Upravljanje industrijskim robotom primjenom serijske veze

Puškarić, Juro

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:805660

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb







SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentori:

Doc. dr. sc. Tomislav Staroveški, dipl. ing.

Student:

Juro Puškarić

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

UPRAVLJANJE INDUSTRIJSKIM ROBOTOM PRIMJENOM SERIJSKE VEZE

Mentori:

Doc. dr. sc. Tomislav Staroveški, dipl. ing.

Student:

Juro Puškarić

Zagreb, 2019.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Veliku zahvalu želim pružiti mentoru doc. dr. sc. Tomislavu Staroveškom na stručnoj pomoći, motivaciji te uloženom trudu, vremenu i strpljenju.

Također želim zahvaliti i svojim roditeljima na strpljenju, potpori i motivaciji tijekom studija.

Juro Puškarić

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **Juro Puškarić** Mat. br.: 0035205181

Naslov rada na hrvatskom jeziku: **Upravljanje industrijskim robotom primjenom serijske veze**

Naslov rada na engleskom jeziku: **Robot control using serial communication**

Opis zadatka:

U radu je potrebno izraditi programsku podršku za serijsku komunikaciju (RS-232) s robotima proizvođača ABB koji su opremljeni S4C upravljačkim računalom. Rad je potrebno realizirati na slijedeći način:

1. Osmisliti jednostavan protokol koji bi omogućavao izdavanje naredbi za odabir koordinatnog sustava te inkrementalno i apsolutno pozicioniranje robota. Protokol bi uz navedeno također trebao imati mogućnost praćenja statusnih podataka robota.
2. Izraditi programsku podršku za serijsku komunikaciju između industrijskog PC računala i robota. Dio koda koji će se izvršavati na računalu potrebno je napisati u programskom jeziku Python.
3. Testirati sustav sa svim instrukcijama podržanim previđenim protokolom.
4. Dati zaključke rada.

Zadatak zadan:
29. studenog 2018.

Rok predaje rada:
1. rok: 22. veljače 2019.
2. rok (izvanredni): 28. lipnja 2019.
3. rok: 20. rujna 2019.

Predviđeni datumi obrane:
1. rok: 25.2. - 1.3. 2019.
2. rok (izvanredni): 2.7. 2019.
3. rok: 23.9. - 27.9. 2019.

Zadatak zadao:

Doc. dr. sc. Tomislav Staroveški

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

Sadržaj

1. UVOD 1
2. INDUSTRIJSKI ROBOTI 2
2.1 Razvoj robota 2
2.2 Generacije robota 3
3. ABB ROBOTI 4
3.1 ABB IRB 6400 4
3.1.1 Upravljačko sučelje 7
3.1.2 Koordinatni sustavi upravljačkog sustava robota 8
4. SERIJSKA KOMUNIKACIJA 10
5. UPRAVLJANJE INDUSTRIJSKIM ROBOTOM 11
5.1 Rapid programski jezik 11
5.2 S4C kontroler 12
6. IMPLENTACIJA SERIJSKE VEZE ZA UPRAVLJANJE INDUSTRIJSKIM ROBOTOM 15
6.1 Uspostavljanje komunikacije putem PC računala 15
6.2 Uspostavljanje komunikacije putem robota 18
7. ZAKLJUČAK 22

POPIS SLIKA

Slika 1. Prvi industrijski robot Unimate[2] 2
Slika 2. Prikaz stupnjeva slobode gibanja robota [5] 5
Slika 3. ABB model IRB 6400/2.4-200 6
Slika 4. Upravljačko sučelje [5] 7
Slika 5. Operatorska ploča za upravljanje [5] 8
Slika 6. Koordinatni sustavi robota [5] 8
Slika 7. Prikaz serijske komunikacije [6] 10
Slika 8. Prikaz hijerarhije programskog jezika Rapid 12
Slika 9. S4C upravljačko računalo [5] 13
Slika 10. S4C upravljačko računalo iz Laboratorija za alatne strojeve 14
Slika 11. Dijagram toka za programski kod napisan u Python-u 16
Slika 12. Otvaranje serijskog porta 17
Slika 13. Slanje tekstualnih naredbi u formatu G koda 17
Slika 14. Slanje pomaka robotu putem serijske veze 18
Slika 15. Dijagram toka za programski kod napisan u Rapid-u 19
Slika 16. Formiranje stringa te slanje povratne informacije 20
Slika 17. Programski kod za apsolutno i inkrementalno pozicioniranje robota 21

POPIS TABLICA

Tablica 1. Verzije IRB 6400 robota [5] 4

POPIS OZNAKA

Oznaka	
ABB	Asea Brown Boveri
UART	Universal asynchronous receiver-transmitter
EIA	Electronic Industries Alliance

SAŽETAK

Primjenom serijske veze potrebno je osmisliti protokol kojim bi se omogućilo upravljanje industrijskim ABB robotom IRB 6400. Serijsku komunikaciju (RS-232) potrebno je ostvariti između računala te robota koji je opremljen S4C upravljačkim računalom. Kako bi uspostavila komunikacija između robota i računala potrebno je razviti jednostavan protokol na robotu u programskom jeziku Rapid, a isto tako na računalu u programskom jeziku Python. Komunikacija će omogućavati izdavanje naredbi za odabir koordinatnog sustava, ali i apsolutnog i inkrementalnog pozicioniranja robota.

Ključne riječi: serijska veza, RS - 232 , IRB 6400, S4C

SUMMARY

Using a serial connection, it is necessary to devise a protocol that will allow the control of the industrial ABB robot IRB 6400. Serial communication (RS-232) needs to be accomplished between the computer and the robot that is equipped with the S4C control computer. In order to establish communication between the robot and the computer, it is necessary to develop a simple robot protocol in the programming language Rapid and also on a computer in the Python programming language. The communication will allow issuing commands for selecting the coordinate system, as well as absolute and incremental positioning of the robot.

Key words: serial connection, RS - 232 , IRB 6400, S4C

1. UVOD

Gledajući kroz povijest, čovjek je raznim načinima pokušavao sebi olakšati posao, a time ujedno i pokušati ostvariti u što većoj mjeri korisnost sustava. Danas, automatizaciju i modernizaciju bilo kojeg proizvodnog sustava nemoguće je zamisliti bez upotrebe industrijskog robota. Razvijanjem tehnologije te konstantom modernizacijom i automatizacijom proizvodnih procesa u svim granama industrije, raznovrsnost industrijskih robota u stalnom je porastu. Razlozi primjene industrijskih robota su višestruki, a jedan od glavnih razloga je i nemogućnost postizanja ujednačene kvalitete proizvoda. Također, robot može raditi 24 sata u danu čime u velikoj mjeri pridonosi većoj efikasnosti, no također anulira čovjekove mane poput umora, dekoncentracije i ostalih nedostataka koji se pojavljuju pri dugotrajnom monotonom poslu.

Postoje različite definicije industrijskih robota, a prema normi ISO 8373 koja se odnosi na robote i robotske uređaje, industrijski roboti predstavljaju automatski kontrolirane, reprogramibilne, višefunkcionalne manipulatore, programibilne u 3 ili više osi, koji mogu biti fiksirani na određenom mjestu ili pokretni te služe za primjenu u automatskim operacijama.

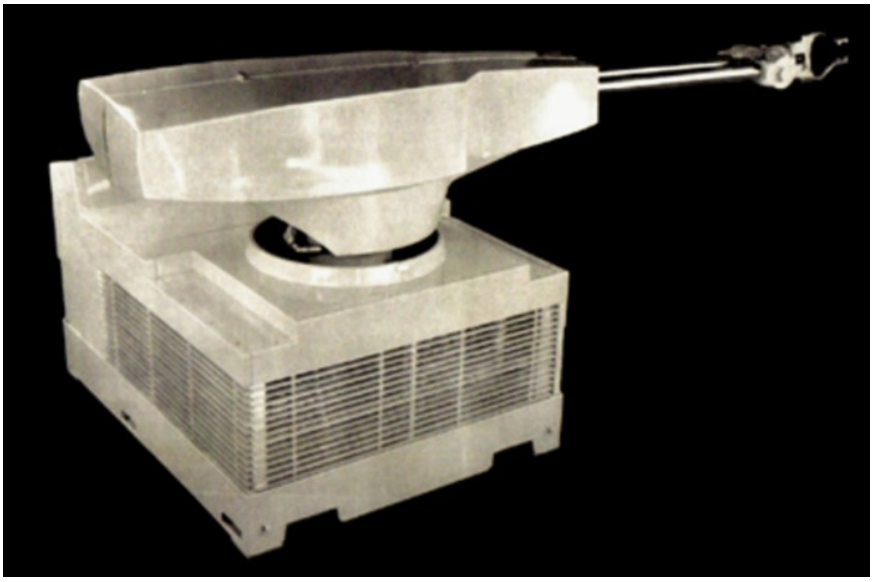
Danas, postoji nekoliko načina kojima se može upravljati industrijskim robotom, a u ovom radu cilj je postići upravljanje industrijskim robotom primjenom serijske veze. Serijsku komunikaciju potrebno je ostvariti između robota proizvođača ABB koji su opremljeni zastarjelim upravljačkim sustavom, a kojemu je jedino i tim putem odnosno serijskom vezom, omogućeno povezivanje sa nekim od nadzornih sustava te PC računala. Standard koji definira serijsku vezu u ovom radu je RS-232. Nakon ostvarenja serijske komunikacije između robota i računala, postoje neograničene mogućnosti koje je moguće implementirati na robotu.

2. INDUSTRIJSKI ROBOTI

Postoje razne definicije robotike, a jedna od njih definira robotiku kao interdisciplinarno znanstveno područje koje predstavlja kombinaciju strojarstva i znanosti o računalima, a također obuhvaća razna područja kao što su elektrotehnika, elektronika, automatizacija i umjetna inteligencija. Razvoj robotike potaknut je čovjekovim razmišljanjem da uspije pronaći zamjenu za sebe koja bi imala mogućnost oponašanja njegovih svojstava u različitim uvjetima, uzimajući u obzir i okolinu koja ga okružuje.[1]

2.1 Razvoj robota

Prvi početci razvoja industrijskih robota javljaju se početkom pojave prvih automata, a napose njihovim uvođenjem u proizvodnju u prvoj polovici 20. stoljeća. Taj razvoj također je ubrzao i razvoj numeričko upravljanih CNC alatnih strojeva. Paralelno su se počeli razvijati i manipulatori, odnosno robotske ruke s velikom slobodom pokreta, a budući da im je bila dodana mogućnost programiranja, stvoreni su prvi roboti. Prvi takav „uređaj za programirano premještanje predmeta“ projektirao je 1954. američki izumitelj Georg Devol, a 1958. u suradnji sa J. Engelbergerom izrađen je i prvi industrijski robot Unimate (slika 1). Prvi industrijski robot Unimate ujedno je bio i početak prve tvornice robota Unimation Inc, koja je dugo vremena bila vodeći proizvođač robota. [2]



Slika 1. Prvi industrijski robot Unimate[2]

2.2 Generacije robota

Danas, uz sve veći razvoj znanosti, istraživanja na robotima se uglavnom provedena u sklopu više znanstvenih disciplina kao što je robotika, kibernatika, automatizacija i dr. Pojavom prvog robota, razvoj robotike značajno napreduje pa tako dolazi do pojave razlika u stupnju autonomnosti, ali i inteligencije te mogućnosti interakcije s okolinom. Prema tome, razvoj robota može se opisati u nekoliko etapa [3]:

I. Roboti prve generacije

Prvoj generaciji robota pripadaju programirani roboti kod kojih se proces upravljanja odvija u upravljačkom lancu. Roboti prve generacije u svom sustavu ne koriste povratnu informaciju o svojem stvarnom stanju te kao posljedica toga ne mogu korigirati pogreške vođenja. Jedna od stvari po kojoj su ovi roboti poznati je da automatski ponavljaju zadani pokret, a upravljački sustav se vrlo lako prilagođava ručnim operacijama. Ti roboti su najbrojniji u tvorničkim pogonima a koriste se pretežito za prešanje, zavarivanje i sl.

II. Roboti druge generacije

Kod robota druge generacije dolazi do značajne promjene uvođenjem niza senzora kojim su roboti opremljeni kako bi imali mogućnost dobivanja povratnih informacija o svojem stvarnom stanju i stanju okoline. Dobivanjem povratne informacije roboti sad mogu korigirati pogreške vođenja, a osim toga može optimizirati procese vođenja te ga adaptirati s obzirom na promjenu stanja robota i okoline. Ovakvi roboti se uglavnom koriste za rad na pokretnoj vrpci, montažnim operacijama i sl.

III. Roboti treće generacije

Treća generacija robota je ujedno i posljednja, a njoj pripadaju inteligentni roboti koji imaju sposobnost učenja, rezoniranja te donošenja zaključaka što im omogućava da se snalaze u potpuno nepoznatoj okolini te novonastalim nepredviđenim situacijama. Dakle, sposobni su prepoznavati okolinu, analizirati svoje učinke te učiti iz svojih grešaka. Također posjeduju i visoki stupanj funkcionalne, organizacijske i mobilne autonomnosti.

3. ABB ROBOTI

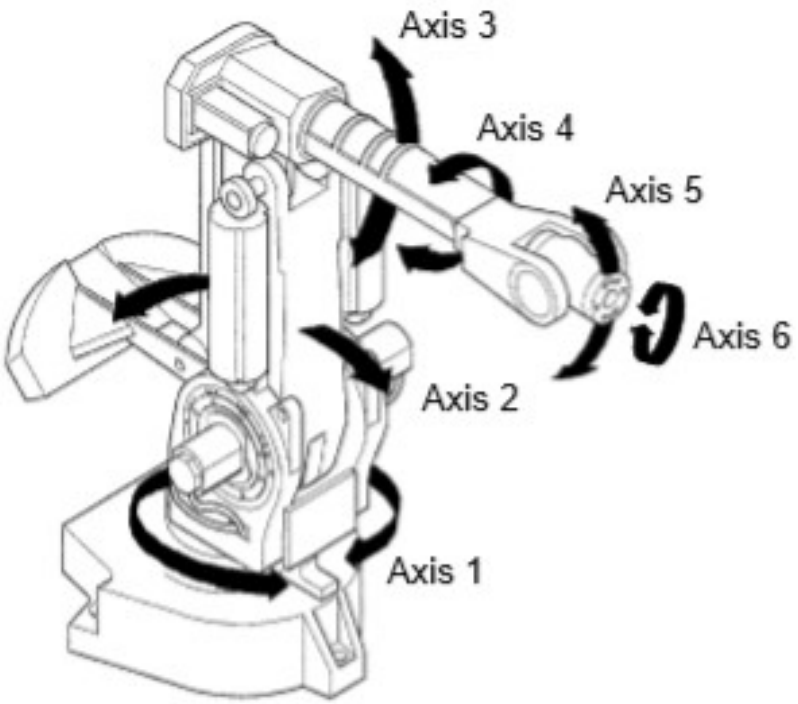
ABB je švicarsko – švedska multinacionalna kompanija smještena u Zurichu, Švicarskoj, koja djeluje uglavnom u robotici i tehnološkim područjima automatizacije i energije. ABB je prva tvrtka koja je promovirala robota kojeg u potpunosti kontrolira električni mikroprocesor. Do danas, ABB je zadržao tržišno i tehničko liderstvo s preko 175 000 prodanih robota korisnicima diljem svijeta što predstavlja jednu od najvećih instaliranih baza u industriji robotike. Koliko su zapravo ABB roboti rasprostranjeni i zastupljeni govori podatak da se gotovo svakodnevno koristi neki od proizvoda koji su proizveli ABB roboti. [4]

3.1 ABB IRB 6400

ABB robot IRB 6400 ima 6 stupnjeva slobode gibanja kao što je prikazano na slici 2. , a upravo je taj model konstruiran specifično za proizvodnu industriju koja koristi fleksibilnu robotsku automatizaciju. Robot IRB 6400 ima otvorenu strukturu koja je posebno prilagođena za fleksibilnu upotrebu, a također može komunicirati sa vanjskim sustavima.
Model IRB 6400 dolazi u nekoliko različitih verzija koji su prikazani u tablici 1. U ovome radu korišten je model IRB 6400/ 2.4- 200, a to govori da robot može raditi sa masama i do 200 kg a maksimalni doseg iznosi 2,4 metra. Model IRB 6400/ 2.4- 200 prikazan je na slici 3.

Tablica 1. Verzije IRB 6400 robota [5]

Robot Versions	
IRB 6400/ 2.4-120	IRB 6400F/ 2.4-120
IRB 6400/ 2.4-150	IRB 6400F/ 2.4-150
IRB 6400/ 2.4-200	IRB 6400F/ 2.4-200
IRB 6400/ 2.8-120	IRB 6400F/ 2.8-120
IRB 6400/ 3.0-75	IRB 6400F/ 3.0-75
IRB 6400S/ 2.9-120	IRB 6400FS/ 2.9-120
IRB 6400PE/ 2.25 -75	IRB 6400FHD



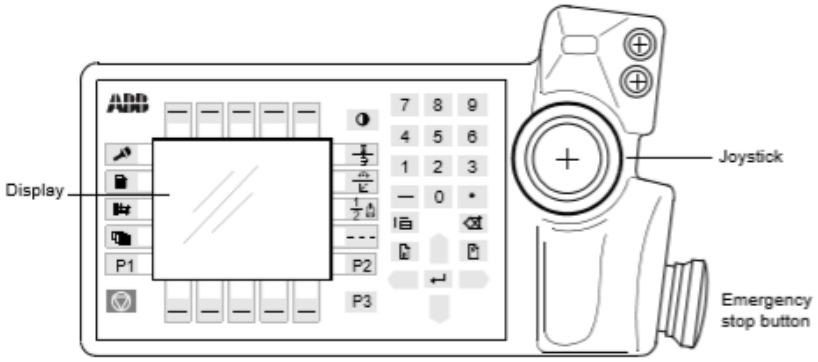
Slika 2. Prikaz stupnjeva slobode gibanja robota [5]



Slika 3. ABB model IRB 6400/2.4-200

3.1.1 Upravljačko sučelje

Na robotu IRB 6400 sve operacije te sve programirane kretnje mogu biti izvedene koristeći terminal odnosno upravljačko sučelje(eng. Teach Pendant) (slika 4.).

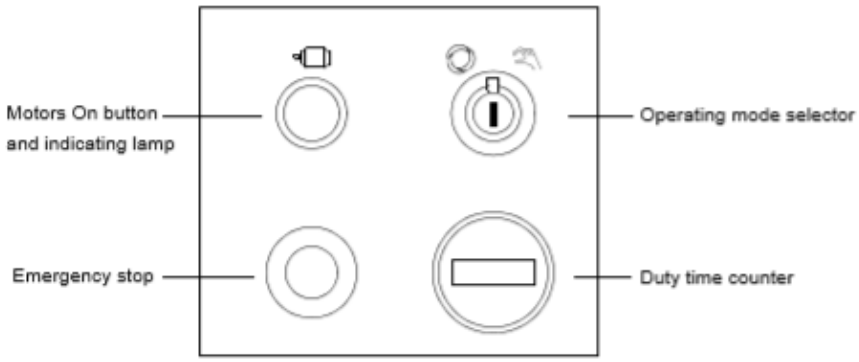


Slika 4. Upravljačko sučelje [5]

Upravljačko sučelje opremljeno je ekranom, koji pokazuje grafičko sučelje te na njemu sve upite, greške te informacije. Koristeći upravljačko sučelje, robot se može i ručno pokretati, a sam korisnik odabire brzinu kojom će se robot kretati. Kao zaštita od neplaniranih gibanja robota, koja bi mogla naštetiti čovjeku i okolini postoji udarno tipkalo za zaustavljanje u nuždi(eng. emergency stop button) što je prikazano na slici 4. Prije svakog kretanja robota potrebno je držati sigurnosni gumb koji se nalazi na stražnjoj strani upravljačkog sučelja, koji onda omogućava gibanje robota. Puštanjem sigurnosnog gumba sve kretnje robota su obustavljanje.

Koristeći ključić koji se nalazi na elektro ormaru, operator može odabrati nekoliko radnih načina (slika 5.) u kojem će robot obavljati zadane naredbe, a to su [5]:

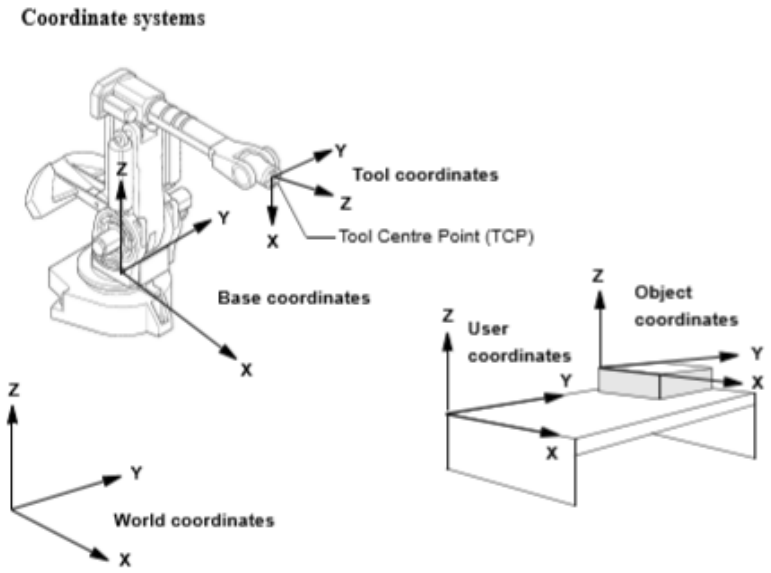
- Automatski način – Robot se giba na način kojim je programiran te operator ne može utjecati na njega
- Ručni način sa smanjenom brzinom



Slika 5. Operatorska ploča za upravljanje [5]

3.1.2 Koordinatni sustavi upravljačkog sustava robota

Prilikom upravljanja robotom IRB 6400, gibanja robota razlikuju se ovisno o koordinatnom sustavu u kojem se robot nalazi. Postoje više koordinatnih sustava koji su prikazani na slici 6., a prema [5] su objašnjeni u nastavku.

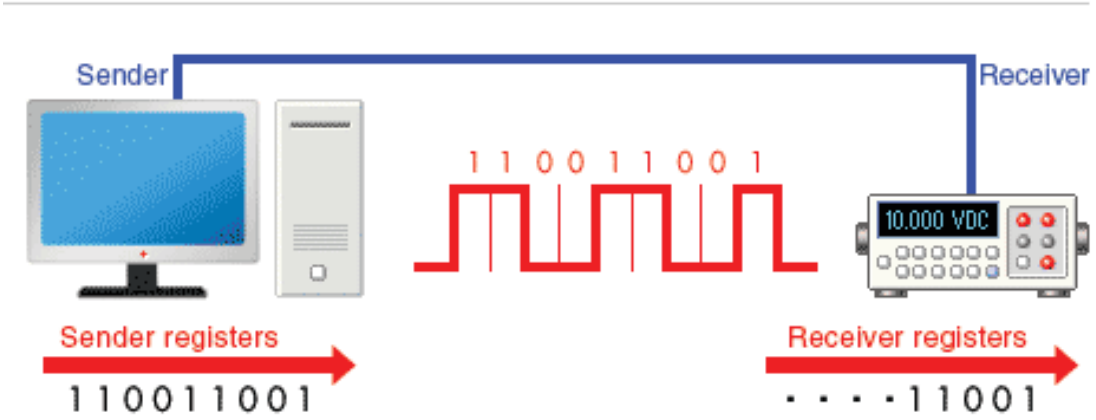


Slika 6. Koordinatni sustavi robota [5]

- Globalni koordinatni sustav – glavni koordinatni sustav pokazuje referencu s nepomičnom nultočkom u temelju robota, a služi kao početna točka za ostale koordinatne sustave. Koristeći ovaj koordinatni sustav moguće je povezati poziciju robota sa tvorničkom nultočkom
- Bazni koordinatni sustav – koordinatni sustav vezan za bazu robota
- Koordinatni sustav alata – koordinatni sustav alat definira središnju točku alata(eng. tool centre point) i orijentaciju alata
- Koordinatni sustav korisnika – koordinatni sustav korisnika definira poziciju i položaj robotskog manipulatora
- Koordinatni sustav objekta – koordinatni sustav koji definira poziciju i položaj radnog komada

4. SERIJSKA KOMUNIKACIJA

Serijska komunikacija je komunikacijska metoda koja koristi jednu ili dvije prijenosne linije za slanje i primanje podataka. Putem serijske komunikacije ostvaruje se razmjena podataka između dvaju uređaja, drugim riječima, podatci se šalju bit po bit. Serijskom komunikacijom obično upravljaju specijalizirani logički sklopovi, poznatiji kao univerzalni asinkroni primopredajnici(UART), koji zatim obrađuju formatiranje, premještanje i obnavljanje podataka na logičkoj razini. UART se koristi zajedno sa RS-232, a služi za pretvaranje podataka između serijskih i paralelnih formata. UART konfigurira brzinu prijenosa te format slanja podataka, a za pravilan rad nužno je da UART na strani primatelja i na strani odašiljača bude postavljen na iste parametre veze. Na slici 7. nalazi se jednostavan prikaz serijske komunikacije[6].



Slika 7. Prikaz serijske komunikacije [6]

Iako se ovaj rad većim dijelom bavi komunikacijom s robotom zasnovanom na RS-232 vezi, sličnu vezu robot IRB 6400 može ostvariti ostvariti putem RS-422 veze ili putem Etherneta, koji pak za svoju komunikaciju zahtijeva poseban softver.

5. UPRAVLJANJE INDUSTRIJSKIM ROBOTOM

5.1 Rapid programski jezik

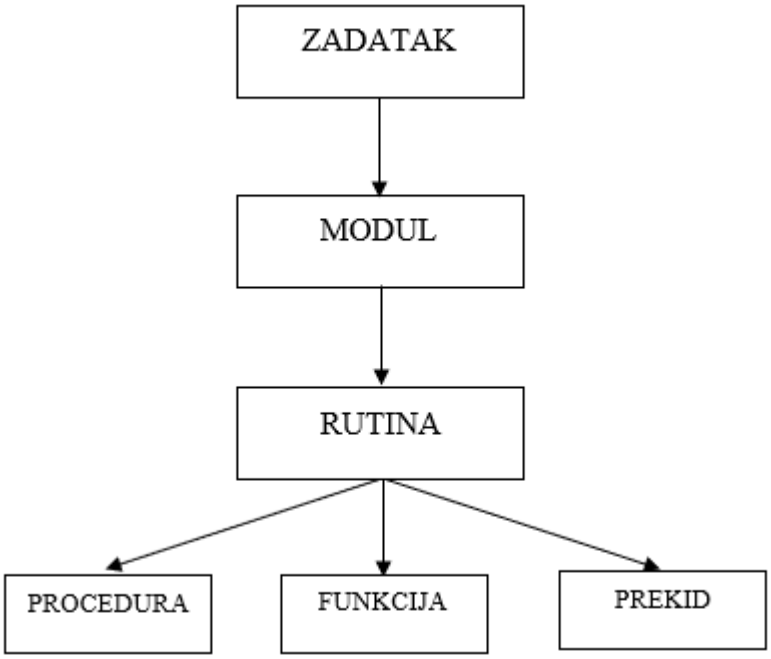
Upravljanje industrijskim ABB robotom u ovome radu se ostvaruje pomoću Rapid programskog jezika te S4C kontrolnog sustava.

U Rapidu postoji niz naredbi koji opisuju način na koji robot treba izvršavati određenu radnju. Ono što je važno napomenuti je da su programi hijerarhijski podijeljeni na:

- Zadatke
- Module
- Rutine

Zadatak je najviša razina hijerarhije programa. Zadatci se sastoje od modula koji omogućavaju lakšu organizaciju koda te bolje snalaženje u programu. Moduli se zatim sastoje od rutina koje predstavljaju najnižu razinu hijerarhije programa. Rutine se dijele u 3 podgrupe a to su: procedure, funkcije i prekidi.

Procedure su potprogrami koje se mogu pozivati bilo gdje u programu, dok funkcija je rutina koja vraća vrijednost neke vrste a uglavnom se koristi kao argument za izvođenje neke matematičke funkcije. Prekidi(eng. Trap) je vrsta rutine koja se izvodi neovisno o glavnom program, a može izazvati prekid u bilo kojem trenutku izvođenja. Gore opisana hijerarhija prikazana je na slici 8. [7]

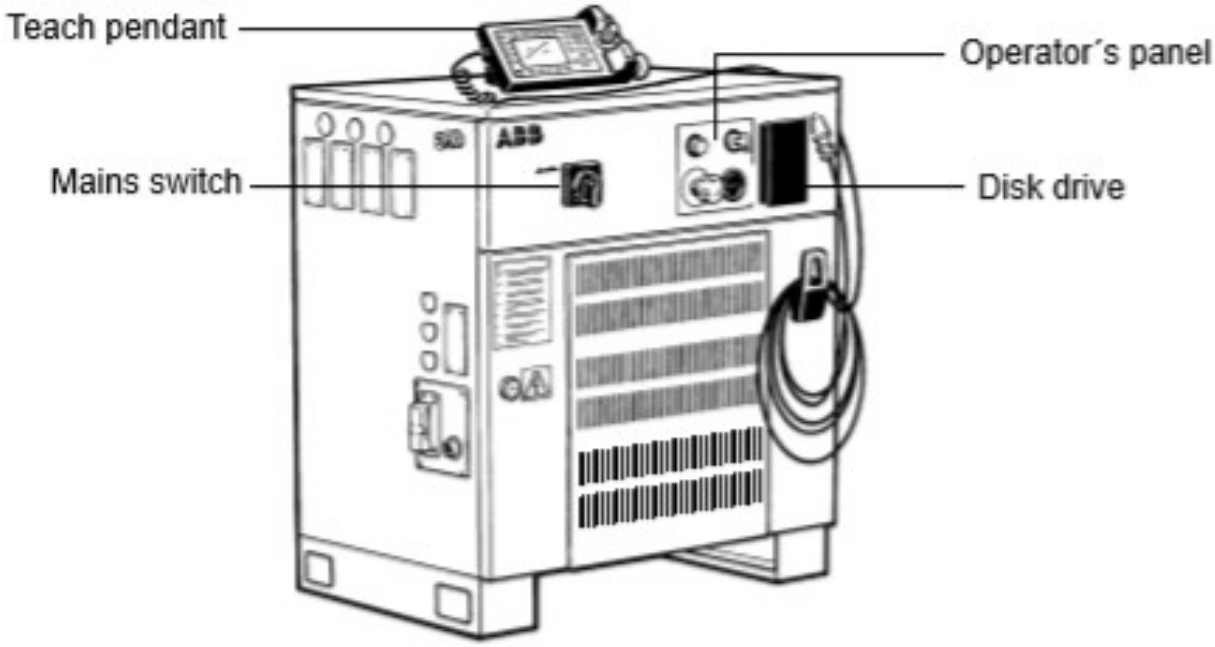


Slika 8. Prikaz hijerarhije programskog jezika Rapid

5.2 S4C kontroler

ABB-ovo S4C upravljačko računalo ustvari je zamjena S4 upravljačkog računala, koji je proizveden kako bi bio manja i niža verzija. Sam naziv S4C u sebi sadrži slovo C što predstavlja kompaktnost (engl. Compact). Većina računalnog sustava ostala je ista sa tek ponešto ažuriranja hardvera i softvera. S4C upravljačko računalo je konfigurabilno, a posjeduje višeprocesorski sustav koji može kontrolirati jednu ruku robota ili upravljati cijelom radnom ćelijom. Ukoliko je potrebno, može se proširiti nizom analognih ili digitalnih ulaza i izlaza. [8]

Na slici 9. prikazano je upravljačko računalo S4C, a na slici 10,. upravljačko računalo iz Laboratorija za alatne strojeve.



Slika 9. S4C upravljačko računalo [5]



Slika 10. S4C upravljačko računalo iz Laboratorija za alatne strojeve

6. IMPLENTACIJA SERIJSKE VEZE ZA UPRAVLJANJE INDUSTRIJSKIM ROBOTOM

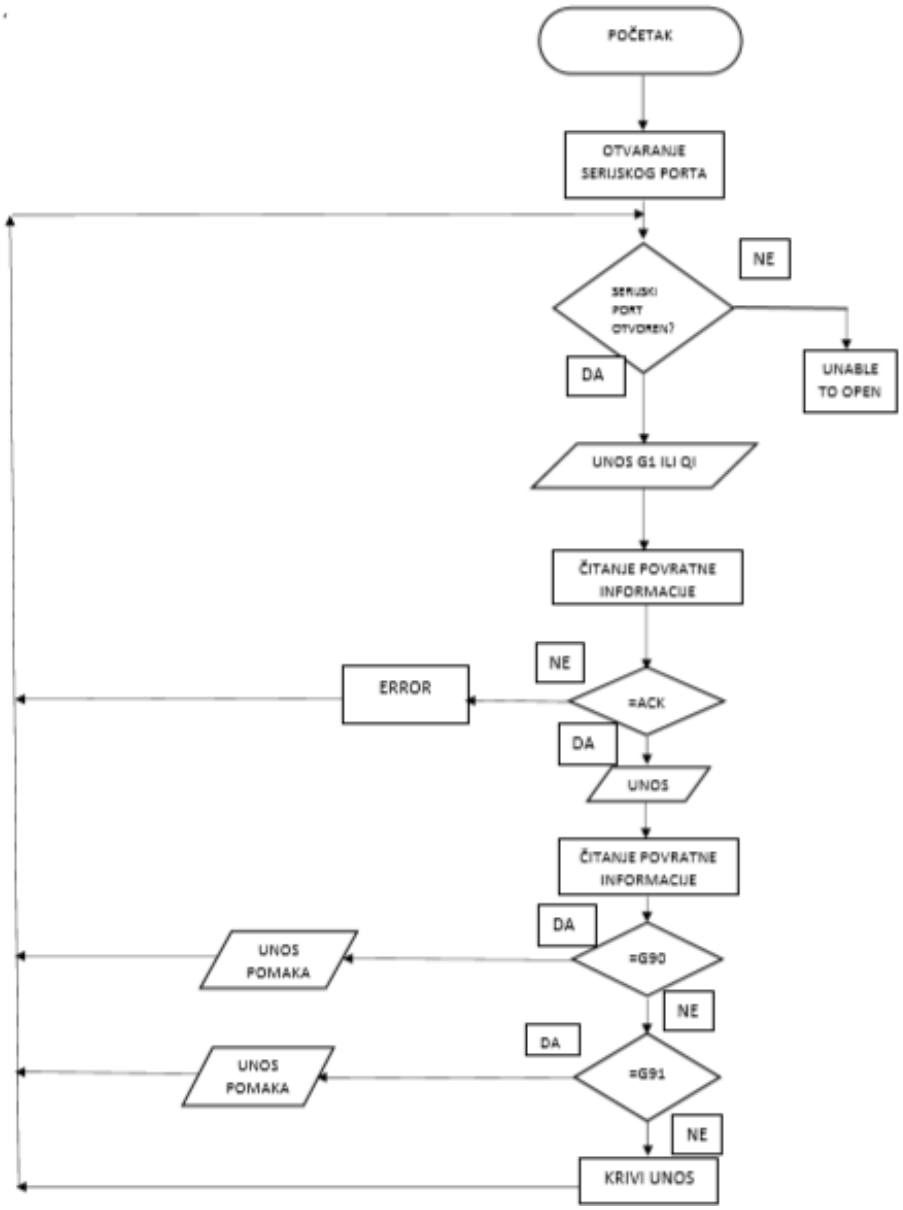
Za upravljanje industrijskim robotom, potrebno je izraditi programsku podršku za komunikaciju između industrijskog PC računala i robota. Program koji će se izvršavati na PC računalo napisan je u programsku jeziku Python. Python je interpreterski, interaktivni, objektno orijentirani programski jezik, kojeg je 1990. godine prvi razvio Guido van Rossum. Već do konca 1998., Python ima bazu od 300 000 korisnika, a od 2000. već je prihvaćen od strane ustanova kao MIT, NASA, IBM, Google, Yahoo i drugih. Python ne donosi neke nove revolucionarne značajke u programiranju, već na optimalan način ujedinjuje sve najbolje ideje i načela rada drugih programskih jezika. U neku ruku može se smatrati hibridom: nalazi se između tradicionalnih skriptnih jezika (kao što su Tcl, Schema i Perl) i sistemskih jezika (kao što su C, C++ i Java). To znači da nudi jednostavnost i lako korištenje skriptnih jezika (poput Matlab-a), uz napredne programske alate koji se tipično nalaze u sistemskim razvojnim jezicima.[9]

Radi lakše interpretacije i razumijevanja komunikacije između robota i računala, kodovi su prikazani putem dijagrama tokova koji se nalaze na sljedećim stranicama. Slika 11. predstavlja dijagram toka za dio koda napisan u Python-u .

6.1 Uspostavljanje komunikacije putem PC računala

Prvi korak, potreban za ostvariti komunikaciju putem PC računala, je otvoriti serijski port. Preko tog porta će se odvijati daljnja komunikacija između robota i računala.

Nakon što je serijski port otvoren, slijedi provjera. Provjera će se izvršiti pomoću naredbe IF te ako je serijski port zaista otvoren, komunikaciju je moguće izvršiti. Kada je komunikacija uspostavljenja, moguće je početi sa slanjem tekstualnih naredbi nalik formatu G koda. Gore navedeni postupak, prikazan je na slici 12.



Slika 11. Dijagram toka za programski kod napisan u Python-u

```
import serial
import sys
import time
from threading import Thread

ser = serial.Serial()
ser.baudrate = 19200
ser.port = 'COM3'
print ("Serial port object: %s" % (ser))
ser.open()
if ser.isOpen:
    print ("SERIAL PORT OPEN")
    try:
        while ser.isOpen:
```

Slika 12. Otvaranje serijskog porta

Sljedeći korak je slanje tekstualnih naredbi u formatu G koda. Najprije je potrebno odrediti način gibanja. Ova linija koda ujedno je i mjesto gdje korisnik može završiti sa pozicioniranjem robota tako što unosi string „QI“. String „QI“ označava prekid unosa tekstualnih naredbi u formatu G koda. U ovom radu se koristi linearno gibanje pa sukladno tome, tekstualna naredba koja će se poslati za nastavak pozicioniranja, je „G1“. Nakon poslane prve tekstualne naredbe, potrebno je pričekati povratnu informaciju od strane robota koja treba glasiti „ack“. Nakon što se pročita povratna informacija slijedi provjera da li uistinu ta povratna informacija odgovara željenom stringu. Ukoliko povratna informacija zadovoljava provjeru, program dalje nastavlja sa radom. Nakon što se odabrao način gibanja, slijedi odabir apsolutnog ili inkrementalnog pozicioniranja. Također se unosi tekstualna naredba u formatu G koda. Za odabir apsolutnog pozicioniranja potrebno je unijeti „G90“ odnosno „G91“ ukoliko korisnik želi odabrati inkrementalno pozicioniranje. Gore navedeni postupak prikazan je na slici 13.

```
while ser.isOpen:
    _txt_to_send = raw_input("Enter G code cmd:")
    print ("SENDING: %s" %(_txt_to_send))
    ser.write(_txt_to_send)
    print ("Waiting for response...")
    gfirst=ser.read(3)
    print type(gfirst), len(gfirst), gfirst
    if gfirst == "ack":
        print("Received OK response")
        _txt_to_sendl=raw_input("Unesi G90 ili G91:")
        print ("SENDING: %s" % (_txt_to_sendl))
        ser.write(_txt_to_sendl)
        print("Waiting for response...")
```

Slika 13. Slanje tekstualnih naredbi u formatu G koda

Nakon što je korisnik odabrao željeno pozicioniranje, ponovno se čeka povratna informacija koja se zatim provjerava za željenim unosom. Razlog ovoliko čestih provjera leži u tome, da se upravo ovim načinom eliminira moguća pogreška u komunikaciji koja kasnije može dovesti do raznih posljedica. Nakon uspješne provjere, unosi se pomak robota po x, y i z smjeru. Nakon što je korisnik unio potrebne vrijednosti, one se putem serijske veze šalju robotu. Gore navedeni postupak prikazan je na slici 14.

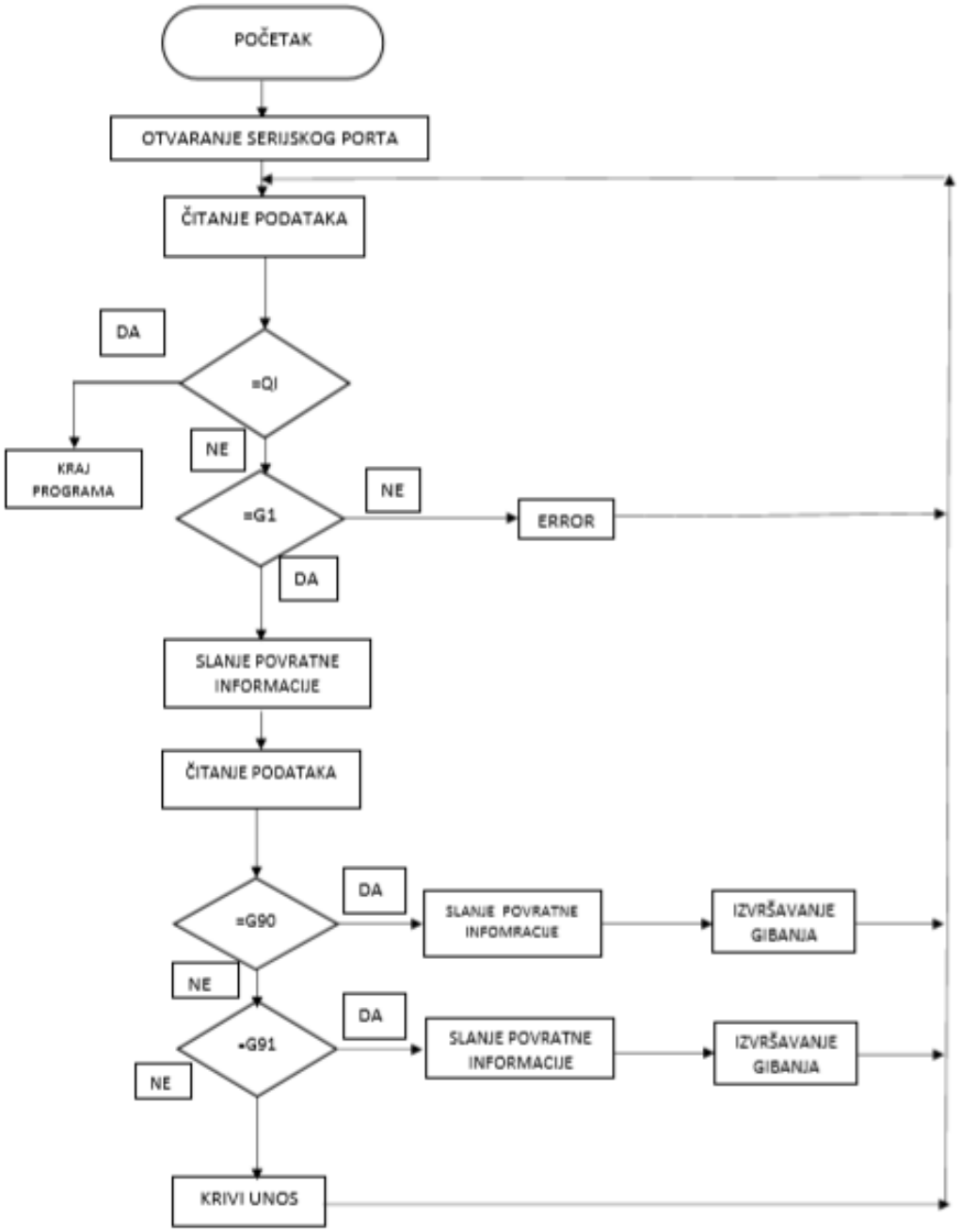
```
print("Waiting for response...")
gsecond=ser.read(3)
print type(gsecond), len(gsecond), gsecond
if gsecond == "G90":
    print("Naredba je G90")
    print("Waiting for instructions")
    x = raw_input("unesi pomak po x")
    y = raw_input("unesi pomak po y")
    z = raw_input("unesi pomak po z")
    ser.write(x)
    ser.write(y)
    ser.write(z)
elif gsecond == "G91":
    print("Naredba je G91")
    print("Waiting for instructions")
    x = raw_input("unesi pomak po x")
    y = raw_input("unesi pomak po y")
    z = raw_input("unesi pomak po z")
    ser.write(x)
    ser.write(y)
    ser.write(z)
```

Slika 14. Slanje pomaka robotu putem serijske veze

Slanjem pomaka završava se ciklus u programskom jeziku Python, a petlja se ponavlja dokle god je serijski port otvoren.

6.2 Uspostavljanje komunikacije putem robota

Na slici 15. prikazan je dijagram toka za dio koda napisan u Rapid-u. Nakon otvaranja porta, koji je potrebno otvoriti i sa strane robota, u Rapidu se najprije izvršava provjeru unosa poslanog sa strane računala. Kako se komunicira putem serijske veze, nužno je da se formira string iz niza bajtova, koji pristižu serijskim portom. Kako bi dobili željeni string, bajtove koji pristižu, potrebno je pročitati sa serijske veze pomoću naredbe ReadBin te spremiti u varijablu. Bajtovi se, pomoću naredbe ByteToStr, pretvaraju u string. Ukoliko se to napravi za svaki dolazni bajt, formira se dobiveni string.



Slika 15. Dijagram toka za programski kod napisan u Rapid-u

Nakon što su podaci uspješno pročitani, a string formiran slijedi provjera. Prva provjera se odnosi na nastavak ili prekid programa. Ukoliko formirani string odgovara „QI“, program se prekida. U suprotnom, program se ponavlja dokle god se formirani string razlikuje od „QI“. Može se reći da „QI“ u pravilu označava izlazak iz programa. Sljedeća provjera odnosi se na tekstualnu naredbu vezanu za linearno gibanje. Naime, prema gore opisanom postupku provodi se provjera za „G1“. Nakon uspješne provjere, sa robota se šalje povratna informacija o uspješnoj komunikaciji. Gore navedeni postupak prikazan je na slici 16.

```
PROC main()
        Open "sio1:", channel\Bin;
        input{1}:=ReadBin(channel);
        ser1:=ByteToStr(input{1}\char);
        input{2}:=ReadBin(channel);
        ser2:=ByteToStr(input{2}\char);
        ser:=ser1+ser2;
        home:=CRobT(\Tool:=t_flandja \WObj:=wobj_refr);
        WHILE ser <> "QI" DO
                IF ser = "G1" THEN
                        TPWrite "Na serijalcu pise "+ ser1 + ser2;
                        out_buffer{1}:= StrToByte("a"\Char);
                        out_buffer{2}:= StrToByte("c"\Char);
                        out_buffer{3}:= StrToByte("k"\Char);
                        WriteBin channel, out_buffer,3;
```

Slika 16. Formiranje stringa te slanje povratne informacije

Nakon poslane povratne informacije, potrebno pročitati da li se radi o apsolutnom ili inkrementalnom pozicioniranju. Postupak kojim se formira string opisan je na prethodnom primjeru. Kod apsolutnog pozicioniranja robot se pomiče za željeni pomak u x, y i z smjeru od fiksne početne točke te svako sljedeće gibanje odnosi se na upravo tu fiksnu početnu točku. Kod inkrementalnog pozicioniranja, za razliku od apsolutnog, početna točka nije fiksna. Početna točka je naime zadnji položaj robota, te se sljedeće gibanje odnosi relativni pomak u odnosu na zadnji položaj robota. Zadnji položaj robota omogućuje naredba CRobT koja sprema trenutni položaj robota, a kasnije će upravo taj položaj biti početna točka. Linearno gibanje robota se ostvaraje pomoću naredbe MoveLOffs koja onda pomiče robot za pomak koji je korisnik unio u x, y i z smjeru. Ovime se završava gibanje robota te korisnik može pomicati robota, onoliko puta koliko želi, tekstualnim instrukcijama nalik G kodu. Navedeni postupak prikazan je na slici 17.

```
input{1}:=ReadBin(channel);
ser1g:=ByteToStr(input{1}\char);
input{2}:=ReadBin(channel);
ser2g:=ByteToStr(input{2}\char);
input{3}:=ReadBin(channel);
ser3g:=ByteToStr(input{3}\char);
serg:= ser1g + ser2g + ser3g;
IF serg = "G90" THEN
        out_buffer{1}:= StrToByte("G"\Char);
        out_buffer{2}:= StrToByte("9"\Char);
        out_buffer{3}:= StrToByte("0"\Char);
        WriteBin channel, out_buffer,3;
        input{1}:= ReadBin(channel);
        x1:= ByteToStr(input{br}\char);
        ok:=StrToVal(xc,x);
        input{1}:= ReadBin(channel);
        y1:= ByteToStr(input{1}\char);
        ok2:=StrToVal(y1,y);
        input{1}:= ReadBin(channel);
        zs:= ByteToStr(input{1}\char);
        ok3:=StrToVal(zs,z);
        TPWrite "Pomak po x " + xc + "Pomak po y " + y1 + "Pomak po z " +  zs;
        MoveL Offs(home, x, y , z), v50, fine, t_flandja\Wobj:=wobj_refr;
        WaitUntil\ inpos,TRUE;
ELSEIF serg = "G91" THEN
        out_buffer{1}:= StrToByte("G"\Char);
        out_buffer{2}:= StrToByte("9"\Char);
        out_buffer{3}:= StrToByte("1"\Char);
        WriteBin channel, out_buffer,3;
        input{1}:= ReadBin(channel);
        x1:= ByteToStr(input{1}\char);
        input{1}:= ReadBin(channel);
        y1:= ByteToStr(input{1}\char);
        input{1}:= ReadBin(channel);
        zs:= ByteToStr(input{1}\char);
        TPWrite "Pomak po x " + x1 + "Pomak po y " + y1 + "Pomak po z " +  zs;
        p:=CRobT(\Tool:=t_flandja \WObj:=wobj_refr);
        MoveL Offs(p, x, y , z), v50, fine, t_flandja\Wobj:=wobj_refr;
        WaitUntil\ inpos,TRUE;
```

Slika 17. Programski kod za apsolutno i inkrementalno pozicioniranje robota

7. ZAKLJUČAK

Cilj ovoga rad bio je, primjenom serijske veze između industrijskog PC računala i robota, omogućiti izdavanje tekstualnih naredbi za odabir koordinatnog sustava te inkrementalnog i apsolutnog pozicioniranja robota.

Robot IRB 6400 s upravljačkim sustavom, koji danas spada pod robote starije generacije, uspješno je povezan s nadzornim računalom putem serijske veze. U sklopu ovoga rada, formirano je niz instrukcija koje omogućavaju apsolutno i inkrementalno pozicioniranje robota. Same instrukcije izrađene su na način da svojim formatom odgovaraju sintaksi G koda.

U realizaciji ovoga rada bilo je potrebno napisati dva programa. Program koji se izvršava na računalu napisan je u programskom jeziku Python dok je program za gibanje robota napisan u u programskom jeziku Rapid.

Prikazani sustav uspješno je testiran u Laboratoriju za alatne strojeve te se pokazalo kako sustav ispravno funkcionira.

LITERATURA

[1] http://people.etf.unsa.ba/~jvelagic/laras/dok/Robotika_uvod.pdf. (datum posjeta 05.07.2019.)

[2] T. Šurina, M. Crneković, Industrijski roboti, školska knjiga, Zagreb, 1990.

[3] http://www.enciklopedija.hr/natuknica.aspx?ID=53100 (datum posjeta: 07.07.2019.)

[4] https://new.abb.com/ (datum posjeta: 12.07.2019.)

[5] http://robotforum.ru/assets/files/ABB_pdf/6400_all.pdf

[6] https://www.contec.com/support/basic-knowledge/daq-control/serial-communicatin/ (datum posjeta: 23.07.2019.)

[7] Filipović Ivan: ″Robotska stanica za posluživanje stroja za injekcijsko prešanje“ Diplomski rad, Fakultet strojarstva i brodogradnje, Zagreb, 2017.

[8] https://library.e.abb.com/public/b0cf5a85b4b7e911c1257b4b00521469/EngS4Cplus_spec_Rev4.pdf

[9] Mario Essert: Python, Digitalni udžbenik

PRILOZI

I. CD-R disc