

Elektrotehnički fakultet
Univerzitet u Beogradu

PRVI DOMAĆI ZADATAK IZ ROBOTIKE I
AUTOMATIZACIJE

Mentor:

Zaviša Gordić

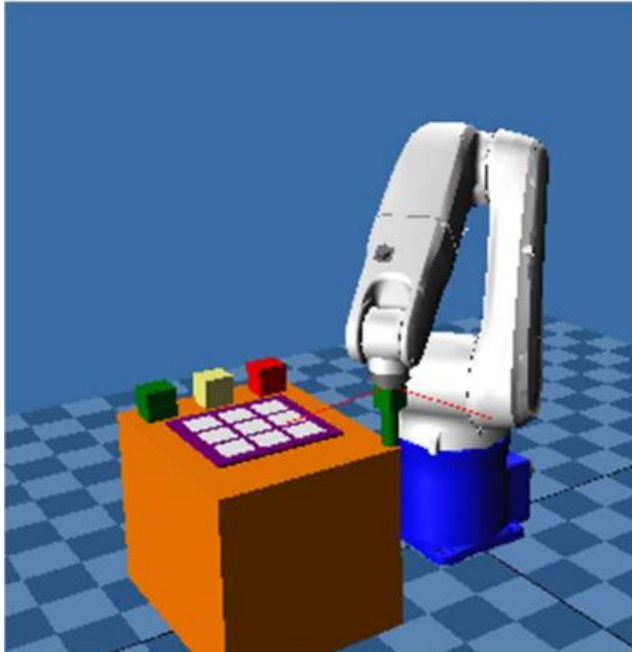
Studenti:

Marina Svilar 2020/0411

Teodora Predojević 2020/0447

Beograd, jul 2023. godina

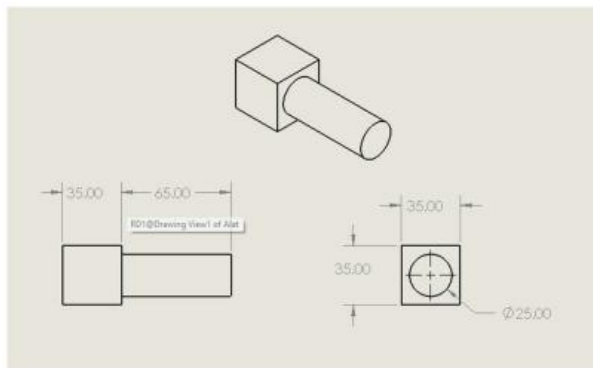
Izgled radnog okruženja U programskom okruženju WINCAPSIII smo realizovali naš radni sistem. Radni sistem se sastoji iz postolja, palete i magacina sa kockicama različitih boja. Magacin sa kockicama je namešten kao što je prikazano na slici uz zadatak. Paleta se sastoji od devet polja. Definisana su dva kordinatna sistema, work0 koji je u bazi robota i work1 koji ke vezan za centar na gornjoj površini postolja. Kockice i paleta koji se nalaze na postolju su vezani za samo postolje.



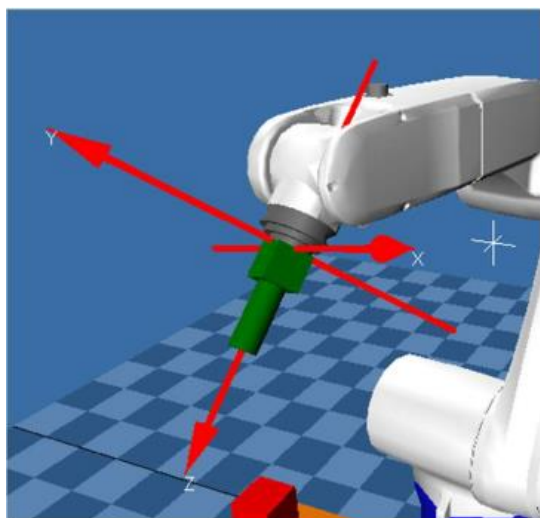
Slika1. Radno okruzenje

Work			
Jump		Smart View	Easy setting
No.	X	Y	Z
1	400	0	300
2	0	0	0

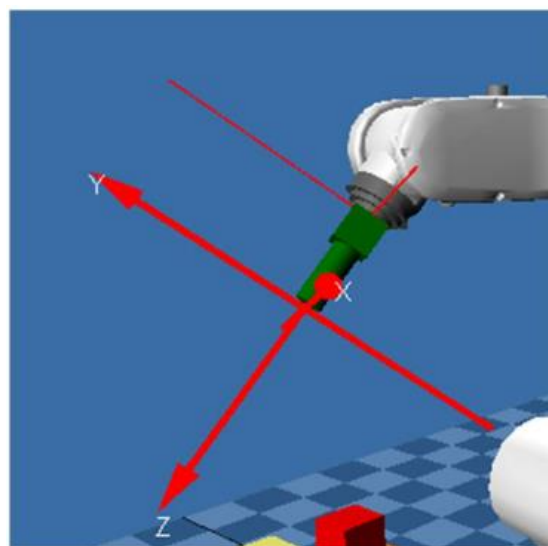
Slika2. Definisanje položaja sistema work1 u odnosu na sistem baze robota



Slika3. Završni uredjaj



Slika6. Koordinatni sistem na na završnom uredjaju



Slika5. Kordinatni sistem koji se nalazi na samom vrhu hvataljke

Sada definišemo i koordinatni sistem koji se nalazi na samom vrhu hvataljke kako bismo jednostavnije upravljali kretanjem završnog uređaja.

Tool			
Jump		Smart View	Easy setting
No.	X	Y	Z
1	0	0	100

Slika6. Koordinatni sistem na vrhu hvataljke

Definisane su odgovarajuće tačke u prostoru koje se odnose na položaje kockica, polja na paleti i neutralnog položaja robota.

Type P									
Jump		Smart View	Get Position	Move	ExJoint				
No.	X	Y	Z	RX	RY	RZ	FIG	Usage	
0	-140.0581	191.8314	-1.705303...	180	0	101.4081	5 - Lefty A	W1 T1:	
1	-100	-150	50	180	9.262476E...	138.3985	5 - Lefty A	W1 T1:	
2	0	-150	50	180	0	138.3985	5 - Lefty A	W1 T1:	
3	100	-150	50	180	0	138.3985	5 - Lefty A	W1 T1:	
4	60	60	10	180	0	138.3985	5 - Lefty A	W1 T1:	
5	60	0	10	180	0	138.3985	5 - Lefty A	W1 T1:	
6	60	-60	10	180	7.830645E...	138.3985	5 - Lefty A	W1 T1:	
7	0	60	10	180	-5.987645...	138.3985	5 - Lefty A	W1 T1:	
8	0	0	10	180	-7.20676E...	138.3985	5 - Lefty A	W1 T1:	
9	0	-60	10	180	0	138.3985	5 - Lefty A	W1 T1:	
10	-60	60	10	180	0	138.3985	5 - Lefty A	W1 T1:	
11	-60	0	10	180	0	138.3985	5 - Lefty A	W1 T1:	
12	-60	-60	10	180	-2.723751...	138.3985	5 - Lefty A	W1 T1:	
13	0	0	0	0	0	0	-1 - Undefir		

Slika7. Def. Tacaka u prostoru

Planiranje kretanja robota

Ideja je da se robot iz neutralnog položaja (koji je definisan tako da ne ometa snimanje kamere) kreće do zahtevane kockice i da je premesti na zahtevano polje na paleti. To možemo učiniti tako što će robot doći na određenu visinu iznad kockice, spustiti se do kockice, aktivirati hvataljku (io24), uhvatiti kockicu, doći na određenu visinu iznad odgovarajućeg polja, polako spustiti kockicu i otpustiti hvataljku(io24). Korisnički unos podataka u Type1 (korisnik unosi broj xy gde je x broj kockice, a y odgovarajuće polje na paleti):

Type I		
Jump		Smart View
No.	Value	Usage
0	15	
1	26	
2	31	
3	33	
4	11	
5	0	
6	0	

Slika8. Korisnicki unos

Pri unosu 0, podrazumeva se da korisnik ne želi više da unosi parametre. Realizacija datog problema je implementirana kroz sledeći kod:

```
robot1prog.pac Arm 3D View
001 'TITLE "<Title>"
002 PROGRAM Robot1Prog
003
004 #INCLUDE "DIO_TAB.H" 'Do not delete if you wish to keep Macro namase for IO. Activation (deactivation) iof the output is perfomed as
005 dim tmpx as double
006 dim tmpy as double
007 dim tmprz as double
008
009
010 '===== Beginning of the code for communication with camera =====
011 goto *CameraInfo      'insert this AND the following line when you want to get information from camera
012 *CameraReply:
013 '===== End of the code for communication with camera =====
014
103 '===== Do NOT modify following content =====
104 *CameraInfo:
105 'Function provides shape, size and orientation information of the detected object.
106 'Function is called using <gosub *CameraInfo> ststatement.
107 '
108 'Positions (x,y) of the pallet in camera coordinate frame are stored in variables (f1,f2)
109 'Orientation is stored in global float variable f3, and it represents rotation about vertical axis z.
110 tmpx = RND (-TIMER)
111 tmpy = RND (-TIMER+150)
112 tmprz = RND (-TIMER+400)
113
114 f1=(tmpx-0.5)*35      'along work table's x-axis
115
116 f2=(tmpy-0.5)*50+70   'along work table's y-axis
117
118 f3=(tmprz-0.5)*360    'orientation - Rz
119 goto *CameraReply
120 '===== Do NOT modify preceeding content =====
121
```

Slika9. Funkcija koja oponasa rad kamere

```
015 'Provera da li je korisnik aktivirao input 15
016 if io15 then
```

```
017
018     takearm
019     motor on
020
021     changework 1
022     changetool 1
023
024     dim brojac as integer
025     brojac = 0
026
027     dim pom as integer
028     pom = 0
029
030     dim br_kocke as integer
031     dim br_polja as integer
032     dim br_poljal as integer
033
034     br_kocke = 0
035     br_polja = 0
```

```
036     br_poljal = 0
```

```
037
038     'Pomocne promenljive za zahtev2
039     dim br0 as integer
040     br0 = 0
041     dim br1 as integer
042     br1 = 0
043     dim br2 as integer
044     br2 = 0
045     dim br3 as integer
046     br3 = 0
047     dim brojac1 as integer
048     brojac1 = 0
```

```
049
050     'Niz identifikatora koji su inicijalizovani na 0 a cuvaju br_kockice na odredjenoj poziciji.
```

```

051     Dim ind(9) As integer
052     do
053         ind(brojac) = 0
054         brojac = brojac + 1
055         loop until brojac>8
056         brojac = 0
057
058         move p, p0, s=70
059
060     do
061         pom = i[brojac]
062
063         'korisnik unosi broj kocke i polja dok ne unese 0 sto tumacimo kao da ne zeli dalji unos
064         if pom = 0 then
065             move p, p0, s=70
066             motor off
067             givearm
068         end if
069
070         'Dobijanje broja kockice i pozicije kockice iz unosa korisnika.
071         br_poljal = pom mod 10
072         br_polja = br_poljal + 3
073
074         br_kocke = pom - br_poljal
075         br_kocke = br_kocke / 10
076
077         'Ako je identifikator na na poziciji jednak 0 ili br_kockice koju je korisnik zadao da zeli na tu poziciju da stavi onda se izvršava kod.
078         'Zahtev1: Dve kockice razlicitih boja ne smeju biti na istoj poziciji.
079         if ind(br_poljal - 1) = 0 or ind(br_poljal - 1) = br_kocke then
080
081             approach p, p[br_kocke], 100, s=70
082
083             move l, @e p[br_kocke], s = 70
084
085             'picking point
086             'hvataljka je aktivirana - zatvorena
087             set io24
088
089             delay 500
090
091             depart l, 150, s=10
092
093             'placing point
094             approach l, p[br_polja], 150, s=10
095             approach l, p[br_polja], 55.5, s=10
096
097             'hvataljka je deaktivirana - otvorena
098             reset io24
099
100         else
101             move p, p0, s=70
102             motor off
103             givearm
104         end if
105
106         ind(br_poljal - 1) = br_kocke
107
108         'Zahtev2: Potrebno je da budu rasporedjene kockice sve tri boje u jednoj iteraciji ili
109         'se preostale kockice rasporeduju na prazne pozicije ako ih ima na paleti ako nema onda se obustavlja rad robota.
110         if brojac = 8 then

```



```

150         set io24
151
152         delay 500
153
154         depart l, 150, s=10
155
156         'placing point
157         approach l, p[brojac1 + 1 + 3], 150, s=10
158         approach l, p[brojac1 + 1 + 3], 55.5, s=10
159
160         'hvataljka je deaktivirana - otvorena
161         reset io24
162
163         br1 = br1 + 1
164         ind(brojac1) = 1
165
166         end if
167
168     elseif br2 = 0 then
169         if ind(brojac1) = 0 then
170
171             elseif br2 = 0 then
172                 if ind(brojac1) = 0 then
173
174                     approach p, p2, 100, s=70
175
176                     move l, @e p2, s = 70
177
178                     'picking point
179                     'hvataljka je aktivirana - zatvorena
180                     set io24
181
182                     delay 500
183
184                     depart l, 150, s=10
185
186                     'placing point
187                     approach l, p[brojac1 + 1 + 3], 150, s=10
188                     approach l, p[brojac1 + 1 + 3], 55.5, s=10
189
190                     'hvataljka je deaktivirana - otvorena
191                     reset io24

```

```

190         br2 = br2 + 1
191         ind(brojac1) = 2
192
193     end if
194
195 elseif br3 = 0 then
196     if ind(brojac1) = 0 then
197
198         approach p, p3, 100, s=70
199
200         move 1, @e p3, s = 70
201
202         'picking point
203         'hvataljka je aktivirana - zatvorena
204         set io24
205
206         delay 500
207
208         depart 1, 150, s=10
209
210         'placing point
211         approach 1, p[brojac1 + 1 + 3], 150, s=10
212         approach 1, p[brojac1 + 1 + 3], 55.5, s=10
213
214         'hvataljka je deaktivirana - otvorena
215         reset io24
216
217         br3 = br3 + 1
218         ind(brojac1) = 3
219
220     end if
221
222 end if
223
224 brojac1 = brojac1 + 1
225 loop until brojac1 > 8
226
227 end if
228
229 brojac = brojac + 1
230 loop until brojac > 9

```

```

231
232
233     move p, p0, s=70
234     motor off
235     givearm
236
237 else
238     'Ako korisnik nije aktivirao io15 ceka se 2s pa se ponovo proverava io15
239     delay 2000
240 end if
241
242 END
243
244

```

Ovo je funkcija koja oponaša rad kamere i daje položaj palete u odnosu na koordinatni sistem vezan za površinu postolja

Proveravamo da li je korisnik aktivirao digitalni ulaz io15 pre početka paletiranja, ako jeste, paletiranje može da počne, a ako nije onda se čeka 2s kako bi se taj ulaz aktivirao. U okviru do while petlje izvršavamo paletiranje tako što čitamo podatke iz tabele. Iz podatka koji pročitamo dobijamo informaciju o željenoj kockici i njenom položaju na paleti. Zahtev da se ne mogu dve kockice različite boje staviti na isto polje na paleti ispunjavamo tako što konstruišemo niz indikatora sa inicijalnim vrednostima nula, gde je svaki član niza odgovarajuća pozicija na paleti. Ako je neka boja smeštena na odredjeno polje onda se indikator tog polja promeni na vrednost te kockice. Upoređivanjem indikatora na trenutnoj poziciji i boje kockice možemo zaključiti da li je to polje već zauzeto nekom kockicom druge boje, u tom slučaju, robot se vraća u neutralni položaj i završava svoj rad.