

# PROJEKTOVANJE SISTEMA NA ČIPU

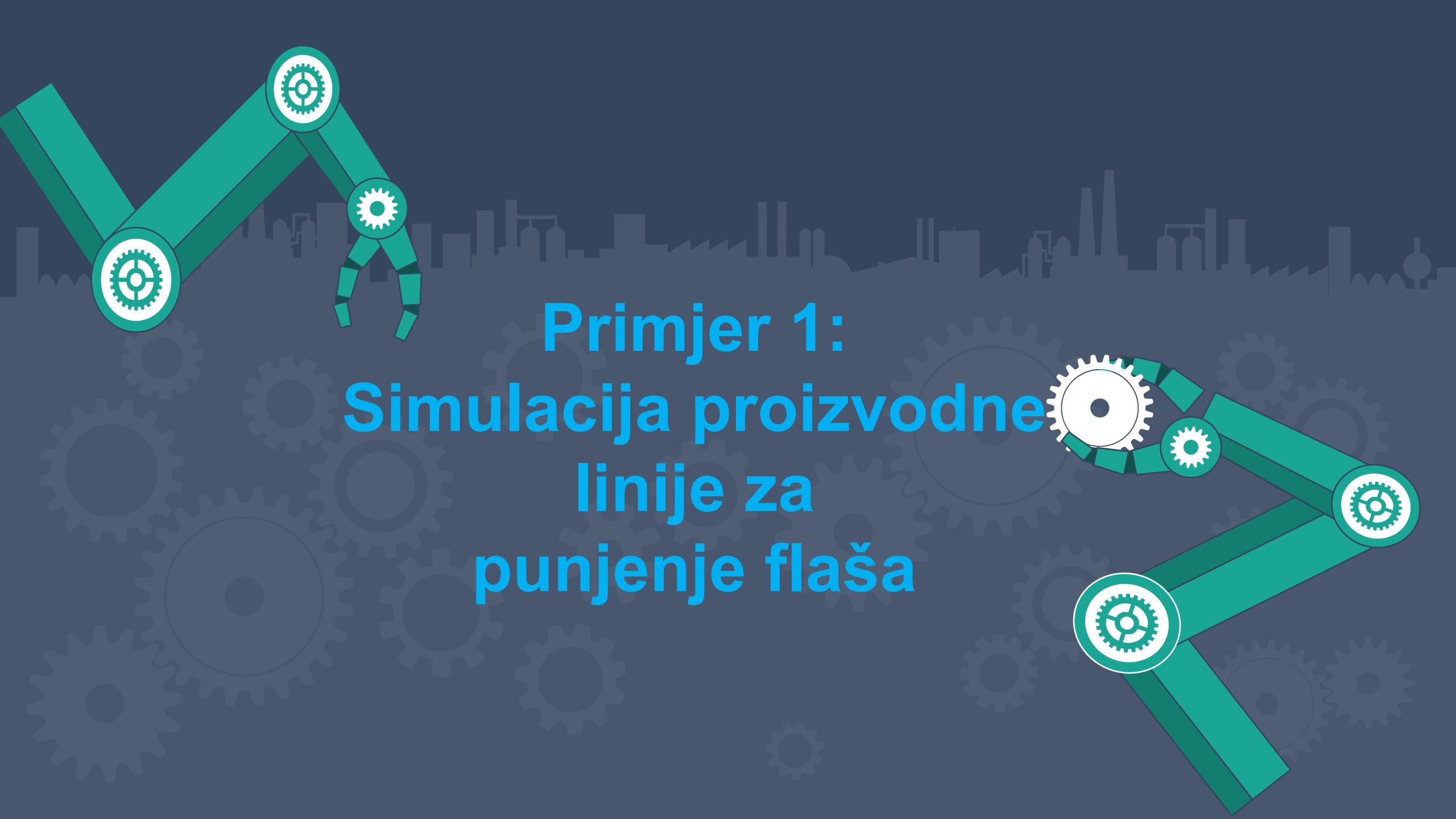
**Eksperimentalni  
dizajn SoC i FPGA  
baziranih sistema**

prof. dr. Lejla Banjanović-  
Mehmedović

Industry 4.

# Sadržaj izlaganja

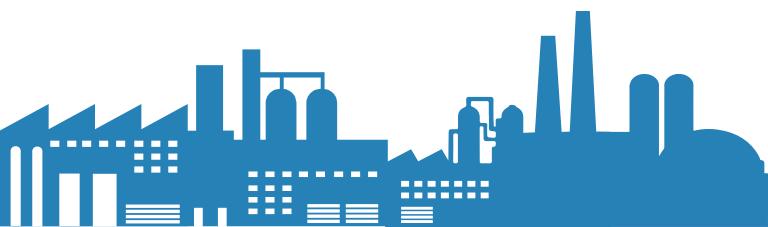
- Eksperimentalni dizajn FPGA i SoC baziranih sistema:
  - Projektovanje proizvodne linije za punjenje flaša
  - Sortiranje predmeta na pokretnoj traci na osnovu oblika
  - Projektovanje mobilnog robota sa manipulatorom za prikupljanje otpada
  - Automatizovani sistem za prepoznavanje i transport povrća



# Primjer 1: Simulacija proizvodne linije za punjenje flaša

# Ideja projekta

- Pokretna traka kao jedan od najstarijih i najzastupljenijih elemenata industrije.
- Savremene pokretne trake koriste se u gotovo svim oblicima industrije.
- Zbog jednostavne konstrukcije i održavanja, pokretne trake su izdržljive, ne kvare se često i troše relativno malo energije u odnosu na količinu prevezенog tereta.

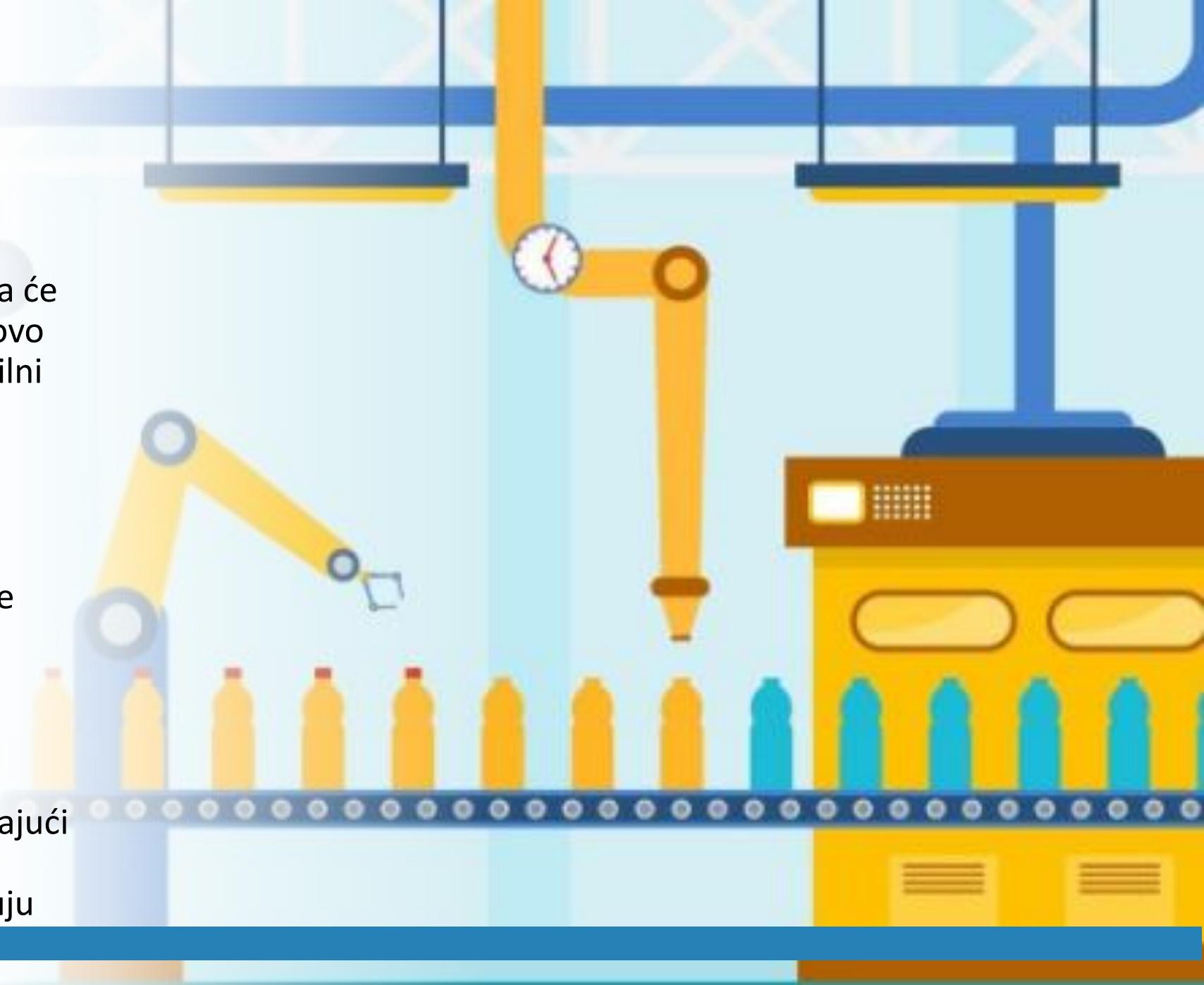


# Prednosti i primjena pokretnih traka



# Ciljevi projekta

- Dizajnirati pokretnu traku koja će služiti za punjenje flaša i njihovo zatvaranje, a u nastavku mobilni robot će napunjene flaše transportovati u skladište.
- Benefit koji donosi upotreba FPGA u odnosu na standardne mikrokontrolere u smislu upravljanja sa više aktuatora istovremeno.
- Spoznavanje kako se odgovarajući senzori povezuju na FPGA i Rasberry Pi te kako se obrađuju

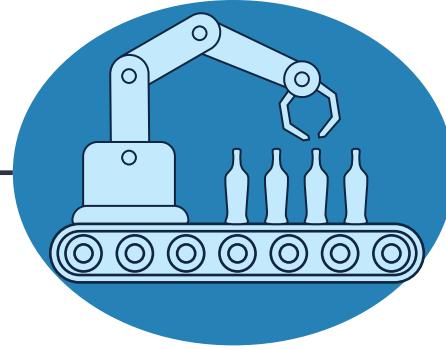


# Zadaci projekta

1

**Realizirati pokretnu traku**

- a. Obrada podataka sa senzora na pokretnoj traci
- b. Upravljanje pokretnom trakom pomoću stepper motora i odgovarajućih drajvera na osnovu informacija sa senzora
- c. Upravljanje pumpom
- d. Izrada makete koja će praktično pokazati funkcionalnosti



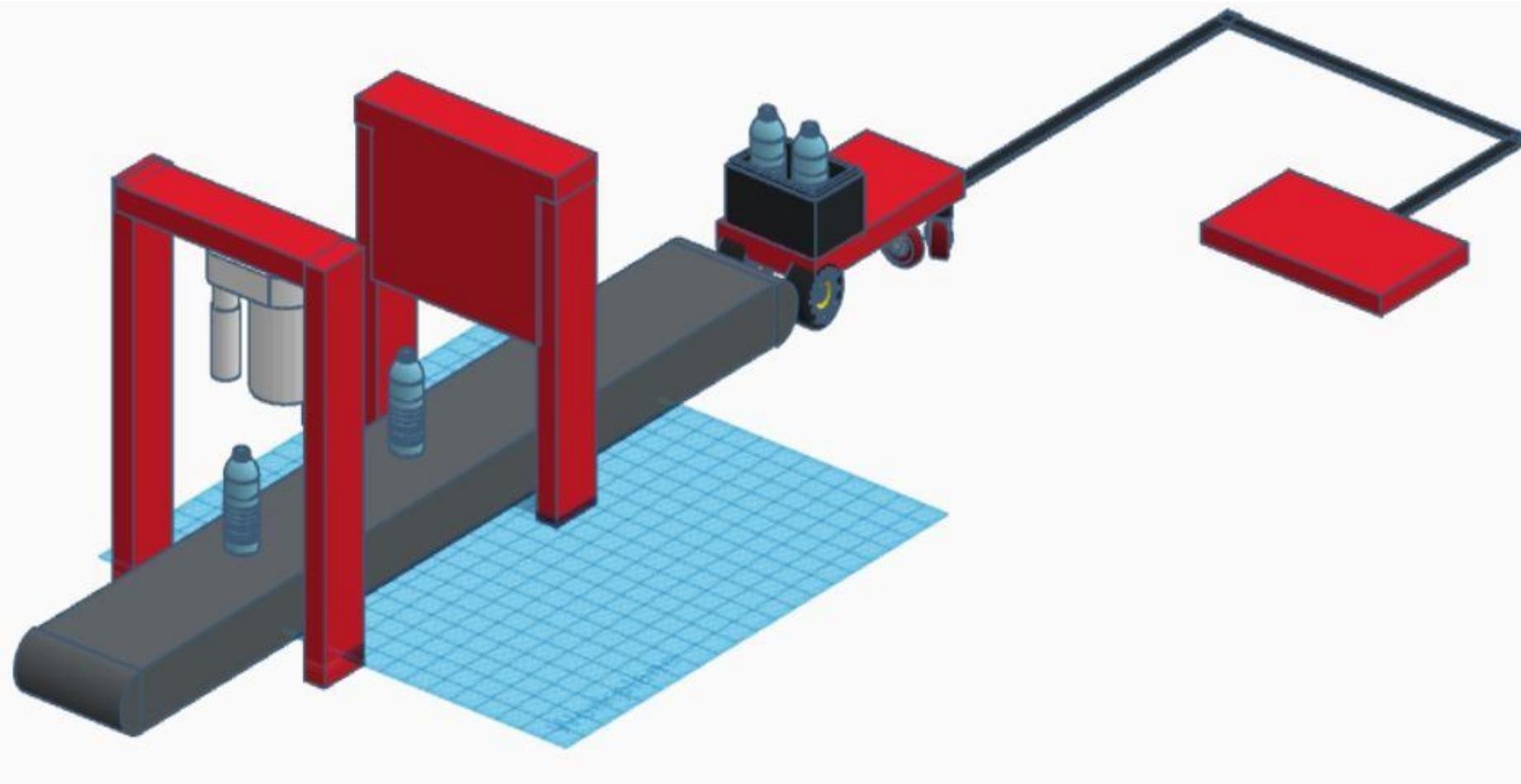
2

**Realizirati mobilnog robota**

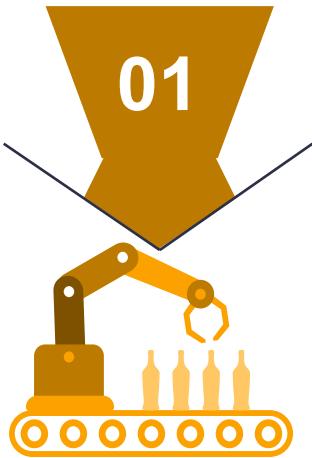
- a. Obrada podataka sa IR senzora
- b. Upravljanje mobilnim robotom na osnovu informacija sa IR senzora
- c. Slijedjenje linije
- d. Izrada makete mobilnog robota



# Opis projektnog zadatka



# Korištene komponente



FPGA



Raspberry Pi



Senzori  
(ulazi)

- IR senzor

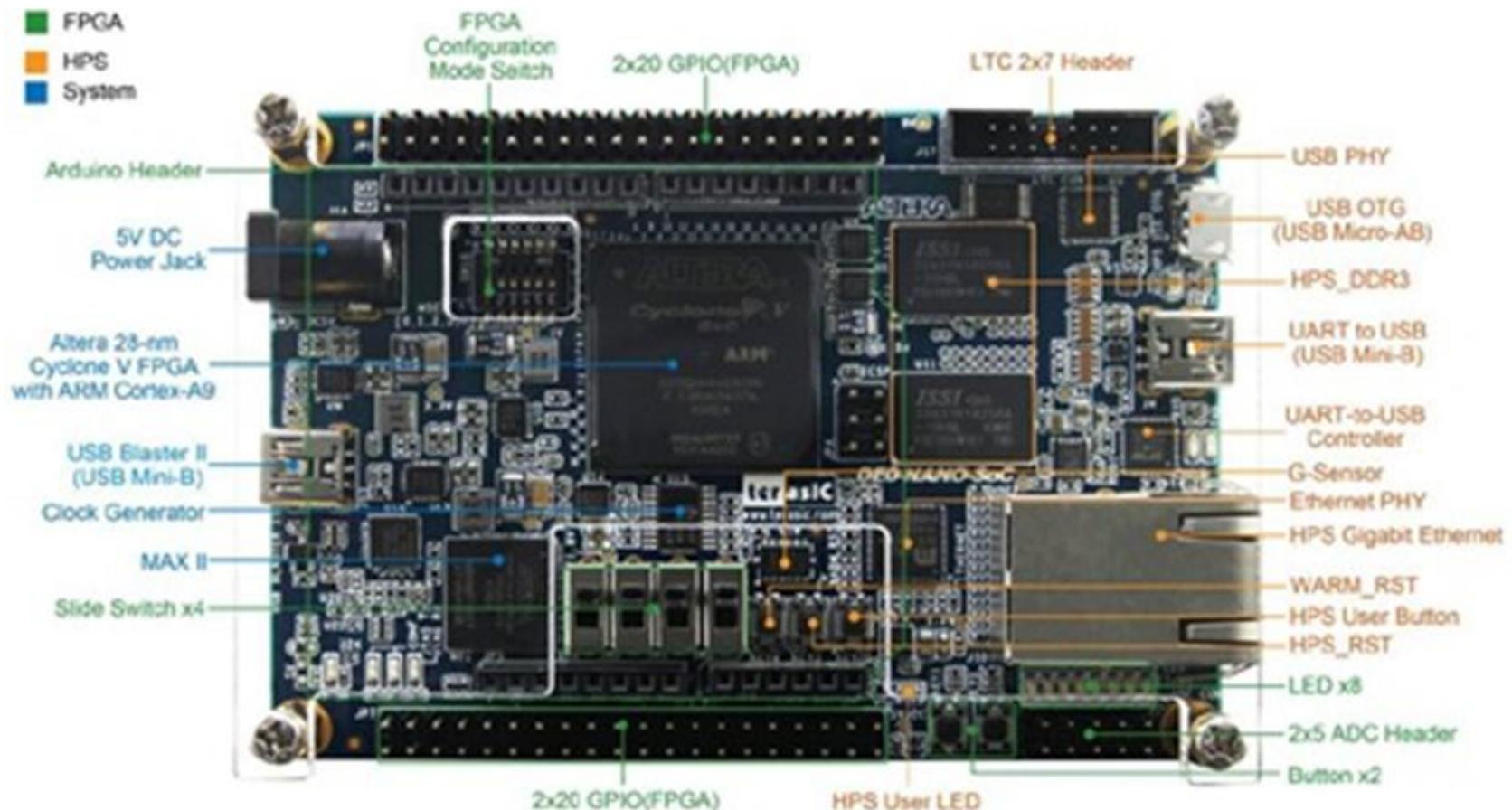


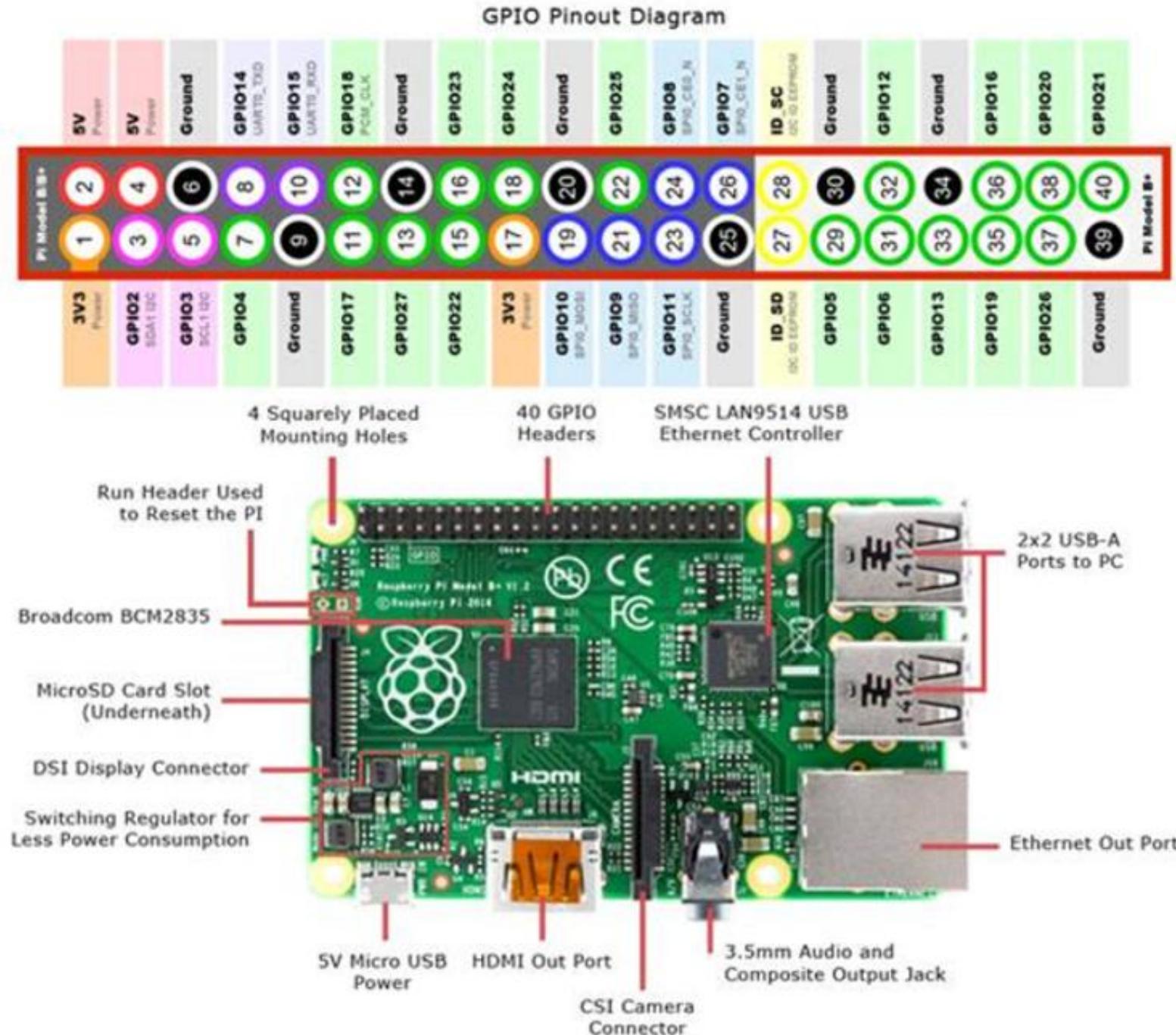
Izlazi

- DC motor sa točkom
- Stepper motor
- Servo motor
- Pumpa
- L298N motor driver
- ULN2003 stepper motor driver
- Jednokanalni relez

# FPGA-SoC platforma

- DE0-Nano-SoC razvojni komplet predstavlja robusnu platformu za dizajn hardvera izgrađenu oko Altera System-on-Chip (SoC) FPGA.
- Alterin SoC integriše ARM baziran hard procesorski sistem (HPS) koji se sastoji od procesora, perifernih uređaja i memorijskih interfejsa.





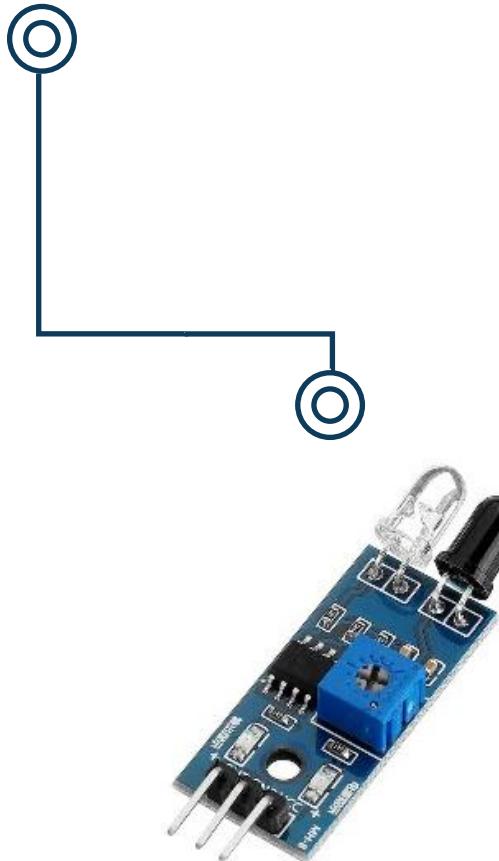
# Raspberry Pi

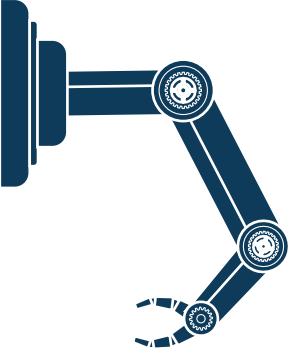
- Raspberry Pi je mali računar na jednoj ploči.
- Popularno se koristi za obradu slika/videa u realnom vremenu, aplikacije zasnovane na IoT-u i aplikacije robotike.
- Ima ARM baziran Broadcom procesor SoC zajedno sa GPU-om (Graphics Processing Unit) na čipu.
- Raspberry Pi ploča sadrži:
  - programsku memoriju (RAM),
  - procesor i grafički čip,
  - CPU,
  - GPU,
  - Ethernet port,
  - GPIO pinove,
  - Xbee socket,
  - UART,
  - konektor za izvor napajanja.

# Senzori (ulazi)

## IR senzor

- Infracrveni (IR) senzor je elektronski uređaj koji mjeri i detektuje infracrveno zračenje u svom okruženju.
- IR je nevidljiv ljudskom oku, jer je njegova talasna dužina duža od talasne dužine vidljive svetlosti.
- Sve što emituje toplotu daje infracrveno zračenje.





# Izlazi

## DC motor sa točkom



- Istosmjerni motor je elektromehanički uređaj koji električnu energiju pretvara u mehaničku odnosno pretvara istosmjernu struju u kružno kretanje.



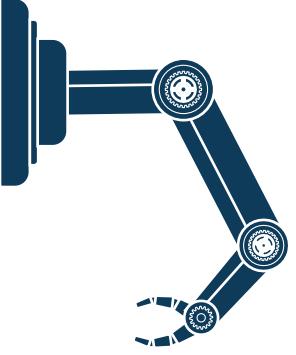
## Stepper motor

- Stepper motor je elektromehanički uređaj koji pretvara električnu snagu u mehaničku.
- To su istosmjerni motori bez četkica koji jedan obrtaj dijele u odgovarajući broj jednakih koraka.
- Pretvaraju strujne impulse u fiksne inkremente ugaonog pomjeraja - korake.
- Najčešće dolaze zajedno sa ULN2003 drajverom.

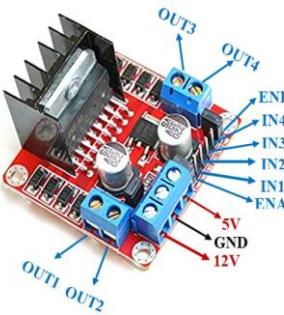
## Servo motor



- Servo motor je vrsta motora koji se može okretati s velikom preciznošću.
- Servo motor radi na principu PWM (Pulse width modulation).
- Servo motor se može rotirati od 0 do 180 stepeni, ali može ići do 210 stepeni, u zavisnosti od proizvodnje.



# Izlazi



## Pumpa

- Pumpa za vodu je uređaj sposoban generirati protok tekućine koristeći kinetičku energiju

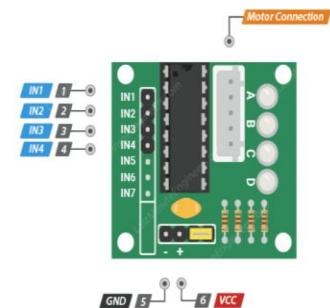


## L298N motor driver

- L298N je dvostruki H-most driver motora koji omogućava kontrolu brzine i smjera dva DC motora u isto vrijeme.

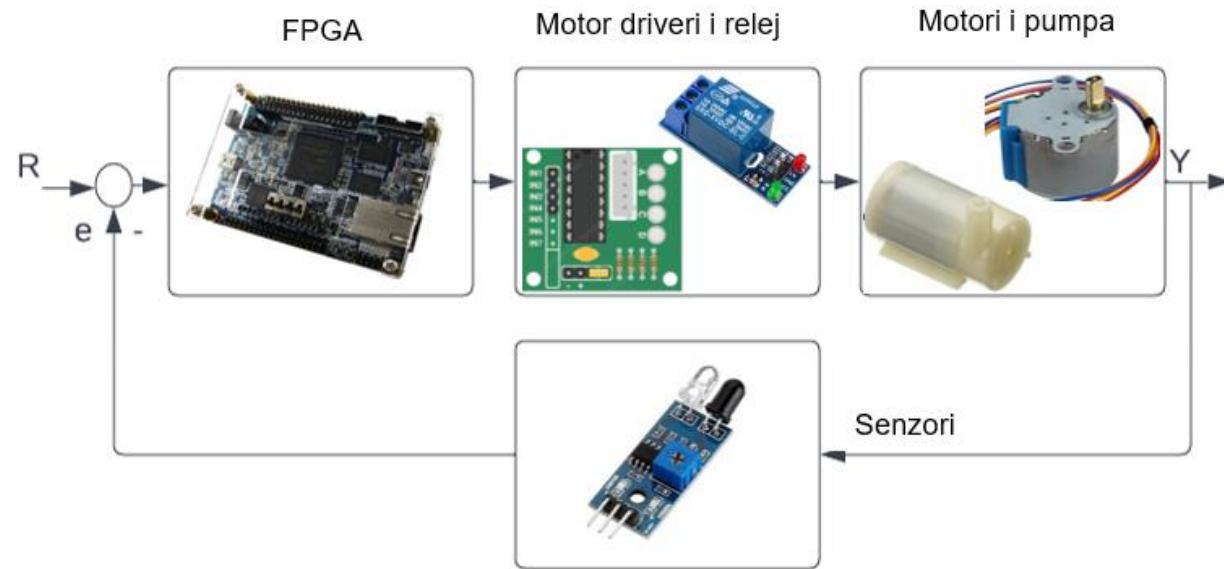
## ULN2003 stepper motor driver

- Stepper motor obično dolazi sa ULN2003 drajverskom pločom.



## Jednokanalni relej

- Jednokanalni relejni modul koristi se za uključivanje i isključivanje uređaja čija je nazivna struja veća od one koju može dati kontroler.
- Konkretno u ovom projektu se koristi za uključivanje i isključivanje vodene pumpe.



FPGA	
Ulazi	Izlazi
IR senzor	Pumpa
IR senzor	Stepper motor (za pokretanje trake)
IR senzor	Stepper motor (zatvaranje flaša)

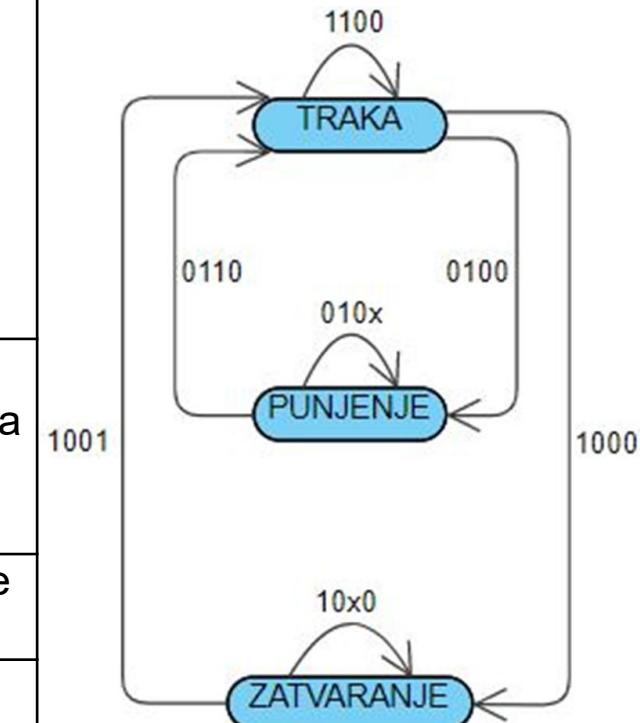
# Algoritam upravljanja

# Upravljanje pokretnom trakom

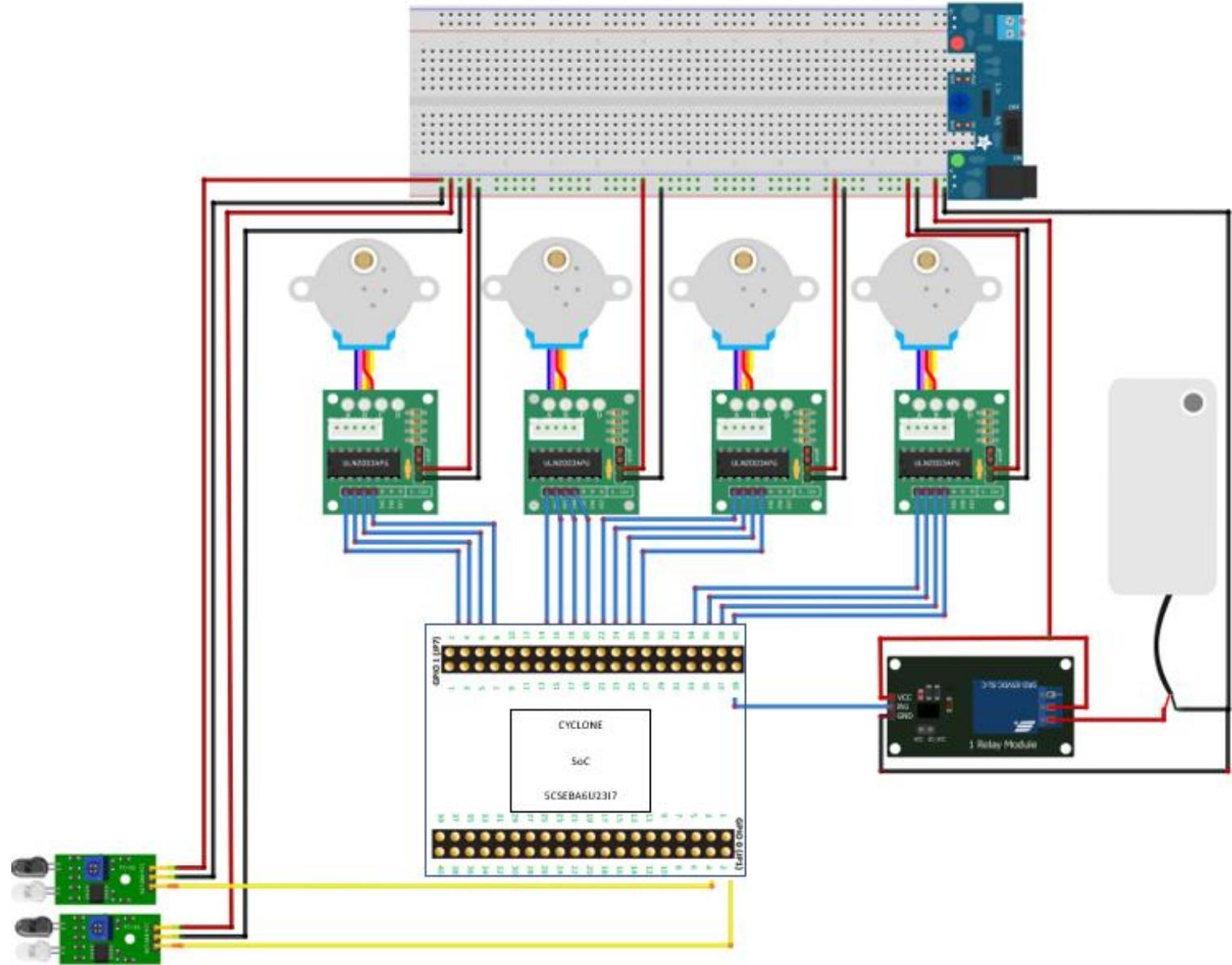
# Tabela i FSM za pokretnu traku

**FSM\_ulazi:** IR\_pumpa, IR\_zatvaranje,  
tajmer\_pumpa, tajmer\_zatvaranje

IR 1 (pumpa)	IR 2 (zatvaranje)	POKRETNA TRAKA	PUMPA	ZATVARANJE	OPIS
1	1	ON	OFF	OFF	Traka se vrti, nije detektovana flaša za punjenje ili zatvaranje
0	1	OFF	ON	OFF	Traka stoji, punjenje flaše
1	0	OFF	OFF	ON	Traka stoji, zatvaranje flaše



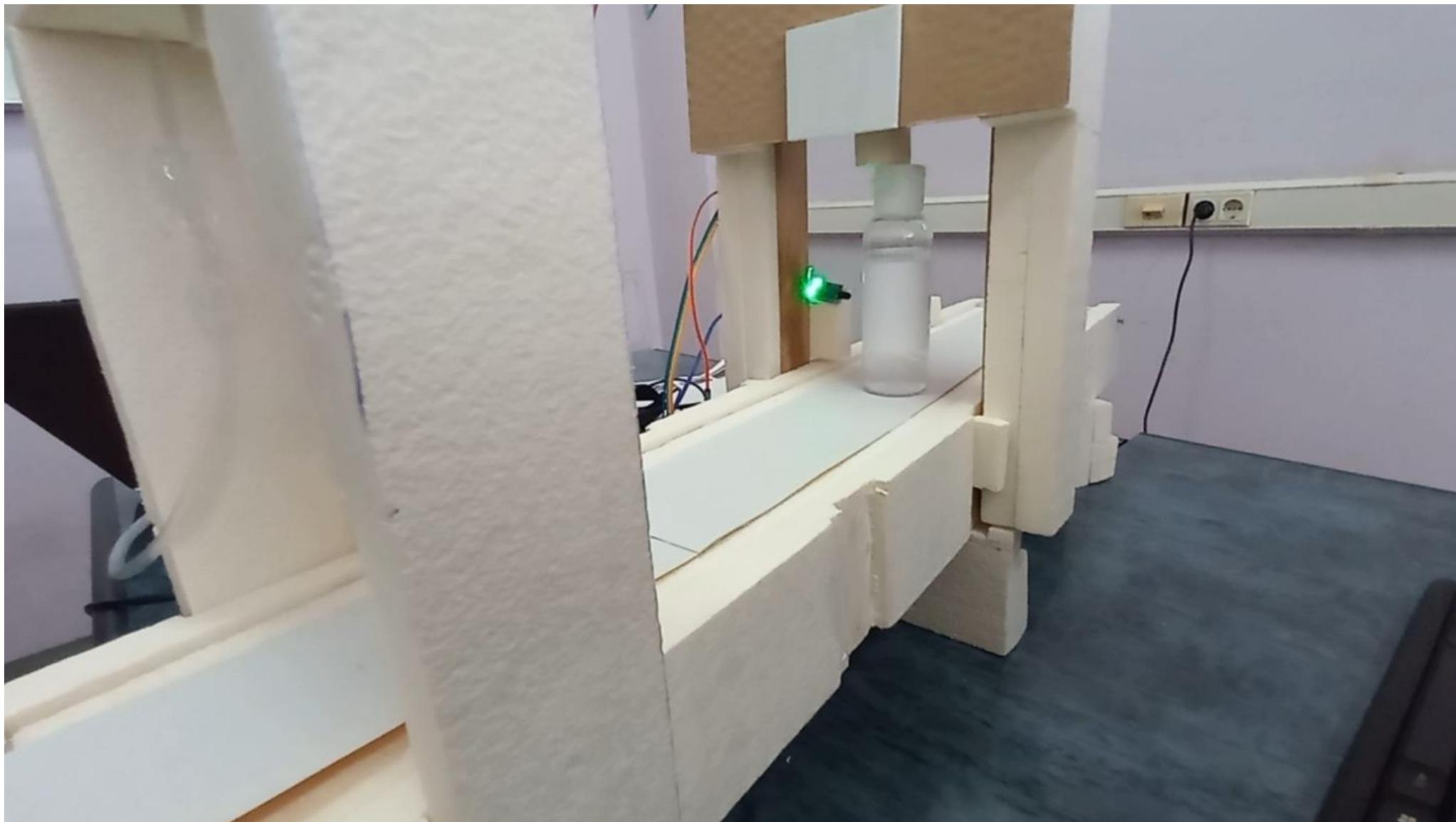
# Šema povezivanja komponenti na FPGA



# Prikaz makete pokretne trake

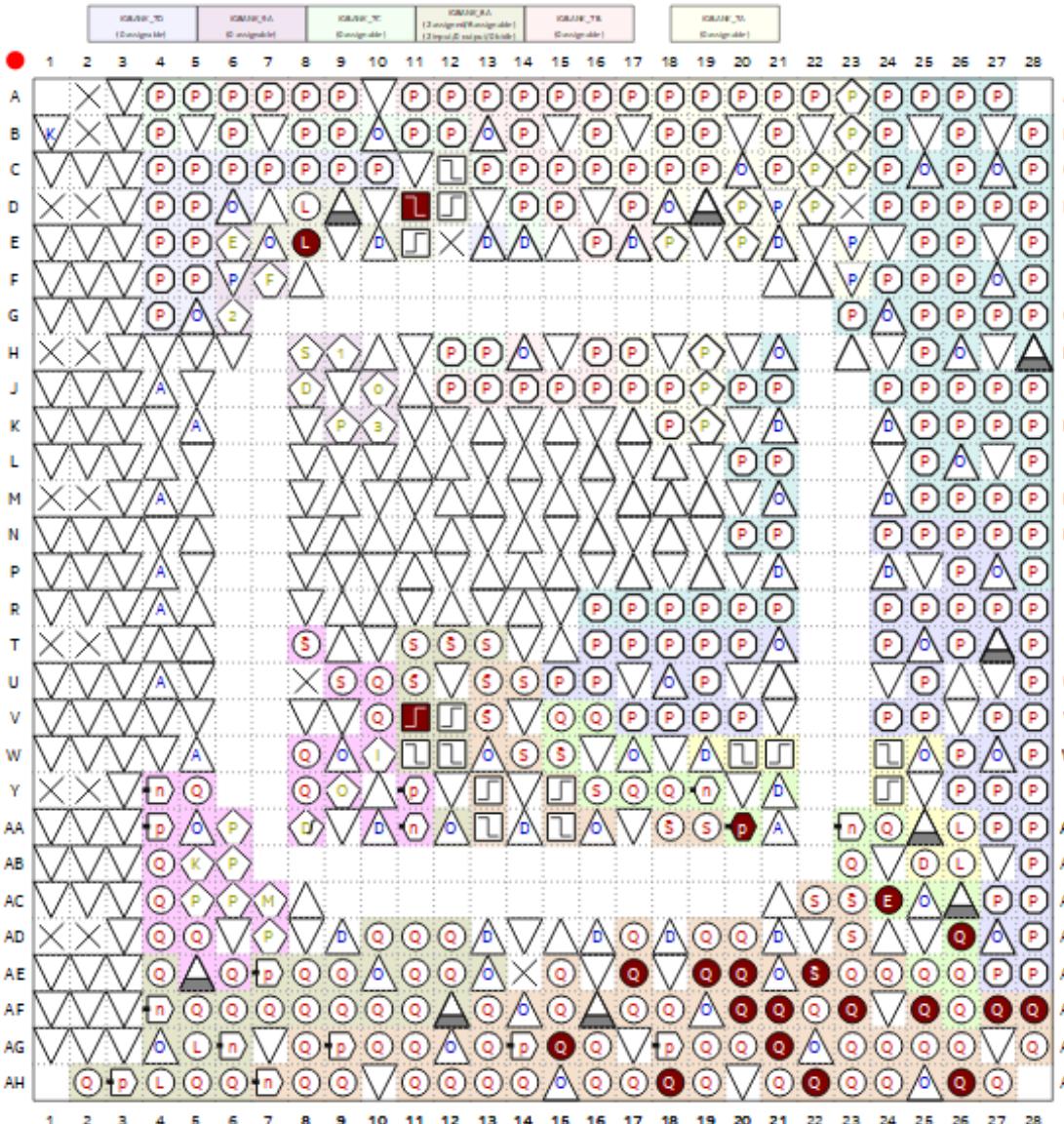


# Prikaz makete pokretne trake



# Raspored pinova na Cyclone V

Cyclone V - 5CSEBA6U23I7



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
clk	Input	PIN_V11	3B	B3B_N0	PIN_V11
dioda_punjene	Output	PIN_AG15	4A	B4A_N0	PIN_AG15
ir_pumpa	Input	PIN_D11	8A	B8A_N0	PIN_D11
ir_zatvaranje	Input	PIN_E8	8A	B8A_N0	PIN_E8
motor_traka1[3]	Output	PIN_AH22	4A	B4A_N0	PIN_AH22
motor_traka1[2]	Output	PIN_AF23	4A	B4A_N0	PIN_AF23
motor_traka1[1]	Output	PIN_AF25	4A	B4A_N0	PIN_AF25
motor_traka1[0]	Output	PIN_AH26	4A	B4A_N0	PIN_AH26
motor_traka2[3]	Output	PIN_AF21	4A	B4A_N0	PIN_AF21
motor_traka2[2]	Output	PIN_AE22	4A	B4A_N0	PIN_AE22
motor_traka2[1]	Output	PIN_AA20	5A	B5A_N0	PIN_AA20
motor_traka2[0]	Output	PIN_AG21	4A	B4A_N0	PIN_AG21
motor_zatvaranje1[3]	Output	PIN_AF27	4A	B4A_N0	PIN_AF27
motor_zatvaranje1[2]	Output	PIN_AF28	4A	B4A_N0	PIN_AF28
motor_zatvaranje1[1]	Output	PIN_AD26	5A	B5A_N0	PIN_AD26
motor_zatvaranje1[0]	Output	PIN_AC24	5A	B5A_N0	PIN_AC24
motor_zatvaranje2[3]	Output	PIN_AE17	4A	B4A_N0	PIN_AE17
motor_zatvaranje2[2]	Output	PIN_AE20	4A	B4A_N0	PIN_AE20
motor_zatvaranje2[1]	Output	PIN_AF20	4A	B4A_N0	PIN_AF20
motor_zatvaranje2[0]	Output	PIN_AH18	4A	B4A_N0	PIN_AH18
pumpa_switch	Output	PIN_AE19	4A	B4A_N0	PIN_AE19

# Kod

```

module top(clk,
    ir_zatvaranje,
    ir_pumpa,
    motor_zatvaranje1,
    motor_zatvaranje2,
    motor_traka1,
    motor_traka2,
    pumpa_switch,
    dioda_punjene,
    );
// deklaracija ulaza
input clk;
input ir_zatvaranje;
input ir_pumpa;
// deklaracija izlaza
output [3:0] motor_zatvaranje1;
output [3:0] motor_zatvaranje2;
output [3:0] motor_traka1;
output [3:0] motor_traka2;
output pumpa_switch;
output dioda_punjene;
// poziv funkcije za pokretanje trake
stepper_traka step_motor_traka (
    .clk (clk),
    .ir_zatvaranje (ir_zatvaranje),
    .ir_pumpa(ir_pumpa),
    .motor_traka1 (motor_traka1),
    .motor_traka2 (motor_traka2)
);
// poziv funkcije za punjenje flase
pumpa pumpa (
    .clk(clk),
    .ir_pumpa(ir_pumpa),
    .pumpa_switch(pumpa_switch),
    .dioda_punjene(diода_punjene)
);
// poziv funkcije za zatvaranje flase
stepper_zatvaranje step_motor_zatvaranje (
    .clk (clk),
    .ir_zatvaranje (ir_zatvaranje),
    .motor_zatvaranje1 (motor_zatvaranje1),
    .motor_zatvaranje2 (motor_zatvaranje2)
);
endmodule

//-----PUMPA-----
//-----TRAKA STOJI, PUNJENJE FLAŠE-----
else if ((ir_pumpa == 1'b0) && (StepCounter < vrijeme_punjena) && (blokiranje_pumpa == 0))
begin
    StepCounter <= StepCounter + 1'b1;
    motor_traka1 <= 4'b0000;
    motor_traka2 <= 4'b0000;
end
//-----FLAŠA NAPUNJENA-----
else if ((ir_pumpa == 1'b0) && (StepCounter == vrijeme_punjena) && (blokiranje_pumpa == 0))
begin
    blokiranje_pumpa = 1;
    StepCounter <= 31'b0;
end
//-----POMICANJE FLAŠE ISPRED IR SENZORA NAKON PUNJENJA-----
else if (blokiranje_pumpa == 1)
begin
    StepCounter <= StepCounter + 31'b10;
    if (StepCounter == step_delay)
        begin
            StepCounter <= 31'b0;
            if (angle < input_angle)
                begin
                    step1 <= step1 - 1;
                    step2 <= step2 - 1;
                    angle <= angle + 1;
                end
            //-----motor1-----
            case (step1)
                0: motor_traka1 <= 4'b1100;
                1: motor_traka1 <= 4'b0110;
                2: motor_traka1 <= 4'b0011;
                3: motor_traka1 <= 4'b1001;
            endcase
            //-----motor2-----
            case (step2)
                0: motor_traka2 <= 4'b1100;
                1: motor_traka2 <= 4'b0110;
                2: motor_traka2 <= 4'b0011;
                3: motor_traka2 <= 4'b1001;
            endcase
        end
end
//-----PUMPA UGASENA, CEKANJE DA SE FLASA POMJERI ISpred IR SENZORA-----
else if ((ir_pumpa == 1'b0) && (StepCounter >= vrijeme_punjena) && (StepCounter < (vrijeme_punjena + vrijeme_cekanja)))
begin
    pumpa_switch <= 1'b1; //pumpa ugasena (obrnuta logika)
    dioda_punjene <= 1'b0; //dioda ugasena kada pumpa radi
    StepCounter <= StepCounter + 1'b1;
end
//-----PUNJENJE FLASE-----
if ((ir_pumpa == 1'b0) && (StepCounter < vrijeme_punjena))
begin
    pumpa_switch <= 1'b0; //pumpa upaljena (obrnuta logika)
    dioda_punjene <= 1'b1; //dioda upaljena kada pumpa radi
    StepCounter <= StepCounter + 1'b1;
end
//-----PUMPA UGASENA, CEKANJE DA SE FLASA POMJERI ISpred IR SENZORA-----
else if ((ir_pumpa == 1'b0) && (StepCounter >= vrijeme_punjena) && (StepCounter < (vrijeme_punjena + vrijeme_cekanja)))
begin
    pumpa_switch <= 1'b1; //pumpa ugasena (obrnuta logika)
    dioda_punjene <= 1'b1; //dioda upaljena kada pumpa ne radi
    StepCounter <= StepCounter + 31'b1;
    if (StepCounter == (vrijeme_cekanja + vrijeme_punjena))
        begin
            pumpa_switch <= 1'b1; //pumpa ugasena (obrnuta logika)
            dioda_punjene <= 1'b1; //dioda upaljena kada pumpa ne radi
            StepCounter <= 32'b0;
        end
end
end
endmodule

```

# Grafički prikaz odziva

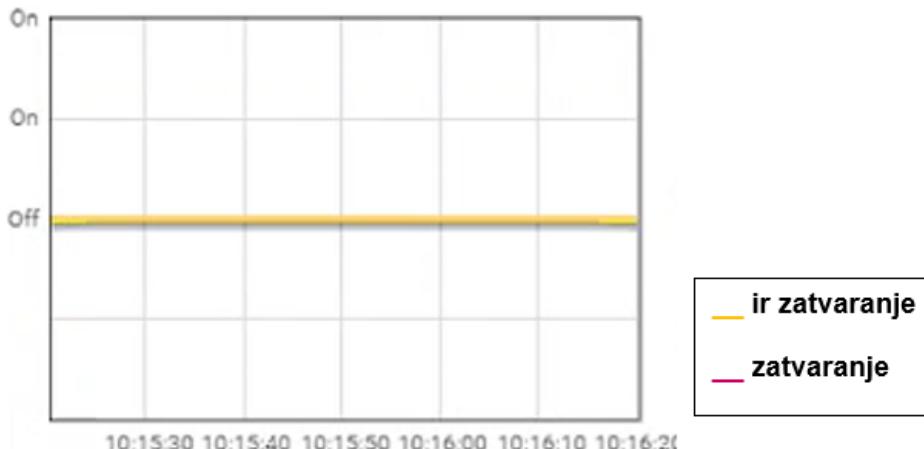
## Scenarij - punjenje flaše



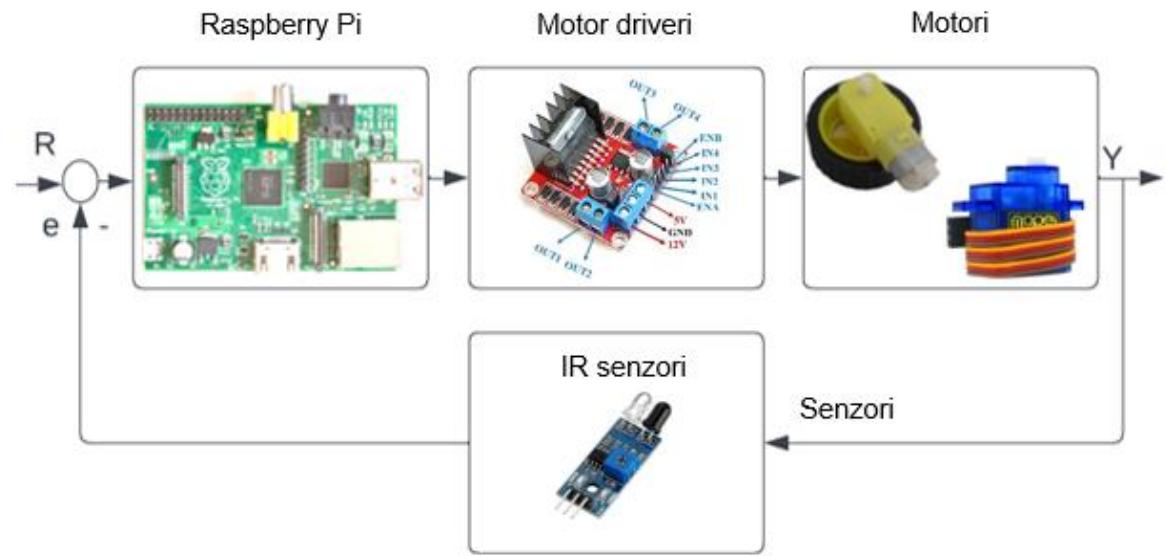
Scenarij - punjenje flaše (traka)



Scenarij - punjenje flaše (pumpa)



Scenarij - punjenje flaše (Mehanizam za zatvaranje)

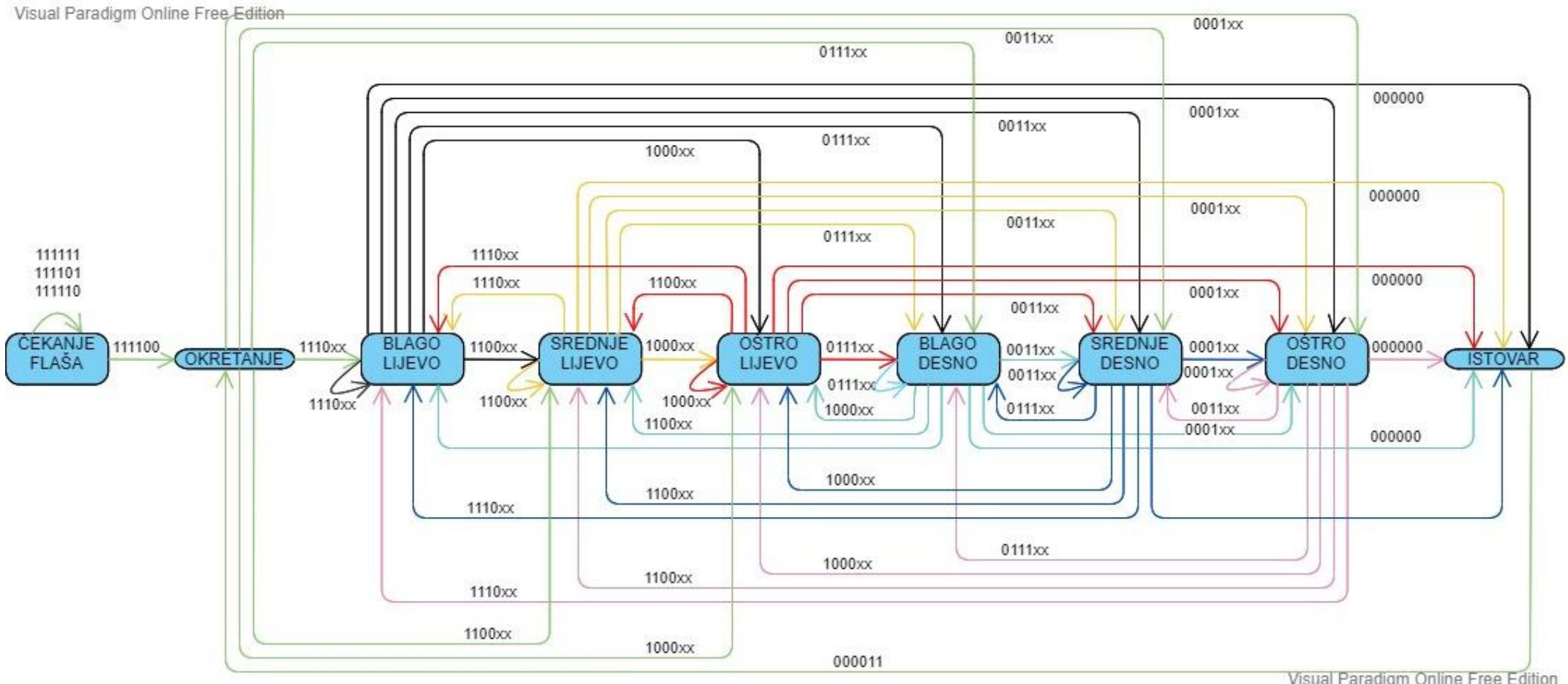


Raspberry Pi	
Ulazi	Izlazi
<b>IR senzori x4</b>	DC motori (kretanje mobilnog robota)
<b>IR senzori x2</b>	Servo motor (Upravljanje nosećom platformom)

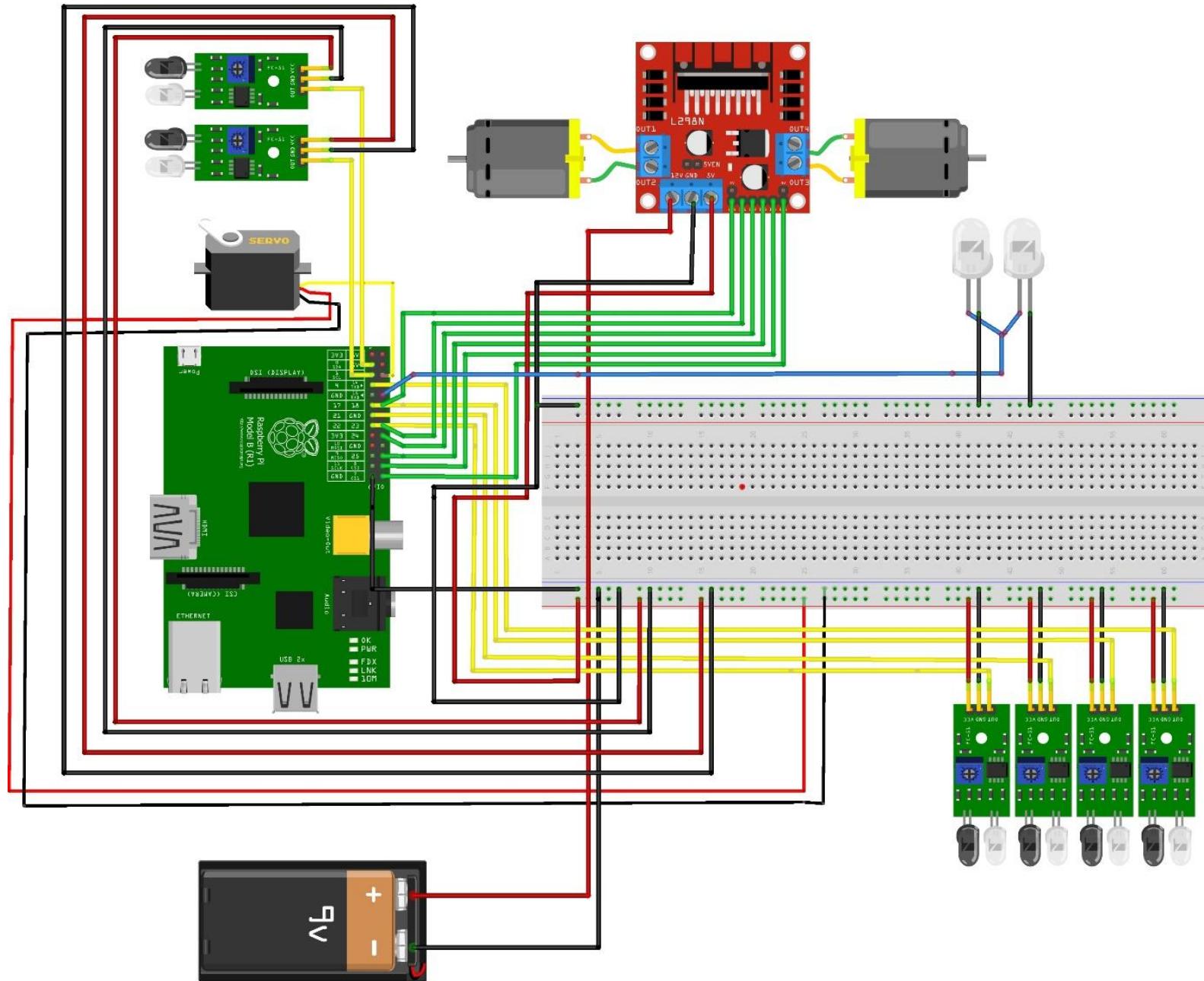
## Algoritam upravljanja

## Upravljanje mobilnim robotom

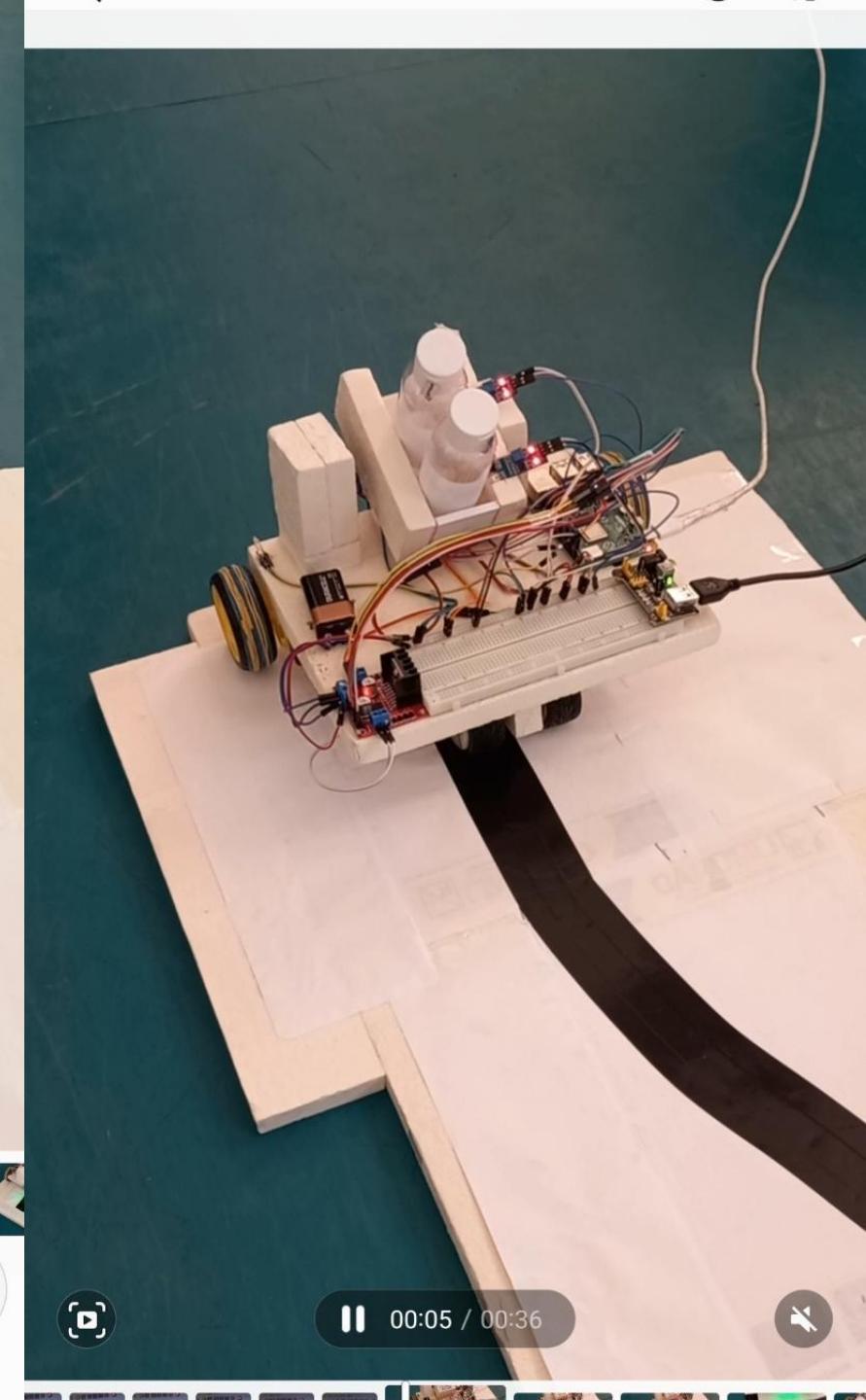
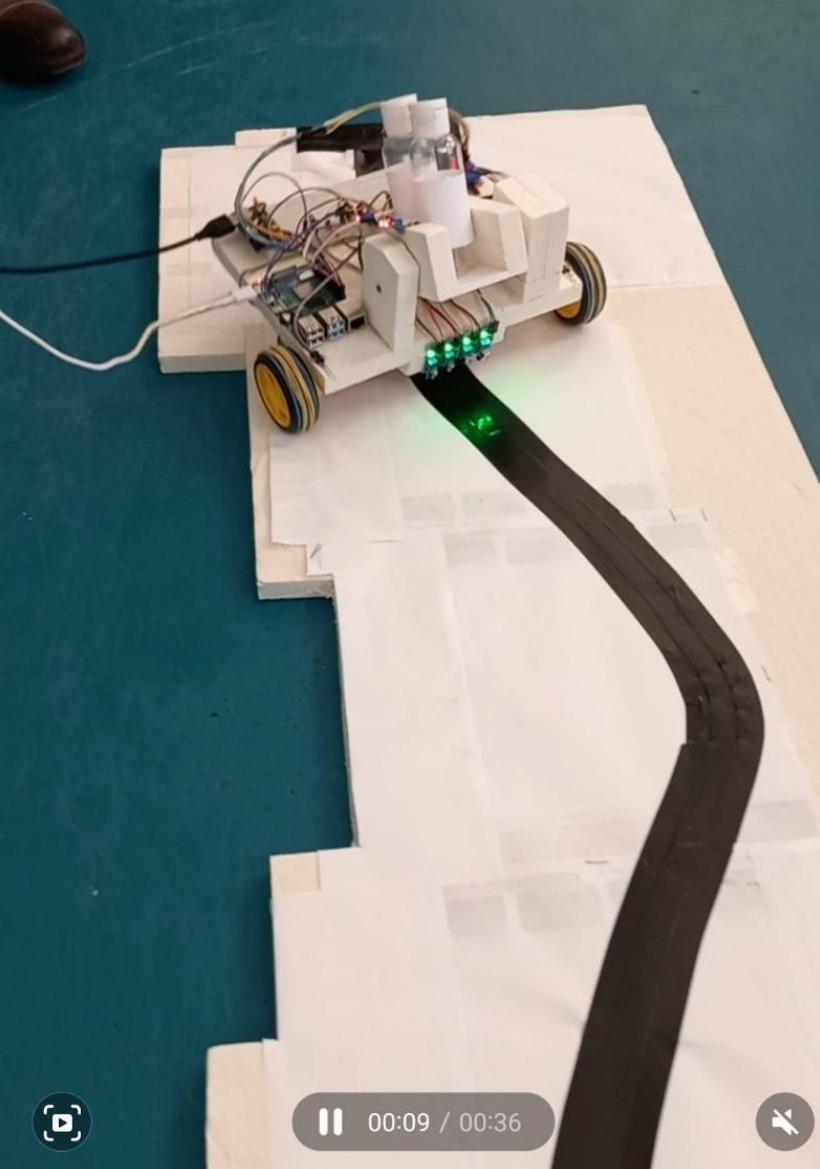
# Mašina konačnog stanja - robot



# Šema povezivanja komponenti na Raspberry Pi



# Prikaz makete roboata



# Kod

```
# funkcija za kretanje naprijed
def move_forward():
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.HIGH)
    GPIO.output(dioda, GPIO.HIGH)

# funkcija za kretanje nazad
def move_backward():
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.HIGH)
    GPIO.output(IN3, GPIO.HIGH)
    GPIO.output(IN4, GPIO.LOW)
    GPIO.output(dioda, GPIO.LOW)

# funkcija za skretanje desno
def turn_right():
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.HIGH)
    GPIO.output(IN4, GPIO.LOW)
    GPIO.output(dioda, GPIO.HIGH)

# funkcija za skretanje lijevo
def turn_left():
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.HIGH)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.HIGH)
    GPIO.output(dioda, GPIO.HIGH)

# Stop
def stop():
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.LOW)
    GPIO.output(dioda, GPIO.LOW)

#funkcija za pomjeranje nazad od trake i okretanje za 180°
def okreni():
    sleep(2)

    #okretanje, prvi dio
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.HIGH)
    GPIO.output(IN4, GPIO.LOW)

    EN1_pwm.ChangeDutyCycle(0)
    EN2_pwm.ChangeDutyCycle(35)
    sleep(1.6)

    #okretanje, drugi dio
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.LOW)

    EN1_pwm.ChangeDutyCycle(35)
    EN2_pwm.ChangeDutyCycle(0)
    sleep(2.3)
    stop()

    #funkcija za pomjeranje nazad nakon istovara i okretanje za 180°
    def okreni2():
        # nazad
        sleep(2)
        move_backward()
        EN1_pwm.ChangeDutyCycle(30)
        EN2_pwm.ChangeDutyCycle(30)
        sleep(0.1)

        #okretanje, prvi dio
        GPIO.output(IN1, GPIO.LOW)
        GPIO.output(IN2, GPIO.HIGH)
        GPIO.output(IN3, GPIO.LOW)
        GPIO.output(IN4, GPIO.LOW)

        EN1_pwm.ChangeDutyCycle(35)
        EN2_pwm.ChangeDutyCycle(0)
        sleep(2)

        #okretanje, drugi dio
        GPIO.output(IN1, GPIO.LOW)
        GPIO.output(IN2, GPIO.LOW)
        GPIO.output(IN3, GPIO.LOW)
        GPIO.output(IN4, GPIO.HIGH)

        EN1_pwm.ChangeDutyCycle(0)
        EN2_pwm.ChangeDutyCycle(35)
        sleep(2.1)
        stop()

        #istovar flasa, pomjeranje nazad i okretanje za 180°
        def istovar():
            stop()
            EN1_pwm.ChangeDutyCycle(0)
            EN2_pwm.ChangeDutyCycle(0)
            sleep(1)
            SERVO_pwm.ChangeDutyCycle(5)
            sleep(2) # dok servo spusta platformu
            SERVO_pwm.ChangeDutyCycle(7)
            sleep(1) # dok servo podize platformu
            sleep(1) # dok servo podize platformu
            okreni2()

    # srednje desno skretanje
    elif val_ir_1 == 0 and val_ir_2 == 0 and val_ir_3 == 1 and val_ir_4 == 1:
        turn_left()
        EN1_pwm.ChangeDutyCycle(35)
        EN2_pwm.ChangeDutyCycle(30)

    # jako desno skretanje
    elif val_ir_1 == 0 and val_ir_2 == 0 and val_ir_3 == 0 and val_ir_4 == 1:
        turn_left()
        EN1_pwm.ChangeDutyCycle(40)
        EN2_pwm.ChangeDutyCycle(30)

    # čekanje flaša
    elif (val_ir_1 == 1 and val_ir_2 == 1 and val_ir_3 == 1 and val_ir_4 == 1 and
          val_ir_flasa1 == 1 and val_ir_flasa2 == 1) \
            or (val_ir_1 == 1 and val_ir_2 == 1 and val_ir_3 == 1 and val_ir_4 == 1 and
                val_ir_flasa1 == 0 and val_ir_flasa2 == 1) \
            or (val_ir_1 == 1 and val_ir_2 == 1 and val_ir_3 == 1 and val_ir_4 == 1 and
                val_ir_flasa1 == 1 and val_ir_flasa2 == 0):
        stop()

    # okretanje robota
    elif val_ir_1 == 1 and val_ir_2 == 1 and val_ir_3 == 1 and val_ir_4 == 1 and val_ir_flasa1 == 0 and val_ir_flasa2 == 0:
        okreni()

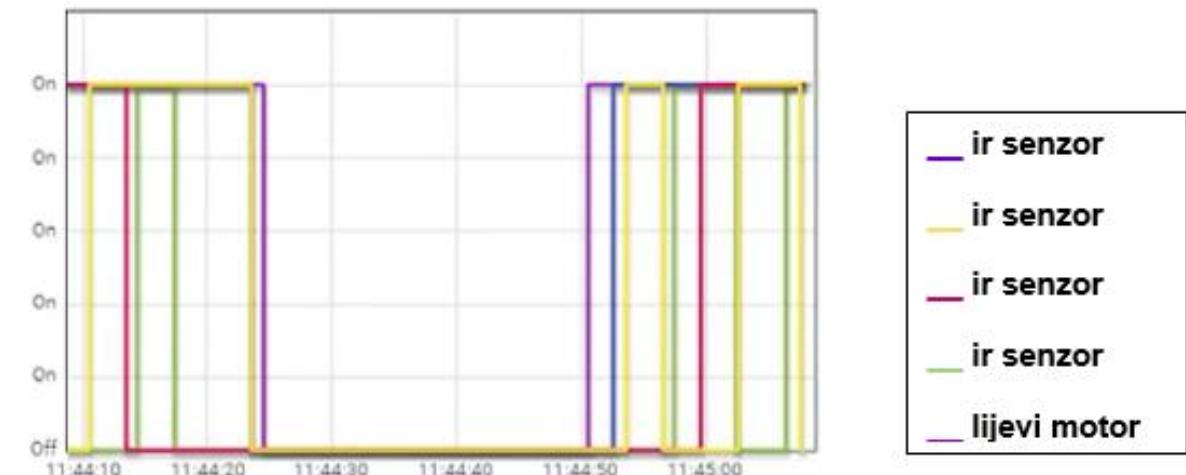
    # istovar flaša
    elif val_ir_1 == 0 and val_ir_2 == 0 and val_ir_3 == 0 and val_ir_4 == 0 and val_ir_flasa1 == 0 and val_ir_flasa2 == 0:
        istovar()

    else:
        stop()
```

# Grafički prikaz odziva



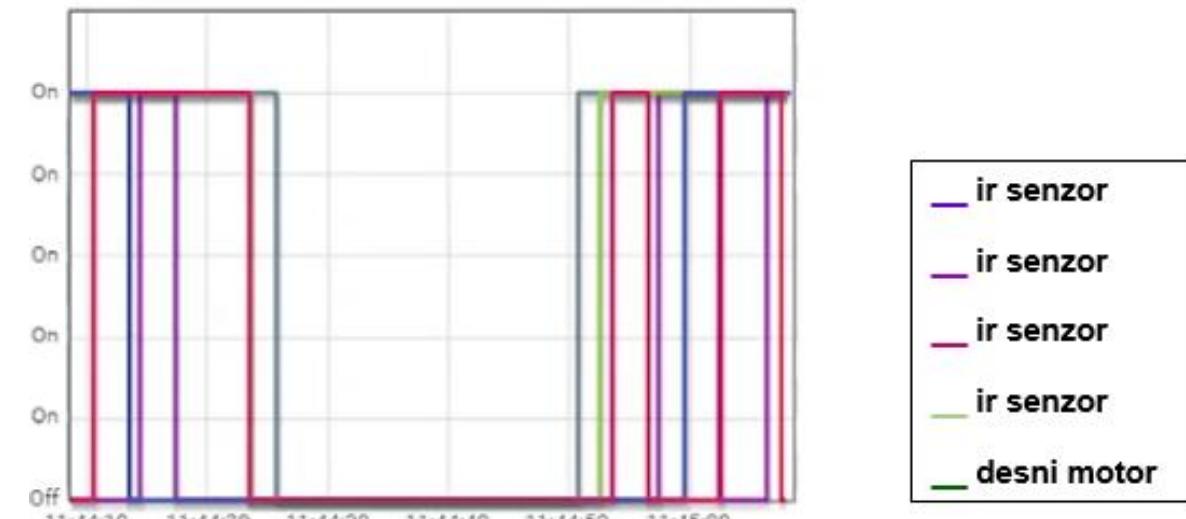
Desni motor: čekanje flaša – praćenje linije



Desni motor: praćenje linije - istovar flaša – praćenje linije

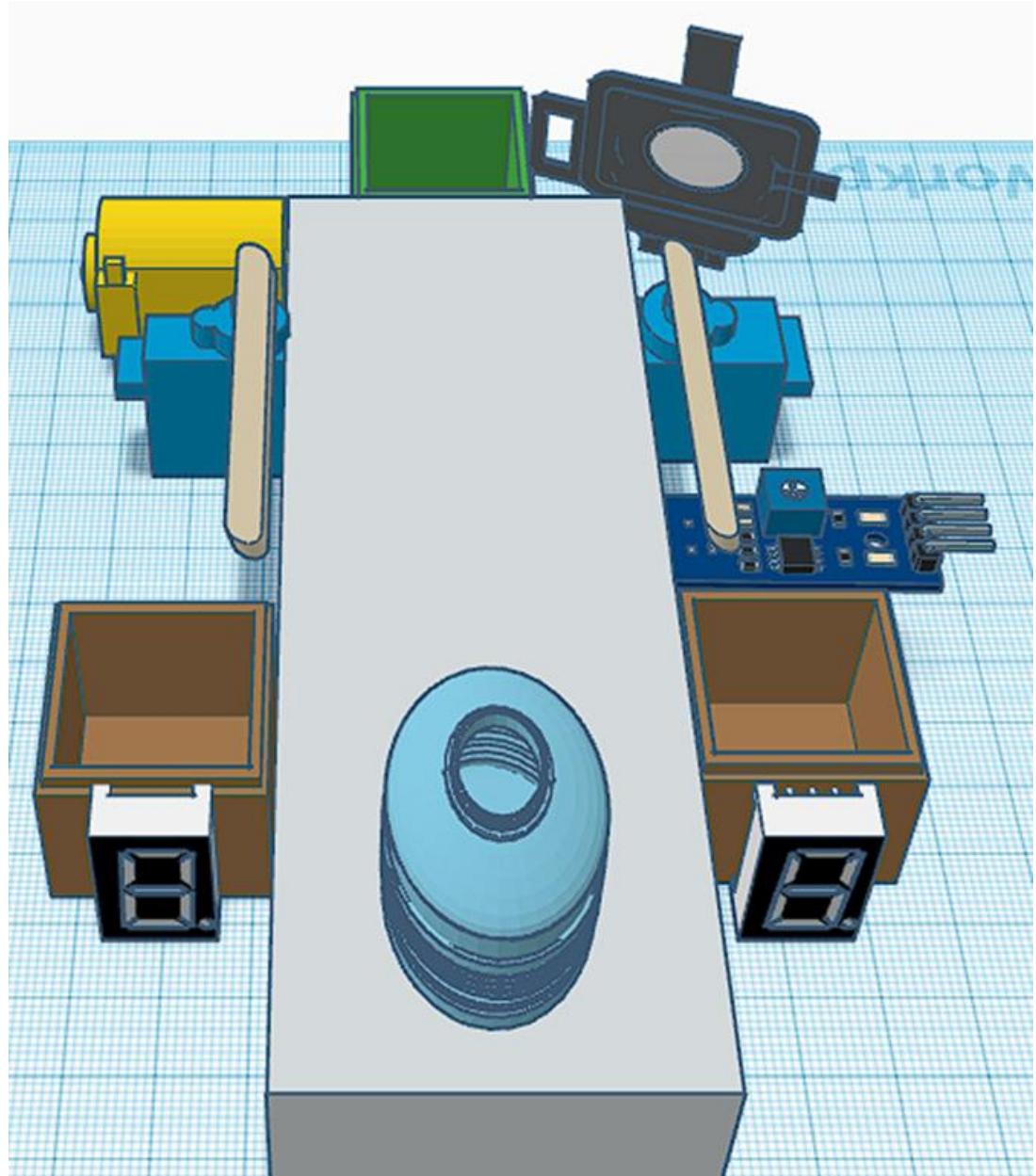


Desni motor: čekanje flaša – praćenje linije



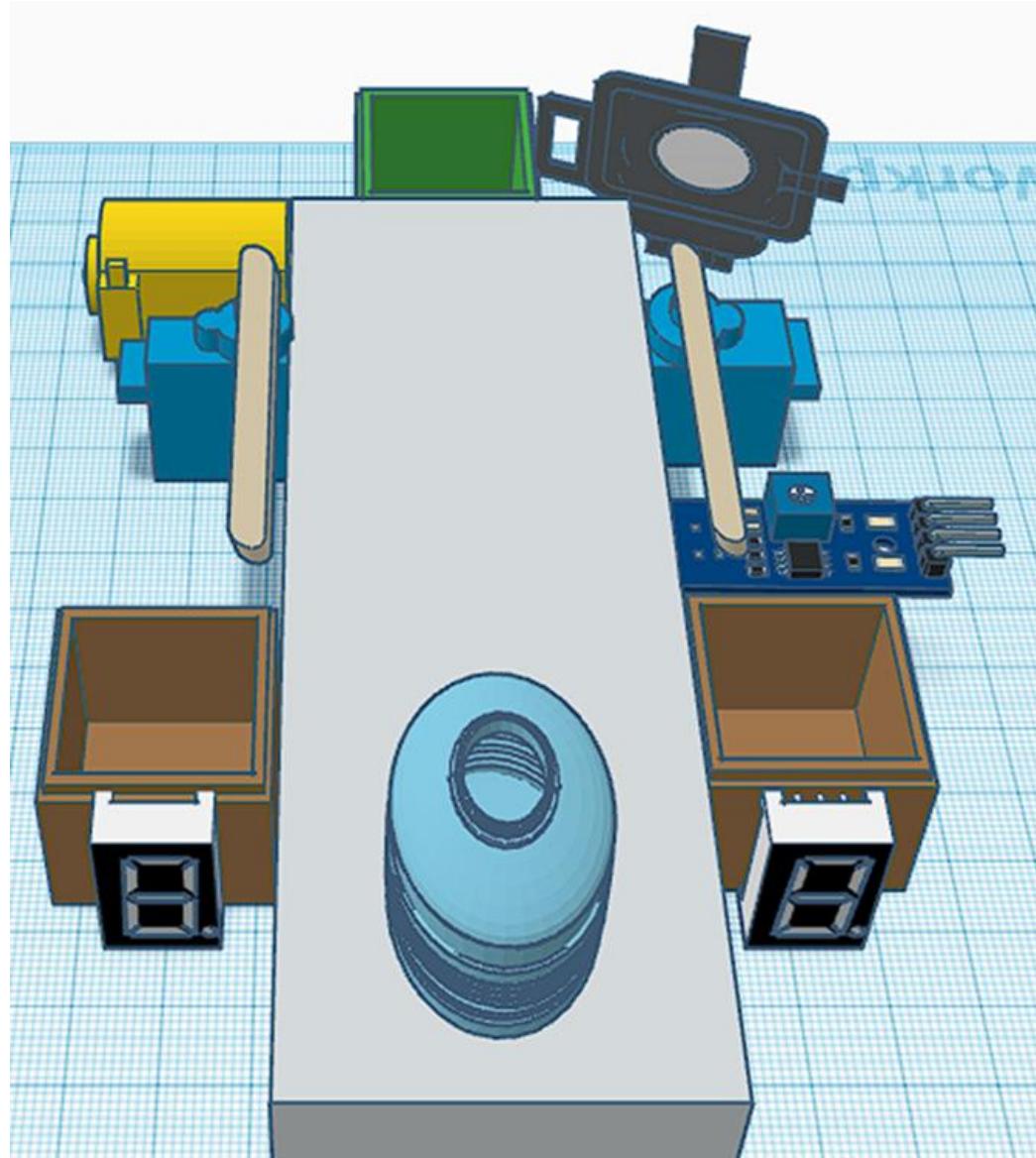
Lijevi motor: praćenje linije - istovar flaša – praćenje linije

## Primjer 2: Sortiranje predmeta na pokretnoj traci na osnovu oblika



# Opis projekta

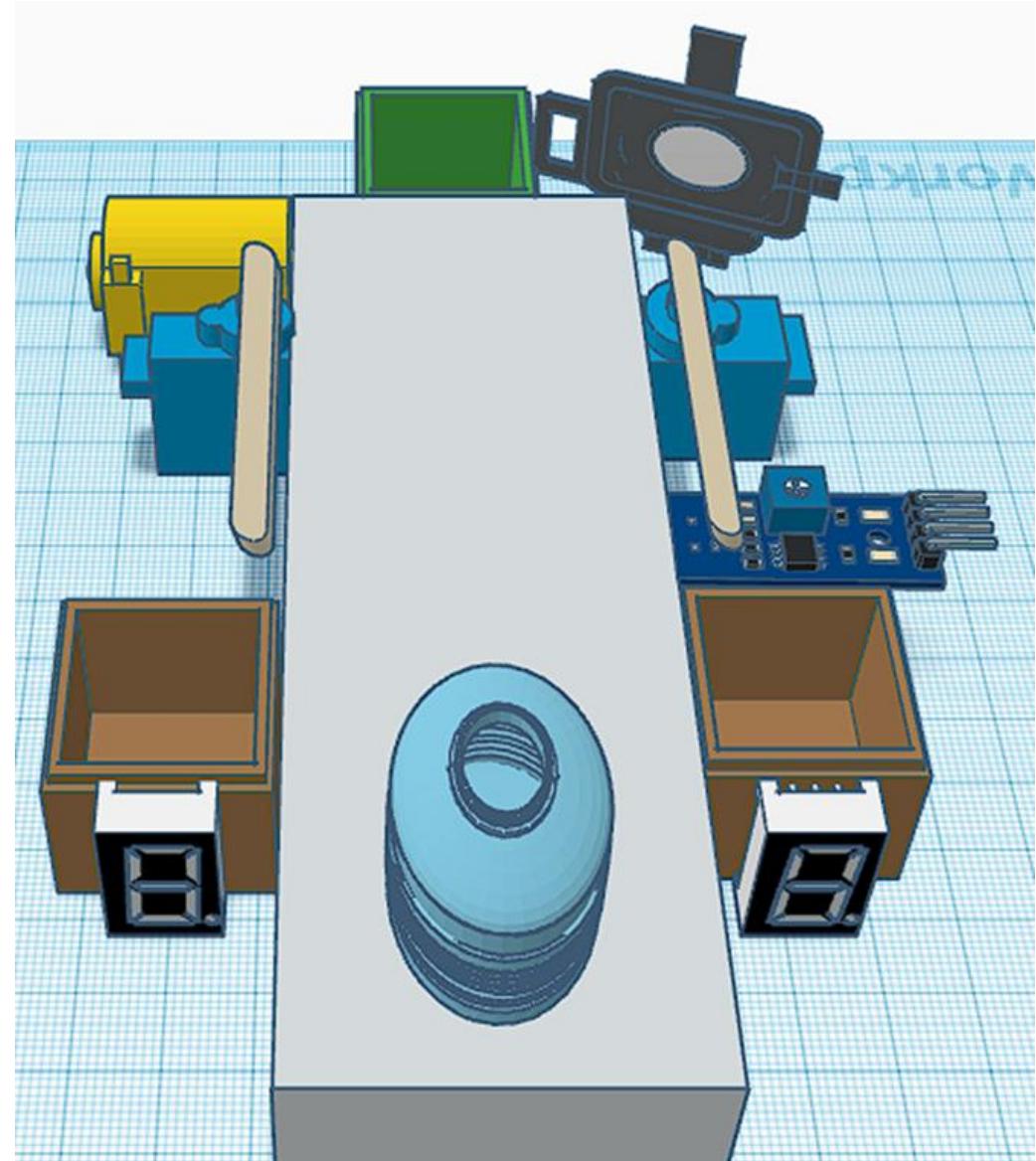
- Na pokretnoj traci, koju pokreće akumulator od 12 V, nalaze se predmeti različitih oblika (bottle, cup, can) i kreću se po istoj.
- Raspberry Pi kamera Module 2 implementirana je na samom početku pokretne trake, kako bismo odmah detektovali oblik predmeta.
- Svaki servo motor se pokreće u odnosu na to koji oblik predmeta na traci imamo.
- Pored trake se nalaze koševi,takođe implementirani su 7 segment-display koji broji koliko predmeta se nalazi u košu.



# Komponente

---

- Za realizaciju projekta, koristene su sljedeće komponente
  - Cyclone IV EP4CE6
  - Raspberry Pi
  - Rasberry Pi Kamera
  - Servo motori
  - Ir Senzor
  - Napajanje 12V
  - USB blaster
  - 7 segmenti display i
  - DC motor

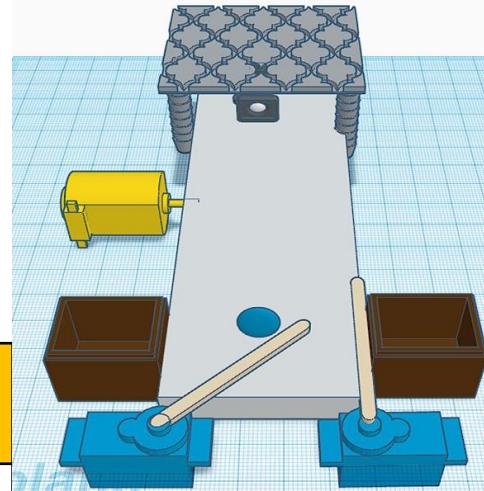


# Ulazi sistema

Ulazi :	Pločica koja ga upravlja	Namjena	Slika
Kamera	Raspberry Pi	Raspoznavanje oblika	
Opto senzor	FPGA	Brojanje predmeta	
IR senzor	FPGA	Detekciju predmeta	

# Izlazi sistema

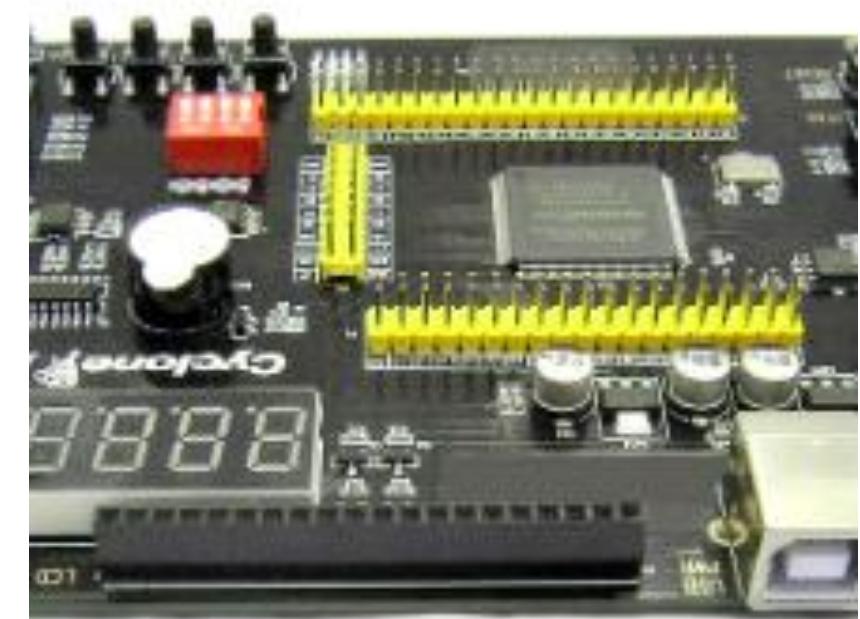
Izlazi :	Pločica koja ga upravlja :	Namjena :	Slika :
Stepper motor 24 V	Raspberry Pi	Pokretanje pokretne trake	
Led Display	Raspberry Pi	Broj predmeta u undefined kutiji	
Servo motor	FPGA	Pokretanje desnog upravljača	
Servo motor	FPGA	Pokretanje lijevog upravljača	
Led Display	FPGA	Broj predmeta u desnoj kutiji	
Led Display	FPGA	Broj predmeta u lijevoj kutiji	



# Cyclone IV

---

- Optimizirana za najmanju potrošnju energije
- Sadrži 6000 logičkih elemenata
- Sadrži 18x18 mnozaca
- 270KB embedded memorije



# Raspberry Pi

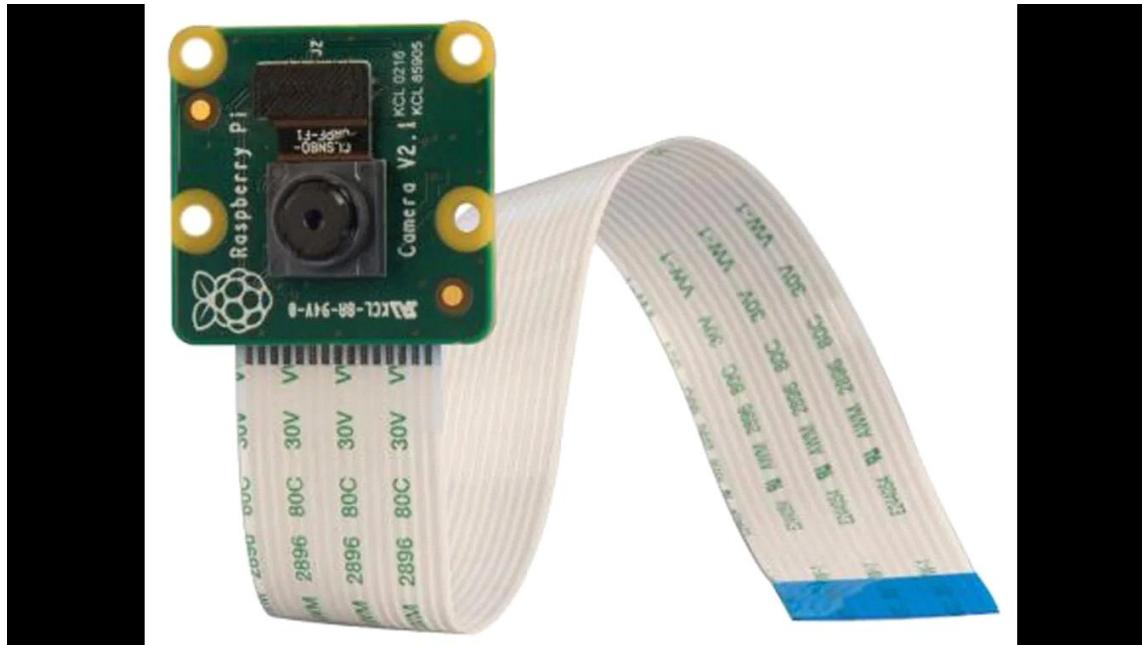
---

- 4 GB brze memorije,
- podršku 4K zaslona,
- uključivanje USB 3.0.
- Raspberry Pi 4 koristi Broadcom BCM2711 SoC s 1,5 GHz (kasniji modeli: 1,8 GHz) 64-bitnim četverozezgrenim ARM Cortex-A72 procesorom, s 1 MB dijeljene L2 predmemorije. Dostupan je od 1, 2, 4 ili 8 GB RAM memorije



# Raspberry Pi Kamera

---



- Kamera Module ima Sony IMX219 senzor od 8 megapiksela sa lećom za fokusiranje, te se može koristiti za snimanje videa i fotografija.
- Jednostavna je za korištenje, ali bez obzira na to može puno da ponudi.
- Podržava 1080p30, 720p30 i VGA90 video načine.
- Što se tiče slika, kamera može snimati fotografije do 3280 x 2464 piksela.
- Priklučuje se trakastim kablom od 15 cm na CSI priključak na Raspberry Pi.

# Servo motor

---



- Servomotor je elektromotor koji prema primljenom upravljačkom signalu zauzima određeni zakretni položaj, ili mjesto na nekoj putanji, odnosno razvija odgovarajući zakretni momenat ili silu.
- Pokretanjem i zaustavljanjem sevomotora upravlja se regulacijskim krugom, koji primljeni upravljački signal manje snage pretvara u pomake, najčešće s većom izvršnom mehaničkom snagom.
- Između regulacijskog djela i servomotora obično se ugrađuje mjerni instrument, naprimjer davač položaja ili enkoder, koji utvrđuje tačan položaj izvršnog dijela servomotora, te izmjereni položaj po potrebi popravlja.

# DC motor

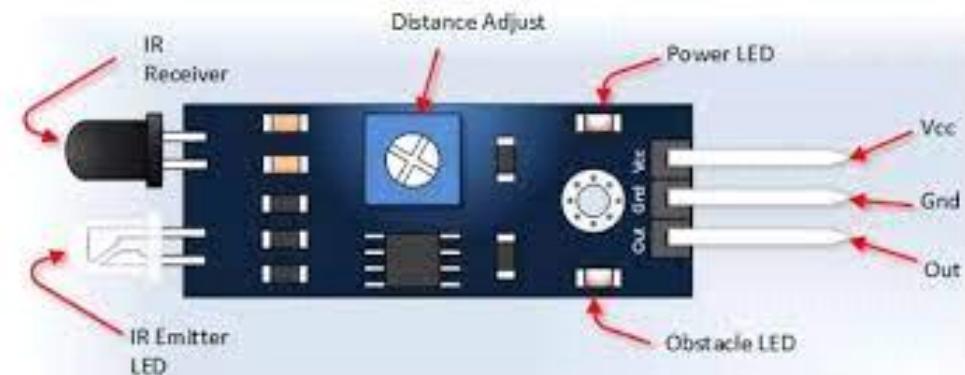
---



- Istosmjerni motor je elektromehanički uređaj koji električnu energiju pretvara u mehaničku odnosno pretvara istosmjernu struju u kružno kretanje, ali mogu raditi i u suprotnom smjeru i pretvarati kružno kretanje u električnu energiju.
- Istosmjerni motori nalaze primjenu u velikom broju uređaja.
- Glavni dijelovi motora su rotor i stator kao i kod svakog drugog motora, ali ova vrsta motora ima još i komutator s grafitnim četkicama.
- Kod istosmjernog motora magneti na statoru nam stvaraju magnetsko polje, a kroz namotaje rotora propuštamo struju i na namotaje djeluje sila koja zakreće rotor.

# IR senzor

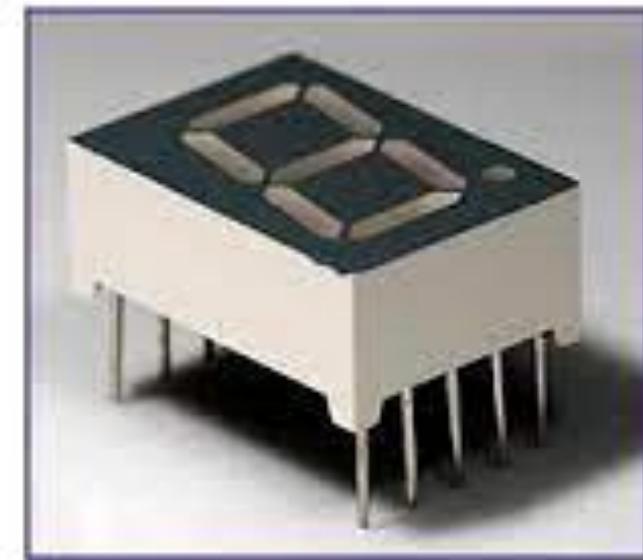
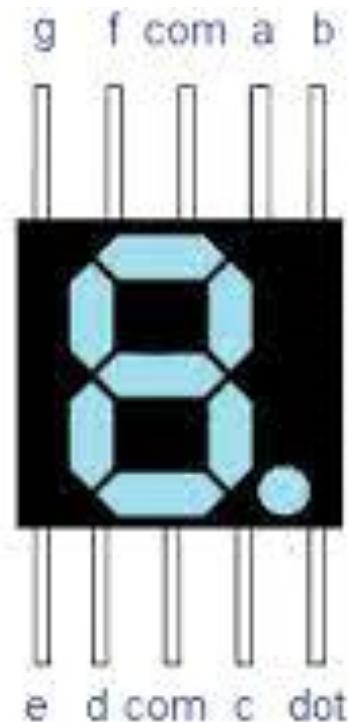
- Glavne prednosti IR senzora su mala potrošnja energije i njihov jednostavan dizajn.
- Obično se valne duljine tih valova kreću od  $0,7 \mu\text{m}$  do  $1000 \mu\text{m}$ .
- IR spektar možemo podijeliti u tri regije poput bliske infracrvene, srednje i daleke infracrvene.
- Ovaj senzor uključuje IR LED i IR fotodiodu, tako da se kombiniranjem ove dvije slike mogu oblikovati kao foto-spojnica, inače optički sprežnik.



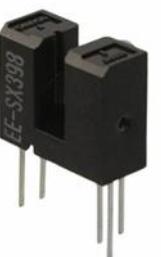
# 7 segmenti display

---

- Display ima 10 nožica, gdje se srednje nožice sa gornje i donje strane odnose na zajedničku katodu, odnosno anodu, zavisno o modelu.
- Ostalih 8 nozica preostaje za prikaz broja plus takčku koja se nalazi u desnom donjem uglu.
- Prilikom spajanja, samo trebamo obratiti pažnju da li imamo display sa zajedničkom katodom ili anodom.

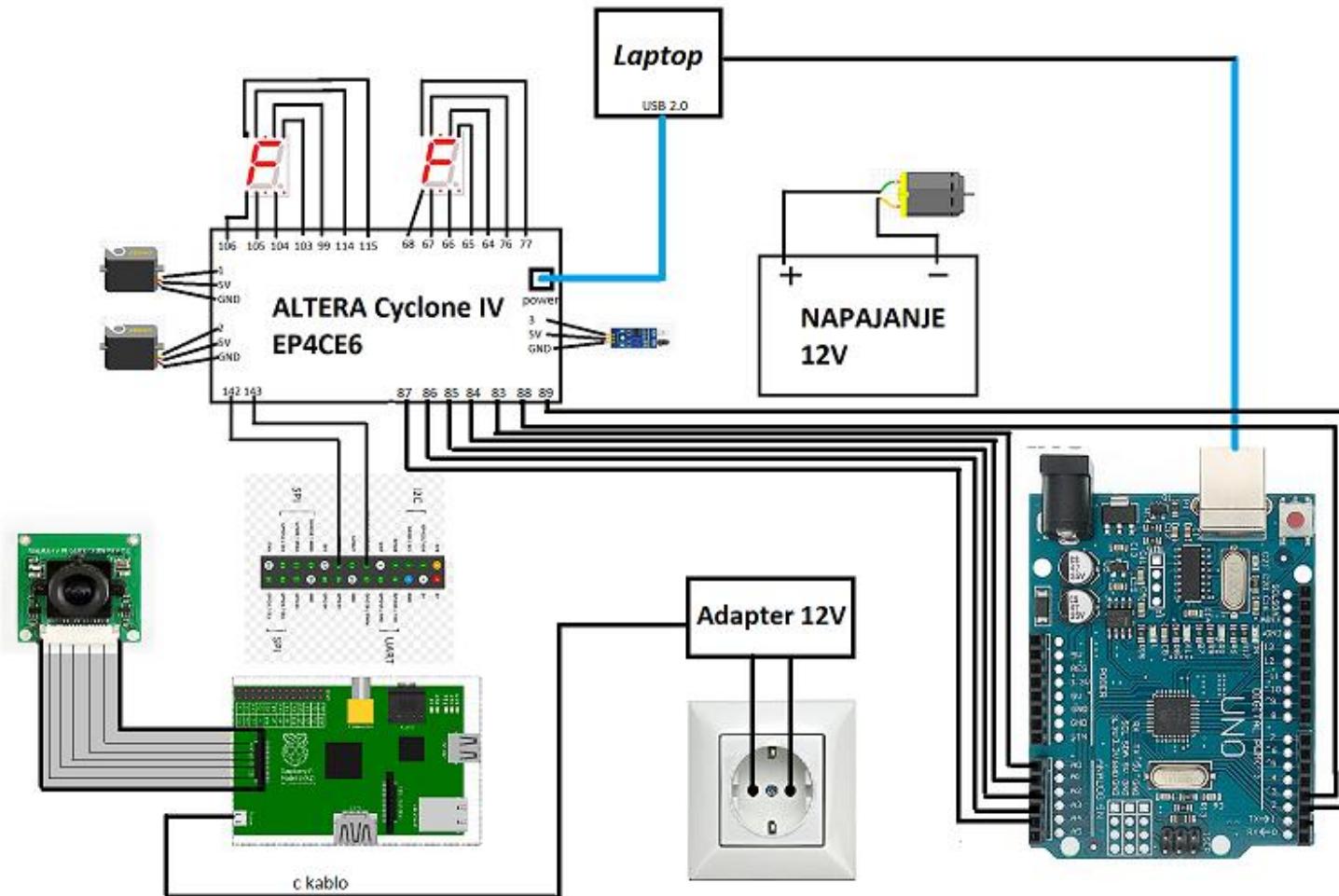


# Logika procesa

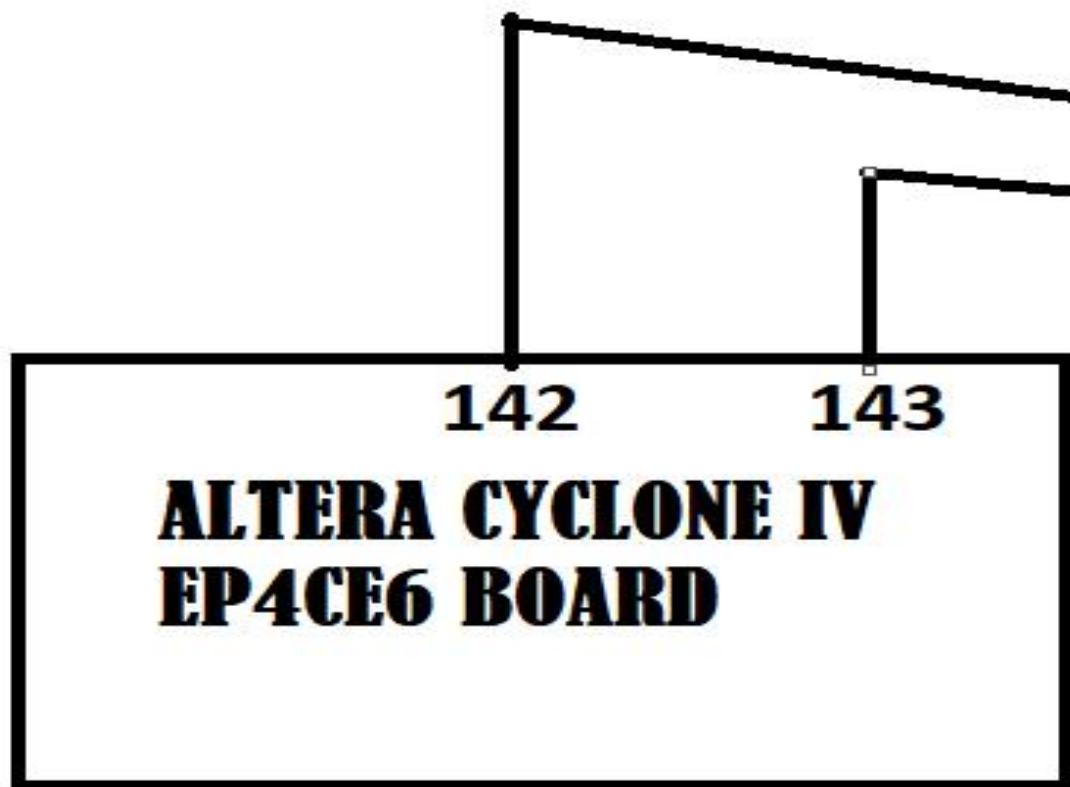


1. Kamera – detektuje o kojem se obliku radi.
2. Ako je kamera detektovala oblik “cup”, šalje signal Raspberry-u koji prenosi signal na FPGA da je na traci časa, koja mora biti odložena u lijevi koš (aktivira se motor1 – okreće se za 90 stepeni i pomjera se branik na traci).
3. Traka se pokreće pomoću DC motora koji je spojen na akumulator. Čaša kretanjem nailazi na IR senzor.
4. Kada je čaša došla do IR senzora, senzor šalje signal motoru da je detektovao predmet (čašu), te da se motor treba vratiti u početni položaj, odnosno –90 stepeni. Vraćanjem u nulto stanje, branik na motoru udara u čašu i ona ulazi u koš sa lijeve strane, namjenjen za predmete u obliku čaše. Na 7 segmentnom display-u povećava se broj za jedan (+1), gdje u svakom trenutku mozemo ocitati koliko predmeta se nalazi u košu.
5. Ukoliko kamera detektuje oblik “bottle”, šalje signal Raspberry-u, koji prenosi signal na FPGA. Pokreće se motor2 (desni motor), flaša se kreće, nailazi na IR senzor koji šalje signal motoru2 da se treba vratiti u početni položaj. Tim pokretanjem, improvizovana ruka na motoru udara flašu i ona pada u kutiju sa desne strane trake u kojoj je predviđeno da se odlažu flaše. Na display-u imamo uvećan broj za 1 (u odnosu na prethodno stanje u kutiji).

# Šema spajanja

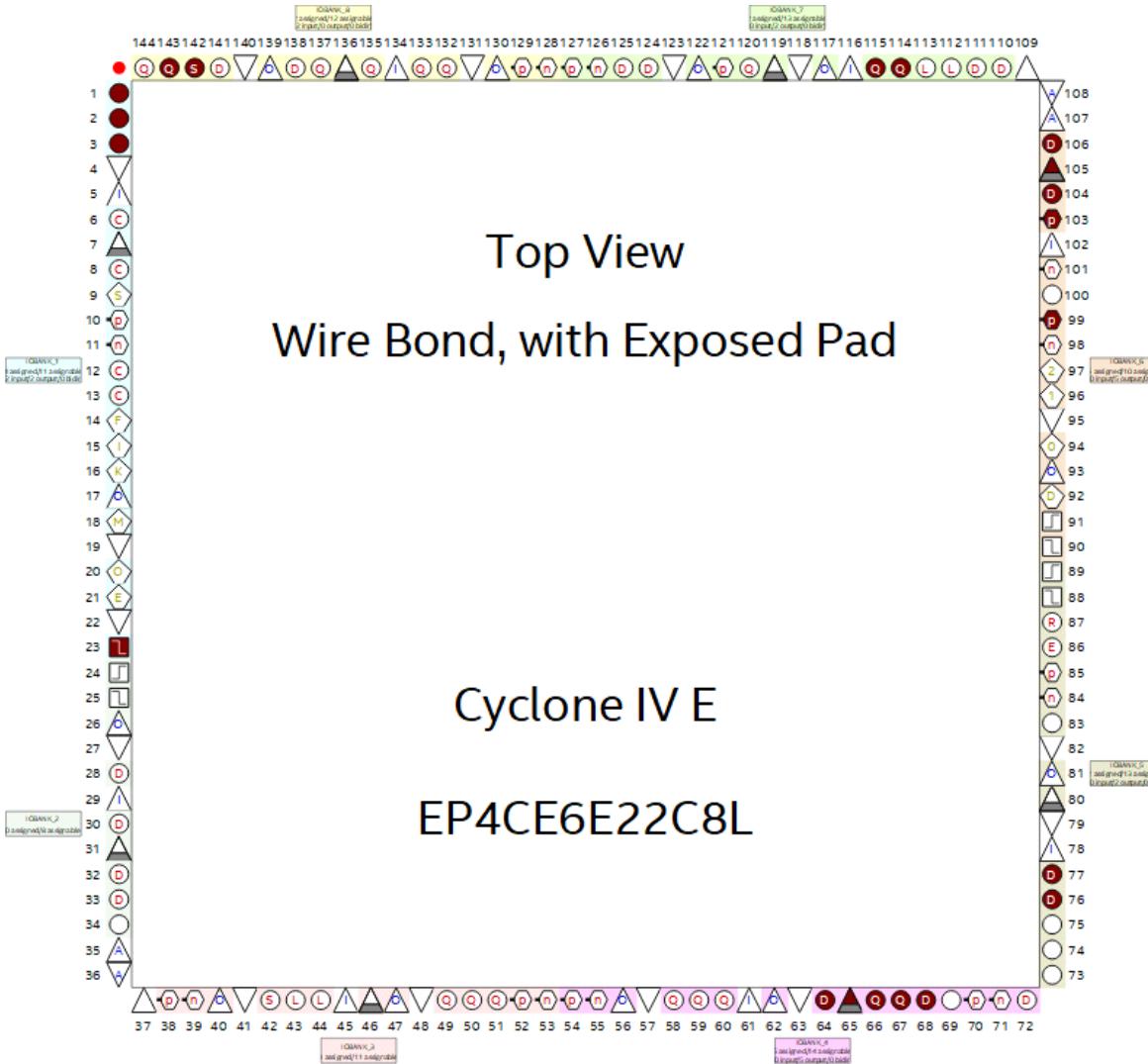


# RPi i FPGA komunikacija

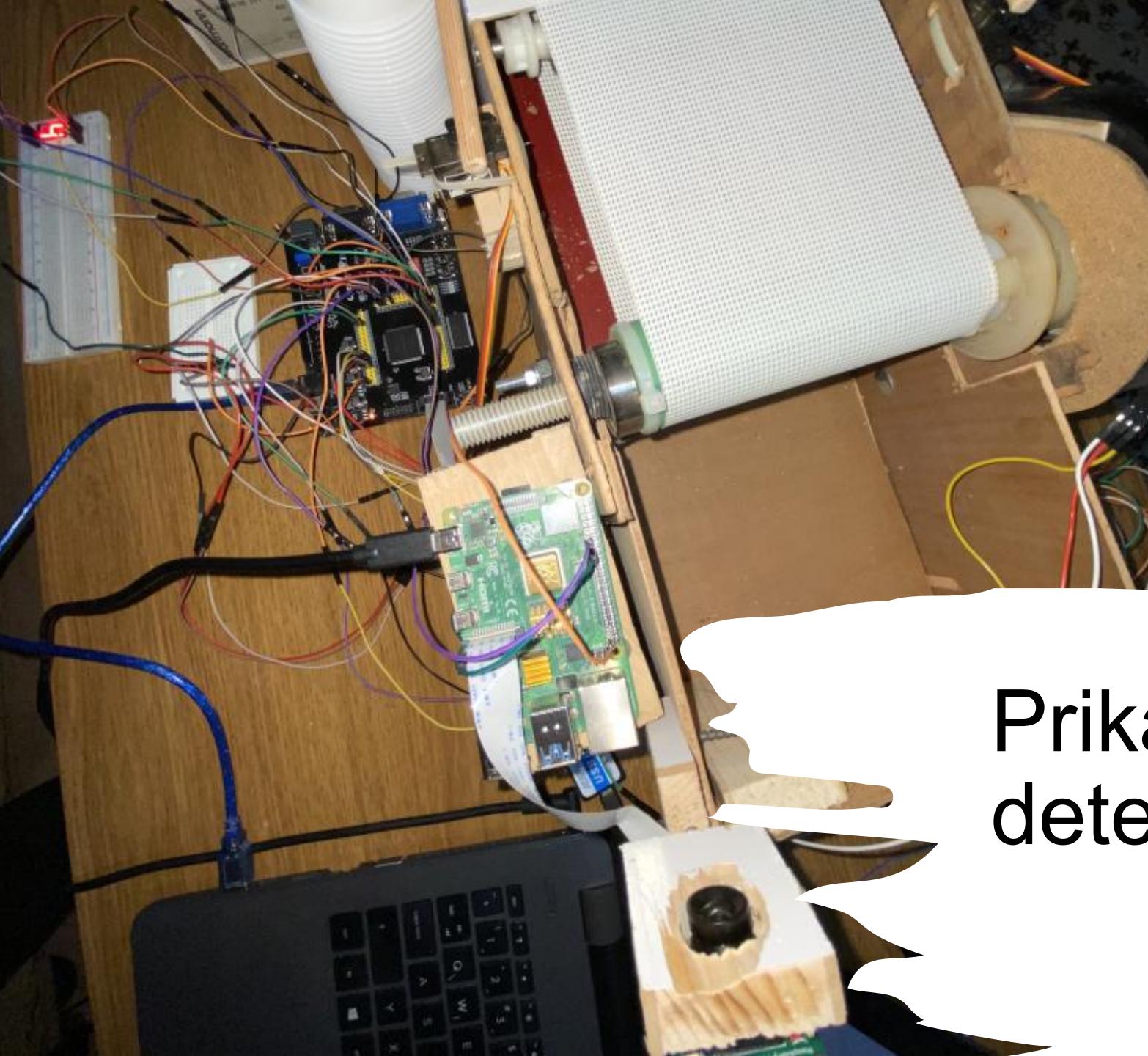


Raspberry Pi Pinout	
3v3 Power	1 2
BCM 2 (SDA)	3 4
BCM 3 (SCL)	5 6
BCM 4 (GPCLK0)	7 8
Ground	9 10
BCM 17	11 12
BCM 18	13 14
BCM 19	15 16
BCM 20	17 18
3v3 Power	19 20
BCM 10 (MOSI)	21 22
BCM 9 (MISO)	23 24
BCM 11 (SCL0)	25 26
Ground	27 28
BCM 0 (ID_SD)	29 30
BCM 5	31 32
BCM 6	33 34
BCM 13 (PWM1)	35 36
BCM 19 (MISO)	37 38
BCM 26	39 40
Ground	

# Pin planner



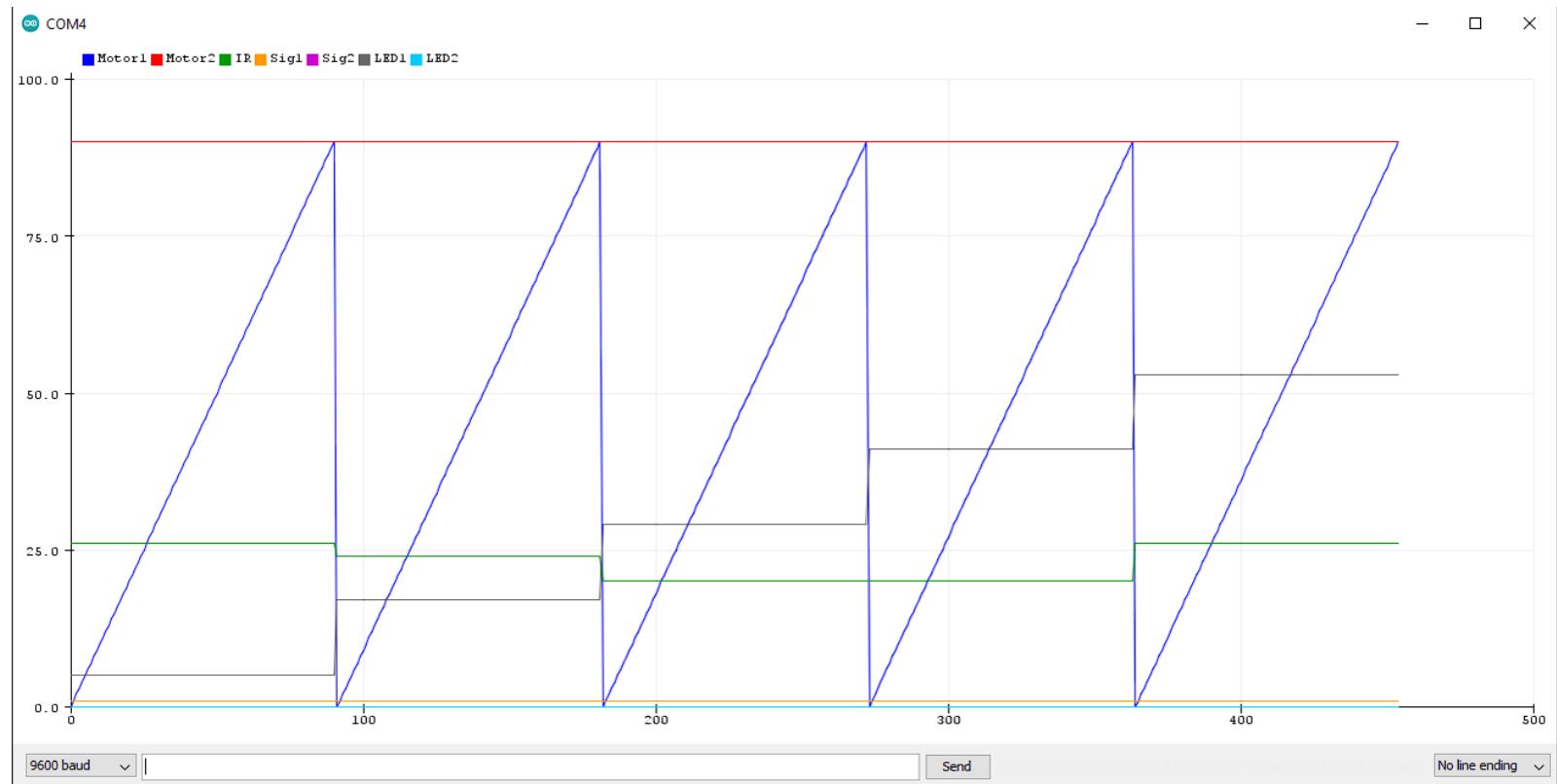
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Electrical Preset Value
in IrPin	Input	PIN_3	1	B1_N0	PIN_3	2.5 V		8mA (default)			
out LED_out1[0]	Output	PIN_64	4	B4_N0	PIN_64	2.5 V		8mA (default)	2 (default)		
out LED_out1[1]	Output	PIN_65	4	B4_N0	PIN_65	2.5 V		8mA (default)	2 (default)		
out LED_out1[2]	Output	PIN_66	4	B4_N0	PIN_66	2.5 V		8mA (default)	2 (default)		
out LED_out1[3]	Output	PIN_67	4	B4_N0	PIN_67	2.5 V		8mA (default)	2 (default)		
out LED_out1[4]	Output	PIN_68	4	B4_N0	PIN_68	2.5 V		8mA (default)	2 (default)		
out LED_out1[5]	Output	PIN_76	5	B5_N0	PIN_76	2.5 V		8mA (default)	2 (default)		
out LED_out1[6]	Output	PIN_77	5	B5_N0	PIN_77	2.5 V		8mA (default)	2 (default)		
out LED_out2[0]	Output	PIN_99	6	B6_N0	PIN_99	2.5 V		8mA (default)	2 (default)		
out LED_out2[1]	Output	PIN_103	6	B6_N0	PIN_103	2.5 V		8mA (default)	2 (default)		
out LED_out2[2]	Output	PIN_104	6	B6_N0	PIN_104	2.5 V		8mA (default)	2 (default)		
out LED_out2[3]	Output	PIN_105	6	B6_N0	PIN_105	2.5 V		8mA (default)	2 (default)		
out LED_out2[4]	Output	PIN_106	6	B6_N0	PIN_106	2.5 V		8mA (default)	2 (default)		
out LED_out2[5]	Output	PIN_114	7	B7_N0	PIN_114	2.5 V		8mA (default)	2 (default)		
out LED_out2[6]	Output	PIN_115	7	B7_N0	PIN_115	2.5 V		8mA (default)	2 (default)		
in clk	Input	PIN_23	1	B1_N0	PIN_23	2.5 V		8mA (default)			
out motor1	Output	PIN_1	1	B1_N0	PIN_1	2.5 V		8mA (default)	2 (default)		
out motor2	Output	PIN_2	1	B1_N0	PIN_2	2.5 V		8mA (default)	2 (default)		
in sig1	Input	PIN_142	8	B8_N0	PIN_142	2.5 V		8mA (default)			
in sig2	Input	PIN_143	8	B8_N0	PIN_143	2.5 V		8mA (default)			
<<new node>>											



Prikaz makete i  
detekcija objekta

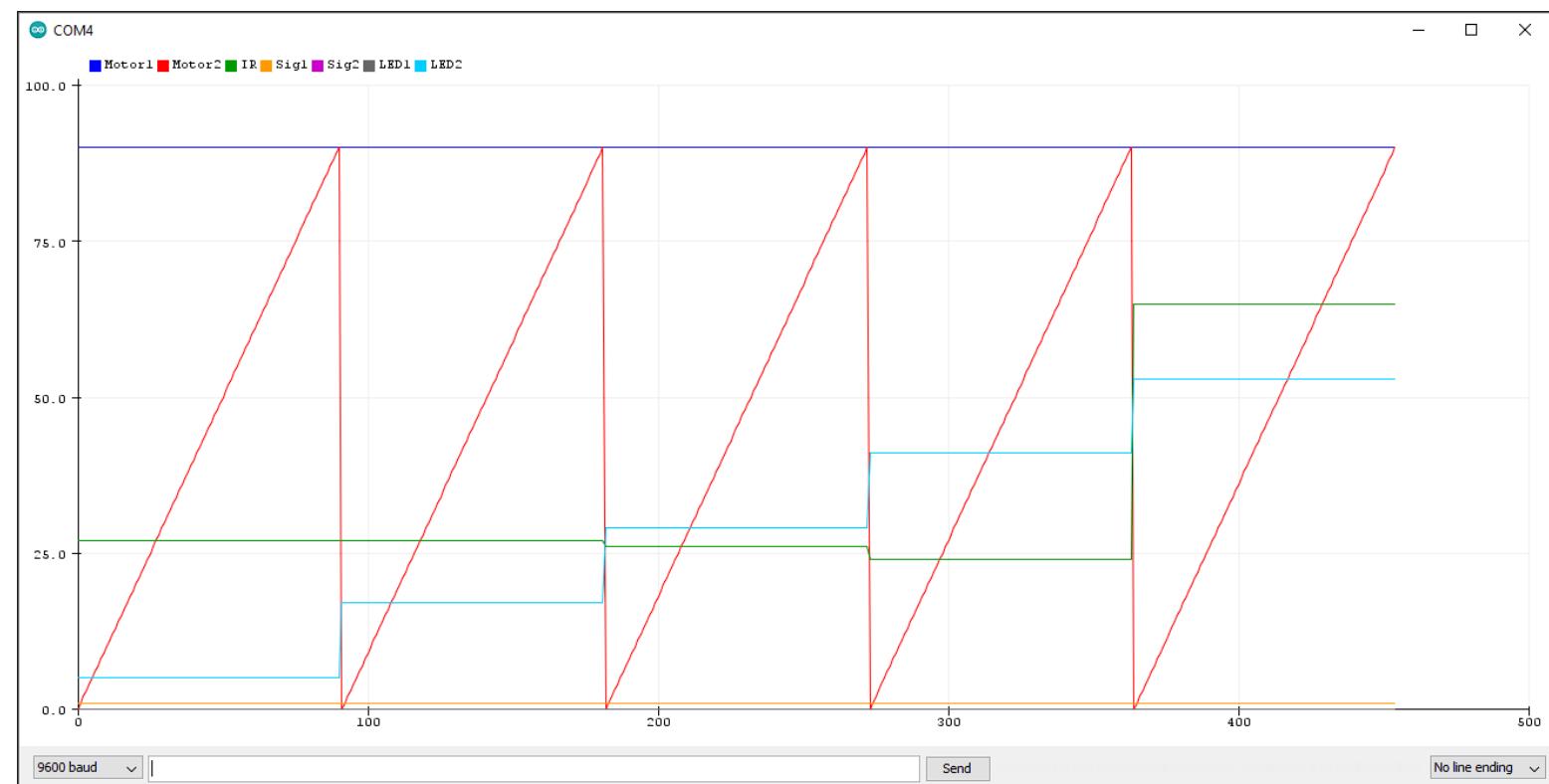
# Rezultati eksperimenta

- Slučaj 1. RPi je nakon detekcije koju je izvršila RPi kamera poslao signale  $\text{sig1}=1$ ,  $\text{sig2}=0$ . Na svaku IR detekciju motor1 mjenja stanje, 7seg display1 mjenja stanje, motor2 miruje.

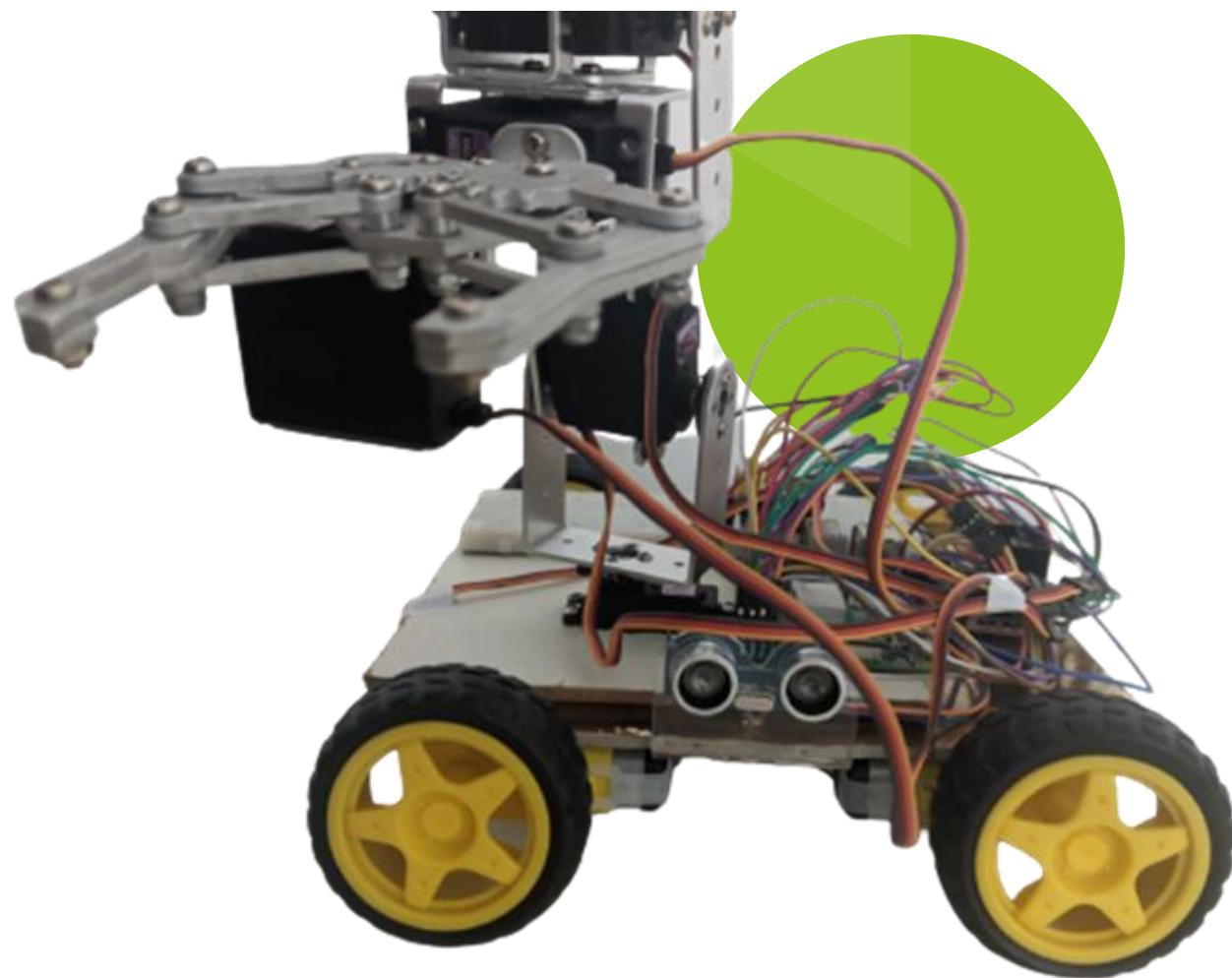


# Rezultati eksperimenta

- Slučaj 2. RPi je nakon detekcije koju je izvršila RPi kamera poslao signale  $\text{sig1}=0$ ,  $\text{sig2}=1$ . Na svaku IR detekciju motor2 mjenja stanje, 7seg display2 mjenja stanje, motor1 miruje.



# Primjer 3: Garbage Collector



# Cilj projekta

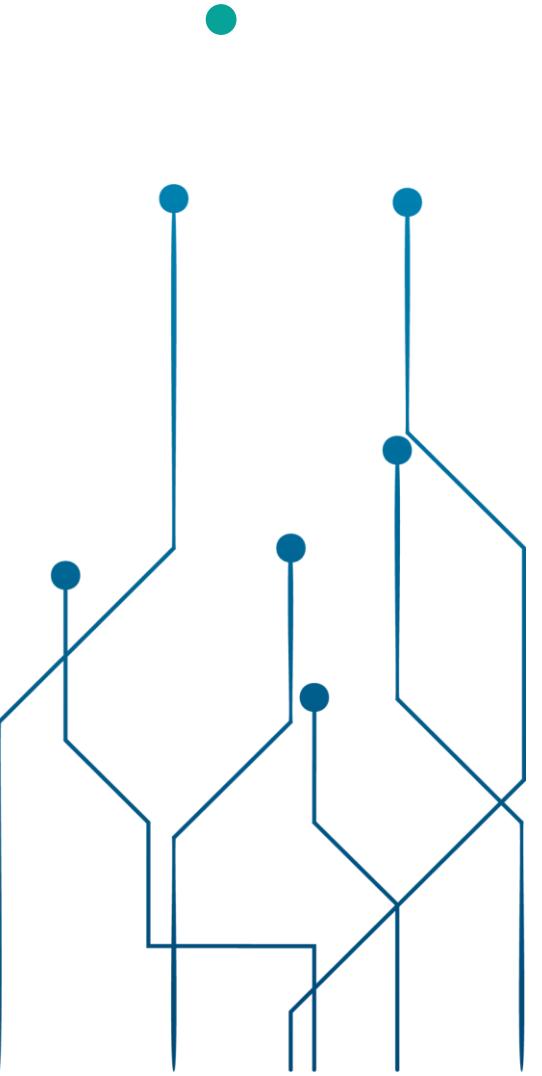
Cilj projekta je realizovati robotizovani sistem za automatsko skupljanje i pražnjenje kanti sa smećem, koji bi trebao da znatno ubrza trenutni sistem prikupljanja otpada, te vrši njegovu dalju distribuciju na ekološki prihvativ način.

## Opis projekta

Garbage collector sistem je sistem koji je moguće upravljati na dva načina. Prvi način je automatski način rada, tj. praćenjem linije pomoću IR senzora. U tom slučaju, robot se nalazi na startnoj poziciji te pomoću IR senzora prati liniju uz najmanju grešku praćenja. Prilikom dolaska na mjesto gdje se nalazi kanta, pomoću ultrasoničnog senzora se detektuje prisustvo kante koja se potom prazni pomoću robotske ruke sastavljene od aluminija i servo motora. Postupak se ponavlja sve dok robot ne dođe do kraja svoje putanje.

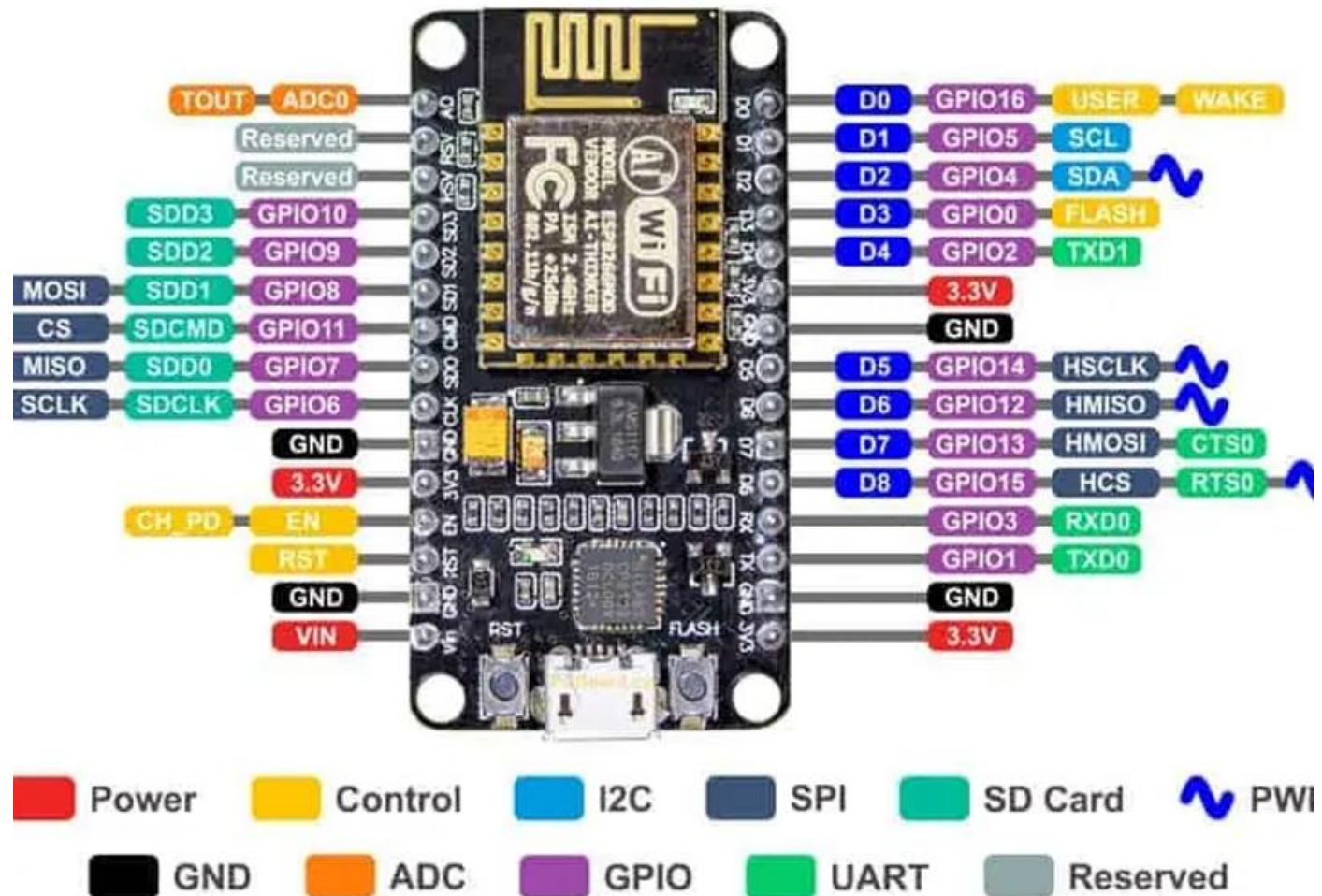
Također, implementiran je i drugi način upravljanja robotom, tzv. manuelni način koji podrazumijeva upravljanje robotom pomoću mobilne aplikacije.

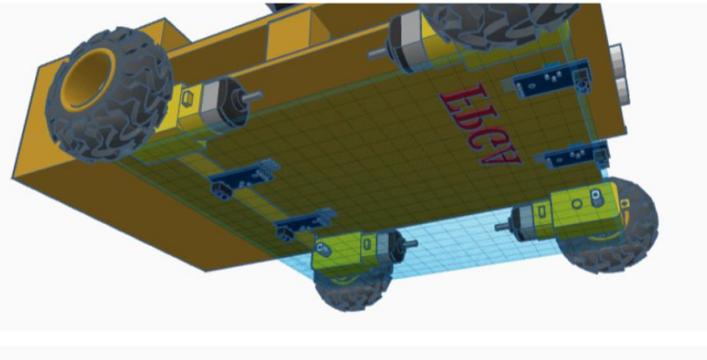
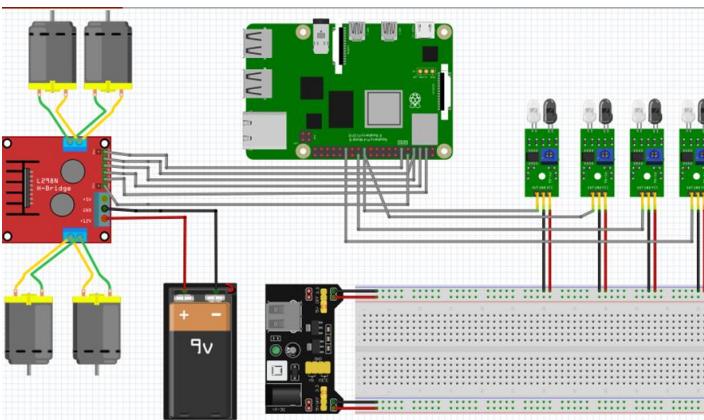
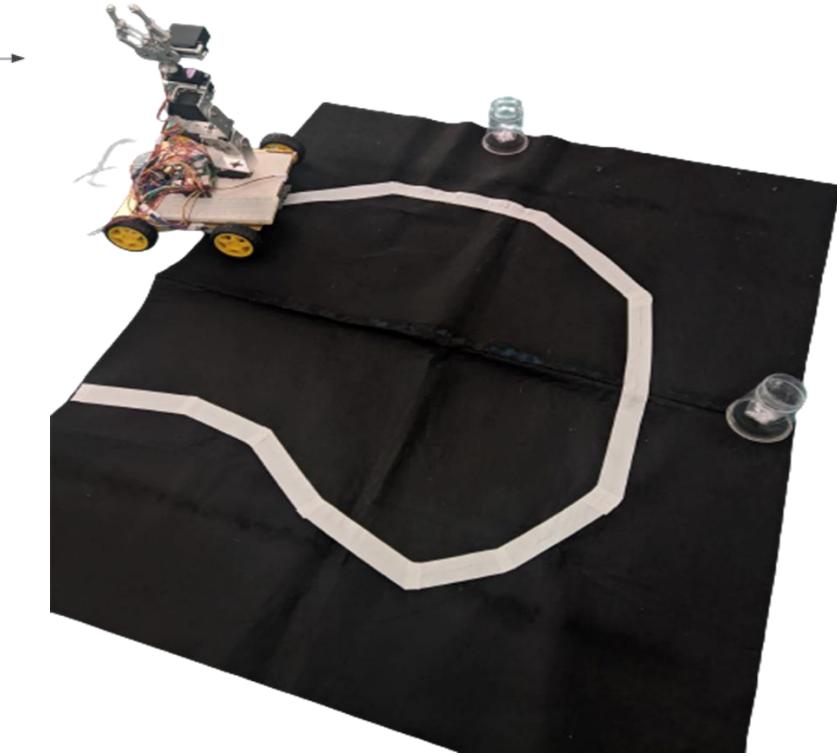
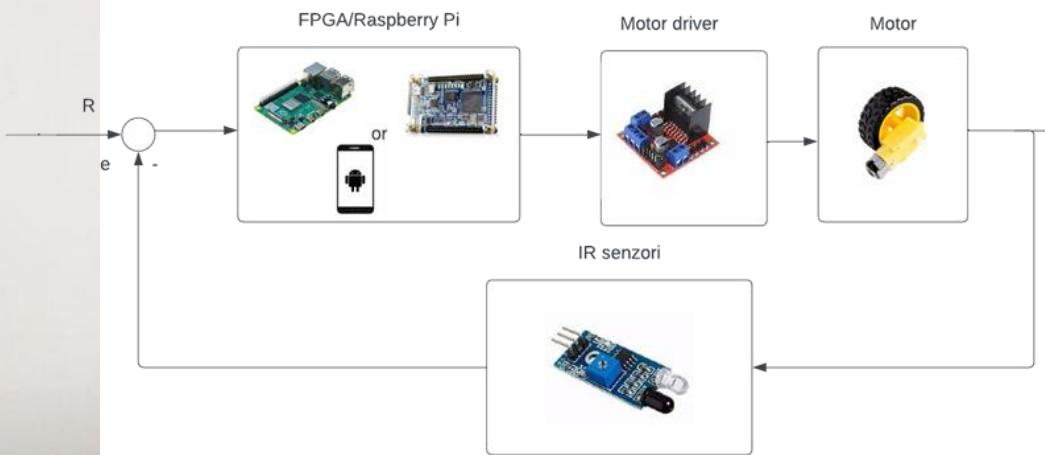
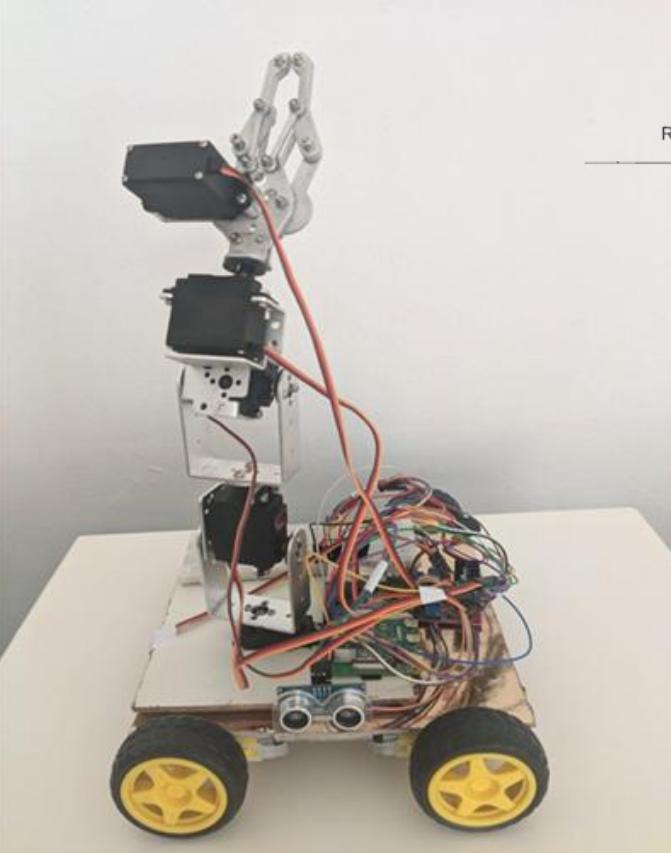
# Hardverske komponente



- **01** Cyclone IV EP4CE6E22C8N / Raspberry Pi 4
- 02** Esp8266 NodeMCU **06** HC-SR04 ultrasonični senzor
- 03** DC motor x4 **07** IR senzor x4
- 04** Servo motor x5 **07** Breadboard sa žicama
- 05** L298N motor driver **08** Baterija

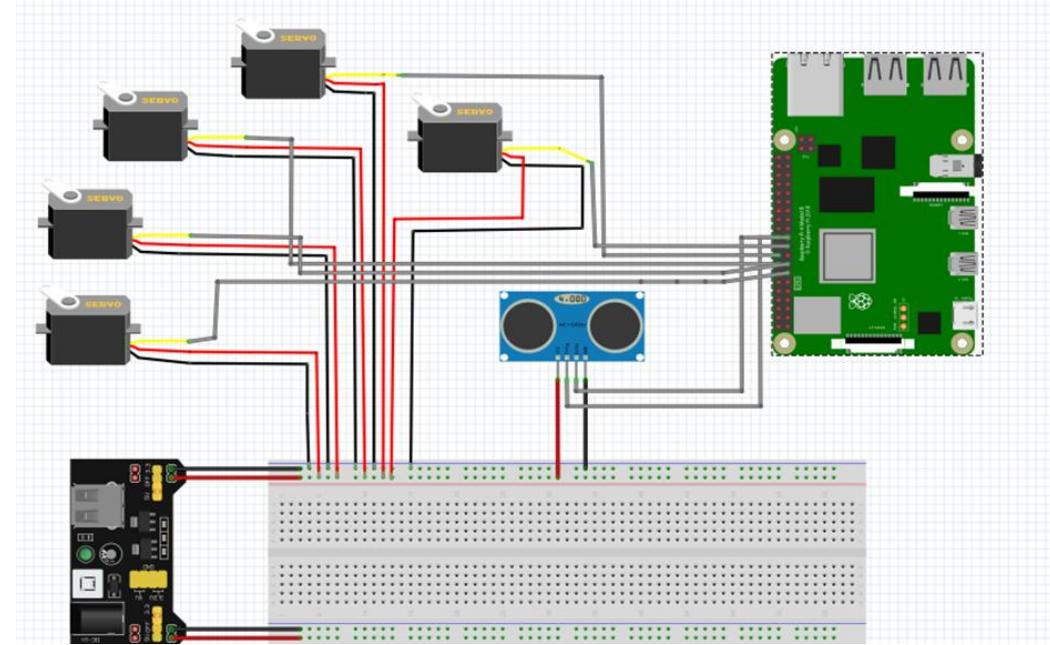
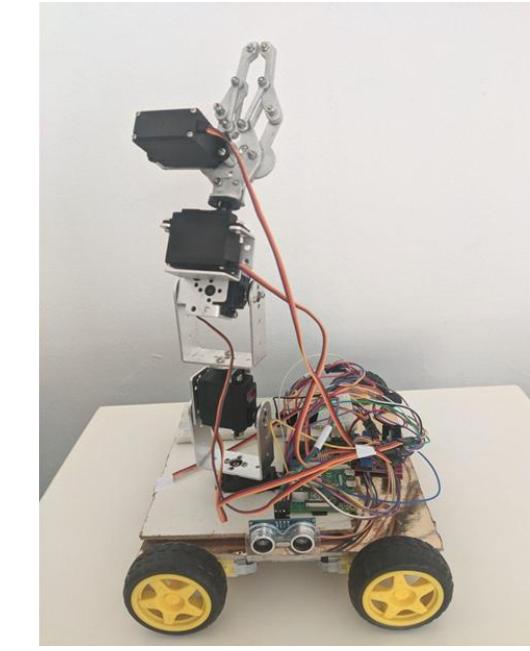
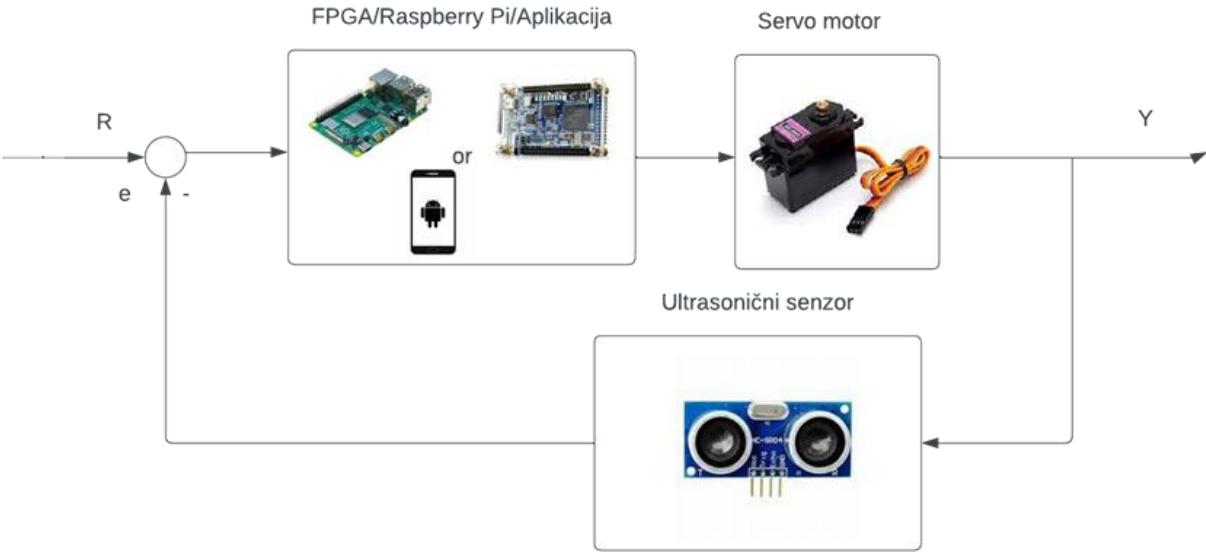
# ESP8266



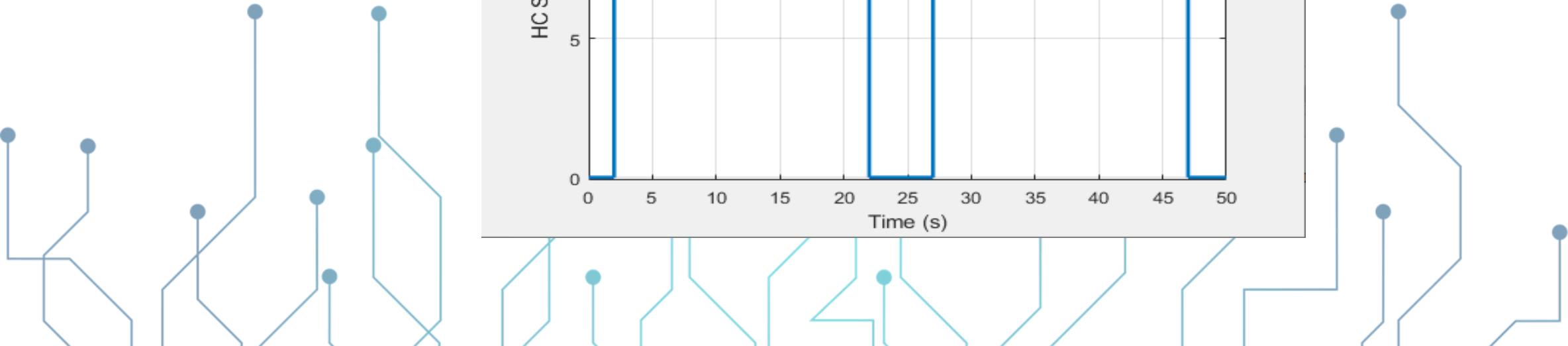
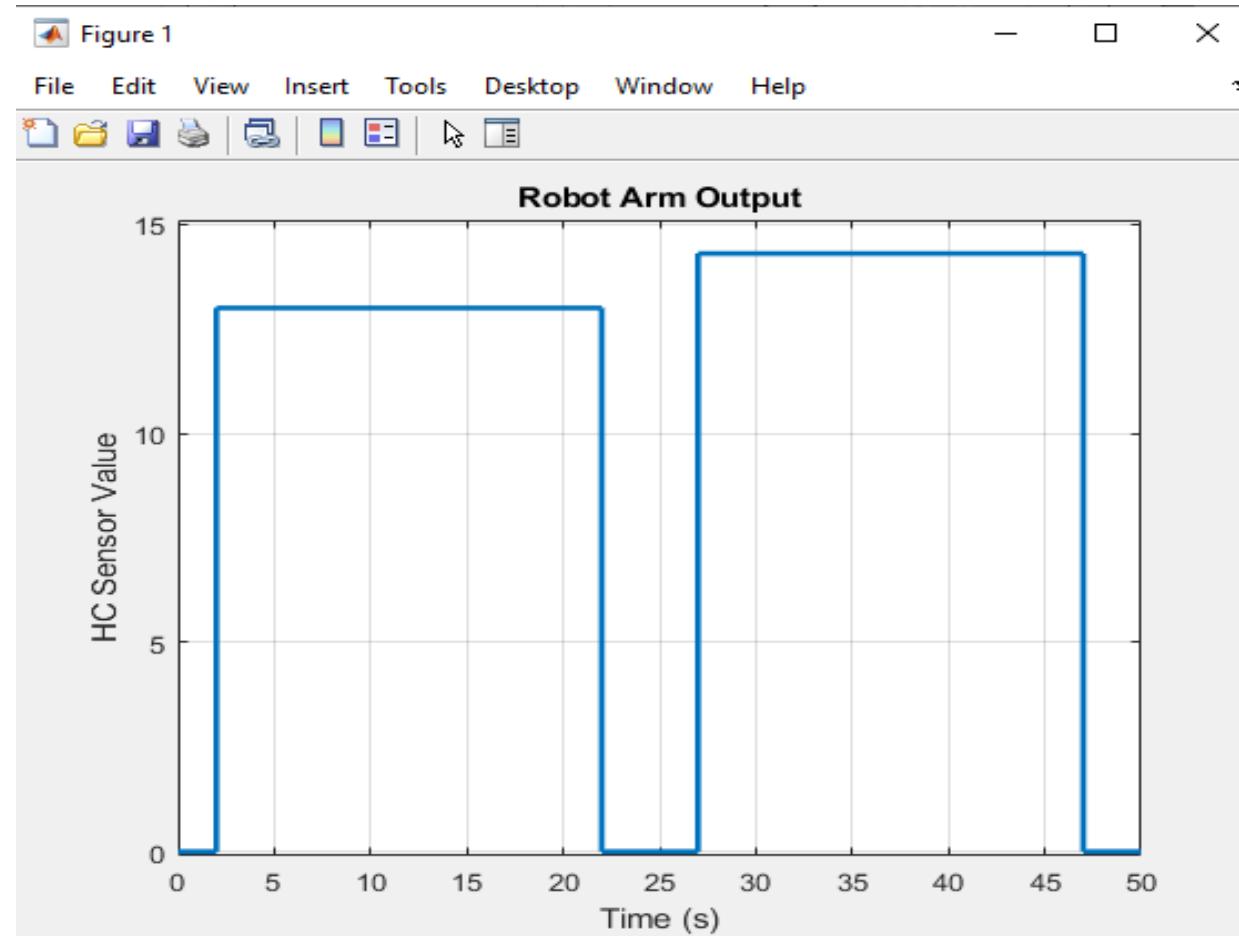


**Primjer aplikacije: Skupljanje otpada - upravljanje kretanjem mobilnog robota**

# Skupljanje otpada - upravljanje robotskom rukom



# Detekcija sa HC ultrasonicnim senzorom

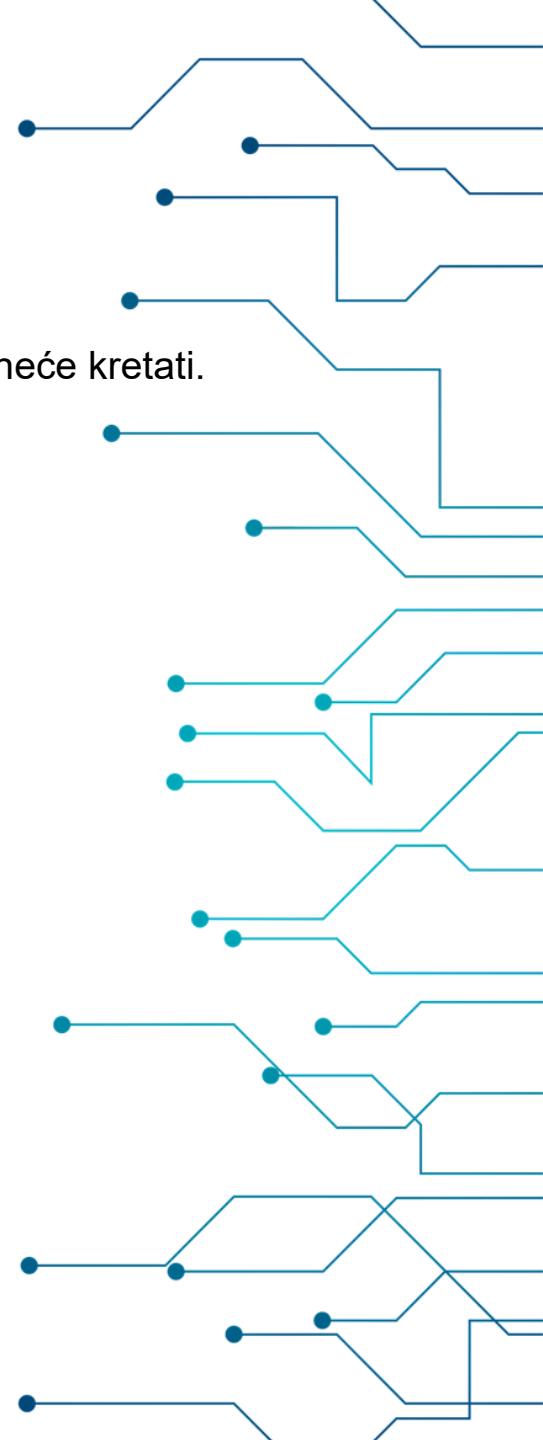


# Logika projektovanja

## Logika pokretanja DC motora(dio)

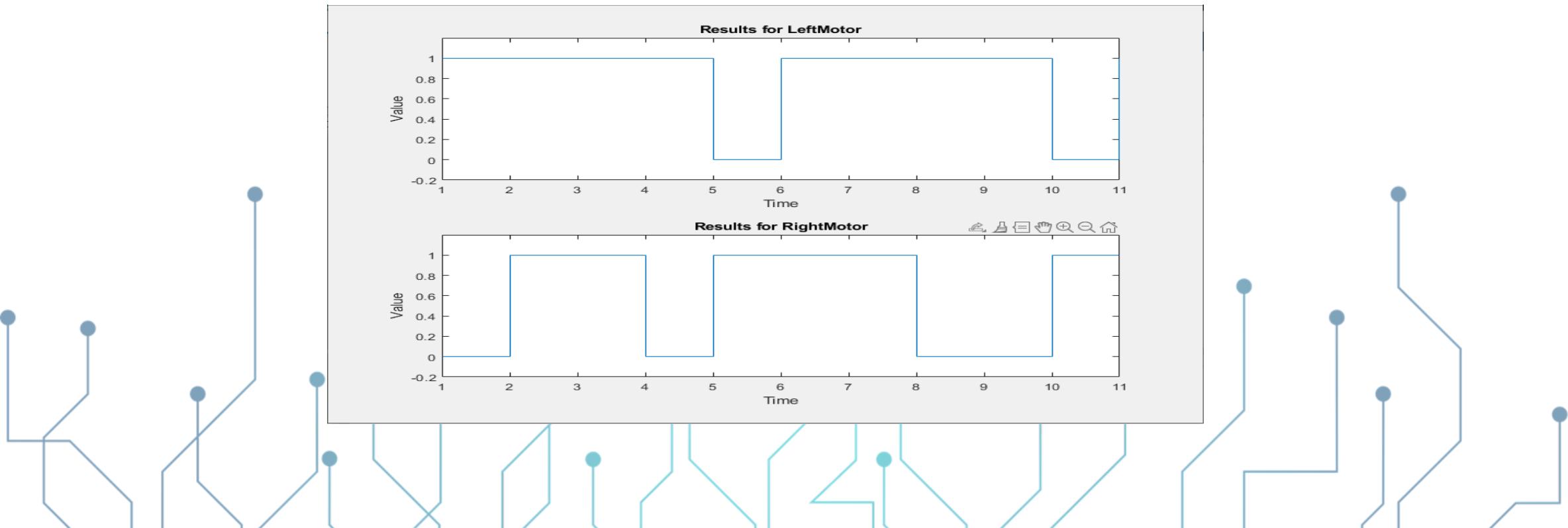
ostale kombinacije motori se neće kretati.

IR1	IR2	IR3	IR4	Desni motori	Lijevi motori	Opis
1	0	0	1	50	50	Ide pravo
0	0	0	1	70	70	Blago desno skreće sa linije
0	0	1	1	70	75	Srednje blago desno skreće sa linije
0	1	1	1	70	80	Oštro desno skreće sa linije
1	1	1	1	0	0	Zaustavlja se

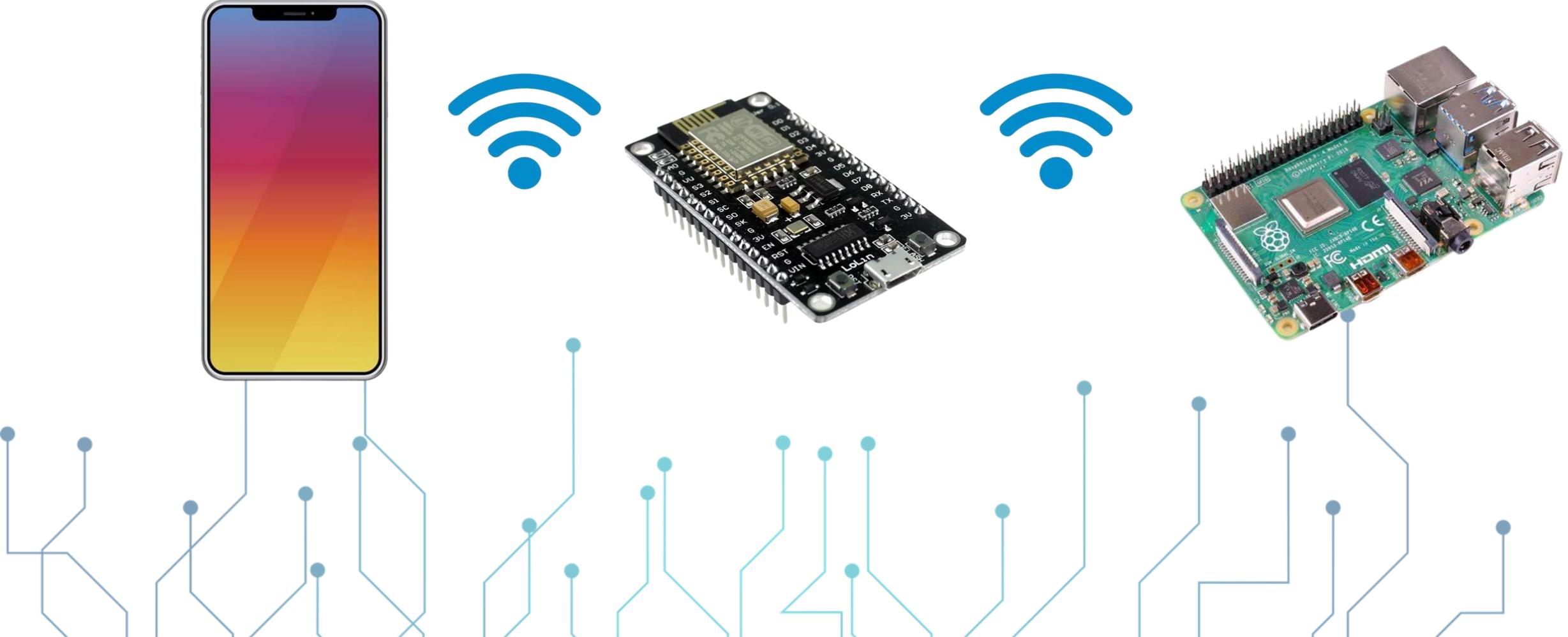


## Graficki prikaz dijagrama za izlaze motora robota pri detekciji IR senzora

	1000	1001	1001	1110	0001	1001	1001	1000	1100	0001	1001
L	1	1	1	1	0	1	1	1	1	0	1
R	0	1	1	0	1	1	1	0	0	1	1



# IoT sistem





# HTTP komunikacija

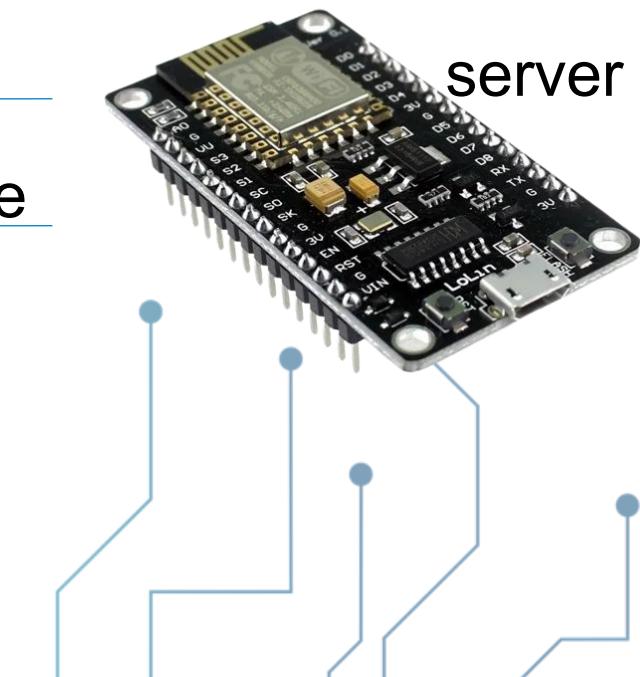
class DirectionForCar( val direction:String )

data class DirectionForRobotArm(  
val angle:Int,  
val motor:String)



Request

Response

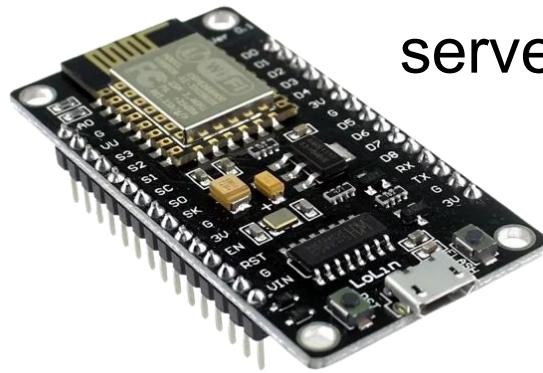


# HTTP komunikacija



Request

Response



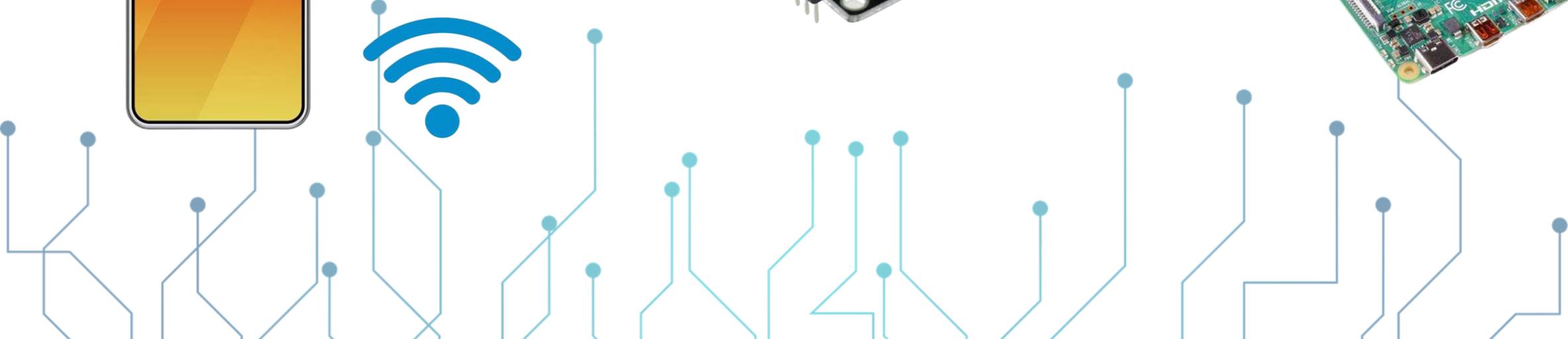
server

Request

Response



server





# Mobilna aplikacija

Internet of Things(IoT) omogućava WiFi kontrolu robota putem mobilne aplikacije.

Mobilna aplikacija, koja je kreirana striktno za android platformu, omogućava korisnicima jednostavan i učinkovit interfejs za upravljanje mobilnog robota.

Uloga android aplikacije je direktna konekcija sa robotom i upravljanje svih dijelova robota, odnosno upravljanje putanje kretanja robota i manipulacija robotskom rukom.

Aplikacija je kreirana preko IntelliJ IDEA uz korištenje Kotlin programskog jezika sa potporom XML-a za dizajn. Podržana je jedino na novijim verzijama android uređaja (10 ili više) i vrši direktnu konekciju na WiFi našeg robota. Koristi jedan od standardnih protokola za komuniciranje na mreži tj. HTTP protokol.

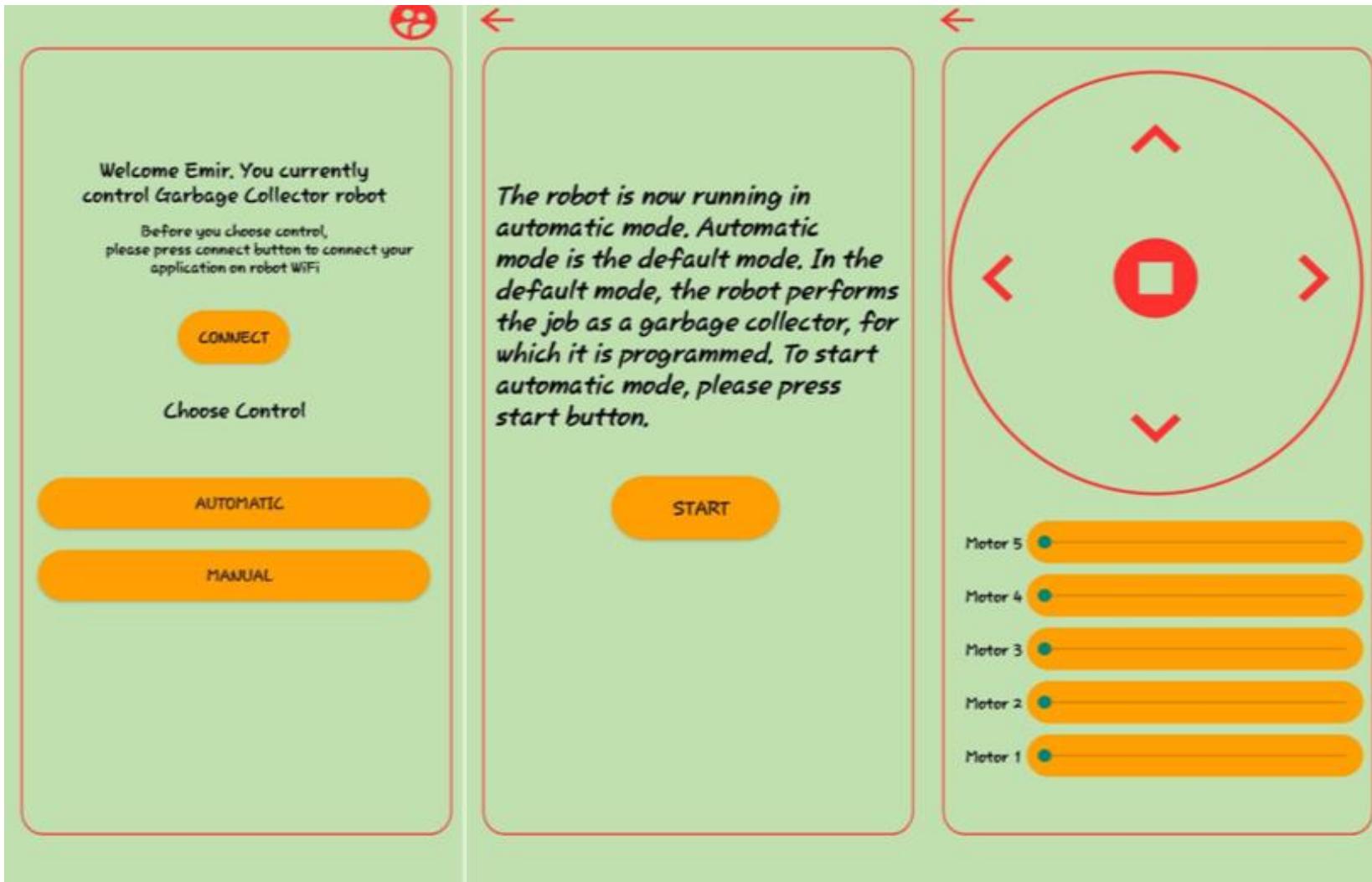
Prilikom pritiska na bilo koji button, u pozadini se vrši slanje HTTP POST zahtjeva na server našeg robota korištenjem JSON formata za serijalizaciju podataka.

Dodana je ORM baza podataka koja vrši pohranu svih korisnika koji se prijave u aplikaciju, kako bi se detaljno pratilo vrijeme i datum korisnika koji upravlja aplikacijom i robotom.

# Mobilna aplikacija-prikaz zaslona

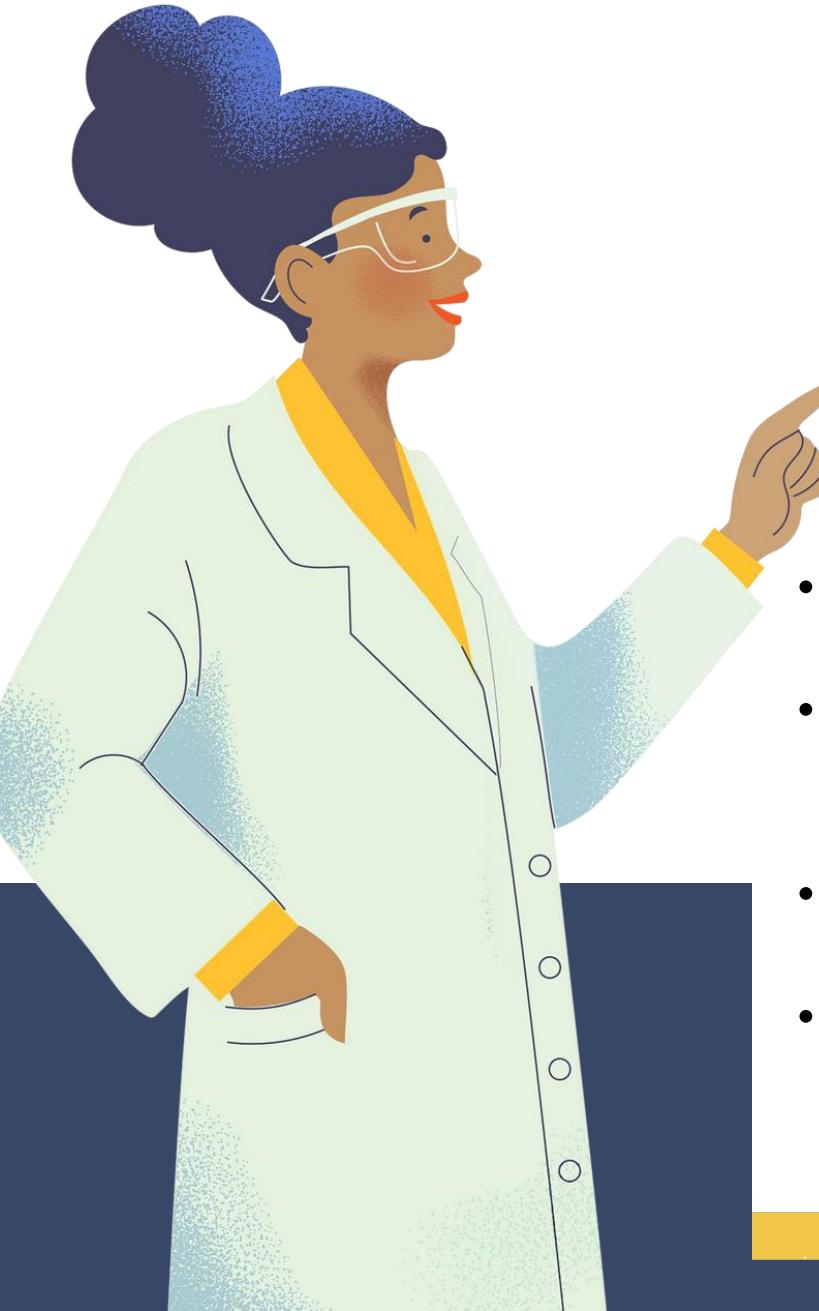


# Mobilna aplikacija-prikaz zaslona





## Primjer 4: Automatizovani sistem selekcije i transporta povrća



# Opis projektnog zadatka:

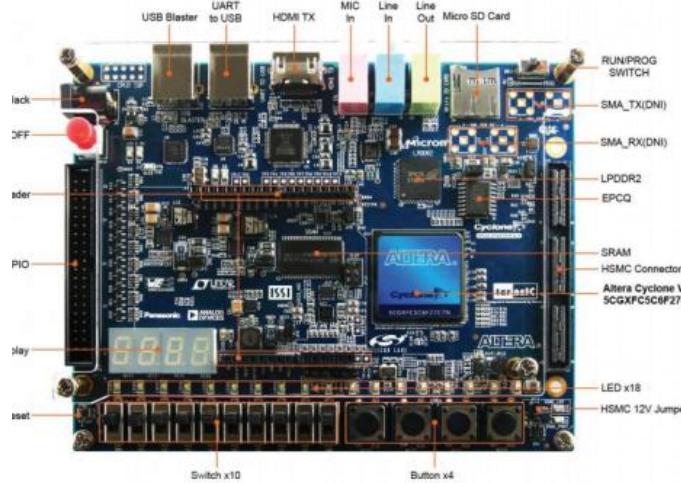
- Pokretna traka: Sistem se pokreće pritiskom na taster i zaustavlja u slučaju iznimnih situacija.
- Detekcija proizvoda: Kamera detektuje proizvod na traci, i algoritam procenjuje zrelost (0 za zeleni proizvod, 1 za zreli proizvod) **korištenjem vještačke inteligencije**
- Pokretanje robota: Pokretni robot se koristi za premještanje proizvoda na osnovu detekcije. Servo motor se aktivira prije nego što se robot pokrene.
- Završna obrada proizvoda: Robot prati liniju dok ne dođe do krajnje destinacije, gdje ispušta proizvod, te se vraća na početnu poziciju.

# Clij:

- Povećanje efikasnosti: Efikasnost i brzina obrade su ključni ciljevi projekta.
- Upotreba FPGA: Prikazati prednosti upotrebe FPGA u odnosu na Raspberry Pi u pogledu brzine obrade i paralelne obrade.
- Optimizacija algoritama: Optimiziranje algoritama za specifične zadatke vezane za detekciju i sortiranje.



# Opis korištenih komponenti



A

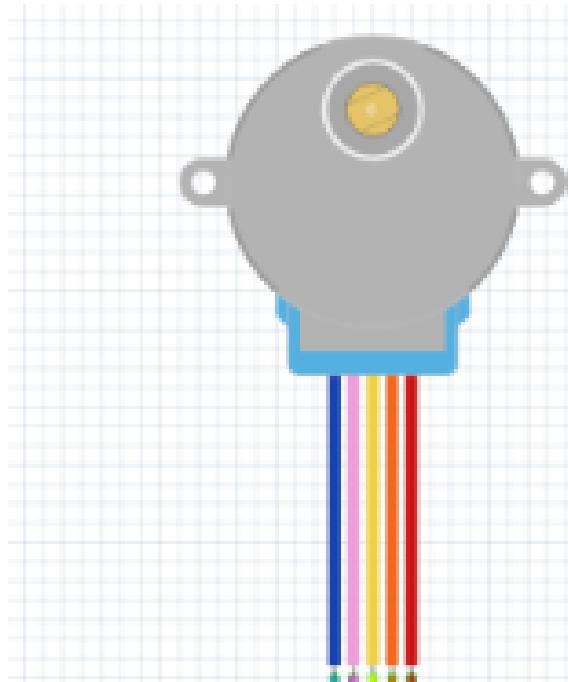


B



C

# Opis korištenih komponenti



D



E



F

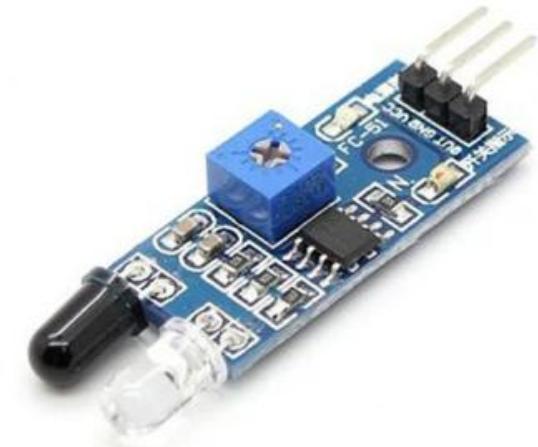
# Opis korištenih komponenti



G

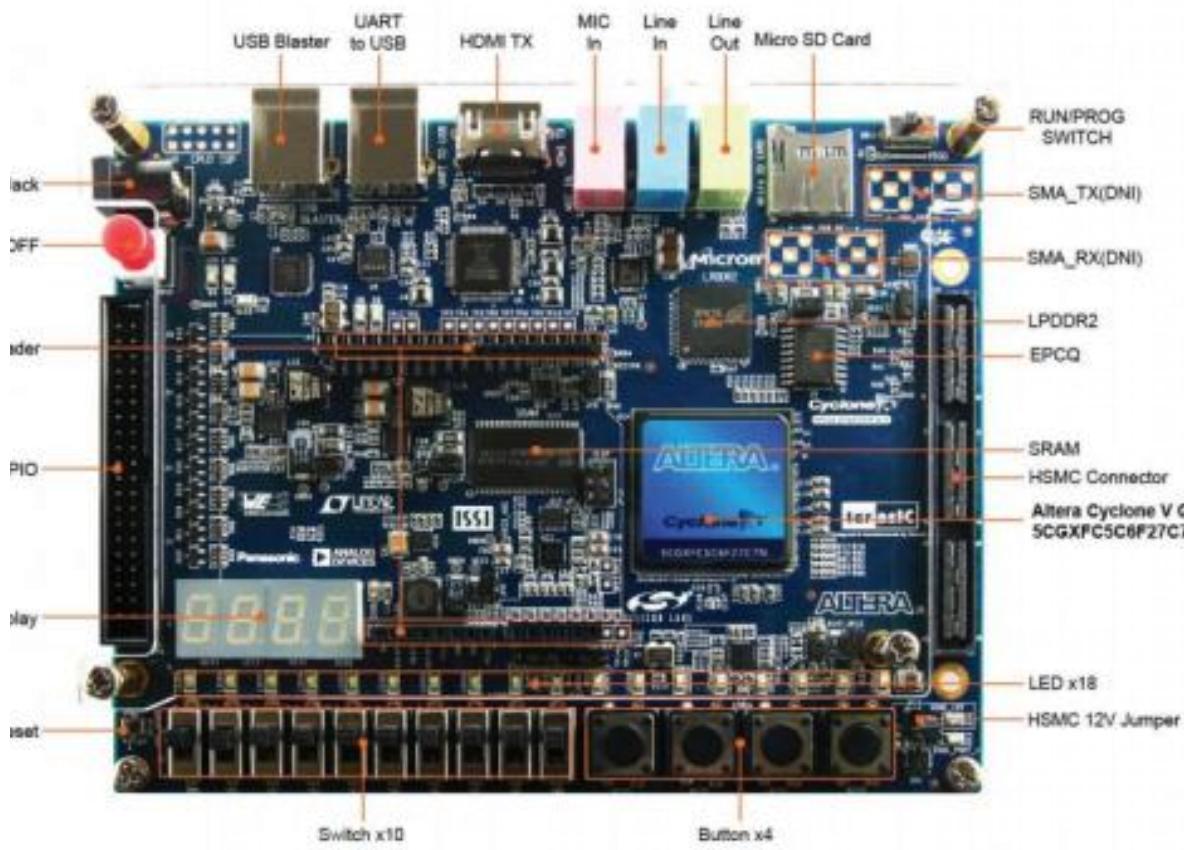


H



F

# FPGA CYCLONE IV



## Cyclone IV FPGA

- Pruža fleksibilnost i procesorsku snagu za upravljanje digitalnim signalima i komponentama.
- Koristi se u industrijskim i inženjerskim aplikacijama za prototipiranje i razvoj digitalnih dizajna.

## Ključne karakteristike Cyclone V:

- Logički elementi: Sadrži veliki broj logičkih elemenata za implementaciju digitalnih dizajna.
- Memorijski blokovi: Ugrađeni memorijski blokovi (RAM, ROM, FIFO) za efikasno skladištenje i obradu podataka.
- DSP blokovi: Blokovi za efikasnu obradu signala.
- I/O priključci: Podrška za razne I/O interfejse za povezivanje s vanjskim uređajima.
- Resursi za generisanje takta: Ugrađeni resursi za upravljanje taktom i vremenskom sinhronizacijom.
- Konfiguracija: Mogućnost konfiguracije putem vanjske memorije ili JTAG za programiranje u stvarnom vremenu.

# RASPBERRY Pi



Raspberry Pi je mali računar na jednoj ploči. Popularno se koristi za obradu slika/videa u realnom vremenu, aplikacije zasnovane na IoT-u i aplikacije robotike.

Ima ARM baziran Broadcom procesor SoC zajedno sa GPU-om (Graphics Processing Unit) na čipu.

Raspberry Pi ploča sadrži:

- programsku memoriju (RAM),
- procesor i grafički čip,
- CPU,
- GPU,
- Ethernet port,
- GPIO pinove,
- Xbee socket,
- UART,
- konektor za izvor napajanja.

## ESP82

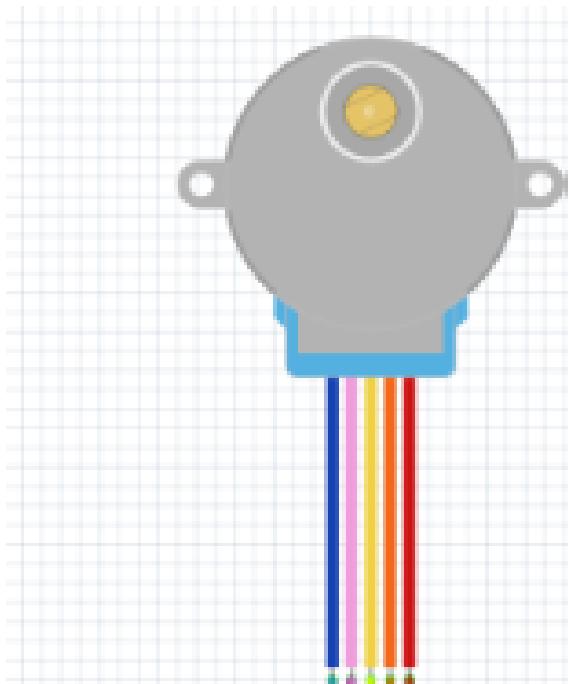


ESP8266 je jeftin, Wi-Fi modul koji omogućava bežično povezivanje uređaja s internetom. Razvio ga je Espressif Systems, a koristi se u IoT (Internet of Things) projektima zbog svoje jednostavnosti i pristupačnosti.

### Ključne karakteristike:

- **Wi-Fi povezivost:** Pruža 2.4 GHz Wi-Fi konekciju s podrškom za TCP/IP protokol, što omogućava direktno povezivanje na internet.
- **Mikrokontroler:** Ugrađeni mikroprocesor (Tensilica Xtensa) omogućava obradu podataka i kontrolu senzora, bez potrebe za dodatnim mikrokontrolerima.
- **GPIO pinovi:** Ima nekoliko GPIO (General Purpose Input/Output) pinova za povezivanje senzora, motora i drugih uređaja.
- **Niska potrošnja energije:** Pogodan za projekte koji zahtijevaju dugotrajan rad na baterijama zbog efikasne upotrebe energije.
- **Jednostavno programiranje:** Programira se putem Arduino IDE ili drugim platformama, uz podršku za standardne programske jezike kao što su C i Python (MicroPython).

## Izlazi: STEPPER MOTOR



Stepper motor je elektromehanički uređaj koji pretvara električnu snagu u mehaničku. To su istosmjerni motori bez četkica koji jedan obrtaj dijele u odgovarajući broj jednakih koraka. Pretvaraju strujne impulse u fiksne inkremente ugaonog pomjeraja - korake. Najčešće dolaze zajedno sa ULN2003 drajverom.

Postoje 3 moda rada ovih motora:

- Wave drive
- Half drive
- Full drive

## Izlazi: SERVO MOTOR



Servo motor je vrsta motora koji se može okretati s velikom preciznošću.

Obično se ova vrsta motora sastoji od upravljačkog kruga koji daje povratnu informaciju o trenutnom položaju osovine motora.

Servo motor radi na principu PWM (Pulse width modulation).

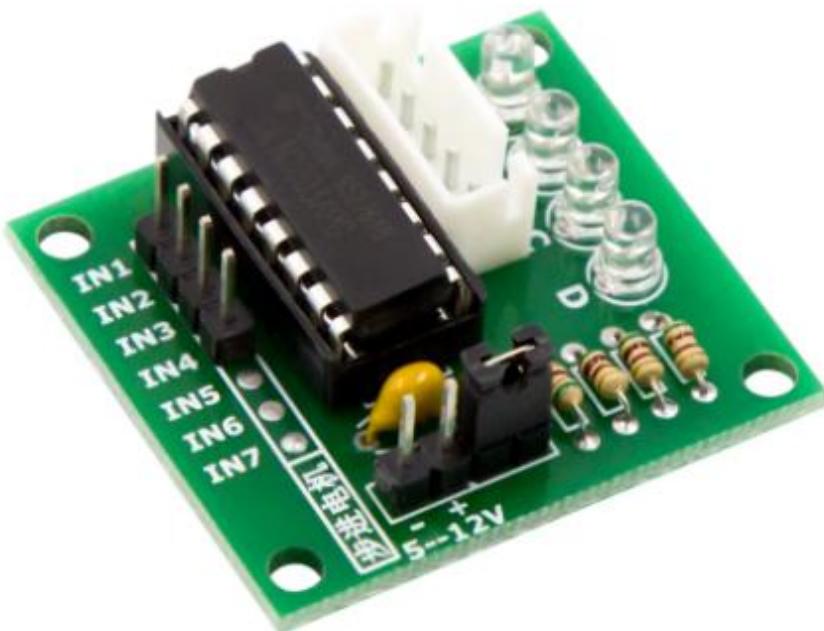
Servo motor se može rotirati od 0 do 180 stepeni, ali može ići do 210 stepeni, u zavisnosti od proizvodnje.

## Izlazi: PARALLAX CONTINOUS ROTATION SERVO



Parallax Continuous Rotation Servo je specifičan servo motor koji omogućava neprekidnu rotaciju osovine u oba smjera. Za razliku od standardnih servo motora koji mogu rotirati samo unutar ograničenog ugla, ovaj motor kontrolira brzinu i smjer rotacije putem PWM signala, umjesto položaja osovine. Zbog te karakteristike, često se koristi u projektima poput mobilnih robotičkih vozila i drugih aplikacija gdje je potrebna kontinuirana rotacija. Također, jednostavno se integriše s platformama poput Arduino, BASIC Stamp i micro, što ga čini pogodnim za široku primjenu u robotici.

## Izlazi: ULN2003 STEPPET MOTOR DRIVER



Stepper motor obično dolazi sa ULN2003 drajverskom pločom.

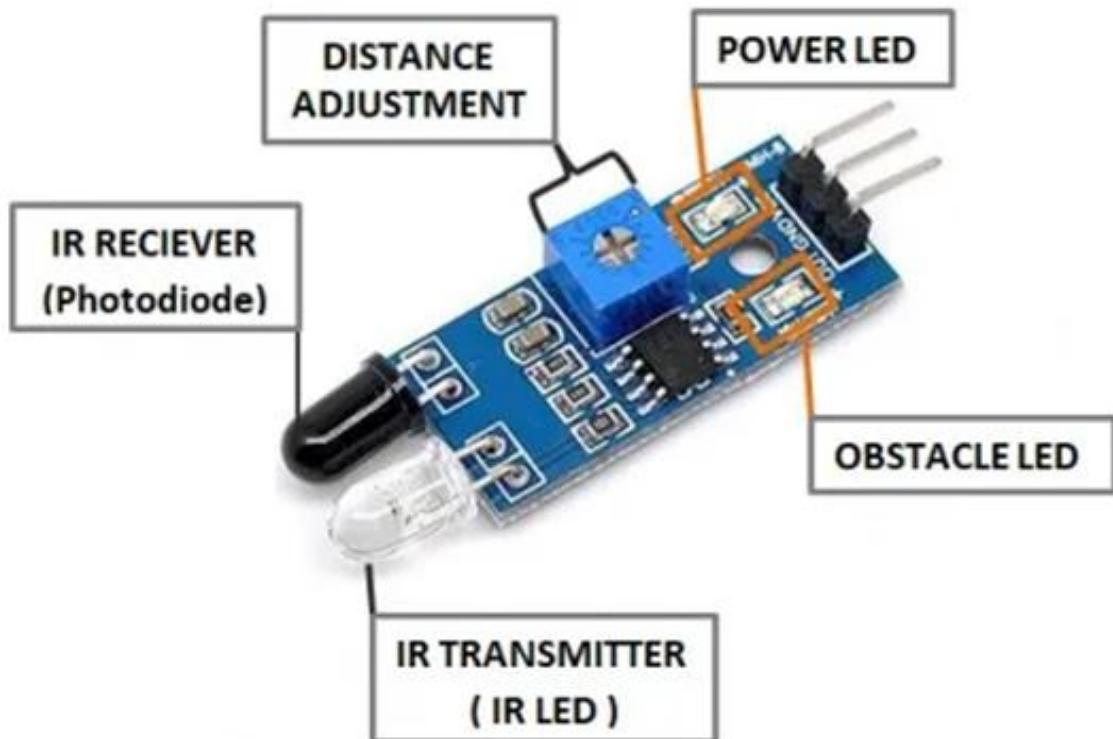
ULN2003 isporučuje veću struju od jednog tranzistora i omogućava niskonaponskom niskostrujnom izlazu mikrokontrolera da pokreće stepper motore velike struje.

## Ulazi: IMI WEBCAM W88S



IMI WEBCAM W88S je web kamera dizajnirana za visokokvalitetno video snimanje i video pozive. Opremljena je HD rezolucijom koja omogućava jasne i oštре slike, što je čini pogodnom za online sastanke, konferencije, streaming i druge potrebe za video komunikacijom. Kamera ima ugrađen mikrofon s funkcijom redukcije šuma, što pomaže u poboljšanju kvaliteta zvuka tokom video poziva.

## Ulazi: IR SENZOR



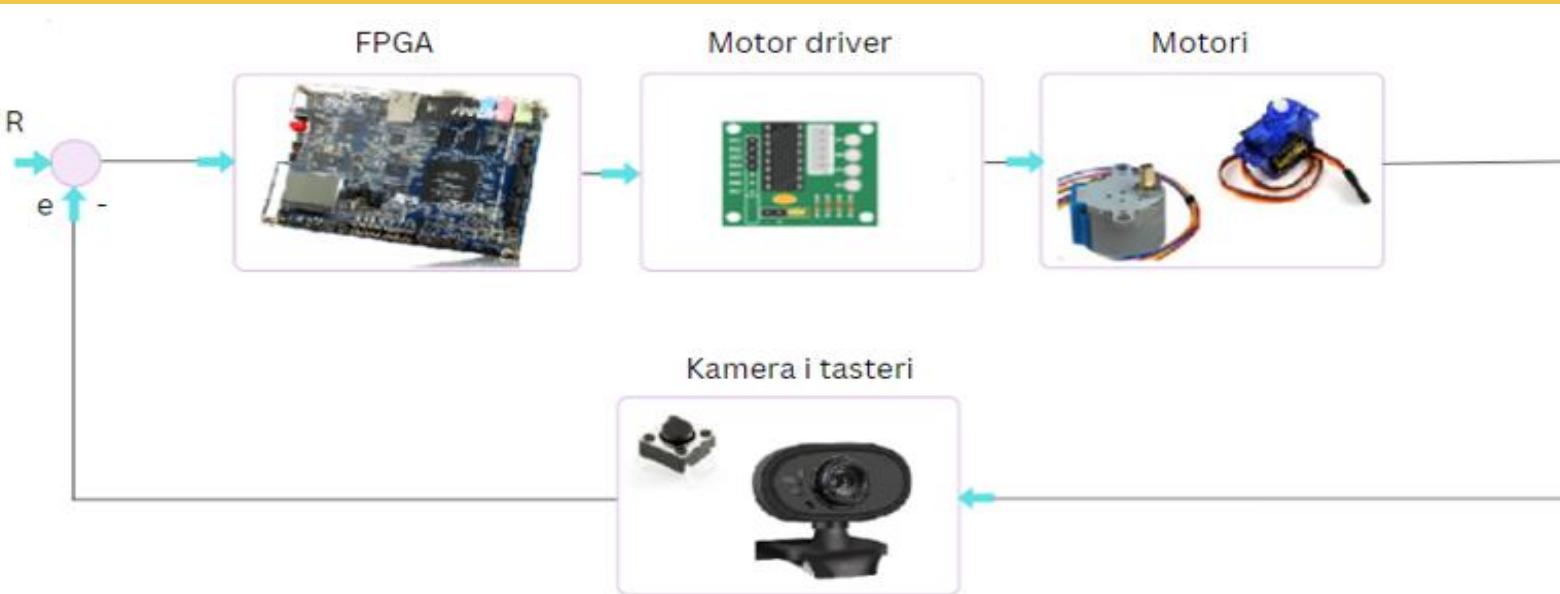
Infracrveni (IR) senzor je elektronski uređaj koji mjeri i detektuje infracrveno zračenje u svom okruženju.

IR je nevidljiv ljudskom oku, jer je njegova talasna dužina duža od talasne dužine vidljive svetlosti.

Sve što emituje toplotu daje infracrveno zračenje.

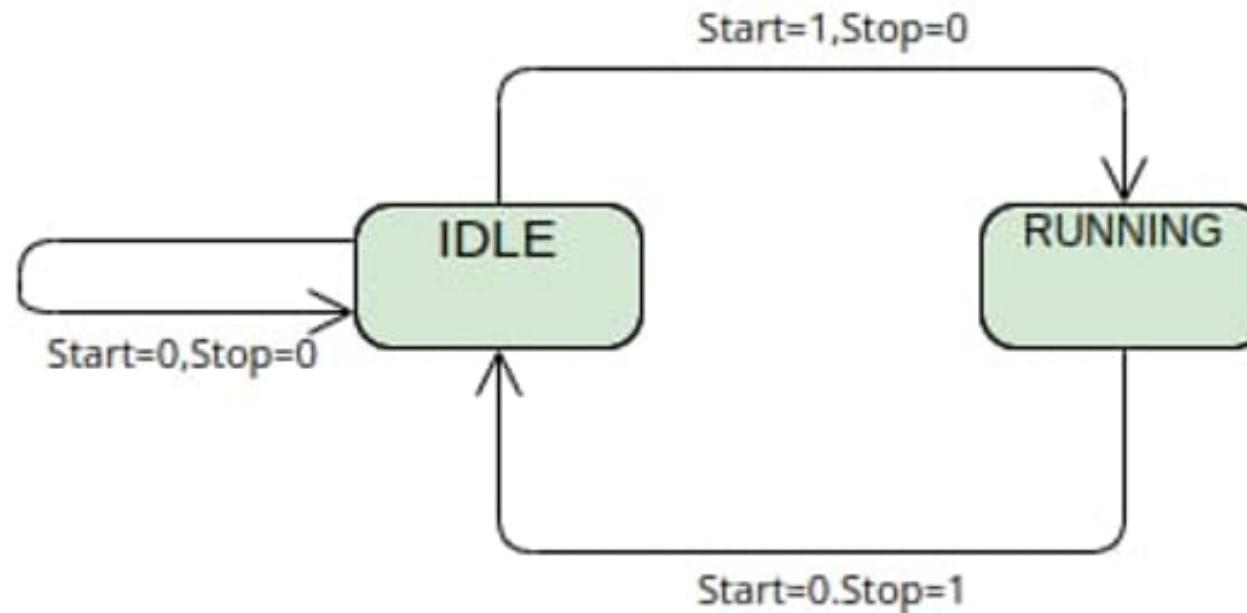
# Algoritam upravljanja

## Upravljanje pokretnom trakom

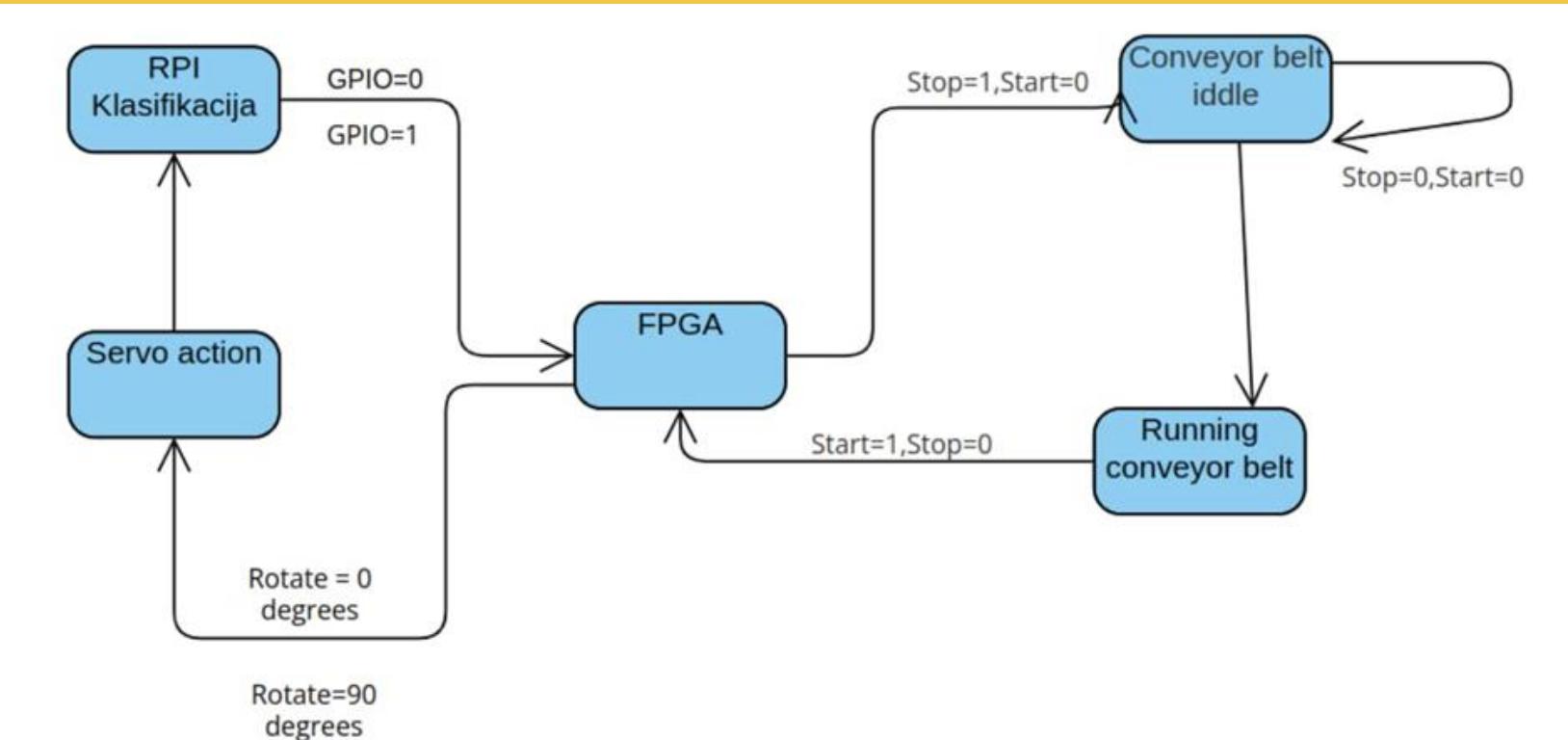


FPGA	
ULAZI	IZLAZI
Kamera	Stepper motor
Tasteri	Servo motor

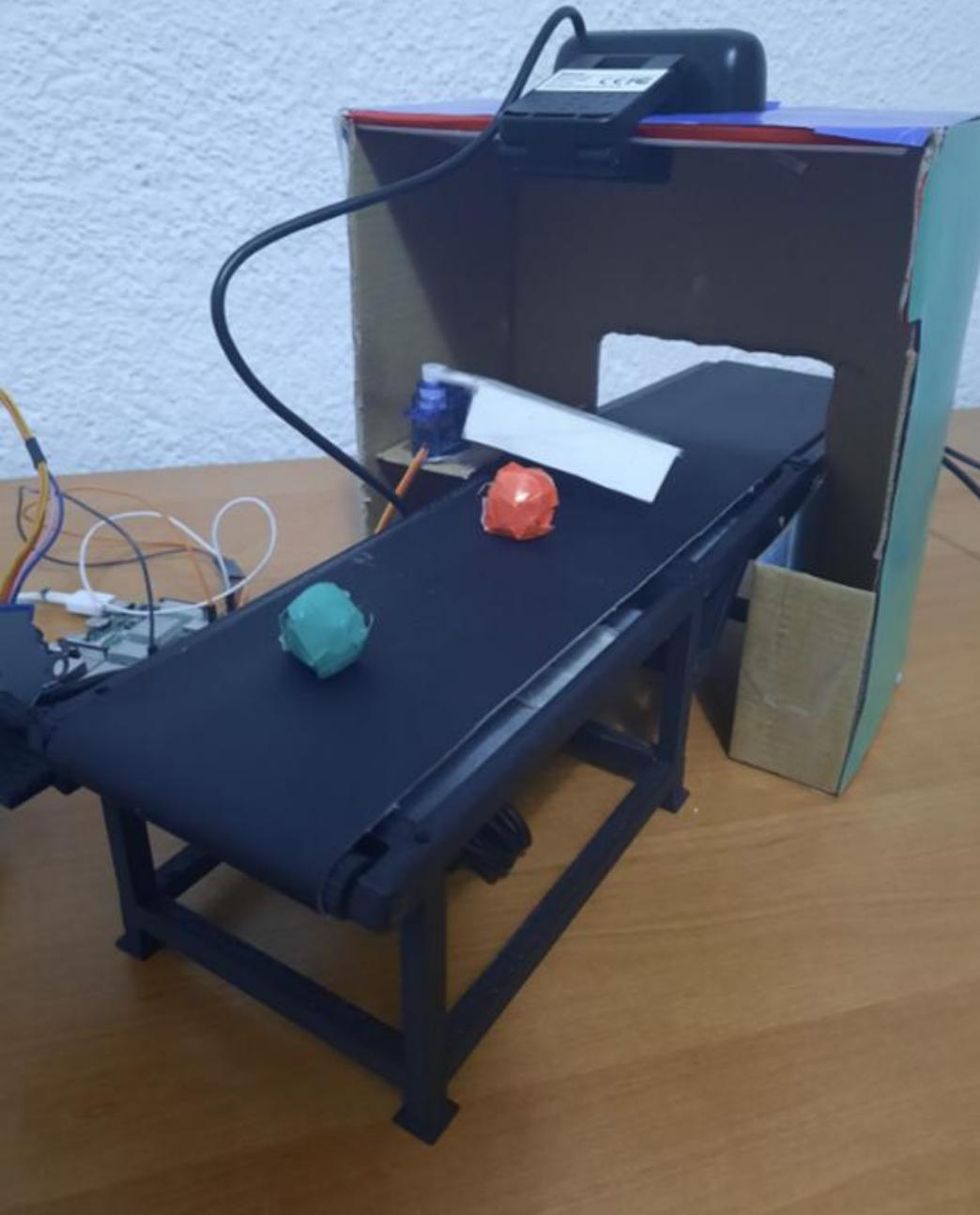
# Mašina konačnog stanja za pokretnu traku



# Mašina konačnog stanja kompletног sistema

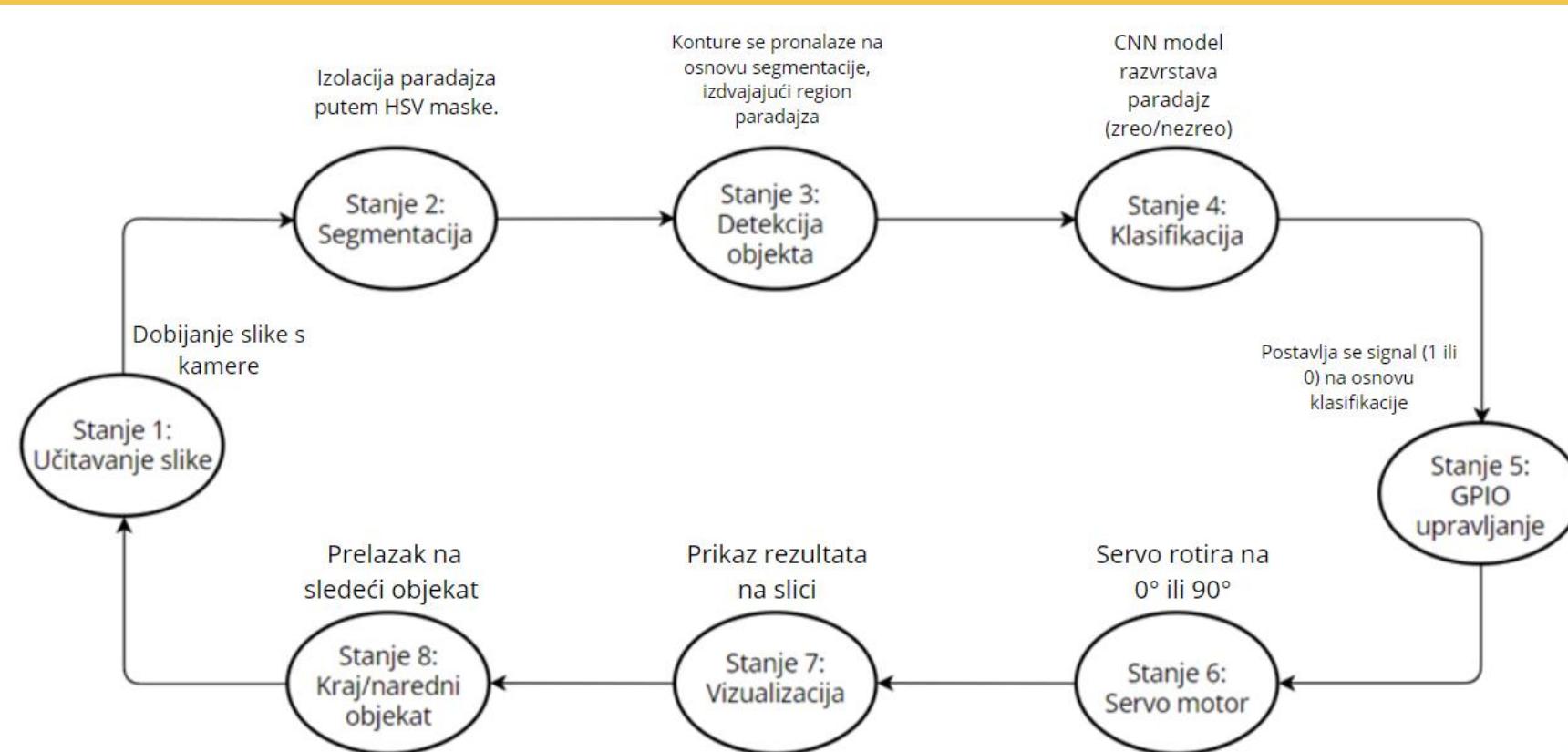


# Maketa sistema

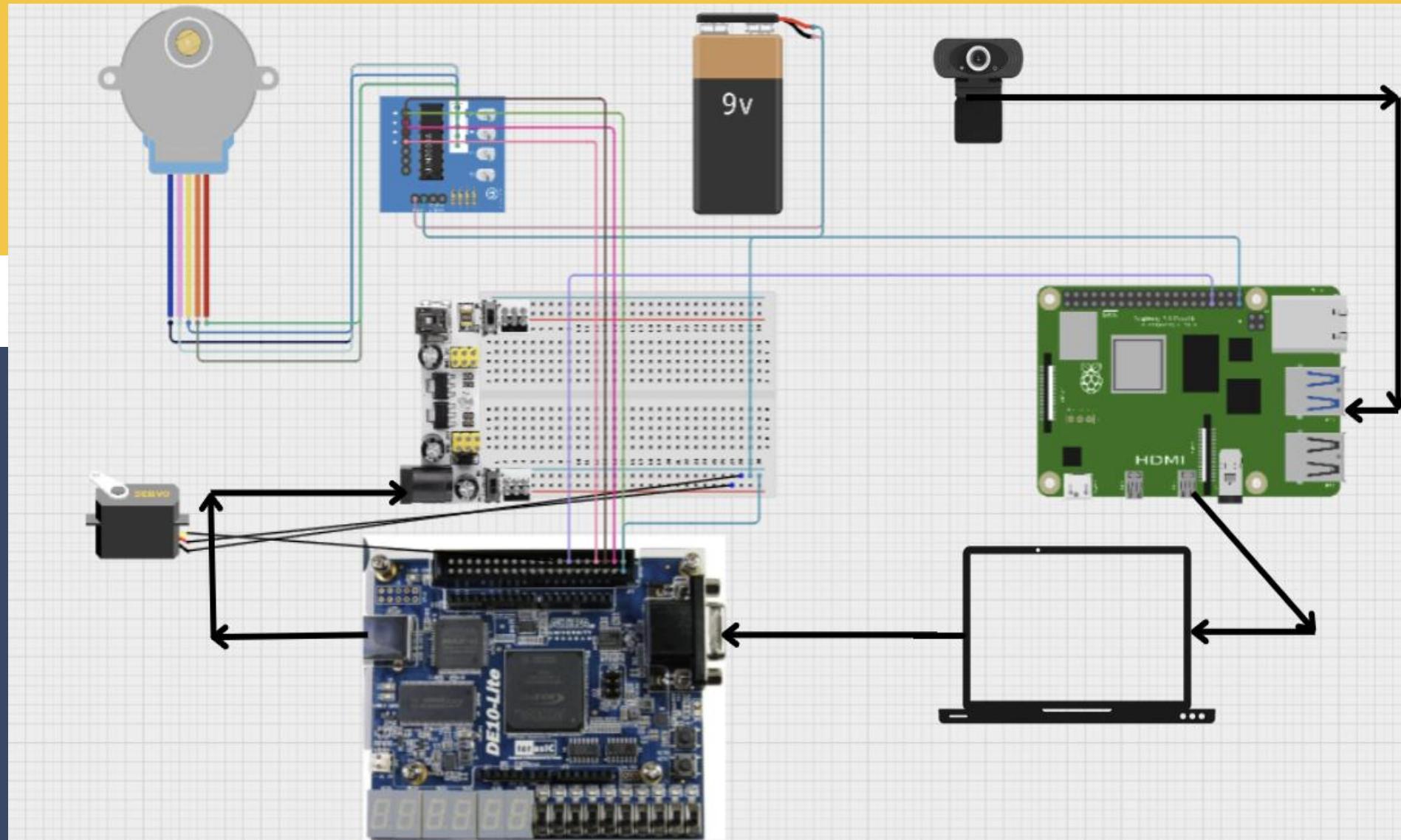


Faza	Komponente	Ulaz	Obrada	Izlaz	Akcija
1	Učitavanje slike	Slika sa kamere	Provjera formata i veličine slike	Pripremljena slika	
2	Segmentacija	Pripremljena slika	Konverzija u HSV, primjena maski	Maske za zrele/nezrele	
3	Detekcija objekta	Maske	Pronalaženje kontura	Koordinate objekta (ROI)	
4	Klasifikacija	Izdvojeni ROI	CNN model predviđa stanje	Klasifikacija (zreo/nezreo)	
5	GPIO upravljanje	Rezultat klasifikacije	Postavljanje GPIO pinova	GPIO stanje (high/low)	Ako zreo: GPIO=High  Ako nezreo: GPIO=Low
6	Slanje signala na FPGA	Servo motor	Ovisno o stanju GPIO pinova	Položaj servo motora	Ako GPIO High: 0°  Ako GPIO Low: 90°, zatim povratak na 0°
7	Vizualizacija	Originalna slika, klasifikacija	Crtanje rubova i teksta	Slika sa iscrtanimi rubovima	
8	Kraj	-	-	-	Čekanje novog objekta ili se zatvara

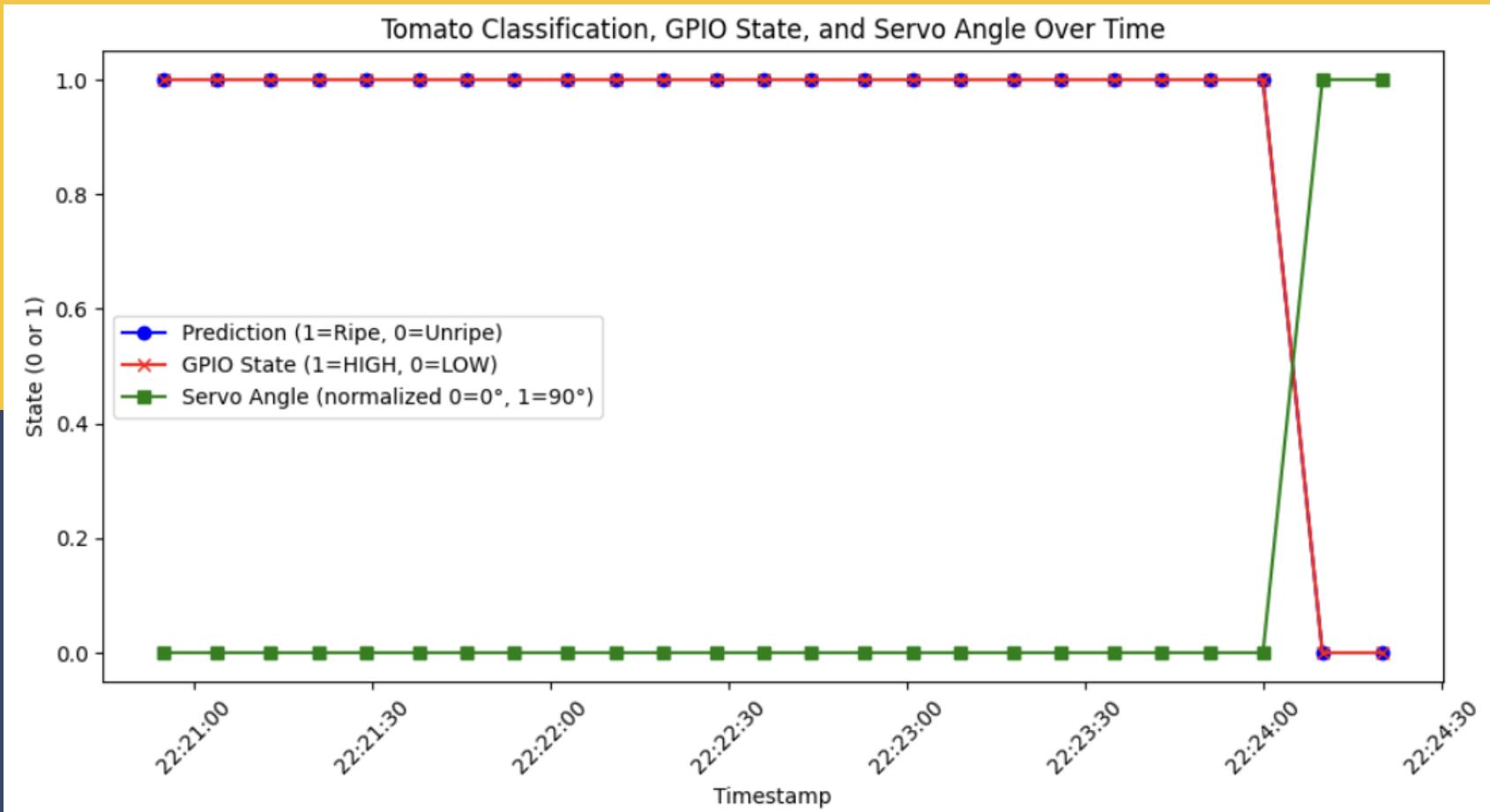
# Mašina konačnog stanja za pokretanje servo motora



# Šema povezivanja komponenti na FPGA i RPI

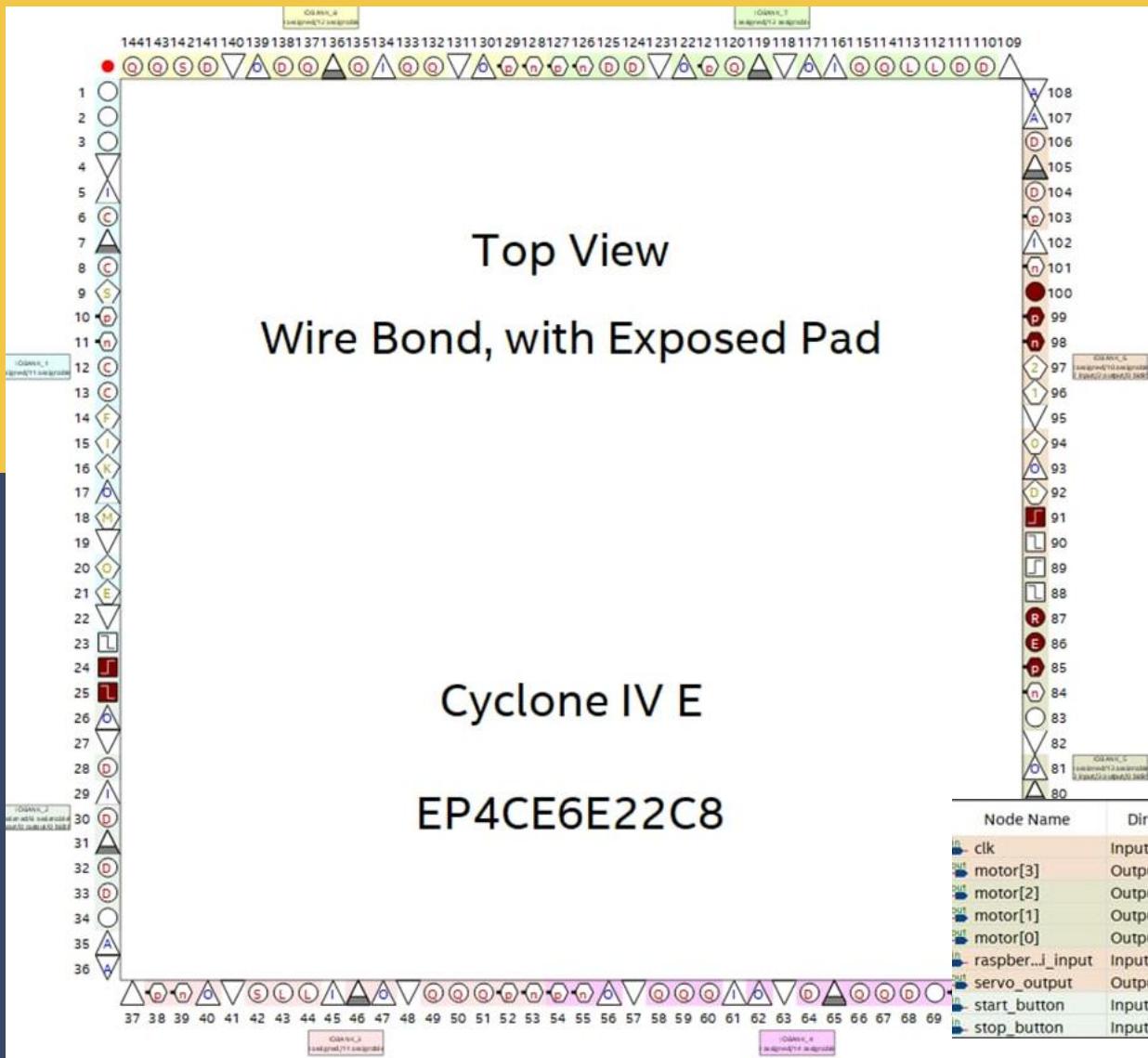


# Grafički prikaz odziva



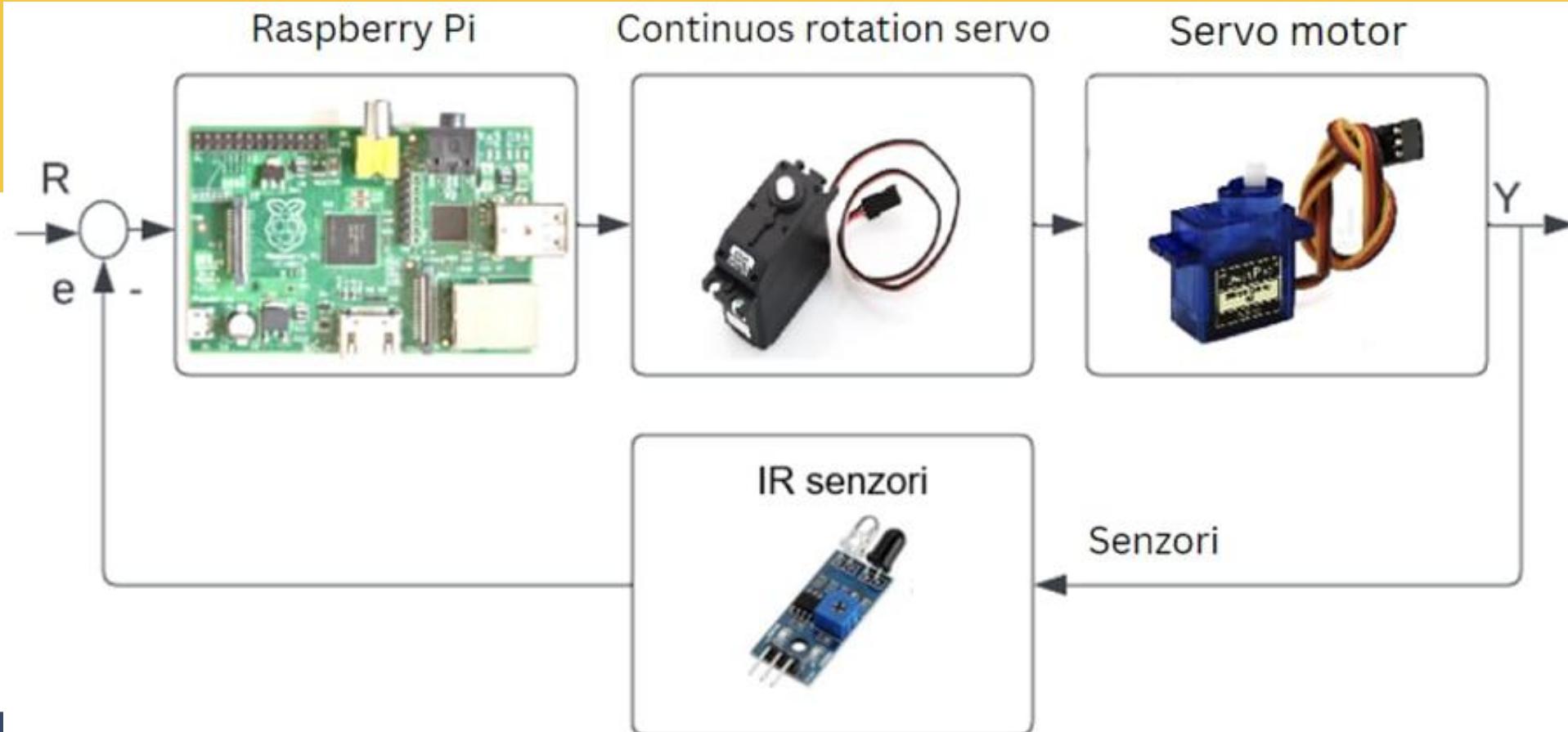
Rezultat  
akvizicije

# Raspored pinova na Cyclone IV

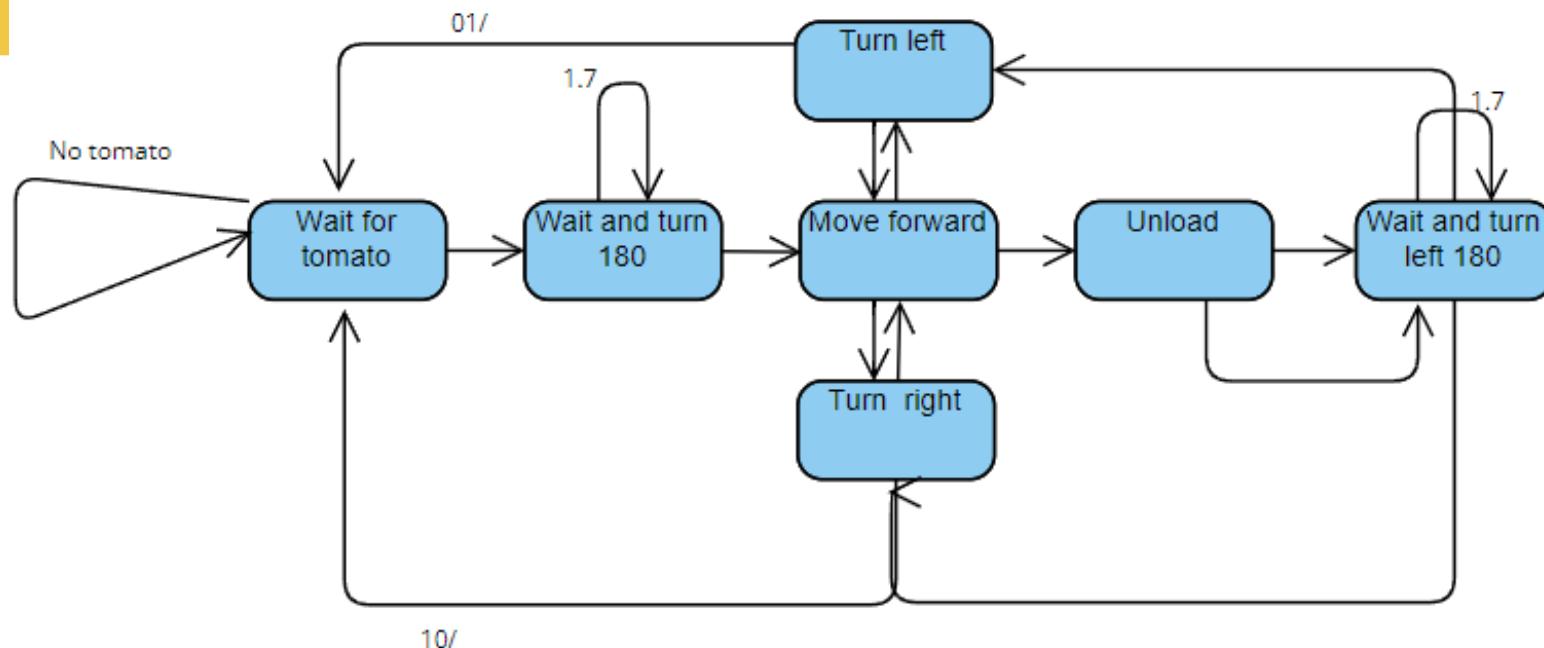


Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Locator	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Electrical Preset
in_clk	Input	PIN_91	6	B6_N0	PIN_91	2.5 V		8mA (default)			
out_motor[3]	Output	PIN_99	6	B6_N0	PIN_99	2.5 V		8mA (default)	2 (default)		
out_motor[2]	Output	PIN_87	5	B5_N0	PIN_87	2.5 V		8mA (default)	2 (default)		
out_motor[1]	Output	PIN_86	5	B5_N0	PIN_86	2.5 V		8mA (default)	2 (default)		
out_motor[0]	Output	PIN_85	5	B5_N0	PIN_85	2.5 V		8mA (default)	2 (default)		
in_raspberry_input	Input	PIN_98	6	B6_N0	PIN_98	2.5 V		8mA (default)			
out_servo_output	Output	PIN_100	6	B6_N0	PIN_100	2.5 V		8mA (default)	2 (default)		
in_start_button	Input	PIN_25	2	B2_N0	PIN_25	2.5 V		8mA (default)			
in_stop_button	Input	PIN_24	2	B2_N0	PIN_24	2.5 V		8mA (default)			

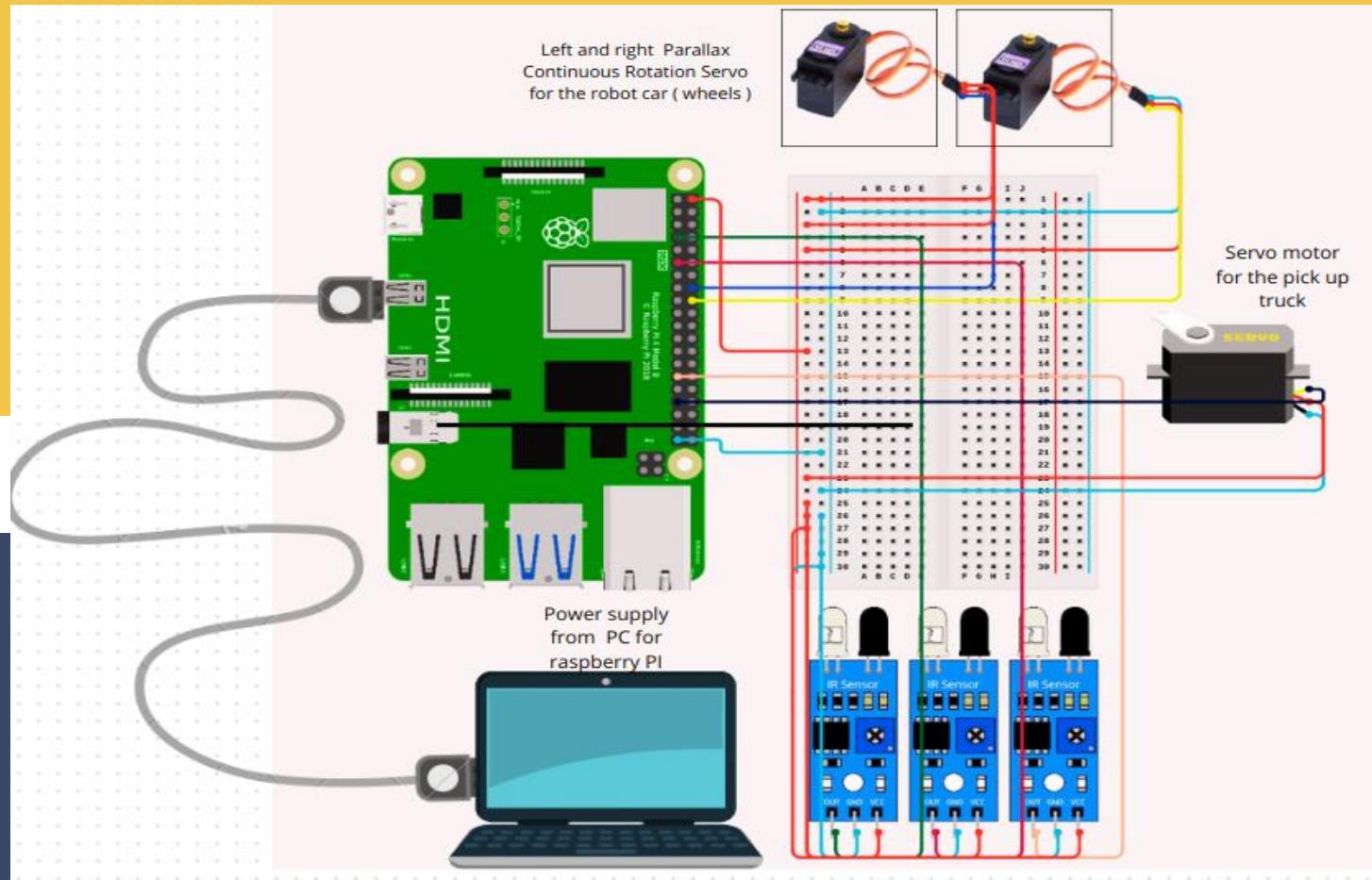
# Algoritam upravljanja Upravljanje mobilnim robotom



# Mašina konačnog stanja za mobilnog robota



# Šema povezivanja komponenti na Raspberry PI

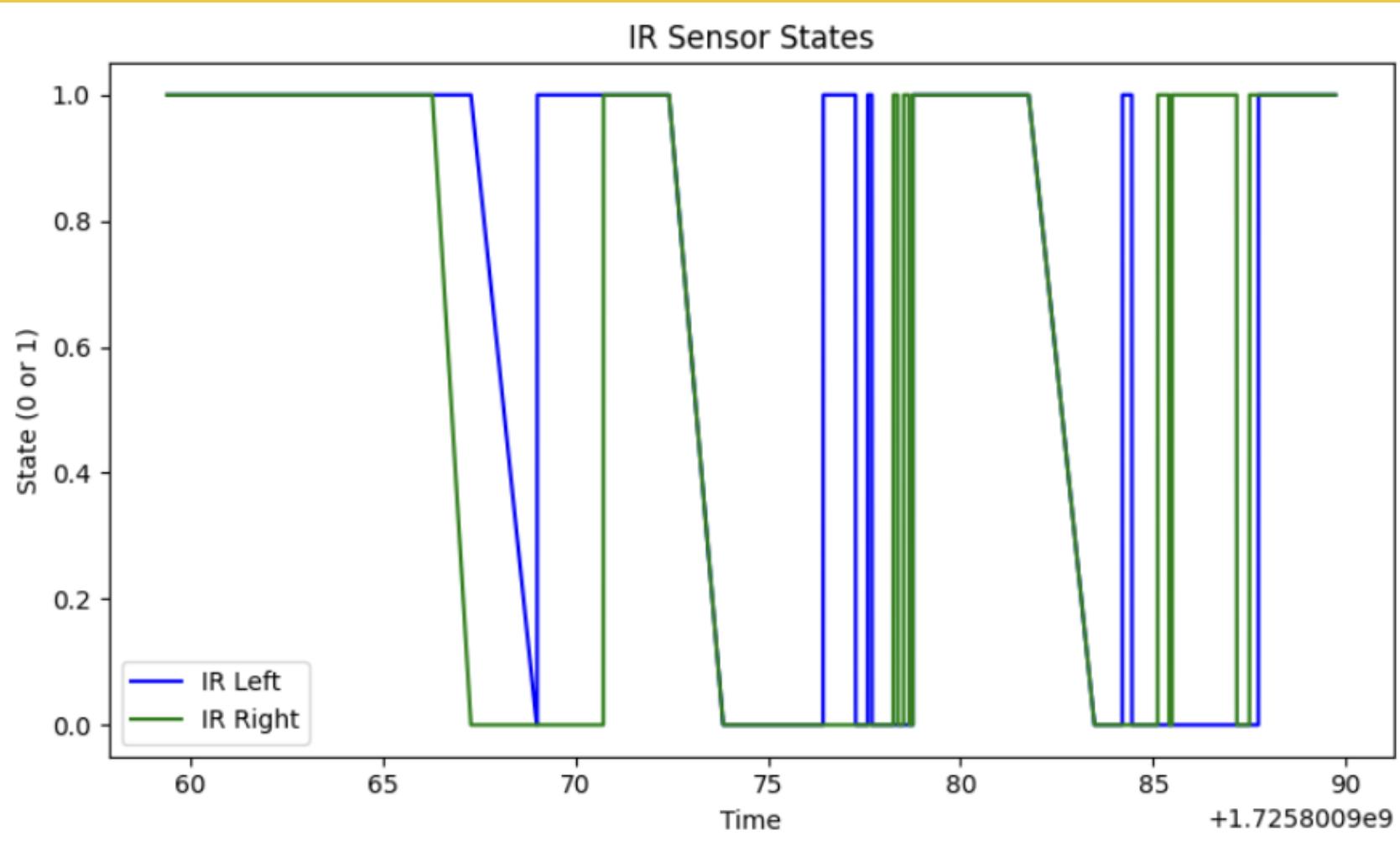


Raspberry Pi	
ULAZI	IZLAZI
IR senzori x2	Continuous Rotation Servo (kretanje mobilnog robota)
IR senzori x1	Servo motor (Upravljanje nosećom platformom)

# Logika rada

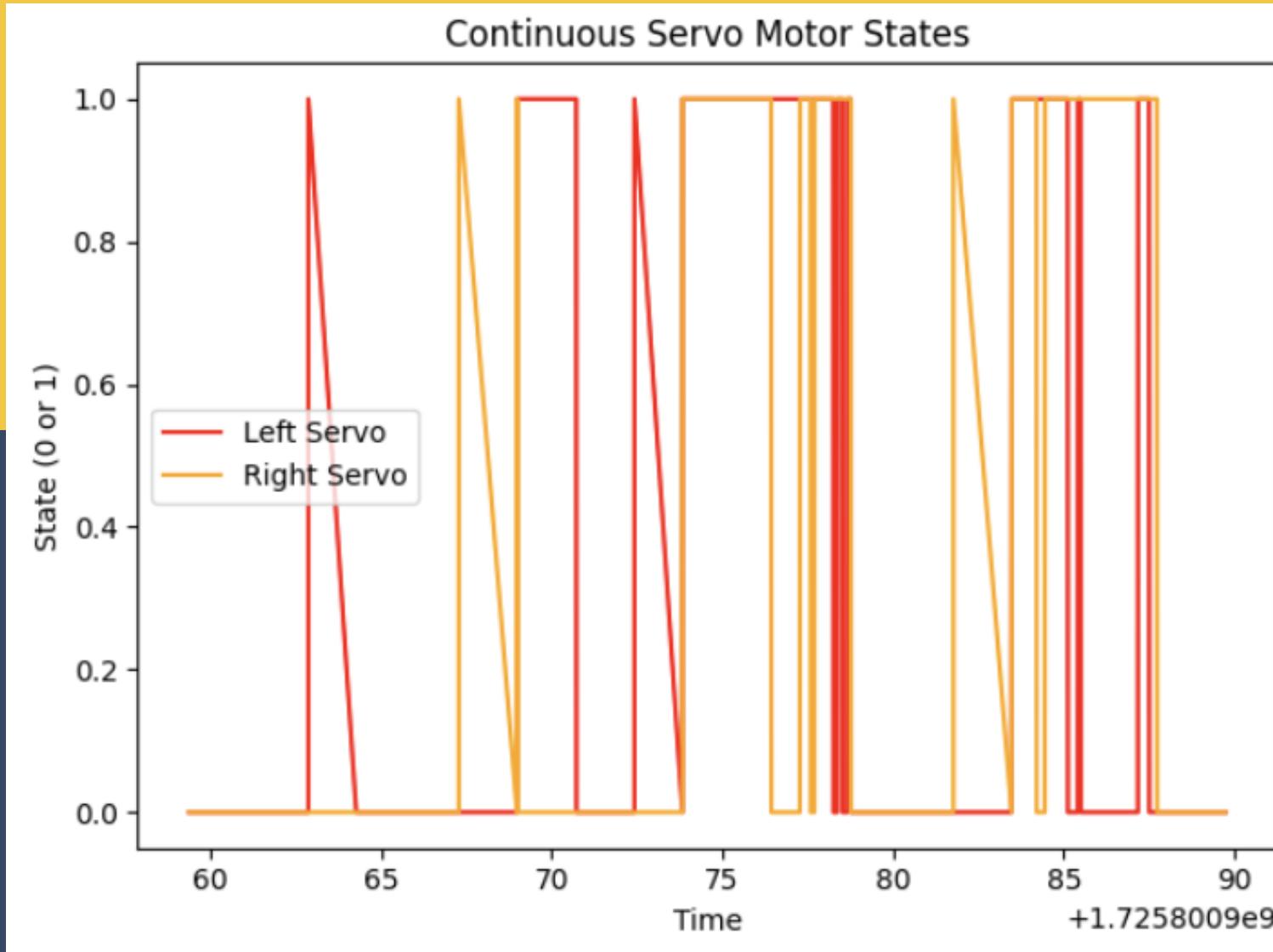
IR lijevi	IR desni	Paradajz detektovan	Akcija	Opis
0	0	0	Kreći se naprijed	Oba senzora detektuju liniju,pomjeri se naprijed
1	0	0	Okreni se desno	Lijevi senzor OFF,desni senzor ON,idi desno
0	1	0	Okreni se lijevo	Lijevi senzor ON,desni senzor OFF,idi lijevo
1	1	0	Stop	Oba senzora nisu na liniji,STOP (kraj linije)
x	x	1	Idi nazad, okrenite se i prati liniju	Paradajz detektovan,idi unazad,okreni se,prati liniju
x	x	x	Čekanje paradajza	Čekanje paradajza da bude detektovan na prikolici

# Grafički prikaz odziva



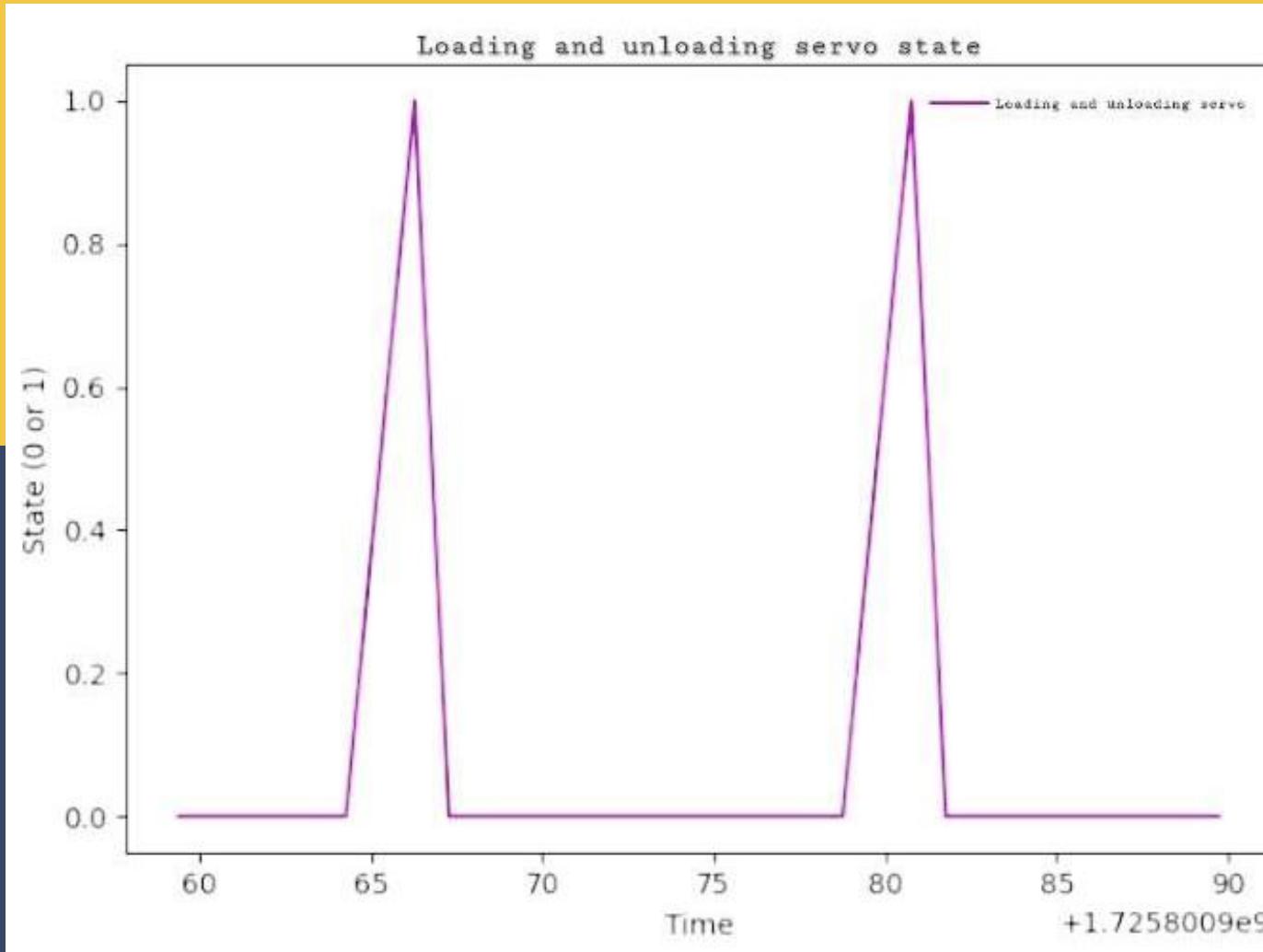
Aktivnost IR  
senzora

# Grafički prikaz odziva



Aktivnost  
servo motora

# Grafički prikaz odziva



Utovar/istovar