

1 Beispielanwendung mit speichern der Daten

inputencodinginputencoding

```
1 def myProg():
2
3     global richtung=False
4
5     // festlegen der maximalen beschleunigung über die a_max variable
6     global a_max = d2r(40)
7
8     // festlegen er maximalen geschwindigkeit über die v_max variable
9     global v_max = d2r(60)
10
11    // Startpunkt für blaue seite des würfels
12    global blue_deg = [d2r(79.34), d2r(-130.29), d2r(-108.28), d2r(-31.43), d2r
        (-90.0), d2r(-1.02)]
13
14
15    // Funktion rotiert den Würfel über die übergebene Achse (1= X-Achse, 2=Y-
        Achse, 3=Z-Achse) um a grad
16    def rotate_axis(a, b):
17        rotationPos=p[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
18        a= d2r(a)
19        if b == 1:
20            rotationPos = p[0.0, 0.0, 0.0, a, 0.0, 0.0]
21        elif b == 2:
22            rotationPos = p[0.0, 0.0, 0.0, 0.0, a, 0.0]
23        elif b == 3:
24            rotationPos = p[0.0, 0.0, 0.0, 0.0, 0.0, a]
25        end
26        movel(pose_add(get_actual_tcp_pose(), rotationPos), a=0.5, v=0.1)
27    end
28
29
30    // Funktion dreht den letzten Joint bei digital in HIGH zur Nächsten form um
        71.299 grad, Bis alle formen auf der Ebene erreicht wurden
31    def move_toy():
32        schleife_1 = 4
33        while schleife_1 > 0:
34            while digital_in[1] == False:
35                sync()
```

```
36     end
37     r = d2r(71.299)
38     pos = get_actual_joint_positions()
39     pos= [pos[0], pos[1], pos[2], pos[3], pos[4], pos[5]+r]
40     movej(pos, a=a_max, v=v_max)
41     schleife_1=schleife_1-1
42     end
43 end

44 // justiert den roboter in die richtung vorgegeben von der variable direction
45 thread justify():
46     if direction == True:
47         move_z=0.5
48     else:
49         move_z=-0.5
50     end
51     movel(pose_add(get_actual_tcp_pose(), p[0.0,0.0, move_z, 0.0,0.0,0.0]), a
52           =0.1, v=0.15)
53 end

54 // Funktion Fragt Spieler ob der Roboter in der höhe Justiert werden soll
55 def justify_height():
56     // Ja/Nein anfrage vom Benutzer holen ob justiert wird
57     global justieren = request_boolean_from_primary_client("roboter justieren?"
58     )
59     varmsg("justieren", justieren)
60     while (justieren == True ):
61         // Ja/Nein anfrage vom Benutzer holen in welche richtung justiert wird
62         global richtung = request_boolean_from_primary_client("hoch ~> ja ;
63         runter ~> nein")
64         varmsg("richtung", richtung)
65         # wait until button 1 is pressed
66         while digital_in[1] == False:
67             sync()
68         end
69         // justify function bewegt den Roboter solange der digitale eingang auf
70         high steht in die vorgegebene richtung
71         justify_handler= run justify()
72         while digital_in[1] == True:
73             sync()
74         end
75         kill justify_handler
76         blue_deg= get_actual_joint_positions()
77         justieren = request_boolean_from_primary_client("roboter erneut justieren
78         ?")
79         varmsg("justieren", justieren)
80     end
81 end

82 // funktion fragt ob ein neuer spieler erstellt werden soll, wenn ja, wird ü
83 ber die socket verbindung der spieler mit dem neuen Namen gespeichert.
84 def spieler_erst():
```

```
82     neu = request_boolean_from_primary_client("Neuen Spieler erstellen?")
83     name = request_string_from_primary_client("Name des Spielers?")
84     if neu == True:
85         socket_send_string("new_patient")
86         socket_send_string(name)
87     end
88 end

90 // Abfrage vom programm bei programmstart ob das programm weiterlaufen soll.
91 start_var = request_boolean_from_primary_client("program starten?")
92 varmsg("start_var", start_var)

94 // öffnen der socket verbindung zum Daten Server
95 data_server = socket_connect("141.100.101.48", 8000)

98 if start_var == True:
99     movej(blue_deg, a=a_max, v=v_max)
100     if( data_server):
101         spieler_erst()
102     else:
103         global p_name = request_string_from_primary_client("name des Spielers?")
104         varmsg("p_name", p_name)
105     end
106     justify_height()

108 // laufe solange gespielt werden soll
109 while start_var == True:
110     movej(blue_deg, a=a_max, v=v_max)
111     popup("Wenn Wuerfel entleert ist, kann losgelegt werden(Mit Button 1 den
112           Wuerfel drehen.)", "Nachricht", False, False)
113     move_toy()
114     movej(red_deg, a=a_max, v=v_max)
115     rotate_axis(-40, 1)
116     move_toy()
117     start_var = request_boolean_from_primary_client("erneut beginnen?")
118     varmsg("start_var", start_var)
119 end
120 popup("programm wird beendet", "Nachricht", False, False)
121 socket_close()
122 end
```