



## **Robot Pajek**

(raziskovalna naloga)

Mentor:  
Franc Vrbančič, učitelj

Avtorica:  
Anastazija Nograšek, dijakinja

Ptuj, februar 2023



## **Povzetek:**

Robotika je od svojih začetkov prehodila dolgo pot, napredek v tehnologiji pa jo je naredil bolj dostopno kot kdajkoli prej. Pred razvojem sodobne robotike so se roboti običajno uporabljali za industrijsko avtomatizacijo v proizvodnih obratih. Ti roboti so bili veliki, dragi in so za delovanje zahtevali specializirano programiranje.

Sčasoma se je robotska tehnologija razvila in postala bolj vsestranska, sofisticirana in cenovno dostopna. Robote danes najdemo na različnih področjih, kot so zdravstvo, kmetijstvo in zabava. Razvoj manjših, lažjih in učinkovitejših robotskih sistemov jim je omogočil opravljanje kompleksnih nalog z večjo natančnostjo.

Eden glavnih dejavnikov, ki so spodbudili rast robotike, je napredek računalniške tehnologije. Razvoj zmogljivih računalnikov in naprednih algoritmov je robotom omogočil izvajanje nalog, ki jih prej ni bilo mogoče izvesti. Poleg tega je široka razpoložljivost nizkocenovnih mikrokontrolerov in senzorjev olajšala izdelavo in programiranje robotov.

Drug dejavnik, ki prispeva k rasti robotike, je okrepljeno sodelovanje med šolstvom in industrijo. Raziskovalci in razvijalci z različnih področij sodelujejo pri ustvarjanju inovativnih rešitev za nekatere najbolj pereče težave na svetu. To sodelovanje je pripeljalo do razvoja novih aplikacij in tehnologij, ki so naredile robotiko dostopnejšo širšemu krogu uporabnikov..

Ena najpomembnejših sprememb v robotiki je razvoj odprtokodnih platform in skupnosti. Te platforme omogočajo dostop do modelov programske in strojne opreme, kar ljubiteljem, študentom in razvijalcem olajša ustvarjanje njihovih robotov. Razpoložljivost cenovno dostopnih robotskih kompletov in komponent je ljudem olajšala tudi eksperimentiranje z robotsko tehnologijo.

Na splošno je napredek v robotski tehnologiji naredil bolj dostopno in vsestransko kot kdajkoli prej. Vse večja razpoložljivost poceni komponent in odprtokodnih platform omogoča več ljudem, da sodelujejo pri razvoju robotike in ustvarjajo inovativne rešitve za težave v resničnem svetu. Posledično lahko pričakujemo znatno rast in širitev na področju robotike v naslednjih letih.

**Ključne besede:** avtomatizacija, robot, algoritem, vodenje z zvokom

## **Summary:**

Robotics has come a long way since its inception, and advancements in technology have made it more accessible than ever before. Before the development of modern robotics, robots were typically used for industrial automation in manufacturing plants. These robots were large, expensive, and required specialized programming to operate.

Over time, robotics technology has evolved, becoming more versatile, sophisticated, and affordable. Today, robots can be found in various fields, such as healthcare, agriculture, and entertainment. The development of smaller, lighter, and more efficient robotic systems has enabled them to perform complex tasks with greater precision and accuracy.

One of the main factors that have driven the growth of robotics is the advancement of computer technology. The development of powerful computers and advanced algorithms has allowed robots to perform tasks that were previously impossible. Additionally, the widespread availability of low-cost microcontrollers and sensors has made it easier to build and program robots.

Another factor contributing to the growth of robotics is the increased collaboration between academia and industry. Researchers and developers from various fields are working together to create innovative solutions to some of the world's most pressing problems. This collaboration has led to the development of new applications and technologies that have made robotics more accessible to a wider audience.

One of the most significant changes in robotics is the development of open-source platforms and communities. These platforms provide access to software and hardware designs, making it easier for hobbyists, students, and developers to create their robots. The availability of affordable robotics kits and components has also made it easier for people to experiment with robotics technology.

Overall, the advancements in robotics technology have made it more accessible and versatile than ever before. The increasing availability of low-cost components and open-source platforms is enabling more people to participate in robotics development and create innovative solutions to real-world problems. As a result, we can expect to see significant growth and expansion in the field of robotics in the coming years.

**Key words:** automation, robot, algorithm, sound remote control

***Strokovna terminologija***

<b>Kratica, tujka, pojem</b>	<b>Prevod, sinonim</b>	<b>Kratek opis</b>
Arduino	Arduino	Odprtokodna mikrokrmilniška platforma ali plošča in programsko opremo, ki se uporablja za njeno programiranje
USB	Univerzalno serijsko vodilo	Uporablja za povezavo računalnikov in zunanjih naprav
Application programming interface	Aplikacijski programski vmesnik	Omogoča, da se dve aplikaciji pogovarjata med seboj
PWM	Modulacija širine impulza	Metoda kodiranja informacij, ki temelji na variacijah trajanja nosilnih impulzov
Bluetooth	Standard brezžične tehnologije kratkega dosega	Za brezžično povezavo kratkega dosega mobilnih telefonov, računalnikov in drugih elektronskih naprav
COM	Komunikacijska vrata	Ime vmesnika serijskih vrat na računalnikih, združljivih z osebnim računalnikom
Microphone	Mikrofon	Naprava, ki spreminja zvok v električni signal
FHSS	Razpršen spekter s frekvenčnimi skoki	Ponavljajoče se preklapljanje nosilne frekvence med radijskim prenosom, da se zmanjšajo motnje in prepreči prestrezanje
I <sup>2</sup> C	Vodilo in protokol komunikacije	Sinhrono, večglavno/več podrejeno, enostransko, serijsko komunikacijsko vodilo
Microprocessor	Mikroprocesor	Digitalno elektronsko vezje, ki obdeluje (procesira) in nadzira podatke, ter opravlja ostale enote
WiFi	Brezžična tehnologija	Omogoča, da se lahko naprava poveže v računalniško omrežje
Algorithm	Algoritem	Zaporedje ukazov, s katerimi opravimo določeno nalogo
3D	Tridimenzionalno	Tri prostorske dimenzije: širina, višina in globina
Pin	Priključek	Priključek na elektronskem elementu

<b>Kratica, tujka, pojem</b>	<b>Prevod, sinonim</b>	<b>Kratek opis</b>
Coxa	Kolk	Strokovno ime za kolčno kost ali kolčni sklep
Femur	Stegnenica	Kost stegna ali zgornjega zadnjega uda, ki se zgiba v kolku in kolenu
Tibia	Golenica	Notranja in običajno večja od dveh kosti med kolenom in gležnjem
IDE	Integrirano razvojno okolje	Programska aplikacija, ki programerjem pomaga pri učinkovitem razvoju programske kode

## Kazalo vsebine

1	Uvod.....	8
1.1	Idejna shema .....	9
2	Opredelitev problema in raziskovalno vprašanje.....	10
2.1	Hipotezi z opisom .....	10
2.2	Opis metodologije raziskovanja .....	11
2.3	Načrt raziskave .....	11
2.4	Omejitve raziskave.....	12
3	Praktični del .....	13
3.1	Metode praktičnega dela .....	14
3.2	Strojni del.....	14
3.2.1	Arduino Uno.....	14
3.2.2	Mikro servo motor sg90 .....	15
3.2.3	HC-06 Bluetooth modul .....	17
3.2.4	Gonilnik Adafruit PCA9685 PWM.....	19
3.2.5	Ostalo .....	19
3.3	Finančna konstrukcija .....	19
4	Strojna oprema - praktična izvedba .....	20
4.1	Povezovanje komponent na Arduino Uno sistem .....	20
5	Glasovno upravljanje Pajka .....	21
6	Programska oprema .....	23
6.1	Algoritem.....	23
6.2	Dodatni napotki glasovnega krmiljenja .....	24
6.3	Koda.....	25
6.3.1	Komunikacija računalnika s HC-06 .....	25
6.3.2	Komunikacija računalnika z robotom.....	26
6.3.3	Krmiljenje servo motorjev .....	27
6.3.4	Premikanje robota .....	31
6.3.5	Funkcija za premik naprej .....	32
7	Rezultati hipotez .....	34
7.1	Preizkušanje robota .....	34
7.2	Podrobnejši opis preizkusa delovanja robota .....	35
7.3	Pravilnost hipotez.....	36
7.4	Končni sklep.....	36
8	Zaključek .....	37

9	Viri in literatura .....	38
9.1	Viri slik.....	38
9.2	Viri videov .....	39

## Kazalo slik

Slika 1: Idejna shema Pajka .....	9
Slika 2: Elektronske komponente robota Pajka .....	13
Slika 3: Arduino Uno .....	15
Slika 4: Servo Motor.....	16
Slika 5: Položaj servo motorjev na nogi.....	16
Slika 6: Priključitev servo motorjev na gonilnik PCA9685 in Arduino Uno.....	17
Slika 7: Bluetooth module .....	18
Slika 8: Priključitev bluetooth modula .....	18
Slika 9: Servo pogon Adafruit.....	19
Slika 10: Diagram poteka robota .....	23
Slika 11: Načrt pozicije servo motorjev in njihovo premikanje .....	28
Slika 12: Pozicije servo motorjev na robotu.....	29
Slika 13: Sklepi noge.....	30
Slika 14: Noge robota.....	30
Slika 15: Robot Pajek.....	31
Slika 16: Robot Pajek z baterijo .....	34

## Kazalo tabel

Tabela 1: Seznam sestavnih delov s cenikom.....	19
Tabela 2: Povezovanje modulov Arduino - PCA9685.....	20
Tabela 3: Povezovanje modulov Arduino – HC-06 Bluetooth .....	20
Tabela 4: Seznam ukazov robota.....	24



# 1 Uvod

Robotika je področje študija, ki se osredotoča na oblikovanje, razvoj in delovanje robotov. Robot je stroj, ki ga je mogoče programirati za izvajanje različnih nalog. Roboti so zasnovani tako, da posnemajo človeške gibe in vedenje, v številnih panogah pa se uporabljajo za avtomatizacijo nalog, ki so za ljudi pretežke ali nevarne.

Ena glavnih prednosti robotov nad ljudmi je, da se ne utrudijo. Delajo lahko brez potrebe po odmorih in lahko opravijo naloge s stopnjo natančnosti, ki se je ljudem težko ujemati. Poleg tega lahko roboti opravljajo delovna mesta, ki so preveč nevarna za ljudi, na primer delo v nevarnih okoljih ali ravnanje s strupenimi materiali.

Druga prednost robotov je, da lahko opravljajo težje naloge kot ljudje. Na primer, roboti lahko z lahkoto dvigujejo težke predmete in lahko delujejo v prevroče, hladnem ali nevljudnem za ljudi. Zaradi tega so idealna izbira za panoge, kot so proizvodnja, gradnja in rudarjenje.

Poleg tega so roboti lahko cenejši in hitrejši od človeških delavcev, zlasti pri ponavljajočih nalogah. To lahko privede do prihrankov stroškov in večje produktivnosti za podjetja.

Na splošno lahko robotika spremeni številne panoge z izboljšanjem učinkovitosti, varnosti in produktivnosti.

Kot je že iz imena razvidno - robot Pajek - bo naš robot posnemal gibe pajka, vendar ne bo izvajal popolnoma enakih gibov telesa, saj bomo namesto osmih nog uporabili samo štiri [1].

S pomočjo štirih nog se bo gibal naprej, nazaj in izvajal ostale predvidene gibe. Za izvedbo določenega giba bo delovanje ustrezno vodeno. Gibanje vsake noge bo povezano z drugimi nogami, da se prepozna položaj telesa robota in tudi nadzoruje ravnotežje telesa robota. Za štiri noge smo se odločili, ker je robot tako cenejši in enostavnejši od robota z več nogami, a kljub temu lahko doseže stabilnost in izvedbo kompleksnih gibov.

## 1.1 Idejna shema

Pajek bo deloval na način, da računalnik analizira zvok, ki ga prejme od mikrofona. Zvočni ukaz se nato pošlje na HC-06 Bluetooth modul, ki se nahaja na robotu. Robot bo krmiljen z mikrokrmilnikom. Pajek nato izvede podan ukaz, če je ta pravilen (slika 1). Prepoznal in izvedel bo le tiste ukaze, ki jih bo imel že vnaprej sprogramirane. Na računalniku bo program za prepoznavo zvoka.



Slika 1: Idejna shema Pajka

## 2 Opredelitev problema in raziskovalno vprašanje

V tem poglavju bomo pojasnili osnovni namen raziskovalne naloge in si zastavili raziskovalno vprašanje.

### 2.1 Hipotezi z opisom

Na osnovi izbranih podatkov smo si pred začetkom raziskovalnega procesa oz. izdelave naprave zastavili dve hipotezi.

Prva hipoteza se glasi:

**H1:** Podatki o robotu Pajek, ki ga lahko krmilimo z Arduino Uno, obstajajo in so prosto dostopni.

Informacije bomo iskali predvsem po spletu, saj smatramo, da jih bo tam dovolj za potrditev ali zavrnitev hipoteze. Zavedamo se, da s tem, ne bomo prišli do vseh virov, ki bi nam lahko koristili pri dokazovanju te hipoteze. Ogromno informacij o tematiki je tudi v revijah in drugih pisnih virih.

Predvidevamo, da bomo prišli do ene izmed ugotovitev:

- Naprava obstaja v celoti kot jo predvidevamo. V tem primeru jo bomo izvedli po najdenem viru ter preverili ali deluje, kot je opisano;
- Obstajajo rešitve posameznih opravil naše (želene) naprave a ne kot celota. V tem primeru bomo posamezne rešitve vključili v našo napravo in preverili delovanje naprave kot celote;
- Naprava ali deli naprave ne obstajajo – napravo ali dele naprave bomo načrtovali samostojno (na osnovi teorije in teoretičnega delovanja).

Drugo hipotezo zapišemo kot:

**H2:** Na osnovi izsledkov raziskave hipoteze 1 je možno narediti delujočo napravo za krmiljenje robota Pajka z zvočnim vmesnikom.

Druga hipoteza je vezana na izvedbo in testiranje delovanja robota Pajka. Odločili smo se za izvedbo robota, kjer bo uporabnik z glasom krmilil robota, ki ima že vnaprej pripravljene in sprogramirane ukaze.

Protokol delovanja robota bo potekal takole:

- uporabnik se bo povezal z Bluetooth krmilnikom iz računalnika in zagnal program za prepoznavo zvoka,
- uporabnik bo nato podal ukaz. Ukazi bodo že vnaprej programirani,
- če bo prepoznava ukaza uspešna, bo ta poslal ukaz Bluetooth modulu in se bo ukaz izvedel,

- program se bo končal, ko bo uporabnik pritisnil na tipko za konec izvajanja programa ali po 10 minutah branja zvoka.

Naprava bo delovala pravilno, če:

- se bo robot odzival na naše ukaze,
- prepoznal naš ukaz in
- pravilno deloval po ukazu.

V vseh drugih primerih bomo hipotezo zavrnil.

Za robota Pajka načrtujemo osnovne funkcije kot so: premik robota naprej, nazaj, levo, desno. Robota bomo vodili preko zvočnega vmesnika Bluetooth. Programirali bomo v programskem jeziku C++. Robota bomo tudi izvedli, mu namestili programsko opremo, testirali njegovo delovanje in podali mnenje o uspešnosti izvedbe naloge.

## 2.2 Opis metodologije raziskovanja

Z metodo študije pisnih in drugih predvsem e-virov bomo prišli do seznama materiala in seznama programske opreme. Pridobili bomo tudi informacije katero strojno opremo rabimo, kako deluje, kako jo povezati v delujočo celoto in kako jo programirati za izvedbo podane naloge – gibanje robota po zvočnih ukazih.

## 2.3 Načrt raziskave

Do izsledkov raziskave H1 smo prišli s pomočjo študije pisnih in drugih virov. Osnova so nam bile iskalne besede: robot pajek, Arduino, servo motorji.

Načrt dela smo si zastavili takole:

- izdelava vsebinskega načrta raziskave
- opredelitev raziskave, metode raziskovalnega dela, delovanje robota in namen raziskave,
- študija pisnih in drugih virov,
- načrtovanje in izvedba strojnega in programskega dela naprave,
- preizkušanje delovanja naprave
- rezultati in sklep.

## 2.4 Omejitve raziskave

Zavedamo se, da je dandanes na trgu možno najti rešitve, ki vsaj deloma podpirajo našo idejo o robotu Pajku, ki opravlja osnovna premikanja, ki se naj izvedejo:

- Obstaja že več primerov robotov pajkov, ki so trenutno v uporabi ali v razvoju, vključno z:
  - Robugtix T8X: To je komercialni robot pajek, ki je zasnovan za navdušence. Ima osem nog in se lahko programira za hojo, plazenje ali ples;
  - NASA Spidernaut: Tega robota pajka je NASA razvila za uporabo pri raziskovanju vesolja. Zasnovan je za plazenje skozi tesne prostore in za raziskovanje v vesolju;
  - Festo's BionicWheelBot: Ta robot pajek se zgleduje po gibanju pajka flic-flac in se lahko kotali po tleh ali plazi kot živi pajek;
  - RoboBee s Harvarda: Ta majhen robot pajek je zasnovan tako, da posnema gibanje čebele in lahko leti in se plazi skozi tesne prostore;
  - RoboSpiders: Raziskovalci na Univerzi v Manchestru so razvili vrsto robotov pajkov za uporabo v iskalnih in reševalnih operacijah. Ti roboti so opremljeni s kamerami in senzorji ter se lahko plazijo skozi ruševine in druge odpadke, da bi našli preživele;
  - Crabster: Ta robot pajek je zasnovan za podvodno raziskovanje in je opremljen s šestimi nogami, ki mu omogočajo plazenje po oceanskem dnu.

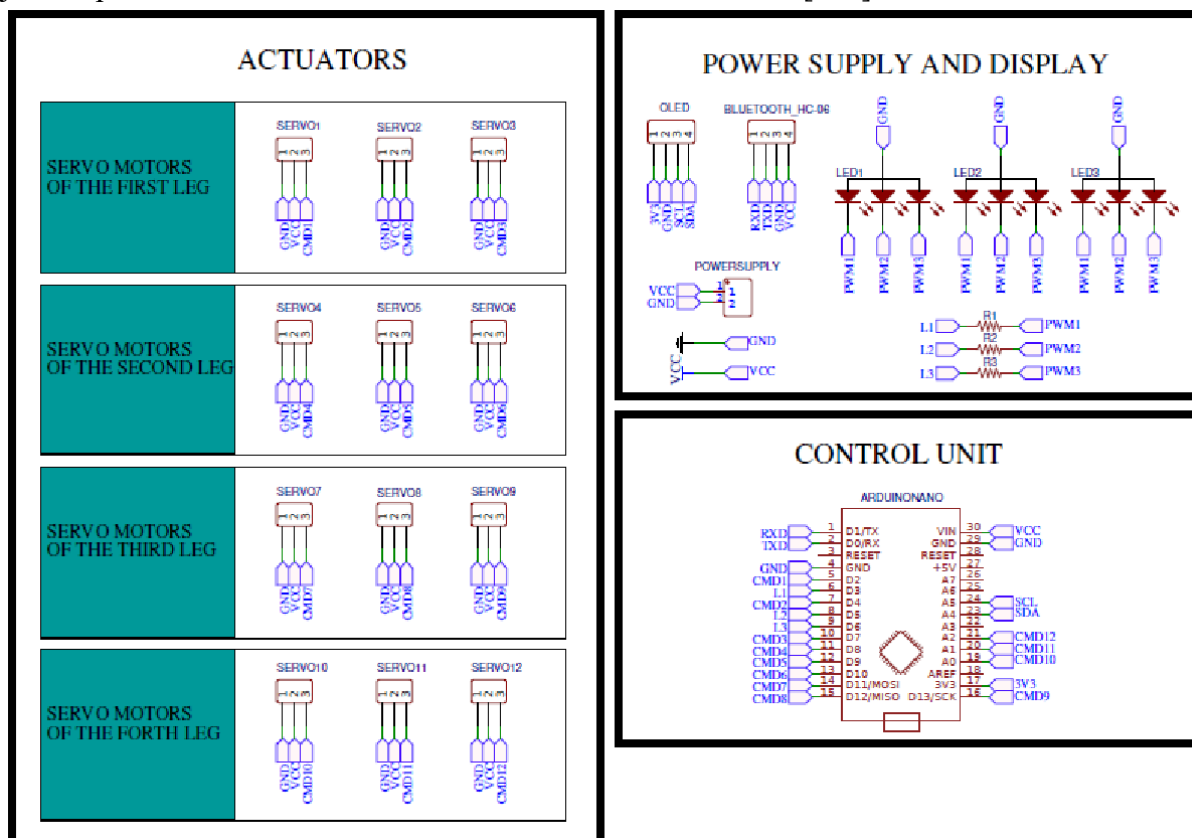
Z napredkom tehnologije bomo v prihodnosti verjetno videli veliko več aplikacij za robote pajke.

Opisi so zgolj informacija, da obstajajo tudi druge možnosti načrtovanja in izvedbe robota, ki pa zahtevajo mnogo več časa, znanja in denarja, ki pa ne bodo predmet te raziskave.

Zavedamo se tudi, da bi v primeru načrtovanja in razvoja naprave za tržišče, morali najprej raziskati, ali na tržišču sploh obstaja potreba po tovrstnih napravah. V našem primeru smo tovrstno raziskavo preskočili. Zaenkrat nam zadostuje raziskovanje, ali je predlaganega robota Pajka sploh možno izvesti. Pred morebitno masovno proizvodnjo robota bi bilo smiselno zraven raziskave trga izvesti tudi pretvorbo laboratorijske rešitve v industrijsko.

### 3 Praktični del

V tem poglavju bomo opisali delovanje in nakup komponent za izvedbo strojnega dela robota. S pomočjo e-virov smo zbrali potrebne komponente in sestavili delujočega robota. Pomagali smo si z shemo povezav elektronskih komponent (slika 2). Pogledali smo si tudi kako poteka krmiljenje servo motorjev. Robotu smo kodo programirali v Arduino IDE v C++ jeziku. Robotu smo nato dodali še HC-06 Bluetooth modul. Imeli smo tudi program za prepoznavo zvoka, ki je nato poslal ukaze modulu HC-06. Nato smo robota testirali [1, 2].



Slika 2: Elektronske komponente robota Pajka

(vir: [content.instructables.com/ARDUINO-SPIDER-ROBOT-QUADRUPED](http://content.instructables.com/ARDUINO-SPIDER-ROBOT-QUADRUPED))

Ideja za Pajka je, da bo Pajek imel 4 noge, na vsaki 3 sklepe, vsak sklep bo imel 1 servo motor. Robota bomo imenovali Pajek, ker se bo premikal kot pravi pajek. Čeprav ima le 4 noge in ne 8 kot jih ima pravi pajek, nas njegovo premikanje prepriča, da ga uvrščamo v robote pajke.

Želeli bi prikazati osnovno premikanje Pajka, ki bo izgledalo čim bolj podobno kot premikanje pravega pajka. Ideja je tudi, da bi njegovo premikanje krmilili brezžično, kar bi nam omogočilo enostavnejše krmiljenje robota in prenašanje ukazov, ki naj jih izvede.

Pajka bi lahko uporabili v primerih naravnih nesreč recimo potresih, ko bi iskali pogrešane ljudi v ruševinah. Lahko bi tudi »tavalili« po težko dostopnih področjih (močvirja, gorata področja) in jih kartirali. In še bi lahko naštevali uporabnost tovrstnih robotov, kar pa ni namen te naloge. Glede na specifikko uporabe robota bi ga bilo koristno opremiti še s kamero, senzorji, sončnimi celicami itd., s čimer bi mu povečali uporabnost, samostojnost in mobilnost.

Lahko pa se ga uporabi tudi za zabavo, raziskovanje ali prenašanje bremen.

## 3.1 Metode praktičnega dela

Pri izvajanju praktičnega dela smo izvedli naslednje postopke:

- Načrtovanje sistema (zbiranje idej po spletu, izbira komponent in materiala);
- Nakup materiala;
- Priprava komponent, kot so 3D ohišje, servo motorji, gonilnik PCA9685 PWM, povezava komponent na Arduino Uno;
- Sestava komponent;
- Programiranje robota;
- Preizkušanje izdelka ter vrednotenje rezultatov preizkušanja.

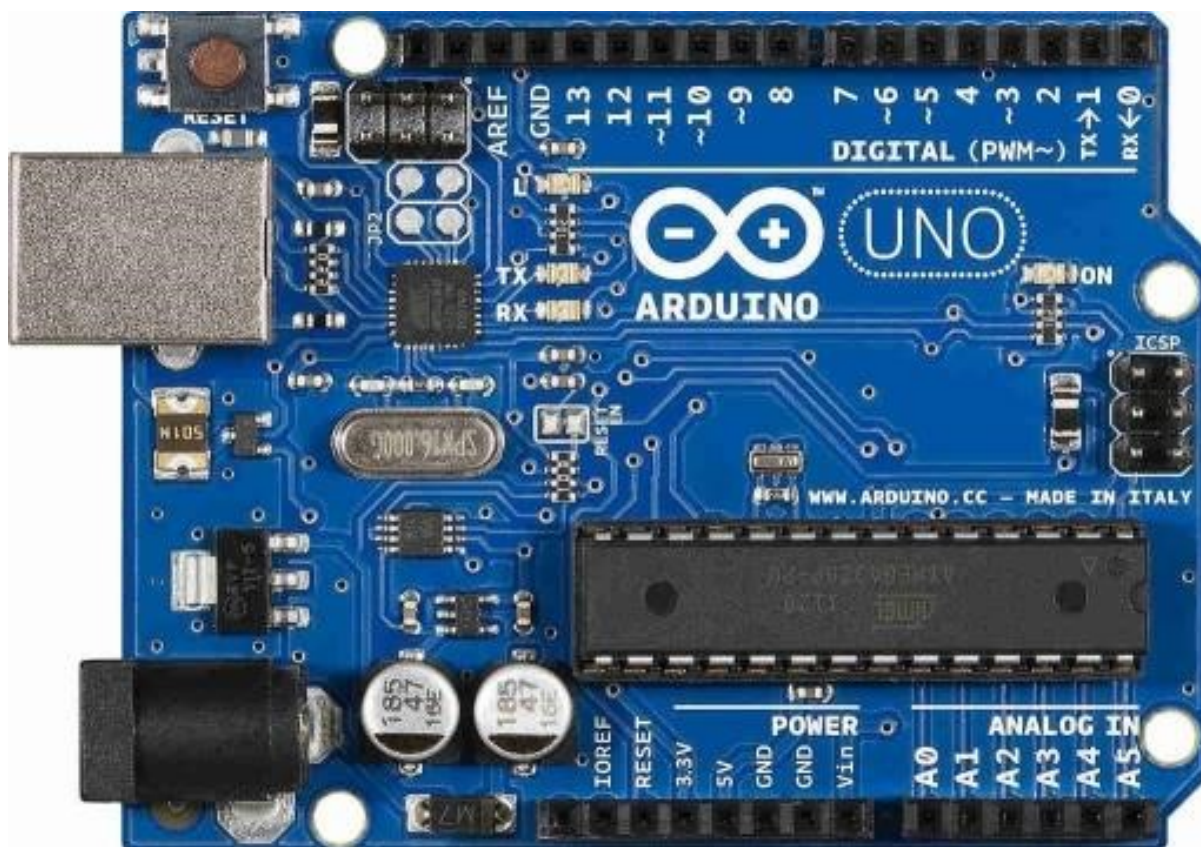
## 3.2 Strojni del

Odločili smo se, da bomo vključili: 3D ohišje, Arduino Uno, servo motorje, HC-06 Bluetooth modul, gonilnik Adafruit PCA9685 PWM, prepoznavo zvoka Google API, krmiljenje s prepoznavo zvoka. V nadaljevanju bomo opisali delovanje posameznih komponent, ki so potrebne za izvedbo strojnega dela robota [3, 4, 5, 6].

### 3.2.1 Arduino Uno

Arduino je odprtokodna strojna platforma, katere namen je enostavna uporaba strojnega krmilnika, na katerega lahko priklopimo različne senzorje za temperaturo, vlago, servo motorčke, ethernet modul, WiFi modul ter še veliko drugih stvari. S programsko opremo Arduino IDE lahko krmilnik sprogramiramo tako, da se obnaša po naših željah. Krmilnik je na voljo v različnih izvedbah. Najbolj znani in uporabljeni modeli so: Arduino Uno, Leonardo, Mega 2560, ki jih lahko najdemo v različnih velikostih. Različne Arduino ploščice imajo različne mikrokrmilnike, ki so si tudi med seboj podobni. Razlikujejo se po številu vhodov in izhodov. Izbira je vsekakor velika.

Za svoj projekt smo si izbrali Arduino Uno (slika 3), ki podpira 14 priključkov, za katere lahko posebej določimo, ali bo pin predstavljal digitalni vhod ali izhod. Razvojnica se programira v programu Arduino IDE, katerega lahko prenesemo iz njihove uradne strani. Programski jezik je podoben C++ ali oz. C-ju. Priključitev krmilnika na računalnik je enostavna, saj ga lahko priklopimo kar na običajna USB vrata. Preko USB-ja lahko tako nalagamo lastno programsko opremo na krmilnik, ki ga programiramo.



Slika 3: Arduino Uno

(vir: cdn.shopify.com)

### 3.2.2 Mikro servo motor sg90

Mikro servo motor SG90 je majhen, visoko zmogljiv servo motor, ki se pogosto uporablja v robotiki, izdelavi modelov in drugih ljubiteljskih projektih. Ima kompaktno obliko in je relativno poceni, zaradi česar je privlačna izbira za številne aplikacije. SG90 ima maso 9 gramov in velikost 22,8 x 11,8 x 22,7 mm, zaradi česar je dovolj majhen za uporabo v kompaktnih in lahkih robotskih oblikah. Ima nazivni navor 1,8 kg/cm, kar zadostuje za večino ljubiteljskih aplikacij in majhnih robotskih projektov. Servo motor ima tudi širino mrtvega pasu le 1  $\mu$ s, kar zagotavlja natančen nadzor in lego izhodne gredi servo motorja.

Servo se lahko vrti za približno 180 stopinj (90 v vsako smer) in deluje tako kot standardne vrste, vendar manjši. Za nadzor teh servomotorjev lahko uporabimo katerokoli servo kodo, strojno opremo ali knjižnico.

Potrebovali bomo 12 mikro servo motorjev (slika 4). Na vsaki nogi pajka po 3 motorje.

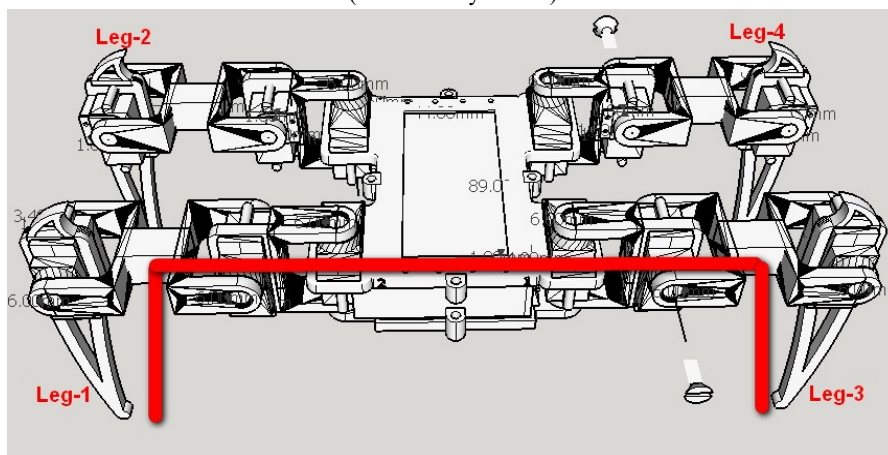
Servo motorje je pred vgradnjo potrebno ustrezno – potrebno je odviti in odstraniti servo ročice, da se motorji prosto vrtijo ter jih postaviti v začetno lego glede na položaj v robotu (slika 5). Vse servo motorje povežemo z gonilnikom PCA9685, ki je povezan z Arduino Uno (slika 6).





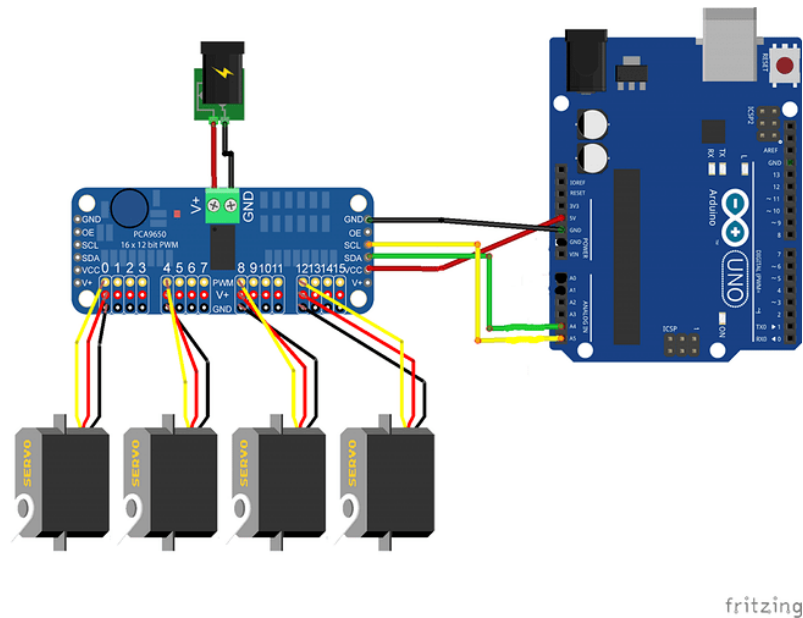
Slika 4: Servo Motor

(vir: [www.flyrobo.in](http://www.flyrobo.in))



Slika 5: Položaj servo motorjev na nogi

(vir: [github.com](https://github.com))



Slika 6: Priklučitev servo motorjev na gonilnik PCA9685 in Arduino Uno

(vir: <https://europe1.discourse-cdn.com>)

### 3.2.3 HC-06 Bluetooth modul

HC-06 je Bluetooth modul (slika 7), zasnovan za vzpostavitev brezžične podatkovne komunikacije kratkega dosegga med dvema mikrokontrolerjema ali dvema poljubnima sistemoma opremljenih s to komunikacijo. Modul deluje na komunikacijskem protokolu Bluetooth 2.0 in lahko deluje le kot pomožna naprava. To je najcenejši način brezžičnega prenosa podatkov in bolj prilagodljiv v primerjavi z drugimi načini ter lahko prenaša celo podatke s hitrostjo do 2,1 Mb/s.

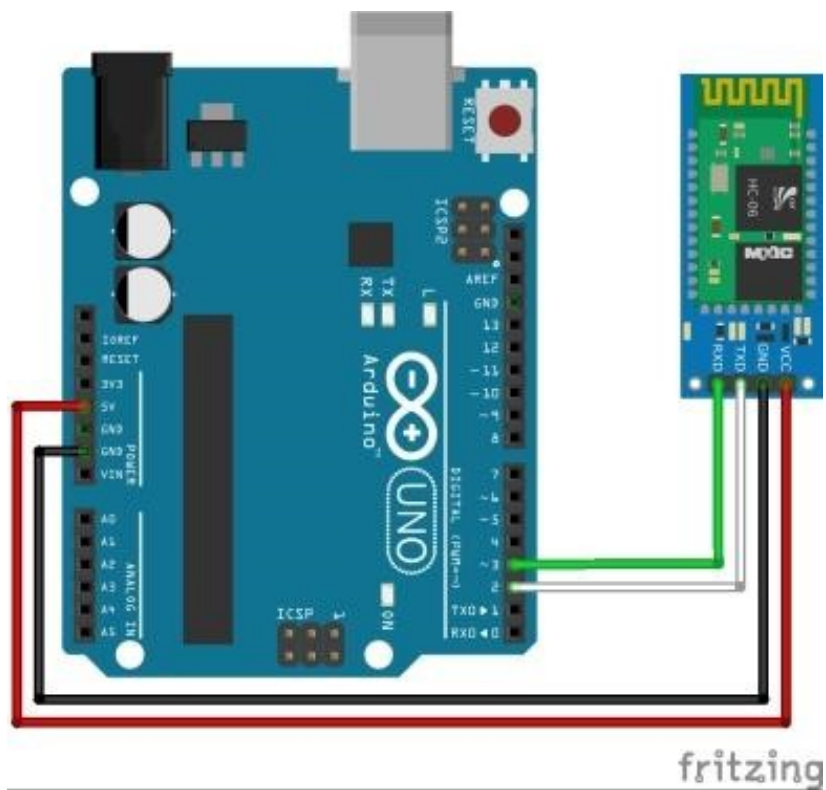
HC-06 uporablja tehniko razširjenega spektra s frekvenčnimi skoki (FHSS), da se izogne motnjam drugih naprav in omogoči popoln duplex prenos. Naprava deluje na frekvenčnem območju od 2,402 GHz do 2,480 GHz.

Uporabili smo HC-06 Bluetooth module, ki je kompatibilen samo z Android sistemi. Priklučitev modula na arduino vezje je prikazana na sliki 8.



Slika 7: Bluetooth module

(vir: [i0.wp.com/brainstormtech.in](http://i0.wp.com/brainstormtech.in))



Slika 8: Priključitev bluetooth modula

(vir: [www.aranacorp.com/wp-content](http://www.aranacorp.com/wp-content))

### 3.2.4 Gonilnik Adafruit PCA9685 PWM

Pogon servo motorjev s knjižnico Arduino Servo je precej enostaven, vendar vsak porabi dragocen priključek - da ne omenjamo nekaj procesorske moči Arduino. Adafruit 16-kanalni 12-bitni PWM/servo gonilnik bo poganjal do 16 servo motorjev prek I2C s samo 2 priključkoma. Krmilnik PWM bo na robotu upravljal vseh 16 kanalov hkrati brez dodatnih stroškov obdelave Arduino. Lahko jih povežete v verigo do 62 za nadzor do 992 servomotorjev – vse z istima 2 priključkoma. (slika 9).

Gonilnik Adafruit PWM je odlična rešitev za vsak projekt, ki zahteva veliko servo motorjev, zato smo ga uporabili za svojih 12 servo motorjev.



Slika 9: Servo pogon Adafruit

(vir: grobotronics.com/images)

### 3.2.5 Ostalo

Za izvedbo robota smo uporabili še:

- napajalnik 5V, 3500 mA,
- napajalnik Arduino,
- povezovalne žice za povezovanje modulov na Arduino krmilnik,
- 3D ohišje (med najdenimi spletnimi viri smo našli načrt za tiskanje 3D ohišja, ki služi kot sestava robota. Natisnili smo jih sami, saj smo imeli 3D tiskalnik).

## 3.3 Finančna konstrukcija

V tabeli prikazujemo spletno trgovino in cenik komponent, ki smo jih uporabili. Odločili smo se za nakup potrebnih komponent preko spletne trgovine Amazon de. Ostalo, kar ni na seznamu smo že imeli.

Tabela 1: Seznam sestavnih delov s cenikom

Komponenta	Cena (Amazon de)
Arduino Uno	25€
Micro servo motor sg90	1.98€
HC-06 Bluetooth modul	9.99€
Gonilnik Adafruit PCA9686 PWM	7.5€

Opomba: Rabimo 12 servo motorjev =  $12 * 1.98€ = 23,76€$

## 4 Strojna oprema - praktična izvedba

Robota smo izvedli po korakih:

- Načrtovanje sistema (izbira idej, raziskava in ugotovitve, postavitve hipotez, izbira komponent in materiala, načrtovanje algoritma in programskega dela);
- Nakup materiala;
- Priprava komponent, kot so servo motorji, Arduino Uno, PCA9685, HC-06 Bluetooth modul, mikrofoni;
- Programiranje sistema;
- Testiranje naprave.

### 4.1 Povezovanje komponent na Arduino Uno sistem

Arduino in PCA9685 komunicirata z uporabo I<sup>2</sup>C vodila

*Tabela 2: Povezovanje modulov Arduino - PCA9685*

<b>Arduino Uno</b>	<b>PCA9685</b>
5V	VCC
A4	SDA
A5	SCL
GND	GND

*Tabela 3: Povezovanje modulov Arduino – HC-06 Bluetooth*

<b>Arduino Uno</b>	<b>HC-06</b>
3.3V	VCC
GND	GND
TX	RXD
RX	TXD

Opomba:

Servo motorji potrebujejo večje tokove (nekaj deset mA). Da bi preprečili preobremenitev gonilnika PWM, ima PCA9685 svoj napajalni priključek.

## 5 Glasovno upravljanje Pajka

Za glasovno upravljanje robota smo potrebovali:

- **Git za Windows**

Naložimo Git za Windows.

- **Visual Studio 2019**

Prenesimo in namestimo Visual Studio 2019 Community Edition. Izberimo Razvoj namizja z delovno okolje C++ in se prepričamo, da so izbrana tudi orodja C++ CMake za komponento Windows.

Opomba: Vse spodnje ukaze je treba izvesti iz Start → Visual Studio 2019 → Orodja Visual Studio → Ukazni poziv razvijalca za VS 2019. Nekateri ukazi zahtevajo zvišanje (Zaženi kot skrbnik).

- **vcpkg**

V povišanem ukaznem pozivu zaženemo:

```
git clone https://github.com/Microsoft/vcpkg.git C:\vcpkg
C:\vcpkg\vcpkg\bootstrap-vcpkg.bat
C:\vcpkg\vcpkg integrate install
```

- **grpc in protobuf**

Prevedimo in namestimo grpc in protobuf z uporabo vcpkg:

```
C:\vcpkg\vcpkg install grpc:x64-windows protobuf:x64-windows
```

- **korenine.pem (roots.pem)**

gRPC zahteva spremenljivko okolja za konfiguracijo shrambe zaupanja za potrdila SSL. Nastavimo spremenljivko okolja `GRPC_DEFAULT_SSL_ROOTS_FILE_PATH`, da kaže na datoteko `roots.pem` v korenu tega skladišča.

V nadaljevanju opisujemo postopek inštalacije in aktivacije glasovnega upravljanja robota.

### Ustvarimo projekt v konzoli Google Cloud Platform Console

Če še nismo ustvarili projekta, ga ustvarimo zdaj. Projekti nam omogočajo upravljanje vseh virov Google Cloud Platform za našo aplikacijo, vključno z uvajanjem, nadzorom dostopa, obračunavanjem in storitvami.

1. Odprimo konzolo Cloud Platform.
2. V spustnem meniju na vrhu izberimo Ustvari projekt.
3. Poimenujemo svoj projekt.
4. Zabeležimo si ID projekta, ki se lahko razlikuje od imena projekta. ID projekta se uporablja v ukazih in konfiguracijah.

### Omogočimo obračunavanje za svoj projekt

Če za svoj projekt še nismo omogočili zaračunavanja, omogočimo zaračunavanje zdaj. Če omogočimo obračunavanje, lahko aplikacija porabi plačljive vire, kot so klici Speech API. Za

več informacij o nastavitvah obračunavanja si oglejte pomoč za konzolo Cloud Platform Console.

### **Omogočimo API-je za svoj projekt**

Kliknemo tukaj, če želimo obiskati konzolo Cloud Platform Console in omogočiti Speech API.

### **Po potrebi prigrasimo projekt obračunavanja**

Če za preverjanje pristnosti uporabljamo uporabniški račun, moramo nastaviti spremenljivko okolja `GOOGLE_CLOUD_CPP_USER_PROJECT` za projekt, ki smo ga ustvarili v prejšnjem koraku. Zavedajmo se, da morate imeti dovoljenje `serviceusage.services.use` za projekt. Druga možnost je, da uporabimo storitveni račun, kot je opisano v nadaljevanju.

### **Prenesimo poverilnice storitvenega računa**

Ti vzorci lahko uporabljajo storitvene račune za preverjanje pristnosti.

1. Obiščimo Cloud Console in se pomaknimo do: API Manager > Poverilnice > Ustvari poverilnice > Ključ računa storitve
2. V razdelku Račun storitve izberimo Nov račun storitve.
3. Pod Ime računa storitve vnesimo ime računa storitve po vaši izbiri. Na primer prepisovalka.
4. V razdelku Vloga izberimo Projekt > Lastnik.
5. V razdelku Vrsta ključa pustimo JSON izbran.
6. Kliknemo Ustvari, da ustvarimo nov storitveni račun, in prenesti datoteko s poverilnicami json.
7. Nastavimo spremenljivko okolja `GOOGLE_APPLICATION_CREDENTIALS`, da kaže na naše prenesene poverilnice storitvenega računa.

## 6 Programska oprema

V tem poglavju bomo opisali načrtovanje in izvedbo programskega algoritma naprave ter njegovo delovanje.

Algoritem bomo izvedli v prilagojenem jeziku C++ – Arduino verzija. Pajek deluje na način, da računalnik analizira zvok za ukaz, ki ga nato Pajek izvede.

### 6.1 Algoritem

Algoritem je navodilo, s katerim se rešuje nek problem. Zapisan je po seznamu točno določenih korakov, ki pripeljejo do rešitve problema. Značilnosti algoritma so: ima podatke, vrne rezultat, je natančno določen, se vedno konča in ga je mogoče opraviti. Robota krmilimo z glasovnimi ukazi. Ukaze sprejme in analizira programska oprema na računalniku. Če je ukaz prepoznan kot veljaven, ga računalnik pošlje na robota, ki ga izvrši (slika 10).



Slika 10: Diagram poteka robota



## 6.2 Dodatni napotki glasovnega krmiljenja

Koda sprejema ukaze preko Bluetootha in izvede zahtevano potezo. Ukazi so že vnaprej programirani v kodi. Program zaženemo z naslednjim ukazom:

```
cd speech\api
```

```
.build\Debug\streaming_transcribe --language-code sl-SI --bitrate 32000 COM5
```

Uporabimo jezikovne kode sl-SI(slovensko) ali en (angleško).

Uporabimo ista vrata COM, ki so dodeljena HC-06 v Bluetooth.

Najboljši uspeh pri prepoznavanju je bil opažen pri frekvenci vzorčenja 32000 Hz in profesionalnem mikrofону. 16000 Hz se je izkazalo za preveč nezanesljivo za ženske in otroške glasove.

Streaming\_transcribe bo poslušal zvok na privzeti snemalni napravi 10 minut. Lahko ga prekličemo s Ctrl+C.

*Tabela 4: Seznam ukazov robota*

<b>Glasovni ukaz</b>	<b>Pomen ukaza</b>
robot pojdi naprej	Robot stopi korak naprej.
robot pojdi nazaj	Robot stopi korak nazaj.
robot pojdi levo	Robot se obrne levo.
robot obrni se levo	Robot se obrne levo.
robot pojdi desno	Robot se obrne desno.
robot obrni se desno	Robot se obrne desno.

## 6.3 Koda

V nadaljevanju bomo opisali pomembnejše programske ukaze

### 6.3.1 Komunikacija računalnika s HC-06

Za primerjavo vzemimo naslednji del kode, da vidimo kako komunikacija med HC-06 in računalnikom deluje.

```
auto t = transcript.c_str();
if (std::strcmp(t, "robot pojdi naprej") == 0 || std::strcmp(t, "robot go forward") == 0)
    command = "f";

else if (std::strcmp(t, "robot pojdi nazaj") == 0 || std::strcmp(t, "robot go backward") == 0)
    command = "b";

else if (std::strcmp(t, "robot pojdi levo") == 0 || std::strcmp(t, "robot obrni se levo") == 0 ||
std::strcmp(t, "robot turn left") == 0)
    command = "l";

else if (std::strcmp(t, "robot pojdi desno") == 0 || std::strcmp(t, "robot obrni se desno") == 0 ||
std::strcmp(t, "robot turn right") == 0)
    command = "r";

else
    command.clear();
```

### Opis kode

To del kode uporabe API-ja Google Cloud Speech-to-Text za prepoznavanje govornega vnosa iz mikrofona in nadzor robota na podlagi izgovorjenih ukazov.

Ta del blok kode C++, ki vzame niz vnosa iz spremenljivke, imenovane transkript. Uporablja niz pogojnih stavkov if in else if za določitev ustreznega ukaznega niza.

Prva vrstica kode, `auto t = transcript.c_str();`, ustvari kazalec `const char*` na niz prepisa z uporabo funkcije `c_str()`. To se naredi, da se zagotovi, da se lahko funkcija `std::strcmp()` (ki kot argumenta prejme dva niza v slogu C) uporabi za primerjavo prepisa s ključnimi besedami ukaza.

Nato koda uporablja niz funkcij `std::strcmp()` za primerjavo kazalca `t` (ki kaže na niz prepisa) z že vnaprej določenih ukazov. Če se najde ujemanje, se ustreznemu ukaznemu nizu dodeli vrednost. Na primer, če transkript vsebuje niz "robot pojdi naprej" ali "robot pojdi naprej", se ukaznemu nizu dodeli vrednost "f".

Če se nobena od ključnih besed ukaza ne ujema z nizom prepisa, se ukazni niz počisti s funkcijo `clear()` in tako robotu ni poslan noben ukaz.

### 6.3.2 Komunikacija računalnika z robotom

Ta del kode je shranjen v Arduino Uno na Robotu. Ta koda bo dobila sprejet ukaz od prepoznave zvoka in izvedla dani ukaz, ki je že sprogramiran v kodi.

```
void loop()
{
  while(Serial.available())
  {
    switch (Serial.read()) {
      case 'f':
        step_forward(2);
        break;

      case 'b':
        step_back(2);
        break;

      case 'l':
        turn_left(2);
        break;

      case 'r':
        turn_right(2);
        break;
    }
  }
}
```

#### Opis kode

To je del kode programa Arduino, ki posluša serijske podatke, ki prihajajo skozi serijska vrata. Funkcija `Loop()` se v programu Arduino se izvaja neskončno.

S funkcijo `Serial.available()` koda preveri, ali so v serijskem medpomnilniku na volji kakšni podatki. Preverimo ali Arduino dobi shranjene bajte iz serijskih vrat, ki so na voljo za branje. To so podatki, ki so že shranjeni in prispejo v serijski medpomnilnik. Serijski vmesnik v Arduino hrani 64 bajtov.

Če so na voljo podatki, beremo s funkcijo `Serial.Read()` in primerjamo z štirimi možnimi primeri z uporabo stavka `Switch`.

Če so prejeti podatki 'f', se funkcija `Step_forward ()` pokliče z argumentom 2. Če so prejeti podatki 'b', se funkcija `Step_back ()` pokliče z argumentom 2. Če so prejeti podatki 'l', funkcija `turn_left ()` se imenuje z argumentom 2. Če so prejeti podatki 'r', se funkcija `Turn_right ()` imenuje z argumentom 2.

Nato bo ta klical dano funkcijo, v kateri je sprogramiran ukaz in se nato ukaz izvede. Kako deluje argument v funkciji bomo izvedeli pri naslednjem primeru za korak naprej.

### 6.3.3 Krmiljenje servo motorjev

Pri delu s servo motorji, zlasti modelom SG90, je treba upoštevati nekaj stvari, da zagotovimo varno in učinkovito delovanje našega robota:

- **Napajanje:** Zagotovimo, da servo motor prejme ustrezno napetost in tok iz napajalnika. Servo motorji SG90 običajno zahtevajo 4,8 V do 6 V DC in tokovno zmogljivost vsaj 500 mA. Uporaba nezadostnega napajanja lahko povzroči okvaro ali celo uničenje servo motorja;
- **Ožičenje:** pravilno priključitev servo motor na krmilno vezje in zagotovimo, da je ožičenje varno in ne-razrahljano. Zrahljane povezave lahko povzročijo nepravilno gibanje in poškodbe servo motorja;
- **Preobremenitev:** Izogibajmo se preobremenitvi servo motorja z uporabo prekomerne sile ali navora na gredi. Servo motorji SG90 niso zelo zmogljivi in imajo omejen navor, zato lahko preobremenitev povzroči zaustavitev ali uničenje motorja;
- **Temperatura:** Servo motorji med delovanjem proizvajajo toploto in pregrevanje lahko povzroči poškodbe motorja. Izogibajmo se neprekinjenemu dolgotrajnemu delovanju servo motorja, zlasti pri visokih obremenitvah ali visokih hitrostih, in zagotovimo zadostno prezračevanje in hlajenje;
- **Montaža:** Servo motor namestimo varno in v pravilni orientaciji, da preprečimo kakršno koli nepotrebno obremenitev ali obremenitev motorja. Prepričajmo se, da je pritrdilna oprema trdno zategnjena, vendar ne premočno, saj lahko to povzroči zaklep motorja.

Prikazana je koda za krmiljenje servo motorjev:

```
#include <Servo.h>
Servo servo[4][3];
//Definiramo porte za servo motorje
//Za vsako nogo, skupaj 4 in 3 servo motorji na vsaki nogi
//Vsaka vrstica predstavlja nogo, vsak stolpec pa servo motor, povezan s to nogo
const int servo_pin[4][3] = { {2, 3, 4}, {5, 6, 7}, {8, 9, 10}, {11, 12, 13} };
void setup()
{ //initialize all servos
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].attach(servo_pin[i][j]);
      delay(20);
    }
  }
}
```

```

    }
  }
}

void loop(void)
{
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].write(90);
      delay(20);
    }
  }
}

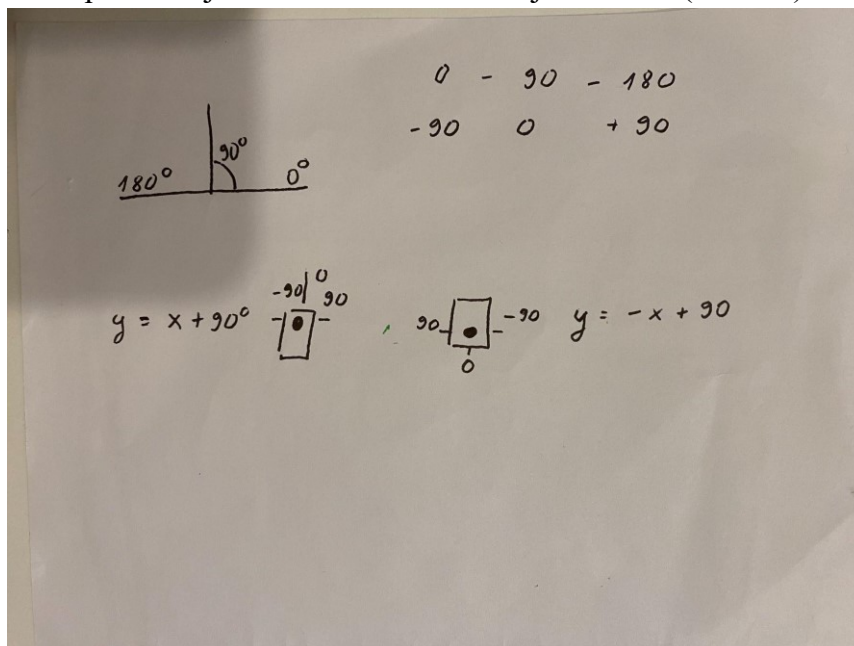
```

### Opis kode

Koda je namenjena za krmiljenje skupine 12 servo motorjev z uporabo plošče Arduino. Za nadzor servomotorjev uporabljamo knjižnico "Servo". Koda definira matriko 4x3 servo objektov, od katerih vsak predstavlja en servo motor. Priključki, na katere so povezani servomotorji, so definirani v 2-dimenzionalnem nizu, imenovanem "servo\_pin".

V funkciji setup() so objekti Servo pritrjeni na svoje ustrezne priključke z uporabo metode attach(). Majhna zakasnitev se doda po vsakem priklopu servo.

V funkciji loop() so vsi servo motorji nastavljeni na položaj 90 stopinj z uporabo metode write(). Po vsakem premikanju servo se doda še en majhen zamik (slika 11).



Slika 11: Načrt pozicije servo motorjev in njihovo premikanje

Pozicije nog:

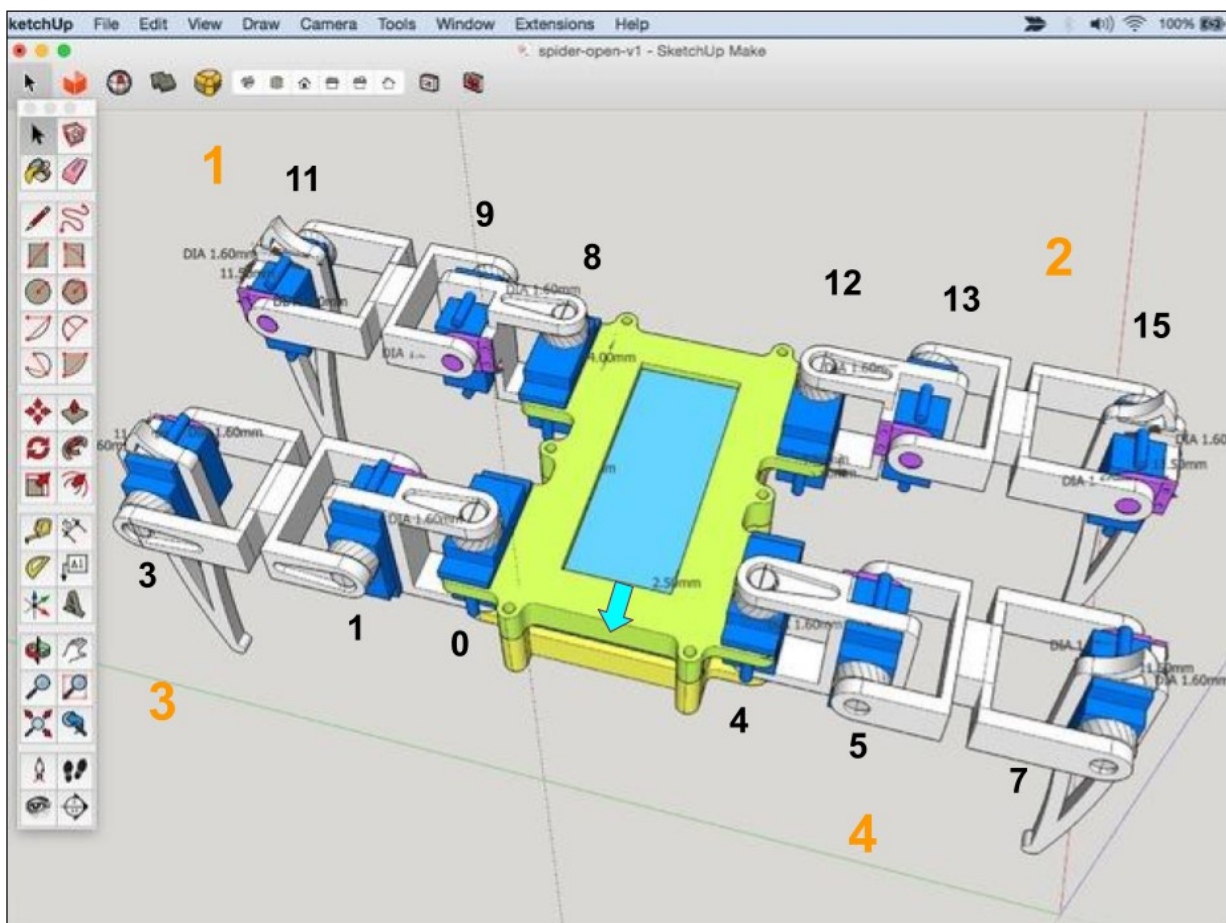
```
//          Leg 0 (FR)  Leg 1 (RR)  Leg 2 (FL)  Leg 3 (RL)
//          F C T    F C T    F C T    F C T
const int servo_pin[4][3] = {{ 1, 3, 0}, { 9, 11, 8}, { 5, 7, 4}, { 13, 15, 12}};
```

Ta koda je del matrike, ki shranjuje številke priključkov za servo motorje, ki nadzorujejo sklepe nog štirinožnega robota.

Ta vrstica kode definira dvodimenzionalno matriko, imenovano "servo\_pin" velikosti 4x3 s celimi vrednostmi. Vsaka vrstica predstavlja nogo robota, vsak stolpec pa predstavlja priključek servo motorja za ta nožni sklep. Komentarji navajajo, katera noga (FR = sprednja desna, RR = zadnja desna, FL = sprednja leva, RL = zadnja leva) in sklep (F = stegnenica, C = kolk, T = golenica) ustrezata posameznemu stolpcu.

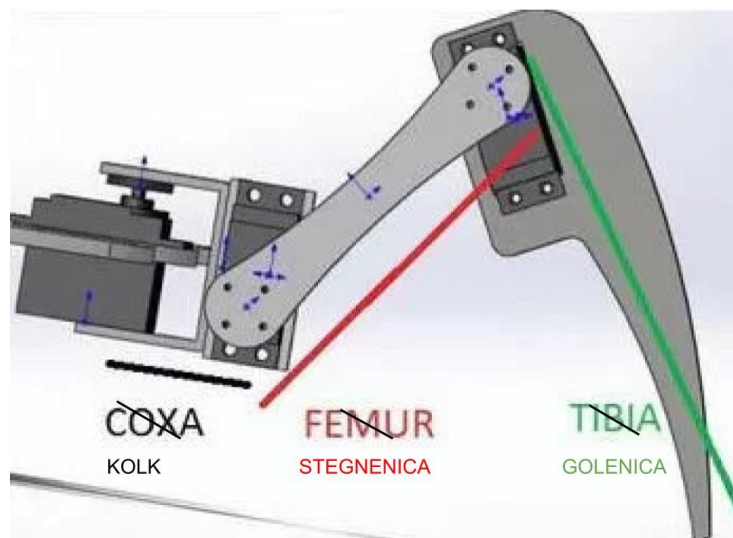
Tako na primer servo\_pin[0][0] predstavlja številko priključka za sklep stegnenice sprednje desne noge, ki je 1. servo\_pin[1][1] predstavlja številko priključka za kolčni (kolk) sklep zadnje desne noge, ki je 11.

Iz slike 12 je razvidna pozicija nog in pozicija servo motorjev.



Slika 12: Pozicije servo motorjev na robotu

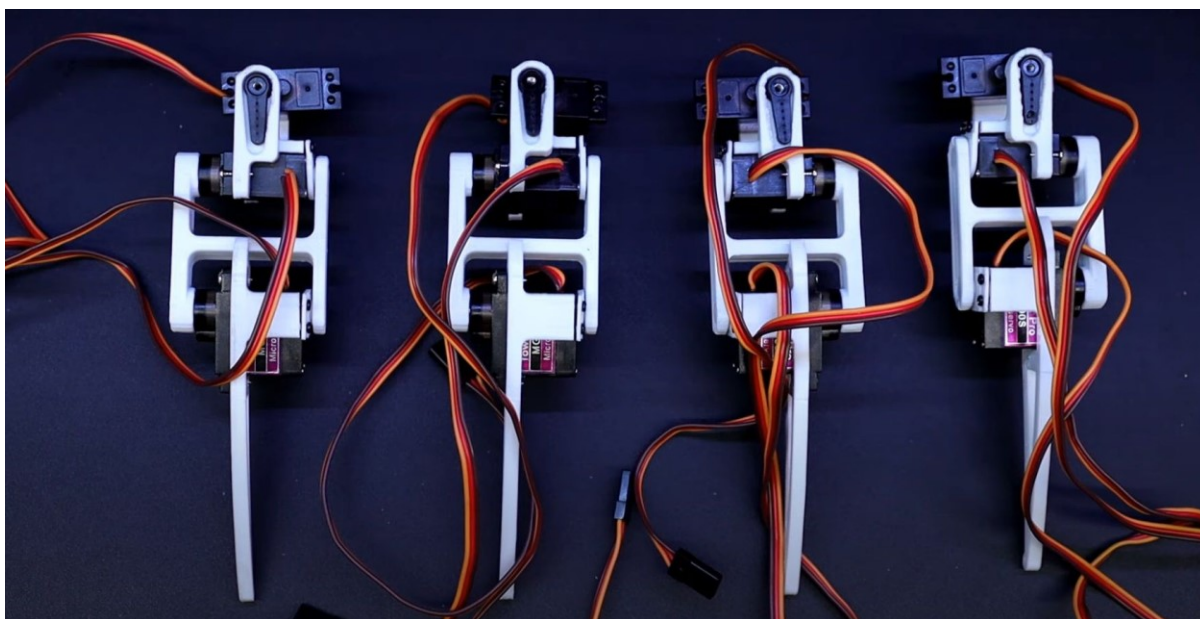
Vsaka noga takšnega robota je sestavljena iz treh sklepov, in sicer kot pravijo: kolka, stegenice in golenice. Od teh vsaka poganja en sam servo motor (slika 13).



Slika 13: Sklepi noge

(vir: content.instructables.com )

Ti trije sklepi skupaj omogočajo, da se noga robota pajka premika v različne smeri in opravlja različne naloge, kot so hoja, plazenje in obračanje. S krmiljenjem servo motorjev, ki aktivirajo vsak sklep, lahko robot pajek neodvisno premika vsako nogo in dosega zapletene gibe. Imamo štiri noge s tremi servo motorji na vsakem sklepu (slika 14).



Slika 14: Noge robota

(vir: <https://images.interestingengineering.com>)

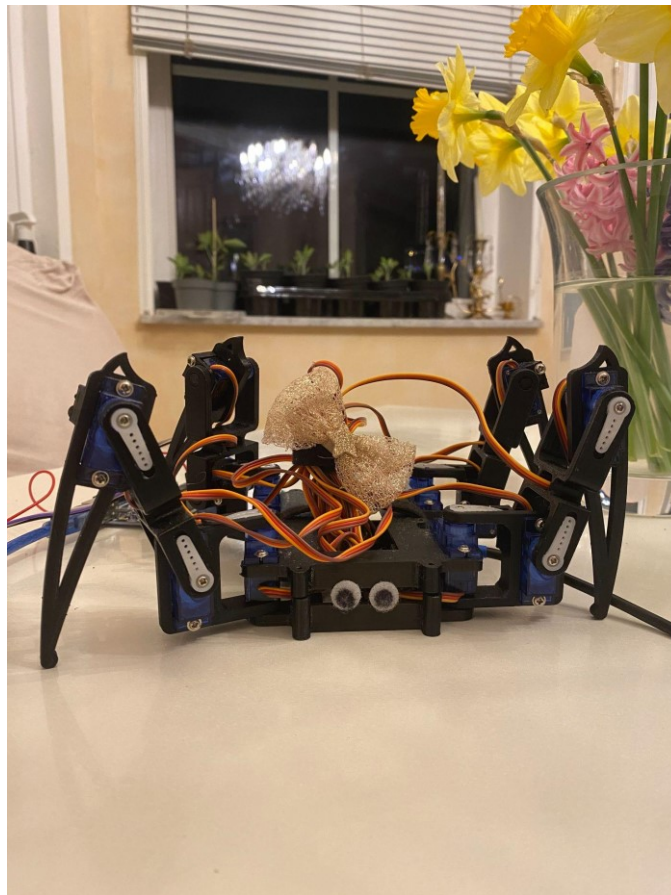


### 6.3.4 Premikanje robota

Vsak sklep na nogi robota pajka ima posebno funkcijo, ki mu omogoča premikanje in podporo telesu robota. Trije sklepi na nogi robota Pajka:

- Kolk, ki je prvi in najdebelejši sklep noge, ki se pritrjuje na telo robota. Njegova glavna naloga je zagotoviti stabilnost in podporo nogi;
- Stegnenica: Stegnenica je drugi sklep noge in je odgovoren za dvigovanje in spuščanje noge. Na goleno pritrjen s členkom, ki mu omogoča premikanje v navpični ravnini;
- Golenica: Golenica je tretji in zadnji sklep noge in je odgovorna za premikanje noge naprej in nazaj. Na stegnenico je pritrjena s sklepom, ki omogoča premikanje v vodoravni ravnini.

Ko želi robot pajek narediti korak naprej, mora biti vsak sklep noge pravilno poravnana. Kolk zagotavlja oporo in stabilnost, medtem ko stegnenica dvigne nogo, golenica pa jo premakne naprej. Ko se noga premika naprej, se morajo sklepi noge premikati usklajeno, da ohranijo ravnotežje in stabilnost. Sklep med kolkom in stegnenico ter sklep med stegnenico in golenico je treba skrbno nadzorovati, da se zagotovi nemoteno gibanje in prepreči upogibanje ali porušitev noge. Robot je prikazan na sliki 15.



Slika 15: Robot Pajek

Kratek video gibanja robota je dosegljiv na

[https://drive.google.com/file/d/1tE7OmGdylReDkJJl09ipGU\\_Q5N00Smxm/view?usp=sharing](https://drive.google.com/file/d/1tE7OmGdylReDkJJl09ipGU_Q5N00Smxm/view?usp=sharing)



### 6.3.5 Funkcija za premik naprej

Koda funkcije za premik naprej je podana v prilogi A.

#### **Opis kode:**

Koda iz priloge A je del algoritma gibanja robota pajka. Zasnovan je za premikanje štirih nog robota pajka naprej pri hoji.

Koda najprej preveri, ali trenutni položaj nog robota pajka (shranjen v spremenljivki `site_now`) kaže, da so robotove noge v začetnem položaju (ki je definiran s spremenljivko `y_start`). Če so noge v začetnem položaju, koda premakne drugo in prvo nogo naprej v nizu korakov.

Prvi korak je nastavitev položaja druge noge na dvignjen položaj s funkcijo `set_site`, ki sprejme tri parametre: ID noge, ki jo želite premakniti, koordinate X, Y in Z, na katere nastavite nogo. Nato se pokliče funkcija `wait_all_reach`, da počaka, da vse noge dosežejo svoj ciljni položaj.

Drugi korak je premakniti drugo nogo naprej in navzdol proti tlam v dveh korakih, pri čemer je `y_step` razdalja za vsak korak. Tretji korak je nastavitev položaja druge noge nazaj na privzeti položaj. Vsakemu od teh korakov sledi klic `wait_all_reach` za čakanje, da vse noge dosežejo svoj ciljni položaj.

Ko je bila druga noga premaknjena, koda nastavi spremenljivko `move_speed` na vrednost `body_move_speed`, ki je konstanta, ki določa hitrost, s katero robot pajek premika svoje telo. Koda nato nastavi položaj vseh štirih nog na način, da bo telo robota pajka premaknilo naprej, medtem ko bosta dve nogi ostali v začetnem položaju.

Ko je telo robota pajka v pravilnem položaju, koda nastavi spremenljivko `move_speed` nazaj na vrednost `leg_move_speed`, ki je konstanta, ki določa hitrost, s katero robot pajek premika svoje noge. Koda nato premakne prvo nogo naprej in navzgor ter nato nazaj navzdol na tla. Končno se prva noga vrne v privzeti položaj.

Ta koda določa funkcijo, imenovano `Step_forward`, ki robota premakne naprej z določenim številom korakov. Funkcija vzame en argument, celo število, imenovano `korak`, ki določa, koliko korakov naj naredi robot.

Funkcija je sestavljena iz zanke, ki izvaja čase korakov in v vsaki ponovitvi preveri položaj robotovih nog, da ugotovi, katere je treba premakniti.

Funkcija `set_site()` nastavlja položaje vsake noge na različnih točkah med korakom.

Parametri, posredovani funkciji, so:

- `site_id`: celo število, ki določa, katero spletno mesto je nastavljeno. Število 0 ustreza desni sprednji nogi, 1 desni zadnji nogi, 2 levi sprednji nogi in 3 levi zadnji nogi,
- `x`: vrednost s plavajočo vejico, ki določa x-koordinato položaja mesta. Ta koordinata je relativna glede na neko privzeto vrednost `x_default` plus odmik `x_offset`,
- `y`: vrednost s plavajočo vejico, ki določa y-koordinato položaja mesta. Ta koordinata je relativna glede na neko začetno vrednost `y_start` in se v nekaterih primerih poveča za `y_step`,

- `z`: vrednost s plavajočo vejico, ki določa `z`-koordinato položaja mesta. Ta koordinata je bodisi `z_default` ali `z_up`, odvisno od gibanja, ki se izvaja.

Znotraj zanke funkcija preveri trenutni položaj robota s pregledom matrike `site_now`. Če je trenutni položaj robota tak, da sta nogi 2 in 1 na tleh, funkcija izvede zaporedje gibov, zaradi katerih se bosta 2 in 1 dvignila s tal, se premikala naprej in spet pristala. Če sta nogi 0 in 3 na tleh, funkcija izvede drugačno zaporedje gibov, zaradi katerih se bosta 0 in 3 dvignila s tal, se premikala naprej in spet pristala.

Funkcija `Wait_all_reach` se pokliče po vsakem klicu `set_site`, da se zagotovi, da se je robot končal na novo mesto, preden se začne naslednje gibanje.

Funkcija prilagodi tudi spremenljivko `Move_Speed` na različnih točkah med korakom, ki nadzira hitrost, s katero se različni deli robota premikajo.

Na koncu funkcije bi moral robot narediti določeno število korakov naprej.

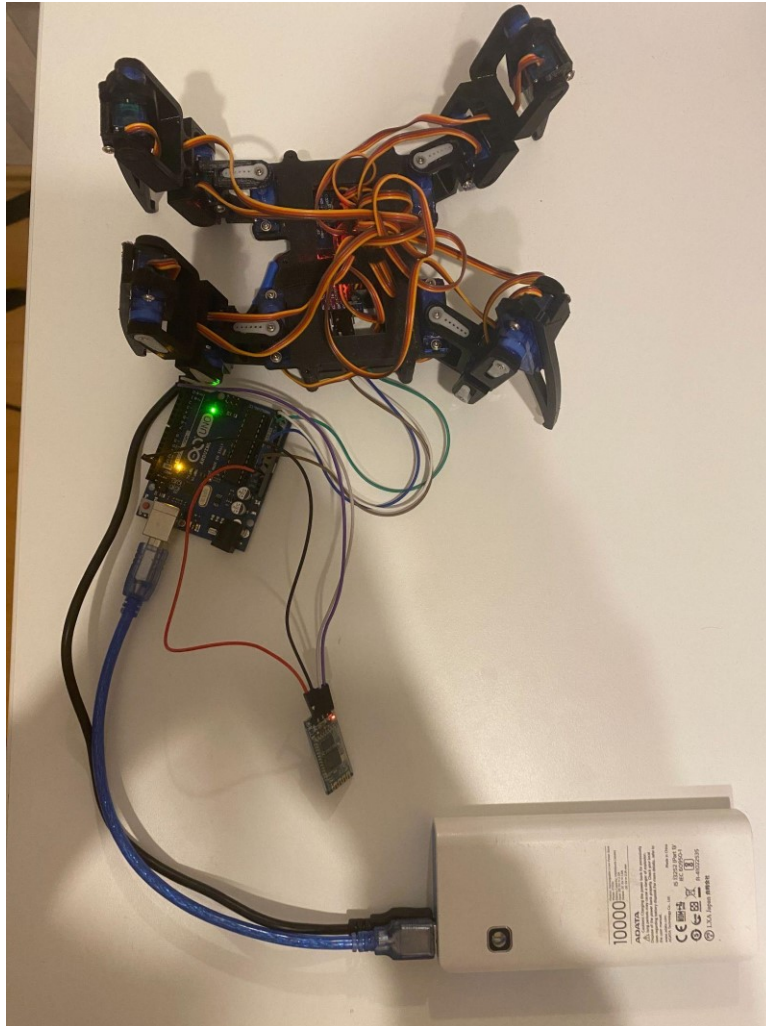
## 7 Rezultati hipotez

V nadaljevanju opišemo postopek preizkušanja delovanja robota.

### 7.1 Preizkušanje robota

Robota smo preizkusili na način, da smo na začetku robota postavili v našo bližino na varno mesto, kjer se robot ne bi zaletel v zid v bližini ali padel iz mize ali podobnega.

Na sliki 16 je vidna naša vezava s komponentami za delovanje robota. Arduino Uno in servo motorje smo povezali na prenosno baterijo.



Slika 16: Robot Pajek z baterijo

Nato, ko smo povezali vse komponente in mu dali napajanje kot tudi servo motorjem smo na računalniku naredili naslednje:

#### **Bluetooth**

Seznani smo HC-06 (zvočni krmilnik) robota Pajek z računalnikom. Uporabili smo Start → Nadzorna plošča → Strojna oprema in zvok → Naprave in tiskalniki, da preverimo, katera vrata COM so bila dodeljena HC-06.

## 7.2 Podrobnejši opis preizkusa delovanja robota

Na računalniku zaženemo kodo:

```
cd speech\api  
cmake -S. -B.build -  
DCMAKE_TOOLCHAIN_FILE=C:\vcpkg\scripts\buildsystems\vcpkg.cmake  
cmake --build .build
```

nato zaženemo kodo:

```
cd speech\api  
.build\Debug\streaming_transcribe --language-code sl-SI --bitrate 32000 COM5
```

cd speech\api ukaz nas premakne tja, kjer imamo shranjen naš program prepoznave zvoka na našem računalniku.

Z ukazom .build\ Debug\ streaming\_transcribe –language-code sl-SI —bitrate 32000 COM5 zaženemo naš program.

Uporabimo lahko sl-SI ali en (privzeto) jezikovno kodo. Uporabili smo vrata COM, ki so dodeljena HC-06 v Bluetooth.

Streaming\_transcribe bo zajemal zvok na privzeti snemalni napravi 10 minut. Lahko pa ga prekličemo s Ctrl+C.

Sledilo je preizkušanje glasovnih ukazov, katera prepoznavna se je izvršila na računalniku, ki jih je preko Bluetooth-ja po potrebi pošiljal robotu. Opazovali in ocenjevali smo pravilnost prepoznave glasovnih ukazov. Opazovali in ocenjevali smo tudi vedenje robota glede na podan ukaz.

Vsak ukaz smo preizkusili 10-krat. Rezultati prvih testiranj so bili porazni, predvsem zaradi uporabljenega mikrofona in frekvence vzorčenja. Uporabili smo profesionalni mikrofona, povečali frekvenco vzorčenja in ponovili preizkus delovanja.

Pri frekvenci vzorčenja 32 kHz in profesionalnem mikrofona smo dosegli želeno delovanje, ko je sistem zanesljivo prepoznal in izvedel ukaze ne glede na osebo, ki je glasovno krmilila sistem. Pri nižjih frekvencah sistem ni bil zanesljiv za ženske in otroške glasove.

## 7.3 Pravilnost hipotez

Za dokazovanje pravilnosti prve raziskovalne hipoteze:

**H1:** Podatki o napravi, ki krmili premikanje robota s pomočjo Arduina in servo motorjev, obstajajo in so prosto dostopni,

smo z iskanimi besedami: servo motorji, Arduino, preučili spletne informacije. Ugotovili smo, da obstajajo rešitve posameznih opravil naše (zelene) naprave, opis naprave kot celote pa nismo zasledili. Odločili smo se, da bomo posamezne rešitve vključili v našo napravo, jo izvedli ter preizkusili njeno delovanje.

Glede na navedeno prvo hipotezo označimo kot **pravilno**, saj podatki za izvedbo posameznih pravil naprave obstajajo na spletu in so prosto dostopni.

Drugo hipotezo:

**H2:** Na osnovi izsledkov raziskave hipoteze 1 je možno narediti delujočo napravo za krmiljenje robota Pajka z zvočnim vmesnikom,

smo preverili z empiričnim preizkusom delovanja robota. Preizkusili smo pravilnost delovanja glavnih opravil naprave: krmiljenje robota z zvočnim vmesnikom, izvajanje ukazov, branje in prepoznavanje ukaza po prepoznavi zvoka. Vsak ukaz smo desetkrat ponovili in spremljali delovanje robota Pajka. Končno izvedbo robota smo lahko brez težav krmilili z uporabo glasovnih ukazov. Ukazi so bili prepoznani in ustrezno izvedeni. Tako tudi drugo hipotezo označimo kot **pravilno**. Robota smo znali izvesti, ga ustrezno programirati in preizkusiti njegovo delovanje.

## 7.4 Končni sklep

Na osnovi kritične analize rezultatov testiranja robota lahko z veliko gotovostjo zaključimo, da je robota Pajka možno izvesti in ga zanesljivo glasovno krmiliti. Po našem mnenju bi z dodatkom kamer, sončnih celic in podobno robotu povečali učinkovitost, samostojnost in mobilnost. Verjetno tudi uporabnost.

## 8 Zaključek

Robotika je hitro napredujoče področje s številnimi aplikacijami v različnih panogah, kot so proizvodnja, zdravstvo, kmetijstvo, raziskovanje in zabava. Robotika lahko avtomatizira naloge, ki se ponavljajo, so nevarne ali zahtevajo visoko natančnost, s čimer se izboljša produktivnost, učinkovitost in varnost. Roboti lahko tudi dostopajo do nevarnih ali nedostopnih okolij, izvajajo zapletene operacije, spremljajo pridelke in živino, raziskujejo vesolje in pod vodo ter zagotavljajo zabavo in druženje.

Odločili smo se, da bomo te nove tehnologije kar najbolje izkoristili in naredili robota, ki bo upravljan preko zvočnih signalov z možnostjo nadgrajevanja robota v prihodnosti. Pri načrtovanju robota Pajka, njegovi izvedbi in preverjanju njegovega delovanja, smo ugotovili, da je tovrstnega robota možno načrtovati in izvesti.

Roboti pajki so lahko uporabni za naloge, ki zahtevajo okretnost, stabilnost in prilagodljivost. S svojimi številnimi nogami in sklepi lahko roboti pajki krmarijo po grobem terenu, plezajo po stenah in stropih ter se prilagajajo neravnim površinam. Uporabljajo se lahko za misije iskanja in reševanja, opazovanje in raziskovanje. Robote pajke je mogoče uporabiti tudi v izobraževanju in zabavi za poučevanje robotike ter spodbujanje ustvarjalnosti in inovativnosti.

Na splošno lahko robotika in roboti pajki revolucionirajo različne industrije in izboljšajo naša življenja na številne načine. Ker tehnologija še naprej napreduje, lahko pričakujemo bolj izpopolnjene robote z naprednimi zmogljivostmi, ki lahko opravljajo še bolj zapletene naloge.

## 9 Viri in literatura

- [1] Arduino, Spider robot, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://www.thingiverse.com/thing:2204279>
- [2] Arduino, Arduino spider robot (quadruped), (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo <https://www.instructables.com/ARDUINO-SPIDER-ROBOT-QUADRUPED/>
- [3] Earl, B., Adafruit PCA9685 16-Channel Servo Driver, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all>
- [4] Husnain, M., , (online), Serial Port Connection in C++, obiskano oktober 2022, uporabljeno december 2022, dosegljivo na [https://www.delftstack.com/howto/cpp/cpp-serial-communication/#google\\_vignette](https://www.delftstack.com/howto/cpp/cpp-serial-communication/#google_vignette)
- [5] Google Cloud, Speech-to-Text request construction, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://cloud.google.com/speech-to-text/docs/speech-to-text-requests>
- [6] Google Cloud, GoogleCloudPlatform-cpp-samples, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://github.com/GoogleCloudPlatform/cpp-samples/tree/1da4a7559eb1d7538fe41dfb71a8e1d4fd59b6/speech/api>

### 9.1 Viri slik

Slika 1: Idejna shema Pajka, lasten.

Slika 2: Elektrotehniška zasnova robota Pajka, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na [content.instructables.com/ARDUINO-SPIDER-ROBOT-QUADRUPED](https://content.instructables.com/ARDUINO-SPIDER-ROBOT-QUADRUPED).

Slika 3: Arduino Uno, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://cdn.shopify.com/s/files/1/0581/3577/3362/products/Arduino-Uno-R3-Development-Board-Microcontroller-for-DIY-Project.jpg?v=1643218499>

Slika 4: Servomotor, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://www.flyrobo.in/image/cache/catalog/sg90-9g-micro-pro-servo/sg90-9g-micro-pro-servo2-1024x1024.jpg>

Slika 5: Položaj servo motorjev na nogi, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na [https://github.com/NoAnastazija/Spider\\_robot/raw/main/doc/alignment.jpg](https://github.com/NoAnastazija/Spider_robot/raw/main/doc/alignment.jpg)

Slika 6: Priključitev servomotorja na arduino Uno, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo [https://europe1.discourse-cdn.com/arduino/optimized/4X/d/3/3/d33870595d393d77aab2b02fdf9672fed3a0a8f0\\_2\\_651x500.png](https://europe1.discourse-cdn.com/arduino/optimized/4X/d/3/3/d33870595d393d77aab2b02fdf9672fed3a0a8f0_2_651x500.png)

Slika 7: Bluetooth modul, obiskano oktober 2022, (online), uporabljeno december 2022, dosegljivo na <https://i0.wp.com/brainstormtech.in/wp-content/uploads/2021/04/hc-06-4-pin-bluetooth-rf-transceiver-module-500x500-1.jpg?fit=500%2C409&ssl=1>

## Anastazija Nograšek – Robot Pajek

Slika 8: Priključitev bluetooth modula, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na [https://www.aranacorp.com/wp-content/uploads/arduino-bluetooth-hc06\\_bb.png](https://www.aranacorp.com/wp-content/uploads/arduino-bluetooth-hc06_bb.png)

Slika 9: Servo pogon Adafruit, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na [https://grobotronics.com/images/detailed/115/htb1ww4snvxxxxcaxxxq6xxfxxd2\\_grobo.jpg](https://grobotronics.com/images/detailed/115/htb1ww4snvxxxxcaxxxq6xxfxxd2_grobo.jpg)

Slika 10: Diagram poteka robota, lasten

Slika 11: Pozicije servomotorja, lasten

Slika 12: Pozicije servo motorjev na robotu, lasten

Slika 13: Sklepi noge, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://content.instructables.com/F3N/BML7/JKN8SEFP/F3NBML7JKN8SEFP.jpg?auto=webp&frame=1&fit=bounds&md=275a1a4c8e37b39777aac9f04a7b768d>

Slika 14: Shema noge s servomotorji, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://images.interestingengineering.com/img/iev/qrwBXxVWG5/arduino-spider-all-legs.jpg>

Slika 15: Noge robota, (online), obiskano oktober 2022, uporabljeno december 2022, dosegljivo na <https://content.instructables.com/F4I/37H1/JKN8S7SC/F4I37H1JKN8S7SC.jpg?auto=webp&frame=1&width=1024&height=1024&fit=bounds&md=d45cbff689a17edd3af97d3d9e2ccd28>

Slika 16: Robot Pajek, lasten

## 9.2 Viri videov

Kratek video gibanja robota je dosegljiv na

[https://drive.google.com/file/d/1tE7OmGdylReDkJI09ipGU\\_Q5N00Smxm/view?usp=sharing](https://drive.google.com/file/d/1tE7OmGdylReDkJI09ipGU_Q5N00Smxm/view?usp=sharing)



## Priloga A

//Koda funkcije za premik naprej

```
{
  move_speed = leg_move_speed;
  while (step-- > 0)
  {
    if (site_now[2][1] == y_start)
    {
      //leg 2&1 move
      set_site(2, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(2, x_default + x_offset, y_start + 2 * y_step, z_up);
      wait_all_reach();
      set_site(2, x_default + x_offset, y_start + 2 * y_step, z_default);
      wait_all_reach();

      move_speed = body_move_speed;

      set_site(0, x_default + x_offset, y_start, z_default);
      set_site(1, x_default + x_offset, y_start + 2 * y_step, z_default);
      set_site(2, x_default - x_offset, y_start + y_step, z_default);
      set_site(3, x_default - x_offset, y_start + y_step, z_default);
      wait_all_reach();

      move_speed = leg_move_speed;

      set_site(1, x_default + x_offset, y_start + 2 * y_step, z_up);
      wait_all_reach();
      set_site(1, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(1, x_default + x_offset, y_start, z_default);
      wait_all_reach();
    }
    else
    {
      //leg 0&3 move
      set_site(0, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(0, x_default + x_offset, y_start + 2 * y_step, z_up);
      wait_all_reach();
      set_site(0, x_default + x_offset, y_start + 2 * y_step, z_default);
      wait_all_reach();

      move_speed = body_move_speed;

      set_site(0, x_default - x_offset, y_start + y_step, z_default);
      set_site(1, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(2, x_default + x_offset, y_start, z_default);  
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_default);  
wait_all_reach();
```

```
move_speed = leg_move_speed;
```

```
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_up);  
wait_all_reach();  
set_site(3, x_default + x_offset, y_start, z_up);  
wait_all_reach();  
set_site(3, x_default + x_offset, y_start, z_default);  
wait_all_reach();  
}  
}  
}
```