

**Mladi za napredek Maribora 2018**

**35. srečanje**

# **DALJINSKO VODENA ROBOTSKA ROKA**

**Raziskovalna naloga**

**Raziskovalno področje: ELEKTROTEHNIKA, ELEKTRONIKA**

Avtor: GAŠPER MAJAL, NEJC GROBELNIK

Mentor: ROBERT GAŠPARIČ

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

**MARIBOR, JANUAR 2018**

**Mladi za napredek Maribora 2018**

**35. srečanje**

# **DALJINSKO VODENA ROBOTSKA ROKA**

**Raziskovalna naloga**

**Raziskovalno področje: ELEKTROTEHNIKA, ELEKTRONIKA**

**MARIBOR, JANUAR 2018**

# 1. KAZALO

## 1.1. Kazalo vsebine

1.	KAZALO .....	1
1.1.	Kazalo vsebine .....	1
1.2.	Kazalo slik .....	2
1.3.	Kazalo tabel .....	3
2.	POVZETEK.....	4
3.	ZAHVALA .....	4
4.	UVOD .....	4
5.	VSEBINSKI DEL .....	5
5.1.	Robotska roka .....	5
5.2.	Arduino razvojne ploščice.....	6
5.3.	Bluetooth komunikacija.....	8
5.3.1.	Bluetooth modul HC-05.....	8
5.4.	Povezovanje bluetooth modulov .....	9
5.5.	Servo motorji.....	13
5.5.1.	Krmiljenje servo motorjev z Arduino ploščico .....	15
5.6.	Robotska roka na pogonskem vozičku .....	16
5.6.1.	Enosmerni motorji z reduktorjem.....	17
5.6.2.	Krmiljenje DC motorčkov .....	18
5.6.3.	PWM signal za krmiljenje hitrosti DC motorja .....	21
5.6.4.	Proženje PWM signalov s ploščico Arduino MEGA za krmiljenje hitrosti DC motorčka.....	22
5.7.	Igralna palica, ang. »Joystick«.....	26
5.8.	Daljinsko upravljanje servo motorja z igralno palico preko bluetooth povezave .....	28
6.	POTEK IZDELAVE PROJEKTA .....	32
7.	UGOTOVITVE IN REZULTATI .....	32
8.	ZAKLJUČEK.....	33
9.	DRUŽBENA ODGOVORNOST .....	33
10.	LITERATURA IN VIRI.....	34
10.1.	Knjižni viri .....	34
10.2.	Spletni viri.....	34

10.3.	Viri slik .....	34
-------	-----------------	----

## 1.2. Kazalo slik

Slika 1: Skica robota (vir: avtor naloge) .....	4
Slika 2: Petosna robotska roka (vir: spletna stran Robohub) .....	5
Slika 3: Razvojni ploščici Arduino MEGA in Arduino UNO (vir: avtor naloge) .....	7
Slika 4: Bluetooth modul HC-05 (vir: avtor naloge) .....	8
Slika 5: Vezalna shema Bluetooth modulov za nastavljanje parametrov (vir: avtor naloge).....	9
Slika 6: Nastavljanje parametrov za bluetooth modul – suženj (vir: avtor naloge) ....	10
Slika 7: Nastavljanje parametrov za bluetooth modul – gospodar (vir: avtor naloge)	11
Slika 8: Notranjost servo motorja (vir: spletna stran PyroElectro) .....	13
Slika 9: Opazovanje PWM signalov na zaslonu osciloskopa (vir: avtor naloge) .....	14
Slika 10: Pogonski voziček (vir: avtor naloge) .....	16
Slika 11: Enosmerni motor z reduktorjem (vir: spletna stran Prohobi) .....	17
Slika 12: H-mostič (vir: mentor) .....	18
Slika 13: Vezalna shema L293DNE (vir: katalog) .....	19
Slika 14: Časovni potek PWM signala (vir: avtor naloge) .....	21
Slika 15: Prikaz delovanja timerja2 in timerja4 in proženja PWM signala v načinu fast PWM (vir: avtor naloge) .....	24
Slika 16: Usklajeno proženje PWM signala s timerjem2 in timerjem4 (vir: avtor naloge).....	25
Slika 17: Igralna palica (vir: avtor naloge);.....	26
Slika 18: Naš daljinski upravljalnik (vir: avtor naloge) .....	27
Slika 19: Vezava oddajnika (vir: Avtor naloge) .....	28
Slika 20: Vezava sprejemnika (vir: avtor naloge) .....	31
Slika 21: Vezje narisano v programu Eagle (vir: avtor naloge) .....	32
Slika 22: Rezkanje vezja (vir: avtor naloge) .....	33
Slika 23: Sestavljanje pogonskega vozička (vir: avtor naloge) .....	32
Slika 24: Končni izdelek (vir: avtor naloge) .....	33

### 1.3.      **Kazalo tabel**

Tabela 1: Krmiljenje vrtenja levega motorja .....	20
Tabela 2: Krmiljenje vrtenja desnega motorja.....	20
Tabela 3:register TCCRxA.....	23
Tabela 4:register TCCRxB.....	23

## 2. POVZETEK

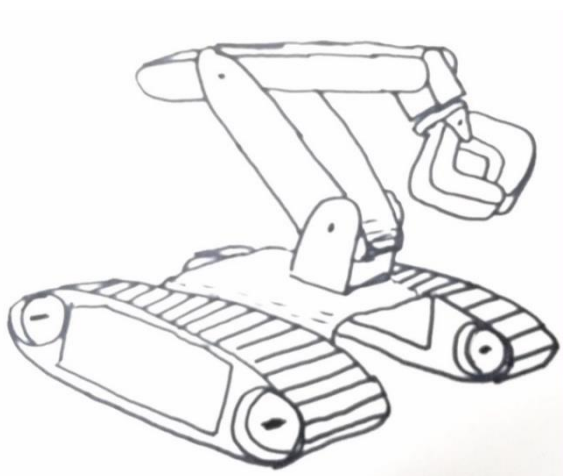
V raziskovalni nalogi bomo raziskovali delovanje in samogradnjo mobilne robotske roke. Najprej bomo pojasnili delovanje posameznih komponent, ki smo jih uporabili za izdelavo raziskovalne naloge, potem bomo obrazložili nastavitve in programiranje Arduino razvojne ploščice na koncu pa bomo prikazali delovanje robotske roke z našimi ugotovitvami.

## 3. ZAHVALA

Zahvaljujemo se mentorju za vso podporo in pomoč ter nasvete pri izdelavi in pisanju raziskovalne naloge. Zahvaljujemo se tudi vsem preostalim, ki so na kakršenkoli način prispevali k uspešnemu zaključku naše raziskovalne naloge.

## 4. UVOD

Ker smo navdušeni nad raziskovanjem na področju elektrotehnike in elektronike, nas je posledično začela zanimati tudi robotika. Ugotovili smo, da bi izdelava robotske roke bila primerna naloga za raziskovanje robotike. Da pa bi projekt bil resničen konstrukcijski izziv, pa smo se odločili za izdelavo mobilne robotske roke, ki bi jo lahko prevažali na daljinsko krmiljenem vozičku. Robotsko roko in voziček pa bi naj upravljali z igralno palico. Na samem začetku snovanja projekta smo si podali določene zahteve kako bi naj robotska roka delovala, sestavili pa smo tudi seznam materiala, ki ga potrebujemo za realizacijo projekta. Za lažjo predstavbo kaj bomo naredili, smo robotsko roko tudi skicirali (slika1).



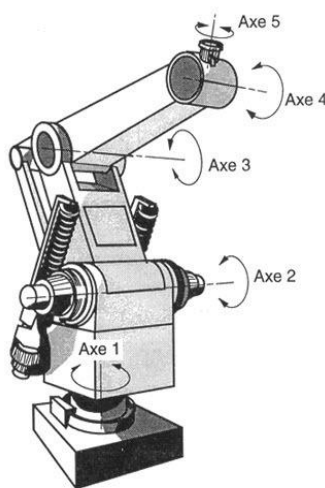
*Slika 1: Skica robota (vir: avtor naloge)*

## 5. VSEBINSKI DEL

V nadaljevanju bomo opisovali sestavne dele naše robotske roke, predvsem tehniške karakteristike komponent, načine priključitve in nastavitve. Omenili bomo tudi vire informacij, ki smo jih uporabili za učenje in spoznavanje komponent.

### 5.1. Robotska roka

Robotska roka, ki jo nameravamo narediti, naj bi bila petosna, kar pomeni, da se pregiba na petih rotacijskih sklepih (glej sliko 2).



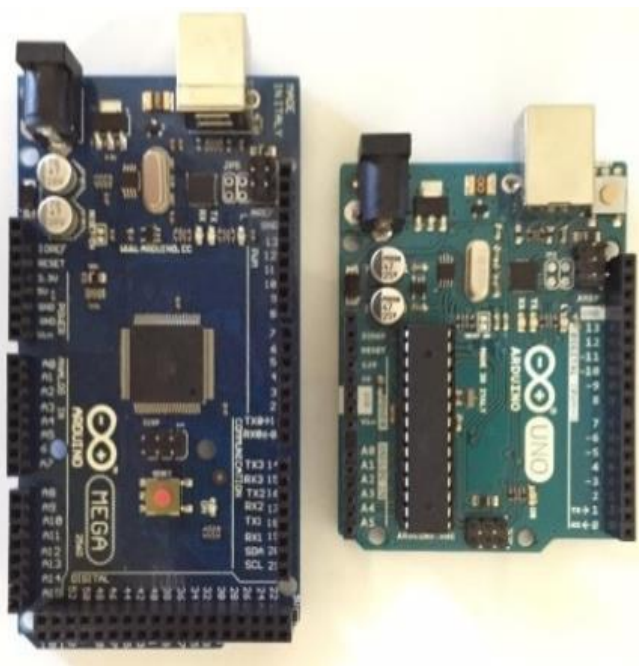
Slika 2: Petosna robotska roka (vir: spletna stran Robohub)

Za premikanje rotacijskih sklepov robota smo uporabili servo motorčke, ki omogočajo natančno pozicioniranje robotskih sklepov, v razponu od 0 do 180 geometrijskih stopinj.

## 5.2. Arduino razvojne ploščice

Arduino je italijansko podjetje, ki se ukvarja z izdelavo in razvojem programirljivih ploščic, katerih delovanje temelji na mikrokontrolniku Atmel. Pri naši raziskovalni nalogi, bomo uporabili dva tipa razvojnih ploščic, to sta Arduino UNO ter Arduino MEGA 2560, ki se medseboj razlikujeta po zmogljivosti in številu vhodnih in izhodnih enot. Prednost razvojnih ploščic je v cenovni dostopnosti, preprosti uporabi, uporabnosti pri najrazličnejših projektih in v enostavni ter pregledni programski kodi, ki jo napišemo v programskem okolju Arduino IDE potem pa jo lahko naknadno spreminjamo in ponovno nalagamo.

Arduino MEGA temelji na 8-bitnem Atmelovem mikrokontrolerju ATmega 2560, Arduino UNO, pa na mikrokontrolerju Atmega328p. Arduino UNO ima 14 digitalnih vhodno-izhodnih (I/O) priključkov; od tega jih 6 podpira PWM signal, prav tako pa ima še 6 analognih vhodov. Arduino MEGA je opremljen s 54-imi digitalnimi vhodno-izhodnimi (I/O) priključki, od teh jih 15 podpira PWM, 16 vhodov pa je analognih.



Slika 3: Razvojni ploščici Arduino MEGA in Arduino UNO (vir: avtor naloge)



Delovni takt pri obeh različicah Arduino ploščic narekuje 16 MHz kristalni oscilator. Delovna napajalna napetost ploščic je 5VDC, na voljo pa je tudi priključek za napajanje z napetostjo 3.3 VDC. Maksimalni izhodni tok na posameznem priključku je 40 mA, celotni izhodni tok pa praviloma ne sme preseči 200 mA. Vgrajena je tudi tokovna omejitev, ki pri toku izhodnem toku nad 500 mA izklopi ploščico. Ko se pa vrednost ponovno zmanjša pod to mejo, se ploščica ponovno aktivira. Mikrokontroler lahko napajamo na več načinov. Lahko ga napajamo s 5V preko vhoda USB na računalniku. Tukaj je maksimalni vhodni tok omejen na 250-500 mA, odvisno od računalnika. Naslednji način napajanja je preko 2.1 mm vtičnice. Tu je zahtevana napetost med 6 V in 20 V, priporočeno napetostno območje pa je med 7 in 12VDC. Med ta vir napetosti in regulator napetosti je nameščena zaščitna dioda za zaščito vezja pred nepravilno priključeno polariteto enosmerne napetosti.

### 5.3. Bluetooth komunikacija

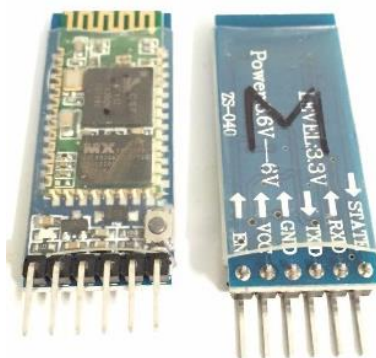
Bluetooth povezava je tehnologija za brezžično komunikacijo elektronskih naprav, ki jo pogosto srečujemo pri mobilnih napravah (bluetooth slušalke, prostoročno telefoniranje, prenos podatkov ipd.).

Pomanjkljivost Bluetooth komunikacije je v tem, da ta povezava med napravami deluje le do razdalje nekaj metrov. Ena izmed prednosti pa je ta, da Bluetooth naprave lahko zaščitimo z varnostnim geslom, tako da je prenos podatkov omogočen samo določenim napravam ki so »vidne« napravi z Bluetooth vmesnikom.

Bluetooth komunikacija poteka v frekvenčnem pasu med 2,4 do 2,8 GHz.

#### 5.3.1. Bluetooth modul HC-05

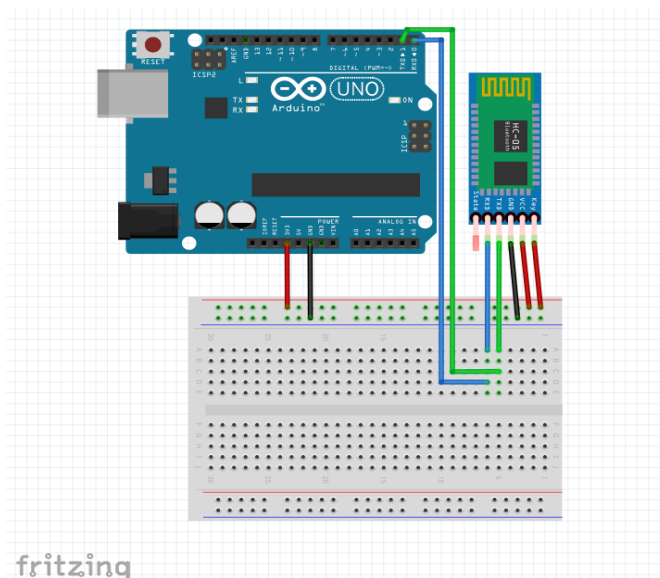
V naši raziskovalni nalogi sva uporabila bluetooth modul z oznako HC-05, katerega delovanje temelji na BC417 integriranem vezju, ki ima 8 MB »flash« spomina ter funkcijo Master/Slave (»gospodar/suženj«, po slovensko) . Omenjena funkcija je komunikacijski protokol pri katerem ena od naprav, ki je nastavljena kot gospodar, oddaja ukaze drugi napravi ali skupini električnih naprav ki so v »podrejenem položaju«.



Slika 4: Bluetooth modul HC-05 (vir: avtor naloge)

## 5.4. Povezovanje bluetooth modulov

Ker je idejna zasnova naše naloge zahtevala, da dva modula HC-05 delujeta v paru, smo povezovalnima moduloma morali nastaviti parametre tako, da se bosta samodejno povezovala ko bosta v dosegu Bluetooth signala. Vezavo za nastavljanje parametrov Bluetooth modulov prikazuje naslednja slika:



Slika 5: Vezalna shema Bluetooth modulov za nastavljanje parametrov (vir: avtor naloge)

Pri tej vezavi smo uporabili naslednje priključke oz. povezave:

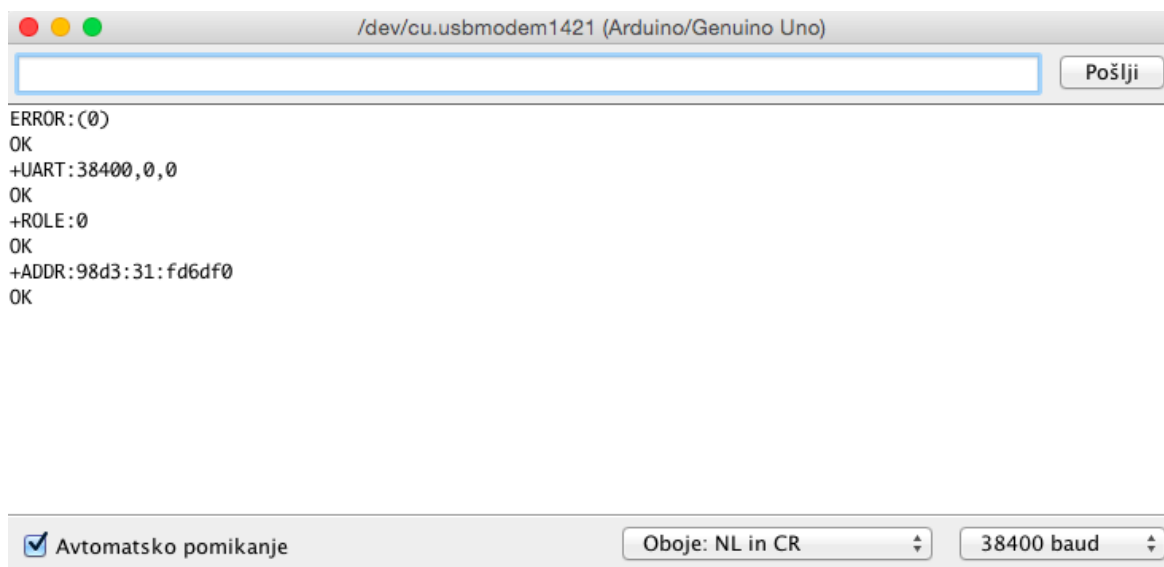
- Vcc – 3,3 V
- EN (ENABLE) – 3,3 V
- GND – potencial 0 V (»masa«)
- TX priključek modula na TX priključek Arduino ploščice
- RX priključek modula na RX priključek Arduino ploščice

Po priključitvi napajalne napetosti na Arduino ploščico je začela utripati signalna lučka v dvosekundnem intervalu, kar je pomenilo, da povezava deluje in da lahko začnemo z nastavljanjem parametrov preko računalniške USB povezave in programa zapisanega v računalniškem programskem okolju.

Nastavljanje Bluetooth modula smo uredili z AT-ukazi v programskem okolju za programiranje Arduino razvojnih ploščic. V programski vrstici se AT-ukazi začnejo z besedico AT+, nato pa vstavimo ustrezeni parameter. Pri tem velja, da če na koncu ukazne vrstice napišemo znak »?«, se nam v nadaljevanju izpiše trenutna vrednost parametra, če pa zanko zaključimo z enačajem in poljubno vrednostjo, pa lahko nastavimo novo vrednost parametra.

Modul smo nastavili v režim delovanja »suženj«, za to ureditev pa smo v programu uporabili naslednje parametre:

- +UART? – izpiše se hitrost prenosa podatkov. Hitrost prenosa podatkov v našem primeru je nastavljena na 38400 bit/s
- +ROLE ?– izpiše se 0 (suženj) ali 1 (gospodar)
- +ADDR? – izpiše se nam bluetooth naslov naprave, ki si ga moramo zabeležiti za nadaljnjo uporabo v programiranju ploščice

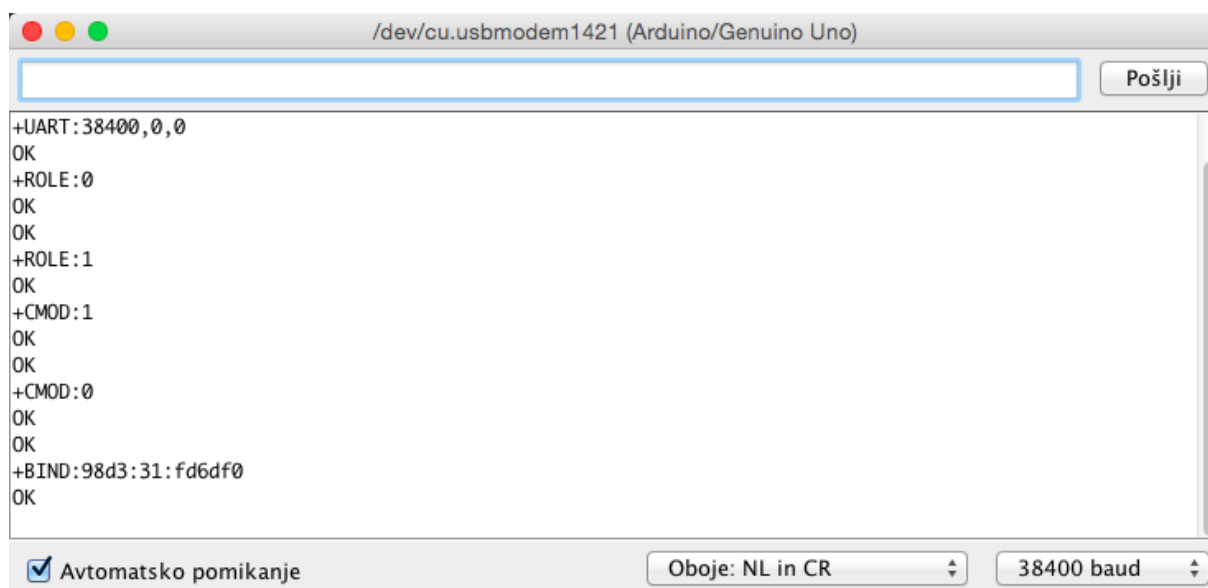


Slika 6: Nastavljanje parametrov za bluetooth modul – suženj (vir: avtor naloge)

Ko smo uspešno končali z nastavitvami prvega modula smo se lotili nastavljanja drugega modula.

Najprej smo preverili, kakšna je nastavljena hitrost prenosa podatkov pri drugem modulu, da smo lahko zagotovili sinhronizirano delovanje s prvim modulom, s podatkovno hitrostjo 38400 Bit/s. Drugi modul smo nastavili na delovanje v režimu »gospodar«. Uporabili smo naslednje parametre:

- +ROLE=1 – nastavili smo modul, kot gospodar
- +CMODE? – izpiše se nam število 1 (naprava se lahko poveže s katero koli drugo napravo), ali 0 (naprava je omejena na povezavo s samo eno napravo)
- +CMODE=0 – omejili smo povezovanje naprave
- +BIND=98d3,31,fd6df0 – vnesemo naslov druge naprave s katero nam se prva naprava potem poveže



Slika 7: Nastavljanje parametrov za bluetooth modul – gospodar (vir: avtor naloge)

Po koncu urejanja zgoraj zapisanih nastavitev modulov, smo aktivirali priključka TX in RX na Bluetooth modulu pa tudi na Arduino ploščici, tako da smo vzpostavili naslednjo električno povezavo:

- Vcc – napajalna napetost, ki znaša 3,3 V
- GND – potencial 0 V (»masa«)
- TX – priključek je za sprejemanje podatkov, zato smo ta priključek priključili na RX priključek na ploščici Arduino, ki podatke oddaja.
- RX – priključimo, na priključek TX razvojne ploščice, tako da Bluetooth modul posreduje podatke ploščici Arduino.

Ob uspešni povezavi modulov z Arduino ploščico, sta signalni lučki začeli utripati v polsekundnem intervalu, ko pa sta modula zaznala še vzpostavitev Bluetooth povezave, pa sta začela utripati v usklajenem ritmu z dvojnimi hitrimi utripi, oz. v dvosekundnem intervalu.

V nadaljevanju urejanja nastavitev smo preverili kako je s preostalima priključkoma STATE ter EN, ki sta ostala neuporabljena.

Signal iz priključka STATE (»stanje«), nam pove ali je povezava vzpostavljena ali ne, to pa nam povesta tudi indikatorski lučki oz. interval njunega utripanja.

Priključek EN (ENABLE) oz. »priključek za omogočanje« se uporablja samo kot vhodna enota naprave za nastavitve parametrov. Ta priključek nismo uporabili.

## 5.5. Servo motorji

Servo motorji so posebno konstruirani enosmerni motorji, ki omogočajo natančno postavitev oz. spreminjanje položaja rotorja. Imajo široko področje uporabe, npr. pri daljinsko vodenih avtomobilčkih, dandanes pa tudi v osebnih električnih avtomobilih. V industriji so nepogrešljivi, saj omogočajo hitre in zelo natančne premike. Odlikuje jih konstantni navor skozi celotno območje vrtljajev. V naši raziskovalni nalogi jih bomo uporabili za premikanje rotacijskih sklepov robotske roke.

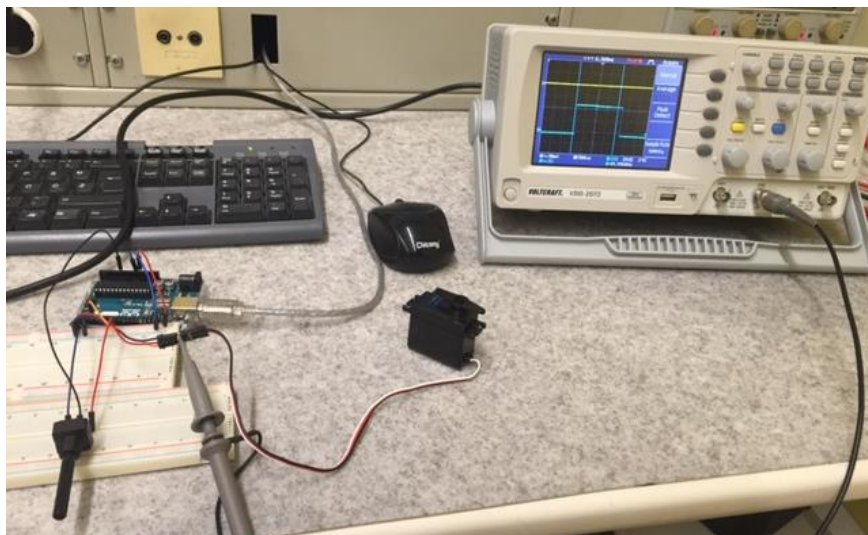
Servo motorji imajo zelo visoke vrtljaje in malo navora na gredi zato jih izdelujejo z zobniškimi prenosi oz. reduktorji za povečanje vrtilnega momenta.



*Slika 8: Notranjost servo motorja (vir: spletna stran PyroElectro)*

V motorju je integrirano elektronsko vezje na tiskani ploščici, s senzorjem položaja. Ta senzor, ki je po navadi kar potenciometer, zaznava za koliko se je zavrtela gred motorja. Gred motorja se lahko premika v obsegu od  $0^\circ$  do  $180^\circ$ , oziroma od  $-90^\circ$  do  $+90^\circ$ . Pozicija gredi motorja, pa je odvisna od dolžine električnega impulza na priključku za signal. Frekvenca signala za servo motor je po navadi okoli 50-60 Hz, impulzi pa so pravokotne oblike, skratka položaj določamo s PWM signalom.

Časovna dolžina impulzov PWM signala znaša 0,7 ms, postavi pa gred motorja na -90°, 1,5 ms dolgi impulzi pa na 0°, impulzi dolgi 2 ms pa postavijo gred servo motorja na pozicijo +90°. Zadevo smo najprej preizkusili tako, da smo z Arduino ploščico krmilili servo motor, obliko in velikost PWM signala pa smo opazovali na osciloskopu.



*Slika 9: Opazovanje PWM signalov na zaslonu osciloskopa (vir: avtor naloge)*



### 5.5.1. Krmiljenje servo motorjev z Arduino ploščico

Za generiranje PWM signala smo uporabili knjižnico z naborom ukazov, tako da nam ni bilo treba dodatno nastavljati vrednosti časovnikov, kar bi nam odvzelo precej časa. Knjižnico smo uvozili v program po naslednjem postopku:

```
#include <Servo.h> // uvoz knjižnice za servo motorje
```

Vsak servo motor smo morali poimenovati in mu določiti priključek na katerega bo priključen:

```
Servo mojServo; // poimenovanje našega servo motorja  
mojServo.attach(priključek); // PWM priključek servo motorja
```

Nato smo v program vnesli spremenljivko, ki jo bomo poimenovali s celim številom med 0 in 180, v bistvu pa gre za spremenljivko, ki podaja kot v geometrijskih stopinjah, s katerimi potem določamo pozicijo gredi servo motorja.

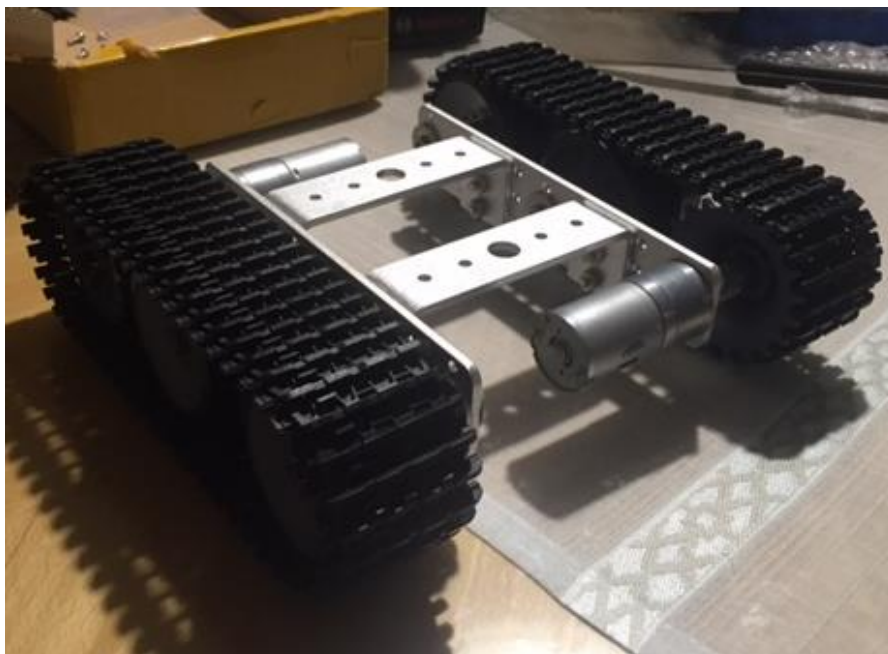
```
mojServo.write(int kot); //določanje pozicije servo  
motorja
```

## 5.6.      **Robotska roka na pogonskem vozičku**

Pogonski voziček smo si zamislili kot nosilno ploščad s parom gumijastih gosenic katere premikanje omogočata dva enosmerna motorčka ki, krmiljena neodvisno eden od drugega, poganjata gosenici. Predvidevali smo, da bomo na ta način enostavno spreminjali smer vožnje ob naslednjih predpostavkah:

- če se obe gosenici premikata naprej z enako hitrostjo, se voziček pomika naravnost naprej, podobno velja za vožnjo nazaj.
- če se leva gosenica premika naprej hitreje, kot desna, ali če slednja stoji, se bo vozilo obračalo v desno.
- če leva gosenica stoji oziroma se giblje počasneje, kot desna, se bo vozilo pomikalo v levo.

Pogonski voziček smo naročili v spletni trgovini, kupljeni komplet pa je vseboval tudi pogonske motorje z reduktorji.



*Slika 10: Pogonski voziček (vir: avtor naloge)*

### 5.6.1. Enosmerni motorji z reduktorjem

Enosmerni motor je električni stroj ki pretvarja električno energijo v mehansko. Zgrajen je iz mirujočega, statorskega železnega jedra, na katerem se nahaja vzbujačno navitje za ustvarjanje magnetnega polja. Med magnetnimi poli statorja se nahaja rotor z navitjem, povezanim preko komutatorja in grafitnih ščetk ki so priključene na vir enosmerne napetosti. Če zamenjamo polariteto priključene enosmerne napetosti, motorček spremeni smer vrtenja.

Za zagotovitev večjega navora oz. boljših zmogljivosti motorskega pogona smo izbrali enosmerne motorje z reduktorji.

Reduktorji so gonila, ki zmanjšujejo število obratov na izhodu, s tem pa povečajo navor na gredi motorja.



Slika 11: Enosmerni motor z reduktorjem (vir: spletna stran Prohobi)

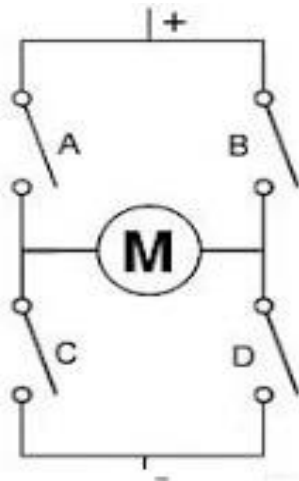
### 5.6.2. Krmiljenje DC motorčkov

Mostično vezje za krmiljenje enosmernega motorja imenujemo H-mostič.

Slika 12 nam prikazuje principiarno vezalno shemo vezja H-mostiča, s štirimi stikali, s katerimi si ponazorimo način spreminjanja smeri vrtenja enosmernega motorja.

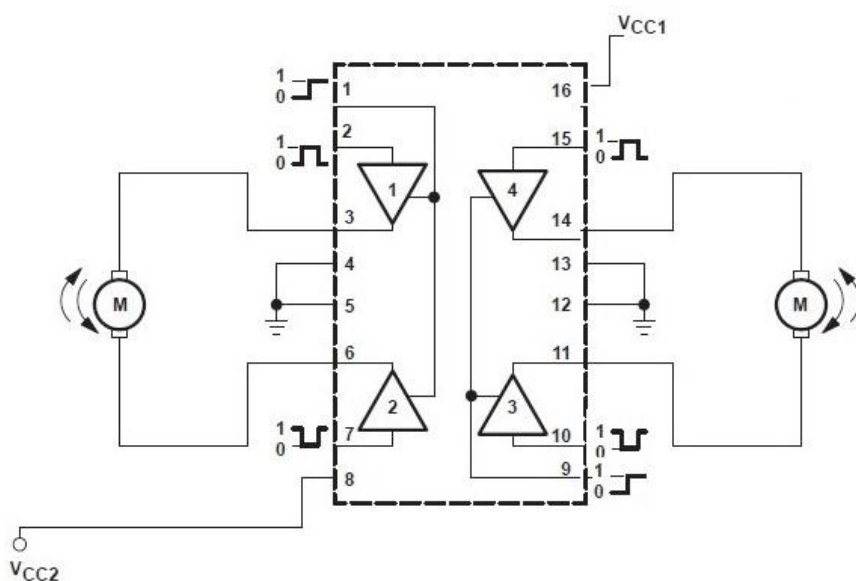
Opis delovanja H –mostiča:

- V primeru če so vsa stikala izklopljen, motor ne dobi el. energije in se ne bo vrtel
- ob vklopu stikal A in D, se motorček vrti v smeri urinega kazalca,
- Pri vklopu stikal B ter C, se motorček vrti v nasprotni smeri urinega kazalca,
- Pri vklopu stikal A ter B, se bo vrtenje motorčka ustavilo,
- Če, pa vklopimo vsa stikala istočasno, pa je vezje v kratkem stiku.



Slika 12: H-mostič (vir: mentor)

V naši raziskovalni nalogi bomo namesto mehanskih stikal uporabili tranzistorska stikala integrirana v integriranem vezju L293DNE, saj vemo da so tranzistorji elektronska stikala, ki jih lahko upravljamo z električno napetostjo. Obenem pa ima integrirano vezje L293DNE dodatne funkcije, kot so reguliranje hitrosti samih motorčkov, preprečevanje kratkega stika v vezju ter pregrevanja, vgrajene pa so tudi hitre diode, ki ščitijo vezje pred lastnimi induciranimi napetostmi navitij motorčkov ob izklopih. Integrirano vezje vsebuje dva H-mostiča, kar pomeni, da lahko krmilimo dva enosmerna motorja hkrati, vendar pa motor mora imeti ustrezno moč da njegov obratovalni tok ne presega 1 A, napetost pa ne sme presegati 36 V.



Slika 13: Vezalna shema L293DNE (vir: katalog)

Integrirano vezje ima 16 priključkov, ki jih povežemo na predpisan način:

**Priključke 4, 5, 12 in 13**, priključimo na »maso« elektronskega vezja.

**Priključek 16 (Vcc1)**, priključimo na napetost 5 V za interno napajanje vezja,

**Priključek 8 (Vcc2)**, je namenjen priključitvi maksimalne napetosti za enosmerne motorčke, v našem primeru je to 9 V

**Priključka 1 in 9** sta »pina« za omogočanje (»enable«), vklopa motorjev, zato smo ta dva pina povezali z Arduino ploščico, da smo omogočili pošiljanje PWM signala za krmiljenje hitrosti motorjev.

**Priključki 3 in 6 ter 11 in 14**, so izhodni priključki, kamor priklopimo motorčke.

**Priključki 2 in 7 ter 10 in 15**, so vhodni priključki, s katerimi določamo, v katero smer se bo vrtel motor. S priključkoma 2 in 7, določimo smer vrtenja za levi motor, s priključkoma 10 in 15, pa za desni motor, in sicer je to odvisno od tega na katere priključke priključimo napajalno napetost 5 V, oziroma logično 1. Te priključke smo povezali z digitalnimi izhodnimi priključki Arduino ploščice.

V nadaljevanju je preglednica s seznamom priključkov in opisom funkcije:

*Tabela 1: Krmiljenje vrtenja levega motorja*

PRIKLJUČEK 2	PRIKLJUČEK 7	VRTENJE MOTORJA
0	1	Vrtenje v smeri urinega kazalca
1	0	Vrtenje v nasprotni smeri urinega kazalca
0	0	Motor stoji
1	1	Motor stoji

*Tabela 2: Krmiljenje vrtenja desnega motorja*

PRIKLJUČEK 10	PRIKLJUČEK 15	VRTENJE MOTORJA
0	1	Vrtenje v nasprotni smeri urinega kazalca
1	0	Vrtenje v smeri urinega kazalca
0	0	Motor stoji
1	1	Motor stoji

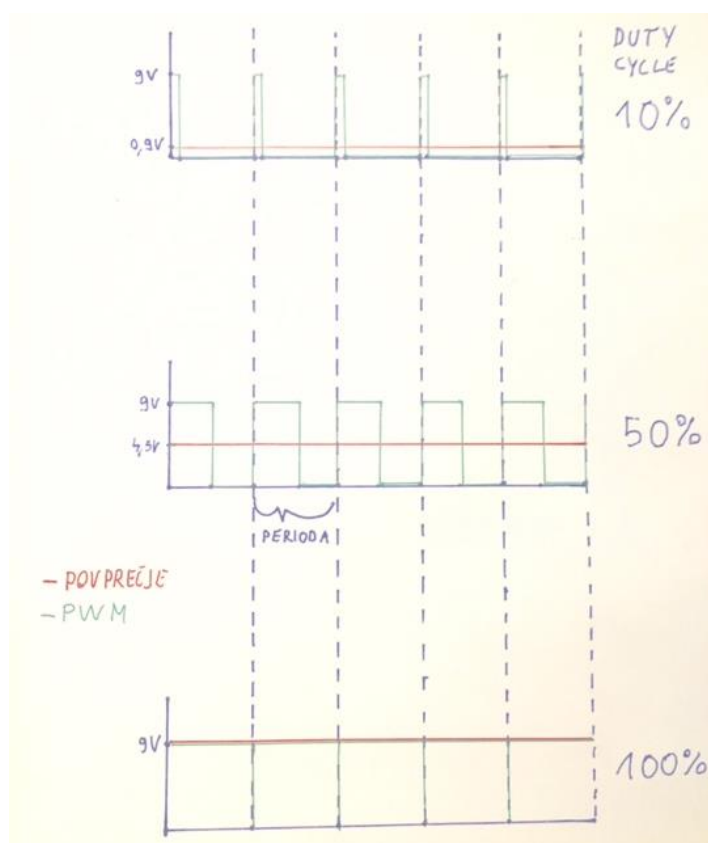
### 5.6.3. PWM signal za krmiljenje hitrosti DC motorja

PWM ali pulzno širinska modulacija (ang. Pulse Width Modulation), je generiranje električnega signala pravokotne oblike, s katerim je možno krmiliti hitrost enosmernega motorčka.

Če pogledamo sam signal na časovnem diagramu (na sliki 14) vidimo, da je sestavljen iz ponavljajočih se impulzov. Razmerje med impulzi ter periodami imenujemo »duty cycle« ali obratovalni cikel, ki ga izražamo v odstotkih, odstotke pa lahko enačimo s povprečno vrednostjo napetosti signala. Na ta način lahko krmilimo hitrost vrtenja motorčka.

Kot primer označevanja bomo navedli PWM signal z maksimalno napetostjo 9 V:

- duty cycle 10%  $\rightarrow$  0,9 V,
- duty cycle 50%  $\rightarrow$  4,5 V,
- duty cycle 100%  $\rightarrow$  9 V



Slika 14: Časovni potek PWM signala (vir: avtor naloge)

#### 5.6.4. Proženje PWM signalov s ploščico Arduino MEGA za krmiljenje hitrosti DC motorčka

Kot smo že uvodoma omenili, na vhodu ploščice Arduino MEGA imamo 15 priključkov, ki podpirajo PWM signal. PWM signal prožijo časovniki ali ang. »timerji«. Arduino MEGA vsebuje 6 časovnikov: timer0, timer1, timer2, timer3, timer4, timer5.

Vsak časovnik je namenjen dvema ali trem priključkom na razvojni ploščici Arduino. Časovniki imajo naslednje lastnosti:

- Priključka 4 in 13: krmiljena s časovnikom 0, 8-bitni časovnik;
- Priključka 11 in 12: krmiljena s časovnikom 1, 16-bitni časovnik;
- Priključka 9 in 10: krmiljena s časovnikom 2, 8-bitni časovnik;
- Priključki 2, 3 in 5: krmiljeni s časovnikom 3, 16-bitni časovnik;
- Priključka 6, 7 in 8: krmiljeni s časovnikom 4, 16-bitni časovnik;
- Priključka 44, 45 in 46: krmiljeni s časovnikom 5, 16-bitni časovnik;

PWM signal je najenostavneje sprožiti s pomočjo že obstoječih funkcij v programskem okolju Arduino IDE, to sta »analogWrite()« in »tone()«.

```
analogWrite(priključek, 8-bitna_vrednost);
```

Funkcija »analogWrite« nam omogoča, da prožimo PWM signal pri frekvenci okoli 500 Hz, kar je uporabno predvsem kadar želimo nastavljanje svetilnosti svetlobnih virov, npr. LED diode, kajti človeško oko takšnih frekvenc utripanja svetlobe ne zazna, vendar pa je ta frekvenca za krmiljenje motorja nekoliko previsoka za radi induktivnega značaja motorja kot bremena. Obratovalni ali »duty« cikel s to funkcijo nastavljam z 8-bitno vrednostjo, kar pomeni od 0 do 255, seveda pri frekvenci 500 Hz. Duty cycle za 8-bitno vrednost izračunamo po naslednji formuli:

$$\text{Osem bitna vrednost} = \frac{255 \times \text{duty cycle}(\%)}{100\%}$$

```
tone(priključek, frekvenca);
```

Funkcija »tone()«, ima obratne lastnosti, kot funkcija »analogWrite«, generira pa signale pravokotne oblike, katerim lahko spreminjamo frekvenco, vendar pa ne moremo spreminjati duty cycle, ki je konstantno 50%. Sicer to funkcijo najpogosteje uporabljamo za generiranje tonov za Piezzo zvočnik.



Ukvarjajoč se s temi nastavitvami, smo ugotovili, da nam ti dve funkciji ne bosta ravno najbolje koristili za krmiljenje hitrosti motorčka, zato smo zadevo morali nastaviti »ročno«. Vrednost registrov v časovnikih, ki smo jih uporabili za proženje PWM signala, smo sami določili tako da smo lahko nastavljali frekvenco in duty cycle.

Uporabili smo »enable« priključke modula L293DNE (za krmiljenje hitrosti), in sicer priključka 8 in 9, potem še dva časovnika za ta dva priključka, »timer4« (priključek 8 (OC4C)), ki je 16-bitni in »timer2« (priključek 9 (OC2B)), ki pa je 8-bitni. Časovniki v razvojni ploščici delujejo tako da povečujejo svojo vrednost od 0 do vnaprej določene vrednosti, npr. 8-bitni časovniki od 0 do 255, 16-bitni časovniki pa od 0 do 65535. Časovniki povečujejo vrednosti po korakih od 0,0625  $\mu$ s, saj delovni takt narekuje 16MHz kristalni oscilator, katerega perioda signala je 0,0625  $\mu$ s.

$$T = \frac{1}{f} = \frac{1}{16\,000\,000\,Hz} = 0,0625\,\mu s$$

Delovanje samih časovnikov lahko tudi spreminjamo, npr. tako da upočasnimo delovanje časovnika z nastavitvijo periode oscilatorja ipd. Za delovanje vsakega časovnika skrbi več registrov, najbolj pomembna pa sta 8-bitna registra TCCRxA in TCCRxB (»x« predstavlja zaporedno število časovnika od 0 do 5), torej za timer4 TCCR4A in TCCR4B ter za timer2 TCCR2A in TCCR2B.

Tabela 3: register TCCRxA

COMxA1	COMxA0	COMxB1	COMxB0	FOCxA	FOCxB	WGMx1	WGMx0
--------	--------	--------	--------	-------	-------	-------	-------

Tabela 4: register TCCRxB

ICNCx	ICESx	-	WGMx3	WGMx2	CSx2	CSx1	CSx0
-------	-------	---	-------	-------	------	------	------

PWM signal pa lahko s časovniki prožimo tudi na več načinov, najbolj pogost postopek je proženje s »fast PWM«, ki se je nam zdel tudi najlažji. S spreminjanjem bitov WGMx2, WGMx1 ter WGMx0, spremenimo tudi delovanje časovnikov. Če želimo ohraniti način »fast PWM«, moramo postaviti bite na naslednje vrednosti:

- WGMx2 = 0, (za timer4, WGM42 = 0; za timer2, WGM22 = 0)
- WGMx1 = 1, (za timer4, WGM41 = 1; za timer2, WGM21 = 1)
- WGMx0 = 1, (za timer4, WGM40 = 1; za timer2, WGM20 = 1)

Modulacija »fast PWM« deluje tako, da se vrednost 8-bitnega časovnika povečuje od 0 do 255, to pa pri 16-bitnem časovniku pomeni do vrednosti 65535. Z registri OCRxy (kjer sta x številka časovnika in y A, B ali C, odvisno pač od tega na katerem priključku želimo imeti PWM izhod), določimo vrednost, pri kateri se bo izhodni priključek PWM signala postavil, na 1. Kadar pa časovnik doseže svojo maksimalno vrednost, se izhodni priključek resetira oz. postavi nazaj na vrednost 0.

Za priključek 8 (ali OC4C) je register OCRxy = OCR4C,

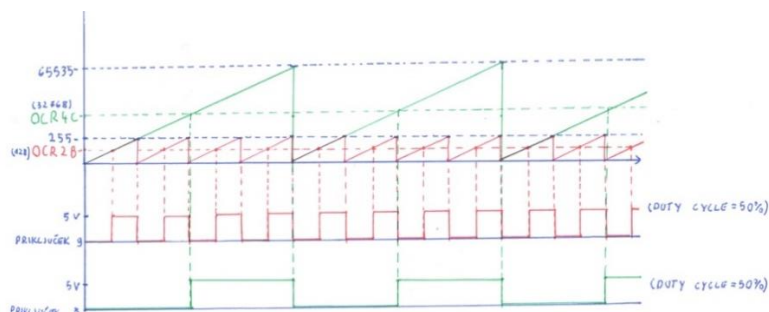
Za priključek 9 (ali OC2B) je register OCRxy = OCR2B.

Če želimo npr. imeti 50 odstotni duty cycle, moramo vrednosti registrov nastaviti na polovico maksimalne vrednosti timerjev, se pravi  $OCR4C = 32768$ , ter  $OCR2B = 128$ . Vrednost registrov OCRxy, pa za nek željen duty cycle izračunamo po naslednji enačbi:

$$OCR_{xy} = 2^{n-bitni\ časovnik} \times duty\ cycle$$

$$OCR4C = 2^{16} \times 50\% = 65536 \times 50\% = 32768$$

$$OCR2B = 2^8 \times 50\% = 256 \times 50\% = 128$$



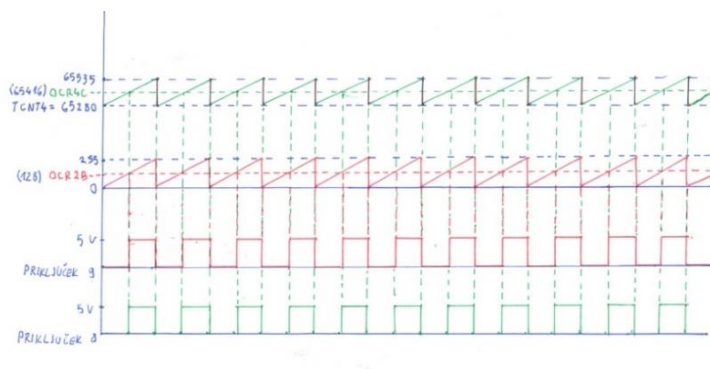
Slika 15: Prikaz delovanja timerja2 in timerja4 in proženja PWM signala v načinu fast PWM (vir: avtor naloge)

Kot lahko vidimo na zgornjem grafu (slika 15) je duty cycle vrednosti 50% na obeh priključkih 8 in 9, čeprav imamo različni frekvenci signala. Na grafu je sicer prikazana minimalna razlika med frekvencama, vendar pa je razlika v realnosti med njima 256-kratna. Naslednji korak v nastavljanju delovanja vezja je da 16-bitnemu časovniku, določimo začetno vrednost, tako da bosta časovnika usklajeno dosegla prekoračitev in tako usklajeno prožila PWM signal. Začetno vrednost časovnika določa 16-bitni register TCNTx (x je število časovnika, v našem primeru 4, zato bomo nastavljali register TCNT4).

Če želimo, da pride do prekoračitve po 255 korakih, moramo nastaviti TCNT4 na vrednost 65 280, saj je njegova maksimalna vrednost 65 535.

$$65\,535 - 65\,280 = 255$$

V nadaljevanju smo morali spremeniti vrednost registra OCR4C, ki postavlja izhodni priključek na logično vrednost 1. To storimo tako, da rezultat, ki smo ga dobili pri formuli za izračun vrednosti registra OCR2B (v našem primeru 128), prištejemo k novi začetni vrednosti (v našem primeru 65 280), torej  $65\,280 + 128 = 65\,416$ . Sedaj nastavimo register OCR4C na vrednost 65 416.



Slika 16: Usklajeno proženje PWM signala s timerjem2 in timerjem4 (vir: avtor naloge)

Sedaj je frekvenca PWM signala na priključku 9 nastavljena tako, da je enaka frekvenci PWM signala na priključku 8. Njuno sedanjo frekvenco PWM signala lahko izračunamo po sledeči enačbi:

$$f_{PWM} = \frac{f_{oscilatorja}}{256} = \frac{16\,000\,000\,Hz}{256} = 62\,500\,Hz$$

Vidimo, da je dobljena frekvenca signala zelo visoka in ni primerna za krmiljenje motorčka (induktivno breme).

Frekvenco PWM signala zmanjšamo tako, da omejimo povečevanje vrednosti po določenih korakih za vrednost 1. To naredimo s ti. »preddelilniki«, ki omogočajo, da se vrednost časovnika poveča za vrednost ena ob vsaki 8., 32., 64., 128., 256. ali 1024. periodi oscilatorja. Vrednost preddelilnika določamo z biti CSx0, CSx1 in CSx2 v TCCRxB registru. Ker želimo preddelilnik nastaviti na 1024, moramo vrednosti vseh treh bitov (CSx0, CSx1 in CSx2), postaviti na 1. Sedaj je frekvenca signala:

$$f_{PWM} = \frac{f_{oscilatorja}}{256 \times \text{preddelilnik}} = \frac{16\,000\,000\,Hz}{256 \times 1024} \approx 61\,Hz$$

### 5.7. Igralna palica, ang. »Joystick«

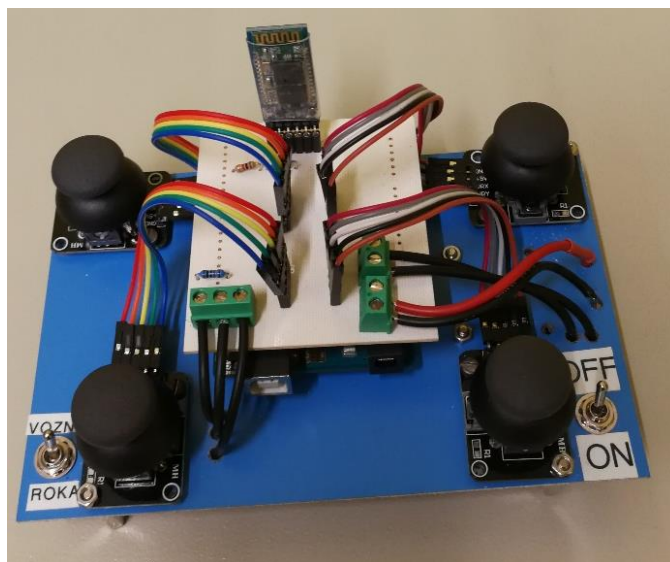
Za izdelavo daljinskega upravljalnika smo najprej raziskovali delovanje igralne konzole Play Station 3. Ugotovitve so nam pomagale pri snovanju našega daljinskega upravljalnika. Uporabili smo štiri igralne palice, ki smo jih namestili na primerno podlago za udobno upravljanje robotske roke. Potem smo izdelali elektronski načrt za daljinski upravljalnik oz. igralne palice. Obliko izbranih igralnih palic lahko vidimo na sliki 7. Ta igralna palica ima pet priključkov, in sicer: GND in V+5, na katera priključimo napajalno napetost 5 V; VRX ter VRY, preko katerih bomo lahko odčitavali položaj igralne palice ter še priključek SW, ki je angleška kratica za »switch« oz. stikalo.



*Slika 17: Igralna palica (vir: avtor naloge);*

Igralna palica ima ergonomsko obliko in jo lahko z lahkoto upravljamo z roko oz. prsti, tako da premikamo »gobico« v katerokoli smer.

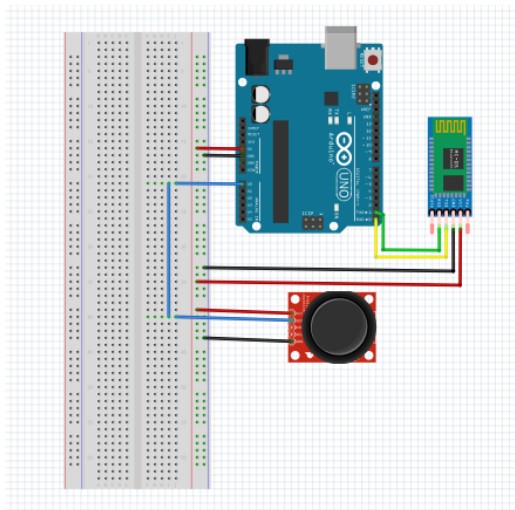
Igralne palice smo, skupaj z Arduino ploščico, pritrdili na prilagojeno nosilno ploščico. Ob konstruiranju ploščice smo ugotovili da nam Arduino ploščica omogoča branje podatkov iz največ šestih potenciometrov, mi pa smo jih potrebovali osem. Problem smo rešili tako, da smo v vezje dodali preklopno stikalo s katerim lahko izbiramo med dvema načinoma delovanja igralne palice kot sta »voznja« ali »upravljanje roke«.



*Slika 18: Naš daljinski upravljalnik (vir: avtor naloge)*

## 5.8. Daljinsko upravljanje servo motorja z igralno palico preko bluetooth povezave

Vezje s katerim omogočamo brezžično Bluetooth povezavo naredimo tako, kot kaže slika 16. Priključek enega izmed dveh potenciometrov na igralni palici smo priključili z analognim vhodom z oznako A0.



Slika 19: Vezava oddajnika (vir: Avtor naloge)

V programu smo najprej poimenovali vse spremenljivke, ki jih bomo uporabili. Spremenljivka »joystick« je rezervirana za vrednosti med 0 in 1023, ki jo spreminjamo s spreminjanjem položaja igralne palice.

```
int joystick; //vrednost, joysticka od 0 do 1023
int podatek; //podatek, ki ga bomo pošiljali preko BT signala
```

V funkciji void.setup() določimo, vhodni priključek, ki se nahaja na A0, kjer je priključen potenciometer. Prav tako zaženemo serijsko komunikacijo med Arduino ploščico in Bluetooth modulom.

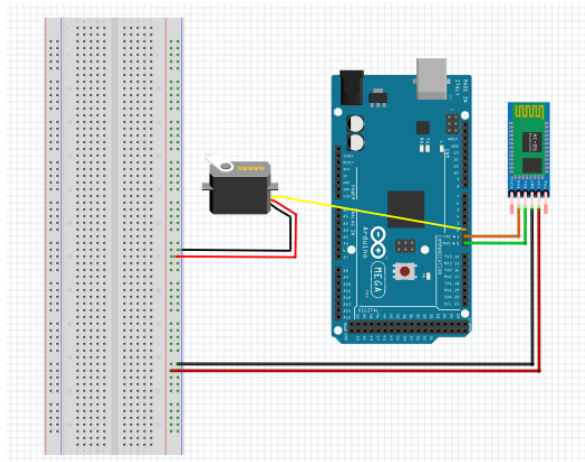
```
void.setup() {
  pinMode (A0, INPUT); //priključek A0, je vhodni priključe
  Serial.begin(38400); //začetek serijske komunikacije
}
```

V funkciji void.loop(), se izvaja glavni program:

```
void.loop() {
```

```
joystick = analogRead(A0);  
//Vrednost joysticka se shrani v spremenljivki joystick  
if (joystick>700) {  
    podatek = podatek + 1;  
    //Če je vrednost joysticka, več od 700, se podatek povečuje za  
    1  
  
}  
else if (joystick<300) {  
    podatek = podatek - 1;  
    //Če je vrednost joysticka, manj od 300, se podatek zmanjšuje  
    za 1  
  
}  
if (podatek>179) {  
    podatek = 180;  
    //Če je podatek več, kot 179, naj ne prekorači vrednosti 180  
  
}  
if (podatek<1) {  
    podatek = 0;  
    //Če je podatek manj od 1, je enak 0 (ne negativno število)  
  
}  
Serial.println(podatek); //arduino pošlje podatek BT modulu  
}
```

Vezavo sprejemnega modula prikazuje slika 20. Iz slike je razvidno, da imamo povezan Bluetooth modul na Arduino ploščico v funkciji sprejemnika signala. Priključki servo motorja za sprejemanje PWM signala pa so povezani z digitalnim priključkom številka 2.



Slika 20: Vezava sprejemnika (vir: avtor naloge)

Poimenovati smo morali še vse spremenljivke, ki jih bomo uporabili, podobno kot v prejšnjem primeru. Poimenovali smo tudi spremenljivko, ki nam bo služila za shranjevanje podatkov, prejetih preko Bluetooth povezave. Za pravilno delovanje sistema smo morali uvoziti še knjižnico za servo motor. Servo motor smo poimenovali »mojservo«.

```
int podatek;    //podatek, ki ga bomo prejeli preko BT signala  
  
#include <Servo.h> //uvozimo knjižnico za servo motorje  
  
Servo mojservo; //poimenujemo servo motor
```

V funkciji void.setup() določimo, na katerem mestu je priključek za PWM signal servo motorja (v našem primeru 2), ter zaženemo serijsko komunikacijo med Arduino ploščico in BT modulom.

```
void.setup() {  
  
mojservo.attach(2); //servo je priključen na priključek 2  
  
Serial.begin(38400); //začetek serijske komunikacije  
  
}
```

V funkciji void.loop(), se izvaja glavni program, ki ga besedno opišemo z besedami: če bluetooth modul pošlje podatek Arduino ploščici, se ta podatek shrani v bazi spremenljivk podatkov. Ta podatek je vrednost, ki določa položaj gredi servo motorja.



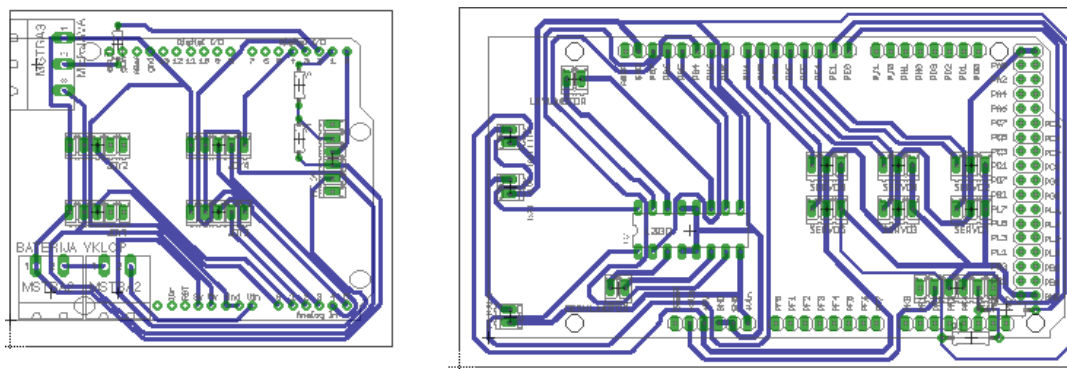
```

void.loop() {
  if (Serial.available>1) {
    podatek = Serial.parseInt;
  }
  mojservo.write(podatek);
}

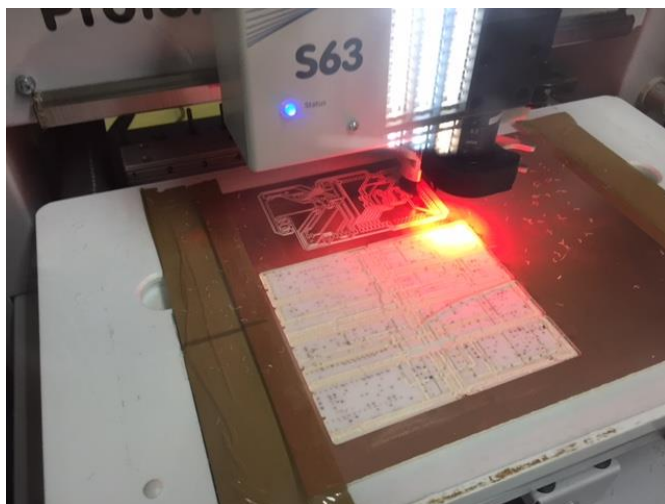
```

## 6. POTEK IZDELAVE PROJEKTA

Največ informacij za izdelavo robotske roke smo dobili na spletu, predvsem od proizvajalcev sestavnih delov projekta (servo motorji, bluetooth moduli,...) Raziskovali smo tudi delovanje podobnih že izdelanih projektov. Potrebne komponente za sestavljanje robota smo naročili po spletu, delovanje le-teh pa smo najprej preizkušali na eksperimentalnih ploščicah, nato pa smo izdelali vezje za daljinski upravljalnik in za robotsko roko v programu za načrtovanje elektronskih vezij Eagle. Končna vezja na elektronskih ploščicah smo naredili z rezkalnikom elektronskih ploščic. Na ta način smo se izognili jedkanju vezja in smo tako prispevali k manjšemu obremenjevanju našega naravnega okolja. (slika 22)



Slika 21: Vezje narisano v programu Eagle (vir: avtor naloge)



*Slika 22: Rezkanje vezja (vir: avtor naloge)*

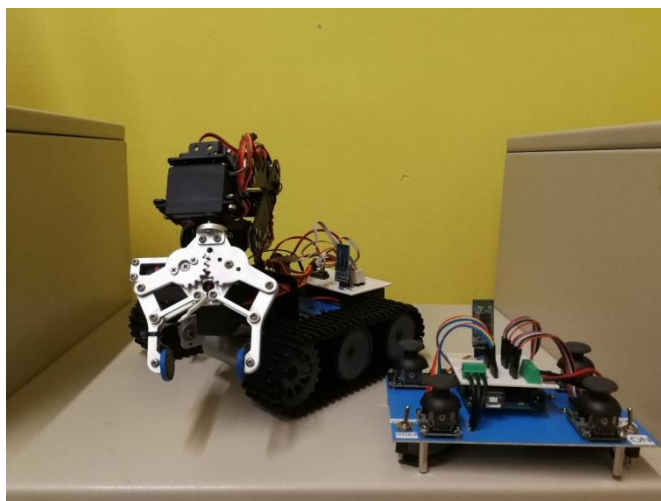
Sestavne dele za pogonski voziček vidimo na sliki 23.



*Slika 23: Sestavljanje pogonskega vozička (vir: avtor naloge)*

## 7. UGOTOVITVE IN REZULTATI

Rezultat naše raziskovalne naloge je funkcionalna robotska roka, kot jo lahko vidite na sliki 24. Z relativno skromnimi sredstvi nam je uspelo sestaviti zmogljivo robotsko roko. Prva testiranja daljinskega upravljanja robotske roke so potekala dokaj uspešno. Opazili smo nekaj manjših težav pri pozicioniranju servomotorjev ki pa jih v nadaljnjem optimiranju projekta nameravamo odpraviti.



*Slika 24: Končni izdelek (vir: avtor naloge)*

## 8. ZAKLJUČEK

Raziskovalno delo na področju robotike nam je pomagalo k boljšemu razumevanju delovanja elektronskih komponent in medsebojnih povezav, saj smo se s preizkušanjem in sestavljanjem vezja naučili nekaj novih veščin in znanj. Ugotovili smo, da sta natančno krmiljenje robotske roke kakor tudi določanje položaja vrha roke v prostoru zahtevni nalogi ki zahtevata precej znanja iz področja elektrotehnike, matematike in strojništva.

## 9. DRUŽBENA ODGOVORNOST

Z našim izdelkom, daljinsko voden robotsko roko, bi lahko prispevali k družbeno odgovornemu ravnanju na različnih področjih človekove dejavnosti tako na področju vzdrževanja naprav in objektov, kjer so prisotna določena tveganja za zdravje in življenje človeka, kakor tudi za uporabo na vojnih območjih za deaktiviranje eksplozivnih sredstev in reševanje žrtev iz nevarnih območij vojaških spopadov.

## 10. LITERATURA IN VIRI

### 10.1. Knjižni viri

- Bervar, G.(2008). **C++ NA KOLENIH**. 2. posodobljena izdaja. Ljubljana: Študentska založba (Zbirka Scripta)
- Fitzgerald, S. in Shiloh, M.(2015). **ARDUINO PROJECTS BOOK**. 1.izdaja. Torino, Italija (Arduino).

### 10.2. Spletni viri

<https://www.arduino.cc/> (15. December, 2017)

<https://arduino-info.wikispaces.com/BlueTooth-HC05-HC06-Modules-How-To> (15. December, 2017)

<http://www.instructables.com/id/Arduino-Servo-Motors/> (18. December, 2017)

<https://sl.wikipedia.org/wiki/Elektromotor> (2. Januar, 2018)

<https://www.eprojectzone.com/how-to-modify-the-pwm-frequency-on-the-arduino-part1/> (5. Januar, 2018)

<https://arduino.stackexchange.com/questions/38553/how-do-i-generate-exact-number-of-pwm-pulses-in-arduino> (5. Januar, 2018)

<https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM> (5. Januar, 2018)

<http://howtomechatronics.com/> (19. Januar 2018)

<https://www.youtube.com/watch?v=hyME1osgr7s> (19. Januar 2018)

### 10.3. Viri slik

Slika 2: <http://robohub.org/how-many-axes-does-my-robot-need/> (12. Januar, 2018),

Slika 8: [http://www.pyroelectro.com/tutorials/servo\\_motor/servomotor.html](http://www.pyroelectro.com/tutorials/servo_motor/servomotor.html) (12. Januar, 2018)

Slika 11: <https://www.prohobi.net/ps2016/si/dc-motorji/859-kovinski-mikro-motor-z-reduktorjem-10-1.html> (19. Januar, 2018)

Slika 13: <http://www.alldatasheet.com/datasheet-pdf/pdf/26889/TI/L293DNE.html> (19. Januar, 2018).