

# Osnovne upute

Koristiti stanicu [Roboguide stanicu 9.40 LR Mate 200iD7L](#) s e-učenja za Roboguide-FANUC zadatke.

Zadatke rješavati samostalno. Programski kod je potrebno komentirati.

Bodovanje i skala ocjenjivanja:

Broj bodova	Ocjena
51-64	Dovoljan (2)
65-80	Dobar (3)
81-90	Vrlo dobar (4)
91-125	Izvrstan (5)

Zadatak	Broj bodova
Projektni zadatak 1 (RoboDK) - 1. zadatak	15
Projektni zadatak 1 (RoboDK) - 2. zadatak	15
Projektni zadatak 1 (RoboDK) - 3. zadatak	25
Roboguide-FANUC zadatak 1	15
Roboguide-FANUC zadatak 2 (15+15)	30
Roboguide-FANUC zadatak 3	25
<b>Ukupno:</b>	<b>125</b>

# Projektni zadatak (RoboDK) - 1. zadatak – 15 bodova

1. Potrebno je odabrati proizvoljnog robota te mu zadati dva *target*-a. Između dva targeta potrebno je programirati linearno gibanje (*MoveL*), gibanje interpolacijom zglobova (*MoveJ*) te kružno gibanje (*MoveC*). Prilikom sva tri tipa gibanja potrebno je spremati položaj matrice i pozicije svakog zgloba. Potom je potrebno plotati 6 grafova za položaj vrha alata:  $\mathbf{x}(t) - t$  gdje je

$$\mathbf{x}(t) = [x(t) \quad y(t) \quad z(t) \quad r_x(t) \quad r_y(t) \quad r_z(t)]^T.$$

Takoder, potrebno je plotati 6 grafova za pozicije zglobova  $\mathbf{q}(t) - t$ , 6 grafova za brzine zglobova  $\dot{\mathbf{q}}(t) - t$  i 6 grafova za akceleracije zglobova  $\ddot{\mathbf{q}}(t) - t$  gdje je

$$\mathbf{q}(t) = [q_1(t) \quad q_2(t) \quad q_3(t) \quad q_4(t) \quad q_5(t) \quad q_6(t)]^T \text{ itd.}$$

# Projektni zadatak (RoboDK) - 2. zadatak – 15 bodova

2. Za zadanu stanicu u prilogu zadatka, potrebno je napraviti program korištenjem *Python RoboDK API* modula. Stanica je zadana te ju nije potrebno raditi. Zadan je video kako bi stanica trebala odradivati zadatak.

<https://e-ucenje.fsb.hr/mod/folder/view.php?id=33663>

# Projektni zadatak (RoboDK) - 3. zadatak – 25 bodova

3. Potrebno je proučiti dolje navedeni *add-in* i napraviti vlastitu stanicu s primjenom istoga kako bi se vidjele sve funkcionalnosti. Također, potrebno je tekstualno i sa slikama opisati korištenje i primjenu navedenog *add-in*-a.

- (a) Machining Tools (Nikola Akrap)
- (b) Sensor2D (Antonio Ćuk)
- (c) Surface Pattern Generator (Bruno Gugo)
- (d) Weave Generator (Fran Haraminčić)
- (e) Welding (Krešimir Hartl)
- (f) Collision Free Planner (Antonio Jurić)
- (g) Cable Simulation (Ivan Noršić)
- (h) Collision Sensor (Matija Pongračić)
- (i) Coordinated Motion (Demion Šavrljuga)
- (j) Lock TCP (Lovro Šibenik)

# Roboguide-FANUC zadatak 1

## ZADATAK:

Napisati KAREL program koji će generirati koordinate za 16 paletnih mjesta i spremiti ih u pozicijske registre od PR[20] do PR[35].

U istom KAREL programu napraviti mogućnost sljedećih odabira (proizvoljnim redoslijedom, napraviti uputu kod unosa kratkim ispisom na ekran – User screen):

- Identifikator paletnog mjesta (1-16)
- Aktivni korisnički koordinatni sustav (8 ili 9)
- Brzina izvođenja gibanja u rasponu cjelobrojnih vrijednosti (INTEGER) [10-100]

Nakon sva tri uspješna unosa koji moraju programski biti ograničeni gornjim rasponima cjelobrojnih vrijednosti, putem naredbe za poziv programa CALL\_PROG() pozvati TP/LS program pomoću kojeg robot izvodi Joint kretnju u zadanom koordinatnom sustavu i zadanom postotnom brzinom (zapisanima u proizvoljne registre R[x]) te se giba u koordinatu u odabranom PR-u.

Nakon što se robot pozicionirao u zadanu točku i vratio u početnu poziciju ponavlja se mogućnost unosa za drugu točku. Ukoliko se neke vrijednosti ne trebaju mijenjati unosi se 0 (nula).

Prilikom unosa vrijednosti 999 u bilo koji od tri gornja parametra KAREL program se treba prekinuti tj. završiti izvođenje.

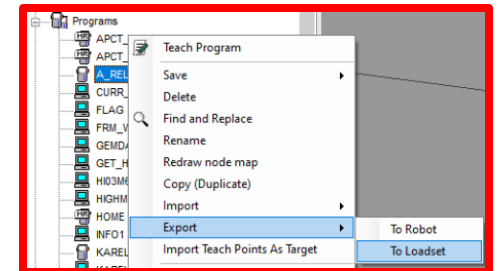
\*Neke od korisnih KAREL funkcija: GET\_REG(), SET\_POS\_REG(), CALL\_PROG(), SET\_INT\_REG(),  
WRITE(), READ(), CNV\_STR\_REAL(), UNINIT(), ROUND(),  
\$GROUP[1].\$UFRAME = \$MNUFRAME[1,9], \$GROUP[1].\$UTOOL = \$MNUTOOL[1,1], CURPOS(0,0)  
primjer: **CALL\_PROG('NAZIV\_PROGRAMA',prog\_idx)**      *-- poziv TP programa naziva CIRC \**  
prog\_idx treba biti deklariran kao **INTEGER**

# Roboguide-FANUC zadatak 1 – 15 bodova (nastavak)

Uspješno prenijeti funkcionalan program paletizacije iz Roboguide-a na proizvoljan pravi robot u laboratorij.

Potrebno je predati izvorni KAREL program i LS program. Naziv datoteka: prezime\_ime\_01.KL i prezime\_ime\_DZ.LS

\*LS program moguće je *exportirati* iz Roboguide-a koristeći naredbu Export to Loadset:



\*Potrebno je predati video ili pokazati rad robota u laboratoriju.

\*Sve kretnje robota u simulaciji i u laboratoriju mogu biti definirane u „zraku” bez rada/kontakta s pravim predmetima.



# Roboguide-FANUC zadatak 2 – 15+15 bodova

## ZADATAK:

Sve niže navedene zadatke moguće je validirati na proizvoljnom robotu u laboratoriju. Za dodatnu validaciju u laboratoriju predviđeno je dodatnih 15 bodova na redovnih 15 boda.

Prije rada na realnom robotu sve je potrebno testirati i napraviti u Roboguide-u.

Potrebno je napisati programe te samostalno odrediti način izvođenja iz TP programa (RUN, CALL) kao i odabir potrebnih KAREL funkcija. Broj zasebnih programa nije određen i samostalno možete osmisлити strukturu rješenja.

Rad robota treba imati sljedeću funkcionalnost:

1. Prilikom pokretanja glavnog programa (MAIN.LS) na robotu potrebno je pokrenuti Karel (+TP) program iz zadatka 1
2. Potrebno je napisati dodatan program (ili funkcionalnost integrirati u postojeće programe) (koji će služiti kao *data logger*) koji će putem TELNET-a računalu u laboratoriju i u simulaciji vraćati sljedeće podatke:
  1. Proizvoljnim redoslijedom datum (dan, mjesec, godina), trenutno vrijeme (sat, minuta, sekunda) i jedinstveni identifikator vremena u milisekundama (sistemska varijabla \$FAST\_CLOCK u KAREL-u, nije ju potrebno deklarirati) \* za trenutno vrijeme moguće je koristiti funkciju GET\_TIME() te prema potrebi dodatne funkcije kako bi se od pojedinih bitova izvukli podatci o sekundama...
  2. Pozicija robota (XYZ u aktivnom koordinatnom sustavu) – (koristiti funkciju CURPOS(0,0) i prema potrebi sistemske varijable za definiciju aktivnih koordinatnih sustava u KAREL-u
  3. Aktivni koordinatni sustav (broj/integer)
  4. Kutevi zakreta svih 6 zglobova zapisani sa tri decimalna mjesta (koristiti funkcije CURJPOS(0,0) i CNV\_JPOS\_REL ())
  5. Ako je zastava 1 tj.FLG[1]/F[1]=ON/TRUE putem TELNETA podatci se šalju samo kada robot izvrši gibanje u novu točku
  6. Ako je zastava 2 tj.FLG[2]/F[2]=ON/TRUE putem TELNETA podatci se šalju kontinuirano (odabrati proizvoljnu frekvenciju u rasponu od 1 do 5 Hz)
  7. Ako su obje zastave 1 ili obje 0 ne šalje se ništa.
  8. Nakon prekida glavnog programa koji se izvršava unosom „999” na USER ekranu potrebno je osigurati uredno zatvaranje TELNET veze i prekid svih programa koji su se paralelno izvodili



# Roboguide-FANUC zadatak 3 – 25 bodova

## ZADATAK:

Potrebno je napraviti KAREL i LS/TP program(e) koji će učitati podatke iz datoteke KOORDINATE.txt te na temelju toga generirati i pokrenuti program **robotiziranog koordinatnog bušenja tj. pozicioniranja**.

U KOORDINATE.txt datoteci nalaze se redom sljedeći podatci odvojeni praznim mjestima („space“):

- x koordinata [mm], y koordinata [mm], linearni posmak/brzina [mm/s], dubina bušenja [mm]
- Dubina bušenja (paralelna sa z osi alata) je opcionalni parametar, ukoliko nije navedena u TXT-u koristiti vrijednost od 25 mm
- U prilogu se nalazi primjer TXT datoteke – npr. red 3 je neispravan i treba ga preskočiti, ostali redovi su ispravni
- Maksimalan broj točaka/redova u TXT datoteci je 200
- Za sva gibanja koristiti UFRAME[0] i UTOOL[0], program testirati na modelu robota kao u CRTA-i (u programu navesti model robota u komentaru)
- Početna (home) točka programa treba biti: 

J1:	0.000	J2:	0.000	J3:	0.000
J4:	0.000	J5:	-90.000	J6:	0.000

KAREL program treba ispravno pročitati sve podatke te pokrenuti TP/LS program koji pozicionira robota u sve točke iz TXT-a.

## Napomene:

- napraviti sve potrebne provjere prilikom generiranja kretnji/točaka/PR[], kao i čitanja TXT datoteke, moguće je koristiti princip čitanja znak po znak *read file\_var(s::1)*
- TP/LS program je moguće pozivati iz KAREL programa tako da se za cijeli proces koristi samo jedan PR[] za pozicioniranje i jedan PR[] za tool offset.
- Koristiti 'NUT000' konfiguraciju robotske ruke
- Neke od korisnih KAREL funkcija i programa s e-učenja:
  - CHECK\_EPOS(), SET\_POS\_REG(), CALL\_PROG(), REACH.KL, MAIN\_KL.KL, RWTEST2.KL,

Potrebno je predati izvorne KAREL i LS programe. Ime datoteka– „PREZIME\_IME\_03.KL/LS”





# Roboguide-FANUC zadatak 3 -Prilog

KOORDINATE.txt:

```
150 200 300 50  
200 300 200 20  
100 100  
310.15 0 100  
-50.13 120.15 150
```