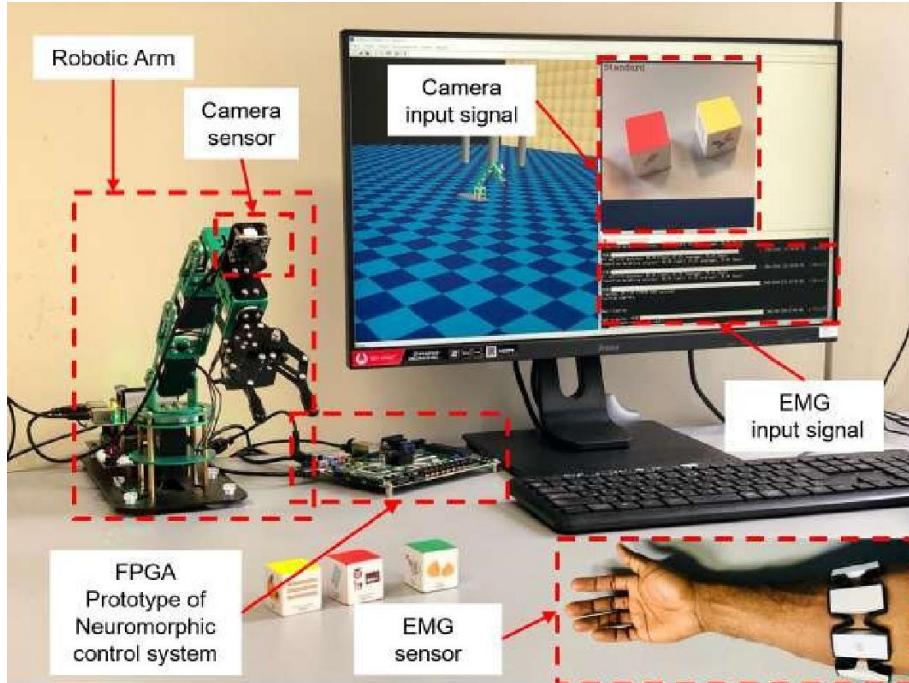
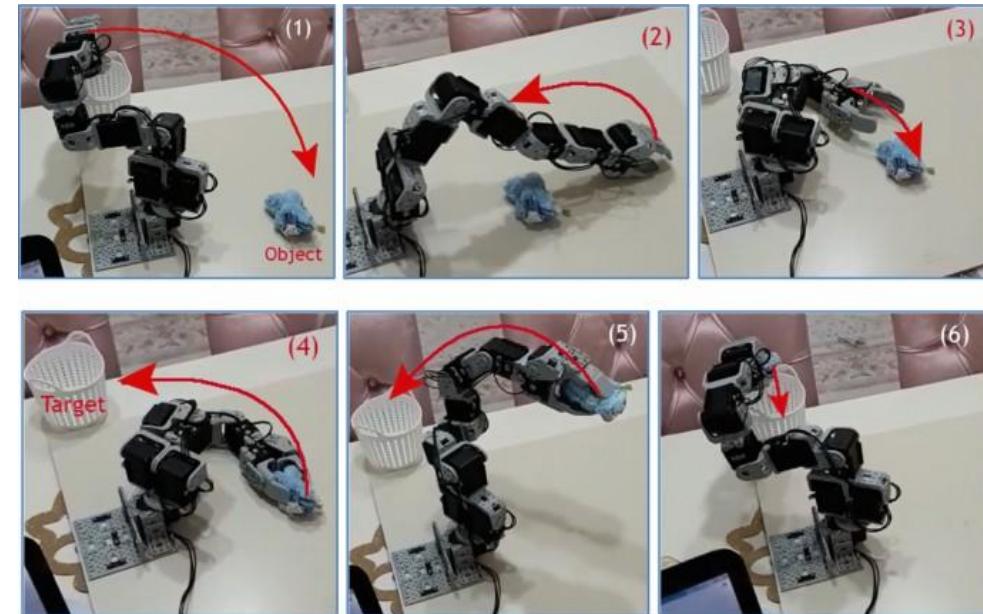


PROJEKTOVANJE SISTEMA NA ČIPU



Eksperimentalni dizajn FPGA baziranih sistema

prof. dr. Lejla Banjanović-Mehmedović



Sadržaj izlaganja

- 
- Osnovne postavke eksperimentalnog dizajna sa FPGA
 - Savremene aplikacije
 - Percepcija, aktuacija i donošenje odluka (softare dizajn) u mobilnoj robotici
 - Primjeri
 - Projektovanje robota – Smart garden
 - Logistika sa mobilnim robotom
 - Dezinfekcija objekata sa robotskom rukom
 - Sortiranje predmeta po boji

Eksperimentalni dizajn FPGA baziranih sistema

- Razvoj savremenih elektronskih sistema sve više zahtijeva
 - **Paralelnu obradu podataka**, što omogućava brze i efikasne algoritme u realnom vremenu;
 - **Programabilnost** koja omogućava visoku fleksibilnost, eksperimentalni dizajn i testiranje novih arhitektura i algoritama;
 - **Integraciju sa senzorima i aktuatorima** za implementaciju cyber-fizičkih sistema;
 - **Smanjenje kašnjenja u obradi podataka** u odnosu na standardne procesore (CPU/GPU).
- FPGA (Field-Programmable Gate Array) predstavljaju ključnu tehnologiju u ovom kontekstu, jer omogućavaju **dizajn digitalnih sklopova koji se mogu programirati nakon proizvodnje**, čime se kombinuje **prednost hardverske akceleracije sa fleksibilnošću softverskog pristupa**.

Eksperimentalni dizajn FPGA baziranih sistema

- FPGA bazirani sistemi se koriste u širokom spektru aplikacija:
 - od industrijskih kontrolnih sistema,
 - robotike,
 - telekomunikacija,
 - do autonomnih i cyber-fizičkih sistema.

Primjena FPGA

- FPGA bazirani sistemi nalaze široku primjenu u oblastima gdje je potrebna visoka brzina obrade podataka i fleksibilnost u dizajnu.
- **Industrijska automatizacija i kontrola procesa.** FPGA se koriste za kontrolu motora, PLC funkcionalnosti i senzorskih mreža u realnom vremenu. **Eksperimentalni setup može uključivati FPGA modul povezan sa aktuatorima i senzorima**, gdje se testira brzina odziva i tačnost kontrolnih algoritama.
- **Robotika i autonomni sistemi.** FPGA omogućava paralelnu obradu senzorskih podataka (LIDAR, kamere, IMU) i izvršenje kontrolnih algoritama u realnom vremenu. Eksperimentalni setup može sadržavati **mobilnog robota ili manipulator**, gdje FPGA akcelerira obradu senzorskih ulaza i generiše kontrolne signale za motore.
- **Edge AI i obrada signala.** FPGA moduli mogu **akcelerirati duboke neuronske mreže ili DSP algoritme na rubu sistema**, smanjujući kašnjenje i opterećenje centralnog procesora. Eksperimentalni setup uključuje FPGA integriran sa senzorom i mikrokontrolerom ili računarom, gdje se treniraju i testiraju AI modeli u realnom vremenu.

Proces eksperimentalnog testiranja

- Eksperimentalni setup obično se realizuje kroz sljedeće korake:
 - Specifikacija funkcionalnih zahtjeva sistema
 - Izbor FPGA arhitekture
 - Konfiguracija FPGA uređaja sa odgovarajućim HDL dizajnom ili HLS modulima;
 - Povezivanje perifernih komponenti, uključujući senzore, aktuatoriske module i komunikacijske interfejse;
 - Simulacija funkcionalnosti u softverskom okruženju (ModelSim, Vivado Simulator) prije implementacije na fizičkom prototipu;
 - Integraciju sa perifernim modulima, senzorima i računarskim platformama za praćenje i kontrolu.
 - Mjerenje performansi, uključujući kašnjenje, propusnost podataka i tačnost izlaza;
 - Iterativna optimizacija dizajna na osnovu rezultata testiranja.

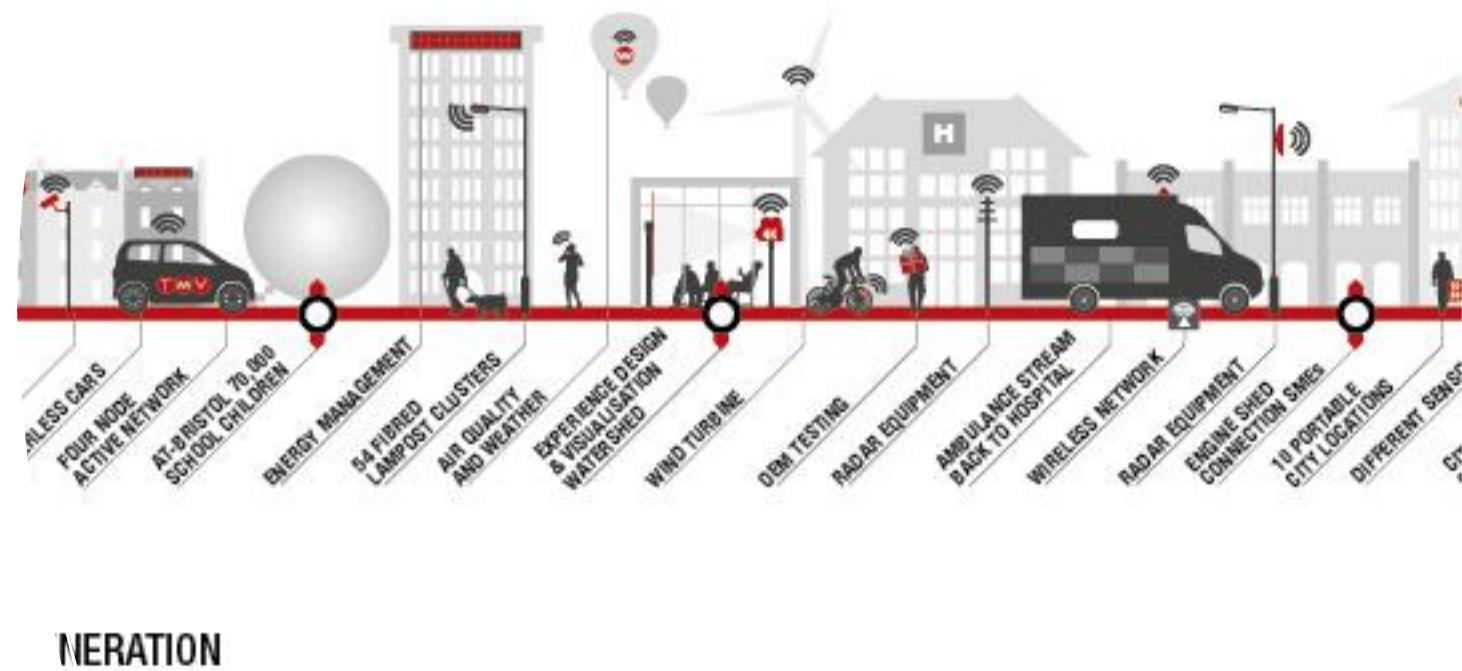
Savremene aplikacije - FPGA platforme i pametni gradovi





Pametni gradovi

- ▶ Podaci od građana
- ▶ Podaci iz okoline
- ▶ Podaci od vozila
- Kako prikupiti podatke?
 - Senzori
 - Kamere
 - Mobilni telefoni



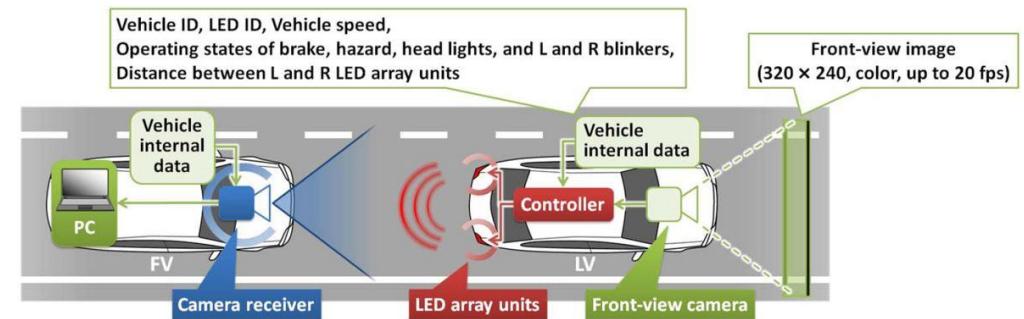
Transportne inovacije - Povezane tehnologije vozila

Tehnologije povezivanja vozila podrazumijeva:

- međusobne konekcije i razmjene podataka do
- potpune autonomnosti.

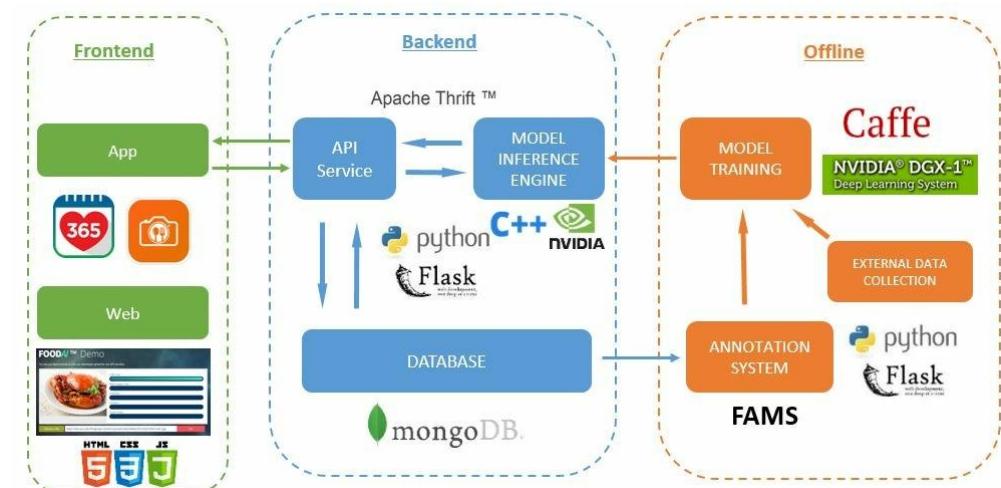
Vozila nove generacije imaju za cilj:

- poboljšane sigurnosti na cestama,
- bolje upravljanje saobraćajem,
- smanjene emisije i putnih troškova, više udobnosti.



Industry 4.0

- Primjer: Klasifikaciju voća i povrća na osnovu kvalitete, kao i za identifikaciju oštećenja na njima.
- Organizacija proizvodnog procesa kroz komunikaciju između mašina i sirovina



Trend u servisnoj robotici

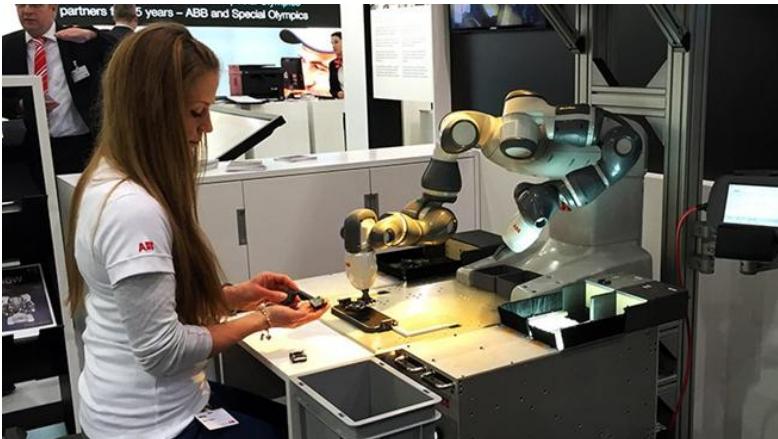
Projektovanje sistema na čipu 4



Copyright: Lejla Banjanovic-Mehmedovic

Kobot dizajn sa više manipulatora

- Dizajn kolaborativnih roboata kao kombinacija više manipulatora s različitim zadacima.



Industrijske aplikacije - asembliranje elektronskih komponenti

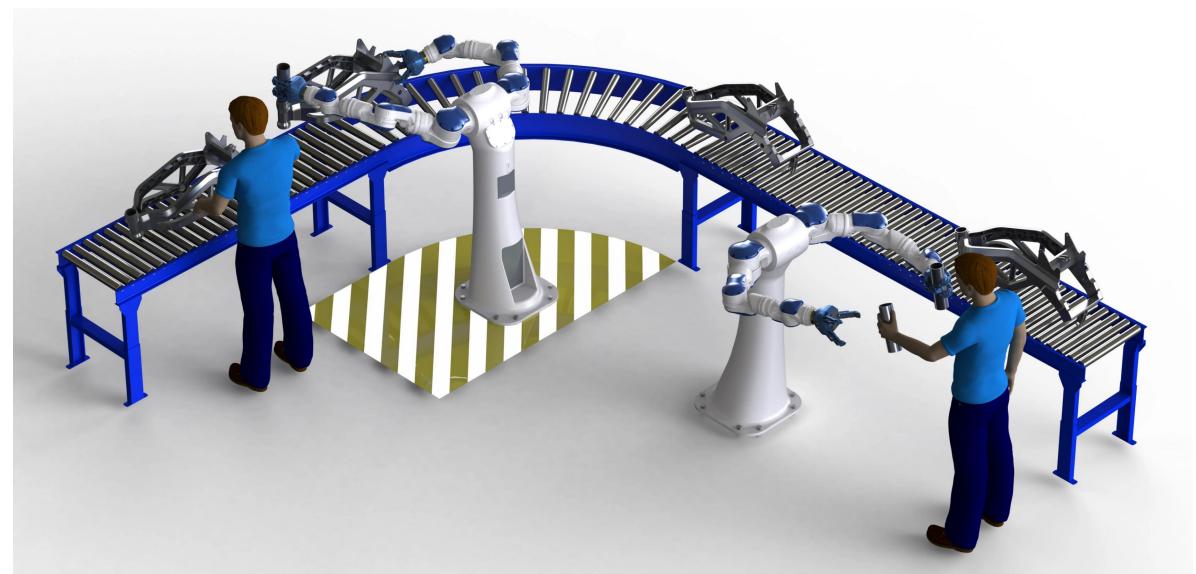
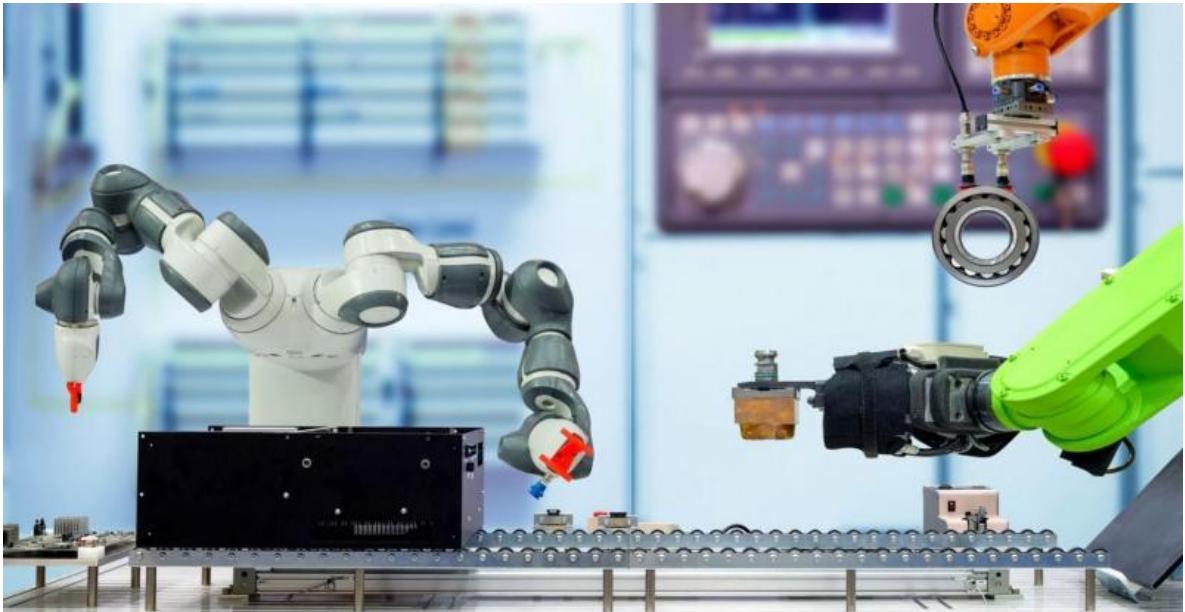
Smart House - Collaborative Robot in Kitchen (WordlessTech 2015)



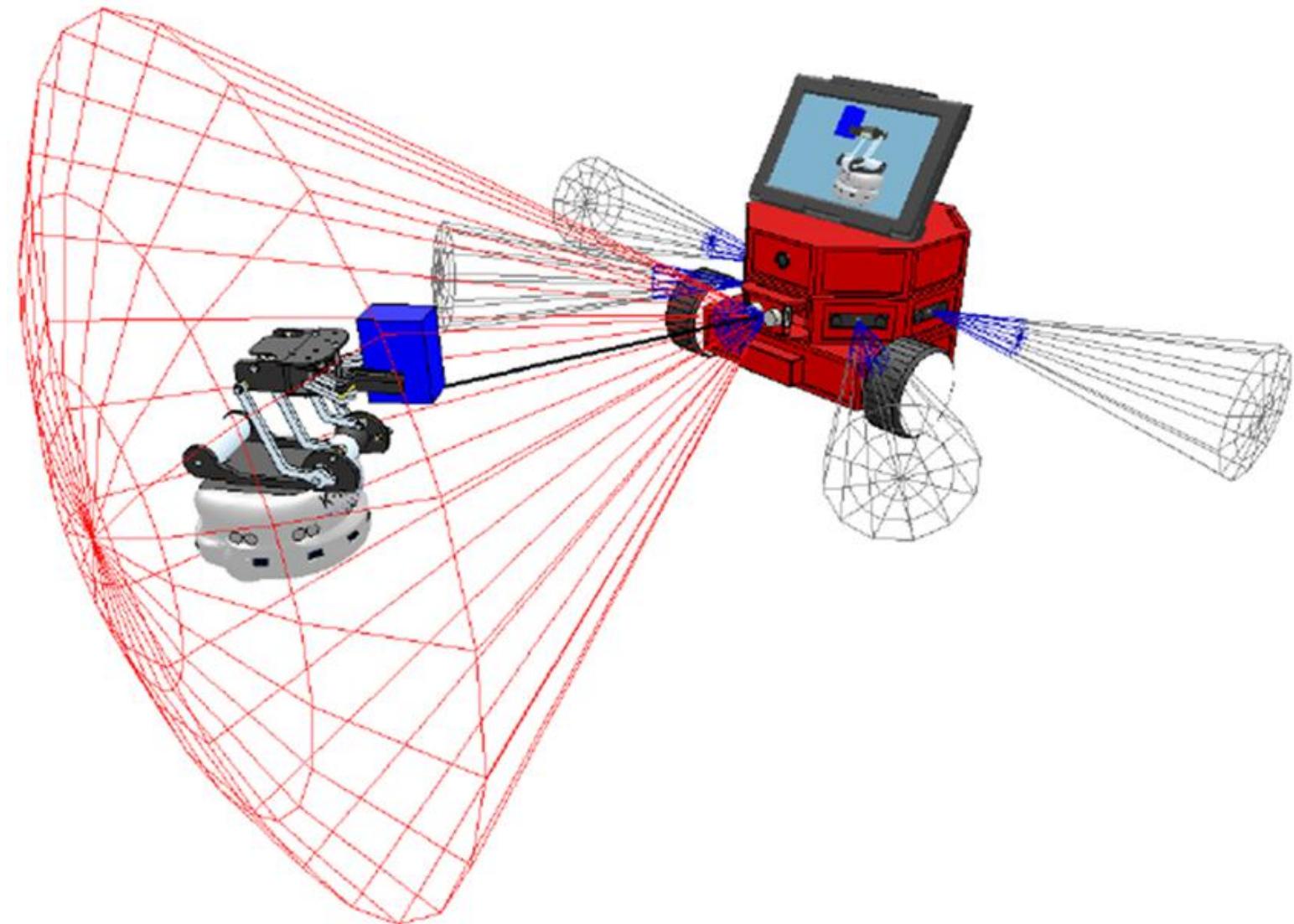
ABB kolaborativni robot u medicinskoj laboratoriji u Texas Medical Center (TMC) inovacioni kampus Houston, Texas (Robotics and automation 2019)



Coboti u akciji



Percepcija, aktuacija i donošenje odluka (softare dizajn) u mobilnoj robotici



Komponente robota

- Tjelo
- Senzori
- Aktuatori
- Efektori
- Kontroler (Arduino, Rpi, FPGA Cyclone, Nvidia,...)
- Software



Prikupljanje senzorskih podataka

- Robotske sposobnosti zavise o:
 - Kvantiteta i kvaliteta senzora
 - Sposobnosti i brzine procesiranja senzorskih ulaza
 - Sposobnosti da se djeluje na ono što se prihvata
- Sposobnost robota da postane svjestan svog okruženja kroz senzore - *percepcija*.



Primjeri FPGA kontrolera u robotici

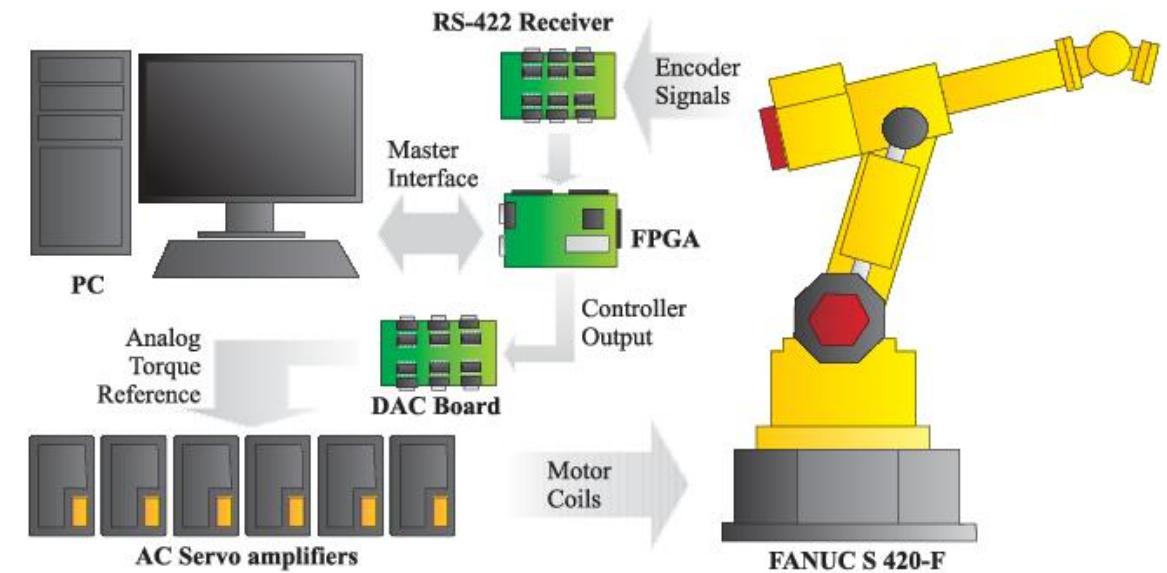
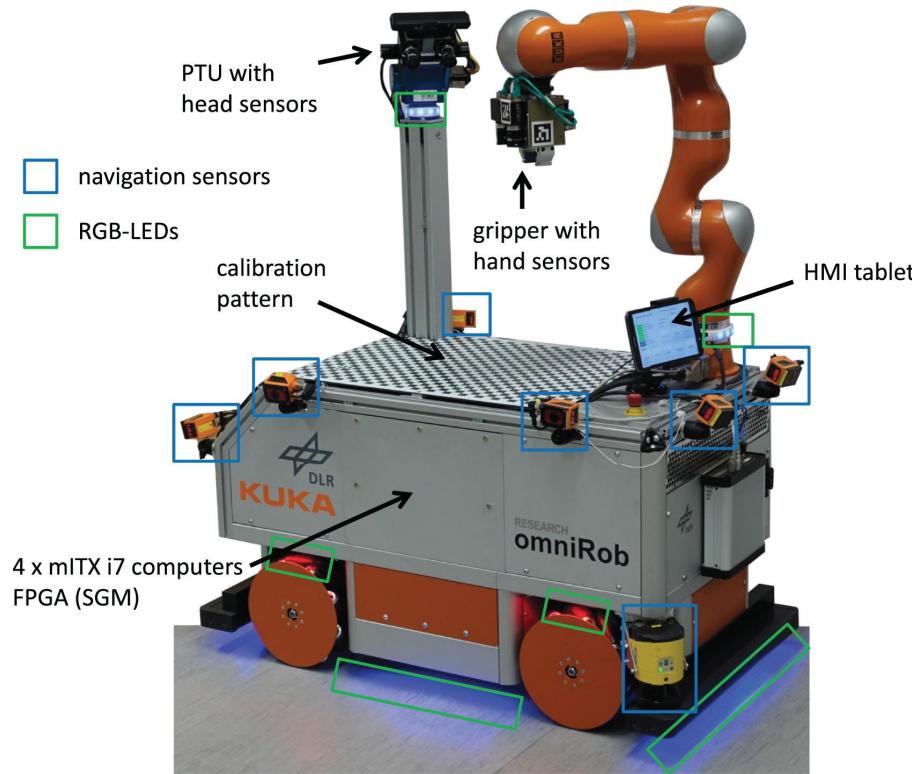
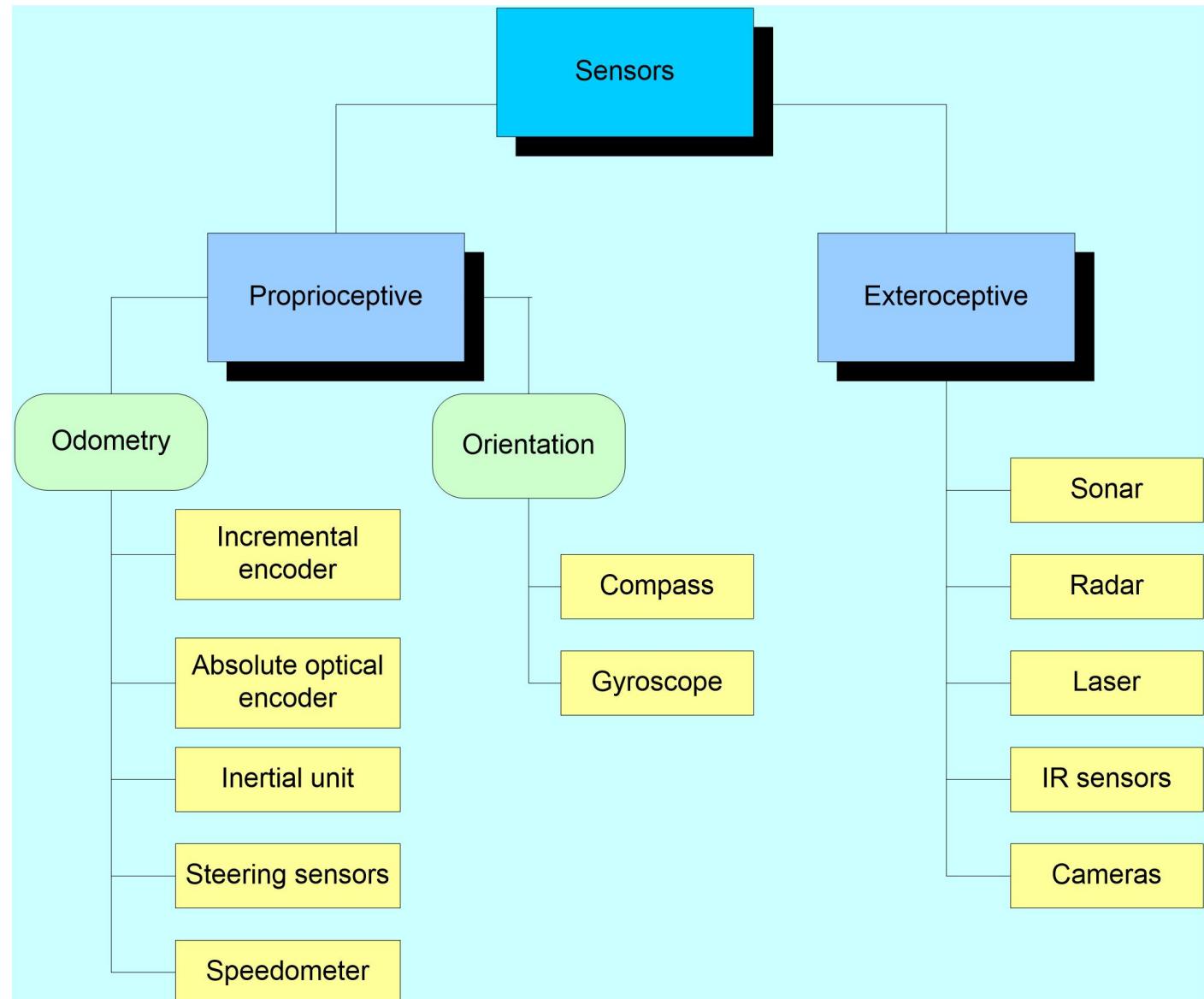


FIGURE 1. Robot controller hardware setup.

Senzori u mobilnoj robotici

- **Propriocepcijski senzori** (mjere unutrašnje kretanje mehanizma MR):
 - **Odometrijski** (pozicija, brzina)
 - **Orijentacioni** (kompasi, žiroskopi)
- **Eksterceptivni senzori** (mjere pojave vezane za okolni prostor MR):
 - Ultrazvučni senzori udaljenosti, laserski senzori udaljenosti, IR senzori, kamere



Eksteroeceptivni senzori



Kamere se obično koriste za modeliranje komplikovanih scena



Ultrasonični senzori se često koriste za dinamičko izbjegavanje prepreka na većim robotima



IR senzori se tipično koriste za izbjegavanje prepreka, slijedenje zida, mapiranje i sl. ... na manjim robotima

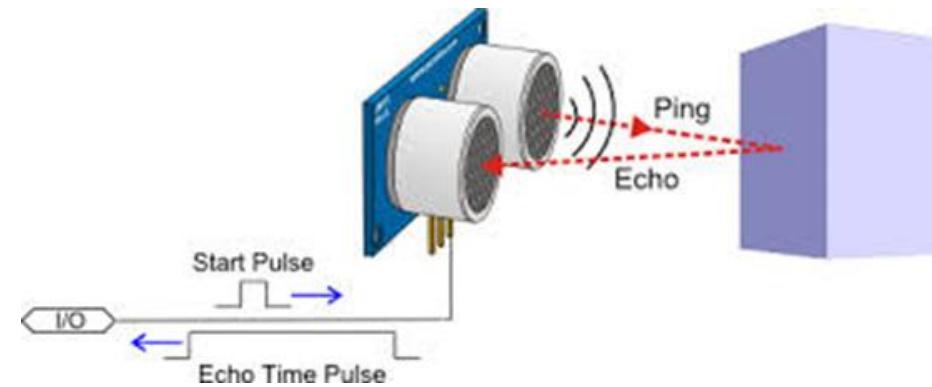
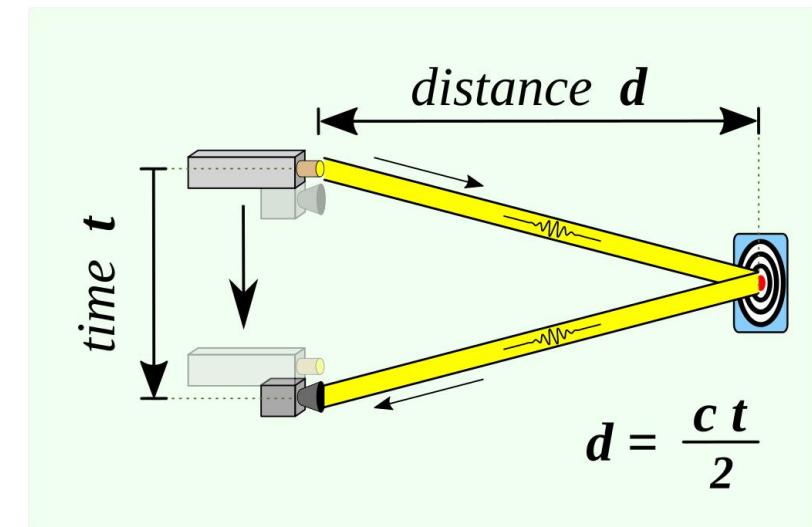
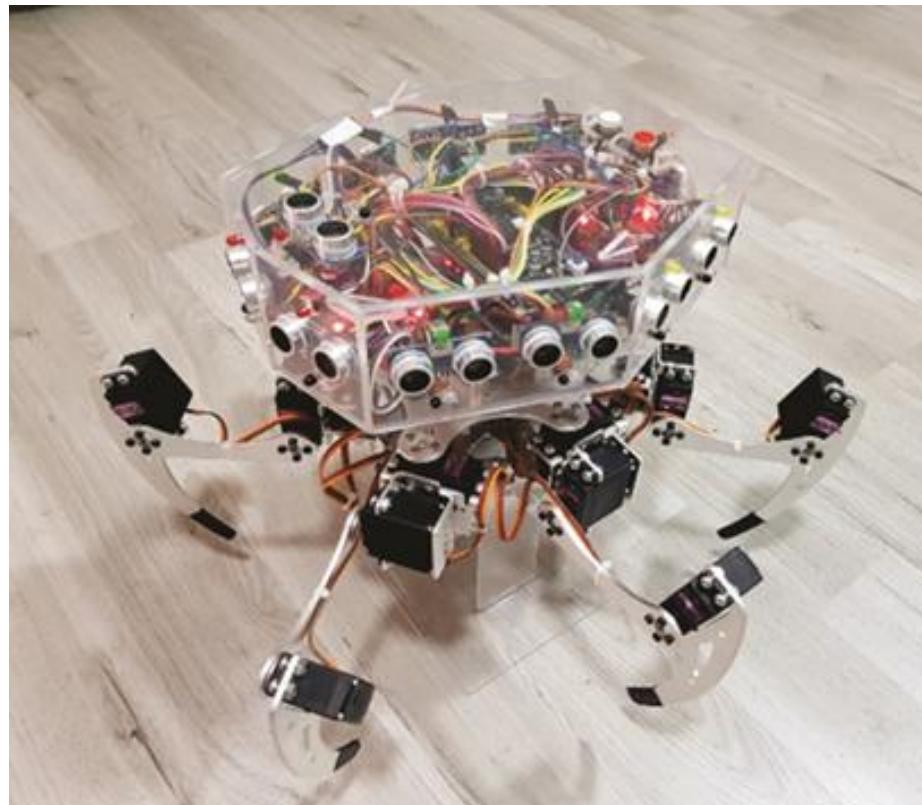


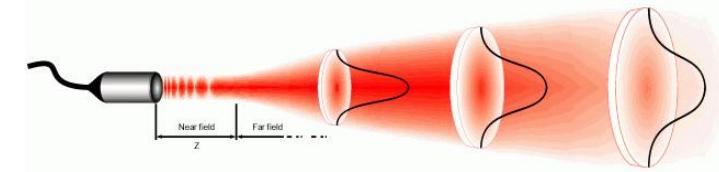
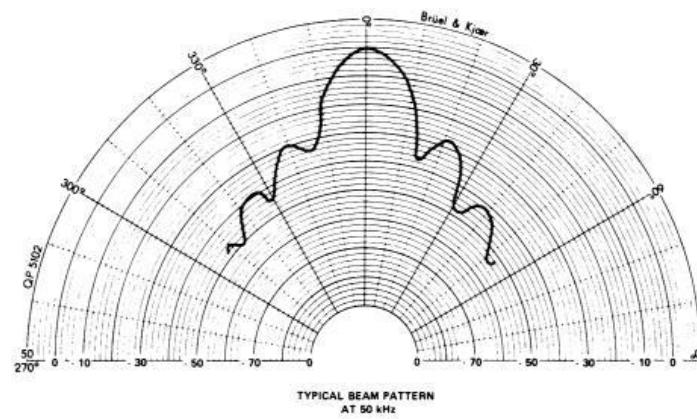
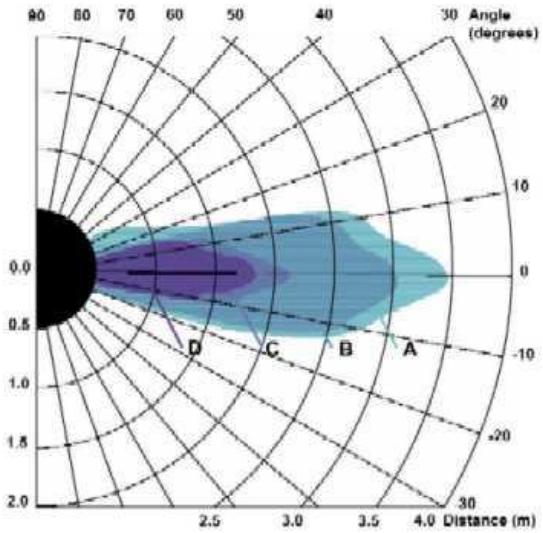
Senzori područja (distance)

- **Senzori područja (Range Sensors)**
 - Taktilni (Tactile Sensors)
 - Senzori distance ili blizine (Proximity Sensors)
 - Ultrasonični (**zvuk**)
 - Infracrveni senzori (**svjetlo**)
 - Laserski (**svjetlo**)

Senzori distance

Ultrasonični (kroz zrak)





Ultrasonični senzori

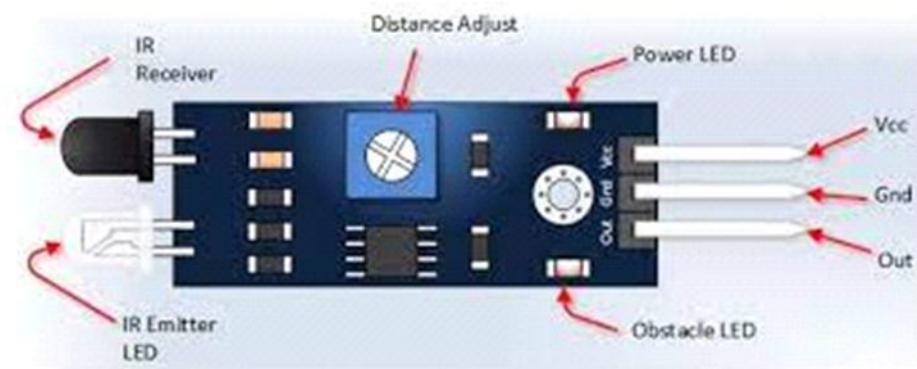
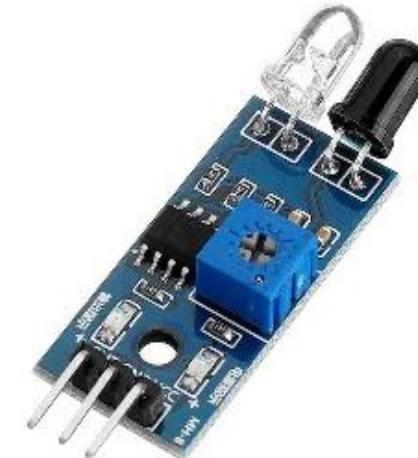
- Oblik zrake nije jednostavne forme
- Širi objekti bliže centru zrake rezultiraju u boljoj tačnosti

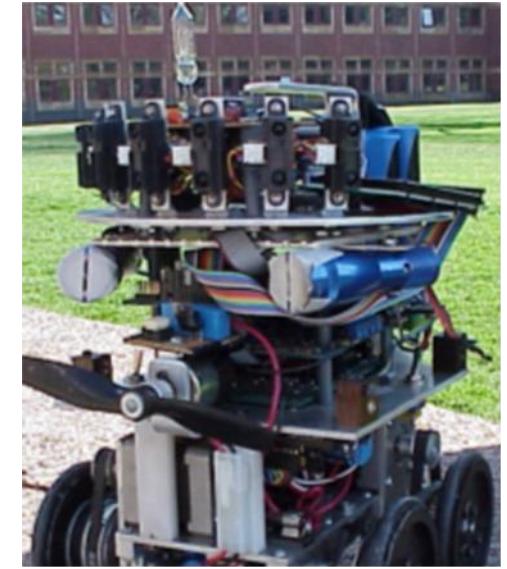
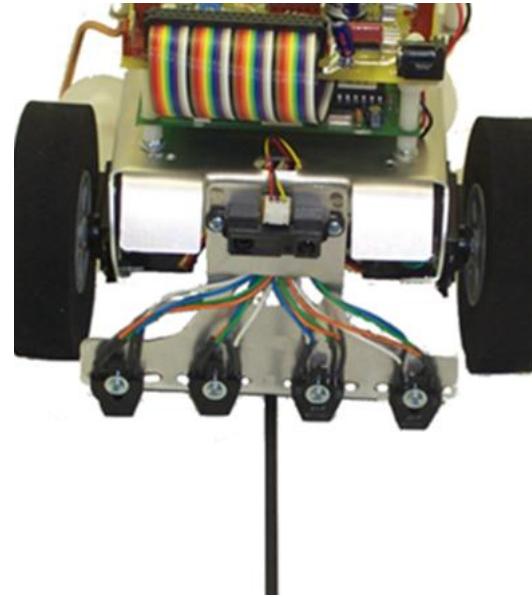
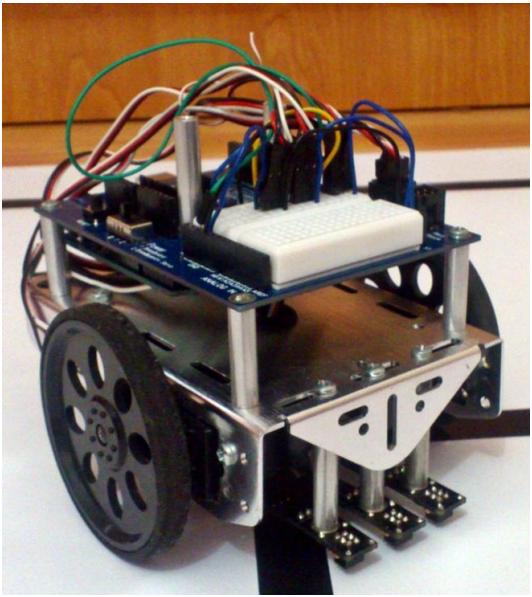
IR senzor

- Infracrveni (IR) senzor - mjeri i detektuje infracrveno zračenje u svom okruženju.
- IR - nevidljiv ljudskom oku, jer je njegova talasna dužina veća od talasne dužine vidljive svjetlosti.

Bazni IR senzori rade na jednostavnom principu:

- Uključenje IR diode (tj., svjetlosti)
- Svjetlost se reflektuje od prepreke, neki dio
- Svjetlosti se vraća
- Detektor mjeri snagu svjetlosti koja se vratila.



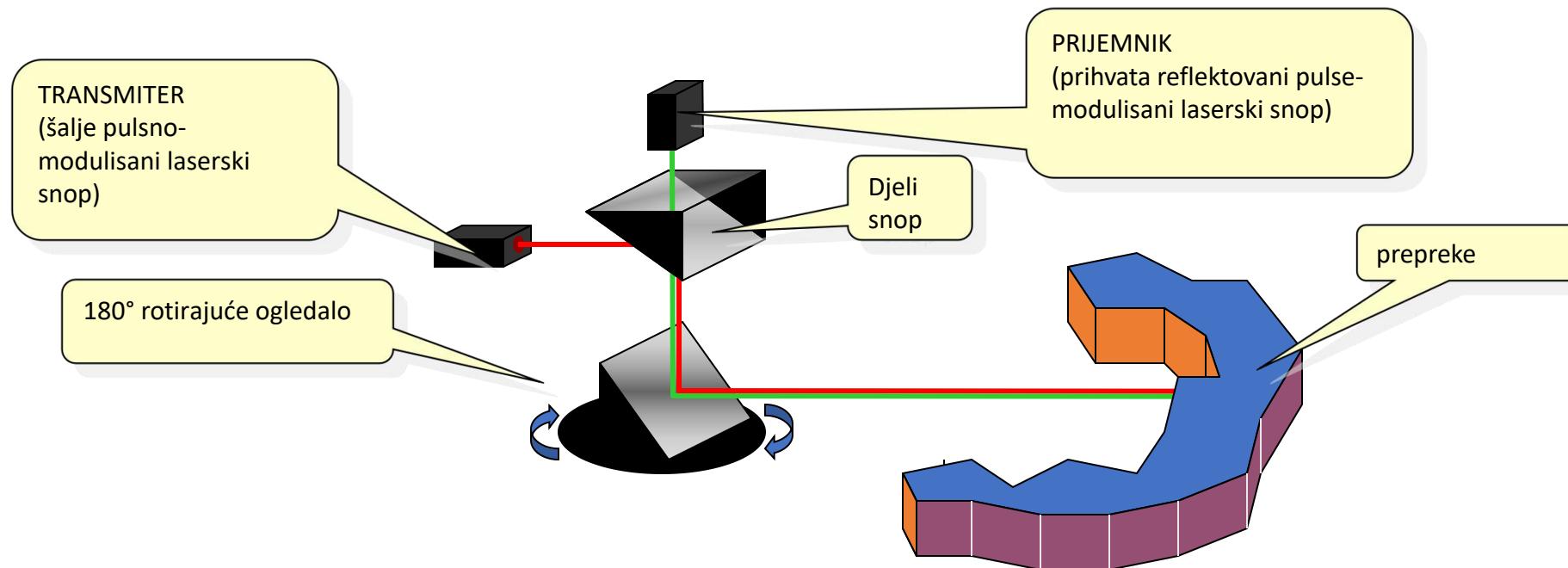


Dizajn robota za praćenje linije

- Kao i sonari, signal je jako ovisan o refleksivnim karakteristikama objekta (metali uzrokuju probleme)
- Opseg IR ovisi o boji prepreke
- Bijele /crne površine reflektuju razl. opsege
- Ne mogu detektovati staklo

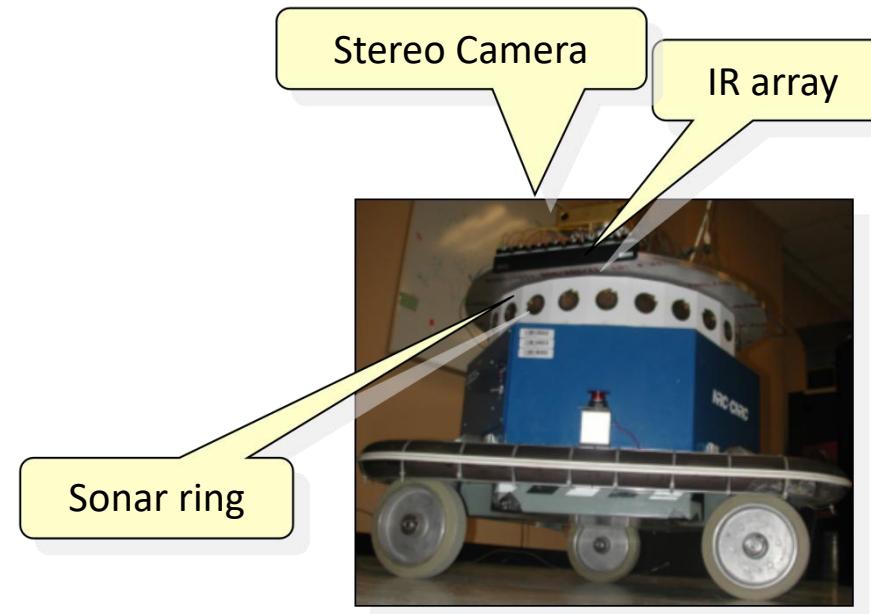
Laserski senzori (Lidar)

- **Svjetlost** putuje puno većim brzinama od zvuka



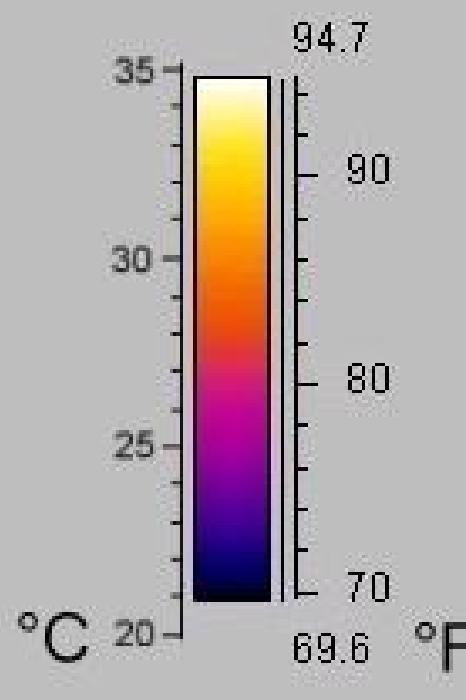
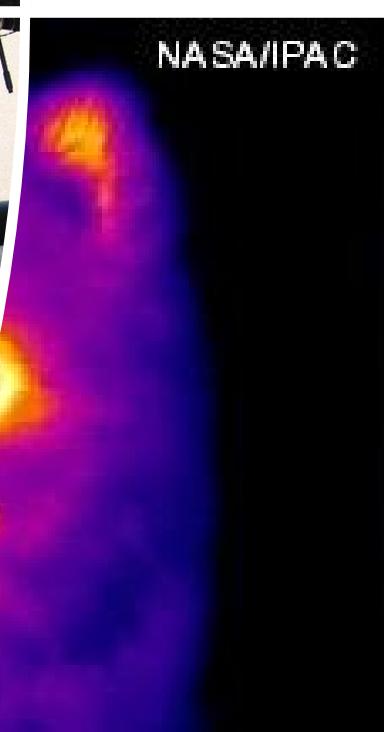
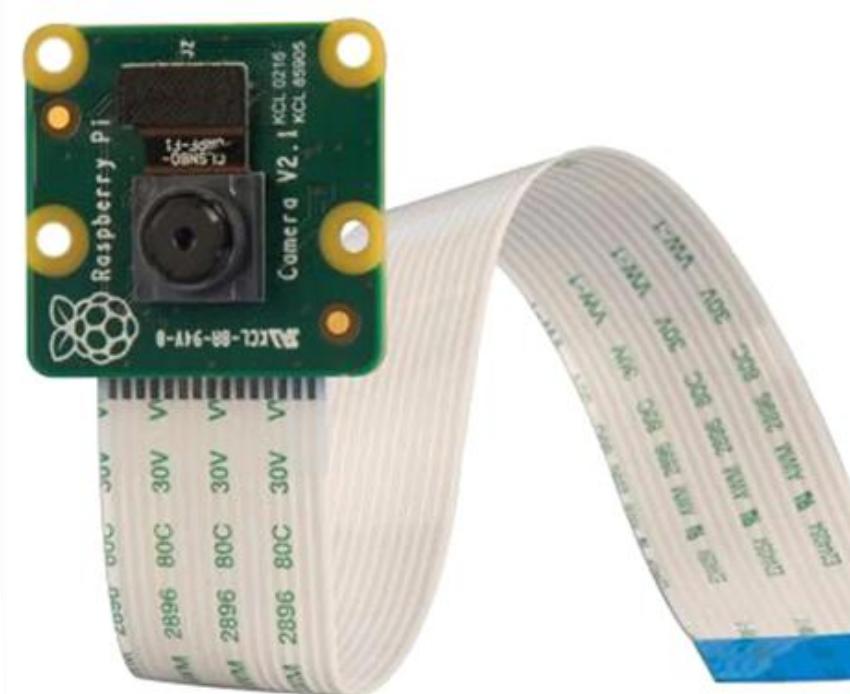
Senzorska fuzija

- Za izbjegavanje netačnosti, kombinovanje (fuzija) senzorskih očitanja
- ***Senzorska fuzija*** kombinuje:
 - Iste senzore
 - Mnoštvo sličnih istog tipa
 - Individualna senzorska očitanja iz različitih pravaca (npr. Sonarni "prsten")
 - Različite senzore
 - Npr., sonar sa IR i vizijom

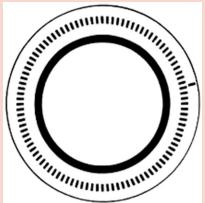


Senzori vizije

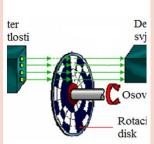
- Monokularne kamere
- Stereo kamere
- Omnidirekcionе kamere (vozila)
- Infracrvene kamere
- TOF kamere



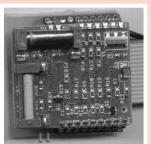
Proprioceptivni senzori



Inkrementalni optički enkoderi mjere rotacijsku brzinu iz koje se može izračunati relativna vrijednost pozicije.

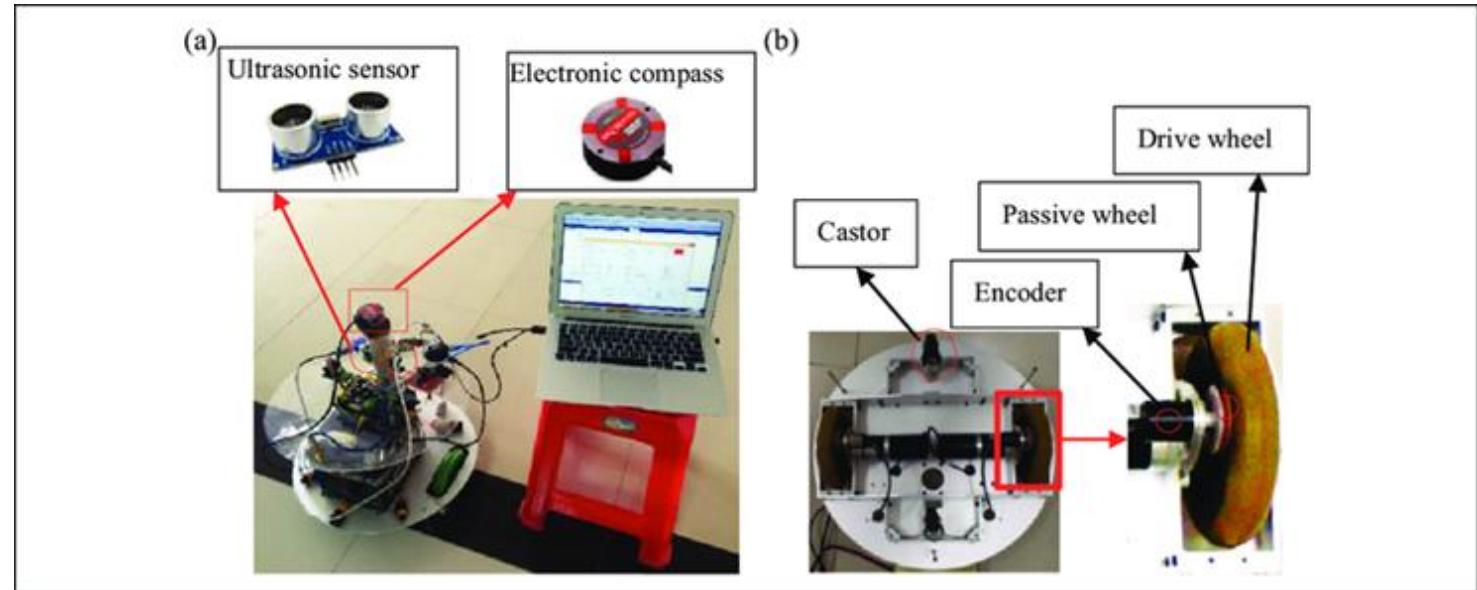


Apsolutni optički enkoderi direktno mјere ugaonu poziciju, na osnovu koje se može izračunati brzina.

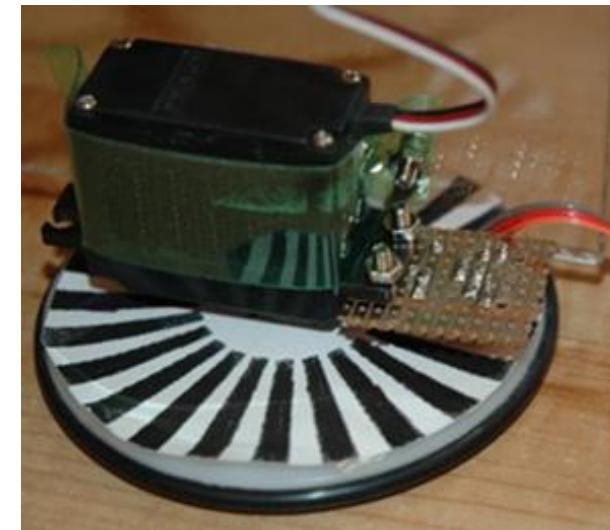


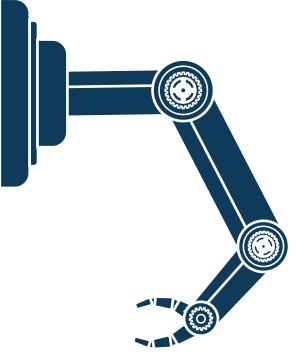
Digitalni kompas - omogućava mnogo više smjerova razlučivosti.

Odometrijski senzori



- Optički enkoderi - dobivanje informacija neophodnih za upravljanje brzinom i pozicijom mobilnih robota.



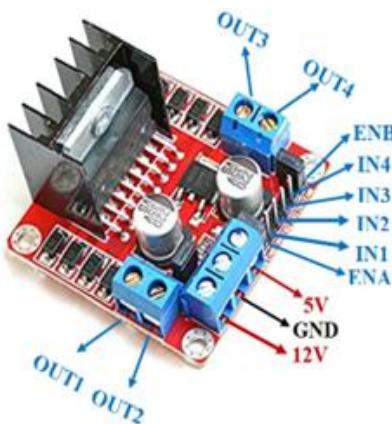


Izlazi -aktuatori



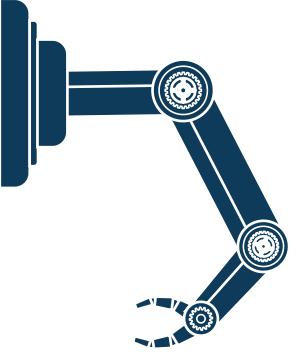
DC motor sa točkom

- Istosmjerni motor - **elektromehanički uređaj** koji električnu energiju pretvara u mehaničku odnosno **pretvara istosmjernu struju u kružno kretanje**.
- Glavni dijelovi motora su rotor i stator kao i kod svakog drugog motora, sadrži i komutator s grafitnim četkicama.

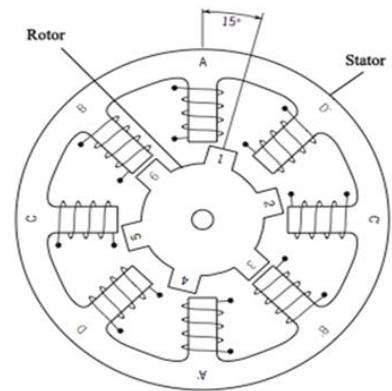
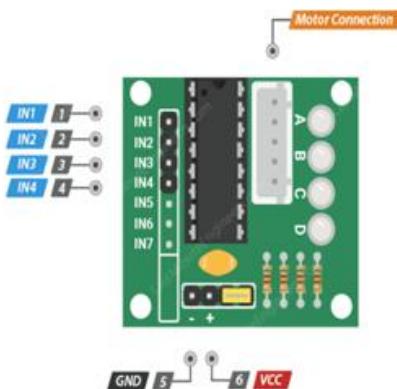
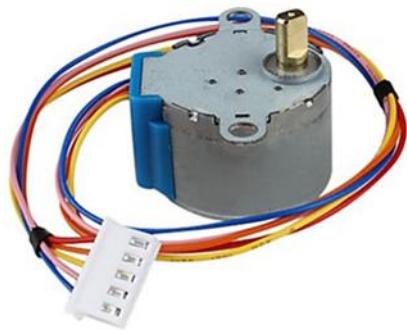


L298N motor driver

- L298N je dvostruki H-most driver motora koji omogućava kontrolu brzine i smjera dva DC motora u isto vrijeme.
- Modul može pokretati DC motore koji imaju napon između 5 i 35V, sa radnom strujom do 2A.



Izlazi -aktuatori

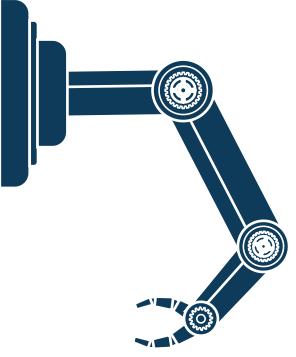


Stepper motor

- 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. Sadrže pojedinačne setove zavojnica.
- **Pretvaraju strujne impulse u fiksne inkrementne ugaonog pomjeraja - korake.**

ULN2003 stepper motor driver

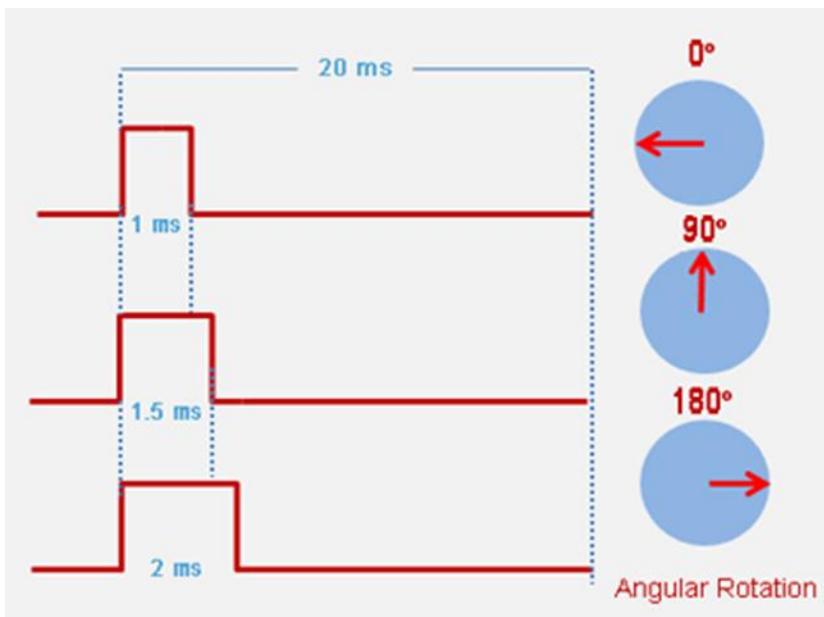
- Stepper motor obično dolazi sa ULN2003 drijverskom pločom.
- ULN2003 isporučuje veću struju od jednog tranzistora i omogućava niskonaponskom niskostrujnom izlazu mikrokontrolera da pokreće stepper motore velike struje.

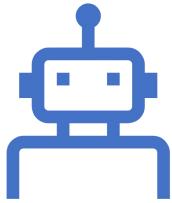


Izlazi - aktuatori

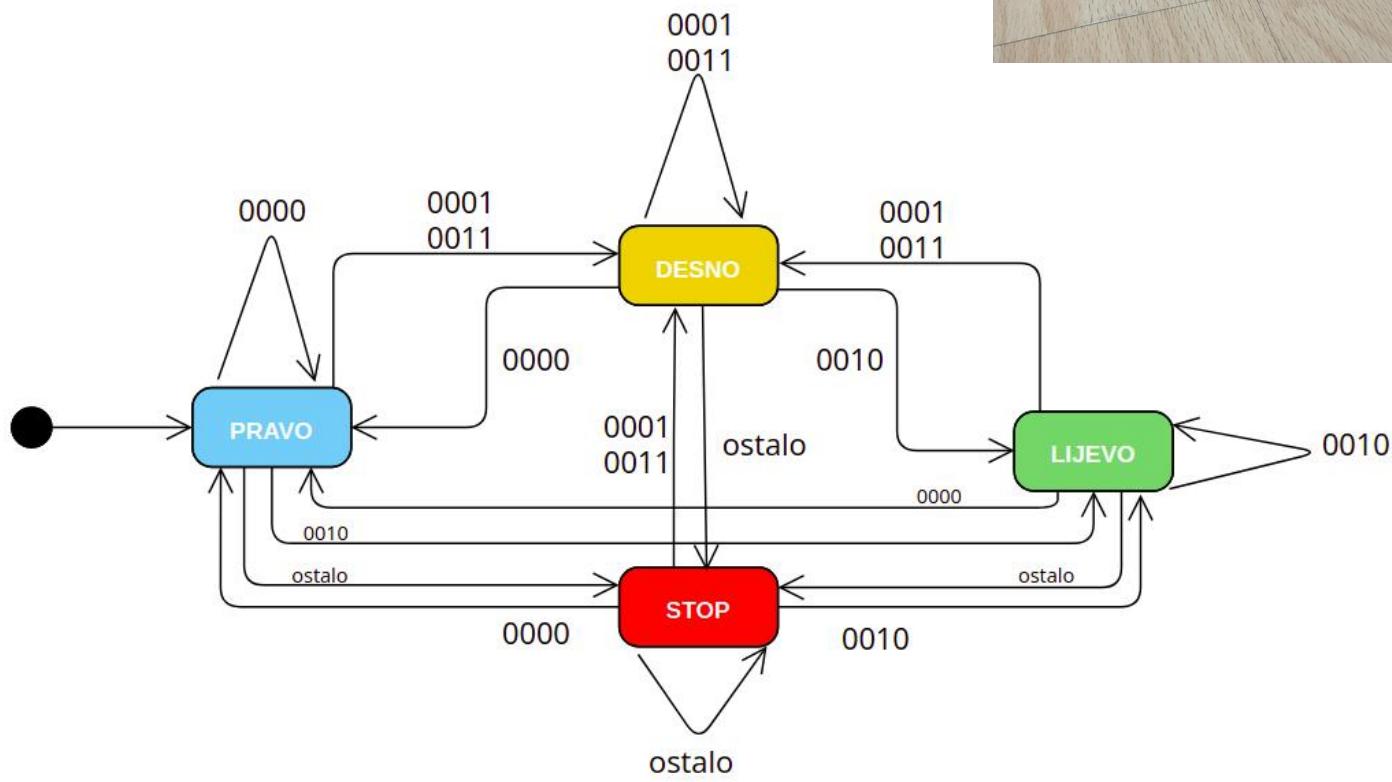
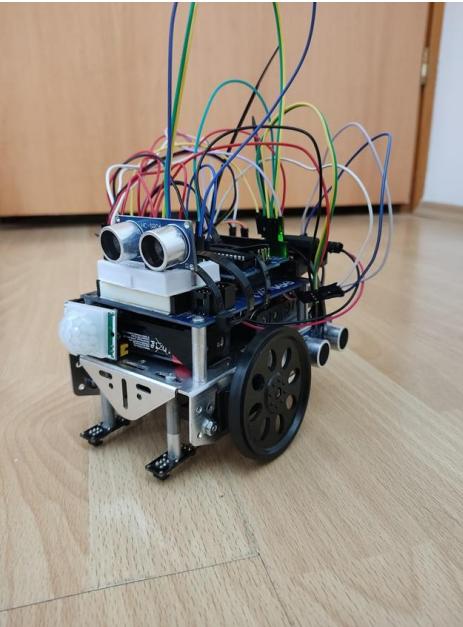
Servo motor

- Ova vrsta motora sastoji od upravljačkog kruga koji **primljeni upravljački signal manje snage pretvara u pomake**, tj. daje informaciju o trenutnom položaju osovine motora
- Servo motor radi na principu PWM (Pulse width modulation).
- **Trajanje impulsa određuje poziciju osovine.**





Software dizajn



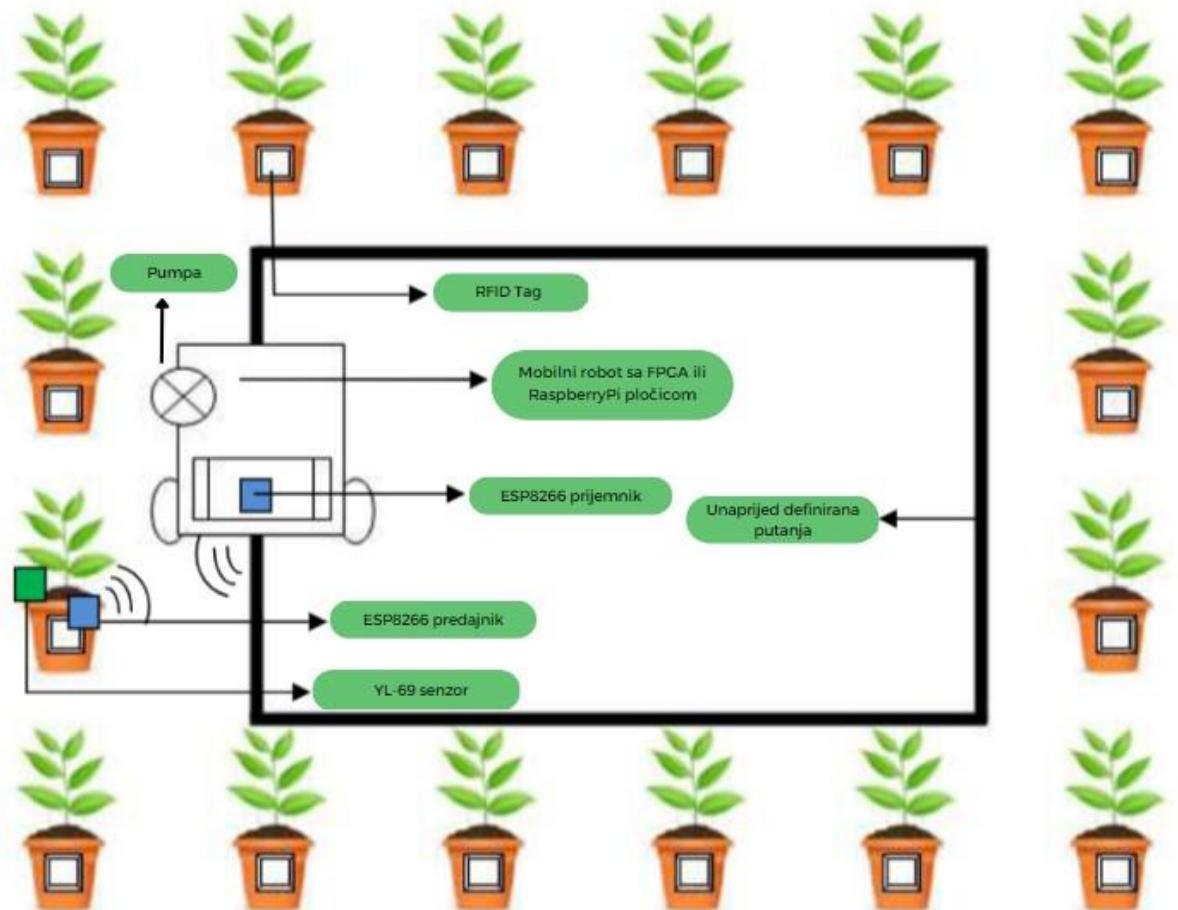
TRENUTNO STANJE	P1 P2 US1 US2	NAREDNO STANJE
PRAVO	0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 ostalo	PRAVO DESNO LIJEVO DESNO STOP
LIJEVO	0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 ostalo	PRAVO DESNO LIJEVO DESNO STOP
DESNO	0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 ostalo	PRAVO DESNO LIJEVO DESNO STOP
STOP	0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 ostalo	PRAVO DESNO LIJEVO DESNO STOP



Primjer 1: Smart garden - mobile robot solution



LOGIKA RADA

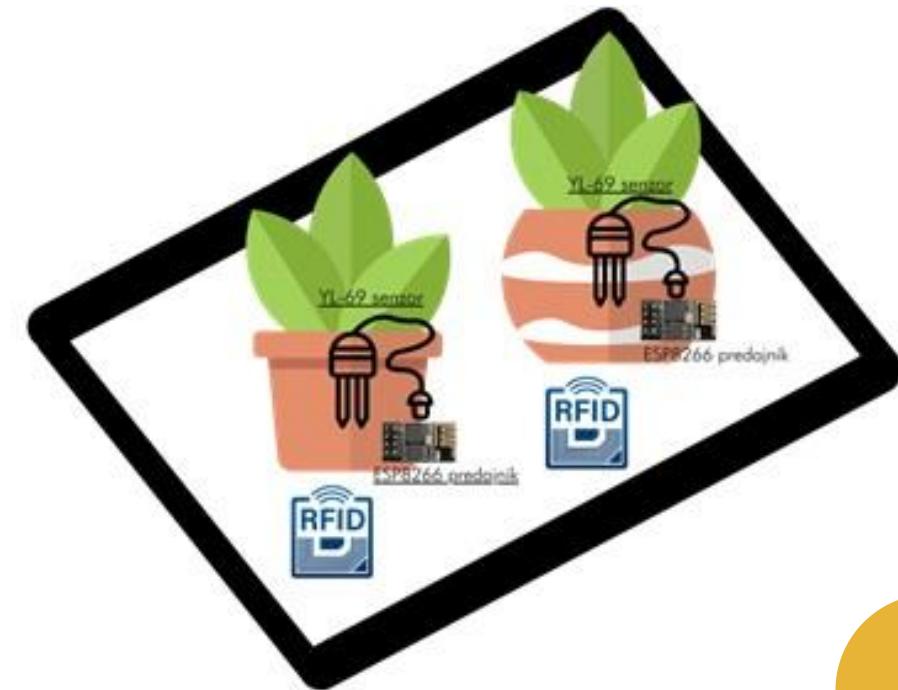
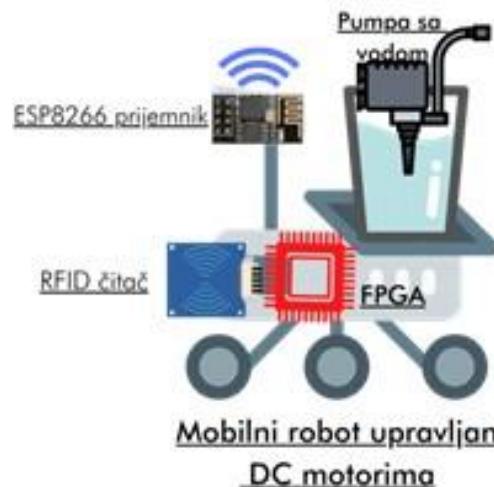


- 01** Za realizaciju projekta planirana je upotreba FPGA ili Raspberry Pi pločice
- 02** Staza koju robot treba da prati mora biti crna na bijeloj pozadini zbog IR senzora koji apsorbuju svjetlost i razlikuju boje na osnovu količine svjetlosti koju apsorbuju.
- 03** U polju sa biljkama se nalaze senzori vlažnosti tla (YL69), koji signaliziraju nedostatak vode u zemlji.
- 04** Ukoliko je potrebno zaliti tlo, iz bašte se, pomoću Wi-Fi mehanizma šalje signal mobilnom robotu koji će značiti da je vrijeme za zalijanje. Koristit će se ESP2866 nodeMCU mikrokontroler za slanje takvog signala robotu i robot započinje kretanje.
- 05** Ispred svake biljke se nalazi RFID oznaka. Na robotu se nalazi RFID čitač i on očitavajući RFID oznake omogućava skretanje robota i finalnu lokaciju.
- 06** Kada robot dobije informaciju da je potrebno zaliti tlo, on kreće iz svoje pozicije prema mjestu gdje je neophodno zalijanje. Robot slijedi unaprijed određenu putanju koja će ga dovesti u blizinu biljake koju treba zaliti.
- 07** Za navodnjavanje se koristi kontejner napunjen vodom u kojem se nalazi pumpa.



CILJ PROJEKTA

- 01** Predstaviti mogućnosti upotrebe FPGA i RaspberryPi pločice realizacijom potpuno autonomnog sistema koji pomaže u zalijevanju saksijskih biljaka koje su raspoređene duž unaprijed definisane putanje.
- 02** Performanse sistema su procijenjene između broja biljaka koje se zalijevaju i vremena potrebnog mobilnom robotu da izvrši operaciju navodnjavanja, što pokazuje da sistem nije samo isplativ, već i efikasan u smislu vremena.



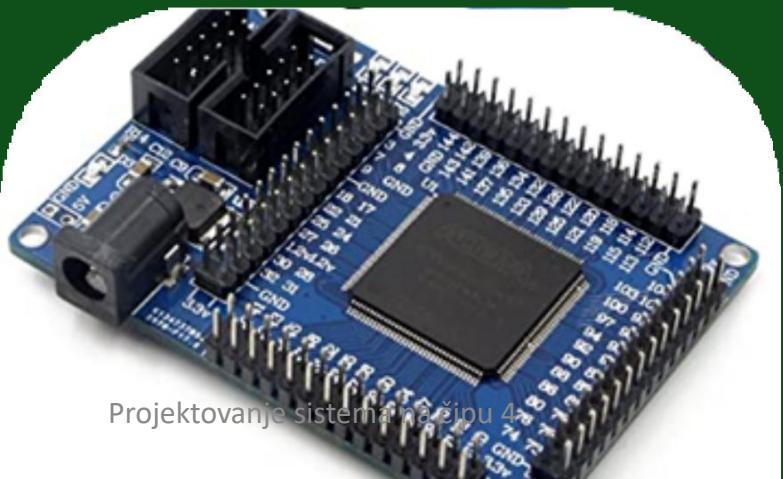


Korištena oprema



Cyclone II FPGA

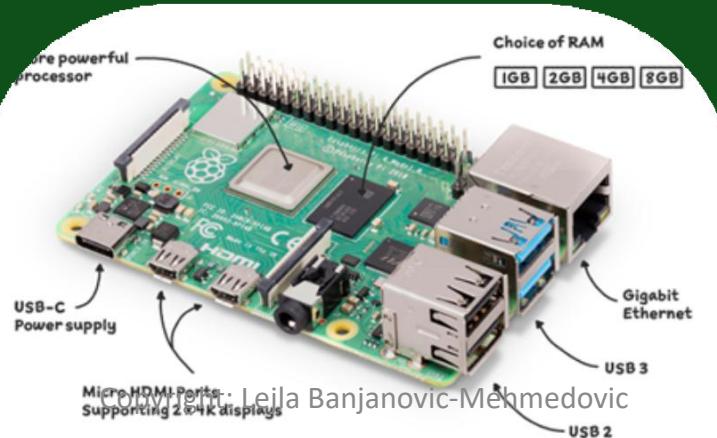
Temelji se na 1,2V, 90-nm SRAM procesu s gustoćom preko 68K logičkih elemenata (LE) i do 1,1 Mbita ugrađenog RAM-a.



Projektovanje sistema na čaju 4

Raspberry Pi

Raspberry Pi SBC imaju Broadcom sistem na čipu sa integriranom ARM kompatibilnom centralnom procesorskom jedinicom i grafičkom procesorskom jedinicom na čipu. CPU dostiže brzinu od 700 MHz do 1.2 GHz.



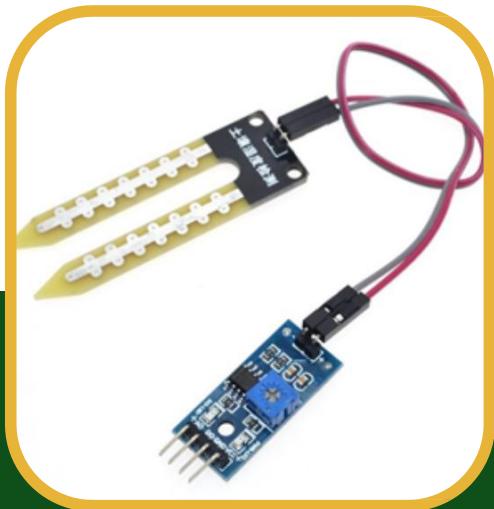
ESP8266 NodeMCU

ESP8266 je jeftin Wi-Fi mikročip, sa ugrađenim TCP/IP mrežnim softverom i mogućnošću mikrokontrolera. Ovaj mali modul omogućava mikrokontrolerima da se povežu na Wi-Fi mrežu i naprave jednostavne TCP/IP veze.





KOMPONENTE - ULAZI



YL-69 senzor vlage tla

Senzor se sastoji od dva dijela: elektronske pločice sa dva jastučića, koja detektuje sadržaj vode. Imo ugrađen potenciometar za podešavanje osjetljivosti digitalnog izlaza, LED za napajanje i LED za digitalni izlaz.



IR senzor

Infracrveni senzor je elektronski uređaj, koji emituje svjetlost da bi osjetio neke aspekte okoline. IR senzor može mjeriti toplinu objekta kao i detektirati kretanje.

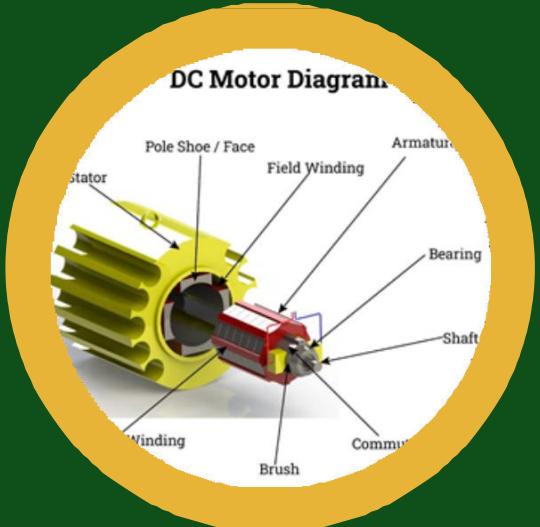


RFID

RFID (Radio-frequency identification) je bežična i beskontaktna tehnologija koja koristi radio frekvenciju kako bi se razmjenjivale informacije između prijenosnih uređaja/memorija i host računara. RFID je sastavljen iz RFID čitača i RFID medija koji izmjenjuju signale putem radio valova.



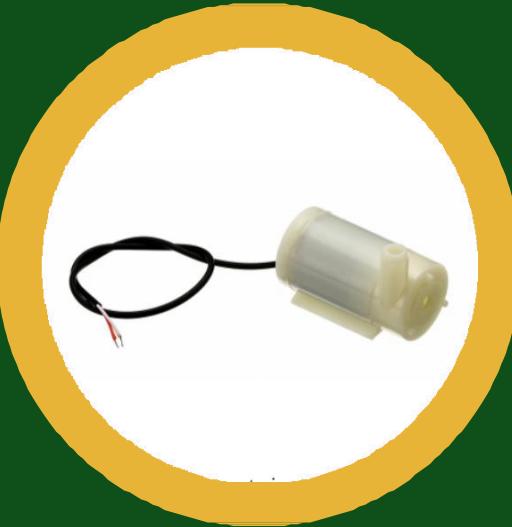
KOMPONENTE - IZLAZI



DC motori

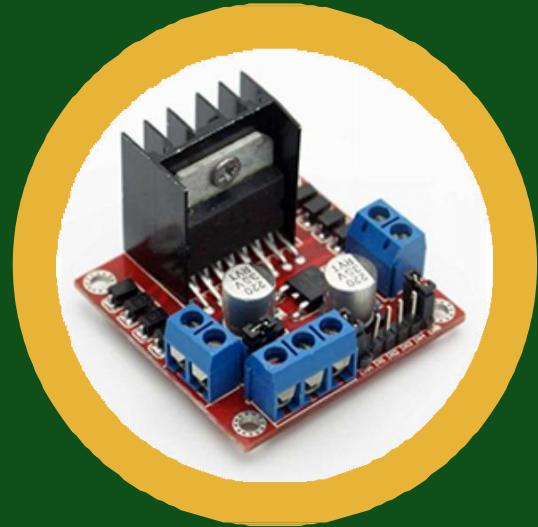
Električni DC motor je kontinualni aktuator koji pretvara električnu energiju u mehaničku. Ovo se postiže proizvodnjom kontinualne ugaone rotacije koja se može koristiti za pokretanje pumpi, ventilatora, kompresora, točkova, itd.

Brzinom vrtnje DC motora upravljamo pomoću PWM signala



Vodena pumpa

Pumpa za vodu se pokreće pomoću 6V releja koji djeluje kao prekidač, ali umjesto da ga fizički dodirnemo kako bismo ga uključili/isključili, mi dovodimo napon kako bismo ga uključili.

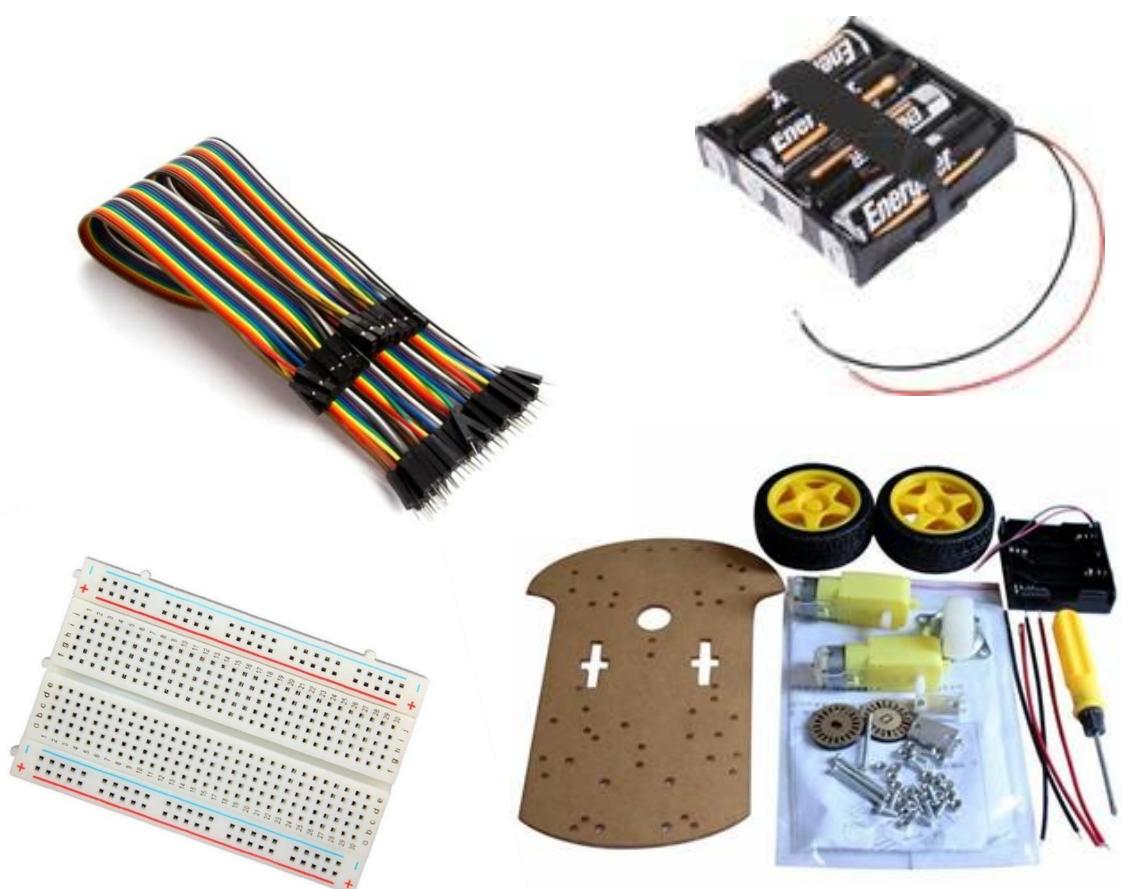


L298N Motor driver

L298N je driver motora sa dvostrukim H-mostom, koji dozvoljava kontrolu brzine i smjera vrtnje DC motora istovremeno. Napon napajanja ovog drivera je između 5 i 35V. L298 H Bridge je baziran na l298 čipu.



DODATNA OPREMA



Projektovanje sistema na čipu 4

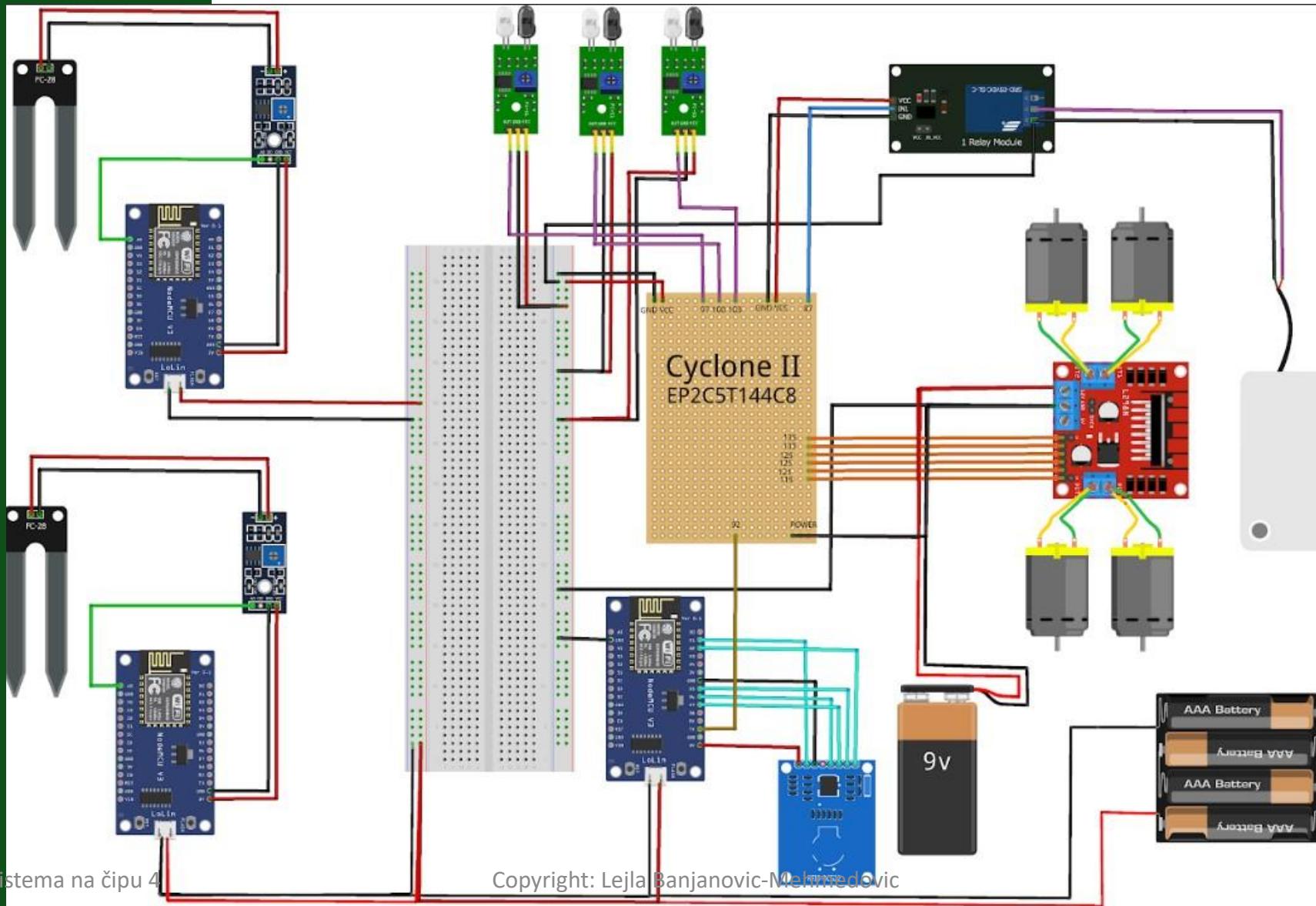


Copyright: Lejla Banjanovic-Mlekmendovic

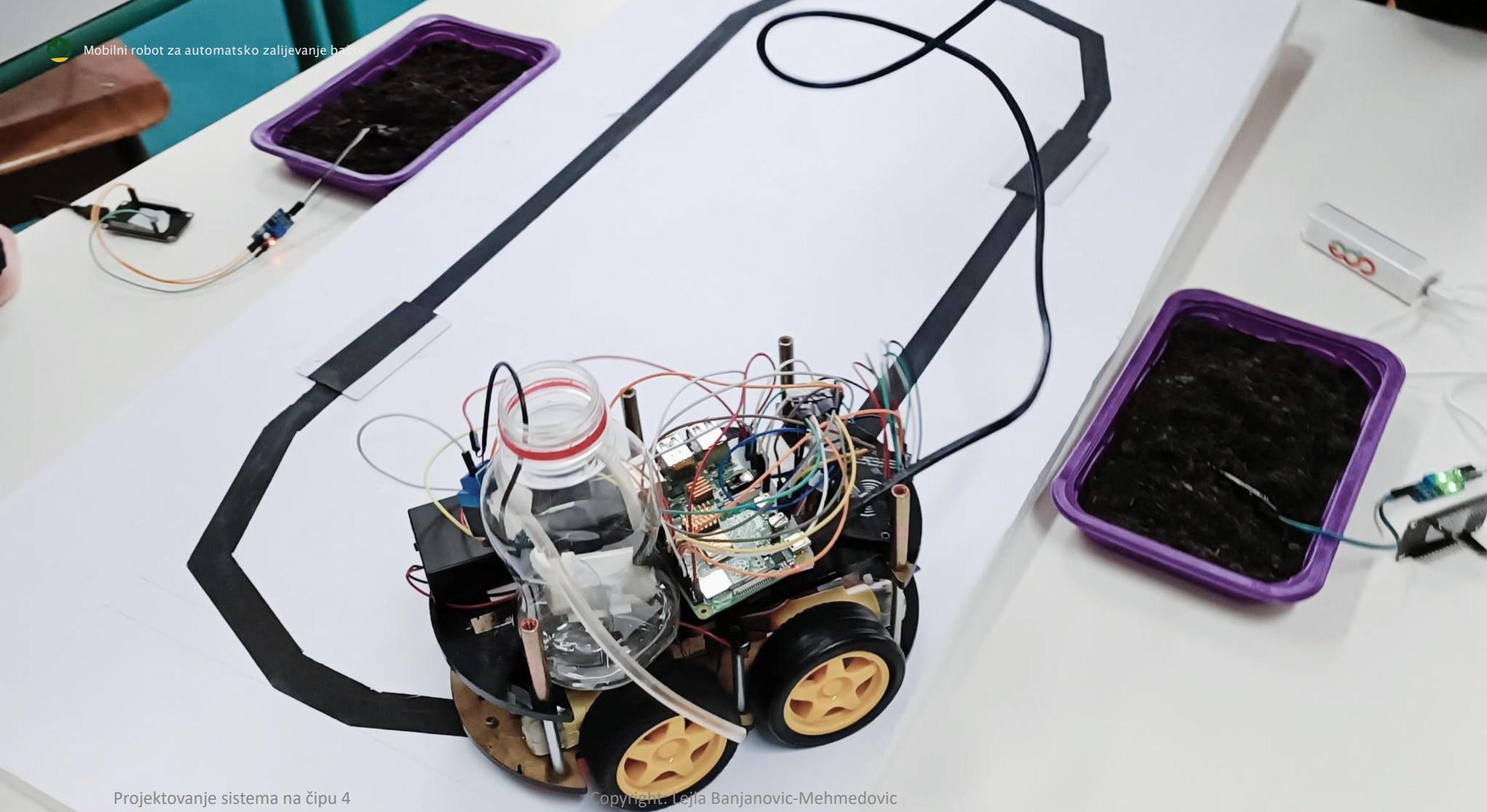
- Baterije
- Breadbord
- Jumper žice
- Arudino Smart Robot Car Kit



S h e m a p o v e z i v a n j a k o m p o n e n t i

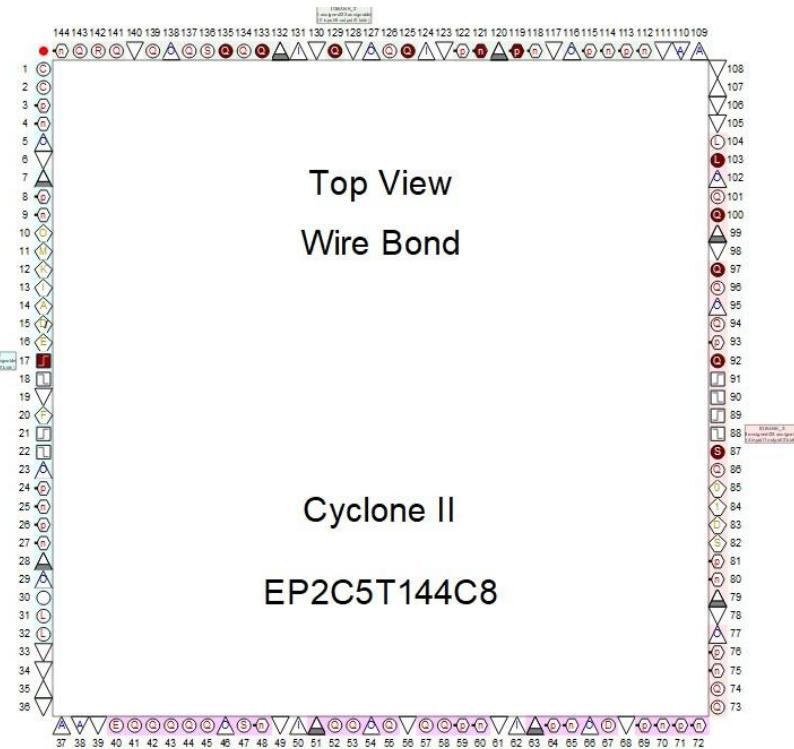


F P G A



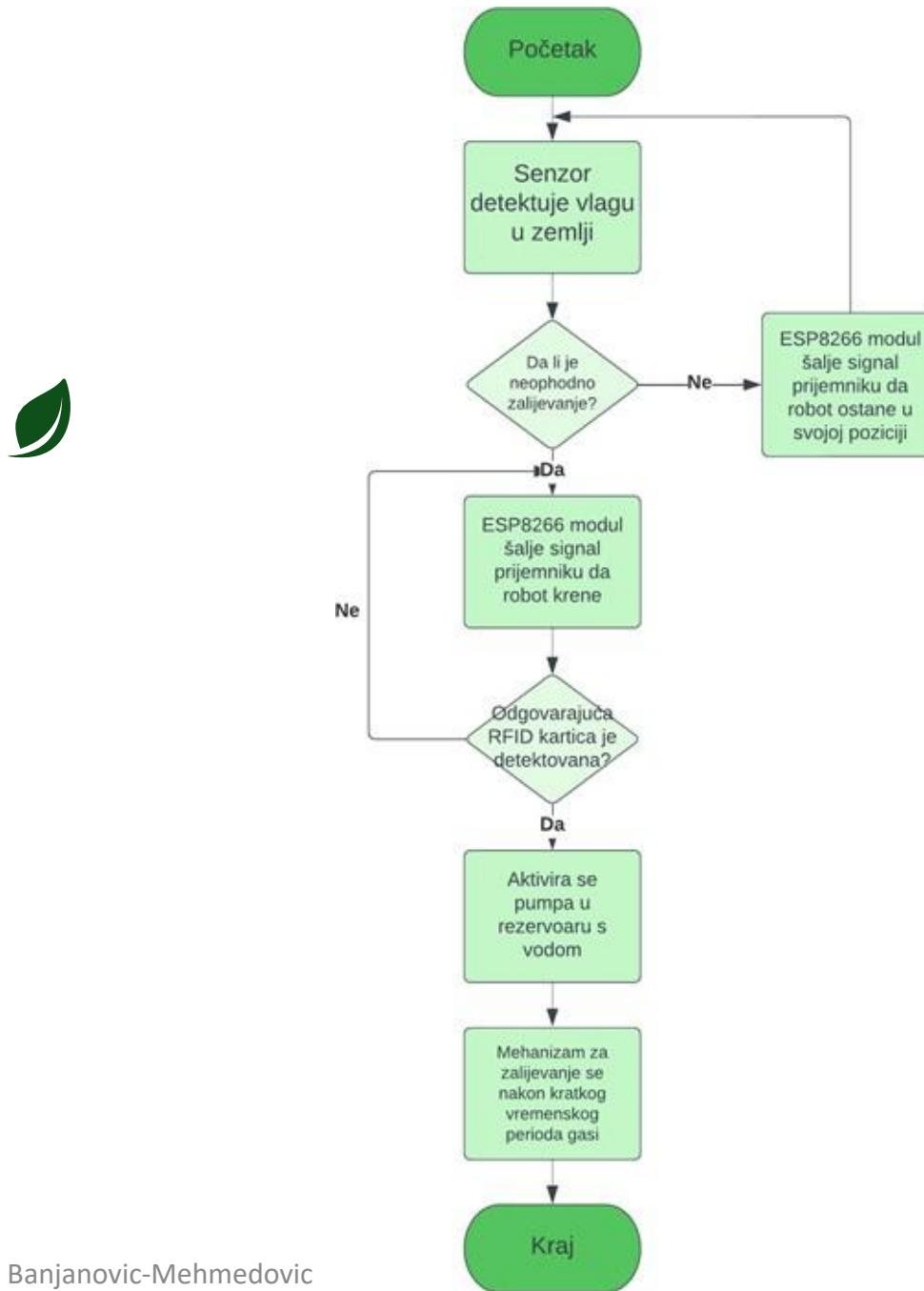


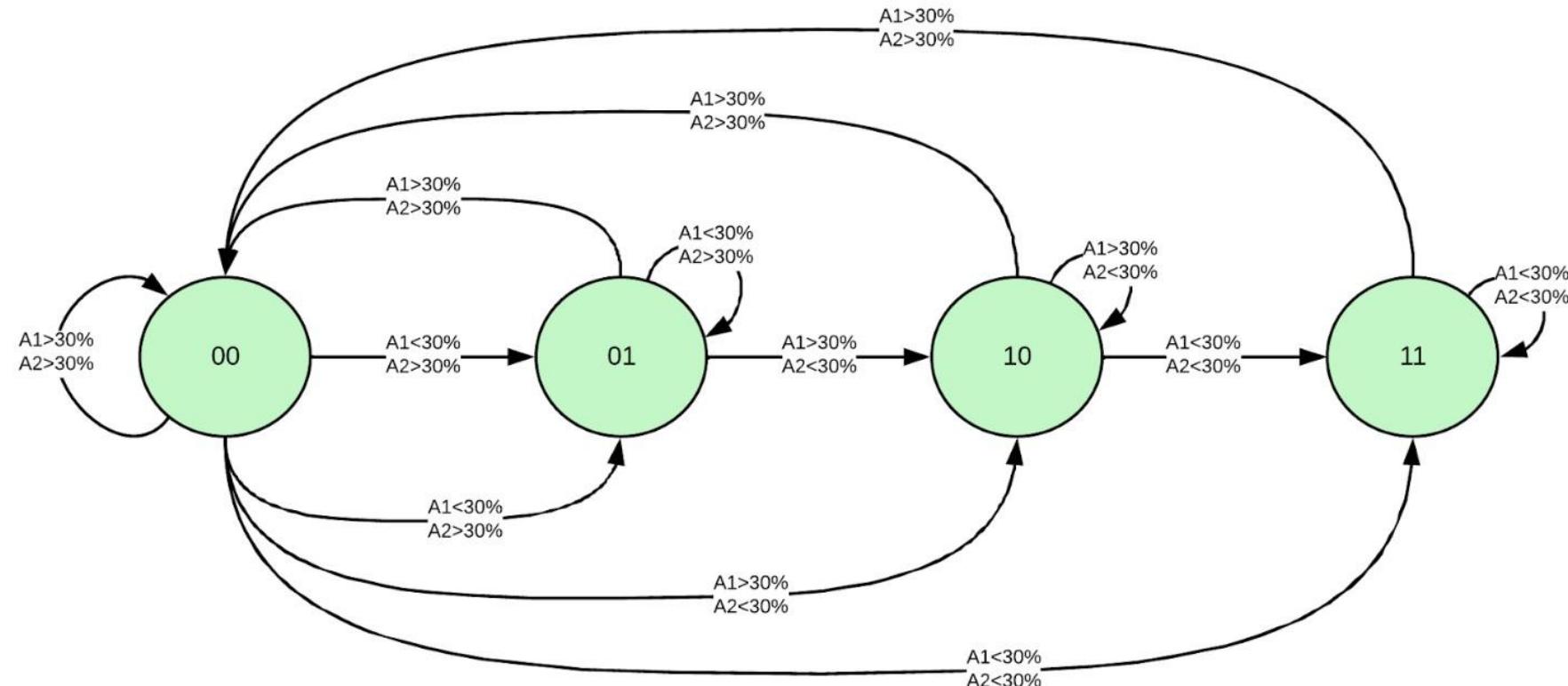
Raspored pinova na CYCLONE II



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
in_RxD	Input	PIN_92	3	B3_N0	PIN_92	3.3-V L...efault)		24mA (default)	
in_clk	Input	PIN_17	1	B1_N0	PIN_17	3.3-V L...efault)		24mA (default)	
out_ena	Output	PIN_135	2	B2_N1	PIN_135	3.3-V L...efault)		24mA (default)	
out_enb	Output	PIN_119	2	B2_N0	PIN_119	3.3-V L...efault)		24mA (default)	
out_in1	Output	PIN_133	2	B2_N1	PIN_133	3.3-V L...efault)		24mA (default)	
out_in2	Output	PIN_129	2	B2_N1	PIN_129	3.3-V L...efault)		24mA (default)	
out_in3	Output	PIN_125	2	B2_N0	PIN_125	3.3-V L...efault)		24mA (default)	
out_in4	Output	PIN_121	2	B2_N0	PIN_121	3.3-V L...efault)		24mA (default)	
out_pumpa	Output	PIN_87	3	B3_N1	PIN_87	3.3-V L...efault)		24mA (default)	
in_qt1	Input	PIN_97	3	B3_N0	PIN_97	3.3-V L...efault)		24mA (default)	
in_qt2	Input	PIN_100	3	B3_N0	PIN_100	3.3-V L...efault)		24mA (default)	
in_qt3	Input	PIN_103	3	B3_N0	PIN_103	3.3-V L...efault)		24mA (default)	

Dijagram toka



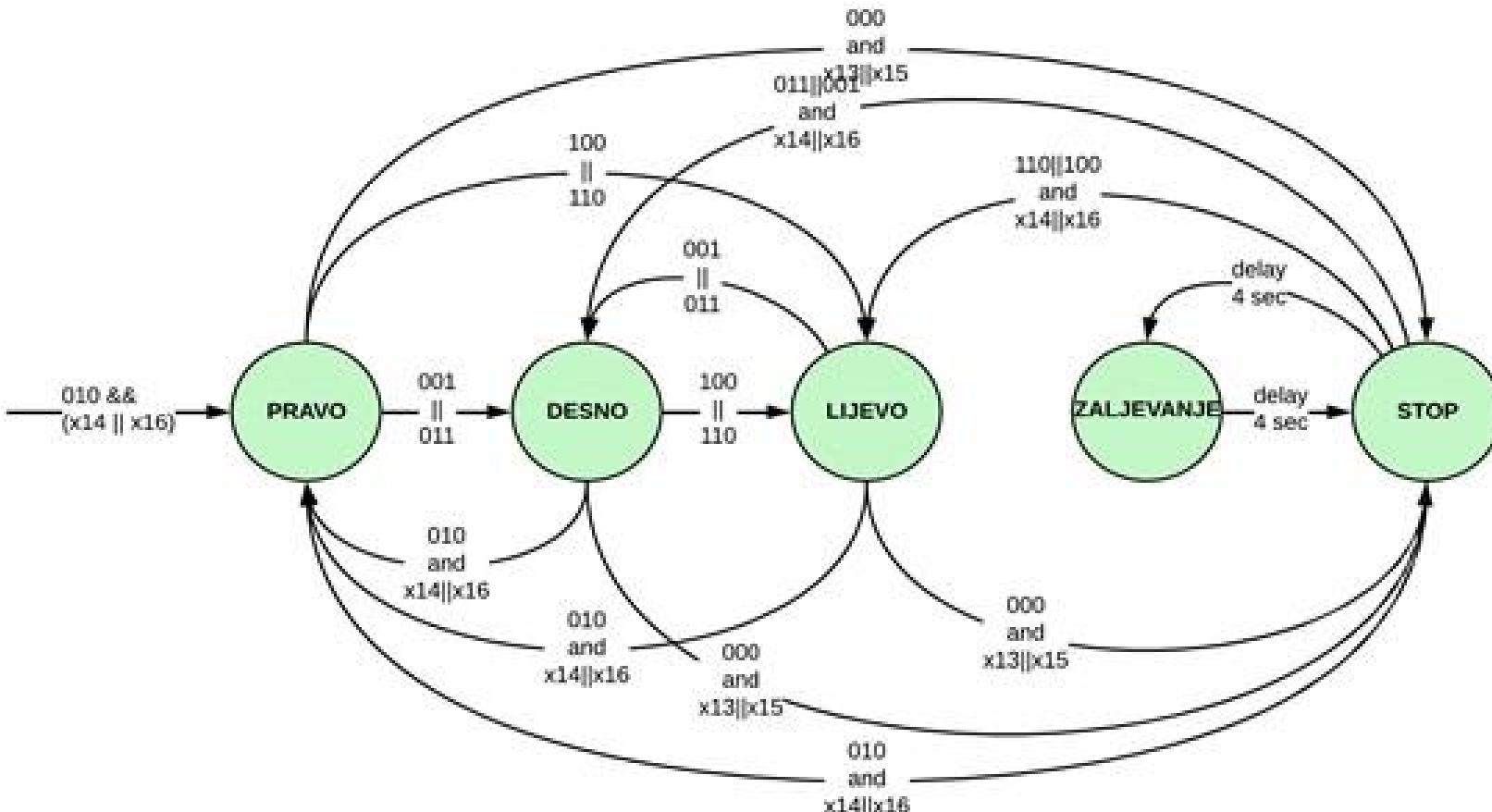


Četiri moguća slučaja:

- 00 – Nije potrebno zaljevanje niti jednu saksiju
- 01 – Potrebno je zaliti prvu saksiju
- 10 – Potrebno je zaliti drugu saksiju
- 11 – Potrebno je zaliti obje saksije

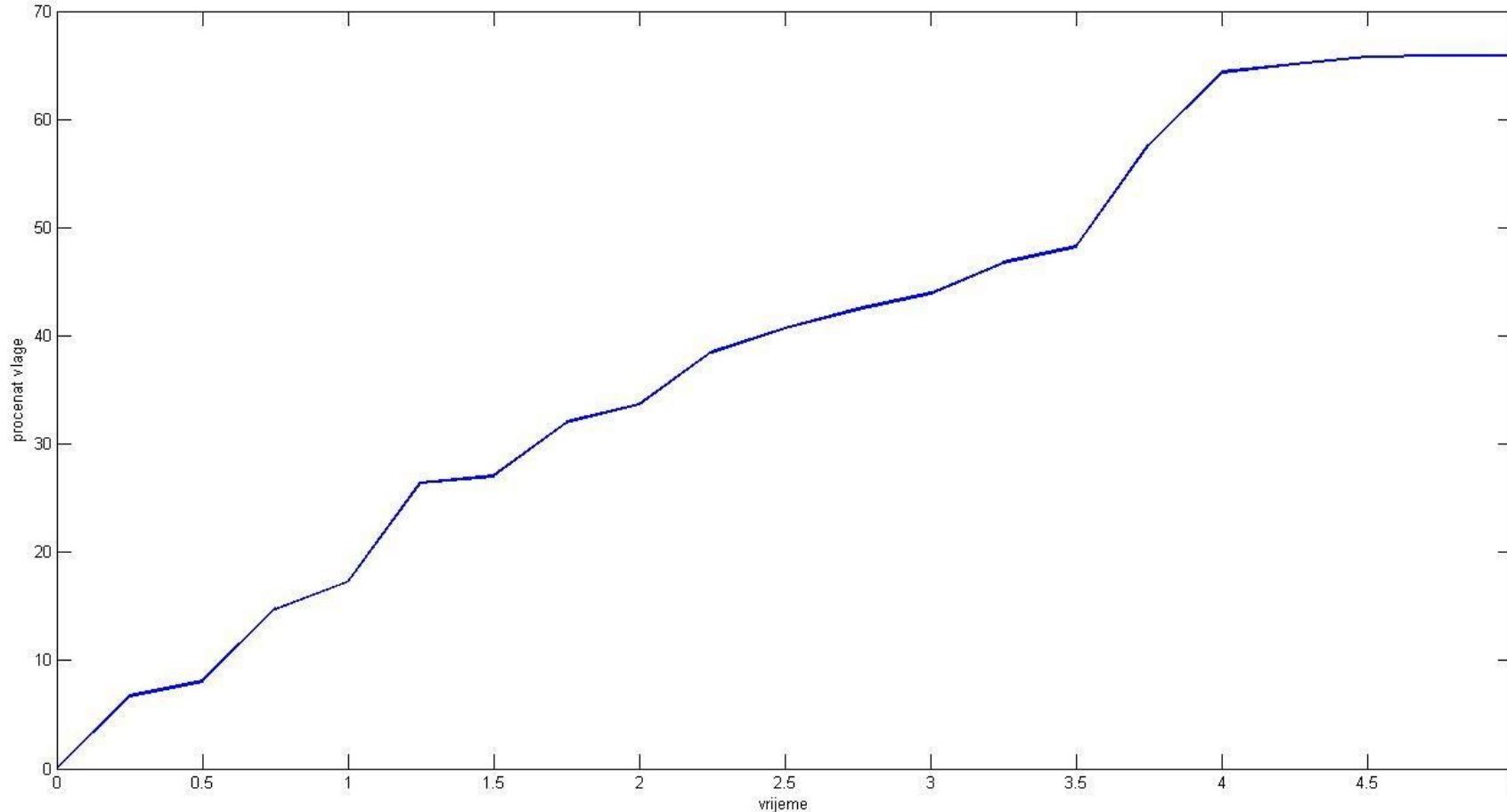


ULAZI	IZLAZI
IR senzori (praćenje crne linije)	DC Motori (točkovi robota)
Rx (podatak koji dolazi sa NodeMCU)	Pumpa (zalijevanje)



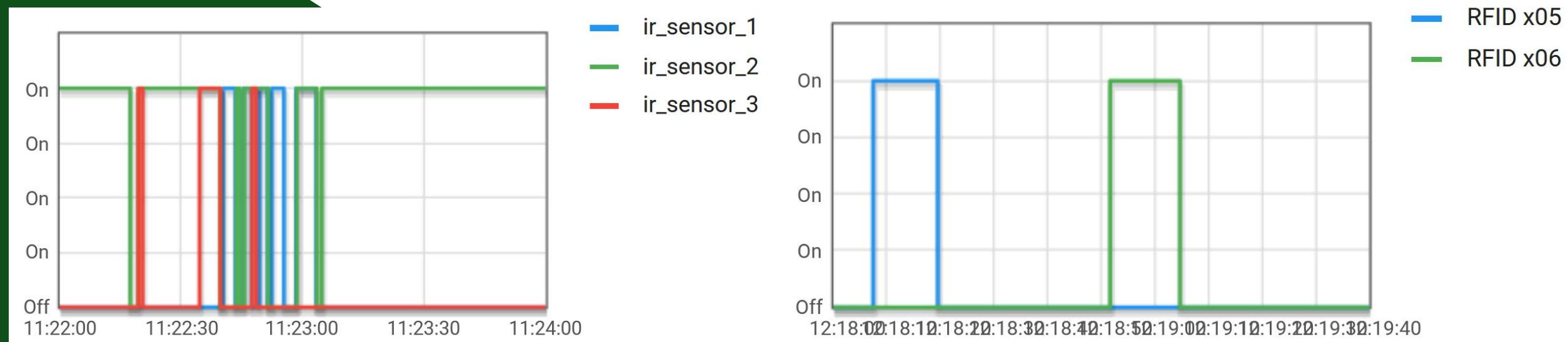
Grafička interpretacija

PRIKAZ PROCESA ZALIJEVANJA



Grafička interpretacija

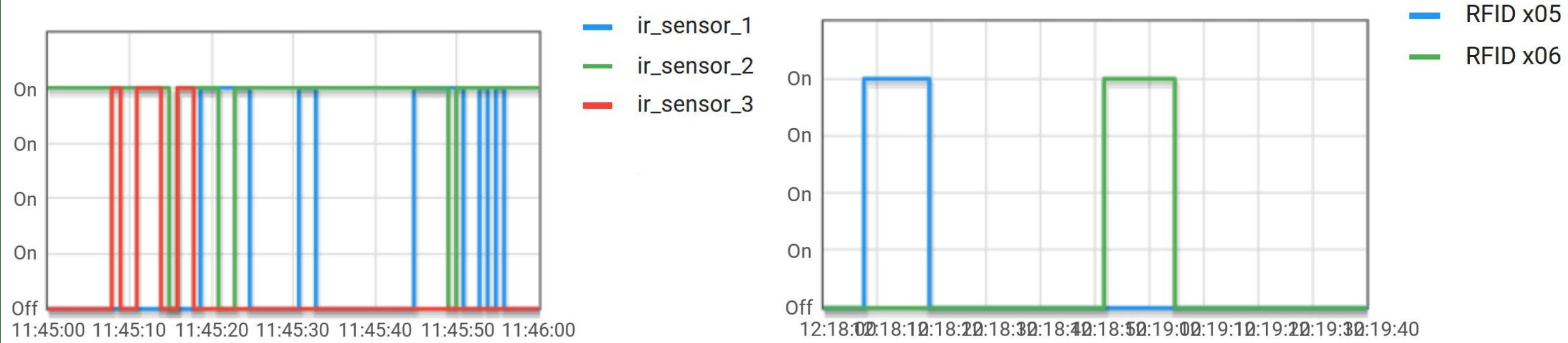
PRIKAZ KRETANJA ROBOTA NA OSNOVU IR SENZORA I RFID TAGOVA



Slučaj 1:
Zalijevanje prve saksije

Grafička interpretacija

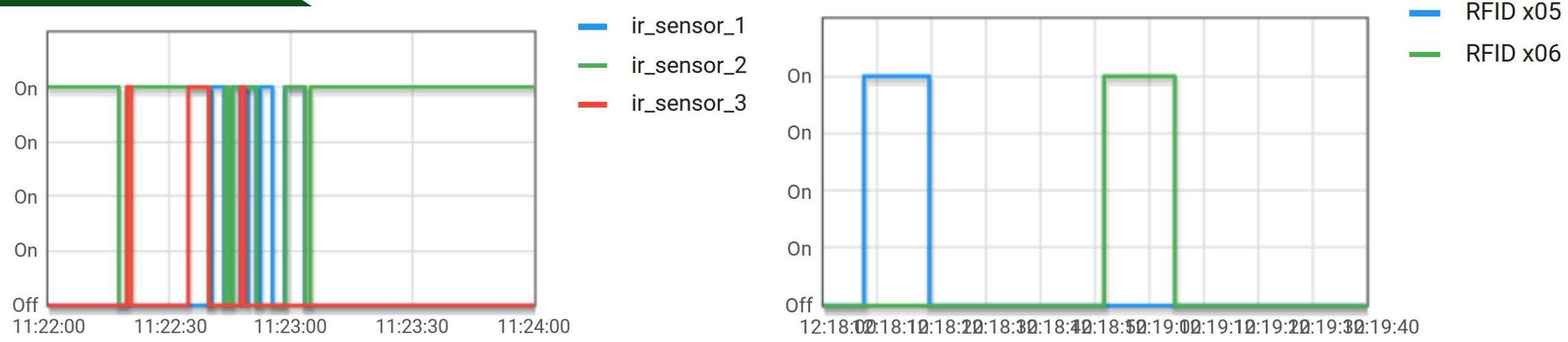
PRIKAZ KRETANJA ROBOTA NA OSNOVU IR SENZORA I RFID TAGOVA



Slučaj 2:
Zalijevanje druge saksije

Grafička interpretacija

PRIKAZ KRETANJA ROBOTA NA OSNOVU IR SENZORA I RFID TAGOVA



Slučaj 3:
Zalijevanje obje saksije



Projektovanje sistema na čipu 4

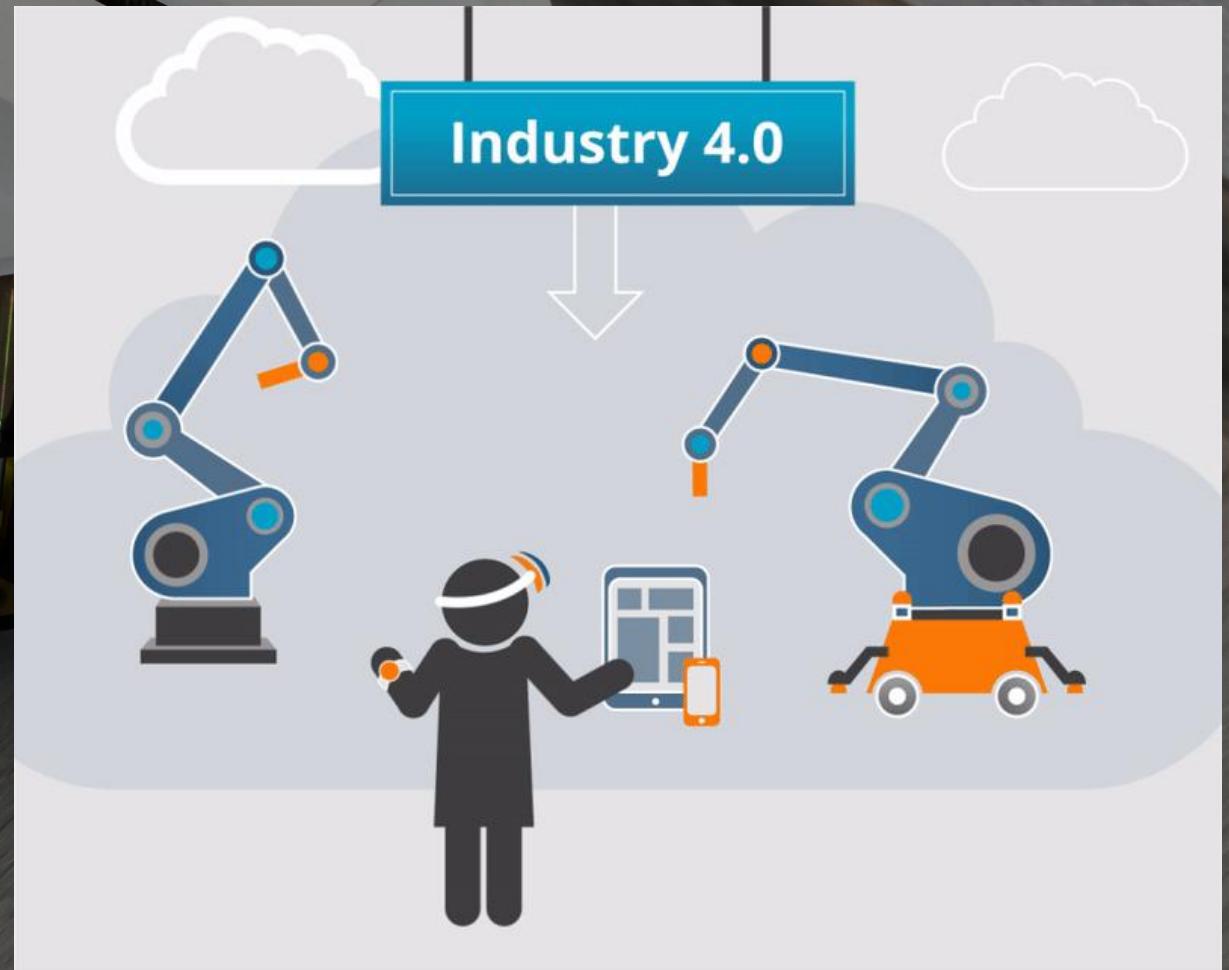
Copyright: Lejla Banjanovic-Mehmedovic



Primjer 2: Dostava
proizvoda u skladištu

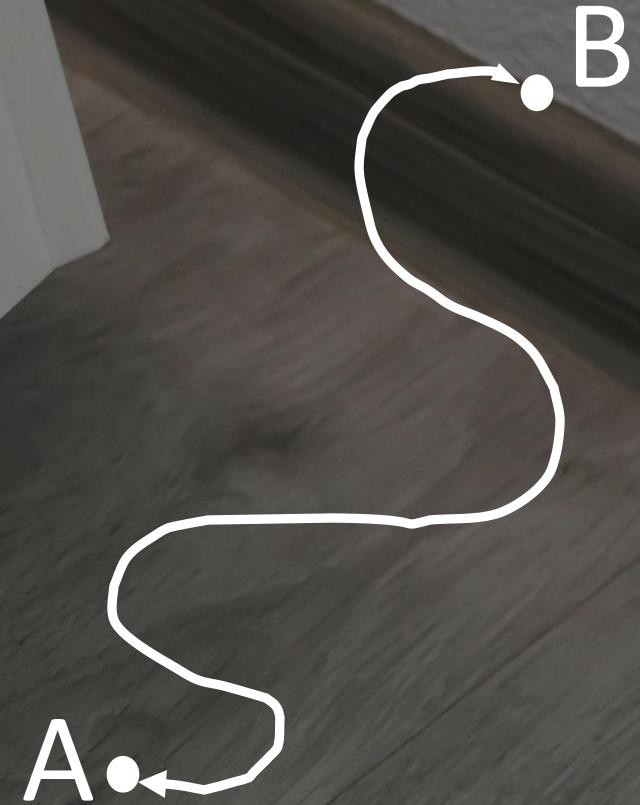
Industrija 4.0

- Industrija 4.0 odnosi na povezivanje digitalnog i fizičkog svijeta koji je srž današnje proizvodnje.
- To uključuje korištenje tehnologije "Interneta stvari" (IoT), umjetne inteligencije, sveprisutnog umrežavanja i povezivanja, velikih podataka i boljih sučelja ljudskih mašina za stvaranje pametnijeg načina proizvodnje.



Logistika 4.0

- Iako postoji mnogo vrsta i definicija logistike, u najjednostavnijem smislu logistika predstavlja prenos robe, imovine i podataka u poslovnom smislu.
- Logistika 4.0 obuhvata sljedeće aplikacije: od transporta bez vozača do inteligentnih kontejnera, pametnog skladištenja, pametnih portova, pametnih polica do razmjene ljudi i informacija u svim mogućim logističkim lancima i kontekstima.

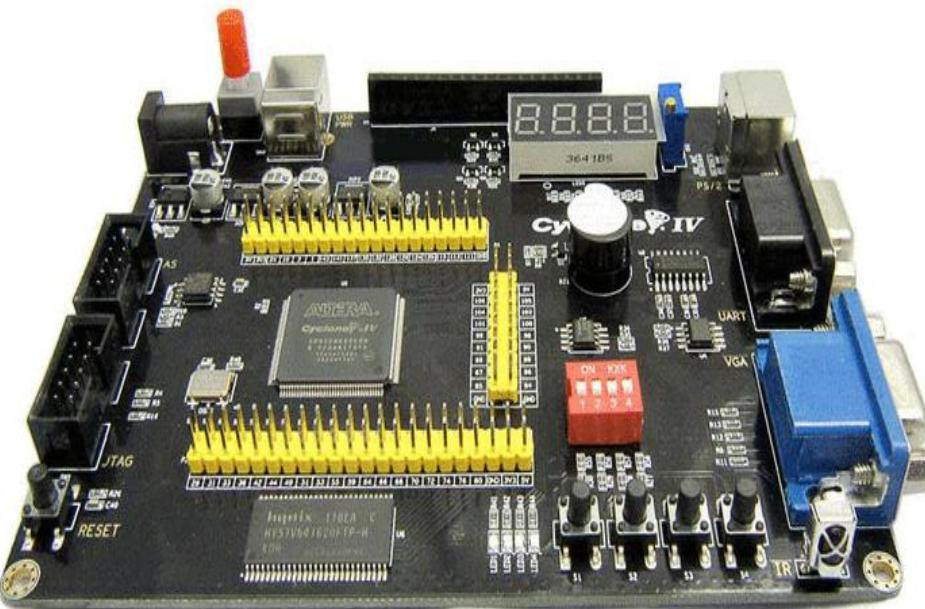


Mobilni roboti u logistici

- Sama ideja projekta je podstaknuta mobilnim robotima za logistiku koji se primjenjuju u kompanijama za e-commerce kao što su Amazon, Ebay, Rakuten i slično.



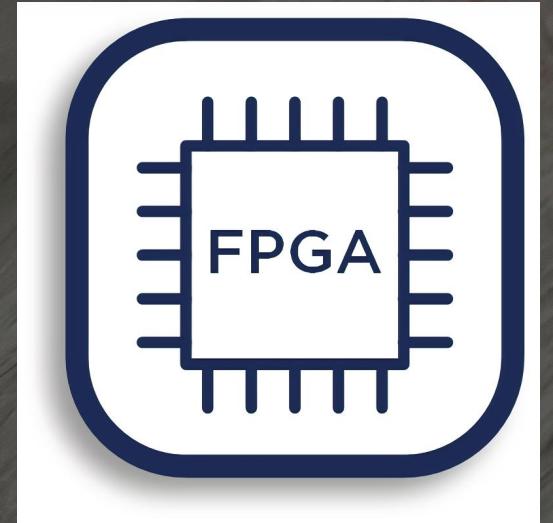
FPGA



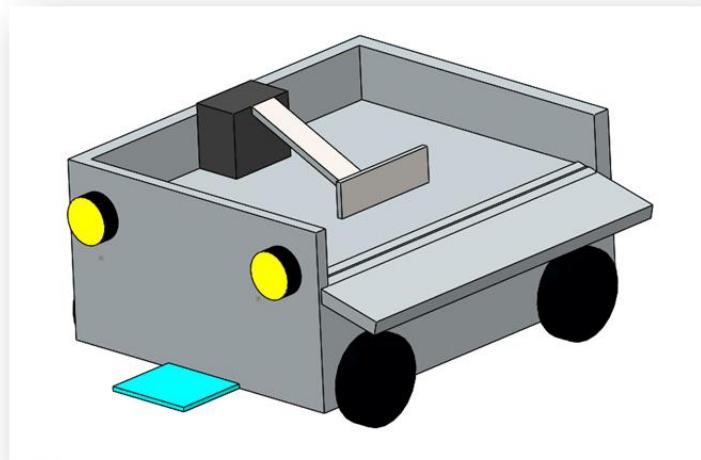
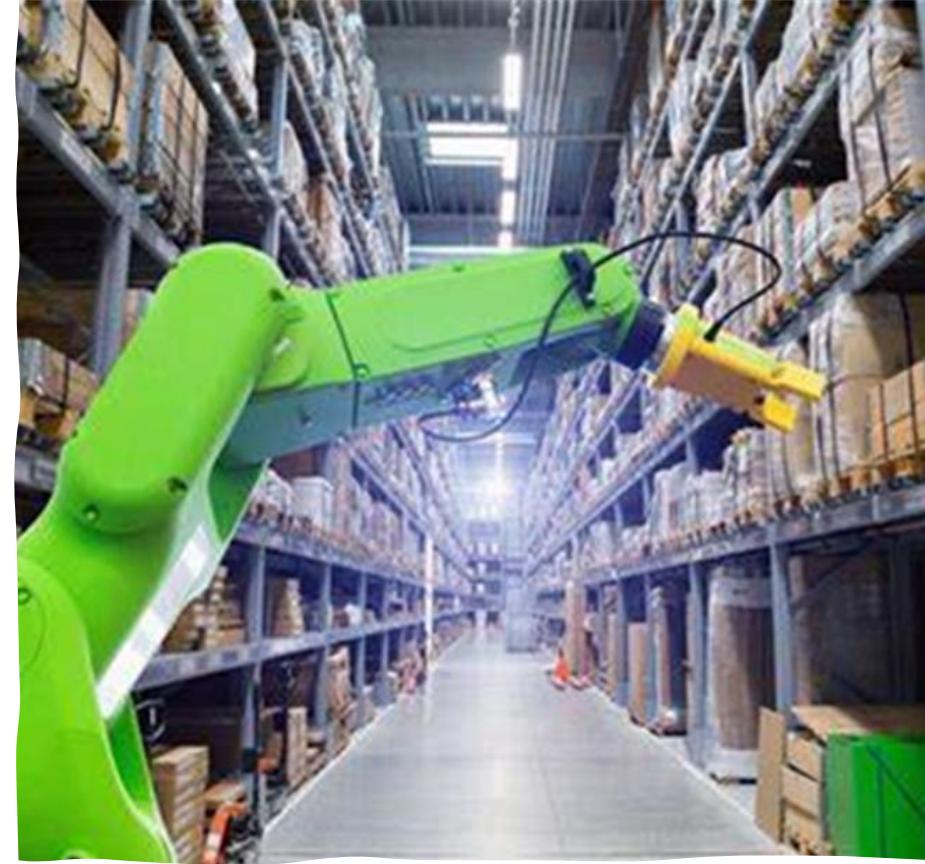
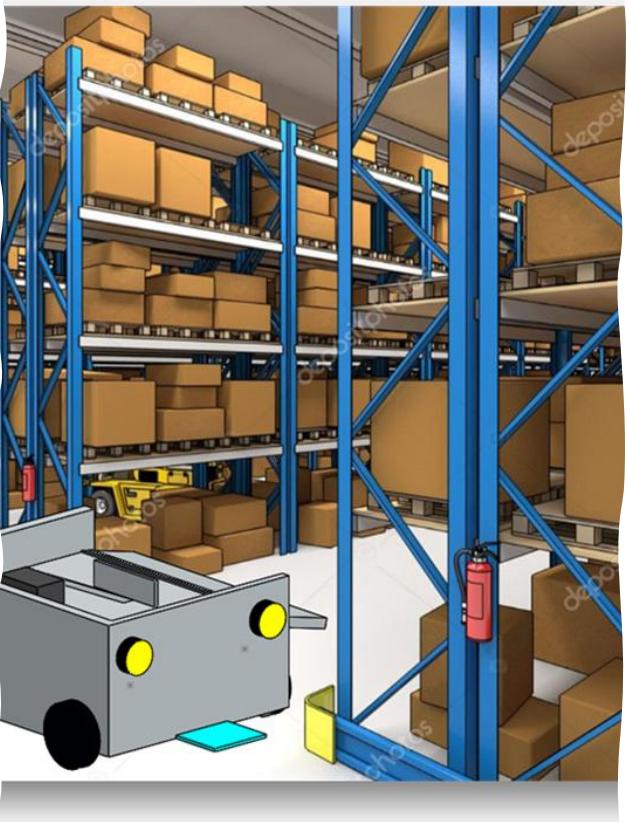
- FPGA (engl. Field-Programmable Gate Array) predstavlja integrisano kolo projektovano tako da se njegova unutrašnja struktura može konfigurisati od strane krajnjeg korisnika.
- Za konfigurisanje unutrašnje strukture se koriste HDL (engl. Hardware Description Language).
- Kada se kreira dizajn za FPGA, tj kada se opisuje hardver, sve će na kraju završiti kao fizičko kolo.

Zašto baš FPGA?

- Energetski efikasan.
- Svaki PIN može vršiti bilo koju potrebnu funkciju (I/O, UART, itd.)
- FPGA imaju osobinu paralelizma.
- Precizno dizajniran FPGA može izvršiti bilo koju funkciju brže od procesora.

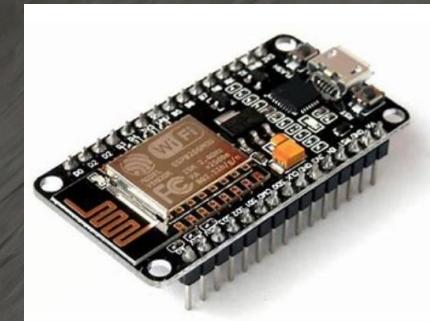
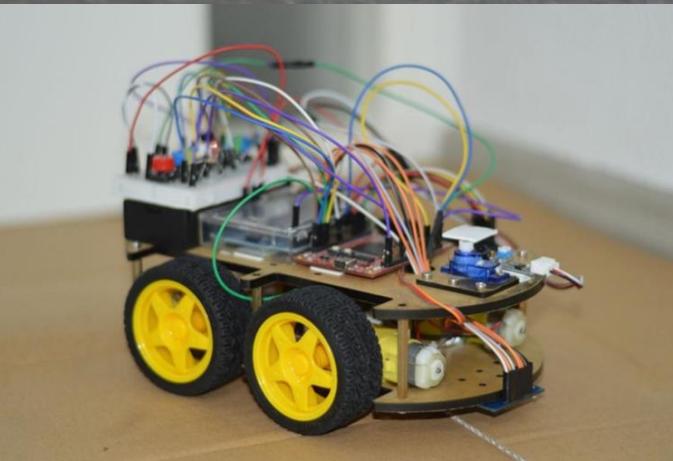
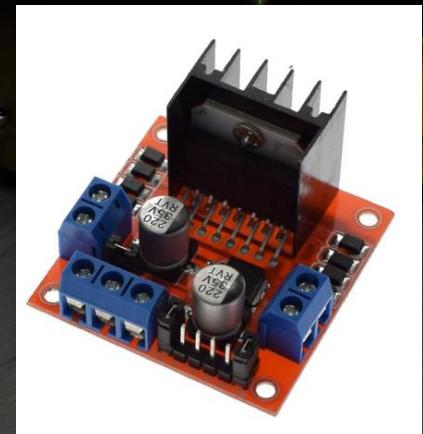


Specifikacija taska i dizajn rješenja



Korištene komponente

- DC motor
- RFID RC522
- RFID tagovi
- IR senzor
- L298N motor driver
- Servo motor
- Cyclone IV
- NodeMCU



RFID

U našem projektu osnovnu poveznicu čini RFID koji preko Arduina prikuplja informacije o kretanju robota te zatim te podatke dalje šalje UART vezom na Cyclone pločicu.

RFID sistem je sastavljen iz RFID čitača i RFID medija koji izmjenjuju signale putem radio valova

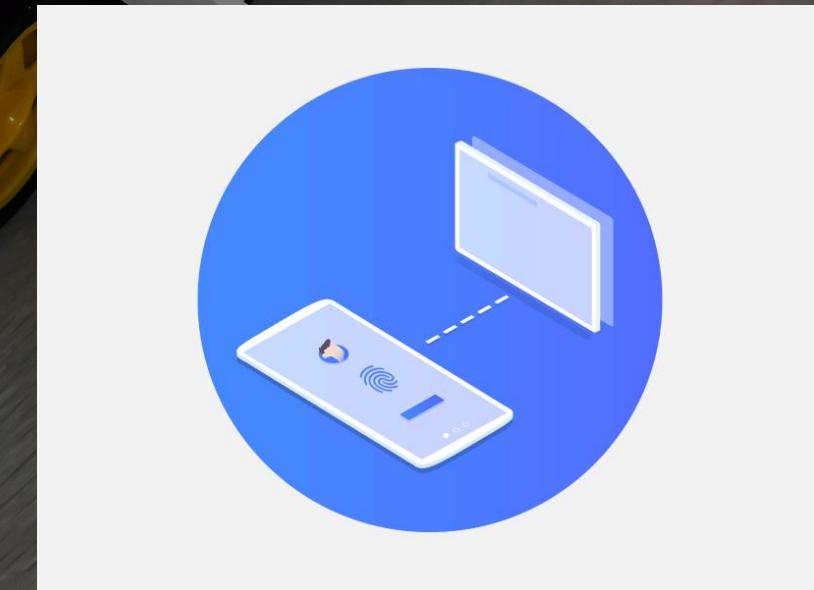
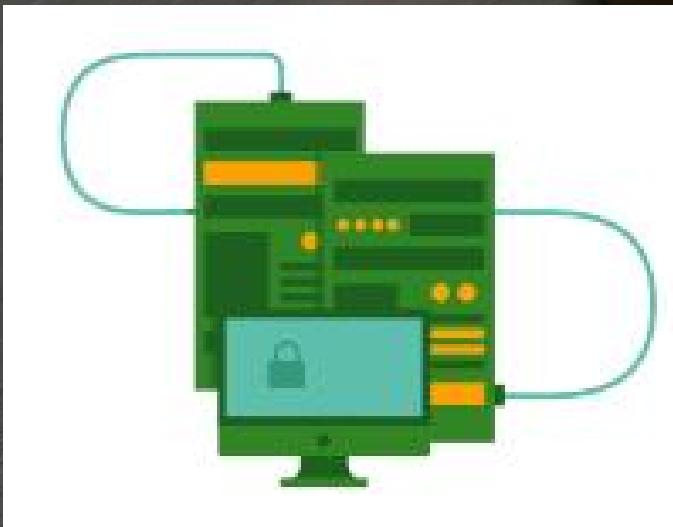
Mediji za identifikaciju koji su dostupni za identifikaciju mogu biti kartice, privjesci, narukvice, NFC..

Pinovi RFID-a sa Arduinom su povezani na sljedeći način:

RFID pinovi	ARDUINO digital pinovi
RST	9
SDA	10
MOSI	11
MISO	12
SCK	13

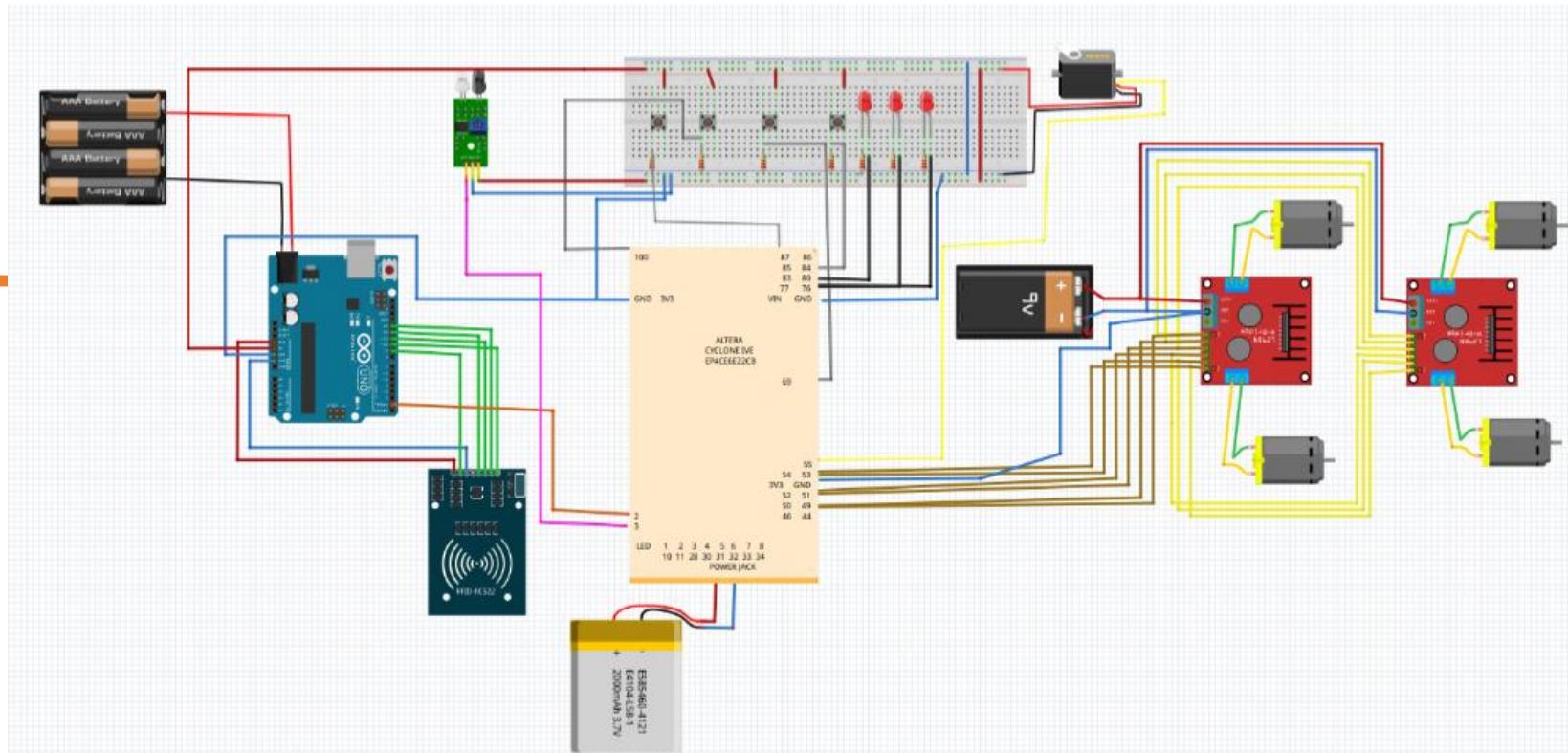
Komunikacija

- UART protokol
- Baziran na serijskoj komunikaciji.
- MQTT protokol
- Omogućava povezivanje uređaja putem interneta.



Shema spajanja

- Prednosti:
- Implementacija UART protokola na FPGA
- Paralelno upravljanje radom 4 motora



fritzina

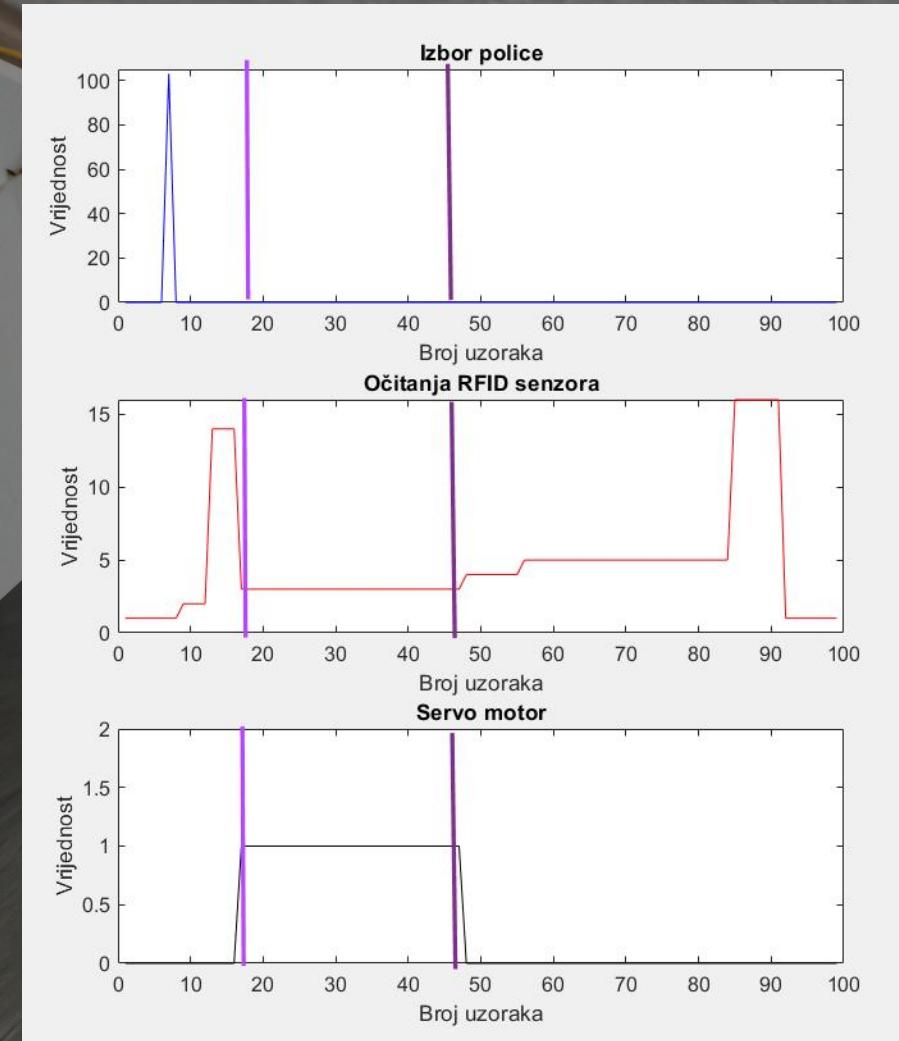
Logika rada mobilnog robo

- Sa Arduina UART protokolom "dolaze" podaci, to jeste dolaze brojevi od 1-11, u zavisnosti od broja zavisiti ce i logika rada
- Brojevi 2,5,6,8,11 su za skretanje dok brojevi 3,4,9,10 označavaju police, 3,4 su u prvom redu, dok su 9,10 u drugom redu
 - 1 je startni
 - 7 je završni broj
- U zavisnosti od broja primljenog sa Arduina palit će se i built-in diode na FPGA

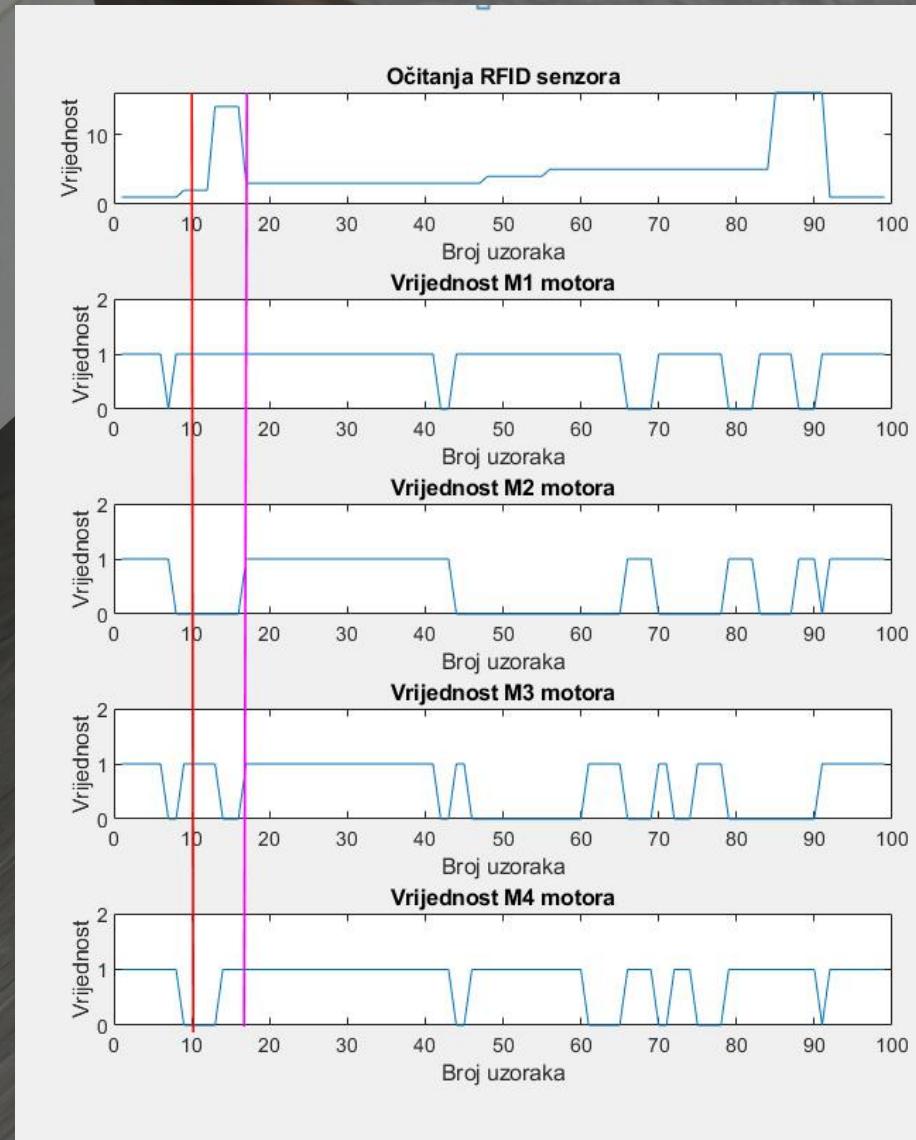
Algoritam rada mobilnog robota za logističke usluge

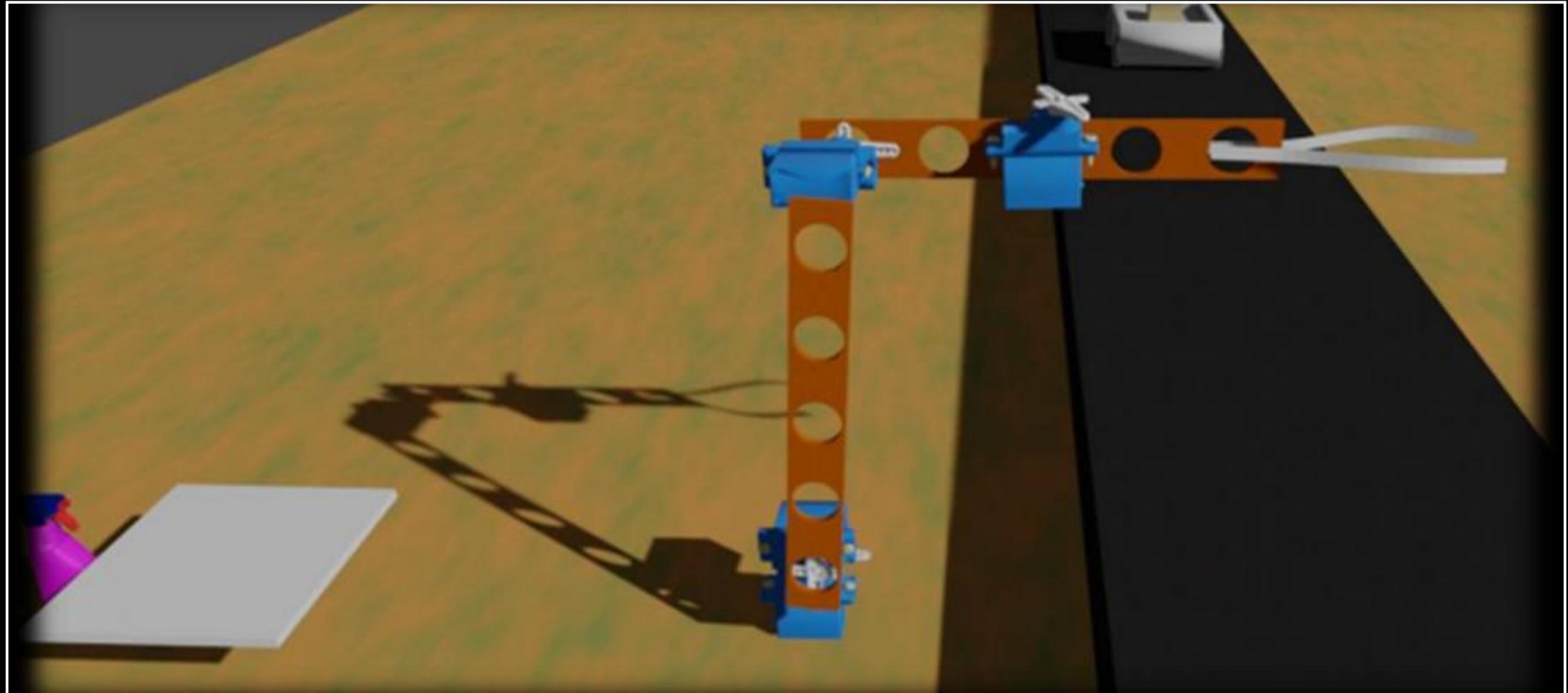


Realtime rezultati



Realtime rezultati



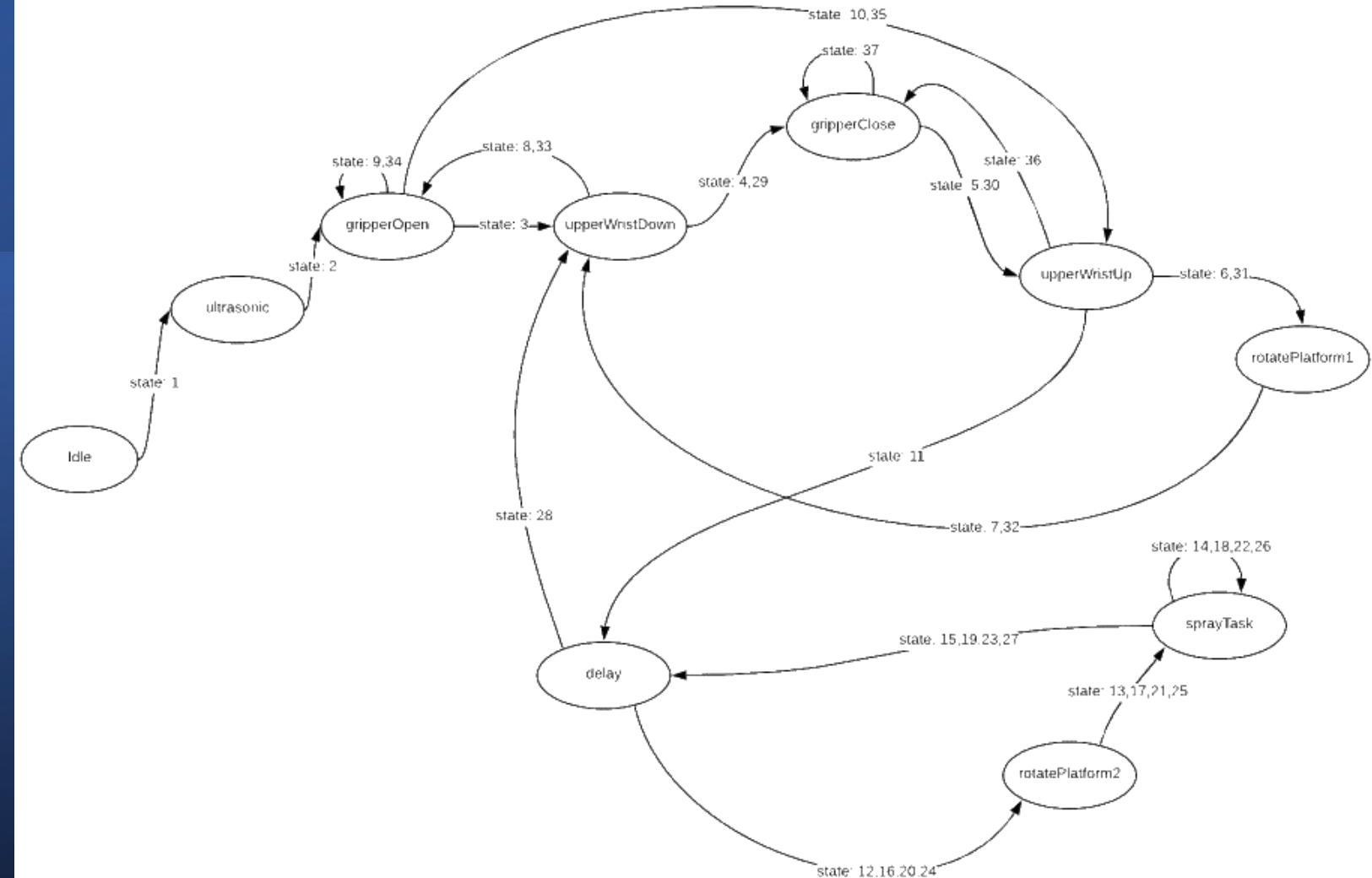


Primjer 3: Dezinfekcija paketa pomoću robotske ruke

Dezinfekcija paketa pomoću robotske ruke

- Pokretni objekat (traku ili neku vrstu vozila) koji će paket premjestiti do određene prepreke
- Nakon toga, robotska ruka podiže paket i prebacuje ga na drugu lokaciju i pokreće sistem za dezinfekciju
- Nakon uspješne dezinfekcije paketa, robotska ruka paket prebacuje do početnog pokretnog objekta

Dijagram konačnog stanja



Opis korištenog hardvera

Robotska ruka:

- 4x SG90 servo motorR
- 1x HC-SR04 ultrasonični senzor
- Konstrukcijski dijelovi (drveni štapići, plastična hvataljka, žica za hvataljku)



Prskalica:

- 2x SG90 servo motor
- Posuda sa tekucinom

Pokretna platforma:

1x SG90 servo motor

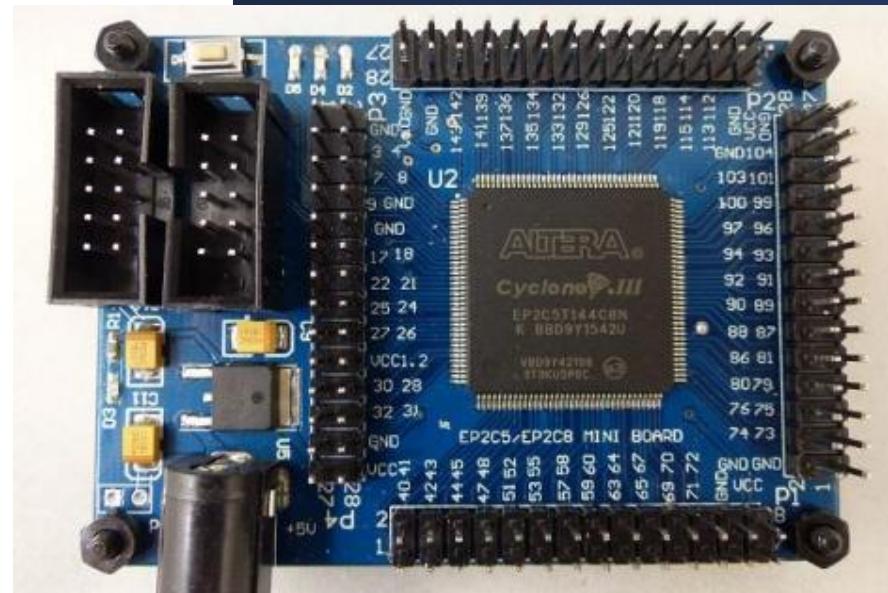
Altera Cyclone ii fpga

- Altera Cyclone II 90nm FPGA izrađeni su u cilju postizanja niske cijene za troškovno osjetljive aplikacije
- Cyclone II FPGA isporučuju visoke performanse i nisku potrošnju energije
- Sadrži do 68.416 logičkih elemenata (LE) i pruža do 622 upotrebljiva I/O pina
- Proizvode se na pločicama od 300 mm koristeći TSMC 90nm tehnologiju
- nude 60% veće performanse i upola manju potrošnju energije od konkurenčkih 90nm FPGA



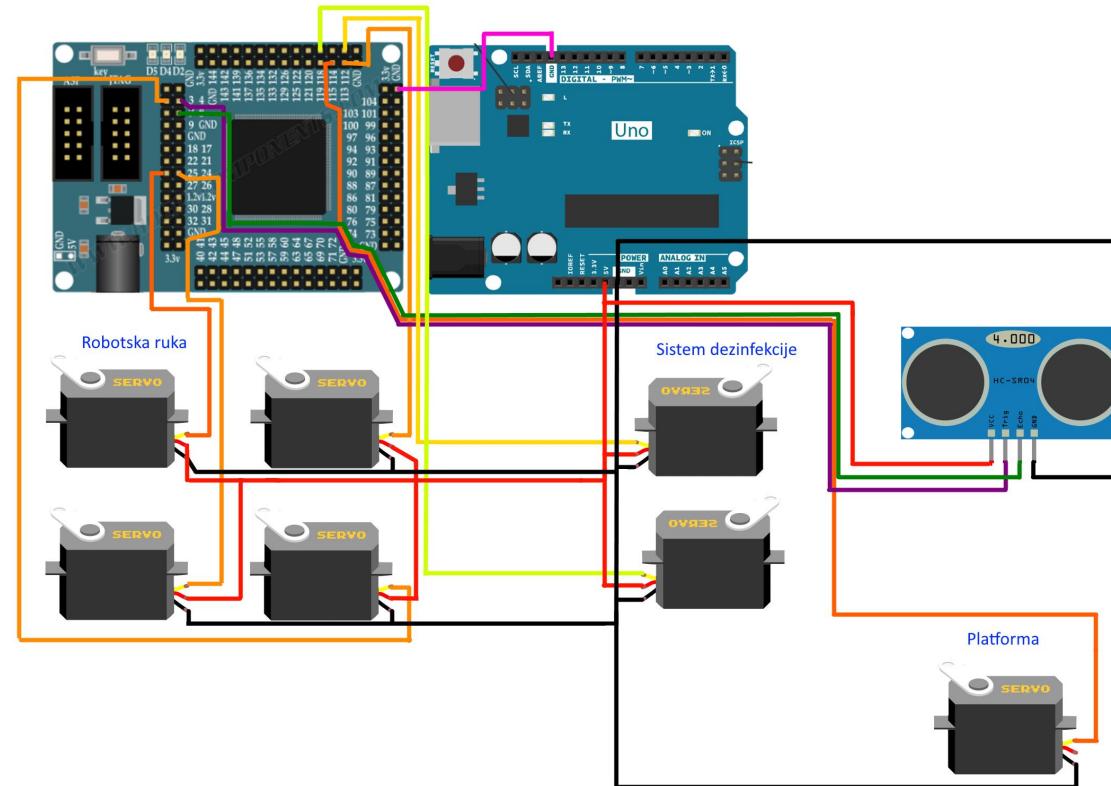
Cyclone II EP2C5 Mini Dev Board

- EP2C5T144 čip
- EPICS4 za konfiguraciju EPROM (4 Mbit)
- On-board oscilator od 50 MHz
- On-board DC soket za napajanje od 5V
- On-board LED za indikaciju napajanja
- Reset prekidač
- Visoko kvalitetni kondenzator za filtriranje napajanja
- Podržava NIOS II



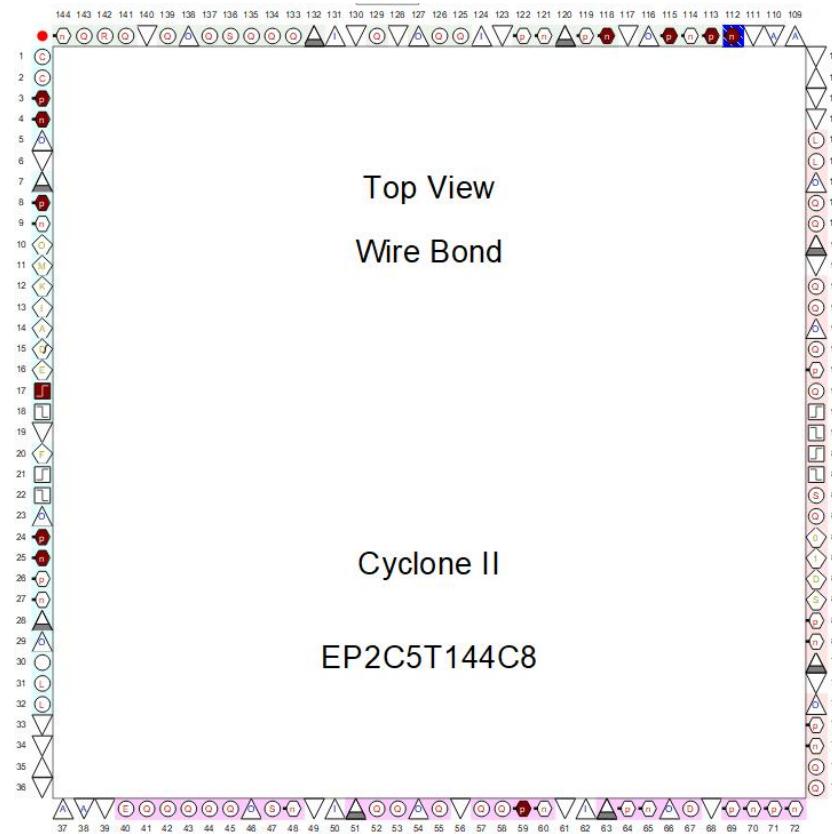
ŠEMA SPAJANJA HARDVERA

HARDVER	PIN	I/O
CLOCK	17	Input
Hvataljka (SG90)	3	Output
Gornji dio ruke	113	Output
Donji dio ruke	25	Output
Platforma ruke	24	Output
Echo (HC-SR04)	8	Input
Trigger (HC-SR04)	4	Output
Platforma za dezinfekciju (SG90)	115	Output
Sistem za dezinfekciju (SG90)	118 i 112	Output



Pin planner za fpga

Node Name	Direction	Location
out armPlatform	Output	PIN_24
in clk	Input	PIN_17
out downWrist	Output	PIN_25
in echo	Input	PIN_8
out gripper	Output	PIN_3
out led	Output	PIN_59
out sparyingPlatform	Output	PIN_115
out sparyingServoOne	Output	PIN_118
out sparyingServoTwo	Output	PIN_112
out trig	Output	PIN_4
out upWrist	Output	PIN_113
<<new node>>		



VERILOG KOD

Deklarisanje modula

```
module sg90(input wire clk, output wire gripper, output wire upWrist, output wire downWrist,  
output wire armPlatform, output wire sprayingPlatform, output wire sprayingServoOne, output  
wire sprayingServoTwo, output wire trig, input wire echo, output wire led);  
  
reg [5:0] state=6'd1;
```

Globalna varijabla za state mašinu:

reg [5:0] state

Ulazi u sistem:

input wire clk - omogućavanje takt impulsa

input wire echo - varijabla u koju dolazi impuls sa ultrasoničnog senzora

Izlazi iz sistema:

output wire gripper - hvataljka na robotskoj ruci

output wire upWrist - gornji zglob na robotskoj ruci

output wire downWrist - donji zglob na robotskoj ruci

output wire armPlatform - motor koji rotira robotsku ruku

output wire sprayingPlatform - motor koji rotira platofrm za dezinfekciju

output wire sprayingServoOne - motor koji vrši dezinfekciju

output wire sprayingServoTwo - motor koji vrši dezinfekciju

output wire trig - varijabla preko koje saljemo impuls u ultrasonični sensor

VERILOG KOD

Dodijeljivanje varijabli za robotsk ruku

```
assign gripper = gripper_bit;
assign upWrist = upWrist_bit;
assign downWrist = downWrist_bit;
assign armPlatform = armPlatform_bit;
assign sparyingPlatform = sparyingPlatform_bit;
assign sparyingServoOne = sparyingServoOne_bit;
assign sparyingServoTwo = sparyingServoTwo_bit;
```

Implementacija taska za slanje pulsa na motor hvataljke(gripper)

```
task gripperMove();
    input reg [20:0] ugao;
begin
    pulse_duration=ugao;
    gripper_bit <= (pf_reg < pulse_duration) ?1:0;
    pf_reg<=(pf_reg < 1000000)?(pf_reg+1):0;
end
endtask
```

- Na isti način su implementirani taskovi za slanje pulsa na ostale motore

VERILOG KOD

Task koji omogućava pokretanje motora sa različitim brzinama i željenim uglovima

```
task moveServo();
input reg [20:0] startUgao;
input reg [20:0] endUgao;
input reg [8:0] motor;
begin

    reg [20:0] startPulse = 25000 + (555*startUgao);
    reg [20:0] endPulse = 25000 + (555*endUgao);
    if(bool==0)
        begin
            ugao=startPulse;
            bool=1;
        end
    counter=counter+1;
    if(counter>counterLimit)
        begin

            case(motor)
                1:gripperMove(ugao);
                2:upWristMove(ugao);
                3:downWristMove(ugao);
                4:armPlatformMove(ugao);
                5:sparyingPlatformMove(ugao);
                6:sparyingServoOneMove(ugao);
                7:sparyingServoTwoMove(ugao);
            endcase
        end
    end
endtask

if(counter>32'd2000000)
begin
    counter = 0;
    //Smanjuj ugao za jedan stepen=555 puls
    if(startUgao>endUgao)
        begin
            ugao=ugao-555;
        end
    else
        begin
            ugao=ugao+555;
        end
    if(startUgao>endUgao)
        begin
            if(ugao<=endPulse)
                begin
                    ugao=endPulse;
                    state = state+1;
                    bool=0;
                end
            .
        end
    .

```

VERILOG KOD

Task koji omogućava delay izvršavanja programa

```
reg [31:0] counterDelay = 0;  
reg [31:0] counterDelayLimit = 100000000;  
task delay();  
begin  
    counterDelay = counterDelay + 1;  
  
    if(counterDelay>=counterDelayLimit)  
    begin  
        state = state + 1;  
        counterDelay = 0;  
    end  
end  
endtask
```

VERILOG KOD

Task koji implementira ultrasonični senzor

```
task ultraSonic();
begin
    counterUltraSonic<=counterUltraSonic+1'b1;
    if(counterUltraSonic<=500)
        begin

            trig_value <=1'b1;
        end
    else
        begin
            trig_value <= 1'b0;
            if(echo && counterUltraSonic<700000)
                begin
                    echo_counter<=echo_counter+1'b1;
                    if(echo_counter > 19'd294118 && echo_counter < 19'd300000)
                        state = state+1;
                    else
                        led_t <= 3'b000;
                end
            else
                if(echo_counter !== 19'd0)
                    begin
                        delay_counter <= delay_counter+1;
                        if(delay_counter >= 20'd100000000)
                            begin
                                counterUltraSonic <= 22'd0;
                                echo_counter <= 19'd0;
                                delay_counter <= 20'd0;
                            end
                        end
                    end
                end
            endtask
```

VERILOG KOD

Task koji implementira pokretanje motora za dezinfekciju

```
reg [31:0] counterSpray = 0;
task sprayTask();
begin

    counterSpray = counterSpray + 1;

    //Kada prode 50ms uđi u ovaj if uslov, u suprotnom uđi u
    else
        if(counterSpray>25000000 && counterSpray<=50000000)
            begin
                sparyingServoOneMove(75000);
                sparyingServoTwoMove(75000);
            end

        else if(counterSpray>50000000)
            begin
                state= state+1;
                counterSpray=0;
            end

    end
endtask
```

```
else
begin
sparyingServoOneMove(25000);
sparyingServoTwoMove(125000);
end
```

VERILOG KOD

Logika sistema

```
always @ (posedge clk)
begin

    case(state)
        1:ultraSonic();
        2:moveServo(90,30,1); //Gripper se otvara
        3:moveServo(90,150,2); //UpWrist se spusta
        4:moveServo(30,120,1); //Gripper se zatvara
        5:moveServo(150,90,2); //UpWrsit se podize
        6:moveServo(90,0,4); //Platforma se rotira
        7:moveServo(90,140,2); //upWrist se spusta
        8:moveServo(120,90,1); //Gripper se otvara
        9:moveServo(90,30,1); //Gripper se otvara
        10:moveServo(140,90,2); //upWrist se podize
        11:delay(); //Delay
        12:moveServo(90,0,5); //Okretanje platforme2
        13:sprayTask(); //Dezinfekcija
        14:sprayTask(); //Dezinfekcija
        15:delay();
        16:moveServo(0,90,5); //Okretanje platforme2
        17:sprayTask(); //Dezinfekcija
        18:sprayTask(); //Dezinfekcija
        19:delay();
        20:moveServo(90,180,5); //Okretanje platforme2
        21:sprayTask(); //Dezinfekcija
        22:sprayTask(); //Dezinfekcija
        23:delay();
        24:moveServo(180,90,5); //Okretanje platforme
        25:sprayTask(); //Dezinfekcija
        26:sprayTask(); //Dezinfekcija
        27:delay();
        28:moveServo(90,140,2); //Spustanje upWrist
        29:moveServo(30,120,1); //Zatvaranje grippera
        30:moveServo(140,90,2); //Podizanje upWrist
        31:moveServo(0,90,4); //Okretanje platforme1
        32:moveServo(90,140,2); //upWrist se spusta
        33:moveServo(120,90,1); //Otvori gripper
        34:moveServo(90,30,1); //Otvori gripper
        35:moveServo(150,90,2); //Podizanje upWrist
        36:moveServo(30,120,1); //Zatvaranje grippera
        37:moveServo(120,90,1); //Zatvaranje grippera

    endcase
end
```

ODZIV SISTEMA IZ REAL-TIME EKSPERIMENTA (matlab kod)

```
clear all

a=arduino;

delay=.01;
time = 0;
gripper = 0;
upWrist = 0;
armPlatform = 0;
sprayingPlatform=0;
sparyingServoOne=0;

count=0;

figure
plotGraph = plot(time,gripper);
xlabel('time');
ylabel('gripper pwm');

figure
plotGraph2 = plot(time,upWrist);
xlabel('time');
ylabel('upWrist pwm');
```

```
figure
plotGraph3 = plot(time,armPlatform);
xlabel('time');
ylabel('armPlatform pwm');

figure
plotGraph4 =
plot(time,sprayingPlatform);
xlabel('time');
ylabel('sprayingPlatform pwm');

figure
plotGraph5 =
plot(time,sparyingServoOne);
xlabel('time');
ylabel('sparyingServoOne pwm');

hold on
```

ODZIV SISTEMA IZ REAL-TIME EKSPERIMENTA (matlab kod)

```
tic

while ishandle(plotGraph)

    datGripper = readVoltage(a, 'A0');
    datupWrist = readVoltage(a, 'A1');
    datarmPlatform =
readVoltage(a, 'A2');
    datsprayingPlatform =
readVoltage(a, 'A3');

datsparyingServoOne=readVoltage(a, 'A4
');

    count = count + 1;
    time(count) = toc;

    gripper(count) = datGripper(1);
    upWrist(count) = datupWrist(1);
    armPlatform(count) =
datarmPlatform(1);
    sprayingPlatform(count) =
datsprayingPlatform(1);
    sparyingServoOne(count) =
datsparyingServoOne(1);

    set(plotGraph, 'XData',time, 'YData',grip
per);
    set(plotGraph2, 'XData',time, 'YData',upW
rist);

    set(plotGraph3, 'XData',time, 'YData',arm
Platform);

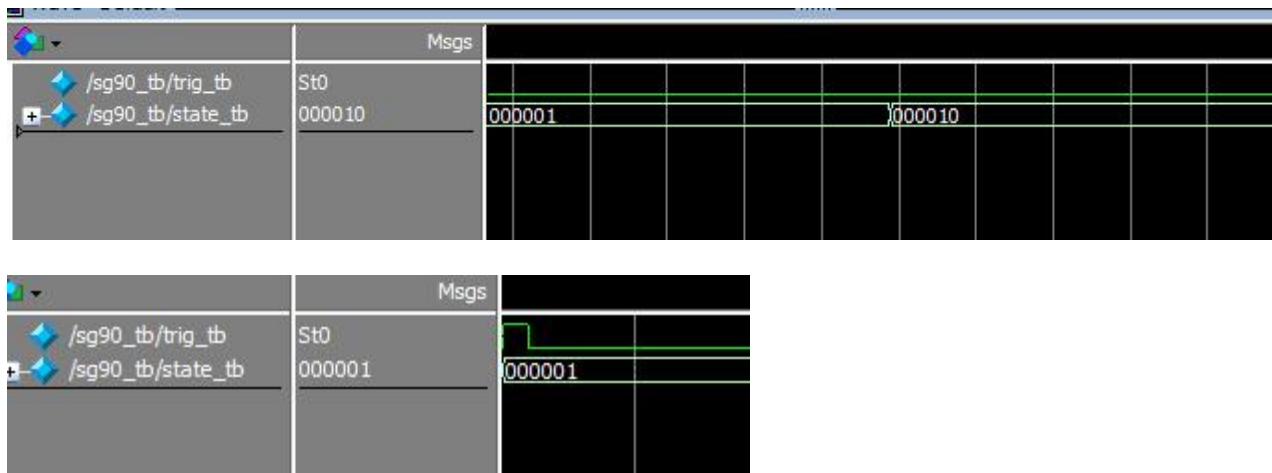
    set(plotGraph4, 'XData',time, 'YData',spr
ayingPlatform);

    set(plotGraph5, 'XData',time, 'YData',spa
ryngServoOne);

    pause(delay);
end
```

MODELSIM

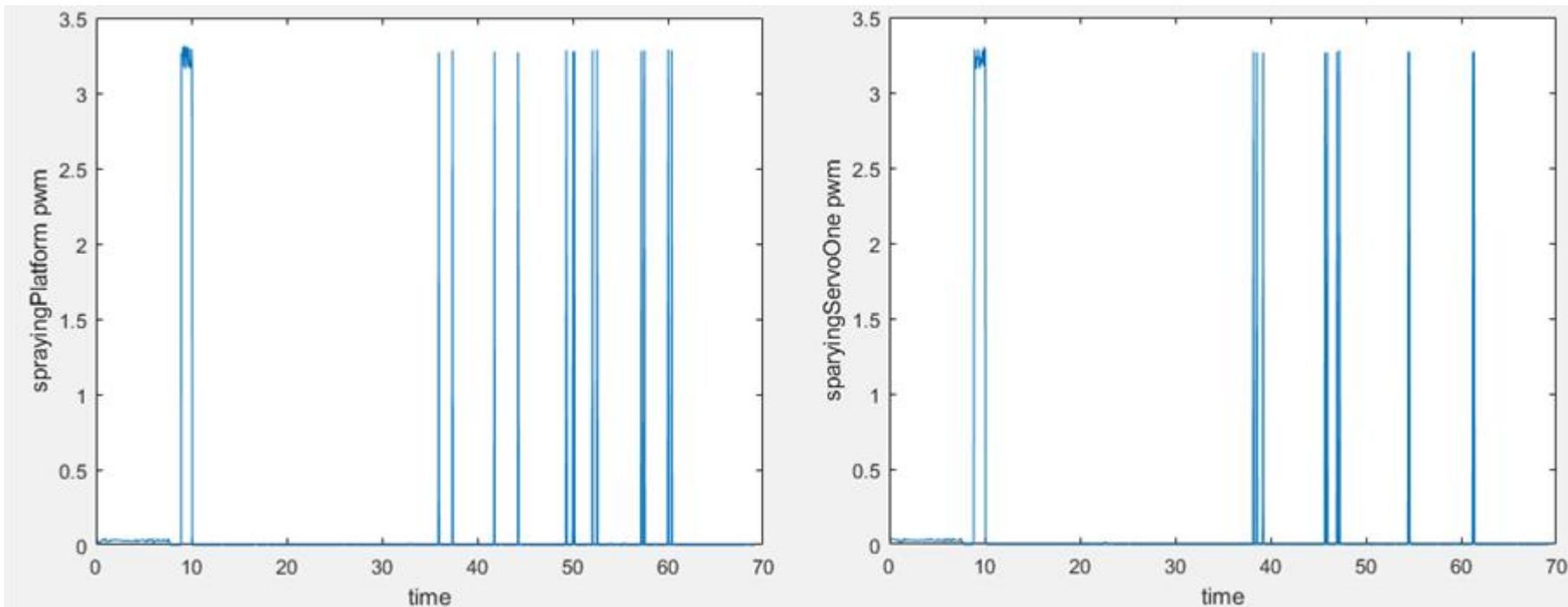
- Nakon kompajliranja koda za modul i testbench, simulirali smo te gledali promjenu našeg stanja



- Na prvoj slici možemo vidit okidanje triggera te vrijednost našeg stanja koje je 1.
- Nakon određenog vremena (2 sec) stanje mijenja vrijednost sa 1 na 2.
- Zbog korištenja delaya u kodu a samim tim zahtjevanja brzog kloka tako i postavljanjem rezolucije na nanosekunde nije bilo moguće simulirati više od promjene stanja sa 1 na 2.

ODZIV SISTEMA IZ REAL-TIME EKSPERIMENTA (Grafovi)

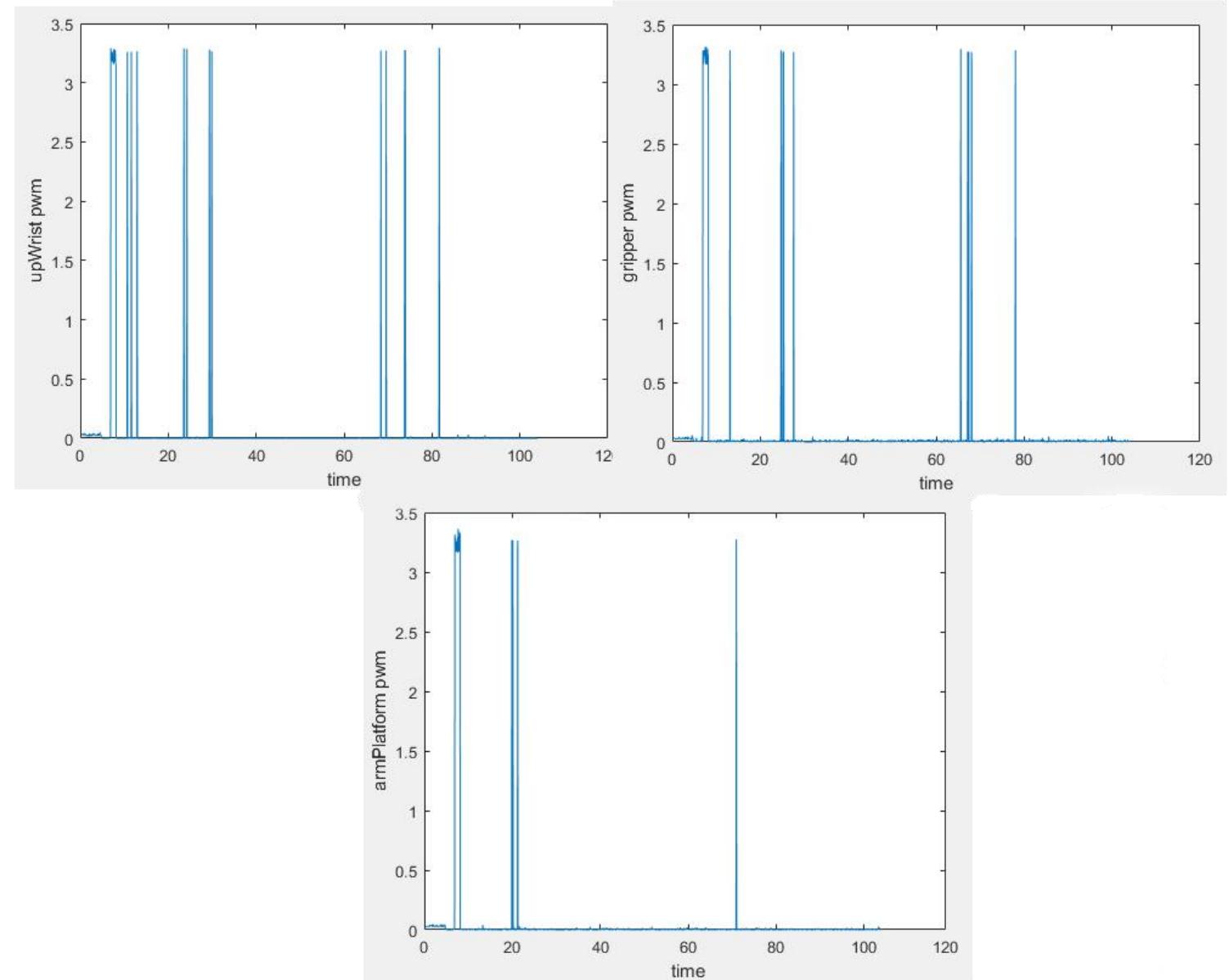
- Nakon pokretanja i spajanja na sistem imamo sljedeće grafove:



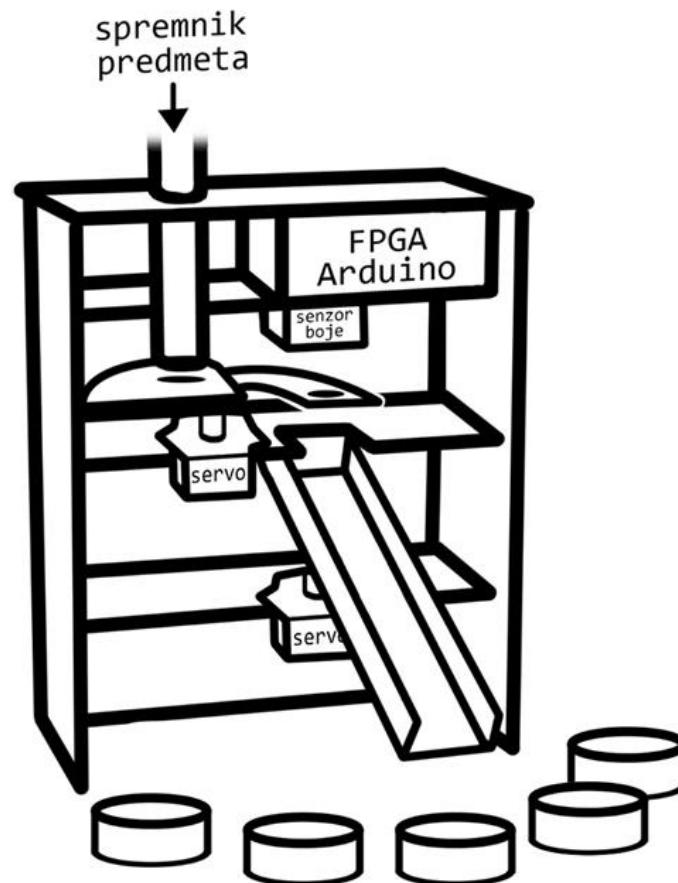
- Na grafovima su prikazane real-time vrijednosti voltaža sa servo motora te vidimo da se voltaža mijenja u zavisnosti da li je PWM signal aktivan. Uspoređujući vrijednosti sa grafova vidimo da se vrijednosti mijenaju kako se i mijenaju stanja sistema.

ODZIV SISTEMA IZ REAL-TIME EKSPERIMENTA (Grafovi)

Nakon pokretanja i spajanja na sistem imamo sljedeće grafove:



Primjer 4: Sistem za sortiranje predmeta po boji



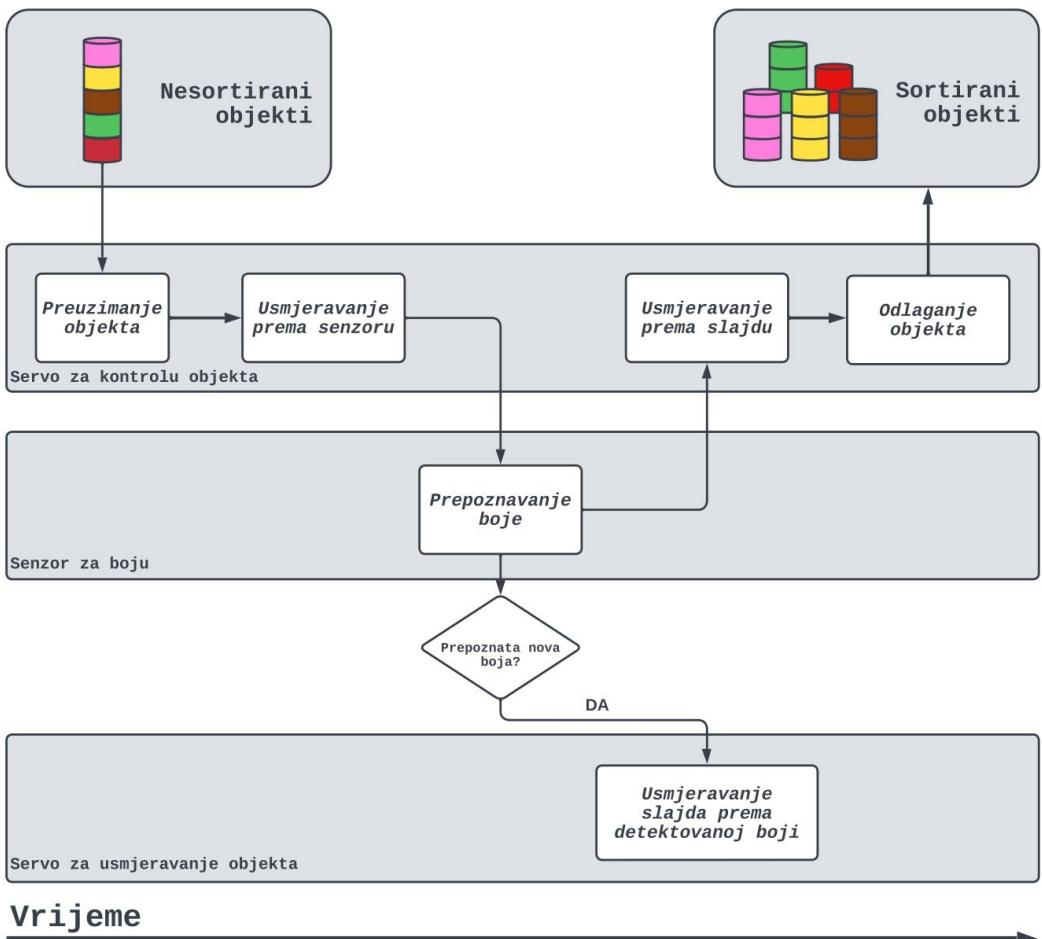
Zadaci

- I. Konstruisati dio industrijskog sistema za sortiranje proizvoda prema boji
- II. Povezati color senzor i motore na konstrukciju
- III. Isprogramirati dio logike za prikupljanje podataka sa color senzora na Arduino-u.
- IV. Korištenjem Veriloga razviti logiku za način sortiranja proizvoda na osnovu zadanih kriterija korištenjem FPGA
- V. Pratiti signale i aktivnost komponenti tokom procesa.
- VI. Prikazati rezultate procesa

Ciljevi

- I. Bliže upoznavanje sa mogućnostima FPGA-a, Veriloga
- II. Razvoj savremenih industrijskih aplikacija baziranih na senzorima i aktuatorima



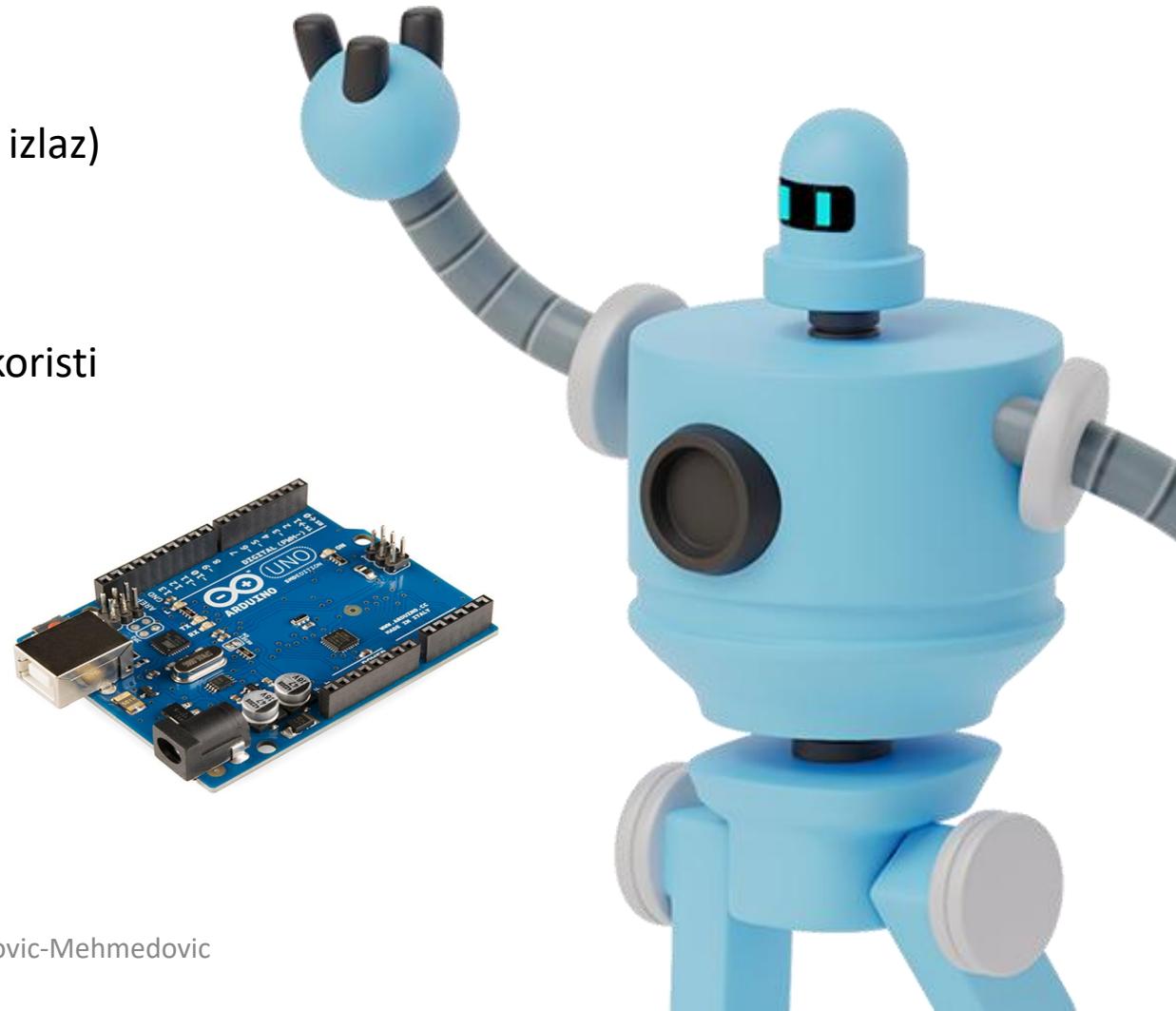


KOMPONENTE

ARDUINO UNO

Neke od osnovnih karakteristika su:

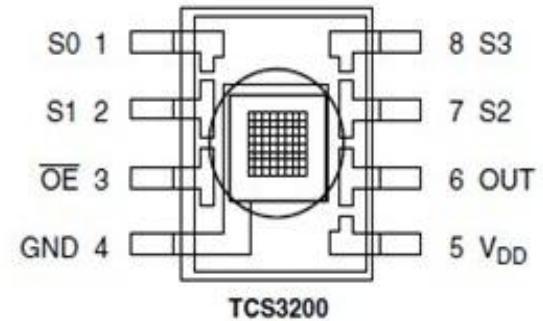
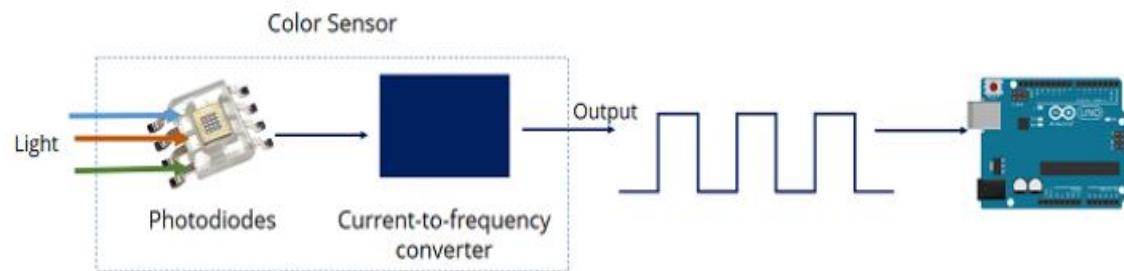
- Mikrokontroler: ATmega328P
- Radni napon: 5V
- Ulagani napon (preporučen): 7-12V
- Digitalni I/O pinovi: 14 (od kojih 6 omogućavaju PWM izlaz)
- Pinovi za analogni ulaz: 6
- DC struja po I/O pinu: 20 mA
- DC struja za 3.3V pin: 50 mA
- Flash memorija: 32 KB (ATmega328P) od čega 0.5 KB koristi bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Brzina clock-a: 16 MHz
- Dužina: 68.6 mm
- Širina: 58.4 mm
- Masa: 25 g



TCS3200 color senzor

- Napajanje: 2.7V do 5.5V
- Veličina: 28,4 x 28,4 mm (1,12 x 1,12")
- Interfejs: digitalni TTL
- Pretvaranje intenziteta svjetlosti u frekvenciju visoke rezolucije
- Programabilna boja i izlazna frekvencija pune skale
- Komunicira direktno sa mikrokontrolerom
- 16 fotodioda sa crvenim filterom – osetljive na crvenu talasnu dužinu
- 16 fotodioda sa zelenim filterom – osetljive na zelenu talasnu dužinu
- 16 fotodioda sa plavim filterom – osetljive na plavu talasnu dužinu
- 16 fotodioda bez filtera



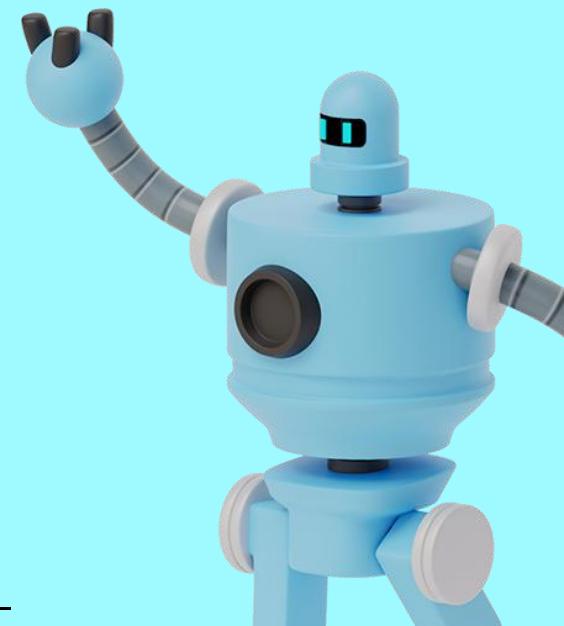


Pin Name	I/O	Description
GND(4)		Power supply ground
OE(3)	I	Enable for output frequency (active low)
OUT(6)	O	Output frequency
S0, S1 (1, 2)	I	Output frequency scaling selection inputs
S2, S3 (7, 8)	I	Photodiode type selection inputs
VDD (5)		Voltage supply

- Postavljanje S2 i S3 LOW i HIGH u različitim kombinacijama omogućava nam da odaberemo različite fotodiode.

Photodiode type	S2	S3
Red	LOW	LOW
Blue	LOW	HIGH
No filter (clear)	HIGH	LOW
Green	HIGH	HIGH

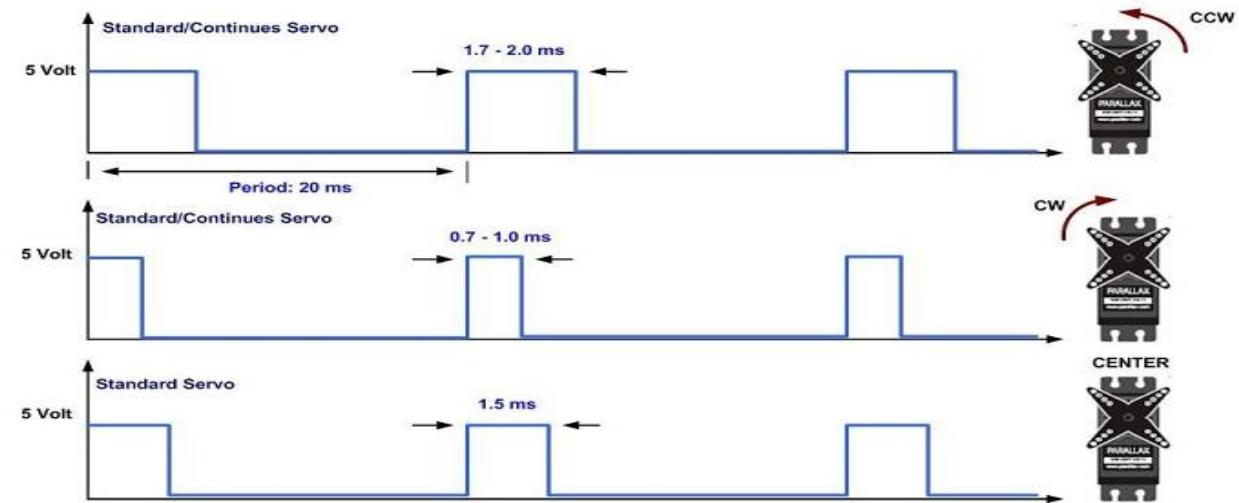
- Pinovi S0 i S1 se koriste za skaliranje izlazne frekvencije. Može se skalirati na sljedeće unaprijed postavljene vrijednosti: 100%, 20% ili 2%.



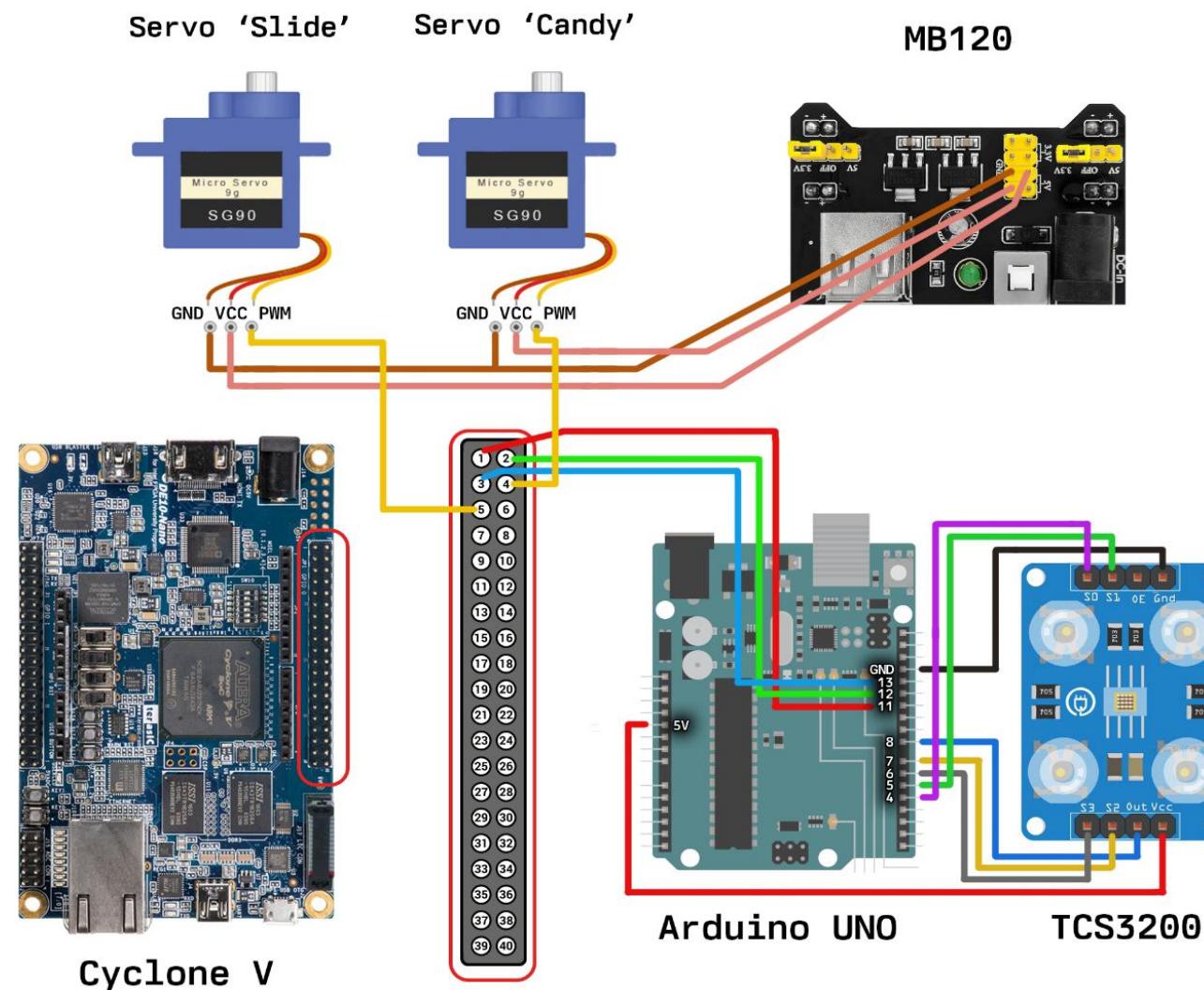
Output frequency scaling	S0	S1
Power down	L	L
2%	L	H
20%	H	L
100%	H	H

Servo motori

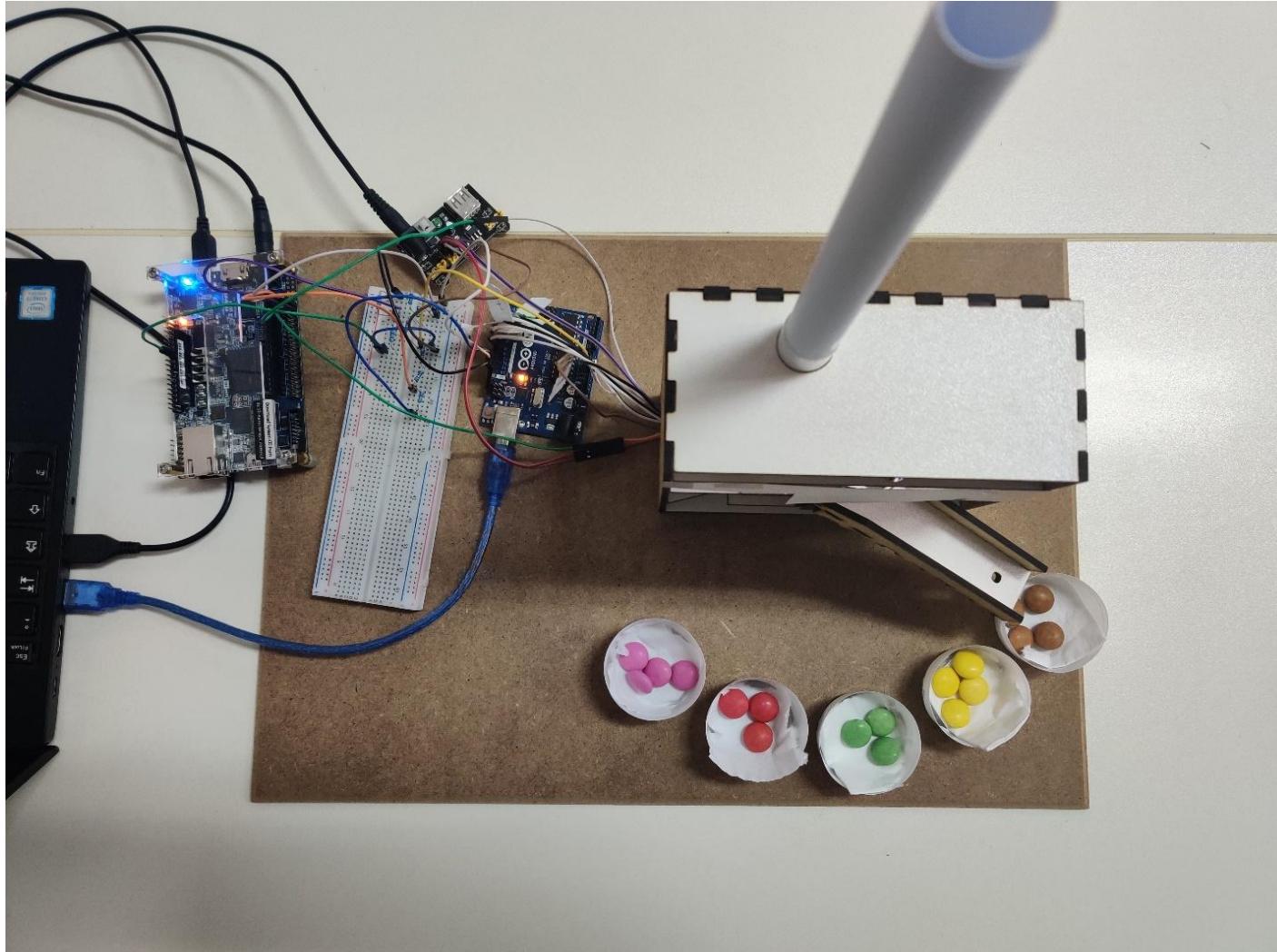
- Servo motori su rotacioni aktuatori koji se koriste u aplikacijama gdje je potrebno upravljanje ugaonom pozicijom, brzinom i ubrzanjem
 - Sastoje se od odgovarajućeg motora u kombinaciji sa senzorom koji daje informaciju o poziciji. Servo motori takođe sadrže i relativno sofisticirani kontroler, često odvojen modul, namijenjen za upotrebu sa servo motorima



Realizacija pomoću DE10-Nano platforme

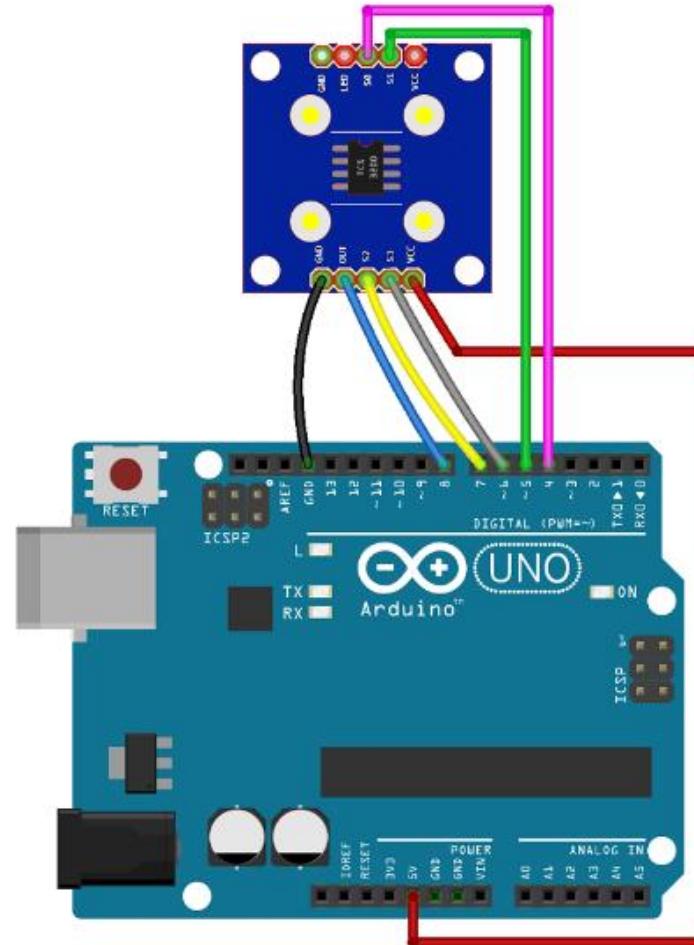


Realizacija pomoću FPGA i Arduino

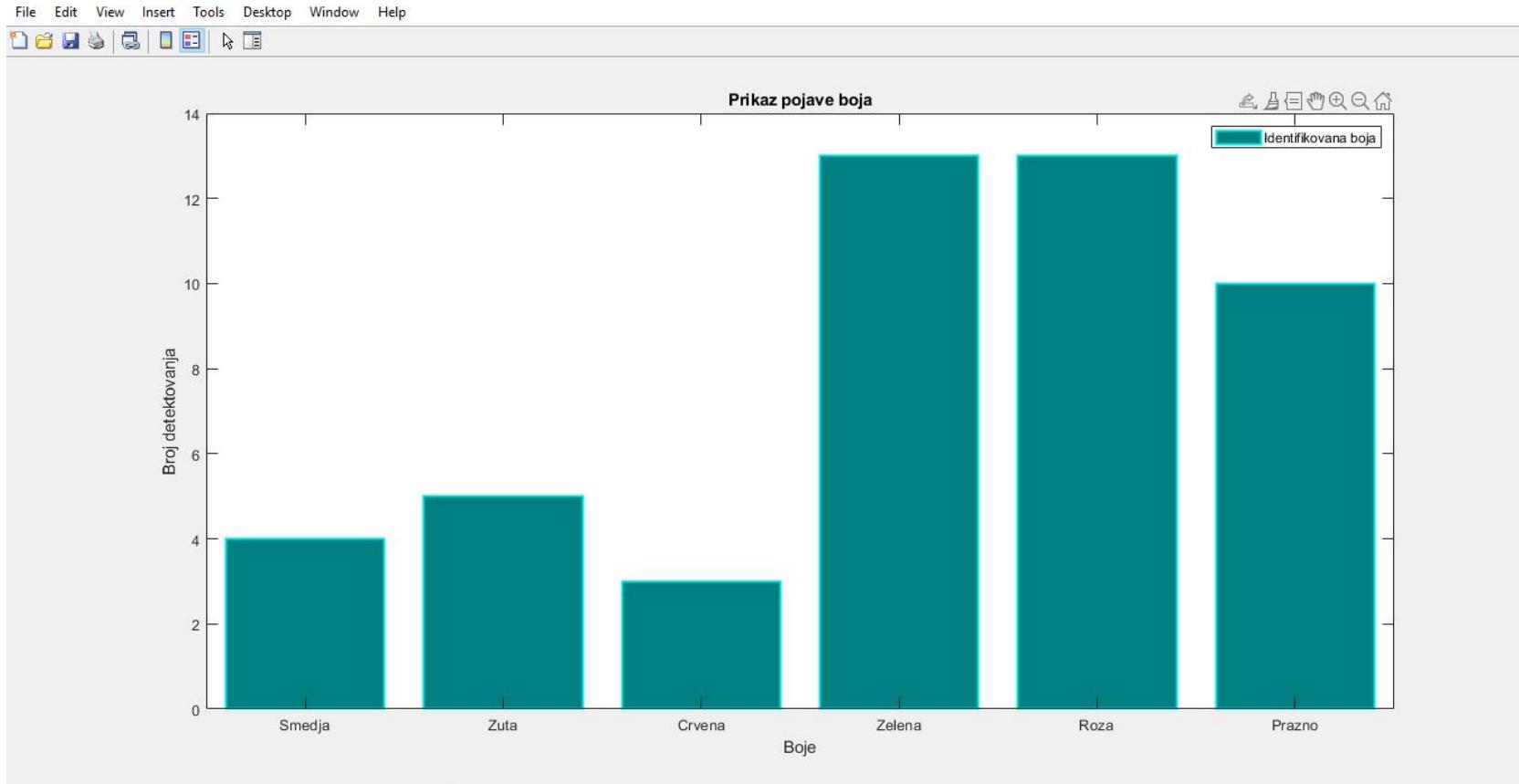


Veza između TCSP3200 i Arduina:

- S0: digitalni pin
- S1: digitalni pin 5
- VCC: 5V
- S3: digitalni pin 6
- S4: digitalni pin 7
- IZLAZ: digitalni pin 8

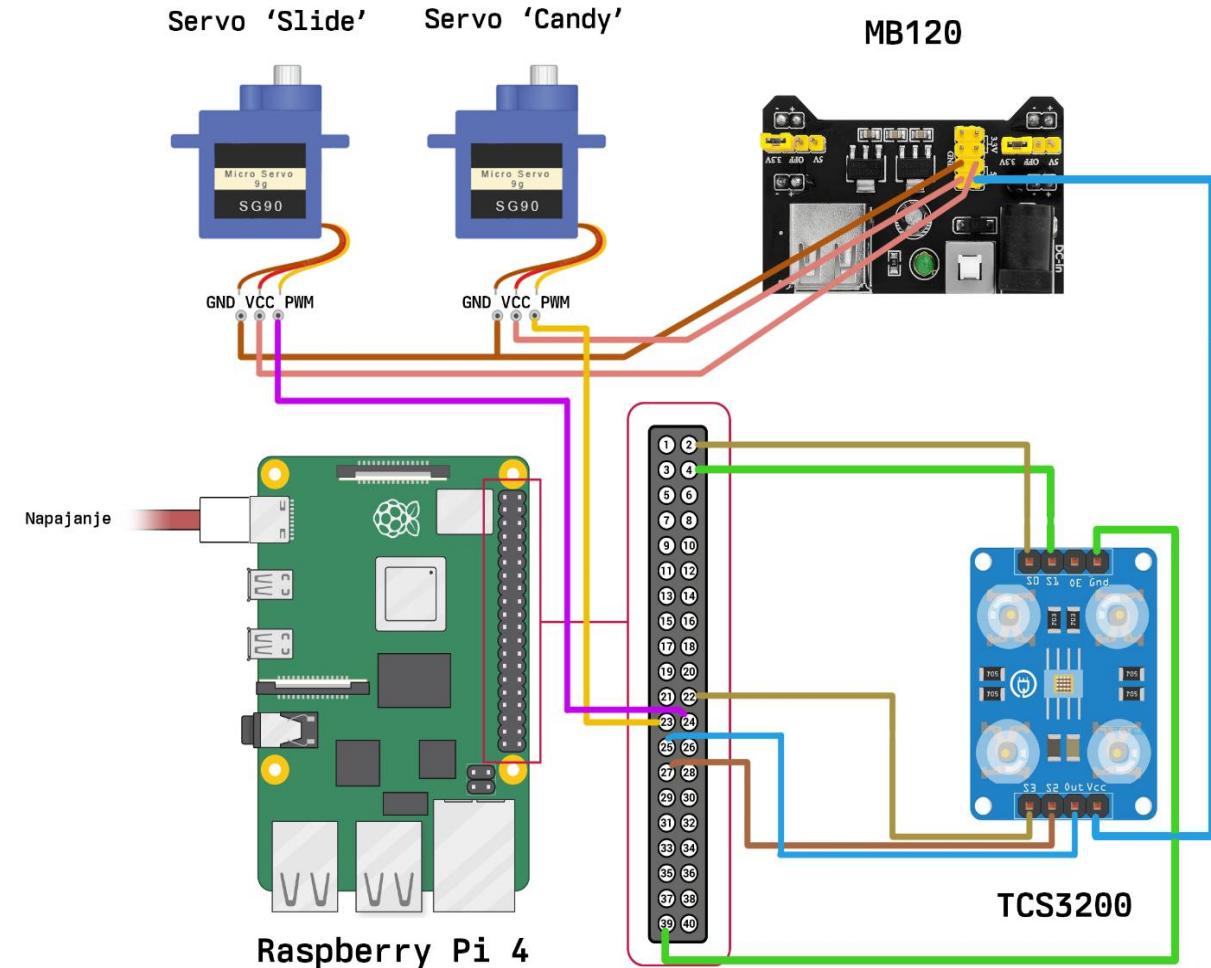


Vremenski dijagrami

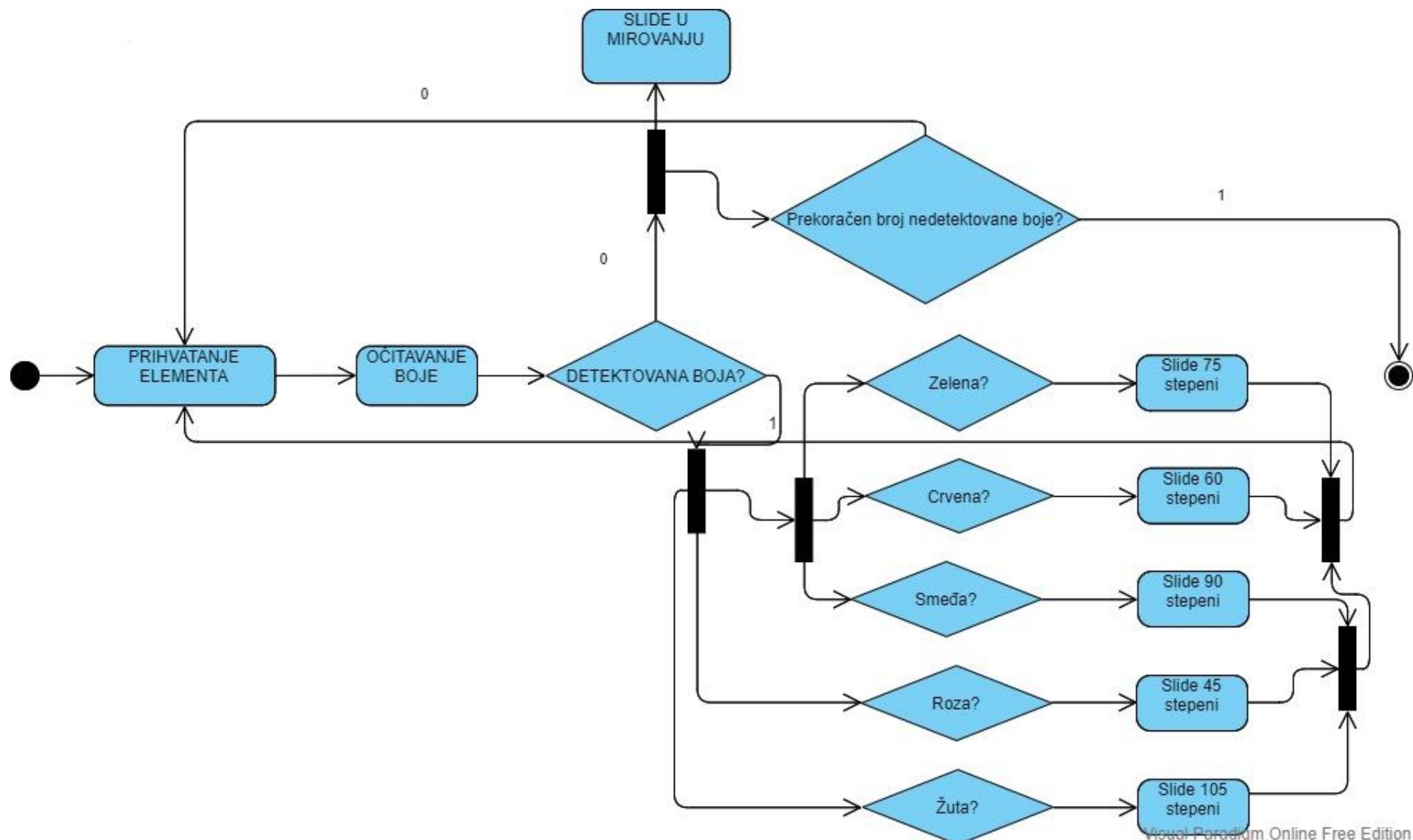


Realizacija projekta pomoću Raspberry Pi 4

Projekat je također realizovan pomoću Raspberry Pi 4 SoC-a, gdje su color senzor i servo motori direktno spojeni na Raspberry Pi.



•FSM



Visual Paradigm Online Free Edition

Realizacija pomoću Raspberry Pi

