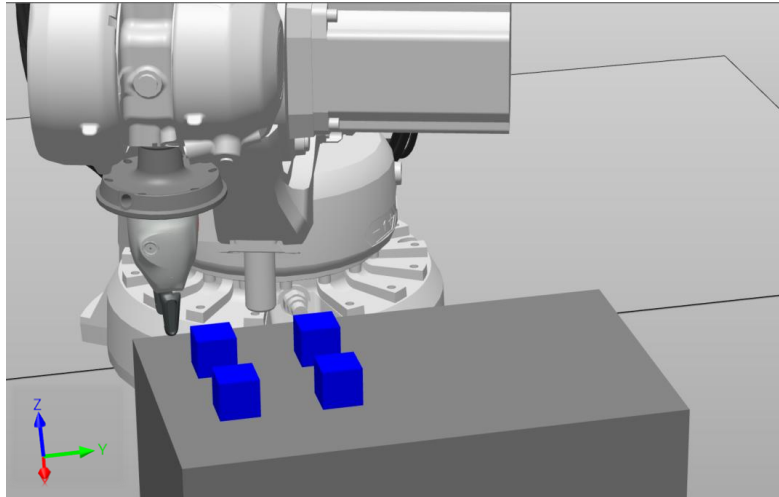


Aufgabenstellung: Umlagern- und Suchaufgabe mit ABB RAPID

Szenario

Ein ABB IRB 4600 Roboter wird zum Platzieren von 4 Boxen eingesetzt und ist dafür mit einem Greifer („Servo“) ausgestattet. Er soll vier Boxen (40 x 40 mm) die in einer 2x2 Matrix (mit 100 mm Abstand) angeordnet sind um 200 mm in Y Richtung versetzen (siehe Bild).



Der Roboter verfügt über:

- Eine definierte **Home-Position** (Ruheposition).
- Einen Referenzpunkt **Target_10** als Startpunkt für das Sortieren (Greifpunkt linke hintere Box).
- Einen Greifer („Servo“), der geöffnet/geschlossen werden kann.
- Einen **Linien-Sensor (diLineSensor)** zur Erkennung von Boxen.

Aufgabe 1

1. Pseudocode erstellen

- Definieren Sie Konstanten und Startbedingungen (Home, Target_10, distBox).
- Strukturieren Sie das Hauptprogramm mit Unterprogrammen:
 - Relocate (Umlagern, Schleife)
 - PickAndPlace (Aufnehmen und Ablegen)
- Nutzen Sie Schleifen zur systematischen Bearbeitung der Boxen.
- Beschreiben Sie die Bewegungsabläufe (PTP, Linear, Wartezeiten, Greiferaktionen).

2. RAPID-Programm schreiben

- Implementieren Sie den Pseudocode in RAPID.
- Verwenden Sie `PROC` für die Unterprogramme und ein `Main`-Programm als Einstieg.
- Nutzen Sie Variablen für Positionsrechnungen (`x_pos`, `y_pos`, `CurrentTarget`).
- Verwenden Sie die Bewegungsbefehle (`MoveJ`, `MoveL`) mit unterschiedlichen Geschwindigkeiten und Genauigkeiten (`v100`, `z10`, `fine`).
- Stellen Sie sicher, dass der Roboter am Ende in die Home-Position zurückkehrt.

Aufgabe 2

Im Produktivbetrieb zeigt sich, dass die beiden linken Boxen zu Beginn produktionsbedingt nicht korrekt positioniert sind und eine Abweichung in positiver bzw. negativer Y-Richtung aufweisen. Diese Abweichung liegt im Bereich von ± 150 mm. Daher müssen die beiden Boxen vor dem Umplatzen zunächst von dem Roboter auf ihre korrekte Position gesetzt werden. Zur Erfassung der tatsächlichen Boxposition steht am Endeffektor ein nach unten (in Z-Richtung) ausgerichteter Lasersensor zur Verfügung, der Objekte innerhalb eines Bereichs von 150 mm detektieren kann.

1. Pseudocode erstellen

- Erweitern Sie den Code aus Aufgabe 1.
- Ergänzen Sie die Unterprogrammen:
 - `SearchAndPlace` (Suche und Platzieren, Schleife)
 - `SearchPickAndPlace` (Suche nach falsch stehenden Boxen und Umplatzieren)
- Nutzen Sie Schleifen zur systematischen Bearbeitung der Boxen.
- Beschreiben Sie die Bewegungsabläufe (PTP, Linear, Wartezeiten, Greiferaktionen).

2. RAPID-Programm schreiben

- Implementieren Sie den Pseudocode in RAPID.
- Verwenden Sie `PROC` für die Unterprogramme und ein `Main`-Programm als Einstieg.
- Nutzen Sie Variablen für Positionsrechnungen (`x_pos`, `Start_Search_Target`, `Found_Pick_Pos`, `Scan_Check_Pos`, `StartPos_Found`).
- Verwenden Sie die Bewegungsbefehle (`MoveJ`, `MoveL`) mit unterschiedlichen Geschwindigkeiten und Genauigkeiten (`v100`, `z10`, `fine`).
- Stellen Sie sicher, dass der Roboter am Ende in die Home-Position zurückkehrt.
- Implementieren Sie die Sensorabfrage (`diLineSensor`) zur Boxerkennung.

LÖSUNG - Pseudocode:

Konstanten & Definitionen

Home: Definierte Ruheposition des Roboters

Target_10: Referenz-Startpunkt für das Sortieren

Hauptprogramm: Main

Aktion: Greifer öffnen (Initialisierung)

Fahre PTP (schnell, Überschleifen) zu Position: Home

Rufe Unterprogramm: **SearchAndPlace** auf.

Rufe Unterprogramm: **Relocate** auf

Fahre PTP (schnell, Überschleifen) zu Position: Home

Unterprogramm: Relocate

Wiederhole für x_index von 0 bis 1

Wiederhole für y_index von 0 bis 1

Multipliziere zu: $x_pos = x_index * 100$

Multipliziere zu: $y_pos = y_index * 100$

Rufe Unterprogramm: **PickAndPlace** (x_pos, y_pos)

Ende Schleife

Ende Schleife

Unterprogramm: PickAndPlace (x_rel, y_rel)

Berechnung: CurrentTarget = Position: Target_10 + x_rel in X und + y_rel in Y

Ablauf:

1. *Vorposition:* Fahre PTP (schnell, Überschleifen) zu Position: CurrentTarget + 80mm in Z
2. *Greifposition:* Fahre linear (langsam, fein) zu Position: CurrentTarget
3. *Aktion:* Greifer schließen
4. *Warte:* 0,3 Sekunden
5. *Anheben:* Fahre linear (langsam, Überschleifen) zu Position: CurrentTarget + 80mm in Z
6. *Transport:* Fahre linear (schnell, Überschleifen) zu Position: CurrentTarget + 200mm in Y + 80mm in Z
7. *Ablageposition:* Fahre linear (langsam, fein) zu Position: CurrentTarget + 200mm in Y
8. *Aktion:* Greifer öffnen
9. *Warte:* 0,2 Sekunden
10. *Wegfahren:* Fahre linear (langsam, Überschleifen) zu Position: CurrentTarget + 200mm in Y + 80mm in Z

Unterprogramm: SearchAndPlace (Vorwärts-Sortierung)

Wiederhole für x_index von 0 bis 1

Multipliziere zu: $x_pos = x_index * 100$

Rufe Unterprogramm: **SearchPickAndPlace** (x_pos)

Ende Schleife

Unterprogramm: SearchPickAndPlace (x_rel)

Berechnung Startpunkt für die Suche: Start_Search_Target = Position: Target_10 + x_rel in X & -200mm in Y & + 80mm in Z

Ablauf:

1. *Startposition:* Fahre PTP (schnell, Überschleifen) zu Position: Start_Search_Target.
2. *Suche Box (Erste Reihe):* Fahre linear (langsam) von Start_Search_Target in Y-Richtung (+1000mm) bis der Sensor (diLineSensor) schaltet.
3. *Speichere die Position, wenn Box gefunden:* in StartPos_Found

4. Wenn Box gefunden (diLineSensor = 1) dann:
 1. *Berechnung Greifposition*: Found_Pick_Pos = Position:
StartPos_Found + 20mm in Y
 2. *Berechnung Prüfposition nächste Box*: Scan_Check_Pos = Position:
Found_Pick_Pos + 41mm in Y
 3. *Zwischenposition*: Fahre linear (schnell, fein) zu Position:
Scan_Check_Pos.
 4. *Suche der zweiten Box in der Reihe*: Fahre linear (langsam) von
Scan_Check_Pos in Y-Richtung (+200mm) bis der Sensor
(diLineSensor) schaltet.
 5. *Speichere die Position, wenn Box gefunden*: in Scan_Check_Pos
 6. Wenn Box gefunden (diLineSensor = 1) dann:
 1. *Vorposition Pick*: Fahre linear (schnell, fein) zu Position:
Found_Pick_Pos + 80mm in Z.
 2. *Greifposition*: Fahre linear (langsam, fein) zu Position:
Found_Pick_Pos - 80mm in Z.
 3. *Aktion*: Greifer schließen (Set Greifer).
 4. *Warte*: 0.3 Sekunden.
 5. *Anheben*: Fahre linear (langsam, Überschleifen) zu Position:
Found_Pick_Pos + 80mm in Z.
 6. *Transport*: Fahre linear (schnell, Überschleifen) zu Position:
Scan_Check_Pos - 80mm in Y + 80mm in Z (Ablage-Überfahrt).
 7. *Ablageposition*: Fahre linear (langsam, fein) zu Position:
Scan_Check_Pos - 80mm in Y - 80mm in Z.
 8. *Aktion*: Greifer öffnen (Reset Greifer).
 9. *Warte*: 0.2 Sekunden.
 10. *Wegfahren*: Fahre linear (langsam, Überschleifen) zu Position:
Scan_Check_Pos - 80mm in Y + 80mm in Z.
 7. *Sonst Ausgabe*: "Keine Box in der ersten Reihe gefunden"
5. *Sonst Ausgabe*: "Keine Box in der zweiten Reihe gefunden";

Lösung - Aufgabe 1 und 2 – RAPID:

MODULE Module1

```
CONST robtarget Home:=[[0.000021,-0.000012419,199.999981547],[-0.000000015,1,-0.000000057,-0.000000014],[-1,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget Target_10:=[[125,125,30],[0,1,0,0],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR num distBox := 100;
```

```
PROC main()
```

```
MoveJ Home,v1000,z100,Servo\WObj:=Workobject_4;
```

```
SearchAndPlace;
```

```
Relocate;
```

```
MoveJ Home,v1000,z100,Servo\WObj:=Workobject_4;
```

```
ENDPROC
```

```
PROC Relocate()
```

```
VAR num x_index;
```

```
VAR num y_index;
```

```
FOR x_index FROM 0 TO 1 DO
```

```
FOR y_index FROM 0 TO 1 DO
```

```
PickAndPlace x_index * distBox,y_index * distBox;
```

```
ENDFOR
```

```
ENDFOR
```

```
ENDPROC
```

```
PROC PickAndPlace(Num x_rel, Num y_rel)
```

```
VAR robtarget CurrentTarget;
```

```
CurrentTarget := Offs(Target_10, x_rel, y_rel, 0);
```

```
MoveJ Offs(CurrentTarget, 0, 0, distBox),v1000,z50,Servo\WObj:=Workobject_4;
```

```
MoveL CurrentTarget,v500,fine,Servo\WObj:=Workobject_4;
```

```
Set Greifer;
```

```
WaitTime 0.3;
```

```
MoveL Offs(CurrentTarget, 0, 0, distBox),v500,z50,Servo\WObj:=Workobject_4;
```

```
MoveL Offs(CurrentTarget, 0, distBox*2, distBox),v1000,z50,Servo\WObj:=Workobject_4;
```

```
MoveL Offs(CurrentTarget, 0, distBox*2, 0),v500,fine,Servo\WObj:=Workobject_4;
```

```
Reset Greifer;
```

```
WaitTime 0.2;
```

```
MoveL Offs(CurrentTarget, 0, distBox*2, distBox),v500,z50,Servo\WObj:=Workobject_4;
```

```
ENDPROC
```

```
PROC SearchAndPlace()
```

```
VAR num x_index;
```

```
FOR x_index FROM 0 TO 1 DO
```

```
SearchPickAndPlace x_index * distBox;
```

```
ENDFOR
```

```
ENDPROC
```

```
PROC SearchPickAndPlace(Num x_rel)
```

```
VAR robtarget Start_Search_Target;
```

```
VAR robtarget Found_Pick_Pos;
```

```
VAR robtarget Scan_Check_Pos;
```

```
VAR robtarget StartPos_Found;
```

```

Start_Search_Target := Offs(Target_10, 0 + x_rel, -distBox*2, distBox);
MoveJ Start_Search_Target,v1000,z50,Servo\WObj:=Workobject_4;

SearchL \Stop, diLineSensor, StartPos_Found, Offs(Start_Search_Target, 0, 1000, 0), v80,
Servo\WObj:=Workobject_4;
IF diLineSensor = 1 THEN
    Found_Pick_Pos := Offs(StartPos_Found, 0, 20, 0);
    Scan_Check_Pos := Offs(Found_Pick_Pos, 0, 41, 0);
    MoveL Scan_Check_Pos, v1000, fine, Servo\WObj:=Workobject_4;
    SearchL \Stop, diLineSensor, Scan_Check_Pos, Offs(Scan_Check_Pos, 0, distBox*2, 0), v80,
    Servo\WObj:=Workobject_4;
    IF diLineSensor = 1 THEN
        MoveL Offs(Found_Pick_Pos, 0, 0, distBox), v1000, fine, Servo\WObj:=Workobject_4;
        MoveL Offs(Found_Pick_Pos, 0, 0, -distBox), v500, fine, Servo\WObj:=Workobject_4;
        Set Greifer;
        WaitTime 0.3;
        MoveL Offs(Found_Pick_Pos, 0, 0, distBox), v500, z50, Servo\WObj:=Workobject_4;
        MoveL Offs(Scan_Check_Pos, 0, -distBox +20, distBox), v1000, z50,
        Servo\WObj:=Workobject_4;
        MoveL Offs(Scan_Check_Pos, 0, -distBox +20, -distBox), v500, fine,
        Servo\WObj:=Workobject_4;
        Reset Greifer;
        WaitTime 0.2;
        MoveL Offs(Scan_Check_Pos, 0, -distBox +20, distBox), v500, z50,
        Servo\WObj:=Workobject_4;
    ELSE
        TPWrite "Keine Box in der ersten Reihe gefunden";
    ENDIF
ELSE
    TPWrite "Keine Box in der zweiten Reihe gefunden";
ENDIF
ENDPROC

ENDMODULE

```