



Ардуино радионица

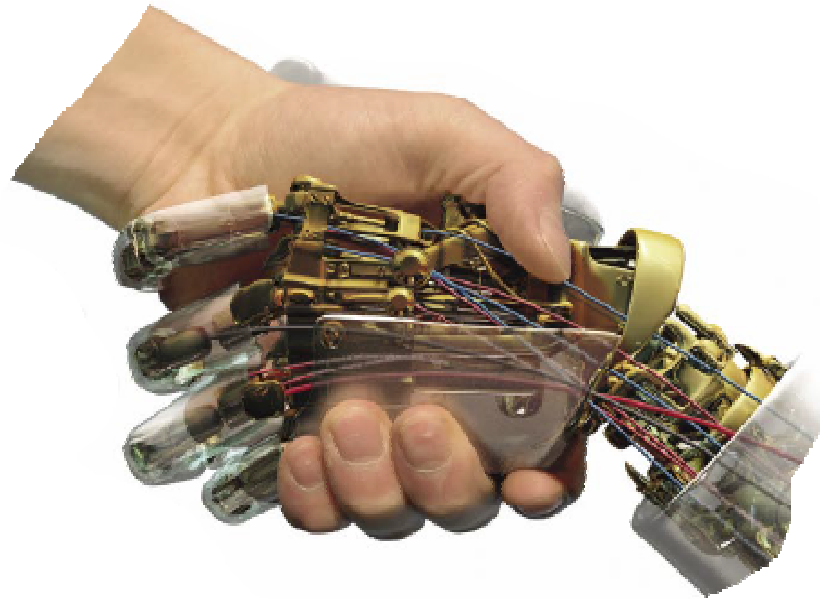
ARDUINO RADIONICA

Kako da napravite svog prvog mobilnog robota



- Roboti će se i u fizičkoj i elektronskoj formi integrisati u ljudsko društvo.
- Roboti će funkcionalno izražavati emocije i rasuđivanje.
- Napredni roboti-androidi biće slični ljudskim bićima i popuniti uloge u uslužnim delatnostima, komunalnim delatnostima i upravi.
- Robotska efikasnost i preciznost će transformisati oblasti proizvodnje, medicine, transporta, istraživanja i industrije i zameniti kvalifikovanu ljudsku radnu snagu.
- Robotska industrija će postati biznis vredan više milijardi dolara i tako omogućiti mnoge nove karijere i poslovne mogućnosti.
- Ljudska bića će usvojiti robotska humanoidna poboljšanja da postignu nadljudske mogućnosti.
- Kiborzi, delimično ljudi, delimično roboti, razviće veštine superiornije od prirodnih ljudi i tako odgovoriti visokim zahtevima specijalizovanih radnih mesta.
- Mi ćemo se suočiti sa ozbiljnim etičkim, bezbednosnim i socijalnim pitanjima zbog naših robotskih kreacija.
- Roboti će obezbediti udobnost, sigurnost i produktivnost koja će biti od koristi ljudskoj zajednici i značajno uticati na stil življenja.
- Androidi će ostvariti osnovni nivo samosvesti.



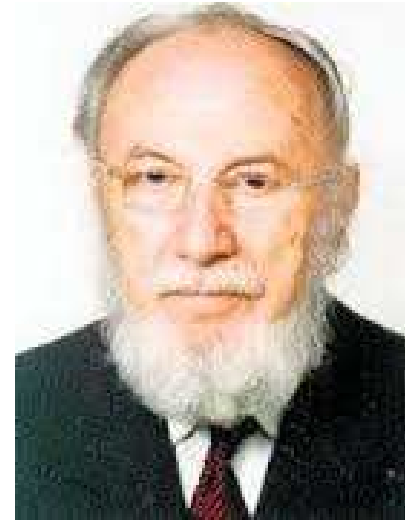


- Definicija robota zasnovana na standardu **ISO 8373** (slobodan prevod):
"automatski upravljani, reprogramabilni, multifunkcionalni manipulator programiran za pomeranje u tri ili više osa, koji može biti fiksiran za neki oslonac ili pokretan (mobilan) u domenu automatskih industrijskih primena"

- Ideje koje predstavljaju začetke automatizovane mašine možemo pronaći ako zavirimo daleko u prošlost prateći mitove o mehaničkim stvorenjima, zamišljene koncepte i pronalaskе.
- Jedan od koncepta potiče još iz 4. veka pre n.e. kada je grčki matematičar *Archytas od Tarentuma* osmislio mehaničku pticu, goluba koji je trebalo da pokreće vodena para.
- Značajan pronalazač tog doba jeste *Ctesibius od Alexandrie*, fizičar i pronalazač iz Egipta, koji je oko 250. god. pre n.e. napravio vodeni sat - *clepsydra*.
- *Heron od Alexandrie* (10-70. god. n.e.), grčki matematičar i inženjer, imao je brojne izume među mašinama od kojih se za jednu tvrdilo da je navodno mogla da imitira ljudski glas.



Rajko Tomović

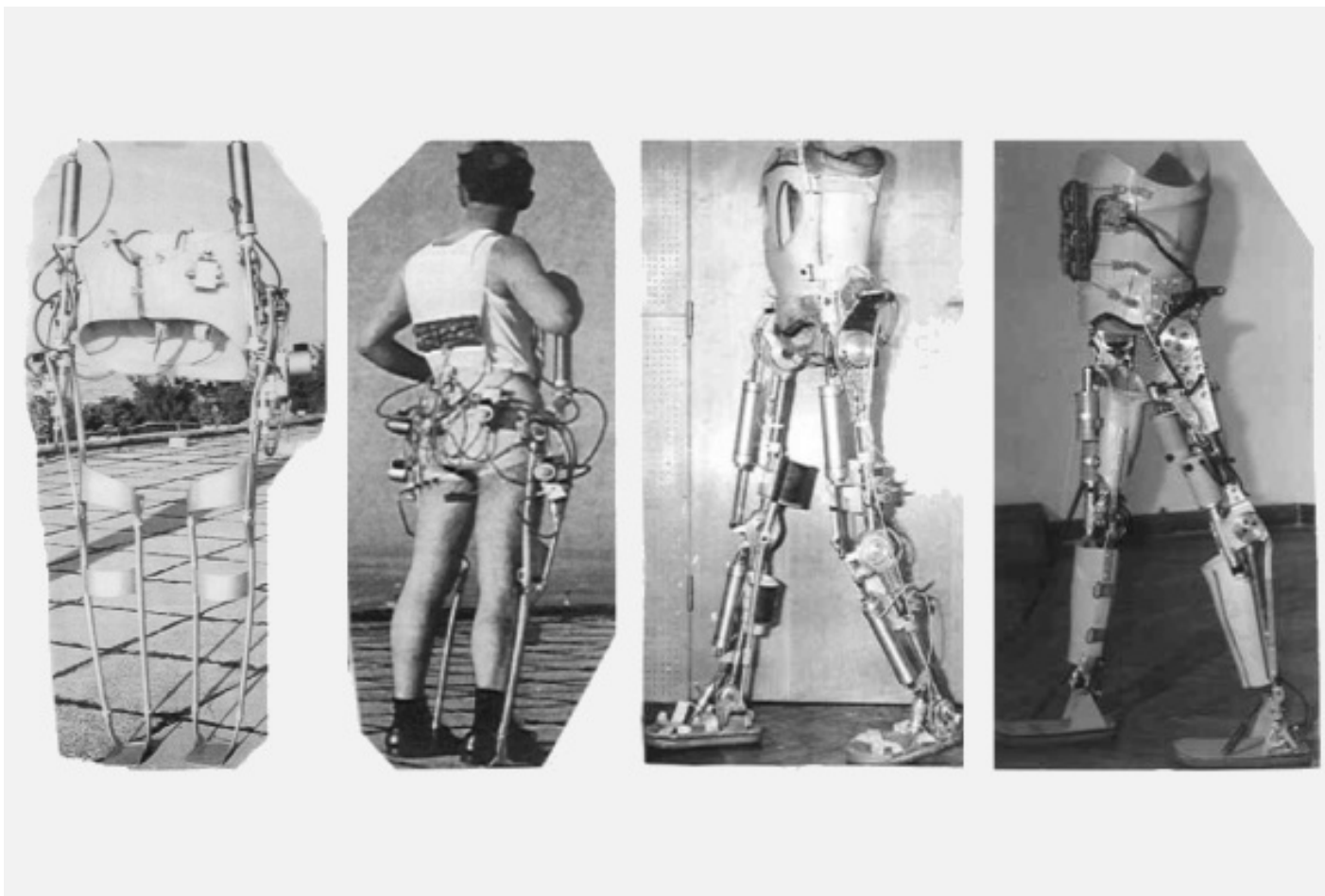


Miomir Vukobratović

- Prve pokušaje pravljenja robota pokrenuo je **profesor dr Rajko Tomović 1963. godine**, gradeći takozvanu **beogradsku ruku/šaku**.
- Najava intenzivnih istraživanja na Institutu "Mihajlo Pupin" došla je s prvim radovima **dr Miomira Vukobratovića** iz oblasti matematičkog modelovanja **antropomorfnih** aktivnih mehanizama.

- Ubrzo, 1971. godine, istraživanja su se okrenula **razvoju prvog aktivnog egzoskeleta**. Ova mašina je služila kao osnovni pokretač za paraplegičare tako što je nadoknađivala 50 odsto sile potrebne za pokretanje tela.
- **Upotrebljivi prototipi egzoskeleta napravljeni su 1974.** kao kompletni uređaji snabdeveni elektropneumatskim i elektromehaničkim motorima. Sledeće godine, istraživanja na Institutu se vode u laboratoriji za biodinamiku i okreću se industrijskoj robotici.
- U **Institutu Mihailo Pupin je 1978. napravljen u svetu prvi industrijski robot sa antropomorfnom konfiguracijom**, koji je potom počeo da se upotrebljava u jugoslovenskoj industriji. Sledećih godina su razvijeni novi modeli aktivnih skeleta kao i cela **ortopedska ruka za distrofičare**.
- Tokom osamdesetih, a posebno devedesetih, istraživanja robota u Beogradu gube na intenzitetu.

- Pionirski **doprinos Beogradske škole robotike** pokrenuo je istraživanja ortopedskih pomagala i uporedo, **postavio robote na noge**

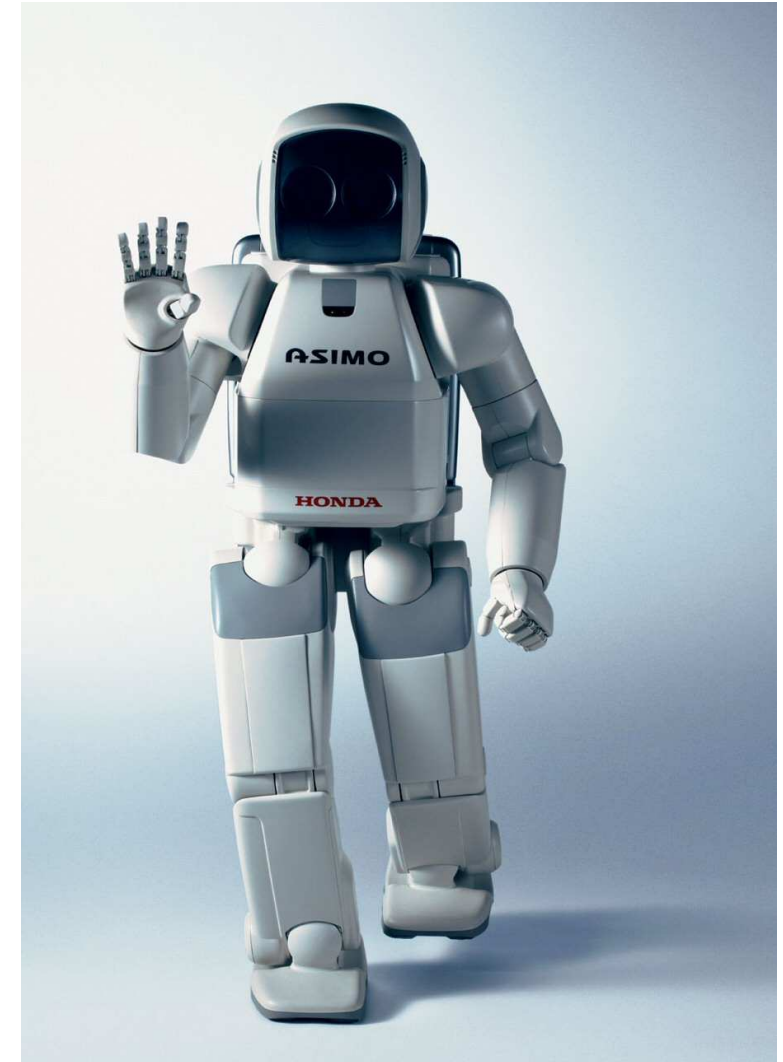


- Najvažniji korak u razvoju **Beogradske škole robotike**

bio je Vukobratovićev rad o

tački nula momenta, objavljen još **1969. godine.**

Reč je o teorijskom modelu kretanja humanoidnog robota na kome se i danas zasniva kretanje svih vrsta androida. Pre nego što je ovaj model kretanja počeo da se primenjuje kod humanoida, **roboti su imali veliki problem sa održavanjem ravnoteže u pokretu.** Princip koji je osmislio profesor **Vukobratović** i danas se, gotovo pola veka kasnije, primenjuje kod mnogih čovekolikih robota, uključujući i **ASIMO** robota.

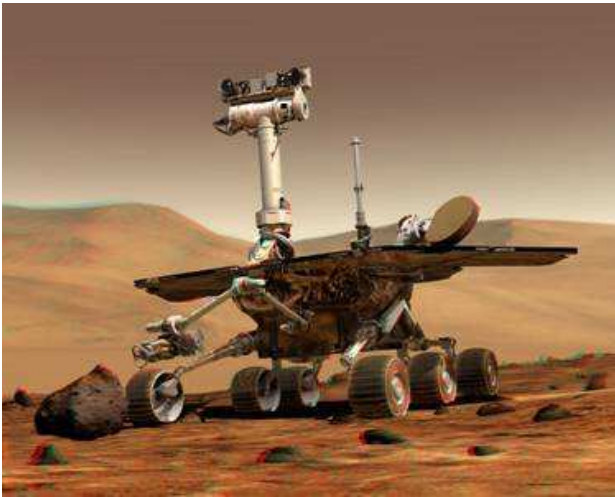


Kretanje robota

- Premda su izgled i sposobnosti robota različite, **svi roboti su** uglavnom **mehaničke strukture** koje predstavljaju **kinematički lanac**.
- Lanac je formiran od **karika** (kao kosti kod čoveka) i **motora** (mišića) koji združeno funkcionišu izvodeći različite pokrete sa više stepena slobode.
- Ono što zapravo **pokreće** robota su
 - **DC motori** (motori jednosmerne struje),
 - **step (koračni) motori** sa kontrolisanim uglom pomeranja,
 - **piezo (ultrasonični) motori** fundamentalno drugačijeg principa rada, kao i
 - **hidraulički aktuatori** velikog izlaznog momenta.

Mobilni robot

- **Mobilni roboti** su klasa robota koji mogu **autonomno** da se kreću, menjajući svoju poziciju i/ili orijentaciju u prostoru i obavljaajući pri tome neke specifične zadatke.
- Njihova pozicija u prostoru ograničena je izgledom radnog prostora (veličinom i pozicijom prepreka), i autonomnim radijusom kretanja robota (potreba za energijom i komunikacijom).

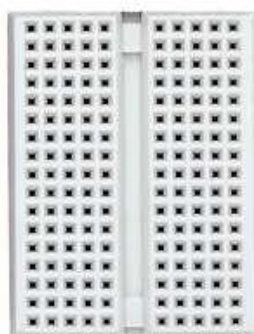
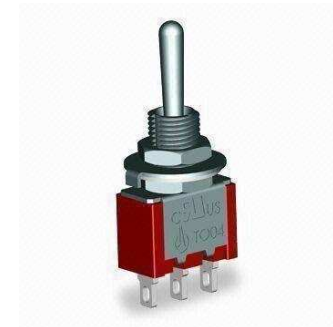


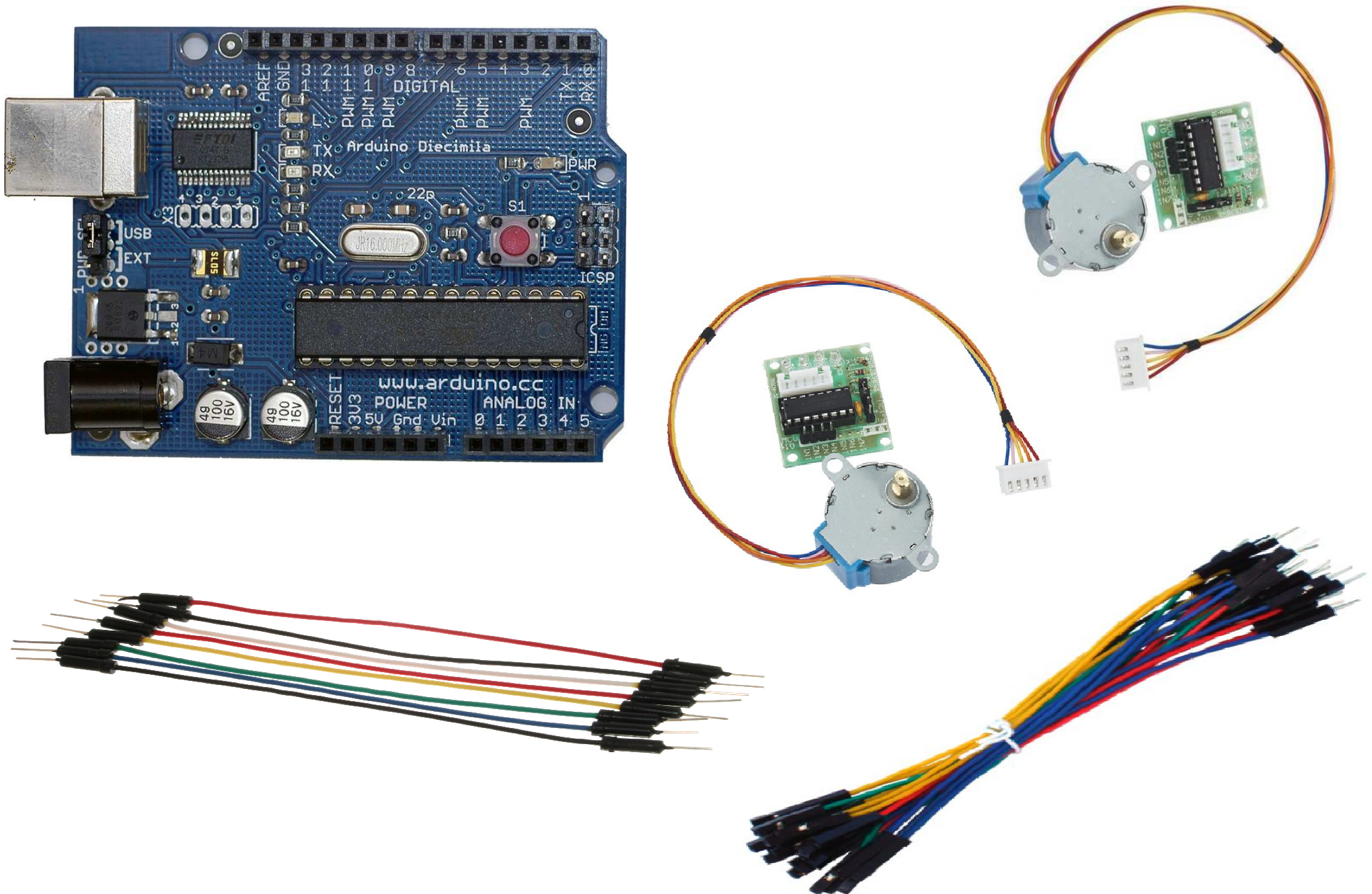
- **Mobilni robot**, koji je predmet ovog izlaganja, realizovan je **sa pogonom na dva točka u konfiguraciji sa tzv. diferencijalnim pogonom (*differential drive*)**. Ose obrtanja pogonskih točkova su fiksne i podudaraju se, a svaki točak ima nezavisan pogon.
- Kretanje ovog mobilnog robota je zasnovano na konceptu trenutnog pola rotacije. Odnosno, **u zavisnosti od međusobnog odnosa brzine kretanja pogonskih točkova, biće izveden određeni način kretanja.**
- Da bi se postigla statička stabilnost, pored dva pogonska točka, dodat je jedan pomoćni točak, tzv. caster.
- Za platformu (šasiju) upotrebiće se plastična kutija za vijke.
- Pogonski motori – 2 koračna motora 28BIY-48, 5V sa drajverom ULN2003
- Dva gumena točka
- Mikrokontrolerska pločica Arduino
- 4 baterije od 1.5 V za napajanje
- 1 pregibni prekidač za uključivanje napajanja
- IR dioda TSOP34838 - prijemnik
- Daljinski upravljač za televizor LG - komandni emiter

- Robotom se upravlja korišćenjem komercijalnog LG daljinskog upravljača za TV. Komande prima **IR** prijemnik koji je priključen na Arduino mikrokontrolersku pločicu.



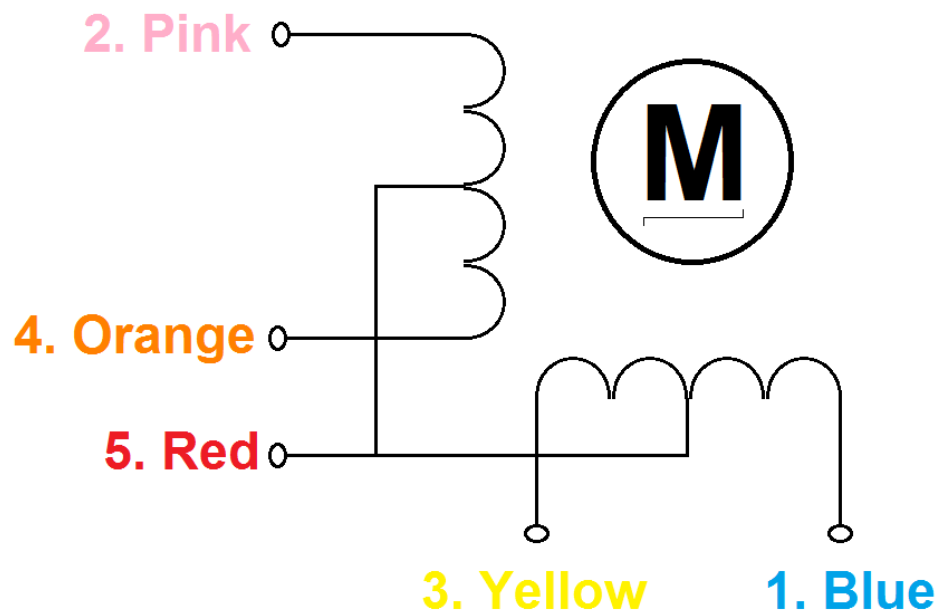
... Delovi za mobilnog robota





- **Koračni motor** ([stepping motor](#)) je jednosmerni (DC) motor koji svoju punu rotaciju može ostvariti u nekoliko koraka i može se zaustaviti na određenom koraku bez ijednog povratnog senzora (feedback sensor). Za razliku od običnih jednosmernih motora koji se konstantno okreću kada na krajeve namotaja (polove) dovedemo jednosmerni napon, na polove koračnog motora dovodimo povorke (nizove) pravougaonih impulsa.
- Ovi motori se dele u dve grupe: **unipolarne** i **bipolarne** step motore. Kod oba tipa motora [jedan dovedeni impuls uzrokuje pomeranje za jedan korak](#) čija veličina u stepenima zavisi od vrste primenjenog step motora.
- Ovaj motor se sastoji od rotora proizvedenog od nemagnetnog čelika i statora koji se sastoj od većeg broja polova. Motor se upravlja tako da se na određeni par polova dovodi napajanje koji privlači polarizovane delove rotora.
- Step motori mogu da rade u [tri različita režima](#): monofazni, dvofazni i [režim polukoraka](#).

Koračni unipolarni motor 28BIY-48 5V



Motor



SWITCHING SEQUENCE

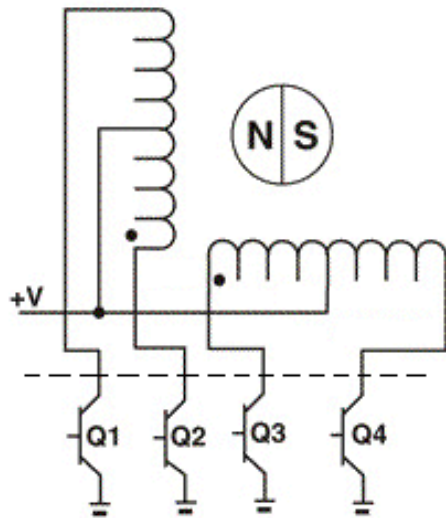
Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 Orange	-	-						-
3 Yellow		-	-	-				
2 Pink				-	-	-		
1 Blue						-	-	-

Drayver
ULN2003

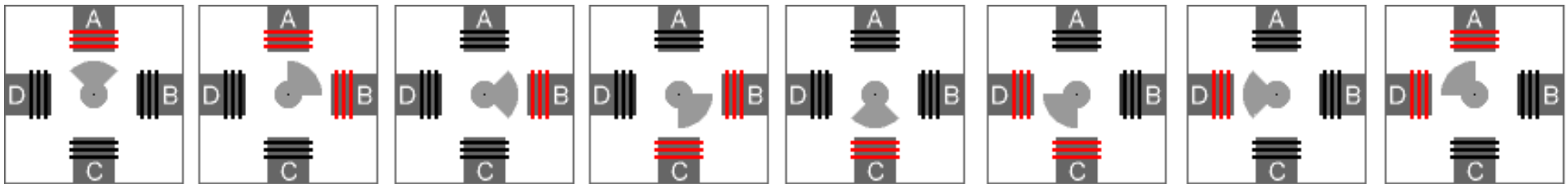
Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz,No load,10cm)
Model	28BYJ-48 – 5V

- Unipolarni koračni motor ima dva namotaja po fazi, po jedan za svaki smer magnetnog polja. Imajte u vidu da reč “uni” znači da je **samo jedan smer struje potreban za ovaj tip motora.**

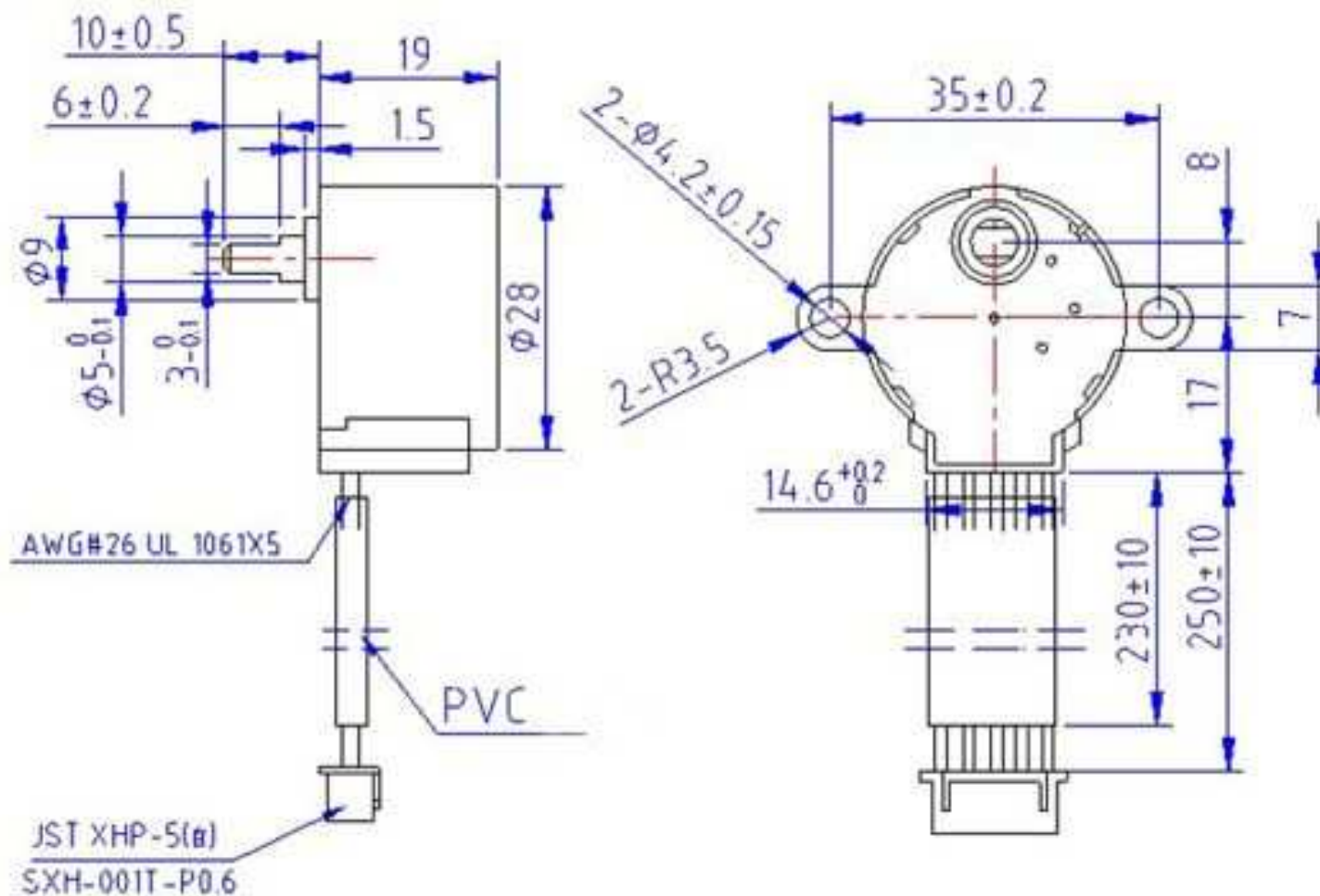
UNIPOLAR



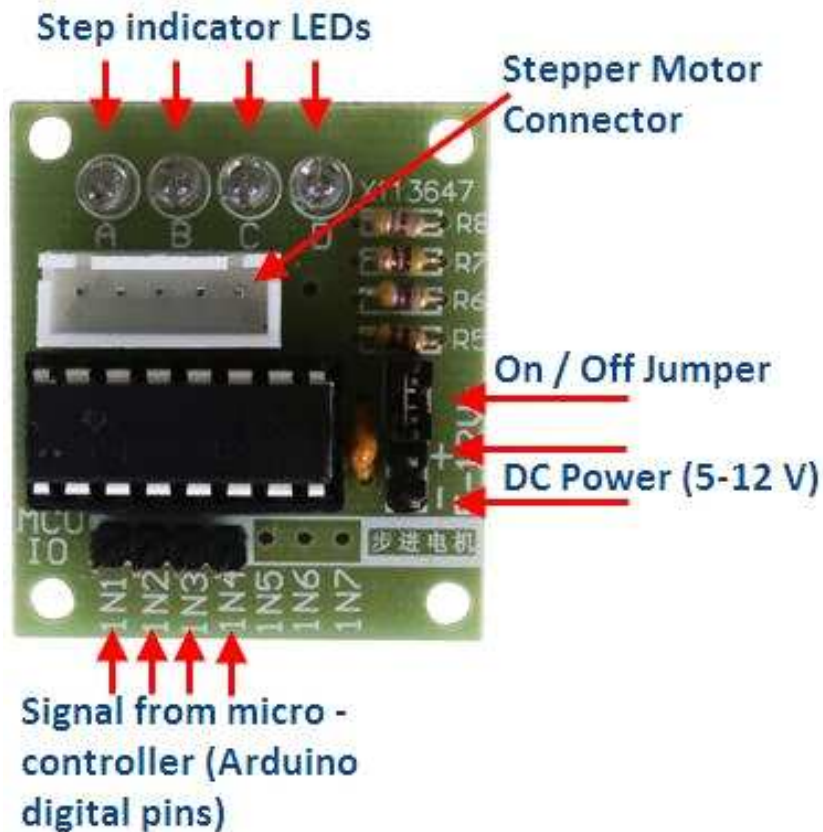
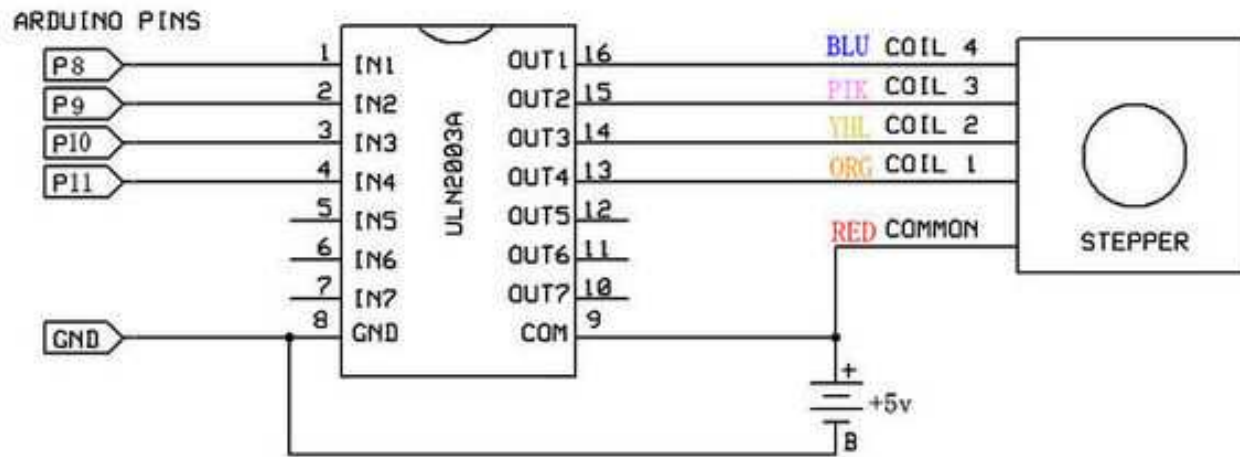
Kod tzv. **polukoračnog režima** (half stepping) motora, **naizmenično se vrši smenjivanje broja faza pod naponom od dve na jednu dok se ne završi ceo ciklus.** Ovime se povećava ugaona rezolucija, ali motor ima manji obrtni momenat (70%) na poziciji pola koraka (kada je samo jedna faza pod naponom). Ovo se može prevazići povećanjem struje u aktivnom namotaju. Polukoračnim režimom povećava se tačnost izlaza.



... podaci za montažu koračnog motora



... Priključenje na drajver ULN2003



Kolo **ULN2003A** u sebi sadrži tranzistore u tzv. Darlingtonovom spoju i imaju ulogu da se na njegovim izlazima direktno upravlja ili vrši pobuda. Ukupno ima osam izlaza.

- Uloga točka je da prenese mehaničku snagu motora na podlogu.
- Robotova funkcija, očekivano stanje podloge, kao i karakteristike motora moraju se uzeti u obzir prilikom izbora najpogodnijeg točka.



Većina točkova se sastoji iz dva dela.

Prvi, gumeni deo (guma/pneumatik) kojim se točak oslanja na površinu.

Drugi je snažno jezgro (felna) koje prenosi snagu motora i održava oblik točka

- **Vazduhom punjene gume** imaju težnju da se prilagođavaju površinama, čime se obezbeđuje maksimalan kontakt za vuču. Takođe, sabijaju se prilikom kontakta sa malim objektima i česticama, za razliku od punih guma koje odskaču ili se izdižu kod prepreka tog tipa. Vazduh deluje kao amortizer, čime redukuje habanje motora i ležajeva.
- **Pune gume** dobro nose težinu ali ne i stisak, niti upijaju udarce kao pneumatske gume. Međutim, pošto ne mogu da se stisnu, one dodaju manje greške preciznom kretanju.



- **Oblik gume** značajno utiče na vuču i otpornost kretanju.
- Zaobljen poprečni presek gume, sličan obliku koji ima guma na biciklu, ima malu dodirnu površinu sa podlogom tokom kretanja. Zbog toga postoji mali otpor kretanju i ove gume se lako okreću i imaju mali otpor kotrljanja. **Otpor kotrljanja** je **iznos trenja koji se generiše spram podloge i on određuje koliko lako se guma kotrlja napred ili nazad**. Zbog ograničenog kontakta sa podlogom, gume sa zaobljenim poprečnim presekom su energetske efikasnije i zato su dobar izbor za dugotrajnu eksploataciju.
- Kod **balon guma** centralni deo malo štrči u odnosu na obodne strane. Dobro apsorbiraju udarce i dobro podnose neravne površine. Preporučuju se za kretanje **po grubim površinama**.

- Postoje gume koje su oblika koji je između prethodno navedenih, tj. centralni deo je blago spljošten a prema krajevima zaobljen. Pružaju kompromisne karakteristike, tj. nisu najbolje niti najgore u bilo kojoj kategoriji.
- **Gume sa ravnom površinom** obezbeđuju najbolju vuču na ravnim površinama. Dobar su izbor kada se želi dobro iskorišćenje motora na putu, mada to ima svoju cenu. Ove gume imaju najveći otpor kotrljanja. Biraju se za situacije kod kojih se želi **maksimalno guranje, vuča ili ubrzavanje na glatkim površinama**.
- **Šire gume** obično **obezbeđuju veću stabilnost i vuču**, bez obzira na oblik poprečnog preseka. Takođe, manje su šanse da se šire gume zaglave u paralelnim žljebovima na površinama preko kojih se kreću. Sa druge strane, **uže gume imaju manji otpor kotrljanja i mogu mnogo lakše da se okreću**. One su lakše i zauzimaju manje prostora kod konstrukcije robota.



Sa leva na desno: zaobljena, balon, hibridna, ravna

- **Stil (tip) gazišta** na gumi treba izabrati tako da odgovara površini i vremenskim uslovima pod kojima se očekuje da će raditi.
- **Glatke gume obezbeđuju maksimalan kontakt u suvim uslovima i na ravnim površinama.** Iz tih razloga takve gume koriste automobili formule-1. Međutim, u vlažnim uslovima, na šljunkovitim i neravnim podlogama, glatke gume teže da se izdižu iznad tih prepreka usled čega se smanjuje trenje a time i vuča. Zato vozači formule-1 menjaju gume kada počne da pada kiša.
- **Ožlebljene gume** su slične glatkim, ali sa trakama i šarama koje su utisnute u gazište. One obezbeđuju pristojno trenje (vuču) na ravnim površinama. Na zaprašenim i vlažnim površinama, žlebovi usmeravaju čestice tako da ne budu u kontaktnim tačkama i oni su ti koji omogućavaju gumama da budu nešto fleksibilnije. Zbog te **izbalansirane performanse u suvim i vlažnim vremenskim uslovima, ožlebljene gume ili gume sa šarama se nalaze na skoro svim automobilima.**

- **Knobby gume** imaju neravnine koje izbijaju iz površine pneumatika. Ove gume su izvrsne u off road ili vlažnim terenima. Prave veću buku i imaju grublju vožnju (klačenje), ali isplati se jer su **stabilne na težim terenima i u blatu**. Nisu pogodne za precizna kretanja, kretanja koja se izračunavaju ili guranje na ravnim površinama.



Sa leva na desno: glatka, ožlebljena, knobby

- Verovali ili ne, **prečnik** ("visina") pneumatika (gume) je **verovatno najvažniji parametar robotove gume**. Očigledan efekat je uticaj na visinu robota. Međutim, postoji mnogo značajniji efekat.
- **Prečnik pneumatika radikalno utiče na obrtni momenat i brzinu**. Što je veći prečnik, to je veća dobijena brzina. Što je manji prečnik, veća je sila uvijanja (guranje, vučenje).
- **Ako se vaš robot kreće suviše brzo, zamenite postojeće gume gumama manjeg prečnika.**

Ako robot **ne može da se kreće zbog velike mase** ili slabog motora, opet zamenite postojeće gume gumama **manjeg prečnika**. Nasuprot tome,

ako se robot kreće suviše sporo ili teži da proklizava kada počinje da se kreće, zamenite postojeće gume gumama **većeg prečnika**.

Slika ilustruje **odnose** između **poluprečnika**, **prečnika** i **obima** točka.



r



$d = 2 * r$



O

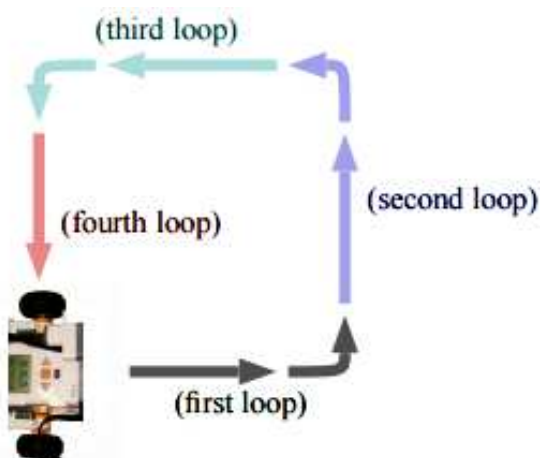
$O = 2 * r * \pi$

Upravljanje robotom pomoću “daljinca”

- Robot sa diferencijalnim pogonom – 2 koračna motora
- **Upravljanje** – komercijalni daljinski upravljač za TV
- Dva režima rada: ručno i automatski
- **Ručni režim:** napred, nazad, levo, desno
- **Automatski režim:** kretanje po kvadratnoj putanji



Ručno



Automatski

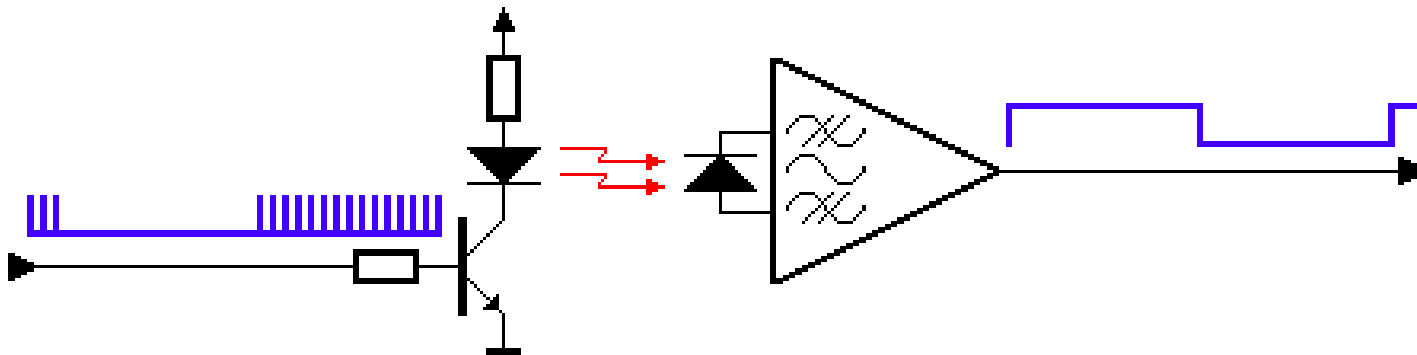


Kako radi “daljinac”?

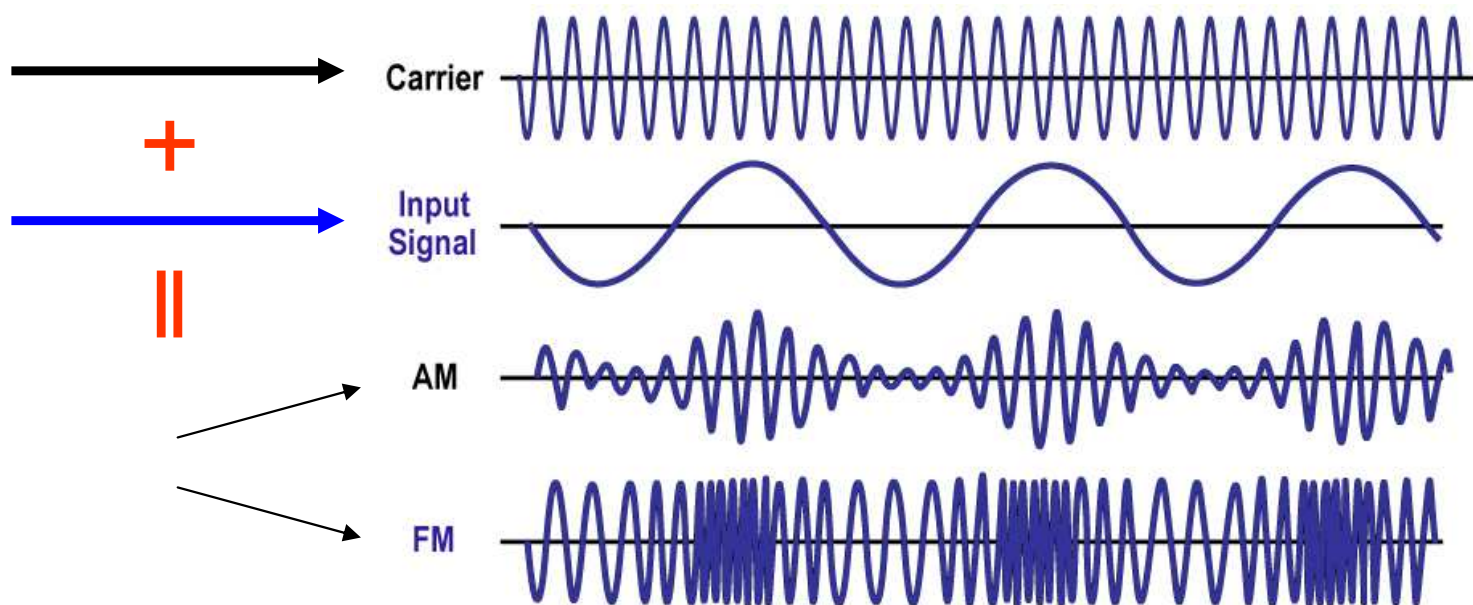
- Daljinska kontrola nekih kućnih uređaja poput TV prijemnika ili klima uređaja, odavno je zastupljena u formi **infracrvenih (IR) daljinskih upravljača**. Međutim, i ostalim uređajima poput celokupnog sistema rasvete, sigurnosti, vrata, grejanja, kao i utičnicama i prekidačima, može se upravljati korišćenjem daljinskih upravljača. A zašto ne bi moglo i robotom?
- Daljinski upravljač **šalje komande kao seriju modulisanih impulsa i pauza**. Ovde signal ili postoji ili ne postoji (tj. ili se šalje modulisani signal frekvencije 30-60kHz ili ne).
- **Sadržaj poruke** je **definisan dužinom signala i dužinom pauze**. Svaki daljinski upravljač ima algoritam za konverziju digitalnih instrukcija u seriju impulsa i pauza (0 i 1) tj. **kodiranje**.

- Talasna dužina IR signala je malo iznad opsega koje ljudsko oko može da registruje.
- IR je skraćenica za infracrvenu svetlost što znači da je izvan vidljivog dela spektra. IR sistemi za prenos podataka koriste veoma visoke frekvencije, nešto više od vidljivog dela spektra. Kao i vidljiva svetlost, infracrvena svetlost ne prolazi kroz zidove i druge neprozirne predmete pa **u slučaju IR prenosa mora da postoji optička vidljivost između predajnika i prijemnika i oni moraju biti u dometu prijemnika-predajnika.**
- Talasna dužina svetlosti koja se koristi za prenos je između 800 i 1000nm, najčešće oko 940nm. **IR prijemnici su najosetljivi na talasne dužine oko 940nm.**
- **Prijemnik će registrovati signal samo ako je modulisan frekvencijom na koju je podešen i ako je talasna dužina tog modulisanog signala odgovarajuća.**

- Da bi se realizovao IR prenos, moramo imati **prijemnik**, **predajnik** i **protokol**.
- **Predajnik** šalje impulse **IR diodom**. Ovi impulsi modulišu nosilac po definisanom protokolu. IR signal se emituje i IR prijemnik ga detektuje.



- Da bi se omogućio neometan prenos potrebno ga je zaštititi od neželjenih izvora. Kao što je poznato, **jedan od izvora infracrvenog zračenja je sunce**, zato se filtrira signal koji dolazi na IR prijemnik.
- Filtar koji je integrisan sa prijemnikom propušta samo signale odgovarajuće talasne dužine (oko 940nm). Pošto je **sunce prirodan izvor emisije nemodulisanih IR zraka** (emituje nemodulisanu IR svetlost) **koristi se modulacija**.
- **Modulacija** nam omogućuje da korektno primamo signale.



- **Kada korisnik pritisne tipku/taster** na daljinskom upravljaču, IR dioda emituje povorku IR impulsa.
IR prijemnik u uređaju **prima te impulse i konvertuje ih u odgovarajuće naponske nivoe koji predstavljaju pritisnuti taster**. Ti impulsi se vode na mikrokontroler koji registruje kôd i zatim izvršava komandu.
- Pošto **prijemnik radi sa inverznom logikom**, **kada dioda emituje izlaz prijemnika je na niskom logičkom nivou**, a kada ne emituje onda je na visokom.

Da bismo utvrdili koja komanda je poslata **moramo znati po kom protokolu** su kodirane komande.

IR (Infra Red) detektori

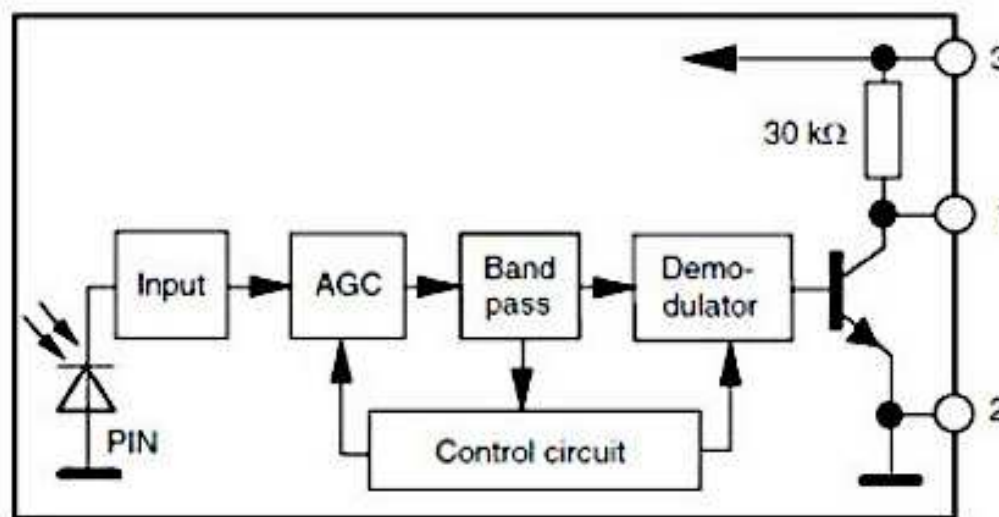
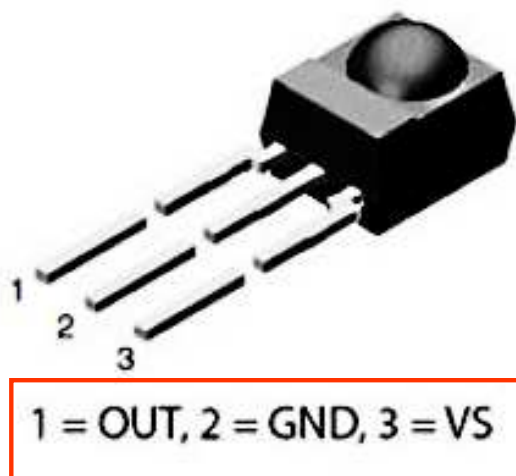
- **IR** (Infra Red) **detektori** su detektori koji su osetljivi na zračenja u infra-crvenom delu spektra. Dva glavna tipa ovih detektora su **termički** i **foto-detektori**. Najčešću primenu nalaze u daljinskim kontrolama i upravljanju uređajima. Ovde ćemo koristiti IR detektor, odnosno, **prijemnu foto diodu model TSOP34838**, namenjenu prvenstveno detekciji signala TV daljinskih upravljača.
- Za razliku od klasičnih foto-detektora, **IR detektori** su konstruisani tako da filtriraju samo zračenja u **infra-red** delu spektra i nisu namenjeni detekciji zračenja iz vidljivog dela spektra, dok su klasične foto-čelije dobri detektori vidljivog dela spektra (žuta/zelena), ali ne i IR dela.
- IR detektori poseduju integrisan demodulator koji ima ulogu u demodulisanju IR signala modulisanog odgovarajućom frekvencijom nosioca. Konkretno, kod modela TSOP34838, ta frekvencija je 38KHz. To zapravo znači da **obična zračeća IR LED dioda ne može biti detektovana od strane TSOP34838, ukoliko ne zrači signal sa frekvencijom nosioca od 38KHz.**

- Za razliku od klasičnih foto-čelija koje nemaju integrisan demodulator i koje se u zavisnosti od količine svetlosti koja pada na foto-čeliju ponašaju kao otpornik promenljive otpornosti, **IR detektori imaju digitalni izlaz.**
- Tačnije, **ukoliko je na mestu prijema detektovan modulisani signal, na digitalnom izlazu je prisutan napon 0V što odgovara logičkoj nuli.** Ukoliko modulisani signal nije prisutan, na izlazu je napon od 5V (3.3V u našem slučaju), što odgovara logičkoj jedinici.
- **TSOP34838** predstavlja prijemnik infracrvenih signala daljinskih upravljača čiji se **izlaz može direktno povezati sa mikrokontrolerom.** Ovaj model IR detektora najčešću primenu nalazi u okviru TV prijemnika i kompatibilan je sa velikom većinom standarda kao što su npr: RC5, RC6, NEC, Panasonic, Sharp, ...

- Karakteristike **IR** prijemne diode **TSOP34838**

Model	TSOP34838
Pinning	1 = OUT, 2 = GND, 3 = Vs
Frekvencija nosioca	38 KHz
Dimenzije	6,9 x 5,6 x 6,0 (W x H x L) [mm]
Napon napajanja – Vs	-0,3 V do +6 V
Struja napajanja – Is	3 mA
Napon na izlazu – Vout	-0,3 V do (Vs+0,3 V)
Struja na izlazu – Iout	5 mA
Temperatura sredine	-25 do +85°C
Snaga	10 mW
Implementirano	Fotodetektor, predpojačavač, filter
Zaštita od neželjenog zračenja	Poboljšana zaštita od elektromagnetske interferencije (EMI), kao i od ambijentalnog osvetljenja
Protokoli	RC-5, RC-6, Panasonic, NEC, Sharp, r-step, Thomson RCA, r-map

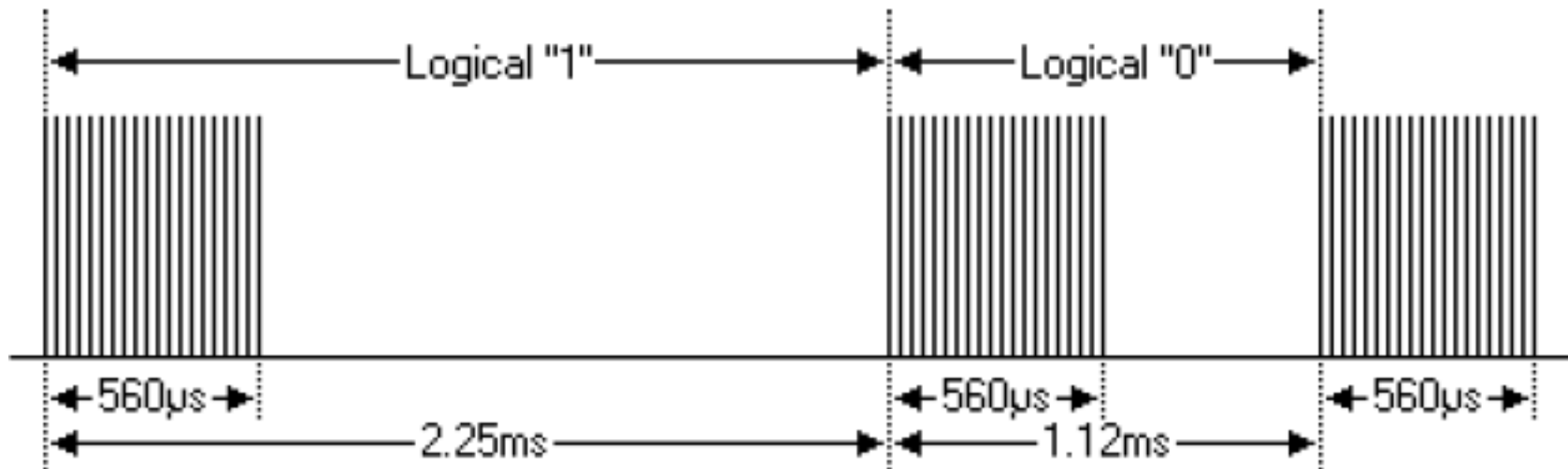
- Izgled i blok dijagram TSOP34838



- Infracrvena komunikacija je bežična komunikacija koja zahteva optičku vidljivost, odnosno prijemnik i predajnik moraju biti na direktnoj liniji, i nijedan objekat se ne sme nalaziti između njih. **Ova vrsta komunikacije služi za povezivanje uređaja na malim razdaljinama, obično u istoj sobi.**
- Različiti proizvođači kućnih uređaja su razvili ili koriste različite protokole za infracrvenu komunikaciju i daljinsko upravljanje. Svi ovi **protokoli za daljinsko upravljanje koriste digitalno modulisan signal.** Noseća frekvencija signala može biti različita, ali se **u 90% slučajeva koristi frekvencija od 38kHz.**
- Za upravljanje audio i video opremom najširu primenu su našli Philipsov **RC-5** (u Evropi i SAD) i **NEC** protokol (u Japanu).

- **NEC protokol** definiše noseću frekvenciju od 38 kHz, dužinu bita 1,125 ms (logička nula) ili 2,25 ms (logička jedinica).

Za kodovanje signala se koristi kodovanje dužinom intervala između pulseva, gde je svaki bit predstavljen visokim stanjem u trajanju od 562,5 μ s, koje je praćeno niskim stanjem u trajanju od 562,5 μ s za logičku nulu, ili od 1,6875 ms za logičku jedinicu.



Kodovanje dužinom signala između pulseva – logička jedinica i logička nula

- Po **NEC** standardu, **poruka počinje** **uvodnim pulsem** u trajanju od 9 ms, **praćenim pauzom** od 4,5 ms.

Sledećih 8 bita predstavljaju **adresu uređaja**, a zatim slede 8 bita koji predstavljaju **logički invertovanu adresu uređaja**, a šalju se radi provere ispravnosti poruke.

Nakon toga, šalje se **sama komanda od 8 bita**, kao i **logički invertovana komanda (8 bita)** radi provere ispravnosti poruke.

Na kraju sledi **puls dužine 562,5 μ s**, koji **označava kraj prenosa** poruke.

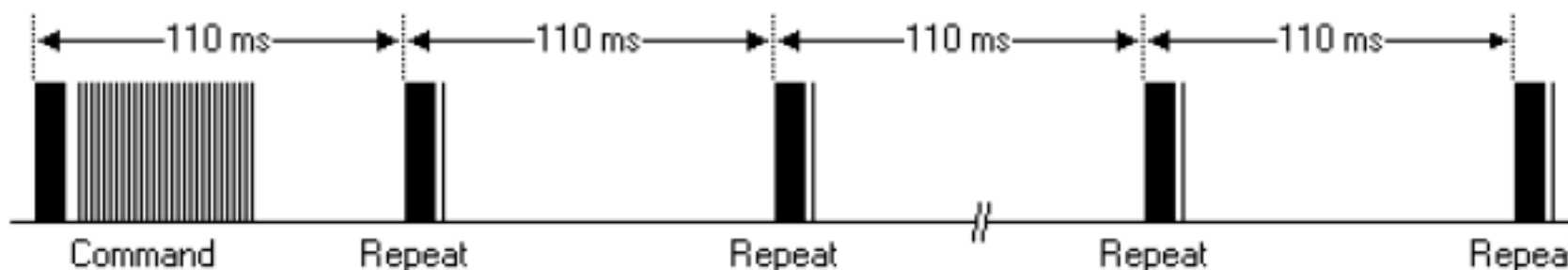
Ukupno trajanje poruke je konstantno, jer se svaki bit šalje dva puta, po jednom u neinvertovanom i invertovanom obliku.



NEC protokol – poruka

- **Poruka se šalje samo jednom, čak iako taster na daljinskom upravljaču ostane aktiviran.**

U tom slučaju se nakon poruke šalje kôd koji se ponavlja svakih 110 ms, a sastoji se od pulsa od 9 ms, pauze od 2,25 ms i pulsa od 562,5 μ s.



Poruka praćena kodom koji se ponavlja na 110 ms

- **IR senzor** nakon otkrivanja/detektovanja IR komande **pretvara svetlosne impulse u strujne i šalje ih ka mikrokontroleru.**
- **Mikrokontroler** od IR senzora prima niz impulsa određene frekvencije. Nakon toga sledi filtriranje po frekvenciji i poređenje dobijene frekvencije sa očekivanom. Očekivane frekvencije su **frekvencije koje koriste IR protokoli**, dakle, 36, 38, 40 i 56 kHz.
- Ukoliko se otkrije željena frekvencija sledi dalje **prepoznavanje protokola**, tj. trajanje impulsa i način njihovog pojavljivanja. Na osnovu ovih podataka može se proverom u tabeli protokola zaključiti koji protokol se koristi.
- Kada se prepozna protokol sledi **dekodovanje signala** po normama odgovarajućeg protokola.

Dekodovanje signala se sastoji u prepoznavanju nula i jedinica.

Tako dekodovane nule i jedinice se dalje obrađuju i u prozoru serijskog monitora se ispisuje **heksakod** komande.

- **IRremote** je biblioteka autora **Ken Shirriffa** koja omogućava da primete i prenosite/šaljete infracrvene kontrolne kodove daljinskih uređaja. Možete napraviti svoje uređaje koje će kontrolisati komercijalni daljinski upravljač ili napraviti svoje uređaje koji će upravljati TV-om, muzičkim stubom ili klima uređajem.
- Ponaša se **kao dve biblioteke**, jedna za slanje a druga za prijem **kodova**. Obično je najlakše detektovati kodove za slanje korišćenjem prijemnika. Ova biblioteka koristi vremenski tajmer mikrokontrolera i za emitovanje/slanje i za prijem.
- Ako želite da koristite još neku biblioteku koja zahteva korišćenje tajmera, morate voditi računa da definišete da ne koriste isti tajmer pošto će doći do blokiranja.

- **Naredba za kreiranje prijemnog objekta** (koristite ime po svom nahođenju).
`IRrecv irrecv(receivePin)`
- **Započinjanje procesa prijema.** Omogućava se tajmerska prekidna rutina (tzv. interapt) koja troši malu količinu CPU resursa svakih 50 mikrosekundi.
`irrecv.enableIRIn()`
- **Naredba koja pokušava da primi IR kodove.** Vraća rezultat `true` ako je kôd primljen, ili `false` ako još ništa nije primila. Kada je kôd primljen, smešta se u promenljivu (struktura podataka) "`results`".
`irrecv.decode(&results)`

- Struktura podataka **results** ima sledeću formu:

`results.decode_type` - jedna od vrednosti: NEC, SONY, RC5, RC6 ili UNKNOWN

`results.value` - aktuelni IR kod (0 ako je tip UNKNOWN)

`results.bits` - broj bita koji taj kôd koristi

`results.rawbuf` - niz IR vremenskih pulseva

`results.rawlen` - broj stavki smeštenih u niz

- **Naredba koja se po prijemu mora pozvati** da bi resetovala prijemnik da ga pripremi za prijem sledećeg koda.

`irrecv.resume()`

- Naredba koja omogućava **blinkovanje svetleće diode za vreme prijema**. Kako **ne možemo da vidimo infracrveno svetlo**, **blinkovanje svetlosne diode** može biti od pomoći ukoliko ima nekih problema i ujedno **daje vizuelnu povratnu informaciju**.

```
irrecv.blink13(true);
```

- **Naredba za kreiranje objekta za emitovanje**. Koristi se uvek fiksni pin, u zavisnosti koji tajmer biblioteka koristi.

```
IRsend irsend;
```

- Naredba za slanje koda u NEC formatu.

```
irsend.sendNEC(IRcode, numBits);
```

- Naredba za slanje koda u Sony formatu.

```
irsend.sendSony(IRcode, numBits);
```

- Naredba za slanje koda u RC5 formatu.

```
irsend.sendRC5(IRcode, numBits);
```

- Naredba za slanje koda u RC6 formatu.

`irsend.sendRC6(IRcode, numBits);`

- Naredba za slanje sirovog (raw) koda. Normalno, treba dobiti sadržaj i dužinu bafera (rawbuf i rawlen) korišćenjem prijemnika više puta i usrednjavanjem rezultata. Može da se desi da budu potrebna određena podešavanja da bi se dobile što bolje performanse. **Očekivana noseća frekvencija je 38.**

`irsend.sendRaw(rawbuf, rawlen, frequency);`

- Ukoliko treba da detektujete kodove daljinskog upravljača možete iskoristiti skeč **IRrecvDemo** biblioteke IRremote.

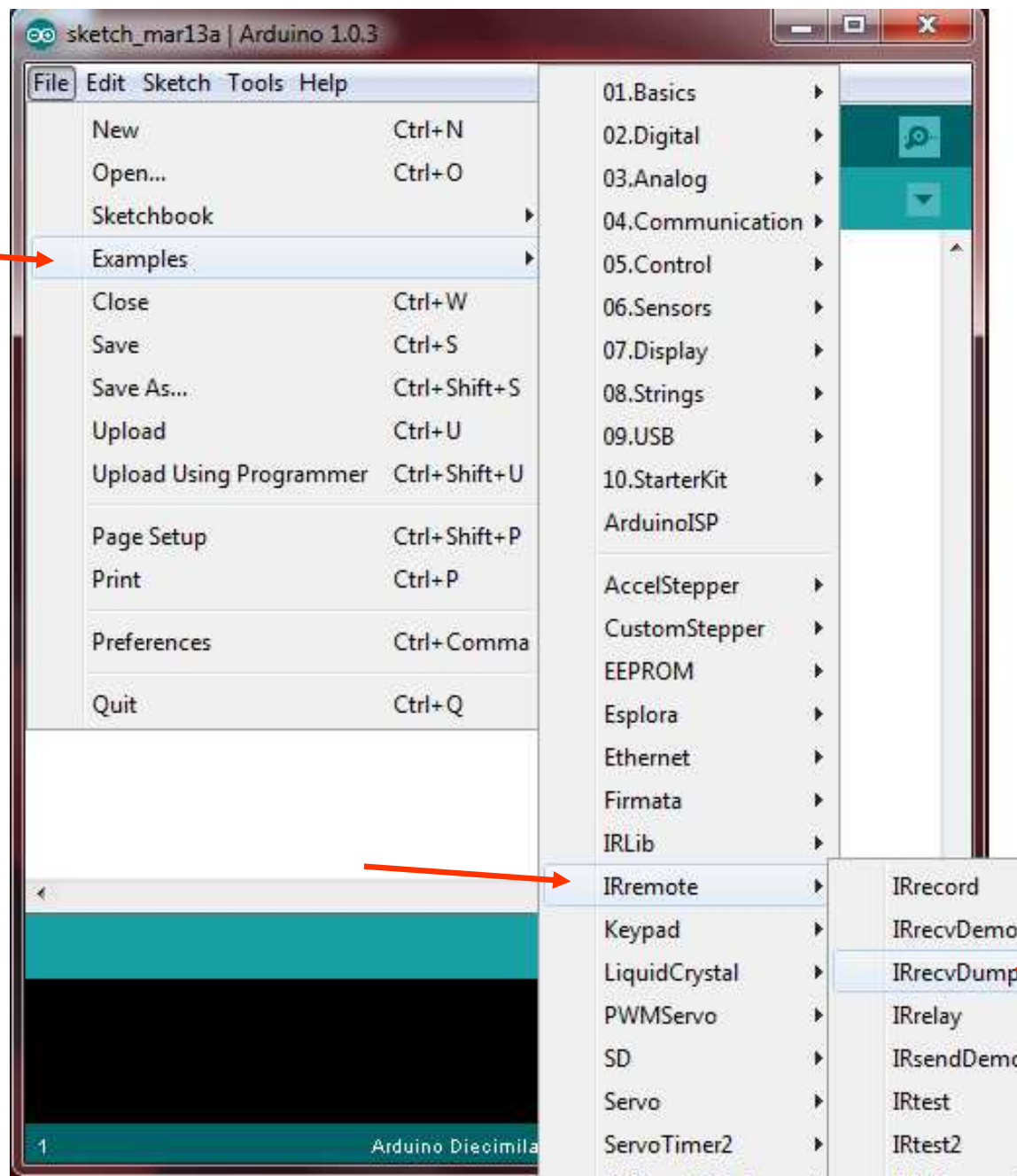
- Nakon preuzimanja (download) biblioteke **IRremote**,

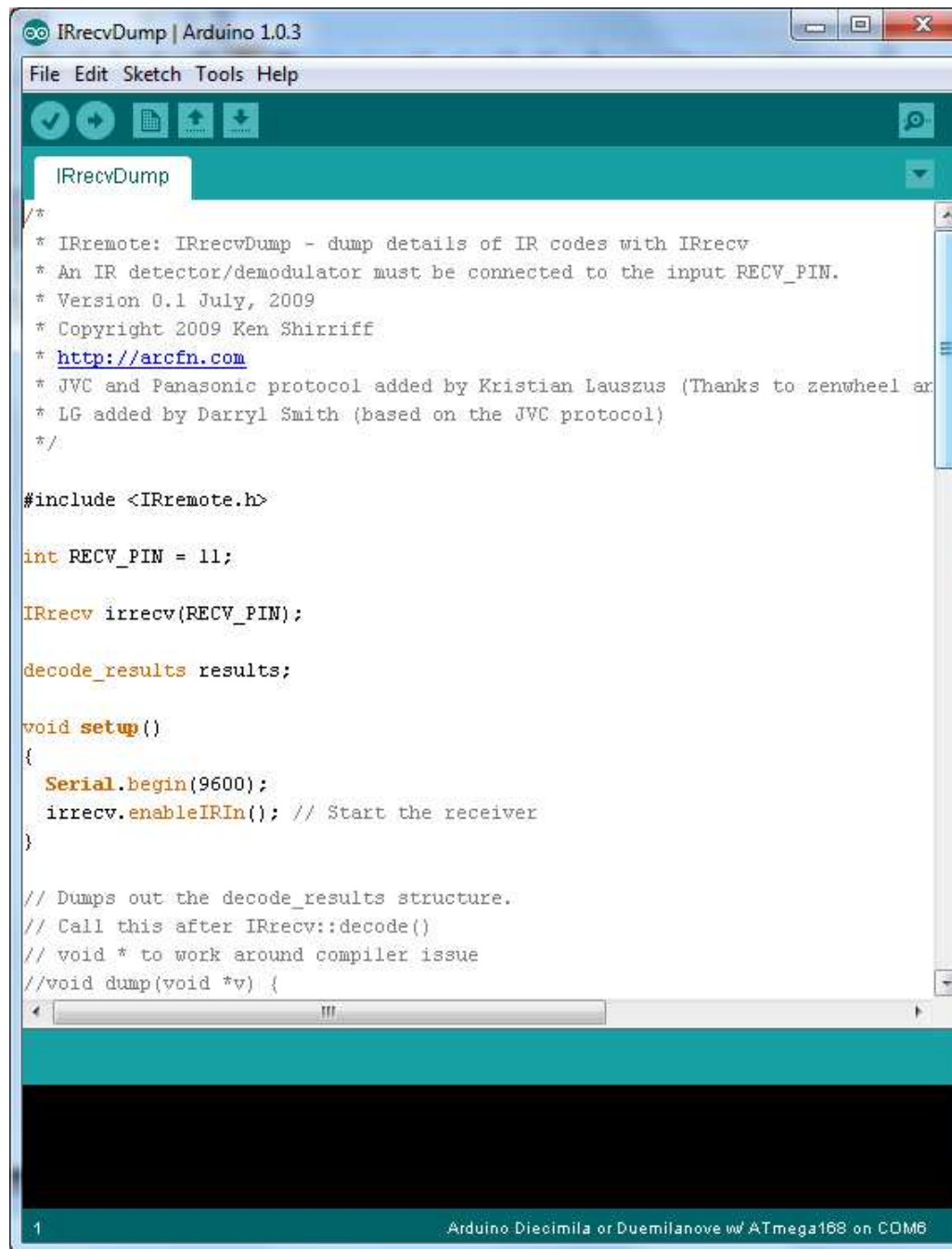
pređite u direktorijum Arduino projekata. Pronađite poddirektorijum [libraies/examples/IRrecvDemo](#) i otvorite fajl [IRrecvDemo.ino](#).

Po kompajliranju, otpremite binarni fajl u svoj Arduino. Skeč će automatski dekodirati tip protokola koji koristi daljinski upravljač i dekodirati kontrolni kôd dugmeta koje je pritisnuto na daljinskom upravljaču.

Otvorite prozor **Serial Monitor** Arduino okruženja, **podesite brzinu komunikacije sa aktivnim serijskim portom na 9600** a zatim pritisnite dugme na daljinskom upravljaču. [Rezultat rada skeča videćete u Serial Monitoru.](#)

... IRremote





```
/*
 * IRremote: IRrecvDump - dump details of IR codes with IRrecv
 * An IR detector/demodulator must be connected to the input RECV_PIN.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 * JVC and Panasonic protocol added by Kristian Lauszus (Thanks to zenwheel and other people at that site)
 * LG added by Darryl Smith (based on the JVC protocol)
 */

#include <IRremote.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

// Dumps out the decode_results structure.
// Call this after IRrecv::decode()
// void * to work around compiler issue
//void dump(void *v) {
```

1

Arduino Diecimila or Duemilanove w/ ATmega168 on COM6

```
/*
```

- * IRremote: IRrecvDump - dump details of IR codes with IRrecv
 - * An IR detector/demodulator must be connected to the input RECV_PIN.
 - * Version 0.1 July, 2009
 - * Copyright 2009 Ken Shirriff
 - * <http://arcfn.com>
 - * JVC and Panasonic protocol added by Kristian Lauszus (Thanks to zenwheel and other people at the original blog post)
 - * LG added by Darryl Smith (based on the JVC protocol)
- ```
*/
```

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

```
int RECV_PIN = 11;
```

```
IRrecv irrecv(RECV_PIN);
```

```
decode_results results;
```



```
void setup()
{
 Serial.begin(9600);
 irrecv.enableIRIn(); // Start the receiver
}

// Dumps out the decode_results structure.
// Call this after IRrecv::decode()
// void * to work around compiler issue
//void dump(void *v) {
// decode_results *results = (decode_results *)v
void dump(decode_results *results) {
 int count = results->rawlen;
 if (results->decode_type == UNKNOWN) {
 Serial.print("Unknown encoding: ");
 }
 else if (results->decode_type == NEC) {
 Serial.print("Decoded NEC: ");
 }
}
```

```
else if (results->decode_type == SONY) {
 Serial.print("Decoded SONY: ");
}
else if (results->decode_type == RC5) {
 Serial.print("Decoded RC5: ");
}
else if (results->decode_type == RC6) {
 Serial.print("Decoded RC6: ");
}
else if (results->decode_type == PANASONIC) {
 Serial.print("Decoded PANASONIC - Address: ");
 Serial.print(results->panasonicAddress, HEX);
 Serial.print(" Value: ");
}
else if (results->decode_type == LG) {
 Serial.print("Decoded LG: ");
}
else if (results->decode_type == JVC) {
 Serial.print("Decoded JVC: ");
}
```

```
Serial.print(results->value, HEX);
Serial.print(" ");
Serial.print(results->bits, DEC);
Serial.println(" bits");
Serial.print("Raw ");
Serial.print(count, DEC);
Serial.print(": ");

for (int i = 0; i < count; i++) {
 if ((i % 2) == 1) {
 Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
 }
 else {
 Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);
 }
 Serial.print(" ");
}
Serial.println("");
}
```

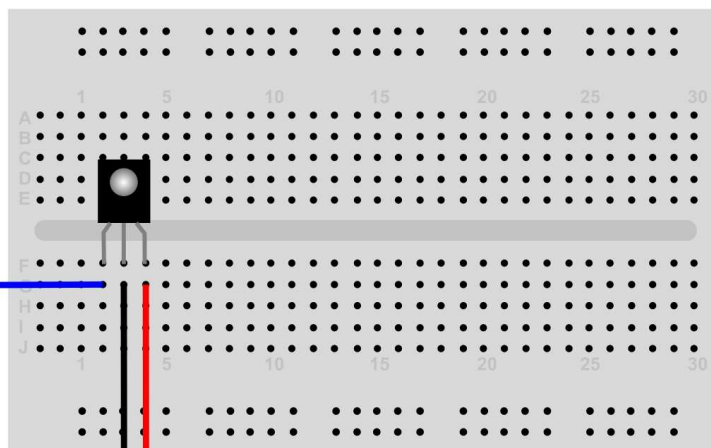
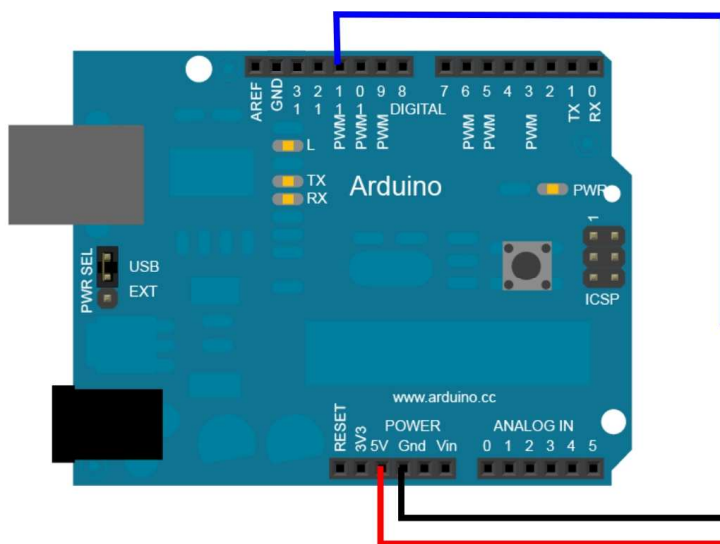
```
void loop() {
 if (irrecv.decode(&results)) {
 Serial.println(results.value, HEX);
 dump(&results);
 irrecv.resume(); // Receive the next value
 }
}
```

# ... Rad sa IR prijemnikom

TSOP 34838



11



GND

Vcc



# ... dekodiranje komandi

```
IRrecvDump | Arduino 1.0.3
File Edit Sketch Tools Help

IRrecvDump
* An IR detector/demodulator must be connected to the input RE
* Version 0.1 July, 2009
* Copyright 2009 Ken Shirriff
* http://arcfn.com
* JVC and Panasonic protocol added by Kristian Lauszus (Thanks
* LG added by Darryl Smith (based on the JVC protocol)
*/

#include <IRremote.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
 Serial.begin(9600);
}

Done uploading.
Binary sketch size: 10,262 bytes (of a 32,256 byte maximum)

1 Arduino Uno on COM7
```

COM7

Send

20DF02FD  
Decoded NEC: 20DF02FD (32 bits)  
Raw (68): -22708 9000 -4300 600 -500 650 -500 600 -1600

20DFE01F  
Decoded NEC: 20DFE01F (32 bits)  
Raw (68): -7348 8950 -4400 600 -500 600 -500 650 -1600

20DF609F  
Decoded NEC: 20DF609F (32 bits)  
Raw (68): 5120 8950 -4350 650 -500 600 -500 650 -1550

☒ Autoscroll Newline 9600 baud



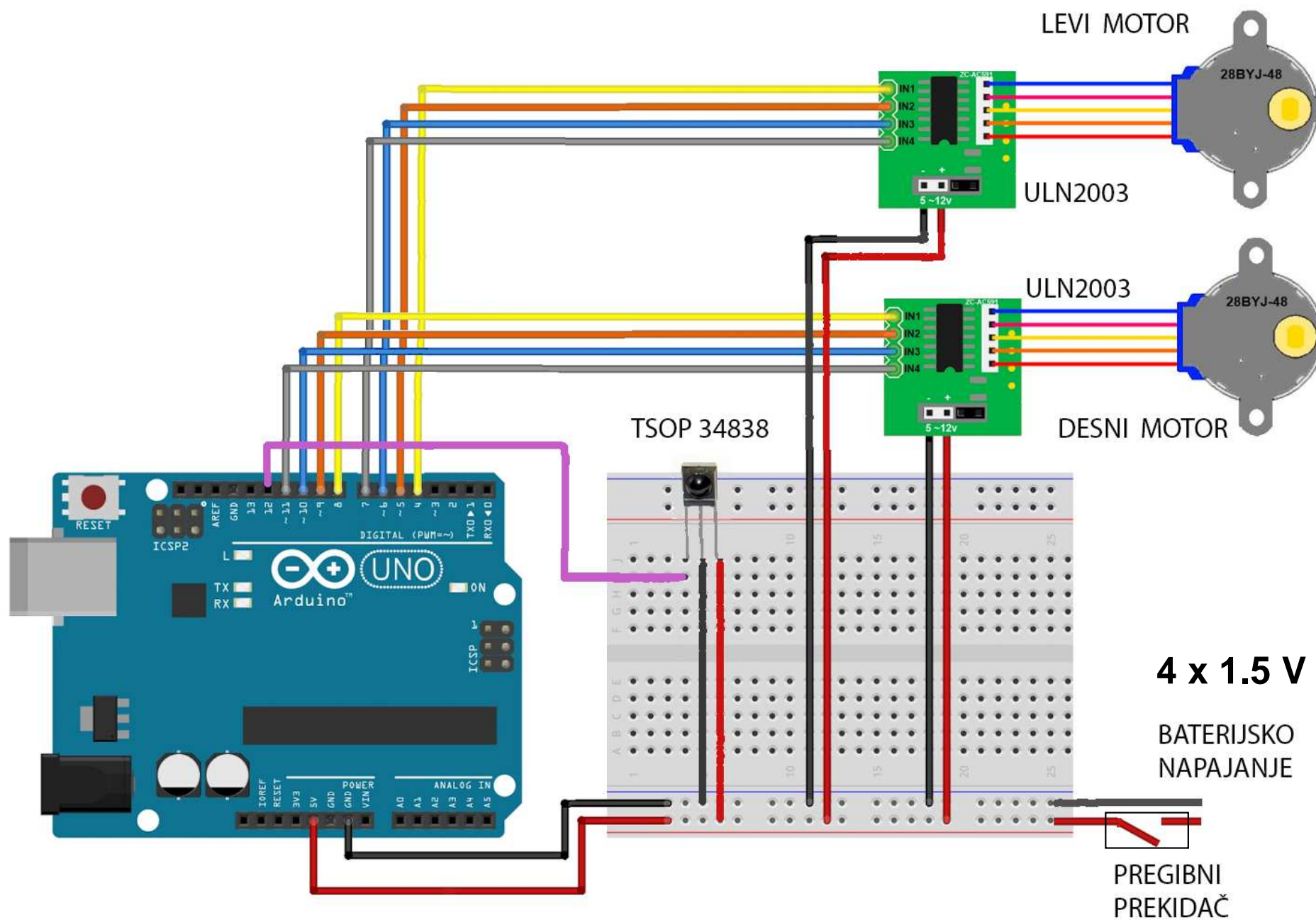


- **Ručni režim rada** – proizvoljno kretanje zadavanjem komandi
  - Napred
  - Nazad
  - Levo
  - Desno
  - Stop
- **Automatski režim rada** – kretanje po programski definisanoj kvadratnoj putanji
- **Prebacivanje režima** – u izabranom režimu rada pritiskom na taster za drugi režim rada

# Kodovi tastera “daljinca” - mapiranje

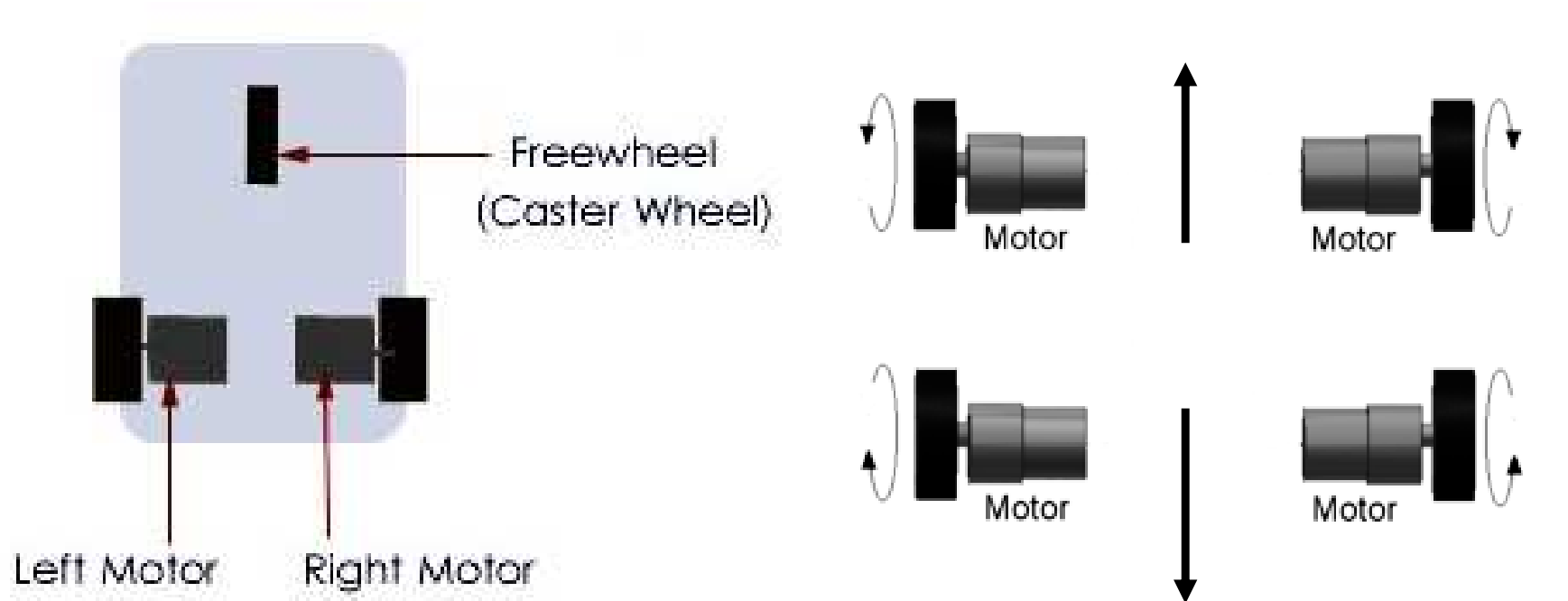


# Šema povezivanja

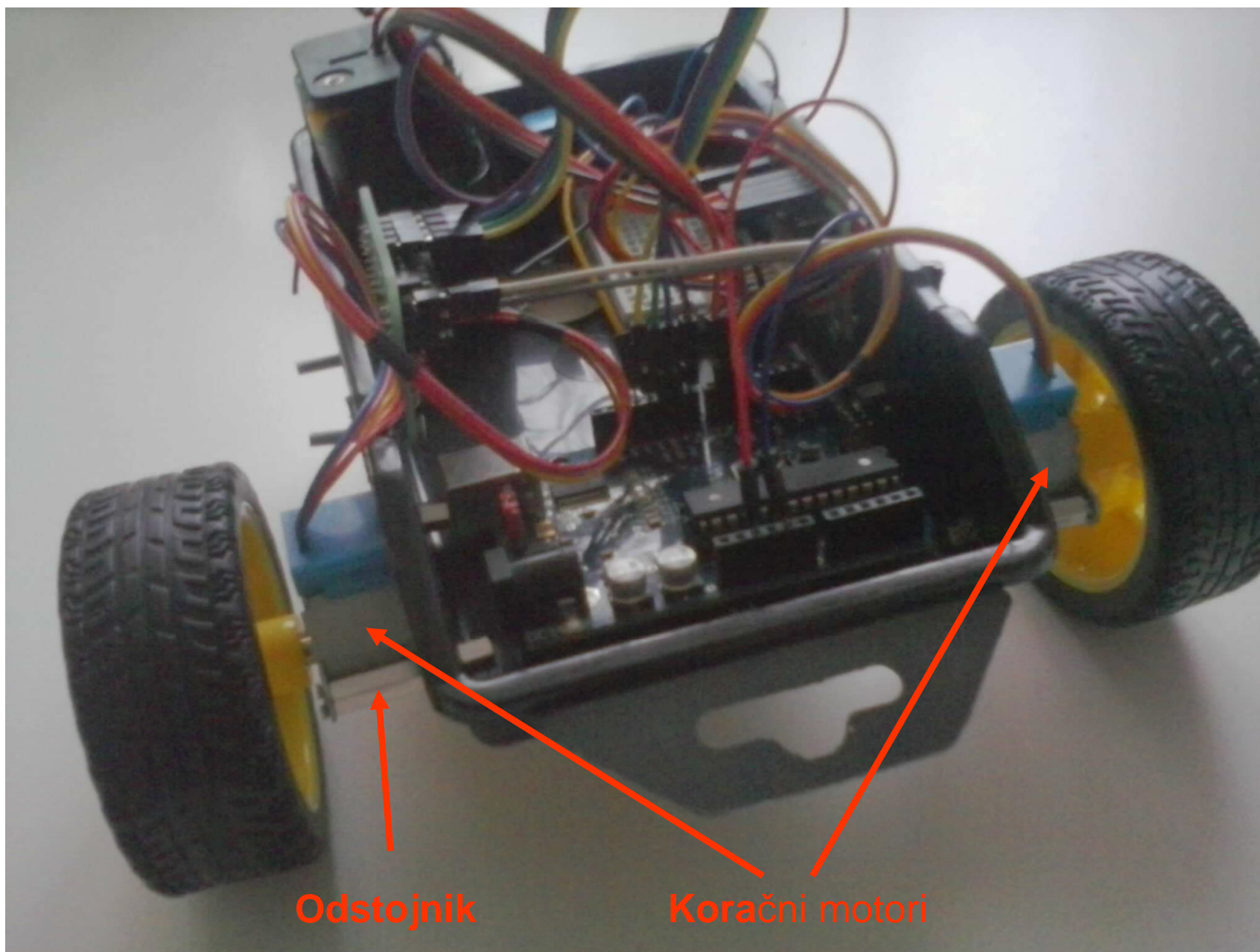


# Mobilni robot sa diferencijalnim pogonom

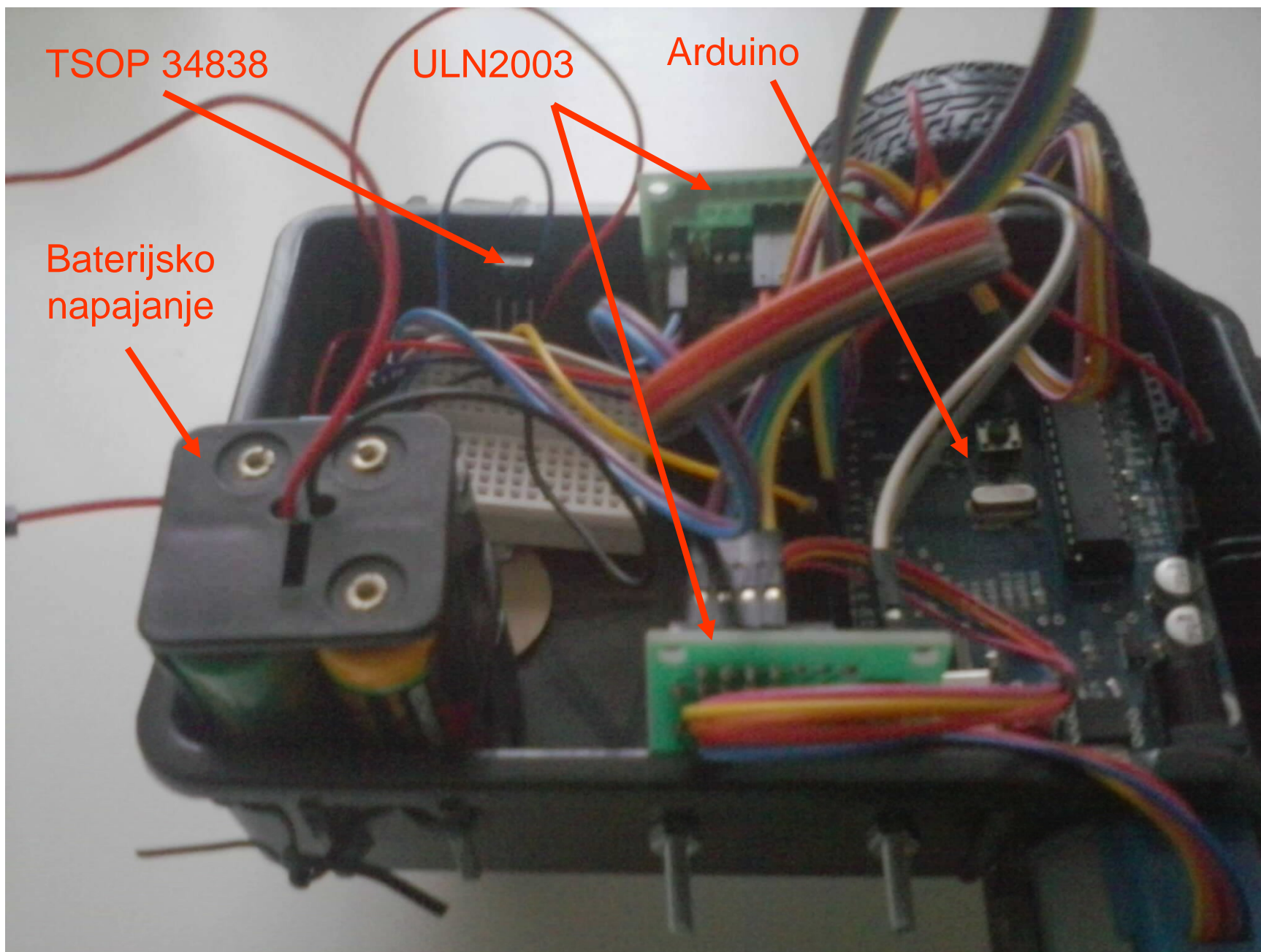
- Mnogi mobilni roboti su bazirani na konstrukciji sa **dva supotno upravljana točka**, tj. sa **diferencijalnim pogonom**. Robot baziran na ovoj konstrukciji koristi **dva nezavisno upravljiva točka**.
- Kontrolisanjem svakog od dva glavna motora za točkove omogućuje se jednostavno manevrisanje robotom

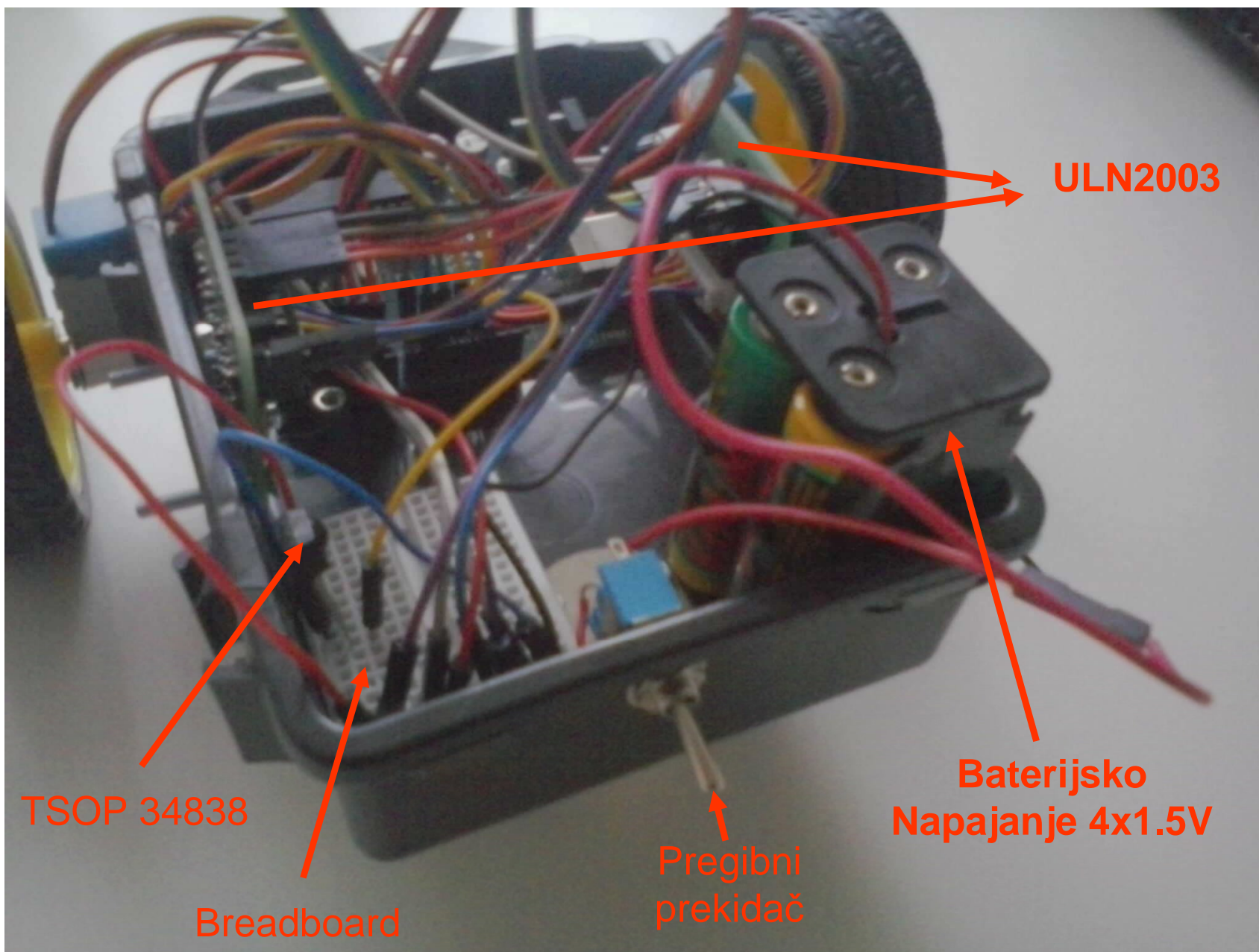


## ... Izmontiran mobilni robot

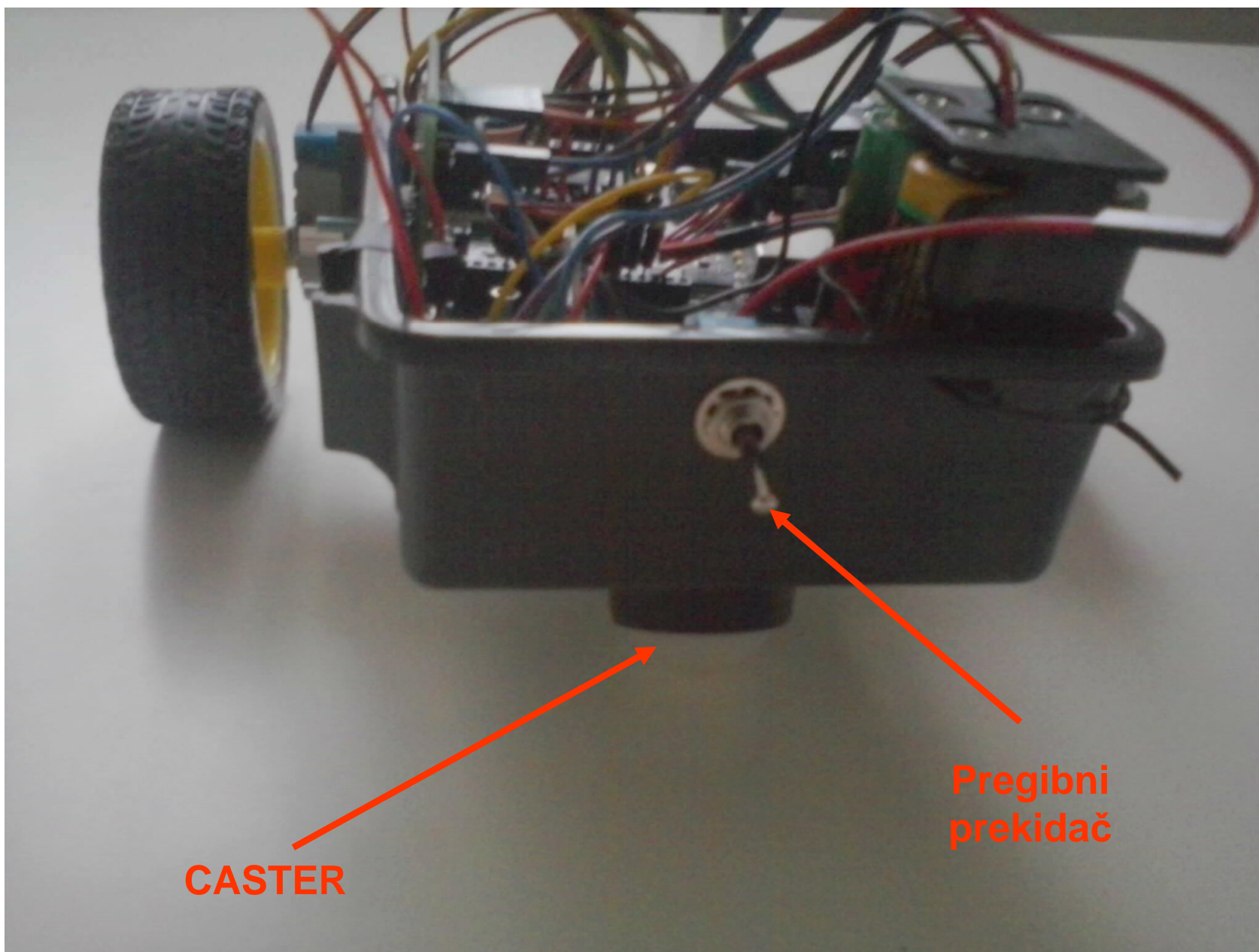












## Kretanje diferencijalnog robota napred za zadato **rastojanje od 40 cm**

Robot ima točak (točkove) **prečnika 6.6 cm**, tako da možemo izračunati **koliko puta treba točak da se okrene da bi prešao željeno rastojanje**.

Za **jedan puni okret**, točak pređe rastojanje:

$$\text{obim} = 3.145 * 6.6 \text{ cm} = 20.757 \text{ cm}$$

Ukoliko želimo da robot pređe rastojanje od **40 cm**, potrebno je da se točkovi okrenu (rotiraju) ...

$$N = 40 \text{ cm} / \text{obim}$$

Međutim, umesto korišćenja broja rotacija točka, može se koristiti **broj ugaonih stepeni** ...

$$\text{stepeni} = N * 360$$

## ... dimenzije korišćenih točkova

- Outer dia of wheel: 65mm
- Width: 28mm
- Internal Axe Diamter: 3.7x5.3mm
- External Axe Diameter: 13.8mm



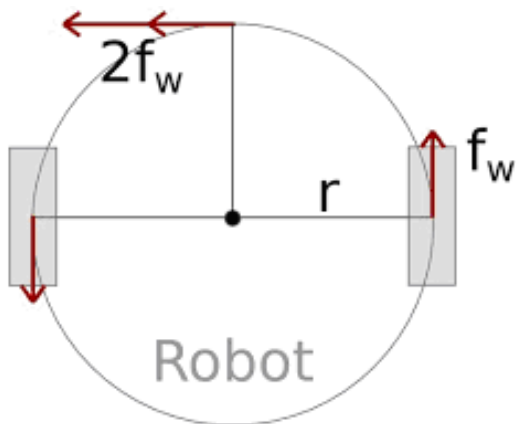
Koristi se koračni motor koji radi u režimu **polukoraka**. Taj motor za jedan pun okretaj (360 stepeni) **napravi 4096 koraka u tom režimu**.

Za nas je zgodnije da **za upravljanje** koračnim motorom koristimo **broj koraka**.

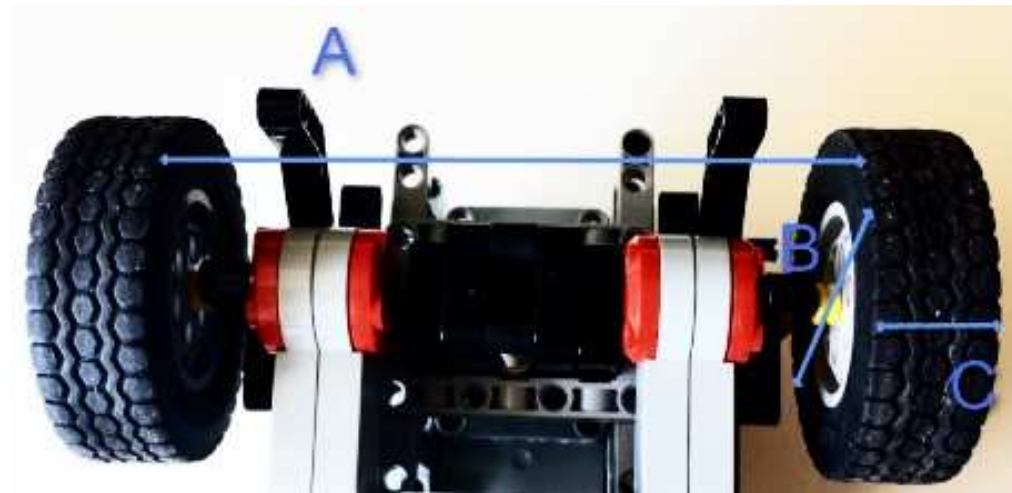
Prema tome sledi:

Za rastojanje od **40 cm** motor treba da napravi  
 **$1.927 * 4096 = 7893$  koraka**

# Rotacija diferencijalnog robota za 90 stepeni (skretanje)

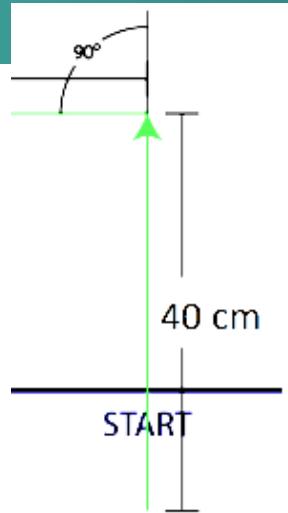


$$A + C = 16.4 \text{ cm}$$



Kako ovaj tip robota može da rotira oko tačke (točkovi se rotiraju istom brzinom ali u različitim pravcima), on će pri tom manevru napraviti **krug** poluprečnika približno **polovini širine robota** (tj. međuosno rastojanje između levog i desnog točka).

$$\text{pun\_krug} = \text{širina} * 3.145 = 16.4 \text{ cm} * 3.145 = 51.578 \text{ cm}$$



Međutim, u ovom slučaju nama nije potrebna puna rotacija, već rotacija za 90 stepeni, tj. **četrтина круга** ...

$$\text{put\_skretanje} = \text{pun\_krug} / 4 = 51.578 / 4 = 12.895 \text{ cm}$$

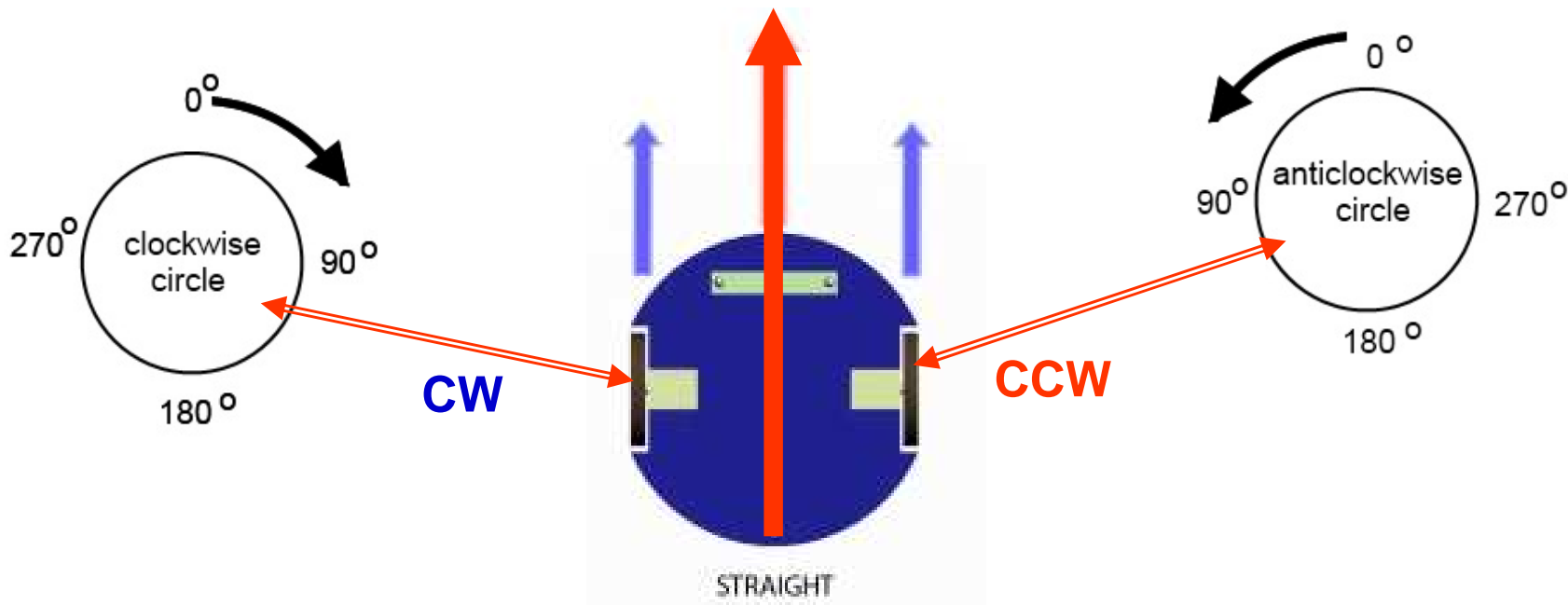
To sada prevodimo u **broj rotacija točka** (preko **obima** točka):

$$\text{br\_rotacija} = \text{put\_skretanje} / \text{obim} = 12.895 / 20.757 = 0.6212$$

**broj\_koraka** =  $0.6212 * 4096 = 2544.43 = 2544$  koraka za skretanje od 90 stepeni

# Programiranje robota za vožnju napred

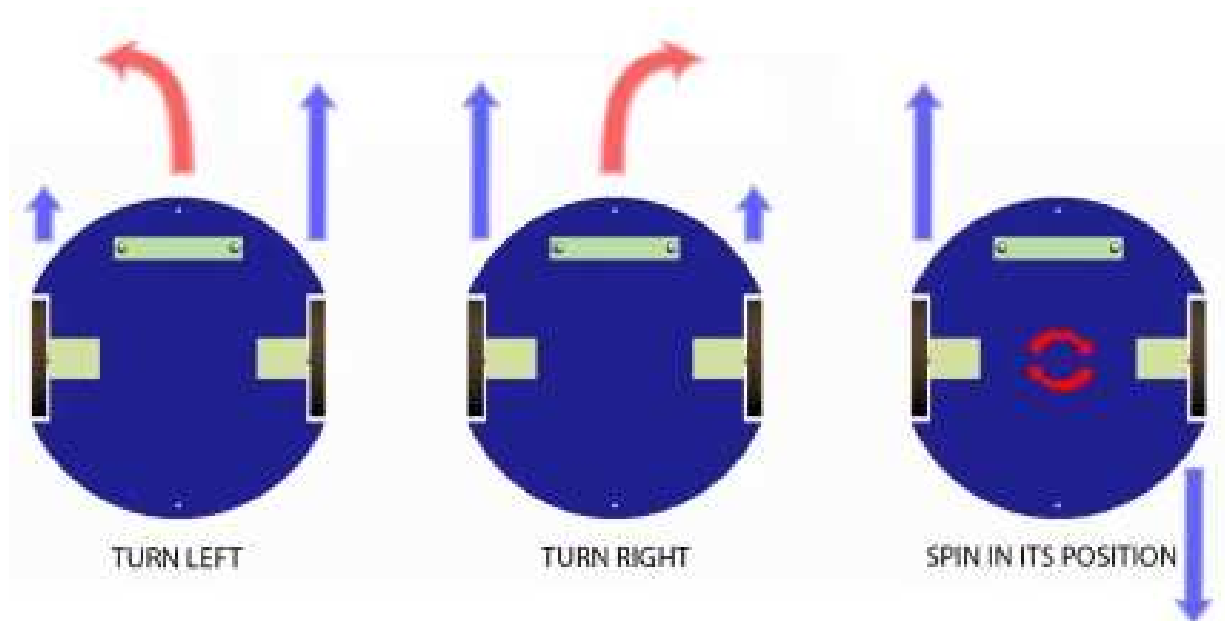
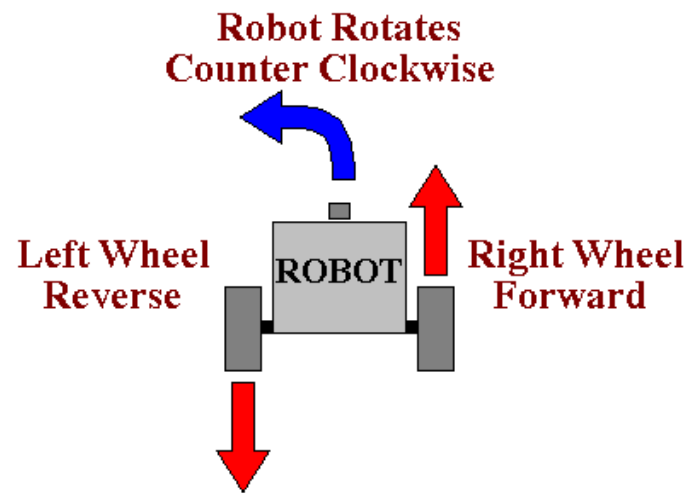
- Potrebno je upravljati sa oba motora kako bi se primenila snaga za pokretanje robota napred. Međutim kako su montirani u suprotnim pravcima, to rezultira smerom naredbe koja je suprotna za svaki motor.
- Da se oba motora kreću u pravcu napred, **mora se programirati levi motor u smeru kazaljke sata**, dok je **desni u pravcu suprotnom od smera kazaljke sata**.





# Programiranje robota za skretanje

- Potrebno je napraviti obrtanje oba motora u istom pravcu umesto suprotnom.



# Programiranje robota za vožnju po kvadratu

- Može se programirati vožnja robota po kvadratu jednostavnim sledom naredbi za vožnju napred i okretanjem za  $90^\circ$ .
- Kako kvadrat ima četiri identične stranice i iste uglove, može se **programirati vožnja robota po kvadratu izvođenjem dva koraka** – 1) vozi pravo i 2) zakreni za  $90^\circ$ .
- **Testiranje** programa će pokazati da **robot ne vozi po idealnom kvadratu**. To je zbog toga što **uglovi nisu baš tačno  $90^\circ$** .
- **Slabljenjem** baterija **menja se ponašanje robota**.
- Takođe, **karakteristike robota zavise od površine po kojoj se kreće**. **Razlika u trenju** kod različitih površina ima snažan uticaj na ponašanje robota.



■ ■ ■



- [ Move Forward S
- Turn Right 90°
- [ Move Forward S
- Turn Right 90°
- [ Move Forward S
- Turn Right 90°
- [ Move Forward S
- Turn Right 90°

or

Repeat times

- Move Forward S
- Turn Right  $90^\circ$

Remember: Turn Right 90 is  
drive goto(26, -25)

- **AccelStepper** je biblioteka za kontrolu različitih drajvera koračnih motora kao što je POLOLU koji se obično nalazi u 3D printerima, zatim EasyDriver, ULN2003 itd. **Može da upravlja sa više koračnih motora istovremeno.**
- Naredba za definisanje i konfigurisanje motora kada se koristi drajver ULN2003  
`AccelStepper mystepper(4, pinA1, pinA2, pinB1, pinB2);`
- Naredba za postavljanje **maksimalne brzine**  
`mystepper.setMaxSpeed(stepsPerSecond);`

Postavlja maksimalnu brzinu. Podrazumevana brzina je vrlo mala, tako da je poželjno konfigurisati ovu vrednost. Kada se koračni motor upravlja postavljanjem položaja/pozicije, on će se ka odredištu ubrzavati ka toj brzini a usporavati kako dostiže odredište.

- Naredba za postavljanje **ubrzanja** koje će se koristiti `mystepper.setAcceleration(stepsPerSecondSquared);`
- **Naredbe koje se odnose na položaj/poziciju**
- Pomeranje motora na novu **apsolutnu poziciju**. Odmah vraća rezultat. Stvarno kretanje realizuje funkcija `run()`.  
`mystepper.moveTo(targetPosition);`
- Pomeranje motora na **relativnu poziciju** u odnosu na tekuću. Odmah se izlazi iz funkcije. Stvarno kretanje realizuje naredba `run()`.  
`mystepper.move(distance);`

- Naredba za očitavanje tekuće apsolutne pozicije motora  
`mystepper.currentPosition();`
- Naredba za očitavanje rastojanja motora od svog odredišta. Može da se koristi za proveru da li je motor dostigao svoje krajnje odredište.  
`mystepper.distanceToGo();`
- Naredba za stvarno aktiviranje motora (ažuriranje pozicije). Mora se pozivati repetitivno da bi se ostvarilo kretanje motora.  
`mystepper.run();`
- Naredba za stvarno aktiviranje motora (ažuriranje pozicije) koja čeka da motor stigne do svog odredišta. Ova funkcija se izvršava sve dok se motor ne zaustavi, tako da je korisna samo ako nema drugih motora u pokretu.  
`mystepper.runToPosition();`



- Naredba za postavljanje brzine motora (u koracima po sekundi). Odmah se izlazi iz funkcije. **Stvarno kretanje realizuje naredba `runSpeed()`.**  
`mystepper.setSpeed(stepsPerSecond);`
- Naredba za stvarno aktiviranje motora (ažuriranje pozicije). Mora se pozivati repetitivno da bi se ostvarilo kretanje motora.  
`mystepper.runSpeed();`

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

```
#include <AccelStepper.h>
```

→ 

```
#define HALFSTEP 8
```

```
// LEFT MOTOR
```

```
#define motorPin1 3 // IN1 on ULN2003 ==> Blue on 28BYJ-48
```

```
#define motorPin2 4 // IN2 on ULN2004 ==> Pink on 28BYJ-48
```

```
#define motorPin3 5 // IN3 on ULN2003 ==> Yellow on 28BYJ-48
```

```
#define motorPin4 6 // IN4 on ULN2003 ==> Orange on 28BYJ-48
```

```
// RIGHT MOTOR
```

```
#define motorPin5 8 // IN1 on the ULN2003 driver 2
```

```
#define motorPin6 9 // IN2 on the ULN2003 driver 2
```

```
#define motorPin7 10 // IN3 on the ULN2003 driver 2
```

```
#define motorPin8 11 // IN4 on the ULN2003 driver 2
```

```
int RECV_PIN = 12; //IR pin
IRrecv irrecv(RECV_PIN);
decode_results results;
```

```
AccelStepper stepper1(HALFSTEP, motorPin1, motorPin3, motorPin2, motorPin4); //
AccelStepper stepper2(HALFSTEP, motorPin5, motorPin7, motorPin6, motorPin8); //
```

```
int k = 6; // STOP moving
int kk = 0; // kk = 1 AUTOMATIC, kk = 2 MANUAL
```

```
int turnSteps = 2480; // number of steps for a 90 degree turn
int lineSteps = -6200; //number of steps to drive straight
//
```

```
int stepperSpeed = 1000; //speed of the stepper (steps per second)
```

```
int steps1 = 0; // keep track of the step count for motor 1
int steps2 = 0; // keep track of the step count for motor 2
```

```
boolean turn1 = false; //keep track if we are turning or going straight next
boolean turn2 = false; //keep track if we are turning or going straight next
```

```
void setup()
{
 //delay(3000); //some time to put the robot down after swithing it on

 irrecv.enableIRIn(); // Start the receiver

 stepper1.setMaxSpeed(2000);
 stepper2.setMaxSpeed(2000);

 stepper1.move(-1);
 stepper2.move(1);

 stepper1.setSpeed(0);
 stepper2.setSpeed(0);
}
```

```
/******/
void loop() {
 //-----
 if (irrecv.decode(&results)) {

 if (results.value == 0x20DF4EB1){
 kk = 2; //MANUAL
 }
 //-----
 if (results.value == 0x20DF8E71){
 kk = 1; // AUTOMATIC
 }
 }
 //****
```

```
if ((results.value == 0x20DF02FD)&& (kk == 2))
{
 k = 2; // FORWARD
}
if ((results.value == 0x20DF827D) && (kk == 2))
{
 k = 3; //BACKWARD
}
if ((results.value == 0x20DFE01F)&& (kk == 2))
{
 k = 4; // LEFT
}
if ((results.value == 0x20DF609F)&& (kk == 2))
{
 k = 5; // RIGHT
}
if (results.value == 0x20DF22DD)//&& (kk == 0)
{
 k=6; // STOP
}
```



```
irrecv.resume(); // Receive the next IR value
```

```
}
```

```
//----- AUTOMATIC
```

```
if(kk==1){
```

```
 stepper1.setSpeed(stepperSpeed);
```

```
 stepper2.setSpeed(stepperSpeed);
```

```
 if (steps1 == 0) {
 int target = 0;
```

```
 if (turn1 == true) {
 target = turnSteps;
```

```
 }
```

```
 else {
```

```
 target = lineSteps;
```

```
 }
```

```
 stepper1.move(target);
```

```
 stepper1.setSpeed(stepperSpeed);
```

```
 turn1 = !turn1;
```

```
 } // end steps1
```

```
if (steps2 == 0) {
 int target = 0;

 if (turn2 == true) {
 target = turnSteps; ←
 }
 else {
 target = -lineSteps; ←
 }
 stepper2.move(target);
 stepper2.setSpeed(stepperSpeed);
 turn2 = !turn2;
} //end steps2
```

```
steps1 = stepper1.distanceToGo();
steps2 = stepper2.distanceToGo();
```

```
stepper1.runSpeedToPosition();
stepper2.runSpeedToPosition();
} // end kk==1
```

```
//----- MANUAL
```

```
// STOP
```

```
if(k==6){
```

```
 stepper1.setSpeed(0); //
```

```
 stepper2.setSpeed(0);
```

```
}
```

```
// FORWARD
```

```
if(k==2){
```

```
 stepper1.setSpeed(-stepperSpeed); // CCW, LEFT MOTOR
```

```
 stepper2.setSpeed(stepperSpeed); //CW, RIGHT MOTOR
```

```
}
```

```
// BACKWARD
```

```
if(k==3){
```

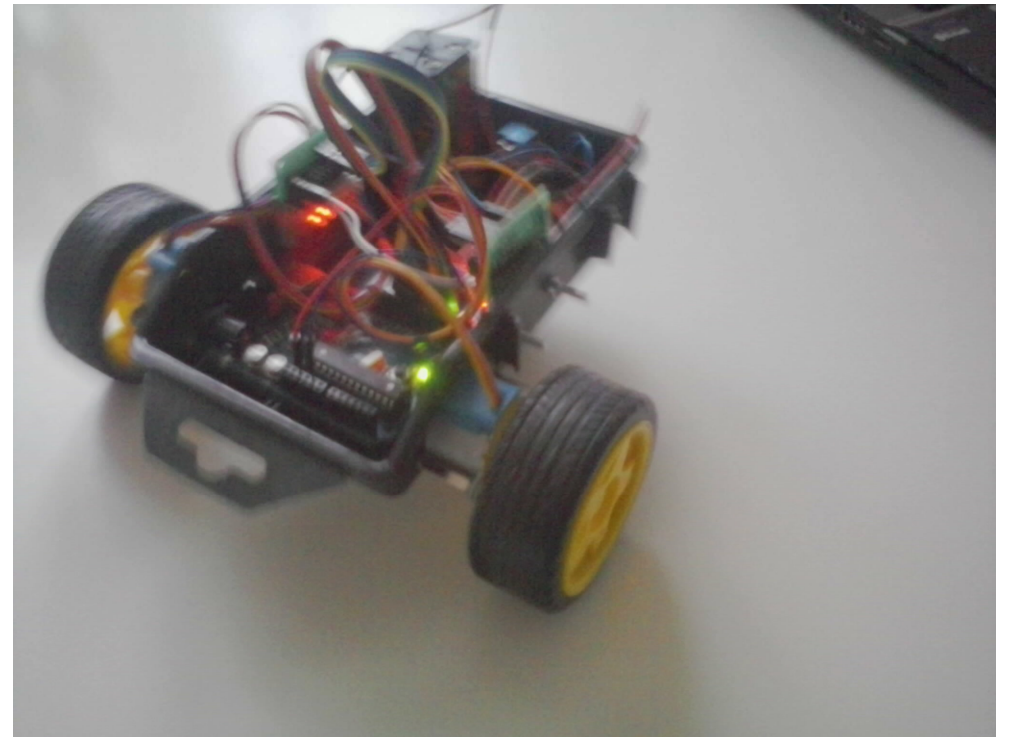
```
 stepper1.setSpeed(stepperSpeed); // CW
```

```
 stepper2.setSpeed(-stepperSpeed); // CCW
```

```
}
```

```
// LEFT
 if(k==4){
 stepper1.setSpeed(150); // CW
 stepper2.setSpeed(500); // CW
 }
// RIGHT
 if(k==5){
 stepper1.setSpeed(-500); // CCW
 stepper2.setSpeed(-150); // CCW
 }
// MANUAL
 if (kk==2){
 stepper1.runSpeed();
 stepper2.runSpeed();
 }

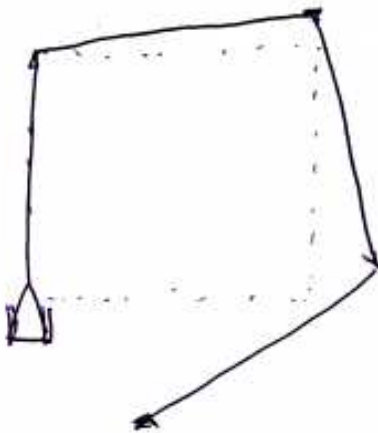
}
```



# Nedostaci i poboljšanja!

- Kalibracija?! Problem preciznog realizovanja programske putanje.

## Mobile robot GUARDIAN



<http://www.robotnik.eu/mobile-robots/guardian/>

- Uključivanje optičkih senzora koji omogućuju praćenje linije.
- Uključivanje ultrazvučnih senzora za detektovanje prepreka
- GPS moduli
- Kamere





Prezentacija se nalazi na stranici predmeta **Teorija sistema**

<http://predmet.singidunum.ac.rs/mod/folder/view.php?id=312>

**Ova prezentacija je nekomercijalna.**

Slajdovi mogu da sadrže materijale preuzete sa Interneta, stručne i naučne građe, koji su zaštićeni Zakonom o autorskim i srodnim pravima. Ova prezentacija se može koristiti samo privremeno tokom usmenog izlaganja nastavnika u cilju informisanja i upućivanja studenata na dalji stručni, istraživački i naučni rad i u druge svrhe se ne sme koristiti –

**Član 44** - Dozvoljeno je bez dozvole autora i bez plaćanja autorske naknade za nekomercijalne svrhe nastave:  
(1) javno izvođenje ili predstavljanje objavljenih dela u obliku neposrednog poučavanja na nastavi; - ZAKON O AUTORSKOM I SRODNIM PRAVIMA ("Sl. glasnik RS", br. 104/2009 i 99/2011)

Dragan S. Marković