM) vsak za svojo smer vrtenja elektromotorja, dva sta digitalna pina, ostala dva pina pa nismo rabili vezati na arduino ploščo. Zraven krmilnih pinov so še štirje pini, dva vhoda za baterijo B+ in B- in dva izhoda za motor M+ in M-.

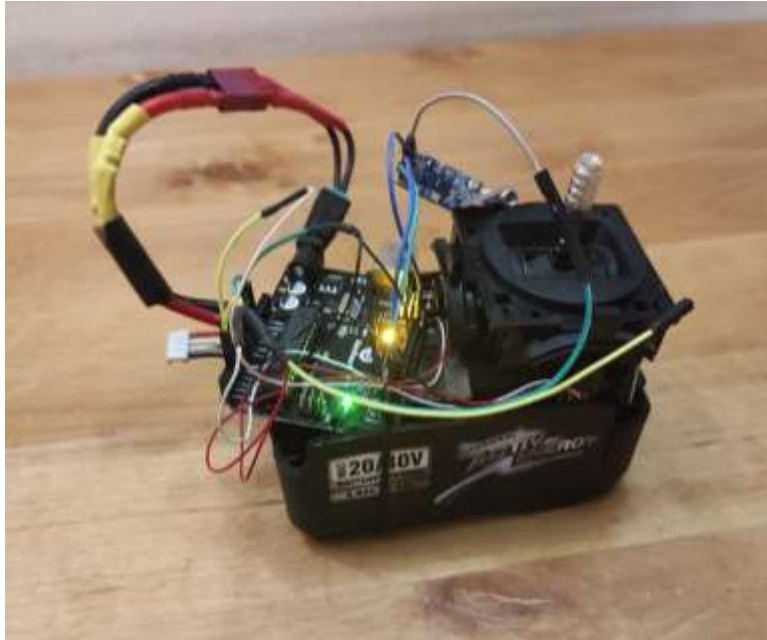
Tehnične specifikacije BTS7960 H-Mostiča:

- Vhodna napetost: 6V – 27V
- Model: IBT-2
- Maksimalen tok: 43A
- Zmožnost PWM do 25 kHz
- Način nadzora: PWM ali nivo
- Delovni cikel: 0% - 100%
- Dimenzije: 50mm x 50mm x 43mm
- Teža: 66g



Slika 6: H-Mostič BTS7960 (vir. Google slike)

4.6 Daljinec – krmilna ročica



Slika 7: Krmilna ročica (vir. Avtorji naloge)

Daljinec je sestavljen iz dveh potenciometrov z ročico, ohišja, mikrokrmilnika arduino-uno in litij-polimerne baterije 7,4 V. Prvi potenciometer z ročico premikamo gor- dol za vožnjo namrej – nazaj, drugega pa premikamo levo-desno za vožnjo vozila levo - desno.

Vsak potenciometer ima 3 žice, ki jih priključimo na arduino uno mikrokrmilno ploščo. Prvo (črno) priključimo na maso, drugo (rdečo) priključimo na 5 voltov in tretjo (belo) na enega izmed analognih pinov na arduino uno plošči. S premikanjem gredi poteciometra speminjamo upornost na potenciometru in s tem napetost oz. analogno vrednost na arduino uno plošči. Če je gred obrnjena do konca v eno smer, potem je na analognem pinu 0 voltov, in preberemo vrednost 0. Če je gred obrnjena do konca v drugo smer, potem je na analognem pinu 5 voltov, in preberemo vrednost 1024. Funkcija `analogRead ()` vrne število med 0 in 1024, ki je sorazmerno z napetostjo, ki je na analognem pinu.

Analogno vrednost za vsak potenciometer smo definirali kot spremenljivko. Prvi vrednosti smo spremenili obseg iz 0 do 1024 na 0 do 512. Ker je ročica, ko je ne premikamo, na

sredini, to pomeni, da je na potenciometru vrednost 256. Zato smo v kodi napisali, da če je vrednost prve spremenljivke večja od 256, se bo vozilo vozilo naprej, če pa je vrednost manjša od 256 pa se bo vozilo nazaj.



Slika 8: Potenciometer (vir. <https://en.wikipedia.org/wiki/Potentiometer>)

5 PROGRAMIRANJE

5.1 Programiranje HC-12 komunikacijskega modula

Najprej moramo, za lažje programiranje modula, v kodo vključiti knjižnico SoftwareSerial.h.

```
#include <SoftwareSerial.h> // vključena je <SoftwareSerial.h> knjižnica za HC-12 modul
```

Nato smo definirali pine za HC-12 modul

```
SoftwareSerial HC12(2, 3); // HC-12 pin (2) za oddajnik, HC-12 pin (3) za sprejemnik
```

Krmilna ročica je sestavljena iz dveh potenciometrov, in vsak potenciometer ima 3 žice. Dve za napajanje in tretjo za analogno vrednost, ki jo pošljemo iz potenciometra do arduino uno plošče. Ti dve analogni vrednosti smo priključili na pina A2 in A3. S funkcijo `pinMode()` smo ta dva pina definirali kot (analogna) vhoda.

Vsako vrednost smo definirali pod svojo spremenljivko.

```
pinMode(A2, INPUT); // smer (levo-desno)
pinMode(A3, INPUT); // hitrost (naprej-nazaj)

int val = map(analogRead(A3), 0, 1024, 512, 0); // nastavili smo obseg analogne vrednosti A3 iz 0 do 1024 na 0 do 512
smer = (analogRead(A2));
```

Spremenljivko smer smo razdelili na 9 delov

```
if (smer<=112) val=val+1000;
if (smer>112&& smer<212) val=val+2000;
if (smer>212&& smer<312) val=val+3000;
if (smer>312&& smer<412) val=val+4000;
if (smer>412&& smer<512) val=val+5000;
if (smer>512&& smer<624) val=val+6000;
if (smer>624&& smer<724) val=val+7000;
if (smer>724&& smer<824) val=val+8000;
if (smer>824&& smer<1024) val=val+9000;
```

Vrednost iz daljinca smo poslali do vozila in do serijskega monitorja (vrednost, ki jo je poslal HC-12 se nam izpiše na monitorju)

```
while (HC12.available()) { // Če ima HC-12 podatke
  incomingByte = HC12.read(); // Shrani vsak prihajajoči karakter HC-12
  str += incomingByte; // Dodaja vsak prihajajoči karakter in tvori string
}

hitro = str.toInt(); // pretvori string v integer število
```

V programu za vozilo smo napisali, da prihajajočo vrednost sprejema karakter po karakter in jih tvori v string. Nato ta string pretvori v število s `str.toInt()` funkcijo.

```
while (HC12.available()) { // Če ima HC-12 podatke
  Serial.write(HC12.read()); // Pošlje podatke serijskemu monitorju
}

if(millis() > cas){
  cas=millis()+200;
  Serial.println(val); // izpiše na monitor
  HC12.print(val); // HC12 pošlje vrednost daljinsko
  HC12.println();
  delay(200);
}
```

6 ZAKLJUČEK

Spoznali smo brezžično komunikacijo med vozilom in daljincem s pomočjo HC-12 modula in tudi kako se programira v arduino okolju. Najprej smo napisali program za vsak modul in ga stestirali, potem pa smo vse komponente združili v eno vozilo in napisali program, da je vse delovalo tako kot smo si na začetku zamislili. In zdaj deluje brezžična komunikacija med krmilno ročico in vozilom. To nam je vzelo veliko časa, a ni nam žal časa ki smo ga v to vse skupaj vložili saj smo se naučili veliko koristnega.

7 DRUŽBENA ODGOVORNOST

V današnjih časih je veliko tehnološko naprednih naprav. Naše vozilo je napredna rešitev saj nam je lahko v pomoč kadar sumimo nevarnost v nekem prostoru oz. stavbi.

Na primer: na vozilo namestimo senzor vnetljivih plinov, da preverimo če je v opazovalnem prostoru možnost nastanka eksplozije, ali na vozilo namestimo katerikoli senzor, da zaznava različne nevarnosti v prostoru.

8 CILJI

Naš cilj je bil izdelati vozilo, ki ga bomo brezžično upravljali s krmilno ročico preko radijskih valov in izdelati programsko opremo, ki bo to omogočila. Naš cilj smo uresničili s pomožno zgoraj navedenih modulov in razvili programsko opremo, ki odlično upravlja naše vozilo preko radijskih valov.

9 VIRI

Arduino UNO, <https://www.arduino.cc/en/Main/ArduinoBoardUnoSMD> (26. februar 2021)

PWM, <https://www.arduino.cc/en/pmwiki.php?n=Tutorial/SecretsOfArduinoPWM> (26. februar 2021)

Arduino IDE, https://en.wikipedia.org/wiki/Arduino_IDE (26. februar 2021)

HC-12 Modul, <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/> (26. februar 2021)

Servo motor, <https://components101.com/motors/mg996r-servo-motor-datasheet> (26. februar 2021)

H-mostič, <https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%20Driver.pdf> (26. februar 2021)

Potenciometer, <https://www.arduino.cc/en/tutorial/potentiometer> (26. februar 2021)

10 PRILOGE

10.1 Programska koda

10.1.1 Oddajnik (krmilna ročica)

```
#include <SoftwareSerial.h> // vključena je <SoftwareSerial.h> knjižnica za HC-12 modul

unsigned long cas;
int smer;

SoftwareSerial HC12(2, 3); // HC-12 pin za oddajnik, HC-12 pin za sprejemnik
void setup() {
  Serial.begin(9600); // Serial port to computer
  HC12.begin(9600); // Serial port to HC12
  pinMode(A2, INPUT); // smer (levo-desno)
  pinMode(A3, INPUT); // hitrost (naprej-nazaj)
}
void loop() {
  int val = map(analogRead(A3),0,1024,512,0); // nastavili smo obseg analogne vrednosti A3 iz 0 do 1024 na 0 do 512
  smer = (analogRead(A2));
  if(smer<=112)val=val+1000;
  if(smer>112&&smer<212)val=val+2000;
  if(smer>212&&smer<312)val=val+3000;
  if(smer>312&&smer<412)val=val+4000;
  if(smer>412&&smer<512)val=val+5000;
  if(smer>512&&smer<624)val=val+6000;
  if(smer>624&&smer<724)val=val+7000;
  if(smer>724&&smer<824)val=val+8000;
  if(smer>924&&smer<1024)val=val+9000;

  while (HC12.available()) { // Če ima HC-12 podatke
    Serial.write(HC12.read()); // Pošlje podatke serijskemu monitorju
  }

  while (HC12.available()) { // Če ima HC-12 podatke
    Serial.write(HC12.read()); // Pošlje podatke serijskemu monitorju
  }

  if(millis() > cas){
    cas=millis()+200;
    Serial.println(val); // izpiše na monitor
    HC12.print(val); // HC12 pošlje vrednost daljinsko
    HC12.println();
    delay(200);
  }
}
```

10.1.2 Sprejemnik (vozilo)

```
#include <SoftwareSerial.h> // vključena je <SoftwareSerial.h> knjižnica za HC-12 modul
#include <Servo.h> // vključena je <Servo.h> knjižnica za servo motor

unsigned long cas;
String kom;
char incomingByte;
String str = "";
char val;
int smer;
int hitro;
int j=0;
int pvm=0;
int pvm1=0;
float km=0.6;
SoftwareSerial HC12(2, 3); // HC-12 pin za oddajnik, HC-12 pin za sprejemnik
Servo myservo;
void setup() {
  Serial.begin(9600); // Serial port to computer
  HC12.begin(9600); // Serial port to HC12
  myservo.attach(9); //
}
void loop() {
  while (HC12.available()) { // Če ima HC-12 podatke
    incomingByte = HC12.read(); // Shrani vsak prihajajoči karakter HC-12
    str += incomingByte; // Dodaja vsak prihajajoči karakter in tvori string
  }
  //Serial.println(str); // pretvori string v integer število
  hitro = str.toInt();
  if (hitro > 20)cas= millis();
  if (hitro > 20) { Serial.println(hitro); }
  if (hitro <= 20) { Serial.print("napaka pri sprejemu, hitro = ");
    Serial.println(hitro);
  }
  str = "";

  if(hitro > 2011)
if (hitro < 1255 && hitro>1000) {
  kom = "nazaj";
  pvm = 1255-hitro;
  smer = 60;
  myservo.write(smer);
  Serial.println(pvm);
}

  if (hitro >= 1255 && hitro<2000) {
  kom = "naprej";
  smer = 60;
  pvm = hitro-1255;
  myservo.write(smer);
  Serial.println(pvm);
}

  if (hitro >= 2255 && hitro<3000) {
  kom = "naprej";
  smer = 70;
  pvm = hitro-2255;
  myservo.write(smer);
  Serial.println(pvm);
}

  if (hitro < 2255 && hitro>2000) {
  kom = "nazaj";
  pvm = 2255-hitro;
  smer = 70;
  myservo.write(smer);
  Serial.println(pvm);
}

  if (hitro >= 3255 && hitro<4000) {
  kom = "naprej";
  smer = 80;
  pvm = hitro-3255;
  myservo.write(smer);
  Serial.println(pvm);
}

  if (hitro < 3255 && hitro>3000) {
  kom = "nazaj";
  pvm = 3255-hitro;
  smer = 80;
  myservo.write(smer);
  Serial.println(pvm);
}
```

```

if (hitro >= 4255 && hitro<5000) {
kom = "naprej";
smer = 85;
pvm = hitro-4255;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro < 4255 && hitro>4000) {
kom = "nazaj";
pvm = 4255-hitro;
smer = 85;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro >= 5255 && hitro<6000) {
kom = "naprej";
smer = 90;
pvm = hitro-5255;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro < 5255 && hitro>5000) {
kom = "nazaj";
pvm = 5255-hitro;
smer = 90;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro >= 6255 && hitro<7000) {
kom = "naprej";
smer = 95;
pvm = hitro-6255;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro < 6255 && hitro>6000) {
kom = "nazaj";
pvm = 6255-hitro;
smer = 95;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro >= 7255 && hitro<8000) {
kom = "naprej";
smer = 100;
pvm = hitro-7255;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro < 7255 && hitro>7000) {
kom = "nazaj";
pvm = 7255-hitro;
smer = 100;
myservo.write(smer);
Serial.println(pvm);
}

if (hitro >= 8255 && hitro<9000) {
kom = "naprej";
smer = 110;
pvm = hitro-8255;
myservo.write(smer);
Serial.println(pvm);
}

```

```

if (hitro < 9255 && hitro>9000) {
  kom = "naza";
  pvm = 9255-hitro;
  smer = 110;
  myservo.write(smer);
  Serial.println(pvm);
}

if (hitro >= 9255) {
  kom = "napre";
  smer = 120;
  pvm = hitro-9255;
  myservo.write(smer);
  Serial.println(pvm);
}

if (hitro < 9255 && hitro>9000) {
  kom = "naza";
  pvm = 9255-hitro;
  smer = 120;
  myservo.write(smer);
  Serial.println(pvm);
}

pvm1 = pvm * ko;

if(kom == "napre") {
  analogWrite(10, pvm1);
  digitalWrite(11, LOW);
  delay(50);
  myservo.write(smer);
}

if(kom == "naza") {
  analogWrite(11, pvm1);
  digitalWrite(10, LOW);
  delay(50);
  myservo.write(smer);
}

  Serial.print("pvm");
  Serial.println(pvm);
}

delay(200);

if (millis()-cas > 900) {
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
}
kom = "";
}

```